User's Manual

P/N 067212-003

TRAKKER® T2090 Batch Hand-Held Computer

Intermec

A UNOVA Company

Intermec Technologies Corporation
6001 36th Avenue West
P.O. Box 4280
Everett, WA 98203-9280

U.S. service and technical support: 1-800-755-5505
U.S. media supplies ordering information: 1-800-227-9947

Canadian service and technical support: 1-800-688-7043
Canadian media supplies ordering information: 1-800-268-6936

Outside U.S. and Canada: Contact your local Intermec service supplier.

Manual Change Record

This page records changes to the manual. The manual was released at Revision 001.

Revision	Date	Description of Change
001	5/98	Original release.
002	7/98	Updated the manual.
003	1/99	Updated for the Version 2.0 software release: • Added two new reader commands: Receive File XMODEM-1K Transmit File XMODEM-1K • Added 18 new application functions: im_cputs im_draw_underline im_get_kbmode im_get_system_julian_date im_set_2of5_lengths im_set_com_mode im_set_kbmode im_set_port_direction im_set_scanning im_timed_status_line im_tm_callback_cancel im_tm_callback_register im_xm_receive_file im_xm1k_receive_file im_xm_transmit_file im_xm1k_transmit_file im_ym_receive_file im_ym_transmit_file Corrected the Default Configuration reader command and added the Save Configuration to File reader command. Revised the Interleaved 2 of 5 configuration command to include the Case Code option, which lets you read labels with 6 or 14 characters. Revised the im_set_scanning application function. Added a section on using CTTY sessions to manage files. Added support for the underline attribute in im_draw_underline, im_cputs, im_putchar, and im_receive_field. Added a note to the “Character Set” section in Appendix A explaining that the character set does not match DOS code page 850. Expanded the description of TRANSFER.EXE and its options.

Contents

6

Configuration Command Reference

8

Hardware Specifications

A

Appendix A - Demo Software

B

Appendix B - Microsoft Visual C/C++ Settings

C

Appendix C - Full ASCII Charts

Before You Begin

This section introduces you to standard warranty provisions, safety precautions, warnings and cautions, document formatting conventions, and sources of additional product information.

Warranty Information

To receive a copy of the standard warranty provision for this product, contact your local Intermec support services organization. In the U.S. call 1-800-755-5505, and in Canada call 1-800-688-7043. Otherwise, refer to the Worldwide Sales & Service list that ships with this manual for the address and telephone number of your Intermec sales organization.

Safety Summary

Your safety is extremely important. Read and follow all warnings and cautions in this book before handling and operating Intermec equipment. You can be seriously injured, and equipment and data can be damaged if you do not follow the safety warnings and cautions.

Do not repair or adjust alone Do not repair or adjust energized equipment alone under any circumstances. Someone capable of providing first aid must always be present for your safety.

First aid Always obtain first aid or medical attention immediately after an injury. Never neglect an injury, no matter how slight it seems.

Resuscitation Begin resuscitation immediately if someone is injured and stops breathing. Any delay could result in death. To work on or near high voltage, you should be familiar with approved industrial first aid methods.

Energized equipment Never work on energized equipment unless authorized by a responsible authority. Energized electrical equipment is dangerous. Electrical shock from energized equipment can cause death. If you must perform authorized emergency work on energized equipment, be sure that you comply strictly with approved safety regulations.

Warnings, Cautions, and Notes

The warnings, cautions, and notes in this manual use the following format.

Warning
A warning alerts you of an operating procedure, practice, condition, or statement that must be strictly observed to avoid death or serious injury to the persons working on the equipment.

Avertissement
Un avertissement vous avertit d'une procédure de fonctionnement, d'une méthode, d'un état ou d'un rapport qui doit être strictement respecté pour éviter l'occurrence de mort ou de blessures graves aux personnes manupulant l'équipement.

Caution
A caution alerts you to an operating procedure, practice, condition, or statement that must be strictly observed to prevent equipment damage or destruction, or corruption or loss of data.

Conseil
Une précaution vous avertit d'une procédure de fonctionnement, d'une méthode, d'un état ou d'un rapport qui doit être strictement respecté pour empêcher l'endommagement ou la destruction de l'équipement, ou l'altération ou la perte de données.

Notes: *Notes are statements that either provide extra information about a topic or contain special instructions for handling a particular condition or set of circumstances.*

About This Manual

This manual contains all of the information necessary to install, operate, configure, troubleshoot, and maintain the TRAKKER T2090.

This manual was written for analysts and programmers who operate and program the T2090 to a network or system. A basic understanding of DOS, programming, and data communications is necessary.

Format Conventions for Input From a Keyboard or Keypad

This table describes the formatting conventions for input from PC or host computer keyboards and reader keypads:

Convention	Description
`Special` text	Shows the command as you should enter it into the reader.
Italic text	Indicates that you must replace the parameter with a value.
Bold text	Indicates the keys you must press on a PC or host computer keyboard. For example, "press **Enter**" means you press the key labeled "Enter" on the PC or host computer keyboard.

Format Conventions for Commands

This manual includes sample commands that are shown exactly as you should type them on your reader. The manual also describes the syntax for many commands, defining each parameter in the command. This example illustrates the format conventions used for commands:

This table defines the conventions used in the example:

Convention	Description
`Special` font	Commands appear in this font. You enter the command exactly as it is shown.
Italic text	Italics indicate a variable, which you must replace with a real value, such as a number, filename, or keyword.
[ ]	Brackets enclose a parameter that you may omit from the command. Do not include the brackets in the command.
Required parameters	If a parameter is not enclosed in brackets [ ], the parameter is required. You **must** include the parameter in the command; otherwise, the command will not execute correctly.
where	This word introduces a list of the command's parameters and explains the values you can specify for them.

Other Intermec Manuals

The following manuals provide additional information about capturing data with your T2090.

Manual	Intermec Part Number
TRAKKER T2090 Hand-Held Batch Computer Getting Started Guide	067215
TRAKKER T2090 Battery Replacement Instruction Sheet	067378
Data Communications Reference Manual	044737
The Bar Code Book	051241

You are invited to visit our web site on the Internet where you will find other Intermec manuals in PDF format. Our Internet address is www.intermec.com.

1

Getting Started

This chapter contains an overview of the TRAKKER T2090's features and offers basic information about its use.

Overview of the TRAKKER T2090

The TRAKKER T2090 is a very lightweight, pocket-size computer and bar code scanner. Designed for true one-hand operation, the T2090 fits very well into repetitive data collection applications, such as inventory management, route sales, and asset tracking. Based on a DOS-compatible platform, the T2090 is well suited for application development in a standard DOS or Windows environment.

When developing applications for the T2090, keep the following information in mind. The T2090 shares some of the basic building blocks of PC architecture, such as processor, memory, and communications. It differs from PCs, however, in that the display is not 80 or 132 columns and not color, there are no floppy drives, and there are certain limitations in terms of available keys. Several standard DOS commands do not produce the same results on the T2090 as they do on a DOS PC. Graphics are handled differently. Also, the T2090 interfaces with peripherals using a communications cradle instead of a direct connect.

Applications created for the T2090 should use the application functions in Intermec's software library to open COM ports and read bar codes rather than equivalent functions from other C libraries.

Technical support at Intermec is always ready to assist you with any questions you may have about how the T2090 operates.

Features

The TRAKKER T2090 has the following external features:

- 8 line by 20 character display panel with backlight
- A scanner read LED indicator
- F1, F2, and F3 function keys
- Large, centrally located ENTER key
- Backlight on / off key
- Keyboard mode lock key (alpha or numeric modes)
- 13 alphanumeric data entry keys
- Integrated bar code scanner
- Rechargeable battery pack

The following picture shows the locations of the various features.

Charging the Battery

Before using the T2090, you need to fully charge the battery pack. This takes four hours or less. For information about replacing the battery pack, see the *TRAKKER T2090 Battery Replacement Instruction Sheet* (part number 067378) included with the TRAKKER T2090.

To charge the battery:

- Place the unit into the EZ Dock Docking System. If you are charging the battery for the first time, let it charge a full four hours.

While the battery is charging, the multi-function LED blinks red. Once charged, the battery continues to receive a "trickle" charge while sitting in the dock. This is indicated by a steady green LED.

Depending on the application you are running, you can operate the TRAKKER T2090 from 10 to 20 hours on a single charge.

***Note**: The battery charger operates normally in ambient temperatures between 50° F and 104° F (10° C and 40° C). If you continually use the docking system in an environment outside this range, the battery may not charge properly.*

209XU.005

Collecting Data

With the TRAKKER T2090, you can collect data either by entering the information using the EZ Key Keypad or by scanning bar codes. The following two sections describe both methods.

About the EZ Key Keypad

The TRAKKER T2090's EZ Key keypad has 22 keys, 13 of which are used to enter the letters A through Z, the numbers 0 through 9, the dash mark (-), period (.), and apostrophe ('). The three unlabeled keys just below the display are the F1, F2, and F3 function keys.

To switch the unit on and off:

- Press the ⏻ key to turn the unit on and off.

To switch between alpha and numeric modes:

- Press the ⇧ key. If the unit is in numeric mode, it switches to alpha mode, and will stay in alpha mode until you press the ⇧ key again to switch it back to numeric.

When you first start up the TRAKKER T2090, you will see "123" on the display screen. This indicates that the keypad is in numeric mode. Pressing the ⇧ key toggles the keypad mode to alpha and changes the display indicator to "ABC."

To turn the backlight on:

- Press the backlight key.

The backlight automatically turns off when the unit goes into the SLEEP state, but turns back on when you press any key. See Chapter 2 for information about the unit's power states.

To enter a line of data:

- Press the **ENTER** key.

Entering Data Using the EZ Key Keypad

To enter numbers:

1. If the keypad is not in numeric mode, press the ⇧ key. Now when you press any letter/number key, you will enter the number on the key.
2. Press the key(s) labeled with the number(s) you want to enter. The display shows the number or numbers that you pressed.
3. Press **ENTER** to complete the data entry.

To enter letters:

1. If the keypad is not in alpha mode, press the ⇧ key. Now when you press any letter/number key, you will enter the letter shown on the left or right side of the key.
2. Press the key labeled with the letter you want to enter, pressing on the left or right side of the key, depending on the letter you want. For example, to enter G, press the key labeled with G and H, pressing on the left side of the key. To clear a letter, press the BS (backspace) key.
3. Press **ENTER** to complete the data entry.

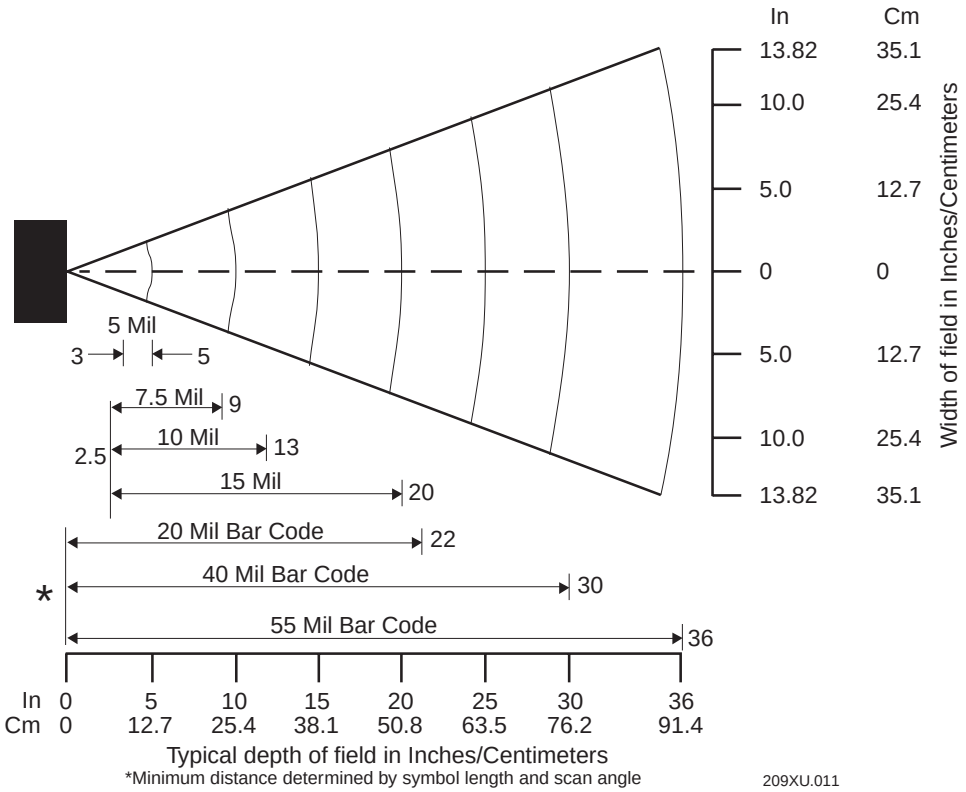
***Note**: The ability to correct mistakes is controlled by the application software you are running.*

About Scanning

The application software running on your TRAKKER T2090 controls when the bar code scanner may be activated. Intermec supplies, through the software libraries, all the necessary functions to control and configure the scanner (see Chapter 6).

If the unit is not docked and a PSK function is being used (im_receive_input or im_receive_field), and if no character input keys have been pressed since the call was made, pressing the Enter key causes a scan. If character keys have been pressed, the Enter key returns the string containing the keystrokes.

The readability of a bar code depends on the type of bar code symbology, distance, and the condition or clarity of the printed material. The following diagram approximates the scanning range for various bar code densities. A slight amount of scanning distance allowance must be made for the device case assembly the outer edge of which is approximately 1/2-inch from the actual laser scanner.



Scanning Bar Codes

This section applies only if your TRAKKER T2090 has an integrated bar code scanner. The application software running on the unit controls when the bar code scanner is activated.

***Note**: The scanner in the T2090 cannot be used simultaneously with the serial port or when the unit is in the dock because the internal UART is shared between the scanner and the communications serial port.*

To scan a bar code:

1. Hold the unit between 1-36 inches from the bar code label.
2. Press and hold **ENTER**. Make sure the red scanner beam spans the entire bar code. The T2090 provides feedback:
   - When the scanner is activated, the LED displays red.

- If the bar code is read successfully, the LED indicator turns green for one second. You also hear a beep if the unit's audible feature has been enabled.
- If the bar code is not read successfully, the LED indicator turns red for one second.
- If you release **ENTER** before the bar code is read successfully, the scanner and LED turn off.

EZ Dock Docking System

The EZ Dock Docking System includes a Power Module and 2-Slot Cradle. The docking system serves as a storage/charging location for the TRAKKER T2090 and provides a communications link between the T2090 and the host system. The RJ-45 connectors integrated into the module can be configured for RS-232/485 device-to-host communications.

To add communications and charging capacity, you can attach up to four more 2-Slot Cradles onto the initial Power Module with 2-Slot Cradle. This increases the capacity to 10 slots from one power source. Configurations with one or two cradles should use Intermec Power Supply part number 590822. Configurations with three to five cradles require the larger Power Supply, part number 590817.

The docking system includes one screw and one wing nut, which you can use to hang the unit on a wall or secure it to any horizontal surface.

Transferring Data

To transfer data between the TRAKKER T2090 and a host PC, you need the following:

- T2090 and an EZ Dock Docking System
- Universal Power Supply
- Cable from Power Module to PC host (RJ-45 to DB-25)

Transferring Files via XMODEM:

The program, XFER.EXE is preloaded on the ROMDISK (drive A). This program implements XMODEM file transfer and supports both Standard Checksum and CRC16. The program also supports both 128-byte block size and 1K block size.

Note: *You can use a more flexible version of this program called TRANSFER.EXE. For help, see "Utility Programs" in Chapter 2.*

To transfer a file from a PC to the T2090:

1. On your PC, run a communications program that supports XMODEM file transfer. (You can use Intermec's FileCopy program, described in "Using the Serial Port to Transfer Applications and Files" later in this section.) If the program provides an options menu for XMODEM settings, choose CRC16 and 1K block size. Set the connection baud rate to match what you use on the T2090.
2. Start the XMODEM send from the PC to the unit.
3. On the T2090, type: XFER -R -B38400 -COM1. (38400 is the baud rate used in this example).
4. Place the T2090 in the dock.

To transfer a file from the T2090 to a PC:

1. On your PC, run a communications program that supports XMODEM file transfer. If the program provides an options menu for XMODEM settings, choose CRC16 and 1K block size. Set the connection baud rate to match what you use on the T2090.
2. On the T2090, type: XFER -S -B38400 -COM1 (38400 is the baud rate used in this example).
3. Place the T2090 in the dock.
4. Start the XMODEM receive from the PC to the unit.

***Note**: If you plan to use XMODEM often, put the XFER command in a DOS batch file and put it on the T2090's C drive.*

Setting Up for File Sharing

To set up the TRAKKER T2090 for file sharing:

1. Set up the Power Module with attached 2-Slot Cradle and apply power.
2. Place the unit into the cradle.
3. Make the cable connection between the cradle and the host PC.

   ***Note**: An executable file named REMSERV.EXE resides in ROM as well as an executable batch file named RT.bat.*

4. Turn the unit on.
5. At the C> prompt, type [RT] and press **Enter**.

To set up the host PC for data communication:
Log in to the directory where the file REMDISK.EXE is stored. This is an executable file supplied by Intermec and resides in the directory chosen for installing PSK (c:\intermec, by default).

***Note**: REMDISK must be run in DOS mode; it cannot be run safely from a DOS window under Windows or Windows 95. If you are running Windows, choose EXIT from the file menu to return to DOS. If you are running Windows 95, either choose Shutdown from the Start menu and select "Restart the computer in MS-DOS Mode" or reboot and hold down the F8 key. When you get a list of choices, select "Command Prompt only."*

At the C : prompt, type this command:

```
remdisk /COM[x]: /B38400+ /+T10
```

where *x* is the number for the COM port on the PC you are using to communicate with the T2090/cradle hardware.

You should see the following message on the PC display:

```
Remote Disk VV 6.22 ( Revision XX )

Copyright ( C ) 1989-1997  Datalight Inc.

Installed as drive ? :  /com[1 or 2]/B38400+/T3
```

The flash drive ROM within the T2090 is now logged in as the next lettered logical drive (such as D, E, or F) on the PC and will respond to normal DOS commands, such as DIR and DEL. You can now copy, add, or delete files as needed.

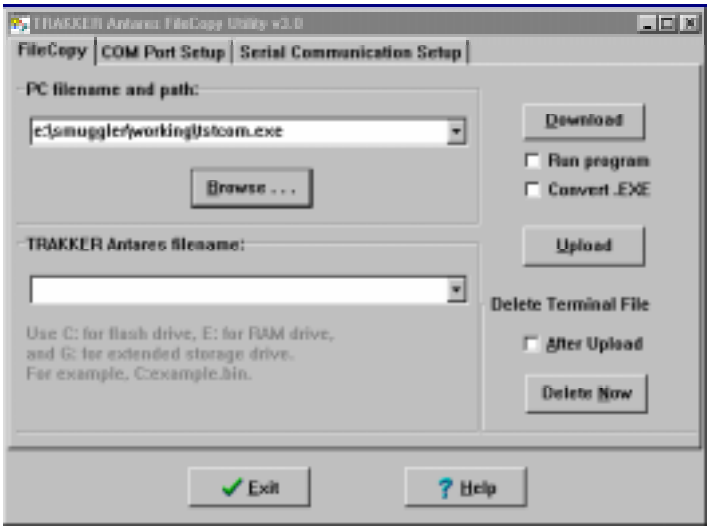
Using the Serial Port to Transfer Applications and Files

The instructions in this section explain how to use the FileCopy utility to download or upload applications and files between a PC or host computer and the T2090 using serial communications. First, you connect the power module's serial port to the host, then you use the FileCopy utility that ships with the T2090's PSK.

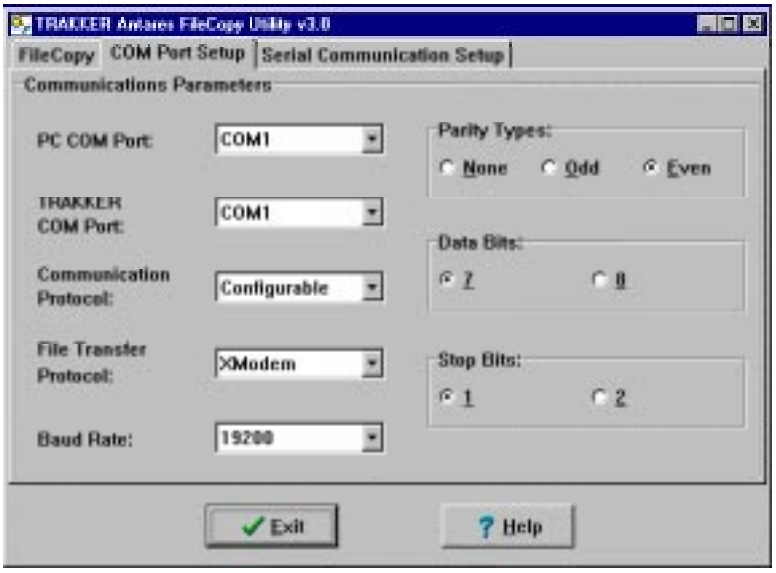
***Note**: If the T2090 is not waiting for input from a PSK function, you need to start the transfer program on the TRAKKER to receive or send the file. See "Transferring Files via XMODEM."*

To download or upload applications and files to or from the T2090 computer:

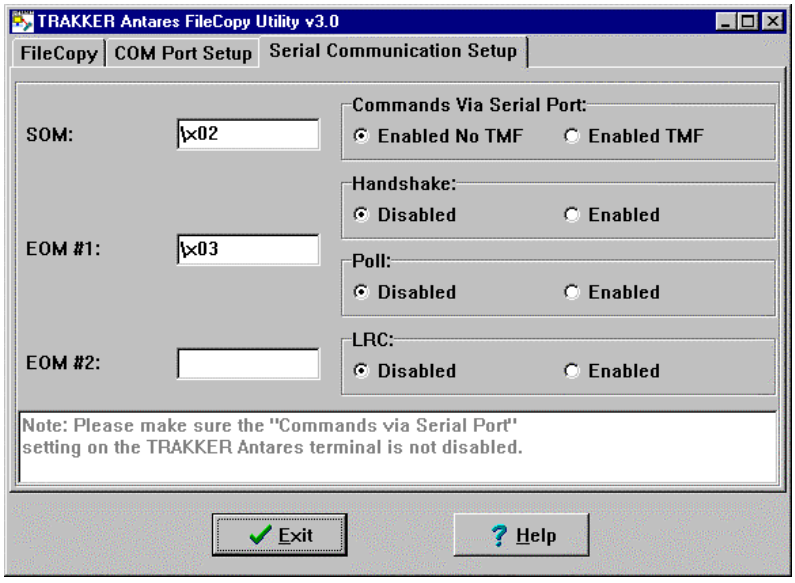
1. Connect the T2090 to your PC.
2. Start Microsoft Windows on your PC.
3. From Program Manager, start FileCopy. The FileCopy utility screen appears. Make sure that the Convert .EXE check box is not checked.



4. Check the serial port and serial communications parameters to verify that the settings for your PC match the values that are set for the T2090 computer's serial port.
   - Select the COM Port Setup. The TRAKKER COM Port must be set to COM1, Communication Protocol must be set to Configurable, and File Transfer Protocol must be set to XMODEM. All other parameters must be set to match the configuration of your T2090.



- Select the Serial Communications Setup tab to verify and configure the PC serial communications parameters.



- Use the application to configure the serial port parameters on the T2090.

5. Select the FileCopy tab to download or upload applications and files.
   - In the PC filename and path filed, type the path and filename (FILENAME.EXE) for the file on your PC. You can select a previously used filename from a list by clicking on the down arrow. To view a list of available files on your PC, click the Browse button.
   - In the TRAKKER Antares filename field, type the drive and filename on the T2090. You can select a previously used filename from a list by clicking on the down arrow.

     You can only download applications to drive C.
6. Click Download to copy the file from the PC to the T2090, or click Upload to copy the file from the T2090 to the PC.
7. Click exit to close the FileCopy utility.

Definitions

These terms are used in this manual:

Term	Definition
Real Memory	The section of memory under 1 megabyte.
ROMDOS	A 6.22-compatible operating system developed by Datalight.

ROMDISK	A virtual disk placed within flash, in accordance with the development tools of Datalight ROM-DOS version 6.22.
FLASHDISK	A virtual hard disk placed within flash in accordance with the development tools of Datalight CardTrick® ver. 3.0.

Managing Files

Because the T2090 keypad does not include the backslash (\) and colon (:) characters, you might want to establish a CTTY session between your T2090 and a PC (or terminal) in order to use DOS commands to manage the files on the T2090. During a CTTY session, the commands that you type on the PC keyboard are executed on the T2090.

Follow these steps to establish a CTTY session:

1. Determine the baud rate that is currently set on the T2090:
   - If CTTY is the first progam to use the serial port after the T2090 has been booted, then the T2090 serial port is set to 2400 baud (the DOS default).
   - Otherwise, the T2090 serial port uses the baud rate set by the last program to use the serial port. For example, if the RT batch file was run last, the baud rate is 38,400. To learn more about the RT batch file, see "REMSERV.EXE, REMOTE.BAT, and RT.BAT" in Chapter 2.
2. At the C> prompt on the T2090, type CTTY COM1 and press **ENTER**.
3. Place the T2090 in the dock. The dock must be connected to the PC.
4. On the PC, start a terminal emulation program, such as HyperTerminal.
5. On the PC, set the serial communications port parameters. For the COM port, specify the PC COM port to which the dock is connected. Set the baud rate to the speed you determined in Step 1. Set 8 data bits, 1 stop bit, and no parity.
6. Press **Enter** on the PC. The T2090 C> prompt should be displayed on the PC. The CTTY session has been established.
7. Type A:RETURN on the PC and press **Enter**. The T2090 A> prompt should be displayed on the PC. You can use the DIR command to list the files on the A drive of the T2090. You can use any DOS command to manipulate the files on the T2090.

   Note: *Drive A on the T2090 is not writable. It is reserved for the pre-loaded T2090 utilities only. For details, see Chapter 2.*

8. To end the CTTY session, type CTTY CON and press **Enter** on the PC.

For more help with CTTY, you can type ? after the command to get syntax information.

2

User Interface

This chapter describes the system software that resides on the TRAKKER T2090 computer and describes the main setup screens and display indicators.

Default System Files

Files Stored on ROMDISK

The following files are stored in the read-only ROMDISK (drive A):

File name	Description
ROM-DOS SYS	DOS system file
CONFIG SYS	Resident unit configuration file
AUTOEXEC BAT	Resident autoexec file
COMMAND COM	DOS command shell
COMMAND HLP	Help file for DOS commands
ATTRIB COM	File attribute program
CHOICE COM	Batch menu choice command program
FDISK COM	Disk utility, allows toggling of boot condition
MEM EXE	Memory interrogator program
MODE COM	Com port mode utility
REMOTE BAT	Batch file for starting remote server from keyboard
REMSERV EXE	Remote server program
TRANSFER EXE	Xmodem transfer program
DLFMT EXE	C drive formatter
RT BAT	Batch file for Quick REMSERV start.
SETUP BAT	Batch file to invoke setup
FWTSETUP EXE	T2090 setup program
RS-485 COM	Utility to change COM1 to RS-485
RS-232 COM	Utility to change COM1 to RS-232
DISKVER COM	Disk version identifier

ROM-DOS.SYS

The DOS system is contained in the ROM-DOS.SYS file, which is loaded into RAM after the BIOS startup completes unless a bootable C drive is detected. In that case, the DOS system as specified on the C drive takes precedence.

COMMAND.COM

The command shell contains the standard 6.22 ROM-DISK commands and is located on the ROMDISK (A) drive.

BREAK	CALL	CD	CHDIR
CLS	COPY	CTTY	DATE
DEL	DIR	ECHO	ERASE
EXIT	FOR	GOTO	HELP
IF	MD	MKDIR	PATH
PAUSE	PROMPT	RD	REM
REN	RENAME	RMDIR	SET
SHIFT	TIME	TYPE	VER
VERIFY	VOL		

Modifications and Special Considerations

The T2090 keypad does not contain the backslash (\) and colon (:) characters, which are typically needed for many CD commands and for selecting drives.

None of these commands have been modified from standard PC DOS screen format to the T2090 screen, because given the read-only nature of information on drive A, it is assumed that the average end user would only work within drive C and have little need to run DOS commands from the prompt.

You can use any DOS command and access all the keys on a standard PC keyboard by establishing a CTTY session between your PC and the T2090. For help, see "Managing Files" in Chapter 1

CONFIG.SYS

A default CONFIG.SYS file resides on ROMDISK, which is the read-only A drive of the T2090. This is a minimal configuration file which you cannot customize. However, you can create a customized CONFIG.SYS file on the C drive.

Each time the CONFIG.SYS file on the A drive is invoked, it executes all of its command code and then calls the CONFIG.SYS file on the C drive. If the file exists, it is invoked. If not, the message "Bad or Missing NewFile *file*" appears.

Note: *"Bad or Missing NewFile* file*" also appears the first time you boot the T2090 because the computer by default does not contain a CONFIG.SYS file on the C drive.*

Similarly, the A drive AUTOEXEC.BAT file executes next and calls the AUTOEXEC.BAT file located on the C drive.

If you do not create the configuration files on the C drive, the two minimal configuration files located on the A drive, which cannot be altered, effectively boot the system. (As described in the next section, if AUTOEXEC.BAT on the A drive does not find the configuration files on the C drive, it creates a minimal version of each file there.)

AUTOEXEC.BAT

The resident AUTOEXEC.BAT file performs the following activities:

- Run the T2090 setup on detection of FWT environment variable set to ON. At boot, this variable is set to OFF or ON based on detection of the F1 key. (positive detection = ON). Like any other environmental variable, it may be altered or interrogated by batch files and programs.
- Run the T2090 setup if AUTOEXEC.BAT or CONFIG.SYS not found on C drive.
- Set the minimal path to be A:\;C:\.
- Set the prompt to include drive identifier and > symbol.
- Create a minimal CONFIG.SYS file in the root directory of C:\ if not found.
- Create a minimal AUTOEXEC.BAT file in the root directory of C:\ if not found.
- Change operational drive to C.
- Execute C:\AUTOEXEC.BAT file.

Utility Programs

The following utility programs and one help file are located on ROMDISK:

- COMMAND.HLP
- TRANSFER.EXE
- XFER.EXE
- REMSERV.EXE, REMOTE.BAT, and RT.BAT
- FWTSETUP.EXE
- CLEANUP.EXE

COMMAND.HLP

This file will be used in conjunction with the HELP command. You can use the HELP command to display syntax information for any command:

`c> help` *`command`*

TRANSFER.EXE

Provides serial transfer of files via XMODEM or YMODEM protocols into and out of the unit. The program runs on the T2090 and can be found in the PSK software toolkit in the INTERMEC\IMT209\SAMPLE directory.

Syntax

```
c> transfer {options} {filenames}
```

where *options* are:

-Q	Quiet mode. Titles, messages, and status lines are not displayed.
-M	Messages are not displayed. The title and status line may appear.
-I	Status line is not displayed. The title and messages may appear.
-B*nnn*	Baud rate, where *nnn* may be 110, 300, 600, 1200, 2400, 4800, 9600 (default), 19200, 38400, 57600, 11500, or 115 (short for 115200)
-R	Receive the file. You must specify either this option or -S.
-S	Send the file. You must specify either this option or -R.
-COM*n*	COM port, where *n* is the port number. Set *n* to 1. (Default)
-P*protocol*	Protocol, where *protocol* is XMODEM, XMODEMCRC (default), XMODEM1K, XMODEMG, YMODEM, YMODEMG, YMODEMI, or KERMIT. Keep in mind these notes about the protocols: • XMODEMCRC provides 1K block transmission with CRC and fall back to Checksum mode. • XMODEM1K is only 1K block transmission with CRC. • XMODEM is 128 byte block transmission with Checksum. • XMODEMG and YMODEMG assume an errorless connection, typically used with MNP connections. • YMODEMI is a special variant of YMODEM that allows a file name to be specified on receive to which the first received file is written.

where *filenames* are:

filenames	File names depend on the protocol and the send / receive option: • For all XMODEM types, both send and receive require a file name. • For all YMODEM types and KERMIT, the send requires at least one file name and will accept multiple names separated by a space. • For YMODEMI, a file name is optional for receiving the first file and altering its name. If the sender continues with additional files they will be stored per normal YMODEM standards.

XFER.EXE

This is a custom version of TRANSFER.EXE which waits for the T2090 to be placed in the dock. This utility should be used to initiate XMODEM transfers on the T2090. It does not run on a PC.

Syntax

```
c> xfer {options} filename
```

where *options* are:

-r	Receive file
-s	Send file
-B#	Sets the baud rate to 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, or 115200 (Default = 9600)
-COM#	Specify COM port 1 (must be 1) (Default)

REMSERV.EXE, REMOTE.BAT, and RT.BAT

Provides remote disk access to a server via the COM port. The program REMDISK.EXE must be run on the remote server for this program to function properly. This file is accessed through the batch file REMOTE.BAT, which eliminates the need for the colon (:) and slash (/) keys in the parameters. Alternatively, the RT.BAT file will start the remote, exporting the C:\ drive at 38400 bps.

Syntax

c>remote {*drive*} [B*nnnn*] [COM*n*] [T]

where:

drive	Letter designation for the T2090 drive now seen as the next logical drive to the host system.
B*nnnn*	Baud rate 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, or 115200 (Default = 115K) All baud rates above 9600 use the packet transmission(+ in remdisk) option automatically.
COM*n*	Port number 1 or 2 (Default = 1).
T	Communications Timeout value (in seconds 2 - 3640). You should always use /T10 or greater.
?	Help

Alternate Syntax

c>rt *nnnn*

where *nnnn* is the baud rate without the B. The default baud rate is 38400 or 38.4k baud.

FWTSETUP.EXE

T2090 setup program (discussed later in this chapter in "Setup Menu").

Syntax

```
c>fwtsetup
```

CLEANUP.EXE

This utility program is intended as a performance enhancer for applications that perform a large number of disk writes during their execution. To understand what this file does, it's necessary to describe how a disk drive in flash memory operates.

Flash, unlike a real disk, cannot just be written over by new data. The area to be written must first be erased. For this reason, as the flash disk is written to or as files are deleted, blocks of flash are marked as dirty (that is, used and needing erasure). When the time comes to use these blocks, valid data is copied from an entire 64K zone into a spare zone, then the dirty zone erased. The device driver itself periodically performs this cleaning operation automatically during the process of reading and writing to the C drive. This automatic cleaning typically ranges from 0.5 to 3 seconds, depending on the condition of the flash at the time.

In situations where large blocks of data are to be written to the drive, or latency times for writes are important, performance can be improved by first cleaning the disk manually. CLEANUP.EXE looks at the entire drive and collects all dirty blocks, making them available for writes. This delays the need to automatically perform a cleaning function for a significant period of time.

Syntax

```
CLEANUP 80 /S     (from the command line)

CLEANUP C:./S     (from a batch file or CTT mode)
```

where:

<80>	specifies drive C for the T2090.
/S	prevents any nonfatal messages from appearing on the screen (Silent mode).

EMS Driver

An Expanded Memory Manager is included with the T2090 Programmer's Software Kit and conforms to LIM EMS 4.0. This driver places all RAM above 640K into Expanded Memory available to applications through INT 67. In the T2090, with a 2MB RAM chip installed, the amount of EMS RAM available is:

2048K - 64K(BIOS shadow) - 640K(DOS area) = 1344K

To install the driver, the developer needs to place the driver file, VG330EMS.SYS on the Read / Write drive of the T2090 (drive C). This is usually done at the root directory. Then the line shown below must be added to the C:\CONFIG.SYS file:

```
DEVICE=C:\VG330EMS.SYS
```

It is important within this device installation line to include the full drive and path because during system boot, the C drive is not part of the initial path; only the A drive is part of the assumed path.

RAMDRIVE Driver

A RAMDRIVE system driver is included with the T2090 to allow the developer to use RAM as a virtual drive. This driver will, by default, use Expanded Memory if available. If not, then RAM will be taken from the lower 640K DOS area.

To install the driver, the developer needs to place the driver file, MEMEMS.SYS on the Read / Write drive of the T2090 (C:). This is usually done at the root directory. Then the following line must be added to the C:\CONFIG.SYS file:

DEVICE=C:\MEMEMS.SYS [*disk size* [*sector size* [*root entries*]]]

where:

disk size	is the size of the desired ramdisk in kbytes (min. 16, default = 64)
sector size	is the desired sector size (128, 256, 512 or 1024) in bytes (default = 512)
root entries	is the number of root directory entries from 2 to 1024 (default = 64)

It is important within this device installation line to include the full drive and path because during system boot, the C drive is not part of the initial path; only the A drive is part of the assumed path.

Docking Indicators

The unit has visual feedback on the state of being placed in the EZ Dock Docking System. When the unit is initially placed in the docking system, it beeps and the LED turns RED. After five seconds, the status of the charging cycle is checked. If the system is actively charging, it blinks red off/on with a duty cycle of four seconds (two seconds off, two seconds on) for a minimum of 15 minutes.

Once the battery is fully charged, the LED turns green and is on in a steady state. When the unit is removed from the docking system, it beeps and turns off the LED. The unit can be removed at any point during the charging cycle. This indicator turns red, then green, within five seconds and remains green if no battery is installed.

Battery Monitor

Applications can monitor the state of the battery by using the software Int 15h discussed in Chapter 7. A low battery condition occurs when the battery voltage is 2.2v or less. If a very low battery condition occurs (2.0 v or less), the unit enters SUSPEND state (see Chapter 6 for handling very low battery conditions). Also, under cold boot conditions, if a low battery is detected, the unit will power off without booting.

Keyboard Mode Indicator

An alpha/numeric keyboard mode indicator can be called by applications or displayed on the screen at all times. Displaying the indicator is selectable as an option under the Setup routine (discussed later in this chapter) or is callable by applications through the Int 10h mechanism. The indicator is a 16 x 8 pixel area mapped out to read either "123" for numeric mode or "ABC" for alpha mode. The default condition is numeric.

Pressing the ⇧ key toggles between the two modes and changes the indicator on the screen.

Developers should note that if the keyboard mode indicator is displayed, the small area of the screen being used by the indicator is protected and is not available for displaying other characters. (See Chapter 7.)

System Defaults and Initial Conditions

Power States

There are five power states:

- ON
- DOZE
- SLEEP
- SUSPEND
- OFF

SUSPEND and OFF are mutually exclusive and can be set by the developer or the user. The ⏻ key can be used at any time in the ON, DOZE or SLEEP state to enter either SUSPEND or OFF, depending on which is selected, and pressing the ⏻ key when in SUSPEND or OFF enters the ON state.

The figure on the next page is a state diagram for the automatic power saving functions.

Automatic states are reached sequentially on a decreasing power consumption curve from the first state, ON (CPU at full speed, all systems powered up). The second state is DOZE (CPU slows to 25% of normal speed). The third state is SLEEP, (backlight shuts off). The last state is SUSPEND/OFF (everything powered down, except RTC; if SUSPEND is set, power is retained for RAM).

***Note**: If any one of the powered-down states time parameters are set to zero, it has the effect of canceling that particular state as an option and all the other states below it. That is, if SLEEP is set to zero, the unit will never get to the SUSPEND state, the backlight will never shut off and there won't be any auto timeout, and if DOZE is set to 0, NO power management occurs.*

Activities are defined as keypress, disk access, serial communications and the default video access.

DOZE times can be set in 1/8 second increments, up to 2 seconds, then in one second increments up to 14 seconds. SLEEP and SUSPEND/OFF can be set from 0 to 15 minutes.

Current defaults from the factory are:

To enter DOZE from ON:	1/2 second
To enter SLEEP from DOZE:	1 minute
To enter SUSPEND/OFF from SLEEP:	1 minute

PREVIOUS STATE(S)	ENTRY CAUSE(S)	CURRENT STATE	EXIT CAUSE(S)	NEXT STATE
N/A	HARD RESET	ON	NO ACTIVITY	DOZE
SUSPEND/ OFF	⏻ KEY		⏻ KEY	SUSPEND/ OFF
DOZE OR SLEEP	ACTIVITY			
ON	NO ACTIVITY	DOZE	ACTIVITY	ON
			⏻ KEY	SUSPEND/ OFF
DOZE	NO ACTIVITY	SLEEP	ACTIVITY	ON
			⏻ KEY	SUSPEND/ OFF
SLEEP	NO ACTIVITY	SUSPEND/ OFF	⏻ KEY	ON
ON, DOZE, SLEEP	⏻ KEY		RTC ALARM	ON
			UNIT DOCKED	ON

ACTIVITY = one of the keys was pressed, disk was accessed, serial communications were attempted.

NO ACTIVITY = none of above occurred within a preset timeframe.

Timeouts are set through Setup as discussed later in this chapter.

Starting Setup Mode

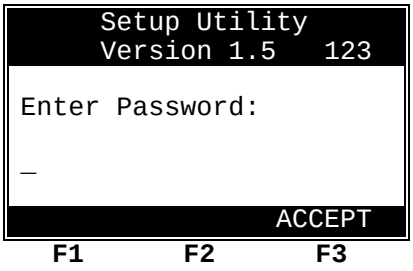
The setup mode is entered when the user presses the F1 (left key under the display) key after pressing the ⏻ key or by typing FWTSETUP at the DOS prompt. The Enter Password screen is then displayed.

Note: *Examples of screens illustrated in the remainder of this chapter have rows and columns numbered to show actual placement on the display. All of the data entry screens have the 'BS' destructive backspace key available. Except for 'BS', keys not specified as having any action are ignored.*

Enter Password Screen

The keys and actions for the Enter Password screen are as follows:

- 0 - 9, A - Z, the dash mark (-), the space key, and the period (.) are all valid for the password
- F3 "ACCEPT" or **ENTER** keys accepts the value entered for verification. The default factory password is "123."



As the password is entered, an asterisk is displayed for each character entered. If the user fails to enter the correct password after three tries, Setup access is denied.

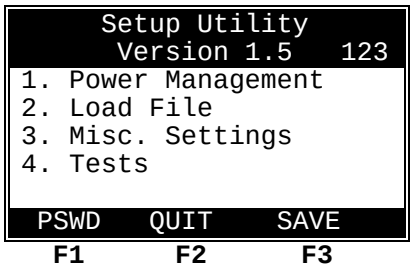
If a correct password is entered, Setup access is allowed and the Setup Main Menu is displayed.

Note*: Password data is stored in the C: drive in a hidden file, IMP.DAT. If this file is corrupted or deleted, it will be re-written with the default password ("123") on the next boot.*

Setup Main Menu

The Setup main menu screen has four menu items and three valid function keys. The keys and actions for the Setup main menu are as follows:

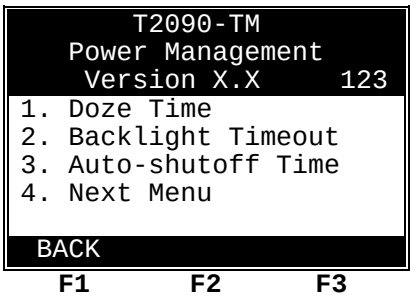
- **1** calls up the Power Management screen
- **2** enters calls up the Load File screen
- **3** calls up the Misc. Settings screen
- **4** calls up the Tests screen
- **F1** "PSWD" key calls up the Password Setting screen
- **F2** "QUIT" key exits without saving
- **F3** "SAVE" key exits and saves the new selections



Power Management Screen

This screen is a group of four menu items for setting the power management parameters. The keys and actions for the Power Management screen are as follows:

- **1** calls up the Doze Time screen
- **2** calls up the Backlight Timeout screen
- **3** calls up the Auto-shutoff Time screen
- **4** calls up the Power Management II screen
- **F1** "BACK" returns to the main menu



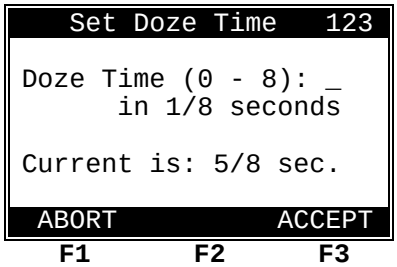
Doze time can be set from 0 to 15 seconds. Settings 1 through 8 are in 1/8-second increments, while settings 9 to 15 are full, one-second increments. The valid range time for both backlight timeout and auto-shutoff time is from 1 to 15 minutes. Item 4, Next Menu, takes you to a second power management screen for configuring the ⏻ key and choosing a preferred shut-down mode. See Chapter 7 for more information on Power Management.

***Note**: Timeout periods are sequential; that is, the doze countdown timer must trigger before the countdown begins for backlight (SLEEP mode), and the backlight countdown timer must trigger before the auto-shutoff countdown timer begins. If you set doze to 10 seconds, backlight to one minute, and auto-shutoff to two minutes, the auto-shutoff will occur (from the ON state) after three minutes, 10 seconds.*

Set Doze Time Screen

The keys and actions for the Set Doze Time, Backlight, and Auto-Shutoff Time screens are as follows:

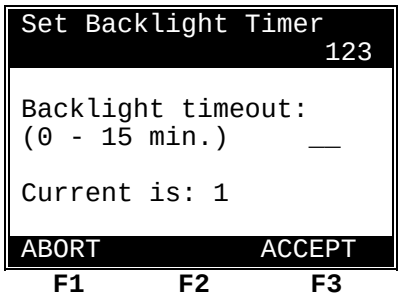
- **0 - 8** enters the time.
- **F1** "ABORT" key returns to the main menu without saving new values.
- **F3** "ACCEPT" or **ENTER** keys accepts new values and returns to the main menu.



Backlight Timeout Screen

Keys and actions for the Set Backlight Time screen are as follows:

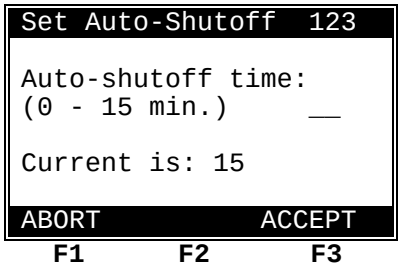
- **0 - 9** enters the time. (Valid entries are 0 through 15.) Where you see the underline in the screen is where the entry is displayed.
- **F1** "ABORT" key returns to the main menu without saving new values.
- **F3** "ACCEPT" or **ENTER** keys accepts new values and returns to the main menu.



Set Auto-Shutoff Screen

Keys and actions for the Set Auto Shutoff Time are as follows:

- **0 - 9** enters the time. (Valid entries are 0 through 15.) Where you see the underline in the figure is where the entry is displayed.
- **F1** "ABORT" key returns to the main menu without saving new values.
- **F3** "ACCEPT" or **ENTER** keys accepts new values and returns to the main menu.

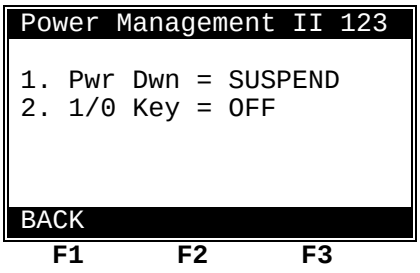


Power Management II Screen

This screen is used to set the auto shutoff mode and the ⏻ key mode. The ⏻ key can be configured to either turn the unit off or activate the power down mode.

Keys and actions on the Power Management II screen are as follows:

- **1** sets the Power Down function toggles between SUSPEND & OFF
- **2** sets the ⏻ key function and toggles between OFF and Power Down. The ⏻ key can be configured to enter and exit from full ON or OFF or to enter and exit reduced power states. Types of configurations and possible choices are detailed further in Chapter 7.
- **F1** "BACK" returns to the main menu.

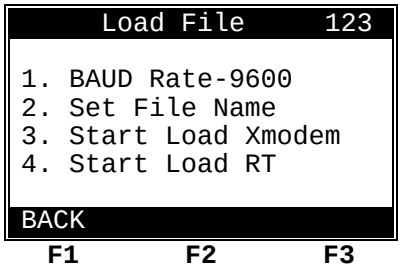


Load File Screen

This screen is used to load files onto the T2090's C drive using TRANSFER, an XMODEM file transfer utility. The user can specify the baud rate.

Keys and actions for the Load File screen are as follows:

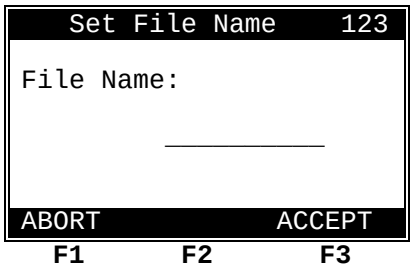
- **1** scrolls different baud rates: 9600, 19200, 38400, 57600 and 115200 .
- **2** enters the "Set File Name" Screen (see below).
- **3** begins receiving file using an Xmodem protocol by spawning XFER.EXE.
- **4** begins file transfer by spawning RT.BAT.
- **F1** "BACK" key returns to the main menu.



Set File Name Screen

Keys and actions for the Set File Name screen are as follows:

- Keys **1** - **0**, A - Z and the period (.) are used to enter an eight-character file name with file extension.
- **F1** "ABORT" key returns to the main menu without saving the name.
- **F3** "ACCEPT" or **ENTER** keys saves the name and returns to the main menu.

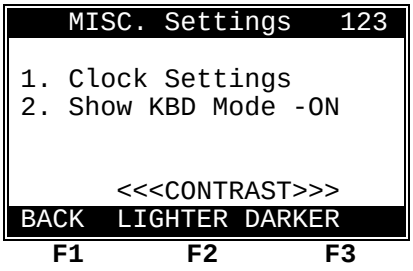


Miscellaneous Settings Screen

This screen contains three settings: clock, show keyboard mode, and display contrast. Exiting this screen sets the display contrast whether or not the settings are saved in the main screen.

Keys and their actions for MISC. Settings Screen:

- **1** "Clock Settings" enters "Set Clock" screen (see below).
- **2** toggles between displaying the keyboard mode indicator in the top left corner of the display or not.
- **F1** "BACK" key returns to the main menu.
- **F2** "LIGHTER" lightens the display contrast.
- **F3** "DARKER" darkens the display contrast.

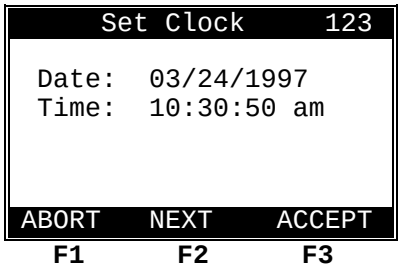


Set Clock Screen

The Set Clock Screen has seven entry fields: month, day, year, hour, minutes, seconds and am / pm. Entering new data in a field overwrites the previous data and when the field is full, it automatically tabs to the next field, and returns to the first field when all data has been entered. If illegal data is entered, an error message is displayed and the cursor is placed on the first illegal field. The year value has to be in the range 1980 - 2099.

Keys and actions for the Set Clock screen are as follows:

- **0 -9** for all fields.
- **F1** "ABORT" key returns to the main menu without saving new values
- **F2** "NEXT" key tabs to the next field.
- **F3** "ACCEPT" or **ENTER** keys sets new values then returns to the main menu



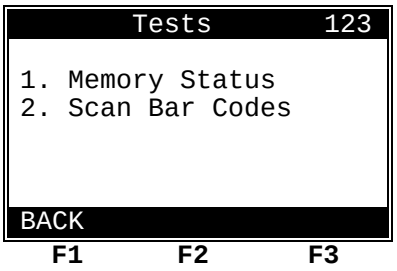
Note: If the clock has not been set or has been reset by a cold boot, the screen will appear following boot.

Tests Screen

The tests screen has two selections: display memory status and test bar code scanner.

Keys and actions for the Tests screen are as follows:

- **1** enters the "Memory Status" screen.
- **2** enters the Scan Bar Codes screen (see below).
- **F1** "BACK" key returns to the main menu.

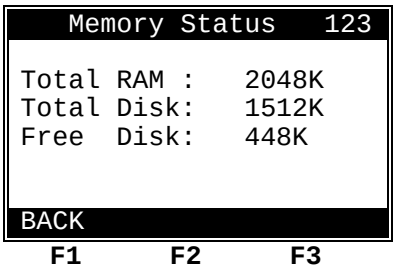


Memory Status Screen

This screen displays the current status of the device's memory.

Keys and actions for the Memory Status screen are as follows:

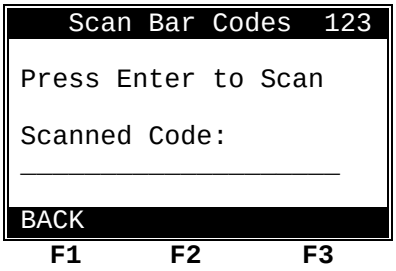
- **F1** "BACK" returns to the main menu.



Test Scanner Screen

Keys and actions for the Test Scanner screen are as follows:

- ENTER starts a bar code scan. The front panel LED turns RED. Once the scan has been successfully completed, the LED turns GREEN briefly and the data scanned is displayed where the underscore is shown in the figure.
- **F1** "BACK" key returns to the main menu.

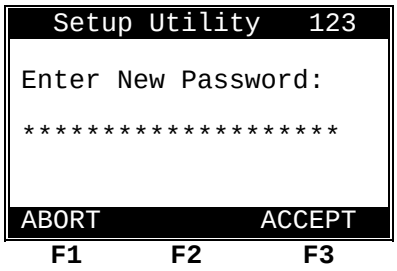


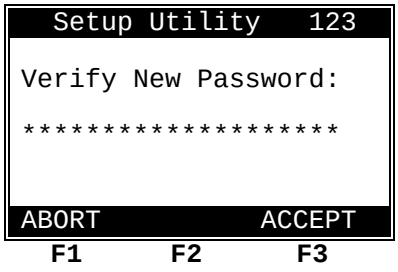
Password Setting Screens

The Password Setting screen (available from pressing **F1** on the Setup Utility screen) lets you set a password to enter the Setup Menu. The password can be up to twenty characters long. The default password is "123."

Keys and actions for both the Enter New Password and the Verify New Password screens are as follows:

- **0 -9**, **A**-**Z**, dash (**-**), **comma (,), apostrophe (‘), space, and period (.)** are all valid for the password.
- **F3** or **ENTER** enters the password.





3

Application Development

This chapter describes the system architecture used in the TRAKKER 2090 and provides basic information about developing applications using the Intermec-supplied library of functions.

System Architecture

The TRAKKER T2090 hardware system has been designed with three factors in mind:

- Maintain a high degree of DOS compatibility.
- Allow application designers as much flexibility as possible in system configuration and application design.
- Provide simple boot-up capability even if user programs cause a temporary system lockup.

At the lowest level of the system is an enhanced 80186 BIOS. This provides the hardware/software interface for all operating system functions. The BIOS has been adapted to support unique screen, keyboard, and communications features of the T2090. In all cases, the link to higher level code is made through software interrupts, thereby eliminating restrictions upon code development languages.

At the BIOS level, through extensions, a device driver is incorporated for creation of the user's drive C within the FLASH chips. Incorporation of the driver at this level ensures that DOS, when booted, can recognize the drive and, if desired, utilize system files from this drive for booting. As an additional extension, we have provided a fail-safe mechanism for boot protection.

Memory

Like other PCs, the TRAKKER T2090 has only 1MB, (1,048,576 bytes) of real addressable space. Above this limit, all memory must be paged into some other address. In the T2090, the lower 640K RAM is available to DOS, less the system drivers loaded prior to DOS boot. The remainder of the real address space is used for the mapping of EMS (extended memory system) RAM, FLASH drives and the BIOS.

The following table summarizes the availability of addressable memory.

Address Range	Use
FFFF:Fh F000:0h	BIOS
EFFF:Fh E000:0h	Device Driver Mapping
DFFF:Fh D000:0h	Unavailable
CFFF:Fh C000:0h	Drive A and drive C mapping
BFFF:Fh B000:0h	Unavailable
AFFF:Fh A000:0h	EMS Mapping
9FFF:Fh 0000:0h	Application RAM

Development Environment

The TRAKKER T2090 has a DOS 6.22 compatible operating system. Contained within the T2090 are utilities to assist in application development. Additionally, a development library exists for C language applications.

QuickStart Kit Contents

In most cases this manual has been distributed as part of a released QuickStart Kit, which consists of a TRAKKER T2090 unit, the EZ Dock Docking System and Universal Autoranging Power Supply, the appropriate serial cable to connect the Power Module to a host PC development system, PSK software toolkit, and a long-life nickel hydride battery.

The software, supplied on either diskette or CD-ROM, consists of a C language software library for Microsoft C/C++ version 1.52, printable text files for this manual, demonstration program, and C language source code.

Compiler Requirements

The software libraries distributed in support of the TRAKKER T2090 and this development kit were developed on Windows 3.11 and Windows 95, Pentium based systems in conjunction with the Microsoft Visual C/C++ version 1.52 compiler.

Refer to Appendix B for correct settings and options.

Device Communications and Connectivity

The TRAKKER T2090 has a single serial port for communications. The port can be configured as RS-232 via a program function call. Connection to the serial port can be found underneath the power module.

Two female RJ-45 connectors reside on the power module. An RJ-45 to DB9 RS-232 serial cable can be connected from the 2-slot cradle to a host computer. Intermec supplies a DOS utility called REMDISK that runs under MS-DOS 6.22, and can communicate with the T2090 via the power module. On the T2090, a utility called REMSERV can be run to communicate with the REMDISK program on the host DOS PC. This allows the user to remotely access drive C of the T2090 on the host DOS PC. Drive C will become the next available drive on the host DOS PC. If the last drive on the host DOS PC is D, the T2090 drive is now E.

To run REMSERV on the T2090 and REMDISK on the PC, you can run the batch file RT on the T2090. RT is built in to the T2090, so try running it. You should get the following message:

```
Initializing RS232
RS232 Enabled!


Press ENTER to quit.
```

With the T2090 sitting in the dock and REMSERV running via the RT batch file, copy REMDISK from the Intermec CD onto the host PC. On the PC, boot your computer into DOS. Type REMDISK COM1 /B38400+ on the PC to run the REMDISK program at 38400 BPS if you are connected to the PC on COM1. If you are connected to COM2, then replace COM1 with COM2 in the REMDISK command.

To uninstall REMDISK on the PC, type REMDISK /U. The RT batch file sets up REMSERV on the T2090 to run at 38400 BPS. When the REMDISK utility is invoked, it will identify the drive letter which will be used to access the T2090. It uses the next logical drive on the PC; that is, if last used drive on your PC is D, then the T2090 drive is E. Select drive E, then type DIR <ENTER>.

The T2090 should have a CONFIG.SYS and an AUTOEXEC.BAT file on it. You can now alter these as necessary, and you can now copy files to the T2090 by copying from your PC drives to the T2090 drive.

***Note**: The scanner in the T2090 cannot be used simultaneously with the serial port or when the unit is in the dock, because the internal UART is shared between the scanner and the communications serial port.*

Drive A of the T2090 contains MS-DOS and other utilities. You can access this by going to the C> prompt on the T2090, and then typing CTTY COM1 on the T2090. On a host DOS PC, go into a terminal emulation program. CTTY uses the current baud rate on the T2090. If it is the first program to use the serial port after boot up, then the T2090 serial port will be set to 2400 baud (DOS default). If the RT batch file has been run, then the baud rate will be 38400, and so on. Set the serial communications port parameters to be the COM port you are connected to on the PC to the baud rate of the T2090, 8 data bits, 1 stop bit, and no parity. Press the ENTER key on the host DOS PC, and you should see C> on the host PC. Type A: <ENTER>, and you should now have the A> prompt. To see a list of the files, run the DOS DIR command.

You can exit the remote access CTTY COM1 mode by typing CTTY CON on the PC.

***Note**: Drive A is not writable. It is reserved for the preloaded T2090 utilities only.*

Application Library Support

Intermec supplies a library (IMT209ML.LIB) to support application development for Microsoft compilers. The library functions allow the programmer to access all unique capabilities of the T2090 using the C programming language. These include serial communications, scanning, keyboard processing, sound capabilities, and power management.

Link your Microsoft compiler with the library file, IMT209ML.LIB.

4

Application Functions

This section describes the functions contained in the library, IMT209ML.LIB.

Introduction

The following example (function_name) explains the parts of the function descriptions.

function_name

Purpose Briefly describes the function and its typical use.

Syntax Lists the C-language function prototype and the required include file.

IN Parameters Describes the input parameters (arguments) for the function and lists acceptable values. Not all functions have input parameters.

OUT Parameters Describes the output parameters (arguments) for the function and lists acceptable values. Not all functions have output parameters.

IN/OUT Parameters Describes the parameters (arguments) for the function that are passed into the function and back out of the function and lists acceptable values. The function usually changes the value before returning. Not all functions have in/out parameters.

Return Value Describes the value returned by the function and lists acceptable values. Not all functions have a return value.

Notes Describes any additional requirements for using the function. Not all functions have notes.

See Also Lists similar PSK functions.

Example Contains a code that illustrates how to use the function.

Note: *The PSK requires Microsoft Visual C/C++, Professional Edition v1.0 or v1.5x, which can create 16-bit DOS applications. See Appendix B for more information.*

Application Functions Listed by Category

The following table groups the application functions described in this chapter.

Communications	Display
im_receive_buffer	im_clear_screen
im_receive_field	im_cputs
im_receive_input	im_dbyte_setfont
im_set_com_mode	im_dbyte_symbology_set
im_set_port_direction	im_draw_underline
im_transmit_buffer	im_get_cursor_style
im_xm_receive_file	im_get_cursor_xy
im_xm1k_receive_file	im_get_display_mode
im_xm_transmit_file	im_get_display_size_physical
im_xm1k_transmit_file	im_get_display_type
im_ym_receive_file	im_message
im_ym_transmit_file	im_putchar
	im_puts
	im_set_cursor_style
	im_set_cursor_xy
	im_set_display_mode
	im_status_line
	im_timed_status_line

Double Byte Character Set

im_putchar_dbyte

im_puts_dbyte

im_offset_dbyte

Input

im_get_input_mode

im_get_label_symbology

im_get_label_symbologyid

im_get_length

im_input_status

im_irl_a

im_irl_k

im_irl_n

im_irl_v

im_irl_y

im_receive_buffer

im_receive_field

im_receive_input

im_set_2of5_lengths

im_set_input_mode

Program Control

im_set_scanning

im_standby_wait

System

im_command

im_get_config_info

im_get_kbmode

im_get_system_julian_date

im_message

im_set_kbmode

im_sound

Timer Callback

im_tm_callback_register

im_tm_callback_cancel

im_clear_screen

Purpose This function erases the entire display and moves the cursor to the upper left corner (home). The display font remains the same. The line attributes are all set to IM_NORMAL.

Syntax

```
#include "imt209x.h"
void im_clear_screen
    (void);
```

IN Parameters None.

OUT Parameters None.

Return Value None.

im_command

Purpose This function sends reader commands to the computer. For example, you can use this function to set the backlight contrast or change the baud rate on the computer . For help, see “Using Reader Commands” and “Using Configuration Commands” Chapters 5 and 6.

Syntax

```
#include "imt209x.h"
IM_STATUS im_command
    (IM_UCHAR far *command,
    IM_USHORT command_length);
```

IN Parameters *command* Computer command string. The command string may include more than one command.

command_length Length of the computer command string.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_SUCCESS Successfully parsed and implemented command.

IM_PARSE_ERROR Unable to parse command.

Notes If you are using ASCII escape sequences in your command string, the hex value for the escape sequence only counts as one character when designating the command length. For example, the following command string uses the ASCII escape sequence ETX (hex value 0x03) and has a character length of 5:

```
im_command ("$+PF\x03",5);
```

For help using im_command, see Chapters 5 and 6.

Example To transmit DATA.TXT on the T2090's drive C, the function would be:

```
im_command ("%%X1,C:DATA.TXT",15);
```

To receive a file through the serial port, use a command string of the form, .%X1,drive:filename. For example, to receive ITEMS.TXT on the T2090 and put the file on drive C, the function would be:

```
im_command (".%X1,C:ITEMS.TXT",16);
```

To change the configuration, a string of the form $+CCP, where CC represents the command as described in Chapter 6, and the P represents some string of parameters to the configuration command. For example:

```
im_command("$+IA8",5); //Set Baud Rate to 38400
im_command("$+PE\x02",5); //Set SOM to ASCII STX character
im_command("$+PF\x0d\x0a",6); //Set EOM to Carriage Return, Line Feed
```

im_cputs

Purpose This function displays a string on the terminal screen at the current cursor location without appending a carriage return and line feed (CR LF) to the string.

Syntax

```
#include "imt209x.h"
IM_STATUS far im_cputs
    (IM_UCHAR far *string,
    IM_ATTRIBUTES attrib) ;
```

IN Parameters *string* Far pointer to the string to be displayed.

attrib Identifies the display attribute for the string. Choose one of these constants:

IM_NORMAL Plain text.

IM_INVERSE Inverse color text.

IM_UNDERLINE Underline text.

IM_UNCHANGED Leave the attribute unchanged.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_OK Success.

IM_BAD_ADDRESS The value for *string* is invalid.

IM_INVALID_PARAM_2 The value for *attrib* is invalid.

Notes This function is similar to im_puts except that it does not automatically append a carriage return and line feed (CR LF) to the string.

On the TRAKKER T2090 computer, displaying a character with the IM_INVERSE attribute causes the entire line to be inverted.

See Also im_puts

im_dbyte_setfont

Purpose This function initializes the .DBFS font specified. Any current DBFS font is disconnected from the display, and the display is cleared. If switching from "native" font display, the ABC / 123 indicator and native mode cursor (if active) are disabled and the screen cleared. The altered fonts are not persistent between applications.

Syntax

```
#include "imt209x.h"
IM_STATUS im_dbyte_setfont
    (IM_UCHAR *fontfile,
    IM_USHORT glyph_high,
    IM_USHORT glyph_wide,
    IM_USHORT offset);
```

IN Parameters *fontfile* Pointer to the name of the font file to initialize. A value of 0 causes a return to native mode and restoration of the ABC / 123 indicator and native cursor to the states they had prior to the first DBFS im_dbyte_setfont() in this application.

glyph_high Height of the character cell in pixels (ignored when going to native mode). Values of 1 to 72 are legal, but the 72nd row of the display is not visible.

glyph_wide Width of the character cell in pixels (ignored when going to native mode). The value must be an integral multiple of 8 between 8 and 120. The 120th column of the display is not visible.

offset Index of the first glyph in the fontfile (in glyphs).

OUT Parameters None.

Return Value This function returns one of these codes:

IM_SUCCESS Font installed.

IM_INVALID_FILE The file open failed. The font file could not be found.

IM_INVALID_PARAM_2 Specified height falls outside the acceptable limits.

IM_INVALID_PARAM_3 Specified width falls outside the acceptable limits or is not a multiple of 8.

Notes The font file is a series of bitmaps. Each bitmap is glyph_wide/8 bytes per row by glyph_high rows. The most significant bit maps to the leftmost pixel. The rows build from the top down. An offset into the file indicates the index of the glyph whose bitmap starts at position 0 in the file. (Copyright information can be inserted at the beginning of the file, so long as it occupies an integral number of glyph spaces and the offset is adjusted accordingly.) Bottom and right side bearings should be built into the font except where successive characters should make contact. If the font is of width of 8, 24, or 120, allowance should be made for the fact that the rightmost pixel of the last column will not be visible since only 119 columns of the display are active. If the font is of height 1, 2, 3, 4, 6, 8, 9, 12, 18,24,36 or 72, allowance should be made for the fact that the bottom row of pixels for the last row of characters will not be visible since only 71 rows of the display are active. Since this is the row in which the cursor is displayed, it may be more important than the width restriction.

The following functions are aware of DBFS fonts and adjust their actions when DBFS is active:

- im_set_cursor_xy and im_get_cursor_xy use the DBFS character cell size and adjust maximum positions to match.
- im_clear_screen sets the cursor to 0,0 in DBFS font space
- im_set_cursor_style and im_get_cursor_style set and report the style for the DBFS cursor, which inherits the native cursor mode at initialization, but does not pass altered modes back when DBFS is terminated.
- im_putchar_dbyte and im_puts_dbyte place glyphs from the current font at the current position. (They return IM_INVALID_FUNCTION and have no display effect if DBFS is not active.)
- im_offset_dbyte allows a remapping of the font.
- im_puts and im_putchar do no plotting while DBFS is active. This has the intended effect that im_receive_field and im_receive_input also do no character plotting, such as clear fields or echo data.

No other functions are aware of DBFS actions. Functions which change the display using native mode will cause unpredictable effects which may change over a suspend/resume cycle and are therefore not usually recommended.

See Also im_putchar_dbyte, im_puts_dbyte, im_offset_dbyte

im_dbyte_symbology_set

Purpose This function indicates that DBFS is currently active.

Syntax

```
#include "imt209x.h"
IM_BOOL im_dbyte_symbology_set
    ( void );
```

IN Parameters None.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_TRUE DBFS Font installed.

IM_FALSE DBFS Font not installed.

See Also im_dbyte_setfont

im_draw_underline

Purpose This function draws an underline on the screen between two points.

Syntax

```
#include "imt209x.h"
void im_draw_underline
    (IM_USHORT start_row,
    IM_USHORT start_column,
    IM_USHORT end_row,
    IM_USHORT end_column,
    IM_USHORT underline);
```

IN Parameters

start_row Specifies the number of the row where the underline starts.

start_column Specifies the number of the column where the underline starts.

end_row Specifies the number of the row where the underline ends.

end_column Specifies the number of the column one position after the column where the underline ends.

underline Choose one of these constants:

IM_SET_UNDERLINE to draw the underline.

IM_CLEAR_UNDERLINE to erase the underline.

OUT Parameters None.

Return Value None.

Notes The *end_column* parameter specifies one column beyond the underline to make it easy for you to use im_get_cursor_xy to determine the values for *start_row*, *start_column*, *end_row*, and *end_column*. Follow these steps:

1. Use im_get_cursor_xy to determine the cursor position, passing *start_row* and *start_column* variables.
2. Display the text.
3. Use im_get_cursor_xy again to determine the cursor position (which is one position beyond the end of the text displayed), passing *end_row* and *end_column* variables.
4. Use im_draw underline to draw an underline under the text, passing *start_row*, *start_column* , *end_row*, and *end_column* variables.

When you underline text, you should display the text first and then draw the underline with this function.

If you use a function like im_puts, which lets you specify the underline attribute for the text being put, you do not have to use im_draw_underline.

im_get_config_info

Purpose This function retrieves the current computer configuration information string and its length. The command code is passed in as a string, and the current configuration is returned in the same string.

Syntax

```
#include "imt209x.h"
IM_STATUS im_get_config_info
    (IM_UCHAR far *config,
    IM_USHORT far *length);
```

IN Parameters None.

IN/OUT Parameters *config* As input, this parameter is the desired computer command (two characters) and should be NULL terminated. You can pass in several command codes at one time. As output, this parameter contains the requested configuration information string. The first two characters specify the configuration command returned. Any subsequent characters specify the configuration options currently set. For example, to get the beep duration setting, set *config* to "BD". The function returns BD and the current configuration for beep duration. The user of this function must ensure that this character pointer points to a block of memory large enough to fit the returned NULL terminated configuration information string.

OUT Parameters *length* Length of the configuration information string.

Return Value This function returns one of these codes:

IM_SUCCESS Successfully parsed and returned configuration info string.

IM_UNKNOWN_CONFIG Unable to parse configuration request string.

IM_ERROR_TOO_BIG_FOR_BUFFER Returned for "DC" configuration request string which is unsupported due to the large amount of information that this configuration request string would require.

Notes For a list of the configuration commands, see Chapter 6.

This function differs from im_command in that you only pass the two-character command identifier. The im_command function passes an entire command string.

See Also im_command

im_get_cursor_style

Purpose This function returns the style used to display the cursor. It is DBFS aware.

Syntax
```
#include "imt209x.h"
IM_CURS_TYPE im_get_cursor_style
    (void);
```

IN Parameters None.

OUT Parameters None.

Return Value This function returns a flag indicating cursor style:

IM_UNDERLINE Single underline.

IM_NO_CURSOR No cursor displayed.

im_get_cursor_xy

Purpose This function retrieves the current cursor position. It is DBFS aware.

Syntax
```
#include "imt209x.h"
IM_STATUS im_get_cursor_xy
    (IM_USHORT far *row,
    IM_USHORT far *col);
```

IN Parameters None.

OUT Parameters *row* Pointer to the vertical position. The top of the display is row 0. The bottom of the display is row 7 (when using standard font).

col Pointer to the horizontal position. The left edge of the display is column 0.

Return Value IM_SUCCESS Success

im_get_display_mode

Purpose This function returns the display font, character height and width, and scrolling and wrapping status.

Syntax

```
#include "imt209x.h"
IM_STATUS im_get_display_mode
    (IM_FONT_TYPE far *font,
    IM_UCHAR far *phys_width,
    IM_UCHAR far *phys_height,
    IM_BOOL far *scroll,
    IM_BOOL far *wrap);
```

IN Parameters None.

OUT Parameters *font* Specifies the font type and is one of these constants:

IM_FONT_STANDARD Text is 5 x 7 pixels.

IM_FONT_LARGE Text is 5 x 14 pixels.

IM_FONT_SPECIAL Text is 10 x 14 pixels.

phys_width Specifies the width of the physical display in the number of characters in the current font that the display can hold.

phys_height Specifies the height of the physical display in the number of characters in the current font that the display can hold.

scroll Because the TRAKKER T2090 computer always scrolls at the bottom of the screen, this will always be returned as positive.

wrap Because the TRAKKER T2090 computer always wraps at the right edge of the screen, this will always be returned as positive.

Return Value IM_SUCCESS Success.

Notes To omit a parameter, set it to 0. No information for that parameter is returned.

See Also im_set_display_mode

im_get_display_size_physical

Purpose This function returns the current display size.

Syntax
```
#include "imt209x.h"
void far im_get_display_size_physical
    (IM_USHORT far *rows,
    IM_USHORT far *cols);
```

IN Parameters None.

OUT Parameters *rows* Current setting for the number of rows in the physical display.

cols Current setting for the number of columns in the physical display.

Return Value IM_SUCCESS

Notes The default physical display for the T2090 is 20 columns by 8 rows.

See Also im_get_display_mode, im_get_display_type, im_set_display_mode

im_get_display_type

Purpose This function gets the hardware display type for the TRAKKER T2090 computer.

Syntax
```
#include "imt209x.h"
IM_STATUS im_get_display_type
    (IM_DISPLAY_TYPE *type);
```

IN Parameters None.

OUT Parameters *type* The constant returned is IM_LCD_20X8 (TRAKKER T2090 display).

Return Value IM_SUCCESS Success.

See Also im_get_display_mode, im_set_display_mode, im_get_display_size_physical

im_get_input_mode

Purpose This function provides compatibility with the JANUS PSK functions. This function retrieves the current input mode setting. Input modes affect how the reader interprets and stores input.

Syntax
```
#include "imt209x.h"
IM_STATUS im_get_input_mode ();
```

IN Parameters None.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_PROGRAMMER Input is returned as a string (default). Simple line editing is permitted using the backspace key.

IM_WEDGE Input is returned as a string. Use Backspace for simple line editing.

IM_DESKTOP Keyboard characters are returned as 4 bytes. The first byte is the ASCII code. The second byte is the scan code, and the last 2 bytes are flags for modifier keys (Shift, Control, and Alt). For label input, the entire string is returned.

See Also im_set_input_mode, im_receive_input

im_get_kbmode

Purpose This function identifies the current keyboard mode.

Syntax
```
#include "imt209x.h"
int IM_GET_KBMODE(void);
```

IN Parameters None.

OUT Parameters None.

Return Value KEYBOARD_MODE_ALPHA The keyboard is in alphabetic mode.

KEYBOARD_MODE_NUMBER The keyboard is in numeric mode.

See Also im_set_kbmode

im_get_label_symbology

Purpose This function gets the symbology, such as Code 39, from the most recently scanned label. Call this function after receiving data using im_receive_input or im_receive_field.

Syntax

```
#include "imt209x.h"
IM_STATUS im_get_label_symbology
    (IM_DECTYPE far *symb);
```

IN Parameters None.

OUT Parameters *symb* Identifies the label symbology and is one of these constants:

IM_UNKNOWN_DECODE Unknown bar code.

IM_CODABAR Codabar bar code.

IM_CODE_39 Code 39 bar code.

IM_CODE_128 Code 128 bar code.

IM_I_2_OF_5 Interleaved 2 of 5.

IM_MSI MSI bar code.

IM_UPC Universal Product code.

Return Value This function returns one of these codes:

IM_SUCCESS Successfully retrieved.

IM_NO_SYMBOLOGY No symbology code available, or no scans received.

See Also im_receive_input, im_receive_field, im_get_label_symbologyid

Example

```
/********************* im_get_label_symbology *********************/
#include <conio.h>
#include "stdio.h"
#include "im209x.h"
static char *bar_code[] = {
  "unknown",
  "Code 39",      /* See typedef enum { ...} IM_DECTYPE in Im209x.h*/
  "I 2 of 5",
  "Codabar",
  "UPC and EAN",
  "Code 128",
  "MSI"
};
void main (void)
{
   IM_UCHAR   input[256];
   IM_ORIGIN  source;
   IM_DECTYPE symbol;
im_clear_screen(); /* Clear the screen  */
   printf("Demo im_get_label_symbology\n'q' to quit\n");
   /* Input loop  */
```

```
   do
   {
      source = IM_LABEL_SELECT | IM_KEYBOARD_SELECT;
      im_receive_field(source, IM_INFINITE_TIMEOUT, IM_INVERSE,
          IM_RETURN_ON_FULL, 10, &source, input);
      printf("\nReceive Field:\n");
      printf("%s\n", input);
      im_get_label_symbology( &symbol);
      /* Display symbology */
      printf("\nSYMBOLOGY: %d\n%s\n", symbol, bar_code[symbol]);
   } while (input[0] != 'q' && input[0] != 'Q');  /* 'q' to quit */
}
```

im_get_label_symbologyid

Purpose This function gets the AIM symbology ID, such as]A0, from the most recently scanned label. Call this function after receiving data using im_receive_input or im_receive_field.

Syntax

```
#include "imt209x.h"
IM_STATUS im_get_label_symbologyid
    (IM_UCHAR far *symb);
```

IN Parameters None.

OUT Parameters *symb* Pointer to the buffer for the label symbology ID string and must be at least 6 byes in length.

Return Value This function returns one of these codes:

IM_SUCCESS Successfully retrieved.

IM_NO_SYMBOLOGY No symbology code available, or no scans received.

See Also im_receive_input, im_receive_field, im_get_label_symbology

im_get_length

Purpose This function returns the length of the string received from the designated source by the most recent input function (im_receive_input, im_receive_field, or im_receive_buffer).

Syntax

```
#include "imt209x.h"
IM_USHORT im_get_length
    (IM_ORIGIN source);
```

IN Parameters *source* Specifies the source of the input. Choose one of these constants:

IM_LABEL_SELECT Label selected.

IM_KEYBOARD_SELECT Keypad selected.

IM_COM1_SELECT COM1 selected.

OUT Parameters None.

Return Value This function returns the length of the last input string read from the designated source.

Notes All input from the keypad or labels has a null termination character added to the end of the string so that it can be used as a normal C string. However, some data might contain embedded null characters, such as data from COM or NET sources. If so, this function supplies the true data length.

See Also im_receive_input, im_receive_field

Example See example for im_receive_input.

im_get_system_julian_date

Purpose This function generates the current date in the Julian Date format, where each day in the year is numbered from 1 to 365 (or 366 in a leap year).

Syntax

```
#include "imt209x.h"
void im_get_system_julian_date
    (IM_USHORT digits_in_year,
    (IM_CHAR *julian_date);
```

IN Parameters *digits_in_year* Number of digits to specify the year: 0, 1, 2, 3, or 4.

OUT Parameters *julian_date* Current date in Julian Date format.

Return Value None.

Example

```
#include "im209x.h"

void main ()
{
    IM_USHORT digitsInYear = 4;    //4-digit year
    IM_CHAR *juliandate;

    im_get_system_julian_date(digitsInYear, juliandate);
    printf("Julian Date is %s\n", julianDate);
}
```

im_input_status

Purpose This function provides compatibility with the JANUS PSK functions. This function checks to see if any input buffers have data and returns the buffer identification.

Syntax

```
#include "imt209x.h"
IM_ORIGIN im_input_status (void);
```

IN Parameters None.

OUT Parameters None.

Return Value This function returns one or more of these constants:

IM_NO_SELECT No input buffer has data.

IM_KEYBOARD_SELECT Keypad was pressed.

IM_COM1_SELECT COM1 selected.

IM_ALL_SELECT All input buffers are selected.

Notes To avoid entering a battery-wasting infinite loop waiting for input, use an input function instead.

See Also im_receive_input, im_receive_field

im_irl_a

Purpose This function returns input from bar code labels or the keypad in the same manner as IRL command A (ASCII input). This function returns the input data to the buffer and displays the data.

Syntax

```
#include "imt209x.h"
IM_USHORT im_irl_a
    (IM_USHORT timeout,
    IM_LENGTH_SPEC test_table[],
    IM_UCHAR mask_string[],
    IM_UCHAR *instring,
    IM_USHORT *cmd_count,
    IM_DECTYPE *symbology);
```

IN Parameters *timeout* Currently, the only receive timeout period is the constant:

IM_INFINITE_TIMEOUT Wait forever—the function does not return until the end of message character has been received.

test_table Specifies acceptable lengths for input. Data is returned only if its length matches one of the five lengths specified in the *test_table.* The *test_table* parameter is a matrix in the form shown to the right:

{a, b, c, d},
{a, b, c, d},
{a, b, c, d},
{a, b, c, d},
{a, b, c, d},

a This position in the matrix is one of these values:

IM_NO_LENGTH Accept data of any length. Set any unused table entries to {IM_NO_LENGTH,0,0,0}.

IM_LENGTH Accept data with a specific length. The actual length of the data string is placed in the d position (and b and c are not used).

IM_RANGE Accept data within a length range. The data length must be within the range of b and *c* (and *d* is not used).

*mask_*string Sets up a data mask that the received data must match. mask_string can accept a string of constants or wildcard characters. For example, use the string ### - #### to accept only phone numbers. If you define a mask, the computer beeps when input does not fit the mask. You can use one or more of these wildcard characters to define the mask:

\# Numeric

@ Alpha

? Alphanumeric printable

NULL (CHR$(0)) No mask

OUT Parameters *instring* Input string.

cmd_count Returns a 0.

symbology Returns IM_UNKNOWN_DECODE.

Return Value This function returns the status code:

IM_SUCCESS Successfully received input.

Notes TRAKKER T2090 computers do not support IRL. This function provides compatibility with previous versions of PSK and is not recommended for new development. For more information on IRL and command A, refer to the *IRL Programming Reference Manual.*

See Also im_irl_k, im_irl_n, im_irl_v, im_irl_y

im_irl_k

Purpose This function receives input from the keypad in any format in the same manner as IRL command K (ASCII input). This function returns the input data to the buffer and displays the data.

Syntax

```
#include "imt209x.h"
IM_USHORT im_irl_k
    (IM_USHORT timeout,
    IM_LENGTH_SPEC test_table[],
    IM_UCHAR mask_string[],
    IM_UCHAR *instring,
    IM_USHORT *cmd_count);
```

IN Parameters *timeout* Currently, the only receive timeout period is the constant:

IM_INFINITE_TIMEOUT Wait forever—the function does not return until the end of message character has been received.

test_table Specifies acceptable lengths for input. Data is returned only if its length matches one of the five lengths specified in the *test_table.* The *test_table* parameter is a matrix in the form shown to the right:

{a, b, c, d},
{a, b, c, d},
{a, b, c, d},
{a, b, c, d},
{a, b, c, d},

a This position in the matrix is one of these values:

IM_NO_LENGTH Accept data of any length. Set any unused table entries to {IM_NO_LENGTH,0,0,0}.

IM_LENGTH Accept data with a specific length. The actual length of the data string is placed in the *d* position (and *b* and *c* are not used).

IM_RANGE Accept data within a length range. The data length must be within the range of *b* and *c* (and *d* is not used).

mask_string Sets up a data mask that the received data must match. *mask_string* can accept a string of constants or wildcard characters. For example, use the string ### - #### to accept only phone numbers. If you define a mask, the computer beeps when input does not fit the mask. You can use one or more of these wildcard characters to define the mask:

Numeric

@ Alpha

? Alphanumeric printable

NULL (CHR$(0)) No mask

OUT Parameters *instring* Input string.

cmd_count Returns a 0.

Return Value This function returns the status code:

IM_SUCCESS Successfully received input.

Notes TRAKKER T2090 computers do not support IRL. This function provides compatibility with previous versions of PSK and is not recommended for new development. For more information on IRL and command K, refer to the *IRL Programming Reference Manual*.

See Also im_irl_a, im_irl_n, im_irl_v, im_irl_y

im_irl_n

Purpose This function receives numeric input from the keypad or a label in the same manner as IRL command N (numeric input). Nonnumeric data is not accepted as valid input. This function returns the input data to the buffer and displays the data.

Syntax

```
#include "imt209x.h"
IM_USHORT im_irl_n
    (IM_USHORT timeout,
    IM_LENGTH_SPEC test_table[],
    IM_UCHAR *instring,
    IM_USHORT *cmd_count,
    IM_DECTYPE *symbology);
```

IN Parameters

timeout Specifies the receive timeout period. The return status indicates whether the function was successful or a timeout occurred. Either enter a number from 1 to 65,534 to indicate the length of the timeout in milliseconds, or choose one of these constants:

IM_ZERO_TIMEOUT No wait.

IM_INFINITE_TIMEOUT Wait forever—the function does not return until the end of message character has been received.

test_table Specifies acceptable lengths for input. Data is returned only if its length matches one of the five lengths specified in the *test_table*. The *test_table* parameter is a matrix in the form shown to the right:

{a, b, c, d},
{a, b, c, d},
{a, b, c, d},
{a, b, c, d},
{a, b, c, d},

a This position in the matrix is one of these values:

IM_NO_LENGTH Accept data of any length. Set any unused table entries to {IM_NO_LENGTH,0,0,0}.

IM_LENGTH Accept data with a specific length. The actual length of the data string is placed in the *d* position (and *b* and *c* are not used).

IM_RANGE Accept data within a length range. The data length must be within the range of *b* and *c* (and *d* is not used).

mask_string Sets up a data mask that the received data must match. *mask_string* can accept a string of constants or wildcard characters. For example, use the string ### - #### to accept only phone numbers. If you define a mask, the computer beeps when input does not fit the mask. You can use one or more of these wildcard characters to define the mask:

Numeric

@ Alpha

? Alphanumeric printable

NULL (CHR$(0)) No mask

instring Input string.

OUT Parameters

cmd_count Returns a 0.

symbology Returns IM_UNKNOWN_DECODE.

Return Value This function returns one of these codes:

IM_SUCCESS Successfully received input.

IM_TIMEDOUT A timeout occurred.

IM_EDIT_ERROR Error occurred in a computer command.

Notes TRAKKER T2090 computers do not support IRL. This function provides compatibility with previous versions of PSK and is not recommended for new development. For more information on IRL and command N, refer to the *IRL Programming Reference Manual*.

See Also im_irl_a, im_irl_k, im_irl_v, im_irl_y

im_irl_v

Purpose This function receives input from any specified source in any format, in the same manner as an IRL command V (universal input).

Syntax

```
#include "imt209x.h"
IM_USHORT im_irl_v
    (IM_USHORT timeout,
    IM_CONTROL edit,
    IM_LABEL_BEEP_CONTROL beep,
    IM_CONTROL display,
    IM_ORIGIN *source,
    IM_UCHAR *instring,
    IM_USHORT *cmd_count,
    IM_DECTYPE *symbology);
```

IN Parameters

timeout Specifies the receive timeout period. The return status indicates whether the function was successful or a timeout occurred. Either enter a number from 1 to 65,534 to indicate the length of the timeout in milliseconds, or choose one of these constants:

IM_ZERO_TIMEOUT No wait.

IM_INFINITE_TIMEOUT Wait forever—the function does not return until the end of message character has been received.

edit Determines the amount of data to accept. Choose one of these constants:

IM_DISABLE Accepts one character of keypad data.

IM_ENABLE Accepts strings of data from the keypad.

beep Determines whether the IRL V command beeps or not when data is entered. Choose one of these constants:

IM_APPLI_BEEP Application controls the beep—you code the application to sound a beep when your program design requires one.

IM_WEDGE_BEEP Beeps occur automatically—the reader always beeps when data is entered.

display Determines if the data is displayed as it is entered. Choose one of these constants:

IM_DISABLE Disable display of data.

IM_ENABLE Enable display of data.

IN/OUT Parameters

source Determines which input sources are allowed. When the IRL V command returns, *source* indicates where the data came from. The *source* parameter is one of these constants:

IM_NO_SELECT No source selected.

IM_LABEL_SELECT Label selected.

IM_KEYBOARD_SELECT Keypad selected.

IM_COM1_SELECT COM1 selected.

IM_ALL_SELECT All sources are selected.

OUT Parameters

instring Input string.

cmd_count Returns a 0.

symbology Specifies the type of bar code and is one of these constants:

IM_UNKNOWN_DECODE Unknown bar code.

IM_CODABAR Codabar bar code.

IM_CODE_39 Code 39 bar code.

IM_CODE_128 Code 128 bar code.

IM_I_2_OF_5 Interleaved 2 of 5.

IM_MSI MSI bar code.

IM_UPC Universal Product code.

Return Value

This function returns one of these codes:

IM_SUCCESS Successfully received input.

IM_TIMEDOUT A timeout occurred.

Notes

TRAKKER T2090 computers do not support IRL. This function provides compatibility with previous versions of PSK and is not recommended for new development. For more information on IRL and command V, refer to the *IRL Programming Reference Manual*.

See Also

im_irl_a, im_irl_k, im_irl_n, im_irl_y

im_irl_y

Purpose The im_irl_y function receives input from the designated communications port the same as IRL command Y (ASCII input). This function receives a single block, not an entire file. This function always clears input from the host and checks the data for computer commands from input.

Unlike the other IRL-type instructions, the data is not automatically displayed.

Syntax

```
#include "imt209x.h"
IM_STATUS im_irl_y
    (IM_USHORT timeout,
    IM_COM_PORT port_id,
    IM_UCHAR *eom_char,
    IM_PROTOCOL_CMD protocol,
    IM_UCHAR *instring,
    IM_USHORT *cmd_count);
```

IN Parameters

timeout Specifies the receive timeout period. The return status indicates whether the function was successful or a timeout occurred. Either enter a number from 1 to 65,534 to indicate the length of the timeout in milliseconds, or choose one of these constants:

IM_ZERO_TIMEOUT No wait.

IM_INFINITE_TIMEOUT Wait forever—the function does not return until the end of message character has been received.

port_id Identifies the communications port. Use this constant:

IM_COM1 COM1.

eom_char Provides compatibility with the JANUS PSK. This parameter is ignored on the TRAKKER T2090 computer.

protocol Provides compatibility with the TRAKKER Antares and JANUS PSK. This parameter is ignored on the TRAKKER T2090 computer. Use the im_command to control protocol on the computer.

OUT Parameters

instring Input string.

cmd_count Returns a 0.

Return Value This function returns one of these codes:

IM_SUCCESS Successfully received input.

IM_TIMEDOUT A timeout occurred.

Notes TRAKKER T2090 computers do not support IRL. This function provides compatibility with previous versions of PSK and is not recommended for new development. For more information on IRL and command Y, refer to the *IRL Programming Reference Manual.*

See Also im_irl_a, im_irl_k, im_irl_n, im_irl_v

im_message

Purpose This function displays the error message associated with a specific status code returned by a PSK function. Use this function to display additional information about status codes during application development.

Syntax
```
#include "imt209x.h"
void im_message(IM_USHORT status_code);
```

IN Parameters *status_code* Standard status code returned from various PSK functions.

OUT Parameters None.

Return Value None.

Notes The status message is displayed at the current cursor location without any formatting.

im_offset_dbyte

Purpose This function resets the offset of the first glyph in the current DBFS font.

Syntax
```
#include "imt209x.h"
IM_STATUS im_offset_dbyte
    (IM_USHORT offset);
```

IN Parameters *offset* New index of the first glyph in the fontfile (in glyphs).

OUT Parameters None.

Return Value This function returns one of these codes:

IM_SUCCESS Offset changed.

IM_INVALID_FUNCTION im_dbyte_setfont() must be called first.

Notes The offset may be changed as often as desired.

See Also im_putchar_dbyte, im_puts_dbyte, im_dbyte_setfont

im_putchar

Purpose This function places a character at the current cursor position and changes the line attribute to the specified attribute.

Syntax

```
#include "imt209x.h"
IM_STATUS im_putchar
    (IM_UCHAR char,
    IM_ATTRIBUTES attrib);
```

IN Parameters *char* Specifies the character to be displayed

attrib Specifies the display attribute for the character. Choose one of these constants:

IM_NORMAL Plain text.

IM_INVERSE Inverse color text.

IM_UNDERLINE Underline text.

IM_UNCHANGED Leave attribute unchanged.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_SUCCESS Success.

IM_INVALID_PARAM_1 Invalid attribute value.

Notes On the TRAKKER T2090 computer, placing a character with the IM_INVERSE attribute causes the entire line to be inverted.

See Also im_puts

im_putchar_dbyte

Purpose This function places the indicated glyph in the current DBFS font on the screen.

Syntax

```
#include "imt209x.h"
IM_STATUS im_putchar_dbyte
    (IM_USHORT dbfschar,
     IM_ATTRIBUTES attrib);
```

IN Parameters *dbfschar* Index to the desired glyph in the fontfile (in glyphs).

attrib Specifies the display attribute for the glyph. Choose one or more of these constants:

IM_NORMAL Set bits in the bitmap are set on the screen.

IM_INVERSE…….Cleared bits in the bitmap are set on the screen. *Mutually exclusive with IM_NORMAL.*

IM_UNDERLINE The last row of pixels in the character cell are plotted as though they were set in the bitmap.

IM_NOT_ADVANCING_CURSOR The cursor is not advanced after the glyph is plotted. This is useful in preventing scrolling when plotting a glyph in the last position of the last row.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_SUCCESS Success.

IM_INVALID_PARAM_1 Either a glyph could not be mapped or the attribute requested is invalid. Glyphs up to the bad mapping point will be plotted.

IM_INVALID_FUNCTION `im_dbyte_setfont()` must be called first

Notes Glyph 0 cannot be plotted with this function because that value is used to signify ‘end of string’. If offset != 0, this is not a problem.

See Also im_dbyte_setfont, im_puts_dbyte, im_offset_dbyte

im_puts

Purpose This function places a string on the screen at the current cursor location and appends a carriage return and line feed (CR LF) to the string.

Syntax

```
#include "imt209x.h"
IM_STATUS im_puts
    (IM_UCHAR far *string,
    IM_ATTRIBUTES attrib);
```

IN Parameters *string* Far pointer to the text string to be displayed.

attrib Specifies the display attribute for the string. Choose one of these constants:

IM_NORMAL Plain text.

IM_INVERSE Inverse color text.

IM_UNDERLINE Underline text.

IM_UNCHANGED Leave attribute unchanged.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_SUCCESS Success.

IM_BAD_ADDRESS Invalid string address.

IM_INVALID_PARAM_2 Invalid attribute value.

Notes On the TRAKKER T2090 computer, placing a character with the IM_INVERSE attribute causes the entire line to be inverted.

See Also im_cputs, im_putchar

im_puts_dbyte

Purpose This function places the indicated glyphs in the current DBFS font on the screen.

Syntax

```
#include "imt209x.h"
IM_STATUS im_puts_dbyte
    (IM_USHORT *dbfschar,
     IM_ATTRIBUTES attrib);
```

IN Parameters

dbfschar Pointer to a series of indices for glyphs in the fontfile (in glyphs). Values of *offset* will plot the bitmap starting at *0* in the file. The series is terminated by 0x0000.

attrib Specifies the display attribute for the glyphs. Choose one or more of these constants:

IM_NORMAL Set bits in the bitmap are set on the screen.

IM_INVERSE Cleared bits in the bitmap are set on the screen. *Mutually exclusive with IM_NORMAL.*

IM_UNDERLINE The last row of pixels in the character cell are plotted as though they were set in the bitmap.

IM_NOT_ADVANCING_CURSOR The cursor is not advanced after the glyph is plotted. This is useful in preventing scrolling when plotting a glyph in the last position of the last row, but causes successive characters in a string to be plotted in the same position.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_SUCCESS Success.

IM_INVALID_PARAM_1 Either the glyph could not be mapped or the attribute requested is invalid.

IM_INVALID_FUNCTION `im_dbyte_setfont()` must be called first.

Notes The string must be terminated by a double byte null (i.e., two bytes of 0).

See Also im_dbyte_setfont, im_putchar_dbyte, im_offset_dbyte

im_receive_buffer

Purpose This function receives the contents of a data buffer from the serial communications port.

Syntax

```
#include "imt209x.h"
IM_STATUS im_receive_buffer
    (IM_COM_PORT port_id,
    IM_USHORT length,
    IM_UCHAR far *data_buffer,
    IM_LTIME timeout,
    IM_USHORT far *comm_length);
```

IN Parameters

port_id Identifies the communications port. Use this constant:

IM_COM1 COM1 port.

length Specifies the maximum number of bytes to receive.

data_buffer Far pointer to the data array where you want to place the received data. This buffer must hold at least the number of bytes passed in as *length*.

timeout Specifies the receive timeout period. Either enter a number from 0 to 4,294,967,294 to indicate the length of the timeout in milliseconds (55ms granularity), or choose one of these constants:

IM_INFINITE_NET_TIMEOUT Never timeout.

IM_ZERO_TIMEOUT No wait.

OUT Parameters

comm_length Far pointer to the variable that will hold the actual number of bytes received upon completion of the call. If IM_SUCCESS is not returned this value will be 0.

Return Value This function returns one of these codes:

IM_SUCCESS Successfully received data.

IM_NET_BAD_DATA Data pointer is null or invalid data length.

IM_NET_DATA_LENGTH Data buffer size is too small for frame.

IM_TIMEDOUT No data was received in the timeout period.

IM_INVALID_PORT Invalid *port_id*.

IM_BUFFER_OVERFLOW Receive buffer overflowed and needed to be flushed. This error will only be returned in character mode.

Notes This function does not return until an end of message, a buffer is full, a timeout occurs or an error occurs. If no EOM character is defined, the function returns after a character is received.

See Also im_receive_field, im_transmit_buffer

Example

```
/******************** im_receive_buffer ****************************/
#include <string.h>
#include <conio.h>
#include "stdio.h"
#include "im209x.h"

void main ( void )
{
char      szRxBuffer[1024];
IM_STATUS iStatus;
IM_USHORT iCommLength;

    im_clear_screen();

    iStatus = im_receive_buffer ( IM_NETUDP, 1024, szRxBuffer, 10000L, &iCommLength );

    if(iStatus == IM_SUCCESS)
      printf("\nData Receive: %s, Length: %u\n", szRxBuffer, iCommLength);
    else
    {
      printf("\nReceive Buffer Err:\nStatus Code#: %x\n", iStatus);
      im_message(iStatus);
    }

    getch();
}
```

im_receive_field

Purpose This function manages an input field area on the screen. You can specify display attributes for the field and control the length of the input data.

Syntax

```
#include "imt209x.h"
IM_STATUS im_receive_field
    (IM_ORIGIN allowed,
    IM_UINT timeout,
    IM_ATTRIBUTES attrib
    IM_ULONG flags
    IM_SHORT length
    IM_ORIGIN *source,
    IM_CHAR *received);
```

IN Parameters

allowed Defines the available input source. Choose one or more of these constants:

IM_LABEL_SELECT Label selected.

IM_KEYBOARD_SELECT Keypad selected.

IM_COM1_SELECT COM1 selected

IM_ALL_SELECT All sources selected.

timeout Specifies the receive timeout period. Either enter a number from 1 to 65,534 to indicate the length of the timeout in milliseconds, or choose one of these constants:

IM_ZERO_TIMEOUT No wait.

IM_INFINITE_TIMEOUT Wait forever—the function does not return until the end of message character is received, a return is entered, or one of the conditions set with the *flags* parameter is met. If COM1 is selected, IM_INFINITE_TIMEOUT acts like IM_INFINITE_NET_TIMEOUT.

attrib Specifies the display attributes. Choose one of these constants:

IM_NORMAL Plain text.

IM_INVERSE Inverse color text.

IM_UNDERLINE Underline text.

IM_UNCHANGED Leave attribute unchanged.

flags Controls the field action. Choose one or more of these constants:

IM_ERASE_FIELD Clear field data and display any field attributes on screen, filling the field area. If this flag is not set, the old data is displayed and the field is padded with blanks. Attributes are applied to the current row.

IM_RETURN_ON_FULL If the input data fills the field, display the truncated data, and then exit the field.

IM_RETURN_ON_FUNCTION If a function key is pressed, then display all of the data entered into the field and return the data in *received*.

IM_DISPLAY_ONLY Display the field and its attributes without waiting for input.

IM_AT_END Move the cursor to the end of the data already in the field.

IM_STAY_IN_FIELD Cursor stays in the input field upon field exit.

IM_NO_DISPLAY Receive input but do not echo the input to the display.

IM_START_IN_INSERT Line editing mode is set to insert (default value).

IM_UPCASE Changes input to upper case.

IM_LOCASE Changes input to lower case.

Note: *If both IM_UPCASE and IM_LOCASE are set, then IM_UPCASE is used. If neither flag is set, keys are interpreted as pressed.*

length Defines the display field size. The buffer needs to be at least one byte larger.

OUT Parameters

source Specifies the actual input source and is one of these constants:

IM_NO_SELECT No selection made.

IM_LABEL_SELECT Label selected.

IM_KEYBOARD_SELECT Keypad selected.

IM_COM1_SELECT COM1 selected.

received Pointer to the variable where the data is placed.

Return Value

This function returns one of these codes:

IM_SUCCESS Successfully received input.

IM_TIMEDOUT Timeout occurred.

IM_INPUT_FULL Maximum number of characters was received and input was stopped. All characters entered are returned.

IM_RETURN_F1 F1 was received.

IM_RETURN_F2 F2 was received.

IM_RETURN_F3 F3 was received.

Notes If input from more than one source is received before this function is called, the first available input is returned in this order: label, keypad, and COM1.

When *allowed* is set to IM_COM1_SELECT and *timeout* is set to IM_INFINITE_TIMEOUT, this function performs as if *timeout* were set to IM_INFINITE_NET_TIMEOUT.

See Also im_receive_buffer

Example

```
/******************** im_receive_field*****************************/
/*Example of doing a data input screen using im_receive_field */
/* Also validates input for length and draws box.            */

#include "imt209x.h"
#include "string.h"

/* Fields and prompts */
char sBadge[10] ={0}, sPart[26]={0}, sOrderNo[10]={0};
char Label0[] ="Job Setup",         Label1[]="Enter Badge:",
     Label2[] ="Scan Part Number:", Label3[] = "Enter Order Number:";

#define FIELD_FLAGS IM_RETURN_ON_TAB | IM_RETURN_ON_FULL | IM_AT_END

/* Table of information to drive display and data input */
struct screen{
    char * pszText;
    IM_USHORT iRow, iCol, iLength, iMinLength;
    IM_ATTRIBUTES  iAttribute;
    IM_USHORT iFlags;
    } aScreen[] = {
     { Label0  ,  1, 5,  sizeof(Label0)-1, 0, IM_NORMAL,     IM_DISPLAY_ONLY },
     { Label1  ,  3, 1,  sizeof(Label1)-1, 0, IM_NORMAL,     IM_DISPLAY_ONLY },
     { Label2  ,  5, 1,  sizeof(Label2)-1, 0, IM_NORMAL,     IM_DISPLAY_ONLY },
     { sBadge  ,  4, 1,     9,             3, IM_INVERSE,  FIELD_FLAGS },
     { sPart   ,  6, 1,    25,             0, IM_INVERSE,  FIELD_FLAGS },
     {(void *)0,0,0,0,0,0} /*Termination line*/
    };
    /*note that sizeof includes the null terminator and we pass in the displayable size*/

/* Simple example validation routine. */
IM_BOOL DoValidation ( char * szField, IM_USHORT iMinLength )
{
  if ( (IM_USHORT)strlen( szField ) >= iMinLength )
    return IM_TRUE;
  else
  {
    im_status_line("Input to short", IM_TRUE, 50);
    return IM_FALSE;
  }
}
```

```
void main (void)
{
   IM_ULONG  iSetup = IM_DISPLAY_ONLY, iPassFlags, ii=0;
   IM_STATUS iStatus;
   IM_ORIGIN iSource;

   /* set up display */
   im_clear_screen();

/*****************************************************************************/
/* loops through once to display prompts and fields then comes back through to gather  */
/* input. If validation fails stays in field until validation passes.          */
/*****************************************************************************/
   do
   {
      im_set_cursor_xy( aScreen[ii].iRow, aScreen[ii].iCol );
      iPassFlags = aScreen[ii].iFlags | iSetup ;
      iStatus = im_receive_field( IM_KEYBOARD_SELECT | IM_LABEL_SELECT,
                                  IM_INFINITE_TIMEOUT, aScreen[ii].iAttribute,
                                  iPassFlags,aScreen[ii].iLength, &iSource,
                                  aScreen[ii].pszText);
      /* Validate if not display pass */
      if (iSetup & IM_DISPLAY_ONLY)
      {
         ii++;
      }
      else if (DoValidation(aScreen[ii].pszText, aScreen[ii].iMinLength ) )
      {
         ii++;
         iSetup = iSetup & !IM_AT_END;
      }
      else  /* Must have been error, go to end of same field */
        iSetup = iSetup | IM_AT_END;

      /*See if display pass done and if is, turn into data input pass. */
      if ((iSetup & IM_DISPLAY_ONLY) && ( aScreen[ii].pszText == (void *)0 ) )
      {
          iSetup = iSetup & !IM_DISPLAY_ONLY ; /*reset the display only bit  */
          ii = 0;                              /* and start again at the top */
      }
   } while ( aScreen[ii].pszText != (void *)0 ) ;
}
```

im_receive_input

Purpose This function gets input from the source and places it into the received buffer. You can use the im_get_length function after this function to get the input length.

Syntax

```
#include "imt209x.h"
IM_STATUS im_receive_input
    (IM_ORIGIN allowed,
    IM_UINT timeout,
    IM_ORIGIN *source,
    IM_CHAR *received);
```

IN Parameters

allowed Defines the available input source. Choose one or more of these constants:

IM_LABEL_SELECT Label selected.

IM_KEYBOARD_SELECT Keypad selected.

IM_COM1_SELECT COM1 selected.

IM_ALL_SELECT All sources selected.

Use these variables to modify the input by performing a logical OR with the above input sources.

IM_KEYCODE_ENABLE Applies when the keyboard or a label is a source. Returns each key pressed and its key-code, or returns one character of a label per call. Does not echo to the screen. This is the functional equivalent to setting the input mode to IM_DESKTOP.

Keyboard characters are returned as 4 bytes. The first byte is the ASCII code. The second byte is the scan code, and the last 2 bytes are flags for modifier keys (Shift, Control, and Alt).

timeout Specifies the receive timeout period. Either enter a number from 1 to 65,534 to indicate the length of the timeout in milliseconds, or choose one of these constants:

IM_ZERO_TIMEOUT No wait.

IM_INFINITE_TIMEOUT Wait forever—the function will not return until the end of message character has been received. If COM1 is selected, IM_INFINITE_TIMEOUT acts like IM_INFINITE_NET_TIMEOUT.

OUT Parameters

source Specifies the actual input source and is one of these constants:

IM_NO_SELECT No selection made.

IM_LABEL_SELECT Label selected.

IM_KEYBOARD_SELECT Keypad selected.

IM_COM1_SELECT COM1 selected.

received Pointer to the variable where the data is placed.

Return Value

This function returns one of these codes:

IM_SUCCESS Success.

IM_TIMEDOUT Timeout occurred.

Notes If input from more than one source is received before this function is called, the first available input is returned in this order: label, keypad, and COM1.

When *allowed* is set to IM_COM1_SELECT and *timeout* is set to IM_INFINITE_TIMEOUT, this function performs as if *timeout* were set to IM_INFINITE_NET_TIMEOUT.

See Also im_receive_buffer, im_receive_field

Example

```
/******************** im_receive_input ****************************/
#include <stdio.h>
#include <stdlib.h>
#include "im209x.h"
IM_UCHAR   input[1024];
void main (void)
{
IM_USHORT  length;
IM_ORIGIN  source;
IM_STATUS  status;

   im_clear_screen();          /* Clear the screen  */
   printf("Demo \nim_receive_input\n'Q' to quit\n\'C' to clear screen\n");

   /* Input loop  */
   do
   {
   /* Set up input source */
   source = IM_LABEL_SELECT | IM_KEYBOARD_SELECT;
   /* Request input from label, keypad */
   status = im_receive_input(source, IM_INFINITE_TIMEOUT, &source, input);
   length = im_get_length(source);
   if (IM_ISGOOD(status))
   {
      /* Show the input source */
      if (source == IM_LABEL_SELECT)
         printf("\nLabel input:\n");
      else if (source == IM_KEYBOARD_SELECT)
         printf("\nKeybd input:\n");
      /* Display input data */
      printf("%s\nInput length: %d\n", input, length);
   }
   else /* input error */
      printf("input error\n");
   /* Upper case first char of input for simplifying to test input */
   input[0] = toupper(input[0]);
   /*If the first char in string is 'C', then clear screen.*/
   if (input[0] == 'C')
      im_clear_screen();
} while (input[0] != 'Q');  /* First number in string is 'Q', then stop */
}
```

im_set_2of5_lengths

Purpose This function sets one or two lengths for Interleaved 2 of 5 bar code labels. If you specify two lengths, the T2090 can read I 2 of 5 bar code labels that conform to either length.

Syntax

```
#include "imt209x.h"
IM_STATUS im_set_2of5_length
    (unsigned char length1,
    unsigned char length2) ;
```

IN Parameters *length1* First length for the labels. Choose a number:

0 Interleaved 2 of 5 is disabled on the T2090.

2 to 30 Even numbers only.

length2 Second length for the labels. Choose a number:

0 Only one length (*length1*) is valid for Interleaved 2 of 5 bar code labels.

2 to 30 Even numbers only.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_SUCCESS Success.

IM_INVALID_PARAM_1 The value for *length1* is invalid (not an even number, or larger than 30).

IM_INVALID_PARAM_2 The value for *length2* is invalid (not an even number, or larger than 30).

im_set_com_mode

Purpose This function specifies whether the communications port operates in RS-232 or RS-485 mode.

Syntax

```
#include "imt209x.h"
IM_STATUS im_set_com_mode
    (IM_USHORT mode) ;
```

IN Parameters *mode* Identifies the communication mode. Choose one of these constants:

IM_MODE_RS232 RS-232 communication mode.

IM_MODE_RS485 RS-485 communication mode.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_OK Success.

IM_INVALID_PORT The value for *port_id* is invalid.

IM_INVALID_MODE The value for *mode* is invalid.

See Also im_set_port_direction

im_set_cursor_style

Purpose This function sets the style used to display the cursor. It is DBFS aware.

Syntax
```
#include "imt209x.h"
IM_STATUS im_set_cursor_style
    (IM_CURS_TYPE cursor);
```

IN Parameters *cursor* Specifies the style of cursor. Choose one of these constants:

IM_UNDERLINE Single underline.

IM_NO_CURSOR No cursor displayed.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_SUCCESS Success.

IM_NOT_SUPPORTED Not supported for current font type.

im_set_cursor_xy

Purpose This function sets the current cursor position.

Syntax
```
#include "imt209x.h"
IM_STATUS im_set_cursor_xy
    (IM_USHORT row,
    IM_USHORT col);
```

IN Parameters None.

OUT Parameters *row* Vertical position. The top of the display is 0.

col Horizontal position. The left edge of the display is 0.

Return Value This function returns one of these codes:

IM_SUCCESS Success.

IM_X_RANGE *col* value out of range, cursor moved to right edge.

IM_Y_RANGE *row* value out of range, cursor moved to bottom.

IM_BOTH_RANGE *col* and *row* values out of range, cursor moved to lower right corner.

Example See example for im_receive_field.

im_set_display_mode

Purpose This function sets the character height of the display. Scroll and wrap parameters are included for compatibility with programs written for some TRAKKER Antares terminals.

Syntax

```
#include "imt209x.h"
IM_STATUS im_set_display_mode
    (IM_FONT_TYPE font,
    IM_BOOL scroll
    IM_BOOL wrap);
```

IN Parameters *font* Specifies the font type code. Choose one of these constants:

IM_FONT_STANDARD Text is 5 x 7 pixels, the scroll boundary is line 8, and the wrap boundary is column 20.

IM_FONT_LARGE Text is 5 x 14 pixels, the scroll boundary is line 4, and the wrap boundary is column 10.

IM_FONT_SPECIAL Text is 10 x 14 pixels

scroll Should always be passed in as non-zero for scroll at bottom of screen.

wrap Should always be passed in as non-zero for wrap at the right edge of the screen.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_SUCCESS Success.

IM_INVALID_PARAM_1 Invalid font parameter.

IM_INVALID_PARAM_2 Invalid scroll parameter (can not disable).

IM_INVALID_PARAM_3 Invalid wrap parameter (can not disable).

See Also im_get_display_mode, im_get_display_type, im_get_display_size_physical

im_set_input_mode

Purpose This function sets the reader input mode to Wedge, Programmer, or Desktop. These modes affect how the reader interprets and stores input.

Syntax

```
#include "imt209x.h"
IM_STATUS im_set_input_mode
    (IM_MODE mode);
```

IN Parameters *mode* Specifies the mode. Choose one of these constants:

IM_PROGRAMMER Input is returned as a string (default). Simple line editing is permitted using backspace.

IM_WEDGE Input is returned as a string. Use ⇦ (Backspace) for simple line editing.

IM_DESKTOP Keyboard characters are returned as 4 bytes. The first byte is the ASCII code. The second byte is the scan code, and the last 2 bytes are flags for modifier keys (Shift and Control).

OUT Parameters None.

Return Value IM_SUCCESS Success.

See Also im_get_input_mode, im_receive_input

im_set_kbmode

Purpose This function sets the keyboard mode to either alphabetic or numeric.

Syntax

```
#include "imt209x.h"
void IM_SET_KBMODE(int mode);
```

IN Parameters *mode* Choose one of these constants:

KEYBOARD_MODE_ALPHA Sets the keyboard mode to alphabetic.

KEYBOARD_MODE_NUMBER Sets the keyboard mode to numeric.

OUT Parameters None.

Return Value None.

Notes When the user presses keys in alphabetic mode, the alphabetic characters marked on the right and left edges of the keys are returned.

When the user presses keys in numeric mode, the numbers marked in the moddle of the keys are returned.

See Also im_get_kbmode

im_set_port_direction

Purpose This function specifies whether the communications port receives or transmits data.

Syntax
```
#include "imt209x.h"
IM_STATUS im_set_port_direction
    (IM_USHORT direction) ;
```

IN Parameters *direction* Identifies the direction of data transfer. Choose one of these constants:

IM_RS485_RECV The serial COM port receives data.

IM_RS485_XMIT The serial COM port transmits data.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_OK Mode set successfully.

IM_INVALID_PORT The value for *port_id* is invalid.

IM_INVALID_DIRECTION The value for *direction* is invalid.

See Also im_set_com_port

im_set_scanning

Purpose This function enables or disables the TRAKKER T2090 computer's scanning capability.

Syntax
```
#include "imt209x.h"
IM_STATUS im_set_scanning
    (IM_CONTROL enable_disable);
```

IN Parameters *enable_disable* Specifies if the scanning beam will be activated when the user presses the ENTER key with no data keyed in. Choose one of these constants:

IM_ENABLE Enable the scanner.

IM_DISABLE Disable the scanner.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_SUCCESS Success.

IM_SCAN_PORT_NOT_OPEN The port was not open.

IM_SCAN_PORT_FAILED The port failed to open for scanning.

IM_COMM_PORT_OPENED The port is open for serial communication. The port must be closed before the T2090 computer can scan.

IM_RESUME_OCCURRED The function determined that the T2090 computer recently came out of Suspend mode.

IM_UNIT_DOCKED The T2090 computer was docked in the cradle when the scanner was activated.

im_sound

Purpose This function generates a beep of specified pitch and duration. For example, use no beep or a short beep for library use, a long beep for manufacturing use, or a unique beep to distinguish among other computers.

Syntax

```
#include "imt209x.h"
IM_STATUS im_sound
    (IM_USHORT pitch,
    IM_USHORT duration,
    IM_USHORT volume);
```

IN Parameters *pitch* Specifies the frequency of the beep you want the computer to make. Either enter a number from 20 to 8189 to indicate the pitch in Hz, or choose one of these constants:

IM_HIGH_PITCH 2400 Hz.

IM_LOW_PITCH 1200 Hz.

IM_VERY_LOW_PITCH 600 Hz.

duration Specifies the length of the beep. Either enter a number from 2 to 79999 to indicate the duration of the beep in milliseconds, or choose one of these constants (55 ms granularity):

IM_BEEP_DURATION 50 ms.

IM_CLICK_DURATION 5 ms.

volume Specifies the volume. Choose one of these constants:

IM_OFF_VOLUME Off.

IM_QUIET_VOLUME Quiet.

IM_NORMAL_VOLUME Normal.

IM_LOUD_VOLUME Loud.

IM_EXTRA_LOUD_VOLUME Extra loud.

IM_CURRENT_VOLUME Use volume from configuration menu.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_SUCCESS Successful beep.

IM_INVALID_PARAM1 Pitch is outside allowed range.

IM_INVALID_PARAM2 Duration is outside allowed range.

IM_INVALID_PARAM3 Volume is outside allowed range.

Notes The beep volumes for quiet, normal, loud, and extra loud are actually all the same volume on the TRAKKER T2090 computer.

im_standby_wait

Purpose This function places the application and computer in standby mode for a specific period of time to save the battery power.

Syntax

```
#include "imt209x.h"
IM_STATUS im_standby_wait
    (IM_USHORT timeout);
```

IN Parameters *timeout* Specifies the amount of time to wait in standby mode. Either enter a number from 1 to 65,535 (resolution of 10 ms) to indicate the length of the timeout in milliseconds, or choose one of these constants:

IM_ZERO_TIMEOUT No wait.

IM_INFINITE_TIMEOUT Wait forever.

OUT Parameters None.

Return Value IM_SUCCESS Success.

im_status_line

Purpose This function briefly displays an error message in the status line without wrapping or scrolling the display. The status line is displayed until a key is pressed or a time out occurs. The original contents of the line reappear after the message is erased.

Syntax

```
#include "imt209x.h"
IM_STATUS im_status_line
    (char far *stmessage
    IM_BOOL wait
    IM_USHORT row);
```

IN Parameters *stmessage* Pointer to the error message string to display.

wait Specifies if the application should wait for a key to be pressed. Choose one of these values:

0 Do not wait for a key.

non-zero Pause until any key is pressed or until a timeout occurs.

row Row number to display the message in. If this number is larger than the number of visible rows, the last row is used. The first row is 0.

OUT Parameters None.

Return Value Returns 0 or the key pressed to terminate waiting.

Notes If the *wait* parameter is set, the message will be erased after 20 seconds or when a key is pressed. Then, the screen is refreshed to look as it did before displaying the message.

im_timed_status_line

Purpose This function displays an error message in the status line without wrapping or scrolling the display. The status line is displayed until a key is pressed or a time out occurs. The original contents of the line reappear after the message is erased.

Syntax

```
#include "imt209x.h"
void im_timed_status_line
    (char far *string,
    IM_USHORT timeout
    IM_USHORT line);
```

IN Parameters *string* String to be displayed.

timeout Number of milliseconds to wait before erasing the string. Either enter a number from 1 to 65,535 (resolution of 10 ms) to indicate the length of the timeout in milliseconds, or choose one of these constants:

IM_ZERO_TIMEOUT No wait.

IM_INFINITE_TIMEOUT Wait forever.

line Positive number identifying the line, relative to the top of the screen, where the string is to be displayed. If this number is larger than the number of lines on the screen, the last line is used. The first line is 0.

OUT Parameters None.

Return Value None.

See Also im_status_line

im_tm_callback_cancel

Purpose This function removes a registered function from the timer callback database.

Syntax

```
#include "imt209x.h"
IM_STATUS im_tm_callback_cancel
    (IM_USHORT index);
```

IN Parameters *index* Identifies the registered function in the timer callback database. This number was returned by im_tm_callback_register when the function was added to the timer callback database.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_SUCCESS Success.

IM_INVALID_INDEX The value for *index* is invalid.

See Also im_tm_callback_register

Example See example for im_tm_callback_register.

im_tm_callback_register

Purpose This function adds a function to the timer callback database and specifies how the function will be called back.

Syntax

```
#include "imt209x.h"
IM_STATUS im_tm_callback_register
    (PTIMERCALLBACK function,
    time_t start_time,
    IM_USHORT repeat_count
    IM_ULONG period,
    IM_BOOL fDisable,
    IM_USHORT *index);
```

IN Parameters

function Pointer to the function to be called.

start_time Specifies when to perform the first callback. Either enter the time as the number of seconds elapsed since midnight on January 1, 1970, or choose this constant:

IM_CALLBK_NOW Start the first callback immediately.

repeat_count Specifies the number of callbacks. Either enter any number from 2 to 65534 (two-byte range of IM_USHORT), or choose one of these constants:

IM_CALLBK_ONCE Callback once.

IM_CALLBK_CONTINUOUS Callback continuously.

period Specifies the interval between callbacks. Either enter any number to indicate the interval in seconds, or choose one of these constants:

IM_CALLBK_SECOND Callback every second.

IM_CALLBK_MINUTE Callback every minute.

IM_CALLBACK_HOUR Callback every hour.

IM_CALLBK_DAY Callback every day.

IM_CALLBK_WEEK Callback every week.

IM_CALLBK_MONTH Callback every month.

IM_CALLBK_QUARTER Callback every quarter.

IM_CALLBK_YEAR Callback every year.

fDisable Specifies whether the callback is enabled or disabled if the application does not control the screen. Use this constant only:

IM_FALSE Enable callback whether or not the application controls the screen.

OUT Parameters *index* Index number for the function in the timer callback database.

Return Value This function returns one of these codes:

IM_SUCCESS Success.

IM_TMCALLBK_TABLE_FULL Timer callback database table is full.

IM_INVALID_TMCALLBK_PERIOD The *period* value is out of range.

IM_INVALID_TMCALLBK_REPETITION The *repeat_count* value is out of range.

Notes You must specify the *index* value when you use im_tm_callback_cancel to remove the function from the timer callback database.

Because the T2090 is a DOS unitasking environment, the application always controls the screen. The application does not release control of the screen until all the callbacks have been completed or until the user terminates the function call by entering Q to quit the loop. Therefore, you set the *fDisable* parameter to IM_FALSE.

See Also im_tm_callback_cancel

Example

```
#include <time.h>
#include <stdio.h>
#include “imt209x.h”

void printHello(void)
{
  im_sound(1000,50,IM_NORMAL_VOLUME);
  im_puts("hello world! ",IM_NORMAL);
}

void main(void)
{

    IM_UCHAR input[300];
    IM_ORIGIN source;
    IM_STATUS iStatus = 11;

    PTIMERCALLBACK pFunction;

    IM_USHORT iIndex = 0;
    time_t timeToStart;
    IM_BOOL fDisplay;
    IM_USHORT iRepeatCount;
    IM_ULONG iPeriod;

    //initialize

    // callback 10 times
    iRepeatCount = 10;

    // 10-second period
    iPeriod = IM_CALLBK_SECOND*10;

    //print hello
    pFunction = printHello;

    //enable callback
    fDisplay = IM_FALSE;

    //time to first callback
    time(&timeToStart);    //get current time

    timeToStart += 16;    //future after 16seconds or
    //timeToStart -= 20;    // or pass current 20 seconds
    //timeToStart = IM_CALLBK_NOW;

    //register the callback function
    iStatus = im_tm_callback_register(pFunction, timeToStart,
                    iRepeatCount, iPeriod, fDisplay, &iIndex);
    //cancel the callback function
    iStatus = im_tm_callback_cancel(iIndex);

}
```

im_transmit_buffer

Purpose This function transmits the contents of a data buffer through the serial communications port. This function continues operating until the buffer transmission is complete or until an error status is detected.

Syntax

```
#include "imt209x.h"
IM_STATUS im_transmit_buffer
    (IM_COM_PORT port_id,
    IM_USHORT length,
    IM_UCHAR far *data_buffer,
    IM_LTIME timeout);
```

IN Parameters *port_id* Identifies the communications port. Use this constant:

IM_COM1 COM1 selected.

length Length of the data string that you want to transmit.

data_buffer Far pointer to the data array that you want to transmit.

timeout Specifies the transmit timeout period. Either enter a number from 0 to 4,294,967.294 to indicate the length of the timeout in milliseconds, or choose one of these constants:

IM_INFINITE_NET_TIMEOUT Never timeout.

IM_ZERO_TIMEOUT No wait.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_SUCCESS Transmit completed.

IM_NET_BAD_DATA Data pointer is null or invalid data length.

IM_TIMEDOUT ACK not received or function timeout expired.

IM_INVALID_PORT Invalid *port_id*.

IM_NET_CONFIG_ERROR Incorrect configuration or hardware fault.

Notes For IM_COM1, a numeric timeout larger than 65534 is converted to 65534. The hardware cannot support long timeout values. The timeout granularity is 55 ms.

Once the transmission begins, program control remains inside this function until the transmission is completed. There is no way to check the transmission status while transmitting.

See Also im_receive_buffer

Example

```
/******************** im_transmit_buffer ***************************/
#include <string.h>
#include <conio.h>
#include "stdio.h"
#include "im209x.h"

void main ( void )
{
   char szTxBuffer[1024];
   IM_STATUS iStatus;

   im_clear_screen();

   strcpy ( szTxBuffer, "MSG_HEADER,Testing Message 1, 2, 3, ..." );

   iStatus = im_transmit_buffer ( IM_COM1, strlen(szTxBuffer), szTxBuffer, 5000 );
   if(iStatus == IM_SUCCESS)
   {
      printf("\nData sent: %s\n", szTxBuffer);
   }
   else
   {
      printf("\nTransmit Buffer Error: ");
      im_message(iStatus);
   }

   getch();
}
```

im_xm_receive_file

Purpose This function receives a file through the serial communications port using XMODEM protocol.

Syntax

```
#include "imt209x.h"
IM_STATUS far im_xm_receive_file
    (IM_CHAR far *file_name
    IM_COM_PORT port_id) ;
```

IN Parameters *port_id* Identifies the communications port. Use this constant:

IM_COM1 COM1 selected.

OUT Parameters *file_name* Identifies the file to be received by the TRAKKER T2090 computer.

Return Value This function returns one of these codes:

IM_SUCCESS Success.

IM_INVALID_PORT The value for *port_id* is invalid. The parameter must be set to IM_COM1 for the T2090 computer.

IM_PARSER_ERROR A parsing error occurred when receiving the file.

Notes The host must be set up to send a file (using XMODEM protocol) to the T2090 computer for this function to execute successfully. Otherwise, this function returns IM_PARSER_ERROR to indicate a communications error. For help, see “Transferring Data” in Chapter 1.

See Also im_xm_transmit_file

im_xm_transmit_file

Purpose This function transmits a file through the serial communications port using XMODEM protocol.

Syntax

```
#include "imt209x.h"
IM_STATUS im_xm_transmit_file
    (IM_CHAR far *file_name,
    IM_COM_PORT port_id) ;
```

IN Parameters *file_name* Identifies the file to be transmitted from the TRAKKER T2090 computer.

port_id Identifies the communications port. Use this constant:

IM_COM1 COM1 selected.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_SUCCESS Success.

IM_INVALID_PORT The value for *port_id* is invalid. The parameter must be set to IM_COM1 for the T2090 computer.

IM_PARSER_ERROR A parsing error occurred when sending the file.

Notes The host must be set up to receive a file (using XMODEM protocol) from the T2090 computer for this function to execute successfully. Otherwise, this function returns IM_PARSER_ERROR to indicate a communications error. For help, see “Transferring Data” in Chapter 1.

im_xm1k_receive_file

Purpose This function receives a file through the serial communications port using XMODEM-1K protocol.

Syntax

```
#include "imt209x.h"
IM_STATUS far im_xm1k_receive_file
    (IM_CHAR far *file_name,
    IM_COM_PORT port_id) ;
```

IN Parameters *port_id* Identifies the communications port. Use this constant:

IM_COM1 COM1 selected.

OUT Parameters *file_name* Identifies the file to be received by the TRAKKER T2090 computer.

Return Value This function returns one of these codes:

IM_SUCCESS Success.

IM_INVALID_PORT The value for *port_id* is invalid. The parameter must be set to IM_COM1 for the T2090 computer.

IM_PARSER_ERROR A parsing error occurred when receiving the file.

Notes The host must be set up to send a file (using XMODEM-1K protocol) to the T2090 computer for this function to execute successfully. Otherwise, this function returns IM_PARSER_ERROR to indicate a communications error. For help, see “Transferring Data” in Chapter 1.

See Also im_xm1k_transmit_file

im_xm1k_transmit_file

Purpose This function transmits a file through the serial communications port using XMODEM-1K protocol.

Syntax

```
#include "imt209x.h"
IM_STATUS im_xm1k_transmit_file
    (IM_CHAR far *file_name,
    IM_COM_PORT port_id) ;
```

IN Parameters *file_name* Identifies the file to be transmitted from the TRAKKER T2090 computer.

port_id Identifies the communications port. Use this constant:

IM_COM1 COM1 selected.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_SUCCESS Success.

IM_INVALID_PORT The value for *port_id* is invalid. The parameter must be set to IM_COM1 for the T2090 computer.

IM_PARSER_ERROR A parsing error occurred when sending the file.

Notes The host must be set up to receive a file (using XMODEM-1K protocol) from the T2090 computer for this function to execute successfully. Otherwise, this function returns IM_PARSER_ERROR to indicate a communications error. For help, see “Transferring Data” in Chapter 1.

See Also im_xm1k_receive_file

im_ym_receive_file

Purpose This function receives a file through the serial communications port using YMODEM protocol.

Syntax

```
#include "imt209x.h"
IM_STATUS far im_ym_receive_file
    (IM_CHAR far *file_name,
    IM_COM_PORT port_id) ;
```

IN Parameters *port_id* Identifies the communications port. Use this constant:

IM_COM1 COM1 selected.

OUT Parameters *file_name* Identifies the file to be received by the TRAKKER T2090 computer.

Return Value This function returns one of these codes:

IM_SUCCESS Success.

IM_INVALID_PORT The value for *port_id* is invalid. The parameter must be set to IM_COM1 for the T2090 computer.

IM_PARSER_ERROR A parsing error occurred when receiving the file.

Notes The host must be set up to send a file (using YMODEM protocol) to the T2090 computer for this function to execute successfully. Otherwise, this function returns IM_PARSER_ERROR to indicate a communications error. For help, see “Transferring Data” in Chapter 1.

See Also im_xm_transmit_file

im_ym_transmit_file

Purpose This function transmits a file through the serial communications port using YMODEM protocol.

Syntax

```
#include "imt209x.h"
IM_STATUS im_ym_transmit_file
    (IM_CHAR far *file_name,
    IM_COM_PORT port_id) ;
```

IN Parameters *file_name* Identifies the file to be transmitted from the TRAKKER T2090 computer.

port_id Identifies the communications port. Use this constant:

IM_COM1 COM1 selected.

OUT Parameters None.

Return Value This function returns one of these codes:

IM_SUCCESS Success.

IM_INVALID_PORT The value for *port_id* is invalid. The parameter must be set to IM_COM1 for the T2090 computer.

IM_PARSER_ERROR A parsing error occurred when sending the file.

Notes The host must be set up to receive a file (using YMODEM protocol) from the T2090 computer for this function to execute successfully. Otherwise, this function returns IM_PARSER_ERROR to indicate a communications error. For help, see “Transferring Data” in Chapter 1.

See Also im_ym_receive_file

5

Reader Command Reference

This chapter describes the reader commands that you can use while operating the computer. Reader commands, such as Backlight On, allow you to perform a task on the computer.

Using Reader Commands

A reader command causes the computer to perform a task. Some reader commands temporarily override the configuration settings and some actually change the configuration settings. For example, you can turn the backlight on to easily view the T2090 computer screen when you are working in a dimly lit environment.

You can execute reader commands by

- scanning a command from a Code 39 bar code label.
- sending a command from a device through the serial port.
- passing the command string from an application using the im_command function.

There are three general types of reader commands:

- Accumulate mode commands
- Operating commands
- File management commands

The reader commands are listed in alphabetical order within these three categories. You will find the purpose, command syntax, and bar code labels for each reader command in this chapter.

Note: *The Code 39 bar code labels in this chapter show an asterisk (*) at the beginning and end of the human-readable interpretation to represent the start and stop codes. If you are creating your own Code 39 bar code labels, your bar code printing utility may automatically supply the asterisks as the start/stop code.*

Using Accumulate Mode

You can use Accumulate mode to collect data from a series of bar code labels and enter them as a single label. When you put the computer in Accumulate mode, the computer will collect all scanned bar code labels in the computer's buffer until you scan either the Enter or Exit Accumulate mode command.

As you accumulate the data from bar code labels, the data is visible on the bottom line of the screen. You can edit the accumulated data with the Backspace and Clear commands.

Backspace This command deletes the last character from the current data record you are accumulating.

Clear This command deletes the entire data record you are accumulating.

Note: *If you are not in Accumulate mode, the Backspace and Clear commands have no effect and you will hear an error beep.*

When you exit Accumulate mode, the accumulated data is "entered" as a data record. Up to 250 characters can be held in the buffer. If the data record count exceeds 250 characters, the data is truncated. If you reset the computer (software or hardware reset), you exit Accumulate mode, the entire buffer is cleared, and all data accumulated is lost.

To use Accumulate mode

The syntax to use the Enter Accumulate command is:

`+/data`

where:

+/ is the syntax for the Enter Accumulate mode command.

data is the optional data you want to enter. *Data* can be a reader command that is executed when you exit Accumulate mode.

1. Scan this bar code label to Enter Accumulate mode:

   Enter Accumulate Mode

   +/

2. Scan the bar code label(s) for the data you want to enter. You can scan labels from the "Full ASCII Charts" in Appendix C.

   For example, scan this label to change the computer's configuration and set the preamble to the characters ABC.

   Change Configuration / Set Preamble to ABC

   $+ADABC

   Or, to edit the accumulated data, scan one of these bar code labels:

   Backspace

   -+

   Clear

   - -

Note: *You can create one bar code label by combining Steps 1 and 2 above. Most of the examples in this manual use one bar code label.*

3. Scan this bar code label to exit Accumulate mode and enter the data record.

Exit Accumulate Mode

-/

Enter Accumulate Mode

Purpose: Enters Accumulate mode. You can accumulate data from a series of bar code labels and enter them as a single label.

From COM Port: Not supported

Scan: Enter Accumulate Mode

+/

Backspace

Purpose: Deletes the last character from the current data record being accumulated. If there is no data in the buffer, the command has no effect.

From COM Port: Not supported

Scan: Backspace

-+

Clear

Purpose: Deletes the entire data record you are accumulating. If there is no data in the buffer, the command has no effect.

From COM Port: Not supported

Scan: Clear

- -

Exit Accumulate Mode

Purpose: Exits Accumulate mode and transmits the current data record. If no data has been accumulated, an empty data record is entered.

From COM Port: Not supported

Scan: Exit Accumulate Mode

-/

Operating Reader Commands

The reader commands you can use to operate or change the computer's configuration are listed in this section. The operating commands are listed in alphabetical order. You will find the purpose, syntax for commands sent from a device connected to the serial port, and bar code labels for these reader commands:

- Backlight On
- Backlight Off
- Change Configuration
- Default Configuration
- Reset Firmware
- Save Configuration to File
- Test and Service Mode

Backlight On

Purpose: Turns the backlight on to easily view the computer screen in dimly lit environments.

From COM Port: `%.1`

Scan: To turn the backlight on, scan this bar code:

Backlight On

%.1

Note: Use the key with the light bulb symbol to toggle the backlight on and off.

Backlight Off

Purpose: Turns the backlight off.

From COM Port: `%.0`

Scan: To turn the backlight off, scan this bar code:

Backlight Off

%.0

Note: Use the key with the light bulb symbol to toggle the backlight on and off.

Change Configuration

Purpose: This command must precede any configuration command. If you enter a valid string, the computer configuration is modified and the computer sounds a high beep. For help on the configuration commands, see Chapter 6.

From COM Port: $+*command*[$+*command*]...[$+*command*]

where *command* is a configuration command with the value you want to set.

Example: Change Configuration / Turn Off Beep Volume

$+BV0

The Change Configuration command is followed by the configuration command to turn off the beep volume (BV0).

Default Configuration

Purpose: Sets the computer to its default configuration and reboots the T2090. The defaults are in effect for the current application only. When you start another PSK application, the T2090 is reset to the configuration stored in the TR2090.CFG configuration file on the C drive.

From COM Port: `.+0`

Scan: To set the default configuration, scan this bar code:

Default Configuration

.+0

Note: To save the current configuration to TR2090.CFG, use the Save Configuration to File reader command.

Reset Firmware

Purpose: Reboots the computer.

From COM Port: `-.`

Scan: Reset Firmware

-.

Save Configuration to File

Purpose: Saves the current runtime configuration to the TR2090.CFG configuration file on the C drive. Each time you start a PSK application, the T2090 is set to the configuration stored in TR2090.CFG.

From COM Port: `.+1`

Scan: Save Configuration to File

.+1

Test and Service Mode

Purpose: Runs the T2090 Setup program, allowing the user to configure the T2090. When the user exits the Setup program, control returns to the program that was active when the command was issued. The display is restored to the state it was in immediately before the command was executed.

From COM Port: `. . - .`

Scan: Test and Service Mode

..-.

File Management Reader Commands

The reader commands you can use to manage files and applications are listed in this section. The file management commands are listed in alphabetical order. You will find the purpose, syntax for commands sent from a device connected to the serial port, and bar code labels for these reader commands:

- Abort Program
- Delete File
- Receive File XMODEM
- Receive File XMODEM-1K
- Receive File YMODEM
- Rename File
- Run Program
- Transmit File XMODEM
- Transmit File XMODEM-1K
- Transmit File YMODEM

Abort Program

Purpose: Aborts or exits the current application, and the computer returns to DOS.

From COM Port: `/$`

Scan: Abort Program

/$

Delete File

Purpose: Deletes a file from a drive on the computer.

From COM Port: `...--drive:filename`

where:

`..--`	is the command to delete a file.
drive:	indicates the drive where you want to delete a file. You must include the colon (:) after the drive letter.
filename	is the file you want to delete.

Scan:

1. Scan this bar code label:

   Enter Accumulate Mode / Delete File

   +/..--

2. Scan the bar code label(s) for the file you want to delete. You can scan labels from the "Full ASCII Charts" in Appendix C. The label must use this format:

   `drive:filename`

3. Scan this bar code label to exit Accumulate mode and delete the file.

   Exit Accumulate Mode

   -/

Or: You can create your own bar code labels to delete files by creating a bar code in this command format:

`...--drive:filename`

Example: To delete the file SHIPPING.EXE from drive C, use this command:

```
...--c:shipping.exe
```

Receive File XMODEM

Purpose: Receives a file from the host computer through the serial port and saves it on the T2090 computer. You must have the T2090 connected to the host through the communications dock. On the host, you need to transmit the file using a serial communications package that supports XMODEM protocol (i.e., Windows 3.1 Terminal or Win95 Hyperterminal) or TRANSFER.EXE.

From COM Port: `.%X1,drive:filename`

where:

`.%X`	is the command to receive a file from a host using XMODEM protocol.
`1`	indicates the T2090 computer's serial port.
drive:	indicates the drive on the T2090 computer where you want to receive and store the file. You must include the colon (:) after the drive letter.
filename	is the file you want to receive and save on the T2090.

Scan:

1. Scan this bar code label:

   Enter Accumulate Mode / Receive File

   +/.%X1,

2. Scan the bar code label(s) for the file you want to receive. You can scan labels from the "Full ASCII Charts" in Appendix C. The label must use this format:

   drive:filename

3. Scan this bar code label to exit Accumulate mode and receive the file.

   Exit Accumulate Mode

   -/

Or: You can create your own bar code labels to receive files by creating a bar code in this command format:

`.%X1,drive:filename`

Example: To receive the file SHIPPING.EXE on the T2090's drive C, use this command:

`.%X1,c:shipping.exe`

Receive File XMODEM-1K

Purpose: Receives a file from the host computer through the serial port and saves it on the T2090 computer. You must have the T2090 connected to the host through the communications dock. On the host, you need to transmit the file using a serial communications package that supports the XMODEM-1K protocol (i.e., Windows 3.1 Terminal or Win95 Hyperterminal) or TRANSFER.EXE.

From COM Port: `.%X1,`*`drive:filename`*

where:

`.%X`	is the command to receive a file from a host using XMODEM protocol.
`1`	indicates the T2090 computer's serial port.
drive:	indicates the drive on the T2090 computer where you want to receive and store the file. You must include the colon (:) after the drive letter.
filename	is the file you want to receive and save on the T2090.

Scan: 1. Scan this bar code label:

Enter Accumulate Mode / Receive File

+/.%X1,

2. Scan the bar code label(s) for the file you want to receive. You can scan labels from the "Full ASCII Charts" in Appendix C. The label must use this format:

   `drive:filename`

3. Scan this bar code label to exit Accumulate mode and receive the file.

Exit Accumulate Mode

-/

Or: You can create your own bar code labels to receive files by creating a bar code in this command format:

`.%X1,`*`drive:filename`*

Example: To receive the file SHIPPING.EXE on the T2090's drive C, use this command:

```
.%X1,c:shipping.exe
```

Receive File YMODEM

Purpose: Receives a file from the host computer through the serial port and saves it on the T2090 computer. You must have the T2090 connected to the host through the communications dock. On the host, you need to transmit the file using a serial communications package that supports the YMODEM protocol (i.e., Windows 3.1 Terminal or Win95 Hyperterminal) or TRANSFER.EXE.

From COM Port: `.%Y1,drive:filename`

where:

`.%Y`	is the command to receive a file from a host using YMODEM protocol.
`1`	indicates the T2090 computer's serial port.
drive:	indicates the drive on the T2090 computer where you want to receive and store the file. You must include the colon (:) after the drive letter.
filename	is the file you want to receive and save on the T2090.

Scan:

1. Scan this bar code label:

   Enter Accumulate Mode / Receive File

   +/.%Y1,

2. Scan the bar code label(s) for the file you want to receive. You can scan labels from the "Full ASCII Charts" in Appendix C. The label must use this format:

   drive:filename

3. Scan this bar code label to exit Accumulate mode and receive the file.

   Exit Accumulate Mode

   -/

Or: You can create your own bar code labels to receive files by creating a bar code in this command format:

`.%Y1,`*`drive:filename`*

Example: To receive the file SHIPPING.EXE on the T2090's drive C, use this command:

```
.%Y1,c:shipping.exe
```

Rename File

Purpose: Renames a file stored on the T2090 computer.

From COM Port: `...-drive:oldfilename,drive:newfilename`

where:

`...-`	is the command to rename a file.
drive:	indicates the drive where the *oldfilename* is stored. You must include the colon (:) after the drive letter.
oldfilename	is the name of the file you want to rename.
drive:	indicates the drive where the *newfilename* is stored. You must include the colon (:) after the drive letter. The drive letter **MUST** match the drive letter you entered for the *oldfilename.*
newfilename	is the new name of the file.

Scan: 1. Scan this bar code label:

Enter Accumulate Mode / Rename File

+/...-

2. Scan the bar code label(s) for the file you want to rename. You can scan labels from the "Full ASCII Charts" in Appendix C. The label must use this format:

   drive:oldfilename,drive:newfilename

3. Scan this bar code label to exit Accumulate mode and rename the file.

Exit Accumulate Mode

-/

Or: You can create your own bar code labels to rename files by creating a bar code in this command format:

`...-`*drive:oldfilename,drive:newfilename*

Example: To rename the file SHIPPING.EXE on drive C to DOCK1.EXE, use this command:

```
...-c:shipping.exe,c:dock1.exe
```

Run Program

Purpose: Runs the specified program or application that is stored on the T2090 computer.

From COM Port: `//drive:filename`

where:

`//`	is the command to run an application.
drive:	indicates the drive where the application is stored. You must include the colon (:) after the drive letter.
filename	is the application you want to run.

Scan:

1. Scan this bar code label:

   Enter Accumulate Mode / Run Program

   +///

2. Scan the bar code label(s) for the application you want to run. You can scan labels from the "Full ASCII Charts" in Appendix C. The label must use this format:

   `drive:filename`

3. Scan this bar code label to exit Accumulate mode and run the application.

   Exit Accumulate Mode

   -/

Or: You can create your own bar code labels to run applications by creating a bar code in this command format:

`//drive:filename`

Example: To run the application SHIPPING.EXE, use this command:

```
//c:shipping.exe
```

Transmit File XMODEM

Purpose: Transmits a file from the T2090 computer through the serial port and saves it on the host computer. You must have the T2090 computer connected to the host through the EZ Dock docking. On the host, you need to receive the file using a serial communications package that supports the XMODEM protocol (i.e., Windows 3.1 Terminal or Win95 Hyperterminal) or TRANSFER.EXE.

From COM Port: %%X1,*drive:filename*

where:

%%X	is the command to transmit a file using XMODEM protocol.
1	indicates the T2090 computer's serial port.
drive:	indicates the drive where the file is stored on the T2090. You must include the colon (:) after the drive letter.
filename	is the file you want to transmit.

Scan:

1. Scan this bar code label:

   Enter Accumulate Mode / Transmit File

   +/%%X1,

2. Scan the bar code label(s) for the file you want to transmit. You can scan labels from the "Full ASCII Charts" in Appendix C. The label must use this format:

   drive:filename

3. Scan this bar code label to exit Accumulate mode and transmit the file.

   Exit Accumulate Mode

   -/

Or: You can create your own bar code labels to transmit files by creating a bar code in this command format:

%%X1,*drive:filename*

Example: To transmit the file SHIPPING.DAT from drive C to the host, use this command:

%%X1,c:shipping.dat

Transmit File XMODEM-1K

Purpose: Transmits a file from the T2090 computer through the serial port and saves it on the host computer. You must have the T2090 computer connected to the host through the EZ Dock docking. On the host, you need to receive the file using a serial communications package that supports the XMODEM-1K protocol (i.e., Windows 3.1 Terminal or Win95 Hyperterminal) or TRANSFER.EXE.

From COM Port: %%X1,*drive:filename*

where:

%%X	is the command to transmit a file using XMODEM protocol.
1	indicates the T2090 computer's serial port.
drive:	indicates the drive where the file is stored on the T2090. You must include the colon (:) after the drive letter.
filename	is the file you want to transmit.

Scan:

1. Scan this bar code label:

   Enter Accumulate Mode / Transmit File

   +/%%X1,

2. Scan the bar code label(s) for the file you want to transmit. You can scan labels from the "Full ASCII Charts" in Appendix C. The label must use this format:

   drive:filename

3. Scan this bar code label to exit Accumulate mode and transmit the file.

   Exit Accumulate Mode

   -/

Or: You can create your own bar code labels to transmit files by creating a bar code in this command format:

%%X1,*drive:filename*

Example: To transmit the file SHIPPING.DAT from drive C to the host, use this command:

%%X1,c:shipping.dat

Transmit File YMODEM

Purpose: Transmits a file from the T2090 computer through the serial port and saves it on the host computer. You must have the T2090 computer connected to the host through the EZ Dock docking. On the host, you need to receive the file using a serial communications package that supports the YMODEM protocol (i.e., Windows 3.1 Terminal or Win95 Hyperterminal) or TRANSFER.EXE.

From COM Port: %%Y1,*drive:filename*

where:

%%Y	is the command to transmit a file using YMODEM protocol.
1	indicates the T2090 computer's serial port.
drive:	indicates the drive where the file is stored on the T2090. You must include the colon (:) after the drive letter.
filename	is the file you want to transmit.

Scan: 1. Scan this bar code label:

Enter Accumulate Mode / Transmit File

+/%%Y1,

2. Scan the bar code label(s) for the file you want to transmit. You can scan labels from the "Full ASCII Charts" in Appendix C. The label must use this format:

   drive:filename

3. Scan this bar code label to exit Accumulate mode and transmit the file.

Exit Accumulate Mode

-/

Or: You can create your own bar code labels to transmit files by creating a bar code in this command format:

%%Y1,*drive:filename*

Example: To transmit the file SHIPPING.DAT from drive C to the host, use this command:

%%Y1,c:shipping.dat

6

Configuration Command Reference

This chapter contains an alphabetical list of all the configuration commands supported on the TRAKKER T2090.

Using Configuration Commands

A configuration command changes the way the T2090 computer operates while running a PSK application. You can execute configuration commands by:

- sending a command from a device connected to the serial port.
- calling an im_command from an application.
- scanning a label.

You can find the following information about each configuration command in this chapter:

- Command description and purpose
- Command syntax and options
- Default setting

The configuration commands are listed alphabetically by command name. For a list of bar code symbology, operations, or communications commands, use the next table, "Configuration Commands Listed by Category."

All configuration commands begin with $+, followed by the specific two-character command and optional parameters.

The configuration that you specify with these commands is distinct from the BIOS level configuration specified with the setup utility. This configuration is active only through an application built with the PSK library.

The configuration is stored in a hidden file on the C drive, TR2090.CFG. This file is referenced only when an application calls a PSK function. If TR2090.CFG is deleted or corrupted, a new file will be created with default values when a PSK function is called.

***Note**: Placeholders may be used in the syntax of some commands to allow compatibility with other programmable Intermec products that have more variables.*

Configuration Commands Listed by Category

The following table lists the configuration commands you may need to set for bar code symbologies, operations, or serial port device communications.

Bar Code Symbologies	Default Setting
Codabar	Disabled
Code 39	Full ASCII Code 39 enabled with no check digit
Code 128	Standard
Interleaved 2 of 5	Disabled
MSI	Disabled
UPC/EAN	Enabled.

Operations	Default Setting
Append Time	Disabled
Automatic Shutoff	Disabled
Beep Volume	Normal
Command Processing	All reader commands enabled
Display Backlight Timeout	10 seconds
Display Contrast	3
Display Font Type	6x9
Keypad clicker	Disabled
Postamble	No characters (disabled)
Preamble	No characters (disabled)
RAM Drive Size	0
Resume Execution	Allowed
Time and Date	920101120000
Time in Seconds	Disabled

Communications	Default Setting
Baud Rate	19200
Configuration Commands Via Serial Port	Enabled without TMF
Data Bits	7
End of Message (EOM)	\x03 (hexadecimal value for ETX)
Flow Control	None
Handshake	Disabled
LRC (Longitudinal Redundancy Check)	Disabled

Parity	Even
Poll (Polling)	Disabled
Start of Message (SOM)	\x02 (hexadecimal value for STX)
Stop Bits	1
Timeout Delay	10 seconds

Entering Variable Data in a Configuration Command

You can enter variable data for many of the configuration commands. For example, you can set a preamble that is up to 25 ASCII characters long. You need to follow these general instructions to enter variable data.

To enter variable data in a configuration command

1. Scan a bar code label with this syntax:

   +/$+*command*

   where:

+/	is the syntax for the Enter Accumulate Mode command.
$+	is the syntax for the Change Configuration command.
command	is the syntax for the command you want to change.

   For example, the command syntax for a preamble is AD*data*. To change or set a preamble, scan this bar code:

   Enter Accumulate Mode / Change Configuration / Set Preamble

   +/$+AD

2. Scan a bar code label from the "Full ASCII Charts" in Appendix C. To set the preamble to the character T, scan this label:

   T

   T

   Note: *To use the bar code labels in Appendix C, you must configure the computer to use Code 39 in Full ASCII mode. For help, see "Code 39" later in this chapter.*

Entering Variable Data in a Configuration Command (continued)

3. Scan the Exit Accumulate Mode bar code label to update the computer's configuration:

Exit Accumulate Mode

-/

Append Time

Purpose: Appends the time to data records that are transmitted from the computer. You can also use the Time in Seconds command to append the time in hours and minutes only, or hours, minutes, and seconds. The time is appended to each data record in the form HH:MM:SS. For help, see "Time in Seconds" later in this chapter.

Syntax: DE*data*

Acceptable values for *data* are:

0	Disabled
1	Enabled

Default: Disabled

Scan: One of these bar codes:

Disable Append Time

$+DE0

Enable Append Time

$+DE1

Automatic Shutoff

Purpose: Defines the maximum length of time the computer remains on when there is no activity. When you do not use the computer for the length of time set with this command, the computer automatically turns off as if you had pressed ⏻ to turn it off.

When you press ⏻ to turn on the computer, the computer either resumes exactly where it was when you turned it off or the computer boots and restarts your application. Resume is controlled through the Resume Execution command. For help, see "Resume Execution" later in this chapter.

Power Management Tip: *You should use the Automatic Shutoff feature to preserve the battery pack's power.*

Syntax: EZ*data*

Acceptable values for *data* are one or two digits:

0	Disabled (always on)
1-15	Shutoff time in minutes

Default: Disabled

Scan: To disable automatic shutoff, scan this bar code:

Disable Automatic Shutoff

$+EZ0

Or: To set a timeout:

1. Scan this bar code:

   Enter Accumulate Mode / Set Automatic Shutoff

   +/$+EZ

2. Scan a numeric value for *data* from these bar codes:

   0

   1

   2

   3

Automatic Shutoff (continued)


4


5


6


7


8


9

3. Scan this bar code:

Exit Accumulate Mode


-/

Baud Rate

Purpose: Sets the baud rate for the serial port on the T2090 computer. The baud rate must match the baud rate of the device (i.e., the host computer) that the T2090 computer is communicating with through the serial port.

Syntax: IA*data*

Acceptable values for *data* are:

3	1200 baud
4	2400 baud
5	4800 baud
6	9600 baud
7	19200 baud
8	38400 baud
9	57600 baud
A	115200 baud

Default: 19200 baud

Scan: One of these bar codes:

1200 Baud


$+IA3

2400 Baud


$+IA4

4800 Baud

$+IA5

9600 Baud

$+IA6

19200 Baud

$+IA7

38400 Baud

$+IA8

57600 Baud

$+IA9

115200 Baud

$+IAA

Example: im_command (“$+IA8”,5);

will set the baud rate to 38400.

Beep Volume

Purpose: Set the beep volume according to operator preference and work environment.

Syntax: BV*data*

Acceptable values for *data* are:

0	Off
2	Normal

Default: Normal

Scan: One of these bar codes:

Beep Volume Off

$+BV0

Beep Volume Normal

$+BV2

Codabar

Purpose: Enables or disables decoding of Codabar symbology. Codabar is a self-checking, discrete symbology. The American Blood Commission (ABC) Codabar requires that you retain and transmit the start / stop code digits when processing a Codabar symbol. Start / stop code digits are always transmitted.

Syntax: CD*data*

Acceptable values for *data* are one or two digits, corresponding to:

First digit	0	Disabled
	2	Enabled
Second digit	0 or 1	Ignored

Default: Disabled

Note: The configuration CD20 is not permitted for consistency with other Intermec products.

Scan: One of these bar codes:

Disabled

$+CD00

Enable ABC, Transmit ABCD Start/Stop

$+CD11

Example: im_command ("$+CD11",6);

will enable codabar.

Code 39

Purpose: Enables or disables decoding of Code 39 symbology. Code 39 is discrete, variable length, and self-checking. The character set is uppercase A to Z, 0 to 9, dollar sign ($), period (.), slash (/), percent (%), space ( ), plus (+), and minus (-). The maximum character length for a label is 23 characters.

The computer decodes three types of ASCII:

- Code 39 non-full ASCII
- Code 39 full ASCII
- Code 39 mixed-full ASCII

Code 39 non-full ASCII Non-full ASCII uses a one-character encoding scheme. For example, you encode the data “SAMPLE” as follows:

SAMPLE

This label decodes as *SAMPLE*.

Code 39 full ASCII Full ASCII uses a two-character encoding scheme to extend the character set to 128 characters. You use the dollar sign ($), slash (/), percent (%), or plus (+) followed by an uppercase letter to represent one of the characters in the extended set. You must encode lowercase letters as a plus sign (+) followed by their uppercase equivalents. For a list of ASCII characters and their Code 39 representations, see the “Full ASCII Charts” in Appendix C.

Use Code 39 full ASCII to enter ASCII control characters or lowercase characters as data. You should also enable Code 39 full ASCII to use ASCII command characters.

For example, you encode the data “sample” in Code 39 full ASCII as follows:

+S+A+M+P+L+E

In Code 39 non-full ASCII, this label decodes as *+S+A+M+P+L+E*. In Code 39 full ASCII, this label decodes as *sample*.

Code 39 mixed-full ASCII Use mixed-full ASCII when printers encode the same label two different ways. For example, if you have a bar code with the data $%a, some printers encode the data as follows:

/D/E+A

Code 39 (continued)

In the "Full ASCII Charts" in Appendix C, /D represents $ and /E represents %. If you configure the computer for Code 39 full ASCII, the computer decodes the data as $%a because there are three valid full ASCII character pairs to represent the data.

Other printers encode the data $%a as:

$%+A

The $ and % are valid Code 39 characters in the non-full ASCII character set. However, the computer will not decode this label if it is configured for full ASCII because the data is not represented by valid full ASCII character pairs. To decode the label correctly, you need to configure the computer for mixed-full ASCII.

When you configure the computer for Code 39 mixed-full ASCII, the computer will decode both of the labels above as $%a.

Mixed-full ASCII interprets any valid full ASCII character pairs that appear in the label, but does not require that all data be encoded with a valid full ASCII character pair. If you are uncertain how your labels are encoded, configure the computer for mixed-full ASCII, which decodes all valid Code 39 labels.

If you configure the computer for Code 39 full ASCII, you should check for Code 39 mixed-full ASCII. Mixed-full ASCII does not apply when you configure the computer for non-full ASCII.

Note: *The interpretive text shown under bar code labels does not always accurately reflect the data that is encoded in the label. The interpretive text represents how the label should be decoded.*

Use this table to help configure your computer.

Code 39 Option	Bar Code Label	Decodes
Non-full ASCII	$%+A	$%+A
	/D/E+A	/D/E+A
Full ASCII	$%+A	No decode
	/D/E+A	$%a
Mixed-full ASCII	$%+A	$%a
	/D/E+A	$%a

Syntax: CB*data*

Acceptable values for *data* must be three digits, corresponding to:

First digit:	0	Disabled
	1	Enabled with no check digit

Second digit:	0	Ignored
	1	Ignored
Third digit:	0	Code 39 non-full ASCII
	1	Code 39 full ASCII
	2	Code 39 mixed-full ASCII

Default: Enable Code 39 Full ASCII with no check digit (111)

Scan: To disable Code 39:

Disable Code 39

$+CB000

To enable Code 39 with non-full ASCII:

Enable Code 39, non-full ASCII

$+CB110

To enable Code 39 with full ASCII:

Enable Code 39; full ASCII

$+CB111

To enable Code 39 with mixed-full ASCII:

Enable Code 39; mixed-full ASCII

$+CB112

Example: im_command (“$+CB110”,7);

will enable code 39 non-full ASCII.

Note: For compatibility with other Intermec products, the second digit must be included. Even though it is ignored, it must be 0 or 1. All digits must be in the valid ranges shown; otherwise, the command is invalid and has no effect.

Code 128

Purpose: Enables or disables decoding of Code 128 symbology. Code 128 is a very high density alphanumeric symbology that supports the extended ASCII character set. It is a variable length, continuous code that uses multiple element widths.

Syntax: CH*data*

Acceptable values for *data* are:

0	Disabled
1	Standard Code 128

Default: Standard Code 128

Scan: One of these bar codes:

Disable Code 128

$+CH0

Enable Standard Code 128

$+CH1

Notes: If you configure Standard Code 128, the computer will not decode Function Code 1 characters in the first position of a bar code label. Any subsequent Function Code 1 characters are translated to the ASCII GS control character as a separator for variable length fields.

UCC/EAN function code 1 extensions are not supported.

Command Processing

Purpose: Command processing allows you to disable or enable reader commands. For example, you can disable the Backlight command.

You may want to disable reader commands to prevent a user from accidentally entering a command, or to use data that would otherwise be treated as a command. Any bar code label that contains the 2- to 4-character commands for Command Processing is treated as a reader command unless the command is disabled.

Syntax: DC*data*

Acceptable values for *data* are:

0	Disable all reader commands
1	Enable all reader commands
2	Disable override

3	Enable override
*command*0	Disable reader command
*command*1	Enable reader command

The override option is a temporary setting that allows you to enable all the reader commands for as long as you need them. When you want to return to the previous configuration, you disable the override.

Note: *The Enable Override option is the only bar code label you can scan to enable reader commands if you have disabled all reader commands (DC0).*

Default: Enable all reader commands

Scan: To enable all the reader commands or override the current settings, scan one of these bar codes:

Disable All Reader Commands

$+DC0

Enable All Reader Commands

$+DC1

Disable Override

$+DC2

Enable Override

$+DC3

Or: To disable or enable specific reader commands, perform these steps:

1. Scan this bar code:

   Enter Accumulate Mode / Command Processing

   + / $+DC

2. Scan the bar code to disable or enable one reader command.

   Abort Program

   * / $*

   Backlight

   %.

   Change Configuration

   $+

   Default/Save Configuration

   .+

   Delete File

   ..--

Command Processing (continued)

Exit Accumulate Mode

-

(continued)

/

Receive File

.%

Rename File

...-

Reset

-.

Run Program

/ /

Test and Service Mode

..-.

Transmit File

%%

3. Scan one of these bar codes:

Disable the Command

0

Enable the Command

1

4. Repeat Steps 2 and 3 to disable or enable another reader command.

   Note: *You can accumulate up to 250 characters in the buffer. If the data accumulated exceeds 250 characters, you will hear an error beep and the computer will reject the last bar code read.*

5. Scan this bar code:

   Exit Accumulate Mode

-/

Or: To disable or enable the ability to scan multiple-read labels, scan one of these bar codes:

Disable Multiple-Read Labels

$+DC 0

Enable Multiple-Read Labels

$+DC 1

Configuration Commands Via Serial Port

Purpose: Allows you to control the data the T2090 computer receives through the serial port. You can set this command to execute reader and configuration commands received through the serial port, or treat all data as data without checking for special command syntax. There are two options:

Disabled All data received through the serial port is treated as data. The computer will not execute reader or configuration commands sent or encoded in the data.

Enabled The computer will check for and execute all reader and configuration commands (i.e., Receive File reader command or Beep Volume change configuration command).

Note: *Before you can enable Configuration Commands Via Serial Port, you must configure the EOM command.*

Syntax: IT*data*

Acceptable values for *data* are:

0 Disabled
2 Enabled

Default: Enabled

Scan: One of these bar codes:

Commands Via Serial Port Disabled

$+IT0

Commands Via Serial Port Enabled

$+IT2

Data Bits

Purpose: Sets the number of data bits the T2090 computer uses when communicating with another device (i.e., host computer) through the serial port.

Syntax: II*data*

Acceptable values for *data* are:

7 7 data bits
8 8 data bits

Default: 7 data bits

Scan: One of these bar codes:

7 Data Bits

$+II7

8 Data Bits

$+II8

Example: im_command ("$+II8",5);

will set the serial port to send and receive 8 data bits.

Display Backlight Timeout

Purpose: Defines the amount of time the backlight remains on. The backlight timeout setting significantly affects the computer's battery life. If you set a longer backlight timeout value, you use the power in the battery pack at a faster rate.

Syntax: DF*data*

Acceptable values for *data* are:

0 Disabled
1 - 15 Timeout in minutes

Default: 10 seconds

Scan: To disable the backlight timeout, scan this bar code:

Disable Backlight Timeout

$+DF00

Or: To set the backlight timeout:

1. Scan this bar code:

   Enter Accumulate Mode / Set Backlight Timeout

   +/$+DF

2. Scan a numeric value (one or two digits) for *data* from these bar codes:

   0

   1

   2

   3

   4

   5

   6

   7

   8

   9

3. Scan this bar code:

   Exit Accumulate Mode

   -/

Display Contrast

Purpose: Defines the contrast (light or dark) of the characters against the computer screen.

Syntax: D*J*data*

Acceptable values for *data* are:

0 - 7	Contrast level
8	Lighten contrast (reduce level by 1; minimum level is 0)
9	Darken contrast (increase level by 1; maximum level is 7)

Default: 3

Scan: One of these bar codes:

0 - Light Display Contrast

$+DJ0

1

$+DJ1

2

$+DJ2

3 - Maximum Display Contrast

$+DJ3

4

$+DJ4

5

$+DJ5

6

$+DJ6

7 - Dark Display Contrast

$+DJ7

Lighter Display Contrast

$+DJ8

Darker Display Contrast

$+DJ9

Display Font Type

Purpose: Selects the type or size of font that is used on the computer screen. You can set a regular size font (6x9), a font with double-height characters (6x18), or a font with double-width and double-height characters (12x18).

Syntax: DT*data*

Acceptable values for *data* are:

0 6 pixels by 9 pixels (6x9) font
1 6 pixels wide by 18 pixels high (6x18) font
2 12 pixels wide by 18 pixels high (12x18) font

Default: 6x9

Scan: One of these bar codes:

Set Display Font Type to 6x9

$+DT0

Set Display Font Type to 6x18

$+DT1

Set Display Font Type to 12x18

$+DT2

End of Message (EOM)

Purpose: Attaches an EOM to the end of a data block to indicate the end of data transmission to and from a computer. When EOM is disabled, the computer communicates in Character mode. When EOM is enabled, the computer communicates in Frame mode.

You must configure a value for EOM before you can set these other serial communications commands:

- Configuration Commands Via Serial Port
- Handshake
- LRC
- Start of Message (SOM)

If EOM is disabled or not set, you need to disable these serial communications commands.

EOM **cannot** equal the same value that is set for SOM. You **cannot** set EOM to any of these values:

- AFF (ACK)
- DLE
- NEG (NAK)
- Poll
- RES (EOT)
- REQ (ENQ)
- SEL
- XOFF
- XON

Syntax: PF*data*

Acceptable values for *data* are one or two ASCII characters.

Default: \x03 (hexadecimal value for ETX)

Scan: To disable EOM, scan this bar code:

Disable EOM

$+PF

Or: To set EOM to one or two ASCII characters:

1. Scan this bar code:

   Enter Accumulate Mode / Set EOM

   +/$+PF

2. Scan one or two bar codes for *data* from the "Full ASCII Charts" in Appendix C.
3. Scan this bar code:

Exit Accumulate Mode

-/

Example: im_command ("$+PF\x03",5);

will set EOM for the serial port to ETX.

Flow Control

Purpose: Regulates the data transmission through the serial port. The T2090 computer is a DTE device.

XON/XOFF response and control specify that the computer responds to and transmits XON/XOFF characters.

Note: *Before you can enable Flow Control with the XON/XOFF options, you must disable EOM.*

Syntax: IL*data*

Acceptable values for *data* are:

0 None
4 XON/XOFF response and control

Default: None

Scan: One of these bar codes:

No Flow Control

$+IL0

XON/XOFF Response and Control

$+IL4

Handshake

Purpose: Enables or disables the handshake event that is an affirmative acknowledge to a message received through the serial port.

Note: *Before you can enable Handshake, you must configure the EOM command. Handshake is also referred to as AFF (affirmative acknowledge) on other Intermec data collection devices.*

Syntax: PG*data*

Acceptable values for *data* are:

No data	Disable handshake
\x06	Enable handshake ($F is the label for \x06)

Default: Disabled

Scan: One of these bar codes:

Disable Handshake

$+PG

Enable Handshake (Set to ACK)

$+PG$F

Example: im_command ("$+PG\x06",5);

will set AFF character to ACK and enable handshaking.

Interleaved 2 of 5

Purpose: Enables or disables decoding of Interleaved 2 of 5 (I 2 of 5) symbology. I 2 of 5 is a high-density, self-checking, continuous numeric symbology. It is mainly used in inventory distribution and the automobile industry.

Syntax: CA*data*

Acceptable values for *data* are:

0	Disabled
2-30	Fixed length (even number only)
98	Case Code (6 or 14 characters) with a check digit

Variable length is not supported.

Default: Disabled

Scan: One of these bar codes:

Disable Interleaved 2 of 5

$+CA0

Enable Interleaved 2 of 5, Case Code (6 or 14)

$+CA98

Or: To set Interleaved 2 of 5 to a fixed length:

1. Scan this bar code:

Enter Accumulate Mode / Set Fixed Length

+ / $+CA

2. Scan a numeric value for *data* from these bar codes. (Use even numbers 2-30 only.)

0

1

2

3

Interleaved 2 of 5 (continued)

4

6

8

3. Scan this bar code to exit:

Exit Accumulate Mode

-/

Example: im_command ("$+CA16",6);

will enable I2 of 5 with a fixed length of 16 characters.

Example: im_command ("$+CA98",6);

will enable I2 of 5 Case Code (either 6 or 14 characters).

Keypad Clicker

Purpose: Enables or disables the keypad clicks. The computer sounds a click each time you press a key or decode a row of a two-dimensional symbology.

Syntax: KC*data*

Acceptable values for *data* are:

0	Disable keypad clicker
1	Enable keypad clicker

Default: Disabled

Scan: One of these bar codes:

Disable Keypad Clicker

$+KC0

Enable Keypad Clicker

$+KC1

LRC (Longitudinal Redundancy Check)

Purpose: The Longitudinal Redundancy Check (LRC) character is an error-checking character that you can append to transmitted and received blocks of data.

Note: *Before you can enable LRC, you must configure the EOM command.*

Syntax: IF*data*

Acceptable values for *data* are:

0 LRC disabled
1 LRC enabled

Default: Disabled

Scan: One of these bar codes:

Disable LRC

$+IF0

Enable LRC

$+IF1

Example: im_command (“$+IF1”,5);

will enable LRC on serial communication.

MSI

Purpose: Enables or disables decoding of MSI symbology. MSI code is similar to Plessey code. MSI code includes a start pattern, data characters, a check digit, and a stop pattern. The check digit is always transmitted.

Syntax: CN*data*

Acceptable values for *data* are exactly two digits, corresponding to:

First digit:	0	Disabled
	1,2, or 3	Enabled - 1 modulus 10 check digit
Second digit:	0 or 1	Ignored

Default: Disabled

Scan: To disable MSI, scan this bar code:

Disable MSI

$+CN01

To enable MSI and transmit a check digit, scan this bar code:

MSI With 1 Modulus 10 Check Digit, Transmit Check Digit

$+CN21

Example: im_command ("$+CN21",6);

will enable MSI.

Parity

Purpose: Sets the parity for the serial port. The T2090 computer uses parity for error checking in data transmissions.

Syntax: IB*data*

Acceptable values for *data* are:

0 No parity
1 Even parity
2 Odd parity

Default: Even parity

Scan: One of these bar codes:

No Parity

$+IB0

Even Parity

$+IB1

Odd Parity

$+IB2

Example: im_command (“$+IB2”,5);

will set parity to odd.

Poll (Polling)

Purpose: Solicits or requests data from a polled device.

Note: *Before you can enable Poll, you must configure the EOM and Handshake commands.*

Syntax: HB*data*

Acceptable values for *data* are:

Disabled	No polling
Enabled	Set to FS (File Separator)

Default: Disabled

Scan: One of these bar codes:

Disable Poll

$+HB

Enable Poll (Set to FS)

$+HB%B

Example: im_command ("$+HB\x1C",5);

will set poll character to FS.

Postamble

Purpose: Sets the postamble that is appended to any data you scan with the computer. Common postambles include cursor controls such as tabs or carriage return line feeds.

Syntax: AE*data*

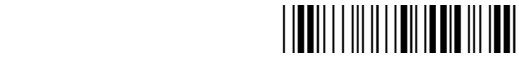
Acceptable values for *data* are up to 25 ASCII characters. If you enter the AE command without *data*, the postamble is disabled. If you are entering quotation marks as data or grouping configuration commands, you need to enclose the *data* within quotation marks (see the example).

Note: *To scan a bar code label that includes quotes, you must configure the computer to use Code 39 in Full ASCII mode. For help, see "Code 39" earlier in this chapter.*

Default: Disabled (no characters)

Scan: To disable the postamble, scan this bar code:

Disable Postamble



$+AE

Or: To set the postamble to an ASCII character string:

1. Scan this bar code:

   Enter Accumulate Mode / Set Postamble

   +/$+AE

2. Scan a value for *data* from the "Full ASCII Charts" in Appendix C. The postamble can be from 1 to 25 characters.

3. Scan this bar code:

   Exit Accumulate Mode

   -/

Example: You want to set a postamble that includes quotation marks. Enter the postamble by scanning this full ASCII bar code label:

Set Postamble to "B"

$+AE""""B""""

You must enclose the data within quotation marks and precede each quotation mark with another quotation mark so that the quotation marks are not treated as the end of the data.

Preamble

Purpose: Sets the preamble that precedes any data you scan with the computer. Common preambles include a data location number or an operator number.

Note: *You can set the preamble to use characters from the extended ASCII character. However, you cannot scan in extended ASCII characters in the Preamble command.*

Syntax: AD*data*

Acceptable values for *data* are up to 25 ASCII characters. When you enter the AD command without *data*, the preamble is disabled. If you are entering quotation marks as data or grouping configuration commands, you need to enclose the *data* within quotation marks (see the example).

Note: *To scan a bar code label that includes quotes, you must configure the computer to use Code 39 in Full ASCII mode. For help, see "Code 39" earlier in this chapter.*

Default: Disabled (no characters)

Scan: To disable the preamble, scan this bar code:

Disable Preamble

$+AD

Or: To set the preamble to an ASCII character string:

1. Scan this bar code:

   Enter Accumulate Mode / Set Preamble

   +/$+AD

2. Scan a value for *data* from the "Full ASCII Charts" in Appendix C. The preamble can be from 1 to 25 characters.
3. Scan this bar code:

   Exit Accumulate Mode

   -/

Example: You want to set a preamble that includes quotation marks. Enter the preamble by scanning this full ASCII bar code label:

Set Preamble to "B"

$+AD""""B""""

You must enclose the data within quotation marks and precede each quotation mark with another quotation mark so that the quotation marks are not treated as the end of the data.

RAM Drive Size

Purpose: Configures the size and use of the RAM drive (E). You can disable the RAM drive and use the additional 256K for programmable (Malloc) memory allocations or configure the RAM drive to temporarily store data and files.

Important: After you set the RAM drive, you must use the Save Configuration to File reader command in Chapter 5 to save the current configuration to TR2090.CFG. Then boot the computer for the change to take effect.

Note: *When you boot or reset the computer, all files on the RAM drive are destroyed.*

Syntax: FR*data*

Acceptable values for *data* are:

0	Disabled, no RAM drive
16-1344	RAM drive size in kilobytes (K)

Default: Disabled

Scan: To disable the RAM drive, scan this bar code:

Disable RAM Drive

$+FR0

Or: To set the RAM drive size:

1. Scan this bar code:

   Enter Accumulate Mode / Set RAM Drive Size

   + / $+FR

RAM Drive Size (continued)

2. Scan a numeric value for *data* from these bar codes:


0


1


2


3


4


5


6


7


8


9

3. Scan this bar code:

Exit Accumulate Mode


-/

4. Scan this bar code to save the configuration change in flash memory:

Save Configuration in Flash Memory


.+1

5. Scan this bar code to boot the computer and use the RAM drive:

Reset Firmware


-.

Resume Execution

Purpose: Defines the way in which the computer resumes when you press I/O to turn on the computer. If you set this parameter to resume not allowed and you press I/O to turn on the computer, the computer will boot and restart the default application. If you set this parameter to resume allowed and press I/O to turn on the computer, the computer resumes exactly where it was when you turned off the computer.

Syntax: ER*data*

Acceptable values for *data* are:

0	Not allowed
1	Allowed

Default: Allowed

Scan: One of these bar codes:

Resume Execution Not Allowed

$+ER0

Resume Execution Allowed

$+ER1

Start of Message (SOM)

Purpose: SOM is the first character in a message sent to or received from the host computer through the T2090 computer's serial port. SOM cannot equal the same value that is set for EOM. You cannot set SOM to any of these values:

- AFF (ACK)
- DLE
- NEG (NAK)
- Poll
- RES (EOT)
- REQ (ENQ)
- SEL
- XOFF
- XON

Note: *Before you can enable SOM, you must configure the EOM command.*

Syntax: PE*data*

An acceptable value for *data* is any ASCII character**.** No data will disable SOM.

Default: \x02 (hexadecimal value for STX)

Scan: To disable SOM, scan this bar code:

Disable SOM

$+PE

Or: To set SOM to an ASCII character:

1. Scan this bar code:

   Enter Accumulate Mode / Set SOM

   +/$+PE

2. Scan a bar code for *data* from the "Full ASCII Charts" in Appendix C.
3. Scan this bar code:

   Exit Accumulate Mode

   -/

Example: im_command ("$+PE\x02",5);

will set SOM to STX.

Stop Bits

Purpose: Sets the number of stop bits on the serial port.

Syntax: IC*data*

Acceptable values for *data* are:

1 1 stop bit
2 2 stop bits

Default: 1 stop bit

Scan: One of these bar codes:

1 Stop Bit

$+IC1

2 Stop Bits

$+IC2

Example: im_command (“$+IC2”,5);

will set serial port to 2 stop bits.

Time and Date

Purpose: Sets the time and date on the computer.

Syntax: DB*data*

Acceptable values for *data* are 12 digits corresponding to:

yy	00-99	Year
mm	01-12	Month of the year
dd	01-31	Day of the month
hh	01-12	Hour
mm	00-59	Minutes
ss	00-59	Seconds

Default: 920101120000

Time and Date (continued)

Scan: To set the time and date:

1. Scan this bar code:

   Enter Accumulate Mode / Set Time and Date

   + / $+DB

2. Scan a numeric value for each digit from these bar codes:

   0

   1

   2

   3

   4

   5

   6

   7

   8

   9

3. Scan this bar code:

   Exit Accumulate Mode

   *- / *

Example: im_command ("$+DB011225010101",16);

will set the date to December 25, 2001

Time in Seconds

Purpose: If you enable the Append Time command, you can enable the Time in Seconds command to append the seconds to each transaction transmitted from the computer. To append the time in hours and minutes, disable the Time in Seconds command.

Syntax: DA*data*

Acceptable values for *data* are:

0	Disabled
1	Enabled

Default: Disabled

Scan: One of these bar codes:

Disable Time in Seconds

$+DA0

Enable Time in Seconds

$+DA1

Timeout Delay

Purpose: If handshaking is enabled, the T2090 computer expects a response to each message that is sent to the host through the serial port. The timeout delay is the amount of time the T2090 computer waits to receive a response. When the timeout expires, the T2090 computer tries sending the message again. If no response is received, a timeout error occurs.

Syntax: IE*data*

Acceptable values for *data* are:

0	5 ms
1	100 ms
2	500 ms
3	2 sec
4	10 sec
5	20 sec
6	40 sec
7	60 sec
8	No timeout

Default: 10 seconds

Scan: One of these bar codes:

No Timeout Delay

$+IE8

Timeout Delay 5 ms

$+IE0

Timeout Delay 100 ms

$+IE1

Timeout Delay 500 ms

$+IE2

Timeout Delay 2 sec

$+IE3

Timeout Delay 10 sec

$+IE4

Timeout Delay 20 sec

$+IE5

Timeout Delay 40 sec

$+IE6

Timeout Delay 60 sec

$+IE7

Example: im_command (“$+IE7”,5);

will set timeout delay to 60 seconds.

UPC/EAN

Purpose: Enables or disables decoding of UPC-A, UPC-E, EAN-8 and EAN-13. When enabled, supplementals are auto-detected and are transmitted. UPC Preamble of System and Country Codes are detected and transmitted. Check digits for UPC-A and UPC-E are detected and transmitted. The 6 digits of UPC-E are expanded into a UPC-A 12-digit code.

Syntax: CE*data*

Acceptable values for *data* are four, five, six, or seven digits. The first digit must be 0, 1, or 2. The other digits must be 0 or 1.

0000 Disabled
0111 Enabled

Default: Enabled

Note If any of the first four digits are non-zero (and in the valid range), UPC/EAN decoding is enabled. If any of the last three digits is included, they are ignored.

All digits must be in the valid ranges. Otherwise, the command is invalid and has no effect. For example, $+CE2010010 enables decoding, but S+CE012111 is invalid because the third digit is 2, so the command has no effect.

Scan: To disable UPC/EAN, scan this bar code:

Disable UPC/EAN

$+CE0000

UPC/EAN (continued)

To enable UPC/EAN, scan this bar code:

Enable UPC/EAN

$+CE1000

Example: im_command ("$+CE1000",8);

will enable UPC/EAN and disallow supplementals.

7

BIOS Support

This chapter describes the BIOS services that are supported for the TRAKKER T2090's keypad, display, and cursor.

Display BIOS Support

BIOS display services are available in INT 10h.

The TRAKKER T2090 display most closely approximates display Mode 6, a B&W CGA graphics mode. The display is 8 lines by 20 characters and supports a single character set that includes 208, 5 x 7 ASCII characters. The screen is 119 wide by 71 pixels high. Multiple screen images or display pages are not supported, only page 0 is available and accommodates text as well as graphics. The blink display attribute is not supported, but inverse video is handled in a special manner (refer to int 10h service 33h). The underline character attribute is NOT supported. The T2090 does support character enlargement (refer to int 10h service 32h). If any of the register values are illegal, the function is not performed.

Service Number	Type	Support Notes
00h	Set video mode	Supported but will only allow you to set the video mode to 6. The screen will be cleared each time this service is called. No registers are used. Calling this mode just clears screen, does not look at any registers.
01h	Set cursor size / Enable / Disable Cursor	Cursor size is not supported. The cursor is a fixed size and shaped as an underscore. enable / disable cursor Inputs: ch = if bit 5 == 1 (20h) cursor is disable else if bit 5 == 0 cursor is enable All other bits are ignored.
02h	Set cursor position	Supported. Inputs: dh = cursor position (row = 0-7) dl = cursor position (col = 0-19)

03h	Read Cursor position	Supported. input: bh = page number (ignored) output: dh = cursor position (row = 0-7) dl = cursor position (col = 0-19) ch = 20h cursor disable 00h cursor enable
04h	Read light-pen position	Not supported.
05h	Set active display page	Not supported only page 0 is available and this is fixed.
06h	Scroll window up	Supported. Graphics are not scrolled. Input: al = # of lines to scroll up (if 0, screen is cleared) bh = Attribute for lines blanked at bottom of the window Bit 2 Bit 1 Bit -------- -------- ------- X 0 0 Is Normal X 0 1 Is Double Wide X 1 0 Is Double Height X 1 1 Double High & Wide 1 Reverse bl = scroll attribute **0** scroll text (no graphics on the display) **1** scroll text (but only scroll text actually written on display. Not Graphics. **2** scroll graphics (Not supported) **3** scroll graphics and text (Not supported) ch = Top Row (0 - 7) dh = Bottom Row (0 - 7)

07h	Scroll window down	Supported. Graphics are not scrolled. Input: al = # of lines to scroll down(if 0, screen is cleared) bh = Attribute for lines blanked at top of the window Bit 2 Bit 1 Bit -------- -------- ------- X 0 0 Is Normal X 0 1 Is Double Wide X 1 0 Is Double Height X 1 1 Double High & Wide 1 Reverse bl = scroll attribute **0** scroll text (no graphics on the display). **1** scroll text (but only scroll text actually written on display, not graphics). **2** scroll graphics (Not supported). **3** scroll graphics and text (Not supported). ch = Top Row (0 - 7) dh = Bottom Row (0 - 7)
08h	Read character and attribute	Supported, but no attribute support.
09h	Write character and attribute at cursor position	Supported, but no attribute support. Inputs: cx = count of characters to do, ignored if zero al = char to write
0Ah	Write character	Supported. Input: cx = count of characters to write al = character to write
0Bh	Set color palette	Not supported.
0Ch	Write pixel dot	Supported. Inputs al= 0 (turn pixel on) 1 (turn pixel off) (cx)= x(col)1-119 (dx)= y(row)1-71

0Dh	Read pixel dot	Not supported.
0Eh	write character as TTY	Supported Inputs: al = character
0Fh	get current video mode	Supported Outputs: ah = Columns on screen Always 20h al = Video Mode: Always 06h bh = Active Page: Always 00h
13h	Write character string	Supported. Displays the string at ES:BP on the display. Attributes are ignored. Inputs: al = contains the display subfunction, **0** = String is characters only, cursor not updated **1** = String is characters only, cursor updated **2** = String char / attrib, cursor not updated **3** = String char / attrib, cursor updated cx = length of string dh = cursor position (row = 0-7) dl = cursor position (col = 0-19) es = segment address for string bp = offset address for string
30h	Set backlight	Intermec Special BIOS Service Inputs: al = 0 backlight off al = 1 backlight on
31h	Adjust Contrast	Intermec Special BIOS Service Inputs: al = 0 Contrast Lighter al = 1 Contrast Darker cx = number of steps (range is 1 - 32, Ignores extra steps)

32h	Enlarge Line(s)	Intermec Special BIOS Service - Enlarges text on entire line. Inputs: al = Bit 1 Bit 0 ----- ----- 0 0 is Normal 0 1 is Double Wide 1 0 is Double Height 1 1 is Double Wide & Double Height dh = start row (0-7) dl = end row (0-7)
33h	Reverse Video	Intermec special BIOS service - Reverse video on entire line Inputs: al = 0 Normal Video al = 1 Reverse Video dh = start row (BIOS is 0-7) dl = end row (BIOS is 0-7)
34H	Display Keyboard Mode Indicator	Intermec special BIOS service Inputs : al = 0 Disable keyboard mode display al = 1 Get keyboard mode display status Outputs for al=1: ah = 0 = Disabled 1 = Enabled dh = Text Row (0-7) dl = Pixel Column (0-104) al = 2 Enable keyboard mode display Inputs for al=2: dh = Text Row (0-7) dl = Pixel Column (0-104)

Scrolling Text Lines with Attributes and Graphics

When scrolling either up or down, and graphics are on the display, the graphics remains stationary relative to the scrolling text as long as the scrolling attribute (BL) in INT 10h services 6 & 7 is set to 1. This allows you to preserve a graphics image while text is scrolled around it. If a text line is scrolled up which contains actual text in the location where the graphics image is positioned, the graphics image will be overwritten.

Line attributes (normal, reverse, Double Wide and / or Double Height) move with the text when scrolling lines either up or down. Please note, if the application specifies line number 8 to be Double Height, only the top half of the line is displayed on line 8.

Superimposing Text over Graphics

When the T2090 RESUMES after SUSPENDING, the LCD restores graphics first, then text. If an application superimposes graphics over text in such a way that individual characters are impacted, the result after a restore may be different than expected. It is recommended that text be superimposed over graphics, not visa versa.

Keyboard BIOS Support

Keyboard BIOS services are available in INT 16h.

All keyboard INT 16h BIOS services are supported. Note the keyboard has no Caps Lock, NUM Lock, Scroll Lock, Alt, or Ctrl keys. Alpha keys are always uppercase. Keys are actuated at initial closure and are repeatable. ENTER, Backlight, left shift, and Function keys (F1 - F3) are not repeatable. Keys beep only (if keyclick is on) at initial closure and when auto repeating.

The ⇧ Key is used as a toggle between Alpha and Numeric mode. This key is reported through keyboard INT 16h function 2 as a left shift key and toggles between being DOWN and UP, i.e. pressing it once, the key is considered DOWN until the user presses it a second time, then it is UP. At reset and boot, it is initialized to UP which corresponds to the 'numeric' mode. When left shift key is DOWN, alpha keys are reported, once the left side or right side is closed. When left shift key is UP, numeric and other center keys (-'.) are reported, left side and right side is ignored. SPACE and BACKSPACE are always active. The Function keys ARE NOT shifted when in ALPHA mode. The backlight key is always a toggle between ON and OFF and is handled by the keyboard scanner.

The **ENTER** key will report through keyboard INT 16 function 2 NUMLOCK when key is DOWN and no NUMLOCK when key is UP for use in the user controlled Laser scanning.

Displaying Keyboard Mode Indicator

If the display of the Keyboard Mode Indicator is enabled it will be redrawn at the end of every INT 10h call (Display software Interrupts) and whenever the ⇧ key is pressed. If anything has been written in the location of the status symbol, it will be overdrawn up to two characters on each end. If it is disabled, the user is responsible for clearing the Indicator on the screen.

Power Management BIOS Support

BIOS power management services are available in INT 15h.

Power / Battery Status

Determining the charging rate status while a unit is docked in a cradle and determining the battery capacity status while a unit is outside of a cradle is done with software interrupt 15h. The following is a description of required inputs and the possible outputs.

Entry:

```
AX = 530A hex
BX = 0001  (all devices)
```

Exit:

```
---- Parms OK ----
   CF = 0 success.
       BH = Docked/not Docked status
             BH == 0 is not docked (corresponds to AC
             off-line)
                   BL has battery status (same as it ever
             was i.e. standard)
                         0 = OK
                         1 = low main battery
                         2 = critical (very low main)
                   battery
                         255 = unknown
             BH == 1, is docked (corresponds to AC online)
                   BL has battery charging status
                         0 = trickle charge (fully charged)
                         1 = charging
                         2 = just started
                         255 = unknown
                         CL = remaining battery life
                         255 = unknown
---- Failed ----
   CF = 1 failure
       AH = 09 unrecognized device id (bx<>0001)
```

Power Management Functions

The functions described in this section set and manage the various levels of power management within the TRAKKER T2090. While several technical issues and software calls are being utilized to accomplish these various power states the simplistic concept to keep in mind is that the power down states increase incrementally as to the amount of characteristics which are turned off. "Doze" turns off the backlight, "sleep" adds slowing down the processor to 25 percent of the normal clock speed and "suspend" turns everything "off," except the RAM.

Set Power Management Times

This will set the PM times when APM is not running

Entry:

```
AX = 5380 hex
BH = 02h
```

Input:

```
DH  =
    0 set Doze Timeout
    1 set Sleep Timeout
    2 set Suspend/Off Timeout
    3 set Power Down mode to Either OFF or Suspend
    4 set Wakeup Activities
    5 set I/O Key Action

when DH == 0 for Doze Timeout,
DL =

    0-8   is * 1/8 of Second, where if == 8, is 1 sec
    9     2 Seconds
    A     4 Seconds
    B     6 Seconds
    C     8 Seconds
    D     10 Seconds
    E     12 Seconds
    F     14 Seconds

when DH == 1 or 2 for Sleep or Suspend/Off Timeout,
    DL =
          0 - F   is * 1 minute

when DH == 3 for Power Down mode to Either OFF or Suspend
    DL =
    0 Power Down Mode is OFF
    1 Power Down Mode is SUSPEND

when DH == 4 for Wakeup Activity Flag
```

```
        DL =
              0 No extra Wakeup activities
              1 Calling INT 10h is a wake activity

    when DH == 5 for I/O Key action
        DL =
              0 Same as Power Down Mode, either OFF or SUSPEND
              1 Always OFF, regardless of Power Down Mode
              2 Ignores any pressing of the I/O key while in ON
or                  SLEEP modes
```

Exit:

```
    CF = 0  success, settings are set
    CF = 1  FAILED
```

Get Power Management Times

This will get the Power Management values

Entry:

```
AX = 5380 hex
BH = 03h
```

Input:

```
    DH  =
        0 get Doze Timeout
        1 get Sleep Timeout
        2 get Suspend/Off Timeout
        3 get Power Down mode to Either OFF or Suspend
        4 get Wakeup Activities
        5 get I/O Key Action
```

Exit:

```
    CF = 0  success
        when DH == 0 for Doze Timeout,
              DL =
                    0-8 is * 1/8 of sec, where if == 8, is 1 sec
                    9     2 Seconds
                    A     4 Seconds
                    B     6 Seconds
                    C     8 Seconds
                    D     10 Seconds
                    E     12 Seconds
                    F     14 Seconds

        when DH == 1 or 2 for Sleep or Suspend/Off Timeout,
              DL =
                    0 - F   is * 1 minute

        when DH == 3 for Power Down mode to Either OFF or Suspend
              DL =
```

```
                0 Power Down Mode is OFF
                1 Power Down Mode is SUSPEND

        when DH == 4 for Wakeup Activity Flag
              DL =
                0 No extra Wakeup activities
                1 Calling INT 10h is a wake activity

        when DH == 5 for I/O Key action
              DL =
                0 Same as Power Down Mode, either OFF or
                  SUSPEND
                1 Always OFF, regardless of Power Down Mode
                2 Ignores any pressing of the I/O key while
                  in ON or SLEEP modes
```

Exit:

```
   CF = 1  FAILED
```

Get Unit ID

The Unit ID is a unique four byte number. INT 15h is used to retrieve it as two words, register AX has the low word and BX has the high word. The FLASH Build Version is the top level build and include the BIOS, ROMDOS and all the extensions as a total released product. The BIOS Build Version is for the BIOS only.

Entry:

```
AX = 5380 Hex
BH = 00
```

Exit:

```
   CF = 0 success.
   AX = least significant word of ID
   BX = most significant word of ID
   CL = FLASH hex build version letter
   CH = FLASH Hex build version number
   DX = BIOS hex build version
```

Controlling the LEDs

This will Turn Off or On LEDs.

Entry:

```
AX = 5380h
BH = 01h
BL  =
```

Data Bits 7 - 2 are reserved and should be set to 0

DB 1	DB 0	LED State
0	0	OFF
0	1	Green
1	0	Red
1	1	Red & Green together

Exit:

```
CF = 0  success
```

Very Low Battery

When a very low battery occurs, the TRAKKER T2090 attempts to go into the SUSPEND state. If the application is writing to the clock or writing to the display, the unit cannot enter the SUSPEND state. This procedure will repeat every second until the SUSPEND mode can be reached. The unit will not exit suspend until the voltage reading for the battery is valid. If the unit is in the OFF state and the boot command is attempted, a check of the battery status is made and if the Very Low Battery state is detected the unit will enter the OFF state rather than continue with the boot process.

System Initialization

This section gives an overview of the mechanisms used to bring the T2090 up to a DOS prompt.

Upon completion of the initial BIOS boot, the BIOS will search the real memory area of C000:0 to F000:0 for valid BIOS extensions. Two such extensions reside in this memory area, the CardTrick (FlashDisk) driver and the loader for ROMDOS. The first will cause the FlashDisk driver to be installed, the second will install INT19h for boot to DOS.

Failsafe Boot

Since it is possible for the developer, through the C:\CONFIG.SYS or AUTOEXEC.BAT, to create a fatal run time condition, we have provided a means to ensure a boot of the unit and by-passing all CONFIG.SYS and AUTOEXEC.BAT and c:\ drive system files. This is known as the Failsafe Boot and will cause the unit to boot up to drive A (read only) but still allow access to drive C. To invoke this mode, press F3 upon reset of the unit, releasing the key when the 'B' diagnostic appears during boot. A password request will appear.

To activate this feature, do the following:

1. While the T2090 is booting, press the F3 key (under the display, the unlabeled key on the right).
2. Release the key after the KEYBOARD FAILED message (very brief) appears on the display. The following screen will be displayed:

```
 #
Enter
Password:
```

3. Press the ⇧ key once to shift the keyboard to alpha mode.
4. Type in the PASSWORD "**BIOS**" followed by **ENTER**. If you type something other than BIOS, a screen with the word WRONG is displayed. No additional retries are allowed; the unit simply continues a normal boot cycle and the fail-safe option screen is not displayed.

   If you enter the correct password, you see this screen:

```
Boot options:
1. Fail-safe Boot.
2. Normal Boot
Choose option:
```

5. Press the ⇧ key again to shift the keyboard back to numeric mode.
6. Press 1 or 2 on the keypad. If you press 1, the unit will boot from ROM and skip the CONFIG.SYS and AUTOEXEC.BAT files on the C drive. If you press 2, the unit will perform a normal boot process.

A fail-safe boot will boot to drive A and prompt the operator for time and date.

FWTSETUP.EXE

Pressing the F1 key during reset will cause invocation of a setup batch file from which the Setup executable will be run.

Operational Note: Invocation of FWTSETUP.EXE will occur in the AUTOEXEC.BAT via a new system variable FWT. Therefore, the developer should be aware of this default environment variable and not use the same name. FWT=ON denotes F1 key was pressed.

Boot Diagnostics

During the Boot process, a number of pieces of information pass the screen. Some are revision identifiers for particular software components, and others are diagnostic checkpoints. These checkpoints are B-0.....9 and appear sequentially during the booting of DOS. Should you experience difficulties, take note of the revisions and the last diagnostic number to appear.

First DOS Boot

Initial run will utilize the default CONFIG.SYS file located on the ROMDISK drive A. This CONFIG.SYS will attempt to locate a user drive (C) CONFIG.SYS and run it, causing a nonfatal warning to occur. The default AUTOEXEC.BAT file located on ROMDISK will then run, testing the existence of a root directory on drive C, if this does not exist, drive C will be formatted, FWTSETUP.EXE will be run, a user CONFIG.SYS file and an AUTOEXEC.BAT file are created on drive C. The initial Path of A:\;C:\ will be set, the current drive changed to C and the newly created AUTOEXEC.BAT run. This will leave the operator at the C:\ prompt with path access to the runtime programs resident on drive A (transfer and remote disk).

Drive C Initialization

During the very first boot cycle, or whenever the user's C drive is corrupted or unformatted, the following screen will appear immediately following the '5' diagnostic during boot. This screen and procedure allow the developer an opportunity to format the C drive.

```
C: drive not found!
1. Format C:
2. Do NOT format

Choose option:
```

Select 1 or 2 according to the operator preference, no ENTER key is required.

The T2090 will begin formatting immediately upon completion of the boot cycle to DOS if 1 is selected. You will be prompted for format confirmation and must respond with either Y or N.

Warm DOS Boot

Once the unit has booted to DOS once, all subsequent boots, will run both the resident CONFIG.SYS (ROMDISK) and user CONFIG.SYS (FLASHDISK), and the resident and user AUTOEXEC.BAT files. This allows application developer flexibility in configuration and boot control for the application.

Forcing a COLD Boot

Force a COLD boot condition by pressing the left corner of the BS (backspace) key, the right corner of the SP (space) key and the I/O (power) key simultaneously.

The RTC (real time clock) will default to a 1-1-92 date, enter the setup routine and reset the time and date. Forcing a cold boot will also reset all the power management and screen characteristics to the factory defaults.

Flash Update Utility

This update program reprograms the lower 448K of the TRAKKER flash without damaging the existing drive C. The program reads only the lower 448K from the target flash file located on the PC, even though that file is 2MB in size.

To run the flash update utility:

1. Place the FLASH UPDATE UTILITY diskette in drive A.
2. Extract the update utility files by running A:\FLSH_UP.EXE from within Windows to a path of your choice. Selecting the root directory will create a \FLSH_UP subdirectory with all of the necessary update program files.
3. Place FLASH IMAGE MASTER diskette in drive A.
4. Extract the target image file to the same directory (that is, \FLSH_UP) as the update utilities by running A:\SW300108.EXE from within Windows (the filename, SW300108.EXE, will change with later versions).
5. Important: reboot the PC to DOS. The update utilities create virtual drives and do not run reliably within Windows, particularly Windows 95.
6. Run UPDATE.EXE (see syntax below) using the /F switch to specify the exact image filename. Specify the appropriate COM port and use 38400 baud rate. 115K may not be reliable with certain versions of DOS on the host PC.

UPDATE PROGRAM SYNTAX:

`update [/F`*`filename`* `] [/COM# ] [/B`*`baud`*`]`
where:

filename is your image file
is a COM port number (default=1, such as COM1)
baud is a valid baud rate (9600, 19200, 38400 . . . or 115)
(default baud rate=115)

The following screen appears:

```
                    FLASH UPDATER PROGRAM
          copyright 1998, Intermec Technologies Corp.

1) Reboot the T2090 and enter Setup by hitting F1 during boot
2) Enter Setup password (default 123) and select LOAD FILE
3) Set the correct Baud and select START LOAD

Remote Drive = G:           Local Drive = C:

Remote Disk v6.22 (Revision 2.50.11) (XT)
Copyright © 1989-1997 Datalight, Inc.
Installed as Drive G: /COM1   /B38400+   /T10

Waiting for drive, Put T2090 in cradle!
If unit wasn't in remote mode, then there may be a delay

Hit a key to Terminate

Looking for Drive G:  . . . . .
```

7. On the T2090, enter setup by first turning the unit off and then depressing the F1 key immediately upon power up.

   If the real-time clock has not been set, you will be asked to set the time. Press F3 to accept the default. The clock will be reset after the load. The setup program may also be entered by typing FWTSETUP.

8. Once in setup, enter the password (default=123) and select LOAD FILE. Set the Baud Rate to 38400 and select Start Load RT.

9. Replace the T2090 in its cradle. The PC will automatically detect the T2090 in remote mode and begin sending files. Once the PC changes screen and is in SERVER mode, then select BACK and QUIT on the T2090. This causes the unit to reboot.

10. Leave the unit in the cradle until all further operations are automatically completed.
11. At the end of the process, the unit reboots automatically. Leave the unit in the cradle until all further operations are automatically completed. Certain versions of software may not reboot automatically. Reboot manually by simultaneously pressing BS, SP, and ⓘ/ⓞ keys.

8

Hardware Specifications

This chapter provides specific information about the TRAKKER T2090 equipment, including mechanical, electrical, and environmental specifications for the T2090 and EZ Dock Docking System.

TRAKKER T2090 Specifications

The following tables provide hardware specifications for the TRAKKER T2090.

Mechanical Specifications

Item	Description
Case Material	High impact thermoplastic, tamper resistant case design non-slip surfaces sealed for dust and moisture
Color	Intermec blue or Intermec gray
Dimensions	3.16” W x 7.38” L x 1.68” H
Ergonomics	Designed for true one-hand operation, either hand, with or without gloves
Weight	10 oz. (including batteries)
Display	8 line by 20 character, temperature compensated, Film Supertwist LCD with user enabled backlight operation
Keypad	Unique 13 key alphanumeric, 3 function keys, tactile and audio feedback
Connectors	Contact RS-485 / RS-232 port and charging
Visual Indicators	Power ON/OFF, Low Battery and GOOD/BAD scanner READ
Audio Indicator	Keyclick - User enabled /disabled using Int 16h

Electrical Specifications

Item	Description
External Power	8vDC @450mA (charging requirement)
Internal Power	Rechargeable NiMH, 2.4v @ 1,200 mAh
Battery Capacity	10-20 hours typical usage including 600 bar code scans
Recharging Time	Less than 4 hours in approved docking station
Power Management:	5 advanced states: On, Doze, Sleep, Suspend, and Off

Environmental Specifications

Item	Description
Operating temperature	-4° to +122° F (-20° to +50° C)
Storage temperature	-4° to +158° F (-20° to +70° C)
Humidity	5 to 95 percent
Drop / Shock	Survives multiple, four-foot drops to concrete
Vibration	5-50 Hz @ 1 Grms, 50-300 Hz @ 0.5 Grms

Bar Code Scanner Specifications

Item	Description
Operation	Keyboard trigger for TRUE one hand operation for either hand
Indicator	Displayed on LCD screen and a GOOD/BAD scanner READ LED, with or without audio
Emission Class	CDRH Class II
Emission Duration	Greater than 0.25 seconds
Emission Limit	Less than 0.001 W
Scan Range	1 to 36 inches
Symbologies	Standard width modulated bar codes

Processing Specifications

Item	Description
Microprocessor	NEC V30, 32 MHz
Flash ROM	2 MB for DOS, BIOS and Flash File, upgradable to 4 MB
RAM	2 MB
Real Time Clock	Internal, 1 second resolution
Operating System	DOS 6.22 compatible

EZ Dock Specifications

The following tables provide specifications for the EZ Dock Docking System.

Power Module Mechanical Specifications

Item	Description
Case Material	High impact thermoplastic
Color	Intermec blue or Intermec gray
Dimensions	5.40” W x 5.40” L x 5.70” H
Weight	19.5 oz.
Connectors	(2) RJ45 connectors for data Power in is a mini Molex Contact RS-232/485 port and charging
Visual Indicators	Power ON/OFF on front

2-Slot Cradle Mechanical Specifications

Item	Description
Case Material	High impact thermoplastic
Color	Intermec blue or Intermec gray
Dimensions	4.00” W x 5.40” L x 5.70” H
Weight	13.0 oz.
Contacts	Contact RS-232/485 port and charging

Docking System Environmental Specifications

Item	Description
Operating temperature	-4° to +122° F (-20° to +50° C)
Storage temperature	-4° to +158° F (-20° to +70° C
Ambient temperature	32° to +104° F (0° to +40° C)
Humidity	5 to 95 percent
Vibration	5-50 Hz @ 1 Grms, 50-300 Hz @ 0.5 Grms

RS-232 and RS-485 Communications Configurations

Serial communications can be configured for either RS-232 or RS-485 protocols.

On the 2-slot cradle, there are two resident RJ45 connectors, which are used to make hardwire connections to host systems and other peripheral equipment.

For single-cradle configurations, two types of connections can be made: dedicated mode or multidrop.

- In multidrop mode, the user would communicate with both hand-held units at the same time whether RS-232 or RS-485 by connecting a single COM cable to the innermost RJ45 connector.
- For dedicated configurations the user can communicate to each of the two hand-helds individually by connecting two cables and by activating a small switch located on the inside panel of the 2-slot cradle. This will separate the signals for the two-cradle positions.

The switch is in the UP position for multidrop communications and DOWN for dedicated configurations.

The dedicated option is not configurable for multi-cradle applications.

Pin	RS-232 Signal	Description	RS-485 Signal
1	N/C	No Connect	N/C
2	N/C	No Connect	N/C
3	GND	Ground	GND
4	RxD	Serial Data	Data -
5	TxD	Serial Data	Data +
6	V_{IN}	Power IN	V_{IN}
7	N/C	No Connect	N/C
8	N/C	No Connect	N/C

These signals are transmitted through the power module and 2-slot cradle assemblies from the RJ45 connectors positioned in the unit.

Cabling Information

Intermec makes available seven different cables to assist the developer in building applications :

590819 RS-232 power module to PC only (9F pin, 12')

590818 RS-232 power module to PC or modem (25 pin, 12')

590816 RS-485 power module to PC (25F pin, 8')

598485 RS-485 power module to PC (25F pin, 50')

598486 RS-485 power module to PC (25F pin, 100')

590809 RS-485 termination cable (0-50')

590484 RS-485 termination cable (51-100')

Power Module connectors are all RJ45 type. Pin outs for individual cables are available from Intermec Technical Support. To comply with FCC regulations, the cables must be shielded.

Power Supply Information

There are two power supplies available through Intermec. Both models will operate on input voltages of 100 to 250 volts AC and have outputs of +9 volts DC. Model # **590822** is designed to power up to two cradle configurations and has a output rating of 1.9 amps, the # **590817** model has the same output voltage but output amperage of 5.0 amps and is designed to power up to five cradle configurations.

Power supplies are UL, CE, VDE, and CSA listed.

# of 2-Bay Cradles	Input Amps	Power Supply
1	0.9	590822
2	1.8	9v 1.9A
3	2.7	590817
4	3.6	9v 5.0A
5	4.5	

A

Demo Software

Preparing to Run the Demo

The demo program (SAMPLE.EXE) is loaded on the T2090 at the factory and the default AUTOEXEC.BAT file runs the demo each time the T2090 is booted. If desired, you can edit the AUTOEXEC.BAT file to remove this command. To run the demo, type SAMPLE at the DOS prompt.

The following tables describe the files used to run the demo on the T2090.

Files needed to run the demo

Filename	Usage
SAMPLE.EXE	The demonstration application.
TO-HOST.BAT	A batch file called by the executable to control data communications with the host PC.
TRANSFER.EXE	A XMODEM protocol file transfer program supplied by DataLight specifically for the T2090. The utility will wait for the T2090 to be placed in the dock.
GOTO1-1.EXE	A batch utility that (on the T2090) sends the cursor to position 1,1 on the display.
SWAP.BAT	A batch file used to determine the contents of TO-HOST.BAT.
TO-HOST.TAB	One of the possible sets of instructions for SWAP.BAT to place in TO-HOST.BAT.
NO-HOST.TAB	One of the possible sets of instructions for SWAP.BAT to place in TO-HOST.BAT.
AUTOEXEC.BAT	May call the demonstration application. The Demo can also be invoked from the command line if it is in the current path.
POSTXMIT.BAT	File used to determine what will happen to the data on the T2090 after it has been copied to the host PC. Depending on the desired action, copy one of the following files from the Intermec\T2090\Sample directory to the T2090 and rename it to POSTXMIT.BAT.

Filename	Desired Action
POSTXMIT.APP	Append the next series of data to its existing file.
POSTXMIT.SAV	Save the existing file as LASTDATA.DAT and start a new DEMODATA.DAT
POSTXMIT.DEL	Delete the existing file and start a new DEMODATA.DAT.

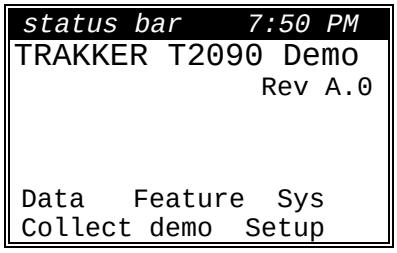
DEMODATA.DAT	The collected scan data. This file is purged if it is in the vicinity of .5 Mb when the program is begun. It may also be removed via the DEL command from the command line, by a keystroke in the MDCD Data Review Demo, or by the POSTXMIT.BAT file transmitted from the Host PC. Only data from the MDCD portion of the demo is stored here.
LOGO.COM	Displays the logo.
DOCKTEST.COM	Used to detect when the T2090 computer is placed in the dock.

Notes

- The ⇧ key toggles the keyboard state between alpha and numeric modes.
- The T2090 detects the cradle and power off conditions during scanning, and will report that these occur on the scanning screen. Control is returned to the screen prior to the **ENTER to Scan** message after the message has been acknowledged. To avoid this, do not turn the unit off or place it in the cradle during scanning. Also, do not attempt to scan while in the cradle.
- If the scanned data file is not open, the program will run the `TO-HOST.BAT` file when docking is detected.

Main Screen

On the Main screen the **F1** key starts the Data Collection Demo, the **F2** key the T2090 Feature Demo, and the F3 key runs the System Setup Menu. A version number is displayed on the second line.



Data Collection Demo

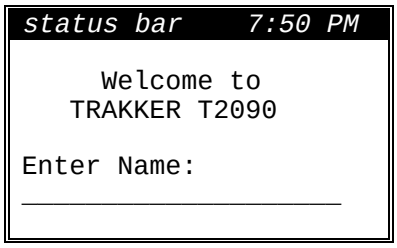
The Status Bar

All screens have a status bar across the top of the display. This status bar shows the following:

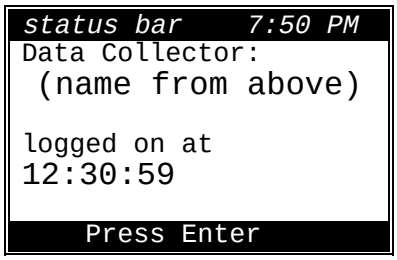
- current keyboard state (ALPHA or NUM)
- battery Icon indicating charge or charging levels
- charging indicator, which is either an animated sine wave (when charging) or a + sign (when not charging)
- the current time

Welcome Screen

The first screen of the Data Collection demo is the Welcome screen. This screen **requires** input from the keyboard to enter a user name. The name is finished using the <enter> key. No function keys are active, but the **SP** and **BS** keys are. The maximum name length allowed in the demo is one line. Remember that the default state for the keyboard is numeric, press the ⇧ key to toggle to 'alpha' mode.

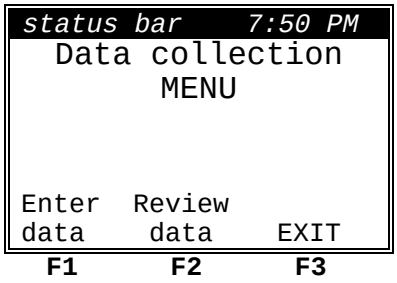


The Welcome screen is followed by the Login response screen. Press **ENTER** to continue.



Data Collection Menu

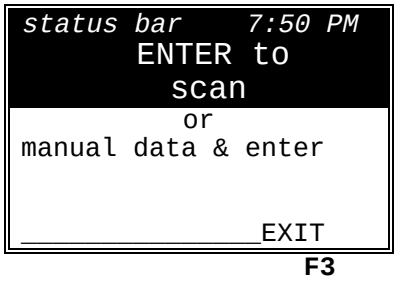
On this menu, only the three oval function keys below the display are active. F1 starts the data entry routine, F2 starts the Data review routine, and F3 Exits the Data Collection demo, going back to the Main screen.



Enter Data Screen

This allows data to be either scanned or manually entered into the log file DEMODATA.DAT. If data is entered from the keyboard prior to pressing <enter>, that data goes into the log file. If no data is entered, the scanner is started up and attempts to scan a code. The F3 key EXITS the screen and control returns to the Data Collection Menu.

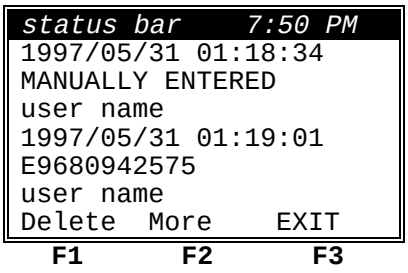
Pressing the **ENTER** key starts the scanner. If a legal bar code is scanned, it is displayed for 2 seconds. If no legal code is scanned, nothing is displayed or recorded. A disfigured (and therefore unscannable) code can be entered through the keypad. A maximum of 20 characters is accepted for manual data. The last few characters may overwrite the EXIT label for the F3 key. All keypad entries, regardless of format, are accepted and recorded, but are not displayed for the 2 second interval as the scanned data is.



Review Data Screen

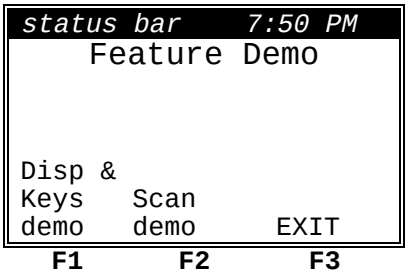
This screen allows the data entered into the log file to be examined. Each record takes up three lines on the screen - first a date line, then a data line and finally the user name for the user who entered the data. The date is in a formatted as (CCYY/MM/DD HH:MM:SS).. If the end of the file has not been reached, F2

may be used to read more data. F3 can always be used to stop reading the file and return to the Data Collection Demo Menu. As an additional indicator that the end of the file has been reached, an `<< End of File >>` message is displayed after the last record. The F1 key will delete the file without any further keystrokes. The following example is from mid-file.



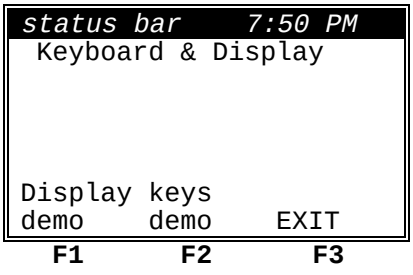
Feature Demo

The Feature demo showcases some features of the T2090 Hand-Held computer. F1 invokes the Display & Keys Demo Menu. F2 starts the Scan demo, and F3 returns to the Main Menu.



Display & Keys Demo Menu

This allows a choice between the Display demo (invoked via the F1 key), the Keyboard demo (invoked via the F2 key), and returning to the Feature Demo Menu (via F3).



Display Demo

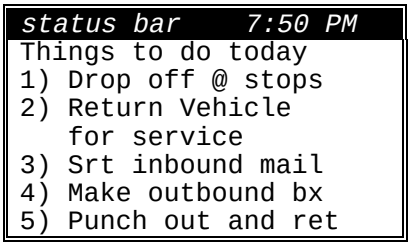
This simulates a typical PDA application, putting up a to-do list . It then manipulates the display image to illustrate some possibilities. The demonstration runs about three minutes, with the following effects:

- The lines are individually reversed starting at the top and working down.
- The lines are unreversed, starting at the bottom and working up.
- The last line is flashed.
- Each line is run through Double Hi and Double Wide modes, starting at the bottom and working up.
- The last line is scrolled to the left, showing a message length of 55 characters in its entirety.
- The last line is scrolled to the right, showing a message length of 55 characters in its entirety.

Notes:

- *Flashing is done by reversing and unreversing the line. The Blink attribute isn't supported.*
- *Horizontal scrolling of the last line is done by replotting the line.*
- *There currently is no BIOS supported horizontal scroll. Because it is the last line, it is important not to place a character in column 20, or the whole screen would scroll upwards.*

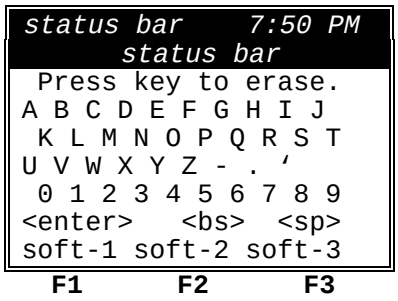
Pressing any key terminates the demo. At the completion of the demo, control returns to the Display and Keys demo Menu.



Keyboard Demo

The Keyboard Demo allows the user to try out each key. This allows a user to get the feel of the keyboard, and become used to the effects of the ⇧ key on both the status line and the characters produced by the keystrokes. A display containing the names of all the keys is presented, and the names can be deleted by pressing the appropriate key. This rule is changed for F3 - the first time it is pressed, its label changes from soft-3 to EXIT. If it is pressed again, the test terminates and control returns to the Keyboard and Display Demo Menu.

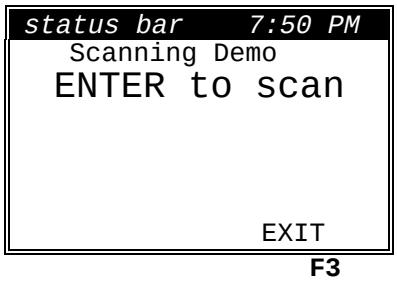
***Note**: This demonstrates how labels on the screen can be changed when conditions change - either the label can be removed, indicating that the function isn't available, or the actual label and the function a key performs can be changed.*



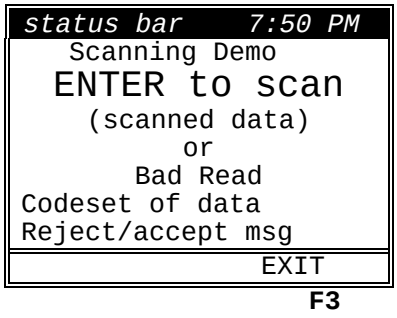
Scan Demo

The scan demo is operated by activating the scanner (using the <enter> key) and attempting to scan bar coded data. The code set for successful scans is reported along with the scanned data. Pressing F3 returns control to the Feature Demo menu.

This scan demo has the BEEP_GOOD and BUZZ_BAD options set.



If the data is scannable, the scanned data, its code set, and a message regarding the code set's acceptability are displayed. If the scan fails, lines 2, 3 and 4 are replaced with a BAD READ message. In either case, another code can be scanned by pressing **ENTER**.



System Setup

***Note**: This section of the program illustrates a number of approaches to menu systems. Some combination of effects may be useful in your programs.*

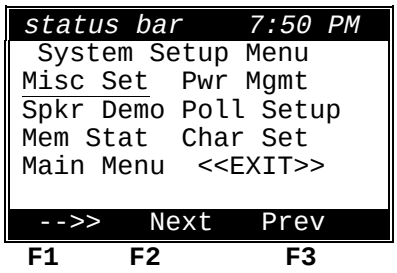
The System Setup screen is reached via the F3 key on the Main menu It has six sub options and two possible exits. The exits are on the last row of the menu:

- Main Menu (Back to the Initial Menu)
- Exit (to DOS).

The sub options are:

- Misc Set — Miscellaneous Settings
- Spkr Demo — Speaker demonstration
- Mem Stat — Status of the T2090 Memory and disk
- Pwr Mgmt — Power Management
- Poll Setup — Selection of content of the `TO-HOST.BAT` file
- Char Set — A display of the character set native to the LCD

They are arranged in two columns as follows:



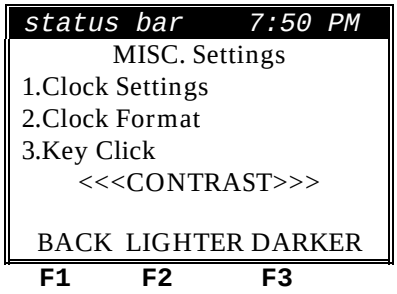
A cursor is placed beneath the current option *(Misc Set on initial entry).*

The Function keys move the cursor, and the ENTER key is used as an EXECUTE key. F1 changes the current column and the direction of the arrow over it (the `-->>` becomes `<<--`) to indicate that a press will change the position and the indicator back to the other column. F2 moves down within the column one entry at a time, stepping to the next column when the end of a column is reached. F3 moves up one entry in a similar fashion. No other keys are active. Since the Function keys do not auto-repeat, reaching the EXIT requires at least four keystrokes in this menu design.

Miscellaneous Settings

This screen is for miscellaneous Settings. There are four settings: Clock, Clock Format, Key Click and display contrast.

Keys and their actions for MISC. Settings Screen:



- **1** enters “Clock Settings” screen
- **2** toggles the clock format between 24hr and AM/PM modes. (Due to space considerations, the AM/PM mode does not have a seconds display). The current mode is reflected in the status bar, as well as the Clock Setting menu.
- **3** is used to turn Key Click on and off.
- **F1** “BACK” key returns to the main menu
- **F2** “LIGHTER” lightens the display contrast

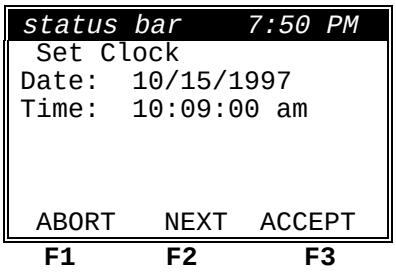
- **F3** "DARKER" darkens the display contrast

***Note**: The current state of Key Click cannot be detected by application programs. The default state for Key Click is OFF. For this reason, the first time Key Click is selected in a given instance of the menu, Key Click is turned ON and a label is displayed to this effect. The second time, Key Click is turned OFF and the message is changed to reflect this. Since Key Click has to be ON to be turned OFF, turning it OFF causes a click (Actually a tone) while turning it on doesn't.*

The Set Clock Screen has seven entry fields, month, day, year, hour, minutes, seconds and am / pm. (Exception: If the time format is 24 hour, the AM/PM field is not displayed, and therefore cannot be entered.)

Entering new data in a field overwrites the previous data and when the field is full, it automatically tabs to the next field, and wraps up to the first field. If illegal data has been entered, and error message is displayed and the cursor is placed on the first illegal field.

Keys and their actions for Set Clock Screen:



- **1 - 0** for all fields except am / pm, where only '**A**' and '**P**' are legal.
- **F1** "ABORT" key returns to the main menu without saving new values
- **F2** "NEXT" key tabs to the next field.
- **F3** "ACCEPT" or '**ENTER**' keys sets new values then returns to the main menu

Speaker Demonstration

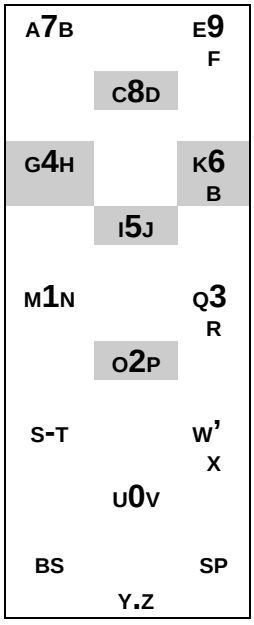
The speaker demonstration shows the variety of sounds that can be easily generated by the speaker. The Function keys are used to select a sound, which is then played by pressing the ENTER key. The selection <EXIT> is not a sound, but is actually the way back to the System Setup Menu.

This screen is arranged in three columns as follows.

```
status bar    7:50 PM
     Speaker Demo
Beep   Buzz   Pipe
Fog    Clicks Siren
Cuckoo 2500   4000
1500   3000   4500
2000   3500   <EXIT>
 -->>   Next   Prev
 F1     F2     F3
```

***Note**: On this menu, the alphanumeric keypad has also been enabled to drive the cursor around and execute selections. Both the alphabetic and numeric values of a given key have the same effect. This allows the Autorepeat function of the keypad to be used to speed up selection.*

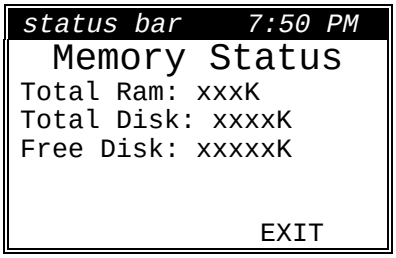
The following keyboard illustration show the uses of the keys. Only the tinted keys are used for this example. Other possibilities include using 7, 9, 1, and 3 as 'go to the corner' keys.



7 and 9 keys are not used.
8 key is UP (same as Prev key).
4 key is Left , 6 is Right.
5 key is Execute (same as ENTER key).
1 and 3 keys are inactive.
2 key is Down (same as Next key).
These keys are inactive.
These keys are inactive.
These keys are inactive.
These keys are inactive.

Memory Status

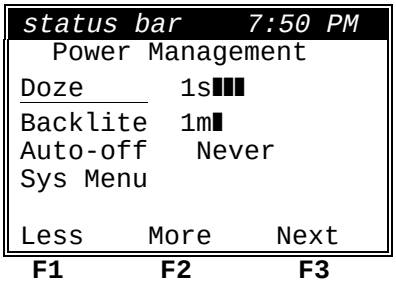
The Memory status screen shows the total Ram, Disk, and available disk space in Kilobytes. The EXIT key returns to the Main Menu. Available Disk space will vary, depending on how big the data file DEMODATA.DAT is and whether previously polled data has been retained. (See the section on HOST software.) Space remaining is also changed by the presence of other software and data files.



Power Management

This screen allows setting of the Power Management timeouts. The values are graphically illustrated by 'slider' bars, which are adjusted on the selected field through the **Less** and **More** keys. The **Next** key allows selection of different values to change, as well as selecting the Sys Menu selection, which returns control to the System Menu.

When Sys Menu is selected, the **Less** and **More** labels become **Abort** and **Save.** One of these keys must be pressed to exit. When a timer value gets to 0, the message **Never** is displayed in place of the value indicator and slider bar. Also, the **Less** label is suppressed. When a value reaches its maximum, the **More** label is suppressed. Note that the first 7 values for Doze are 1/8 seconds, and move the slider bar much less than subsequent steps.



On this menu, the alphanumeric keypad has also been enabled to drive the cursor around and adjust selections. Both the alphabetic and numeric values of a given key have the same effect. This allows the Autorepeat function of the keypad to be used to speed up selection and adjusting of values. The keypad does not have the ability to select an option for exit to the System Setup Menu, however. The Abort/Save choice must be made using the function keys. This helps to avoid accidental selections of the wrong values. The following keyboard illustration shows the uses of the keys. Only the tinted keys are used for this example.

A7B E9F C8D	7 and 9 keys are not used. 8 key is UP. (No Function key does this. It is an advantage of using the keypad.)
G4H K6B I5J	4 key is Less , 6 is More. This key is inactive.
M1N Q3R O2P	1 and 3 keys are inactive. 2 key is Down (same as Next key).
S-T W'X U0V	These keys are inactive. These keys are inactive.
BS SP Y.Z	These keys are inactive. These keys are inactive.

Poll Setup

This allows setting the TO-HOST.BAT file to have the effect of Polling or Not Polling when docking is detected. The screen is as follows:

```
The demo can be set
to be Polled for
the collected data
or Not Polled.
(P)olled or (N)ot?
[P,N]?
```

This must be answered with a P or N keystroke. Doing the setup for this takes several seconds, but eventually control is returned to the System Setup Menu.

Note: This is done by doing a system() call for the batch file SWAP.BAT. Changing the content of the batch file allows (almost) anything to be done. For example, SWAP.BAT

does a CLS, which causes the status line to be erased. It does not re-appear until the System Setup Menu is re-entered. Several useful batch file tools are installed on the A: drive for your convenience.

Host Polling of Data

The Demo has the ability to communicate in a rudimentary fashion with a host PC. This is accomplished by running the file server utility on the host. This utility is located in the Intermec\Tools\FileServer folder after the installation. Start it from your start menu by selecting "Intermec File Copy" from the Intermec programs Group.

Once started, on the page selected with the Communications Options tab, select your communications port and the filename to send the file to on the host. The append/overwrite radio buttons are meaningless on the page, as any file copied up using XMODEM will replace other files of the same name.

On the Serial Port Setup page, set the baud rate to 9600 and press the Run button. The server is now set to receive a file any time a T2090 is dropped into a dock.

After the file has been received, it can be imported into a spreadsheet for manipulation, or processed by some other program.

The records in the file are arranged as follows:

- All of the record is a single line of ASCII, terminated with a <CR><LF> pair
- Each field of the file is wrapped in double quotes
- The fields are separated by commas
- The fields are Name, Date (ccyy/mm/dd hh:mm:ss), a data entry (Just the data portion of the scan, or the manual entry) and source (MANUAL or SCAN).

Transmission will be attempted by the T2090 when docking is first detected *if* there is a file to transmit *and* the T2090 has been configured for Polling (see the section on the System Setup).

Notes:

- *The polling is controlled on the T2090 by the batch file TO-HOST.BAT. The described actions expect that you haven't changed this file. Should you wish to change it, keep your messages to a single line to avoid scrolling whichever screen is current on the T2090 when it is docked. The utility GOTO1-1.COM is supplied to direct the cursor to the upper left corner of the screen between steps of you r batch file.*

- *The actions on the file after polling are controlled by the batch file on the T2090. This is named POSTXMIT.BAT.*

The following files may be copied to POSTXMIT.BAT depending on the desired post transmit action.

POSTXMIT.DEL - deletes the data file.
POSTXMIT.SAV - save the data file in LASTDATA.DAT, destroying any existing LASTDATA.DAT file in the process.
POSTXMIT.APP - actually does nothing, so the T2090 appends to the existing data file the next time data is collected.

By default, POSTXMIT.BAT leaves the file for more data to be appended.

Files are transferred via XMODEM protocol. A program on the PC can substitute for File Server when configured to receive a file with XMODEM at 9600 baud.

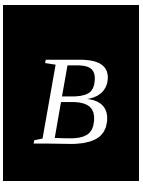
Character Set

This displays the various characters available through the ROM on the LCD display. The characters are displayed 32 to the page across 8 pages. Use any key to advance a page (the Function keys go two pages). The first 16 characters are used as buffers, and aren't consistent. There is a hole in the character set - entries 80h through 9Fh always display as blanks. 20h through 7Dh conform to ASCII. (The ~ and DEL characters 7Eh and 7Fh are displayed as Right and Left pointing arrows.)

This is the second of 8 screens.

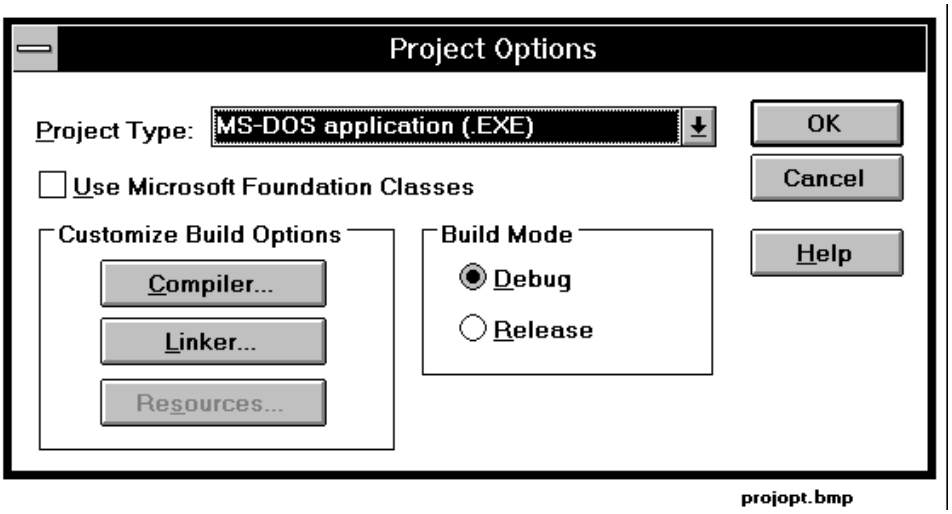
```
20:  28:( 30:0 38:8
21:! 29:) 31:1 39:9
22:" 2a:* 32:2 3a::
23:# 2b:+ 33:3 3b:;
24:$ 2c:, 34:4 3c:<
25:% 2d:- 35:5 3d:=
26:& 2e:. 36:6 3e:>
27:' 2f:/ 37:7 3f:?
```

Note: *This character set does not match the standard DOS code page 850. For help, contact Intermec Technical Support.*



Microsoft Visual C/C++ Settings

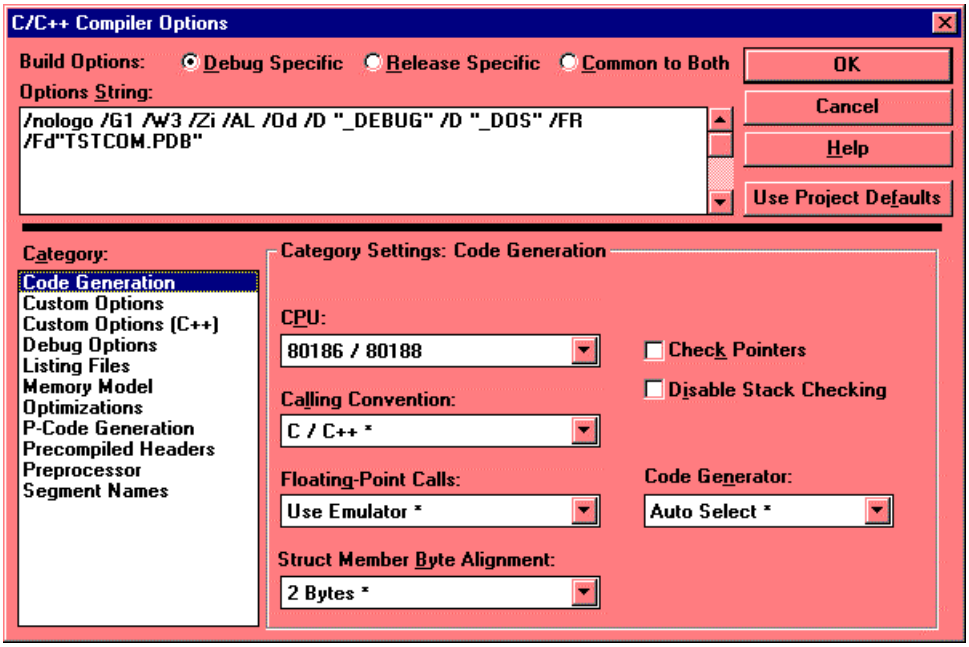
Project Options



***Note**: This example uses Microsoft Visual C/C++, Professional Edition v1.5. Your screen may look different. The PSK requires Microsoft Visual C/C++, Professional Edition v1.0 or v1.5x, which can create 16-bit DOS applications.*

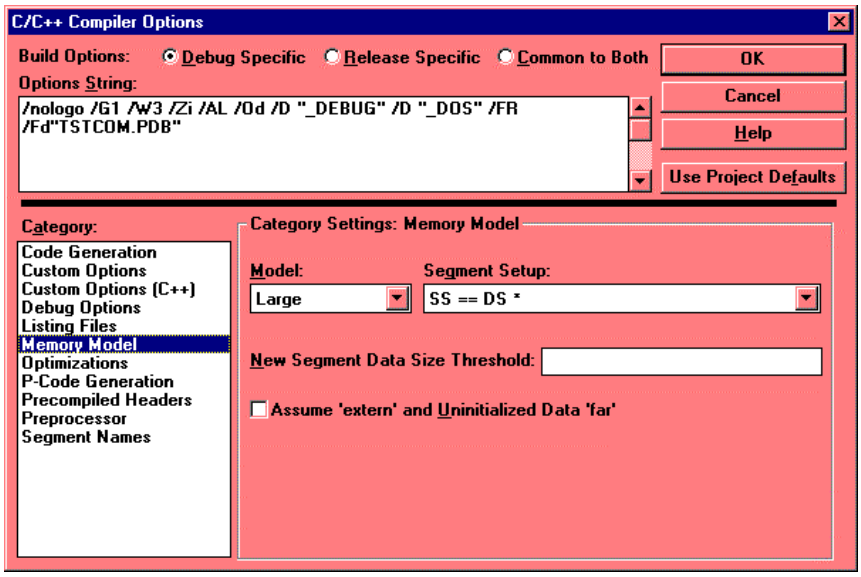
Select Project Type MS-DOS application (.EXE).

Compiler Options: Code Generation



Select CPU 80186/80188.

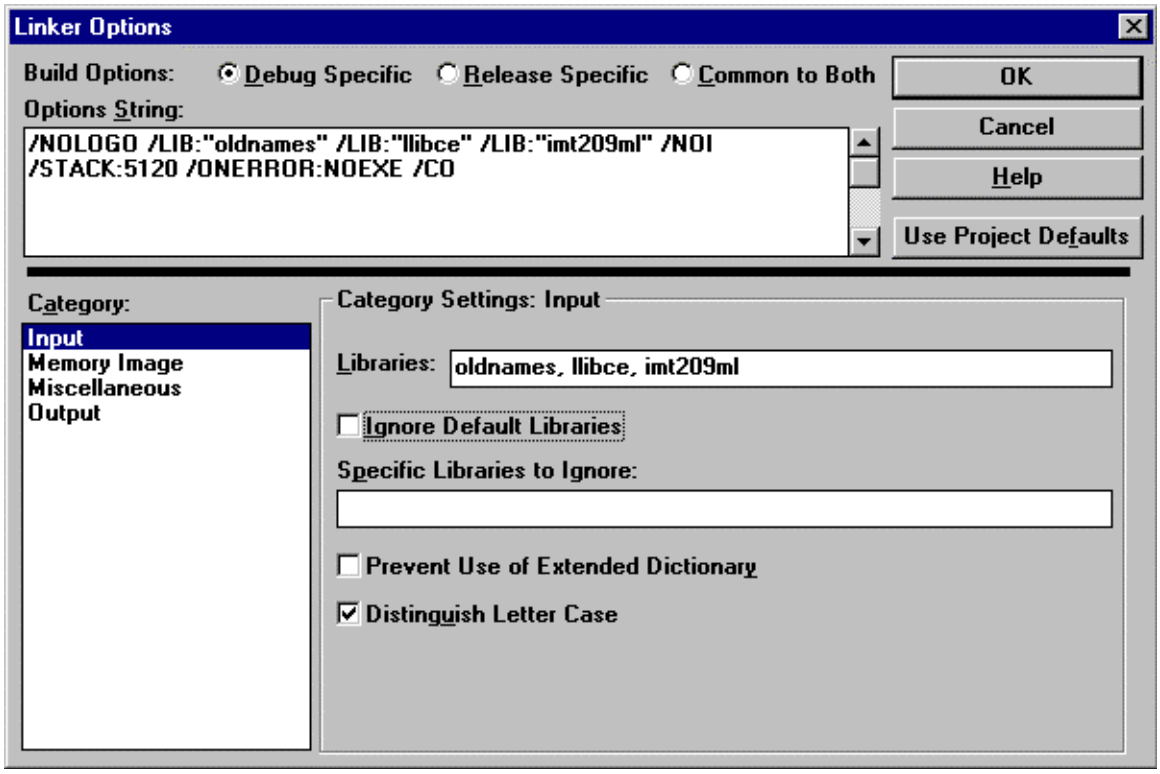
Compiler Options: Memory Model



Select the large memory model.

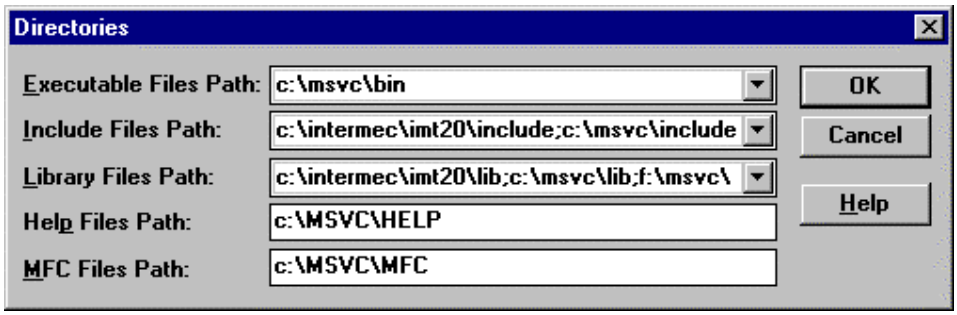
Note: *To change the string shown in the Options String box, you must make selections on the C/C++ Compiler Options dialog box and choose OK. You cannot edit the string directly. For example, when you select the large memory model, the /AM in the string changes to /AL.*

Linker Options



Add imt209ml to the list of libraries.

Directory Settings



List the whole path to c:\instance\imt20\include and to c:\instance\imt20\lib.

C

Full ASCII Charts

This appendix contains a full ASCII chart and charts of Code 39 bar code labels that you can scan with the TRAKKER T2090.

Full ASCII Table

This table lists the ASCII characters and their binary, hexadecimal, and Code 39 equivalents.

Full ASCII Table

Binary[0]	Hex[1]	Decimal	Code 39	ASCII[2]	Binary[0]	Hex[1]	Decimal	Code 39	ASCII[2]
00000000	00	00	%U	NUL	00100000	20	32	SP	SP[3]
00000001	01	01	$A	SOH	00100001	21	33	/A	!
00000010	02	02	$B	STX	00100010	22	34	/B	"
00000011	03	03	$C	ETX	00100011	23	35	/C	#
00000100	04	04	$D	EOT	00100100	24	36	/D	$
00000101	05	05	$E	ENQ	00100101	25	37	/E	%
00000110	06	06	$F	ACK	00100110	26	38	/F	&
00000111	07	07	$G	BEL	00100111	27	39	/G	'
00001000	08	08	$H	BS	00101000	28	40	/H	(
00001001	09	09	$I	HT	00101001	29	41	/I	)
00001010	0A	10	$J	LF	00101010	2A	42	/J	*
00001011	0B	11	$K	VT	00101011	2B	43	/K	+
00001100	0C	12	$L	FF	00101100	2C	44	/L	,
00001101	0D	13	$M	CR	00101101	2D	45	/M	-
00001110	0E	14	$N	SO	00101110	2E	46	/N	.
00001111	0F	15	$O	SI	00101111	2F	47	/O	/
00010000	10	16	$P	DLE	00110000	30	48	/P[4]	0
00010001	11	17	$Q	DC1	00110001	31	49	/Q	1
00010010	12	18	$R	DC2	00110010	32	50	/R	2
00010011	13	19	$S	DC3	00110011	33	51	/S	3
00010100	14	20	$T	DC4	00110100	34	52	/T	4
00010101	15	21	$U	NAK	00110101	35	53	/U	5
00010110	16	22	$V	SYN	00110110	36	54	/V	6
00010111	17	23	$W	ETB	00110111	37	55	/W	7
00011000	18	24	$X	CAN	00111000	38	56	/X	8
00011001	19	25	$Y	EM	00111001	39	57	/Y	9
00011010	1A	26	$Z	SUB	00111010	3A	58	/Z	:
00011011	1B	27	%A	ESC	00111011	3B	59	%F	;
00011100	1C	28	%B	FS	00111100	3C	60	%G	<
00011101	1D	29	%C	GS	00111101	3D	61	%H	=
00011110	1E	30	%D	RS	00111111	3E	62	%I	>
00011111	1F	31	%E	US	00111111	3F	63	%J	?

Full ASCII Table (continued)

Binary[0]	Hex[1]	Decimal	Code 39	ASCII[2]	Binary[0]	Hex[1]	Decimal	Code 39	ASCII[2]
01000000	40	64	%V	@	01100000	60	96	%W	`
01000001	41	65	A	A	01100001	61	97	+A	a
01000010	42	66	B	B	01100010	62	98	+B	b
01000011	43	67	C	C	01100011	63	99	+C	c
01000100	44	68	D	D	01100100	64	100	+D	d
01000101	45	69	E	E	01100101	65	101	+E	e
01000110	46	70	F	F	01100110	66	102	+F	f
01000111	47	71	G	G	01100111	67	103	+G	g
01001000	48	72	H	H	01101000	68	104	+H	h
01001001	49	73	I	I	01101001	69	105	+I	i
01001010	4A	74	J	J	01101010	6A	106	+J	j
01001011	4B	75	K	K	01101011	6B	107	+K	k
01001100	4C	76	L	L	01101100	6C	108	+L	l
01001101	4D	77	M	M	01101101	6D	109	+M	m
01001110	4E	78	N	N	01101110	6E	110	+N	n
01001111	4F	79	O	O	01101111	6F	111	+O	o
01010000	50	80	P	P	01110000	70	112	+P	p
01010001	51	81	Q	Q	01110001	71	113	+Q	q
01010010	52	82	R	R	01110010	72	114	+R	r
01010011	53	83	S	S	01110011	73	115	+S	s
01010100	54	84	T	T	01110100	74	116	+T	t
01010101	55	85	U	U	01110101	75	117	+U	u
01010110	56	86	V	V	01110110	76	118	+V	v
01010111	57	87	W	W	01110111	77	119	+W	w
01011000	58	88	X	X	01111000	78	120	+X	x
01011001	59	89	Y	Y	01111001	79	121	+Y	y
01011010	5A	90	Z	Z	01111010	7A	122	+Z	z
01011011	5B	91	%K	[	01111011	7B	123	%P	{
01011100	5C	92	%L	\	01111100	7C	124	%Q	|
01011101	5D	93	%M	]	01111101	7D	125	%R	}
01011110	5E	94	%N	^	01111110	7E	126	%S	~
01011111	5F	95	%O	_	01111111	7F	127	%T[5]	n[6]

Notes for the Full ASCII Table

0 Bit positions are 76543210.

1 This column lists the hexadecimal value.

2 This column lists the ASCII character.

3 SP is the SPACE character.

4 The Code 39 characters /P through /Y may be interchanged with the numbers 0 through 9.

5 %T may be interchanged with %X or %Y or %Z.

6 n is the Delete character.

Full ASCII Control Characters Table

Control Character	Definition	Control Character	Definition
NUL	Null or all zeroes	DC1	Device Control 1 (XON)
SOH	Start of Heading	DC2	Device Control 2
STX	Start of Text	DC3	Device Control 3 (XOFF)
ETX	End of Text	DC4	Device Control
EOT	End of Transmission	NAK	Negative Acknowledge
ENQ	Enquiry	SYN	Synchronous Idle
ACK	Acknowledgment	ETB	End Transmission Block
BEL	Bell	CAN	Cancel
BS	Backspace	EM	End of Medium
HT	Horizontal Tab	SUB	Substitute
LF	Line Feed	ESC	Escape
VT	Vertical Tab	FS	File Separator
FF	Form Feed	GS	Group Separator
CR	Carriage Return	RS	Record Separator
SO	Shift Out	US	Unit Separator
SI	Shift In	SP	Space
DLE	Data Link Escape	DEL	Delete

Full ASCII Bar Code Chart

The charts in this section list the Code 39 bar code label for each ASCII character. To use these bar code labels, you must configure the TRAKKER T2090 computer to use Code 39 in Full ASCII mode.

Control Characters

NUL

%U

SOH

%A

STX

$B

ETX

$C

EOT

$D

ENQ

$E

ACK

$F

BEL

$G

BS

$H

HT

$I

LF

$J

VT

$K

FF

$L

CR

$M

SO

$N

SI

$O

DLE

$P

DC1

$Q

DC2

$R

DC3

$S

DC4

$T

NAK

$U

SYN

$V

ETB

$W

Control Characters (continued)

CAN

$X

EM

$Y

SUB

$Z

ESC

%A

FS

%B

GS

%C

RS

%D

US

%E

DEL

%T

Symbols and Punctuation Marks

! (exclamation point)

/A

" (quotation marks)

/B

#

/C

$

/D

%

/E

&

/F

' (apostrophe)

/G

(

/H

)

/I

* (asterisk)

/J

+

/K

- (dash)

/M

/

/O

=

%H

. (period)

/N

, (comma)

/L

: (colon)

/Z

; (semicolon)

%F

Symbols and Punctuation Marks (continued)

?

%J

<

%G

>

%I

@

%V

[

%K

]

%M

~ (tilde)

%S

^

%N

_ (underline)

%O

\

%L

` (left single quote)

%W

¦ (pipe)

%Q

{

%P

}

%R

Space

* *

Numbers

0

0

1

1

2

2

3

3

4

4

5

5

6

6

7

7

8

8

9

9

Uppercase Letters

A

A

B

B

C

C

D

D

E

E

F

F

G

G

H

H

I

I

J

J

K

K

L

L

M

M

N

N

O

O

P

P

Q

Q

R

R

S

S

T

T

U

U

V

V

W

W

X

X

Y

Y

Z

Z

Lowercase Letters

a

+A

b

+B

c

+C

d

+D

e

+E

f

+F

g

+G

h

+H

i

+I

j

+J

k

+K

l

+L

m

+M

n

+N

o

+O

p

+P

q

+Q

r

+R

s

+S

t

+T

u

+U

v

+V

w

+W

x

+X

y

+Y

z

+Z