GSE 60 SERIES

Technical Reference Manual

PROGRAMMABLE PROCESS CONTROLLERS

VERSION 3.0

Part Number: 39-10-X60REF

GSE scale systems

GSE 60 Series Programmable Process Controllers Technical Reference Manual

Published by:

GSE Scale Systems
22705 Heslip Drive
Novi, MI 48375
USA

Information in this *Technical Reference Manual* is subject to change *without notice* due to correction or enhancement. The information described in this manual is solely the property of GSE. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording and sold for any monetary figure without the express written permission of GSE.

This *Technical Reference Manual* is a reference tool that explains installation, setup, and operation of the Model 60 Series Instrumentation. Due to the programmable nature of these controllers, your application configuration is likely to be unique. Any questions concerning setup or service of this product should be addressed to your local GSE distributor.

Information in this *Technical Reference Manual* reflects the installation, setup and operation of the Model 60 Series Instrumentation manufactured at the time this manual was released.

Your GSE distributor is:

GSE Locations Worldwide

GSE Scale Systems
22705 Heslip Drive
Novi, MI 48375
U.S.A.
Phone:(800) 755-7875
www.gse-inc.com

GSE Canada, Inc.
617 East Lake Road
Airdrie, Alberta Canada T4B 2B8
Phone:(403) 948-9921
Fax: (403) 948-1449

GSE Advanced Industrial Technologies GmbH
Hausinger Strasse 4
Langenfeld, Germany 40764
Phone: (49) 21 739 16399-0
Fax: (49) 21 739 16399-9

GSE 60 SERIES

Technical Reference Manual

Version 2.0

TABLE OF CONTENTS

Chapter 1 INTRODUCTION

This chapter contains information on the conventions used within this manual.

OVERVIEW

INTRODUCTION

The 60 Series Technical Reference Manual contains installation, configuration, and calibration information for all GSE 60 Series Instrumentation. Newer firmware is usually backward compatible with prior revisions. Refer to previous documentation for units with older firmware.

ABOUT THIS MANUAL

This manual is divided into chapters that focus on specific topics:

Chapters 2, 3 and 4 contain installation, configuration, and calibration information.

Chapter 5 describes the weigh mode operation.

Chapters 6 through 12 cover advanced configuration information.

Chapter 13 describes the installation and configuration of optional components.

Chapter 14 contains information parameter information.

Appendixes A through F contain supplementary information such as specifications, assembly drawings, and error codes and messages.

CONVENTIONS

Throughout this manual the term "*60 Series Instrumentation*" makes a collective reference to all models for the 660 Series, 560 Series and 460 Series family of instrumentation. The term "*indicator*" also refers to all 60 Series instrumentation when used without reference to a specific model.

Where differences arise between product families, the term "*660 Series Controllers*", "*560 Series Controllers*" and "*460 Series Indicators*" will be used to identify the respective family.

Reference to a specific model within a family uses that model's number. For example, *661* refers specifically to the Model 661.

SYMBOLS

WARNING!	Identifies an electrical shock hazard.
CAUTION!	Indicates important considerations.
REMINDER	Emphasizes a statement.
INFORMATION	Provides additional information.

OPERATOR INTERFACE

Procedures often refer to information that appears on the 60 Series display as well as the keystrokes required to perform a certain function.

Display Information

Display information is shown in examples exactly as it would appear on the 7-segment VFD or 4X20 VFD:

```
SetUP   Keyin
        Code:
```

```
SetUP    Keyin Code:
¾¾¾¾¾¾¾¾¾¾¾¾¾¾¾¾¾¾¾¾
¾¾¾¾¾¾¾¾¾¾¾¾¾¾¾¾¾¾¾¾
¾¾¾¾¾¾¾¾¾¾¾¾¾¾¾¾¾¾¾¾
```

Display information may also appear within chapter text, such as the message "Need Code" or "Mot'n Delay".

Keystrokes

Keystrokes are listed in examples exactly as they should be keyed in using the front panel keypad:

[1] [0] [0] [SELECT] [2] [3] [6] [4] [0] [ID] [ENTER YES]

Multiple keys that need to be pressed simultaneously are designated as:

[CLR NO] + [SELECT]

Keystrokes listed within chapter text may also appear in bold type:

100 [SELECT] 23640 [ID] [ENTER]
- or -

[CLR] + [SELECT]

EXAMPLES

Examples combine sequential keystroke and display information:

Example:
Access Setup Mode

[1] [0] [0] [SELECT]

SetUP	Keyin Code:

[2] [3] [6] [4] [0] [ID] [ENTER YES]

SetUP	Allow Mods!

P10*01	Scale 1

Chapter 2 INSTALLATION

This chapter provides instructions for installation of the 60 Series instruments and load cells connections.

OVERVIEW

INSTALLATION

This chapter provides instructions for installing the 60 Series instruments and connecting them to load cells. Please review the step-by-step procedures for your particular controller model before you install it.

ENVIRONMENTAL SUITABILITY

See Appendix A for controller dimensions and drawings.

All 60 Series instrumentation share the following features:

- The keypads are made of silicon rubber. Specifications that describe the reactions of this material to various solvents are available from GSE.
- The display window is made from a polycarbonate material. The gasket for the display is a closed cell polyethylene foam with a rubber based pressure sensitive adhesive. Although resistant to water, detergent, and alcohol, this gasket may be adversely affected by aliphatic and aromatic hydrocarbons.
- The rear panel gasket on stainless steel indicators is made from an FDA approved Silicone.
- 60 Series instruments are supplied in a sealed enclosure. Stainless steel models may be used in a washdown environment. Care must be taken to ensure that the AC power socket outlet is properly protected.

For the 460 Series, 560 Series and models 660, 661, and 662, GSE offers a Splash Guard transparent vinyl cover for added protection. For models 465, 560, 562, 660, 661, 662, and 665, a dura-shield is also available to protect the lens and keypad. Contact your local GSE distributor for more information.

460 AND 465 INDICATORS

Model 460 and 465 indicators are offered in standard and panel mount versions. All general setup, calibration, and custom programming operations of the panel-mount versions are identical to those of the respective standard versions. Differences between the standard and panel mount versions are mainly in the enclosure and the position of both the main printed circuit board and display.

Standard 460 and 465 controllers have a swivel bracket for table-top use.

Panel-mount versions of the 460 and 465 controllers are designed for permanent mounting in washdown environments and do not have a swivel bracket. The main printed circuit board is orientated so that the component side faces away from the display and the non-component side faces the same direction as the keypad, allowing easy servicing. In addition, a rear cover protects the main board and components against physical and electrical damage.

STANDARD 460 AND 465 INDICATORS

The standard 460 and 465 indicators can be installed for table-top use or permanent mounting.

Any operation that involves opening the controller enclosure should be performed only by qualified service personnel and only after the power is completely disconnected from the power source. Hazardous voltage is accessible within the enclosure!

TABLE-TOP USE

For tabletop use, position the indicator for convenient viewing. A swivel bracket with non-slip rubber feet enables tilting of the controller face to any viewing angle without the controller sliding when you press the keys.

PERMANENT MOUNTING

The swivel bracket has four holes for mounting to a fixed surface. The holes are 0.7 mm (0.28 in) diameter to accommodate M6 metric or 1/4 in fasteners. For the mounting hole pattern and overall dimensions, see Appendix A.

To remove the rear panel:

1. Remove AC power by unplugging the indicator from the power outlet socket.
2. Disconnect any cables from their source connectors.
3. Using a #2 cross-tip (Phillips) screwdriver or an 8mm socket, remove the eight 8-mm hex head screws from the rear panel (see Figure 2-1).

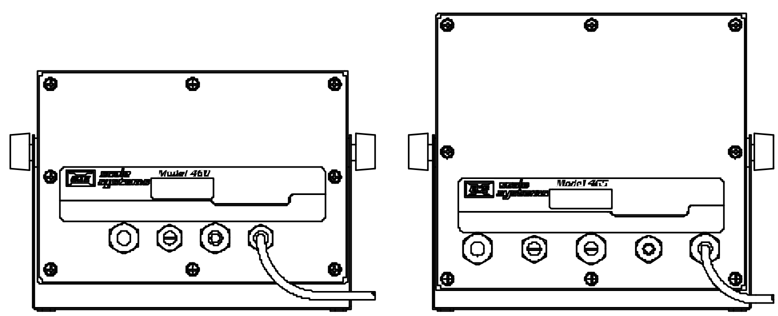


Figure 2-1: 460 and 465 Rear Panel

4. Carefully lift the rear panel from the enclosure, and disconnect the keypad ribbon cable from J8 on the main printed circuit board (see Figure 2-2).

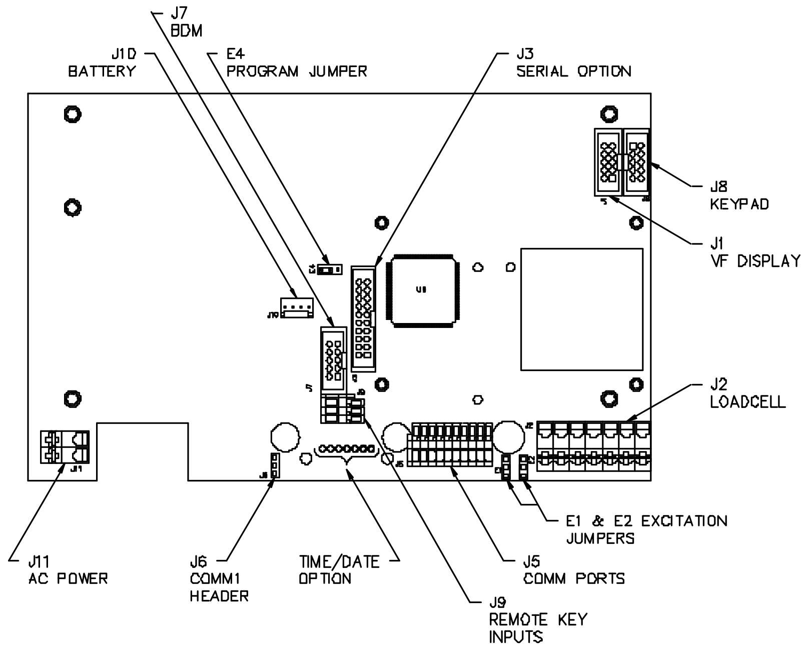


Figure 2-2: 460 and 465 Main Board

5. Disconnect the display cable from J1 on the main printed circuit board.
6. Connect the load cell leads to J2 on the main board (see Figure 2-2 and refer to page 2-19).

To re-install the rear panel:

1. Connect the keypad ribbon cable to J8 on the main board (see Figure 2-2).
2. Connect the display cable to J1 on the main printed circuit board.
3. Move the rear panel into position.
4. Tighten the strain relief to ensure a firm grip on each cable.
5. Mount the rear panel to the indicator with the eight 8-mm hex head screws.
6. Using a #2 cross tip (Phillips) screwdriver, tighten the screws on the rear panel. First tighten all the screws hand tight in order to provide even pressure on the gasket, then tighten each screw until it begins to visibly compress the gasket. This ensures the best environmental seal and maximum electromagnetic interference (EMI), radio-frequency (RFI), and electrostatic discharge (ESD) shielding performance. Over tightening the screws can deform the rear panel resulting in a poor seal.
7. Reconnect any cables to the respective source connectors.
8. Plug the indicator to the AC power outlet socket.
9. Test the indicator for proper operation.

PANEL MOUNT 460 AND 465 INDICATORS

Panel mount versions of the 460 and 465 indicators are designed for permanent mounting in washdown environments.

PANEL MOUNT VERSION REAR COVER

Any operation that involves opening the controller enclosure should be performed only by qualified service personnel and only after the power is completely disconnected from the power source! Hazardous voltage is accessible within the enclosure.

Panel mount versions of the 460 and 465 indicators offer a rear cover to protect the main board against physical and electrical damage. The rear cover enables easy access to the main board. There is no need to remove the entire panel mount unit for servicing.

The rear cover snaps onto the board mounting plate. A single M4x0.7 hex bolt is included for installing a legal-for-trade seal and for fastening the rear cover.

Two slots are positioned at the bottom of the cover to enable power, load cell, printer, computer, and other wire routing. All wire strain relieves and terminals should be positioned below or near this area.

The serial number, power requirements, and approvals label is on the cover.

Refer to Appendix A for dimensional drawings.

INSTALLATION

The procedure for installing a 460 and 465 panel mount indicator is identical with the exception of the panel cut-out size (see Figure 2-3 and Figure 2-4).

Panel Cut-Out Dimensions (460)

Figure 2-3 shows the cut-out dimensions for the 460 panel mount indicator.

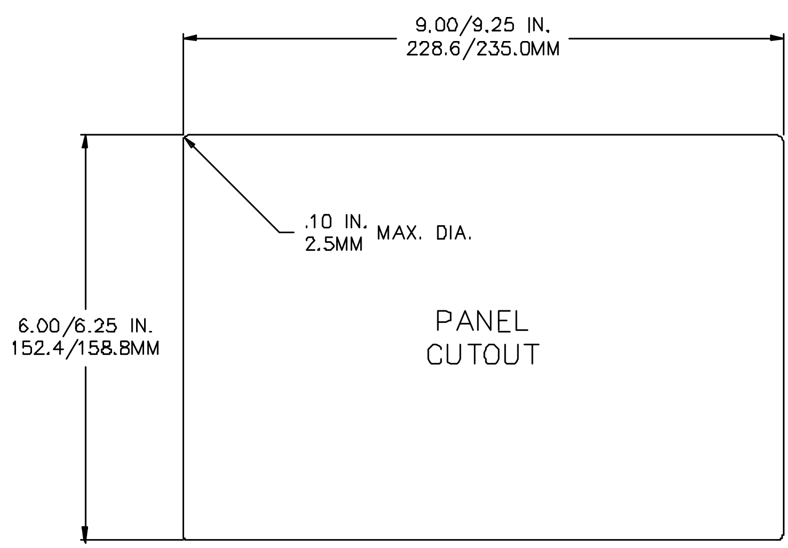


Figure 2-3: 460 Panel Mount Cut-Out Dimensions

Panel Cut-Out Dimensions (465)

Figure 2-4 shows the cut-out dimensions for the 465 panel mount indicator.

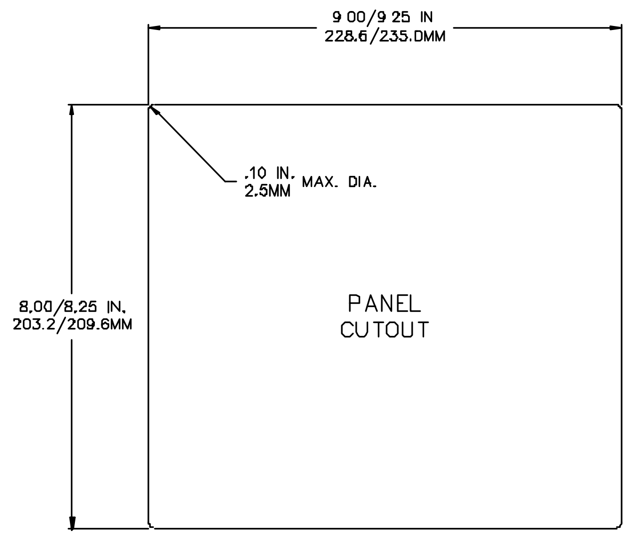


Figure 2-4: 465 Panel Mount Cut-Out Dimensions

To install panel-mount 460 and 465 indicators:

1. Make a cut-out in the panel using the dimensions specified in Figure 2-3 for a 460 indicator or Figure 2-4 an for a 465 indicator.

   The corners are typically 0.1R maximum.
2. Remove the 8 hex nuts holding the main board front panel to the back bracket.
3. Make sure the gasket remains on the front panel side. When installed, the gasket will be compressed against the front of the enclosure cutout.
4. Position the main board front panel inside the cut-out making sure the keypad is facing in the correct upright position.
5. From the inside of the enclosure fit the back bracket over the studs on the main board front panel.
6. Install the 8 hex nuts from the inside of the panel.
7. Tighten nuts until they are hand tight and pressing against the back panel surface. Then tighten each nut by 3/4 turn to compress the gasket to the front of the panel.

To open the panel-mount 460 and 465 indicators:

1. Remove AC power by unplugging the indicator from the power outlet socket.
2. Disconnect any additional cables from the respective source connectors.

3. Remove the M4 hex bolt, if installed. Hook a finger under the edge of the cover. Pull outward and upward to disengage the cover from the mounting plate tabs.
4. Install the load cell leads into J2 on the main board (see Figure 2-2 and refer to page 2-19).

To re-install the rear panel:

1. Move the rear cover into position. Ensure that all of the cables are positioned within the cutouts of the cover. Press the cover straight towards the panel until the left and right side slots snap over the tabs on the mounting plate.
2. Mount the rear cover, if used, to the indicator with the four 8-mm hex head screws.
3. Install the M4 hex bolt in the upper left corner using a 7mm socket.
4. Reconnect any cables from the respective source connectors.
5. Plug the indicator into the AC power outlet socket.
6. Test the indicator for proper operation.

560 AND 562 CONTROLLERS

Model 560 and 562 controllers are offered in standard and panel mount versions. All general setup, calibration, and custom programming operations of the panel-mount versions are identical to those of the respective standard versions. Differences between the standard and panel mount versions are mainly in the enclosure and the position of both the main printed circuit board and display.

Standard 560 and 562 controllers have a swivel bracket for table-top use.

Panel-mount versions of the 560 and 562 controllers are designed for permanent mounting in washdown environments and do not have a swivel bracket. The main printed circuit board is orientated so that the component side faces away from the display and the non-component side faces the same direction as the keypad, allowing easy servicing. In addition, a rear cover protects the main board and components against physical and electrical damage.

STANDARD 560 AND 562 CONTROLLERS

The standard 560 and 562 controllers can be installed for table-top use or permanent mounting.

TABLE-TOP USE

For tabletop use, position the controller for convenient viewing. A swivel bracket with non-slip rubber feet enables tilting of the controller face to any viewing angle without the controller sliding when you press the keys.

PERMANENT MOUNTING

The swivel bracket has four holes for mounting to a fixed surface. The holes are 0.7 mm (0.28 in) diameter to accommodate M6 metric or 1/4 in fasteners.

Refer to Appendix A for dimensional drawings.

Any operation that involves opening the controller enclosure should be performed only by qualified service personnel and only after the power is completely disconnected from the power source. Hazardous voltage is accessible within the enclosure!

To remove the rear panel:

1. Remove AC power by unplugging the controller from the power outlet socket.
2. Disconnect any cables from their source connectors.
3. Using a #2 cross-tip (Phillips) screwdriver or an 8mm socket, remove the eight 8-mm hex head screws from the rear panel (see Figure 2-5).

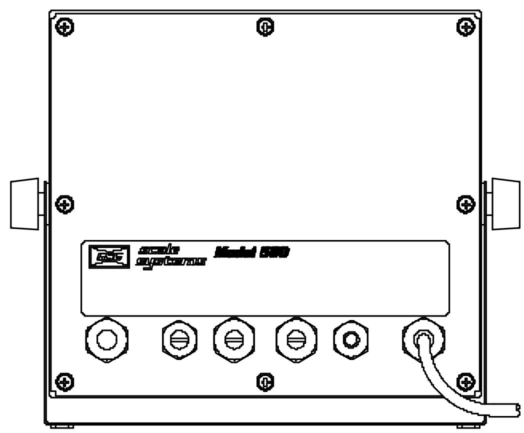


Figure 2-5: 560 and 562 Rear Panel

4. Carefully lift the rear panel from the enclosure, and disconnect the keypad ribbon cable from J5 on the main printed circuit board (see Figure 2-6).

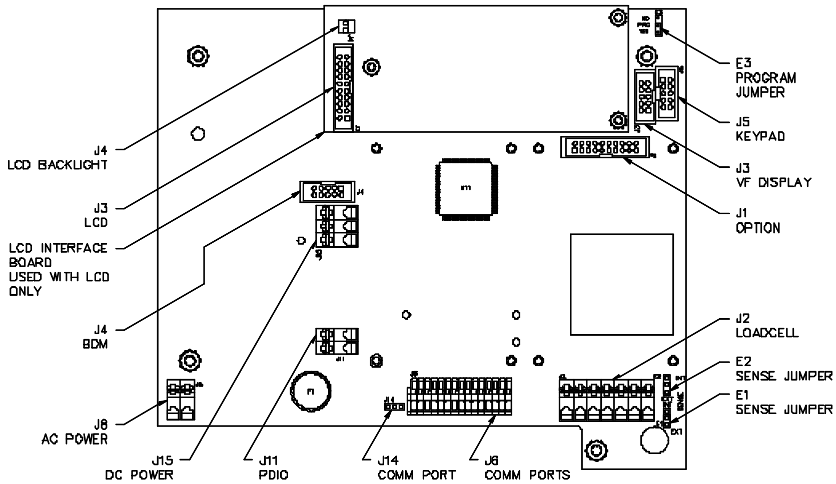


Figure 2-6: 560 Series Main Board

5. For the Model 560, disconnect the display cable from J3 on the main printed circuit board.

   - or -

   For the Model 562, disconnect the display ribbon cable from J3 on the LCD driver board #420921-36617. Disconnect the backlight cable from J4 on the LCD driver board.

6. Connect the load cell leads to J2 on the main board (see and refer to page 2-19).

To re-install the rear panel:

1. Connect the keypad ribbon cable to J5 on the main board (see Figure 2-6).
2. For the Model 560, connect the display cable to J3 on the main printed circuit board.

   or -

   For the Model 562, connect the display ribbon cable to J3 on the LCD driver board #420921-36617. Connect the backlight cable to J4 on the LCD driver board.

3. Move the rear panel into position.
4. Tighten the strain relief to ensure a firm grip on each cable.
5. Mount the rear panel to the controller with the eight 8-mm hex head screws.
6. Using a #2 cross tip (Phillips) screwdriver, tighten the screws on the rear panel. First tighten all the screws hand tight in order to provide even pressure on the gasket, then tighten each screw until it begins to visibly compress the gasket. This ensures the best environmental seal and maximum electromagnetic interference (EMI), radio-frequency (RFI), and electrostatic discharge (ESD) shielding performance. Over

tightening the screws can deform the rear panel resulting in a poor seal.

7. Reconnect any cables to the respective source connectors.
8. Plug the controller to the AC power outlet socket.
9. Test the controller for proper operation.

PANEL MOUNT 560 AND 562 CONTROLLERS

Panel mount versions of the 560 and 562 controllers are designed for permanent mounting in washdown environments.

PANEL MOUNT VERSION REAR COVER

Any operation that involves opening the controller enclosure should be performed only by qualified service personnel and only after the power is completely disconnected from the power source! Hazardous voltage is accessible within the enclosure.

Panel mount versions of the 560 and 562 controllers offer a rear cover to protect the main board against physical and electrical damage. The rear cover enables easy access to the main board. There is no need to remove the entire panel mount unit for servicing.

The rear cover snaps onto the board mounting plate. A single M4x0.7 hex bolt is included for NTEP sealing and for fastening the rear cover.

Two slots are positioned at the bottom of the cover to enable power, load cell, printer, computer, and other wire routing. All wire strain relieves and terminals should be positioned below or near this area.

The serial number, power requirements, and approvals label is on the cover.

Refer to Appendix A for dimensional drawings.

INSTALLATION

The procedure for installing a 560 and 562 panel mount controller is identical.

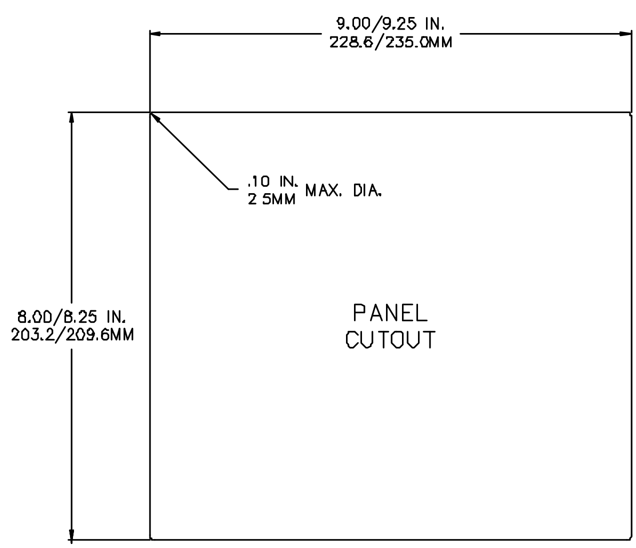


Figure 2-7: 560 and 562 Panel Mount Cut-Out

To install panel-mount 560 and 562 controllers:

1. Make a cut out in the panel using the dimensions shown in Figure 2-7.

   The corners are typically 0.1R maximum.
2. Remove the 8 hex nuts holding the main board front panel to the back bracket.
3. Make sure the gasket remains on the front panel side. When installed, the gasket will be compressed against the front of the enclosure cutout.
4. Position the main board front panel inside the cutout making sure the keypad is facing in the correct upright position.
5. From the inside of the enclosure fit the back bracket over the studs on the main board front panel.
6. Install the 8 hex nuts from the inside of the panel.
7. Tighten nuts until they are hand tight and pressing against the back panel surface. Then tighten each nut by 3/4 turn to compress the gasket to the front of the panel.

To open the panel-mount 560 and 562 controllers:

1. Remove AC power by unplugging the controller from the power outlet socket.
2. Disconnect any additional cables from the respective source connectors.
3. Remove the M4 hex bolt, if installed. Hook a finger under the edge of the cover. Pull outward and upward to disengage the cover from the mounting plate tabs.
4. Install the load cell leads into J2 on the main board (see page 2-19).

To re-install the rear panel:

1. Move the rear cover into position. Ensure that all of the cables are positioned within the cutouts of the cover. Press the cover straight towards the panel until the left and right side slots snap over the tabs on the mounting plate.
2. Mount the rear cover, if used, to the controller with the four 8-mm hex head screws.
3. Install the M4 hex bolt in the upper left corner using a 7mm socket.
4. Reconnect any cables from the respective source connectors.
5. Plug the controller into the AC power outlet socket.
6. Test the controller for proper operation.

660, 661, AND 662 CONTROLLERS

Model 660, 661 and 662 controllers are offered in standard and panel mount versions. All general setup, calibration, and custom programming operations of the panel-mount versions are identical to those of the respective standard versions. Differences between the standard and panel mount versions are mainly in the enclosure and the position of both the main printed circuit board and display.

Standard 660, 661 and 662 controllers have a swivel bracket for table-top use.

Panel-mount versions of the 660, 661, and 662 controllers are designed for permanent mounting in washdown environments and do not have a swivel bracket. The main printed circuit board is orientated so that the component side faces away from the display and the non-component side faces the same direction as the keypad, allowing easy servicing. In addition, a rear cover protects the main board and components against physical and electrical damage.

STANDARD 660, 661 AND 662 CONTROLLERS

The standard 660, 661 and 662 controllers can be installed for table-top use or permanent mounting.

TABLE-TOP USE

For tabletop use, position the controller for convenient viewing. A swivel bracket with non-slip rubber feet enables tilting of the controller face to any viewing angle without the controller sliding when you press the keys.

PERMANENT MOUNTING

The swivel bracket has four holes for mounting to a fixed surface. The holes are 0.7 mm (0.28 in) diameter to accommodate M6 metric or 1/4 in fasteners.

Refer to Appendix A for dimensional drawings.

⚠ **Any operation that involves opening the controller enclosure should be performed only by qualified service personnel and only after the power is completely disconnected from the power source. Hazardous voltage is accessible within the enclosure!**

To remove the rear panel:

1. Remove AC power by unplugging the controller from the power outlet socket.
2. Disconnect any cables from their source connectors.
3. Using a #2 cross-tip (Phillips) screwdriver or an 8mm socket, remove the eight 8-mm hex head screws from the rear panel (see Figure 2-8).

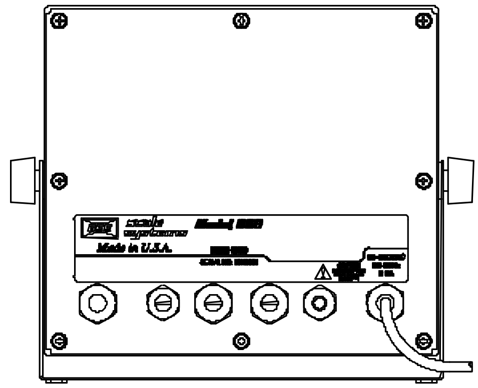


Figure 2-8: 660, 661 and 662 Rear Panel

4. Carefully lift the rear panel from the enclosure, and disconnect the keypad ribbon cable from J5 on the main printed circuit board (see Figure 2-9).

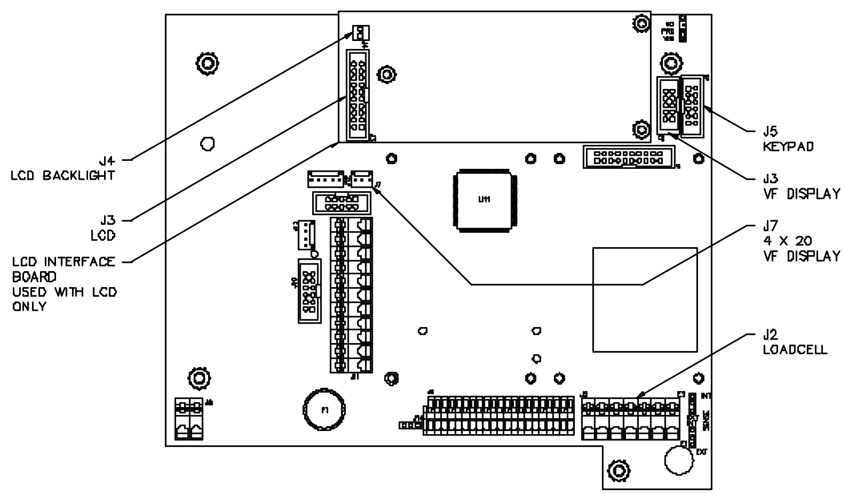


Figure 2-9: 660 Series Main Board

5. For the Model 660, disconnect the display cable from J3 on the main printed circuit board.

   - or -

   For the Model 661, disconnect the three wire display cable from J7 on the main printed circuit board.

   - or -

   For the Model 662, disconnect the display ribbon cable from J3 on the LCD driver board #420921-36617. Disconnect the backlight cable from J4 on the LCD driver board.

6. Connect the load cell leads to J2 on the main board (see Figure 2-9 and refer to page 2-19).

To re-install the rear panel:

1. Connect the keypad ribbon cable to J5 on the main board (see Figure 2-9).
2. For the Model 660, connect the display cable to J3 on the main printed circuit board.

   or -

   For the Model 661, connect the three wire display cable to J7 on the main printed circuit board.

   or -

   For the Model 662, connect the display ribbon cable to J3 on the LCD driver board #420921-36617. Connect the backlight cable to J4 on the LCD driver board.
3. Move the rear panel into position.
4. Tighten the strain relief to ensure a firm grip on each cable.
5. Mount the rear panel to the controller with the eight 8-mm hex head screws.
6. Using a #2 cross tip (Phillips) screwdriver, tighten the screws on the rear panel. First tighten all the screws hand tight in order to provide even pressure on the gasket, then tighten each screw until it begins to visibly compress the gasket. This ensures the best environmental seal and maximum electromagnetic interference (EMI), radio-frequency (RFI), and electrostatic discharge (ESD) shielding performance. Over tightening the screws can deform the rear panel resulting in a poor seal.
7. Reconnect any cables to the respective source connectors.
8. Plug the controller to the AC power outlet socket.
9. Test the controller for proper operation.

PANEL MOUNT 660, 661 AND 662 CONTROLLERS

Panel mount versions of the 660, 661, and 662 controllers are designed for permanent mounting in washdown environments.

PANEL MOUNT VERSION REAR COVER

⚠ **Any operation that involves opening the controller enclosure should be performed only by qualified service personnel and only after the power is completely disconnected from the power source! Hazardous voltage is accessible within the enclosure.**

Panel mount versions of the 660, 661, and 662 controllers offer a rear cover to protect the main board against physical and electrical damage. The rear cover enables easy access to the main board. There is no need to remove the entire panel mount unit for servicing.

The rear cover snaps onto the board mounting plate. A single M4x0.7 hex bolt is included for NTEP sealing and for fastening the rear cover.

Two slots are positioned at the bottom of the cover to enable power, load cell, printer, computer, and other wire routing. All wire strain relieves and terminals should be positioned below or near this area.

The serial number, power requirements, and approvals label is on the cover.

Refer to Appendix A for dimensional drawings.

INSTALLATION

The procedure for installing a 660, 661 and 662 panel mount controller is identical.

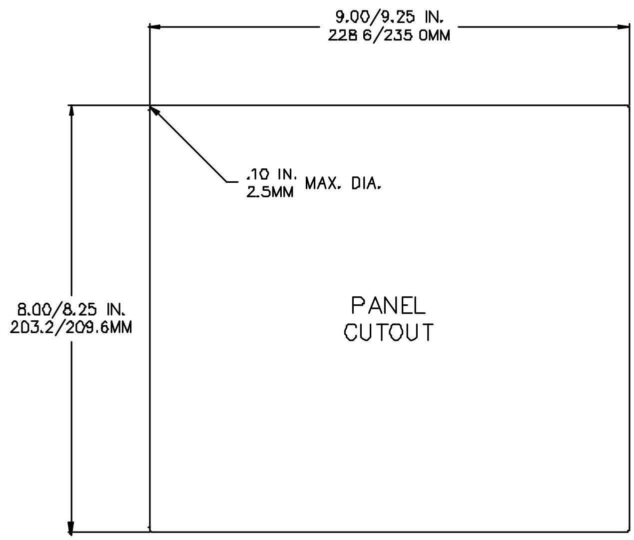


Figure 2-10: 660, 661 and 662 Panel Mount Cut-Out

To install panel-mount 660, 661 and 662 controllers:

1. Make a cut out in the panel using the dimensions shown in Figure 2-10.

   The corners are typically 0.1R maximum.

2. Remove the 8 hex nuts holding the main board front panel to the back bracket.
3. Make sure the gasket remains on the front panel side. When installed, the gasket will be compressed against the front of the enclosure cutout.
4. Position the main board front panel inside the cutout making sure the keypad is facing in the correct upright position.
5. From the inside of the enclosure fit the back bracket over the studs on the main board front panel.
6. Install the 8 hex nuts from the inside of the panel.
7. Tighten nuts until they are hand tight and pressing against the back panel surface. Then tighten each nut by 3/4 turn to compress the gasket to the front of the panel.

To open the panel-mount 660, 661, and 662 controllers:

1. Remove AC power by unplugging the controller from the power outlet socket.
2. Disconnect any additional cables from the respective source connectors.
3. Remove the M4 hex bolt, if installed. Hook a finger under the edge of the cover. Pull outward and upward to disengage the cover from the mounting plate tabs.
4. Install the load cell leads into J2 on the main board (see page 2-19).

To re-install the rear panel:

1. Move the rear cover into position. Ensure that all of the cables are positioned within the cutouts of the cover. Press the cover straight towards the panel until the left and right side slots snap over the tabs on the mounting plate.
2. Mount the rear cover, if used, to the controller with the four 8-mm hex head screws.
3. Install the M4 hex bolt in the upper left corner using a 7mm socket.
4. Reconnect any cables from the respective source connectors.
5. Plug the controller into the AC power outlet socket.
6. Test the controller for proper operation.

663 CONTROLLER

The Model 663 controller is designed for permanent installation as a control panel. It has four mounting flanges. The holes and slots are 0.44 in (11.2 mm) diameter to accommodate 7/16-in diameter or M8 metric fasteners.

Refer to Appendix A for dimensional drawings.

⚠ **Any operation that involves opening the controller enclosure should be performed only by qualified service personnel and only after the power is completely disconnected from the power source! Hazardous voltage is accessible within the enclosure.**

To open the 663 Controller:

1. Turn off the controller by disconnecting it from the AC power supply.
2. Using a large, flat-tip screwdriver, loosen the spring-loaded 1/4-20 slotted pan head screws in the two fastening clamps on the right side of the controller, and slide the clamps to the right (see Figure 2-11).
3. Open the controller door.
4. Remove the M4 hex bolt, if installed. Hook a finger under the edge of the cover. Pull outward and upward to disengage the cover from the mounting plate tabs.
5. Locate the main board near the middle right of the controller door interior (see Figure 2-12).

6. Install the load cell leads into J2 on the main board (see Figure 2-9 and refer to page 2-19).

To close the 663 Controller:

1. Move the rear cover into position. Ensure that all of the cables are positioned within the cutouts of the cover. Press the cover straight towards the panel until the left and right side slots snap over the tabs on the mounting plate.
2. Mount the rear cover, if used, to the controller with the four 8-mm hex head screws.
3. Install the M4 hex bolt in the upper left corner using a 7mm socket.
4. Reconnect any cables from the respective source connectors.
5. Close the controller enclosure door.
6. Slide the exterior door fastening clamps to the left (see Figure 2-11).
7. Using a large, flat-tip screwdriver, tighten the 1/4-20 slotted pan head screws on the fastening clamps.
8. Connect the controller to an AC power supply to power it on.

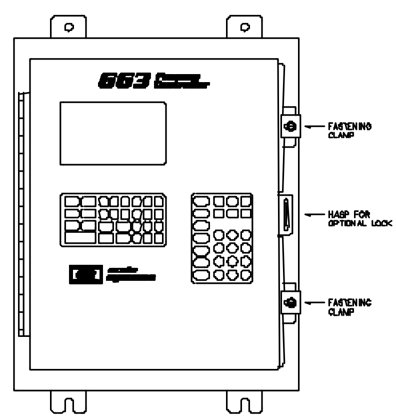


Figure 2-11: 663 Front Panel

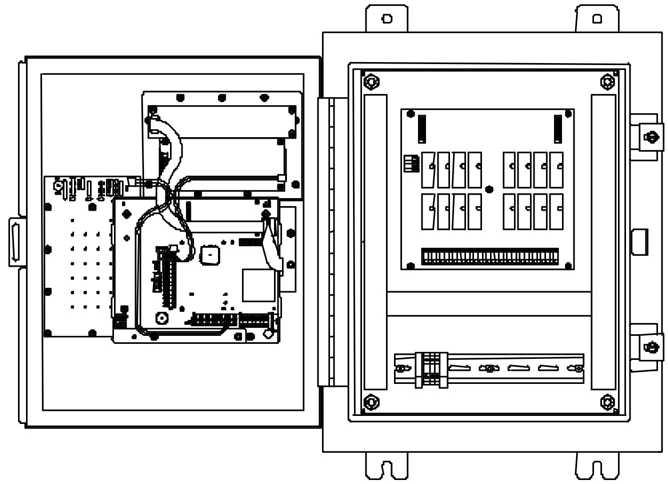


Figure 2-12: 663 Internal View

665 CONTROLLER

The 665 controller is offered in standard and panel mount version. All general setup, calibration, and custom programming operations of the panel-mount versions are identical to those of the standard version. Differences between the standard and panel mount versions are mainly in the enclosure and the position of both the main printed circuit board and display.

The standard 665 controller has a swivel bracket for table-top use.

The panel-mount version of the 665 controller is designed for permanent mounting in washdown environments and does not have a swivel bracket. The main printed circuit board is orientated so that the component side faces away from the display and the non-component side faces the same direction as the keypad, allowing easy servicing. In addition, a rear cover protects the main board and components against physical and electrical damage.

STANDARD 665 CONTROLLER

The standard 665 controller can be installed for table-top use or permanent mounting.

TABLE-TOP USE

For tabletop use, position the 665 controller for convenient viewing. The 665 has a swivel bracket that enables tilting of the controller face to any angle for viewing. The bracket also has non-slip rubber feet to prevent sliding when you press the keys.

PERMANENT MOUNTING

⚠ Any operation that involves opening the controller enclosure should be performed only by qualified service personnel and only after the power is completely disconnected from the power source! Hazardous voltage is accessible within the enclosure.

The swivel bracket on the 665 controller has four holes for mounting to a fixed surface. The holes are 0.7 mm (0.28 in) diameter to accommodate M6 metric or 1/4 in fasteners.

Refer to Appendix A for dimensional drawings.

To remove the rear panel:

1. Remove AC power by unplugging the controller from the power outlet socket.
2. Disconnect any additional cables from their source connectors.
3. Using a #2 cross-tip (Phillips) screwdriver or an 8mm socket, remove the 10 8-mm hex head Phillips screws from the rear panel. See Figure 2-13 for a view of the rear panel.

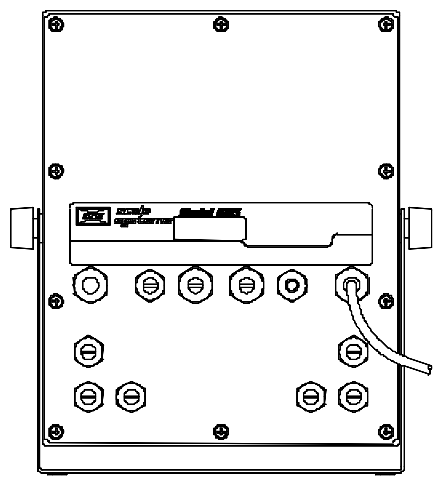


Figure 2-13: 665 Rear Panel

4. Disconnect the keypad ribbon cable from J5 on the main printed circuit board. See Figure 2-9.
5. For the 665 VF display model, disconnect the six digit display ribbon cable from J3 on the main printed circuit board. Disconnect the 4x20 cable from J7 on the main printed circuit board.

   For the 665 LCD display model, disconnect the display ribbon cable from J3 on the LCD driver board #420921-36617. Disconnect the backlight cable from J4 on the LCD driver board.
6. Connect the load cell leads to J2 on the main board (see Figure 2-9 and refer to page 2-19).

To re-install the rear panel:

1. Reconnect the keypad ribbon cable to J5 on the main board.
2. Move the rear panel into position.
3. Tighten the strain relief to ensure a firm grip on each cable.
4. Mount the rear panel to the controller with the 10 8-mm hex head Philips screws.
5. Using a #2 cross tip (Phillips) screwdriver, tighten the screws on the rear panel. First tighten all the screws hand tight in order to provide even pressure on the gasket, then tighten each screw until it begins to visibly compress the gasket. This ensures the best environmental seal and maximum electromagnetic interference (EMI), radio-frequency (RFI), and electrostatic discharge (ESD) shielding performance. Over tightening the screws can deform the rear panel.
6. Reconnect any cables to the respective source connectors.
7. Plug the controller to the AC power outlet socket.
8. Test the controller for proper operation.

PANEL MOUNT 665 CONTROLLER

The panel mount version of the 665 controller is designed for permanent mounting in washdown environments.

PANEL MOUNT VERSION REAR COVER

Any operation that involves opening the controller enclosure should be performed only by qualified service personnel and only after the power is completely disconnected from the power source! Hazardous voltage is accessible within the enclosure.

The panel mount 665 controller offers a rear cover to protect the main board against physical and electrical damage. The rear cover enables easy access to the main board. There is no need to remove the entire panel mount unit for servicing.

The rear cover snaps onto the board mounting plate. A single M4x0.7 hex bolt is included for NTEP sealing and for fastening the rear cover.

Two slots are positioned at the bottom of the cover to enable power, load cell, printer, computer, and other wire routing. All wire strain relieves and terminals should be positioned below or near this area.

The serial number, power requirements, and approvals label is on the cover.

Refer to Appendix A for dimensional drawings.

INSTALLATION

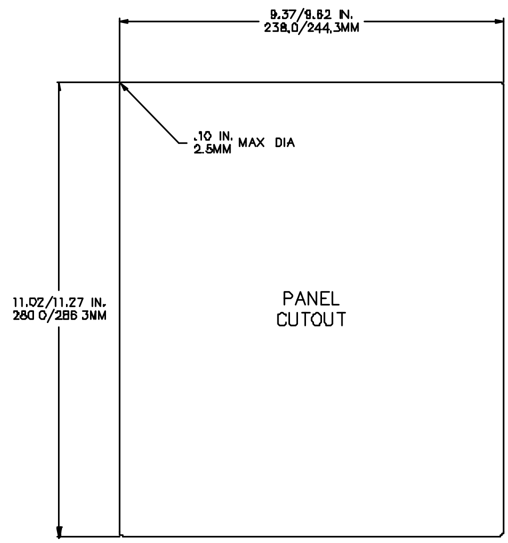


Figure 2-14: 665 Panel Mount Cut-Out Dimensions

To install panel-mount 660, 661 and 662 controllers:

1. Make a cut out in the panel using the dimensions shown in.

   The corners are typically 0.1R maximum.

2. Remove the 8 hex nuts holding the main board front panel to the back bracket.
3. Make sure the gasket remains on the front panel side. When installed, the gasket will be compressed against the front of the enclosure cutout.
4. Position the main board front panel inside the cutout making sure the keypad is facing in the correct upright position.
5. From the inside of the enclosure fit the back bracket over the studs on the main board front panel.
6. Install the 8 hex nuts from the inside of the panel.
7. Tighten nuts until they are hand tight and pressing against the back panel surface. Then tighten each nut by 3/4 turn to compress the gasket to the front of the panel.

To open the panel-mount 660, 661, and 662 controllers:

1. Remove AC power by unplugging the controller from the power outlet socket.
2. Disconnect any additional cables from the respective source connectors.

3. Remove the M4 hex bolt, if installed. Hook a finger under the edge of the cover. Pull outward and upward to disengage the cover from the mounting plate tabs.
4. Install the load cell leads into J2 on the main board (see page 2-19).

To re-install the rear panel:

1. Move the rear cover into position. Ensure that all of the cables are positioned within the cutouts of the cover. Press the cover straight towards the panel until the left and right side slots snap over the tabs on the mounting plate.
2. Mount the rear cover, if used, to the controller with the four 8-mm hex head screws.
3. Install the M4 hex bolt in the upper left corner using a 7mm socket.
4. Reconnect any cables from the respective source connectors.
5. Plug the controller into the AC power outlet socket.
6. Test the controller for proper operation.

LOAD CELL CONNECTIONS

This section provides information on connecting the load cell the 60 Series instruments.

TRANSDUCER EXCITATION

All 60 Series instruments are designed to be connected to any Wheatstone bridge design strain gage-based force measuring transducer. The instrument supplies 10 VDC developed from +5VDC and -5VDC referenced to common for the excitation voltage. The maximum excitation supply current available for the load cell (s) is

- 460 Series 350 mA; twelve (12) 350 ohm load cells
- 560 Series 400 mA; fourteen (14) 350 ohm load cells
- 660 Series 400 mA; fourteen (14) 350 ohm load cells

CABLE RECOMMENDATIONS

A high quality cable that has an overall braided shield and 16 to 24 gage (AWG) stranded wire is recommended for the connection to the weigh platform. The load cell cable should be routed into the controller enclosure through the strain relief nearest the J2 loadcell connector.

SENSE LEAD CONNECTIONS

Four-lead or six-lead scale platforms can be connected to any 60 Series instrument. Refer to Table 2-1 for wiring information.

Table 2-1: Load Cell Connections

Load Cell Function	GSE Platform Color Code
+ Excitation	Red
- Excitation	Black
+ Signal	White
- Signal	Green
+ Sense	Red (optional)
- Sense	Black (optional)

Six-lead cables include two extra wires for sensing the actual excitation voltage at the load cell. This connection compensates for variations in the resistance of the excitation wiring which can change due to variations in temperature, especially over long distances.

If the load cell has six leads, the jumpers (E1 and E2) next to the J2 connector should be set to the EXT (bottom) position.

If the load cell has four leads, the jumpers (E1 and E2) must be set to the INT (top) position. This jumpers the excitation voltage to the sense leads on the main board, providing the required excitation feedback in the absence of the sense leads.

INSTALLATION

The PC board mounting nut is used to establish the earth ground reference for the PC board circuitry, so the nut must always be in place when the controller is powered up!

For load cells with six conductors, two jumpers (E1 and E2, next to J2 on the Main PC Board) must be adjusted so external sensing can be operational.

Load cell connections are labeled on the PC board next to the J2 connector.

To connect the load cell:

1. Strip back the jacket of the weigh platform load cell cable approximately 3 cm (1.25 in) from the end of the cable.
2. Using a small screwdriver, create an opening in the braided shield, just past the end of the jacket. Pull the wires out of the braided shield.
3. Strip back the insulation of each conductor wire 0.6 cm (1/4 in).
4. Twist the strands of each conductor to aid insertion into J2. Do not tin the leads. The lever connectors are designed to provide a gas tight vibration proof connection to stranded wires. Tinning the leads decreases the connection reliability.
5. Loosen the nearest strain relief and route the load cell cable through it.
6. View the main PC board and locate the lever connector labeled J2.
7. Connect the four or six conductors to the proper terminals, as described in Table 2-1. Refer to the load cell's color wiring code for the proper colors for each connection.
8. Use a small screwdriver to apply force to the lever connector, as shown in Figure 2-15.

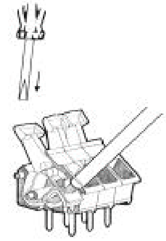


Figure 2-15: Inserting Load Cell Wires

9. Insert the lead in place, as shown in Figure 2-16 for a 460 Series indicator or Figure 2-17 for a 560 Series or 660 Series controller.
10. Test all connections by pulling lightly on each conductor.
11. Connect the load cell shield to the lever connector SHIELD.
12. Pull any excess cable back out of the controller through the strain relief until there is no slack between the shield connection and the strain relief.
13. Tighten down the strain relief securely to ensure a firm grip on the cable.

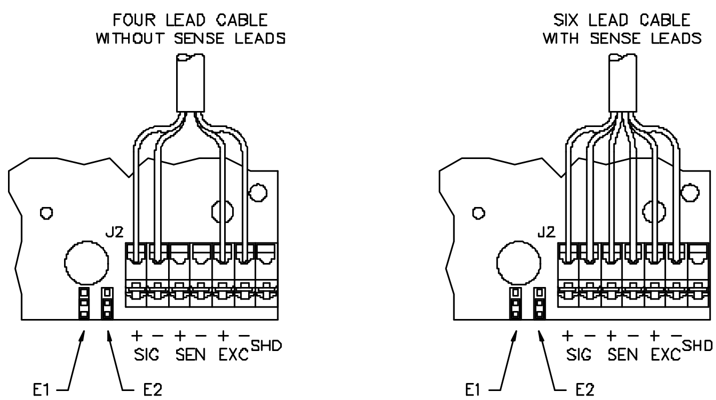


Figure 2-16: 460 Series Load Cell Connections

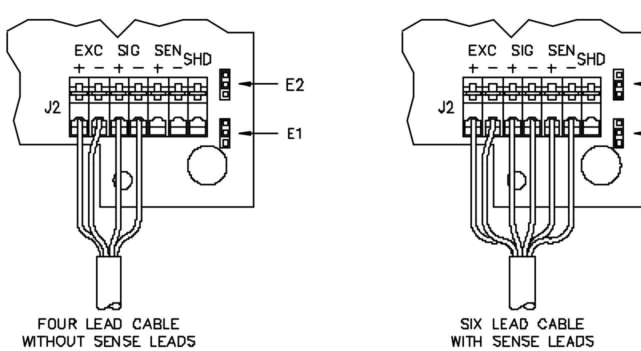


Figure 2-17: 660 Series Load Cell Connections

DC POWER CONNECTIONS

All 60 Series instruments can be powered by an external 10-32 VDC power supply.

460 SERIES

Connect an external DC power source to the 24V and GND pins on the J10 battery connector.

The mating female connector for J10 is available from GSE:

- 4-Position Female Connector P/N: 26-20-3380
- 4-Position Connector Cover P/N: 26-20-3393

⚠ **DO NOT** connect an external power source to the 'L' or 'S' pins on J10.

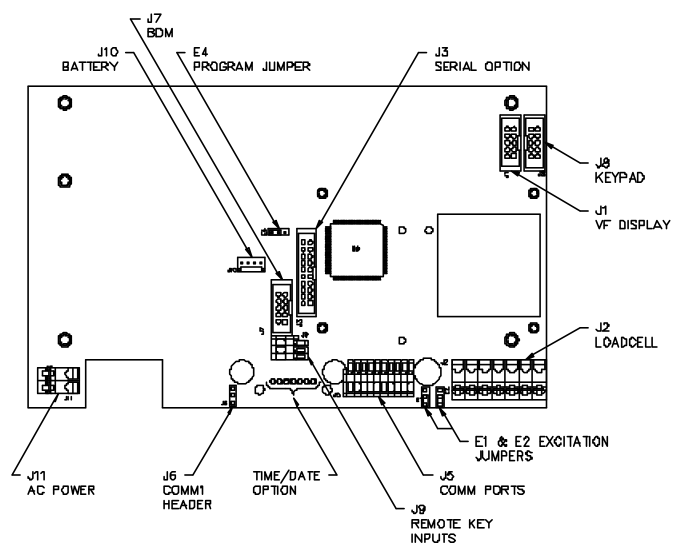


Figure 2-18: 460 Series Main Board J10 DC Power Connection

560 SERIES

Connect an external DC power source to the 24V and GND spring-lever terminals on the J15 connector.

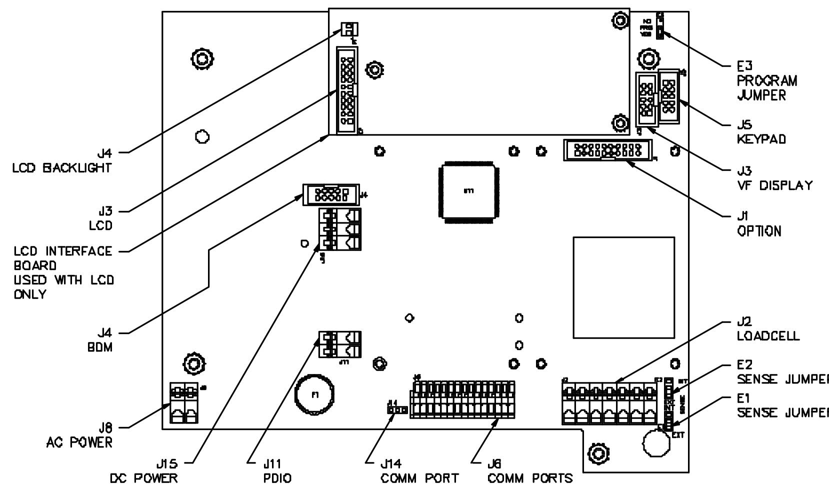


Figure 2-19: 560 Series Main Board J15 DC Power Connection

660 SERIES

Connect an external DC power source to the 24V and GND spring-lever terminals on the J11 connector.

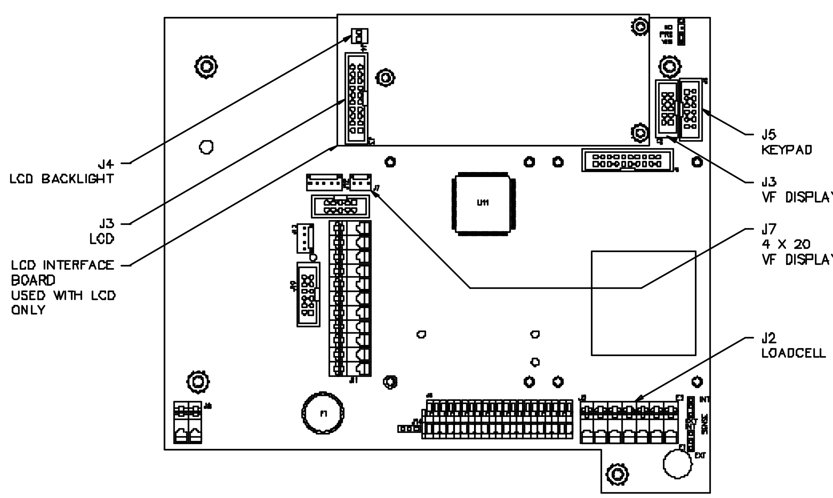


Figure 2-20: 660 Series Main Board J11 Power Connection

LCD OPERATION

The 8X40 LCD is the standard display for the 562 and 662 controllers. A 16X40 LCD is available as the primary display for the 663 and 665 controllers. Both displays interface to the same factory-installed LCD driver board as shown in Figure 2-19 and Figure 2-20.

LCD HARDWARE SETUP

The LCD driver board is installed on new indicators at the factory. Jumpers E2 and E3 are both set to match the display size (SM = 8X40 LCD; LG = 16X40 LCD) as shown in Figure 2-21.

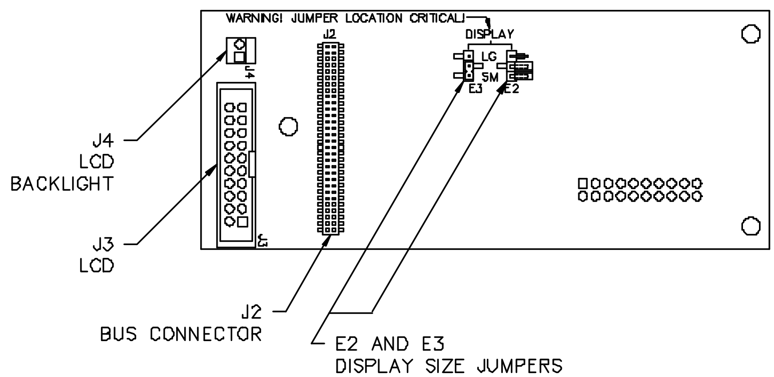


Figure 2-21: LCD Interface Board

Connections to the LCD driver board are the LCD interface ribbon cable (J3) and the LCD backlight (J4).

A low-tone, variable speed warble will occur at power-up if the LCD driver board is installed but the display is not connected to J3.

LCD PARAMETER SETUP

The LCD driver board is auto-detected at power-up. Once detected, the LCD is automatically selected as the primary display. Therefore, it is not possible to use the 7-segment VFD or 4X20 VFD with the LCD connected. Model 660 Series controllers that use the LCD will have P425 set to "LCD Alwys", indicating that the LCD display will always be used as the primary display when the driver board is installed.

USING THE 4X20 VFD WITH AN LCD

Although it is possible to connect a 4X20 VFD while using an LCD, there are several issues that complicate simultaneous use of both displays:

- It would not be possible to use macro commands to transmit data to the 4X20 VFD as they will instead be automatically routed to the LCD.

- It would not be possible to get an operator input on the 4X20 VFD using the %G or %K commands.
- Only a custom transmit directed out comm4 would allow communication to the 4X20 VFD.

LCD Contrast Adjustment

The contrast of the LCD changes with temperature. A contrast setting that allows good viewing at a high temperature might make the display impossible to read at a low temperature. This would make it difficult to access P430 to change the contrast.

If the display is not visible at power-up you can adjust the contrast as follows:

1. Power down.
2. Hold down the left, down and right arrow keys.
3. Power up.
4. Continue to hold the left, down and right arrow keys until you can see the contrast adjustment menu on the display, then release.
5. Make fine adjustments to the contrast by pressing the up and down arrow keys.
6. Press **[ENTER]** to exit the menu and permanently store the new contrast setting.

Keypad Configuration

All 60 Series indicators are shipped with the keypad properly configured. However proper keypad operation can not always be guaranteed when installing a replacement main board due to the fact that some indicators support multiple keypad styles. A keypad will not operate correctly when it does not match the configuration of the main board. And since the keypad does not operate correctly, it is impossible to use the conventional method to access the setup mode and change P450 to the correct style keypad. Instead, holding down various keys at power-up as described below will automatically configure P450 to reflect the connected keypad.

460 Keypad

If the keypad does not operate correctly at power-up, configure the keypad as follows:

1. Power down.
2. Hold down the **[ENTER]** key.
3. Power up.
4. Continue holding the key until "M460 Keypd" is displayed, then release.

465 KEYPAD

If the keypad does not operate correctly at power-up, configure the keypad as follows:

1. Power down.
2. Hold down the **[SCALE SELECT] + [F1] + [SELECT]** keys.*
3. Power up.
4. Continue holding the keys until "M460 Keypd" is displayed, then release.

* *Use these keys for firmware dated after September 28, 2001. Previously the* ***[ZERO] + [SCALE SELECT] + [F1]*** *keys were used to configure the keypad at power-up.*

560 SERIES KEYPAD

The 560 Series controllers offers only one 23-key keypad style and therefore should not require reconfiguration. However, should the keypad not operate correctly, follow the same procedure as described above for the 465 to display "M560 Keypd" at power-up.

660 SERIES KEYPAD

The 660 Series controllers offers only one 28-key keypad style. However, the 25-key keypad used by the 650 Series is also compatible and would require reconfiguration if connected to a 660 Series main board.

If the keypad does not operate correctly at power-up, configure the keypad as follows:

1. Power down.
2. Hold down the **[ID] + [F1] + [F2]** keys.*
3. Power up.
4. Continue holding the keys until either "M660 Keypd" or "M653 Keypd" is displayed, then release.**

* *Use these same three keys on both the 25-key and 28-key keypads.*

** *"**M660**" is displayed for firmware dated after September 28, 2001. Previously "**M650**" was displayed.*

Chapter 3 SETUP PARAMETERS

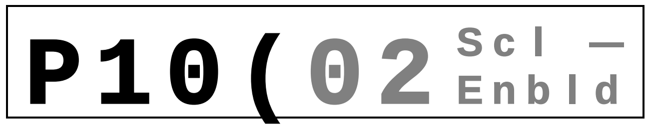
This chapter contains detailed information on the parameter setup mode. Topics covered include accessing the parameter setup mode, navigating setup parameters, entering parameter values, exiting the parameter setup mode, and downloading setup parameters.

OVERVIEW

SETUP PARAMETERS

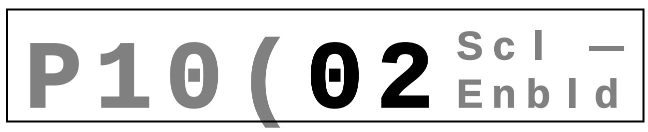
Setup parameters are dedicated memory registers within the 60 Series Instrument that collectively define how the scale operates. These parameters can only be accessed by entering the setup mode as described later in this chapter.

A setup parameter is identified by the letter “P” followed by a three, four or five-digit number and a decimal point. The example below shows parameter 109, the parameter used to enable a scale for use.

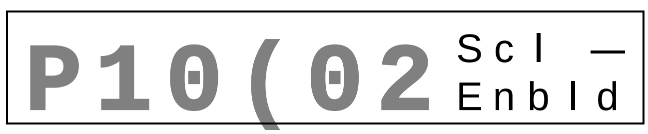


Refer to Appendix D for a complete list of setup parameters and parameter selections.

The digits to the right of the decimal point define the current numeric *selection* for the displayed parameter.



The parameter name and text selection appears in the prompting section of the display.



ACCESSING THE PARAMETER SETUP MODE

Viewing and/or editing setup parameters requires that you first access the setup mode.

VIEW-ONLY ACCESS

You can key in the full-edit access code when viewing parameters if you with to begin making changes.

It is possible to view the current configuration of any 60 Series Instrument by simply keying in the desired parameter number and pressing **[SELECT]**, then **[ENTER]**. For example,

will display the first setup mode parameter. Note that the display briefly shows --No- Mods! before accessing the parameter. This indicates that no modifications can be made to any parameter. Changing parameter selections are only possible by entering the setup mode using the full-edit access method.

FULL-EDIT ACCESS

Example:
Accessing Setup Mode / Full-Edit

[1] [0] [0] [SELECT]

SetUp Keyin Code:

[2] [3] [6] [4] [0] [ID] [ENTER YES]

SetUp Allow Mods!

P10*01 Scale 1

In order to change the current selection for any of the setup parameters you must access the setup mode by keying in the desired parameter number and entering a security access code. The default GSE access code is **23640 [ID] [ENTER]**. For example,

[1] [0] [0] [SELECT] [2] [3] [6] [4] [0] [ID] [ENTER YES]

will display the first setup mode parameter and allow changes to all parameter selections (see example: *Accessing Setup Mode / Full-Edit*).

Accessing a Specific parameter

You can access a specific parameter by keying in the parameter number (and applicable instance) prior to pressing **[SELECT]**. For example,

[5] [1] [0] [0] [·] [2] [SELECT] [2] [3] [6] [4] [0] [ID] [ENTER YES]

will proceed directly to parameter 5100 for setpoint #2 upon entering the setup mode. This method also works with view-only and limited access.

LIMITED ACCESS

If the program jumper is set to 'NO' you will not be granted access to edit setup parameters by any means. Attempting to do so will result in a Code 16 Check Jump r **error message.**

Whereas the full-edit access method allows changes to all setup parameter selections, using the limited access code permits changes to be made to all parameters except any macros or custom transmits that are protected by limited access.

The default GSE limited access code is **21353 [ENTER]**. Thus,

[1] [0] [0] [SELECT] [2] [1] [3] [5] [3] [ENTER YES]

will access the setup mode using limited access. Any macros or custom transmit tables that are protected by limited access (yes) at P9994 or P999 respectively will not be viewable, cannot be changed and will not appear in the download output at P64000. This allows the programmer to give others access to important field configurable parameters such as zero tracking and motion delay while protecting the main program routines from unauthorized changing or copying. When using the limited access code it is advisable to assign a PIN number to the full-edit access code at P400 as the default GSE access code is widely known and if used will give full access to all setup parameters.

Table 3-1: Parameters Affected by Limited Access Code

PARAMETER	DESCRIPTION COMMENT
P10001-19999	Macros will not be viewable if specified as limited access at P9994.
P400-402	PIN#s will not be viewable.
P1000-4999	Custom transmits will not be viewable or printable if specified as limited access at parameter P999.
P50001	Macro debug will not be viewable if specified as limited access at P50000.
P64000-64001	Sending setup will not print any parameter protected by limited access.

PIN NUMBER ACCESS

A PIN number can be assigned at P400 and P402 to override the GSE default full-edit access and limited access codes. If a PIN number is in effect, use the PIN number as the access code omit the **[ID]** key from the keypress sequence. For example,

[1] [0] [0] [SELECT] [7] [8] [7] [5] [ENTER YES]

will allow access to the setup mode if stored PIN number is '7875'.

⚠ DO NOT FORGET YOUR PIN NUMBER!

If you forget your PIN number, you will not be able to access the setup mode by any means. There is no "back door" access. If the PIN number is forgotten, you must either send the indicator back to GSE to have the PIN number cleared and restored to the default access code or replace the EEPROM. The later will force a default setup upon power-up, however ALL SETUP INFORMATION WILL BE PERMANENTLY LOST!

ACCESS DENIED

If you cannot access the setup mode for edit using the methods previously described, there are two probable reasons – the wrong code was entered or the main board program jumper is in the 'NO' position.

Wrong Access Code

If you enter the setup mode access code incorrectly, a **Code11** WRONG CODE! error message is displayed and access is denied. If you are sure you entered the code correctly, then it is likely that a PIN number has been entered or changed.

Program Jumper in 'NO' Position

It is not possible to enter the setup mode by any means if the main board program jumper is in the 'NO' position. Attempting to do so will result in a **Code16** Check Jumpr error message. Move the program jumper to the 'YES' position and try again.

Keys Disabled

Front panel keys can be disabled in the setup mode or redefined to invoke macro routines. This could effect the normal use of the keys required to invoke the setup mode. This situation usually does not generate an error message – it simply ignores key presses or performs other functions when keys are pressed. You can reset the keypad to a normal condition by holding down the **[CLR]** key on power-up until Macro Disbl is displayed. This will enable all keys to their normal function and inhibit the execution of all macros, thus allowing you to access the setup mode.

ACCESSING THE 460 SETUP MODE

The M460 uses the same access codes as previously described in this section, however since it does not have a numeric keypad the access method is different.

VIEW-ONLY ACCESS (MODEL 460)

Access the setup parameters in the view-only mode by pressing **[ZERO] + [SELECT]**, then **[TARE/ENTER]**.

ZERO + SELECT TARE

FULL-EDIT ACCESS (MODEL 460)

Access the setup parameters in the full-edit mode by pressing **[ZERO] + [SELECT]**, then **[SELECT] [ZERO] [PRINT] [UNITS] [TARE/ENTER]**.

ZERO + SELECT SELECT ZERO PRINT UNITS TARE

Accessing a Specific parameter

Accessing a specific parameter requires the scale to be in a mode such as the tare mode that allows numeric entry using the character scroll keys (see *Character Entry* on page 5-6). You can then scroll in the desired parameter number prior to pressing **[SELECT]**. For example, from the tare mode

5 0 0 SELECT SELECT ZERO PRINT UNITS TARE

will access P500 directly.

LIMITED ACCESS & PIN NUMBER ACCESS (MODEL 460)

Accessing the setup mode using limited access or PIN number access requires use of the character scroll keys (see *Character Entry* on page 5-6) to enter the code. For example,

ZERO + SELECT 2 1 3 5 3 TARE

will access the setup parameters using the GSE default limited access code. PIN numbers are also entered this way.

NAVIGATING SETUP PARAMETERS

Once you have entered the setup mode you can move freely through all parameters to view and/or change any parameter's configuration.

ADVANCING THROUGH PARAMETERS

Press **[SELECT]** to advance sequentially through all parameters. Multiple-instance parameters will be repeated for each enabled instance.

Press **[SCALE SELECT]** or **[.] [SELECT]** to move back one parameter. On the M460 press **[PRINT] [SELECT]**.

ACCESSING A SPECIFIC PARAMETER

Key in a parameter number and press **[SELECT]** to access that parameter directly. For example,

3 0 0 SELECT

will take you directly to P300 from any other parameter. On the M460, use of the character scroll keys (see *Character Entry* on page 5-6) to enter the parameter number.

Multiple Instance Parameters

If an instance is not specified for multiple instance parameters, the parameter accessed will be the first instance of the first parameter within the parameter group.

A multiple instance parameter can be directly accessed by including the desired instance number with the parameter number. Separate the parameter number and instance with a decimal. For example,

2 0 0 · 2 SELECT

will proceed directly to P200 (baud rate) for communication port 2.

OTHER NAVIGATING TOOLS

There are several short-cut and special function keys used to aid in navigating the setup mode. The function of these keys depends on the currently selected setup parameter (see Table 3-2). For a complete list of key functions, see page A-6.

Table 3-2: Setup Mode Key Functions

660 SERIES KEY	560 SERIES KEY	465 KEY	460 KEY	DESCRIPTION
SELECT	SELECT	SELECT	SELECT	Advances to the next parameter or directly to a keyed-in parameter.
SCALE SELECT	SCALE SELECT	SCALE SELECT	TARE + SELECT	Moves back one parameter.
ENTER YES	ENTER YES	ENTER YES	TARE	Scrolls through a list of choices or enters a keyed-in value.
CLR NO	CLR NO	CLR NO	ZERO	Clears a keyed-in value or an entry in process.
TARE	· · SELECT	· · SELECT	UNITS + TARE	Advances to the next instance of the currently displayed setup parameter.
UNITS	· 0 SELECT	· 0 SELECT	PRINT + TARE	Moves back to the previous instance of the currently displayed setup parameter.
· # SELECT	· # SELECT	· # SELECT		Moves directly to the instance specified by '#' for the currently displayed setup parameter (for example · 1 2 SELECT).
ID	F3 ID	ID	PRINT + SELECT	Shows the instance of the currently displayed multi-instance setup parameter.
F2	F2 STOP	TARGET	PRINT + UNITS	Toggles between an operating parameter's name and its parameter number / instance.
F1	F1 START	F1	PRINT	Begins alpha entries. Scrolls forward through alpha characters. Scrolls through the list of operating parameters in the "Pick Parm" list.
F4	F2 STOP	TARGET	PRINT + UNITS	Scrolls backward through alpha characters. Toggles between the normal and expanded view modes in a custom transmit table.
F5	TARE	TARE	UNITS	Advances one character in the custom transmit, input interpreter and macro tables. Shifts right during alpha entry to begin scrolling next character.
F3	UNITS	UNITS	ZERO + PRINT	Moves back one character in the custom transmit, input interpreter and macro tables. Shifts left (backspace) one character during alpha entry.
ZERO	ZERO	ZERO	ZERO	Exit setup mode or enter calibration.

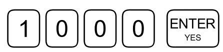
PARAMETER TYPES

There are three types of setup parameters – parameters that require a value to be keyed in, parameters that require a numeric entry representing one selection from a list of choices and parameters that require the entry of an operating parameter. Refer to Table 3-3 for each parameter's type.

KEY-IN VALUE PARAMETERS

A key-in value parameter requires a number or name to be entered. The entry will appear as the new parameter value exactly as it was keyed in. Examples of key-in parameters include full scale capacity (P110), time (P500) and variable names (P682).

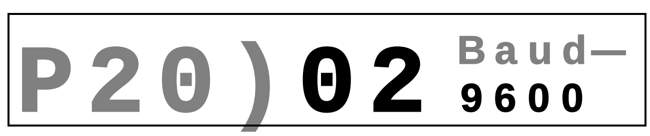
To change the value of a key in parameter, simply key in the desired value and press **[ENTER]**. For example,



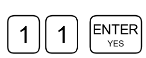
will change the full scale capacity to 1000.

SELECTABLE VALUE PARAMETERS

A selectable value parameter requires the entry of a numeric value that corresponds to a selection from a list of choices. The number entered will be displayed to the right of the parameter number and the lower portion of the prompting display will show the text equivalent of the selection. Examples of selectable value parameters are units (P131), baud rate (P200) and beeper volume (P460).



To change a selection, key in the number corresponding to the desired selection (see Appendix D for a complete list of parameter selection values). For example,



will change the baud rate to 57600 ('11' is the selection value for 57600 baud). If you don't know the selection number you can press **[ENTER]** to scroll sequentially through all selections.

OPERATING PARAMETER ENTRY PARAMETERS

Some setup parameters require the entry of an operating parameter (see Table 7-1for a complete list of operating parameters). The entry will appear as the name of the referenced parameter. Examples of setup parameters requiring operating parameter entries are select modes (P300), database columns (P701) and setpoint comparison parameters (P5150).

```
P515)   CmPar
        Gross
```

To change an operating parameter, key in the desired parameter number and instance (if required) and press **[ENTER]**. For example, selecting P5150 and keying in

[1] [.] [3] [ENTER YES]

will enter the net weight of scale 2 as the setpoint compare parameter. If you forget to enter a required instance, the display will prompt you to enter one.

If you do not know the number of the parameter you wish to specify, you can press **[ENTER]** to display the `Pick Parm:` list. The list starts with the “Gross” parameter. Pressing the up/down arrow keys will scroll forwards and backwards through all parameters. Press **[ENTER]** again to choose the displayed parameter and enter the instance if prompted.

Clearing an Operating Parameter

If you wish to clear an operating parameter rather than enter a new one, key in **99 [ENTER]** as the operating parameter selection. The display will show `None!` as the operating parameter name. Note that some setup parameters such as the setpoint compare parameter (P5150) require you to specify an operating parameter before allowing you to exit the setup mode. If you have not entered a required operating parameter before exiting, the display will revert to the offending parameter and allow you to enter one.

Viewing an Operating Parameter’s Number & Instance

When viewing an operating parameter entry, all you see is the parameter name. You can toggle the display to show the parameter’s number and instance by pressing **[F2]** on a 660 Series Controller, **[TARGET]** on a 465 or **[UNITS]** on a 460.

EXITING THE PARAMETER SETUP MODE

You can exit the setup mode from any parameter by pressing **[ZERO]** to initiate the exit routine. When exiting you are given the opportunity to calibrate and save or undo all changes you made while in the setup mode.

EXIT AND SAVE CHANGES

The most common procedure for exit the setup mode is pressing

[ZERO] [CLR/NO] [ENTER/YES] [ENTER/YES]

to exit and save all changes without calibrating the scale. With each keystroke you are prompted through the exit sequence (see example: *Exiting & Saving Changes*).

Example:
Exiting & Saving Changes

[ZERO]

`SetUp CAL ?`

`SetUp ENTER =CAL!`

[CLR/NO]

`SETUP Save Mods?`

`SETUP ENTER =SAVE`

[ENTER/YES]

`SETUP Sav'g Setup`

`SETUP MODS SAVED`

`SETUP Doing Setup`

`SETUP Exit Setup`

`SETUP ENTER =EXIT`

[ENTER/YES]

`)00 lb± 2 3 Gross`

EXIT AND UNDO CHANGES

You can exit the setup mode without saving any parameter changes by pressing **[CLR/NO]** at the `ENTER=SAVE` prompt. The display will then show

SETUP ENTER =UNDO

at which point you can press **[ENTER]** to undo all changes and **[ENTER]** again at the `ENTER=EXIT` prompt to exit without saving.

CANCEL EXIT

Pressing **[SELECT]** at any time while exiting the setup mode will cancel the exit routine and revert to the last parameter accessed. If changes were already saved, returning the setup mode before exiting will not undo changes.

EXIT ERROR MESSAGES

When you exit the setup mode, parameters are checked to ensure proper selections were made. For example, selecting even parity and 2 stop bits is not a valid combination for communication protocol. When you press **[ZERO]** to exit the setup mode, an error message is displayed. This type of error must be corrected before you are allowed to exit. Pressing any key while the error message is displayed will advance immediately to the offending parameter and allow you to correct it.

Less critical errors will display a message and wait for you to press a key to acknowledge the error. A `Code39 ƒA/D Cal` is an example of this type of warning.

Non-critical messages such as an indication that the clock speed has changed are displayed briefly when exiting the setup mode and do not require acknowledgement to complete the exit process.

A complete list of error messages can be found in Appendix E.

CALIBRATE DURING EXIT ROUTINE

Every time you exit the setup mode, the first prompt to appear is a request to calibrate the scale.

SETUP	**ENTER =CAL!**

Usually this is only necessary during initial setup or when a change has been made to a scale parameter such as the full scale capacity. To enter the calibration mode, press **[ENTER]** at the calibration prompt. When the calibration is complete the exit routine will resume with the ENTER=SAVE prompt at which point both calibration and setup information can be saved.

Refer to chapter 4 for information on the calibration procedure.

DOWNLOADING SETUP PARAMETERS

Once you have completed the scale setup you can download all of the setup information through any of the communication port to a computer to create a backup file or to another indicator to “clone” the setup. Refer to Chapter 14, Information Parameters for the download procedure.

SETUP PARAMETER MAP

Table 3-3 lists all setup mode parameters for the 60 Series Instrumentation. Parameters and selections apply to all 60 Series Instruments except as otherwise noted.

Table Key:

M Multiple Instance Parameter
L Select from List
K Key-In Parameter
P Operating Parameter

Table 3-3: Complete Parameter Listing

SETUP PARAMETER	M	L	K	P	DESCRIPTION	SELECTIONS	REFERENCE
Scale Configuration							
P10*01 Scale 1	✓				**Scale Instance Selection** *(Defines instance for P109 → P136)*	**460 Series** Scale: 1 → 2 **560 Series** Scale: 1 → 4 **660 Series** Scale: 1 → 8	3-26
P10(02 Scl — Enbld	✓	✓			**Scale Enable**	Disabled, Saved, Enabled	3-26
P11) F.S.— 100.0	✓		✓		**Full Scale Capacity**	0.01 → 1,000,000	3-27
P11!09 1div— .01	✓	✓			**Division Size**	0.00001 → 500	3-27
P11@ ZTAp— 1.0 d	✓		✓		**Zero Track Divisions**	Off, 0.1d → 20.0d *(Enter as 0 → 200)*	3-27
P11# ZTDl— 0.5 s	✓		✓		**Zero Track Delay**	0.05s → 10.0s *(Enter as 0 → 100)*	3-27
P11$ Motn— 1.0 d	✓		✓		**Motion Divisions**	Off, 0.1d → 20.0d *(Enter as 0 → 200)*	3-27
P11% MtDl— 1.0 s	✓		✓		**Motion Delay**	0.05s → 10.0s *(Enter as 0 → 100)*	3-27
P11^09 Fltr— 4.0sA	✓	✓			**Digital Filter**	Off, 0.06s → 8.0s	3-27
P11& Rate— 0.05s	✓		✓		**Display Update Rate**	0.05s → 20.0s *(Enter as 0 → 200)*	3-28
P11*12 Zrng— 100.%	✓	✓			**Zero Range**	0.01% → 100%	3-28
P11(00 Lnrz— Disbl	✓	✓			**Linearization**	Disabled, Enabled	3-28
P12@03 RTZ — 0.1%	✓	✓			**Return To Zero**	0.01% → 100%	3-28
P12$ QRes— 0.000	✓		✓		**Count Resolution** *(P179 must be 'Enabled')*	100.0 → 1,000,000	3-28
P12% EFac— 1.000	✓		✓		**Count Adjustment Factor** *(P179 must be 'Enabled')*	0.100 → 20.00	3-28
P12^ L.R.— 0.000	✓		✓		**Low Range Capacity**	0.01 → 1,000,000	3-28
P12&09 LDiv— .01	✓	✓			**Low Range Division Size**	0.00001 → 500	3-28

SETUP PARAMETER		M	L	K	P	DESCRIPTION	SELECTIONS	REFERENCE
Scale Configuration (continued)								
P12*	M.R.— 0.000	✓		✓		**Middle Range Capacity**	0.01 → 1,000,000	3-28
P12(09	MDiv— .01	✓	✓			**Middle Range Division Size**	0.00001 → 500	3-28
P13)00	Rnge— Gross	✓	✓			**Multi-Range Mode**	Gross, Net	3-28
P13!00	Unit— lb	✓	✓			**First Units**	lb = pounds kg = kilograms oz = ounces g = grams ton = US tons t = metric tons ????1 = custom unit 1 ????2 = custom unit 2 lb oz = pounds & ounces NONE = disable units 2, 3, 4	3-28
P13@01	Unit— kg	✓	✓			**Second Units**		
P13#09	Unit— NONE	✓	✓			**Third Units**		
P13$09	Unit— NONE	✓	✓			**Fourth Units**		
P13%	RMP.— Off	✓		✓		**Rate Measurement Period**	0=Off, 0.02s → 900s	3-28
P13^00	RTU.— sec	✓	✓			**Rate Time Unit**	Seconds, Minutes, Hours	3-28
P14@01	DspCZ Enbld		✓			**Center-Of-Zero Annunciator**	Disabled, Enabled	3-29
Status								
P14#	OLNam O			✓		**Status Overload Name**	Name Entry *(9 characters maximum)*	3-29
P14$	MtNam M			✓		**Status Motion Name**		
P14%	StNam S			✓		**Status Stable Name**		
P14^	ULNam O			✓		**Status Underload Name**		
P14&	ErNam E			✓		**Status Error Name**		
Units								
P15)00	UNITS =lb		✓			**Default Units**	Same as P131 → P134	3-29
P15!	Unam1 ????1			✓		**Custom Unit1 Name**	Name Entry *(5 characters maximum)*	3-29
P15@	Ucon1 1.000			✓		**Custom Unit1 Conversion Factor**	0.000001 → 9,999,999	
P15#	Unam2 ????2			✓		**Custom Unit2 Name**	Name Entry *(5 characters maximum)*	
P15$	Ucon2 1.000			✓		**Custom Unit2 Conversion Factor**	0.000001 → 9,999,999	
Tare Functions								
P16@00	TrNEG Disbl		✓			**Negative Tare Enable**	Disabled, Enabled	3-29
P16#01	TrRND Enbld		✓			**Tare Rounding Enable**	Disabled, Enabled	3-29
Analog Output *(P171 must be enabled to access P172 → P176)*								
P17)01	AnOut 1	✓				**Analog Output Instance Selection** *(Defines instance for P171 → P176)*	**460 Series** Analog Output: 1 → 2 **560 Series** Analog Output: 1 → 4 **660 Series** Analog Output: 1 → 8	3-30

SETUP PARAMETER	M	L	K	P	DESCRIPTION	SELECTIONS	REFERENCE
Analog Output (continued)							
P17!00 AOut— Disbl	✓	✓			**Analog Output Enable**	Disabled, Enabled	3-30
P17@ Parm— Gross	✓			✓	**Output Parameter**	Valid Operating Parameter	3-30
P17# F.S.— None!	✓			✓	**Full Scale Output**		3-30
P17$ Zero— None!	✓			✓	**Zero Offset**		3-30
P17% Rnge— None!	✓			✓	**Output Signal Range**		3-30
P17^00 Dflt— Max	✓	✓			**Default Output in Setup Mode**	Maximum, Minimum, Same	3-30
P17&00 Type— 0-10v	✓	✓			**Output Signal Type**	0-10VDC, 0-20mA, 4-20mA	3-31
Counting ***(P179 must be enabled to access P180 → P189)***							
P17(01 Count Enbld		✓			**Count Enable**	Disabled, Enabled	3-31
P18)00 Asmpl off		✓			**Auto Sample Enable**	Off, On	3-31
P18!01 Aenhn on		✓			**Auto Enhance Enable**	Off, On	3-31
P18@10 SmpSz 10			✓		**Default Sample Size**	1 → 9999	3-31
P18# %Accy 98.52			✓		**Required Accuracy**	0, 90% → 99.96%	3-31
P18$00 AcDsp off		✓			**Accuracy Display Enable**	Off, On	3-31
P18%00 PreSm None!		✓			**Pre-Sample Scale**	**460 Series** Scale: None, 1 → 2 **560 Series** Scale: None, 1 →4 **660 Series** Scale: None, 1 →8	3-31
P18^00 AftSm None!		✓			**After-Sample Scale**	**460 Series** Scale: None, 1 → 2 **560 Series** Scale: None, 1 → 4 **660 Series** Scale: None, 1 → 8	3-31
P18&06 SmpFl 4.0 s		✓			**Sample Filter**	None, 0.13s → 8.0s	3-32
P18*00 AcEnf off		✓			**Enforce Sample Accuracy**	Off, On	3-32
P18(00 SmpMt 0.0 d			✓		**Sample Motion Divisions**	0.0d → 1.5d *(Enter as 0 → 15)*	3-32
Communication Ports							
P19(01 Port— 1	✓				**Serial Port Instance Selection** *(Defines instance for P200 → P219)*	**460 Series** Port: 1 → 2 **560 Series** Port: 1 → 3 **660 Series** Port: 1 → 4	3-32
Communication Ports (continued)							

SETUP PARAMETER	M	L	K	P	DESCRIPTION	SELECTIONS	REFERENCE
P20)02 Baud— 9600	✓	✓			**Baud Rate**	**460 Series** 58300*, 38400, 37400, 19200, 9600, 4800, 2400, 1200, 600, 300, 150 * *Available for comm1 only* **560 Series** 58300*, 38400, 37400, 19200, 9600, 4800, 2400, 1200, 600, 300, 150 * *Available for comm1 only* **660 Series** 115K*, 112K*, 57600*, 56200*, 39300, 38400, 19200, 9600, 4800, 2400, 1200, 600, 300, 150 * *Available for comm1 only*	3-32
P20!01 Data— 8bits	✓	✓			**Data Bits**	7, 8	3-32
P20@00 Prty— none	✓	✓			**Parity**	None, Even, Odd	
P20#00 Stop— 1bit	✓	✓			**Stop Bits**	1, 2	
P20$02 Flow— Xon	✓	✓			**Handshaking**	None, CTS, Xon, Both	
P20%01 Recv— Std	✓		✓		**Receive Mode**	0=Disabled, 1=Standard, 2=Interpreter, 3=Modbus, Macro: 4 → 250	3-32
P20^00 Full— delay	✓	✓			**Transmit Mode**	Delay, Abort	3-32
P20& TxBf— 512	✓		✓		**Transmit Buffer Size (bytes)**	**460 Series** 8 → 3967 **560 Series** 8 → 3967 **660 Series** 8 → 16255	3-33
P20* RxBf— 1024	✓		✓		**Receive Buffer Size (bytes)**	**460 Series** 8 → 3967 **560 Series** 8 → 3967 **660 Series** 8 → 16255	
P20(01 MdAd— 1	✓		✓		**Modbus Address** *(P205 must be 'Modbus')*	1 → 247	3-33
P21)00 MdMd— ASCII	✓	✓			**Modbus Mode** *(P205 must be 'Modbus')*	ASCII, RTU	3-33
P21!00 MdWO— HiLo	✓	✓			**Modbus Word** *(P205 must be 'Modbus')*	HiLo, LoHi	3-33
Input Interpreter ***(First select a communication port at P199; P219 must be assigned to access P220 → P224)***							
P21& NoNUL Disbl	✓		✓		**NULL Character Enable**	Disabled, Enabled	3-33
P21* RxTrm <LF>	✓		✓		**Receive Termination Character**	0 → 255 *(Enter as .000 → .255)*	3-33
P21(00 RxIn# None!	✓				**Input Interpreter Instance Selection** *(Defines instance for P220 → P224)*	**460 Series** Interpreter: 1 → 15 **560 Series** Interpreter: 1 → 100 **660 Series** Interpreter: 1 → 250	3-33
Input Interpreter (continued)							
P22) RxNam None!	✓		✓		**Interpreter Name**	Name Entry *(79 characters maximum)*	3-33

Setup Parameter	M	L	K	P	Description	Selections	Reference
P22!00 RxTyp Char	✓	✓			**Interpreter Type**	Character, Line	3-33
P22@ Rx 1 ††††‡	✓		✓	✓	**Line Interpreter Entry Table** *(P221 must be set for 'Line')*	Table Entry *(text, parameter, control code)*	3-33
P22# RxChr <NUL>	✓		✓		**Interpreter Character** *(P221 must be set for 'Character')*	0 → 255 *(Enter as .000 → .255)*	3-34
P22$00 RxMac None!	✓		✓		**Interpreter Macro#**	**460 Series** 1 → 15 **560 Series** 1 → 100 **660 Series** 1 → 250	3-34
Numeric Parameter Formatting							
P24)08 Width 8			✓		**Minimum Transmit Width**	1 → 15	3-34
P24!00 SgnJu Right		✓			**Sign Justification**	Left, Right	3-34
Networking & Remote Communications							
P25)00 NtWrk Disbl		✓			**Network Enable** *(Applies to communication port #1 only)*	Disabled, Enabled	3-34
P25!00 Addrs Disbl			✓		**Network Address** *(P250 must be 'Enbld', P205 not 'Modbs')*	0=Disable, 4 → 254	3-34
P29)00 Echo Off		✓			**Echo Display**	**460 Series** Off, Comm: 1 → 2 **660 Series** Off, Comm: 1 → 3 **660 Series** Off, Comm: 1 → 4	3-35
P29!02 Start <STX>			✓		**Echo Start Character**	0 → 255	3-35
P29@03 End <ETX>			✓		**Echo End Character**	0 → 255	3-35
P29#00 RmDsp Disbl	✓	✓			**Remote Display Enable**	0=Disable, 1=LCD, 2=LED	3-35
P29$00 Rm BL Disbl	✓	✓			**Remote Display Backlight Enable**	0=Disable, 1=Enable	3-35
Weigh Mode Parameter Selections							
P30) MODE0 Gross				✓	**[SELECT] Mode 0**	Valid Operating Parameter	3-35
P30! MODE1 Net				✓	**[SELECT] Mode 1**		
P30@ MODE2 Tare				✓	**[SELECT] Mode 2**		
P30# MODE3 None!				✓	**[SELECT] Mode 3**		
P30$ MODE4 None!				✓	**[SELECT] Mode 4**		
P30% MODE5 None!				✓	**[SELECT] Mode 5**		
P30^ MODE6 None!				✓	**[SELECT] Mode 6**		
Weigh Mode Parameter Selections (continued)							
P30& MODE7 None!				✓	**[SELECT] Mode 7**		

Setup Parameter		M	L	K	P	Description	Selections	Reference
P30*	MODE8 None!				✓	[SELECT] Mode 8		
P30(	MODE9 None!				✓	[SELECT] Mode 9		
Access Codes								
P40)	PIN None!			✓		**Personal Identification Number**	Alpha-Numeric Entry *(5 characters maximum)*	3-35
P40!	QCAL None!			✓		**Quick Calibration Access Number**	Alpha-Numeric Entry *(5 characters maximum)*	3-35
P40@	Lmt'd 21353			✓		**Limited Access Number**	Alpha-Numeric Entry *(5 characters maximum)*	3-35
OIML								
P41)	OIML Disbl			✓		**OIML Enable**	Disabled, Enabled *(Enter as 9990 or 9991)*	3-36
P41!00	LANG USA		✓			**Language Character Set**	USA, France, German, UK, Denmark, Sweden, Italy, Spain, Japan, Norway, Denmark2, Spain2, Latin America	3-36
P41@00	PrSET Disbl		✓			**Preset Enable**	Disabled, Enabled	3-36
VFD/LCD Display Setup								
P42)01	Dsply ON		✓			**Standard VF Display Mode**	Off, On, Auto	3-36
P42!02	WtThr 6d		✓			**Weight Threshold Divisions** *(Used only if P420 set to 'Auto')*	2d → 32d	3-36
P42@05	TmOut 5min		✓			**Timeout** *(Used only if P420 set to 'Auto')*	30s → 2hr	3-36
P42#10	Brite 100%		✓			**Display 'ON' Brightness**	10% → 100%	3-37
P42$00	Dim OFF		✓			**Display 'OFF' Dimness** *(Used only if P420 set to 'Auto')*	Off, 10% → 100%	
P42%00	4x20 Disbl		✓			**4X20 VF / LCD Display Enable** *(660 Series; Auto-detects LCD display)*	Disabled, Enabled, LCD Always	3-37
P43)12	Contr 12		✓			**LCD Contrast** *(LCD display must be installed)*	0 → 255	3-37
NTEP								
P44)00	NTEP Disbl		✓			**NTEP Enable**	Disabled, Enabled	3-37
Keypad								
P45)00	Keypd 28Key		✓			**Keypad Selection**	**460 Series** 5 Key, 22 Key *(Entered as 9990 or 9991)* **560 Series** 22 Key *(No entry allowed)* **660 Series** 28 Key, 25 Key *(Entered as 9990 or 9991)*	3-37
P45!06	KyRpt Fast		✓			**Keypad Repeat Rate**	None, Very Slow, Slow, Medium Slow, Medium, Medium Fast, Fast, Very Fast	3-37
P46)03	Beepr Low		✓			**Beeper Volume**	Off, Minimum, Extra Low, Low, Medium, Medium High, High, Maximum	3-37
Time & Date								
P50)00	Time 00:00			✓		**Time**	Time Entry *(Enter as HH.MM.SS)*	3-38
Time & Date (continued)								
P50!70	Date 01/01			✓		**Date**	Date Entry *(Enter as MM.DD.YY)*	3-38

SETUP PARAMETER		M	L	K	P	DESCRIPTION	SELECTIONS	REFERENCE
P50@00	TDAcc Disbl		✓			**Time/Date Access**	Disabled, Enabled	3-38
P50#01	AM/PM yes		✓			**AM/PM Time Format**	No, Yes	3-38
P50$00	Style U.S.A		✓			**Date Format**	USA, International	3-38
DSD Configuration *(P590 must be enabled to access P591 → P595)*								
P59)00	DSD Disbl		✓			**DSD Enable**	Disable, Enable	3-38
P59!1	Port None!		✓			**DSD Serial Port Selection**	**560 Series** Comm: 0 → 3 **660 Series** Comm: 0 → 4	3-38
P59@0	RxChr <NUL>		✓	✓		**DSD Receive Character**	0 → 255 *(Enter as .000 → .255)*	3-39
P59#1	CusTx 1		✓	✓		**DSD Custom Transmit Selection**	**560 Series** Custom Transmit: 1 → 100 **660 Series** Custom Transmit: 1 → 250	3-39
P59$0	MxRow 0			✓		**DSD Maximum Number of Rows**	0 → Maximum Rows Available	3-39
P59%0	#Warn 0			✓		**DSD Number of Warning Rows**	0 → Maximum Number of Rows *(per P594)*	3-39
Parameter Renaming								
P60)	Gross None!			✓		**Rename Gross**	Name Entry *(79 characters maximum)* Note: Only the first 5 characters of a parameter's name will be displayed. All characters will be printed.	3-39
P60!	Net None!			✓		**Rename Net**		
P60@	Tare None!			✓		**Rename Tare**		
P60#	GrTOT None!			✓		**Rename Gross Total**		
P60$	GrT+C None!			✓		**Rename Gross Total + Current**		
P60%	GrT-C None!			✓		**Rename Gross Total - Current**		
P60^	NtTOT None!			✓		**Rename Net Total**		
P60&	NtT+C None!			✓		**Rename Net Total + Current**		
P60*	NtT-C None!			✓		**Rename Net Total – Current**		
P60(	Accum None!			✓		**Rename Accumulation**		
P61)	Scale None!			✓		**Rename Scale Number**		
P61!	Tm/Dt None!			✓		**Rename Time/Date**		
P61%	AvGrs None!			✓		**Rename Average Gross**		
Parameter Renaming (continued)								
P61^	AvNet None!			✓		**Rename Average Net**		

SETUP PARAMETER		M	L	K	P	DESCRIPTION	SELECTIONS	REFERENCE
P61&	AvgCt None!			✓		Rename Average Count		
P61*	PkGrs None!			✓		Rename Peak Gross		
P61(	PkNet None!			✓		Rename Peak Net		
P62)	RndGr None!			✓		Rename Rounded Gross		
P62!	RndNt None!			✓		Rename Rounded Net		
P62#	Rate None!			✓		Rename Rate		
P62$	FreFl None!			✓		Rename Free Fall 1		
P62%	FutGr None!			✓		Rename Future Gross 1		
P62^	FutNt None!			✓		Rename Future Net 1		
P62&	FrFl2 None!			✓		Rename Free Fall 2		
P62*	FuGr2 None!			✓		Rename Future Gross 2		
P62(	FuNt2 None!			✓		Rename Future Net 2		
P63)	Qty None!			✓		Rename Quantity		
P63!	QtTOT None!			✓		Rename Quantity Total		
P63@	QtT+C None!			✓		Rename Quantity Total + Current		
P63#	QtT-C None!			✓		Rename Quantity Total - Current		
P63$	APW None!			✓		Rename Average Piece Weight		
P63%	APW*K None!			✓		Rename Average Piece Weight x K		
P63^	%Accy None!			✓		Rename Percent Accuracy		
P63&	Sampl None!			✓		Rename Sample		
P64)	GrAll None!			✓		Rename Gross Total of All Scales		
P64!	NeAll None!			✓		Rename Net Total of All Scales		
P64@	TrAll None!			✓		Rename Tare Total of All Scales		
P64#	GTAll None!			✓		Rename Total of All Gross Totals		
P64$	NTAll None!			✓		Rename Total of All Net Totals		
Parameter Renaming (continued)								
P64%	QuAll None!			✓		Rename Quantity Total of All Scales		

SETUP PARAMETER	M	L	K	P	DESCRIPTION	SELECTIONS	REFERENCE
P64^ QTAll None!			✓		**Rename Total of All Quantity Totals**		
Total & Tare Save							
P66)02 TotSv Auto		✓			**Total Values Save Method**	No Save, On Request, Auto	3-39
P66!00 TarSv NoSav		✓			**Tare Value Save Method**	No Save, On Request, Auto	
Variables *(Variables must allocated at P680 for access to P681 → P689)*							
P68)00 #Vars None!			✓		**Variable Allocation**	**460 Series** 0 → 15 **560 Series** 0 → 100 **660 Series** 0 → 999	3-40
P68!01 Var.# 1	✓				**Variable Instance Selection** *(Defines instance for P682 → P689)*	**460 Series** 1 → 15 **560 Series** 1 → 100 **660 Series** 1 → 999	3-40
P68@ VName None!	✓		✓		**Variable Name**	Name Entry *(79 characters maximum)*	3-40
P68$00 VSave NoSav	✓	✓			**Variable Value Save Method**	No Save, On Request, Auto	3-40
P68%00 VLock Disbl	✓	✓			**Variable Lock**	Disabled, Enabled	3-40
P68^00 VType Float	✓	✓			**Variable Type**	Float, Integer, Unsigned Integer, String	3-40
P68&06 FStyl Auto	✓	✓			**Float Style** *(P686 must be 'Float')*	**460 Series** 1 → 5 decimal places, Auto, Scale: 1 → 2 **560 Series** 1 → 5 decimal places, Auto, Scale: 1 → 4 **660 Series** 1 → 5 decimal places, Auto, Scale: 1 → 8	3-40
P68*00 IStyl Numbr	✓	✓			**Integer Style** *(P686 must be 'Int' or 'U-Int')*	Number, Time/Date, Time, Date	3-41
P68(10 Ssize 10	✓		✓		**String Size** *(P686 must be 'Strng')*	1 → 63	3-41
Database *(Databases must be assigned at P699 for access to P700 → P799; P701 → P799 are sequentially allocated as needed)*							
P69(00 DB #: None!	✓				**Database Instance Selection** *(Defines instance for P700 → P799)*	**460 Series** Database: 1 → 15 **560 Series** Database: 1 → 100 **660 Series** Database: 1 → 250	3-41
P70) DBNam None!	✓		✓		**Database Name**	Name Entry *(79 characters maximum)*	3-41
P70! Col01 *END*	✓			✓	**Database Column Parameter**	Valid Operating Parameter	3-41

Setup Parameter	M	L	K	P	Description	Selections	Reference
Keypad Key Assignments							
P80)00 Selct Enbld			✓		**[SELECT] Key Function**	**460 Series** Enabled, Macro: 1 → 15 **560 Series** Enabled, Macro: 1 → 100 **660 Series** Enabled, Macro: 1 → 250	3-41
P80!00 Zero Enbld			✓		**[ZERO] Key Function**		
P80@00 Tare Enbld			✓		**[TARE] Key Function**		
P80#00 Units Enbld			✓		**[UNITS] Key Function**		
P80$00 SSlct Enbld			✓		**[SCALE SELECT] Key Function**		
P80%00 Print Enbld			✓		**[PRINT] Key Function**		
P80^00 IDUse Enbld			✓		**[ID] Key Function**	**465 Only** 0=None, 1=Menu, 2=Database, 3=Menu & Database, Macro: 4 → 15 **560 Series** 0=None, 1=Menu, 2=Database, 3=Menu & Database, Macro: 4 → 100, 101 = Macro 3 **660 Series** 0=None, 1=Menu, 2=Database, 3=Menu & Database, Macro: 4 → 250	
P80&00 Enter Enbld			✓		**[ENTER] Key Function**	**460 Series** Enabled, Macro: 1 → 15 **560 Series** Enabled, Macro: 1 → 100 **660 Series** Enabled, Macro: 1 → 250	
P80*00 Clear Enbld			✓		**[CLEAR] Key Function**		
P80(00 DecPt Enbld			✓		**[.] Key Function**		
P81)00 '0' Enbld			✓		**[0] Key Function**		
P81!00 '1' Enbld			✓		**[1] Key Function**		
P81@00 '2' Enbld			✓		**[2] Key Function**		
P81#00 '3' Enbld			✓		**[3] Key Function**		
P81$00 '4' Enbld			✓		**[4] Key Function**		
P81%00 '5' Enbld			✓		**[5] Key Function**		
P81^00 '6' Enbld			✓		**[6] Key Function**		
P81&00 '7' Enbld			✓		**[7] Key Function**		
P81*00 '8' Enbld			✓		**[8] Key Function**		
P81(00 '9' Enbld			✓		**[9] Key Function**		
P82)00 AnyKy Enbld			✓		**Any Key Function**		

SETUP PARAMETER		M	L	K	P	DESCRIPTION	SELECTIONS	REFERENCE
Programmable Digital Input/Output – 660 Series Only *(Channels must be assigned a function at P851 for access to P852 → P864)*								
P85)01	Chan# 1	✓				**Channel Instance Selection** *(Defines instance for P851 → P864)*	Channel: 1 → 8	3-41
P85!02	Func— Setpt	✓	✓			**Channel Function**	None, Freq Out, Setpoint, Freq In A, Freq In B, Phase Time, Dly In, Dly Out, Q-Decode2, Q-Decode3, Freq Debounce	3-42
P85@	FNam— None!	✓		✓		**First I/O Parameter Name**	Name Entry *(79 characters maximum)*	3-42
P85#	Pnam— None!	✓		✓		**Second I/O Parameter Name**	Note: One or more of these parameters may not appear or may appear with a different name depending on the selection for P851.	3-42
P85$	Rnam— None!	✓		✓		**Third I/O Parameter Name**		3-42
P85^	MaxF— 65536	✓		✓		**Maximum Expected Frequency** *(P851 must be 'FDbnc')*	48 Hz → 10752 Hz	3-42
P85&00	CSrc— ClckA	✓	✓			**Clock Source** *(P851 must be 'FqOut', 'FqInA' or 'DlyOt')*	Clock A, Clock B	3-42
P85*00	Edge— Rise	✓	✓			**Pulse Edge Detection** *(P851 must be 'FqInA-B', 'DlyIn' or 'QdDc3')*	Rise, Fall	3-42
P85(00	Perd— 0.000	✓		✓		**Pulse Measurement Period** *(P851 must be 'FqInA' or 'FDbnc')*	0.001s → 500.0s	3-42
P86)00	#Pls— 1	✓		✓		**Number of Pulses to Measure** *(P851 must be 'FqInB' or 'PhsTm')*	1 → 255	3-42
P86!00	Mtyp— Low	✓	✓			**Phase Measurement Type** *(P851 must be 'PhsTm')*	Low, High	3-42
P86@00	Ptyp~ Low	✓	✓			**Pulse Type** *(P851 must be 'DlyOt')*	Low, High	3-43
P86#00	SFac— None!	✓			✓	**Pulse Scaling Factor** *(P851 'FqInA-B', 'PhsTm' or 'QdDc2-3')*	Valid Operating Parameter	3-43
P86$00	# DP— 0 dp	✓	✓			**Number of Decimal Places** *(P863 must be assigned a parameter)*	0 → 5 Decimal Places, Auto, Scale: 1 → 8	3-43
Custom Transmit *(Custom Transmits must be assigned at P989 for access to P990 → P4999)*								
P98)	TxRat 0.5 s			✓		**Continuous Transmit Rate**	0=Disabled; 0.1s → 25.0s *(Enter as 1 → 250)*	3-43
P98(01	CusTx 1	✓		✓		**Custom Transmit Instance Selection** *(Defines instance for P990 → P4999)*	**460 Series** Custom Transmit: 1 → 4 **660 Series** Custom Transmit: 1 → 100 **660 Series** Custom Transmit: 1 → 250	3-43
P99)	TxNam None!	✓		✓		**Transmit Name**	Name Entry *(79 characters maximum)*	3-43
P99!01	Send: OnReq	✓	✓			**Transmit Mode**	Off, On Request, Prompt *(On Request not available on the 460)*	3-43
P99@01	Port Comm1	✓	✓			**Serial Port Selection**	**460 Series** Comm: 1 → 2 **560 Series** Comm: 1 → 3, LCD **660 Series** Comm: 1 → 4, LCD	3-43
P99#01	CSMtn Dly'd	✓	✓			**Current Scale Motion**	Ignored, Delayed	3-44
Custom Transmit (continued)								

SETUP PARAMETER		M	L	K	P	DESCRIPTION	SELECTIONS	REFERENCE
P99$00	Mot'n Ignrd	✓	✓			**Scale Motion**	**460 Series** Any combination of scale numbers 1 → 2 **560 Series** Any combination of scale numbers 1 → 4 **660 Series** Any combination of scale numbers 1 → 8	3-44
P99%00	See P994!	✓				**Scale 2 Motion**	Ignored, Delayed *(These parameters are maintained for upload backward compatibility with the M650 and are not intended to accept keyboard entry; refer to P994).*	3-44
P99^00	See P994!	✓				**Scale 3 Motion**		
P99&00	See P994!	✓				**Scale 4 Motion**		
P99*00	Cont. Disbl	✓	✓			**Continuous Transmit Enable**	Disabled, Enabled	3-44
P99(00	LmtAc no	✓	✓			**Transmit Table Limited Access**	No, Yes	3-44
P100)	Tx 1 ††††,	✓		✓	✓	**Transmit Entry Table**	Table Entry *(text, parameter, control code)*	3-44
Setpoints ***(Setpoints must be assigned at P5099 for access to P5100 → P5150)***								
P509(1	Setpt 1	✓				**Setpoint Instance Selection** *(Defines instance for P5100 → P5150)*	**460 Series** Setpoint: 1 → 16 **560 Series** Setpoint: 1 → 48 **660 Series** Setpoint: 1 → 256	3-44
P510)0	SPTyp Disbl	✓	✓			**Setpoint Mode**	Disabled, Output, Input	3-45
P510!	SPNam None!	✓		✓		**Setpoint Name**	Name Entry *(79 characters maximum)*	3-45
P511)0	Activ Above	✓	✓			**Activation Condition**	**460 Series** Above, Below, Between, Outside, Always, Never, Motion1-2, Stable 1-2, Motion Current, Stable Current **560 Series** Above, Below, Between, Outside, Always, Never, Motion1-4, Stable 1-4, Motion Current, Stable Current **660 Series** Above, Below, Between, Outside, Always, Never, Motion1-8, Stable 1-8, Motion Current, Stable Current	3-45
P511!	AcDly 0.00	✓		✓		**Activation Delay**	0.01s → 5,767,168s	3-45
P511@0	AcMac None!	✓		✓		**Activation Macro#**	**460 Series** Macro: 1 → 15 **560 Series** Macro: 1 → 100 **660 Series** Macro: 1 → 250	3-45
P511#0	AcMtn Ign'd	✓	✓			**Activation Motion**	Ignored, Delayed	3-45
P511$	ALPar None!	✓			✓	**Lower Activation Parameter**	Valid Operating Parameter	3-45
P511%	AUPar None!	✓			✓	**Upper Activation Parameter**	Valid Operating Parameter	3-45
Setpoints (continued)								

SETUP PARAMETER	M	L	K	P	DESCRIPTION	SELECTIONS	REFERENCE
P513)0 Deact Above	✓	✓			**Deactivation Condition**	**460 Series** Above, Below, Between, Outside, Always, Never, Motion1-2, Stable 1-2, Motion Current, Stable Current **560 Series** Above, Below, Between, Outside, Always, Never, Motion1-4, Stable 1-4, Motion Current, Stable Current **660 Series** Above, Below, Between, Outside, Always, Never, Motion1-8, Stable 1-8, Motion Current, Stable Current	3-45
P513! DeDly 0.00	✓		✓		**Deactivation Delay**	0.01s → 5,767,168s	3-45
P513@0 DeMac None!	✓		✓		**Deactivation Macro#**	**460 Series** Macro: 1 → 15 **560 Series** Macro: 1 → 100 **660 Series** Macro: 1 → 250	3-45
P513#0 DeMtn Ign'd	✓	✓			**Deactivation Motion**	Ignored, Delayed	3-45
P513$ DLPar None!	✓			✓	**Lower Deactivation Parameter**	Valid Operating Parameter	3-46
P513% DUPar None!	✓			✓	**Upper Deactivation Parameter**	Valid Operating Parameter	3-46
P515) CmPar Gross	✓			✓	**Compare Parameter**	Valid Operating Parameter	3-46
Modbus Parameter Map *(Parameters are sequentially allocated at P6001 → P6247 as needed)*							
P600! Modbs None!				✓	**Modbus Address Translation Table**	Valid Operating Parameter	3-46
Macros *(Macros must be assigned at P9990 for access to P9991 → P19999)*							
P998)0 Abort None!			✓		**Abort Macro#**	**460 Series** Macro: 1 → 15 **560 Series** Macro: 1 → 100 **660 Series** Macro: 1 → 250	3-46
P998!0 Abort Menu		✓			**Macro Abort Method**	Menu, Immediate	3-46
P999)0 Mac.# None!	✓				**Macro Instance Selection** *(Defines instance for P9991 → P19999)*	**460 Series** Macro: 1 → 15 **560 Series** Macro: 1 → 100 **660 Series** Macro: 1 → 250	3-46
P999! MName None!	✓		✓		**Macro Name**	Name Entry *(79 characters maximum)*	3-47
P999@0 Invok Std	✓	✓			**Macro Priority**	Standard, Immediate	3-47
P999#0 Menu Disbl	✓	✓			**Macro Menu Enable**	Disabled, Enabled	3-47
P999$0 LmtAc no	✓	✓			**Macro Table Limited Access**	No, Yes	3-47
P1000! Mc 1 ††††‡	✓		✓		**Macro Entry Table**	Table Entry *(text only)*	3-47
Macro Debug							
P5000) LmtAc no		✓			**Macro Debug Table Limited Access**	No, Yes	3-47
P5000! None! ‡‡‡‡‡					**Macro Debug Table**	Read-Only Diagnostic Table	3-47

PARAMETER DESCRIPTIONS

This section provides a brief description of each setup parameter. Parameters are presented in numeric order. Refer to this section when installing options to ensure proper configuration.

SCALE CONFIGURATION

The scale configuration parameters provide the basic configuration for each enabled scale.

P108: Scale Instance

Sets the scale# in effect when accessing the remainder of the scale configuration parameters P109 → P145.

P109: Scale Enable

Determines whether the scale in effect at P108 is disabled, saved or enabled.

When a scale is **disabled**, the scale is not accessible from the weigh mode. All of the scale's setup parameters (P110 → P145) are disabled for viewing in the setup mode and any previous configuration for that scale is lost. Calibration data for that scale is also lost along with A/D calibration values. Therefore, do not disable a scale if you intend to re-enable it. Instead select the save option.

A **saved** scale is not accessible from the weigh mode, however all scale configuration including calibration data and A/D calibration values are retained. Thus a saved scale can be re-enabled without having to be reconfigured or re-calibrated.

An **enabled** scale is a fully active scale accessible from the weigh mode for viewing via the **[SCALE SELECT]** key. All weight-based operating parameters for the enabled scale will be considered valid instances when using them in macros or when assigning operating parameters to setup parameters.

P110: Full Scale Capacity

Sets the scale's full scale capacity. The capacity is entered in terms of the default units specified at P150. Capacity entries of 100,000 or greater will be displayed with the 'kilo' abbreviation (i.e. 100K).

An overload condition is considered to be 104% of full scale.

P111: Division Size

Selects the scale's division size. Pressing **[CLR]** will automatically select the nearest division size less than or equal to 10,000 based on the capacity selected at P110. A warning message is displayed if you select a division size that exceeds 25,000 divisions.

P112: Zero Track Divisions

Selects the number of zero tracking divisions to a resolution of 0.1 divisions. For example, an entry of **35** will be accepted as ±3.5 divisions of zero tracking. If the live weight on the scale remains within the zero tracking range for a period of time specified by the zero track delay (P113), then the weight is tracked to center-of-zero.

Note that when the weight on the scale falls within the zero tracking range, the weight is not displayed providing a visual indication that zero tracking is in effect.

P113: Zero Track Delay

Selects the zero track time delay to a resolution of 0.1 seconds. For example, an entry of **15** will be accepted as 1.5 seconds.

P114: Motion Divisions

Selects the number of motion divisions to a resolution of 0.1 divisions. For example, an entry of **35** will be accepted as ±3.5 divisions of motion. If the live weight on the scale remains within the motion range for a period of time specified by the motion delay (P115), then the weight is considered to be stable.

Note that when the weight on the scale is considered to be in motion, the units will be not visible on the display. The units will be displayed once the scale becomes stable.

P115: Motion Delay

Selects the motion time delay to a resolution of 0.1 seconds. For example, an entry of **25** will be accepted as 2.5 seconds.

P116: Digital Filter

Selects the degree of A/D filtering used in calculating weight-based parameters. The longer the filter duration, the more stable the weight will appear. However, increasing the filter duration will also result in a slower response to rapidly changing weights and may therefore be undesirable in applications that require a prompt and accurate response to weight fluctuations. Auto-filter selections (identified as ‘sA’) can be used in such situations to provide a stable reading (heavy filter) when weight changes are small and switch to a light filter when the rate of change in weight increases.

P117: Display Update Rate

Selects the display update time delay to a resolution of 0.1 seconds. For example, an entry of **5** will be accepted as 0.5 seconds. The display update rate does not provide any filtering effects. It can be thought of as a shutter, controlling how often the display is updated to view the current weight value.

P118: Zero Range

Selects the amount of weight as a percentage of full scale that can be zeroed out using the **[ZERO]** key.

P119: Linearization

Enables the five-point linearization feature used during load cell calibration.

P122: Return-to-Zero

Selects a weight threshold as a percentage of full scale below which the gross weight must fall before another accumulation can be performed.

P124: Count Resolution

Selects the internal count resolution used in determining the quantity.

Normally this is set to zero (0) which uses the maximum internal resolution when determining the quantity. However, when displaying a very large count of very light pieces, the display may appear unstable. Decreasing the count resolution will increase the count division size, thus making the count appear more stable.

P125: Count Adjustment Factor

Assigns a conversion factor to the number of additional pieces that can be added and still ensure the required accuracy is met. For example, if the adjustment factor is 1.0 and the display reports you can add up to 200 additional pieces after performing a sample, changing the adjustment factor to 0.1 would allow you to add up to 2000 pieces – changing the adjustment factor to 10.0 would only allow you to add up to 20 pieces.

P126 – P130: Multi-Range

Configures the operation of the multi-range feature (see page 5-17).

P131 – P134: Units

Assigns the units selectable via the **[UNITS]** key from the weigh mode. The units assigned at P131 will become the power-up units for the scale presently specified at P108. Three additional units can be assigned at P132 → P134 for access via the **[UNITS]** key.

P135: Rate Measurement Period

Assigns the rate measurement period (RMP) over which the rate of weight change is averaged. For example, an entry of 2.5 will result in a 2.5 second rate averaging period. Thus, every 1/60th of a second (the A/D conversion rate) the rate reported at 23P will be updated to reflect the average rate over the last 2.5 seconds (150 readings). A longer the RMP will yield a more stable the rate display but will be slower to respond to rapid changes in rate.

P136: Rate Time Unit

Sets the time measurement criteria for calculating the rate (i.e. rate/second, rate/minute, rate/hour).

P142: Center-of-Zero Annunciator

Enables the center-of-zero annunciator. In multi-scale applications, disabling the center-of-zero annunciator will allow you to view the scale number when the weight is at center-of-zero.

STATUS

The status parameters allow renaming of the status word transmitted when using operating parameter 97P.

P143 – P147: Status Name

Assigns the transmitted text of the status parameter (97P) for overload, underload, motion, stability, underload and error (bad A/D) status at P143 → P147 respectively. Pressing **[CLR]** without an entry in process will restore the default status name.

UNITS

The units parameters are used to assign the default units of measure and provide custom unit configuration.

P150: Default Units

Selects the scale's default units of measure. All weight-based parameter values are stored in terms the default units.

P151 – P154: Custom Units

Configures up to two (2) custom units of measure. P151 and P153 are used to assign the name for custom unit 1 and custom unit 2 respectively. P152 and P154 are the conversion factors for custom unit 1 and custom unit 2 respectively. The conversion factor is a conversion from the default units specified at P150.

TARE FUNCTIONS

The tare functions are used to enable the negative tare and tare rounding features.

P162: Negative Tare

Enables the entry and use of negative tare values.

P163: Tare Rounding

Enables tare rounding. When enabled, the tare value is stored internally to the display resolution. This is done to ensure that the addition of multiple tare and net values will yield the correct sum when compared to the displayed values. When disabled, the tare value is stored to a higher precision and may result in a discrepancy between the accumulation of tare and net values as compared to the displayed values.

ANALOG OUTPUT

The analog output parameters provide configuration of the analog output modules.

P170: Analog Output Instance

Sets the analog output# in effect when accessing the remainder of the analog output configuration parameters P171 → P177.

P171: Analog Output Enable

Enables the analog output currently specified at P170.

P172: Output Parameter

Assigns the operating parameter that the analog output will track.

P173: Full Scale Output

Assigns the full scale analog output. If set to "***None!***" the full scale capacity assigned at P110 is assumed. To specify a different full scale value, assign a variable to P173. Then, assign the desired full scale value to the assigned variable. It is advisable to configure the variable for auto-save at P684 to ensure the value is retained during power loss.

P174: Zero Offset

Assigns the zero offset for the analog output. If set to "***None!***" the full zero offset is assumed to be zero (0). To specify a different zero offset value, assign a variable to P174. Then, assign the desired zero offset value to the assigned variable. It is advisable to configure the variable for auto-save at P684 to ensure the value is retained during power loss.

P175: Output Signal Range

Assigns the signal range for the analog output. If set to "***None!***" the signal range is assumed to be the maximum allowable output (10V or 20mA). To specify a different signal range value, assign a variable to P175. Then, assign the desired signal range value to the assigned variable. It is advisable to configure the variable for auto-save at P684 to ensure the value is retained during power loss.

If the output signal range is specified to be 5 for a 0-10VDC output, then the maximum output will be linearized between 0VDC (with no zero offset) and 5VDC at full scale. If the output signal range is specified to be 16 for a 4-20mA output, then the maximum output will be linearized between 4mA (with no zero offset) and 16mA at full scale.

P176: Default Output in Setup Mode

Selects the analog output signal level when entering the setup mode. When you enter the setup mode, the A/D conversion process and all weight calculations are suspended. Thus the analog output can no longer track the value of a weight-based parameter.

Select **Max** to generate the maximum analog output signal while in the setup mode (10V or 20mA).

Select **Min** to generate the minimum analog output signal while in the setup mode (0V or 0mA or 4mA).

Select **Same** to keep the analog output at the same level it was at immediately prior to entering the setup mode.

P177: Output Signal Type

Selects the analog output signal type (0-10VDC, 0-20mA or 4-20mA).

COUNTING

The counting parameters provide configuration of the counting feature.

P179: Count Enable

Enables the counting feature and makes P124, P125 and P180 → P189 available for configuration. It also makes all of the counting operating parameters available as valid parameter selections.

P180: Auto Sample Enable

Enables the auto sample feature. When the auto sample feature is in effect, a quantity will be automatically calculated upon stability after adding the sample.

P181: Auto Enhance Enable

Enables the auto enhance feature. When the auto enhance feature is in effect, adding additional pieces will result in a recalculation of the APW upon stability, providing the number of pieces added did not exceed the accuracy requirement. The APW is continually enhanced based on a larger sample size without the need for counting additional pieces.

P182: Default Sample Size

Sets the default sample size.

P183: Required Accuracy

Selects the accuracy requirement for sampling and auto enhancement.

P184: Accuracy Display Enable

Enables the accuracy display. When enabled, the calculated accuracy of the current sample will be shown on the display.

P185: Pre-Sample Scale

Selects the scale# to be automatically selected every time a sample routine is initiated.

P186: After-Sample Scale

Selects the scale# to be automatically selected after a sample routine is completed.

P187: Sample Filter

Selects the filter setting to be used when performing a sample operation (separate from P116).

P188: Enforce Sample Accuracy

Enables sample accuracy enforcement. If the required accuracy specified at P183 is not achieved during a sample routine, the sample will not be accepted. More pieces will be required to complete the sample routine.

P189: Sample Motion Divisions

Selects the number of motion divisions to be used during a sampling routine (separate from P111) to a resolution of 0.1 divisions. For example, an entry of **15** will be accepted as ±1.5 divisions of motion.

COMMUNICATION PORTS

The communication port parameters provide comm port configuration.

P199: Serial Port Instance

Sets the communication port# in effect when accessing the remainder of the serial port configuration parameters P200 → P211.

P200 – P204: Protocol

Selects the comm port's communication protocol. The port's protocol must match that of the connected device.

P205: Receive Mode

Selects the comm port's receive mode.

If the port is **disabled**, all received data is ignored

If the port is set to **standard** receive, all receive characters are processed normally.

If the port is set to **interpret**, all received characters are screened through the input interpreter (P217 → P224) before being used or discarded.

If the port is set for **Modbus**, all received characters are processed as Modbus protocol. Enabling Modbus at P205 makes P209 → P211 available for configuration.

If the port is set for selection 4 → 250, a received character will invoke macro 4 → 250 respectively.

P206: Transmit Mode

Selects whether or not a transmission will be delayed when the transmit buffer becomes full.

If set to **delay**, the transmission will be put on hold until the transmit buffer empties to the point where handshaking is asserted.

If set to **abort**, the transmission will be immediately aborted once the transmit buffer becomes full.

P207 – P208: Transit / Receive Buffer Size

Sets the size of the transmit and receive buffers at P207 and P208 respectively.

P209: Modbus Address

Selects the Modbus address. P205 must be set for **Modbus** for this parameter to be available.

P210: Modbus Mode

Selects Modbus ASCII or RTU mode. P205 must be set for **Modbus** for this parameter to be available.

P211: Modbus Word

Selects the Modbus word format. **HiLo** will transmit a Modbus high byte followed by the low byte, **LoHi** will reverse the order.

INPUT INTERPRETER

The input interpreter parameters provide configuration of each communication port's input interpreter. P205 of the communication port setup must be set to "*interpret*" or have been over-ridden by the %H macro command for P217 → P224 to take effect.

Refer to 8-48 for a more complete definition of the input interpreter configuration.

P217: NULL Character Enable

Enables the use of a null character in an input string. If enabled, any null character included in an input will be converted to a US control code (0x1F).

P218: Receive Termination Character

Selects the receive termination character for all line type input interpreters. Received data will not be interpreted until the termination character is received.

P219: Input Interpreter Instance

Sets the interpreter# in effect when accessing the remainder of the input interpreter configuration parameters P220 → P224.

P220: Interpreter Name

Assigns a name to the input interpreter for documentation purposes.

P221: Interpreter Type

Selects whether the interpreter is a **character** type or **line** type.

P222: Line Interpreter Entry Table

Begins the input interpreter table for line-type interpreters.

P223: Interpreter Character

Assigns the interpreter character for character-type interpreters.

P224: Interpreter Macro#

Assigns the macro to be invoked upon receiving a valid interpreter character/string.

NUMERIC PARAMETER FORMATTING

The numeric parameter formatting parameters set the data format for transmitting numeric data.

P240: Minimum Transmit Width

Assigns the minimum number of character to send when transmitting weight data. If the number of digits that make up the weight is less than the minimum transmit width, the number is left-padded with spaces to make up the difference.

P241: Sign Justification

Selects whether the polarity sign will appear right justified (to the immediate left of the most significant digit) or left justified (in the left-most position of the data field) when transmitted.

For example, assuming a minimum transmit width of 8 specified at P240, the number –10.25 would be transmitted as

```
  -10.25
```

with left justification, and

```
-  10.25
```

with right justification.

Note that with right justification, the left-most space or padded zero (0) will be replaced with the polarity sign. Thus if you wish to maintain space for eight digits in the data field, you must specify a minimum transmit width of nine (9).

NETWORKING & REMOTE COMMUNICATIONS

The networking parameters configure an indicator's network address. Remote communication parameters configure communication to a remote display, other indicators used in the remote display mode and other devices requesting display information.

P250: Network Enable

Enables network addressing and makes P251 available for configuration provided P205 is not set for Modbus.

P251: Network Address

Assigns the network address for the indicator. Note that the network address is assigned as a single ASCII byte value. An entry of '4' does not refer to the actual number four (4), rather it is and ASCII '4' or an <EOT> control code. If you want the indicator's address to be the number four (4), you would have to enter the address as 52 (the ASCII value of 4). Refer to Appendix B for a list of ASCII character values.

P290: Echo Display

Selects the communication port# to be used to echo the displayed data to a remote indicator or other serial device. Once a port is chosen, display data will begin echoing immediately without requiring you to save changes and exit the setup mode.

P291: Echo Start Character

Assigns the start character that signifies the beginning of echoed display data.

P292: Echo End Character

Assigns the end character that signifies the ending of echoed display data.

P293: Remote Display Enable

Selects the type of remote display connected to the optional remote display serial bus module.

P294: Remote Display Backlight Enable

Enables the LCD remote display backlight.

WEIGH MODE PARAMETERS

The weigh mode parameters assign the order in which operating parameters are selected for viewing in the weigh mode via the **[SELECT]** key.

P300 – P309: Mode Selections

Assigns the operating mode selectable via the **[SELECT]** key from the weigh mode. The parameter assigned at P300 will become the power-up mode for the scale.

ACCESS CODES

The access code parameters assign alternate user-defined access codes for entering the setup mode and calibration mode. See page 3-5 for complete details in using PIN numbers.

P400: Personal Identification Number

Assigns a custom PIN number as the setup mode access code.

P401: Quick Calibration Access Number

Assigns a custom PIN number as the quick calibration access code.

P402: Limited Access Number

Assigns a custom PIN number as the setup mode limited access code.

OIML

The OIML parameters are used to configure the indicator for alternate language character sets and for compliance with OIML regulations. See page 6-2 for complete details on OIML configuration.

P410: OIML Enable

Enables OIML operation.

P411: Language Character Set

Selects the language for the ASCII character set. Selecting alternate languages will result in various display character substitutions as shown in Table 6-4 on page 6-6.

P412: Preset Enable

Enables the preset status identifier for manually entered tare and accumulation values.

VFD / LCD DISPLAY SETUP

The VFD/LCD display setup parameters define the operation of the standard displays.

P420: Standard VF Display Mode

Selects the operation of the standard 7-segment VF display and control the backlighting of the LCD.

If set to **ON**, the VFD will always remain on with the intensity set at P423. The LCD backlight will also remain on provided the backlight is enabled at P294.

If set to **OFF**, the VFD will always remain off with the intensity set at P424. The LCD backlight will also remain off regardless of the backlight setting at P294.

If set to **AUTO**, the VFD will remain on with the intensity set at P423 as long as the scale does not become stable within the number of divisions set at P421 for the length of time specified at P422. This condition will also maintain the LCD backlight. Once stability is achieved within the parameters of P421 and P422, the VFD will turn off with the intensity set at P424. In the case of the LCD, the backlight will turn off. This serves as a power saving feature when the indicator is being powered by battery. Use of this feature may also extend the life of the VFD.

P421: Weight Threshold Divisions

Selects the number of divisions required to yield a stable condition for P420.

P422: Time-out

Selects the time-out period required to yield a stable condition for P420.

P423 – P424 Display Brightness (7-segment VFD only)

Selects the degree of VFD brightness when the display is on (P423) and off (P424).

P425: 4X20 VF / LCD Display Enable

Enables the 4X20 VF display for displaying the auto-update information normally sent to the 7-segment VFD. This parameter is normally enabled only for the model 661 where the 7-segment display is not present and there is no other means of displaying the auto-update information.

When an LCD display is installed, it is auto-detected by hardware. This will result in P425 being automatically set to **LCD Always**. It will not be possible to over-ride this setting as long as the LCD is installed.

P430: LCD Contrast (LCD only)

Selects the power-up contrast setting for the LCD display. Pressing **[CLR]** will restore the default contrast setting.

The LCD contrast can also be set at power-up without accessing the setup mode. Refer to the section on LCD Contrast Adjustment on page 2-27.

NTEP

The NTEP parameter is used to aid in ensuring compliance with NTEP regulations. See page 6-8 for complete details on NTEP configuration.

P440: NTEP Enable

Enables NTEP operation.

KEYPAD

The keypad parameters are used to assign the type of keypad in use and sets the keypress characteristics.

P450: Keypad Selection

Selects the type of keypad (number of keys) installed.

The keypad selection can also be set at power-up without accessing the setup mode. Refer to the section on Keypad Configuration on page 2-28.

P451: Keypad Repeat Rate

Selects the speed at which a held key will repeat its keypress.

P460: Beeper Volume

Selects the volume of the beeper when a key is pressed.

TIME & DATE

The time & date parameters are used to set and format the time and date.

P500: Time

Assigns the current time. Time must be entered in 24-hour format using the form **hh.mm.ss** or **hh:mm:ss** (seconds are optional).

P501: Date

Assigns the current date. If P504 is set for USA style, the date is entered using the form **mm.dd.yy** or **mm/dd/yy**. If P504 is set for International style, the date is entered using the form **dd.mm.yy** or **dd/mm/yy**.

P502: Time / Date Access

Enables the time & date accessibility so they can be changed from the weigh mode when the time/date parameter is selected.

P503: AM / PM Time Format

Selects 12-hour or 24-hour format for displaying and transmitting time & date values.

P504: Date Format

Selects USA (mm/dd/yy) or international (dd.mm.yy) format when displaying and transmitting date values.

DSD CONFIGURATION

DSD parameters are used to configure the Data Storage Device (DSD) feature. Refer to page 6-13 for a complete description of DSD operation.

P590: DSD Enable

Enables the Data Storage Device feature (DSD) and provides access to the other DSD parameters (P591 → P595)

Enabling DSD will override P806 to redefine the **[ID]** key to invoke the DSD Menu. It will also redefine P205 as the DSD receive mode for the specified DSD communication port.

⚠ When enabling or disabling DSD, you will be prompted to clear the DSD database records before the change is allowed. Be sure to download any stored data before proceeding.

P591: DSD Serial Port

Selects the communication port to be used for DSD transmissions.

The usage of the DSD port selected at P591 can be temporarily overridden by usage of the %H macro command. If the selection is turned off, then no DSD transmits or receives will occur. If the port number is changed, then the new port will be used.

Nothing prevents other indicator transmissions from being sent over the DSD port. No other processing of received data will occur on this port.

Note that if a comm port selected is programmed as receive disabled, selecting it does not turn the port on, no data will be received.

P592: DSD Receive Character

Specifies a single character used to create a row in the DSD database when received on the DSD communication port.

P593: DSD Custom Transmit

Specifies a custom transmit used to automatically transmit DSD data after a row has been created in the DSD database. The custom transmit specified will not allow non-DSD parameter entries. Transmission will be motion delayed by virtue of the stored data row. The custom transmit communication port can be specified at P991.

P594: DSD Maximum Number of Rows

Specifies the maximum number of DSD data rows that can be stored in the database. An attempt to store a record in a full database will result in a 1 second OVER-WRITE warning message indicating that the oldest record will be deleted before storing the new data row.

P595: DSD Number of Warning Rows

Specifies the number of unused rows at which point a 1 second warning message will be displayed. For example, if the maximum number of rows is 1000 and the number of warning rows is 100, then a warning message will be displayed for every data row stored after the 900th record. The maximum number of warning rows is 999.

PARAMETER RENAMING

The parameter renaming parameters provide the ability to rename various operating parameters.

P600 – P646: Rename Parameters

Assigns alternate names to operating parameters. Assigned names are displayed and transmitted in place of the default names. This feature is useful when configuring the indicator for foreign languages.

Note that counting must be enabled at P179 in order for the counting rename parameters to become available.

TOTAL & TARE SAVE

The total & tare save parameters are used to enable the tare and accumulation parameter values to be saved in non-volatile memory and restored in the even of power loss.

P660 – P661: Total / Tare Save Method

Selects whether accumulation values and tare values will be written to the EEPROM each time the values change so that the values may be retained and restored in the event of a power loss.

If set to **AUTO**, the values are automatically stored to the EEPROM when changed.

If set to **No Save**, the values are not written to the EEPROM and will be lost during a power failure.

If set to **On Request**, the values can only be written to the EEPROM via the %v macro command.

VARIABLES

The variable parameters allow configuration of variables.

P680: Number of Variables

Assigns the number of variable registers to be dynamically allocated. Once allocated, the number of variables may be increased or decreased as necessary (memory permitting).

P681: Variable Instance

Sets the variable# in effect when accessing the remainder of the variable configuration parameters P682 → P689.

P682: Variable Name

Assigns a name to the variable for documentation and display purposes.

To display the name of a string variable or scale-specific float variable, the name must be 5 characters or less. If the name is greater than 5 characters, the default variable name (V#xxx) will be displayed. However, the entire given name will still be transmitted.

To display the name of a all other variables, the name must be 10 characters or less. If the name is greater than 10 characters, the default variable name (V#xxx) will be displayed. However, the entire given name will still be transmitted.

P684: Save Method

Selects whether variable values will be written to the EEPROM each time the values change so that the values may be retained and restored in the event of a power loss.

If set to **AUTO**, the values are automatically stored to the EEPROM when changed.

If set to **No Save**, the values are not written to the EEPROM and will be lost during a power failure.

If set to **On Request**, the values can only be written to the EEPROM via the %v macro command.

P685: Variable Lock

Enables a variable to be locked so that its value cannot be changed manually through the front panel keypad.

P686: Variable Type

Selects the variable type; float, integer, unsigned integer or string.

P687: Float Style

Selects the style of a float-type variable in terms of the number of decimal places. Selecting a scale specific float style ties the number of decimal places and displayed division size of the float value to that of the specified scale. Scale specific float values can also be viewed in alternate units of measure by pressing the **[UNITS]** key when displayed.

P686 must be set for “Float” for P687 to be available for selection.

P688: Integer Style

Selects the style of an integer-type variable. Integers and unsigned integers may be represented as either a whole number, a time, a date, or a time & date.

P686 must be set for “Int” or “U-Int” for P688 to be available for selection.

P689: String Size

Assigns the maximum string size allowed for the string variable.

P686 must be set for “Strng” for P689 to be available for selection.

DATABASE

The database parameters provide configuration of database structure.

P699: Database Instance

Sets the database# in effect when accessing the remainder of the database configuration parameters P700 → P799.

P700: Database Name

Assigns a name to the database for documentation purposes.

P701 – P798: Column Parameter

Assigns an operating parameter to each column of the database. P701 refers to column 1, P702 refers to column 2 and so on.

KEYPAD KEY ASSIGNMENTS

The key assignments parameters allows individual keys to be redefined to invoke macros.

P800 – P820: Key Assignments

Assigns a front panel key to invoke a macro. Doing so disables the key's normal function. To completely disable an individual key, assign the key to an undefined macro number.

PROGRAMMABLE DIGITAL INPUT / OUTPUT

The programmable digital I/O parameters provide configuration for each of the eight (8) PDIO channels (660 Series only). Refer to chapter 12 for a complete description of PDIO functions.

P850: Channel Instance

Sets the channel# in effect when accessing the remainder of the PDIO configuration parameters P851 → P864.

P851: Channel Function

Selects the PDIO channel's function.

P852: First I/O Parameter Name

Assigns a name to the PIOA parameter for documentation and display purposes.

The name must be 10 characters or less if it is to be displayed. If the name is greater than 10 characters, the default name (PIOAx) will be displayed. However, the entire given name will still be transmitted.

P853: Second I/O Parameter Name

Assigns a name to the PIOB parameter for documentation and display purposes.

The name must be 10 characters or less if it is to be displayed. If the name is greater than 10 characters, the default name (PIOBx) will be displayed. However, the entire given name will still be transmitted.

P854: Third I/O Parameter Name

Assigns a name to the PIOC parameter for documentation and display purposes.

The name must be 10 characters or less if it is to be displayed. If the name is greater than 10 characters, the default name (PIOCx) will be displayed. However, the entire given name will still be transmitted.

P856: Maximum Expected Frequency

Assigns the maximum expected frequency when P851 is set for frequency debounce.

P857: Clock Source

Selects **clock source A** (65,536 Hz) or **clock source B** (524,288 Hz). A higher clock speed will provide more accuracy for some functions but could also shorten the duration of the measurement period for others.

P858: Pulse Edge Detection

Selects either a **rising** or **falling** edge as the reference for determining pulse count and frequency.

P859: Pulse Measurement Period

Assigns the time period in milliseconds for determining the frequency when P851 is set for Frequency Input A.

P860: Number of Pulses to Measure

Assigns the number of pulses to measure for determining the frequency when P851 is set for Frequency Input B.

P861: Phase Measurement Type

Selects whether to monitor **Low** time or **High** time for determining the duty cycle when P851 is set for Phase Time.

P862: Pulse Type

Permanently set for **sink** when P851 is set for Delay Output.

P863: Pulse Scaling Factor

Assigns a variable to be used as the pulse input scaling factor. If a variable is not assigned to P863, the scaling factor will be 1:1 (each pulse will be counted and displayed).

If a variable is assigned, then the variable's value is used as a conversion factor for displaying received pulses. For example, a scaling factor of 0.1 will display one pulse for every 10 pulses received.

P864: Number of Decimal Places

Selects the number of decimal places to use when displaying a pulse count with a scaling factor.

CUSTOM TRANSMIT

The custom transmit parameters provide configuration of the custom transmit tables and the criteria for transmission.

P980: Continuous Transmit Rate

Assigns the interval to a resolution of 0.1 seconds at which continuous transmits are transmitted.

P989: Custom Transmit Instance

Sets the custom transmit# in effect when accessing the remainder of the custom transmit configuration parameters P990 → P4999.

P990: Transmit Name

Assigns a name to the custom transmit for documentation purposes.

P991: Transmit Mode

Selects the method of initiating a custom transmit.

If set to **Off**, the **[PRINT]** key or %p macro command cannot be used to initiate the transmission. Only the %Q macro command can initiate a custom transmit set to **Off**.

If set to **On Request**, the **[PRINT]** key can be used to initiate the custom transmit. If multiple custom transmits are set to **On Request**, all of those transmits will be transmitted in sequential order with one press of the **[PRINT]** key.

If any custom transmits are set to **Prompt**, the **[PRINT]** key will invoke the prompt "*Which Tx# ?*". The operator can then key in the desired transmit number and press **[ENTER]** to send only that transmit. Any custom transmits set for **On Request** will immediately follow the prompted transmit in sequential order. The selection for **Prompt** is not available on the 460.

P992: Serial Port

Selects the communication port that the custom transmit will be sent out.

P993: Current Scale Motion

Selects whether the custom transmit will be motion delayed based on motion of the currently selected scale.

P994: Scale Motion

Selects whether the custom transmit will be motion delayed based on motion of any combination of enabled scales. To specify a scale for motion delay, include that scale number in the entry at P994. For example, to specify motion delay for scales 1, 2, and 3, key in

123 [ENTER]

The display shows the scales in sorted order after pressing **[ENTER]**. If more than 5 scales are specified, the display would show the first 4 scales and an inverse down arrow. Pressing the down-arrow key will scroll the remaining specified scale numbers onto the display, one at a time. Continued pressing or the down arrow will cause the first 4 scales to be shown. If all scales were specified, then the bottom line will display "All". If zero (0) was entered, the bottom line will show "None!".

Requiring motion delay on multiple scales is useful for applications such as multiple-axle truck scales where all scales must become stable before printing a ticket.

P995 – P997: Scale Motion (Model 650 Compatibility)

Provided only for backward compatibility with the GSE 650 Series upload files. Selections entered here for scale's 2, 3 and 4 will be reflected in P994. For example, enabling motion delay for scale #2 at P995 (by entering a value of '1') will result in scale #2 appearing at P994.

P998: Continuous Transmit Enable

Enables the custom transmit for continuous transmission. The continuous transmit interval is specified at P980 or through use of the %I macro command.

P999: Transmit Table Limited Access

Enables the contents of the custom transmit table to be blocked from viewing, downloading or editing if only the limited access code specified at P402 was used to enter the setup mode.

P1000: Transmit Entry Table

Begins the custom transmit data table.

SETPOINTS

The setpoint parameters provide configuration of physical and logical input/output.

P5099: Setpoint Instance

Sets the setpoint# in effect when accessing the remainder of the setpoint configuration parameters P5100 → P5150.

P5100: Setpoint Mode

Selects whether the setpoint is configured as an **Output**, **Input** or **Disabled** setpoint. The mode selected will determine which of the subsequent setpoint setup parameters will be available for configuration.

P5101: Setpoint Name

Assigns a name to the setpoint for documentation purposes.

P5110: Activation Condition

Selects the setpoint's activation criteria.

P5111: Activation Delay

Assigns an activation delay (in seconds) before the setpoint will become active once the activation criteria is met.

P5112: Activation Macro#

Selects the macro to be invoked when the setpoint becomes activated.

P5113: Activation Motion

Selects whether the activation of the setpoint will be motion **delayed** or **ignored**.

P5114: Lower Activation Parameter

Assigns the operating parameter, usually a variable with a pre-assigned value, to be used as the activation threshold. The setpoint will activate when the value of the parameter assigned at P5114 is equal to or greater than the value of the compare parameter assigned at P5150.

If the setpoint is set for **Outside** or **Between** at P5110, P5114 serves as the lower limit value for the activation window.

P5115: Upper Activation Parameter

Assigns the operating parameter, usually a variable with a pre-assigned value, to be used as the upper limit value for the activation window when the setpoint is set for **Outside** or **Between** at P5110.

P5130: Deactivation Condition

Selects the setpoint's deactivation criteria.

P5131: Deactivation Delay

Assigns an deactivation delay (in seconds) before the setpoint will become active once the activation criteria is met.

P5132: Deactivation Macro#

Selects the macro to be invoked when the setpoint becomes deactivated.

P5133: Deactivation Motion

Selects whether the deactivation of the setpoint will be motion **delayed** or **ignored**.

P5134: Lower deactivation Parameter

Assigns the operating parameter, usually a variable with a pre-assigned value, to be used as the deactivation threshold. The setpoint will deactivate when the value of the parameter assigned at P5134 falls below the value of the compare parameter assigned at P5150.

If the setpoint is set for **Outside** or **Between** at P5130, P5134 serves as the lower limit value for the deactivation window.

P5135: Upper deactivation Parameter

Assigns the operating parameter, usually a variable with a pre-assigned value, to be used as the upper limit value for the deactivation window when the setpoint is set for **Outside** or **Between** at P5130.

P5150: Compare Parameter

Assigns the operating parameter to be monitored for activation and deactivation criteria.

MODBUS PARAMETER MAP

The Modbus parameter map provides mapping for the memory locations of operating parameters to be used with Modbus reading and writing register functions.

P6001: Modbus Address Translation Table

Begins the Modbus address translation table. An operating parameter is assigned to each address. The first one is assigned at P6001, the second at P6002 and so on.

MACROS

The macro parameters provides configuration for each macro table.

P9980: Abort Macro#

Assigns the macro# to be invoked when another macro is aborted via the **[CLR] + [SELECT]** key combination or the serial macro abort character <248>.

P9981: Macro Abort Method

Specifies whether or not the abort menu will be invoked when a macro is aborted via the **[CLR] + [SELECT]** key combination or the serial macro abort character <248>.

If set to **Menu**, the abort menu will be displayed awaiting operator input to continue.

If set to **Immediate**, the abort menu will not appear. The macro will be immediately aborted and the abort macro# assigned at P9980 will be executed.

P9990: Macro Instance

Sets the macro# in effect when accessing the remainder of the macro configuration parameters P9991 → P19999.

P9991: Macro Name

Assigns a name to the macro for documentation purposes. A named macro can also be used as a method of invoking a macro via the macro menu feature.

P9992: Macro Priority

Selects the priority of the macro when invoked.

If set for **Standard**, the macro will be pushed onto the macro stack if another macro is running.

If set for **Immediate**, any macro presently running will be interrupted so that the interrupt macro can begin immediate execution. The interrupted macro will resume execution upon completion of the interrupt macro.

Refer to *Interrupt Macros* on page 9-161 for complete details.

P9993: Macro Menu Enable

Enables the named macro (P9991) to be invoked by selecting it via the macro menu.

P9994: Macro Table Limited Access

Enables the contents of the macro table to be blocked from viewing, downloading or editing if only the limited access code specified at P402 was used to enter the setup mode.

P10001: Macro Entry Table

Begins the macro program table.

MACRO DEBUG

The macro debug parameters provide access to the macro debug table for program diagnostics.

P50000: Macro Debug Table Limited Access

Enables the contents of the macro debug table to be blocked from viewing, downloading or editing if only the limited access code specified at P402 was used to enter the setup mode.

P50001: Macro Debug Table

Begins the macro debug diagnostic table. Press **[PRINT]** and select a comm port to download the macro debug to a printer or PC for analysis.

Pressing **[ID]** (or **[PRINT] + [SELECT]** on the 460) at P50001 will briefly display the total number of bytes allocated for the macro trace buffer (Bsize) followed by the number of bytes used (Bused). If no macros are programmed both numbers will be zero (0).

SCRIPT FILES

Script files are pre-programmed setup files residing within the firmware that can be loaded to perform various applications. Script file parameters, macros, custom transmits, etc. are loaded into the setup parameters directly from the firmware by defaulting the indicator to a script file selection.

LOADING SCRIPT FILES

To load a script file:

1. Access the setup mode using the setup access code.
2. Select the Default All parameter (P65001).
3. Press **[TARE]** or **[UNITS]** on the 465, 560 and 660 Series or **[ZERO]** + **[PRINT]** or **[UNITS]** on the 460 to scroll forward and backward through the available script files.
4. When the desired script file is displayed, press **[ENTER]** to initiate the default process.
5. When the default is complete, exit and save changes to begin using the script file software.

CONFIGURING REMOTE KEYS (460 SERIES)

Remote keys are configured in the same manner as script file. To configure a remote key, access P65003 and follow the same procedure used to load a script file.

Chapter 4 CALIBRATION

In addition to general calibration information, this chapter provides information on the New Zero, Last Zero, Temp Zero, Only Zero, Cal Reset, and Known LCOut calibration methods. You will also learn about multi-scale calibration, calibration units, and A/D calibration.

OVERVIEW

CALIBRATION

The weight indicator to load cell(s) is calibrated by establishing zero (no load) and span (known test load) reference points. There are two ways to access the calibration menu, directly from the weight mode using the Quick Calibration procedure, or upon exiting the setup mode.

QUICK CALIBRATION

Once all setup parameters are configured, the Quick Cal procedure is the easiest way to access the calibration routine. Typically used for re-calibration, Quick Cal can be accessed directly from the weigh mode without entering the setup mode.

QUICK CAL ACCESS

Example:
Accessing Quick Calibration Mode

[1] [0] [0] [SELECT]

SetUp Keyin Code:

[5] [4] [3] [2] [1] [ID] [ENTER YES]

SetUp Quick Cal!

)00 New Zero?

An access code is required to enter the Quick Cal. From the weigh mode, key in **100 [SELECT]** to access the Keyin Code: display. Key in the Quick Cal access code routine (see example: *Accessing Quick Calibration Mode*). The default Quick Cal access code is **54321 [ID] [ENTER]**.

QUICK CAL ACCESS (460)

The GSE default Quick Calibration access code for the 460 is

[ZERO] + [SELECT] [ZERO] [PRINT] [UNITS] [TARE]

Changing the Quick Cal Access Code

The default Quick Cal access code can be changed at P401 of the setup mode. Access the setup mode (see *Accessing The Parameter Setup Mode* on page 3-3), then key in **401 [SELECT]** to access the Quick Calibration Access Number parameter. If P401 displays QCAL None!, then the default access code is in effect. Otherwise the custom code will be displayed. To enter a new Quick Cal access code, key in the new code and press **[ENTER]**. To restore the default Quick Cal access code, press **[CLR]**.

The Quick Cal access code consists of up to five alpha-numeric characters. Alpha characters may be entered through the front panel (see Character Entry on page 5-6). Alpha characters should not be included in the access code unless an alpha keyboard is used to access the calibration routine. Although it is possible to scroll in the code from the front keypad, characters are not viewable during entry.

CALIBRATION UPON EXITING SETUP MODE

Example:
Calibration When Exiting Setup

[ZERO]

SETUP CAL ?

SETUP ENTER =CAL!

[ENTER YES]

)05 New Zero?

You should calibrate the scale system after making changes to the setup parameters, especially after setting the capacity and division size.

Every time you exit the setup mode, the prompt ENTER=CAL! is displayed. Press **[ENTER]** at this prompt to access the calibration routine (see example: *Calibration When Exiting Setup*). Changes to parameters and calibration are saved upon exiting the calibration routine.

Example:
Selecting a Calibration Method

)05 New Zero?

SELECT

)05 Last Zero?

SELECT

)05 Temp Zero?

SELECT

)05 Only Zero?

SELECT

)05 Cal Reset

SELECT

)05 Known LCOut

CALIBRATION METHODS

There are six methods of calibration. Press **[SELECT]** to select a calibration method (see example: *Selecting a Calibration Method*). Press **[ENTER]** to begin the calibration method selected. Refer to the appropriate section for calibration instructions.

- **New Zero** - Establishes a new zero (no load) and span (test load)
- **Last Zero** - Performs a span re-calibration without removing the test load. *(This selection is not available with linearization enabled.)*
- **Temp Zero** - Performs a calibration without removing the current gross weight. The zero reference determined during the last calibration is maintained. *(This selection is not available with linearization enabled.)*
- **Only Zero** - Establishes a new zero reference without affecting span.
- **Cal Reset** - Adjusts the zero and gain factors of the A/D amplifier to default values for maximum sensitivity.
- **Known LCOut** - Calibrates without the use of test weights. The mV/V value and full scale capacity of each load cell must be known.

GENERAL NOTES ON CALIBRATION

- Pressing **[CLR]** at any point in the calibration routine moves back one
- Pressing **[CLR]** at the New Zero? prompt exits calibration mode.
- A calibration weight can be applied before or after entering the calibration weight value. The display prompts you to Keyin CalWt (key in calibration weight) or Add CalWT (add calibration weight) at the appropriate time.
- The digital filter is automatically set to 4 seconds during calibration.
- A motion delay is enforced during zero and span calibration.
- New calibration values are not permanently saved until the calibration mode is exited and changes are saved by pressing **[ENTER]** at the ENTER=SAVE prompt. If power is lost during calibration, the previously saved calibration values will be in effect when power is restored.

Example:
New Zero Calibration

`)05 lb Gross`

[1] [0] [0] [SELECT]

`SETUP Keyin Code:`

[5] [4] [3] [2] [1] [ID] [ENTER YES]

`SETUP Quick Cal!`

`)05 New Zero?`

[ENTER YES]

`)05 Mot'n Delay`

`)05 Units =lb`

`)00 Keyin CalWt`

ADD TEST WEIGHT

[1] [0] [0] [ENTER YES]

`10)00 Mot'n Delay`

`10)00 CAL OK ?`

[ENTER YES]

`SETUP Save Mods?`

`SETUP ENTER =SAVE`

[ENTER YES]

`SETUP Exit Setup`

`SETUP ENTER =EXIT`

[ENTER YES]

`10)00 lb Gross`

NEW ZERO

The most common calibration procedure, **New Zero** establishes a new zero (no load) and span (test load) calibration reference. Use this method for first-time calibration and complete re-calibration.

To perform a New Zero calibration:

1. Remove all weight from the scale.
2. Access the calibration mode as described on page 4-2.
3. Select the New Zero calibration method as described in *Calibration Methods* on page 4-3.
4. Press **[ENTER]** at the `New Zero?` prompt to establish the new zero reference.
5. After establishing the zero reference, the default calibration units are displayed momentarily followed by the `Keyin CalWt` prompt.

   Apply the calibration weight, key in the calibration weight value in terms of the default calibration units and press **[ENTER]** to establish span.

⚠ If the calibration weight value was entered before the weight was applied, the display will prompt `Add CalWT`. Add the calibration weight and press **[ENTER]**.

6. After establishing span, `CAL OK?` is displayed suggesting that the calibration is acceptable, or `ReCal???` is displayed suggesting that the calibration procedure should be repeated.

   Accept the calibration by pressing **[ENTER]** at the `CAL OK?` prompt or **[CLR]** at the `ReCal???` prompt.

   - or -

   Repeat the calibration by pressing **[CLR]** at the `CAL OK?` prompt or **[ENTER]** at the `ReCal???` prompt.

8. Once the calibration is accepted in step 6, press **[ENTER]** at the `ENTER=SAVE` prompt and again at the `ENTER=EXIT` prompt to save the new calibration and exit the calibration mode.

   - or -

   To exit the calibration mode without saving the new calibration, press **[CLR]** at the `ENTER=SAVE` prompt. Then press **[ENTER]** at the `ENTER=UNDO` prompt and again at the `ENTER=EXIT` prompt to exit the calibration mode.

⚠ If you choose to "undo" the calibration when exiting the setup mode, you will also undo any unsaved changes made to the setup parameters.

Example:
Last Zero Calibration

`2) lb Gross`

[ZERO]

`0) lb Gross`

ADD 10,000 LB TEST WEIGHT

`1004) lb Gross`

[1] [0] [0] [SELECT]

`SETUP Keyin Code:`

[5] [4] [3] [2] [1] [ID] [ENTER YES]

`SETUP Quick Cal!`

`1004) New Zero?`

[SELECT]

`1004) Last Zero?`

[ENTER YES]

`1004) Units =lb`

`1004) Keyin CalWt`

[1] [0] [0] [0] [0] [ENTER YES]

`1000) CAL OK ?`

[ENTER YES]

`SETUP Save Mods?`

`SETUP ENTER =SAVE`

[ENTER YES]

`SETUP Exit Setup`

`SETUP ENTER =EXIT`

[ENTER YES]

LAST ZERO

Last Zero allows span re-calibration without removing the applied test weight. The last zero established by pressing **[ZERO]** from the weigh mode will be used as the zero reference. This procedure is especially useful when performing routine tolerance checks on large capacity scales. A scale found to be out-of-tolerance can be easily calibrated without having to remove the test weights to reestablish a zero reference.

To perform a Last Zero calibration:

1. Remove all weight from the scale.
2. Press **[ZERO]** to zero the scale in the weigh mode.
3. Apply the calibration test weight.
4. Access the calibration mode as described on page 4-2.
5. Select the Last Zero calibration method as described in *Calibration Methods* on page 4-3.
6. Press **[ENTER]** at the `Last Zero?` prompt to display the `Keyin CalWT` prompt.
7. Key in the calibration weight value in terms of the default calibration units and press **[ENTER]** to establish span.
8. After establishing span, `CAL OK?` is displayed suggesting that the calibration is acceptable, or `ReCal???` is displayed suggesting that the calibration procedure should be repeated.

   Accept the calibration by pressing **[ENTER]** at the `CAL OK?` prompt or **[CLR]** at the `ReCal???` prompt.

   - or -

   Repeat the calibration by pressing **[CLR]** at the `CAL OK?` prompt or **[ENTER]** at the `ReCal???` prompt.
9. Once the calibration is accepted in step 6, press **[ENTER]** at the `ENTER=SAVE` prompt and again at the `ENTER=EXIT` prompt to save the new calibration and exit the calibration mode.

   - or -

   To exit the calibration mode without saving the new calibration, press **[CLR]** at the `ENTER=SAVE` prompt. Then press **[ENTER]** at the `ENTER=UNDO` prompt and again at the `ENTER=EXIT` prompt to exit the calibration mode.

⚠ If you choose to "undo" the calibration when exiting the setup mode, you will also undo any unsaved changes made to the setup parameters.

TEMPORARY ZERO

Temp Zero is used to calibrate without establishing a new zero. Calibration can be performed without removing the currently applied gross load. A temporary zero is established so that test weights can be added during calibration. The original zero reference determined during the previous calibration is not affected. This procedure is commonly used to calibrate hopper scales where it is impractical to empty the product before calibrating.

To perform a Temp Zero calibration:

1. Access the calibration mode as described on page 4-2.
2. Select the Temp Zero calibration method as described in *Calibration Methods* on page 4-3.
3. Press **[ENTER]** at the Temp Zero? prompt to establish a temporary zero reference.
4. After establishing the temporary zero reference, the default calibration units are displayed momentarily followed by the Keyin CalWT prompt.
5. Apply the calibration weight, key in the calibration weight value in terms of the default calibration units and press **[ENTER]** to establish span.

   If the calibration weight value was entered before the weight was applied, the display will prompt Add CalWT. Add the calibration weight and press **[ENTER]**.
6. After establishing span, CAL OK? is displayed suggesting that the calibration is acceptable, or ReCal??? is displayed suggesting that the calibration procedure should be repeated.

   Accept the calibration by pressing **[ENTER]** at the CAL OK? prompt or **[CLR]** at the ReCal??? prompt.

   - or -

   Repeat the calibration by pressing **[CLR]** at the CAL OK? prompt or **[ENTER]** at the ReCal??? prompt.
7. Once the calibration is accepted in step 6, press **[ENTER]** at the ENTER=SAVE prompt and again at the ENTER=EXIT prompt to save the new calibration and exit the calibration mode.

   - or -

   To exit the calibration mode without saving the new calibration, press **[CLR]** at the ENTER=SAVE prompt. Then press **[ENTER]** at the ENTER=UNDO prompt and again at the ENTER=EXIT prompt to exit the calibration mode.

Example:
Temporary Zero Calibration

2105) lb Gross

1 0 0 SELECT

SETUP Keyin Code:

5 4 3 2 1 ID ENTER YES

SETUP Quick Cal!

2105) New Zero?

SELECT SELECT

2105) Temp Zero?

ENTER YES

2105) Mot'n Delay

2105) Units =lb

0) Keyin CalWt

ADD TEST WEIGHT

2 0 0 0 ENTER YES

201) Mot'n Delay

200) CAL OK ?

ENTER YES

SETUP Save Mods?

SETUP ENTER =SAVE

ENTER YES

If you choose to "undo" the calibration when exiting the setup mode, you will also undo any unsaved changes made to the setup parameters.

Example:
Only Zero Calibration

`9% lb Gross`

[1] [0] [0] [SELECT]

`SETUP Keyin Code:`

[5] [4] [3] [2] [1] [ID] [ENTER YES]

`SETUP Quick Cal!`

`9% Only Zero?`

[ENTER YES]

`9% Mot'n Delay`

`0) CAL OK ?`

[ENTER YES]

`SETUP Save Mods?`

`SETUP ENTER =SAVE`

[ENTER YES]

`SETUP Exit Setup`

`SETUP ENTER =EXIT`

[ENTER YES]

`0) lb Gross`

ONLY ZERO

Only Zero is used for zero calibration only. This calibration procedure is primarily used for the zero reference after changing a scale's dead-load, such as adding safety rails to a scale deck or installing a mixer motor on a hopper scale. Because the full scale capacity is referenced from the last zero calibration, performing a zero calibration helps to ensure that the full scale over-load will not occur prematurely due to the additional dead-load.

To perform an Only Zero calibration:

1. Remove all weight from the scale.
2. Access the calibration mode as described on page 4-2.
3. Select the Only Zero calibration method as described in *Calibration Methods* on page 4-3.
4. Press **[ENTER]** at the `Only Zero?` prompt to establish the new zero reference.
5. After establishing zero, `CAL OK?` is displayed suggesting that the calibration is acceptable.

   Accept the calibration by pressing **[ENTER]** at the `CAL OK?` prompt.

   - or -

   Repeat the calibration by pressing **[CLR]** at the `CAL OK?` prompt.
6. Once the calibration is accepted in step 5, press **[ENTER]** at the `ENTER=SAVE` prompt and again at the `ENTER=EXIT` prompt to save the new calibration and exit the calibration mode.

   - or -

   To exit the calibration mode *without* saving the new calibration, press **[CLR]** at the `ENTER=SAVE` prompt. Then press **[ENTER]** at the `ENTER=UNDO` prompt and again at the `ENTER=EXIT` prompt to exit the calibration mode.

⚠ If you choose to "undo" the calibration when exiting the setup mode, you will also undo any unsaved changes made to the setup parameters.

CALIBRATION RESET

Cal Reset sets the gain factors of the A/D amplifier to minimum values and clears the A/D's zero offset. These gain values are stored in the Information Parameters at P61104 → P61107 (see the *Calibration Parameters* section). A Cal Reset should be performed if calibration is not possible due to an over-load condition, or if the displayed weight value does not change when the test weight is applied.

If an over-load condition exists at the time of calibration, the calibration method prompts are replaced by an Over load! message. Press [CLR] to proceed directly to the Cal Reset procedure.

To perform a Calibration Reset:

1. Access the calibration mode as described on page 4-2.
2. Select the Cal Reset calibration method as described in *Calibration Methods* on page 4-3.
3. Press **[ENTER]** at the `Cal Reset` prompt reset the A/D amplifier.
4. The display prompts `New Zero?`. Proceed with calibration.

KNOWN LOADCELL OUTPUT

Known LCOut is used to calibrate without test weights. The exact full scale mV/V rating must be known for each load cell. All load cells must be of the same full scale capacity. This procedure works best for hopper scales where weight is evenly distributed and signal trimming is not required.

To perform a Known Loadcell Output calibration:

1. Access the calibration mode as described on page 4-2.
2. Select the Known LCOut calibration method as described in *Calibration Methods* on page 4-3.
3. Press **[ENTER]** at the `Known LCOut` prompt to display `#of LC`.

   The number of load cells specified during the last calibration will also be displayed. A value of zero (0) indicates that this calibration method has not yet been performed.
4. Key in the number of load cells (8 maximum) and press **[ENTER]**.

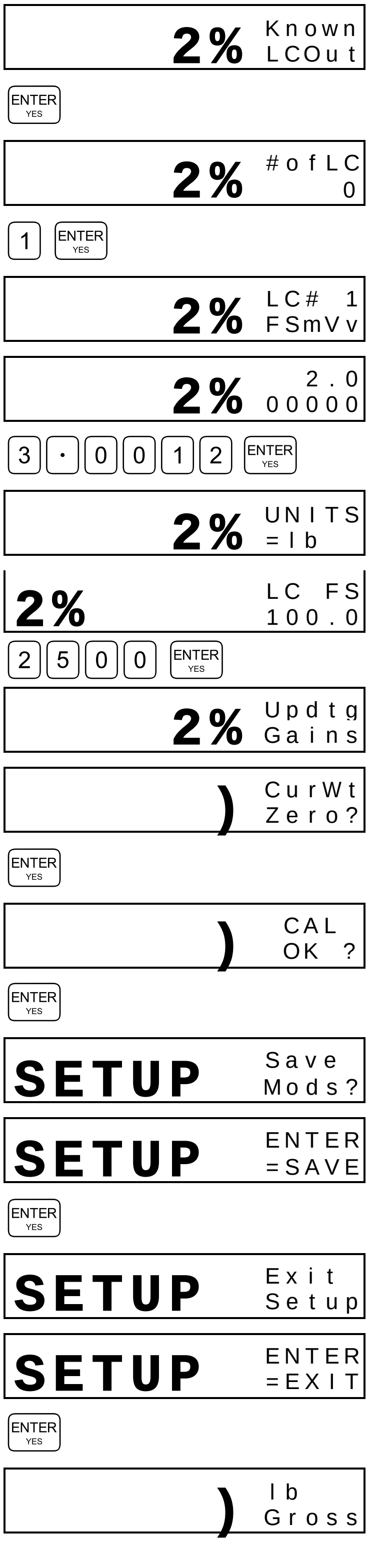
   - or -

   Press **[ENTER]** to accept the displayed value.
5. The display prompts `LC#x mVv` (where 'x' is the load cell number) and then shows the mV/V value (0.1 → 5.0) last entered for this load cell.
6. Key in the load cell's mV/V value and press **[ENTER]**.

   - or -

   Press **[ENTER]** to accept the displayed value.

Example:
Known Loadcell Calibration



7. Steps 5-6 will be repeated for as many load cells as specified in step 4.

8. The display prompts LC FS showing the value last entered for the load cell full scale.

9. Key in the full scale capacity for the load cell(s) and press **[ENTER]**.

   - or -

   Press **[ENTER]** to accept the displayed value.

10. The display briefly shows Updtg Gains as it updates the gain values, then prompts CurWt Zero?.

11. Press **[ENTER]** to establish the current input signal as the zero reference.

    - or -

    Press **[SELECT]** to display Zero=0mVv?. Press **[ENTER]** to use a 0mV/V output as the zero reference.

    - or -

    Press **[SELECT]** to display Keyin CurWt. Key in the known gross weight already applied to the scale and press **[ENTER]**.

    - or -

    Press **[CLR]** to bypass the zeroing option.

12. The display shows CAL OK? suggesting that the calibration is acceptable.

    Accept the calibration by pressing **[ENTER]** at the CAL OK? prompt.

    - or -

    Repeat the calibration by pressing **[CLR]** at the CAL OK? prompt.

13. Once the calibration is accepted in step5, press **[ENTER]** at the ENTER=SAVE prompt and again at the ENTER=EXIT prompt to save the new calibration and exit the calibration mode.

    - or -

    To exit the calibration mode without saving the new calibration, press **[CLR]** at the ENTER=SAVE prompt. Then press **[ENTER]** at the ENTER=UNDO prompt and again at the ENTER=EXIT prompt to exit the calibration mode.

MULTI - SCALE CALIBRATION

When more than one scale is enabled, the prompt `Keyin Scl#` appears before accessing the calibration method selections. Key in the scale number to be calibrated and press **[ENTER]**. Proceed with a calibration method as described in *Calibration Methods* on page 4-3.

After completing a calibration, the `Keyin Scl#` appears once again. Enter the next scale number to be calibrated, or press **[CLR]** to exit the calibration mode and save the new calibration data.

CALIBRATION UNITS

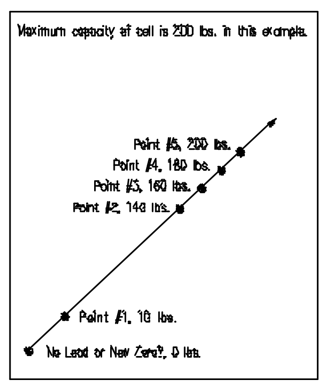
It is not possible to select lb/oz as calibration units.

It is expected that a calibration weight will be entered in terms of the selected calibration units as determined by P150 in the setup mode. The default calibration units are displayed briefly during calibration just prior to the `Keyin CalWT` prompt.

Alternate units may be used during calibration by pressing the **[UNITS]** key at the `Keyin CalWT` prompt. This will scroll through any units that have been assigned to P131 → P134 of the setup mode for the selected scale.

MULTI - POINT LINEARIZATION

Multi-point linearization provides up to five calibration points for each scale input. This allows you to compensate for load cell non-linearity. Multi-point linearization must be enabled in the setup mode at P119. Once enabled, you can define up to five points of linearization in during the calibration routine.



To perform Multi-Point Linearization:

1. Access the calibration mode as described on page 4-2.
2. Select a calibration method as described in *Calibration Methods* on page 4-3 and proceed with the calibration routine.
3. Enter the calibration weight value for the first linearization point when prompted to `Keyin CalWT`.
4. Once the first linearization point is established, `Keyin Pnt2` is displayed.

If there was a significant change in span for the first linearization point, ReCal??? is displayed suggesting that the calibration procedure must be repeated for the first point.

5. Enter the calibration weight value for the second linearization point.
6. Repeat this process for up to five linearization points. If fewer than five linearization points are specified, press **[ENTER]** without keying in the next weight value to end calibration.
7. After establishing all linearization points, CAL OK? is displayed suggesting that the calibration is acceptable.

Linearization calibration weights and calibration factors can be viewed at P61130 → P61139 in the information parameters.

Press **[ENTER]** at the CAL OK? prompt to accept the calibration and exit.

- or -

Press **[CLR]** at the CAL OK? prompt to repeat the calibration.

A/D Calibration

A/D calibration is performed at the factory and should never require re-calibration.

Disabling a scale at P109 de-allocates memory reserved for it's A/D calibration values. A/D values will be lost!

Electrical characteristics of every Analog-to-Digital converter vary slightly. The analog-to-digital converter for each scale input is factory calibrated to achieve optimum linear response throughout the entire signal input range.

A/D calibration should not be confused with the load cell calibration procedure. It is a one-time factory procedure that requires the use of a precision load cell simulator with a 1 mV/V output. This procedure calculates a series of A/D calibration values which can be viewed in the information parameters P61110 → P61121. These values are stored in EEROM. A copy of these values are also permanently stored in the FLASH ROM for the purposes of restoring them in the EEPROM as needed. Defaulting the instrument will not affect the A/D calibration values.

Printing A/D Calibration Values

A/D calibration values should be printed or transferred to another storage medium for permanent record. They can then be restored in the event they are inadvertently deleted or when transferring multi-scale options from one scale to another.

The following figure shows a typical list of A/D calibration values. A similar printout is provided with each multi-scale option. Once saved in EEPROM, this information can be transmitted out any of the communication ports to a printer or computer as described in the following procedure.

```
100%s23640%i%e            Access Setup Modes, Allowing Changes

61099%s2%e                P61099.  Scale     2

61100%s%e                 P61100.  Crrnt mv/v
61101%s  1.000000%e       P61101.   CAL Factr   1.000000
61102%s  0.000000%e       P61102.  ReZro Wght   0.000000
61103%s  0.000000%e       P61103.  ZrTrk Wght   0.000000
61104%s8%e                P61104.  CZero    0%%
61105%s    704623%e       P61105.  Fine Zero      704623
61106%s2%e                P61106.  CGain   100
61107%s  0.914005%e       P61107.  Fine Gain    0.914005
61110%s     71451%e       P61110.  Zero Adj25     71451
61111%s     -9366%e       P61111.  Zero Adj50     -9366
61112%s   -164049%e       P61112.  Zero Ad100   -164049
61113%s  0.941097%e       P61113.  Gain Adj1   0.941097
61114%s  0.944094%e       P61114.  Gain Adj2   0.944094
61115%s  0.950346%e       P61115.  Gain Adj4   0.950346
61116%s  0.951587%e       P61116.  Gain Adj8   0.951587
61117%s   -186805%e       P61117.  AIN  NROff   -186805
61118%s   -372531%e       P61118.  AIN  NROff   -372531
61119%s   -739554%e       P61119.  AIN  NROff   -739554
61120%s  -1571870%e       P61120.  AIN  NROff  -1571870
61121%s     -3374%e       P61121.  VREF NROff      -3374
61122%s 123456%e          P61122.  SN:  123456

64102%s                   View errors after uploading!
```

Figure 4-1: *A/D Calibration Values*

If A/D calibration values have not been entered for P61110 → P61121, these parameters will show values of 0 or 1.

To print A/D calibration values:

1. From the weigh mode, key in **60100 [SELECT]** to access the GSE copyright parameter.
2. Select A/D calibration values for the scale(s):

   Key in **23640 [PRINT]** to send A/D calibration values for all scales.*

   Key in **23641 [PRINT]** to send A/D calibration values for scale #1.*

   Key in **23642 [PRINT]** to send A/D calibration values for scale #2.*

   Key in **23643 [PRINT]** to send A/D calibration values for scale #3.

   Key in **23644 [PRINT]** to send A/D calibration values for scale #4.

   Key in **23645 [PRINT]** to send A/D calibration values for scale #5.

   Key in **23646 [PRINT]** to send A/D calibration values for scale #6.

   Key in **23647 [PRINT]** to send A/D calibration values for scale #7.

   Key in **23648 [PRINT]** to send A/D calibration values for scale #8.
3. The display prompts Enter Comm#. Key in the communication port number (1 → 4).*
4. A/D calibration values are transmitted (note that all parameters from P60000 → P61122 are transmitted if you use the code **23640**).

* *When using the 460, scroll in the number and press **[ENTER]** instead of **[PRINT]** in step #2. It is also necessary to press **[ENTER]** after scrolling in the comm# in step #3.*

Restoring A/D Calibration Values

The error `Code 39 ƒA/D Cal` appears when exiting the setup mode if a scale's A/D calibration values are not found. This will occur after installing a new multi-scale option without completing the entire installation procedure. Since A/D calibration values must be stored in EEPROM, a list of the factory calibration values accompanies each option and must be entered in P61110 → P61121 of the information parameters. Once a multi-scale's A/D calibration data has been saved in the EEPROM, a copy of these values are also permanently stored in FLASH ROM.

Likewise, if a scale is disabled at P109 and changes are saved when exiting the setup mode, reserved EEPROM memory is de-allocated and A/D calibration values will be lost for that scale. To avoid this problem when temporarily disabling a scale, set P109 for "Saved" rather than "Disabled". This retains the scale's A/D calibration values in EEPROM yet the scale will not be accessible from the weigh mode. The scale can later be enabled without having to restore the calibration values.

If an error `Code 39 ƒA/D Cal` appears but you know that the A/D calibration values were previously entered and saved, it is possible to recover them from the FLASH ROM by pressing **[ENTER]** at the `Code 39` prompt and entering the module's serial# as prompted.

The serial number of the 60 Series main PC board and multi-scale options is used to reference A/D calibration values. Compare the board serial number with the serial number recorded at P61122 to ensure the correct values will be entered. If you do not have access to the correct values, contact GSE to obtain them or perform the A/D calibration procedure described in the *Entering A/D Calibration* section.

Entering A/D Calibration Values

Always verify that the serial number of the multi-scale option board or 60 Series main board matches the serial number shown for P61122.

If the A/D calibration values are stored as a text file on a computer using the method described in *Printing A/D Calibration Values* on page 4-11, then the same file can be transmitted back to the scale to restore the values. These values can also be entered manually through the front panel keypad by accessing each parameter and entering the appropriate value.

Press [F1] to begin an entry with a minus (-) sign.

To perform an A/D Calibration:

1. Power down the scale and disconnect existing load cell connections.
2. Move the E1 & E2 sense jumpers to the external (EXT) position. Failure to do so will short the load cell input connections resulting in a system reset!
3. Short together the following load cell J1 connections:

`+ SIG`	positive signal
`- SIG`	negative signal
`+ SEN`	positive sense
`- SEN`	negative sense
`SHD`	shield connection

4. Restore power and enter the calibration routine:

   100 [SELECT] 54321 [ID] [ENTER]

5. At the `New Zero?` prompt, key in **23640 [ENTER]**.
6. The display prompts `ReCAL A/D?`. Press **[ENTER]**.
7. The display prompts `Gnd. Input`. Assuming you have already made the connections in step 3, press **[ENTER]**.
8. The display will show `Mot'n Delay` and begin processing calibration values for several seconds.
9. The display prompts `Undo GND`, then `Set To 0 mV`.
10. Remove the connections from step 3 which short the signal, sense and shield.
11. Attach a precision load cell simulator to the J1 load cell connector using excitation and signal connections.
12. Move the sense jumpers E1 & E2 back to the internal (INT) position.
13. Set the simulator to 0 mV and press **[ENTER]**.
14. The display will show `Mot'n Delay` and begin processing calibration values for several seconds.
15. The display prompts `Set To 1 mV`. Set the simulator to 1 mV and press **[ENTER]**.
16. The display shows `Mot'n Delay` and begin processing calibration values for several seconds.
17. The display shows `A/D CAL'd` indicating that calibration is complete. Press **[ENTER]**.
18. Press **[ENTER]** at the `ENTER=SAVE` prompt and again at the `ENTER=EXIT` prompt to save the new calibration and exit the calibration mode.

    - or -

    To exit the calibration mode *without* saving the new calibration, press **[CLR]** at the `ENTER=SAVE` prompt. Then press **[ENTER]** at the `ENTER=UNDO` prompt and again at the `ENTER=EXIT` prompt to exit the calibration mode.

⚠ **If the calibration was performed for a multi-scale option, you must enter a serial number. Key in the serial number found on the option board and press [ENTER]. This number is stored in P61122.**

CALIBRATION ERROR MESSAGES

If calibration problems occur despite an apparently proper configuration, carefully inspect all hardware. Moisture, obstructions, defective or worn components, improper grounding techniques, and an unsuitable environment are factors which could contribute to calibration problems.

Table 4-1: Calibration Error Messages

ERROR MESSAGE	DESCRIPTION
Error Messages During Calibration	
Code30 F.S.> MAX !	The entered calibration weight, together with the currently applied signal, results in a full scale output that exceeds the maximum allowed. Verify that correct calibration weight was entered, and that the capacity at P110 of the setup mode is appropriate for the connected load cell(s). Refer to P61100 of the information parameters to view the mV/V output of the connected load cell(s).
Code31 F.S.< .1mVv	The entered calibration weight, together with the currently applied signal, results in a full scale output less than the minimum allowed. Verify that correct calibration weight was entered, and that the capacity at P110 of the setup mode is appropriate for the connected load cell(s). Refer to P61100 of the information parameters to view the mV/V output of the connected load cell(s).
Code32 ADD MORE!	The calibration weight applied does not produce an adequate change in the output signal. Increase the test weight.
Code33 ReCal ???	The calibration is insufficient to guarantee accurate results due to the calibration weight being less than 5% of capacity, or because the coarse gain was adjusted due to a significant difference in the calibration values. Repeating calibration is recommended.
)00 CalWt <0.1%	The calibration weight is less than 0.1% of full scale. Increase the test weight.
)00 CalWt <Norm	The calibration weight is less than 2% of full scale. Increasing the test weight is recommended.
)00 Entry >F.S.	The calibration weight value entered exceeds the scale capacity set at P110 of the setup mode. Verify the weight entry and the full scale capacity.
)00 Large Offst	The zero reference has changed by more than ±175%. This usually indicates the presence of a significant load on the scale when the zero reference was established. Remove all weight from the scale, press **[CLR]** and recalibrate.
)00 Add CalWT	The calibration weight was not added prior to entering the weight value. Apply the test weight and press **[ENTER]**.
)00 Must Keyin	The calibration weight value was not entered before pressing **[ENTER]**. Key in the test weight value and press **[ENTER]**. This message is also displayed if a scale number was not entered at the **Keyin Scl#** prompt before pressing **[ENTER]**.
)00 Entry Error	An invalid entry was made. For example, entering scale #3 at the **Keyin Scl#** prompt with only 2 scales enabled, or entering a test weight value during multi-point linearization that was less than the previously calibrated value.
Errors Messages When Entering or Exiting Calibration	
Code02 Under Load!	The input signal is less than negative full scale. Verify that the signal connections are not reversed and that the load cell is properly installed. Check **P61100** of the information parameters to view the mV/V output of the connected load cell(s).
Code03 Over Load!	The input signal is greater than full scale. Verify that the load cell is properly installed and that all dead load has been removed from the scale. Check **P110** of the setup mode for the correct capacity. If this is a first-time calibration, a calibration reset may be necessary. Press **[CLR]** from the calibration mode to access the Cal Reset method.
Code08 Check Conn.	The input signal is greater than ±2 times full scale. Check for proper load cell connections.
Code39 ƒ A/D Cal	The A/D calibration values for one or more enabled scales have not been entered. Press any key to continue. Enter the A/D calibration values as described in the *Restoring A/D Calibration Values* section beginning on page 4-13.

RESTORING THE CALIBRATED ZERO REFERENCE

The calibrated zero reference cannot be changed by pressing **[ZERO]** from the weigh mode or through zero tracking. These actions only serve to establish a new gross zero reference. In the case of a larger hopper scale partially full of material, the possibility of inadvertently zeroing out the existing material can pose a significant problem if it is not possible or practical to empty the hopper to reestablish zero. Should this situation occur, you can restore the last zero calibration reference, and thus restore the gross weight, by clearing P61102 (re-zero weight) and P61103 (zero track weight) in the information parameters. To prevent this situation, set the zero tracking parameters (P112, P113) and zero range parameter (P118) appropriately.

Chapter 5 WEIGH MODE OPERATION

The *Weigh Mode Operation* section of this book provides information on keypad functions, time & date, accumulation, and counting.

OVERVIEW

WEIGH MODE OPERATION

The Weigh Mode is the primary operating mode of any 60 Series instrument when it is connected to a scale platform or load cell. Depending upon the operating mode, the standard keypad keys assume different functions. This chapter defines the functions of the keys when the controller is in the Weigh Mode, plus how to disable the keys and use them in tare operations.

KEYPAD FUNCTIONS

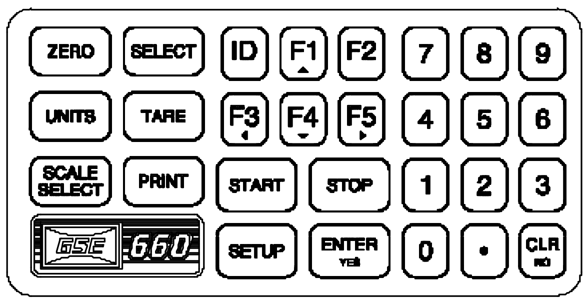


Figure 5-1: 660 Series Keypad

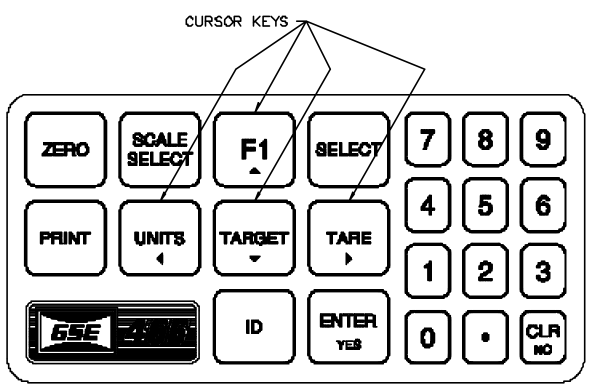


Figure 5-2: 465 Keypad

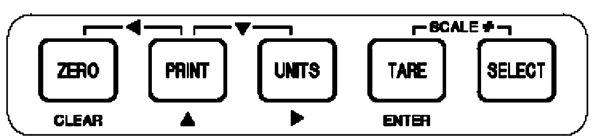


Figure 5-3: 460 Keypad

ZERO

Press **[ZERO]** to zero the current quantity/weight reading. When the controller is at *Center-of-Zero*, the international center-of-zero symbol →**0**← appears on the top line of the dot matrix display. If the name of a Custom Unit is greater than two characters, the center-of-zero symbol does not display.

If the controller is in the quantity mode, pressing **[ZERO]** sets the current mode to a gross zero quantity. If it is in the weigh mode, pressing **[ZERO]** sets the current mode to Gross Weight.

UNITS

Pressing the **[UNITS]** key in the quantity mode has no effect. Pressing **[UNITS]** in the weigh mode toggles the display through the available units selections. The operation of the units key is associated with a specific scale. Each scale can have its own units selection. The units key only affects the units of the selected scale. When another scale is selected, the units reverts to the units selection that was in effect the last time that scale was selected.

To change units on a non-selected scale, key in the desired units selection followed by a decimal point and the scale number to be affected, and then press **[ENTER]**. Converted units are rounded to the appropriate increment automatically. If the "lb/oz" units selection is used, the first digits of the numeric display show the value for pounds and the last digits show ounces. Since the upper line of the dot matrix display is used to show the units, there is no center zero symbol. As with other units designations, characters that show units turn to blanks when motion is present.

SCALE SELECT

Pressing **[SCALE SELECT]** toggles the weight display through all the enabled scale inputs, maintaining the mode of the currently selected scale for all other scales.

SELECT

Press **[SELECT]** to toggle the display through the gross, net and tare weights. Additional or alternate modes of operation can be assigned to the **[SELECT]** key at P300 → P309 in the setup mode.

The **[SELECT]** key can also be used to access an operating parameter directly by keying in the parameter number (followed by a decimal and the parameter instance if required) before pressing **[SELECT]**. For example, key in

3.2 [SELECT] to view the Gross total on Scale #2

30.4 [SELECT] to view the Quantity on Scale #4

80.17 [SELECT] to select Variable 17

80.204 [SELECT] to select Variable 204

TARE

Press **[TARE]** alone to perform an auto-tare. (A net weight of zero is then displayed.) To enter a known Tare Weight, key in the value and press **[TARE]**. In either case, the controller is placed in the net mode, unless it is already in the tare mode.

Auto-Tare and / or Keyboard Tare can be disabled in the setup mode (P802). This parameter enables the standard auto-tare key operations or allows a macro to be selected. If the macro is not programmed, the tare key is effectively disabled. The macro can also be programmed to detect keyboard entries. A macro setup such as this allows for keyboard tare entries or strictly auto-tare operations.

Also refer to P162, P163 and P660 for setup parameters that control the tare operation.

PRINT

Press **[PRINT]** to send custom specified data to a printer, computer, or other device. If the specified custom transmit does not exist (for example, has not been set up yet) or is set for "off" at P991 then "NOT FOUND" is displayed and the print operation is aborted.

ID

The **[ID]** key has multiple functions. You can use it to:

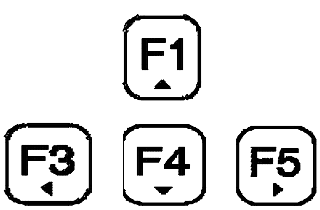
- Access variables
- Access the macro menu selections (see page 9-9)
- Access the database menu selections (see page 11-6)
- Invoke a specific macro (see page 3-41).

The exact function of this key depends on the how it is defined in the setup mode at P806 (see page 3-41).

To access a variable using the **[ID]** key, simply key in the variable number and press **[ID]**. For example, pressing

5 [ID]

will access variable #5.



ARROW / FUNCTION KEYS

The arrow keys, also known as the cursor keys or function keys, have two primary functions outside the setup mode – invoking macros and scroll in alpha-numeric characters during an operator entry.

Invoking Macros

In the weigh mode, pressing **[F1]**, **[F2]**, **[F3], [F4]** or **[F5]** will invoke macros 1 → 5 respectively. If a macro is *not* programmed for one of these keys, the key is considered disabled for invoking macros in this manner.

Entering Alpha-Numeric Characters

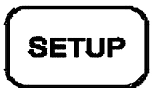
During operator entries for string variables or entries initiated by a macro command, the arrow keys take on the role of cursor keys allowing you to scroll in the alpha-numeric text to be entered. See *Character Entry* on page 5-6 for complete details.

START

Pressing **[START]** will invoke macro 6. This key has no other function. If macro 6 is *not* programmed, this key is disabled.

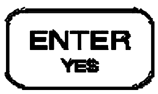
STOP

Pressing **[STOP]** will invoke macro 7. This key has no other function. If macro 7 is *not* programmed, this key is disabled.



SETUP

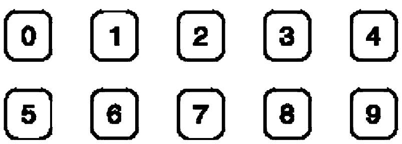
Pressing **[SETUP]** will invoke macro 8. This key has no other function. If macro 8 is *not* programmed, this key is disabled.



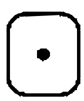
ENTER/YES

Press **[ENTER/YES]** to confirm certain numeric entries. As entries are keyed into the entry buffer, the **[ENTER Yes]** key completes the entry.

THE NUMERIC KEYS



Press the numeric keys to enter numeric values 0 through 9.



THE DECIMAL POINT KEY

Press **[.]** to establish a decimal point or perform an accumulation in the weigh or count mode (see page 5-9).

CLR/NO

Press **[CLR/NO]** to clear an entry in process.

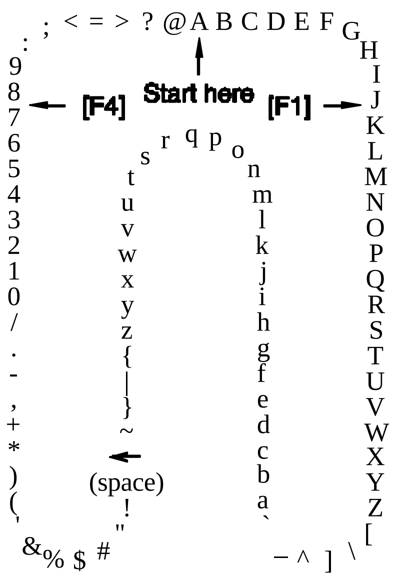


Figure: Character Listing

CHARACTER ENTRY

When alpha characters and other non-numeric characters are to be entered using the front panel keypad, use the **[F1▲]**, **[F3◀]**, **[F4▼]**, and **[F5▶]** arrow keys to "build" the entry:

[F1▲] scrolls forward through the list of uppercase and lowercase alphabetic, ASCII punctuation, and numeric characters. See the figure *Character Listing* for the order of how these characters are listed.

[F4▼] scrolls backward through the list of characters.

[F3◀] is used as a backspace key.

[F5▶] advances the cursor to the next character position.

To begin data entry:

1. Press **[F1▲]** to display an "A" in the bottom-right position of the dot matrix display.
2. Use the **[F1▲]** and **[F4▼]** keys to cycle through the character set. Holding down the **[F1▲]** or **[F4▼]** key cycles the controller through the selections more quickly.
3. When the desired character is displayed, press **[F5▶]** to move the cursor to the next location where another "A" will be displayed. Repeat the character selection process of step 2 until you have selected all of the characters needed for the entry.
4. Press **[ENTER]** to complete the entry.

460 CURSOR KEYS

Refer to Figure 5-4 for the cursor key functions on a Model 460. Note that the left and down arrow keys are two-key combinations.

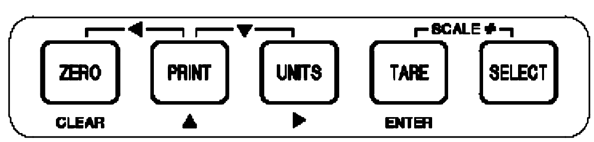


Figure 5-4: 460 Cursor Keys

DISABLING FRONT PANEL KEYS

Front panel keys can be individually disabled at P800 → P819 in the setup mode by assigning an undefined macro number to a key. Refer to page 3-41 for more details.

Disabled keys can be temporarily re-enabled by holding the **[CLR]** key at power-up until Macro Disbl is displayed. This resets all key functions to normal until power is cycled or upon exiting the setup mode.

TIME & DATE

Your controller's time and date is set at the factory.

The 660 Series controllers include a battery backed time-date feature. This means the time and date *do not* have to be entered every time the controller is powered up.

The 460 Series indicators offer an *optional* battery backed time-date module. Without this module, the time and date must be re-entered each time the indicator is powered up.

The Time-Date feature enables printouts of the time, day of the week, and date in many possible formats. By combining macro and setpoint capabilities with the Time-Date feature, *alarm sequences* can be devised to cause certain events to occur at pre-timed intervals.

VIEWING THE TIME AND DATE

0%2!01 12:00 :00am

When the indicator is in the weigh mode, the time and date can be displayed simultaneously by keying in **11 [SELECT]**. The date is then displayed on the large numeric display in the format MM.DD.YY (or DD.MM.YY for international style) and the time is displayed on the dot matrix display in the format HH:MM:SS.

The time can be displayed in a 24-hour or 12-hour format, showing a.m. or p.m. as appropriate, depending on how the time-date feature is set up. You must press **[SELECT]** to return to where you were.

ENTERING THE TIME

You can enter a new time in the *Setup Mode* at P500 by keying in HH.MM.SS in a 24-hour format. You must separate hours and minutes entries must by a decimal point. Seconds entries are optional, and if omitted, they are initially set to zero.

To specify seconds, you must also use a decimal point to separate them from minutes. You do not have to enter leading zeroes. For example, if you enter 8.9.45 and press **[ENTER]**, the time will be set to 08:09:45. If you enter 15.02 and press **[ENTER]**, the time is set to 15:02:00.

If you enter the time improperly, the prompt “try h.m.s” is displayed.

ENTERING THE DATE

You can enter a new date at power-up or into P501 by keying in the numbers for MO.DA.YR (or DA.MO.YR if the international format was selected) and pressing **[ENTER]**.

You must separate month, day, and year entries by decimal points. You do not have to enter leading zeroes. For example, if you enter 8.1.96 and press **[ENTER]**, the date is set to 08/01/96. If you enter the date improperly, the prompt “try m.d.y” (or “try d.m.y” for the international format) is displayed.

OPERATOR ACCESS TO TIME/DATE

You can access and change the time and date directly from the weigh mode by selecting parameter 11. If P502 is set for “Enbld” (changeable), then you will be allowed to change the time when you select parameter 11. This allows end users to adjust the time without having to know the setup access code for the controller. If Parameter P502 is set for “Disbl”, then the time/date can only be changed by accessing the setup mode at P500 and P501.

TIME/DATE TRANSMIT CODE (FORMATS)

Format code combinations are included in the time/date format selections for transmitting the T/D parameter from a custom transmit table or a macro. These format codes allow a time/date type variable (current time, or variables defined as U-INTs) to be transmitted as a numeric value in terms of the number of seconds elapsed since Jan 1, 1970. This is useful in uploading time/date information to a computer for spreadsheet applications. Refer to *Chapter 8, Communications* for more information on time/date transmitting (multiple formats).

TIME AND DATE PARAMETER SETUP

Time and date setup parameters begin at P500 → P504. Refer to page 3-37 for complete details.

The default name used when transmitting the time & date parameter can be changed in the setup mode at P611 (see page 3-38).

TIME/DATE SPECIFICATIONS

The 60 Series battery-backed time/date module is a continuously running clock, even when the indicator is powered down. The time and date is set (as Eastern Standard Time) at the factory prior to shipment. See page A-16 for complete specifications.

TIME / DATE OPERATION

Apply power to the controller and verify proper operation. To verify the time and date values, select the time/date mode. Key in **11 [SELECT]**

The display will show the date on the large numeric digits and the time on the dot matrix display. For example, for July 16, 1996 at 3:27:23 p.m., the display will appear as 07.16.96 03:27:23pm.

Refer to the P500 setup parameters to enter the time and date or to change the format of the time and date. To include the current time and / or date into the data transmissions to a printer or other peripheral, refer to the *Custom Transmit (Selections) Parameter Setup* section *Chapter 8, Communications*, *Chapter 9, Macros*, and *Chapter 10, Setpoints* for examples on setting up alarm routines (alarm routines configured using macros and setpoints).

ACCUMULATION

Many applications such as multiple dump batching systems, axle weighing systems, and inventory control require accumulation of weight values. The accumulation parameters provide an easy method for totaling multiple weighments. Separate accumulation parameters are maintained for GROSS and NET weights for each enabled scale.

ACCUMULATION PARAMETERS

Table 5-1 lists all operating parameter that are affected an accumulation operation.

Table 5-1: Accumulator Parameters

PARAMETER	DESCRIPTION
3	Gross Total
4	Gross Total + Current Gross Weight
5	Gross Total – Current Gross Weight
6	Net Total
7	Net Total + Current Net Weight
8	Net Total – Current Net Weight
9	Number of Accumulations
43	Total of All Gross Totals
44	Total of All Net Totals

PERFORMING ACCUMULATIONS

Before attempting an accumulation, either the GROSS parameter (0P), GROSS TOTAL parameter (3P), NET parameter (1P) or NET TOTAL parameter (6P) must be selected. An accumulation may then be performed by placing a load on the scale and pressing **[.] [ENTER]**.

Accumulations are motion inhibited, meaning the accumulation will not occur until the scale becomes stable within the limits defined in the setup mode at P114 (motion divisions) and P115 (motion time delay). ***Mot'n Delay*** is displayed while the scale is unstable.

Once the scale becomes stable, all accumulation parameters for both GROSS and NET are updated.

INITIALIZING ACCUMULATION TOTALS

Accumulation parameters are stored in ***non-volatile*** memory, saving their values during power loss so that accumulations can resume when power is restored.

To reset accumulation parameters, select either the GROSS TOTAL or NET TOTAL parameter and press **[CLR]**. The display prompts ***ENTER =CLR!***. Press **[ENTER]** to confirm. All accumulation parameters are reset to zero (0).

To initialize accumulation parameters, select either the GROSS TOTAL or NET TOTAL parameter, key in the desired total and press **[ENTER]**. The display prompts ***ENTER =New!***. Press **[ENTER]** again to confirm. Only the selected parameter is initialized, the other remains unchanged. The NUMBER OF ACCUMULATIONS parameter is reset regardless of which parameter is initialized.

PREVENTING DOUBLE ACCUMULATIONS

Double accumulations occur when the same load is added to the accumulation totals more than once. This potential problem can be eliminated by properly setting the Return-to-Zero setup parameter (P122 RTZ).

The RTZ parameter mandates that the applied gross weight fall below a predetermined value before the next accumulation may occur. If the weight is not removed, ***CLEAR WGHT!*** is displayed indicating that an accumulation can not occur.

As an example, suppose we are accumulating the weight of aluminum cans on a 100 pound capacity scale. The cans are dumped into a container that has been tared on the scale, then accumulated. The container is emptied, refilled and accumulated again just before the change of a work shift. A new worker arrives at the same station to find a container full of cans and attempts to perform an accumulation. The display prompts ***CLEAR WGHT!*** indicating that these cans have already been accumulated. Assuming we had set P122 to a 1% Return-to-Zero value, the GROSS weight must fall below 1 pound before the next accumulation can occur.

Note that if P122 was set to 100%, a double accumulation would have occurred and the same cans would have been counted twice.

ACCUMULATION COUNTER

Operating parameter 9, accumulation counter, is incremented by one (1) each time after every accumulation. This parameter tell us how many accumulations have occurred. This can be useful to determine the average weight of each weighment. For example, the macro assignment

```
80.1P=3.0P/9.0P%o
```

divides the GROSS TOTAL of the currently selected scale by the NUMBER OF ACCUMULATIONS and assigns the average to variable #1.

The value of P9 can not be preset manually. It is reset (set to zero) any time either the GROSS TOTAL or the NET TOTAL parameter is reset or initialized manually. It can, however, be assigned a value with a macro assignment

```
9.1P=10%o
```

LARGE ACCUMULATION VALUES

As the GROSS TOTAL or NET TOTAL increases, the number of decimal places will change to accommodate the number on the 6 digit display. For example, a value of 9999.00 may be displayed exactly as that number.

Yet a value of 99999.00 will be displayed as 99999.0 (one decimal place is lost). When an accumulation value exceeds 999,999 it can not be represented on the 6 digit display. Instead, ***Code04 Num > Dsply*** is displayed. The actual value is still retained and may be transmitted to another display or printer.

ACCUMULATIONS AND NTEP

It should be noted that enabling setup parameter 440, NTEP, has several significant effects on the operation of the accumulation parameters. When using the accumulation feature in a legal-for-trade application these effects should be studied. Refer to page 6-8 for more details.

COUNTING

The ability to establish a piece-weight and count parts to a high degree of accuracy is another feature of the 60 Series instruments. The counting feature is *not* enabled as a standard default.

The Counting Operation is the ability of the controller to *count* the quantity of parts on the weigh platform. In order for the controller to accomplish this, a sampling operation must first be performed to establish the actual weight of one of the pieces to be counted.

COUNTING MODE (KEY OPERATION)

In order to activate the counting operations, the controller must be in the *quantity* mode. To access the quantity mode, place parameter 30 in the selectable modes of controller operation. Pressing the **[SELECT]** key, simply step through all enabled modes of controller operation, including the quantity mode, or enter the quantity mode directly by keying in **30 [SELECT]**.

Once in the quantity mode, if you press the **[ENTER]** key alone, without first pressing a number key, the controller will perform an auto-tare, establishing a net zero as a starting point.

The current net weight will then be actively displayed and the prompt "**Add XX**" will be shown, where "XX" is the default value set at parameter **P182**.

If the specified number of parts is added, press the **[ENTER]** key to accept the sample. If a different number of parts are added other than the default sample size set at **P182**, key in the actual number of parts added and then press the **[ENTER]** key. The subsequent results depend on the selections made for the auto-enhance and minimum accuracy selections. Refer to those sections for further information.

If you enter a number before you press **[ENTER]**, the controller does ***not*** perform an auto-tare. The entry is assumed to be the number of parts already present on the scale platform. If, when you enter a number, the current mode is GROSS or GrTot, then the stored tare weight is assumed to be from a previous weighment and is cleared out. Otherwise, any previous tare is assumed to be for a container that is in use, the tare is retained, and the current net weight is used to calculate the piece weight based on sample size entered. This allows for a situation where the tare weight of the container (if any) is consistent and has already been established. Then you can add the parts and key in the number of parts.

Note, however, that the first method is recommended and will eliminate the possibility of inappropriate tare weights from affecting the piece weight calculation.

When the quantity mode is accessed and the residing piece weight value is 0.00, the prompt "Must Sampl" will be displayed. Press [ENTER] to begin a sample to determine a piece weight mode (refer to the next section) or key in a piece weight value at P34.

Although the sampling process may be performed in a number of ways, the recommended method is to *access the quantity mode*, place an empty box or empty container on the scale platform, and then press the **[ENTER]** key. The controller will perform an auto-tare resulting in a zero net weight. The display will then prompt you to *Add xx* where the "xx" is the sample quantity of parts (sample size) set by parameter P182. (The manufacturer default setting is 10 pieces)

Now, add the requested number of parts to the controller's scale platform, and press the **[ENTER]** key. If the sample's total weight was sufficient, the piece weight will be calculated and the sample quantity will be displayed. Otherwise, you may be prompted to *add* more parts. Additional parts are requested if accuracy minimums are not met. The exact prompts will depend on whether the auto-enhance and/or minimum accuracy assurance features (Parameter 181 and Parameter 183, respectively) have been enabled.

The minimum amount of weight required for the sample routine to meet the selected accuracy requirements for the specified scale capacity is considered. If the weight of the sample is not detectable or barely detectable then the message *Code 32 ADD MORE!* is displayed briefly. This will most often occur when you press the **[ENTER]** key without adding any parts. If, in fact, you have placed the parts on the scale, either the parts are too light to count on that capacity platform or you must hand count a much larger quantity of parts in order to perform the sample.

NEGATIVE PIECE SAMPLING

In order to perform a negative sample routine, access the *quantity mode*, place a *full* or *partially full* container of parts on the scale, and press **[ENTER]**. The controller will perform an auto-tare resulting in a zero net weight.

The display will prompt you to *Add: xx* where "xx" is the sample quantity of parts (sample size) set by parameter P182. Disregard the *Add:xx* prompt, assume *Take:xx*, and *remove* the requested number of parts. The weight difference of the requested number of parts is then calculated by the controller. Press the **[ENTER]** key. If the sample's total weight was sufficient, the piece weight will be calculated and the sample quantity will be displayed. Otherwise, you may be prompted to *Take* additional parts. The exact prompts will depend on whether the auto-enhance and/or minimum accuracy assurance features have been enabled.

The minimum amount of weight required for the sample routine to meet the selected accuracy requirements for the specified scale capacity is considered. Continue to enhance, as desired. Key in the tare weight of the empty container, add the parts removed for sampling back into the container. The quantity displayed will be the total quantity of parts in the container.

If the weight of the sample removed is not detectable or barely detectable, the message *Code 32 ADD MORE!* is displayed briefly. This actually means "Take" if a negative sampling is to be performed. The controller does not know if the operator intends to perform a positive or a negative sampling if no weight is removed. This prompt indicates that more weight must be *removed* adding to the overall sample weight. This will most often occur when the **[ENTER]** key is pressed without taking out any parts. If the parts were in fact taken from the scale, either the parts are too light to count on that capacity platform or a much larger quantity of parts must be hand counted out in order to perform the sample.

USING AUTO-ENHANCE

When the auto-enhance feature (P181) is enabled, after a sample operation is performed the controller calculates the number of parts that may be added to the scale while keeping the controller's uncertainty of the number of parts on the platform within +/- 1/3 of a part.

If the total number of parts that may be counted without error is more than the number sampled, the controller will briefly display the maximum number of parts that can be used to perform an auto-enhance. If additional parts are added—while keeping the total displayed quantity *less than or equal to* the displayed maximum enhance amount—then as soon as motion ceases, the controller will automatically re-calculate the piece weight based on the new larger quantity. Then it will display the new maximum enhancing quantity and the process will repeat. The controller does, however, have a minimum threshold of the piece weight for which enhancements can occur. This minimum enhancing piece weight is 0.0084% of capacity. If the piece weight is less than this amount then enhancements are not possible and the message *Can't Enhnc* will be displayed. Whenever you key in a piece weight, the controller clears its calculated accuracy register and last sample register since neither the accuracy nor the sampled quantity of the entered piece weight are known. Therefore, auto-enhancements are not possible with entered piece weights.

USING MINIMUM ACCURACY ASSURANCE

The minimum accuracy feature (P183) ensures that parts counting operations will result in a pre-specified minimum accuracy. This is accomplished two ways:

- By requiring the sampled parts to meet or exceed a minimum weight
- When the enhance feature is enabled, by requiring additional enhancements after the initial sample operation has been performed before a large number of parts may be counted.

If the enhance feature is disabled and the weight of the sampled parts is insufficient to guarantee the required accuracy (as set by P183), you are prompted to *Add xx* parts. "Xx" represents the additional number of parts that you must hand-count and add to the scale in order for the accuracy to be achieved. However, if the enhance feature has been *enabled*, the results of a sampling operation may vary. These variations are detailed in the following sections.

ACHIEVED ACCURACY LESS THAN REQUIRED

Achieved accuracy is insufficient to allow an enhancement to occur with at least 5 additional pieces: The controller prompts you to *Add xx* parts, where "xx" is the number of additional parts required to allow either enhancements to be achieved, or the number required to achieve the required accuracy, whichever is less. Add the specified number of parts and press **[ENTER]**, or add even more parts, key in the *TOTAL* number of parts on the scale, and press **[ENTER]**.

The controller will show the current quantity on the numeric display and the message *Enhn# xx* on the alpha display. "Xx" is the maximum number of parts that may be on the scale in order for an enhancement to occur. You can then add more parts, up to the number shown.

As soon as motion ceases, if the quantity displayed is more than the original sampled amount and less than the displayed maximum enhance amount, the piece weight will be re-calculated. Then the achieved accuracy will be re-evaluated. If the required accuracy still has not been achieved, another enhancement will be required and this step will be repeated with a larger maximum enhance amount. If you add more parts than specified, *Code 53 Accy >Req'd* is displayed, indicating that the required accuracy has not been achieved and counting may not continue.

ACHIEVED ACCURACY MET REQUIREMENTS

Achieved accuracy is insufficient to allow an enhancement to occur: The message *Can't Enhnc* is displayed briefly. This will not normally occur unless the sampled number of parts was very large and / or the piece weight was relatively small. Sampled amount is sufficient to allow enhancements: The maximum number of parts which can be counted and allow an enhancement to occur is displayed briefly and the quantity is then displayed.

In order for a given accuracy to be achieved, the weight of the sampled parts must meet or exceed a specific minimum. Table 5-2 shows the required sample weight for various accuracy requirements on a variety of platform capacities.

Table 5-2 s calculated for the filter selection of 8 seconds. It also is based on a load cell full scale input of 2 mV/V.

- If a 1 mV/V full scale cell is used, all minimum weight requirements should be 2 times the value stated.
- If a 3 mV/V cell is used, the value should be 2/3 the value shown.

Table 5-2: Minimum Sample Weight Accuracy Requirements

REQUIRED ACCURACY	PERCENT OF CAPACITY	2-LB PLATFORM	10-LB PLATFORM	50-LB PLATFORM	200-LB PLATFORM
90%	0.028%	0.00056lb	0.0028lb	0.014lb	0.056lb
91%	0.032%	0.00064lb	0.0032lb	0.016lb	0.064lb
92%	0.035%	0.00070lb	0.0035lb	0.018lb	0.070lb
93%	0.040%	0.00080lb	0.0040lb	0.020lb	0.080lb
94%	0.047%	0.00094lb	0.0047lb	0.024lb	0.094lb
95%	0.056%	0.00112lb	0.0056lb	0.028lb	0.112lb
96%	0.070%	0.00140lb	0.0070lb	0.035lb	0.140lb
97%	0.094%	0.00187lb	0.0094lb	0.047lb	0.187lb
98%	0.140%	0.00280lb	0.0140lb	0.070lb	0.280lb
98.48%	0.185%	0.00370lb	0.0185lb	0.093lb	0.370lb
99%	0.280%	0.00560lb	0.02800lb	0.140lb	0.560lb
99.20%	0.350%	0.00700lb	0.0350lb	0.175lb	0.700lb
99.40%	0.470%	0.00940lb	0.0470lb	0.235lb	0.940lb
99.60%	0.700%	0.01400lb	0.0700lb	0.350lb	1.400lb
99.68%	0.875%	0.01750lb	0.0875lb	0.438lb	1.750lb
99.80%	1.400%	0.0280lb	0.1400lb	0.700lb	2.800lb
99.88%	2.340%	0.04670lb	0.2340lb	1.170lb	4.680lb
99.96%	7.000%	0.14000lb	0.7000lb	3.500lb	14.000lb

If a different filter selection is used other than the 8 second filter, refer to Table 5-2 for minimum weight required multiplication factors. Take the minimum weight required value from the table using the 8 second filter and multiply it by the factor for the specified filter selection. This will yield the minimum weight required for the new filter selected.

Note, however, that parts counting based on weight is dependent on a reasonably consistent part weight. Some plastic parts vary in weight from piece to piece by 10% or more. Attempting to count these items with a high degree of accuracy will require a very large hand-counted random sample of the items during the piece weight calculation process. The minimum accuracy assurance is intended to guide the operator in sampling parts with a fairly consistent piece weight.

COUNTING PARTS

Several ways to count parts using the 60 Series indicators:

Method 1: Counting a specific number of parts

1. Access the quantity mode.
2. Place the empty container on platform (optional).

3. Press the **[ENTER]** key. The controller tares to a zero net weight. The display shows the current net weight and prompts, *Add 10* (the actual number can be programmed by setup parameter P182).
4. Place the specified number of parts on the scale.
5. Press **[ENTER]**. (Or add any number of parts, key in the number added, then press **[ENTER]**.) The controller calculates the piece weight of the sample parts. If the number of parts added was insufficient to achieve the required accuracy as set by P183, then you are prompted to add an more parts.
6. Add the specified number of parts and press **[ENTER]**.
7. Add the additional parts to be counted on the scale.

Method 2: Counting with piece weight enhancement

1. Access the quantity mode.
2. Place empty container on platform (optional).
3. Press the **[ENTER]** key. The controller tares to a zero net weight. The display shows the current net weight and prompts: *Add 10* (the actual number can be programmed in setup parameter P182).
4. Place the specified number of parts on the scale.
5. Press **[ENTER]**.
6. The controller calculates the piece weight of the sample parts and momentarily displays the maximum number of parts that can be added for a piece weight enhancement to occur. Then it displays the minimum achieved accuracy.
7. If a greater accuracy is desired, add more parts but not more than the maximum enhancing quantity.
8. As soon as motion ceases, the controller recalculates the piece weight and briefly displays the new maximum number of pieces that can be added and still accurately enhance the piece weight.
9. Repeat as many times as desired.
10. Add the additional parts to be counted.

Method 3: Counting with a known container weight

1. Access the quantity mode.
2. Place the full container of parts on the weigh platform.
3. Press the **[ENTER]** key.

   The controller tares to a zero net weight. The display shows the current net weight and prompts, *Add 10*. (Add means *take* in this instance). The actual number can be programmed in setup parameter P182).
4. *Remove* the specified number of parts from the box.
5. Press **[ENTER]**.

The controller calculates the piece weight of the sample parts removed.

6. Key in (or bar-code scan) the tare weight of the container.
7. Add the sample parts back in that were removed.

   The displayed quantity then is the total number of parts in the container.

ADDITIONAL COUNTING RELATED PARAMETERS TO CONSIDER

For the best counting results, certain setup parameter selections should be set. Refer to parameters P112 → P116, P119 and P162 → P163 beginning on page 3-27 for additional information.

MULTI-RANGE OPERATIONS

Multi-Range operation allows the scale's division size to change as weight increases. Two or three ranges can be specified. This feature is often used in truck/railroad scales where a truck might be weighed in 20 pound divisions up to 200,000 pounds, then switch to 50 pound divisions up to 400,000 pounds for weight rail cars.

SCALE NUMBER AND RANGE INDICATION

12250	lb ˆ Gross

Range 1 (Low Range)

55550	lb ‰ Gross

Range 2 (Middle Range)

277250	lb Š Gross

Range 3 (High Range)

The current range is determined by the gross weight. Upon power-up, range 1 is in effect, the division size specified at P127. When the gross weight exceeds the limit of range 1 specified at P126, range 2 goes into effect using the increment specified at P129. When the gross weight exceeds range 2 specified at P128, range 3 goes into effect. The limit of range 3 is the full scale capacity set at P110 using the division size specified at P111.

In order for the range in effect to change back to the lowest range, any of the following three events must occur.

- The gross weight must fall below plus ¼ grads on range 1

  i.e. P127 = 20 Lbs. The scale must fall below 5 Lbs.

- The scale must be re-zeroed with the **[ZERO]** key (or %z macro command)

If the net weight is less than –5 grads and the gross weight is within the zero range (per P118), then the gross weight is zeroed and the tare is cleared and the range is reset to range 1.

The current range applies to the displaying and printing of the gross, net, and tare weights. Also, whenever rounded gross (20P) or rounded net (21P) are referenced, the rounding is done according to the range in effect at the time of the reference. Of course, the rounding of 20P and 21P is done to the default units, not the current units.

Scale number and range indication are not shown for net or gross if the net or gross is less than 1.25 grads.(same as before, except it now applies to 1.25 grads for lowest defined range.) When **[ZERO]** is pressed, if the indicator is within the zero range, then it re-establishes a new zero, clears the tare, and sets the range to the low range.

ACCUMULATIONS

When accumulations are performed, the rounded values (20P & 21P) are added to the previous total. Since a total may be the result of several additions of data from different ranges (thus having different increments), it is necessary that the increment used for rounding displayed and transmitted totals to be the number of decimal places of the increment of the current units in the lowest range. This should cause the total to be displayed as the exact value of the sum of the individual values accumulated, at least in the default units.

i.e.

range 1 x .02 example value: 1.38

range 2 x .05 example value: 5.35

range 3 x .1 example value: 10.2

----

total: 16.93

Note: The least significant digit of the total, 0.03, is not divisible by any of the specified increments for the three ranges.

WARNINGS

A warning message will appear when (P126) LowRangeMax > (P110) capacity OR (P128) MidRangeMax < (P126) LowRangeMax then error message "Code 38" "RangeError" is displayed briefly and you cannot exit the setup mode. Press any key and you are bounced back into the setup mode for the offending parameter.

Also, if the resolution in either the low or middle range exceed the high or low values, then warnings are displayed only. If resolution exceeds minimum or maximum values an error is displayed:

- MAX_RES 100000.
- HIGH_RES 25000.
- LOW_RES 100
- MIN_RES 1

In case of a resolution or range error, once a key is pressed the indicator returns to the setup parameter in question.

CAL MODE

The multi-range feature does not take effect during the cal mode. The increment during cal is always that specified by P111 or the associated increment for alternate units.

PRINTING

When default units is specified in the format code for printing a parameter that has units, the default units will always print with the increment specified for full scale (P111), regardless of the current range in effect on that scale.

VARIABLES

Float type variables that are set to be scale specific at P687 will be rounded according to the magnitude of the variable's value in comparison to the range settings for that scale. For example, if the multi-range limits are

Low Range	**0 - 10 by .001**
Middle Range	**10 - 50 by .002**
High Range	**50 - 100 by .01**

then copying the value 45.327 into a scale specific variable would be represented as 45.328. Copying the value 55.327 into the same variable would be represented as 55.33.

Chapter 6 LEGAL-FOR-TRADE

This chapter contains information on NTEP and OIML regulations, sealing and audit trails, and other requirements.

OVERVIEW

LEGAL FOR TRADE

Legal-for-Trade requirements vary by location. Ensure that the indicator is installed in accordance with all local regulations.

The 60 Series parameter setup does not ensure compliance with legal-for-trade installations as mandated by local weights and measures authorities. This section explains how to configure the 60 Series instruments to comply with various regulations and describes other features that make the indicators suitable for installations worldwide.

OIML AND INTERNATIONAL OPERATION

The International Organization of Legal Metrology is an inter-governmental body which harmonizes the national metrology regulations of its worldwide members. A list of regulation publications can be obtained from the Bureau International de Métrologie Légale (BIML) in Paris, France.

In order to configure the indicator to comply with OIML requirements, P410 must be enabled in the setup mode. Doing this will ensure the following:

An over-load condition will result when the gross weight exceeds nine (9) graduations over the full scale capacity.

Full scale capacity is always referenced from the last zero calibration reference, not the last zero acquired by pressing **[ZERO]**.

The keypad is remapped for the international version (see the *International Keypad* section on page 6-3).

Presettable parameters will give indication that a value has been entered manually (see *Renaming Operating Parameters* on page 7-7).

Most NTEP requirements will also apply. See the *NTEP* section of this chapter for additional considerations.

Enabling OIML Operation

P41)	OIML Enbld

OIML operation is enabled at P410 of the setup mode.

To enable OIML operation:

1. Access the setup mode as described in the *Accessing The Parameter Setup Mode* section beginning on page 3-3.
2. Key in **410 [SELECT]** to access the “OIML” parameter.
3. Key in **9991 [ENTER]**. The display prompts `Enter toCLR`.
4. Press **[ENTER]** to enable OIML.
5. Exit the setup mode as described in the *Exiting The Parameter Setup Mode* section on page 3-9.

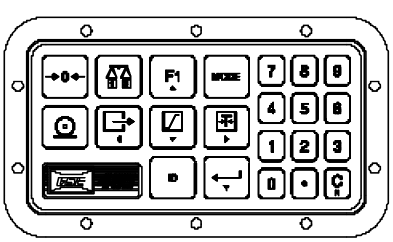
To disable OIML, key in 9990 [ENTER] in step 3. Performing a Default All (P65001) or Default – Cal (P65002) will also disable OIML.

INTERNATIONAL KEYPAD (460 SERIES)

Table 6-1 shows the correlation between the US keypad and the international keypad for the 460 Series controllers.

Table 6-1: 460 Series International Keypad Functions

INT'L	USA	FUNCTION	SERIAL COMMAND	DECIMAL	HEX	REASSIGNMENT PARAMETER
•	•	Decimal Point	.	46	x2E	P809
0	0	Zero	0	48	x30	P810
1	1	One	1	49	x31	P811
2	2	Two	2	50	x32	P812
3	3	Three	3	51	x33	P813
4	4	Four	4	52	x34	P814
5	5	Five	5	53	x35	P815
6	6	Six	6	54	x36	P816
7	7	Seven	7	55	x37	P817
8	8	Eight	8	56	x38	P818
9	9	Nine	9	57	x39	P819
F1	F1	Invokes Macro 1	1%^	128	x80	-
		Invokes Macro 2	2%^	129	x81	-
		SCALE SELECT	%`	224	xE0	P804
		CLEAR/NO	%c	227	xE3	P808
		ENTER/YES	%e	229	xE5	P807
ID	ID	ID	%i	233	xE9	P806
		PRINT	%p	240	xF0	P805
		SELECT	%s	243	xF3	P800
		TARE	%t	244	xF4	P802
		UNITS	%u	245	xF5	P803
→0←		ZERO	%z	250	xFA	P801



The following models use the 460 Series international keypad:

- Model 460 (U.K. line cord) P/N: 200460-03100
- Model 460 (Euro line cord) P/N: 200460-04100
- Model 465 (U.K. line cord) P/N: 200465-03100

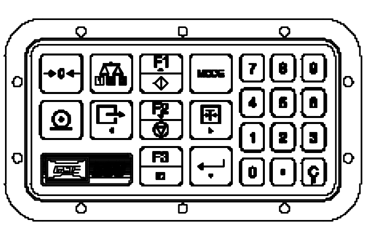
- Model 465 (Euro line cord) P/N: 200465-04100

INTERNATIONAL KEYPAD (560 SERIES)

Table 6-2 shows the correlation between the US keypad and the international keypad for the 560 Series controllers.

Table 6-2: 560 Series International Keypad Functions

INT'L	USA	FUNCTION	SERIAL COMMAND	DECIMAL	HEX	REASSIGNMENT PARAMETER
.	.	Decimal Point	.	46	x2E	P809
0	0	Zero	0	48	x30	P810
1	1	One	1	49	x31	P811
2	2	Two	2	50	x32	P812
3	3	Three	3	51	x33	P813
4	4	Four	4	52	x34	P814
5	5	Five	5	53	x35	P815
6	6	Six	6	54	x36	P816
7	7	Seven	7	55	x37	P817
8	8	Eight	8	56	x38	P818
9	9	Nine	9	57	x39	P819
F1	F1 START	Invokes Macro 1	1%^	128	x80	-
F2	F2 STOP	Invokes Macro 2	2%^	129	x81	-
F3 ID	F3 ID	ID	%i	233	xE9	P806
	SCALE SELECT	SCALE SELECT	%`	224	xE0	P804
	CLR NO	CLEAR/NO	%c	227	xE3	P808
	ENTER YES	ENTER/YES	%e	229	xE5	P807
	PRINT	PRINT	%p	240	xF0	P805
MODE	SELECT	SELECT	%s	243	xF3	P800
	TARE	TARE	%t	244	xF4	P802
	UNITS	UNITS	%u	245	xF5	P803
→0←	ZERO	ZERO	%z	250	xFA	P801



The following models use the 560 Series international keypad:

- Model 560 (U.K. line cord) P/N: 200560-03100
- Model 560 (Euro line cord) P/N: 200560-04100
- Model 562 (U.K. line cord) P/N: 200562-03100

- Model 562 (Euro line cord) P/N: 200562-04100

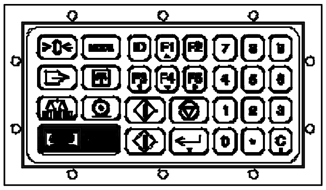
INTERNATIONAL KEYPAD (660 SERIES)

Table 6-3 shows the correlation between the US keypad and the international keypad for the 660 Series controllers.

Table 6-3: 660 Series International Keypad Functions

INT'L	USA	FUNCTION	SERIAL COMMAND	DECIMAL	HEX	REASSIGNMENT PARAMETER
.	.	Decimal Point	.	46	x2E	P809
0	0	Zero	0	48	x30	P810
1	1	One	1	49	x31	P811
2	2	Two	2	50	x32	P812
3	3	Three	3	51	x33	P813
4	4	Four	4	52	x34	P814
5	5	Five	5	53	x35	P815
6	6	Six	6	54	x36	P816
7	7	Seven	7	55	x37	P817
8	8	Eight	8	56	x38	P818
9	9	Nine	9	57	x39	P819
F1	F1	Invokes Macro 1	1%^	128	x80	-
F2	F2	Invokes Macro 2	2%^	129	x81	-
F3	F3	Invokes Macro 3	3%^	130	x82	-
F4	F4	Invokes Macro 4	4%^	131	x83	-
F5	F5	Invokes Macro 5	5%^	132	x84	-
		Invokes Macro 6	6%^	133	x85	-
		Invokes Macro 7	7%^	134	x86	-
		Invokes Macro 8	8%^	135	x87	-
		SCALE SELECT	%`	224	xE0	P804
		CLEAR/NO	%c	227	xE3	P808
		ENTER/YES	%e	229	xE5	P807
ID	ID	ID	%i	233	xE9	P806
		PRINT	%p	240	xF0	P805
		SELECT	%s	243	xF3	P800
		TARE	%t	244	xF4	P802

		UNITS	%u	245	xF5	P803
		ZERO	%z	250	xFA	P801



The following models use the international keypad:

- Model 660 (U.K. line cord) P/N: 200660-03100
- Model 660 (Euro line cord) P/N: 200660-04100
- Model 661 (U.K. line cord) P/N: 200661-03100
- Model 661 (Euro line cord) P/N: 200661-04100
- Model 662 (U.K. line cord) P/N: 200662-03120
- Model 662 (Euro line cord) P/N: 200662-04120
- Model 665 VFD (U.K. line cord) P/N: 200665-03110
- Model 665 LCD (U.K. line cord) P/N: 200665-03120
- Model 665 VFD (Euro line cord) P/N: 200665-04110
- Model 665 LCD (Euro line cord) P/N: 200665-04120

INTERNATIONAL CHARACTERS

The 60 Series instruments can display a variety of international characters. Set P411 to a character set as shown in Table 6-4. The language selection can also be changed temporarily without accessing the setup mode using the *%L Language Selection* macro command.

Table 6-4: International Characters

INTERNATIONAL CHARACTER SET			RE-MAPPED CHARACTERS											
		Decimal Character →	35	36	64	91	92	93	94	96	123	124	125	126
		Hex Character →	23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
Selection (P411)	**Name (P411)**	**Description**												
0	USA	United States	#	$	@	[	\	]	^	`	{	|	}	~
1	Frnce	France	#	$	à	°	Ç ç	§	^	`	é	ù	è	¨
2	Gernn	Germany	#	$	§	Ã	Ö	Ü	^	`	ä	ö	ü	ß
3	UK	England	£	$	@	[	\	]	^	`	{	|	}	~
4	Dnmrk	Denmark I	#	$	@	Æ	ø	Â	^	`	æ	ø	å	~
5	Swedn	Sweden	#	¤	É	Ã	Ö	Â	Ü	é	ä	ö	å	ü
6	Italy	Italy	#	$	@	°	\	é	^	ù	à	í	è	Ì ì
7	Spain	Spain I	þ	$	@	¡	Ñ	¿	^	`	¨	ñ	}	~
8	Japan	Japan	#	$	@	[	¥	]	^	`	{	|	}	~
9	Norwy	Norway	#	¤	É	Æ	ø	Â	Ü	é	æ	ø	å	ü
10	Dnmrk2	Denmark II	#	$	É	Æ	ø	Â	Ü	é	æ	ø	å	ü
11	Spn2	Spain II	#	$	á	¡	Ñ	¿	é	`	í	ñ	ó	ú

12	LatAm	Latin America	#	$	á	¡	Ñ	¿	é	ü	í	ñ	ó	ú

RENAMING OPERATING PARAMETERS

0) kg ± ² ³ Bruto

Operating parameters can be permanently renamed in the setup mode or with the *%R Rename Mode* macro command. A renamed parameter will display the new name every time it is accessed. This allows you to customize the standard display for international applications and provides additional prompting capability. Refer to *Renaming Operating Parameters* on page 7-7 for complete details on renaming operating parameters.

PRESETTABLE PARAMETERS

P41@01 PrSET Enbld

In some legal-for-trade applications it is required to identify certain parameters that contain manually entered values rather than automatically calculated values. By enabling the "Preset" parameter (P412), the indicator identifies manually entered parameter values with a "P" preceding the parameter name. Table 6-5 lists the presettable operating parameters.

PRESETTING PARAMETERS

Legal-for-trade installations may require the preset character to be defined on each weight receipt, such as including the text "*PTare = Manual Tare Entry*".

Parameters in Table 6-5 are considered "preset" if their value was not automatically calculated by the 660. Entering values manually, changing values through serial communication, recalling values from a database or copying values from other parameters are examples of actions that will preset a parameter.

The %F macro command can be used to determine if a parameter is preset.

Table 6-5: Presettable Operating Parameters

OPERATING PARAMETER	PARAMETER NAME	DISPLAYED NAME	DEFAULT PRESET NAME
2	Tare	Tare	PTare
3	Gross Total	GrTOT	PGrTO
6	Net Total	NtTOT	PntTO
31	Quantity Total	QtTOT	PQtTO
34	Average Piece Weight	APW	PAPW
35	Average Piece Weight X1000	APW*K	PAPW*

Tare Preset

The preset status for the tare parameter is cleared (not preset) when an auto-tare is performed, when it is cleared by pressing **0 [TARE]**, or after an interruption in power and the tare save parameter (P641) is set for "NoSav".

Gross, Net & Quantity Total Presets

The preset status for the gross total, net total and quantity total parameters is cleared (not preset) when the accumulation total is cleared. Performing an accumulation does not affect the preset status of an accumulation parameter.

Average Piece Weight Preset

The preset status for the average piece weight parameters is cleared (not preset) when the average piece weight is determined through the sampling routine.

CHANGING THE PRESET IDENTIFIER

Presettable parameters should not be renamed with the %R macro command.

The preset parameter identifier can be changed from the default "P" to any other character. If an operating parameter is renamed in the setup mode (see the *Renaming Parameters in the Setup Mode* section beginning 7-7), then the first character of the new name will be used as the preset identifier. For example, if the tare parameter is renamed "MTare" at P602, then MTare will be displayed when the tare parameter is preset. If not preset, Tare is displayed. The first character of the new parameter name is ignored and will not be displayed or printed.

NTEP

The National Type Evaluation Program (NTEP) is a widely accepted weights and measures standard in the United States. Most states abide by some or all of the requirements set forth by NTEP. A complete list of these regulations is available in the "Handbook 44" publication distributed by the National Institute of Standards and Technology (NIST). For more information on this and other NIST publications, visit their web site at http://www.nist.gov.

ENABLING NTEP OPERATION

Legal-for-trade installations using accumulations require the "number of accumulations" parameter (9) to be accessible when not using a printer. When using a printer, this parameter must be printed on the receipt.

In order to configure the indicator to comply with NTEP requirements, the NTEP parameter (P440) must be enabled in the setup mode. This will have the following effects on the standard operation:

- Pressing **[TARE]** with a gross weight of zero (0) or pressing **0 [TARE]** will not automatically switch to the net mode.
- Negative tare values are not accepted regardless of the selection for the "Negative Tare Enable" parameter (P162).
- Tare values are automatically rounded regardless of the selection for the "Tare Rounding Enable" parameter (P163).
- Received serial data will not be processed while in the setup mode until P440 is manually enabled.
- Accumulations using the **. [ENTER]** method can only be performed from the gross, net or quantity mode.
- Printing using the **[PRINT]** key is only possible from the gross, net or quantity mode.

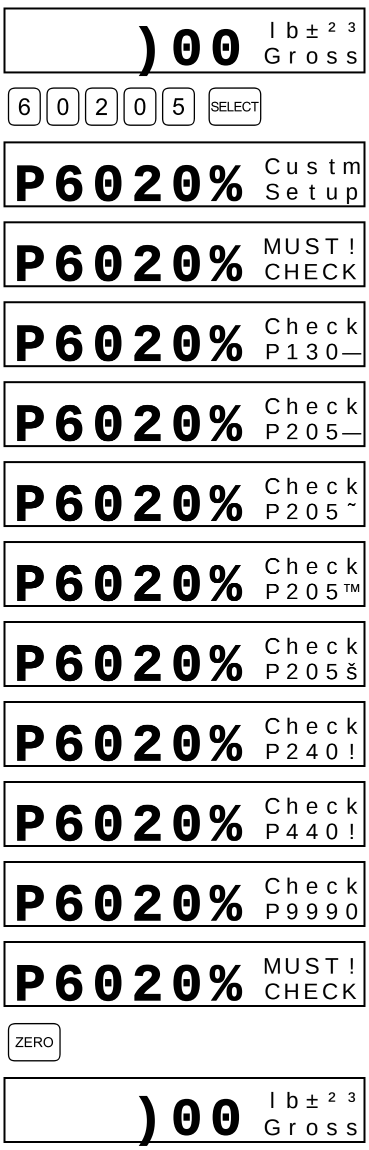
- Weight values that exceed the minimum width specified at P240 will be transmitted as dashes "-------".

NTEP CUSTOM SETUP

The “Custom Setup” parameter, P60205 of the information parameters, displays a list of parameters which, if configured improperly, could facilitate fraud in a legal-for-trade installation. A weights & measures inspector might check this parameter and inquire about the configuration of any parameters that appear in this list.

Accessing the Custom Setup List

Example:
Accessing the NTEP Custom Setup List



DO NOT ATTEMPT TO ACCESS THE CUSTOM SETUP LIST DURING CRITICAL WEIGHT PROCESSING! It is important to note that all functions of the operating mode will be suspended immediately upon accessing the information parameters. This includes suspension of weight conversions, deactivation of all setpoints and cancellation of custom transmits. The “Custom Setup” list may be accessed from the weigh mode as shown in the example – *Accessing the NTEP Custom Setup List*. An access code is not required to view this list.

To access the custom setup list:

1. From the weigh mode, key in **60205 [SELECT]**.
2. The Custom Setup list begins scrolling through each parameter to check. If there are no parameters to check, `Std. Setup` is displayed.
3. The Custom Setup list may be repeated by pressing **[ENTER]** at P60205.
4. Press **[ZERO]** to return to the weigh mode.

Custom Setup Parameters

A setup parameter that appears in the “Custom Setup” does not imply that it is improperly configured. Consider the application and refer to the following descriptions to determine if the parameter is configured appropriately.

P130 – Multi-Range Mode

P130 must be set based on “Gross” to ensure the highest range achieved remains in effect until the gross weight returns to zero.

P205 – Receive Mode

If the receive mode is enabled for any of the four communication ports, any device connected to that port should not be used to transmit data to the indicator which could facilitate fraud.

P205 will appear in the “Custom Setup” list for each receive port enabled. For example, if the receive mode is enabled for all four ports, the list will display **P205—**, **P205˜**, **P205™**, and **P205š**.

P240 – Minimum Transmit Width

A weight value that cannot be displayed due to the 6-digit limitation of the standard VF display may not be printed. To ensure this is not possible, P240 must be set to a width of not greater than 7 (6 digits and a decimal point). NTEP must also be enabled at P440. Any weight value that exceeds the minimum width specified will be printed as dashes "-------".

P440 – NTEP Enable

P440 appears in the "Custom Setup" list if disabled. Refer to *Enabling NTEP Operation* on page 6-8 for possible implications.

P9990 – Macro Instance Selection

P9990 appears in the "Custom Setup" list if at least one macro is configured. Macro operation should be checked to verify its conformance to all regulations.

ADDITIONAL CONFORMANCE CONSIDERATIONS

Several parameters must be considered on an individual basis as their configuration may vary with different applications. These parameters include, but are not limited to those listed in Table 6-6.

Table 6-6: Additional Conformance Parameters

PARAMETER	DESCRIPTION	COMMENT
P110	Full Scale Capacity	Verify proper scale capacity.
P111	Division Size	Verify allowable scale divisions.
P112	Zero Track	Verify required selection.
P114	Motion	Verify required selection.
P118	Zero Range	Verify required selection.
P212	Print Stability	Verify required selection.
P126 → P130	Multi-Range	Verify proper configuration.
P131 → P134	Unit selection	Verify certifiable unit selection. †
P151 → P154	Custom Units	Verify name and conversion factor.
P600 → P646	Rename Parameters	Verify acceptable names.

† Custom units must be site approved. Lb/oz is not approved for legal-for-trade installations.

SEALING AND AUDIT TRAILS

Most legal-for-trade installations will require the indicator to be sealed. A sealed indicator cannot be accessed for setup or calibration changes without breaking a physical seal or incrementing an event counter, thus providing evidence that the unit has been tampered with. Each model in the 60 Series has two types of sealing provisions:

- Physical seal used in conjunction with an internal program jumper
- Three-event audit trail counter

Before applying a wire seal, be sure to move the main board program jumper to the 'NO' position to prevent access to the setup and calibration modes.

Check with your local weights and measures authority to determine which method(s) are required.

PHYSICAL SEAL

The most common sealing method is a lead-wire seal. The 460, 465, 660, 661, 662 and 665 provide two tamper-proof screws used for sealing the rear panel to the front of the enclosure. A lead-wire seal can be applied by passing the lead-wire seal wire through a hole in these two screws, thus preventing the screws from being removed without breaking the seal. The 460, 465, 660, 661, 662 and 665 panel mount versions use a lead-wire seal and one screw. The 663 has a locking clasp on the front door to which the seal can be applied, or the indicator can be sealed using the same method as with the panel mount versions.

See Figure 6-1 through Figure 6-6 for seal locations.

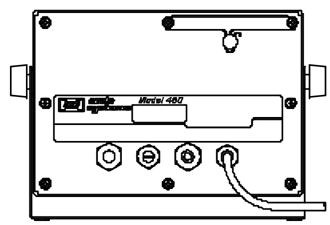


Figure 6-1: 460 Legal-for-Trade Seal

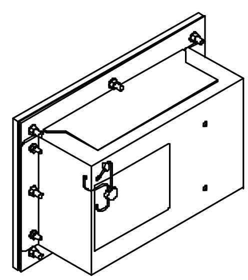


Figure 6-2: 460 Panel Mount Legal-for-Trade Seal

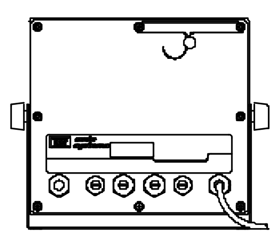


Figure 6-3: 465, 660, 661 and 662 Legal-for-Trade Seal

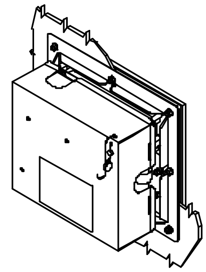


Figure 6-4: 465, 660, 661, 662, and 665 Panel Mount Legal-for-Trade Seal

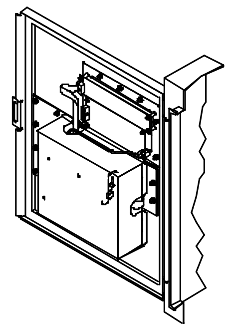


Figure 6-5: 663 Legal-for-Trade Seal

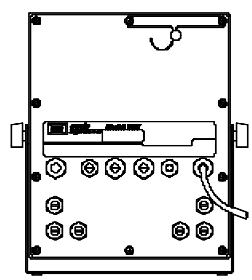


Figure 6-6: 665 Legal-for-Trade Seal

AUDIT TRAIL PARAMETERS

Three separate incrementing, non-resettable audit trail parameters are used to indicate changes to various parameters:

- P60201 – OIML
- P60203 – Calibration
- P60204 – Setup

⚠ **DO NOT ATTEMPT TO ACCESS AUDIT TRAIL PARAMETERS DURING CRITICAL WEIGHT PROCESSING! Weight conversions and custom transmits will be suspended and all setpoints will be deactivated!**

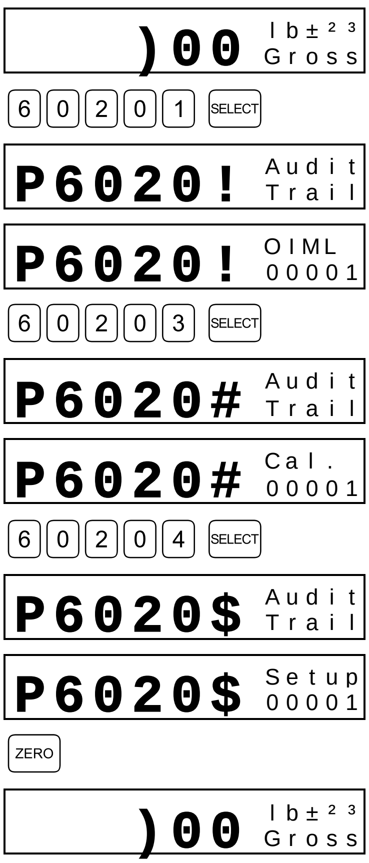
An audit trail counter will increment only once upon exiting the setup mode and saving changes regardless of how many settings were changed. Each audit trail counter will increment to 99999 before beginning again at 00001.

Accessing Audit Trails

The audit trails may be accessed from the weigh mode as shown in example – *Accessing Audit Trail Parameters*. An access code is not required to view audit trail parameters.

1. Key in **60201 [SELECT]** to access the OIML audit trail.
2. Key in **60203 [SELECT]** to view the Calibration audit trail.
3. Key in **60204 [SELECT]** to view the Setup audit trail.
4. Press **[ZERO]** to return to the weigh mode.

Example:
Accessing Audit Trail Parameters



OIML Audit Trail

Changes to any of the following parameters will increment the OIML audit trail at P60201:

- P109 → P119 Scale Setup
- P122 Return to Zero
- P131 → P134 Units
- P150 → P154 Calibration & Custom Units
- P162, P163 Negative Tare Enable, Tare Rounding Enable
- P300 → P309 Selectable Modes
- P410, P412 OIML Enable, Preset Enable
- P420 Standard VF Display Mode
- P600 → P646 Rename Parameters
- P800 → P820 Key Functions
- P989 → P4999 Custom Transmit
- P61101 → P61140 Calibration & Linearization
- P65001, P65002 Default All, Default –Cal

Calibration Audit Trail

Any changes to the existing calibration will increment the Calibration audit trail at P60203. This includes any changes to P60101 → P61140 of the information parameters.

Setup Audit Trail

Changes to any of the setup mode parameters will increment the Setup audit trail at P60204.

DATA STORAGE DEVICE (DSD)

The DSD feature of the 560 and660 Series controllers provides a means of recording weight data in a secure database structure that ensures data integrity. The database option is required for DSD usage. Information can only be written to the database manually via the DSD menu or through a serial command. Standard macro database commands do not apply to the DSD database. Once written, stored data cannot be updated. Nor can data rows be individually deleted at will. A checksum is stored with each data row and verified each time the row is accessed to further ensure data integrity.

DSD data can be printed via custom transmit at the time it is stored, printed in report format or downloaded in a comma-delimited format.

DATABASE STRUCTURE

The DSD database is a fixed structure using the data columns shown below. Operation is completely independent of existing database routines. When a row is accessed, data values are recalled into parameters 64.1P → 64.9P. These parameters can be used in macro commands, but cannot be assign values. They are also the only parameters permissible in the DSD custom transmit specified at P593.

ID# (64.1P)	**SCALE #** (64.2P)	**GROSS** (64.3P)	**NET** (64.4P)	**TARE** (64.5P)	**GROSS TOTAL** (64.6P)	**NET TOTAL** (64.7P)	**# OF ACCUMS** (64.8P)	**TIME / DATE** (64.9P)
000001	1	101.2 kg	101.2 kg	0.0 kg	101.2 kg	101.2 kg	1	14:03:32 14/03/01
000002	1	150.1 kg	100.1 kg	50.0 kg	251.3 kg	201.3 kg	2	14:03:58 14/03/01
000003	2	10030 kg	9900 kg	130 kg	0 kg	0 kg	0	15:24:07 14/03/01
↓								
999999	1	10.5 kg	0.5 kg	10.0 kg	10.5 kg	0.5 kg	1	08:43:30 11/04/01

P64.1: ID#

Serves as a unique "lookup" value for recalling rows in the DSD database. It is automatically incremented and stored with each new row. The ID# can not be reset or preset by any means. It begins at '1' and will increment to a value of 999,999. When this value is exceeded, the value returns to zero (0) and continues incrementing as before.

P64.2: SCALE#

Identifies which scale was considered when storing the data values for a given row.

P64.3: GROSS

Represents the gross weight for the specified SCALE# at the time the data row was created.

P64.4: NET

Represents the net weight for the specified SCALE# at the time the data row was created.

P64.5: TARE

Represents the tare weight for the specified SCALE# at the time the data row was created.

P64.6: GROSS TOTAL

Represents the accumulated gross total for the specified SCALE# at the time the data row was created.

P64.7: NET TOTAL

Represents the accumulated net total for the specified SCALE# at the time the data row was created.

P64.8: # OF ACCUMS

Represents the number of times an accumulation was performed to derive the gross and net totals reported in the previous columns.

P64.9: TIME / DATE

Represents the time and date the data row was created.

Units of Measure

The DSD database weight data is stored in the default calibration units specified at P150. If this parameter where changed, it would corrupt the display of the database data. Therefore, P150 cannot be changed if the DSD is enabled and data rows exist. Trying to do so will generate an error message.

Memory Allocation

The DSD database requires the use of the 60 Series database option. The maximum amount of memory that can be allocated will depend on the total amount of database option memory and how much is allocated to other functions. Allocation of this memory requires that this memory exist and has been initialized.

The amount of memory to allocate for DSD must be specified in terms of a number of data rows at P594. The amount allocated can be changed, but any stored DSD data will be cleared.

A maximum of 99,999 rows can be specified. If more rows than can be assigned are requested, the maximum available will be created.

The ID# starts at 1. Its max value is 999999, at which point it rolls over to 0. This number is stored in the database memory.

If database boards are swapped, this will be detected by the indicator serial number copied into the database initialization not matching that of the new indicator. At this point new data records cannot be generated. A warning message is generated at power up or when trying to create a row, indicating this condition. You can print/download and then clear the database, at which point the serial number will be updated and all functionality will be restored. A warning message is generated at the beginning of the print/download indicating this condition. Note that if no data records exist, the serial number field will be automatically updated to the current value.

The database also contains a count of how many columns are defined for the database. If this number does not match what the indicator expects, then no functionality of the database is available. This would be expected if new data columns were added in the future and the database being used did not match the firmware being used. This problem requires placing the database into a indicator with the appropriate firmware, or initializing the database memory (P65001 or P65010) or disabling the database. Note that if no data records exist, the storage will automatically be released and reallocated, using the current indicators database structure.

Disabling the DSD database (P590) causes all storage allocated for the database to be released, meaning the data is lost.

DATA INTEGRITY

Each DSD database row includes a checksum for data integrity. The checksum will be verified each time a row is accessed. If the accessed row fails it checksum test, a error message will be displayed. When printing or downloading, the next line sent will be an error message.

SETUP PARAMETERS

P590: DSD Enable

Enables the Data Storage Device feature (DSD) and provides access to the other DSD parameters (P591 → P595).

When enabling or disabling DSD, you will be prompted to clear the DSD database records before the change is allowed. Be sure to download any stored data before proceeding.

Enabling DSD will override P806 to redefine the **[ID]** key to invoke the DSD Menu. It will also redefine P205 as the DSD receive mode for the specified DSD communication port.

P591: DSD Serial Port

Selects the communication port to be used for DSD transmissions.

The usage of the DSD port selected at P591 can be temporarily overridden by usage of the %H macro command. If the selection is turned off, then no DSD transmits or receives will occur. If the port number is changed, then the new port will be used.

Nothing prevents other indicator transmissions from being sent over the DSD port. No other processing of received data will occur on this port.

Note that if a comm port selected is programmed as receive disabled, selecting it does not turn the port on, no data will be received.

P592: DSD Receive Character

Specifies a single character used to create a row in the DSD database when received on the DSD communication port.

P593: DSD Custom Transmit

Specifies a custom transmit used to automatically transmit DSD data after a row has been created in the DSD database. The custom transmit specified will not allow non-DSD parameter entries. Transmission will be motion delayed by virtue of the stored data row. The custom transmit communication port can be specified at P991.

P594: DSD Maximum Number of Rows

Specifies the maximum number of DSD data rows that can be stored in the database. An attempt to store a record in a full database will result in a 1 second OVER-WRITE warning message indicating that the oldest record will be deleted before storing the new data row.

P595: DSD Number of Warning Rows

Specifies the number of unused rows at which point a 1 second warning message will be displayed. For example, if the maximum number of rows is 1000 and the number of warning rows is 100, then a warning message will be displayed for every data row stored after the 900th record. The maximum number of warning rows is 999.

DSD FUNCTION SELECTION MENU

The various functions of the DSD database are manually accessible via a Selection Menu. This menu is accessible after enabling DSD (P590). Once enabled, pressing **[ID]** will display the first DSD menu selection. Pressing **[SELECT]** / **[MODE]** will advance to the next menu selection. Pressing **[ENTER]** will invoke the displayed menu selection routine. The display will revert to the weigh mode after completing a selected function. Pressing **[ID]** or **[ZERO]** / **[>0<]** will exit the DSD menu.

```
10!2       kg
           Gross
```

[ID]

```
      Dsd  Make
           Row
```

[SELECT] / [MODE]

```
      Dsd  Print
           dbase
```

[SELECT] / [MODE]

```
      Dsd  Down-
           Load
```

[SELECT] / [MODE]

```
      Dsd  View
           Data
```

[SELECT] / [MODE]

```
      Dsd  Make
           Row
```

[ID]

10!2 Kg Gross

MAKE ROW

Waits for motion delay (displays `Mot'n Delay` while in motion), then creates a new sequential row in the DSD database.

Pressing **[CLR]** will abort from the `Mot'n Delay` prompt without storing or printing data.

If a DSD custom transmit is specified, the transmission will occur immediately after the creation of a row in the DSD database.

The system then reverts to the gross weight display.

Creating DSD Database rows

The stored information will be based upon the current scale which is selected (i.e. for gross weight,…).

If the number of unused rows is less than the warning threshold, a warning will be displayed on the screen for one second.

If the database is full, the oldest record will be overwritten. A one second error message will be displayed on the screen.

Requests to create rows will be delayed in processing if another function (make, print, download) is using the database. Printing or downloading will block row creation until the user decides whether to clear the database or not.

Clearing Data

Data rows can only be cleared after performing a PRINT or DOWNLOAD operation. The data will be sent out the defined DSD port. If the port is not setup or has been turned off by the %H macro command, then this operation will fail. Clearing rows will clear all stored DSD data, however it will not reset the ID# to be used for the next created row.

Printing or downloading will block row creation until the user decides whether or not to clear the database.

Performing a Default All (P65001), Database Reset (P65010) or an Operational RAM test (P62001) will also result in all data rows being cleared. Performing a Default All will also reset the ID#.

PRINT DBASE

Prints the entire contents of the DSD database in row/column format. Data is sent out the port specified as the DSD printer port in the setup mode.

```
ID# Scale#     Gross           Net            Tare       Gross Total        Net Total    #Accum   Time        Date
200      1    0.43 lb       0.43 lb       0.00  lb         0.00  lb         0.00  lb         0   03:26:22pm 10/02/01
```

```
201      1    0.43 lb       0.43 lb       0.00  lb         0.00  lb         0.00  lb          0   03:26:23pm 10/02/01
202      1    0.43 lb       0.43 lb       0.00  lb         0.00  lb         0.00  lb          0   03:26:24pm 10/02/01
203      1    0.43 lb       0.43 lb       0.00  lb         0.00  lb         0.00  lb          0   03:26:25pm 10/02/01
204      1    0.43 lb       0.33 lb       0.10p lb       100.44p lb       100.24p lb          1   03:32:19pm 10/02/01

204      1    0.43 lb       0.33 lb       0.10p lb       100.44p lb       100.24p lb          1   15:32:19   02/10/01
204      1     --- lb        --- lb        ---p lb          ---p lb          ---p lb          1   15:32:19   02/10/01
```

After printing, a Y/N prompt will query the operator to clear the database. The system then reverts to the gross weight display.

Data Formatting

Time & date formatting look at P503 and P504 to determine whether to use 24hr or am/pm time, and US or international date.

Weight units are presented as the default calibration units specified at P150.

Weights are by default formatted to a field width of 7, total weights at a width of 9. Values requiring more width expand as needed. If NTEP (P440) is enabled, then P240 is used for assigning field widths. Values requiring more width are instead printed as dashes.

Preset data (manually entered tare, gross total and net total values) are identified with a ‘p’ immediately following the weight value.

DOWNLOAD DBASE

Transmits the entire contents of the DSD database in comma-delimited ASCII text format. Data is sent out the port specified as the DSD PC port in the setup mode.

```
200,1,0.43 lb,0.43 lb,0.00  lb,0.00  lb,0.00  lb,0,03:26:22pm,10/02/01
201,1,0.43 lb,0.43 lb,0.00  lb,0.00  lb,0.00  lb,0,03:26:23pm,10/02/01
202,1,0.43 lb,0.43 lb,0.00  lb,0.00  lb,0.00  lb,0,03:26:24pm,10/02/01
203,1,0.43 lb,0.43 lb,0.00  lb,0.00  lb,0.00  lb,0,03:26:25pm,10/02/01
204,1,0.43 lb,0.33 lb,0.10p lb,100.44p lb,100.24p lb,1,03:32:19pm,10/02/01

204,1,0.43 lb,0.33 lb,0.10p lb,100.44p lb,100.24p lb,1,15:32:19,02/10/01
204,1,--- lb,--- lb,---p lb,---p lb,---p lb,1,15:32:19,02/10/01
```

After downloading, a Y/N prompt will query the operator to clear the database.

The system then reverts to the gross weight display.

Data Formatting

Fields are compressed to use only as much width as is required. If NTEP is enabled, then the same limits are imposed as for printing.

The number of decimal places used (for weight values) is as setup for each scale.

Time & date formatting look at P503 and P504 to determine whether to use 24hr or am/pm time, and US or international date.

Weight units are presented as the default calibration units specified at P150.

Preset data (manually entered tare, gross total and net total values) are identified with a ‘p’ immediately following the weight value.

VIEW DATA

Puts the indicator into a view mode for reviewing stored data on the display. If an entry is made prior to pressing **[ENTER]**, the indicator searches for the entered ID#. If found, the ID# is displayed, otherwise NOT FOUND is displayed and the display reverts to the "view data" selection of the DSD menu.

- If **[ENTER]** is pressed without an entry, the most recently created ID# is recalled and displayed.
- Once an ID# is displayed, the arrow keys can be used to scroll through rows/columns of the DSD database.
- The right/left arrows will scroll left/right to adjacent columns within the current row.
- The up arrow will decrement the ID# for the currently displayed column.
- The down arrow will increment the ID#.
- While incrementing/decrementing, the current row's ID# is displayed.

```
105  DSD
     ID#
```

- The ID# will 'wrap' to the beginning/end of the database if the last/first row is exceeded when pressing the up/down arrows.
- Keying in an ID# and pressing **[ID]** will recall that ID#. If the ID# does not exist, NOT FOUND is displayed and the current ID# remains in effect.
- Pressing **[ID]** will display the ID# of the current row for one second, then revert back to the previously displayed data.
- Pressing **[SELECT]** / **[MODE]** exits the VIEW DATA mode and returns to the gross weight mode.
- Pressing **[PRINT]** will print the DSD custom transmit for the data in the currently viewed row.

In the view mode, the prompting display will show the units of measure on the top line and the bottom line will show the letters below to identify the displayed data:

"ds Sc"	(scale#)
"ds G"	(gross)
"ds N"	(net)
"ds T"	(tare)
"ds GT"	(gross total)
"ds NT"	(net total)
"ds Ac"	(number of accumulations)
"ds Tm"	(time)
"ds Dt"	(date)

```
10280   kg
        ds  T
```

Preset (manually entered) data will be identified by a ‘p’ in the mode description (i.e. ds**p**T, ds**p**GT, ds**p**NT, etc.).

```
10280   kg
        dspT
```

DSD Custom Transmit

The defined DSD custom transmit only accepts data from the DSD parameters (P64.1 → P64.9). Other parameter information added to the custom transmit will result in an error message when trying to exit setup. This transmit will occur after creating a database row.

DSD Communication Port

The usage of the DSD port selected at P591 can be temporarily overridden by usage of the %H macro command. If a DSD port is not specified, then no DSD transmits or receives will occur. If the port number is changed, then the new port will be used.

Nothing prevents other indicator transmissions from being sent over the DSD port. No other processing of received data will occur on this port.

Note that if a comm port selected is programmed as receive disabled, selecting it does not turn the port on, no data will be received.

DSD Macro Commands

Several macro commands have been added to work with the DSD database. These commands are more fully explained in chapter 9.

%H Macro

The %H macro command has been expanded to allow changing the DSD port selection. This change is temporary, and will be lost on power-up or if the setup mode is entered and saved.

Expanded Syntax: **591,<comm>%H**

The DSD function temporarily overrides whatever other receive function was setup for use of the port (P205). When the DSD functionality is moved to another port, the previous behavior is restored.

Note that if a comm port is programmed as receive disabled at P205, then it will not be possible to use the %H macro command to turn the comm port on. No data will be received.

Use of this macro command to change the operation of a comm port that is in use by DSD does not take control away from DSD. If DSD is then moved to another port, this previous selection would then begin operation.

%f MACRO

The %f macro command is used to get the preset information for the current database row stored in P64. Requesting P64.5 (tare), P64.6 (gross total), or P64.7 (net total) is now supported.

%w MACRO

The %w macro command can be used to access DSD database rows. The possible commands are:

`I[id number]%w`	Retrieves the DSD database row with the specified id number.
`N%w`	Returns the number of existing data rows present in the DSD database (not the maximum number of rows).
`L%w`	Returns the lowest id number present in the database.
`H%w`	Returns the highest id number present in the database.

Chapter 7 OPERATING PARAMETERS

OVERVIEW

OPERATING PARAMETERS

The 60 Series instruments use many system defined memory registers to store operational data. Most of these registers, or parameters, may be accessed from the weigh mode to monitor or change the scale's operation. The gross, net and tare registers are examples of operating parameters. In the setup mode, operating parameters can be used as conditions for setpoints, define analog outputs, define database columns, and serve countless functions when used in macros. Operating parameters are also important elements of communications. They may be formatted to transmit values from within a macro or a custom transmit table. Many parameters can receive values through direct serial communication, input interpreters or Modbus communication.

OPERATING PARAMETER IDENTIFICATION

Most operating parameters are accessible directly from the weigh mode. Whereas setup parameters are numbered P108 → P50001, operating parameters are numbered 0 → 99. To further distinguish between setup parameters and operating parameters, this manual identifies setup parameters with a preceding "P" (i.e. **P108** = full scale capacity) and operating parameters with a trailing "P" (i.e. **11P** = time & date).

Refer to Table 7-1 for a list of all 60 Series operating parameters. Note that some parameters, such as counting, are not available or usable without first configuring certain setup parameters. A few others cannot be displayed in the weigh mode and are intended only for macro operations and/or communication functions.

PARAMETER INSTANCES

Many parameters require an *instance* to be specified when accessing their values. For example, we can refer to the gross weight (parameter 0). However, if multiple scales are enabled each scale will have its own gross weight. An instance allows you to select or specify a particular scale, Programmable Digital I/O channel, timer value, or variable.

Specifying An Instance

To reference a parameter instance, include the instance number immediately after the parameter number, separating each with a decimal point. For example:

0.1P refers to the gross weight for scale #1

0.2P refers to the gross weight for scale #2

0.3P refers to the gross weight for scale #3

0.4P refers to the gross weight for scale #4

Likewise,

76.200P refers to the countdown timer for setpoint #200

80.10P refers to variable #10

81.2P refers to the elapsed ticks for independent timer #2

Instance '0'

All weight parameters (i.e. gross, net, tare, etc.) can be specified with an instance of zero '0'. This instance refers to the "current" scale – the weight for the scale presently displayed. For example:

For weight parameters, instance '0' refers to the weight of the scale currently selected.

1.0P refers to the net weight for the current scale

For single scale applications, the current scale will always be scale #1. Therefore **1.0P** will always report the same net weight as **1.1P**. In multiple scale applications, the current scale may be scale #1, or scale #2. Thus if **[SCALE SELECT]** was pressed to view the net weight of scale #2, then **1.0P** will report the same net weight as **1.2P**.

Using the current scale instance '0' is useful when specifying weight parameters in a custom transmit table. This allows you to use a single custom transmit to print a ticket showing weights only for the currently selected scale. Other parameters such as time/date, APW and total of all scales, have only a '0' instance. For these single-instance parameters, instance '0' does *not* refer to the current scale. In fact, specifying an instance is generally not necessary. For example:

11P refers to the time/date parameter

Since there is only one time/date you do not need to specify an instance. One exception is when formatting single-instance parameters in macros. Here, the instance is a required argument in the parameter syntax. For example:

11.0.18561P refers to the time/date formatted with a 4-digit year

Without the instance delimiter, the macro would attempt to interpret this parameter as 11.18561P, or time/date with an instance of 18561 and an error would occur.

ACCESSING OPERATING PARAMETERS

All operating parameters are accessible from the weigh mode, except those listed as "not displayable" in Table 7-1.

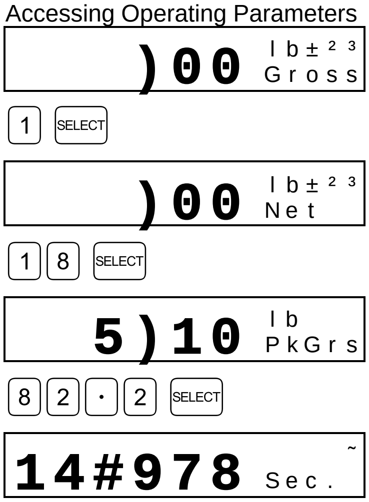
Table 7-1: Operating Parameters

OPERATING PARAMETER	PARAMETER NAME	DISPLAYED NAME (WEIGH MODE)	RENAME PARAMETER	VALID INSTANCE			COMMENTS
				460 SERIES	560 SERIES	660 SERIES	
0	Gross	Gross	P600	0 – 2	0 – 4	0 – 8	
1	Net	Net	P601	0 – 2	0 – 4	0 – 8	
2	Tare	Tare	P602	0 – 2	0 – 4	0 – 8	
3	Gross Total	GrTOT	P603	0 – 2	0 – 4	0 – 8	
4	Gross Total + Current Gross	GrT+C	P604	0 – 2	0 – 4	0 – 8	
5	Gross Total – Current Gross	GrT-C	P605	0 – 2	0 – 4	0 – 8	
6	Net Total	NtTOT	P606	0 – 2	0 – 4	0 – 8	
7	Net Total + Current Net	NtT+C	P607	0 – 2	0 – 4	0 – 8	
8	Net Total – Current Net	NtT-C	P608	0 – 2	0 – 4	0 – 8	
9	Number of Accumulations	# Accum	P609	0 – 2	0 – 4	0 – 8	
10	Scale	(not displayable)	P610	0	0	0	
11	Time & Date	(shows time/date)	P611	0	0	0	

15	Average Gross	AvGrs	P615	0 – 2	0 – 4	0 – 8	Updated by the %+ macro command
16	Average Net	AvNet	P616	0 – 2	0 – 4	0 – 8	
17	Average Count	AvgCt	P617	0 – 2	0 – 4	0 – 8	
18	Peak Gross	PkGrs	P618	0 – 2	0 – 4	0 – 8	
19	Peak Net	PkNet	P619	0 – 2	0 – 4	0 – 8	
20	Rounded (Displayed) Gross	(not displayable)	P620	0 – 2	0 – 4	0 – 8	
21	Rounded (Displayed) Net	(not displayable)	P621	0 – 2	0 – 4	0 – 8	
23	Rate	Rate	P623	0 – 2	0 – 4	0 – 8	Must specify Rate Measurement Period (RMP) at P135 to utilize rate-based parameters
24	Free Fall 1	FreFl	P624	0 – 2	0 – 4	0 – 8	
25	Future Gross 1	FutGr	P625	0 – 2	0 – 4	0 – 8	
26	Future Net 1	FutNt	P626	0 – 2	0 – 4	0 – 8	
27	Free Fall 2	FrFl2	P627	0 – 2	0 – 4	0 – 8	
28	Future Gross 2	FuGr2	P628	0 – 2	0 – 4	0 – 8	
29	Future Net 2	FuNt2	P629	0 – 2	0 – 4	0 – 8	
30	Quantity	Qty	P630	0 – 2	0 – 4	0 – 8	Must enable Count feature at P179 to access counting parameters
31	Quantity Total	QtTOT	P631	0 – 2	0 – 4	0 – 8	
32	Quantity Total + Current Quantity	QtT+C	P632	0 – 2	0 – 4	0 – 8	
33	Quantity Total - Current Quantity	QtT-C	P633	0 – 2	0 – 4	0 – 8	
34	Average Piece Weight	APW	P634	0	0	0	
35	Average Piece Weight * 1000	APW*K	P635	0	0	0	
36	Percent Accuracy	%Accy	P636	0	0	0	
37	Last Sample Size	Last Sampl	P637	0	0	0	
40	Gross Total of All Scales	GrAll	P640	0	0	0	
41	Net Total of All Scales	NeAll	P641	0	0	0	
42	Tare Total of All Scales	TrAll	P642	0	0	0	
43	Total of All Gross Totals	GTAll	P643	0	0	0	
44	Total of All Net Totals	NTAll	P644	0	0	0	
45	Quantity Total of All Scales	QuAll	P645	0	0	0	Must enable Count feature at P179
46	Total of All Quantity Totals	QTAll	P646	0	0	0	
50	Programmable I/O Function	PIOA	P852	N/A	N/A	1 – 8	Dependent upon function at P851
51	Programmable I/O Function	PIOB	P853	N/A	N/A	1 – 8	
52	Programmable I/O Function	PIOC	P854	N/A	N/A	1 – 8	
60	Extended Resolution Gross	XGros	-	0 – 2	0 – 4	0 – 8	
61	Extended Resolution Net	XNet	-	0 – 2	0 – 4	0 – 8	
62	Extended Resolution Tare	XTare	-	0 – 2	0 – 4	0 – 8	
63	A/D Conversion Number	#A/Dx	-	0 – 2	0 – 8	0 – 8	
64	DSD Parameter	DSD	-	N/A	1 – 9	1 – 9	See page 6-13
76	Setpoint Countdown Timer	(not displayable)	-	1 – 16	1 – 48	1 – 256	
77	Setpoint Delay Timer	(not displayable)	-	1 – 16	1 – 48	1 – 256	
78	Setpoint Status	SPxxx	-	1 – 16	1 – 48	1 – 256	
79	Random Number	(not displayable)	-	0	0	0	
80	Variable	V#xxx	P862	1 – 15	1 – 100	1 – 999	Allocate at P680
81	Timer Ticks	Ticks	-	1 – 8	1 – 8	1 – 8	
82	Timer Seconds	Sec.	-	1 – 8	1 – 8	1 – 8	
90	Macro Select	(macro name)	P9991	1 – 15	1 – 100	1 – 250	
91	User Defined Weigh Mode Message	Mode 91Msg	-	pg 7-36	pg 7-36	pg 7-36	
92	Get Display Data	(not displayable)	-	0 or 10	0 or 10	0 or 10	
97	Status	(not displayable)	-	0 – 8	0 – 4	0 – 2	
98	Current Displayed Weight/Count	(not displayable)	-	0 – 2	0 – 4	0 – 8	
99	Extended Gross	Extnd	-	Diagnostic	Diagnostic	Diagnostic	

DIRECT ACCESS

Example:
Accessing Operating Parameters



Operating parameters can be accessed directly from the weigh mode as shown in the example – *Accessing Operating Parameters* by keying in the desired parameter number, then pressing **[SELECT]**. For example:

1 [SELECT] selects the net mode

18 [SELECT] selects the peak gross mode

If an instance is required, include a decimal point and the instance after the parameter number. The instance number will be shown inverted in the upper-right corner of the display. For example:

82.2 [SELECT] selects the timer seconds for timer #2

Accessing Weight Parameters

It is not necessary to specify an instance when selecting a single-instance parameter or a weight parameter. Specifying an instance for a weight parameter is only necessary when more than one scale is enabled and you want to change the currently selected scale. For example, if the gross weight for scale #2 is currently displayed, you can key in **1 [SELECT]** to view the net weight for scale #2. You could also have keyed in **1.2 [SELECT]** (or even **1.0 [SELECT]**), but the instance is not necessary since scale #2 is already the current scale. To view the gross weight for scale #1, key in **0.1 [SELECT]**. Note that it was not necessary to press **[SCALE SELECT]** to switch from scale #2 to scale #1. Specifying an instance of '1' selected the desired scale automatically.

ACCESSING VARIABLES

Variables must be allocated at P680 before they can be accessed.

Variables may be accessed in the same method as other parameters:

80.1 [SELECT] selects variable #1

80.20 [SELECT] selects variable #20

Variables may also be accessed using the **[ID]** key if it is not redefined at P806:

1 [ID] selects variable #1

20 [ID] selects variable #20

This method requires fewer keystrokes as only the instance number must be keyed in prior to pressing **[ID]**.

INVALID INSTANCE

The error message Invld Instn will be displayed whenever an invalid instance is specified. For example, if scale #3 is not enabled and you key in **0.3 [SELECT]** to access the gross weight for scale #3, Invld Instn is briefly displayed indicating that the selection was not valid. The gross mode *will* be selected. However, the instance displayed will be the nearest previous valid instance.

If an invalid parameter is specified, Invld Instn is also displayed indicating that the selection was not valid. The parameter displayed will be the nearest previous valid parameter. For example, if the counting mode is disabled and you key in **37 [SELECT]** to access the last sample size, Invld Instn is briefly displayed indicating that the selection was not valid. This is because the counting mode must be enabled to access 30P → 37P as indicated in Table 7-1. The instance displayed will be the nearest previous valid instance, in this case 29P, future net #2. If an instance is specified for a parameter that does not require and instance, the message `No Instn` is displayed.

Displayable parameters 50P → 82P *require* an instance. Failure to specify one will result in the automatic selection of the first instance for the selected parameter.

MODE MENU ACCESS

A convenient way to access frequently used operating parameters is to assign them to the Mode Menu parameters at P300 → P309 of the setup mode. This allows you to toggle through up to ten different operating parameters in the weigh mode using only the **[SELECT]** key. The first parameter in the Mode Menu list will be the first parameter displayed at power-up or upon exiting the setup mode.

By default, the gross weight is assigned to P300, the net weight to P301 and the tare weight to P302. Thus, the instrument powers-up in the gross mode and pressing **[SELECT]** toggles the weigh mode to net, then to tare, and then back to gross.

EDITING THE MODE MENU SELECTIONS

The following procedure describes how to edit the Mode Menu selections. In this example, the default configuration is altered for an application performing gross weight accumulations. The time/date will be displayed at power-up. Pressing **[SELECT]** will toggle through gross, gross total, gross total + current gross, number of accumulations, then back to time/date.

To edit the Mode Menu selections:

1. Access the setup mode.
2. Key in **300 [SELECT]** to access the first menu selection (mode 0).
3. Key in **11 [ENTER]** to select the time/date parameter.
4. Press **[SELECT]** to access the next menu selection at P301 (mode 1).
5. Key in **0.0 [ENTER]** to enter the gross parameter for the current scale.
6. Press **[SELECT]** to access the next menu selection at P302 (mode 2).
7. Key in **3.0 [ENTER]** to enter the gross total parameter for the current scale.
8. Press **[SELECT]** to access the next menu selection at P303 (mode 3).
9. Key in **4.0 [ENTER]** to select the gross total + current gross weight parameter for the current scale.
10. Press **[SELECT]** to access the next menu selection at P304 (mode 4).
11. Key in **9.0 [ENTER]** to select the number of accumulations parameter for the current scale.
12. Save changes and exit the setup mode.

Refer to the *Parameter Entry Parameters* section more details on entering parameters in the setup mode.

Example:
Editing the Mode Menu Parameters

[3] [0] [0] [SELECT]

P30) MODE0 Gross

[1] [1] [ENTER YES]

P30) MODE0 Tm/Dt

[SELECT]

P30! MODE1 Net

[0] [.] [0] [ENTER YES]

P30! MODE1 Gross

[SELECT]

P30@ MODE2 Tare

[3] [.] [0] [ENTER YES]

P30@ MODE2 GrTOT

[SELECT]

P30# MODE3 None!

[4] [.] [0] [ENTER YES]

P30# MODE3 GrT+C

[SELECT]

P30$ MODE4 None!

[9] [.] [0] [ENTER YES]

P30$ MODE4 Accum

[ZERO] [CLR NO] [ENTER YES] [ENTER YES]

Exits the setup mode, saves changes and returns to the weigh mode showing time/date as the new power-up mode.

0*0!99 12:00 :00am

Press [SELECT] to toggle through the new Mode Menu parameters.

To clear an existing parameter in the Mode Menu, access the desired Mode setup parameter and key in 99 [ENTER]. The display will then show None! for the parameter selection.

The %s and %i macro commands perform the same functions as pressing [SELECT] and [ID] respectively.

MACRO ACCESS

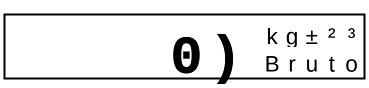
Macros can access operating parameters by duplicating the keystrokes that would be required using the direct access method. For example:

1%s	selects the net mode for the current scale
1.0%s	selects the net mode for the current scale
1.2%s	selects the net mode for scale #2
11%s	selects time/date
80.1%s	selects variable #1
1%i	selects variable #1
%s	selects the next parameter in the Mode Menu

The following series of macro commands demonstrates how to briefly display the tare weight for scales 1 → 3, then return to the gross mode for scale #1.

2.1%s	selects the tare mode for scale #1
%P	pause (one second)
2.2%s	selects the tare mode for scale #2
%P	pause (one second)
2.3%s	selects the tare mode for scale #3
%P	pause (one second)
0.1%s	selects the gross mode for scale #1

RENAMING OPERATING PARAMETERS



Operating parameters can be permanently renamed in the setup mode or with the *%R Rename Mode* macro command. A renamed parameter will display the new name every time it is accessed. This allows you to customize the standard display for international applications and provides additional prompting capability.

RENAMING PARAMETERS IN THE SETUP MODE

Rather than displaying Gross, Net and Tare, you can permanently rename these parameters to become the Spanish names Bruto, Neto and Tara at P600, P601 and P602 respectively. Likewise, other operating parameters can be renamed in the setup mode at P600 → P646 (see Table 7-1). Note that the last two digits of the setup mode parameter correspond with the operating parameter number.

Example:
Renaming Operating Parameters
[6] [0] [0] [SELECT]

P60) Gross None!

Bruto [ENTER YES]

P60) Gross Bruto

To rename an Operating Parameter in the setup mode:

1. Access the setup mode.
2. Key in the desired parameter number, 600 → 646 as shown in Table 7-1, and press **[SELECT]** to access the "Rename" parameter.
3. Key in the new name and press **[ENTER]**. Alpha characters may be entered through the front panel as described in the *Key In Value Parameters* section.
4. Repeat steps 2 and 3 to rename additional parameters.
5. Exit the setup mode.

RESTORING DEFAULT PARAMETER NAMES

The default names for all operating parameters are restored after defaulting all setup parameters (see the *Default Setup* section of the *Information Parameters* chapter). To restore the default name for individual parameters, access the desired "Rename" parameter as previously described and press **[CLR]** rather than entering a new name. The prompt will show None!, indicating that the parameter is no longer renamed.

RENAMING PARAMETERS WITH MACROS

Operating parameters can be renamed with the *%R Rename Mode* macro command. This allows parameters to be renamed at any time without accessing the setup mode. The new name will remain in effect until changed again by another %R command. If the changes to the setup mode are saved when a new name is in effect, the new name will become permanent, retaining the name even after a power loss.

Using this method of renaming parameters, you could develop a macro routine that prompts the user to select a language at power-up, then renames parameters accordingly. Parameters could also be renamed to serve as prompts. For example, a 2-speed filling application could rename Net to be Fast, then Slow, and finally Done! at the appropriate times to indicate system status. Refer to the *%R Rename Mode* macro command for full details and examples.

WEIGHT PARAMETERS

The basic weight parameters are Gross, Net and Tare. A separate Gross, Net and Tare register is maintained for each enabled scale. These values are recalculated after every A/D interrupt, generally 60 times per second.

GROSS (MODE 0)

2!25 lb Gross

The *gross* weight parameter represents the total live weight on the scale since the last time a zero reference was established by pressing **[ZERO]** or through zero tracking. The gross weight is calculated internally and its value cannot be changed by any other means.

When displaying the gross weight, the internally calculated value is rounded to the nearest display division size. However, the gross weight stored in **0P** remains the same as the internally calculated value, a value of greater precision than the displayed value. This is an important fact to consider when using the gross weight parameter in macros to accumulate gross weight values. It is likely that the result of several such accumulations will exceed the accumulation of the displayed values. If the accumulated gross weight values must agree with displayed values, use **20P**, the rounded gross weight parameter (see *Rounded Weight Parameters* on page 7-15).

NET (MODE 1)

2)00 lb Net

The net weight parameter represents the difference between gross and tare:

NET = GROSS – TARE

The *net* weight can be used to determine the weight of product in a container if the tare weight of the container has been established. The net weight is also used for multiple ingredient filling applications where a new tare weight is established prior to each fill. Thus each ingredient can fill from a net weight of zero to the desired target net weight. The net weight is calculated internally and its value can only be changed indirectly by specifying a new tare weight.

Like the gross weight parameter, the displayed net weight is rounded to the nearest display division size while the value stored in **1P** remains as the internally calculated net weight. When performing macro accumulations where the accumulated net weight must agree with displayed values, use **21P**, the rounded net weight parameter (see *Rounded Weight Parameters* on page 7-15).

TARE (MODE 2)

!25 lb Tare

The *tare* weight parameter represents a deduction from gross weight made to allow for the weight of a container or other such weight not to be considered as part of the resulting net weight.

The tare value stored in **2P** depends on P163 in the setup mode. If tare rounding is enabled at P163, the value stored in **2P** will be the displayed (rounded) tare weight. If tare rounding is disabled, the tare weight will be stored as a higher precision value. Storing negative tare weights is not possible if negative tare is disabled at P162. The current tare weight is retained during a power interruption if tare save is set to "Auto" at P661. If NTEP is enabled at P440, tare rounding will be enabled and negative tare disabled regardless of the other settings.

The macro command *2.0P=0.0P%o* is equivalent to performing an auto-tare without motion delay.

The tare weight can be changed through the following methods:

- Press **[TARE]** to perform an auto-tare.
- Key in the tare weight and press **[TARE]** to perform a manual tare.
- Assign a tare value using macros.

ACCUMULATION PARAMETERS

The accumulation parameters are primarily used when performing gross and net accumulations as described in the *Accumulation* section. Separate accumulation parameters are maintained for each enabled scale. Accumulation totals are reset to zero (0) during a power loss if P660 is set to “NoSav”.

GROSS TOTAL (MODE 3)

538&28 lb GrTOT

The *gross total* parameter maintains a total of gross weight accumulations. The current gross weight is added to this total each time an accumulation is performed by pressing **[.] [ENTER]** in the gross, net, gross total or net total mode.

The gross total can be initialized to any value by accessing the gross total parameter, keying in the desired value and pressing **[ENTER]**. To clear the gross total, access the gross total parameter and press **[CLR]**. Initializing or clearing the gross total in this manner will reset the number of accumulations parameter (**9P**) to zero. Clearing the gross total in this manner also clears the net total (**6P**).

GROSS TOTAL + CURRENT GROSS (MODE 4)

30!02 lb GrT+C

The *gross total* + *current gross* parameter is an active weight parameter that represents the current gross total (**3P**) plus the current gross weight (**0P**). This parameter is commonly used in conjunction with the accumulation procedure for multiple-dump batching applications. Consider the following scenario:

1. A 5000 lb capacity hopper is to be used to batch 200,000 lbs of material into a rail car. This would require the hopper to be filled and discharged 40 times.
2. A start-batch macro is programmed to zero the scale and clear the gross total (**3P**). Doing so also clears the gross total + current gross parameter.
3. The start-batch macro then activates two setpoints:
4. The first setpoint is configured to be active “never” and deactive “above” the hopper target weight of 5000 lb. This setpoint compares the gross weight to a variable that contains the hopper target value.
5. The second setpoint is configured to be active “never” and deactive above the batch target weight of 200,000 lbs. This setpoint compares the *gross total* + *current gross* parameter to a variable that contains the batch target value.
6. When the hopper target is reached, the first setpoint deactivates and invokes a macro programmed to perform a gross weight accumulation before opening the hopper gate. After the accumulation is performed, the display “locks” the *gross total* + *current gross* weight value. Thus when the hopper gate opens, the value of 4P will not begin to update until the weight in the hopper falls below the Return-to-Zero (RTZ) range specified at P122. This “locking” feature is a function of the

gross total + current gross parameter and does not require additional setup or macro control.

7. Once the hopper empties, the first setpoint reactivates to begin filling again.
8. Steps 6 and 7 are repeated until the value of 4P reaches the 200,000 batch target. This deactivates the second setpoint which invokes another macro programmed to deactivate the first setpoint and complete the batch.

Gross Total - Current Gross (Mode 5)

```
29*98 lb GrT-C
```

The *gross total - current gross* parameter is an active weight parameter that represents the current gross total (**3P**) minus the current gross weight (**0P**).

Net Total (Mode 6)

```
419^21 lb NtTOT
```

The *net total* parameter maintains a total of net weight accumulations. The current net weight is added to this total each time an accumulation is performed by pressing **[.] [ENTER]** in the gross, net, gross total or net total mode.

The net total can be initialized to any value by accessing the net total parameter, keying in the desired value and pressing **[ENTER]**. To clear the net total, access the net total parameter and press **[CLR]**. Initializing or clearing the net total in this manner will reset the number of accumulations parameter (**9P**) to zero. Clearing the net total in this manner also clears the gross total (**3P**).

Net Total + Current Net (Mode 7)

```
20!02 lb NtT+C
```

The *net total + current net* parameter is an active weight parameter that represents the current net total (**6P**) plus the current net weight (**1P**).

Net Total - Current Net (Mode 8)

```
19*98 lb NtT-C
```

The *net total - current net* parameter is an active weight parameter that represents the current net total (**6P**) minus the current net weight (**1P**).

Number Of Accumulations (Mode 9)

```
10 # Accum
```

The *number of accumulations* parameter represents the number of times an accumulation has been performed using the **[.] [ENTER]** method. This value is reset to zero whenever the gross total (**3P**) or net total (**6P**) is cleared using the front keypad. It cannot be cleared or preset in any other way except through the use of macros.

Scale Number (Mode 10)

The *scale* parameter is used only in custom transmits or as a database column to represent the currently selected scale number.

TIME & DATE

A battery backed time/date clock is standard on all 660 and 560 Series Controllers and is optional on 460 Series Indicators. The clock module maintains the time/date even during power loss. When power is restored the time/date is read from the clock module and subsequently maintained by the firmware.

TIME & DATE (MODE 11)

0*0!99	12:00 :00am

The *time & date* parameter maintains the current time and date in 1 second intervals, storing them together as the cumulative number of seconds elapsed since midnight January 1, 1970. The number is stored in a 32-bit binary register (unsigned integer) allowing a maximum value of 4,294,967,295. This value translates to 6:28:15am on February 6, 2106.

When the scale is powered up, the time and date value is initially set at 0 (January 1, 1970, 12:00:00pm). The current time and date is read from the U12 clock chip on the main board and copied to the time & date parameter (**P11**). Adjustments to the time and date can be entered manually or assigned through a macro. The date is manually entered as MM.DD.YY (month.day.year) and then the time is entered as HH.MM.SS (hours.minutes.seconds). These entries are converted to the number of seconds elapsed since midnight January 1, 1970 and then stored in 11P and the clock chip where it is incremented every second.

Although the time & date register can handle dates from three centuries (19xx, 20xx and 21xx), there is a two-digit year entry limitation. A two-digit year entry of 70 or greater is interpreted as 19xx and a two-digit year entry of less than 70 is interpreted as 20xx. This results in a maximum starting date of 11:59:59 pm on December 31, 2069. To overcome this limitation, a short macro could be written to accept a four-digit entry and convert the time and date entry accordingly.

A 10 year lithium battery powers the U12 clock module. The module keeps track of the time and date independently; the instrument only reads the module's time and date at power-up. After power-up, time and date is maintained by the indicator separately from the module. When a new time and date is entered into the unit, it is written to the clock module, thus the limitation of a two-digit entry for the year still applies.

The scale automatically adjusts for Leap Years. However, time changes for Daylight Savings are not accounted for. Once again a small macro routine could easily handle this adjustment.

IMPORTING TIME & DATE TO MICROSOFT® EXCEL

Microsoft® Excel handles time and date similar to the method of the 60 Series instrument, but with two significant differences:

- Excel treats a time & date value of zero (0) as January 01, 1900.
- Excel represents date as the whole number of days since January 01, 1900 and represents the time as a fraction of a day.

For example, 12:00:00 pm, August 1, 1999 would be internally represented as:

36373.50 days in Excel

933508800 seconds in the 60 Series instrument

Both the 60 Series and Excel can format their respective numbers many different ways to represent the time/date in a more readable manner. However, importing the 60 Series time/date number to Excel will require a simple computation to convert it to the Excel format.

To import time & date to Excel:

1. Import the 60 Series format unchanged to a column in Excel. This column can be configured as a hidden column or included on a separate sheet if you do not want to view this column in the main spreadsheet.
2. The number of seconds reported by the 60 Series instrument must undergo two computations:
   - The number of days and fractions of a day must be determined.

     933508800 seconds ¸ 86400 seconds/day = 10804.5 days
   - The number of days between January 1, 1900 and January 1, 1970 must be added to the result of the first computation.

     10804.5 days + 25569 days = 36373.5 days

Both computations can be performed in a single Excel column. For example, if the 60 Series value is imported to Excel column ‘D’, then column ‘E’ can be formatted with the following formula:

E1 = (D1 / 86400) + 25569

Column ‘E’ will now contain the correct Excel time/date value. Format this column to express the time/date as desired.

WEIGHT AVERAGING PARAMETERS

The averaging parameters are used to calculate the average gross or net weight over a period of time. This feature is useful in a variety of applications such as in-motion weighing (i.e. mono-rail scales, truck or rail scales, check-weighing, etc.), weighing hoppers or vessels with mixers or agitators, weighing live animals, or any other application that requires accurate weighing of an unstable object. The *%+ Averaging* macro command is used to start and stop the averaging routine.

AVERAGE GROSS (MODE 15)

1)00	lb AvGrs

The *average gross* parameter represents the average gross weight calculated through use of the *%+ Averaging* macro command. This command can be used to start, stop and resume averaging. A separate average gross parameter is maintained for each enabled scale.

Once averaging begins, **15P** becomes an active weight parameter continuously recalculating the average filtered gross weight until averaging is stopped. If the digital filter is set to 1 second or less at P116 (or by the *%k Digital Filter* macro command), then the average weight is recalculated every 1/60th second). If the filter is set for 2, 4 or 8 seconds, then the average weight is recalculated every 2/60th, 4/60th or 8/60th seconds respectively.

The average gross parameter contains an accumulated weight value. The average gross weight is calculated when accessed by dividing the accumulated weight by the average count of **17P**. Thus if a value is to be entered directly into **15P**, the average count should first be assigned at **17P**. When recalling a gross average value from a database, the average count column should precede the average gross column.

AVERAGE NET (MODE 16)

```
1)00 lb
     AvNet
```

The *average net* parameter represents the average net weight calculated through use of the *%+ Averaging* macro command. The characteristics of the average net parameter are identical to that of the average gross (**15P**).

AVERAGE COUNT (MODE 17)

```
240 AvgCt
```

The *average count* parameter represents the number of times the gross and net weight were accumulated when using the *%+ Averaging* macro command. Typically, the average count increments 60 times per second. The accumulated weight stored internally in 15P and 16P is divided by the average count to calculate the average gross and net values.

```
10#04 lb
      PkGrs
```

PEAK WEIGHT PARAMETERS

The peak weight parameters are active weight parameters representing the peak gross and net weight. They continuously monitor the gross and net weight of each scale and record the current weight each time it exceeds the previously stored value. These parameters can be accessed at any time to determine the maximum weight applied since the last time the peak weight was cleared. They are often used in applications that measure an object's compression or tensile strength.

PEAK GROSS (MODE 18)

The *peak gross* parameter represents the maximum gross weight applied since last cleared. To clear the current peak weight, access **18P** and press **[CLR]**. The peak gross weight will be immediately updated with the current gross weight.

Example:
Invoking a Macro Using Peak Gross

```
5099%s1%e           Setpt     1
5100%s1%e           SPTyp Outpt
5101%sPEAK%e        SPNam PEAK
5110%s5%e           Activ Never
5111%s0.00%e        AcDly 0.00
5112%s0%e           AcMac None!
5130%s1%e           Deact Below
5131%s0.00%e        DeDly 0.00
5132%s10%e          DeMac    10
5133%s0%e           DeMtn Ign'd
5134%s18.1%e        DLPar PEAK
5150%s0.1%e         CmPar Gross

MACRO #6 - START PRESS
%z                  zero scale
18.1P=0%o           reset peak
1%A                 start press

MACRO #10 - PRINT PEAK
2%Q                 print peak
```

USING PEAK GROSS TO INVOKE A MACRO

The following routine demonstrates how to use the peak gross parameter to test and report an object's compression strength:

1. A hydraulic press is used to compress a concrete core sample with increasing force until the sample breaks.

2. An operator presses **[START]** to activate the press. This action invokes macro #6 which first zeros the scale, clears **18P** (both gross and peak gross are now at zero), then starts the press by the activation of setpoint #1.
3. The hydraulic press increases the force on the scale through the concrete block. As the gross weight increases, so does the peak gross weight. Both values remain identical as the force continues to increase.
4. When the concrete breaks, the gross weight immediately drops back to zero. The peak gross weight maintains the maximum gross weight recorded. At this time, the peak gross weight exceeds the gross weight and setpoint #1 deactivates, stopping the press.
5. The deactivation of setpoint #1 invokes macro #10 which sends custom transmit #2 to print the peak gross weight.

PEAK NET (MODE 19)

```
10#04 lb
      PkNet
```

The *peak net* parameter represents the maximum net weight applied since last cleared. To clear the current peak weight, access **19P** and press **[CLR]**. The peak net weight will be immediately updated with the current net weight.

ROUNDED WEIGHT PARAMETERS

The rounded weight parameters represent the displayed gross and net weight for each enabled scale. These parameters are primarily used for performing macro calculations involving gross and net weights where the result of such calculations must agree with the displayed weights.

The rounded weight parameters cannot be displayed from the weigh mode or used in custom transmits. This would be redundant since the rounded gross and net weight is identical to the displayed or printed gross and net weight. However, the displayed gross and net weights are calculated to a higher precision value internally. Table 7-2 shows the relationship between the internal and rounded weight values. Notice that when the gross and net weights are added using **0.0P** or **1.0P**, the result may not agree with the displayed values. Use **20.0P** or **21.0P** to ensure the total will be correct.

Table 7-2: Internal Gross/Net Vs Rounded Gross/Net

INTERNAL GROSS WT. (0.0P)	DISPLAYED GROSS WT. (0.0P)	ROUNDED GROSS WT. (20.0P)	INTERNAL NET WT. (1.0P)	DISPLAYED NET WT. (1.0P)	ROUNDED NET WT. (21.0P)
11.0046	11.00	11.00	10.0046	10.00	10.00
11.0032	11.00	11.00	10.0032	10.00	10.00
11.0029	11.00	11.00	10.0029	10.00	10.00
33.0107	**33.0107**	**33.00**	**30.0107**	**30.0107**	**30.00**

ROUNDED GROSS (PARAMETER 20)

The *rounded gross* parameter represents the displayed gross weight exactly as displayed, rounded internally to the nearest scale division. This parameter cannot be displayed and therefore cannot be selected as a mode of operation.

ROUNDED NET (PARAMETER 21)

The *rounded net* parameter represents the displayed net weight exactly as displayed, rounded internally to the nearest scale division. This parameter cannot be displayed and cannot be selected as a mode of operation.

RATE PARAMETERS

The *rate* parameters are used to indicate the rate of weight change on the scale. Rate is often used to calculate and control a product's flow rate in batching applications. Used in conjunction with the free-fall and future gross/net parameters, the rate parameters can be used to provide real-time adjustment to a target cutoff value in order to achieve accurate fill weights with varying flow rates.

RATE (MODE 23)

)00	lb/s Rate

The rate parameter is an active weight parameter that represents the change in weight over a specified time period. A separate rate is maintained for each enabled scale. Before the rate feature can be used, it must first be enabled by specifying a rate measurement period at P135 of the setup mode. The rate value can either be positive indicating a gain-in-weight, or negative indicating a loss-in-weight. Rate is displayed in the currently selected units per rate time unit (RTU).

RATE MEASUREMENT PERIOD (RMP)

The rate feature must be enabled at P135 for each scale before it can be used to calculate rate. This setup parameter defines the rate measurement period (RMP), in seconds, over which the average rate will be calculated. Valid RMP entries are 0 → 900 seconds. An RMP of zero (0) disables the rate feature. The value is stored internally in $1/60^{th}$ second intervals. The displayed value is rounded off to two decimal places, one decimal place for entries 100 seconds or greater.

A short RMP results in a rate calculation that responds quickly to a change in weight, whereas a larger RMP provides a more stable, accurate rate indication where the change in weight is gradual. For example, an RMP of 0.017 (stored as 0.02) seconds ($1/60^{th}$ second) ensures that a new rate value will be calculated with each new gross weight reported from the A/D converter. An RMP of 1.0 seconds will display the average of the last 60 rate calculations.

Rate Time Unit (RTU)

The rate time unit (RTU) specifies the time unit for displaying the calculated rate. The RTU is specified at P136 as either seconds, minutes or hours.

Resetting the Average Rate

When a long RMP is used to indicate rate in an application such as a loss-in-weight system, it may be necessary to clear the rate history at certain times. For example, consider a slowly discharging hopper scale with a 60 second RMP. When the hopper discharges to a low-limit value, it will need to be refilled. Refilling is usually a very quick process during which time the flow rate reverses as product is added to the hopper much faster than it is being discharged. Once the hopper is refilled, the rate again reverses as the system continues to discharge. With a 60 second RMP, the displayed rate still reflects the average of the previous 3600 rate calculations. Thus even though actual the rate is now slightly negative, the indicated rate value will likely be positive for the next minute until the rate history has moved beyond the time during which the hopper was filling.

The rate history can be cleared at any time using the *%k Digital Filter* macro command. The command R%k will instantly clear the rate history and begin recalculating the average rate value with the next A/D conversion.

Free Fall (Mode 24)

```
!00 FreFl
```

The *free fall* parameter represents the number of seconds it takes for product in free fall to reach the scale. The free fall value is not calculated by the 60 Series instrument. It must be assigned manually or through a macro command. Once assigned, this value is used to calculate the weight of product in free fall based on the current flow rate. For example, if the current flow rate is 2.5 lb/sec and the free fall time is 2 seconds, then there would be 5 lbs of product in free fall. This free fall weight, recalculated with each A/D conversion, can then be used to *predict* what the weight will be 2 seconds in the future. This becomes the basis for the future gross and future net parameters.

Calculating Free Fall

If the precise free fall time is known, it can be entered directly into 24P. In many cases this value can more accurately be determined using macros. A *learn* cycle can be run whereby free fall is calculated as follows:

1. Product is allowed to fill to its target weight.
2. When the target weight is reached, a gate or valve is closed by the deactivation of the fill setpoint.
3. The setpoint deactivation invokes a macro which immediately copies the instantaneous rate to a variable, for example:

   80.1P=23.1P%o

4. Next, the overfill amount is determined after a motion delay by subtracting the target weight from the gross weight:

   M1%,
   80.3P=0.1P-80.2P%o

5. where 80.2P is the target weight and 80.3P is the overfill weight.
6. Finally, the free fall time is calculated as the overfill weight divided by the rate at the time the target was reached:

   24.1P=80.3P/80.1P%o

FUTURE GROSS (MODE 25)

2$03 lb FutGr

The *future gross* parameter is an active weight parameter that represents a *predicted* gross weight calculated by multiplying the current rate by the free fall time and adding the result to the current gross weight:

Future Gross (25P) = Rate (23P) ´ Free Fall (24P) + Gross (0P)

The future gross weight is recalculated with each A/D conversion. This parameter provides a very accurate means of determining the proper cutoff for filling applications as it can automatically adjust the cutoff value to account for variations in flow rate.

For example, suppose you want to fill a hopper to a target weight of 500 lbs. If the current flow rate is 5.0 lb/sec and the free fall time is 2 seconds, then there would be 10 lbs of product in free fall. The future gross weight then becomes the current gross weight plus 10 lbs. Thus when the gross weight reaches 490 lbs, the future gross weight will indicate 500 lbs. If we use the future gross weight as the basis for the fill valve setpoint, the valve will close when the future gross weight reaches 500 lbs. The gross weight is only 490 lbs at that time, but we know there will be 10 lbs of additional free falling product.

Now suppose that the flow rate changes to 10 lb/sec for the next fill cycle. The free fall time remains constant at 2 seconds, so the free fall weight now becomes 20 lbs. Again, the fill valve does not close until the future gross weight reaches 500 lbs. However, this time the actual gross weight will be 480 lbs when the valve closes, thus accounting for the additional 20 lbs of free falling product.

FUTURE NET (MODE 26)

2)03 lb FutNt

The *future net* parameter is an active weight parameter that represents a *predicted* net weight calculated by multiplying the current rate by the free fall time and adding the result to the current net weight:

Future Net (26P) = Rate (23P) ´ Free Fall (24P) + Net (1P)

The future net weight is otherwise identical to the characteristics of the future gross weight.

FREE FALL 2 (MODE 27)

!00 FrFl2

The *free fall 2* parameter is identical to **24P** except that it is used to calculate the future gross and future net weight for **28P** and **29P** respectively.

FUTURE GROSS 2 (MODE 28)

2$03 lb FuGr2

The *future gross 2* parameter is identical to **25P** except that it uses the free fall time of **27P** to calculate its value. This provides a second future gross parameter for use in two-speed filling applications so both fast and slow fill cutoff values can take advantage of the rate feature. When using **28P** to determine the slow fill cutoff, be sure to clear the rate history with the R%k command immediately after achieving the fast fill target.

FUTURE NET 2 (MODE 29)

2$03 lb FuNt2

The *future net 2* parameter is identical to **28P** except that it tracks the net weight rather than the gross weight.

COUNTING PARAMETERS

The counting parameters represent various information for use in the counting mode. These parameters are only accessible if the counting mode is enabled at P179 of the setup mode. Refer to the *Counting* section for complete details on the counting feature.

QUANTITY (MODE 30)

1) Qty

The *quantity* parameter is an active weight parameter that represents a number of pieces on the scale. A separate quantity is maintained for each enabled scale. The quantity is calculated by dividing the net weight by the average piece weight (APW):

Quantity (30P) = Net (1P) ¸ APW (34P)

The quantity can be established by two methods:

- Performing a piece sample.
- Assigning a value to the average piece weight parameter (**34P**).

If an APW has not been established, the prompt Must Sampl will be displayed when attempting to access the quantity mode. Press **[ENTER]** to tare the scale and begin the sampling routine or enter the APW at **34P**. It is also possible to assign an APW through macros or by recalling an APW from a database.

When the display shows a quantity greater than zero (0), you can change the quantity by keying in the correct value and pressing **[ENTER]**. The APW will be recalculated accordingly and the newly entered quantity will be displayed.

QUANTITY TOTAL (MODE 31)

`100) QtTOT`

The *quantity total* parameter maintains a total of quantity accumulations. The current quantity is added to this total each time an accumulation is performed by pressing **[.] [ENTER]** in the quantity or quantity total mode.

The quantity total can be initialized to any value by accessing the quantity total parameter, keying in the desired value and pressing **[ENTER]**. To clear the quantity total, access the quantity total parameter and press **[CLR]**. Initializing or clearing the quantity total in this manner will also clear the gross total (**3P**) and net total (**6P**) and reset the number of accumulations parameter (**9P**) to zero. Note that clearing the gross total or net total in the same manner does <u>not</u> affect the quantity total.

QUANTITY TOTAL + CURRENT QUANTITY (MODE 32)

`12501 QtT+C`

The *quantity total + current quantity* parameter is an active weight parameter that represents the current quantity total (**31P**) plus the current quantity (**30P**). This parameter is commonly used in conjunction with the accumulation procedure for multiple-dump batching applications based on piece count.

QUANTITY TOTAL - CURRENT QUANTITY (MODE 33)

`7499 QtT-C`

The *quantity total - current quantity* parameter is an active weight parameter that represents the current quantity total (**31P**) minus the current quantity (**30P**).

AVERAGE PIECE WEIGHT (MODE 34)

`)02277 lb APW`

The *average piece weight* parameter (APW) represents the average weight of an individual piece as calculated during the sampling routine. Only one APW is maintained for all enabled scales. The net weight of each scale is divided by the APW to determine the quantity (**30P**):

Quantity (30P) = Net (1P) ¸ APW (34P)

An APW can be assigned through macros or by recalling an APW from a database. This allows the quantity to be calculated without having to resample. Note that the APW is cleared at power-up.

AVERAGE PIECE WEIGHT X 1000 (MODE 35)

`2@7700 lb APW*K`

The *average piece weight ´ 1000* parameter (APW*K) represents the average weight of 1000 pieces as calculated during the sampling routine. Only one APW*K is maintained for all enabled scales.

PERCENT ACCURACY (MODE 36)

9(806 %Accy

The *percent accuracy* parameter represents the minimum accuracy achieved during the last sample routine. Only one percent accuracy parameter is maintained for all enabled scales.

LAST SAMPLE SIZE (MODE 37)

10 Last Sampl

The *last sample size* parameter represents the number of pieces used during the last sample routine to determine the current APW. Only one last sample size parameter is maintained for all enabled scales.

MULTI-SCALE PARAMETERS

4202) lb GrAll

The multi-scale parameters automatically calculate the total of individual gross, net, tare, quantity, gross total, net total, and quantity total weight parameters for all enabled scales.

Most legal-for-trade applications will require that you display the same units and division size on all scales when displaying the gross total of all scales. Set P111 the same for all enabled scales. To toggle the units for all scales, reassign the [UNITS] key at P803 to invoke a macro. Configure the macro to toggle the units for each scale with the *%uUnits* macro command.

GROSS TOTAL OF ALL SCALES (MODE 40)

The *gross total of all scales* parameter is an active weight parameter that represents the total gross weight of all enabled scales. The total weight is displayed in the current units and division size of the current scale. This parameter is commonly used with multiple-axle truck scales where the weight of individual axles as well as the total truck weight is required.

Gross Total of All Scales (40P) = 0.1P + 0.2P + 0.3P + 0.4P

NET TOTAL OF ALL SCALES (MODE 41)

2128) lb NeAll

The *net total of all scales* parameter is an active weight parameter that represents the total net weight of all enabled scales. The total weight is displayed in the current units and division size of the current scale.

Net Total of All Scales (41P) = 1.1P + 1.2P + 1.3P + 1.4P

TARE TOTAL OF ALL SCALES (MODE 42)

2074) lb TrAll

The *tare total of all scales* parameter represents the total net weight of all enabled scales. The total weight is displayed in the current units and division size of the current scale.

Tare Total of All Scales (42P) = 2.1P + 2.2P + 2.3P + 2.4P

TOTAL OF ALL GROSS TOTALS (MODE 43)

4512)5 lb GTAll

The *total of all gross totals* parameter represents the total of all gross totals. The total weight is displayed in the current units and division size of the current scale. This parameter is used for accumulation applications to instantly determine the total gross accumulation of all scales.

TOTAL OF ALL NET TOTALS (MODE 44)

4512)5 lb NTAll

The *total of all net totals* parameter represents the total of all net totals. The total weight is displayed in the current units and division size of the current scale.

Total of All Net Totals (44P) = 6.1P + 6.2P + 6.3P + 6.4P

QUANTITY TOTAL OF ALL SCALES (MODE 45)

328 QuAll

The *quantity total of all scales* parameter is an active weight parameter that represents the total piece count of all enabled scales.

Quantity Total of All Scales (45P) = 30.1P + 30.2P + 30.3P + 30.4P

TOTAL OF ALL QUANTITY TOTALS (MODE 46)

80225 QTAll

The *total of all quantity totals* parameter represents the total of all quantity total parameters.

Total of All Quantity Totals (46P) = 31.1P + 31.2P + 31.3P + 31.4P

PROGRAMMABLE DIGITAL I/O PARAMETERS (PDIO)

The programmable I/O parameters are used in conjunction with the eight PDIO channels of the M660 main board J11 connector. These channels can be configured to perform a variety of input/output functions. The purpose of each I/O parameter depends upon the function of each PDIO channel defined at P851.

Table 7-3 illustrates the relationship between each PDIO function and their corresponding I/O parameters. *Chapter 11, Programmable Digital I/O* provides complete details on all PDIO functions.

Table 7-3: PDIO Parameter Functions

PDIO FUNCTION	PURPOSE OF 50P (PDIO A)	PURPOSE OF 51P (PDIO B)	PURPOSE OF 52P (PDIO C)
Frequency Out	Specifies output frequency.	Specifies output duty cycle.	-
Setpoint	-	-	-
Frequency Input A	Reports the input frequency.	-	-
Frequency Input B	Reports the input frequency.	Reports the input pulse count.	-
Phase Time	Reports the phase time of the input frequency.	Reports the input pulse count.	-
Delay Input	-	-	-
Delay Output	Specifies the width of the output pulse.	Specifies the delay before sending the output pulse.	-
Quadrature Decode 2	Reports the input pulse count.	-	-
Quadrature Decode 3	Reports the absolute pulse count.	Reports the pulse count at the time of the last index pulse.	Reports the pulse count relative to the index pulse.
Frequency Debounce	Reports the input frequency.	Reports the input pulse count.	Reports the average input frequency.

PDIO A (MODE 50)

0 PIOA1

The purpose of the *PDIO A* parameter is determined by the function for each PDIO channel (see Table 7-3). When accessing **50P**, be sure to specify an instance (1 → 8) to identify the desired channel.

PDIO B (MODE 51)

0 PIOB1

The purpose of the *PDIO B* parameter is determined by the function for each PDIO channel (see Table 7-3). When accessing **51P**, be sure to specify an instance (1 → 8) to identify the desired channel.

PDIO C (MODE 52)

0 PIOC1

The purpose of the *PDIO C* parameter is determined by the function for each PDIO channel (see Table 7-3). When accessing **52P**, be sure to specify an instance (1 → 8) to identify the desired channel.

EXTENDED WEIGHT PARAMETERS

The extended weight parameters allow you to print the gross, net or tare values using their full internal precision. The A/D conversion number provides a reference number used in calculating each weight reading. These parameters can be used in a custom transmit or stored in a database. However, they are not available for the selectable modes at P300 → P309 of the setup mode.

EXTENDED RESOLUTION GROSS (MODE 60)

The *extended resolution gross* parameter represents the full precision value of displayed (rounded) gross weight.

EXTENDED RESOLUTION NET (MODE 61)

The *extended resolution net* parameter represents the full precision value of displayed (rounded) net weight.

EXTENDED RESOLUTION TARE (MODE 62)

The *extended resolution tare* parameter represents the full precision value of displayed (rounded) tare weight.

A/D CONVERSION NUMBER (MODE 63)

The A/D Conversion Number is the number of the last A/D conversion of a particular scale used in the calculation of the weight. A/D conversions are done every 60^{th} of a second (16.66 millisecond). This parameter provides a way to determine the time between two calculated weights. This number starts at zero at power-up and upon exiting the setup mode. It rolls over at 4,294,967,295. The number will increment regardless of overload and underload errors and is independent of the filter setting at P116. Ensure that the operation to get P63 and the desired parameter are done together and cannot be interrupted or the A/D conversion number and desired parameter will not correspond. Thus the weight and the conversion number should be included in a single transmission or a 'create row' command for a database. The A/D conversion number is not a displayable parameter. This means that it cannot be displayed from weigh mode with the use of the select key (i.e. pressing **63 [SELECT]** or pressing **63.[#] [SELECT]**).

SETUP MODES

Parameter 63 can be selected as a parameter for use with analog outputs (P172 → P175), input interpreters (P222), database (P701 → P799), custom transmits (P1000), setpoints (P5114, P5115, P5134, P5135, P5150), and as a parameter for Modbus (P6001 → P6247). When selecting this parameter #A/D is shown if the scale is not known yet. When the scale is known (the instance) then #A/D$ is displayed where $ is the number of the scale.

A/D CONVERSION NUMBER WITH %O

You can **get** the value of a scale's A/D conversion number by using =63.xP%o where x represents the number of the scale for which the current A/D conversion number is desired.

You can **set** the state of a scale's A/D conversion number by using 63.xP=$%o where x represents the number of the scale for which the current A/D conversion number is desired and $ represents the number you wish to set the A/D conversion number to. Remember this exact number may not be seen unless you read the A/D conversion number immediately after setting it.

USING WITH CUSTOM TRANSMITS

The A/D conversion number parameter was created primarily for use with custom transmits (P1000). The same custom transmit must contain 63P and the desired parameters for which the corresponding A/D conversion number is desired. To send a continuous transmit at high rates (60 Hz maximum) the baud rate (P200) must be fast enough, the scale filter settings at P116 must be fast enough and the amount of data sent by the custom transmit must be limited. Finally, P980, TxRate only allows multiples of 0.1 second. To change the transmit rate to alternate multiples the I%Q macro must be used. It is important to realize that this rate will be reset to the value at P980 on power up and when exiting setup mode!

Example

To send a continuous transmit every 30th of a second out comm port 1, first set the filter at P116 for the scale you are interested in to 1.0 second. Set P998 (Continuous Transmit) to “enabled” and configure the custom transmit table as desired (use 63P). After saving and exiting setup mode or after powering up, send a .033I%Q to override the interval at P980. It important to note that setpoints, programmable IO operations, macros, and use of other comm ports will affect the transmission rate.

DSD PARAMETERS

The Data Storage Device (DSD) parameters are associated only with the DSD feature. DSD must be enabled at P590 and configured at P591 → P595 in order to use these parameters.

DSD PARAMETERS (PARAMETERS 64.1 – 64.9)

Refer to the DSD database structure on page 6-14 for a complete description of DSD parameters.

SETPOINT TIMERS

The setpoint timer parameters are used to display or manipulate the 256 independent setpoint delay timers. These parameters are not displayable modes of operation, but can be used in custom transmit tables or macros to display or change timer values.

Example:
Using a Countdown Timer

```
989%s2%e%e      CusTx     2
990%sMix Time%e TxNam Mix
Time
991%s0%e        Send: Off
992%s4%e        Port  Comm4
993%s0%e        CSMtn Ignrd
994%s0%e        S1Mtn Ignrd
995%s0%e        S2Mtn Ignrd
996%s0%e        S3Mtn Ignrd
997%s0%e        S4Mtn Ignrd
998%s1%e        Cont. Enbld
999%s0%e        LmtAc  no
4999.2%s%c%e    Transmit #2

.027%e           <ESC>
.072%e           position
cursor
.076%e
%e76.2%e21016%e%e       mm:ss


5099%s2%e         Setpt     2
5100%s1%e         SPTyp Outpt
5101%sMixer%e     SPNam Mixer
5110%s5%e         Activ Never
5111%s0.00%e      AcDly 0.00
5112%s0%e         AcMac None!
5130%s4%e         Deact Alwys
5131%s90.00%e     DeDly 90.00
5132%s0%e         DeMac None!
5133%s0%e         DeMtn Ign'd


MACRO #6 – START MIXER
2%A               start mixer
```

SETPOINT COUNTDOWN TIMER (PARAMETER 76)

The *setpoint countdown timer* parameter represents the number of seconds remaining in the activation or deactivation delay for a specified setpoint. This parameter stores the delay time as an integer value, thus it reports only a whole number of seconds. When an activation or deactivation delay starts, **76P** begins with the total delay time value and decrements by one (1) each second. A value of zero (0) indicates no delay in progress.

Since **76P** is stored as an integer, it may be formatted as an integer in custom transmit tables. This allows you to transmit the delay value using time formats such as *hh:mm:ss*. This is useful for displaying elapsed time values such as remaining mix time, etc. The example – *Using a Countdown Timer* shows how to configure a simple countdown timer to display remaining mix time in *mm:ss* format in the bottom right corner of the 4X20 VF display. Setpoint #2 runs the mixer for 90 seconds and serves as the countdown timer using 76.2P to display the mix time. The **[START]** key will start the mixer.

Additional time can be added to the remaining activation or deactivation time delay. For example, the macro command

76.2P+=60%o

could be used to add an additional 60 seconds of mix time to a mixer controlled by setpoint #2.

SETPOINT DELAY TIMER (PARAMETER 77)

The *setpoint delay timer* parameter represents the number of seconds remaining in the activation or deactivation delay for a specified setpoint. This is similar to the countdown timer **76P** except that the delay timer is stored as a floating point value. This means that you cannot specify time formats for **77P** in custom transmit tables. However it does have the advantage of being specified in fractions of a second.

SETPOINT STATUS (PARAMETER 78)

The *setpoint status* parameter provides access to individual setpoint status. For example, assigning 78.1P to P303 of the setup mode give easy access to the status of setpoint #1 via the **[SELECT]** key. This also give you the opportunity to change setpoint status as described in the setpoint chapter on page 10-3.

RANDOM NUMBERS

The 460 Series indicators will not generate a true (pseudo) random number without the optional time & date module installed. Without this module, the random number generator seed value will likely be the same each time the indicator is powered up.

The random number parameter will generate and display a true random number. This feature is useful in quality assurance applications to randomly select items for weight verification.

RANDOM NUMBER (PARAMETER 79)

The *random number* parameter generates a random number between 0.0000000 → 1.0000000 exclusive of the end points. A random number is produced by copying **79P** to another operating parameter (usually a float type variable) or by copying **79P** to the entry buffer:

80.1P=79P%o	stores a random number in variable #1 (float)
79P%o	copies a random number to the entry buffer
80.1P=79P*100%o **80.2P=80.1P+.5%o**	stores a random number from 0 → 100 in variable #2 (integer)
80.1P=79P*100%o **80.2P=80.1P+1%o**	stores a random number from 1 → 100 in variable #2 (integer)
80.1P=79P*90%o **80.2P=80.1P+10.5%o**	stores a random number from 10 → 100 in variable #2 (integer)

The random number parameter can be seeded to reproduce a series of values:

79P=x%o	seeds the random number generator (x > 0)

Restart the random number generator at a random value after seeding it by assigning a value of zero (0):

79P=0%o	restart generator at a random point

VARIABLES

Variables must be allocated at P680 and then configured at P681 → P689 before they are available for use.

Variables, referred to hereafter as VARs, are user defined memory registers within the 60 Series instrument which store various types of data values. The stored values may be saved, recalled or changed at any time. Up to 999 VARs may be dynamically allocated at P680 (memory permitting).

VARIABLE TYPES

The 60 Series uses four types of variables as described in Table 7-4

The VAR type is defined in the setup mode at P686. Any VAR can be configured as any one of the four VAR types.

Table 7-4: Variable Types

VARIABLE TYPE	DESCRIPTION	VALID RANGE	EXAMPLES
FLOATING POINT (Float)	A number with a decimal place. A float has an integer part to the left of the decimal, and a fractional part to the right of the decimal.	Resolution of 1 part in 16,777,216	3.1415927 0.00356 -10.549 50000.0
INTEGER (Int)	A positive or negative whole number.	-2,147,483,648 +2,147,483,647	12543 -32689 0
UNSIGNED INTEGER (U-Int)	A positive whole number.	0 +4,294,967,295	2356120 0
STRING (Strng)	An alpha-numeric value comprised of numbers and/or letters and/or other ASCII characters.	Number of characters limited by the size specified at P689 (maximum of 63 characters).	123456 123-A-2b Test Run $10.00

ACCESSING VARIABLES

Variables may be accessed in the same method as other parameters by specifying the parameter (80) and instance:

80.1 [SELECT] selects VAR #1

80.20 [SELECT] selects VAR #20

Variables can be accessed from the weigh mode via the [SELECT] key alone if included in the Mode Menu at P300 → P309.

You can access variables using the **[ID]** key if it is not redefined at P806:

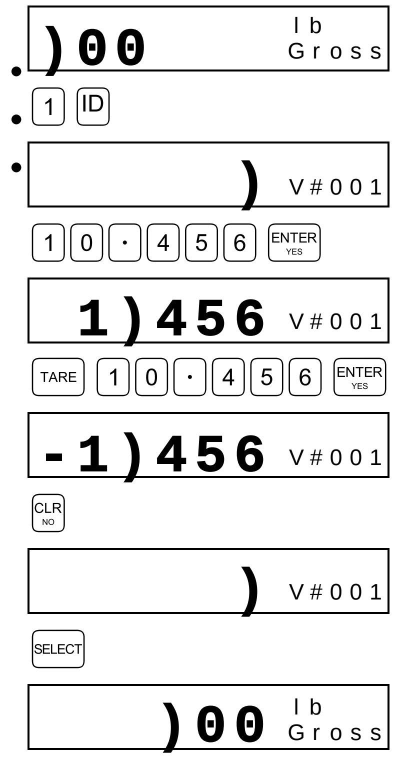
1 [ID] selects VAR #1

20 [ID] selects VAR #20

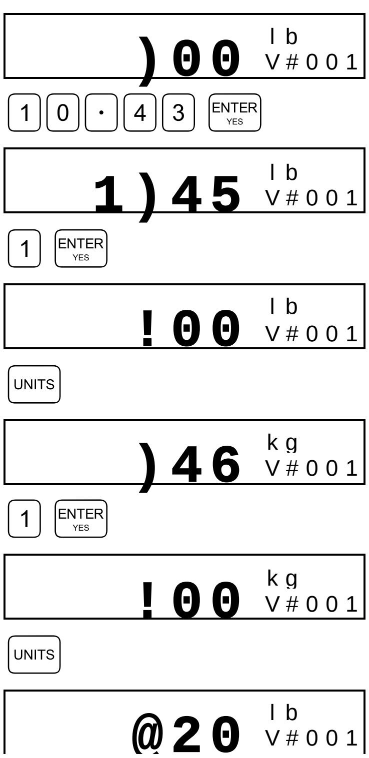
This method requires fewer keystrokes as only the instance number must be keyed in prior to pressing **[ID]**.

ASSIGNING VALUES TO VARIABLES

Example:
Entering Float Values



Example:
Scale Specific Float Values
(Division Size = 0.05 lb = 0.02 kg)



Variables may be values assigned manually through the front keypad, through serial communications, or through macro assignments. For additional information on macro assignments, refer to the following sections in *Chapter 9, Macros*:

%m Modify String

%o Math Assignment

%v Write Value to EEPROM

FLOATS

Floating point variables are used to store numeric values that may have an integer part to the left of the decimal, and a fractional part to the right of the decimal. Floating point values are stored with an internal resolution of 1 part in 16,777,216. Any value that exceeds this range may be rounded. Thus floats are not recommended for very large values, such as part numbers, where the stored value must be retained exactly as entered.

Entering Float Values

To manually enter a value, access the desired variable as described in *Accessing Variables* on page 7-5, key in the desired value and press **[ENTER]**. Negative values may be entered by pressing **[TARE]** before beginning the entry. Press **[CLR]** to set the displayed float value to zero (0).

Decimal Places

The number of decimal places used when displaying or transmitting a float value is determined by P687. Standard rounding techniques apply.

Scale Specific Floats

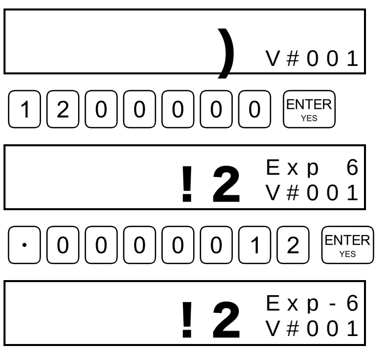
When a float type VAR is defined as a scale specific value, the entered value is accepted in the current weight units and rounded to the nearest scale division size. Pressing **[UNITS]** will convert the displayed and transmitted value for the new units. Internally, the float value is stored in terms of the default units selection at P150. This is important to note when assigning and calculating scale specific float values using the *%o Math Assignment* macro command. All math operations will consider the float value in terms of the default units regardless of the current units selection.

Table 7-5: Rounding Float Values

P687	ENTERED VALUE	DISPLAYED VALUE	TRANSMITTED VALUE
0 dp	10.456456	10	10
1 dp	10.456456	10.5	10.5
2 dp	10.456456	10.46	10.46
3 dp	10.456456	10.456	10.456
4 dp	10.456456	10.4565	10.4565

5 dp	10.456456 10.45 -10.456456	10.4565 10.4500 -10.456	10.45646 10.45000 -10.45646
Auto	10.456456 10.450000 10	10.4565 10.45 10.	10.45646 10.45 10
Scl#1	10.46 lb	10.46 lb 4.745 kg	10.46 lb 4.745 kg

Example:
Exponential Float Values



Exponential Numbers

Floating point variables represent very small or large numbers in exponential form (scientific notation). Any value less than 0.00001 or larger than 999999, the limits of the 6-digit display, will be represented in exponential form. When transmitting values beyond this range, the full decimal value is sent rounded to an accuracy of one part in 16,777,216

Exponential notation does not apply to scale specific float values. If a scale specific float exceeds the displayable range, Code04 Num >Dsply is displayed.

Table 7-6: Exponential Representation

ENTERED VALUE	DISPLAYED VALUE	TRANSMITTED VALUE
999999	999999.	999999
1000000	1. Exp 6	1000000
12345678	1.23457 Exp 7	12345678
-12345678	-1.2346 Exp 7	-12345678
.0001	0.0001	0.0001
.00001	1. Exp -5	0.00001
.0000125	0.00001	0.0000123
.00000125	1.25 Exp –6	0.0000012
-.00000125	-1.25 Exp –6	-0.0000012

When using the *%o Math Assignment* macro command to copy an unformatted exponential value to the entry buffer, the value will appear in the entry buffer in exponential notation:

80.1P=12345678%o	assigns the value 12345678 to VAR #1
80.1P%o	copies 1.23457e+07 to the entry buffer
80.1.16384P%o	copies 12345678 to the entry buffer

INTEGERS

Integers are used to store positive and negative whole numbers ranging from -2,147,483,648 to +2,147,483,647. Integers are typically used to increment and decrement values and store ID numbers. Integers can also be used to store time/date values, although unsigned integers are better suited for this purpose.

An integer may be formatted as a number, time, date or time & date value. Select the desired format at P688. This selection will determine how the integer value is displayed and how it must be entered. Internally, the value remains stored as a number. If a float containing a fractional value is copied to an integer, the fractional portion is truncated, not rounded.

If a signed integer is assigned a value past its upper limit, the register will "roll over" and begin again from the lower limit. For example, entering a value of +2,147,483,648 will yield a value of -2,147,483,648. An entered value of +2,147,483,649 yields a value of -2,147,483,647. The opposite holds true if a negative value is entered. Entering a value of -2,147,483,649 will yield a value of +2,147,483,647. An entered value of -2,147,483,650 yields a value of +2,147,483,646.

An integer value that less than -99999 or greater than 999999 will be displayed as `Code04 Num >Dsply`.

NUMBER FORMAT

`0 V#002`

Select "Numbr" at P688 to format the integer as a signed whole number. To manually enter a number value, access the desired variable as described in the *Accessing Variables* section on page 7-5, key in the desired value and press **[ENTER]**. Negative values may be entered by pressing **[TARE]** before beginning the entry. Press **[CLR]** to set the displayed integer value to zero (0).

TIME & DATE FORMAT

`0*0!99 01:00 :00pm`

Select "TmDat" at P688 to format the integer as a time/date value. Time/date values are entered in the same manner as date-only and time-only formats, with the date value entered first.

For example, to enter a time/date of 1:00pm on August 1, 1999, key in:

8.1.99 [ENTER]

13.00 [ENTER]

Internally, this time value is stored as 933512400, the number of seconds since 12:00:00am on 01/01/99. See *Time & Date (Mode 11)* on page 7-12 for more information on how the 60 Series instruments handles time/date values.

TIME ONLY FORMAT

`1#0)00 Time V#002`

Select "Time" at P688 to format the integer as a time value. Time values must be entered in one of the following formats:

`hh.mm.ss` hours.minutes.seconds – 24 hour format (seconds optional)

`hh:mm:ss` hours:minutes:seconds – 24 hour format (seconds optional)

For example, to enter a time of 1:00pm, key in:

13.00.00 [ENTER]

Internally, this time value is stored as 46800, the number of seconds since 12:00:00am. See *Time & Date (Mode 11)* on page 7-12 for more information on how the 60 Series instruments handles time/date values.

Press **[CLR]** to set the displayed time value to **00.00.00** (12:00:00 am). This also clears the internal integer value to zero (0).

DATE ONLY FORMAT

0*0!99	Date V#002

Select "Date" at P688 to format the integer as a date value. Date values must be entered in one of the following formats:

`mm.dd.yy` month.day.year – if P504 set for U.S.A style

`mm/dd/yy` month.day.year – if P504 set for U.S.A style

`dd.mm.yy` day.month.year – if P504 set for Int'l style

`dd/mm/yy` day.month.year – if P504 set for Int'l style

For example, to enter a date of August 1, 1999 in USA style, key in:

8.1.99 [ENTER]

Internally, this date value is stored as 933465600, the number of seconds since 12:00:00am on 01/01/1970. See *Time & Date (Mode 11)* on page 7-12 for more information on how the 60 Series instruments handles time/date values.

Press **[CLR]** to set the displayed time value to **01.01.70** (January 1, 1970). This also clears the internal integer value to zero (0).

UNSIGNED INTEGERS

Use unsigned integers whenever dealing with time/date values.

Unsigned integers are used to store positive whole numbers ranging from 0 to +4,294,967,295. Unsigned integers are typically used to store large ID numbers and time/date values.

Aside from the fact that they allow only positive values up to twice the value of integers, unsigned integers are treated identical to integers in terms of formatting choices and methods of entry. Refer to the previous section on *Integers* for full details.

If an unsigned integer is assigned a value past its upper limit, the register will "roll over" and begin again from the lower limit. For example, entering a value of +4,294,967,296 will yield a value of zero (0). An entered value of +4,294,967,297 yields a value of 1. The opposite holds true if a negative value is entered. Entering a value of -1 will yield a value of +4,294,967,295. An entered value of -2 yields a value of +4,294,967,294.

STRINGS

Strings are the most versatile of the four variable types, storing up to 63 alpha-numeric characters each. When containing only numbers, string variables can perform the same math functions as floats and integers. Strings can be combined or used to combine strings and numbers. Refer to the *%m* macro command for full details about the various string functions.

The only configuration for string variables is defining the maximum number of characters at P689. Although you can allocate up to 63 characters for a single string, it is best to limit the maximum size to conserve memory, especially when using a string as a column in a database.

ENTERING STRING VALUES

To manually enter a value, access the desired variable as described in the *Accessing Variables* section on page 7-5, key in the desired value and press **[ENTER]**. Numeric characters can be entered through the numeric keys on the front keypad. Alpha characters and other ASCII symbols can be entered via the cursor (arrow) keys on the front panel as described in the *Key In Value* Parameters section. Press **[CLR]** to clear all characters in the string.

In applications where alphanumeric entries will be common, the alpha keyboard option or the alphanumeric serial keyboard converter kit should be installed to simplify the entry process.

DISPLAYING STRING VALUES

Alpha V#004

When displaying string variables, the 2X5 character section of the VFD is used to identify both the variable number and its value while the 7-segment section remains blank. The top five character positions represent the first five characters of the stored value. The bottom five characters are used to display the variable name. If a string contains more than 5 characters, you can use the right and left cursor (arrow) keys to scroll forward and backward through all characters of the string.

NAMING VARIABLES

12	Oper. ID# ?

Variables can be named at P682. This allows you to display useful prompts such as `Enter Targt`, `P-Act Value`, `Fast Fill`, `Slow Fill`, or `Oper. ID# ?` when selecting a variable as the current mode of operation. In most cases, the entire 2X5 character section of the VFD can be used to display a variable's name. Exceptions to this are names for scale specific floats and strings where only the first five characters of the name will be displayed. Time/date type integers will not display a given name since the 2X5 character section is reserved for displaying the time value. Integers configured as time-only or date-only values will keep the `Time` or `Date` prompt on the upper 5 character section and display the given name on the lower 5 character section if the name is 5 characters or less. If the name exceeds 5 character, the name will occupy the entire 2X5 character section.

Regardless of how many characters can be displayed, all characters of a variable's name will be transmitted when included in a custom transmit table. When naming variables, keep in mind how the name will appear on the display. Abbreviate names to 5 characters or less and include spaces where necessary to center text or to wrap text to the next line.

Saving Values During Power Loss

`P68$02 VSave Auto`

In order to save the value of a variable during a power loss, P684 should be configured as “Auto“ or “OnReq“. If set for “Auto“ save, the variable value is written to EEPROM every time it changes. If set for “OnReq“, the value is only written to EEPROM after issuing the *%v Write Value To EEprom* macro command.

Locking Variables

`P68%02 Vlock Enbld`

“Locking” a variable prevents its value from being changed manually. Variable locking can be enabled at P685. A locked VAR can be viewed to verify its value, however its value cannot be changed or cleared from the front keypad. It is still possible to change the value of a locked VAR through the use of various macro commands.

Independent Timers

The 60 Series instruments have 8 independent timers that begin at zero (0) upon power-up and increment by 1/747 second intervals continuously thereafter. These timers can be used in macros to calculate elapsed times or in setpoints to provide a precise time interval between events. Each timer can be displayed in terms of “ticks” or “seconds”.

Timer Ticks (Mode 81)

`0 Ticks`

The *timer ticks* parameter represents the number of 1/747 second intervals that have elapsed for the specified independent timer. This value increments 747 times each second. This value can be reset to zero (0) or preset to a specific value, but the timer cannot be paused or stopped by any means.

[CLR]	resets displayed timer to 0
747 [ENTER]	presets displayed timer to 747 (one second)
81.5P=0%o	resets timer #5 via macro command

Timer Seconds (Mode 82)

`)000 Sec.`

The *timer seconds* parameter represents each of the 8 tick timers in terms of elapsed seconds. As with the timer ticks parameter, the timer seconds value can be reset to zero (0) or preset to a specific value in the same way. Doing so will also reset the timer tick parameter for the specified instance.

When the timer seconds parameter begins at zero (0), the elapsed time is displayed to 3 decimal places. When the value exceeds the displayable value of 999.999, the value is displayed to 2 decimal places. If the timer is allowed to continue, the decimal will shift again to 1 decimal place, and once more to display the value only as a whole number of seconds. When the value exceeds 999999, the display shows `Code04 Num >Dsply`.

PROMPTING PARAMETERS

The prompting parameters allow you to display user defined messages anywhere on the 7-segment VF display (or the auto-update portion of all other displays) or retrieve displayed information. The Macro Select parameter adds the ability to invoke named macros through the select menu at P300 → P309.

MACRO SELECT (MODE 90)

Macros that are menu enabled are accessible via P90. The instance of parameter 90 determines the macro number. Pressing **90.X [SELECT]** in the weigh mode (where 'X' is a macro number) will bring up a display showing macro name (assigned at P9991) of the nearest menu enabled macro. The display is formatted with the numeric portion of the display blank. The 2x5 character portion of the display can have three different formats:

- If the macro has a name that is longer than 5 characters then the first 10 characters of the name are shown.
- If the name is 5 characters or less in length then the top line will show Mname and the bottom line will show then name.
- If the macro has no name then the top line shows Mac.# and the bottom line shows the number.

Pressing **[ENTER]** while viewing a macro name will invoke that macro. If an entry is made while viewing the macro name and then **[ENTER]** is pressed, then that entry will in the entry buffer when the macro starts. Thus the macro should be written to handle that possibility.

SETUP MODE CONFIGURATION

Parameter 90 can be selected as a parameter at P300-P309. Thus any macro may be added to the select menu, similar to the ID Macro Menu on the 660 (when P806 = Menu). While scrolling through the available parameters, the parameter name shown is McNam. After an instance is specified, the actual macro name will show in the lower portion of the character display or the macro number if the name is not programmed. Only macros that are Menu enabled (P9993) are selectable. However, if the parameter and instance are specified at once as is often done in a setup file (i.e. 90.4%e), then any macro (up to the maximum possible macro) may be specified. But if the macro does not exist or is not menu enabled, then upon exiting the setup mode the warning `Check Setup` will appear. In this case, upon pressing a key the indicator will advance to the mode where the invalid parameter has been specified.

MACRO COMMAND %O

You can only **get** the value of 90.xP; it cannot be **set** except at P9991. It will evaluate as the name of the macro if the macro exists, is menu enabled, and the name is not empty. If a name is not specified, the 90.xP reference will evaluate as the number of the macro. Finally, if the macro does not exist or is not menu enabled, it will not be evaluated and thus it will be processed as the actual characters entered (i.e. 90.xP).

ARROW KEYS

When mode 90 is selected in the weigh mode, the arrow keys will operate as arrow keys to allow scrolling in of entries.

WEIGH MODE MESSAGE (MODE 91)

When 91P is selected in the weigh mode, user defined text may be displayed. This allows a macro to define what will be on the display after a macro has ended. Also, this mode allows the user to specify text and/or non-standard formatting of numeric data on the numeric portion of the display.

The mode can be selected either by keying in **91 [SELECT]** in the weigh mode or by assigning 91P to one of the P300 setup mode parameters and then cycling through the selectable modes using the **[SELECT]** key in the weigh mode. At P300, the instance specified is not significant.

ASSIGNING TEXT TO THE 2X5 PROMPTING DISPLAY

Text can be specified using references to P91. The parameter's instances define the specific area of the display being referenced. For example:

`P91.1P=HelloWorld%o`

would store "HelloWorld" as the text to be displayed on the 2X5 prompting portion of the display.

The instance specifies the character position where the text will be displayed. For example, an instance of '1' indicates the top left position and '10' indicates the lower right position.

Assigning data to any portion of the dot matrix area of the display only over-writes the data for the length of the text supplied. Any previous or subsequent text remains unchanged. For example, if after the above example the following were executed:

`91.5P=NO%o`

Then the display would show H e l l N O o r l d, i.e. the "NO" overwrites only the specified positions.

If the data ends in P then the value of that parameter will be inserted. To get the characters to display then enclose any characters ending with P, q, or p in quotes. For example:

`91.5P=0.0P%o`

prints the weight starting at character 5, while:

`91.5P="0.0P"%`

puts 0.0P starting at position 5.

ASSIGNING TEXT TO THE NUMERIC DISPLAY

In order to write to the numeric portion of the display, the instance should be '11'. For example,

`91.11P=HELLO%o`

would cause “HELLO” to appear on the seven-segment numeric display. Data written to the numeric portion is always right justified and if less than the full display is specified, the remainder is blanked. For example,

`91.11P=123%o`

would cause “   123” to show up right justified on the display.

Note that the numeric display only holds 6 digits, not counting decimal points. Up to 6 digits with 6 interspersed decimal points can be displayed. For example,

`91.11P=1.2.3.4.5.6.%o`

would result in “1.2.3.4.5.6.” on the display.

However, since not all alpha characters can be represented in seven segments, certain alpha/symbol characters are allowed. If the supplied character is not possible in the specified case, then the other case will be used. If the character is not at all possible, then three horizontal lines are shown for that character. For example:

`91.11P=aBcDef%o`

will result in “AbcdEF” on the display. However, K, M, Q, V, W, X, and Z are not possible, as well as most symbols.

If the characters to be displayed are in excess of the 6 character limit then the result will be truncated to the right of a decimal point if it exists otherwise “- - - - - -“ will be displayed instead. For example,

`91.11P=1234.56789%o`

will result in “12345.56” on the display, while

`91.11P=abcdefgh%o`

or

`91.11P=1234567%o`

will result in “- - - - - -“ on the display.

Two consecutive decimal points will have a blank inserted between them. For example:

`91.11P=1...4%o`

will result in “   1. . .4” on the display.

Any leading blanks not followed by a decimal will be stripped if the number is too big to be displayed.

If the data ends in P then the value of that parameter will be inserted. To get the characters to display then enclose any strings ending with P, q, or p in quotes. For example,

`91.11P=0.0P%o`

displays a snapshot of the gross weight of the current scale, right justified, while

`91.11P=“0.0P”%`

displays “   0.0P” right justified. (The quote marks do not display or take any character positions!)

Assigning 'Live' Parameters to the Numeric Display

To allow displaying live operating parameters in the seven segment area at the same time as user supplied data is shown in the dot matrix area of the display, a weigh mode parameter (0P – 98P) can be displayed on the seven-segment portion of the display.

In order to cause a weigh mode parameter to be continuously updated on the numeric portion of the display, the instance specified for 90P must be '12'. For example,

`91.12P="0.0P"%o`

The quotes are required, otherwise you will get only a static snapshot of the parameter. The following are equivalent:

`91.12P=0.0P%o`

`91.11P=0.0P%o`

⚠ Note the missing quotes! It is also important to realize that adding quotes has different effect for instances 11 & 12 of parameter 91.

Get Displayed Data (Mode 92)

Parameter 92 is used to return the string data from either the 2x5 character portion or the numeric portion of the display. This data is placed in the entry buffer.

`92.0P%o` place 2x5 character data in entry buffer

`92.10P%o` place numeric data in entry buffer

This parameter not directly accessible from the keypad, nor is it usable in input interpreters, or at p300 → 309. Data can only be retrieved to the entry buffer.

Diagnostic Weight Parameters

The diagnostic weight parameters are used to report information about the displayed weight parameters.

Status (Parameter 97)

The *status* parameter is used in custom transmits to indicate the status (motion, stable, over/under load) of each scale. By default, the status characters transmitted are:

- **O** = Overload/Underload
- **M** = Motion

- **S** = Stable

The transmitted status character(s) can be changed at P143 → P145 in the setup mode.

When transmitting 97P through a custom transmit, the field width is that of the largest string assigned at P143 → P147.

When transmitting **97P** via Modbus, only the first character of the names defined at P143 → P147 is transmitted.

DISPLAYED WEIGHT/COUNT (PARAMETER 98)

The *current displayed weight/count* parameter is used most commonly in custom transmit tables to provide and indication of the current gross, net, tare or quantity value, whichever is currently displayed. If a parameter other than gross, net, tare or quantity is selected as the current operating mode, **98P** will begin reporting only the gross weight.

EXTENDED GROSS (PARAMETER 99)

The *extended gross* parameter is used to momentarily display the current gross weight at 10X the selected division size. This is useful when verifying calibration to determine how close the gross weight is to the next weight division.

Chapter 8 COMMUNICATIONS

This chapter covers all aspects of the communications setup for all 60 Series instruments.

OVERVIEW

COMMUNICATION CONNECTIONS

If communication cables are used, they should be routed into the controller through the rear panel strain relief nearest the communication port connector. These strain relief's are designed to accommodate cables ranging in diameter from 0.187 inches to 0.312 in (4.75 mm to 7.92 mm). Wires can range in size from 28 to 20 AWG. Insulation resistance should be rated at a minimum of 30 volts.

Use a cable with a braid or a foil shield and drain wire. A braided shield will perform better in high electrical noise environments. The capacitance rating of the cable should be low for long cable runs. The shield for the communications cable should be grounded to the closest available rear panel stud inside the controller. Keep the length of the shield between the stud and the end of the cable jacket to an absolute minimum and the length of unshielded wires to a minimum. This is important in order to reduce the effects of EMI, RFI, and ESD during controller operation.

COMM PORT CONNECTIONS (GENERAL)

There are several different ways of communicating from the indicator to another device. Your application will dictate the required connections and the number of conductors. For more information, refer to Table 8-1 through Table 8-2. These tables list several different communication methods and suggested connections.

COMM1 & COMM2 PORT CONNECTIONS

Communication ports 1 & 2 provide identical wiring possibilities for connecting RS-232 communications. Before wiring the port, determine the type of flow control to be used. Then refer to Table 8-1 for the proper connections. You must also configure P204 in the setup mode for the required flow control (see page 3-32).

Table 8-1: COMM1 & COMM2 Port Connections

CONNECTION	FUNCTION
TX1 RX1 GND	Bi-directional with software handshake, or bi-directional with no handshake, or uni-directional with software handshake.
TX1 RX1 CTS RTS GND	Bi-directional with hardware handshake or "both" handshake.
TX1 CTS GND	Uni-directional with hardware handshake (transmit only).

CONNECTION	FUNCTION
RX1 RTS GND	Uni-directional with hardware handshake (receive only).
TX1 GND	Uni-directional with no handshake (transmit only).
RX1 GND	Uni-directional with no handshake (receive only).

COMM3 PORT CONNECTIONS (560/660 SERIES)

Communication port 3 offers the same wiring possibilities as COMM1 & COMM2 with the exception of the RTS output signal for hardware handshaking. COMM3 does not support the RTS signal. Thus hardware handshaking is limited to detecting the CTS input. This makes COMM3 the ideal choice for communicating with uni-directional devices such as printers the make use of the CTS signal for flow control.

COMM4 PORT CONNECTIONS (660 SERIES)

Communication port 4 is a bi-directional RS-232 port that offers only Xon/Xoff software handshaking. Refer to Table 8-2 for proper connections.

COMM4 provides two (2) +5VDC and two (2) digital ground connections. These extra terminals are provided to allow connection of scanners and other digital devices that require an external power source.

The default baud rate and parity for COMM4 is different than the other ports (see Appendix D). The purpose for this is to accommodate the 4X20 VFD and/or the Alpha Keypad option without requiring a change to COMM4's configuration. If you are connecting an alternate device, be sure to match protocols.

Table 8-2: COMM4 Port Connections

CONNECTION	FUNCTION
TX4 RX4 GND	Bi-directional with software handshake, or bi-directional with no handshake, or uni-directional with software handshake.
TX4 GND	Uni-directional with no handshake (transmit only).
RX4 GND	Uni-directional with no handshake (receive only).

4X20 VFD CONNECTION

The 4X20 VF display requires a TTL level communication signal. This is provided by the J7 header on the main board. Do not connect the 4X20 VFD to COMM4 on the J6 terminals!

When connecting the 4X20 VFD to J7, the COMM4 transmit terminal will not be available for connection to another device as J7 is considered to be COMM4. Any transmissions sent out COMM4 will be received by the 4X20 VFD.

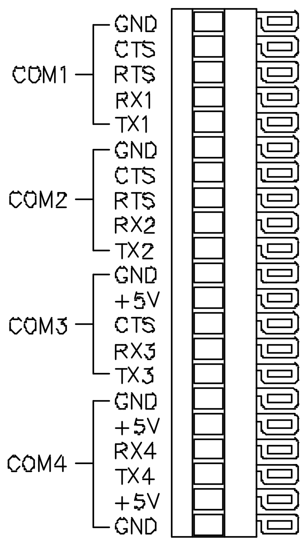


Figure: RS-232 Port (J6)

COMMUNICATIONS CABLES

1. Strip back the jacket of the communications cable 7 inches (178 mm) for the J4 strain relief, and 8 inches (203 mm) for the J2 strain relief.
2. Strip the insulation of each conductor back 0.25 in (6.35 mm).
3. Twist the strands of each wire so no strands are loose. You can tin the wires so the strands will not come loose. Use a minimal amount of solder so that the wire will fit into the connector.
4. Route the cables for any of the COMM ports through an available rear panel strain relief's, J2 or J4. Loosen the strain relief and route the cable(s) through.
5. Ground the cable shield to the stud nearest the strain reliefs.
6. Determine the connections required for your application. (The RS-232 ports and respective functions are labeled on the main PC board. Also refer to Table 8-1 and Table 8-2.)
7. Insert each wire into the proper connection on J6 of the main PC board (see figure – *RS-232 Port*).
8. Secure all of the wires together with a tie wrap near the J6 connector. This will prevent wires from coming loose and into contact with other signals.
9. Once the wires and shield are connected, pull any excess cable out of the controller through the strain relief to eliminate any slack between the shield termination and the strain relief.
10. Tighten the strain relief securely.

COMM PORT SETUP PARAMETERS

When transmitting data to or receiving data from another device, make sure the indicator's protocol matches that of the device. The default communication protocol for all 60 Series instruments is 9600 baud, 8 data bits, 1 stop bit and Xon/Xoff handshaking. Setup parameters governing communication protocol begin at P199 → P204 (see page 3-32). Additional parameters, P205 → P211, provide configuration for receive modes, transmission delays and buffer sizes (see page 3-32).

RECEIVE OPERATIONS

All 60 Series instruments are capable of executing commands received through any of the serial communication ports. This means you can use any external serial device (i.e. computer, scanner, another indicator, etc.) to perform macro operations, assign parameter values, request parameter values, simulate key presses, etc.

RECEIVE BUFFER

The receive buffer for each communication port can be programmed to a specified byte size at P208 in the setup mode. Received characters are stored in this buffer until the indicator has a chance to process the data. Normally, data is processed quickly after it is received. One exception is during macro execution. Received data will continue to be buffered during macro execution until retrieved via macro command or upon termination of all macro execution.

Both software and hardware handshaking are asserted and de-asserted at the same time, regardless of the flow control selected at P204 of the setup mode. Thus, when using software handshaking do not connect the CTS/RTS signals.

If a port is configured for input interpreter or Modbus, received data is transferred to a temporary buffer separate from the receive buffer. There, characters are analyzed simultaneously with macro execution and are processed accordingly.

ASSERTING / DE-ASSERTING HANDSHAKING

Both software and hardware handshaking are asserted and de-asserted at the same time, regardless of the flow control selected at P204 in the setup mode. Thus, when using software handshaking do not connect the CTS/RTS signals.

Handshaking is de-asserted when the receive buffer becomes 75% full based on the maximum buffer size specified at P208 in the setup mode. Handshaking is re-asserted when number of bytes in the buffer drops to 50% of the buffer size.

For example, if you specify a receive buffer size of 4K bytes, the indicator will de-assert handshaking when the receive buffer is 3K bytes full, leaving 1K in reserve. Thus if the transmitting device's transmit buffer is 1K or less, the indicator will be able to receive the additional data while handshaking is de-asserted without resulting in an over-run error (lost data). When the indicator's receive buffer drops to 2K bytes full, handshaking will be re-asserted and data transfer will resume.

MACRO LANGUAGE

The macro language used by the controller is based on the percent (%) character. Any displayable character preceding a percent (%) is considered to be an argument for the macro command. Any character received immediately following a percent character is analyzed as a command. If a received command is invalid, the command is ignored and flushed from the entry buffer. In this case, any text preceding the invalid command will remain in the entry buffer awaiting a valid macro command.

Single-byte values greater than 127 are also treated as macro commands (see Table 8-4).

SIMULATING FRONT PANEL KEYS

Table 8-3 lists the RS-232 keypad macro commands that are used to simulate the front panel keys. These commands can be executed using the actual macro command syntax, or by receiving the single-byte ASCII equivalent.

Table 8-3: Serial Keypad Commands

DESCRIPTION	MACRO SYNTAX	8 BIT HEX	8 BIT DECIMAL	REF.
[F1]		0x80	128	9-8
[F2] / [TARGET]		0x81	129	
[F3]		0x82	130	
[F4]		0x83	131	
[F5]		0x84	132	
[START]		0x85	133	
[STOP]		0x86	134	
[SETUP]		0x87	135	
[SCALE SELECT]	%`	0xE0	224	9-92
[CLR]	%c	0xE3	227	9-102
[ENTER]	%e	0xF5	229	9-102
[ID]	%i	0xE9	233	9-104
[PRINT]	%p	0xF0	240	9-123
[SELECT]	%s	0xF3	243	9-126
[TARE]	%t	0xF4	244	9-127
[UNITS]	%u	0xF5	245	9-129
[CLR] + [SELECT] (macro abort)		0xF8	248	9-17
[ZERO]	%z	0xFA	250	9-150

EXECUTING MACRO COMMANDS

Table 8-4 lists the RS-232 macro commands. These commands can be executed using the actual macro command syntax, or by receiving the single-byte ASCII equivalent.

Table 8-4: Serial Macro Commands

DESCRIPTION		MACRO SYNTAX	8 BIT HEX	8 BIT DECIMAL	REF.
[F1]	(invoke macro 1)		0x80	128	9-8
[F2] / [TARGET]	(invoke macro 2)		0x81	129	
[F3]	(invoke macro 3)		0x82	130	
[F4]	(invoke macro 4)		0x83	131	
[F5]	(invoke macro 5)		0x84	132	
[START]	(invoke macro 6)		0x85	133	
[STOP]	(invoke macro 7)		0x86	134	
[SETUP]	(invoke macro 8)		0x87	135	
-RESERVED-			0x88	136	
-RESERVED-			0x89	136	
-RESERVED-			0x8A	138	
-RESERVED-			0x8B	139	
-RESERVED-			0x8C	140	
-RESERVED-			0x8D	141	
-RESERVED-			0x8E	142	
-RESERVED-			0x8F	143	
-RESERVED-			0x90	144	
-RESERVED-			0x91	145	
-RESERVED-			0x92	146	
-RESERVED-			0x93	147	
-RESERVED-			0x94	148	
-RESERVED-			0x95	149	
-RESERVED-			0x96	150	
-RESERVED-			0x97	151	
-RESERVED-			0x98	152	
-RESERVED-			0x99	153	
-RESERVED-			0x9A	154	
-RESERVED-			0x9B	155	
-RESERVED-			0x9C	156	
-RESERVED-			0x9D	157	

DESCRIPTION	MACRO SYNTAX	8 BIT HEX	8 BIT DECIMAL	REF.
-RESERVED-		0x9E	158	
-RESERVED-		0x9F	159	
Backspace	%<space>	0xA0	160	
Enable / Disable Comm Port	%!	0xA1	161	9-31
Select Comm Port	%”	0xA2	162	9-32
Current Scale	%#	0xA3	163	9-33
Send Text	%$	0xA4	164	9-34
%	%%	0xA5	165	9-4
Send Control Code	%&	0xA6	166	9-35
Record Received Serial Data	%’	0xA7	167	9-36
If Character Received	n%(	0xA8	168	9-37
Clear Receive Buffer	%)	0xA9	169	9-40
Record A/D Data	%*	0xAA	170	9-41
Averaging	%+	0xAB	171	9-44
Motion Delay	%,	0xAC	172	9-45
Perform Scale Specific Function	%-	0xAD	173	9-45
-RESERVED-	%.	0xAE	174	
If Macro Interrupted	%/	0xAF	175	9-46
-RESERVED-	%0	0xB0	176	
-RESERVED-	%1	0xB1	177	
-RESERVED-	%2	0xB2	178	
-RESERVED-	%3	0xB3	179	
-RESERVED-	%4	0xB4	180	
-RESERVED-	%5	0xB5	181	
-RESERVED-	%6	0xB6	182	
-RESERVED-	%7	0xB7	183	
-RESERVED-	%8	0xB8	184	
-RESERVED-	%9	0xB9	185	
-RESERVED-	%:	0xBA	186	
-RESERVED-	%;	0xBB	187	
-RESERVED-	%<	0xBC	188	
-RESERVED-	%=	0xBD	189	
-RESERVED-	%>	0xBE	190	
-RESERVED-	%?	0xBF	191	
Set Pause Time	%@	0xC0	192	9-47
Activate Setpoint	%A	0xC1	193	9-48

DESCRIPTION	MACRO SYNTAX	8 BIT HEX	8 BIT DECIMAL	REF.
Break Macro	%B	0xC2	194	9-51
Display Text (4X20 VFD / LCD)	%C	0xC3	195	9-53
Deactivate Setpoint	%D	0xC4	196	9-59
End If	%E	0xC5	197	9-61
If Setpoint Deactivated	%F	0xC6	198	9-61
Get Entry	%G	0xC7	199	9-62
Redefine Comm Port Function	%H	0xC8	200	9-64
Refresh Display	%I	0xC9	201	9-66
Jump to Tag	%J	0xCA	202	9-66
Get Entry (4X20 VFD / LCD)	%K	0xCB	203	9-68
Language Selection	%L	0xCC	204	9-73
Mode Selection	%M	0xCD	205	9-74
Else	%N	0xCE	206	9-75
If Setpoint Activated	%O	0xCF	207	9-75
Pause	%P	0xD0	208	9-76
Send Custom Transmit	%Q	0xD1	209	9-77
Rename Mode	%R	0xD2	210	9-80
Sound Beeper	%S	0xD3	211	9-81
Tag Position	%T	0xD4	212	9-83
Transmit Buffer	%U	0xD5	213	9-85
-RESERVED-	%V	0xD6	214	
Wait for Keypress	%W	0xD7	215	9-85
Request Display Data	%X	0xD8	216	9-86
If Yes	%Y	0xD9	217	9-87
-RESERVED-	%Z	0xDA	218	
Save Entry Buffer	%[	0xDB	219	9-87
IF No Entry	%\	0xDC	220	9-88
Restore Entry Buffer	%]	0xDD	221	9-88
Call / Go To Macro	%^	0xDE	222	9-89
If Database Error	%_	0xDF	223	9-91
Scale Select	%`	0xE0	224	9-92
Target Accuracy	%a	0xE1	225	9-93
Perform Sample	%b	0xE2	226	9-94
Clear Entry Buffer	%c	0xE3	227	9-94
Display Control	%d	0xE4	228	9-97
Enter / Sample	%e	0xE5	229	9-102

DESCRIPTION	MACRO SYNTAX	8 BIT HEX	8 BIT DECIMAL	REF.
If Parameter Preset	%f	0xE6	230	9-103
Sample / Macro Error	%g	0xE7	231	9-104
-RESERVED-	%h	0xE8	232	
ID	%i	0xE9	233	9-104
If Key / Remote Key Held	%j	0xEA	234	9-105
Digital Filter	%k	0xEB	235	9-106
-RESERVED-	%l	0xEC	236	
Modify String	%m	0xED	237	9-108
Get Numeric Entry	%n	0xEE	238	9-111
Math Operation	%o	0xEF	238	9-113
Print	%p	0xF0	240	9-123
Enable RS-485 Transmitter	%q	0xF1	241	9-123
A/D Interval	%r	0xF2	242	9-124
Select Mode	%s	0xF3	243	9-126
Tare	%t	0xF4	244	9-127
Units	%u	0xF5	245	9-129
Write Value to EEPROM	%v	0xF6	246	9-130
-RESERVED-	%w	0xF7	247	
Macro Abort	%x	0xF8	248	9-17
Database Operation	%y	0xF9	249	9-131
Zero	%z	0xFA	250	9-150
Start Group	%{	0xFB	251	9-152
Or	%|	0xFC	252	9-152
End Group	%}	0xFD	253	9-152
-RESERVED-	%~	0xFE	254	
-RESERVED-	%DEL	0xFF	255	

CUSTOM TRANSMIT

A *custom transmit* is a sequence of characters, control codes and parameter values to be transmitted out a communication port to a peripheral device such as a printer, remote display, computer or another weight indicator. As the name suggests, each custom transmit may be configured to send data in virtually any format.

CUSTOM TRANSMIT SETUP PARAMETERS

Memory permitting, each custom transmit can contain up to 5000 characters. Setup parameters P989 → P998 set the criteria for initiating each custom transmit (i.e. which comm port, motion delayed, continuous, etc.). Refer to page 3-43 for more details.

Custom transmit tables can be protected from being viewed, edited and downloaded in the setup mode through use of the limited access code (see *Limited Access* on page 3-4). To protect individual custom transmit tables through limited access, enable limited access at P999.

The actual custom transmit table begins at P1000 → P4999.

DEFAULT CUSTOM TRANSMIT

When an indicator is defaulted, custom transmit #1 will generate the following output out comm port 1:

```
    0.00 lb    Gross
    0.00 lb    Tare
    0.00 lb    Net
```

Each line in the default custom transmit is transmitted in the following format:

< weight > < space > < units > < space > < mode > < CR > < LF >

where *weight* is an 8-digit value (including decimal), *units* is a 5-character units identifier (right spaces filled), and *mode* is a 5-character mode identifier (right spaces filled).

SENDING A CUSTOM TRANSMIT

There are three ways to initiate a custom transmit:

1. Pressing the **[PRINT]** key will send a custom transmit out the port specified at P992 provided P991 is set for ‘on request’ or ‘prompt’.

   Every custom transmit set for ‘on request’ will begin transmission when **[PRINT]** is pressed. Custom transmits will be sent sequentially in ascending order by transmit number.

   If any custom transmits are set for ‘prompt’, pressing **[PRINT]** will first display the prompt “***Which Tx# ?***”. This gives you the opportunity to choose one of several transmits. For example, a vendor may supply components to ten different customers. The vendor can create ten custom transmits, each with a different name, address and phone number to be printed on the shipping label. The operator can then print the appropriate label by pressing **[PRINT]** to display the "***Which Tx# ?***" prompt and then key in the desired transmit number and press **[ENTER]** to print the label.

Note that only one 'prompt' transmit can be sent at a time. After a 'prompt' transmit is sent, any transmits set for 'on request' will also be sent.

2. Custom transmits can be sent automatically on a continual basis by setting P989 to 'continuous'. The time between each continuous transmit is defined at P980 (transmit rate). Continuous transmits are used to send data to devices such as scoreboards, remote displays and computers.
3. A macro command can initiate a custom transmit. This is possible even if P991 is set to 'off'. This allows custom transmits to be event driven such as having weight tickets printed automatically each time a new weight is applied.

DEFINING A NEW CUSTOM TRANSMIT

To define a new custom transmit:

1. Enter the setup mode and access P989.
2. Key in the desired transmit number and press **[ENTER]**.
3. If the transmit does not exist, "***Make? NewTx***" is displayed. Press **[ENTER]** to confirm.
4. Select P990 and enter a name for the custom transmit if desired. The name is only used for documentation purposes.
5. Select P991 and select whether the transmit will be 'off', 'on request' or 'prompt'.
6. Select P992 and select which communication port the transmit should use.
7. Select P993 – P994 and select whether the transmit should 'ignore' the motion of each scale or be 'inhibited' by it.
8. If a custom transmit is to be 'continuous', enable P998.
9. Select P1000 to access the custom transmit table.

CREATING A CUSTOM TRANSMIT TABLE

The custom transmit table begins at P1000 in the setup mode. It contains the information to be transmitted. When a custom transmit is first defined, the transmit table is empty. Text, parameters and control codes must be entered in the proper sequence to achieve the desired output. For example, suppose we wish to print the following ticket:

```
GSE Scale Systems

   50.00 lb    Gross
    5.00 lb    Tare
   45.00 lb    Net
```

This ticket begins with the text “GSE Scale Systems” on the first line, followed by one blank line, followed by three consecutive lines of parameter information. The custom transmit will be configured in the same sequence. The following sections on entering fixed text, control codes and parameters builds on this example.

ENTERING FIXED TEXT

Fixed text can be entered into any custom transmit at any position within the transmit table. This is useful for incorporating text headers and other text information that will never change.

```
GSE Scale Systems

   50.00 lb    Gross
    5.00 lb    Tare
   45.00 lb    Net
```

Using the example above, the fixed text header “GSE Scale Systems” is to appear on every printed ticket. Since this is the first element of the transmit table, begin entering the header text at P1000.

To enter text using the alpha keypad or serial port, simply key in the desired character(s) and press **[ENTER]**.

To enter text using the scrolling method proceed as follows:

1. Press **[▲]** to scroll through the character set.
2. Once the desired character is displayed, press **[▶]** to shift right to the next character position.
3. Repeat steps 1 and 2 until the entire name is displayed ending at step 1.
4. Press **[ENTER]** to accept insert the entered text into the transmit table.

Note: Pressing **[▼]** will scroll backwards through the character set.
Pressing **[◀]** will shift left, or backspace.
Pressing **[CLR]** will delete an entry in process.

Note that the text appears in the custom transmit table exactly as entered.

ENTERING CONTROL CODES

Control codes are generally considered to be non-displayable ASCII characters such as a carriage return < CR > or line feed < LF >. Since you cannot *see* these characters, control codes must be entered using their ASCII value as shown below. Refer to the ASCII chart in Appendix B for a complete list of control codes.

Having entered the fixed text for the header line of the ticket in the previous example, we must next account for the blank line that follows and position the print 'cursor' at the beginning of the third line (the 'cursor' now resides at the end of '*Systems*'). Positioning the 'cursor' requires use of special non-displayable characters call *control codes*. Our example requires the use of two common control codes – carriage return <CR> and line feed <LF>.

> A carriage return <CR> positions the 'cursor' to the left-most position of the current line.
>
> A line feed <LF> moves the 'cursor' down one line without moving it left or right.

Therefore, to move the 'cursor' from the end of the first line to the beginning of the third line we must transmit a carriage return and two line feeds <CR><LF><LF>. Similarly, a <CR> and <LF> will be required after each subsequent line as illustrated below. Note the form feed <FF> at the end of the ticket. This is another control code which is used to advance the printer paper to the top of the next ticket.

```
GSE Scale Systems<CR><LF>
<LF>
   50.00 lb    Gross<CR><LF>
    5.00 lb    Tare<CR><LF>
   45.00 lb    Net<CR><LF>
<FF>
```

A control code must be entered into the custom transmit table using it's three-digit ASCII value preceded by a decimal '.'. Reference an ASCII chart in Appendix B for these and other character values. For example, to enter the carriage return, key in

.013 [ENTER]

The control code appears in the transmit table as a single $^{c}{}_{c}$ character.

Enter the two line feed characters next:

.010 [ENTER]

.010 [ENTER]

The transmit table now shows the three control codes $^{c}{}_{c}{}^{c}{}_{c}{}^{c}{}_{c}$.

Note that four lines in our example use the <CR><LF> combination. Since this is a very common combination of control codes, a single entry unique to GSE indicators was created to make entering these characters more convenient. Keying in

.256 [ENTER]

will enter the <CR><LF> combination.

Any other ASCII character, including printable characters, may be entered using this method. Referencing the ASCII chart in Appendix B, 'GSE' could have been entered as:

```
.071 [ENTER]   G
.083 [ENTER]   S
.069 [ENTER]   E
```

CUSTOM GSE CONTROL CODES

When communicating with the LCD or 4X20 VF displays, many of the standard ASCII control codes are used to perform specific functions or display custom characters. Additionally, many custom control codes were created to transmit other status information. Refer to Table 8-5 for a complete list of these codes. All custom control codes are entered in the same manner as standard control codes using the three-digit ASCII value preceded by a decimal '.'.

Table 8-5: Custom GSE Control Codes

DECIMAL VALUE (CONTROL CODE)	DESCRIPTION			REFERENCE
LCD Control Codes (Character Set)				
000 → 181	Transmit standard LCD characters (Comm Port 5; refer to LCD character set)			C-1
800 → 866	Transmit custom LCD characters (Comm Port 5 only; refer to LCD character set)			
4X20 VFD Control Codes (Custom Display Commands)				
008	Move cursor left one space	Backspace	<BS>	13-60
009	Move cursor right one space	Horizontal Tab	<HT>	
010	Move cursor down one row	Line Feed	<LF>	
012	Move cursor to top left corner (home)	Form Feed	<FF>	
013	Move cursor to beginning of line	Carriage Return	<CR>	
014	Clear entire display	Shift Out	<SO>	
017	Scroll mode OFF	Device Control 1	<DC1>	
018	Scroll mode ON	Device Control 2	<DC2>	
020	Cursor OFF	Device Control 4	<DC4>	
021	Cursor ON	Negative Acknowledge	<NAK>	
024	Set INTERNATIONAL character set	Cancel	<CAN>	
025	Set KATAKANA character set	End of Medium	<EM>	
027, 067	Define custom character	Escape + C	<ESC> + 'C'	
027, 072	Position cursor at specified location	Escape + H	<ESC> + 'H'	
027, 073	Reset display default values	Escape + I	<ESC> + 'I'	
027, 076	Set display intensity	Escape + L	<ESC> + 'L'	
027, 084	Set cursor blink speed	Escape + T	<ESC> + 'T'	
4X20 VFD Control Codes (Character Set)				
032 → 213	Transmit standard VFD characters (Comm Port 5; refer to LCD character set)			C-1
Combination ASCII Control Codes				
256	Insert both carriage return and line feed <CR> <LF>			8-22
Checksum Control Codes				
300	Stop checksum calculation (do not transmit)			8-41
301	Start CCITT checksum calculation			
302	Start SDLC/HDLC checksum calculation			
303	Start CRC-16 checksum calculation (initial checksum value = 0000)			
304	Start CRC-12 checksum calculation			
305	Start LRCC-16 checksum calculation			
306	Start LRCC-8 checksum calculation			
307	Start XMODEM checksum calculation			
308	Start SUM-16 checksum calculation			
309	Start SUM-8 checksum calculation			

DECIMAL VALUE (CONTROL CODE)	DESCRIPTION	REFERENCE
310	Transmit checksum (LSB first)	
311	Transmit checksum (MSB first)	
312	Start CRC-16 checksum calculation (initial checksum value = FFFF)	
Scale# Control Code		
350	Transmit current scale number (1 → 8)	8-45
Scale Status Control Code		
360	Transmit current range of current scale (1=low range; 2=medium range; 3=high range; 0=unavailable)	
361	Transmit current range of scale 1 (1=low range; 2=medium range; 3=high rang; 0=unavailable)	
362	Transmit current range of scale 2 (1=low range; 2=medium range; 3=high range; 0=unavailable)	
363	Transmit current range of scale 3 (1=low range; 2=medium range; 3=high range; 0=unavailable)	
364	Transmit current range of scale 4 (1=low range; 2=medium range; 3=high range; 0=unavailable)	
365	Transmit current range of scale 5 (1=low range; 2=medium range; 3=high range; 0=unavailable)	
366	Transmit current range of scale 6 (1=low range; 2=medium range; 3=high range; 0=unavailable)	
367	Transmit current range of scale 7 (1=low range; 2=medium range; 3=high range; 0=unavailable)	
368	Transmit current range of scale 8 (1=low range; 2=medium range; 3=high range; 0=unavailable)	
Binary-to-Text Conversion Control Codes		
400	End binary-to-text conversion	
401	Start binary-to-text conversion	8-45
402	Transmit network address (P251) as a single ASCII character	
Setpoint Status Control Codes		
501	Transmit setpoint status byte (SP# 1 → 8)	
502	Transmit setpoint status byte (SP# 9 → 16)	
503	Transmit setpoint status byte (SP# 17 → 24)	
504	Transmit setpoint status byte (SP# 25 → 32)	
505	Transmit setpoint status byte (SP# 33 → 40)	
506	Transmit setpoint status byte (SP# 41 → 48)	
507	Transmit setpoint status byte (SP# 49 → 56)	
508	Transmit setpoint status byte (SP# 57 → 64)	
509	Transmit setpoint status byte (SP# 65 → 72)	
510	Transmit setpoint status byte (SP# 73 → 80)	
511	Transmit setpoint status byte (SP# 81 → 88)	
512	Transmit setpoint status byte (SP# 89 → 96)	
513	Transmit setpoint status byte (SP# 97 → 104)	
514	Transmit setpoint status byte (SP# 105 → 112)	
515	Transmit setpoint status byte (SP# 113 → 120)	
516	Transmit setpoint status byte (SP# 121 → 128)	
517	Transmit setpoint status byte (SP# 129 → 136)	8-46
518	Transmit setpoint status byte (SP# 137 → 144)	
519	Transmit setpoint status byte (SP# 145 → 152)	
520	Transmit setpoint status byte (SP# 153 → 160)	
521	Transmit setpoint status byte (SP# 161 → 168)	
522	Transmit setpoint status byte (SP# 169 → 176)	
523	Transmit setpoint status byte (SP# 177 → 184)	
524	Transmit setpoint status byte (SP# 185 → 192)	
525	Transmit setpoint status byte (SP# 193 → 200)	
526	Transmit setpoint status byte (SP# 201 → 208)	
527	Transmit setpoint status byte (SP# 209 → 216)	
528	Transmit setpoint status byte (SP# 217 → 224)	
529	Transmit setpoint status byte (SP# 225 → 232)	
530	Transmit setpoint status byte (SP# 233 → 240)	
531	Transmit setpoint status byte (SP# 241→ 248)	
532	Transmit setpoint status byte (SP# 249 → 256)	
Units Control Codes (The following units codes will transmit "Disbl" for disabled scales)		
600	Transmit current units (minimum width) of the currently selected scale	
601	Transmit units #1 (minimum width) of the currently selected scale	
602	Transmit units #2 (minimum width) of the currently selected scale	
603	Transmit units #3 (minimum width) of the currently selected scale	
604	Transmit units #4 (minimum width) of the currently selected scale	
610	Transmit current units (minimum width) of scale #1	
611	Transmit units #1 (minimum width) of scale #1	
612	Transmit units #2 (minimum width) of scale #1	
613	Transmit units #3 (minimum width) of scale #1	
614	Transmit units #4 (minimum width) of scale #1	

DECIMAL VALUE (CONTROL CODE)	DESCRIPTION	REFERENCE
620	Transmit current units (minimum width) of scale #2	
621	Transmit units #1 (minimum width) of scale #2	
622	Transmit units #2 (minimum width) of scale #2	
623	Transmit units #3 (minimum width) of scale #2	
624	Transmit units #4 (minimum width) of scale #2	
630	Transmit current units (minimum width) of scale #3	
631	Transmit units #1 (minimum width) of scale #3	
632	Transmit units #2 (minimum width) of scale #3	
633	Transmit units #3 (minimum width) of scale #3	
634	Transmit units #4 (minimum width) of scale #3	
640	Transmit current units (minimum width) of scale #4	
641	Transmit units #1 (minimum width) of scale #4	
642	Transmit units #2 (minimum width) of scale #4	
643	Transmit units #3 (minimum width) of scale #4	
644	Transmit units #4 (minimum width) of scale #4	
650	Transmit current units (minimum width) of scale #5	
651	Transmit units #1 (minimum width) of scale #5	
652	Transmit units #2 (minimum width) of scale #5	
653	Transmit units #3 (minimum width) of scale #5	
654	Transmit units #4 (minimum width) of scale #5	
660	Transmit current units (minimum width) of scale #6	
661	Transmit units #1 (minimum width) of scale #6	
662	Transmit units #2 (minimum width) of scale #6	
663	Transmit units #3 (minimum width) of scale #6	
664	Transmit units #4 (minimum width) of scale #6	
670	Transmit current units (minimum width) of scale #7	
671	Transmit units #1 (minimum width) of scale #7	
672	Transmit units #2 (minimum width) of scale #7	
673	Transmit units #3 (minimum width) of scale #7	
674	Transmit units #4 (minimum width) of scale #7	
680	Transmit current units (minimum width) of scale #8	
681	Transmit units #1 (minimum width) of scale #8	
682	Transmit units #2 (minimum width) of scale #8	
683	Transmit units #3 (minimum width) of scale #8	
684	Transmit units #4 (minimum width) of scale #8	
690	Transmit default units specified at P150 (minimum width)	
LCD Control Codes		
701 → 740	Position cursor at column 1 → 40 respectively	
741	Select normal text (black on white)	
742	Select inverse text (white on black)	
743	Clear to end of row	
744	Clear to end of display	
751 → 758	Position cursor at row 1 → 8 respectively	
761	Select small font size	
762	Select medium font size	
764	Select large font size	
771 → 786	Position cursor at row 1 → 16 respectively	
LCD Custom Characters		
800 → 866	Transmit custom LCD characters (Comm Port 5 only; refer to LCD character set)	

ENTERING PARAMETER DATA

Operating parameters may be entered in a custom transmit table. Parameters may be formatted to include the parameter's value, units (if applicable) and name. See Table 7-1 on page 7-3 for a complete list of operating parameters.

To enter a parameter into the transmit table:

1. Press **[ENTER]**
2. The display prompts ***Pick Parm:***, ***Parm= Gross***.
3. Select the desired parameter by:
   - Pressing **[▲]** or **[▼]** to scroll through the parameter list, or
   - Keying in the parameter number and instance if applicable (you will be prompted ***Pick Inst****:* by the indicator if an instance is required).
4. The display prompts ***Set Formt***, ***Formt 00000***.
5. Key in the desired format and press **[ENTER]** (see *Parameter Format Codes* below).
6. The display confirms the entered format.
7. Press **[ENTER]** to accept the parameter configuration.
8. The formatted parameter is entered into the transmit table as a single $^{P}{}_{A}$ character.

For example, the third line of the ticket in the previous example

```
50.00 lb    Gross
```

contains the gross weight value (50.00), the weigh units (lb) and the parameter name (Gross).

To enter this parameter into the transmit table:

1. Press **[ENTER]**
2. The display prompts ***Pick Parm:***, ***Parm= Gross***.
3. Key in the gross weight parameter for scale #1, **0.1**, and press **[ENTER]**.
4. The display prompts ***Set Formt***, ***Formt 00000***.
5. Press **0 [ENTER]**.
6. The display confirms the entered format.
7. Press **[ENTER]** to accept the parameter configuration.

To complete the example, enter a <CR><LF> after the gross weight parameter and repeat steps 1 – 7 substituting the tare and net parameter numbers at step 3. Remember the <CR><LF> control codes after each parameter. Also include the <FF> character at the end.

PARAMETER FORMAT CODES

A format code defines how a parameter value is to be transmitted. For example, a customer may prefer to have the parameter names Gross, Tare and Net appear to the left of the weight values instead of the to the right. Perhaps the weight values should be left-padded with zeros instead of spaces. A time value may be printed in 12 or 24 hour format, with or without the date. Do you want to include seconds? Include the name of the day? These and many more variations are possible using format codes. Refer to Table 8-6 through Table 8-13 for a list of available formats.

Each type of parameter (floats, integers and strings) has it's own format code selections. Keep this in mind when selecting format codes for weight parameters, time & date parameters, variables, etc. Refer to the appropriate format table.

Establishing the desired format code is a simple matter of addition. First decide how you want the data to appear, then add the various format codes that will generate the desired output. For example, suppose you want the gross weight to appear as follows:

`Gross +00050.00 lb`

The criteria would be parameter name first, weight padded with left zeros, print a plus sign, and print current units. Since the gross weight is a float-type parameter, we must reference Table 8-9 for the appropriate format codes. Looking down the "Add Value" column, begin adding the numbers that will result in the desired format:

ADD VALUE	DESCRIPTION
32768	Print Name First
64	Plus Sign for Positive Data
16	Pad With Zeros
32848	***FORMAT CODE***

NAVIGATING A CUSTOM TRANSMIT TABLE

If we could examine the entire transmit table from our example, it would appear as:

`GSE Scale Systems` $^{c}_{c}\,^{c}_{c}\,^{c}_{c}\,^{P}_{A}\,^{c}_{c}\,^{c}_{c}\,^{P}_{A}\,^{c}_{c}\,^{c}_{c}\,^{P}_{A}\,^{c}_{c}\,^{c}_{c}\,^{c}_{c}$

Of course the display will only show five characters of the transmit table at once. Using the **[◄]** and **[►]** keys you can scroll backward and forward through the transmit table. Text characters are easy to identify as they appear exactly as entered. However the $^{c}_{c}$ and $^{P}_{A}$ characters can make the custom transmit data appear rather cryptic. We know that $^{c}_{c}$ represents a control code, but which one? Likewise, $^{P}_{A}$ represents an unidentified parameter of unknown format.

```
P101*  Tx   1
       tems?
```

To determine what these characters represent, press **[▼]** to access the *expanded* display mode.

```
P101*  <CR>
       ^M=13
```

The display now shows the details of the rightmost display character. Using the **[◄]** and **[►]** keys you can now scroll backward and forward through the transmit table in the *expanded* mode, examining details of each character in sequence. Text characters and control codes are shown with their corresponding ASCII value. Press **[▼]** again to return to the normal display mode.

When viewing parameters in the expanded mode, parameter names can be displayed with format codes.

```
P102!  Tx   1
       s???,
```

To determine what parameter is represented by the $^{P}_{A}$ character, press **[▼]** to access the *expanded* display mode.

```
P102!  00000
       Gross
```

To view the parameter number and instance of the expanded parameter, press **[▼]** a second time.

```
P102!  00000
        0   1
```

The first digit (0) represents the parameter number (in this case Gross) and the second number identifies the instance (scale #1). Using the **[◄]** and **[►]** keys you can scroll backward and forward through the transmit table in the *expanded* mode, examining details of each parameter in sequence. Text characters and control codes are shown with their corresponding ASCII value.

Press **[▼]** a third time to return to the normal display mode.

The parameter number appearing in the large digits represents the position of the rightmost displayed character.

```
P102!  Tx    1
       s???,
```

This information is helpful when attempting to access the middle of a large transmit table. If you wish to access to 500th element of a table, rather than scrolling to that position you can simply key in **1500 [SELECT]**. To immediately access the end of a custom transmit table, key in **4999 [SELECT]** (the maximum entry position).

EDITING A CUSTOM TRANSMIT TABLE

Information may be deleted from or added to an existing custom transmit table. Pressing **[CLR]** will delete the rightmost displayed character. To delete the entire custom transmit, access the end of the table and press **[CLR] [ENTER]**. When adding elements, characters are *inserted* between the two rightmost characters.

If a parameter's format code must be changed, you must first delete the parameter, then re-enter it using the desired format code.

Table 8-6: INT/U-INT Parameters with Numeric Value Output Format Codes

GROUP	CHOICES	
	Add Value	**Description**
Name	32768	Print Name First
	16384	Omit Name
	0	Print Name Last
Standard Time/Date	* 2048	Standard Time/Date
2-Digit / 4-Digit Year:	128	4-digit year (YYYY).
	0	2-digit year (YY).
Omit Date:	64	Do not include the date.
	0	Include the date in the output..
Name of Day of Week:	32	Include the name of the day.
	0	Do not include the name of the day.
Date Format	16	Use international date format.
	0	Use U.S.A. format for date.
Text Date:	8	Use text format for date.
	0	Use numeric format for date.
Omit Time:	4	Do not include the time.
	0	Include the time in the output..
24 Hour Clock	2	Use military time (0 - 23 hours).
	0	Use 12 hour clock with am/pm.
Seconds:	1	Include seconds with time.
	0	Do not include seconds with time.

***You must add this value for this table to take effect.**

Table 8-7: INT/U-INT Parameters with Extended Time Date Output Format Codes

GROUP	CHOICES	
	Add Value	**Description**
Name	32768	Print Name First
	16384	Omit Name
	0	Print Name Last
Type	* 4096	Extended Time/Date Codes
Separator	1792	Separate numbers with comma (,).
	1536	Separate numbers with period (.).
	1280	Separate numbers with backslash (\).
	1024	Separate numbers with space ( ).
	768	Separate numbers with slash (/).
	512	Separate numbers with colon (:).
	256	Separate numbers with dash (-).
	0	Do not separate numbers.
Julian Date	128	Include Julian date.
(day of the year)	0	Do not include Julian date.
Seconds since midnight:	64	Include the seconds since midnight.
	0	Do not include the seconds since midnight.
Hours Digits:	32	Include hours digits.
	0	Do not include hours digits.
Minutes Digits:	16	Include minutes digits.
	0	Do not include minutes digits.
Seconds Digits:	8	Include seconds digits.
	0	Do not include seconds digits.
Date Codes:	7	WWYY (week and year)
	6	YYWW (year and week)
	5	DDMMYY (day, month, year)
	4	DDYYMM (day, year, month)
	3	YYDDMM (year, day, month)
	2	YYMMDD (year, month, day)
	1	MMDDYY (month, day, year)

Table 8-8: INT/U-INT Parameters with Standard Time Date Output Format Codes

GROUP	CHOICES	
	Add Value	**Description**
Name	32768	Print Name First
	16384	Omit Name
	0	Print Name Last
Plus Sign	64	Print plus sign if positive data.
	0	Don't print '+'.
Justification: Left/Right	32	Left Justify.
	0	Right Justify.
Zero/Space Fill	16	Pad with zeroes.
	0	Pad with spaces.
Width	2 ... 15	Specifies the minimum width.
	1	Print with minimum width.
	0	Use print width specified by P240.

Table 8-9: Float Parameters Format Codes

GROUP	CHOICES	
	Add Value	Description
Name	49152	* Print Name Only
	32768	Print Name First
	16384	Omit Name
	0	Print Name Last
Binary		Refer to Table 8-12
Convert to Whole number.	4096	Convert data to whole number.
	0	Print data normal.
Which Units:	2560	Default Units (per P150)
	2048	Fourth Units (per P134)
	1536	Third Units (per P133)
	1024	Second Units (per P132)
	512	First Units (per P131)
	0	Current Units (as currently displayed)
Print Units	256	Don't Print Units
	0	Print Units Name
Print DP Always	128	Always print a decimal point.
	0	No decimal point if no fractional portion.
Plus Sign	64	Print plus sign if positive data.
	0	Don't print '+'.
Justification: Left/Right	32	Left Justify.
	0	Right Justify.
Zero/Space Fill	16	Pad with zeroes.
	0	Pad with spaces.
Width	2 ... 15	Specifies minimum width
	1	Print with minimum width
	0	Use print width specified by P240.

Table 8-10: String Type Parameter Format Codes

GROUP	CHOICES	
	Add Value	Description
Name	32768	Print Name First
	16384	Omit Name
	0	Print Name Last
Justification: Left/Right	128	Left Justify.
	0	Right Justify.
Width	2 ... 127	Value specifies the minimum width to output.
	1	Output with minimum possible width.
	0	Use print width specified by P240.

Table 8-11: Parameter Name Output Format Codes

GROUP	CHOICES	
	Add Value	Description
Name	* 49152	Print Name Only
Justification: Left/Right	128	Left Justify.
	0	Right Justify.
Width	2 ... 127	Value specifies the minimum width to output.
	1	Output with minimum possible width.
	0	Use print width specified by P240.

***You must add this value for this table to take effect.**

Table 8-12: Format Codes for Binary Output of Float Type Parameter's Value

GROUP	CHOICES	
	Add Value	Description
Name	32768	Print Name First
	16384	Omit Name
	0	Print Name Last
Binary	* 8192	Print binary data
Convert to Whole Number.	4096	Convert data to whole number.
	0	Keep as fractional value.
Which Units:	2560	Default Units (per P150)
	2048	Fourth Units (per P134)
	1536	Third Units (per P133)
	1024	Second Units (per P132)
	512	First Units (per P131)
	0	Current Units (as currently displayed)
Send as comma delimited ASCII decimal values	4	Output each byte as ASCII decimal value with commas (,) between
	0	Send binary data.
Output Type:	3	Output 4 byte float value.
	2	Output 4 byte integer value.
	1	Output 2 byte integer value.
	0	Output 1 byte integer value.

***You must add this value for this table to take effect.**

Table 8-13: Format Codes for Binary Output of INT/U-INT Type Parameter’s Values

GROUP	CHOICES	
	Add Value	Description
Name	32768	Print Name First
	16384	Omit Name
	0	Print Name Last
Binary	* 8192	Print binary data
Send as comma delimited ASCII decimal values	4	Output each byte as ASCII decimal value with commas (,) between
	0	Send binary data.
Output Type:	3	Output 4 byte float value.
	2	Output 4 byte integer value.
	1	Output 2 byte integer value.
	0	Output 1 byte integer value.

***You must add this value for this table to take effect.**

TRANSMITTING DISPLAY DATA

A series of single byte ASCII commands allow a remote device to request the indicator to send its displayed information. This can be very useful in setting up communications with a remote indicator via a modem connection. The commands are not available for use within a macro since they would require the entry of control codes within macro setup. They must be sent to the indicator as a single byte code. A separate code identifies each comm port as listed in Table 8-14.

Table 8-14: Communication Port ASCII Commands

PORT	DECIMAL CODE	HEX CODE
COMM1	<149>	0x95
COMM2	<150>	0x96
COMM3	<151>	0x97
COMM4	<152>	0x98

When one of these four codes is received on a Comm port that is set for Receive Standard (P205, choice 1), the 650 will transmit the currently displayed information out of the specified port in a fixed width format.

The data is sent in normal order as follows: (numeric display, top line auxiliary display, bottom line auxiliary display) followed by <CR> <LF> only. These commands cannot be included in a macro because they are %<CC> (i.e. control codes). They may be sent to the indicator (if 8 data bits is enabled) as stated in the table above.

Transmitted data format is:

```
-1.2345        lb    Gross<CR><LF>
|              |     |    |
0              15    21   26
```

i.e. The numeric data is sent first and right padded with spaces to allow the top line data to appear in the 15th position. Another space is sent after the 5 top line characters, then the bottom line is sent, followed by a <CR> <LF>. This is quite valuable for modem connections to a 60 Series instrument in the field for troubleshooting purposes.

MODBUS ä COMMUNICATIONS

Modbus™ is a serial communications protocol supported by a variety of PLCs and other industrial equipment. It is an asynchronous serial protocol that does not specify the baud rate, or parity bit. It does specify 7 data bits and either one stop bit with parity or two stop bits without parity. It is a master-slave protocol where the master device initiates communications with the slave and the slave responds accordingly. Modbus may be used with either RS-232 or RS-485 at the physical layer.

The transfer of numeric and alpha data via Modbus is based upon each data item having a register address within the slave. The master then specifies the address of the data item to sent to or received from the slave and the slave accepts the received data or sends the specified data in a format that is intended to be independent of data type.

Modbus should not be confused with Modbus Plus which is a proprietary protocol owned by Modicon, Inc.

SUPPORTED MODBUS COMMANDS

Following are brief descriptions of the Modbus commands supported by 60 Series instruments. They are listed with the Modbus function code followed by the function name.

READ COIL STATUS/READ INPUT STATUS

Since the 60 Series instruments do not use separate numbering for input and output setpoints (coils) both of these commands will read the status of the specified setpoint number, regardless of type.

READ HOLDING REGISTERS

This command contains addresses offset from address 40001. A programmable translation table in the 60 Series instrument is used to translate register addresses to parameter numbers and instances. Each row of the translation table contains three elements:

Parameter Number

Instance

Translation Type

FORCE MULTIPLE COILS

Allows a several sequential setpoint output states to be specified.

PRESET MULTIPLE REGISTERS

Allows multiple parameters within the indicator to be given new data.

SETUP PARAMETERS

ENABLING MODBUS

P205 "**RecvX**" now has another choice, "Modbs". This selection specifies that all communications (both transmit and receive) on the current serial port are Modbus format.

While Modbus is specified for a given COMM port, then that port cannot be used to send data native 60 Series commands to the indicator. Also, any attempt to send data out a Modbus Comm port (using the will result in the "NoTxX Allow" message being displayed briefly.

At this time, the indicator can only act as a slave on the Modbus (network. Operation of the 60 Series instrument as a master may be considered sometime in the future if there is deemed to be significant demand for this capability.

SPECIFYING MODBUS ADDRESS

P209 "**MbAdX**": This parameter is used to specify the Modbus address which is recognized by the indicator on the current serial port, as specified by the 'X'. If P205 is not set for "Modbs" then P209 is not accessible. Enter a value between 1 and 247, or press **[ENTER]** alone to increment the address by one until the desired address is displayed.

MODBUS "INPUT" SETPOINTS

When a Modbus command is used to turn on or off an input setpoint above #128, the setpoint's current state is not tested or affected. Therefore, if the setpoint invoked a macro when it turned on, it would still invoke the macro even if the setpoint should have already been on.

The input type setpoints will not work correctly for setpoints 129-144 due to the way PDIO input setpoints are handled. Recommend only using setpoint inputs above 144 if invoking macros via Modbus.

In other words, a macro may be invoked via a Modbus command via the following method: Choose a setpoint above 144 and set it up as an input type that invokes the desired macro upon activation and/or deactivation. When a Modbus command is received to activate or deactivate that setpoint, the macro will be invoked.

MODBUS PROTOCOL (RTU MODE)

RTU (Remote Terminal Unit) messaging framing is used on a Modbus network in place of ASCII message framing. With this method of framing, each 8-bit byte in the message contains two 4-bit hexadecimal characters. The main advantage of this mode over ASCII is that there is greater character density allowing better data throughput for the same baud rate.

Note: The following information pertains to setting up the 60 Series for Modbus networks with RTU framing. The following examples are 60 Series setups for slave operation on a Modbus network with RTU framing. To make use of this protocol with the 60 Series requires existing knowledge of the Modbus protocol. For any further information on Modbus Protocol and ASCII & RTU framing, refer to the following document.

Modicon®, Modbus Protocol Reference Guide, PI-MBUS Rev. G

Notes on Using MODBUS RTU on 60 Series instruments :

Works only on comm port 1

8 data bits are forced regardless of P201.1

Flow control is forced to none regardless of P204.1 (also forced now for ASCII mode)

Setup mode:

Set Modbus receive type

Set Modbus address

Set Modbus mode to RTU instead of ASCII (new parameter P210)

Sending device must transmit 1-byte binary address, 1-byte binary function code, n binary message bytes, low byte of crc-16 (initial value 0xffff) of all preceding bytes, high byte of crc-16 of all preceding bytes.

Master device cannot intersperse delays of more than 1.5 character times into a message to a slave device.

MODBUS ADDRESS TRANSLATION TABLE

P6001: "Modbus" is used to specify which of the 60 Series parameters are mapped to which Modbus address. Since Modbus handles the transferring of data by referencing the data's address, it is necessary to specify which parameters are assigned to which Modbus address.

The parameter assigned to the first Modbus address (address 40001) is specified at P6001 of the setup mode. The subsequent parameters specified at P6002, P6003, etc.… are then assigned the next available Modbus address, which is dependent upon the number of registers required for the preceding parameters.

To specify a parameter, three items are required:

Parameter Number

Instance

Translation Type

SPECIFYING A PARAMETER

The parameter selection process is very similar to other setup modes requiring a parameter selection.

Press **[ENTER]** to access the standard parameter selection menu. Then use the up and down arrow keys to scroll through the menu to find the desired parameter, then press **[ENTER]**. Again, use the up and down arrow keys to scroll through the menu to find the desired instance of the specified parameter, and press **[ENTER]** again. Then key in a format code (from the appropriate table below) and press **[ENTER]**, or press the up and down arrow keys to cycle through your choices. Then press **[ENTER]** to save your choice.

Alternatively, simply key in the parameter number, a decimal point and then the instance number followed by **[ENTER]** to specify the parameter. Then choose the format code as described above.

New parameter entries are inserted in front of the displayed entry. In effect, you cannot write over a parameter, but you can insert a new one and delete an old one. If you make an entry in the middle of a list the entry you were viewing and all the subsequent entries are pushed down to make room for the new entry.

The relevant formatting codes are dependent upon the parameter's data type.

String type parameters do not require format codes. The displayed information for format type is "StrXX" where XX indicates the length of the string.

DELETING A PARAMETER

To remove a parameter from the list, press **[CLR]** while viewing the parameter to be deleted. To delete all of the parameters, press **[CLR]** at the end of the table, while viewing " End ".

VIEWING THE MODBUS ADDRESS TRANSLATION TABLE

The dot matrix display can be toggled between a few different viewing modes. Pressing the [F4] key toggles between the following two modes:

"Modbs" Top Line

"Name " Bottom line (name or number and instance)

and

"40XXX" (The effective Modbus address of the parameter)

"Type " (One of the types shown in Table 8-15 and Table 8-16.)

Table 8-15: Integer (Signed or Unsigned) Translation Types

CODE	DESCRIPTION	NO. OF REGISTERS	DISPLAYED TYPE
0	16 bit integer	1	Int16
1	32 bit integer	2	Int32

Table 8-16: Float Translation Types

CODE	DESCRIPTION	NO. OF REGISTERS	DISPLAYED TYPE
0	Float to Fixed Point 16 bit	1	Int16
1	Float to Fixed Point 32 bit	2	Int32
2	IEEE floating point	2	FltIE

When the first viewing mode is selected, pressing **[F2]** will toggle between showing the name of the parameter or the number (first two digits) and instance (next three digits)

To step through the translation table, press **[SELECT]** to view subsequent parameters. Then press **[F2]** or **[F4]** to toggle between the different viewing modes. When you reach the end of the Modbus translation table, the display will read " End ".

If the Modbus Address Translation Table has not been programmed then P6001 will show "None".

OTHER SETUP PARAMETERS

The normal communication protocol parameters must be programmed so that the devices communicating have the same settings as each other. These parameters include:

Parameter Name

Parameter Number

Available Choices (per Modbus Specification)

Baud Rate (P200)

Data bits P201 (always 7 data bits)

Parity P202 (none, odd, or even)

Stop bits P203 (2 stop bits if no parity, 1 stop bit otherwise)

Flow Control P204 (None)

Transmit Buffer Size P207

> For ASCII mode Modbus transmissions, the size of the transmit and receive buffers is not critical. However for most efficient operation, the transmit buffer size should be set as large as the longest anticipated transmission.

It is possible to invoke a 60 Series macro via a Modbus command. Normally, setpoints which are set up to invoke macros will only invoke those macros when the setpoints change state due to their setup conditions being met or due to a timer expiring. The macros will not be invoked when the setpoints are activated on command. However, if one of the upper setpoints (setpoints 129 through 256) is set to be an 'Input' type, then if that setpoint is activated via Modbus command, the macro associated with that state change of the setpoint will be invoked.

Receive Buffer Size P208 (Same as transmit buffer)

RS-485 Multi-Drop P250

> Enabling P250 and installing the Network Option Board allows a number of Modbus devices to be networked together on the same pair of lines. This option is available only for COM1. When Modbus is enabled, access to P251 is not allowed since P251 has no effect.

Setpoint Assignments

> When assigning the 60 Series setpoints to specific devices, it is advantageous to group all of the setpoints that will be accessed via the Modbus interface together. This is not essential, but it will reduce the number of commands required to read and/or write several setpoints.

MODBUS PACKET FORMATS

Following are byte by byte descriptions of the data transferred for each of the supported Modbus commands.

MODBUS PACKET FORMAT DETAILS

01 Read Coil Status/02 Read Input Status

Since the indicator does not use separate numbering for input and output setpoints (coils) both of these commands will read the status of the specified setpoint number, regardless of type.

Setpoint 12 is requested as #11, which in hexadecimal is 0B. 42 -12 = 30, 30 + 1 = 31 (you must add 1 to include both the first and last setpoints) 31 = 1F in hex. Address 23 = 17 hex.

The master sends a packet containing the first setpoint number to read (start address, high and low byte) and the number of setpoints to read. However the setpoint numbers are offset by one, thus setpoint #1 is referenced as setpoint 0 in the Modbus packet. Following is an example of a request to read setpoints 12 through 42 from an indicator whose Modbus address (P209) is set to 23. Refer to Table 8-17

Table 8-17: Query Packet Sent from Master to Slave

FUNCTION CODES 1 & 2: READ COIL/INPUT STATUS			
Field Name	Hex Value	ASCII Character 1	ASCII Character 2
Header	3A	: (colon)	none
Slave Address	17	1	7
Function Code	01	0	1
Start Address (High Byte)	00	0	0
Start Address (Low Byte)	0B	0	B
Number of Setpoints (High Byte)	00	0	0
Number of Setpoints (Low Byte)	1F	1	F
LRC (Error Checking)	B6	B	6

The slave's resulting response packet contains data indicating the state of each of the requested setpoints. The status of each setpoint is represented by a single bit, with an activated setpoint being a '1', and a deactivated setpoint being a '0'. A sample response packet to the preceding query packet would appear as in Table 8-18.

Table 8-18: Response Packet Sent from Slave to Master

FUNCTION CODES 1 & 2: READ COIL/INPUT STATUS			
Field Name	Hex Value	ASCII Character 1	ASCII Character 2
Header	3A	: (colon)	none
Slave Address	17	1	7
Function Code	01	0	1
Byte Count	05	0	5
Data (Coils 19 - 12)	2A	2	A
Data (Coils 27 - 20)	10	1	0
Data (Coils 35 - 28)	C4	C	4
Data (Coils 43 - 36)	7D	7	D
Data (Coils 51 - 44)	1F	1	F
LRC (Error Checking)	B6	B	6

The individual setpoint states specified by the example response above are detailed below. A state of '1' indicates a bit value of 1 which corresponds to an activated setpoint. Refer to Table 8-18.

Table 8-19: Setpoint States Specified by the Example Above

Bit#	7	6	5	4	3	2	1	0	Hex Value
Setpt#	19	18	17	16	15	14	13	12	
State	0	0	1	0	1	0	1	0	2A
Setpt#	27	27	25	24	23	22	21	20	
State	0	0	0	1	0	0	0	0	10
Setpt#	35	34	33	32	31	30	29	28	
State	1	1	0	0	0	1	0	0	C4
Setpt#	43	42	41	40	39	38	37	36	
State	0	1	1	1	1	1	0	1	7D
Setpt#	51	50	49	48	47	46	45	44	
State									1F

03 Read Holding Registers

This command contains addresses offset from address 40001. A programmable translation table in the 60 Series instruments is used to translate register addresses to parameter numbers and instances. Each row of the translation table contains three elements:

Parameter Number

Instance

Translation Type

Hexadecimal Values Definition—Hexadecimal is a base 16 numbering system. That means that each digit has sixteen possible values which are known as zero through fifteen in the decimal numbering system. The characters A through F are used to represent the value 10 through 15 respectively.

05 Force Single Coil

Allows a single setpoint output state to be specified.

06 Preset Single Register

Allows a single parameter within the indicator to be given new data.

15 Force Multiple Coils

Allows a several sequential setpoint output states to be specified.

16 Preset Multiple Registers

Allows multiple parameters within the indicator to be given new data.

CHECKSUM PROTOCOL

One of the most effective and popular error-detection methods is the cyclic redundancy check (CRC). The CRC method is used in virtually every field where transmitting serial data is involved. The CRC is basically an error detection mechanism. The CRC follows three basic rules in order to insure the data that has been transmitted has been received properly.

1. Along with the message, provide the device receiving the data with some means of knowing it received it correctly.
2. The receiving device should send a return message, acknowledging receipt or asking for retry.
3. Continue to send the message until it gets to its destination.

In Europe, if a printer is not located adjacent to the controller then the transmission must include a checksum and a mechanism to re-attempt a transmission in case of errors in order to be PTB approved. A CRC would suffice in this application.

Several different styles of checksums can be calculated by the controller to help insure the integrity of the transmitted data. One of these checksum calculation methods matches that used by Epson printers in a protocol commonly used in Europe. Together with capabilities of the input interpreter (P205), the indicator can be used with these Epson printers, insuring correct data transfer by re-sending the transmission if the required acknowledge is not received.

A data checksum calculation consists of three commands:

1. Initialize and begin calculating a specific type of checksum starting with the next transmitted character.
2. Stop calculating the checksum (optional). Required only when the checksum is not to be transmitted until after some additional characters are transmitted.
3. Transmit the checksum. Since most supported checksums are two-byte, there are two commands, one for most significant byte (msb) first and one for least significant byte (lsb) first. Both bytes will be transmitted in succession. For single byte checksums, either command can be issued.

Several codes have been defined which allow these commands to be embedded at the proper locations in a custom transmit or to be done at a particular time within a macro. Similar to the way a carriage return/line feed combination can be programmed into a custom transmit setup by entering .256, the codes for the checksums can be entered as shown in Table 8-20.

To allow one these checksum commands to be issued directly from a macro, use the appropriate code listed above along with the “send control code” command, “%&.” For example, use 306%& to begin a LRCC-8 checksum.

Table 8-20: Checksum Format Codes

CHECKSUM CODE	FUNCTION	DESCRIPTION
.300	stop	Stop calculating the checksum but do not transmit yet.
.301	CCITT	International standard CRC
.302	SDLC/HDLC	CRC used by IBM
.303	CRC-16	Most commonly used CRC in the United States
.304	CRC-12	Used when bytes are 6 bits
.305	LRCC-16	16 bit CRC
.306	LRCC-8	8 bit CRC, used by Epson
.307	XMODEM	Registers are shifted left, opposite CCITT method which shifts right. Used with transmissions up to 9,600 baud.
.308	Sum 16	2 byte additive checksum
.309	Sum-8	1 byte additive checksum
.310	Send Checksum	Transmit checksum sending LSB first
.311	Send Checksum	Transmit checksum sending MSB first
.312	Alternate CRC-16	CRC used by GE FANUC
.313	Inverse LRCC-8	Binary inverse (binary negated) of the LRCC-8 checksum code 306.

The following polynomial equations are used in calculating the checksums for the specified checksum format codes in.

CCITT

$G(x) = x^{16}\ x^{12} + x^{5} + 1$

Feedback = 8408h

Initial checksum value = 0000

SDLC

$G(x) = x^{16}\ x^{12} + x^{5} + 1$

Feedback = 8408h

Initial checksum value = FFFF

CRC-16

$G(x) = x^{16}\ x^{15} + x^{2} + 1$

Feedback = A001h

Initial checksum value = 0000

CRC-12

$G(x) = x^{12} + x^{11} + x^3 + x^2 + x + 1$

Feedback = F01h

Initial checksum value = 0000

LRCC-16

$G(x) = x^{16} +1$

Feedback = 8000h

Initial checksum value = 0000

LRCC-8

$G(x) = x^8 +1$

Feedback = 80h

Initial checksum value = 0000

This feature only allows the transmission of checksums, not the receipt of checksum data.

XMODEM

$G(x) = x^{16} x^{12} + x^5 +1$

Feedback = 1021h

Initial checksum value = 0000

Alternative CRC-16

$G(x) = x^{16} x^{15} + x^2 +1$

Feedback = A001h

Initial checksum value = FFFF

Sum-16

Additive checksum

Initial checksum value = 0000

Sum-8

Additive checksum

Initial checksum value = 0000

Refer to the *Printer Interface Example* section below for an example of how to interface an Epson printer with the 60 Series instrument using checksums.

PRINTER INTERFACE EXAMPLE

While there are numerous ways of accomplishing various tasks with the 60 Series instrument. One possible method follows below. This implementation uses Macro 1, 2, 3 ,4, Custom Transmit 1, and Var#80.1.

Macro 2 is named Print Block so that it can be started from the **[F2]** key. Macro 1 which is invoked by setpoint 1 upon controller power-up or after exiting setup mode clears Var#80.1. Var#1 is used to keep track of the state of the interface.

1. When macro 2 is executed, it checks Var#80.1 to determine if a print is in progress. If no print is in progress, <ENQ> is sent out COMM port #1 to indicate the beginning of a transmission and Var#1 is incremented to 1 to prevent other transmissions. If a print is in progress, the message “Can’t Send” is displayed if macro 2 is invoked.
2. The input interpreter #1 is set to execute macro 3 when a <ACK>is received. Macro 3 checks if Var#80.1. If it is, custom transmit 1 is sent and Var#1 is set to 2.
3. After the transmission is sent, the printer will respond with either an <ACK> or a <NAK>. If <ACK> is received at this point (Var#80.1 not =1) then macro 3 changes Var#1 back to 0. This means another transmission could be initiated.

If <AK> is received, the input interpreter #2 will cause macro 4 to run. This macro will send an <ACK> to the printer and set Var#80.1 to 1. Then step 2 is repeated.

Custom transmit 1 describes the format of a custom transmit using CRCs. The file, (LRCC8.SET) contains this implementation.

Printer Interface Example

```
100%s23640%i%e Access Setup Modes,
               Allowing Changes

65010%s1%e%e   P65010. dbase Reset

108%s1%e       P101.01 Scl 1 Scale
109%s2%e       P109.02 Scl 1 Enbld

205%s2%e       P205.02 Recv1 Intrp

218%s.000%e    P218.00 RxTrm <NUL>
219%s1%e%e     P219.00 RxIn#1
221%s0%e       P221.00 RxTyp Char
223%s%c%e      P223. RxFmt
.006%e               <ACK>^F=06
224%s3%e       P224.03 RxMac 3

219%s2%e%e     P219.00 RxIn#2
221%s0%e       P221.00 RxTyp Char
223%s%c%e      P223. RxFmt
.021%e               <NAK>^U=21
224%s4%e       P224.03 RxMac 4

9990%s1%e%e    macro#1
10001%s
80.1P=0%%o%e   copy register

9990%s2%e%e    macro#2
9991%s
PrintBlock%e   10001%s
80.1P==0%o%e   0001 compare
1%%"%e         0005 select port
Ap1%%)%e       0007 flush keys
80.1P=1%%o%e   0009 copy register
5%%&%e         0014 send code
%%N%e          0016 if not
Can'tSend!%
%P%e           0017 pause
%%E%e          0028 end if

9990%s3%e%e    macro#3
10001%s
80.1P==1%%o%e  0001 compare
80.1P=2%%o%e   0006 copy register
1%%Q%e         0011 custom
               transmit
%%N%e          0013 if not
80.1P=0%%o%e   0014 copy register
%%E%e          0019 end if

9990%s4%e%e    macro#4
10001%s
80.1P==2%%o%e  0001 compare
1%%"%e         0006 select port
80.1P=1%%o%e   0008 copy register
6%%&%e         0013 send code
```

TRANSMIT THE CURRENT SCALE NUMBER (ASCII)

Table 8-21 shows the format code for transmitting the current scale number as a single ASCII character.

Table 8-21: Transmit Current Scale Number Format Code

FORMAT CODE	FUNCTION	DESCRIPTION
.350	Transmit current scale #	This format will Transmit the current scale number as a single ASCII character

Examples

Scale #	Decimal Value/ASCII character
1	49 = 1
2	50 = 2

3 51 = 3

4 52 = 4

MISCELLANEOUS PROTOCOL (BINARY TO TEXT CONVERSION)

The following example shows how to set up a Custom Transmit for binary to text communication. Refer to Table 8-22.

Format code 401 begins the conversion of a one byte binary character into two HEX text characters. Format code 400 ends the conversion back to a single byte binary character.

Table 8-22: Binary to Text Format Codes

FORMAT CODE	FUNCTION	DESCRIPTION
.400	Stop	End binary to text conversion.
.401	Begin	Start binary to text conversion

TRANSMIT NETWORK ADDRESS (ASCII)

Table 8-23 shows the format code for transmitting the address of the indicator.

Table 8-23: Transmit Network Address in ASCII

FORMAT CODE	FUNCTION	DESCRIPTION
.402	Transmit network address	The network address selected at P251 is transmitted as a single ASCII character.

The indicator's address is selected at parameter P251. Address selections are (4 through 250). This format code will transmit the address as a single ASCII character.

Examples

ADDRESS Decimal Value/ASCII character

4 4 = <EOT>

13 13 = <CR>

49 49 = 1

TRANSMIT SETPOINT STATUS AS A BINARY CODE

These format codes allow for the ability to transmit the current status of setpoints as a binary code (one bit per setpoint). The least significant bit is the lower setpoint number. To send these codes key in ".5xx (per Table 8-5) into the custom transmit setup or execute the macro "5xx%&".

Example #1

502%& will send setpoint 9-16 status if 9, 11, 12 & 15 are on

bit position number:	7	6	5	4	3	2	1	0
setpoint #:	16	15	14	13	12	11	10	9
bit value:	0	1	0	0	1	1	0	1
hex equivalent:	0x4D							

Thus the character 0x4D would be transmitted for this example.

Example #2

To send Example #1 as displayable characters.

Setup as a custom transmit:

.401%e Begin binary to text conversion
.502%e Send setpoint 9-16 status
.400%e End binary to text conversion

Setup to send using a macro:

401,502,400%&

The resulting transmission would be two bytes:

"4", "D" which in hexadecimal equals: 0x34, 0x44

PRINTING OPERATIONS

Depending upon how the transmission parameters have been set up, the following information can be printed when you press **[PRINT]**:

Stored data and other information that was entered into a Custom Transmit Setup

Refer to *Creating a Custom Transmit Table* on page 8-21.

A default Custom Transmit Setup is programmed into your controller at the factory. In the following example, the font size of the text is a function of the printer capabilities.

STR#1

9876.54 lbs Gross Weight

9864.20 lbs Net Weight

12.34 lbs Tare Weight

There are 250 Custom Transmits that can be set up in the controller. To print a Custom Transmit setup, press **[n] [PRINT]** where “n” represents 1 through 250, and:

1 = Custom Transmit 1

2 = Custom Transmit 2

3 = Custom Transmit 3

250 = Custom Transmit 250

Most printing transmissions are begun by pressing the **[PRINT]** key. However, the *Continuous Print* can be programmed with a setpoint and a macro printing the programmed data each time the display is updated. This feature is particularly useful when the controller is connected to a remote display or interfacing with a computer that is monitoring the process.

If the receiving device (printer, display or computer) goes offline, is powered down, or cannot receive the data being sent for any other reason, the message Tx On Hold will appear for a few seconds.

Press **[CLR]** to abort the transmission. If this situation occurs while the Continuous Print feature is being used, the continuous transmission is suspended, but it can be resumed by pressing the **[PRINT]** key.

INPUT INTERPRETER

The Input Interpreter, when enabled, operates on data received through the serial port on the 60 Series instrument. It enables the controller to be programmed to perform complex custom applications, as well as recognize specific commands that may be unique to other indicators. This option enables any GSE 60 Series instrument to emulate commands from weigh indicators manufactured by other companies.

The Input Interpreter consists of 250 input specifications (250 for each port). Each specification operates independently and can be Line type, Character type, or left unused. When the received data matches one of the specifications, a macro can be initiated.

A Character type input specification will match a single received character. Although the specification may be several characters long, only the first character is compared. When a match occurs, all preceding data is cleared, and if a macro number is programmed, it is invoked.

With Line type input specifications, the received data is held in a buffer until a terminating character is received. This terminating character is programmable, but the default is a decimal 10, which is an ASCII LINEFEED. All Line type input specifications use the same terminating character.

When the terminating character is received, the data in the buffer is compared against the Line type input specifications. If a match is found and a macro number is programmed, it is invoked. Whether or not a match is found, the buffer will be cleared of all data up to and including the terminating character.

Line specifications can contain text, control codes, and parameters. If there are no parameters, the received data is simply compared against the specification, and they must be identical to be considered a match.

If parameters exist in the line specification, characters from the received data will be stored into that parameter. Characters before the parameter in the line specification must match characters in the received data. A match occurs when data has been stored into all parameters in the line specification. Input specification format lines can be up to 255 characters long. However, the input buffer for the input interpreter accepts no more than 85 characters, which is the maximum number of characters that can be interpreted. When this buffer fills up, it is cleared.

Table 8-24 describes the Input Interpreter setup parameters.

Table 8-24: Input Interpreter Setup Parameters

PARAMETER	SELECTIONS	DESCRIPTION
199 PortX	0	None
	1	COM1 port selected
	2	COM2 port selected
	3	COM3 port selected
	4	COM4 port selected
205 Recv#	2	Interpreter Enable
218 RxTrm	0 through 255	Line Type Termination Character
219 RxIn#	1 through 250	Specify the interpretation number for the COMM port specified at P199.
220 RxNam	XXXXXX	Key in more descriptive name for the specified interpretation.
221 RxTyp	0	Character Type
	1	Line Type
222 Rx X	Key in Text, Parms, Format Line control codes for the line type interpretation number specified at P219.	
223 RxChr	0 through 255	Select the character interpretation for the interpretation number specified at P219.
224 RxMac	1 through 250	Macro Number
	0	No Macro

SETUP

The Input Interpreter is an advanced software feature. that must be set up properly in order to function properly. Setup data for the Input Interpreter is stored in electronically-erasable, programmable read-only memory (EEPROM) along with all the other scale setup information.

GENERAL SETUP AND COMM PORT SELECTION

The Input Interpreter can be enabled or disabled for at parameter P219 (selections may be different for each COMM port). All remaining input interpreter setup parameters are always retained regardless how this parameter is set up.

The instance or interpretation for a specified port is specified at Parameter P219 (selections are 1 through 250). **Parameter P199** specifies which COMM port the interpretation is associated with. When an interpretation is specified at P219, all received serial data is captured and used by the input interpreter. When P219 is disabled or no interpretations specified, all serial data is received and used by the controller command processor.

The interpretation input specification terminating character is programmed at Parameter 218. For values 0 through 99, the value is displayed on the numeric display, while the ASCII interpretation of it is shown on the lower line of the dot matrix display. For values greater than 99, only the value is displayed on the lower line of the character display.

For each of the 250/COMM port input specifications, two parameters must be set-up:

Type

Format line

TERMINATION CHARACTER

How you handle your terminating character is very important. Suppose you want the controller to execute a macro when it receives the word “START” through the serial port. You set up the input specification #1 to be line type, the format line to “START” and the macro number to 0. The terminating character is set to 13, which is a carriage return.

This setup will work fine if what is sent to the scale is “START” followed by a carriage return. But if a line feed is sent following the carriage return, this will only work the first time, because the line feed will remain in the buffer and be taken as the first character of the next transmission.

There are two ways around this. First, if you know the transmissions will always include a line feed, set the terminating character to 10 (line feed) and insert the carriage return at the end of the format line.

Alternately, you can set up another input specification to be character type, with a line feed as the format line, and no macro. This way, the linefeed will simply clear the buffer, which was already done by the carriage return, so in effect the line feed is ignored. This will allow all line type input specifications to handle transmissions with or without a line feed.

INPUT SPECIFICATION TYPE

The input specification type is programmed at parameter P221. Selections for the type are:

0 - Char (character)

1 - Line (line)

To scroll through the selections, press **[ENTER]**.

To make a selection, key in the selection and press **[ENTER]**.

When one of the selections is specified over another, certain parameters specific to that selection will be available in the following parameters.

INPUT SPECIFICATION FORMAT LINE

The format line is programmed at Parameter 222. The format line is displayed on the lower line of the character display, and the edit position is the last character on the right. The numeric display shows how the edit position is offset from the first character of the format line. A solid box character indicates the end of the format line, a lowercase P_A indicates a parameter, and a lowercase $^{C}_{C}$ indicates a control code. Here the following keys perform special functions:

[F3◄] Moves the edit position left.

[F5►] Moves the edit position right.

[F4▼] Expands character at edit position.

[F1▼] Enters Alphanumeric entry mode.

[ENTER] With no entry, enters the parameter select mode. Following an entry, inserts entry into format line at edit position.

[CLR] With edit position at the end of the format line will ask if you want to delete the entire line by prompting "**Clear All?**"; press **[ENTER]** for "yes," any other key for "no." Otherwise, it deletes the character at the edit position

In the *Alphanumeric* entry mode, **[F1▲]** and **[F4▼]** are used to scroll up and down through the ASCII character set. **[F3◄]** acts as a backspace, removing the character at the edit position and moving the edit position to the left by one character. The **[F5►]** key moves the edit position one character to the right and places an "A" there.

In the *parameter select* mode, **[F1▲]** and **[F4▼]** are used to scroll up and down through the available parameters. The parameter number is displayed in the last two digits of the numeric display. If you know the parameter number, you can enter it directly. When the name of the parameter you want is displayed, pressing **[ENTER]** will insert it into the format line at the edit position.

Control codes are inserted by entering ".XXX" when XXX is the decimal code for the control code. For example, keying in **[·] 013 [ENTER]** inserts a carriage return at the edit position.

Use caution when entering a "%" in the format line. In order to match a "%" in the received data, the format line must contain two percent signs, or "%%." To enter two percent signs in the setup mode requires that four percent signs be sent to the scale. A single percent sign has a special meaning for the input interpreter, which is described in the *For Programmers Only* section in this Chapter.

INPUT SPECIFICATION MACRO NUMBER

The macro number is programmed at Parameter P224. Here you select the number of the macro you want to initiate when a match is found. The choices are 1 through 250 (macro number), 0 = none. Key in the number of your selection and press **[ENTER]** to select a specific selection, or press **[ENTER]** by itself to scroll through the selections.

CLEAR INTERPRETER

Specify the interpretation number in question at P219. Press **[n] [ENTER]** to review a specific interpreter (n = 1 through 250). Press **[CLR]** to delete an interpreter specification. A prompt will read "**Clear One?**". Press **[ENTER]** to verify clear. To clear all key in **[999] [ENTER]** at this prompt.

OPERATION

A few aspects of the Input Interpreter deserve to be highlighted. Failure to fully understand these concepts can result in unacceptable operation of the Input Interpreter.

MULTIPLE PARAMETERS

Any Line type input specification can contain several parameters. Suppose you want the scale to receive and interpret the following transmission:

T1.234,PWT.05<CR>

The objective is to store "1.234" into the TARE register, ".05" as the PIECE-WEIGHT, and execute a macro when done. This can be accomplished by setting up an input specification as line type, with a format line of "T<pa=TARE>,PWT<pa=APW>," macro number set to 0, and the terminating character set to 13 (carriage return). Here <pa=TARE> means to insert the parameter TARE at that point in the format line.

If we receive an incomplete transmission, for example T1.234,PW<cr>, the data "1.234" will be stored in the TARE register. However, nothing will be stored in the PIECE-WEIGHT register. The macro will not be executed, because a match occurs only when data is stored into all the parameters in the format line.

In some applications, you may not want the value in the parameters to change unless a match has occurred. This can be accomplished by setting up the input specification to store the data into a unused variable (VAR), and have the macro copy the value into the desired parameter (TARE, QUANTITY, and so on) using the "%o" macro command.

TRAILING DATA

In an input specification with at least one parameter, any data that follows the data of the last parameter but precedes the terminating character is ignored. For example: a format line of "T<pa=TARE>" will match received data of "T1.234" and "T1.234 hello there."

To prevent this, "%5s" can be appended to the end of the format line, as in "T<pa=TARE>%5s." This tells the input interpreter to store up to five characters of trailing data in a dummy parameter. If the received data contains trailing data, data will be stored into two parameters, TARE and the dummy. Since the format line contains only one parameter, a match has not occurred, and the macro is not executed.

MULTIPLE MATCHES

The input specifications are checked in order from 1 to 250. If the received data could match more than one input specification, the first one checked that generates a match has priority. Once a match occurs, the comparison stops. The other input specifications are not checked.

DISABLING THE INPUT INTERPRETER

When enabled, the input interpreter software intercepts all received data, except for the following circumstances:

Input interpreter is disabled in the setup mode, parameter 100 and above. To download a new setup to the scale while the input interpreter is enabled, you must enter the setup mode manually by entering **100 [SELECT] 23640 [ID] [ENTER]**from the scale keypad.

The macro commands %G, %W and %Y suspend the input interpreter while waiting for operator input.

The input interpreter is suspended during database unloads.

Under these conditions, received data is accepted directly into the command/entry buffer of the scale.

USING A STRING AS A PARAMETER

When STRs are used as parameters, spaces in the received data are treated differently than other characters. Leading spaces in the received data are ignored. The first non-space character is the first character stored in the String. Data will continue to be stored into the String until the next space, the end of the received data, or the maximum size of the String is reached.

ADVANCED CONCEPTS

As you were reading the *Trailing Data* section you probably thought, "That **%5s** looks like a format string from a C language printf or scanf function!" You are right! A derivative of scanf is the heart of the line type input interpreter specification with parameters. This knowledge can be useful, as outlined below.

Three components are used by the input interpreter: the input string, the format string, and a parameter address list. The input string consists of the data up to but not including the terminating character.

The format string is derived from the format line which is input in the setup mode and stored in the EEPROM. At power-up or when exiting the setup mode (when the display reads "Doing Setup") the format line is scanned and all parameters are replaced by format codes appropriate to their type. The address list is also built so that scanf will know where to store the data for each parameter. One additional address is added to the end of the list, that of the dummy parameter for the trailing data previously discussed.

The format code for most parameters is "%f," for floating point data. The strings whose length is programmable through the setup mode, so their format code is "**%Xs**," where "X" is the programmed length, (for example, "**%20s**" for a string length of 20). The time/date parameters, are unsigned long type data, so their format code is "**%lu**."

It is possible to override these format codes by inserting a "%" immediately before the parameter in the format line. Then you can enter your own format code ahead of the "%." If you have multiple parameters you have to insert your format codes ahead of the first overridden parameter's format code.

This can be useful for parsing fixed-width data that contains no delimiters, because the maximum field width can be specified. For example, “%5f%6f%<pa=VAR1>%<pa=VAR2>“ will store the first 5 characters of data in floating point format into VAR1, and the next six characters in floating point format into VAR2.

An interesting but not very useful application is to override the format for an unsigned long parameter with “**%lx**.” This will interpret the incoming data as hex! “**%lo**” will interpret it in octal!

Back to more worthwhile things now, an asterisk immediately following the percent sign will cause a field to be scanned but not stored. So if you know that there is a floating number in the incoming data that you want to ignore, use “**%*f**” to skip over it, or to skip a single character use “**%*c**.”

This is not meant to be a tutorial on the uses of the scan function. Many resource books available go into much further depth. We have tried to suggest some possible ways of using its characteristics to good advantage.

INPUT INTERPRETER EXAMPLES

The following input interpreter example shows how the indicator can be programmed to receive a tare followed by a units identifier. A macro, which tests the transmitted units name and adjusts the data before storing it away as the new tare value, is invoked. Refer to the next several pages for input interpreter examples.

Example 1: Units identifier

File Name: TAREINP.SET

100%s23640%i%e	Access Setup Modes, Allowing Changes
681%s1%e	VAR instance 1
682%sTare units%e	P682.-- NAME1 Tare units
686%s12%e	P686.-- STR size=12
9990%s1%e	P9990 macro (instance 1)
10001%s%c%e	P10001 Macro table
80.1P==lb%%o%e	compare
80.1P=2P%%o%e	copy register
GOTlbtare!%%P%e	pause
%%N%e	if not
80.1P==kg%%o%e	compare
80.1P*=0.4535925%%o%e	multiply
80.1P=2P%%o%e	copy register

```
GOTkgtare!%%P%e      pause
%%N%e                if not
not Tared%%P%e       pause
%%E%e                end if

220%s1%e        P220.01  RxInp Enbld
221%s10%e            P221.10  RxTrm <LF>

230%s1%e        P230.01  (instance 1)
231%s2%e        P231.02  RxTyp Line
232%s%c%e            P232.04  RxFmt
%e80.1%e%e           Var#1
%e
%e80.1%e%e           VAR#1:
.013%e       <CR> ^M=13
233%s1%e        P33.01  RxMac   1
230%s2%e        P230.02  (instance 2)
232%s%c%e            P232.01  RxFmt
.026%e       <SUB>^Z=26
233%s0%e        P233.0  RxMac none
%z           Exit Setup Mode
```

Below are two lines of sample data transmissions which would be converted to lb and stored as tare weights when received by a 60 Series instrument with the setup specified above:

50kg

50 kg

The next two lines of sample data transmissions would be stored as tare weights without any units adjustments.

100lb

100 lb

The following four lines of sample data transmissions would *not* be stored as tare weights.

50xx

50 xx

50 xxsd

100 lbd

Example 2: Process AIAG Barcodes

Another potentially useful application for the GSE Input Interpreter is to process scanned in AIAG barcodes. For those not familiar with the AIAG standard, the following provides a brief primer on the subject.

AIAG	Automotive Industry Action Group
Identifier	The first character(s) of the bar-coded data which indicates the type of data in that barcode.
Type	AIAG uses Code 3 of 9.
Usage	This barcode is typically required to be used by suppliers on shipments to the automotive manufacturers.

The following are a few of the most commonly used identifiers:

P Part No.

Q Quantity

S Serial No.

V Vendor No.

The setup file below shows how the indicator may be setup to properly store away data scanned from AIAG barcodes.

File Name: AIAGINP.SET

```
100%s23640%i%e  Access Setup Modes, Allowing Changes

681%s1%e        VAR instance 1
682%sVndr:%e    P682.-- NAME1 Vndr units
686%s12%e       P686.--  STR size=12

681%s2%e        VAR instance 2
682%sP/N:%e     P682.-- NAME1 P/N
686%s10%e       P686.--  STR size=10

681%s3%e        VAR instance 3
682%sS/N:%e     P682.-- NAME1 S/N:
686%s10%e       P686.--  STR size=10

681%s4%e        VAR instance 4
682%sQ:%e       P682.-- NAME Q
```

```
686%s12%e           P686.--

9990%s1%e           P9990 macro 1
10000%s%c%e         P10000  Macro table
Vend#saved%%P%e     pause

9990%s2%e           P9990 macro 2
10000%s%c%e         P10000  Macro table
Qty  saved%%P%e     pause

9990%s3%e           P9990 macro 3
10000%s%c%e         P10000  Macro table
Part#saved%%P%e     pause

9990%s4%e           P9990 macro 4
10000%s%c%e         P10000  Macro table
S/N  saved%%P%e     pause

220%s1%e        P220.01  RxInp Enbld COMM 1
221%s10%e           P221.10  RxTrm <LF>
230%s1%e        P230.01  (instance 1)
231%s2%e        P231.02  RxTyp Line
232%s%c%e           P232.04  RxFmt
V%e
%e80.1%e%e          Vndr:
.013%e       <CR> ^M=13
233%s1%e        P912.00  RxMac   1
221%s10%e           P221.10  RxTrm <LF>
230%s2%e        P230.01  (instance 2)
231%s2%e        P231.02  RxTyp Line
232%s%c%e           P232.04  RxFmt
Q%e
%e80.4%e%e          INT (integer 4)
.013%e       <CR> ^M=13
233%s2%e        P233.02  RxMac   2

221%s10%e           P221.10  RxTrm <LF>
```

230%s3%e P230.01 (instance 3)

231%s2%e P231.02 RxTyp Line

232%s%c%e P232.03 RxFmt

P%e

%e80.2%e%e P/N:

.013%e <CR> ^M=13

233%s3%e P233.02 RxMac 3

221%s10%e P221.10 RxTrm <LF>

230%s4%e P230.01 (instance 4)

231%s2%e P231.02 RxTyp Line

232%s%c%e P232.03 RxFmt

S%e

%e80.3%e%e S/N:

.013%e <CR> ^M=13

233%s4%e P233.04 RxMac 4

%z Exit Setup Mode

Several transmissions were tested with a 660 that has the above setup. These are shown below along with their respective results.

TRANSMITTED OR BAR-CODED DATA	RESULT
Q123	Stores “123” into STR 4.
Q 234	Stores “234” into STR 4 (leading space ignored).
Q 345hello	Stores “345” into STR 4 (hello is stripped off since STR 4 is a numeric field).
Q321 987	Stores “321” in STR 4 (“ 987” is lost due to space).
P10-40-5503	Stores “10-40-5503” into STR 2.
P 09-30-0238	Stores “09-30-0238” in STR 2 (does not store leading space!).
P200550-00000	Stores “200550-000” in STR 2 (loses last two “0”s since STR size is only 10!
MPNOT STORED	Nothing stored! “M” not defined, “P” not first.
MP NOT STORED	Nothing stored! “M” not defined, “P” not first.
S112233	Stores “112233” into STR 3.
S112-233	Stores “112-233” into STR 3.

S112-233x	Stores “112-233x” into STR 3.
S112,233x	Stores “112,233x” into STR 3.
VCRYSTAL SEMI	Stores “CRYSTAL” into STR 1 (the “ SEMI” is lost due to the space).
VCRYSTALSEMI	Stores “CRYSTALSEMI” into STR 1.
V987654	Stores “987654” into STR 1.
V 987 654	Stores “987” into STR 1 (the leading space is stripped and the space in the middle caused the “654” to not be stored).
V“GSE INC.”	Stores “"GSE" into STR 1 (strips off trailing "INC."” due to space).

Example 3: Input Interpreter with Spaces in Input

This example shows the method that will allow the input interpreter to include spaces in the scanned in data. The following excerpt is from a 660 setup file:

220%s1%e P220.01 RxInp Enbld

221%s10%e P221.10 RxTrm <LF>

231%s2%e P231.02 RxTyp Line

232%s%c%e P232.08 RxFmt

T%%[^%e

.013%e <CR> ^M=13

]%%%e

%e80.1%e%e VAR #1:

233%s1%e P233.1 RxMac 1

The input specification shown tells the controller to take every character except the <LF> character. The “T” at the beginning of the specification should be set to the appropriate identifier for your specific application and the number 49 following the percent character should be set to match the size of the STR specified for that input, for example, the same value as P686 if the parameter is VAR 1.

RS-485 NETWORKING (OPTION)

The 60 Series instruments support address recognition. This allows a further degree of multi-drop communications implementation. This feature is supported by *software*. It is recommended that *additional hardware*, such as a 485 transceiver device, be added.

This section describes the setup, operation of the GSE RS-485 network option, GSE part number 24660B-401A0.

SETUP

P250 must be enabled to access P251. P250 by itself only causes the RTS of COM1 to become a driver enable for the network option. Changing the P251 address to a non-zero value enables the network address recognition receive feature. The valid values for P251 are from 4 to 254.

The controller attempts to send data as a complete packet. This is accomplished by not enabling the transmitter until the transmission is complete or until the transmit buffer becomes full. Transmission completion is defined by the end of a Custom Transmit or by using the "%q" command to transmit with the "%$" and "%&" macro commands. Therefore, it is desirable to set the size of the transmit buffer large enough to hold a complete transmission (refer to P207.1).

When P251 is enabled, the controller ignores all data until an <STX> character (for example, <^B>) is followed immediately by a character that matches P251 or the NULL character (for example, an address of 0, <^@>). When that happens, all subsequent data is processed exactly as if networking were disabled except that when the <ETX> character (for example, <^C>) is received, the receive routine of the controller resets to look for the <STX> character again.

The *data packet* format recognized by the controller is defined as follows:

<STX> <ADDRESS> <DATA> <DATA> <DATA> <DATA> <DATA> ... <ETX>

The address is a single byte. There are 250 possible addresses (4 through 254). The address should not be an <STX> or an <ETX>.

The indicator's address character (defined by P251) can be transmitted using code 402. For example, enter "**.402 [ENTER]**" in a Custom Transmit setup or execute "**402 %&**" in a macro.

The <DATA> can be any information recognized by the controller, including direct commands -- such as a %p (Print). This would direct the addressed unit to send its Custom Transmit over the network.

The networking feature does not affect transmitted data. Therefore, to send data to another indicator or another device operating on the same protocol, you would need to program your Custom Transmit with an <STX> followed by the address of the device you want to transmit to, followed by the data you want to send, followed by an ETX. Of course, the same transmission can be accomplished with the macro transmit commands.

Filling strings, setting target variables, updating databases, and so on, are all possible scenarios.

Example: Macro Data Packet Setup

1%”Select COMM 1 (485 port)

2,402%& <STX> <ADDRESS>

Hello 650%$ <DATA>

3%& <ETX>

%q Send data packet (send buffer)

OPERATION

When a character is received, it is compared to the start of block character, <STX>. If it is the start character, then the very next character is compared to the address as defined by setup parameter P251. If it matches or if the transmitted address is 0, the controller processes all of the subsequent data until the end of block character, <ETX>, is received.

If the received address character is not 0 and it does not match P251, then all of the subsequent data is ignored until the next start of packet character is received.

NETWORK PROTOCOL

MODBUSÔ protocol supports RS-485 electrical standard for the 60 Series instruments. Refer to *MODBUSÔ Communications* on page 1-25 for more information.

Each COMM port on the indicator can be set up with a unique protocol. The COM1 port is set aside for use with the network board once the option is installed. The protocol settings for the *network board* are the same as the settings for *COM1*. These settings are found starting at parameter P200.

All devices connected to the network must have matching protocol settings.

Chapter 9 MACROS

Chapter 9 covers macros setup, execution and a complete list of macro commands.

Throughout this chapter, various references are made to the maximum allowable number of macros, databases, communication ports, etc. These references reflect the maximum value allowed for the 660 Series controllers. The maximum values for other models may be limited as noted in the specifications section of Appendix A.

OVERVIEW

MACRO SETUP

One of the most powerful operating features of the 60 Series instruments is the macro programming language. A macro is a programmable routine used to automate a process. A simple macro might consist of a single command to zero a scale upon receiving a remote key input. Complex process control applications are also possible using the vast macro command set to program multiple event-driven macro routines.

Macros are dynamically allocated at P9990. Each macro can contain up to 9998 instruction characters providing adequate memory has been installed. Macro instructions are entered as text characters into the *macro table* at P10001 → P19999 for the specified macro.

MACRO SETUP PARAMETERS

Macro setup parameters are multiple instance parameters. This means that the same parameter numbers apply to all macros.

ACCESSING MACRO SETUP PARAMETERS

An instance number must be given to identify a macro parameter before it can be accessed. The instance number can be specified at P9990 before selecting other macro parameters. For example,

Pressing [ID] while viewing P9991 → P9994 will display the macro number for one second.

9990 [SELECT] 6 [ENTER]

selects the macro instance parameter and identifies macro 6 as the current macro for the following macro setup parameters P9991 → P19999. If macro 6 did not exist, Make? NewMC would be displayed prompting you to create a new macro (see *Creating a New Macro* on page 9-4).

A more direct method of accessing macro setup parameters is to key in both the parameter number and instance at the same time. For example,

9991.6 [SELECT]

will provide direct access P9991 (macro name) for macro 6. Refer to the *Navigating Setup Parameters* section on page 3-6 for complete details on how to access multiple instance parameters.

MACRO INSTANCE SELECTION

P999)6	Mac.# 6

The *macro instance selection* parameter is used to assign new macros and to access existing ones. Key in the macro number to create or access and press **[ENTER]**. See page 9-4 for information on creating new macros.

P9991 – MACRO NAME

P999!	MName None!

The *macro name* parameter is used to assign or identify the name of the currently selected macro. Assigning a macro name is not required, however the name can prove useful for documentation purposes.

A macro name can be up to 79 characters long. When accessing P9991, only the first five characters of the name are displayed. Use the right and left cursor arrows on the front panel keypad to scroll forward and backward through the entire name. To enter a macro name at P9991, simply key in the name and press **[ENTER]**. Alpha characters may be entered through the front panel as described in the *Key-In Value Parameters* beginning on page 3-8. The previous name will be replaced with the entered name. A macro name can be deleted by pressing **[CLR]**.

P9992 – MACRO PRIORITY

The *macro priority* parameter determines whether a macro can only be invoked by standard methods, or executed immediately when invoked by a setpoint or input interpreter.

Invoke Standard

```
P999@0  Invok
          Std
```

Set P9992 to “Std” to allow a macro to be invoked in a sequential manner. If another macro is running, the invoked macro will be acknowledged and placed on the macro stack to be executed in the order it was invoked.

Invoke Immediate

```
P999@1  Invok
        Immed
```

Set P9992 to “Immed” to allow a macro to be executed immediately when invoked by a setpoint or input interpreter. If another macro is running, its execution will be interrupted so the invoked macro can execute immediately. The interrupted macro can resume operation after the interrupt macro completes execution. Refer to the *Interrupt Macros* section beginning on page 9-161 for complete details on interrupt macros.

P9993 – MACRO MENU

```
P999#0  Menu
        Disbl
```

The *macro menu* parameter is used to enable the macro to be invoked via the **[ID]** key. P806 must be configured for “Menu”. Refer to the *Front Panel Keypress* section on page 9-8 for complete details on macro menu operation.

P9994 – LIMITED ACCESS

```
P999$0  LmtAc
          no
```

The *limited access* parameter is used to protect against unauthorized access to a macro. If set for “yes”, it will not be possible to view, edit or download the macro when the setup mode is accessed using the limited access code assigned at P402. Refer to *Accessing The Parameter Setup Mode* beginning on page 3-3 for complete details on the limited access code.

P10001 → P19999 – MACRO TABLE

```
P1000!  Mc   6
        ††††‡
```

The *macro table* is a block of 9999 parameter locations used to store macro instructions. Each parameter of the macro table represents one macro character. For example, the first character entered in a macro is stored at P10001. The next character is stored at P10002, and so on. If 52 characters were entered in the macro table, P10052 would contain the last character entered. The last parameter of a macro table is the “end-of-table” character which is represented by a solid block character ‡. Refer to *Creating a New Macro* on page 9-4 and *Editing Macros* on page 9-5 for complete details on creating and editing macro tables.

USING THE MACRO % CHARACTER

When using a serial device to enter macro commands, you must type the % character twice consecutively (%%) to enter it in a macro table or any other setup parameter.

The percent character (**%**) is common to all macro commands, always preceding the macro command character. Whenever the **%** character is encountered during macro execution, the next character is examined to determine what function is to be performed. For example,

%S

is the macro command that sounds the beeper. When the **%** character is encountered, the next character **S** is analyzed and the beeper function is performed. It is important to realize that most macro commands can be executed from any mode, including the setup mode! This is often a source of confusion with beginning programmers, especially when using a serial device such as a computer keyboard to enter macro commands into a macro table. To better understand the concept of the **%** character, consider the following:

Executing a Macro Command

% SHORTCUT
When using the front panel keypad to enter macro characters, press [.] then [▼] nine times to quickly access the % character.

Suppose you have just created a new macro number and have selected P10001 to begin entering macro commands using a computer keyboard connected through one of the comm ports. You are about to execute the **%S** macro command as previously described. As you type the **%** character to begin the entry, notice that nothing appears on the display! Remember, most macro commands can be executed in the setup mode. Having received a **%** character, the indicator is now waiting for the next character to determine what macro function, if any, to perform. Thus when you type **S** to complete the entry, the beeper function is performed immediately. Since the macro command was processed, nothing was entered into the macro table.

Entering a Macro Command

Using a serial device to enter a **%** character into a macro table or any other setup parameter is accomplished by transmitting two consecutive **%** characters (**%%**). Considering the same **%S** example above, press **%** to begin the entry. As before, nothing appears on the display. Type **%** again. This time the **%** character appears in the entry buffer. Press **S**, then **[ENTER]** to save the command in the macro table.

CREATING A NEW MACRO

Each new macro must be assigned at P9990 and configured as described in the following procedure. New macros can be assigned in any order beginning with any available macro number 1 → 250.

To create a new macro:

1. From the setup mode, key in **9990 [SELECT]** to access the macro instance selection parameter.
2. Key in the macro number to create and press **[ENTER]**.
3. Press **[ENTER]** again at the `Make? NewMc` prompt to confirm the entry. P9990 now shows the new macro number.

Example:
Creating a New Macro

[9] [9] [9] [0] [SELECT]

P999)0 Mac.# None!

[6] [ENTER YES]

P999)0 Make? NewMc

[ENTER YES]

P999)6 Mac.# 6

[SELECT]

P999! Mname None!

Start Batch [ENTER YES]

P999! Mname Start

[SELECT]

P999@0 Invok Std

[SELECT]

P999#0 Menu Disbl

[SELECT]

P999$0 LmtAc no

[SELECT]

P1000! Mc 6 ††††‡

1%A [ENTER YES]

P1000$ Mc 6 †1%A‡

[F5] **= scroll forward** ➔

[F3] **= scroll backward** ←

4. Press **[SELECT]** to advance to the *macro name* parameter at P9991. If desired, key in the macro name and press **[ENTER]**. Alpha characters may be entered through the front panel as described in the *Key-In Value Parameters* section.
5. Press **[SELECT]** to advance to the *macro priority* parameter at P9992. Choose either Std or Immed by pressing **[ENTER]** to toggle between the two choices.
6. Press **[SELECT]** to advance to the *macro menu* parameter at P9993. Choose Enbld or Disbl by pressing **[ENTER]** to toggle between the two choices.
7. Press **[SELECT]** to advance to the *limited access* parameter at P9994. Choose yes or no by pressing **[ENTER]** to toggle between the two choices.
8. Press **[SELECT]** to advance to the *macro table* at P10001. Begin entering the macro characters. Alpha characters may be entered through the front panel as described in the *Key-In Value Parameters* section. As you key in characters, they are placed in the entry buffer until you press **[ENTER]** to save them in the macro table. Up to 79 characters can be held in the entry buffer before you must press **[ENTER]**, however it is good practice to press **[ENTER]** after each macro command to avoid time consuming mistakes. Note that the parameter number automatically advances once per entered character.
9. Save the macro configuration by exiting the setup mode as described in the *Exiting The Parameter Setup Mode* section. Macros are saved to EEPROM or instead to the database setup RAM if allocated at P60040.

⚠ If you exit the setup mode without entering any characters in a macro table, the configuration for that macro will be lost when changes are saved. The macro will no longer exist!

EDITING MACROS

Characters may be added to or deleted from any point in a macro table.

NAVIGATING A MACRO TABLE

A macro table begins at P10001 and ends at the end-of-table character. Each character in the macro table, including the end-of-table character, represents one unique parameter location within the table. Thus, it is possible to directly access any point in a macro table by selecting the appropriate parameter. For example,

10052.2 [SELECT]

will access the 52nd character in macro table #2. For larger macros it is helpful to obtain a copy of the macro download to identify the exact location of a macro command within a table (see *Downloading Individual Macros* on page 9-166).

The right and left arrow keys on the front panel keypad can also be used to scroll the characters of the macro table forward and backward across the 5-character macro window. When scrolling past the beginning or end or the macro table, the display “wraps” to the opposite end of the table.

INSERTING CHARACTERS IN A MACRO TABLE

Example – *Inserting Characters in a Macro Table* demonstrates how to insert a character into an existing macro table. Before inserting characters in a macro table, you must first access the macro table parameter that represents the desired insertion point. Insertion will always occur *between* the last two characters of the 2X5 matrix display. Access the insertion point by keying in the macro table parameter number (if known) and pressing **[SELECT]**. You can also use the right and left cursor arrows on the front panel keypad to scroll forward and backward through the macro table to the desired location.

Example:

Inserting Characters in a Macro Table

[1] [9] [9] [9] [9] [·] [6] [SELECT]

P1000$ Mc 6 †1%A‡

[F3] [F3]

P1000@ Mc 6 †††1%

[0] [ENTER YES]

P1000# Mc 6 ††10%

[F5] [F5]

P1000% Mc 6 10%A‡

P1000@ Mc 6 †††1%

Insertion Point

Insert characters by keying them in and pressing **[ENTER]**. Alpha characters may be entered through the front panel as described in the *Key-In Value Parameters* section.

Selecting P19999 as shown in the example – *Inserting Characters in a Macro Table* will always access the end-of-table parameter directly, regardless of the macro size.

DELETING CHARACTERS IN A MACRO TABLE

Before deleting characters in a macro table, you must access the macro table character that represents the desired deletion point. The last character of the 2X5 matrix display will be cleared each time **[CLR]** is pressed.

P1000@ Mc 6 †††1%

Deletion Point

DELETING MACROS

Macros can be deleted collectively or individually. Deleting macros collectively is usually performed just prior to uploading macros of a new setup file. This ensures that no macros from the previous setup will remain after the new setup is loaded. Deleting macros individually is done for several reasons. Delete an unused macro to conserve memory. Also, before uploading a single macro that already exists, the existing macro table must first be cleared. Failure to clear the macro will result in the new macro being appended to the existing one rather than replacing it.

To delete all macros:

Delete all macros before uploading a new setup file.

1. From the setup mode, key in **9990 [SELECT]** to access the macro instance selection parameter. The display shows the first assigned macro number.
2. Press **[CLR]**. The display prompts `Clear One?`.
3. Key in **999 [ENTER]**. The display shows `Mac.# None!` indicating that all macros have been deleted.

   - or -

   Press **[CLR]** to cancel deletion.

When uploading macros in a text file, add the following line prior to the first macro to ensure all existing macros are cleared first.

```
9990%s%c999%e     Clear all existing setups
```

To delete individual macros:

Delete unused macros to conserve memory and prevent unexpected macro execution.

1. From the setup mode, key in **9990 [SELECT]** to access the macro instance selection parameter. The display shows the first assigned macro number.
2. Press **[CLR]**. The display prompts `Clear One?`.
3. Press **[ENTER]** to delete the macro.

   - or -

   Press **[CLR]** to cancel deletion.

To clear a macro table:

Clear an existing macro table before uploading a new one.

1. From the setup mode, key in **19999.X [SELECT]** *(where X = macro number)* to access the end-of-table parameter. The display shows **‡** in the last character position.
2. Press **[CLR]**. The display prompts `Clear All?`.
3. Press **[ENTER]** to clear the macro table.

   - or -

   Press **[CLR]** to cancel deletion.

When uploading a macro in a text file, add the following line prior to the first instruction of the macro to ensure the table is cleared. Change the macro number (6) as necessary.

```
19999.6%s%c%e     Clear macro table #6
```

INVOKING MACROS

The following sources can be used to invoke macro execution:

- Front Panel Keypress
- Remote Key (460 Series)
- Macro Menu
- Setpoint Activation or Deactivation
- Input Interpreter
- Serial Data
- Another Macro

FRONT PANEL KEYPRESS

Example:
Redefining a Key to Invoke a Macro

`801%s10%e`	Zero Mc 10
`MACRO #10 – ZERO SCALE`	
`EnterCode?%G`	get code entry
`==23640%o`	if entry = 23640
`%z`	zero
`%E`	end if

Any key on the front panel keypad can be used to invoke a macro. The function keys, along with **[START]**, **[STOP]** and **[SETUP]**, are predefined to invoke specific macros. Other keys can be redefined at P800 → P820 to invoke a specified macro rather than perform its original function. This allows you to completely customize the keypad for any application. The example – *Redefining a Key to Invoke a Macro* demonstrates how to redefine the **[ZERO]** key to invoke a macro which will require the operator to enter a password before the scale will perform the zero function.

Table 9-1: Front Panel Keypad Macro Assignments

460 Keys	465 Keys	560 Series Keys	660 Series Keys	ASCII Value	Predefined Macro#	Macro Assignment Parameter	Equivalent Macro Command
-	**F1**	**F1 / START**	**F1**	128	1	-	1%^
-	**TARGET**	**F2 / STOP**	**F2**	129	2	-	2%^
-	**-**	**-**	**F3**	130	3	-	3%^
-	**-**	**-**	**F4**	131	4	-	4%^
-	**-**	**-**	**F5**	132	5	-	5%^
Remote Key 1	**Remote Key 1**	**Remote Key 1**	**START**	133	6	-	6%^
Remote Key 2	**Remote Key 2**	**Remote Key 2**	**STOP**	134	7	-	7%^
-	**-**	**-**	**SETUP**	135	8	-	8%^
SELECT	**SELECT**	**SELECT**	**SELECT**	243	-	800	%s
ZERO	**ZERO**	**ZERO**	**ZERO**	250	-	801	%z
TARE	**TARE**	**TARE**	**TARE**	244	-	802	%t
UNITS	**UNITS**	**UNITS**	**UNITS**	245	-	803	%u
SCALE SEL.	**SCALE SEL.**	**SCALE SEL.**	**SCALE SEL.**	224	-	804	%`
PRINT	**PRINT**	**PRINT**	**PRINT**	240	-	805	%p
-	**ID**	**F3 / ID**	**ID**	233	-	806	%i
-	**ENTER**	**ENTER**	**ENTER**	229	-	807	%e
CLEAR	**CLEAR**	**CLEAR**	**CLEAR**	227	-	808	%c
-	**.**	**.**	**.**	46	-	809	-
-	**0**	**0**	**0**	48	-	810	-
-	**1**	**1**	**1**	49	-	811	-
-	**2**	**2**	**2**	50	-	812	-
-	**3**	**3**	**3**	51	-	813	-
-	**4**	**4**	**4**	52	-	814	-
-	**5**	**5**	**5**	53	-	815	-
-	**6**	**6**	**6**	54	-	816	-
-	**7**	**7**	**7**	55	-	817	-
-	**8**	**8**	**8**	56	-	818	-
-	**9**	**9**	**9**	57	-	819	-
Any Key	**Any Key**	**Any Key**	**Any Key**	-	-	820	-

Example

Using the Macro Menu – Sample Setup

```
806%s1%e             IDUse Menu

9990%s11%e%e         Mac.#11
9991%sSTARTFILL%e    MName
9993%s1%e            Menu Enbld

19999.11%s%c%e       Macro #11
%%c%e                clear
1%%A%e               start fill

9990%s12%e%e         Mac.#12
9991%sTargtWght?%e   MName
9993%s1%e            Menu Enbld

19999.12%s%c%e       Macro #12
%%\%e                if no entry
1%%i%e               select VAR#1
TargtWght?%%G%e      get entry
%%E%e                end if
=80.1P%%o%e          save entry
1%%s%e               net mode

9990%s13%e%e         Mac.#13
9991%sFree Fall?%e   MName
9993%s1%e            Menu Enbld

19999.13%s%c%e       Macro #13
%%\%e                if no entry
2%%i%e               select VAR#2
Free Fall?%%G%e      get entry
%%E%e                end if
=80.2P%%o%e          save entry
1%%s%e               net mode
```

)0 lb ± ² ³ Net

[ID]

START FILL

Targt Wght?

[SELECT]

Free Fall?

[2] [·] [5] [ENTER YES]

)0 lb ± ² ³ Net

[ID]

START FILL

[ENTER YES]

Filling Routine Begins

The 460 Series indicators will process each keypress in a mode dependent manner. For example, the **[TARE / ENTER]** key for the 460 will interpret the key either as a **[TARE]** key or an **[ENTER]** key depending on the current mode of operation (i.e. performing an auto-tare versus entering a numeric prompt).

MACRO MENU

The macro menu provides a means of invoking any named macro from the front panel keypad. The following setup is required to use the macro menu:

- The **[ID]** key must be set for Menu at P806.
- Macros to appear in the menu must be set for Menu Enbld at P9993.
- A macro name must be assigned at P9991. The first 10 characters of the macro name will be displayed when the macro is selected in the menu.

To invoke a macro from the macro menu:

1. Press **[ID]** to display the first named macro.
2. Press **[ENTER]** to execute the displayed macro.

   - or -

   Press **[SELECT]** to display the next macro in the menu.

   - or -

   Press **[ID]** or **[ZERO]** to exit the menu without invoking a macro.

When in the macro menu mode, the keypad cannot be used to invoke macros by any means other than the macro menu. Most keys are disabled in this mode.

Combining Entry Prompts with the Macro Menu

Since a macro's name is used as the macro menu prompt, you may consider naming the macro as an entry prompt. Example - *Using the Macro Menu* shows a scenario where the macro menu invokes a macro that will start a filling operation. Two other macros were included in the menu to prompt an operator for a target or free fall weight. Prompting for an entry directly from the menu name is possible due to the fact that any characters in the entry buffer at the time **[ENTER]** is pressed will be passed on to the invoked macro.

In this scenario, macro 10 starts the fill operation by activating setpoint #1 to open the fill valve. Note the **%c** (clear) command at the beginning of this macro. This is done as a precaution to clear the entry buffer in the event an operator had pressed some keys before starting the fill operation. For example, if the operator had pressed **3 [ENTER]** at the START FILL menu prompt, then the '**3**' would have been passed from the entry buffer to the macro resulting in the macro executing the instruction **31%A** rather than **1%A** as intended.

In the case of the prompts for macro 11 and 12, operator entry is expected. The entry is passed from the entry buffer to the macro where it is stored in a variable. If the operator neglected to make an entry before pressing **[ENTER]**, these macros re-prompt the operator to do so.

SETPOINT ACTIVATION / DEACTIVATION

Any of the 256 setpoints, whether inputs or outputs, can be configured to invoke any macro upon activation and/or deactivation. This is perhaps the most versatile of all methods used to invoke macros, providing true process control capability. Chapter 9 provides complete details on setpoint configuration.

Example:

Invoking a Macro at Power-Up

```
5099%s2%e          Setpt     2
5100%s1%e          SPTyp Outpt
5101%sPower-Up%e   SPNam Power-Up
5110%s4%e          Activ Alwys
5111%s0.00%e       AcDly 0.00
5112%s250%e        AcMac   250
5113%s0%e          AcMtn Ign'd
5130%s5%e          Deact Never
5131%s0.00%e       DeDly 0.00
5132%s0%e          DeMac None!
```

USING A SETPOINT TO INVOKE A MACRO AT POWER-UP

A setpoint can be used to automatically invoke a macro at power-up. This “power-up” macro can then be used to perform a variety of operations such as displaying an operating menu on the 4X20 VFD, initializing variables, identifying a power failure, etc. As shown in Example - *Invoking a Macro at Power-Up*, a power-up setpoint is configured to be active always and deactive never. This causes the setpoint to become active immediately upon power-up thus invoking the power-up macro. Since the setpoint is never deactivated, it will never change states until a power interruption or upon entering the setup mode.

INPUT INTERPRETER

The input interpreter can be used to invoke a macro when a specified character or group of characters is received through a communication port. Refer to the *Input Interpreter* section for complete details on the input interpreter.

SERIAL DATA

A macro will be invoked if the *%^ Call \ Go To Macro* command is received and processed by one of the communication ports. For example, transmitting the macro command **100%^** to the indicator will result in macro 100 being invoked.

The *%H Redefine Comm Port Function* macro command can also be used to invoke a macro with each character received on the specified port.

OTHER MACROS

Macros can invoke other macros with the *%^ Call \ Go To Macro* command or with a variation of the *%J Jump to Tag* macro command.

MACRO EXECUTION

Macros instructions are executed sequentially within a macro, beginning with the first instruction and continuing until it reaches a stopping point, such as the end of the macro or a break command. Branching within a macro or from one macro to another is made possible with the use of Boolean logic commands, macro go-to or call commands and jump & tag commands.

Only one macro can execute at a time. Requests to invoke additional macros during the execution of another are pushed onto the macro stack to be executed in turn. It is possible to interrupt a macro to immediately execute another. It is also possible to abort or suspend macro execution via the front panel keypad.

The speed at which macro commands are executed depends on:

- Enabled Scales
- Enabled Setpoints
- Custom Transmits
- Enabled Analog Outputs
- A/D Interval

In general, the more non-macro functions the scale must perform, the less often the processor can execute macro commands. For example, the processor receives 60 A/D interrupts each second during which time it must update all active weight parameters. Macro commands are executed between these interrupts. If a second scale is added, approximately 60 more A/D interrupts occur each second. A continuous custom transmit will require yet more processor resources. The time it takes to process these additional functions results is less time to execute macro commands.

Even with numerous demands on the processor resources, macro execution speed will be adequate for most applications. Execution speed can be significantly increased by reducing the A/D interrupt interval (at the expense of weight conversions) with the *%r A/D Interval* macro command.

MACRO STACK

MACRO STACK	INVOKED MACRO#
200	
199	
198	
↑	↑
4	
3	
2	102
1	101
EXECUTE	6

Figure: Macro Stack

A macro will be executed immediately when invoked unless another macro is in process. If so, the invoked macro will be pushed to the top of the *macro stack*. Macros on the macro stack are executed in a first-in, first-out basis.

The figure – *Macro Stack* illustrates the macro stack concept. Here there are 2 macros on the macro stack, macros 101 and 102. These macros were invoked, but cannot run because macro 6 is presently executing. If another macro is invoked while macro 6 is still running, it will be pushed onto the 3rd position of the macro stack. When macro 6 ends, macro 101 will execute and macro 102 will move down to the 1st position on the stack. Any other macros on the stack will also move down one position.

Up to 200 macros can be stacked in this manner. If the macro stack limit is exceeded, a macro stack error results. The display will show `Code81 Macro Stack` and all macro execution stops. This situation can occur

Use the macro debug download to analyze macro stack activity.

if macros are continually invoked faster than they can execute. Avoid using program loops or long entry sequences when other macros are being invoked. If this is not practical, consider using interrupt macros (see the *Interrupt Macros* section beginning on page 9-161) to help prioritize macro execution. Other macro tools such as the *%B Break Macro* command can also help manage the macro stack. You can analyze macro stack activity by performing a macro debug download (see the *Macro Debug* section beginning on page 9-163).

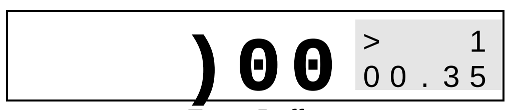
CALLING MACROS

If one macro calls another, the called macro is executed and the calling macro is pushed onto the *bottom* of the macro stack. Any other macros on the stack will move up one position. Thus the calling macro will resume execution immediately after the called macro ends.

ENTRY BUFFER

The *entry buffer* is perhaps the most versatile tool in macro programming. A thorough understanding of the entry buffer concept will not only help you write more efficient programs, it will also help you to avoid many unforeseen problems encountered by novice programmers.

In simple terms, the entry buffer can be defined as a temporary data register used to store operator input before it is used as part of another function. Characters in the entry buffer appear in the 2X5 display matrix as shown below. Here, the number **100.35** was keyed into the entry buffer.

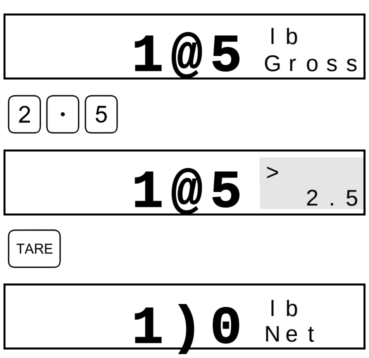


Entry Buffer

Notice how the 2X5 display matrix identifies characters in the entry buffer with the > symbol in the upper left corner.

MANUAL ENTRIES

Example:
Using the Entry Buffer to Perform a Manual Tare Entry



The entry buffer can hold up to 79 characters before it must either be cleared or used as part of another function. For example, if you perform a manual tare by keying in **2.5 [TARE]**, the number **2.5** appears in the entry buffer where it remains until the **[TARE]** key is pressed. The number becomes part of the tare function and is assigned as a manual tare entry (see example – *Using the Entry Buffer to Perform a Manual Tare Entry*). The following are examples of other functions that can use the entry buffer:

1 [UNITS]	select units #1 (kg)
5 [PRINT]	send custom transmit #5
3 [SCALE SELECT]	select scale #3
2.2 [SELECT]	select tare parameter for scale #2

10 [ID]	access VAR #10
50 [ENTER]	set parameter value = 50
[CLR]	clear the entry buffer

MACRO ENTRIES

Example:

Using the Entry Buffer to Begin a Macro

```
MACRO #6 - START
%A                    start auger
```

5

)0 > 5

START

Setpoint #5 is activated (5%A)

Macros can use characters in the entry buffer in the same manner as the keypad functions. Example – *Using the Entry Buffer to Begin a Macro* demonstrates how a number in the entry buffer can be used to activate any setpoint when macro 6 is invoked. Macro 6 consists of only the **%A** activate setpoint command which will not activate a setpoint unless a setpoint number is specified. Thus if the operator presses only the **[START]** key, nothing will happen. However, if the operator keys in the desired setpoint number, this number will be used to begin the macro and the macro now becomes **5%A**.

Any characters in the entry buffer at the time a macro is invoked will result in these characters being used by the macro. For example, if the **[STOP]** key is programmed to deactivate all setpoints with the ***%D** command and the operator keys in **10 [STOP]**, the macro will execute as **10*%D**. This is not a valid command and the macro will not execute properly. In these situations, be sure to clear the entry buffer with the **%c** or **%[** macro command at the beginning of the macro.

MACRO EXECUTION

The most versatile use of the entry buffer is during macro execution. Values and strings can be manipulated in much the same way a word processor allows you to copy and paste text. Refer to the macro ***Entry Buffer*** commands for complete details on entry buffer-related commands.

When a macro is running, it is not possible to put characters in the entry buffer except through the macro itself. Characters put in the entry buffer during macro execution are not shown on the 2X5 matrix display as with the manual entries. A single keypress or characters received on the comm ports will be buffered until all macros on the macro stack have ended. Buffered characters will then appear in the entry buffer in the order received.

Copying Values to the Entry Buffer

Batch# (VAR#1)	Name (VAR#2)	Target (VAR#3)	Valve# (VAR#4)
10	40# Salt	40	1
11	80# Salt	80	2
12	Sand	50	3
13	Cement	50	4
14	Soil	25	5

Figure: Sample Database

If a macro copies a value into the entry buffer, the copied information is temporarily inserted into the macro code and is used by the following macro command. For example, consider the following routine which copies the current time & date in text format to string variable #1:

11.0.18561P%o *copy formatted time/date to entry buffer*

=80.1P%o *copy entry buffer to VAR#1*

When this code is executed, the first line copies the time/date in text format to the entry buffer, thus inserting it before the assignment on the second line. Effectively, this code will execute as follows:

11.0.18561P%o

12:00:01 am 08/01/1999=80.1P%o

A more powerful use of this technique is illustrated in the following example. Here, the operator is prompted for a batch number that is subsequently recalled from the database shown in Figure 9-2. This updates variable #2, #3 and #4 with the corresponding name, target and valve number. The batch name is then displayed on the 4X20 VFD and the corresponding valve is activated.

```
EnterBatch%G   prompt for entry
=80.1P%o       copy entry to VAR#1
1,1%y          recall batch#
80.2P%o        copy batch name to entry buffer
P4,1c%C        copy entry buffer to 4X20 VFD
80.4P%o        copy valve# to entry buffer
%A             activate valve#
```

Assuming the operator entries a batch number of 12, this code will execute as:

```
EnterBatch%G
=80.1P%o
1,1%y
80.2P%o
SandP4,1c%C
80.4P%o
3%A
```

As you can see, the entry buffer allows this same routine to display any batch name and activate any valve number.

Saving and Restoring the Entry Buffer

Contents of the entry buffer can be saved in a temporary register, much like saving text to the clipboard in a word processor. This data can then be restored (pasted) into the executing macro code. This is an extremely useful technique made possible by the *%[Save Entry Buffer* and *%] Restore Entry Buffer* macro commands.

```
11.0.18496P%o   copy formatted time to entry buffer
%[              save entry buffer (copy to clipboard)
                                  12:00 am

Time %]P4,1%C   display time on 4X20 VFD (paste clipboard)
                                  12:00 am

Time %]%$       send time out comm port (paste clipboard)
                                  12:00 am
```

In the example above, the **%[** macro command transfers the time from the entry buffer to the temporary register (clipboard). This data is then be pasted into the following macro commands with the **%]** macro command which displays and transmits the time value. Notice that data stored in the temporary register can be pasted more than once. Contents of the temporary register will not change unless updated by another **%[** command. If there is no data in the entry buffer when a **%[** command is encountered, the temporary register (clipboard) is cleared.

Example:
Saving and Restoring an Entry During Macro Execution

5099%s210%e	Setpt 210
5100%s1%e	SPTyp Outpt
5101%sTRANSMIT%e	SPNam TRANSMIT
5110%s5%e	Activ Never
5111%s0.00%e	AcDly 0.00
5112%s250%e	AcMac 250
5130%s4%e	Deact Alwys
5131%s10.00%e	DeDly 10.00
5132%s210%e	DeMac 210
5133%s0%e	DeMtn Ignrd

MACRO #210 – TRANSMIT DATA	
%[	save entry
80.6P+=1%o	increment VAR#6
3%Q	send data
210%A	re-start timer
%]	restore entry

Another important use of the save and restore entry buffer commands is maintaining an operator entry while at the same time allowing a macro to be invoked asynchronously. For example, suppose an operator is keying in a manual tare weight when the macro of Example – *Saving and Restoring an Entry During Macro Execution* is invoked before the **[TARE]** key is pressed. As the example shows, the entry will be stored in the temporary register before proceeding with the following macro commands. When the macro ends, the entry is restored to the entry buffer making the process completely transparent to the operator! Had the save and restore commands been omitted, the entry would have become part of the invoked macro resulting in a macro execution error and a lost tare entry.

A string variable can be used as a substitution for the save and restore commands as used in Example – *Saving and Restoring an Entry During Macro Execution.*

=80.10P%o	copy entry buffer to VAR#10
80.6P+=1%o	increment
3%Q	send data
210%A	re-start timer
80.10P%o	copy VAR#10 to entry buffer

This method allows you to use the **%[** and **%]** commands for other macro functions while still maintaining an operator entry in a variable.

USING MACROS IN THE SETUP MODE

A macro can be used to enter the setup mode and will continue to execute thereafter, even after exiting the setup mode, until it ends. This technique can be used in many ways as shown in the example – *Using Macros to Access the Setup Mode*. Here, macro #1 provides a single **[F1]** keystroke to invoke the calibration mode. The **[SETUP]** key will access the setup mode via macro #8 without having to key in the access code. Macro #2 will also access the setup mode, but only if the program jumper (E4) is in the 'YES' position. It will then proceed to change the name and conversion factor of custom unit #1 as entered in VAR#1 and VAR#2 respectively, thus providing an easy method for changing between any number of custom units.

Example:
Using Macros to Access the Setup Mode

MACRO #1 – ACCESS CALIBRATE MODE	
100%s54321%i%e	access Quick CAL

MACRO #8 – ACCESS SETUP MODE	
100%s23640%i%e	access setup

MACRO #2 – CHANGE CUSTOM UNIT	
0%O	if jumper = YES
UNIT NAME?%G	get entry
=80.1P%o	save entry
CONV.FACTR%G	get entry
=80.2P%o	save entry
100%s23640%i%e	access setup
151.1%s80.1P%o%e	assign unit name
152.1%s80.2P%o%e	assign factor
%z%c%e%e	exit setup
%E	end if

⚠ Once macro execution stops, it is not possible to invoke another macro from within the setup mode. Thus if a macro error occurs causing the macro to abort, you will have to exit the setup mode manually. Also remember that setpoints and weight conversions are not monitored in the setup mode. Do not attempt to check the status of setpoints or weight values while in the setup mode.

Macro #8 in the previous example could be changed to **`%s23640%i%e`** allowing you to key in any parameter number to access directly. For example,

200.1 [SETUP]

would automatically access P200.1 (comm1 baud rate).

MACRO ABORT

In certain situations it may be necessary to stop the execution of a macro. This feature is primarily used during program development as a debug tool, allowing you to break an endless loop or perhaps abort a long entry or computation routine. A macro can be aborted via the front key panel or through serial communications.

Example:

Accessing the Macro Abort Menu

[CLR NO] + [SELECT]

)0 Abort Mac??

[SELECT]

)0 Sspnd Mac??

[SELECT]

)0 Disbl Mac??

[SELECT]

)0 Resum Mac??

THE MACRO ABORT MENU

Pressing **[CLR] + [SELECT]** during macro execution will stop the current macro. (Press **[ZERO] + [TARE] + [SELECT]** to abort a macro on a 460.) If P9981 is set for `Abort Immed`, the current macro is stopped and the macro stack is cleared. If P9981 is set for `Abort Menu`, the current macro is temporarily suspended pending a selection from the Macro Abort Menu shown in Example – *Accessing the Macro Abort Menu*. Press **[SELECT]** to scroll through the abort menu. Press **[ENTER]** to choose the displayed abort option. The four options are as follows:

- `Abort Mac??` (Abort Macro)

  Stops execution of the current macro and clears the macro stack. This selection will only appear if a macro was running when the abort command was issued.

- `Sspnd Mac??` (Suspend Macros)

  Suspends execution of the current macro and inhibits execution of other macros on the stack. A macro invoked when macros are suspended will be pushed onto the macro stack. Macro execution will not resume until `Resum Mac??` is selected from the Macro Abort Menu.

- `Disbl Mac??` (Disable Macros)

  Disables execution of all macros. The macro stack is cleared. Any request to invoke a macro is ignored. Macro execution will not resume until `Resum Mac??` is selected from the Macro Abort Menu.

- `Resum Mac??` (Resume Macros)

  Resumes normal execution of all macros on the macro stack and allows new macros to be invoked and executed. This is the only way to resume normal operation after previously suspending or disabling macros.

Macros will remain suspended or disabled indefinitely until `Resum Mac??` is selected from the Macro Abort Menu. Macro execution will not resume automatically upon exiting the setup mode. The message `Macro Suspd` will be displayed briefly upon exiting the setup mode to remind you that macro execution is inhibited.

The serial macro abort command will only work for comm ports set for "standard" receive at P205 or enabled for standard receive via the %H command.

ABORTING MACROS VIA SERIAL COMMAND

A macro will be aborted if the single decimal byte 248 is received on any of the enabled communication ports during macro execution. If P9981 is set for `Abort Menu`, the Macro Abort Menu is invoked. Characters in the communication port receive buffer are retained.

THE ABORT MACRO

When a macro is aborted, you can specify an "Abort Macro" at P9980. The Abort Macro is executed immediately after aborting the current macro. If a macro is disabled, the Abort Macro will not execute until macros are resumed. Suspending macros will not invoke the Abort Macro.

DISABLING MACROS AT POWER-UP

Macros can be disabled at power-up by pressing and holding the **[CLR]** key when applying power. Continue to hold the **[CLR]** key until the display shows `Macro Disbl`. At this point all macro execution will be inhibited and the keypad will assume standard operation. The beeper volume will be set to maximum. This allows you to troubleshoot problems that occur immediately upon power-up that would otherwise prevent you from accessing the setup mode. When disabling macros by this method, it is only possible to resume macro execution by exiting the setup mode or cycling power.

MACRO LANGUAGE

The 60 Series macro language is a full-featured, straightforward programming language tailored specifically for weight-based process control. This section categorizes the entire macro command set with reference to individual macro command syntax.

MACRO COMMANDS

Macro commands are listed in the following general categories. Some commands may apply to more than one category

- Assignment Commands
- Branching Commands
- Communication Commands
- Comparison Commands
- Database Commands
- Diagnostic Commands
- Display Commands
- Entry Buffer Commands
- Keyboard Commands
- Operator Interface Commands
- Scale Performance Commands
- Setpoint Commands

ASSIGNMENT COMMANDS

Assignment commands write new values to data registers.

MACRO COMMAND	DESCRIPTION	PAGE
%o	Math Assignment	9-112
%o	String Concatenation	9-121
%m	Modify String	9-107
%v	Write to Non-Volatile Memory	9-130

BRANCHING COMMANDS

Branching commands, generally used in conjunction with comparison commands, redirect macro execution to non-sequential program locations.

MACRO COMMAND	DESCRIPTION	PAGE
%^	Call/Goto Macro	9-89
%B	Break Macro	9-50
%T	Tag Position	9-83
%J	Jump to Tag	9-66
%{	Start Group	9-152
%}	End Group	9-152
%N	Else (If Not)	9-75
%E	End If	9-61

COMMUNICATION COMMANDS

Communication commands control keypad and comm port operation.

MACRO COMMAND	DESCRIPTION	PAGE
%!	Enable/Disable Comm Port	9-31
%H	Redefine Comm Port Function	9-63
%”	Select Comm Port	9-32
%$	Send Text	9-34
%&	Send Control Code	9-34
%(	If Character Received	9-37
%)	Clear Receive Buffer	9-40
%’	Record Received Serial Data	9-35
%U	If Transmit Buffer Empty	9-84
%q	Enable RS-485 Transmitter	9-123
%Q	Send Custom Transmit	9-77
%X	Request Display Data	9-86

COMPARISON COMMANDS

Comparison commands are used to evaluate a condition. Branching commands are then used to redirect macro execution depending on whether the condition was true or false.

MACRO COMMAND	DESCRIPTION	PAGE
%{	Start Group	9-152
%}	End Group	9-152
%|	Or	9-152
%N	Else (if not)	9-75
%E	End If	9-61
%\	If No Entry	9-88
%Y	If Yes (Enter)	9-87
%#	If Current Scale	9-34
%u	If Current Units	9-128
%M	If Current Mode	9-74
%Q	If Custom Transmit Continuous	9-78
%U	If Transmit Buffer Empty	9-85
%(	If Character Received	9-37
%m	If Character in String	9-109
%O	If Setpoint Activated / Queued for Activation	9-75
%F	If Setpoint Deactivated / Queued for Deactivation	9-61
%B	If Macro on Stack	9-52
%/	If Macro Interrupted	9-45
%g	If Macro Error	9-104
%_	If Database Error	9-90
%g	If Sample Error	9-104
%a	If Accuracy Achieved	9-93
%f	If Parameter Preset	9-102
%j	If Keypress/Remote Key Held	9-105
%o	If Math Comparison	9-119
%S	If Beeper Program Running	9-82

DATABASE COMMANDS

Database commands are used exclusively with the database option.

MACRO COMMAND	DESCRIPTION	PAGE
1%y	Recall Row	9-131
2%y	Update Row	9-132
3%y	Make Row	9-133
4%y	Print Database	9-133
5%y	First/Last Row	9-134
6%y	Next/Previous Row & Get/Recall Row Number	9-135
7%y	Next Match	9-137
8%y	Delete Row	9-138
9%y	Clear Column	9-138
10%y	Clear Database	9-139
11%y	Set Database	9-140
12%y	Set Column	9-140
13%y	Download Database	9-141
14%y	Print Row	9-143
15%y	Print Errors	9-144
16%y	Upload New	9-145
17%y	Upload Update	9-146
18%y	Sort Database	9-147
19%y	Database Auto-Test	9-148
%_	If Database Error	9-90
%w	DSD Database Functions	9-149

DIAGNOSTIC COMMANDS

Diagnostic commands are used primarily for debugging system operation.

MACRO COMMAND	DESCRIPTION	PAGE
%*	Record A/D Data	9-40
R%`	Reset A/D	9-92
%'	Record Received Serial Data	9-35
%B	Macro Debug Trace Buffer	9-50
E%g	If Macro Error Occurred	9-104
%X	Request Display Data	9-86
19%y	Database Auto-Test	9-148

DISPLAY COMMANDS

Display commands provide direct control of displayed information.

MACRO COMMAND	DESCRIPTION	PAGE
%d	Display Control	9-95
%C	Display Text on 4X20 VFD	9-52
%C	Display Text on 8X40 and 16X40 LCD	9-55
%C	Draw Box/Line on 8X40 and 16X40 LCD	9-57
%I	Refresh Display	9-65
%R	Rename Mode	9-80
%u	Rename Units	9-128
%s	Select Mode	9-126

ENTRY BUFFER COMMANDS

Entry buffer commands copy data to the entry buffer. Data copied to the entry buffer may then be saved or used immediately to check a condition or to serve as part of another macro command.

MACRO COMMAND	DESCRIPTION	PAGE
%c	Clear Entry Buffer	9-94
%[	Save Entry Buffer	9-87
%]	Restore Entry Buffer	9-88
%\	If No Entry	9-88
%#	Get Current Scale	9-33
%k	Get Current Filter	9-106
%L	Get Current Language	9-73
%M	Get Current Mode	9-74
%m	Get String Length	9-108
%m	Get Character Position in String	9-109
%o	Get Value	9-112
%U	Get Number of Characters in Transmit Buffer	9-85
?%a	Get Target Accuracy	9-94
?%g	Get Sample Error	9-104

KEYBOARD COMMANDS

Keyboard commands emulate front panel key functions.

MACRO COMMAND	DESCRIPTION	PAGE
%`	Scale Select	9-92
%c	Clear	9-94
%e	Enter / Sample	9-98
%i	ID	9-104
%p	Print	9-123
%s	Select	9-126
%t	Tare	9-127
%u	Units	9-129
%z	Zero	9-149
1%^	F1 Key	9-8
2%^	F2 Key	9-8
3%^	F3 Key	9-8
4%^	F4 Key	9-8
5%^	F5 Key	9-8
6%^	Start Key	9-8
7%^	Stop Key	9-8
8%^	Setup Key	9-8

OPERATOR INTERFACE COMMANDS

Operator interface commands get user input or provide user feedback.

MACRO COMMAND	DESCRIPTION	PAGE
%[	Save Entry Buffer	9-87
%]	Restore Entry Buffer	9-88
%\	If No Entry	9-88
%G	Get Entry	9-62
%n	Get Numeric Entry	9-111
%K	Get Entry From 4X20 VFD	9-67
%K	Get Entry From 8X40 and 16X40	9-71
%P	Pause	9-76
%S	Sound Beeper	9-81
%W	Wait For Keypress	9-85
%X	Request Display Data	9-86
%Y	If Yes (Enter)	9-87

SCALE PERFORMANCE COMMANDS

Scale performance commands are used to change or indicate the status of various system properties.

MACRO COMMAND	DESCRIPTION	PAGE
%r	Set A/D Interval	9-124
%@	Set Pause Time	9-47
%P	Pause	9-76
%,	Motion Delay	9-44
%s	Select Mode	9-126
%-	Perform Scale Specific Function	9-45
%#	Get Current Scale	9-33
%+	Averaging	9-43
%f	If Parameter Preset	9-102
%k	Digital Filter	9-105
%L	Language Selection	9-72
%M	Mode Selection	9-74
%\	If Macro Interrupt	9-45
%a	Target Accuracy	9-93
%b	Perform Sample	9-94
%g	Sample / Macro Error	9-103

SETPOINT COMMANDS

Setpoint commands are used to change or indicate the status of individual setpoints.

MACRO COMMAND	DESCRIPTION	PAGE
%A	Activate Setpoint	9-48
%D	Deactivate Setpoint	9-58
%O	If Setpoint Activated/Queued for Activation	9-75
%F	If Setpoint Deactivated/Queued for Deactivation	9-61

COMPLETE MACRO COMMAND

This section details all macro commands sequentially by equivalent ASCII value.

< *required* >	Required syntax arguments are enclosed in angle brackets.
[*optional*]	Optional syntax arguments are enclosed in square brackets.
< A | B >	Syntax arguments separated by a vertical bar require one selection, as in '**A**' or '**B**'.

%! ENABLE/DISABLE COMM PORT

Syntax — **Enable/Disable Comm Port Receive**

`[ - ] [ comm ] %!`

Enable/Disable Comm Port Transmit

`[ - ] T [ comm ] %!`

Arguments

-	Disable receive/transmit.
comm	Communication port (0 → 4; port 0 = front panel keypad).

Notes — Omit *comm* to specify all ports.

See Also

%U Transmit Buffer

%) Clear Receive Buffer

%H Redefine Comm Port Function

Enable/Disable Comm Port Receive

`[ - ] [ comm ] %!`

Example:

Enabling Only Comm Port 2

`-%!`	disable all ports
`2%!`	enable port 2
`Enter ID#?%G`	get entry (from comm2)
`=80.4P%o`	store entry
`A%)`	clear all receive buffers
`%!`	enable all ports

Disabling the *comm port receive* inhibits the processing of received serial data. This is useful when it is necessary to ensure that received data can only be processed from one source. Incoming data on disabled ports will continue to be buffered and will be processed when the comm port receive is re-enabled. When the receive buffer becomes full, handshaking (if specified at P204) is asserted informing the connected device to temporarily stop transmitting. Disabling port 0 will disable the front panel keypad. Only the first keypress will be buffered while the keypad is disabled. A buffered keypress will perform its function when the keypad is re-enabled. If received data is to be completely ignored while a port is disabled, clear the buffer with the **%)** command before re-enabling the port.

%!	Enables all ports 0 → 4 to process received data.
0%!	Enables the front panel keypad.
1%!	Enables comm port 1 to process received data.
-%!	Disables all ports 0 → 4 for processing received data.
-0%!	Disables the front panel keypad.
-1%!	Disables comm port 1 for processing received data.

Use caution when disabling the keypad with the **0%!** command. Since the keypad will be disabled, it will not be possible to invoke a macro from the keypad to re-enable it! Make provisions to re-enable the keypad by some other means.

Enable/Disable Comm Port Transmit

`[ - ] T [` *`comm`* `] %!`

Disabling the *comm port transmit* inhibits new data from being transmitted out a specified comm port. Any data already in the transmit buffer will still be transmitted. Subsequent transmission requests will be ignored and transmit data will not be buffered.

T%!	Enables all ports 1 → 4 to transmit data.
T1%!	Enables comm port 1 to transmit data.
-T%!	Disables all ports 1 → 4 for transmitting data.
-T1%!	Disables comm port 1 for transmitting data.

%” *SELECT COMM PORT*

Syntax **Select Comm Port**
< *comm* > %″

Arguments

Comm Communication port (1 → 4).

See Also
%$ Send Text
$& Send Control Code
%U Transmit Buffer

Select Comm Port

Used in conjunction with the **%$** and **%&** macro commands to specify which port macro text and control codes will be transmitted from. The port specified by the **%”** command will remain in effect until changed by another **%”** command. Comm port 1 is automatically selected upon power-up.

Example:

Selecting A Comm Port

`Enter ID# ?%G`	get entry
`=80.5P%o`	save entry
`1%”`	select comm1
`Operator# %$`	send text
`80.5P%o%$`	send VAR#5
`13,10%&`	send <CR><LF>
`2%”`	select comm2
`Operator# %$`	send text
`80.5P%o%$`	send VAR#5

%# CURRENT SCALE

Syntax

Get Current Scale

`[ character ] %#`

If Current Scale

`< scale# > %#`

Arguments

scale#	Scale number (1 → 8).
character	Any character except 1 → 8.

Get Current Scale

`[ character ] %#`

Copies the current scale number to the entry buffer. When used with the optional *character* argument, the preceding character is replaced by the current scale number.

%#	Copies the current scale number to the entry buffer.
_%#	Replaces the preceding character ‘_’ with the current scale number. Any character other than 1, 2, 3, 4, 5, 6, 7 or 8 could be used as the preceding character.

Using the *character* argument is especially useful for recording the current scale number and for using the current scale number as an index within other macro commands.

80.8P=_%#%o	Copy the current scale number to VAR#8
80._%#=0.0P%o	Stores the gross weight of the current scale in variable 1, 2, 3, 4, 5, 6, 7 or 8, as determined by the current scale number.
80.2_%#=1.0P%o	Stores the net weight of the current scale in variable 21, 22, 23, 24, 25, 26, 27 or 28.
1,_%#%y	Recall a row from database 1, 2, 3, 4, 5, 6, 7 or 8, as determined by the current scale number.

Example:

Restoring the Last Selected Scale Number at Power-Up

`80.8P=_%#%o`	save scale number
`MACRO #250 – POWER-UP`	
`80.8P%o`	get scale number
``%` ``	select scale

3,1_%#%y	Make a row in database 11, 12, 13, 14, 15, 16, 17 or 18.
%#%Q	Send custom transmit 1, 2, 3, 4, 5, 6, 7 or 8, as determined by the current scale number.
1._%#%Q	Send custom transmit #1 out port 1, 2, 3 or 4 as determined by the current scale number.

If Current Scale

`< scale# > %#`

Determines if the specified *scale#* argument is the current scale number.

Example:
Branching Based on the Current Scale

1%#	if scale = 1…
101%^	go to macro 101
%N	else
2%#	if scale = 2…
102%^	go to macro 102
%E	end if

1%#	Determine if the current scale is scale #1.
2%#	Determine if the current scale is scale #2.
3%#	Determine if the current scale is scale #3.
4%#	Determine if the current scale is scale #4.

%$ SEND TEXT

Syntax **Send Text**
`[ text ] %$`

Arguments
text Text to be transmitted out selected port.

Notes Omit *text* to send only characters in the entry buffer.

See Also **%” Select Comm Port**
%& Send Control Code

Send Text

Transmits up to 79 alphanumeric characters out the comm port last specified by the **%”** command.

%$	Sends the contents of the entry buffer out the selected port
Station#1%$	Sends the text ‘Station#1’ out the selected port.
80.1P%o%$	Sends the contents of VAR#1 out the selected port.

%& *SEND CONTROL CODE*

Syntax **Send Control Code**

[control code ,] < control code > %&

Arguments

control code ASCII character to be transmitted out selected port (0 → 255).

See Also **%” Select Comm Port**

%$ Send Text

Send Control Code

Transmits a single byte ASCII character out the comm port last specified by the **%”** command. This command is often used in conjunction with the **%$** command to send printer commands such as a carriage return, line feed, or form feed. Refer to the *ASCII Chart* on page B-17 for a list of control codes and text characters. Use the ASCII decimal value when specifying each *control code*.

12%&	Sends <FF> form feed character out the selected port.
13,10%&	Sends a <CR> carriage return and <LF> line feed character out the selected port.
73,68,35%&	Sends the text ‘ID#’ out the selected port.

Passing a value of 256 will generate two control codes, <CR> and <LF> in that order.

%’ *RECORD RECEIVED SERIAL DATA*

Syntax **Display Available Data Collection Memory**

%′

Free Data Collection Memory

F%′

Start Data Collection

S%′

End Data Collection

E < $comm_r$ > %′

Resume Data Collection

R < $comm_r$ > %′

Print Collected Data

`P < commr > < A | B > < commt > %’`

Arguments

A	Print format ‘A’ – print data as Hex characters only.
B	Print format ‘B’ – print data as both Hex and decimal characters with time stamp and port number.
comm$_r$	Communication port (1 → 4; * = all ports) for receiving data.
comm$_t$	Communication port (1 → 4) for transmitting (printing) collected data.

Record Received Serial Data

Serves as a diagnostic tool that collects serial data from all comm ports and stores the received data in the database RAM for analysis. This is especially useful for troubleshooting input interpreter problems. All but 4K of the remaining database RAM will be allocated for data collection (no RAM will be allocated for the 460 Series). There must be at least 526 bytes available (50 rows of data). It takes 10 bytes of storage for every character received. A 4K data base can hold 400 characters and a 256K database about 258,000 characters. To display data from only one port, use multiple E%’ commands to turn off each unwanted comm port

Data is displayed in rows starting with the oldest data first. The oldest data is written after the buffer is filled. In this way the most recent bytes are retained.

⚠ This feature remains enabled, even when the indicator is powered down, until explicitly disabled with the F%’ command.

Example :
Recorded Serial Data Print Format ‘A’

```
54
45
53
54
25
50
```

Collected serial data can be represented in two different print formats:

Format ‘A’ sends each character as the ASCII Hex value followed by a carriage return <CR> (see example – *Print Format ‘A’*).

Format ‘B’ sends a time stamp, the ASCII Hex value, the ASCII decimal value, the ASCII character and the comm port number followed by a carriage return <CR> (see example – *Print Format ‘B’*).

S%’	Allocates all but 4K (0K for 460 Series) of remaining database memory and starts collecting data on all ports.
%’	Displays (for one second) the number of rows of serial data that can be stored. Data is stored 1 byte per row.
E2%’	Ends (suspends) data collection on comm 2.
R2%’	Resumes data collection on comm 2.
E*%’	Ends (suspends) data collection on all comm ports.
R*%’	Resumes data collection on all comm ports.

Example:
Recorded Serial Data Print Format ‘B’

```
12016.463: 0x54  84 T [2]
12016.463: 0x45  69 E [2]
12016.465: 0x53  83 S [2]
12016.466: 0x54  84 T [2]
12016.467: 0x25  37 % [2]
12016.469: 0x50  80 P [2]
```

P2A1%'	Prints data collected on comm 2 out comm 1 using print format 'A'.
P*B1%'	Prints data collected on all comm ports out comm 1 using print format 'B'.
F%'	Ends data collection on all comm ports and frees the database memory previously allocated for serial data collection.

%(IF CHARACTER RECEIVED

Syntax

Get Character from Any Port
`[ text ] G%(`

Get Character from Specified Port
`[ text ] P < comm > g%(`

Get Decimal Value at Interpreter Port
`P < comm > < d | h > %(`

If Any Character at Port
`[ ! ] [ P comm ] %(`

If Specified Character at Port
`[ ! ] < byte1 > [ , byte2 ] [ P comm ] [C] %(`

Arguments

!	Reverses an *if* condition – if character NOT received.
C	Clears character from receive buffer if found.
d	Puts the decimal value (0 → 255) of the next character in the input interpreter's receive buffer into the entry buffer.
h	Puts the decimal value (0 → 15) of the next character in the input interpreter's receive buffer into the entry buffer. The character is assumed to be a Hex character.
text	Alphanumeric text to which a received character will be appended.
comm	Communication port (0 → 4; port 0 = front panel keypad).
$byte_1$	ASCII character (0 → 255) to evaluate at selected port.
$byte_2$	ASCII character (0 → 255) used to specify a range of characters, beginning with $byte_1$, to evaluate at selected port.

Notes

Omit *comm* to test for character on all ports.

Receive buffers are tested in order of port 1, 2, 3, 4, followed by the front panel keypad (port 0).

When *comm* is specified, the port is tested regardless of whether that port has been disabled by the %! command.

When *comm* is omitted, ports disabled by the %! command are not tested.

Tested characters remain in the receive buffer until cleared.

<u>**See Also**</u> **%) Clear Receive Buffer**
%H Redefine Comm Port Function

Get Character From Any Port

`[ text ] G%(`

Ports disabled by the %! command are not tested by the G%(command.

Copies the next character in the receive buffer of any enabled receive port to the entry buffer. Ports are tested in order of priority from 1 to 4. If *text* precedes the **G%(** command, it is copied to the entry buffer where the next character is appended to the text. The example - *Using the %(Command to Get an Entry* shows how to use this command in conjunction with the **%H** command to "build" an operator entry without suspending macro execution as with the **%G** or **%K** commands.

Get Character From Specified Port

`P < comm > g%(`

Copies the next character in the receive buffer of the specified *comm* port to the entry buffer.

P1g%(	Copies the next character from comm port 1 receive buffer to the entry buffer.
ScaleP2g%(	Copies "Scale" to the entry buffer and appends the next character from comm port 2 receive buffer.
%]P3g%(	Restores the entry buffer and appends the next character from comm port 3 receive buffer.

Get Decimal Value at Interpreter Port

`P < comm > < d | h > %(`

Copies characters within an input interpreter's buffer to the entry buffer. Characters can be copied as ASCII Hex or decimal values. This is a useful diagnostic tool for examining the contents of an input interpreter's buffer in the event an expected interpreter match does not occur. It is also possible to use this feature to develop checksum algorithms on received data.

P1d%(	Copies the decimal value of the next character in comm port 1 interpreter buffer to the entry buffer. Possible decimal values are 0 → 255.

Example:
Using the %(Command to Get an Entry

This routine operates similar to the %G or %K "get entry" function except that macro execution is not suspended pending the completion of the entry. Thus other macros can be invoked with an entry in progress.

String VAR#3 is used to build an entry. Only when the entry is complete is the entry value copied to the destination VAR#1. This routine could be easily modified to use VAR#1 as a pointer (80.1p=80.3P%o) thus allowing macro 100 to be used for multiple entries.

```
=================================
681%s1%e         VAR #1
682%sTarget%e    VName Target
684%s0%e         VSave NoSav
685%s0%e         VLock Disbl
686%s0%e         VType Float
687%s6%e         FStyl Auto

681%s3%e         VAR #3
682%sEntry%e     VName Entry
684%s0%e         VSave NoSav
685%s0%e         VLock Disbl
686%s3%e         VType Strng
689%s6%e         Ssize     6
MACRO #1 – ENTRY PROMPT
80.3P=""%o          clear entry
Enter TargetP3,1A%C
P4,1%C              set cursor
4%"                 select comm 4
21%&                blink cursor
0,100%H             invoke macro 100
MACRO #100 – BUILD ENTRY
46,57P0%(           if [.] – [9] pressed…
  %{
  N3%m<6%o            if length < 6…
    G%(                 get keypress
    \=80.3P%o           append entry
    80.3P%o             display entry
    P4,1%C              set cursor
    21%&                blink cursor
  %N                  else
    %)                  clear keypress
  %E                  end if
  %}
%N                  else
----------------------------
229P0C%(            if [ENTER] pressed…
  %{
  N3%m>0%o            if length > 0…
    80.1P=80.3P%o       save entry
  %E                  end if

  P3,1A%C             clear rows 3-4
  0,0%H               keypad normal
  %}
%N                  else
--------------------------------
227P0%(             if [CLR] pressed…
  80.3P=""%o          clear entry
  P4,1c%C             clear row 4
  P4,1%C              set cursor
  21%&                blink cursor
%N                  else
--------------------------------
  %)                  clear keypress
%E                  end if
```

P2h%(Copies the decimal value of the next character in comm port 2 interpreter buffer to the entry buffer. Assuming the received character is a Hex value (i.e. 0 → 9 or A →.F), the output value will be the equivalent decimal value 0 → 15.

If Any Character at Port

`[ ! ] [ P` *`comm`* `] %(`

Determines if a character is available at the specified *comm* port(s).

%(	Determines if any character is available at any comm port.
!%(	Determines if no characters are available at any comm port.
P1%(	Determines if any character is available at comm port 1.
P2%(	Determines if any character is available at comm port 2.
!P3%(	Determines if no characters are available at comm port 3.
P0%(	Determines if a key was pressed on the front panel keypad.
P13%(	Determines if any character is available at comm port 1 or 3.

If Specified Character at Port

`[ ! ] <` *`byte`*$_1$ `> [ ,` *`byte`*$_2$ `] [ P` *`comm`* `] [C] %(`

Determines if a specific character is available at the specified *comm* port(s). It is also possible to determine if a received character falls within a specified range. This is useful for validating characters for numeric-only or alpha-only entries as shown in the previous example.

65%(	Determines if the character 'A' is available at any comm port.
65,90%(	Determines if a character within the range 'A' → 'Z' is available at any comm port.
!97,122%(	Determines if a character within the range 'a' → 'z' is not available at any comm port.
48,57P1%(	Determines if a character within the range '0' → '9' is available at comm port 1.
133P0%(	Determines if the **[START]** key was pressed.
134P0C%(	Determines if the **[STOP]** key was pressed. The keypress is removed from the keypad buffer.
46P014C%(	Determines if a decimal '.' character was received via the front panel keypad or comm ports 1 or 4. The character is removed from the keypad or comm port buffer.

⚠ Checking for characters on comm ports does not remove characters from the receive buffers. This is especially important to note when using this feature in conjunction with the **%H** command. In this case, failure to clear the receive buffer will result in a macro being continuously invoked in an endless loop.

%) CLEAR RECEIVE BUFFER

Syntax	**Clear Receive Buffer** `[A] [ comm ] %)`
Arguments	
A	Clears all characters in specified receive buffer.
comm	Communication port (0 → 4; port 0 = front panel keypad).
Notes	Omit *comm* to specify all ports. Multiple ports may be specified for *comm*.
See Also	**%(If Character Received** **%H Redefine Comm Port Function**

Clear Receive Buffer

Clears one or all characters from the specified *comm* port(s). This is usually done in conjunction with the %) command to clear a character after determining whether or not it exists as the next character in the receive buffer.

%)	Clears the next character from any comm port.
A%)	Clears all characters from all comm ports. Also clears the input interpreter buffer.
1%)	Clears the next character from comm port 1.
24%)	Clears the next character from comm port 2 and 4.
A13%)	Clears all characters from comm ports 1 and 3.

%* RECORD A/D DATA

Syntax

Display Allocated Data Collection Memory

`%*`

Free Data Collection Memory

`F%*`

Start Data Collection

`S < scale# > [ : seconds ] %*`

End Data Collection

`E%*`

Resume Data Collection

`R%*`

Print Collected Data

`P < comm > %*`

Specify Parameter for Data Collection

`C < column > = < parm > . < instance > %*`

Example:
Recorded A/D Data Print Format

```
466,,,,Scale #1
      1,,,, 09/25/99 @ 22:23:06: 50
      2,    -6148,       0.00,       0.00
      3,    -6148,       0.00,       0.00
      4,    -6147,       0.00,       0.00
      5,    -6149,       0.00,       0.00
      6,    -6149,       0.00,       0.00
      7,    -6148,       0.00,       0.00
      8,    -6148,       0.00,       0.00
      9,    -6147,       0.00,       0.00
     10,    -6153,       0.00,       0.00

      ↓        ↓          ↓          ↓

    456,    40819,       9.39,       0.00
    457,    40821,       9.39,       0.00
    458,    40819,       9.39,       0.00
    459,    40823,       9.39,       0.00
    460,    40823,       9.39,       0.00
    461,    40820,       9.39,       0.00
    462,    40820,       9.39,       0.00
    463,    40820,       9.39,       0.00
    464,    40818,       9.39,       0.00
    465,,,, 09/25/99 @ 22:23:14: 33
    466,,,, 09/25/99 @ 22:23:20: 13
```

Arguments

scale#	Scale number (1 → 8).
seconds	A/D data recording buffer size in terms of seconds.
comm	Communication port (1 → 4).
column	Column (1 or 2) to redefine as alternate parameter.
parm	Operating parameter (must be a float-type parameter).
instance	Valid parameter instance.

Notes

Omit *seconds* to allocate all remaining database RAM.

Record A/D Data

Serves as a diagnostic tool that collects A/D and parameter values in the database RAM for analysis. Recorded data can then be transmitted to a PC and imported into a spreadsheet. Here the data can be graphed and analyzed to determine various characteristics of the weighing system such as vibrations, mechanical influences, event timing, rate of flow, etc. (see figure - *Data Graph).*

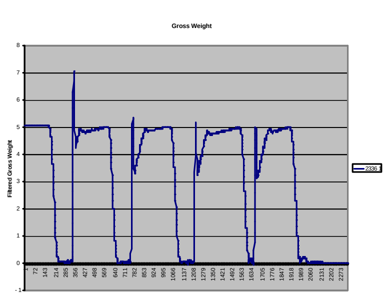


Figure: Data Graph

All but 4K of the remaining database RAM will be allocated for data collection. Data is collected in a first-in-first-out (FIFO) basis collection (no RAM will be allocated for the 460 Series). Thus when data collection begins it will continue indefinitely, maintaining the most recent information. The amount of data that can be stored is determined by the amount of available database RAM. A 4K database can store 333 rows while a 256K database can store over 20,000 rows of data. Since the A/D update rate is updated at 60 times per second 4K of database translates to 5.5 seconds of data, while 256K of database translates to almost 6 minutes of data respectively. A 2-Meg database can record over 45 minutes of data!

Collected A/D data can be transmitted out any comm port. Data is sent in a fixed width, comma delimited format as shown in example – *Recorded A/D Data Print Format*. The first column is a sequential record number, followed by the raw A/D value, followed by two selectable parameter columns. By default, these columns represent the filtered gross weight (parm 0.0) and the rate (parm 23.0) on the current scale. (See parms 135 & 136 for more information on rate). They can be changed to any float type parameter. The data includes a header row with the number of data samples taken and the scale number on which this data was acquired. The first row is a time stamp at which the data collection was started or resumed. If data collection has been left running so that the buffer has filled then this will contain data. Data is printed in oldest first order (i.e. lowest row number = oldest data). The last row will contain a time stamp showing the time it was printed. If data collection has been stopped, the last non-data entry will contain the time stamp at which data collection was stopped. Data collection stops when requested by the user or after setup mode is entered (which includes displaying of any information parameter such as the amount of NV ram available). Repeatedly printing data after data collection is stopped will add another time stamp to the end of the list.

It is only possible to redefine the selectable collection columns <u>after</u> allocating database memory using the S%* command.

It is only possible to redefine the selectable collection columns <u>after</u> allocating database memory using the S%* command.

This feature remains enabled, even when the indicator is powered down, until explicitly disabled with the F%* command.

S1%*	Allocates all but 4K (0K for the 460 Series) of remaining database memory and starts collecting data for scale 1.
S13%*	Allocates all but 4K (0K for the 460 Series) of remaining database memory and starts collecting data for scales 1 and 3.
S2:120%*	Allocates enough database memory to buffer 120 seconds of data and starts collecting data for scale 2.
%*	Displays (for one second) the number of data rows that can be stored for each scale.
C1=1.0%*	Redefines the first selectable collection column (3) to record the net weight.
C2=50.3%*	Redefines the second selectable collection column (4) to record a frequency input on PDIO channel 3.
E%*	Ends (suspends) data collection for all scales.
R%*	Resumes data collection for all for scales.
P1%*	Prints data collected out comm 1.
F%*	Ends data collection for all scales and frees the database memory previously allocated for A/D data collection.

%+ AVERAGING

<u>Syntax</u>

Start Averaging
`S < scale# > %+`

End Averaging
`E < scale# > %+`

Resume Averaging
`R < scale# > %+`

<u>Arguments</u>

scale# Scale number (1 → 8).

<u>Notes</u> Multiple scales may be specified.

<u>See Also</u> **Weight Averaging Parameters**

Averaging

Averages the gross or net weight over a specified period of time. The averaging is performed at a rate of up to 60 times per second and can continue indefinitely once started by the **S%+** command. This feature is ideal for in-motion weighing systems such as high-speed check-weighing, in-motion truck scales or live animal weighing.

S1%+	Starts averaging for scale 1.
S123%+	Starts averaging for scales 1, 2 and 3.
E2%+	Ends (suspends) averaging for scale 2.
R2%+	Resumes averaging for scale 2.

Once averaging begins, **15P** and **16P** become active weight parameters continuously recalculating the average filtered gross and net weight until averaging is stopped. If the digital filter is set to 1 second or less at P116 (or by the *%K Digital Filter* macro command), then the average weight is recalculated every 1/60th second). If the filter is set for 2, 4 or 8 seconds, then the average weight is recalculated every 2/60th, 4/60th or 8/60th seconds respectively.

The average gross and net parameters contain an accumulated weight value. The average weight is calculated when accessed by dividing the accumulated weight by the average count of **17P**. Thus if a value is to be entered directly into **15P**, the average count should first be assigned at **17P**. When recalling a gross average value from a database, the average count column should precede the average gross column.

%, *MOTION DELAY*

Syntax **Motion Delay**

```
M < scale# > [ ;prompt ] %,
```

Arguments

prompt Text to be displayed as a prompt on the 2X5 character matrix of the 7-segment VFD.

scale# Select from the following scale numbers:

0 Current scale
1 Scale 1
2 Scale 2
3 Scale 3
4 Scale 4
5 Scale 5
6 Scale 6
7 Scale 7
8 Scale 8
* All scales

Motion Delay

Suspends macro execution until a stable weight is achieved as determined by the settings for P114 (motion divisions) and P115 (motion delay).

M0%,	Pause during motion on the currently selected scale.
M1%,	Pause during motion on scale 1.
M*%,	Pause during motion on all scales. All scales must be stable before macro execution will resume.
M1;Mot'nDelay%,	Display "Mot'n Delay" while pausing for motion on scale 1.

%- *PERFORM SCALE SPECIFIC FUNCTION*

Syntax **Perform Scale Specific Function**

< `|t|u|z > S < *scale#* > [;*argument*] %-

Arguments

`	Scale select
t	Tare
u	Units
z	Zero
scale#	Scale number (1 → 8).
argument	Valid argument to the %`, %t, %u or %z command.

Notes

The %- command is mode independent. Use caution when using this command to ensure the scale for which the function is intended will not be performing a critical function at the time the function is executed.

This command replaces the 660 Series mode independent commands previously available with the %`, %t, %u and %z commands.

See Also

%`	**Scale Select**
%t	**Tare**
%u	**Units**
%z	**Zero**

Perform Scale Specific Function

Performs a mode independent scale-select, tare, units or zero function on a specific scale. These functions can be performed on the currently selected scale as well as all other enabled scales regardless of the current mode of operation. Motion delayed commands will be motion delayed for the intended scale whether or not it is the currently selected scale.

Commands that normally accept arguments such as 'tare' can also pass these arguments with the %- command.

`S3%-	Selects scale 3 as the current scale.
tS2%-	Performs an auto-tare on scale 2.
tS1;10%-	Establishes a tare weight of 10 on scale 1. This operation is considered a manual tare entry. If preset is enabled at P412, the preset status is considered preset.
uS1%-	Toggles the units selection for scale 1.
uS2;2%-	Selects the third enabled units (P133) for scale 2. Note that the first enabled unit (P131) is considered unit #0, the second enabled unit (P132) is unit #1 and so on.
zS4%-	Zeros scale 4.
zS1;15R%-	Performs a relative zero offset of 15 for scale 1.

%/ IF MACRO INTERRUPTED

Syntax **If Macro Interrupted**
%/

See Also **Interrupt Macros**

Example:
Re-Prompt Entry if Macro Interrupted

```
1%T                          tag 1
Enter ID%G               get entry
** INTERRUPT HERE AND RETURN **
%/                           if interrupted…
  1%J                          jump to tag 1
%E                           end if
```

If Macro Interrupted

Determines if one macro was interrupted by another macro set for Invok Immed at P9992. When a macro is configured as an interrupt macro at P9992, an internal flag is set whenever the interrupt macro is invoked during the execution of another macro. The **%/** command will be true if this flag is set. When the interrupt macro ends and the interrupted macro resumes, it is possible to determine if there was an interruption. Thus if a macro was interrupted during an entry as with the **%G** or **%K** command, you can jump back to the beginning of the prompt routine as shown in example – *Re-Prompt Entry if Macro Interrupted*.

The following is a list of operator interface commands that can be interrupted that would otherwise suspend macro execution:

%P	Pause
%Y	If Yes
%W	Wait for Keypress
W%r	Wait for A/D Interval
%K	Get Entry from 4X20 VFD
%G	Get Entry
%n	Get Numeric Entry
%z	Zero
%t	Tare
%p	Print

Any of the commands listed above will reset the internal interrupt flag for the ***%I*** command. Therefore, if any of the above commands are used in the interrupting macro, the interrupted flag would be cleared and the interrupted macro would be unable to discern that an interrupt had occurred.

%@ SET PAUSE TIME

Syntax **Set Pause Time**

```
< seconds > %@
```

Arguments

seconds Number of seconds (0.01 → 5,000,000) to pause macro execution for each %P command encountered.

Notes Pause time is set to 1 second at power up.

A new pause time remains in effect until changed by another %@ command.

See Also **%P Pause**

Set Pause Time

Defines the time period for each **%P** command. By default, the pause time is 1 second. The current pause time period remains in effect until changed by another **%@** command.

10%@ Sets the pause time period to 10 seconds.

1%@ Sets the pause time period to 1 second (default).

Example:
Setting the Pause Time Period

`5%@`	pause time = 5 seconds
`BatchDone!%P`	prompt and pause
`1%@`	pause time = 1 second

%A ACTIVATE SETPOINT

Syntax

Activate Setpoint

[L | U] < *setpoint#* > [: *delay*] %A

Activate All Setpoints

* [L | U] %A

Arguments

L	Locks specified setpoint(s) in an active state.
U	Unlocks specified setpoint(s).
setpoint#	Setpoint(s) (1 → 256) to be activated.
delay	Delay time in seconds (0.002 → 5,767,168) before the specified setpoint(s) will activate.

Notes

The %A command does not apply to setpoints configured as inputs (except inputs used by Modbus to invoke macros).

A range or list of setpoints may be specified for *setpoint#* as described in the following examples.

A *delay* cannot be specified when locking or unlocking setpoints.

A variable value can be substituted for *setpoint#* and/or *delay* using the syntax

< variable# >P

where *variable#* is a valid variable 1 → 999.

See Also

%D Deactivate Setpoint

%O If Setpoint Activated

Activate Setpoint

[L | U] < *setpoint#* > [: *delay*] %A

Activates any setpoint configured as an output or disabled.

An activation *delay* can be specified to postpone the activation of a setpoint after the **%A** command is executed. Specifying a *delay* will override the activation delay setting at P5111. If an activation delay has not expired before issuing another activation delay for the same setpoint, the delay timer is reset to the new delay time. Macros assigned at P5112 to be invoked upon activation of a setpoint will not execute until the activation delay time has expired. If *delay* is omitted from the activation command, the macro assigned at P5112 will not be invoked unless a delay time is specified at P5111. If *delay* is specified with a value of zero (0), then any delay specified at P5111 is cancelled and the macro assigned at P5112 will not be invoked.

Example:
Canceling an Activation Delay Without Invoking a Macro

If the activation delay of setpoint #2 expires precisely after the operator presses **[START]**, macro #9 will be pushed on the stack. Removing macro #9 from the stack prevents it from being invoked should this happen.

```
==================================
5099%s2%e          SPT #2
5100%s1%e          SPTyp Outpt
5101%sMIXER%e      SPNam MIXER
5110%s5%e          Activ Never
5111%s4.00%e       AcDly 4.00
5112%s9%e          AcMac     9
5130%s5%e          Deact Never
5131%s0.00%e       DeDly 0.00
5132%s0%e          DeMac None

MACRO #6 – MANUAL START
MANUAL STARTP4,1c%C
2:0%A              start mixer
9%B                clear macro from stack

MACRO #9 – MIXER STARTED
AUTO STARTP4,1c%C
```

Example:
Locking a Setpoint During a Weight Surge

This setup operates a fill output with setpoint #1. When **[START]** is pressed, setpoint #1 activates and locks. This prevents the initial product surge from deactivating the fill output prematurely. After a 2-second delay, setpoint #200 deactivates invoking macro #12. This unlocks the fill output after the weight surge allowing the output to eventually deactivate at the appropriate weight.

```
==================================
5099%s1%e           SPT #1
5100%s1%e           SPTyp Outpt
5101%sFILL%e        SPNam FILL
5110%s5%e           Activ Never
5111%s0.00%e        AcDly 0.00
5112%s0%e           AcMac None!
5130%s0%e           Deact Above
5131%s0.00%e        DeDly 0.00
5132%s13%e          DeMac Mc 13
5133%s0%e           DeMtn Ign'd
5134%s80.5%e        DLPar TARGET
5150%s1.0%e         CmPar Net

5099%s200%e         SPT #200
5100%s1%e           SPTyp Outpt
5101%sSPIKE%e       SPNam SPIKE
5110%s5%e           Activ Never
5111%s0.00%e        AcDly 0.00
5112%s0%e           AcMac None!
5130%s5%e           Deact Never
5131%s0.00%e        DeDly 0.00
5132%s12%e          DeMac Mc 12

MACRO #6 – START FILL
L1%A                lock output
200%A               activate timer
200:2%D             start spike delay

MACRO #12 – UNLOCK FILL OUTPUT
U1%A                unlock output

MACRO #13 – FILL COMPLETE
Fill Done!%S%P      prompt
```

Activating a setpoint with a delay of zero (0) does not guarantee that the macro specified at P5112 will not be invoked. If an activation delay was already in progress, the delay may have expired during execution of the macro that is supposed to cancel the delay, resulting in that macro being placed on the macro stack and invoked upon completion of the cancellation macro. To prevent this, always clear the unwanted macro from the stack immediately after canceling the activation delay (see example – *Canceling an Activation Delay Without Invoking a Macro*).

Setpoints can also be "locked" in an active state to prevent unwanted deactivation. This technique is often used in filling applications where the initial surge of product can actually spike the weight reading above the target value, potentially deactivating the setpoint prematurely (see example – *Locking a Setpoint During a Weight Surge*). An active-locked setpoint cannot be deactivated outside the setup mode by <u>any</u> means. It must first be unlocked before being deactivated. Note that unlocking a setpoint does not automatically change its state.

1%A	Activates setpoint 1 immediately unless a delay is specified at P5111. A macro specified at P5112 will not be invoked unless a delay is specified at P5111.
1:10%A	Activates setpoint 1 in 10 seconds. A macro specified at P5112 will be invoked after the 10 second delay regardless of any delay specified at P5111.
5:0%A	Activates setpoint 5 immediately. A macro specified at P5112 will not be invoked.
L1%A	Activate and lock setpoint 1 immediately without invoking the macro specified at P5112.
U1%A	Unlock setpoint 1 without changing its state. No macros are invoked as a direct result of this command.
17-32%A	Activates setpoints 17 through 32 immediately unless a delay is specified at P5111. A macro specified at P5112 will not be invoked unless a delay is specified at P5111.

A group of setpoints can be activated by specifying a range and/or comma delimited list of setpoint numbers. The criteria for activation delays, invoking macros are the same as previously described.

1,3,5%A	Activates setpoints 1, 3 and 5.
1-8,15,16%A	Activates setpoints 1 through 8, 15 and 16.
1-4,5:10%A	Activates setpoints 1 through 4 immediately and activates setpoint 5 after a 10 second delay.

When locking or unlocking a group of setpoints, only the setpoints listed without a delay time will be locked or unlocked. Those with a specified delay time will activate when the delay expires.

L1,3,5%A	Activates and locks setpoints 1, 3 and 5.
U1,3,5%A	Unlock setpoints 1, 3 and 5.

L7-10,13:10%A	Activates and locks setpoints 7 through 10. Setpoint 13 will activate after a 10 second delay.
U1-7,9:5,21-23%A	Unlocks setpoints 1 through 7 and 21 through 23. Setpoint 9 will activate after a 5 second delay.

The current value of a setpoint's activation delay timer can be accessed via 76P and 77P. See *Setpoint Timers* for more details.

Variable values can also be used to specify a setpoint number or activation delay. This technique is useful when recalling setpoint numbers from a database to serve as valve numbers, mixer numbers, etc. This allows you to write one routine to handle a fill routine for multiple ingredients.

1P%A	Activates the setpoint specified by the value of VAR#1.
5P:6P%A	Activates the setpoint specified by the value of VAR#5 using the delay specified by the value of VAR#6.

Activate All Setpoints

`* [ L | U ] %A`

Immediately activates all setpoints. No delay time can be specified. Delays in progress are cancelled.

***%A**	Activates all setpoints immediately, overriding all delays. No macros are invoked as a direct result of this command.
***L%A**	Activate and lock all setpoints immediately without invoking macros.
***U%A**	Unlock all setpoints without changing states. No macros are invoked as a direct result of this command.

%B BREAK MACRO

Syntax

Abort Current Macro
%B

Abort All Other Macros
$%B

Abort All Macros
*%B

Remove Macro from Stack
[D] < *macro#* > %B

If Macro on Stack
? < *macro#* > %B

Clear Macro Trace Buffer

TB%B

Reset Macro Trace Timer

TT%B

Suspend Macro Trace

TS [*] [*macro#*] %B

Resume Macro Trace

TR [*] [*macro#*] %B

Arguments

D	Remove only first occurrence of specified macro from stack.
*	Suspend or resume all macros individually.
macro#	Macro number (1 → 250).

See Also

%^ Call \ Go To Macro

%/ If Macro Interrupted

Abort Current Macro

%B

Stops execution of a macro before its natural end. If the current macro was called by another macro, the calling macro is removed from the macro stack. This command does not remove additional occurrences of the current macro from the macro stack.

Abort All Other Macros

$%B

Clears the macro stack but allows the current macro to continue execution. This command could be used in an interrupt macro to ensure that no other system functions are queued before proceeding.

Abort All Macros

*%B

Clears the entire macro stack and stops execution of the current macro.

Remove Macro From Stack

[D] < *macro#* > %B

Clears one or all occurrences of a specified macro from the macro stack.

10%B	Removes all occurrences of macro 10 from the macro stack.

Example:
Using the %B Command

This routine uses VAR#1 to count how many times macro 10 appears on the macro stack. If it is not on the stack, the macro stack is cleared.

=================================

80.1P=0%o	clear VAR#1
%T	tag
?10%B	if macro on stack…
80.1P+=1%o	increment VAR#1
D10%B	remove from stack
%J	jump to tag
%E	end if
80.1P==0%o	if VAR#1 = 0…
*%B	abort all macros
%E	end if
10%^	go to macro 10

D5%B	Removes one occurrence of macro 5 from the macro stack.

If Macro on Stack

`? < macro# > %B`

Determines if a specified macro is on the stack.

?10%B	Determines if macro 10 is on the macro stack.

Macro Debug Trace Buffer

Controls the data recorded in the macro debug table at P50001.

TB%B	Clears the macro trace buffer.
TT%B	Resets the macro trace timer to zero (0).
TS%B	Suspends tracing for all macros collectively.
TR%B	Resumes tracing for all macros collectively.
TS*%B	Suspends tracing for all macros individually. Each macro may be resumed individually.
TR*%B	Resumes tracing for all macros that were individually suspended.
TS10-50%B	Suspends tracing individually for macros 10 through 50.
TR10-20%B	Resumes tracing for macros 10 through 20 that were suspended individually.

The TR%B will not resume tracing for macros suspended individually.

Example:
Disabling Standard Text on the 4X20 VFD

```
GSE SCALE SYSTEMSP1,2a%C
4X20 VFDP2,7%C
Use %%C to send text;P3,1%C
Use %%K to get entry.P4,1%C
```

```
GSE SCALE SYSTEMS
    4X20 VFD
Use %C to send text;
Use %K to get entry.
```

```
P3,1A%C
```

```
GSE SCALE SYSTEMS
    4X20 VFD
```

```
11.0.18433P%oP4,1%C
```

```
GSE SCALE SYSTEMS
    4X20 VFD

12:00:01 pm 09/30/99
```

```
P4,13C%C
```

```
GSE SCALE SYSTEMS
    4X20 VFD

12:00:01 pm
```

%C DISPLAY TEXT ON 4X20 VFD

Syntax

Display Standard Text on 4X20 VFD
`[ text ] P [ row , column ] [ clear ] %C`

Display Extended Text on 4X20 VFD
`[ text ] p [ row , column ] [ clear ] %C`

Arguments

text	Text to be displayed beginning at cursor position.
row	Row (1 → 4) to position cursor.
column	Column (1 → 20) to position cursor.

clear	Select one of the following clear options: a Clears entire display before displaying text A Clears to the end of display after displayed text c Clears entire line before displaying text C Clears to the end of line after displayed text
Notes	Omitting *row* and *column* assumes the current cursor position.
See Also	**%d Display Control** **4X20 VF Display**

Display Standard Text on 4X20 VFD

`[ text ] P [ row , column ] [ clear ] %C`

Sends text out comm port 4 to be displayed on the 4x20 VFD. This command uses an upper case 'P' to separate preceding text from the row/column coordinates where the text will be displayed.

Pa%C	Clears the entire display. Cursor position does not change.
P2,1a%C	Clears the entire display and positions the cursor on line 2, column 1.
P2,1%C	Positions the cursor on line 2, column 1.
P2,1c%C	Clears line 2 and positions the cursor on line 2, column 1.
P2,1A%C	Clears lines 2, 3 & 4 and positions the cursor on line 2, column 1.
Enter TargetP3,1%C	Displays text starting on line 3, column 1.
Enter TargetP3,1c%C	Clears line 3 and displays text starting on line 3, column 1.
Enter TargetP3,1A%C	Clears lines 3 & 4 and displays text starting on line 3, column 1.
11.0.18496P%oP3,5C%C	Displays current time starting on line 3, column 5, clearing to the end of line 3.

Using the **%C** command to display standard text will cancel a blinking cursor. You can maintain a blinking cursor after a **%C** command by transmitting an ASCII decimal value of 21 to the display (i.e. via *%& Send Control Code* command or by including '\021' within the **%C** extended text command).

Example:
Disabling Extended Text on the 4X20 VFD

The lower case 'p' identifies the ASCII decimal values 025, 153 and 021 as the commands to select the Katakana character set, display the right-arrow, and blink the cursor.

```
================================
CUSTOMER ENTRY FORMP1,1a%C
Customer Name?P3,1%C
\025\153\021p4,1%C
```

```
CUSTOMER ENTRY FORM

Customer Name?
™
```

Display Extended Text on 4X20 VFD

`[ text ] p [ row , column ] [ clear ] %C`

Allows non-displayable control codes and extended ASCII characters to be transmitted to the LCD. This command is identical to the *display standard text on 4X20 VFD* with the exception of the position identifier 'p'. Here, the lower case 'p' is used to identify three digits preceded by a backslash '\' character as a single ASCII decimal value.

For example,

`\025\152  Use Arrow Keys  \154p1,1a%C`

will display

˜   U s e   A r r o w   K e y s   š

In this command, the '\025' is sent as a single ASCII decimal value. This control code selects the Katakana character set. The '\152' and '\154' ASCII decimal values are extended control codes that display the down and up arrows. You can also use this method to send display commands such as blink cursor, backspace, scroll mode, etc.

International characters are used when the top line of the 4X20 VFD is enabled for auto-update. When these symbols must be displayed, the international character set is automatically selected, possibly resulting in the display of unexpected symbols when using the Katakana character set. Always reselect the Katakana character set when the top line auto-update is enabled for Katakana characters.

%C DISPLAY TEXT ON 8X40 AND 16X40 LCD

Syntax

Display Standard Text on LCD

`[ text ] P [ row , column ] [ clear ] [ , F|f size ] %C`

Display Extended Text on LCD

`[ text ] p [ row , column ] [ clear ] [ , F|f size ] %C`

Arguments

F	Normal text (black on white)
f	Inverse text (white on black)
text	Text to be displayed beginning at cursor position.
row	Row (1 → 16) to position cursor.
column	Column (1 → 40) to position cursor.

clear	Select one of the following clear options: a Clears entire display before displaying text c Clears entire line before displaying text C Clears to the end of line after displayed text
size	Select one of the following font size options: 1 Small font size (H = 1 line, W = 1 column) 2 Medium font size (H = 2 lines, W = 2 columns) 4 Large font size (H = 4 lines, W = 4 columns)

Notes

Omitting *row* and *column* assumes the current cursor position.

Once a font *size* is selected, that size remains in effect until changed.

Addressing the LCD does not utilize the transmit port of comm 4 as with the 4X20 VFD.

See Also **%d** **Display Control**

Display Standard Text on 8X40 and 16X40 LCD

`[ text ] P [ row , column ] [ clear ] [ ,F|f size ] %C`

Sends text out the LCD interface to be displayed on the 4x20 VFD. This command uses an upper case ‘P’ to separate preceding text from the row/column coordinates where the text will be displayed.

Pa%C	Clears the entire display. Cursor position does not change.
P2,1a%C	Clears the entire display and positions the cursor on line 2, column 1.
P2,1%C	Positions the cursor on line 2, column 1.
P2,1c%C	Clears line 2 and positions the cursor on line 2, column 1.
TOTALP7,1,F2%C	Displays medium size standard text starting on line 7, column 1.
TOTALP7,1a,f2%C	Clears the entire display and displays medium size inverse text starting on line 7, column 1.
WELCOMEP3,7a,F4%C	Clears the entire display and displays large size standard text starting on line 3, column 7.
11.0.18496P%oP3,5C,f1%C	Displays current time in small size inverse text starting on line 3, column 5, clearing to the end of line 3.

Display Extended Text on 4X20 VFD

`[ text ] p [ row , column ] [ clear ] [ ,F|f size ]%C`

Allows non-displayable control codes and extended ASCII characters to be transmitted to the LCD. This command is identical to the *display standard text on LCD* with the exception of the position identifier 'p'. Here, the lower case 'p' is used to identify three digits preceded by a backslash '\' character as a single ASCII decimal value.

For example,

```
\179  Use Arrow Keys  \180p1,1a,F2%C
```

will display

```
↑   U s e   A r r o w   K e y s   ↓
```

The '\179' and '\180' ASCII decimal values are extended control codes that display the up and down arrows. Refer to Appendix C for a complete list of displayable LCD characters.

%C DRAW BOX / LINE ON 8X40 AND 16X40 LCD

Syntax

Draw Line / Box on LCD

`<start row> , <start column> , <end row> , <end column> < W|w > <line code> [ C|c|A ] B%C`

Arguments

start row	Row (1 → 16) to position cursor at starting coordinate.
start column	Column (1 → 40) to position cursor at starting coordinate.
end row	Row (1 → 16) to position cursor at ending coordinate.
end column	Column (1 → 40) to position cursor at ending coordinate.
W	Normal line (black on white).
w	Inverse line (white on black).
line code	A line style number (1 → 21). See Table 9-2.
	1 → 7 box lines
	8 → 15 horizontal lines
	16 → 21 vertical lines
C	Clears characters within the box (clear blank / white).
c	Clears characters within the box (clear solid / black).
A	Clears characters with alternating pixels (gray fill).

Notes	Variable values can be substituted for all numeric values. See *Substituting Variables for Numeric Values* in the following sections.

Draw Box on 8X40 and 16X40 LCD

To draw a box on the LCD, specify beginning (upper-left) and ending (lower-right) coordinates. Use *line codes* 1 → 7 (see Table 9-2).

5,20,8,40W1B%C	Draws a box in the lower-right corner of the 8X40 LCD, black lines (style 1)
5,1,8,40w2AB%C	Draws a box in the bottom half of the 8X40 LCD, white lines (style 2), gray fill
1,1,16,40W7CB%C	Draws a box bordering the entire 16X40 LCD, black lines, clear box

Draw Horizontal Line on 8X40 and 16X40 LCD

To draw a horizontal line on the LCD, specify the same *start row* and *end row*. Use *line codes* 8 → 15 (see Table 9-2).

5,1,5,40W8B%C	Draws a horizontal black line across row 5 (style 8)

Draw Vertical Line on 8X40 and 16X40 LCD

To draw a vertical line on the LCD, specify the same *start column* and *end column*. Use *line codes* 16 → 21 (see Table 9-2).

1,20,8,20W1B%C	Draws a vertical black line down column 20 (style 1)

Substituting Variables for Numeric Values

Variables can be substituted in the syntax for all numeric values. One possible use of this feature is storing box/line coordinates in a database.

1P,2P,3P,4PW1PB%C	Draws a box using the coordinates specified by the values stored in variables 1, 2, 3 and 4, black line (style specified by value of variable 5)

Table 9-2: Box / Line Drawing Styles

STYLE	DESCRIPTION
1	Box : 1-pixel line width
2	Box : 2-pixel line width
3	Box: 3-pixel line width
4	Box: 4-pixel line width
5	Box: 5-pixel line width
6	Box: 1-pixel line width; double border
7	Box: 2-pixel line width; double border
8	Horizontal line: 1-pixel line width
9	Horizontal line: 2-pixel line width
10	Horizontal line: 3-pixel line width
11	Horizontal line: 4-pixel line width
12	Horizontal line: 5-pixel line width
13	Horizontal line: 6-pixel line width
14	Horizontal line: 7-pixel line width
15	Horizontal line: 8-pixel line width
16	Vertical line: 1-pixel line width
17	Vertical line: 2-pixel line width
18	Vertical line: 3-pixel line width
19	Vertical line: 4-pixel line width
20	Vertical line: 5-pixel line width
21	Vertical line: 6-pixel line width

%D DEACTIVATE SETPOINT

Syntax

Deactivate Setpoint

`[ L | U ] < setpoint# > [ : delay ] %D`

Deactivate All Setpoints

`* [ L | U ] %D`

Arguments

L — Locks specified setpoint(s) in a deactivated state.

U — Unlocks specified setpoint(s).

setpoint# — Setpoint(s) (1 → 256) to be deactivated.

delay — Delay time in seconds (0.002 → 5,767,168) before the specified setpoint(s) will deactivate.

Notes

The %D command does not apply to setpoints configured as inputs (except Modbus inputs used to invoke macros).

A range or list of setpoints may be specified for *setpoint#* as described in the following examples.

A *delay* cannot be specified when locking/unlocking setpoints.

A variable value can be substituted for *setpoint#* and/or *delay* using the syntax

`< variable# >P`

where *variable#* is a valid variable 1 → 999.

See Also **%A Activate Setpoint**
%F If Setpoint Deactivated

Deactivate Setpoint

`[ L | U ] < `*`setpoint#`*` > [ : `*`delay`*` ] %D`

Deactivates any setpoint configured as an output or disabled.

Example:
Canceling a Deactivation Delay Without Invoking a Macro

If the deactivation delay of setpoint #2 expires precisely after the operator presses **[STOP]**, macro #9 will be pushed on the stack. Removing macro #9 from the stack prevents it from being invoked should this happen.

```
=================================
5099%s2%e            SPT #2
5100%s1%e            SPTyp Outpt
5101%sMIXER%e        SPNam MIXER
5110%s5%e            Activ Never
5111%s0.00%e         AcDly 0.00
5112%s0%e            AcMac None!
5130%s4%e            Deact Alwys
5131%s60.00%e        DeDly 60.00
5132%s9%e            DeMac     9
5133%s0%e            DeMtn Ing'd

MACRO #6 – MANUAL START
2%A                  start mixer
Mixing...P4,1c%C

MACRO #7 – MANUAL STOP
2:0%D                stop mixer
9%B                  clear macro from stack
Mixer Stopped!P4,1c%C

MACRO #9 – MIX COMPLETE
Mix Complete!P4,1c%C
```

A deactivation *delay* can be specified to postpone the deactivation of a setpoint after the **%D** command is executed. Specifying a delay overrides the deactivation delay setting at P5111. If a deactivation delay has not expired before issuing another deactivation delay for the same setpoint, the delay timer is reset to the new delay time. Macros assigned at P5112 to be invoked upon deactivation of a setpoint will not execute until the deactivation delay time has expired. If delay is omitted from the deactivation command, the macro assigned at P5112 will not be invoked unless a delay time is specified at P5111. If delay is specified with a value of zero (0), then any delay specified at P5111 is cancelled and the macro assigned at P5112 will not be invoked.

Deactivating a setpoint with a delay of zero (0) does not guarantee that the macro specified at P5112 will not be invoked. If a deactivation delay was already in progress, the delay may have expired during execution of the macro that is supposed to cancel the delay, causing that macro to be placed on the macro stack and to be invoked upon completion of the cancellation macro. To prevent this, always clear the unwanted macro from the stack immediately after canceling the deactivation delay (see example – *Canceling a Deactivation Delay Without Invoking a Macro*).

Setpoints can also be "locked" in a deactive state to prevent unwanted activation. This technique is often used in emergency stop routines to prevent outputs from activating when the normal activation condition occurs. A deactive-locked setpoint cannot be activated by <u>any</u> means. It must first be unlocked before being activated. Note that unlocking a setpoint does not automatically change its state.

1%D	Deactivates setpoint 1 immediately unless a delay is specified at P5131. A macro specified at P5132 will not be invoked unless a delay is specified at P5131.
1:10%D	Deactivates setpoint 1 in 10 seconds. A macro specified at P5132 will be invoked after the 10 second delay regardless of any delay specified at P5131.
5:0%D	Deactivates setpoint 5 immediately. A macro specified at P5132 will not be invoked.
L1%D	Deactivate and lock setpoint 1 immediately without invoking the macro specified at P5132.
U1%D	Unlock setpoint 1 without changing its state. No macros are invoked as a direct result of this command.

17-32%D	Deactivates setpoints 17 through 32 immediately unless a delay is specified at P5131. A macro specified at P5132 will not be invoked unless a delay is specified at P5131.

A group of setpoints can be deactivated by specifying a range and/or comma delimited list of setpoint numbers. The criteria for deactivation delays, invoking macros are the same as previously described.

1,3,5%D	Deactivates setpoints 1, 2 and 3.
1-8,15,16%D	Deactivates setpoints 1 through 8, 15 and 16.
1-4,5:10%D	Deactivates setpoints 1 through 4 immediately and deactivates setpoint 5 after a 10 second delay.

When locking or unlocking a group of setpoints, only the setpoints listed without a delay time will be locked or unlocked. Those with a specified delay time will deactivate when the delay expires.

L1,3,5%D	Deactivates and locks setpoints 1, 3 and 5.
U1,3,5%D	Unlock setpoints 1, 3 and 5.
L7-10,13:10%D	Deactivates and locks setpoints 7 through 10. Setpoint 13 will deactivate after a 10 second delay.
U1-7,9:5,21-23%D	Unlocks setpoints 1 through 7 and 21 through 23. Setpoint 9 will deactivate after a 5 second delay.

The current value of a setpoint's deactivation delay timer can be accessed via 76P and 77P. See *Setpoint Timers* beginning on page 7-24 for more details.

Variable values can also be used to specify a setpoint number or deactivation delay. This technique is useful when recalling setpoint numbers from a database to serve as valve numbers, mixer numbers, etc. This allows you to write one routine to handle a fill routine for multiple ingredients.

1P%D	Deactivates the setpoint specified by the value of VAR#1.
5P:6P%D	Deactivates the setpoint specified by the value of VAR#5 using the delay specified by the value of VAR#6.

Deactivate All Setpoint

`* [ L | U ] %D`

Immediately deactivates all setpoints. No delay time can be specified. Delays in progress are cancelled.

***%D**	Deactivates all setpoints immediately, overriding all delays. No macros are invoked as a direct result of this command.
***L%D**	Deactivate and lock all setpoints immediately without invoking macros.

***U%D** Unlock all setpoints without changing states. No macros are invoked as a direct result of this command.

%E END IF

Syntax **End If**
%E

See Also
%N Else (If Not)
%{ Start Group
%} End Group
Boolean Logic

End If

Every *IF* command should be terminated with an *END IF*.

Serves as the termination point for all comparison (*if*) statements. If the condition of the comparison is false, macro execution will skip ahead to the next **%N** or **%E** command, whichever occurs first. Every comparison command should be terminated with a **%E** command.

%F IF SETPOINT DEACTIVATED

Syntax **If Setpoint Deactivated**
`< setpoint# > %F`

If Setpoint Queued for Deactivation
`. < setpoint# > %F`

Arguments

setpoint# Setpoint (0 → 256) to check for deactivation.

Notes The %F command applies to all setpoint configurations.

Setpoint ‘0’ (zero) checks for the program jumper in the ‘NO’ position (yields a true condition).

See Also
%D Deactivate Setpoint
%O If Setpoint Activated

Example:
Using the %F Command to Toggle a Flag

This routine uses setpoint #201 to toggle a display flag every time **[F1]** is pressed.

```
=================================
5099%s201%e          SPT #201
5100%s0%e            SPTyp Disbl
5101%sSYSTEM%e       SPNam SYSTEM

MACRO #1 – TOGGLE SYSTEM
201%F             if SYSTEM flag not set…
  201%A             set SYSTEM flag
  SYSTEM #1P4,1%C
%N                else
  201%D             clear SYSTEM flag
  SYSTEM #2P4,1%C
%E                end if

MACRO #6 – SYSTEM START
201%F             if SYSTEM flag not set…
  10%^              go to macro 10
%N                else
  11%^                go to macro 11
%E                end if
```

If Setpoint Deactivated

`< setpoint# > %F`

Determines if a setpoint input or output is deactivated. The example – *Using the %F Command to Toggle a Flag* shows how to use the **%F** command to toggle a setpoint used as a prompting flag.

If Setpoint Queued for Deactivation

`. < setpoint# > %F`

The *if setpoint queued for deactivation* command is a conditional statement that determines if a setpoint output has a deactivation delay in effect.

.5%F Determines if setpoint 5 is queued for deactivation.

%G GET ENTRY

Syntax **Get Entry**

`[ prompt ] [ ,* ] %G`

Arguments

prompt Text to be displayed as an entry prompt on the 2X5 character matrix of the 7-segment VFD.

* Entered characters will be displayed as an asterisks '*'.

Notes The *prompt* should be limited to 10 characters. If more than 10 characters are specified, only the last 10 will be used for the *prompt*.

The first 5 characters of the *prompt* will be displayed on the top line of the 2X5 matrix, the last 5 characters on the bottom line.

See Also

%K Get Entry from 4x20 VFD
%n Get Numeric Entry
%\ If No Entry
%[Save Entry Buffer
%o Math Assignment

Get Entry

Accepts alphanumeric user input. When this command is executed, the macro is suspended until the entry is completed by pressing **[ENTER]**. The optional *prompt* will be displayed until the first entry character is received. The **%G** command will accept an entry from the front panel keypad or any enabled comm port. Up to 79 characters can be entered and will remain in the entry buffer after **[ENTER]** is pressed (or a carriage return <CR> is received on one of the comm ports) requiring the next macro command to retrieve and/or store the entry. An entry in process can be cleared by pressing **[CLR]**. This clears the entire entry from the entry buffer, displays the optional *prompt*, and restarts the entry process.

Use of the asterisks '*' argument will cause each entered character to appear as an asterisks on the display. This provides a means of adding security to a user entry such as a password. Note that the asterisks

Example:
Qualifying an Operator Entry

This routine requires a valid entry before allowing it to be stored in VAR#2. The operator will be prompted to repeat the entry if no entry was made or if the entry was beyond the acceptable range.

```
==================================
80.2P=0%o            clear target value

%T                   tag position
EnterTargt%G         get entry

%\                   if no entry…
  Must Enter%P         prompt
  %J                   jump to tag
%E                   end if

%[                   save entry
%]>1000%o            if entry > 1000
%|                   OR
%]<0%o               if entry < 0…
  OutOfRange%P         prompt
  %J                   jump to tag
%E                   end if

80.2P=%]%o           store entry (VAR#2)
```

characters only appear during the entry. When **[ENTER]** is pressed to complete the entry, the entered characters are put into the entry buffer and become visible on the display. Be sure to copy the entry to a variable or other parameter immediately after the entry is complete to prevent this from happening.

Alphanumeric characters may be entered during the **%G** command using the front panel as described in the *Key-In Value Parameters* section on page 3-8.

Unexpected results may occur when entries contain both number and alpha characters, especially when including math symbols. Refer to the section on string assignments on page 9-117 for information on how to handle these special cases.

KeyInTargt%G =80.11P%o	Prompts for target entry and stores entry in VAR#11.
11%i %G %e	Simulates a manual entry into VAR#11. VAR#11 is selected as the current mode using the variable's name as the prompt. Use this method when entering time/date values or scale-specific float values for selectable units.
EnterTank#%G %A	Prompts for a tank number, expecting a valid output setpoint number for activation.
TruckID# ?%G %[	Prompts for a truck ID# and moves the entry from the entry buffer to the temporary buffer.
EnterTare?%G %t	Prompts for a tare entry and uses the entry as manual tare value.
EnterCode?,*%G	Prompts for a code entry and displays an asterisks '*' symbol in place of each character.

Example:
Using the %H Command to Redefine Keys

The **[SETUP]** key invokes macro 8 which displays the main setup menu and reassigns keys to invoke macro 9. Macro 9 will display a new menu depending on which key was pressed, **[F1]**, **[F2]** or **[F3]**. Each new menu reassigns keys to yet another macro intended to perform the functions of the sub-menu. The **[F4]** key is used to exit the menu. Other keys are ignored.

```
==================================
MACRO #8 – SETUP KEY
[F1] Set TargetsP1,1a%C
[F2] Set TimersP2,1%C
[F3] Set CountersP3,1%C
[F4] Exit MenuP4,1%C
0,9%H

MACRO #9 – SETUP MENU SELECTION
128P0C%(            if [F1] key pressed…
  [F1] Final TargetP1,1a%C
  [F2] Slow FillP2,1%C
  [F3] Pre-ActP3,1%C
  [F4] Exit MenuP4,1%C
  0,10%H
%N                  else
--------------------------------
129P0C%(            if [F2] key pressed…
  [F1] Mix TimerP1,1a%C
  [F2] Surge TimerP2,1%C
  [F3] Pump TimerP3,1%C
  [F4] Exit MenuP4,1%C
  0,20%H
%N                  else
--------------------------------
130P0C%(            if [F3] key pressed…
  [F1] Batch CountP1,1a%C
  [F2] Master CountP2,1%C
  [F3] Ticket No.P3,1%C
  [F4] Exit MenuP4,1%C
  0,30%H
%N                  else
--------------------------------
131P0C%(            if [F4] pressed…
  Pa%C                clear display
  X%d                 auto-update
  0,0%H               reset keypad normal
%N                  else
--------------------------------
  %)                  clear keypress
%E                  end if
```

%H REDEFINE COMM PORT FUNCTION

Syntax

Redefine Comm Port Function

`< comm > , < macro# | receive mode > %H`

Redefine Comm Port Function For DSD

`591, < comm > %H`

Arguments

comm	Communication port (0 → 4; port 0 = front panel keypad or disabled in the case of DSD).
macro#	Macro number (0 → 250 for port 0; 4 → 250 for ports 1 → 4) to be invoked upon receiving a character from specified port.
receive mode	Receive mode for ports 1 → 4 corresponding to selections for P205 of the setup mode:

0 Disable port receive
1 Set port receive standard
2 Set port receive for input interpreter
3 Set port receive for Modbus

Notes

A port will remain redefined as specified until changed again by another %H command.

The *receive mode* argument does not change the setup mode selection at P205, rather it temporarily changes the receive mode function until power is interrupted or upon saving changes when exiting the setup mode.

When a macro is invoked, the received character remains in the receive buffer. Thus the macro can test the port with the %(command and identify the character that invoked the macro.

When specifying port 0, keys on front panel keypad will no longer perform standard functions automatically.

Specify macro# 0 for port 0 to restore standard front panel keypad functions.

See Also

%(If Character Received
%) Clear Receive Buffer
%T Tag Position
%! Enable/Disable Comm Port

Redefine Comm Port Function

< comm > , < macro# | receive mode > %H

Temporarily changes a comm port's receive function (disabled, enabled, input interpreter or Modbus) or it can specify a macro to be invoked when a character appears in the receive buffer. When used in conjunction with the front panel keypad to invoke a macro, the **%H** command allows you to customize the function of every key. Refer to the example – *Using the %(Command to Get Entry* command on page 9-37 for additional information.

The **%H** command cannot be used to interrupt a macro. Characters received during macro execution are buffered and will invoke a specified macro once the macro stack is cleared. The **%H** command takes precedence over keypad macro assignments at P800 → P820.

Macro entry commands that require a keypress (%G, %W, %Y, etc.) revert to normal keypad operation while the entry command is in effect. A subsequent keypress resumes the function set forth by the last **%H** command.

A character or keypress received while the **%H** command is in effect will remain in the receive buffer until cleared. Failure to clear a received character will result in the macro being invoked in an endless loop, thus locking up the system. Should this happen, press **[CLR] + [SELECT]** to invoke the macro abort menu. Select the "suspend macro" option to stop macro execution. Remember to resume execution after correcting the problem by pressing **[CLR] + [SELECT]** and selecting the “resume macro” option.

1,0%H	Disables receive on comm 1.
1,1%H	Enables the standard receive mode on comm 1.
1,2%H	Enables the input interpreter receive mode on comm 1.
1,3%H	Enables the Modbus receive mode on comm 1.
1,4%H	Invokes macro 4 when a character is received on comm 1.
1,10%H	Invokes macro 10 when a character is received on comm 1.
2,0%H	Disables receive on comm 2.
2,1%H	Enables the standard receive mode on comm 2.
2,2%H	Enables the input interpreter receive mode on comm 2.
2,3%H	Enables the Modbus receive mode on comm 2.
2,4%H	Invokes macro 4 when a character is received on comm 2.
0,1%H	Invokes macro 1 when a front panel key is pressed.
0,2%H	Invokes macro 2 when a front panel key is pressed.
0,0%H	Restores the front panel keypad to normal operation.

Redefine Comm Port Function For DSD

591 , < comm > %H

Changes a comm port’s receive function for use with the Data Storage Device (DSD) feature (see page 6-13). It allow overriding the DSD port selection at P591. This change is temporary, and will be lost on power-up or if the setup mode is entered and saved.

The DSD function temporarily overrides whatever other receive function was setup for use of the port (P205). When the DSD functionality is moved to another port, the previous behavior is restored.

Note that if a comm port is programmed as receive disabled at P205, then it will not be possible to use the %H macro command to turn the comm port on. No data will be received.

Use of this macro command to change the operation of a comm port that is in use by DSD does not take control away from DSD. If DSD is then moved to another port, this previous selection would then begin operation.

%I REFRESH DISPLAY

Syntax **Refresh Display**

`%I`

Example:
Updating a Displayed Variable Value

%T	tag position
80.4P<100%o	if VAR#4 < 100…
80.4P+=1%o	increment
%I	refresh display
%J	jump to tag
%E	end if

Refresh Display

Updates a displayed value during macro execution. If the display is not refreshed, the displayed value will not change until macro execution has ended.

%J JUMP TO TAG

Syntax **Jump to Tag**

`[ tag# ] %J`

Jump to Tag (Macro Independent)

`@ < text > [ , macro# ] %J`

Example:
Using a Jump-to-Tag to Copy a Database

5,1%y	get first row in dbase#1
1%T	tag position #1
4%_	if row not found…
%N	else
3,2%y	make row in dbase#2
6,1%y	next row in dbase#1
1%J	jump to tag #1
%E	end if

Arguments

tag# Tagged position (0 → 99) to jump to.

text Alphanumeric tag identifier.

macro# Macro number (1 → 250) to search for specified tag.

Notes Omitting *tag#* is the equivalent of specifying a tag position of 0.

Omit *macro#* to search for the specified tag within the same macro.

See Also **%T Tag Position**

Example:
Jumping to Resume After a Power Failure

If VAR#5 contains a saved resume location at power-up, macro 250 will jump to that location. The resume location can be changed in other macros by assigning a different tag in VAR#5.

================================	
MACRO #6 – START FILL	
80.3P+=1%o	increment batch#
%t	tare
@RESUME%T	tag location
80.5P="@RESUME,6"%o	save resume location
1%A	activate fill output
Filling...P4,1c%C	prompt
MACRO #250 – POWER-UP	
80.5P!=""%o	if resume saved…
80.5P%o%J	jump to resume
%E	end if

Jump to Tag

`[ tag# ] %J`

Jumps backwards in a macro to a previously tagged position. Jumping is most commonly performed after a conditional statement that determines whether or not a particular routine should be repeated. Both jump and tag must occur within the same macro. Each tag within a macro should be unique, however individual jump commands can be used as often as necessary. It is not possible to jump to a tagged location that has not been executed within the macro. Even though a tag may be positioned before a corresponding jump, the jump will be invalid if the tag was skipped due to a branching command.

0%J Jumps backward to the last **0%T** or **%T** tag.

10%J Jumps backward to the last **10%T** tag.

Jump to Tag (Macro Independent)

`@ < text > [ , macro# ] %J`

Jumps backwards or forwards to a tag in the same macro or to a tag in another macro. The tag identifier can consist of up to 79 alphanumeric characters, but must not include a comma (,). Each tag within a macro should be unique, however individual jump commands can be used as often as necessary. When a macro independent jump is performed, the jump function begins searching for an '@' character from the beginning of the specified macro. When it encounters this character, it proceeds to compare the tag identifier with the jump identifier. When an exact match is found, macro execution resumes with the command following the tag. This tag search routine allows jumping to a tagged location that has not been executed within a macro.

@START%J	Searches the current macro for a **@START%T** command and resumes execution with the following command.
@FILL,10%J	Searches macro 10 for a **@FILL%T** command and resumes execution with the following command.

%K GET ENTRY FROM 4X20 VFD

Syntax **Get Entry from 4X20 VFD**

<*row*,*column*,*window*>[,*max entry*][n][b][u][g][*]%K

Example:
Simple 4X20 VFD Entry Screen

This entry routine displays a complete entry form before blinking the cursor in the first entry field. When **[ENTER]** is pressed to complete the part number entry, the part number is stored in VAR#12. The cursor will then blink in the tare weight entry field awaiting another entry. Note the use of square brackets [ ] to help identify the location and size of the entry window.

```
=================================
PACKAGE INFORMATIONP1,1a%C
Part Number [      ]P3,1%C
Tare Wt.    [      ]P4,1%C
3,14,6,6%K
=80.12P%o
4,14,6,6n%K
=80.13P%o
```

```
PACKAGE  INFORMATION

Part Number [?     ]
Tare Wt.    [      ]
```

Arguments

n	Allow only numeric entry characters (0 → 9, '. ', ' -'. ' +').
b	Do not blink cursor.
u	Display underscore '_' in place of cursor.
g	Defines the 4X20 VFD entry window as the new entry buffer. Operator input is not expected and macro execution continues, however any keypress or received character that would normally be displayed on the 2X5 matrix of the 7-segment VFD will instead be displayed in the entry window of the 4X20 VFD. The entry window will persist, even while in the setup mode, until cancelled by a %K command issued without arguments.
*	Entered characters will be displayed as an asterisks '*'.
row	Row (1 → 4) to position beginning entry position.
column	Column (1 → 20) to position beginning entry position.
window	Entry window size (1 → 79) in terms of characters.
max entry	Maximum number of characters (1 → 79) to be entered.

Notes

Omitting *max entry* assumes a maximum entry of 79 characters.

An entry error occurs if *window* exceeds the number of character locations from the cursor origin to the end of the display.

See Also

%G Get Entry
%n Get Numeric Entry
%\ If No Entry
%[Save Entry Buffer
%o Math Assignment

Get Entry from 4X20 VFD

Formats operator entries using the 4X20 VFD. An entry "window" can be defined by specifying the beginning coordinates, window size and maximum entry length. The entry window will overwrite any underlying text with spaces. By default, the cursor will blink at the leftmost location of the entry window. With each character entered, the cursor will shift right one position until it reaches the end of the window. If the entry length exceeds the window size, the cursor will remain in the rightmost location and characters in the window will shift left as additional characters are appended to the entry. Additional characters may not be entered once the maximum entry length is reached.

When the **%K** command is executed, the macro is suspended until the entry is completed by pressing **[ENTER]**. Any prompting should be displayed prior to the **%K** command. Entries will be accepted from the front panel keypad or any enabled comm port. Entered characters will remain in the entry buffer until **[ENTER]** is pressed (or a carriage return <CR> is received on one of the comm ports) allowing the next macro command to retrieve and/or store the entry. An entry in process can be cleared by pressing **[CLR]**. This clears the entire entry from the entry buffer and restarts the entry process.

Use of the asterisks '*' argument will cause each entered character to appear as an asterisks on the display. This provides a means of adding security to a user entry such as a password. Note that the asterisks characters only appear during the entry. When **[ENTER]** is pressed to complete the entry, the entered characters are put into the entry buffer and become visible on the display. Be sure to copy the entry to a variable or other parameter immediately after the entry is complete to prevent this from happening.

Alphanumeric characters may be entered during the **%K** command using the front panel as described in the *Key-In Value Parameters* section on page 3-8.

4,10,6,6%K Creates a 6-character entry window beginning at row 4, column 10. A maximum of 6 characters can be entered.

1,17,4%K Creates a 4-character entry window beginning at row 1, column 17. A maximum of 79 characters can be entered.

Example:
Displaying Values Before New Entry.

A break-tag is used to stop macro #1 execution until the first entry character is pressed. The %H command then redirects the keypress to macro #10 which determines if the character is alphanumeric. If so, the character is passed back to macro #1 at the break-tag resume location (note that the character is not cleared from the receive buffer in macro #10). The %K command then completes the entry process and continues with the next entry field.

```
MACRO #1 – CUSTOM DATA ENTRY
CUSTOMER INFORMATIONP1,1a%C
Customer Name:P3,1%C
\025\153p4,1%C
80.2.16403P%oP4,2%C
\021p4,2%C        blink cursor
0,10%H            keypad invokes macro 10
4,10%H            alpha key invokes macro 10
1B%T              break-tag #1
4,2,19,19%K       get entry
%\                if no entry…
%N                else
  =80.2P%o          save entry in VAR#2
%E                end if
--------------------------------
Customer Address:P3,1%C
\025\153p4,1%C
80.3.16403P%oP4,2%C
\021p4,2%C        blink cursor
0,10%H            keypad invokes macro 10
4,10%H            alpha key invokes macro 10
1B%T              break-tag #1
4,2,19,19%K       get entry
%\                if not entry…
%N                else
  =80.3P%o          save entry in VAR#3
%E                end if

MACRO #10 – GET FIRST KEYPRESS
32,126P04%(       if alphanumeric keypress
%|                OR
229P04%(          if [ENTER] pressed…
  0,0%H             normal keypad operation
  4,1%H             normal alpha operation
  1%J               jump to break-tag
%N                else
--------------------------------
227P0C%(          if [CLR] pressed…
  0,0%H             normal keypad operation
  4,1%H             normal alpha operation
  20%^              go to macro 20
%N                else
--------------------------------
  %)                clear keypress
%E                end if
```

3,1,6,6n%K Creates a 6-digit (numeric-only) entry window beginning at row 3, column 1. 6 digits can be entered.

3,1,6,6nb%K Same as above but without a blinking cursor.

3,1,20g%K Relocates the entry buffer from the 7-segment VFD to row 3, column 1of the 4X20 VFD. The entry buffer size is increased to 20 characters.

3,1,20gb*%K Same as above but without a blinking cursor. Entered characters to be displayed an asterisks ‘*’.

%K Cancels the 4X20 VFD entry buffer and restores the entry buffer to the 7-segment VFD.

When prompting for an operator entry, it is often desirable to display the current value of an entry field. Although the **%K** command cannot pre-load a value within the entry window, it is possible to display a formatted value in the location of the entry window and wait for the first entry keypress *before* executing the **%K** command. The example – *Displaying Values Before New Entry* demonstrates an advanced entry technique using the **%H** command to pass the first entered character to the **%K** entry window. Because the **%K** command is not in effect when the entry begins, the parameter’s value can be displayed without being overwritten by the entry window. Note that macro 10 makes it possible to check for other keys pressed at the beginning of the entry. In this example, pressing **[CLR]** without an entry in process will abort the entire entry routine.

```
CUSTOMER INFORMATION

Customer Name:
™ACME Products
```

G

```
CUSTOMER INFORMATION

Customer Name:
™GØ
```

eneral Products

```
CUSTOMER INFORMATION

Customer Name:
™General ProductsØ
```

ENTER YES

%K GET ENTRY FROM 8X40 AND 16X40 LCD

Syntax

Get Entry from 8X40 and 16X40 LCD

<*row*,*column*,*window*>[,*max entry*][n][u][g][*][,F|f *size*]%K

Arguments

F — Normal entry text (black on white)

f — Inverse entry text (white on black)

n — Allow only numeric entry characters (0 → 9, '. ', ' -'. ' +').

u — Display underscore '_' in place of cursor.

g — Defines the 4X20 VFD entry window as the new entry buffer. Operator input is not expected and macro execution continues, however any keypress or received character that would normally be displayed on the 2X5 matrix of the 7-segment VFD will instead be displayed in the entry window of the 4X20 VFD. The entry window will persist, even while in the setup mode, until cancelled by a %K command issued without arguments.

* — Entered characters will be displayed as an asterisks '*'.

row — Row (1 → 16) to position beginning entry position.

column — Column (1 → 40) to position beginning entry position.

window — Entry window size (1 → 79) in terms of characters.

max entry — Maximum number of characters (1 → 79) to be entered.

size — Select one of the following font size options:

- 1 Small font size (H = 1 line, W = 1 column)
- 2 Medium font size (H = 2 lines, W = 2 columns)
- 4 Large font size (H = 4 lines, W = 4 columns)

Notes

Omitting *max entry* assumes a maximum entry of 79 characters.

An entry error occurs if *window* exceeds the number of character locations from the cursor origin to the end of the display.

Once a font *size* is selected, that size remains in effect until changed.

Addressing the LCD does not utilize the transmit port of comm 4 as with the 4X20 VFD.

See Also

%G Get Entry

%n Get Numeric Entry

%\ If No Entry

%[Save Entry Buffer

%o Math Assignment

Get Entry from 8X40 and 16X40 LCD

Formats operator entries using the 8X40 and 16X40 LCD. An entry “window” can be defined by specifying the beginning coordinates, window size and maximum entry length. The entry window will overwrite any underlying text with spaces. By default, the cursor will be positioned at the leftmost location of the entry window. With each character entered, the cursor will shift right one position until it reaches the end of the window. If the entry length exceeds the window size, the cursor will remain in the rightmost location and characters in the window will shift left as additional characters are appended to the entry. Additional characters may not be entered once the maximum entry length is reached.

When the **%K** command is executed, the macro is suspended until the entry is completed by pressing **[ENTER]**. Any prompting should be displayed prior to the **%K** command. Entries will be accepted from the front panel keypad or any enabled comm port. Entered characters will remain in the entry buffer until **[ENTER]** is pressed (or a carriage return <CR> is received on one of the comm ports) allowing the next macro command to retrieve and/or store the entry. An entry in process can be cleared by pressing **[CLR]**. This clears the entire entry from the entry buffer and restarts the entry process.

Use of the asterisks ‘*’ argument will cause each entered character to appear as an asterisks on the display. This provides a means of adding security to a user entry such as a password. Note that the asterisks characters only appear during the entry. When **[ENTER]** is pressed to complete the entry, the entered characters are put into the entry buffer and become visible on the display. Be sure to copy the entry to a variable or other parameter immediately after the entry is complete to prevent this from happening.

Alphanumeric characters may be entered during the **%K** command using the front panel as described in the *Key-In Value Parameters* section on page 3-8.

4,10,6,6,F1%K	Creates a 6-character, small font size entry window beginning at row 4, column 10. A maximum of 6 characters can be entered.
1,17,4,8,F2%K	Creates a 4-character, medium font size entry window beginning at row 1, column 17. A maximum of 8 characters can be entered.
1,17,4%K	Creates a 4-character entry window beginning at row 1, column 17. A maximum of 79 characters can be entered. The previous font size remains in effect.
7,1,6,6n,f2%K	Creates a 6-digit (numeric-only), medium inverse font size entry window beginning at row 3, column 1. 6 digits can be entered.
5,1,20g%K	Relocates the entry buffer from the LCD auto-update to row 5, column 1of the 4X20 VFD. The entry buffer size is increased to 20 characters.

%K	Cancels the relocated LCD entry buffer and restores the entry buffer to the LCD auto-update.

%K *IF ENTRY TERMINATED BY FUNCTION KEY*

Syntax **If Entry Terminated by Function Key**

`? < key > %K`

Arguments

key Select one of the following function keys:

0	**[ENTER]**	
2	**[F2]**	(660 Series only)
6	**[START]**	
7	**[STOP]**	
8	**[SETUP]**	

See Also **%G Get Entry**

%n Get Numeric Entry

If Entry Terminated by Function Key

Determines which *key* was used to terminate an entry from a %G, %K or %n command.

Pass-word?%G **?7%K** **Abort%P** **%N** **=80.1P%o** **%E**	Prompts for a password, then determines if the **[STOP]** key terminated the entry. If **[STOP]** was pressed, then "*Abort*" is displayed. Otherwise the entry is stored in VAR#1.

%L *LANGUAGE SELECTION*

Syntax **Get Current Language**

`%L`

Set Language

`< language# > %L`

Arguments

language# Select one of the following language numbers:

0	USA
1	France
2	German
3	UK
4	Denmark
5	Sweden
6	Italy
7	Spain
8	Japan
9	Norway
10	Denmark 2
11	Spain 2
12	Latin America

Notes The *language#* argument does not change the setup mode selection at P411, rather it temporarily changes the language until power is interrupted or upon saving changes when exiting the setup mode.

See Also
%[Save Entry Buffer
%o Math Assignment
Table 6.2: *International Characters*

Get Current Language

%L

Copies the current language number to the entry buffer.

%L	Copies the current language number to the entry buffer.
%L=80.11P%o	Saves the current language number in VAR #11.
%L%[	Saves the current language number in the temporary buffer.

Set Language

< *language#* > %L

Overrides the power-up language selection at P411.

12%L	Selects the Latin American character set.
0%L	Selects the USA character set.
80.11P%o%L	Selects the character set as determined by the value of VAR#11.

%M MODE SELECTION

Syntax

Get Current Mode

`%M`

If Current Mode

`< mode > %M`

Arguments

mode — Operating parameter with the syntax:

`< parm > . [ instance ]`

where *parm* is a operating parameter with a valid *instance*.

Notes

The *instance* argument is required when specifying parameter 50, 51, 52, 80, 81 or 82.

See Also

%[	**Save Entry Buffer**
%o	**Math Assignment**
%s	**Select Mode**

Get Current Mode

`%M`

Copies the current mode to the entry buffer. Example 9-36 shows how to use this command to save and restore the operating mode. This is useful when getting entries such as time/date or scale-specific variables where the mode must be temporarily changed in order to accept the entry in the proper format. This technique could also be used to restore a mode upon power-up if an auto-save variable is used to save and restore the mode.

%M	Copies the currently displayed parameter number to the entry buffer.
%M=80.3P%o	Saves the currently displayed parameter number in VAR #3.
%M%[	Saves the currently displayed parameter number in the temporary buffer.

Example:
Saving and Restoring the Current Mode

If VAR#5 is configured as a scale-specific float, a macro must simulate a manual entry rather than performing a direct assignment. This is the only way to ensure the value is entered in terms of the current units. Simulating a manual entry requires the mode to be changed to the entry parameter. This routine restores the previous parameter selection, thus making the mode change transparent to the operator.

```
================================
%M%[               save current mode
5%i                select VAR#5
EnterTargt%G       get target weight entry
%e                 enter target weight
%]%s               restore saved mode
```

If Current Mode

`< mode > %M`

Determines if the specified *mode* argument is the currently displayed parameter number.

0%M	Determines if the current mode is *gross*.
1%M	Determines if the current mode is *net*.
1.2%M	Determines if the current mode is *net* of *scale 2*.

Example:
Simulating the Mode Selections at P300.

Pressing **[F1]** will scroll through the operating parameters included in macro 1, similar to how the **[SELECT]** key scrolls through the modes assigned at P300 → P309. Note that this routine does not limit the number of selectable parameters. Macro 4 could also be programmed to scroll in reverse order via the **[F4]** key.

```
===============================
0%M                if gross mode…
  1%s                select net mode
%N                 else
1%M                if net mode…
  2%s                select tare mode
%N                 else
2%M                if tare mode…
  0%s                select gross mode
%E                 end if
```

80.5%M	Determines if the current mode is *variable 5*.

%N ***ELSE (IF NOT)***

Syntax **Else (If Not)**
%N

See Also
%E End If
%{ Start Group
%} End Group
Boolean Logic

Else (If Not)

Serves as the point where macro execution will resume after determining a comparison (*if*) statement to be false. Macro commands between the comparison (*if*) statement and the **%N** command will not be executed.

%O ***IF SETPOINT ACTIVATED***

Syntax **If Setpoint Activated**
`< setpoint# > %O`

If Setpoint Queued for Activation
`. < setpoint# > %O`

Arguments

setpoint# Setpoint (0 → 256) to check for activation.

Notes The %O command applies to all setpoint configurations.

Setpoint '0' (zero) checks for the program jumper in the 'YES' position (yields a true condition).

See Also
%A Activate Setpoint
%F If Setpoint Deactivated

Example:
Assigning One Macro for Setpoint Activation and Deactivation.

Setpoint input 10 invokes macro 100 when activated and when deactivated. The %O command allows the macro to branch to the appropriate routine.

```
================================
5099%s10%e          SPT #10
5100%s2%e           SPTyp Input
5101%sLEVEL%e       SPNam LEVEL
5112%s100%e         AcMac   100
5132%s100%e         DeMac   100

MACRO #100 – START/STOP MIXER
10%O                    if input active…
  Valve OpenP4,1c%C       display text
  1%D                     stop mixer
%N                      else
  Valve ShutP4,1c%C       display text
  1%A                     start mixer
%E                      end if
```

If Setpoint Activated

`< setpoint# > %O`

Determines if a setpoint input or output is active. The example – *Assigning One Macro for Setpoint Activation and Deactivation* shows how the **%O** command can be used to allow a single macro to handle both the activation and deactivation condition of a setpoint input. This helps simplify program development by reducing the number of macros required and by making the setup easier to follow.

10%O	Determines if setpoint 10 is activated.

If Setpoint Queued for Activation

`. < setpoint# > %O`

Determines if a setpoint output has an activation delay in effect.

.10%O	Determines if setpoint 10 is queued for activation.

%P PAUSE

Syntax **Pause**
`[ prompt ] %P`

Arguments

prompt Text to be displayed as a prompt on the 2X5 character matrix of the 7-segment VFD.

See Also **%@ Set Pause Time**

Pause

Suspends macro execution while displaying an optional prompt. The default pause time set at power-up is one second. The **%@** command can be used to change the pause time from 0.01 → 5,000,000 seconds.

%P	Suspends macro execution for 1 second.
BatchDone!%P	Suspends macro execution for 1 second while displaying **Batch Done!**.
5%@ **BatchDone!%P** **1%@**	Suspends macro execution for 5 seconds while displaying **Batch Done!**. The default 1 second pause time is restored.

%Q SEND CUSTOM TRANSMIT

Syntax **Send Custom Transmit**

Example:
Toggling Continuous Transmits.
Pressing **[SELECT]** toggles between GROSS and NET for 2 scales on the 4X20 display. Pressing **[PRINT]** will uses the same custom transmits to redirect data out comm port 1. Note how the <FF> serves a dual purpose, advancing the printed form and positioning the cursor.

```
================================
425%s0%e             4x20  Disbl
800%s24%e            Selct Mc 24
```

`< transmit# > [ . comm ] %Q`

Enable / Disable Continuous Transmit
`< transmit# > [ . comm ] [C | D] %Q`

Set Continuous Transmit Interval
`< seconds > I%Q`

Set Continuous Transmit Interval to Display Rate
`< transmit# > [ . comm ] X%Q`

If Custom Transmit Continuous
`< transmit# > [ . comm ] ?%Q`

Arguments

C	Send custom transmit continuos.
D	Cancel continuous custom transmit.
X	Send custom transmit at display rate while in net or gross mode.
transmit#	Custom transmit (1 → 250) to send.
comm	Communication port (1 → 4).
seconds	Number of seconds (0.01 → 2,883,584) to delay between continuous transmits.

Notes

The *seconds* argument does not change the setup mode selection at P980, rather it temporarily changes the continuous transmit interval until power is interrupted or upon saving changes when exiting the setup mode.

Only one custom transmit is allowed to be transmitted at the display rate any new #X%Q will replace the previous. 0X%Q or X%Q will stop this custom transmit.

See Also **%p Print**

Send Custom Transmit

`< transmit# > [ . comm ] %Q`

Initiates the transmission of a specified custom transmit. The **%Q** command will send a custom transmit regardless of the send criteria at P991 (even if set to 'Off'). The transmission will occur out the comm port specified at P992 unless a different port is specified with the *comm* argument. Motion delay criteria at P993 → P997 will be enforced and macro will be suspended for the duration of any motion delay.

1%Q	Sends custom transmit 1 out the comm port specified at P992.
2.3%Q	Sends custom transmit 2 out comm 3.

Enable / Disable Continuous Transmit

`< transmit# > [ . comm ] [C | D] %Q`

Allows any custom transmit to be transmitted on a continuous basis. Enabling or disabling a continuous transmit with the **%Q** command will override the continuous transmit selection at P998. The transmission will occur out the comm port specified at P992 unless a different port is specified with the *comm* argument. Motion delay criteria at P993 → P997 will be enforced but will not suspend macro execution during a motion delay.

As many as 16 custom transmits may be set for continuous transmission. An attempt to send more than 16 continuous transmits will be disregarded and result in a `Code72 ConTx >Max!` error message. Attempting to specify a continuous transmit that is already continuous at the specified comm port will result in a `Code75 Tx is Cont.` error message.

The interval at which the continuous transmit list will be sent is based on the interval specified at P980. This interval can be overridden with the **I%Q** command.

Continuous transmits are sent in the order in which they were added to the continuous transmit list. For example, if custom transmit 3 is specified as continuous prior to custom transmit 1, then custom transmit 3 will be sent before custom transmit 1 at the beginning of each custom transmit interval. A custom transmit set for continuous transmission that is set for motion delay at P993 → P997 will be skipped if the motion criteria exists at the time of the next custom transmit interval. Continuous transmits are not sent if the transmit buffer of the intended port is not empty. This prevents a backlog of transmission data. If the size of a transmission exceeds the transmit buffer size, the weight conversion process may become delayed by the transmission. Make sure the transmit buffer size at P207 is large enough to accommodate the largest transmission.

1C%Q	Sends custom transmit 1 continuously out the comm port specified at P992.
2.3C%Q	Sends custom transmit 2 out comm port 3 continuously.
2D%Q	Cancels the continuous transmission of custom transmit 2 out the comm port specified at P992.
2.3D%Q	Cancels the continuous transmission of custom transmit 2 out comm port 3.
D%Q	Cancels all continuous transmissions.

Set Continuous Transmit Interval to Display Rate

`< transmit# > [ . comm ] [X] %Q`

Allows any custom transmit to be transmitted on a continuous basis at the display update rate in when the scale is in gross or net weigh modes. Any other mode will stop the transmission until the gross or net weigh modes are displayed. Enabling or disabling a continuous transmits does not affect the continuous transmit display rate. The transmission will occur out the comm port specified at P992 unless a different port is specified with the *comm* argument. Motion delay criteria at P993 → P997 is **ignored** for continuous transmits at display rate.

Only one custom transmit can be sent at the display update rate. Sending a new custom transmit will cause the new transmit to replace the old one.

If one selects any weigh mode other than gross or net the display custom transmits will stop until you return. Entering setup mode will stop the custom transmit at display rate but it will startup at on exit provided the custom transmit still exists. The custom transmit at display rate will be lost on power off and on.

1X%Q	Sends custom transmit 1 at the display update rate out the comm port specified at P992.
2.3X%Q	Sends custom transmit 2 out comm port 3 at the display update rate and will replace any previously custom transmit specified by X%Q
X%Q	Cancels the continuous custom transmit at the display update rate.
0X%Q	Cancels the continuous custom transmit at the display update rate

If Custom Transmit Continuous

`<` *`transmit#`* `> [ .` *`comm`* `] ?%Q`

Determines if a custom transmit is set for continuous transmission.

1C?%Q	Determines if custom transmit 1 is set for continuous transmission out the comm port specified at P992.
2.3C?%Q	Determines if custom transmit 2 is set for continuous transmission out comm port 3.
?%Q	Determines if any custom transmits are set for continuous transmission.

Set Continuous Transmit Interval

`<` *`seconds`* `> I%Q`

Sets the number of seconds between each attempt to send all continuous transmits. This command overrides the transmit interval assigned by P980 at power-up.

2I%Q	Sets the continuous transmit interval to 2 seconds.

.2I%Q	Sets the continuous transmit interval to 0.2 seconds.
0I%Q	Suspends all continuous transmits.

%R RENAME MODE

Syntax **Rename Mode**

`< parm > , < name > %R`

Arguments

parm Operating parameter (*do not specify an instance*).

name New name to appear in place of the default parameter name on the 2X5 character matrix of the 7-segment VFD.

Notes Only operating parameters that appear in the setup mode at P600 → P646 can be renamed with the %R command.

A parameter cannot be renamed with the %R command unless it has first been renamed in the setup mode. The number of characters for *name* cannot exceed that of the parameter's given name in the setup mode.

Although the %R command does not change a parameter's given name in the setup mode, the new name is retained indefinitely if it is in effect when saving changes to the setup mode. The %R command must be used to restore the default name.

See Also **%u Units**

%s Select Mode

Example:
Using a Parameter's Name as a Prompt.

When **[START]** is pressed, the net mode is renamed to prompt "Fast" when the fast fill output is activated. When setpoint #1 deactivates, macro #101 is invoked to activate the slow fill output and prompt "Slow". When the fill is complete, setpoint #2 deactivates invoking macro #102 to prompt "Done!" and then restore the original parameter name.

```
================================
601%sNet  %e        Net   Net

MACRO #6 START FAST FILL
%t                tare
1%A               activate fast fill output
1,Fast%R          rename "Net"

MACRO #101 START SLOW FILL
2%A               activate slow fill output
1,Slow%R          rename "Net"

MACRO #102 FILL COMPLETE
1,Done!%R         rename "Net"
%S%P              sound beeper and
                  pause
1,Net%R           restore "Net" name
```

Rename Mode

Allows a parameter's displayed name to be changed. Once changed, the new name will be displayed every time the parameter is accessed. Renaming the mode in this manner allows you to display parameters in multiple languages or use a parameter's name to display a prompt without suspending macro execution.

0,Bruto%R	Renames "Gross" to display "Bruto" when the gross mode is selected.
1,Neto%R	Renames "Net" to display "Neto" when the net mode is selected.
2,Tara%R	Renames "Tare" to display "Tara" when the tare mode is selected.
1,Fast%R	Renames "Net" to display "Fast" when the net mode is selected.

Example:
Rename Mode Data Format

```
 10)00 kg
       Gross
```

0,Bruto%R

```
10)00  kg
       Bruto
```

Example:
Beeper Frequency Conversion Chart

Note frequencies are represented in Hertz (Hz).

C	33	65	130
D^b	35	69	139
D	38	73	147
E^b	39	78	156
E	41	82	165
F	44	87	175
Gb	46	92	185

1,Slow%R	Renames “Net” to display “Slow” when the net mode is selected.
1,Done!%R	Renames “Net” to display “Done!” when the net mode is selected.

The example – *Using a Parameter’s Name as a Prompt* demonstrates how to use the **%R** command to prompt the various cycles of a filling routine. Note that in this example P601 was renamed as “Net”, the original parameter name. This retains the parameter name while allocating memory for the **%R** command. Note also the extra two spaces appended to “Net” in P601. This reserves the full 5 characters for the **%R** command to use when prompting.

%S SOUND BEEPER

Syntax

Sound Beeper
%S

Program Beeper Sequence
`{[C][F frequency][D duration][V volume]%S`

Set Keypad Beeper Volume
`{K [ volume ] %S`

If Beeper Sequence Running
`{?%S`

Arguments

C	Cancel the execution of a programmed beeper sequence.
frequency	Frequency (10 → 10,000 Hz) of the beeper tone.
duration	Duration of the beeper tone in milliseconds.
volume	Beeper volume (0 → 7; 0 = silent, 7 = loudest).

Notes A sequence of tones may be specified as in the following examples.

Sound Beeper

%S

Produces a 0.5 second, 2 KHz tone through the internal beeper. Macro execution is not suspended while the beeper is running. A longer tone duration is possible using the program beeper sequence commands.

Program Beeper Sequence

`{ [C] [ F frequency ] [ D duration ] [ V volume ] %S`

Provides variation of the beeper's tone, duration and volume. A sequence of various tones can be queued for execution without suspending macro operation by including multiple commands in a comma-delimited list.

Once a frequency, duration or volume is specified, it is not necessary to include them in subsequent beeper commands if the previous value will remain the same. The *program beeper sequence* command does not affect characteristics of the keypad beeper.

{F2000D1000V7%S	Sounds the beeper at 2 KHz for 1 second at the loudest volume.
{F2000D50V7,V0,V7%S	Produces a fast "double-beep". The frequency and duration is maintained while the volume changes from maximum, to silent, back to maximum.
{F2000D100,F1600,F2000%S	Produces a "two-tone" warble by varying the frequency while maintaining duration and volume.
{C%S	Cancels all queued beeper sequences.

Set Keypad Beeper Volume

`{K [ volume ] %S`

Overrides the power-up keypad volume set at P460.

K0%S	Turns off the keypad beeper.
K1%S	Sets the minimum keypad beeper volume.
K7%S	Sets the maximum keypad beeper volume.

If Beeper Sequence Running

`{?%S`

Determines if a programmed beeper sequence is still running. This command could be used in a loop to suspend macro operation until a beeper sequence is completed.

%T TAG POSITION

Syntax **Tag Position**

`[ tag# ] %T`

Tag Resume Position

`[ tag# ] B%T`

Tag Position (Macro Independent)

`@ < text > %T`

Example:
Tagging a Location to Force an Entry

```
1%T              tag position #1
EnterID# ?%G     get entry
%\               if no entry…
  1%J              jump to tag #1
%E               end if

=80.1P%o         save entry in VAR#1
```

Example:
Tagging Menu Locations in One Macro

```
MACRO #250 – POWER-UP

@MAIN MENU%T
[F1] Set TargetsP1,1a%C
[F2] Set TimersP2,1%C
[F3] Set CountersP3,1%C
[F4] Exit MenuP4,1%C
0,9%H
%B

@TARGET MENU%T
[F1] Final TargetP1,1a%C
[F2] Slow FillP2,1%C
[F3] Pre-ActP3,1%C
[F4] Exit MenuP4,1%C
0,10%H
%B

@TIMER MENU%T
[F1] Mix TimerP1,1a%C
[F2] Surge TimerP2,1%C
[F3] Pump TimerP3,1%C
[F4] Exit MenuP4,1%C
0,20%H
%B

@COUNTER MENU%T
[F1] Batch CountP1,1a%C
[F2] Master CountP2,1%C
[F3] Ticket No.P3,1%C
[F4] Exit MenuP4,1%C
0,30%H
%B
```

Arguments

B	Stop macro execution and tag as a resume location.
tag#	Tag position (0 → 99).
text	Alphanumeric tag identifier.

Notes Omitting *tag#* is the equivalent of specifying a tag position of 0.

See Also

%J Jump To Tag

%H Redefine Comm Port Function

Tag Position

`[ tag# ] %T`

Marks a location within a macro that can be jumped back to using the *%J Jump To Tag* command. A tag must be executed before it can be jumped to. Therefore, it is not possible to jump forward to a tag using this method. Also, a tag skipped due to other branching commands will not be recognized. Avoid duplicating tag numbers within a macro to minimize confusion and eliminate potential branching errors.

%T	Tags a position that can be jumped to with a **%J** or **0%J** command.
10%T	Tags a position that can be jumped to with a **10%J** command.

Tag Resume Position

`[ tag# ] B%T`

Stops macro execution and marks a location within a macro that can be jumped back to using the *%J Jump To Tag* command. This command is used in conjunction with the **%H** command, allowing another macro to execute and later resume operation at the tagged location in the original macro. When the **B%T** command stops macro execution, the original macro and all calling macros are removed from the macro stack.

Unlike the standard *tag position* command, the *tag resume position* command is commonly used multiple times within one macro. This allows you to develop a common entry routine using the **%H** command where only one jump command is required to branch back to multiple tag locations. Refer to the example – *Displaying a Parameter's Value Prior to Entering a New Value* of the **%K** command on page 9-68 for a practical application using the **B%T** command.

Tag Position (Macro Independent)

`@ < text > [ , macro# ] %T`

Marks a location within a macro that can be jumped to from within the same or other macros. The tag identifier can consist of up to 79 alphanumeric characters, but must not include a comma (,). Each tag within a macro should be unique, however individual jump commands can be used as often as necessary. When a macro independent jump is performed, the jump function begins searching for an '@' character from the beginning of the specified macro. When it encounters this character, it proceeds to compare the tag identifier with the jump identifier. When an exact match is found, macro execution resumes with the command following the tag. This tag search routine allows jumping to a tagged location that has not been executed within a macro.

@START FILL%T	Tags a position that can be jumped to with an **@START FILL%J** command.
@RESUME FILL%T	Tags a position that can be jumped to with an **@RESUME FILL%J** command.
@MAIN MENU%T	Tags a position that can be jumped to with an **@MAIN MENU%J** command.

The example - *Tagging Menu Locations in One Macro* shows how one macro could be used to set up various menus and redirect operator interface to different macros. Each menu is identified by a descriptive tag location. Menus and operator interface routines are easy to identify as they are all contained in one macro. Additional menus are easily added. A break command ends each menu item to prevent macro execution from continuing to the next menu. This macro could be used as a power-up macro to automatically display the main menu selections.

%U TRANSMIT BUFFER

Syntax

If Transmit Buffer Empty
`%U`

Get Number of Characters in Transmit Buffer
`< comm > %U`

Clear Transmit Buffer
`< comm > * %U`

Arguments

comm Communication port (1 → 4).

Notes When used as an *if* condition, the %U command tests the port last specified by the %" command.

See Also **%" Select Comm Port**

Example:
Verifying a Successful Transmission

This routine sends custom transmit #1 to a printer connected to comm 2. Assuming the printer uses hardware handshaking, the transmission will not be completed if the printer is off. The transmitted characters will remain in the indicator's transmit buffer and generate an error prompt to make the operator aware of the problem. The number of characters remaining in the transmit buffer is stored in VAR#4 and the buffer is then cleared.
==================================

If Transmit Buffer Empty

%U

Determines if the comm port transmit buffer currently selected by the **%”** command is empty.

2%” **%U**	Determines if the transmit buffer on comm 2 is empty.

Get Number of Characters in Transmit Buffer

< *comm* > %U

Copies the number of characters remaining in the specified comm port transmit buffer to the entry buffer.

1%U	Copies the number of characters remaining in the comm port 1 transmit buffer to the entry buffer.

Clear Transmit Buffer

< *comm* > *%U

Clears all remaining characters from the specified comm port transmit buffer.

3*%U	Clears the comm port 3 transmit buffer.

%W WAIT FOR KEYPRESS

Syntax **Wait for Keypress**
[*prompt*] %W

Arguments

prompt Text to be displayed as a prompt on the 2X5 character matrix of the 7-segment VFD.

See Also **%Y If Yes (Enter)**

Wait for Keypress

Suspends macro execution until any front panel key is pressed or any character is received on one of the enabled comm ports. An optional *prompt* can be displayed while waiting for the keypress. The keypress or received character is immediately cleared from the receive buffer. This command can be used to require an operator’s acknowledgement before continuing a process. It is useful as a debugging aid, providing a means of “stepping” through a macro routine.

%W	Suspends macro execution and waits for a keypress or a received character before resuming with the next macro command.
BatchDone!%W	Suspends macro execution and prompts **Batch Done!** until a keypress or character is received.

Example:
Echo Display Data Format

```
 10)00 kg
       Gross
```

The display above would be transmitted as shown below when using the %X command.

```
===============================
<STX><NUL>kg   Gross<ETX>
<STX><LF> 100.00<ETX>
```

Display data may be sent continuously out a comm port specified at P290 in a format similar to the %X command.

%X REQUEST DISPLAY DATA

Syntax **Request Display Data**
%X

See Also **Echo Display**

Request Display Data

Echoes the 7-segment display in a format compatible with the remote display mode of the GSE M450 and M550 series indicators. Display data is transmitted out the comm port last selected by the *% Select Comm Port* command.

1%"%X	Echoes display data out comm port 1.
2%"%X	Echoes display data out comm port 2.
%X	Echoes display data out last selected comm port.

Display data is sent in the following format:

```
<STX> <NUL> <UPPER> <LOWER> <ETX> <STX> <LF> <7-SEGMENT> <ETX>
```

UPPER	is a fixed-width field containing the 5 characters of the upper row of the 2X5 display matrix.
LOWER	is a fixed-width field containing the 5 characters of the lower row of the 2X5 display matrix.
7-SEGMENT	is a variable-width field containing the 6 digits of the 7-segment display. Blank digits are sent as spaces. Each displayed decimal point is sent as a separate byte.

%Y IF YES (ENTER)

Syntax **If Transmit Buffer Empty**
`[ prompt ] %Y`

Arguments

prompt	Text to be displayed as a prompt on the 2X5 character matrix of the 7-segment VFD.

See Also **%W Wait for Keypress**

Example:
Prompting a YES / NO Question

```
1%T
Enter Item# [      ]P4,1%C
4,14,6,6%K
=80.6P%o
Add Another Item?P2,3a%C
[YES] / [NO]P4,5%C
%Y                       if YES…
  1%J                      jump to tag #1
%N                       else
  Pa%C                     clear display
%E                       end if
```

If Yes

Determines if the **[ENTER/YES]** key was pressed. An optional *prompt* can be displayed while waiting for the keypress. When a **%Y** command is encountered, macro execution is suspended until a front panel key is pressed or until a character is received on any enabled comm port. If the **[ENTER/YES]** key is press, or if a carriage return <CR> is received on a comm port, then the condition is true. Any other key or received character yields a false condition. The keypress or received character is immediately cleared from the receive buffer.

StartFill?%Y	Suspends macro execution and prompts **Start Fill?** until a keypress or character is received.

%[Save Entry Buffer

Syntax **Save Entry Buffer**
%[

See Also **%] Restore Entry Buffer**
Entry Buffer

Save Entry Buffer

Stores all data in the entry buffer in a temporary register . The entry buffer is then cleared. Contents of the temporary register can be restored to the entry buffer with the *%] Restore Entry Buffer* command. Restoring the entry buffer does not change the contents of the temporary register. Thus, the original contents of the entry buffer can be restored multiple times. The temporary register can only be cleared by issuing a **%[** command when the entry buffer is empty.

%[	Saves the entry buffer contents in a temporary register.
%c **%[**	Clears the entry buffer and the temporary register.
80.1P%o%[	Copies the value of VAR#1 to the entry buffer and saves it in the temporary register.
100%[	Copies the value 100 to the temporary register.

%\\ IF NO ENTRY

Syntax **If No Entry**
%\

See Also
%G Get Entry
%K Get Entry from 4X20 VFD
%n Get Numeric Entry
%] Restore Entry Buffer

Example:
Determining if an Entry Was Made

`1%T`	tag #1
`EnterID# ?%n`	get numeric entry
`%\`	if no entry…
`  Must!Enter%P`	prompt
`  1%J`	jump to tag #1
`%E`	end if

If No Entry

Determine if the entry buffer is empty. This command is commonly used following a "get entry" command to determine if an entry was made prior to pressing **[ENTER]**. It can also be used after a the *%] Restore Entry Buffer* command to determine if the temporary register is empty.

%] RESTORE ENTRY BUFFER

Syntax **Restore Entry Buffer**
%]

See Also
%[Save Entry Buffer
Entry Buffer

Restore Entry Buffer

The *restore entry buffer* command is used in conjunction with the *%[Save Entry Buffer* command, copying the contents of the temporary register back to the entry buffer. The temporary register is unaffected by the **%]** command, allowing the contents of the temporary register to be copied to the entry buffer multiple times.

%]	Restores the contents of the temporary register to the entry buffer.
Scale %]	Copies "Scale " to the entry buffer and appends the contents of the temporary register.
3,%]%y	Inserts the contents of the temporary register as the database number in this "make row" command.
80.1P="P3,10"%o **80.1P%o%[** **80.2P%o%]%C**	Displays the contents of VAR#2 on the 4X20 VFD beginning at the position assigned in VAR#1.

%^ CALL \ GO TO MACRO

Syntax

Go To Macro

< *macro#* > %^

Call Macro

< *macro#* > C%^

Arguments

macro# Macro number (1 → 250).

See Also

%T Tag Position

%J Jump to Tag

Example:
Branching from One Macro to Another

`1.1P<10%o`	if net < 10 (under tolerance)…
`  110%^`	go to UNDER WT macro
`%N`	else
`1.1P>10.5%o`	if net > 10.5 (over tolerance)…
`  111%^`	go to OVER WT macro
`%E`	end if

Go To Macro

< *macro#* > %^

Ends execution of one macro and resumes execution at the beginning of another. This command is typically used after a conditional statement to invoke a new macro routine based on the outcome of the comparison (see example – *Branching from One Macro to Another*).

Another method of branching to a new macro uses a variable's value as the macro number to "go to". The macro number can be assigned in many ways, such as through a "get entry" command or recalling the macro number from a database.

Example:
Using a Macro as a Subroutine

Macro 20 is a subroutine called by macros 11 and 12 to deactivate system outputs. After macro 20 completes, execution resumes in the calling macro to display the cause of shut-down.

```
================================
MACRO #11 – HI LEVEL INPUT
20C%^
HI LEVEL SHUT-DOWNP1,2a%C
[START] to ResumeP3,1%C
[STOP]  to AbortP4,1%C

MACRO #12 – THERMAL OVERLOAD
20C%^
THERMAL SHUT-DOWNP1,2a%C
[START] to ResumeP3,1%C
[STOP]  to AbortP4,1%C

MACRO #20 – SHUT-DOWN SEQUENCE
U1-5%D
1-5:0%D
10%B
11%B
```

110%^	Ends the current macro and executes macro 110.
80.15P%o%^	Ends the current macro and executes the macro specified by the current value of VAR#15.
EnterProd#%G %^	Ends the current macro and executes the macro specified by the operator entry.

Call Macro

< *macro#* > C%^

Suspends execution of one macro while executing another. The suspended macro is placed in the first position of the macro stack. Thus when the called macro completes, the suspended macro resumes execution ahead of any other macros. The macro resumes with the command immediately following the call statement.

A called macro can call yet another macro. The first called macro is then placed in the first position of the macro stack ahead of its calling macro. Macros can call other macros in this manner up to the remaining capacity of the macro stack allowing called macros to resume in reverse order.

Calling a macro can be used to invoke a common subroutine as shown in the example – *Using a Macro as a Subroutine*. This reduces memory consumption and helps streamline you macros.

110C%^ Suspends execution of the current macro, executes macro 110, then returns to the next command of the calling macro.

If the execution of a called macro is terminated by a *%B Break* command, all calling macros are cleared from the macro stack and will not be resumed.

%_ IF DATABASE ERROR

Syntax **If Database Error**

`[ error# ] %_`

Arguments

error# Database error that occurred as a result of the last database macro command. Select one of the following database errors:

0 If no error occurred

1 If bad entry (invalid entry type)

2 If invalid database specified (database not defined)

3 If invalid column specified (column not defined)

4 If record not found

5 If not enough memory

6 If checksum error (row contains corrupt data)

7 If list corrupt (bad link - list of stored rows not intact)

8 If operation aborted (i.e. search, print, sort, upload)

9 If ID too long (entry exceeds maximum string length)

10 If data type mismatch

11 If greater than maximum number of row allowed

12 If invalid data type (string stored in numeric parameter; results in '0' stored for numeric parameter)

13 If extra characters found (string characters found when numeric-only data was expected)

14 If not enough columns / row received during database upload

15 If too many columns / row received during database upload

16 If greater than maximum number of columns found (likely due to missing <CR> at end of each data row)

Example:
Determining if a Database Record Exists

This routine determines if the entered ingredient number exists in database 2. If not, an error message is displayed and the macro stops.

```
==============================
EnterIng#?%G        prompt for entry
%[                  save entry
1,2;%]%y            recall from database

4%_                 if record not found…
  NOT FOUND%S%P       error prompt
  %B                  break
%E                  end if
```

Notes Omit *error* to test for <u>any</u> database error.

See Also **%y Database Commands**

If Database Error

`[ error# ] %_`

Determines if a database error occurred during the last database operation. The error code generated by the last **%y** database command remains unchanged until the next **%y** command is executed.

%_	Determines if any database error occurred.
0%_	Determines if no database error occurred.
4%_	Determines if the specified record is not found.
9%_	Determines if a string entry exceeded the maximum length.

%` SCALE SELECT

Syntax

Scale Select
`[scale#] %``

Reset A/D Converter
`R [scale#] %``

Arguments

scale# Select from the following scale numbers:

0	Current scale (Reset A/D Converter only)
1	Scale 1
2	Scale 2
3	Scale 3
4	Scale 4
5	Scale 5
6	Scale 6
7	Scale 7
8	Scale 8
*	All scales (Reset A/D Converter only)

Notes Omitting *scale#* for the Scale Select command selects the next enabled scale.

Omitting *scale#* for the Reset A/D Converter command is the equivalent of specifying the current scale (0).

See Also **%# Current Scale**

%- Perform Scale Specific Function

Scale Select

`[scale#] %``

Simulates the operation of the **[SCALE SELECT]** key. It can be used to select the next enabled scale as the current scale, or it can access a specific scale number directly.

%`	Selects the next enabled scale as the current scale. If the current scale is the last enabled scale, the first enabled scale is selected.
1%`	Selects scale 1.
2%`	Selects scale 2.
3%`	Selects scale 3.
4%`	Selects scale 4.

Reset A/D Converter

`R [scale#] %``

Reset a scale's A/D converter in the event the displayed weight locks up. This command is used primarily as a diagnostic tool.

R%`	Resets the A/D converter for the current scale.
R0%`	Resets the A/D converter for the current scale.
R1%`	Resets the A/D converter for scale 1.
R2%`	Resets the A/D converter for scale 2.
R*%`	Resets the A/D converter for all scales.

%a TARGET ACCURACY

Syntax **If Target Accuracy Achieved**

%a

Set Target Accuracy

`< %accuracy > %a`

Get Target Accuracy

?%a

Restore Default Accuracy

*%a

Arguments

%accuracy Target accuracy percentage (90 → 99.96; 0 = Not Enforced).

Notes

The *%accuracy* argument does not change the setup mode selection at P183, rather it temporarily changes the accuracy requirement until power is interrupted or upon saving changes when exiting the setup mode.

A *%accuracy* less than 90 eliminates the accuracy requirement.

See Also

%b Perform Sample

%g Sample / Macro Error

If Target Accuracy Achieved

%a

Determines if the last sample was large enough to meet the accuracy requirement set at P183. This command can be used to ensure an accurate sample before proceeding with other macro routines (see example – *Checking for Sample Accuracy*).

Example:
Checking for Sample Accuracy

`1%T`	tag #1
`%b`	perform sample
`%a`	if accuracy achieved
`  10%^`	go to macro 10
`%N`	else
`  Re-Sampl%S%P`	error prompt
`  1%J`	jump to tag #1
`%E`	end if

Set Target Accuracy

< *%accuracy* > %a

Override the power-up accuracy selection at P183.

99.48%a	Sets the target accuracy requirement to 99.48%.
99.92%a	Sets the target accuracy requirement to 99.92%.
0%a	Eliminates the accuracy requirement.

Get Target Accuracy

?%a

Copies the current accuracy requirement to the entry buffer where it can be saved to a parameter and/or used in math commands.

Restore Default Accuracy

*%a

Restores the accuracy requirement to the value set at P183.

%b PERFORM SAMPLE

Syntax **Perform Sample**
%b

See Also **%a Target Accuracy**
%g Sample / Macro Error
Sampling to Establish The Piece Weight

Perform Sample

Suspends macro execution and performs the sample routine. The quantity mode is selected automatically and a tare is performed. The display then prompts for the default sample size and the standard sample routine continues.

Once the sample routine is completed, macro execution resumes with the next instruction after the **%b** command. Pressing **[CLR]** will completes the sample routine by aborting the sample process.

%c CLEAR ENTRY BUFFER

Syntax **Clear Entry Buffer**
%c

See Also **%[Save Entry Buffer**

Clear Entry Buffer

Simulates the operation of the **[CLR]** key. It can be used at the beginning of a macro to ensure the entry buffer is clear before the macro begins. The **%c** command can also be used to clear the value of the currently selected parameter (see example – *Clearing the Currently Selected Parameter*).

Example:
Clearing the Currently Selected Parameter

%c	clear entry buffer
1%i	select VAR#1
%c	clear VAR#1 (set to 0)
%G	get entry
%e	save entry in VAR#1

%d DISPLAY CONTROL

Syntax **7-Segment VFD On/Off**
< *mode* > %d

7-Segment VFD and Backlight Brightness

< %brightness > P%d

4X20 VFD and LCD Enable/Disable Auto-Update
< X | x > %d

The LED remote display is not available at this time. Please contact GSE for more information.

LCD Auto-Update Position (Standard, Large Font)
< *row* > < H | h > %d

LCD Auto-Update Position (Standard, Medium Font)
< *row* > , < *column* > < I | i > %d

LCD Auto-Update Position (Single-Line, Medium Font)
< *row* > < J | j > %d

LCD Auto-Update Position (Single-Line, Small Font)
< *row* > , < *column* > < K | k > %d

Enable LCD/LED Remote Display Auto-Update
R%d

Disable LCD Remote Display Auto-Update
[*text*] r%d

Disable LED Remote Display Auto-Update
< A|B|C|D|E|F|G|H|I|J > [*text*] r%d

LCD Enable/Disable Backlight
< B | b > %d

LCD Remote Display Enable/Disable Mirror Image
< M | m > %d

Arguments (VFD and LCD Displays)

H	Select large font standard auto-update, normal (black on white).
h	Select large font standard auto-update, inverse (white on black).
I	Select medium font standard auto-update, normal (black on white).
i	Select medium font standard auto-update, inverse (white on black).
J	Select medium font single-line auto-update, normal (black on white).
j	Select medium font single-line auto-update, inverse (white on black).
K	Select small font single-line auto-update, normal (black on white).

k	Select small font single-line auto-update, inverse (white on black).
X	Enable VFD and LCD auto-update.
x	Disable VFD and LCD auto-update.
mode	Select from the following standard VFD controls: display off display on A auto shut-off
%brightness	Percentage of brightness for standard VFD (0 → 100).
row	Row (1 → 16) to position LCD cursor.
column	Column (1 → 40) to position LCD cursor.

<u>Arguments (Remote Displays)</u>

A	Clear LED remote display.
B	Display first 12 LED text characters.
C	Scroll LED text from left to right.
D	Scroll LED text from right to left.
E	Set LED scroll speed to very fast.
F	Set LED scroll speed to fast.
G	Set LED scroll speed to medium.
H	Set LED scroll speed to slow.
I	Set LED scroll speed to very slow.
J	Turn on all pixels of the LED display
B	Enable LCD backlight.
b	Disable LCD backlight.
M	Enable LCD mirror imaging.
m	Disable LCD mirror imaging.
text	Text to be displayed.

<u>Notes</u> Arguments for the %d command will not change the corresponding setup mode selections (P420 for *mode*, P423 for *%brightness*, P425 for *X* and *x*), rather they temporarily change the function until power is interrupted or upon saving changes when exiting the setup mode.

<u>See Also</u> **%C Display Text on 4X20 VFD and LCD**
%K Get Entry from 4X20 VFD and LCD

Standard VFD On/Off

< *mode* > %d

Override the power-up display selection at P420. The display can be turned on, off, or set for auto shut-off.

Turning the display off when the scale is idle or when the display is not required can help reduce power consumption and improve macro execution speed. Turning the display on or off uses the percentage of brightness or dimness settings at P423 and P424 respectively. P424 must be set to “OFF” to completely blank the display.

0%d	Turns the standard 7-segment VF display off using the dimness specified at P424.
1%d	Turns the standard 7-segment VF display on using the brightness specified at P423.
A%d	Sets the standard 7-segment VF display to auto shut-off.

4X20 VFD Auto-Update

< X | x > %d

Override the power-up display selection at P425. The top line auto-update of the 4X20 VFD can be turned on or off. This allows the 4X20 VFD to use the top line auto-update feature to mimic the information of the standard 7-segment VFD, or turn the auto-update off to use the entire display for prompting. Turning the top line auto-update feature off does not clear the displayed information.

X%d	Turns the 4X20 VF display auto-update on. Any characters on the top line will be overwritten.
x%d	Turns the 4X20 VF display auto-update off. Any characters on the top line will continue to be displayed until cleared.

Example:
Disabling/Enabling the 4X20 Auto-Update

```
MACRO #250 – POWER-UP
x%d                  disable auto-update
GSEP1,9a%C
Batch SystemP3,5%C
v 1.00P4,8%C
%P%P                 pause (2 sec)
Pa%C                 clear display
X%d                  enable auto-update
```

Standard VFD Brightness

< *%brightness* > P%d

Overrides the power-up display brightness selection at P423. The brightness of the standard 7-segment VF display can be changed from 0 → 100%.

Example 9-55 shows how to flash the 7-segment VFD using the brightness command in combination with a timer setpoint providing an eye-catching strobe effect. The flash routine can be started in any macro with a **100%A** command. Invoking macro 101 will stop the flashing and ensure the display reverts to 100% brightness.

0P%d	Turns the standard 7-segment VF display off.
50P%d	Sets the standard 7-segment VF display brightness to 50%.
100P%d	Sets the standard 7-segment VF display brightness to 100%.

Example:
Flashing the 7-Segment Display

```
5099%s100%e          Setpt   100
5100%s1%e            SPTyp Outpt
5101%sFLASH%e        SPNam FLASH
5110%s5%e            Activ Never
5111%s0.25%e         AcDly 0.25
5112%s100%e          AcMac   100
5130%s4%e            Deact Alwys
5131%s0.75%e         DeDly 0.75
5132%s100%e          DeMac   100
5133%s0%e            DeMtn Ign'd

MACRO #100 – FLASH VFD
100%O           if FLASH active…
  100P%d          set brightness to 100%
%N              else
  10P%d           set brightness to 10%
  100%A           restart FLASH timer
%E              end if

MACRO #101 – STOP FLASHING VFD
100:0%D         cancel FLASH timer
100%B           clear macro 100 from stack
100P%d          set brightness to 100%
```

VF Display Brightness Control

This command is for the 6 digit VF display. Please refer to chapter 17 for brightness control of the 4 x 20 VFD.

Syntax: XP%d `Where ‘X’ is the percentage of brightness desired, and ‘P’ is required as a fixed character prefix.`

`LCD backlight ON/OFF only: 0=OFF any other number = ON`

Example:
VF Display Brightness control

```
23P%d          this will set the brightness to 23%
```

Example:
LCD Display Brightness control

```
23P%d          This will set the brightness to 100%

0P%d       This will turn the backlight OFF
```

LCD Auto-Update Position (Standard, Large Font)

< *row* > < H | h > %d

Allows the row position of the standard LCD auto-update display window (large size font) to be specified. For example,

```
5H%d
```

will position the auto-update window in the lower half of the 8X40 LCD.

The 6-digit weight display data will be displayed in large font size, while the 2X5 character prompting area will be displayed in medium font size.

LCD Auto-Update Position (Standard, Medium Font)

< *row* > , < *column* > < I | i > %d

Allows the coordinates of the standard LCD auto-update display window (medium size font) to be specified. For example,

```
15,21I%d
```

will position the auto-update window in the lower right corner of the 16X40 LCD.

The 6-digit weight display data will be displayed in medium font size, while the 2X5 character prompting area will be displayed in small font size.

LCD Auto-Update Position (Single-Line, Medium Font)

< *row* > < J | j > %d

Allows the row position of the single-line LCD auto-update display window (medium size font) to be specified. For example,

```
3J%d
```

will position the auto-update window in across the 3rd and 4th lines of the LCD.

All display data will be displayed in medium font size across the entire width of the display (similar to the standard auto-update of the 4X20 VFD).

LCD Auto-Update Position (Single-Line, Small Font)

< *row* > , < *column* > < K | k > %d

Allows the coordinates of the single-line LCD auto-update display window (small size font) to be specified. For example,

```
1,21K%d
```

will position the auto-update window in the upper-right corner of the LCD.

All display data will be displayed in small font size across 20 columns of the display.

Enable LCD/LED Remote Display Auto-Update

R%d

Re-enables the auto-update of weight data on the LCD/LED remote display. This command would normally be used revert back to a remote weight display after displaying text.

Disable LCD Remote Display Auto-Update

`[ text ] r%d`

Disables the auto-update of weight data on the LCD remote display. This allows you to display alphanumeric data on the remote display using the *text* argument.

For example,

```
PEN 15r%d
```

or

```
80.1P=”PEN 15”%o
80.1P%or%d
```

will display a pen location on the remote display. The text will remain on the display until changed with another r%d command, or until the auto-update is re-enabled with the R%d command.

Note that the LCD remote display is not capable of displaying all alpha characters. Choose your text to include only characters that can be represented in 7-segment style. Any non-displayable LCD character will be displayed as three horizontal bars. Also note that the LCD remote display will only display the first 6 characters of a string.

Used without the *text* argument, the r%d command will “freeze” the displayed weight.

Disable LED Remote Display Auto-Update

`< A|B|C|D|E|F|G|H|I|J > [ text ] r%d`

Disables the auto-update of weight data on the LED remote display. This allows you to display alphanumeric data on the remote display using the *text* argument as well as control display features such as scrolling text.

Text will remain on the display until changed with another r%d command, or until the auto-update is re-enabled with the R%d command.

Note that the LED remote display will only display the first 12 characters of a string unless the scroll mode is used.

Ar%d	Clears the LED remote display.
BMix Completer%d	Displays “Mix Complete” on the LED

	remote display.
C Drive Truck On Scaler%d	Scrolls the message “Drive Truck On Scale” from left to right using the last scroll speed specified (see note below).
DGDrive Truck On Scaler%d	Scrolls the message “Drive Truck On Scale” from right to left using a medium scroll speed.
Fr%d	Changes the current scroll speed to fast.

Note: The first two characters of a scroll command are reserved for command characters. Thus, if the second character of a scroll string is A → H, it will be misinterpreted as a display control command. To avoid this problem, use a space to separate the control command from the first character of the scroll string. For example, the command

```
CGSE Scale Systemsr%d
```

would interpret the ‘C’ as a scroll command, and the ‘G’ as a set scroll speed command, leaving only “SE Scale Systems” as the scroll characters. Instead, write the command as

```
C GSE Scale Systemsr%d
```

Since the space character is not a valid command character, it will have no effect on the remote display.

LCD Enable/Disable Backlight

```
< B | b > %d
```

Enables or disables the LCD remote display backlight.

B%d	Enables the LCD remote display backlight.
b%d	Disables the LCD remote display backlight.

LCD Remote Display Enable/Disable Mirror Image

```
< M | m > %d
```

Enables or disables the LCD remote display mirror imaging. When enabled, displayed characters can be viewed correctly through the use of a mirror.

M%d	Enables the LCD remote display mirror imaging.
m%d	Disables the LCD remote display mirror imaging.

%e ENTER / SAMPLE

Syntax **Enter / Sample**

`[ entry ] %e`

Arguments

entry In the weigh mode, *entry* is typically a parameter value to be entered. In the counting mode, *entry* is the sample quantity.

See Also
- **%b Perform Sample**
- **%G Get Entry**
- **%K Get Entry from 4X20 VFD**
- **%n Get Numeric Entry**

Enter / Sample

Simulates the operation of the **[ENTER]** key. It can be used in a macro:

After a "get entry" command to save an entry to the currently selected parameter.

To perform an accumulation if the currently selected parameter is a valid accumulation parameter.

To initiate the sample routine from the quantity mode.

1%i	Selects VAR#1 as the current mode and stores an operator entry in VAR#1.
%G	
%e	
0%s	Selects the gross mode and performs an accumulation.
.%e	
30%s	Selects the quantity mode and initiates the sample routine.
%e	

Example:
Identifying a Preset Parameter

This routine transmits the Gross, Tare and Net values out comm port 1. If the tare weight of the current scale is preset, "MANUAL ENTRY" is sent at the end of the tare data.

```
================================
MACRO #10 – PRINT Gross-Tare-Net
1%"                    select comm 1
0.0.0P%o%$             transmit gross weight
13,10%&                transmit <CR><LF>
2.0.0P%o %$            transmit tare weight

2.0%f                  if tare preset…
(MANUAL ENTRY)%$         send text
%E                     end if

13,10%&                transmit <CR><LF>
1.0.0P%o%$             transmit net weight
13,10%&                transmit <CR><LF>
```

%f IF PARAMETER PRESET

Syntax **If Parameter Preset**

`< parm > . < instance > %f`

Arguments

parm Presettable operating parameter.

instance Valid parameter instance (0 → 4).

Notes Parameters 34P and 35P are common to all scales. An instance of 0 should be specified, however any instance number will yield the same result.

See Also **Presettable Parameters**

If Parameter Preset

Determines if a presettable parameter from the following table is preset (i.e. contains a manually entered value):

OPERATING PARAMETER	PARAMETER NAME
2	Tare
3	Gross Total
6	Net Total
31	Quantity Total
34	Average Piece Weight
35	Average Piece Weight X1000
64.5	DSD Tare Weight
64.6	DSD Gross Total
64.7	DSD Net Total

2.1%f Determines if the tare value for scale 1 is preset.

3.2%f Determines if the gross total value for scale 2 is preset.

34.0%f Determines if the average piece weight is preset.

%g SAMPLE / MACRO ERROR

Syntax

If Sample Error

[*error#*] %g

Get Sample Error

?%g

If Macro Error

E%g

Arguments

error# Sample error that occurred as a result of the last sample routine. Select one of the following sample errors:

0 Sample OK
1 Sample too small
2 Sample not accurate
3 Sample size error (sample entry was 0 or > 9999)
4 Sample cannot be counted (required sample > 9999)
5 Sample aborted acc
6 Sample aborted by pressing the **[CLR]** key

7	APW entered manually
8	Sample aborted during auto-tare

Notes Omit *error* to test for any sample error.

See Also **%a Target Accuracy**
%b Perform Sample

If Sample Error

`[ error# ] %g`

Determines if a specific error occurred during the last sample routine.

0%g	Determines if no error occurred during the last sample.
7%g	Determines if the APW was entered manually.
%g	Determines if any sample error occurred during the last sample.

Get Sample Error

?%g

Copies the last sample error code to the entry buffer where it can be saved to a parameter and/or used in math commands.

If Macro Error

E%g

Determines if an error occurred during macro execution. After testing for errors with the **E%g** command, the error flag is cleared until another macro error is encountered.

%i ID

Syntax **ID**
`[ variable# ] %i`

Arguments

variable# Variable number (1 → 999) to select as the current operating mode.

See Also **%e Enter / Sample**

ID

Simulates the operation of the **[ID]** key. It is most commonly used to select a variable as the current mode of operation to simulate manual entries.

1%i	Selects VAR#1 as the current operating mode.

100%i	Selects VAR#100 as the current operating mode.
%I	Invokes the macro menu (if enabled at P806).

%j IF KEY/REMOTE KEY HELD

Syntax **If Key/Remote Key Held**
`[ key ] %j`

Arguments

key ASCII value of the key being held.

Notes Omit *key* to test for any key held.

If Key/Remote Key Held
Checks if a specified key is being held (see Table 9-3). If the specified key is held, the condition is considered true.

Table 9-3: Keypress ASCII Values

KEY	ASCII VALUE
F1	128
F2	129
F2	130
F4 / TARGET	131
F5	132
START	133
STOP	134
SETUP	135
SELECT	243
ZERO	250
TARE	244
UNITS	245
SCALE SELECT	224
PRINT	240
ID	233
ENTER	229
CLEAR	227
.	46
0	48
1	49
2	50
3	51
4	52
5	53
6	54
7	55
8	56
9	57
Any Key	Omit
460/560 Series Remote Key 1	133
460/560 Series Remote Key 2	134

128%j	Checks if **[F1]** key is held.
133%j	Checks if **[START]** key is held on 660 Series.
	Checks if remote key #1 is held on 460 / 560 Series.
%j	Checks if any key is held.

%k *DIGITAL FILTER*

Syntax

Get Current Filter
%k

Set Filter
< *filter#* > %k

Restore Default Filter
*%k

Clear Rate Filter History
R%k

Example:
Changing the Filter Setting While Filling

This setup changes the filter setting during a filling process. The filter is set to a low value during the fast fill process to provide a fast response to the rapidly changing weight. When the fast fill target is reached, macro 12 is invoked to begin the slow fill. The filter setting is increased to provide a more stable weight reading as it approaches the final target. Once the target is achieved, the default filter setting is restored to 2.0 seconds providing a stable static weight display.

```
==================================
116%s5%e          fltr1 2.0 s

MACRO #6 – START FAST FILL
2%k               set 0.25 second filter
1%A               activate fast output

MACRO #12 – START SLOW FILL
3%k               set 0.5 second filter
2%A               activate slow output

MACRO #13 – FILL COMPLETE
*%k               set default filter (2 sec)
Fill Done!%S%P    prompt
```

Arguments

filter# Select one of the following filter numbers:

0.06 seconds
0.13 seconds
0.25 seconds
0.50 seconds
1.00 seconds
2.00 seconds
4.00 seconds
8.00 seconds
No filtering

Notes

The *filter#* argument does not change the setup mode selection at P116, rather it temporarily changes the filter until power is interrupted or upon saving changes when exiting the setup mode.

Auto-filter selections cannot be selected with the %k command, but can be restored with the *%k command.

Get Current Filter

%k

Copies the current filter selection for the selected scale to the entry buffer where it can be saved to a parameter and/or used in math commands.

Set Filter

< *filter#* > %k

Override the power-up filter selection at P116. This command affects only the currently selected scale. Auto-filter selections are not valid with the *set filter* command.

0%k	Selects 0.06 second filter.
11%k	Disables filtering.

Restore Default Filter

*%k

Restores the filter setting for the currently selected scale to the value set at P116.

Restore Default Filter

R%k

Clears the rate history and begins recalculating the average rate with the next A/D conversion. This is useful in applications such as loss-in-weight during a reversal in flow rate. For example, consider emptying a hopper using a rate measurement period of 10 seconds. When the low limit is reached the supply hopper begins refilling the weigh hopper. This causes a reversal in the flow rate. However, since the rate is an average of the last 600 A/D readings (60 A/D per second X 10 seconds), the displayed rate will not be accurate until 10 seconds after the rate reversal. If the R%k command was executed at the time of rate reversal, then the rate history would be cleared and the rate reversal would be instantly realized.

%m MODIFY STRING

Syntax **Parse String**

M < *variable#* > , < *position* > [, *length*] %m

Set String Case
< U | L > < *variable#* > %m

Get String Length
N < *variable#* > %m

Get Character Position in String
P <variable#>,<character>[,position][,length] %m

If Character Found in String
I < variable# > , < character > %m

Notes A variable value can be substituted for *position* and/or *length* using the syntax

< variable# > P

Arguments

U	Convert all characters in string to upper case .
L	Convert all characters in string to lower case.
variable#	String variable number (1 → 999) to modify or evaluate.
position	Starting position in string variable.
length	Number of characters to evaluate within string variable.
character	Character to find within string variable.

See Also **%o String Concatenation**

Parse String

M < *variable#* > , < *position* > [, *length*] %m

Modifies the contents of a string variable to contain a subset of the original string (see Table 9-4). The subset is specified in terms a starting position within the string followed by the number of characters (length) to parse. If a length is not specified, all characters from the starting position to the end of the string are assumed.

Set String Case

< U | L > < *variable#* > %m

Changes the case of all characters in a string to either upper or lower case (see Table 9-4).

Get String Length

N < *variable#* > %m

Copies the length of a string to the entry buffer where it can be saved to a parameter and/or used in math commands (see Table 9-5).

Get Character Position in String

`P <variable#> , <character> [, position] [, length] %m`

Copies the position of a character in a string to the entry buffer where it can be saved to a parameter and/or used in math commands (see Table 9-5). If a starting position is specified, characters preceding the starting position are ignored. If a length is specified, trailing characters are ignored. A value of zero (0) is copied to the entry buffer is the character is not found.

If Character Found in String

`I <variable#> , <character> [, position] [, length] %m`

Determines if a string contains a specified character (see Table 9-6). If a starting position is specified, characters preceding the starting position are ignored. If a length is specified, trailing characters are ignored.

Table 9-4: String Operations with Result Replacing Original String

String Command (VAR#2 = 2) (VAR#3 = 3)	String 1 contains "1234567890"	String 1 contains "abcdefgh"	String 1 contains "AbCdE12345"	String 1 contains "abcdeabcde"
M1,3,4%m	3456	cdef	CdE1	cdea
M1,7%m	7890	gh	2345	bcde
M1,2P,3P%m	234	bcd	bCd	bcd
M1,2P,1%m	2	b	b	b
U1%m	1234567890	ABCDEFGH	ABCDE12345	ABCDEABCDE
L1%m	1234567890	abcdefgh	abcde12345	abcdeabcde

Table 9-5: String Operations with Result Stored in Entry Buffer

String Command (VAR#2 = 2) (VAR#3 = 3)	String 1 contains "1234567890"	String 1 contains "abcdefgh"	String 1 contains "AbCdE12345"	String 1 contains "abcdeabcde"
P1,b%m	0	2	2	2
P1,b,4%m	0	0	0	7
P1,1,1,3%m	1	0	0	0
P1,2,3P%m	0	0	7	0
P1,C%m	0	0	3	0
N1%m	10	8	10	10

Table 9-6: Conditional String Operations

String Command (VAR#2 = 2) (VAR#3 = 3)	String 1 contains "1234567890"	String 1 contains "abcdefgh"	String 1 contains "AbCdE12345"	String 1 contains "abcdeabcde"
I1,b%m	False	True	True	True
I1,b,4%m	False	False	False	True
I1,1,1,3%m	True	False	False	False
I1,C,2P,3P%m	False	False	True	False
I1,b,3P%m	False	False	False	True

%n GET NUMERIC ENTRY

Syntax **Get Numeric Entry**
`[ prompt ] [ ,* ] %n`

Arguments

prompt Text to be displayed as an entry prompt on the 2X5 character matrix of the 7-segment VFD.

* Entered characters will be displayed as an asterisks '*'.

Notes Limit the *prompt* to 10 characters. If more than 10 characters are specified, only the last 10 will be used.

The first 5 characters of the *prompt* are displayed on the top line of the 2X5 matrix, the last 5 characters on the bottom line.

See Also

%G Get Entry
%K Get Entry from 4X20 VFD
%\ If No Entry
%[Save Entry Buffer
%o Math Assignment

Get Numeric Entry

`[ prompt ] [ ,* ] %n`

Example:
Getting a Numeric Operator Entry

`80.2P=0%o`	clear target value
`%T`	tag position
`EnterTargt%n`	get entry
`%\`	if no entry…
`  Must Enter%P`	prompt
`  %J`	jump to tag
`%N`	else
`  =80.2P%o`	save entry
`%E`	end if

Accepts operator numeric-only input. When this command is executed, the macro is suspended until the entry is completed by pressing **[ENTER]**. The optional *prompt* will be displayed until the first entry character is received. The **%n** command will accept an entry from the front panel keypad or any enabled comm port. Numbers remain in the entry buffer until **[ENTER]** is pressed or a carriage return <CR> is received on one of the comm ports, allowing the next macro command to retrieve and/or store the entry. Press **[CLR]** to clear the entire entry from the entry buffer, display the optional *prompt*, and restart the entry process.

Use of the asterisks '*' argument will cause each entered character to appear as an asterisks on the display. This provides a means of adding security to a user entry such as a password. Note that the asterisks characters only appear during the entry. When **[ENTER]** is pressed to complete the entry, the entered characters are put into the entry buffer and become visible on the display. Be sure to copy the entry to a variable or other parameter immediately after the entry is complete to prevent this from happening.

KeyInTargt%n **=80.11P%o**	Prompts for target entry and stores entry in VAR#11.
11%i **%n**	Simulates a manual entry into VAR#11. Here, VAR#11 is selected as the current mode, using the variable's name as the prompt. Use this method when entering time/date values or scale-specific

%e	float values when using selectable units.
EnterTank#%n **%A**	Prompts for a tank number, expecting a valid output setpoint number for activation.
TruckID# ?%n **%[**	Prompts for a truck ID# and moves the entry from the entry buffer to the temporary buffer.
EnterTare?%n **%t**	Prompts for a tare entry and performs a tare using the entry as manual tare value.
EnterCode?,*%n	Prompts for a code entry and displays an asterisks '*' symbol in place of each digit.

%o MATH ASSIGNMENT

Syntax

A = B (*Copy Value*)
< parm | const > = < parm | const > %o

A = Entry Buffer (*Entry Buffer Assignment*)
= < *parm* > %o

A = B + C (*Equation Assignment*)
<parm>=<parm|const> <math> <parm|const> %o

A = A + B (*Modify Original Value*)
< *parm* > < *math* > = < *parm* | *const* > %o

A = A + (B + C) (*Modify Original Value*)
<parm><math>=<parm|const> <math> <parm|const> %o

Entry Buffer = A + B (*Equation Assignment*)
<*parm|const*> <*math*> <*parm|const*> %o

Copy A to Entry Buffer
< *parm* > %o

Arguments

parm — Operating parameter with the syntax:
< *parm* > [. *instance*] [. *format*] < P | p | q >

where *parm* is an operating parameter with a valid *instance* and *format* code, and

P represents *parm* as any parameter value,

p represents *parm* as a pointer to another variable, q — represents *parm* as a pointer to a non-variable parameter.

const — Constant value.

math — Select one of the following math operators:

+ Add
- Subtract
* Multiply
/ Divide
| Modulus (divide and determine remainder)
^ Exponent

Notes — The *instance* argument is required when specifying parameter 50, 51, 52, 80, 81 or 82.

Omitting *instance* for a scale-specific parameter assumes the current scale.

See Also

%[Save Entry Buffer

%] Restore Entry Buffer

%v Write Value to EEPROM

Pointers

A = B (Copy Value)

A math assignment can be used to copy the value of one parameter or constant to another parameter. Assignment is from right to left. For example,

```
80.1P=0.1P%o
```

copies the gross weight of scale 1 to VAR#1. It is possible to reverse this statement and copy the value of VAR#1 to the gross weight of scale 1. However, the gross weight is an active weight parameter that will be recalculated when the next A/D conversion occurs (typically every 1/60th second). This holds true for all other active weight parameters such as net, quantity, rate, etc.

A = Entry Buffer (Entry Buffer Assignment)

When assigning a value directly from the entry buffer, assignment is from left to right. For example,

```
=80.1P%o
```

copies the contents of the entry buffer to VAR#1.

A = B + C (Equation Assignment)

A math assignment can assign the math operation of two parameters and/or constants to another parameter. Assignment will be in the direction of the single parameter. For example,

80.2P=1.1P+1.2P%o

1.1P+1.2P=80.2P%o

are equivalent statements that copies the sum of the net weights for scales 1 and 2 to VAR#2.

A = A + B (Modify Original Value)

A math operator can be used in the assignment location of a math operation to modify the original value of a parameter. For example,

80.3P+=1%o

is equivalent to

80.3P=80.3P+1%o

which increments the original value of VAR#3 by one. Likewise,

80.3P^=.5%o

will calculate the square root of VAR#3.

A = A + (B + C) (Modify Original Value)

Two parameters and/or constants can be used when modifying the original value of another parameter. For example,

80.3P+=0.1P+0.2P%o

is equivalent to

80.3P=80.3P+0.1P+0.2P%o   (*invalid syntax*)

which adds the gross weights of scales 1 and 2 to the original value of VAR#3. Note that the second macro statement could not be used as a valid command because a math operation may not contain more than two parameters and/or constants.

Entry Buffer = A + B (*Equation Assignment*)

Performing a math operation without an assignment parameter will copy the result of the operation to the entry buffer. For example,

6.1P+6.2P%o

copies the sum of the net totals for scales 1 and 2 to the entry buffer.

Copy A to Entry Buffer

Performing a math operation without an assignment or math operation will copy the contents of a parameter to the entry buffer. For example,

80.1P%o

copies the contents of VAR#1 to the entry buffer.

Formatted Math Assignments

Parameters in a *math assignment* command can be formatted as described in *Entering Parameters* on page 8-26. This is useful when copying numeric values to a string variable or to the entry buffer. When formatting parameters in a math assignment, you must specify an instance. For parameters that do not have an instance, specify an instance of zero. For example,

```
80.3P=11.0.18560%o
```

copies the a text format of the time/date parameter to string VAR#3. Similarly,

```
11.0.18560P%o%$
```

copies a text format of the time/date parameter to the entry buffer, then transmits it out the comm port last selected by the **%"** command.

Math Assignments Using Different Data Types

When assigning a float value to a float-type parameter, the result will be stored as a float value. Similarly, assignments using only integers will store integer values. Assignments using only strings will produce string values. However, different rules apply when using math assignments to combine more than one data type (see Table 9-7).

Table 9-7: Rules for Assignments Using Different Data Types

WHEN COPYING…	RULES
Float Values to Integer Parameters	The decimal portion of the float value is truncated.
Float Values to Unsigned Integer Parameters	The decimal portion of the float value is truncated.
Float Values to String Variables	The original float value is stored as a string.
Integer Values to Float Parameters	The original integer value is stored as a float to a resolution of 1 part in 16,000,000.
Integer Values to Unsigned Integer Parameters	A positive integer value is stored as an unsigned integer. A negative integer value is subtracted from the roll-over unsigned integer value of +4294967296. For example, copying an integer value of –1 to an unsigned integer yields a value of 4294967295.
Integer Values to String Variables	The original integer value is stored as a string.
Unsigned Integers to Float Parameters	The original unsigned integer value is stored as a float to a resolution of 1 part in 16,000,000.
Unsigned Integers to Integer Parameters	An unsigned integer value less than +2147483648 is stored as a positive integer value. An unsigned integer value of +2147483648 or greater is subtracted from the roll-over value of +4294967296 and stored as a negative integer value. For example, copying an unsigned integer value of +2147483648 to an integer yields a value of -2147483648.
Unsigned Integers to String Variables	The original integer value is stored as a string.
STRING VALUES TO FLOAT PARAMETERS	The original string value up to the first non-numeric character is stored as a float value. If the string begins with a non-numeric character, the float is set to zero (0). For example, A string value of "10.55" will be stored in a float parameter as 10.55. A string value of "1.3a45" will be stored in a float parameter as 1.3. A string value of "A100.50" will be stored in a float parameter as 0.
String Values to Integer Parameters	The original string value up to the first non-numeric character is stored as an integer value. If the string begins with a non-numeric character, the integer is set to zero (0). All other rules for assigning values to integer parameters apply.
String Values to Unsigned Integer Parameters	The original string value up to the first non-numeric character is stored as an unsigned integer value. If the string begins with a non-numeric character, the unsigned integer is set to zero (0). All other rules for assigning values to unsigned integer parameters apply.

String Assignments

Strings containing numeric values can be used to perform math calculations. For example, if variable #5 is a string containing the value 10, then

`80.5P+=1%o`

will increment VAR#5 to a value of 11.

Due to a string's ability to perform math calculations, assigning a value to a string directly from the entry buffer, as with a "get entry" command, requires special consideration when a math operator could appear as part of the entry. For example,

```
EnterPart#%G
=80.5P%o
```

will store an operator part number entry in VAR#5. However, suppose the part number was entered as **100-25**. Since variable 5 is a string, the presence of a "**-**" character entry will cause the entry to be treated as a math command. The value **75** will be stored in VAR#5!

To prevent math operations on string assignments, the assigned value should be encapsulated in quotes " ". This can be accomplished using the *%[Save Entry Buffer* and *%] Restore Entry buffer* commands to store the entry in the temporary register and insert it within quotes in the assignment command. The following example will store a value in string VAR#5 exactly as entered:

```
EnterPart#%G
%[
80.5P=”%]”%o
```

Table 9-8: Math Assignment Examples

ASSIGNMENT	TYPE	VAR#1 (FLOAT)	VAR#2 (INT)	VAR#3 (U-INT)	VAR#4 (STRING)	ENTRY BUFFER
A = B (Copy Value)						
80.1P=10.95%o	A = B	**10.95**				
80.2P=80.1P%o	A = B	10.95	**10**			
100=80.3P%o	A = B	10.95	10	**100**		
80.4P=50-10%o	A = B	10.95	10	100	**40**	
80.4P="50-10"%o	A = B	10.95	10	100	**50-10**	
80.3P=80.4P%o	A = B	10.95	10	**50**	50-10	
80.2P=2147483647%o	A = B	10.95	**2147483647**	50	50-10	
80.2P=2147483648%o	A = B	10.95	**-2147483648**	50	50-10	
80.2P=2147483649%o	A = B	10.95	**-2147483647**	50	50-10	
80.2P=-2147483648%o	A = B	10.95	**-2147483648**	50	50-10	
80.2P=-2147483649%o	A = B	10.95	**2147483647**	50	50-10	
80.3P=4294967295%o	A = B	10.95	2147483647	**4294967295**	50-10	
80.3P=4294967296%o	A = B	10.95	2147483647	**0**	50-10	
80.3P=4294967297%o	A = B	10.95	2147483647	**1**	50-10	

ASSIGNMENT	TYPE	VAR#1 (FLOAT)	VAR#2 (INT)	VAR#3 (U-INT)	VAR#4 (STRING)	ENTRY BUFFER
80.4P=12A34.8%o	A = B	10.95	2147483647	4294967295	**12A34.8**	
80.1P=80.4P%o	A = B	**12**	2147483647	4294967295	12A34.8	
80.1P=0.99%o	A = B	**.99**	2147483647	4294967295	12A34.8	
80.2P=0.99%o	A = B	.99	**0**	4294967295	12A34.8	
80.4P=0.99+.01%o	A = B	.99	0	4294967295	**1**	
80.4P="0.99+.01"%o	A = B	.99	0	4294967295	**0.99+.01**	
A = Entry Buffer (Entry Buffer Assignment)						
=80.1P%o	A = Entry Buffer	**1.5**				1.5
=80.2P%o	A = Entry Buffer	1.5	**-25**			-25.8
=80.3P%o	A = Entry Buffer	1.5	-25	**100**		100
=80.4P%o	A = Entry Buffer	1.5	-25	100	**70**	100-30
=80.4P%o	A = Entry Buffer	1.5	-25	100	**100-30**	"100-30"
A = B + C (Equation Assignment)						
80.1P=10*10%o	A = B + C	**100**				
10*5.59=80.2P%o	B * C = A	100	**55**			
80.3P=80.1P+80.2P%o	A = B + C	100	55	**155**		
80.1P=80.2P-100%o	A = B – C	**-45**	55	155		
80.1P=80.3P/10%o	A = B / C	**15.5**	55	155		
80.2P=80.3P/10%o	A = B / C	15.5	**15**	155		
80.4P=80.3P/10%o	A = B / C	15.5	15	155	**15.5**	
80.2P=2*80.1P%o	A = B * C	15.5	**31**	155	15.5	
80.3P=80.2P|10%o	A = B | C	15.5	31	**1**	15.5	
80.3P=80.2P/10%o	A = B / C	15.5	31	**3**	15.5	
80.1P=80.3P^80.3P%o	A = B ^ C	**27**	31	3	15.5	
80.2P=80.3P^4%o	A = B ^ C	27	**81**	3	15.5	
80.3P=80.2P^.5%o	A = B ^ C	27	81	**9**	15.5	
A = A + B (Modify Original Value)						
80.1P*=0%o	A = A + B	**0**				
80.1P+=1%o	A = A + B	**1**				
80.1P+=1%o	A = A + B	**2**				
80.1P-=1%o	A = A - B	**1**				
80.1P*=10%o	A = A * B	**10**				
80.1P*=10%o	A = A * B	**100**				
80.1P/=50%o	A = A / B	**2**				
80.1P^=3%o	A = A ^ B	**8**				
80.1P|=3%o	A = A | B	**2**				
A = A + (B + C) (Modify Original Value)						
80.1P*=80.2P*0%o	A = A * (B * C)	**0**	10	3	2	
80.2P/=80.3P+80.4P%o	A = A / (B + C)	0	**2**	3	2	
80.3P+=80.2P*10%o	A = A + (B * C)	0	2	**23**	2	
80.4P-=80.3P*-1%o	A = A - (B * C)	0	2	23	**25**	
80.2P^=80.2P^80.2P%o	A = A ^ (A ^ A)	0	**16**	23	25	
80.3P|=80.2P/4%o	A = A | (B / C)	0	16	**3**	25	
Entry Buffer = A+ B (Equation Assignment)						
80.1P+80.2P%o	Buffer = A + B	10	20	30	40	**30**
80.4P-80.3P%o	Buffer = A - B	10	20	30	40	**10**
80.1P*8.1%o	Buffer = A * B	10	20	30	40	**81**

ASSIGNMENT	TYPE	VAR#1 (FLOAT)	VAR#2 (INT)	VAR#3 (U-INT)	VAR#4 (STRING)	ENTRY BUFFER
80.3P/6%o	Buffer = A / B	10	20	30	40	**5**
80.2P|6%o	Buffer = A | B	10	20	30	40	**2**
2^80.1P%o	Buffer = A ^ B	10	20	30	40	**1024**
Copy A to Entry Buffer						
80.1P%o	Entry Buffer = A	10	20	30	40	**10**
80.3P%o	Entry Buffer = A	10	20	30	40	**30**

%o MATH COMPARISON

Syntax

If A = B

```
< parm | const > < cond > < parm | const >
%o
```

If A = B + C

```
<parm> <cond> < parm  | const > <math> <
parm | const > %o
```

If A + B = C

```
< parm | const > <math> < parm | const >
<cond> < parm > %o
```

If Entry Buffer = A

```
< cond > < parm | const > %o
```

Arguments

parm Operating parameter with the syntax:

```
< parm > [ . instance ] [ . format ] < P |
p | q >
```

where *parm* is an operating parameter with a valid *instance* and *format* code, and

P represents *parm* as any parameter value,

p represents *parm* as a pointer to another variable,

q represents *parm* as a pointer to a non-variable parameter.

const Constant value.

math Select one of the following math operators:
- \+ Add
- \- Subtract
- * Multiply
- / Divide
- | Modulus (divide and determine remainder)
- ^ Exponent

cond Select one of the following conditional operators:
- > Greater than

>= Greater than or equal to
< Less than
<= Less than or equal to
== Equal to
!= Not equal to

See Also **%] Restore Entry Buffer**
Boolean Logic
Pointers

If A = B

Determine the relationship between two parameters and/or constants.

80.1P==10%o Determines if VAR#1 equals 10.

0.0P<=0%o Determines if the gross weight is less than or equal to 0.

2.0P!=0%o Determines if the tare weight is NOT equal to 0.

34P>80.3P%o Determines the APW is greater than the value in VAR#3.

If A = B + C

If A + B = C

Evaluates a mathematical expression.

80.1P==6.1P+6.2P%o Determines if VAR#1 equals the sum of the net totals for scales 1 & 2.

80.2P>=80.3P*10%o Determines if VAR#2 is greater than or equal to 10 times VAR#3.

80.5P^.5>80.4P%o Determines if the square root of VAR#5 is greater than VAR#4.

80.2P|80.3P<2%o Determines if the remainder of VAR#2 divided by VAR#3 is less than 2.

If Entry Buffer = A

Evaluates the contents of the entry buffer.

==6.1P+6.2P%o Determines if the entry buffer equals the sum of the net totals for scales 1 & 2.

>20*80.4P%o Determines if the entry buffer is greater than 20 times VAR#4.

%o *STRING CONCATENATION*

<u>**Syntax**</u>

A = Concatenation of B & C

< parm > = < parm | const > \ < parm | const > %o

A = Concatenation of A & B

< *parm* > \= < *parm* | *const* > %o

A = Concatenation of A & B & C

< parm > \= < parm | const > \ < parm | const > %o

Entry Buffer = Concatenation of A & B

< parm | const > \ < parm | const > %o

<u>**Arguments**</u>

parm Operating parameter with the syntax:

< *parm* > [. *instance*] [. *format*] < P | p | q >

where *parm* is an operating parameter with a valid *instance* and *format* code, and

P represents *parm* as any parameter value,

p represents *parm* as a pointer to another variable,

q represents *parm* as a pointer to a non-variable parameter.

const Constant value.

<u>**See Also**</u>

%[Save Entry Buffer
%m Modify String
%v Write Value to EEPROM
Pointers

A = Concatenation of B & C

“Pastes” two variables together.

80.1P=80.2P\80.3P%o	Copies VAR#2 to VAR#1 and appends the value of VAR#3 to VAR#1.
80.1P="Scale #"\80.4P%o	Copies the text “Scale #” to VAR#1, then appends the value of VAR#4.

A = Concatenation of A & B

A = Concatenation of A & B & C

Append alphanumeric data to an existing string.

80.1P\=80.2P%o	Appends the value of VAR#2 to the value of VAR#1.
80.1P\=" Cycles"%o	Appends the text " Cycles" to the value of VAR#1.
80.1P\=80.2P\80.3P%o	Appends the value of VAR#2 and the value of VAR#3 to VAR#1.

Entry Buffer = Concatenation of A & B

Combine alphanumeric data in the entry buffer.

80.1P\80.2P%o	Copies the value of VAR#1 to the entry buffer and appends the value of VAR#2.
3%" **Scale #\80.1P%o** **%$**	Copies the text "Scale #" to the entry buffer and appends the value of VAR#1, then sends the data out comm port #3.

Table 9-9: Concatenation Examples

ASSIGNMENT	TYPE	VAR#1 (STRING)	VAR#2 (STRING)	VAR#3 (STRING)	VAR#4 (INT)	ENTRY BUFFER
80.1P="Bin #"\80.2P%o	A = B \ C	**Bin #7**	7			
80.1P=80.2P\80.3P%o	A = B \ C	**Scale #2**	Scale #	2		
80.1P=80.2P\80.4P%o	A = B \ C	**Bin #5**	Bin #		5	
80.1P\=80.2P%o	A = A \ B	**Bin #50**	0			
80.1P\=80.2P\80.3P%o	A = A \ B \ C	**Bin #5007**	0	7		
80.1P\=A%o	A = A \ B	**Bin #5007A**				
80.4P=80.2P\80.3P%o	A = B \ C		20	44	**2044**	
80.4P\=80.1P%o	A = A \ B	13A77			**204413**	
80.1P\80.2P%o	Entry Buffer	Row:	14			**Row:14**
Batch #\80.4P%o	Entry Buffer				23	**Batch #23**

%p *PRINT*

Syntax **Print**
`[ transmit# ] %p`

Arguments
transmit# Custom transmit (1 → 250) to send.

Notes At least one *transmit#*, P991must be set for 'onreq' or 'prmpt'.

See Also **%Q Send Custom Transmit**

Print

Simulates the operation of the **[PRINT]** key. All custom transmits set for 'onreq' (on request) at P991 will be transmitted in numeric order out their respective comm ports.

If any custom transmits are set for 'prmpt' (prompt) at P991, then the message `Which Tx# ?` will be displayed prompting the entry of a transmit number. The entered custom transmit number will then be sent, along with any other custom transmits specified as 'onreq'.

Specify a *transmit#* preceding the print command to send <u>only</u> the specified custom transmit. For example,

2%p

sends only custom transmit #2, regardless of any others set for 'onreq'. The message `Which Tx# ?` will be not displayed for transmits set for 'prmpt'. Transmits set for 'off' cannot be sent using this method. Instead, use the %Q Send Custom Transmit command.

%q *ENABLE RS-485 TRANSMITTER*

Syntax **Enable RS-485 Transmitter**
%q

See Also **%$ Send Text**
%& Send Control Code

Enable RS-485 Transmitter

Enables the transmit interrupt for comm port #1. This command is primarily used in conjunction with the RS-485 network option to immediately send all information in the transmit buffer.

Normally, this interrupt is enabled as soon as data is put into the transmit buffer. However if networking is enabled at P250, the transmit interrupt s not enabled until the entire transmission is assembled. In the case of custom transmits or database transmissions, the transmit interrupt is not enabled until the transmit is complete or the transmit buffer becomes full. However in the case of the **%$** and **%&** macro commands, the transmit interrupt is not enabled by itself. Instead it requires the **%q** command to enable transmitter and begin the transmission. Otherwise data will continue to collect in the transmit buffer until it becomes full at which time the transmitter will become enabled automatically.

%r   A/D INTERVAL

Syntax

Set A/D Interval
< *scale#* > , < *interval* > %r

Get A/D Interval
G < *scale#* > %r

Wait for A/D Interval
W < *scale#* > %r

Arguments

scale# Select from the following scale numbers:

0 Current scale
Scale 1
Scale 2
Scale 3
Scale 4
Scale 5
Scale 6
Scale 7
Scale 8
* All scales (Set A/D Interval only)

interval A/D interval at which weight values are recalculated.

Notes

The *interval* is specified in terms of 1/60th second intervals with an offset of 1 (*interval* + 1 = A/D conversion rate). For example, an *interval* of 0 results in the fastest A/D conversion rate, 1/60th second. An *interval* of 1 yields a conversion rate of 2/60th second. An *interval* of 59 yields 1 conversion per second.

Setpoint status is monitored and updated using the A/D interval specified for scale #1.

Set A/D Interval

< *scale#* > , < *interval* > %r

Sets the rate at which weight values for the specified scale are recalculated. Reducing the A/D conversion rate can significantly increase

Example:
Changing the A/D Interval to Increase Macro Execution Speed

This macro example demonstrates how to reduce the A/D interval to more quickly execute a routine that searches through all rows of a database to add one week of time to time/date VAR#2.

```
===================================
*,59%r                  A/D interval = 1/sec

5,1%y                   recall first row

1%T                     tag #1
4%_                     if row not found
%N                      else
  80.2P+=604800%o         add 1 week
  2,1%y                   update row
  6,1%y                   get next row
  1%J                     jump to tag #1
%E                      end if

*,0%r                   A/D interval = 60/sec
```

macro execution speed by reducing the time the processor must spend calculating active weight parameters. For example,

1,59%r

sets the A/D interval for scale #1 to once per second. This results in 59 fewer interrupts from the A/D converter reporting new weight values. Consequently, more macro commands can be executed each second.

Reducing the A/D interval is most useful when executing lengthy macro routines (see example – *Changing the A/D Interval to Increase Macro Execution Speed*). However, it is important to realize that weight data and setpoint status will not processed as often. Therefore, do not reduce the A/D interval significantly during a critical process such as a high speed filling cycle. Also remember to set the A/D interval back to a short interval after the macro routine is complete.

Setpoint status is not updated between A/D intervals. Do not specify a long interval during a critical I/O process!

Specifying an A/D interval does not guarantee that the interval will be achieved. For example, it is not possible to achieve 60 updates per second on four scales simultaneously even if you specify this interval. However, it is possible to obtain 60 updates per second on any one scale by reducing the interval of the others.

Get A/D Interval

G < *scale#* > %r

The *get A/D interval* command copies the current A/D conversion rate for the specified scale to the entry buffer.

G1%r	Copies the A/D interval for scale #1 to the entry buffer.
G2%r	Copies the A/D interval for scale #2 to the entry buffer.
G3%r	Copies the A/D interval for scale #3 to the entry buffer.
G4%r	Copies the A/D interval for scale #4 to the entry buffer.
G1%r **=80.6P%o**	Copies the A/D interval of scale #1 to the entry buffer, then stores the value in VAR#6.
G3%r **>10%o**	Copies the A/D interval of scale #3 to the entry buffer, then determines if it is greater than 10.

Example:
Waiting for the Next A/D Interval

This macro example demonstrates the need to wait for the next A/D interval before activating a setpoint based on the net weight. Without the wait command, it is uncertain that the next A/D interval would occur immediately after the tare function. Thus the net weight would not be updated and consequently the fill output may have deactivated if the previously reported net weight exceeded the new ingredient target.

```
===================================
MACRO #6 - START PRESS
%t                tare scale
W1%r              wait for A/D interval
1%A               activate fill output
```

Wait for A/D Interval

W < *scale#* > %r

Suspends macro execution until the next A/D interval for the specified scale occurs. This command is typically used immediately following a tare function to ensure that all affected weight parameters have been recalculated before checking the result of the tare. This is especially important in batching applications where a tare is performed prior to loading each ingredient. The example – *Waiting for the Next A/D Interval* demonstrates the potential problem of activating the fill output for next ingredient.

The *wait for A/D interval* command can also be used to temporarily "freeze" weight values. This is accomplished by waiting for the next A/D interval, then immediately setting a longer interval. Then wait for the next interval before restoring the original interval duration. Again, it is important to remember that setpoint status will also remain unchanged during this interval.

%s *SELECT MODE*

Syntax **Select Mode**

```
[ parm ] %s
```

Exit Macro Menu

```
*%s
```

Arguments

parm Operating parameter with the syntax:

```
< parm > [ . instance ]
```

where *parm* is a displayable operating parameter with a valid *instance*.

Notes Omitting *parm* toggles through the operating modes specified in the setup mode at P300-P309.

See Also **%M Mode Selection**

Select Mode

```
[ parm ] %s
```

Selects the specified parameter as the currently displayed operating mode. Omitting the parameter instance for scale-specific parameters assumes the currently selected scale.

0%s	Selects the gross weight mode for the current scale.
0.0%s	Selects the gross weight mode for the current scale.
2.2%s	Selects the tare weight mode for scale #2.
80.1%s	Selects VAR#1 as the current operating mode.
11%s	Selects the time & date parameter as the current operating mode.
%s	Selects the next operating mode from the operating modes specified in the setup mode at P300-P309.

Exit Macro Menu

`*%s`

Exits the macro menu without invoking a macro. This can also be accomplished with the **%i** or **%z** command.

%t TARE

Syntax **Tare**
`[ value ] %t`

Arguments

value	Tare weight value.
scale#	Select from the following scale numbers: 0 Current scale 1 Scale 1 2 Scale 2 3 Scale 3 4 Scale 4 5 Scale 5 6 Scale 6 7 Scale 7 8 Scale 8

Notes Specifying a tare *value* sets the preset flag for the tare weight.

See Also
%z Zero
%` Scale Select
%- Perform Scale Specific Function

Tare

`[ value ] %t`

Establishes a new tare weight for the current scale.

%t	Performs a motion delayed auto-tare (tare = gross).
10%t	Simulates a manual tare entry (tare = 10).
0%t	Clears the tare weight (tare = 0).

A tare command must originate from the gross, net, tare, or any of the accumulation parameters (parameters 0P → 9P). If NTEP is disabled in the setup mode at P440, then the net mode is automatically selected after performing a tare (with the exception of performing a tare from the tare mode). If NTEP is enabled, the net mode is not automatically selected if the gross weight is zero.

Negative tares will not be allowed if the negative tare parameter is enabled at P162 in the setup mode.

%u *UNITS*

Syntax

Units

`[ unit#1 ] %u`

If Current Units

`< unit#2 > [ . scale# ] ?%u`

Rename Units

`< unit#2 > , < name > %u`

Arguments

unit#$_1$ Select from the following unit parameter selections:

0 Units assigned at P131 for the current scale
1 Units assigned at P132 for the current scale
2 Units assigned at P133 for the current scale
3 Units assigned at P134 for the current scale

unit#$_2$ Select from the following unit numbers:

0 lb (pounds)
1 kg (kilograms)
2 oz (ounces)
3 g (grams)
4 ton (tons)
5 t (metric ton)
6 ????1 (custom unit 1)
7 ????2 (custom unit 2)
8 LbOz (pounds/ounces)

scale# Select from the following scale numbers:

0 Current scale
1 Scale 1
2 Scale 2
3 Scale 3
4 Scale 4
5 Scale 5
6 Scale 6
7 Scale 7
8 Scale 8

name New name to appear in place of the default unit name on the 2X5 character matrix of the 7-segment VFD.

Notes If a unit is renamed, the new name remains in effect until changed again with the %u command or until power is interrupted. Accessing the setup will not change a unit's name.

If a unit's name exceeds 2 characters, the center-of-zero indication will not appear.

If a unit's name exceeds 4 characters, the scale number will not appear in the case of multiple scale operation.

<u>**See Also**</u>

%R Rename Mode

%` Scale Select

%- Perform Scale Specific Function

Units

`[ unit#`$_1$` ] %u`

Selects the displayed units of measure for the current scale. Omitting *unit#*$_1$ toggles through the units specified in the setup mode at P131→P134.

0%u	Selects the units assigned at P131 for the current scale.
1%u	Selects the units assigned at P132 for the current scale.
2%u	Selects the units assigned at P133 for the current scale.
3%u	Selects the units assigned at P134 for the current scale.
%u	Toggles to the next units assigned at P131→P134.

Example:
Determining the Current Unit of Measure

This example shows how you can use the **[PRINT]** key to send a unique custom transmit for each displayed unit of measure.

```
===================================
805%s10%e          Print Mc 10

MACRO #10 - PRINT PRESS
0?%u                if current units = lb…
  1%Q                 send transmit #1
%N                  else
1?%u                if current units = kg…
  2%Q                 send transmit #2
%E                  end if
```

If Current Units

`< unit#`$_2$` > [ . scale# ] ?%u`

Determine if a specific unit of measure is currently selected.

0?%u	Determines if "lb" is the current units for the current scale.
0.0?%u	Determines if "lb" is the current units for the current scale.
0.1?%u	Determines if "lb" is the current units for scale #1.
0.2?%u	Determines if "lb" is the current units for scale #2.
3.2?%u	Determines if "g" is the current units for scale #2.

Example:
Renaming Units for Prompting

```
10)00  kg
       Gross
```

The display above could be used to show a bag count in VAR#1 along with the gross weight.

```
================================
1,Bag#%u            rename units
80.1P%o%[           copy bag# to temp buffer
0,%]%R              rename gross mode
```

```
10)00  Bag#
       00101
```

Rename Units

`< units#`$_2$` > , < name > %u`

Allows a unit's displayed name to be changed. Once changed, the new name will be displayed every time the specified units are accessed. Renaming the units in this manner allows you to use a unit's name to display a prompt without suspending macro execution. The example – *Renaming Units for Prompting* shows how to use the **%u** command along with the **%R** command to display a bag count on the 2X5 prompting display during a filling cycle.

0,LB%u	Renames “lb” to display “LB” when pounds is selected.
3,Count%u	Renames “g” to display “Count” when grams is selected.
6,Liter%u	Renames custom unit #1 to display “Liter” when selected.

%v WRITE VALUE TO EEPROM

Syntax **Write Value to EEPROM**
`80. < variable# > %v`

Arguments
variable# Valid variable number to store in non-volatile memory.

Notes A variable must be set for “OnReq” (on request) at P684 to utilize the **%v** command.

See Also **%o Math Assignment**

Write Value to EEPROM

Stores the current value of a specified variable to non-volatile to the EEPROM. The stored value will then be retained while the indicator is powered down and be restored upon power-up. Note that the value restored at power-up will be the last value stored using the **%v** command. If the variable’s value was changed thereafter, the new value is not stored.

80.1%v	Stores the value of VAR#1 to non-volatile EEPROM.
80.5%v	Stores the value of VAR#5 to non-volatile EEPROM.

Use the **%v** command should be used instead of the “Auto” save selection at P684 whenever practical. This reduces the number of writes to EEPROM which has a suggested life expectancy of 100,000 writes.

%y RECALL ROW

Syntax **Recall Row**

`1 [ , dbase# ] [, column ] [;value] %y`

Arguments

dbase# Database number (1 → 250).

column Operating parameter representing a database search column with the syntax:

`< parm > . [ instance ]`

where *parm* is a operating parameter with a valid *instance*.

value Specific value or text to recall from the lookup column.

Notes Omitting *dbase#* assumes the last database accessed. If no databases have been accessed, the first defined database is assumed.

Value can be specified without specifying *column*.

See Also **%y Next Match in Database**

%y Set Column in Database

%_ If Database Error

Example:
Recall Row - Sample Database Structure

```
699%s1%e            DB #: 1
700%sPARTS%e        DBNam PARTS
701%s80.1%e         Col01 Part#
702%s80.2%e         Col02 Name
703%s2.1%e          Col03 Tare
```

Box# (80.1P)	Name (80.2P)	Tare (2.1P)
1	SAE5W30	0.95
2	SAE10W30	0.90
3	SAE10W40	0.92
4	SAE15W40	0.89
5	SAE20W50	0.87
6	SAE30	0.90
7	SAE40	0.90
8	SAE50	0.90
9	SAE20W	0.92

Recall Row

Accesses the first database row that matches the search criteria. The search always begins with the first row in the database and continues sequentially to the end. When a match is found, all parameters included in the database are updated with their corresponding values. If no matching rows are found, the 'record not found' flag is set (see *%_ If Database Error*) and the parameter values are unchanged. If a *column* is not specified, the first column is used for the search criteria. Use the data in the example – *Recall Row* for the following Recall Row commands:

80.1P=3%o Recalls the row with a box# of 3.

1,1%y

1,1;8%y Recalls the row with a box# of 8.

1;5%y Recalls the row with a box# of 5 (assuming database #1 was the last database to be accessed).

80.2P=SAE30%o Recalls the row with a name of SAE30.
1,1,80.2%y

1,1,80.2;SAE20W%y Recalls the row with a name of SAE20W.

1,1,2.1;.9%y Recalls the first row with a tare weight of 0.90.

1,1,80.1;0%y Sets the 'record not found' flag because a box# of '0' does not exist in database #1.

%y ***UPDATE ROW***

Syntax **Update Row**

`2 [ , dbase# ] [, column] [;value] %y`

Arguments

dbase# Database number (1 → 250).

column Operating parameter representing a database search column with the syntax:

`< parm > . [ instance ]`

where *parm* is a operating parameter with a valid *instance*.

value Specific value or text to recall from the lookup column.

Notes Omitting *dbase#* assumes database last accessed. If no databases have been accessed, the first defined database is assumed.

See Also **%y Recall Row in Database**

%_ If Database Error

Update Row

Updates the values of all parameters in one row of a specified database with the current parameter values. If a *value* is not specified, the last row accessed is updated. When specifying a *value*, the update row command functions similar to the recall row command. The database is searched for the *value*, then the row is updated. If the *value* is not found, a new row is created at the end of the database. Note that the update row command does not set the 'record not found' flag if the search value is not found.

2%y	Updates the values of the last row accessed in last database selected.
2,3%y	Updates the values of the last row accessed in database #3.
2;5%y	Searches the last database accessed for a row with a value of 5 in the first column, then updates all other values in that row. If the search value does not exist, then a new row is created.
80.2P=A%o **2,5,80.2%y**	Searches database #5 for the value 'A' in the column for VAR#2, then updates all other values in that row. If the search value does not exist, then a new row is created.
2,2,80.2;B%y	Searches database #2 for the value 'B' in the column for VAR#2, then updates all other values in that row. If the search value does not exist, then a new row is created.

%y MAKE ROW

Syntax **Make Row**

```
3 [ , dbase# ] %y
```

Arguments

dbase#	Database number (1 → 250).

Notes Omitting *dbase#* assumes database last accessed. If no databases have been accessed, the first defined database is assumed.

See Also **%y Update Row in Database**

Make Row

Stores the current value of each database parameter as a new row appended to the end of the specified database.

3%y	Makes a new row in the last database selected.
3,1%y	Makes a new row in database #1.
3,2%y	Makes a new row in database #2.

%y PRINT DATABASE

Syntax **Print Row**

```
4 [ , dbase# ] [ ; comm ] [ . lines/page ]
[ . header ] %y
```

Arguments

dbase#	Database number (1 → 250).
comm	Communication port (1 → 4).
lines/page	Number of lines per page before form-feed character is transmitted.
header	Custom transmit number (1 → 250) to use as a custom header.

Notes Omitting *dbase#* assumes database last accessed. If no databases have been accessed, the first defined database is assumed.

Omitting *comm* assumes comm port #1.

Omitting *lines/page* assumes 60 lines per page.

Omitting *header* generates a default header. Parameter names will be used for column headings.

The header will be printed at the beginning of each new page.

See Also **%y Print Row in Database**
%y Download Database

Print Database

Transmits an entire database in fixed-width, spreadsheet format suitable for printing as a report. Data may be sent out any of the four communication ports. The number of lines per page (lines between automatic form-feed commands) can be changed. It is even possible to specify a custom transmit as the report header.

When specifying a custom transmit as a database print header, remember to select the correct comm port at P992.

4%y	Prints the last database selected out comm port 1, 60 lines/page, default page header.
4,2%y	Prints database #2 out comm port 1, 60 lines/page, default page header.
4,2;2%y	Prints database #2 out comm port 2, 60 lines/page, default page header.
4,3;1.10%y	Prints database #2 out comm port 1, 10 lines/page, default page header.
4,3;3.15.2 %y	Prints database #3 out comm port 3, 15 lines/page, using custom transmit #2 as the page header.

Example:
Printing a Database with Variable Page Length

This example shows how you can use the **[PRINT]** key to prompt for a form length before printing a database.

```
==================================
805%s10%e          Print Mc 10

MACRO #10 - PRINT PRESS
Lines/Pg.?%G       get operator entry
%[                 save entry
4,1;1.%]%y         print database
```

The following is a printout of the database in the example – *Printing a Database with Variable Page Length* using the command

```
4,1%y
```

```
BOX#      NAME         Tare
       1 SAE5W30        0.95 lb
       2 SAE10W30       0.90 lb
       3 SAE10W40       0.92 lb
       4 SAE15W40       0.89 lb
       5 SAE20W50       0.95 lb
       6 SAE30          0.90 lb
       7 SAE40          0.90 lb
       8 SAE50          0.90 lb
       9 SAE20W         0.92 lb
```

%y FIRST / LAST ROW

Syntax **Recall First Row**

`5 [ , dbase# ] %y`

Recall Last Row

`5 [ , dbase# ] ;L%y`

Arguments

dbase#	Database number (1 → 250).

Notes Omitting *dbase#* assumes database last accessed. If no databases have been accessed, the first defined database is assumed.

See Also **%y Next / Previous & Get / Recall Row in Database**

Recall First Row

`5 [ , dbase# ] %y`

Accesses the first row in the specified database. All parameters included in the database are updated with their corresponding values. If no rows are found, the 'record not found' flag is set (see *%_ If Database Error*) and the parameter values are unchanged.

5%y	Recalls the first row in the last database selected.
5,1%y	Recalls the first row in database #1.
5,2%y	Recalls the first row in database #2.

Recall Last Row

`5 [ , dbase# ] ;L%y`

Performs the same function the same as the *recall first row* command, except that the last row is accessed.

5;L%y	Recalls the last row in the last database selected.
5,1;L%y	Recalls the last row in database #1.
5,2;L%y	Recalls the last row in database #2.

The *recall last row* command can be used together with the *get row number* command to determine the number of rows in a database. For example,

```
5,1;L%y
6,1;G%y
=80.5P%o
```

copies the total number of rows in database #1 to VAR#5.

%y NEXT/PREVIOUS & GET/RECALL ROW

Syntax

Recall Next Row

`6 [ , dbase# ] %y`

Recall Previous Row

`6 [ , dbase# ] ;P%y`

Get Row Number

`6 [ , dbase# ] ;G%y`

RECALL ROW NUMBER

`6 [ , dbase# ] ; < row# > %y`

Arguments

dbase#	Database number (1 → 250).
row#	Database row number.

Notes Omitting *dbase#* assumes database last accessed. If no databases have been accessed, the first defined database is assumed.

See Also **%y First/Last Row in Database**

Example:
Copying One Database to Another

This macro routine uses the First Row and Next Row commands to copy all data from database #1 to a duplicate database #11.

```
==================================
699%s1%e            DB #: 1
700%sPARTS%e        DBNam PARTS
701%s80.1%e         Col01 Part#
702%s80.2%e         Col02 Name
703%s2.1%e          Col03 Tare

699%s11%e           DB #: 11
700%sTEMP%e         DBNam TEMP
701%s80.1%e         Col01 Part#
702%s80.2%e         Col02 Name
703%s2.1%e          Col03 Tare

MACRO #100 – COPY DATABASE
10,11%y         clear TEMP dbase
5,1%y           recall first row in DB#1

1%T             tag #1
4%_             if row not found…
  Done!%P         prompt
%N              else
  3,11%y          make row in DB#11
  6,1%y           recall next row DB#1
  1%J             jump to tag #1
%E              end if
```

Recall Next Row

`6 [ , dbase# ] %y`

Advances the database pointer to the next row. This command is typically used after a *recall first row* or *recall row* command to sequentially access each row of the database.

6%y	Recalls the next row in the last database selected.
6,1%y	Recalls the next row in database #1.
6,2%y	Recalls the next row in database #2.

Recall Previous Row

`6 [ , dbase# ] ;P%y`

Moves the database pointer backward to the previous row. It is the reverse function of the *recall next row* command.

6;P%y	Recalls the previous row in the last database selected.
6,1;P%y	Recalls the previous row in database #1.
6,2;P%y	Recalls the previous row in database #2.

Get Row Number

`6 [ , dbase# ] ;G%y`

Copies the current row number of the database pointer to the entry buffer. Used in conjunction with the *recall row number* command, it can be used as a bookmark – identifying a specific row in a database to be recalled later. It can also be used with the *recall last row* command to determine the number of rows in a database.

6;G%y	Copies the current row number of last database selected to the entry buffer.
6,1;G%y =80.9P%o	Copies the current row number of database#1 to the entry buffer, then stores the row number in VAR#9.

A row number of '0' is reported if the specified database has not yet been access, or if the result of the last search command resulted in a 'record not found' error.

Recall Row Number

`6 [ , dbase# ] ; < row# > %y`

Accesses a row in a database by moving the database pointer to specific row number. If the specified row number is not found, the 'record not found' flag is set.

6;10%y	Recalls the 10th row of last database selected.
6,3;253%y	Recalls the 253rd row of database #3.
80.9P%o%[	Recalls the row in database #1 as specified by VAR#9.
6,1;%]%y	

Example:
Finding Multiple Database Matches

This macro routine uses the Recall Row and Next Match commands to print all rows matching the entered search criteria.

```
==================================
MACRO #101 – PRINT ALL MATCHES
ENTERBOX#?%G        get operator entry
%[                  save entry in temp buffer
1,1;%]%y            recall row

1%T                 tag #1
4%_                 if record not found…
  Done!               prompt
%N                  else
  14,1%y              print row
  7,1%y               next match
  1%J                 jump to tag #1
%E                  end if
```

%y *NEXT MATCH*

Syntax — **Next Match**

`7 [ , dbase# ] %y`

Arguments

dbase# — Database number (1 → 250).

Notes — Omitting *dbase#* assumes database last accessed. If no databases have been accessed, the first defined database is assumed.

See Also — **%y Recall Row in Database**

Next Match

Finds the next row matching the original search criteria (see example – *Finding Multiple Database Matches*). This command is used after a recall row, update row, recall first/next/previous row or get row number.

The next match command always searches forward through a database, even after a *recall previous row* command.

7%y	Recalls the next match in the last database selected.
7,2%y	Recalls the next match in database #2.

%y DELETE ROW

Syntax **Delete Row**

`8 [ , dbase# ] %y`

Arguments

dbase# Database number (1 → 250).

Notes Omitting *dbase#* assumes database last accessed. If no databases have been accessed, the first defined database is assumed.

See Also **%y Clear Database**
%y Clear Column in Database

Example:
Deleting Database Rows
This macro routine uses the Recall Row and Delete Row commands to delete all rows matching the entered search criteria.

```
=================================
MACRO #102 – DELETE ALL MATCHES
ENTERBOX#?%G     get operator entry
%[               save entry in temp buffer

1%T              tag #1
1,1;%]%y         recall row

4%_              if record not found…
  Done!            prompt
%N               else
  8,1%y            delete row
  1%J              jump to tag #1
%E               end if
```

Delete Row

Deletes the last row accessed in the specified database. The database pointer then moves to the previous row, however parameter values will remain unchanged. If the first row of a database is deleted, the database pointer is set to ‘0’.

8%y	Deletes the current row in the last database selected.
8,2%y	Deletes the current row in database #2.

%y CLEAR COLUMN

Syntax **Clear Column**

`9 [ , dbase# ] ; < column > %y`

Arguments

dbase# Database number (1 → 250).

column Operating parameter representing a database search column with the syntax:

`< parm > . [ instance ]`

where *parm* is a operating parameter with a valid *instance*.

Notes Omitting *dbase#* assumes database last accessed. If no databases have been accessed, the first defined database is assumed.

See Also **%y Clear Database**

%y Delete Row in Database

Clear Column

Clears all data in a specified database column. No rows are deleted. In columns containing numeric data, all data in the specified column is set to zero (0). In columns containing string data, all data in the specified column is set to an empty (null) string.

9;80.2%y	Clears the VAR#2 column in the last database selected.
9,2;2.2%y	Clears the TARE SCALE#2 column in database #2.
9,3;6.1%y	Clears the NET TOTAL SCALE #1column in database #3.

%y CLEAR DATABASE

Syntax **Clear Database**

`10 [ , dbase# ] %y`

Arguments

dbase# Database number (1 → 250).

Notes Omitting *dbase#* assumes database last accessed. If no databases have been accessed, the first defined database is assumed.

See Also **%y Delete Row in Database**

%y Clear Column in Database

Clear Database

Deletes all rows in the specified database. This command can be used to ensure a database is empty before making new rows or uploading database information.

10%y	Deletes all rows in the last database selected.

10,2%y	Deletes all rows in database #2.
DB#toCLR?%G **%[** **10,%]%y**	Prompts for a database# entry, then deletes all rows in the database specified by the entry.

%y ***SET DATABASE***

Syntax **Set Database**

`11 , < dbase# > %y`

Arguments

dbase# Database number (1 → 250).

Set Database

Sets the currently selected database. This command rarely used since all other database commands allow you to specify the current database within the command syntax.

11,1%y	Sets database #1 as the current database.
11,2%y	Sets database #2 as the current database.

%y ***SET COLUMN***

Syntax **Set Column**

`12 [ , dbase# ] [ ; . column ] %y`

Arguments

dbase# Database number (1 → 250).

column Operating parameter representing a database search column with the syntax:

`< parm > . [ instance ]`

where *parm* is a operating parameter with a valid *instance*.

Notes Omitting *dbase#* assumes database last accessed. If no databases have been accessed, the first defined database is assumed.

Set Column

Sets the search column for which subsequent database commands will use. Once a column is set, it will remain in effect as the search column until changed by another *set column* command or another search command. The *set column* is rarely used since all other database commands requiring a column argument allow you to specify a column within the command syntax.

12;80.2%y	Sets the search column as VAR#2 in the current database.
12,2;2.3%y	Sets the search column as TARE SCALE#3 in database #2.
12%y	Sets the first column in the current database as the search column.

%y ***DOWNLOAD***

Syntax **Download Database**

`13 [ , dbase# ] [ ; comm ] [ . format ] [ . time/date ] %y`

Arguments

dbase# Database number (1 → 250).

comm Communication port (1 → 4).

format Transmit database with or without upload information:
Do not include upload information
Include upload information

time/date Transmit time and/or date parameters as a number or as text:
Transmit as a number
Transmit as text

Notes Omitting *dbase#* assumes database last accessed. If no databases have been accessed, the first defined database is assumed.

Omitting *comm* assumes port 1.

Omitting *format* or *time/date* assumes a selection of 0.

See Also **%y Print Database**
%y Upload New Database

Download Database

Transmits all database rows out a designated communication port in a comma-delimited ASCII text format.

13,3%y	Transmits database #3 out comm port 1.
13,3;1%y	Transmits database #3 out comm port 1.
13,4;3%y	Transmits database #4 out comm port 3.
13%y	Transmits the last database selected out comm port 1.
13;2%y	Transmits the last database selected out comm port 2.
13,2;2.1%y	Transmits database #2 out comm port 2 with upload information, time/date sent as a number.
13,2;2.0.1%y	Transmits database #2 out comm port 2 without upload information, time/date sent as text.
13,2;2.1.1%y	Transmits database #2 out comm port 2 with upload information, time/date sent as text.

All weight values are transmitted in the default units specified in the setup mode at P150. All floating point data (including weight values) are downloaded using full precision values (no rounding). String variable text is encapsulated in double-quotes (" ").

A *format* code can be specified after the comm port number to include upload information at the beginning and end of the transmission. This allows you to save the downloaded database as a text file that can later be uploaded to restore or transfer database information. Consider the following sample database:

PRODUCT#	NAME	TARGET	PRE-ACT	TIME/DATE
1	Aggregate	5000	250	958662865
2	Cement	1000	100	958662875
3	Fly Ash	500	50	958662883
4	Recycle	1000	250	958662899
5	River Rock	2000	250	958662908

Assuming this is database #1, the command

`13,1;2%y`

will produce the following download out comm port 2

When you include upload information in a download output, remember that you must use the same comm port if you later restore (upload) the database file.

```
1,”Aggregate”,5000,250,958662865
2,”Cement”,1000,100,958662875
3,”Fly Ash”,500,50,958662883
4,”Recycle”,1000,250,958662899
5,”River Rock”,2000,250,958662908
```

Including the format argument

`13,1;2.1%y`

produces the following download out comm port 2 (including upload information)

```
16,1;2%y
1,”Aggregate”,5000,250,958662865
2,”Cement”,1000,100,958662875
3,”Fly Ash”,500,50,958662883
4,”Recycle”,1000,250,958662899
5,”River Rock”,2000,250,958662908
ENDofDB
```

Note that the upload command 16,1;2%y includes the comm port information. Thus the database file must be uploaded via comm port 2.

Although downloading the time/date in text format produces a more readable output, using the number format is often more advantageous if exporting to a PC spreadsheet. See *Time & Date (Mode 11) on page 7-12* for more information.

Including the time/date argument

`13,1;2.1.1%y`

changes any time and/or date parameter output to text format

```
16,1;2%y
1,”Aggregate”,5000,250,03:14:25 pm 05/18/00
2,”Cement”,1000,100,03:14:35 pm 05/18/00
3,”Fly Ash”,500,50,03:14:43 pm 05/18/00
4,”Recycle”,1000,250,03:14:59 pm 05/18/00
5,”River Rock”,2000,250,03:15:08 pm 05/18/00
ENDofDB
```

Note that time/date text values are <u>not</u> encapsulated in double-quotes. When using variables to represent time/date values, the download format will reflect the selection at P688 of the setup mode (time only, date only, or time & date).

%y PRINT ROW

<u>Syntax</u> **Print Row**

`14 [ , dbase# ] [ ; comm ] %y`

<u>Arguments</u>

dbase# Database number (1 → 250).

comm Communication port (1 → 4).

<u>Notes</u> Omitting *dbase#* assumes database last accessed. If no databases have been accessed, the first defined database is assumed.

Omitting *comm* assumes port 1.

<u>See Also</u> **%y Print Database**

Print Row

Transmits the currently selected row of a database in fixed-width, spreadsheet format similar to the *Print Database* command. This allows you to print selective rows rather than the entire database (see example – *Printing Database Rows*).

Example:
Printing Database Rows

This macro routine prints all rows in a database that fall within the range of dates entered in time/date VAR#1 and VAR#2. VAR#3 is used as a time/date column in database #2.

```
==================================
MACRO #100 – PRINT ALL MATCHES
StartTime?%G          get operator entry
=80.1P%o              save entry
End  Time?%G          get operator entry
=80.2P%o              save entry

5,2%y                 recall first row

1%T                   tag #1
4%_                   if record not found…
  Done!                 prompt
%N                    else
  %{
  80.3P>=80.1P%o        if >= start time, and
  80.3P<=80.2P%o        if <= end time…
    4,2%y                 print row
  %E                    end if

  6,2%y                 get next row
  1%J                   jump to tag #1
   %}
%E                    end if
```

14,1%y	Transmits the currently selected row in database #1 out comm port 1.
14,1;1%y	Transmits the currently selected row in database #1 out comm port 1.
14,2;3%y	Transmits the currently selected row in database #2 out comm port 3.
14%y	Transmits the currently selected row of the last database selected out comm port 1.
14;3%y	Transmits the currently selected row of the last database selected out comm port 3.

%y *PRINT DATABASE ERRORS*

Syntax **Print Database Errors**

`15 [ , dbase# ] [ ; comm ] %y`

Arguments

dbase# Database number (1 → 250).

comm Communication port (1 → 4).

Notes Omitting *dbase#* assumes database last accessed. If no databases have been accessed, the first defined database is assumed.

Omitting *comm* assumes port 1.

Print Database Errors

Prints database rows containing corrupted data. This is a diagnostic tool used to check the integrity of stored data. A checksum is stored with each row of a database. Each time the row is accessed, the data's checksum is recalculated and compared with the stored checksum. If the checksums do not match, the data is considered corrupt.

The print format is similar to the *%y Print Database* command, beginning with a header line identifying the columns, followed by corrupt rows and ending with a summary of the number of corrupt rows found.

15,3%y	Transmits errors for database #3 out comm port 1.
15,4;3%y	Transmits errors for database #4 out comm port 3.
15%y	Transmits errors for the last database selected out comm port 1.

15;2%y	Transmits errors for the last database selected out comm port 2.

%y ***UPLOAD NEW***

Syntax **Upload New**
`16 [ , dbase# ] [ ; comm ] %y`

Arguments

dbase# Database number (1 → 250).

comm Communication port (1 → 4).

Notes Omitting *dbase#* assumes database last accessed. If no databases have been accessed, the first defined database is assumed.

Omitting *comm* assumes port 1.

See Also **%y Upload Update Database**
%y Download Database

Upload New

A comm port must be specified for the database upload file. If upload data is received by a non-specified port, it will not be processed and the message Waiting... will be displayed.

Initiates the upload of new data rows to a database. New data is appended to existing data rows. Data must be sent in a comma-delimited ASCII text format with fields matching the database structure and column data types. Each data row must be terminated with a carriage return <CR>. The last line of the upload file must be the text **`ENDofDB`** followed by a carriage return <CR>, *or* an end-of-file character <EOF> (**[CTRL]+Z** on a PC keyboard).

Upload data must be received on the specified comm port. Once the upload command is executed, the prompt `Waiting...` is displayed until the first data row is received on the specified port. A line count is then displayed and incremented with each row received. Finally, `Done!` is displayed when the **`ENDofDB`** or <EOF> character is received.

16,3%y	Upload database #3 via comm port 1.
16,3;1%y	Upload database #3 via comm port 1.
16,4;3%y	Upload database #4 via comm port 3.

If the upload count exceeds 99999 then the displayed count becomes ####x where #### are the four most significant digits and ‘x’ represents a place holder for the least significant digits.

The following is an example of an upload file format:

```
16,1;1%y
1,”Aggregate”,5000,250,958662865
2,”Cement”,1000,100,958662875
3,”Fly Ash”,500,50,958662883
4,”Recycle”,1000,250,958662899
5,”River Rock”,2000,250,958662908
ENDofDB
```

The *Upload New* command is often used to restore database information acquired using the *%y Download Database* command. Ensure an empty database before uploading by including the *%y Clear Database* command at the beginning of the transmission as shown below:

```
10,1%y
16,1;1%y
1,”Aggregate”,5000,250,958662865
2,”Cement”,1000,100,958662875
3,”Fly Ash”,500,50,958662883
4,”Recycle”,1000,250,958662899
5,”River Rock”,2000,250,958662908
ENDofDB
```

%y UPLOAD UPDATE

Syntax **Upload Update**

`17 [ , dbase# ] [ ; comm ] %y`

Arguments

dbase# Database number (1 → 250).

comm Communication port (1 → 4).

Notes Omitting *dbase#* assumes database last accessed. If no databases have been accessed, the first defined database is assumed.

Omitting *comm* assumes port 1.

See Also **%y Upload New Database**

Because the Upload Update command must search the entire database before creating a new row, this method of upload is significantly slower than the Upload New command, especially for larger databases.

Upload Update

Initiates the upload of new or existing data rows to a database. This command operates similar to the *Upload New Database* command with one exception. As new each row is received, the *Upload Update* command searches the existing rows for a match of the first column data. If a match is found, all other columns in that row are updated with the new data. If a match is not found, the new row is appended to the end of the database. This method prevents the creation of duplicate records.

As with the *Upload New* command, data must be sent in a comma-delimited ASCII text format with fields matching the database structure and column data types. Each data row must be terminated with a carriage return <CR>. The last line of the upload file must be the text **ENDofDB** followed by a carriage return <CR>, *or* an end-of-file character <EOF> (**[CTRL]+Z** on a PC keyboard).

Upload data must be received on the specified comm port. Once the upload command is executed, the prompt **Waiting...** is displayed until the first data row is received on the specified port. A line count is then displayed and incremented with each row received. Finally, **Done!** is displayed when the **ENDofDB** or <EOF> character is received.

17,3%y	Upload database #3 via comm port 1.
17,3;1%y	Upload database #3 via comm port 1.
17,4;3%y	Upload database #4 via comm port 3.

If the upload count exceeds 99999 then the displayed count becomes ####x where #### are the four most significant digits and 'x' represents a place holder for the least significant digits.

%y *SORT DATABASE*

Syntax **Sort Database**

`18 [ , dbase# ] [ ; parm ] [ , parm ] %y`

Arguments

dbase# Database number (1 → 250).

parm Operating parameter with the syntax:

`< parm > [ . instance ]`

where *parm* is a displayable operating parameter with a valid *instance*.

Notes Omitting *dbase#* assumes database last accessed. If no databases have been accessed, the first defined database is assumed.

Data can be sorted in descending order by preceding *parm* with a decimal point.

Sort Database

Sorts the rows of a database based on the sort criteria. Data can be sorted in ascending or descending order in any database column. Multiple parameters can be specified for sub-sort columns. Rows with exact matches are not sorted further unless a sub-sort column is specified.

18%y	Sorts the first column of the last selected database in ascending order.
18,7%y	Sorts the first column of database #7 in

	ascending order.
18,4;80.4%y	Sorts the column for VAR#4 of database #4 in ascending order.
18,4;.80.4%y	Sorts the column for VAR#4 of database #4 in descending order.
18;.23.1%y	Sorts the column for the RATE of scale #1 of the last selected database in descending order.
18,2;80.1,.80.2%y	Sorts the column for VAR#1 of database #2 in ascending order with a sub-sort of the column for VAR#2 in descending order.

The time it takes to sort a database depends on the number of rows in the database and the complexity of the sort criteria. Macro execution does not resume until sorting is complete. During the sorting routing, the number of rows sorted increments on the display. Pressing **[CLR]** will abort the sorting process.

%y ***DATABASE AUTO-TEST***

Syntax **Database Auto-Test**

`19; < 0 | 1 > %y`

Arguments

0	Disables Auto-Test
1	Enables Auto-Test

Notes Database Auto-Test is disabled at power-up.

Database Auto-Test

Tests the integrity of the database after each database command is executed. If enabled, the auto-test adds the number of used and unused bytes and compares it to the number of bytes available for the database. If the total does not match, an error is displayed and logged in the macro debug.

This command is a diagnostic tool that should only be enabled to trouble shoot database corruption problems. Enabling the auto-test will slow the execution of database macro commands.

19;1%y	Enables auto-test.
19;0%y	Disables auto-test.

%w DSD DATABASE FUNCTIONS

Syntax

Lookup Data by ID#
`I [ id# ] %w`

Get Number of Existing Rows
`N%w`

Get Lowest ID# in Database
`L%w`

Get Highest ID# in Database
`H%w`

Arguments

id# DSD ID#

Notes This command only works with the DSD database.

See Also Data Storage Device (DSD) **(page 6-13)**

Lookup Data by ID#

`I [ id# ] %w`

Searches the database for the data row containing the specified ID#. If found, parameter P64 will reference this row's data. This statement can be used in a macro IF conditional statement to detect whether the row was found or not (4%_).

Get Number of Existing Rows

`N%w`

Returns the number of database rows which are in existence. The result can range between zero and the maximum number of rows defined at P594. The result is placed in the entry buffer.

Get Lowest ID# in Database

`L%w`

Searches the database and finds the lowest ID# that exists. The result can range between zero and the maximum ID# (999999). The result is placed in the entry buffer.

Get Highest ID# in Database

`H%w`

Searches the database and finds the highest ID# that exists. The result can range between zero and the maximum ID# (999999). The result is placed in the entry buffer.

%z ZERO

Syntax

Zero
%z

Absolute Zero
< *value* > A%z

Relative Zero
< *value* > R%z

Arguments

scale# Select from the following scale numbers:

0 Current scale
1 Scale 1
2 Scale 2
3 Scale 3
4 Scale 4
5 Scale 5
6 Scale 6
7 Scale 7
8 Scale 8

value Zero weight value.

See Also

%t Tare
%` Scale Select
%- Perform Scale Specific Function

Zero

%z

Establishes a gross zero reference for the current scale.

%z Performs a motion delayed auto-zero (gross = 0).

A *zero* command must originate from the gross, net, tare, or any of the accumulation parameters (parameters 0P → 9P). The gross mode is automatically selected after performing a zero.

An Absolute Zero value references the last calibrated zero; it does not look at the re-zero value acquired by pressing the [ZERO] key.

Absolute Zero

< *value* > A%z

Changes the calibrated zero reference by an absolute offset.

10.5A%z	Increases the original calibrated zero reference by 10.5 units.
-21A%z	Decreases the original calibrated zero reference by 21 units.
0A%z	Restores the original calibrated zero reference.

This command operates similar to performing a "Zero Only" calibration. For example, if 100 kg of dead-load was added to a scale platform, the command

```
100A%z
```

increases the calibrated zero reference by 100 kg to compensate for the dead-load. This change in the zero reference is permanent, even in the event of a power loss. The original calibrated zero reference can be restored with the command

```
0A%z
```

The *Absolute Zero* command always uses the original calibrated zero reference established when adding or subtracting an absolute value. Thus the original zero reference can always be restored.

Relative Zero

< *value* > R%z

Adjusts the re-zero reference by adding or subtracting an offset value.

5R%z	Increases the re-zero reference by 5 units.
-1R%z	Decreases the re-zero reference by 1 unit.
0R%z	Does not change the re-zero reference.

This command operates similar to the zero tracking feature. Subsequent *Relative Zero* commands have a cumulative effect on the displayed zero reference. Changes to the re-zero weight are permanent, even in the event of a power loss. For example, if the **[ZERO]** key is pressed to establish a displayed gross weight of zero (0), then the command

```
10R%z
```

results in a displayed gross weight of –10. Issuing this command a second time results in a displayed gross weight of –20.

The Relative Zero command can be used to make changes to the re-zero reference in dynamic applications such as conveyor belt scales. Here, the zero reference is not an instantaneous value, rather it is an average of the dead-load of the conveyor belt over it's entire length. The average weight of the belt could be acquired using the *%+ Averaging* command. The average zero value could then be assigned as the *Relative Zero* value

```
15.1P%oR%z
```

This process could be repeated as desired to incorporate a dynamic zero-tracking routine for the conveyor belt.

%{ START GROUP

Syntax **Start Group**

%{

See Also **%} End Group**
Boolean Logic

Start Group

Begins a new level of nesting for "IF" statements, or groups Boolean logic statements. Each *Start Group* command should have a corresponding *End Group* command later in the macro.

%| OR

Syntax **Or**
%|

See Also **Boolean Logic**

Or

%|

Provides a logical "OR" between two or more "IF" statements.

%} END GROUP

Syntax **End Group**
%}

See Also **%{ Start Group**
Boolean Logic

End Group

Ends a level of nesting for "IF" statements, or group of Boolean logic statements. Each *End Group* command should have a corresponding *Start Group* command earlier in the macro.

BOOLEAN LOGIC

The 60 Series instruments are capable of making simple and complex decisions based on the results of conditional macro commands. Each conditional statement is evaluated and determined to be either TRUE or FALSE. Program execution is then allowed to "branch" in one of two directions depending on the outcome of this evaluation.

All conditional macro commands follow a few fundamental rules. From these simple rules you can build virtually any conceivable logic algorithm.

- <u>EVERY</u> "IF" statement (or group of "IF" statements) should have a corresponding "END IF". Failure to follow this rule could yield unpredictable results.
- If an "IF" statement is TRUE, program execution resumes with the macro command immediately following the "IF" statement. Subsequent macro commands are executed in sequence until a corresponding "ELSE" command is encountered. All macro commands after the "ELSE" (if present) are skipped until the corresponding "END IF" is encountered.
- If an "IF" statement is FALSE, program execution resumes with the macro command immediately following the first corresponding "ELSE" or "END IF" command.
- Two or more sequential "IF" statements constitute a logical 'AND'. This means that all "IF" statements in sequence must be TRUE for the 'AND' condition to be TRUE.
- A logical 'OR' can be created by separating two or more sequential "IF" statements with an "OR" macro command. If any one of the "IF" statements is TRUE, the entire 'OR' condition is considered to be TRUE. When the first TRUE condition is encountered, macro execution skips to the first command following the last "IF" statement of the 'OR' condition. It does not waste time evaluating additional "IF" statements since the 'OR' condition was already found to be TRUE.
- *Start Group* and *End Group* braces must be used for nested "IF" statements. This ensures that the nested "ELSE" and "END IF" commands will not correspond with the "IF" statement outside the nest.
- *Start Group* and *End Group* braces can be used to group combinations of 'AND' and 'OR' statements to change the standard logic conventions.

THE "IF" STATEMENT

The most basic "IF" statement uses a single macro comparison command to determine if a condition is TRUE as shown below. Macro commands that are executed are shown in gray.

The following example demonstrates macro execution for a TRUE condition. All macro commands following the "IF" statement are executed.

```
1%A       activate setpoint #1

1%O       if setpoint #1 is on…
  %S           sound beeper
  SP#1 Activ%P     prompt
%E           end if

< Next macro command(s) >
```

This next example shows the execution of the same "IF" statement for a FALSE condition. Note that the commands after the "IF" statement are not executed. Since the condition if FALSE, execution skips to the “END IF” command.

```
1%D       deactivate setpoint #1

1%O       if setpoint #1 is on…
  %S           sound beeper
  SP#1 Activ%P     prompt
%E            end if

< Next macro command(s) >
```

THE "ELSE" (IF NOT) STATEMENT

The “ELSE” statement allows you to execute macro commands if the condition of a comparison if FALSE.

When an “ELSE” command is encountered after a TRUE conditional statement, all macro commands following the “ELSE” command are skipped up to the corresponding “END IF”.

```
1%A       activate setpoint #1

1%O       if setpoint #1 is on…
  %S           sound beeper
  SP#1 Activ%P     prompt
%N            else
  SP#1 Deact%P
%E            end if

< Next macro command(s) >
```

If the condition is FALSE, macro execution skips ahead and resumes with the first command after the “ELSE”.

```
1%D       deactivate setpoint #1

1%O       if setpoint #1 is on…
  %S           sound beeper
  SP#1 Activ%P     prompt
%N            else
  SP#1 Deact%P
%E            end if

< Next macro command(s) >
```

THE 'AND' CONDITION

A logic ‘AND’ condition is achieved by using sequential macro comparison commands. A TRUE condition requires that all conditions in the sequence are TRUE.

```
1-3%A         activate setpoints #1,#2,#3

1%O       if setpoint #1 is on, and
2%O       if setpoint #2 is on, and
3%O       if setpoint #3 is on…
  %S           sound beeper
  SPTs Activ%P     prompt
%N            else
  SPT  Error%P
%E            end if
```

```
< Next macro command(s) >
```

If one of the comparison statements are FALSE, macro execution skips ahead and resumes with the first corresponding “ELSE” or “END IF”.

```
1,3%A         activate setpoints #1 & #3
2%D        deactivate setpoint #2

1%O        if setpoint #1 is on, and
2%O        if setpoint #2 is on, and
3%O        if setpoint #3 is on…
  %S            sound beeper
  SPTs Activ%P     prompt
%N            else
  SPT  Error%P
%E            end if

< Next macro command(s) >
```

THE 'OR' CONDITION

A logic ‘OR’ condition is achieved by using the %| macro command to separate sequential “IF” statements. A TRUE condition only requires one of the “IF” statements to be TRUE. When the first TRUE condition is encountered, macro execution skips to the first command following the last “IF” statement of the ‘OR’ condition.

```
1,3%D         deactivate setpoints #1 & #3
2%A        activate setpoint #2

1%O        if setpoint #1 is on
%|         OR
2%O        if setpoint #2 is on
%|         OR
3%O        if setpoint #3 is on…
  %S            sound beeper
  SPTs Activ%P     prompt
%N            else
  SPT  Error%P
%E            end if

< Next macro command(s) >
```

If none of the conditional statements are TRUE, then the condition as a whole is considered to be FALSE.

```
1-3%D         deactivate setpoints #1,#2,#3

1%O        if setpoint #1 is on
%|         OR
2%O        if setpoint #2 is on
%|         OR
3%O        if setpoint #3 is on…
  %S            sound beeper
  SPTs Activ%P     prompt
%N            else
  SPT  Error%P
%E            end if

< Next macro command(s) >
```

Grouping "If" Statements

Complex conditional statements can be created by combining 'AND' and 'OR' operations. In doing so, always consider the rules for these operations. You may find it necessary to use braces to group "IF" statements in order to achieve the desired results.

Consider the following example:

```
1-2%A         activate setpoints #1 & #2

1%O       if setpoint #1 is on, and
2%O       if setpoint #2 is on…
%|        OR
3%O       if setpoint #3 is on, and
4%O       if setpoint #4 is on…
  %S           sound beeper
  SPTs Activ%P     prompt
%N            else
  SPT  Error%P
%E            end if

< Next macro command(s) >
```

The intention is to create a TRUE condition if setpoints #1 and #2 are active, 'OR' if setpoints #3 and #4 are active. Since setpoints #1 and #2 are indeed active, we would expect the condition to be TRUE. However, this is not the case. Remember the rule for 'OR' conditions – when the first TRUE condition is encountered, macro execution skips to the first command following the last "IF" statement of the 'OR' condition. Here, the last statement of the 'OR' condition is 3%0. Thus the macro resumes with 4%0 which is FALSE, making the entire condition FALSE.

In order to make this condition function as intended, the lines

```
3%O       if setpoint #3 is on, and
4%O       if setpoint #4 is on…
```

must be treated collectively as a single condition. This can be accomplished by "grouping" these lines in braces.

```
%{
3%O       if setpoint #3 is on, and
4%O       if setpoint #4 is on…
%}
```

Now, both setpoints #3 and #4 must be active for this single condition to be TRUE.

```
1-2%A         activate setpoints #1 & #2

1%O       if setpoint #1 is on, and
2%O       if setpoint #2 is on…
%|        OR
%{
3%O       if setpoint #3 is on, and
4%O       if setpoint #4 is on…
%}
```

```
  %S            sound beeper
  SPTs Activ%P     prompt
%N            else
  SPT  Error%P
%E            end if

< Next macro command(s) >
```

More complex conditional statements can be created by nesting groups of conditions.

```
1-4%A         activate setpoints #1,#2,#3,#4

1%O       if setpoint #1 is on, and
2%O       if setpoint #2 is on, and
%{
  3%O         if setpoint #3 is on, and
  4%O         if setpoint #4 is on…
  %|
  %{
  5%O         if setpoint #5 is on, and
  6%O         if setpoint #6 is on…
  %}
%}
  %S            sound beeper
  SPTs Activ%P     prompt
%N            else
  SPT  Error%P
%E            end if

< Next macro command(s) >
```

This condition yields TRUE if setpoints #1 and #2 are active ‘AND’ if setpoints #3 and #4 ‘OR’ #5 and #6 are active.

NESTED "IF" STATEMENTS

It is possible to have up to 254 levels of nesting.

Nested “IF” statements are required when one “IF” statement is contained within another. Nesting uses braces to keep track of corresponding “ELSE” and “END IF” commands. Without braces, macro execution may not function as intended.

INCORRECT NESTING TECHNIQUES

In the following example, the intention is to make sure setpoint #1 is active before a filling process begins. If so, another (nested) “IF” statement will transmit the product ID# if one was assigned. However, the program does not execute exactly as expected if both conditions are TRUE.

```
1%”       select comm1
1%A       activate setpoint #1
80.1P=5%o    VAR#5 (Product ID) = 5

1%O       if setpoint #1 is on…
  StartFill%P      prompt

  80.1P!=0%o      if PRODUCT ID not 0…
    Product ID %$       send text
    80.1P%o%$        send PRODUCT ID
  %E           end if
```

```
  SPT#1 is ON%$    send text
  2%A          activate fill output
%N        else
  SPT#1 is OFF%$   send text
%E        end if

< Next macro command(s) >
```

Note that every line of code was executed – even the line stating that setpoint #1 is off! To understand why, look at the first "IF" statement (1%O). When this statement is TRUE, it will execute all subsequent macro commands up to the first "ELSE" or "END IF" command it encounters. In this case, the first "END IF" command it encounters is the one corresponding to the product ID# (`80.1P!=0%o`). This terminates the first "IF" statement before it reaches the "ELSE" statement. Thus the "ELSE" is ignored and the prompt is send indicating that setpoint #1 is off.

Now consider what happens if setpoint #1 is off.

```
1%"       select comm1
1%D       deactivate setpoint #1
80.1P=5%o    VAR#5 (Product ID) = 5

1%O       if setpoint #1 is on…
  StartFill%P      prompt

  80.1P!=0%o       if PRODUCT ID not 0…
    Product ID %$       send text
    80.1P%o%$       send PRODUCT ID
  %E          end if

  SPT#1 is ON%$    send text
  2%A          activate fill output
%N        else
  SPT#1 is OFF%$   send text
%E        end if

< Next macro command(s) >
```

Once again, the wrong "END IF" command is used to terminate the first "IF" statement. This time the prompt is sent indicating that setpoint #1 is on. Even more concerning is the fact that the fill output was activated!

CORRECT NESTING TECHNIQUES

In order to avoid the problems encountered in the previous example, Start Group and End Group braces should be used to ensure "ELSE" and "END IF" commands will only correspond to their respective "IF" statements. The following example will operate as intended.

```
1%"       select comm1
1%A       activate setpoint #1
80.1P=5%o    VAR#5 (Product ID) = 5

1%O       if setpoint #1 is on…
  %{
  StartFill%P      prompt

  80.1P!=0%o       if PRODUCT ID not 0…
    Product ID %$       send text
```

```
    80.1P%o%$        send PRODUCT ID
  %E           end if

  SPT#1 is ON%$    send text
  2%A          activate fill output
  %}
%N        else
  SPT#1 is OFF%$   send text
%E        end if

< Next macro command(s) >
```

Note how the commands between the *Start Group* and *End Group* braces are treated as the TRUE condition of the first “IF” statement. The second “IF” statement and corresponding “END IF” command are executed independent of the first “IF”. The “ELSE” now corresponds to the first “IF” statement.

Here is the same code as executed with a FALSE condition for setpoint #1.

```
1%”       select comm1
1%D       deactivate setpoint #1
80.1P=5%o    VAR#5 (Product ID) = 5

1%O       if setpoint #1 is on…
  %{
  StartFill%P      prompt

  80.1P!=0%o       if PRODUCT ID not 0…
    Product ID %$       send text
    80.1P%o%$        send PRODUCT ID
  %E           end if

  SPT#1 is ON%$    send text
  2%A          activate fill output
  %}
%N        else
  SPT#1 is OFF%$   send text
%E        end if

< Next macro command(s) >
```

Again, note how the braces control macro execution. The first “IF” statement is FALSE. Since a *Start Group* brace immediately follows, all subsequent commands are skipped until the corresponding *End Group* brace is encountered.

POINTERS

Macro pointers use the value one variable to access or change the value of another variable or operating parameter. This allows you to write concise routines to manipulate multiple data registers.

VARIABLE POINTERS

A pointer variable uses the number it stores as a "pointer" to address another parameter. Any variable can be a pointer. For example,

80.21P=1%o

is a standard variable assignment that assigns a value of 1 to VAR#21. In order to use VAR#21 as a pointer, use a lower-case 'p' in place of the upper-case 'P'. Thus,

80.21p=5%o

"points" to VAR#1 and assigns it a value of 5. Note that VAR#21 still contains a value of 1. Similarly,

80.21P=2%o

80.21p=5%o

assigns a value of 2 to VAR#21, then "points" to VAR#2 and assigns it a value of 5. We could continue incrementing the value of VAR#21 in this manner and initialize all variables 1-20 with a value of 5. Of course this would result in twice as many lines of code as compared to simply assigning these 20 variables a value of 5 directly. Instead, consider the following example:

```
80.21P=1%o     initialize VAR#21

%T         tag
80.21P<21%o    if the value of VAR#21 is less than 21…
  80.21p=5%o         assign a value of 5 to the addressed VAR
  80.21P+=1%o        increment the value of VAR#21
  %J             jump to tag
%E         end if
```

Here, only a few lines of code are required to initialize all 20 variables. In fact, this same routine could be used to initialize 200 variables simply by referencing VAR#201 instead of VAR#21.

```
80.201P=1%o    initialize VAR#201

%T         tag
80.201P<201%o      if the value of VAR#201 is less than 201…
  80.201p=5%o        assign a value of 5 to the addressed VAR
  80.201P+=1%o       increment the value of VAR#201
  %J             jump to tag
%E         end if
```

Pointers can also be used in comparison commands and even in formatting commands. The next example demonstrates this in a routine used to print the value of any variable 1-100 that contains a value greater than zero (0). Here, the pointer variable #101 is used both as a truck ID# and as a data register to store the truck's WEIGH-IN weight. Note how VAR#11 is formatted to transmit it's value as the truck ID# and then reformatted to send it's addressed value as the WEIGH-IN weight.

```
80.101P=1%o                       TRUCK ID# = 1
1%"                               select comm1
```

```
ID#    WEIGH-IN%$                  send text
13,10%&                            send <CR><LF>
---------------%$                  send text
13,10%&                            send <CR><LF>

1%T                                tag #1
80.101P<101%o                      if TRUCK ID# < 101...
  %{
  80.101p>0%o                        if WEIGH-IN > 0 lbs...
    80.101.16387P%o   %$               send TRUCK ID#
    80.101.16384p%o%$                  send WEIGH-IN weight
    13,10%&                            send <CR><LF>
  %E                                 end if

  80.101P+=1%o                       increment TRUCK ID#
  1%J                                jump to tag #2
  %}
%E                                 end if

---------------%$                  send text
13,10%&                            send <CR><LF>
End of Data%$
13,10,10%&                         send <CR><LF>
```

NON-VARIABLE POINTERS

A variable can be used to address a non-variable parameter. The syntax is similar to the variable pointer except that a lower case 'q' is used to identify the pointer rather than the lower-case 'p'. For example,

```
80.1P=50.1%o
```

is a standard variable assignment that assigns a value of 50.1 to VAR#1. The command

```
80.1q=500%o
```

therefore assigns a value of 500 to parameter 50.1P, the PDIO 'A' parameter for channel #1. If channel #1 was configured as a frequency output, this would result in an assignment of a 500Hz output. Similarly,

```
80.1P=2.1%o
80.1q=10%o
```

will assign a tare weight of 10 to scale #1 (parameter 2.1P).

INTERRUPT MACROS

Macros can be set for immediate execution at P9992 of the setup mode.

Macros can be configured to interrupt the execution of other macros. A macro can be set for immediate execution by setting P9992 of the setup mode to `Invok Immed` (invoke immediate) instead of the default selection `Invok Std` (invoke standard). When set to Invoke Standard, a macro is placed on the macro stack if another macro is running. When set to be invoked immediately, a macro is able to interrupt the execution of any other macro. This feature could be used as an emergency stop routine to immediately respond to an input and deactivate all setpoint control outputs.

INVOKING AN INTERRUPT MACRO

Interrupt macros can only be invoked for immediate execution An interrupting macro is essentially 'called' when invoked, eventually returning to the next macro command of interrupted macro. The following example shows the sequence of macro execution if a macro is interrupted. Note that the interruption could have occurred after any command in the interrupted macro.

```
INTERRUPTED MACRO

80.1P=0.0P%o
80.2P=1.0P%o
80.3P=2.0P%o

<<<INTERRUPT>>>

80.4P=3.0P%o
80.5P=6.0P%o


INTERRUPTING MACRO

1%Q
2%A
```

Although a macro interrupt will stop the execution another macro, it will not stop the execution of individual macro commands. In other words, the last macro command will perform it's function before the interrupt macro is executed. There are a few exceptions to this rule. For example, the *%G Get Operator Entry* command halts macro execution until the **[ENTER]** key is pressed. This would obviously be a problem if the operator walked away from the scale without completing an entry in process. The %G command would not have completed its function and the interrupt macro would be put on hold indefinitely. To avoid this potential problem, entry commands and various other operator interface and motion delayed macro commands are allowed to be terminated before completing their function. Table 9-10 lists all macro commands allowed to be interrupted immediately without completing their functions.

Table 9-10: Interruptible Macro Commands

COMMAND	DESCRIPTION
%G	Get Entry
%K	Get Entry From 4X20 VFD
%n	Get Numeric Entry
%P	Pause
%Y	If Yes (Enter)
%W	Wait For Keypress
W%r	Wait For A/D Interval
%t	Tare
%z	Zero
%p	Print

Example:
Repeating An Interruptible Macro Command After Interrupt

If a macro is interrupted during an entry such as the %G command, the entry will be lost. This macro routine will repeat the target entry routine if it was aborted due to a macro interrupt.

```
================================
=
MACRO #1 – ENTER TARGET
1%T              tag #1
EnterTargt%G     get operator entry

%/               if macro interrupted…
  1%J              jump to tag#1
%E               end if

=80.1P%o         save entry
```

? If the macro debug table is accessed without entering the setup access code, it will not be possible to print the table.

If the macro debug table is accessed using the limited access code and limited access is enabled at P50000, then it will not be possible to view or print the table.

An internal flag is set if any of the interruptible macro commands in are interrupted. This flag can be tested with *the %/ If Macro Interrupted* command. This allows you to determine if a macro command was not executed when the interrupted macro continues and repeat the command if required (see example – *Repeating an Interruptible Macro Command After Interrupt*).

*NOTE: The **%b** - Pause and perform Sample Routine - and the **%,** - Pause until No-Motion - should not be used with an interrupt macro.

Keep in mind that any of the commands listed above will reset the Interrupt Flag for the If Interrupted Command. This is part of it's normal function. Therefore, if any of the above commands are used in the interrupting macro, the Interrupted Flag would be reset and the original macro would be unable to discern that an interrupt had occurred.

Since an Interrupt Macro can be invoked at any point within a current Macro, it is possible to have data in the Entry Buffer at the time of the interrupt. For this reason it is advisable to use the **%[** - Save Entry - at the beginning of the Interrupt Macro and the **%]** - Retrieve Entry - at the end of the Interrupt Macro. This would restore the data to the Entry Buffer. Any data previously saved with **%[** is lost.

If it is not desirable to return to the Interrupted Macro, then that Macro can be deleted from the stack by using the **n%B** command. A new variation of the Break Macro Command **- $%B** has been added. This command will delete all Macros from the current stack except the currently running Macro.

An Interrupt Macro must be invoked via Setpoint Control or Input Interpreter. Front Panel Keys or Serial Commands to invoke a Macro do not actually put a Macro on the Macro Stack and thus cannot cause an interrupt to occur.

MACRO DEBUG

The macro debug is a comprehensive diagnostic tool that allows you to track macro execution. The debug uses a portion of RAM permanently allocated to record each macro command executed. The debug buffer is a circular buffer. When the buffer becomes full, new macro commands will continue to be recorded while deleting the oldest records. The macro debug is always enabled to record macro execution. Thus, it is possible to review the most recent macro activity at any time. Printing the macro debug provides additional information such as relative execution times, macro stack activity, branching results and error conditions.

REVIEWING THE MACRO DEBUG TABLE

The macro debug table is located at P50001 of the setup mode. To access the debug parameter from the weigh mode, key in

50001 [SELECT] 23640 [ID] [ENTER]

P5003%	Mc 10 4%D%E

The last macro command executed (the last entry in the debug buffer) will be shown in the rightmost position of the display. From this point, the **[?]** and **[?]** keys can be used to scroll backwards and forwards through the debug buffer. Scrolling forward past the end of the buffer will select back to the beginning.

As you scroll through the debug buffer the display will indicate the macro number (top right), macro commands (bottom right) and location in the macro table (last four digits of the parameter number). Contiguously executed macro commands will appear consecutively in “blocks” of code. Whenever macro execution is no longer contiguous (i.e. upon completion of a macro, calling another macro, branching within a macro, etc.), a solid block character ‡ will appear indicating the separation between contiguous “blocks”.

P5002*	Mc 10 1%O‡L

Since the macro debug table can become quite large, it is often desirable to skip forward or backward more than one location at a time. You can skip past “blocks” of macro commands by first keying in the number of “blocks” to skip before pressing **[?]** or **[?]**. For example, keying in

20 [?]

will skip backwards 20 “blocks” from the currently selected point in the macro debug table. To advance directly to the beginning or end of the table from any location, press **[.]** before pressing **[?]** or **[?]**. This will take you to the ENDof TABLE prompt. Then press **[?]** to access the end of the table, or **[?]** to access the beginning of the table. You can also access the last record of the debug buffer by keying in **50001 [SELECT]** at any time.

PRINTING THE MACRO DEBUG TABLE

The most effective way to analyze the macro debug is to print it. Printing the macro debug table allows you to transmit the debug in full or partial detail out any of the four communication ports.

PRINTING “FULL” DETAIL

To print the macro debug table in full detail, select the desired location in the table, press **[PRINT]** and key in the desired comm port number as prompted. The debug buffer will print from the currently selected location to the end of the table. To print the entire debug table, advance to the last record in the table before pressing **[PRINT]**. The following example demonstrates how to print the entire debug table.

50001 [SELECT]

```
P5003% Mc  10
       4%D%E
```

[PRINT]

```
P5003% Enter
       Comm#
```

1

```
** Relative Trace Print Time:  636.160 s  **

    Macro 2   ; Start Time:   22.493 s; Stack:   0
11%O                      002:0004 if  setpoint on

    Macro 2   ; Start Time:   22.510 s; Stack:   0
L134%D                    002:0033 deactivate setpoint
%E                        002:0035 end if
```

The first line of the debug output shows the relative trace print time, the time elapsed since the last reset caused by a power up condition or exiting the setup mode. This is followed by the "blocks" of executed macro code. Each block begins with a header indicating the macro number, relative start time and the macro stack count at the time the block was executed.

PRINTING "BRIEF" DETAIL

The macro debug buffer holds a significant amount of data, resulting in a lengthy debug output. The debug transmission can be shortened by pressing **[UNITS]** instead of **[PRINT]** to print the table in "Brief" mode. This mode reports only the starting "block" information without the macro code as shown below.

50001 [SELECT]

```
P5003% Mc  10
       4%D%E
```

[UNITS]

```
P5003% Enter
       Comm#
```

1

```
** Relative Trace Print Time:   638.864 s  **

   Macro 2
  Range of Offsets: 10001 to 10004; Start Time:   22.493 s; Stack:   0
```

```
  Range of Offsets: 10028 to 10035; Start Time:   22.510 s; Stack:   0
```

CLEARING THE MACRO DEBUG

The macro debug buffer is only cleared automatically when the indicator is powered down. The debug buffer can be cleared while powered up by accessing the macro debug table at P50001 and pressing **[CLR]**.

50001 [SELECT]

P5003%	Mc 10 4%D%E

[CLR] [ENTER]

After exiting the setup mode, all but a small of the remaining RAM is allocated for the macro debug trace buffer. The amount of RAM allocated for the debug buffer is reported at P60005. The amount of remaining RAM is reported at P60004. To increase the amount of remaining RAM at P60004 in order to increasing the size of setup RAM at P60040 or pile RAM at P60050, clear the macro debug trace buffer at P50001.

USING MACRO DEBUG MACRO COMMANDS

There are several macro commands that can be used during macro execution to customize the operation of the macro debug. These commands provide the ability to clear the debug buffer, suspend and resume the recording of individual macros or groups of macros and reset the trace timer. Refer to the %B macro command for more details.

DOWNLOADING INDIVIDUAL MACROS

You can download or print individual macros to verify the contents of the macro table. This feature also serves as an editing tool. A macro can be saved to disk, edited with a text editor, then reloaded to replace the existing macro without having to upload the entire setup file.

To download an individual macro:

1. Access the setup mode using the full access code. (A macro accessed and protected by the limited access code cannot be downloaded.)
2. Select any parameter within the macro table (P10001 → P19999).
3. Press **[PRINT]**. The display will show `Enter Comm#`.
4. Press **[1]**, **[2]**, **[3]** or **[4]** to transmit the macro table out the desired comm port.

The following example shows how to download macro #10:

10001.10 [SELECT] 23640 [ID] [ENTER]

```
P1000! Enter
       Comm#
```

[1]

```
19999.10%s%c%e           P10001.  Macro #10
11%%O%e                    002:0004 if  setpoint on
        %%{%e                      002:0006 Start Group
        U134%%D%e                  002:0012 deactivate setpoint
        10%%O%e                    002:0016 if  setpoint on
        134%%A%e                   002:0021 activate setpoint
        %%E%e                      002:0023 end if
        %%}%e                      002:0025 End Group
%%N%e                      002:0027 if not
L134%%D%e                  002:0033 deactivate setpoint
%%E%e                      002:0035 end if
```

Note that the macro download contains an offset value indicating the location of each macro command within the macro table. For example, the offset **002:0033** in the line

`L134%%D%e` **`002:0033`** `deactivate setpoint`

indicates that this command ends at the 33rd location of macro #2. This can help you quickly locate a specific location to edit within a large macro table. To access the location in this example, key in

10033.2 [SELECT]

```
P1003# Mc    2
       134%D
```

Chapter 10 SETPOINTS

This chapter describes the various version of the Input/Output modules. Setpoint setup is also described and several common setpoint setup examples are provided.

Throughout this chapter, various references are made to the maximum allowable number of setpoints. These references reflect the maximum value allowed for the 60 Series instruments. The maximum values for other models may be limited as noted in the specifications section of Appendix A.

OVERVIEW

SETPOINTS

Each 660 Series controller maintains 256 setpoint values. The first 128 setpoints can control physical inputs or outputs. Setpoints 145 → 256 can be set up as inputs to invoke macros via the Modbus protocol operations (refer to *Chapter 8, Communications*). All 256 can be used as logical setpoints. The interface connections for these logic I/O are on J1 of the main board. The interface connections for setpoints 131 through 138 are on J11 of the main board.

The 560 Series indicator maintains 48 setpoint values. The first 32 setpoints can control physical inputs or outputs. Setpoints 33 → 48 can be set up as inputs to invoke macros via the Modbus protocol operations (refer to *Chapter 8, Communications*). All 48 can be used as logical setpoints. The interface connections for these logic I/O are on J1 of the main board.

Similarly, each 460 Series indicator maintains 16 setpoint values. The first 8 setpoints can control physical inputs or outputs. Setpoints 9 → 16 can be set up as inputs to invoke macros via the Modbus protocol operations (refer to *Chapter 8, Communications*). All 16 can be used as logical setpoints. The interface connections for these logic I/O are on J3 of the main board.

By using macros, all of the setpoints can be configured to trigger certain events, such as print, tare, accumulate, or zero operations or total system control and so on, when the setpoint activates, deactivates, or does both. In this way, many applications can make use of setpoints without actually having a relay or logic output connected to anything. Refer to *Chapter 9, Macros* for more details on Macro operations.

One selection includes a time minimum that simulates the operation of a timer relay.

Your selections specify the conditions that activate (energize) or deactivate (de-energize) the setpoints. For the simplest uses, the setpoint can be set up so the output will activate when the gross weight exceeds a certain value and deactivate when the weight falls below that same value. However, you can program the setpoint change of state to be based on any operating parameter as compared to any other.

SETPOINT SETUP PARAMETERS

To set up Setpoint #1, press 5099 [SELECT]. You will be asked to enter the program security code before changes can be made. Key in 23640 [ID] [ENTER].

The setpoint setup parameters listed in detail in *Chapter 3, Setup Parameters* beginning on page 3-44.

The setup for *all* the setpoints begins at parameter **P5099** and continues through **P5150**. To access the setup for any setpoint, you must enter the Setpoint Setup Mode number and press **[SELECT]**. You must also specify a setpoint *instance* (a setpoint number between 1 and 256) at parameter **P5099**. Once a setpoint instance is specified, the other parameters through **P5150** are associated with that setpoint. Each setpoint uses the same parameters for setup. The only difference is the *instance* of the setpoint selected at P5099.

SETPOINT OPERATION

When a setpoint is inactive, only the conditions relating to the activation of the setpoint are checked and reacted to accordingly. Similarly, when a setpoint is active, only the conditions relating to the de-activation of the setpoint are checked and acted upon. Each setpoint is checked and updated 60 times per second, with the following exceptions:

- Installing additional multi-scale modules may reduce the update.
- The execution of a command can delay the next setpoint update slightly.
- Continuous transmit transmissions can delay the setpoint update.
- Accessing the setup mode will deactivate all setpoints.
- Setpoints for the net or gross weight will be based upon the weight after it is rounded off to the selected display increment (set in P111).

SETPOINT STATUS MODE

Once the setpoints are set up for your application, you can check the status of a particular setpoint by accessing the *Setpoint Status Mode*. The setpoint continues to be updated in this mode. In addition, any or all setpoints can be set to an active or inactive state.

To reach the Setpoint Status Mode, key in **78 [SELECT]**. The numeric display will read SP- 1. The top line of the dot matrix display will indicate whether the setpoint is enabled or disabled. The bottom line will display Activ or DeAct, depending on the current state of the setpoint.

Once you are in the Setpoint Status Mode, you can view the status of a particular setpoint by keying in the setpoint number and pressing **[ENTER]**. For example, to view the status of Setpoint 13, key in **13 [ENTER]**. You can also view the status of the other setpoints by pressing **[ENTER]** once per setpoint, or by pressing the arrow keys, **[F1▲]** or **[F4▼]** on the 660 Series, **[F1]** or **[F2]** on the 560 Series and **[F1▲]** or **[TARGET]** on the Model 465.

You can change the state of any setpoint you are viewing in the Setpoint Status Mode. Press **[·] [1] [ENTER]** to activate <Setpoints;Activate>, or **[·] [0] [ENTER]** to de-activate <Setpoints;Deactivate> or, if you prefer, you can use the **[F5▶]** key on the 660 Series or **[TARE]** on the Model 465 and 560 Series to activate a setpoint and the **[F3◀]** key on the 660 Series or **[UNITS]** on the Model 465 and 560 Series to deactivate a setpoint.

To activate or deactivate a setpoint other than the one you are viewing, key in its number followed by the **[·] [1]** or **[·] [0]** command and press **[ENTER]**. You can select all setpoints by keying in **999**. For example:

- To select and activate Setpoint #18, key in **18 [·] [1] [ENTER]**.
- To change *all* the setpoints to the activated state, key in **999 [·] [1] [ENTER]**.
- To de-activate all setpoints, key in **999 [·] [0] [ENTER]**.

  - or -

Setpoints that are not enabled in the Setup Mode will be activated or deactivated by the commands above. They will not change state on their own except to be deactivated during the initial power-up and power-down of the controller.

Key in **78 [SELECT]** to access the setpoint test mode.

Pressing the right and left arrow keys turn the current relay off and on respectively.

The external setpoint option outputs will also change state as these commands are executed. However, a setpoint might not stay in the revised state if the conditions governing its setup cause it to change to the opposite state.

To exit the Setpoint Status Mode, press **[SELECT]**.

For more information on input/output parameters, see *Chapter 3, Setup Parameters*.

SETPOINT STATUS SERIAL TRANSMISSION

The *status* of the setpoints can be transmitted out a specified port in binary text form. Special format codes allow for this capability. Refer to page 8-46 for more information.

SETPOINT INPUTS

The multiple setpoint option cards enable as many as 128 input connections. Various devices — such as photo-detectors or proximity switches — can be connected to the controller, signaling a closed or open contact. Such devices can be used to sense the presence of a container before a filling process is begun, or the presence of a truck over an axle-weighing scale.

The first step in setting up a setpoint is to define it as an *input* or *output* mechanism. Selecting the setpoint as an *input* enables a macro to be invoked each time the input changes state. Separate macros can be invoked when the input becomes active and deactivated. Another macro, activated by some other condition, can branch around part of the commands of the original macro, depending on the state of one of the inputs.

This *conditional branching* depends on whether or not the input is active or deactivated.

An input setpoint can only be activated or deactivated by an external signal. The %A and %D macro commands will not change the state of an input.

For clarity, you specify which parameter number is being set before each [SELECT] command on each line. When entering the setup data, you can press the [SELECT] key to advance to the next parameter.

A. Enable Setpoint instance 1:
5099 [SELECT] [1] [ENTER]
P5099.1 Setpt instance 1 Enbld

B. Setpoint Type:
5100 [SELECT] [1] [ENTER]
P5100.1 SPt 1 Output

C. Choose proper selections for activation of Setpoint #1:

5110 [SELECT] [0] [ENTER]
P5110.0 Activ Above

5112 [SELECT] [1] [ENTER]
P5112.X Macro "1"

5113 [SELECT] [0] [ENTER]
P5113.0 Mot'n Ign'd

5114 [SELECT] 80.1 [ENTER]
P5114.X ALPar (VAR#1)

Enter a target value in VAR#1.

D. Choose proper selections for de-activation of Setpoint #1:

5130 [SELECT] [1] [ENTER]
P5130.1 Deact Below

5132 [SELECT] [0] [ENTER]
P5132.X Macro none

5133 [SELECT] [0] [ENTER]
P5133.0 DeMtn Mot'n Ign'd

5134 [SELECT] 80.2 [ENTER]
P5134.X DLPar (VAR#2)

Enter the value of 2 in VAR #2.

E. Select the parameter that Setpoint #1 is based upon:

5150 [SELECT] 30 [ENTER]
P5150.X Par 30 QTY

F. Setpoint Setup is now complete. Proceed to setup macro 1 which is set to invoke on activation of setpoint#1.

SETPOINT KEYPAD ENTRY

A good application example is to have a macro automatically invoked as a container of parts is placed on the scale platform. This macro could be programmed to do a number operations that would be cumbersome if they were done manually. A macro could be used to auto-accumulate parts, prompt as a check-weigher (good, bad or over), and so on.

The setpoint would be actuated when the *quantity* exceeded a threshold of three pieces. As motion settled, the macro would be invoked to perform a specific operation, such as an accumulation. As the quantity fell below two pieces, the setpoint would reset. It would now be ready for the next cycle.

SETPOINT OUTPUT RESPONSE TIME

The setpoint output response time is directly related to the filter response time of the *input* signal. Table 10-1 shows the filter selections for parameter P116 and the response times. Any additional time to update the setpoints from values shown in Table 10-1 due to propagation delays is negligible.

Table 10-1: Filter Response Times

Filter		Response (in ms)	
Number	**Name**	**90%**	**100%**
11	off	90	100
0	0.06	150	170
1	0.12	225	250
2	0.25	350	400
3	0.5	400	475
4	1	500	600
5	2	900	1200
6	4	1800	2300
7	8	3600	4600
8	2 sA	500 to 900	600 to 1200
9	4 sA	500 to 1800	600 to 2300
10	8 sA	500 to 3600	600 to 4600

SETPOINT SETUP EXAMPLES

The following setup examples are structured like an ASCII file. It is possible to enter these setups manually from the controller front panel keypad, using a slightly different approach. Using a remote keyboard or a terminal will make setup entry a little easier. If the file is in ASCII form, the setup procedure is the quickest of all approaches. GSE recommends that you back up all custom setups on a PC and diskette. This will make service calls and resale of the same program much easier.

Example #1: Over/Under Controller

```
100%s23640%i%e Access Setup Mode
```

SET LOWER ACTIVATION SETPOINT LEVEL (ACTIVE BELOW 15 LBS.)

```
5099%s1%e   P5099.1  Setpt 1
5100%s1%e   P5100.1  SPTyp Outpt
5110%s1%e   P5110.1  Activ Below
5111%s0%e   P5111.1  AcDly 0.00
5112%s0%e   P5112.X  AcMac none
5113%s0%e   P5113.0  AcMtn Ign'd
5114%s80.1%e   P5114.X  ALPar VAR#1
            (set VAR1 to 15)
5130%s0%e   P5130.0  Deact Above
5131%s0%e   P5131.1  DeDly 0.00
5132%s0%e   P5132.X  DeMac none
5133%s0%e   P5133.0  DeMtn Ign'd
5134%s80.1%e   P5134.1  DLPar VAR#1
5150%s0%e   P5150.0  CmPar Gross
```

SET UPPER ACTIVATION SETPOINT LEVEL (ACTIVE ABOVE 25 LBS.)

```
5099%s2%e   P5099.2  Setpt  2
5100%s1%e   P5100.1  SPTyp Outpt
5110%s0%e   P5110.0  AcAct Above
5111%s0%e   P5111.1  AcDly 0.00
5112%s16%e  P5112.X  AcMac none
5113%s0%e   P5113.0  AcMtn Ign'd
5114%s80.2%e   P5114.0  ALPar VAR#2
         (set VAR2 to 25)
5130%s1%e   P5130.1  DeAct Below
5131%s0%e   P5131.1  DeDly 0.00
5132%s0%e   P5132.X  DeMac none
5133%s0%e   P5133.0  DeMtn Ign'd
5134%s80.2%e   P5134.1  DLPar VAR#2
5150%s0%e   P5150.0  CmPar Gross

%z Exit Setup Mode
```

Example #2: Latching Relays

```
100%s23640%i%e Access Setup Mode

NAME VAR#1 FOR SETPOINT 1 TARGET VALUE
682%sSP-1 VALUE%e P682.--  Var#1 SP-1 VALUE
```

SET SP-1 ACTIVATION POINT TO 100% OF VAR#1 VALUE

```
5099%s1%e   P5099    Setpt 1
5100%s1%e   P5100.1  SPTyp Outpt
5110%s0%e   P5110.0  Activ Above
5111%s0%e   P5111.1  AcDly 0.00
5112%s1%e   P5112.X  AcMac 1*
5113%s0%e   P5113.0  AcMtn Ign'd
5114%s80.1%e   P5114.1  ALPar VAR#1
```

**Set macro 1 up to multiply the value of VAR1 by 95% and place the result in VAR2.*

SET SP-1 DE-ACTIVATION POINT TO 95% OF VAR#1 VALUE

```
5130%s1%e   P5130.1  DeAct Below
5131%s0%e   P5131.1  DeDly 0.00
5132%s0%e   P5132.X  DeMac none
5133%s0%e   P5133.0  DeMtn Ign'd
5134%s80.2%e   P5134.X  DLPar VAR#2
5150%s0%e   P5150.0  CmPar Gross

%z%c%e%e    Exit Setup Mode
```

Example #3: Activate on [TARE]

```
100%s23640%i%e Access Setup Modes
```

NAME VAR#1 FOR SETPOINT 1 TARGET VALUE

```
682%sSP1  TARGET%e   P682.--  Var#1 SP1 TARGET
```

SET TARE KEY TO ACTIVATE MACRO #1

```
802%s1%e set to invoke macro 1
```

SETUP SETPOINT TO DEACTIVATE AT TARGET (100% OF VAR#1 VALUE)

```
5099%s1%e   P5099.1  Setpt  1
5100%s1%e   P5100.1  SPTyp Outpt
5110%s6%e   P5110.4  Activ Never

5130%s0%e   P5130.0  Deact Above
5131%s0%e   P5131.1  DeDly 0.00
5132%s16%e  P5132.X  DeMac none
5133%s0%e   P5133.0  DeMtn Ign'd
5134%s80.1%e   P5134.1  DLPar VAR#1
5150%s1%e   P5150.1  CmPar Net

9990%s1%e   macro 1.
10000%s%c%e
%%t%e    tare.
1%%A%e      activate setpoint 1.

%z%c%e%e    Exit Setup Mode
```

Example #4: Grading (Up to 32 Ranges)

```
100%s23640%i%e Access Setup Modes
```

SET SMALLEST GRADE RANGE

```
5099%s1%e   P5099.1  Setpt 1
```

```
5100%s1%e   P5100.1  SPTyp Outpt
5110%s2%e   P5110.2  Activ Betwn
5112%s0%e   P5112.X  AcMac none
5113%s1%e   P5113.1  AcMtn Inhib
5114%s80.1%e   P5114.   ALPar VAR#1*
5115%s80.2%e   P5115.   AUPar VAR#2*
5130%s3%e   P5130.3  DeAct Outsd
5132%s0%e   P5132.X  DeMac none
5133%s1%e   P5133.1  DeMtn Inhib
5134%s80.1%e   P5134.   DLPar VAR#1*
5135%s80.2%e   P5135.   DUPar VAR#2*
5150%s0%e   P5150.0  CmPar Gross
*(set VAR#1 to 0.5 and VAR#2 to 10)

SET 2ND SMALLEST GRADE RANGE
5099%s2%e   P5099.2  Setpt 2
5100%s1%e   P5100.1  SPTyp Outpt
5110%s2%e   P5110.2  Activ Betwn
5112%s0%e   P5112.X  AcMac none
5113%s1%e   P5113.1  AcMtn Inhib
5114%s80.3%e   P5114.   ALPar VAR#3*
5115%s80.4%e   P5115.   AUPar VAR#4*
5130%s3%e   P5130.3  DeAct Outsd
5132%s0%e   P5132.X  DeMac none
5133%s1%e   P5133.1  DeMtn Inhib
5134%s80.3%e   P5134.   DLPar VAR#3*
5135%s80.4%e   P5135.   DUPar VAR#4*
5150%s0%e   P5150.0  CmPar Gross
*(set VAR#3 to 10 and VAR#4 to 20)

SET THIRD SMALLEST GRADE RANGE
5099%s3%e   P5099.3  Setpt 3
5100%s1%e   P5100.1  SPTyp Outpt
5110%s2%e   P5110.2  Activ Betwn
5112%s0%e   P5112.X  AcMac none
5113%s1%e   P5113.1  AcMtn Inhib
5114%s80.5%e   P5114.   ALPar VAR#5*
5115%s80.6%e   P5115.   AUPar VAR#6*
5130%s3%e   P5130.3  DeAct Outsd
5132%s0%e   P5132.X  DeMac none
5133%s1%e   P5133.1  DeMtn Inhib
5134%s80.5%e   P5134.   DLPar VAR#5*
5135%s80.6%e   P5135.   DUPar VAR#6*
5150%s0%e   P5150.0  CmPar Gross
*(set VAR#5 to 20 and VAR#6 to 30)
%z%c%e%e    Exit Setup Mode
```

Example #5: Power Up Setpoint

```
100%s23640%i%e Access Setup Modes
5099%s250%e    SPT #250
5100%s1%e      SPTyp Outpt
5101%s%c       SPNam POWER-UP
5110%s4%e      Activ Alwys
5111%s0.00%e      AcDly 0.00
```

```
5112%s250%e    AcMac 250
5113%s0%e      AcMtn Ign'd
5130%s5%e      Deact Never
5131%s0.00%e      DeDly 0.00
5132%s0%e      DeMac None!

9990%s250%e%e        MACRO #250
9991%s%c       POWER-UP
9994%s1%e            limited access enabled

19999.250%s%c%e

@POWER-UP%%T%e       tag POWER-UP
   1%%Q%e            send DISPLAY TEST
   %%P%%P%e          pause
   CUSTOMER NAME INC.P1,2a%%C%e  display 4X20 text
   Packaging SystemP3,3%%C%e  display 4X20 text
   BHT-23 v1.00P4,5%%C%e      display 4X20 text
   %%P%%P%%P%e          pause (3 seconds)

%z%c%e%e       Exit Setup Mode
```

In this example a POWER-UP setpoint is used to invoke Macro 250 upon power up/power reset and display the version of program loaded in a Model 661. With the setpoint always active, it will not deactivate until power is lost. The setpoint will again be active when power is restored. This also applies to entering and exiting the setup mode.

Example #6: Setpoint Timer

```
100%s23640%i%e Access Setup Modes

5099%s2%e      SPT #2
5100%s1%e      SPTyp Outpt
5101%s%c       SPNam TIMER
5110%s5%e      Activ Never
5111%s0.00%e      AcDly 0.00
5112%s0%e      AcMac None!
5113%s0%e      AcMtn Ign'd
5130%s4%e      Deact Alwys
5131%s15.00%e     DeDly 15.00
5132%s15%e     DeMac 15


9990%s2%e%e       MACRO #2
9991%s%c       ACTIVATE TIMER
19999.2%s%c%e
4%%A%e         activate timer


9990%s15%e%e         MACRO #15
9991%s%c       TIMER ACTIVATED MACRO
19999.1%s%c%e
MIX TIMER EXPIREDP5,4,f2%%C%e  display text
%%P%%P%%P%e            3 second pause
Pa%%C%e                clear display
```

This example demonstrates using a setpoint as a timer. Pressing **[F2]** or **[TARGET]** will start the timer. Macro 15 will be invoked after the 15 second timer expires (setpoint 2 deactivates). The time delay may also be set through a macro. Refer to Chapter 9 for more details.

SETPOINT OPTIONS

The options boards mount in a variety of locations on the different models. Refer to the Options Installation Chapter 13 for step by step installation instructions.

Chapter 11 DATABASE

This chapter describes the use of the database option.

OVERVIEW

INSTALLATION

This chapter describes some applications that can be accomplished by using a Database and operating instructions. The 460 Series indicators have a database resident on board which has a size of 4 KB. The 660 Series controller will accept one of the battery backed RAM Database Option boards specified below:

Database	**Part Number**
Database Option Board, 256 KB	24660B-256K0
Database Option Board, 1 MB	24660B-001M0
Database Option Board, 2 MB	24660B-002M0

The 560 Series controllers can also use any of the database options listed above, however it can only address a maximum of 256K. Thus using the 1M or 2M database options will provide no added benefit to using a 256K.

WHAT IS A DATABASE?

A database is a collection of useful information that is organized in a specific manner for easy access. For instance, a telephone directory can be thought of as a database. See Figure 11-1.

	Column ↓	**Column** ↓	**Column** ↓
Column Name or “Field” →	**Name**	**Telephone Number**	**Address**
Row→	Name 1	Phone No. 1	Address 1
Row→	Name 2	Phone No. 2	Address 2
Row→	Name 3	Phone No. 3	Address 3
Row→	Name 4	Phone No. 4	Address 4
Row→	Name 5	Phone No. 5	Address 5

Figure 11-1: Telephone Directory

Components of a database include *fields* and *records*. Database terms are defined in the table below.

Term	**Definition**
Field	A *parameter type* or *column name*. For instance, “Name 1” and “Name 2” are fields that refer to data stored in the column defined as **Name** in the preceding telephone directory example.
Record	One *row* in a database, or a collection of fields in a *column* of a database. In the preceding telephone directory example, one of the four records shown includes Name 2, Telephone Number 2 and Address 2 fields.

Database	A collection of records. For example, the entire telephone directory is a database.

A database is defined by parameters. These parameters and the data in them are stored in the database. Each stored parameter becomes a *column* in the database.

Rows of data are created using the “make row” command. Copies of the current data in the row parameters are stored together with the column parameters in the database.

Once a row exists, that entire row of data can be *recalled*. To recall data, you must first specify which column of data is to be searched. If no column is specified, the first column becomes the default search column.

After you specify a column, you must specify the exact data that is to be located. To do this:

- You make an entry at the “Recall Row” command

or

- You store the search value into the search parameter and press **[ENTER]** at the “Recall Row” command without making an entry

If a row is found whose data in the search column matches the specified data exactly, then all of the data within that row is copied into the parameters associated with that column.

DATABASE APPLICATIONS

Databases can be used for different purposes. Some of the most popular weighing applications use a database as a transaction recorder, or for looking up part numbers, the quantity of parts on hand, and batch formulas.

There are many other possible uses for a Database, as well. Up to 250 databases can be accessed on the 660 Series controllers, up to 100 database can be accessed on the 560 Series controllers and 15 databases may be used on the 460 Series indicators. Several different applications can be combined in one installation.

TRANSACTION RECORDER DATABASE

Using a Transaction Recorder database, every weighing operation that occurs on the scale can be stored to memory with other pertinent data. This data can include time/date, employee ID, job number, part number, number of boxes, customer number, and so on.

Normally this type of database only *makes rows*. The data is never recalled. A report can be sent to a computer or printer at the end of the shift, day, or week, and then the data is deleted.

PART NUMBER LOOK-UP DATABASE

When a part is weighed, the part number is keyed in. Information about the part — for example, description, bin location, etc. — can be recalled from a Part Number Look-Up database. The information can be used to print a label, which is placed on the part. This same concept can be used to recall a name and address based on customer ID.

QUANTITY ON HAND DATABASE

A Quantity on Hand database can be used to track the inventory levels of various products based on their collective weight (or quantity by using the counting capabilities.

When an item number is keyed in, the current amount on hand (in weight or quantity) is recalled into the Gross or Net total register. Then, depending on whether the stock is being increased or reduced, an add or subtract accumulation operation is performed. The database row is then updated to show the revised amount on hand.

Using a Variable (INT) that defines a *safety stock level*, a buy report can be issued when the amount of stock on hand nears or drops below the defined safety stock amount. Another database can also be used to track the stock amounts on order.

BATCH FORMULA LOOKUP DATABASE

Many applications require that specific proportions of various ingredients be mixed for a batch or formula. The Batch Formula Lookup database can store the amounts of each ingredient either as a fixed amount or as a percentage of the total batch size. One column of the database can contain the number of the setpoint that must be activated to supply the required ingredient.

DATABASE SETUP

The setup mode for the database can be accessed only if the Database Option is installed (560/660 Series). This is necessary because the setup information for the database is stored within the Database Option. This makes servicing easy because a database board can be moved from one controller to another if, for any reason, the controller must be replaced (560/660 Series).

Each database has its own setup mode. Table 11-1 lists the database setup parameters.

Table 11-1: Database Setup Parameters

PARAMETER	DESCRIPTION
P699: DB #:	Specifies the database *number* to be setup. Selections are 1 - 250.
P700: DBNam	Allows for specifying the database name (customize to usage). Use arrow keys to key in alpha characters.
P701: Col01	Specifies the parameter to be stored in column *one* of the database. Instances must be chosen for most parameters.
P702: Col02	Specifies the parameter to be stored in column *two* of the database. Instances must be chosen for most parameters.
P703: Col03	Specifies the parameter to be stored in column *three* of the database. Instances must be chosen for most parameters.

All subsequent database parameters through P798 follow the pattern of those listed in the Database Setup Parameters table. As many as 98 columns are allowed in each database.

The database setup parameter number is P699. To access the database setup mode:

1. Key in **699**.
2. Press **[SELECT]**.

Now, you can specify the database number you want to access by keying in that number and then press **[ENTER]**. For the Model 460, use the **[PRINT]** and **[UNITS]** keys to enter the database number desired. Press **[SELECT]** to move to parameter 700 and name the database. Press **[SELECT]** to move to parameter 701 and specify the parameter to be stored in column 1. Press **[SELECT]** again to move to parameter 702 and specify the parameter to be stored in column 2, and so on (through parameter 798 for 98 columns, if you like).

Most parameters (variables, etc.) entered as the column definition will require that you also specify an *instance* (for example, 80.**3**).

To assign a parameter to a column, key in the parameter ID along with the instance (80.4, for example), and press **[ENTER]**. If you are unsure which parameter you require, use the **[F1▲]** and **[F4▼]** on the 660 series, **[F1]** and **[F2]** on the 560 and **[F1▲]** and **[TARGET▼]** keys on the Model 465 and the **[PRINT▲]** key on the Model 460 to scroll through the list of parameters. When the appropriate parameter number is displayed, press **[ENTER]** to save it. The indicator will prompt you to enter an instance. Repeat this procedure for all the columns (P701 through P798) your application will require.

Only the first five characters of a parameter name are shown on the lower line of the dot matrix display initially. You can view the rest of the name by pressing either the left or right arrow keys, respectively. If no name has been assigned to the parameter, “None” is displayed.

To delete a column definition, press the **[CLR]** key when that column is shown. To delete *all* the current selections for the columns of a database:

1. Specify the database instance at parameter P699 (for example, **1**), and press **[ENTER]** or **[ZERO] + [TARE]** keys on the Model 460.
2. Press the **[CLR]** key or **[TARE]** on the Model 460.

If rows have already been created in a database and you attempt to redefine the columns, the prompt “CLEAR DBASE” displays briefly followed by “ENTER toCLR.” Press **[ENTER]** to delete the existing rows in the database, or any other key to abort the operation.

DATABASE MENU

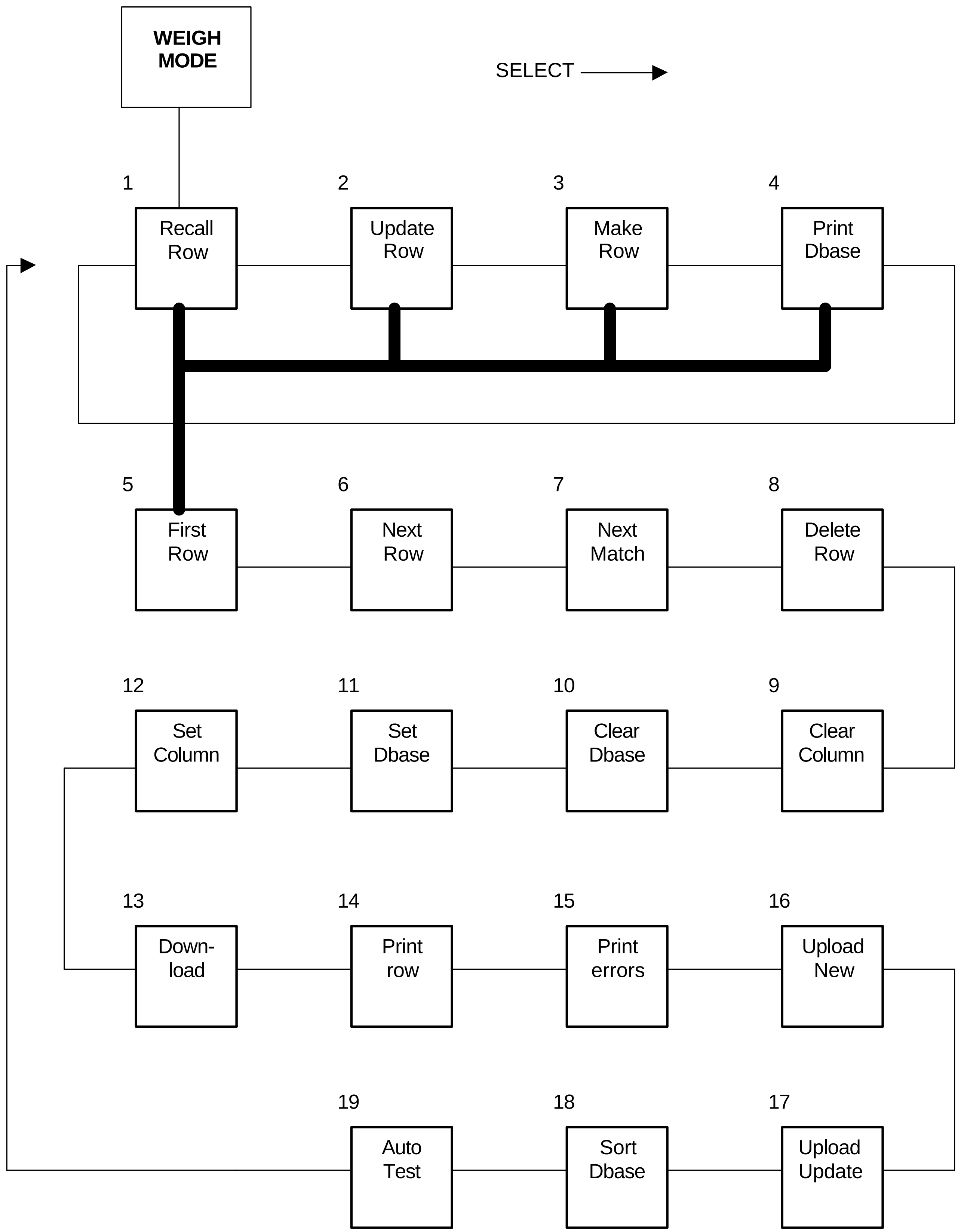
When parameter **P806** is set for “dbase,” the basic Database Menu can be accessed. To access this menu, you press the **[ID]** key while the controller is in one of the weigh modes (any mode below mode 90).

The basic Database Menu provides just *four* selections of operations. To view these selections you press the **[SELECT]** key. To invoke the selected operation, you can:

- Simply press the **[ENTER]** key alone, or
- Key in an (alpha)numeric entry, and then press the **[ENTER]** key

The operations listed on the basic Database Menu are considered simple operations.

More complex operations can be performed, but only by keying in the proper numeric operation code and pressing the **[SELECT]** key (see the chart on the following page).


WEIGH MODE
SELECT
1
Recall Row
2
Update Row
3
Make Row
4
Print Dbase
5
First Row
6
Next Row
7
Next Match
8
Delete Row
12
Set Column
11
Set Dbase
10
Clear Dbase
9
Clear Column
13
Down-load
14
Print row
15
Print errors
16
Upload New
19
Auto Test
18
Sort Dbase
17
Upload Update

RECALL ROW (FUNCTION 1)

A. **[ENTER]** key alone:

For the “Recll row” command, the first row matching the current data in the parameter (that corresponds to the currently selected column in the currently selected database) is located. All other information in the row is copied into the corresponding parameters.

Unless otherwise specified, the selected column defaults to being the first column. Thus, if the first column is set up as VAR#1 (VAR 80.1), and the data stored in VAR#1 is “ABC,” the current database is searched from the beginning for a row in which “ABC” appears in the first column.

Once the row is found, it is recalled. If a matching row is not found, an error message is displayed briefly. In addition, the error code is saved so a macro can test for the occurrence of that error.

The recall operation is used when information — such as the description, target weight, and/or quantity on hand — is needed about an item. This particular operation is used when the look-up data has already been entered into the corresponding controller parameter.

B. An entry followed by the **[ENTER]** key:

This is the same as method A, except the row that is being searched for is the one whose data in the selected column matches the entry that was made. This selection is used when the information to be searched has not already been entered into the controller.

UPDATE ROW (FUNCTION 2)

A. **[ENTER]** key alone:

For the “Updte row” command, the most recently accessed row in the current database is updated with the corresponding data from the parameters defined for that database. If the last attempt to locate something in the current database was not successful, a new row is created.

This selection is most often used after a row of data has been recalled. After one or more of the parameters defined as a column are changed (such as total amount on hand), the row in memory is updated with the revised information.

B. An entry followed by the **[ENTER]** key:

The first row in the currently selected database whose selected column matches the entered data is located. This row is updated with the information of the other parameters defined in that database.

Unless otherwise specified, the selected column defaults to being the first column. Thus, if the first column is set up as VAR#1 (VAR# 80.1), and the entry made was “ABC” **[ENTER]**, the current database is searched from the beginning, for a row in which “ABC” is the data in the first column.

The located row is changed so all other columns in it match the current values of their corresponding parameter. If a matching row is not found, a new row is created using the values from the corresponding parameters.

MAKE ROW (FUNCTION 3)

A. **[ENTER]** key alone:

For the “Make row” command, a new row is created using the corresponding data from the parameters defined for the currently selected database. Data in any or all columns can be a duplicate of other existing rows.

This selection is used most often when storing transaction data to memory when all data to be stored has already been entered into the controller.

B. An entry followed by the **[ENTER]** key:

A new row is created in the currently selected database. The entered data is stored in the selected column and the other columns are copies of the data from their corresponding parameters. Data in any or all columns can be a duplicate of other existing rows.

This selection is used most often when storing transaction data to memory when the key field (such as part number) has not already been entered in to the controller.

PRINT DATABASE (FUNCTION 4)

A. **[ENTER]** key alone:

The “Print dbase” command causes the stored data to be transmitted in a report format. “Enter Comm#” is displayed, prompting you to specify which port should be used, for example, COMM 1, 2, 3 or 4. Key in **[1]** for COM1, **[2]** for COM2, **[3]** for COM3 or **[4]** for COM4, and press **[ENTER]**. Refer to the printer documentation for details about the format for this transmission.

This selection is normally used to generate hard copy reports of the information stored within the controller. It can also be useful in generating an end of day report .

B. An entry followed by the **[ENTER]** key:

CAUTION! This method **MUST** be used when a macro is programmed to print the database. See Chapter 9 – Macros for details.

The only information that can be entered here is the port number. Key in **[1]** for COM1, **[2]** for COM2, **[3]** for COM3 or **[4]** for COM4, and press **[ENTER]**. There is no port selection prompt, but this selection operates the same as method **A**. If you do not need the port number prompt, this is a more straightforward method of initiating a printout.

ADVANCED DATABASE MENUS

Complementing the Basic Database Menu selections are complex commands that allow almost any operation you can imagine to be performed. Since a series of commands is generally required to accomplish a particular task, each of these complex commands normally is used only with a macro. To access the advanced commands:

1. Press **[ID]** to access the basic Database Menu.
2. Key in the command number.

3. Press **[SELECT]**.The name of the command will appear in the dot matrix display
4. Press **[ENTER]** to invoke the command.

Most advanced commands require an entry before you press **[ENTER]**. Once the command is executed, most commands return the controller to the weigh mode that was in effect before you pressed **[ID]**.

Once an advanced command is selected (but before you invoke it by pressing **[ENTER]**), you can view other advanced commands by pressing **[SELECT]**. These are shown on the display.

To invoke a displayed command, press **[ENTER]**. To return to the weigh mode that was in effect when you first accessed the Database Menu, press **[ID]**.

You use the command **[5] [SELECT]** to access to the advanced database commands from the basic Database Menu. Actually, you can enter any command number between 5 and 19 and the corresponding advanced command will appear. These numeric command selections can be issued from any of the database commands.

Once a numeric command is issued, you can scroll through all of the database commands by pressing the **[SELECT]** key repeatedly, until the database command menu terminates. Termination occurs when one of the database commands is executed or when you press the **[CLR]** key to exit the menu.

Anytime the database menu is accessed, only the four basic commands are available initially.

Each of the advanced database commands is detailed in this section. Note that you can use the arrow keys, **[F1▲]**, **[F3◄]**, **[F4▼]**, and **[F5►]** to key in alphanumeric data from the standard keypad.

FIRST ROW (FUNCTION 5)

A. **[ENTER]** key alone:

The “First row” command causes the very first row of the currently selected database to be recalled. When used with the “Next row” command, this operation is useful if every stored row is to be sequentially recalled and processed in some manner. For example, using the Custom Transmit setups, a report can be printed in a format that is not possible for the standard printout.

B. An entry followed by the **[ENTER]** key:

No entry is defined to precede the “First row” command, and any entry you make will result in an “ENTRY ERROR” message.

NEXT ROW (FUNCTION 6)

A. **[ENTER]** key alone:

The “Next row’ command causes the very next row of the currently selected database to be recalled. This can be used after the “First row” command to cause every stored row to be sequentially recalled and processed in some manner.

B. An entry followed by the **[ENTER]** key:

No entry is defined to precede the "Next row" command, and any entry you make will result in an "ENTRY ERROR" message.

NEXT MATCH (FUNCTION 7)

A. **[ENTER]** key alone:

The "Next Match" command operates much like the basic "Recll row" command, except that the search for the matching record begins with the record that follows the last accessed record. Specifically, the next row that matches the current data in the parameter — that corresponds to the currently selected column in the currently selected database — is located. Then all other information in that row is copied into the corresponding parameters.

This command can be used successively after a "Recll row" command. It enables multiple occurrences of the same data to be located and processed. For example, every occurrence of a transaction involving a specific part number can be recalled and printed.

B. An entry followed by the **[ENTER]** key:

This form of the "Next Match" command operates much like method **B** of the basic "Recll row" command, except that the search for the matching record begins with the record that follows the last accessed record. In effect, this is the same as method A of the "Next Match" command, except the data being searched for is the entered data.

CLEAR ROW (FUNCTION 8)

A. **[ENTER]** key alone:

The "Clear row" command is used to delete a stored row from memory. When invoked without an entry, this command deletes the last accessed row from the currently selected database.

Before you can delete a row from memory, the warning message "SURE? ???" is displayed. Press **[ENTER]** to delete the row from memory, or any other key to abort the deletion.

This command is normally used where information is stored in the database temporarily, such as tracking all the item numbers that are currently processed a certain way. After that process is completed, the rows can be deleted.

B. An entry followed by the **[ENTER]** key:

This form of the "Clear row" command enables you to specify which row is to be deleted at the time of the deletion. The data you enter before you press **[ENTER]** is the value used to locate the targeted row.

Before you can delete a row from memory, the warning message "SURE? ???" is displayed. Press **[ENTER]** to delete the row from memory, or any other key to abort the deletion.

CLEAR COLUMN (FUNCTION 9)

A. **[ENTER]** key alone:

The “Clear Colmn” command cannot be invoked without a numeric entry. If it is invoked without an entry, the “ENTRY ERROR” message is displayed.

B. An entry followed by the **[ENTER]** key:

This version of the “Clear Colmn” command allows you to clear a particular column of a database. If the column is numeric, the numeric values for that column in each row are set to zero. For alphanumeric columns, the data is set to be blank.

The number you enter is the parameter ID of the column to be cleared. As a safeguard, the warning message “SURE? ???” is displayed. Press **[ENTER]** to clear the column, or any other key to abort the command.

This command is useful in applications where the accumulated weight must be deleted periodically. This allows for the total amount of a product made during a shift to be totaled and printed, and then cleared for the next shift.

CLEAR DATABASE (FUNCTION 10)

A. **[ENTER]** key alone:

The “Clear Dbase” command is used to delete the entire stored database. As a safeguard, the warning message “SURE? ???” is displayed. If you press **[ENTER]**, every row in the currently selected database will be deleted. Press any other key to abort the command.

B. An entry followed by the **[ENTER]** key:

This form of the “Clear Dbase” command allows you to delete a non-selected database. As a safeguard, the warning message “SURE? ???” is displayed. Press **[ENTER]** to delete all the rows in the database, or any other key to abort the deletion.

This command is used most often in transaction applications where information about every transaction is stored. Then, at desired intervals (such as the end of the day, week, or shift), all of the stored data can be printed or downloaded to a computer. Afterward, the entire database generally is deleted.

SET DATABASE (FUNCTION 11)

A. **[ENTER]** key alone:

The “Set Dbase” command resets the current database so it becomes the first defined database. All database operations are performed on the currently selected database. Therefore, you must first select the required database before a series of operations can be performed on a specific database. If only one database is used, you never need to use the “Set Dbase” command.

B. An entry followed by the **[ENTER]** key:

To use this method for invoking the “Set Dbase” command, key in the number of the database (for example, 1) , and press **[ENTER]**. This sets the current database so all subsequent database commands will be performed on it until another “Set Dbase” command is issued.

Multiple databases can be used in many applications. For example, one database might include the part number, description, net weight on hand, amount on order, and so on while a second database might consist of transaction data for various part numbers, and a third might contain purchase orders for various part numbers, as well as quantities ordered and their due dates. If you try to select a database that is not currently set up, the error message “NOT SETUP” is displayed briefly.

SET COLUMN (FUNCTION 12)

A. **[ENTER]** key alone:

The “Set Colmn” command is used to specify a *key* column for the next database command. If you do not key in a column number before you invoke this command, the default (for example, look-up) column will be reset to the first column of the selected database. This occurs automatically every time the controller is powered up or is re-initialized after you exit the setup mode and save changes.

B. An entry followed by the **[ENTER]** key:

For this form of the “Set Colmn” command, you *must enter the parameter ID of one of the columns* for the selected database (for example, 80.9 for variable instance number 9) before you press **[ENTER]**. The specified column has no immediate effect but it will affect the next database command if it is dependent on the selected column.

Normally, the selected column defaults back to the first column after the first database command is issued. However, if your entry starts with a decimal point, the newly selected key column will remain in effect until the “Set Colmn”

command is issued again. If another database is selected and it does not include the parameter ID for the currently selected column, the key column will be reset to the first column.

Also, if the controller is re-initialized, the selected column will be reset to the first column of the first database that has been set up. For example, keying in **[·] 80.9 [ENTER]** will semi-permanently set the key field for the selected database to be the column whose parameter ID is variable 80 for instance number 9.

The “Set Colmn” command can be used to recall a row from memory based on information stored in any column other than the first column. Generally databases should be set up so the first column is searched. However, some applications require searches of other columns during certain operations, for example, attempting to determine a part number when only the description is known or searching through a transaction database (whose first column is part number) for a specific account number.

DOWNLOAD DATABASE (FUNCTION 13)

A. An entry followed by the **[ENTER]** key:

The “Down-load” command enables transmission of the currently selected database in a format that is easily accepted by a computer. The format enables the database to be imported into most spreadsheet programs, such as Excel©, Lotus 1-2-3©, Quattro©, Microsoft Access© and other database programs.

Key in **[1]** for COM1, **[2]** for COM2, **[3]** for COM3 or **[4]** for COM4, and press **[ENTER]**.

If you only press **[ENTER]**, **Enter Comm#** is displayed, prompting you to specify a port. Press any key other than **[ENTER]** to abort the transmission.

The format for this transmission is compatible with the “Upld new” and “Upld Updat” database commands of the controller. Backed-up databases can be easily reloaded into the controller.

The database transmitting commands enable you to specify any of the four COMM ports. The download command allows for several variations of download file formatting to be specified as part of the COMM port number.

- To specify which COMM port to use, enter 1, 2, 3, or 4.
- To add the upload characters to the download output, add a “**.1**” to the entry.
- To specify that the format for time/date parameters match their set up selection (for example, Time, Date, Time/Date, and Number) append another “**.1**” to the entry.

Then press the **[ENTER]** key.

For example, at Dbase menu #13, press:

3 [ENTER] to send the database to COM3 without the upload characters but with numeric values for time/date parameters.

2.1 [ENTER] to send the database to COM2 with the upload characters and numeric values for time/date parameters.

1.1.1 [ENTER] to send the database to COM1 with the upload characters and the time/date parameters formatted to match their setup parameter (P688). Variable type is a U-INT set at P686.

4.0.1 [ENTER] to send the database to COM4 without the upload characters but with the time/date parameters formatted to match their setup parameter (P688). Variable type is a U-INT set at P686.

If you are downloading the database and intend to reload it back into a controller in the future, add “**.1**” to the COMM number selection (for example, **2.1**), press **[ENTER]**, select COM2, and allow upload characters to be added to the beginning of the file. This will cause the database to be preceded by the necessary commands so the database upload command can be accessed when the file is sent back to the controller later on. The pre-pended information is:

16,d;p%y

where the "**d**" will be replaced by the **d**atabase number being downloaded, for example, 1 through 250. The "**p**" is replaced with the COMM **p**ort number 1 - 4.

The **".1**" entry also causes the last row of data to be followed by a line with "**ENDofDB**". When the file is reloaded, this causes the database upload mode to be exited.

The "Down-load" command is useful for *backing up* the database (saving a copy of the database somewhere else). For applications where scale transaction information is collected throughout the day, it can be used to transfer the collected data to a computer for permanent storage, integrating it into other databases, or for further analysis.

GSE recommends that you periodically back up onto a computer any controller databases you consider valuable to safeguard the data. While the controller provides reasonably secure storage, all computer applications recommend that data be backed up to prevent loss.

PRINT ROW (FUNCTION 14)

A. An entry followed by the **[ENTER]** key:

The "Print row" command causes the current row within the currently selected database to be transmitted in a format that will line up well on a printout when this command is issued again later. This command can be used for selectively building a report.

Key in **[1]** for COM1, **[2]** for COM2, **[3]** for COM3, or **[4]** for COM4, and press **[ENTER]**. If you press only the **[ENTER]** key, "**Enter Comm#**" is displayed, prompting you to specify which port to use. Press any key other than **[ENTER]** to abort the transmission.

The database transmit row commands enable you to specify any of the four COMM ports. The print row command allows you to specify several variations of row formatting as part of the COMM port number.

- To specify which COMM port to use, enter 1, 2, 3, or 4.
- To specify printing of row information only, key in the port selection, and press **[ENTER]**.
- To specify transmitting the header line of the database (which consists of the column names of the selected database only), key in the COMM port number followed by a decimal point and a non-zero value

For example, if you key in **2.1 [ENTER]**, COM2 will be selected and only the header line of the database will be printed.

For example, at Dbase menu #14, press:

3 [ENTER] to send the row to COM3 without the any header information.

2.0 [ENTER] to send the row to COM2 without the any header information.

2.1 [ENTER] to send only the database header to COM2.

If you press any key other than **[1]** through **[4]** to select a port, the transmission will be canceled.

The “Print row” command can be used to generate a report that contains only certain stored rows. This might be implemented with a series of “Recll Next” commands. This custom printing can also be implemented by appropriately programming a Custom Transmit.

PRINT ERRORS (FUNCTION 15)

A. **[ENTER]** key alone:

The “Print err’s” command is much like the “Print dbase” command, except that only the rows whose data is suspect are printed. A report format that causes multiple rows to line up well on a printout is used.

If you press only the **[ENTER]** key, “**Enter Comm#**” is displayed, prompting you to specify which port to use. Key in **[1]** for COM1, **[2]** for COM2, **[3]** for COM3, or **[4]** for COM4, and press **[ENTER]**. Press any key other than **[ENTER]** to abort the transmission.

The “Print err’s” command should be used only when problems are suspected in the stored data. It can also be used periodically to check for problems in the stored data.

The resulting transmission will begin with the header line (column names), followed by any corrupt rows, followed by a summary of the number of rows found to be corrupt compared to the total number of rows found.

Also if a bad link is detected in the list of rows, then that error will also be reported. A form feed character (^L) is sent at the end of the transmission.

Each row that is stored away is accompanied by a checksum that verifies the integrity of the stored data. Every time a row is accessed, integrity is checked by re-calculating the checksum. If the checksum is found to be incorrect, an error message warns the operator of this condition. Once a row is updated, the checksum is recalculated.

B. An entry followed by the **[ENTER]** key:

Key in **[1]** for COM1, **[2]** for COM2, **[3]** for COM3, or **[4]** for COM4, and press **[ENTER]**. Press any key other than **[ENTER]** to abort the transmission.

UPLOAD NEW (FUNCTION 16)

A. **[ENTER]** key alone:

The “Upld. new” command enables uploading of new rows of data into the database through one of the bi-directional RS-232 communication ports.

After you press the **[ENTER]** key for this selection, the controller begins reading in rows of data from the COMM port. During the upload, a counter displays the number of rows created. If the count exceeds 99999 then the displayed count becomes ####x where #### are the four most significant digits and ‘x’ represents a place holder for the least significant digits.

You can press **[CLR]** to abort the process. Normally, you should not do this until the transmission has first been halted at the source. In the absence of a **[CLR]** command, the controller will continue processing data until it receives a “^Z” (DOS end- of-file character).

The format required for the data is a comma delimited ASCII format, with a carriage return and optional line feed after each row. This format is compatible with the “Down-load” database command. Backed-up databases can be easily reloaded into the controller using this selection

The “Upld. new” command can be useful for restoring *backed up* databases. Also, for applications where the database that must be stored in the controller is available on another computer, this command allows for easy loading of that information.

You must have a computer communication program, such as Procomm or GSE 550 Simulation Software Terminal Window on your IBM-compatible PC to send files to the controller and receive files from the controller.

B. An entry followed by the **[ENTER]** key:

You do not have to enter any data before you press the **[ENTER]** key for this selection. If you do, the “ENTRY ERROR” message is displayed briefly.

UPLOAD UPDATE (FUNCTION 17)

A. **[ENTER]** key alone:

The “Upld. Updat” command enables the uploading of new rows or the updating of existing row of data to the controller database through one of the bi-directional RS-232 communication ports.

After you press **[ENTER]** for this selection, the controller begins reading in rows of data from the COMM port. During the upload, a counter displays the number of rows created. If the count exceeds 99999 then the displayed count becomes ####x where #### are the four most significant digits and ‘x’ represents a place holder for the least significant digits.

You can press **[CLR]** to abort the process. Normally, you should not do this until the transmission has first been halted at the source. If you do not press **[CLR]**, the controller will continue processing data until it receives a “^Z” (DOS end- of-file character).

The format required for the data is a comma delimited ASCII format, with a carriage return and optional line feed after each row. This format is compatible with the “Down-load” database command. Thus, backed up databases can be reloaded easily into the controller using this selection

This command is much like the “Upld. new” command, except that before a row is created, the database is searched to verify whether the row already exists. This procedure prevents duplicate rows from being created. However, this can slow down the upload significantly, especially for larger databases. Thus, the “Upload New” selection should be used whenever possible for faster response.

B. An entry followed by the **[ENTER]** key:

You do not have to enter data before you press **[ENTER]** for this selection. If you do, the “ENTRY ERROR” message is displayed briefly.

SORT DATABASE (FUNCTION 18)

A. **[ENTER]** key alone:

The “Sort Dbase” command enables sorting of the currently selected database. After you press **[ENTER]** for this selection, the controller begins sorting the current database based on the current column selection in a lowest to highest order. If an exact match of data in the current column is found in two records, then the SORT function automatically compares the data in additional columns, starting with the first column through subsequent columns, until a non-matching field is found. During the sort, a counter that indicates the number of rows that have been sorted is displayed.

You can press **[CLR]** to abort the process. If you do not press **[CLR]**, the controller will continue sorting until it reaches the end of the database.

If you are strictly sorting numeric values, it is best to put the values in a *numeric variable*. The numeric values will be sorted in numerical order. Numbers placed in a data string will be sorted as a DOS sort, for example:

1

10

11

2

20

21

22

If alphanumeric characters are used, a data string must be used as the sort column. The data will be sorted as a DOS sort, for example:

ABC-1.DOC

ABC-10.DOC

ABC-11.DOC

ABC-2.DOC

ABC-20.DOC

The “Sort Dbase” command is useful for generating reports that are printed in some way other than the order in which the rows were created. However, this operation can be quite slow, depending upon both the number of records being sorted and the current order of the records. As an extreme example, a completely disordered database with thousands of records could take a few hours to sort!

Sorting a database can occur after working hours or any time the controller is not in use. To do this, you can set an Alarm to invoke a macro that, in turn, will perform the database sort. The controller must be powered up for sorting to occur.

B. An entry followed by the **[ENTER]** key:

If you make an entry, then the database is sorted according to the column whose parameter ID was keyed in. This is simply a quicker method of requesting a sort on a column other than the current column. It avoids having to first use the "Set Colmn" command. Also, if you enter a decimal point **[•]** before the parameter ID, the sort direction can be reverse from highest to lowest.

An available feature sorts on a secondary column. This can be used where two rows have identical data in the primary sort column. The parameter ID for the secondary sort column is specified by keying in a comma (,) or space ( ) after the primary sort parameter ID.

Refer to the following examples for clarification on the method of specifying the sort criteria.

Example: Specifying the Sort Criteria

11 [ENTER] Sorts on time / date in ascending order. (oldest to newest)

.11 [ENTER] Sorts on time/date in descending order. (newest to oldest)

80.1, .11 [ENTER] Sorts on STR 1 in ascending order. If an exact match of STR1 is encountered, the matching records are sorted according to time/date in descending order. (newest records first).

.80.1, 80.2, 0 [ENTER] Sorts on STR 1 in descending order. If an exact match of STR 1 is encountered, the matching records are sorted according to STR 2 in ascending order. If an exact match of STR 1 and STR 2 is found, the matching records are sorted according to the stored gross in ascending order.

AUTO TEST (FUNCTION 19)

A. **[ENTER]** key alone:

The "Auto Test" command tests the integrity of the database after each database command is executed. If enabled, the auto-test adds the number of used and unused bytes and compares it to the number of bytes available for the database. If the total does not match, an error is displayed and logged in the macro debug (see page 9-163 for macro debug information).

This command is a diagnostic tool that should only be enabled to trouble shoot database corruption problems.

PRINT FORMAT

When using the "Print dbase" operation, the format for the transmitted information is fixed and will conform to the following rules:

- At the top of the first page, headings are printed followed by a blank line. The headings consist of the names of the parameters. If custom names have been defined (using parameters **P600 - P637** and **P682**) then these defined names will be used, otherwise, the default names of parameters will be used.

- The width of each column of data is set to one plus the larger of the data size and the column size.
- A space is transmitted between each column to separate the columns. All widths specified below do not include the space between columns.
- The normal printed data size for numeric variables is eight characters. However, if the value is too large to be represented in eight characters, then additional characters are sent for that row. This will disrupt the alignment of the rest of the data in that row. Data for which units are appropriate will have the data converted to the current units as previously selected in the weigh modes and the units name will be appended.
- The format of the time/date parameter is dependent upon **P503** for the 12- hour or 24-hour format, and upon **P504** for the USA or International date format (Variable type must be set to a U-INT at P686 and style at P688). The width of time/date is 20 characters for 12-hour format and 17 characters for 24-hour format.
- The width of the VARs (alpha-numeric string type) will be the maximum size as set in their respective setup modes, **P689**.
- Every 55 lines, a form-feed <FF>, character will be sent. Immediately afterward, the headings also will be resent. Also, after the last record another form-feed will be sent.

PART #	DESCRIPTION	FINAL WT.	MAX %ERROR	TM/DT
12-345-67	Transistor	0	0	04:39:23 pm 08/16/95
32-20-3020	Capacitor	0	0	08:06:08 am 08/16/95
398-002	Resistor	0	0	08:06:28 am 08/16/95
939-9-321	Diode	0	0	08:06:50 am 08/16/95
24-3456-78	Microprocessor	0	0	11:02:59 am 08/16/95

DOWN-LOAD FORMAT

The “Down-load” command, selection 13, is used to transmit the contents of one of the controller databases in a comma delimited, ASCII file. This format is suitable for uploading the information into a computer database or spreadsheet or for transmitting to another GSE 60 Series instrument using the “Up-load” selection. either the “Upld. new” or “Upld. Updat” selection.

The following rules describe the format of the downloaded data:

- Each column is separated by a comma.
- Each record is terminated by a carriage return and line feed.
- Alpha-numeric parameters (Strings) are enclosed in quotation marks (“ ”).
- All parameters are sent in a minimum width format, with no leading or trailing zeros or spaces.

- After the data for all of the stored rows has been sent, the end of the transmission is indicated by the transmission of a row with only “ENDofDB” (without quotation marks).
- The units for weight data is strictly in default weight units, as defined in the setup mode, **P150**, of the controller. However, the units descriptors, such as “lb” or “kg,” are not sent along with the data.
- All of the transmitted data consists of ASCII characters, that can be printed as text, but no graphics.

Most parameters are downloaded just as they would be viewed. However, the time/date is handled differently.

The data sent for the time/date parameter is the ASCII representation of the numeric value of the number of seconds elapsed since midnight on January 1, 1970. This is the simplest form to allow calculations on time/date to be performed in other spreadsheet and database programs. Some programs might require some manipulation of this value so the time/date data is usable.

For instance, Lotus 1-2-3 requires the time/date data as the number of days elapsed since the year 1900, with the hours, minutes, and seconds represented as the fractional portion of that value. To achieve this format, the numeric time value received from the controller must be divided by 86400 (number of seconds per day) and then add to 25569 (number of days between the years 1900 and 1970). This value can then be displayed in one of the six display formats depending on the requirements. Format 1 for instance, would produce the date in the form “02-Apr-92” while format 6 would display the time in the “12:05:47 pm” format.

A sample of a database download file is shown below.

“12-345-67”,“Transistor”,0,0,702059963
“32-20-3929”,“Capacitor”,0,0,702029168
“398-002”,“Resistor”,0,0,702029188
“939-9-321”,“Diode”,0,0,702029210
“24-3456-78”,“Microprocessor”,0,0,702039779

UPLOAD DATA FORMAT

The same rules for the format of a downloaded file apply to a file being uploaded into the controller database. However, the alpha parameters are not required to be enclosed in double quotation marks (“ ”).

Before you can load any database into a 60 Series database, the setup must be established to match that of the file. If the uploaded file has more columns than the database setup in the controller, the additional columns being uploaded will be ignored. If the uploaded file has fewer columns, then the additional columns will be cleared out.

TIME/DATE HANDLING

When parameter 11, the current time/date parameter, is selected to be a column in a database, then whenever a row is created or updated, the current time/date is copied into that row from parameter 11. However, when a recall is performed, the recalled time/date is not copied to parameter 11 as this would disrupt the current time/date of the controller.

Therefore, when a recall is performed on a database which includes the time/date parameter 11) as a column, the recalled time/date is ignored. If the time is required to be recalled it is recommended that it is first copied into a variable and then create the row. The variable could be a U-INT type if the recalled time is to be left in its original raw form. If a string type variable is used the data could be formatted and copied to the string in a form such as 09\12\96 before the row is created. Once this parameter is recalled it can then be printed using the Custom Transmits or accessed to view the recalled time/date.

The time/date is handled in a special manner which is somewhat common in computers. It is stored as the number of seconds elapsed since midnight on January 1, 1970. Currently this value is around 800,000,000. At the rate of 86400 seconds per day (60 x 60 x 24), (31,536,000 per year) this value changes quickly.

Keeping track of time and date in this manner simplifies the process of calculating the difference between two times. It also compacts the information such that it can be stored in 4 bytes of memory, helping to keep the number of memory bytes required to store a row as small as possible.

DATABASE INFORMATION PARAMETERS

The database information parameters begin at P60010 → P60053. The information parameters supply information about the amount of memory installed, available, already used for by the database option, number of rows used, etc. These parameters are described in detail in Chapter 14, *Information Parameters* beginning on page 14-2.

DEFAULTING THE ENTIRE DATABASE

P65010 can be used to completely clear out the controller database data and setup. However, the deletion is not final until you exit the setup mode and save all changes.

Use the following procedure to clear out the database.

- Press **[ENTER]** to simply default the database setups. Whether or not any database memory is allocated to general usage is not affected.

Allocating database memory to general usage can be used to allow variably sized setup modes (such as Custom Transmit setup and macro setup) to increase when they would otherwise be limited by the occurrence of the “Out Of RAM” error message. If this is done, the only limitation on the amount of information entered into these setup modes is the amount of available EEPROM memory.

MEMORY CONSUMPTION

When a database RAM option is completely blank, **P60020** will indicate that 14 bytes are in use. This amount is due to the initialization of the module to make it usable for database operations.

You can approximate the amount of memory required for your particular database application. Simply access P60033 *before* and *after* you define a database to determine the number of bytes that the database definition has consumed. Then refer to P60034. Compare the number of bytes that a single record (for example, a row) will consume if the strings of that database were of their maximum size (as defined by P689 for each string).

Alternatively, you can use the following formulas to calculate each database's memory requirements.

To calculate the total database memory consumption:

1. Overhead = (6 X Number of Columns) + 30 + Name Size
2. Bytes/Row = (4 X Number of Columns) + 10 + String Bytes
3. Total = Overhead + Bytes/Row

To calculate the Name Size:

1. Add 5 to the number of characters in the name given at P700 and then round up to the nearest even number.
2. If the result of step 1 is less than 8, round up to 8.
3. If there is no database name, then **Name Size** = 0.

To calculate the String Bytes:

1. Round up the maximum string size given at P689 (or the number of characters entered at P143 → P145 if using the status parameter 97P) to the nearest even number and add 4.
2. If the result of step 1 is less than 8, round up to 8.
3. Repeat steps 1 and 2 for each string in the row, accumulating each result as the total number of **String Bytes**.

This method for calculating String Bytes is a worst-case scenario assuming all stored string data is of maximum string size. In reality, the number of bytes required to store each individual string is dependent on the individual string size.

DATABASE ERRORS

If an error occurs during the execution of any database command, the corresponding error message displays for one second. Also the error status of the command is recorded so a macro can be programmed to react appropriately if a database error occurs. However, if a macro is performing the recall operation, error messages will not be displayed. Therefore, it is the responsibility of the person programming the macro to check for possible errors at the appropriate times to ensure that the operation will occur as expected. Refer to the *If Database Error* on page 9-91.

NOTES REGARDING THE STORING AND RECALLING OF WEIGHT DATA

If the default weight units **P150** are changed on the controller, any weight data stored in a database is not affected. However, if it is later printed or recalled, the controller assumes that the stored data is in the current default units of measure. For this reason, GSE recommends that you do not change the default units of measure if records are stored in a database that contains weight data.

The value stored in memory for a weight parameter (Gross, Net, Tare, Gross Total, and so on.) is the value rounded off to the nearest weight increment as defined in setup parameter **P111**.

When rows that contain actively calculated weight data are recalled, the recalled data is overwritten as soon as the next weight conversion process occurs, which is immediately unless a macro is running which is postpones the weight conversion process. Therefore, these actively calculated weight parameters will not normally be included in a database if the created records are going to be recalled. If this is a requirement, the macro, which is written to perform the recall operation, should immediately copy the weight data to another parameter if the recalled weight data will be needed.

The actively calculated parameters include Gross, Net and Quantity.

DATABASE EXAMPLES

The following setup combines the database and macro capabilities for establishing and recalling piece weights. If you press **[ID]**, you will be prompted to enter a part number. Key in the part number and press **[ENTER]**.

If the piece weight associated with that part number does not exist, the controller lets you know that the piece weight does not exist by prompting “NOT FOUND” briefly. The controller then prompts “Add Smple.”

Place the sample on the scale and press **[ENTER]** for the default sample size, or key in a sample size and then press **[ENTER]**. A row is created in the database. The controller is then placed in the quantity mode. If the piece weight exists, the controller prompts “DATA FOUND” . The controller is then placed in the quantity mode automatically.

Example #1: Recall piece weights per part numbers.

This example is structured as an ASCII text upload file.

```
100%s23640%i%e Access Setup Modes, Allowing Changes
125%s1.000000%e    P125.XX  ErFac  1.00
```

Parts Counting Parameter Setup

```
179%s1%e   P179.01 Count on
180%s0%e   P180.00  ASmpl off
181%s0%e   P181.00  AEnhn off
182%s10%e  P182.10  SmpSz    10
183%s98.0000%e P183.XX  %%Accy 98.00
184%s0%e   P184.00  AcDsp off
```

```
186%s0%e   P186.00  PreSm None!
187%s0%e   P187.00  AftSm None!

Name ID #1
681%s1%e   P681.-- Specify VAR#1
682%sPART#%e   P682.-- NAME, PART#
686%s0%e   P686.-- Float Type.
```

Setup Database Structure

```
806%s2%e   P806.02  iduse: dbase

699%s1%e%e Enable database #1
700%s%PARTS%e  dbase #1 name
701%s80.1%e Column 1, PART#
702%s34.1%e APW  scl#1
```

Macro “1” Setup

```
9990%s1%e  P9990.XX  Macro #1
9991%sEnterID#%e   P9991. Name Macro#1
9993%s1%e  P9993 Add/macro menu
P19999.1%s%c%e P19999.1 macro table
%%\%e   0001 if no entry
%%N%e   0002 if not
%%[%e   0003 save entry
80.1%%i%e  0004 VAR#1 (str. type)
%%]%e   0006 get entry
%%e%e   0007 enter
%%E%e   0008 end if        more =>
1,1%y   recall row
4%%_%e  0018 if dbase error
%%N%e   0020 if not
DATA FOUND%%P%e    0021 pause
30%%s%e 0032 select
%%B%e   0035 break
%%E%e   0036 end if
NOT  FOUND%%S%e    0037 sound beeper
%%P%e   0048 pause
%%p%e   0049 sample
1%%T%e  0050 tag position
Add  Smple%%G%e    0052 get operator entry
%%p%e   0063 sample
1C%%^%e 0064 call macro1
1%%g%e  0065 if sample error
1%%J%e  0067 jump to tag
%%E%e   0069 end if
2%%T%e  0070 tag position
2C%^%e  0072 call macro2
%%a%e   0073 if accurate
3,1%y   make row in db#1
-OK!-%%P%e  0083 pause
%%B%e   0089 break
%%E%e   0090 end if
%%Y%e   0091 if yes
%%N%e   0092 if not
ABORTSMPLE%%S%e    0093 sound beeper
%%P%e   0104 pause
%%B%e   0105 break
%%E%e   0106 end if
%%e%e   0107 enter
2%%J%e  0108 jump to tag
```

Macro “2” Setup (“call” subroutine)

```
9990%s2%e  P9990.XX  Macro #2
P19999.2%s%c%e P19999.2 macro table
3%%g%e  0001 if sample error
4%%g%e  0003 if sample error
5%%g%e  0005 if sample error
%%B%e   0007 break
```

```
%%E%e   0008 end if
6%%g%e  0009 if sample error
ABORTSMPLE%%S%e    0011 sound beeper
%%P%e   0022 pause
%%B%e   0023 break
%%E%e   0024 end if%z          Exit Setup Mode
```

Chapter 12 PROGRAMMABLE DIGITAL I/O

The Programmable I/O feature allows 660 Series controllers to process various digital inputs and provide programmable digital outputs. The processor has a separate TPU (Time Processing Unit) 'coprocessor' built into it. This allows the controllers to process digital inputs and outputs in background, without supervision by the main CPU.

The 460 Series indicators do not have PDIO capability.

OVERVIEW

APPLICATIONS

ENCODERS

Encoders are devices that put out a pair of pulse streams that are used to measure the rate and direction of travel of a device. These are most commonly used in rotating devices to measure angular speed but are also used in linear devices for precisely determining a relative location of a device. The Quadrature Decode function may be used to process the signal from a 2 or 3 wire encoder device and provide both position and direction information.

FLOW METERS

Many applications use flow meters to measure the rate of flow of a liquid. These meters generally put out a stream of pulses whose rate is proportional to the rate of flow of the material.

You can connect these flow meters directly into a 660 Series controller, allowing you to pull together several vital system parameters. The total pulse count would provide the amount of material that has flowed (i.e. number of gallons), while a frequency measurement would indicate the rate of flow (i.e. GPM, gallons per minute).

Some applications may use the frequency or pulse feature to measure the rate of flow or total amount of one ingredient such as water, and then use a scale input to measure and control the rate of other ingredients.

CONVEYORS

The speed and overall total travel of a conveyor may be determined using this feature.

DIVERTERS

This data can be used in check weighing applications to actuate the appropriate diverter downstream from an in-line conveyor scale.

FLOW RATE

Continuous flow of material applications which are attempting to control or monitor the amount of material/length (i.e. lb/ft) or material/time (lb/sec) on a conveyor can make use of this feature in determining the speed of the conveyor.

CONNECTIONS

The option I/O terminal block (J11) on the main board is an 11 position lever connector. It provides +5v, +24V and ground along with connections for eight independent external devices.

The connector is a spring-loaded lever connector and accepts 28-12 AWG stranded or solid wire. Press down on the lever, insert the wire into the hole, then release the lever. It is not necessary to tin the stripped wire. If multiple wires are needed in one position it is imperative that they first be securely twisted together before inserting them into the terminal block.

Each of the 11 positions are labeled on the circuit board for easy identification.

Each I/O is automatically configured to operate as an *input or output,* depending upon the function assigned to that channel.

Each I/O position is pulled up to +5v through a 20KΩ resistor and a diode. This resistor guarantees that an unconnected channel that is configured as an input will be biased to the inactive state. The diode protects the circuitry from damage in case higher voltages are connected to the channel since it becomes reverse-biased in that case.

All I/O configured as outputs are implemented as open drain outputs, (except for the weak 20k ohm pull-up to +5v mentioned above). Thus when an output is deactivated, the only applied signal is the 20k ohm pull-up through the diode to +5v. When an output is activated, it becomes grounded through an FET. At power-up each output will be inactive unless/until it is activated.

The maximum voltage that can be applied to an output is 30 $V_{DC.}$ The maximum current that an output can sink is 30 $mA_{DC.}$

TERMINOLOGY

The following technical terms are used in this chapter:

Waveform A graphical representation of an electrical signal where amplitude (generally voltage) is plotted on the vertical axis versus time on the horizontal axis.

Rising Edge The point where the signal goes from a low to a high.

Falling Edge The point where the signal goes from a high to a low.

Cycle One complete waveform, measured from one location on a signal to the next similar location. For digital signals, this is generally measured from one edge (rising or falling) to the next similar edge.

ms (milli-second) A time measurement of one thousandth (0.001) of a second. 1000 ms equals one second.

ms (micro-second) A time measurement of one millionth (0.000001) of a second. 1000 µs equals one milli-second.

Period One cycle of a signal, often measured in milli-seconds.

Frequency The measure of how many cycles of a signal occur in one second. Frequency is usually measured in Hertz (Hz) which is equivalent to cycles per second. Frequency can be calculated by taking the inverse of the period of a waveform. For example, if a signals period is 2.13 ms (0.00213 seconds), the frequency is 1/.00213 = 469.48 Hz.

Duty Cycle The proportion of a signal that is high as compared to the signal's period. For example, if the period of a signal is 2.5 ms and the high time is 1 ms, then the duty cycle is 1 ÷ 2.5 = 0.4, which can also be expressed as 40%.

I/O PARAMETERS

I/O PARAMETER NUMBERS

All of the available I/O functions (except setpoints) have one or more weigh mode accessible parameters associated with them. These parameters are accessed starting at parameter 50 with the channel number appended as an instance. For example, to access the first parameter for channel 3, parameter 50.3 would be specified. Some functions provide for more than one accessible parameter, in which case these are referenced as parameter 51 and 52. The default name for I/O parameter 50 - 52 are 'PIOAX', 'PIOBX', and 'PIOCX' respectively, where the 'X' indicates the channel number.

REFERENCING I/O PARAMETERS

Any setup mode that allows parameters to be specified, such as analog output (P17X), selectable parameters (P30X), Input Interpreter (P222), Database columns (P7XX), custom transmits (P1XXX), setpoints (P51XX), etc may specify these parameters. Macro commands that allow a parameter reference may also use these parameters.

VIEWING I/O PARAMETERS

To view any of these parameters, enter the parameter number, a period, then the channel number, followed by **[SELECT]**. If a name has been programmed for that parameter then the name is displayed in the dot matrix portion of the display.

SCALING I/O PARAMETERS

Since many of the input values may require a conversion factor to make the data more informative, another parameter (typically a Var) may be specified as a scaling factor for all input signals. When this is done all references to the I/O parameter will be referencing the scaled value. For example if a certain flow meter provides 200 pulses per gallon of flow, the Var specified for the scaling factor could be programmed with the value of 0.3 (1 sec ÷ 200 x 60 sec/min) and the I/O parameter could be named GPM. Then referencing the I/O value would provide GPM directly instead of pulses per second.

SETUP AND OPERATIONAL DETAILS

P850 **"Chan#" Channel Number**: Specifies which I/O channel is being setup, i.e. the 'Instance' number. Valid choices are channels one through eight.

P851 **"FuncX" Function for Channel X**: Specifies which function will be running on the current channel. The choices are detailed in Table 12-1.

Table 12-1: Programmable I/O Channel Type Selections

CHOICE #	CHOICE NAME	INPUT OR OUTPUT	FUNCTION NAME	DESCRIPTION
0	None	-----	-----	Specifies that a particular channel is not being used.
1	FqOut	Output	Frequency Output	Specifies that a channel is an output that has programmable frequency and duty cycle.
2	Setpt	(1) Input or (1) Output	Setpoint	Specifies that a channel's operation is based upon setpoint 13X, where 'X' is the channel number For instance channel three would be tied to setpoint 133. The setpoint's setup determines whether the channel is an input or an output.
3	FqInA	Input	Frequency Measurement Method 'A'	Specifies that a channel will be used as an input of digital pulse data. This can be used to determine the frequency (pulses per second) of an incoming signal. Method 'A' counts pulses over a programmable time period.

CHOICE #	CHOICE NAME	INPUT OR OUTPUT	FUNCTION NAME	DESCRIPTION
4	FqInB	Input	Frequency Measurement Method 'B'	Specifies that a channel will be used as an input of digital pulse data. This can be used to determine the frequency (pulses per second) of an incoming digital signal. Method 'B' measures the elapsed time for a programmable number of pulses to occur. This function also accumulates the total number of pulses
5	PhsTm	Input	Phase Time	Allows for measuring the average time for either the high or low portions of an incoming signal. This can be used to determine duty cycle if the frequency is known.
6	DlyIn	Input	Delay Pulse	Allows an output pulse to be time delayed from an input pulse. This function uses two I/O channels and will automatically assign the DlyOt function to the next higher channel.
7	DlyOt	Output	Delay Pulse	Allows an output pulse to be time delayed from an input pulse. Multiple DlyOt channels can be associated with a single DlyIn channel by assigning the DlyOt function to subsequent, adjacent channels.
8	QdDc2	(2) Inputs	Quadrature Decode 2 Wire	Specifies that an encoder's output is connected into this channel and the next channel. This function uses two input channels to determine the direction and amount of travel of an encoder sensor.
9	QdDc3	(3) Inputs	Quadrature Decode 3 Wire	Specifies that an encoder's output is connected into the channel. This function uses three channels. It operates the same as QdDc2 except that the third wire provides for an index signal to determine an absolute or home position.
10	FDbnc	Input	Frequency Debounce	Eliminates unwanted high frequency "noise" such as that caused by the contact closure of a mechanical switch.

The subsequent setup parameters are dependent upon which of the above functions has been chosen for each channel. Below is a list of possible parameters.

P852: **"XNam"**: Provides for naming the 1st I/O parameter for the current channel. When this setup parameter is accessed, the 'X' preceding the 'Nam' is changed to the first letter of the descriptor of the parameter being named, i.e. 'CNam' for Count Name. This parameter applies to FqOut, FqInA, FqInB, PhsTm, DlyOt, QdDc2, & QdDc3.

P853: **"YNam"**: Provides for naming a 2nd I/O parameter for the current channel. This parameter applies to FqOut, FqInB, DlyOt, & QdDc3.

P854: **"ZNam"**: Provides for naming a 3rd I/O parameter for the current channel. This parameter applies to QdDc3 only.

P857: **“CSrc” Clock Source**: Selects clock A or clock B. Clock A’s frequency is 65,536 Hz, Clock B’s frequency is 524,288 Hz. (Note that the ratio of clock B to clock A is 8:1.) A higher clock rate will provide more accuracy for some functions but could also shorten the duration of the measurement period for some functions. Review the notes for the desired function for more information on the specific effects of the clock source on that function. This parameter applies to FqOut, FqInA, PhsTm, & DlyOt.

P858: **“Edge” Edge Type**: Selects rising or falling edge as the reference edge for the current channel. This parameter applies to FqInA, FqInB, DlyIn, & QdDc3.

P859: **“Perd” Measurement Period**: Specifies the duration of the measurement window in milliseconds. This parameter applies to FqInA.

P860: **“#Pls” Number of Pulses**: Specifies the number of pulses (i.e. periods) to measure. This parameter applies to FqInB, & PhsTm.

P861: **“MTyp” Measurement Type**: Selects either low time or high time. This parameter applies to PhsTm.

P862: **“PTyp” Pulse Type**: Permanently set to “sink” (sinks to ground with active output). This parameter applies to DlyOt only.

P863: **“SFac” Scaling Factor**: Provides for specifying a parameter whose value is used as a scaling factor for the I/O parameter for the current channel. For example, this allows pulse inputs to be pre-scaled for use directly in the desired units, i.e. lb, gallons, liters, etc. This parameter applies to FqInA, FqInB, PhsTm, QdDc2, & QdDc3.

P864: **“# DP” Number of Decimal Places**: Specifies the number of decimal places to which the I/O parameter should be displayed. Choices are between 0 (none) and 5 decimal places. This parameter applies to FqInA, FqInB, PhsTm, QdDc2, & QdDc3 but only if P863 for that channel is has a selection other than ‘None’.

FUNCTIONS

Following is a description of the setup, operation, and usage of each of the functions listed in Table 12-1.

NONE: (FUNCTION 0)

A channel that is not being used. No further setup is required for such a channel. Selecting this choice when a channel is not being used will keep the loading of the TPU of the 660 to a minimum thus insuring optimal performance.

The default selection of choice 2, ‘Setpt’ has been made to ensure backward compatibility with older 650 indicators.

FREQUENCY OUTPUT: (FUNCTION 1)

This function allows the 660 to output a digital signal with variable frequency and/or duty cycle. This capability may be used along with some appropriate external devices to control the speed of a conveyor motor, the amount a valve is opened or closed, or any other function normally requiring an analog signal. However, due to the nature of digital data, the output signal is less susceptible to electrical noise than an analog signal.

The setup parameters that pertain to the 'FqOut' selection are:

P852: "FNam": Allows a name to be assigned to I/O parameter 50 which is used to specify the output frequency for this function.

P853: "DNam": Allows a name to be assigned to I/O parameter 51 which is used to specify the duty cycle of the output frequency for this function.

P857: "CSrc": Specifies which clock will be used to generate the signal specified for this channel. The effects of the two alternatives are shown in Table 12-2.

Table 12-2: Frequency Output Based on Clock Source

	Clock A	Clock B
Clock Frequency	**65,536 Hz**	**524,288 Hz**
Clock Period	**15.26 ms**	**1.91 ms**
Minimum Generated Frequency with Full Duty Cycle Range	**2 Hz**	**16 Hz**
Absolute Minimum Generated Frequency (Duty Cycle = 50%)	**1 Hz**	**8 Hz**
Absolute Maximum Generated Frequency (allows 50% duty cycle only)	**32768 Hz**	**262,144 Hz**
Maximum Recommended Generated Frequency (allows 1% duty cycle resolution)	**655 Hz**	**5242 Hz**

The general restrictions listed in Table 12-2 can be attributed to the following facts:

1. The period and high time of the generated signal can only be a multiple of the period of the selected clock.

2. The maximum high and low times are 32,768 periods each.

The ‘weigh mode’ parameters available for this function are:

P50: Specifies the frequency of the output signal.

When a new value is entered for the frequency, it does not take effect until the duty cycle value is entered. This is designed to ensure that both the frequency and the duty cycle take effect at the exact same time. Accessing parameter 50 will provide the frequency in effect at that time, not a recently entered frequency which will not take effect until the duty cycle is changed. Therefore the display does not reflect an entered value for P50 until P51 has subsequently been entered.

The output frequency is displayed to one decimal place. The displayed value is the actual output frequency which may differ slightly from the specified frequency due to the resolution and/or range of the specified clocks.

The name used for displaying and transmitting P50 is specified by P852. If no name is specified, PIOA’X’ is used, where ‘X’ is the channel number.

P51: The duty cycle of the output signal. This is displayed as a value between zero and one, shown to five decimal places. A value of 0.50000 would indicate 50% duty cycle, i.e. a signal high half the time and low half the time.

The displayed value is the actual duty cycle which may differ slightly from the specified duty cycle due to the resolution and/or range of the specified clocks. The name used for displaying and transmitting P51 is specified by P853. If no name is specified, PIOA’X’ is used, where ‘X’ is the channel number.

Example

The frequency output can be used to generate an output signal that represents a signal within the 660, such as the gross weight. Either the frequency or the duty cycle can be used to represent the value of the 660’s parameter, depending on the requirements of the device that will be processing that signal. Varying the duty cycle is the recommended alternative since the update rate remains constant.

To set the frequency output’s duty cycle to be a representation of the gross weight, first determine how often you would like the output updated. For example, to provide a new updated output 20 times per second, set the frequency to 20 Hz.

Next, determine if the selected frequency will provide the desired resolution for the represented signal. Since a 20 Hz signal has a period of 50 ms, and 20 Hz is above the “Minimum Generated Frequency with Full Duty Cycle Range” shown in the chart above, Clock B may be specified for P857. Since clock B’s period is 0.0019 ms, the resolution of the signal will be 26178. (50 ms / 0.00191 ms). This should provide sufficient resolution for most applications.

Finally, to implement the duty cycle output, a macro must be programmed to copy the necessary values to the output registers. Upon power-up a macro should be programmed to copy 20 to the frequency value: (We will assume that channel 6 is being used for the FqOut function.)

50.6P=20%o

If the capacity of the scale is 500 lb, then a macro must be programmed to divide the gross weight by 500 and copy the result to the duty cycle register:

51.6P=0.0P/500%o

This macro could then be programmed to be executed at a rate acceptable to the specific application through the use of a setpoint timer.

SETPOINT (FUNCTION 2)

The setpoint mode of operation has no additional setup parameters. The operation of the setpoint is governed by the associated setpoint, 131 through 138, as determined by the channel number. Refer to parameter P5100 for the specific setpoint being used.

When this choice is made, the I/O channel operates the same as it has operated in earlier firmware releases, before the Programmable I/O feature was released.

FREQUENCY MEASUREMENT METHOD 'A' (FUNCTION 3)

COUNT PULSES OVER A FIXED TIME PERIOD

This frequency measurement method determines the frequency of the input signal by counting the number of specified edges (i.e. rising or falling edges) over a programmable time period. It provides a guarantee of a new calculated frequency at a regular, specified interval. It is useful for applications which need a continually updated indication of the input frequency regardless of how high or low the frequency is.

The setup parameters that pertain to the 'FqInA' selection are:

P852: "FNam": Allows a name to be assigned to I/O parameter 50 which is used to read the input frequency determined by this function.

P857: "CSrc": Specifies which clock will be used to time the period used to measure the frequency for this channel. The effects of the two alternatives are shown in Table 12-3.

Table 12-3: Frequency Measurement Method 'A' Output Based on Clock Source

	Clock A	Clock B
Clock Frequency	**65,536 Hz**	**524,288 Hz**
Clock Period	**15.26 ms**	**1.91 ms**
Minimum Measurement Period	**0.015 ms**	**0.0019 ms**
Resolution of Measured Frequency using Period Listed Above	±**65 kHz**	±**524 kHz**
Typical Measurement Period	**100 ms**	**10 ms**
Resolution of Measured Frequency using Period Listed Above	±**10 Hz**	±**100 Hz**
Maximum Measurement Period	**500 ms**	**62.5 ms**
Resolution of Measured Frequency using Period Listed Above	±**2 Hz**	±**16 Hz**
Maximum Input Frequency using Period Listed Above	**131,070 Hz**	**1024 kHz**

The general restrictions listed in Table 12-3 can be attributed to the following facts:

- The maximum measurement period is 32,768 clocks.
- The function can only count complete pulses (i.e. periods). As an example, a change of one pulse in a 100 ms period results in a frequency change of 10 Hz. (i.e. 1 pulse in 100 ms = 10 Hz, 2 pulses = 20 Hz, etc…).
- The maximum number of pulses that may be counted in one measurement period is 65,535. Therefore, the period should not be set longer than the time required for 65,535 pulses to occur at the maximum expected frequency. This should not be a concern in most applications.

  As shown in Table 12-3, most applications should work well with 'Clock A' specified and a long measurement period. A period of 16.67 ms will result in the frequency being calculated at about the same rate as the weight is re-calculated, i.e. about 60 Hz.
- **P858: "Edge":** Specifies whether a rising or falling edge is used as the reference edge for counting pulses. For most applications the setting for this parameter will not have any noticeable effect. However, if pulses are occurring sporadically and infrequently it may become desirable to detect a specific edge, either rising or falling.

- **P859: "Perd":** Specifies the period of time, in milli-seconds, over which pulses are counted. Reference the table provided with P857 for the valid range of entries which is dependent upon the setting of P857.

  If you attempt to exit the setup mode with a period that is too large for the selected clock, the warning "Per'd > Max" will appear briefly. Press any key and the indicator will display P859, showing the invalid selection. Then change the period to a valid value.

  Press **[CLEAR]** to default the period to the maximum value based upon the current selected clock for this channel, as specified by P857.

- **P863: "SFac":** Specifies which parameter is intended to provide a scale factor for the input frequency. This allows the input signal to be converted to a value with meaningful units such as GPM (gallons per minute), FPM (feet per minute), etc. To specify a scaling factor, first allocate a Var for this purpose, then key in 80.X **[ENTER]** where 'X' is the Var number to be used. (Although we allow parameters other than Vars to be used as scaling factors, we do not expect parameters other than Vars to be used here.)

- **P864: "# DP":** This parameter is only available if P863 is *not* set to 'None'. It specifies the number of decimal places shown when viewing or transmitting the scaled frequency measurement. When this parameter is not available, zero decimals are used for the measured frequency.

The 'weigh mode' parameter available for this function is:

- **P50:** The frequency of the input signal. This parameter will be updated at the interval specified by the measurement period, P859. If P863 is not 'None', then the value provided by P50 will be the measured frequency multiplied by the specified scaling factor. Otherwise 'FqInA's P50 will show zero decimal places.

  The value of this parameter may not be entered. It is only a calculated value. Any entries or macro assignments to this parameter will be ignored.

The name used for displaying and transmitting P50 is specified by P852. If no name is specified, PIOA'X' is used, where 'X' is the channel number.

'FqInA' Detailed Explanation

The accuracy of this frequency measurement technique is dependent upon the duration of the measurement period and the range of frequencies being measured. Specifying a longer measurement period results in a more accurate measurement for a given frequency, but the longer period causes a lag in the response to a changing frequency. The measurement uncertainty can be calculated using the following formula:

$$\text{Uncertainty} = \frac{1}{MeasurementPeriod}\ \text{Hz}$$

For example with a measurement period of 250 ms measuring a fixed frequency of 10 Hz counting falling edges the error is calculated as follows:

$$\frac{1}{.250} = 4\,\text{Hz}$$

This error will manifest itself as shown in Table 12-4. See also Figure 12-1.

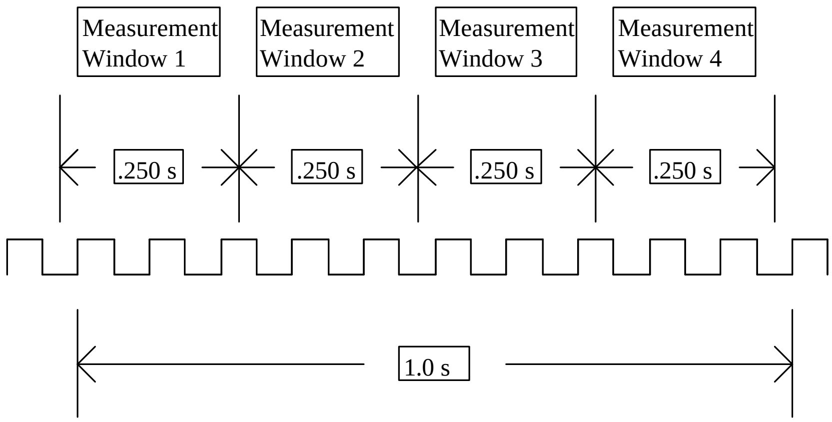


Figure 12-1: Example of Fixed Measurement Method 'A'

For an 11 Hz input signal the error would appear as shown in Table 12-5. Notice the calculated frequency is 8 Hz for only one of the measurement windows compared to Table 12-4 where 8 Hz was calculated for 2 of the measurement windows. If you were to average several sampled frequencies together, you would get the actual frequency. See also Figure 12-2.

Table 12-4: Calculating Error for Measuring Fixed Frequency of 10 Hz

Measurement Window	Pulses Counted	Calculated Frequency
1	2	$\frac{2}{.250} = 8Hz$
2	3	$\frac{3}{.250} = 12Hz$
3	2	$\frac{2}{.250} = 8Hz$
4	3	$\frac{3}{.250} = 12Hz$

Table 12-5: Calculating Error for Measuring Fixed Frequency of 11 Hz

Measurement Window	Pulses Counted	Calculated Frequency
1	2	$\frac{2}{.250} = 8Hz$
2	3	$\frac{3}{.250} = 12Hz$
3	3	$\frac{3}{.250} = 12Hz$
4	3	$\frac{3}{.250} = 12Hz$

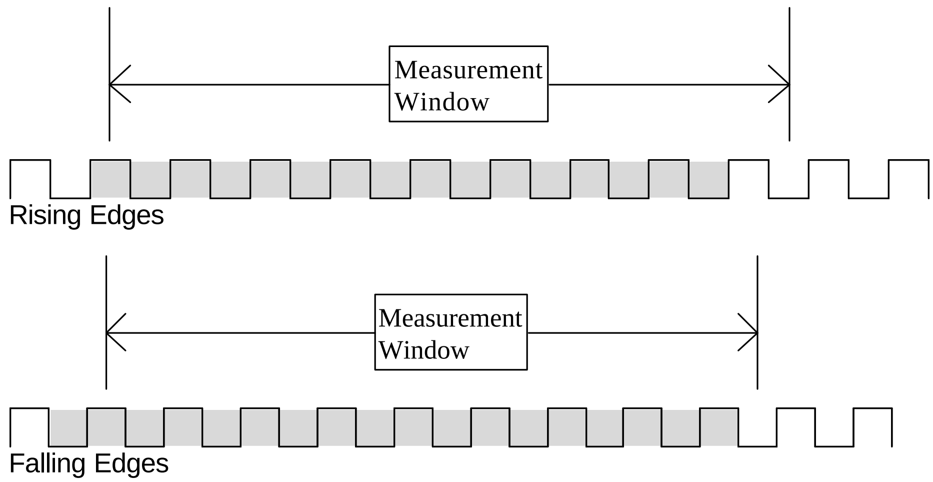


Figure 12-2: Fixed Frequency Measurement Window for Rising and Falling Edges

Note that the error amount is much smaller for the higher frequencies which this function is more suited toward. However, this function is ideal for applications where the frequency will normally be fairly high (such as a conveyor putting out thousands of pulses per second at its normal speed) but which could fall to a low value during a fault condition (if the conveyor slows way down or stops due to a mechanical problem).

FREQUENCY MEASUREMENT METHOD 'B' (FUNCTION 4)

MEASURE ELAPSED TIME FOR A FIXED NUMBER OF PULSES TO OCCUR

This frequency measurement method determines the frequency of the input signal by measuring the elapsed time for a specified number of input pulses to occur. It provides a higher degree of accuracy for the calculated frequency than 'FqInA' since the calculation is based upon complete pulses, whereas the time interval used by 'FqInA' may include a partial pulse, resulting in a round-off error. Also, FqInB accumulates the total number of pulses which may be accessed at any time.

However, the interval at which a new calculated frequency becomes available is not a constant; instead it is completely dependent upon the frequency of the input signal. 'FqInB' is more appropriate than 'FqInA' for applications requiring a high degree of accuracy and/or applications which need to keep a count of the total number of pulses.

The setup parameters that pertain to the 'FqInB' selection are:

- **P852: "FNam"**: Allows a name to be assigned to I/O parameter 50 which is used to read the input frequency determined by this function.
- **P853: "PNam"**: Allows a name to be assigned to I/O parameter 51 which is used to read the total number of pulses counted by this function.
- **P858: "Edge":** Specifies whether a rising or falling edge is used as the reference edge for counting pulses. For most applications the setting for this parameter will not have any noticeable effect. However, if pulses are occurring sporadically and infrequently it may become desirable to detect a specific edge, either rising or falling.
- **P860: "#Pls":** Specifies the number of pulses (or cycles) whose time will be measured. The range of valid choices that may be specified is between 1 and 255. A smaller number of pulses will provide quicker results for a changing input signal while a higher number will provide a more accurate, averaged frequency measurement.
- **P863: "SFac":** Specifies which parameter is intended to provide a scale factor for the input frequency and pulse count. This allows the input signal to be converted to a value with meaningful units such as GPM (gallons per minute) and total gallons, FPM (feet per minute) and total feet, etc. To specify a scaling factor, first allocate a Var for this purpose, then key in 80.X **[ENTER]** where 'X' is the Var number to be used. (Although we allow parameters other than Vars to be used as scaling factors, we do not expect parameters other than Vars to be used here.)
- **P864: "# DP":** This parameter is only available if P863 is *not* set to 'None'. It specifies the number of decimal places that will be shown when viewing or transmitting the scaled frequency and/or number of pulses. When P864 is not available (due to P863 = 'None'), three decimal places are used for the measured frequency and zero decimal places are used for the pulse count.

The 'weigh mode' parameters available for 'FqInB' are:

- **P50:** The frequency of the input signal. This parameter will be updated every time the specified number of input pulses occurs. If P863 is not 'None', then the value provided by P50 will be the measured frequency multiplied by the scaling factor specified by P863 and the number of decimal places used when displaying and transmitting P50 is specified by P864. Otherwise 'FqInB's P50 will show three decimal places.

- This value may be entered, however it will be overwritten once the frequency has been recalculated. This may be useful for detecting when the frequency is updated.
- The name used for displaying and transmitting P50 is specified by P852. If no name is specified, PIOA'X' is used, where 'X' is the channel number.
- **P51:** The total count of pulses of the input signal. Whenever this parameter is accessed, it includes all complete pulses up to that instant.

If P863 is not 'None', then the value provided by P51 will be the total number of pulses multiplied by the scaling factor specified by P863 and the number of decimal places used when displaying and transmitting P51 is specified by P864. Otherwise, zero decimal places are displayed for the pulse count.

This total count may be entered or cleared at any time and the new total will be further incremented by any further pulses.

The name used for displaying and transmitting P51 is specified by P853. If no name is specified, PIOB'X' is used, where 'X' is the channel number.

These parameters may be viewed on the display by keying in the parameter number followed by a decimal point and then the channel number followed by the **[SELECT]** key (i.e. 51.3 **[SELECT]** causes P51 for channel 3 to be displayed). While a parameter is displayed, simply key in a new value and press **[ENTER]**. **[CLEAR]** may be pressed to reset a value to zero.

'FQINB' DETAILED EXPLANATION

The specified number of pulses is divided by the amount of elapsed time (measured with a resolution of 1.91 µs) during those pulses (i.e. periods) to determine the exact frequency of the incoming signal. See Figure 12-3.

However, as the frequency of the incoming signal decreases, the time in between updates of the calculated frequency increases. For example, if the number of pulses to count is set to ten, then if the rate of pulses were to fall to one per second, then the pulse frequency would be updated once every ten seconds.

Since the update rate of the frequency calculated using 'FqInB' is not constant, using a macro to average several sequential values of the frequency will not necessarily result in the average frequency. If a longer term average frequency is desired, it is recommended to record the pulse count (P51) and a timer (P82) (or zero both parameters) at one instant in time, then later subtract the recorded values from the new values and divide the number of pulses by the elapsed time to determine a long term frequency.

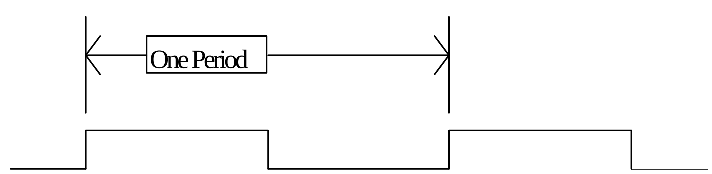


Figure 12-3: Time Period of any Frequency

The accuracy of this frequency measurement technique is dependent upon the frequency of the clock source (used for measuring time), the number of pulses measured, and the frequency being measured.

Specifying a larger number of pulses results in a more accurate measurement for a given frequency. However, the penalty is a longer time period between updates of the calculated frequency. The measurement error can be calculated using the following formula:

$$f_e = f_s - \frac{N * f_c}{\frac{N * f_c}{f_s} + 1}$$

where:

f_e = maximum error in frequency measurement.

f_s = frequency of signal being measured.

f_c = frequency of selected clock “A” or “B”.

N = number of periods to measure.

Table 12-6 illustrates the latency (response time) and maximum error that can be expected for various input frequencies and number of periods measured:

Table 12-6: Latency (Response Time) and Maximum Error

Periods Measured:	1 period		10 periods		100 periods	
Input Freq (Hz)	**Latency (s)**	**Error (Hz)**	**Latency (s)**	**Error (Hz)**	**Latency (s)**	**Error (Hz)**
10	0.100000	0.000191	1.000000	0.000019	10.000000	0.000002
50	0.020000	0.004768	0.200000	0.000477	2.000000	0.000048
100	0.010000	0.019070	0.100000	0.001907	1.000000	0.000191
500	0.002000	0.476383	0.020000	0.047679	0.200000	0.004768
1,000	0.001000	1.903718	0.010000	0.190698	0.100000	0.019073
5,000	0.000200	47.233264	0.002000	4.763828	0.020000	0.476792
10,000	0.000100	########	0.001000	19.037176	0.010000	1.906985
20,000	0.000050	########	0.000500	76.004013	0.005000	7.626485
50,000	0.000020	########	0.000200	########	0.002000	47.638284

Note that there is no error for the pulse count. Every single pulse will be counted within a few microseconds of when the specified edge (per P858) has occurred.

However, the incoming signal must have clean transitions. If the input signal is the result of a switch closure that experiences contact bounce, then the bounces may be detected as extra pulses, resulting in a higher than actual calculated frequency and pulse count.

PHASE TIME (FUNCTION 5)

The Phase Time function will measure the total high or low time for the specified number of pulses. This can be used to determine the duty cycle of an input signal if the frequency or period is known. Also, PhsTm accumulates the total number of pulses which have occurred and this count may be accessed at any time.

The setup parameters that pertain to the ‘PhsTm’ selection are:

- **P852: "TNam"**: Allows a name to be assigned to I/O parameter 50 which is used to read the phase time of the specified portion (high or low time) of the input signal, as determined by this function.
- **P860: "#Pls":** Specifies the number of pulses (or cycles) whose time will be measured. The range of valid choices that may be specified is between 1 and 255.
- **P861: "MTyp":** Specifies whether the measured portion of the input signal is the high or low time of the signal.
- **P863: "SFac":** Specifies which parameter is intended to provide a scale factor for the time measurement of the input signal. This allows the input signal to be converted to a value with meaningful units such as GPM (gallons per minute), FPM (feet per minute), etc.

To specify a scaling factor, first allocate a Var for this purpose, then key in 80.X **[ENTER]** where 'X' is the Var number to be used. (Although we allow parameters other than Vars to be used as scaling factors, most applications would logically use only Vars as scale factors.) Then, in the weigh mode, enter the value of the pre-scale value into the specified Var.

- **P864: "# DP":** This parameter is only available if P863 is *not* set to 'None'. It specifies the number of decimal places shown when viewing or transmitting the scaled phase time and pulse count. When P864 is not available (due to P863 = 'None'), four decimal places are used for displaying the phase time and zero decimal places for the pulse count.

The 'weigh mode' parameters available for 'PhsTm' are:

- **P50:** The high or low time (as specified by P861) of the input signal. This parameter will be updated every time the specified number of input pulses (per P860) occurs. If P863 is not 'None', then the value provided by P50 will be the measured time multiplied by the scaling factor specified by P863 and the number of decimal places used when displaying and transmitting P50 is specified by P864. Otherwise 'PhsTm's P50 will show four decimal places.

This value may be entered; however it will be overwritten once the frequency has been recalculated. This may be useful for detecting when the frequency is updated.

The name used for displaying and transmitting P50 is specified by P852. If no name is specified, PIOA'X' is used, where 'X' is the channel number.

- **P51:** The total count of pulses of the input signal. Whenever this parameter is accessed, it includes all complete pulses up to that instant.

If P863 is not 'None', then the value provided by P51 will be the total number of pulses multiplied by the scaling factor specified by P863 and the number of decimal places used when displaying and transmitting P51 is specified by P864. Otherwise, zero decimal places are displayed for the pulse count.

This total count may be entered or cleared at any time and the new total will be further incremented by any further pulses.

The name used for displaying and transmitting P51 is specified by P853. If no name is specified, PIOB'X' is used, where 'X' is the channel number.

These parameters may be viewed on the display by keying in the parameter number followed by a decimal point and then the channel number followed by the **[SELECT]** key (i.e. 51.3 **[SELECT]** causes P51 for channel 3 to be displayed). While a parameter is displayed, simply key in a new value and press **[ENTER]**. **[CLEAR]** may be pressed to reset a value to zero.

‘PhsTm’ Detailed Explanation

The elapsed amount of high or low time (measured with a resolution of 1.91 µs) during a specified number of pulses (i.e. periods) is divided by the incoming number of periods signal to determine the average high or low time for those pulses.

The Phase Time measurement is intended for use with input signals that have a fixed frequency. Therefore the number of pulses may be determined based upon choosing a reasonable balance between response time and accuracy.

‘PhsTm’ Example

As an example, let us consider a PWM (Pulse Width Modulated) input signal with a fixed frequency of 10 kHz that is being provided by a tank level sensor. We will assume that the level of the tank is proportional to the high time of the PWM signal. Then the value of the time measured by the ‘PhsTm’ function is proportional to the tank level.

If we want a new calculated value 60 times per second, then the number of pulses to be used for each measurement may be calculated as follows:

$\dfrac{\dfrac{10{,}000\,Cycles}{Second}}{\dfrac{60\,Measurements}{Second}}$	= 166.66 Cycles (or Pulses) per measurement. Thus set P860 to 167.

Since the High time is proportional to the parameter being measured, set P861 to ‘High’.

Through the use of a scaling factor, the measured time may be converted into a value with meaningful units for the parameter being measured. Since the frequency is 10,000 Hz, the high time may range from 0/10,000 to 1/10,000 (0.0001) Seconds.

To determine the scale factor, divide the desired maximum value by the phase time that will represent that value. Specifically:

$$\frac{MaximumDisplayedValue}{CorrespondingMeasuredValue} = ScaleFactorValue$$

For example,

To display the tank level as a percentage of its capacity, the maximum displayed value would be 100 (%) and that would correspond to a phase time measurement of 0.1 ms. Therefore, the scale factor value should be 100/0.1 = of 1,000,000.

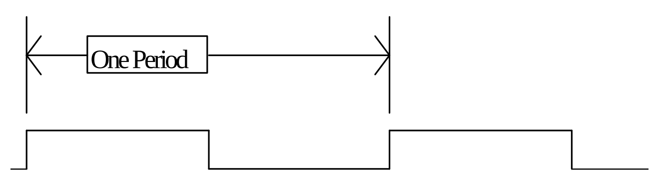


Figure 12-4: Phase Time (The Period)

The accuracy of phase time measurement is similar to the accuracy shown for the ‘FqInB’ function. However since we are measuring time instead of frequency, we will present the error equations in terms of time.

$$ErrorTime = \frac{1}{ClockPerSecond \times NumberOfPeriods} = \frac{1}{524{,}288 \times P860} = \frac{1.91mSec}{P860}.$$

Then if the frequency or period of the input signal is known, the error may be calculated as a percentage of the maximum value of the phase time, i.e. the period. See Figure 12-4. Also refer to Table 12-7.

Table 12-7: Calculated Error Times with Known Input Signal Frequency or Period

Input Frequncy (Hz):		**10**	**100**	**1000**	**10000**	**50000**
Number of Periods	**Error Time (micro-seconds)**	**Uncertainty (expressed as a percentage of signal's period)**				
1	1.907	0.00191%	0.01907%	0.19073%	1.90735%	9.53674%
10	0.191	0.00019%	0.00191%	0.01907%	0.19073%	0.95367%
25	0.076	0.00008%	0.00076%	0.00763%	0.07629%	0.38147%
50	0.038	0.00004%	0.00038%	0.00381%	0.03815%	0.19073%
100	0.019	0.00002%	0.00019%	0.00191%	0.01907%	0.09537%
200	0.010	0.00001%	0.00010%	0.00095%	0.00954%	0.04768%
250	0.008	0.00001%	0.00008%	0.00076%	0.00763%	0.03815%

Delay In, Delay Out (Functions 6 & 7)

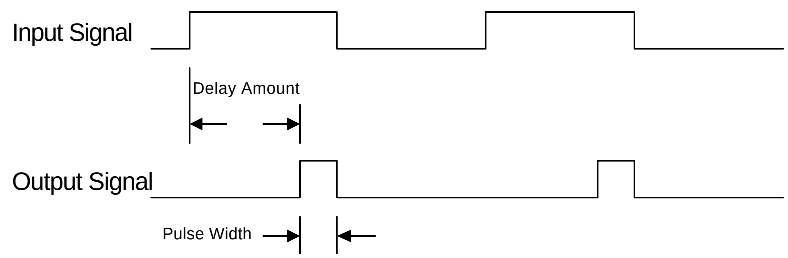
These functions allow an output pulse to be generated that has a programmable width and occurs a programmable delay after an incoming signal. In order to use this feature, one channel (channel 7 or lower) must be programmed to use the ‘DlyIn’ (Delay In) function. The next channel is then locked to be ‘DlyOt’ (Delay Out). Additional Delay Out channels that use the same Delay In signal may be programmed at adjacent channels.

For example if channel 3 is programmed to be ‘DlyIn’ then channel 4 will automatically be set to ‘DlyOt’. Channel 5 could then also be set as ‘DlyOt’ and it would reference the input signal of channel 3. If channel 5 were set to ‘DlyOt’ then channel 6 could also be ‘DlyOt’ and so on. Figure 12-5 shows the relationship between the input and output signals.

The 660 allows “Delay Out” type PDIO channels to be gated by their respective setpoint. I.E. a setup parameter of delay out (P855) is "Stpt" and the choices are "Ignrd" and "Gated". Default is "Ignrd" and in this case the PDIO channel would continue to work as it has. However enabling this parameter would cause the PDIO output to go to its de-active state whenever the associated setpoint is off and work normally when the setpoint is on. This feature is essential for the SCR control option. The associated setpoint is setpoint 131 - 138 for PDIO channels 1 - 8 respectively.

The PDIO output pin will consistently go to the specified (P862) state when the pulses are inhibited by the setpoint going off.

Figure 12-5: Relationship Between Input and Output Signals



The only setup parameter that pertains to the 'DlyIn' selection is:

- **P858: "Edge":** Specifies whether a rising or falling edge of the input signal is used as the reference edge for delaying the output pulse. The diagram above shows a rising edge being referenced.

The setup parameters that pertain to the 'DlyOt' selection are:

- **P852: "PNam"**: Allows a name to be assigned to I/O parameter 50 which is used to specify the pulse width of the output signal.
- **P853: "DNam"**: Allows a name to be assigned to I/O parameter 51 which is used to specify the amount of time delay of the output signal.
- **P857: "CSrc"**: Specifies which clock will be used to time the pulse width and delay time for this channel. The effects of the two alternatives are shown in Table 12-8.

Table 12-8: Selected Clock Specifications

	Clock A	Clock B
Clock Frequency	**65,536 Hz**	**524,288 Hz**
Clock Period	**15.26 μs**	**1.91 μs**
Resolution of Pulse Width and	**0.015 ms**	**0.0019 ms**

Delay Amounts		
Maximum Delay or Pulse Width	**1000 ms**	**125 ms**

The general restrictions listed Table 12-8 can be attributed to the fact that the maximum times for the delay and the pulse width are 65,535 periods of the selected clock.

- **P862: "PTyp":** This parameter has a permanent selection of 'Sink' specifying that the output is normally at a high state, sinking to ground to provide the output pulse.

The 'weigh mode' parameters available for this function are:

- **P50:** Used to specify the pulse width of the output signal. It is shown to three decimal places in units of milli-seconds. This parameter does not take effect until a value for P51 is subsequently entered. The displayed value reflects the value in effect at any given time, even if a new value has been entered and is pending taking effect when P51 is entered. Note that the channel number referenced is that of the channel set for the "DlyOt" function. For example if channel 5 is DlyIn and channel 6 & 7 are set for 'DlyOt', then the first pulse width and delay time would be referenced as P50.6 and P51.6 while the second DlyOt would be P50.7 and P51.7.
- **P51:** Used to specify the delay amount for the output signal. It is shown to three decimal places in units of milli-seconds. When a value for this parameter is entered, the most recent entered value for P50 also takes effect. The maximum delay time which may be specified is dependent upon the clock selected, as shown in the table above.

  If a time longer than that shown for the selected clock is specified, the maximum allowed time is implemented. If a negative time is specified, a time of zero is implemented. The minimum response time for the delay output is around 8 µs (0.008 ms).

'DLYOT' DETAILED OPERATION

When an input signal edge is detected an event is scheduled, based upon the value of P50, using the specified internal clock. This event causes the output signal to change state. When that state change occurs another event is scheduled, based upon P51, which then causes the signal to revert back to its original state.

DELAY FUNCTION APPLICATIONS

The delay in/out function pair can be used for various applications, most notably the triggering of SCRs for vibration control of vibratory feeders.

Vibratory feeders are typically supplied with an AC half wave rectified signal which is supplied by an SCR that fires some variable time after the zero cross of the AC waveform.

A TTL signal must be generated which indicates the zero cross of the AC power source. That signal must be connected to the DlyIn input.

The 'DlyOt' signal may then be connected to a circuit which is used to fire an SCR at the desired phase point of the AC waveform. A 0° (0 ms delay) SCR firing results in maximum vibration while a 180° (8.33 ms @ 60 Hz) firing produces no vibration. Thus varying the delay time between 0 and 8.33 ms provides an adjustable vibratory control.

Appropriate isolation techniques must be implemented to prevent damage to the 660 electronics and to reduce the risk of electrical shock.

QUADRATURE DECODE, 2 WIRE (FUNCTION 8)

The 2 wire quadrature decode feature processes a pair of signals output by an encoder. It can be used to determine the position of a device that rotates or moves back and forth since it simultaneously senses both distance and direction of travel of the device.

Two sequential channels are required on the Programmable I/O connector in order to use the two wire quadrature decode feature. Therefore this function cannot be assigned to channel 8.

The setup parameters that pertain to the ‘QdDc2’ selection are:

- **P852: “Pnam”**: Allows a name to be assigned to I/O parameter 50 which is used to read the current encoder count value, as determined by this function.
- **P863: “SFac”:** Specifies which parameter is intended to provide a scale factor for the encoder count value. This allows the input signal to be converted to a value with meaningful units such as Feet, Meters, Inches, Degrees of rotations, etc.

To specify a scaling factor, first allocate a Var for this purpose, then key in 80.X **[ENTER]** where ‘X’ is the Var number to be used. (Although we allow parameters other than Vars to be used as scaling factors, most applications would logically use only Vars as scale factors.) Then, in the weigh mode, enter the value of the pre-scale value into the specified Var.

P864: “# DP”: This parameter is only available if P863 is *not* set to ‘None’. It specifies the number of decimal places shown when viewing or transmitting the scaled encoder count. When P864 is not available (due to P863 = ‘None’), zero decimal places are used for displaying the encoder count value.

Please note that the next channel of the Programmable I/O port will automatically be set to ‘QdDc2’ and is not changeable. There are no setup or weigh mode parameters associated with the second channel.

The ‘weigh mode’ parameter available for ‘QdDc2’ is:

- **P50:** The current encoder count value. This parameter is updated every time the encoder signal changes. If P863 is not ‘None’, then the value provided by P50 will be the encoder count value multiplied by the scaling factor specified by P863 and the number of decimal places used when displaying and transmitting P50 is specified by P864. Otherwise ‘QdDc2’s P50 will show zero decimal places.

A new value may be entered at any point. Any subsequent pulses input from the encoder will increment or decrement the new value accordingly.

The encoder count value is a 16 bit counter and thus is limited to a maximum value of +32767 and a minimum value of -32768. If the encoder output is allowed to increment or decrement past this value it will wrap around to the other extreme value and continue from there.

When the 660 is first powered up and when setup mode changes are saved the value of this parameter is zeroed out.

The name used for displaying and transmitting P50 is specified by P852. If no name is specified, PIOA'X' is used, where 'X' is the channel number.

This parameter may be viewed on the display by keying in the parameter number followed by a decimal point and then the channel number followed by the **[SELECT]** key (i.e. 50.6 **[SELECT]** causes P50 for channel 6 to be displayed). The value of the parameter may be changed while a parameter is displayed by simply keying in a new value and pressing **[ENTER]**. **[CLEAR]** may be pressed to reset a value to zero.

Of course the parameter's value may also be referenced or changed using the parameter number and instance with the %o macro command, i.e. 50.5P=x%o.

'QDDC2' DETAILED EXPLANATION

An encoder puts out two square wave signals which are 90° out of phase with each other. When the direction of travel is one way then one signal rises before the other. The opposite direction of travel results in the opposite signal rising first. When the encoder is not moving, the signals are stable.

Note that the width of the signals is relevant only to how fast the travel is occurring. The relationship between which signal rises first determines the direction of travel.

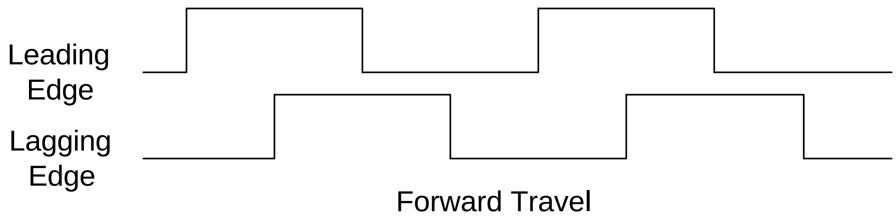
To achieve a proper direction determination, the lagging input should be connected to the lower numbered I/O channel. If the encoder count is going up when you expect it to go down, simply reverse the connection of the two wires.

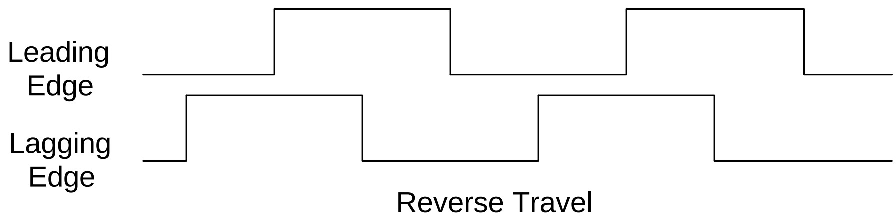
The quadrature decode function counts every edge on the two encoder's signals. When two edges occur on one signal without any edges on the other signal then the direction of travel has changed.

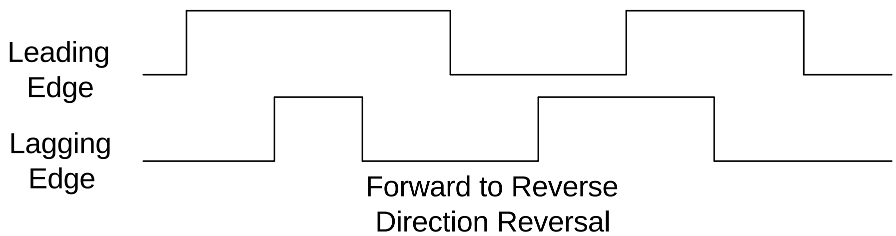
The quadrature decode function is not capable of determining speed. If speed of travel is of interest, one of the two signals may be connected to another input channel and that channel may be programmed with the 'FqInA' or 'FqInB' function to measure the rate of pulses. This method will not distinguish direction of travel. Travel in either direction will result in a positive result. If directional speed is of concern, then a macro may be used to capture the encoder value and a timer. The change in encoder value may then be divided by the change in time to determine directional speed.

See Figure 12-6 for clarification of an encoder's signals.

Figure 12-6: Encoder Signals







'QDDC2' EXAMPLE

When an encoder is connected to a conveyor system, the encoder may be used to determine the distance that the conveyor has traveled since an item was weighed.

To determine the proper scaling factor in order to display the amount of travel in feet, determine how many pulses that the encoder is putting out per foot of travel Enter the inverse of that number as the scale factor. For example, if 250 pulses per foot are occurring, the scale factor would be 1/250 = 0.004. The maximum travel that can be accumulated for this example would be 32767 x 0.004 = 131 ft. The resolution of the measurement would be 0.004 ft which is equal to 0.004' x 12 in./ft. = 0.048".

QUADRATURE DECODE, 3 WIRE (FUNCTION 9)

The three wire quadrature decode feature uses three sequential channels of the eight channel programmable I/O. It functions the same as the two wire quadrature decode function except that a third wire is added to provide an absolute position reference.

Three sequential channels are required on the Programmable I/O connector in order to use the three wire quadrature decode feature. Therefore this function cannot be assigned to channel 7 or 8 as the first channel.

The setup parameters that pertain to the ‘QdDc3’ selection are:

- **P852: “PNam”**: Allows a name to be assigned to I/O parameter 50 which is used to read the absolute encoder count value, as determined by this function.
- **P853: “INam”**: Allows a name to be assigned to I/O parameter 51 which is used to read the value of the encoder count at the time of the last index pulse edge.
- **P854: “RNam”**: Allows a name to be assigned to I/O parameter 52 which is used to read the relative encoder count value. This value is zeroed when the index pulse occurs.
- **P858: “Edge”:** Specifies whether the rising or falling edge of the index signal is used as the reference edge for determining the index count.
- **P863: “SFac”:** Specifies which parameter is intended to provide a scale factor for the three encoder values. This allows the input signal to be converted to a value with meaningful units such as Feet, Meters, Inches, etc.

To specify a scaling factor, first allocate a Var for this purpose, then key in 80.X **[ENTER]** where ‘X’ is the Var number to be used. (Although we allow parameters other than Vars to be used as scaling factors, most applications would logically use only Vars as scale factors.) Then, in the weigh mode, enter the value of the pre-scale value into the specified Var.

- **P864: “# DP”:** This parameter is only available if P863 is *not* set to ‘None’. It specifies the number of decimal places that will be shown when viewing or transmitting any of the three scaled encoder values. When P864 is not available (due to P863 = ‘None’), zero decimal places are used for displaying the encoder count values.

Please note that the next two channels of the Programmable I/O port will automatically be set to ‘QdDc3’. There are however no setup or weigh mode parameters associated with the second and third channel numbers.

The ‘weigh mode’ parameters available for ‘QdDc3’ are:

- **P50:** The absolute encoder count value. This parameter is updated every time the encoder signal changes. If P863 is not ‘None’, then the value provided by P50 will be the encoder count value multiplied by the scaling factor specified by P863 and the number of decimal places used when displaying and transmitting P50 is specified by P864. Otherwise ‘QdDc3’s P50 will show zero decimal places.

The name used for displaying and transmitting P50 is specified by P852. If no name is specified, PIOA’X’ is used, where ‘X’ is the channel number of the first of the three channels used by ‘QdDc3’.

- **P51:** The value of the encoder count when the specified edge of the index pulse last occurred. This parameter is updated every time the index pulse occurs. If P863 is not ‘None’, then the value provided by P51 will be the stored encoder count value multiplied by the scaling factor specified by P863 and the number of decimal places used when displaying and transmitting P51 is specified by P864. Otherwise ‘QdDc3’s P51 will show zero decimal places.

The name used for displaying and transmitting P51 is specified by P853. If no name is specified, PIOB’X’ is used, where ‘X’ is the channel number of the first of the three channels used by ‘QdDc3’.

- **P52:** The relative encoder count value. This parameter is updated every time the encoder signal changes.. This value is automatically zeroed whenever the index pulse’s specified edge occurs. If P863 is not ‘None’, then the value provided by P52 will be the relative encoder count value multiplied by the scaling factor specified by P863. The number of decimal places used when displaying and transmitting P52 is specified by P864. Otherwise ‘QdDc3’s P52 will show zero decimal places.

The name used for displaying and transmitting P52 is specified by P854. If no name is specified, PIOC'X' is used, where 'X' is the channel number of the first of the three channels used by 'QdDc3'.

The values of P50 and P51 may be entered at any point. Any subsequent pulses input from the encoder will increment or decrement the new value of P50 accordingly.

The encoder count value is a 16 bit counter and thus is limited to a maximum value of +32767 and a minimum value of -32768. If the encoder output is allowed to increment or decrement past this value it will wrap around to the other extreme value and continue from there.

When the 660 is first powered up and when setup mode changes are saved the values of these parameters are zeroed out.

These parameters may be viewed on the display by keying in the parameter number followed by a decimal point and then the channel number followed by the **[SELECT]** key (i.e. 50.6 **[SELECT]** causes P50 for channel 6 to be displayed). The value of the parameter may be changed while a parameter is displayed by simply keying in a new value and pressing **[ENTER]**. **[CLEAR]** may be pressed to reset a value to zero. Of course the parameter's value may also be referenced or changed using the parameter number and instance with the %o macro command, i.e. 50.5P=x%o.

Connections

The first of the three channels used by 'QdDc3' must have the index signal connected to it. The next two channels should have the two encoder output signals connected. If the encoder count increments when you expect it to decrement, simply reverse the connections of the two encoder wires.

'QdDc3' Detailed Explanation

An encoder puts out two square wave signals which are 90° out of phase with each other. When the direction of travel is one way then one signal rises before the other. The opposite direction of travel results in the opposite signal rising first. When the encoder is not moving, the signals are stable.

Note that the width of the signals is relevant only to how fast the travel is occurring. The relationship between which signal rises first determines the direction of travel.

To achieve a proper direction determination, the lagging input should be connected to the lower numbered I/O channel.

The quadrature decode function counts every edge on the two encoder's signals. When two edges occur on one signal without any edges on the other signal then the direction of travel has changed.

The quadrature decode function is not capable of determining speed. If speed of travel is of interest, one of the two signals may be connected to another input channel and that channel may be programmed with the 'FqInA' or 'FqInB' function to measure the rate of travel. This method will not distinguish direction of travel. Travel in either direction will result in a positive result. If directional speed is of concern, then a macro may be used to capture the encoder value and a timer. Then some time later, the change in encoder value may be divided by the change in time to determine directional speed.

INDEX OPERATION

When the index pulse occurs, the value of the encoder, P50, is copied to the Index value, P51. P52 is calculated by the 660 as P50 - P51. It is calculated so that it has the same maximum and minimum values as P50, specifically +32,767 and -32,768. This capability can be used to determine a home or zero reference position for a positioning device.

See Figure 12-6 included with the 'QdDc2' description for clarification of the encoder signals.

PDIO SAMPLED FREQUENCY (FUNCTION 10)

The frequency debounce function is typically used to eliminate unwanted high frequency "noise" such as that caused by the contact closure of a mechanical switch. Since the PDIO port can detect very high frequencies, this "noise" can result in additional false pulse counts. The frequency debounce acts as a high frequency filter in which you can set the maximum expected frequency and ignore any pulses that exceed that frequency.

- **P852: "Fnam":** Allows a name to be assigned to I/O parameter 50 which is used to report the pulse frequency determined by this function.
- **P853: "Pnam":** Allows a name to be assigned to I/O parameter 51 which is used to report the pulse count determined by this function.
- **P854: "AvFN":** Allows a name to be assigned to I/O parameter 52 which is used to report the average pulse frequency determined by this function.
- **P856: "MaxF":** Used to specify the maximum expected frequency (48 to 10752 Hz).
- **P859: "Perd":** Used to specify the frequency measurement period in milliseconds.
- **P863: "Sfac":** Used to specify a variable to be used as a scaling (multiplication) factor for reporting received pulses. For example, a scaling factor of 0.1 would multiply the received pulses by 0.1 and thus report one count for every 10 pulses received.
- **P864: "# DP":** Used to specify the number of decimal places when reporting the received pulses.

There are three (3) weigh mode parameters used to report frequency debounce data:

- **P50.X** Frequency

- **P51.X** Pulse Count
- **P52.X** Avg. Freq.

where 'X' is a PDIO channel number.

P51 and P52 may be initialized to any value. However the Avg. count is only accurate upon power-up or after P52 is cleared to zero. This will not clear the total pulse count to zero though. The total pulse count will be equal to the number of pulses that have occurred within the current period.

Clearing P51 will reset the total pulse count to zero, however the average frequency will have an initial error.

Note: Clearing P52 will clear both frequency (P50) and pulse count (P51).

Maximum frequency is based on a 50% duty cycle. If using less than 50%, the maximum frequency specified should be increased.

Expected Max Frequency = 100 Hz/10% Duty Cycle

10% is 1/5th of the normal 50% duty cycle. The maximum frequency should therefore be increased to 5X the expected maximum frequency (500Hz).

```
____|-----|____|-----|____
 + + + + + + + + + +
```

50% Duty Cycle using expected Max Frequency: System Samples (+) must see two "Highs" in a row to see an actual Pulse rather than noise.

```
____|--|______|--|______:
 + + + + + + + + + +
```

10% Duty Cycle using above expected Max Frequency System Samples (+) never see two "Highs" in a row and thus considers the Pulses to be noise.

```
____|--|______|--|______
++++++++++++++++++
```

10% Duty Cycle using increased expected Max Frequency: System Samples (+) now see two "Highs" in a row and now sees a pulse.

Chapter 13 OPTIONS

This chapter contains installation information on the numerous hardware options that maximize the capabilities and functions of 60 series indicators and controllers.

OVERVIEW

OPTIONS

The following options are available for 60 Series Instruments:

660 Series Bus Options

24660B-001M0 DATABASE MODULE, 1M

24660B-002M0 DATABASE MODULE, 2M

24660B-256K0 DATABASE MODULE, 256K

60 Series Serial Bus Modules [SBM]

24660B-110A0 DC 4-POSITION OUTPUT

24660B-111A0 DC 4-POSITION INPUT

24660B-112A0 DC 2-POSITION IN / 2-POSITION OUT

24660B-121A0 LOW VAC 4-POSITION INPUT

24660B-122A0 LOW VAC 2-POSITION IN / 2-POSITION OUT

24660B-130A0 AC 4-POSITION OUTPUT

24660B-131A0 HIGH VAC 4-POSITION INPUT

24660B-132A0 HIGH VAC 2-POSITION IN / 2-POSITION OUT

24660B-200B0 MULTI-SCALE MODULE

24660B-203B0 ANALOG OUTPUT MODULE

24660B-202B0 REMOTE DISPLAY, 2” LCD

Other Options

24660B-404A0 20mA CURRENT LOOP OPTION

24663B-201B0 ALPHA KEYPAD KIT (MODEL 663)

24660B-402B0 BDM FLASH PROGRAM KIT

24660B-403B0 REFLASH PROGRAM KIT

24460B-204A0 TIME/DATE MODULE (460 SERIES ONLY)

24660B-401A0 NETWORK RS-485 OPTION

24660B-300A0 2-OPTION MOUNTING KIT

24663B-100C0 16 POSITION I/O BOARD (MODEL 663)

24665B-300A0 TERMINAL BLOCK KIT (MODEL 665)

24660B-102C0 DIN SCR MODULE

660 SERIES BUS OPTIONS

The 660 Series controllers contain components which could be damaged by Electrostatic Discharge (ESD) if serviced improperly. Use proper ESD precautions (wear a wrist strap connected to ground, use grounded work stations, etc.) when opening the enclosure.

High voltages may exist within the enclosure! To prevent the risk of electrical shock, ALWAYS unplug the indicator when opening the enclosure. Installation and servicing of the indicator should be performed by authorized and qualified service personnel only.

Never connect or disconnect option board cables while the indicator is powered. Doing so may result in circuit board damage.

The 660 Series Bus Options are mounted directly on the 660 Series main board using the hardware included in the kit. See Figure 13-1 or Figure 13-2 for database module installation in the 660 Series controllers.

To install 660 Series bus options:

1. Install three aluminum standoffs on the three short studs protruding through the main board using a 6mm socket or wrench. See Figure 13-1.
2. Tighten the standoffs hand tight then do a ¼ turn. Do not over torque.
3. Align the three mounting holes in the option board over the standoffs to ensure the shrouded male connector on the bottom of the board aligns with the mating connector J9 on the main board.
4. Gently press downward on the board to seat the connector.
5. Install the three 5.5mm hex nuts on the standoffs to secure the board.

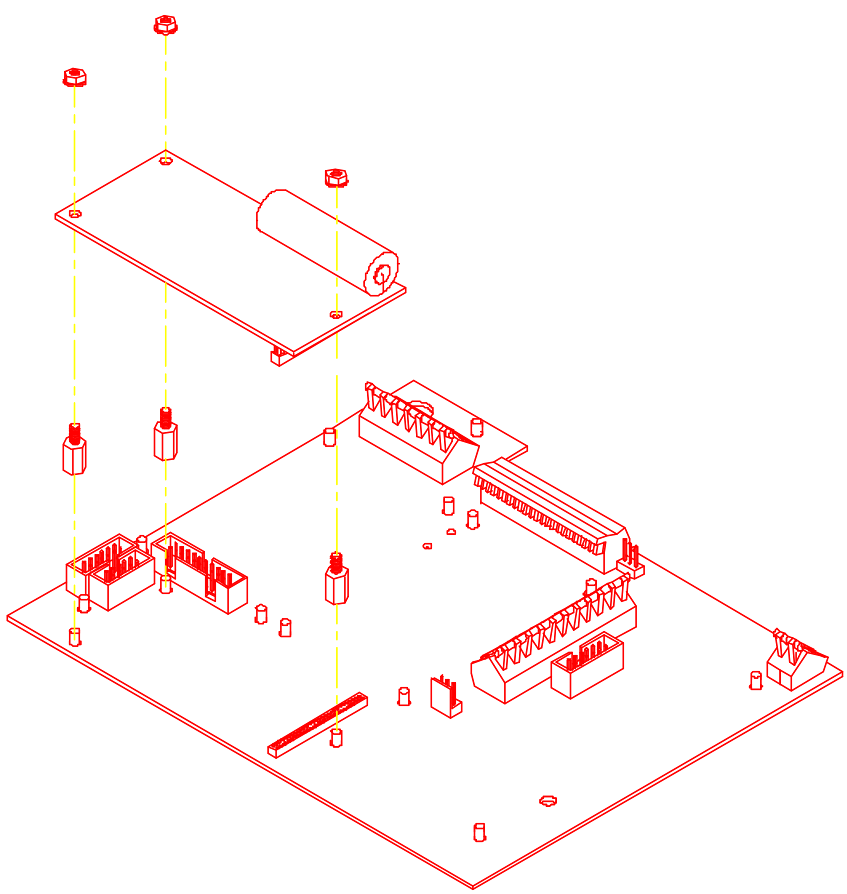


Figure 13-1: Database Module Installation

The 662, 665 LCD, and 663 LCD will have an LCD Interface Board installed upon the J9 main board connector. The database module is designed to mount directly on top of the LCD Interface Board. See Figure 13-2.

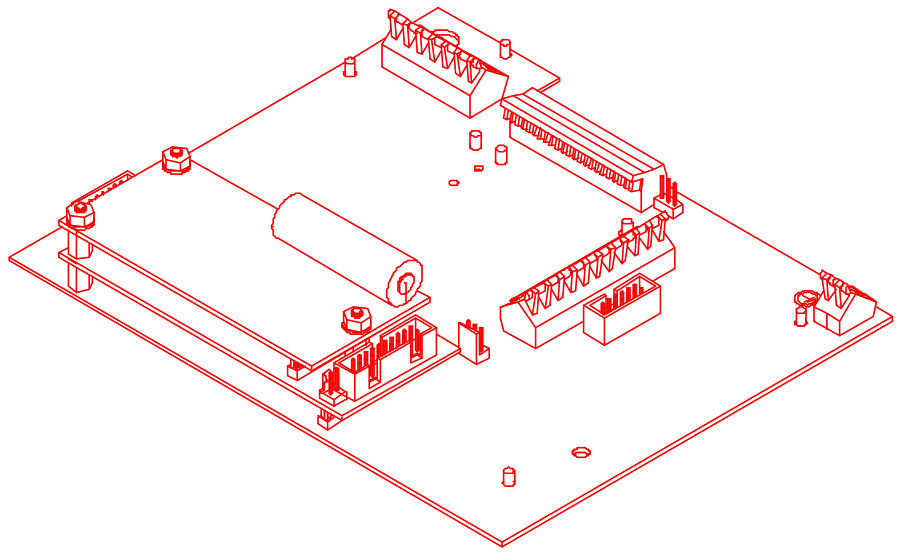


Figure 13-2: Database Module On Top of The LCD Interface

To install 660 Series bus options (LCD models):

1. Remove the three 5.5mm hex nuts from the LCD Interface Board Studs.
2. Install the 3 aluminum standoffs on the threads of the lower set of standoffs. Do not over tighten as the threads are aluminum and require very little torque.
3. Align the three mounting holes in the option board over the standoffs to ensure the shrouded male connector on the bottom of the board aligns with the mating connector J2 on the LCD Interface Board.
4. Gently press downward on the board to seat the connector.
5. Install the three M3 5.5mm hex nuts on the standoffs to secure the option board.

The database modules are meant to be installed on top of the bus option stack. Future options will be developed that make use of the J9 connector on the main board. Up to three options can be stacked upon that connector.

60 SERIES BUS MODULES (SBM)

The 60 Series Bus Modules are designed to be daisy-chained together. Setpoint boards must be at the end of the chain and there is a maximum limit of 128 total I/O points on the 660 Series controllers and 8 on the 460 Series indicators. The SBM daisy chain can include I/O options other than the 60 Series I/O modules including the 4, 8, and 16 position I/O boards previously used on the 650 Series controllers.

The mounting hardware and cable connections are the same for all of the 60 Series Serial Options. On the 660 Series, typically the first two options installed on the indicator will be mounted to the main board (only one module can be installed on a 460 Series main board).

MOUNTING INTERNAL 660 SERIES SBMS

The 60 Series controllers contains components which could be damaged by Electrostatic Discharge (ESD) if serviced improperly. Use proper ESD precautions (wear a wrist strap connected to ground, use grounded work stations, etc.) when opening the enclosure.

High voltages may exist within the enclosure! To prevent the risk of electrical shock, ALWAYS unplug the indicator when opening the enclosure. Installation and servicing of the indicator should be performed by authorized and qualified service personnel only.

Never connect or disconnect option board cables while the indicator is powered. Doing so may result in circuit board damage.

There are two sets of mounting studs available for the 60 Series SBMs on the 660 Series main board. The board can be installed on either set but it is recommended to install the first module on the left hand set of studs because then the board will not cover the J1 OPTION connector on the main board. It is also recommended to install high voltage modules in locations other than directly over the main board to avoid EMI/RFI concerns. These installation instructions are valid for all 60 Series instrumentation including standard and panel mount configurations.

To mount options on the 660 Series main board:

1. Install the 4 M3 Male-Female aluminum standoffs on the 4 short studs protruding through the main board using a 6mm socket or wrench. See Figure 13-3.
2. Tighten the standoffs hand tight then do a ¼ turn. Do not over torque.
3. Place the board over the threaded standoffs.
4. Install the four M3 5.5mm hex nuts on the standoffs to secure the module.
5. Connect the ribbon cable from the 'MAIN' connector on the module to the J1 OPTION connector on the main board.

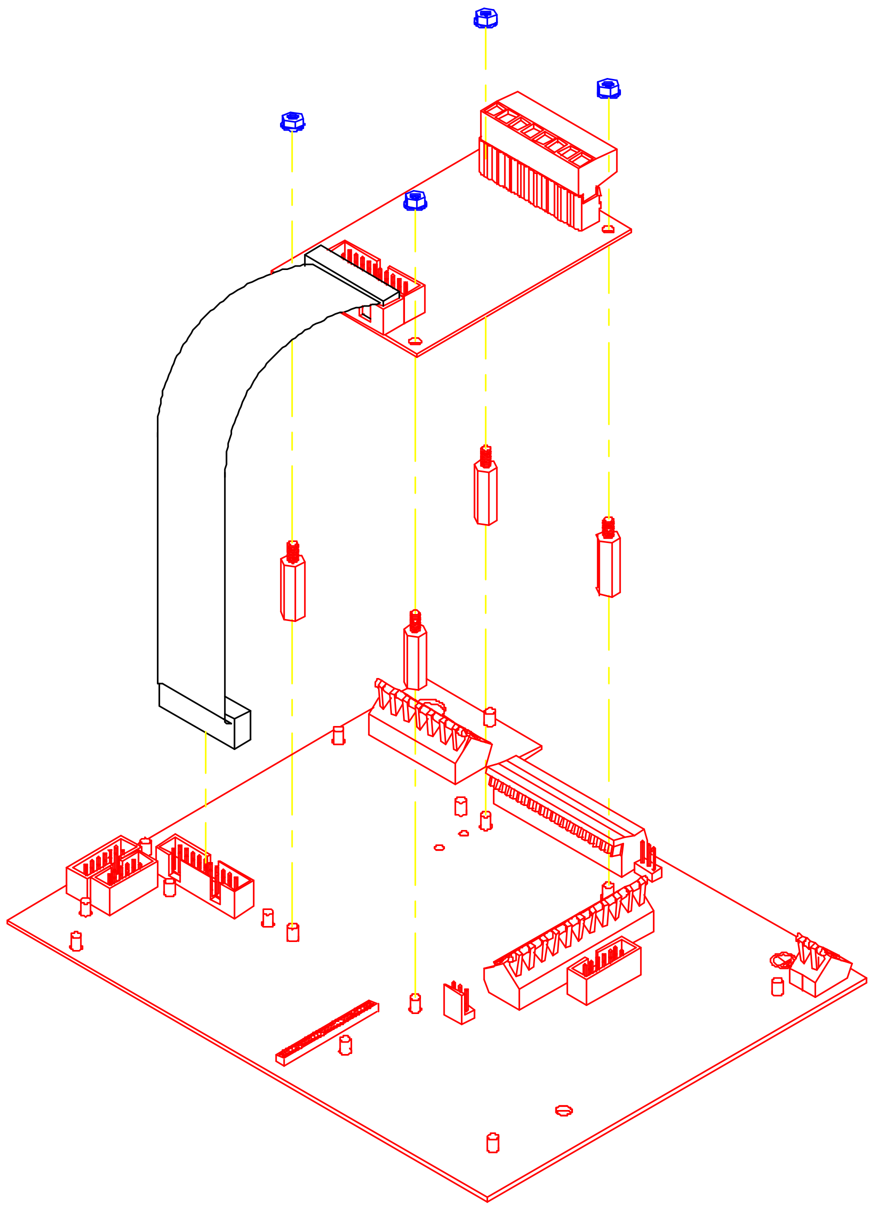


Figure 13-3: 660 Series Internal SBM Installation

The second module installs in the same manner with the ribbon cable connected between the first option board's 'NEXT' connector and the second module's 'MAIN' connector. Figure 13-4 shows a multi-scale module installed in the left position and a serial I/O module in the right position.

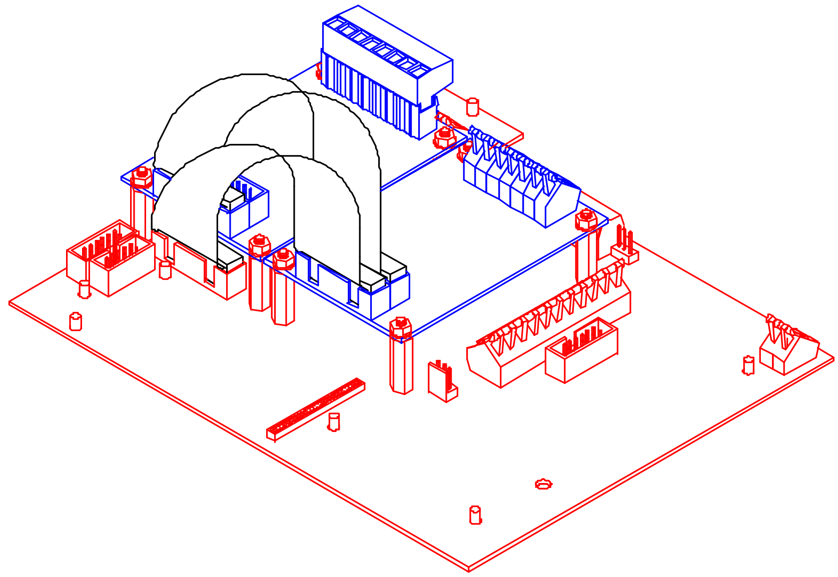


Figure 13-4: Serial I/O and Multi-Scale Module Installation

MOUNTING AN INTERNAL 460 SBM

The 60 Series contains components which could be damaged by Electrostatic Discharge (ESD) if serviced improperly. Use proper ESD precautions (wear a wrist strap connected to ground, use grounded work stations, etc.) when opening the enclosure.

High voltages may exist within the enclosure! To prevent the risk of electrical shock, ALWAYS unplug the indicator when opening the enclosure. Installation and servicing of the indicator should be performed by authorized and qualified service personnel only.

Never connect or disconnect option board cables while the indicator is powered. Doing so may result in circuit board damage.

There is one set of mounting studs available for the 60 Series SBM on the 460 Series main board. It is recommended to install high voltage modules in locations other than directly over the main board to avoid EMI/RFI concerns. These installation instructions are valid for all 60 Series instrumentation including standard and panel mount configurations.

To mount 460 Series SBMs on the main board:

1. Install the 4 M3 Male-Female aluminum standoffs on the 4 short studs protruding through the main board using a 6mm socket or wrench. See Figure 13-5.
2. Tighten the standoffs hand tight then do a ¼ turn. Do not over torque.
3. Place the board over the threaded standoffs.
4. Install the four M3 5.5mm hex nuts on the standoffs to secure the module.
5. Connect the ribbon cable from the ‘MAIN’ connector on the module to the J3 OPTION connector on the main board.

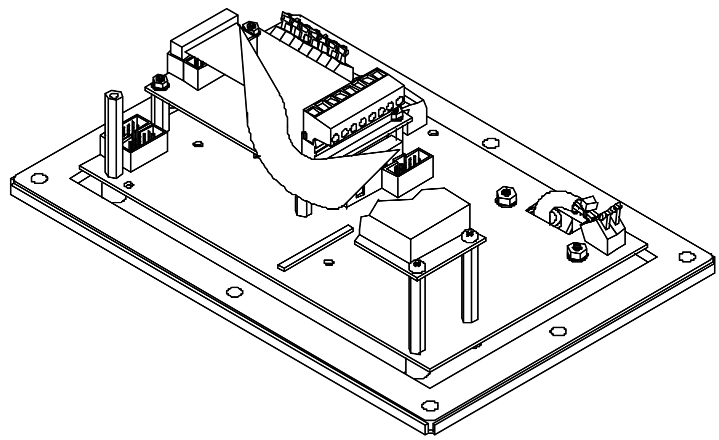


Figure 13-5: 460 Series SBM Installation

2-OPTION MOUNTING PLATE KIT

The 60 Series contains components which could be damaged by Electrostatic Discharge (ESD) if serviced improperly. Use proper ESD precautions (wear a wrist strap connected to ground, use grounded work stations, etc.) when opening the enclosure.

High voltages may exist within the enclosure! To prevent the risk of electrical shock, ALWAYS unplug the indicator when opening the enclosure. Installation and servicing of the indicator should be performed by authorized and qualified service personnel only.

Never connect or disconnect option board cables while the indicator is powered. Doing so may result in circuit board damage.

This kit is meant to provide an easy method to mount two of the 60 Series SBMs in a variety of applications. The two-option mounting plate simplifies installation to the RF cover on the 60 Series panel mounts see Figure 13-7. The plate is also required to mount 60 Series SBMs to the sub-panel of the 663 enclosure see Figure 13-8. The sub-panel is drilled and tapped to accept four plates for a total of 8 SBMs. Note that installing 60 Series SBMs to the 663 sub-panel will prevent the installation of the 16 position I/O board. It can also be used to mount two additional modules on the RF cover inside of the 663 enclosure.

MOUNTING SBMS TO THE RF COVER

The 2-Option mounting plate and ribbon cable #22-30-3061 are required to install SBMs on the RF Cover of the 460, 465, 660, 661, 662 or 665 panel mount Indicators as well as the 663. This type of installation is recommended for installing the third and fourth SBM in a daisy-chain or installing high voltage AC modules. Note that if any AC modules are installed on the RF cover of the panel mount they should be inside of an electrical enclosure during operation in order to avoid having exposed dangerous voltages.

To mount SBMs to the RF Cover:

1. Remove the RF cover M4 7mm hex head NTEP screw from the upper left corner if it's installed using a 7mm or 9/32" socket.
2. Remove the RF cover.
3. Install the 22-30-3061 ribbon cable into J1 OPTION connector on the 660 Series main board or J3 OPTION connector on the 460 Series main board. If no SBMs are installed on the main board then plug the

ribbon cable into the connector labeled ‘NEXT’ on the installed module.

4. Install the 2-Option mounting plate onto the RF cover using the M5 Philips screw and M5 hex nut.
5. Align the slots on the mounting plate with the two holes in the RF Cover.
6. Insert the screws though the holes so the heads of the screws are on the mounting plate and the screw threads protrude into the inside of the RF cover.
7. Tighten the nuts onto the screw threads to hold the mounting plate in place.
8. Install the 4 aluminum standoffs included in each 60 Series Option Kit onto the studs on the mounting plate. See Figure 13-6.
9. Tighten the standoffs hand tight. Do not over torque!
10. Place the module over the threaded standoffs.
11. Install the 4 5.5mm hex nuts on the standoffs to secure the module.
12. Connect the ribbon cable from the main board to J1 on the module.

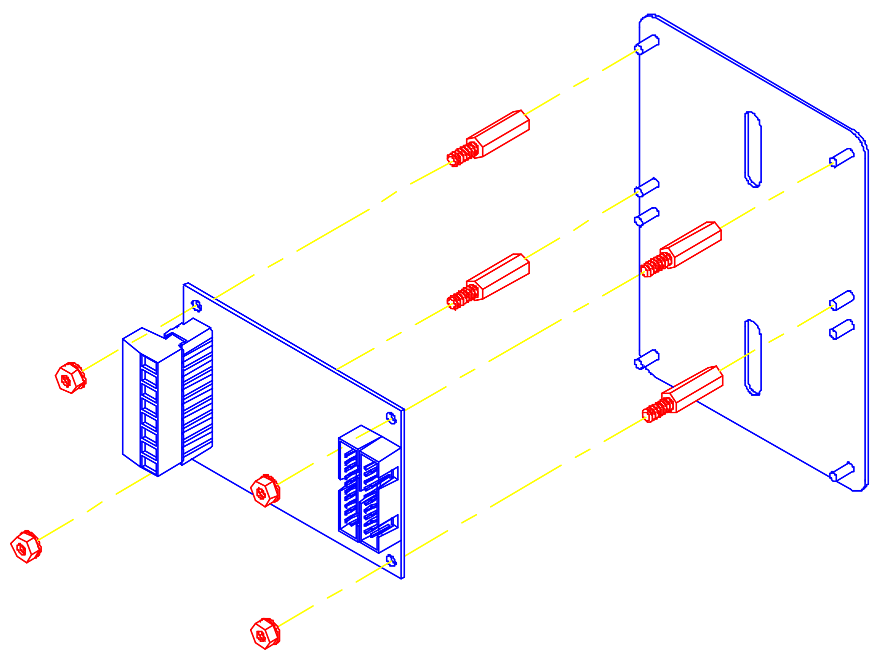


Figure 13-6: Installing an SBM on a Two-Option Mounting Plate

The second module installs in the same manner with the ribbon cable connected between the first module’s ‘NEXT’ connector and the second module’s ‘MAIN’ connector. Figure 13-7 shows two serial I/O modules installed on the RF cover.

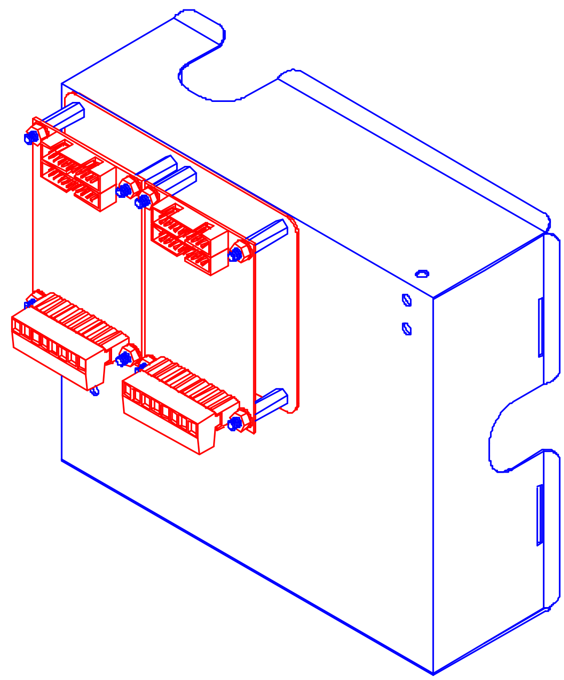


Figure 13-7: RF Cover Setpoint Installation

MOUNTING SBMS TO THE 663 SUB-PANEL

The sub-panel is drilled and tapped to accept four 2-Option mounting plates for a total of 8 SBMs. The plates must be attached to the sub-panel prior to installing the SBMs on the mounting plates. The plates are attached to the sub-panel using the M5 Philips screws included in each 2-Option Mounting Plate Kit. The nuts included in the kit are not used when installing the plates into an 663. See Figure 13-8 for the location and orientation of the option boards and plates. (the wiring duct is removed for clarity). Note that the boards are oriented so the interconnecting ribbon cables avoid the I/O wiring.

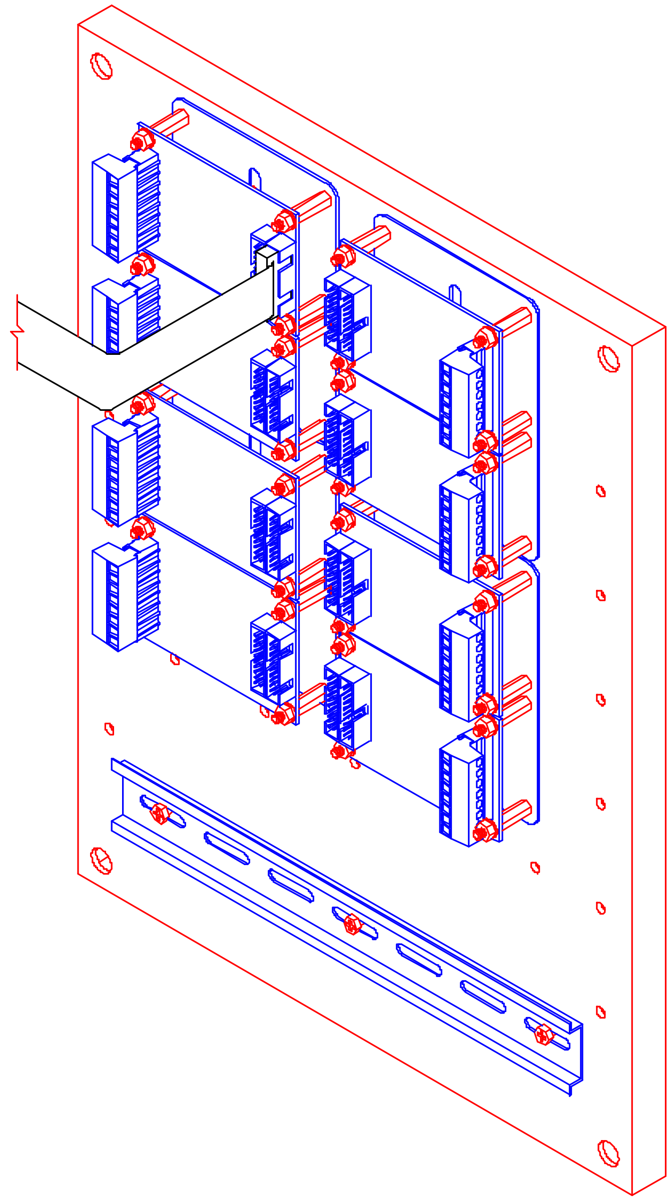


Figure 13-8: Option Board/Plate Location and Orientation

MOUNTING SBMS TO THE 665 OPTION MOUNTING BRACKET

The bracket is designed to mount two SBMs internal to the 665 enclosure. Extra hardware can be purchased to allow mounting of up to 6 SBMs on the bracket (some restrictions apply). This allows a total of 8 SBMs to be installed in the 665. Mount any AC serial I/O module on the bracket instead of above the 660 main board. This will reduce any EMI/RFI interference concerns.

To mount SBMs to the 655 option mounting bracket:

1. Install the 4 aluminum standoffs included in each 60 Series Option Kit onto the studs on the mounting plate. See Figure 13-9 (the top of bracket is removed for clarity).
2. Tighten the standoffs hand tight and then do a ¼ turn. Do not over torque.
3. Place the module over the threaded standoffs.
4. Install the 4 5.5mm hex nuts on the standoffs to secure the module.
5. If there are no other SBMs installed, connect the 22-30-35454 ribbon cable included with the 665 from the ‘MAIN’ connector on the SBM to the J1 OPTION connector on the main board.
6. If other SBMs are installed above the main board, connect the 22-30-35454 ribbon cable to the ‘NEXT’ connector on the last module in the daisy-chain.

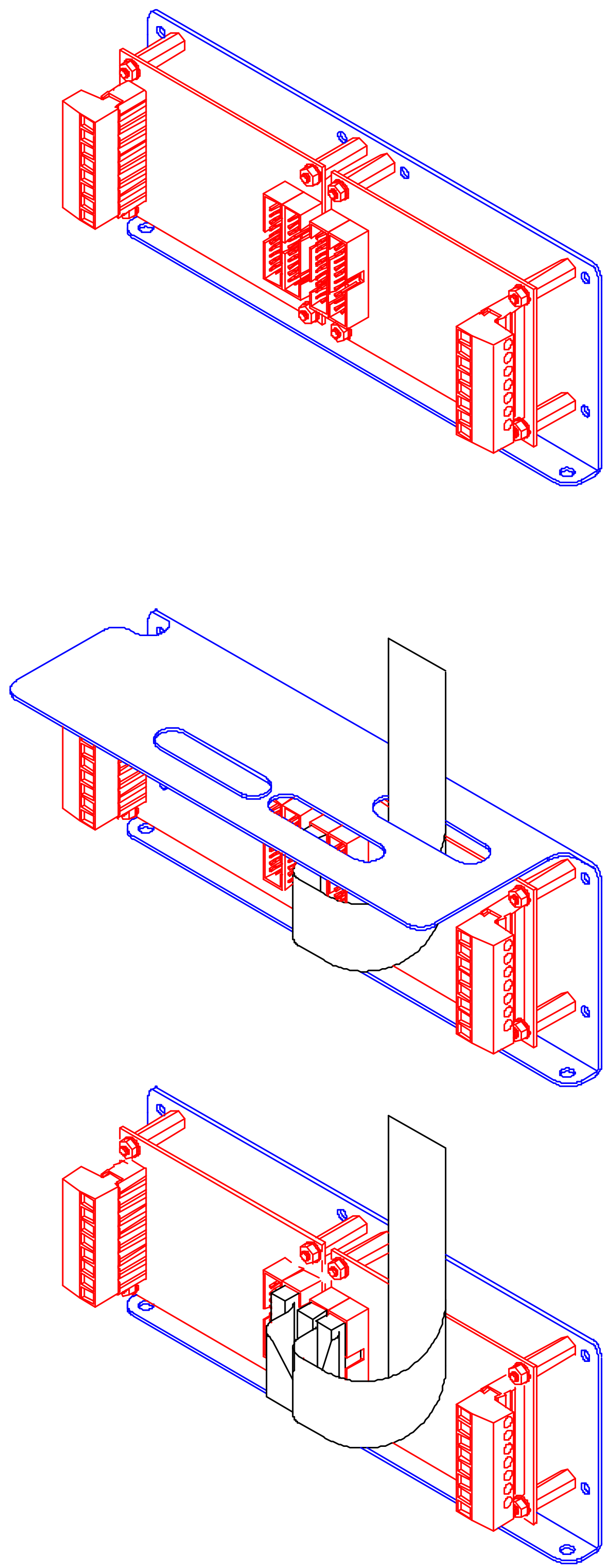


Figure 13-9: 665 Serial I/O Module Installation

Mounting More Than 2 SBMs on the 665 Option Mounting Bracket

You can install up to 6 serial bus modules on the 665 mounting bracket.

Installing Multi-scale or Analog Modules on the Backside of the Bracket

These mounting locations are only available for the multi-scale and analog output modules because of mechanical interference. You will need to purchase 4 pieces each of the following hardware per module:

- 17-20-3001 M3-0.5x8mm Male Female Aluminum Standoff
- 38-21-0101 M3-0.5x6mm,Pan Head, Phillips, Sems Screw

The four 19mm aluminum standoffs included in the option kit will not be used. The 22-30-35454 ribbon cable included with the 665 will be used in place of the 6” ribbon cable included in the option kit.

The 60 Series contains components which could be damaged by Electrostatic Discharge (ESD) if serviced improperly. Use proper ESD precautions (wear a wrist strap connected to ground, use grounded work stations, etc.) when opening the enclosure.

High voltages may exist within the enclosure! To prevent the risk of electrical shock, ALWAYS unplug the indicator when opening the enclosure. Installation and servicing of the indicator should be performed by authorized and qualified service personnel only.

Never connect or disconnect option board cables while the indicator is powered. Doing so may result in circuit board damage.

To install multi-scale or analog modules on the backside of the bracket:

1. Insert the M3 Phillips screw through the hole in the option mounting bracket so the screw threads are pointing toward the 660 series main board.
2. Place the 8mm standoff onto the screw threads and hold it while tightening the screw. Repeat this process to install the four standoffs necessary for each board.
3. Place the module over the threaded standoffs.
4. Install the four 5.5mm hex nuts on the standoffs to secure the module.
5. If there are no other SBMs installed, connect the 22-30-35454 ribbon cable included with the 665 from the ‘MAIN’ connector on the module to the J1 OPTIONS connector on the main board.
6. If other SBMs are installed above the main board then connect the 22-30-35454 ribbon cable to the ‘NEXT’ connector on the last module in the daisy-chain.

An additional 22-30-35454 cable will need to be purchased if SBMs are going to be installed on the studded side of the bracket. This longer cable is necessary to continue the daisy-chain connection to the modules installed on the studded side of the bracket.

Double Stacking SBMs

It is possible to install SBMs onto the studs on the option bracket and then place another set on top of them. You will need to purchase eight pieces of 17-20-3001 M3-0.5x8mm male/female aluminum standoff in order to stack two modules. The first two stacked modules should be installed on the right-hand portion of the bracket below the cable cutouts. Please note that if modules are going to be installed on the back side of the bracket as discussed in the previous section then they must be installed prior to installing modules on the studded side of the bracket.

To double stack SBMs:

1. Install four 8 mm standoffs onto the bracket studs.
2. Place the module over the threaded ends of the standoffs.
3. Install the 19mm standoffs included in the option kit onto the threaded standoff ends that protrude through the module board. This will hold the module in place.
4. Connect the ribbon cable to the connector on the module labeled ‘MAIN’ and move the 6” cable included in the option kit to the connector labeled ‘NEXT’.
5. Terminate your field wiring onto the module. This is important because installation of the stacked module will make terminating field wiring difficult.
6. Install the 8mm standoffs on top of the 19mm standoffs installed in Step 3.
7. Place the module over the threaded ends of the standoffs and fasten in place with the M3 5.5mm hex nuts included with the option kits.
8. Connect the ribbon cable from the previously installed module to the ‘MAIN’ connector on the next module.
9. Terminate you field wiring to the module.
10. Repeat this set of instructions for the installation of the second set of stacked modules.

⚠ **The 60 Series indicators contain components which could be damaged by Electrostatic Discharge (ESD) if serviced improperly. Use proper ESD precautions (wear a wrist strap connected to ground, use grounded work stations, etc.) when opening the enclosure.**

High voltages may exist within the enclosure! To prevent the risk of electrical shock, ALWAYS unplug the 60 when opening the enclosure. Installation and servicing of the indicator should be performed by authorized and qualified service personnel only.

Never connect or disconnect option board cables while the indicator is powered. Doing so may result in circuit board damage.

MOUNTING SBMS EXTERNALLY

SBMs can be mounted very easily in an external enclosure. If SBMs are being used in a wash down environment, be sure that the enclosure meets all wash down specifications. The enclosure will need to be at least 6”W x 8” L. The option mounting kit (part # 24660B-300A0) will be needed for installing modules. Also an interface cable (part # 22-30-32357) will be needed to connect the modules to the main board.

Installing Mounting Plate in the Enclosure:

1. Drill a 19mm hole in the side of the enclosure for the strain relief of the interface cable to fit through.
2. Line up the mounting plate in the enclosure.
3. Drill two 5mm hole for the mounting screws.
4. Screw down the mounting plate to the enclosure.

Installing SBMs in the External Enclosure:

1. Install the 4 aluminum standoffs included in each 60 Series Option Kit onto the studs on the mounting plate. See Figure 13-10.
2. Tighten the standoffs hand tight and then do a ¼ turn. Do not over torque.

3. Place the module over the threaded standoffs.
4. Install the 4 5.5mm hex nuts on the standoffs to secure the module.
5. Connect the module on the mounting plate to the another module in line or the main board via the interface cable. If a second module is installed on the mounting plate, connect the 22-30-35454 ribbon cable included between the two modules.

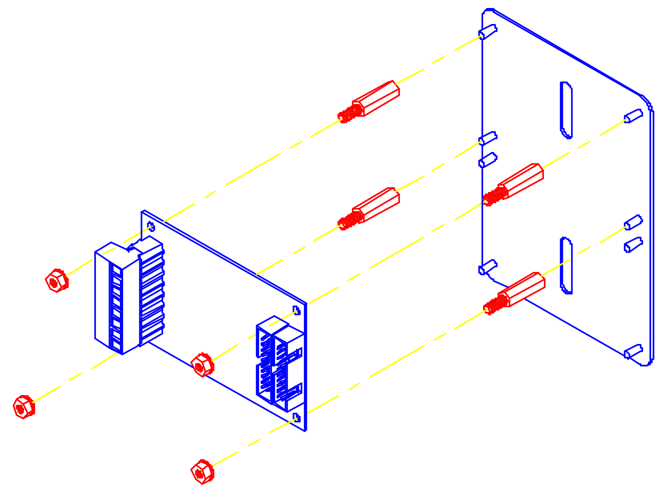


Figure 13-10: Two-Option Mounting Bracket

Serial I/O Modules

The 60 Series serial I/O modules were designed to provide the most flexibility for configuring the I/O for process control applications. Eight (8) serial I/O Kits were developed to provide a wide variety of I/O choices.

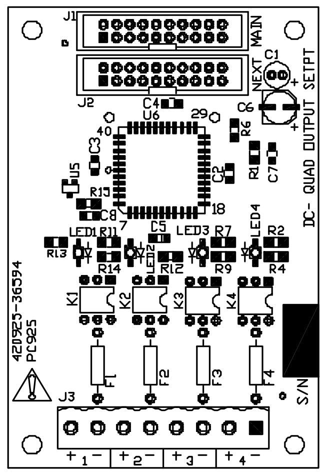
24660B-110A0	DC 4-POSITION OUTPUT
24660B-111A0	DC 4-POSITION INPUT
24660B-112A0	DC 2-POSITION IN / 2-POSITION OUT
24660B-121A0	LOW VAC 4-POSITION INPUT
24660B-122A0	LOW VAC 2-POSITION IN / 2-POSITION OUT
24660B-130A0	AC 4-POSITION OUTPUT
24660B-131A0	HIGH VAC 4-POSITION INPUT
24660B-132A0	HIGH VAC 2-POSITION IN / 2-POSITION OUT

The serial I/O modules mount in a variety of locations on the different models. Refer to the 60 Series Bus Modules section on page 13-5 for step by step installation instructions. The modules are meant to be connected in a serial daisy-chain configuration. They must be installed at the end of the daisy-chain behind any non-I/O modules. Although it is necessary to group the I/O modules at the end of the daisy-chain, within the group they can be in any order of inputs and outputs. The first 4 position board in the chain will be I/O #1-4, the second will be #5-8 and so on. 16 position I/O boards can be included within the same SBM daisy-chain.

Each board has LED's to indicate the status of the input or output. The LED's are located above each I/O position.

60 SERIES I/O MODULES

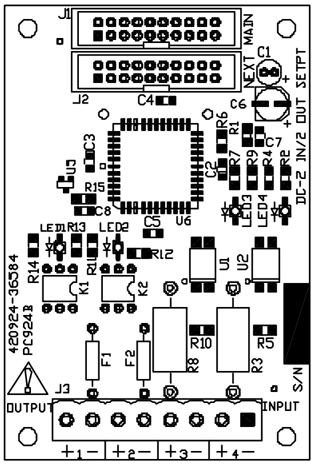
24660B-110A0 Setpoint Option,
DC, 4 Output, 60 Series



Position	Signal	Description
1	+	DC Input Power
	-	Switched DC Output
2	+	DC Input Power
	-	Switched DC Output
3	+	DC Input Power
	-	Switched DC Output
4	+	DC Input Power
	-	Switched DC Output

Maximum output voltage	60V dc
Minimum output voltage	3V dc
Maximum output current	2A dc
Maximum off-state voltage	60V dc
Maximum off-state leakage current	N/A
Maximum power dissipation	.7W

24660B-112A0 Setpoint Option DC, 2 Input 2 Output, 60 Series



Position	Signal	Description
1	+	DC Input Power
	-	Switched DC Output
2	+	DC Input Power
	-	Switched DC Output
3	+	DC Input Signal
	-	DC Return (Ground)
4	+	DC Input Signal
	-	DC Return (Ground)

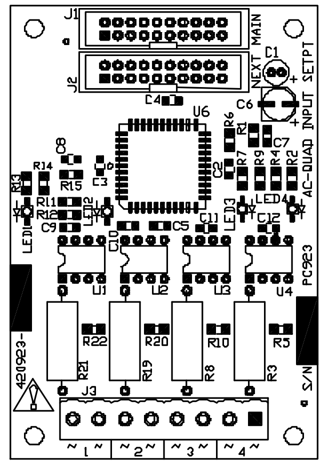
Outputs (Positions 1 & 2)

Maximum output voltage	60V dc
Minimum output voltage	3V dc
Maximum output current	2A dc
Maximum off-state voltage	60V dc
Maximum off-state leakage current	N/A
Maximum power dissipation	.7W

Inputs (Positions 3 & 4)

Maximum input voltage	30V dc
Minimum input voltage	3V dc
Maximum input current	47mA dc
Minimum input current	3mA dc
Input resistance	1.1K ohm
Maximum allowed off-state input current	1mA dc
Maximum allowed off-state input voltage	1.1V dc
Maximum power dissipation	1.6W

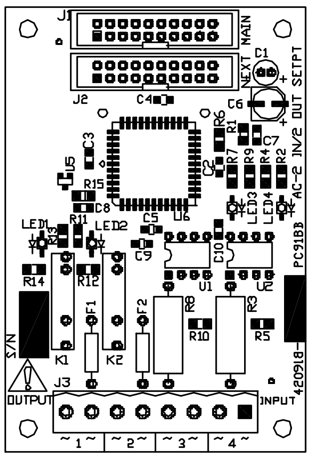
24660B-121A0 Setpoint Option Low Voltage AC,4 Input, 60 Series



Position	Signal	Description
1	~	AC Input Signal
	~	AC Return
2	~	AC Input Signal
	~	AC Return
3	~	AC Input Signal
	~	AC Return
4	~	AC Input Signal
	~	AC Return

Nominal input voltage	120V rms
Maximum input voltage	135V rms
Minimum input voltage	60V rms
Maximum input current	20mA rms
Minimum input current	8mA rms
Maximum allowed off-state input current	2mA rms
Maximum allowed off-state input voltage	15V rms
Maximum power dissipation	3W

24660B-122A0 Setpoint Option, Low Voltage AC, 2 Input 2 Output, 60 Series



Position	Signal	Description
	~	AC
	~	Switched AC
	~	AC
	~	Switched AC
	~	AC Input Signal
	~	AC Return
	~	AC Input Signal
	~	AC Return

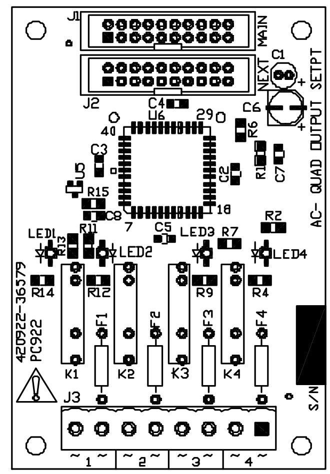
Outputs (Positions 1 & 2)

Maximum output voltage	240V rms
Minimum output voltage	20V rms
Maximum output current	1A rms
Maximum off-state voltage	400V peak
Maximum off-state leakage current	1mA rms
Maximum power dissipation	1.7W

Inputs (Positions 3 & 4)

Nominal input voltage	120V rms
Maximum input voltage	135V rms
Minimum input voltage	60V rms
Maximum input current	20mA rms
Minimum input current	8mA rms
Maximum allowed off-state input current	2mA rms
Maximum allowed off-state input voltage	15V rms
Maximum power dissipation	3W

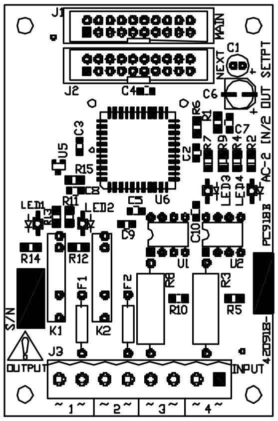
24660B-130A0 Setpoint Option 20-240VAC 4 Output, 60 Series



Position	Signal	Description
	~	AC
	~	Switched AC
	~	AC
	~	Switched AC
	~	AC
	~	Switched AC
4	~	AC
	~	Switched AC

Maximum output voltage	240V rms
Minimum output voltage	20V rms
Maximum output current	1A rms
Maximum off-state voltage	400V peak
Maximum off-state leakage current	1mA rms
Maximum power dissipation	1.7W

24660B-132A0 Setpoint Option Hi Voltage AC, 2 Input 2 Output, 60 Series



Position	Signal	Description
	~	AC Line
	~	Switched AC Load
	~	AC Line
	~	Switched AC Load
	~	AC Input Signal
	~	AC Return
	~	AC Input Signal
	~	AC Return

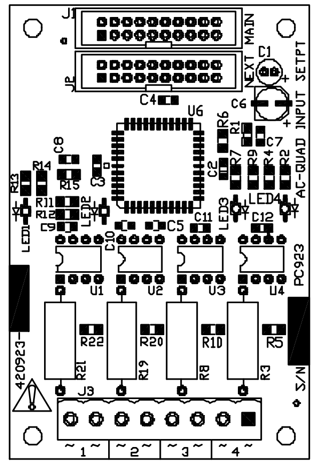
Outputs (Positions 1 & 2)

Maximum output voltage	240V rms
Minimum output voltage	20V rms
Maximum output current	1A rms
Maximum off-state voltage	400V peak
Maximum off-state leakage current	1mA rms
Maximum power dissipation	1.7W

Inputs (Positions 3 & 4)

Nominal input voltage	230V rms
Maximum input voltage	250V rms
Minimum input voltage	200V rms
Maximum input current	11.5mA rms
Minimum input current	9.1mA rms
Maximum allowed off-state input current	2mA rms
Maximum allowed off-state input voltage	45V rms
Maximum power dissipation	3W

24660B-131A0 Setpoint Option, Hi Voltage AC, 4 Input, 60 Series



Position	Signal	Description
	~	AC Input Signal
	~	AC Return
	~	AC Input Signal
	~	AC Return
	~	AC Input Signal
	~	AC Return
4	~	AC Input Signal
	~	AC Return

Nominal input voltage	230V rms
Maximum input voltage	250V rms
Minimum input voltage	200V rms
Maximum input current	11.5mA rms
Minimum input current	9.1mA rms
Maximum allowed off-state input current	2mA rms
Maximum allowed off-state input voltage	45V rms
Maximum power dissipation	3W

16 POSITION I/O SETPOINT OPTION

The 16 position I/O setpoint board is available in the following configurations:

- 420819-31396 Board only, no cables or hardware
- 24663B-100C0 Board, standoffs and hardware for mounting on the sub-panel of the 663 controller (The interconnecting cable is included with the 663).

The 16 position I/O board may also be connected to the 460 Series indicators. Only positions 1-8 are useable.

Note that other cables are available for connecting between the 60 Series Indicators and the 16 position I/O board. See the cable section of this manual for available sizes and configurations.

The 663 can accommodate both the new variety of SBMs and the 16 position I/O option board. Mounting a 16 position I/O option on the sub-panel of the 663 will prevent the installation of any 60 series modules on the sub-panel. Note however that 60 series modules can be installed on the 660 series main board and RF cover (as described in the previous sections) in addition to the 16 position I/O option board on the sub-panel.

I/O boards must be at the end of the chain and there is a maximum limit of 128 total physical setpoints for the 660 Series controllers and 8 physical setpoints for the 460 Series indicators. This serial option daisy chain can include I/O options other than the 60 Series I/O modules including the 4, 8 and 16 position I/O boards previously used on the 650 Series controllers. If the 16 position I/O board is used with a 460 Series indicator, only the first 8 I/O can be addressed.

Although the 650 Series 4 and 8 position I/O boards are compatible with 60 Series instruments, none of the 60 Series instruments provide a means of mounting them. The mounting footprint is not the same as the SBMs.

INSTALLING THE 24663B-100C0 16 POSITION I/O BOARD IN THE 663

1. Open the 663 enclosure.
2. Locate the five tapped holes on the sub-panel that are used for mounting the option board. Thread the five M4 6mm hex male female standoffs into the sub-panel. Make sure the sub-panel threads are clean and free of debris to avoid difficulty during installation. It may be helpful to run a M4x0.7 tap through the holes to clean them. Do not over tighten the standoffs because they will shear at high torque.
3. Align the 16 position I/O board over the standoffs and use the M4 screws to attach the board to the standoffs.

4. If no other serial options are installed connect the ribbon cable that leads from the main board J1 OPTION connector to the connector on the I/O board labeled CABLE TOWARD MAIN BOARD. This cable should already be installed on the main board and is stored in the wiring duct next to the I/O board location.
5. If other serial options are installed on the main board or RF cover then use this cable to connect to the last serial option in the daisy-chain.

INSTALLING THE 420819-31396 16 POSITION I/O BOARD

The mounting hole pattern for the board is shown in Figure 13-12. All five hole locations should be used to properly support the board. Make sure there is at least 5/16" of space between the back of the board and the sub-panel.

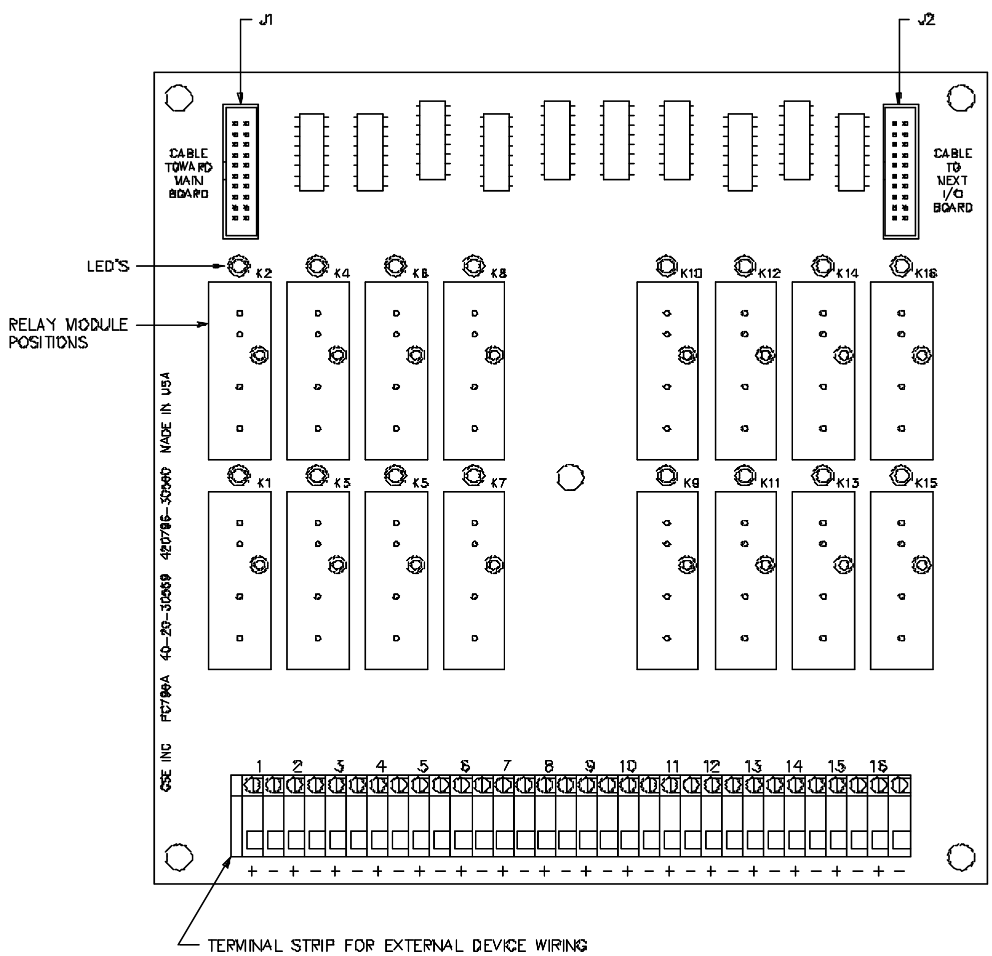


Figure 13-11: 16 Position I/O Board

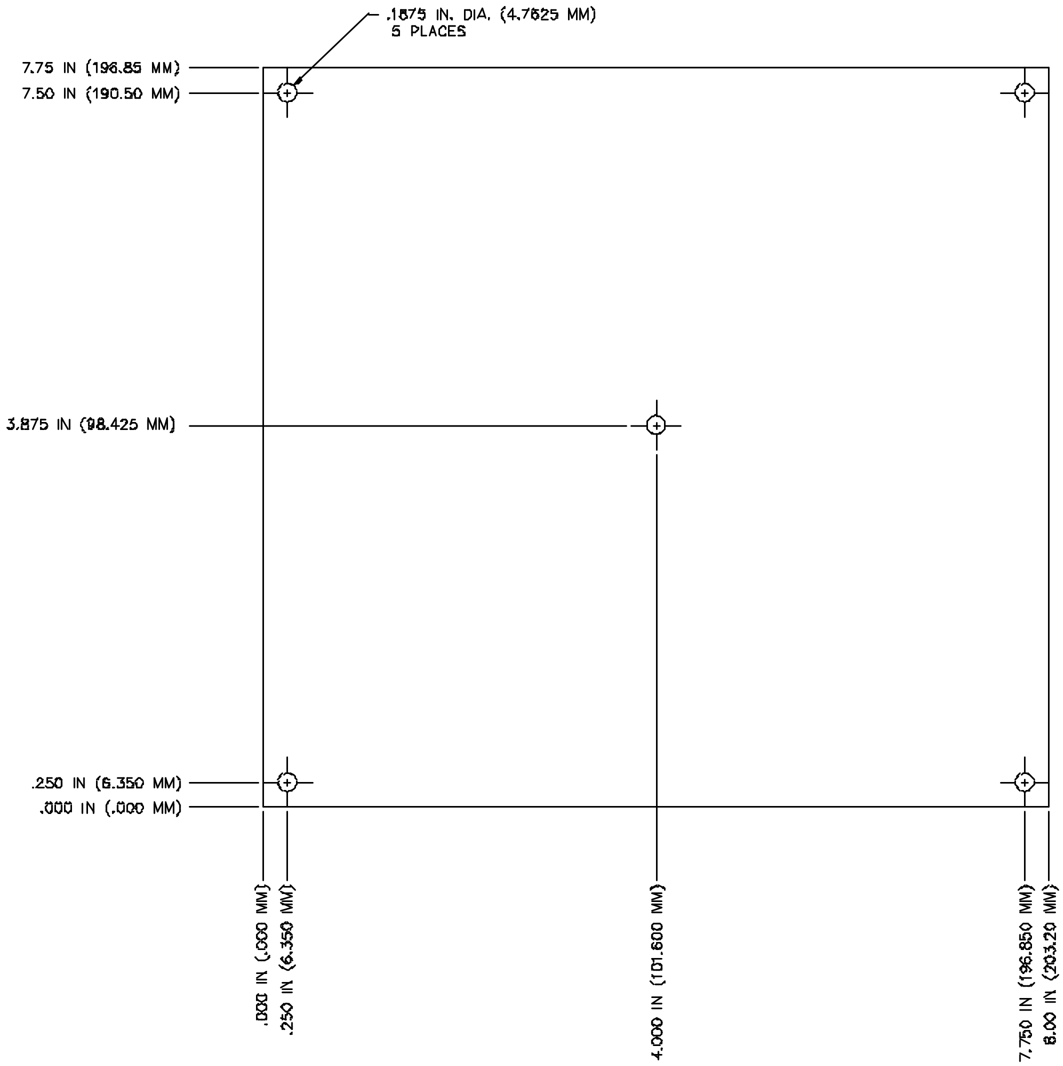


Figure 13-12: Mounting Holes for The 16 Position I/O Board

660 SERIES INTERFACE CONNECTIONS

The 16-Position I/O Board interfaces to the 60 Series main board with a 20-position, jacketed or non-jacketed ribbon cable, available in various lengths. The cable transmits the control signals for the relay modules.

MULTI-SCALE INSTALLATION

This section provides procedures for installing the multi-scale module.

The multi-scale input modules enable the 60 Series instruments to receive input signals from additional scales. Up to seven (7) multi-scale modules can be connected to a 660 Series controller for a total of 8 scale inputs; one (1) can be connected to a 460 Series indicator for a total of 2 scale inputs.

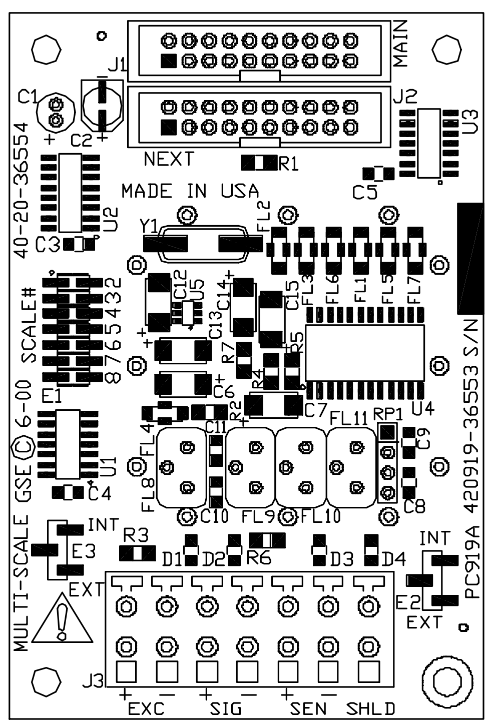
Two (2) multi-scale boards can be mounted *inside* the 660, 661, and 662 enclosures. Up to seven (7) can be installed within the 663 and 665 enclosures. One (1) can be installed in either the 460 and 465 enclosures. Review the methods of installation in the **60 Series Bus Modules (SBM)** section on page 13-5.

Interface cables of various lengths and an optional mounting plates are available for custom installations. These can be ordered separately. Refer to the table below for these accessories and other replacement parts.

Table 13-1: Multi-Scale Module Accessories and Replacement Parts

Description	GSE Part Number
6 inch interface ribbon cable	22-30-3060
18 inch interface ribbon cable	22-30-3061
22 inch option cable	22-30-28043
60 inch option cable	22-30-28048
Two-option mounting plate	24660B-300A0

A single jumper on the Multi-Scale PC board **(E1)** selects whether it will be addressed as Scale 2, 3, 4, 5, 6, 7, or 8.



The Scale 1 position is not available because the scale input on the *main board* is addressed as Scale 1.

INSTALLATION NOTES

- Make sure **E1** is configured for the correct scale number selection. If this is the first multi-scale option card installed (control of scale 2), then the jumper should be in the 2 position. (GSE default is scale 2).
- Connect the load cell wiring to **J3** of the multi-scale option board.
- **If a 6 lead load cell is used**, jumpers **E2** and **E3** on the multi-scale module should both be moved to the external (EXT) position.
- **If a 4 lead load cell is used**, jumpers **E2** and **E3** on the multi-scale module should both be moved to the internal (INT) position.
- If another option is in use, such as another multi-scale or analog module, connect it to **J2 (NEXT)** on the previous module.
- If any I/O modules are to be installed, they must be daisy chained *after* all multi-scale and analog output modules.
- Route any cables through the rear panel strain reliefs making sure to connect any shield wires to a grounding stud.
- Re-assemble the controller and install the rear panel.

MULTI-SCALE SETUP PARAMETERS

Multi-scale setup parameters beginning at P108 → P142 are used to setup the multi-scale module. Each scale is selected as an instance, 1 → 8 on the 660 Series controllers and 1 → 2 on the 460 Series indicators. Refer to page 3-26 for complete configuration details.

A/D CALIBRATION

A printout of the A/D calibration information accompanies each multi-scale module on a sheet inside the static bag with the circuit board. These parameters must be entered into the indicator in order for it to read the scale input accurately. A sample printout appears below:

```
100%s23640%i%e
61099%s2%e                P61099.  Scale 2
61101%s  1.000000%e       P61101.   CAL Factr   1.000000
61102%s  0.000000%e       P61102.  ReZro Wght   0.000000
61103%s  0.000000%e       P61103.  ZrTrk Wght   0.000000
61104%s8%e                P61104.  CZero    0%%
61105%s     60573%e       P61105.  Fine Zero       60573
61106%s2%e                P61106.  CGain   100
61107%s  1.250029%e       P61107.  Fine Gain    1.250029
61110%s     64409%e       P61110.  Zero Adj25     64409
61111%s    -26859%e       P61111.  Zero Adj50    -26859
61112%s   -198972%e       P61112.  Zero Ad100   -198972
61113%s  0.942661%e       P61113.  Gain Adj1   0.942661
61114%s  0.945947%e       P61114.  Gain Adj2   0.945947
61115%s  0.955080%e       P61115.  Gain Adj4   0.955080
61116%s  0.956094%e       P61116.  Gain Adj8   0.956094
61117%s       808%e       P61117.  AIN  NROff       808
61118%s     -1519%e       P61118.  AIN  NROff     -1519
61119%s     -1995%e       P61119.  AIN  NROff     -1995
61120%s     -4857%e       P61120.  AIN  NROff     -4857
```

```
61121%s     -1070%e      P61121.  VREF NROff      -1070
61122%s 111111%e         P61122.  SN:  111111
64102%s                  View errors after uploading!
Operator: rg
Date:mmddyy=090610
420919-36553 Multi-scale Option Board
ATR-31565
```

The values indicated in bold define the A/D calibration values for the multi-scale module. They must be keyed into and stored in their respective parameters for accurate operation. The serial number (P61122) is intended to allow you to associate these setup parameters to a specific multi-scale option module. Contact GSE with the part number and serial number of the multi-scale board if the calibration sheet is misplaced.

Refer to chapter 4 on page 4-11 for complete details on A/D calibration.

SCALE CALIBRATION

Load cell calibration must performed after the multi-scale module is installed. Refer to chapter 4 for complete calibration instructions.

WEIGH MODE OPERATION

From the weigh mode, pressing **[SCALE SELECT]** will toggle the display to the next enabled scale. Regardless of which scale is currently selected for viewing, all enabled multi-scale modules will continue to process weight data. Thus, any setpoints, analog outputs, custom transmits, etc. that are based on an active weight parameter will be continuously updated independent of any other process (i.e. other multi-scale modules, macros, transmits, etc.).

CONNECTING MULTIPLE SBMS

It possible to install many SBMs inside the standard enclosures. Note that the standard SBM kits supply a standard interface ribbon cable. If multiple output devices are daisy-chained, it is possible that the cable length supplied will reach the next device on the link. This is due in some instances to improper positioning of the option devices. If extensive output devices are to be used take care to install them in a manner that the cables supplied will be adequate. Longer interface cables can be purchased if necessary.

ANALOG OUTPUT INSTALLATION

This section provides procedures for installing the analog output module.

The analog output modules enable the 60 Series instruments to generate a 0-10VDC, 0-20mA or 4-20mA output signal corresponding to the value of most operating parameters.

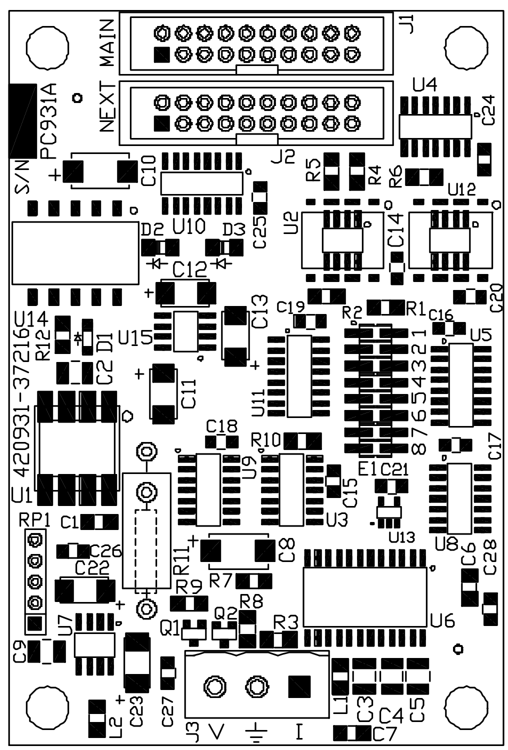
Two (2) analog output boards can be mounted *inside* the standard 660, 661, and 662 enclosures. Up to eight (8) can be installed within the 663 and 665 enclosures. One (1) module can be installed *inside* of the 460 and 465 enclosures. Review the methods of installation in the **60 Series Bus Modules (SBM)** section on page 13-5.

Interface cables of various lengths and an optional mounting plate are available for custom installations. These can be ordered separately. Refer to the table below for these accessories and other replacement parts.

Table 13-2: Analog Output Module Accessories and Replacement Parts

Description	**GSE Part Number**
6 inch option interface ribbon cable	22-30-3060
18 inch option interface ribbon cable	22-30-3061
22 inch option cable	22-30-28043
60 inch option cable	22-30-28048
Two-option mounting plate	24660B-300A0

A single jumper on the analog output PC board **(E1)** selects whether it will be addressed as analog output 1, 2, 3, 4, 5, 6, 7, or 8.



INSTALLATION NOTES

- Make sure **E1** is configured for the correct analog output number selection.
- If another module is in use, such as another analog or multi-scale module, connect it to **J2 (NEXT)** on the previous module.
- If any I/O modules are to be installed, they must be daisy chained *after* all multi-scale and analog output modules.
- Route any cables through the rear panel strain reliefs making sure to connect any shield wires to a ground stud.
- Re-assemble the controller and install the rear panel.

ANALOG OUTPUT PARAMETER SETUP

Analog output setup parameters beginning at P170 → P177 are used to setup the output module. Each output is selected as an instance, 1 → 8 on the 660 Series controllers and 1 → 2 on the 460 Series indicators. Refer to page 3-30 for complete configuration details.

ANALOG OUTPUT CALIBRATION

A printout of the A/D calibration information accompanies each analog output module on a sheet inside the static bag with the circuit board. These parameters must be entered into the indicator in order for it to operate accurately. A sample printout appears below:

```
60100%s%e P60100. 1995*GSE*
60101%s%e P60101. 0660y01030
60102%s%e P60102. Sep262000

60200%s%e P60200. B SN:05619
60201%s%e P60201. AuditTrail OIML 00017
60202%s%e P60202. I SN:00000
60203%s%e P60203. AuditTrail Cal. 00003
60204%s%e P60204. AuditTrail Setup 00034
60205%s%e P60205. MUST!CHECK

61200%s1%e P61200. AnOut 1
61201%s2432%e P61201. V Z 2432
61202%s51300%e P61202. V G 51300
61203%s0%c P61203. 0mAZ None!
61204%s54633%e P61204. 0mAG 54633
61205%s10920%e P61205. 4mAZ 10920
61206%s54633%e P61206. 4mAG 54633
61207%s%e P61207. SN :01022

64102%s View errors after uploading!
```

The values indicated in bold define the calibration values for the analog output option. They must be keyed into and stored in their respective parameters for accurate operation. The serial number (P61207) is intended to allow you to associate these setup parameters to a specific analog output option board. Contact GSE with the part number and serial number of the analog output option board if the calibration sheet is misplaced.

Refer to page 14-8 for complete details on analog output calibration.

PRINTING CALIBRATION VALUES

Once installed, the analog output calibration values can be downloaded to a printer or PC for backup.

To print A/D calibration values:

1. From the weigh mode, key in **60100 [SELECT]** to access the GSE copyright parameter.
2. Select analog output calibration values:

   Key in **23900 [PRINT]** to send calibration values for all analog outputs.*

   Key in **23901 [PRINT]** to send calibration values for analog output #1.*

   Key in **23902 [PRINT]** to send calibration values for analog output #2.*

   Key in **23903 [PRINT]** to send calibration values for analog output #3.

   Key in **23904 [PRINT]** to send calibration values for analog output #4.

   Key in **23905 [PRINT]** to send calibration values for analog output #5.

   Key in **23906 [PRINT]** to send calibration values for analog output #6.

   Key in **23907 [PRINT]** to send calibration values for analog output #7.

   Key in **23908 [PRINT]** to send calibration values for analog output #8.
3. The display prompts Enter Comm#. Key in the communication port number (1 → 4).*
4. Analog output calibration values are transmitted.

* *When using the 460, scroll in the number and press **[ENTER]** instead of **[PRINT]** in step #2. It is also necessary to press **[ENTER]** after scrolling in the comm# in step #3.*

CONNECTING MULTIPLE SBMS

It possible to install many SBMs inside the standard enclosures. Note that the standard SBM kits supply a standard interface ribbon cable. If multiple output devices are daisy-chained, it is possible that the cable length supplied will reach the next device on the link. This is due in some instances to improper positioning of the option devices. If extensive output devices are to be used take care to install them in a manner that the cables supplied will be adequate. Longer interface cables can be purchased if necessary.

REMOTE DISPLAY MODULE INSTALLATION

This section provides procedures for installing the remote display module. Review the methods of installation in the **60 Series Bus Modules (SBM)** section on page 13-5.

The remote display module will connect via cable (not provided with the kit) to the Model 1500 2” LCD remote display. The Model 1500 remote display will interface to any 60 Series instrument using the remote display module.

Interface cables of various lengths and an optional mounting plate are available for custom installations. These can be ordered separately. Refer to the table below for these accessories and other replacement parts.

Table 13-3: Remote Display Module Accessories and Replacement Parts

Description	**GSE Part Number**
Remote Display PCB	420905-35649
Standoff (4)	17-20-3019
Lock nut (4)	38-21-1640
6 inch option interface ribbon cable	22-30-3060
18 inch option interface ribbon cable	22-30-3061
22 inch option cable	22-30-28043
60 inch option cable	22-30-28048
Two-option mounting plate	24660B-300A0

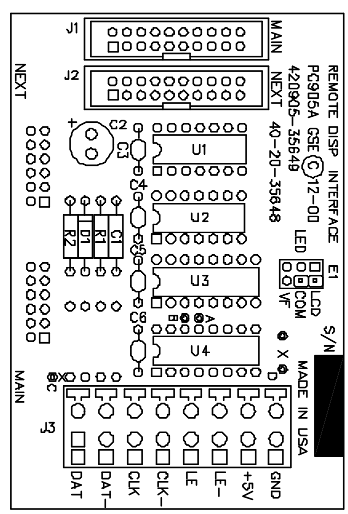


Figure 13-13: Remote Display Module

Figure 13-14 shows the field wiring connector from the Model 1500 remote display. One end of the cable will be wired into this connector and the other end will be wired into the connector on the remote display module. It is recommended that a 4-twisted pair conductor Teflon cable (GSE P\N 22-10-4660), or a 4-twisted pair PVC cable (GSE P\N 22-10-4665) is used. Also make sure the interface cable is installed through the Model 1500 strain relief before installing the connector. Refer to Table 13-4 for the connector's pin descriptions.

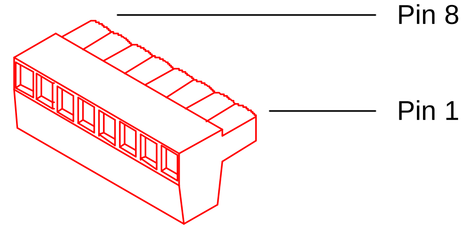


Figure 13-14: Remote Display Connector

Table 13-4: Remote Display Pin Out

PIN	**NAME**	**DESCRIPTION**
1	DAT	Positive data line
2	DAT-	Negative data line
3	CLK	Positive clock line
4	CLK-	Negative clock line
5	DCS	+ Data clear send line
6	DCS-	- Data clear send line
7	+5v	+ 5 volt line
8	GROUND	Common ground

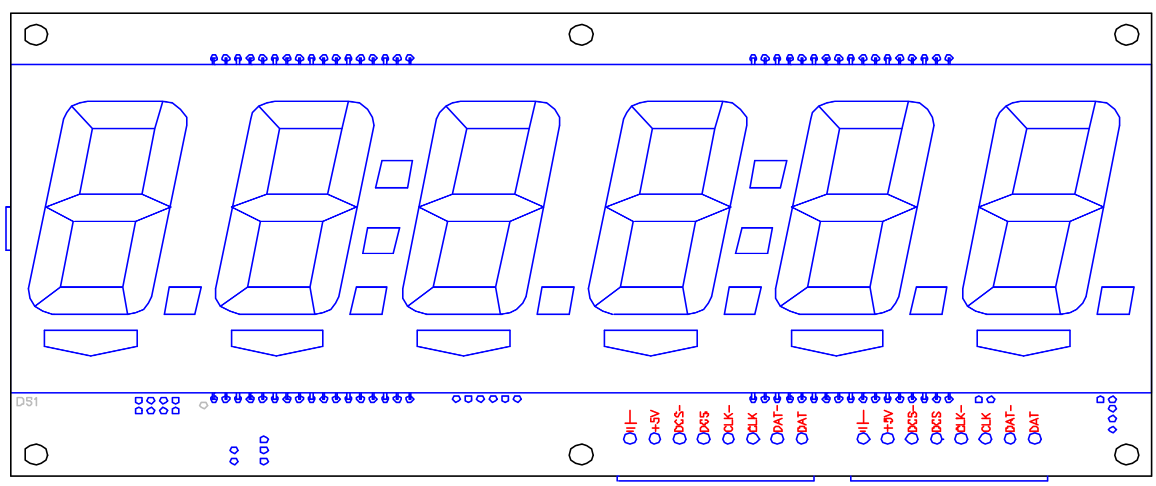


Figure 13-15: M1500 Remote Display Board

INSTALLATION NOTES

- Make sure **E1** is configured for LED when connecting to the Model 1500 remote display.

- Using a 4-twisted pair PVC cable(not supplied with the kit), connect the remote display module to the Model 1500 connector.
- If another option is in use, such as another multi-scale or analog module, connect it to **J2 (NEXT)** on the previous module.
- If any I/O modules are to be installed, they must be daisy chained *after* all multi-scale, analog output or remote display modules.
- Route any cables through the rear panel strain reliefs making sure to connect any shield wires to a grounding stud.
- Re-assemble the controller and install the rear panel.

REMOTE DISPLAY PARAMETER SETUP

Remote display setup parameters beginning at P293 → P294 are used to setup the remote display. Refer to page 3-35 for complete configuration details.

CONNECTING MULTIPLE SBMS

It possible to install many SBMs inside the standard enclosures. Note that the standard SBM kits supply a standard interface ribbon cable. If multiple output devices are daisy-chained, it is possible that the cable length supplied will reach the next device on the link. This is due in some instances to improper positioning of the option devices. If extensive output devices are to be used take care to install them in a manner that the cables supplied will be adequate. Longer interface cables can be purchased if necessary.

RS-485 MODULE INSTALLATION

This section describes the installation of the RS-485 module. Installing this module will convert comm port 1 from RS-232 to RS-485.

To install the RS-485 module (all models):

1. Open the indicator and gain access to the main board.
2. Remove the U17 IC (on the 660 Series main board) or U3 IC (on the 460 Series main board) from its socket.
3. Remove the white wire jumper from the IC socket where the chip in step 2 was removed.
4. Snap the plastic spacers into the three mounting holes on the main board.
5. Gently press the option board into the socket.
6. Apply the included sticker to comm port 1. The label will go over the silk screen on the board. Use the portion of the label titled “60 Series.”

Align RB+ over CTS on comm port 1 and adhere the label over the silkscreen.

7. Reassemble the indicator.

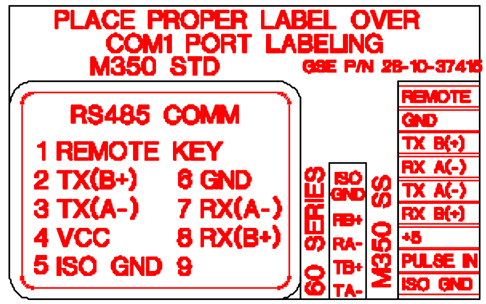


Figure 13-16: Label for RS-485 COMM PORT pin designations

NETWORK PARAMETER SETUP

The RS-485 module does not have to be enabled in the setup mode. The module simply converts the standard RS-232 communication on comm port 1 to RS-485. However the advantage of using the RS-485 module, aside from the ability to transmit over long distances, is the ability to network multiple indicators using the same communication wires. When networking indicators, it is necessary to set up a network address for each indicator. The network module itself does not require addressing, rather each indicator must be enabled for network addressing in the setup mode. Setup parameters beginning at P250 → P251 (or P205 and P209 → P211 for Modbus) are used to configure the network. Refer to page 3-34 for complete configuration details.

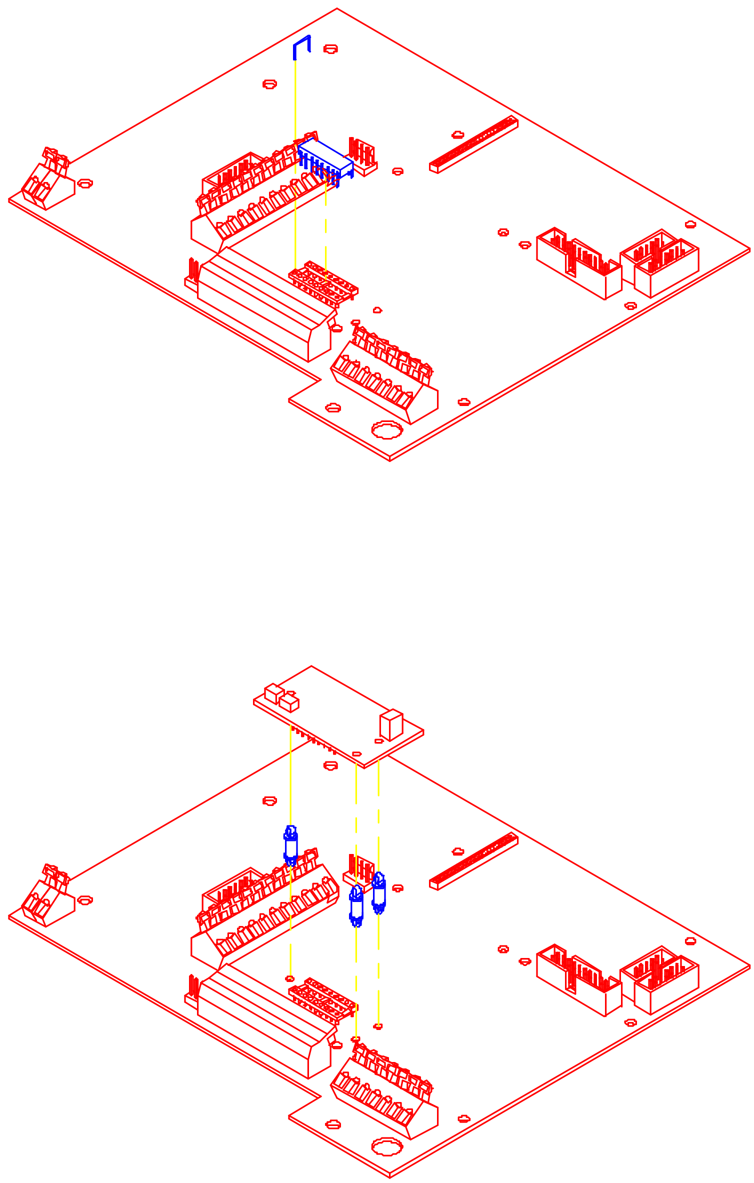


Figure 13-17: RS-485 Installation

HALF DUPLEX (2-WIRE)

Installing jumpers 1, 2 and 4 on the RS-485 option board electrically connects pin RX B(+) to pin TX B(+), and pin RX A(-) to pin TX A(-) on the option board. This effectively provides two + and two - pin connections, enabling easy connection of network lines in parallel from device to device without having to position two wires into the same lever socket. A B(+) line from each device on the network should be connected in parallel to the next device on the network. This is also true for all A(-) lines.

The units inside the two end-points of the network loop will utilize both A(-) pin connections and both B(+) pin connections. The units at the end-points of the network will utilize only one A(-) pin connection and one B (+) pin connection.

FULL DUPLEX (4-WIRE)

Removing jumpers 1, 2 and 4 on the RS-485 option board requires that the transmit and receive lines be wired independently of one another. The RX B(+) and RX A(-) receive lines must be wired in parallel to the next device's RX B(+) and RX A(-) receive lines , and the TX B(+) and TX A(-) transmit lines must be wired in parallel to the next device's TX B(+) and TX A(-) transmit lines.

In order to connect network lines in parallel from device to device it is necessary to position two wires into the same lever socket. This requires that the wire used to build the network be 24AWG or smaller to allow both wires to fit into the same lever socket.

BOTH HALF DUPLEX AND FULL DUPLEX

The network boards on *both* end-points should install jumper 3 on the RS-485 option board to engage the 120 Ω termination resistor (R8). The boards between the two end-points should remove jumper 3 on the RS-485 option board.

The isolated ground (ISO GND) should be connected in parallel from unit to unit. A shielded twisted two pair cable is recommended throughout the network.

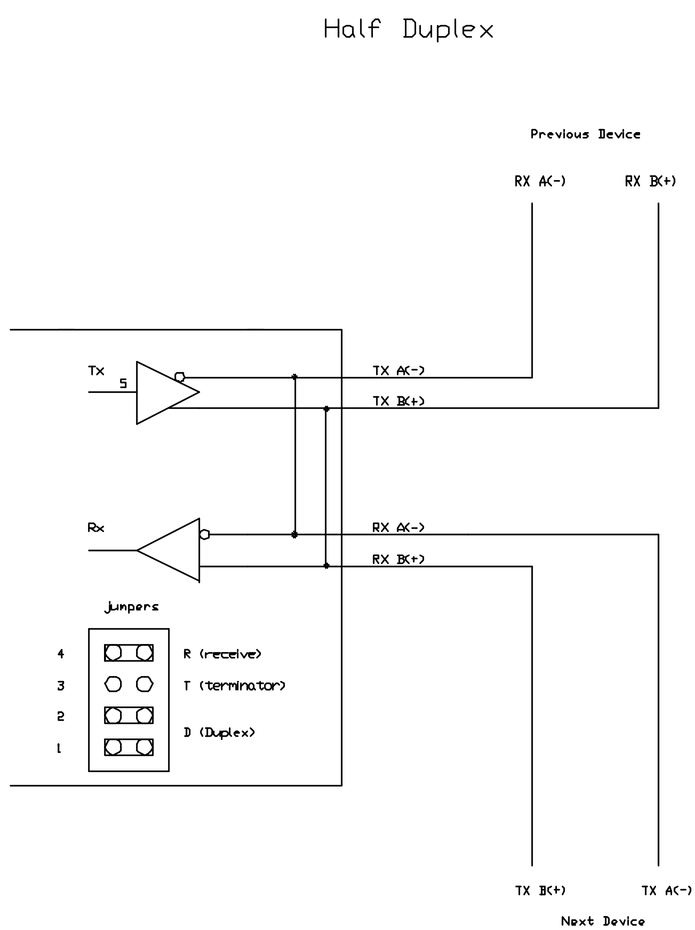


Figure 13-18: Half Duplex Wiring Schematic

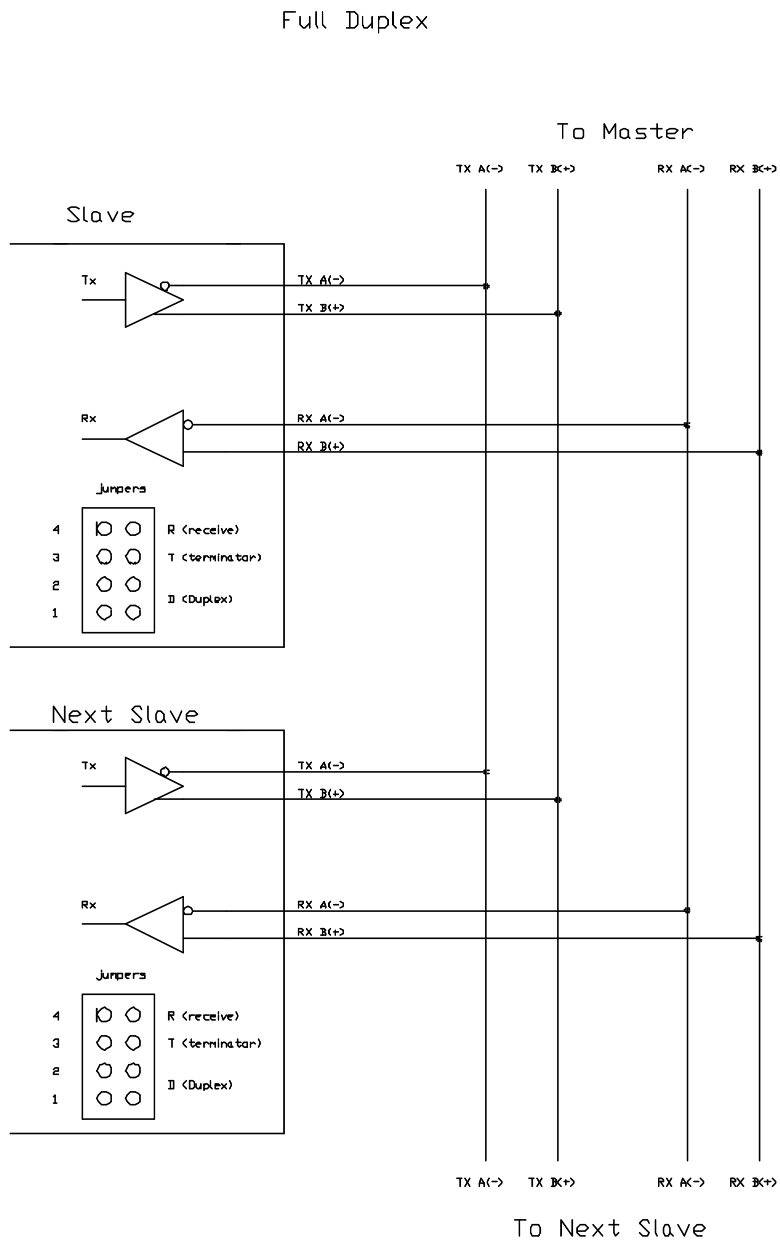


Figure 13-19: Full Duplex Wiring Schematic

20mA Current Loop Module

This section describes the installation of the 20mA Current Loop module. Installing this module will convert comm port 1 from RS-232 to 20mA current loop (not to be confused with 0-20mA or 4-20mA analog output).

To install the 20 mA Current Loop Module (all models):

1. Open the indicator and gain access to the main board.
2. Remove the U17 IC (on the 660 Series main board) or U3 IC (on the 460 Series main board) from its socket.
3. Remove the white wire jumper from the IC socket where the chip in step 2 was removed.
4. Snap the plastic spacers into the three mounting holes.
5. Gently press the option board into the socket.
6. Apply the included sticker to comm port 1. The label will go over the silk screen on the board. Use the portion of the label titled “60 Series.” Align TA over GND on comm port 1 and adhere the label over the silkscreen.
7. Reassemble the indicator.

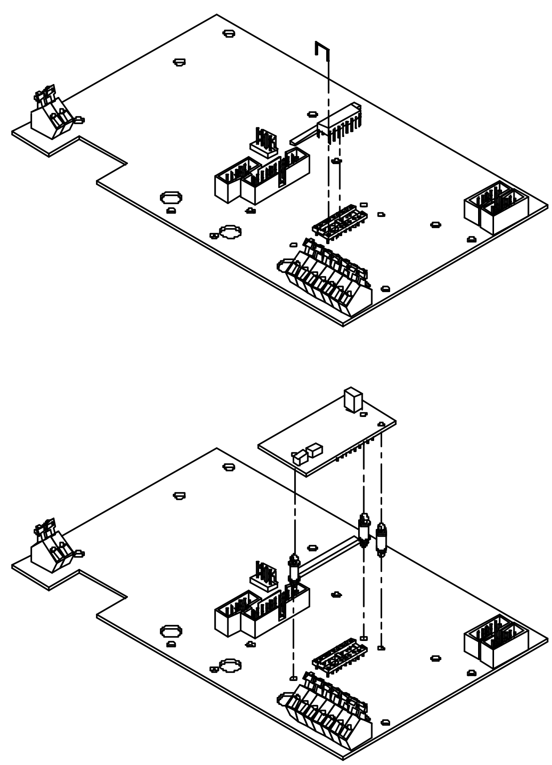


Figure 13-20: 460 Series 20 mA Current Loop Installation

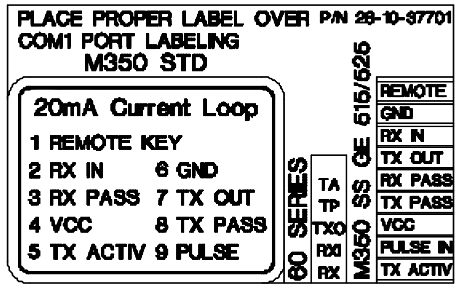


Figure 13-21: Label for 20mA COMM PORT pin designations

20MA PARAMETER SETUP

The 20mA module does not have to be enabled in the setup mode. The module simply converts the standard RS-232 communication on comm port 1 to 20mA current loop.

20 MA CURRENT LOOP CONNECTIONS

Apply the label over the silk screening on comm port 1 as previously described (J5 of the 460 Series or J6 of the 660 Series).

Transmit Current Input Active = TA
Transmit Current Input Passive = TP
Transmit Output = TXO
Receive Current Input = RXI
Receive Current Output = RX

The signal is bi-directional. Both the transmit output and the receive input of the indicator are available as 20 mA signals. The handshaking signals are not supported by the 20 mA current loop. Only baud rates of 9600 or less are supported.

The TXO output may be used as an active or passive output from the indicator. Either active or passive is chosen depending upon which terminals are used for the connections. In active mode the indicator supplies the current. In passive mode, the external device supplies the current. The Rx input is available in passive mode only.

The input and output are electrically isolated from the main boards, earth ground and each other. This applies for both passive and active modes. Isolation is a minimum of 1000 volts.

The active mode transmit current loop provides a driving voltage of 12 VDC. This will allow 20 mA current flow with up to a 600 ohm load. Passive mode will work with an external driving voltage of up to 50 VDC.

SCR MODULE

The 660 Series contains components which could be damaged by Electrostatic Discharge (ESD) if serviced improperly. Use proper ESD precautions (wear a wrist strap connected to ground, use grounded work stations, etc.) when opening the enclosure.

High voltages may exist within the enclosure! To prevent the risk of electrical shock, ALWAYS unplug the indicator when opening the enclosure. Installation and servicing of the indicator should be performed by authorized and qualified service personnel only.

Never connect or disconnect option board cables while the indicator is powered. Doing so may result in circuit board damage.

This section provides procedures for installing the SCR module.

SPECIFICATIONS

P/N: 24660B-102C0

Maximum Load Current:	1.75A RMS (continuous)
Load Voltage:	20-280 VAC
SCR Switching:	AC Neutral
Control Voltage:	0–5 VDC (Pulse)
Mounting:	Standard DIN Rail (model 663)
Dimensions:	2.45”L x 0.88”W x 3.88”H 62.2mmL x 22.4mmW x 98.6mmH

A maximum of seven SCR modules may be controlled independently by one 660 Series controller.

INSTALLATION

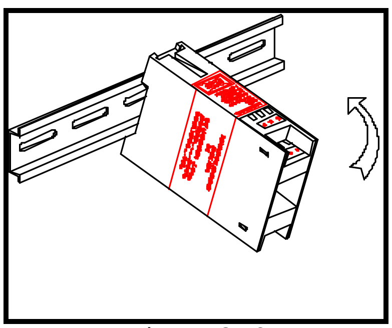


Figure 12-16

To install the SCR Module onto the DIN rail of a 663 enclosure:

1. DISCONNECT POWER TO THE CONTROLLER.
2. Position the SCR Module with the terminals marked A-F face up.
3. Hook the bottom groove of the SCR Module under the bottom DIN rail edge as shown in Figure 12-16.
4. Lift the module upward until the upper spring tab snaps onto the upper DIN rail edge.

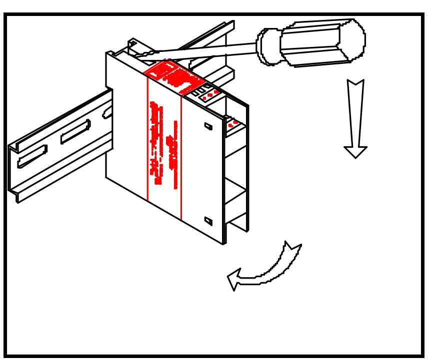


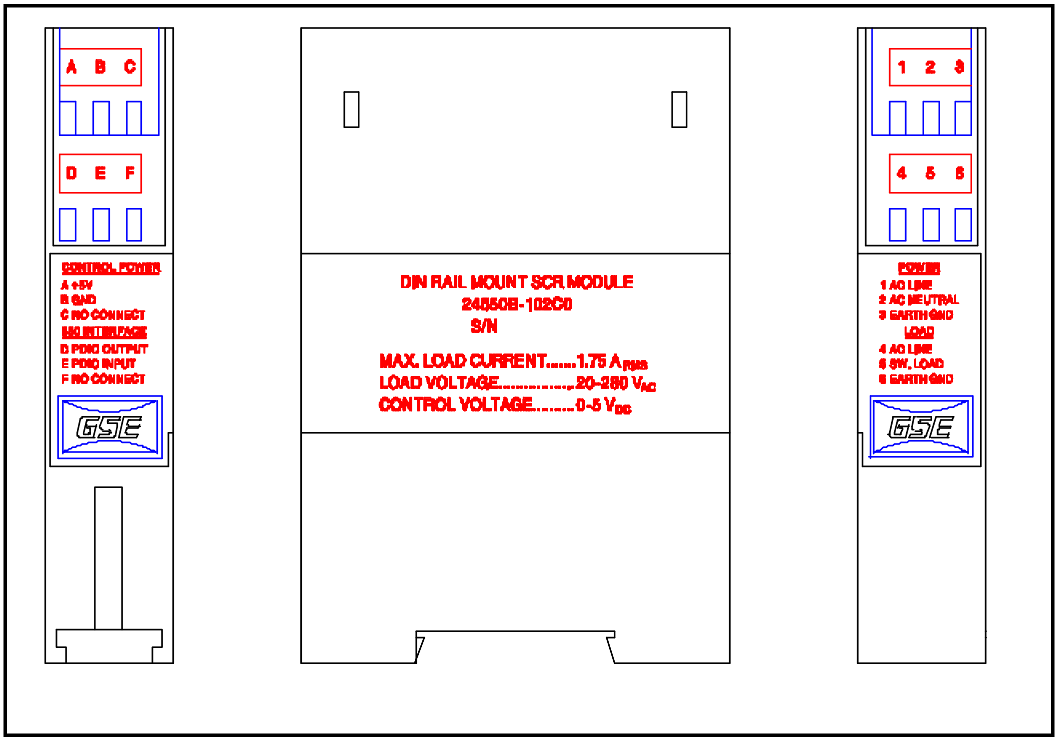
Figure 12-17

To remove an SCR Module:

1. DISCONNECT POWER TO THE CONTROLLER.
2. Remove all wiring from the SCR Module terminals.
3. Insert the tip of a small flat ended screwdriver into the slot in the upper spring tab.
4. Pull the screwdriver downward to lift the spring tab from beneath the upper DIN rail edge as shown in Figure 12-17.
5. Pull down on the SCR module to remove.

Do NOT remove the circuit board from its enclosure! There are no user serviceable parts inside. Improper orientation of circuit board could result in damage to module, controller and device.

CONNECTIONS



SCR Module	Terminal Description	660 Series PDIO (J11)	AC Power	Control Device
1	AC LINE		LINE	
2	AC NEUTRAL		NEUTRAL	
3	EARTH GROUND		GROUND	
4	AC LINE			LINE
5	SWITCHED LOAD			NEUTRAL
6	EARTH GROUND			GROUND
A	+5 VDC	+5V		
B	DIGITAL GROUND	GND		
C	NO CONNECT			
D*	PDIO OUTPUT	IO2		
E**	PDIO INPUT	IO1		
F	NO CONNECT			

* Use +5V, GND and PDIO J11 output terminals IO3-IO8 to add SCR modules 2-7 respectively.

** Only one module is required to supply the zero-cross signal on the PDIO input terminal.

SCR MODULE PARAMETER SETUP

SCR parameters beginning at P850 → P862 are used to setup the output module. Configuring an SCR module requires two consecutive PDIO channels for the first module. The first channel must be configured as a

"Delay Input", the next channel as a "Delay Output". Additional SCR modules can be added by configuring successive PDIO channels as "Delay Outputs" (only one "Delay Input" channel is required for multiple SCR modules). Refer to page 3-41 for complete configuration details.

TESTING THE SCR MODULE

To test an SCR module configured for output on channel #2:

1. Select the standard weigh mode (i.e. gross weight display).
2. Key In **50.2 [SELECT]** to select the pulse width parameter.
3. Key In **1 [ENTER]** to enter a pulse width of 1 ms.
4. Key In **51.2 [SELECT]** to select the delay output parameter.
5. Key In **4 [ENTER]** to enter a pulse output delay of 4 ms.
6. Key In **78 [SELECT]** to select the manual setpoint control mode.
7. Key In **132 [ENTER]** to select setpoint 132 for manual control.

   This setpoint may have no effect if setpoint 132 is configured as an output.
8. Press **[F5]** to activate setpoint 132 (runs device).
9. Press **[F3]** to deactivate setpoint 132 (stops device).
10. Press **[SELECT]** to return to the standard weigh mode.

    If P862 is not configured for gating, the SCR output activates immediately upon assigning a value to the delay output pulse (P51.2).

460 Series Time/Date Module

The time / Date option module installs in the Time / Date Option socket located below the J9 Remote Key connector. See Figure 13-22

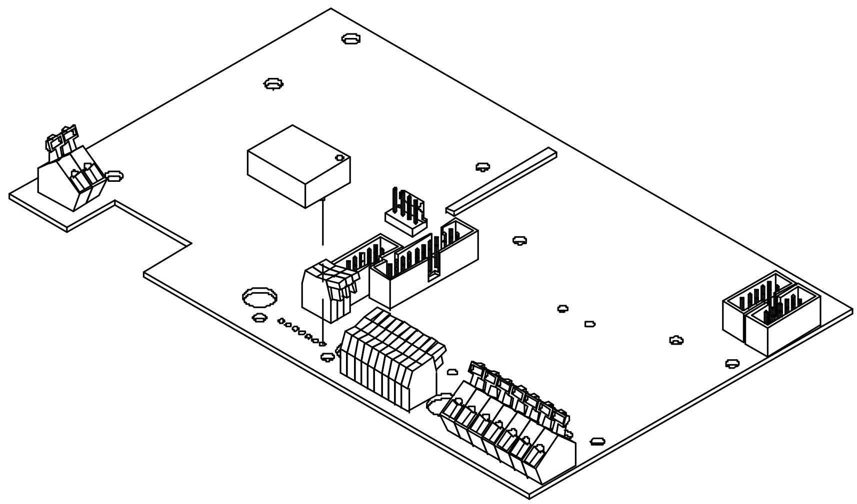


Figure 13-22: 460 Time / Date Option Installation

Make sure the dot on the time / date module is located in the upper right corner to match up with the dot on the main board. The dot designates pin 1. Refer to 3-37 for detailed instructions on setting the time and date.

Splash Guards

The GSE Splash Guard is a vinyl bag that encases the controller. The Splash Guard is available for the 660, 661, and 662 controllers. The GSE part number for this product is 44-70-37438.

DuraShield

The GSE Dura-Shield provides front panel protection for stainless steel enclosures used in heavy washdown environments. A clear Lexan® cover, the Dura-Shield adheres to the metal portion of the front panel and not to the display lens or keypad, shielding the controller from the elements while leaving the display unobscured, the keys easily pressed. The non-yellowing Lexan® is resistant to abrasion and chemical attack.

GSE offers the following Dura-Shield variations:

- For the 660, 661, and 662 (and their panel-mount equivalents): part number 31-70-35319

- For the 665: part number 31-70-35900
- For the keypad only: part number 31-70-36924

ALPHA KEYPAD

The Alpha Keypad Option is designed for installation on GSE 663 controllers. The Alpha Keypad complements the standard keypad. It simplifies the task of entering alphabetic data and facilitates programming.

Depending on the application requirements, the Alpha Keypad can be installed initially, or it can be added later. The GSE part number for the Alpha Keypad Option kit is 24663B-201B0.

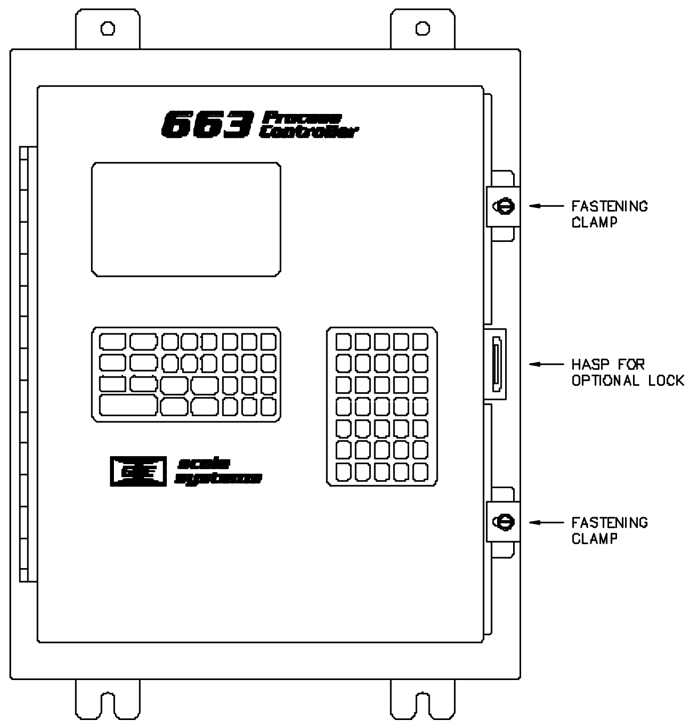


Figure 13-23: 663 with Alpha Keypad

ALPHA KEYPAD FEATURES

The Alpha Keypad is made of a durable silicon rubber, well-suited to the industrial environment. It has 35 keys arranged in seven rows of five keys and offers the following features:

- Alphabetic keys are arranged in alphabetic order, to be read from left to right, top to bottom.
- Alphabetic and ASCII characters make data entry straightforward and convenient, and programming easier.

- Alphabets are displayed, by default, in an *all caps* format.
- Alphabets can be displayed in a *lower case* format by pressing the lower case shift key at the same time the desired alphabetic key is pressed.
- A secondary ASCII character function is assigned to each of the 26 alphabetic keys and the colon key. Character functions are indicated in the lower, gray-shaded portion of the keys.
- Non-alphabetic characters are displayed when the “2nd” key is pressed at the same time the desired character key is pressed.
- The “%” (percent sign) key can be pressed *once* to display a percent sign, or in conjunction with the “2nd” key to enable the next character that is pressed to define a command.
- Non-alphabetic, non-character *function* keys are located on the bottom row.
- The “ENTER” (↵) key is located in the lower right corner for convenient confirmation of data entry.
- The glossy, elastomeric keys are wear-resistant and can be cleaned easily with a soft, damp non-abrasive cloth.
- Keys are ergonomically sized and highly responsive to input.
- Protocols for the Alpha Keypad are fully selectable.

FUNCTION KEYS

Five function keys are located in the bottom row of the keypad:

- Lower case shift key
- “2nd” key to be used during selection of a non-alphabetic character key
- “ ← ” (LEFT arrow), or backspace key
- “ → ” (RIGHT arrow), or space key
- “ ↵” (ENTER key), which transmits “%e” (the *enter command*) to the Process Controller

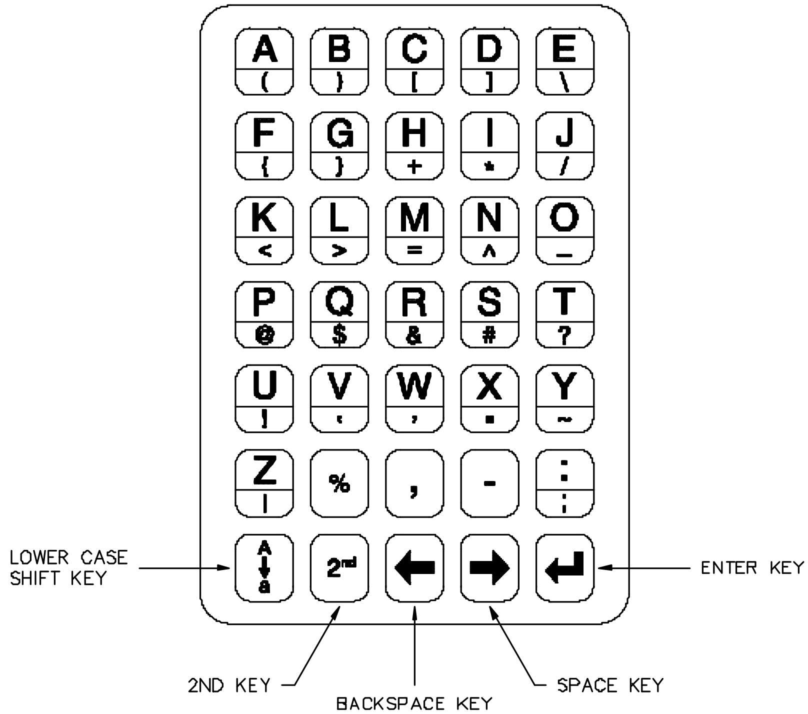


Figure 13-24: Alpha Keypad

INSTALLING THE ALPHA KEYPAD

Installation is easy, requiring few tools. Before you begin installation, review the required tools and make sure you have all the parts supplied with your kit.

TOOLS REQUIRED

The table below lists the tools required to install the Alpha Keypad into your controller.

Table 13-5: Required Tools

Large flat tip screwdriver to loosen and tighten ¼-20 slotted pan head screws in the exterior door fastening clamps.
7 mm (9/32 in) nut driver for unfastening and fastening nuts.
Small flat tip screwdriver for installing wires into the controller main board connector.

PARTS SUPPLIED

QUANTITY	DESCRIPTION	GSE PART NUMBER
1	Alpha Keypad	420813-31168
1	Alpha Keypad Cable	299290-3720012X
1	Cable Clamp	31-80-0140
1	Insulator	44-30-31317
1	Spacer	44-70-311972
1	Tie Wrap	31-80-0175

The figure below shows the parts included in the Alpha Keypad Kit.

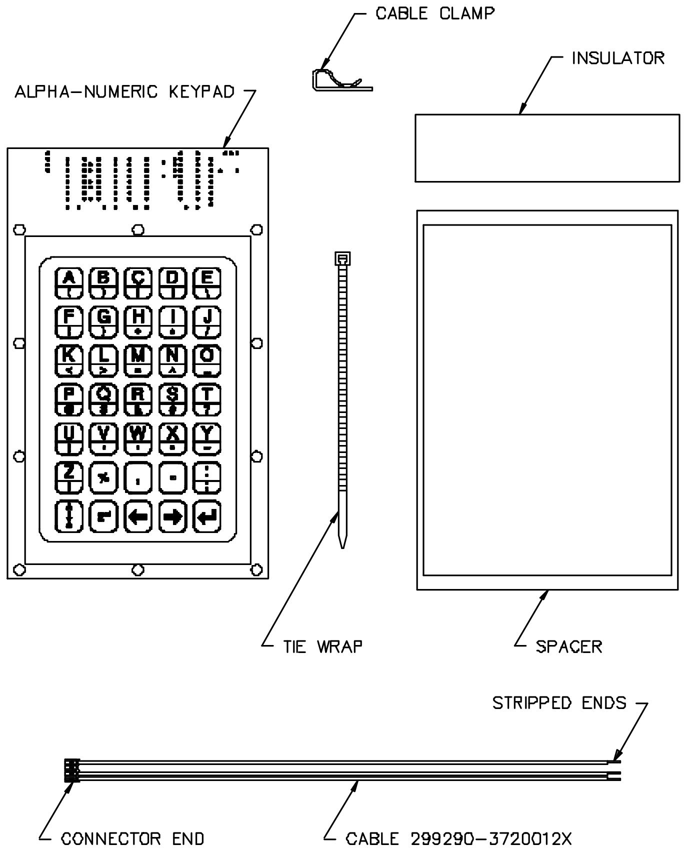


Figure 13-25: Alpha Keypad Parts Kit

The 60 Series instruments contains components which could be damaged by Electrostatic Discharge (ESD) if serviced improperly. Use proper ESD precautions (wear a wrist strap connected to ground, use grounded work stations, etc.) when opening the enclosure.

High voltages may exist within the enclosure! To prevent the risk of electrical shock, ALWAYS unplug the indicator when opening the enclosure. Installation and servicing of the indicator should be performed by authorized and qualified service personnel only.

Never connect or disconnect option board cables while the indicator is powered. Doing so may result in circuit board damage.

INSTALLATION

To install the Alpha Keypad option:

1. Turn off the Process Controller by disconnecting it from the AC power supply.

2. If a lock is installed through the hasp on the right side of the Process Controller enclosure, remove the lock.
3. Using a large, flat-tip screwdriver, loosen the spring-loaded 1/4-20 slotted pan head screws in the two fastening clamps on the right side of the Process Controller, and slide the clamps to the right.
4. Open the Process Controller door.
5. Remove the RF cover.
6. If the optional shield housing is installed on the back of the door, use a 7 mm (9/32 in) nut driver to remove the 13 hex nuts that secure it, and then remove the housing. The housing must be removed to gain access to the door-mounted hardware.
7. Locate the plate that covers the opening for the Alpha Keypad Option. The plate is near the lower left corner of the door interior on the 663 Process Controller.
8. Using a 7 mm (9/32 in) nut driver, remove the 10 hex nuts from the studs that retain the cover plate.

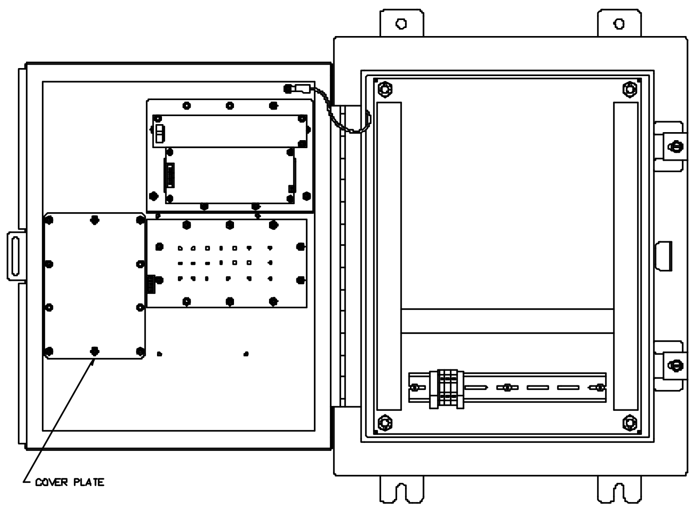


Figure 13-26: Location of The Cover Plate on The 663

9. Remove the cover plate from the retaining studs.
10. Remove the gasket.
11. Remove the Alpha Keypad and other kit parts from the plastic bags.
12. Install the spacer over the retaining studs.

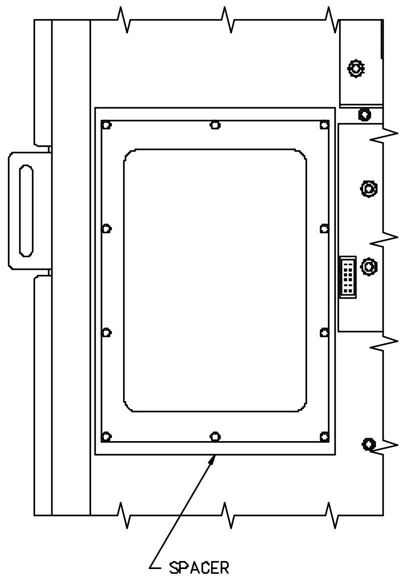


Figure 13-27: Installing The Spacer

13. Remove the backing from the adhesive side of the rectangular insulator.
14. Center the insulator directly above the spacer, pressing it firmly into place.

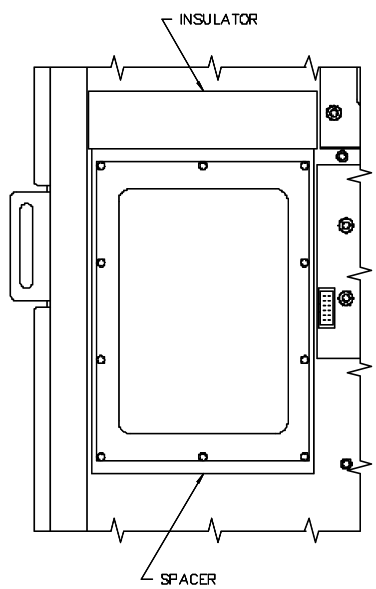


Figure 13-28: Positioning The Insulator

15. Install the Alpha Keypad onto the retaining studs, over the insulator and spacer.

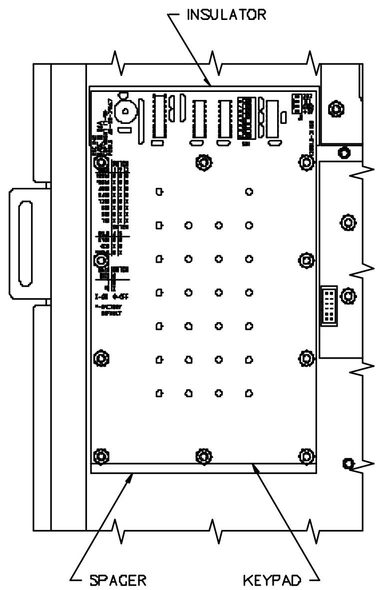


Figure 13-29: Installing The Alpha Keypad

16. Install the 10 hex nuts on the retaining studs, and tighten them using a 7 mm (9/32 in) nut driver. Do not over tighten to distort the keypad.
17. Remove the backing from the adhesive side of the cable clamp.

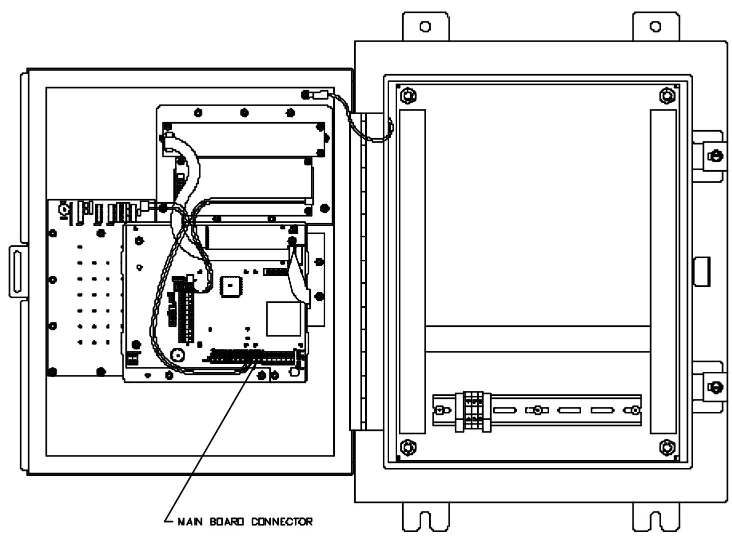


Figure 13-30: Installing The Cable on The 663

18. Set the dual in-line package (DIP) switch. The factory protocol settings are 9,600 baud, no parity, and no data bits.

Baud	Switch			Parity	Switch	
	1	2	3		4	5
**19,200	0	0	0	**Even	0	0
*9,600	X	0	0	Odd	X	0
4,800	0	X	0	*No Parity	0	X
2,400	X	X	0	**Data Bits**	Switch	
1,200	0	0	X		6	
600	X	0	X	7	0	
300	0	X	X	***8	X	
150	X	X	X			

X = ON, 0 = OFF
* = FACTORY DEFAULT (option received before 6/1/96)
** = FACTORY DEFAULT (option received after 6/1/96)
*** = FACTORY DEFAULT (received before or after 6/1/96)
Switches 7 and 8 have no function and can be set X or 0.

The Process Controller has independent selections for the baud, parity, and data bits for *each of the* COMM ports.

Use the *highest* baud rate setting *common to all connected devices* — displays, keypads, printers, personal computers, etc. — when setting the Alpha Keypad DIP switch.

If any of the devices can accept even *or* odd parity, use *either* setting, otherwise, use the *no parity* setting.

Use the 8 data bits setting — the preferred setting assuming each device on the specified port can accept 8 data bits. For example, if the connected device interfaced to COM2 can operate at 19,200 baud, even parity, and 8 data bits, set the 660 COM2 2 port to match this baud rate and protocol.

Check the documentation supplied with each device to determine the recommended protocol settings.

You can view device settings on the Process Controller by keying in "199 [SELECT] 23640 [ID] [ENTER]" using the standard keypad select the desired COMM port. Then view parameters P200 for baud rate, P201 for data bits, and P202 for parity.

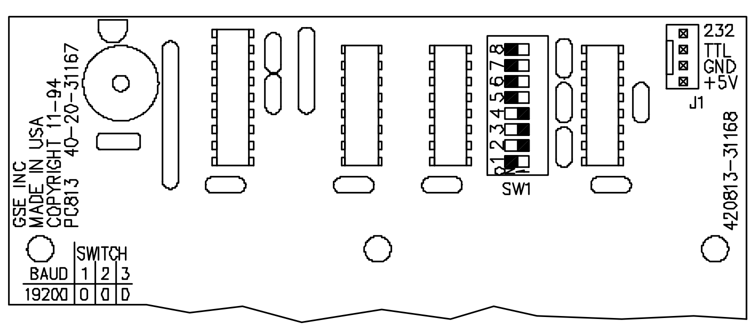


Figure 13-31: DIP Switch and Four-Pin Cable Connector on the Alpha Keypad

19. Install the three-color cable to the Alpha Keypad.: To do this, insert the connector end of the cable over the four-pin cable connector (J1) on the top right corner of the Alpha Keypad.

The pins are marked 232, TTL, GND, and +5V. The connectors “snap” together when they are mated properly. Be sure the wires are connected as follows:

- *No wire* mates with the TTL pin.
- The white wire mates with the 232 pin.

20. Twist the stripped ends on each of the three cable wires.
21. Replace the RF cover.
22. Close the Process Controller enclosure door.

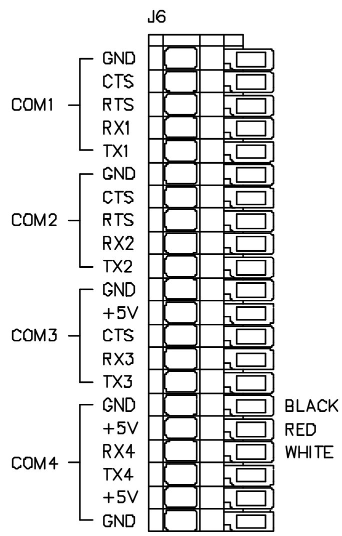


Figure 13-32: Connecting The Black, White and Red Cable Wires to The Main Board Connector

23. Slide the exterior door fastening clamps to the left.
24. Using a large, flat-tip screwdriver, tighten the 1/4-20 screws on the fastening clamps.
25. Install the lock, if you have one.
26. Connect the Process Controller to an AC power supply to power it *on*.

    Installation is complete.

USING THE ALPHA KEYPAD

This section summarizes the functions of the Alpha Keypad option.

KEYING IN ALL CAPS DATA

To key in alphabetic information in an *all caps* format, simply press the desired alphabetic keys. View the display.

KEYING IN LOWER CASE DATA

To enter alphabetic information in a *lower case* format, hold in the lower case shift key in the bottom left corner of the Alpha Keypad along with the desired alphabetic keys. View the display.

CONFIRMING DATA ENTRY

When the data you have keyed in appears on the display to your satisfaction, press the ENTER key (↵) located in the lower right corner of the Alpha Keypad. This action transmits the *Enter* command (%e) to the Process Controller.

KEYING IN ASCII CHARACTERS

To enter non-alphabetic ASCII characters, press the "2nd" key (on the bottom row of keys) along with the desired character, which appears in the lower, gray-shaded portion of the Alpha Keypad keys.

KEYING IN A PERCENT SIGN

To key in a percent sign (%), press the "%" key. This action transmits two percent sign signals to the Process Controller, but creates just one percent sign on the display.

Example: Key in "**2%Q**" to insert the command to send custom transmit #2.

INITIATING A COMMAND

To initiate a command, press the "2nd" key along with the percent sign (%) key. This action transmits one "%" character to the Process Controller, enabling the next character you press to represent a command.

Example:

Key in "2[2nd]%Q" to send custom transmit #2 manually. (You press 2, then holding in the "2nd" key, you press %, and then you press Q).

BACKSPACING

To backspace, press the left arrow (←) key.

SPACING

To create a space in the text, press the right arrow (→) key.

TROUBLESHOOTING

If the Process Controller is not operating, do the following:

- Be sure you have connected the Process Controller to the AC power supply to power it *on*.
- Make sure the red, white, and black wires from the Alpha Keypad to the main controller board are connected as specified.
- COM4 Port protocol is improperly set and does not match the alpha keypad protocol.
- Check the display for error messages.
- The **[¿]** key responds improperly (functions as a "clear" operation instead of an "enter". The firmware <u>must</u> be updated to a later version.

ERROR MESSAGES

Two communications error messages are associated with an improperly installed Alpha Keypad. These will appear on the two lines of the dot matrix display. The messages, causes, and probable remedy follow:

frmgX error	This indicates that the stop bit of a received character did not occur when it was expected. This could be the result of an incorrect baud rate (P200), incorrect number of data bits (P201), or incorrect parity setting (P202).
prtyXerror	This indicates that the parity of a received character did not match the parity in the Process Controller Setup Mode, parameter P202. This also could result if the baud rate (P200) or the number of data bits (P201) are incorrect.

REPLACEMENT PARTS

Parts on the Alpha Keypad Option that can be replaced without replacing the entire Alpha Keypad Option kit include:

- Cable (GSE part number 299290-3720012X)
- Cable clamp (GSE part number 31-80-0140)
- Rubber keypad (GSE part number 44-35-30808)

ALPHANUMERIC SERIAL KEYBOARD CONVERTER KIT

P/N: 24574B-401A0

INSTALLATION

To connect the Serial Converter to the Keyboard:

- Simply connect the standard PS2 female connector of the serial converter to the mating keyboard connector.
- The serial converter may also be connected directly to an RS-232 port by cutting off the DB-9 connector and wiring it directly to the serial port terminals according to the color codes listed on the converter label.

CHARACTER MAP

The converter will send the standard ASCII value for each key, except as noted in the following table.

Table 13-6: GSE Defined Keypress Functions

Keypress Combinations	Decimal Output	Hex Output	GSE Function
F1	128	80	Invoke Macro #1
F2	129	81	Invoke Macro #2
F3	130	82	Invoke Macro #3
F4	131	83	Invoke Macro #4
F5	132	84	Invoke Macro #5
F6	133	85	Invoke Macro #6
F7	134	86	Invoke Macro #7
F8	135	87	Invoke Macro #8
F9	136	88	Invoke Macro #9
F10	138	8A	Invoke Macro #10
F11	187	BB	Invoke Macro #11
F12	188	BC	Invoke Macro #12
Alt ~`	224	E0	SCALE SELECT
Alt C	227	E3	CLEAR
Delete	277	E3	CLEAR
Return	229	E5	ENTER
Alt I	233	E9	ID
Alt P	240	F0	PRINT
Alt S	243	F3	SELECT
Alt T	244	F4	TARE
Alt U	245	F5	UNITS
Alt Z	250	FA	ZERO
Alt Esc	248 and 255	F8 and FF	Macro Abort
Shift %5	37 and 37	25 and 25	Double ‘**%**’ Character
Shift Alt %5	37	25	Single ‘**%**’ Character

Note: Some functions do not apply to certain weigh indicators.

CABLES

This section describes the cables used between the 660 Series controllers and peripherals.

SCANNER/KEYBOARD ADAPTER CABLE

(GSE Part Number 299250-42020)

For use with the Symbol laser scanner (LS8125 & LS8525), the adapter (GSE Part # 210625-SSA01) is required in order to use this cable. Cable length is 2 feet.

This cable interfaces the 660 Series controller to a barcode scanner and keyboard. If both a scanner and keyboard are to be connected into one controller, two of these cables are required. Since the scanners and keyboards are not washdown devices, the connectors used are simply standard D-Type connectors.

Table 13-7: Scanner Adapter Cable Connections

660 Series Connection on J6 RS-232	Color	PIN#
TTL – not used on 60 Series	WHITE	2
RX1, RX2, RX3 or RX4	GREEN	3
GND	BLACK	5
+5V	RED	9

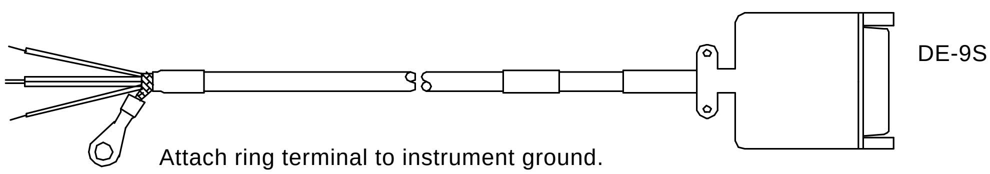


Figure 13-33: Scanner/Keyboard Adapter Cable

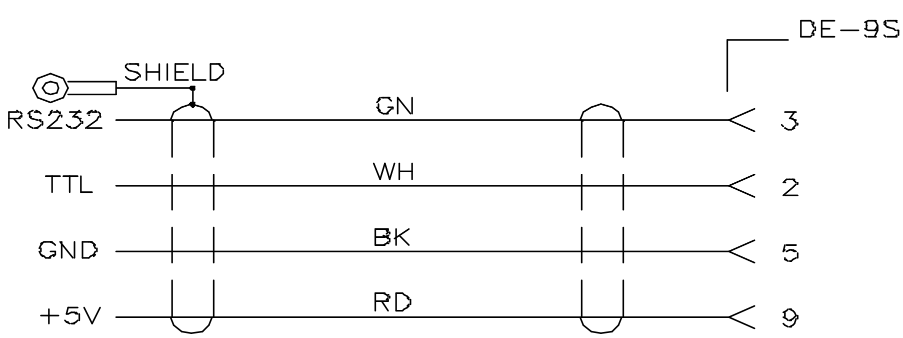


Figure 13-34: Scanner/Keyboard Adapter Cable Wiring Diagram

RS232 CABLE FOR 660 TO STANDARD PRINTER

(GSE Part Number 299240-45080)

This cable is used to interface a 660 Series controller to a standard line printer. The cable length is 8 feet.

Table 13-8: Computer Style Printer Cable Connection

660 Series Connection on J6 RS-232	Color	PIN#
TX1, TX2, TX3 or TX4	RED	3(RXD)
GND	BLACK	7(GND)
CTS	WHITE	20 (BUSY)

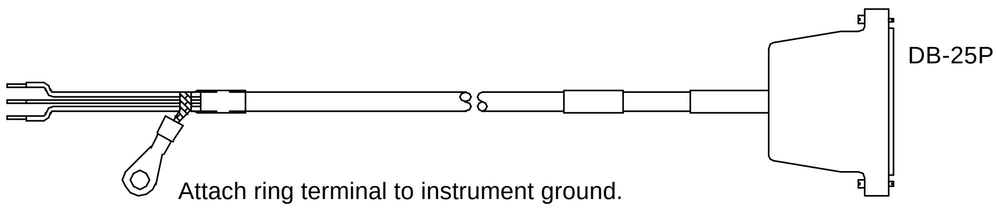


Figure 13-35: Standard Line Cable Printer

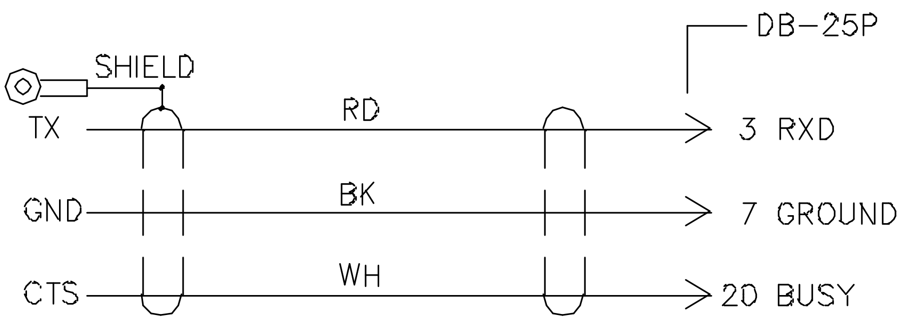


Figure 13-36: Standard Line Printer Cable Wiring Diagram

660 SERIES TO ELTRON THERMAL PRINTER

(GSE Part Number 299240-48080)

This cable is used to interface a 660 Series controller to an Eltron Thermal Printer. The cable length is 8 feet.

Table 13-9: Eltron Printer Cable Connections

660 Series Connection on J6 RS-232	Color	PIN#
TX1, TX2, TX3 or TX4	RED	3(RXD)
GND	BLACK	5(GND)
CTS	WHITE	6 (RDY)

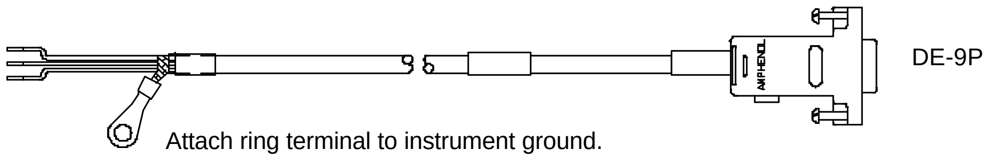


Figure 13-37: Eltron Printer Cable

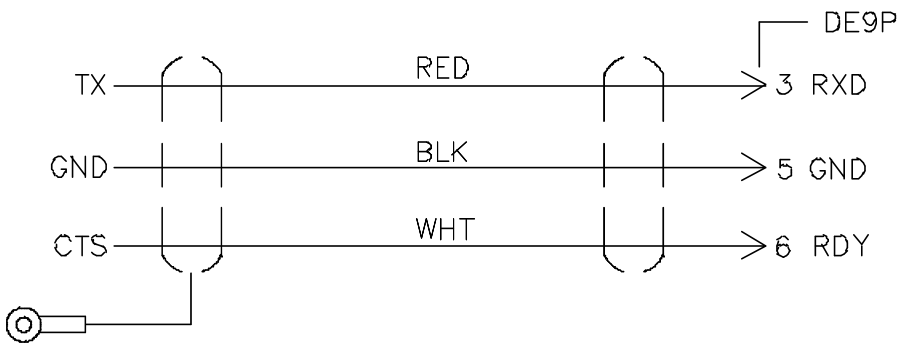


Figure 13-38: Eltron Printer Cable Wiring Diagram

660 SERIES TO COMPUTER CABLE

(GSE Part Number 299250-44080)

This cable is used to interface a 660 Series controller to a PC/AT or Notebook type computer. The cable length is 8 feet.

Table 13-10: PC/AT or Notebook Type Computer Cable Connections

660 Series Connection on J6 RS-232	Color	PIN#
TX1, TX2, TX3 or TX4	RED	2(RXD)
GND	BLACK	5(GND)
RX1, RX2, RX3 or RX4	WHITE	3 (TXD)

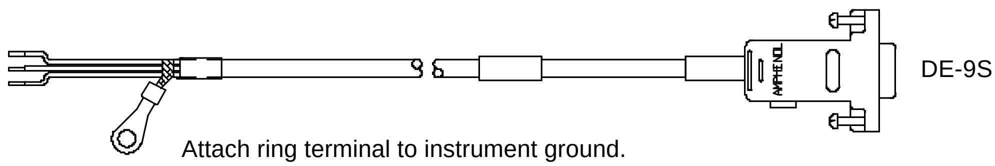


Figure 13-39: PC/AT or Notebook Computer Cable

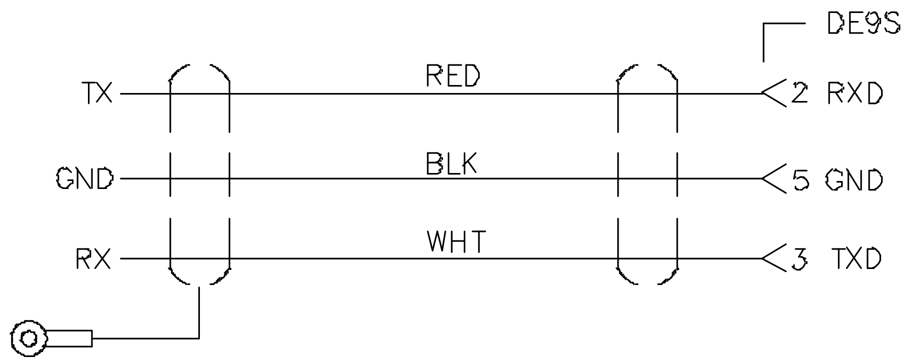


Figure 13-40: PC/AT Computer Cable Wiring Diagram

4X20 VF DISPLAY

The 4X20 VF display is a standard feature of the 661 and of the VFD version of the 663 and 665.

HARDWARE SETUP

The 4X20 VF display is installed at the factory. It connects to the main board at the J7 header via a 3-wire cable (see Table 13-11).

Table 13-11: 4X20 VFD Interface Cables

4X20 VFD Interface Cable Part#	Model	Length
299290-3711610X	663 All Panel Mounts	10”
299290-3711610X	661 665	18”

The J7 connector supplies +5VDC, Ground and the TTL transmit line to the display.

⚠ The 4X20 VF display is addressed via comm4. However, DO NOT connect the display to the RS-232 comm4 terminals of the J6 communications connector! The display is not RS-232 compatible. The J7 4X20 VFD connector converts the comm4 transmission to the display to a TTL signal. Since both J7 and J6 (comm4) transmit terminals carry the same transmission data, it is not possible to connect another device to the J6 comm4 when using the 4X20 VFD.

PROTOCOL SETTINGS

The 4X20 VF display is hardware configured for 19200 baud, 8 data bits, 1 stop bit, even parity and international character set. These settings can be changed by changing the jumper settings of J10 of the display board (see Table 13-12).

Table 13-12: 4X20 VFD Jumper Settings

Function	Selection	J0	J1	J2	J3	J4	JA
Baud	19200	O	O	O			
	9600	X	O	O			
	4800	O	X	O			
	2400	X	X	O			
	1200	O	O	X			
	600	X	O	X			
	300	O	X	X			
Parity	Even				O	O	
	Odd				X	O	
	None				O	X	
Character Set	International						O
	Katakana						X
Factory Setting		O	O	O	O	O	O
X = Shorted Jumper; O = Open Jumper							

DISPLAY TEST

Having the TTL signal connected to Ground at power-up will initiate a display test with the entire character set scrolling across the 4 lines of the display. To end the display test, remove power and reconnect the TTL signal to its proper terminal.

DISPLAY COMMANDS

Table 13-13 lists the various control commands used to address the 4X20 VFD. These commands can be transmitted to the display as control codes via custom transmit or macro %& command.

Table 13-13: 4X20 VFD Control Commands

DECIMAL VALUE (CONTROL CODE)	DESCRIPTION
008	Move cursor left one space
009	Move cursor right one space
010	Move cursor down one row
012	Move cursor to top left corner (home)
013	Move cursor to beginning of line
014	Clear entire display
017	Scroll mode OFF

DECIMAL VALUE (CONTROL CODE)	DESCRIPTION
018	Scroll mode ON
020	Cursor OFF
021	Cursor ON
024	Set INTERNATIONAL character set
025	Set KATAKANA character set
027, 067	Define custom character
027, 072	Position cursor at specified location
027, 073	Reset display default values
027, 076	Set display intensity
027, 084	Set cursor blink speed

CURSOR LEFT (008)

Moves the cursor one character to the left. When the cursor reaches the left end of a line, it moves to the right end of the line above. It does not move beyond the left end of the top line.

CURSOR RIGHT (009)

Moves the cursor one character to the right. When the cursor reaches the right end of a line, it moves to the left end of the line below. How the cursor moves when it reaches the right end of the bottom line depends on whether the Scroll Mode is off or on.

- If the Scroll Mode is off, the cursor moves to the left end of the top line.
- If the Scroll Mode is on, all displayed characters are scrolled up one line, the cursor moves to the left end of the bottom line, all characters on the top line disappear, and the bottom line is cleared.

CURSOR DOWN (010)

Moves the cursor down one line in the same column. How the cursor moves when it reaches the bottom line depends on whether the Scroll Mode is off or on.

- If the Scroll Mode is off, the cursor moves to the top line in the same column.
- If the Scroll Mode is on, all displayed characters are scrolled up one line, the cursor maintains its column position on the bottom line and the bottom line is cleared.

CURSOR HOME (012)

Moves the cursor to the left end of the top line.

CURSOR TO BEGINNING OF LINE (013)

Moves the cursor to the left end of the same line.

CLEAR DISPLAY (14)

Clears all displayed characters.

Scroll Mode Off (17)

Causes the cursor to advance from the lower-right corner to the top-left corner.

Scroll Mode On (18)

Causes displayed data to scroll up one line when the cursor advances past the lower-right corner. Previously displayed data on the top line is lost and the bottom line becomes blank.

Cursor Off (020)

Makes the cursor invisible until the alternative Cursor On code is selected. Cursor Off is the default cursor mode.

Cursor On (021)

Makes the cursor appear as a blinking, solid-block character. The blinking speed can be varied with the Set Cursor Blink Speed command.

Set International Character Set (024)

Selects the international characters to be displayed (see Appendix C). At power-up the international character set is selected by default.

Set Katakana Character Set (025)

Selects the katakana characters to be displayed (see Appendix C).

Define Custom Character (ESC + 067)

One or two custom characters can be created and displayed on the 4X20 VFD. When a third custom character is created, it replaces the first of the two custom characters previously created.

The following syntax is used to create a custom character:

ESC + C + < character > + D1 + D2 + D3 + D4 + D5

where 'ESC' is the escape character (027), 'C' is the custom character command (67), 'character' is the ASCII character to be redefined, and D1 through D5 determine which of the custom character's pixels will be on or off as shown below.

D1 **1**	D1 **2**	D1 **4**	D1 **8**	D1 **16**
D1 **32**	D1 **64**	D1 **128**	D2 **1**	D2 **2**
D2 **4**	D2 **8**	D2 **16**	D2 **32**	D2 **64**
D2 **128**	D3 **1**	D3 **2**	D3 **4**	D3 **8**
D3 **16**	D3 **32**	D3 **64**	D3 **128**	D4 **1**
D4 **2**	D4 **4**	D4 **8**	D4 **16**	D4 **32**
D4 **64**	D4 **128**	D5 **1**	D5 **2**	D5 **4**

D1 **1**	D1 **2**	D1 **4**	D1 **8**	D1 **16**
D1 **32**				D2 **2**
D2 **4**				D2 **64**
D2 **128**				D3 **8**
D3 **16**				D4 **1**
D4 **2**				D4 **32**
D4 **64**	D4 **128**	D5 **1**	D5 **2**	D5 **4**

The example above demonstrates how to create an empty box character. Begin by adding the values for D1 through D5:

D1 = 1 + 2 + 4 + 8 + 16 + 32 = 63
D2 = 2 + 4 + 64 + 128 = 198
D3 = 8 + 16 = 24
D4 = 1 + 2 + 32 + 64 + 128 = 227
D5 = 1 + 2 + 4 = 7

Next, choose a standard display character which will be replaced by the custom character and determine its ASCII value. Choose a character that will not be needed elsewhere on the display as it will appear as the custom character each time it is displayed. For example, if you choose the '@' character use the value 64 (refer to the ASCII chart in Appendix B).

Substituting the chosen ASCII value and the values calculated for D1 through D5, you can create the custom character with the macro command

```
4%”
27,67,64,63,198,24,227,7%&
```

Now each time the '@' character is sent to the 4X20 VFD it will be displayed as the empty box character.

POSITION CURSOR AT SPECIFIED LOCATION (ESC + 072)

The cursor can be moved to any position on the display using the following escape sequence:

ESC + H + < position >

where 'ESC' is the escape character (027), 'H' is the positioning command (072) and 'position' is a value between 0 and 79 as shown below.

ROW	DISPLAY POSITION

1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
2	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
3	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
4	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79

For example,

```
4%″
27,72,40%&
This is line #3%$
```

will send the text “This is line #3” beginning at position 40 of the display (beginning of row #3).

RESET DISPLAY DEFAULT VALUES (ESC + 073)

Clears all displayed characters and restores all settings to the factory defaults.

SET DISPLAY INTENSITY (ESC + 076)

When the controller is powered on, the 4X20 VFD is set to 100% brightness. The following syntax is used to control the display intensity:

ESC + L + < intensity >

where ‘ESC’ is the escape character (027), ‘L’ is the intensity command (76), and ‘intensity’ is the brightness value where

0 = approximately 25% brightness
64 = approximately 50% brightness
128 = approximately 75% brightness
192 = approximately 100% brightness

For example,

```
4%″
27,76,64%&
```

will set the display intensity to 50%.

SET CURSOR BLINK SPEED (ESC + 084)

The blinking speed of the cursor can be varied by the following escape sequence:

ESC + T + < speed >

where ‘ESC’ is the escape character (027), ‘T’ is the blink speed command (084) and ‘speed’ is a value between 1 and 255 in 30 millisecond intervals. For example,

```
4%″
27,84,25%&
```

will send the command setting the cursor blink rate to .75 seconds (25 X 30ms = 750ms).

The default blink rate is 20 (600ms).

REFLASH SOFTWARE

The ReFlash software is a utility which allows the indicator to be flashed through a communication port. The firmware is loaded into a flash memory IC. This allows for ease in loading firmware updates.

BDM FLASH SOFTWARE

The BDM flash software provides the means of flashing the boot loader and the firmware. If the boot block is somehow been corrupted, the BDM flash would be needed to restore it. The BDM connector is used on the main board (J4 on the 660 and J7 on the 460). This is a faster method of loading firmware.

Chapter 14 INFORMATION PARAMETERS

The Information parameters, P60000 → P65010, are used for informational and diagnostic purposes. These parameters include information pertaining to memory usage, serial numbers, audit trails, calibration, self diagnostics, debug and data transfer.

OVERVIEW

INFORMATION PARAMETERS

ACCESSING INFORMATION PARAMETERS

Example:
Accessing Information Parameters

[6] [0] [0] [0] [0] [SELECT]

```
P6000)  E2Ins
         4096
```

DO NOT ATTEMPT TO ACCESS THE INFORMATION PARAMETERS DURING CRITICAL WEIGHT PROCESSING! It is important to note that all functions of the operating mode will be suspended immediately upon accessing the information parameters. This includes suspension of weight conversions, deactivation of all setpoints and cancellation of custom transmits.

The information parameters can be accessed from within the setup mode or directly from the weigh mode. Key in the desired parameter number and press **[SELECT]** (see example - *Accessing Information Parameters*) . You will not be prompted to enter the setup access code in order to view the information parameters. However, in order to enter data in certain parameters you must first key in the access code:

[1] [0] [0] [SELECT] [2] [3] [6] [4] [0] [ID] [ENTER YES]

Example:
Accessing Multiple Instance Parameters

[6] [0] [0] [3] [2] [·] [2] [SELECT]

```
P6003@  NumOf
        Rows
```

```
P6003@    210
```

Navigating Information Parameters

Pressing **[SELECT]** advances to the next information parameter. Pressing **[SCALE SELECT]** moves back one parameter. To immediately access a specific parameter, key in the parameter number and press **[SELECT]**.

To access parameters with multiple instances such as databases and A/D calibration, include the instance number following the parameter number as shown in the example – *Accessing Multiple Instance Parameters*. Pressing **[ID]** will toggle through all display messages for multiple instance parameters for easy identification (see example – *Identifying Multiple Instance Parameters*). Refer to page 3-6 for additional information regarding multiple instance parameters.

Example:
Identifying Multiple Instance Parameters

[ID]

```
P6003@  DB  #:
             2
```

```
P6003@  NumOf
        Rows
```

```
P6003@    210
```

If the setup access code was entered, it is also possible to access the setup parameters from the information parameters. Key in the desired parameter number and press **[SELECT]**.

Exiting Information Parameters

Exit the information parameters by pressing **[ZERO]**. Operation will immediately resume from a power-up status if the information parameters were accessed directly from the weigh mode without entering the access code. If the access code was used, follow the setup mode exit routine as described on page 3-9.

MEMORY PARAMETERS

This manual references three types of memory:

- FLASH ROM   Re-Programmable Read-Only Memory
- EEPROM (E^2)   Electrically Erasable Read-Only Memory
- RAM   Random Access Memory

The FLASH ROM, U24 on the main board, contains the system firmware. It is programmed at the factory, and firmware can be upgraded in the field using the GSE ReFlash software (see page 13-65). The EEPROM, U27 & U28 on the main board, are used to permanently store all setup parameter data. Firmware operating instructions, along with the setup parameter data, are loaded into RAM at power-up where they are used to operate the system.

Table 14-1 describes the memory information parameters.

Table 14-1: Memory Information Parameters

INFORMATION PARAMETER	DESCRIPTION
P6000) E2Ins 8192	Displays the total EEPROM (E^2) size available for setup data storage.
P6000! E2Avl 7186	Displays the amount of E^2 available for setup parameter data. If P60040 is enabled, the display directs you to See P60040 for available setup parameter storage.
P6000@ RAMsz 256K	Displays the amount of RAM installed on the main board (64K for 460 Series; 128K for 560 Series; 256K for 660 Series).
P6000# RAMdy 192K	Displays the amount of dynamically allocated RAM for program execution. If P60050 is enabled, the display directs you to See P60050 for available RAM.
P6000$ RAMav 190K	Displays the amount of general purpose RAM available. If RAM is allocated for the macro debug trace buffer, the display directs you to See P60005. The amount of RAM reported at P60005 can also be allocated for general purpose RAM by clearing the macro debug trace buffer at P50001. Additional RAM may be allocated at P60050 if required for extremely large setup files.
P6000% MTBsz 170K	Displays the amount of RAM allocated for the macro debug trace buffer. All but a small amount (20K for the 660 Series; 16K for the 560 Series; 2K for the 460 Series)of remaining RAM is allocated for the debug buffer. No memory is allocated if no macros are programmed.
P6000^ Ftype AMD	Displays AMD or SHARP depending on if programming is similar to AMD or SHARP. Pressing **[ENTER]** will briefly display the actual manufacturer's code in hexadecimal (0x1 for AMD, 0x20 for ST, 0x B0 for Sharp).
P6000& Fsize 1024K	Displays the size of the flash in Kbytes (Bytes if less than 99999 bytes). Pressing **[ENTER]** will briefly display the actual device id in hex (0x50 = Sharp 8 megaBIT, 0x58 = ST 8 megaBIT, 0xD6 = ST 4 megaBIT, 0x0x2223 = AMD 4 megaBIT bottom boot block, 0x2258 = AMD 8 megaBIT bottom boot block).
P6000* F Avl 445K	Display shows the amount of blank flash left in Kbytes (Bytes if less than 99999 bytes). the amount of blank flash left in Kbytes (Bytes if less than 99999 bytes).
P6001) Opt'n RAM	Displays the size of an installed database option, either 256K, 1024K (1Meg), 2048K (2Meg) or 0K if a database is not installed. This number will be fixed at 4K for the 460 Series (there is not database option).
P6001! OpRAM Bytes	Displays the value of P60010 in terms of bytes, either 262144, 524288, 1048574 or 2097152.
P6001@ OpRAM Avail	Displays the amount of database RAM available for storing additional database records and/or allocating towards additional setup memory and RAM.
P6001# OpRAM Used	Displays the total amount of database RAM used.
P6001$ OpRAM Block	Displays the size of the largest contiguous block of memory available. Generally, this value will be approximately the same as P60012. However, database memory may become fragmented as the result of updating/deleting databases. Thus, there might not be enough memory to store large blocks of information even though P60012 may indicate sufficient memory.

INFORMATION PARAMETER	DESCRIPTION
P6001* OpBat Good	Displays the condition of the database battery voltage. The battery should be changed when the voltage drops below 2.50V, at which time OpBat Low is displayed. This parameter does not exist for the 460 Series.
P6002) DbRAM Used	Displays the amount of database RAM used specifically for storing database information.
P6002! DBase Error	Displays the result of the last database operation.
P6003) DB #: None!	Used to enter the database number for P60031 → P60033. For example, key in **2 [ENTER]**, then access P60031 → P60033 to view the name, number of rows and bytes used for database #2.
P6003! DBNam None!	Displays the database name assigned in the setup mode at P700 for the database specified at P60030. Press **[ID]** to view the database number.
P6003@ NumOf Rows	Displays the number of data rows for the database specified at P60030. Press **[ID]** to view the database number.
P6003# Bytes Used	Displays the total number of bytes used by the database specified at P60030. Press **[ID]** to view the database number.
P6003$ Bytes /Row	Displays the number of bytes per row required to store one row of information in the database specified at P60030. Press **[ID]** to view the database number.
Parameters P60040 → P60053 do not exist for the 460 Series.	
P6004) Setup RAM	Displays the amount of database RAM allocated for non-volatile storage of setup parameter data.
P6004! Setup Bytes	Displays P60040 in terms of bytes.
P6004@ Setup Avail	Displays the amount of Setup RAM allocated at P60040 that remains available for use.
P6004# Setup Used	Displays the amount of Setup RAM allocated at P60040 that has been used.
P6005) P-Ram Size	Displays the amount of database RAM allocated for general purpose RAM.
P6005! P-Ram Bytes	Displays P60050 in terms of bytes.
P6005@ P-Ram Avail	Displays the amount of Pile RAM allocated at P60050 that remains available for use.
P6005# P-Ram Used	Displays the amount of database RAM allocated at P60050 that has been used.

ALLOCATING DATABASE MEMORY FOR SETUP STORAGE

Database memory may be allocated for non-volatile storage of setup parameter data. This allows you to increase setup storage capacity beyond that provided by the EEPROM (E^2). Other benefits include:

- Faster access time for read/write operations such as saving setup changes
- Loading setup into RAM upon power-up and updating data registers
- Unlimited read/write cycles – will not degrade the life of the memory

With the exception of critical configuration parameters such as A/D calibration, audit trails, serial numbers, etc., the entire setup configuration is stored in the database when Setup RAM is allocated at P60040. Since

Example:
Allocating Setup RAM

[6] [0] [0] [4] [0] [SELECT]

P6004) Setup RAM

P6004) Disbl

[2] [4] [ENTER YES]

P6004) Sure? ???

[ENTER YES]

P6004) Setup RAM

P6004) 24K

ALWAYS DISCONNECT POWER BEFORE INSTALLING OR REMOVING OPTIONS!

A/D calibration must be re-entered after database Setup RAM is transferred to another unit.

database memory is used *instead* of E^2, there is no advantage to adding additional E^2.

Allocating Setup RAM

The example – *Allocating Setup RAM* shows how to allocate database Setup RAM. Setup RAM must be entered in increments of 1K and must be at least large enough to accommodate the existing setup. The maximum amount that may be specified cannot exceed the option RAM block size (P60014) plus any amount already allocated at P60040. Once specified, Setup RAM may later be increased if necessary.

Reducing Setup RAM

The Setup RAM size may be reduced providing it remains large enough to hold the existing setup. Key in the desired amount at P60040 and press **[ENTER]**. If the entry is invalid, the error message MinNV will indicate the minimum Setup RAM required.

To disable the Setup RAM, enter a value of zero (0). This will transfer the entire setup back to the E^2 provided there is enough E^2 installed to hold the setup.

Affected Parameters

- **P60001** will display See P60040 when Setup RAM is allocated.
- **P60050** may require additional RAM to be allocated.
- **P64000 & P64001** will add P60040 to the beginning of the download transmission if Setup RAM is allocated.
- **P65001& P65002** will disable and erase the Setup RAM as part of the default process.
- **P65010** will disable and erase the Setup RAM as part of the database reset process.

Removing & Installing A Database With Setup RAM

A database containing setup parameters in Setup RAM may be removed from one indicator and installed in another. This results in the complete transfer of setup parameters with the following exceptions:

- Scale Enable/Save/Disable (P109)
- OIML (P410)
- 4X20 Display Enable/Disable (P425)
- Keypad Type (P450)
- Serial Numbers & Audit Trails

These critical parameters remain stored in E^2, U27 on the main board. Note that the A/D calibration values (P61110 → P62221) are transferred along with the Setup Parameters. This will result in an inaccurate A/D calibration. To avoid this problem, acquire the A/D calibration values of the unit you are transferring Setup RAM to *before* installing the database option (see *Printing A/D Calibration Values* on page 4-11). Then enter these values separately *after* the database option is installed.

Installing a database containing Setup RAM will not erase the existing E^2 setup. However, the E^2 setup will not be accessed as long as Setup RAM is allocated. If database option is later removed, the system will revert to using the E^2 setup.

If Setup RAM is disabled at P60040, the system will attempt to transfer the setup to E^2, thus overwriting the previous E^2 setup. If there is not enough E^2 installed to hold the transferred setup no data will be transferred and an error message will display the minimum setup storage required.

ALLOCATING DATABASE MEMORY FOR PILE RAM

Of the 256K RAM installed on the main board, 178K is available for holding and executing setup parameters. This is more than adequate to handle any setup that can be stored in the maximum 16K E^2. However, larger setups that reside in the database Setup RAM may require additional "Pile" RAM.

As with Setup RAM, a portion of the database memory may be allocated to increase the amount of Pile RAM. This is only necessary for extremely large setups where available RAM at P60004 approaches zero. Another more obvious indication that additional RAM is required is the error message NoRAM AVAIL while uploading or changing a setup file.

Allocating Pile RAM

Example:
Allocating Pile RAM

[6] [0] [0] [5] [0] [SELECT]

P6005) P-Ram Size

P6005) Disbl

[5] [6] [ENTER YES]

P6005) Sure? ???

[ENTER YES]

P6005) P-Ram Size

P6005) 200K

The example – Allocating Pile RAM shows how to allocate database Pile RAM. Pile RAM must be entered in increments of 1K and must be at least large enough to accommodate the existing setup. The maximum amount that may be specified cannot exceed the option RAM block size (P60014) plus any amount already allocated at P60050. Once specified, Pile RAM may later be increased if necessary.

When allocated, database Pile RAM is used *instead* of main board RAM. Since there is always 178K of RAM available on the main board, it is never necessary to allocate less than 178K of Pile RAM.

Reducing Pile RAM

The Pile RAM size may be reduced providing it remains large enough to accommodate the existing setup. Key in the desired amount at P60050 and press **[ENTER]**. If the entry is invalid, the error message MinP= will indicate the minimum Pile RAM required.

To disable the Pile RAM, enter a value of zero (0). This will utilize the main board RAM provided the setup can be run with less than 178K RAM.

Affected Parameters

- **P60003** will display See P60050 when Pile RAM is allocated.
- **P64000 & P64001** will add P60050 to the beginning of the download transmission if Pile RAM is allocated.
- **P65001& P65002** will disable the Pile RAM as part of the default process.
- **P65010** will disable the Pile RAM as part of the database reset process.

Removing & Installing A Database With Setup RAM

RAM is volatile memory and only contains setup data after it is read from E^2 or database Setup RAM during power-up. Therefore transferring a database option from one unit to another does not pose any special considerations. However it may be useful to know that P60050 is always checked at power-up. If allocated, the database Pile RAM will always be used instead of main board RAM.

PROCESSING SPEED PARAMETERS

The processing speed information parameters indicate the microprocessor, RAM and ROM processing speed.

Table 14-2: Processing Speed Parameters

PROCESSING SPEED PARAMETER	DESCRIPTION
P6009) Clock 24MHz	Displays the current processing speed of the microprocessor. The speed is only changed indirectly by selecting various baud rates above 19200.
P6509! RomWS 0	Displays the number of ROM wait-states.
P6509@ RamWS FastT	Displays the number of RAM wait-states.

IDENTIFICATION PARAMETERS

The identification parameters contain firmware revision codes and serial numbers used to track the service history of each unit. Table 14-3 describes these parameters. Identification parameter values cannot be changed.

Table 14-3: Identification Parameters

IDENTIFICATION PARAMETER	DESCRIPTION
P6010) ©1995 *GSE*	Displays the software copyright. This parameter is also used to transmit A/D calibration data (page 4-11)and analog output calibration data (page 13-29).
P6010! 0660- 01???	Displays the firmware revision code. This number also serves as the EPROM part number as displayed on the EPROM label (240660-VVRRR, where V = version; R = revision).
P6010@ ????? 2001	Displays the firmware revision date.
P6010# B660b 01502	Displays the ReFlash boot loader revision number (format = Bxxx.
P6010$ B1103 2000b	Displays the ReFlash boot loader revision date (format = Bmmddyyyx where mm = month; dd = day; yyyy = year; x = daily revision).
P6020) B SN: 12345	Displays the serial number of the main PC board.
P6020@ I SN: 12345	Displays the serial number of the indicator.

AUDIT TRAIL PARAMETERS

Several audit trail parameters are used to satisfy the requirements of various weights & measures agencies for sealing an instrument in legal-for-trade applications. These are non-resettable parameters that increment each time certain information is changed within the setup mode. Table 14-4 shows the different audit trail parameters. Refer to chapter 5 for more information on legal-for-trade applications.

Table 14-4: Audit Trail Parameters

AUDIT TRAIL PARAMETER	DESCRIPTION
P6020! OIML 00001	Displays the OIML audit trail number.
P6020# Cal. 00001	Displays the calibration audit trail number. This number increments each time a new calibration is saved.
P6020$ Setup 00001	Displays the setup audit trail number. This number increments each time setup changes are saved when exiting the setup mode.
P6020% MUST! Check	Displays a list of parameters which, if configured improperly, could facilitate fraud in a legal-for-trade installation. A weights & measures inspector may check this parameter and inquire about the setup of any parameter that appears in this list.

CALIBRATION PARAMETERS

The calibration parameters contain data internally calculated as the result of the load cell, analog, and A/D converter calibration. These values are unique to each instrument and should be recorded so they can be restored in the event they are inadvertently deleted from the setup. **With few exceptions, calibration parameter values described in** Table 14-5 **should never be changed from the original values calculated during initial calibration.**

Table 14-5: Calibration Parameters

CALIBRATION PARAMETER	DESCRIPTION
Calibration Parameters	
P6109(Scale 1	Used to enter the scale number for P61110 → P61121. For example, key in **2 [ENTER]**, then access P61110 → P61121 to view calibration data for scale #2.
P6110) Crrnt mv/v	Displays the current mV/V output of the load cell. The range is –5.0 mV/V to +5 mV/V with an accuracy of 0.02%. For example, a 100 lb capacity load cell rated at 3.00 mV/V at full scale should display 1.5 mV/V with 50 lb applied.
P6110! CAL Factr	Displays the calibration factor calculated during the last load cell calibration. This parameter does not appear with linearization enabled (P119). Refer instead to P61130 → P61139 in this table for information on linearization values.
P6110@ ReZro Wght	Displays the amount of weight (in default units per P150) that has been zeroed through use of the **[ZERO]** key since the last calibration.
P6110# ZrTrk Wght	Displays the amount of weight (in default units per P150) automatically tracked off by zero tracking since the **[ZERO]** key was last pressed.
P6110$ CZero 0%	Displays the coarse zero value calculated during the last load cell calibration.

CALIBRATION PARAMETER	DESCRIPTION
P6110% Fine Zero	Displays the fine zero value calculated during the last load cell calibration.
P6110^ CGain 50	Displays the coarse gain value calculated during the last load cell calibration.
P6110& Fine Gain	Displays the fine gain value calculated during the last load cell calibration. This parameter does not appear with linearization enabled (P119). Refer instead to P61130 → P61139 in this table for information on linearization values.
Parameters P61110 → P61121 refer to factory calibrated A/D values.	
P6111) Zero Adj25	Coarse zero adjustment verifications (range = ±209,715 for Adj25; ±419,430 for Adj50; ±838,860 for Ad100). These values must be entered (as provided) when installing a multi-scale option to ensure stability and linearity of the A/D converter. Failure to enter these values will result in a **ƒ A/D Cal** error message each time you exit the setup mode.
P6111! Zero Adj50	
P6111@ Zero Ad100	
P6111# Gain Adj1	Coarse gain compensation (range = .89969444 → 1.0145491). These values must be entered (as provided) when installing a multi-scale option to ensure stability and linearity of the A/D converter. Failure to enter these values will result in a Code39 ƒ A/D Cal error message each time you exit the setup mode.
P6111$ Gain Adj2	
P6111% Gain Adj4	
P6111^ Gain Adj8	
P6111& AIN1 NROff	Gain factor zero offset compensation (range = ±8,388,608). These values must be entered (as provided) when installing a multi-scale option to ensure stability and linearity of the A/D converter. Failure to enter these values will result in a Code39 ƒ A/D Cal error message each time you exit the setup mode.
P6111* AIN2 NROff	
P6111(AIN4 NROff	
P6112) AIN8 NROff	
P6112! VRER NROff	Zero offset compensation of A/D reference voltage derived by the sense leads (range = ±8,388,608).
P6112@ SN: 12345	Displays the scale's serial number. This number should be entered at the time of multi-scale installation for future reference (range = 0 → 999,999).
P6112# BckUp GOOD!	Shows whether A2D calibration for a given scale has a backup. It will show BckUp and report Good! if a backup for the scale selected at 61099 is in flash, None! if there is no copy in flash, Err 1 if a checksum error has occurred for the data in flash, Err 2 if the size of the data in the flash block is wrong, or Full if the flash has no further space for storing A2D cal data (74 calibrations).
Linearization must be enabled at P119 for access to P61130 → P61139.	
P6113) -CAL- WGHT1	Displays the weight (in default units per P150) used for the 1st linearization calibration point.
P6113! -CAL- FACT1	Displays the calibration factor for weights within the range of the 1st linearization calibration point.
P6113@ -CAL- WGHT2	Displays the weight (in default units per P150) used for the 2nd linearization calibration point.
P6113# -CAL- FACT2	Displays the calibration factor for weights within the range of the 2nd linearization calibration point.
P6113$ -CAL- WGHT3	Displays the weight (in default units per P150) used for the 3rd linearization calibration point.
P6113% -CAL- FACT3	Displays the calibration factor for weights within the range of the 3rd linearization calibration point.
P6113^ -CAL- WGHT4	Displays the weight (in default units per P150) used for the 4th linearization calibration point.

Calibration Parameter	Description
P6113& -CAL- FACT4	Displays the calibration factor for weights within the range of the 4th linearization calibration point.
P6113* -CAL- WGHT5	Displays the weight (in default units per P150) used for the 5th linearization calibration point.
P6113(-CAL- FACT5	Displays the calibration factor for weights within the range of the 5th linearization calibration point.
Parameters 61150 → 61160 refer to known load cell calibration values.	
P6115) #0fLC 0	Displays the number of load cells entered during the last "known load cell output" calibration. This value is reset to zero (0) if the "known load cell output" was not the last calibration method used.
P6115! LC #1 FSmVv	Displays the full scale mV/V output entered for load cell #1.
P6115@ LC #2 FSmVv	Displays the full scale mV/V output entered for load cell #2 (if specified).
P6115# LC #3 FSmVv	Displays the full scale mV/V output entered for load cell #3 (if specified).
P6115$ LC #4 FSmVv	Displays the full scale mV/V output entered for load cell #4 (if specified).
P6115% LC #5 FSmVv	Displays the full scale mV/V output entered for load cell #5 (if specified).
P6115^ LC #6 FSmVv	Displays the full scale mV/V output entered for load cell #6 (if specified).
P6115& LC #7 FSmVv	Displays the full scale mV/V output entered for load cell #7 (if specified).
P6115* LC #8 FSmVv	Displays the full scale mV/V output entered for load cell #8 (if specified).
P6115(LC FS 100.0	Displays the full scale capacity entered for the individual load cells.
P6116) Avg. LCmVv	Displays the calculated average full scale mV/V for all values entered in P61151 → P61158.
Parameters 61200 → 61207 refer to analog output calibration values.	
P6120) AnOut 1	Used to enter the analog output number for P61201 → P61207. For example, key in **2 [ENTER]**, then access P61201 → P61207 to view calibration data for analog output #2.
P6120! V Z — xxxxx	Sets the zero reference for the 0-10VDC analog output. Enter the value that results in exactly 0.00VDC output (range is 0 → 15000).
P6120@ V G — xxxxx	Sets the 10V reference for the 0-10VDC analog output. Enter the value that results in exactly 10.00VDC output (range is 50000→ 65535).
P6120# 0mAZ— xxxxx	Sets the zero reference for the 0-20mA analog output. Enter the value that results in exactly 0.00mA output (range is 0 → 15000).
P6120$ 0mAG— xxxxx	Sets the 20mA reference for the 0-20mA analog output. Enter the value that results in exactly 20.00mA output (range is 50000→ 65535).
P6120% 4mAZ— xxxxx	Sets the zero reference for the 4-20mA analog output. Enter the value that results in exactly 4.00mA output (range is 0 → 15000).
P6120^ 4mAG— xxxxx	Sets the 20mA reference for the 0-20mA analog output. Enter the value that results in exactly 20.00mA output (range is 50000→ 65535).
P6120& SN : 12345	Displays the analog output's serial number. This number should be entered at the time of installation for future reference.

DIAGNOSTIC PARAMETERS

The parameters listed in Table 14-6 provide valuable information regarding the integrity of system memory and parameter setup file transfers.

Table 14-6: Diagnostic Parameters

DIAGNOSTIC PARAMETER	DESCRIPTION
P6200) Dsply Test	Performs display test. Press **[ENTER]** to illuminate all segments and pixels of the 7-segment VF display. Press any key to exit display test mode.
P6200! OpRAM Test	Tests the integrity of the optional RAM (database). Press **[ENTER]** to initiate test. On the 460 Series, this procedure tests the 4K EEPROM database memory.
P6200@ EEROM Test	Tests the integrity of the E^2 (U27 & U28 on the main board). Press **[ENTER]** to initiate test.
P6201) AnOut 1	Used to enter the analog output number for P62011 → P62013. For example, key in **2 [ENTER]**, then access P62011 → P62013 to perform tests for analog output #2.
P6201! 0-10— 0.00v	Used to test the linearity of the 0-10VDC analog output. Press **[ENTER]** to toggle the analog output through 0V, 2V, 4V, 6V, 8V and 10V, or key in a value within the output range and press **[ENTER]**.
P6201@ 0-20— 0.00A	Used to test the linearity of the 0-20mA analog output. Press **[ENTER]** to toggle the analog output through 0mA, 5mA, 10mA, 15mA, and 20mA, or key in a value within the output range and press **[ENTER]**.
P6201# 4-20— 4.00A	Used to test the linearity of the 4-20mA analog output. Press **[ENTER]** to toggle the analog output through 4mA, 8mA, 12mA, 16mA, and 20mA, or key in a value within the output range and press **[ENTER]**.
P6400) Send Setup	Transmits all setup parameters out a specified communication port. Data is transmitted in ASCII text format.
P6400! Send All	Same as P64000 with the addition of operating parameter values such as variables, totals, tare weight, etc.
P6410) LnCnt 0	Displays the number of lines received while uploading a setup file. The line count is reset to zero (0) whenever the setup access code is received.
P6410! ErCnt 0	Displays the number of errors encountered while uploading a setup file.
P6410@ 1stEr None!	Displays the parameter number and line number of the first error encountered while uploading a setup file.
P6410# Debug Off	Enables/disables the transmission of each error encountered while uploading a setup file.

All database information will be erased when performing the Optional RAM test.

Example:
Optional RAM Test

[6] [2] [0] [0] [1] [SELECT]

P6200! OpRAM Test

[ENTER YES]

P6200! Erase OpRAM

P6200! Sure? ???

[ENTER YES]

P6200! OpRAM Good!

OPTIONAL RAM TEST

The Optional RAM test at P62001 checks the integrity of database memory by writing and reading test values to database RAM. All database RAM will be erased! Back-up all vital database information before initiating this test (refer to the *Download Database* command on page 9-141).

The example – *Optional RAM Test* shows how to perform the Optional RAM test. After pressing **[ENTER]** to erase RAM, a series of test patterns are displayed. The test may last up to several minutes depending on the size of the database. When complete, the display will show either OpRAM Good! indicating the test was successful, or OpRAM Bad indicating a problem with one or more memory locations. Press any key to exit the test mode.

If the Optional RAM tests bad, make sure that the database option is properly installed and that the battery voltage (P60018) is good before re-testing.

460 Series Database Memory

On the 460 this procedure tests the 4K portion of the EEPROM that is reserved for the database. Note that the error messages and displays are that of EEPROM (i.e. not op ram) This means that you will see OpRam test, then Erase OpRam, Sure????. You will not see Pass=X and pttrn XXXX during testing the actual test. You will see Pass= then a rapidly changing address (the actual relative address: i.e. for 460 from 4000 to 7FFF on the upper character portion and the pattern being used on the lower portion. The completed message OpRam Good! is the same. The error messages are different from a product using a database board (option ram). The error messages are the same as for EEPROM test except that "EEerrPatrn" becomes "OpErrPatrn", "EEerrWrite" becomes "OpErrWrite", "EEerr Read" becomes "OpErr Read", and "Time Out" remains the same.

EEPROM TEST

The EEPROM test at P62002 checks the integrity of the E^2 by writing and reading test values to U27 & U28 on the main board. It is important to keep the unit powered up during the EEPROM test since the setup parameter values only exist in volatile RAM while test patterns over-write the setup in E^2.

Example 13-7 shows how to perform the EEPROM test. After pressing **[ENTER]** to begin, a series of test patterns are displayed. The test may last up to several minutes depending on the amount of E^2 installed. When complete, the display will show either EEROM Good! indicating the test was successful, or EEROM Bad indicating a problem with one or more memory locations. Press any key to exit the test mode.

If the E^2 test fails, make sure that U27 & U28 are properly installed before re-testing. A bad EEPROM may be the result of electrostatic discharge from improper handling or degraded memory due to continual writing at one location. The E^2 in used in the 60 Series instruments is rated for 100,000 writes. Care should be taken when programming to avoid a situation where values are constantly written to E^2, such as continuously copying the gross weight to a variable that is configured for auto-save.

ANALOG OUTPUT TEST

The Analog Output test at P62010 → P62013 checks the linearity of each analog output.

The example *Analog Output Test* shows how to perform the analog output test for a 0-20mA output. Each time **[ENTER]** is pressed, the analog output increases as displayed and can be measured directly to verify output linearity. Once the upper limit of the analog output range is reached, pressing **[ENTER]** again will restart the test at the minimum output level.

Example:
Analog Output Test

[6] [2] [0] [1] [3] [·] [2] [SELECT]

P6201@ 0-20~ 0.00v

P6201@ Enter ToTst

[ENTER YES]

P6201@ 0-20~ 0.00A

Analog output = 0.00mA

[ENTER YES]

P6201@ 0-20~ 5.00A

Analog output = 5.00mA

[ENTER YES]

P6201@ 0-20~ 10.0A

Analog output = 10.00mA

[ENTER YES]

P6201@ 0-20~ 15.0A

Analog output = 15.00mA

[ENTER YES]

P6201@ 0-20~ 20.0A

Analog output = 20.00mA

DOWNLOAD SETUP

P64000 allows you to transmit (download) the entire parameter to another device such as a printer or computer. If sent to a computer, the setup information can be saved as a text file. The file can then serve as a backup of the custom configuration that can later be used to restore the setup. It can also be used to copy the setup to other indicators (see the *Upload Setup* section below).

The example – *Download Setup* shows how to initiate the setup download. Verify that the protocol of the receiving device matches that of the indicator. It is also important to utilize software or hardware handshaking to prevent loss of data. Transmission begins immediately after selecting the communication port. Transfer is complete when the display returns to the download parameter.

P64001 transmits the same data as P64000, but also includes operating parameter values such as variables, totals, tare weight, etc., at the end of the transmission. Parameter values are downloaded in a format that allows them to be automatically restored during an upload.

Example:
Download Setup

[6] [4] [0] [0] [0] [SELECT]

P6400) Send Setup

[ENTER YES]

P6400) Enter Comm#

[1]

UPLOAD SETUP

"Uploading" refers to having an indicator receive a parameter setup file from a computer or another indicator.

A setup file may be uploaded to an indicator through any of the four communication ports. It is not necessary to be in the setup mode to begin uploading provided the access code appears at the beginning of the setup file. The indicator can receive a setup file from the weigh mode provided:

- P440 (NTEP) is disabled
- A macro is not executing
- The communication port receiving the setup is not disabled, set to invoke a macro, or set for input interpreter
- An entry is not in progress

Before uploading a setup file, verify that the protocol of the sending device matches that of the indicator. It is also important to utilize software or hardware handshaking to prevent data loss as indicated by an `ovrn1 error` message during upload.

The total line count for the upload file is registered at P64100. This parameter counts the number of carriage return characters received since the access code was last entered. Any errors encountered during the upload process are registered in several ways:

- The first error to occur will generate a two-tone beep and will be registered at P64102. This parameter should be accessed after every upload *before* saving and exiting the new setup to verify that no errors occurred. If an error did occur, P64102 will alternate between the effected parameter number and the setup file line number at which the error was encountered. This error flag is cleared each time the setup access code is entered or by pressing **[CLR]** at P64102.
- The total error count is registered at P60101.
- Upload errors can be transmitted back to the sending device as they occur by enabling the upload debug at P60103.

UTILITY PARAMETERS

The utility parameters allow you to restore setup parameters to factory default values. Refer to Table 14-7 for details on the utility parameters.

Table 14-7: Utility Parameters

UTILITY PARAMETER	DESCRIPTION
P6500) Copy A->B?	Copies the contents of the U27 EEPROM to the U28 EEPROM. Press **[ENTER]** to initiate copy. This does not apply to the 460 Series indicators.
P6500! Deflt All	Defaults **ALL** setup parameters to factory default values. † Press **[ENTER]** to initiate default.
P6500@ Deflt -CAL	Defaults all setup parameters to factory default values EXCEPT scale setup parameters P109 → P136. Press **[ENTER]** to initiate default.
P6501) Dbase Reset	Resets the database memory. Press **[ENTER]** to initiate reset. **ALL** data residing in the database will be lost.
P6502) New Prog?	Initiates the ReFlash process. Pressing **[ENTER]** will blank the indicator until the ReFlash process is complete. Requires GSE ReFlash software!

† Selections for Scale Enable/Save/Disable (P109), 4X20 Display Enable/Disable (P425), Keypad Type (P450), Serial Numbers & Audit Trails are not affected by defaulting.

COPY EEPROM

The EEPROM copy procedure should be performed whenever upgrading or replacing the U27 E^2 on the main board. This is done primarily to transfer serial numbers and audit trail parameters to the new E^2, although *all* data in U27 will be copied to U28. Serial numbers and audit trail information are erased from the original E^2 in U27 after the copy is complete.

DEFAULT SETUP

Original factory parameter configuration can be restored by defaulting an indicator at P65001. Except for the parameters noted in Table 14-7, **ALL SETUP PARAMETERS WILL BE RESET TO THE FACTORY DEFAULT! LOAD CELL CALIBRATION, MULTI-SCALE A/D CALIBRATION AND DATABASE INFORMATION WILL BE LOST!**

The example – *Default All* shows the Default All procedure. The prompt to Clr ? AllDB will only be displayed if a database option is installed and

Example:
Default All

6 5 0 0 1 SELECT

P6500! Deflt All

ENTER YES

P6500! Sure? ???

P6500! Enter =Dflt

ENTER YES

P6500! Clr ? AllDB

P6500! Enter ToCLR

P6500! Enter =Dflt

ENTER YES

P6500! Done!

initialized. A newly installed database option that has not been initialized will be initialized after defaulting.

While defaulting, the display will blank for several seconds, then return to P65001. Saving changes will then overwrite the previous setup with factory default values. If changes are not saved before exiting the setup, the previous setup is retained.

The Default –CAL at P65002 is similar to P65001, except that all scale setup parameters (P109 → P136) are left unchanged after defaulting. Thus it is not necessary to recalibrate the load cell or reinstall the multi-scale A/D values.

DATABASE RESET

A newly installed database must be initialized before it is recognized by the indicator. P65010 can be used to initialize a database option without having to default. An uninitialized database is evident by the message `NewDB Opt'n` after saving and exiting the setup mode. The example – *Database Reset* shows the Database Reset procedure.

For a database option that has already been initialized, Database Reset will clear the database memory. If Setup RAM is allocated at P60040, the parameter setup stored in the database option will not be erased by the Database Reset. If Pile RAM is allocated at P60050, it will not be possible to reset the database without defaulting. The message `See P65001` will direct you to the Default All parameter.

Example:
Database Reset

6 5 0 1 0 SELECT

P6501) dbase Reset

ENTER YES

P6501) Erase DBRAM

P6200@ Sure? ???

ENTER YES

P6200@ Done!

Appendix A Specifications

This appendix provides specifications for all 60 Series instruments.

PERFORMANCE

	460 SERIES	560 SERIES	660 SERIES
Resolution	100,000 displayed (±500,000 internal)	100,000 displayed (±500,000 internal)	100,000 displayed (±500,000 internal)
A/D Conversion Rate	60 Hz	60 Hz	60 Hz
Flash Memory	512K bytes	1MB	1MB
Macros	15 macro tables	100 macro tables	250 macro tables
Database	15 database tables (4K max)	100 database tables	250 database tables
Non-Linearity	0.005% of full scale (input dependent)	0.005% of full scale (input dependent)	0.005% of full scale (input dependent)
Calibration	Selectable; 5 point linearization or enter loadcell mV/V @ F.S.	Selectable; 5 point linearization or enter loadcell mV/V @ F.S.	Selectable; 5 point linearization or enter loadcell mV/V @ F.S.
A/D Filtering	GSE FIR (Finite Input Response) Filter with selectable display update rate	GSE FIR (Finite Input Response) Filter with selectable display update rate	GSE FIR (Finite Input Response) Filter with selectable display update rate
Units of Measure	Pounds, Kilograms, Ounces, Grams, Tons, Metric Tons, Pounds-Ounces, Two (2) Custom Units	Pounds, Kilograms, Ounces, Grams, Tons, Metric Tons, Pounds-Ounces, Two (2) Custom Units	Pounds, Kilograms, Ounces, Grams, Tons, Metric Tons, Pounds-Ounces, Two (2) Custom Units
Zero Adjustment	Selectable, 0.01 – 100% F.S.	Selectable, 0.01 – 100% F.S.	Selectable, 0.01 – 100% F.S.
Span Adjustment	0.1 – 20.0 mV/V	0.1 – 20.0 mV/V	0.1 – 20.0 mV/V
Non-Linearity	0.005% F.S., Loadcell dependent	0.005% F.S., Loadcell dependent	0.005% F.S., Loadcell dependent
Operating Temperature	-10°C to 40°C	-10°C to 40°C	-10°C to 40°C
Multi-Scale Capability	Up to 2 scale input modules	Up to 4 scale input modules	Up to 8 scale input modules
Loadcell Excitation Power	12 – 350 ohm bridges	14 – 350 ohm bridges	14 – 350 ohm bridges
Input Signal Connect	4 or 6 conductor with sense	4 or 6 conductor with sense	4 or 6 conductor with sense
Excitation Voltage	10VDC, Short circuit protected	10VDC, Short circuit protected	10VDC, Short circuit protected
Excitation Current	350mA	400mA	400mA
Time/Date Clock	Volatile, battery backed optional	Non-Volatile, battery backed standard	Non-Volatile, battery backed standard
Warranty	2 year covering defective parts and workmanship	2 year covering defective parts and workmanship	2 year covering defective parts and workmanship

ELECTRICAL

	460 SERIES	560 SERIES	660 SERIES
Power Input	90-250 VAC, 50/60 Hz 10-32VDC	90-250 VAC, 50/60 Hz 10-32VDC	90-250 VAC, 50/60 Hz 10-32VDC
Fuse	0.5A Slow Blow	0.8A Slow Blow	0.8A Slow Blow

COMMUNICATIONS

	460 SERIES	560 SERIES	660 SERIES
Port 1	Bi-directional RS-232 Serial (Xon/Xoff & CTS/RTS hardware handshaking) RS-485 multi-drop (optional)	Bi-directional RS-232 Serial (Xon/Xoff & CTS/RTS hardware handshaking) RS-485 multi-drop (optional)	Bi-directional RS-232 Serial (Xon/Xoff & CTS/RTS hardware handshaking) RS-485 multi-drop (optional)
Port 2	Bi-directional RS-232 Serial (Xon/Xoff & CTS/RTS hardware handshaking)	Bi-directional RS-232 Serial (Xon/Xoff & CTS/RTS hardware handshaking)	Bi-directional RS-232 Serial (Xon/Xoff & CTS/RTS hardware handshaking)
Port 3	N/A	Bi-directional RS-232 Serial (Xon/Xoff & CTS hardware handshaking)	Bi-directional RS-232 Serial (Xon/Xoff & CTS hardware handshaking)
Port 4	N/A	N/A	Bi-directional TTL or RS-232 (Xon/Xoff handshaking)
Data Protocol	Data bits 7-8; stop bits 1-2; even, odd or no parity	Data bits 7-8; stop bits 1-2; even, odd or no parity	Data bits 7-8; stop bits 1-2; even, odd or no parity
Baud Rate	Selectable: 150-58300 baud	Selectable: 150-58300 baud	Selectable: 150-115K baud
Communication Protocol	Modbus™	Modbus™	Modbus™
Custom Transmit Tables	4 user defined; continuous, prompt or on-request	100 user defined; continuous, prompt or on-request	250 user defined; continuous, prompt or on-request
Input Interpreter Tables	15 user defined input tables	100 user defined input tables	250 user defined input tables

APPROVALS

	460 SERIES	560 SERIES	660 SERIES
Metrological	NTEP COC#01-031, class III & IIIL @ 10,000d, Canadian W&M pending, class IIIHD @ 25,000	NTEP COC#01-081, class III & IIIL @ 10,000d, Canadian W&M pending, class IIIHD @ 25,000	NTEP COC# 01-013, class III & IIIL @ 10,000d, Canadian W&M pending, class IIIHD @ 25,000
Electrical	ETL	UL, CSA	UL, CSA

INPUT/OUTPUT

	460 SERIES	560 SERIES	660 SERIES
I/O Scan Rate	16ms (max)	16ms (max)	16ms (max)
Number of I/O	8 maximum physical, optional 16 total (8 physical, 8 logical)	32 maximum physical, optional 48 total (32 physical, 16 logical)	128 maximum physical, optional 256 total (128 physical, 128 logical)
Output Voltage Rating	20-240VAC @ 1A 3-60VDC @ 2A	20-240VAC @ 1A 3-60VDC @ 2A	20-240VAC @ 1A 3-60VDC @ 2A
Input Voltage Rating	120VAC, 230VAC, 3-30VDC	120VAC, 230VAC, 3-30VDC	120VAC, 230VAC, 3-30VDC
I/O Timer Resolution	10 ms	10 ms	10 ms
Program Timer Resolution	2 ms	2 ms	2 ms
PDIO Frequency Range	N/A	N/A	8 Hz – 100 KHz (5VDC TTL)
Programmable Digital I/O	2 remote key (contact closure) inputs	2 remote key (contact closure) inputs	8 independent channels with selectable functions; frequency input, frequency output, phase time, delay in/delay out, quadrature input (2 or 3 wire), frequency debounce

ENCLOSURE

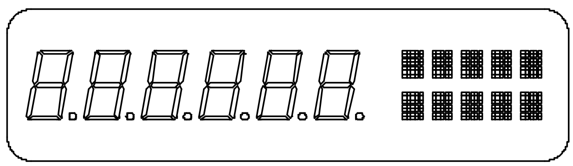
MODEL	ENCLOSURE
460	Stainless steel, NEMA 4X design (IP66); Shipping weight 8lb/ 4kg Available in stainless steel panel mount design
465, 560, 562, 660, 661, 662	Stainless steel, NEMA 4X design (IP66); Shipping weight 8lb/ 4kg All available in stainless steel panel mount design
M663	Mild steel, powder coat NEMA 12X design; Shipping weight 44lb/20kg Available in stainless steel, NEMA 4X design (IP66)
M663 Big Box	Mild steel, powder coat NEMA 12X design; Shipping weight 84lb/38kg Available in stainless steel, NEMA 4X design (IP66)
M665	Stainless steel NEMA 4X design (IP66) Available in stainless steel panel mount design
Switch Panel	All keypads are elastomeric silicone rubber, self-sealing, chemical resistant

DISPLAY

Display types differ according to model. The table below describes the display(s) offered for each controller.

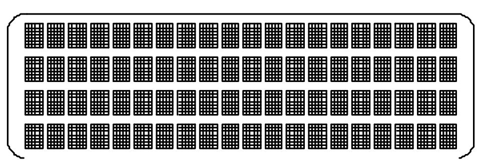
MODEL	DISPLAY
460, 465, 560, 660	Six-digit, 7-segment VF display (0.75"; 19mm high digits) with 2X5 prompting area
661	Four-line by 20-character alphanumeric VF display
562, 662	8 x 40 character backlit LCD display
663, 665	Six-digit, 7-segment VF display (0.75"; 19mm high digits) with 2X5 prompting AND 4x20 character VF display *-or-* 16x40 character backlit LCD display

6-DIGIT, 7-SEGMENT DISPLAY



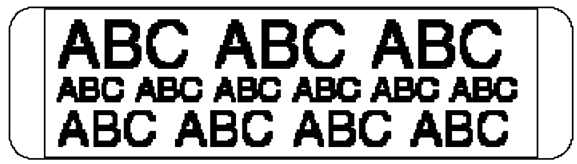
- Six fully active 7-segment digits
- 0.75" (19 mm) high digits
- 2-line-by-5-character alphanumeric dot matrix
- Vacuum Fluorescent

4 X 20 CHARACTER VFD



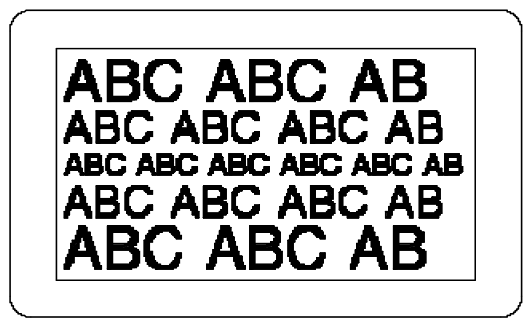
- 4-line-by-20-character alphanumeric dot matrix
- 0.20" (5 mm) high characters
- Vacuum Fluorescent

8 X 40 CHARACTER GRAPHIC LCD



- 8-line-by-40-character alphanumeric (small font size)
- 4-line-by-20-character alphanumeric (medium font size)
- 2-line-by-10-character alphanumeric (large font size)
- Graphic box and line drawing

16 X 40 CHARACTER GRAPHIC LCD



- 16-line-by-40-character alphanumeric (small font size)
- 8-line-by-20-character alphanumeric (medium font size)
- 4-line-by-10-character alphanumeric (large font size)
- Graphic box and line drawing

KEYPAD

All models have an elastomeric keypad. An international (OIML) keypad is also available for the 560 Series, 660 Series and 465 indicator (see page 6-2).

460 KEYPAD

The 460 keypad uses a 5-key front panel keypad and provides additional functions when pressing multiple keys simultaneously (see Table A-1).

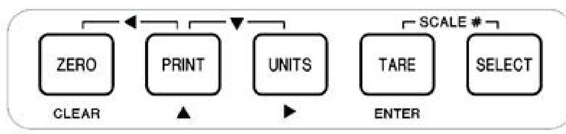


Table A-1: 460 Key Functions

KEYPRESS	GROSS	NET	QTY	OTHER WEIGH MODES	ACCESS SETUP	SETUP (GENERAL)
[ZERO / CLEAR]	ZERO	ZERO	ZERO	ZERO * CLEAR †	special case #1	ZERO * CLEAR †
[PRINT]	PRINT	PRINT	PRINT	?	special case #2	?
[UNITS]	UNITS	UNITS	UNITS	UNITS * ? †	special case #2	?
[TARE / ENTER]	TARE	TARE	TARE	ENTER	ENTER	ENTER
[SELECT]	SELECT	SELECT	SELECT	SELECT	special case #3	SELECT
Multiple Key Combinations						
[ZERO] + [SELECT]	Access setup	access setup	access setup	access setup	access setup	access setup
[ZERO] + [PRINT]	?	?	?	?	?	?
[ZERO] + [TARE]	CLEAR	CLEAR	CLEAR	CLEAR	CLEAR	CLEAR
[ZERO] + [UNITS]	N/A	N/A	N/A	N/A	N/A	Alpha Entry
[PRINT] + [UNITS]	?	?	?	?	?	?
[PRINT] + [SELECT]	ext. res. OIML	ext. res. OIML	ext. res. OIML	ext. res. OIML	N/A	ID
[TARE] + [SELECT]	Scale Select	Scale Select	Scale Select	Scale Select	N/A	Backup select
[UNITS] + [SELECT]	Accum. function	Accum. function	Accum. function	ENTER	ENTER	ENTER
[UNITS] + [TARE]	N/A	N/A	N/A	N/A	N/A	Next Instance
[PRINT] + [TARE]	N/A	N/A	N/A	N/A	N/A	Previous Instance
[ZERO] + [TARE] + [SELECT]	Abort macro menu	Abort macro menu	Abort macro menu	Abort macro menu	N/A	N/A

* = No entry in progress

† = Entry in progress

More specific key definitions while in setup (Work in progress)

- The **[UNITS]** key will execute the **[F2]** (M660) key while picking a parm or picking an instance to provide more feedback.
- To scroll through the available setup script files use the **[UNITS]** to increment and the ? (**[ZERO]** + **[PRINT]**) to decrement through the scripts located at P65001, P65002, P65003 (used to setup the remote keys and remote display).

M460 Special Cases

- Pressing the **[ZERO / CLEAR]** key during access setup (Keyin Code:):

  If an entry is started with either the **[SELECT]** or **[ZERO / CLEAR]** key then a lower case 'z' will be entered when **[ZERO / CLEAR]** is pressed. Otherwise, pressing this key will cause the entry in process to be cleared (i.e. to abort scrolling in an access code).

- Pressing the **[PRINT]** or **[UNITS]** key while accessing setup:

  If an access code entry is in progress that was started with either the **[ZERO / CLEAR]** or **[SELECT]** key then the units key will cause a lower case 'u' to be entered. However, if the access code entry is started with either the **[PRINT]** or **[UNITS]** key then the response will be to start scrolling in numeric or ASCII characters respectively.

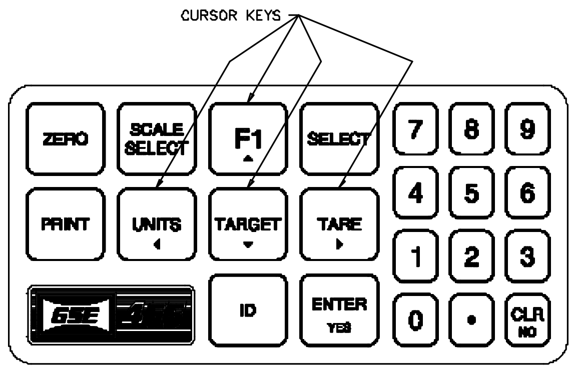
  If a numeric entry has been started with the **[PRINT]** key, the **[UNITS]** key will perform the ? function, appending a decimal '.'. If an ASCII character entry has been started with the **[UNITS]** key, the **[UNITS]** key will perform the ? function, appending the upper case 'A'.

- Pressing the **[SELECT]** key while accessing setup:

  If an entry is started with either the **[SELECT]** or **[ZERO / CLEAR]** key then a lower case 's' will be entered when **[SELECT]** is pressed. Otherwise the **[SELECT]** key will serve no purpose.

465 KEYPAD

The 465 uses a 22-key front panel keypad.



Special Cases

- The **[F1]** key invokes macro 1 or performs the up-arrow function.
- The **[TARGET]** key invokes macro 2 or performs the down-arrow function.
- The **[UNITS / ←]** combination key functions as follows:

  In the weigh mode the **[UNITS]** key will perform the units function. The only exception is at 78P (setpoints) where the ← function is executed which de-activates the viewed setpoint.

  If Accessing setup or in setup, the **[UNITS]** key will execute the ← function.
- The **[TARE/→]** combo keys function is defined below:

  In the weigh mode the **[TARE]** key will perform the tare function with two exceptions:

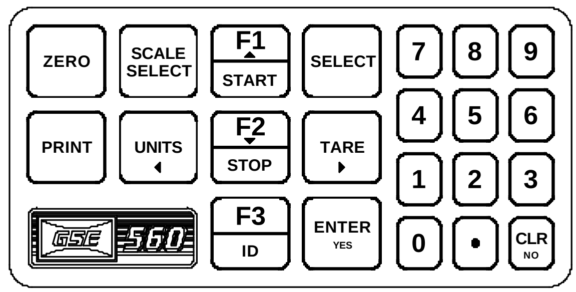
    If in quantity mode pressing the **[TARE]** key will perform a tare and then require that a new sample be performed

    At 78P (setpoints), pressing the **[TARE]** key will execute the right arrow function, activating the setpoint.

  If Accessing setup or in setup the **[TARE]** key will execute the → function (same as **[F5]** on a 660).
- While viewing a parameter having instances, all that is necessary to choose a specific instance is to enter **[.] [#]** followed by **[SELECT]**, where # is the instance of interest. Or simply enter **[.] [.] [SELECT]** to increment the instance. This change will apply to all 60 series products. For example, while viewing Var #2's name, entering **[.] [.] SELECT]** will change the mode to be that of viewing Var 3's name. Entering **[.] [1] [1] [SELECT]** will change the mode to be Var 11's name. Similarly, to decrement the instance number, you may enter **[.] [0] [SELECT]**.
- The **[ID]** key performs the standard ID function if enabled at P806.
- The **[ID]** key can be used to invoke the macro menu and/or database menu if configured at P806.
- The **[ID]** key can invoke macros 3 → 100 if configured at P806.

560 SERIES KEYPAD

All 560 Series controllers use the same 22-key front panel keypad.

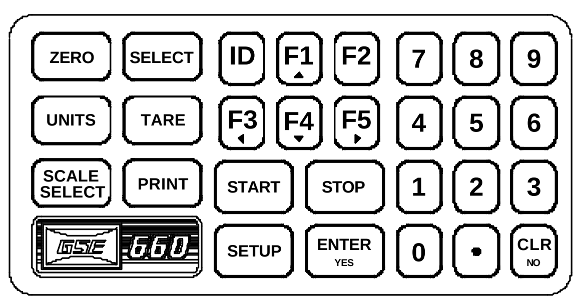


Special Cases

- The **[F1 / START]** key invokes macro 1 or performs the up-arrow function.
- The **[F2 / STOP]** key invokes macro 2 or performs the down-arrow function.
- The **[F3 / ID]** key performs the standard ID function if enabled at P806.
- The **[F3 / ID]** key can be used to invoke the macro menu and/or database menu if configured at P806.
- The **[F3 / ID]** key can invoke macros 3 → 100 if configured at P806.

660 SERIES KEYPAD

All 660 Series controllers use the same 28-key front panel keypad.



CONFIGURATIONS

MODEL	GSE PART NUMBER	OPERATING VOLTAGE	STANDARD DISPLAY	KEYPAD	LINE CORD
460 Stainless	200460-00000	90-250 VAC	VF 6 Digit	5-Key US	US
460 Stainless	200460-03100	90-250 VAC	VF 6 Digit	5-Key OMIL	UK
460 Stainless	200460-04100	90-250 VAC	VF 6 Digit	5-Key OMIL	Euro
460 Panel Mount	200460-19000	90-250 VAC	VF 6 Digit	5-Key US	None
465 Stainless	200465-00000	90-250 VAC	VF 6 Digit	22-Key US	US
465 Stainless	200465-03100	90-250 VAC	VF 6 Digit	22-Key OMIL	UK
465 Stainless	200465-04100	90-250 VAC	VF 6 Digit	22-Key OMIL	Euro
465 Panel Mount	200465-19000	90-250 VAC	VF 6 Digit	22-Key US	None
560 Stainless	200560-00000	90-250 VAC	VF 6 Digit	22-Key US	US
560 Stainless	200560-03100	90-250 VAC	VF 6 Digit	22-Key OMIL	UK
560 Stainless	200560-04100	90-250 VAC	VF 6 Digit	22-Key OMIL	Euro
560 Panel Mount	200560-19000	90-250 VAC	VF 6 Digit	22-Key US	None
562 Stainless	200562-00020	90-250 VAC	LCD 240X64	22-Key US	US
562 Stainless	200562-03100	90-250 VAC	LCD 240X64	22-Key OMIL	UK
562 Stainless	200562-04100	90-250 VAC	LCD 240X64	22-Key OMIL	Euro
562 Panel Mount	200562-19020	90-250 VAC	LCD 240X64	22-Key US	None
660 Stainless	200660-00000	90-250 VAC	VF 6 Digit	28-Key US	US
660 Stainless	200660-03100	90-250 VAC	VF 6 Digit	28-Key OMIL	UK
660 Stainless	200660-04100	90-250 VAC	VF 6 Digit	28-Key OMIL	Euro
660 Panel Mount	200660-19000	90-250 VAC	VF 6 Digit	28-Key US	None
661 Stainless	200661-00010	90-250 VAC	VF 4X20	28-Key US	US
661 Stainless	200661-03100	90-250 VAC	VF 4X20	28-Key OMIL	UK
661 Stainless	200661-04100	90-250 VAC	VF 4X20	28-Key OMIL	Euro
661 Panel Mount	200661-19010	90-250 VAC	VF 4X20	28-Key US	None
662 Stainless	200662-00020	90-250 VAC	LCD 240X64	28-Key US	US
662 Stainless	200662-03100	90-250 VAC	LCD 240X64	28-Key OMIL	UK
662 Stainless	200662-04100	90-250 VAC	LCD 240X64	28-Key OMIL	Euro
662 Panel Mount	200662-19020	90-250 VAC	LCD 240X64	28-Key US	None
663 Stainless	200663-09000	90-250 VAC	VF 6 Digit & 4X20	28-Key US	None
663 Stainless	200663-09020	90-250 VAC	LCD 240X128 Backlit	28-Key US	None
663 Painted	200663-29000	90-250 VAC	VF 6 Digit & 4X20	28-Key US	None
663 Painted	200663-29020	90-250 VAC	LCD 240X128 Backlit	28-Key US	None
663 Painted Big Box	200663-39000	90-250 VAC	VF 6 Digit & 4X20	28-Key US	None
663 Painted Big Box	200663-39020	90-250 VAC	LCD 240X128 Backlit	28-Key US	None
665 Stainless	200665-00000	90-250 VAC	VF 6 Digit & 4X20	28-Key US	US
665 Stainless	200665-00020	90-250 VAC	LCD 240X128 Backlit	28-Key US	US
665 Stainless	200665-03100	90-250 VAC	VF 6 Digit & 4X20	28-Key OMIL	UK
665 Stainless	200665-03120	90-250 VAC	LCD 240X128 Backlit	28-Key OMIL	UK
665 Stainless	200665-04100	90-250 VAC	VF 6 Digit & 4X20	28-Key OMIL	Euro
665 Stainless	200665-04120	90-250 VAC	LCD 240X128 Backlit	28-Key OMIL	Euro
665 Panel Mount	200665-19000	90-250 VAC	VF 6 Digit & 4X20	28-Key US	None
665 Panel Mount	200665-19020	90-250 VAC	LCD 240X128 Backlit	28-Key US	None

DIMENSIONAL DRAWINGS

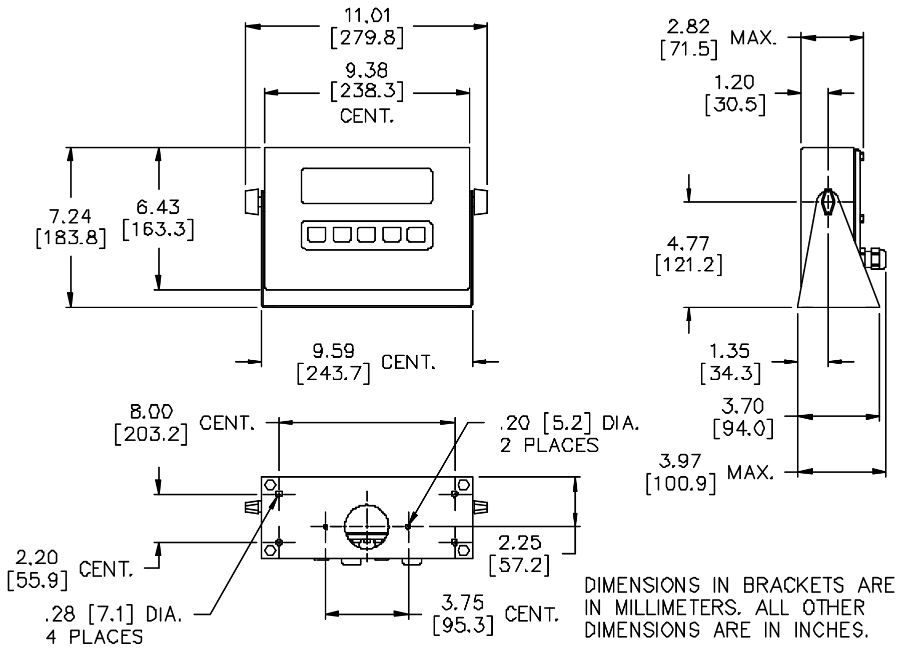


Figure 14-1: 460 Outline Drawing

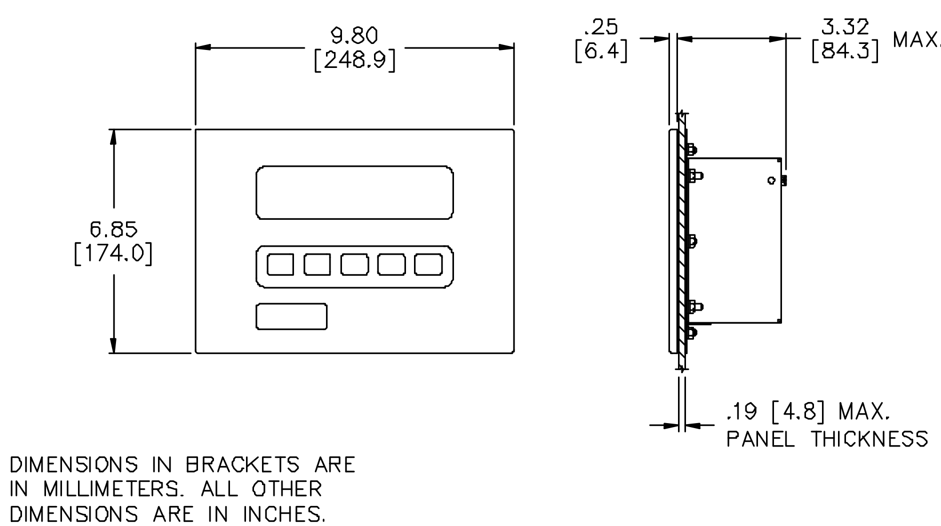


Figure 14-2: 460 Panel Mount Outline Drawing

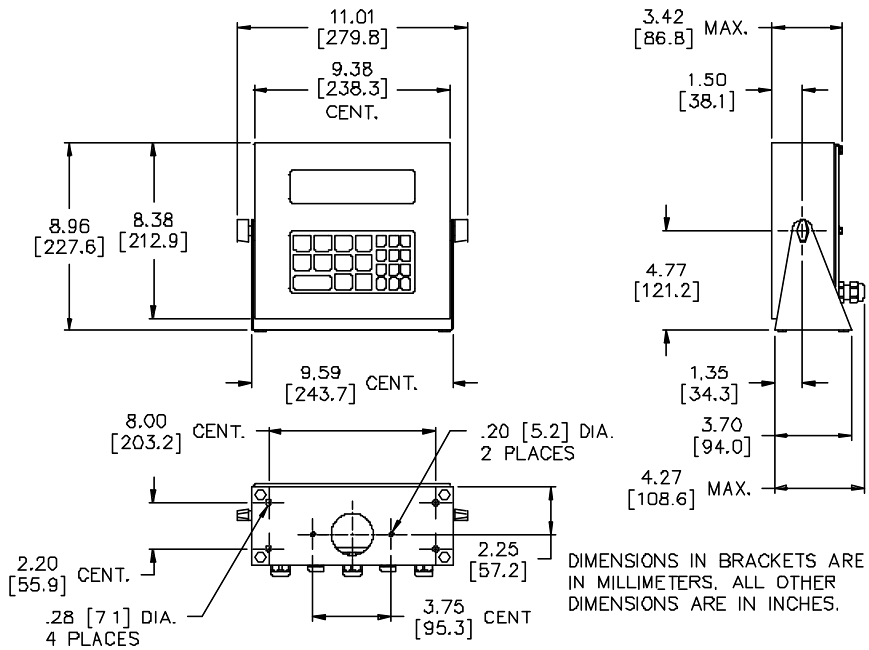


Figure 14-3: 465, 560, 562 Outline Drawing

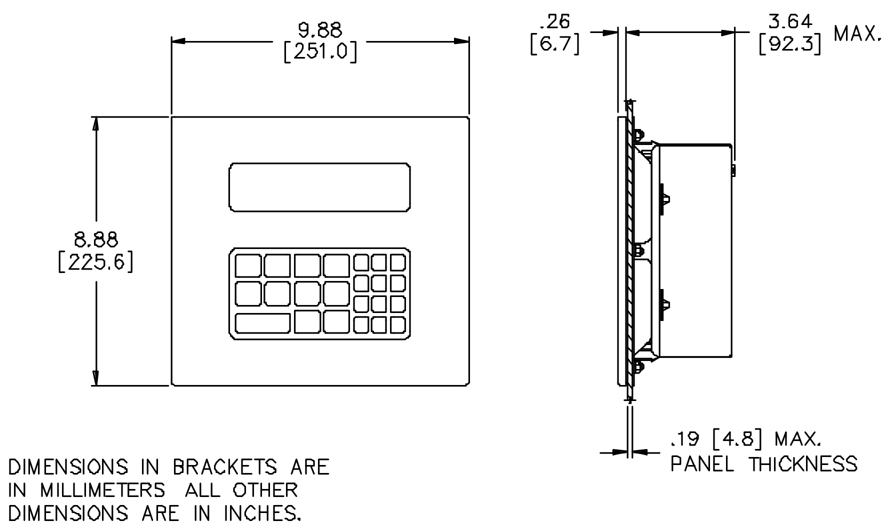


Figure 14-4: 465, 560, 562 Panel Mount Outline Drawing

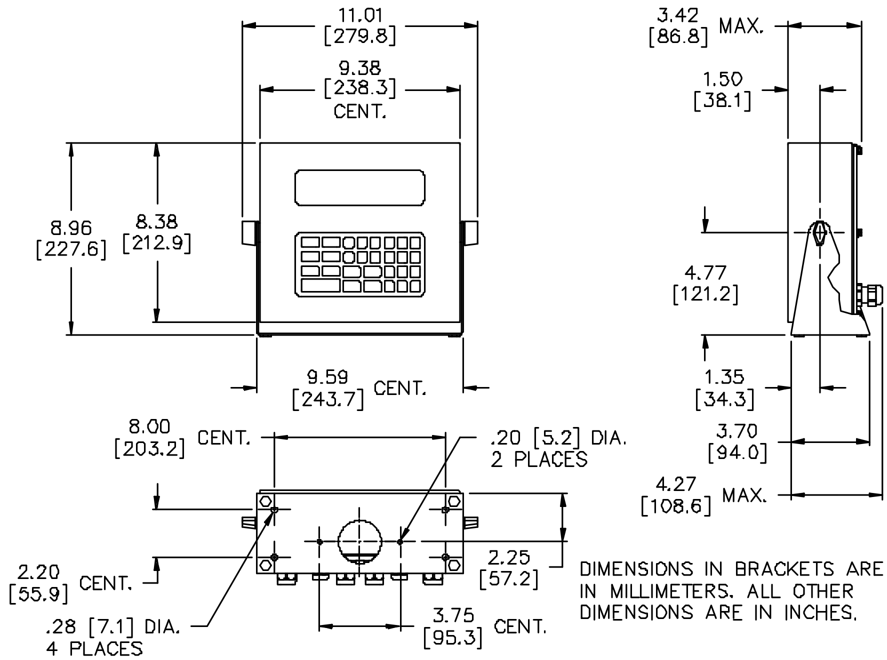


Figure 14-5: 660, 661, 662 Outline Drawing

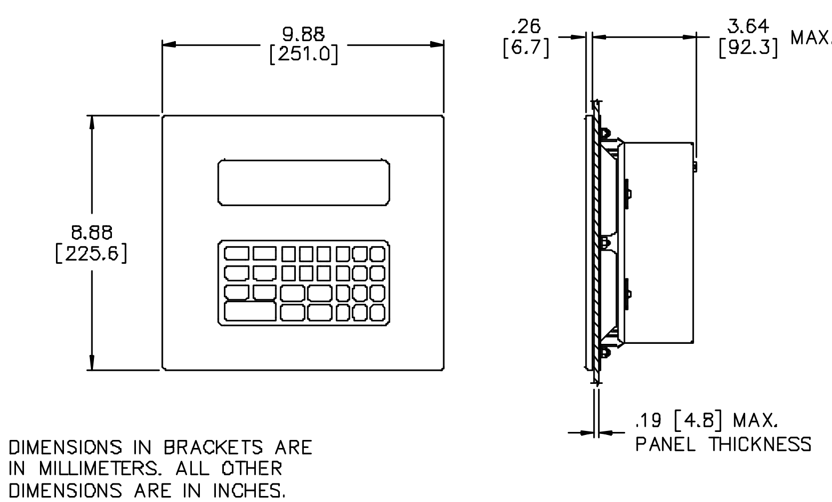


Figure 14-6: 660, 661, 662 Panel Mount Outline Drawing

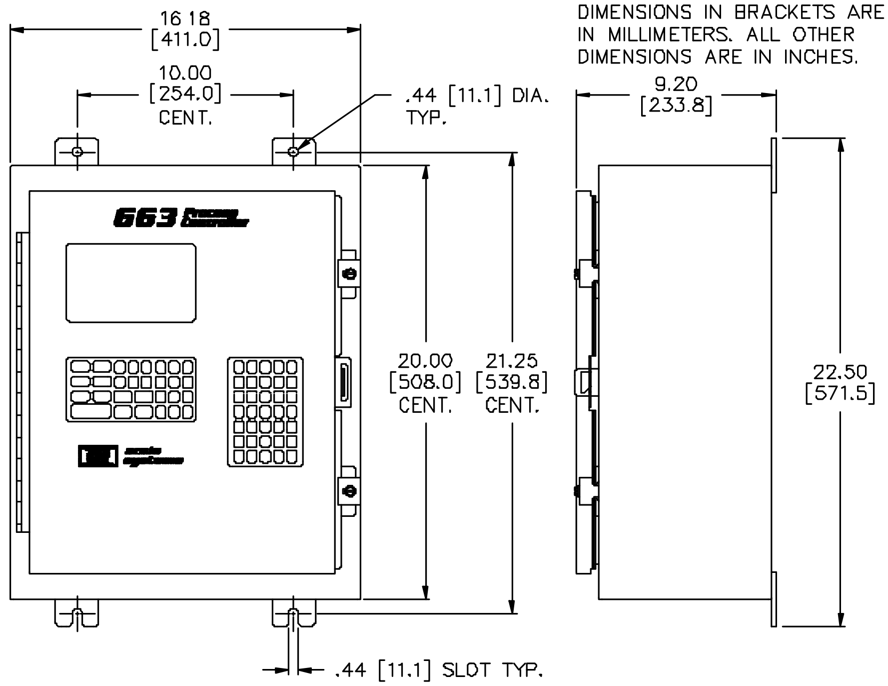


Figure 14-7: 663 Outline Drawing

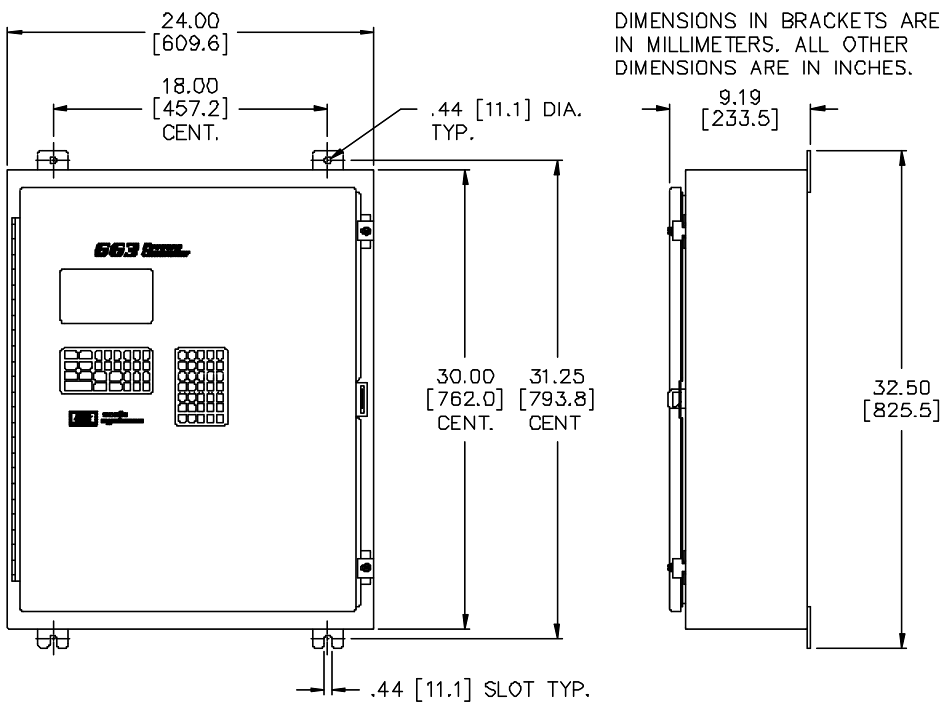


Figure 14-8: 663 Big Box Outline Drawing

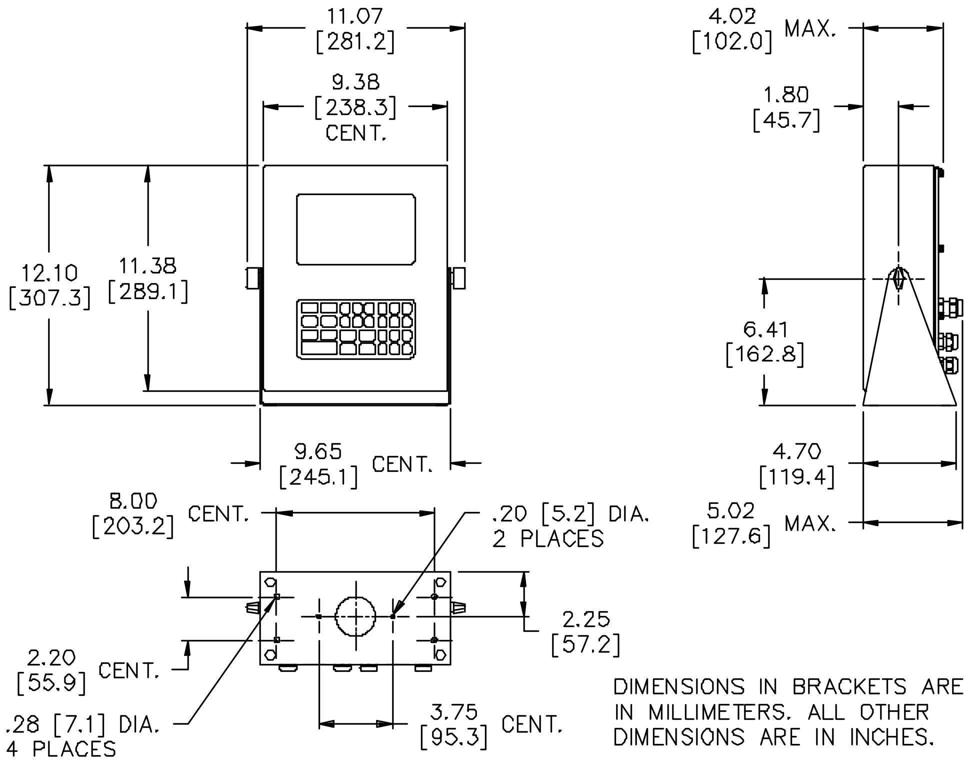


Figure 14-9: 665 Outline Drawing

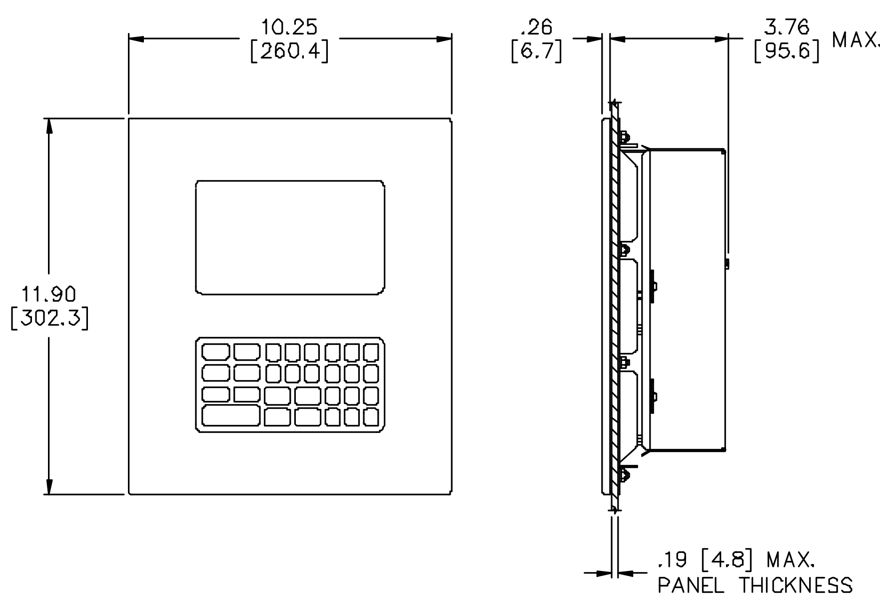


Figure 14-10: 665 Panel Mount Outline Drawing

Clock Module Specifications

The clock module is a real time clock/elapsed time counter. It is designed to count seconds when Vcc power is applied, and to count seconds continually under battery backup power with an additional counter regardless of the condition of Vcc. The continuous counter is used with a GSE software algorithm to derive time of day, week, month, and year.

The device contains a 32.768 kHz crystal that keeps track of time to within +/-2 min/mo. An internal lithium energy source contains enough energy to power the continuous seconds counter for more than 10 years.

Specification	Description
Accuracy	Self contained crystal provides an accuracy of +/-2 minutes per month at an *operating temperature* range between 0 and 70°C (absolute maximum). The module automatically corrects for leap year.
Battery Life	Minimum of 10 years with controller power off.
Storage Temperature	-40°C to 70°C (absolute maximum. (Clock exposure to absolute maximum rating conditions for extended periods of time may affect reliability.)
Soldering Temperature	260°C for 10 seconds (absolute maximum).
Any Pin to GND Voltage	The absolute maximum applied voltage on any pin relative to ground is -0.3V to +7.0V (absolute maximum).

FCC Compliance Information

This equipment has been tested and found to comply within the limits for a Class A digital device pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with this *Technical Reference Manual*, can cause harmful interference to radio communications. Operation of this equipment in a residential area may cause harmful interference, in which case the user will be required to correct the interference at his own expense.

CABLES: Shielded cables must be used with this equipment to ensure compliance with the Class A FCC limits.

Appendix B ASCII CHART

American Standard Code for Information Interchange (ASCII) Chart

Dec xHex Binary **Symbol** Name (Esc. Seq.)	**000** x00 0000 0000 **NUL** NUL (\0)	**001** x01 0000 0001 **SOH** Start Of Header	**002** x02 0000 0010 **STX** Start Of Text	**003** x03 0000 0011 **ETX** End Of Text	**004** x04 0000 0100 **EOT** End Of Transmission	**005** x05 0000 0101 **ENQ** Enquiry	**006** x06 0000 0110 **ACK** Acknowledge	**007** x07 0000 0111 **BEL** Bell (\a)	**008** x08 0000 1000 **BS** Back Space (\b)
009 x09 0000 1001 **HT** Horizontal Tab (\t)	**010** x0A 0000 1010 **LF** Line Feed (\n)	**011** x0B 0000 1011 **VT** Vertical Tab (\v)	**012** x0C 0000 1100 **FF** Form Feed (\f)	**013** x0D 0000 1101 **CR** Carriage Return (\r)	**014** x0E 0000 1110 **SO** Shift Out	**015** x0F 0000 1111 **SI** Shift In	**016** x10 0001 0000 **DLE** Data Link Escape	**017** x11 0001 0001 **DC1** Device Control 1 - XON	**018** x12 0001 0010 **DC2** Device Control 2
019 x13 0001 0011 **DC3** Device Control 3 - XOFF	**020** x14 0001 0100 **DC4** Device Control 4	**021** x15 0001 0101 **NAK** Negative Acknowledgement	**022** x16 0001 0110 **SYN** Synchronous Idle	**023** x17 0001 0111 **ETB** End Of Transmission Block	**024** x18 0001 1000 **CAN** Cancel	**025** x19 0001 1001 **EM** End Of Medium	**026** x1A 0001 1010 **SUB** Substitute	**027** x1B 0001 1011 **ESC** Escape	**028** x1C 0001 1100 **FS** File Separator
029 x1D 0001 1101 **GS** Group Separator	**030** x1E 0001 1110 **RS** Record Separator	**031** x1F 0001 1111 **US** Unit Separator	**032** x20 0010 0000 Space	**033** x21 0010 0001 **!** Exclamation	**034** x22 0010 0010 **"** Double Quote	**035** x23 0010 0011 **#** Number / Pound	**036** x24 0010 0100 **$** Dollar	**037** x25 0010 0101 **%** Percent	**038** x26 0010 0110 **&** Ampersand
039 x27 0010 0111 **'** Apostrophe/Single Quote	**040** x28 0010 1000 **(** Open Parenthesis	**041** x29 0010 1001 **)** Close Parenthesis	**042** x2A 0010 1010 * Asterisk	**043** x2B 0010 1011 **+** Plus	**044** x2C 0010 1100 **,** Comma	**045** x2D 0010 1101 **-** Dash/Minus/Hyphen	**046** x2E 0010 1110 **.** Period	**047** x2F 0010 1111 **/** Forward Slash	**048** x30 0011 0000 **0** Zero
049 x31 0011 0001 **1** One	**050** x32 0011 0010 **2** Two	**051** x33 0011 0011 **3** Three	**052** x34 0011 0100 **4** Four	**053** x35 0011 0101 **5** Five	**054** x36 0011 0110 **6** Six	**055** x37 0011 0111 **7** Seven	**056** x38 0011 1000 **8** Eight	**057** x39 0011 1001 **9** Nine	**058** x3A 0011 1010 **:** Colon
059 x3B 0011 1011 **;** Semicolon	**060** x3C 0011 1100 **<** Less Than	**061** x3D 0011 1101 **=** Equal	**062** x3E 0011 1110 **>** Greater Than	**063** x3F 0011 1111 **?** Question	**064** x40 0100 0000 **@** At	**065** x41 0100 0001 **A**	**066** x42 0100 0010 **B**	**067** x43 0100 0011 **C**	**068** x44 0100 0100 **D**
069 x45 0100 0101 **E**	**070** x46 0100 0110 **F**	**071** x47 0100 0111 **G**	**072** x48 0100 1000 **H**	**073** x49 0100 1001 **I**	**074** x4A 0100 1010 **J**	**075** x4B 0100 1011 **K**	**076** x4C 0100 1100 **L**	**077** x4D 0100 1101 **M**	**078** x4E 0100 1110 **N**
079 x4F 0100 1111 **O**	**080** x50 0101 0000 **P**	**081** x51 0101 0001 **Q**	**082** x52 0101 0010 **R**	**083** x53 0101 0011 **S**	**084** x54 0101 0100 **T**	**085** x55 0101 0101 **U**	**086** x56 0101 0110 **V**	**087** x57 0101 0111 **W**	**088** x58 0101 1000 **X**
089 x59 0101 1001 **Y**	**090** x5A 0101 1010 **Z**	**091** x5B 0101 1011 **[** Open Bracket	**092** x5C 0101 1100 **** Backslash (\\)	**093** x5D 0101 1101 **]** Close Bracket	**094** x5E 0101 1110 **^** Caret	**095** x5F 0101 1111 _ Underscore	**096** x60 0110 0000 ` Grave Accent	**097** x61 0110 0001 **a**	**098** x62 0110 0010 **b**
099 x63 0110 0011 **c**	**100** x64 0110 0100 **d**	**101** x65 0110 0101 **e**	**102** x66 0110 0110 **f**	**103** x67 0110 0111 **g**	**104** x68 0110 1000 **h**	**105** x69 0110 1001 **i**	**106** x6A 0110 1010 **j**	**107** x6B 0110 1011 **k**	**108** x6C 0110 1100 **l**
109 x6D 0110 1101 **m**	**110** x6E 0110 1110 **n**	**111** x6F 0110 1111 **o**	**112** x70 0111 0000 **p**	**113** x71 0111 0001 **q**	**114** x72 0111 0010 **r**	**115** x73 0111 0011 **s**	**116** x74 0111 0100 **t**	**117** x75 0111 0101 **u**	**118** x76 0111 0110 **v**
119 x77 0111 0111 **w**	**120** x78 0111 1000 **x**	**121** x79 0111 1001 **y**	**122** x7A 0111 1010 **z**	**123** x7B 0111 1011 **{** Open Brace	**124** x7C 0111 1100 **|** Vertical Bar / Pipe	**125** x7D 0111 1101 **}** Close Brace	**126** x7E 0111 1110 **~** Tilde	**127** x7F 0111 1111 **DEL** Delete	

Appendix C VFD / LCD Character Sets

4X20 VF Display International/Katakana Character Sets

Dec	Hex	000	x00	001	x01	002	x02	003	x03	004	x04	005	x05	006	x06	007	x07	008	x08	009	x09	010	x0A	011	x0B
Int	Kat																	Cursor Left		Cursor Right		Cursor Down			
012	x0C	013	x0D	014	x0E	015	x0F	016	x10	017	x11	018	x12	019	x13	020	x14	021	x15	022	x16	023	x17	024	x18
Cursor Home		Cursor to Start of Line		Clear Display						Scroll Mode Off		Scroll Mode On				Cursor Off		Cursor On						International Characters	
025	x19	026	x1A	027	x1B	028	x1C	029	x1D	030	x1E	031	x1F	032	x20	033	x21	034	x22	035	x23	036	x24	037	x25
Katakana Characters				Escape										Space		!	!	"	"	#	#	$	$	%	%
038	x26	039	x27	040	x28	041	x29	042	x2A	043	x2B	044	x2C	045	x2D	046	x2E	047	x2F	048	x30	049	x31	050	x32
&	&	'	'	(	(	)	)	*	*	+	+	,	,	-	-	.	.	/	/	0	0	1	1	2	2
051	x33	052	x34	053	x35	054	x36	055	x37	056	x38	057	x39	058	x3A	059	x3B	060	x3C	061	x3D	062	x3E	063	x3F
3	3	4	4	5	5	6	6	7	7	8	8	9	9	:	:	;	;	<	<	=	=	>	>	?	?
064	x40	065	x41	066	x42	067	x43	068	x44	069	x45	070	x46	071	x47	072	x48	073	x49	074	x4A	075	x4B	076	x4C
@	@	A	A	B	B	C	C	D	D	E	E	F	F	G	G	H	H	I	I	J	J	K	K	L	L
077	x4D	078	x4E	079	x4F	080	x50	081	x51	082	x52	083	x53	084	x54	085	x55	086	x56	087	x57	088	x58	089	x59
M	M	N	N	O	O	P	P	Q	Q	R	R	S	S	T	T	U	U	V	V	W	W	X	X	Y	Y
090	x5A	091	x5B	092	x5C	093	x5D	094	x5E	095	x5F	096	x60	097	x61	098	x62	099	x63	100	x64	101	x65	102	x66
Z	Z	[	[	\	\	]	]	^	^	_	_	`	`	a	a	b	b	c	c	d	d	e	e	f	f
103	x67	104	x68	105	x69	106	x6A	107	x6B	108	x6C	109	x6D	110	x6E	111	x6F	112	x70	113	x71	114	x72	115	x73
g	g	h	h	i	i	j	j	k	k	l	l	m	m	n	n	o	o	p	p	q	q	r	r	s	s
116	x74	117	x75	118	x76	119	x77	120	x78	121	x79	122	x7A	123	x7B	124	x7C	125	x7D	126	x7E	127	x7F	128	x80
t	t	u	u	v	v	w	w	x	x	y	y	z	z	{	{	|	|	}	}	~	~	•	•	€	€
129	x81	130	x82	131	x83	132	x84	133	x85	134	x86	135	x87	136	x88	137	x89	138	x8A	139	x8B	140	x8C	141	x8D
?	?	,	,	ƒ	ƒ	„	„	…	…	†	†	‡	‡	ˆ	ˆ	‰	‰	Š	Š	‹	‹	Œ	Œ	?	?
142	x8E	143	x8F	144	x90	145	x91	146	x92	147	x93	148	x94	149	x95	150	x96	151	x97	152	x98	153	x99	154	x9A
?	?	?	?	?	?	‘	‘	?	?	“	“	?	?	•	•	–	–	—	—	˜	˜	™	™	š	š
155	x9B	156	x9C	157	x9D	158	x9E	159	x9F	160	xA0	161	xA1	162	xA2	163	xA3	164	xA4	165	xA5	166	xA6	167	xA7
›	›	œ	œ	?	?	?	?	Ÿ	Ÿ			¡	¡	¢	¢	£	£	¤	¤	¥	¥	¦	¦	§	§
168	xA8	169	xA9	170	xAA	171	xAB	172	xAC	173	xAD	174	xAE	175	xAF	176	xB0	177	xB1	178	xB2	179	xB3	180	xB4
¨	¨	©	©	ª	ª	«	«	¬	¬	-	-	®	®	¯	¯	°	°	±	±	²	²	³	³	´	´
181	xB5	182	xB6	183	xB7	184	xB8	185	xB9	186	xBA	187	xBB	188	xBC	189	xBD	190	xBE	191	xBF	192	xC0	193	xC1
µ	µ	¶	¶	·	·	¸	¸	¹	¹	º	º	»	»	¼	¼	½	½	¾	¾	¿	¿	À	À	Á	Á
194	xC2	195	xC3	196	xC4	197	xC5	198	xC6	199	xC7	200	xC8	201	xC9	202	xCA	203	xCB	204	xCC	205	xCD	206	xCE
Â	Â	Ã	Ã	Ä	Ä	Å	Å	Æ	Æ	Ç	Ç	È	È	É	É	Ê	Ê	Ë	Ë	Ì	Ì	Í	Í	Î	Î
207	xCF	208	xD0	209	xD1	210	xD2	211	xD3	212	xD4	213	xD5	214	xD6	215	xD7	216	xD8	217	xD9	218	xDA	219	xDB
Ï	Ï	Ð	Ð	Ñ	Ñ	Ò	Ò	Ó	Ó	Ô	Ô	Õ	Õ	Ö	Ö	×	×	Ø	Ø	Ù	Ù	Ú	Ú	Û	Û
220	xDC	221	xDD	222	xDE	223	xDF	224	xE0	225	xE1	226	xE2	227	xE3	228	xE4	229	xE5	230	xE6	231	xE7	232	xE8
Ü	Ü	Ý	Ý	Þ	Þ	ß	ß	à	à	á	á	â	â	ã	ã	ä	ä	å	å	æ	æ	ç	ç	è	è
233	xE9	234	xEA	235	xEB	236	xEC	237	xED	238	xEE	239	xEF	240	xF0	241	xF1	242	xF2	243	xF3	244	xF4	245	xF5
é	é	ê	ê	ë	ë	ì	ì	í	í	î	î	ï	ï	ð	ð	ñ	ñ	ò	ò	ó	ó	ô	ô	õ	õ
246	xF6	247	xF7	248	xF8	249	xF9	250	xFA	251	xFB	252	xFC	253	xFD	254	xFE	255	xFF						
ö	ö	÷	÷	ø	ø	ù	ù	ú	ú	û	û	ü	ü	ý	ý	þ	þ	ÿ	ÿ						

8X40 and 16X40 LCD Character Sets

Appendix D Parameter Setup

Table D-1: 460 Series Parameter Setup

PARAMETER	SELECTION	DESCRIPTION (460 SERIES)
Scale Configuration		
P108.01	Scale 1	**Scale Instance Selection**
P108.02	Scale 2	Enter 1 → 2
P109.00	Disabled	**Scale Enable**
P109.01	Saved	Enter 0 → 2
P109.02 †	Enabled	
P110. †	100.00	**Full Scale Capacity**
		Enter 0.01 → 1,000,000
P111.00	.00001	**Division Size**
P111.01	.00002	Enter 0 → 23
P111.02	.00005	
P111.03	.0001	
P111.04	.0002	
P111.05	.0005	
P111.06	.001	
P111.07	.002	
P111.08	.005	
P111.09 †	.01	
P111.10	.02	
P111.11	.05	
P111.12	.1	
P111.13	.2	
P111.14	.5	
P111.15	1	
P111.16	2	
P111.17	5	
P111.18	10	
P111.19	20	
P111.20	50	
P111.21	100	
P111.22	200	
P111.23	500	
P112.00	OFF	**Zero Track Divisions**
P112.01	0.1d	Enter 0 → 200
P112.02	0.2d	
P112.03	0.3d	
P112.04	0.4d	
P112.05	0.5d	
P112.06	0.6d	
P112.07	0.7d	
P112.08	0.8d	
P112.09	0.9d	
P112.10 †	1.0d	
↓	↓	
P112.200	20.0d	
P113.00	0.05s	**Zero Track Delay (seconds)**
P113.01	0.1s	Enter 0 → 100
P113.02	0.2s	
P113.03	0.3s	
P113.04	0.4s	
P113.05 †	0.5s	
↓	↓	
P113.100	10.0s	
P114.00	OFF	**Motion Divisions**
P114.01	0.1d	Enter 0 → 200
P114.02	0.2d	
P114.03	0.3d	
P114.04	0.4d	
P114.05	0.5d	
P114.06	0.6d	
P114.07	0.7d	
P114.08	0.8d	
P114.09	0.9d	
P114.10 †	1.0d	
↓	↓	
P114.200	20.0d	
P115.00	0.05s	**Motion Delay (seconds)**
P115.01	0.1s	Enter 0 → 100
P115.02	0.2s	
P115.03	0.3s	
P115.04	0.4s	
P115.05	0.5s	
P115.06	0.6s	
P115.07	0.7s	
P115.08	0.8s	
P115.09	0.9s	
P115.10 †	1.0s	
↓	↓	
P115.100	10.0s	
P116.00	0.06s	**Digital Filter**
P116.01	0.13s	Enter 0 → 11
P116.02	0.25s	
P116.03	0.50s	
P116.04	1.0s	
P116.05	2.0s	
P116.06	4.0s	
P116.07	8.0s	
P116.08	2.0sA	
P116.09 †	4.0sA	
P116.10	8.0sA	
P116.11	OFF	
P117.00 †	0.05s	**Display Update Rate (seconds)**
P117.01	0.1s	Enter 0 → 200
P117.02	0.2s	
↓	↓	
P117.200	20.0s	
P118.00	0.01%	**Zero Range**
P118.01	0.02%	Enter 0 → 12
P118.02	0.04%	
P118.03	0.1%	
P118.04	0.2%	
P118.05	0.4%	
P118.06	1.0%	
P118.07	2.0%	
P118.08	4.0%	
P118.09	10.0%	
P118.10	20.0%	
P118.11	40.0%	
P118.12 †	100%	
P119.00 †	Disabled	**Linearization**
P119.01	Enabled	Enter 0 → 1
P122.00	0.01%	**Return To Zero (RTZ)**
P122.01	0.02%	Enter 0 → 12
P122.02	0.04%	
P122.03 †	0.1%	
P122.04	0.2%	
P122.05	0.4%	
P122.06	1.0%	
P122.07	2.0%	
P122.08	4.0%	
P122.09	10.0%	
P122.10	20.0%	
P122.11	40.0%	
P122.12	100%	
P124. †	0.000	**Count Resolution**
		Enter 100 → 1,000,000
P125. †	1.000	**Count Adjustment Factor**
		Enter 0.1 → 20.0
P126. †	0.000	**Low Range Capacity**
		Enter 0.001 → 1,000,000
P127.00	.00001	**Low Range Division Size**
P127.01	.00002	Enter 0 → 23

PARAMETER	SELECTION	DESCRIPTION (460 SERIES)
P127.02	.00005	
P127.03	.0001	
P127.04	.0002	
P127.05	.0005	
P127.06	.001	
P127.07	.002	
P127.08	.005	
P127.09 †	.01	
P127.10	.02	
P127.11	.05	
P127.12	.1	
P127.13	.2	
P127.14	.5	
P127.15	1	
P127.16	2	
P127.17	5	
P127.18	10	
P127.19	20	
P127.20	50	
P127.21	100	
P127.22	200	
P127.23	500	
P128.	0.000	**Middle Range Capacity** Enter 0.01 → 1,000,000
P127.00	.00001	**Middle Range Division Size** Enter 0 → 23
P127.01	.00002	
P127.02	.00005	
P127.03	.0001	
P127.04	.0002	
P127.05	.0005	
P127.06	.001	
P127.07	.002	
P127.08	.005	
P127.09 †	.01	
P127.10	.02	
P127.11	.05	
P127.12	.1	
P127.13	.2	
P127.14	.5	
P127.15	1	
P127.16	2	
P127.17	5	
P127.18	10	
P127.19	20	
P127.20	50	
P127.21	100	
P127.22	200	
P127.23	500	
P130.00 †	Gross	**Multi-Range Mode** Enter 0 → 1
P130.01	Net	
P131.00 †	Pounds	**First Units** Enter 0 → 8
P131.01	Kilograms	
P131.02	Ounces	
P131.03	Grams	
P131.04	US Tons	
P131.05	Metric Tons	
P131.06	Custom Unit1	
P131.07	Custom Unit2	
P131.08	Pounds-Ounces	
P132.00	Pounds	**Second Units** Enter 0 → 9
P132.01 †	Kilograms	
P132.02	Ounces	
P132.03	Grams	
P132.04	US Tons	
P132.05	Metric Tons	
P132.06	Custom Unit1	
P132.07	Custom Unit2	
P132.08	Pounds-Ounces	
P132.09	NONE	
P133.00	Pounds	**Third Units** Enter 0 → 9
P133.01	Kilograms	

PARAMETER	SELECTION	DESCRIPTION (460 SERIES)
P133.02	Ounces	
P133.03	Grams	
P133.04	US Tons	
P133.05	Metric Tons	
P133.06	Custom Unit1	
P133.07	Custom Unit2	
P133.08	Pounds-Ounces	
P133.09 †	NONE	
P134.00	Pounds	**Fourth Units** Enter 0 → 9
P134.01	Kilograms	
P134.02	Ounces	
P134.03	Grams	
P134.04	US Tons	
P134.05	Metric Tons	
P134.06	Custom Unit1	
P134.07	Custom Unit2	
P134.08	Pounds-Ounces	
P134.09 †	NONE	
P135. †	Off	**Rate Measurement Period** Enter 0 → 900s
P136.00 †	Seconds	**Rate Time Unit** Enter 0 → 2
P136.01	Minutes	
P136.02	Hours	
P142.00	Disabled	**Center-of-Zero Annunciator** Enter 0 → 1
P142.01 †	Enabled	
P143. †	O	**Over/Under Load Status Name** Enter Name
P144. †	M	**Motion Status Name** Enter Name
P145. †	S	**Stable Status Name** Enter Name
P146. †	O	**Underload Status Name** Enter Name
P147. †	E	**Error Status Name** Enter Name
Units		
P150.00 †	Pounds	**Default Units** Enter 1 → 6
P150.01	Kilograms	
P150.02	Ounces	
P150.03	Grams	
P150.04	US Tons	
P150.05	Metric Tons	
P150.06	Custom Unit1	
P151. †	????1	**Custom Unit1 Name** Enter Name
P152. †	1.000	**Custom Unit1 Calibration Factor** Enter 0.000001 → 9,999,999
P153. †	????2	**Custom Unit 2 Name** Enter Name
P154. †	1.000	**Custom Unit2 Calibration Factor** Enter 0.000001 → 9,999,999
Tare Functions		
P162.00 †	Disabled	**Negative Tare Enable** Enter 0 → 1
P162.01	Enabled	
P163.00	Disabled	**Tare Rounding Enable** Enter 0 → 1
P163.01 †	Enabled	
Analog Output		
P170.01	Analog Out 1	**Analog Output Instance Selection** Enter 1 → 2
P170.02	Analog Out 2	
P171.00 †	Disabled	**Analog Output Enable** Enter 0 → 1
P171.01	Enabled	
P172. †	Gross	**Output Parameter** Enter Operating Parameter
P173. †	None!	**Full Scale Output** Enter Operating Parameter
P174. †	None!	**Zero Offset** Enter Operating Parameter
P175. †	None!	**Output Signal Range** Enter Operating Parameter
P176.00 †	Maximum	**Default Output in Setup Mode** Enter 0 → 2
P176.01	Minimum	

PARAMETER	SELECTION	DESCRIPTION (460 SERIES)
P176.02	Same	
P177.00 † P177.01 P177.02	0-10 VDC 0-20 mA 4-20 mA	**Analog Output Type** Enter 0→ 2
Counting		
P179.00 † P179.01	Disabled Enabled	**Count Enable** Enter 0→ 1
P180.00 † P180.01	off on	**Auto Sample Enable** Enter 0→ 1
P181.00 P181.01 †	off on	**Auto Enhance Enable** Enter 0→ 1
P182.01 P182.02 P182.03 P182.04 P182.05 P182.06 P182.07 P182.08 P182.09 P182.10 † ↓ P182.9999	1 piece 2 pieces 3 pieces 4 pieces 5 pieces 6 pieces 7 pieces 8 pieces 9 pieces 10 pieces ↓ 9999 pieces	**Default Sample Size** Enter 1→ 9999
P183. †	98.52%	**Required Accuracy** Enter 0, 90.00→ 99.96
P184.00 † P184.01	off on	**Accuracy Display Enable** Enter 0→ 1
P185.00 † P185.01 P185.02	None! Scale 1 Scale 2	**Pre-Sample Scale** Enter 0→ 2
P186.00 † P186.01 P186.02	None! Scale 1 Scale 2	**After-Sample Scale** Enter 0→ 2
P187.00 P187.01 P187.02 P187.03 P187.04 P187.05 P187.06 † P187.07	None! 0.13s 0.25s 0.50s 1.0s 2.0s 4.0s 8.0s	**Sample Filter** Enter 0→ 7
P188.00 P188.01	off on	**Enforce Sample Accuracy** Enter 0→ 1
P189.00 † P189.01 P189.02 ↓ P189.15	0.0d 0.1d 0.2d ↓ 1.5d	**Sample Motion Divisions** Enter 1→ 15
Communication Ports		
P199.01 P199.02	Comm Port 1 Comm Port 2	**Serial Port Instance Selection** Enter 1→ 2
P200.00 P200.01 P200.02 † P200.03 P200.04 P200.05 P200.06 P200.07 P200.08 P200.12 P200.13 **	37400 19200 9600 4800 2400 1200 600 300 150 58300 38400	**Baud Rate** Enter 0→ 8, 12→ 13 *** Alternate processor clock speed will be automatically selected (14MHz)*
P201.00 P201.01 †	7 bits 8 bits	**Data Bits** Enter 0→ 1
P202.00 † P202.01 P202.02	None Even Odd	**Parity** Enter 0→ 2
P203.00 † P203.01	1 bit 2 bits	**Stop Bits** Enter 0→ 1
P204.00 P204.01 P204.02 † P204.03	None CTS/RTS Xon/Xoff Both	**Handshaking** Enter 0→ 3
P205.00 P205.01 † P205.02	Disabled Standard Interpreter	**Receive Mode** Enter 0→ 250
P205.03 P205.04 ↓ P205.250	Modbus Macro 4 ↓ Macro 250	
P206.00 † P206.01	Delay Abort	**Transmit Mode** Enter 0→ 1
P207.08 P207.09 P207.10 ↓ P207.128 † ↓ P207.9999	8 bytes 9 bytes 10 bytes ↓ 128 bytes ↓ 9999 bytes	**Transmit Buffer Size (bytes)** Enter 8→ 3967 *Maximum combined value for Transmit and Receive buffers of both ports is 4096 bytes.*
P208.08 P208.09 P208.10 ↓ P208.256 † ↓ P207.9999	8 bytes 9 bytes 10 bytes ↓ 256 bytes ↓ 9999 bytes	**Receive Buffer Size (bytes)** Enter 8→ 3967 *Maximum combined value for Transmit and Receive buffers of both ports is 4096 bytes.*
P209.01 P209.02 P209.03 ↓ P209.247	1 2 3 ↓ 247	**Modbus Address** Enter 1→ 247
P210.00 † P210.01	ASCII RTU	**Modbus Mode** Enter 0→ 1
P211.00 † P211.01	Hi/Lo Lo/Hi	**Modbus Word Hi/Lo** Enter 0→ 1
Input Interpreter		
P217.00 † P217.01	Disabled Enabled	**Interpreter NULL** Enter 0→ 1
P218.00 P218.01 P218.02 ↓ P218.10 † ↓ P218.255	<NUL> <SOH> <STX> ↓ <LF> ↓ <255>	**Receive Termination Character** Enter ASCII Value .000→ .255
P219.00 † P219.01 P219.02 P219.03 ↓ P219.15	None! Interp. 1 Interp. 2 Interp. 3 ↓ Interp. 15	**Input Interpreter Instance Selection** Enter 1→ 15
P220. †	None!	**Interpreter Name** Enter Name
P221.00 † P221.01	Character Line	**Interpreter Type** Enter 0→ 1
P222.00		**Line Interpreter Entry Table** Enter text, operating parameters, control codes
P223.00 † P223.01 P223.02 ↓ P218.255	<NUL> <SOH> <STX> ↓ <255>	**Interpreter Character** Enter ASCII Value .000→ .255
P224.00 † P224.01 P224.02 P224.03 ↓ P224.15	None! Macro 1 Macro 2 Macro 3 ↓ Macro 15	**Interpreter Macro#** Enter 1→ 15
Numeric Parameter Formatting		
P240.01 P240.02 P240.03 P240.04 P240.05 P240.06 P240.07 P240.08 † ↓ P240.15	1 2 3 4 5 6 7 8 ↓ 15	**Minimum Transmit Width** Enter 1→ 15
P241.00 † P241.01	Right Left	**Sign Justification** Enter 0→ 1
Networking & Remote Communications		

PARAMETER	SELECTION	DESCRIPTION (460 SERIES)
P250.00 †	Disabled	**Network Enable**
P250.01	Enabled	Enter 0→ 1
P251.00 †	Disabled	**Network Address**
P251.04	4	Enter 0, 4→ 254
P251.05	5	
P251.06	6	
↓	↓	
P251.254	254	
P290.00 †	Off	**Echo Display**
P290.01	Comm Port 1	Enter 0→ 2
P290.02	Comm Port 2	
P291.00	<NUL>	**Echo Start Character**
P291.01	<SOH>	Enter ASCII Value 0→ 255
P291.02 †	<STX>	
P291.03	<ETX>	
↓	↓	
P291.255	<255>	
P292.00	<NUL>	**Echo End Character**
P292.01	<SOH>	Enter ASCII Value 0→ 255
P292.02	<STX>	
P292.03 †	<ETX>	
↓	↓	
P292.255	<255>	
P293.00 †	Disabled	**Remote Display**
P293.01	LCD	Enter 0→ 2
P293.02	LED	
P294.00 †	Disabled	**LCD Remote Display Backlight**
P294.01	Enabled	Enter 0→ 1
Weigh Mode Parameter Selections		
P300. †	Gross	**[SELECT] Mode 0**
		Enter Operating Parameter
P301. †	Net	**[SELECT] Mode 1**
		Enter Operating Parameter
P302. †	Tare	**[SELECT] Mode 2**
		Enter Operating Parameter
P303. †	None!	**[SELECT] Mode 3**
		Enter Operating Parameter
P304. †	None!	**[SELECT] Mode 4**
		Enter Operating Parameter
P305. †	None!	**[SELECT] Mode 5**
		Enter Operating Parameter
P306. †	None!	**[SELECT] Mode 6**
		Enter Operating Parameter
P307. †	None!	**[SELECT] Mode 7**
		Enter Operating Parameter
P308. †	None!	**[SELECT] Mode 8**
		Enter Operating Parameter
P309. †	None!	**[SELECT] Mode 9**
		Enter Operating Parameter
Access Codes		
P400. †	None!	**Personal Identification Number**
		Enter Code
P401. †	None!	**Quick Calibration Access Code**
		Enter Code
P402. †	21353	**Limited Access Code**
		Enter Code
OIML		
P410. †	Disabled	**OIML Enable**
		Enter 9990, 9991
P411.00 †	USA	**Language Character Set**
P411.01	France	Enter 0→ 12
P411.02	German	
P411.03	UK	
P411.04	Denmark	

PARAMETER	SELECTION	DESCRIPTION (460 SERIES)
P411.05	Sweden	
P411.06	Italy	
P411.07	Spain	
P411.08	Japan	
P411.09	Norway	
P411.10	Denmark2	
P411.11	Spain2	
P411.12	Latin America	
P412.00 †	Disable	**Preset Enable**
P412.01	Enable	Enter 0→ 1
VFD Setup		
P420.00	Off	**Standard VF Display Mode**
P420.01 †	On	Enter 0→ 2
P420.02	Auto	
P421.00	2d	**Weight Threshold Divisions**
P421.01	4d	Enter 0→ 15
P421.02 †	6d	
P421.03	8d	
P421.04	10d	
P421.05	12d	
P421.06	14d	
P421.07	16d	
P421.08	18d	
P421.09	20d	
P421.10	22d	
P421.11	24d	
P421.12	26d	
P421.13	28d	
P421.14	30d	
P421.15	32d	
P422.00	30 seconds	**Timeout**
P422.01	1 minutes	Enter 0→ 15
P422.02	2 minutes	
P422.03	3 minutes	
P422.04	4 minutes	
P422.05 †	5 minutes	
P422.06	10 minutes	
P422.07	15 minutes	
P422.08	20 minutes	
P422.09	25 minutes	
P422.10	30 minutes	
P422.11	35 minutes	
P422.12	40 minutes	
P422.13	45 minutes	
P422.14	1 hour	
P422.15	2 hours	
P423.01	10%	**Display 'ON' Brightness**
P423.02	20%	Enter 1→ 10
P423.03	30%	
P423.04	40%	
P423.05	50%	
P423.06	60%	
P423.07	70%	
P423.08	80%	
P423.09	90%	
P423.10 †	100%	
P424.00 †	Off	**Display 'OFF' Dimness**
P424.01	10%	Enter 0→ 10
P424.02	20%	
P424.03	30%	
P424.04	40%	
P424.05	50%	
P424.06	60%	
P424.07	70%	
P424.08	80%	
P424.09	90%	
P424.10	100%	
P425.00 †	Disabled	**4X20 VF Display Enable**
P425.01	Enabled	Enter 0→ 1
NTEP		
P440.00 †	Disabled	**NTEP Enable**
P440.01	Enabled	Enter 0→ 1
Keypad		
P450.9990	5 Key (460)	**Keypad Selection**
P450.9991	22 Key (465)	Enter 9990, 9991
P451.00	None	**Keypad Repeat Rate**
P451.01	Very Slow	Enter 0→ 7

PARAMETER	SELECTION	DESCRIPTION (460 SERIES)
P451.02 P451.03 P451.04 P451.05 P451.06 † P451.07	Slow Medium Slow Medium Medium Fast Fast Very Fast	
P460.00 P460.01 P460.02 P460.03 † P460.04 P460.05 P460.06 P460.07	Off Minimum Extra Low Low Medium Medium High High Maximum	**Beeper Volume** Enter 0→ 7
Time & Date		
P500. †	00:00	**Time** Enter 24 Hour Time (hh.mm.ss)
P501.70 †	01/01/70	**Date** Enter Date (mm.dd.yy)
P502.00 † P502.01	Disabled Enabled	**Time/Date Access** Enter 0→ 1
P503.00 P503.01 †	No Yes	**AM/PM Time Format** Enter 0→ 1
P504.00 † P504.01	U.S.A. International	**Date Format** Enter 0→ 1
Parameter Renaming		
P600. †	None!	**Rename Gross** Enter Name
P601. †	None!	**Rename Net** Enter Name
P602. †	None!	**Rename Tare** Enter Name
P603. †	None!	**Rename Gross Total** Enter Name
P604. †	None!	**Rename Gross Total + Current** Enter Name
P605. †	None!	**Rename Gross Total – Current** Enter Name
P606. †	None!	**Rename Net Total** Enter Name
P607. †	None!	**Rename Net Total + Current** Enter Name
P608. †	None!	**Rename Net Total – Current** Enter Name
P609. †	None!	**Rename Accumulation** Enter Name
P610. †	None!	**Rename Scale#** Enter Name
P611. †	None!	**Rename Time/Date** Enter Name
P615. †	None!	**Rename Average Gross** Enter Name
P616. †	None!	**Rename Average Net** Enter Name
P618. †	None!	**Rename Peak Gross** Enter Name
P619. †	None!	**Rename Peak Net** Enter Name
P620. †	None!	**Rename Rounded Gross** Enter Name
P621. †	None!	**Rename Rounded Net** Enter Name
P623. †	None!	**Rename Rate** Enter Name
P624. †	None!	**Rename Free Fall 1** Enter Name
P625. †	None!	**Rename Future Gross 1** Enter Name
P626. †	None!	**Rename Future Net 1** Enter Name
P627. †	None!	**Rename Free Fall 2** Enter Name
P628. †	None!	**Rename Future Gross 2**

PARAMETER	SELECTION	DESCRIPTION (460 SERIES)
		Enter Name
P629. †	None!	**Rename Future Net 2** Enter Name
P630. †	None!	**Rename Quantity** Enter Name
P631. †	None!	**Rename Quantity Total** Enter Name
P632. †	None!	**Rename Quantity Total+Current** Enter Name
P633. †	None!	**Rename Quantity Total - Current** Enter Name
P634. †	None!	**Rename Average Piece Weight** Enter Name
P635. †	None!	**Rename Avg Piece Weight x K** Enter Name
P636. †	None!	**Rename Percent Accuracy** Enter Name
P637. †	None!	**Rename Sample** Enter Name
P640. †	None!	**Rename Grs Total of All Scales** Enter Name
P641. †	None!	**Rename Net Total of All Scales** Enter Name
P642. †	None!	**Rename Tare Total of All Scales** Enter Name
P643. †	None!	**Rename Total of All Grs Totals** Enter Name
P644. †	None!	**Rename Total of All Net Totals** Enter Name
P645. †	None!	**Rename Qty Total of All Scales** Enter Name
P646. †	None!	**Rename Total of All Qty Totals** Enter Name
Total & Tare Save		
P660.00 P660.01 P660.02 †	No Save On Request Auto Save	**Total Values Save Method** Enter 0→ 2
P661.00 P661.01 P661.02 †	No Save On Request Auto Save	**Tare Value Save Method** Enter 0→ 2
Variables		
P680.00 † P680.01 P680.02 P680.03 ↓ P680.15	None! 1 2 3 ↓ 15	**Variable Allocation** Enter 0→ 15
P681.01 P681.02 P681.03 ↓ P681.15	Variable 1 Variable 2 Variable 3 ↓ Variable 15	**Variable Instance Selection**
P682. †	None!	**Variable Name** Enter Name
P684.00 † P684.01 P684.02	No Save On Request Auto Save	**Variable Value Save Method** Enter 0→ 2
P685.00 † P685.01	Disabled Enabled	**Variable Lock** Enter 0→ 1
P686.00 † P686.01 P686.02 P686.03	Float Integer Unsigned Int String	**Variable Type** Enter 0→ 3
P687.00 P687.01 P687.02 P687.03 P687.04 P687.05 P687.06 † P687.07	0 decimal pt 1 decimal pt 2 decimal pt 3 decimal pt 4 decimal pt 5 decimal pt Auto decimal Scale 1	**Float Style** Enter 0→ 14

PARAMETER	SELECTION	DESCRIPTION (460 SERIES)
P687.08	Scale 2	
P688.00 †	Number	**Integer Style**
P688.01	Time/Date	
P688.02	Time	
P688.03	Date	
P689.01	1 character	**String Size**
P689.02	2 characters	Enter 1 → 63
P689.03	3 characters	
P698.04	4 characters	
P698.05	5 characters	
P698.06	6 characters	
P698.07	7 characters	
P698.08	8 characters	
P698.09	9 characters	
P698.10 †	10 characters	
↓	↓	
P689.63	63 characters	
Database		
P699.00 †	None!	**Database Instance Selection**
P699.01	Database 1	Enter 1 → 15
P699.02	Database 2	
P699.03	Database 3	
↓	↓	
P699.15	Database 15	
P700. †	None!	**Database Name**
		Enter Name
P701.	Column 1	**Database Column Parameter**
P702.	Column 2	Enter Operating Parameter
P703.	Column 3	
↓	↓	
P798.	Column 98	
Keypad Key Assignments		
P800.00 †	Enabled	**[SELECT] Key Function**
P800.01	Macro 1	Enter 0 → 15
P800.02	Macro 2	
P800.03	Macro 3	
↓	↓	
P800.15	Macro 15	
P801.00 †	Enabled	**[ZERO] Key Function**
P801.01	Macro 1	Enter 0 → 15
P801.02	Macro 2	
P801.03	Macro 3	
↓	↓	
P801.15	Macro 15	
P802.00 †	Enabled	**[TARE] Key Function**
P802.01	Macro 1	Enter 0 → 15
P802.02	Macro 2	
P802.03	Macro 3	
↓	↓	
P802.15	Macro 15	
P803.00 †	Enabled	**[UNITS] Key Function**
P803.01	Macro 1	Enter 0 → 15
P803.02	Macro 2	
P803.03	Macro 3	
↓	↓	
P803.15	Macro 15	
P804.00 †	Enabled	**[SCALE SELECT] Key Function**
P804.01	Macro 1	Enter 0 → 15
P804.02	Macro 2	
P804.03	Macro 3	
↓	↓	
P804.15	Macro 15	
P805.00 †	Enabled	**[PRINT] Key Function**
P805.01	Macro 1	Enter 0 → 15
P805.02	Macro 2	
P805.03	Macro 3	
↓	↓	
P805.15	Macro 15	
P806.00 †	None	**[ID] Key Function**
P806.01	Menu	Enter 0 → 15
P806.02	Database	
P806.03	Menu & Dbase	
P806.04	Macro 4	
P806.05	Macro 5	
P806.06	Macro 6	
↓	↓	

PARAMETER	SELECTION	DESCRIPTION (460 SERIES)
P806.15	Macro 15	
P807.00 †	Enabled	**[ENTER] Key Function**
P807.01	Macro 1	Enter 0 → 15
P807.02	Macro 2	
P807.03	Macro 3	
↓	↓	
P807.15	Macro 15	
P808.00 †	Enabled	**[CLEAR] Key Function**
P808.01	Macro 1	Enter 0 → 15
P808.02	Macro 2	
P808.03	Macro 3	
↓	↓	
P808.15	Macro 15	
P809.00 †	Enabled	**[.] Key Function**
P809.01	Macro 1	Enter 0 → 15
P809.02	Macro 2	
P809.03	Macro 3	
↓	↓	
P809.15	Macro 15	
P810.00 †	Enabled	**[0] Key Function**
P810.01	Macro 1	Enter 0 → 15
P810.02	Macro 2	
P810.03	Macro 3	
↓	↓	
P810.15	Macro 15	
P811.00 †	Enabled	**[1] Key Function**
P811.01	Macro 1	Enter 0 → 15
P811.02	Macro 2	
P811.03	Macro 3	
↓	↓	
P811.15	Macro 15	
P812.00 †	Enabled	**[2] Key Function**
P812.01	Macro 1	Enter 0 → 15
P812.02	Macro 2	
P812.03	Macro 3	
↓	↓	
P812.15	Macro 15	
P813.00 †	Enabled	**[3] Key Function**
P813.01	Macro 1	Enter 0 → 15
P813.02	Macro 2	
P813.03	Macro 3	
↓	↓	
P813.15	Macro 15	
P814.00 †	Enabled	**[4] Key Function**
P814.01	Macro 1	Enter 0 → 15
P814.02	Macro 2	
P814.03	Macro 3	
↓	↓	
P814.15	Macro 15	
P815.00 †	Enabled	**[5] Key Function**
P815.01	Macro 1	Enter 0 → 15
P815.02	Macro 2	
P815.03	Macro 3	
↓	↓	
P815.15	Macro 15	
P816.00 †	Enabled	**[6] Key Function**
P816.01	Macro 1	Enter 0 → 15
P816.02	Macro 2	
P816.03	Macro 3	
↓	↓	
P816.15	Macro 15	
P817.00 †	Enabled	**[7] Key Function**
P817.01	Macro 1	Enter 0 → 15
P817.02	Macro 2	
P817.03	Macro 3	
↓	↓	
P817.15	Macro 15	
P818.00 †	Enabled	**[8] Key Function**
P818.01	Macro 1	Enter 0 → 15
P818.02	Macro 2	
P818.03	Macro 3	
↓	↓	
P818.15	Macro 15	
P819.00 †	Enabled	**[9] Key Function**
P819.01	Macro 1	Enter 0 → 15

PARAMETER	SELECTION	DESCRIPTION (460 SERIES)
P819.02 P819.03 ↓ P819.15	Macro 2 Macro 3 ↓ Macro 15	
P820.00 † P820.01 P820.02 P820.03 ↓ P820.15	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 15	**Any Key Function** Enter 0→ 15
Custom Transmit		
P980.00 P980.01 P980.02 P980.03 P980.04 P980.05 † ↓ P980.250	Disabled 0.1 seconds 0.2 seconds 0.3 seconds 0.4 seconds 0.5 seconds ↓ 25.0 seconds	**Continuous Transmit Rate** Enter 0→ 250
P989.01 † P989.02 P989.03 P989.04	Transmit 1 Transmit 2 Transmit 3 Transmit 4	**Custom Transmit Instance Selection** Enter 1→ 4
P990. †	None!	**Custom Transmit Name** Enter Name
P991.00 P991.01 † P991.02	Off On Request Prompt	**Transmit Mode** Enter 0→ 2
P992.01 † P992.02	Comm Port 1 Comm Port 2	**Serial Port Selection** Enter 1→ 2
P993.00 † P993.01 *	Ignored Delayed	**Current Scale Motion** Enter 0→ 1 ** Default custom transmit #1 is motion delayed.*
P994. †	Ignored	**Scale Motion** Enter 0 for motion ignored Enter any combination of 1→ 2 for motion delay on entered scale numbers.
P995.00 † P995.01	Ignored Delayed	**Scale 2 Motion** See P994.
P996.00 † P996.01	Ignored Delayed	**Scale 3 Motion** Not used on 460 Series.
P997.00 † P997.01	Ignored Delayed	**Scale 4 Motion** Not used on 460 Series.
P998.00 † P998.01	Disabled Enabled	**Continuous Transmit Enable** Enter 0→ 1
P999.00 † P999.01	No Yes	**Transmit Table Limited Access** Enter 0→ 1
P1000. P1001. P1002. ↓ P4999.		**Transmit Table Entry** Enter text, operating parameters, control codes
Setpoints		
P5099.1 P5099.2 P5099.3 ↓ P5099.16		**Setpoint Instance Selection** Enter 1→ 16
P5100.0 † P5100.1 P5100.2	Disabled Output Input	**Setpoint Mode** Enter 0→ 2
P5101. †	None!	**Setpoint Name** Enter Name
P5110.0 † P5110.1 P5110.2 P5110.3 P5110.4 P5110.5 P5110.6 P5110.7 P5110.10 P5110.11	Above Below Between Outside Always Never Motion Scale1 Motion Scale2 Stable Scale1 Stable Scale2	**Activation Condition** Enter 0→ 7, 10, 11, 14, 15

PARAMETER	SELECTION	DESCRIPTION (460 SERIES)
P5110.14 P5110.15	Mot'n Current Stabl Current	
P5111. †	0.0 seconds	**Activation Delay (seconds)** Enter 0.01→ 5,767,168
P5112.0 † P5112.1 P5112.2 P5112.3 ↓ P5112.15	None! Macro 1 Macro 2 Macro 3 ↓ Macro 15	**Activation Macro#** Enter 0→ 15
P5113.0 † P5113.1	Ignored Delayed	**Activation Motion** Enter 0→ 1
P5114. †	None!	**Lower Activation Parameter** Enter Operating Parameter
P5115. †	None!	**Upper Activation Parameter** Enter Operating Parameter
P5130.0 † P5130.1 P5130.2 P5130.3 P5130.4 P5130.5 P5130.6 P5130.7 P5130.10 P5130.11 P5130.14 P5130.15	Above Below Between Outside Always Never Motion Scale1 Motion Scale2 Stable Scale1 Stable Scale2 Mot'n Current Stabl Current	**Deactivation Condition** Enter 0→ 7, 10, 11, 14, 15
P5131. †	0.00 seconds	**Deactivation Delay (seconds)** Enter 0.01→ 5,767,168
P5132.0 † P5132.1 P5132.2 P5132.3 ↓ P5132.15	None! Macro 1 Macro 2 Macro 3 ↓ Macro 15	**Deactivation Macro#** Enter 0→ 15
P5133.0 † P5133.1	Ignored Delayed	**Deactivation Motion** Enter 0→ 1
P5134. †	None!	**Lower Deactivation Parameter** Enter Operating Parameter
P5135. †	None!	**Upper Deactivation Parameter** Enter Operating Parameter
P5150. †	Gross	**Compare Parameter** Enter Operating Parameter
Modbus Parameter Map		
P6001. † P6002. P6003. ↓ P6247.	None!	**Modbus Address Translation Table** Enter Operating Parameter
Macros		
P9980.0 † P9980.1 P9980.2 P9980.3 ↓ P9980.15	None! Macro 1 Macro 2 Macro 3 ↓ Macro 15	**Abort Macro#** Enter 0→ 15
P9980.1 † P9981.1	Menu Immediate	**Macro Abort Method** Enter 0→ 1
P9990.0 P9990.1 P9990.2 P9990.3 ↓ P9990.15	None! Macro 1 Macro 2 Macro 3 ↓ Macro 15	**Macro Instance Selection** Enter 1→ 15
P9991. †	None!	**Macro Name** Enter Name
P9992.0 † P9992.1	Standard Immediate	**Macro Priority** Enter 0→ 1
P9993.0 † P9993.1	Disabled Enabled	**Macro Menu Enable** Enter 0→ 1
P9994.0 † P9994.1	No Yes	**Macro Table Limited Access** Enter 0→ 1
P10001. P10002.		**Macro Entry Table** Enter Text

PARAMETER	SELECTION	DESCRIPTION (460 SERIES)
P10003. ↓ P19999.		
P50000.0 † P50000.1	No Yes	**Macro Debug Limited Access** Enter 0→ 1
P50001.		**Macro Debug Table** Read-Only
Information & Diagnostic Parameters		
P60000. †	4096	**EEPROM (E^2) Memory**
P60001.	3520	**EEPROM (E^2) Memory Available**
P60002. †	98304	**RAM Installed (bytes)**
P60003.	40960	**RAM Dynamically Allocated**
P60004.	39428	**RAM Available**
P60005.	0	**Macro Trace Buffer Size (bytes)**
P60006.	AMD	**FRAM Type**
P60007.	512K	**FRAM SIze**
P60008.	22904	**FRAM Available**
P60010.	4K	**Optional RAM Installed (K-bytes)**
P60011.	4096	**Optional RAM Installed (bytes)**
P60012.	4032	**Optional RAM Available (bytes)**
P60013.	64	**Optional RAM Used (bytes)**
P60014.	4028	**Optional RAM Block (bytes)**
P60020.	0	**Database RAM Usage**
P60021.	none	**Database Error**
P60030.0 † P60030.1 P60030.2 P60030.3 ↓ P60030.15	None! 1 2 3 ↓ 15	**Database Number** Enter 1→ 15
P60031. †	None!	**Database Name**
P60032.	0	**Number of Rows**
P60033.	28	**Total Bytes Used**
P60034.	14	**Bytes/Row**
P60090.	Clock 16MHz	**Processor Clock Speed**
P60091.	0 1	**ROM Wait States**
P60092.	FastT 0 1	**RAM Wait States**
P60100.	©1995 *GSE*	**Copyright**
P60101.	0460-xxxxx	**Firmware Revision**
P60102.	mmm dd yyyy	**Firmware Date**
P60103.	Bxxxb010yy	**Flash Boot Block Revision**
P60104.	Bmmddyyyyx	**Flash Boot Block Date**
P60200.	xxxxx	**Circuit Board Serial Number**
P60201.	xxxxx	**OIML Audit Trail Counter**
P60202.	xxxxx	**Indicator Serial Number**
P60203.	xxxxx	**Calibration Audit Trail Counter**
P60204.	xxxxx	**Setup Audit Trail Counter**
P60205.	CHECK	**NTEP Compliance Parameters**
P61099.1 † P61099.2	1 2	**Scale# for P61110→ P61122** Enter 1→ 2
P61100.	mVv0.00000	**Current mV/V Output**
P61101. †	1.000000	**Calibration Factor**
P61102.	0.000000	**ReZero Weight** Press [CLR] to Reset
P61103.	0.000000	**Zero Track Weight** Press [CLR] to Reset
P61104.0 † P61104.1 P61104.2 P61104.3 P61104.4 P61104.5 P61104.6 P61104.7 P61104.8 P61104.9	0% 25% 50% 75% 100% 125% 150% 175% 0% -25%	**Coarse Zero Calibration** Enter 1→ 15

PARAMETER	SELECTION	DESCRIPTION (460 SERIES)
P61104.10 P61104.11 P61104.12 P61104.13 P61104.14 P61104.15	-50% -75% -100% -125% -150% -175%	
P61105.	0	**Fine Zero Calibration**
P61106.0 P61106.1 † P61106.2 P61106.3	25 50 100 200	**Coarse Gain Calibration** Enter 1→ 4
P61107.	1.000000	**Fine Gain Calibration**
P61110.	xxxxxx	**Zero Adjust 25** Enter Factory Determined Value
P61111.	xxxxxx	**Zero Adjust 50** Enter Factory Determined Value
P61112.	xxxxxx	**Zero Adjust 100** Enter Factory Determined Value
P61113.	x.xxxxxx	**Gain Adjust 1** Enter Factory Determined Value
P61114.	x.xxxxxx	**Gain Adjust 2** Enter Factory Determined Value
P61115.	x.xxxxxx	**Gain Adjust 4** Enter Factory Determined Value
P61116.	x.xxxxxx	**Gain Adjust 8** Enter Factory Determined Value
P61117.	xxxxxx	**Analog In 1 NR Offset** Enter Factory Determined Value
P61118.	xxxxxx	**Analog In 2 NR Offset** Enter Factory Determined Value
P61119.	xxxxxx	**Analog In 4 NR Offset** Enter Factory Determined Value
P61120.	xxxxxx	**Analog In 8 NR Offset** Enter Factory Determined Value
P61121.	xxxxxx	**Voltage Reference Comp.** Enter Factory Determined Value
P61122.	xxxxx	**Multi-Scale Serial Number** Enter Serial Number
P61123.	BckUp GOOD!	**A/D Calibration Backup**
P61130.	xxxxxx	**First Linearization Weight**
P61131.	xxxxxx	**First Linearization Factor**
P61132.	xxxxxx	**Second Linearization Weight**
P61133.	xxxxxx	**Second Linearization Factor**
P61134.	xxxxxx	**Third Linearization Weight**
P61135.	xxxxxx	**Third Linearization Factor**
P61136.	xxxxxx	**Fourth Linearization Weight**
P61137.	xxxxxx	**Fourth Linearization Factor**
P61138.	xxxxxx	**Fifth Linearization Weight**
P61139.	xxxxxx	**Fifth Linearization Factor**
P61150.	0	**Number of Load Cells**
P61151.	2.000000	**Load Cell #1 F.S mV/V Rating**
P61152.	2.000000	**Load Cell #2 F.S. mV/V Rating**
P61153.	2.000000	**Load Cell #3 F.S. mV/V Rating**
P61154.	2.000000	**Load Cell #4 F.S. mV/V Rating**
P61155.	2.000000	**Load Cell #5 F.S. mV/V Rating**
P61156.	2.000000	**Load Cell #6 F.S. mV/V Rating**
P61157.	2.000000	**Load Cell #7 F.S. mV/V Rating**
P61158.	2.000000	**Load Cell #8 F.S. mV/V Rating**
P61159.	100.0000	**Load Cell Full Scale Capacity**
P61160.	2.000000	**Average Load Cell mV/V Rating**
P61200.1 † P61200.2	1 2	**Analog# for P61201→ P61207** Enter 1→ 2
P61201.	xxxxx	**Analog Zero Ref. (0-10V)**
P61202.	xxxxx	**Analog Gain (0-10V)**
P61203.	xxxxx	**Analog Zero Ref. (0-20mA)**
P61204.	xxxxx	**Analog Gain (0-20mA)**
P61205.	xxxxx	**Analog Zero Ref. (4-20mA)**
P61206.	xxxxx	**Analog Gain (4-20mA)**
P61207.	xxxxx	**Analog Option Serial Number**
P62000.		**Display Test** Press [ENTER]
P62001.		**Optional RAM Test** Press [ENTER]

PARAMETER	SELECTION	DESCRIPTION (460 SERIES)
P62002.		**EEPROM (E^2) Test** Press [ENTER]
P62010.1 † P62010.2	1 2	**Analog Output Test for P62011 → P62013** Enter 1 → 2
P62011.		**0-10V Analog Output Test** Press [ENTER]
P62012.		**0-20mA Analog Output Test** Press [ENTER]
P62013.		**4-20mA Analog Output Test** Press [ENTER]
P64000.		**Send Setup** Press [ENTER]
P64001.		**Send Setup & Parameter Values** Press [ENTER]
P64100.	0	**Upload Line Count**
P64101.	0	**Upload Error Count**
P64102.	None!	**First Upload Error**
P64103.0 P64103.1 P64103.2	Off Comm1 Comm2	**Upload Debug Output** Enter 0 → 2
P65001.		**Complete Factory Default** Press [ENTER]
P65002.		**Factory Default (except CAL)** Press [ENTER]
P65003.		**Script File** Press [TARE], then [ENTER]
P65010.		**Database Reset** Press [ENTER]
P65020.		**New Program** Press [ENTER], then COMM#

Table D-2: 560 Series Parameter Setup

PARAMETER	SELECTION	DESCRIPTION (560 SERIES)
Scale Configuration		
P108.01	Scale 1	**Scale Instance Selection**
P108.02	Scale 2	Enter 1 → 4
P108.03	Scale 3	
P108.04	Scale 4	
P109.00	Disabled	**Scale Enable**
P109.01	Saved	Enter 0 → 2
P109.02 †	Enabled	
P110. †	100.00	**Full Scale Capacity** Enter 0.01 → 1,000,000
P111.00	.00001	**Division Size**
P111.01	.00002	Enter 0 → 23
P111.02	.00005	
P111.03	.0001	
P111.04	.0002	
P111.05	.0005	
P111.06	.001	
P111.07	.002	
P111.08	.005	
P111.09 †	.01	
P111.10	.02	
P111.11	.05	
P111.12	.1	
P111.13	.2	
P111.14	.5	
P111.15	1	
P111.16	2	
P111.17	5	
P111.18	10	
P111.19	20	
P111.20	50	
P111.21	100	
P111.22	200	
P111.23	500	
P112.00	OFF	**Zero Track Divisions**
P112.01	0.1d	Enter 0 → 200
P112.02	0.2d	
P112.03	0.3d	
P112.04	0.4d	
P112.05	0.5d	
P112.06	0.6d	
P112.07	0.7d	
P112.08	0.8d	
P112.09	0.9d	
P112.10 †	1.0d	
↓	↓	
P112.200	20.0d	
P113.00	0.05s	**Zero Track Delay (seconds)**
P113.01	0.1s	Enter 0 → 100
P113.02	0.2s	
P113.03	0.3s	
P113.04	0.4s	
P113.05 †	0.5s	
↓	↓	
P113.100	10.0s	
P114.00	OFF	**Motion Divisions**
P114.01	0.1d	Enter 0 → 200
P114.02	0.2d	
P114.03	0.3d	
P114.04	0.4d	
P114.05	0.5d	
P114.06	0.6d	
P114.07	0.7d	
P114.08	0.8d	
P114.09	0.9d	
P114.10 †	1.0d	
↓	↓	
P114.200	20.0d	
P115.00	0.05s	**Motion Delay (seconds)**

PARAMETER	SELECTION	DESCRIPTION (560 SERIES)
P115.01	0.1s	Enter 0 → 100
P115.02	0.2s	
P115.03	0.3s	
P115.04	0.4s	
P115.05	0.5s	
P115.06	0.6s	
P115.07	0.7s	
P115.08	0.8s	
P115.09	0.9s	
P115.10 †	1.0s	
↓	↓	
P115.100	10.0s	
P116.00	0.06s	**Digital Filter**
P116.01	0.13s	Enter 0 → 11
P116.02	0.25s	
P116.03	0.50s	
P116.04	1.0s	
P116.05	2.0s	
P116.06	4.0s	
P116.07	8.0s	
P116.08	2.0sA	
P116.09 †	4.0sA	
P116.10	8.0sA	
P116.11	OFF	
P117.00 †	0.05s	**Display Update Rate (seconds)**
P117.01	0.1s	Enter 0 → 200
P117.02	0.2s	
↓	↓	
P117.200	20.0s	
P118.00	0.01%	**Zero Range**
P118.01	0.02%	Enter 0 → 12
P118.02	0.04%	
P118.03	0.1%	
P118.04	0.2%	
P118.05	0.4%	
P118.06	1.0%	
P118.07	2.0%	
P118.08	4.0%	
P118.09	10.0%	
P118.10	20.0%	
P118.11	40.0%	
P118.12 †	100%	
P119.00 †	Disabled	**Linearization**
P119.01	Enabled	Enter 0 → 1
P122.00	0.01%	**Return To Zero (RTZ)**
P122.01	0.02%	Enter 0 → 12
P122.02	0.04%	
P122.03 †	0.1%	
P122.04	0.2%	
P122.05	0.4%	
P122.06	1.0%	
P122.07	2.0%	
P122.08	4.0%	
P122.09	10.0%	
P122.10	20.0%	
P122.11	40.0%	
P122.12	100%	
P124. †	0.000	**Count Resolution** Enter 100 → 1,000,000
P125. †	1.000	**Count Adjustment Factor** Enter 0.1 → 20.0
P126. †	0.000	**Low Range Capacity** Enter 0.001 → 1,000,000
P127.00	.00001	**Low Range Division Size**

PARAMETER	SELECTION	DESCRIPTION (560 SERIES)
P127.01	.00002	Enter 0 → 23
P127.02	.00005	
P127.03	.0001	
P127.04	.0002	
P127.05	.0005	
P127.06	.001	
P127.07	.002	
P127.08	.005	
P127.09 †	.01	
P127.10	.02	
P127.11	.05	
P127.12	.1	
P127.13	.2	
P127.14	.5	
P127.15	1	
P127.16	2	
P127.17	5	
P127.18	10	
P127.19	20	
P127.20	50	
P127.21	100	
P127.22	200	
P127.23	500	
P128.	0.000	**Middle Range Capacity** Enter 0.01 → 1,000,000
P127.00	.00001	**Middle Range Division Size** Enter 0 → 23
P127.01	.00002	
P127.02	.00005	
P127.03	.0001	
P127.04	.0002	
P127.05	.0005	
P127.06	.001	
P127.07	.002	
P127.08	.005	
P127.09 †	.01	
P127.10	.02	
P127.11	.05	
P127.12	.1	
P127.13	.2	
P127.14	.5	
P127.15	1	
P127.16	2	
P127.17	5	
P127.18	10	
P127.19	20	
P127.20	50	
P127.21	100	
P127.22	200	
P127.23	500	
P130.00 †	Gross	**Multi-Range Mode** Enter 0 → 1
P130.01	Net	
P131.00 †	Pounds	**First Units** Enter 0 → 8
P131.01	Kilograms	
P131.02	Ounces	
P131.03	Grams	
P131.04	US Tons	
P131.05	Metric Tons	
P131.06	Custom Unit1	
P131.07	Custom Unit2	
P131.08	Pounds-Ounces	
P132.00	Pounds	**Second Units** Enter 0 → 9
P132.01 †	Kilograms	
P132.02	Ounces	
P132.03	Grams	
P132.04	US Tons	
P132.05	Metric Tons	
P132.06	Custom Unit1	
P132.07	Custom Unit2	
P132.08	Pounds-Ounces	
P132.09	NONE	
P133.00	Pounds	**Third Units**

PARAMETER	SELECTION	DESCRIPTION (560 SERIES)
P133.01	Kilograms	Enter 0 → 9
P133.02	Ounces	
P133.03	Grams	
P133.04	US Tons	
P133.05	Metric Tons	
P133.06	Custom Unit1	
P133.07	Custom Unit2	
P133.08	Pounds-Ounces	
P133.09 †	NONE	
P134.00	Pounds	**Fourth Units** Enter 0 → 9
P134.01	Kilograms	
P134.02	Ounces	
P134.03	Grams	
P134.04	US Tons	
P134.05	Metric Tons	
P134.06	Custom Unit1	
P134.07	Custom Unit2	
P134.08	Pounds-Ounces	
P134.09 †	NONE	
P135. †	Off	**Rate Measurement Period** Enter 0 → 900s
P136.00 †	Seconds	**Rate Time Unit** Enter 0 → 2
P136.01	Minutes	
P136.02	Hours	
P142.00	Disabled	**Center-of-Zero Annunciator** Enter 0 → 1
P142.01 †	Enabled	
P143. †	O	**Overload Status Name** Enter Name
P144. †	M	**Motion Status Name** Enter Name
P145. †	S	**Stable Status Name** Enter Name
P146. †	O	**Underload Status Name** Enter Name
P147. †	E	**Error Status Name** Enter Name
Units		
P150.00 †	Pounds	**Default Units** Enter 1 → 6
P150.01	Kilograms	
P150.02	Ounces	
P150.03	Grams	
P150.04	US Tons	
P150.05	Metric Tons	
P150.06	Custom Unit1	
P151. †	????1	**Custom Unit1 Name** Enter Name
P152. †	1.000	**Custom Unit1 Calibration Factor** Enter 0.000001 → 9,999,999
P153. †	????2	**Custom Unit 2 Name** Enter Name
P154. †	1.000	**Custom Unit2 Calibration Factor** Enter 0.000001 → 9,999,999
Tare Functions		
P162.00 †	Disabled	**Negative Tare Enable** Enter 0 → 1
P162.01	Enabled	
P163.00	Disabled	**Tare Rounding Enable** Enter 0 → 1
P163.01 †	Enabled	
Analog Output		
P170.01	Analog Out 1	**Analog Output Instance Selection** Enter 1 → 4
P170.02	Analog Out 2	
P170.03	Analog Out 3	
P170.04	Analog Out 4	
P171.00 †	Disabled	**Analog Output Enable** Enter 0 → 1
P171.01	Enabled	
P172. †	Gross	**Output Parameter** Enter Operating Parameter
P173. †	None!	**Full Scale Output** Enter Operating Parameter
P174. †	None!	**Zero Offset** Enter Operating Parameter
P175. †	None!	**Output Signal Range** Enter Operating Parameter
P176.00 †	Maximum	**Default Output in Setup Mode**

PARAMETER	SELECTION	DESCRIPTION (560 SERIES)
P176.01 P176.02	Minimum Same	Enter 0→ 2
P177.00 † P177.01 P177.02	0-10 VDC 0-20 mA 4-20 mA	**Analog Output Type** Enter 0→ 2
Counting		
P179.00 † P179.01	Disabled Enabled	**Count Enable** Enter 0→ 1
P180.00 † P180.01	off on	**Auto Sample Enable** Enter 0→ 1
P181.00 P181.01 †	off on	**Auto Enhance Enable** Enter 0→ 1
P182.01 P182.02 P182.03 P182.04 P182.05 P182.06 P182.07 P182.08 P182.09 P182.10 † ↓ P182.9999	1 piece 2 pieces 3 pieces 4 pieces 5 pieces 6 pieces 7 pieces 8 pieces 9 pieces 10 pieces ↓ 9999 pieces	**Default Sample Size** Enter 1→ 9999
P183. †	98.52%	**Required Accuracy** Enter 0, 90.00 → 99.96
P184.00 † P184.01	off on	**Accuracy Display Enable** Enter 0→ 1
P185.00 † P185.01 P185.02 P185.03 P185.04	None! Scale 1 Scale 2 Scale 3 Scale 4	**Pre-Sample Scale** Enter 0→ 4
P186.00 † P186.01 P186.02 P186.03 P186.04	None! Scale 1 Scale 2 Scale 3 Scale 4	**After-Sample Scale** Enter 0→ 4
P187.00 P187.01 P187.02 P187.03 P187.04 P187.05 P187.06 † P187.07	None! 0.13s 0.25s 0.50s 1.0s 2.0s 4.0s 8.0s	**Sample Filter** Enter 0→ 7
P188.00 P188.01	off on	**Enforce Sample Accuracy** Enter 0→ 1
P189.00 † P189.01 P189.02 ↓ P189.15	0.0d 0.1d 0.2d ↓ 1.5d	**Sample Motion Divisions** Enter 1→ 15
Communication Ports		
P199.01 P199.02 P199.03	Comm Port 1 Comm Port 2 Comm Port 3	**Serial Port Instance Selection** Enter 1→ 3
P200.00 P200.01 * P200.02 † P200.03 P200.04 P200.05 P200.06 P200.07 P200.08 P200.12 P200.13 **	37400 19200 9600 4800 2400 1200 600 300 150 58300 38400	**Baud Rate** Enter 0→ 13 for Comm 1 only Enter 0→ 8 or 13 for all others ** *Alternate processor clock speed will be automatically selected – 14MHz*
P201.00 P201.01 †	7 bits 8 bits	**Data Bits** Enter 0→ 1
P202.00 † P202.01 P202.02	None Even Odd	**Parity** Enter 0→ 2
P203.00 † P203.01	1 bit 2 bits	**Stop Bits** Enter 0→ 1

PARAMETER	SELECTION	DESCRIPTION (560 SERIES)
P204.00 P204.01 P204.02 † P204.03	None CTS/RTS Xon/Xoff Both	**Handshaking** Enter 0→ 3
P205.00 P205.01 † P205.02 P205.03 P205.04 P205.05 P205.06 ↓ P205.100	Disabled Standard Interpreter Modbus Macro 4 Macro 5 Macro 6 ↓ Macro 100	**Receive Mode** Enter 0→ 250
P206.00 † P206.01	Delay Abort	**Transmit Mode** Enter 0→ 1
P207.08 P207.09 P207.10 ↓ P207.512 † ↓ P207.9999	8 bytes 9 bytes 10 bytes ↓ 512 bytes ↓ 9999 bytes	**Transmit Buffer Size (bytes)** Enter 8→ 3967 *Maximum combined value for Transmit and Receive buffers of all ports is 3967 bytes.*
P208.08 P208.09 P208.10 ↓ P208.1024† ↓ P207.9999	8 bytes 9 bytes 10 bytes ↓ 1024 bytes ↓ 9999 bytes	**Receive Buffer Size (bytes)** Enter 8→ 3967 *Maximum combined value for Transmit and Receive buffers of all ports is 3967 bytes.*
P209.01 P209.02 P209.03 ↓ P209.247	1 2 3 ↓ 247	**Modbus Address** Enter 1→ 247
P210.00 † P210.01	ASCII RTU	**Modbus Mode** Enter 0→ 1
P210.00 † P210.01	ASCII RTU	**Modbus Mode** Enter 0→ 1
P211.00 † P211.01	Hi/Lo Lo/Hi	**Modbus Word Hi/Lo** Enter 0→ 1
Input Interpreter		
P217.00 † P217.01	Disabled Enabled	**Interpreter NULL** Enter 0→ 1
P218.00 P218.01 P218.02 ↓ P218.10 † ↓ P218.255	<NUL> <SOH> <STX> ↓ <LF> ↓ <255>	**Receive Termination Character** Enter ASCII Value .000→ .255
P219.00 † P219.01 P219.02 P219.03 ↓ P219.100	None! Interp. 1 Interp. 2 Interp. 3 ↓ Interp. 250	**Input Interpreter Instance Selection** Enter 1→ 100
P220. †	None!	**Interpreter Name** Enter Name
P221.00 † P221.01	Character Line	**Interpreter Type** Enter 0→ 1
P222.00		**Line Interpreter Entry Table** Enter text, operating parameters, control codes
P223.00 † P223.01 P223.02 ↓ P218.255	<NUL> <SOH> <STX> ↓ <255>	**Interpreter Character** Enter ASCII Value .000→ .255
P224.00 † P224.01 P224.02 P224.03 ↓ P224.100	None! Macro 1 Macro 2 Macro 3 ↓ Macro 100	**Interpreter Macro#** Enter 1→ 100
Numeric Parameter Formatting		

PARAMETER	SELECTION	DESCRIPTION (560 SERIES)
P240.01 P240.02 P240.03 P240.04 P240.05 P240.06 P240.07 P240.08 † ↓ P240.15	1 2 3 4 5 6 7 8 ↓ 15	**Minimum Transmit Width** Enter 1 → 15
P241.00 † P241.01	Right Left	**Sign Justification** Enter 0 → 1
Networking & Remote Communications		
P250.00 † P250.01	Disabled Enabled	**Network Enable** Enter 0 → 1
P251.00 † P251.04 P251.05 P251.06 ↓ P251.254	Disabled 4 5 6 ↓ 254	**Network Address** Enter 0, 4 → 254
P290.00 † P290.01 P290.02 P290.03 P290.04	Off Comm Port 1 Comm Port 2 Comm Port 3 Comm Port 4	**Echo Display** Enter 0 → 4
P291.00 P291.01 P291.02 † P291.03 ↓ P291.255	<NUL> <SOH> <STX> <ETX> ↓ <255>	**Echo Start Character** Enter ASCII Value 0 → 255
P292.00 P292.01 P292.02 P292.03 † ↓ P292.255	<NUL> <SOH> <STX> <ETX> ↓ <255>	**Echo End Character** Enter ASCII Value 0 → 255
P293.00 † P293.01 P293.02	Disabled LCD LED	**Remote Display** Enter 0 → 2
P294.00 † P294.01	Disabled Enabled	**LCD Remote Display Backlight** Enter 0 → 1
Weigh Mode Parameter Selections		
P300. †	Gross	**[SELECT] Mode 0** Enter Operating Parameter
P301. †	Net	**[SELECT] Mode 1** Enter Operating Parameter
P302. †	Tare	**[SELECT] Mode 2** Enter Operating Parameter
P303. †	None!	**[SELECT] Mode 3** Enter Operating Parameter
P304. †	None!	**[SELECT] Mode 4** Enter Operating Parameter
P305. †	None!	**[SELECT] Mode 5** Enter Operating Parameter
P306. †	None!	**[SELECT] Mode 6** Enter Operating Parameter
P307. †	None!	**[SELECT] Mode 7** Enter Operating Parameter
P308. †	None!	**[SELECT] Mode 8** Enter Operating Parameter
P309. †	None!	**[SELECT] Mode 9** Enter Operating Parameter
Access Codes		
P400. †	None!	**Personal Identification Number** Enter Code
P401. †	None!	**Quick Calibration Access Code** Enter Code
P402. †	21353	**Limited Access Code** Enter Code
OIML		
P410. †	Disabled	**OIML Enable**

PARAMETER	SELECTION	DESCRIPTION (560 SERIES)
		Enter 9990, 9991
P411.00 † P411.01 P411.02 P411.03 P411.04 P411.05 P411.06 P411.07 P411.08 P411.09 P411.10 P411.11 P411.12	USA France German UK Denmark Sweden Italy Spain Japan Norway Denmark2 Spain2 Latin America	**Language Character Set** Enter 0 → 12
P412.00 † P412.01	Disable Enable	**Preset Enable** Enter 0 → 1
VFD/LCD Setup		
P420.00 P420.01 † P420.02	Off On Auto	**Standard VF Display Mode** Enter 0 → 2
P421.00 P421.01 P421.02 † P421.03 P421.04 P421.05 P421.06 P421.07 P421.08 P421.09 P421.10 P421.11 P421.12 P421.13 P421.14 P421.15	2d 4d 6d 8d 10d 12d 14d 16d 18d 20d 22d 24d 26d 28d 30d 32d	**Weight Threshold Divisions** Enter 0 → 15
P422.00 P422.01 P422.02 P422.03 P422.04 P422.05 † P422.06 P422.07 P422.08 P422.09 P422.10 P422.11 P422.12 P422.13 P422.14 P422.15	30 seconds 1 minutes 2 minutes 3 minutes 4 minutes 5 minutes 10 minutes 15 minutes 20 minutes 25 minutes 30 minutes 35 minutes 40 minutes 45 minutes 1 hour 2 hours	**Timeout** Enter 0 → 15
P423.01 P423.02 P423.03 P423.04 P423.05 P423.06 P423.07 P423.08 P423.09 P423.10 †	10% 20% 30% 40% 50% 60% 70% 80% 90% 100%	**Display 'ON' Brightness** Enter 1 → 10
P424.00 † P424.01 P424.02 P424.03 P424.04 P424.05 P424.06 P424.07 P424.08 P424.09 P424.10	Off 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%	**Display 'OFF' Dimness** Enter 0 → 10
P425.00 † P425.01	Disabled Enabled	**4X20 VF Display Enable** Enter 0 → 1
NTEP		
P440.00 †	Disabled	**NTEP Enable**

PARAMETER	SELECTION	DESCRIPTION (560 SERIES)
P440.01	Enabled	Enter 0→ 1
Keypad		
P450.00 †	22 Key	**Keypad Selection**
P451.00 P451.01 P451.02 P451.03 P451.04 P451.05 P451.06 † P451.07	None Very Slow Slow Medium Slow Medium Medium Fast Fast Very Fast	**Keypad Repeat Rate** Enter 0→ 7
P460.00 P460.01 P460.02 P460.03 † P460.04 P460.05 P460.06 P460.07	Off Minimum Extra Low Low Medium Medium High High Maximum	**Beeper Volume** Enter 0→ 7
Time & Date		
P500. †	00:00	**Time** Enter 24 Hour Time (hh.mm.ss)
P501.70 †	01/01/70	**Date** Enter Date (mm.dd.yy)
P502.00 † P502.01	Disabled Enabled	**Time/Date Access** Enter 0→ 1
P503.00 P503.01 †	No Yes	**AM/PM Time Format** Enter 0→ 1
P504.00 † P504.01	U.S.A. International	**Date Format** Enter 0→ 1
DSD Parameters		
P590.00 † P590.01	Disable Enable	**DSD Enable** Enter 0→ 1
P591.00 † P591.01 P591.02 P591.03	None! Comm 1 Comm 2 Comm 3	**DSD Communication Port** Enter 0→ 3
P592.00 P592.01 P592.02 ↓ P592.82 † ↓ P592.255	<NUL> <SOH> <STX> ↓ "R" ↓ <255>	**DSD Receive Character** Enter ASCII Value .000→ .255
P593.00 † P593.01 ↓ P593.100	None! Transmit 1 ↓ Transmit 100	**DSD Custom Transmit** Enter 1→ 100
P594.00 †	0	**DSD Maximum Number of Rows** Enter 0→ Max Available
P595.00 †	0	**DSD Number of Rows Warning** Enter 0→ Max Rows @ P594
Parameter Renaming		
P600. †	None!	**Rename Gross** Enter Name
P601. †	None!	**Rename Net** Enter Name
P602. †	None!	**Rename Tare** Enter Name
P603. †	None!	**Rename Gross Total** Enter Name
P604. †	None!	**Rename Gross Total + Current** Enter Name
P605. †	None!	**Rename Gross Total – Current** Enter Name
P606. †	None!	**Rename Net Total** Enter Name
P607. †	None!	**Rename Net Total + Current** Enter Name
P608. †	None!	**Rename Net Total – Current** Enter Name
P609. †	None!	**Rename Accumulation** Enter Name
P610. †	None!	**Rename Scale#**

PARAMETER	SELECTION	DESCRIPTION (560 SERIES)
		Enter Name
P611. †	None!	**Rename Time/Date** Enter Name
P615. †	None!	**Rename Average Gross** Enter Name
P616. †	None!	**Rename Average Net** Enter Name
P618. †	None!	**Rename Peak Gross** Enter Name
P619. †	None!	**Rename Peak Net** Enter Name
P620. †	None!	**Rename Rounded Gross** Enter Name
P621. †	None!	**Rename Rounded Net** Enter Name
P623. †	None!	**Rename Rate** Enter Name
P624. †	None!	**Rename Free Fall 1** Enter Name
P625. †	None!	**Rename Future Gross 1** Enter Name
P626. †	None!	**Rename Future Net 1** Enter Name
P627. †	None!	**Rename Free Fall 2** Enter Name
P628. †	None!	**Rename Future Gross 2** Enter Name
P629. †	None!	**Rename Future Net 2** Enter Name
P630. †	None!	**Rename Quantity** Enter Name
P631. †	None!	**Rename Quantity Total** Enter Name
P632. †	None!	**Rename Quantity Total+Current** Enter Name
P633. †	None!	**Rename Quantity Total - Current** Enter Name
P634. †	None!	**Rename Average Piece Weight** Enter Name
P635. †	None!	**Rename Avg Piece Weight x K** Enter Name
P636. †	None!	**Rename Percent Accuracy** Enter Name
P637. †	None!	**Rename Sample** Enter Name
P640. †	None!	**Rename Grs Total of All Scales** Enter Name
P641. †	None!	**Rename Net Total of All Scales** Enter Name
P642. †	None!	**Rename Tare Total of All Scales** Enter Name
P643. †	None!	**Rename Total of All Grs Totals** Enter Name
P644. †	None!	**Rename Total of All Net Totals** Enter Name
P645. †	None!	**Rename Qty Total of All Scales** Enter Name
P646. †	None!	**Rename Total of All Qty Totals** Enter Name
Total & Tare Save		
P660.00 P660.01 P660.02 †	No Save On Request Auto Save	**Total Values Save Method** Enter 0→ 2
P661.00 P661.01 P661.02 †	No Save On Request Auto Save	**Tare Value Save Method** Enter 0→ 2
Variables		
P680.00 † P680.01	None! 1	**Variable Allocation** Enter 0→ 100

PARAMETER	SELECTION	DESCRIPTION (560 SERIES)
P680.02 P680.03 ↓ P680.100	2 3 ↓ 100	
P681.01 P681.02 P681.03 ↓ P681.	Variable 1 Variable 2 Variable 3 ↓ Variable 100	**Variable Instance Selection**
P682. †	None!	**Variable Name** Enter Name
P684.00 † P684.01 P684.02	No Save On Request Auto Save	**Variable Value Save Method** Enter 0→ 2
P685.00 † P685.01	Disabled Enabled	**Variable Lock** Enter 0→ 1
P686.00 † P686.01 P686.02 P686.03	Float Integer Unsigned Int String	**Variable Type** Enter 0→ 3
P687.00 P687.01 P687.02 P687.03 P687.04 P687.05 P687.06 † P687.07 P687.08 P687.09 P687.10	0 decimal pt 1 decimal pt 2 decimal pt 3 decimal pt 4 decimal pt 5 decimal pt Auto decimal Scale 1 Scale 2 Scale 3 Scale 4	**Float Style** Enter 0→ 14
P688.00 † P688.01 P688.02 P688.03	Number Time/Date Time Date	**Integer Style**
P689.01 P689.02 P689.03 P698.04 P698.05 P698.06 P698.07 P698.08 P698.09 P698.10 † ↓ P689.63	1 character 2 characters 3 characters 4 characters 5 characters 6 characters 7 characters 8 characters 9 characters 10 characters ↓ 63 characters	**String Size** Enter 1→ 63
Database		
P699.00 † P699.01 P699.02 P699.03 ↓ P699.100	None! Database 1 Database 2 Database 3 ↓ Database 100	**Database Instance Selection** Enter 1→ 100
P700. †	None!	**Database Name** Enter Name
P701. P702. P703. ↓ P798.	Column 1 Column 2 Column 3 ↓ Column 98	**Database Column Parameter** Enter Operating Parameter
Keypad Key Assignments		
P800.00 † P800.01 P800.02 P800.03 ↓ P800.100	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 100	**[SELECT] Key Function** Enter 0→ 100
P801.00 † P801.01 P801.02 P801.03 ↓ P801.100	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 100	**[ZERO] Key Function** Enter 0→ 100
P802.00 † P802.01 P802.02 P802.03	Enabled Macro 1 Macro 2 Macro 3	**[TARE] Key Function** Enter 0→ 100

PARAMETER	SELECTION	DESCRIPTION (560 SERIES)
↓ P802.100	↓ Macro 100	
P803.00 † P803.01 P803.02 P803.03 ↓ P803.100	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 100	**[UNITS] Key Function** Enter 0→ 100
P804.00 † P804.01 P804.02 P804.03 ↓ P804.100	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 100	**[SCALE SELECT] Key Function** Enter 0→ 100
P805.00 † P805.01 P805.02 P805.03 ↓ P805.100	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 100	**[PRINT] Key Function** Enter 0→ 100
P806.00 † P806.01 P806.02 P806.03 P806.04 P806.05 P806.06 ↓ P806.100 P806.101	None Menu Database Menu & Dbase Macro 4 Macro 5 Macro 6 ↓ Macro 100 Macro 101	**[ID] Key Function** Enter 0→ 100 101 = Macro 3
P807.00 † P807.01 P807.02 P807.03 ↓ P807.100	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 100	**[ENTER] Key Function** Enter 0→ 100
P808.00 † P808.01 P808.02 P808.03 ↓ P808.100	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 100	**[CLEAR] Key Function** Enter 0→ 100
P809.00 † P809.01 P809.02 P809.03 ↓ P809.100	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 100	**[.] Key Function** Enter 0→ 100
P810.00 † P810.01 P810.02 P810.03 ↓ P810.100	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 100	**[0] Key Function** Enter 0→ 100
P811.00 † P811.01 P811.02 P811.03 ↓ P811.100	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 100	**[1] Key Function** Enter 0→ 100
P812.00 † P812.01 P812.02 P812.03 ↓ P812.100	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 100	**[2] Key Function** Enter 0→ 100
P813.00 † P813.01 P813.02 P813.03 ↓ P813.100	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 100	**[3] Key Function** Enter 0→ 250
P814.00 † P814.01 P814.02 P814.03	Enabled Macro 1 Macro 2 Macro 3	**[4] Key Function** Enter 0→ 100

PARAMETER	SELECTION	DESCRIPTION (560 SERIES)
↓ P814.250	↓ Macro 250	
P815.00 † P815.01 P815.02 P815.03 ↓ P815.100	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 100	**[5] Key Function** Enter 0→ 100
P816.00 † P816.01 P816.02 P816.03 ↓ P816.100	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 100	**[6] Key Function** Enter 0→ 100
P817.00 † P817.01 P817.02 P817.03 ↓ P817.100	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 100	**[7] Key Function** Enter 0→ 100
P818.00 † P818.01 P818.02 P818.03 ↓ P818.100	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 100	**[8] Key Function** Enter 0→ 100
P819.00 † P819.01 P819.02 P819.03 ↓ P819.100	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 100	**[9] Key Function** Enter 0→ 100
P820.00 † P820.01 P820.02 P820.03 ↓ P820.100	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 100	**Any Key Function** Enter 0→ 100
Custom Transmit		
P980.00 P980.01 P980.02 P980.03 P980.04 P980.05 † ↓ P980.250	Disabled 0.6 seconds 0.7 seconds 0.8 seconds 0.9 seconds 0.10 seconds ↓ 25.0 seconds	**Continuous Transmit Rate** Enter 0→ 250
P989.01 † P989.02 P989.03 ↓ P989.100	Transmit 1 Transmit 2 Transmit 3 ↓ Transmit 100	**Custom Transmit Instance Selection** Enter 1→ 100
P990. †	None!	**Custom Transmit Name** Enter Name
P991.00 P991.01 † P991.02	Off On Request Prompt	**Transmit Mode** Enter 0→ 2
P992.01 † P992.02 P992.03	Comm Port 1 Comm Port 2 Comm Port 3	**Serial Port Selection** Enter 1→ 3
P993.00 † P993.01 *	Ignored Delayed	**Current Scale Motion** Enter 0→ 1 ** Default custom transmit #1 is motion delayed.*
P994. †	Ignored	**Scale Motion** Enter 0 for motion ignored Enter any combination of 1→ 4 for motion delay on entered scale numbers.
P995.00 † P995.01	Ignored Delayed	**Scale 2 Motion** See P994. Model 650 backward compatibility parameter.

PARAMETER	SELECTION	DESCRIPTION (560 SERIES)
P996.00 † P996.01	Ignored Delayed	**Scale 3 Motion** See P994. Model 650 backward compatibility parameter.
P997.00 † P997.01	Ignored Delayed	**Scale 4 Motion** See P994. Model 650 backward compatibility parameter.
P998.00 † P998.01	Disabled Enabled	**Continuous Transmit Enable** Enter 0→ 1
P999.00 † P999.01	No Yes	**Transmit Table Limited Access** Enter 0→ 1
P1000. P1001. P1002. ↓ P4999.		**Transmit Table Entry** Enter text, operating parameters, control codes
Setpoints		
P5099.1 P5099.2 P5099.3 ↓ P5099.48		**Setpoint Instance Selection** Enter 1→ 48
P5100.0 † P5100.1 P5100.2	Disabled Output Input	**Setpoint Mode** Enter 0→ 2
P5101. †	None!	**Setpoint Name** Enter Name
P5110.0 † P5110.1 P5110.2 P5110.3 P5110.4 P5110.5 P5110.6 P5110.7 P5110.8 P5110.9 P5110.10 P5110.11 P5110.12 P5110.13 P5110.14 P5110.15	Above Below Between Outside Always Never Motion Scale1 Motion Scale2 Motion Scale3 Motion Scale4 Stable Scale1 Stable Scale2 Stable Scale3 Stable Scale4 Mot'n Current Stabl Current	**Activation Condition** Enter 0→ 15
P5111. †	0.00 seconds	**Activation Delay (seconds)** Enter 0.01→ 5,767,168
P5112.0 † P5112.1 P5112.2 P5112.3 ↓ P5112.100	None! Macro 1 Macro 2 Macro 3 ↓ Macro 100	**Activation Macro#** Enter 0→ 100
P5113.0 † P5113.1	Ignored Delayed	**Activation Motion** Enter 0→ 1
P5114. †	None!	**Lower Activation Parameter** Enter Operating Parameter
P5115. †	None!	**Upper Activation Parameter** Enter Operating Parameter
P5130.0 † P5130.1 P5130.2 P5130.3 P5130.4 P5130.5	Above Below Between Outside Always Never	**Deactivation Condition** Enter 0→ 15

PARAMETER	SELECTION	DESCRIPTION (560 SERIES)
P5130.6 P5130.7 P5130.8 P5130.9 P5130.10 P5130.11 P5130.12 P5130.13 P5130.14 P5130.15	Motion Scale1 Motion Scale2 Motion Scale3 Motion Scale4 Stable Scale1 Stable Scale2 Stable Scale3 Stable Scale4 Mot'n Current Stabl Current	
P5131. †	0.00 seconds	**Deactivation Delay (seconds)** Enter 0.01 → 5,767,168
P5132.0 † P5132.1 P5132.2 P5132.3 ↓ P5132.100	None! Macro 1 Macro 2 Macro 3 ↓ Macro 100	**Deactivation Macro#** Enter 0 → 100
P5133.0 † P5133.1	Ignored Delayed	**Deactivation Motion** Enter 0 → 1
P5134. †	None!	**Lower Deactivation Parameter** Enter Operating Parameter
P5135. †	None!	**Upper Deactivation Parameter** Enter Operating Parameter
P5150. †	Gross	**Compare Parameter** Enter Operating Parameter
Modbus Parameter Map		
P6001. † P6002. P6003. ↓ P6247.	None!	**Modbus Address Translation Table** Enter Operating Parameter
Macros		
P9980.0 † P9980.1 P9980.2 P9980.3 ↓ P9980.100	None! Macro 1 Macro 2 Macro 3 ↓ Macro 100	**Abort Macro#** Enter 0 → 100
P9980.1 † P9981.1	Menu Immediate	**Macro Abort Method** Enter 0 → 1
P9990.0 P9990.1 P9990.2 P9990.3 ↓ P9990.100	None! Macro 1 Macro 2 Macro 3 ↓ Macro 100	**Macro Instance Selection** Enter 1 → 100
P9991. †	None!	**Macro Name** Enter Name
P9992.0 † P9992.1	Standard Immediate	**Macro Priority** Enter 0 → 1
P9993.0 † P9993.1	Disabled Enabled	**Macro Menu Enable** Enter 0 → 1
P9994.0 † P9994.1	No Yes	**Macro Table Limited Access** Enter 0 → 1
P10001. P10002. P10003. ↓ P19999.		**Macro Entry Table** Enter Text
P50000.0 † P50000.1	No Yes	**Macro Debug Limited Access** Enter 0 → 1
P50001.		**Macro Debug Table** Read-Only
Information & Diagnostic Parameters		
P60000. †	8192	**EEPROM (E^2) Memory Installed**
P60001.	7206	**EEPROM (E^2) Memory Available**
P60002. †	224K	**RAM Installed**
P60003.	192K	**RAM Dynamically Allocated**
P60004.	190K	**RAM Available**
P60005.	0	**Macro Trace Buffer Size (bytes)**
P60006.	AMD	**FRAM Type**
P60007.	1024K	**FRAM SIze**
P60008.	489K	**FRAM Available**

PARAMETER	SELECTION	DESCRIPTION (560 SERIES)
P60010.	256K	**Optional RAM Installed (K-bytes)**
P60011.	262144	**Optional RAM Installed (bytes)**
P60012.	262080	**Optional RAM Available (bytes)**
P60013.	64	**Optional RAM Used (bytes)**
P60014.	262076	**Optional RAM Block (bytes)**
P60018.	Good	**Optional RAM Battery Voltage**
P60020.	0	**Database RAM Usage**
P60021.	none	**Database Error**
P60030.0 † P60030.1 P60030.2 P60030.3 ↓ P60030.100	None! 1 2 3 ↓ 100	**Database Number** Enter 1 → 100
P60031. †	None!	**Database Name**
P60032.	0	**Number of Rows**
P60033.	28	**Total Bytes Used**
P60034.	14	**Bytes/Row**
P60040.0 P60040.1 P60040.2 P60040.3 ↓ P60040.255	Disabled 1K 2K 3K ↓ 255K	**Setup RAM (K-bytes)** Enter 1 → 255 * * Maximum entry is 1K less than RAM installed (P60010)
P60041.		**Setup RAM (bytes)**
P60042.		**Setup RAM Available (bytes)**
P60043.		**Setup RAM Used (bytes)**
P60050.0 P60050.14 P60050.15 P60050.16 ↓ P60050.255	Disabled 14K 15K 16K ↓ 255K	**Pile RAM (K-bytes)** Enter 1 → 255 * * Maximum entry is 1K less than RAM installed (P60010)
P60051.		**Pile RAM (bytes)**
P60052.		**Pile RAM Available (bytes)**
P60053.		**Pile RAM Used (bytes)**
P60090.	Clock 24MHz	**Processor Clock Speed**
P60091.	0 1	**ROM Wait States**
P60092.	FastT 0 1	**RAM Wait States**
P60100.	©1995 *GSE*	**Copyright**
P60101.	0560-xxxxx	**Firmware Revision**
P60102.	mmm dd yyyy	**Firmware Date**
P60103.	Bxxxb010yy	**Flash Boot Block Revision**
P60104.	Bmmddyyyyx	**Flash Boot Block Date**
P60200.	xxxxx	**Circuit Board Serial Number**
P60201.	xxxxx	**OIML Audit Trail Counter**
P60202.	xxxxx	**Indicator Serial Number**
P60203.	xxxxx	**Calibration Audit Trail Counter**
P60204.	xxxxx	**Setup Audit Trail Counter**
P60205.	CHECK	**NTEP Compliance Parameters**
P61099.1 † P61099.2 P61099.3 P61099.4	1 2 3 4	**Scale# for P61110 → P61122** Enter 1 → 4
P61100.	mVv0.00000	**Current mV/V Output**
P61101. †	1.000000	**Calibration Factor**
P61102.	0.000000	**ReZero Weight** Press [CLR] to Reset
P61103.	0.000000	**Zero Track Weight** Press [CLR] to Reset
P61104.0 † P61104.1 P61104.2 P61104.3 P61104.4 P61104.5	0% 25% 50% 75% 100% 125%	**Coarse Zero Calibration** Enter 1 → 15

PARAMETER	SELECTION	DESCRIPTION (560 SERIES)
P61104.6 P61104.7 P61104.8 P61104.9 P61104.10 P61104.11 P61104.12 P61104.13 P61104.14 P61104.15	150% 175% 0% -25% -50% -75% -100% -125% -150% -175%	
P61105.	0	**Fine Zero Calibration**
P61106.0 P61106.1 † P61106.2 P61106.3	25 50 100 200	**Coarse Gain Calibration** Enter 1 → 4
P61107.	1.000000	**Fine Gain Calibration**
P61110.	xxxxxx	**Zero Adjust 25** Enter Factory Determined Value
P61111.	xxxxxx	**Zero Adjust 50** Enter Factory Determined Value
P61112.	xxxxxx	**Zero Adjust 100** Enter Factory Determined Value
P61113.	x.xxxxxx	**Gain Adjust 1** Enter Factory Determined Value
P61114.	x.xxxxxx	**Gain Adjust 2** Enter Factory Determined Value
P61115.	x.xxxxxx	**Gain Adjust 4** Enter Factory Determined Value
P61116.	x.xxxxxx	**Gain Adjust 8** Enter Factory Determined Value
P61117.	xxxxxx	**Analog In 1 NR Offset** Enter Factory Determined Value
P61118.	xxxxxx	**Analog In 2 NR Offset** Enter Factory Determined Value
P61119.	xxxxxx	**Analog In 4 NR Offset** Enter Factory Determined Value
P61120.	xxxxxx	**Analog In 8 NR Offset** Enter Factory Determined Value
P61121.	xxxxxx	**Voltage Reference Comp.** Enter Factory Determined Value
P61122.	xxxxx	**Multi-Scale Serial Number** Enter Serial Number
P61123.	BckUp GOOD!	**A/D Calibration Backup**
P61130.	xxxxxx	**First Linearization Weight**
P61131.	xxxxxx	**First Linearization Factor**
P61132.	xxxxxx	**Second Linearization Weight**
P61133.	xxxxxx	**Second Linearization Factor**
P61134.	xxxxxx	**Third Linearization Weight**
P61135.	xxxxxx	**Third Linearization Factor**
P61136.	xxxxxx	**Fourth Linearization Weight**
P61137.	xxxxxx	**Fourth Linearization Factor**
P61138.	xxxxxx	**Fifth Linearization Weight**
P61139.	xxxxxx	**Fifth Linearization Factor**
P61150.	0	**Number of Load Cells**
P61151.	2.000000	**Load Cell #1 F.S mV/V Rating**
P61152.	2.000000	**Load Cell #2 F.S. mV/V Rating**
P61153.	2.000000	**Load Cell #3 F.S. mV/V Rating**
P61154.	2.000000	**Load Cell #4 F.S. mV/V Rating**
P61155.	2.000000	**Load Cell #5 F.S. mV/V Rating**
P61156.	2.000000	**Load Cell #6 F.S. mV/V Rating**
P61157.	2.000000	**Load Cell #7 F.S. mV/V Rating**
P61158.	2.000000	**Load Cell #8 F.S. mV/V Rating**
P61159.	100.0000	**Load Cell Full Scale Capacity**
P61160.	2.000000	**Average Load Cell mV/V Rating**
P61200.1 † P61200.2 P61200.3 P61200.4	1 2 3 4	**Analog# for P61201 → P61207** Enter 1 → 4
P61201.	xxxxx	**Analog Zero Ref. (0-10V)**
P61202.	xxxxx	**Analog Gain (0-10V)**
P61203.	xxxxx	**Analog Zero Ref. (0-20mA)**
P61204.	xxxxx	**Analog Gain (0-20mA)**
P61205.	xxxxx	**Analog Zero Ref. (4-20mA)**
P61206.	xxxxx	**Analog Gain (4-20mA)**
P61207.	xxxxx	**Analog Option Serial Number**
P62000.		**Display Test** Press [ENTER]
P62001.		**Optional RAM Test** Press [ENTER]
P62002.		**EEPROM (E^2) Test** Press [ENTER]
P62010.1 † P62010.2 P62010.3 P62010.4	1 2 3 4	**Analog Output Test for P62011 → P62013** Enter 1 → 4
P62011.		**0-10V Analog Output Test** Press [ENTER]
P62012.		**0-20mA Analog Output Test** Press [ENTER]
P62013.		**4-20mA Analog Output Test** Press [ENTER]
P64000.		**Send Setup** Press [ENTER]
P64001.		**Send Setup & Parameter Values** Press [ENTER]
P64100.	0	**Upload Line Count**
P64101.	0	**Upload Error Count**
P64102.	None!	**First Upload Error**
P64103.0 P64103.1 P64103.2 P64103.3	Off Comm1 Comm2 Comm3	**Upload Debug Output**
P65000.		**Copy EEPROM (E^2)** Press [ENTER]
P65001.		**Complete Factory Default** Press [ENTER]
P65002.		**Factory Default (except CAL)** Press [ENTER]
P65003.		**Script File** Press [TARE], then [ENTER]
P65010.		**Database Reset** Press [ENTER]
P62002.		**EEPROM (E^2) Test** Press [ENTER]

Table D-3: 660 Series Parameter Setup

PARAMETER	SELECTION	DESCRIPTION (660 SERIES)
Scale Configuration		
P108.01 P108.02 P108.03	Scale 1 Scale 2 Scale 3	**Scale Instance Selection** Enter 1 → 8

PARAMETER	SELECTION	DESCRIPTION (660 SERIES)
P108.04 P108.05 P108.06 P108.07 P108.08	Scale 4 Scale 5 Scale 6 Scale 7 Scale 8	
P109.00 P109.01 P109.02 †	Disabled Saved Enabled	**Scale Enable** Enter 0→ 2
P110. †	100.00	**Full Scale Capacity** Enter 0.01→ 1,000,000
P111.00 P111.01 P111.02 P111.03 P111.04 P111.05 P111.06 P111.07 P111.08 P111.09 † P111.10 P111.11 P111.12 P111.13 P111.14 P111.15 P111.16 P111.17 P111.18 P111.19 P111.20 P111.21 P111.22 P111.23	.00001 .00002 .00005 .0001 .0002 .0005 .001 .002 .005 .01 .02 .05 .1 .2 .5 1 2 5 10 20 50 100 200 500	**Division Size** Enter 0→ 23
P112.00 P112.01 P112.02 P112.03 P112.04 P112.05 P112.06 P112.07 P112.08 P112.09 P112.10 † ↓ P112.200	OFF 0.1d 0.2d 0.3d 0.4d 0.5d 0.6d 0.7d 0.8d 0.9d 1.0d ↓ 20.0d	**Zero Track Divisions** Enter 0→ 200
P113.00 P113.01 P113.02 P113.03 P113.04 P113.05 † ↓ P113.100	0.05s 0.1s 0.2s 0.3s 0.4s 0.5s ↓ 10.0s	**Zero Track Delay (seconds)** Enter 0→ 100
P114.00 P114.01 P114.02 P114.03 P114.04 P114.05 P114.06 P114.07 P114.08 P114.09 P114.10 † ↓ P114.200	OFF 0.1d 0.2d 0.3d 0.4d 0.5d 0.6d 0.7d 0.8d 0.9d 1.0d ↓ 20.0d	**Motion Divisions** Enter 0→ 200
P115.00 P115.01 P115.02 P115.03 P115.04 P115.05 P115.06	0.05s 0.1s 0.2s 0.3s 0.4s 0.5s 0.6s	**Motion Delay (seconds)** Enter 0→ 100

PARAMETER	SELECTION	DESCRIPTION (660 SERIES)
P115.07 P115.08 P115.09 P115.10 † ↓ P115.100	0.7s 0.8s 0.9s 1.0s ↓ 10.0s	
P116.00 P116.01 P116.02 P116.03 P116.04 P116.05 P116.06 P116.07 P116.08 P116.09 † P116.10 P116.11	0.06s 0.13s 0.25s 0.50s 1.0s 2.0s 4.0s 8.0s 2.0sA 4.0sA 8.0sA OFF	**Digital Filter** Enter 0→ 11
P117.00 † P117.01 P117.02 ↓ P117.200	0.05s 0.1s 0.2s ↓ 20.0s	**Display Update Rate (seconds)** Enter 0→ 200
P118.00 P118.01 P118.02 P118.03 P118.04 P118.05 P118.06 P118.07 P118.08 P118.09 P118.10 P118.11 P118.12 †	0.01% 0.02% 0.04% 0.1% 0.2% 0.4% 1.0% 2.0% 4.0% 10.0% 20.0% 40.0% 100%	**Zero Range** Enter 0→ 12
P119.00 † P119.01	Disabled Enabled	**Linearization** Enter 0→ 1
P122.00 P122.01 P122.02 P122.03 † P122.04 P122.05 P122.06 P122.07 P122.08 P122.09 P122.10 P122.11 P122.12	0.01% 0.02% 0.04% 0.1% 0.2% 0.4% 1.0% 2.0% 4.0% 10.0% 20.0% 40.0% 100%	**Return To Zero (RTZ)** Enter 0→ 12
P124. †	0.000	**Count Resolution** Enter 100→ 1,000,000
P125. †	1.000	**Count Adjustment Factor** Enter 0.1→ 20.0
P126. †	0.000	**Low Range Capacity** Enter 0.001→ 1,000,000
P127.00 P127.01 P127.02 P127.03 P127.04 P127.05 P127.06	.00001 .00002 .00005 .0001 .0002 .0005 .001	**Low Range Division Size** Enter 0→ 23

PARAMETER	SELECTION	DESCRIPTION (660 SERIES)
P127.07	.002	
P127.08	.005	
P127.09 †	.01	
P127.10	.02	
P127.11	.05	
P127.12	.1	
P127.13	.2	
P127.14	.5	
P127.15	1	
P127.16	2	
P127.17	5	
P127.18	10	
P127.19	20	
P127.20	50	
P127.21	100	
P127.22	200	
P127.23	500	
P128.	0.000	**Middle Range Capacity** Enter 0.01 → 1,000,000
P127.00	.00001	**Middle Range Division Size** Enter 0 → 23
P127.01	.00002	
P127.02	.00005	
P127.03	.0001	
P127.04	.0002	
P127.05	.0005	
P127.06	.001	
P127.07	.002	
P127.08	.005	
P127.09 †	.01	
P127.10	.02	
P127.11	.05	
P127.12	.1	
P127.13	.2	
P127.14	.5	
P127.15	1	
P127.16	2	
P127.17	5	
P127.18	10	
P127.19	20	
P127.20	50	
P127.21	100	
P127.22	200	
P127.23	500	
P130.00 †	Gross	**Multi-Range Mode** Enter 0 → 1
P130.01	Net	
P131.00 †	Pounds	**First Units** Enter 0 → 8
P131.01	Kilograms	
P131.02	Ounces	
P131.03	Grams	
P131.04	US Tons	
P131.05	Metric Tons	
P131.06	Custom Unit1	
P131.07	Custom Unit2	
P131.08	Pounds-Ounces	
P132.00	Pounds	**Second Units** Enter 0 → 9
P132.01 †	Kilograms	
P132.02	Ounces	
P132.03	Grams	
P132.04	US Tons	
P132.05	Metric Tons	
P132.06	Custom Unit1	
P132.07	Custom Unit2	
P132.08	Pounds-Ounces	
P132.09	NONE	
P133.00	Pounds	**Third Units** Enter 0 → 9
P133.01	Kilograms	
P133.02	Ounces	
P133.03	Grams	
P133.04	US Tons	
P133.05	Metric Tons	
P133.06	Custom Unit1	

PARAMETER	SELECTION	DESCRIPTION (660 SERIES)
P133.07	Custom Unit2	
P133.08	Pounds-Ounces	
P133.09 †	NONE	
P134.00	Pounds	**Fourth Units** Enter 0 → 9
P134.01	Kilograms	
P134.02	Ounces	
P134.03	Grams	
P134.04	US Tons	
P134.05	Metric Tons	
P134.06	Custom Unit1	
P134.07	Custom Unit2	
P134.08	Pounds-Ounces	
P134.09 †	NONE	
P135. †	Off	**Rate Measurement Period** Enter 0 → 900s
P136.00 †	Seconds	**Rate Time Unit** Enter 0 → 2
P136.01	Minutes	
P136.02	Hours	
P142.00	Disabled	**Center-of-Zero Annunciator** Enter 0 → 1
P142.01 †	Enabled	
P143. †	O	**Over/Under Load Status Name** Enter Name
P144. †	M	**Motion Status Name** Enter Name
P145. †	S	**Stable Status Name** Enter Name
P146. †	O	**Underload Status Name** Enter Name
P147. †	E	**Error Status Name** Enter Name
Units		
P150.00 †	Pounds	**Default Units** Enter 1 → 6
P150.01	Kilograms	
P150.02	Ounces	
P150.03	Grams	
P150.04	US Tons	
P150.05	Metric Tons	
P150.06	Custom Unit1	
P151. †	????1	**Custom Unit1 Name** Enter Name
P152. †	1.000	**Custom Unit1 Calibration Factor** Enter 0.000001 → 9,999,999
P153. †	????2	**Custom Unit 2 Name** Enter Name
P154. †	1.000	**Custom Unit2 Calibration Factor** Enter 0.000001 → 9,999,999
Tare Functions		
P162.00 †	Disabled	**Negative Tare Enable** Enter 0 → 1
P162.01	Enabled	
P163.00	Disabled	**Tare Rounding Enable** Enter 0 → 1
P163.01 †	Enabled	
Analog Output		
P170.01	Analog Out 1	**Analog Output Instance Selection** Enter 1 → 8
P170.02	Analog Out 2	
P170.03	Analog Out 3	
P170.04	Analog Out 4	
P170.05	Analog Out 5	
P170.06	Analog Out 6	
P170.07	Analog Out 7	
P170.08	Analog Out 8	
P171.00 †	Disabled	**Analog Output Enable** Enter 0 → 1
P171.01	Enabled	
P172. †	Gross	**Output Parameter** Enter Operating Parameter
P173. †	None!	**Full Scale Output** Enter Operating Parameter
P174. †	None!	**Zero Offset** Enter Operating Parameter
P175. †	None!	**Output Signal Range** Enter Operating Parameter
P176.00 †	Maximum	**Default Output in Setup Mode** Enter 0 → 2
P176.01	Minimum	

Parameter	Selection	Description (660 Series)
P176.02	Same	
P177.00 † P177.01 P177.02	0-10 VDC 0-20 mA 4-20 mA	**Analog Output Type** Enter 0→ 2
Counting		
P179.00 † P179.01	Disabled Enabled	**Count Enable** Enter 0→ 1
P180.00 † P180.01	off on	**Auto Sample Enable** Enter 0→ 1
P181.00 P181.01 †	off on	**Auto Enhance Enable** Enter 0→ 1
P182.01 P182.02 P182.03 P182.04 P182.05 P182.06 P182.07 P182.08 P182.09 P182.10 † ↓ P182.9999	1 piece 2 pieces 3 pieces 4 pieces 5 pieces 6 pieces 7 pieces 8 pieces 9 pieces 10 pieces ↓ 9999 pieces	**Default Sample Size** Enter 1→ 9999
P183. †	98.52%	**Required Accuracy** Enter 0, 90.00→ 99.96
P184.00 † P184.01	off on	**Accuracy Display Enable** Enter 0→ 1
P185.00 † P185.01 P185.02 P185.03 P185.04 P185.05 P185.06 P185.07 P185.08	None! Scale 1 Scale 2 Scale 3 Scale 4 Scale 5 Scale 6 Scale 7 Scale 8	**Pre-Sample Scale** Enter 0→ 8
P186.00 † P186.01 P186.02 P186.03 P186.04 P186.05 P186.06 P186.07 P186.08	None! Scale 1 Scale 2 Scale 3 Scale 4 Scale 5 Scale 6 Scale 7 Scale 8	**After-Sample Scale** Enter 0→ 8
P187.00 P187.01 P187.02 P187.03 P187.04 P187.05 P187.06 † P187.07	None! 0.13s 0.25s 0.50s 1.0s 2.0s 4.0s 8.0s	**Sample Filter** Enter 0→ 7
P188.00 P188.01	off on	**Enforce Sample Accuracy** Enter 0→ 1
P189.00 † P189.01 P189.02 ↓ P189.15	0.0d 0.1d 0.2d ↓ 1.5d	**Sample Motion Divisions** Enter 1→ 15
Communication Ports		
P199.01 P199.02 P199.03 P199.04	Comm Port 1 Comm Port 2 Comm Port 3 Comm Port 4	**Serial Port Instance Selection** Enter 1→ 4
P200.00 P200.01 * P200.02 † P200.03 P200.04 P200.05 P200.06	39300 19200 9600 4800 2400 1200 600	**Baud Rate** Enter 0→ 13 for Comm 1 only Enter 0→ 8 or 13 for all others ** Comm4 default* *** Alternate processor clock speed will be automatically*

Parameter	Selection	Description (660 Series)
P200.07 P200.08 P200.09 ** P200.10 P200.11 ** P200.12 P200.13 **	300 150 115K 112K 57600 56200 38400	*selected – 23MHz for 57600 baud, 21MHz for 115K and 38400 baud.*
P201.00 P201.01 †	7 bits 8 bits	**Data Bits** Enter 0→ 1
P202.00 † P202.01 P202.02	None Even Odd	**Parity** Enter 0→ 2
P203.00 † P203.01	1 bit 2 bits	**Stop Bits** Enter 0→ 1
P204.00 P204.01 P204.02 † P204.03	None CTS/RTS Xon/Xoff Both	**Handshaking** Enter 0→ 3
P205.00 P205.01 † P205.02 P205.03 P205.04 P205.05 P205.06 ↓ P205.250	Disabled Standard Interpreter Modbus Macro 4 Macro 5 Macro 6 ↓ Macro 250	**Receive Mode** Enter 0→ 250
P206.00 † P206.01	Delay Abort	**Transmit Mode** Enter 0→ 1
P207.08 P207.09 P207.10 ↓ P207.512 † ↓ P207.9999	8 bytes 9 bytes 10 bytes ↓ 512 bytes ↓ 9999 bytes	**Transmit Buffer Size (bytes)** Enter 8→ 16255 *Maximum combined value for Transmit and Receive buffers of all ports is 16384 bytes.*
P208.08 P208.09 P208.10 ↓ P208.1024† ↓ P207.9999	8 bytes 9 bytes 10 bytes ↓ 1024 bytes ↓ 9999 bytes	**Receive Buffer Size (bytes)** Enter 8→ 16255 *Maximum combined value for Transmit and Receive buffers of all ports is 16384 bytes.*
P209.01 P209.02 P209.03 ↓ P209.247	1 2 3 ↓ 247	**Modbus Address** Enter 1→ 247
P210.00 † P210.01	ASCII RTU	**Modbus Mode** Enter 0→ 1
P210.00 † P210.01	ASCII RTU	**Modbus Mode** Enter 0→ 1
P211.00 † P211.01	Hi/Lo Lo/Hi	**Modbus Word Hi/Lo** Enter 0→ 1
Input Interpreter		
P217.00 † P217.01	Disabled Enabled	**Interpreter NULL** Enter 0→ 1
P218.00 P218.01 P218.02 ↓ P218.10 † ↓ P218.255	<NUL> <SOH> <STX> ↓ <LF> ↓ <255>	**Receive Termination Character** Enter ASCII Value .000→ .255
P219.00 † P219.01 P219.02 P219.03 ↓ P219.250	None! Interp. 1 Interp. 2 Interp. 3 ↓ Interp. 250	**Input Interpreter Instance Selection** Enter 1→ 250
P220. †	None!	**Interpreter Name** Enter Name
P221.00 † P221.01	Character Line	**Interpreter Type** Enter 0→ 1
P222.00		**Line Interpreter Entry Table** Enter text, operating parameters, control codes

Parameter	Selection	Description (660 Series)
P223.00 † P223.01 P223.02 ↓ P218.255	<NUL> <SOH> <STX> ↓ <255>	**Interpreter Character** Enter ASCII Value .000→ .255
P224.00 † P224.01 P224.02 P224.03 ↓ P224.250	None! Macro 1 Macro 2 Macro 3 ↓ Macro 250	**Interpreter Macro#** Enter 1→ 250
Numeric Parameter Formatting		
P240.01 P240.02 P240.03 P240.04 P240.05 P240.06 P240.07 P240.08 † ↓ P240.15	1 2 3 4 5 6 7 8 ↓ 15	**Minimum Transmit Width** Enter 1→ 15
P241.00 † P241.01	Right Left	**Sign Justification** Enter 0→ 1
Networking & Remote Communications		
P250.00 † P250.01	Disabled Enabled	**Network Enable** Enter 0→ 1
P251.00 † P251.04 P251.05 P251.06 ↓ P251.254	Disabled 4 5 6 ↓ 254	**Network Address** Enter 0, 4→ 254
P290.00 † P290.01 P290.02 P290.03 P290.04	Off Comm Port 1 Comm Port 2 Comm Port 3 Comm Port 4	**Echo Display** Enter 0→ 4
P291.00 P291.01 P291.02 † P291.03 ↓ P291.255	<NUL> <SOH> <STX> <ETX> ↓ <255>	**Echo Start Character** Enter ASCII Value 0→ 255
P292.00 P292.01 P292.02 P292.03 † ↓ P292.255	<NUL> <SOH> <STX> <ETX> ↓ <255>	**Echo End Character** Enter ASCII Value 0→ 255
P293.00 † P293.01 P293.02	Disabled LCD LED	**Remote Display** Enter 0→ 2
P294.00 † P294.01	Disabled Enabled	**LCD Remote Display Backlight** Enter 0→ 1
Weigh Mode Parameter Selections		
P300. †	Gross	**[SELECT] Mode 0** Enter Operating Parameter
P301. †	Net	**[SELECT] Mode 1** Enter Operating Parameter
P302. †	Tare	**[SELECT] Mode 2** Enter Operating Parameter
P303. †	None!	**[SELECT] Mode 3** Enter Operating Parameter
P304. †	None!	**[SELECT] Mode 4** Enter Operating Parameter
P305. †	None!	**[SELECT] Mode 5** Enter Operating Parameter
P306. †	None!	**[SELECT] Mode 6** Enter Operating Parameter
P307. †	None!	**[SELECT] Mode 7** Enter Operating Parameter
P308. †	None!	**[SELECT] Mode 8** Enter Operating Parameter
P309. †	None!	**[SELECT] Mode 9** Enter Operating Parameter

Parameter	Selection	Description (660 Series)
Access Codes		
P400. †	None!	**Personal Identification Number** Enter Code
P401. †	None!	**Quick Calibration Access Code** Enter Code
P402. †	21353	**Limited Access Code** Enter Code
OIML		
P410. †	Disabled	**OIML Enable** Enter 9990, 9991
P411.00 † P411.01 P411.02 P411.03 P411.04 P411.05 P411.06 P411.07 P411.08 P411.09 P411.10 P411.11 P411.12	USA France German UK Denmark Sweden Italy Spain Japan Norway Denmark2 Spain2 Latin America	**Language Character Set** Enter 0→ 12
P412.00 † P412.01	Disable Enable	**Preset Enable** Enter 0→ 1
VFD/LCD Setup		
P420.00 P420.01 † P420.02	Off On Auto	**Standard VF Display Mode** Enter 0→ 2
P421.00 P421.01 P421.02 † P421.03 P421.04 P421.05 P421.06 P421.07 P421.08 P421.09 P421.10 P421.11 P421.12 P421.13 P421.14 P421.15	2d 4d 6d 8d 10d 12d 14d 16d 18d 20d 22d 24d 26d 28d 30d 32d	**Weight Threshold Divisions** Enter 0→ 15
P422.00 P422.01 P422.02 P422.03 P422.04 P422.05 † P422.06 P422.07 P422.08 P422.09 P422.10 P422.11 P422.12 P422.13 P422.14 P422.15	30 seconds 1 minutes 2 minutes 3 minutes 4 minutes 5 minutes 10 minutes 15 minutes 20 minutes 25 minutes 30 minutes 35 minutes 40 minutes 45 minutes 1 hour 2 hours	**Timeout** Enter 0→ 15
P423.01 P423.02 P423.03 P423.04 P423.05 P423.06 P423.07 P423.08 P423.09 P423.10 †	10% 20% 30% 40% 50% 60% 70% 80% 90% 100%	**Display 'ON' Brightness** Enter 1→ 10

PARAMETER	SELECTION	DESCRIPTION (660 SERIES)
P424.00 † P424.01 P424.02 P424.03 P424.04 P424.05 P424.06 P424.07 P424.08 P424.09 P424.10	Off 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%	**Display 'OFF' Dimness** Enter 0→ 10
P425.00 † P425.01	Disabled Enabled	**4X20 VF Display Enable** Enter 0→ 1
NTEP		
P440.00 † P440.01	Disabled Enabled	**NTEP Enable** Enter 0→ 1
Keypad		
P450.00 † P450.01	28 Key 25 Key	**Keypad Selection** Enter 0→ 1
P451.00 P451.01 P451.02 P451.03 P451.04 P451.05 P451.06 † P451.07	None Very Slow Slow Medium Slow Medium Medium Fast Fast Very Fast	**Keypad Repeat Rate** Enter 0→ 7
P460.00 P460.01 P460.02 P460.03 † P460.04 P460.05 P460.06 P460.07	Off Minimum Extra Low Low Medium Medium High High Maximum	**Beeper Volume** Enter 0→ 7
Time & Date		
P500. †	00:00	**Time** Enter 24 Hour Time (hh.mm.ss)
P501.70 †	01/01/70	**Date** Enter Date (mm.dd.yy)
P502.00 † P502.01	Disabled Enabled	**Time/Date Access** Enter 0→ 1
P503.00 P503.01 †	No Yes	**AM/PM Time Format** Enter 0→ 1
P504.00 † P504.01	U.S.A. International	**Date Format** Enter 0→ 1
DSD Parameters		
P590.00 † P590.01	Disable Enable	**DSD Enable** Enter 0→ 1
P591.00 † P591.01 P591.02 P591.03 P591.04	None! Comm 1 Comm 2 Comm 3 Comm 4	**DSD Communication Port** Enter 0→ 4
P592.00 P592.01 P592.02 ↓ P592.82 † ↓ P592.255	<NUL> <SOH> <STX> ↓ "R" ↓ <255>	**DSD Receive Character** Enter ASCII Value .000→ .255
P593.00 † P593.01 ↓ P593.250	None! Transmit 1 ↓ Transmit 250	**DSD Custom Transmit** Enter 1→ 250
P594.00 †	0	**DSD Maximum Number of Rows** Enter 0→ Max Available
P595.00 †	0	**DSD Number of Rows Warning** Enter 0→ Max Rows @ P594
Parameter Renaming		
P600. †	None!	**Rename Gross** Enter Name
P601. †	None!	**Rename Net** Enter Name

PARAMETER	SELECTION	DESCRIPTION (660 SERIES)
P602. †	None!	**Rename Tare** Enter Name
P603. †	None!	**Rename Gross Total** Enter Name
P604. †	None!	**Rename Gross Total + Current** Enter Name
P605. †	None!	**Rename Gross Total – Current** Enter Name
P606. †	None!	**Rename Net Total** Enter Name
P607. †	None!	**Rename Net Total + Current** Enter Name
P608. †	None!	**Rename Net Total – Current** Enter Name
P609. †	None!	**Rename Accumulation** Enter Name
P610. †	None!	**Rename Scale#** Enter Name
P611. †	None!	**Rename Time/Date** Enter Name
P615. †	None!	**Rename Average Gross** Enter Name
P616. †	None!	**Rename Average Net** Enter Name
P618. †	None!	**Rename Peak Gross** Enter Name
P619. †	None!	**Rename Peak Net** Enter Name
P620. †	None!	**Rename Rounded Gross** Enter Name
P621. †	None!	**Rename Rounded Net** Enter Name
P623. †	None!	**Rename Rate** Enter Name
P624. †	None!	**Rename Free Fall 1** Enter Name
P625. †	None!	**Rename Future Gross 1** Enter Name
P626. †	None!	**Rename Future Net 1** Enter Name
P627. †	None!	**Rename Free Fall 2** Enter Name
P628. †	None!	**Rename Future Gross 2** Enter Name
P629. †	None!	**Rename Future Net 2** Enter Name
P630. †	None!	**Rename Quantity** Enter Name
P631. †	None!	**Rename Quantity Total** Enter Name
P632. †	None!	**Rename Quantity Total+Current** Enter Name
P633. †	None!	**Rename Quantity Total - Current** Enter Name
P634. †	None!	**Rename Average Piece Weight** Enter Name
P635. †	None!	**Rename Avg Piece Weight x K** Enter Name
P636. †	None!	**Rename Percent Accuracy** Enter Name
P637. †	None!	**Rename Sample** Enter Name
P640. †	None!	**Rename Grs Total of All Scales** Enter Name
P641. †	None!	**Rename Net Total of All Scales** Enter Name
P642. †	None!	**Rename Tare Total of All Scales** Enter Name
P643. †	None!	**Rename Total of All Grs Totals**

PARAMETER	SELECTION	DESCRIPTION (660 SERIES)
		Enter Name
P644. †	None!	**Rename Total of All Net Totals** Enter Name
P645. †	None!	**Rename Qty Total of All Scales** Enter Name
P646. †	None!	**Rename Total of All Qty Totals** Enter Name
Total & Tare Save		
P660.00 P660.01 P660.02 †	No Save On Request Auto Save	**Total Values Save Method** Enter 0 → 2
P661.00 P661.01 P661.02 †	No Save On Request Auto Save	**Tare Value Save Method** Enter 0 → 2
Variables		
P680.00 † P680.01 P680.02 P680.03 ↓ P680.999	None! 1 2 3 ↓ 999	**Variable Allocation** Enter 0 → 999
P681.01 P681.02 P681.03 ↓ P681.	Variable 1 Variable 2 Variable 3 ↓ Variable 999	**Variable Instance Selection**
P682. †	None!	**Variable Name** Enter Name
P684.00 † P684.01 P684.02	No Save On Request Auto Save	**Variable Value Save Method** Enter 0 → 2
P685.00 † P685.01	Disabled Enabled	**Variable Lock** Enter 0 → 1
P686.00 † P686.01 P686.02 P686.03	Float Integer Unsigned Int String	**Variable Type** Enter 0 → 3
P687.00 P687.01 P687.02 P687.03 P687.04 P687.05 P687.06 † P687.07 P687.08 P687.09 P687.10 P687.11 P687.12 P687.13 P687.14	0 decimal pt 1 decimal pt 2 decimal pt 3 decimal pt 4 decimal pt 5 decimal pt Auto decimal Scale 1 Scale 2 Scale 3 Scale 4 Scale 5 Scale 6 Scale 7 Scale 8	**Float Style** Enter 0 → 14
P688.00 † P688.01 P688.02 P688.03	Number Time/Date Time Date	**Integer Style**
P689.01 P689.02 P689.03 P698.04 P698.05 P698.06 P698.07 P698.08 P698.09 P698.10 † ↓	1 character 2 characters 3 characters 4 characters 5 characters 6 characters 7 characters 8 characters 9 characters 10 characters ↓	**String Size** Enter 1 → 63

PARAMETER	SELECTION	DESCRIPTION (660 SERIES)
P689.63	63 characters	
Database		
P699.00 † P699.01 P699.02 P699.03 ↓ P699.250	None! Database 1 Database 2 Database 3 ↓ Database 250	**Database Instance Selection** Enter 1 → 250
P700. †	None!	**Database Name** Enter Name
P701. P702. P703. ↓ P798.	Column 1 Column 2 Column 3 ↓ Column 98	**Database Column Parameter** Enter Operating Parameter
Keypad Key Assignments		
P800.00 † P800.01 P800.02 P800.03 ↓ P800.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[SELECT] Key Function** Enter 0 → 250
P801.00 † P801.01 P801.02 P801.03 ↓ P801.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[ZERO] Key Function** Enter 0 → 250
P802.00 † P802.01 P802.02 P802.03 ↓ P802.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[TARE] Key Function** Enter 0 → 250
P803.00 † P803.01 P803.02 P803.03 ↓ P803.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[UNITS] Key Function** Enter 0 → 250
P804.00 † P804.01 P804.02 P804.03 ↓ P804.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[SCALE SELECT] Key Function** Enter 0 → 250
P805.00 † P805.01 P805.02 P805.03 ↓ P805.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[PRINT] Key Function** Enter 0 → 250
P806.00 † P806.01 P806.02 P806.03 P806.04 P806.05 P806.06 ↓ P806.250	None Menu Database Menu & Dbase Macro 4 Macro 5 Macro 6 ↓ Macro 250	**[ID] Key Function** Enter 0 → 250
P807.00 † P807.01 P807.02 P807.03 ↓ P807.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[ENTER] Key Function** Enter 0 → 250
P808.00 † P808.01 P808.02 P808.03 ↓ P808.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[CLEAR] Key Function** Enter 0 → 250

PARAMETER	SELECTION	DESCRIPTION (660 SERIES)
P809.00 † P809.01 P809.02 P809.03 ↓ P809.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[.] Key Function** Enter 0→ 250
P810.00 † P810.01 P810.02 P810.03 ↓ P810.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[0] Key Function** Enter 0→ 250
P811.00 † P811.01 P811.02 P811.03 ↓ P811.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[1] Key Function** Enter 0→ 250
P812.00 † P812.01 P812.02 P812.03 ↓ P812.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[2] Key Function** Enter 0→ 250
P813.00 † P813.01 P813.02 P813.03 ↓ P813.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[3] Key Function** Enter 0→ 250
P814.00 † P814.01 P814.02 P814.03 ↓ P814.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[4] Key Function** Enter 0→ 250
P815.00 † P815.01 P815.02 P815.03 ↓ P815.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[5] Key Function** Enter 0→ 250
P816.00 † P816.01 P816.02 P816.03 ↓ P816.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[6] Key Function** Enter 0→ 250
P817.00 † P817.01 P817.02 P817.03 ↓ P817.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[7] Key Function** Enter 0→ 250
P818.00 † P818.01 P818.02 P818.03 ↓ P818.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[8] Key Function** Enter 0→ 250
P819.00 † P819.01 P819.02 P819.03 ↓ P819.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**[9] Key Function** Enter 0→ 250
P820.00 † P820.01 P820.02 P820.03 ↓ P820.250	Enabled Macro 1 Macro 2 Macro 3 ↓ Macro 250	**Any Key Function** Enter 0→ 250
Programmable Digital Input/Output		
P850.01 P850.02 P850.03 ↓ P850.08	Channel 1 Channel 2 Channel 3 ↓ Channel 8	**Channel Instance Selection** Enter 1→ 8

PARAMETER	SELECTION	DESCRIPTION (660 SERIES)
P851.00 P851.01 P851.02 † P851.03 P851.04 P851.05 P851.06 P851.07 P851.08 P851.09 P851.10	None Freq Output Setpoint Freq Input A Freq Input B Phase Time Delay Input Delay Output Quadrature 2 Quadrature 3 Freq Debounce	**Channel Function** Enter 0→ 10
P852. †	None!	**First I/O Parameter Name** Enter Name
P853. †	None!	**Second I/O Parameter Name** Enter Name
P854. †	None!	**Third I/O Parameter Name** Enter Name
P856. †	10752 Hz	**Maximum Expected Frequency** Enter 48 → 10752 Hz
P857.00 † P857.01	Clock A Clock B	**Clock Source** Enter 0→ 1
P858.00 † P858.01	Rising Edge Falling Edge	**Pulse Edge Detection** Enter 0→ 1
P859. †	0.0 ms	**Pulse Measurement Period** Enter 0.001 → 666.0 seconds
P860.01 † P860.02 P860.03 ↓ P860.255	1 Pulse 2 Pulses 3 Pulses ↓ 255 Pulses	**Number of Pulses to Measure** Enter 1→ 255
P861.00 † P861.01	Low High	**Phase Measurement Type** Enter 0→ 1
P862.00 †	Sink	**Pulse Type** Enter 0
P863. †	None!	**Pulse Scaling Factor** Enter Operating Parameter
P864.00 † P864.01 P864.02 P864.03 P864.04 P864.05 P864.06 P864.07 P864.08 P864.09 P864.10 P864.11 P864.12 P864.13 P864.14	0 decimal pt 1 decimal pt 2 decimal pt 3 decimal pt 4 decimal pt 5 decimal pt Auto decimal Scale 1 Scale 2 Scale 3 Scale 4 Scale 5 Scale 6 Scale 7 Scale 8	**Number of Decimal Places** Enter 0→ 14
Custom Transmit		
P980.00 P980.01 P980.02 P980.03 P980.04 P980.05 † ↓ P980.250	Disabled 0.11 seconds 0.12 seconds 0.13 seconds 0.14 seconds 0.15 seconds ↓ 25.0 seconds	**Continuous Transmit Rate** Enter 0→ 250
P989.01 † P989.02 P989.03 ↓ P989.250	Transmit 1 Transmit 2 Transmit 3 ↓ Transmit 250	**Custom Transmit Instance Selection** Enter 1→ 250
P990. †	None!	**Custom Transmit Name** Enter Name
P991.00 P991.01 † P991.02	Off On Request Prompt	**Transmit Mode** Enter 0→ 2
P992.01 † P992.02 P992.03 P992.04	Comm Port 1 Comm Port 2 Comm Port 3 Comm Port 4	**Serial Port Selection** Enter 1→ 4

PARAMETER	SELECTION	DESCRIPTION (660 SERIES)
P993.00 † P993.01 *	Ignored Delayed	**Current Scale Motion** Enter 0 → 1 ** Default custom transmit #1 is motion delayed.*
P994. †	Ignored	**Scale Motion** Enter 0 for motion ignored Enter any combination of 1 → 8 for motion delay on entered scale numbers.
P995.00 † P995.01	Ignored Delayed	**Scale 2 Motion** See P994. Model 650 backward compatibility parameter.
P996.00 † P996.01	Ignored Delayed	**Scale 3 Motion** See P994. Model 650 backward compatibility parameter.
P997.00 † P997.01	Ignored Delayed	**Scale 4 Motion** See P994. Model 650 backward compatibility parameter.
P998.00 † P998.01	Disabled Enabled	**Continuous Transmit Enable** Enter 0 → 1
P999.00 † P999.01	No Yes	**Transmit Table Limited Access** Enter 0 → 1
P1000. P1001. P1002. ↓ P4999.		**Transmit Table Entry** Enter text, operating parameters, control codes
Setpoints		
P5099.1 P5099.2 P5099.3 ↓ P5099.256		**Setpoint Instance Selection** Enter 1 → 256
P5100.0 † P5100.1 P5100.2	Disabled Output Input	**Setpoint Mode** Enter 0 → 2
P5101. †	None!	**Setpoint Name** Enter Name
P5110.0 † P5110.1 P5110.2 P5110.3 P5110.4 P5110.5 P5110.6 P5110.7 P5110.8 P5110.9 P5110.10 P5110.11 P5110.12 P5110.13 P5110.14 P5110.15 P5110.16 P5110.17 P5110.18 P5110.19 P5110.20 P5110.21 P5110.22 P5110.23	Above Below Between Outside Always Never Motion Scale1 Motion Scale2 Motion Scale3 Motion Scale4 Stable Scale1 Stable Scale2 Stable Scale3 Stable Scale4 Mot'n Current Stabl Current Motion Scale5 Motion Scale6 Motion Scale7 Motion Scale8 Stable Scale5 Stable Scale6 Stable Scale7 Stable Scale8	**Activation Condition** Enter 0 → 23
P5111. †	0.00 seconds	**Activation Delay (seconds)** Enter 0.01 → 5,767,168
P5112.0 † P5112.1 P5112.2 P5112.3 ↓ P5112.250	None! Macro 1 Macro 2 Macro 3 ↓ Macro 250	**Activation Macro#** Enter 0 → 250
P5113.0 †	Ignored	**Activation Motion**

PARAMETER	SELECTION	DESCRIPTION (660 SERIES)
P5113.1	Delayed	Enter 0 → 1
P5114. †	None!	**Lower Activation Parameter** Enter Operating Parameter
P5115. †	None!	**Upper Activation Parameter** Enter Operating Parameter
P5130.0 † P5130.1 P5130.2 P5130.3 P5130.4 P5130.5 P5130.6 P5130.7 P5130.8 P5130.9 P5130.10 P5130.11 P5130.12 P5130.13 P5130.14 P5130.15 P5130.16 P5130.17 P5130.18 P5130.19 P5130.20 P5130.21 P5130.22 P5130.23	Above Below Between Outside Always Never Motion Scale1 Motion Scale2 Motion Scale3 Motion Scale4 Stable Scale1 Stable Scale2 Stable Scale3 Stable Scale4 Mot'n Current Stabl Current Motion Scale5 Motion Scale6 Motion Scale7 Motion Scale8 Stable Scale5 Stable Scale6 Stable Scale7 Stable Scale8	**Deactivation Condition** Enter 0 → 15
P5131. †	0.00 seconds	**Deactivation Delay (seconds)** Enter 0.01 → 5,767,168
P5132.0 † P5132.1 P5132.2 P5132.3 ↓ P5132.250	None! Macro 1 Macro 2 Macro 3 ↓ Macro 250	**Deactivation Macro#** Enter 0 → 250
P5133.0 † P5133.1	Ignored Delayed	**Deactivation Motion** Enter 0 → 1
P5134. †	None!	**Lower Deactivation Parameter** Enter Operating Parameter
P5135. †	None!	**Upper Deactivation Parameter** Enter Operating Parameter
P5150. †	Gross	**Compare Parameter** Enter Operating Parameter
Modbus Parameter Map		
P6001. † P6002. P6003. ↓ P6247.	None!	**Modbus Address Translation Table** Enter Operating Parameter
Macros		
P9980.0 † P9980.1 P9980.2 P9980.3 ↓ P9980.250	None! Macro 1 Macro 2 Macro 3 ↓ Macro 250	**Abort Macro#** Enter 0 → 250
P9980.1 † P9981.1	Menu Immediate	**Macro Abort Method** Enter 0 → 1
P9990.0 P9990.1 P9990.2 P9990.3 ↓ P9990.250	None! Macro 1 Macro 2 Macro 3 ↓ Macro 250	**Macro Instance Selection** Enter 1 → 250
P9991. †	None!	**Macro Name** Enter Name
P9992.0 † P9992.1	Standard Immediate	**Macro Priority** Enter 0 → 1
P9993.0 † P9993.1	Disabled Enabled	**Macro Menu Enable** Enter 0 → 1
P9994.0 † P9994.1	No Yes	**Macro Table Limited Access** Enter 0 → 1
P10001. P10002.		**Macro Entry Table** Enter Text

PARAMETER	SELECTION	DESCRIPTION (660 SERIES)
P10003. ↓ P19999.		
P50000.0 † P50000.1	No Yes	**Macro Debug Limited Access** Enter 0→ 1
P50001.		**Macro Debug Table** Read-Only
Information & Diagnostic Parameters		
P60000. †	8192	**EEPROM (E^2) Memory Installed**
P60001.	7206	**EEPROM (E^2) Memory Available**
P60002. †	224K	**RAM Installed**
P60003.	192K	**RAM Dynamically Allocated**
P60004.	190K	**RAM Available**
P60005.	0	**Macro Trace Buffer Size (bytes)**
P60006.	AMD	**FRAM Type**
P60007.	1024K	**FRAM SIze**
P60008.	445K	**FRAM Available**
P60010.	256K	**Optional RAM Installed (K-bytes)**
P60011.	262144	**Optional RAM Installed (bytes)**
P60012.	262080	**Optional RAM Available (bytes)**
P60013.	64	**Optional RAM Used (bytes)**
P60014.	262076	**Optional RAM Block (bytes)**
P60018.	Good	**Optional RAM Battery Voltage**
P60020.	0	**Database RAM Usage**
P60021.	none	**Database Error**
P60030.0 † P60030.1 P60030.2 P60030.3 ↓ P60030.250	None! 1 2 3 ↓ 250	**Database Number** Enter 1→ 250
P60031. †	None!	**Database Name**
P60032.	0	**Number of Rows**
P60033.	28	**Total Bytes Used**
P60034.	14	**Bytes/Row**
P60040.0 P60040.1 P60040.2 P60040.3 ↓ P60040.255	Disabled 1K 2K 3K ↓ 255K	**Setup RAM (K-bytes)** Enter 1→ 255 * * Maximum entry is 1K less than RAM installed (P60010)
P60041.		**Setup RAM (bytes)**
P60042.		**Setup RAM Available (bytes)**
P60043.		**Setup RAM Used (bytes)**
P60050.0 P60050.14 P60050.15 P60050.16 ↓ P60050.255	Disabled 14K 15K 16K ↓ 255K	**Pile RAM (K-bytes)** Enter 1→ 255 * * Maximum entry is 1K less than RAM installed (P60010)
P60051.		**Pile RAM (bytes)**
P60052.		**Pile RAM Available (bytes)**
P60053.		**Pile RAM Used (bytes)**
P60090.	Clock 24MHz	**Processor Clock Speed**
P60091.	0 1	**ROM Wait States**
P60092.	FastT 0 1	**RAM Wait States**
P60100.	©1995 *GSE*	**Copyright**
P60101.	0660-xxxxx	**Firmware Revision**
P60102.	mmm dd yyyy	**Firmware Date**
P60103.	Bxxxb010yy	**Flash Boot Block Revision**
P60104.	Bmmddyyyyx	**Flash Boot Block Date**
P60200.	xxxxx	**Circuit Board Serial Number**
P60201.	xxxxx	**OIML Audit Trail Counter**
P60202.	xxxxx	**Indicator Serial Number**
P60203.	xxxxx	**Calibration Audit Trail Counter**
P60204.	xxxxx	**Setup Audit Trail Counter**
P60205.	CHECK	**NTEP Compliance Parameters**
P61099.1 †	1	**Scale# for P61110→ P61122**

PARAMETER	SELECTION	DESCRIPTION (660 SERIES)
P61099.2 P61099.3 P61099.4 P61099.5 P61099.6 P61099.7 P61099.8	2 3 4 5 6 7 8	Enter 1→ 8
P61100.	mVv0.00000	**Current mV/V Output**
P61101. †	1.000000	**Calibration Factor**
P61102.	0.000000	**ReZero Weight** Press [CLR] to Reset
P61103.	0.000000	**Zero Track Weight** Press [CLR] to Reset
P61104.0 † P61104.1 P61104.2 P61104.3 P61104.4 P61104.5 P61104.6 P61104.7 P61104.8 P61104.9 P61104.10 P61104.11 P61104.12 P61104.13 P61104.14 P61104.15	0% 25% 50% 75% 100% 125% 150% 175% 0% -25% -50% -75% -100% -125% -150% -175%	**Coarse Zero Calibration** Enter 1→ 15
P61105.	0	**Fine Zero Calibration**
P61106.0 P61106.1 † P61106.2 P61106.3	25 50 100 200	**Coarse Gain Calibration** Enter 1→ 4
P61107.	1.000000	**Fine Gain Calibration**
P61110.	xxxxxx	**Zero Adjust 25** Enter Factory Determined Value
P61111.	xxxxxx	**Zero Adjust 50** Enter Factory Determined Value
P61112.	xxxxxx	**Zero Adjust 100** Enter Factory Determined Value
P61113.	x.xxxxxx	**Gain Adjust 1** Enter Factory Determined Value
P61114.	x.xxxxxx	**Gain Adjust 2** Enter Factory Determined Value
P61115.	x.xxxxxx	**Gain Adjust 4** Enter Factory Determined Value
P61116.	x.xxxxxx	**Gain Adjust 8** Enter Factory Determined Value
P61117.	xxxxxx	**Analog In 1 NR Offset** Enter Factory Determined Value
P61118.	xxxxxx	**Analog In 2 NR Offset** Enter Factory Determined Value
P61119.	xxxxxx	**Analog In 4 NR Offset** Enter Factory Determined Value
P61120.	xxxxxx	**Analog In 8 NR Offset** Enter Factory Determined Value
P61121.	xxxxxx	**Voltage Reference Comp.** Enter Factory Determined Value
P61122.	xxxxx	**Multi-Scale Serial Number** Enter Serial Number
P61123.	BckUp GOOD!	**A/D Calibration Backup**
P61130.	xxxxxx	**First Linearization Weight**
P61131.	xxxxxx	**First Linearization Factor**
P61132.	xxxxxx	**Second Linearization Weight**
P61133.	xxxxxx	**Second Linearization Factor**
P61134.	xxxxxx	**Third Linearization Weight**
P61135.	xxxxxx	**Third Linearization Factor**
P61136.	xxxxxx	**Fourth Linearization Weight**
P61137.	xxxxxx	**Fourth Linearization Factor**
P61138.	xxxxxx	**Fifth Linearization Weight**
P61139.	xxxxxx	**Fifth Linearization Factor**
P61150.	0	**Number of Load Cells**
P61151.	2.000000	**Load Cell #1 F.S mV/V Rating**
P61152.	2.000000	**Load Cell #2 F.S. mV/V Rating**

PARAMETER	SELECTION	DESCRIPTION (660 SERIES)
P61153.	2.000000	**Load Cell #3 F.S. mV/V Rating**
P61154.	2.000000	**Load Cell #4 F.S. mV/V Rating**
P61155.	2.000000	**Load Cell #5 F.S. mV/V Rating**
P61156.	2.000000	**Load Cell #6 F.S. mV/V Rating**
P61157.	2.000000	**Load Cell #7 F.S. mV/V Rating**
P61158.	2.000000	**Load Cell #8 F.S. mV/V Rating**
P61159.	100.0000	**Load Cell Full Scale Capacity**
P61160.	2.000000	**Average Load Cell mV/V Rating**
P61200.1 † P61200.2 P61200.3 P61200.4 P61200.5 P61200.6 P61200.7 P61200.8	1 2 3 4 5 6 7 8	**Analog# for P61201 → P61207** Enter 1 → 8
P61201.	xxxxx	**Analog Zero Ref. (0-10V)**
P61202.	xxxxx	**Analog Gain (0-10V)**
P61203.	xxxxx	**Analog Zero Ref. (0-20mA)**
P61204.	xxxxx	**Analog Gain (0-20mA)**
P61205.	xxxxx	**Analog Zero Ref. (4-20mA)**
P61206.	xxxxx	**Analog Gain (4-20mA)**
P61207.	xxxxx	**Analog Option Serial Number**
P62000.		**Display Test** Press [ENTER]
P62001.		**Optional RAM Test** Press [ENTER]
P62002.		**EEPROM (E^2) Test** Press [ENTER]
P62010.1 † P62010.2 P62010.3 P62010.4 P62010.5 P62010.6	1 2 3 4 5 6	**Analog Output Test for P62011 → P62013** Enter 1 → 8

PARAMETER	SELECTION	DESCRIPTION (660 SERIES)
P62010.7 P62010.8	7 8	
P62011.		**0-10V Analog Output Test** Press [ENTER]
P62012.		**0-20mA Analog Output Test** Press [ENTER]
P62013.		**4-20mA Analog Output Test** Press [ENTER]
P64000.		**Send Setup** Press [ENTER]
P64001.		**Send Setup & Parameter Values** Press [ENTER]
P64100.	0	**Upload Line Count**
P64101.	0	**Upload Error Count**
P64102.	None!	**First Upload Error**
P64103.0 P64103.1 P64103.2 P64103.3 P64103.4	Off Comm1 Comm2 Comm3 Comm4	**Upload Debug Output**
P65000.		**Copy EEPROM (E^2)** Press [ENTER]
P65001.		**Complete Factory Default** Press [ENTER]
P65002.		**Factory Default (except CAL)** Press [ENTER]
P65003.		**Script File** Press [TARE], then [ENTER]
P65010.		**Database Reset** Press [ENTER]
P62002.		**EEPROM (E^2) Test** Press [ENTER]

Appendix E Error Codes / Messages

This sections describes all error codes generated by 60 Series instruments. Most error codes show a two-digit code reference along with a short text message. Possible causes and remedies are described for each error.

Operational Mode Error Codes

Message		Description
Code02	Under Load!	Input signal less than negative full scale. If this is due to excessive loading, reduce the load. Otherwise check the load cell connections. If a 4 wire load cell cable is being used, check that the sense jumpers are in place. Verify that the capacity selection **P110** is correct. Use the information parameters, especially **P61103** and **P61104**, to check the setup and input signal.
CoDE03	Over-Load!	Input signal is greater than positive full scale. Use same check as for underload.
CoDE04	#> Dsply	Number to be displayed will not fit within 6 digits. This will not normally occur for the Gross, Net or Tare Weights but may result while displaying the accumulated totals if the amount exceeds 999,999. Either clear the totals or settle for only being able to transmit the totals.
CoDE05	Zero> Max.!	An attempt was made to zero out more than allowed per **P118** selection. Use the **[TARE]** key for subtracting off container weights or if large dead-load is always to be present, apply this dead-load during the No Load? prompt during calibration to permanently eliminate the offset.
CoDE06	Tare> F.S.!	Tare entry was greater than full scale. Most likely the entered tare value was incorrect.
CoDE07	Tare< 0!	Negative tare attempted, but not allowed per **P162**. For auto-tares, the GROSS Weight must be greater than zero unless **P162** is changed to allow negative tares.
CoDE08	Check Conn	The signal into the A/D is greater than +/- 2 times the expected full scale signal. For example if the full scale capacity at P110 is 100, then the error message will be displayed at +/- 208 taking into consideration the 4% overload. This error usually indicates a defective or incorrectly wired load cell.

SETUP MODE ERROR CODES

MESSAGE		DESCRIPTION
CoDE10	Entry >Max!	An entry was made which had more characters than allowed. The most likely cause is making an entry for an ID that is longer than the programmed size of that ID.
CoDE11	WRONG CODE!	The incorrect access code was entered, thus preventing changes. In order to access the Setup Mode, either the proper code must be entered or the **[ENTER]** key must be pressed alone (to view selections without making changes).
CoDE12	No Mods!	The Setup Mode is being accessed, but changes are prevented.
CoDE13	Outof Range	An entry made for a selection was beyond the range of valid choices. Also, an out of range error will occur during the execution of a macro utilizing the “%m” command. For example, If the command wants to strip out characters 5 through 8 and the string is only set for 2 characters, this error will occur.
CoDE14	Must Keyin	The choice for the current parameter must be keyed in.
CoDE15	Size >Max!	The size of one of the input interpreter, macro or custom transmit tables has exceeded it’s limit. Input interpreter size is limited to 198 for 660 Series, 98 for all other products; macro size is limited to 9996; custom transmit size is limited to 3997.
CoDE16	CHECK JUMPR	A programming operation was attempted when the program jumper is installed. Installation of this jumper will prohibit any programming changes.

HARDWARE PROBLEM ERROR CODES

MESSAGE		DESCRIPTION
CoDE17	A/D BAD!	The processor has detected a problem with the A/D chip. Several situations could cause this error message to be displayed. The most severe situation would be a damaged or defective A/D. In this case it will have to be replaced. Other possibilities include: 1. **Faulty chip solder:** The solder on the surface mount chip (U4) is faulty and should be re-soldered. This task should be performed by a technician qualified for surface mount repair. 2. **Wet/damp:** The unit was water damaged or is simply in a damp environment and condensation has built up on the A/D and is shorting the component. A heat gun or common blow dryer used *carefully* will sufficiently dry the area.
CoDE18	BufSz Max!	The accumulative *total* buffer size for both the TX and RX buffers of all COMM ports exceeds the reserved storage capacity (8192 bytes for the 660 Series; 4096 bytes for the 460 Series). If entries to the “buffer size” parameters (P207-P208) exceed this total, this error message will be displayed.
CoDE19	„Data &Stop	Certain combinations of protocol are not available. The protocol combination selections are in P201, P202 and P203. This error occurs if an illegal protocol combination is selected. Refer to *Chapter 8, Communications* for more information.

MESSAGE		DESCRIPTION
CoDE20	Defit A/D	The A/D calibration data is corrupt. When the information modes are accessed (P61100) this message will be displayed for 1 second and P61113 – P61116 are defaulted to a factor of 1. This will also happen on power-up if the check-sum for the A/D data are corrupted.
CoDE21	Write NVErr	Error **reading** data from the EEPROM. Possible U27 or U28 problem.
CoDE22	Read NVErr	Error **writing** data to the EEPROM. Possible U27 or U28 problem.
CoDE23	Check NVPar	Supplementary error message for above errors.
CoDE24	NVPar Full!	The setup being attempted requires more EEPROM than is currently installed.
CoDE25	Defit Setup	Upon power-up the indicator has **not** found the proper codes. Therefore, all parameters have been reset to factory default values.
CoDE26	Bad Setup	The stored data has a checksum error. Check all parameters or re-load setup.
CoDE27	RE- BOOT!	The indicator cannot use the EEPROM for data storage and is attempting to power-up again to cure the problem.
CoDE29	PIN error	The E^2 is corrupted in the PIN section. Check E^2 for problems. The access code is then defaulted to the manufacturer (GSE) access code. Also refer to Error 11.

CALIBRATION ERROR CODES

MESSAGE		DESCRIPTION
CoDE30	F.S.> MAX!	The entered calibration weight, together with the currently applied signal, indicates that the full scale signal will be greater than the allowed maximum of the controller. Verify that correct entries have been made for the capacity, **P110**, and for the calibration weight. If all appears correct, refer to the use of the information parameter **P61100**, and determine the output (in mV / volt) of the connected load cell.
CoDE31	F.S.< .1mVv	The entered calibration weight, together with the currently applied signal, indicates that the full scale signal will be less than the allowed minimum of the controller. Verify the proper entries for the capacity, **P110**, and for the calibration weight. If all appears correct, refer to the use of the information parameters, **P61100**, and determine the output (in mV / volt) of the connected load cell.
CoDE32	ADD MORE!	The applied weight during calibration was less than 0.1% of capacity. More weight than this is required. Refer to **P61100** if this is incorrect.
CoDE33	ReCAL Req'd	The just completed calibration is insufficient to guarantee accurate results due to either the cal weight being less than 5% of capacity or this was the first calibration of this platform to this Indicator and, therefore, the coarse gain was adjusted by the Indicator.
CoDE34	RES> 25K!	The current combination of capacity **P110** and increment **P111** result in a resolution greater than 25,000 graduations. This is simply a warning in case this was not intended.
CoDE35	RES> 100K!	The current combination of capacity **P110** and increment **P111** result in a resolution greater than 100,000 graduations. This is not allowed and as soon as any key is pressed the controller will jump back into the setup mode to parameter **P110** to verify the settings.

MESSAGE		DESCRIPTION
CoDE36	RES< 100!	The current combination of capacity **P110** and increment **P111** result in a resolution less than 100 graduations. This is simply a warning in case this was not intended.
CoDE37	RES< 1!!	The current combination of capacity **P110** and increment **P111** result in a resolution less than 1 graduation (for example, the increment is greater than capacity). This is not allowed and as soon as any key is pressed the controller will jump back into the Setup Mode to parameter **P110** to verify the settings.
CoDE38	Range Error	In the multi-range setup, the low range exceeds the full scale capacity, or the middle range is less than the low range. Pressing any key will automatically select the parameter and allow you to correct it.
CoDE39	„A/D˜ Cal	The A/D calibration values for one of the scales (shown as an inverse digit) have not been entered. Refer to *Restoring A/D Calibration Values* in Chapter 4.

GENERAL ERROR CODES

MESSAGE		DESCRIPTION
CoDE40	No PDIO	PDIO channel #8 is enabled on a 660 Series controller with an LCD display and Modbus configured for RTU mode. This combination is not allowed. PDIO channel #8 must be set to “None” at P851 or Modbus set to ASCII at P210.
CoDE41	Idnot Used!	If a string type variable that has not been setup (for example, had its size set to a non-zero value.) is accessed (by pressing **[3] [ID]** for example), or if no string type variables have been setup, then this message will occur.
CoDE42	Check Setup	A sample operation was performed with a sample size of “0” at parameter P34.
CoDE43	dbNOT SETUP	An attempt to access an undefined database occurred. Make sure the intended database is configured beginning at P699. This error could also indicate that the database option has not yet been initialized at P65010.
CoDE44	Name Macro	No macros were *named* when attempting to invoke the macro menu. If P806 is configured as “Menu”, make sure there is at least one macro named at P9991.
CoDE45	dbCol error	Cannot change the *type* of a variable after it is associated in a column of a database. For example, changing a variable from a string type to a float type is not acceptable. When this error message is displayed, pressing any key will proceed to the column of the database associated with the variable which has been redefined. This will allow the operator to clear the database or prompt him to redefine the variable back to its original type. This message will occur for each variable that has been redefined and associated with a database.
CoDE48	E2Typ Error	Both FRAM and standard EERPOM are installed in U27 & U28. Memory types cannot be intermixed. Power down and install one or two FRAM chips ***or*** one or two EEPROM chips and power up.
CoDE51	Too Small	The sample placed on the platform is too small to accurately compute the piece weight. Increase the sample size.
CoDE52	Can't Count	There is an insufficient quantity on the platform to perform an accurate count.
CoDE53	Accy< Req'D	The accuracy is less than required. The accuracy requirements specified at P183 has not been achieved. Increase the sample size.
CoDE54	Scale Disbl	An attempt to select a *disabled* scale was made. Make sure the intended scale is enabled and properly configured beginning at P108.

MESSAGE		DESCRIPTION
CoDE60	New FLASH	A new FLASH file has been uploaded via the ReFlash or BDM flash utilities in order to upgrade the firmware. Press any key to acknowledge.
CoDE61	Need Code	The setup mode access code has not been entered prior to attempting to change the configuration of a setup parameter. Key in the access code (i.e. **100 [SELECT] 23640 [ID] [ENTER]**).
CoDE62	Comm# Error	An entry error occurred at the "EnterCOMM#" prompt. This message appears during a setup download (P64000, P64001), database download, database print, etc. if the COMM port number was omitted or invalid.
CoDE71	TxNot Exist	The specified custom transmit does not exist. An invalid custom transmit was entered at the "Which Tx#?" prompt or incorrectly specified with a %Q macro command. Make sure the intended custom transmit is configured beginning at P989.
CoDE72	ConTx >Max!	The maximum number of continuous transmits have been exceeded (16 for the 660 Series controllers; 4 for the 460 Series indicators).
CoDE73	TxNot Cont	A request was made to disable a continuous custom transmit using the D%Q command for a transmit that was **not** currently being continuously transmitted.
CoDE74	TxRat >Max!	A request was made to set the continuous transmit interval larger than the maximum value using the I%Q macro command.
CoDE75	Tx is Cont.	A request was made to enable a continuous custom transmit using the C%Q command for a transmit that **was** currently being continuously transmitted.

MACRO ERROR CODES

MESSAGE		DESCRIPTION
	No Macro	An attempt was made to abort a macro when no macros were defined.
	Macro error	An error occurred during macro execution. Check for proper syntax. Analyze the macro debug buffer to help determine the cause of the error.
	No Start	A serial or A/D database collection command (%' or %*) was executed prior to the "start collection" command.
	Must Free	A serial or database start collection command (S%' or S%*) was issued without first freeing memory with the F%' or F%* command.
	Wrong parm	The parameter specified for data collection using the %* command was invalid or not of type float.
CoDE81	Macro Stack	The maximum number of macros pushed onto the stack has been exceeded. This error usually indicates that macros are being invoked faster than they can be executed (for example, macros invoked by continuous, short-interval input setpoints, multiple macro "calls", etc.) The maximum number of macros on the stack is 200.
CoDE82	Macro Abort	A macro was aborted from within a macro via one of the macro abort operations or from the front keypad by pressing **[CLR] + [SELECT]**.
CoDE83	Macro UnDef	A call or similar reference to an undefined macro occurred. Make sure the intended macro is configured beginning at P9990.
CoDE84	Math error	An incorrect math operation has been performed. This could be caused by trying to divide by zero or any other non-acceptable algebraic operation. This message will also occur if trying to take the negative or zero modulus of a number.
CoDE85	Syntx Error	An error occurred during macro execution as the result of invalid syntax. Re-check the macro or analyze the macro debug table to find out where the error occurred. Refer to *Chapter 9* for proper macro syntax.

MESSAGE		DESCRIPTION
CoDE86	Macro Brace	The number of opening and closing braces within a macro are different. When nesting conditional statements or grouping conditional Boolean statements, make sure the proper use the %{ and %} brace commands.
CoDE87	NoTag Found	An attempt was made to jump to an undefined tag. When using simple jump-tag commands, make sure the intended tag is properly specified and that the tag command is executed **before** the jump. When using macro-independent jump-tag commands, make sure the correct macro number is specified and that the jump text matches that of the tag.

SETUP ERROR CODES

MESSAGE		DESCRIPTION
Code92	„Stpt Setup	A parameter entry is required for a setpoint's Activation Limit, Deactivation Limit, or Compare parameter. Pressing any key will automatically select the offending parameter and allow you to correct it.
CoDE95	SyErr NvRam	The EEPROM size is too small to allocate the database requested.
Code96	Erase Boot!	An attempt was made to enter an instrument serial number or board serial number with the flash already programmed.
CoDE99	Can't Set!	An attempt to enter a value for a parameter which is not field changeable, such as the serial numbers or the audit trail counter results in this message.
	Cksum error	Upon each power-up, the indicator tests the integrity of its firmware. If the result is not correct this message is displayed and the indicator is not usable. ReFlash the indicator.

COMMUNICATIONS ERROR MESSAGES

MESSAGE		DESCRIPTION
	prtyX error	The parity of a received character did not match the parity specified in the setup mode at P202. This could also result if the baud rate (P200) or the number of data bits (P201) are incorrect. The 'X' in the error message represents the COMM port number on which the problem occurred.
	ovrnX error	An overrun error occurred where additional characters were received while the receive buffer was full. The additional characters will be lost. The 'X' in the error message represents the COMM port number on which the problem occurred.
	frmgX error	The stop bit of a received character did not occur when it was expected. This could be the result of an incorrect baud rate (P200), incorrect number of data bits (P201), or incorrect parity setting (P202). The 'X' in the error message represents the COMM port number on which the problem occurred.

MESSAGE	DESCRIPTION
portX error	The indicator did not check its receive data register in time, thus missing a character. To prevent the problem, try reducing the baud rate (P200). The 'X' in the error message represents the COMM port number on which the problem occurred.
NoTxX Allow	Associated with Modbus. This is selected at P205. This message indicates that a transmission out the specified port was attempted. This is not acceptable if the port is set for Modbus. The 'X' in the error message represents the COMM port number on which the problem occurred.
tx on hold	Occurs if a data transmission is held up for two seconds of more due to a deasserted handshake. Refer to the description of parameter P209 for more information.
tx abort	Occurs if the **[CLR]** key is pressed when the tx on hold error message is shown or if P209 is set for abort and the transmit buffer becomes full.
tx Con'd	Appear briefly when the handshake is re-asserted after the tx on hold message occurs.
BadTx Port	Appears briefly after an attempt was made to put a byte in an invalid comm port receive buffer.
Wrong Comm#	An invalid communication port number was specified.

MISCELLANEOUS MESSAGES

MESSAGE	DESCRIPTION
Entry Error	An invalid entry was made. When entering data, make sure the values are within the acceptable limits and of the proper type as required by the entry mode.
Enter Comm#	This prompt appears during a setup download (P64000, P64001), database download, database print, etc. if a COMM port was not specified.
Which Tx#?	This prompt will appear when the instrument is setup with more than one custom transmit with parameter P991 set for "Prmpt" (Prompt) and the **[PRINT]** key is pressed. The "WhichTx#?" message is asking for a custom transmit number to be entered. Key in the custom transmit desired and then press **[ENTER]**. For example, **[2] [ENTER]**, for custom transmit number 2.
Clear All?	This prompt will appear when the **[CLR]** key is pressed when the cursor is at the end of a custom transmit table or macro table. Press **[ENTER]** to clear all information or any other key to retain the table information.
Enter toCLR	This prompt is used at P65010. When **[ENTER]** is pressed this message will appear. It is then followed by "Enter=Dflt".
Sure? ???	This prompt is displayed for verification of resetting or clearing information. This message occurs at parameters such as P65001, P65002, P65010, etc. Press **[ENTER]** to clear or reset all information or any other key to retain the information.
No Instn	Indicates an instance was specified when entering an operating parameter that does not have multiple instances.
Insuf OpRAM	There is not enough operational RAM available to perform the requested function.
NoOpt RAM!	There is no database present or it has not been initialized at P65010.
OVER-WRITE	The maximum number of DSD data rows has been exceeded (P594). The oldest row in the DSD database has been overwritten with a new row of data.
Rows: < XXX	The number of DSD data rows has exceeded the warning threshold. "XXX" represents the number of warning rows specified at P595. Print and/or download

	the DSD database to prevent loss of data before the maximum number of DSD data rows is exceeded.
OutOf Memry	The current setup requires more RAM than is currently installed.
>Max Rows	Number of rows received during upload exceeded 2,147,483,648. Contact GSE if this error occurs.
Invld Data1	Occurs during upload new or upload update, followed by the row location (relative to the beginning of the upload) of the error and the column in which it was found. This code means that a float, int or unsigned value could not be interpreted and was therefore regarded as a zero (0) value. It was mostly likely caused by a string entry occurring where a numeric entry was expected. The upload will continue. Check the error codes at P60021 after an upload or use the %_ macro after the upload macros to test for this error.
Invld Data2	Occurs during upload new or upload update, followed by the row location (relative to the beginning of the upload) of the error and the column in which it was found. This code means that a float, int or unsigned value contained extra characters that were ignored. It was mostly likely caused by a string entry containing at least some numbers being where a numeric entry was expected. Some of the numbers will be used for the data in this case. The upload will continue. Check the error codes at P60021 after an upload or use the %_ macro after the upload macros to test for this error.
< Col /Row	Occurs during upload new or upload update, followed by the row location (relative to the beginning of the upload) of the error and the # of columns found. Missing columns are filled with zeros (0). This may be intentional if for example you are creating a database where a column will be filled in later. The upload will continue. Check the error codes at P60021 after an upload or use the %_ macro after the upload macros to test for this error.
> Col /Row	Occurs during upload new or upload update, followed by the row location (relative to the beginning of the upload) of the error and the # of columns found. Extra columns were encountered and ignored. This may be intentional if for example you are creating a database from an another database which does need all the columns of the existing database. The upload will continue. Check the error codes at P60021 after an upload or use the %_ macro after the upload macros to test for this error.
Colmn > Max	Occurs during upload new or upload update when a row of data is contains more than the maximum number of columns in the database. Uploading stops immediately when this occurs. This is most like due to upload data containing no carriage returns. Ensure the presence of carriage returns and not some other character at the end of each row of data. Check the error codes at P60021 after an upload or use the %_ macro after the upload macros to test for this error.

Appendix F Assembly Drawings

The following fold-out pages are assembly drawings of each 60 Series instrument. These drawings are subject to change without prior notice and are updated with each new revision of this manual.

Model	Drawing Number
460	37198
460 Panel Mount	37954
465	37673
465 Panel Mount	37637
560 & 660	37134
560 & 660 Panel Mount	37100
661	37135
661 Panel Mount	37101
562 & 662	37136
562 & 662 Panel Mount	37102
663 (VFD)	37168 (2 pages)
663 (LCD)	37474 (2 pages)
663 (VFD Big Box)	37697 (2 pages)
663 (LCD Big Box)	37537 (2 pages)
665 (VFD)	37129 (2 pages)
665 (LCD)	37133 (2 pages)
665 Panel Mount (VFD)	37758
665 Panel Mount (LCD)	37759

INDEX

An index will be compiled for this version of the 60 Series Technical Reference Manual and made available in PDF format on the GSE web site (www.gse-inc.com).

Part Number 39-10-X60REF