ALPHA™

COMMUNICATIONS PROTOCOL

VERSION 1.0
08/04/95

Adaptive Micro Systems, Inc.
7840 North 86th Street
Milwaukee, Wisconsin 53224

1.0 DOCUMENT IDENTIFIER

1.1 File Details

Document Name:	9708-8061-01
File Name:	97088061.WPD
File Format:	WordPerfect 6.1

1.2 Revision History

Version	Date	Author	Approval Signature
1.0	May 17, 1995	Mike Peters	

1.3 Revision Comments

Version	Comments
1.0	First version using WordPerfect. PrintPak added. Printable character transmissions added. Added identifier page with sign-offs and revision list.

2.0 INTRODUCTION

This document is designed to allow a user to communicate with the ALPHA™ line of electronic message centers. The message centers must have the ALPHA firmware (EPROM) installed. The standard ALPHA EPROM contains two protocols by which you may communicate with a ALPHA sign. The two types are the PrintPak™ Protocol and the EZ KEY II™ Protocol.

The PrintPak Protocol was engineered to facilitate the transmitting of messages to the ALPHA line of products via MicroSoft® Windows™ Generic Printer Driver. This makes it possible to transmit from your ALPHA sign by simply selecting print from your favorite text editor.

The EZ KEY II network was also engineered to facilitate the transmitting of messages, but also has the capabilities to send counters, pictures and more.

3.0 PRINTPAK PROTOCOL

The PrintPak Protocol allows a single message to be transmitted to a sign using a text only protocol. Special bracketed commands allow the insertion of modes, character sets, etc., into the message. PrintPak Protocol transmissions must be made at 9600 baud. PrintPak Protocol transmissions must always begin with the bracketed command {{Begin Message}}. This command may be followed by any combination of additional text and commands. Following is a list of all the supported commands. All commands are surrounded by double braces, and all commands are case sensitive. The end of the message is signaled by a timeout on the serial transmission.

PrintPak Protocol transmissions must always begin with the following message:

{{Begin Message}}

Mode commands are «{{» followed by a position, one space, the mode name, and then «}}».

The following example is for Fill Automode:

{{Fill Automode}}

Possible positions are:	Fill	Top
	Bot	Mid

Possible modes are:

Automode	Flash	Hold	Interlock
Roll Down	Roll Up	Roll In	Roll Out
Roll Left	Roll Right	Rotate	Scroll
Slide	Snow	Sparkle	Spray
Starburst	Switch	Turn Page	Twinkle
Wipe Down	Wipe Up	Wipe In	Wipe Out
Wipe Left	Wipe Right	Cherry Bomb	Fireworks
No Smoking	Running Animal	Slot Machine	Thank You
Turbo Car	Welcome	Condensed Rotate	Don't Drink & Drive

Control commands are «{{» followed by a control command, and then «}}». The following example is for Green characters:

{{Green}}

Possible Control commands are:

15/16 Row Fancy	15/16 Row Normal	Ten Row	Seven Row Normal
Seven Row Fancy	Five Row	On Wide	Off Wide
On Flash	Off Flash	On Double High	Off Double High
On True Descenders	Off True Descenders	On Fixed Width	Off Fixed Width
Red	Green	Amber	Light Red
Light Green	Brown	Orange	Yellow
Rainbow1	Rainbow2	Mix	Autocolor
New Line	New Page	Time	Fahrenheit Temp.
Celsius Temperature	Speed1	Speed2	Speed3
Speed4	Speed5	No Hold Time	

NOTE: For specific mode and control code definitions, see EZ KEY II Protocol Section 4.2.0 Text Files on Page 13.

4.0 EZ KEY II PROTOCOL

The ALPHA line of products support several types of files and a number of special functions which are used for specific applications. They are as follows:

TEXT FILE

The ASCII message data and display mode information, along with various other control codes, are stored in TEXT files. DOTS PICTURE files and STRING files may be inserted into a TEXT file.

DOTS PICTURE FILE

The DOTS PICTURE files contain data patterns that correspond to a display picture. These patterns can be used to create virtually any logo pattern on the display of the message center. These DOTS PICTURE files are accessed via TEXT files.

ALPHAVISION™ DOTS PICTURE FILE

The ALPHAVISION DOTS PICTURE file is supported only on the ALPHAVISION products. It is similar to the standard DOTS PICTURE file as described above. The ALPHAVISION DOTS PICTURE file can be much larger than the standard DOTS PICTURE file. The ALPHAVISION DOTS PICTURE file supports data compression during serial transmission.

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 a temperature, a quantity, or a timer.

SPECIAL FUNCTIONS

The ALPHA network supports a range of special functions which give you access to internal registers, diagnostics, and other miscellaneous items.

4.1.0 Transmission Frame Format (example page 58)

This section describes the basic outline of transmissions on an EZ KEY II network.

Transmission speed:	1200, 2400,4800 or 9600 baud
Data bits:	7
Start bits:	1
Stop bits:	2
Parity:	Even
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 4.1.1 on page 9.):

<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.

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

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

<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.

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

Unit Type Codes

	"Z" (5AH)	-	ALL Message Centers
	"?" (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 i.e., "TRANSMISSION OK" on the message center display, of whether or not it received the transmission frame without error.)
	"1" (31H)	-	One-line message centers
	"2" (32H)	-	Two-line message centers
	"#" (23H)	-	ALPHAVISION products
	"$" (24H)	-	Full matrix ALPHAVISION
	"%" (25H)	-	Character matrix ALPHAVISION
	"&" (26H)	-	Line matrix ALPHAVISION
	"a" (61H)	-	4120C "c" (63H) - 4200C
	"b" (62H)	-	4160C "d" (64H) - 4240C
*	"U" (55H)	-	790i "e" (65H) - 215
*	"C" (43H)	-	430i "f" (66H) - 215C
*	"D" (44H)	-	440i "k" (6BH) - 300C
	"E" (45H)	-	460i "l" (6CH) - 7000C
	"g" (67H)	-	4120R "m" (6DH) - PowerView 16 Row
	"h" (68H)	-	4160R "n" (6EH) - PowerView 24 Row
	"I" (69H)	-	4200R """ (22H) - Serial Clocks
*	"j" (6AH)	-	4240R "^" (5EH) - BETA-BRITES®
	"z" (7AH)	-	All message centers will first configure memory for 26 files of 150 characters ("A" - "Z") then execute the specified command.

NOTE: See Unit Type Codes 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)

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

* All unit Type Codes with an asterisk(*) in the description above are one-line message centers. The remaining units are

two-line message centers.

4.1.1 Transmission Frame Variations

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.

<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:

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

Command Codes
"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
"I" (49H) - Write DOTS PICTURE file
"J" (4AH) - Read DOTS PICTURE file
"M" (4DH) - Write ALPHAVISION DOTS PICTURE file
"N" (4EH) - Read ALPHAVISION DOTS PICTURE 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. Refer to the proper section for Data Field formats.

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

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

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

4.1.1 Transmission Frame Variations (cont.)
(example page 58)

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.

<NUL> x5	<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:

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	Command Code	Data Field	<ETX>	Check Sum	<STX>

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:

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	Command Code	Data Field	<ETX>

...

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

4.1.1 Transmission Frame Variations (cont.)

D. Type Code/Address Field Variation:
Format = Aaa,Bbb,Ccc,Ddd ...

; where A B C D
= 1 ASCII HEX character representing the Unit Type Code. See UNIT TYPE CODES 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 (example, page 58).

E. Pager Compatible Transmissions.

Many pagers and computer systems can not send control codes (characters lower than 20H). This variation of the transmission frame allows the entire EZ KEY II Protocol to be transmitted without sending any control codes and thus allowing its use via pager. This can be implemented in two ways, as shown below. However, an exception code must precede all control codes that are used in a transmission.

* 1. For this method the following must always be present.

<EXCEPTION CODE> <CONTROL CODE + "20H" OFFSET> MESSAGE DATA

Where the exception code is "5DH" and the control code is a value between "01H" to "1FH" hexadecimal. These hexadecimal numbers must then be converted to their respective printed characters before transmitting. For example, to send the message "HELLO THERE.", the following can be done:

<5DH> <01H + 20H> <Z00> <02H + 20H> <AA> HELLO THERE.
<04H + 20H>

Where,

<5DH> =] , <01H + 20H> = ! , <02H + 20H> = " <04H + 20H> = $

when converted to printed characters.

and,

<Z00> = Unit type code (Z) and address field (00).
<AA> = Write text file command (A) and text file label (A).
Therefore, the transmission would look like the following:

]!Z00]"AAHELLO THERE.]$

* Must use 9600 baud rate with 7 data bits and even parity.

4.1.1 Transmission Frame Variations (cont.)

* 2. Similarly, the following must always be present for this method:

<EXCEPTION CODE> <CONTROL CODE> MESSAGE DATA

Where the exception code is "5FH" and the control code is a value between "01H" to "1FH" hexadecimal. Again, these hexadecimal numbers must be converted to printed characters before transmission. The example shown above can be used as a guide. Following the same procedure the transmission would look like the following. Notice that there is no "offset" in this case.

_01Z00_02AAHELLO THERE._04

POCSAG Compatible Control Codes

EZ KEY II			POCSAG Compatible
CTL-A (01H)	-	replaced with -	"]!" - (5DH)(21H)
CTL-B (02H)	-	replaced with -	"]"" - (5DH)(22H)
CTL-C (03H)	-	replaced with -	"]#" - (5DH)(23H)
CTL-D (04H)	-	replaced with -	"]$" - (5DH)(24H)
CTL-H (08H)	-	replaced with -	"](" - (5DH)(28H)
CTL-I (09H)	-	replaced with -	"])" - (5DH)(29H)
CTL-J (0AH)	-	replaced with -	"]*" - (5DH)(2AH)
CTL-M (0DH)	-	replaced with -	"]-" - (5DH)(2DH)
CTL-P (10H)	-	replaced with -	"]0" - (5DH)(30H)
CTL-Q (11H)	-	replaced with -	"]1" - (5DH)(31H)
CTL-R (12H)	-	replaced with -	"]2" - (5DH)(32H)
CTL-S (13H)	-	replaced with -	"]3" - (5DH)(33H)
CTL-T (14H)	-	replaced with -	"]4" - (5DH)(34H)
CTL-U (15H)	-	replaced with -	"]5" - (5DH)(35H)
CTL-V (16H)	-	replaced with -	"]6" - (5DH)(36H)
CTL-W (17H)	-	replaced with -	"]7" - (5DH)(37H)
CTL-X (18H)	-	replaced with -	"]8" - (5DH)(38H)
CTL-Y (19H)	-	replaced with -	"]9" - (5DH)(39H)
CTL-Z (1AH)	-	replaced with -	"]:" - (5DH)(3AH)
CTL-[(1BH)	-	replaced with -	"];" - (5DH)(3BH)
CTL-\ (1CH)	-	replaced with -	"]<" - (5DH)(3CH)
CTL-^ (1EH)	-	replaced with -	"]>" - (5DH)(3EH)
"]" (5DH)	-	replaced with -	"_5D" - (5FH)(35H)(44H)
"^" (5EH)	-	replaced with -	"_5E" - (5FH)(35H)(45H)
"_" (5FH)	-	replaced with -	"_5F" - (5FH)(35H)(46H)
"~" (7EH)	-	replaced with -	"_7E" - (5FH)(37H)(45H)

IMPORTANT NOTES

- ***When nesting commands, only one "READ" Command Code may be used, and it must be the last Command Code before the <EOT>.***

- ***The "WRITE" Special Functions to Speaker Tone Generation must be the last command in a nested string.***

- ***When nesting commands, following all <STX> characters, it is a requirement that there be approximately a 100 millisecond delay (not to exceed the time-out period) before the Command Code is transmitted. When sending multiple transmission frames one right after another, it is also important to observe this minimum delay period.***

* Must use 9600 baud rate with 7 data bits and even parity.

4.2.0 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, refer to the section in SPECIAL FUNCTIONS on MEMORY CONFIGURATION for further 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 DOTS PICTURE as part of a TEXT file, you may simply include a call to a DOTS PICTURE file in the proper location in your TEXT file. Refer to the DOTS PICTURE files section or the STRING files section for further information.

4.2.1 Write Text File

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

<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. Refer to Appendix A for File Label descriptions (page 43).

<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. Refer to "TEXT FILE DATA FORMAT" for details (page 14).

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«A» (41H)	File Label	TEXT File Data	<EOT>

4.2.2 Read Text File
(example page 59)

Command Character: "B" (42H)

Transmission Frame Format:

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«B» (42H)	File Label	<EOT>

File Label: One ASCII character indicating the TEXT file being accessed. Refer to Appendix A for File Label descriptions (page 43).

<NUL>	<SOH>	Type	Addr.	<STX>	«B»	File	<EOT>

x5		Code	Field		(42H)	Label	

4.2.3 Response to Read Text File
(example page 59)

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

<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. Refer to Appendix A for File Label descriptions (page 43).

<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. Refer to "TEXT FILE DATA FORMAT" for details (below).

<NUL> x20	<SOH>	«000"	<STX>	«A» (41H)	File Label	TEXT File Data	<ETX>	Check Sum	<EOT>

IMPORTANT: ***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***.

4.2.4 Text File Data Format
(example page 59)

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).

{see NOTE 1 - page 15}

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

MODE FIELD

MODE FIELD:

<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.

{see NOTE 1 - below}

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

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.

{see NOTE 1 - below}

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

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

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:

{see NOTE 1 - above}

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

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.
"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.

* Indicates default setting

	"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)	- 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.

{see NOTE 1 - below)

<ESC> (1BH)	Display Position	«n» (6EH)	Special Specifier

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

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.

* Indicates default setting

SPECIAL GRAPHICS

"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.

NOTE: 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 (example page 60)

{see NOTE 1 - page 15}

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

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 on page 18. The Control codes are used to alter, among other things, the character size, color, and display speed.

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 - 1/2sp
2FH - /	3FH - ?	4FH - O	5FH - _	6FH - o	

sp = space
1/2sp = 1/2 space
cnt = cent sign

CONTROL CODES

CTL-E (05H) - Double High: This switch enables or disables the double height character control. Followed by:
(2 byte)

*, ** "0" (30H) - Double height off
"1" (31H) - Double height on

CTL-F (06H) - True Descenders: This switch will cause characters with descenders (i.e., "g" and "y") to be displayed
(2 byte) with descenders extended below text base line. Followed by:

*, ** "0" (30H) - True descenders off
"1" (31H) - True descenders on

CTL-G(07H) - Character Flash: This switch will cause characters to flash. Followed by:
(2 byte)

*, ** "0" (30H) - Character flash off
"1" (31H) - Character flash on

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.

EXTENDED CHARACTER SETS lists the valid characters and their codes:

20H - Ç	2DH - ì	3AH - Ü	47H - º	54H - š
21H - ü	2EH - Ä	3BH - ¢	48H - ¿	55H - _
22H - é	2FH - Å	3CH - £	49H - °	56H - _
23H - â	30H - É	3DH - ¥	4AH - ¡	57H - ß
24H - ä	31H - æ	3EH - _	4BH - sc	58H - Š
25H - à	32H - Æ	3FH - *f*	4CH - _	59H - ß
26H - å	33H - ô	40H - á	4DH - _	5AH - Á
27H - ç	34H - ö	41H - í	4EH - _	5BH - À
28H - ê	35H - ò	42H - ó	4FH - _	5CH - Á
29H - ë	36H - û	43H - ú	50H - _	5DH - á
2AH - è	37H - ù	44H - ñ	51H - _	5EH - É
2BH - ï	38H - ÿ	45H - Ñ	52H - _	5FH - Í
2CH - î	39H - Ö	46H - ª	53H - Đ	60H - _

where: sc = single column space 61H - _

CTL - \ (1CH) - Temperature display in Celsius (See NOTE 1)
CTL -] (1DH) - Temperature display in Fahrenheit (See NOTE 1)
"z" (7AH) - Display counter 1 current value
"{" (7BH) - Display counter 2 current value
"|" (7CH) - Display counter 3 current value
"}" (7DH) - Display counter 4 current value
"~" (7EH) - Display counter 5 current value

NOTE 1: Available on incandescent message centers only (790i, 430i, 440i, and 460i.)

CONTROL CODES

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 (10H) - Call STRING file: Must be followed by a STRING file label. Refer to the STRING
(2 byte) files section (page 34) for more information.

* CTL-Q (11H) - Disable wide characters.

CTL-R (12H) - Enable wide characters.

CTL-S (13H) - Call Time: The time-of-day will be called up. Refer to the SPECIAL FUNCTIONS (page 23) for time-of-day setting and time display format selection.

CTL-T (14H) - Call DOTS PICTURE file:
(2 byte) Must be followed by a DOTS PICTURE file Label. Refer to the DOTS PICTURE files section (page 35) for more information.

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 (18H) - Select Speed 4
CTL-Y (19H) - Select Speed 5

CTL-Z (1AH) - Select Character Set:

*Indicates Default Setting

(2 byte) 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. (ALPHAVISION only)
"8" (38H) - Full height fancy characters
"9" (39H) - Full height standard characters

CTL-\ (1CH) (2 byte) - 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
"4" (34H) - Dim Red
"5" (35H) - Dim Green
"6" (36H) - Brown
"7" (37H) - Orange
"8" (38H) - Yellow
"9" (39H) - Rainbow 1
"A" (41H) - Rainbow 2
"B" (42H) - Color mix (each char. is a different color)
* "C" (43H) - Autocolor selection

NOTE 1: Some message center models do not support the full range of colors. 4000C and ALPHAVISION series units support only Red, Green, Amber, Rainbow, and Mix.

CTL-] (1DH) (3 byte) - 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-^ (1EH) (2 byte) - 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

CTL-_(1FH) (15 byte) - Call ALPHAVISION DOTS PICTURE file. This command is followed by a field formatted as follows:

data = SFFFFFFFFFtttt (14 characters)

S = Status "C" (43H) if file is running as a part of a Quick Flick. The display is cleared before each ALPHAVISION DOTS PICTURE is

displayed.

* Indicates Default Setting

4.2.4 Text File Data Format (cont.)

Status "L" (4CH) if file is running as a DOTS PICTURE file. If text from a TEXT file is displayed with the DOTS PICTURE file, the display hold time is ignored and the TEXT file display speed is utilized.

FFFFFFFFF = Nine character file name (if file name consists of less than nine characters, spaces (20H) should precede the file name, so the total number of characters stay fixed at nine characters.)

tttt = Display hold time. Four digit ASCII hex number indicates tenths of seconds. Leading zero"s are required. (i.e. "0020" = 32 = 3.2 second hold time)

4.2.5 Priority Text File (example page 60)

This is a special 125 byte TEXT file which is pre-configured into all message centers. The transmission frame for accessing the PRIORITY TEXT file follows the TEXT file format (page 13). 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:

- No TEXT File Data (blank file) is sent to the PRIORITY TEXT file.
- A serial write to the RUN TIME TABLE takes place.
- A serial write to the RUN DAY TABLE takes place.
- Any serial error occurs during the PRIORITY TEXT file write.
- 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.

5.0 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

5.1 WRITE SPECIAL FUNCTIONS (example page 60)

Command Character: "E" (45H)
Transmission Frame Format:

<NUL> x5	<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. Refer to the SPECIAL FUNCTIONS DATA FORMATS (page 23).

<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. Refer to the section on SPECIAL FUNCTIONS DATA FORMATS for details (page 23).

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«E» (45H)	S.F. Label	S.F. Data	<EOT>

5.2 Read Special Functions (example page 60)

Command Character: "F" (46H)
Transmission Frame Format:

<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. Refer to the SPECIAL FUNCTIONS DATA FORMATS.

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«F» (46H)	S.F. Label	<EOT>

5.3 Response to Read Special Functions (example page 60)

This is the data sent from the message center following a READ SPECIAL FUNCTIONS.

Transmission Frame Format:

<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. Refer to the SPECIAL FUNCTIONS DATA FORMATS.

<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. Refer to the section on SPECIAL FUNCTIONS DATA FORMATS for details.

<NUL> x20	<SOH>	«000"	<STX>	«E» (45H)	S.F. Label	S.F. Data	<ETX>	Check Sum	<EOT>

Important: ***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.***

5.4 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.

" " (20H) - **Time-of-day Setting** (Read/Write) (example page 60)
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. Refer to the CONTROL CODES (page 19) for Clock Display, and refer to Time Display Format in the SPECIAL FUNCTIONS section (page 27) for available display formats.

"!" (21H) - **Speaker Status** (Read/Write) (example page 61)
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, refer to the Speaker Tone Generation portion of the SPECIAL FUNCTIONS section (page 27).

""" (22H) - **General Information** (Read Only) (example page 61)
data = <NUL>FFFFFFFFfMmYyHhNnRSSPOOL,pool (28 or 29 ASCII characters total)

***	<NUL>	=	(00H)
	FFFFFFFF	=	The firmware (EPROM) chip, Adaptive Micro Systems' 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	=	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.

* Indicates default setting
*** This byte is transmitted only on some message center models.

"#" (23H) - **Memory Pool** (Read Only) (example page 61)
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.

"$" (24H) - **Memory Configuration** (Read/Write) (example page 61)

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).

IMPORTANT: Message centers without address plugs may have their address cleared from memory.

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. Refer to Appendix A (page 43) for valid File Labels.

T = One ASCII character representing the file type. Valid entries are listed below:

"A" (41H) - TEXT file
"B" (42H) - STRING file
"D" (43H) - DOTS PICTURE 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"). For DOTS PICTURE files, the first two ASCII hexadecimal digits represent the number of pixel rows, while the second two ASCII hexadecimal digits represent the number of pixel columns.

Important: ***The summation of all the file sizes (except for DOTS PICTURE files and FAR DOTS PICTURE files) plus eleven bytes of overhead for each file should not exceed the total amount of available memory in the pool.***

NOTE: "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. Refer to Appendix B (page 44) for the table of valid Start/Stop time values.

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

QQQQ (DOTS PICTURE File)
= DOTS PICTURE Color Status. Valid entries are shown below:
"1000" = monochrome Dots Picture
"2000" = three color Dots Picture
"4000" = eight color Dots Picture

"%" (25H) - **Memory Dump** (Read Only) (example page 62)
data = multiple nested transmission frames with Checksums (refer to Section 1.1B) 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, STRING, or DOTS PICTURE file)
4. Run Sequence
5. Run Day Table
6. Day-of-Week Setting
7. Counter Functions

Refer to the appropriate section for format details on each of the above transmission frames.

"&" (26H) - **Day-of-Week setting** (Read/Write) (example page 63)
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 are listed below:

"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) (example page 63)

data = One ASCII character representing the time format and how it is displayed by the message center. Valid entries are:

* "S" (53H) = Standard a.m./p.m. format
"M" (4DH) = 24 hour (military) format

' (28H) - **Speaker Tone Generation** (Write Only) (example page 63)

data = B (See NOTE 1.)

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." (See NOTE 2.)
"B" (42H)	-	Turn speaker port "off." (See NOTE 2.)
"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".

NOTE 1: 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.

NOTE 2: 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.

")" (29H) - **Run Time Table** (Read/Write) (example page 64)

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. Refer to Appendix A (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. Refer to Appendix B (page 44) for the table of valid Start/Stop time values. These will overwrite what is in the Memory Configuration.

* Indicates default setting

(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" - 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. Refer to Appendix A for valid File Labels (page 43).

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. Refer to Appendix B for the table of valid Start/Stop time values (page 44).

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) (example page 64)
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).

"," (2CH) - Soft Reset (Write Only) (example page 64)
data = none

There is no data in this data field. Writing this will re-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).

* Indicates default setting

"-" (2DH) - **Network Query** (Read Only) (example page 64)

data = UAAZ

U = One ASCII character representing the unit type. See the UNIT TYPE CODES (page 8) 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 the Serial Error Status portion of SPECIAL FUNCTIONS, for further details (page 28).

Please NOTE that 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.

"." (2EH) - **Run Sequence** (Read/Write) (example page 64)

data = KPF
|
Repeating portion

K = One ASCII character representing the Run Sequence key code. Valid entries are shown below:

* "T"(54H) - 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) - 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) (example page 64)

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.

* Indicates default setting

Refer to Appendix B (page 44) 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, 430i, 440i, and 460i.

"2"(32H) - **Run Day Table** (Read/Write) (example page 65)

data = FSs
|__|
|
Repeating field

F = One ASCII character representing the TEXT File Label. Refer to Appendix A for valid File Labels (page 43).

S = One ASCII hexadecimal character representing the TEXT file run start day. Valid start day codes are listed below:

*	"0" (30H) = Daily	"6" (36H) = Friday
	"1" (31H) = Sunday	"7" (37H) = Saturday
	"2" (32H) = Monday	"8" (38H) = Monday-Friday
	"3" (33H) = Tuesday	"9" (39H) = Weekends
	"4" (34H) = Wednesday	"A" (41H) = Always
	"5" (35H) = Thursday	"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	"5" (35H) = Thursday
*	"2" (32H) = Monday	"6" (36H) = Friday
	"3" (33H) = Tuesday	"7" (37H) = Saturday
	"4" (34H) = Wednesday	

NOTE: If the start day covers multiple days (i.e., daily, never, etc.) the stop day is ignored, but still required.

"4" (34H) - **Clear Serial Error Status** (Write Only) (example page 65)

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.

";" (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.

* Indicates default setting

"5" (35H) - **Counter Functions** (Read/Write) (example page 65)
Refer to Appendix C (page 45) for further information

data = 1Cone2Ctwo3Cthree4Cfour5Cfive
- 1 = "1" (31H) - represents Counter "1"
- Cone = Counter "1" data
- 2 = "2" (32H) - represents Counter "2"
- Ctwo = Counter "2" data
- 3 = "3" (33H) - represents Counter "3"
- Cthree = Counter "3" data
- 4 = "4" (34H) - represents Counter "4"
- Cfour = Counter "4" data
- 5 = "5" (35H) - represents Counter "5"
- Cfive = Counter "5" data

All counter data (Cone, Ctwo, Cthree, Cfour, Cfive) takes the following format:

Counter data = BBTTttSSSSSSSSiiiiiiiiVVVVVVVVttttttttFFmmHH

BB = Two ASCII hexadecimal characters representing the Counter Control Byte. The default setting Counter Control value is 64H (01100100B). Each bit of the Counter Control Byte has special meaning regarding the counter's functionality as shown below:

(MSB) bit 7 = 1 (counter on) or 0 (counter off)
bit 6 = 1 (increment) or 0 (decrement)
bit 5 = 1 (count minutes) or 0 (other)
bit 4 = 1 (count hours) or 0 (other)
bit 3 = 1 (count days) or 0 (other)
bit 2 = 1 (weekends on) or 0 (weekends off)
bit 1 = 1 (auto-reload on) or 0 (auto-reload off)
(LSB) bit 0 = 0

Auto-reload "on" will reload the Current Counter Value with the Counter Start Value, once the Counter Target Value has been reached.

TT = Two ASCII hexadecimal characters representing the Counter Start Time (Default value = "FF" - Always).

tt = Two ASCII hexadecimal characters representing the Counter Stop Time (Default value = "00" - ignored when counter start time is "Always").

Refer to Appendix B (page 44) for the table of valid Start/Stop time values. Time codes "FD" and "FE" are invalid codes for both Counter Start and Stop Times. Also, time code "FF" is invalid for a Counter Stop time.

SSSSSSSS
= Eight digit BCD number representing the Counter Start Value. The default value is "00000000," and the maximum value is "99999999." Leading "0"s must be sent.

iiiiiiii = Eight digit BCD number representing the number that is incremented or decremented (as dictated by Bit 6 of the Counter Control byte) from the Current Counter Value. This is called the Counter Change (Increment/Decrement) Value. The default value is "00000001," and the maximum value is "99999999." Leading "0"s must be sent.

VVVVVVVV
= Eight digit BCD number representing the Current Counter Value. The default value is

"00000000," and the maximum value is "99999999." Leading "0"s must be sent.

5.0 SPECIAL FUNCTIONS (cont.)

tttttttt = Eight digit BCD number representing the Counter Target Value. When the Current Counter Value reaches the Counter Target Value, the Target file(s) which are set up to trigger (as dictated in the Target File Byte) will be activated. The default value is "00000000," and the maximum value is "99999999." Leading "0"s must be sent.

FF = Two ASCII hexadecimal characters representing the Target File Byte. The Target File Byte controls the Target file(s) to be triggered when the Current Counter Value reaches the Counter Target Value. The default Target File Byte value is 00H (00000000B). Bit assignments are shown below:

(MSB) bit 7 = 0
bit 6 = 0
bit 5 = 0
bit 4 = Target file 1 status(1-enabled,0-disabled)
bit 3 = Target file 2 status(1-enabled,0-disabled)
bit 2 = Target file 3 status(1-enabled,0-disabled)
bit 1 = Target file 4 status(1-enabled,0-disabled)
(LSB) bit 0 = Target file 5 status(1-enabled,0-disabled)

mm = Two ASCII hexadecimal characters setting the time-of-day (minutes) when the Counter Value will change. This only applies when counting hours or days (as dictated in the Counter Control Byte). This is called the Counter Change Synchronization - Minutes. If minutes are being counted, this value is ignored. Valid values are "00" thru "3B" (00D thru 59D). The default value is "00."

HH = Two ASCII hexadecimal characters setting the time-of-day (hours) when the Counter Value will change. This only applies when counting days (as dictated in the Counter Control Byte). This is called the Counter Change Synchronization - Hours. If minutes or hours are being counted, this value is ignored. Valid values are "01" thru "18" (01D thru 24D). "01" represents 1 a.m., "18" represents 12 midnight. The default value is "18."

"7"(37H) - Serial Address (Write Only)

data = AA

AA = Two ASCII hexadecimal characters representing the desired serial address for the ALPHAVISION sign.

NOTE: If the serial address has been set using the hardware dip switches to an address other than "00," the dip switch address will override the serially configured serial address upon power-up. The serially configured serial address is stored in battery backed RAM.

"8"(38H) - ALPHAVISION DOTS PICTURE file Memory Configuration (Read/Write)

Command allows reads and writes to the ALPHAVISION dots configuration table.

Data format is as follows:

data = FFFFFFFFFPRRRRCCCCccrrrr (24 ASCII characters)
|_____|
|
This data field repeats for each file configured in the message center.

FFFFFFFFF
= 9 character file name (if file name consists of less than nine characters, spaces (20H)

should precede the file name, so the total number of characters stay fixed at nine characters.)

5.0 SPECIAL FUNCTIONS (cont.)

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 hand-held keyboard.
"L"(4CH) - Locked - This makes the file inaccessible from the hand-held keyboard.

RRRR = Four ASCII hexadecimal digits representing the number of pixel rows. Leading zeros are required (i.e., "0040" = 64 rows).

CCCC = Four ASCII hexadecimal digits representing the number of pixel columns. Leading zeros are required (i.e., "0060" = 96 columns).

cc = Two ASCII hexadecimal digits representing the number of colors in the Far Dots Picture.

"01" represents a monochrome Dots Picture,
"02" represents a tricolor Dots Picture.

rrrr = Reserved for future use. Four ASCII zeros are required.

"9" (39H) - Append to ALPHAVISION DOTS PICTURE file Memory Configuration (Write only)

Command allows appending to the ALPHAVISION DOTS PICTURE file Memory Configuration. The data format is the same as the ALPHAVISION DOTS PICTURE file Memory Configuration data format.

"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."

NOTE: Temperature Offset only applies on incandescent message center Models 790i, 430i, 440i, and 460i

6.0 STRING FILES

STRING files are used to store short ASCII strings of characters which may be "called up" from a TEXT file. The main purpose of STRING files is to display frequently changing information. When writing STRING files to a message center, the display will not blank as it does when writing TEXT files. This is because the STRING FILE DATA is buffered and TEXT file internal Checksum does not change. Because the STRING FILE DATA is buffered, there is a limit to the size of the STRING file of 125 bytes.

Before being able to write to a STRING file, memory must be allocated for the STRING file in the message center. Refer to MEMORY CONFIGURATION (page 25) for details.

STRING files are "called up" from TEXT files utilizing the TEXT file control code designated for a "Call STRING file." Refer to the CONTROL CODES (page 18) for further information.

When 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 APPENDIX D (page 46).

WRITE/READ STRING FILE

6.1 WRITE STRING FILE (example page 66)

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

<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. Refer to Appendix A for File Label descriptions (page 43).

<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. Refer to "STRING FILE DATA FORMAT" for details (page 35).

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«G» (47H)	File Label	STRING File Data	<EOT>

6.2 READ STRING FILE (example page 66)

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

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«H» (48H)	File Label	<EOT>

File Label: One ASCII character indicating the STRING file being accessed. Refer to Appendix A for File Label descriptions (page 43).

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«H» (48H)	File Label	<EOT>

6.3 RESPONSE TO READ STRING FILE (example page 67)
Transmission Frame Format for data sent following a READ STRING file

<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. Refer to Appendix A for File Label descriptions (page 43).

<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. Refer to "STRING FILE DATA FORMAT" for details (below).

<NUL> x20	<SOH>	«000"	<STX>	«G» (47H)	File Label	STRING File Data	<ETX>	Check Sum	<EOT>

Important: ***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.***

6.4 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 the ASCII CHARACTER (page 17). Refer to the CONTROL CODES (page 18) for further definition of the following control codes which are acceptable within a STRING file. All other control codes are NOT acceptable.

	CTL-I (09H)	-	"No Hold" Speed
	CTL-M (0DH)	-	Carriage Return
*	CTL-Q (11H)	-	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)	-	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

* Indicates default setting

7.0 DOTS PICTURE FILES

DOTS PICTURE files are used to store dot patterns which may be "called up" from a TEXT file. The main purpose of DOTS PICTURE files is to allow the user to display custom logos or pictures. When writing DOTS PICTURE files to a message center, the display will blank until the transmission is complete.

ALPHAVISION units support both DOTS PICTURE files and ALPHAVISION DOTS PICTURE files. When a DOTS PICTURE exceeds a pixel height of 31 rows or a pixel width of 255 columns, the ALPHAVISION DOTS PICTURE file must be used.

DOTS PICTURE files are "called up" from TEXT files utilizing the TEXT file control code designated for a DOTS PICTURE call. Refer to the CONTROL CODES (page 18) for further information.

When reading from a DOTS PICTURE 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.

WRITE/READ DOTS PICTURE FILE

7.1 WRITE DOTS PICTURE FILE (example page 67)

Command Character: "I" (49H)
Transmission Frame Format:

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«I» (49H)	File Label	DOTS PICTURE File Data	<EOT>

File Label: One ASCII character indicating the DOTS PICTURE file being accessed. Refer to Appendix A (page 43) for File Label descriptions.

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«I» (49H)	File Label	DOTS PICTURE File Data	<EOT>

DOTS PICTURE File Data: The contents of the DOTS PICTURE file. Refer to "DOTS PICTURE FILE DATA FORMAT" for details (page 37).

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«I» (49H)	File Label	DOTS PICTURE File Data	<EOT>

7.2 READ DOTS PICTURE FILE (example page 68)

Command Character: "J" (4AH)
Transmission Frame Format:

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«J» (4AH)	File Label	<EOT>

File Label: One ASCII character indicating the DOTS PICTURE file being accessed. Refer to Appendix A (page 43) for File Label descriptions.

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«J» (4AH)	File Label	<EOT>

7.3 RESPONSE TO READ DOTS PICTURE FILE (example page 68)

This is the data sent from the message center following a READ DOTS PICTURE file.
Transmission Frame Format:

<NUL> x20	<SOH>	«000»	<STX>	«I» (49H)	File Label	DOTS PICTURE File Data	<ETX>	Check Sum	<EOT>

NOTE: Response Type Code and Address Fields are filled in with the ASCII string "000."

File Label: One ASCII character indicating the DOTS PICTURE File being accessed. Refer to Appendix A (page 43) for File Label descriptions.

<NUL> x20	<SOH>	«000»	<STX>	«I» (49H)	File Label	DOTS PICTURE File Data	<ETX>	Check Sum	<EOT>

DOTS PICTURE File Data: The contents of the DOTS PICTURE file. Refer to "DOTS PICTURE FILE DATA FORMAT" for details (below).

<NUL> x20	<SOH>	«000»	<STX>	«I» (49H)	File Label	DOTS PICTURE File Data	<ETX>	Check Sum	<EOT>

Important: ***Whenever doing a READ DOTS PICTURE 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.***

7.4 DOTS PICTURE FILE DATA FORMAT

This section outlines the format of the DOTS PICTURE file data field. The height (Y) and width (X) of the DOTS PICTURE are in terms of pixels. The first row of the DOTS PICTURE file is the top and the first column is the leftmost.

Height (Y)	Width (X)	Row bit pattern	CTL-M (ODH)	CTL-J (0AH) (Optional)

repeating field
(Y times)

Height: Two ASCII hexadecimal bytes representing the number of pixel rows (Y) in the DOTS PICTURE bit pattern. This must match the pixel row bytes set up in the MEMORY CONFIGURATION for this DOTS PICTURE file. For ALPHAVISION DOTS PICTURE files, four ASCII hexadecimal bytes are used to represent the number of pixel rows.

Height (Y)	Width (X)	Row bit pattern	CTL-M (ODH)	CTL-J (0AH) (Optional)

Width: Two ASCII hexadecimal bytes representing the number of pixel columns (X) in the DOTS PICTURE bit pattern. This must match the pixel column bytes set up in the MEMORY CONFIGURATION for this DOTS PICTURE file. For ALPHAVISION DOTS PICTURE files, four ASCII hexadecimal bytes are used to represent the number of pixel columns.

Height (Y)	Width (X)	Row bit pattern	CTL-M (ODH)	CTL-J(0AH) (Optional)

NOTE: When doing a WRITE DOTS PICTURE file, the message center will clear the current DOTS PICTURE file in memory immediately following the width information.

Important: ***Following the width bytes, there should be approximately a 100 millisecond delay (not to exceed the time-out period) before sending the Row bit pattern information.***

Row Bit Pattern: Every pixel is represented by an ASCII character (including unlit pixels).

Height (Y)	Width (X)	Row bit pattern	CTL-M (ODH)	CTL-J(0AH) (Optional)

| repeating field (Y times)

The first character is the leftmost pixel of the DOTS PICTURE. If the number of row pixel characters sent exceeds the DOTS PICTURE width, the extra pixel characters will be discarded. If the number of row pixel characters sent is less than the DOTS PICTURE width, the DOTS PICTURE file in the message center will leave the remaining row bits cleared (off). The ASCII representations for the various colors are listed below:

"0" (30H) - pixel off
"1" (31H) - pixel on - red
"2" (32H) - pixel on - green
"3" (33H) - pixel on - amber
"4" (34H) - pixel on - dim red
"5" (35H) - pixel on - dim green
"6" (36H) - pixel on - brown
"7" (37H) - pixel on - orange
"8" (38H) - pixel on - yellow

NOTE: Some message center models do not support the full range of colors. 4000C series, ALPHAVISION, and 221C units support only red, green, and amber.

Data Compression: ALPHAVISION products support pixel data compression for the row bit pattern. The data compression command can be inserted anywhere within the row bit pattern. The format for the data compression is:

<CTL-Q>XXB

Where: CTL-Q = (11H)

XX = Pixel repeat count. Two ASCII hex characters define the number of times to repeat data. (i.e., 01 will write 2 pixels to defined color and FF will write 256 pixels to defined color.)

B = ASCII character defines the pixel color to be repeated as defined above.

7.0 DOTS PICTURE FILES (cont.)

CTL-M (0DH): The carriage return signals the end of the row of pixels, and the beginning of the next row. This is not required following the last row bit pattern transmission.

CTL-J (0AH): Line feed is not required, but if sent during a WRITE DOTS PICTURE file, is discarded by the message center. The message center will not send any line feeds following the carriage return during a RESPONSE TO READ DOTS PICTURE file.

8.0 ALPHAVISION DOTS PICTURE FILES

ALPHAVISION DOTS PICTURE files are used to store dot patterns which may be "called up" from a TEXT file. The main purpose of ALPHAVISION DOTS PICTURE files is to overcome the limitations set in the DOTS PICTURE file formatting. One of these limitations is the number of columns possible in a DOTS PICTURE file. 255 columns are maximum on a DOTS PICTURE file. Another advantage of using ALPHAVISION DOTS PICTURE files is the capability of data compression. When writing ALPHAVISION DOTS PICTURE files to an ALPHAVISION product, the display will blank until the transmission is complete.

ALPHAVISION DOTS PICTURE files are "called up" from TEXT files utilizing the TEXT file control code designated for an ALPHAVISION DOTS PICTURE call. Refer to the CONTROL CODES for further information (page 18).
When reading from an ALPHAVISION DOTS PICTURE file, the display will either pause or blank when sending the transmission frame. Once the unit has completely transmitted the file, it will continue displaying the message from where it was interrupted.

8.1 WRITE ALPHAVISION DOTS PICTURE FILE

Command Character: "M" (4DH)
Transmission Frame Format:

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«M» (4DH)	File Name	DOTS PICTURE File Data	<EOT>

File Name: Nine ASCII characters indicating the ALPHAVISION DOTS PICTURE file being accessed.

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«M» (4DH)	File Name	DOTS PICTURE File Data	<EOT>

DOTS PICTURE File Data: The contents of the DOTS PICTURE file. Slight differences exist between the DOTS PICTURE data format and the ALPHAVISION DOTS PICTURE data format. Refer to "DOTS PICTURE FILE DATA FORMAT" for details. (page 37)

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«M» (4DH)	File Name	DOTS PICTURE File Data	<EOT>

8.2 READ ALPHAVISION DOTS PICTURE FILE

Command Character: "N" (4EH)
Transmission Frame Format:

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«N» (4EH)	File Name	<EOT>

File Name: Nine ASCII characters indicating the FAR DOTS PICTURE file being accessed.

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«N» (4EH)	File Name	<EOT>

Important: ***Whenever doing a READ ALPHAVISION DOTS PICTURE 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.***

8.0 ALPHAVISION DOTS PICTURE FILES

8.3 RESPONSE TO READ ALPHAVISION DOTS PICTURE FILE

This is the data sent from the message center following a READ DOTS PICTURE file.

Transmission Frame Format:

<NUL> x20	<SOH>	«000"	<STX>	«M» (4DH)	File Name	DOTS PICTURE File Data	<ETX>	CHECK SUM	<EOT>

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

File Name: Nine ASCII characters indicating the FAR DOTS PICTURE file being accessed.

<NUL> x20	<SOH>	«000"	<STX>	«M» (4DH)	File Name	DOTS PICTURE File Data	<ETX>	CHECK SUM	<EOT>

DOTS PICTURE File Data: The contents of the DOTS PICTURE file. Refer to "DOTS PICTURE DATA FORMAT" for details (page 37).

<NUL> x20	<SOH>	«000"	<STX>	«M» (4DH)	File Name	DOTS PICTURE File Data	<ETX>	CHECK SUM	<EOT>

9.0 ALPHAVISION BULLETIN MESSAGING

Description: Bulletin Message. Allows a message of up to 200 characters to be rotated on the display without interrupting the current operation.

Command Format:

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«0" (4DH)	Position	...

... | Justification | Width | Count | Text | <EOT> |
|---|---|---|---|---|

Where:

"0": Actual Command Code

Position: Position of bulletin. Use:
"T" for Top of display,
"B" for Bottom of display.

Justification: "L" for Left side of display
"C" for Center of display
"R" for Right side of display

Width: Two ASCII HEX digits specifying the number of characters to be displayed in the bulletin window. The actual size of the window will be rounded up to the nearest 32 column width. The maximum is 256 columns.

Count: Two ASCII HEX digits specifying the number of times the Bulletin message should be displayed.

Text: ASCII characters composing the message. The only control code allowed are color selection. The maximum number of characters allowed for this command is 225 characters. All longer messages will be truncated.

To terminate the Bulletin:

<NUL> x5	<SOH>	Type Code	Addr. Field	<STX>	«0" (4DH)	'T'	<EOT>

NOTE:
- Only the size of the Bulletin window is cleared - not the entire line.
- Only seven high characters are supported.
- All modes and flashing will stop. The display will continue to update string data and cycle through pages, but only with the HOLD mode.
- Only the ALPHAVISION products support varying the window size and location. The ALPHA 7000 Series performs the bulletin message across the entire width of the sign.

APPENDIX A

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 the section on PRIORITY TEXT file for further information (page 21).

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

APPENDIX B

TEXT FILE START AND STOP TIMES

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

APPENDIX C

COUNTER PROTOCOL

1. TEXT files have an additional CONTROL CODE available for the displaying of the Counter Value. For information on how to display the Counter Values, refer to the EXTENDED CHARACTER SETS located in TEXT file DATA FORMAT Section 4.2.4 (page 14). Also, refer to the appropriate section of APPENDIX H - PROTOCOL EXAMPLES, for further information (page 58).

2. The default Memory Configuration on a unit equipped with the COUNTER UPGRADE (in addition to the default TEXT file "A" and DOTS PICTURE file "A") contains five TARGET TEXT files with labels "1" thru "5". Each file is set up with a keyboard status of "unlocked" and is 100 bytes in length (64H). The default Run Start Time for each is "Never" (FEH). Refer to the Memory Configuration portion of SPECIAL FUNCTIONS DATA FORMATS (page 25), for additional information. It is important to keep in mind that when writing a new Memory Configuration that TEXT files "1" through "5" are included, as these are the TARGET files. Refer to the appropriate section of APPENDIX H - PROTOCOL EXAMPLES, for further information (page 58).

3. The Memory Dump (see SPECIAL FUNCTIONS DATA FORMATS - page 26) response, from a message center equipped with the Counter upgrade, also contains the Counter Functions information (page 65). Also, refer to the appropriate section of APPENDIX H - PROTOCOL EXAMPLES, for further information (page 58).

4. It is important to set up a Run Sequence (page 29) which runs according the file run times. Also, all five Target File Labels ("1" thru "5") should always be included in the Run Sequence, along with other desired TEXT files. Also, refer to the appropriate section of APPENDIX H - PROTOCOL EXAMPLES, for further information (page 58).

5. It is important to set up a Run Day Table (page 30) which accounts for, in addition to all user TEXT files, the Target files. The default Start Day value for all Target TEXT files is "0" (Daily), and the default Stop Day value is "2" (ignored). Also, refer to the appropriate section of APPENDIX H - PROTOCOL EXAMPLES, for further information (page 58).

6. All the Counter information does not exist in standard EZ KEY II firmware. Refer to the Counter Functions portion of the SPECIAL FUNCTIONS DATA FORMATS (page 31), for information on how to READ and WRITE the following information:

   Counter Control Byte
   Counter Target File Byte
   Counter Start Time
   Counter Stop Time
   Counter Start Value
   Counter Target Value
   Counter Change (Increment/Decrement) Value
   Current Counter Value
   Counter Change Synchronization - Minutes
   Counter Change Synchronization - Hours

   Also, refer to the appropriate section of APPENDIX H - PROTOCOL EXAMPLES, for further information (page 58).

APPENDIX D

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, a count-down timer.

One large advantage of using STRING files to update some displayed data is that the LED display won't "blink" or flash during the update, as it will during the updating of TEXT files. 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.

1. To allocate memory for one STRING file and the TEXT file from which it is called, send the data stream below. The Address Field is set up to talk to all signs on your network. For example:

<NUL> X5	<SOH>	«ZOO»	<STX>	«E$AAU0400FF001BL00200000"	<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)

"00" - (30H,30H) - ignored
"00" - (30H,30H) - ignored

<EOT> - (04H) - "End of Transmission" character

2. To write the TEXT file which calls the STRING file, see below:

<NUL> x5	<SOH>	«ZOO »	<STX>	«AA»,<ESC>,» Bthe count is «,<DLE>,»1"	<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
"A" - (41H) - Write TEXT File Command Code
"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

"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

3. To update the STRING file, see below:

<NUL> x5	<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"

APPENDIX D (cont.)

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.

APPENDIX E

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.
 * Adaptive Micro Systems, Inc.  Milwaukee, WI  USA.
 *
 *****************************************************************/

#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, F = 480 etc) */
        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);
        }
```

APPENDIX F

SAMPLE BASIC PROGRAM

```
 10 CLS:PRINT"ALPHA 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 1200 BAUD 7 BITS EVEN PARITY
 60 REM ( NOTE: 4800 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 ("E" = 460 ONLY)
230 REM "00"             = ALL ADDRESSES RESPOND("01","02" ETC. 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
```

APPENDIX G

NETWORK PIN-OUTS

Below is a list of the ALPHA units and their series grouping. The series is referenced frequently in the pin-out appendix.

AV SERIES - ANY ALPHAVISION SIGN
4000 SERIES - 4120C, 4160C, 4200C, 4120R, 4160R, 4200R
200 SERIES - 215, 215C
"ES" SERIES - 440A, 460A, 480A
"T" SERIES - 210B, 221B, 221C, 430A
700 SERIES - 710, 715

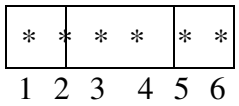
25 POS. FEMALE SUB-D/6 POS. RJ11 ADAPTER (P/N 4370-0001B)

FUNCTION: ADAPTS 25 POS. COMPUTER RS232 COM PORT TO AMS RS232 DATA CABLE

APPLICATION: COMPUTER TO SINGLE SIGN RS232 COMMUNICATIONS. USED IN CONJUNCTION WITH 6 CON. DATA CABLE (P/N 1088-8625 OR 1088-8627). FUNCTIONAL WITH AV, 4000, 200 SERIES AND BETA-BRITE MODELS. BE SURE WHEN USING RS232 COMMUNICATIONS ON THE 4000 SERIES UNITS, THE SHORTING JUMPER LOCATED BELOW THE EPROM INSIDE THE RIGHT ENDCAP OF THE 4000 SERIES UNIT IS IN THE RS232 POSITION. ALSO BE SURE THE CABLE IS PLUGGED INTO THE JACK ON THE REAR OF THE UNIT WHICH IS LABELED RS485/TTL OR RS485/RS232.

25 POS. SUB-D (FEMALE PINS)		6 POS. RJ11
PIN 2	TXD	PIN 4
PIN 3	RXD	PIN 3
PIN 4	RTS	
PIN 5	CTS	
PIN 7	SIG. GND	PIN 6
PIN 6	DSR	
PIN 8	DCD	
PIN 20	DTR	

RJ11 JACK OUTER VIEW



9 POS. FEMALE SUB-D/6 POS. RJ11 ADAPTER (NOT AVAILABLE FROM AMS)

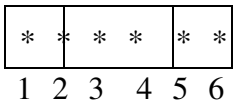
FUNCTION: ADAPTS 9 POS. COMPUTER RS232 COM PORT TO AMS RS232 DATA CABLE

APPLICATION: COMPUTER TO SINGLE SIGN RS232 COMMUNICATIONS. USED IN CONJUNCTION WITH 6 CON. DATA CABLE (P/N 1088-8625 OR 1088-8627). FUNCTIONAL WITH AV, 4000, 200 SERIES AND BETA-BRITE MODELS. BE SURE WHEN USING RS232 COMMUNICATIONS ON THE 4000 SERIES UNITS, THE SHORTING JUMPER LOCATED BELOW THE EPROM INSIDE

THE RIGHT ENDCAP OF THE 4000 SERIES UNIT IS IN THE RS232 POSITION. ALSO BE SURE THE CABLE IS PLUGGED INTO THE JACK ON THE REAR OF THE UNIT WHICH IS LABELED RS485/TTL OR RS485/RS232.

9 POS. SUB-D (FEMALE PINS)		6 POS. RJ11
PIN 3	TXD	PIN 4
PIN 2	RXD	PIN 3
PIN 7	RTS	
PIN 8	CTS	
PIN 5	SIG. GND	PIN 6
PIN 6	DSR	
PIN 1	DCD	
PIN 4	DTR	

RJ11 JACK OUTER VIEW



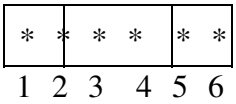
6 CON. DATA CABLE PIN-OUT (P/Ns 1088-8625 & 1088-8627)

FUNCTION: CARRY RS232 DATA FROM COM PORT ADAPTER TO AN RS232 CAPABLE ALPHA UNIT. CABLE ASSEMBLY CONSISTS OF A LENGTH OF 6 CON. DATA CABLE AND (TWO) 6 POS. RJ11 PLUGS.

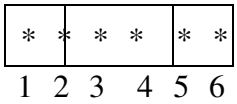
APPLICATION: COMPUTER TO SINGLE SIGN RS232 COMMUNICATIONS. P/N 1088-8625 IS 25 FEET IN LENGTH, 1088-8627 IS 50 FEET IN LENGTH. USED IN CONJUNCTION WITH 25 POS. SUB-D/TO 6 POS. RJ11 ADAPTER (P/N 4370-0001B). FUNCTIONAL WITH AV, 4000, 200 SERIES AND BETA-BRITE MODELS. BE SURE WHEN USING RS232 COMMUNICATIONS ON THE 4000 SERIES UNITS, THE SHORTING JUMPER LOCATED BELOW THE EPROM INSIDE THE RIGHT ENDCAP OF THE 4000 SERIES UNIT IS IN THE RS232 POSITION. ALSO BE SURE THE CABLE IS PLUGGED INTO THE JACK ON THE REAR OF THE UNIT WHICH IS LABELED RS485/TTL.

PLUG 1	PLUG 2
PIN 2	PIN 2
PIN 2	PIN 2
PIN 3	PIN 3
PIN 4	PIN 4
PIN 5	PIN 5
PIN 6	PIN 6

PLUG 1 END VIEW



PLUG 2 END VIEW



TYPE "A" CABLE (P/N 1088-8602)

FUNCTION: CONNECT 25 POS. SUB-D COMPUTER COM PORT TO CONVERTER BOX (P/N 1088-2001NR)

APPLICATION: UTILIZED WHENEVER USING A COMPUTER TO SEND RS232 COMMUNI-CATIONS WHICH NEED TO BE CONVERTED TO RS485 EITHER BECAUSE THERE IS ONE ALPHA UNIT ON THE SYSTEM WHICH IS MORE THAN 50 FEET AWAY, OR THERE ARE MORE THAN ONE ALPHA UNITS ON THE SYSTEM (AV, 4000, 200 SERIES). ON A COMMUNICATIONS NETWORK WHICH CONTAINS "ES" SERIES, "T" SERIES, OR 700 SERIES UNITS, THE TYPE "A" CABLE IS ALWAYS USED EXCEPT IN MODEM APPLICATIONS.

25 POS. SUB-D		25 POS. SUB-D
COM PORT END		CONVERTER BOX END
(FEMALE PINS)		(MALE PINS)
PIN 2	TXD	PIN 2
PIN 3	RXD	PIN 3
PIN 4	RTS	
PIN 5	CTS	
PIN 6	DSR	PIN 6
PIN 7	SIG. GND	PIN 7
PIN 8	DCD	PIN 8
PIN 20	DTR	PIN 20

9 POS. FEMALE SUB-D/25 POS. MALE SUB-D (NOT AVAILABLE FROM AMS)

FUNCTION: CONNECT 9 POS. SUB-D COMPUTER COM PORT TO CONVERTER BOX

APPLICATION: UTILIZED WHENEVER USING A COMPUTER TO SEND RS232 COMMUNI-CATIONS WHICH NEED TO BE CONVERTED TO RS485 EITHER BECAUSE THERE IS ONE ALPHA UNIT ON THE SYSTEM WHICH IS MORE THAN 50 FEET AWAY, OR THERE IS MORE THAN ONE ALPHA UNIT ON THE SYSTEM (AV, 4000 SERIES, 200 SERIES). ON A COMMUNICATIONS NETWORK WHICH CONTAINS "ES" SERIES, "T" SERIES, OR 700 SERIES UNITS, THIS IS A SUBSTITUTE FOR A TYPE "A" CABLE, WHICH IS USED IN ALL EXCEPT MODEM APPLICATIONS.

POS. SUB-D		25 POS. SUB-D
COM PORT END		CONVERTER BOX END
(FEMALE PINS)		(MALE PINS)
PIN 3	TXD	PIN 2
PIN 2	RXD	PIN 3
PIN 7	RTS	
PIN 8	CTS	
PIN 6	DSR	PIN 6
PIN 5	SIG. GND	PIN 7
PIN 1	DCD	PIN 8
PIN 4	DTR	PIN 20

TYPE "B" CABLE (P/N 1088-8610)

FUNCTION: CONNECT MODEM COM PORT TO CONVERTER BOX (P/N 1088-2001NR)

APPLICATION: UTILIZED WHENEVER USING A MODEM TO SEND RS232 COMMUNICATIONS WHICH NEED TO BE CONVERTED TO RS485 EITHER BECAUSE THERE IS ONE ALPHA UNIT ON THE SYSTEM WHICH IS MORE THAN 50 FEET AWAY, OR THERE ARE MORE THAN ONE ALPHA UNITS ON THE SYSTEM.

25 POS. SUB-D		25 POS. SUB-D
MODEM END		CONVERTER BOX END
(MALE PINS)		(MALE PINS)
PIN 1	GND	PIN 1
PIN 2	RXD	PIN 3
PIN 3	TXD	PIN 2
PIN 7	SIG. GND	PIN 7
PIN 8	DCD	PIN 20
PIN 20	DTR	PIN 8

9 POS. SUB-D/25 POS. SUB-D ADAPTER (NOT AVAILABLE FROM AMS)

FUNCTION: ADAPT 9 POS. SUB-D COM PORT TO 25 POS. SUB-D

APPLICATION: CONVERT 9 POS. SUB-D COM PORT ON COMPUTER TO 25 POS. SUB-D FOR TYPE "A" CABLE (P/N 1088-8602). THIS IS AN INDUSTRY STANDARD ADAPTER CABLE.

9 POS. SUB-D (FEMALE PINS)		25 POS. SUB-D (MALE PINS)
PIN 1	DCD	PIN 8
PIN 2	RXD	PIN 3
PIN 3	TXD	PIN 2
PIN 4	DTR	PIN 20
PIN 5	SIG. GND	PIN 7
PIN 6	DSR	PIN 6
PIN 7	RTS	PIN 4
PIN 8	CTS	PIN 5
PIN 9		PIN 22

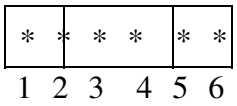
25 POS. SUB-D/TO 6 POS. RJ11 ADAPTER (MODEM APPLICATION) (NOT AVAILABLE FROM AMS)

FUNCTION: ADAPTS 25 POS. MODEM RS232 COM PORT TO AMS RS232 DATA CABLE

APPLICATION: MODEM TO SINGLE SIGN RS232 COMMUNICATIONS. USED IN CONJUNCTION WITH 6 CON. DATA CABLE (P/N 1088-8625 OR 1088-8627). BE SURE WHEN USING RS232 COMMUNICATIONS ON THE 4000 SERIES UNITS, THE SHORTING JUMPER LOCATED BELOW THE EPROM INSIDE THE RIGHT ENDCAP OF THE 4000 SERIES UNIT IS IN THE RS232 POSITION. ALSO, BE SURE THE CABLE IS PLUGGED INTO THE JACK ON THE REAR OF THE UNIT WHICH IS LABELED RS485/TTL.

25 POS. SUB-D (MALE PINS)		6 POS. RJ11
PIN 2	RXD	PIN 3
PIN 3	TXD	PIN 4
PIN 7	SIG. GND	PIN 6
PIN 8	DCD	
PIN 20	DTR	
PIN 6	DSR	

RJ11 JACK OUTER VIEW



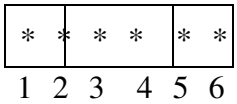
MODULAR NETWORK ADAPTER (P/N 1088-9103)

FUNCTION: A CONNECTION POINT FOR AN AV, 4000, OR 200 SERIES SIGN IN AN RS485 NETWORK.

APPLICATION: THIS IS PLACED IN-LINE WITH THE RS485 NETWORK TWISTED PAIR/ SHIELDED CABLE. IT IS IMPORTANT THAT PINS 2 AND 3 OF THE RJ11 JACK DO NOT CONNECT TO ANYTHING OR SHORT TOGETHER. IT IS RECOMMENDED THAT THESE WIRES ARE CLIPPED OFF WITHIN THE MODULAR NETWORK ADAPTER. ALSO, BE SURE THAT THE INCOMING SHIELD WIRE IS CONNECTED TO THE OUTGOING SHIELD WIRE.

4 POS. RJ11 JACK TWISTED PAIR/SHIELDED WIRE
PIN 1 _____(-)RS485_____ RED WIRE
PIN 2 _____ NO CONNECT
PIN 3 _____ NO CONNECT
PIN 4 _____(+)RS485_____ BLACK WIRE
NO CONNECT ____ SHIELD WIRE

RJ11 JACK OUTER VIEW



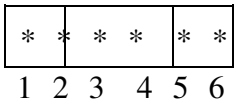
EIGHT FOOT 4-CONDUCTOR DATA CABLE PIN-OUT (P/N 1088-8624)

FUNCTION: CARRY RS485 DATA FROM MODULAR NETWORK ADAPTER TO AN ALPHA 4000 SERIES OR 200 SERIES UNIT. CABLE ASSEMBLY CONSISTS OF AN 8 FOOT LENGTH OF 4 CON. DATA CABLE AND (TWO) 4 POS. RJ11 PLUGS.

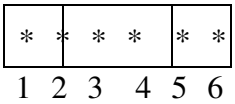
APPLICATION: USED IN CONJUNCTION WITH MODULAR NETWORK ADAPTER (P/N 1088- 9103). FUNCTIONAL WITH AV, 4000, AND 200 SERIES MODELS. BE SURE WHEN USING RS485 COMMUNICATIONS ON THE 4000 SERIES UNITS, THE SHORTING JUMPER LOCATED BELOW THE EPROM INSIDE THE RIGHT ENDCAP OF THE 4000 SERIES UNIT IS IN THE RS485 POSITION. THE CABLE CAN BE PLUGGED INTO EITHER JACK ON THE REAR OF THE 4000 OR 200 SERIES UNIT OR THE TOP OF THE AV UNIT.

PLUG 1 PLUG 2
PIN 1 _____ PIN 1
PIN 2 _____ PIN 2
PIN 3 _____ PIN 3
PIN 4 _____ PIN 4

PLUG 1 END VIEW



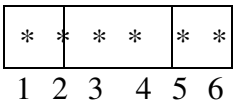
PLUG 2 END VIEW



ALPHA UNIT COM PORT PIN-OUTS

RJ11 SOCKET OUTER VIEW (RS485) AVAILABLE ON ALPHA 4000 SERIES*, AV SERIES, 200 SERIES

PIN 1 - NO CONNECT
PIN 2 - (-) RS485 (RED NETWORK WIRE)
PIN 3 - NO CONNECT
PIN 4 - NO CONNECT
PIN 5 - (+) RS485 (BLACK NETWORK WIRE)
PIN 6 - NO CONNECT



RJ11 SOCKET OUTER VIEW (RS485/TTL/RS232) AVAILABLE ON ALPHA 4000 SERIES*, AV SERIES, 200 SERIES

PIN 1 - +5 VOLTS (200mA MAX) - see NOTE 2
PIN 2 - (-) RS485 (RED NETWORK WIRE)
PIN 3 - TXD (TTL)-TRUE RS232 ON 4000 SERIES ONLY
PIN 4 - RXD (RS232)
PIN 5 - (+) RS485 (BLACK NETWORK WIRE)
PIN 6 - GND

RJ11 SOCKET OUTER VIEW (TTL) AVAILABLE ON BETA-BRITE

PIN 1 - +5 VOLTS (200mA MAX) - see NOTE 2
PIN 2 - NO CONNECT
PIN 3 - TXD (TTL)
PIN 4 - RXD (RS232)
PIN 5 - NO CONNECT
PIN 6 - GND

* NOTE 1: 4000 SERIES REQUIRES THAT A JUMPER BE POSITIONED FOR RS232 OR RS485 COMMUNICATIONS. THIS JUMPER IS LOCATED BELOW THE EPROM INSIDE THE RIGHT ENDCAP OF THE UNIT. THESE UNITS LEAVE ADAPTIVE MICRO SYSTEMS WITH THE JUMPER IN THE RS485 POSITION.

** NOTE 2: 200 SERIES AND BETA-BRITE UNITS REQUIRE JUMPER JP3 (LOCATED NEAR THE EPROM INSIDE THE BACK COVER) TO BE INSTALLED FOR +5V. THESE UNITS LEAVE ADAPTIVE MICRO SYSTEMS WITHOUT THE JUMPER INSTALLED.

APPENDIX H

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 <STX> = 02H <EOT> = 04H
<SOH> = 01H <ETX> = 03H

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

<NUL> represents twenty <NUL>s
x20

1.0 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 430A units with the address "10H" thru "1FH":

<NUL><NUL><NUL><NUL><NUL><SOH>C1?<STX>AAHELLO<EOT>

1.1 Transmission Frame Variations

A. With Checksum Field

<NUL><NUL><NUL><NUL><NUL><SOH>ZOO<STX>AAHELLO<ETX>01F6<EOT>

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

B. Nesting With Checksums

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

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

C. Nesting Without Checksums

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

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

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.

APPENDIX H (cont.)

2.0 TEXT FILES

2.1 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 Section 2.3 (below) for message center response.

2.2 Response to Read TEXT File
This is a response to the example in Section 2.1 (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."

2.3 TEXT File Data Format

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>AD(1B)&aHELLO<EOT>
|_____|
Mode Field

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>
|____| |____|
Mode Fields

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)"n2Hello There(1B)"a(1B)&n8<EOT>
|_____| |____||_____|
Mode Fields

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>
|_____|
Mode Field

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>
|_____| |_____| |_____| |_____|
Mode Fields

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

To display a counter value, the following transmission frame may be sent:

<NUL><NUL><NUL><NUL><NUL><SOH>h00<STX>A1(1B)0bCongratulations!(0D)(08)z days without an accident!<EOT>
|__| Display Counter "1" Value

APPENDIX H (cont.)

Target TEXT file "1" will hold the word "Congratulations!" on the top line of the 4160R (type code "h") and "xxxxxxxx days without an accident!" on the bottom line (xxxxxxxx is the eight digit counter "1" value). Leading zeros in the counter value are not displayed.

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)1SMALL(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).

2.5 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.

3.0 SPECIAL FUNCTIONS

3.1 Write Special Functions
<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>E 0830<EOT>

Writes the time-of-day "0830" to all message centers.

3.2 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 Section 3.3 (below) for message center response.

3.3 Response to Read Special Functions

This is a response to the example in Section 3.2 (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).

3.4 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.

APPENDIX H (cont.)

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 b c d e f

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

Memory Configuration

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>E$AAU0100FF00mDU073C10001BL000A0000<EOT>
|_____||_____||_____|
a b c

a = TEXT file "A" data field
b = DOTS PICTURE file "m" data field
c = STRING file "1" data field

Writes to all message centers the following:
- TEXT file "A" (unlocked), 100H (256D) bytes in length, to run always.
- DOTS PICTURE file "m" (unlocked), 7 pixel rows by 3CH (60D) pixel columns, one color.
- STRING file "1" (locked), 0AH (10D) bytes in length.

If you wish to use counter functions, it is important to include the five TARGET TEXT files labeled "1" thru "5". The same Memory Configuration as above, but written to a message center using counters should look as follows:

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

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>E$AAU0100FF00mDU073C10001BL000A0000

1AU0064FE002AU0064FE003AU0064FE004AU0064FE005AU0064FE00<EOT>

APPENDIX H (cont.)

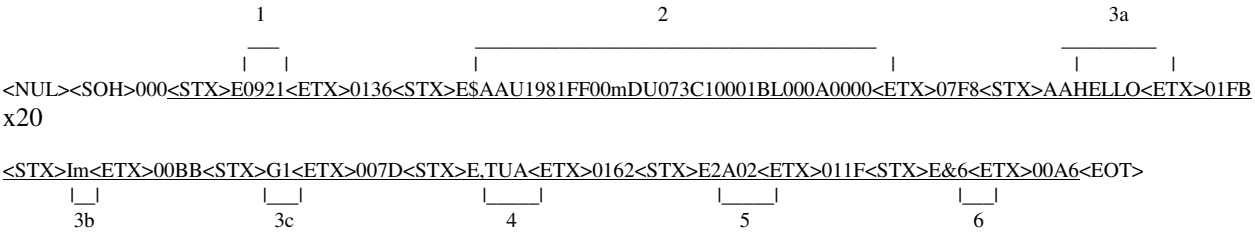
The additional five TARGET TEXT files are "unlocked" and are 100 bytes (64H) in length. They are set up with a Run Start Time of "never". Once a Counter Value reaches its Target Value, all Target files to be triggered (as set up in the Target file byte) will have their Run Start Times modified, automatically, by the message center to "always", and begin running.

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.

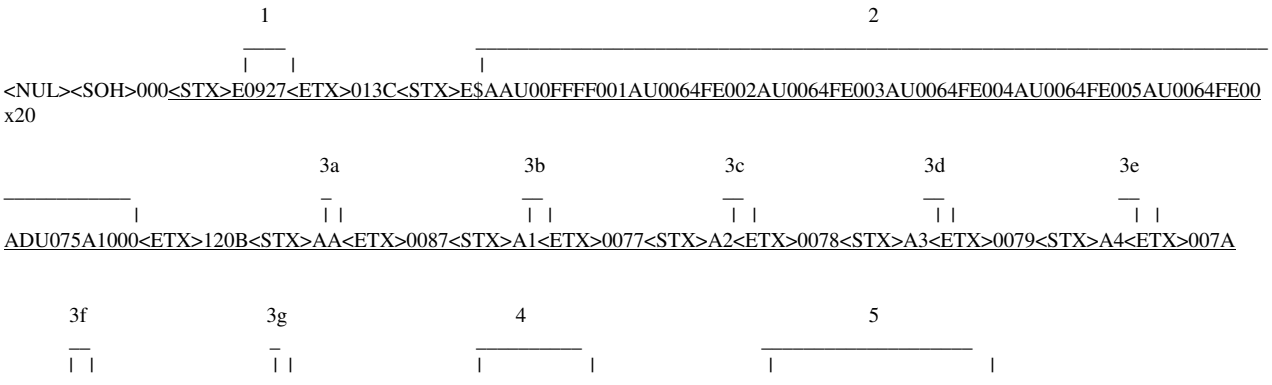


1 - Units time-of-day (9:21 a.m.)
2 - Memory configuration
- TEXT file "A" (unlocked), 1981H (6529D) bytes in length, to run "always"
- DOTS PICTURE file "m" (unlocked), 7 pixel rows by 3CH (60D) pixel columns, one color
- STRING file "1" (locked), 0AH (10D) bytes in length

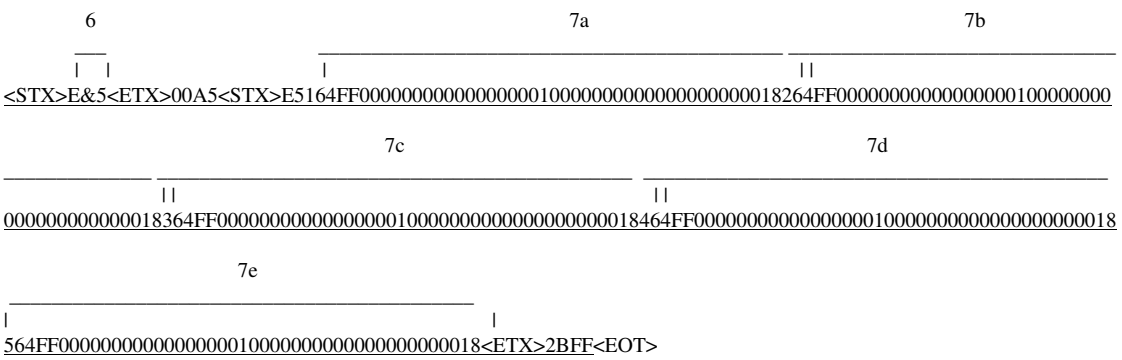
3a- TEXT file "A" contents ("HELLO")
3b- DOTS PICTURE file "m" contents (blank or void)
3c- STRING file "1" contents (blank or void)
4 - Run Sequence (execute according to listed TEXT file run times, unlocked, TEXT file "A" listed only)
5 - Run Day Table (TEXT file "A", daily)
6 - Units day-of-week ("6" is Friday)

COUNTER FEATURE

On a 4160C message center the default response (with nothing programmed) would appear as below (NOTE: The following transmisison in one long string of data. Although it appears on six lines, it is concatenated.):



<STX>A5<ETX>007B<STX>IA<ETX>008F<STX>E,TUA12345<ETX>0261<STX>E2A02102202302402502<ETX>0408



Where:

1	-	Units time-of-day (9:27 a.m.)
2	-	Memory Configuration
		- TEXT file "A" (unlocked), 00FFH (255D) bytes in length, to run "always".
		- Target TEXT file "1" (unlocked), 0064H (100D) bytes in length, to run "never"
		- Target TEXT file "2" (unlocked), 0064H (100D) bytes in length, to run "never"
		- Target TEXT file "3" (unlocked), 0064H (100D) bytes in length, to run "never"
		- Target TEXT file "4" (unlocked), 0064H (100D) bytes in length, to run "never"
		- Target TEXT file "5" (unlocked), 0064H (100D) bytes in length, to run "never"
		- DOTS PICTURE file "A" (unlocked), 7 pixel rows by 5AH (90D) pixel columns, one color
3a	-	TEXT file "A" contents (blank)
3b	-	Target TEXT file "1" contents (blank)
3c	-	Target TEXT file "2" contents (blank)
3d	-	Target TEXT file "3" contents (blank)
3e	-	Target TEXT file "4" contents (blank)
3f	-	Target TEXT file "5" contents (blank)
3g	-	DOTS PICTURE file "A" contents (blank or void)
4	-	Run Sequence (execute according to listed TEXT file run times, unlocked, TEXT files "A" and "1" thru "5" listed)
5	-	Run Day Table (TEXT file "A" and "1" thru "5", daily)
6	-	Units Day-of-week ("5" is Thursday)
7a	-	Counter "1" followed by Counter "1" data
7b	-	Counter "2" followed by Counter "2" data
7c	-	Counter "3" followed by Counter "3" data
7d	-	Counter "4" followed by Counter "4" data
7e	-	Counter "5" followed by Counter "5" data

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.

Run Time Table
<NUL><NUL><NUL><NUL><NUL><SOH>100<STX>E)A3069B6978<EOT>
|____| |____|
| |
File "A" with Start (30) and Stop (69) times |
File "B" 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 |____| | |TEXT «B» disabled
PRIORITY TEXT file not running /
TEXT «A» enabled

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<EOT>

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 TEXT file "B" field

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").

On message center models equipped with the COUNTER UPGRADE, the five Target TEXT files need to be included as part of the Run Day Table. See below for the default Run Day Table:

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>E2A02102202302402502<EOT>
TEXT file «A» field = |__| |__||__| |__| |__||__|
| | | | |
TEXT files «1" thru «5"

TEXT file "A" is set up to run daily ("0") as are all the Target files. All Run Stop days ("2") are ignored.

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.

COUNTER Functions

NOTE: The following transmisison is one long string of data. Although it appears on five lines, it is concatenated.

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>E5

1E2FF0000000000000000010000000300000060150 60D
2943063000060000000000100006000000000550 2060D
364FF000000000000000001000000000000000000000018
464FF000000000000000001000000000000000000000018
564FF000000000000000001000000000000000000000018<ETX>2C37<EOT>
/ \ /\ /\ /|_____||_____| |_____| |_____| \ /\ /\ /
/ / | | / / / / / / / \
1 2 3 4 5 6 7 8 9 10 11

The five lines in the transmission frame above are arranged so the COUNTER Functions data is columnar for each of the five counters (for readability).

1 - Counter Number ("1" thru "5")

2 - Counter Control Byte -

	Counter 1	= "E2" = 11100010B -	Counter on,increment,count minutes,weekends off,auto-reload on
	Counter 2	= "94" = 10010100B -	Counter on,decrement,count hours, weekends on, auto-reload off
*	Counters 3,4,5	= "64" = 01100100B -	Counter off,increment,count minutes,weekends on,auto-reload off

3 - Counter Start Time -

	Counter 1	= "FF" = "always"
	Counter 2	= "30" = 8:00 a.m.
*	Counter 3,4,5	= "FF" = "always"

* Indicates Default Settings
APPENDIX H (cont.)

4 - Counter Stop Time -
Counter 1 = "00" = ignored since Start Time is "always"
Counter 2 = "63" = 4:30 p.m.
* Counter 3,4,5 = "00" = ignored since Start Time is "always"

5 - Counter Start Value -
Counter 1 = "00000000" = 0
Counter 2 = "00006000" = 6,000
* Counter 3,4,5 = "00000000" = 0

6 - Counter Change (Increment/Decrement) Value
Counters 1 - 5 = "00000001" = 1

7 - Current Counter Value
Counter 1 = "00000003" = 3 (has incremented 3 minutes)
Counter 2 = "00006000" = 6,000 (hasn't begun decrementing yet)
* Counter 3,4,5 = "00000000" = 0

8 - Counter Target Value
Counter 1 = "00000060" = 60
Counter 2 = "00000055" = 55
* Counter 3,4,5 = "00000000" = 0

9 - Target File Byte
Counter 1 = "15" = 00010101B = Target files "1", "3", and "5" will trigger
Counter 2 = "02" = 00000010B = Target file "4" will trigger
* Counter 3,4,5 = "00" = 00000000B = No Target files are set to trigger

10 Counter Change Synchronization - Minutes
Counter 1 = "06" = Ignored since we're counting minutes
Counter 2 = "06" = Hour counter will change at six minutes past the hour
* Counter 3,4,5 = "00" = Ignored since we're counting minutes

11 Counter Change Synchronization - Hours
Counter 1,2 = "0D" = 1:00 p.m. (Ignored since we're not counting days)
* Counter 3,4,5 = "18" = 12:00 a.m. (Ignored since we're not counting days)

4.0 STRING FILES

4.1 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".

4.2 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".

* Indicates Default Settings

4.3 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."

4.4 STRING File Data Format

Shown in STRING FILES, Sections 6.4 (page 35). See Appendix D (page 46) - STRING file Application notes for further information.

5.0 DOTS PICTURE FILES

5.1 Writes Dots Picture File

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

<NUL><NUL><NUL><NUL><NUL><SOH>B00<STX>IA0F09000000000(0D)000000000(0D)000100000(0D)000110000(0D)000111000(0D)000111100(0D)111111110(0D)111111111(0D)111111110(0D)000111100(0D)000111000(0D)000110000(0D)000100000(0D)000000000(0D)000000000(0D)<EOT>

Writes to the DOTS PICTURE file labeled "A" a fifteen pixel high "0F" by nine wide ("09") picture. In this case, an arrow which points to the right. When "called" from a TEXT file, it would appear on the message center model 4160C as shown below:

```
---------
---------
---1-----
---11----
---111---
---1111--
11111111-
111111111
11111111-
---1111--
---111---
---11----
---1-----
---------
---------
```

NOTE: Each "-" represents a "0" for readability

Sign Type (f=215C)
Serial Address
5.2 Read DOTS PICTURE File /\
<NUL><NUL><NUL><NUL><NUL><SOH>f02<STX>JA<EOT>

Reads the data contained in the DOTS PICTURE file labeled "A" from the message center model 215C with serial address "02".

5.3 Response to Read DOTS PICTURE File

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

File Label
Height in Hex
Width in Hex

<NUL><SOH>000<STX>IA07100011111000777770(0D)0120011007100770(0D)1111101077777070(0D)1020101070107070(0D)1022121070117170(0D)
1200110071007700(0D)1111100077777000(0D)<ETX>17A4<EOT>

The message center model 215C with serial address "02" will respond with the data found in the DOTS PICTURE file labeled "A". In this case, the DOTS PICTURE file consists of a seven pixel high (07H) by sixteen pixel wide (10H) picture of two multicolor cubes next to each other as roughly illustrated below:

NOTE: Each "-" represents a "0" for readability.

--11111---77777- "1" (31H) - Red
-12--11--71--77- "2" (32H) - Green
11111-1-77777-7- "3" (33H) - Amber
1-2-1-1-7-1-7-7- "4" (34H) - Dim Red
1-22121-7-11717- "5" (35H) - Dim Green
12--11--71--77-- "6" (36H) - Brown
11111---77777--- "7" (37H) - Orange
"8" (38H) - Yellow

5.4 DOTS PICTURE File Data Format

Refer to sections 5.1 and 5.3.

Call Far Dots File From TEXT File
<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>AA<ESC> b(1F)CFILENAME10020<EOT>

Write Far Dots Configuration Table
<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>E8FILENAME1U0040006004<EOT>

5.5.1 Write Far Dots PICTURE File

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

<NUL><NUL><NUL><NUL><NUL><SOH>Z00<STX>MFILE
NAME1000F0009000000000(0D)000000000(0D)000100000(0D)000110000(0D)000111000(0D)000111100
(0D)111111110(0D)111111111(0D)111111110(0D)000111100(0D)000111000(0D)000110000(0D)000100000
(0D)000000000(0D)000000000(0D)<EOT>