WEIGH-TRONIX

PC-820/821 SimPoser^{TM} Software User's Manual

05/25/01 820SIMPO.P65 PN 29725-0011B Printed in USA

Table of Contents

Introduction

About This Manual

All references in this manual to the PC-820 also refer to the PC-821.

This manual covers the information you need to install and operate the PC-820 SimPoser™ software package.

Major sections of this manual are headed by titles in a black bar like *Introduction* above. Subheadings appear in the left column. Instructions and text appear on the right side of the page. You will occasionally see notes, tips, and special instructions in the left column. This information will usually pertain to text in the opposite column.

PC-820 SimPoser™ and the PC-820 Parts Counter

PC-820 SimPoser™ software works exclusively with the PC-820 Parts Counting scale. The word SimPoser™ comes from two root words—Simulator and Composer. These two words describe the strengths of this program. Built into the software is a computer simulation of the PC-820. PC-820 SimPoser lets you compose application programs and configuration setups for the PC-820. You then test an application on the simulator. This makes quick fixes easy and assures identical function when you download the program to a PC-820.

Application programming is done in the WT-BASIC computer language. These application programs and the configuration are saved in a computer file and can be recalled simply by opening that file. Because of this, you can design, buy or trade application programs.

Downloading a program to the PC-820 is as easy as clicking a mouse button. The information you send to the PC-820 is instantly active and the PC-820 is ready to perform the task you have set it up to do. The program becomes part of the PC-820's permanent memory and cannot be lost due to power failure. If you download a new program to the PC-820, the old program is replaced with the new program. There are no chips to change and no long turnaround times. Each program can take the place of expensive software/hardware specials. Turnaround times can be measured in hours or days instead of weeks or months.

Simposer™ Software

Minimum and Recommended Computer Requirements

SimPoser is PC based and requires a certain level of computer power to function. With the minimum setup listed below, the system will work but slowly. The recommended configuration will run the program very well.

Minimum Configuration

- IBM® compatible AT®/PC with an Intel 486 DX2-50MHZ microprocessor
- 8 megabytes of RAM
- 120 megabyte hard disk drive (program takes up a minimum of 5 meg.)
- 3.5" 1.44 megabyte floppy disk drive
- VGA color monitor
- Mouse
- Microsoft Windows® 3.1 running in enhanced mode

Recommended Configuration

- IBM® compatible AT®/PC notebook computer with an Intel Pentium 90 microprocessor
- 16 megabytes of RAM
- 120 megabyte hard disk drive (program takes up a minimum of 5 meg.)
- 3.5" 1.44 megabyte floppy disk drive
- VGA color monitor
- 14.4K baud fax/modem
- Mouse
- Microsoft Windows® 3.1 running in enhanced mode, Win95, Win98 or WinNT

This section of the manual is divided into the following sections:

- SimPoser Installation
- Starting PC-820 SimPoser
- Commands Overview
- SimPoser Operation

PC-820 SimPoser Installation

Place the PC-820 SimPoser diskette in the disk drive and in Windows 3.1 click on File - Run - a:\setup, or in Windows 95 click on Start - Run - a:\setup. Follow all the on-screen prompts as the software is installed.

Starting PC-820 SimPoser

You can move the toolbar anywhere on your screen by clicking and dragging the title bar.

1. Once PC-820 SimPoser is installed, double click the icon with the left mouse button. A license message appears, and upon accepting, this toolbar is displayed:

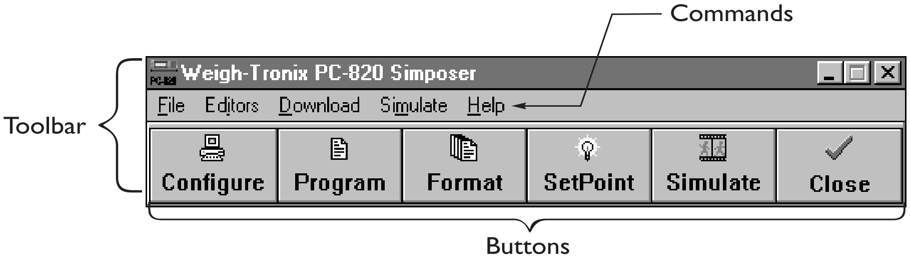


Figure 1
SimPoser's toolbar

Commands Overview

The toolbar consists of five commands and six buttons. Some of these commands are common to all Windows programs and cause a drop down menu to appear. Some commands access the special features of the PC-820 SimPoser program. A quick overview for each command is given below. Specific instructions appear in the *Operations* section of this manual. For help with Windows operations see your Windows documentation.

File

This drop down menu lets you create, open, save and print files and exit the program. This menu also contains a list of the last nine files you have opened. This allows you to click on a file name and open it quickly.

Editors

This is a navigation aid for accessing the different editing windows.

Download

Click on this command, then **COM 1** or **COM 2** to download the active application program to the PC-820.

Simulate

Click on this command, then *Start* to bring up the PC-820 simulator on the computer screen. All the parameters and instructions you have created and saved will be active and running on the display. This allows you to test your programs before downloading to the PC-820.

Help

Click on this command or press F1 to find indexed help documentation.

Toolbar Buttons Overview

The toolbar has six buttons. You select the function you want by clicking on the appropriate button with the mouse cursor. Below are brief descriptions of each button's function. Complete instructions are in the next section, PC-820 SimPoser *Operation*.

Configure button

When you click on this button with your mouse, a tabbed, dialog box appears containing the customizable features you can set to suit your needs. Below is a list of items in this dialog box:

- Base Parameters
- Settings
- Filters
- Ranges
- Counting
- Units
- Key Enable
- Serial Ports
- Display Settings
- Display Values
- Bargraph
- Misc

Program button

The Program button opens a WT-BASIC text editing window. Use this window to create application programs using the WT-BASIC computer language. In these programs you can configure the system to run a specialized counting program, setup a checkweighing function, design special graphics for use on the display, and much more.

Format button

Click on this button to see a screen for setting up print formats. Design your custom format, choose the format # (from 1 to 32) and save it by clicking on the Save button. Any or all of these formats can be recalled in the program you create in the Program window. For example, you might use this feature for IN and OUT tickets for truck loading/unloading situations, spreadsheet reports for managers, or ISO documentation.

Setpoint

Click on this button to bring up a dialog box for configuring setpoints.

Simulate

Click on this button to bring up the PC-820 simulator on the computer. This does the same thing as the Simulate command described earlier.

Close

Click on this button to exit the PC-820 SimPoser program.

SimPoser Operation

Following are specific instructions for each of the commands on the PC-820 SimPoser toolbar.

Toolbar Commands

File

This manual assumes that you know the basic Windows™ procedures. If not, see your Windows™ documentation.

1. Click on **File**. . .

The drop down menu shown in Figure 2 appears. With this menu you can call for a new file, open an existing file, save a file you are working on, save a file under a new name, access the print dialog box, or exit the PC-820 SimPoser program. There is also a file history list which holds up to nine of the most recently opened file names

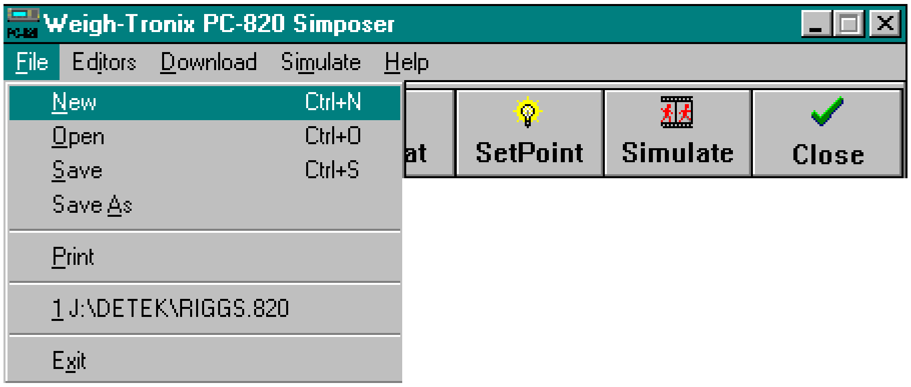


Figure 2
File menu

Editors

2. Click on the **Editors** command. . .

The following is displayed:

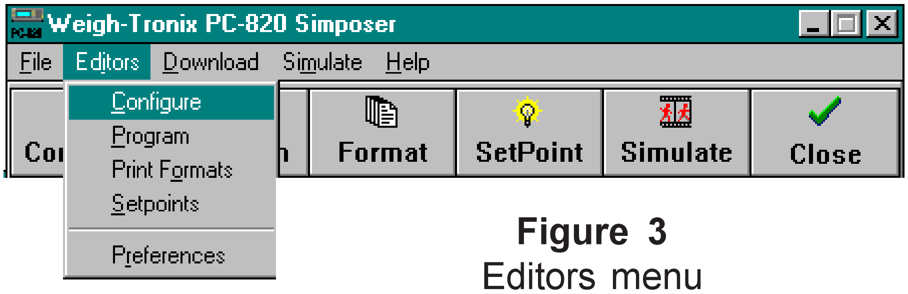


Figure 3
Editors menu

Click on the editing window you want. This is a duplication of the buttons and is handy for navigation of open windows.

The **Preferences** menu item allows you to configure the SimPoser program so that a **Tools** command appears on all the SimPoser toolbars. For more information about using this function see a Windows or DOS manual.

You may be able to recover an unsaved file after a system crash if you have performed a download or simulation. The PC-820 SimPoser program creates a temporary copy of the application in C:\WT\SIMP820\SIM\TEMPCFG.820. To recover the program, open this file in PC-820 SimPoser and do a FILE,SAVE AS,{filename} and use the file name you want for your application program.

Helpful Hints:
1. SAVE often!
2. Do not have any other Windows® programs running.
3. SAVE often!

Download

COM1 or COM 2 under Download refer to the serial output from your computer, NOT the serial ports on your PC-820

HINT: If you have an application with continuous output, you should disable it before you download.

3. Click on the **Download** command. . .

The following menu is displayed:

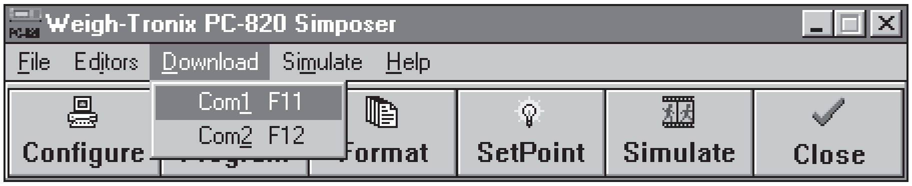


Figure 4
Download menu

When you have created a custom program, use this command to choose which Com port to use to download the program to your PC-820. F11 and F12 keys can be used as hot keys for these functions.

Simulate

Click on the **Simulate** command. . .

A simulation of the PC-820 appears on screen. It will behave the same way as a real PC-820 loaded with the program you have active in SimPoser. Use this to test your program before downloading to the real PC-820.

If you are going to print from the simulator mode you need to add this line to your autoexec.bat file on your computer and reboot before trying to print: SET WTPORT 2=1 What this means is that WTPORT2 is the simulated TT-830 serial port number 2 and =1 is the communication port from your computer. ***Never set both WTPORTS to the same computer COM port number.***

The following computer key strokes take the place of pressing front panel keys when using the simulation:

COMPUTER KEYSTROKE	**EVENT QUEUED**
ALT-H	HIDDEN_KEY
ALT-P	PREVIOUS_KEY
ALT-N	NEXT_KEY
ALT-Z	ZERO_KEY
ALT-T	TARE_KEY
ALT-C	CLEAR_KEY
ENTER	ENTER_KEY
ESCAPE	ESC_KEY
1..9	NUMERIC_KEY
A..Z	ENTRY_KEY
ALT-E	ENTER THE SETUP MENU
F1..F10	F1..F10_KEY

F6-F10 are accessible through a remote keyboard or the simulator.

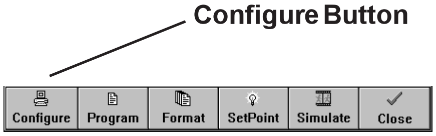
Move the mouse to change weight on the simulator.

- Move the mouse to the right to increase weight value
- Move the mouse to the left to decrease weight value
- To add more weight with shorter mouse movements, click and hold down the left mouse button while moving the mouse to the right
- To add small increments of weight, hold down the right mouse button, while moving the mouse to the right
- To exit the simulation, press both mouse buttons at the same time or press **ALT + X**

Help

Click on the Help command to bring up an indexed help manual on your computer screen. This is the last item on the command line of the PC-820 SimPoser toolbar. The next section describes the toolbar buttons.

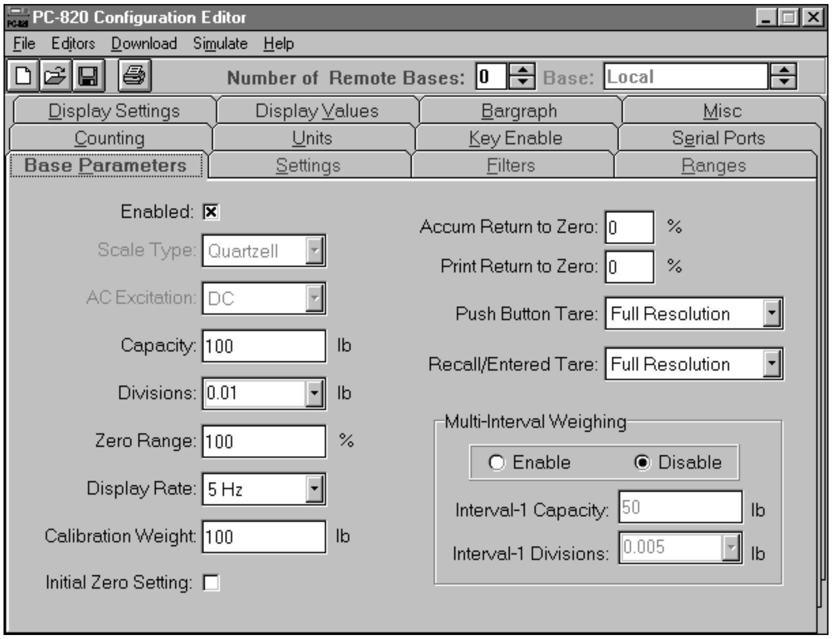
Toolbar Buttons



TIP
Combo Box = A Windows feature which allows you to type in a value or select one from a drop down list.

Text Box = A box into which you type a value or word.

Click on the first toolbar button, **Configure**. The dialog box in Figure 5 appears.



Number of Remote Bases: 0
Base: Local

Enabled: ☒
Scale Type: Quartzell
AC Excitation: DC
Capacity: 100 lb
Divisions: 0.01 lb
Zero Range: 100 %
Display Rate: 5 Hz
Calibration Weight: 100 lb
Initial Zero Setting: □

Accum Return to Zero: 0 %
Print Return to Zero: 0 %
Push Button Tare: Full Resolution
Recall/Entered Tare: Full Resolution

Multi-Interval Weighing
○ Enable ● Disable
Interval-1 Capacity: 50 lb
Interval-1 Divisions: 0.005 lb

Figure 5
Configure dialog box

At the top of the screen are listed the number of remote bases connected and the active scale number. Use these boxes to set the number of remote bases you have and choose which base you are configuring.

Base Parameters tab

This dialog box contains many tabs representing different areas of scale function. Click on a tab to bring that function into view. The first tab is **Base Parameters**.

Following is a brief description of each of the scale parameter items you see in Figure 5.

Enabled — Check this box if you want a base enabled. This means the base is available as a remote base and features such as Auto Base Switching will recognize this base. When the base is disabled, the base is not available for selection by the user and the base is ignored on power-up.

Scale Type — Select analog or Quartzell® weight sensor. Remote 1 is Quartzell by default. You can set Remote 1 to work with an analog base. If two remote bases are used, remote 2 must be analog.

AC Excitation — You can select the analog weight sensor excitation to be a DC level of 10 volts or one of three frequencies for AC excitation which is useful for reducing weight shifts due to extreme temperature changes.

By default, when the **PRINT** *key is pressed, a print operation and an accumulation take place. If you do not want the accumulation to occur, a WT-BASIC program assigning only the DOPRINT command to the* **PRINT** *key needs to be downloaded to the PC-820. A WT-BASIC program can also define an ACCUM. soft key and assign accumulation to that key only.*

Capacity Set the capacity for the chosen scale.

Divisions Set the division size for the chosen scale platform. Values must be a multiple or submultiple of 0, 1, 2, or 5. If you type an incorrect value, the program will automatically select the closest correct value.

Zero Range % Select the percentage of scale capacity you can zero.

Display Rate Set the display update rate (the number of times/second the display is updated.) Choose values between .1 (slowest, 10 times a second) and MAX (fastest) updates per second.

Calibration Weight The amount of test weight used to calibrate the scale. We recommend entering the test weight most commonly used to calibrate this scale capacity by your organization.

Initial Zero Setting Check this box to enable automatic zeroing of the scale upon power up or upon waking from a sleep mode. If this is not enabled the scale will retain the last zeroing function. This means any weight on the scale will not be zeroed upon powerup or wake-up.

Accum Return to Zero % To accumulate weight, the weight must be above this percentage of scale capacity and stable. Before you can perform another accumulation operation the weight must return to zero.

Print Return to Zero % To print, the weight must be above this percentage of scale capacity and stable. Before you can perform another print operation the weight must return to zero.

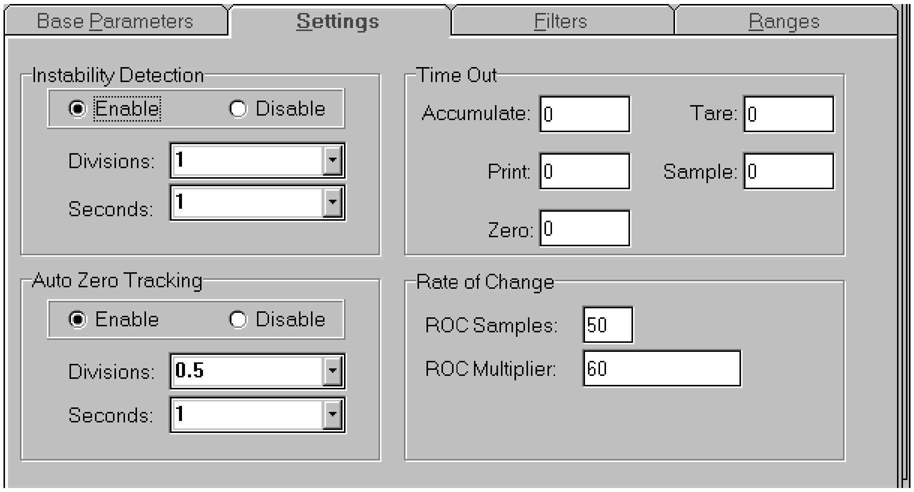
Push Button Tare Resolution Choose from *Full Resolution* or *Round to Division* for push button tare operations. *Full Resolution* allows the scale to use its maximum internal resolution (2,000,000) to set tare. This allows tare values to be double precision floating decimal point. This is usually used for counting scale applications. Choose *Round to Division* to round to the current division size before calculations are performed.

Recall/Entered Tare Resolution Choose from *Full Resolution* or *Round to Division* for tares recalled from a data base or from keyed in tare operations. *Full Resolution* allows the scale to use its maximum internal resolution (2,000,000) to set tare. This allows tare values to be double precision floating decimal point. This is usually used for counting scale applications. Choose *Round to Division* to round to the current division size before calculations are performed.

Multi-Interval Weighing Enable or disable multi-interval weighing. If you enable it, choose a lower capacity and a division size for that range of weight.

Settings tab

Settings is the next tab. See Figure 6.



Instability Detection
● Enable ○ Disable
Divisions: 1
Seconds: 1

Time Out
Accumulate: 0
Tare: 0
Print: 0
Sample: 0
Zero: 0

Auto Zero Tracking
● Enable ○ Disable
Divisions: 0.5
Seconds: 1

Rate of Change
ROC Samples: 50
ROC Multiplier: 60

Figure 6
Settings tab

Following is a brief description of each of the items you see in Figure 6.

Instability Detection (Motion) Use this to set the stability range in divisions and the stability delay in seconds.

For example, if you set the range to three divisions and the delay to one second, if the weight value does not change more than three divisions in one second, the scale is considered stable.

Default is 1 division and 1 second.

Auto Zero Tracking Use this to set AZT. This helps counteract the situation when small amounts of debris are left on the scale.

The division size you pick defines a range above and below zero. When scale weight is inside this range for the number of seconds you picked, ½ of the weight will be zeroed. The indicator will repeat removing ½ the weight every X seconds. X being the number of seconds you have picked.

Default is 0.5 division and 1 second. Set division size to zero to disable AZT.

Time Out Use this key to set the timeout value, in seconds, for each of the following parameters: (default is 0 for all)

- Print timeout
- Accumulate timeout
- Zero timeout
- Tare timeout
- Sample timeout (Used with PCWTSAMP & PCWTZERO)

Timeout is the amount of time the PC-820 will wait for motion to cease and perform the function after the key is pressed.

For example, if Zero Timeout is set to 3 seconds, when

ROC is the rate of material flowing on or off the scale. If the flow of material is constant, the value displayed is zero. If the flow increases, the value is positive. If the flow decreases, the value is negative.

For example: If you have one part that weighs 8 lbs, you calculate the multiplier using this formula:

$$\frac{\textit{Cal Unit}}{\textit{Custom Unit Weight in Cal Units}} = \textit{Multiplier}$$

So in our example the equation becomes:

$$\frac{1}{8} = 0.125$$

and the multiplier is 0.125.

the **ZERO** key is pressed the unit will zero the scale if there is no motion. If there is motion and motion ceases within 3 seconds the unit will zero the scale. If motion doesn't cease the key press is ignored.

ROC Use this to calculate Rate of Change for time/weight based applications. You are prompted to enter an ROC Samples value. Default is 50. You are then asked to enter an ROC Multiplier value. Default is 50. Explanations for these are given below.

ROC Samples
The number of samples over which the rate of change of weight is determined. The PC-820 samples the Quartzell at 50 times per second (60 times per second for external Quartzells and analog bases). If ROC Samples is set to 50, the PC-820 is determining the rate of weight change over one full second.

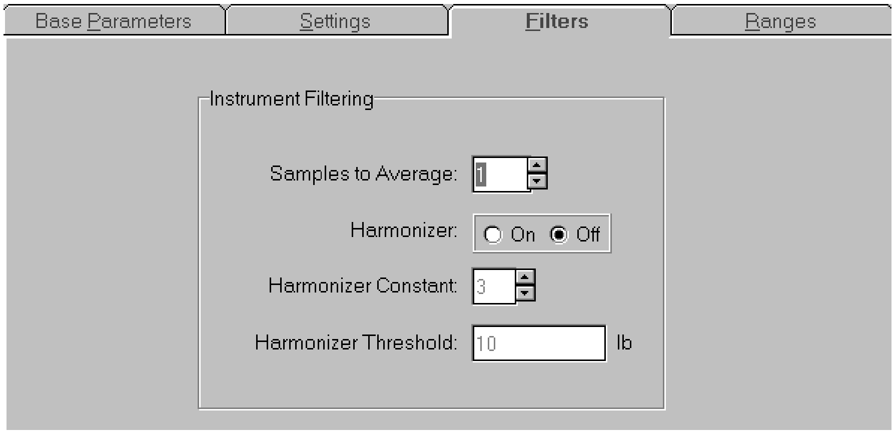
ROC Mult
The ROC Multiplier allows you to enter a conversion factor to translate weight to some other unit of measure, such as gallons, piece weight, or some other weight unit based upon the calibration unit of measure.

Filters Tab

Harmonizer™ threshold is based on actual weight in calibration units, not division size.

The Harmonizer™ default is off. Filtering is not recommended for parts counting applications using Quartzell bases.

Filters is the next tab. See Figure 7.



Instrument Filtering

Samples to Average: 1

Harmonizer: ○ On ● Off

Harmonizer Constant: 3

Harmonizer Threshold: 10 lb

Figure 7
Filters tab

Use this to set up the Harmonizer™ filtering. A full explanation of the Harmonizer™ is given below.

The weight conversion for PC-820 connected bases is:

- Local Quartzell = 50 updates/second
- Remote Quartzell = 6 updates/second
- Remote analog = 60 updates/second

The Harmonizer Constant choices are 0-10 in the PC-820 SimPoser program but it may be best to make the selection in the "real world" through the front panel.

Samples to Average is the number of conversions you want to average. For example, with an analog base, if you pick 30, the unit will average the weight values from the last 30 conversions or ½ second and uses that value for displayed data.

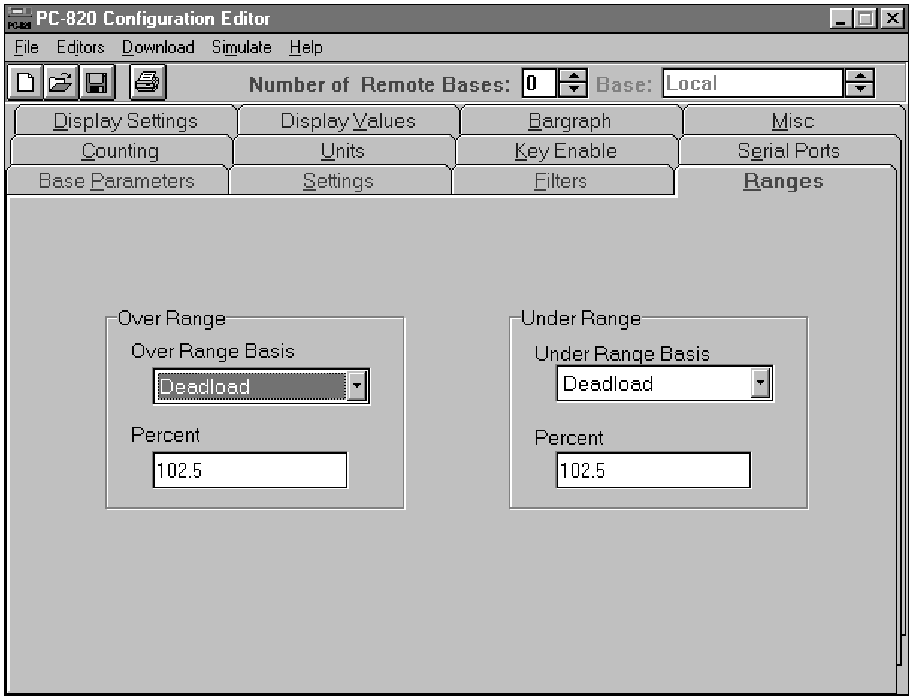
The next choice you have is for turning the Harmonizer filtering on or off. If you turn the Harmonizer filtering on you need to set the Harmonizer Constant. Typical values are between 1 and 8. Set the number low for small vibration problems and higher for more dampening effect.

Pounds is the default calibration unit.

The purpose of the Harmonizer Threshold is so the indicator will respond quickly to large weight changes. Harmonizer Threshold is the amount of weight change, in calibration units, beyond which the Harmonizer will be temporarily disabled. For example, if you set this to 10 lbs, a change in weight greater than 10 lbs between samples will disable the Harmonizer until the weight change during the sample time drops below 10 pounds.

Ranges tab

Ranges is the next tab. See Figure 8.



Number of Remote Bases: 0
Base: Local

Over Range
Over Range Basis: Deadload
Percent: 102.5

Under Range
Under Range Basis: Deadload
Percent: 102.5

Figure 8
Ranges tab

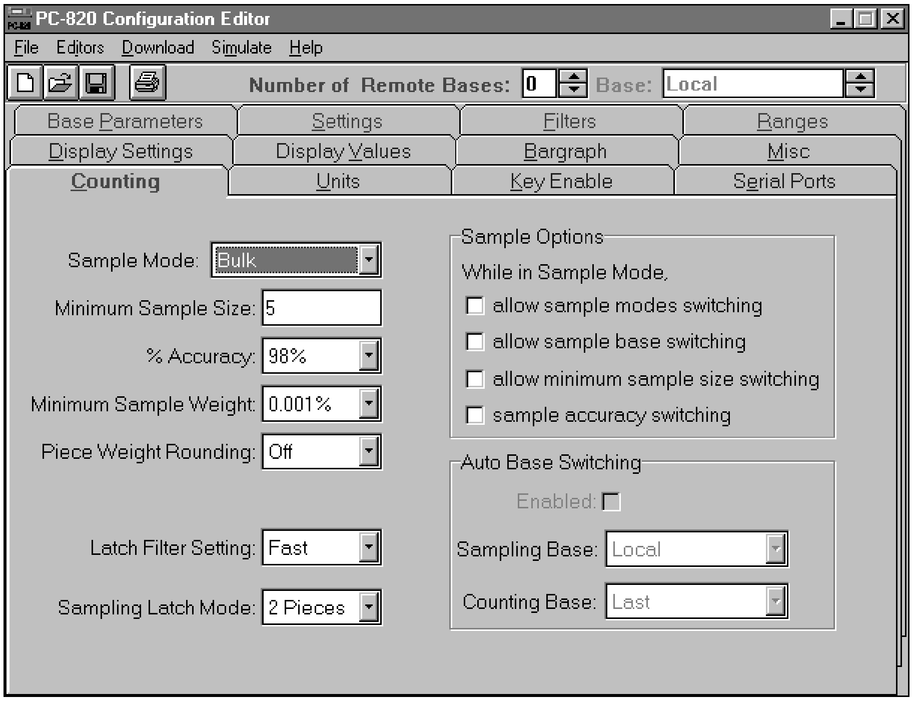
Use this to set the Over Range (overload) and Under Range (underload) limits.

You can base the range on deadload or zero and then choose an over and under range limit. If you choose deadload as the basis, the overload and underload ranges are based on calibration zero and the ranges are entered as a percent of capacity. Default is deadload and 102.5% of capacity.

If you choose Zero as the basis, the overload and underload ranges are based on current zero and the ranges are entered as a number of divisions over capacity. Default is 9 for both over and under.

Counting tab

Counting is the next tab. See Figure 9.



Number of Remote Bases: 0 Base: Local

Sample Mode: Bulk

Minimum Sample Size: 5

% Accuracy: 98%

Minimum Sample Weight: 0.001%

Piece Weight Rounding: Off

Latch Filter Setting: Fast

Sampling Latch Mode: 2 Pieces

Sample Options

While in Sample Mode,

☐ allow sample modes switching

☐ allow sample base switching

☐ allow minimum sample size switching

☐ sample accuracy switching

Auto Base Switching

Enabled: ☐

Sampling Base: Local

Counting Base: Last

Figure 9

Counting tab

Motion and AZT settings do not affect the sampling process, but they do affect the counting process.

Use this dialog box to set all the following:

Sample Mode

Sample Mode:
Choose from Bulk or Dribble (default is Bulk)

Bulk mode: Place all of the sample on scale at one time. After motion stops, the count is automatically displayed.

Dribble mode: Place sample on scale, then press **ENTER**. After motion stops, the count is displayed.

Minimum Sample Size

Key in a minimum sample size (default is 5)

% Accuracy

Choose from OFF, 95%, 98% or 99%. (default is 98%)

Minimum Sample Weight

Minimum Sample Weight as percentage of full scale:
Choose from 0.01%, 0.005%, 0.002%, and 0.001%
(default is 0.001%)

Piece Weight Rounding

Piece Weight Rounding:
Choose from OFF, 0.5%, 1%, and 2%. (default is OFF)

Latch Filter Setting

You can choose a count filter of **OFF**, **Fast** or **Slow**. (Default is Fast).

The count filter, which can also be called count latching, helps lock onto a count value when piece weights are less than a division size. Choose Fast for some filtering and Slow for more filtering.

Sampling Latch Mode Sample

Choose from **Off, 1 piece, 2 pcs**
Latching, if enabled, gives the appearance of stability immediately following the sample process. The sample latch is broken when motion is detected or if the calculated count is over two pieces away from the sample size. The latch will not function again until another sample process is completed.

Sample Options

When the scale is in sampling mode there are several features, functions or modes you can change if they are enabled. If they are not enabled, the configured choices are used. Here is where you can enable or disable those features. The features are:

Sample Mode Switching **-** Lets you switch between bulk and dribble sampling while in the sample mode.

Sample Base Switching **-** Lets you switch bases for sampling and counting while in the sample mode.

Minimum Sample Size Switching **-** Lets you change the minimum sample weight while in the sample mode.

Sample Accuracy Switching **-** Lets you change the sample accuracy while in the sample mode.

Auto Base Switching

Base Switching: (You must have selected and configured a remote base for this function to work.)

Choose to enable or disable auto base switching mode. (default is No)

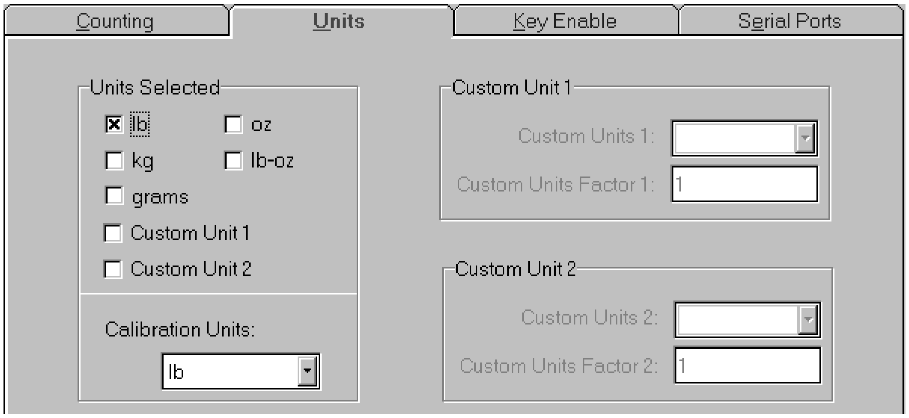
Auto base switching lets you perform sampling on one scale (local or remote) and the scale will automatically switch to another scale for counting.

If you enable this mode you are asked to choose the sampling scale and then the counting scale. The default sampling scale is **Local** and the default counting scale is **Last**. Choose **Last** to return to the scale you were using before you did your sampling.

Units tab

The next tab is **Units**. This is shown in Figure 10.

Setpoint and configuration parameters such as capacity, bargraph, checkweigher, and analog output values are based on the calibration unit, not on displayed unit of measure.



Units Selected

☒ lb
☐ oz
☐ kg
☐ lb-oz
☐ grams
☐ Custom Unit 1
☐ Custom Unit 2

Calibration Units: lb

Custom Unit 1

Custom Units 1:

Custom Units Factor 1: 1

Custom Unit 2

Custom Units 2:

Custom Units Factor 2: 1

Figure 10
Units tab

Following is a brief description of each of the unit of measure items you see in Figure 10.

The custom units conversion factor is the number to be multiplied by the weight (in calibration units) to get the desired custom unit. Example: 1 lb = 5 inches of a certain steel rod. Custom unit is inches. Calibration unit is lb. Conversion factor is 5. With six lbs of weight on the scale, 30 inches would be displayed. (Six lbs x 5 = 30 inches of steel)

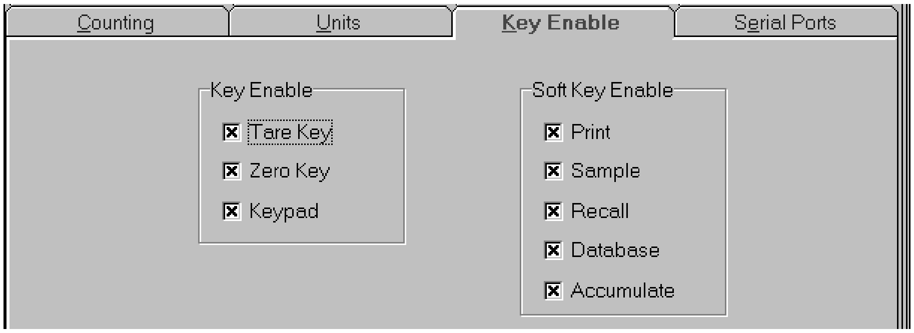
lb, kg, grams, oz lb/oz, Custom Unit 1 & 2 — Select the units of measure you want to use. Those you enable will be available to you as the **UNITS** key is pressed on the PC-820. Conversion factors are preassigned by the factory for lb, kg, grams, oz, and lb/oz.

Calibration Unit — Select the unit of measure used in the calibration of your scale. Choose from lb, kg, grams, or oz.

Custom Units — If you select a custom unit, you can choose from a list of possible units or create your own name for a unit of measure. You must also enter in a conversion factor for that custom unit of measure based on your calibration weight unit.

Key Enable tab

The next tab is **Key Enable**, shown in Figure 11. This dialog box lets you enable or disable the keys listed.



Key Enable

☒ Tare Key
☒ Zero Key
☒ Keypad

Soft Key Enable

☒ Print
☒ Sample
☒ Recall
☒ Database
☒ Accumulate

Figure 11
Key Enable tab

Serial Ports tab

Consult your peripheral device manual for proper serial port parameter selections.

A new capability has been added in the latest revision of SimPoser software. Radio communiation is now possible between PC-820s and a PC.

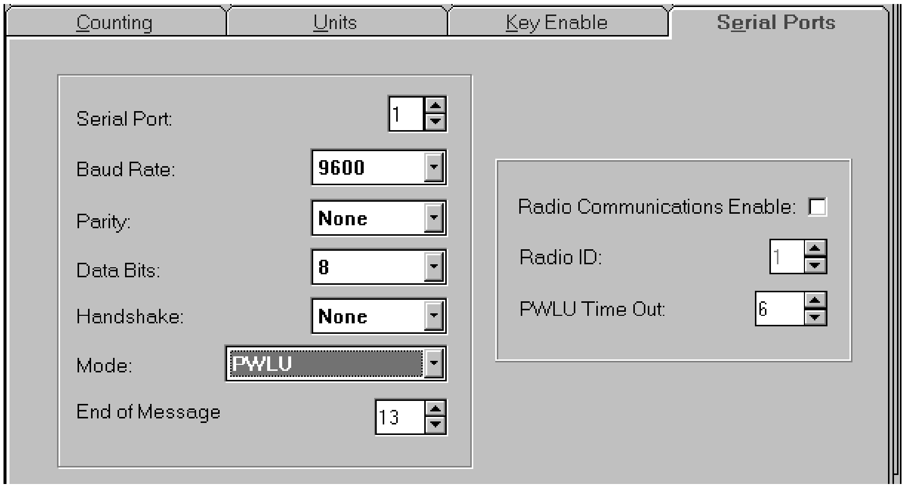
To see the radio dialog box on the right side of Figure 12 you must pick the PWLU mode on the left side.

Enable communication by checking the Enable box.

Radio ID - Use this to assign a unique number to each radio equipped PC-820. This number will correspond to the scale number on the PC.

PWLU Tim Out - An error message is generated after six seconds (example shown) if no radio communication occurs after a command is sent.

Serial Ports is the next tab, shown in Figure 12.



Serial Port: 1

Baud Rate: 9600

Parity: None

Data Bits: 8

Handshake: None

Mode: PWLU

End of Message 13

Radio Communications Enable: □

Radio ID: 1

PWLU Time Out: 6

Figure 12

Serial ports tab (see note at left)

The Serial Ports dialog box allows the user to set the parameters for the two PC-820 serial ports. Each parameter is described below.

Serial Port

This selection switches between Serial Port 1 and Serial Port 2. Use the up/ down arrow box to select the port or position the cursor inside the text box and type the number directly into the box.

Baud Rate

Use the combo box to select the Baud Rate from the list of selections.

Caution: If you use a baud rate above 19.2k, you need to use an error detection or correction protocol as well. Baud rate choices available are:

300	9600
1200	19200
2400	38400
4800	56700

Parity

Use the combo box to select the Parity setting from the list of selections. Choices available are shown in bold below:

	Stop Bits	Data Bits	Parity
None	1 or 2	7 or 8	None
Odd	1 or 2	7	Odd
Even	1 or 2	7	Even
Set	2	7	None
Clear	1	8	None

Data Bits

Use the combo box to select the Data Bits from the list of selections. Choices are 7 or 8.

CTS is a hardware handshake (ready/busy) which requires two extra wires in your cable.

Xon/Xoff is a software handshake requiring no additional hardware.

Software must support this protocol in all devices.

Handshake
Use the combo box to select the Handshake protocol from the list of selections.

Selections include:

None - No Handshake protocols are selected.
CTS - Clear to Send protocol is selected.
Xon/Xoff - Xon/Xoff protocol selected
Both - Both CTS and Xon/Xoff protocols selected.

Mode
Use the combo box to select the mode from the list of selections. The following mode selections are available:

BASIC Control - Control of the serial port is through the WT-BASIC program executing in the PC-820. When BASIC Control is selected, the End of Message box appears. Select or enter the ASCII value for the end of message character to denote the end of the serial transmission. For example, setting the end of message character to 13 would indicate that the transmission would end on the reading of a "carriage return" (ASCII character 13).

A keyboard is an input device that a PC-820 is listening to or receiving data from. You may share a serial port with a printer that just listens or receives data from the PC-820.

Keyboard - Control of the serial port is through an attached keyboard. As in Basic Control above, with this selection, an End of Message character must be entered.

Disabled - The serial port is turned off.

Dual - Dual Mode allows for 2 BASIC CONTROL devices to be connected to Port 2. A "COM2_MESSAGE" event responds to data received on Com 2 REC A, while a "COM2B_MESSAGE" responds to data received on Com 2 REC B. The only stipulation is that the 2 devices on Com Port 2 cannot send data at the same time. This mode is useful if you have 2 devices which may send data and you need to be able to differentiate the data received from the 2 devices (EX: a bar code scanner and a magnetic strip reader).

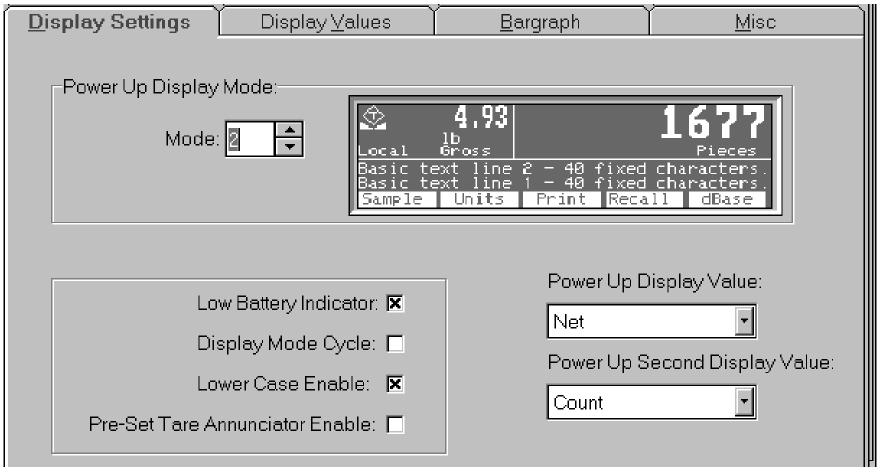
Split - Split Mode allows for a KEYBOARD and a BASIC CONTROL device to be connected to Port 2. The port works like "BASIC CONTROL MODE" for data received on Com 2 REC A, while also allowing for "KEYBOARD MODE" data to be received on Com 2 REC B. The same stipulation (2 devices on Com Port 2 cannot send data at the same time) applies as in "Dual Mode". This mode is useful if you have a keyboard and 2 other devices which may send data (EX: a keyboard and a bar code scanner on Com Port 2, along with a PC on Com Port 1).

PWLU - For use with the Piece Weight Look Up (PWLU) program available from Weigh-Tronix.

End of Message (EOM)
Enter the decimal number that represents the ASCII character the PC-820 expects as a signal for end of data transmitted to it from a communicating device such as a PLC or computer. When an EOM is received a COM1_MESSAGE or COM2_MESSAGE event is queued in the WT-BASIC program. You must then write BASIC code for the COM1_MESSAGE or COM2_MESSAGE event containing the GETCOM$ BASIC command.

Display Settings tab

Display Settings is the next tab, shown in Figure 13.



Power Up Display Mode:
Mode: 2

Low Battery Indicator: ☒
Display Mode Cycle: □
Lower Case Enable: ☒
Pre-Set Tare Annunciator Enable: □

Power Up Display Value: Net
Power Up Second Display Value: Count

See Appendix 1 for samples of displays.

Figure 13
Display Settings tab

Display modes requiring BASIC text will show blank screen space if no WT-BASIC program exists to support screen text.

There are many display modes available. The Power Up Display Mode box lets you scroll through and select the style of display you want for your application. Displays vary in size of text, numbers, type of graphing, and soft key availability.

Other items you can enable or choose are:

Low Battery Indicator	Default is Yes.
Display Mode Cycle	Default is No.
Lowercase Enable	Default is Yes.
Pre-Set Tare Annunciator Enable	Default is No.
Power-up display value	Choose from #0-13. See list below. Default is 1.
Power-up Second display value	Choose from #0-13. See list below. Default is 10.

#0	Gross
#1	Net
#2	Tare
#3	Minimum
#4	Maximum
#5	Rate of Change
#6	Gross Total
#7	Net Total
#8	Count Total
#9	Transaction Total
#10	Count
#11	Value
#12	Piece Weight
#13	A to D Counts

There are two types of TARE:

semi-automatic tare (SAT is a European term—same as push-button tare in USA)

T *SAT or push-button tare annunciator*

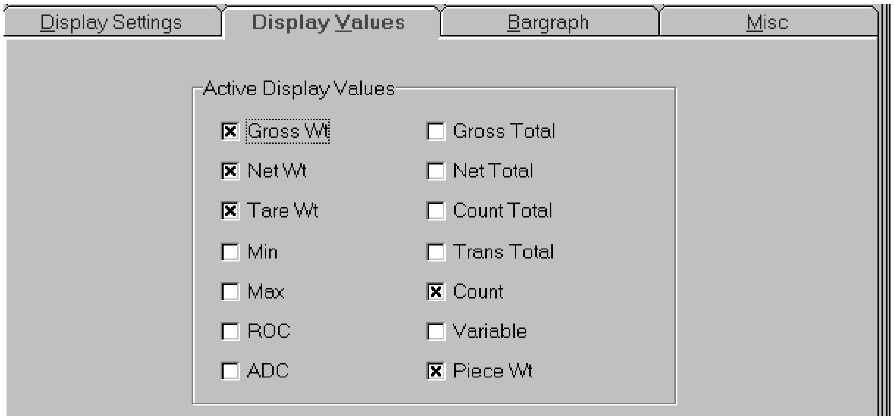
preset tare (PST is a European term—same as entered or recalled tare in USA)

PT *PST or entered/ recalled tare annunciator*

Display Values tab

Display Values is the next tab, shown in Figure 14.

ROC stands for Rate Of Change.
Min = Minimum captured weight
Max = Maximum or peak captured weight.
Variable = a value defined by WT-BASIC programming using the keyword or command (SHOWVAR)



Active Display Values

☒ Gross Wt	☐ Gross Total
☒ Net Wt	☐ Net Total
☒ Tare Wt	☐ Count Total
☐ Min	☐ Trans Total
☐ Max	☒ Count
☐ ROC	☐ Variable
☐ ADC	☒ Piece Wt

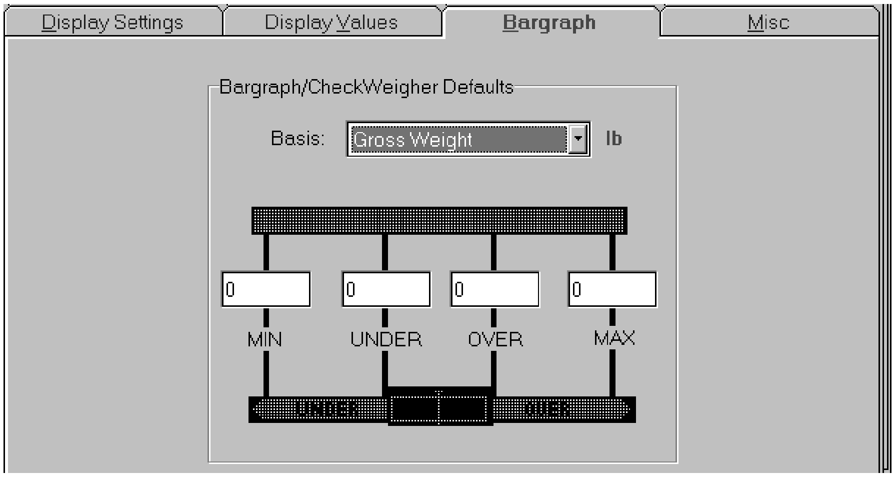
Figure 14
Display Values tab

From this dialog box, enable the types of display values you wish to be active and available during normal weighing operations. The display values you choose will show up on your screen as you repeatedly push the **SELECT** softkey on the front panel of the PC-820 during normal operation. (This assumes you have the **SELECT** softkey enabled.) Below is the list of values and their default status.

Gross	(default is Yes)
Net	(default is Yes)
Tare	(default is Yes)
Minimum	(default is No)
Maximum	(default is No)
Rate of Change	(default is No)
A to D Counts	(default is No)
Gross Total	(default is No)
Net Total	(default is No)
Count Total	(default is No)
Transaction Total	(default is No)
Count	(default is Yes)
Vaariable	(default is No)
Piece Weight	(default is Yes)

Bargraph tab

The next tab is **Bargraph**, shown in Figure 15.



Bargraph/CheckWeigher Defaults

Basis: Gross Weight lb

0 MIN

0 UNDER

0 OVER

0 MAX

Figure15

Bargraph/Checkweigher dialog box

Use this dialog box to enter parameters relating to bargraph or checkweigher functions, if the PC-820 is configured to operate in these modes. To operate in bargraph or checkweigher mode, the proper display mode must be selected.

Basis

Use the combo box to select one of thirteen choices for the Basis upon which the bargraph or checkweigher will be operating. It also is associated with the selected calibration unit selected in the Units dialog box.

Min

Enter the Minimum value of the Basis selection for which the bargraph or checkweigher will begin registering movement. If the PC-820 is set up with a bargraph, this value will determine when the graph on the display will begin moving. If the PC-820 is set up with a checkweighing display, the "Under" portion of the checkweigher graph will begin receding when this value is exceeded. The Minimum value can be set to any value, including negative values.

Under

Enter the Under value of the Basis selection. This value applies only to Checkweigher configurations and represents the Lower Acceptance Value for the checkweigher application. At this point, the "Under" portion of the checkweigher graph will disappear and the needle in the "Accept" range will be at it's extreme left position.

Over

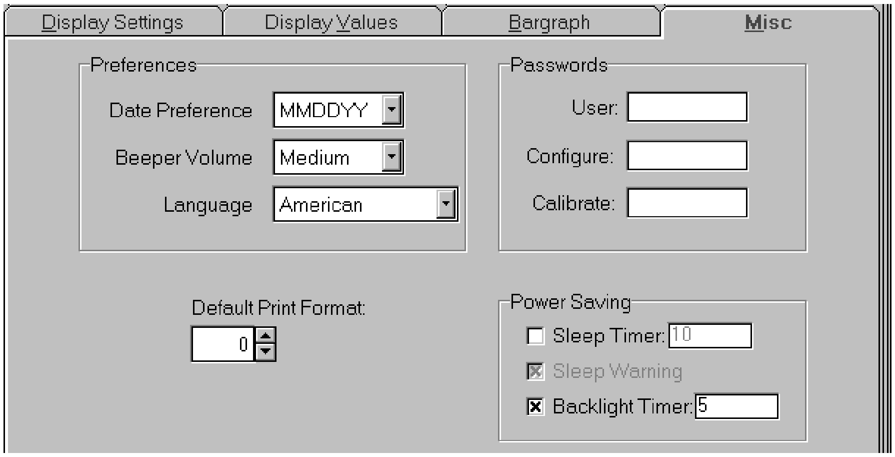
Enter the Over value of the Basis selection. This value applies only to Checkweigher configurations and represents the Upper Acceptance Value for the checkweigher application. At this point, the needle in the "Accept" range will be at it's extreme right position and the "Over" area will not yet be visible.

Max

Enter the Maximum value of the Basis selection for which the bargraph or checkweigher will end registering movement. At this point, both the bargraph and checkweigher modes reach their maximum position and do not register further movement.

Misc tab

Misc is the next tab, shown in Figure 16.



Preferences

Date Preference: MMDDYY

Beeper Volume: Medium

Language: American

Passwords

User:

Configure:

Calibrate:

Default Print Format: 0

Power Saving

☐ Sleep Timer: 10

☒ Sleep Warning

☒ Backlight Timer: 5

Figure 16
Miscellaneous tab

Record miscellaneous parameters for the configuration system in this dialog box. The following items are included:

The time and date may be sent in a variety of formats to the display, printer or computer. Format examples are: AM/PM, 24 hour, numerical reference, or spelled day and month.

Date Preference
Allows you to set the PC-820 System Clock in Month-Day-Year format, Day-Month-Year format, or Year-Month-Day format.

Beeper Volume
Allows you to turn the PC-820 internal beeper off, or select from three volume levels.

Language
This option is not implemented in the the current version of SimPoser software.

Default Print Format

Allows you to select which one of the 32 print formats you want to designate as the format used when the Print key is pressed. The default is format 0 and it is only sent to port #1.

Default format 0 outputs the following information:

G	**120000 LB**
T	**40000 LB**
N	**80000 LB**

Passwords

The next three boxes allow you to change the following passwords:

User - (Default is 111.) Allows access to basic user parameters through the PC-820 keyboard.
Configure - (Default is 2045.) Allows access to configuration parameters.
Calibrate - (Default is 30456.) Allows access to PC-820 calibration routines.

Power Saving

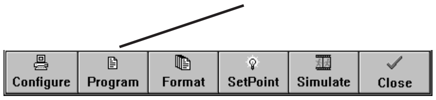
Sleep Timer — Use this to set the number of no activity minutes which will cause the unit to go into sleep mode.

Sleep Warning — If the sleep timer is enabled you can set a warning beep one minute before the sleep mode begins.

Backlight Timer — Set the length of no activity time before the backlight turns off.

Press any key to bring the unit back to operating state once it is in sleep or backlight off mode.

Program Button



The next button on the PC-820 SimPoser toolbar is **Program**. Figure 17 shows the screen which appears when you click on this button with your mouse.

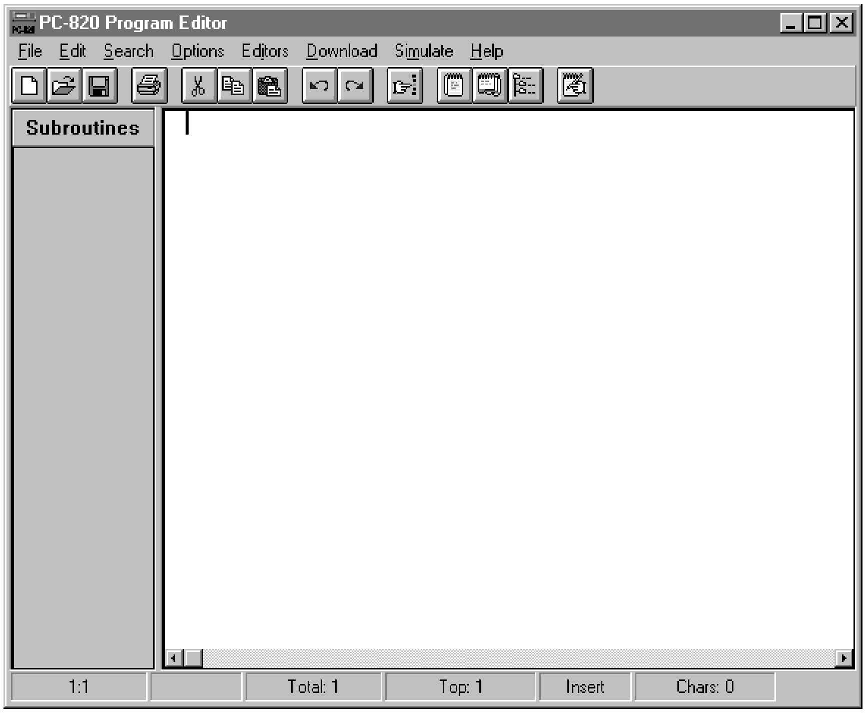


Figure 17
Program editor window

If you forgot to save your file but have either downloaded to a PC-820 indicator or have tested your file by using the simulator, a copy of your .cfg file exists in C:\wt\pc820\sim\tempcfg.820 or your installation directory.

You use the program editor window to enter the code for a WT-BASIC program. When you save the program you create in this window, all the setpoints, print formats, and configuration information you have selected are saved as a whole in a .cfg file. This file can be run in the simulator mode or downloaded to the PC-820.

WT-BASIC programming is the key feature of the PC-820's power and flexibility. Using the programming system in conjunction with the Setup configurations, print formats and the Setpoint configuration allow the PC-820 to be adapted to a wide variety of user defined applications.

Program Editor Window Commands

The Program Editor window can be maximized by clicking the maximize button in the upper right corner of the window or be resized to suit your needs by clicking and dragging a corner of the window.

The program editor window has several commands and buttons. The commands in this window are:

- File
- Edit
- Search
- Options
- Editors
- Download
- Simulate
- Help

File

The file command operates the same way as the file command on the main toolbar but it also lets you access the Windows print setup dialog box and print the text currently in the Program Editor window.

Edit

The edit command allows you to cut, copy, paste, and delete text in the editor window. You can also undo or redo actions, select all text in the window, and toggle the Event List, Key Word List, and Sub Routine List on and off. See explanations of these lists in the *Format Button* section.

Search

This command drops down a menu which helps you find and replace text, This can be very handy when the program is very long and you need to find a particular section for changes. Hot keys for these functions are F2 or Ctrl+F for Find, F3 or Ctrl+G for Find next, and F4 for Replace. When you access the Find function, the dialog box has a place to type in the text you want to find. You can cause the program to look for only exact matches to what you type in or words that contain the letters you type in.

This command also has a **Go to line** feature. This allows you to move to any line of the program simply by typing the line number in a popup dialog box.

The next feature in the search command is called **Program Errors**. This is used to retrieve a list of errors in your program. If you simulate a program and it contains an error or errors, PC-820 SimPoser creates an error file. You can retrieve this file by clicking on **Program Errors** in this drop down menu or clicking the program errors button, shown at left, on the tool bar. A dialog box pops up listing the error and the line it is on. This error file is automatically deleted when you load a new configuration file or test an updated version.

The last features under Search is **Setting Markers** and **Go To Marker**. These allow you to set up to ten markers in your program and quickly find them at a later time. The marker does not affect the program in any way.

Options

This command lets you customize the editor window and its function. Each item is briefly described below.

Auto-indentation — If you enable this function with a checkmark, your lines of text will automatically indent the same as the previous line of text.

Font — Click on this option to bring up a Font popup box. Use this to choose what type font, font style and type size is used in the editor window.

Tabs — This option lets you set three types of tabs and customize the tab size.

Fixed Tabs - Pressing **Ctrl + Tab** keys moves the cursor to the next tab position. This inserts spaces not true tabs.

Real Tabs - Pressing **Ctrl + Tab** keys moves the cursor to the next tab position. This inserts real tabs into the text, not spaces.

Smart Tabs - Pressing **Ctrl + Tab** keys aligns the cursor on the current line to the position of the next closest text on the previous line. This inserts spaces not true tabs.

Tab Size - Choose a tab size.

Editors

Click this command to access the other editing windows without returning to the main toolbar.

Download

Click this command to download the program to your PC-820. You are given a choice of which port to use. The F11 and F12 keys are hot keys for downloading to Com1 and Com2 respectively.

Simulate

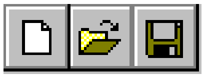
Choose this command to start the PC-820 simulation using the program you are working on.

Help

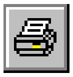
Choose this command to open help documentation on PC-820 SimPoser operation.

Program Editor Window Buttons

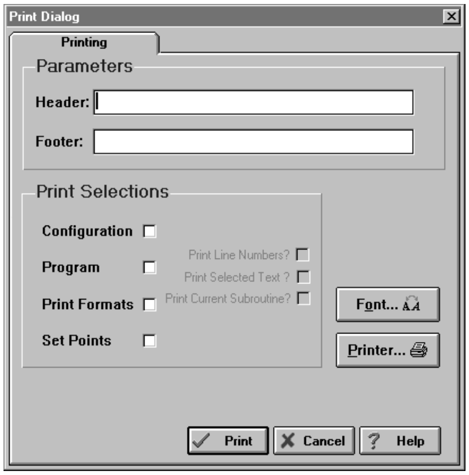
There are several buttons on the button bar of the editor window. Below is a description of each.



These three buttons are for opening a new file, opening an existing file or saving a file you are currently working on.



This button is for printing the program. The following dialog box appears when you click on this button:



Print Dialog ☒

Printing

Parameters

Header:

Footer:

Print Selections

Configuration □

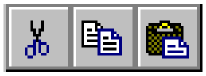
Program □ Print Line Numbers? □

Print Selected Text ? □

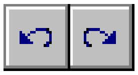
Print Formats □ Print Current Subroutine? □

Set Points □

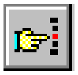
You can insert a header and footer into the printout. You can also choose what information to print: Configuration; Program; Print Formats; and Set Points.



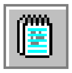
These three buttons are for cutting, copying and pasting text.



These two buttons are for undoing and redoing an action.



This button brings up a dialog box for finding text.

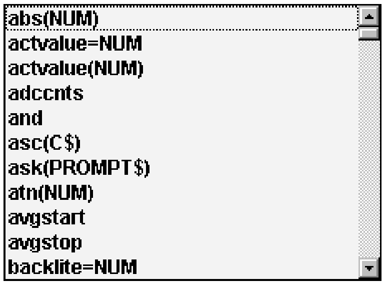


This button is for toggling the event list on and off. Use the elevator button or the arrow keys to scroll through the entire list of events available to you.

This is a list of predefined events and subroutines you can use in your program. Double click an event name to place it at the end of your program. If the event name already exists in your program, double clicking the event name will cause the event to be found and highlighted in the program. After the event name is placed in your program you need to fill in your particular commands.

Teaching BASIC programming language is beyond the scope of this manual. There are many good reference books on the subject. All the WT-BASIC commands available to you in the PC-820 SimPoser program are listed in Appendix 2: The WT-BASIC Interpreter Command Set. *Use these commands to build your program.*

This button is for toggling the WT-BASIC keyword list on and off. Below are some of the items in this list. The entire list is found in *Appendix 2* of this manual.

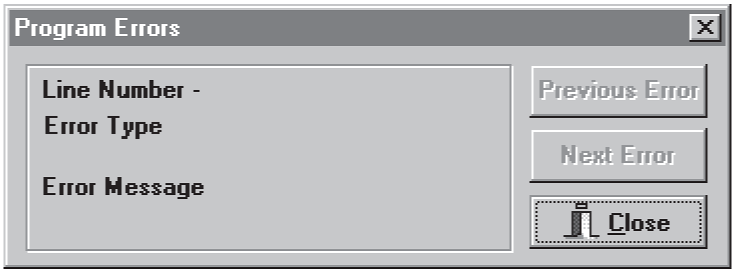


abs(NUM)
actvalue=NUM
actvalue(NUM)
adccnts
and
asc(C$)
ask(PROMPT$)
atn(NUM)
avgstart
avgstop
backlite=NUM

When you double click a selection, it will appear in your program where your cursor is located. You will need to replace the syntax placeholders with your values or strings.

This is a list, which appears along the left side of the editing window, containing currently used event names and self-created subroutine names that are currently in your program. The Subroutine window on the editor screen will appear or disappear when this key is pressed. By double clicking on an event or subroutine name in this list, your cursor is relocated to the beginning of that subroutine.

This button is for bringing up the program error dialog box shown below. This box will only exist after an error is found in the program during simulation. When the simulation is canceled, this box will appear in the Program Editor window. If your program has multiple errors, you can move through them using the Previous Error and Next Error buttons. You can close this window by clicking the Close button. You can make it reappear by clicking the Program Error button again or clicking on Program Errors under the Search command at the top of the Program Editor window.

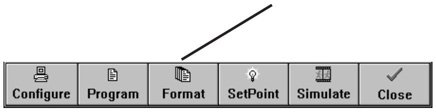


Program Errors

Line Number -
Error Type

Error Message

Format Button



The next button on the toolbar is **Format**. Use this to control the way a particular printer connected to a PC-820 will print a document or bar code label. The program is capable of 32 output formats.

Click the format button and the format editing window shown in Figure 18 appears.

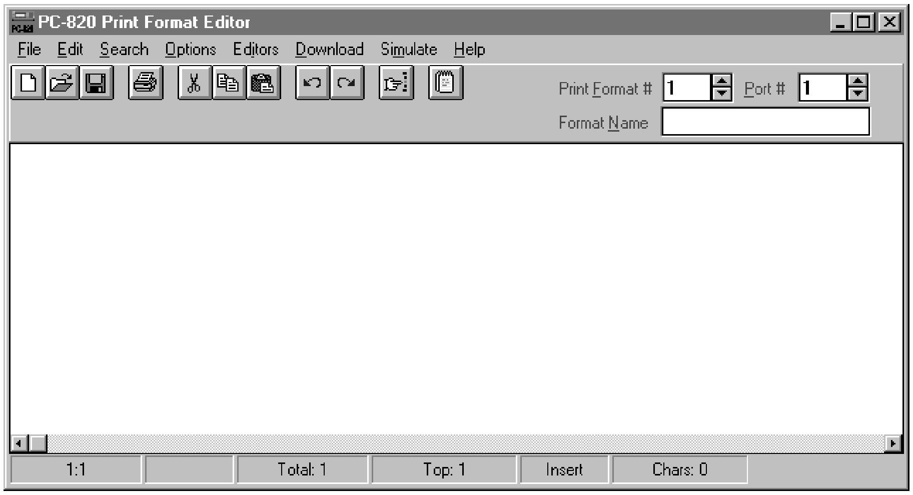

PC-820 Print Format Editor
File Edit Search Options Editors Download Simulate Help
Print Format # 1
Port # 1
Format Name
1:1
Total: 1
Top: 1
Insert
Chars: 0

Figure 18
Print format editing window

All the commands along the top of the window are described in *Program Editor Window Commands*.

Choose the number of the print format you want to create or work with. Choose which port you want it printed from and give the format a name by typing it in the Format Name box. You may use any number of print formats, from 1 to 32, with each configuration file. You do not have to use them in order: it is possible to use Format 1, Format 9 and Format 15, for example, or any other combination you desire.

In a WT-BASIC program, actual values that can change during the process are represented by “VARIABLE NAMES”.
In WT-BASIC there are variables that are predefined. These are referred to as “system variables”, such as “gross weight”, “count”, etc. These are available under TERMS. You may also create and use your own variables in WT-BASIC.

To get the current unit of measure label to print after a weight value, place the system variable {curunit$} after the callout.

***\S** may be used in a print format to comment a section out or prevent a carriage return from being sent. Any character to the right of a **\S** will not be printed.*

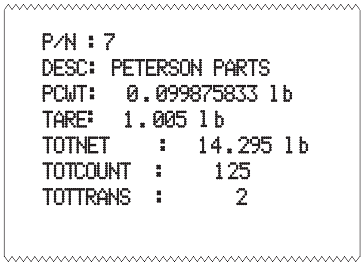
For printed tickets, you use the editor to lay out a format in logical order. The example in Figure 19 shows a typical truck scale ticket format, with item legends on the left side of the ticket and the actual values for the variables to be printed (in brackets) on the right side. As you can see in the adjacent sample ticket, the format looks very much like the actual ticket that is printed.

Example

Format

```
P/N  :  {tpart$}\n\r\S
DESC:  {tdesc1$}\n\r\S
PCWT:  {tpcwt$}  {curunit$}\n\r\S
TARE:  {ttare$}  {curunit$}\n\r\S
TOTNET    :  {tnaccum}
{curunit$}\n\r\S
TOTCOUNT:  {tcaccum}\n\r\S
TOTTRANS:  {ttrannum}\n\n\r\S
```

Actual Ticket



```
P/N : 7
DESC: PETERSON PARTS
PCWT:  0.099875833 lb
TARE:  1.005 lb
TOTNET    :  14.295 lb
TOTCOUNT  :    125
TOTTRANS  :      2
```

Figure 19
Print format example

Printing Titles and Legends
For fixed legends or titles, type the information in position in the edit area of the screen, exactly as you want to see it on the finished ticket. Use the space bar to move to the desired position and the **ENTER** key for additional lines. Lay out the ticket to match the design you desire. Once you have some of the information laid out you may move around the screen using the cursor arrow keys on your keyboard.

Printing Actual Values
The actual values that will be printed on the ticket when run with the PC-820 require a different method of placement. Values from the PC-820 must be surrounded with brackets ({ }) to tell the Print Format that this information will be coming from data stored, generated or collected by the PC-820. Information such as scale weights, accumulated total, transaction counts, piece weights, part counts, and Variables must be enclosed in brackets in the Print Format. This is done automatically when using the TERMS list box shown on the previous page.

Depending on the type of printer you use, you may have to use the following commands to make it respond correctly:

\a -	*07 Output Bell character*
\b -	*08 Output Back Space character*
\f -	*0C Output Form feed character*
\n -	*0A Output Line Feed character*
\r -	*0D Output Carriage return character*
\t -	*09 Output Horizontal Tab character*
\v -	*0B Output Vertical Tab character*
\S -	*Skip to the end of the string.*
\\ -	*Print slash*
\' -	*Print '*
\" -	*Print "*
\? -	*Print ?*
\{ -	*Print {*
\xHH -	*Output character with ASCII value HH*

See Appendix 5 for ASCII chart.

Example: {GROSS, 3.2}

This will print the gross weight as follows: 500.01

Selecting Actual Values

An alternative to typing in actual values is to use the terms list. Click the terms list button (📋)to make the list appear.

1. Place the cursor in the position on the screen where you want to place a new value.

2. Double click with the left mouse button on the value item you want to place in your format.

The term, in its correct syntax, is placed into the print format.

3. Repeat steps 1 and 2 until you have the terms you want in the window.

4. To remove the list box, click the right mouse button with the mouse pointer inside the list box or click on the terms button in the tool bar.

The list contains several "format codes" such as backspace, line feed, carriage return, beep and others designed for special functions when the ticket is printed.

Two selections, "Numeric Variable" and "String Variable" require that you replace the words ‘Numeric Variable’ or ‘String Variable’ with the actual variable name that you created in WT-BASIC.

Defining the way numeric variables are printed is accomplished by the following syntax:

SYNTAX: **{VARIABLE,WIDTH.PRECISION}**
See the example to the left or the print format example on the previous page.

Width and Precision are optional expressions. You do not have to use them. By default, a numeric variable is right-justified. Use *width* to define a minimum width of the numeric variable. Use a negative width to left-justify the numeric variable. Use *Precision* to designate the number of positions to be printed to the right of the decimal point.

When defining a string variable, width is used to define a minimum width. If the string variable does not take the minimum width to print, spaces are printed to fill the gap. String variables are Left Justified by default and will print Right Justified if a negative width is used.

When TIME$ or DATE$ are used in a program's print statement, you may use the following syntax:
TIME$(n) *or* **DATE$(n)**

TIME$ and DATE$ are the two string system variables that have the following optional syntax in a print format:

{TIME$,0.n}

When a TIME$ is printed, (n) is used to tell the system what format of TIME$ to print. A value of 0, 1, 2, or 3 is used in the (n) expression to print TIME$ in the following formats:

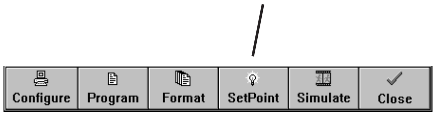
TIME FORMATS		
0 =	**24 Hour format with seconds**	**18:00:00**
1 =	**AM/PM format with seconds**	**1:00:00 AM**
2 =	**24 Hour format without seconds**	**18:00**
3 =	**AM/PM format without seconds**	**1:00 AM**

{DATE$,0.n}

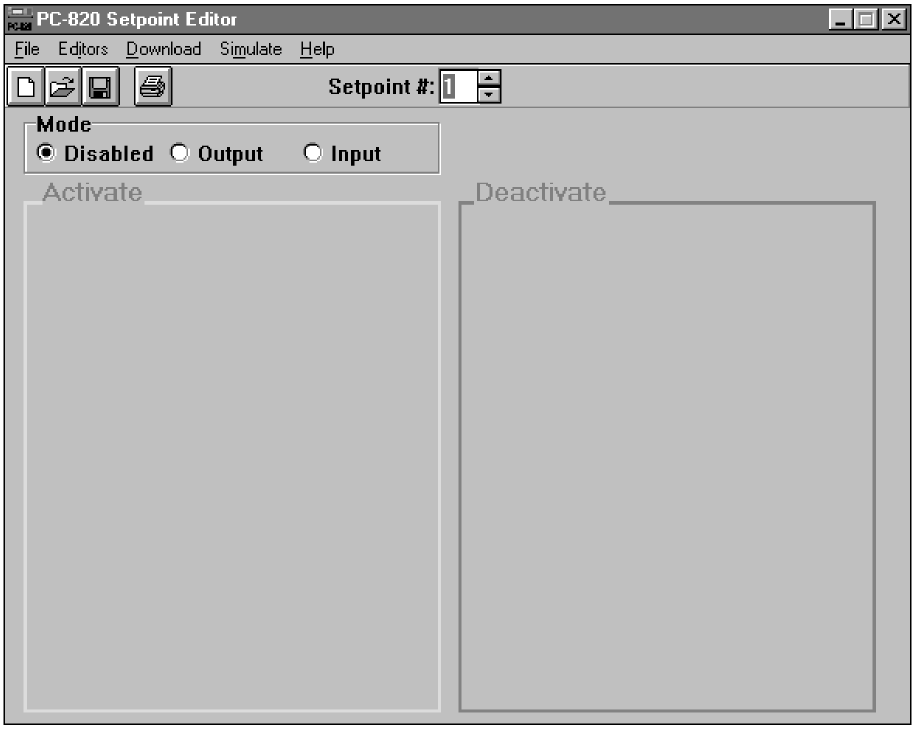
When a DATE$ is printed, (n) is used to tell the system what format of DATE$ to print. A value of 0, 1, 2, 3, or 4 is used in the (n) expression to print DATE$ in the following formats:

Date Formats	**Description**	**MM/DD/YY**	**DD/MM/YY**	**YY/MM/DD**
0	Numbers with 2-digit year	06-14-99	14-06-99	99-06-14
1)	Spelled Month	Jun 14, 1999	14 Jun, 1999	1999 Jun 14
2)	Numbers with day of week	Mon 06-14-99	Mon 14-06-99	Mon 99-06-14
3)	Spelled Month with day of week	Mon Jun 14, 1999	Mon 14 Jun,1999	1999 Mon Jun 14
4)	Numbers with 4-digit year	06-14-1999	14-06-1999	1999-06-14

Setpoint Button



The next button on the PC-820 SimPoser toolbar is **Setpoint**. Press this button to configure setpoint operation. Depending on what things you choose, the window will display the parameters you need to set. The following three screens show the window as it first appears and as it may appear as you select different parameters.



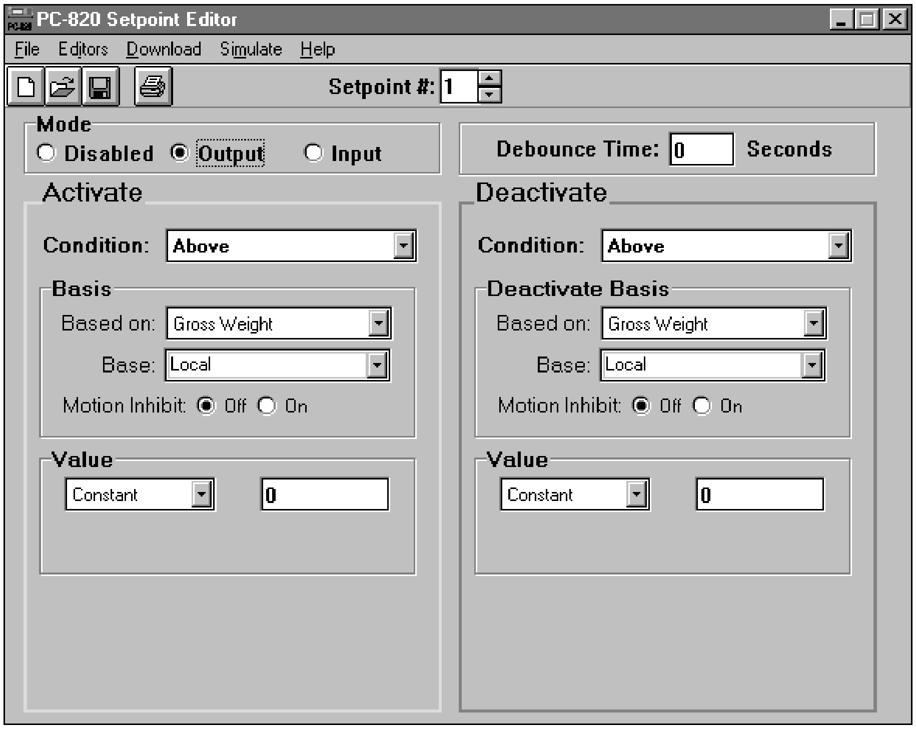
Setpoint #: 1

Mode

● Disabled ○ Output ○ Input

Activate

Deactivate



Setpoint #: 1

Mode

○ Disabled ● Output ○ Input

Debounce Time: 0 Seconds

Activate

Condition: Above

Basis

Based on: Gross Weight

Base: Local

Motion Inhibit: ● Off ○ On

Value

Constant 0

Deactivate

Condition: Above

Deactivate Basis

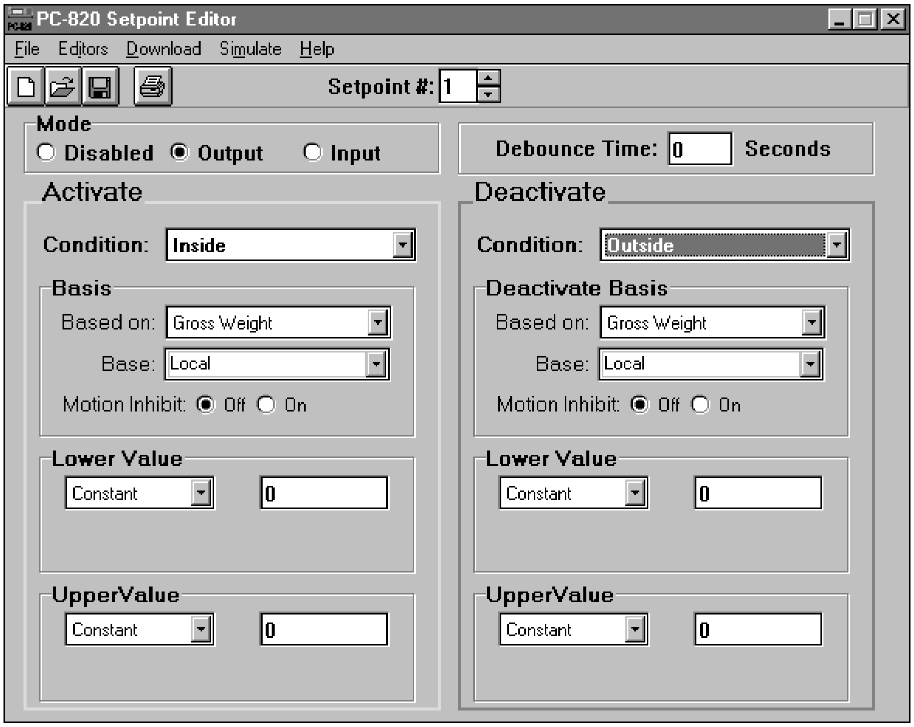
Based on: Gross Weight

Base: Local

Motion Inhibit: ● Off ○ On

Value

Constant 0



Setpoint #: 1

Mode
○ Disabled ● Output ○ Input

Debounce Time: 0 Seconds

Activate
Condition: Inside
Basis
Based on: Gross Weight
Base: Local
Motion Inhibit: ● Off ○ On
Lower Value
Constant 0
UpperValue
Constant 0

Deactivate
Condition: Outside
Deactivate Basis
Based on: Gross Weight
Base: Local
Motion Inhibit: ● Off ○ On
Lower Value
Constant 0
UpperValue
Constant 0

The setpoint window has file commands and buttons which are identical to those explained earlier in this manual.

PLEASE NOTE: Applications using setpoints should be handled and tested with great care to assure that the system operates in conformance with the stated objectives and design parameters of the system. Always completely test all parameters to the fullest! Manual back up controls should always be installed on the most critical components to assure that the system can be adjusted and/or shut down manually, should the system malfunction or deviate from the proper operation sequence.

The setpoint window allows you to define the condition you wish to monitor or detect thus causing the PC-820 program to trigger a new event (SetPoint Act or Deact).

The Setpoint capabilities of the PC-820 range from very simple applications such as turning on an alarm when a weight value is reached, up to very sophisticated batching applications specifically tailored to a customer's requirements. The setpoint system is designed for almost limitless flexibility, to allow the application designer to use the system in conjunction with WT-BASIC programming to solve a wide variety of scale user requirements.

Not limited to batching, however, the PC-820 Setpoint System is adept at solving all types of application needs, such as truck scale control, check-weighing systems, conveyor systems, PLC applications or any other situation where external feedback and control is required.

Use the setpoint window to configure up to 32 setpoints in the PC-820 System. This dialog screen provides a visual representation of the configuration of a particular setpoint so that you can quickly glance at the form and see how the setpoint is configured.

The screen is separated into the following areas:

- Setpoint #
- Mode
- De-Bounce Time In Seconds
- Activate and De-Activate boxes

Each of these areas of the screen is explained below.

Setpoints are commonly used as outputs without I/O modules for program interactions like changing the display, continuous output, or diagnostic flags.

SetPoint #

Determines which setpoint is currently active. Use the up/down arrow box to select the active setpoint or enter the number directly into the text box. You may not enter a number smaller than 1 or greater than 32.

Mode

Mode determines what level of operation the setpoint is in. The possible choices are:

- Disabled - Setpoint not active
- Input - Setpoint receives input from external device
- Output - Setpoint sends signal to external device or setpoints are commonly used as outputs without I/O modules for program interactions like changing the display, continuous output, or diagnostic flags.

Disabled

This is the default state for a setpoint and means that the setpoint is not in use with the current configuration.

Input

Select Input to receive a signal from an external device, such as a switch. The state of an Input setpoint can be detected using commands in the WT-BASIC programming system. When this mode is selected, the only parameter to choose is De-Bounce Time. De-Bounce time is the time in seconds allotted to receive switch activations, to prevent receiving a double signal. For example, if a De-Bounce time of 1 is entered in the text box, the Input setpoint will only receive one activation input per second, no matter how many activations of the switch occur during the 1 second period.

Output

Select Output to send a signal to an external device to activate or deactivate the device. Output requires setting various parameters to achieve the proper control of external equipment.

De-Bounce

When the setpoint is used as an output, the debounce time can be used as a timer interval for events to occur by selecting IMMEDIATE in the Activate and Deactivate drop down boxes.

For example, with Activate/Deactivate both set to immediate, and De-Bounce set to 1.0 seconds, the setpoint activates or turns on for one second, then deactivates or turns off for one second, then turns back on for one second. The time between activate events is two seconds.

Activate and Deactivate boxes

Each Output setpoint must have a condition that activates and deactivates the setpoint. By clicking the combo box next to Activate and Deactivate, a list is displayed for you to select this condition. You must select both an Activate and Deactivate Condition in order for the setpoint to work properly. Make sure you fill in both sides of the form.

The following conditions are available:

- Above
- Below
- Inside
- Outside
- Unstable
- Stable
- Center of Zero
- Not Center of Zero
- BASIC Control
- Immediate
- Accum Operation
- Print Operation
- Zero Operation
- Tare Operation
- Tare Key
- Zero Key
- Equal
- AND
- OR
- XOR

In addition to the conditions defined by the setpoint window, you may also force the setpoint on or off from you BASIC program by using the BASIC commands SETPTON or SETPTOFF.

Selection of the above conditions determine the additional parameters that will be required to make the setpoint functional. Each of these will be described in the following sections and each section will apply to both Activate and Deactivate.

1. Above/Below

Above/Below indicates that the setpoint will activate or deactivate when the scale reading is either Above or Below the selected value. When either of these selections is made, the "Lower Value" box on the screen will become active.

Lower Value
Indicates the value that the setpoint will activate on. Use the mouse to click on the selection in the Lower Value box.

Constant - Indicates that the value will be a fixed value, using the same standard as the calibration Unit of Measure selected for the system, such as 5000 lbs. When Constant is selected, the text box to the right (Value box) becomes active. Enter the constant value in this box.

Percent - Indicates that the value will be a percentage of a Variable amount. When selected, the Variable text box is enabled. Enter the percentage in the box to the right of the form (Value box) and the Variable Name in the box under "Variable". Variables can be used in the WT-BASIC program to control operations of the setpoint system. You may enter both positive and negative percentage amounts and amounts greater than 100% (i.e. 1000%)

Offset - Indicates that the value of a Variable from your BASIC program, plus this offset value in cal units, will control this setpoint.

For example, the value may be 100 lbs. more than the value currently contained in variable "TestAmount" (a variable defined in the WT-BASIC program). For this example, you would enter 100 in the Value text box and the name "TestAmount" in the Variable text box. You may use both positive and negative Offset amounts.

Select Basis
The Basis defines the Weight or other value that the selected setpoint will activate on. By clicking the Combo box next to Select Basis, a list is displayed for you to select from. Depending on your selection, additional selections in the Select Basis box will be enabled for you to select from.

Gross Weight, Net Weight, Tare Weight, Minimum Weight, Maximum Weight, Rate of Change, Gross Weight Total, Net Weight Total, Count
If one of the above Basis selections is made, you need to make the following additional selections from the Select Basis box:

Scale Number - Enter the scale number that this selection will apply to in the Text box.

Motion Inhibit - Click On to enable Motion Inhibit, or Off to disable Motion Inhibit.

Count Total, Transaction Total, Piece Weight

If one of the above Basis selections is made, you need to make the following additional selections from the Select Basis box:

Motion Inhibit - Click On to enable Motion Inhibit, or Off to disable Motion Inhibit.

Variable

If the above Basis selection is made, you need to make the following additional selection from the Select Basis box:

Variable - Enter the name of a Variable used in the WT-BASIC program for this application.

Motion Inhibit - Click On to enable Motion Inhibit, or Off to disable Motion Inhibit.

2. Inside/Outside

Inside/Outside indicates that the setpoint will activate or deactivate when the scale reading is either Inside or Outside a range of selected values. When either of these selections is made, both the "Lower Value" and "Upper Value" boxes on the screen will become active allowing you to enter a range of values to meet this criteria. Make your selections in the same manner as described in *1. Above/Below*, but make sure you fill in selections for both the Lower and Upper Value boxes. Both will become active when either of these selections is made.

3. Unstable, Stable, Center of Zero, Not Center of Zero

The setpoint will activate or deactivate when any one of these chosen conditions is met. The only condition that has to be selected is the Scale Number. Enter the number in the adjacent text box.

4. BASIC Control, Immediate, Accum Operation, Print Operation, Zero Operation, Tare Operation, Tare Key, Zero Key

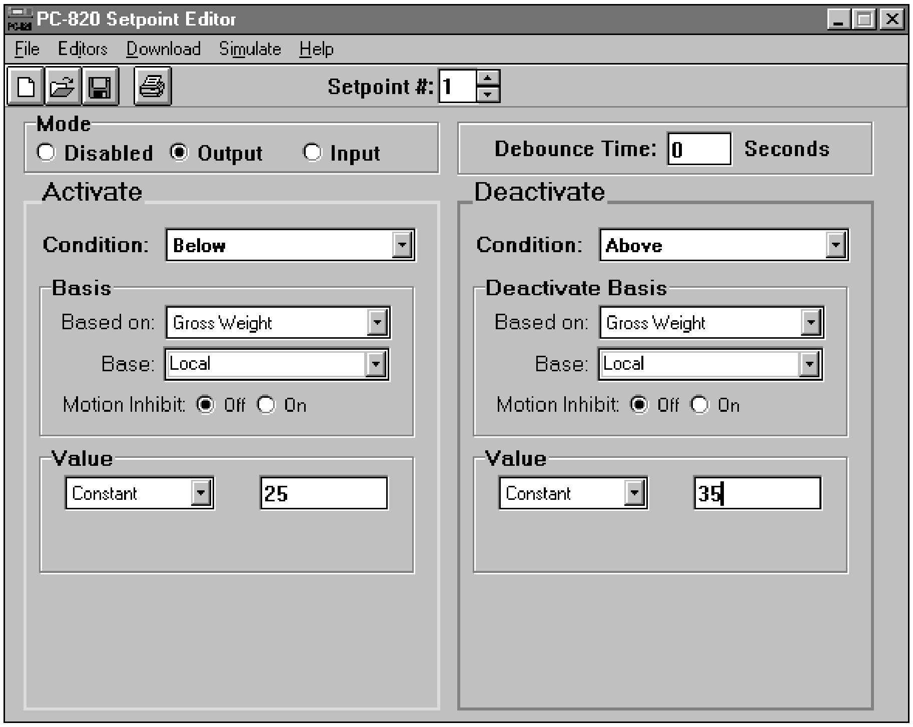
The setpoint will activate or deactivate when any of the above operations are in control or are activated, as in the case of the **Zero** key being pressed and the zero operation successfully completed. There are no additional parameters to be set when one of these conditions is selected. The entire form below Activate/De-Activate will become disabled.

5. AND, OR, XOR

Logical operators used for combining two setpoints.

Example 1:

Following are several examples of how setpoints operate. There is much more you can do with setpoints once you familiarize yourself with all the variables and conditions you can use to trigger the setpoints.



Setpoint #: 1

Mode
○ Disabled ● Output ○ Input

Debounce Time: 0 Seconds

Activate

Condition: Below

Basis
Based on: Gross Weight
Base: Local
Motion Inhibit: ● Off ○ On

Value
Constant 25

Deactivate

Condition: Above

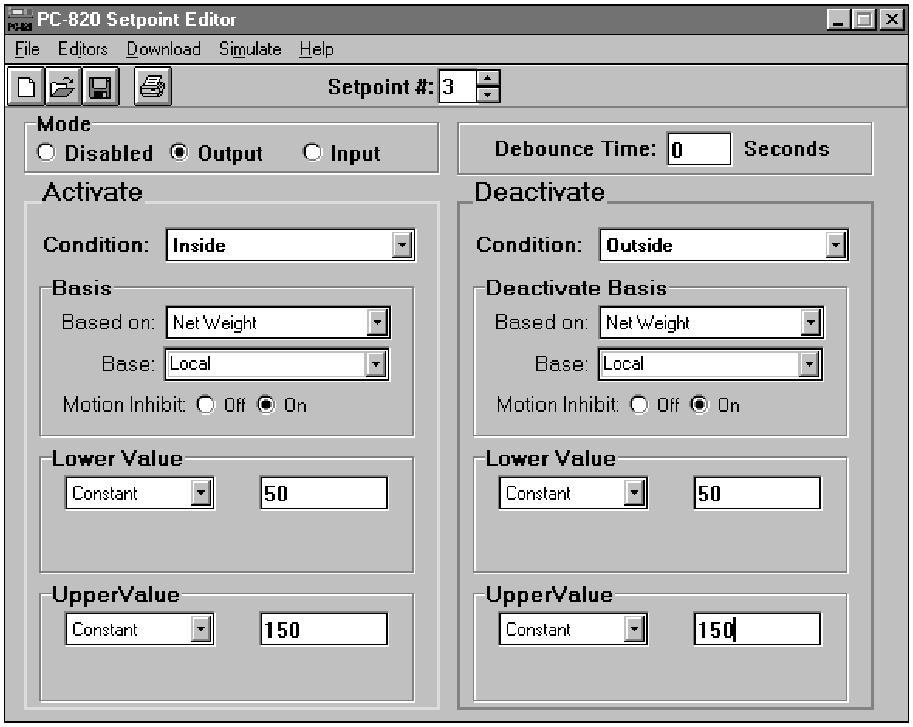
Deactivate Basis
Based on: Gross Weight
Base: Local
Motion Inhibit: ● Off ○ On

Value
Constant 35

Figure 20
Example 1

In example 1, setpoint 1 is an output. The setpoint will activate below a constant value of 25 lbs gross weight. When the gross weight goes above 35 pounds, the setpoint will deactivate. Weight readings are coming from the local scale and the motion inhibit is off.

Example 2



Setpoint #: 3

Mode
○ Disabled ● Output ○ Input

Debounce Time: 0 Seconds

Activate

Condition: Inside

Basis
Based on: Net Weight
Base: Local
Motion Inhibit: ○ Off ● On

Lower Value
Constant 50

UpperValue
Constant 150

Deactivate

Condition: Outside

Deactivate Basis
Based on: Net Weight
Base: Local
Motion Inhibit: ○ Off ● On

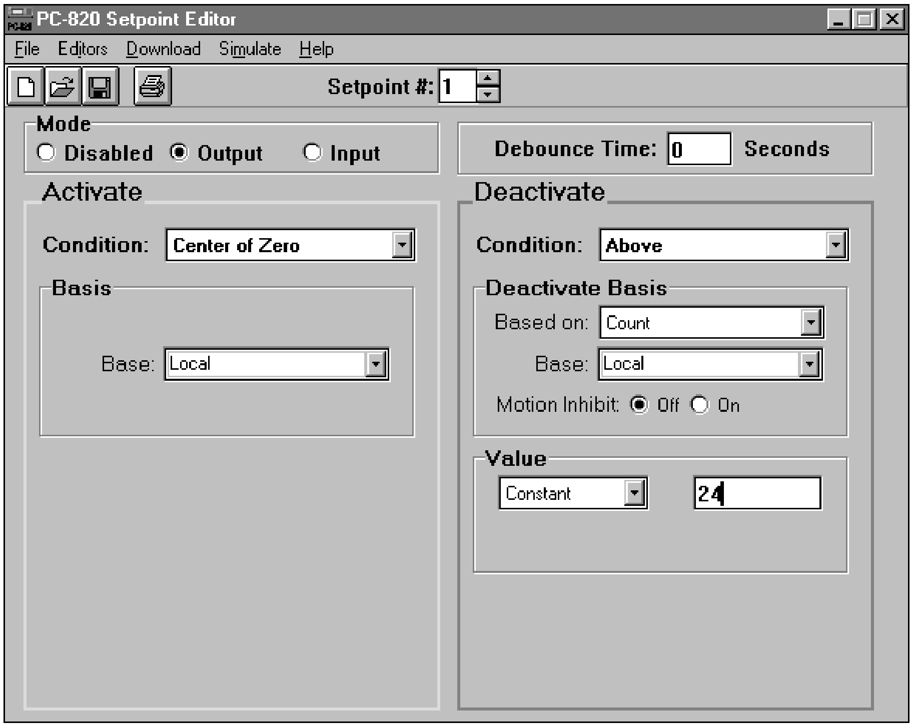
Lower Value
Constant 50

UpperValue
Constant 150

Figure 21
Example 2

Figure 21 shows the setpoint window for example 2. In this example we have chosen Setpoint 3 as an output. The setpoint will activate when the net weight value is between 50 and 150 pounds and will deactivate outside of this range.

Example 3: In this example we will use two setpoints #1 (see Figure 22) and #2 (see Figure 23).



Setpoint #: 1

Mode
○ Disabled ● Output ○ Input

Debounce Time: 0 Seconds

Activate

Condition: Center of Zero

Basis

Base: Local

Deactivate

Condition: Above

Deactivate Basis

Based on: Count

Base: Local

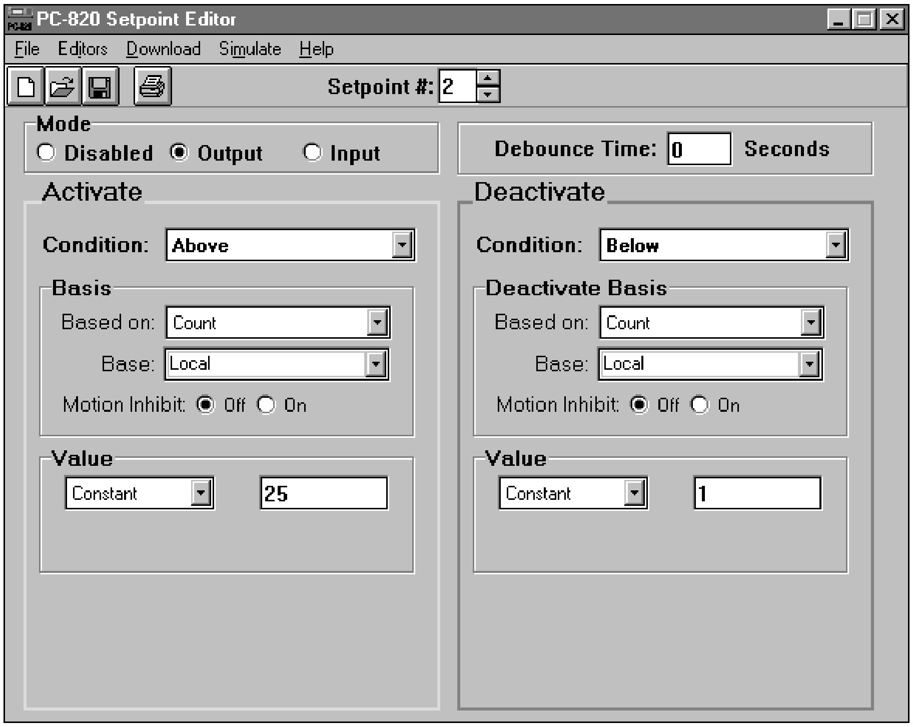
Motion Inhibit: ● Off ○ On

Value

Constant 24

Figure 22

Example 3-Setpoint #1



Setpoint #: 2

Mode
○ Disabled ● Output ○ Input

Debounce Time: 0 Seconds

Activate

Condition: Above

Basis

Based on: Count

Base: Local

Motion Inhibit: ● Off ○ On

Value

Constant 25

Deactivate

Condition: Below

Deactivate Basis

Based on: Count

Base: Local

Motion Inhibit: ● Off ○ On

Value

Constant 1

Figure 23

Example 3 - Setpoint #2

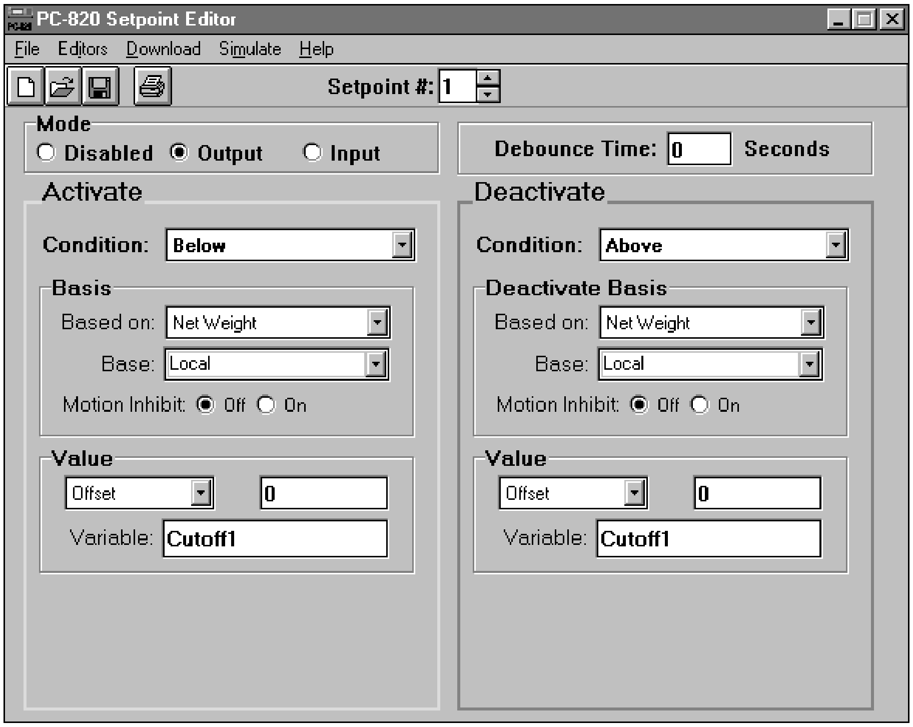
In this third example we are running a **very** simplified batching sequence. We are opening a valve or gate to drop gumballs into a hopper on a scale. We want to count out 25 gumballs and dump the hopper, then repeat the process. (Keep in mind that to run a batching procedure in real life it will probably be necessary to use a WT-BASIC program in conjunction with the setpoint configuration.)

Setpoint 1 controls the valve or gate to allow gumballs to roll in to the hopper. It is set up as an output and activates when the gross weight is at center of zero. The gumballs will roll into the hopper until the count is over 24. The valve will shut and the next setpoint (#2) which controls the dump gate on the scale hopper will activate since the count is now 25 or above. When the count on the scale hopper drops below 1, the gate closes and the fill valve reopens.

As was stated earlier, this is an extremely simplified batching program. With a WT-BASIC program you can design very sophisticated batching sequences.

Example 4

Figure 24 shows two screens using variables. The explanation for these is below.



Setpoint #: 1

Mode: ○ Disabled ● Output ○ Input

Debounce Time: 0 Seconds

Activate

Condition: Below

Basis

Based on: Net Weight

Base: Local

Motion Inhibit: ● Off ○ On

Value

Offset 0

Variable: Cutoff1

Deactivate

Condition: Above

Deactivate Basis

Based on: Net Weight

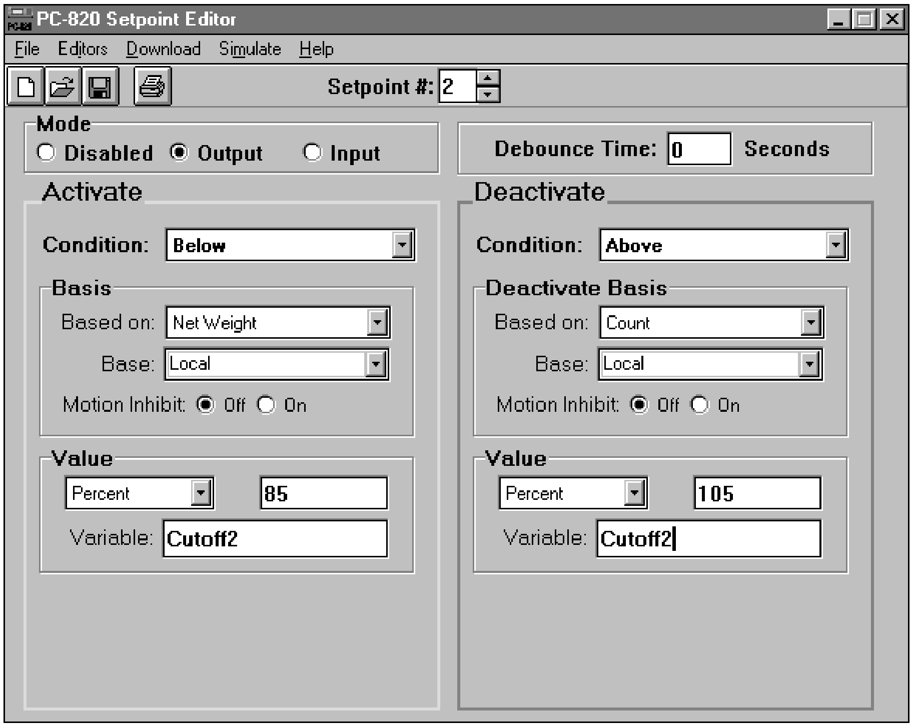
Base: Local

Motion Inhibit: ● Off ○ On

Value

Offset 0

Variable: Cutoff1



Setpoint #: 2

Mode: ○ Disabled ● Output ○ Input

Debounce Time: 0 Seconds

Activate

Condition: Below

Basis
Based on: Net Weight
Base: Local
Motion Inhibit: ● Off ○ On

Value
Percent 85
Variable: Cutoff2

Deactivate

Condition: Above

Deactivate Basis
Based on: Count
Base: Local
Motion Inhibit: ● Off ○ On

Value
Percent 105
Variable: Cutoff2

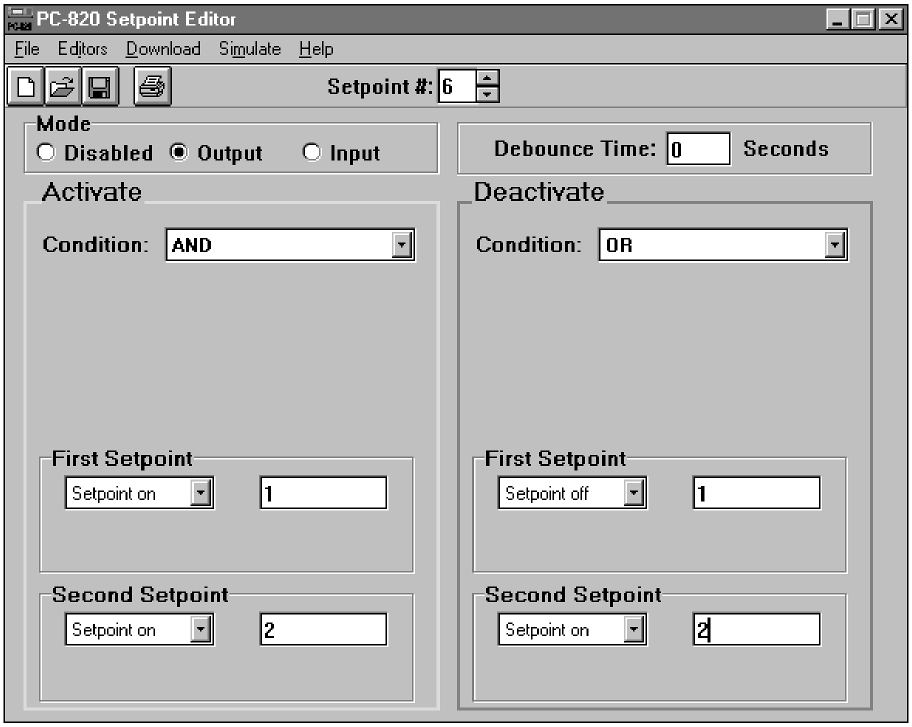
Figure 24
Example 4

In this fourth example we are controlling two setpoint outputs as cutoffs, similar to the way older indicators have worked. The first screen in Figure 25 refers to Cutoff1 as the variable name for control of setpoint #1. The second screen refers to Cutoff2 as the variable name for control of setpoint #2. Both variable names will have values assigned to them in the WT-BASIC program via the front panel.

Setpoint #1 activates when the net weight is below an offset of 0 of the variable Cutoff1. It deactivates when the net weight is above an offset of 0 of the variable Cutoff1.

Setpoint #2 activates when the net weight is below 85% of the variable Cutoff2. It deactivates when the net weight is above 105% of the variable Cutoff2.

Example 5



PC-820 Setpoint Editor

Setpoint #: 6

Mode
○ Disabled ● Output ○ Input

Debounce Time: 0 Seconds

Activate

Condition: AND

First Setpoint
Setpoint on 1

Second Setpoint
Setpoint on 2

Deactivate

Condition: OR

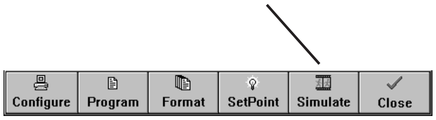
First Setpoint
Setpoint off 1

Second Setpoint
Setpoint on 2

Figure 25
Example 5

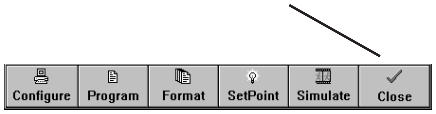
In this example we use the AND and OR conditions. Setpoint #6 will activate when Setpoints 1 AND 2 are on and deactivate when either Setpoint 1 OR 2 is off.

Simulate Button



The next toolbar button is **Simulate**. Click this button to start the PC-820 simulation. This is the same as clicking the Simulate command on the command line. This was covered earlier in this manual.

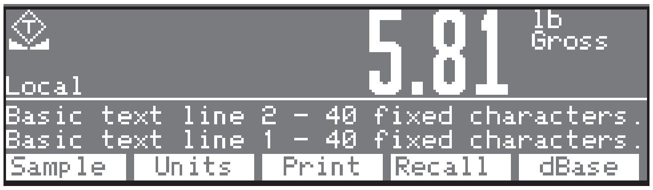
Close Button



The last toolbar button is **Close**. Click this to close your PC-820 SimPoser program. If you have made any changes in a program, a dialog box will pop up asking if you want to save the changes.

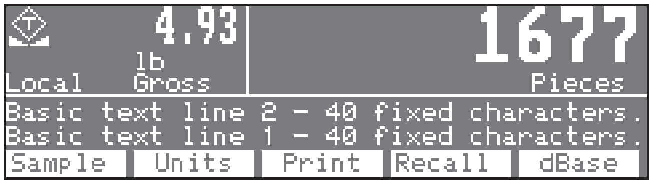
Appendix 1: Display Samples

Mode 1



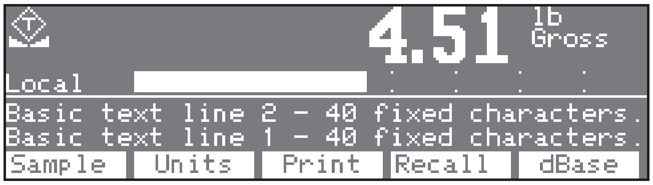
```
                    5.81  lb
                          Gross
Local
Basic text line 2 - 40 fixed characters.
Basic text line 1 - 40 fixed characters.
```

Mode 2



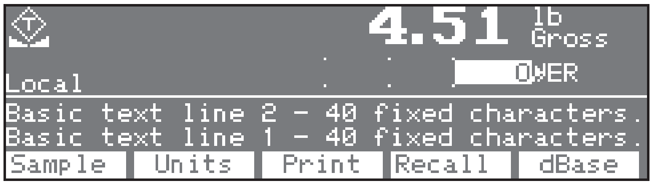
```
          4.93 |               1677
          lb   |
Local     Gross|             Pieces
Basic text line 2 - 40 fixed characters.
Basic text line 1 - 40 fixed characters.
```

Mode 3



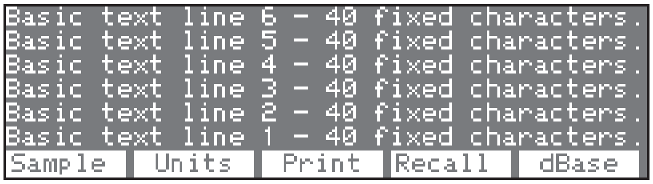
```
                    4.51  lb
                          Gross
Local
Basic text line 2 - 40 fixed characters.
Basic text line 1 - 40 fixed characters.
```

Mode 4



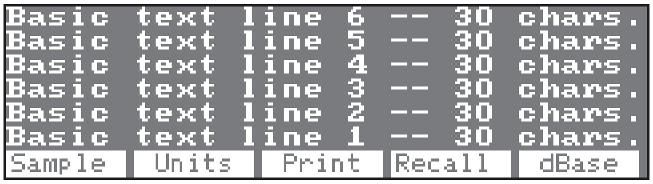
```
                    4.51  lb
                          Gross
                              OVER
Local
Basic text line 2 - 40 fixed characters.
Basic text line 1 - 40 fixed characters.
```

Mode 5



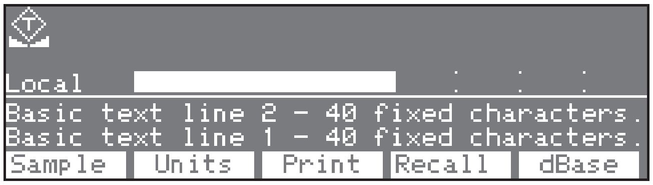
```
Basic text line 6 - 40 fixed characters.
Basic text line 5 - 40 fixed characters.
Basic text line 4 - 40 fixed characters.
Basic text line 3 - 40 fixed characters.
Basic text line 2 - 40 fixed characters.
Basic text line 1 - 40 fixed characters.
```

Mode 6



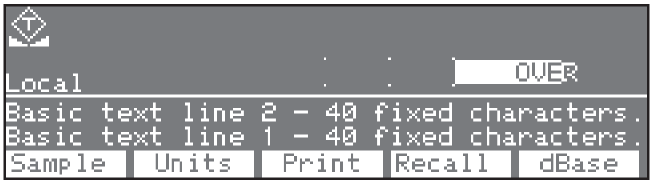
```
Basic text line 6 -- 30 chars.
Basic text line 5 -- 30 chars.
Basic text line 4 -- 30 chars.
Basic text line 3 -- 30 chars.
Basic text line 2 -- 30 chars.
Basic text line 1 -- 30 chars.
```

Mode 7



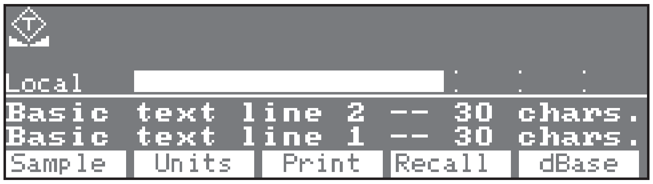
```
Local
Basic text line 2 - 40 fixed characters.
Basic text line 1 - 40 fixed characters.
```

Mode 8



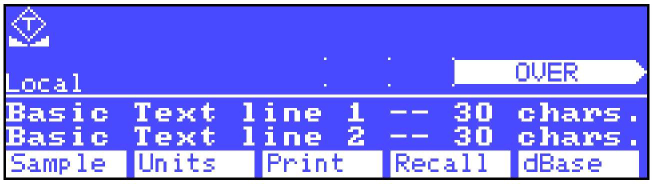
```
                              OVER
Local
Basic text line 2 - 40 fixed characters.
Basic text line 1 - 40 fixed characters.
```

Mode 9



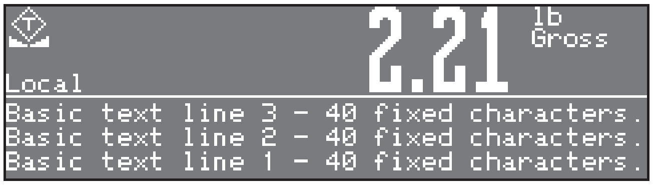
```
Local
Basic text line 2 -- 30 chars.
Basic text line 1 -- 30 chars.
```

Mode 10



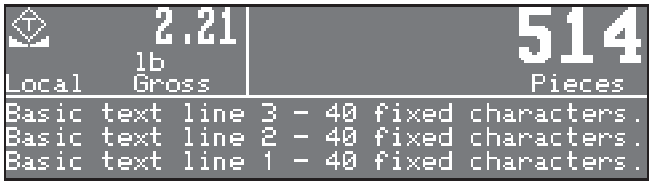
```
                              OVER
Local
Basic Text line 1 -- 30 chars.
Basic Text line 2 -- 30 chars.
```

Mode 11



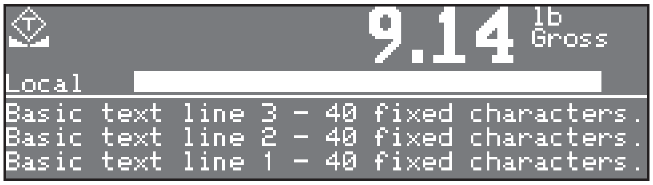
```
2.21 lb
     Gross
Local
Basic text line 3 - 40 fixed characters.
Basic text line 2 - 40 fixed characters.
Basic text line 1 - 40 fixed characters.
```

Mode 12



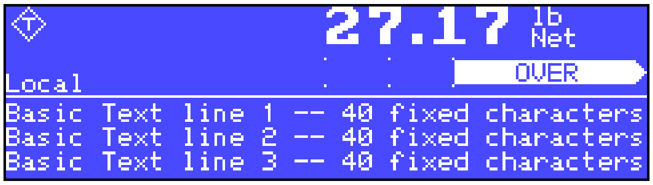
```
2.21                514
lb
Local Gross         Pieces
Basic text line 3 - 40 fixed characters.
Basic text line 2 - 40 fixed characters.
Basic text line 1 - 40 fixed characters.
```

Mode 13



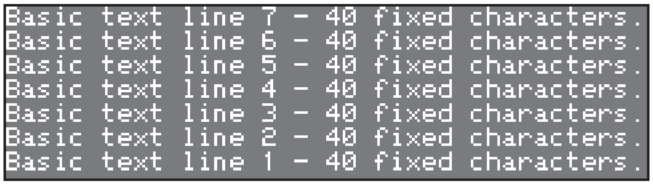
```
9.14 lb
     Gross
Local
Basic text line 3 - 40 fixed characters.
Basic text line 2 - 40 fixed characters.
Basic text line 1 - 40 fixed characters.
```

Mode 14



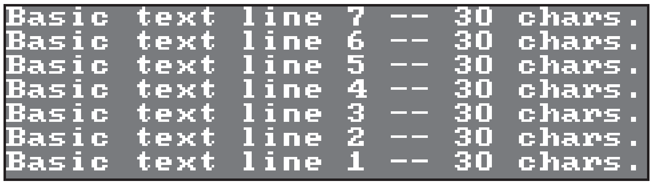
```
27.17 lb
      Net
Local            OVER
Basic Text line 1 -- 40 fixed characters
Basic Text line 2 -- 40 fixed characters
Basic Text line 3 -- 40 fixed characters
```

Mode 15



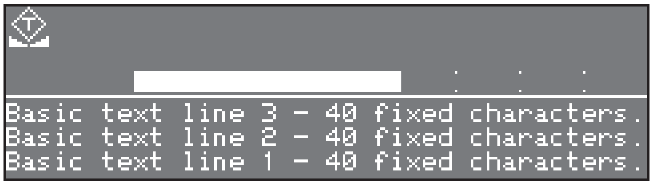
```
Basic text line 7 - 40 fixed characters.
Basic text line 6 - 40 fixed characters.
Basic text line 5 - 40 fixed characters.
Basic text line 4 - 40 fixed characters.
Basic text line 3 - 40 fixed characters.
Basic text line 2 - 40 fixed characters.
Basic text line 1 - 40 fixed characters.
```

Mode 16



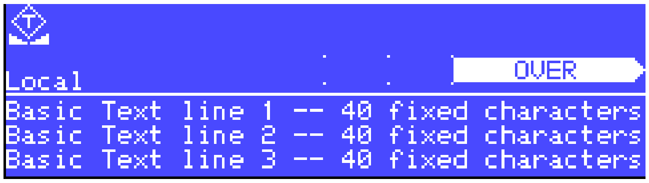
```
Basic text line 7 -- 30 chars.
Basic text line 6 -- 30 chars.
Basic text line 5 -- 30 chars.
Basic text line 4 -- 30 chars.
Basic text line 3 -- 30 chars.
Basic text line 2 -- 30 chars.
Basic text line 1 -- 30 chars.
```

Mode 17



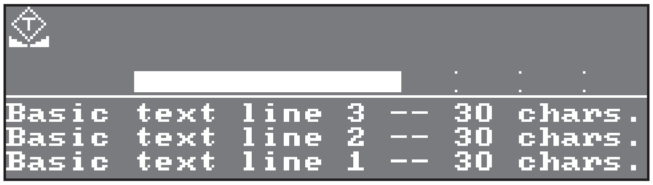
```
Basic text line 3 - 40 fixed characters.
Basic text line 2 - 40 fixed characters.
Basic text line 1 - 40 fixed characters.
```

Mode 18



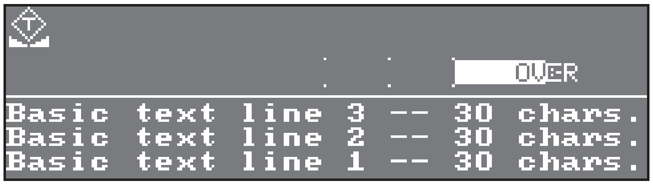
```
Local            OVER
Basic Text line 1 -- 40 fixed characters
Basic Text line 2 -- 40 fixed characters
Basic Text line 3 -- 40 fixed characters
```

Mode 19



```
Basic text line 3 -- 30 chars.
Basic text line 2 -- 30 chars.
Basic text line 1 -- 30 chars.
```

Mode 20



```
                 OVER
Basic text line 3 -- 30 chars.
Basic text line 2 -- 30 chars.
Basic text line 1 -- 30 chars.
```

Appendix 2: WT-BASIC Interpreter Command Set

WT-BASIC

*If you hold in the **CLEAR** key when powering up the PC-820, the WT-BASIC program which is currently resident in the indicator will be temporarily disabled until the next power up.*

This is used when troubleshooting to eliminate the BASIC program as the source of a problem.

The following pages contain the WT-BASIC interpreter command set you use to create programming for the PC-820.

This command set goes beyond the normal BASIC language by adding many commands found only in this expanded WT-BASIC language. This makes programming the PC-820 more flexible than was possible with the original BASIC language. It adheres closely to the QBASIC language included in current versions of MS-DOS. Syntax examples are included where necessary in the following pages.

There is a difference between WT-BASIC and QBASIC. In QBASIC a main program body runs and calls out subroutines and performs some task. In WT-BASIC, you design what are called Event Handlers. These are BASIC subroutines that activate only upon the occurrence of some event. What is an event? It can be any indicator or scale related activity such as

- pressing a soft key
- pressing a hard key
- on a time interval
- reaching a certain gross weight
- scale stability
- scale motion
- serial port data
- setpoint activation or deactivation
- input switch state change
- etc.

The idea is to handle the event then exit the subroutine so that other event handlers can be called. You can create your own names for subroutines that are not triggered by an event but can be called from other subroutines that are triggered (or handled) by an event.

The actual list is very long, but you should get the idea that you can tie a WT-BASIC program to many events. The program you write can cause an unlimited number of things to occur, whether for prompting for data, opening valves, printing tickets or sending data to a computer. It all depends on your imagination and your particular application.

See the included examples for an idea of where to start. For those new to programming or those accustomed to other languages, a book on QBASIC is a good place to begin.

Event Subroutines

Anytime one of the following events is triggered in the instrument, program execution is transferred to the corresponding sub procedure (if the subroutine procedure exists). If more than one event is triggered, they are executed in the order they were tripped. The syntax for a subroutine is identified by the word “SUB” followed by the name of the event subroutine. The subroutine is ended by the command “END SUB”

Some events have default actions that are executed when the event is triggered. The default action is not executed if a subroutine exists in the BASIC program. An example is ZERO_KEY. If the subroutine ZERO_KEY is in the BASIC program the code there is executed instead of the firmware level ZERO function. i.e.:

```
SUB  ZERO_KEY
   x=4
   y=3
   PRINT X,Y
END SUB
```

ACCUM_ABORT

This event is activated if the DOACCUM command is not successfully processed.

ACCUM_OPER

This event is activated if the DOACCUM command is successfully processed.

BASE_OPER

This event is activated if the DOBASE command is successfully processed.

CLEAR_KEY

This event is activated whenever the CLEAR key is pressed.

COM1_MESSAGE

This event is activated when a complete message has arrived in the COM1 serial port input buffer. A “Complete Message” is determined by the End of Message Character in the Serial Ports Configuration.

COM2_MESSAGE

Same as COM1_MESSAGE only for Port #2 primary receive line. (A)

COM2B_MESSAGE

Same as COM1_MESSAGE only for the Port #2 secondary receive line. (B)

ENTER_KEY

An event is activated whenever the ENTER key is pressed.

Attention

When defining this subroutine the key buffer must be emptied using either INKEY$ or an INPUT statement. If the key buffer is not cleared, the indicator may hang.

ENTRY_KEY

This event is activated whenever any key on the remote keyboard is pressed (except function keys).

ESC_KEY

An event is activated whenever the ESCAPE key is pressed.

Fx_KEY

For softkeys. These events queue when the front panel key is pressed or an external keyboard key is pressed. (x; 1-5)

F6 to F10

Only available on an external keyboard or in the simulator. (x; 6-10)

Attention

When defining this subroutine the key buffer must be emptied using either INKEY$ or an INPUT statement. If the key buffer is not cleared, the indicator may hang.

HIDDEN_KEY

This event is activated if the HIDDEN key is pressed. (Key below ESCAPE and above PREVIOUS)

NEXT_KEY

This event is activated if the NEXT key is pressed. (→ on the front panel)

NUMERIC_KEY

This event is activated whenever any numeric key (0-9) is pressed.

PCWTSAMP_ABORT

This event is activated if the PCWTSAMP command is not successfully processed.

PCWTSAMP_OPER

This event is activated if the PCWTSAMP command is successfully processed.

Default: Changes the primary display value to count when not in dual display mode. Changes the secondary display value to count when in the dual display mode.

PCWTZERO_ABORT

This event is activated if the PCWTZERO command is not successfully processed.

PCWTZERO_OPER

This event is activated if the PCWTZERO command is successfully processed.

PREV_KEY

This event is activated if the PREVIOUS key is pressed. (← on the front panel)

PRINT_ABORT

This event is activated if the DOPRINT command is not successfully processed.

PRINT_OPER

This event is activated if the DOPRINT command is successfully processed.

Default: Prints the default print format.

SAMPLE_ABORT

This sample is activated if the DOSAMPLE command is not successfully processed.

SAMPLE_OPER

This event is activated if the DOSAMPLE command is successfully processed.

Default: Changes the primary display value to count when not in dual display mode. Changes the secondary display value to count when in the dual display mode.

SELECT_OPER

This event is activated if the DOSELECT command is successfully processed.

SECOND_OPER

This event is activated if the DOSECOND command is successfully processed.

SETPTx_ACT

These events queue when the SetPoint turns on. There is one subroutine for each setpoint. (x; 1-32)

SETPTx_DEACT

These events queue when the SetPoint turns off. There is one subroutine for each setpoint. (x; 1-32)

SYSTEM_ERROR

Whenever a BASIC error occurs, this event is queued and the error that triggered this event is held in ERR. See also ERR.

SYSTEM_SETUP

There is a 4th setup password that can be entered through the password screen that queues this event.

SYSTEM_STARTUP

If a SYSTEM_STARTUP procedure exists in the WT-BASIC program, it always gets executed at the time of startup or upon exiting CAL or CONFIG menus.

SYSTEM_TIMER

This event is activated on interval determined by Settimer command. This may generate multiple queues in the event buffer.

SYSTEM_ TIMER2

This event is activated on interval determined by Settimer command. This will generate one queue in the event buffer.

TARE_ABORT

This event is activated if the DOTARE command is not successfully processed.

*Defining this blocks the normal default operation from occurring when you press a key on the front panel of the PC-820. Use the **doxxx** commands to make the regular function work. See **doxxx**.*

TARE_KEY

This event is activated if the TARE key is pressed.

Default : Performs a firmware level tare function.

TARE_OPER

This event is activated if the DOTARE command is successfully processed.

Default: Changes the display value to NET if the display value is Gross and Tare does not equal 0 or changes the display value to GROSS if the display value is NET and Tare = 0.

UNITS_OPER

This event is activated if the DOUNITS command is successfully processed.

ZERO_ABORT

This event is activated if the DOZERO command is not successfully processed.

*Defining this blocks the normal default operation from occurring when you press a key on the front panel of the PC-820. Use the **doxxx** commands to make the regular function work. See **doxxx**.*

ZERO_KEY

This event is activated if the ZERO key is pressed.

Default : Performs a firmware level zero.

ZERO_OPER

This event is activated if the DOZERO command is successfully processed.

Variables

Weigh-Tronix strongly recommends the use of the MUSTDIM command in the SYSTEM_STARTUP event subroutine.

The user defined variables can be numeric variables or string variables. The variables can be defined by just referencing them. The type of the variable defined is dependant on the last character of the name of the variable. A variable can be defined also by the command "**DIM *VARIABLE NAME***". For purposes of debugging, the command **MUSTDIM** can be placed in the WT-BASIC program forcing all variables to be defined exclusively through the dimension command.

Numeric Variable

A numeric variable represents a group of numbers only that mathematic functions may be performed on.
A numeric variable is assigned using "**=**" i.e. **X = 42.**
All variables without a special dimensioning character or with a "**#**" character on the end becomes **a 15 digit, double precision, floating point**.
By using a "**%**" character, the variable is dimensioned as a regular **integer**.
An "**!**" character dimensions the variable as a **single precision floating point**.
The "**&**" character dimensions the variable as **Long Integer**.

String Variable

A string variable may represent a group of letters, numbers or a combination of the two that no math functions can be performed on.

A string variable is assigned using "**=**" using quotes to surround the default string. i.e. **X$ = "TARE".**

A standard **string** variable can store a string 16 characters or less and is dimensioned by the character "**$**".

To define a shorter or longer string, the string variable can be succeeded by an "*" and an integer (1-64). The integer defines the length of the string. i.e. "**DIM X$*10**" defines a string X$ with storage space for 10 characters.

System Variable

These variables have already been predefined by Weigh-Tronix in the PC-820 WT-BASIC command set.

Items that bear an asterisk(*) can be set through a WT-BASIC (TARE(3)=100) statement, although they can't be part of an input statement. The rest of the values are set only through firmware level calls. Also, when these are set, they **must** be in calibration unit of measure. When they are recalled the system returns the variable in current unit of measure.

Items marked with the clover (♣) have the ability to include a number in parenthesis after the keyword to select a scale number to access. If the parenthesis is not there the current scale is assumed.

Items that bear an asterisk(*) can be set through a WT-BASIC equivalent statement, although they can't be part of an input statement.

Items marked with the clover (♣) have the ability to include a syntax number in parenthesis after the keyword to select a scale number to access. If the parenthesis is not there the current scale is assumed.

System flags (CZERO,, MOTION, OVERLD, UNDERLD, LOWBATT) will return a true (-1) or a false (0) value depending on their current condition.

ADCCNTS

Returns filtered raw Counts (not scaled to current unit of measure) from the currently selected base. Counts for Quartzell and Analog Base have different ranges.

CAPACITY♣

Returns the capacity of the currently selected scale.

COUNT

Is the actual calculated count value based on the Net Weight.
Count = Net/Pcwt.

COUNTTOT*

Returns or sets count total used by the DoAccum function to total the count.

CURSCALE*

Returns or sets the current scale number selected. (0 through 2)

CURUNIT*♣

Returns or sets the value of the current units.

0=lbs	2=grams	4=lb oz	6=custom2
1=kg	3=oz	5=custom1	

CURUNIT$

Returns the label of the current unit of measure for the current scale.

CZERO

Center of Zero flag. Returns -1 if center of zero, 0 if not on the current scale

DATE$(n)*

Returns or sets the internal date. (n) is the format syntax. See also *Print Format* section of this manual.

DIVISION*♣

Returns or sets the division size of the currently selected scale.

DISPLAY

Returns the active displayed numeric value (i.e., gross, tare, net, count, etc.). When a dual display mode is used (mode 2 or 12), the primary display's value is returned.

ERR

This keyword returns the error number corresponding to a given error which occurred in a WT-BASIC app. The following error numbers and errors are available. Err returns 0 if no error has occurred since power-up. The last error remains until a new error has occurred or until power-down.

SYNTAX:

```
X=ERR
or
PRINT ERR
```

0=	"No error"
1=	“syntax error”
2=	“unbalanced parenthesis”
3=	“no expression present”
4=	“equals sign expected”
5=	“not a variable”
6=	“Label table full”
7=	“duplicate label”
8=	“undefined label”
9=	“THEN expected”
10=	“TO expected”
11=	“too many nested FORs”
12=	“NEXT without FOR”
13=	“too many nested GOSUBs”
14=	“RETURN without GOSUB”
15=	“double quotes needed”
16=	“String Expected”
17=	“Variable Name Too Long”
18=	“Var Not Defined (DIM)”
19=	“too many nested WHILEs”
20=	“WEND without WHILE”
21=	“Division by Zero”
22=	“Ram Sentinel”

GROSS

Returns the gross value of the current scale rounded off by division size.

GROSSTOT*

Returns or sets the gross total used by the DoAccum function to total the gross weights. (Grosstot=Ø to reset)

System flags (CZERO,, MOTION, OVERLD, UNDERLD, LOWBATT) will return a true (-1) or a false (0) value depending on their current condition.

LOWBATT

Returns a True(-1) when the low battery condition exists. Returns a False(0) otherwise.

MAXPEAK*

Returns or sets the highest value on the current scale rounded by division size.

MINPEAK*

Returns or sets the lowest value on the current scale rounded by division size.

MOTION

Motion flag. Returns -1 if motion, 0 if no motion on the current scale.

NET

Returns the net (net=gross - tare) value on the current scale rounded by division size.

NETTOT*

Returns or sets net total used by the DoAccum function to total the net weights.

System flags (CZERO,, MOTION, OVERLD, UNDERLD, LOWBATT) will return a true (-1) or a false (0) value depending on their current condition.

OVERLD

Overload flag. Returns -1 if overload, 0 if not on the current scale. (gross up to 120% of capacity)

PCWT*

Returns the pieceweight value of the current scale.
(pcwt = 0.0017)

RAWGROSS

Returns the unrounded gross weight of the current scale.

RAWNET

Rawnet is calculated from Rawgross-tare.

RAWTARE♣

Rawtare is calculated by converting the tare to A/D counts. Then the tare(in counts) is multiplied by the calibration span factor. The result is not rounded to the nearest division.

ROC

Returns the rate of change of the current scale per configured parameters.

SAMPACCY

This holds the sample accuracy calculated after a sampling process has been completed.

This is most likely to be used in the PCWTSAMPLE_ABORT event routine.

SAMPERR

This holds the error message number after a sample routine has been finished.

0 - No Error; 1 - Accuracy not met;
2 - Insufficient Weight; 3 - Invalid sample size;
4 - Sample Timeout; 5 - Misc. Error;

SAMPSIZE

Allows you to change or print the sample size from WT Basic. Example:
x=SAMPSIZE or SAMPSIZE=x

TARE*♣

Returns or sets the tare value on the current scale. The display will not switch to net mode automatically. Use the ACTVALUE command. Rounded by division size.

TIME$(n)*

Returns or sets the internal time. (n) is the format syntax. See also *Print Format* section of this manual.

TIMER

This command returns the number of seconds (in hundredths) until midnight of the current day.

TRANSTOT*

Returns or sets transaction total used by the DoAccum function to total the number of accumulations.

UNDERLD

Underload flag. Returns -1 if underload, 0 if not on current scale. (gross up to -120% of capacity)

UNIXTIME

This command returns the number of seconds (in hundredths) since midnight January 1, 1970.

Operators

Logical operators return a true (-1) or false (0). Any nonzero value is true.
Example:
14 AND 2 results in a -1 or true.
14 AND 0 results in a 0 or false.

Logical Operators

For an expression using a logical operator to be considered true, the following conditions must be met:

AND
both parts of the expression must be true

OR
either part of the expression must be true

XOR
both parts of the expression must be the opposite

EQV
both parts of the expression must be the same

IMP
first part true, second part false

NOT
inverse of expression or opposite condition

Arithmetic Operators

The arithmetic operators are listed in order of precedence. Multiplication, division, Integer Modulo, and Integer Divide have same precedence. Addition and subtraction have the same precedence. Items with the same precedence are evaluated left to right. Operations within parenthesis are performed first. Inside the parenthesis, the usual order of precedence is maintained.

() Parenthesis

- Negation

^ Exponent or Power of Operator

The result of the Power of operator(^) is the product of multiplying the first operand by itself, the second operand times.

Example:

```
SUB SYSTEM_STARTUP
dispmode=10
Cls
Print (2^8);”=”;(2*2*2*2*2*2*2*2);”=256”
END SUB
```

***** Multiplication

/ Division

m or **MOD** Modulus Operator (Requires a space on each side of operator)

The result of the modulus operator (m) is the remainder when the first operand is divided by the second.

The “m” must always be lower case and must always have a space before and after it.
Example:

```
SUB SYSTEM_STARTUP
dispmode=10
cls
Print (12345678 m 7);”=2”   ‘mod 7
Print (12345678 mod 7);”=2”   ‘mod 7
END SUB
```

\ Integer Divide (Requires a space on each side of operator)

This operator divides 2 numbers and returns the Integer portion of the computation.
Example:

```
SUB SYSTEM_STARTUP
dispmode=10
Cls
Print gross \ 3
END SUB
```

+ Addition

- Subtraction

Relational Operators

=	Equal to
>	Greater than
<	Less than
<=	Less than or equal to
>=	Greater than or equal to
< >	Not equal to
> <	Not equal to

Input/Output

PRINT

To output data to the screen. The area of the display dedicated to WT-BASIC messages is determined by the active display mode.

Maximum of 40 characters/ display line for small font. Maximum of 30 characters/ display line for large font. Both fonts are fixed width.

SYNTAX: PRINT "THIS WILL GO TO THE DISPLAY"

PRINT #*x*

To output data to a serial port.

SYNTAX: PRINT #1, "THIS WILL GO OUT OF PORT #1"

PRINT #2, "THIS WILL GO OUT OF PORT #2"

Any keyword with a printable message in quotation marks must have a space separating the keyword from the message.

Example: Print "Hello"

Space

FMTPRINT(n)

Calls a predefined print format (n = 1 to 32) that will be sent out a serial port, as specified in destination port settings.

INPUT

To receive data input from the keypad or a remote keyboard. The specified prompt is shown on the display. The input softkey interface is selected by the variable type to be input. Integer input allows decimal points to be entered, but the fractional portion is clipped before storage.

SYNTAX: INPUT "Enter id", ID$

INPUTOPT

This stands for input options. This keyword changes how the input statements interact with the user. The functions of the input command are set with this keyword and affect all input statements until another set of options are selected. The default on power-up is (0,0,0,0,0,0).

These input options remain active for all input statements until turned off by using INPUTOPT(0,0,0,0,0,0).

RPN: Reverse Polish Notation

SYNTAX: INPUTOPT(noecho, timeout, noclear, nominus, RPN, val-type)

NOECHO - If noecho is set to "0" then the data entered during an input statement is visible. If noecho is set to "1" or "-1" then the data entered during an input statement appears as an asterisk "*". This is useful for password protecting softkeys.

TIMEOUT - If timeout is set to "0", the input statement waits for an ENTER or ESC key depression. If timeout is set to a number, then the input statement timeouts after that amount of time if a key is not hit. The timeout feature is reset each time a key is hit.

NOCLEAR - If noclear is set to "0" the variable used for the input statement is cleared upon data entry. If noclear is set to "1" or "-1" the variable used for the input statement is NOT cleared upon data entry, and the character/number entered is concatenated to the existing data.

NOMINUS - If nominus is set to "0" the softkeys for numeric entry include a minus sign. If nominus is set to "1" the softkey labeled "-NEG" is not present. (This softkey configuration only affects RPN = 0)

LABELED - This softkey parameter only affects numeric entries if the RPN parameter is set to RPN=0

RPN - Reverse Polish Notation. If RPN is set to “0” the softkeys are placed on the display that assist the numeric or string entry. These softkeys are “BKSP←” “CHR↑” “CHR↓” and “ADV→” for alphanumeric and “BKSP←” and “-NEG” for numeric entry only. In the RPN = 0 mode, the ENTER key ends the entry session and the given variable is stuffed with the entered value or string.

If RPN is set to “1” the softkeys are not changed upon entering the input session. For example, you may have the softkeys labeled, "ID", "TYPE", "PN", "LOC", BIN". Any of the 5 softkeys end the entry session, which activates the softkey subroutine (or corresponding menu subroutine if defined) after passing the new value to the given variable.

VAL-TYPE - If this parameter is set to "0", the standard display entry mode is used, including your input prompt message and softkeys labeled for alphanumeric entries.

If this parameter is set to "1" then the previoiusly selected display mode is currently active which could contain continually updated displayed weight values. The text lines will contain the following messages:

1st Available line **- Key in Value. . .Press Enter**

Last available line - User Define Prompt From Input Statement and Data to be Input

Any other available text lines could be used via print statements. Display modes 1-10, 15 and 16 are recommended for this parameter.

INKEY$ is typically used in a program loop for detecting a specific key depression.

Also used to clear the value of "LASTKEY" or "KEYHIT"

INKEY$

Returns one character read from the remote keyboard or from the PC-820 front panel keypad as a string. Returns an empty string “ “, if no character is available.

SYNTAX: X$ = INKEY$

KEYHIT

This command is used in program loops to detect a key hit on the front panel of the PC-820. This command only recognizes numeric keys on the front panel of the PC-820.

SYNTAX: x = KEYHIT

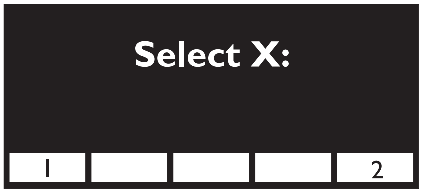
The variable **x** is returned as a true, negative one (-1), or a false, zero (0) depending on whether a key is hit or not.

LASTKEY

LASTKEY returns the ASCII value of the last key pressed on the keyboard as a numeric value.

SYNTAX: x = LASTKEY

Last Key	*Values Returned*	*Last Key*	*Values Returned*
F1	*15104*	*.*	*46*
F2	*15360*	*0*	*48*
F3	*15616*	*1*	*49*
F4	*15872*	*2*	*50*
F5	*16128*	*3*	*51*
F6	*16384*	*4*	*52*
F7	*16640*	*5*	*53*
F8	*16896*	*6*	*54*
F9	*17152*	*7*	*55*
F10	*17408*	*8*	*56*
TARE	*5120*	*9*	*57*
ZERO	*11264*	*Hidden*	*8960*
ESCAPE	*27*	*Previous*	*6400*
ENTER	*13*	*Next*	*12544*
CLEAR	*11776*		



Select X:

ASK(MSG$)

ASK is a function that allows you to prompt the operator for a Yes/No type response. The ASK command accepts 2 optional arguments which allow the user to change the legends of the F1 key and the F5 key. The ASK(MSG$) function returns a negative one (-1) or true for a YES response and a zero (0) or false for a NO response.

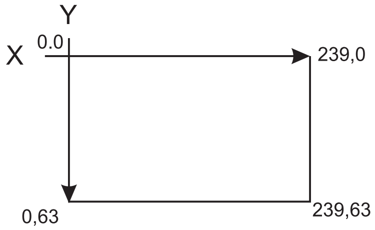
SYNTAX: ASK("MSG$", "TRUE", "FALSE")

An example would be as follows:

```
SUB SYSTEM_STARTUP
dispmode=10
if ask("Select X:","1","2") then x=2 else x=1
cls
print "X=",x
END SUB
```

Graphics

The PC-820 has a display height of 64 dots and a width of 240 dots. Coordinate 0,0 is the upper leftmost dot.



Graphics can only be used when a display mode is chosen which does not contain a weight display. The upper left-hand corner of the display is (0,0). The lower right-hand corner is (239,63). The "X" references the horizontal coordinates (0-239). "Y" references the vertical coordinates (0-63).

DOT(X,Y)

Draws a dot on the screen at coordinates X,Y.

CIRCLE(X,Y,R)

Draws a circle on the screen starting at coordinates X,Y, using the value of R as the radius (in pixels).

LINE(X1,Y1,X2,Y2)

Draws a line on the screen from coordinates X1, Y1 to X2, Y2.

Structure Control

Use REM or ‘ for describing operations, setup of the instrument, temporarily disabling a command, etc.

*All forms of **IF.. commands** are used to to make a decision regarding program flow based on the result of an expression.*

Example of an expression:
(A>B)

Examples of statements:
PRINT “BIG NUMBER”
or
PRINT “LITTLE NUMBER”
or
*C = (A*B)+ d*

WARNING
GOTOs and GOSUBs are not recommended

“Statement label” is any name or number used to identify the next “line of code” or statement to be executed.

REM

Allows for comments to be placed in the body of a program.

‘ (apostrophe)

This also works in place of REM.

IF. .THEN (singular)

SYNTAX: IF *expression* THEN *statement*

IF. . THEN . .ELSE (singular)

SYNTAX: IF *expression* THEN *statement* ELSE *statement*

IF. .THEN (block)

SYNTAX:

```
IF expression THEN
        statement
        statement
        statement
END IF
```

IF. .THEN.. ELSE (block)

SYNTAX:

```
IF expression THEN
        statement
        statement
        statement
ELSE
        statement
        statement
        statement
END IF
```

END IF

Used to stop a block of statements.

ELSEIF

This command allows absolute detection of a condition.

SYNTAX:

```
IF expression THEN
        statement
        ELSEIF expression THEN
                statement
        ELSEIF expression THEN
                statement
ELSE
        statement
END IF
```

GOTO

To branch out of the normal program sequence of instructions to a specified instruction identified by a “statement label”.

SYNTAX: GOTO *statement label*

FOR. . NEXT loops only count up, not down.

FOR. .TO. .NEXT

To execute a series of instructions a specified number of times in a loop. Must also use "NEXT" command.

EXIT FOR

Used to terminate a FOR. . NEXT loop upon detection of a certain condition prior to the loop expiring naturally.

SYNTAX:

```
FOR variable = X TO Y
        statement
IF expression THEN EXIT FOR
NEXT variable
```

WHILE. . WEND

To execute a series of statements in a loop as long as a given condition is true.

EXIT WHILE

Used to terminate a WHILE. . WEND loop upon detection of a certain condition prior to the loop expiring naturally.

SYNTAX:

```
WHILE expression
        statement
IF expression THEN EXIT WHILE
        statement
WEND
```

*It is usually better to use **CALL** instead of **GOSUB***

GOSUB . . . RETURN

To branch to a subroutine and exit from a subroutine. These work the same as CALL and EXIT SUB.

RETURN

To return from a subroutine.

SUB . . . END SUB

Defines a subroutine procedure.

EXIT SUB

Used to terminate a subroutine upon detection of a certain condition prior to the subroutine expiring naturally.

SYNTAX:

```
        SUB subname
                statement(s)
        IF expression THEN EXIT SUB
                statement(s)
        END SUB
```

CALL

Transfers control to a second subroutine procedure. After the second subroutine is done, control automatically transfers back to the next line after the CALL command in the previous subroutine.

SYNTAX: CALL PrintRoutine

Display Control

CLS

Clears the display of any WT-BASIC text. Not necessary before a DISPMODE command.

REFRESH

Updates the displayed data while in a WT-BASIC program loop.

DISPMODE(n) or DISPMODE=n

Sets the active display mode. The display modes may be viewed in accompanying software or in Appendix 1 of this manual. There are 20 displays to choose from. (n) is the number representing the display mode.

```
SYNTAX:    DISPMODE(n)
```

Pay close attention to the location of spaces used with keywords that have a message inside of quotation marks. A space usually precedes the first quotation mark.

KEY x,"keyname"

Names a function key on the screen. (x; 1-5)

```
SYNTAX:    KEY 1, "START"
           KEY 2, "STOP"
           KEY 3, "SET UP"
           KEY 4, "MORE"
```

#0	Gross
#1	Net
#2	Tare
#3	Minimum
#4	Maximum
#5	Rate of Change
#6	Gross Total
#7	Net Total
#8	Count Total
#9	Transaction Total
#10	Count
#11	Value
#12	Piece Weight
#13	A to D Counts

ACTVALUE

As a command ACTVALUE sets the displayed numeric data on the indicator to a different active value. See list at left.

```
SYNTAX:    ACTVALUE(n)     n=0 to 13
```

As a system variable ACTVALUE returns number (n) which represents the currently displayed active value per list at left.

```
SYNTAX:    n=ACTVALUE     n=0 to 13
```

SECVALUE (only used in dispmode 2 & 12)

As a command SECVALUE sets the secondary displayed numeric value on the indicator to a different active value. See list at left.

```
SYNTAX:    SECVALUE(n)     n=0 to 13
```

As a system variable SECVALUE returns number (n) which represents the currently displayed secondary display value. See list at left.

```
SYNTAX:    n=SECVALUE     n=0 to 13
```

If the auto backlight timer is enabled, it controls the backlight causing this keyword not to function.

BACKLITE

Turns the backlight of the display on and off.

```
SYNTAX:    BACKLITE(n)
           n=BACKLITE        n=0 to 1
           0=OFF 1=ON
```

CONTRAST

Sets the contrast for the display.

```
SYNTAX     CONTRAST(n)
           n=CONTRAST        n=0 to 63
```

GRBASIS(*n,[varname$]*)

This command sets the basis for the bar graph representation on the display. (n = 0 to 13) Varname$ only applies when the graph basis is number 11, variable.

0 = Gross	4 = Max	8 = Count Total	12=Piece Weight
1 = Net	5 = Rate of Change	9 = # Total Trans.	13=ADC
2 = Tare	6 = Gross Total	10=Count	
3 = Min	7 = Net Total	11=Variable	

SETBAR(*MinValue,MaxValue*)

Sets the values for the bar graph. The (MIN) is the minimum value where the graph starts. The (MAX) is the maximum value where the graph stops.

SETCHECK(*Min,Under,Over,Max*)

Sets the values for the checkweigher graph. The (MIN) is the minimum value where the graph starts. The (MAX) is the maximum value where the graph stops. (UNDER) is the lowest acceptable value for the checkweigher graphic display. (OVER) is the highest acceptable value for the checkweigher graphic display.

In the SHOWVAR syntax the precision value is the number of digits on the right side of the decimal point for your displayed numeric value. Remember to select a display mode that will allow you to see all the digits.

SHOWVAR

This lets you display a numeric value on the display where the weight value usually appears along with custom labels or legends. See note at left.

SYNTAX: SHOWVAR (varname$,legend1$,legend2$,precision)

SHOWSETP

Shows setpoints in the upper right of the display. Use only for troubleshooting. **This command should not be left in a customer's final program version.**

MENU

Menu is used to reduce the amount of code needed in order to structure menus. The keyword has the following syntax:

MENU prev$, next$, F1$, F1sub$, F2$, F2sub$, F3$, F3sub$, F4$, F4sub$, F5$, F5sub$

prev$ and next$ are strings of subroutine names that are activated when the previous and next keys are pressed instead of the default subroutines “PREV_KEY” and “NEXT_KEY”.

F1$, F2$, … F5$ are strings that are used to place labels above the softkeys.

F1sub$, F2sub$, … F5sub$ are strings of subroutine names that are activated when the F1 through F5 keys are pressed.

MENU reads up to 12 strings separated by commas. To force the function keys, next and previous to activated their default subroutines, call the MENU function with no parameters. If any of the strings are left out (i.e. only 10 strings are given) those keys revert back to activating their default subroutines. In order to disable a key, an empty subroutine should be assigned. To name a key, but leave that key's subroutine at default, use empty strings (“”) as the subroutine name.

EXAMPLE:

```
NEXT$ = “NEXT_MENU”
PREV$ = “PREV_MENU”

SUB MT_SUB
END SUB
```

```
SUB UNITS_SUB
   dounits
END SUB

SUB F2_KEY
   dosample
END SUB

SUB OFF_SUB
   shutdown
END SUB

MENU NEXT$, PREV$, “Units”, “UNITS_SUB”, “Sample”, “”, “”, “MT_SUB”, “”,
“MT_SUB”, CHR$(173) + “OFF!”, “OFF_SUB”
```

MENUOPT

MENUOPT provides a convenient way of disabling the previous, next, or softkeys.

SYNTAX: MENUOPT p, n, f1, f2, f3, f4, f5

p - Set to 1 disables the previous key
n - Set to 1 disabled the next key
f1 through f5 - Set to 1 disables function keys f1 throught f5

To disable a key enter a 1 in the appropriate spot. The key is then disabled until the scale cycles power or until the MENUOPT keyword is called again with a 0 to replace the 1 value. When a softkey is disabled, the associated subroutine name is not activated when a softkey is pressed. Also the softkey label is not shown if the key is disabled.

RPN is Reverse Polish Notation

RPNMENU

RPNMENU allows the user to configure the softkeys for an RPN input statement. The menu is automatically loaded when the input statement is used and the RPN flag is set in inputopt. If more than 1 menu command is issued, the extra softkey label will be turned on.

SYNTAX: RPNMENU index$,label1$,sub1$, label2$,sub2$, label3$,sub3$, label4$,sub4$, label5$,sub5$,

Where index$ represents the menu level for scrolling. If index$ equals “1”, the memory locations for index “2” and “3” will be cleared. There may be a total of three rpnmenu statements active at a time. A menu statement should not be used in directly following an input statement. This would changed the address of the event handler which was queued which would result in the wrong event being executed. The menu statement should be in the event which was queued from the input statement. The following example shows this keywords usage:

```
SUB SYSTEM_STARTUP
rpnmenu “1”,”TEMP”,”TEMP”,””,””,””,””,””,””,”TEMP
1",”TEMP1"
rpnmenu “2”,”TEMP 2",”TEMP2",””,””,””,””,””,””,”TEMP
3",”TEMP3"
rpnmenu “3”,”TEMP 4",”TEMP4",””,””,””,””,””,””,”TEMP
5",”TEMP5"
call Level0
inputopt(0,0,0,0,1,1)
END SUB
```

```
SUB OFF
shutdown
END SUB

SUB SAMPLE
 doSample
END SUB

SUB LEVEL0
menu
“LEVEL10”,”LEVEL10",”SAMPLE”,”SAMPLE”,””,””,””,””,””,””,””,””
END SUB

SUB LEVEL10
menu
“LEVEL0”,”LEVEL0",”OFF”,”OFF”,””,””,””,””,””,””,””,””
END SUB

SUB TEMP
cls
print temp$
print “TEMP HERE”
call Level0
END SUB

SUB TEMP1
cls
print temp$
print “TEMP1 HERE”
call Level0
END SUB

SUB TEMP2
cls
print temp$
print “TEMP2 HERE”
call Level0
END SUB

SUB TEMP3
cls
print temp$
print “TEMP3 HERE”
call Level0
END SUB

SUB TEMP4
cls
print temp$
print “TEMP4 HERE”
call Level0
END SUB

SUB TEMP5
cls
print temp$
print “TEMP5 HERE”
call Level0
END SUB
```

```
SUB NUMERIC_KEY
input temp$
END SUB

SUB ENTRY_KEY
input temp$
END SUB
```

Serial Port Hardware Control

GETPORT

Returns the numeric value (x) that represents the current protocol of the port selected from the tables below.

SYNTAX: x=getport(port,y)

Pick ‘y’ from the table below (1 to 13)
1 baud rate
2 parity
3 databits
4 handshake
5 mode
6 EOM character
8 CTS (only used in getport)
9 transmit buffer free size (max 512)
10 receive buffer count (512 is full)
11 number of messages in the receive buffer
12 Serial Blocking
13 Download Blocking

EOM - End of Message character as set in the configuration menu.

Baud	Parity	Databits	Handshake	Mode
1=300	0=none	7=7	0=none	0=Basic Control
2=1200	1=odd	8=8	1=CTS	1=Keyboard
3=2400	2=even		2=XonXoff	2=Disable
4=4800	3=set			5=Split (BASIC/Keyboard)
5=9600	4=clear			6=Dual (Basic Control)
6=19200				8=PWLU
7=38400				
8=56700				

CTS
0=off
1=on

SETPORT

SETPORT(port,y,z)   Pick y from the table below.
Pick z from the corresponding list below.

Input buffer size is 512 characters. You may define the length of your string variable in a DIM statement. If the length is not defined, it defaults to 16 characters. Storage of strings in memory is limited to 16 characters per location. ***See Appendix 3 for a sample routine called GETCOM$.***

1 baud rate
2 parity
3 databits
4 handshake
5 mode
6 EOM character
7 RTS (only used in setport)
12 Serial Blocking
13 Download Blocking

RTS	Serial Blocking
0=off	0=on
1=on	1=off

GETCOM$(portnum,<maxchars>)

GETCOM$() reads one message from the specified Com-Port. The function reads from the first character in the input buffer to one of the following:

a) End of Message Character (As specified in the Serial Port configuration Menu in the SimPoser)

b) No more characters in the input buffer.

c) Maximum characters reached (As specified in the optional GETCOM$ parameter)

d) Maximum string size reached

SYNTAX: X$=GETCOM$(portnum,<maxchars>)

portnum represents serial port 1, 2, or 3 (three is the port 2 secondary receive line)

maxchars (optional) represents the maximum number of characters to read and assign to the string (X$).

WARNING: Using CONTOUT may cause system performance problems. i.e.-execution speed may be slowed.

CONTOUT(fmt,rate)

This command activates the continuous output of the chosen format. Destination port is selected from the format menu.

SYNTAX: CONTOUT(FMT,RATE)
FMT = format print number
RATE = update rate

Example: CONTOUT(1,2) This outputs format 1 twice per second.

SETPTON(n)

This command will activate SETPOINT n.

SETPTOFF(n)

This command will deactivate SETPOINT n.

ISSETPT(x)

Tells you whether the setpoint is activated.

SYNTAX: Z = ISSETPT(x) x = 1 to 32

The variable Z is returned as a true, negative one (-1), or a false, zero (0) depending on the condition of the setpoint.

TONE(x)

Turns the tone on. x = 1/Frequency x Offset

TONEOFF

Turns the tone off.

BEEP

Sounds the beeper in the instrument.

GETITEM

Get item is used to recall configuration parameters to be used by BASIC. The GETITEM request has 3 parameters. The first parameter is a two letter code that is used by the downloader to configure the indicator. The second parameter is used to request a particular data item from that list. The third parameter is used to choose a particular scale used by only certain configuration parameters if they fetch multiscale data. The following shows the context for using GETITEM and the two letter configuration codes that are supported.

GETITEM(“XX”, n)

“XX” - Is the configuration parameter to get the value from (See list below for an explanation of the configuration parameters available).

Supports “PF”, “SM”, “LC”, “LA”, “PS”, “UN”, “KY”, “VT”, “MS”, “GL”, “DS”

n - Is the number of the configuration parameter to recall. The first configuration parameter is 1 and the second is 2 and so on. Any item that is numerical in nature is recalled, string parameters and undefined parameters return -2.

PF
Print format port. PF, n returns the port number used for print format n.

SM
Sampling configurations.

1) Minimum sample size (1-9999)
2) Minimum desired accuracy (0 - Off; 1 - 95%; 2 - 98%; 3 - 99%)
3) Minimum sample weight(0 - 0.01%; 1 - 0.005%, 2 - 0.002%; 3 - 0.001%)
4) Sample Mode (0 - Bulk; 1 - Dribble;)
5) Piece Weight Rounding (0 - Off; 1 - 0.5%; 2 - 1%; 3 - 2%)

6) Auto Base Switching (-1 enable, 0 disable)
7) Default Sample Base (0- Local; 1 - Remote1; 2 - Remote2)
8) Default Counting Base (-1 - Last; 0 - Local; 1 - Remote1; 2 - Remote2)

9) Sample Mode Key Options - Sample Mode (-1 enable, 0 disable)
10) Sample Mode Key Options - Sample Base (-1 enable, 0 disable)
11) Sample Mode Key Options - Minimum Sample Weight (-1 enable, 0 disable)
12) Sample Mode Key Options - Sample Accuracy (-1 enable, 0 disable)

LC
Latched count configurations.

1) Count Latching (0 - Disabled; 1 - Fast; 2 - Slow)
2) Sample Latching (0 - Disabled; 1 - 1 piece disables; 2 - 2 pieces disables)

LA
Configure the language to use in the USER menus and in the prompting for the scale. **NOT AVAILABLE AT THIS TIME.**

PS

Configures the power saving modes of the scale.

1) Power saving shutoff (-1 enable, 0 disable)
2) Power saving inactivity timer (integer entry 0 to 60 minutes)
3) Shutdown warning (-1 enable, 0 disable)
4) Backlight display/power saving auto shutoff (-1 enable, 0 disable)
5) Backlight display/power saving inactivity timer (integer entry 0 to 60 minutes)

UN

Configures the number of active scales and the unit of measures.

1) Number of active Scales (1-3 scales (Local only is 1))
2) Calibration units - Selectable 0-lbs to 6-Custom unit 2
3) Pounds enable (-1 enable, 0 disable)
4) Kilograms enable (-1 enable, 0 disable)
5) Grams Enable (-1 enable, 0 disable)
6) Ounces enable (-1 enable, 0 disable)
7) Pound-Ounces enable (-1 enable, 0 disable)
8) Custom unit 1 enable (-1 enable, 0 disable)
9) Custom unit 2 enable (-1 enable, 0 disable)

KY

Key enable configuration.

1) Tare enable (-1 enable, 0 disable)
2) Zero enable (-1 enable, 0 disable)
3) KP Enable (Keys 0-9 on front panel (-1 enable, 0 disable)
4) Softkey print enable (-1 enable, 0 disable)
5) Softkey sample enable (-1 enable, 0 disable)
6) Softkey recall enable (-1 enable, 0 disable)
7) Softkey dBase enable (-1 enable, 0 disable)
8) Softkey accumulate enable (-1 enable, 0 disable)

VT

Display value enable.

1) Gross enable (-1 enable, 0 disable)
2) Net enable (-1 enable, 0 disable)
3) Tare enable (-1 enable, 0 disable)
4) Min enable (-1 enable, 0 disable)
5) Max enable (-1 enable, 0 disable)
6) Rate of change enable (-1 enable, 0 disable)
7) Gross total enable (-1 enable, 0 disable)
8) Net total enable (-1 enable, 0 disable)
9) Count total enable (-1 enable, 0 disable)
10) Transaction count enable (-1 enable, 0 disable)
11) Count enable (-1 enable, 0 disable)
12) Variable enable (-1 enable, 0 disable)
13) Pc. Wt. Enable (-1 enable, 0 disable)
14) ADC Enable (-1 enable, 0 disable)

MS
Miscellaneous configurations.

1) Place holder
2) Default print format (0-32)
3) Date format (0 MMDDYY, 1 DDMMYY, 2 YYMMDD)
4) Beeper volume (0 disabled, 1 low, 2 medium, 3 high)
5) Place holder
6) Place holder
7) Place holder
8) Place holder
9) Small font - enable lower case (0 disable, -1 enable)
10) Enable switching display mode with hidden key (0 disable, -1 enable)
11) Enable low battery indication (0 disable, -1 enable)

GL
Bar graph configuration.

1) Graph minimum (float - weight in cal. unit of measure)
2) Graph below (float - weight in cal. unit of measure)
3) Graph above (float - weight in cal. unit of measure)
4) Graph maximum (float - weight in cal. unit of measure)
5) Graph basis (0 Gross through 13 ADC value)

DS
Selected display mode.

1) Power-up display mode number (1-20 integer)
2) Power-up Display value (0 Gross through 13 ADC value)
3) Power-up Secondary Display Value (0 = Gross through 13 - ADC Value)
(See note at left for display values 0-13)

#0	Gross
#1	Net
#2	Tare
#3	Minimum
#4	Maximum
#5	Rate of Change
#6	Gross Total
#7	Net Total
#8	Count Total
#9	Transaction Total
#10	Count
#11	Value
#12	Piece Weight
#13	A to D Counts

GETITEM(“XX”, n, sc)

“XX” - Is the configuration parameter to get the value from (See list below for an explanation of the configuration parameters available).

Supports “TS”, “MC”, “ZS”, “FS”, “RS”, “SC”, “RN”, “TA”

n - Is the number of the configuration parameter to recall. The first parameter for these configurations are scale number. The parameter immediately following the scale number is 1 and the next is 2 and so on. Any item that is numerical in nature is recalled. String parameters and undefined parameters return -2.

sc - Is provided to choose a scale from which to recall the parameter from. If a scale number is not provided the Local scale (0) is assumed.

EXCEPTIONS:

GETITEM(“PF”,f) - Returns the port number to which the format number ‘f’ is configured to output to.

GETITEM(“UN”,10) Returns a true (-1) if more than one unit of measure is enabled, otherwise returns false (0).

TS

Key time out configuration for **a specific** scale.

1) Accumulator time out (Float in seconds)
2) Print time out (Float in seconds)
3) Zero time out (Float in seconds)
4) Tare time out (Float in seconds)
5) Sample time out (Float in seconds)

MC

Motion detection configuration for **a specific** scale.

1) Motion detection enable (-1 enable, 0 disable)
(If motion detection is disabled Range and Delay are set to zero. Remainder of line is read, but not taken)
2) Range (float in divisions 0.1-20.0)
3) Delay (float in seconds 0.05-10.0)

ZS

Zero tracking configuration for a **specific** scale.

1) Zero tracking enable (-1 enable, 0 disable)
(If zero tracking is disabled Range and Delay are set to zero. Remainder of line is read, but not taken)
2) Range (float in divisions 0.1-20.0)
3) Delay (float in seconds 0.05-10.0)

FS

Filtering configuration for a **specific** scale.

1) Unused
2) Samples to average (1 to 60)
3) Harmonizer enable (-1 enable, 0 disable)
(Disabling the Harmonizer sets the constant value to 0)
4) Harmonizer constant (0 through 10)
5) Harmonize threshold (float weight in cal. unit of measure)

RS

Rate of change configuration for a **specific** scale.

1) Rate of change samples (0 through 60)
2) Rate of change multiplier (float)

SC

Configures a single scale.

1) Scale Type (0 Analog, 1 Quartzell)
2) Excitation configuration (if analog) (0 - DC, 1 - 300 Hz, 2 - 600 Hz, 3 - 1200 Hz)
3) Capacity (Float- weight in cal. unit of measure)
4) Division (Float - weight in cal. unit of measure)
5) Update rate (0-0.5 Hz, 1-1 Hz, 2-2 Hz, 3-5 Hz,)
6) Zero range (0-100%)
7) Not used with this scale
8) Accumulator return to zero (0 to 100%)
9) Print return to zero (0 to 100%)
10) Cal Wt (float - weight in cal. unit of measure)
11) Initial Zero setting (0-disable; -1-enable)
12) Scale Enabled/Disabled (0 disable; -1 enable)

RN

Configures the over and under ranges of a **specific** scale.

1) Over Range basis (0 - based on deadload; 1 based on current zero)
2) Under Range basis (0 - based on deadload; 1 based on current zero)
3) Over Range limit (0 to 200% if basis = 0; div. over capacity if basis = 1)
4) Under Range limit (0 to 200% if basis = 0; div. over capacity if basis = 1)

TA

Configures how tare is used in calculations for a **specific** scale.

1) Push-button tare rounding? (0 - Disabled; 1 - Rounded to division size)
2) Keyboard/Recall Tare Rounding? (0 - Disabled; 1 - Rounded to divisions size)

UNIT$

The command UNIT$(0) returns ‘lb’ and UNIT$(1) returns ‘kg’, etc. (For LB-OZ unit of measure, OZ is returned due to all weight variables return ounces when the LB-OZ unit of measure is selected.)

CALDATA(x)

Returns calibration information for ISO purposes.

x=1 Span Factor
x=2 Zero Count
x=3 Last calibration time
x=4 Last calibration date

VERSION$(x)

Returns the version of the firmware and W-T Basic program.

x=1; Indicator type, SimPoser version, time, date of download, License #, customer name
x=2 ; Firmware date and time (ex, “Oct 27 1998 14:43:00”
x=3; Firmware Part Number (ex. “51515-0019”)
x=4; Firmware Revision Letter (ex “A”)

SETPWD(n,p$)

Sets the system password only.

The index n selects the password to set (1 - System password)

The string p$ is the string containing the new password.

Syntax for setting the system password:

```
 P$ = “PASSWORD”
SETPWD(1,P$)
```

Timer/Event/Sleep Control

There are two timer events available for use in the PC-820. They are SYSTEM_TIMER and SYSTEM_TIMER2. Use SETTIMER to regulate their frequency of occurrence.

SHUTDOWN

This command causes the scale to disable power and shutdown. There are no parameters for this function.

SETTIMER

This command causes the SYSTEM_TIMER event to occur.

SYNTAX: SETTIMER(timernum,seconds)

Example:

SETTIMER(1,0.5) This will activate the event system_timer every 0.5 second. With zero seconds, this disables the timer event specified. This is the default state of the Timer Events upon power up.

SETTIMER(1,0) This shuts the timer off.

EVENTNUM

This command returns the variable (x) which identifies the event name. The variable (x) can then be used in the EVENTRDY and EVENTCLR commands.

```
SYNTAX:   x = EVENTNUM(NAME$)
                The variable (x) is the EVENTNUM.
                Name$ is the event name.
```

EVENTRDY(x)

This command detects the existence of an event name in the event queue.

SYNTAX: Z = EVENTRDY(x)

The variable Z returns true, negative one (-1), or false, zero (0).

The variable x represents the value of the event name from the command EVENTNUM.

EVENTCLR(x)

This clears the oldest instance of EVENTNUM from the event queue.

SYNTAX: EVENTCLR(x)

The variable x represents the value of the event name from the command EVENTNUM.

SLEEP(n)

Causes the program to halt processing for n number of seconds.

SYNTAX: SLEEP(10) (Causes processing to halt for 10 seconds)

CALCSTAT

Calcstat is a function that calculates a number of statistical values based on a series of numbers.

SYNTAX: CALCSTAT(START,COUNT,DEST)

Parameters

Start- Start is the beginning memory location of a series of values to calculate the statistical analysis on.

Count- Count is the number of sequential memory locations from the previous stated START parameter that have values stored to calculate the statistical analysis on.

Destination- Destination is the starting memory location to record the results of the statistical analysis. There must be eight sequential memory locations available for the results of CALCSTAT to be recorded in. The results will be in the following order starting at the Destination memory location:

Average
Minimum
Maximum
Average Deviation
Standard Deviation
Standard Variance
Skewness
Curtosis

ABS(n)

Returns the absolute value of the expression.

ATN(n)

Returns the arctangent of the expression.

SQR(n)

Returns the square root of the expression.

INT(n)

Returns the integer value of the expression.

FIX(n)

Truncates the expression to a whole number.

CINT(n)

Rounds numbers with fractional portions to the next whole number or integer.

SGN(n)

Returns the sign of the expression. 1, 0, or -1

SIN(n)

Calculates the trigonometric sine of the expression, in radians.

COS(n)

Calculates the cosine of the range of the expression.

TAN(n)

Calculates the trigonometric tangent of the expression, in radians.

LOG(n)
Calculates the natural logarithm of the expression.

EXP(n)
Returns e to the power of x.

ROUND(x,y)
Rounds X to the nearest y.

```
SYNTAX:    Z=Round(X,Y)
           With X=10.04 and Y=0.05
           The value of Z is now set to 10.05
```

RANDOM
Generates a random decimal number between 0 and (n).

```
SYNTAX:    x = RANDOM(n)
```

The variable x represents the generated decimal number.

The variable n represents the largest decimal number you want to use.

String Functions

STR$(n)
Converts the value of an expression to a string.

```
SYNTAX:    Z$=STR$(X)
           With X= 5000
```

The string Z$ is now set to “5000” not the numeric value of 5000

MID$(C$,n,[x])
This command copies part of an existing string (C$) and makes a new string. The new string starts at the nth character of the original string (C$) and continues for x number of characters.

```
SYNTAX:    Z$=MID$(C$,n,[x]) ( [ ]=optional )
           With C$ =”The WI-130 Indicator”
           and n = 5 and x = 6
           MID$ returns the new string Z$ or “WI-130”.
```

LEFT$(C$,x)
This command copies the leftmost characters of an existing string (C$) and makes a new string. The new string starts at the leftmost character of the original string and continues for x number of characters to the right.

```
SYNTAX:    Z$=LEFT$(C$,x)
           With C$ =”The WI-130 Indicator”
           and x = 3
           LEFT$ returns the new string Z$ or “The”.
```

RIGHT$(C$,n)
This command copies the rightmost characters of an existing string (C$) and makes a new string. The new string starts at the rightmost character of the original string and continues for x number of characters to the left.

```
SYNTAX:    Z$=RIGHT$(C$,x)
           With C$ =”The WI-130 Indicator”
           and x = 9
           RIGHT$ returns the new string Z$ or “Indicator”.
```

CHR$(n)

Converts an ASCII code to its equivalent character. See Appendix 5.

SYNTAX: Z$ = CHR$(n)

Converts the numeric value of X to a string as defined by width and precision parameters.

FORMAT$

Format$(x,width.prec) Returns x as string with a width and precision the same as in the print formats. See Format section in this manual for more information on width and precision.

SYNTAX: Z$ = FORMAT$(x,w.p)

LCASE$(C$)

Converts the C$ to lower case characters.

SYNTAX: Z$ = LCASE$(C$)

UCASE$(C$)

Converts the C$ to upper case characters.

SYNTAX: Z$ =UCASE$(C$)

HEX$(n)

Creates a new string that represents the hexadecimal value of the numeric argument.

SYNTAX: Z$ = HEX$(n)

LTRIM$(C$)

Strips C$ of any leading spaces.

SYNTAX: Z$ = LTRIM$(C$)

RTRIM$(C$)

Strips C$ of any trailing spaces.

SYNTAX: Z$ = RTRIM$(C$)

Julian date is the number of days since some day in the ancient past. One reference point is Julian day 2,440,000 is May 23, 1968. This allows any date to be represented or stored as a value rather than a string.

Julian dates start at noon.

All 4 digits of the year must be used to ensure correct Julian values beyond the year 2000.

JULDATE$

x$=JULDATE$(y) Where y is a Julian date, this will return the date as the string x$.

JULIAN

y=JULIAN(x$) Where x$ is the date to convert, this will return the value y as the Julian date.

VAL(C$)

Returns the numerical value of string C$.

SYNTAX: Y = VAL(C$)

LEN(C$)

Returns the number of characters in C$.

SYNTAX: Y = LEN(C$)

ASC(C$)

Returns the numeric value that is the ASCII code for the first character of C$.

SYNTAX: Y = ASC(C$)

INSTR(C$,D$)

Searches for the first occurrence of string D$ in C$, and returns the position.

```
SYNTAX:   Z=INSTR(C$,D$)
          C$="WI-130 INDICATOR"
          D$="N"
          Z=INSTR(C$,D$)
          The value of Z is now set to 9
```

SPACE$(n)

SPACE$(n) will return a string with the specified number of spaces.

PLEN(C$)

PLEN returns a numeric value (X) which represents the width of a string (C$) in dots for proportional fonts.

```
SYNTAX:   X = PLEN(C$)
```

CHECKSUM(C$,n)

Calculates a checksum value for the given string (C$). The checksum calculation type (n) is and integer value representing the type of checksum to calculate. The possible values for n are: 1 = 32 Bit Sum; 2 = 8 bit XOR; 3 = CCITT method CRC16; 4 = XMODEM method CRC16; and 5 = CRC-16; The value returned is 32 bits, but not all calculation methods use the whole resolution.

ISALPHA(C$)

Returns a true (-1) if any of the characters in the string are anything but a digit (0-9), decimal point(.), or minus sign (-).

Weighing Functions

DOZERO

This command zeroes the scale if no motion is detected.

DOUNITS

This command switches the display to the next valid unit of measure for all scales. (If other scales are not on the same unit of measure the current scale is incremented to the next unit of measure and the rest of the scales are forced to that same unit of measure)

DOTARE

This command tares the scale if no motion is detected.

DOPRINT

This command prints the selected default print format as previously configured if no motion is detected. The default print format is not printed if PRINT_KEY subroutine appears in the program.

DOACCUM

This command requests an Accumulate event. The weight must be stable within the motion window parameters for the accumulation to occur.

DOSELECT

This command switches the display to the next valid active displayable value (gross, net, tare . . .).

DOSECOND

This command switches the display to the next valid active displayable value (gross, net, tare . . .) for the secondary display value.

DOBASE

This command switches to the next configured and enabled base.

RESETMIN

RESETMIN command resets the value of the system variable MINIMUM to the current value on the scale platform.

RESETMAX

RESETMAX command resets the value of the system variable MAXIMUM to the current value on the scale platform.

The average weight used in these calculations is based upon the system variable GROSS weight and is affected by filtering settings from the configuration menu.

AVGSTART

Starts averaging for in-motion systems.

AVGSTOP

This returns the average weight (X) on the scale since the last AVGSTART command or for the last 256 weight conversions, whichever is shorter.

SYNTAX: X=AVGSTOP

INMOTION(startIO,stopIO,startLocation,quantity,min,max)

This keyword configures the firmware level inmotion system. This is different from the AVGSTART and AVGSTOP keywords.

Explanation of parameters:

startIO - IO location (setpoint) for the input to start the weighment. Positive value indicates start on activate, negative indicates start on deactivate.

stopIO - IO location (setpoint) for the input to stop the weighment. Positive value indicates stop on activate, negative indicates stop on deactivate.

startLocation - Beginning Store/Recall location to store values from in motion weighing.

quantity - Number of values to store beyond startLocation before wrapping around back to startLocation.

startLocation + quantity - New insertion pointer for store/recalls

min - Minimum (inclusive) acceptable gross weight.

max - Maximum (inclusive) acceptable gross weight.

The Store/Recall value at recall(startLocation+quantity) holds the value

stopIO+1 = Accept Setpoint
stopIO+2 = Reject Setpoint
stopIO+3 = Over Setpoint
stopIO+4 = Under Setpoint
Note::These setpoints must be configured for Basic control Activate and Deactivate.

Example:

```
SUB SYSTEM_STARTUP
 dispmode=1
 store(0,0,1023)
 minval=0
 maxval=50
 inmotion(3,-4,0,1000,minval,maxval)
END SUB
```

```
SUB SETPT4_DEACT
 wt=RECALL(x)
 fmtprint(1)
 x=x+1
 if x>1000 then x=0
END SUB

SUB F1_KEY
 inmotion(3,-4,0,1000,minval,maxval)
END SUB

SUB F2_KEY
 input “Enter Min: “,minval
END SUB

SUB F3_KEY
 input “Enter Max: “,maxval
END SUB
```

GETCONV

Gets the conversion factor of the currently displayed unit measure.

SYNTAX: X = GETCONV

DOSAMPLE

Performs the entire sample process. This handles the errors and prompting. The end result is placed in PCWT and SAMPACCY.

PCWTSAMP(SAMPLEsize)

This keyword takes the sample size and tries to calculate a piece weight based on the weight on the scale.

PCWTZERO

This keyword performs a finer zero operation based on the same algorithm the sampling routine uses.

Memory

There are 8192 numeric storage locations and 4096 sixteen character string storage locations available in non-volatile memory. These memory locations are reusable and will remain through a power loss. Four commands are available for accessing these memory locations:

Locations start at 0 numerics go from 0 to 8191
strings go from 0 to 4095

If the optional expanded ram chips are installed the number of numerics changes to 16384 and the number of strings changes to 8192.

n goes from 0 to 16383
m goes from 0 to 8191

STORE(loc, #value, hiloc)

Stores #value in the loc numeric memory location. If a number for hiloc is included, a fill is performed thereby storing the value in all locations from loc to loc+hiloc.

STORE$(loc, "string value", hiloc)

Stores “string value” in the loc string memory location, 16 characters maximum per location. If a number for hiloc is included, a fill is performed, thereby storing the string value in all locations from loc to loc+hiloc.

RECALL(loc)

Recalls the loc numeric memory location and assigns the value stored there to your variable (X).

Syntax: X = RECALL(loc)

RECALL$(loc)

Recalls the loc string memory location and assigns the value stored there to your variable (C$).

SYNTAX: C$=RECALL$(loc)

FIND(VAR,x,y)

Searches numeric memory slots between x and y for a variable and returns the memory location. Returns a (-1) if not found.

SYNTAX: Z = FIND (VAR,x,y)

Optional Argument ‘type’:

SYNTAX: Z = FIND (VAR,x,y,type)

type - provides optional search patterns. If type is not provided, the default is to find an equal value (type = 0) The search options are as follows; type =

0 - Find an equivalent to the given value (=)
1 - Find the first value that is less than the given value(<)
2 - Find the first value that is less than or equal to the given value (<=)
3 - Find the first value that is greater than the given value (>)
4 - Find the first value that is greater than or equal to the given value (>=)
5 - Find the first largest value within the given range (VAR is ignored) (max)
6 - Find the first smallest value within the given range (VAR is ignored) (min)
7 - Find the pass through value within the given range (pass through)

Pass through is defined as the index of the first value from (x+1) to (y) that is greater than the given value where VAR > value(x), or the index of the first value from (x+1) to (y) that is less than the given value where VAR < value(x) If VAR = value(x) then x = x + 1.

FINDSTR(C$,x,y)

Searches numeric memory slots between x and y for an equivalent string and returns the memory location. Returns a (-1) if not found.

SYNTAX: Z = FINDSTR (C$,x,y)

Optional Arguments ‘type’ and ‘nocase’:

SYNTAX: Z = FINDSTR (VAR,x,y,nocase,type)

nocase - if set to true (nonzero) the strings are compared after being converted to all upper case letters. The default is False (0).

type - provides optional search patterns. If type is not provided, the default is to find an equivalent string (type = 0) The search options are as follows; type =

0 - Find an equivalent string to the given string (=)
1 - Find the first string that is less than the given string(<)
2 - Find the first string that is less than or equal to the given string (<=)
3 - Find the first string that is greater than the given string (>)
4 - Find the first string that is greater than or equal to the given string (>=)
5 - Find the first largest string within the given range (VAR is ignored) (max)
6 - Find the first smallest string within the given range (VAR is ignored) (min)
7 - Find the pass through string within the given range (pass through)

> Pass through is defined as the index of the first string from (x+1) to (y) that is greater than the given string where VAR > value(x), or the index of the first string from (x+1) to (y) that is less than the given string where VAR < value(x) If C$ = string(x) then x = x + 1.

8 - Find the string that contains the given string (Contains)
9 - Find the string that is contained by the given string (is contained by)
10 - Find the first string that is less than the given string(<). This selection differs from #1 in that the search criteria is determined by the length of the string, then by the normal string compare conventions.
11 - Find the first string that is less than or equal to the given string (<=). This selection differs from #2 in that the search criteria is determined by the length of the string, then by the normal string compare conventions.
12 -Find the first string that is greater than the given string (>). This selection differs from #3 in that the search criteria is determined by the length of the string, then by the normal string compare conventions.
13 -Find the first string that is greater than or equal to the given string (>=).This selection differs from #4 in that the search criteria is determined by the length of the string, then by the normal string compare conventions.

Piece Weight Look Up (PWLU) Protocol

If you want to use the PWLU features of the PC-820, use the variables, keywords, and subroutines given below:

PWLU System Variables

Weight variables (PWLU_T and PWLU_PW) are retrieved and set in the calibration unit of measure of the scale. They are stored internally as calibration unit of measure.

PWLU_MSG – Message Status Value (See *PWLU_MSG Table*)
PWLU_T – Tare to send or receive from PWLU
PWLU_DS$ – Description for part to send or receive from PWLU (Up to 16 Chars)
PWLU_D2$ – Description 2 for part to send or receive from PWLU (Up to 16 Chars)
PWLU_D3$ – Description 3 for part to send or receive from PWLU (Up to 16 Chars)
PWLU_PW – Piece Weight to send or receive from PWLU
PWLU_ST – Status value for piece weight update and new P/N (See *PWLU_ST Table*)
PWLU_NUM - Assigned scale number. (1-32)
PWLU_AN$ - Transaction counter from PWLU (Recv'd only). Increments everytime ACC+ or ACC- used from any scale connected to PWLU.
PWLU_CN$ - Control number from PWLU (Recv'd only). Tracking number of where the data is coming from. Could be used for lot control also. See PWLU User's manual.

PWLU Keywords

PWLU_GET("P/N") – Sends the proper set of commands to the PWLU program to request part information from the computer. The "P/N" is the part number of the item requested. The retrieved data appears in the PWLU system variables. The PWLU subroutine is queued on a response from the computer or if the communications have timed out.

Possible PWLU_MSG outcomes:
error → 4
data accepted → 3
timeout → 10

PWLU_SET("P/N") – Sends the part information to the computer in hopes that it will update the part database with the information in the PWLU system variables. The PWLU_RECEIVE subroutine is queued upon receipt from the computer that an error has been made, the data was accepted, or that a time-out has occurred.

Possible PWLU_MSG outcomes:
error update disabled → 7
error New P/N disabled → 4
accepted New P/N → 5
accepted update → 6
timeout → 11

PWLU_ACC("P/N", ACC) – Sends the part information to the computer in hopes that it will record a transaction and update the accumulator. The PWLU_RECEIVE subroutine is queued upon receipt from the computer that an error has been made, the data was accepted, or that a time-out has occurred. All PWLU data should be correctly set since it is also recorded in the transaction. **This keyword does not update the parts database.**

Possible PWLU_MSG outcomes:
error → 4
data accepted → 8
Timeout → 12

PWLU_STR(NUM,STR$) – Allows the setting of the miscellaneous strings used for transactions in the PWLU_ACC keyword. Use this keyword to set any of the ten strings (NUM=1-10) and they will be transmitted as part of the transaction packet initiated by PWLU_ACC. All strings can be set to the same value by using '0' in the NUM field. (Useful for clearing all of the values.) These strings are NOT cleared when a transaction is processed.

PWLU Event Subroutines

PWLU_RECEIVE – This subroutine is parsed after a command has been received from the computer or when the PWLU_GET, PWLU_SET, or PWLU_ACC commands have timed out with no response from the computer.

PWLU_ST Table

0 – No updating or adding enabled
1 – Part database update enabled
2 – Add new parts enabled
3 – Both enabled

PWLU_MSG Table

0 – No Message
1 – Scale ONLINE (PWLU_ST & PWLU_NUM are updated when ONLINE message is received from the computer)
2 – Scale OFFLINE
3 – Item data is ready
4 – Item not found
5 – Added new P/N
6 – Updated Item Information
7 – Item update is not allowed
8 – Accumulation transaction accepted
9 – Item not sent (communications problem)
10 – PWLU_GET time-out
11 – PWLU_SET time-out
12 – PWLU_ACC time-out
13 – Time has been set by computer

Appendix 3: Subroutine Examples

Sample : GETCOM$

This is an example of using GETCOM2 for data storage:

```
SUB SYSTEM
dim X$*32
dim SPACE$*32
ROOM$=“<32-Blanks>”
DISPMODE(17)
KEY 1,”RECALL”
END SUB
```

This section reserves room in memory by dimensioning the variables X$ and SPACE$ each to 32 characters. Display mode is set to 17 (See *Appendix 1*). KEY labels softkey 1 as RECALL.

```
SUB COM2_MESSAGE
X$ = GETCOM$(2)
X$ = X$+LEFT$(SPACE$,(32-LEN(X$))
A$=LEFT$(X$,16)
B$=RIGHT$(X$,16)
STORE$(1,A$)
STORE$(2,B$)
END SUB
```

This section retrieves a message from serial port #2 and then makes it 32 characters long and splits this into two 16-character string variables (A$, B$), and stores them in locations 1 and 2.

```
SUB F1_KEY
C$=RECALL$(1)
D$=RECALL$(2)
X$=C$+D$
PRINT X$
END SUB
```

This section recalls the two 16-character strings from memory, combines them and prints them to the display.

Sample : SHOWVAR

```
SUB SYSTEM_STARTUP
dim WT
dim VAR$
dim LEGEND1$
dim LEGEND2$
dim PREC
VAR$=”WT”
LEGEND1$=”kg”
LEGEND2$=”brutos”
PREC=2
SHOWVAR(VAR$,LEGEND1$,LEGEND2$,PREC)
SETTIMER(2,0.5)
ACTVALUE(11)
END SUB
```

This subroutine reserves memory by dimensioning the variables of the program. It assigns strings to the variables LEGEND1$ and LEGEND2$ which are used in the SHOWVAR command.

SETTIMER causes the SUB SYSTEM_TIMER2 event to occur every ½ second. ACTVALUE allows the variable WT to be displayed where the weight normally is displayed on the PC-820 display.

```
SUB SYSTEM_TIMER2
WT=GROSS
END SUB
```

The SUB SYSTEM_TIMER2 subroutine assigns the system variable GROSS into the variable WT.

Sample: STORE and RECALL

```
SUB SYSTEM_STARTUP
MUSTDIM
dim WORD$
dim LOC
dim PASSWORD
dim GUESS
dim Y
dim VNUM
dim X
WORD$=""
PASSWORD=111
DISPMODE(17)
KEY 1, "NEXT"
KEY 2, "STORE"
KEY 4, "CLRMEM"
KEY  5,  "PREV"
LOC=RECALL(1000)
END SUB

SUB F2_KEY
LOC=LOC+1
input "NAME",WORD$
STORE$(LOC,WORD$)
INPUT "PHONE#",VNUM
STORE(LOC,VNUM)
STORE(1000,LOC)
END SUB

SUB F1_KEY
X=X+1
IF X>LOC THEN X=LOC
WORD$ = recall$ (X)
VNUM = RECALL (X)
PRINT WORD$;" ";VNUM
END SUB

SUB F4_KEY
GUESS=0
INPUT "PASSWORD?",GUESS
IF PASSWORD=GUESS THEN
 FOR Y=0 TO 1024
 STORE(Y,0)
 STORE$(Y,"")
 NEXT
 LOC=0
END IF
END SUB

SUB F5_KEY
X=X-1
IF X<0 THEN X=0
WORD$ = recall$ (X)
VNUM = RECALL (X)
PRINT WORD$;" ";VNUM
END SUB
```

This program gives you an example of storage and retrieval of names and numbers from memory.

Sample Setpoint (Cutoff) Application

```
SUB SYSTEM_STARTUP
mustdim
dim cutoff1
dim cutoff2
dim disp
cutoff1=RECALL(1023)
cutoff2=RECALL(1022)
END SUB

SUB F1_KEY
disp=dispmode
dispmode=6
input “Enter Cutoff#1”,cutoff1
if lastkey=27 then
 dispmode=disp
  exit sub
end if
input “Enter Cutoff#2”,cutoff2
if lastkey=27 then
 dispmode=disp
  exit sub
end if
dispmode=disp
store(1023,cutoff1)
store(1022,cutoff2)
END SUB



‘print screen of setpoints
‘remote zero and remote tare examples
SUB SETPT1_ACT
dozero
END SUB

SUB SETPT2_ACT
dotare
END SUB
```

Sample Continuous Output

```
‘old way for continuous output
format #1
\x02 G {SEND$} LB\r\n

SUB SYSTEM_STARTUP
DISPMODE=10
ZERO$=”000000"
SETTIMER(1,0.5)
END SUB

SUB SYSTEM_TIMER
GROSS$=STR$(ABS(GROSS))
X=6-LEN(GROSS$)
TEMP$=LEFT$(ZERO$,X)+GROSS$
 IF GROSS<0 THEN
  SEND$=”-”+TEMP$
  ELSE
  SEND$=”+”+TEMP$
 END IF
FMTPRINT(1)
END SUB



‘new way for continuous ouput
‘max is 10 times a second
SUB SYSTEM_STARTUP
contout(1,2)                    ‘format #1 output twice a second
END SUB
```

Sample Enquire via Serial Port

```
‘do a print screen of format 1 and serial port??
‘when a enquire EOM is received
‘print format#1 will be sent
SUB COM2_MESSAGE
fmtprint(1)
END SUB
```

Appendix 4: Error Messages

	Message	Meaning
When Saving	Drive not Ready	No disk in drive
	Disk Write Protected	Write protect tab on diskette is in place.
Edit Menu	Invalid Line Number	A line number that doesn't exist. (Goto Line# in Edit Menu)
	Search Item Not in this Program!	You tried to find something that wasn't there.
Downloading	Begin Transfer.......Bad Block	Cabling problem has occurred, intermittent connection.
Other	Subscript Out of Range	Restart PC-820 SimPoser.
	Out of Memory	1. You have too many applications running in Windows®. 2. Your PC-820 SimPoser configuration file is too large.
	Permission Denied	Do a FILE, SAVE AS c:\simposer\dat then a FILE, SAVE AS A:
	Disk not ready	Make sure disk is inserted properly.
Non-Mouse Users:	ALT-F6	Gets you back to the main screen if you were in either SETPOINTS or CONFIGURE

Appendix 5: ASCII Chart

Code #	Cont. Char.	Print Char.
0	NUL	
01	SOH	☺
02	STX	☻
03	ETX	♥
04	EOT	♦
05	ENG	♣
06	ACK	♠
07	BEL	
08	BS	
09	HT	
010	LF	LF
011	VT	♂
012	FF	FF
013	CR	CR
014	S0	♫
015	S1	☼
016	DLE	4
017	DC1	3
018	DC2	ø
019	DC3	Ø
020	DC4	ß
021	NAK	§
022	SYN	
023	ETB	—
024	CAN	↑
025	EM	↓
026	SUB	→
027	ESC	←
028	FS	—
029	GS	—
030	RS	5
031	US	6
032	SP	
033	!	!
034	"	"
035	#	#
036	$	$
037	%	%
038	&	&
039	'	'
040	(	(
041	)	)
042	*	*
043	+	+
044	,	,
045	-	-
046	.	.
047	/	/
048	0	0
049	1	1
050	2	2
051	3	3
052	4	4
053	5	5
054	6	6
055	7	7
056	8	8
057	9	9
058	:	:
059	;	;
060	<	<
061	=	=
062	>	>
063	?	?
064	@	@
065	A	A
066	B	B
067	C	C
068	D	D
069	E	E
070	F	F
071	G	G
072	H	H
073	I	I
074	J	J
075	K	K
076	L	L
077	M	M
078	N	N
079	O	O
080	P	P
081	Q	Q
082	R	R
083	S	S
084	T	T
085	U	U
086	V	V
087	W	W
088	X	X
089	Y	Y
090	Z	Z
091	[	[
092	\	\
093	]	]
094	^	^
095	_	_
096	`	`
097	a	a
098	b	b
099	c	c
0100	d	d
0101	e	e
0102	f	f
0103	g	g
0104	h	h
0105	i	i
0106	j	j
0107	k	k
0108	l	l
0109	m	m
0110	n	n
0111	o	o
0112	p	p
0113	q	q
0114	r	r
0115	s	s
0116	t	t
0117	u	u
0118	v	v
0119	w	w
0120	x	x
0121	y	y
0122	z	z
0123	{	{
0124	|	|
0125	}	}
0126	~	~
0127	DEL	□
0128	NA	Ç
0129	NA	ü
0130	NA	é
0131	NA	â
0132	NA	ä
0133	NA	à
0134	NA	å
0135	NA	ç
0136	NA	ê
0137	NA	ë
0138	NA	è
0139	NA	ï
0140	NA	î
0141	NA	ì
0142	NA	Ä
0143	NA	Å
0144	NA	É
0145	NA	æ
0146	NA	Æ
0147	NA	ô
0148	NA	ö
0149	NA	ò
0150	NA	û
0151	NA	ù
0152	NA	ÿ
0153	NA	Ö
0154	NA	Ü
0155	NA	¢
0156	NA	£
0157	NA	¥
0158	NA	₧
0159	NA	ƒ
0160	NA	╩
0161	NA	í
0162	NA	ó
0163	NA	ú
0164	NA	ñ
0165	NA	Ñ
0166	NA	ª
0167	NA	º
0168	NA	¿
0169	NA	⌐
0170	NA	¬
0171	NA	½
0172	NA	¼
0173	NA	¡
0174	NA	«
0175	NA	»
0176	NA	░
0177	NA	▒
0178	NA	▓
0179	NA	│
0180	NA	┤
0181	NA	╡
0182	NA	╢
0183	NA	╖
0184	NA	╕
0185	NA	╣
0186	NA	║
0187	NA	╗
0188	NA	╝
0189	NA	╜
0190	NA	╛
0191	NA	┐
0192	NA	└
0193	NA	┴
0194	NA	┬
0195	NA	├
0196	NA	─
0197	NA	┼
0198	NA	╞
0199	NA	╟
0200	NA	╚
0201	NA	╔
0202	NA	╩
0203	NA	╦
0204	NA	╠
0205	NA	═
0206	NA	╬
0207	NA	╧
0208	NA	╨
0209	NA	╤
0210	NA	╥
0211	NA	╙
0212	NA	╘
0213	NA	╒
0214	NA	╓
0215	NA	╫
0216	NA	╪
0217	NA	┘
0218	NA	┌
0219	NA	█
0220	NA	▄
0221	NA	▌
0222	NA	▐
0223	NA	▀
0224	NA	α
0225	NA	ß
0226	NA	Γ
0227	NA	π
0228	NA	∑
0229	NA	σ
0230	NA	µ
0231	NA	τ
0232	NA	Φ
0233	NA	Θ
0234	NA	Ω
0235	NA	δ
0236	NA	∞
0237	NA	∅
0238	NA	ε
0239	NA	∩
0240	NA	≡
0241	NA	±
0242	NA	≥
0243	NA	≤
0244	NA	⌠
0245	NA	⌡
0246	NA	÷
0247	NA	≈
0248	NA	°
0249	NA	•
0250	NA	·
0251	NA	√
0252	NA	ⁿ
0253	NA	²
0254	NA	■
0255	NA	

Appendix 6: Alphabetical Listing of WT-BASIC Commands

Weigh-Tronix
1000 Armstrong Dr.
Fairmont, MN 56031 USA
Telephone: 507-238-4461
Facsimile: 507-238-4195
e-mail: industrial@weigh-tronix.com
www.wtxweb.com

Weigh-Tronix Canada, ULC
217 Brunswick Blvd.
Pointe Claire, QC H9R 4R7 Canada
Telephone: 514-695-0380
Facsimile: 514-695-6820

WEIGH-TRONIX
Weighing Products & Systems