THE INTELLIGENT MESSAGE DISPLAY SERIES

MODELS ADI & MDI INSTRUCTION MANUAL

INTRODUCTION

The Alphanumeric Display Intelligent unit (ADI) and Message Display Intelligent unit (MDI) are programmable displays in our multi-purpose series of industrial control products that are field-programmable to solve multiple applications. This series of products is built around the concept that the end user has the capability to program different messages into the unit in order to adapt to various indication and process requirements.

The Intelligent Display which you have purchased has the same high quality workmanship and advanced technological capabilities that have made Red Lion Controls the leader in today's industrial market.

Red Lion Controls has a complete line of industrial indication and control equipment, and we look forward to being of service to you now and in the future.

UL Recognized Component,
File # E171375

CAUTION: Read complete instructions prior to installation and operation of the unit.

CAUTION: Risk of electric shock.

Table of Contents

Table of Contents (Cont'd)

Table of Contents (Cont'd)

Table of Contents (Cont'd)

Table of Contents (Cont'd)

Table of Contents (Cont'd)

This page intentionally left blank

GENERAL DESCRIPTION

This manual will serve as a guide to the operation and programming of the Message Display Intelligent (MDI) and the Alphanumeric Display Intelligent (ADI) units. Product specifications and Hardware information pertaining to the individual models can be found at the end of this manual.

The Intelligent Display units are versatile and powerful message centers. The broad flexibility and functionality of the units make them particularly adaptable to a wide variety of applications, including:

Display and Monitoring of Measured Values
Indication of Warning, Error, and Alarm Conditions
Monitoring of Manufacturing Processes
Display of Machine Start-Up and Operation Procedures

Each unit is capable of storing and displaying up to 256 separate messages. A message can contain up to 250 characters of text and can display any of the 153 customizable characters, including the standard 96 character ASCII set. Individual lines of the message text can be programmed to scroll in a block or character fashion. Individual characters, blocks and lines of text, and entire messages can be programmed to blink.

Message text can be assembled and transmitted to one or more Message Display Slave (MDS) units for remote display of messages. For interfacing with serial printers and ASCII terminals, the message text can be assembled and transmitted in any format the user desires. In this case, the appropriate lower ASCII control characters can be inserted in the message text where necessary.

Messages can display the Current Time and Date and any of the sixteen Elapsed Timer values. A message can also collect and display multiple variable Embedded Data items.

Messages can be requested via the Serial and Parallel ports, which the user can configure to meet the needs of most applications. The Parallel Port can also be configured to issue Automatic Message Requests based on changing port values.

Each unit contains functions for requesting messages on a Periodic basis, and for processing requests based on Elapsed Time and Embedded Data values, as well as for executing Chained and Linked message lists. Any programmed message can be designated for automatic request on Unit Power-Up and Reset. A separate message can be designated for display whenever the unit's Display would be blank.

RLC offers IBM® compatible software for configuring and programming the unit. The Message Display User software (SFMD), with its easy-to-use menus and extensive prompts and on-line Help functions, greatly assists the user in fitting the unit to the application at hand.

The user can easily create and save multiple Configuration, Character, and Message files with the SFMD software. Extensive file handling features are included, such as uploading, downloading, and printing of files. Message simulation and terminal emulation functions also come with the software.

A powerful feature of the Intelligent Display is the Message Queue. With the Queue function disabled, the unit processes messages on a first-come, first-served basis. Once the message is processed, higher priority or equal priorities, go to the display, while lower priority messages are discarded. However, with the Queue function enabled high priority messages are placed on the Display while lower priority messages are placed on the Queue for later display in priority order.

SAFETY SUMMARY

All safety related regulations, local codes and instructions that appear in the manual or on equipment must be observed to ensure personal safety and to prevent damage to either the instrument or equipment connected to it. If equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

Do not use this unit to directly command motors, valves, or other actuators not equipped with safeguards. To do so, can be potentially harmful to persons or equipment in the event of a fault to the unit.

INTERFACING WITH AN INTELLIGENT DISPLAY

To program messages in an Intelligent Display, the unit must be connected to an IBM® compatible computer running the Message Display User Software (SFMD), see Figures 1 and 2, Computer interfacing. User friendly menus with numerous prompts and on-line Help menus assist the user in interfacing with and programming an Intelligent Display. The SFMD software allows the user to create Configuration, Character, and Message files and save them to disk. These files can then be downloaded as needed, from the computer to one unit or more than one unit if a communications loop is used.

The SFMD software also includes utilities for message simulation and terminal emulation. With the Message Simulator, the user can verify the proper operation of a Message or Message file on the computer screen without having to download the file to an Intelligent Display.

The Terminal Emulator can assist the user in verifying and testing the operation of the downloaded Message file. It can also be used to monitor and troubleshoot serial port communications.

SYSTEM REQUIREMENTS

IBM® compatible (286 or greater) with:

RS232 serial port
640K RAM free
DOS 3.0 or later
1.4 Meg floppy drive
Monitor with MDA, CGA, EGA, or VGA graphics card
Note: The SFMD software will not run on a PC-XT computer

Cabling:

Single Unit

ADI: Standard RS232 Cable (9 pin male)
MDI: RLC Model MCCA cable for initial set-up

Multiple Units (ADI or MDI)

RLC GCM232 Serial Converter Module
Serial Communications Cable (straight through)
RLC Auxiliary Power Supply (Model APS) or equivalent 12 VDC power supply.

Message Display User Software (SFMD):

Available on 3.5" floppy disks.
Used for programming both ADI and MDI units
Note: Setup may only be performed via the serial port

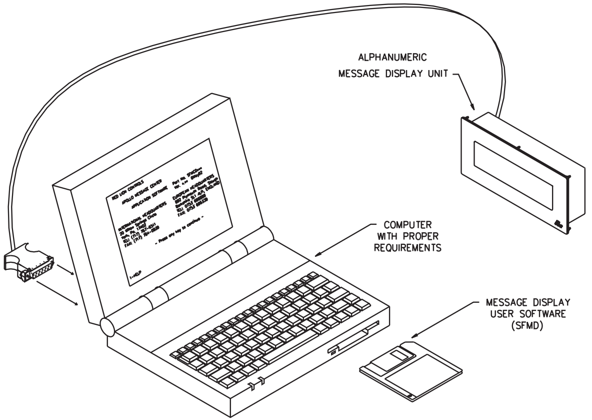


Figure 1, Computer Interfacing with ADI

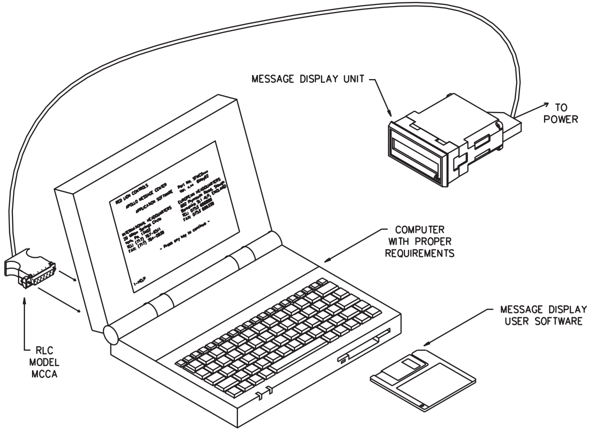


Figure 2, Computer Interfacing with MDI

CONFIGURATION FILE SET-UP

The Unit Function, Parallel Port, Serial Port, Output Pin, and Display Configuration settings are all contained in an Intelligent Display's Configuration File. These Configuration File settings determine the manner in which an ADI/MDI interfaces with other units, as well as how it processes and displays Messages.

There are two different configuration methods available. Most users will want to use the SFMD software's Configuration Editing Screen, which is shown in Figure 3, and described in detail in the remainder of this section.

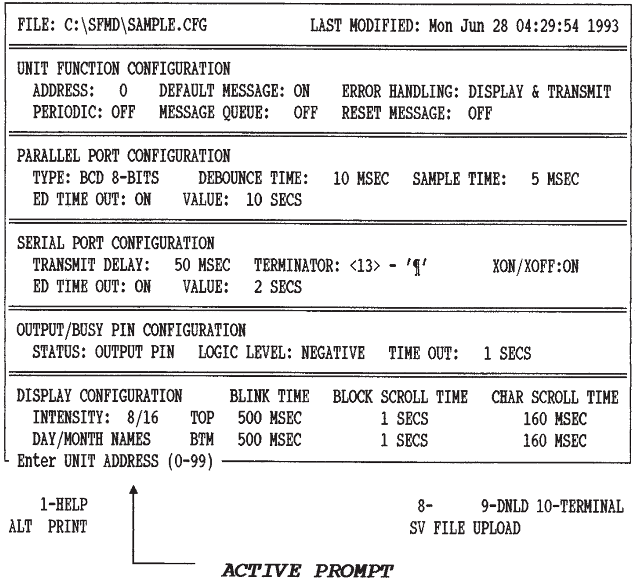
Some applications, however, may require on-line configuration from a controlling device such as a computer or PLC. In this case, you can separately access each of the ADI/MDI's Configuration settings through Commands C01 to C08, either by transmitting a command string to the unit or by programming the desired command string as a Command Message and issuing a request for that message. See *Commands*, page 75, for more details.

The Configuration Screen

You can program the Configuration settings on an IBM® compatible computer (PC) with the Message Display User Software (SFMD). The settings are programmed in a Configuration File (".CFG" file extension), which can be saved on the PC's hard or floppy disk drives. The Configuration File is then downloaded into the non-volatile memory of the unit via the PC's serial port.

Configuration File settings are programmed in the Configuration Editing Screen. The upper left corner of the Configuration Screen displays the name of the current Configuration File, see Figure 3, Configuration Editing Screen. To the right of the file name are displayed the last date and time that the file was modified. Located at the bottom of the screen's border is an Active Prompt Line, which displays basic information about the currently highlighted field. If you require more information about a particular field, on-line Help for the highlighted field can be accessed by pressing F1. General Help is also available from the Main Menu.

Although you program the Configuration File settings on the PC's screen, the new Configuration File is not effective until it is downloaded to the unit. Therefore, the ADI/MDI continues to access the current settings contained in its Configuration File until a new file is downloaded.



```
FILE: C:\SFMD\SAMPLE.CFG              LAST MODIFIED: Mon Jun 28 04:29:54 1993

UNIT FUNCTION CONFIGURATION
  ADDRESS:   0    DEFAULT MESSAGE: ON     ERROR HANDLING: DISPLAY & TRANSMIT
  PERIODIC: OFF   MESSAGE QUEUE:   OFF    RESET MESSAGE:  OFF

PARALLEL PORT CONFIGURATION
  TYPE: BCD 8-BITS     DEBOUNCE TIME:   10 MSEC   SAMPLE TIME:   5 MSEC
  ED TIME OUT: ON    VALUE:  10 SECS

SERIAL PORT CONFIGURATION
  TRANSMIT DELAY:   50 MSEC   TERMINATOR: <13> - '¶'       XON/XOFF:ON
  ED TIME OUT: ON    VALUE:   2 SECS

OUTPUT/BUSY PIN CONFIGURATION
  STATUS: OUTPUT PIN   LOGIC LEVEL: NEGATIVE   TIME OUT:   1 SECS

DISPLAY CONFIGURATION       BLINK TIME   BLOCK SCROLL TIME   CHAR SCROLL TIME
  INTENSITY:  8/16    TOP   500 MSEC          1 SECS            160 MSEC
  DAY/MONTH NAMES     BTM   500 MSEC          1 SECS            160 MSEC
Enter UNIT ADDRESS (0-99)

   1-HELP                                           8-      9-DNLD 10-TERMINAL
ALT  PRINT                                          SV FILE UPLOAD
```

Figure 3, Configuration Editing Screen

The following Configuration File settings are described in groups, in the order in which they appear on the screen.

Unit Function Configuration

The Unit Function settings are used to enable and disable various major functions and features. The definition and operation of each Unit Function Configuration setting is described below.

You can also access all of the Unit Function Configuration settings described below with Command C01. See *Command C01 - Configure Unit Functions*, page 78, for more details on the proper usage of the C01 Command.

Unit Address

The Unit Address can be set from 0 to 99. As with other RLC products, a unit with Address 0 does not require the "N00" address prefix when receiving serial communications.

Since multiple units on a loop with the same address respond simultaneously to activity on the serial loop, RLC recommends that you assign each unit on a loop a unique Unit Address. If two or more units attempt to transmit at the same time, their transmissions will "collide" on the loop, thus becoming corrupted. This action causes no physical harm to the units or the loop, however, the corrupted transmissions may trigger unexpected results from any unit receiving those transmissions.

Default Message

You can turn the Default Message function ON or OFF as required by your application. With the Default function OFF and no pending Message requests (Message Chain, Periodic, Trigger, etc.), the unit blanks the display whenever it removes a Message from its display.

With the Default function ON and no pending Message requests, the unit automatically requests the Default Message instead of blanking the display. Consequently, when there is no other Message to display, the unit displays the Default Message. You can designate any one of the 256 Messages as the Default Message. The actual Default Message number is programmed in the File/Unit Info menu of the Message Editing screen.

If the Default Message is cancelled, and there are no other pending Messages, the ADI/MDI again requests the Default Message. Therefore you should exercise care when selecting and programming the Default Message, as unexpected results may occur if you choose a short Time Out value, or set the Message Destination to something other than Display.

(See *MESSAGE PROCESSING*, page 41, for details on the operation of the Default Message Function.)

Error Handling

You can configure the generation of Error Codes upon encountering certain Error conditions. You can choose how the ADI/MDI responds to an Error condition from the following Error Handling methods:

Display - The Error Message is displayed, containing the Error Code. An Error Message blinks, and has a time out value of 16 seconds, a priority of 0, and does not get placed on the Queue. No other Message can interrupt an Error Message except a Temporary Message or another Error Message.

Transmit - The Error Code is transmitted over the serial port as an ASCII character string.

Display and Transmit - All Error Codes are displayed and transmitted.

Ignore - The unit continues to detect Error conditions, but gives no indication of the Error.

Note: With the Serial Port XON/XOFF function enabled, the unit will always transmit the Error Code regardless of the Error Handling setting you have selected.

(See *Error Codes and Numbers*, page 104, for details on the processing of Error Conditions.)

Periodic Message

You can turn the Periodic Message function ON or OFF. When the Periodic function is ON, each Periodic entry is processed according to that entry's individual configuration. With the Periodic function turned OFF, the Periodic Message function is disabled and all Periodic entries are ignored, regardless of their respective configurations.

(See *PERIODIC MESSAGE FUNCTION*, page 67, for details on the operation of the Periodic Message Function.)

Message Queue

You can turn the Message Queue function ON or OFF. When the Queue function is ON, the requested Messages designated for the Queue are placed on the Queue. With the Queue function OFF, the Queue is frozen and NO newly requested Messages are placed on the Queue. While the Queue is OFF, messages from the Queue are not automatically displayed when the display becomes available.

(See *Message Queue*, page 43, for details on the operation of the Message Queue.)

Reset Message

You can turn the Reset Message function ON or OFF. When the Reset function is ON, the Reset Message is automatically requested on every unit power-up or reset. You can designate any one of the 256 Messages as the Reset Message. The actual Reset Message number is programmed in the File/Unit Info menu of the Message Editing screen.

With the Reset function OFF, the unit blanks the display on unit power-up and reset, and then waits to process Message requests.

(See *MESSAGE PROCESSING*, page 41, for details on the operation of the Reset Message Function.)

Parallel Port Configuration

The Parallel Port is a unidirectional input port. It consists of eight DATA lines (D0 to D7) and two Control lines (STROBE and MESSAGE/DATA). The DATA and Control lines are separately DIP switch selectable for sinking or sourcing current, high or low level input signals, and positive or negative logic. The ADI has additional switch settings for 5 V or 12 V DC bias voltage, while the MDI is preset for 12 VDC operation. (See *PARALLEL PORT*, page 55, for details on the selection of these DIP switch settings.)

You can also access all of the Parallel Port Configuration settings described below with Command C04. See *Command C04 - Configuration Unit Functions*, page 83, for details on the proper usage of the C04 Command.

Type

There are seven Parallel Port message request formats available:

4 Bit BCD
8 Bit BCD
9 Bit BCD
4 Bit Binary
8 Bit Binary
AMR Mode 1
AMR Mode 2

The 4-, 8-, and 9-bit designations refer to the width of the port which set the number of data lines scanned, as well as the number of Strobe pulses necessary for entering a byte of information. The BCD and Binary designations apply to Parallel Port Message requests only. The AMR Modes specify the two methods for generating Automatic Message Requests based on changes of the parallel port DATA lines. Embedded Data items can be entered in ASCII, BCD, or Binary formats, as specified by the Message collecting the data, independent of this Message request setting.

(See *PARALLEL PORT*, page 55, for details on the operation of the various Parallel Port Types.)

Debounce Time

The Debounce Time specifies the minimum wait time after the end of one valid STROBE pulse before detecting the beginning of the next STROBE pulse. This value ranges from 10 milliseconds to 2.5 seconds.

(See *PARALLEL PORT*, page 55, for details on the usage and operation of the Debounce Timer.)

Sample Time

The Sample Time setting specifies the minimum time the DATA lines and the MESSAGE/DATA line must be stable following the start of the STROBE pulse before the value is accepted as valid. You can set this value from 1 to 255 milliseconds.

(See *PARALLEL PORT*, page 55, for details on the usage and operation of the Sample Timer.)

ED Time Out

The Embedded Data (ED) Time Out value ranges from 0 to 254 seconds and specifies how long the ADI/MDI waits to receive a data item over the Parallel Port for a message. You can also turn the ED Time Out function OFF, causing the ADI/MDI to wait indefinitely for a data item. The ADI/MDI maintains separate values for the Parallel Port and the Serial Port ED Time Out settings. In addition, an individual data item within a message can specify its own Time Out value, which overrides this Parallel Port ED Time Out value for that item only.

(*See Collecting Embedded Data*, page 21, for details on the operation of the Embedded Time Out Function.)

Serial Port Configuration

The Intelligent Displays are equipped with an optically isolated two-way 20 mA current loop for serial communications. The ADI is also equipped with a full duplex RS-232 port. (Refer to the *ADI Installation & Specifications Guide* Insert for more details.) The baud rate, parity, and data bit settings are DIP switch selectable. These settings should conform to those of the system in which the ADI/MDI is to be installed. (See *SERIAL COMMUNICATIONS*, page 68 for details on the operation and usage of the Serial Port.)

You can also access all of the Serial Port Configuration settings described below with Command C05. See Command C05 - *Configure Serial Port*, page 84, for details on the proper usage of the C05 Command.

Transmit Delay

The Transmit Delay ranges from 0 to 2.5 seconds, in 0.01 second intervals, and establishes the minimum time between the end of one transmission string and the beginning of the next transmission string.

(See *SERIAL COMMUNICATIONS, Serial Transmit Delay*, page 70, for details on the usage and operation of the Serial Port Transmit Delay Function.)

Serial Terminator

The Serial Terminator character signals the unit that it should process the contents of its Serial Receive buffer. The Terminator character can be programmed from 01h to 07h (1 to 7), 09h to 1Ah (9 to 26), 1Ch to 2Fh (28 to 47), and 3Ah to 40h (58 to 64). An individual Embedded Data item within a message can specify its own Terminator character, which overrides this Serial Terminator character for that item only.

(See *SERIAL COMMUNICATIONS*, *Serial Terminator*, page 70, for details on the usage and operation of the Serial Port Terminator.)

ED Time Out

The Embedded Data (ED) Time Out value ranges from 0 to 254 seconds and specifies how long the unit should wait to receive a data item over the Serial Port for a Message. You can also turn the ED Time Out function OFF, causing the unit to wait indefinitely for a data item. The unit maintains separate values for the Serial Port and the Parallel Port Time Out settings. In addition, an individual data item within a message can specify its own Time Out value, which overrides this Serial Port ED Time Out value for that item only.

(See *Collecting Embedded Data*, page 21, for details on the operation of the ED Time Out Function.)

XON/XOFF

The XON/XOFF setting and the Output Pin function setting select the communications handshaking protocol used when receiving and processing Message Requests and Embedded Data Items. The ASCII XOFF character, 13h (19), is transmitted to indicate the Unit Busy condition, and the ASCII XON character, 11h (17), is transmitted to indicate the Unit Ready condition. The XON/XOFF Mode and the BUSY setting of the Output Pin Function can NOT both be selected at the same time.

(See *HANDSHAKING*, page 50, for details on the operation of the XON/XOFF Function.)

Output/Busy Pin Configuration

Intelligent Displays contain one open-collector Output pin. You can select the function of this pin, its Active and Inactive logic levels, and you can specify a default Time Out value. (See the OUTPUT PIN section for details on the usage and operation of the Output/Busy Pin.)

You can also access all of the Output/Busy Pin Configuration settings described below with Command C06. See *Command C06 - Configure Output/Busy Pin*, page 85, for the proper usage of the C06 Command.

Status

You can configure the Output pin for one of the following states:

OUTPUT Mode
BUSY Mode
DISABLED

In the OUTPUT Mode, the pin functions as an independent Output pin that you can configure and control with Messages and Commands.

In the BUSY Mode, the unit uses the pin for the READY/BUSY handshaking protocol. If you choose the BUSY Mode, the XON/XOFF Mode of the Serial Port is automatically turned OFF since both handshaking functions cannot be in use at the same time. With the pin in the BUSY Mode, Messages cannot access the Output pin.

You can also DISABLE the Output pin, in which case the pin has no function.

(See the *OUTPUT PIN* and *HANDSHAKING* sections for details on the Output and Busy Modes of the Output/Busy Pin.)

Logic Level

You can configure the Output pin for positive or negative logic. In positive logic, with the open-collector pulled-up through a resistor, the pin outputs a logic “1” (does not conduct) when set to the ON state in the OUTPUT Mode, or while the unit is in the READY condition in the BUSY Mode. In negative logic, the converse is true. With the open-collector pulled-up through a resistor, the pin outputs a logic “0” (conducts) for the ON state and the BUSY condition.

The ON state and the BUSY condition are considered the “Active” levels of the pin. The OFF state and the READY condition are considered the “Inactive” levels of the pin.

(See *OUTPUT PIN*, page 54, for details on the various combinations of output levels for the Output Pin.)

Time Out

The ADI/MDI contains a dedicated Output Time Out function which you can configure to automatically turn the Output pin OFF after a predetermined Time Out interval.

You can set this default Output Time Out value from 10 to 1260 milliseconds, 1 to 63 seconds, or 1 to 63 minutes. Messages and Commands can access this value, or can supply their own Time Out value, independent of this default value.

(See *OUTPUT PIN*, page 54, for details on the usage and operation of the Output Time Out Function.)

Display Configuration

The Intelligent Displays utilize a Liquid Crystal Display (LCD) with direct LED backlighting. On all models except the MDI2T, a pot is provided to adjust the display for maximum contrast. The display contrast on the MDI2T however, is fixed at the factory. Other pot adjustments will be discussed below in the applicable section.

You can access all of the Display Configuration settings described below with Command C08. See *Command C08 - Configure Unit Display*, page 88, for details on the proper usage of the C08 Command.

Intensity

The Backlight Intensity of the display is software adjustable through 17 levels and ranges from 0 (OFF) to 16 (FULL INTENSITY) in 1/16 level increments. The Intensity of the backlight is current controlled, and each increment of the Intensity level accounts for about 1/16 of the total backlight current.

If you have a Tri-Color unit, the Backlight Intensity level sets the color of the display. The 0 level is FULL RED, the 16 level is FULL YELLOW/GREEN, the intermediate levels (1-15) represent various shades of RED, ORANGE, and YELLOW. The actual brightness of the backlight on a Tri-color unit is adjustable using rear panel or side mounted pots.

Blink Time

Any character or group of characters in a Message can be programmed to blink by surrounding the desired characters in the Message text with the BLINK control code (b -<ALT-B>). All blinking characters on a given display line blink in unison at the designated Blink time for that line. The Blink time is adjustable from 10 to 1270 milliseconds and can be set separately for the top and bottom lines of the display.

(See *Message String Construction, Blinking Characters*, page 37, for details on the usage and operation of the Blink feature.)

Block Scroll Time

Selecting Block Scrolling for a line of text in a message causes the unit to display each block of text (generally one or more consecutive words in a Message) one after the other on that line. The Block Scroll time sets the amount of time the ADI/MDI displays each Block of text for a given line. Once the last block of text for a line has been displayed, the unit repeats the line from the first block.

Blocks of text can be designated by inserting the SEGMENT control code (s - <ALT-S>) at the desired locations in the text. The Block Scroll time can range from 10 to 1270 milliseconds or 1 to 127 seconds per Block and can be set separately for each line of the display.

(See *Message String Construction, Block Scrolling*, page 37, for details on the usage and operation of the Block Scroll feature.)

Character Scroll Time

Selecting Character Scrolling for a line of text in a message causes the text for that line to advance across the display from right to left, one character position at a time. The Character Scroll time sets the amount of time a single shift in character position takes. Once the last character of a line has scrolled off the display to the left, the unit repeats scrolling the line from the right.

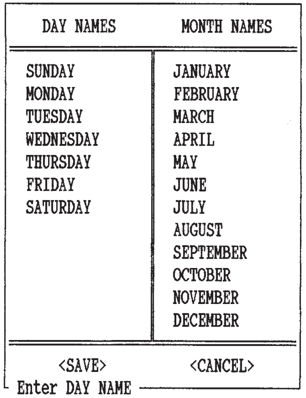
You can adjust the Character Scroll time from 10 to 1270 milliseconds per character position and you can select separate scroll times for each line of the display. You can also designate separate blocks of text for Character Scrolling by inserting the SEGMENT control code (s - <ALT-S>) at the desired locations in the text. In this case, the unit Character scrolls each block of text across the display separately, one block at a time. (See *Message String Construction, Character Scrolling*, page 37, for details on the usage and operation of the Character Scroll feature.)

Day/Month Names

The ADI/MDI contains a file with the names of the days and months. Through the placement of the CURRENT TIME control code (c - <ALT-C>) and the proper format codes, you can have these names inserted in the Message text. You can access the Day/Month name menu by positioning the cursor on the DAY/MONTH NAMES field and hitting the SPACE BAR or clicking the mouse.

The Day/Month Name menu, with the default day and month names, appears on the PC screen as shown in Figure 4. You can program each name with up to thirteen characters.

(See *Message String Construction, Current Time And Date*, page 48, for details on the usage and specification of Day/Month names.)



DAY NAMES	MONTH NAMES
SUNDAY	JANUARY
MONDAY	FEBRUARY
TUESDAY	MARCH
WEDNESDAY	APRIL
THURSDAY	MAY
FRIDAY	JUNE
SATURDAY	JULY
	AUGUST
	SEPTEMBER
	OCTOBER
	NOVEMBER
	DECEMBER

Enter DAY NAME

Figure 4, Day/Month Names

MESSAGE FILE FEATURES

A Message File (".MSG" file extension) consists of all the Message Records for a specific application. A Message Record includes all the information the unit needs to process a request for the specified Message. You can program the Messages for a Message File in the Message Editing Screen of the SFMD software. Once a file is programmed, you can save the file to the PC's hard or floppy drives, and download the file to the unit. Although you can create and save multiple Message files, an ADI/MDI can only hold one Message File at a time.

Message Editing Screen

Individual Message Records are programmed in the Message Editing Screen. The upper left corner of the screen indicates the current Message Record number that you are working on. Each Message Record has a specific Message number assigned to it, from 0 to 255.

To the right of the Message number appears the size of the current Message Record, in bytes. The SFMD software updates the byte count whenever you SAVE the Message (press the F8-SVMSG key). A Message Record contains a minimum of 6 bytes and can contain a maximum of 255 bytes of information.

The current Message File path and name appear to the right of the Message Record size. Additional information about the current Message File and the unit you are programming can be found in the File/Unit Info sub-menu.

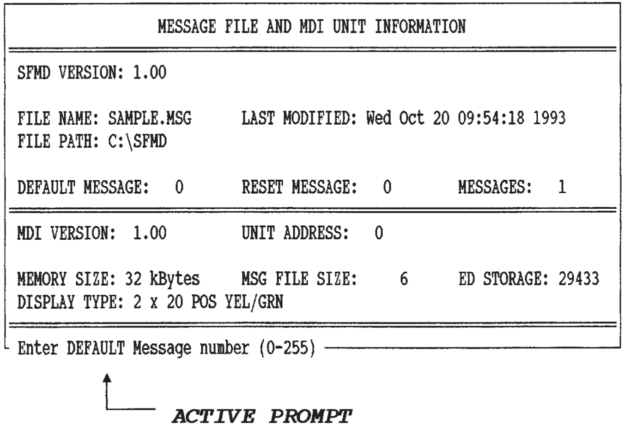
An active prompt line appears at the bottom of the screen's border, which displays basic information about the currently highlighted field. If you require more information about a particular field, on-line Help for the highlighted field can be accessed by pressing the F1-HELP key. General Help is also available from the Main Menu.

Although you program the Message File on the PC's screen, the ADI/MDI does not receive the new Message Records until you download the new Message File containing those Messages to the unit. Therefore, the unit continues to process the current Message Records in its Message File until you download a new file to it.

File/Unit Information

The MESSAGE FILE AND ADI/MDI UNIT INFORMATION sub-menu, as shown in Figure 5, displays general information about the current Message File you are editing and the unit you last downloaded to or uploaded from. This menu provides the current SFMD Version number, the full Message File Name and Message File Path, the date and time of the last file modifications, and the number of programmed Messages in the file.

You can select any of the 256 Message numbers for the Message File's Default and Reset Messages in this sub-menu. The Default and Reset Message functions themselves can be configured in the Configuration Editing Screen, or you can set them by issuing the C01 Command.



```
                  MESSAGE FILE AND MDI UNIT INFORMATION

SFMD VERSION: 1.00

FILE NAME: SAMPLE.MSG      LAST MODIFIED: Wed Oct 20 09:54:18 1993
FILE PATH: C:\SFMD

DEFAULT MESSAGE:   0       RESET MESSAGE:   0        MESSAGES:   1

MDI VERSION:  1.00         UNIT ADDRESS:   0

MEMORY SIZE: 32 kBytes     MSG FILE SIZE:     6      ED STORAGE: 29433
DISPLAY TYPE: 2 x 20 POS YEL/GRN

Enter DEFAULT Message number (0-255)
```

ACTIVE PROMPT

Figure 5, Message File and Unit Information

The MESSAGE FILE SIZE field indicates the size, in bytes, of the current Message File. The ED STORAGE field indicates the available memory, in bytes, for collection and storage of Embedded Data. These two fields are updated after every Message SAVE operation (pressing the F8-SVMSG key).

The ADI/MDI shares its available memory space (29940 bytes) between the Message File and any Embedded Data it collects. Therefore, the size of the Message File downloaded to the unit determines the available memory space for Embedded Data storage. As an example, the unit contains enough memory to hold 256 Messages of 100 bytes each, and still store up to 4 Kbytes of Embedded Data.

The UNIT ADDRESS field should be set for the unit with which the SFMD software must communicate. The corresponding unit information is then updated after each file download to or upload from that unit.

The DISPLAY TYPE setting determines the number of lines the SFMD software allows you to access in the Message Editing Screen, and whether to display the COLOR field in the Message Screen. The SFMD software only displays the COLOR field when the DISPLAY TYPE is set for a Tri-color version.

The Message Simulator uses the DISPLAY TYPE setting to accurately simulate the unit's display. An Intelligent Display unit cannot determine the type or color of its own display. You must set these values accordingly. (Note: The SFMD software is used for both the ADI and the Model MDI message centers. Therefore, certain settings of the DISPLAY TYPE are not applicable to the ADI and should not be selected. See *Ordering Information*, page 139, for available ADI versions.)

Message Destination

There are several types of Message Records; each type is defined by the intended Destination of the Message. How the unit processes a particular Message is determined by the Message's Destination.

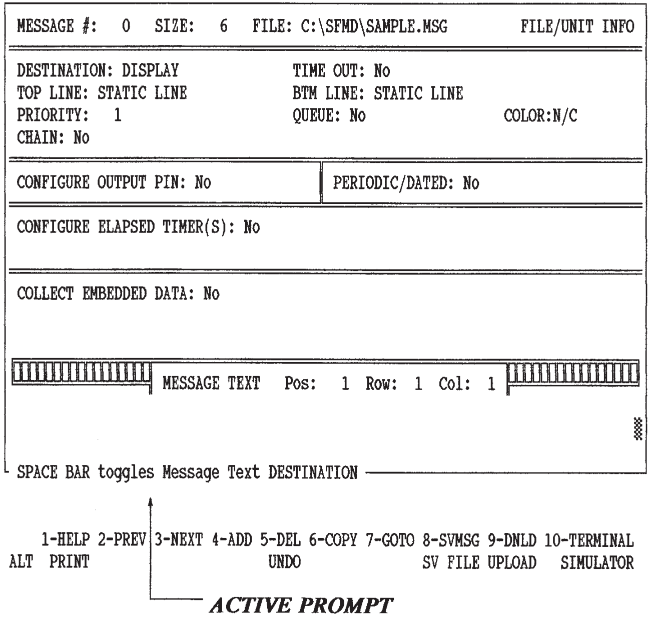
There are six types of Message Destinations. Each Message Destination is described in the following sections:

DISPLAY
MDS UNIT
TRANSMIT
DISPLAY & MDS UNIT
DISPLAY & TRANSMIT
COMMAND MESSAGE

Display Destination

A Message Record designated for DISPLAY specifies that the unit should place the requested Message's text on the display. The unit processes the Message request and then creates a Queue Entry for the Message. It then determines whether to place the message on the display, based on the requested Message's Priority and that of the currently displayed Message. If the Priority of the requested Message is the same or greater than that of the displayed Message, the ADI/MDI places the requested Message on the display. Otherwise, it discards the requested Message, unless the Message has been designated to be placed on the Queue.

In order to specify a Message Chain in a Message record, you must first select one of the DISPLAY destinations for the Message (DISPLAY, DISPLAY & MDS UNIT, DISPLAY & TRANSMIT).



```
MESSAGE #:   0   SIZE:   6   FILE: C:\SFMD\SAMPLE.MSG          FILE/UNIT INFO

DESTINATION: DISPLAY                  TIME OUT: No
TOP LINE: STATIC LINE                 BTM LINE: STATIC LINE
PRIORITY:   1                         QUEUE: No                 COLOR:N/C
CHAIN: No

CONFIGURE OUTPUT PIN: No            PERIODIC/DATED: No

CONFIGURE ELAPSED TIMER(S): No

COLLECT EMBEDDED DATA: No

                 MESSAGE TEXT   Pos:   1  Row:  1  Col:  1

SPACE BAR toggles Message Text DESTINATION

   1-HELP 2-PREV 3-NEXT 4-ADD 5-DEL 6-COPY 7-GOTO 8-SVMSG 9-DNLD 10-TERMINAL
ALT  PRINT                    UNDO               SV FILE UPLOAD   SIMULATOR
```

Figure 6, Display Destination Screen

MDS Unit Destination

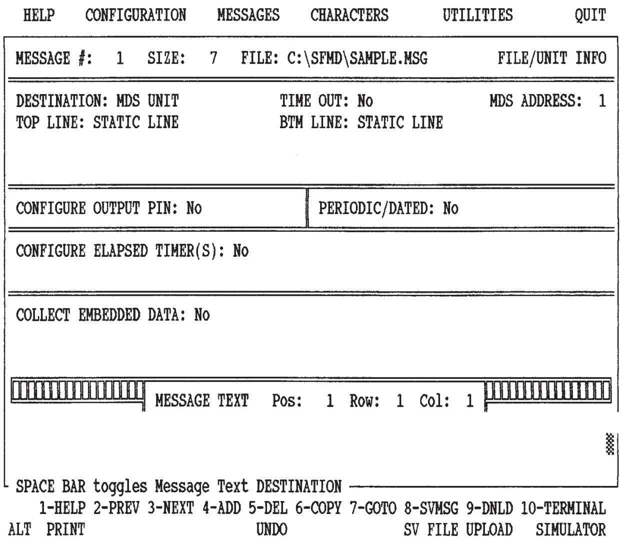
The MDS UNIT Destination causes the ADI/MDI to transmit the requested Message's text over the serial port in a Message Display Slave (MDS) compatible format. This type of transmission is intended for MDS units configured for the RLC Mode of operation and enables an Intelligent Display to remotely display Messages on one or more MDS units.

The unit prefixes an address string, "Nxx", for the Unit Address included in the Message record. This address specifies the MDS unit that is to process and display the transmission.

The assembled transmission can include a Message Time Out value, the unit's current Output Pin configuration and Real Time Clock (RTC) value, multiple Elapsed Timer values, and multiple Embedded Data values. Consequently, whenever the MDS receives this type of transmission, it sets its internal clock to the included RTC value of the transmitting unit, and it configures any specified Elapsed Timers to the respective configurations and values received.

The total transmission string length for the MDS UNIT destination is limited to 127 bytes, the size of the MDS Serial Receive buffer. You can view the resulting transmission string, and validate its length, by simulating the Message (pressing ALT-F10 SIMULATOR).

(See *SERIAL COMMUNICATIONS, MDS Unit Message Destination*, page 71, for details on the usage and operation of the MDS UNIT destination.)



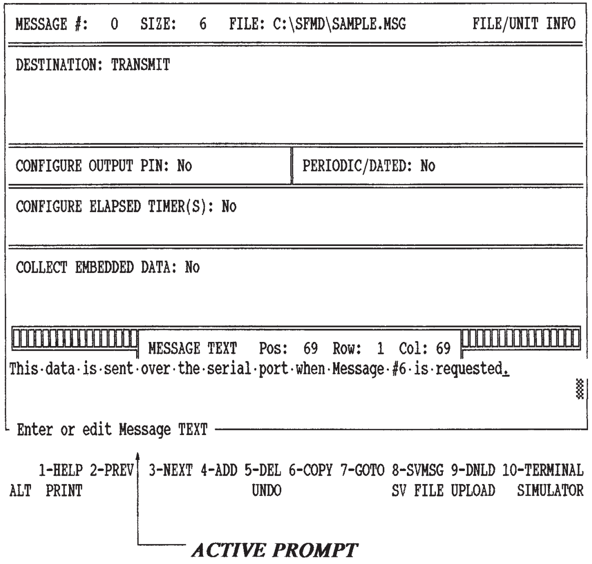
```
 HELP   CONFIGURATION   MESSAGES   CHARACTERS      UTILITIES       QUIT

 MESSAGE #:  1   SIZE:  7   FILE: C:\SFMD\SAMPLE.MSG        FILE/UNIT INFO

 DESTINATION: MDS UNIT              TIME OUT: No            MDS ADDRESS:  1
 TOP LINE: STATIC LINE              BTM LINE: STATIC LINE

 CONFIGURE OUTPUT PIN: No        PERIODIC/DATED: No

 CONFIGURE ELAPSED TIMER(S): No

 COLLECT EMBEDDED DATA: No

                  MESSAGE TEXT  Pos:  1  Row:  1  Col:  1

 SPACE BAR toggles Message Text DESTINATION
    1-HELP 2-PREV 3-NEXT 4-ADD 5-DEL 6-COPY 7-GOTO 8-SVMSG 9-DNLD 10-TERMINAL
ALT  PRINT                      UNDO               SV FILE UPLOAD   SIMULATOR
```

Figure 7, MDS Unit Destination

Transmit Destination

The TRANSMIT Destination causes the unit to transmit the requested Message's text over the serial port. This transmission would generally be intended for computers, PLC's, printers, and other serial devices that require a specific transmission format. The Message text can include the Current Date and Time, multiple Elapsed Timer values, and multiple Embedded Data Items. While assembling the transmission string, the ADI/MDI inserts the current values for any of the Time and Embedded Data Items into the transmission string wherever you specify.

You can include most of the lower ASCII control codes, such as Carriage Return <CR>, Line Feed <LF>, and Form Feed <FF>, in the Message Text.



```
MESSAGE #:   0   SIZE:   6   FILE: C:\SFMD\SAMPLE.MSG        FILE/UNIT INFO

DESTINATION: TRANSMIT

CONFIGURE OUTPUT PIN: No                  PERIODIC/DATED: No

CONFIGURE ELAPSED TIMER(S): No

COLLECT EMBEDDED DATA: No

                  MESSAGE TEXT   Pos:  69  Row:  1  Col: 69
This·data·is·sent·over·the·serial·port·when·Message·#6·is·requested.

Enter or edit Message TEXT

    1-HELP 2-PREV  3-NEXT 4-ADD 5-DEL 6-COPY 7-GOTO 8-SVMSG 9-DNLD 10-TERMINAL
ALT  PRINT                     UNDO                SV FILE UPLOAD   SIMULATOR
```

Figure 8, Transmit Destination

You must explicitly enter any serial string terminator your application requires as the last character in the string, the ADI/MDI does not automatically provide its Serial Terminator for the TRANSMIT destination. The maximum length of the assembled transmission string must be 255 bytes or less, including the terminator character. You can review the assembled transmission string, and validate its length, by simulating the Message (pressing ALT-F10 SIMULATOR).

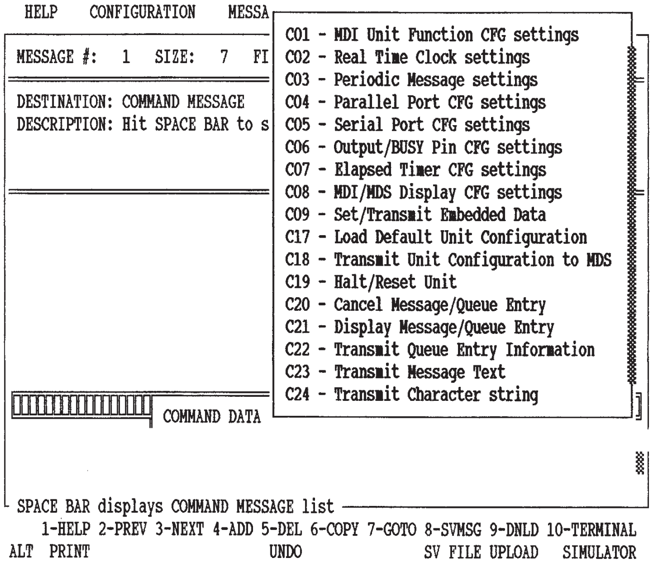
(See *SERIAL COMMUNICATIONS, Transmit Message Destination*, page 71, for details on the usage and operation of the TRANSMIT Destination.)

Combined Destinations

You can combine the DISPLAY destination with the TRANSMIT or MDS UNIT destinations in a single Message Record. When processing a combined Message record, the unit assembles the specified transmission string and initiates the serial transmission. The unit then creates the Queue Entry and decides whether to display the Message or leave it on the Queue. The DISPLAY & TRANSMIT combination facilitates both displaying a Message and transmitting the Message text to a printer or other serial device. The DISPLAY & MDS UNIT combination facilitates multiple remote display locations which mirror the information displayed on the ADI/MDI.

Command Message Destination

A full set of Commands is available to the user for both configuring and interrogating the ADI/MDI. With the COMMAND MESSAGE destination, it is possible for you to program any Command as a Message, resulting in a Command Message. The Command function specified by the Command Message is executed when the unit receives a request for that Message. You can request Command Messages from any of the Message request sources.



```
 HELP   CONFIGURATION   MESSA
                              C01 - MDI Unit Function CFG settings
 MESSAGE #:  1   SIZE:  7  FI C02 - Real Time Clock settings
                              C03 - Periodic Message settings
 DESTINATION: COMMAND MESSAGE C04 - Parallel Port CFG settings
 DESCRIPTION: Hit SPACE BAR to s C05 - Serial Port CFG settings
                              C06 - Output/BUSY Pin CFG settings
                              C07 - Elapsed Timer CFG settings
                              C08 - MDI/MDS Display CFG settings
                              C09 - Set/Transmit Embedded Data
                              C17 - Load Default Unit Configuration
                              C18 - Transmit Unit Configuration to MDS
                              C19 - Halt/Reset Unit
                              C20 - Cancel Message/Queue Entry
                              C21 - Display Message/Queue Entry
                              C22 - Transmit Queue Entry Information
                              C23 - Transmit Message Text
                              C24 - Transmit Character string
                 COMMAND DATA

 SPACE BAR displays COMMAND MESSAGE list
    1-HELP 2-PREV 3-NEXT 4-ADD 5-DEL 6-COPY 7-GOTO 8-SVMSG 9-DNLD 10-TERMINAL
ALT  PRINT                      UNDO              SV FILE UPLOAD  SIMULATOR
```

Figure 9, Command Message Destination

Commands are available for changing any of the Configuration settings on-line. Consequently, the unit can be programmed to adapt itself to changing system requirements and demands. Commands can also cause the ADI/MDI to transmit any of its current configuration settings, which can then be displayed on the SFMD's Terminal Emulator, or other ASCII terminal.

The actual Command function is specified by the selected Command number and description, combined with any required/desired data for the Command included in the Command Data area.

(See *COMMANDS*, page 75, for a list of the available Commands, Command structure and syntax, and the usage and operation of Commands.)

Message Time Out

You can specify a Time Out value for a Message. The Message Time Out value determines how long the unit displays the Message. The unit only runs the Time Out for a Message while that Message is on the display. Whenever the Message is bumped from the display to the Queue, the Time Out value is halted and the remaining value is saved on the Queue. The ADI/MDI then continues the Message's Time Out from this remaining value whenever the Message is placed back on the display. At the end of the Time Out, the Message is automatically canceled. At this point, if the Message is part of a Chained Message list, the ADI/MDI automatically requests the next message in the Chain.

The Message Time Out value can range from 10-1260 milliseconds, 0-63 seconds, or 1-63 minutes. You can also turn the function OFF, in which case the Message does not Time Out and must be explicitly cancelled to remove it from the Queue and/or the display.

(See *MESSAGE PROCESSING*, page 41, for details on the usage and operation of the Message Time Out feature.)

Message Scrolling

Three different methods for displaying Message text for a given line of the display are available. Separate scrolling techniques can be specified for the top and bottom lines of text for a Message and can configure the ADI/MDI's BLOCK and CHARACTER Scroll Times for separate top and bottom line values.

STATIC - The Message text characters do not scroll, but remain in place on the display.

BLOCK SCROLLING - The unit displays BLOCKs of text, which must be designated with the SEGMENT control code (s - <ALT-S>) in the Message Text area, one after another. Each BLOCK of text is displayed for the duration specified by the BLOCK Scroll Time for the particular line of the display on which it appears.

Once the last BLOCK of text for a line is displayed, the unit blanks that line for one BLOCK Scroll Time, and then repeats the text for the line from the beginning. Any BLOCK of text that exceeds the width of the display is truncated to the display width. All remaining characters for that BLOCK are ignored.

CHARACTER SCROLLING - The text advances one character position at a time, from right to left, across the display. The CHARACTER Scroll Time for a line determines the rate at which the text advances across the display.

Once the last character of a line scrolls off the display to the left, the text is repeated beginning from the right. You can designate BLOCKs of text for CHARACTER scrolling with the SEGMENT control code (s - <ALT-S>). In this case, the unit character scrolls each BLOCK of text separately across the screen.

Message Priority

A Message designated for DISPLAY must have a Priority assigned to it. If you do not desire a Prioritized scheme for a Message file, do not change the Priority field in any of the message records. The SFMD software automatically assigns a default Priority of 1 to all Messages. Unless this priority is changed, the ADI/MDI handles all Message requests on an equal basis since all their priorities are equal.

The Priority can be set from 1 (Highest Priority) to 255 (Lowest Priority). The unit determines whether to place a requested Message on the display based on the Priority of the requested Message and that of the displayed Message. If the Priority of the requested Message is the same or greater than that of the displayed Message, the unit places the requested Message on the display. Otherwise, the unit discards the requested Message after processing it, unless the message is designated for placement on the Queue.

The ADI/MDI determines the Queue position of a requested Message based on that Message's Priority. Messages are placed on the Queue in a Last-In/First-Out order within the Priority scheme. All Messages with the same Priority are grouped together on the Queue. These Priority groups are maintained in numerical order from high Priority (numerically low) to low Priority (numerically high). The most recently requested Message for a particular Priority group will be the first Message of the group to be placed on the display. Consequently, the most recently requested, highest Priority Message always appears at the top of the Queue.

When the message on the Display is cancelled and no other message requests are pending, such as a Chain, Trigger or Periodic request, the Message at the Top of the Queue (Queue Entry 0) is automatically placed on the display. As Queue Entries are created and deleted, Messages move closer to the Top or the Bottom of the Queue, based on their respective Priorities.

(See *MESSAGE PROCESSING*, page 41, for details on the handling of Prioritized Messages and the Message Queue.)

Message Queuing

The ADI/MDI contains an extensive Message Queue function. The Message Queue serves as a prioritized waiting list for requested Messages, and can hold up to 32 Queue Entries (0-31). This function allows requested Messages to be saved and recalled based on Priority order.

You can turn the Message Queue function ON or OFF, either in the Configuration File, or on-line through the use of Commands or Command Messages. No new Queue Entries are created while the Queue function is off, except for a temporary Entry it creates for every Message it places on the display. This temporary Entry is removed from the Queue once the Message is removed from the display.

With the Queue function ON, a Queue Entry is created for every DISPLAY Message the unit processes. You can also designate individual Messages for placement on the Queue. Only Messages destined for DISPLAY and designated for the Queue can actually be stored on the Queue while they are not on the display. Messages not designated for the Queue appear on the Queue only so long as they appear on the display. The ADI/MDI automatically removes any non-Queue Message from the Queue whenever it removes that Message from the display.

Messages intended for the MDS UNIT, TRANSMIT, and COMMAND MESSAGE destinations are processed upon request, and then discarded. They do not have a Priority and are never placed on the Queue.

(See *MESSAGE QUEUE*, page 43 , for details on the usage and operation of the Queue.)

MESSAGE QUEUE/PRIORITY EXAMPLE:

DSP	-	'—>' Message is currently Displayed 'NO' Message has never been Displayed 'YES' Message has been Displayed
Q#	-	Position on the Queue, 0 is Top
PR1	-	Message Priority
MSS	-	Message Number
CHN	-	'NO' Message is not part of a Chain
	-	'YES' Message is part of a Chain
DATA	-	Indicates how much data the Message collected
T-O REM	-	Indicates the Remaining Display Time of the Message

MESSAGE QUEUE CFG - ON

DSP	Q#	PRI	MSS	CHN	DATA	T-O	REM
—>	0	1	1	NO	NONE	NO	T-O
YES	1	2	2	NO	NONE	NO	T-O
YES	2	5	37	NO	NONE	2	sec
YES	3	7	12	NO	NONE	1	min.
NO	4	9	10	NO	NONE	20	sec
NO	5	22	55	YES	NONE	40	sec
NO	6	26	17	NO	NONE	30	sec

The ADI/MDI receives a request for Message 3, which is enabled for placement on the Queue and has a Priority of 3. The unit processes Message 3, and then creates a Queue Entry for the Message, as shown. Since Message 3 has a lower Priority than the Message currently on the display, Message 3 is not placed on the display at this time.

DSP	Q#	PRI	MSS	CHN	DATA	T-O	REM
—>	0	1	1	NO	NONE	NO	T-O
YES	1	2	22	NO	NONE	NO	T-O
NO	2	3	3	NO	NONE	45	sec
YES	3	5	37	NO	NONE	2	sec
YES	4	7	12	NO	NONE	1	min.
NO	5	9	10	NO	NONE	20	sec
NO	6	22	55	YES	NONE	40	sec
NO	7	26	17	NO	NONE	30	sec

Message 1, which is currently on the display, is cancelled. The ADI/MDI removes the Queue Entry for the cancelled Message. If no other Message requests are pending (Periodic, Trigger, etc.), the new Top Queue Entry (Entry 0) is placed on the display, as shown.

DSP	Q#	PRI	MSS	CHN	DATA	T-O	REM
—>	0	2	2	NO	NONE	NO	T-O
NO	1	3	3	NO	NONE	45	sec
YES	22	5	37	NO	NONE	2	sec
YES	3	7	12	NO	NONE	1	min.
NO	4	9	10	NO	NONE	20	sec
NO	5	22	55	YES	NONE	40	sec
NO	6	26	17	NO	NONE	30	sec

Message Color

If you have a Tri-Color unit, you can set the Display Backlight color in any DISPLAY Destination message. With the DISPLAY TYPE field in the FILE/UNIT INFO sub-menu, the ADI set for “2x20 POS TRI-COLOR”; and the MDI set for “2x20 NEG TRI-CLR”, the ADI and MDI displays the COLOR field for the various DISPLAY destinations.

To set the color for a Message, position the cursor on the COLOR field and press the space bar or click the mouse. The color selection list will appear. If you do not want to change the color in a Message, select “N/C”, for No Change.

Message Chaining

Only a Message designated for any of the DISPLAY destinations can initiate a sequence of Messages, called a Message Chain. Up to 32 Messages can be included in a Chained Message list. When a Message in the Message Chain expires (times out or is cancelled), the next Message in the Chain is automatically requested. This process is repeated for each Message in the Chain when the preceding Message expires. Consequently, a Message Chain defines a predetermined sequence of Messages and/or Command Messages.

The Priority of every Message in the Chain is the Priority of the Message containing the Chain List. This Chain Priority overrides the individual Message’s Priority whenever the Message is requested by the Chain.

If a new Message request interrupts a Message Chain, the unit saves the current Message in the Chain on the Queue. The Chain resumes from the current Message when the unit places that Message back on the display. If the current Message on the display is NOT enabled for the Queue, or the Queue function is turned OFF, the Message Chain is canceled whenever it processes a request for a higher Priority Message.

Canceling a Chained Message on the display with the “C20” (Cancel Displayed Message) Command string causes the unit to proceed to the next Message in the Chain. Issuing the “C20C” (Cancel Chain) Command string cancels both the current Message and the remainder of the Chain list. Canceling a Chained Message on the Queue, while not displayed, cancels the remainder of the Chain, as well.

You can loop Message Chains back onto themselves at any point in the Message Chain List, providing the user with the means for programming a repetitive sequence or loop of Messages and/or Command Messages. It is also possible to branch to other Message Chains. However, the unit only follows the most recent Chain. Once that Chain expires, the ADI/MDI does not return back to the original Chain.

You cannot directly specify a Message Chain in a Command Message. You can, however, use the Command Message LINK function to link another Message to the Command Message. Whenever the ADI/MDI encounters a LINKed Command Message, it executes the Command and then requests the Message specified at the end of the Command data string.

(See *COMMAND MESSAGE*, page 99, for details on the Command Message Link Function.)

Configuring the Output Pin

The ADI/MDI has one NPN open-collector Output Pin which you can assign to the handshaking BUSY Mode, or the Message and Command controlled OUTPUT Mode. With the Output Pin assigned to the OUTPUT Mode, you can program Messages to configure the pin. The unit processes the new configuration for the Output Pin when the Message is requested. If the ADI/MDI places the Message on the Queue without displaying it, it still configures the Output Pin first.

You can program a Message to configure the Output Pin by setting CONFIGURE OUTPUT PIN to “YES” and pressing ENTER or clicking the mouse with the cursor positioned on the CONFIGURE OUTPUT PIN field. You can program a Message to perform any or all of the following Output pin functions:

UNLOCK ON ENTRY: If the Output pin is LOCKed, the Message must first UNLOCK the pin before it can access any other pin functions, otherwise, any attempt to modify the pin’s configuration is ignored.

LOCK ON EXIT: The Output pin can be LOCKed from accepting any changes. You can have the unit LOCK the pin after it has performed the Output pin functions the Message specifies.

STATUS: The Message can set the Pin to the ON state, the OFF state, or it can TOGGLE the current state of the Pin from ON to OFF or OFF to ON. The actual ON and OFF states of the Output pin are determined by the Pin’s current Logic Level setting.

TIME OUT STATUS: The Output Pin contains a Time Out function. At the expiration of the Time Out value, the unit automatically turns the Output Pin OFF. You can have the Message RUN or HALT this Time Out function and set the Time Out value. HALTing the Time Out function freezes the Time Out at its current value. RUNning the Time Out function does not automatically turn the Output Pin ON, you must do this explicitly. If the Message RUNs the Time Out, you must also select one of the following three Time Out sources (DEFAULT, MESSAGE, RESUME).

SOURCE: If you select DEFAULT, the Time Out value from the default Output Pin Time Out Configuration setting is loaded and then the Time Out status is set to RUN.

If you select MESSAGE, the unit loads the Time Out value included in the Message record and then sets the Time Out status to RUN. The Message can specify the Output Time Out in the range from 10 to 1260 milliseconds, 1 to 63 seconds, or 1 to 63 minutes.

If you select RESUME, the unit RUNs the Time Out from its current value. If the value has already Timed Out, the Time Out status immediately reverts to HALT.

If you do not desire to change a particular Output Pin function for a Message, simply select “N/C” or “NO CHANGE” for that function.

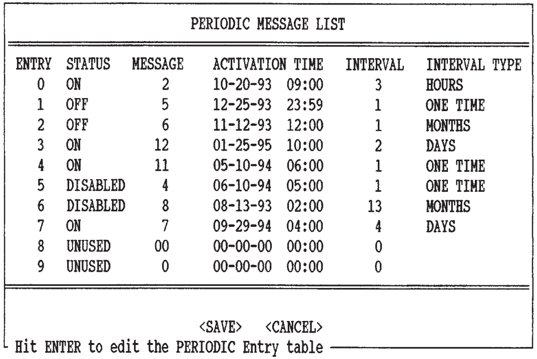
(See *OUTPUT PIN*, page 54, for details on the usage and operation of the Output Pin Function.)

Periodic Messages

The ADI/MDI contains an extensive Periodic Message function, which holds up to 32 separate Periodic Entries. The Periodic function allows you to specify individual Messages for Periodic request over a variety of intervals, such as once per minute, once every 8 hours, once per month, etc.

If the Message being configured is already assigned to one or more Periodic Entries, the PERIODIC/DATED field indicates “Yes” and the ENTRIES field indicates the number of Entries that contain the Message. Individual Messages can be assigned to Periodic Entries in the PERIODIC MESSAGE LIST sub-menu. This sub-menu is accessed by positioning the cursor on the PERIODIC/DATED field and hitting the SPACE BAR, or clicking the mouse. Each Periodic Entry specifies an ENABLE/DISABLE flag, an ON/OFF flag, a Message number, an Activation Time, and a Periodic Interval. A programmed Message can be assigned to any number of Periodic Entries. The Activation Time specifies the Date and Time that the ADI/MDI should next request the Message assigned to the Periodic Entry. The Periodic Interval specifies the frequency that the unit should request the Message.

(See *PERIODIC MESSAGE*, page 67, for details on the usage and operation of the Periodic Message Function.)



```
                         PERIODIC MESSAGE LIST
========================================================================
ENTRY  STATUS    MESSAGE    ACTIVATION TIME    INTERVAL    INTERVAL TYPE
  0    ON           2       10-20-93  09:00        3       HOURS
  1    OFF          5       12-25-93  23:59        1       ONE TIME
  2    OFF          6       11-12-93  12:00        1       MONTHS
  3    ON          12       01-25-95  10:00        2       DAYS
  4    ON          11       05-10-94  06:00        1       ONE TIME
  5    DISABLED     4       06-10-94  05:00        1       ONE TIME
  6    DISABLED     8       08-13-93  02:00       13       MONTHS
  7    ON           7       09-29-94  04:00        4       DAYS
  8    UNUSED      00       00-00-00  00:00        0
  9    UNUSED       0       00-00-00  00:00        0
========================================================================

                         <SAVE>    <CANCEL>
Hit ENTER to edit the PERIODIC Entry table
```

Figure 10, Periodic Message List

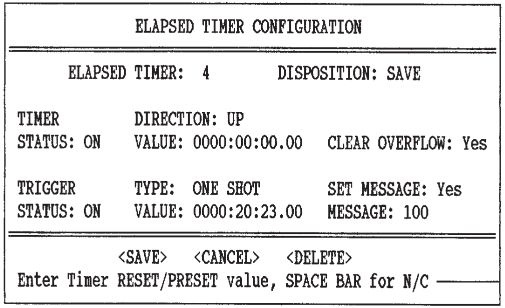
Note: The list above shows entries typical of an ADI/MDI application.

Configuring Elapsed Timers

The ADI/MDI contains sixteen separate, fully configurable Elapsed Timers. Messages can access, configure, and display multiple Elapsed Timers. You can also access and configure any Elapsed Timer with the C07 Command. The unit constantly updates all Elapsed Timer values according to the current configuration for each timer.

You can display an Elapsed Timer value in a Message by inserting the ELAPSED TIMER control code (e - <ALT-E>) followed by the two digit timer number, the appropriate Time format codes, and a closing control code (e) in the Message Text. If a displayed Timer is running, the running value is displayed as it changes. (See *MESSAGE TEXT*, page 32, for details on specifying an Elapsed Timer Value in the Message Text.)

You program a Message to configure an Elapsed Timer by setting CONFIGURE ELAPSED TIMERS to “Yes” and hitting ENTER or clicking the mouse with the cursor positioned on the desired Elapsed Timer number. A Message can access any of the following Elapsed Timer functions through the Elapsed Timer Configuration sub-menu:



```
                  ELAPSED TIMER CONFIGURATION
===============================================================
      ELAPSED TIMER:  4         DISPOSITION: SAVE

TIMER         DIRECTION: UP
STATUS: ON    VALUE: 0000:00:00.00   CLEAR OVERFLOW: Yes

TRIGGER       TYPE:  ONE SHOT        SET MESSAGE: Yes
STATUS: ON    VALUE: 0000:20:23.00   MESSAGE: 100
===============================================================
              <SAVE>   <CANCEL>   <DELETE>
Enter Timer RESET/PRESET value, SPACE BAR for N/C
```

Figure 11, Elapsed Timer Configuration

ELAPSED TIMER: This field indicates the Elapsed Timer you are configuring.

DISPOSITION: On every unit Power-Up and Reset, the ADI/MDI configures the Elapsed Timer based on the Timer's Disposition setting (SAVE or CLEAR). With the Timer's Disposition set for SAVE, the unit maintains the Timer configuration on Power-Up or Reset (i.e.. if the Timer was running when the unit powered-down, the unit runs the Timer when power is restored to the unit).

With the Disposition set for CLEAR, the ADI/MDI resets the Timer configuration to the factory default settings on Power-Up or Reset (i.e. the unit halts the Timer and clears the Timer value to zero).

TIMER STATUS: You can turn the Timer ON (run) or OFF (halt).

DIRECTION: A Timer can run UP or DOWN. You can change the direction of the Timer without halting or resetting the Timer.

VALUE: You can preset the Timer to any value within the range from 0000:00:00.00 to 9999:59:59.99.

CLEAR OVERFLOW: A Timer value rolls-over on Overflow or Underflow and a flag is set to indicate this action. You can have the Message reset this flag, if necessary.

TRIGGER STATUS: Each Timer includes a Trigger function, which causes the unit to request the specified Trigger Message whenever the Timer's value reaches the Trigger value. You can turn the Trigger function ON or OFF.

TYPE: There are two different types of Triggers, ONE-SHOT and RETRIGGER. With the Trigger function set for ONE-SHOT, the ADI/MDI requests its Trigger Message when the Timer reaches the Trigger value and the Trigger Status is ON. The unit then turns OFF the Trigger function and continues to run the Timer. This feature is useful for generating automatic Message requests after a specified time delay.

With the Trigger function set for RETRIGGER, the ADI/MDI requests its Trigger Message when the Timer reaches its Trigger value and the Trigger Status is ON. The unit then resets the Timer value to ZERO, continues to run the Timer, and leaves the Trigger Status ON. This feature is useful for generating automatic Message requests on a repetitive basis.

Note: The Timer does NOT reset to the PRESET value.

VALUE: You can set the Trigger to any value in the range from 0000:00:00.00 to 9999:59:59.99

SET MESSAGE: With the Trigger Status ON, the ADI/MDI requests the Trigger Message whenever the Timer value reaches the Trigger value. You can set or change the assigned Trigger Message by selecting "Yes" in this field, and then entering the new Trigger Message number.

MESSAGE: You can assign any Message as the Trigger Message for a Timer. You can assign the same Message to more than one Elapsed Timer.

If you do not desire to make a change to a particular Elapsed Timer feature, simply select "N/C" or "NO CHANGE" for that feature.

(See *ELAPSED TIMERS*, page 48, for details on the usage and operation of the Elapsed Timers.)

Collecting Embedded Data

The ADI/MDI contains comprehensive and flexible Embedded Data collection, storage, manipulation, and test features. It collects Embedded Data according to the specifications contained in the Message record it is currently processing, and through data sent to it with the C09 Command ("I" mnemonic). All collected data is stored in the non-volatile memory. The amount of Embedded Data the ADI/MDI can hold at one time is determined by the size of the current Message File, all memory that remains after a Message File is downloaded to the unit constitutes the Embedded Data Storage area

You can view the Embedded Data Storage area size for a Message File in the File/Unit Information sub-menu. You can access the current Embedded Data Storage area statistics for a given unit with the "C09S" Command string. The statistics are transmitted as shown:

```
ED STORAGE AREA
MAXIMUM:    29940
CURRENT:    00000
PEAK:       00000
```

The MAXIMUM statistic shows the maximum Embedded Data Storage area, in bytes. The CURRENT statistic shows the amount of Storage space that is currently occupied. The PEAK statistic shows the largest amount of Storage space that has been occupied at any one time, since the last Message File download or Reset Command. This value is reset to 0 on every Message File download, and you can reset it with the "C09SR" Command string. These statistics are helpful for determining the available memory for a particular application. (See *COMMAND C09*, page 89, for details on the C09SR Command String.)

The Parallel Port can be used to interrupt the ADI/MDI while it is collecting data for a Message. In case of an interruption, the unit deletes all data it has acquired for the Message up to the point of interruption, and abandons further processing of that Message. The unit then proceeds to process the interrupting Message request.

You can program a Message to collect up to fifteen separate Embedded Data (ED) Items. You must assign each ED Item as either a Local Item or an Indexed Item. This data item type, either Local or Indexed, determines the scope and accessibility of that item. The ADI/MDI collects the ED Items for a Message in the order they appear on the Message Editing screen, starting with Item 1. Both Local and Indexed Items can be collected in the same Message, but the unit must collect all the Local ED Items for a Message before it collects any Indexed ED Items, thus the Local Items always appear in the Message Editing screen with the lowest ED Item numbers.

You program a Message to collect Embedded Data by first selecting "Yes" for the COLLECT EMBEDDED DATA field, and then by entering the number of LOCAL and/or INDEX items you wish to collect. As you are entering these numbers, a configuration field appears on the screen for each ED Item specified.

You configure each item by positioning the cursor on the respective Item number and hitting the SPACE bar or clicking the mouse. The Embedded Data Item Configuration sub-menu for that item then appears.

Local Embedded Data

Only the Message collecting a Local Item can display that item, and the ADI/MDI cannot perform any data manipulations or comparisons on the item. A single Local Embedded Data Record is created containing all of the Local Embedded Data Items a Message collects and this record is stored in the Embedded Data Storage area while the Message appears on the Queue. A Local Embedded Data Record can hold up to 254 bytes, including certain control bytes the ADI/MDI uses to locate and separate individual items within the record.

When the unit removes a Message from the Queue (through Message time out or cancellation), it deletes the Local Embedded Data Record for that Message. Consequently, Local Embedded Data Items are Message specific and temporary.

You can use the C09 Command to access, but not change, any of the Local Embedded Data Records and Items an ADI/MDI unit may contain. (See *COMMAND C09*, page 89, for details.)

Indexed Embedded Data

Indexed Embedded Data Items offer a more permanent, flexible, and accessible data structure. One Index can store a single data item and the ADI/MDI can store a maximum of 96 Indexed Items, numbered from 0 to 95. An Indexed Item can hold up to 126 bytes of data, plus a length byte and a terminator byte.

When a Message collects an Indexed Data Item, the unit deletes the existing data for the Index and stores the new data at the assigned Index location.

Commands and other Messages can then access the stored data by its Index Number. Any Message can collect, update, and display any Indexed Embedded Data Item.

You can use Indexed values in numerous applications throughout the unit. You can insert the Index value in Transmit Request strings; you can reformat and combine Index values through the Internal Data source; you can include Index values as variables in Command Data strings, etc. You can also increment and decrement the numeric values in Indexed Items, as well as perform various numeric and alphanumeric comparisons which generate conditional Message requests based on the comparison results.

(See *INDEXED DATA ITEMS* and *COMMAND C09* for details on the usage and manipulation of Indexed Embedded Data.)

Embedded Data Sources

When configuring an ED Item in the Embedded Data Item Configuration Screen, first select the source of the data for that item. Choose from among the following four sources:

SERIAL PORT: The data for an ED Item can be received over the Serial Port as an ASCII character string. Specify this source when collecting data from RLC products with serial communications, or from other serial devices.

DISCARD: The ADI/MDI handles the DISCARD data source as a special case of the SERIAL PORT data source. The unit collects the data over the serial port, but discards the characters as they are received. Specify this source whenever a Message collects multiple ED Items from a single serial character string, such as from an RLC “Print Request” function, and you do not need to store all of the data items received.

PARALLEL PORT: The ADI/MDI can receive an ED Item over the Parallel Port in ASCII, BCD, or Binary formats, as specified by the individual ED Item configuration. Specify this source when collecting data from a PLC, thumb-wheel switch, or other device connected to the Parallel Port.

INTERNAL: The ADI/MDI can assemble the ED Item from any combination of the following internal sources; the Current Time/Date, multiple Elapsed Timer values, multiple Index values, and text characters, as specified by the individual ED Item configuration. Specify this source whenever you need to capture a time stamp of the Current Time or an Elapsed Timer value, when you need to combine Index values with other Index values and/or time values, when you need more than one form of an Index value (with and without units, for example), or any time you need to assemble a specific character string and save it as data.

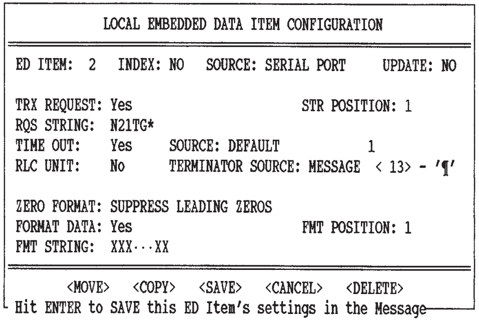
Multiple ED Item configurations for a Message can specify different data sources. In other words, you can program the ADI/MDI to collect data from the Serial Port while discarding some of the serial data, from the Parallel Port, and from any of the Internal sources, all within a single Message.

You can also configure one or more ED Items to collect data, storing the data in different Index locations, and configure other ED Items to combine and reformat those same Index values, storing the results in additional Index locations, all within a single Message.

See *EMBEDDED DATA COLLECTION FLOWCHART* at the end of this section for a graphical description of the various operations and functions of the Embedded Data Sources.

Serial Port

The ADI/MDI can receive data for an Embedded Data Item over the Serial Port. You can configure the ED Item to issue a serial transmission before accepting the data and you can select an individual serial terminator character for the item. You can specify how long the unit should wait to receive data over the Serial Port. You have a choice of several different types of leading zero formatting, and you can format (keep or delete) any combination of the characters received.



```
               LOCAL EMBEDDED DATA ITEM CONFIGURATION
══════════════════════════════════════════════════════════════════
ED ITEM:  2   INDEX: NO   SOURCE: SERIAL PORT     UPDATE: NO

TRX REQUEST: Yes                       STR POSITION: 1
RQS STRING:  N21TG*
TIME OUT:    Yes     SOURCE: DEFAULT            1
RLC UNIT:    No      TERMINATOR SOURCE: MESSAGE  < 13> - '¶'

ZERO FORMAT: SUPPRESS LEADING ZEROS
FORMAT DATA: Yes                       FMT POSITION: 1
FMT STRING:  XXX···XX
══════════════════════════════════════════════════════════════════
       <MOVE>   <COPY>   <SAVE>   <CANCEL>   <DELETE>
Hit ENTER to SAVE this ED Item's settings in the Message
```

Figure 12, Serial Port Embedded Data Sub-menu

Descriptions of each of the fields in the Serial Port Embedded Data sub-menu are presented below.

ED ITEM: This field indicates the ED Item you are currently configuring.

INDEX: If you have configured the ED Item as Indexed data, you set the Index number, from 0 to 95, in this field. If the ED item is configured as Local data, this field is set to "NO". (*See Assigning An Index Number* under EMBEDDED DATA FUNCTIONS later in this section for details.)

SOURCE: You select the data source for the ED Item you are currently configuring in this field, in this case "SERIAL PORT".

UPDATE: If you have configured the Message to update its data, you can select updating for each individual ED Item. If the Message does not update its data, this field is set to "NO". (See *Updating Data* under EMBEDDED DATA FUNCTIONS later in this section for details.)

TRX REQUEST: You can have the ADI/MDI transmit a character string of your choosing before it collects the data. This feature is useful for issuing transmissions that request data from other serial units on the loop, such as RLC Gemini's, TCU's, IM's, etc. (See *Using The Transmit Request Function* under EMBEDDED DATA FUNCTIONS later in this section for details.)

STR POSITION: This field indicates your current position in the RQS STRING field.

RQS STRING: You enter the actual request string to be transmitted in this field. The character string can be up to 127 characters long, and you can include Current Time, Elapsed Time, and Index values, as well as text characters, lower ASCII control codes, and serial transmit delays.

TIME OUT: You can have the unit wait a specified amount of time to receive the data over the Serial Port. The unit accepts characters over the port as data until it receives the appropriate Serial Terminator character (see below), or until the Time Out value you choose expires, whichever occurs first. If you do not select the Time Out function, the unit waits indefinitely for the Terminator character. (See *Using The Embedded Data Time Out Function* under EMBEDDED DATA FUNCTIONS later in this section for details.)

SOURCE: If you select the Time Out function, you must then choose between two sources (DEFAULT or MESSAGE) for the Embedded Data Time Out value.

Selecting DEFAULT causes the Embedded Data Time Out for the item to be set to the Serial Port Time Out value contained in its Configuration File.

Selecting MESSAGE allows you to specify a time out value (from 1 to 254 seconds) for the particular item, or you can set the time out to 0, in which case the unit does not wait to receive any data. This feature is useful if you are transmitting a character string to an RLC unit that does not elicit a response, such as a set point change.

VALUE: If you select the MESSAGE time out source, you must enter the desired time out value for the ED Item in this field.

RLC UNIT: This field is reserved for future use and its setting is ignored by the ADI/MDI at this time.

Embedded Data Sources (Cont'd)

Serial Port (Cont'd)

TERMINATOR SOURCE: You can select the source of the Serial Terminator character (DEFAULT or MESSAGE) that the unit should recognize as the end of a serial data transmission.

Selecting DEFAULT causes the ADI/MDI to wait for the Serial Terminator character contained in its Configuration File.

Selecting MESSAGE allows you to specify the Terminator character (from 00h (0) to 07h (7), 09h (9) to 1Ah (26), and 1Ch (28) to FFh (255)) for the particular item in the Message.

(See *Serial Terminator Character*, page 70, for details.)

ZERO FORMAT: You can choose between three formats (BLANK LEADING ZEROES, SUPPRESS LEADING ZEROES, or KEEP LEADING ZEROES) the unit uses to process the leading zeroes of any numeric fields for a particular item. (See *Formatting Leading Zeroes* under EMBEDDED DATA FUNCTIONS later in this section for details.)

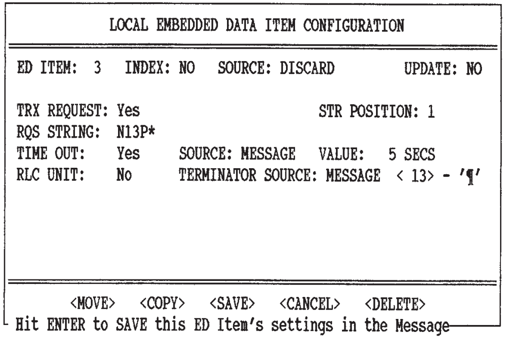
FORMAT DATA: You can have the ADI/MDI keep or delete any combination of characters received for the item. The unit performs data formatting after it has formatted the leading zeroes for any numeric fields in the data string. If you choose not to format the data, the unit keeps all the remaining characters after having performed leading zero formatting. (See *Formatting Data* under EMBEDDED DATA FUNCTIONS later in this section for details.)

FMT POSITION: This field indicates your current position in the FMT STRING field.

FMT STRING: Enter an 'X' for each character position you wish to delete, and a ' ' (SPACE) for each character position you wish to keep.

Discard

You can have the ADI/MDI collect an item over the Serial Port and then discard the data, without saving it. This function is useful when you need to transmit a string, but aren't interested in the response, or when you are receiving multiple data items in a single transmission, such as from an RLC Print Request, and you are only interested in keeping certain items from the string. In such a case, you collect the items you wish to keep with the Serial Port source, and collect the items you wish to discard with the Discard source.



```
               LOCAL EMBEDDED DATA ITEM CONFIGURATION

ED ITEM:  3   INDEX: NO   SOURCE: DISCARD        UPDATE: NO

TRX REQUEST: Yes                          STR POSITION: 1
RQS STRING:  N13P*
TIME OUT:    Yes     SOURCE: MESSAGE   VALUE:   5 SECS
RLC UNIT:    No      TERMINATOR SOURCE: MESSAGE  < 13> - '¶'

       <MOVE>   <COPY>   <SAVE>   <CANCEL>   <DELETE>
Hit ENTER to SAVE this ED Item's settings in the Message
```

Figure 13, Discard Data Embedded Data Sub-menu

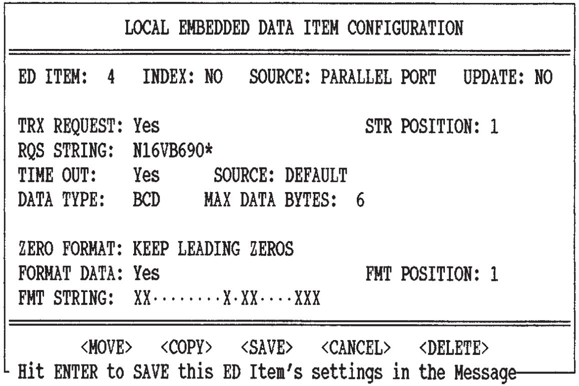
Descriptions for each of the fields in the Discard Data Embedded Data sub-menu are presented below.

ED ITEM: Same as for SERIAL PORT Data.
INDEX: Since the unit does not keep the data for this item, this field is always set to "NO".
SOURCE: Set for DISCARD.
UPDATE: Same as for SERIAL PORT Data.
TRX REQUEST: Same as for SERIAL PORT Data.
RQS STRING: Same as for SERIAL PORT Data.
STR POSITION: Same as for SERIAL PORT Data.
TIME OUT: Same as for SERIAL PORT Data.
RLC UNIT: Same as for SERIAL PORT Data.
TERMINATOR SOURCE: Same as for SERIAL PORT Data.

Since the unit discards the received ED Item, there is no need for formatting the data. Consequently, the format functions are not included for the Discard Data source.

Parallel Port

The ADI/MDI can receive data for an Embedded Data Item over the Parallel Port. You can configure the item to issue a serial transmission before accepting the data. You can choose between ASCII, BCD, and Binary formats for the data, and you must specify the maximum number of bytes the unit should expect to receive. You can also specify how long the unit should wait to receive the data. You have a choice of several different types of leading zero formatting, and you can format (keep or delete) any combination of the characters received.



```
              LOCAL EMBEDDED DATA ITEM CONFIGURATION

ED ITEM:  4   INDEX: NO   SOURCE: PARALLEL PORT   UPDATE: NO

TRX REQUEST: Yes                       STR POSITION: 1
RQS STRING:  N16VB690*
TIME OUT:    Yes      SOURCE: DEFAULT
DATA TYPE:   BCD     MAX DATA BYTES:  6

ZERO FORMAT: KEEP LEADING ZEROS
FORMAT DATA: Yes                       FMT POSITION: 1
FMT STRING:  XX········X·XX····XXX

       <MOVE>   <COPY>   <SAVE>   <CANCEL>   <DELETE>
Hit ENTER to SAVE this ED Item's settings in the Message
```

Figure 14, Parallel Port Embedded Data

Descriptions for each of the fields in the Parallel Port Embedded Data sub-menu are presented below.

ED ITEM: Same as for SERIAL PORT Data.
INDEX: Same as for SERIAL PORT Data.
SOURCE: Set for PARALLEL PORT.
UPDATE: Same as for SERIAL PORT Data.

TRX REQUEST: Same as for SERIAL PORT Data. The Transmit Request function, in this case, can be used for transmitting character strings describing operator actions, and/or Parallel Port and unit activity to a serial printer or ASCII terminal. See *Using The Transmit Request Function* under EMBEDDED DATA FUNCTIONS later in this section for details.)

STR POSITION: Same as for SERIAL PORT Data.

RQS STRING: Same as for SERIAL PORT Data.

TIME OUT: You can have the unit wait a specified amount of time to receive the data over the Parallel Port. The unit accepts data over the port until it has received the specified MAX DATA BYTES (see below), or until the Time Out value you choose expires, whichever occurs first. If you do not select the Time Out function, the unit waits indefinitely to receive the programmed number of bytes. (See *Using The Embedded Data Time Out Function* under EMBEDDED DATA FUNCTIONS later in this section for details.)

SOURCE: Same as for SERIAL PORT Data except the Parallel Port Time Out value is used for the DEFAULT source instead of the Serial Port value.

VALUE: Same as for SERIAL PORT Data.

DATA TYPE: You select the type of data (ASCII, BCD, or BINARY) that the ADI/MDI is to collect over the Parallel Port in this field.

MAX DATA BYTES: You must specify the maximum number of bytes the unit should expect to receive for this item. The Parallel Port width (4 or 8 bits) set in the Configuration File determines the number of strobes per data byte (1 strobe per byte for an 8-bit width, 2 strobes per byte for a 4 bit width). The MAX DATA BYTES setting informs the unit how many bytes of data to expect, not the number of strobes. (See *Receiving Embedded Data*, page 64, for details.)

ASCII: 1 to 23 bytes (ASCII characters).
BCD: 1 to 11 bytes (2 to 22 digits).
BINARY: 1 to 2 bytes (8 or 16 bits, max. value 65535).

ZERO FORMAT: Same as for SERIAL PORT Data.

FORMAT DATA: Same as for SERIAL PORT Data.

FMT STRING: Same as for SERIAL PORT Data.

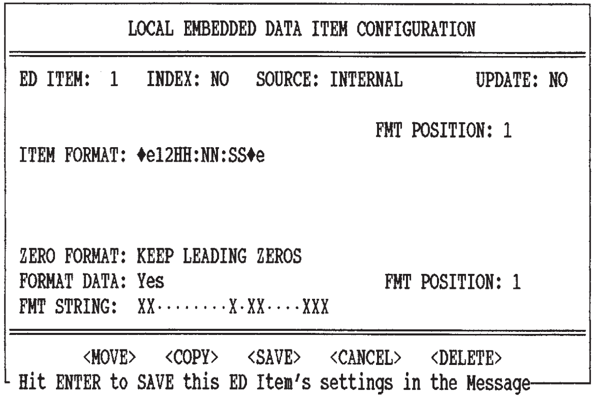
FMT POSITION: Same as for SERIAL PORT Data.

Embedded Data Sources (Cont'd)

Internal

You should specify the Internal Data source whenever you need to capture a Current Time or Elapsed Time value, initialize an Index value or merge multiple Index values into another Index, and anytime you need to assemble a character string which includes any of the above values.

You enter the format of an Internal Item in the ITEM FORMAT field using the same control and format codes for time and Index values as those you would use to specify a Message's text, see Figure 15, Internal Data Source. If the Item Format string specifies a Current Time or Elapsed Time value, the ADI/MDI captures the relevant time value and inserts it in the string, at the location and according to the format you specify. This feature is useful for "time stamping" collected data. See *MESSAGE TEXT*, page 32, for details on control and format codes.



```
              LOCAL EMBEDDED DATA ITEM CONFIGURATION

ED ITEM:  1   INDEX: NO   SOURCE: INTERNAL        UPDATE: NO

                                       FMT POSITION: 1
ITEM FORMAT: ♦e12HH:NN:SS♦e



ZERO FORMAT: KEEP LEADING ZEROS
FORMAT DATA: Yes                        FMT POSITION: 1
FMT STRING:  XX········X·XX····XXX

     <MOVE>   <COPY>   <SAVE>   <CANCEL>   <DELETE>
Hit ENTER to SAVE this ED Item's settings in the Message
```

Figure 15, Internal Data Source

ED ITEM: Same as for SERIAL PORT Data.
INDEX: Same as for SERIAL PORT Data.
SOURCE: Set to INTERNAL.
UPDATE: Same as for SERIAL PORT Data.
ITEM FORMAT: The Item Format string can be up to 127 characters long and is entered similar to Message text. The Item Format specifies how the unit should assemble the data item, and can be composed of the Current Time, Elapsed Time, and Index values, as well as characters and lower ASCII control codes. (See *Formatting Transmit Request and Internal Items* under EMBEDDED DATA FUNCTIONS later in this section for details.)

FMT POSITION: This field indicates your current position in the Item Format string.

ZERO FORMAT: Same as for SERIAL PORT Data. Since you can include existing Index values in the Item Format for the Internal Data source, you can use the Leading Zero Formatting function to reformat existing Index values. (See *Formatting Leading Zeroes* under EMBEDDED DATA FUNCTIONS later in this section for details.)

FORMAT DATA: Same as for SERIAL PORT Data. Since you can include existing Index values in the Item Format for the Internal Data source, you can use the Format Data function to reformat existing Index values. Therefore, if you need to have more than one form of an Index value, you can use the Internal Data source to create and format the various formats. (See *Formatting Data* under EMBEDDED DATA FUNCTIONS later in this section for details.)

FMT POSITION: Same as for SERIAL PORT Data.
FMT STRING: Same as for SERIAL PORT Data.

Since the unit already contains the data for the Internal Data source, a Transmit Request String and a Time Out value are not necessary for an Internal item.

Embedded Data Functions

The following functions are common to two or more of the Embedded Data Sources.

Assigning an Index Number

An Index number specifies the Index location at which the ADI/MDI should store the collected data for an ED Item. The value of an Indexed item is available to every Message and Command programmed in a unit through the Index number. You can specify this number in Transmit Request strings, Internal Item Format strings, Message Text, and Command and Command Message data strings. Whenever the unit encounters an Index number while processing one of these character strings, it inserts the current Index value at the specified location in the string.

(See the *MESSAGE TEXT, FORMATTING TRANSMIT REQUEST STRINGS AND INTERNAL ITEMS*, and *COMMAND MESSAGES* sections for details.)

Updating Data

You can program a Message so that the ADI/MDI periodically and selectively updates the ED Items for that Message while the Message is on the display. To select the Update function for a Message, you must first set the UPDATE DATA field to "Yes" and select an Update Time value ranging from 10 to 1260 milliseconds, 0 to 63 seconds, and 1 to 63 minutes. You must then set the UPDATE field to "Yes" in the Embedded Data Configuration sub-menu for each ED Item you wish to have the unit update.

The ED Update Time starts when the unit places the Message on the display. The unit restarts the ED Update Time each time a data update comes due, and then commences to collect the new data. The unit collects the chosen data items in numerical order, from low to high.

You can have the data items for a Message continuously updated by setting the Update Time to 0 seconds. With continuous update, the unit begins to collect new data for a Message as soon as it has completed processing the latest update it requested.

Using the Transmit Request Function

The Serial Transmit Request function allows you to program a Transmit Request string that the ADI/MDI transmits before collecting a data item. The string can include text characters, Current and Elapsed Time values, Index values, lower ASCII control codes, and/or Transmit Delays, as needed.

Depending on your application, you may need to include Transmit Delays in the Transmit Request string. You insert a Transmit Delay in the Request string by placing the TRX DELAY control code (' t' - <ALT-T>) at the desired location(s). The unit pauses the Transmit Buffer for one Transmit Delay time whenever it encounters a TRX DELAY control code in the Buffer.

You can insert the <ESC> character (1Bh-27) in the Transmit Request String. When the unit encounters the <ESC> character in the Transmit Buffer, it clears the contents of its Receive Buffer at that point in the transmission. You can use this feature to accommodate difficult serial timing constraints. If you need to transmit the <ESC> character, insert the ' ' character in the Request string where necessary. When the unit encounters the ' ' character in a transmission, it transmits the <ESC> character.

You can use the Transmit Request string to request data transmissions from other serial units on the loop.

Example: To collect the RATE from an RLC Legend with address 15, you would program the following Transmit Request string:

♦tN15TH

The first, or leading, asterisk causes all RLC units on the loop to clear their respective Receive buffers. The TRX DELAY (t) control code sequence causes the ADI/MDI to run a Transmit Delay, the length of which is programmed in the unit's Configuration File. The unit pauses the transmission while it runs this delay, thus granting the other units on the loop time to clear their buffers after receiving the leading '*'. The "N15TH*" character string causes unit 15 to transmit its current "TH" value. The unit automatically runs another Transmit Delay after it transmits the last character of a Transmit Request string.

Upon receiving the "N15TH*" character string, unit 15 would then transmit the requested value, which the ADI/MDI would receive and accept as data for the item.

Embedded Data Functions (Cont'd)

Using the Transmit Request Function (Cont'd)

You can also use the Transmit Request function to change programmable settings in RLC units "on-line".

Example: To change the setpoint value on an RLC TCU with address 2, you would program the following Transmit Request string:

♦tN2VB90.0

On receipt of this string, the TCU changes its Setpoint value to 90.0. In this case, you could use the DISCARD data source with a time out value of 0, since the unit does not transmit a response to this string.

Another use for the Transmit Request function would be to transmit "on-line" information, such as instructions to machine operators, character strings to serial printers, and messages for display on ASCII terminals. If desired, you could even configure the item to wait for a response from the operator receiving the transmission before proceeding with the Message.

Example: You want to issue start-up instructions to a machine operator and wait for the operator's response to each instruction before proceeding to the next instruction. You could program a series of data items, each issuing a Transmit Request, which the operator views on an RLC slave unit, or an ASCII terminal. Each item waits for the operator's response, either serially with a key stroke, or over the Parallel Port with a push-button.

You could program a Message with four ED Items, each issuing a Transmit Request, that would transmit the following instruction sequence to an MDS unit configured for the RLC Mode with unit address 4:

N4M:Start Pump A<CR>
ADI/MDI waits for operator response.

N4M:Verify Pump A oil pressure >100 psi<CR>
ADI/MDI waits for operator response.

N4M:Start Pump B<CR>
ADI/MDI waits for operator response.

N4M:Verify Pump B oil pressure >100 psi<CR>
ADI/MDI waits for operator response.

The "N4M:" character sequence informs MDS unit 4 that a Message string follows. The MDS then displays the included string. The "<CR>" sequence represents the Carriage Return character (0Dh - 13), which the MDS has been programmed to accept as its Serial Terminator.

Formatting Transmit Requests and Internal Items

The ADI/MDI processes both of these strings similarly, with the only difference being that an Internal Item string is eventually stored as data, while a Transmit Request string is eventually transmitted over the Serial Port. You can include Current Time and Elapsed Time values, Indexed Item values, lower ASCII control codes, Serial Receive Buffer control codes, and Transmit Delays in Transmit Request strings and Internal Item Format strings.

As with the Message Text area, you can insert control codes in either of the strings to cause the ADI/MDI to insert any of the previously mentioned values in the string before storing or transmitting the string. (See *MESSAGE TEXT*, page 32, for details on usage of the various Message Text control codes.)

Examples: Given the following Current Time, Elapsed Time, and Index values:

Current Time: Wednesday, August 5, 1993 09:17:27
Elapsed Timer 13: 0000:14:37:23.79
Index 00: "23"
Index 01: "78 F
Index 10: "127
Index 11: "123
Index 12: "132

The ADI/MDI would construct the Transmit Request strings and Internal Data items as shown below:

```
TRX REQUEST: "*   t   tN   i00TA*"
      RESULT: "*   t   tN23TA*"
TRX REQUEST:"<CR><LF>   cHA:NN P   c Temp 12:    i01<CR><LF>"
      RESULT: "<CR><LF> 9:17 AM  Temp 12: 78 F<CR><LF>"
   ITEM FMT: "   cDDMAY2 - HH:NN:SS   c    i01"
      RESULT: " 5AUG93 - 09:17:27  78 F"
   ITEM FMT: "   e13HL:NN   e  Unit A:    i10 psi  Unit B:    i11 psi"
      RESULT: "14:37  Unit A: 127 psi  Unit B: 123 psi"
```

Using the Embedded Data Time Out Function

The Embedded Data Time Out function acts as a watch-dog timer for both the Serial Port and the Parallel Port while the unit is collecting data for a Message. A port's time out value specifies how long the unit should wait to receive data over that port before continuing to process a Message.

The separate Serial and Parallel Port Time Out values are contained in the Configuration File. You can turn either function OFF, in which case the unit waits indefinitely for a data item over that port, unless you interrupt the unit with a Message request.

With the time out function turned ON for a port, you can program the time out value for that port in the range from 0 to 254 seconds. A 0 second time out value causes the port to immediately close and not wait for a response. This feature is only useful for the Serial Port when you are issuing Transmit Requests and are not receiving a response from the transmission.

While collecting data over the Serial Port, the unit stores all characters it receives in its Receive Buffer as data until either of the following events occurs.

If the unit receives the designated Serial Terminator for the ED Item, it closes the port (does not accept any more characters) and begins processing the Receive buffer contents as data. If the Serial Port Time Out value expires before the unit detects the Serial Terminator, it closes the port and checks the contents of its Receive buffer. If the buffer contains any characters, the unit processes the Receive buffer contents as data.

If the buffer is empty (no characters were received), the four dashes "----" are stored in the ED Item, indicating that the port timed out and no data was received.

While collecting data over the Parallel Port, the ADI/MDI stores all bytes it receives in its Receive Buffer as data until either of the following events occurs.

If the unit receives the designated MAX DATA BYTES, as specified in the ED Item's configuration, it closes the port and begins processing the Receive Buffer contents as data, according to the DATA TYPE specified in the item's configuration. If the Parallel Port Time Out expires before the proper number of bytes is received, the unit closes the port and checks the contents of its Receive Buffer. If the buffer contains any data bytes, the unit processes the buffer contents as just described.

If the buffer is empty, however, the four dashes "----" are inserted in the ED Item, indicating that the port timed out, and no data was received.

If, at any time during data collection, the ADI/MDI detects a valid STROBE pulse with the MESSAGE/DATA pin set to the MESSAGE level, it immediately interrupts collecting the current data item and aborts processing the remainder of the Message. The current Receive Buffer contents are lost as a result of the interruption, and the unit accepts the data on the Parallel Port as a new Message request.

You can use the Embedded Data Time Out function in a number of different ways. If the ADI/MDI regularly interrogates various units on the communications loop, which are taken in and out of service while the system is active, you can prevent a missing, failed, or unresponsive unit from locking up the entire communications loop. The time out function allows the ADI/MDI to wait a specified amount of time for a given unit to respond. If the unit does not respond, you can program the ADI/MDI to indicate the lack of response, or to continue on with its regular function. Thus, you can use the time out function to detect which units are present and which units are not.

You can also use the time out function to automatically close a port for you. For the Serial Port, this feature is useful if the character string you are attempting to receive has no unique terminating character, such as a "Print Request" string from an RLC unit.

In this case, you can retrieve the entire Print Request transmission in one item if you configure the item to issue a Transmit Request, set the Terminator for a character not contained in the Print Request response, and set the time out value long enough so that the port does not close before the ADI/MDI has received the entire Print Request string.

You can use the time out function to receive variable length data over the Parallel Port. As stated previously, the unit accepts all bytes currently present in its Receive Buffer when the time out value expires. Therefore, with the proper TIME OUT and MAX DATA BYTES settings in an ED Item configuration, you can Strobe in a variable number of data bytes for that item, and allow the unit to close the port for you, thus accepting the bytes you have sent it.

Formatting Leading Zeroes

You can configure an ED Item for one of three leading zero format methods (BLANK LEADING ZEROES, SUPPRESS LEADING ZEROS, KEEP LEADING ZEROES). The method you select for an item applies to every numeric field in that item's data string.

A numeric field consists of a sequence of characters at the beginning of a data string, or preceded by a space in the string, composed of digits, and possibly a decimal point and/or a sign character (+/-). The field may contain other non-numeric characters appended to the right of the numeric portion of the field, such as a units label.

A leading zero is any zero appearing to the left of the first non-zero digit in a numeric field. Selecting BLANK LEADING ZEROES causes all leading zero characters to be replaced with BLANKS. The field size does not change. With SUPPRESS LEADING ZEROES selected, all leading zeroes are deleted. The size of the field is reduced by the number of zeroes deleted. Selecting KEEP LEADING ZEROES causes all zeroes in a field to be retained.

There are two exceptions to the above rules for leading zero blanking and leading zero suppression. For BLANK LEADING ZEROES, if an entire numeric field is comprised of zeroes, the right-most zero is kept, and the remaining leading zeroes are replaced with BLANKs. Also, if the numeric field contains a decimal point, and all the digits to the left of the decimal point are zeroes, the ADI/MDI keeps the zero immediately to the left of the decimal point and replaces the remaining leading zeroes with BLANKs.

For SUPPRESS LEADING ZEROES, if an entire numeric field is comprised of zeroes, the ADI/MDI deletes all the zeroes, but one. If the numeric field contains a decimal point, and all the digits to the left of the decimal point are zeroes, the unit again deletes all the leading zeroes, but one.

Examples:

Original	Blanked	Suppressed
"00000"	" 0"	"0"
"+00123"	"+ 123"	"+123"
"000.1"	" 0.1"	"0.1"
"-00012"	"- 12"	"-12"
"00250psi"	" 250psi"	"250psi"
"CNTA 001234"	"CNTA 1234"	"CNTA 1234"
"ABC00123"	"ABC00123"	"ABC00123"

Formatting Data

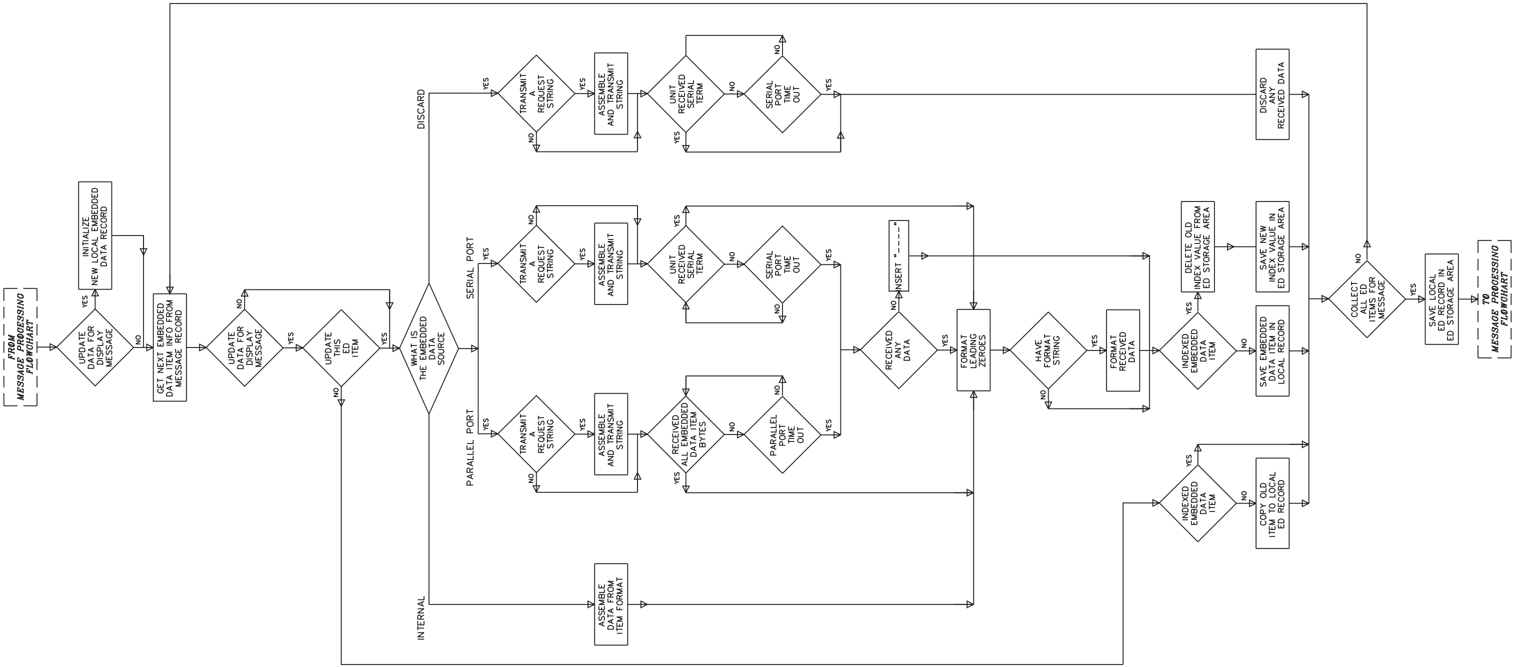
By selecting "Yes" for the FORMAT DATA field and entering the appropriate information in the FMT STRING field, you can configure an ED Item to keep or delete any characters in its received data string. Each position in the FMT STRING field corresponds to a character position in the ED Item's data string. Placing an 'X' at a certain location in the FMT STRING causes the ADI/MDI to delete the character at that position in the data string. Skipping a FMT STRING position, or entering a SPACE in the position, causes the unit to keep the character for that position.

If the data string is longer than the FMT STRING, the unit keeps all data string characters in excess of the FMT STRING positions. If the FMT STRING is longer than the data string, the excess FMT STRING positions are ignored. The unit performs this data formatting after it has performed any leading zero formatting you may have specified for the ED Item.

Examples:

```
DATA STRING:   " 3 CNTA 012345"
FMT STRING:    "XXX           "
RESULT:        "CNTA 012345"

DATA STRING:   "99 ALM1 0123.45"
FMT STRING:    "   XXXX     XXXXXXX"
RESULT:        "99  0123"

DATA STRING:   "15 PRE  001234"
FMT STRING:    "XXXXXXXX"
RESULT:        "001234"
```

EMBEDDED ITEM COLLECTION FLOWCHART


FROM MESSAGE PROCESSING FLOWCHART
UPDATE DATA FOR DISPLAY MESSAGE
YES
NO
INITIALIZE NEW LOCAL EMBEDDED DATA RECORD
GET NEXT EMBEDDED DATA ITEM INFO FROM MESSAGE RECORD
UPDATE DATA FOR DISPLAY MESSAGE
NO
YES
UPDATE THIS ED ITEM
YES
NO
WHAT IS THE EMBEDDED DATA SOURCE
DISCARD
SERIAL PORT
PARALLEL PORT
INTERNAL
TRANSMIT A REQUEST STRING
YES
NO
ASSEMBLE AND TRANSMIT STRING
UNIT RECEIVED SERIAL TERM
YES
NO
SERIAL PORT TIME OUT
YES
NO
TRANSMIT A REQUEST STRING
ASSEMBLE AND TRANSMIT STRING
UNIT RECEIVED SERIAL TERM
SERIAL PORT TIME OUT
TRANSMIT A REQUEST STRING
ASSEMBLE AND TRANSMIT STRING
RECEIVED ALL EMBEDDED DATA ITEM BYTES
PARALLEL PORT TIME OUT
ASSEMBLE DATA FROM ITEM FORMAT
DISCARD ANY RECEIVED DATA
INSERT "____"
RECEIVED ANY DATA
NO
YES
FORMAT LEADING ZEROES
HAVE FORMAT STRING
YES
NO
FORMAT RECEIVED DATA
INDEXED EMBEDDED DATA ITEM
YES
NO
DELETE OLD INDEX VALUE FROM ED STORAGE AREA
SAVE NEW INDEX VALUE IN ED STORAGE AREA
SAVE EMBEDDED DATA ITEM IN LOCAL RECORD
INDEXED EMBEDDED DATA ITEM
YES
NO
COPY OLD ITEM TO LOCAL ED RECORD
COLLECT ALL ED ITEMS FOR MESSAGE
NO
YES
SAVE LOCAL ED RECORD IN ED STORAGE AREA
TO MESSAGE PROCESSING FLOWCHART

Message Text

The Message Text area of the SFMD software functions as a limited text editor. The character position (Char Pos), Row (Row), and Column (Col) identifiers in the lower middle of the screen show the current position of the cursor in the text area. The following key-stroke functions are available in the Message Text area:

HOME: Moves the cursor to the beginning of the current row.

END: Moves the cursor to the end of the current row.

INSERT: Toggles between the INSERT and the OVERTYPE modes. The INSERT mode allows you to insert new characters anywhere in the existing Message Text. The OVERTYPE mode allows you to type new characters over the existing text.

DELETE: DELETEs the character at the current cursor position.

You can use the the ARROW keys to move around the Message Text area.

A Message's text can include any of the characters in the unit's character set, along with most of the lower ASCII control characters, such as the Carriage Return (CR) and Line Feed (LF) characters. The only lower ASCII characters you cannot include in the Message Text are 00h (0), 09h(9), and 14h(20), which are reserved for use by the ADI/MDI. You can enter any of the standard ASCII characters, 20h (32) to 7Eh (126), by hitting the appropriate key on the keyboard. All other characters must be entered through a combination of key strokes.

The characters from 01h (1) to 0Fh (15) are entered by holding down the CTRL key and pressing the 'A' (01h) through 'O' (0Fh) keys, respectively. The characters from 10h (16) through 1Fh (31) and the extended ASCII characters, from 7Fh (127) to B8h (184) are entered by holding down the ALT key and typing in the decimal number of the desired character on the numeric keypad.

You can have the Current Time and Date, Elapsed Timer values, and Local and Indexed ED item values inserted anywhere in the Message text through the use of other control code sequences. For Current and Elapsed Time values, you specify the format of the inserted values with a format code sequence, which specifies the time values you want to include.

For Message text that is transmitted over the Serial Port, such as for the MDS UNIT and TRANSMIT Message destinations, you can force the unit to run additional Transmit Delays at specific points in the Message transmission by inserting the TRX DELAY (t - <ALT-T>) control code anywhere in the Message text.

You enter a control code by pressing the appropriate letter key while holding down the ALT key. All control codes are displayed as two-letter sequences comprised of the diamond ( ) character followed by the lower-case letter of the key you pressed. A table of the various control codes, their usage, and function is included later in this section.

You can insert any number of blinking fields in the Message text by surrounding the desired text characters with the BLINK control code. For instance, to instruct the ADI/MDI to blink the word "Blink" in a Message, you must enter the BLINK control code, " b", by holding down the ALT key and pressing the 'B' key. The " b" sequence appears at the cursor location in the Message Text area. Type in "Blink" followed by another ALT-B control code. The sequence appears on the screen as " bBlink b". The unit blinks the characters sandwiched between the two BLINK control codes whenever it displays the Message.

Certain control codes require that you enter a two-digit number immediately following them. For instance, you would insert Local ED Item 1 in the Message text by including the sequence " d01" at the desired location in the text. The " d" sequence is the control code for a Local ED Item and "01" specifies Local item 1. When the ADI/MDI encounters this sequence, it replaces the " d01" character sequence with the current value of Local ED Item 1. You would insert Indexed Item 95 in the Message text similarly with the sequence " i95".

When inserting a Current Time or Elapsed Time value in the Message text, you must also specify the format of the time field. The format codes used to specify Current and Elapsed Time values are comprised of two-letter upper-case character sequences. Each format code sequence describes a particular time value, such as the Current YEAR, the Current MONTH, the Elapsed MINUTES, etc. A table of the various format codes, their usage, and function is included later in this section.

You string these format codes together to create a time field of your choosing and indicate the beginning and ending of the field with the appropriate control code. For instance, you would insert the Current Hours, Minutes, and Seconds in a Message by including the sequence " cHH:NN:SS c". The " c" sequence is the control code for a Current Time field and the "HH", "NN", and "SS" sequences are the HOURS, MINUTES, and SECONDS format codes, respectively.

Message Text (Cont'd)

When the ADI/MDI encounters this sequence, it replaces the entire string with the current values of the fields you specified.

Inserting an Elapsed Time value is similar to inserting a Current Time value, with the exception that you must also specify the Elapsed Timer number you want to reference.

For instance, to insert the Minutes, Seconds, and tenths of seconds for Elapsed Timer 15 in the Message text, you would include the sequence “ e15NN:SS.U e” in the text. The “ e” sequence is the control code for an Elapsed Time field and the “15” sequence specifies Elapsed Timer 15.

The “NN”, “SS”, and “U” sequences are the MINUTES, SECONDS, and TENTHS of seconds format codes, respectively.

The various Message Text control and format codes are listed in the following tables.

MESSAGE TEXT CONTROL CODES

FUNCTION	CONTROL CODE	KEY STROKE	USAGE	DESCRIPTION	
BLINK	b	ALT-B	btt..t b	tt..t -	Specifies the text characters you want to blink. You must enclose the blinking field with the “ b” control code, as shown.
CURRENT TIME	c	ALT-C	cff..f c	ff..f -	Specifies the Current Time field format string. You must enclose the Current Time field with the “ c” control code, as shown. The ADI/MDI replaces the format codes you include with the appropriate Real Time Clock (RTC) values. See the Current Time Format Codes table for details.
LOCAL ED ITEM	d	ALT-D	dnn	nn -	Specifies the number of the Local ED Item (01 to 15) that you wish to insert in the text. You must include any leading zeroes for this field. The ADI/MDI replaces this control sequence with the current value of the specified Local item.
ELAPSED TIME	e	ALT-E	ennff..f e	nn -	Specifies the number of the Elapsed Timer (00 to 15) that you wish to access. You must include any leading zeroes for this field.
				ff..f -	Specifies the Elapsed Time field format string, which must be enclosed with the “ e” control code, as shown. The ADI/MDI replaces the format codes you include with the appropriate values from the specified Elapsed Timer. See the Elapsed Time Format Codes table for details.
INDEXED ED ITEM	i	ALT-I	inn	nn -	Specifies the number of the Indexed ED Item (00 to 95) that you wish to insert in the text. You must include any leading zeroes for this field. The ADI/MDI replaces this control sequence with the current value of the specified Indexed item.
END-OF-LINE	l	ALT-L	l		Enter this control code at the end of the text you want the ADI/MDI to place on the top line of its display. Any subsequent text will appear on the bottom line of the display.
END-OF-SEGMENT	s	ALT-S	s		Enter this control code at the end of each segment (BLOCK) of text. You do not need to include this control code at the end of a line, or the end of the text, unless you want the ADI/MDI to display a blank BLOCK of text at either of those locations.
TRANSMIT DELAY	t	ALT-T	t		Enter this control code in the text to cause the ADI/MDI to run a Transmit Delay at that point in the text. This code is only useful for the MDS UNIT and TRANSMIT Message destinations.

CURRENT TIME CODE FORMATS	
CODE	**DESCRIPTION**
AA	Day Name, Abbreviated (i.e.. MON)
AF	Day Name, Full (i.e.. MONDAY)
DD	Date, Numeric (i.e.. 1-31)
HA	Hours, 12 Hour Clock (Civilian Time: 01-12)
HH	Hours, 24 Hour Clock (Military Time: 00-23)
HM	Same as above (HH)
MA	Month Name, Abbreviated (i.e.. JAN)
MF	Month Name, Full (i.e.. JANUARY)
MM	Month, Numeric (i.e.. 01-12, 1 = January)
NN	Minutes
P	AM/PM, can only be used with the “HA” code
SS	Seconds
U	Tenths of Seconds
UU	Tenths and Hundredths of Seconds
Y2	Year, Two Digits (i.e.. 93)
Y4	Year, Four Digits (i.e.. 1993)

ELAPSED TIME FORMAT CODES	
CODE	**DESCRIPTION**
HH	Elapsed LOW Hours, ones & tens
HI	Elapsed HIGH Hours, hundreds & thousands
HL	Elapsed LOW Hours, ones & tens
NN	Minutes
SS	Seconds
U	Tenths of Seconds
UU	Tenths and Hundredths of Seconds

Note: All Field Codes must be entered in CAPITAL LETTERS.

You can use any of the characters not included in either the Current Time or Elapsed Time Format Code tables as text in a Current Time or Elapsed Time Field. Consequently, any of the punctuation characters are available for use as field separators, such as the colon :, the hyphen - , the slash /, the ‘.’, etc.

Example: Given these Current Time and Elapsed Time values:

Current Time: Thursday, August 8, 1993 16:49:23.93
Elapsed Timer 00: 0001:23:45.67
Elapsed Timer 15: 9998:56:32.01

The ADI/MDI would replace the following Current and Elapsed Time format strings with the values shown:

FORMAT STRING	DISPLAYED OR TRANSMITTED VALUE
“ cDDMAY2 HA:NN:SS P c”	“ 8AUG93 4:49:23 PM”
“ cAF, MF DD, Y4 c”	“THURSDAY, AUGUST 8, 1993”
“ cHH:NN:SS.U c”	“16:49:23.9”
“ e00HIHL:NN:SS.UU e”	“0001:23:45:67”
“ e15HH:NN:SS.U e”	“98:56:32.0”
“ e00HL:NN e”	“01:23”

It is possible to program more characters for a line than the ADI/MDI can display. In addition, if a line includes any Current or Elapsed Time fields, or Local or Indexed Items, once the unit processes these fields, the resulting line may be longer than the unit can display. In cases such as this, the excess text for a line is ignored. You can avoid this problem by configuring the Message to CHARACTER or BLOCK scroll the line.

When programming Message Text that will be transmitted to a serial printer or ASCII terminal, keep in mind the format of the device with which you are attempting to interface. Some devices require that you terminate each line of text with a Carriage Return (CR) - Line Feed (LF) pair, some require only a CR, some require only a LF, others use another character or character sequence for line termination. Some devices automatically issue a CR and/or LF when they receive more text for a line then the device can handle, others do not.

Message Text (Cont'd)

Most serial printers contain special functions, such as buffer clearing, page ejection, and font control, which you can access through specific control code sequences. Consult your printer's manual for descriptions of these and other special printer functions.

You can insert the <ESC> character (1Bh-27) in the Message Text. If the Message Text is to be transmitted (MDS UNIT or TRANSMIT Destination), when the ADI/MDI encounters the <ESC> character in the Transmit Buffer, it clears the contents of the Receive Buffer at that point in the transmission. You can use this feature to accommodate difficult serial timing constraints. If you need to transmit the <ESC> character (many printers accept the <ESC> character to perform a special function or print a special character), insert the '^' character in the Message Text, where necessary. Whenever the unit encounters the '^' character in a transmission, it transmits the <ESC> character.

The ADI/MDI does not display the lower ASCII control characters. Therefore, if you need to designate a Message for DISPLAY & TRANSMIT, and must include any of the lower ASCII characters in the Message text for transmission to another serial device, the unit simply ignores the lower ASCII characters when it displays the Message.

The ADI/MDI replaces all END-OF-SEGMENT (s) and END-OF-LINE (l) control codes with CR-LF pairs when assembling the transmission for a DISPLAY & TRANSMIT Message. Also, any BLINK control codes you included in the Message are ignored when the unit assembles the transmission string. These characters are treated normally when the Message is displayed.

TEMPORARY MESSAGES

Besides displaying Messages programmed in its Message Memory, the ADI/MDI can display Message Text it receives over the Serial Port. The unit considers the received Message Text a Temporary Message. A Temporary Message has NO Priority, and the unit never places a Temporary Message on the Queue. The next Message request the unit processes that is destined for the display always replaces an existing Temporary Message on the display.

You can issue a Temporary Message with the "M:tt...t<term>" or the "C21M:t...t<term>" Command strings, where "tt...t" represents the Temporary Message Text and <term> represents the programmed Serial Terminator character. This is the same format in which RLC Message Display Slave (MDS) units in the RLC Mode expect to receive Message Text. Since the ADI/MDI and the MDS (in the RLC Mode) respond identically to these Message Text strings, you can "slave" one or more ADI/MDI's to a "master" unit by having the "master" ADI/MDI assemble and transmit MDS UNIT Messages to the "slave" units. The "slave" units treat the received MDS UNIT transmissions as Temporary Messages.

The ADI/MDI uses special control code sequences to specify the various functions that an MDS UNIT destination Message can access, such as the Message Time Out function, the Output Pin functions, scrolling, blinking, and Current and Elapsed Times. These control code sequences are inserted at the appropriate places in the Message Text transmission.

If your application involves transmitting Temporary Messages to an ADI/MDI from a computer, ASCII terminal, or a PLC, you can use the Message Simulator feature of the SFMD software to assemble the required Temporary Message strings for you. Simply program the Message in the Message Editing Screen for the MDS UNIT destination, with all the Message features you want to access, and then simulate the Message with the Message Simulator function.

The Simulator displays the assembled Temporary Message transmission. Ignore any "<XON>" and "<XOFF>" character strings that appear in the transmission string, and replace all "<CLR>" character strings with the 1Bh (27) character, and all "<TRX>" strings with the 14h (20) character. All control code sequences are two characters and begin with '▼' (1Fh - 31). You can use this character, or you can replace it with the '^' character (5Eh - 94).

The construction of a Temporary Message is discussed below.

The ADI/MDI begins processing a Temporary Message string upon receipt of the programmed Serial Terminator. The actual processing of a Message string entails the unit executing the various functions the Message string specifies. The unit can process only one Message string at a time. The maximum processing time of any Message string is 10 milliseconds and, while processing a Message, the unit cannot receive additional information over the Serial Port. The unit processes a Temporary Message string in the following sequence:

1. Configures the Output Pin, if specified.
2. Configures any specified Elapsed Timers.
3. Configures the Top and Bottom lines of the display for the designated scrolling technique, if any.
4. Places the Message on the display and starts the Message Time Out timer.

The various functions you can specify and include in a Message string are described below. You specify a special function for a Message by inserting the appropriate control code sequence in the Message string, as described below. All control code sequences have the form:

▼ x OR ^x

▼ or ^ - Message string control code, 1Fh (31) and 5E (94) are

interchangeable.

x - Control code character, 20h (32 - ' ') to 2Ch (44 - ',').

Note: For all examples, the '^' control code is shown, but the '▼' code can be used, as well.

Transmitting Extended Characters - 80h (128) to B8h (184)

The extended characters (characters from 80h (128) to B8h (184)) all require 8 bits for transmission. If your serial port is configured for 8 data bits, you can include any extended character in a Message string simply by holding down the <ALT> key and typing in the decimal number of the character on the numeric keypad. For applications using 7 data bits on the serial port, you cannot transmit a character greater than 7Fh (127), since the character requires 8 bits for transmission. Therefore, you must convert all extended characters into a 7-bit compatible form. This form is as follows:

^x
'^' - Message string control code, 5Eh (94).
x - Extended character - 50h (80).

Find the decimal equivalent of the extended character you wish to transmit. Subtract 80 decimal from it, and enter the resulting character following the '^' control code. When the ADI/MDI receives this sequence, it converts the control sequence back into the original extended character.

Example: To transmit the 'ü' character, you must find its decimal equivalent, 129, and subtract 80 from it, resulting in 49. Either type in this number while holding down the <ALT> key, or press the key for its ASCII equivalent, which is the character '1'. The resulting control code sequence then becomes:

^1

Displaying Text on the Bottom Line

You can instruct the unit to place text on the bottom line of the display by inserting the"^+" control code sequence at the end of the top line of text. All text following this control sequence is displayed on the bottom line. You can specify a separate scrolling technique for each line of text by including the appropriate control code sequence anywhere in the text for a given line. See Block and Character Scrolling below.

Example: To display "Top Line" on the top line of the display, and "Bottom Line" on the bottom line of the display, you would transmit the following text string:

M:Top Line^+Bottom Line<term>

Block and Character Scrolling

You can instruct the ADI/MDI to Block or Character scroll a line of text by including the appropriate control code sequence in the text for a line. To Block scroll a line, include the "^&" control code sequence anywhere in the line, and insert the "^)" End-of-Block sequence at the end of each Block of text. To Character scroll a line, include the "^'" control code sequence anywhere in the line. If you wish to separately Character scroll Blocks of text, insert the "**^)**" End-of-Block sequence at the end of each Block of text.

Example: To Block scroll the transmission string "This Message Block scrolls", you would transmit the following text string:

M:^&This^)Message^)Block^)scrolls<term>

Notice that there is no End-of-Block control code at the end of the Message string. The unit assumes an End-of-Block whenever it encounters the end of a line of text.

Example: To Character scroll the transmission string "This Message Character scrolls", you would transmit the following text string:

M:^'This Message Character scrolls<term>

Blinking Text

You can instruct the ADI/MDI to Blink text by surrounding the desired text with the Blink control code sequence, "**^<SPACE>**", where <**SPACE**> indicates pressing the space bar. The unit Blinks all text located between two Blink control sequences at the Blink rate specified for the line on which the text appears.

Example: To cause the Message "This Line Blinks" to Blink on the display, you would transmit the following text string:

M:^<SPACE>This Line Blinks^<SPACE><term>

Including the Serial Terminator Character as Text

You can include the Serial Terminator character in the Message text by inserting the "^(" control code sequence at the desired location in the Temporary Message string. When the ADI/MDI encounters this control sequence, it replaces the sequence with the programmed Serial Terminator character.

Example: To display the Message"*** CAUTION ***" on unit with the Serial Terminator set to '*', you would transmit the following text string:

M:^(^(^(CAUTION ^(^(^(<term>

Setting the Message Time Out Value

You can include a Message Time Out value in any Temporary Message string, which determines how long the unit displays the Message. The unit automatically cancels a Message when the Message's Time Out value expires. If you do not select the Time Out function for a Message, the Message is displayed indefinitely, until the Message is cancelled, a new Temporary Message string is received, or a DISPLAY destination Message is requested.

The Message Time Out value can range from 10-1260 milliseconds, 0-63 seconds, or 1-63 minutes. A Time Out value of 0 seconds indicates that the unit should process the Message and, after placing the Message on the display, immediately cancel the Message. This feature is useful for configuring the Output Pin, and/or Elapsed Timers, with a Temporary Message string. You include a Time Out value for a Temporary Message with the following format:

^$xxx

^ - Control code character, 5Eh (94).
$ - Message Time Out control code.
xxx - Time Out value, 3-digits with leading zeroes included. Construct this value as follows:

MILLISECONDS 0 = 128 + (1 to 126), for 10 to 1260 Milliseconds, in 10 millisecond increments.
MINUTES = 64 + (1 to 63) for Minutes.
SECONDS = 0 + (0 to 63) for Seconds.

A value of 255 indicates that the Message should not Time Out. This is the default value, and you do not need to include the Time Out control code sequence if the Message does not Time Out.

Example: To have the Temporary Message "This Message times out in 10 Minutes" appear on the display for ten minutes, you would transmit the following string:

M:^$074 This Message times out in ten Minutes<term> (74 = 64 + 10)

Configuring the Output Pin

You can configure the Output pin in a Temporary Message string. The unit expects to receive the OUTPUT pin configuration in the following format:

^#xxx

^ - Control code character, 5Eh (94).
# - Output Pin control code.
xxx - Configuration settings, 3-digits with leading zeroes included. Construct this value as follows:

Add 128 to UNLOCK the Output Pin, before changing settings.
Add 64 to turn the Pin ON.
Add 32 to turn the Pin OFF.
Add 16 to TOGGLE the Pin's state.
Add 8 to LOCK the Pin after changes have been made.
Add 4 to HALT the Time Out timer.
Add 2 to RUN the Time Out timer from the default value.
Add 1 to RUN the Time Out value from a value contained in the Message or Resume the Time Out from the existing value.

Example: To have a Message UNLOCK the Output Pin, turn the Pin ON, RUN the Time Out from the default value, and LOCK the Pin after all the changes have been made, you would include the following string in the transmission:

^#202 (202 = 128 + 64 + 8 + 2)

You can also set the OUTPUT Pin Time Out value in a Temporary Message string. The unit expects to receive the OUTPUT Pin Time Out configuration in the following format:

^%xxx

^ - Control code character, 5Eh (94).
% - Output Time Out control code
xxx - Time Out value, 3-digits with leading zeroes included. Construct this byte as follows, then convert to decimal for transmission.

MILLISECONDS = 128 + (1 to 126), for 10 to 1260 Milliseconds, in 10 millisecond increments.
MINUTES = 64 + (1 to 63) for Minutes.
SECONDS = 0 + (1 to 63) for Seconds.

Example: To have a Temporary Message set the Output Time Out value to 500 milliseconds, you would include the following string in the transmission:

^$178 (178 = 128 + 50)

Setting and Displaying the Current Time

You can set and display the Current Time in a Temporary Message string. The unit expects to receive the Current Time configuration in the following format:

^!uussnnhhaaddmmyyff...f^!

^^ -	Control code character, 5Eh (94).
! -	Current Time control code.
uussnnhhaaddmmyy -	16-digit value of the Current Time, with leading zeros, in the following format:
uu -	Hundredths of seconds (00 - 99)
ss -	Seconds (00 - 59).
nn -	Minutes (00 - 59).
hh -	Hours (00 - 23).
aa -	Day of the Week (01 - 07, Sunday is 01).
dd -	Day of the Month (01- 31).
mm -	Month (01 - 12, January is 01).
yy -	Year (00 - 99, 80 - 99 is 1980 - 1999, 00 - 79 is 2000 to 2079).
ff...f -	Format string for Current Time (optional). See the Time Codes section for details).
^^ -	Control code character, 5Eh (94).
! -	Current Time control code.

When the unit receives this string, it sets the Current Time to the included 16-digit value. If the string includes a Current Time format, the value of the Current Time is displayed at the location, and in the format you specify. You can also use Command C02 to set the Current Time.

Example: To set the Current Time in the ADI/MDI to October 26, 1993, 10:15:00.00 AM, and display the time in the format 26Oct93 10:15 AM, you would transmit the following string:

M:^!0000151003261093DDMAY2 HA:NN P^!<term>

See *MESSAGE TEXT*, page 32, for details on specifying the Current Time and Date.

Configuring and Displaying Elapsed Timers

You can configure and display one or more Elapsed Times in a Temporary Message string. The unit expects to receive an Elapsed Timer configuration in the following format:

^"xxxsdvuussnnhlhiff...f^"

^^ - Control code character, 5Eh (94).
" - Elapsed Time control code.
xxx - Three-digit Elapsed Timer number, leading zeros must be included.
s - Status flag, 'N' for ON, 'F' for OFF.
d - Direction flag, 'U' for UP, 'D' for DOWN.
v - Overflow flag, 'V' clears an Overflow, ' ' does not.
uussnnhlhi - 10-digit value of the Elapsed Timer, with leading zeros in the following format:
uu - Hundredths of seconds (00 - 99).
ss - Seconds (00 - 59).
nn - Minutes (00 - 59).
hl - Thousands and Hundreds Hours (00 - 99).
hi - Tens and Ones Hours (00 - 99).
ff...f - Format string for the Elapsed Time (optional). (See the Time Codes section for details).
^^ - Control code character, 5Eh (94).
" - Elapsed Time control code.

When the unit receives this string, it sets the specified Elapsed Timer to the included 10-digit value. If the string includes an Elapsed Time format, the value of the specified Elapsed Timer is displayed at the location, and in the format, specified in the string. You can also use Command C07, 'T' mnemonic, to configure the Elapsed Timers.

Example: To set Elapsed Timer 10 to RUN DOWN from 1 Hour, and display the Timer's value in the form xx:xx:xx, you would transmit the following string:

M:^"010ND 0000000100HL:NN:SS^"<term>

See *MESSAGE TEXT*, page 32, for details on specifying an Elapsed Time.

Temporary Message Control Code Sequences

The following is a list of the control code sequences the ADI/MDI recognizes.

'^ ' - Blink operator.
'^!' - Current Time operator, followed by the 16 digit value of the Current Time and an optional Current Time format string.
'^"' - Elapsed Time operator, followed by the three digit Timer number, a three character Timer configuration sequence, the ten digit value of the Elapsed Timer, and an optional Elapsed Time format string.
'^#' - Output Configuration operator, followed by a three-digit value.
'^$' - Message Time Out operator, followed by a three-digit value.
'^%' - Output Time Out operator, followed by a three-digit value.
'^&' - Block scrolling operator, included in a line, if that line BLOCK scrolls.
'^'' - Character scrolling operator, included in a line, if that line CHARACTER scrolls.
'^(' - Terminator operator, allows you to use the Serial Terminator character as a Message text character.
'^)' - END-OF-BLOCK operator.
'^+' - END-OF-LINE operator.
'^0' to '^h' - Extended characters, 80h (128) to B8h (184)

MESSAGE PROCESSING

The ADI/MDI begins processing a Message record in response to a request for that Message. The processing of a Message request entails the unit executing the various functions the Message record specifies. The unit can process only one new Message request at a time. While busy processing a request, the unit logs, but does not process, any additional requests that occur during the Message processing sequence. The unit responds to these requests after it has completed processing the current request.

As long as the ADI/MDI is not already busy processing a Message request, it processes all Message requests as it receives them, regardless of Priority. In other words, a higher Priority Message on the display does not prevent the unit from accepting and processing lower Priority Messages. A Message's Priority only comes into play after processing for a Message is complete and the unit must determine whether to place the newly requested Message on the display.

A Message record is processed in the following sequence:

1. For the COMMAND MESSAGE destination, the unit executes the specified Command string, which completes processing for the Message.
2. For all other Message Destinations the unit configures the Output Pin, if specified.
3. The unit configures any specified Elapsed Timers.
4. The unit collects any specified Embedded Data items, in numerical order, from low to high.
5. For the MDS UNIT or TRANSMIT destinations, the ADI/MDI assembles and initiates the Message text transmission.
6. For the DISPLAY destinations, a Queue Entry is created for the Message.
7. For the DISPLAY destinations, the unit determines whether to place the Message on the display, based on Priority.
8. If the Message is placed on the display, the unit starts the Message Time Out, and Embedded Data Update times.
9. If not destined for DISPLAY, the Message and all Local data items it collects are discarded. Any indexed items the Message collects are retained.
10. At this point, Message processing is complete and the unit can begin processing another new Message request.

You can only interrupt the Message processing sequence with a Message request over the Parallel Port. In this case, the unit deletes all data it has collected up to the point of the interruption, and abandons processing the remainder of the current request. It then begins processing the interrupting Parallel Port Message request.

Message Request Sources

The ADI/MDI accepts Message requests from the following sources:

1. The Serial Port.
2. The Parallel Port.
3. A Chained Message.
4. A Linked Message.
5. The Message Queue.
6. Message Updating Embedded Data.
7. The Default Message Function.
8. The Reset Message Function.
9. The Periodic Message Function.
10. The Trigger Message Function.
11. AMR Mode 1 or 2.
12. An Error Message.

Since the unit can process only one new Message request at a time, it logs any requests from other sources that occur while it is already busy processing a request. Once the unit has completed processing a Message, and has placed that Message on the Queue or discarded it, as necessary, it proceeds to scan for any additional Message requests it may have logged while processing the completed request.

Consequently, except for a Parallel Port interruption, as described above, the unit must always complete processing the current Message request before beginning to process another request. For instance, the ADI/MDI can log requests from other sources, such as the Periodic or Trigger functions, while collecting data for a Message. It cannot, however, begin processing a request from either of those sources until it has completed processing the current request.

Simultaneous requests from within the Periodic function are handled in a round-robin order. For instance, multiple Periodic Entries with identical Activation Times are processed in numerical order from low to high.

If Periodic Entries 1 and 15 specify the same Activation Time, the ADI/MDI requests Entry 1 first, and once it has completed processing that request, and has not logged requests from any of the other sources, it then processes the request from Periodic Entry 15. Simultaneous Trigger Message requests are handled similarly.

Whenever the display becomes available, whether through the expiration of the last Message in a Chain list, or a Chain cancellation, or the cancellation or time out of a non-Chained Message, the unit first scans for a pending Message request. If it finds another request, the unit begins processing that request. If no other request has been received, the ADI/MDI then checks the status of the Queue. If the Queue function is ON, and the Queue is not empty, the unit issues a request for the Message at the Top of the Queue (Queue Entry 0). Otherwise, if the Default function is ON, the Default Message is requested. If the Default function is OFF, the ADI/MDI blanks the display, and continues to scan for Message requests.

Message Priority

Every Message destined for the display (DISPLAY, DISPLAY & MDS UNIT, DISPLAY & TRANSMIT) must have a Priority assigned to it. The Message Priority can range from 1 (Highest) to 255 (Lowest). Temporary Messages have no Priority (they displace any Message on the display and any Message displaces them) and Error Messages have Priority 0 (only another Error Message or a Temporary Message can displace an Error Message on the display). The ADI/MDI determines whether to place a newly requested Message on the display based on the Message's Priority and that of the Message currently on the display. A Message of the same or higher (numerically lower) Priority will displace a Message of the same or lower (numerically higher) Priority on the display.

The unit also determines the Queue position of a requested Message based on the Message's Priority. All Messages with the same Priority are grouped together on the Queue, the most recently requested Message being the first of a like-Priority group. All Priority groups are arranged on the Queue in numerical order from high (numerically low) Priority to low (numerically high) Priority.

Message Time Out

You can assign a Message Time Out value to any Message designated for display (DISPLAY, DISPLAY & MDS UNIT, DISPLAY & TRANSMIT), which determines how long the unit displays the Message. The ADI/MDI runs this Time Out while the Message appears on the display and halts the Time Out and stores the remaining value on the Queue whenever the Message is bumped from the display. The Time Out restarts from this value when the Message is again displayed. If you do not select the Time Out function for a Message, that Message is displayed indefinitely, until it is bumped from the display, or cancelled.

The ADI/MDI automatically cancels a Message when the Message's Time Out value expires. If the message is contained in a Chain, the next Message in the Chain is automatically requested. If the Message is not included in a Chain, or is the last Message in a Chain, the ADI/MDI then looks to the Queue for the next Message to display.

You can also assign a Message Time Out to a Message designated for the MDS UNIT destination. The ADI/MDI transmits this Time Out value along with the Message text to the addressed MDS Unit. The MDS Unit then displays the Message for the Time Out value.

The Message Time Out value can range from 10-1260 milliseconds, 0-63 seconds, or 1-63 minutes. A Time Out value of 0 seconds indicates that the unit should process the Message, and, after placing the Message on the display, immediately cancel the Message. Since only a DISPLAY destination Message can specify a Message Chain list, this feature is useful for initiating a Message Chain that contains only COMMAND MESSAGE, MDS UNIT, and/or TRANSMIT destination Messages.

<Please refer to the Message Processing Flowcharts
at the end of this section.>

Message Queue

The ADI/MDI contains an extensive Message Queue function. The Message Queue serves as a prioritized waiting list for Messages destined for the display (DISPLAY, DISPLAY & MDS UNIT, DISPLAY & TRANSMIT), and can hold up to 32 Queue Entries (0-31). You can turn the Queue function ON and OFF as needed. When OFF, no new Queue Entries are created, except for that of a newly displayed Message. However, this new Queue Entry is removed from the Queue when the Message is removed from the display.

When ON, the ADI/MDI places all Messages destined for the display on the Queue, and then chooses which of these Messages to display, based on their respective Priorities, and order of request. Only Messages designated for one of the DISPLAY destinations can be placed on the Queue. Additionally, you can designate individual DISPLAY Messages for exclusion from the Queue. Messages not designated for the Queue only appear on the Queue while they are on the display, they are removed from the Queue once they leave the display.

Each Queue Entry contains the following information:

Entry Number - The Entry's position on the Queue (0-31, 0 is the TOP of the Queue).

Priority - The Priority of the Message creating the Queue Entry.

Message Number -The Message to be displayed.

Display flag -Indicates that the Message is on the display, or whether or not it has been displayed.

Chain flag -Indicates that the Message is part of a Message Chain.

Data flag -Indicates that the Message collected Local Embedded Data.

Data Size -The number of ED Storage Area bytes the Local Data occupies.

Time Out - The amount of Display Time remaining for the Message.

Queue Entry 0 normally appears on the display. You can override Queue Entry 0 with the C21 Command, which can request any Queue Entry for display. See *Command C21*, page 96, for details.

Every Message designated for one of the DISPLAY destinations must have a Priority assigned to it. You can set the Priority from 1 (Highest Priority) to 255 (Lowest Priority). The ADI/MDI stores all requested Messages on the Queue in Last-In/First-Out order within this Message Priority. The highest Priority, most recently requested message always resides at Queue Entry 0, which is considered the TOP of the Queue.

The unit assigns Messages to the display based on their Priority. A newly requested Message with the same or higher Priority as that on the display always replaces the Message on the display. The previously displayed Message retains its Queue Entry. When a Message is removed from the display (times out or is cancelled) with no other Message requests pending, such as a Chain, Trigger or Periodic request, the Message from Queue Entry 0 (the Top of the Queue) is automatically placed on the display. The remaining Queue Entries each move up one position in the Queue. Again, the newly displayed Message retains its Queue Entry. As the ADI/MDI creates and removes Queue Entries, Messages move up (closer to the top) or down (closer to the bottom) of the Queue.

The unit places Messages on the Queue under any of the following circumstances:

1. The currently displayed Message always appears on the Queue. The only exceptions are Temporary Messages and Error Messages, neither of which have Message records, and can never appear on the Queue.
2. A newly requested Message that is not placed directly on the display (has a lower Priority than the Message currently on the display) is placed on the Queue if the Message is designated for the Queue, and the Queue function is ON. Otherwise, the unit discards the Message.
3. If the unit attempts to place a new Message on the Queue while the Queue is FULL (all 32 Entries are in use), and the new Message has a same or higher Priority than the Bottom Queue Entry, the ADI/MDI removes the Bottom Entry and places the new Message on the Queue. Otherwise, the unit discards the new Message.

Message Queue (Cont'd)

The unit removes the Queue Entry for a Message under any of the following circumstances:

1. The Message on the display times out and is not part of a Chain, or is the last Message in a Chain.
2. The Message on the display is cancelled (C20 Command) and is not part of a Chain, or is the last Message in a Chain.
3. The Message, and any associated Chain, is cancelled with the "C20C" (Cancel Chain) Command string.
4. The Message is bumped from the display by another Message, and is not designated for the Queue, or the Queue function is OFF.
5. The Queue is full (all 32 Entries are in use) and the ADI/MDI attempts to place another Entry on the Queue (Queue Overflow). If the new Entry has the same or greater Priority than the Bottom Queue Entry, the unit removes the Bottom Entry and places the new Message on the Queue. If the new Message has a lower Priority than the Bottom Queue Entry, the unit discards the new Message.
6. The Queue Entry containing the Message is cancelled with the C20M, C20P, or C20Q Command, while the Message is not on the display. See *Command C20*, page 95, for details.

You can review the current contents of the Queue by issuing the C22 Command, 'Q' mnemonic. See *Command C22*, page 97, for details.

<Please refer to the Message Processing Flowcharts
at the end of this section.>

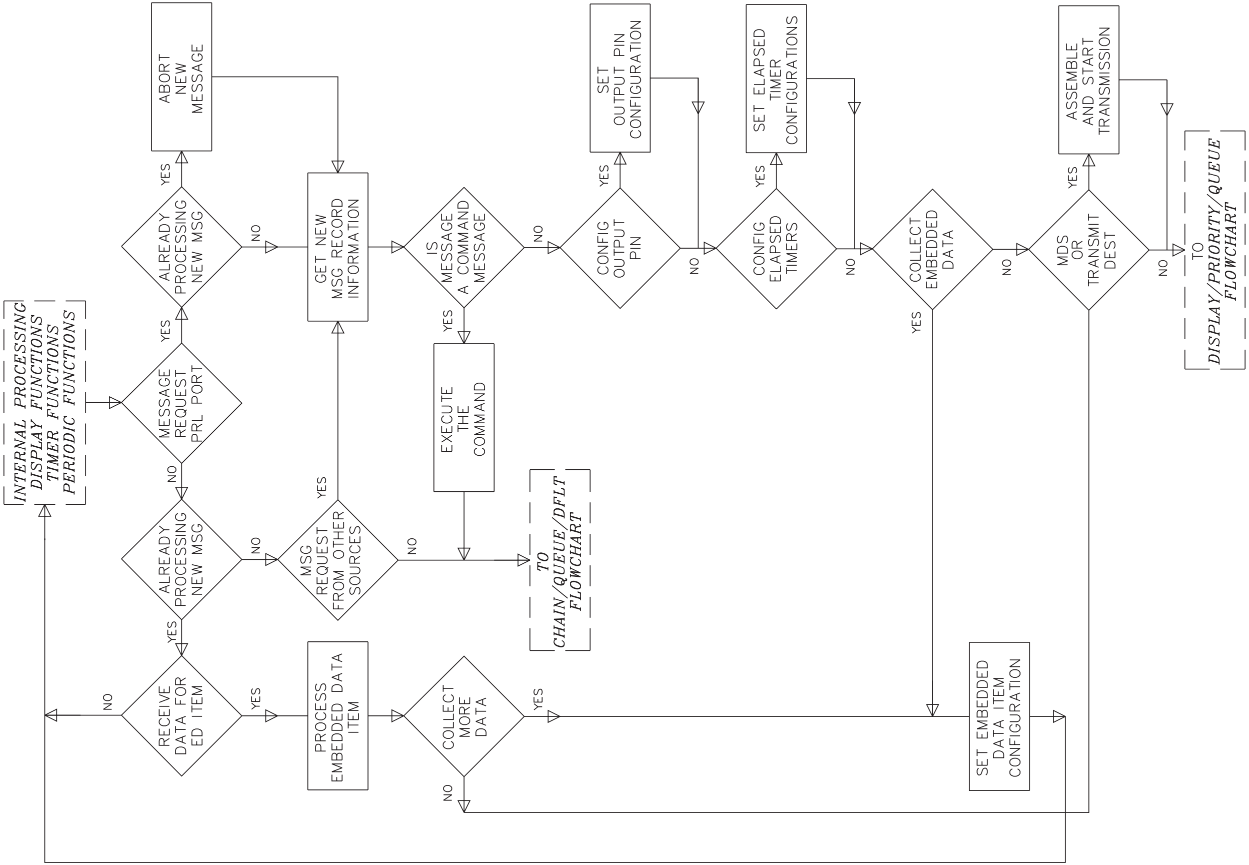
Message Records designated for the MDS UNIT, TRANSMIT, or COMMAND MESSAGE destinations are processed upon request, and then discarded. They have NO Priority and are never placed on the Queue.

You can turn the Queue Function on with the "C01N" Command String, and turn the Queue Function OFF with the "C01F" Command String. The C20 Command provides versatile Message and Queue Entry cancellation features and you can view the current status and contents of the Queue with the C22 Command ('Q' mnemonic). See the appropriate COMMANDS section for details.

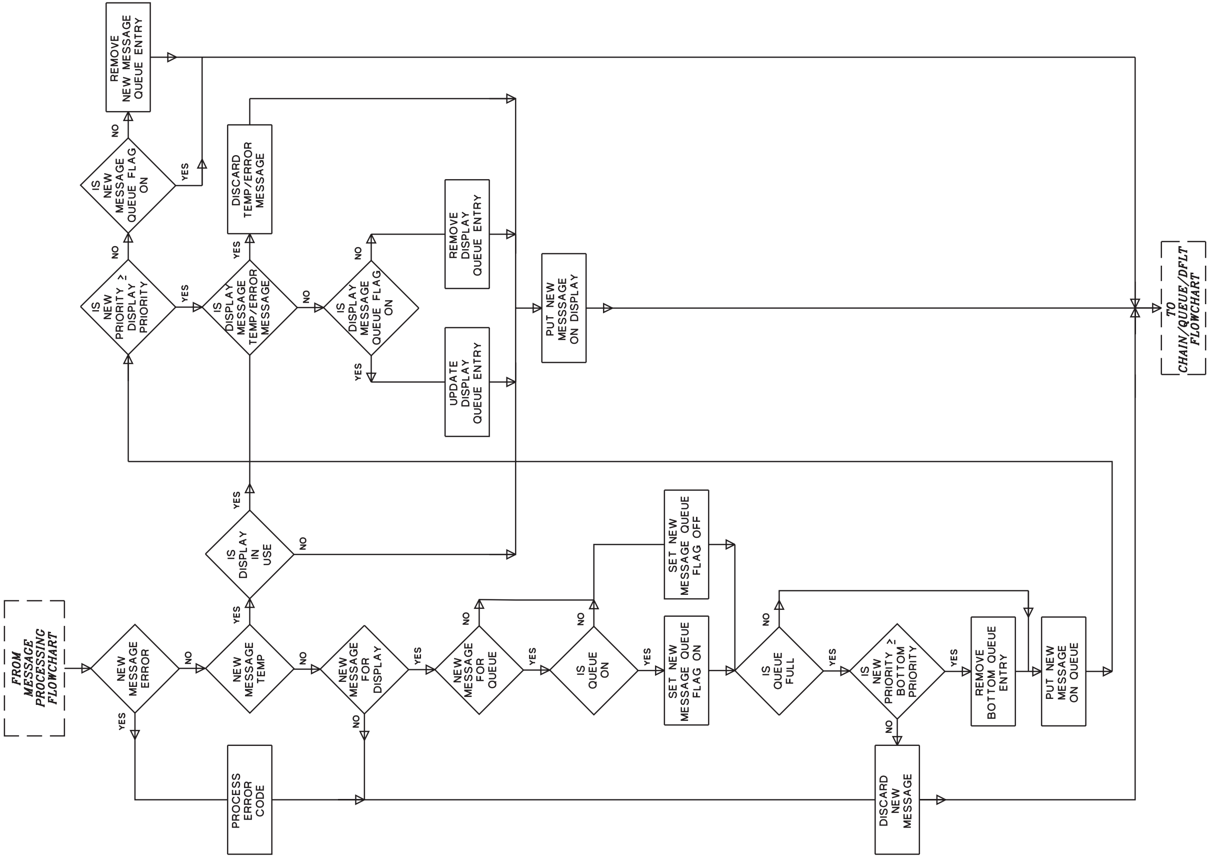
You can also configure the ADI/MDI to save the Queue's contents between unit power-down, power-up cycles by issuing the "C19HLT" Command string before removing power from the unit. This "HALT" Command causes the unit to come to an orderly shutdown, after preserving the current contents of the Message Queue and the Embedded Data Storage area. On the next unit power-up, the ADI/MDI can then recover and retain this information. You can program the Command as a Command Message, which can then be executed through any of the Message request sources before turning the unit OFF, or the system containing and/or powering the unit OFF. (See *Command C19*, page 95, for details).

The ADI/MDI saves the Queue contents and the Embedded Data Storage area automatically as long as the unit is not currently processing a Message request when a power interruption occurs. If the power to the unit is interrupted while the unit is processing a Message request, the ADI/MDI clears the Queue and the Embedded Data Storage area on the next unit power-up. (See *UNIT POWER-UP AND RESET*, page 74, for details).

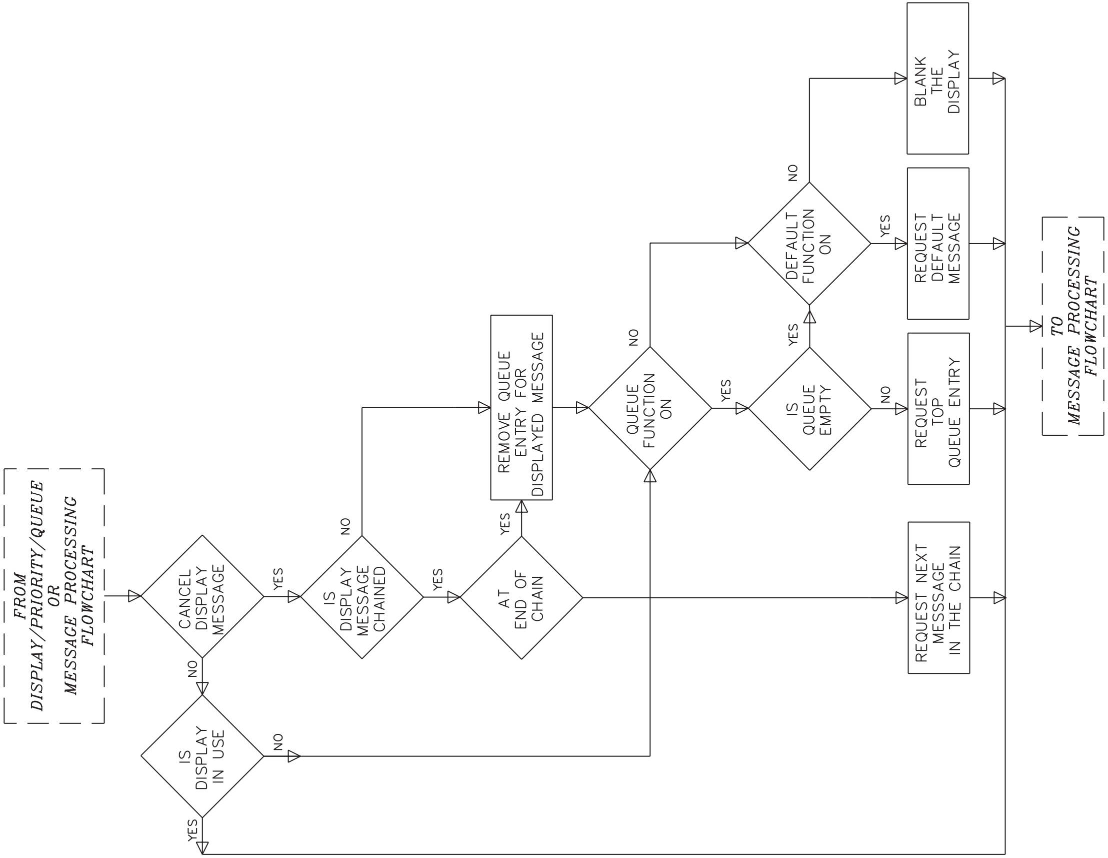
MESSAGE PROCESSING FLOWCHART


INTERNAL PROCESSING DISPLAY FUNCTIONS TIMER FUNCTIONS PERIODIC FUNCTIONS
MESSAGE REQUEST PRL PORT
YES
NO
ALREADY PROCESSING NEW MSG
YES
ABORT NEW MESSAGE
NO
GET NEW MSG RECORD INFORMATION
IS MESSAGE A COMMAND MESSAGE
YES
EXECUTE THE COMMAND
NO
CONFIG OUTPUT PIN
YES
SET OUTPUT PIN CONFIGURATION
NO
CONFIG ELAPSED TIMERS
YES
SET ELAPSED TIMER CONFIGURATIONS
NO
COLLECT EMBEDDED DATA
YES
NO
MDS OR TRANSMIT DEST
YES
ASSEMBLE AND START TRANSMISSION
NO
TO DISPLAY/PRIORITY/QUEUE FLOWCHART
ALREADY PROCESSING NEW MSG
NO
MSG REQUEST FROM OTHER SOURCES
YES
NO
TO CHAIN/QUEUE/DFLT FLOWCHART
YES
RECEIVE DATA FOR ED ITEM
NO
YES
PROCESS EMBEDDED DATA ITEM
COLLECT MORE DATA
YES
NO
SET EMBEDDED DATA ITEM CONFIGURATION

DISPLAY/PRIORITY/QUEUE FLOWCHART


FROM MESSAGE PROCESSING FLOWCHART
NEW MESSAGE ERROR
YES
NO
PROCESS ERROR CODE
NEW MESSAGE TEMP
YES
NO
IS DISPLAY IN USE
YES
NO
IS DISPLAY MESSAGE TEMP/ERROR MESSAGE
YES
NO
DISCARD TEMP/ERROR MESSAGE
IS DISPLAY MESSAGE QUEUE FLAG ON
NO
YES
REMOVE DISPLAY QUEUE ENTRY
UPDATE DISPLAY QUEUE ENTRY
PUT NEW MESSSAGE ON DISPLAY
IS NEW PRIORITY ≥ DISPLAY PRIORITY
YES
NO
IS NEW MESSAGE QUEUE FLAG ON
YES
NO
REMOVE NEW MESSAGE QUEUE ENTRY
NEW MESSAGE FOR DISPLAY
YES
NO
NEW MESSAGE FOR QUEUE
YES
NO
IS QUEUE ON
YES
NO
SET NEW MESSAGE QUEUE FLAG ON
SET NEW MESSAGE QUEUE FLAG OFF
IS QUEUE FULL
YES
NO
IS NEW PRIORITY ≥ BOTTOM PRIORITY
YES
NO
REMOVE BOTTOM QUEUE ENTRY
PUT NEW MESSAGE ON QUEUE
DISCARD NEW MESSAGE
TO CHAIN/QUEUE/DFLT FLOWCHART

CHAIN/QUEUE/DEFAULT FLOWCHART



ELAPSED TIMERS

The ADI/MDI contains sixteen separate, fully configurable Elapsed Timers. You can program any Message to access, configure, and/or display these Elapsed Timers. You can also access and configure any Elapsed Timer with the C07 Command ('T' mnemonic). The unit constantly updates all Elapsed Timer values according to the current configuration for each Timer. Each Elapsed Timer contains the following configurable attributes:

Timer Disposition: On every unit Power-Up and Reset, the unit configures the Elapsed Timer based on the Timer's Disposition setting, SAVE or CLEAR. With the Timer's Disposition set for SAVE, the unit maintains the Timer configuration on Power-Up or Reset (i.e.. if the Timer was running when the ADI/MDI powered-down, the Timer continues to run when power is restored to the unit).

With the Timer's Disposition set for CLEAR, the unit resets the Timer configuration on Power-Up or Reset; i.e. the unit halts the Timer, sets the Timer's direction to Down, clears the value and all trigger settings.

Timer Status: A Timer is either ON (running) or OFF (halted). Turning a Timer ON or OFF does not affect its value.

Timer Direction: Each Timer can run UP or DOWN. You can change the direction of the Timer while it is ON (running), there is no need to halt or reset the Timer.

Timer Value: A Timer can accumulate up to 10,000 hours, to 0.01 second resolution. You can set a Timer to any value within the range from 0000:00:00.00 to 9999:59:59.99. You do not need to halt the Timer to set its value.

Timer Overflow/Underflow: An UP Timer rolls-over (overflows) from 9999:59:59.99 to 0000:00:00.00. A DOWN Timer rolls-over (underflows) from 0000:00:00.00 to 9999:59:59.99. Each Timer contains an Overflow/Underflow flag that indicates this condition. You can reset this flag, as necessary.

Trigger Function: Each Timer includes a Trigger function, which you can turn ON or OFF. You can use a Timer's Trigger function to cause the ADI/MDI to request the specified Trigger Message whenever the Timer's value reaches its Trigger value.

Trigger Type: There are two different types of Triggers, ONE-SHOT and RETRIGGER. With the Trigger function set for ONE-SHOT, the unit requests its Trigger Message when the Timer reaches its Trigger value and the Trigger function is ON. The unit then turns the Trigger function OFF, but continues to run the Timer. The Timer value is unaffected. This feature is useful for generating automatic Message requests after a specified time delay.

With the Trigger function set for RETRIGGER, the Trigger Message is requested when a Timer reaches its Trigger value and its Trigger function is ON. The ADI/MDI then resets the Timer value to zero (0000:00:00.00), continues to run the Timer, and leaves the Trigger function ON. Thus, the Timer "retriggers" the Trigger Message each time its value reaches the Trigger value. This feature is useful for generating automatic Message requests on a repetitive basis.

Trigger Value: With the Trigger function ON, the unit automatically requests the Timer's Trigger Message when the Timer's value reaches the Trigger value. You can set a Timer's Trigger to any value in the range from 0000:00:00.00 to 9999:59:59.99. For the ADI/MDI to request a Trigger Message, the Trigger function must be ON when a Timer reaches its Trigger value. If you turn the Trigger function ON after the Timer has exceeded the Trigger value, the Trigger does not activate.

Trigger Message: With the Trigger function ON for a Timer, the Trigger Message is requested whenever a Timer's value reaches its Trigger value. You can assign any Message as the Trigger Message for a Timer and you can assign the same Message to more than one Elapsed Timer. You can also change the Trigger Message for a Timer while the Timer and its Trigger are both ON.

The ADI/MDI can process simultaneous Trigger Message requests. The Elapsed Timer Triggers are processed in a round-robin order, from Elapsed Timer 0 to Elapsed Timer 15. For instance, if Timer 0 and Timer 12 both "trigger" at the same time, the unit requests the Trigger Message for Timer 0 first, and once it's done processing that Message, and no other requests are pending, it requests The Trigger Message for Timer 12.

You can display an Elapsed Timer value in a Message by inserting the ELAPSED TIMER control code (e - <ALT-E>) followed by the two digit timer number, the appropriate Time format codes, and a closing control code (e) in the Message Text. If a displayed Timer is running, the running value is displayed as it changes. If the Message includes the TRANSMIT destination, the unit captures the instantaneous value of the Timer and inserts it in the transmission string according to the Elapsed Timer format you specified. (See *MESSAGE TEXT*, page 32, for details).

You can also include Timer values in Transmit Request strings and Internal Data Item format strings with the above procedure. In such cases, the unit again captures the instantaneous value of the Timer and inserts it in the appropriate string according to the Elapsed Timer format you specified in the string. (See *Formatting Transmit Request And Internal Items*, page 28, for details.)

When you specify an Elapsed Timer value in the text for a Message with an MDS UNIT destination, the ADI/MDI transmits both the current status and value of the Timer, along with the display format specified in the text, to the addressed MDS unit. When the MDS Unit processes the received Message text, it configures its Elapsed Timers to any Timer settings included in the transmission, thus “slaving” the MDS’s Timers to the transmitting unit’s Timer settings. (See *SERIAL COMMUNICATIONS*, MDS Unit Message Destination, page 71, for more details.)

HANDSHAKING

The ADI/MDI can be configured for one of two Handshaking protocols. The Handshaking protocol determines how a unit responds to various interactions with PLC's, computers, and other controlling devices. Through the chosen protocol, a controlling unit can monitor the status of one or more ADI/MDI's communicating with it.

The choice of Handshaking protocols is based on the type of interface established between the controlling unit and the ADI/MDI. The XON/XOFF protocol applies to serial interfaces, the READY/BUSY protocol applies to parallel port, or hardware interfaces.

Applications involving PLC's, computers, or extensive operator interaction with the ADI/MDI should specify one of the Handshaking protocols. Applications with little or no timing considerations may not need a Handshaking protocol. In addition, applications limited strictly to RLC products do not require a Handshaking protocol.

XON/XOFF

With the ADI/MDI configured to follow the XON/XOFF protocol, it transmits the ASCII XOFF character (13h - 19) whenever the unit enters a BUSY condition, and the XON character (11h - 17) whenever the unit enters the READY condition.

The unit responds to all Message requests by transmitting an XOFF, followed by a lower-case character code for the source of the Message request, and the three digit number of the Message being requested. When the unit has completed processing the current Message, it transmits the XON character, signifying that it is READY to process the next Message request or Command string.

While processing a Message request, intermediate XON and XOFF characters may be issued, depending on whether or not the Message collects certain types of Embedded Data. If a Serial or Parallel Port data item does not contain a Transmit Request string, the unit transmits the XON character when it is READY to receive the data item. For Serial data items, once the ADI/MDI receives the appropriate Terminator character, it issues an XOFF character, signifying that the unit is now BUSY processing the data item. For Parallel Port data items, once the unit has received the appropriate number of bytes, as specified by the data item length, it issues the XOFF character. If either port times out before receiving the specified data, the unit issues an XOFF when closing the port.

The ADI/MDI issues the following Message request response codes, based on the type of Message request it has received:

a - The Message request originated from AMR Mode 1 or AMR Mode 2.
c - The Message request originated in a CHAIN.
d - The DEFAULT Message function issued the request.
e - The ADI/MDI encountered an ERROR condition.
i - The Message request resulted from an INDEX comparison.
l - The previous Command Message had the Message LINKed to it.
m - The MESSAGE was requested over the Parallel or Serial Port.
p - The PERIODIC function issued the request.
q - The Message request originated from the QUEUE.
r - The RESET Message function issued the request
t - An Elapsed Timer's TRIGGER function issued the request.
u - The ADI/MDI is UPDATING the data for the Message on the display.

While configured for the XON/XOFF protocol, the unit always transmits error codes for any error conditions it encounters, regardless of the Error Handling configuration setting.

You can set and verify the Serial Port configuration settings with Command C05.

Examples: If you issue a request for Message 101 over the Parallel Port, the ADI/MDI responds with: "<XOFF>m101<XON>"

<XOFF> - ASCII XOFF character, the Unit is now BUSY
m101 - Response code and Message number
<XON> - ASCII XON character, the Unit is done processing Message 101 and is now READY

If Message 101 is a Command Message that LINKs to Message 102, the ADI/MDI responds with: "<XOFF>m101<XON><XOFF>l102<XON>"

<XOFF> - BUSY
m101 - Processing Message 101
<XON> - READY - Done processing Message 101
<XOFF> - BUSY
l102 - Processing LINKed Message 102
<XON> - READY - Done processing Message 102

If Message 102 collects two data items, the first containing the transmit string "✳ tN99TA✳", the second with no Transmit String, the ADI/MDI responds with:

"<XOFF>m101<XON><XOFF>l102✳<TRX>N99TA✳<XOFF><XON><XOFF><XON>"

<XOFF> - BUSY
m101 - Processing Message 101
<XON> - READY - Done processing Message 101
<XOFF> - BUSY
l102 - Processing LINKed Message 102
✳ - User supplied leading '✳' to CLEAR serial buffers
<TRX> - User supplied Transmit Delay (no characters are transmitted)
N99TA✳ - User supplied transmission request for address N99
<XOFF> - BUSY - Processing the first data item
<XON> - READY to receive the second data item
<XOFF> - BUSY - Processing the second data item
<XON> - READY - Done processing Message 102

Busy Mode

With the Output/Busy pin configured for the BUSY mode, the ADI/MDI follows the READY/BUSY protocol. Whenever the unit enters the BUSY condition, it sets the Output pin to the BUSY state. When the unit returns to the READY condition, it sets the Output pin to the READY state.

The READY and BUSY states are defined by the Output Pin Logic Level configuration setting. For positive logic, the pin conducts whenever the unit is READY. For negative logic, the pin conducts whenever the unit is BUSY.

The ADI/MDI responds to all Message requests by taking the Output pin to the BUSY state. Once the ADI/MDI has completed processing the requested Message, it returns the Output pin to the READY state, signifying that the unit is now READY to process the next Message request or Command string.

While processing a Message request, the ADI/MDI may also respond with intermediate READY and BUSY states, depending on whether or not the Message collects certain types of Embedded Data. If a Serial or Parallel Port data item does not contain a Transmit Request string, the unit takes the Output pin to the READY state when it is READY to receive the data item.

For Serial data items, once the ADI/MDI receives the appropriate Terminator character, it takes the pin to the BUSY state, signifying that the unit is now BUSY processing the data item. For Parallel Port data items, once the unit has received the appropriate number of bytes, as specified by the data item length, it takes the Output pin to the BUSY state.

	READY STATE	BUSY STATE
POSITIVE LOGIC	Conducts	Does Not Conduct
NEGATIVE LOGIC	Does Not Conduct	Conducts

You can set and verify the Output/Busy Pin configuration settings with Command C06.

INDEXED DATA ITEMS

The Indexed Embedded Data Item feature represents one of the most versatile, and powerful of the ADI/MDI's capabilities. The unit can store up to 96 Indexed Data Items, numbered 0 to 95. Each item can hold 126 bytes of variable data, plus a length byte and a string terminator byte, for a maximum size of 128 bytes.

You can program a Message to collect the data for an Indexed Item from any of the Embedded Data sources. Whenever a Message collects an Indexed Data Item, the ADI/MDI first deletes the existing data for the Index and then stores the new data at the assigned Index location. Commands and other Messages can then access the stored data by its Index Number.

You can insert Index values in Message text and Message transmissions, in Request Strings and Internal Item formats, and in Command and Command Message data strings. Indexed Items are accessed by their respective Index number. To insert an Index value in any of the preceding functions, you simply include the INDEX CONTROL CODE, " i" (ALT-I), followed by the two digit number of the desired Index. For instance, to insert the value of Index 01 in a Transmit Request, you would insert the character string " i01" at the desired location in the Request string. (See *MESSAGE TEXT*, page 32, and *FORMATTING TRANSMIT REQUEST AND INTERNAL ITEMS*, page 28, for details).

One Index can store a single data item. However, you can use the Internal Item Data source to combine multiple Index values into a single Index.

Example: Given the following Index values:

INDEX 01: "257.3 F"
INDEX 02: "246.6 F"
INDEX 03: "252.1 F"

You could combine these values into a single Index by collecting an Internal Data item for another Index, say Index 21, after collecting the data for Indexes 1 through 3. To accomplish this task, you could program an Internal Data Item with the following ITEM FORMAT:

ITEM FORMAT: "Unit 1:   i01 Unit 2:   i02 Unit 3:   i03"

After the ADI/MDI assembles this Indexed Item, Index 21 would contain the following information:

INDEX 21: "Unit 1: 257.3 F Unit 2: 246.6 F Unit 3: 252.1 F"

You can also use the Internal Item Data source to add a "time stamp" to an Indexed Item.

Example: The ADI/MDI collects a pressure reading every hour and stores it in Index 01. You want the unit to include the time of the reading with the data in Index 01. To accomplish this task, you could program an Internal ED Item with the following ITEM FORMAT:

ITEM FORMAT: " cHH:NN:SS   Unit 1:   i01 psig"

The unit assembles the new ED Item by inserting the current values of its Real Time Clock (RTC) into the Current Time field specified in the ITEM FORMAT, and by inserting the value of Index 01 at that specified location. The new ED Item is then saved in Index 01, overwriting the previous data. Given the following values for the Current Time and Index 01:

CURRENT TIME: Friday, August 13, 1993 09:57:05.23
INDEX 01: 137.4

The new value of Index 01 would look like this:

INDEX 01: "09:57:05 Unit 1: 137.4 psig"

Any Indexed Item containing a numeric field can be incremented and decremented. In other words, you can use an Index as an event counter, incrementing the Index each time a certain condition occurs. You can perform value comparisons on Indexes, resulting in conditional Message requests. Thus, you can implement an IF-THEN-ELSE Message request structure and assemble conditional, branching Message Chains, such as implementing an Index as a loop counter to control the number of times the ADI/MDI executes a sequence of Messages. (See *COMMAND C09*, page 89, for details).

You can use Indexed Items as variables in Transmit Requests, Internal Item formats, and Command Message data strings. If you have an application that requires a repetitive series of events, such as collecting hourly temperature readings from several units, and storing those readings in consecutive Index locations for later retrieval, you can program this sequence in a Chain of Messages and Command Messages. (See *Index Substitution* under *COMMAND MESSAGE*, page 99, for details).

You can also configure the ADI/MDI to save Indexed values between unit power-down/power-up cycles by issuing the “C19HLT” Command string before removing power from the unit. This “HALT” Command causes the unit to come to an orderly shutdown, after preserving the current contents of the Message Queue and the Embedded Data Storage area. On the next unit power-up, the ADI/MDI can then recover this information. You can program the Command as a Command Message, which can then be requested with any of the Message request sources before turning the unit OFF, or the system containing and/or powering the unit OFF. (See *COMMAND C19*, page 95, for details).

The ADI/MDI saves the Queue contents and the Embedded Data Storage area automatically as long as the unit is not currently processing a Message request when a power interruption occurs. If the power to the unit is interrupted while processing a Message request, the Queue and the Embedded Data Storage area is cleared on the next unit power-up. (See *UNIT POWER-UP AND RESET*, page 74, for details).

OUTPUT PIN

The ADI/MDI has one NPN open-collector Output pin which you can assign to one of two modes, the READY/BUSY handshaking mode, or the Message and Command controlled OUTPUT Mode. With the Output pin assigned to the READY/BUSY Mode, the unit indicates its current READY/BUSY condition by outputting the appropriate logic levels on the pin. (See *HANDSHAKING*, page 50, for details on the operation of the READY/BUSY mode.)

With the Output pin assigned to the OUTPUT Mode, you can program Messages and issue Commands to configure the pin. The Output pin in the OUTPUT Mode contains the following configurable attributes.

Logic Level: You can configure the Output pin for positive or negative logic. With the pin pulled-up through a resistor to a voltage source, the pin outputs a logic 1 (HIGH - does not conduct) for the positive logic ON state, and a logic 0 (LOW - conducts) for the positive logic OFF state. Conversely, the pin outputs a logic 0 (LOW - conducts) for the negative ON state, and a logic 1 (HIGH - does not conduct) for the negative logic OFF state.

Pin Status: You can turn the Output Pin ON or OFF. You can also TOGGLE its current status from ON to OFF and OFF to ON. The actual voltage level appearing at the Output pin is determined by the Logic Level setting, above.

Lock Status: You can LOCK the pin from accepting any further configuration changes. With the pin LOCKed, the unit ignores any attempt to configure the pin, whether by a Message, Command Message, or through a serial Command string. In order to configure a LOCKed Output pin, you must first UNLOCK it. The LOCK function allows you to implement a two-level priority system for the Output pin, as well as prevent unintentional reconfiguration of the pin.

Time Out Status: The Output pin contains a Time Out function, which you can RUN and HALT. The Time Out function is disabled in the HALTed state. Setting the Time Out to RUN enables the Time Out function, but does not automatically turn the Output pin ON, you must do this explicitly. At the expiration of the Output Time Out interval, the unit automatically turns the Output Pin OFF. You can use this function to issue fixed length pulses over the Output pin.

Time Out Value: You can specify the value the unit should use for the Output Time Out. The default Output Pin Time Out value is stored in the unit Configuration File. You can RUN the Time Out from this value, or you can specify a separate value. You can also RESUME the Time Out function from its HALTed value. The Output Pin Time Out value can range from 10 to 1260 milliseconds, 1 to 63 seconds, or 1 to 63 minutes.

OUTPUT PIN LOGIC LEVEL STATES

	POSITIVE LOGIC	**NEGATIVE LOGIC**
OUTPUT ON (Active)	Does Not Conduct	Conducts
OUTPUT OFF (Inactive)	Conducts	Does Not Conduct
READY (Active)	Conducts	Does Not Conduct
BUSY (Inactive)	Does Not Conduct	Conducts

PARALLEL PORT

The ADI/MDI's Parallel Port is an input-only port and can be configured for a wide variety of formats. The port consists of 2 CTRL lines (STROBE and MESSAGE/DATA) and 8 DATA lines. The DATA lines and the CTRL lines are separately DIP switch selectable for current Sinking or Sourcing, and Negative or Positive logic. The ADI is DIP switch selectable for 5 V or 12 V Logic Level compatibility, while the MDI is fixed at 12 V.

The unit accepts Message requests and can receive Embedded Data over the Parallel Port. Applying a valid STROBE pulse to the STROBE line (3 msec minimum) causes the unit to read the MESSAGE/DATA line and the DATA lines. The MESSAGE/DATA line value indicates how the unit should interpret the DATA lines, either as a Message number, or as Embedded Data.

The Parallel Port can also be configured to issue Automatic Message Requests (AMR Mode) based on changes on the port DATA lines without STROBE pulses.

The format of the port is specified in the Configuration file and can be selected for 4, 8, or 9 bit BCD, 4 or 8 bit Binary, or one of the two AMR Modes. The port format determines the number of STROBE pulses that must be applied in order to request a Message and to supply data. Command C04 allows you to interrogate and change the Parallel Port configuration settings.

For port configurations requiring STROBE pulses, the Parallel Port functions in the following manner. Upon detecting a valid STROBE pulse (3 msec minimum length), the ADI/MDI begins sampling the Parallel Port DATA lines and the MESSAGE/DATA Control line. These lines must remain stable for the duration of one Sample time in order for the unit to accept their values. Once the programmed Sample Time (1 to 255 msec) has been attained, the unit processes the DATA lines according to the MESSAGE/DATA line value. The ADI/MDI restarts the Sample Time if any of the lines change level (noise) before the Sample time expires.

(See *Parallel Port Message Requests* later in this section for detailed descriptions of these Message request processes.)

A programmable Debounce Time (10 to 2550 msec) commences after the STROBE line returns to the inactive state. This Debounce Time effectively removes unwanted contact bounce and noise spikes on the STROBE line, thus eliminating spurious STROBE pulses. Any additional activity on the STROBE line during the Debounce time causes the ADI/MDI to restart the Debounce Timer. During the Debounce Time, the unit ignores activity on the strobe line, except to restart the Debounce Timer. Once the Debounce Time expires, the unit is again ready to accept information over the Parallel Port in the same sequence.

For port configurations involving one of the AMR Modes, the Parallel Port functions in the following manner. With the MESSAGE/DATA line at the MESSAGE level, the unit constantly samples the DATA lines, looking for any changes from its most recent sample. If the ADI/MDI detects a change, it begins the Sample timer. Upon finding a new stable value, one that lasts at least the duration of the Sample time, the unit automatically issues a request for the appropriate Message based on the selected AMR Mode and the new port DATA line values.

The AMR function is temporarily disabled while the MESSAGE/DATA line is at the DATA level. The ADI/MDI resumes AMR detection once the Message/data line returns to the Message level. See *AMR Mode 1* and *AMR Mode 2* later in this section for full descriptions of the AMR functions.

Parallel Port Logic Levels

The following table shows the relative input voltage levels required for the Parallel Port MESSAGE/DATA and STROBE control lines and the D0 to D7 DATA lines, based on the LOGIC polarity DIP switch settings.

		CTRL LOGIC	
		–	
MESSAGE/DATA LINE	MESSAGE	HI	LO
	DATA	LO	HI
STROBE LINE	ACTIVE	LO	HI
	INACTIVE	HI	LO

		DATA LOGIC	
		–	
D0-D7 DATA LINES	1 (Active)	LO	HI
	0 (Inactive)	HI	LO

MESSAGE/DATA Line

The ADI/MDI only accepts Message requests over the Parallel Port while the MESSAGE/DATA line is set for the MESSAGE level, with the exception of the 9-bit BCD Mode. In the 9-bit BCD Mode, the MESSAGE/DATA line represents the hundreds digit for Message requests. The unit interprets the MESSAGE level as a 0, and the DATA level as a 100.

The ADI/MDI can only receive Embedded Data over the Parallel Port while the MESSAGE/DATA line is at the DATA level. If the unit is not currently processing a Message collecting an Embedded Data item over the Parallel Port, all port activity is ignored while the MESSAGE/DATA line is at the DATA level.

If the ADI/MDI is currently processing a Message collecting data from the port, and the port is STROBEd with the MESSAGE/DATA line at the MESSAGE level, the unit aborts processing of the existing Message, and begins processing the Parallel Port for a new Message request.

The MESSAGE/DATA line serves as an Enable/Disable line for the AMR Modes. Taking the line to the DATA level disables automatic Message requests. AMR detection is resumed when the MESSAGE/DATA line returns to the MESSAGE level.

The MESSAGE level of the MESSAGE/DATA line is determined by the CTRL LOGIC DIP switch setting.See the preceding table under Parallel Port Logic Levels for the appropriate switch setting and logic levels.

STROBE Line

The ADI/MDI begins sampling the Parallel Port DO-D7 lines and the MESSAGE/DATA line after detecting a valid length STROBE pulse on the STROBE line. A valid STROBE pulse consists of any transition to the active state of the STROBE line for 3 msec or longer while the Debounce timer is not running.

The unit accepts the port line values at the end of the programmed Sample time, whether or not the STROBE line has returned to the inactive state. The Sample time tells the unit how long the port must be stable before accepting its value as valid, and is programmable from 1 to 255 milliseconds. The ADI/MDI automatically adjusts the Debounce time to always be longer than the Sample time.

The Debounce timer starts when the STROBE line returns to its inactive state and determines how long the unit waits before again accepting STROBE pulses. The Debounce time eliminates spurious STROBE pulses due to contact bounce and noise spikes, and is programmable from 10 to 2550 milliseconds. The unit automatically adjusts the Sample time to always be shorter than the Debounce time.

The number of STROBEs required to request a Message or collect a byte of data over the Parallel Port is dependent on the selected port width. See the appropriate section below for a full description.

The active state of the STROBE line is determined by the CTRL LOGIC DIP switch setting. See the preceding table under Parallel Port Logic Levels for the appropriate switch setting and logic levels.

Parallel Port Message Requests

The available configurations for requesting Messages over the Parallel Port are:

BCD 4-Bits
BCD 8-Bits
BCD 9-Bits
Binary 4-Bits
Binary 8-Bits
AMR Mode 1
AMR Mode 2

Except for the BCD 9-bit mode, the MESSAGE/DATA line must be at the MESSAGE level to issue a Message request over the Parallel Port. The number of STROBEs required to request a Message is determined by the chosen width of the port.

The logic level of the parallel port DATA lines is determined by the DATA LOGIC DIP switch setting. See the preceding table under Parallel Port Logic Levels for the appropriate switch setting and logic levels.

4 or 8-Bit Binary

The ADI/MDI can receive requests for any programmed Message (0 to 255) with the Parallel Port configured for one of the Binary modes.

In the 4-bit mode, the unit only reads DATA lines D0 to D3 and requires 2 STROBE pulses per Message request. The first STROBE presents the HIGH nibble (high-order 4 bits) of the Message number to the unit, the second STROBE presents the LOW nibble (low-order 4 bits).

In the 8-Bit Binary mode, the unit reads all 8 Parallel Port DATA lines, D0 to D7. A single STROBE pulse is required to request Messages in this mode.

Upon receiving the proper number of STROBEs, the unit converts the Binary Message number to its decimal equivalent according to the following tables, and issues a request for that Message.

8-BIT BINARY PORT		DECIMAL
D7 D6 D5 D4	D3 D2 D1 D0	VALUE
0 0 0 0	0 0 0 0	0
0 0 0 0	0 0 0 1	1
0 0 0 0	0 0 1 0	2
0 0 0 0	0 1 0 0	4
0 0 0 0	1 0 0 0	8
0 0 0 1	0 0 0 0	16
0 0 1 0	0 0 0 0	32
0 1 0 0	0 0 0 0	64
1 0 0 0	0 0 0 0	128

4-BIT BINARY PORT			
HIGH NIBBLE D3 D2 D1 D0	DECIMAL VALUE	LOW NIBBLE D3 D2 D1 D0	DECIMAL VALUE
0 0 0 0	0	0 0 0 0	0
0 0 0 1	16	0 0 0 1	1
0 0 1 0	32	0 0 1 0	2
0 1 0 0	64	0 1 0 0	4
1 0 0 0	128	1 0 0 0	8

Note: More than one bit may be active. The message number requested is the sum of the decimal values for all ‘1’ bits.

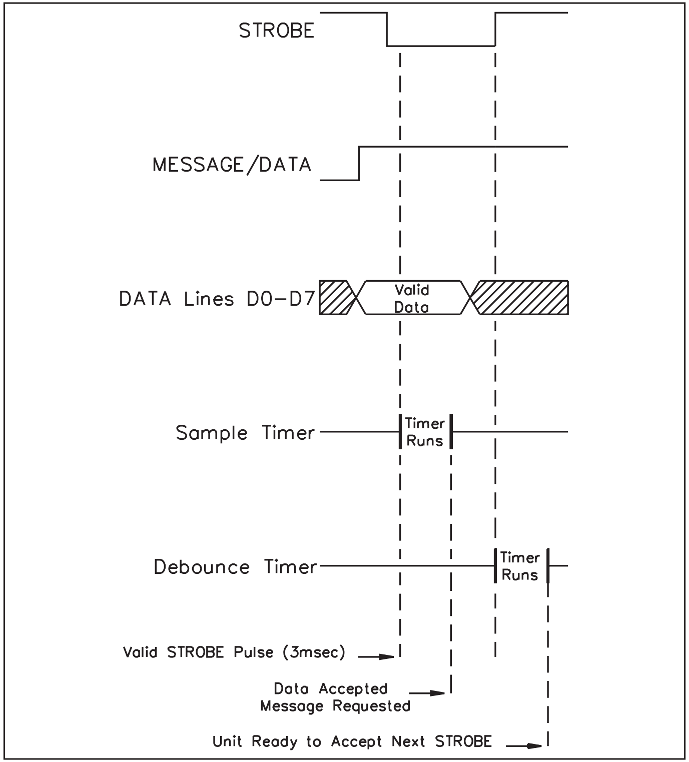


Figure 16, 8-BIT Binary Message Request Timing

Note: Shown for CTRL LOGIC Switch in NEGATIVE position (DOWN).

4 or 8-Bit Binary (Cont'd)

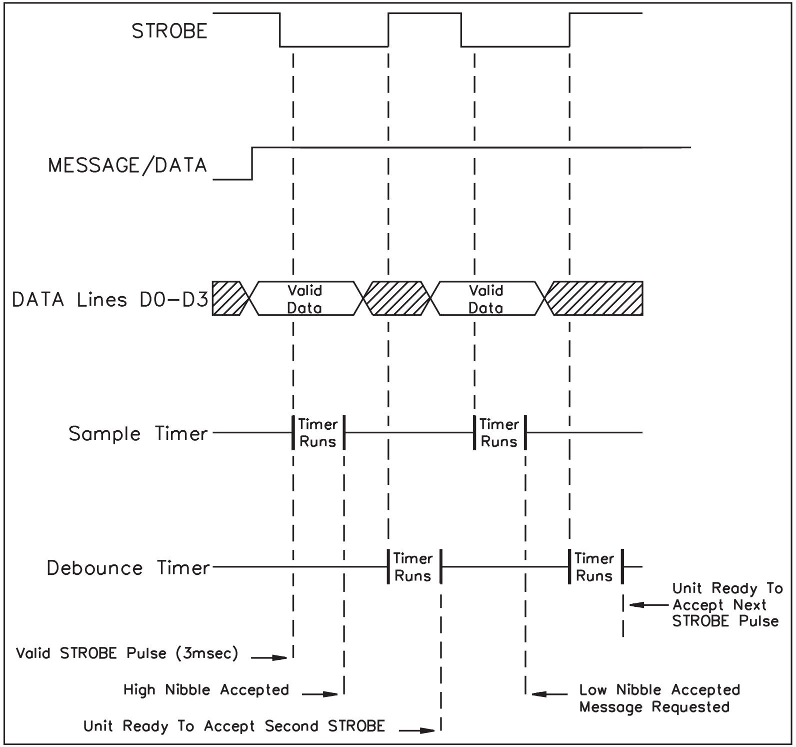


Figure 17, 4-BIT Binary Message Request Timing

Note: Shown for CTRL LOGIC Switch in NEGATIVE position (DOWN).

4 or 8-Bit BCD

The ADI/MDI can receive requests for any programmed Message (0 to 255) with the Parallel Port configured for either of these BCD modes.

In the 8-bit BCD Mode, the unit requires 2 valid STROBE pulses per Message request. The first STROBE presents the hundreds digit on DATA lines D0-D3. DATA lines D4-D7 are ignored. The second STROBE presents the tens digit on lines D4-D7, and the ones digit on lines D0-D3.

In the 4-bit BCD mode, the unit only reads DATA lines D0-D3, and requires 3 valid STROBE pulses per Message request. The first STROBE presents the hundreds digit, the second STROBE presents the tens digit, and the third STROBE presents the ones digit.

Upon receiving the appropriate number of STROBEs, the unit converts the 3-digit BCD number to its decimal equivalent according to the following tables, and issues a request for that Message.

BCD TABLE

DATA LINES D7 D6 D5 D4	BCD VALUE	DATA LINES D3 D2 D1 D0	BCD VALUE
0 0 0 0	0	0 0 0 0	0
0 0 0 1	10	0 0 0 1	1
0 0 1 0	20	0 0 1 0	2
0 0 1 1	30	0 0 1 1	3
0 1 0 0	40	0 1 0 0	4
0 1 0 1	50	0 1 0 1	5
0 1 1 0	60	0 0 1 1	6
0 1 1 1	70	0 1 1 1	7
1 0 0 0	80	1 0 0 0	8
1 0 0 1	90	1 0 0 1	9
1 0 1 0	Invalid	1 0 1 0	Invalid
1 0 1 1	↑	1 0 1 1	↑
1 1 0 0		1 1 0 0	
1 1 0 1		1 1 0 1	
1 1 1 0	↓	1 1 1 0	↓
1 1 1 1	Invalid	1 1 1 1	Invalid

Note: More than one bit may be active. The message number requested is the sum of the decimal values for all '1' bits.

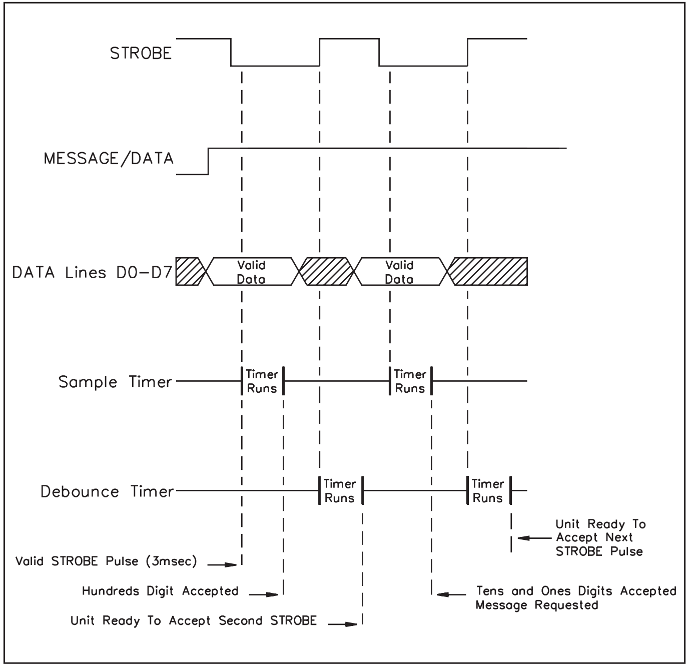

STROBE
MESSAGE/DATA
DATA Lines D0–D7
Valid Data
Valid Data
Sample Timer
Timer Runs
Timer Runs
Debounce Timer
Timer Runs
Timer Runs
Unit Ready To Accept Next STROBE Pulse
Valid STROBE Pulse (3msec)
Hundreds Digit Accepted
Tens and Ones Digits Accepted Message Requested
Unit Ready To Accept Second STROBE

Figure 18, 8-BIT BCD Message Request Timing

Note: Shown for CTRL LOGIC switch in the NEGATIVE position (DOWN).

4 or 8-Bit BCD (Cont'd)

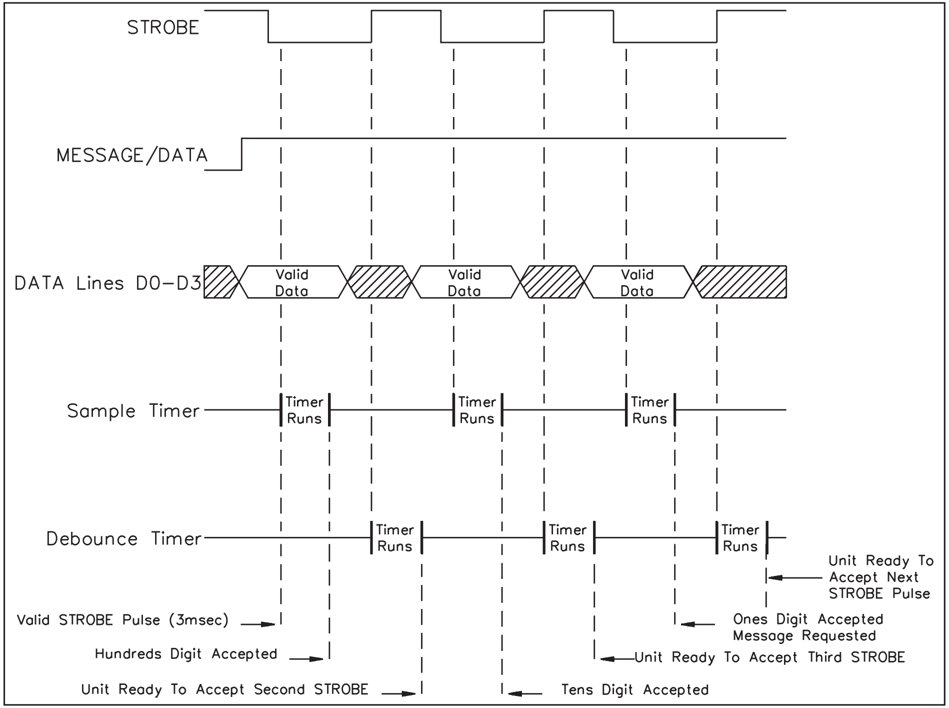


Figure 19, 4-BIT BCD Message Request Timing

Note: Shown for CTRL LOGIC switch in the NEGATIVE position (DOWN).

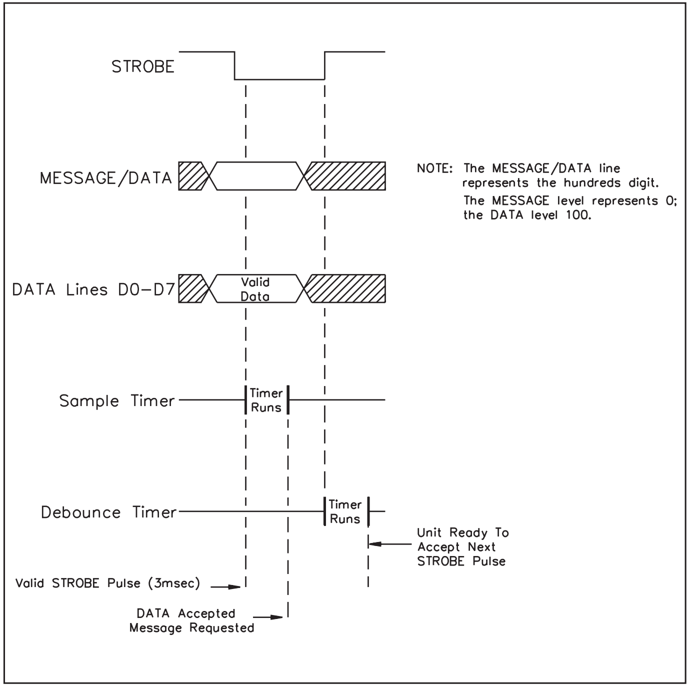
9-Bit BCD

The ADI/MDI can only receive requests for Messages 0 to 199 with the Parallel Port configured for the 9-bit BCD mode. Messages 200 to 255 are NOT accessible from the Parallel Port in this mode. A single STROBE pulse is required to request a Message in the 9-Bit BCD mode.

The MESSAGE/DATA line presents the hundreds digit (MESSAGE level is 0, DATA level is 100), DATA lines D4-D7 present the tens digit, and DATA lines D0-D3 present the ones digit of the desired Message number to the ADI.

Upon receiving the STROBE pulse, the unit converts the 9-bit BCD Message number to its decimal equivalent and issues a request for that Message.

While the unit is collecting data for a Parallel Port Embedded Data item, the MESSAGE/DATA line must be at the DATA level for the unit to accept data over the port.


STROBE
MESSAGE/DATA
DATA Lines D0–D7
Valid Data
Sample Timer
Timer Runs
Debounce Timer
Timer Runs
Valid STROBE Pulse (3msec)
DATA Accepted Message Requested
Unit Ready To Accept Next STROBE Pulse

NOTE: The MESSAGE/DATA line represents the hundreds digit. The MESSAGE level represents 0; the DATA level 100.

Figure 20, 9-BIT BCD Message Request Timing

Note: Shown for CTRL LOGIC Switch in the NEGATIVE position (DOWN).

AMR Mode 1

With the Parallel Port configured for AMR Mode 1, the ADI/MDI monitors each DATA line (D0 to D7) separately for logic level changes. The 8 DATA lines each have two message numbers assigned to them, one for the Active logic level, one for the Inactive logic level. The DATA LOGIC DIP switch setting determines the Active (1) and Inactive (0) levels for the DATA lines. (See the Parallel Port LOGIC LEVEL table earlier in this section for the appropriate Active and Inactive levels for the selected LOGIC polarity.)

The unit automatically requests a DATA line's Active Message whenever that DATA line undergoes an Inactive-to-Active (0 to 1) transition. Conversely, a DATA line's Inactive Message is requested whenever that DATA line undergoes an Active-to-Inactive (1 to 0) transition.

The Sample time determines the sensitivity of the unit to logic level changes. A short Sample time (1 millisecond) allows the unit to respond quickly to a changing line level. A long Sample time (255 milliseconds) causes the unit to wait for a line to stabilize before requesting the appropriate Message. Consequently, the Sample time can be used to suppress Message requests due to push-button and contact bounce, and relay chatter.

The unit can detect and process simultaneous changes of all 8 DATA lines. The Debounce time setting determines the minimum time between simultaneous and successive Message requests generated by AMR Mode 1, and can be adjusted from 10 milliseconds to 2.5 seconds. A round-robin scanning scheme from D7 to D0 is utilized to detect simultaneous line changes. A single line undergoing rapid changes, such as relay chatter, does not monopolize the unit. Having processed the Message request generated by a changing line level, the ADI/MDI continues on to scan the remaining lines before returning to the line that underwent the most recent change.

An AMR Mode 1 generated request for a non-programmed Message causes the unit to request the Default Message. If the Default function is turned OFF, the unit ignores the request and leaves the current Message on the display. Therefore, the Default Message function can be used as a "don't care" condition in AMR Mode 1.

The following table lists the Messages assigned to each of the DATA lines for both the Active and Inactive levels.

DATA Line	Active Message	Inactive Message
D0	Message 1	Message 11
D1	Message 2	Message 12
D2	Message 3	Message 13
D3	Message 4	Message 14
D4	Message 5	Message 15
D5	Message 6	Message 16
D6	Message 7	Message 17
D7	Message 8	Message 18

Note: On each power cycle, the ADI/MDI reads the parallel port and requests a message for each active line.

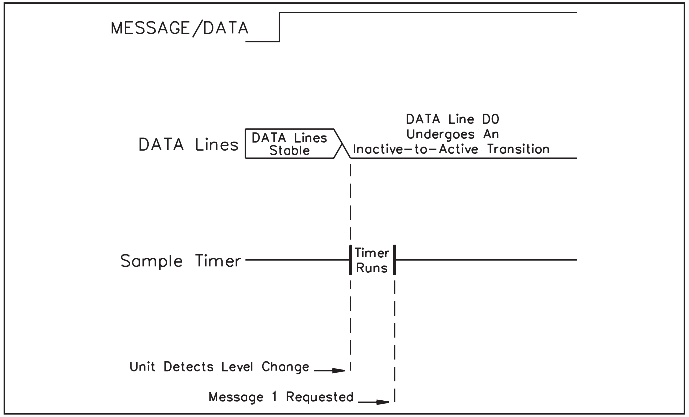

MESSAGE/DATA
DATA Lines
DATA Lines Stable
DATA Line D0 Undergoes An Inactive–to–Active Transition
Sample Timer
Timer Runs
Unit Detects Level Change
Message 1 Requested

Figure 21, AMR Mode 1 Timing

Note: Shown for CTRL and DATA LOGIC Switches in the NEGATIVE position (DOWN).

AMR Mode 2

With the Parallel Port configured for AMR Mode 2, the ADI/MDI constantly monitors the 8-bit binary value of the DATA lines (D0 to D7). Upon detecting a change in this value, the unit automatically issues a request for the Message corresponding to that value.

The DATA LOGIC DIP switch setting determines the active levels for the DATA lines. (See the Parallel Port Logic Level table shown earlier in this section for the appropriate Active (1) and Inactive (0) levels for the selected LOGIC polarity.)

The Sample time determines the sensitivity of the unit to logic level changes. A short Sample time (1 millisecond) allows the unit to respond quickly to a changing port value. A long Sample time (255 milliseconds) causes the unit to wait for the port value to stabilize before requesting the appropriate Message. Consequently, the Sample time can be used to suppress Message requests due to contact or push-button bounce.

The Debounce time setting determines the minimum time between successive Message requests generated by AMR Mode 2, and can be adjusted from 10 milliseconds to 2.5 seconds. An AMR Mode 2 generated request for a non-programmed Message causes the unit to request the Default Message. If the Default function is turned OFF, the unit ignores the request and leaves the current Message on the display. Therefore, the Default Message function can be used as a “don’t care” condition in AMR Mode 2.

Note: On each power cycle, the ADI/MDI reads the Parallel Port and requests a message if the current binary value is not zero. If the current binary value is zero, the unit does not issue a request until the port value changes.

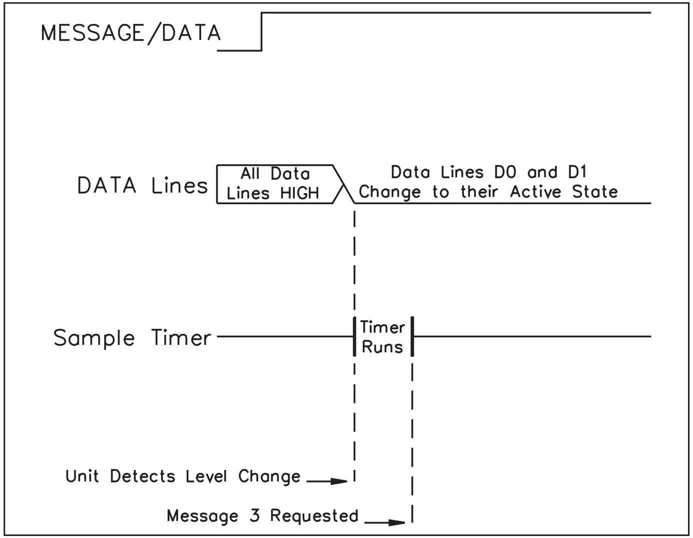


Figure 22, AMR Mode 2 Timing

Note: Shown for CTRL and DATA LOGIC Switches in the NEGATIVE position (DOWN).

Receiving Embedded Data

Messages can collect Embedded Data over the Parallel Port. Individual data items can collect ASCII, BCD, or Binary data, independent of the current port Type configuration setting. If the Parallel Port is configured for 4-bits, the ADI/MDI expects to receive data 4-bits at a time. Otherwise, the unit expects to receive data 8-bits at a time.

Unlike the other Embedded Data types, a data item collected over the Parallel Port must specify the length, in bytes, of the data to be collected. This length, combined with the port Width setting, determines the number of STROBE pulses necessary to input a data item over the port. The unit automatically proceeds with Message processing once it has received the required number of STROBEs.

The parallel port Embedded Data Time Out value determines how long the ADI/MDI waits to receive a data item over the port. This value can be taken from the unit's default parallel port Time Out value, or can be individually specified for any or all items in a Message. The Time Out value can range from 0 to 254 seconds, or can be turned OFF, forcing the unit to wait indefinitely to receive a data item.

If the ADI/MDI receives a Message request over the Parallel Port while already processing a Message, it aborts processing the current Message, deletes all data the Message has collected to the point of interruption, and begins accepting the new Message request by reading the port DATA lines. All Local items, and any Indexed items the Message had collected before the interruption, are deleted. Any remaining Indexed items the Message didn't have a chance to collect retain their old values.

Binary Data

The ADI/MDI can collect a maximum of 2 bytes (16 bits) of data per Binary item. After the unit has received the specified number of Binary bytes, it converts the 8 or 16 bit value to decimal, storing the result as a 5 digit field, padded with leading zeroes, unless formatted otherwise.

The ADI/MDI reads DATA lines D0 to D7 for an 8-Bit parallel port width and requires a single STROBE per byte. With the port configured for 4-bits, the unit only reads DATA lines D0 to D3, and requires two STROBEs per byte. The first STROBE presents the high-order nibble (upper 4 bits) of a byte, the second STROBE presents the low-order nibble.

If the data item specifies a length of 1 byte, the unit only reads one byte, which can have a maximum value of 255, which is stored as "00255" before any specified formatting. For a data length of 2 bytes, the unit reads the high order byte first, followed by the low-order byte. A 2 byte Binary data item can have a maximum value of 65535.

The minimum time between successive STROBE pulses is determined by the Debounce time, and can range from 10 milliseconds to 2.5 seconds.

BINARY TABLE					
HIGH ORDER BYTE		DECIMAL VALUE	LOW ORDER BYTE		DECIMAL VALUE
HIGH NIBBLE	LOW NIBBLE		HIGH NIBBLE	LOW NIBBLE	
D7 D6 D5 D4	D3 D2 D1 D0		D7 D6 D5 D4	D3 D2 D1 D0	
0 0 0 0	0 0 0 0	0	0 0 0 0	0 0 0 0	0
0 0 0 0	0 0 0 1	256	0 0 0 0	0 0 0 1	1
0 0 0 0	0 0 1 0	512	0 0 0 0	0 0 1 0	2
0 0 0 0	0 1 0 0	1024	0 0 0 0	0 1 0 0	4
0 0 0 0	1 0 0 0	2048	0 0 0 0	1 0 0 0	8
0 0 0 1	0 0 0 0	4096	0 0 0 1	0 0 0 0	16
0 0 1 0	0 0 0 0	8192	0 0 1 0	0 0 0 0	32
0 1 0 0	0 0 0 0	16384	0 1 0 0	0 0 0 0	64
1 0 0 0	0 0 0 0	32768	1 0 0 0	0 0 0 0	128

Note: The decimal equivalent of a Binary number is the sum of all the decimal values of the '1' bits.

BCD Data

The ADI/MDI can collect a maximum of 11 bytes of data per BCD item. Each byte collected from the Parallel Port consists of 2 BCD digits, and results in 2 characters being stored in the unit. Therefore, although the maximum size of a BCD item is 22 characters in the unit, its maximum bytes setting is 11 bytes, as counted at the Parallel Port.

For an 8-bit port Width, the ADI/MDI reads DATA lines D4 to D7 as the high-order digit and DATA lines D0 to D3 as the low-order digit, and requires a single STROBE per byte. With the port configured for 4-bits, the unit only reads DATA lines D0 to D3, and requires two STROBEs per byte. The first STROBE presents the high-order digit (upper 4 bits) of a byte, the second STROBE presents the low-order digit. BCD Items containing multiple bytes should be presented to the port from high-order byte to low-order byte (left to right).

The minimum time between successive STROBE pulses is determined by the Debounce time, and can range from 10 milliseconds to 2.5 seconds.

BCD TABLE

MSD				BCD	LSD				BCD
D7	D6	D5	D4	VALUE	D3	D2	D1	D0	VALUE
0	0	0	0	0	0	0	0	0	0
0	0	0	1	10	0	0	0	1	1
0	0	1	0	20	0	0	1	0	2
0	0	1	1	30	0	0	1	1	3
0	1	0	0	40	0	1	0	0	4
0	1	0	1	50	0	1	0	1	5
0	1	1	0	60	0	1	1	0	6
0	1	1	1	70	0	1	1	1	7
1	0	0	0	80	1	0	0	0	8
1	0	0	1	90	1	0	0	1	9
1	0	1	0	Invalid	1	0	1	0	Invalid
1	0	1	1	↑	1	0	1	1	↑
1	1	0	0		1	1	0	0	
1	1	0	1		1	1	0	1	
1	1	1	0	↓	1	1	1	0	↓
1	1	1	1	Invalid	1	1	1	1	Invalid

Note: The decimal equivalent of a BCD number is: (MSD x 10) + LSD.

ASCII Data

The ADI/MDI can collect a maximum of 23 bytes per ASCII item. Each byte is interpreted as an 8-bit ASCII character. The unit does not display the ASCII characters from 0h (0) to 1Fh (31). Extended ASCII characters greater than B8h (184) are displayed as '.'.

For an 8-Bit parallel port, the unit receives one ASCII character per STROBE pulse. With a 4-bit port configuration, the first STROBE presents the high-order nibble (upper 4 bits) of the ASCII character, the second STROBE presents the low-order nibble (lower 4 bits). ASCII items containing multiple bytes should be presented to the unit from high-order character to low-order character (left to right).

The minimum time between successive STROBE pulses is determined by the Debounce time, and can range from 10 milliseconds to 2.5 seconds.

HEXADECIMAL TABLE

MSD				HEX	LSD				HEX
D7	D6	D5	D4	VALUE	D3	D2	D1	D0	VALUE
0	0	0	0	0	0	0	0	0	0
0	0	0	1	10	0	0	0	1	1
0	0	1	0	20	0	0	1	0	2
0	0	1	1	30	0	0	1	1	3
0	1	0	0	40	0	1	0	0	4
0	1	0	1	50	0	1	0	1	5
0	1	1	0	60	0	1	1	0	6
0	1	1	1	70	0	1	1	1	7
1	0	0	0	80	1	0	0	0	8
1	0	0	1	90	1	0	0	1	9
1	0	1	0	A0	1	0	1	0	A
1	0	1	1	B0	1	0	1	1	B
1	1	0	0	C0	1	1	0	0	C
1	1	0	1	D0	1	1	0	1	D
1	1	1	0	E0	1	1	1	0	E
1	1	1	1	F0	1	1	1	1	F

Note: The decimal equivalent of a hexadecimal value is the sum of the decimal values of each '1' bit.

Receiving Embedded Data (Cont'd)

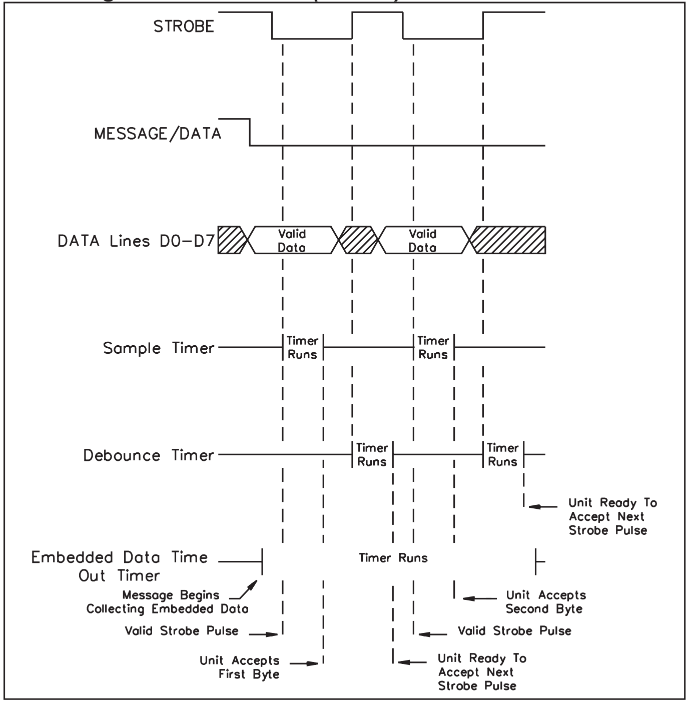


Figure 23, 8-BIT Embedded Data Timing

(Shown for 2 Data Bytes)

Note: Shown for CTRL LOGIC switch in the NEGATIVE position (DOWN).

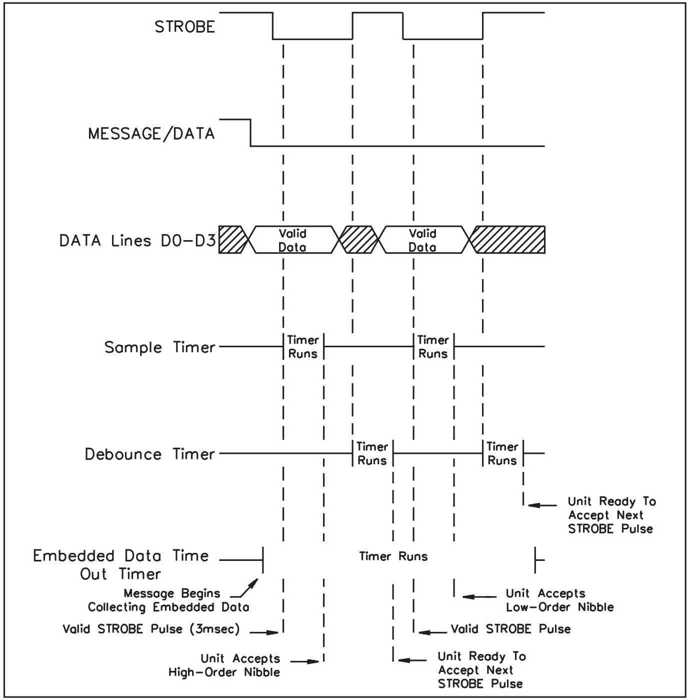

STROBE
MESSAGE/DATA
DATA Lines D0-D3
Valid Data
Valid Data
Sample Timer
Timer Runs
Timer Runs
Debounce Timer
Timer Runs
Timer Runs
Unit Ready To Accept Next STROBE Pulse
Embedded Data Time Out Timer
Timer Runs
Message Begins Collecting Embedded Data
Unit Accepts Low-Order Nibble
Valid STROBE Pulse (3msec)
Valid STROBE Pulse
Unit Accepts High-Order Nibble
Unit Ready To Accept Next STROBE Pulse

Figure 24, 4-BIT Embedded Data Timing

(Shown for 1 data Byte - 2 Nibbles)

Note: Shown for CTRL LOGIC switch in the NEGATIVE position (DOWN).

PERIODIC MESSAGE FUNCTION

The ADI/MDI contains an extensive Periodic Message function, which holds up to 32 separate Periodic Entries. The Periodic function allows you to specify individual Messages for Periodic request over a variety of intervals, such as once per minute, once every 8 hours, once per month, etc. You can turn the Periodic function ON and OFF in the Configuration file, or with the C01 and C03 Commands. You can configure the Periodic Entry settings for a Message File in the PERIODIC MESSAGE LIST sub-menu by positioning the cursor on the PERIODIC/DATED field and hitting the SPACE BAR or clicking the mouse. (See *Periodic Messages*, page 19, for details).

Each Periodic Entry specifies an ENABLE/DISABLE flag, an ON/OFF flag, a Message number, an Activation Time, and a Periodic Interval. You can assign any programmed Message to any number of Periodic Entries. The Activation Time specifies the Date and Time that the unit should next request the Message assigned to the Periodic Entry. The Periodic Interval specifies the frequency that the unit should request the Message.

A list of the Periodic Intervals is shown below. With the exception of the various SECONDS intervals, all Periodic Activation Times occur on a minute boundary.

SECONDS	- Every 1,2,3,4,5,10,15,20,30 Seconds.
MINUTES	- Every 1-99 Minutes.
HOURS	- Every 1-99 Hours.
DAYS	- Every 1-99 Days.
2/MONTH	- Semi-Monthly.
MONTHS	- Every 1-99 Months.
1ST DOM	- Every 1ST Sunday to Saturday (1-7) of the month.
2ND DOM	- Every 2ND Sunday to Saturday (1-7) of the month.
3RD DOM	- Every 3RD Sunday to Saturday (1-7) of the month.
4TH DOM	- Every 4TH Sunday to Saturday (1-7) of the month.
LAST DOM	- Every LAST Sunday to Saturday (1-7) of the month.
END-OF-MON	- Every End-Of-The-Month.
END-OF-QTR	- Every End-Of-The-Quarter.
ONE TIME	- Activate ONCE, and then DISABLE the Periodic Entry.

You can ENABLE and DISABLE individual Periodic Entries, as well as turn them ON and OFF, with the C03 Command (“P” mnemonic). The ADI/MDI ignores all DISABLEd Periodic Entries. All Periodic Entries your application no longer needs should be DISABLEd. This action decreases the processing overhead of the unit.

Any Periodic Entry that is ENABLEd, but not turned ON, is shown as OFF. While a Periodic Entry is in the OFF state, the unit updates its Activation Time with the Periodic Interval, but does not request the Message for that Periodic Entry. In the ON state, the unit requests the Message every time the Current Time reaches the Activation Time. The activation time can be set for any time up to the year 2079.

Command Messages can be programmed to turn Periodic Entries ON and OFF as the application needs them. For instance, if an application executes a Message, or a sequence of Messages, at different intervals depending on the existing system conditions, this function can be implemented by turning the appropriate Periodic Entries ON and OFF.

The ADI/MDI and the SFMD software both contain automatic Periodic Catch-Up routines. These routines come into play when the unit has been turned OFF for an extended period, or when a Message File that hasn’t been accessed recently is OPENed. If the ADI/MDI or the SFMD software detect a Periodic Activation Time that has already passed, as determined by the ADI/MDI’s Current Time, or the PC’s Real Time Clock, and the Periodic Entry is ENABLEd (ON or OFF), the software updates the Activation Time for that Entry by the appropriate number of Intervals, until the Activation Time is current. The unit does not issue Periodic Message requests while performing this “catch-up” routine for an Entry.

The ADI/MDI can process multiple Periodic Message requests programmed to occur at the same time. Periodic Entries are processed in a round-robin fashion in numerical order from Periodic Entry 0 to Periodic Entry 31. For instance, if Periodic Entry 0 and Periodic Entry 10 have identical Activation Times, the unit processes Periodic Entry 0 first, and then Periodic Entry 10.

You can also perform adjustments to the Activation Times for any and all Periodic Entries, as necessary. For example, this action may be required to accommodate changes from Daylight Savings Time (DST) to Standard Time and back. Commands C02 and C03 can be used to adjust the Periodic Activation Time. (See COMMANDS C02 and C03 for details).

SERIAL COMMUNICATIONS

RS232 Port (ADI Only)

The “20 mA DIS.” DIP switch is used to disable the 20 mA receive loop for applications that only use the RS-232 port. See *20 mA port Disable and Serial Port Interaction*, for details on the operation and usage of the RS232 serial port.

20 mA Current Loop

The ADI/MDI products are equipped with a 20 mA current loop, that provides full duplex, isolated, two-way communications. ASCII terminals, programmable controllers, computers, and Red Lion Controls products with 20 mA communications are just some of the devices that can be connected to the serial port.

Two loops are required for all 20 mA hook-ups, a transmit loop (SO+ & SO-) and a receive loop (SI+ & SI-). The built-in +20 mA current source can drive either the transmit loop or the receive loop. The -20 mA SRC is the current loop return connection. To minimize problems with ground loops, keep the -20 mA SRC isolated from earth ground.

Multiple units can be connected to the serial loop, but the actual number of serial connections that the ADI/MDI can drive is limited by its serial hardware specifications.

The following operations can be performed over the serial port:

Uploading and Downloading Files
Requesting Messages
Transmitting and Receiving Message Text
Transmitting and Receiving Commands
Collecting and Transmitting Embedded Data
Transmitting Error Codes

You can set and verify the Serial Port configuration settings with the C05 Command. Through the appropriate DIP switch settings, the ADI/MDI can perform a serial loop-back test to verify the proper operation of its serial communications hardware.

Communication Format

For 20 mA current loop communications, the unit transmits characters by switching the current ON and OFF in its 20 mA transmit loop and receives characters by monitoring the switching action on its 20 mA receive loop. The data format and baud rate must be identical between the transmitting and receiving units for the data to be interpreted correctly.

The baud rate, parity bit, and data bit parameters are DIP switch selectable. The available data formats are:

11 bits: 1 start bit, 8 data bits, Parity bit, 1 stop bit.
10 bits: 1 start bit, 7 data bits, Parity bit, 1 stop bit.
10 bits: 1 start bit, 8 data bits, NO Parity, 1 stop bit.
9 bits: 1 start bit, 7 data bits, NO Parity, 1 stop bit.

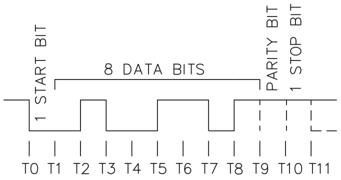


Figure 25, 11 BIT Data Format

(Can also be 8 Data Bits and 1 Stop Bit)

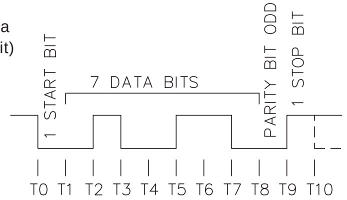


Figure 26, 10 BIT Data Format

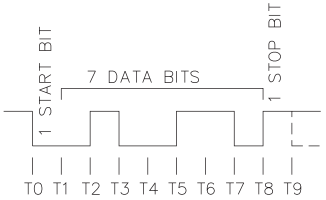


Figure 27, 9 BIT Data Format

The available baud rates are; 300, 600, 1200, 2400, 4800, 9600, and 19200.

If only one ADI/MDI is present in a loop, an address number of "0" may be assigned to that unit. Assigning an address of zero to an ADI/MDI eliminates the need for including an address identifier when communicating with that unit. If multiple unit's are present on the loop, it is recommended that you assign each unit a unique address, from 1-99. Although there are 100 valid addresses, the built in current source, if used, can only drive a loop with a maximum of 6 units on it. You can set the unit address of an ADI/MDI with the C01 Command.

Serial Port Operation

The ADI/MDI utilizes two buffers for its Serial Port, a 128 byte Receive Buffer and a 256 byte Transmit Buffer.The unit can service both the Receive Buffer and the Transmit Buffer at the same time.

Serial Receive Buffer

With a few exceptions, the ADI/MDI places all characters it receives directly into its receive buffer. The unit loads characters into the buffer, without processing them, until it receives its programmed Serial Terminator character. The Terminator character informs the unit that it should process the current contents of its Receive buffer.

The ADI/MDI can accept character strings up to 128 bytes in length, including the Terminator character. Should the unit receive a 129th character without an intervening Terminator, it empties the Receive buffer and places the 129th character in the buffer. The unit does not issue an Error Code for this condition.

Several characters invoke special Receive buffer functions. The BACKSPACE character (08h - 8) causes the ADI/MDI to back-up one character position in its Receive buffer, effectively deleting the last character received. Issuing repeated BACKSPACEs to a unit eventually empties its Receive buffer. The unit ignores the BACKSPACE character while its Receive buffer is empty.

The ESCAPE character (1Bh - 27) causes the ADI/MDI to empty its Receive buffer. In effect, the ESCAPE character deletes all characters sent to the unit since the last Terminator character, clearing the Receive buffer. You can include the ESCAPE character in any transmission. When the unit encounters the ESCAPE in the Transmit Buffer, it clears the Receive Buffer. You can use this feature to accommodate difficult serial timing constraints.

The ASCII NULL character (00h - 0) acts as a Universal Serial Terminator character for all ADI/MDI and MDS units. Pressing <CTRL-END> while in the Terminal Emulator issues this character. See the Serial Terminator description below for details.

The <CTRL-N> character (0Eh - 14) causes the ADI/MDI to insert its unit address prefix string into the Receive buffer at the current character position. Pressing <CTRL-HOME> while in the SFMD Terminal Emulator performs the same function. This function, combined with the ESCAPE character and the Universal Terminator, allows you to access any ADI/MDI or MDS without knowing the unit address or the Serial Terminator of the unit.

Example: The character sequence "<ESC><CTRL-HOME>C1<CTRL-END>", issued from the SFMD Terminal Emulator, causes the following actions:

<ESC> - The ADI/MDI empties the Receive buffer.

<CTRL-HOME> - The SFMD Terminal Emulator transmits a <CTRL-N> character, causing the ADI/MDI to insert its unit address prefix in the Receive buffer. If the address is 34, the character string "N34" now appears in the Receive buffer.

C1 - The ADI/MDI loads these characters into its Receive buffer, which invoke Command C01 when the unit processes the Receive buffer.

<CTRL-END> - The Terminal Emulator transmits a <00h>, causing the unit to terminate the character string in its Receive buffer and begin processing the string. The Receive buffer now contains: "N34C1". While processing the Receive buffer contents, the ADI/MDI identifies its own unit address, and consequently processes the remainder of the string.

The character string issues a request for Command C01, which causes the ADI/MDI to transmit its Unit Function configuration to the Terminal Emulator. The ADI/MDI's unit address, which was unknown to the operator, is included in this transmission.

Note: Multiple units on a loop will all receive and execute the preceding transmission, thus causing a collision between the multiple transmissions. This is not harmful to the units on the loop, but does result in a corrupted transmission.

Serial Transmit Buffer

All characters transmitted by the ADI/MDI originate from the Transmit Buffer. The unit assembles a transmission string in its Transmit buffer before sending that string out on the serial port. Transmission strings are limited in length by the size of the Transmit buffer. The ADI/MDI can transmit character strings of up to 255 bytes, including the designated string terminator, which is usually the unit's Serial Terminator character. Any attempt to assemble transmission strings longer than 255 characters results in the unit issuing an Error Code and aborting the transmission.

You can PAUSE, RESUME, and ABORT certain transmissions by issuing the ESCAPE and <CR> characters. While transmitting responses for any of the Commands, the unit accepts the ESCAPE character as a signal to PAUSE the transmission at the end of the current line. The ADI/MDI RESUMEs the transmission once it receives a <CR> character. If the unit receives a second ESCAPE character before the transmission is RESUMEd, the unit aborts the transmission by transmitting a closing <CR><LF> pair.

Note: The ADI/MDI cannot process Message Time Outs, or any new Message requests, while a transmission is PAUSEd. Periodic Entries that reach their Activation times, and Elapsed Timers that reach their Trigger values are marked for processing while the transmission is PAUSEd, but are not processed until the transmission is RESUMEd.

Serial Terminator Character

The ADI/MDI utilizes its Serial Terminator character both to recognize that it should process its Receive buffer, and to terminate most of its transmission strings in the Transmit Buffer. You can program the Terminator in the range from 01h to 40h, excluding the BACKSPACE character (08h - 8), the ESCAPE character (1Bh - 27), and the 10 numeric characters (30h (48) to 39h (57). As with all RLC products with serial communications, the default value for the Terminator is '✳'.

A Universal Terminator character is also available for your use, the ASCII NULL character (00h - 0). All ADI/MDI and MDS units respond to this character as they do to their respective programmed Serial Terminator character. Therefore, if you forget, or cannot identify a particular unit's programmed Terminator, you can use this character in the interim.

Many ASCII terminals and Terminal Emulation programs cannot transmit this character. However, you can issue this character from the RLC Terminal Emulator by pressing <CTRL-END>.

Serial port Embedded Data items can utilize the ADI/MDI's Serial Terminator, or they can specify their own Terminator character, independent of the unit's Terminator. While the unit is collecting a data item from the serial port, it accepts the Terminator specified by that item. An Embedded Data item's Terminator can range from 01h (1) to FFh (255), excluding the BACKSPACE and ESCAPE characters.

Serial Transmit Delay

All Red Lion Controls products containing serial ports utilize Receive buffers. When a particular RLC unit receives its Terminator character, usually an '✳', it must interrupt the current task it is processing, in order to process the contents of the Receive buffer.

This interruption takes some finite amount of time, on the order of 10's or even 100's of milliseconds. The unit cannot receive any additional characters while it is processing the buffer, and, once finished, must return to the interrupted task at hand. Therefore, a method for establishing a minimum time between consecutive transmissions is necessary.

The Serial Transmit Delay setting controls the minimum delay time between the end of one transmission string and the beginning of the next string from the unit. The Transmit Delay value can range from 0 seconds (NO delay) to 2.5 seconds. The ADI/MDI automatically runs a Transmit Delay at the end of every transmission string. It also runs Transmit Delays after <CR><LF> pairs in all of its Command response transmissions.

In addition, provisions have been made for you to insert Transmit Delays anywhere in a transmission string that you deem necessary or essential. In order to force the unit to run a Transmit Delay at a particular location in a transmission string, you insert the ' t' character (ALT-T) at the desired location in the string. When the unit encounters the ' t' character while processing its Transmit buffer, it runs a Transmit Delay instead of transmitting the character. Once the Transmit Delay has expired, the unit continues on to the next character in the buffer.

TRANSMIT Message Destination

The TRANSMIT destination allows you to construct and transmit character strings in any format you desire. The unit assembles the text for TRANSMIT destination Messages in its Transmit buffer and then commences transmitting the string.

While assembling the transmission, the ADI/MDI replaces all END-OF-LINE (' l') and END-OF-BLOCK (' s') control codes with <CR><LF> pairs and removes all Blink (' b') control codes. The unit replaces all Current Time (' c'), Elapsed Time (' e'), Local ED Item (' d'), and Index ED Item (' i') fields with the appropriate values.

During the transmission, if the ADI/MDI encounters the <ESC> character (1Bh-27), it clears the Receive Buffer at that point in the transmission. Whenever the unit encounters the '^' character, it transmits the <ESC> character. The ADI/MDI does not insert its own Serial Terminator character, you must supply the desired Terminator character(s) at the end of the string.

The unit does run a Transmit Delay at the end of this transmission. If your application requires additional Transmit Delays, you can insert them in the Message text by placing the ' t' character at the desired location(s).

If the assembled transmission string overflows the Transmit buffer (is longer than 255 bytes), the ADI/MDI issues an Error Code and cancels the transmission. The SFMD software does not allow you to program a Message string that exceeds this length. However, the software cannot anticipate the size of any embedded data values the Message text may include. If you have doubts as to the assembled transmission string length, execute the Message in the SFMD Message Simulator while supplying the appropriate data. The Simulator checks the transmission string length and informs you of a Transmit buffer overflow condition. See *MESSAGE SIMULATOR FUNCTION*, page 105, for details.

MDS UNIT Message Destination

The MDS UNIT destination causes the ADI/MDI to transmit the specified Message text in a compatible format to the unit address you selected in the Message. The ADI/MDI assembles the text for the MDS UNIT destination in its Transmit buffer and then commences transmitting the string.

The unit assembles the transmission string intended for the MDS as a Temporary Message. See *Temporary Messages*, page 36, for details on the structure and format of Temporary Message strings.

The unit terminates the transmission string with its own Serial Terminator character, which you should program to match that of the unit to which the string is transmitted. You can view the resulting transmission string by calling the Message Simulator, or by downloading the Message file to the ADI/MDI and requesting the Message from the Terminal Emulator.

Since this transmission string is intended for an MDS unit, and the MDS Receive buffer is only 127 bytes long, the ADI/MDI transmits an Error Code to the MDS if the assembled transmission string exceeds the Receive buffer length of 127 bytes. The unit cancels the remainder of the transmission, and the MDS displays the Error.

The SFMD software does not allow you to program a Message string that exceeds this length. However, it cannot anticipate the size of any embedded data values the Message text may include. If you have doubts as to the assembled transmission string length, execute the Message in the SFMD Message Simulator while supplying the appropriate data. The Simulator checks the transmission string length and informs you of an MDS Receive buffer overflow condition.

Serial Port Timing Diagrams

When you are interfacing with the ADI/MDI over its Serial Port, certain considerations must be made regarding the timing of the various serial receptions and transmissions. Reliable communication over the Serial Port between the ADI/MDI and any units connected to it is dependent upon several factors; the serial baud rate, the serial terminator(s), and the serial transmit delay(s).

The serial baud rate setting directly affects the transmission time for a character string. At 300 baud, the unit takes approximately one second to transmit a string of 30 characters. At 9600 baud, however, the same character string can be transmitted in 30 milliseconds. The following table lists the valid baud rates and the transmission time per character for each baud rate.

300 -	33.3 msecs/character
600 -	16.7 msecs/character
1200 -	8.3 msecs/character
2400 -	4.2 msecs/character
4800 -	2.1 msecs/character
9600 -	1.0 msecs/character
19200 -	0.5 msecs/character

All serial communications with the ADI/MDI are buffered. The unit places all characters it receives in its Serial Receive Buffer until the appropriate Serial Terminator character is received. Upon receipt of the Terminator, the unit “reads” the contents of the Receive Buffer, and performs the action the character string in the buffer specifies. If the ADI/MDI does not “understand” the character string, or the string is not addressed to the unit, no action is taken. If the string requests a valid function, such as a Message request or a Command string, and the ADI/MDI subsequently finds an error in the string, it generates an Error Code, which is then processed according to the configured Error Handling method.

A Serial Transmit Delay causes the unit to delay transmitting the next character until the Delay time expires. Transmit Delays are incorporated in and between transmissions to accommodate the various serial timing and processing constraints of the units with which the ADI/MDI must communicate.

The following figures illustrate the different timing constraints the unit adheres to when receiving and processing Message requests and Command strings, and when collecting Embedded Data for a Message. The READY/BUSY state of the unit is shown for the various stages of processing. For the BUSY Mode of the Output Pin function, these states correspond to the appropriate Output pin logic levels. For the XON/XOFF handshaking mode, the XON and XOFF transmissions are shown, as well. (See *HANDSHAKING*, page 50, for details).

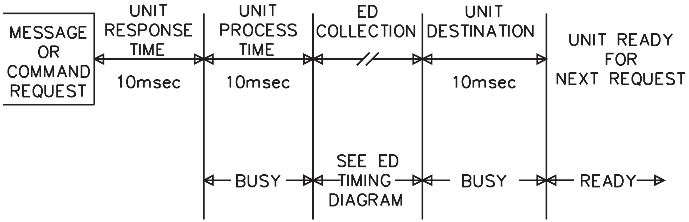


Figure 28, Ready/Busy Handshake Protocol

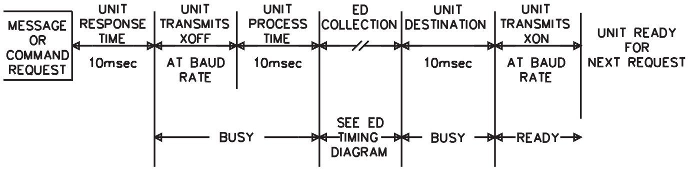


Figure 29, XON/XOFF Handshake Protocol

Embedded Data Timing Diagrams

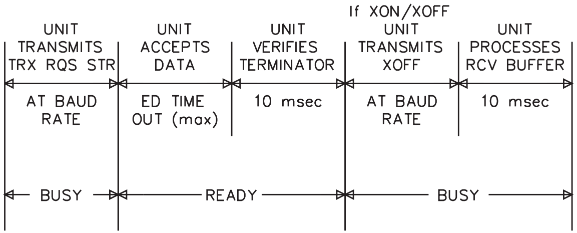


Figure 30, Serial/Discard Data w/Transmit Request

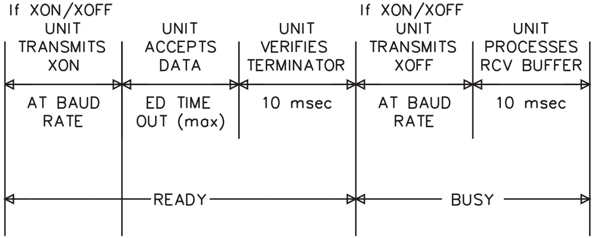


Figure 31, Serial/Discard Data w/o Transmit Request

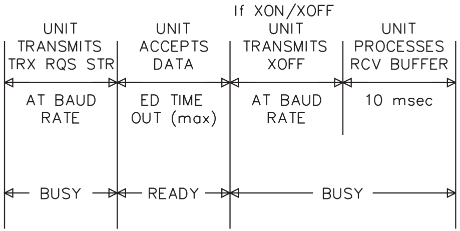


Figure 32, Parallel Port Data w/Transmit Request

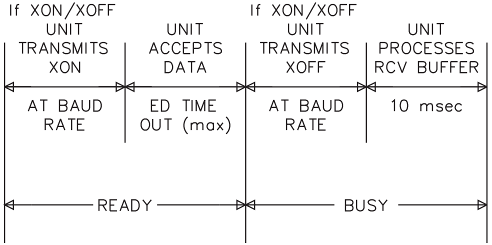


Figure 33, Parallel Port Data w/o Transmit Request

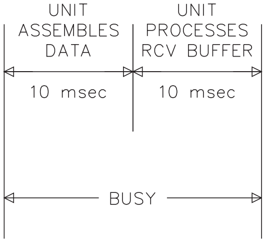


Figure 34, Internal Data

UNIT POWER-UP AND RESET

Upon application of power (unit power-up), the unit executes the following sequence of Reset functions:

1. Sets the Output pin to the current Configuration File settings. While the unit is powering up, the Output Pin will momentarily conduct, before the processor has a chance to configure it. If the Output pin is configured for the Ready/busy mode, the unit sets the pin to the BUSY state. If the pin is in the OUTPUT mode, the unit sets the pin to the OFF state. The actual voltage level of the pin is determined by the current Output Pin Logic Level configuration setting.
2. Retains the configurations of all Elapsed Timers set for SAVE. At this point in the Unit Reset sequence, a SAVEd Timer that was RUNning when the unit powered-down starts to RUN again from its power-down value. The Timer's direction and Trigger settings are unaffected.

   The unit resets the configurations of all Elapsed Timers set for CLEAR. The unit turns a CLEARed Timer OFF, sets the Timer's direction to DOWN, turns the Trigger function OFF, sets the Trigger for ONE-SHOT operation, sets the Trigger Message to 0, and sets both the Timer and Trigger values to 0000:00:00.00.
3. If the ADI/MDI was processing a Message request when the last power-down occurred, it clears the Message Queue and deletes all Indexed and Local Embedded Data. Whenever the unit is processing a Message request, it could be modifying the Queue contents, or saving and deleting Embedded Data. If a power interruption occurs during the Message processing time, the unit cannot fully restore the Message Queue or the Embedded Data Storage area to their pre-request contents. Some data is invariably lost, or corrupted. Under these circumstances, the unit cannot guarantee the integrity of its Queue or the data it holds and must, therefore, clear the Message Queue and the Embedded Data Storage area at power-up.

   You can configure the ADI/MDI to retain its current Queue contents and Embedded Data Storage area on the next power-up by HALTing the unit with the "C19HLT" Command string before powering the unit down, or by guaranteeing that the unit is not processing a Message when power is removed. If you HALT the unit or verify that it is not processing a Message request before removing power, the unit retains all the Queue and Embedded Data information it contained at power-down when power is reapplied.
4. If the Queue function is ON, and there is at least one Entry on the Queue, the unit overrides the Reset Message function, and requests the Message from the Top of the Queue (Queue Entry 0). In this case, the Reset Message is not requested.
5. If the Reset Message function is ON, and the Queue function is OFF or the Queue is empty, the unit issues a request for the Reset Message.
6. If the unit has no Message to request from either the Queue or the Reset Message function, it then checks the status of the Default Message function. If the Default Message function is ON, the unit issues a request for the Default Message.
7. If the Default Message function is OFF, the unit blanks the display and enters the Unit READY condition. The ADI/MDI indicates its READY status through the currently configured handshaking protocol (i.e., it issues an XON for the XON/XOFF mode, or takes the Output pin to the READY state for the READY/BUSY mode).
8. If the unit found a Message to request from the Message Queue, the Default Message function, or the Reset Message function, it does not indicate the Unit READY status until after the requested Message has been processed.

COMMANDS

A full set of Commands is available to the user for both configuring and interrogating the unit. Commands can cause the unit to transmit any of its current Configuration File settings in an easily readable mnemonic format. You can also change any of these configuration settings "on-line" through the use of Commands. Consequently, the ADI/MDI can be programmed to adapt itself to changing system requirements. Commands also provide access to the Message Queue, the Elapsed Timers, the Output Pin, and any Embedded Data the unit collects.

You can issue Commands in one of two ways. You can either transmit the desired Command string to the unit over the serial port, or you can program the Command string as a Command Message in a Message file. This Message is then available for request by any of the available Message request sources. The unit executes the Command string programmed in the Command Message whenever it receives a request for that Message.

Command String format and assembly is described in the following section. The Command Message feature is described later in the Command Message section.

Assembling Commands

The ADI/MDI contains an extensive Command structure. Aside from embedded data collection, all Serial Port communications with the unit are performed through the use of Commands.

The unit can be assigned a specific Unit Address, which allows you to direct Command strings to specific units on the loop. You must access an ADI/MDI with an address other than 0 with the transmission format shown below:

NnnCxxdd..d <TERM>

'N'- Signifies that a unit address follows.

nn - The address number of the unit for which the Command string is intended, leading zeroes need not be included.

'C' - Signifies that a Command string follows.

xx - The number of the Command to be executed, leading zeroes need not be included

dd..d - Represents any data the Command may require.

<TERM> - Programmed Serial Terminator.

If a Unit has an address of 0, the Address identifier, "Nnn", need not precede the Command string; For example: "Cxxdd..d" <TERM>

All future references to Commands assume an Address of 0 and that the Serial Terminator is entered at the end of the string.

Command Syntax

To help you locate and interpret command syntax, the following conventions are used in this document:

1. "UPPER-CASE" characters represent Command Codes and should be entered exactly as shown.
2. "lower-case" characters indicate that a variable field is to be filled in. The number of characters shown for the field indicate that field's maximum size. Leading zeroes need not be included in numeric fields except where specifically indicated for a particular command.
3. "dd...d" indicates that a variable length field or variable length list should be entered.

Commands C01 through C09 accept multiple Command codes in a single data string. In addition, Commands C01 through C09 and Command C24 allow you to LINK the Command to one other Message through the use of the '/' character, followed by the desired Message number.

Example: "C07T1P0UN"

"C07" - Command C07 configures the Elapsed Timer.

"T1" - TIMER Command Code. Specifies TIMER 1.

"P0" - PRESET Command Code. PRESETs the Timer to the value following the Command Code. "0" in this case.

"U" - UP Command Code. Sets the Timer direction to UP.

"N" - ON Command Code. Turns the Timer ON.

Command Syntax (Cont'd)

Example: "C01QFPN/101"

"C01" - Command C01 configures the Unit Functions.

"QF" - QUEUE Command Code. 'F' turns the QUEUE function OFF.

"PN" - PERIODIC Command Code. 'N' turns the PERIODIC function ON.

"/101" - LINKs Message 101. After the ADI/MDI executes the Command string, it automatically requests Message 101.

All Commands can be issued directly through their respective Command numbers, as shown above. A few selected Commands can also be issued with a Mnemonic code related to the specific function of the Command. These selected Commands are listed below, along with their respective Mnemonic codes:

MNEMONIC CODE	COMMAND NUMBER	DESCRIPTION
I	C09	Index Embedded Data Command - The 'I' Mnemonic can be used in place of C09, if the Command data string begins with an 'I', i.e., "C09I1:1234" and "I1:1234" both load Index Item 1 with "1234".
M	C21	Request Message Command - The 'M' Mnemonic can be used in place of C21, if the Command data string begins with an 'M', i.e., "C21M23" and "M23" both issue a request for Message 23.
P	C03	Periodic Message Command - The 'P' Mnemonic can be used in place of C03, if the Command data string begins with a 'P', i.e., "C03P15EN" and "P15EN" both Enable and turn ON Periodic Entry 15.
Q	C22	Transmit Queue Entry Command - The 'Q' Mnemonic can be used in place of C22, if the Command data string begins with 'Q', i.e., "C22Q10" and "Q10" both cause the unit to transmit the Queue information for Queue Entry 10.
T	C07	Elapsed Timer Command - The 'T' Mnemonic can be used in place of C07, i.e., "C07T1P0UN" and "T1P0UN" both set Timer 1 to 0, set the direction to UP, and turn Timer 1 ON.

Commands C03, C07, and C09 allow you to access multiple Periodic Entries, Elapsed Timers, and Indexed Items, respectively, in a single Command string with the '/' character. You can request one transmission per command string and the transmission request must be the last entry in the string, otherwise the unit ignores the transmission request and proceeds to the next Command Code.

Example: "P0EN/P1F/P2F"

"P0EN" - ENABLEs and turns ON PERIODIC entry 0.
'/' - Indicates another Command string follows.
"P1F" - Turns PERIODIC entry 1 OFF.
'/' - Indicates another Command string follows.
"P2F" - Turns PERIODIC entry 2 OFF.

Example: "T1P0UN/T15F/102"

"T1P0UN" - Configures TIMER 1 with a PRESET of 0, sets its direction to UP, and turns the Timer ON.
'/' - Indicates another Command string follows.
"T15F" - Turns TIMER 15 OFF.
'/' - Indicates another Command string follows.
"102" - LINKs Message 102. After the unit executes the Command string, it automatically requests Message 102.

Note: The ADI/MDI executes a Command function as it processes the Command code and data for that function. If a Command is entered incorrectly, the unit aborts processing the Command at the first point of error and issues an error code based on the selected Error Handling method. All Command functions executed prior to the error remain in effect. Any Command functions following the error are ignored. If the Command is part of a Message Chain, the chain is aborted.

COMMAND LIST

COMMAND	FUNCTION	MNEMONICS	LINK
C01	Configure Unit Functions		YES
C02	Configure Real Time Clock		YES
C03	Configure Periodic File	P	YES
C04	Configure Parallel Port		YES
C05	Configure Serial Port		YES
C06	Configure Output/busy Pin		YES
C07	Configure Elapsed Timers	T	YES
C08	Configure Unit Display		YES
C09	Configure Embedded Data	I	YES
C10	Unused		
C11	Unused		
C12	Unused		
C13	Unused		
C14	Unused		
C15	Unused		
C16	Unused		
C17	Restore Default Unit Configuration		
C18	Transmit Char And Day/Month Files To MDS Unit		
C19	Halt/Reset Unit		
C20	Cancel Message		
C21	Request Message	M	
C22	Transmit Queue Information	Q	
C23	Transmit Message Text		
C24	Transmit Character String		YES

Configure Unit Functions - C01

Command C01 configures the Unit Function Configuration settings. The following is the list of valid Command string formats for the C01 Command.

C01	The unit transmits the current Unit Function Configuration settings as shown. FUNCTIONS CFG ADDR-aa CLK-n/f DFLT-n/f mmm ERR-t PRDC-n/f QUE-n/f RST- n/f mmm **Mnemonic** **Description** ADDR Unit Address, aa (0-99) CLK Clock Status, n/f (ON/OFF) DFLT Default Function Status, n/f (ON/OFF) Default Message number, mmm (0-255) ERR Error Handling Type, t[Both (B), Display Ignore (I), Transmit (T)] PRDC Periodic Function Status, n/f (ON/OFF) QUE Queue Function Status, n/f (ON/OFF) RST Reset Function Status, n/f (ON/OFF) Reset Message number, mmm (0-255)
C01Aaa	Sets the Unit ADDRESS to aa (0-99).
C01CF	Turns the Unit CLOCK function OFF, the internal CLOCK halts.
C01CN	Turns the Unit CLOCK function ON, the internal CLOCK runs.
C01DF	Turns the Unit DEFAULT Message function OFF, the current DEFAULT message is retained.
C01DN	Turns the Unit DEFAULT Message function ON, the current DEFAULT message is retained.
C01Dmmm	Sets the Unit DEFAULT message to mmm (0-255), the current function status is retained.
C01DmmmF	Sets the Unit DEFAULT message to mmm (0-255), and turns the function OFF.
C01DmmmN	Sets the Unit DEFAULT message to mmm (0-255), and turns the function ON.

C01EB	Sets the Unit ERROR Handling function to BOTH Display and Transmit Error messages.
C01ED	Sets the Unit ERROR Handling function to DISPLAY Error messages.
C01EI	Sets the Unit ERROR Handling function to IGNORE Error messages.
C01ET	Sets the Unit ERROR Handling function to TRANSMIT Error messages.
C01PF	Turns the Unit PERIODIC function OFF
C01PN	Turns the Unit PERIODIC function ON
C01QF	Turns the Unit QUEUE function OFF, the current QUEUE contents are retained.
C01QN	Turns the Unit QUEUE function ON, the current QUEUE contents are retained.
C01RF	Turns the Unit RESET Message function OFF, the current RESET message is retained.
C01RNN	Turns the Unit RESET Message function ON, the current RESET message is retained.
C01Rmmm	Sets the Unit RESET message to mmm (0-255), the current function status is retained.
C01RmmmF	Sets the Unit RESET message to mmm (0-255), and turns the function OFF.
C01RmmmN	Sets the Unit RESET message to mmm (0-255), and turns the function ON.

Note: Command codes can be strung together to configure multiple settings in a single command string.

Example: C01DNR103NQNPF

DN - Turns the DEFAULT function ON.
R103N - Sets the RESET message to 103 and turns the function ON.
QN - Turns the Message QUEUE function ON.
PF - Turns the PERIODIC function OFF.

Configure Unit Clock (RTC) - C02

Command C02 configures the Unit Real Time Clock (RTC). The following is the list of valid command string formats for the C02 Command

C02	The unit transmits the current software CLOCK configuration as shown CLOCK CFG - n/f DATE: mm-dd-yy a TIME: hh:nn:ss.uu n/f - ON/OFF (Clock is running or halted) mm - MONTH (01-12) dd - DATE (01-31) yy - YEAR (00-99) a - DAY (1-7, Sunday is 1) hh - HOUR (0-23) nn - MINUTE (0-59) ss - SECOND (0-59) uu - HUNDREDTHS (00-99) *Note: The years 1980 to 1999 are indicated by 80 to 99, years 2000 to 2079 are indicated by 00 to 79*
C02Dmm-dd-yy a	Sets the RTC DATE to mm-dd-yy a mm - MONTH (01-12) dd - DATE (01-31) yy - YEAR (01-99) a - DAY (1-7, Sunday is 1) *Note: Leading zeroes must be entered.* *Note: The years 1980 to 1999 are indicated by 80 to 99, years 2000 to 2079 are indicated by 00 to 79.*
C02Thh:nn:ss	Sets the RTC TIME to hh:nn:ss. hh - HOUR (00-23) nn - MINUTE (00-59) ss - SECOND (00-59) *Note: Leading zeroes must be entered.* *Note: The years 1980 to 1999 are indicated by 80 to 99, years 2000 to 2079 are indicated by 00 to 79*
C02F C02N	Turns the Unit CLOCK function OFF. Turns the Unit CLOCK function ON.

C02+D	Adjusts the RTC Date forward 1 DAY.
C02+DP	Adjusts the RTC Date and all PRDC Entries forward 1 DAY.
C02+DPp1,p2..	Adjusts the RTC Date and PRDC Entries p1,p2.. (0-31) forward 1 DAY.
C02+tttD	Adjusts the RTC Date forward ttt (1-255) DAYs.
C02+tttDP	Adjusts the RTC Date and PRDC Entries forward ttt (1-255) DAYs.
C02+tttDPp1,p2..	Adjusts the RTC Date and PRDC Entries p1,p2.. (0-31) forward ttt (1-255) DAYs.
C02-D	Adjusts the RTC Date backward 1 DAY.
C02-DP	Adjusts the RTC Date and all PRDC Entries backward 1 DAY.
C02-DPp1.p2..	Adjusts the RTC Date and PRDC Entries p1,p2.. (0-31) backward 1 DAY.
C02-tttD	Adjusts the RTC Date backward ttt (1-255) DAYs.
C02-tttDP	Adjusts the RTC Date and all PRDC Entries backward ttt (1-255) DAYs.
C02-tttDPp1,p2..	Adjusts the RTC Date and PRDC Entries p1,p2.. (0-31) backward ttt (1-255) DAYs.
C02+H	Adjusts the RTC Time forward 1 HOUR
C02+HP	Adjusts the RTC Time and all PRDC Entries forward 1 HOUR.
C02+HPp1,p2..	Adjusts the RTC Time and PRDC Entries p1,p2... (0-31) forward 1 HOUR.
C02+tttH	Adjusts the RTC Time forward ttt (1-255) HOURs.
C02+tttHP	Adjusts the RTC Time and all PRDC Entries forward ttt (1-255) HOURs.
C02+tttHPp1,p2..	Adjusts the RTC Time and PRDC Entries p1,p2.. (0-31) forward ttt (1-255) HOURs.
C02-H	Adjusts the RTC Time backward 1 HOUR
C02-HP	Adjusts the RTC Time and all PRDC Entries backward 1 HOUR.
C02-HPp1,p2..	Adjusts the RTC Time and PRDC Entries p1,p2... (0-31) backward 1 HOUR.
C02-tttH	Adjusts the RTC Time backward ttt (1-255) HOURs.
C02-tttHP	Adjusts the RTC Time and all PRDC Entries backward ttt (1-255) HOURs.

Configure Unit Clock (RTC) - C02 (Cont'd)

C02-tttHPp1,p2..	Adjusts the RTC Time and PRDC Entries p1,p2.. (0-31) backward ttt (1-255) HOURs.
C02+M	Adjusts the RTC Time forward 1 MINUTE.
C02+MP	Adjusts the RTC Time and all PRDC Entries forward 1 MINUTE.
C02+MPp1,p2..	Adjusts the RTC Time and PRDC Entries p1,p2..(0-31) forward 1 MINUTE.
C02+tttM	Adjusts the RTC Time forward ttt 1-255) MINUTEs.
C02+tttMP	Adjusts the RTC Time and all PRDC Entries forward ttt (1-255) MINUTEs.
C02+tttMPp1,p2..	Adjusts the RTC Time and PRDC Entries p1,p2..(0-31) forward ttt (1-255) MINUTEs.
C02-M	Adjusts the RTC Time backward 1 MINUTE.
C02-MP	Adjusts the RTC Time and all PRDC Entries backward 1 MINUTE.
C02-MPp1,p2..	Adjusts the RTC Time and PRDC Entries p1,p2..(0-31) backward 1 MINUTE.
C02-tttM	Adjusts the RTC Time backward ttt 1-255) MINUTEs.
C02-tttMP	Adjusts the RTC Time and all PRDC Entries backward ttt (1-255) MINUTEs.
C02-tttMPp1,p2..	Adjusts the RTC Time and PRDC Entries p1,p2..(0-31) backward ttt (1-255) MINUTEs
C02+S	Adjusts the RTC Time forward 1 SECOND.
C02-S	Adjusts the RTC Time backward 1 SECOND.
C02+tttS	Adjusts the RTC Time forward ttt 1-255)SECONDs.
C02-tttS	Adjusts the RTC Time backward ttt 1-255) SECONDs.

Note: Command codes can be strung together to configure multiple settings in a single command string.

Example: C02D09-20-92 1T09:05:00N

D09-20-92 1	- Sets the RTC Date to 9-20-92, SUNDAY,
T09:05:00	- Sets the RTC Time to 9:05:00 AM, and
N	- Turns the INTERNAL Clock ON.

Example: C02+HP0,12,23

+H	- Adjusts the RTC forward 1 HOUR,
P0,12,23	- Adjusts PRDC Entries 0,12, and 23 forward 1 HOUR.

Configure Periodic Message List - C03 (P mnemonic)

Command C03 configures the Periodic Message List (PRDC). The following is the list of valid command string formats for the C03 Command.

C03 or P	The unit transmits the current Periodic Message List as shown. PERIODIC FILE CFG - n/f P# MSS DATE TIME INTERVAL pp s mmm mm-dd-yy hh:mm ii ttttt <This line is repeated for each programmed PRDC Entry> n/f - PRDC function status (ON/OFF) pp - PRDC Entry number (0-31) s - PRDC Entry status (Disabled (D), OFF (F), ON (N)) mmm - PRDC Message number (0-255) mm-dd-yy- Next activation DATE hh:mm - Next activation TIME ii - PRDC Interval ttttt - PRDC Interval Type

Note: The ESCAPE key <ESC> and the RETURN key <RET> PAUSE and RESUME transmission of a list of PRDC Entries. Hitting <ESC> PAUSEs the transmission. Hitting <RET> RESUMEs the transmission. Hitting <ESC> while the list is PAUSEd cancels the remainder of the transmission.

Warning: The MDI cannot process any message requests, TIME OUTs, or cancellations while the transmission is PAUSEd.

Note: Any command string containing the 'P' Command code can be issued without the Command number "C03".

C03Mmmm	Transmits a list of all PRDC Entries for MESSAGE mmm (0-255)
C03PD	Transmits a list of all DISABLED PRDC Entries.
C03PE	Transmits a list of all ENABLED PRDC Entries.
C03PF	Transmits a list of all Enabled OFF PRDC Entries.
C03PN	Transmits a list of all Enabled ON PRDC Entries.
C03Ppp	Transmits PRDC Entry pp (0-31).

C03DAYS	Transmits a list of all DAYS interval PRDC Entries.
C03DOM1	Transmits a list of all DOM1 interval PRDC Entries.
C03DOM2	Transmits a list of all DOM2 interval PRDC Entries.
C03DOM3	Transmits a list of all DOM3 interval PRDC Entries.
C03DOM4	Transmits a list of all DOM4 interval PRDC Entries.
C03DOML	Transmits a list of all DOML interval PRDC Entries.
C03EOM	Transmits a list of all END-OF-MONTH interval PRDC Entries.
C03EOQ	Transmits a list of all END-OF-QUARTER interval PRDC Entries.
C03HRS	Transmits a list of all HOURS interval PRDC Entries.
C03MINS	Transmits a list of all MINUTES interval PRDC Entries.
C03MONS	Transmits a list of all MONTHS interval PRDC Entries.
C03ONCE	Transmits a list of all ONCE interval PRDC Entries.
C03SECS	Transmits a list of all SECONDS interval PRDC Entries.
C03SEMI	Transmits a list of all SEMI-MONTHLY interval PRDC Entries.
C03F	Turns Unit Periodic function OFF.
C03N	Turns Unit Periodic function ON.
C03PppD	DISABLEs PRDC Entry pp (0-31), if already ENABLED.
C03PppE	ENABLEs PRDC Entry pp (0-31), the Entry is ENABLED, but turned OFF.
C03PppEN	ENABLEs and turns PRDC Entry pp (0-31) ON.
C03PppF	Turns PRDC Entry pp OFF (0-31), the Entry must already be ENABLED
C03PppN	Turns PRDC Entry pp ON (0-31), the Entry must already be ENABLED
C03P+D	Adjusts all PRDC Entries forward 1 DAY.
C03P+tttD	Adjusts all PRDC Entries forward ttt (1-255) DAYs.
C03Ppp+D	Adjusts PRDC Entry pp (0-31) forward 1 DAY.
C03Ppp+tttD	Adjusts PRDC Entry pp (0-31) forward ttt (1-255) DAYs.
C03P-D	Adjusts all PRDC Entries backward 1 DAY.
C03P-tttD	Adjusts all PRDC Entries backward ttt (1-255) DAYs.
C03Ppp-D	Adjusts PRDC Entry pp (0-31) backward 1 DAY.
C03Ppp-tttD	Adjusts PRDC Entry pp (0-31) backward ttt (1-255) DAYs.

Configure Periodic Message List - C03 (P mnemonic) (Cont'd)

C03P+H	Adjusts all PRDC Entries forward 1 HOUR.
C03P+tttH	Adjusts all PRDC Entries forward ttt (1-255) HOURs.
C03Ppp+H	Adjusts PRDC Entry pp (0-31) forward 1 HOUR.
C03Ppp+tttH	Adjusts PRDC Entry pp (0-31) forward ttt (1-255) HOURs.
C03P-H	Adjusts all PRDC Entries backward 1 HOUR.
C03P-tttH	Adjusts all PRDC Entries backward ttt (1-255) HOURs.
C03Ppp-H	Adjusts PRDC Entry pp (0-31) backward 1 HOUR.
C03Ppp-tttH	Adjusts PRDC Entry pp (0-31) backward ttt (1-255) HOURs.
C03P+M	Adjusts all PRDC Entries forward 1 MINUTE.
C03P+tttM	Adjusts all PRDC Entries forward ttt (1-255) MINUTEs.
C03Ppp+M	Adjusts PRDC Entry pp (0-31) forward 1 MINUTE.
C03Ppp+tttM	Adjusts PRDC Entry pp (0-31) forward ttt (1-255) MINUTEs.
C03P-M	Adjusts all PRDC Entries backward 1 MINUTE.
C03P-tttM	Adjusts all PRDC Entries backward ttt (1-255) MINUTEs.
C03Ppp-M	Adjusts PRDC Entry pp (0-31) backward 1 MINUTE.
C03Ppp-tttM	Adjusts PRDC Entry pp (0-31) backward ttt (1-255) MINUTEs.

Note: Command codes can be strung together to configure multiple functions in a single command string. Multiple PRDC Entries can be configured with the '/' delimiter.

Example: C03P0EN/P1EN/P31EN/P2F

P0EN - ENABLEs and turns PRDC Entry 0 ON,
/ - New PRDC Entry,
P1EN - ENABLEs and turns PRDC Entry 1 ON,
/ - New PRDC Entry,
P31EN - ENABLEs and turns PRDC Entry 31 ON,
/ - New PRDC Entry,
P2F - Turns OFF PRDC Entry 2.

Example: C03P2-2M/P3-H/P31+D

P2-2M - Adjusts PRDC Entry 2 backward 2 MINUTEs,
- New PRDC Entry,
P3-H - Adjusts PRDC Entry 3 backward 1 HOUR,
- New PRDC Entry,
P31+D - Adjusts PRDC Entry 31 forward 1 DAY.

Configure Parallel Port (PRL Port) - C04

Command C04 configures the Parallel Port settings. The following is the list of valid command string formats for the C04 Command.

C04	The unit transmits the current PRL PORT configuration as shown. PRL PORT CFG - pppppppppp DB-ddd0ms SM-sssms TO-ttts pppppppppp - PRL Port Type and Width (BCD 4/8/9 Bits, BIN 4/8 Bits, AMR Mode 1/2) DB - Debounce Time, ddd, in hundredths of seconds (1-255) SM - Sample Time sss, in milliseconds (1-255) TO - Embedded Data Time Out, ttt, in seconds, (OFF, 0-254)
C04BCD4	Sets the PRL Port to 4 Bits and BCD Message numbers (Requires 3 strobes (HI digit, MID digit, LO digit), can request messages 0-255).
C04BCD8	Sets the PRL Port to 8 Bits and BCD Message numbers (Requires 2 strobes (HI digit, MID/LO digits), can request messages 0-255).
C04BCD9	Sets the PRL Port to 9 Bits and BCD Message numbers (Requires 1 strobe, can only request messages 0-199).
C04BIN4	Sets the PRL Port to 4 Bits and BINARY Message numbers (Requires 2 strobes (HI nibble, LO nibble), can request messages 0-255).
C04BIN8	Sets the PRL Port to 8 Bits and BINARY Message numbers (Requires 1 strobe, can request messages 0-255).

C04AMR1	Sets the PRL Port for AMR Mode 1 operation.
C04AMR2	Sets the PRL Port for AMR Mode 2 operation.
C04Dttt	Sets the PRL Port DEBOUNCE Time to ttt (1-255) in hundredths of seconds (the SAMPLE time is automatically adjusted to be less than the DEBOUNCE time).
C04Sttt	Sets the PRL Port SAMPLE Time to ttt (1-255) milliseconds (the DEBOUNCE time is automatically adjusted to be greater than the SAMPLE time).
C04TF	Turns the PRL Port DEFAULT Embedded Data TIME OUT function OFF - NO Time Out.
C04Tttt	Sets the PRL Port DEFAULT Embedded Data TIME OUT to ttt (0-254) seconds.

Note: Command codes can be strung together to configure multiple functions in a single command string.

Example: C04BCD9D5S3T1

BCD9 - Sets the PRL Port to 9 Bits and BCD Message numbers,
D5 - Sets the DEBOUNCE Time to 50 milliseconds,
S3 - Sets the SAMPLE Time to 3 milliseconds,
T1 - Sets the DEFAULT Embedded Data TIME OUT value to 1 second.

Note: The parallel port embedded data must also adhere to this port width setting. However, the embedded data format is set by the message collecting the data (See Receiving Embedded Data under the Parallel Port section).

Configure Serial Port (SRL Port) - C05

Command C05 configures the Serial Port. The following is the list of valid command string formats for the C05 Command.

C05	The unit transmits the current SRL PORT configuration as shown. SRL PORT CFG - d,p,bbbbb DL-ddd0ms ST- TO-ttts X-n/f d - Number of Data bits (7/8) p - Parity type (Even (E), Odd (O), None (N)) bbbbb - baud rate (300 - 19200) DL - Transmit Delay, ddd, in hundredths of seconds (0-255) ST - Serial Terminator, sss, in decimal (0-7, 9-26, 28-47, 58-63) TO - Embedded Data Time Out, ttt, in seconds (OFF, 0-254) X - XON/XOFF Protocol Status (ON/OFF)
C05Dttt	Sets the SRL Port Transmit DELAY Time to ttt (0-255) in hundredths of seconds.
C05Ssss	Sets the SRL Port Receive STRING terminator to sss (1-7, 9-26, 28-47, 58-64 decimal).
C05TF	Turns the SRL Port DEFAULT Embedded Data TIME OUT function OFF - NO Time Out.
C05Tttt	Sets the SRL Port DEFAULT Embedded Data TIME OUT to ttt (0-254) seconds.
C05XF	Turns the SRL Port XON/XOFF function OFF (automatically selects the BUSY function for the OUTPUT/BUSY pin, if previously DISABLED).
C05XN	Turns the SRL Port XON/XOFF function ON (automatically DISABLEs the BUSY function for the OUTPUT/BUSY pin, if previously selected).

Note: Command codes can be strung together to configure multiple functions in a single command string.

Example: C05D5S13T2XN

D5 - Sets the SRL Port Transmit DELAY Time to 50 milliseconds,

S13 - Sets the Receive STRING terminator to 13 <CR>,

T2 - Sets the DEFAULT Embedded Data TIME OUT value to 2 seconds,

XN - Turns the XON/XOFF function ON.

Configure Output/Busy Pin - C06

Command C06 configures the Output/Busy Pin. The following is the list of valid command string formats for C06.

C06	The unit transmits the current OUTPUT/BUSY Pin configuration as shown. OUTPUT PIN CFG abcde TO-ttt uuu TO REM mm:ss.uu a - Enabled (E) or Disabled (D) with the Pin in the Output Mode, Busy (B) with the Pin in the Busy Mode b - Positive logic (+) or Negative logic (-) c - Output is ON (N) or OFF (F) d - Output Time Out Value is RUNning (R) or HALTed (H) e - Output Pin is LOCKed (L) or UNLOCKed (U) TO - Output Default Time Out value, ttt, and units, uuu, min./sec/10ms) TO REM - Time Out remaining in minutes (mm), seconds (ss), and hundredths of seconds (uu). Transmits "—TIMED OUT—" if the Time Out value has expired
C06B	Selects the BUSY Mode of the OUTPUT/BUSY Pin Function. Turns the SRL Port XON/XOFF Function OFF, if previously turned ON.
C06D	DISABLEs the OUTPUT/BUSY Pin Function. BUSY is selected if the SRL Port XON/XOFF Function is OFF.
C06E	ENABLES the OUTPUT Mode of the OUTPUT/BUSY Pin Function.
C06+	Selects POSITIVE logic for the OUTPUT/BUSY Pin - conducts when OFF.
C06-	Selects NEGATIVE logic for the OUTPUT/BUSY Pin - conducts when ON.
C06F	Turns the OUTPUT Pin OFF, if the OUTPUT function is ENABLED.
C06G	TOGGLES the state of the OUTPUT Pin, from ON to OFF, or OFF to ON, if the OUTPUT Function is ENABLED.
C06N	Turns the OUTPUT Pin ON, if the OUTPUT function is ENABLED.

C06R	RUNs the OUTPUT Time Out Timer - does not automatically turn the OUTPUT Pin ON.
C06H	HALTs the OUTPUT Time Out Timer - does not automatically turn the OUTPUT Pin OFF.
C06RD	RUNs the OUTPUT Time Out Timer and sets the Timer to the DEFAULT OUTPUT Time Out value.
C06RtttU	RUNs the OUTPUT Time Out Timer and sets the Timer to ttt (1-126) in HUNDREDTHs of a second.
C06RtttM	RUNs the OUTPUT TIme Out Timer and sets the Timer to ttt (1-63) MINUTEs.
C06RtttS	RUNs the OUTPUT Time Out Timer and sets the Timer to ttt (1-63) SECONDs.
C06TtttU	Sets the OUTPUT TIME OUT default value to ttt (1-126) in HUNDREDTHs of a second.
C06TtttM	Sets the OUTPUT TIME OUT default value to ttt (1-63) MINUTEs.
C06TtttS	Sets the OUTPUT TIME OUT default value to ttt (1-63) SECONDs.
C06L	LOCKs the OUTPUT/BUSY Pin configuration from further changes.
C06U	UNLOCKs the OUTPUT/BUSY Pin configuration.

Note: Command codes can be strung together to configure multiple settings in a single command string.

Example: C06UE-R10MNL

U - UNLOCKs the OUTPUT/BUSY Pin configuration,
E - ENABLEs the OUTPUT Pin function,
- - Sets the OUTPUT Pin logic level to NEGATIVE,
R10M - RUNs the OUTPUT Time Out Timer and sets the Timer to 10 MINUTEs.
N - Turns the OUTPUT Pin ON,
L - LOCKs the OUTPUT/BUSY Pin configuration.

Configure Elapsed Timers (ELP) - C07

Command C07 configures the ELAPSED (ELP) Timer function. The following is the list of valid command string formats for the C07 Command.

C07 or T	The unit transmits the current configurations of all 16 ELP Timers as shown. ELP TIMER tt CFG abcde M-mmm T-n/f TMR hihl:mm:ss.uu TRG hihl:mm:ss.uu tt - ELP Timer number (0-15) a - ELP Timer is ON (N) or OFF (F) b - ELP Timer Counts UP (U) or DOWN (D) c - Trigger Status, RETRIGGER(R) or ONE-SHOT(O) d - ELP Timer Disposition on power-up, SAVE (S) or CLEAR (C) e - Indicates that ELP Timer has OVERFLOWed or underflowed (V), blank otherwise. M - Trigger Message number, mmm (0-255) T - Trigger Status ON (N) or OFF (F) TMR & TRG - hihl:mm:ss.uu - ELP Timer or ELP Trigger value in hundreds and thousands (hi) of hours, ones and tens (hl) of hours, minutes (mm), seconds (ss), and hundredths of seconds (uu).

Note: The ESCAPE key <ESC> and the RETURN key PAUSE <RET> and RESUME the transmission of a list of ELP Timer Entries. Hitting <ESC> PAUSEs the transmission. Hitting <RET> RESUMEs the transmission. Hitting <ESC> while the list is PAUSEd cancels the remainder of the transmission.

***Warning:** The ADI/MDI cannot process any message requests, TIME OUTs, or cancellations while the transmission is PAUSEd.*

Note: Any command string including the "T" Command code can be issued without the Command number "C07".

C07F	Transmits a list of all OFF ELP Timers.
C07N	Transmits a list of all ON ELP Timers.
C07D	Transmits a list of all DOWN ELP Timers.
C07U	Transmits a list of all UP ELP Timers.
C07O	Transmits a list of all ONE-SHOT Triggered ELP Timers.
C07R	Transmits a list of all RE-TRIGGERED ELP Timers.
C07C	Transmits a list of all ELP Timers set for CLEAR on Unit Reset.
C07S	Transmits a list of all ELP Timers set for SAVE on Unit Reset.
C07V	Transmits a list of all OVERFLOWed ELP Timers.
C07TF	Transmits a list of all ELP Timers with TRIGGERs turned OFF.
C07TN	Transmits a list of all ELP Timers with TRIGGERs turned ON.
C07Mmmm	Transmits a list of all ELP Timers with Trigger MESSAGE mmm (0-255)
C07Ttt	Transmits the current configuration for ELP TIMER tt (0-15).
C07TttF	Turns ELP TIMER tt (0-15) OFF - Timer halts.
C07TttG	TOGGLEs ELP TIMER tt's (0-15) ON/OFF status.
C07TttN	Turns ELP TIMER tt (0-15) ON - Timer runs.
C07TttD	Sets ELP TIMER tt (0-15) to count DOWN.
C07TttU	Sets ELP TIMER tt (0-15) to count UP.
C07TttC	Sets ELP TIMER tt (0-15) for CLEAR on Unit Reset.
C07TttS	Sets ELP TIMER tt (0-15) for SAVE on Unit Reset.
C07TttV	Clears ELP TIMER tt's (0-15) OVERFLOW flag.
C07TttP0	PRESETs ELP TIMER tt (0-15) to 0.
C07TttPhihlmmssuu	PRESETs ELP TIMER tt (0-15) to hihl - HOURS (0000-9999) mm-MINUTES (00-59) ss - SECONDS (00-59) uu - HUNDREDTHS (00-99) *Note: Leading zeroes are required. Any zeroes following the last non-zero digit can be excluded. All excluded digits are automatically set to zero.*

C07TttTF	Turns ELP TIMER tt’s (0-15) TRIGGER OFF.
C07TttTN	Turns ELP TIMER tt’s (0-15) TRIGGER ON.
C07TttO	Sets ELP TIMER tt’s (0-15) Trigger for ONE-SHOT operation.
C07TttR	Sets ELP TIMER tt’s (0-15) Trigger for RE-TRIGGER operation.
C07TttMmmm	Sets ELP TIMER tt’s (0-15) Trigger MESSAGE to mmm (0-255).
C07TttT0	Sets ELP TIMER tt’s (0-15) TRIGGER value to 0.
C07TttT0F	Sets ELP TIMER tt’s (0-15) TRIGGER value to 0 and turn TRIGGER OFF.
C07TttT0N	Sets ELP TIMER tt’s (0-15) TRIGGER value to 0 and turn TRIGGER ON.
C07TttThihlmmssuu	Sets ELP TIMER tt’s (0-15) TRIGGER value to hihl - HOURS (0000-9999) mm - MINUTES (00-59) ss - SECONDS (00-59) uu - HUNDREDTHS (00-99) *Note: Leading zeroes are required. Any zeroes following the last non-zero digit can be excluded. All excluded fields are automatically set to 0*
C07TttThihlmmssuuF	Sets ELP TIMER tt’s (0-15) TRIGGER value and turn TRIGGER OFF.
C07TttThihlmmssuuN	Sets ELP TIMER tt’s (0-15) TRIGGER value and turn TRIGGER ON.

Note: Command codes can be strung together to configure multiple functions in a single command string. Multiple ELP Timers can be configured with the ‘/’ delimiter.

Example: C07T0SP0UNT0001NM20R/T1N/T2F

T0	- Configures TIMER 0,
S	- Sets Timer 0 for SAVE on Unit Reset,
P0	- PRESETs Timer 0 value to 0,
U	- Sets Timer 0 Direction to UP,
N	- Turns ON Timer 0
T0001N	- Sets Timer 0’s TRIGGER value to 1 hour and turns ON the Trigger function.
M20	- Sets the Trigger MESSAGE for Message 20,
R	- Sets the Trigger Mode for RE-TRIGGERING.
/	- Indicates new Timer
T1	- Configures TIMER 1,
N	- Turns ON Timer 1,
/	- Indicates new Timer,
T2	- Configure TIMER 2,
F	- Turns OFF Timer 2.

Configure Unit Display - C08

Command C08 configures the Unit Display parameters. The following is the list of valid command string formats for C08.

C08	The unit transmits the current Unit Display configuration settings as shown. DISPLAY CFG INT-ii BLINK BLOCK CHAR TOP aaa0ms bbbuuu ccc0ms BTM ddd0ms eeeuuu fff0ms INT Backlight INTENSITY, ii (0-16) aaa - Top Line BLINK time in hundredths of seconds (1-127) ddd - Bottom Line BLINK time in hundredths of seconds (1-127) bbb - Top line BLOCK SCROLL time (1-127) in uuu (0ms/sec) eee - Bottom Line BLOCK SCROLL time (1-127) in uuu (0ms/sec) ccc - Top Line CHARACTER SCROLL time in hundredths of seconds (1-127) fff - Bottom Line CHARACTER SCROLL time in hundredths of seconds (1-127)
C08Iii	Sets the Display Backlight INTENSITY to ii (0-16) in 16ths. 0 is OFF, 16 is full ON. *Note: For the TRI-COLOR unit. Sets the Backlight Color to ii (0-16). 0 is Full Red. 16 is Full Yellow-Green. 1-15 ranges from Red to Orange to Yellow.*
C08TBttt	Sets the TOP BLINK time to ttt (1-127) in hundredths of seconds.
C08TBSttt	Sets the TOP BLOCK SCROLL time to ttt (1-127) in seconds.
C08TBStttS	Alternate form of above.
C08TBStttU	Sets the TOP BLOCK SCROLL time to ttt (1-127) in hundredths of seconds.
C08TCSttt	Sets the TOP CHAR SCROLL time to ttt (1-127) in hundredths of seconds.
C08BBttt	Sets the BTM BLINK time to ttt (1-127) in hundredths of a second.
C08BBSttt	Sets the BTM BLOCK SCROLL time to ttt (1-127) in seconds.
C08BBStttS	Alternate form of above.
C08BBStttU	Sets the BTM BLOCK SCROLL time to ttt (1-127) in hundredths of seconds.
C08BCSttt	Sets the BTM CHAR SCROLL time to ttt (1-127) in hundredths of seconds.

Note: Command codes can be strung together to configure multiple functions in a single command string.

Example: C08I8TB10BB20TBS1BCS10

I8	- Sets the Display Backlight INTENSITY to 8/16ths,
TB10	- Sets the TOP BLINK Time to 100 milliseconds, ON for 100 msec, OFF for 100 msec,
BB20	- Sets the BTM BLINK Time to 200 milliseconds, ON for 200 msec, OFF for 200 msec,
TBS1	- Sets the TOP BLOCK SCROLL Time to 1 second, Scrolls 1 BLOCK every second,
BCS10	- Sets the BTM CHAR SCROLL Time to 100 msec, Scrolls left 1 position every 100 msec.

Configure Embedded Data (ED) - C09

Command C09 provides access to the Embedded Data functions of the unit. You can retrieve any data item (Local or Index) with Command C09. You can also load index item values (but not local item values) with this Command. The C09 Command also provides Increment and Decrement functions and signed and unsigned NUMERIC comparisons, and variable length ALPHANUMERIC comparisons for Index Data items. The following is the list of valid command string formats for C09.

The following symbols and syntax are used throughout the C09 Command descriptions. See the corresponding C09 Command description for details on the structure, usage, and operation of the listed functions:

Iii:...	- Load Index Item ii with....
Iii=...	- IF Index Item ii IS EQUAL TO....
Iii>...	- IF Index Item ii IS GREATER THAN....
Iii<...	- IF Index Item ii IS LESS THAN....
Iii!=...	- IF Index Item ii IS NOT EQUAL TO....
Iii!>...	- IF Index Item ii IS NOT GREATER THAN....
Iii!<...	- IF Index Item ii IS NOT LESS THAN....
...:mmm	- THEN request Message mmm.
...:mmm;...	- THEN request Message mmm, ELSE....
...:mmm/...	- THEN request Message mmm, ELSE....
...:mm1;mm2	- THEN request Message mm1, ELSE request Message mm2.
dd...d	- A variable length NUMERIC or ALPHANUMERIC data string. The entire Command string must be less than 128 characters inlength, the data string can be any length that doesn't make theCommand string containing it too long.
sdd...d	- A variable length NUMERIC data string. The 's' indicates a NUMERIC SIGN character, either '+' or '-'. The '+' sign is always optional.

A NUMERIC data string can consist only of a NUMERIC SIGN character, optional BLANK characters preceding and/or following the NUMERIC SIGN character, and the digit characters '0' to '9'. The NUMERIC SIGN character may be positioned immediately adjacent to the digit characters, or may be separated from the digits by one or more BLANKs.

An ALPHANUMERIC data string can consist of any combination of valid Message Text characters.

C09I or I	The unit transmits all Indexed Embedded Data Items that have Stored Data as shown. NDX xx <ddd...dd> xx - Index number (0-95) ddd...dd - Data string for Index xx *Note: Transmits a line for each programmed Index.*
	Note: The ESCAPE key <ESC> and the RETURN key <RET> PAUSE and RESUME transmission of a list of EMBEDDED DATA Items. Hitting <ESC> PAUSEs the transmission. Hitting <RET> RESUMEs the transmission. Hitting <ESC>while the list is PAUSEd cancels the remainder of the transmission. ***Warning:*** *The ADI/MDI cannot process any message requests, TIME OUTs, or cancellations while the transmission is PAUSEd.* *Note: Any command string including the "I" Command code can be issued without the Command number "C09".*
C09Iii	Transmits the contents of INDEXed ED Item ii (0-95).
C09Iii:dd...d	Loads the ED INDEX Item ii (0-95) with dd...d.
C09IiiDEL	DELETEs the contents of ED INDEX Item ii (0-95).
C09DELALL	DELETEs ALL ED INDEX Items.

Configure Embedded Data (ED) - C09 (Cont'd)

C09S	Transmits the Current ED STORAGE Area statistics. ED STORAGE AREA MAXIMUM: 03636 CURRENT: 00768 PEAK: 02413
C09SR	RESETs the ED STORAGE Area statistics.
C09D	Transmits all Local ED Items for DISPLAYed Message. QUEUE qq MSS mmm-ii ddd..dd qq - Queue Entry number (0-31) mmm - Message number (0-255) ii - Local Item number (1-15) ddd..dd - Data string *Note: Transmits a line for each Local Item in the Message.*
C09DIii	Transmits Local ED ITEM ii (1-15) for DISPLAYed Message
C09Mmmm	Transmits all Local ED Items of all Queue Entries for MESSAGE mmm (0-255).
C09MmmmIii	Transmits Local ED Item ii (1-15) of all Queue Entries for MESSAGE mmm (0-255).
C09Qqq	Transmits all Local ED Items of QUEUE Entry qq (0-31).
C09QIii	Transmits Local ED Items ii (1-15) for all QUEUE Entries.
C09QqqIii	Transmits Local ED Item ii (1-15) of QUEUE Entry qq (0-31).

The following INCREMENT and DECREMENT functions operate on the right-most NUMERIC data field contained in an Index. You must ensure that the Index value you want to INCREMENT or DECREMENT contains at least one NUMERIC field.

For a NUMERIC field that contains a DECIMAL POINT, both the INCREMENT and DECREMENT functions ignore the DECIMAL POINT and operate on the right most digits of the field, i.e. 100.23 + 1 = 100.24, and 100.23 - 24 = 099.99.

The ADI/MDI handles NUMERIC field OVERFLOWs and UNDERFLOWs resulting from INCREMENTing or DECREMENTing an Index based on the following criteria:

1. If the NUMERIC field operated on is unsigned (a '+' or '-' sign DOES NOT precede the field), then the unit treats the field as a "roll-over" counter, i.e. 999 + 1 = 000 and 000 - 1 = 999.
2. If the NUMERIC field is signed (a '+' or '-' sign DOES precede the field), then the unit processes the field using signed arithmetic, i.e. +000 - 1 = -001 and -000 + 1 = 001.
3. Both the sign and the number of digits the NUMERIC field contains are always preserved, i.e. 999 + 1 = 000 and -999 - 1 = -000. Therefore, you must initialize any Index value you plan on INCREMENTing or DECREMENTing with the maximum number of digits it could contain.

For instance, if you plan on using an Index as a counter or an accumulator, and the maximum value the Index can accumulate is 10000, then you must initialize the Index with a five digit NUMERIC field, i.e. "00000".

C09Iii+	INCREMENTs ED INDEX Item ii (0-95) by 1.
C09Iii+ddd	INCREMENTs ED INDEX Item ii (0-95) by ddd (0-255).
C09Iii-	DECREMENTs ED INDEX Item ii (0-95) by 1.
C09Iii-ddd	DECREMENTs ED INDEX Item ii (0-95) by ddd (0-255).

The following Command strings perform NUMERIC comparisons on Index values. Since the unit ignores leading zeroes when performing NUMERIC comparisons, two NUMERIC fields that you wish to compare need not contain the same number of digits. However, for the unit to properly perform a NUMERIC comparison on two fields that contain DECIMAL POINTS, both the INDEX value and the NUMERIC data must have the DECIMAL POINT located at the same position in the field, i.e. both fields being compared must have the same number of digits to the right of the DECIMAL POINT.

C09Iii>sdd...d:mmm	Performs a SIGNED NUMERIC comparison, IF ED INDEX Item ii (0-95) is GREATER THAN sdd...d, THEN request Message mmm (0-255).

C09Iii>sdd...d:mm1;mm2	Performs a SIGNED NUMERIC comparison,
	IF ED INDEX Item ii (0-95) is GREATER THAN sdd...d, THEN request Message mm1 (0-255),
	ELSE request Message mm2 (0-255).

C09Iii!>sdd...d:mmm	Performs a SIGNED NUMERIC comparison, IF ED INDEX Item ii (0-95) is NOT GREATER THAN sdd...d, THEN request Message mmm (0-255).
C09Iii!>sdd...d:mm1;mm2	Performs a SIGNED NUMERIC comparison, IF ED INDEX Item ii (0-95) is NOT GREATER THAN sdd...d, THEN request Message mm1 (0-255), ELSE request Message mm2 (0-255).

Example: C09I95>0:100IF
ED INDEX Item 95 GREATER THAN 0,
THEN request Message 100.

Example: C09I1>123:2;3
IF ED INDEX Item 1 GREATER THAN +123,
THEN request Message 2,
ELSE request Message 3.

Examples: The ADI/MDI considers the NUMERIC fields in each of the following groups as equivalent during NUMERIC comparisons:

ALL = "123"	ALL = "12.3"	ALL = "-123.45"
"123"	"12.3"	
"+123"	"+12.3"	"-123.45"
"+ 123"	"+ 12.3"	"- 123.45"
"00123"	"012.3"	
"+00123"	"+012.3"	"-0123.45"
"+ 00123"	"+ 012.3"	"- 0123.45"

However, the ADI/MDI does not consider "123", "123.", and "123.0" to be equivalent.

C09Iii<sdd...d:mmm	Performs a SIGNED NUMERIC comparison,
	IF ED INDEX Item ii (0-95) is LESS THAN sdd...d,
	THEN request Message mmm (0-255)

C09Iii<sdd...d:mm1;mm2	Performs a SIGNED NUMERIC comparison,
	IF ED INDEX Item ii (0-95) is LESS THAN sdd...d,
	THEN request Message mm1 (0-255),
	ELSE request Message mm2 (0-255).

C09Iii!<sdd...d:mmm	Performs a SIGNED NUMERIC comparison, IF ED INDEX Item ii (0-95) is NOT LESS THAN sdd...d,

C09Iii!<sdd...d:mm1;mm2	Performs a SIGNED NUMERIC comparison,
	THEN request Message mmm (0-255).
	IF ED INDEX Item ii (0-95) is NOT LESS THAN sdd...d,
	THEN request Message mm1 (0-255),
	ELSE request Message mm2 (0-255).

Example: C09I5<0:99
IF ED INDEX Item 5 LESS THAN 0,
THEN request Message 99

Example: C09I5!<-1:255;0
IF ED INDEX Item 5 NOT LESS THAN -1,
THEN request Message 255,
ELSE request Message 0.

Configure Embedded Data (ED) - C09 (Cont'd)

C09Iii=sdd...d:mmm	Performs a SIGNED NUMERIC comparison,IF ED INDEX Item ii (0-95) is EQUAL TO sdd...d, THEN request Message mmm (0-255).
C09Iii=sdd...d:mm1;mm2	Performs a SIGNED NUMERIC comparison, IF ED INDEX Item ii (0-95) is EQUAL TO sdd...d,THEN request Message mm1 (0-255), ELSE request Message mm2 (0-255).
C09Iii!=sdd...d:mmm	Performs a SIGNED NUMERIC comparison, IF ED INDEX Item ii (0-95) is NOT EQUAL TO sdd...d, THEN request Message mmm (0-255).
C09Iii!=sdd...d:mm1;mm2	Performs a SIGNED NUMERIC comparison, IF ED INDEX item ii (0-95) is NOT EQUAL TO sdd...d, THEN request Message mm1 (0-255), ELSE request Message mm2 (0-255).

Example: C09I2=100:100
IF ED INDEX Item 2 EQUAL 100,
THEN request Message 100.
Example: C09I5!=0:1;0
IF ED INDEX Item 5 NOT EQUAL TO 0,
THEN request Message 1,
ELSE request Message 0.	

The following Command strings perform alphanumeric comparisons on Index values. “Wild Card” and “Don’t Care” conditions can be implemented for all alphanumeric comparisons. The Wild Card character, ‘_’, indicates that a particular character position should be skipped during a comparison. The Don’t Care character, ‘?’, indicates that the comparison should only be performed on the characters preceding the ‘?’. The remainder of the character string should be ignored.

Example:
```
     CI091="___CNTA?":1;255
Index 01 =" 3 CNTA 123456"
```

This Command requests Message 1 as long as the 4th through 7th character positions of Index 1 contain the RLC mnemonic “CNTA”. The first three character positions are ignored due to the wildcard character and the remainder of the character string is accepted due to the DON’T CARE character. Otherwise, Message 255 is requested.

Note: Multiple conditional tests can be strung together for a single ED Index in one command string. Multiple ED INDEX Items can be configured with the ‘/’ delimiter.

Example: C09I10+>300:13;>200:12;>100:11;10
```
     I10+  - INCREMENT ED INDEX Item 10,
  >300:13  - IF INDEX 10 > 300, request Message 13,
 ;>200:12  - ELSE IF INDEX 10 > 200, request Message 12,
 ;>100;11  - ELSE IF INDEX 10 > 100, request Message 11,
      ;10  - ELSE requests Message 10.
```

Example: C09I1:+123/I2:abcd/I3+
```
  I1:+123  - Load ED INDEX Item 1 with “+123”,
        /  - New ED INDEX Item,
  I2:abcd  - Load ED INDEX Item 2 with “abcd”,
        /  - New ED INDEX Item,
      I3+  - INCREMENT ED INDEX Item 3 by 1.
```

C09Iii=“dd...d”:mmm	Performs an ALPHANUMERIC comparison,
	IF ED INDEX Item ii (0-95) is EQUAL TO “dd...d”,
	THEN request Message mmm (0-255).

C09Iii=“dd...d”:mm1;mm2	Performs an ALPHANUMERIC comparison,
	IF ED INDEX Item ii (0-95) is EQUAL TO “dd...d”,
	THEN request Message mm1 (0-255),
	ELSE request Message mm2 (0-255).

C09Iii!=“dd...d”:mmm	Performs an ALPHANUMERIC comparison,
	IF ED INDEX Item ii (0-95) is NOT EQUAL TO “dd...d”,
	THEN request Message mmm (0-255).

C09Iii!=“dd...d”:mm1;mm2	Performs an ALPHANUMERIC comparison,
	IF ED INDEX Item ii (0-95) is NOT EQUAL TO “dd...d”
	THEN request Message mm1 (0-255),
	ELSE request Message mm2 (0-255).

Example: C09I2=“STOP”:200
IF ED INDEX Item 2 EQUAL “STOP”,
THEN request Message 200.

Example: C09I5!=“START”:1;0
IF ED INDEX Item 5 NOT EQUAL “START”,
THEN request Message 1,
ELSE request Message 0.

Configure Embedded Data (ED) - C09 (Cont’d)

The following Command strings perform string length comparisons on Index values. The Index item is not compared for value, but simply for the number of characters it contains. These Commands are useful for determining the size of an Index item. For instance, most RLC serial units can be programmed to transmit both the mnemonic and numeric value for an item, or simply the numeric value. You can use these Commands to determine with which of the two transmit formats a unit will respond.

C09Iii<“dd...d”:mmm	Performs a size comparison,
	IF ED INDEX Item ii (0-95) is SHORTER THAN “dd...d”,
	THEN request Message mmm (0-255).

C09Iii<“dd...d”:mm1;mm2	Performs a size comparison,
	IF ED INDEX Item ii (0-95) is SHORTER THAN “dd...d”,
	THEN request Message mm1 (0-255),
	ELSE request Message mm2 (0-255).

C09Iii!<“dd...d”:mmm	Performs a size comparison,
	IF ED INDEX Item ii (0-95) is
	NOT SHORTER THAN “dd...d”,
	THEN request Message mmm (0-255).

C09Iii!<“dd...d”:mm1;mm2	Performs a size comparison,
	IF ED INDEX Item ii (0-95) is
	NOT SHORTER THAN “dd...d”,
	THEN request Message mm1 (0-255),
	ELSE request Message mm2 (0-255).

Example: C09I2<“ 1 CNTA 000000”:200
IF ED INDEX Item 2 SHORTER THAN “ 1 CNTA 000000”,
THEN request Message 200.

Example: C09I5!<“ 1 CNTA 000000”:1;0
IF ED INDEX Item 5 NOT SHORTER THAN “ 1 CNTA 000000”,
THEN request Message 1,
ELSE request Message 0.

Restore Default Unit Configuration - C17

The C17 Command Restores various settings within the ADI/MDI to their Default Settings. The following is a list of valid command string formats for the C17 Command.

C17ALL	Restores the Configuration File, Character File, and the Day/Month File to their Default Settings.
C17CFG	Restores the Configuration File to its Default Settings.
C17CHR	Restores the Character File to its Default Settings.
C17DM	Restores the Day/Month Name File to its Default Settings.

Transmit Unit Configuration To Slave Unit - C18

Command C18 transmits the Unit CHARACTER and DAY/MONTH Files in the ADI/MDI to the specified MDS Units. The MDS units must be in the RLC Slave Mode. The following is a list of valid command string formats for the C18 Command.

C18Ca1,a2,...	Transmits the CHARACTER File, to MDS Unit(s) with address number(s) a1,a2... (0-99).
C18Da1,a2,...	Transmits the DAY/MONTH File, to MDS Unit(s) with address number(s) a1,a2... (0-99).

Halt/Reset Unit - C19

Command C19 HALTs or RESETs the Unit. The following is a list of valid command string formats for the C19 Command.

C19HLT	HALTs the Unit. Issuing the HALT Command causes an ADI/MDI to save its current Queue contents, and the contents of the Embedded Data Storage area, shut off its display, and come to an orderly HALT. This command should be issued if you wish to turn the unit off and have it retain the Queue and Embedded Data information. The only exit from the HALTed state is to cycle power to the unit.
C19RST	RESETs the Unit. The Unit undergoes a POWER-ON RESET. Use this command if you need to have an ADI/MDI perform its POWER-UP sequence, which includes CLEARing any non-essential Elapsed Timers, deleting all non-Queued Messages, and requesting the RESET Message, if enabled.

Cancel Message/Queue Entry - C20

Command C20 cancels Messages on the Display and/or on the Queue. The following is a list of valid command string formats for the C20 Command.

C20	Cancels the DISPLAYed Message. If the Message is part of a chain, the ADI/MDI requests the next Message in the chain.
C20D	Alternate form of above.
C20C	Cancels the Displayed Message CHAIN. Cancels the Message on the Display, and any associated Message chain.
C20DC	Alternate form of above.
C20ALL	Cancels ALL Queue Entries. The Displayed Message and chain, and all Messages on the Queue are cancelled. The ADI/MDI requests the Default Message, if the function is Enabled.
C20Mmmm	Cancels all QUEUE Entries for MESSAGE mmm (0-255).
C20QMmmm	Alternate form of above.
C20Qqq	Cancels QUEUE Entry qq (0-31).
C20Q=qq	Alternate form of above.
C20Q>qq	Cancels all QUEUE Entries with Queue numbers GREATER THAN qq (0-30).
C20Q<qq	Cancels all QUEUE Entries with Queue numbers LESS THAN qq (1-31).
C20Pppp	Cancels all Queue Entries with PRIORITY ppp (1-255).
C20QPppp	Alternate form of above.
C20P=ppp	Alternate form of above.
C20QP=ppp	Alternate form of above.
C20P>ppp	Cancels all Queue Entries with PRIORITY GREATER THAN ppp (1-254).
C20QP>ppp	Alternate form of above.
C20P<ppp	Cancels all Queue Entries with PRIORITY LESS THAN ppp (2-255).
C20QP<ppp	Alternate form of above.

Request Message/Queue Entry - C21 (M mnemonic)

Command C21 issues Message requests from various sources: new Messages, temporary Messages, and existing Queue Entries. The following is a list of valid command string formats for the C21 Command.

Note: Any Command string including the ‘m’ command code can be issued without the Command number “C21”.

C21 or M	Issues a new request for the Displayed Message.
C21D	Alternate form of above.
C21Mmmm	Issues a request for MESSAGE mmm (0-255).
C21MImmm	Issues a IMMEDIATE request for MESSAGE mmm (0-255). Priority is ignored.
C21M:dd...d	Places a Temporary MESSAGE with text dd...d on the Display.
C21Qqq	Displays the message from QUEUE Entry qq (0-31).
C21Q=qq	Alternate form of above.
C21QMmmm	Displays the message from the first (lowest numbered) QUEUE Entry for MESSAGE mmm (0-255).
C21Pppp	Displays the message from the first (lowest numbered) QUEUE Entry with PRIORITY ppp (1-255).
C21QPppp	Alternate form of above.
C21P=ppp	Alternate form of above.
C21QP=ppp	Alternate form of above.
C21P>ppp	Displays the message from the first (lowest numbered) QUEUE Entry with PRIORITY GREATER THAN ppp (1-254).
C21QP>ppp	Alternate form of above.
C21P<ppp	Displays the message from the first (lowest numbered) QUEUE Entry with PRIORITY LESS THAN ppp (2-255).
C21QP<ppp	Alternate form of above.

Transmit Queue Entry Information - C22 (Q mnemonic)

Command C22 transmits the information for QUEUE Entries. The following is a list of valid command string formats for the C22 Command.

```
MESSAGE QUEUE CFG - n/f
DSP Q# PRI MSS CHN DATA T/O REM
ddd qq ppp mmm ccc ii   sss tttuuu
```

n/f - Indicates the Queue function status, ON (N) or OFF (F)

ddd - Indicates the Display status of the Message
- "--" - Message is currently on the Display
- "YES" - Message has been displayed
- "NO " - Message has not been displayed

qq - Queue number, or position. Starts at 0 and goes to 31

ppp - Priority number ranges from 1 to 255, 1 is the highest priority.
If the message is part of a chain, the priority shown is that of the message initiating the chain.

mmm - Message number for the Queue entry

ccc - Indicates the Chain status of the Message
- "YES" - Message is part of a chain
- "NO" - Message is not part of a chain

ii - Number of Local ED Items collected by the Message (1-15)

sss - Message ED Record size in bytes (2-254)
"NONE" is transmitted if the Message collected no Local ED Items

ttt -Indicates the remaining Message Time Out value, rounded to the nearest unit

uuu - Units for remaining Time Out value
- "min." - minutes
- "sec" - seconds
- "0ms" - milliseconds

"--> DISPLAY EMPTY" is transmitted if there is no Message on the Display.

"--> ERROR/TEMP MESSAGE" is transmitted if an Error Message or a Temporary Message is on the display.

" QUEUE EMPTY" is transmitted if there are no Entries on the Queue.

Note: A line is transmitted for each QUEUE Entry.

Note: The ESCAPE key <ESC> and the RETURN key <RET> PAUSE and RESUME transmission of a list of QUEUE Entries. Hitting <ESC> PAUSEs the transmission. Hitting <RET> RESUMEs the transmission. Hitting <ESC> while the list is PAUSEd cancels the remainder of the transmission.

***Warning:** The ADI/MDI cannot process any message requests, TIME OUTs, or cancellations while the transmission is PAUSEd.*

Note: Any command string including the Q command code can be issued without the command number "C22".

C22	Transmits the information for the Displayed Message.
C22D	Alternate form of above.
C22ED	Transmits the information for all Queue Entries with EMBEDDED DATA.
C22ND	Transmits the information for all Queue Entries with NO EMBEDDED DATA.
C22Q	Transmits the information for all QUEUE Entries.
C22Qqq	Transmits the information for QUEUE Entry qq (0-31).
C22Q=qq	Alternate form of above.
C22Q>qq	Transmits the information for all QUEUE Entries GREATER THAN qq (0-30).
C22Q<qq	Transmits the information for all QUEUE Entries LESS THAN qq (1-31).
C22Mmmm	Transmits the information for all Queue Entries for MESSAGE mmm (0-255)
C22QMmmm	Alternate form of above.
C22Pppp	Transmits the information for all Queue Entries with PRIORITY ppp (1-255).
C22QPppp	Alternate form of above.
C22P=ppp	Alternate form of above.
C22QP=ppp	Alternate form of above.
C22P>ppp	Transmits the information for all Queue Entries with PRIORITY GREATER THAN ppp (1-254).
C22QP>ppp	Alternate form of above.
C22P<ppp	Transmits the information for all Queue Entries with PRIORITY LESS THAN ppp (2-255).
C22QP<ppp	Alternate form of above.

Transmit Text For Message/Queue Entry - C23

Command C23 transmits the Message Text for requested Message and Queue Entries. The following is a list of valid command string formats for the C23 Command.

C23	Transmits the DISPLAYed message text.
C23D	Alternate form of above.
C23Mmmm	Transmits the message text for the first (lowest numbered) Queue Entry for MESSAGE mmm (0-255).
C23QMmmm	Alternate form of above.
C23Qqq	Transmits the message text of QUEUE Entry qq (0-31).
C23Q=qq	Alternate form of above.
C23Pppp	Transmits the message text of the first (lowest numbered) Queue Entry with PRIORITY ppp (1-255).
C23QPppp	Alternate form of above.
C23P=ppp	Alternate form of above.
C23QP=ppp	Alternate form of above.
C23P>ppp	Transmits the message text of the first (lowest numbered) Queue Entry with PRIORITY GREATER THAN ppp (1-255).
C23QP>ppp	Alternate form of above.
C23P<ppp	Transmits the message text of the first (lowest numbered) Queue Entry with PRIORITY LESS THAN ppp (2-255). Included for consistency only, will always transmit QUEUE Entry 0.
C23QP<ppp	Alternate form of above.

Echo Character String - C24

Command C24 transmits (echoes) the received character string. Used to communicate with units downstream of the receiving unit. The following is the valid command string format for the C24 Command.

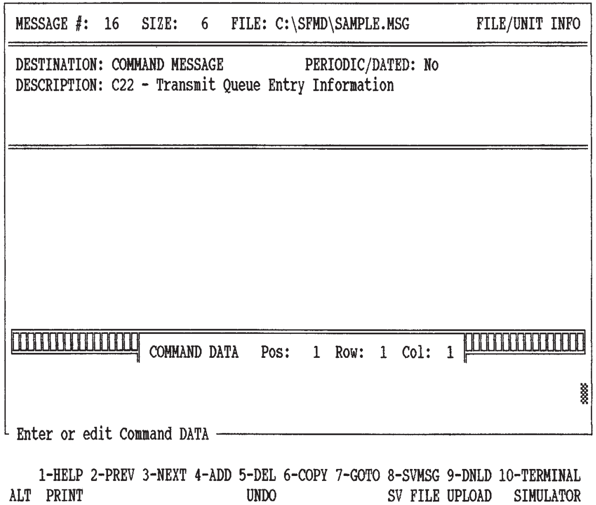
C24:dd...d	Transmits the character string dd...d (echoes string).

Command Messages

Every Command can be programmed as a Command Message. A Command Message is a Message Record that contains a programmed Command data string, rather than Message text. When the Command Message is requested, the ADI/MDI executes the Command string specified in the Message record, the same as if it had received the Command string over the Serial Port. The Command Message feature serves two primary functions:

1. Some Embedded Data and Message Queue features cannot be specified in a normal Message Record, and can only be accessed through Command Messages.
2. Unit configuration settings can be changed "on-line" to adapt to changing system conditions and parameters.

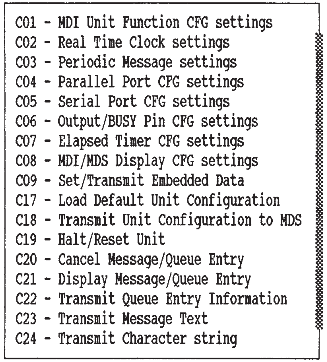
Figure 35, Command Message Screen appears as shown, when the Command Message destination is selected.



```
MESSAGE #:  16   SIZE:   6   FILE: C:\SFMD\SAMPLE.MSG        FILE/UNIT INFO

DESTINATION: COMMAND MESSAGE          PERIODIC/DATED: No
DESCRIPTION: C22 - Transmit Queue Entry Information

               COMMAND DATA   Pos:   1  Row:  1  Col:  1

Enter or edit Command DATA

   1-HELP 2-PREV 3-NEXT 4-ADD 5-DEL 6-COPY 7-GOTO 8-SVMSG 9-DNLD 10-TERMINAL
ALT  PRINT                     UNDO             SV FILE UPLOAD   SIMULATOR
```

Figure 35, Command Message Screen

You must select a Command number and type from the Description sub-menu.



```
C01 - MDI Unit Function CFG settings
C02 - Real Time Clock settings
C03 - Periodic Message settings
C04 - Parallel Port CFG settings
C05 - Serial Port CFG settings
C06 - Output/BUSY Pin CFG settings
C07 - Elapsed Timer CFG settings
C08 - MDI/MDS Display CFG settings
C09 - Set/Transmit Embedded Data
C17 - Load Default Unit Configuration
C18 - Transmit Unit Configuration to MDS
C19 - Halt/Reset Unit
C20 - Cancel Message/Queue Entry
C21 - Display Message/Queue Entry
C22 - Transmit Queue Entry Information
C23 - Transmit Message Text
C24 - Transmit Character string
```

The COMMAND DATA area of the SFMD software is where you enter the desired data string for the command, and it functions identically to the Message Text area. The Command Description supplies the Command number, which SHOULD NOT be entered in this area.

Once you select the Command number, Command specific HELP is available by hitting the F1-HELP key while in the DESCRIPTION or COMMAND DATA fields. The HELP screen(s) for the selected command displays the Command syntax and available command data functions and strings.

Example: You want to change a unit's BOTTOM BLINK time to 250 milliseconds and the Display Intensity to 16 with a Command Message. Select Command CO8 from the Command Description list and enter the data string "BB25I16" in the Command Data area.

Command Messages (Cont'd)

Link Function

You can LINK a Command Message to any other Message by placing a forward-slash, '/',and the desired Message number at the end of the data string. The data string "BB25I16/103" for Command C08 causes the unit to set the BOTTOM BLINK time to 250 milliseconds, the Display Intensity to 16, and then request Message 103.

Since a Command Message cannot specify a Chained Message list itself, the LINK feature is useful for continuing a chain from a Command Message or LINKING multiple command messages.

Example: You want to vary the BLINK time and Display Intensity for a given Message based on the current value of an Index. You program four Messages to perform this task. Message 100 is a Command Message for C09 containing the data string "I10>100:101/I10>50:102;103". This Message performs the Index comparison. Messages 101 and 102 are both Command Messages for C08. The data strings for Messages 101 and 102 are "BB25I16/103" and "BB50I8/103" respectively. These Messages adjust the BLINK time and Display Intensity. Message 103 is the Message you are placing on the Display after adjusting the BLINK time and Display Intensity.

Message 100 determines which Message to request based on the current value of Index 10. Message 101 is requested if the value of Index 10 is GREATER THAN 100, Message 102 is requested if the value is GREATER THAN 50, but LESS THAN 100, and Message 103 is requested if the Index value is LESS THAN OR EQUAL TO 50.

Messages 101 and 102 adjust the BOTTOM BLINK time and the Display Intensity accordingly and then request Message 103. Every time you request Message 100, the ADI/MDI performs this process, just as if the Messages were chained together.

Note: The LINK feature is only available for Commands C01 through C09 and Command C24.

Index Substitution

Just as you can insert the current value of an Index in a Message's text, you can also insert that same value in a Command Message's data string. This is referred to as Index substitution. (See *MESSAGE TEXT*, page 32, for details on inserting an Indexed ED Item in the Command Data area). When the unit encounters an Index number while processing a Command Message, it inserts the current value of that Index in the data string at the location of the Index number. In essence, you can include variables in Command data strings. This feature can be used in any Command Message, but is most useful in collecting and storing embedded data in sequential Index locations.

Example: The unit collects a temperature reading for a process once an hour and you want to see how these readings change over a 48 hour period. You can program the unit to collect this data and store it in sequential Index locations for later retrieval and review.

Index 0 serves as the counter for the process. Indexes 1 to 48 hold the hourly readings. Message 100 is requested by a Periodic Entry every hour, collects the temperature reading, stores it in Index 49, and chains to Message 101, which performs the actual Index functions.

Message 101 is a Command Message for C09 and contains the data string "I0>48:102/I   i00:   i49/I0+". Message 102 is a Command Message for C03 and contains the data string "P0F", which turns OFF the Periodic Entry that requests Message 100 every hour, thus halting the data collection.

Before executing the data string for Message 101, the unit inserts the current values for the specified Indexes in the string. Index 00 points to the next available Index location. Index 49 contains the latest temperature reading collected by Message 100. Therefore, if Index 00 contains "23" and Index 49 contains "134.2 F", the data string becomes "I0>48:102/I23:134.2 F/I0+" after the unit performs the necessary Index substitutions. The unit then executes Command C09 for this string.

The resulting data string reads as follows:

```
IF Index 00 > 48                  If all 48 Indexes have been filled, request
  THEN request Message 102,          Message 102, which halts the process.

OTHERWISE                         Otherwise,
  LOAD Index 23 with "134.2 F".     Save the new data in Index 23,
  INCREMENT Index 00.               Increment Index 00 to 24 for the next
                                    data collection.
```

CHARACTER SET

The ADI/MDI contains 153 displayable characters, from 20h (32) to B8h (184). The 96 standard ASCII characters are included from 20h to 7Fh (127). The extended set of characters, from 80h (128) to B8h contain the European and special characters. The 96 ASCII characters can be entered in a Message simply by typing their respective key on the keyboard. An extended character can be entered by typing its decimal equivalent on the numeric key pad while holding down the ALT key.

The SFMD software allows you to customize the character set to the particular needs of your system. Individual characters can be edited in the Character Editing screen. The customized Character File (".CHR" file extension) can then be downloaded to the unit over the serial port.

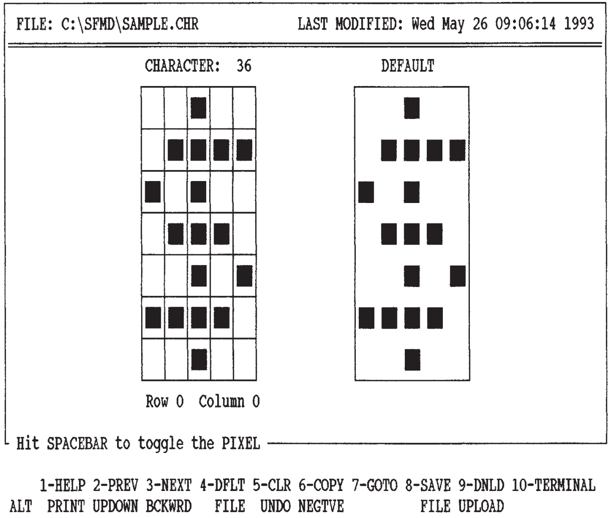
The Command string "C17CHR" causes the ADI/MDI to restore its factory default character set.

RLC Message Display Slave (MDS) units do not contain enough non-volatile memory to retain a custom character set on power-down. If the system you are developing contains one or more MDS units and requires a customized Character set, you can use the C18 Command to cause the ADI/MDI to transmit its programmed Character set to one or more MDS units on the loop. See *Command C18*, page 95, for details.

Customizing A Character Set

The character Editing of the SFMD screen allows you to edit any of the default characters or character files you have created. Each character consists of 35 dots, or pixels, in 7 rows of 5 pixels each. To create or edit a character, position the cursor to the desired pixel locations with either the cursor arrow keys or the mouse. Pressing the SPACE bar or clicking the mouse toggles the selected pixel position ON and OFF.

The factory default character always appears to the right of the character editing area, and you can recover it at any time by pressing the F4-DFLT key. In addition, Function keys are also available for clearing a character, for creating the character's negative image, and for turning the character backwards, and/or upside-down.



```
FILE: C:\SFMD\SAMPLE.CHR            LAST MODIFIED: Wed May 26 09:06:14 1993

                CHARACTER:  36                DEFAULT

                Row 0  Column 0

Hit SPACEBAR to toggle the PIXEL

   1-HELP 2-PREV 3-NEXT 4-DFLT 5-CLR 6-COPY 7-GOTO 8-SAVE 9-DNLD 10-TERMINAL
ALT  PRINT UPDOWN BCKWRD   FILE  UNDO NEGTVE         FILE UPLOAD
```

Default Character Set

ASCII TABLE OF STANDARD CHARACTERS

HEXA DECIMAL	DECIMAL	CHARACTER
20	32	
21	33	!
22	34	"
23	35	#
24	36	$
25	37	%
26	38	&
27	39	’
28	40	(
29	41	)
2A	42	*
2B	43	+
2C	44	,
2D	45	-
2E	46	.
2F	47	/
30	48	0
31	49	1
32	50	2
33	51	3
34	52	4
35	53	5
36	54	6
37	55	7

HEXA DECIMAL	DECIMAL	CHARACTER
38	56	8
39	57	9
3A	58	:
3B	59	;
3C	60	<
3D	61	=
3E	62	>
3F	63	?
40	64	@
41	65	A
42	66	B
43	67	C
44	68	D
45	69	E
46	70	F
47	71	G
48	72	H
49	73	I
4A	74	J
4B	75	K
4C	76	L
4D	77	M
4E	78	N
4F	79	O

ASCII TABLE OF STANDARD CHARACTERS

HEXA DECIMAL	DECIMAL	CHARACTER
50	80	P
51	81	Q
52	82	R
53	83	S
54	84	T
55	85	U
56	86	V
57	87	W
58	88	X
59	89	Y
5A	90	Z
5B	91	[
5C	92	\
5D	93	]
5E	94	^
5F	95	_
60	96	‘
61	97	a
62	98	b
63	99	c
64	100	d
65	101	e
66	102	f
67	103	g

HEXA DECIMAL	DECIMAL	CHARACTER
68	104	h
69	105	i
6A	106	j
6B	107	k
6C	108	l
6D	109	m
6E	110	n
6F	111	o
70	112	p
71	113	q
72	114	r
73	115	s
74	116	t
75	117	u
76	118	v
77	119	w
78	120	x
79	121	y
7A	122	z
7B	123	{
7C	124	|
7D	125	}
7E	126	~
7F	127	n

Default Character Set (Cont'd)

ASCII TABLE OF EXTENDED CHARACTERS

HEXA DECIMAL	DECIMAL	CHARACTER
80	128	Ç
81	129	ü
82	130	é
83	131	â
84	132	ä
85	133	à
86	134	å
87	135	ç
88	136	ê
89	137	ë
8A	138	è
8B	139	ï
8C	140	î
8D	141	ì
8E	142	Ä
8F	143	Å
90	144	É
91	145	æ
92	146	Æ
93	147	ô
94	148	ö
95	149	ò
96	150	û
97	151	ù
98	152	ÿ
99	153	Ö
9A	154	Ü
9B	155	¢
9C	156	£
9D	157	¥
9E	158	Á
9F	159	Í
A0	160	á
A1	161	í
A2	162	ó
A3	163	ú
A4	164	ñ
A5	165	Ñ
A6	166	Ó
A7	167	Ú
A8	168	¿
A9	169	È
AA	170	À
AB	171	ø
AC	172	Ø
AD	173	¡
AE	174	Ù
AF	175	ã
B0	176	õ
B1	177	‘
B2	178	´
B3	179	
B4	180	
B5	181	°
B6	182	
B7	183	
B8	184	

ERROR HANDLING

The ADI/MDI contains an Error detection function which generates Error Codes for the various Error conditions it encounters. An Error Code is processed according to the type of Handling method selected.

DISPLAY: The unit displays the Error Code as a blinking Temporary Message for 16 seconds. The priority of the Error Message is 0 and only another Error Message, or a Temporary Message can interrupt it. You can cancel any Error Message with the C20 Command. Error Messages are not placed on the Queue.

The Error Message appears on the display in the following format:

ERROR

exn

'e'	- Error Code	
x	- Error Type	
	'c'	- Command string
	'd'	- Data
	'f'	- Function
	'm'	- Message request
n	- Error Number, (0-9)	

TRANSMIT: The unit assembles and transmits the Error Code over the serial port. The Error Code is transmitted in the following format:

exn (see above for codes)

DISPLAY & TRANSMIT: The unit both displays and transmits the Error Code as described above.

IGNORE: The unit continues to detect Error conditions, but does not generate Error Codes for them.

You can set and verify the current Error Handling method with the C01 Command.

Note: Error Codes are always transmitted if the Serial Port XON/XOFF Function is ON, regardless of the Error Handling type selected.

Error Codes And Numbers

'c' - Command String Errors

ec0 - Invalid command (Command > 24)
ec1 - Unused command (Command 0 or 10-16)
ec2 - Unexpected or invalid command code
ec3 - Missing numeric command data
ec4 - Numeric command data out-of-range
ec5 - Numeric command data > 255
ec6 - Non-numeric character in numeric field
ec7 - Error in index substitution
ec8 - Command buffer overflow

'd' - Embedded Data (ED) Errors

ed0 - ED record, item, or index missing
ed1 - ED index has no numeric field
ed2 - ED index numeric field Overflow
ed3 - ED index numeric field Underflow
ed4 - ED index non-numeric character in numeric field
ed5 - ED item size overflow
ed6 - ED record size overflow
ed7 - ED memory overflow

'f' - Function Errors

ef0 - Output pin is LOCKed, cannot make changes
ef1 - Periodic entry is not programmed
ef2 - Periodic entry is disabled, cannot turn ON
ef3 - Invalid DIP switch setting

'm' - Message Request Errors

em0 - Requested Message not programmed
em1 - Message or Queue entry not on Queue
em2 - Default Message is not programmed
em3 - Reset Message is not programmed
em4 - Non-numeric digit in Parallel Port BCD message number
em5 - Message length mismatch
em6 - Message memory overflow

MESSAGE SIMULATOR FUNCTION

The SFMD software contains an extensive Message Simulator function which allows you to observe the operation of the message you are currently editing without constantly downloading the entire message file to an ADI/MDI. Pressing ALT-F10 simulator from the Message Editing screen invokes the Simulator, which appears in the lower half of the Message screen. If there is no currently open Configuration file, the Simulator asks you to open one first.

The Simulator then executes the current Message based on the parameters you have specified for that Message. The Message Text appears on the Simulator's display exactly as it would appear on an ADI/MDI of the appropriate configuration. The Simulator uses the current DISPLAY TYPE setting from the FILE/UNIT INFO sub-menu to assemble its Display. It can Simulate any of the available Display configurations, including the Tri-Color Display.

The Simulator executes the following Message features:

1. Processes the current Message according to the specified Message Destination and current Configuration file settings.
2. Configures and displays the Output Pin status, and runs the Output Time Out, as specified in the Message record.
3. Configures, runs, and displays the Elapsed Timers as specified in the Message record. The Trigger Message function is not simulated.
4. Prompts you to enter any Embedded Data the Message record specifies, and then formats, stores, and displays the data as specified for each data item in the Message record.
5. Assembles and displays all transmissions in the proper order and the appropriate formats.
6. Displays Message text using the current Configuration file settings for the Blink time and the Block and Character scroll times. The SFMD software must use the IBM® character set to display Messages, consequently some of the extended characters cannot be displayed as they appear on the unit. In addition, the Simulator cannot display custom character sets.
7. Runs the Message Time Out, if desired.
8. Automatically Simulates the Messages in a chain, if desired
9. Informs you of any Message execution errors, where appropriate.

Current and Elapsed Time fields are filled in with their appropriate values and Embedded Data item values are inserted where requested. Elapsed Timer configurations and values and Indexed Embedded Data item values are retained as long as you do not exit the SFMD software. You can, therefore, perform simulations on blocks of messages that collect and display data, and configure and display Elapsed Timers.

Even though the Simulator does not execute Command Messages, it does assemble Command strings with Index substitution and display the results for you. The Simulator does not scan the Command Message Data string for validity. The unit performs this function whenever you request a Command Message

Simulator Function Keys

F1-HELP - Displays HELP for the Simulator function.

F2-PAUSE - PAUSEs the Simulator. The Simulator Display, and all time values, are frozen.

F3-CONTINUE - RESUMEs the Simulator. Also, CONTINUEs to the next Embedded Data item, or to the next message in a chain.

F4-CANCEL MSG - CANCELs the current Message. Returns to the Message Editing screen if the Message is not chained, or if AUTO CHAIN is OFF.

F5-AUTO T/O - Toggles the automatic Message Time Out feature. If AUTO T/O is selected, the Simulator automatically cancels the Message when its Time Out expires. If AUTO T/O is OFF, the Message's Time Out does not run. Disable the AUTO T/O feature if you want to observe the operation of a Message without having it repeatedly time out on you.

F6-AUTO CHAIN - Toggles the automatic Message Chain feature. If AUTO CHAIN is selected, the Simulator automatically proceeds to the next Message in a chain when the current Message times out or is cancelled. If AUTO CHAIN is OFF, the subsequent Messages of a chain are not executed. Disable the AUTO CHAIN feature if you want to observe the operation of an individual Message in a chain without repeatedly executing the entire chain.

F10-EXIT - EXITs the Message Simulator, returning control to the Message Editing screen.

Simulator Embedded Data Collection

The Message Simulator performs embedded data collection of both Local and Indexed items for all four embedded data types. The individual Embedded Data Item configuration is displayed above the Simulator area while the Simulator is processing a particular item. The Simulator displays all of the actual transmissions the unit would issue while collecting the data item, including the XON/XOFF protocol ("<XON>" and "<XOFF>"), if enabled in the Configuration File and the Transmit Request string, if specified for the item being processed, including any Transmit Delays ("<TRX>") and Receive Buffer clearing operations ("<CLR>") that the string specifies.

The data for an Embedded Data Item is entered through the keyboard. The Simulator displays the raw Receive Buffer contents as you enter the characters. Characters entered for Serial Port data items are processed just as if the ADI/MDI was receiving them. Therefore, the ESCAPE key clears the Receive Buffer contents, and the BACKSPACE key deletes the last character in the buffer. Also, the Simulator accepts characters for a serial data item until the appropriate serial terminator is entered, as specified by that individual item. The Simulator does not run the Embedded Data Time Out.

While collecting a Parallel Port data item, the Simulator displays the various Parallel Port pin values. The Parallel Port operation conforms to the settings of the current Configuration file. In order to enter a Parallel Port data item, you must set the Parallel Port CTRL lines accordingly. The unit only interprets information on the port as data when the MESSAGE/DATA line is set to the DATA level. If you issue a STROBE with the MESSAGE/DATA line at the MESSAGE level, the unit aborts the Message. The Simulator responds identically.

The Parallel Port CTRL lines are set with the following key strokes:

ALT-D - Sets the MESSAGE/DATA line to the DATA level.
ALT-M - Sets the MESSAGE/DATA line to the MESSAGE level.
ALT-S - Issues a STROBE pulse, causing the Simulator to read the MESSAGE/DATA line and the DATA lines.

You set the Parallel Port DATA lines by entering the desired value in the appropriate format, as designated in the current Embedded Data Item configuration and the current Configuration file. For 8-bit ports, enter the HI nibble first, followed by the LO nibble, then issue a STROBE. For 4-bit ports, enter and STROBE the HI nibble first, and then enter and STROBE the LO nibble. The Simulator shows you the current CTRL and DATA Line settings, as well as the nibble and byte counts, as you enter the data. Once the Simulator has collected all of the data for an item, it formats the raw data in its Receive Buffer according to the format the item configuration specifies. The Simulator then displays the formatted result in the Receive Buffer field, as well as the size of the data item. Press ENTER or F3-CONTINUE to continue with the remainder of the Message. Execution proceeds to the next data item, or to the Message Display, if data collection is complete. You can CANCEL the simulation at any time by hitting F4-CANCEL MSG or F10-EXIT.

Message Display

Once the Simulator has collected all of the data for a Message, it then displays any remaining Message transmissions, including the XON/XOFF protocol, and the TRANSMIT or MDS UNIT transmission strings, if selected. The Simulator also displays the Message's Embedded Data record size and the current size of the Embedded Data Storage Area. If the Message is designated for DISPLAY, the Message text then appears in the Simulator Display area, just as the ADI/MDI would show it.

The Message text is displayed according to the SCROLL settings specified in the Message record, as well as the BLINK time and the BLOCK and CHARACTER scroll times contained in the current Configuration file. The Simulator converts the 17 different color settings for the Tri-color Display into three distinct colors. Colors 0-3 are shown as RED, Colors 4-12 are shown as ORANGE, and Colors 13-16 are shown as YELLOW-GREEN.

The Simulator also performs several forms of error checking. It informs you if the Transmit Buffer overflows, and that a transmission to an MDS Unit exceeds 127 characters in length. It tells you when a data item collects more than 127 bytes of data, and that a Message's Embedded Data record exceeds 254 bytes. In addition, the Simulator alerts you to Embedded Data Storage Area overflows, which result from collecting more data than the unit's memory is capable of holding for the current Message file.

TERMINAL EMULATOR FUNCTION

The SFMD software contains a Terminal Emulator which allows you to communicate with the ADI/MDI over the same serial connections that the SFMD software uses to upload from and download to the unit. Consequently, you can use the Terminal Emulator for requesting Messages, issuing commands, interrogating the unit, and most importantly, for testing, debugging, and trouble-shooting your particular application.

You can enter the Terminal Emulator from the Utilities pull-down menu, and from any of the three file editing screens. The current serial port settings are shown at the top of the Terminal Emulator screen, along with the currently specified unit address for uploading and downloading purposes. You can alter these settings from within the Terminal Emulator.

Function Keys

The active function keys are displayed at the bottom of the screen, along with their respective operations.

F1-HELP - Requests on-line HELP for the Terminal Emulator function.

F2-CFG TE - Activates the Terminal Emulator serial configuration fields. While configuring the serial settings,the Terminal Emulator ignores all serial transmissions. Press F2 to accept the new settings, or ESCAPE to keep the old settings. While you are configuring the serial settings, the CFG TE field BLINKs.

F3-RECORD - Activates the RECORD function of the Terminal Emulator and requests a RECORD file name. Enter the file name, including the path, if different than the current path. Hit ENTER to accept the file name, or ESCAPE to keep the existing file. The RECORD field BLINKs while the Terminal Emulator is RECORDing.

If the specified file does not exist, it is created. If the file already exists, the Terminal Emulator begins to RECORD at the end of the current file, consequently, the existing data is kept. If the current data is not desired, hit F4 to REWIND the file to the beginning.

F4-REWIND - REWINDs the current RECORD file to the beginning. All existing data in the file is lost.

F5-LOCK - LOCKs the keyboard from further entry. Press ALT-F5 to UNLOCK the keyboard. The LOCK field BLINKs while the keyboard is LOCKed.

F6-PRINT - PRINTs the current RECORD file to the computer's parallel port.

F10-EXIT - EXITs the Terminal Emulator. Closes any currently open RECORD file.

Record File: The name of the current RECORD file.

All characters received from the serial port are displayed in regular type, all of the characters you enter, and any functions you perform, are displayed in bold type.

The Terminal Emulator displays the characters it receives exactly as transmitted, with the following exceptions:

1. Automatically inserts a Line Feed (LF, 0Ah - 10) for any unmatched Carriage Return (CR, 0Dh - 13) it receives.
2. Automatically inserts a CR for any unmatched LF it receives.
3. Displays "<XON>" for the ASCII XON code (11h - 17) and "<XOFF>" for the ASCII XOFF code (13h - 19).
4. Excluding CR, LF, XOFF, and XON, the hexadecimal value of any character received under 20h (32) or over 7Fh (127) is displayed in the form "[xx]", where "xx" is the two digit hexadecimal code for the character.
5. The character 1Eh (30) is reserved as the Terminal Emulator Serial Command (SRL CMD) character. This character, combined with one of several letter codes, instructs the Terminal Emulator to perform specific functions. The SRL CMD codes can be embedded in Transmit Request strings and message text transmissions to remotely control the Emulator from an ADI/MDI. See *Terminal Emulator Serial Command Functions* later in this section for details.

Special Key Stroke Functions

Except for the previously mentioned function keys, and the entering of a RECORD file name, the Terminal Emulator transmits all the characters you type. In addition, the following special key stroke functions are available for your use.

CTRL-HOME - Transmits an ESCAPE (1Bh - 27) character, followed by the ADI/MDI universal Address code (0Eh - 14) and displays “[ADDR]”. These characters cause a unit to clear its receive buffer and then override its Unit Address and accept the transmission following the universal Address code. This function is useful if you have forgotten the unit address of an ADI.

CTRL-END - Transmits the ADI/MDI universal Serial Terminator (00h - 0) and displays “[TERM]”. Upon receiving this character, a unit begins processing the contents of its serial buffer, just as if it had received its programmed serial terminator. This function is useful if you have forgotten the serial terminator for an ADI/MDI.

Example: You can determine the Unit Address (ADDR) and Serial Terminator (ST) of any ADI/MDI by entering the following Command strings:

```
Operator:   <CTRL-HOME>C1<CTRL-END>
Response:    FUNCTIONS CFG
            ADDR-99  CLK-ON  DFLT-OFF 100  ERR-B
            PRDC-OFF QUE-OFF RST -OFF 101

Operator:   <CTRL-HOME>C5<CTRL-END>
Response:   SRL PORT CFG - 7,O,9600
            DL-0050ms ST-<13> TO-001s X-ON
```

BACKSPACE - Transmits a BACKSPACE (08h - 8) character, causing an ADI/MDI to back-up one character position in its Receive Buffer. The character is removed from the Terminal Emulator’s display, as well. The unit ignores BACKSPACEs while its Receive Buffer is empty. However, the Terminal Emulator continues to delete characters from the screen.

ESCAPE - Transmits an ESCAPE (1Bh - 27) character, causing an ADI/MDI to clear its Receive Buffer without processing its contents. The Terminal Emulator displays “[ESC]” for each stroke of the ESCAPE key.

If the unit receives the ESCAPE character while it is in the process of transmitting a Command response, it pauses the Transmit Buffer at the end of the current line. The unit resumes the transmission upon receiving a CR. If it receives another ESCAPE character while the Transmit Buffer is paused, the unit aborts the transmission.

RETURN - Transmits a Carriage Return (CR, 0Dh - 13). If the Transmit Buffer of an ADI/MDI is currently paused, the unit resumes transmission upon receiving the CR. Otherwise, the unit treats a carriage return as any other character. The Terminal Emulator advances the display one line and displays “[CR]” for every stroke of the ENTER key.

ALT-xxx - Transmits the character represented by xxx in decimal. Any character from 01h (1) to FFh (255) can be issued by holding down the ALT key while entering the decimal equivalent of the desired character on the numeric keypad. The character is transmitted when the ALT key is released.

Terminal Emulator Record Function

The Terminal Emulator includes a RECORD function, which allows you to RECORD all of the actions of a Terminal Emulator session to the hard or floppy disk of the PC on which you are running the SFMD software. You enter the RECORD function by pressing F3-RECORD. The Terminal Emulator asks you for a file name, which can include a directory path, if desired. After you have typed in the RECORD file name, hit ENTER for the Terminal Emulator to open the file and begin RECORDing, or ESCAPE to cancel the function.

If the file you have specified does not exist, the Terminal Emulator creates it for you. If the file already exists, the Terminal Emulator begins RECORDing from the current end of the file, thus preserving any previous data. If you no longer require this data, hit F4-REWIND to position the file at the beginning.

While in RECORD, the Terminal Emulator RECORDs everything that appears on the screen, exactly as you see it, except for the regular and bold type styles. The RECORD file is stored in an ASCII text format, which you can review, edit, and print with almost any word processor. In addition, you can PRINT the contents of the current file from the Terminal Emulator by pressing the F6-PRINT key.

The RECORD function is useful for testing and debugging an ADI/MDI application. Complex applications can be developed and verified through creative use of the Terminal Emulator and the RECORD function. In addition, you can program an ADI/MDI to remotely control the RECORD function. In this case a unit can cause the Terminal Emulator to selectively RECORD information the unit transmits to it, without any operator supervision or intervention.

Terminal Emulator Serial Command Function

A set of serial commands is available to allow the ADI/MDI to directly control the Terminal Emulator, without operator intervention. Most of the commands are directed towards controlling the RECORD feature of the Emulator. Consequently, a unit so programmed can transmit strings to the Terminal Emulator for selective recording of various messages, data values, etc.

With the TRANSMIT destination, the user can specify any transmission format necessary for saving the on-line output of an application to the hard or floppy disk of the computer running the Terminal Emulator.

Serial Command Codes can be issued to the Terminal Emulator in either UPPER or LOWER case. For LOWER case command codes, the Terminal Emulator echoes the code after successfully performing the requested command. The Terminal Emulator does not issue a response after processing an UPPER case command code.

Thus, messages can be programmed to turn the Terminal Emulator RECORD function ON and OFF with Embedded Data Item Transmit Request strings. The data Item can even be programmed to wait for the Emulator to respond, and can store and test that response to detect the presence or absence of the Terminal Emulator. The response mode of the serial commands accommodates variations in the disk access response time of the various PC configurations.

Serial Command Codes

‘▲c’	CLOSEs the current RECORD file.
‘▲e’	EJECTs the current page in an open RECORD file and clears the Terminal Emulator screen.
‘▲f’	Turns RECORDing OFF for the current file.
‘▲l’	LOCKs the keyboard.
‘▲n’	Turns RECORDing ON for the current file.
‘▲o<filename>’	Closes the current RECORD file and OPENs the specified file. Creates the specified file if it does not exist. Does not REWIND an existing file. Turns RECORDing ON for the newly opened file.
‘▲p’	PRINTs the current RECORD file.
‘▲r’	REWINDs the current RECORD file.
‘▲t’	Displays the System TIME. The TIME is inserted in the current RECORD file, if open.
‘▲u’	UNLOCKs the keyboard.

Note: The ‘▲‘ character is entered by pressing and holding the ALT key while typing 30 on the numeric keypad.

TROUBLESHOOTING GUIDE

The majority of problems can be traced to improper connections or incorrect set-up parameters. Be sure all connections are clean and tight and that the set-up parameters are correct. For further technical assistance, contact technical support at the numbers listed on the back cover of the instruction manual.

PROBLEMS	POSSIBLE CAUSE	REMEDIES
DARK DISPLAY (No Backlighting Apparent)	1. Power off. 2. Improperly wired 3. Backlighting intensity level set to 0. 4. Power in a Brown out condition. 5. Voltage selector switch in wrong position. 6. “CONTRAST” pot adjusted improperly. 7. No requested message.	1. Verify power. 2. Check wiring. 3. Verify level setting. 4. Verify power reading. 5. Check switch setting. 6. Adjust pot. 7. Check message file.
BLANK DISPLAY (Backlighting Apparent But No Discernible Character)	1. “CONTRAST” pot adjusted improperly. 2. No requested message.	1. Adjust Pot. 2. Check message file.
ADI/MDI IS NOT RESPONDING TO SFMD SOFTWARE	1. Serial set-up parameters do NOT match the ADI/MDI’s. 2. GCM232 Module in wrong serial port of computer. 3. GCM232 Module NOT powered. 4. GCM232 Module improperly connected to unit. 5. Wrong address(es). 6. Modem or mouse connected to Serial Port. 7. “20mA DIS.” DIP switch in wrong position (ADI only).	1. Verify set-up parameters (baud rate, parity, data bits). 2. Switch module to other port. 3. Verify power. 4 a. Check wiring. b. Pins 2 & 3 of GCM232 not connected properly to computer. 5. Verify address of ADI/MDI, by viewing power-up message with proper DIP switch settings or Transmit <CTRL-HOME> C1<CTRL-END> 6. Use a different serial port or disconnect the modem or mouse. 7. Check switch setting.

TROUBLESHOOTING GUIDE (Cont'd)

PROBLEMS	POSSIBLE CAUSE	REMEDIES
ADI/MDI IS NOT RESPONDING TO MESSAGE REQUEST OVER SERIAL PORT	1. Improperly wired. 2. Incorrect transmission string sent from external unit. 3. Wrong address transmitted. 4. Message NOT programmed in the ADI/MDI. 5. Serial set-up parameters do NOT match the ADI/MDI's. 6. ADI/MDI waiting to collect embedded data. 7. Proper serial terminator character NOT sent. 8. "20mA DIS." DIP switch in wrong position (ADI only).	1. Check wiring. 2. Perform serial hardware test. 3. Verify address. 4. Verify Message File. 5. Verify set-up parameters (baud rate, parity, data bits). 6. Wait until embedded data is collected, then retransmit. 7. Verify Terminator of ADI/MDI on power-up with proper DIP switch setting or Transmit <CTRL-HOME> C5 <CTRL-END> in Terminal Emulator. 8. Check switch setting.
ADI/MDI IS NOT RESPONDING TO MESSAGE REQUEST OVER PARALLEL PORT	1. Improperly wired. 2. Message is NOT programmed into the ADI/MDI. 3. Information on DATA lines in the wrong message mode (ex. Binary on DATA lines, but ADI/MDI expecting BCD). 4. Incorrect number of strobes for parallel port configuration 5. MESSAGE/DATA line in wrong logic state. 6. DATA and/or CTRL SINK/SRC switch in wrong position. 7. DATA and/or CTRL LO/HI Bias switch in wrong position (ADI only).	1. Check wiring. 2. Verify program. 3. Verify parallel port message request mode. 4. Check port width. 5. Verify logic switches. 6. Check switch position. 7. Check switch position.
ADI/MDI IS NOT RECEIVING DATA FROM EXTERNAL UNIT "- - - - " IN EMBEDDED DATA FIELD	1. External unit NOT powered. 2. External unit sending wrong terminator. 3. Embedded data time-out too short. 4. Serial transmit delay too short. 5. Improper data format sent.	1. Verify power. 2. Check terminator. 3. Increase time-out value. 4. Increase delay time. 5. Check data format.

TROUBLESHOOTING GUIDE (Cont'd)

PROBLEMS	POSSIBLE CAUSE	REMEDIES
ADI/MDI IS NOT RECEIVING DATA OVER PARALLEL PORT	1. Improperly wired. 2. Data was NOT strobed in. 3. MESSAGE/DATA line in wrong logic state. 4. DATA and/or CTRL SINK/SRC switch in wrong position. 5. Message does NOT actually expect data over parallel port. 6. DATA and/or CTRL LO/HI Bias switch in wrong position (ADI only).	1. Check wiring. 2. Check strobe levels. 3. Verify logic switches. 4. Check switch position. 5. Verify message program. 6. Check switch position.
ADI/MDI AND SERIAL EXTERNAL UNIT(S) NOT COMMUNICATING	1. Improperly wired. 2. Serial set-up parameters do NOT match the ADI/MDI's. 3. Serial external unit NOT powered. 4. Incorrect serial terminator. 5. Wrong address. 6. "20mA DIS." DIP switch in wrong position (ADI only).	1 a. Check wiring. b. Perform serial hardware test. 2. Verify parameters. 3. Verify power. 4. Verify terminator of data source. 5. Check address. 6. Verify switch setting.
PARTIAL DATA IN EMBEDDED DATA FIELD	1. Embedded data time-out duration too short.	1. Verify time duration.
BLANK LINE ON DISPLAY WHEN CHARACTERS SHOULD BE PRESENT	1. Too many leading blanks for line. 2. Line contains only embedded data and all data has been formatted out. 3. Line contains only embedded data and no data was received, just the correct terminator character.	1. Check message. 2. Check program to verify that format is correct. 3. Verify transmission from external unit.

CONFIGURATION AND SELF-TEST FUNCTIONS

The ADI/MDI contains several DIP switch selectable configuration and self-test functions that can be performed only on power-up. To invoke any of these functions, the three Baud Rate DIP switches (BR0, BR1, & BR2) on the rear of the unit must be in the UP position when power is applied to the unit. So long as the baud rate switches remain in the UP position, any of the DIP switch functions can be performed by changing the DIP switch settings to the next desired function. Configuration information and self-test results are shown on the ADI/MDI's Display. To exit the DIP switch functions, simply move any of the baud rate switches to its DOWN position. While executing the DIP switch functions, the ADI/MDI ignores all Message requests and Command strings. If the Baud Rate switches are placed in the Test position while the unit is powered, the following Error Message is displayed in block scroll fashion at the current block scroll rate:

<ERROR>
INVALID
BAUD RATE
SETTING

The following functions can be performed by setting the DIP switches as shown (1 = DIP switch in the UP position, 0 = DIP switch in the DOWN position).

Note: Data Logic and Control Logic DIP switches are NOT used in the Self-Test Mode.

DIP SWITCH FUNCTIONS						
BR0	**BR1**	**BR2**	**PARITY**	**EVEN/ODD**	**7/8**	**DESCRIPTION**
1	1	1	0	0	0	Displays the Unit Hardware Configuration.
1	1	1	1	0	0	Displays the Unit Configuration File Settings.
1	1	1	0	1	0	Displays the Parallel Port Configuration & current Data & Control Pin levels.
1	1	1	1	1	0	Performs a Serial Self-Test.
1	1	1	0	0	1	Performs a Memory Self-Test.
1	1	1	1	0	1	Performs an RTC Self-Test.
1	1	1	0	1	1	MD ONLY: Sets MD Display Frequency to 225 Hz. *
1	1	1	1	1	1	Clears Non-Volatile Memory restores all files to Factory Default settings.

* *Note that doing a "Unit Clear" will change the unit to the ADI's Display Frequency.*

Unit Hardware Configuration

The Unit Hardware Configuration can be displayed with the following DIP switch settings:

PARITY	EVEN/ODD	7/8
0	0	0

The ADI/MDI block scrolls the Hardware Configuration on the top line of the display.

	ADI	MDI
Hardware Version	VER - x.xx	VER - x.xx
Memory Size	ADI - 32K	MDI - 32K
Display Size	"2 x 20"	"1 x 10" or "2 x 20"
Parallel Port	"PRL PORT"	"PRL PORT" or "NO PORT"

Unit Configuration File Settings

The Unit Configuration File can be displayed with the following DIP switch settings:

PARITY	EVEN/ODD	7/8
1	0	0

The ADI/MDI block scrolls its current Configuration File Settings on the top line of the display. The first three blocks display a Capital letter followed by 8 flag settings (either 1 or 0). The remaining blocks display a lower case "parameter" letter followed by a three digit decimal value. A list of the parameters and their codes is shown below in Display order.

CODE LETTER	DESCRIPTION OF SETTING	
C	**Unit Configuration Byte**	**Factory default settings - 11110000**
	BIT 7	0 - Parallel Port set for 4 or 8 Bits 1 - Parallel Port set for 9-Bit BCD or an AMR Mode
	BIT 6	0 - Parallel Port set for 4 Bits or AMR Mode 2 1 - Parallel Port set for 8 or 9 Bits or AMR Mode 1
	BIT 5	0 - Parallel Port set for Binary Message requests 1 - Parallel Port set for BCD Message requests
	BIT 4	0 - XON/XOFF is OFF 1 - XON/XOFF is ON
	BIT 3	0 - Queue function is OFF 1 - Queue function is ON
	BIT 2	0 - Periodic Message function is OFF 1 - Periodic Message function is ON
	BIT 1	0 - Default Message function is OFF 1 - Default Message function is ON
	BIT 0	0 - Reset Message Function is OFF 1 - Reset Message Function is ON

CODE LETTER	DESCRIPTION OF SETTING	
H	**Unit Hardware Byte**	**Factory default settings** **ADI - XX000110 MDI - XX000X11**
	BIT 7	0 - Parallel Port Control lines set for Negative Logic 1 - Parallel Port Control lines set for Positive Logic
	BIT 6	0 - Parallel Port Data lines set for Negative Logic 1 - Parallel Port Data lines set for Positive Logic
	BIT 5	0 - Serial Port set for 7 data bits 1 - Serial Port set for 8 data bits
	BIT 4	0 - Serial Port set for Even parity 1 - Serial Port set for Odd parity
	BIT 3	0 - Serial Port set for No parity 1 - Serial Port set for parity
	BIT 2	0 - Indicates a 1x10 Display 1 - Indicates a 2x20 Display
	BIT 1	1 - Unit has a Parallel Port
	BIT 0	0 - Unused 1 - Unit has 32 K of memory
O	**Output Pin Configuration Byte**	**Factory default settings - 00000000**
	BIT 7	0 - Output Pin in the Busy Mode (Can Only be in the Busy Mode if XON/XOFF is OFF) 1 - Output Pin in the Output Mode
	BIT 6	0 - Output Pin programmed for Negative Logic 1 - Output Pin programmed for Positive Logic
	BIT 5	0 - Output is OFF 1 - Output is ON
	BIT 4	0 - Output is Unlocked 1 - Output is Locked
	BIT 3	0 - Time Out is Halted (Output Mode Only) 1 - Time Out is Running (Output Mode Only)
	BIT 2	0 - Time Out value not in Hundreds 1 - Time Out value is in Hundreds (Time Out is in SECS if both BITs 1 and 2 are 0)
	BIT 1	0 - Time Out value not in Minutes 1 - Time Out value is in Minutes
	BIT 0	0 - Time Out has not Timed Out 1 - Time Out has Timed Out

Unit Hardware Configuration File Settings

CODE LETTER	DESCRIPTION OF SETTING	RANGE	DEFAULT
c	- Unit Address	000-099	000
d	- Character Scroll Time Top Line	001-127	016
e	- Character Scroll Time Bottom Line	001-127	016
f	- Block Scroll Time Top Line	001-255	001
g	- Block Scroll Time Bottom Line	001-255	001
h	- Blink Time Top Line	001-127	050
i	- Blink Time Bottom Line	001-127	050
j	- Default Message Number	000-255	000
k	- Reset Message Number	000-255	000
l	- Serial Port Transmit Delay Time	000-255	005
m	- Parallel Port Debounce Time	001-255	005
n	- Display Backlight Intensity Level	000-016	008
o	- Output Pin Time Out Value	001-255	010
p	- Serial Port ED Time Out Value	000-255	002
q	- Parallel Port ED Time Out Value	000-255	001
r	- Error Handling Type *Note: Range includes ASCII B (066), D (068), I (073), and T (084)*		066
s	- Serial Port Terminator *Note: Range excludes BS (008), ESC (027), and 0 to 9 (048-057)*	000-064	042
t	- Parallel Port Sample Time	001-255	005

Parallel Port Configuration and Logic Levels

The Parallel Port Configuration Test can be performed with the following DIP switch settings:

PARITY	EVEN/ODD	7/8
0	1	0

The ADI/MDI block scrolls the current Parallel Port DIP switch settings and the logic levels on the Control and Data lines. The line levels shown are the actual levels at the Parallel Port pins, not the logic levels after translation for Positive or Negative logic. The line levels are read only when entering this test.

The results are Displayed as shown below:

PRL PORT

CTRL LVL (+/-) -CTRL LOGIC DIP switch setting '+' (UP) '-' (DOWN)

DATA LVL (+/-) - DATA LOGIC DIP switch position '+' (UP) '-' (DOWN)

xxxxxxxxxx - H (HIGH) or L (LOW) indicating the current Logic level on each of the ten Parallel Port lines - STROBE, MESSAGE/DATA, D7-D0

Serial Port Loop-Back Test

The ADI/MDI must be wired in a serial loop-back configuration to properly perform the Serial Port Loop-Back test. The test may be performed using either the 20 mA or RS-232 serial port as shown in Figure 46, Serial Loop-back Configuration.

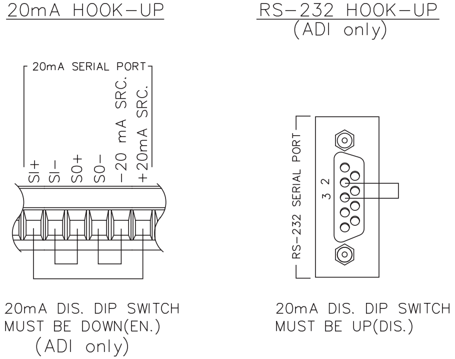


Figure 36, Serial Loop-back Configuration

The Serial Port Loop-Back test can be performed with the following DIP switch settings:

PARITY	EVEN/ODD	7/8
1	1	0

The ADI/MDI transmits 32 characters for each of the 7 baud rates. Upon completion of the test, the unit automatically transmits the test results, either ‘P’ (PASS) or ‘F’ (FAIL) at 1200 baud, 8 bits, and NO Parity. The ADI/MDI also places the test results on the display.

SRL - (PASS/FAIL)

Memory Test

The ADI/MDI performs a limited Non-volatile Memory Test with the following DIP switch settings:

PARITY	EVEN/ODD	7/8
0	0	1

The ADI/MDI tests every non-volatile memory location for Read and Write capability. The unit does NOT change any of the information in any of the Configuration, Message, or Character Files. Upon completion of the test, the unit automatically transmits the test results, either ‘P’ (PASS) or ‘F’ (FAIL) at 1200 baud, 8 bits, and NO Parity. The ADI/MDI also places the test results on the display.

MEM - (PASS/FAIL)

Real Time Clock (RTC) Test

The Real Time Clock Test can be performed with the following DIP switch settings:

PARITY	EVEN/ODD	7/8
1	0	1

The ADI/MDI tests the RTC for Read, Write, and Run capabilities. This test does not change the current value of the clock. Upon completion of the test, the unit automatically transmits the test results, either ‘P’ (PASS), or ‘F’ (FAIL) at 1200 baud, 8 bits, NO Parity. The ADI/MDI also places the test results on the display.

RTC - (PASS/FAIL)

Clearing Non-Volatile Memory

The Clear Non-Volatile Memory function can be performed with the following DIP switch settings and ONLY at unit POWER-UP. If the DIP switches are set for this test while power is applied, the previously mentioned Error Message is Displayed.

PARITY	EVEN/ODD	7/8
1	1	1

Caution: This setting causes the ADI/MDI to clear its non-volatile memory of all programmed information, including the Configuration, Message, and Character files, and restore the RLC Factory Default settings and files. As long as the DIP switches are not in this setting at POWER-UP, it is not possible to accidentally erase the unit’s programmed information by inadvertently switching to these settings at any other time.

ADI INSTALLATION AND SPECIFICATIONS GUIDE

GENERAL DESCRIPTION

The ADI is available in AC and DC models. The display features a transmissive LCD with LED backlighting. The unit is available in positive or negative image LCD versions with a choice of backlight colors. The display consists of two lines of 5X7 dot-matrix characters, with 20 characters per line. A large 0.45 (11.4 mm) character height makes the ADI display readable to 15 feet (4.5 meters). An on-board pot allows for adjustment of the display viewing angle to accommodate various mounting heights.

The sealed front panel of the ADI meets NEMA 4/IP65 requirements for indoor use, allowing for washdown when properly installed. Modern surface-mount technology and extensive testing make the unit extremely reliable in industrial environments. Connections are made on rear panel removable terminal blocks that accept solid or stranded wire in the range of 12 to 24 AWG.

⚠ ⏚ The protective conductor terminal is bonded to conductive parts of the equipment for safety purposes and must be connected to an external protective earthing system.

INSTALLATION AND CONNECTIONS

Installation Environment

The unit should be installed in a location that does not exceed the maximum operating temperature and provides good air circulation. Placing the unit near devices that generate excessive heat should be avoided.

Continuous exposure to direct sunlight may accelerate the aging process of the display.

The display overlay may be cleaned using alcohol compounds (Isopropanol or methanol) or liquid glass cleaners that DO NOT contain ammonia.

Note: Exposure of the display to ketone solvents will cause it to whiten.

The ADI meets NEMA 4/IP65 requirements for indoor use when properly installed. The units are intended to be mounted into an enclosed panel. A gasket and a packet of kep nuts (#10-32), are supplied with the unit. The recommended minimum panel thickness for NEMA 4/IP65 applications is 0.060 (1.52 mm). Maximum panel thickness is 0.375 (9.5 mm).

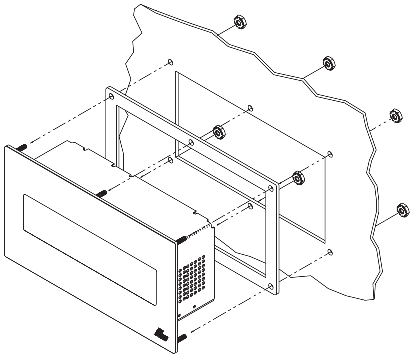


Figure 37, Panel Installation

Refer to Figure 38 for panel cut-out dimensions. After the panel cut-out has been completed and deburred, insert the unit with the panel gasket, into the panel as shown in Figure 37, Panel Installation. Install six kep nuts and tighten evenly for uniform gasket compression.

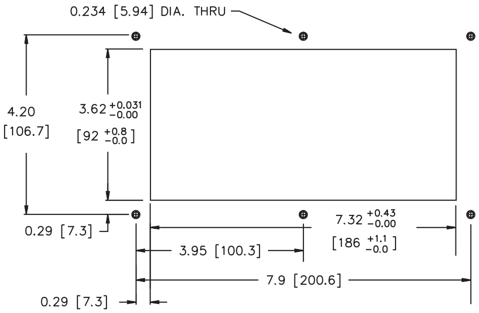


Figure 38, Panel Cut-Out Dimension

EMC INSTALLATION GUIDELINES

Although this unit is designed with a high degree of immunity to ElectroMagnetic Interference (EMI), proper installation and wiring methods must be followed to ensure compatibility in each application. The type of electrical noise, its source, or the method of coupling into the unit may be different for various installations. The unit becomes more immune to EMI with fewer I/O connections. Cable length, routing, and shield termination are very important and can mean the difference between a successful or a troublesome installation. Listed below are some EMC guidelines for successful installation in an industrial environment.

1. The unit should be mounted in a metal enclosure that is properly connected to protective earth.
2. Use shielded (screened) cables for all Signal and Control inputs. The shield (screen) pigtail connection should be made as short as possible. The connection point for the shield depends somewhat upon the application. Listed below are the recommended methods of connecting the shield, in order of their effectiveness.
   a. Connect the shield only at the panel where the unit is mounted to earth ground (protective earth).
   b. Connect the shield to earth ground at both ends of the cable when the noise source frequency is above 1 MHz.
   c. Connect the shield to common of the unit and leave the other end of the shield unconnected and insulated from earth ground.
3. Never run Signal or Control cables in the same conduit or raceway with AC power lines, conductors feeding motors, solenoids, SCR controls, and heaters, etc. The cables should be run in metal conduit that is properly grounded. This is especially useful in applications where cable runs are long and portable two-way radios are used in close proximity, or if the installation is near a commercial radio transmitter.
4. Signal or Control cables within an enclosure should be routed as far away as possible from contactors, control relays, transformers, and other noisy components.
5. In extremely high EMI environments, the use of external EMI suppression devices such as ferrite suppression cores, is effective. Install them on Signal and Control cables as close to the unit as possible. Loop the cable through the core several times or use multiple cores on each cable for additional protection. Install line filters on the power input cable to the unit to suppress power line interference. Install them near the power entry point of the enclosure. The following EMI suppression devices (or equivalent) are recommended:

   Ferrite Suppression Cores for signal and control cables:
   Fair-Rite # 0443167251 (RLC #FCOR0000)
   TDK # ZCAT3035-1330A
   Steward #28B2029-0A0

   Line Filters for input power cables:
   Schaffner # FN610-1/07 (RLC #LFIL0000)
   Schaffner # FN670-1.8/07
   Corcom #1VR3

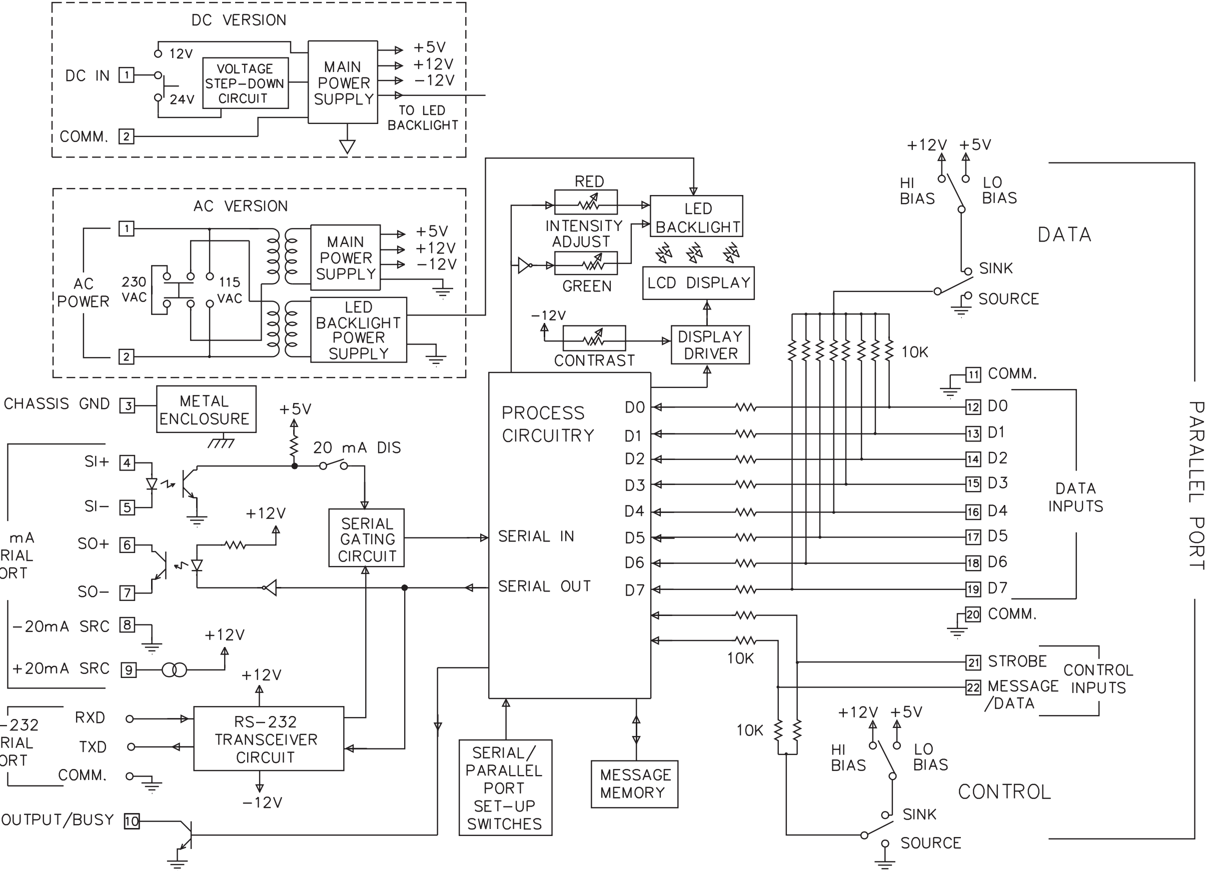
Note: Reference manufacturer's instructions when installing a line filter.

6. Long cable runs are more susceptible to EMI pickup than short cable runs. Therefore, keep cable runs as short as possible.

Wiring Considerations

After the unit has been mechanically mounted, it is ready to be wired. All wiring connections are made on removable plug-in terminal blocks at the rear of the unit. All conductors should meet voltage and current ratings for each terminal. Also, cabling should conform to appropriate standards of good installation, local codes and regulations. It is recommended that power supplied to the unit (AC or DC) be protected by a fuse or circuit breaker. When wiring the unit, remove the terminal block and use the numbers on the label to identifythe terminal positions. Strip the wire, leaving approximately 1/4 bare wire exposed (stranded wire should be tinned with solder). Insert the wire into the terminal and tighten down the screw until the wire is clamped tightly. Each terminal can accept up to one 14-gauge, two 18-gauge or four 20-gauge wire(s). After the terminal block is wired, install it in the proper location at the rear of the unit.

BLOCK DIAGRAM



Rear Panel Description

Terminal Connections

AC Power Wiring and Voltage Selection

AC power is connected to positions #1 and #2 on the terminal block, as indicated on the label. The voltage selector switch, located to the left side of the terminal block, is used to select either 115 or 230 VAC. The switch is a slide movement type and can be set by using a small screwdriver.

Caution: Before applying power to the unit, make sure the selector switch is set for the proper voltage setting. Damage to the unit may occur if the AC selector switch is set incorrectly.

To reduce the chance of noise spikes entering the AC line and affecting the unit, the AC power should be relatively "clean" and within the specified ±10% variation limit. Connecting power from heavily loaded circuits or circuits which also power loads that cycle on and off, (contactors, relays, motors, etc.) should be avoided.

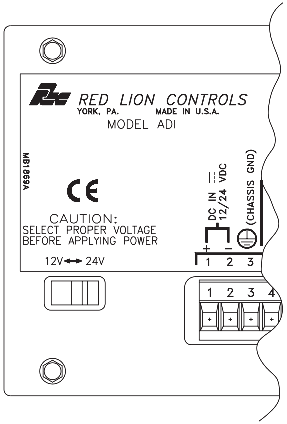
DC Power Wiring and Voltage Selection

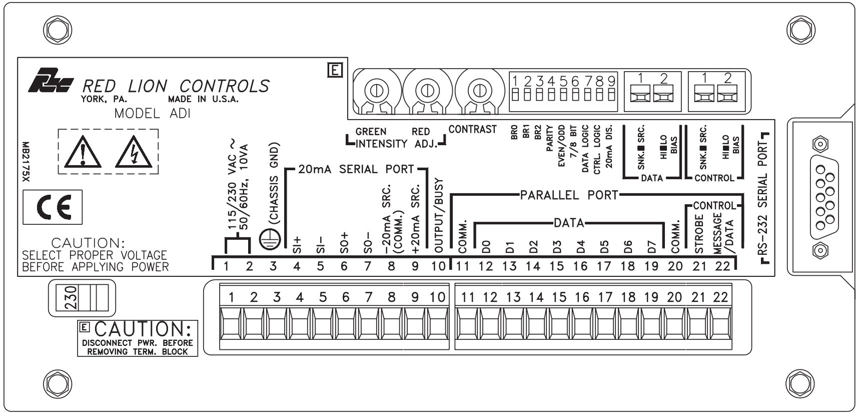
DC power is connected to positions #1 and #2 on the terminal block, as indicated on the label. The DC power source must be capable of supplying the unit's rated current (450 mA max.). The voltage selector switch, located to the left side of the terminal block, is used to select either 12 or 24 VDC. The switch is a slide movement type and can be set by using a small screwdriver.

Caution: Before applying power to the unit, make sure the selector switch is set for the proper voltage setting. Damage to the unit may occur if the DC selector switch is set incorrectly.

Chassis Ground

The CHASSIS GND terminal provides a means for grounding the metal enclosure of the ADI. This terminal is normally connected to power ground, or to the metal enclosure in which the ADI is installed. In this manner, the ADI's enclosure provides an electrical shield for the internal circuitry, thus improving the level of electrical noise immunity. The CHASSIS GND terminal is electrically isolated from the ADI's circuit common (COMM.) terminals.





20 mA Serial Port Wiring

When communicating with the ADI via the 20 mA SERIAL PORT, connections are made on the terminal block at the positions indicated on the label.

Note: When using the 20 mA SERIAL PORT, the "20 mA DIS." DIP switch must be in the "enabled" (down) position for the ADI to receive serial information. See 20 mA Port Disable & Serial Port Interaction, for further description.

20 mA Serial Terminal Descriptions

Receive Terminals: SI+(Serial In +)
SI-(Serial In -)

The ADI receives data, commands and message requests on these terminals. Files from the SFMD software are downloaded to the ADI over these terminals as well. Connect the output terminals of the device that transmits to the ADI to these terminals.

Transmit Terminals: SO+(Serial Out +)
SO-(Serial Out -)

The ADI issues all of its transmissions over these terminals. Data files in the ADI are uploaded over these terminals to the SFMD software as well. Connect the input terminals of the device to which the ADI transmits to these terminals.

Current Loop Power:

The +20 mA SRC terminal provides a 20 milliamp source current for one of the loops. The -20 mA SRC terminal is the return path for the +20 mA terminal.

Output/Busy

The OUTPUT/BUSY terminal is an NPN open-collector output. This terminal can be used as a Ready/Busy handshaking line for communications, or as an Output pin whose status is controlled by messages and commands. See *OUTPUT PIN* for further details on the function of this terminal.

Parallel Port Wiring

The PARALLEL PORT is an input-only port from which the ADI accepts message requests and receives Embedded Data. When communicating via the PARALLEL PORT, connections are made on the terminal block at the positions indicated on the label. (See *PARALLEL PORT* for a detailed functional description.)

Parallel Port Terminal Descriptions

Data Inputs: D0 thru D7

The ADI accepts message requests and receives Embedded Data thru the DATA inputs. These lines are DIP switch selectable to accept outputs from current sinking (SNK.) or current sourcing (SRC.) devices. For sourcing devices, the input trip levels can be set to accept 5 V (LO BIAS) or 12 V (HI BIAS) logic. The logic polarity of the DATA inputs is DIP switch selectable for Positive or Negative logic.

Control Inputs: STROBE and MESSAGE/DATA

The STROBE line is used to provide a trigger signal which prompts the ADI to begin sampling the Parallel Port DATA and MESSAGE/DATA lines. The level of the MESSAGE/DATA line determines whether information on the Parallel Port DATA inputs represents a message request or Embedded Data.

The CONTROL inputs are DIP switch selectable to accept outputs from current sinking (SNK.) or current sourcing (SRC.) devices. For sourcing devices, the input trip levels can be set to accept 5V (LO BIAS) or 12V (HI BIAS) logic. The logic polarity of the CONTROL inputs is DIP switch selectable for Positive or Negative logic.

Circuit Common: COMM.

Provides a circuit common reference for any external devices connected to the PARALLEL PORT.

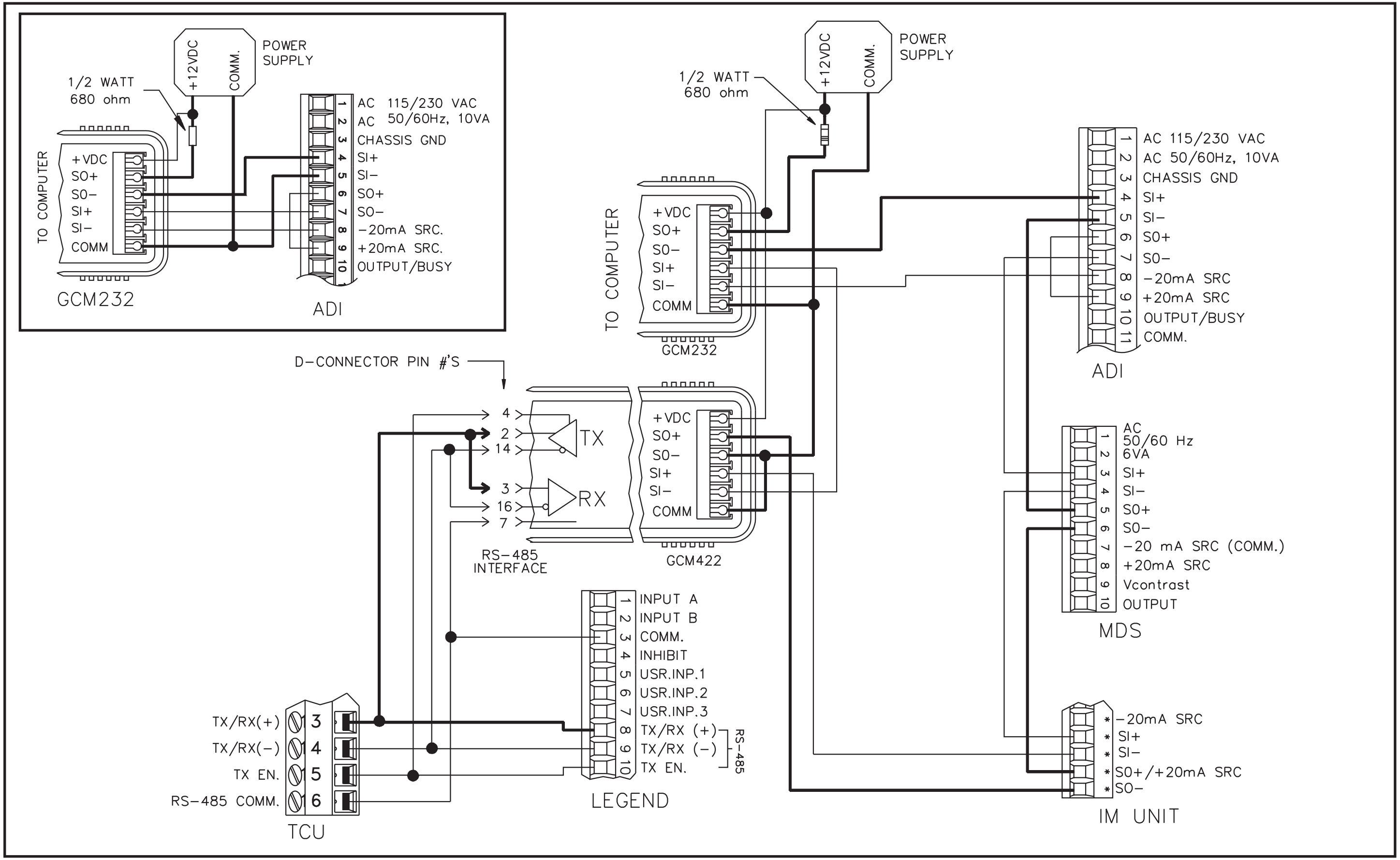
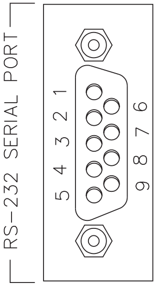


Figure 39, 20 mA Serial Wiring Examples

RS-232 Serial Port Connections

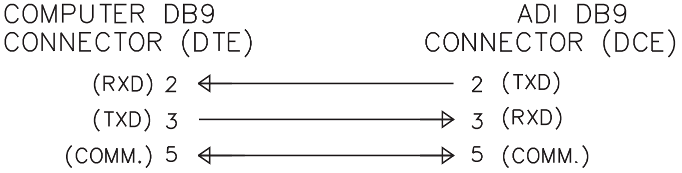
Connections to the ADI's RS-232 SERIAL PORT are made on a 9-Pin Female D-type socket. Pin descriptions are as follows:

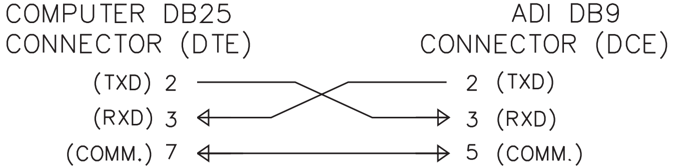


Pin 2: Transmit Data (TXD)- From the ADI to the external device
Pin 3: Receive Data (RXD) - From the external device to the ADI
Pin 5: Signal Common

Note: Normally, when the RS-232 Serial Port is used, the "20mA DIS." DIP switch must be in the "disabled" (up) position for the ADI to receive serial information. See 20mA Port Disable & Serial Port Interaction, for further description.

Typical RS-232 connections between the ADI and a computer are shown below.





Rear Panel Adjustments

Display Contrast

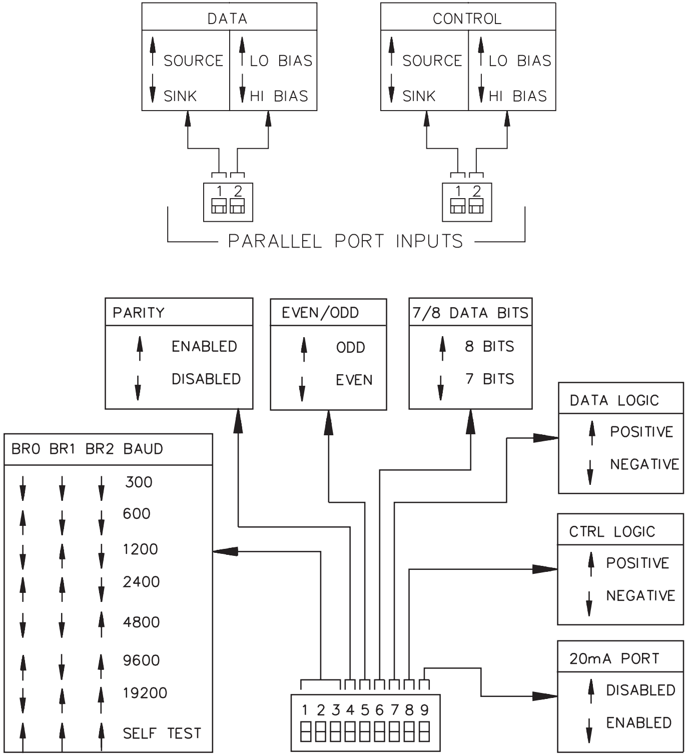
The optimal viewing angle of the display can be changed by adjusting the "CONTRAST" pot located on the rear panel. This pot varies the applied voltage to the LCD display.

Intensity Adjust
(Tri-Color Unit Only)

On single color ADI units, the backlight intensity of the display is software adjustable through 17 levels and ranges from 0(OFF) to 16(FULL INTENSITY) in 1/16 level increments. On a Tri-Color unit, however, this software setting controls the backlight color from FULL RED(0) through FULL YELLOW/GREEN(16). The actual intensity of the backlight is controlled with the INTENSITY ADJ. pots located on the rear panel. Two pots are provided for individual control of the RED and GREEN portions of the backlight. Normally, the pots should only be adjusted with the display at the FULL RED or FULL YELLOW/GREEN settings. Each pot is then adjusted for the desired brightness level of the appropriate backlight color.

Dip Switch Settings

A 9-position DIP switch and two 2-position DIP switches are located on the rear panel of the ADI. These DIP switches are used to establish certain Serial and Parallel Port parameters. The parameter choices are as follows:



BR0	BR1	BR2	BAUD
↓	↓	↓	300
↑	↓	↓	600
↓	↑	↓	1200
↑	↑	↓	2400
↓	↓	↑	4800
↑	↓	↑	9600
↓	↑	↑	19200
↑	↑	↑	SELF TEST

20 mA Port Disable & Serial Port Interaction

The two Serial Ports on the ADI, 20 mA and RS-232, share a common serial input buffer inside the unit. The serial DIP switch parameter settings (Baud Rate, Parity, Data Bits) apply to both serial ports. The serial input lines are internally gated together in a logical “AND” configuration. This means a MARK (Logic 1) condition must be present on both serial inputs for a Logic 1 to be read at the input buffer. Conversely, a SPACE (Logic 0) condition on either input results in a Logic 0 being read at the input buffer.

Whenever an application uses the 20 mA Serial Port for data input, the “20 mA DIS.” DIP switch must be in the “enabled” (down) position. Since the RS-232 Serial Port defaults to the MARK condition when not in use, the ADI can receive data through the 20 mA input regardless of whether an RS-232 device is connected. The 20 mA port, however, defaults to the SPACE condition when not in use.

Consequently, in applications which use only the RS-232 port, the “20 mA DIS.” DIP switch must be set to the “disabled” (up) position. In this position, the DIP switch internally sets the 20 mA input to the MARK condition, allowing the ADI to receive data from an RS-232 device without having to “hard wire” the 20 mA input.

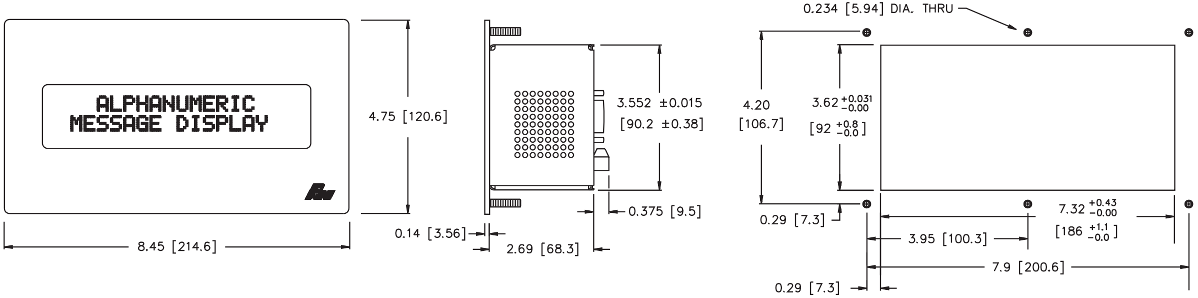
The ADI is capable of receiving data on both the RS-232 and 20 mA ports in the same application, although not simultaneously. For example, the ADI could receive commands and message files from a host computer through the RS-232 port, in addition to collecting embedded data over the 20 mA port. However, since both ports share a common input buffer, care must be taken to avoid simultaneous data reception which would be erroneously interpreted by the ADI. In these applications, the “20 mA DIS.” DIP switch must be in the “enabled” (down) position. Also, the 20 mA input loop must be in the MARK (current flowing) condition in order for the RS-232 port to receive data.

The “20 mA DIS.” DIP switch setting only applies to the serial input and has no effect on the serial output. All serial information transmitted from the ADI is sent to both the 20 mA and RS-232 ports.

SPECIFICATIONS

1. **POWER:**
   AC Version: Switch selectable, 115/230 VAC ±10%, 50/60 Hz, 10 VA max.
   DC Version: Switch selectable, 12/24 VDC ±10%, 450 mA max. Power supplies must be Class 2 or SELV rated.
2. **DISPLAY:**
   2x20: 0.45 (11.4 mm) high characters, readable to 15 feet (4.5 meters).
   Transmissive SBE LCD: Negative Image with Red LED backlighting, OR Positive Image with Yellow-Green or Tri-color LED backlighting. The brightness of the LED backlight is software adjustable through seventeen levels for the single color units. For the Tri-Color unit, the color can be adjusted across 17 shades from Full Red to Orange to Full Yellow-Green. On-board pots separately adjust the intensity of the red and green backlight. The optimal viewing angle of the display may also be adjusted through an on-board pot.
3. **PARALLEL COMMUNICATIONS:**
   Message Request Format:
   Binary: 4 or 8 bits.
   BCD: 4, 8, or 9 bits.
   Embedded Data Format:
   Binary: 4 or 8 bits.
   BCD: 4 or 8 bits.
   ASCII: 4 or 8 bits.
   PARALLEL PORT INPUTS:
   Data Inputs (D0 - D7) & Control Inputs (Strobe & Message/Data):
   V_{MAX} = 30 VDC
   Data Lo/Hi Bias: 5 V or 12 V compatible logic levels, switch selectable
   Lo Bias: V_{IH} = 3.5 VDC min; V_{IL} = 1.5 VDC max.
   Hi Bias: V_{IH} = 8 VDC min; V_{IL} = 4 VDC max.
   Control Lo/Hi Bias: 5 V or 12 V compatible logic levels, switch selectable
   Lo Bias: V_{IH} = 3.5 VDC min; V_{IL} = 1.5 VDC max.
   Hi Bias: V_{IH} = 8 VDC min; V_{IL} = 4 VDC max.
   Data SNK/SRC: Sink or Source, switch selectable.
   Control SNK/SRC: Sink or source, switch selectable.
   Current Sinking: Internal 10 Kohm pull-up, I_{SNK} = 1.2 mA. max.
   Current Sourcing: Internal 10 Kohm pull-down, I_{SRC} = 5.6 mA max. @ 30 VDC.
   Data Logic Level: Positive or negative, switch selectable.
   Control Logic Level: Positive or negative, switch selectable.
   Debounce Time: 0.01 to 2.55 seconds (programmable).
   Strobe Time: 3 to 255 msec (programmable).

DIMENSIONS In inches (mm)



4. **SERIAL COMMUNICATIONS:** RS-232 port, w/9-Pin D-type female connector, full duplex. Also, 20 mA current loop, w/DIP switch Enable/Disable.

Data Format: Four types available, switch selectable.
11 bits: 1 start bit, 8 data bits, 1 parity bit, 1 stop bit
10 bits: 1 start bit, 8 data bits, 1 stop bit
10 bits: 1 start bit, 7 data bits, 1 parity bit, 1 stop bit
9 bits: 1 start bit, 7 data bits, 1 stop bit

Data Code: ASCII

Unit Address: Programmable from 0 to 99. (The number of units in a single loop is limited by the hardware specifications.)

Baud Rate: 300 to 19200, switch selectable.

Parity: Enabled or Disabled, switch selectable.

Even/Odd: Selects Parity Type, switch selectable.

7/8 BIT: Data Bits, switch selectable.

Serial Hardware:

20 mA Current Loop: Terminal block connections

+20 mA SRC: Provides 20 mA nominal @ 12 VDC.
Note: Can power up to 7 units in a loop.

-20 mA SRC: Loop return for +20 mA SRC.

SO/Output Transistor Rating:
V_{CE} = 30 VDC max., V_{SAT}= 1 VDC max. @ 20 mA.
Note: Transistor rating allows for up to 28 units in a loop.

SI/Input Diode Rating: V_F = 1.25 VDC_{TYP}; 1.5 VDC_{MAX} @ 20 mA.
Note: The compliance voltage rating of the source must be greater than the sum of the diode voltage drops around the loop. Typically a 30 VDC source (with adequate current capability) is capable of operating between 18 and 22 units in a loop.

RS-232 Port: 9-Pin D-type female connector (DCE)
Pin 2 (TXD): Transmit Data (From the ADI to the host computer)
Pin 3 (RXD): Receive Data (From the host computer to the ADI)
Pin 5: Signal Common

5. **OUTPUT/BUSY PIN:**

Solid state: NPN open-collector, current sinking, V_{OH} = 30 VDC max., I_{SNK} = 100 mA max. @ V_{OL} = 1 VDC max.

Busy Mode: Indicates the Ready/Busy status of the unit.

Output Mode: Output is activated from a Command or Message for a specified time out value.

Time Out: 10 msec to 63 mins or Latched.

6. **REAL-TIME CLOCK:** Non-volatile Date and Time, accurate to ±1 minute/month.

7. **MEMORY:** 32 Kbytes of non-volatile memory retains all programmed Configuration, Message, and Character settings when power is removed or interrupted. Provides space for 256 Messages of 100 bytes each while capable of storing 4K bytes of Embedded Data. The actual amount of embedded data storage is determined by the size of the message file.

8. **ENVIRONMENTAL CONDITIONS:**

Operating Temperature: 0° to +50° C.
Storage Temperature: -20° to +60° C.
Operating and Storage Humidity: 85% max. relative humidity (non-condensing) from 0° to +50° C
Altitude: Up to 2000 meters

9. **CERTIFICATIONS AND COMPLIANCES:**
ELECTROMAGNETIC COMPATIBILITY

Immunity to EN 50082-2		
Electrostatic discharge	EN 61000-4-2	Level 2; 4 Kv contact Level 3; 8 Kv air
Electromagnetic RF fields	EN 61000-4-3	Level 3; 10 V/m 80 MHz - 1 GHz
Fast transients (burst)	EN 61000-4-4	Level 4; 2 Kv I/O Level 3; 2 Kv power
RF conducted interference	EN 61000-4-6	Level 3; 10 V/rms 150 KHz - 80 MHz
Simulation of cordless telephone	ENV 50204	Level 3; 10 V/m 900 MHz ± 5 MHz 200 Hz, 50% duty cycle
Emissions to EN 50081-2		
RF interference	EN 55011	Enclosure class A Power mains class A

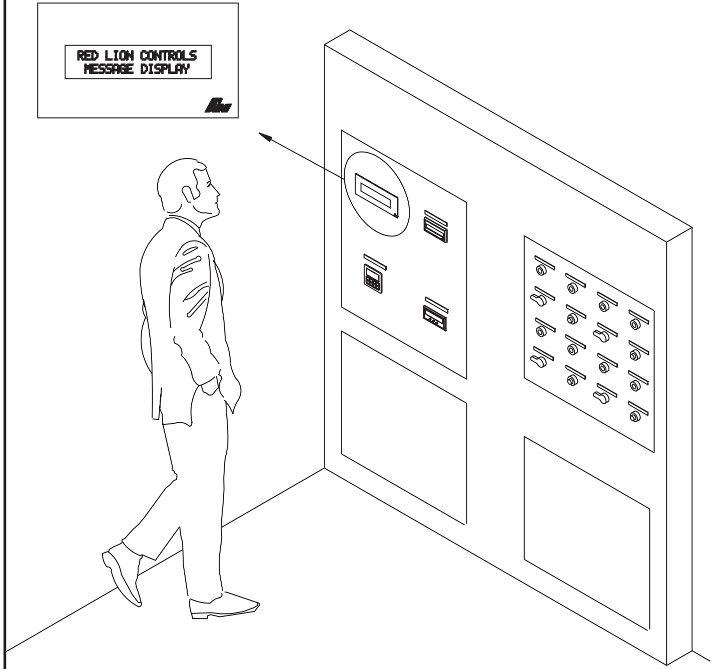
Refer to the EMC Installation Guidelines for additional information.

10. **MOUNTING REQUIREMENTS:** Max. panel thickness is 0.375" (9.5 mm). Min. panel thickness for NEMA 4/IP65 sealing is 0.060" (1.57 mm).

11. **CONSTRUCTION:** Steel construction with textured black polyurethane paint for scratch and corrosion resistance. Sealed front panel meets NEMA 4/IP65 specifications for indoor use when properly installed. (panel gasket and kep nuts included).Installation Category II, Pollution Degree 2.

12. **CONNECTIONS:** Rear panel removable terminal blocks.

13. **WEIGHT:** 3.4 lbs (1.5 Kg).



APPLICATION #1

A customer is installing a PLC based system, and would like to display various messages on the ADI. The PLC is wired to the ADI to request specific messages dependent upon the state of the process. Once the ADI receives and processes the message request, it displays the message that corresponds to the change in the process. The ADI is also set up to transmit the displayed messages to various MDS units located at remote locations.

The ADI and the PLC are connected serially, using the RS232 card of the PLC and the 9 pin RS232 port of the ADI. The cable run between the units is shielded twisted pair (For better noise immunity). The PLC's serial card is requesting messages from the ADI.

The ADI will re-transmit the displayed messages to various MDS units, using the 20 mA current loop. The ADI is set up for "Display & MDS unit" in the software. The MDS's are set up for RLC mode. The output of the ADI is wired to the inputs of the MDS units. With the handshaking off, the outputs of the MDS units do not need to be wired to the input of the ADI.

APPLICATION #2

A customer would like to monitor several process variables at one location. At various stages of the process, the ADI will display the process data for the operator. If a critical condition develops, the ADI will display that condition, as well as the suspected cause.

A TCU is used to monitor and control the heating of the process. An IMH is used to monitor the status of the heating element and an IMI to monitor the speed of the process.

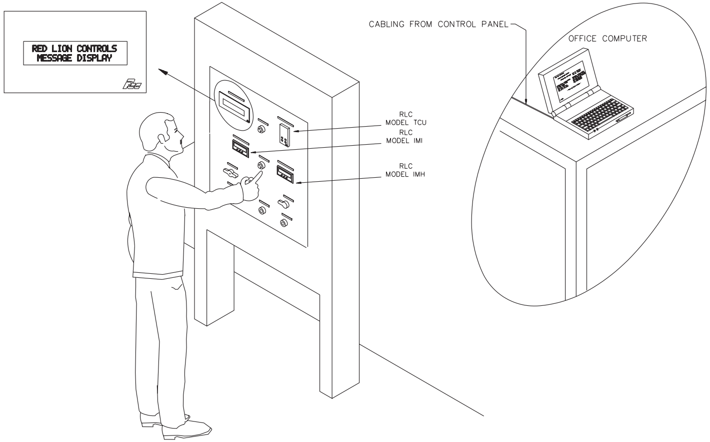
Messages will display the information that is pertinent during the start-up of the process. Messages are programmed to take into account all the various conditions the process may incur. The parallel port is configured for AMR Mode 1 for Message requests over the parallel port.

The TCU, IMH, and IMI are set-up to activate their alarm outputs in the event of an error condition. Each output is tied to a separate DATA line on the parallel port of the ADI. The ADI will automatically request a specific Message from the parallel port when a level change occurs.

Example: The TCU's alarm output 1 is tied to DATA line D0. In the event of a high temperature condition, alarm output 1 closes. The ADI monitors DATA line D0 for a logic level change. DATA line D0 has two message numbers attached to it, Message #1 for the Active logic level and Message #11 for the Inactive logic level. The ADI requests Message #1 when D0 undergoes an Inactive-to-Active transition, and informs the operator of a high temperature condition, and the time it occurred. Message #11 is requested when the D0 undergoes an Active-to-Inactive transition, and indicates that the temperature condition was corrected, and the time of correction.

When the operator presses a switch to start the process, the start-up Messages are requested. This switch is connected to one of the ADI's parallel port DATA lines. Messages can also be requested and monitored via the serial port, which is connected to the office computer.

All Messages and Configuration Settings are programmed on an IBM compatible computer using the Message Display User Software (SFMD). The Message and Configuration files are then downloaded to the ADI via the serial port. After the files are downloaded, the ADI is ready to interface with the system.



MDI INSTALLATION AND SPECIFICATIONS GUIDE

GENERAL DESCRIPTION

The MDI is available in AC and DC models. The MDI utilizes a Liquid Crystal Display (LCD) with direct LED backlighting. Display options include 1x10 and 2x20 character formats, negative image with Red LED backlighting, negative image with Tri-color backlighting, OR positive or negative image with Yellow-Green LED backlighting. The brightness of the display, and the various blink and scroll times can be adjusted.

The metal front bezel of the MDI meets NEMA 4/IP65 requirements for indoor use, allowing for washdown when properly installed. Modern surface-mount technology and extensive testing make the unit extremely reliable in industrial environments. Connections are made on rear panel removable terminal blocks, simplifying installation.

EMC Installation Guidelines

Although this unit is designed with a high degree of immunity to ElectroMagnetic Interference (EMI), proper installation and wiring methods must be followed to ensure compatibility in each application. The type of electrical noise, its source, or the method of coupling into the unit may be different for various installations. The unit becomes more immune to EMI with fewer I/O connections. Cable length, routing, and shield termination are very important and can mean the difference between a successful or a troublesome installation. Listed below are some EMC guidelines for successful installation in an industrial environment.

1. The unit should be mounted in a metal enclosure that is properly connected to protective earth.
2. Use shielded (screened) cables for all Signal and Control inputs. The shield (screen) pigtail connection should be made as short as possible. The connection point for the shield depends somewhat upon the application. Listed below are the recommended methods of connecting the shield, in order of their effectiveness.
   a. Connect the shield only at the panel where the unit is mounted to earth ground (protective earth).
   b. Connect the shield to earth ground at both ends of the cable when the noise source frequency is above 1 MHz.
   c. Connect the shield to common of the unit and leave the other end of the shield unconnected and insulated from earth ground.
3. Never run Signal or Control cables in the same conduit or raceway with AC power lines, conductors feeding motors, solenoids, SCR controls, and heaters, etc. The cables should be run in metal conduit that is properly grounded. This is especially useful in applications where cable runs are long and portable two-way radios are used in close proximity, or if the installation is near a commercial radio transmitter.
4. Signal or Control cables within an enclosure should be routed as far away as possible from contactors, control relays, transformers, and other noisy components.
5. In extremely high EMI environments, the use of external EMI suppression devices such as ferrite suppression cores, is effective. Install them on Signal and Control cables as close to the unit as possible. Loop the cable through the core several times or use multiple cores on each cable for additional protection. Install line filters on the power input cable to the unit to suppress power line interference. Install them near the power entry point of the enclosure. The following EMI suppression devices (or equivalent) are recommended:

   Ferrite Suppression Cores for signal and control cables:
      Fair-Rite # 0443167251 (RLC #FCOR0000)
      TDK # ZCAT3035-1330A
      Steward #28B2029-0A0
   Line Filters for input power cables:
      Schaffner # FN610-1/07 (RLC #LFIL0000)
      Schaffner # FN670-1.8/07
      Corcom #1VR3

   Note: Reference manufacturer's instructions when installing a line filter.
6. Long cable runs are more susceptible to EMI pickup than short cable runs. Therefore, keep cable runs as short as possible.

INSTALLATION AND CONNECTIONS

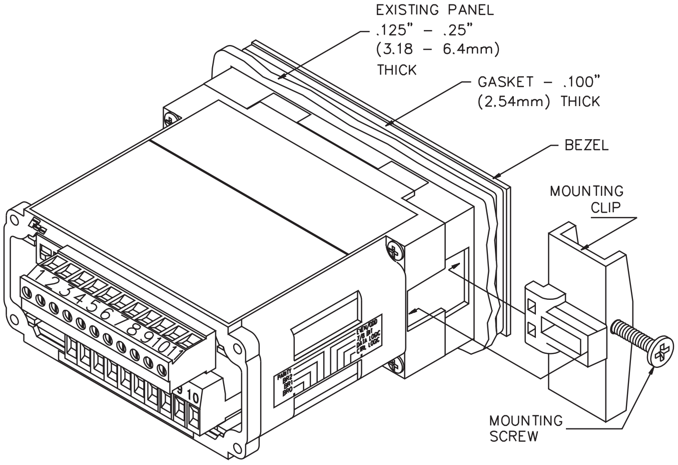


Figure 40, Panel Installation

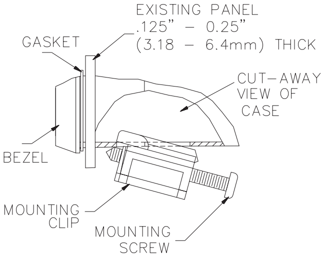
Installation Environment

The unit should be installed in a location that does not exceed the maximum operating temperature and provides good air circulation. Placing the unit near devices that generate excessive heat should be avoided.

Continuous exposure to direct sunlight may accelerate the aging process of the display.

The MDI meets NEMA 4/IP65 requirements for indoor use when properly installed. The units are intended to be mounted into an enclosed panel with a gasket to provide a water-tight seal. Two mounting clips and screws are provided for easy installation. Consideration should be given to the thickness of the panel. A panel that is too thin may distort and not provide a water-tight seal. (Recommended minimum panel thickness is 1/8").

Cut the panel opening to the specified dimensions. Remove burrs and clean around the panel opening. Slide the panel gasket over the rear of the unit to the back of the bezel. Insert the unit into the panel. As depicted in the drawing, install mounting clips. Thread the screws into the clips until the pointed end just protrudes through the other side. Install each of the mounting clips by inserting the wide lip of the clips into the wide end of the hole, located on either side of the case. Then snap the clip into the case. Tighten the screws evenly to apply uniform compression, thus providing a water-tight seal.

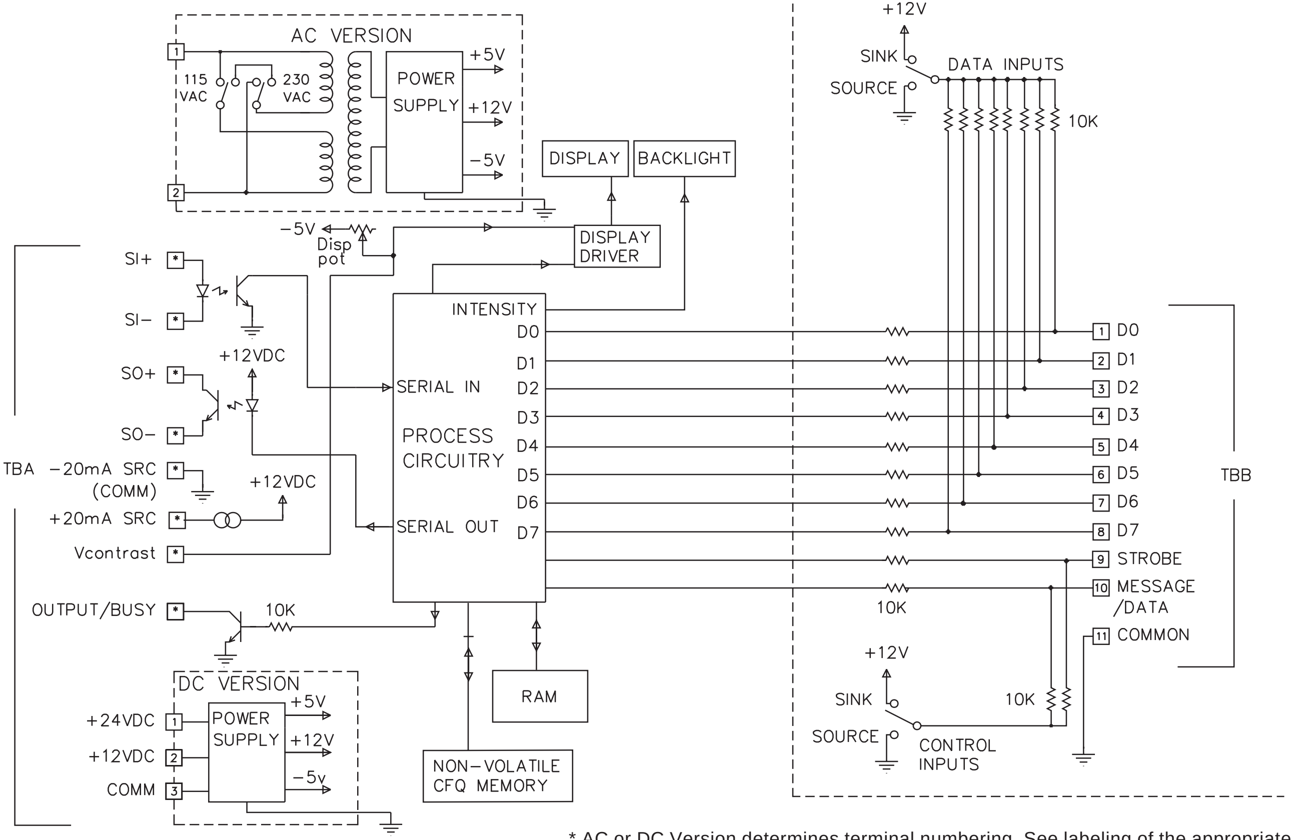


Caution: Only minimum pressure is required to seal the panel. Do NOT overtighten the screws.

Wiring Considerations

After the unit has been mechanically mounted, it is ready to be wired. All wiring connections are made on removable plug-in terminal blocks at the rear of the unit. All conductors should meet voltage and current ratings for each terminal. Also, cabling should conform to appropriate standards of good installation, local codes and regulations. It is recommended that power supplied to the unit (AC or DC) be protected by a fuse or circuit breaker. There is a separate terminal block for the bottom board (TBA)and top board (TBB). When wiring the unit, remove the terminal block and use the numbers on the label to identify the terminal positions. Strip the wire, leaving approximately 1/4" bare wire exposed (stranded wire should be tinned with solder). Insert the wire into the terminal and tighten down the screw until the wire is clamped tightly. Each terminal can accept up to one 14-gauge, two 18-gauge or four 20-gauge wire(s). After the terminal block is wired, install it in the proper location at the rear of the unit. Wire each terminal block in this manner.

BLOCK DIAGRAM



* AC or DC Version determines terminal numbering. See labeling of the appropriate unit.

Vcontrast (Single Color Units Only)

The contrast of the display can be changed by adjusting the internal pot located on the side of the unit. The Vcontrast terminal located on TBA #9 allows for adjustment of the display contrast from a remote location. To set up the unit for remote adjustment do the following:

1. Set the internal pot to the desired contrast.
2. Connect a 50K 1/4Watt Pot between Vcontrast and common.

The external pot, which can be mounted on the front of the panel, can now be used to adjust the display's contrast.

Vcontrast and Intensity(Tri-Color Units Only)

The contrast of the display can be adjusted by connecting a 50K 1/4 Watt Pot between Vcontrast, terminal located on TBA #9, and common. You can mount this pot on the front of the panel if the display needs to be adjusted during normal operation.

The Intensity of the RED portion of the Backlight can be adjusted with the internal pot located on the side of the unit. The Yellow-Green intensity cannot be adjusted.

AC Power Wiring

AC power is connected to TBA #1 and #2 located on the bottom terminal block. The voltage selector switch, located to the left side of TBA, is used to select the proper voltage. The switch is a slide movement type and can be set by using a small screwdriver. If the switch is showing "115", it is set for 115 VAC input. If "230" is showing, the switch is set for 230 VAC input.

⚠ *Caution: Before applying power to the unit, make sure the selector switch is set for the proper voltage setting. Damage to the unit may occur if the AC selector switch is set incorrectly.*

To reduce the chance of noise spikes entering the AC line and affecting the unit, the AC power should be relatively "clean" and within the specified ±10% variation limit. Connecting power from heavily loaded circuits or circuits which also power loads that cycle on and off, (contactors, relays, motors, etc.) should be avoided.

DC Power Wiring

DC power is applied between TBA #1 and TBA #3 (common) for +24 VDC input or between TBA #2 and TBA #3 (common) for +12 VDC input. The DC power source must be capable of supplying the unit's rated current (300mA). It is not necessary to provide battery backup to retain programmed information, the MDI contains its own non-volatile memory and retains programmed information on power down (Refer to block diagram).

Serial Wiring

Connections for the 20mA current loop are made on TBA, the bottom terminal block. Power to the MDI and the current loop should be turned OFF before removing TBA and making any connections. Install wires into the proper location on the terminal block. Refer to the label on top of the unit for locations and descriptions of each terminal. When all necessary connections have been made, replace the terminal block. The terminal block is keyed for proper installation.

Serial Terminal Descriptions

Receive Terminals: SI+(Serial In +)
SI-(Serial In -)

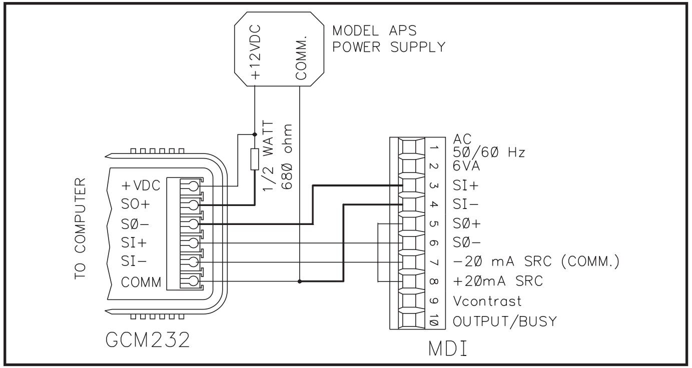
The MDI receives data, commands and message requests on these terminals. Files from the SFMD software are downloaded to the MDI over these terminals, as well. Connect the output terminals of the device that transmits to the MDI to these terminals.

Transmit Terminals: SO+(Serial Out +)
SO-(Serial Out -)

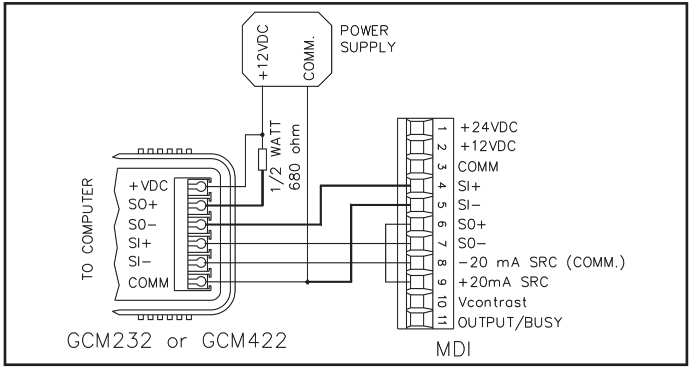
The MDI issues all of its transmissions over these terminals. Files in the MDI are uploaded over these terminals to the SFMD software, as well. Connect the input terminals of the device to which the MDI transmits to these terminals.

Current Loop Power: The +20mA SRC terminal provides a 20 milliamp source current for one of the loops. The -20mA SRC (COMM.) terminal is the return path for the +20mA terminal.

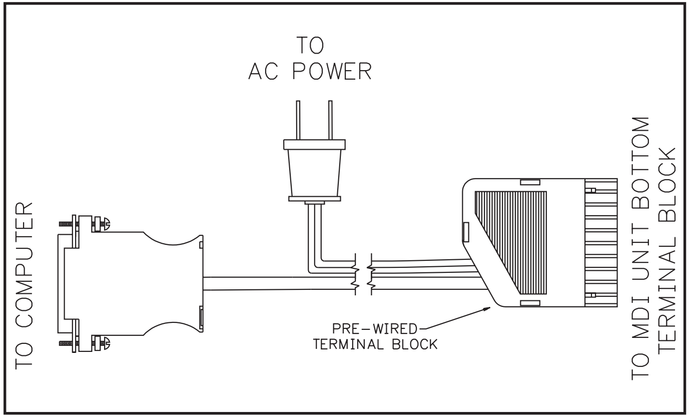
AC Version 20 mA Loop Hook-Up



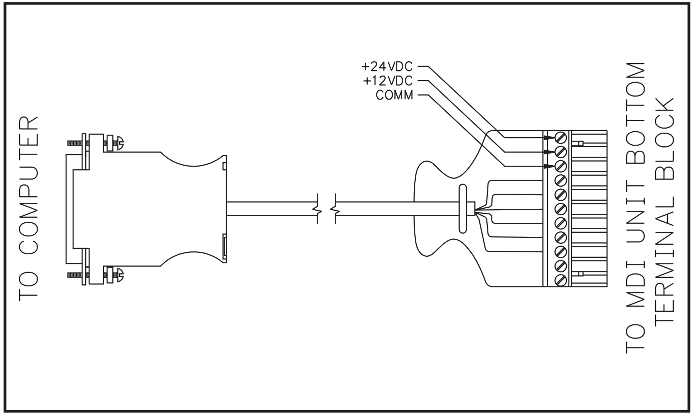
DC Version 20 mA Hook-Up



MCCA AC Cable Hook-Up

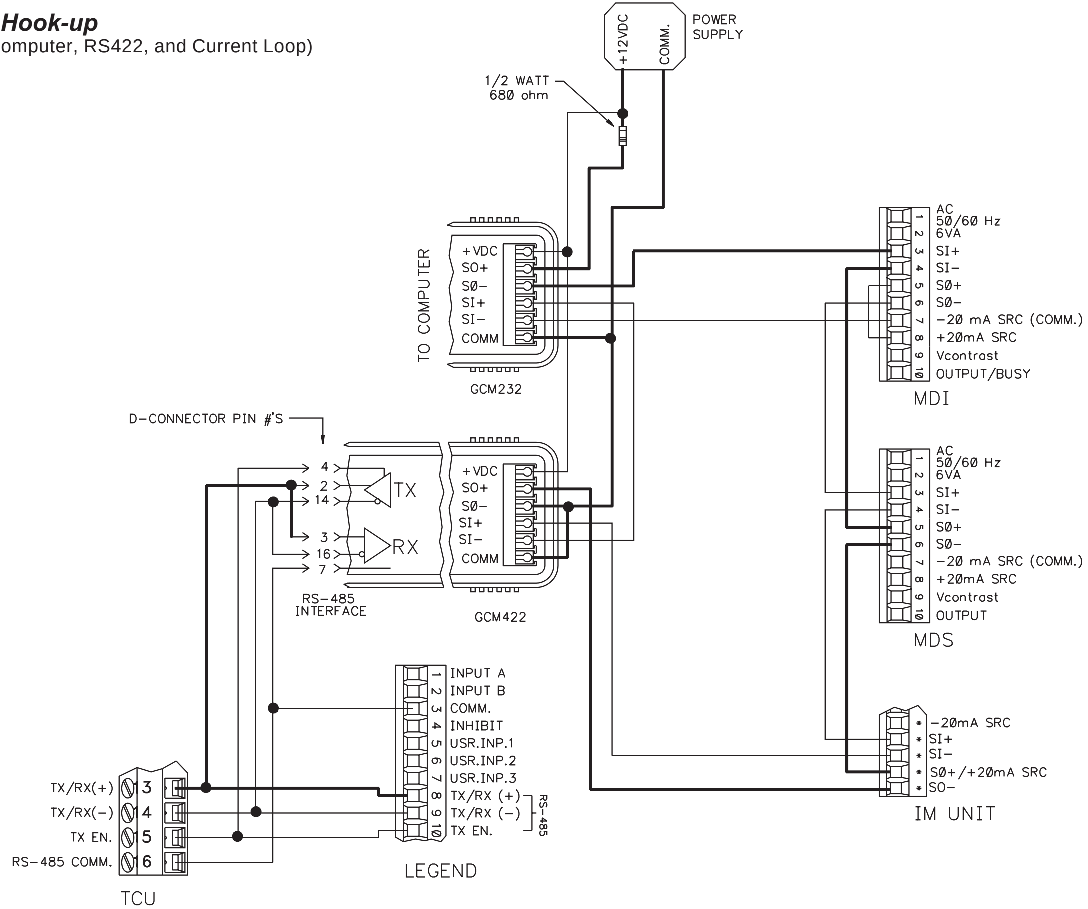


MCCA DC Cable Hook-Up



Multiple Unit Hook-up

(Interfacing with Computer, RS422, and Current Loop)



Output/Busy

The OUTPUT/BUSY terminal is an NPN open-collector output. This terminal can be used as a Ready/Busy handshaking line for communications, or as an Output pin whose status is controlled by messages and commands. See *OUTPUT PIN for further details on the function of this terminal.*

Parallel Port Wiring

The PARALLEL PORT is an input-only port from which the MDI accepts message requests and receives Embedded Data. When communicating via the PARALLEL PORT, connections are made on the terminal block at the positions indicated on the label. (See *PARALLEL PORT* for a detailed functional description.)

Parallel Port Terminal Descriptions

Data Inputs: D0 thru D7

The MDI accepts message requests and receives Embedded Data thru the DATA inputs. These lines are DIP switch selectable to accept outputs from current sinking (SNK.) or current sourcing (SRC.) devices. The logic polarity of the DATA inputs is DIP switch selectable for Positive or Negative logic.

Control Inputs: STROBE and MESSAGE/DATA

The STROBE line is used to provide a trigger signal which prompts the MDI to begin sampling the Parallel Port DATA and MESSAGE/DATA lines. The logic state of the MESSAGE/DATA line determines whether information on the Parallel Port DATA inputs represents a message request or Embedded Data.

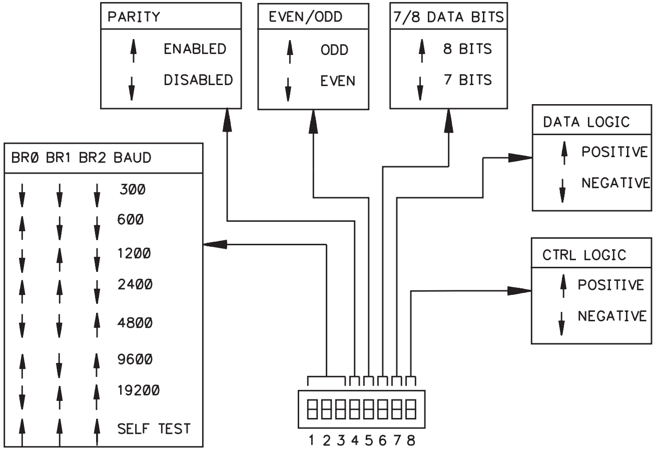
The CONTROL inputs are DIP switch selectable to accept outputs from current sinking (SNK.) or current sourcing (SRC.) devices. The logic polarity of the CONTROL inputs is DIP switch selectable for Positive or Negative logic.

Circuit Common: COMM.

Provides a circuit common reference for any external devices connected to the PARALLEL PORT.

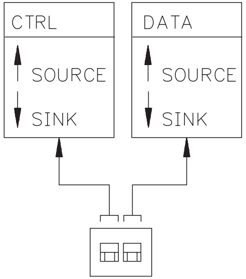
DIP Switch Settings

An 8 position DIP switch is located on the side of the MDI. A 2 position DIP switch is located at the rear of the unit. The DIP switches are used to establish certain Serial and Parallel Port parameters. The parameter choices are as follows:

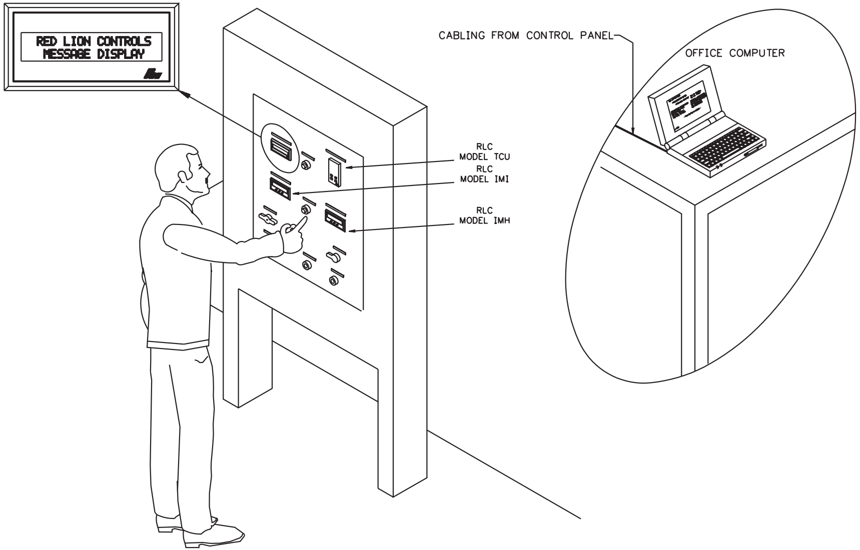


BR0	BR1	BR2	BAUD
↓	↓	↓	300
↑	↓	↓	600
↓	↑	↓	1200
↑	↑	↓	2400
↓	↓	↑	4800
↑	↓	↑	9600
↓	↑	↑	19200
↑	↑	↑	SELF TEST

These DIP switches are located at the rear of the unit, their switch positions and functions are shown at right.



APPLICATION



A customer would like to monitor several process variables at one location. At various stages of the process, the MDI will display the process data for the operator. If a critical condition develops, the MDI will display that condition, as well as the suspected cause.

A TCU is used to monitor and control the heating of the process. An IMH is used to monitor the status of the heating element and an IMI to monitor the speed of the process.

Messages will display the information that is pertinent during the start-up of the process. Messages are programmed to take into account all the various conditions the process may incur. The parallel port is configured for AMR Mode 1 for Message requests over the parallel port.

The TCU, IMH, and IMI are set up to activate their alarm outputs in the event of an error condition. Each output is tied to a separate DATA line on the parallel port of the MDI. The MDI will automatically request a specific Message from the parallel port when a level change occurs.

Example: The TCU's alarm output 1 is tied to DATA line D0. In the event of a high temperature condition, alarm output 1 closes. The MDI monitors DATA line D0 for a logic level change. DATA line D0 has two message numbers attached to it, Message #1 for the Active logic level and Message #11 for the Inactive logic level. The MDI requests Message #1 when D0 undergoes an Inactive-to-Active transition, and informs the operator of a high temperature condition, and the time it occurred. Message #11 is requested when the D0 undergoes an Active-to-Inactive transition, and indicates that the temperature condition was corrected, and the time of correction.

When the operator presses a switch to start the process, the start-up Messages are requested. This switch is connected to one of the MDI's parallel port DATA lines. Messages can also be requested and monitored via the serial port, which is connected to the office computer.

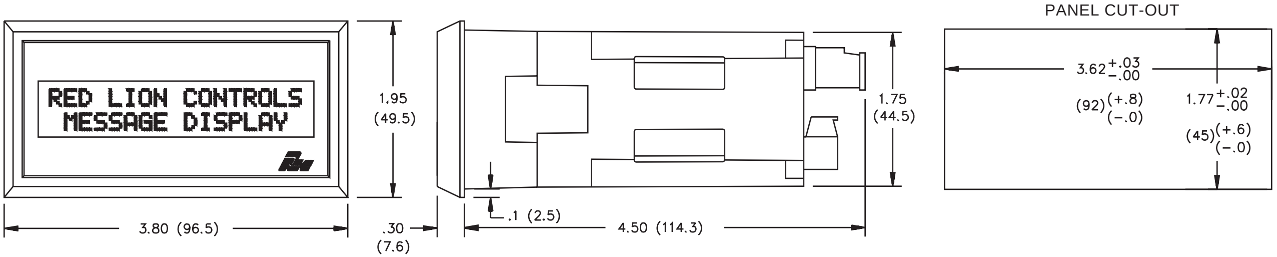
All Messages and Configuration Settings are programmed on an IBM® compatible computer using the Message Display User Software (SFMD). The Message and Configuration files are then downloaded to the MDI via the serial port. The serial link between the computer and the MDI is conveniently established using the MCCA programming cable. After the files are downloaded, the MDI is ready to interface with the system.

SPECIFICATIONS

1. **POWER:**
   AC Version: Switch selectable, 115/230 VAC ±10%, 50/60 Hz, 8 VA. max.
   DC Version: Jumper selectable, 12/24 VDC ±10%, 300 mA max. Power supplies must be Class 2 or SELV rated.
2. **DISPLAY:** (available in various configurations)
   2x20: 0.2" (5 mm) high characters.
   1x10: 0.35" (9 mm) high characters.
   Transmissive STN LCD: Negative Image with Red LED backlighting, Negative Image with Tri-color backlighting (Red, Orange, Yellow-Green), OR Positive or Negative Image with Yellow-Green backlighting. The brightness of the LED backlight is software adjustable through seventeen levels for the single color units. For the Tri-Color unit, the color can be adjusted across 17 shades from Full Red to Orange to Full Yellow-Green. The contrast of the display may be adjusted through an on-board pot for single Color units. For the Tri-Color unit, the on-board pot adjusts the Red Intensity. By connecting an external pot to the Vcontrast pin, the contrast can be adjusted from a remote location.
3. **PARALLEL COMMUNICATIONS:**
   Message Request Format:
   Binary: 4 or 8 bits.
   BCD: 4, 8, or 9 bits.
   Embedded Data Format:
   Binary: 4 or 8 bits.
   BCD: 4 or 8 bits.
   ASCII: 4 or 8 bits.
   PARALLEL PORT INPUTS:
   Data Inputs (D0 - D7) & Control Inputs (Strobe & Message/Data):
   V_{IH} = 8 VDC_{MIN}, V_{IL} = 4 VDC_{MAX}, V_{MAX} = 30 VDC
   Data SNK/SRC: Sink or Source, switch selectable.
   Control SNK/SRC: Sink or source, switch selectable.
   Current Sinking: Internal 10 Kohm pull-up, I_{SNK} = 1.2 mA. max.
   Current Sourcing: Internal 10 Kohm pull-down, I_{SRC} = 3.1 mA max. @ 30 VDC.
   Data Logic Level: Positive or negative, switch selectable.
   Control Logic Level: Positive or negative, switch selectable.
   Debounce Time: 0.01 to 2.5 seconds (programmable).
   Strobe Time: 3 to 255 msec (programmable).
4. **SERIAL COMMUNICATIONS:** 20 mA current loop, full-duplex
   Data Format: Four types available, switch selectable.
   11 bits: 1 start bit, 8 data bits, 1 parity bit, 1 stop bit
   10 bits: 1 start bit, 8 data bits, 1 stop bit
   10 bits: 1 start bit, 7 data bits, 1 parity bit, 1 stop bit
   9 bits: 1 start bit, 7 data bits, 1 stop bit
   Data Code: ASCII
   Unit Address: Programmable from 0 to 99. (The number of units in a single loop is limited by the hardware specifications.)

DIMENSIONS In inches (mm)

Note: Recommended minimum Clearance (behind the panel) for mounting clip installation is 2.1" (53.3) H x 5.5" (140) W.



Baud Rate: 300 to 19200, switch selectable.
Parity: Enabled or Disabled, switch selectable.
Even/Odd: Selects Parity Type, switch selectable.
7/8 BIT: Data Bits, switch selectable.
Serial Hardware: Terminal TBA

+20 mA SRC: Provides 20 mA nominal @ 12 VDC.
Note: Can power up to 6 units in a loop.

-20 mA SRC: Loop return for +20 mA SRC.

SO/Output Transistor Rating:
V_{CE} = 30 VDC max., V_{SAT} = 1 VDC max. @ 20 mA.
Note: Transistor rating allows for up to 28 units in a loop.

SI/Input Diode Rating: V_F = 1.25 VDC_{TYP}; 1.5 VDC_{MAX} @ 20 mA.

Note: The compliance voltage rating of the source must be greater than the sum of the diode voltage drops around the loop. Typically a 30 VDC source (with adequate current capability) is capable of operating between 18 and 22 units in a loop.

5. **OUTPUT/BUSY PIN:**
Solid state: NPN open-collector, current sinking, V_{OH} = 30 VDC max., I_{SNK} = 100 mA max. @ V_{OL} = 1 VDC max.
Busy Mode: Indicates the Ready/Busy status of the unit.
Output Mode: Output is activated from a Command or Message for a specified time out value.
Time Out: 10 msec to 63 mins or Latched.

6. **REAL-TIME CLOCK**: Non-volatile Date and Time, accurate to ±1 minute/month.

7. **MEMORY:** Non-volatile memory retains all programmed Configuration, Message, and Character settings when power is removed or interrupted.
32K: Provides space for 256 Messages of 100 bytes each while capable of storing 4 K bytes of Embedded Data.

8. **ENVIRONMENTAL CONDITIONS:**
Operating Temperature: 0° to +50°C.
Storage Temperature: -20° to +60°C.
Operating and Storage Humidity: 85% max. relative humidity (non-condensing) from 0° to +50°C
Altitude: Up to 2000 meters

9. **CERTIFICATIONS AND COMPLIANCES:**
UL Recognized Component, File # E171375
Recognized to U.S. and Canadian requirements under the Component Recognition Program of Underwriters Laboratories, Inc.

ELECTROMAGNETIC COMPATIBILITY

Immunity to EN 50082-2		
Electrostatic discharge	EN 61000-4-2	Level 2; 4 Kv contact[1] Level 3; 8 Kv air
Electromagnetic RF fields	EN 61000-4-3	Level 3; 10 V/m 80 Mhz - 1 GHz
Fast transients (burst)	EN 61000-4-4	Level 4; 2 Kv I/O[2] Level 3; 2 Kv power[3]
RF conducted interference	EN 61000-4-6	Level 3; 10 V/rms[3] 150 Khz - 80 MHz
Simulation of cordless telephone	ENV 50204	Level 3; 10 V/m 900 Mhz ± 5 MHz 200 Hz, 50% duty cycle
Emissions to EN 50081-2		
RF interference	EN 55011	Enclosure class A Power mains class A

Notes:
1. Metal bezel of unit connected with ground lead from rear bezel screw to metal mounting panel.
2. For operation without loss of performance:
I/O cables are routed in metal conduit connected to earth ground.
3. For operation without loss of performance:
Install power line filter, RLC#LFIL0000 or equivalent.
Refer to the EMC Installation Guidelines for additional information.

10. **CONSTRUCTION:** Sealed front panel meets NEMA 4/IP65 specifications for indoor use when properly installed. Case is black high impact plastic (panel gasket, mounting clips and screws included). Installation Category II, Pollution Degree 2.

11. **CONNECTIONS:** Rear panel removable terminal blocks.

12. **WEIGHT:** 1.1 lbs (0.5 Kg).

ORDERING INFORMATION

MODEL NO.	DESCRIPTION	PART NUMBERS	
		115/230 VAC	12/24 VDC
ADI	Red, Negative Image	ADI2R11A	ADI2R11D
	Yel-Grn, Positive Image	ADI4Y11A	ADI4Y11D
	Tri-Color, Positive Image	ADI4T11A	ADI4T11D
MDI	1x10 Red, Negative Image	MDI1R11A	MDI1R11D
	1x10 Yel-Grn, Negative Image	MDI1Y11A	MDI1Y11D
	1x10 Yel-Grn, Positive Image	MDI3Y11A	MDI3Y11D
	2x20 Red, Negative Image	MDI2R11A	MDI2R11D
	2x20 Yel-Grn, Negative Image	MDI2Y11A	MDI2Y11D
*	2x20 Tri-Color, Negative Image	MDI2T11A	MDI2T11D
SFMD	Apollo Message Display User Software (3 1/2"; 1.44M)	SFMD0	
GCM232	Serial Converter Module RS-232	GCM23201	
APS01	+12 VDC Unregulated Power Supply, 115 VAC	APS01000	
APS02	+12 VDC Unregulated Power Supply, 230 VAC	APS02000	
–	Base Mount, 8-Pin Octal Socket	SKT10000	
MCCA	AC Communication Adapter	MCCA0000	
	DC Communication Adapter	MCCA1000	

Note: Only one copy of software is required for multiple units.
** - Check availability with the factory.*

This page is intentionally left blank.

LIMITED WARRANTY

The Company warrants the products it manufactures against defects in materials and workmanship for a period limited to one year from the date of shipment, provided the products have been stored, handled, installed, and used under proper conditions. The Company's liability under this limited warranty shall extend only to the repair or replacement of a defective product, at The Company's option. The Company disclaims all liability for any affirmation, promise or representation with respect to the products.

The customer agrees to hold Red Lion Controls harmless from, defend, and indemnify RLC against damages, claims, and expenses arising out of subsequent sales of RLC products or products containing components manufactured by RLC and based upon personal injuries, deaths, property damage, lost profits, and other matters which Buyer, its employees, or sub-contractors are or may be to any extent liable, including without limitation penalties imposed by the Consumer Product Safety Act (P.L. 92-573) and liability imposed upon any person pursuant to the Magnuson-Moss Warranty Act (P.L. 93-637), as now in effect or as amended hereafter.

No warranties expressed or implied are created with respect to The Company's products except those expressly contained herein. The Customer acknowledges the disclaimers and limitations contained and relies on no other warranties or affirmations.

A-MDI/IM - E 2/03
DRAWING NO. LP0397

Red Lion Controls
20 Willow Springs Circle
York PA 17402
Tel +1 (717) 767-6511
Fax +1 (717) 764-0839

Red Lion Controls BV
Basicweg 11b
NL - 3821 BR Amersfoort
Tel +31 (0) 334 723 225
Fax +31 (0) 334 893 793

Red Lion Controls Asia
31, Kaki Bukit Road 3 #06-04/05 TechLink
Singapore 417818
Tel +65 6744-6613
Fax +65 6743-3360