A MITEL PRODUCT GUIDE

MiContact Center Enterprise

Wall Display and Real Time Messaging – Description

Release 9.8
Document Version 1.0

August 2025

Mitel
Powering connections

Notices

Trademarks

INTRODUCTION

This document is designed to allow a user to interpret the real-time information interface from Information Manager on MiCC Enterprise.

The interface facilitates the transmitting of messages, but also has the capabilities to send counters, pictures and more. The description found here is the shortened version of the protocol and only includes the features supported by Information Manager.

EZ KEY II PROTOCOL

The interface and protocol support several types of files and a number of special functions which are used for specific applications. They include:

- **TEXT FILE**. The ASCII message data and display mode information, along with various other control codes, are stored in TEXT files. STRING files may be inserted into a TEXT file.
- **STRING FILE**. The STRING files are used to store ASCII characters only. STRING files are used in applications where a string of frequently changing data must be transmitted to, and displayed by, the message center. Applications include the storage of a number, which changes often, such as ACD call center statistics, a temperature, a quantity, or a timer.
- **SPECIAL FUNCTIONS**. These are functions that give you access to internal registers, diagnostics, and other miscellaneous items.

TRANSMISSION FRAME FORMAT

For more information, see Protocol Examples Section 10 on page 59 This section describes the basic outline of transmissions on an EZ KEY II network.

Table 1

Transmission speed:	1200, 2400,4800 or 9600 baud	
Data bits:	7	8
Start bits:	1 OR	1
Stop bits:	2	2
Parity:	Even	None
Time-out Period:	1 Second (any delays between bytes cannot exceed this)	

All transmissions on the system must appear in the following format, see Transmission Frame variations Section 2.1.1 on page 7.

Table 2

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	Command Code	Data Field	<EOT>

<NUL> (00H): Frame synchronizing character, a minimum of five <NUL>s must be transmitted before the <SOH>. Five <SOH>s may be substituted for the five <NUL>s. The message center will establish the baud rate from the frame synchronizing character.

Table 3

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	Command Code	Data Field	<EOT>

<SOH>(01H): "Start of Header" character

Table 4

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	Command Code	Data Field	<EOT>

Type Code: One ASCII character. Selects the type(s) or model(s) of sign that can receive this transmission frame.

Table 5

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	Command Code	Data Field	<EOT>

Table 6: Unit Type Codes

"Z"(5AH)	ALL Message Centers: Use this type code whenever possible.
"?" (3FH)	ALL Message Centers
"0" (30H)	Response Type Code - Used only when a sign responds to a request.
"!" (21H)	ALL Message Centers with Visual Verification (This code will cause the message centers to give a visual indication that is, "TRANSMISSION OK" on the message center display, of whether it received the transmission frame without error.)
"1" (31H)(1)	One-line message centers
"2" (32H)	Two-line message centers
"#" (23H)	MATRIX products
"$" (24H)	Full matrix MATRIX

"%" (25H)	Character matrix MATRIX
"&" (26H)	Line matrix MATRIX
"a" (61H)	4120C
"b" (62H)	4160C
"c" (63H)	4200C
"d" (64H)	4240C
"e" (65H)(1)	215R
"f" (66H)(1)	215C
"g" (67H)	4120R
"h" (68H)	4160R
"i" (69H)	4200R
"j" (6AH)	4240R
"k" (6BH)(1)	300C SERIES
"l" (6CH)	7000C SERIES
"m" (6DH)	PowerLight 16 Row
"n" (6EH)	PowerLight 24 Row
"U" (55H)(1)	790i
"C" (43H)	7430i
"D" (44H)	7440i
"E" (45H)	7460i
""" (22H)	Serial Clocks
"^" (5EH)	205C

(1) One-line message centers. The remaining units are two-line message centers.

Note: See Unit Type Codes Table 6 on page 2 for addressing multiple units without using wildcards ("?") or broadcast addressing.

Address Field: Two ASCII HEX digits. The address must be in the range (00H) to (FFH)

Table 7

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	Command Code	Data Field	<EOT>

Format = aa;
where a = 1 ASCII HEX digit
"0" -> "9", "A" -> "F"
(30H) -> (39H), (41H) -> (46H) "?" -> wildcard digit

The address selects the sign on the network that will process the transmission frame. The wildcard digit can be used as one of the digits to group message centers or for both digits to form a broadcast address. A "?" combined with a "0" as part of the Address Field is NOT considered a broadcast address. Therefore, address "0?" will only access message centers with address "01H" - "0FH." Address "00" is also reserved as a broadcast address. Anytime a wildcard or broadcast address is used, all message centers with the correct Type Code will process the transmission frame. The response Address Field, when a message center is queried for information, is also "00". This is the address sent back by the message center.

<STX> (02H): "Start of Text" character. This always precedes a Command Code.

Table 8

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	Command Code	Data Field	<EOT>

Command Code: One ASCII character. The Command Code defines the transmission and data types. A summary of the available commands follows:

Table 9

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	Command Code	Data Field	<EOT>

Command Codes

Table 10

"A"	(41H)	Write TEXT file
"B"	(42H)	Read TEXT file
"E"	(45H)	Write SPECIAL FUNCTIONS
"F"	(46H)	Read SPECIAL FUNCTIONS
"G"	(47H)	Write STRING file
"H"	(48H)	Read STRING file
"O"	(4FH)	Write Bulletin Message

Data Field: The Data Field is made up of ASCII characters. The format of the Data Field is dependant upon its associated Command Code. See the proper section for Data Field formats.

Table 11

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	Command Code	Data Field	<EOT>

<EOT> (04H): End of Transmission character.

Table 12

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	Command Code	Data Field	<EOT>

TRANSMISSION FRAME VARIATIONS

See Protocol Examples Section 10 on page 59 for more information.

The transmission frame format has a few acceptable variations which have their own advantages, depending on the application.

A. With Checksum field. If an <ETX> character is transmitted before the <EOT>, the message center will expect a Checksum.

Table 13

<NUL>	<SOH>	Type Code	Addr. Field	<STX>	Command Code	Data Field	<ETX>	Check Sum	<EOT>

<ETX> (03H): "End of Text" character

Checksum: This is a 16 bit hexadecimal summation of all transmitted data from the previous <STX> thru the previous <ETX> inclusive. A Checksum is sent as four ASCII hexadecimal digits, with the most significant digit sent first.

format = cccc;
where c = one ASCII hex digit
= "0" -> "9", "A" -> "F"
= (30H)->(39H), (41H)->(46H)

If an invalid Checksum is received by the message center, the associated data will not be processed.

B. Nesting with Checksums. If more than one transmission frame is required consecutively, the multiple commands can be "nested" within a transmission frame:

Table 14

<NUL>	<SOH>	Type Code	Addr. Field	<STX>	Command Code	Data Field	<ETX>	Check Sum	<STX>

Table 15

Command Code	Data Field	<ETX>	Check Sum	<STX>	Command Code	Data Field	<ETX>	Check Sum	<EOT>

Note: This is the format the message center will follow when a MEMORY DUMP is requested serially.

C. Nesting without Checksums. If an <STX> is transmitted immediately following an <ETX>, the message center will expect the next "nested" command:

Table 16

<NUL>	<SOH>	Type Code	Addr. Field	<STX>	Command Code	Data Field	<ETX>

Table 17

<STX>	Command Code	Data Field	<ETX>	<STX>	Command Code	Data Field	<EOT>

D. Type Code/Address Field Variation:

Table 18

Format = Aaa,Bbb,Ccc,Ddd ...
; where A B C D
= 1 ASCII HEX character representing the Unit Type Code. See UNIT TYPE CODES Table 6 on page 4 for valid values.
; where aa bb cc dd
= 2 ASCII HEX characters representing the Address Field. See ADDRESS FIELD section for valid values.
; where ,
= "," (2CH) acts as a separator between the multiple Type Code/Address Fields

The Type Code/Address Field Variation is used to access multiple message centers without using wildcard or broadcast addressing, see Protocol Examples Section 10 on page 59.

TEXT FILES

The ASCII message data and display mode information, along with various other control codes are stored in Text files. On initial power-up, the message center memory is configured with one Text file (File Label "A"). If multiple Text files are required, see Memory Configuration in Table 45 on page 25 for details.

When writing to a Text file, the display will blank. After the transmission is over, the unit will begin displaying the last received Text file.

When reading from a Text file, the display will either pause or blank depending on the type of message center when it is sending the transmission frame. Once the unit has completely transmitted the file, it will continue displaying the message from where it was interrupted.

As well as containing the actual message, "calls" to other types of files may be inserted into Text files. For example, if you wish to include a String File as part of a Text file, you may simply include a call to a String File in the proper location in your Text file. See String Files Section 4 on page 39 for further information.

WRITE TEXT FILE

Command Character: "A" (41H)
Transmission Frame Format:

Table 19

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	"A" (41H)	File Label	TEXT File Data	<EOT>

File Label: One ASCII character indicating the Text file being accessed. See File Label Format Section 5 on page 43 for descriptions.

Table 20

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	"A" (41H)	File Label	TEXT File Data	<EOT>

TEXT File Data: The contents of a Text file. See Text File Data Format Section 2.2.4 on page 11 for details.

Table 21

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	"A" (41H)	File Label	TEXT File Data	<EOT>

READ TEXT FILE

See Protocol Examples Section 10 on page 59 for more information.
Command Character: "B" (42H)
Transmission Frame Format:

Table 22

<NUL X5	<SOH>	Type Code	Addr. Field	<STX>	"B" (42H)	File Label	<EOT>

File Label: One ASCII character indicating the TEXT file being accessed. See File Label Format Section 5 on page 43 for descriptions.

RESPONSE TO READ TEXT FILE

See Protocol Examples Section 10 on page 59 for more information.

This is the data sent from the message center following a Read Text file. Transmission Frame Format:

Table 23

<NUL> X20	<SOH>	"000"	<STX>	"A" (41H)	File Label	TEXT File Data	<ETX>	Check Sum	<EOT>

Note: Response Type Code and Response Address field "000"

File Label: One ASCII character indicating the Text file being accessed. See File Label Format Section 5 on page 43 for descriptions.

Table 24

<NUL> X20	<SOH>	"000"	<STX>	"A" (41H)	File Label	TEXT File Data	<ETX>	Check Sum	<EOT>

TEXT File Data: The contents of a Text file. See Text File Data Format Section 2.2.4 on page 11 for details.

Table 25

<NUL X20	<SOH>	"000"	<STX>	"A" (41H)	File Label	TEXT File Data	<ETX>	Check Sum	<EOT>

Whenever doing a Read Text file on a network with multiple message centers, it is important that all message centers have their own individual serial address and only one message center is being accessed.

TEXT FILE DATA FORMAT

See Protocol Examples Section 10 on page 59 for more information.

This section outlines the format of the TEXT File Data field. The TEXT file Data is the actual information, which the message center stores in the specified file and displays on the screen. Also, within the TEXT file, will be the modes and control codes, which define the presentation of the message data on the display.

If no mode field is specified at the beginning of the TEXT file Data field, the ASCII message data will run using the default mode (Automode).

The following fields can be repeated within the TEXT file Data field until the TEXT file size limitations are reached (memory varies per model - see your Owner's Manual).

Table 26

<ESC> (1BH)	Display Position	Mode Code	Special specifier (1)	ASCII Message Data
MODE FIELD				

(1) The Special Specifier is only required when the Mode Code is "SPECIAL" ("n").

<ESC> (1BH): Control character which always begins the MODE FIELD. The following two bytes will always be the Display Position byte and the Mode Code.

Table 27

<ESC> (1BH)	Display Position	Mode Code	Special specifier (1)

(1) The Special Specifier is only required when the Mode Code is "SPECIAL" ("n").

Display Position: A code which defines the line position on multi-line message center displays where the ASCII Message Data will appear. On one-line message centers, the Display Position code is irrelevant, but must still be included in the MODE FIELD. Position codes are listed below.

Table 28

<ESC> (1BH)	Display Position	Mode Code	Special specifier (1)

(1) The Special Specifier is only required when the Mode Code is "SPECIAL" ("n").

Table 29

POSITION CODES

"sp" (20H)	Middle Line - Text centered vertically
""" (22H)	Top Line - Text begins on the top line of the display. Will utilize all lines needed to display the text associated with this position up to the last line. For example, a 6 line display allows a maximum of 5 lines for the top position. The Top/Bottom line break will remain fixed until the next Middle or Fill position is specified.
"&" (26H)	Bottom Line - The starting position of the bottom line(s) immediately follows the last line of the top position. For example, a 6 line display with 3 lines of text associated with the top position would start the bottom position text on the 4th line of the display.
"0" (30H)	Fill - message center will fill all available lines of display, centering them vertically.
Mode Code:	All message centers have several different ways of displaying messages, which are referred to as display modes. The Mode Code specifies the type of display mode to be used when the message center presents the ASCII message data on the display. Following are the Mode Codes and a short description of each:

Table 30

<ESC> (1BH)	Display Position	Mode Code	Special specifier (1)

(1) The Special Specifier is only required when the Mode Code is "SPECIAL" ("n").

Table 31

MODE CODES

"a" (61H) - ROTATE	Message travels right to left.
“b" (62H) - HOLD	Message remains stationary.
"c" (63H) - FLASH	Message remains stationary and flashes.
"d" (64H) - RESERVED	
“e" (65H) - ROLL UP	Previous message is pushed up by new message.
"f" (66H) - ROLL DOWN	Previous message is pushed down by new message.
"g" (67H) - ROLL LEFT	Previous message is pushed left by new message.
“h" (68H) - ROLL RIGHT	Previous message is pushed right by new message.

MODE CODES

"i" (69H) - WIPE UP	New message is wiped over the previous message from bottom to top.
"j" (6AH) - WIPE DOWN	New message is wiped over the previous message from top to bottom.
"k" (6BH) - WIPE LEFT	New message is wiped over the previous message from right to left.
"l" (6CH) - WIPE RIGHT	New message is wiped over the previous message from left to right.
"m" (6DH) - SCROLL	New message line pushes the bottom line to the top line if two-line unit.
"o" (6FH) (1) - AUTOMODE	Various modes are called upon to display the message automatically.
"p" (70H) - ROLL IN	Previous message is pushed toward the center of the display by the new message.
"q" (71H) - ROLL OUT	Previous message is pushed outward from the center of the display by the new message.
"r" (72H) - WIPE IN	New message is wiped over the previous message in an inward motion.
"s" (73H) - WIPE OUT	New message is wiped over the previous message in an outward motion.
"t" (74H) - COMPRESSED ROTATE	Message travels right to left. Characters are approximately one half their normal width. Available only on certain models. (See your Owner's Manual.)
"n" (6EH) - SPECIAL	This is followed by a single ASCII character which specifies which of a number of Special modes or graphics will run. They are listed in the SPECIAL MODES and SPECIAL GRAPHICS starting below.

(1) Default setting

Table 32

<ESC> (1BH)	Display Position	"N" (6EH)	Special specifier (1)

(1) The Special Specifier is only required when the Mode Code is "SPECIAL" ("n").

Table 33

SPECIAL MODES

"0" (30H) - TWINKLE	The message will twinkle on the display.
"1" (31H) - SPARKLE	The new message will sparkle on the display over the current message.
"2" (32H) - SNOW	The message will "snow" onto the display.
"3" (33H) - INTERLOCK	The new message will interlock over the current message in alternating rows of dots from each end.
"4" (34H) - SWITCH	Alternating characters "switch" off the display up and down. New message "switches" on in a similar manner.
"5" (35H) - SLIDE	The new message slides onto the display one character at a time from right to left.
"6" (36H) - SPRAY	The new message sprays across and onto the display from right to left.
"7" (37H) - STARBURST	"Starbursts" explode your message onto the display.
"8" (38H) - SCRIPT WELCOME	The word "Welcome" is written in script across the display.
"9" (39H) - SLOT MACHINE	Slot machine symbols randomly appear across the display.

SPECIAL GRAPHICS (1)

"S"(53H) - SCRIPT THANK YOU	The words "Thank You" are written in script across the display.
"U"(55H) - NO SMOKING	A cigarette image appears, is then extinguished and replaced with the universal no smoking symbol.
"V"(56H) - DON"T DRINK AND DRIVE	A car runs into a cocktail glass and is replaced with "Please don’t drink and drive."
"W"(57H) - RUNNING ANIMAL	An animal runs across the display.
"X"(58H) - FIREWORKS	Fireworks explode randomly on the display.
"Y"(59H) - TURBO CAR	A car drives across the display.
"Z"(5AH) - CHERRY BOMB	A bomb fuse burns down followed by an explosion.

1) The Special Graphics are not display modes, therefore, if ASCII message data is to be displayed following a Special Graphic, another mode field is required before the ASCII message data, otherwise the message data will appear in AUTOMODE.

ASCII MESSAGE DATA, see Protocol Examples Section 10 on page 59

Table 34

<ESC> (1BH)	Display Position	Mode Code	Special specifier (1)	ASCII Message Data

(1) The special specifier is only required when the Mode Code is “SPECIAL” (“n”).

ASCII Message Data: Actual characters to be displayed. This field contains ASCII characters, which are shown in the ASCII CHARACTER, and may contain "Control" codes as well, see Control Codes Table 37 on page 17.) The Control codes are used to alter, among other things, the character size, color, and display speed.

Table 35: ASCII CHARACTERS

20H - sp	30H - 0	40H - @	50H - P	60H - ‘	70H - p
21H - !	31H - 1	41H - A	51H - Q	61H - a	71H - q
22H - "	32H - 2	42H - B	52H R	62H - b	72H - r
23H - #	33H - 3	43H C	53H - S	63H - c	73H - s
24H - $	34H - 4	44H - D	54H - T	64H - d	74H - t
25H - %	35H - 5	45H - E	55H - U	65H - e	75H - u
26H - &	36H - 6	46H - F	56H - V	66H - f	76H - v
27H - ’	37H - 7	47H - G	57H - W	67H - g	77H - w
28H - (	38H - 8	48H - H	58H - X	68H - h	78H - x
29H -)	39H - 9	49H - I	59H - Y	69H - i	79H - y
2AH - *	3AH - :	4AH - J	5AH - Z	6AH - j	7AH - z
2BH - +	3BH - ;	4BH - K	5BH - [	6BH - k	7BH - {
2CH - ,	3CH - <	4CH - L	5CH - \	6CH - l	7CH - ¦
2DH - -	3DH - =	4DH - M	5DH -]	6DH - m	7DH - }
2EH - .	3EH - >	4EH - N	5EH - cnt	6EH - n	7EH -
2FH - /	3FH - ?	4FH - O	5FH - _	6FH - o	
sp = space	cnt = cent sign		1/2sp = 1/2 space		

CTL-H(08H) - Extended Character: 2 or 3 byte

The byte following the control code is encoded such that (60H) is added to the ASCII value. This allows selection of characters above (7FH). Example: (80H) is sent as (20H) and (0A6H) is sent as (46H). To select a character above (0D0H) the first CTL-H (08H) if followed by a second CTL-H (08H). The byte following the second control code is encoded such that (80H) is added to the ASCII value. This allows selection of ASCII characters above (0DFH). Example (0E0H) is send as

(60H) and (0F2H) is sent as (72h). The Extended Character code also is used to display temperature in Fahrenheit or Celsius on applicable message center models and to display counter values.

Table 36

EXTENDED CHARACTER SETS

lists the valid characters and their codes:

20H Ç	2DH - ì	3AH - Ü	47H - º	54H - s
21H - ü	2EH - Ä	3BH - ¢	48H - ¿	55H - z
22H - é	2FH - Å	3CH - £	49H -	56H - Z
23H - â	30H - É	3DH - ¥	4AH - ¡	57H - ß
24H - ä	31H - æ	3EH Pt	4BH - sc	58H - S
25H - à	32H - Æ	3FH - ƒ	4CH 0	59H - ß
26H - å	33H - ô	40H - á	4DH - 0	5AH - Á
27H - ç	34H - ö	41H - í	4EH c	5BH - À
28H - ê	35H - ò	42H - ó	4FH - C	5CH - A
29H - ë	36H - û	43H - ú	50H - c	5DH - á
2AH - è	37H - ù	44H - ñ	51H C	5EH - É
2BH - ï	38H - ÿ	45H - Ñ	52H - d	5FH - Í
2CH - î	39H - Ö	46H a	53H - Đ	60H - O
where: sc = single column space				
CTL - \ (1CH)		Temperature display in Celsius (1)		
CTL -] (1DH)		Temperature display in Fahrenheit (1)		

(1) Available on incandescent message centers only (790i, 7430i, 7440i, and 7460i.)

EXT FILE DATA FORMAT

Table 37: Control Codes

CONTROL CODES

CTL-E (2 byte)	(05H)	Double High:	This switch enables or disables the double height character control. Followed by: (1), (2) "0" (30H) - Double height off "1" (31H) - Double height on

CONTROL CODES

CTL-F (2 byte)	(06H)	True Descenders:	This switch will cause characters with descenders (that is, "g" and "y") to be displayed with descenders extended below text line. Followed by: (1), (2) "0" (30H) - True descenders off "1" (31H) - True descenders on
CTL-G (2 byte)	(07H)	Character Flash:	This switch will cause characters to flash. Followed by: (1), (2) "0" (30H) - Character flash off "1" (31H) - Character flash on
CTL-I	(09H)	"No Hold" speed:	When used, there will be virtually no hold time following the mode presentation. This is not applicable for the Rotate or Compressed Rotate modes.
CTL-J	(0AH)	Line feed:	Ignored.
CTL-K	(0BH)	Call Date:	The date will be called up. Followed by a specifier: "0" (30H) - MM/DD/YY "1" (31H) - DD/MM/YY "2" (32H) - MM-DD-YY "3" (33H) - DD-MM-YY "4" (34H) - MM.DD.YY "5" (35H) - DD.MM.YY "6" (36H) - MM DD YY "7" (37H) - DD MM YY "8" (38H) MMM,DD,YYYY "9" (39H) - Day of Week Where: DD = 2 Digit Date MM = 2 Digit Month YY = 2 Digit Year MMM = 3 Character Month Abbr. YYYY = 4 Digit Year
CTL-L	(0CH)	New page:	Start of next display page. NOT SUPPORTED ON ALL UNIT TYPES.
CTL-M	(0DH)	Carriage return:	Start of new line.
CTL-P (2 byte)	(10H)	Call STRING file:	Must be followed by a STRING file label. See String Files Section 4 on page 39 for more information.
CTL-Q(1)	(11H)	Disable wide characters.	
CTL-R	(12H)	Enable wide characters.	

CONTROL CODES

CTL-S	(13H)	Call Time:	The time-of-day will be called up. See Special Functions Section 3 on page 23 for time-of-day setting and time display format selection.
The Length of Time the Characters are Displayed:			
(Also refer to No Hold speed above)			
CTL-U	(15H)	Select Speed 1 (slowest)	
CTL-V	(16H)	Select Speed 2	
CTL-W	(17H)	Select Speed 3	
CTL-X(1)	(18H)	Select Speed 4	
CTL-Y	(19H)	Select Speed 5 (fastest)	
CTL-Z (2 byte)	(1AH)	Select Character Set:	Used to specify which character height or set to be used for the subsequent ASCII characters. Followed by a specifier: "1" (31H) - Five high standard characters "3" (33H) - Seven high standard characters "5" (35H) - Seven high fancy characters "6" (36H) - Ten high standard char. (7000 Series and MATRIX only "8" (38H) - Full height fancy characters "9" (39H) - Full height standard characters
CTL-\ (2 byte)	(1CH)	Select Character Color:	Used to specify the subsequent ASCII character color on all color model message centers. This is followed by a specifier: "1" (31H) - Red "2" (32H) - Green "3" (33H) - Amber "9" (39H) - Rainbow 1 "A" (41H) - Rainbow 2 "B" (42H) - Mix (each char. is a different color) "C" (43H)(1) - Autocolor selection

CONTROL CODES

CTL-Z (2 byte)	(1AH)	Select Character Set:	Used to specify which character height or set to be used for the subsequent ASCII characters. Followed by a specifier: "1" (31H) - Five high standard characters "3" (33H) - Seven high standard characters "5" (35H) - Seven high fancy characters "6" (36H) - Ten high standard char. (7000 Series and MATRIX only "8" (38H) - Full height fancy characters "9" (39H) - Full height standard characters
CTL-\ (2 byte)	(1CH)	Select Character Color:	Used to specify the subsequent ASCII character color on all color model message centers. This is followed by a specifier: "1" (31H) - Red "2" (32H) - Green "3" (33H) - Amber "9" (39H) - Rainbow 1 "A" (41H) - Rainbow 2 "B" (42H) - Mix (each char. is a different color) "C" (43H)(1) - Autocolor selection
Some message center models do not support the full range of colors.			
CTL-] (3 byte)	(1DH)	Select Character Attribute:	Used to specify the character Attributes. This is followed by two specifiers: Specifier 1: "0" (30H) - double stroke "1" (31H) - double wide "2" (32H) - double high "3" (33H) - true descenders "4" (34H) - fixed width "5" (35H) - fancy Specifier 2: "0" (30H) - off "1" (31H) - on
CTL-^ (2 byte)(1)	(1EH)	Select Character Spacing	Used to specify the character spacing. This is followed by a specifier: "0" (30H) - Proportional characters "1" (31H) - Fixed width left justified characters

(1) Default Setting

(2) Not Supported on All Unit Types

PRIORITY TEXT FILE

This is a special 125 byte TEXT file which is preconfigured into all message centers. The transmission frame for accessing the Priority text file follows the text file format. The File Label for the Priority text file is "0" (30H). When data is written to the Priority text file, any file(s) currently running will be interrupted, and the Priority text file will run. The Priority text file will continue to run alone, as it overrides all other text files. The Priority text file will only stop running if any of the following conditions occur:

1. No TEXT File Data (blank file) is sent to the Priority text file.
2. A serial write to the Run Time Table takes place.
3. A serial write to the Run Day Table takes place.
4. Any serial error occurs during the Priority text file write.
5. The message center keyboard PROG (Program) key is pressed.

Once the Priority text file stops running, the message center will begin running the other TEXT files, as it was before the Priority text file was written.

While the Priority text file is running, other files and Special Functions may be written or read serially.

If a tone (speaker) beep is desired when a priority text file is sent, the tone must be sent before the priority text file.

SPECIAL FUNCTIONS

There are a number of special function commands which give the user additional information and control of the message center.

WRITE/READ SPECIAL FUNCTIONS

For more information, see Protocol Examples Section 10 on page 59

WRITE SPECIAL FUNCTIONS COMMAND CHARACTER: "E" (45H) TRANSMISSION FRAME FORMAT:

Table 38

<NUL>	<SOH>	Type Code	Addr. Field	<STX>	"E" (45H)	S.F.Label	S.F. Data	<EOT>

S.F. Label: One ASCII character indicating the Special Function being accessed. See Special Functions Data Formats Section 3.1.4 on page 24.

Table 39

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	"E" (45H)	S.F.Label	S.F. Data	<EOT>

S.F. Data: This must follow the data format outlined for each of the special functions. See Special Functions Data Formats Section 3.1.4 on page 24.

Table 40

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	"E" (45H)	S.F.Label	S.F. Data	<EOT>

READ SPECIAL FUNCTIONS COMMAND CHARACTER: "F" (46H) TRANSMISSION FRAME FORMAT:

Table 41

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	"F" (46H)	S.F.Label	<EOT>

S.F. Label: One ASCII character indicating the Special Function being accessed. See Special Functions Data Formats Section 3.1.4 on page 24..

Table 42

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	"F" (46H)	S.F.Label	<EOT>

RESPONSE TO READ SPECIAL FUNCTIONS

This the data sent from the message center following a Read Special Functions.

Transmission Frame Format:

Table 43

<NUL> X20	<SOH>	"000"	<STX>	"E" (45H)	S.F. Label	S.F. Data	<ETX>	Check Sum	<EOT>

Note: Response Type Code and Response Address Field "000" where:

S.F. Label: One ASCII character indicating the Special Function being accessed. See Special Functions Data Formats Section 3.1.4 on page 24.

Table 44

<NUL> X20	<SOH>	"000"	<STX>	"E" (45H)	S.F. Label	S.F. Data	<ETX>	Check Sum	<EOT>

S.F. Data: This must follow the data format outlined for each of the special functions. See Special Functions Data Formats Section 3.1.4 on page 24.

Whenever doing a Read Special Functions on a network with multiple message centers, it is important that all message centers have their own individual serial address and only one message center is being accessed.

SPECIAL FUNCTIONS DATA FORMATS

The Special Functions Label and description is in bold. The access status follows in parenthesis. Each Special Functions Data format is below.

Table 45: Special Functions Data Formats

" " (20H)	Time-of-day Setting (Read/Write) (see Protocol Examples Section 10 on page 59)
	data = HhNn (24 hour format)
	H = one ASCII digit representing hours (ten's digit)
	h = one ASCII digit representing hours (one's digit)
	N = one ASCII digit representing minutes (ten's digit)
	n = one ASCII digit representing minutes (one's digit)
	
	The Time-of-day is the message center's internal clock. See Control Codes Table 37 on page 17 for Clock Display, and see Time Display Format in the Special Functions Section 3

"!" (21H)	Speaker Status (Read/Write) (see Protocol Examples Section 10 on page 59) data = SS SS = "00" - Two ASCII hex characters representing speaker enabled “FF" - Two ASCII hex characters representing speaker disabled If the Speaker Status is disabled, the Speaker Tone Generation will not function. This applies only to message centers with speaker capability. The Speaker Status is reset to its default value upon power-up. For producing a speaker tone, see the Speaker Tone Generation portion of the Special Functions Section 3 on page 23).

Table 45: Special Functions Data Formats

""" (22H)	General Information (Read Only) (see Protocol Examples Section 10 on page 59) data = <NUL>FFFFFFFFfMmYyHhNnRSSPOOL,pool (28 or 29 ASCII characters total) <NUL> = (00H)(1) FFFFFFFF = The firmware (EPROM) chip, Spectrum Corporation's part number f = The firmware revision letter MmYy = The firmware release date, (M-ten's digit month, m-one's digit month, Y-ten's digit year, y-one's digit year) HhNn - Time-of-day where: H = One ASCII digit representing hours (ten's digit) h = One ASCII digit representing hours (one's digit) N = One ASCII digit representing minutes (ten's digit) n = One ASCII digit representing minutes (one's digit) R(1) = Time Display Format where: "S" (53H) = Standard a.m./p.m. format "M" (4DH) = 24 hour (military) format for information SS = Speaker Status "00" = Two ASCII hex characters representing speaker enabled "FF" = Two ASCII hex characters representing speaker disabled POOL, pool = Memory Pool where: POOL = Four digit ASCII hexadecimal number representing the total size of the memory "POOL" in bytes. The most significant digit is first. "," = (2CH) comma pool = Four digit ASCII hexadecimal number representing the size of the unused portion of the memory "pool" in bytes. The most significant digit is first. General Information reading is most useful to obtain a firmware chip number and revision for troubleshooting purposes.
"#" (23H)	Memory Pool (Read Only) (example) data = POOL,pool POOL = Four digit ASCII hexadecimal number representing the total size of the memory "POOL" in bytes. The most significant digit is first "," = (2CH) comma pool = Four digit ASCII hexadecimal number representing the size of the unused portion of the memory "pool" in bytes. The most significant digit is first. The "POOL" is the amount of battery backed RAM available for file storage. Any unused memory is assigned to the first text file listed in the Memory Configuration once the sign begins running.

Table 45: Special Functions Data Formats

"$" (24H)

Memory Configuration (Read/Write) (see Protocol Examples Section 10 on page 59) (Memory Clear)

data = FTPSIZEQQQQ (11 ASCII characters)

This data field repeats for each file configured in the message center.

Note: If the data field is left blank when writing the Memory Configuration, the message center will reboot with the virgin Memory Configuration (all power-up diagnostics will take place) and all files will be lost (destructive).

The Memory Configuration is really the message center's internal directory of RAM. A file cannot be written unless it's first created by writing a new Memory Configuration. Whenever a Memory Configuration is written, it overwrites the previous one. It does NOT append to the current Memory Configuration.

F = One ASCII character representing the File Label. See File Label Format Section 5 on page 43 for descriptions.

T = One ASCII character representing the file type. Valid entries: "A" (41H) - TEXT file

"B" (42H) - STRING file

P = One ASCII character representing the keyboard protection status. Valid entries are shown below:

"U"(55H) - Unlocked - This allows the file to be accessible from the handheld keyboard.

"L"(4CH) - Locked - This makes the file in-accessible from the handheld keyboard.

Note: String files require a locked protection status.

SIZE = Four ASCII hexadecimal characters representing the size of the file in bytes for Text and STRING files. It is necessary that STRING files not exceed

125 bytes in length ("007D").

The summation of all the file plus eleven bytes of overhead for each file should not exceed the total amount of available memory in the pool.

"0000" is a valid size entry for the last file in the Memory Configuration if it is a

TEXT file. This will assign all remaining available memory to the file.

QQQQ = Four ASCII hexadecimal characters which carry different meaning for each of the file types. They are detailed below:

QQQQ (text file) = The first two characters represent the file's run Start Time. The second two characters represent the file's run Stop Time. See Text File Start and Stop Times Section 6 on page 45 for the table of valid Start/Stop time values.

QQQQ (String File) = "0000" - four ASCII "0"s which carry no special meaning

Table 45: Special Functions Data Formats

"%" (25H)	Memory Dump (Read Only) (see Protocol Examples Section 10 on page 59) data = multiple nested transmission frames with Checksums in the following order: 1. Time-of-Day Setting 2. Memory Configuration 3. The Transmission frame of each file in order as they appear in the Memory Configuration (Write text or string file) 4. Run Sequence 5. Run Day Table 6. Day-of-Week Setting Refer to the appropriate section for format details on each of the above transmission frames.
"&" (26H)	Day-of-Week setting (Read/Write) (see Protocol Examples Section 10 on page 59) data = D One ASCII digit representing the day of the week. This is automatically updated by the message center at 12:00 midnight everyday. Valid entries: "1" (31H) = Sunday "2" (32H) = Monday "3" (33H) = Tuesday "4" (34H) = Wednesday "5" (35H) = Thursday "6" (36H) = Friday “7" (37H) = Saturday
’" (27H)	Time Display Format (Read/Write) (see Protocol Examples Section 10 on page 59) data = One ASCII character representing the time format and how it is displayed by the message center. Valid entries are: "S" (53H)(2) = Standard a.m./p.m. format "M" (4DH) = 24 hour (military) format

Table 45: Special Functions Data Formats

“(“ (28H)	Speaker Tone Generation (Write Only) (see Protocol Examples Section 10 on page 59) data = B(3) B = One ASCII character which generates a tone from the speaker. This must be the last transmission frame when sending nested frames. The message center serial port is disabled while the tone is being generated. Therefore, this cannot be part of a transmission containing any type of "read" command. Valid entries are listed below: "A" (41H) - Turn speaker port "on."(4) "B" (42H) - Turn speaker port "off."(4) "0" (30H) - Generate continuous tone for approximately two seconds. "1" (31H) - Generate three short beeps, total time approximately two seconds. "2" (32H) - Generate programmable tone data = FFDR FF - Two ASCII hex characters representing the desired speaker frequency. Valid entry range = 01H" thru FEH" D - One ASCII hex character representing the tone duration in 0.1 second increments. Valid entry range = 1" thru "F". R - One ASCII hex character representing the number of times to repeat the tone. Valid entry range = "0" thru "F". Run Time Table (Read/Write) (see Protocol Examples Section 10 on page 59)
“)" (29H)	data = FQQQQ (Write) or data = LqqqqFQQQQE (Read) Repeating portion when the Run Time Table is Read.
(Write)	This five byte data field repeats for each TEXT file configured in the message center. Not all TEXT files need to be updated, only those that require modification. F = One ASCII character representing the TEXT File Label. See File Label Format Section 5 on page 43 for valid File Labels. QQQQ = Four ASCII hexadecimal characters. The first two characters represent the file’s run Start Time. The second two characters represent the file’s run Stop Time. See Text File Start and Stop Times Section 6 on page 45 for the table of valid Start/Stop time values. These will overwrite what is in the Memory Configuration. (destructive)

Table 45: Special Functions Data Formats

(Read)	The first five bytes of this field represent the Priority Text file status. They are described below: L = "0"(30H) Represents the Priority Text file Label. qqqq = Four ASCII hexadecimal characters which show the Priority Text file status. There are only two possibilities for this: "FE00"(2) - PRIORITY TEXT file "not running" "FF00" - PRIORITY TEXT file "running" This following six byte data field repeats for each Text file configured in the message center (with the exception of the Priority Text file which preceded this field). F = One ASCII character representing the Text File Label. See File Label Format Section 5 on page 43 for valid File Labels. QQQQ = Four ASCII hexadecimal characters. The first two characters represent the file's run Start Time. The second two characters represent the file's run Stop Time. See Text File Start and Stop Times Section 6 on page 45 for the table of valid Start/Stop time values. E = One ASCII hexadecimal character which gives the file enable status. Valid codes are shown below: "1" - The file is currently being displayed "0" - The file is not currently being displayed
"*" (2AH)	Serial Error Status (Read Only) (see Protocol Examples Section 10 on page 59) data = Z Z = One ASCII character representing the serial errors recorded by the message center. This register is reset to its default value only upon message center power-up, or after the Error Status is read serially by either a Serial Error Status read or a Network Query. It is also cleared serially when a Clear Serial Error Status write is done. When a serial error occurs, the appropriate bit in the Error Status register is set. The message center begins error checking following a valid <SOH> (01). The bit designations are listed below: b7 - Always cleared (0) b6 - Always set (1) b5 - Illegal Command Code, File Label, illegal read, or write Special Functions b4 - Serial Checksum error b3 - Insufficient serial buffer space (overflow) b2 - Serial time-out (time-out period exceeded) b1 - Bit framing error (incorrect baud rate) b0 - Parity error (not even Parity) The default Serial Error Status value is "@" (40H or 01000000B).

Table 45: Special Functions Data Formats

"," (2CH)	Soft Reset (Write Only) (see Protocol Examples Section 10 on page 59) data = none There is no data in this data field. Writing this will initialize the message center. The message center will go through all of its power-up diagnostics, as if power was just applied. Memory will not be cleared (non-destructive).
"-" (2DH)	Network Query (Read Only) (see Protocol Examples Section 10 on page 59) data = UAAZ U = One ASCII character representing the unit type. See Unit Type Codes Table 6 on page 4 for valid entries. AA = Two ASCII hexadecimal characters representing the unit's serial address. Z = One ASCII character representing the serial errors recorded by the message center. See Serial Error Status portion of Special Functions Section 3 on page 23, for further details. The response is a timed response. Normally, this is transmitted with a broadcast address ("00") in the Address Field. All units on the network will then respond in the following manner: Once the <EOT> is received by the units, they will then respond at timed intervals of one second plus the product of it's address and 0.50 seconds. See example below: A message center with the address 08 will reply after 1 + (8 x 0.50) = 5 seconds. All message centers on the network will blank once the <EOT> is sent. Once a unit has responded, it will resume normal operation.

Table 45: Special Functions Data Formats

"." (2EH)	Run Sequence (Read/Write) (see Protocol Examples Section 10 on page 59) data = KPF ¦ Repeating portion K = One ASCII character representing the Run Sequence key code. Valid entries are shown below: "T"(54H)(2) - All subsequent Text File Labels in the run sequence will run, in order, according to the file's associated run times. "S"(53H) - All subsequent Text File Labels in the run sequence will run, in order, regardless of the file's associated run times. P = One ASCII character representing the keyboard protection status. Valid entries are shown below: "U" (55H)(2) - Unlocked - This allows the run sequence to be accessible from the hand-held keyboard. "L" (4CH) - Locked - This makes the run sequence inaccessible from the hand-held keyboard. F = One ASCII character representing a Text File Label. This should be a label of a valid Text file. If a label is used for a Text file that does not exist or is invalid, the next File Label will be processed. There can be a maximum of 128 Text File Labels in the Run Sequence.
"/" (2FH)	Dimming Control (Write Only) (Available on outdoor models only) (see rotocol Examples Section 10 on page 59) data = WWww WW = Two ASCII hexadecimal characters representing the start time for the dimming of the display. ww = Two ASCII hexadecimal characters representing the stop time for the dimming of the display. See Text File Start and Stop Times Section 6 on page 45 for the table of valid Start/Stop time values. Time codes 0FDH, 0FEH, and 0FFH are invalid codes for Dimming Control. If Dimming is not desired, set WWww = 0000. This is the default value. Note: Dimming Control is only available on incandescent message center models 790i, 7430i, 7440i, and 7460i.

Table 45: Special Functions Data Formats

"2"(32H)	Run Day Table (Read/Write) (see Protocol Examples Section 10 on page 59) data = FSs, where FS is Repeating field F = One ASCII character representing the Text File Label. See File Label Format Section 5 on page 43 for valid File Labels. S = One ASCII hexadecimal character representing the Text file run start day. Valid start day codes are listed below: "0" (30H)(2) = Daily "1" (31H) = Sunday "2" (32H) = Monday "3" (33H) = Tuesday "4" (34H) = Wednesday "5" (35H) = Thursday "6" (36H) = Friday "7" (37H) = Saturday "8" (38H) = Monday-Friday "9" (39H) = Weekends "A" (41H) = Always "B" (42H) = Never s = One ASCII hexadecimal character representing the Text file run stop day. Valid stop day codes are listed below: "1" (31H) = Sunday "2" (32H) = Monday "3" (33H)(2) = Tuesday "4" (34H) = Wednesday "5" (35H) = Thursday ”6" (36H) = Friday "7" (37H) = Saturday If the start day covers multiple days (that is, daily, never, and so on) the stop day is ignored, but still required.
"4" (34H)	Clear Serial Error Status (Write Only) (see Protocol Examples Section 10 on page 59) data = none This command provides a means of initializing the Serial Error Status to its default value. This is useful as the first command in a nested transmission frame to be sure that all subsequent serial errors or lack of serial errors recorded are applicable to that nested transmission frame. The last command in the nested transmission frame should then be a Serial Error Status read.

Table 45: Special Functions Data Formats

";" (3BH)	Date setting (Read/Write) data = mmddyy mm - Two ASCII digits representing the month. dd – Two ASCII digits representing the day. yy - Two ASCII digits representing the year.
"7"(37H)	Serial Address (Write Only) data = AA AA = Two ASCII hexadecimal characters representing the desired serial address for the MATRIX sign. If the serial address has been set using the hardware dipswitches to an address other than "00," the dipswitch address will override the serially configured serial address upon power-up. The serially configured serial address is stored in battery backed RAM.
"T" (54H)	Temperature Offset (Read/Write) Allows for improvement in temperature accuracy as displayed on message centers which support temperature display. data = SO S = One ASCII character representing the sign of the temperature offset. Valid values are "+" (2BH) and "-" (2DH). O = One ASCII hexadecimal character representing the temperature offset. Valid values are "0" through "9." Temperature Offset only applies on incandescent message center Models 790i, 7430i, 7440i, and 7460i

(1) This byte is transmitted only on some message center models. (2) Default setting

(3) Since the serial port is disabled while the message center is generating a tone (either "0" or "1"), wait a minimum of approximately three seconds before the next transmission. When generating the programmable tone ("2"), no transmissions should occur to sign until the sign has completed its tone generation.

(4) This is not to be used with the standard speaker/peizo alarm which is provided inside the message center, as it may cause damage. This is only to be used when using the speaker port to drive an auxiliary device.

STRING FILES

String files are used to display, on the wallboards, rapidly changing information, such as ACD Statistics or Production counts. String files store ASCII characters which are “called up” from Text files. When writing and calling a String file

the wallboard will not blank as it does when writing a Text file. This happens because the String file data is buffered and the Text file internal checksum does not change.

Note: Because the String file data is buffered, the String file size is limited to 125 bytes

Before being able to write to a String file, memory must be allocated for the String file in the message center. See Memory Configuration in Table 45 on page 25 for details.

String files are "called up" from Text files utilizing the Text file control code designated for a "Call STRING file." See Control Codes Table 37 on page 17 for further information.

Wen reading from a String file, the display will either pause or blank, depending on the type of message center, when it is sending the transmission frame. Once the unit has completely transmitted the file, it will continue displaying the message from where it was interrupted.

String file application notes are located in Sample C Program Section 8 on page 51.

WRITE/READ STRING FILE

WRITE STRING FILE

Command Character: "G" (47H) Transmission Frame Format:

Table 46

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	"G" (47H)	File Label	STRING File Data	<EOT>

File Label: One ASCII character indicating the String file being accessed. See File Label Format Section 5 on page 43 for File Label descriptions.

Table 47

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	"G" (47H)	File Label	STRING File Data	<EOT>

String File Data: The contents of a String file. See String File Data Format ection 4.1.4 on page 41 for details.

READ STRING FILE

Command Character: "H" (48H) Transmission Frame Format:

Table 48

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	"H" (48H)	File Label	STRING File Data	<EOT>

File Label: One ASCII character indicating the String file being accessed. See

File Label Format Section 5 on page 43 for File Label descriptions.

Table 49

<NUL> X5	<SOH>	Type Code	Addr. Field	<STX>	"H" (48H)	File Label	<EOT>	<NUL>

RESPONSE TO READ STRING FILE

Transmission Frame Format for data sent following a Read String file.

Table 50

<NUL> X20	<SOH>	"000"	<STX>	"G" (47H)	File Label	STRING File Data	<ETX>	Check Sum	<EOT>

Note: Response Type Code and Response Address Field "000"

File Label: One ASCII character indicating the String file being accessed. See

File Label Format Section 5 on page 43 for File Label descriptions.

Table 51

<NUL> X20	<SOH>	"000"	<STX>	"G" (47H)	File Label	STRING File Data	<ETX>	Check Sum	<EOT>

STRING File Data: The contents of a String file. See String File Data Format Section 4.1.4 on page 41 for details .

Table 52

<NUL> X20	<SOH>	"000"	<STX>	"G" (47H)	File Label	STRING File Data	<ETX>	Check Sum	<EOT>

Whenever doing a Read String file on a network with multiple message centers, it is important that all message centers have their own individual serial address and only one message center is being accessed.

STRING FILE DATA FORMAT

This section outlines the format of the String File Data field. The String File Data is the actual data which the message center stores in the specified file and displays on its screen when its "called" from a Text file. With a few exceptions, the only acceptable data that String files will accept can be found in ASCII CHARACTER Table 35 on page 15. See Control Codes Table 37 on page 17 for further definition of the following control codes which are acceptable within a String file. All other control codes are NOT acceptable.

Table 53

CTL-I (09H)	"No Hold" Speed Should be used for Real-time ACD and Production applications
CTL-M (0DH)	Carriage Return
CTL-Q (11H)(1)	Disable Wide Characters
CTL-R (12H)	Enable Wide Characters
CTL-S (13H)	Call Time
CTL-U (15H)	Select Speed 1
CTL-V (16H)	Select Speed 2
CTL-W (17H)	Select Speed 3
CTL-X (18H)(1)	Select Speed 4
CTL-Y (19H)	Select Speed 5
CTL-Z (1AH)	Select Character Set
CTL-\ (1CH)	Select Character Color (Rainbow color selection does not function within STRING files.
CTL-^ (1EH)	Select Character Spacing

(1) Default setting

FILE LABEL FORMAT

A File Label is a single ASCII character. Messages are stored in or retrieved from the memory file that is defined by this label in the **Memory Configuration**. Legal File Labels can be anywhere in the range "sp" (20H) thru "res" (7FH) inclusive. The only special case occurs when File Label "0" (30H) is used for a Text file. This is an illegal label for a Text file in the **Memory Configuration**.

It is already configured as a set portion of memory outside of the **Memory Pool**, as a **Priority Text** file.

See Priority Text File Section 2.2.6 on page 21 for further information.

Table 54

20H - sp	30H - 0	40H - @	50H - P	60H - ‘	70H - p
21H - !	31H - 1	41H - A	51H - Q	61H - a	71H - q
22H - "	32H - 2	42H - B	52H R	62H - b	72H - r
23H - #	33H - 3	43H - C	53H - S	63H - c	73H - s
24H - $	34H - 4	44H - D	54H - T	64H - d	74H - t
25H - %	35H - 5	45H - E	55H - U	65H - e	75H - u
26H - &	36H - 6	46H - F	56H - V	66H - f	76H - v
27H - ’	37H - 7	47H - G	57H - W	67H - g	77H - w
28H - (	38H - 8	48H - H	58H - X	68H - h	78H - x
29H -)	39H - 9	49H - I	59H - Y	69H - i	79H - y
2AH - *	3AH - :	4AH - J	5AH - Z	6AH - j	7AH - z
2BH - +	3BH - ;	4BH - K	5BH - [	6BH - k	7BH - {
2CH -,	3CH - <	4CH - L	5CH - \	6CH - l	7CH - ¦
2DH - -	3DH - =	4DH - M	5DH -]	6DH - m	7DH - }
2EH - .	3EH - >	4EH - N	5EH - cnt	6EH - n	7EH -1/2sp
2FH - /	3FH - ?	4FH - O	5FH - _	6FH - o	7FH - res
	sp = space 1/2sp = 1/2 space cnt = cent sign				

TEXT FILE START AND STOP TIMES

Table 55

12:00 a.m. - 00H	8:00 a.m. - 30H	4:00 p.m. - 60H
12:10 a.m. - 01H	8:10 a.m. - 31H	4:10 p.m. - 61H
12:20 a.m. - 02H	8:20 a.m. - 32H	4:20 p.m. - 62H
12:30 a.m. - 03H	8:30 a.m. - 33H	4:30 p.m. - 63H
12:40 a.m. - 04H	8:40 a.m. - 34H	4:40 p.m. - 64H
12:50 a.m. - 05H	8:50 a.m. - 35H	4:50 p.m. - 65H
1:00 a.m. - 06H	9:00 a.m. - 36H	5:00 p.m. - 66H
1:10 a.m. - 07H	9:10 a.m. - 37H	5:10 p.m. - 67H
1:20 a.m. - 08H	9:20 a.m. - 38H	5:20 p.m. - 68H
1:30 a.m. - 09H	9:30 a.m. - 39H	5:30 p.m. - 69H
1:40 a.m. - 0AH	9:40 a.m. - 3AH	5:40 p.m. - 6AH
1:50 a.m. - 0BH	9:50 a.m. - 3BH	5:50 p.m. - 6BH
2:00 a.m. - 0CH	10:00 a.m. - 3CH	6:00 p.m. - 6CH
2:10 a.m. - 0DH	10:10 a.m. - 3DH	6:10 p.m. - 6DH
2:20 a.m. - 0EH	10:20 a.m. - 3EH	6:20 p.m. - 6EH
2:30 a.m. - 0FH	10:30 a.m. - 3FH	6:30 p.m. - 6FH
2:40 a.m. - 10H	10:40 a.m. - 40H	6:40 p.m. - 70H
2:50 a.m. - 11H	10:50 a.m. - 41H	6:50 p.m. - 71H
3:00 a.m. - 12H	11:00 a.m. - 42H	7:00 p.m. - 72H
3:10 a.m. - 13H	11:10 a.m. - 43H	7:10 p.m. - 73H
3:20 a.m. - 14H	11:20 a.m. - 44H	7:20 p.m. - 74H
3:30 a.m. - 15H	11:30 a.m. - 45H	7:30 p.m. - 75H
3:40 a.m. - 16H	11:40 a.m. - 46H	7:40 p.m. - 76H
3:50 a.m. - 17H	11:50 a.m. - 47H	7:50 p.m. - 77H
4:00 a.m. - 18H	12:00 p.m. - 48H	8:00 p.m. - 78H

4:10 a.m. - 19H	12:10 p.m. - 49H	8:10 p.m. - 79H
4:20 a.m. - 1AH	12:20 p.m. - 4AH	8:20 p.m. - 7AH
4:30 a.m. - 1BH	12:30 p.m. - 4BH	8:30 p.m. - 7BH
4:40 a.m. - 1CH	12:40 p.m. - 4CH	8:40 p.m. - 7CH
4:50 a.m. - 1DH	12:50 p.m. - 4DH	8:50 p.m. - 7DH
5:00 a.m. - 1EH	1:00 p.m. - 4EH	9:00 p.m. - 7EH
5:10 a.m. - 1FH	1:10 p.m. - 4FH	9:10 p.m. - 7FH
5:20 a.m. - 20H	1:20 p.m. - 50H	9:20 p.m. - 80H
5:30 a.m. - 21H	1:30 p.m. - 51H	9:30 p.m. - 81H
5:40 a.m. - 22H	1:40 p.m. - 52H	9:40 p.m. - 82H
5:50 a.m. - 23H	1:50 p.m. - 53H	9:50 p.m. - 83H
6:00 a.m. - 24H	2:00 p.m. - 54H	10:00 p.m. - 84H
6:10 a.m. - 25H	2:10 p.m. - 55H	10:10 p.m. - 85H
6:20 a.m. - 26H	2:20 p.m. - 56H	10:20 p.m. - 86H
6:30 a.m. - 27H	2:30 p.m. - 57H	10:30 p.m. - 87H
6:40 a.m. - 28H	2:40 p.m. - 58H	10:40 p.m. - 88H
6:50 a.m. - 29H	2:50 p.m. - 59H	10:50 p.m. - 89H
7:00 a.m. - 2AH	3:00 p.m. - 5AH	11:00 p.m. - 8AH
7:10 a.m. - 2BH	3:10 p.m. - 5BH	11:10 p.m. - 8BH
7:20 a.m. - 2CH	3:20 p.m. - 5CH	11:20 p.m. - 8CH
7:30 a.m. - 2DH	3:30 p.m. - 5DH	11:30 p.m. - 8DH
7:40 a.m. - 2EH	3:40 p.m. - 5EH	11:40 p.m. - 8EH
7:50 a.m. - 2FH	3:50 p.m. - 5FH	11:50 p.m. - 8FH
ALL DAY - 0FDH	NEVER - 0FEH	ALWAYS - 0FFH

STRING FILE APPLICATION NOTES

String File Definition: A String file, as it applies to the EZ KEY II Protocol, is a short stream of data that is "called" from a Text file.

A typical application of String files involves the updating of a count that is continuously displayed on a message center, for example, ACD Call Center Statistics or Real-time Production updates.

One advantage of using String files to update some displayed data is that the wallboard won’t "blink" or flash during the update, as it will during the updating of Text files. Also when using String file, information is updated faster than rewriting an entire Text file. Another advantage is that it is a saver of memory space. For example; if some important data is displayed multiple times within a Text file, it need only be stored once as a String file, then "called" from the appropriate location within the Text file.

To implement String files, there are three essential steps:

1. Allocate memory within the message center unit for the String file (and the Text file from which it is called).
2. Write the Text file which calls the String file.
3. Update the String file.

To allocate memory for one String file and the Text file from which it is called, send the data stream below. This is also called a memory configuration. The Address Field is set up to talk to all signs on your network. For example:

Table 56

<NUL> X5	<SOH>	"ZOO"	<STX>	"E$AAU0400F 001BL 00200000"	F<EOT>

where:

<NUL>	(00H)	Five of them are required by the message center to lock on to the baud rate (sometimes called autobauding
<SOH>	(01H)	"Start of Header" character
"Z00"	(5AH,30H,30H)	Unit Type Code/Address Field
<STX>	(02H)	"Start of Text" character
"E"	(45H)	Write Special Functions Command Code
"$"	(24H)	Special Functions label for Memory Configuration (directory)
"A"	(41H)	File Label
"A"	(41H)	Text file type

"U"	(55H)	"Unlocked" keyboard status
"0400"	(30H,34H,30H,30H)	Text file size in bytes (hexadecimal or 1024 decimal)
"FF"	(46H,46H)	Text file run start time ("FF" represents "always")
"00"	(30H,30H)	Text file run stop time (ignored when start time is "always")
"1"	(31H)	File Label
"B"	(42H)	String file type
"L"	(4CH)	"Locked" keyboard status
"0020"	(30H,30H,32H,30H)	String file size in bytes (hexadecimal or 32 decimal)
(30H,30H)		Ignored
"00"	(30H,30H)	Ignored
<EOT>	(04H)	"End of Transmission" character

To write the Text file which calls the String file, see Table 57on page 38.

Table 57

<NUL> X5	<SOH>	"ZOO"	<STX>	“AA”, <ESC>,”bTh count is “, <DLE>, “1”	<EOT> e

where:		
<NUL>	(00H)	Five of them are required by the message center to lock on to the baud rate (sometimes called autobauding)
<SOH>	(01H)	"Start of Header" character
"Z00"	(5AH,30H,30H)	Unit Type Code/Address Field
<STX>	(02H)	"Start of Text" character
"A"	(41H)	Write Text File
"A"	(41H)	Text File Label
<ESC>	(1BH)	Signifies the start of a mode field
"b"	(20H,62H)	Space is the middle line position, "b" is the "HOLD" mode code

<NUL> X5	<SOH>	"ZOO"	<STX>	"AA", <ESC>,"bTh count is ", <DLE>, "1"	<EOT> e
"The count is"		(54H,68H,65H,20H,63H ,6FH,75H,6EH,74H ,20H,69H,73H,20H)		Text File Data	
<DLE>		(10H)		String file call	
"1"		(31H)		String File Label	
<EOT>		(04H)		"End of Transmission" character	

To update the String file, see Table 58 on page 49.

Table 58

<NUL>	<SOH>	"ZOO"	<STX>	"G1364"	<EOT>
where:					
<NUL>		(00H)		five of them are required by the message center to lock on to the baud rate (sometimes called autobauding)	
<SOH>		(01H)		"Start of Header" character	
"Z00"		(5AH,30H,30H)		Unit Type Code/Address Field	
<STX>		(02H)		"Start of Text" character	
"G"		(47H)		Write STRING file Command Code	
"1"		(31H)		String File Label	
"364"		(33H,36H,34H)		String File Data	
<EOT>		(04H)		"End of Transmission" character	

To update the String File Data regularly, repeat step 3 above with changing String File Data. The message center will display the following data by utilizing the previous 3 step example: "The count is 364"

A few things to keep in mind:

1. The default character spacing is proportional width, rather than fixed width. Thus, when constantly changing String files are updated, and different width characters are sent, the message center's auto-centering will move the displayed data around with the changing character widths, in an effort to keep the data centered. There are two things to do to avoid this from happening, since this is distracting to the viewer.

a. Always send the same number of characters in the String File Data. b Always use fixed width characters by embedding the following 2 byte sequence in your Text file before the String file "call": CTL-^,"1" (1EH,31H)

2. The maximum file size for a String file is 125 bytes. Do not exceed this.

SAMPLE C PROGRAM

```
/****************************************************************

* Program Name.......... SIMPLE C NETWORK PROGRAM NO LIBRARIES

* Filename .................. SIMPLEC.C

* Version .................... 1.0

* Version Date ............ February 27, 1991

* Comments ............... none

*

* COPYRIGHT (C) 1991.  All Rights Reserved.

* Spectrum Corporation, Inc.  Houston, Tx.

*
****************************************************************/

#define PORT_SETUP 0xde /* = 4800 baud */

/*

#define PORT_SETUP 0x9e /* = 1200 baud */

#define PORT_SETUP 0xbe /* = 2400 baud */

#define PORT_SETUP 0xde /* = 4800 baud */

#define PORT_SETUP 0xfe /* = 9600 baud */

*/

#define COM_PORT 0 /* = com port 1 */

/*

#define COM_PORT 0 /* = com port 1 */

#define COM_PORT 1 /* = com port 2 */

*/
struct WORDREGS {
```

```
unsigned int ax, bx, cx, dx, si, di, cflag, flags;

};

struct BYTEREGS {

unsigned char al, ah, bl, bh, cl, ch, dl, dh;

};

union REGS {

struct WORDREGS x;

struct BYTEREGS h;

};

main()

{ int x;

/* open the com port */

serinit();

/* send 20 nulls */

for (x = 0; x < 20; x++) outc(0,COM_PORT);
outc(0x01,COM_PORT); /* send a SOH */
outc("Z",COM_PORT); /* send the sign type (Z
= all signs, c = 4200C */
outc("0",COM_PORT); /* send the address (00 = all signs) */

outc("0",COM_PORT);

outc(0x02,COM_PORT); /* send a STX */

outc("A",COM_PORT); /* send the command "WRITE TEXT file" */

outc("A",COM_PORT); /* send TEXT File Label to write to

(A = default) */

outc(0x1b,COM_PORT); /* send an escape

(precedes all mode commands) */
outc(0x20,COM_PORT); /* send a position code

(0x20 = middle full height) */

outc("b",COM_PORT); /* send a mode (b = hold) */

outs("HELLO",COM_PORT); /* send out the string of characters */

outc(0x04,COM_PORT); /* send out the EOT to end the transmission */

return(0);

}

/* function that outputs a string to the com port */

outs (unsigned char *s,int port)

{

while (*s)

outc(*s++,port);

return(0);

}

/* function that outputs a char to the com port */

outc (unsigned char c,int port)

{

union REGS regs;

regs.h.ah = 01;

regs.h.al = c;

regs.x.dx = port;

int86(0x14,&regs,&regs); /* Turbo C function which triggers the serial interrupt. Check
compiler for similar function */

return(0);

}

/* function which opens the com port */

serinit()
```

```
{
union REGS regs;
regs.h.ah = 0;
regs.h.al = PORT_SETUP;
regs.x.dx = COM_PORT;
int86(0x14,&regs,&regs);
return(0);
}
```

SAMPLE BASIC PROGRAM

SAMPLE BASIC PROGRAM-WRITE TEXT FILE A

```
10 CLS:PRINT"SPECTRUM NETWORK INSTALL PROGRAM":PRINT: PRINT: INPUT
"COMMUNICATION PORT (1 OR 2) :";A$
20 IF A$ = "1" THEN OPEN "COM1:4800,E,7,,CS,DS,CD" AS #1
30 IF A$ = "2" THEN OPEN "COM2:4800,E,7,,CS,DS,CD" AS #1
35 IF A$ <> "1" AND A$ <> "2" THEN CLS:
PRINT "ERROR IN COM PORT SELECTION":END
40 REM
50 REM OPEN THE COMMUNICATIONS PORT FOR 4800 BAUD 7 BITS EVEN PARITY
60 REM ( NOTE: 1200 OR 9600 ETC CAN BE USED)
70 REM
130 CLS
140 FOR X = 1 TO 20:  PRINT #1, CHR$(0);:NEXT
150 REM
160 REM SEND 20 NULLS
170 REM
180 A$ = CHR$(1)+"Z00"+CHR$(2)+"AA"+CHR$(27)+" b"+STR$(Y)+CHR$(4)
190 REM
200 REM
210 REM CHR$(1) = START OF HEADER MARKER
220 REM "Z" = ALL SIGNS RESPOND
230 REM "00" = ALL  ADDRESSES RESPOND("01","02", ...  CAN BE SUBSTITUTED)
240 REM CHR$(2) = START OF TEXT MARKER
```

```
250 REM "A" = WRITE TO TEXT file COMMAND

260 REM "A" = TEXT file LABEL ("A" FILE IS THE DEFAULT)

270 REM CHR$(27) = ESCAPE CODE TELLS SIGN THAT A MODE IS COMING

280 REM " " = BIG CHARS(OTHER CODES CAN BE SUB'D FOR TOP OR BOTTOM)

290 REM "b" = HOLD MODE (OTHER MODES CAN BE SUB'D)

300 REM STR$(Y) = TEXT TO BE DISPLAYED (IN THIS CASE ITS A NUMBER)

310 REM CHR$(4) = END OF TRANSMISSION MARKER

320 REM

330 PRINT #1, A$

340 REM

350 REM SEND THE MESSAGE TO THE SIGN

360 PRINT:PRINT " ";Y

370 REM

380 FOR X = 1 TO 10000:NEXT

390 REM

400 REM DELAY A LITTLE

410 REM

420 Y = Y + 1:  IF Y = 10000 THEN Y = 1

430 REM

440 REM INC THE COUNTER, RESET IF 10000

450 REM

460 REM DELAY A LITTLE

470 REM

480 GOTO 140

490 REM GO BACK AND LOOP AGAIN
```

SAMPLE BASIC PROGRAM-WRITE STRING FILES

```
100 OPEN “COM1:9600,E,7,2,PE,CS,DS,CD” AS #1

110 NUL$=CHR$ (0)

120 SOH$=CHR$ (1)

130 STX$=CHR$ (2)

140 ETX$=CHR$ (3)

150 EOT$=CHR$ (4)

155 HDR$=STRING$ (5,NUL$)

160 PARAM$=HDR$+SOH$+”Z00”

165 PRINT #1, PARAM$ + STX$ + “E$AAU0100FF00”;

170 FOR X=32 TO 62: PRINT #1, CHR$(X) +”BL00200000”;: NEXT X

175 PRINT #1, EOT$

180 FOR X = 1 TO 10000: NEXT X

185 PRINT #1, PARAM$+STX$+”AA”+CHR$(27)+CHR$(32)+”b”+CHR$(9)+
CHR$(16)+CHR$(32)+CHR$(16)+CHR$(33)+CHR$(16)+CHR$(34)+CHR$(
16)+CHR$(35)+EOT$

190 FOR X=1 TO 10000: NEXT X

195 DADA = 0

200 STING=32

210 PRINT #1, PARAM$;

220 PRINT #1, STX$+”G”+CHR$(STING)+CHR$(28)+”3”;

230 FOR X = 1 TO 3: PRINT #1, CHR$(OOOH);: NEXT X

240 PRINT #1, ETX$;

250 STING=STING+1: IF STING < 62 THEN GOTO 220

260 PRINT #1, EOT$

270 STING=32
```

```
280 PRINT: PRINT “TRANSMISSION NUMBER “ +STR$(COUNT)

290 PRINT PARAM$;

300 PRINT STX$+”G”+CHR$(STING)+CHR$(28)+”3”;

310 FOR X = 1 TO 3:  PRINT CHR$(OOOH);:  NEXT X

320 PRINT ETX$;

330 STING=STING+1:IF STING < 62 THEN GOTO 300

340 PRINT EOT$

350 Z = TIMER + DADA

360 OOOH = OOOH + 1:  IF OOOH > 57 THEN OOOH = 48

370 IF TIMER < Z THEN 370

380 PRINT TIME$

390 FOR X=0 TO 10000!:  NEXT X

400 PRINT TIME$

700 COUNT = COUNT +1 :  GOTO 270
```

PROTOCOL EXAMPLES

The Protocol examples will follow the same corresponding sections as the Protocol itself. For all examples, the following will be true:

<NUL> = 00H

<SOH> = 01H

<STX> = 02H

<ETX> = 03H

<EOT> = 04H

Also, all values within parenthesis are hexadecimal numbers, that is, (1C) and all other characters are ASCII characters.

<NUL> represents twenty <NUL>s x20

TRANSMISSION FRAME FORMAT

The following transmission frame will go to all unit types regardless of serial address:

<NUL><NUL><NUL><NUL><NUL><SOH>**ZOO**<STX>AAHELLO<EOT>

The transmission frame below will go to all one-line units with the address "02H":

<NUL><NUL><NUL><NUL><NUL><SOH>**102**<STX>AAHELLO<EOT>

The next transmission frame will go to all 4120c units with the address "10H" thru "1FH":

<NUL><NUL><NUL><NUL><NUL><SOH>al?<STX>AAHELLO<EOT>

TRANSMISSION FRAME VARIATIONS

1. With Checksum Field

<NUL><NUL><NUL><NUL><NUL><SOH>ZOO<STX>AAHELLO<ET X>01F6<EOT>

The Checksum for the previous <STX> thru <ETX> inclusive is 01F6H.

2. Nesting With Checksums

<NUL><NUL><NUL><NUL><NUL><SOH>ZOO<STX>E'S<ETX>00C4<S
TX>AAHELLO<ETX>01F6<EOT>

The Checksum for "<STX>E'S<ETX>" is "00C4H". The Checksum for "<STX>AAHELLO<ETX>" is 01F6H".

3. Nesting Without Checksums

<NUL><NUL><NUL><NUL><NUL><SOH>ZOO<STX>E'S<ETX><STX>AAHELLO<ETX><E
OT>

The Checksum is not required following the "<ETX>".

4. Type Code/Address Field Variation

<NUL><NUL><NUL><NUL><NUL><SOH>a01,Z1?,U26<STX>AAHELLO<EOT>

The "a01" accesses the 4120C message center with address "01" and the "Z1?" accesses all message centers with the address "10H" thru "1FH" and the "U26" accesses the 790I message center with address "26." Note the

"," (2CH) separator between each of the Type Code/Address Fields.

TEXT FILES

READ TEXT FILE

<NUL><NUL><NUL><NUL><NUL><SOH>Z06<STX>BC<EOT>

Reads the data contained in the Text file labeled "C" from any message center with serial address "06." See the Text File Data Format Section 10.2.3 on page 60 for message center response.

RESPONSE TO READ TEXT FILE

This is a response to the example in the Read Text File section above.

<NUL>x20<SOH>000<STX>ACFILE C<ETX>020C<EOT>

The message center will respond with the data found in the Text file labeled "C." In this case, the data is "FILE C."

TEXT FILE DATA FORMAT

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>AD(1B)&aHELLO<E OT>

Text file "D" will rotate the word "HELLO" on the bottom line. If this is a one-line message center, the position code is ignored.

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>A+(1B) jHELLO(1B)a<EOT>

Text file "+" will wipe down the word "HELLO" on to the middle of the message center, then the word "HELLO" will rotate off the message center (trailing rotate mode).

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>A>(1B)"n2HelloThere(1B)"a(1B)&n8<EOT>

Text file ">" will snow the words "Hello There" on to the top line of the message center, then it will rotate off the message center. Then the "SCRIPT WELCOME" graphic will appear on the bottom line.

<NUL><NUL><NUL><NUL><NUL><SOH>200<STX>AA(1B)0bHello(0D)There<EOT>

Text file "A," for two-line message centers, will hold the word "Hello" on the top line and "There" on the bottom line. For one-line message centers, "Hello" will hold on the display for a short time, then "There" will replace it.

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>AA (1B) bHello(1B)iThere(1B) aEveryone(1B) a<EOT>

Text file "A" will hold the word "Hello," then "There" will wipe up over it, then "Everyone" will rotate on, then off, the display.

ASCII Message Data

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>Az(1B)0b(15)The Time is(0D)(13)<EOT>

Text file "z" will hold (in speed 1) the words "The Time is" on the top line, and the current time on the bottom line. If this is a one-line message center, "The Time is" will hold on the display, then be replaced by the current time (also holding).

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>A@(1B)"a(19)(1A)1SMA LL(1B)"a(1B)&b(1A)3(1C)1C(1C)20(1C)3L(1C)20(1C)1R<EOT>

Text file "@" will rotate (in speed 5) the word "SMALL" in five pixel high characters on to, then off the message centers top line. Following this, the word "COLOR" will hold on the bottom line of the display in seven pixel high standard characters. Each of the characters will be a different color (on multi-color models only).

PRIORITY TEXT FILE

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>A0(1B)c(1A)9EMERGENCY<EOT>

The Priority Text file will flash the word "EMERGENCY" in full height characters until the Priority Text file is disabled.

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>A0<EOT>

The above transmission frame will disable the Priority Text file. Whatever was running on the message center when the Priority Text file was first sent will resume running.

Note: If you wish the sign to beep when a Priority Text file is being sent, the beep command must be sent before the Priority Text file is sent.

SPECIAL FUNCTIONS

WRITE SPECIAL FUNCTIONS

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>E 0830<EOT> Writes the time-of-day "0830" to all message centers.

READ SPECIAL FUNCTIONS

<NUL><NUL><NUL><NUL><NUL><SOH>Z04<STX>**F&**<EOT>

Reads the Day-of-week setting from the message center with serial address "04." See the Response to Read Special Functions (below) for message center response.

RESPONSE TO READ SPECIAL FUNCTIONS

This is a response to the example in the Read Special Functions above.

<NUL><SOH>000<STX>E&6<ETX>00A6<EOT>

x20

The message center will respond with the data found in the Day-of-week register. In this case, the data is "6" (Friday).

SPECIAL FUNCTIONS DATA FORMATS

Time-of-day Setting

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>E 0348<EOT> Writes the time-of-day "0348" (3:48 a.m.) to all message centers.

Speaker Status

<NUL><NUL><NUL><NUL><NUL><SOH>Z0?<STX>E!FF<EOT>

Disables the Speaker Status on all message centers with serial address "01H" thru "0FH" inclusive.

General Information

<NUL><NUL><NUL><NUL><NUL><SOH>M03<STX>F"<EOT>

Will read the General Information available from the Model 4160C message center with serial address "03." The response may appear as follows:

<NUL><S0H>000<STX>E"<NUL>10685403b07910108001C5E,1BF9<ETX>066F<EOT>

X20

a = EPROM part number (10185403)

b = firmware revision (g)

c = firmware release date (March 1995)

d = unit time-of-day (11:13 a.m.)

e = speaker status (00 = enabled)

f = memory pool (total size = 6E51H (28241D), unused portion = 6B92H (27538D)

Memory Pool

<NUL><NUL><NUL><NUL><NUL><SOH>D08<STX>F#<EOT>

Will read the Memory Pool from the Model 4160C message center with serial address "08." The response may appear as follows:

<NUL><SOH>000<STX>E#19EE,14BA<ETX>0275<EOT> x20

Where E#19EE,14BA:

memory pool (total size = 6E51H (28241D)

unused portion = 6B92H (27538D).

Memory Configuration

NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>E$AAU0100FF00mDU

073C10001BL000A0000<EOT>

E$AAU0100FF = Text file "A" data field (If the data field is left blank when writing the Memory Configuration the Message Center will reboot and all files will be lost. This clears the memory of the Message Center.)

001BL000A00 = String file "1" data field

Writes to all message centers the following:

Text file "A" (unlocked), 100H (256D) bytes in length, to run always. String file "1" (locked), 0AH (10D) bytes in length.

Note: The following transmission is one long string of data. Although it appears on three lines, it is concatenated.

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>E$AAU0100FF00mD
U073C10001BL000A00001AU0064FE002AU0064FE003AU0064FE004AU

0064FE005AU0064FE00<EOT>

Memory Dump

<NUL><NUL><NUL><NUL><NUL><SOH>Z01<STX>F%<EOT>

Will dump memory from the message center with serial address "01". The response may appear as below:

Note: The following transmission is one long string of data. Although it appears on two lines, it is concatenated.

NUL><SOH>000<STX>E0921<ETX>0136<STX>E$AAU1981FF00mDU073C

10001BL000A0000<ETX>07F8 <STX>AAHELLO<ETX>01FB

x20

<STX>Im<ETX>00BB<STX>G1<ETX>007D<STX>E,TUA<ETX>0162<STX>E

2A02<ETX>011F<STX>E&6<ETX> 00A6<EOT>

Where:

<STX> - Units time-of-day (9:21 a.m.) E$AAU1981FF00mDU073C10001BL000A0000<ETX> - Memory configuration

- TEXT file "A" (unlocked), 1981H (6529D) bytes in length, to run "always" - STRING file "1" (locked), 0AH (10D) bytes in length

AAHELLO - Text file "A" contents ("HELLO")

<STX>G1 - STRING file "1" contents (blank or void)

E,TUA - Run Sequence (execute according to listed TEXT file run times, unlocked, TEXT file "A" listed only)

E2A02 - Run Day Table (TEXT file "A", daily) E&6 - Units day-of-week ("6" is Friday)

Day-of-Week Setting

<NUL><NUL><NUL><NUL><NUL><SOH>Z??<STX>E&2<EOT> Writes the Day-of-Week "2" (Monday) to all message centers.

Time Display Format

<NUL><NUL><NUL><NUL><NUL><SOH>?00<STX>E’M<EOT>

Formats all message centers to display the Time-of-Day in military (24 hour) format, whenever the time-of-day is to be displayed.

SPEAKER TONE GENERATION

<NUL><NUL><NUL><NUL><NUL><SOH>Z02<STX>**E(1**<EOT>

All message centers with the serial address "02" will generate a continuous tone for about two seconds.

un Time Table

<NUL><NUL><NUL><NUL><NUL><SOH>100<STX>E)A3069B6978<EOT> File "A", A3069, with Start (30) and Stop (69) times

File "B", B6978, with Start (69) and Stop (78) times

All one-line message centers will run Text file "A" from 8:00 a.m. until 5:30 p.m. and TEXT file "B" from 5:30 p.m. until 8:00 p.m.

<NUL><NUL><NUL><NUL><NUL><SOH>U01<STX>F)<EOT>

The above transmission frame will request the Run Time Table from the message center model 790i with the serial address "01". The Read format differs from the Write format in that the Priority Text file is included, as is each files enable status, as shown below:

<NUL><SOH>000<STX>E)0FE00A30691B69780<ETX>0422<EOT>

x20

Priority Text file not running Text A, A30691, enabled Text B, B69780, disabled

Serial Error Status

<NUL><NUL><NUL><NUL><NUL><SOH>Z09<STX>F*<EOT>

This transmission frame will request the contents of the Serial Error Status register from the message center with serial address "09". The response could appear as below:

<NUL><SOH>000<STX>E*D<ETX>00B8<EOT>

x20

"D" = 44H = 01000100B

Bit 6 is always set by definition, and bit 2 was set due to a serial time-out.

Soft Reset

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>E,<EOT>

All signs on the network will do a "Soft" reset. (No memory clear; non-destructive).

Network Query

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>F-<EOT>

Will query the message center network to see what message centers are "listening". One response may appear as follows:

<NUL><SOH>000<STX>E-f05@<ETX>0182<EOT>

x20

The above response would take place approximately 3.5 seconds after the <EOT> was received from the network query. The 215C model message center ("f") with serial address "05", had no serial errors recorded ("@").

Run Sequence

<NUL><NUL><NUL><NUL><NUL><SOH>F=215CsL00<STX>E,TUABC<E OT>

The above transmission frame will write a Run Sequence consisting of the files with labels "A", "B", and "C". The files will run according to their associated run times ("T"), and the Run Sequence will be accessible from the handheld keyboard ("U").

Dimming Control

<NUL><NUL><NUL><NUL><NUL><SOH>U00<STX>E/7524<EOT>

The above transmission frame will program all 790i model message centers to dim at 7:30 p.m. ("75") and to go back to regular brightness at 6:00 a.m. ("24").

Run Day Table

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>E2A82B12<EOT> Text file "A" field, E2A

Text file "B" field, 82B

The transmission frame shown above will set up the "Run Day Table" with Text file "A" to run Monday thru Friday ("8"). The stop day for "A" is ignored ("2"). Text file "B" will start running on Monday ("2"), and stop running on Tuesday ("3").

Clear Serial Error Status

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>E4<EOT>

The Serial Error Status register will be cleared to its default value (40H) in all units.

STRING FILES

WRITE STRING FILE

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>G17,345<EOT> Writes to the String file labeled "1" the data "7,345".

READ STRING FILE

<NUL><NUL><NUL><NUL><NUL><SOH>F08<STX>H2<EOT>

Reads the data contained in the STRING file labeled "1" from the message center Model 215C with serial address 08".

RESPONSE TO READ STRING FILE

<NUL><SOH>000<STX>G28,234,000<ETX>0237<EOT>

x20

The message center Model 215C with serial address "08" will respond with the data found in the String file labeled "1". In this case, the data is "8,234,000."

String File Data Format

See String File Application Notes Section 7 on page 47 for further information.

Mitel
Powering connections

mitel.com