ALPHA™ Sign Communications Protocol

Revision C

Contents

For Y2K information, see "Y2K NOTE" on page 17.

This document explains how to use the EZ KEY II and EZ95 protocols communications protocols to send messages and graphics to ALPHA signs.

ADAPTIVE

Adaptive Micro Systems • 7840 North 86th Street • Milwaukee, WI 53224 USA • 414-357-2020 • 414-357-2029 (fax) • www.ams-i.com

Form number 9708-8061C • Revised: 7/1/99

NOTE: Due to continuing product innovation, specifications in this document are subject to change without notice.

Trademarked names appear throughout this document. Rather than list the names and entities that own the trademarks or insert a trademark symbol with each mention of the trademarked name, the publisher states that it is using the names for editorial purposes and to the benefit of the trademark owner with no intention of improperly using the trademark.

BetaBrite, Big Dot, and Smart Alec are trademarks of Adaptive Micro Systems, Inc. registered in the United States Patent and Trademark Office.

ALPHA, AlphAlert, AlphaNet, AlphaNet *plus*, AlphaNet *plus* II, ALPHAVISION, Automode, BetaBrite Director, Director, BB Messaging Software, EZ KEY II, EZ95, PagerNET, PPD, PrintPak, Solar, and TimeNet are trademarks of Adaptive Micro Systems, Inc.

Visit our Internet World Wide Web site: http://www.ams-i.com

1.0 Document information

1.1 Revision history

Table 1: Revision history

Revision date	Document part number	Notes
May 17, 1995	9708-8061	First release.
August 4, 1995	9708-8061A	• PrintPak information added • Printable character terminations added • Identifier page with revision list added
May 1, 1998	9708-8061B	• Document reformatted
May 28, 1998	9708-8061B	• Corrections to 5/1/98 release.
July 1, 1999	9708-8061C	• Various corrections to 5/28/98 release. • "POCSAG" changed to "ASCII Printable" • PrintPak protocol information removed • Y2K date correction information added

1.2 Document conventions

Table 2: Document conventions

Convention	Description
{{ Begin Message }}	Indicates text that must be entered as shown
<SOH>	ASCII control character abbreviation (see page 72)
"A"	ASCII character (in this case the letter A)
11D	Decimal number (in this case, 11). Numbers that are not followed by any letter are also decimal.
0BH	Hexadecimal number (0B hex = 11 decimal)
01001100B	Binary number

2.0 Introduction

This document is designed to allow a user to understand how to communicate with the ALPHA line of electronic signs manufactured by Adaptive Micro Systems. The signs must have the ALPHA firmware (EPROM) installed. The standard ALPHA EPROM contains two protocols by which you may communicate with a ALPHA sign:

- EZ KEY II protocol
- EZ95 protocol

These protocols were also engineered to facilitate the transmitting of messages, but also have the capabilities to send counters, pictures, and more.

3.0 EZ KEY II / EZ95 protocol

The ALPHA line of products supports several types of files and a number of special functions which are used for specific applications:

TEXT files

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 files

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. DOTS PICTURE files have a maximum size of 31 x 255 pixels.

ALPHAVISION DOTS PICTURE files

ALPHAVISION DOTS PICTURE files are 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. ALPHAVISION DOTS PICTURE files have a maximum size of 65535 x 65535 pixels.

STRING files

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.

3.1 Transmission frame formats

3.1.1 Standard transmission frame specifications and format

This section describes the basic outline of transmissions on an EZ KEY II/ EZ95 network:

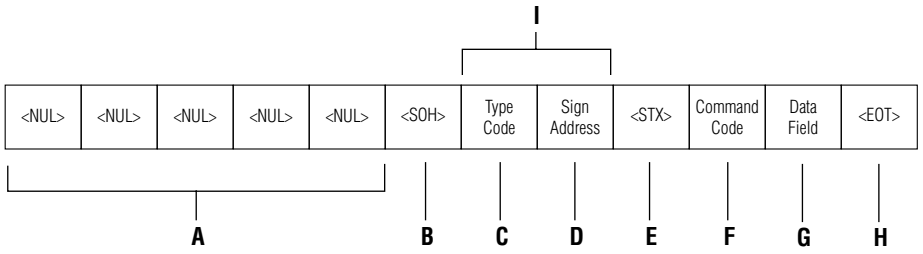
Table 3: EZ KEY II / EZ95 / ASCII Printable transmission specifications

Baud rate:	1200, 2400, 4800, or 9600 baud (9600 only for EZ95 / ASCII Printable)
Data bits:	7 (8 for EZ95 / ASCII Printable)
Start bits:	1
Stop bits:	2 (1 for EZ95 / ASCII Printable)
Parity:	Even (None for EZ95 / ASCII Printable)
Time-out period:	1 second (any delays between bytes cannot exceed this)

SHOW ME

An example of the Standard transmission frame is on page 48.

Table 4: Standard transmission frame format



Item	Name	Description
A	<NUL>	A minimum of five <NUL>s (00H) must be transmitted as frame synchronization characters. Five <SOH>s (01H) may be substituted for the five <NUL>s. The sign uses these five characters to establish the baud rate.
B	<SOH>	The <SOH> (01H) is the "Start Of Header" ASCII character.
C	Type Code	A single ASCII character (to send multiple Type Codes, see item I): "!" 21H All signs with Visual Verification. This code causes a sign to display the TRANSMISSION OK message when a transmission frame is received without an error. Otherwise, TRANSMISSION ERROR will appear. """" 22H Serial clock "#" 23H ALPHAVISION sign "$" 24H Full matrix ALPHAVISION sign "%" 25H Character matrix ALPHAVISION sign "&" 26H Line matrix ALPHAVISION "0" 30H Response code — used only when a sign responds to a request. "1" 31H One-line signs "2" 32H Two-line signs "?" 3FH All signs "C" 43H 430i sign "D" 44H 440i sign "E" 45H 460i sign "U" 55H 790i sign "Z" 5AH All signs "^" 5EH BETA-BRITE sign "a" 61H 4120C sign "b" 62H 4160C sign "c" 63H 4200C sign "d" 64H 4240C sign "e" 65H 215 sign "f" 66H 215C sign "g" 67H 4120R sign "h" 68H 4160R sign "i" 69H 4200R sign "j" 6AH 4240R sign "k" 6BH 300 series sign "l" 6CH 7000 series sign "m" 6DH 96x16 matrix Solar sign "n" 6EH 128x16 matrix Solar sign "o" 6FH 160x16 matrix Solar sign "p" 70H 192x16 matrix Solar sign "q" 71H PPD sign "r" 72H Director sign "t" 74H 4080C sign "u" 75H 210C and 220C signs "z" 7AH All signs will first configure memory for 26 files ("A" - "Z") of 150 characters each and then execute the specified command.
D	Sign Address	The identifier or "address" of the sign represented by two ASCII digits as a number between "00" and "FF" (0 to 255). Address "00" is reserved as a broadcast address. The wildcard character "?" (3FH) can be used to send messages to a range of addresses. For example, a Sign Address of "0?" will access signs with address between 01H and 0FH (1 and 15). To send multiple Sign Addresses, see item I.

Table 4: Standard transmission frame format

E	<STX>	"Start of TeXt" (02H) character. <STX> always precedes a Command Code. **NOTE:** When nesting frames, there must be at least a 100 millisecond delay after the <STX>.
F	Command Code	One ASCII character that defines the transmission and data types: "A" 41H Write TEXT file (see page 12) "B" 42H Read TEXT file (see page 13) "E" 45H Write SPECIAL FUNCTIONS (see page 15) "F" 46H Read SPECIAL FUNCTIONS (see page 21) "G" 47H Write STRING file (see page 28) "H" 48H Read STRING file (see page 29) "I" 49H Write DOTS PICTURE file (see page 30) "J" 4AH Read DOTS PICTURE file (see page 32) "M" 4DH Write ALPHAVISION DOTS PICTURE file (page 33) "N" 4EH Read ALPHAVISION DOTS PICTURE file (page 34) "O" 4FH Write ALPHAVISION BULLETIN message (page 35) **NOTE:** When nesting commands, only one "Read" Command Code may be used, and it must be the last Command Code before the <EOT>. **NOTE:** The "Write SPECIAL FUNCTIONS" to Speaker Tone Generation must be the last command in a nested string.
G	Data Field	Made up of ASCII characters. The Data Field format is dependent on the preceding Command Code.
H	<EOT>	"End Of Transmission" (04H) character
I	Multiple Type Codes and Sign Address	Instead of sending a single Type Code and Sign Address (like "g02"), multiple Type Codes and Sign Addresses can be transmitted using the following format: `Aaa,Bbb,Ccc, . . .` where A, B, and C are ASCII Type Codes and aa, bb, cc are ASCII Sign Addresses separated by commas (2CH), for example, `g02,U01,21F,220`

3.1.2 Transmission frame with Checksum

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

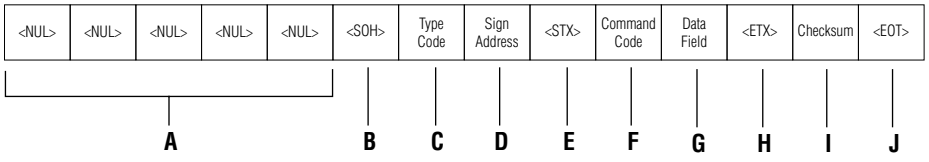
If an <ETX> character is transmitted before the <EOT>, the sign will expect a Checksum.

When a sign receives an invalid Checksum, the associated data will not be processed.

SHOW ME

An example of the Transmission frame with Checksum is on page 49.

Table 5: Transmission frame with Checksum format



Item	Name	Description
A	<NUL>	See Table 4, "Standard transmission frame format," on page 5.
B	<SOH>	
C	Type Code	
D	Sign Address	
E	<STX>	
F	Command Code	
G	Data Field	
H	<ETX>	"End of TeXt" (03H) character

Table 5: Transmission frame with Checksum format

I	Checksum	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first. **NOTE:** When a sign receives an invalid Checksum, the associated data will not be processed.
J	<EOT>	See Table 4, "Standard transmission frame format," on page 5.

3.1.2.1 Nesting with Checksums

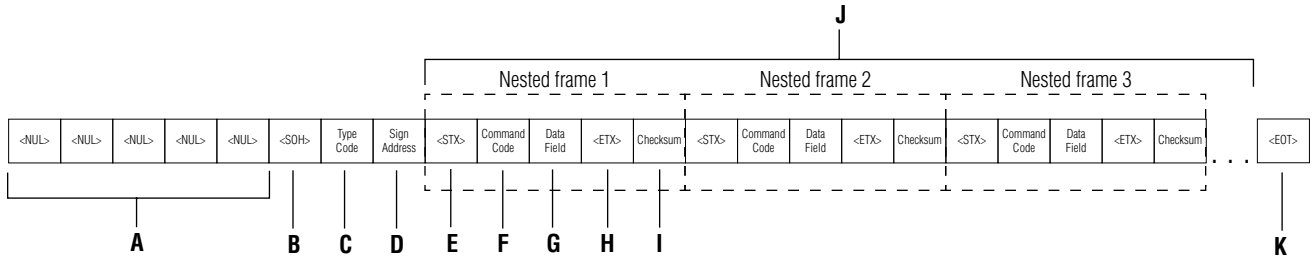
SHOW ME

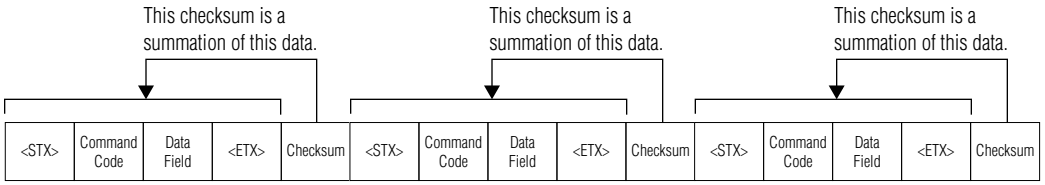
An example of the Nesting with Checksums is on page 50.

If more than one transmission frame is required consecutively, multiple Commands can be repeated or "nested" within a transmission frame.

A sign uses this format when a Memory Dump [see "Read SPECIAL FUNCTIONS Command Code — "F" (46H)" on page 21] is requested serially.

Table 6: Nesting with Checksums transmission frame



Item	Name	Description
A	<NUL>	See Table 4, "Standard transmission frame format," on page 5.
B	<SOH>	
C	Type Code	
D	Sign Address	
E	<STX>	
F	Command Code	
G	Data Field	
H	<ETX>	"End of TeXt" (03H) character
I	Checksum	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first. For example, in the following three nested frames, this is how the checksums are calculated:  **NOTE:** When a sign receives an invalid Checksum, the associated data will not be processed. **NOTE:** When nesting frames, there must be at least a 100 millisecond delay after the <STX>. **NOTE:** When nesting commands, only one "Read" Command Code may be used, and it must be the last Command Code before the <EOT>. **NOTE:** The "Write SPECIAL FUNCTIONS" to Speaker Tone Generation must be the last command in a nested string.
J	Nested Commands with Checksums	Multiple Commands can be "nested" in a transmission frame. This is the format of the nested frame *with a Checksum*: 
K	<EOT>	See Table 4, "Standard transmission frame format," on page 5.

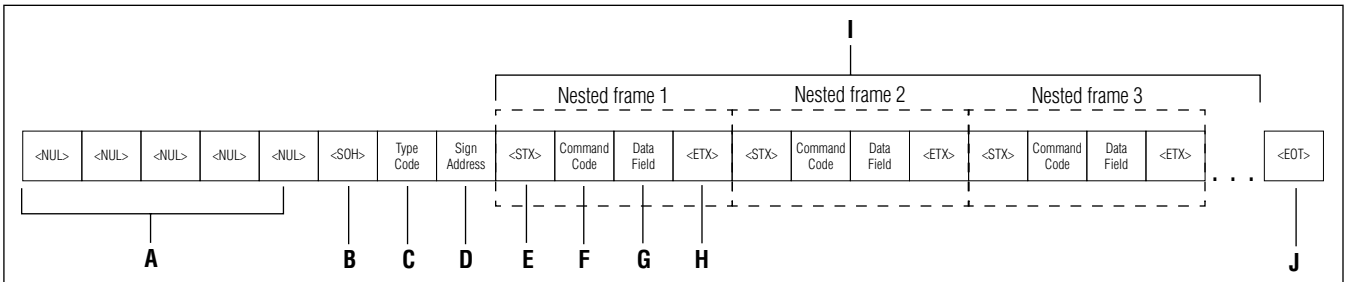
3.1.2.2 Nesting without Checksums

SHOW ME

An example of the Nesting without Checksums is on page 51.

If an <STX> is transmitted immediately following an <ETX>, the sign will expect the next "nested" command.

Table 7: Nesting without Checksums transmission frame



Item	Name	Description
A	<NUL>	See Table 4, "Standard transmission frame format," on page 5.
B	<SOH>	
C	Type Code	
D	Sign Address	
E	<STX>	
F	Command Code	
G	Data Field	
H	<ETX>	"End of TeXt" (03H) character
I	Nested Commands without Checksums	Multiple Commands can be "nested" in a transmission frame. This is the format of the nested frame *without a Checksum*: <STX> | Command Code | Data Field | <ETX> **NOTE:** When nesting frames, there must be at least a 100 millisecond delay after the <STX>. **NOTE:** When nesting commands, only one "Read" Command Code may be used, and it must be the last Command Code before the <EOT>. **NOTE:** The "Write SPECIAL FUNCTIONS" to Speaker Tone Generation must be the last command in a nested string.
J	<EOT>	See Table 4, "Standard transmission frame format," on page 5.

3.1.3 ASCII Printable (or pager) transmission format

SPECIAL NOTE

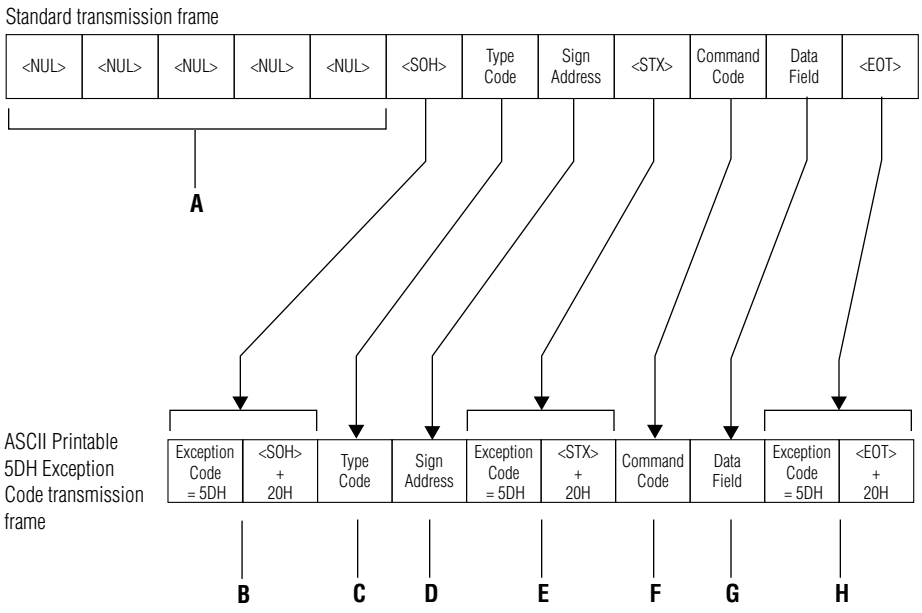
For ASCII Printable baud rate, parity, etc., see Table 3, "EZ KEY II / EZ95 / ASCII Printable transmission specifications," on page 4.

Many pagers and computer systems cannot receive or send ASCII control codes (characters lower than 20H). This variation of the transmission frame allows the entire EZ KEY II / EZ95 protocol to be transmitted without sending any ASCII control codes and thus allowing its use with pagers.

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.

3.1.3.1 ASCII Printable 5DH Exception Code transmission frame format ("2-byte" code)

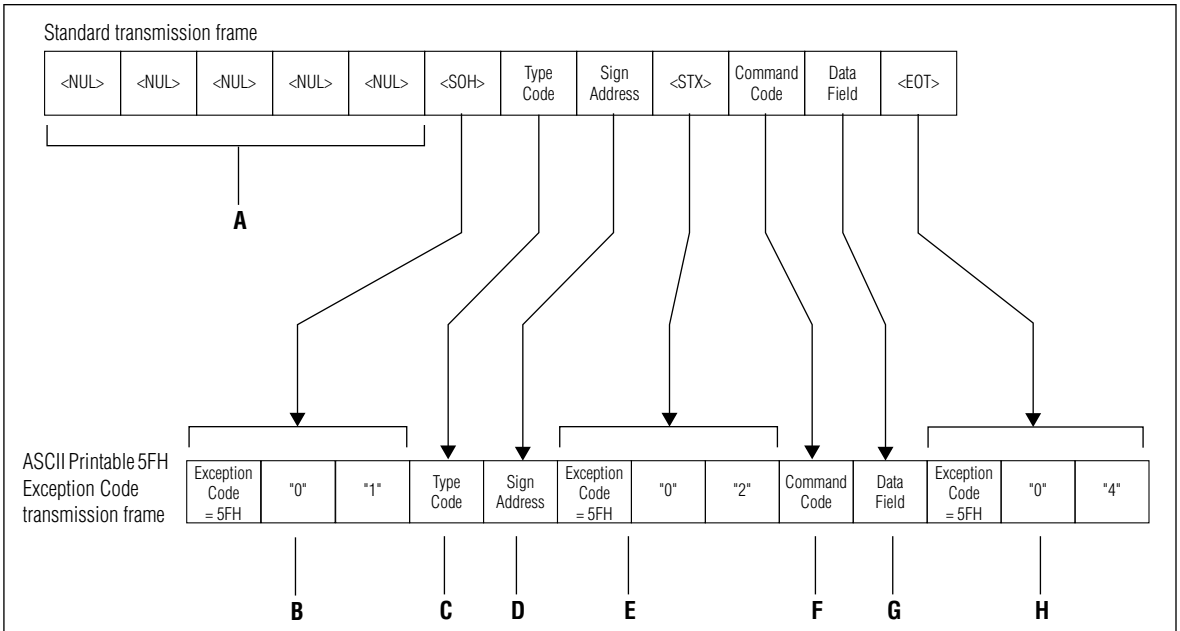
Table 8: Standard transmission frame compared with ASCII Printable 5DH Exception Code transmission frame



Item	Name	Description
A	<NUL>	The five <NUL>s (00H) used as frame synchronization are not used in the ASCII Printable transmission format. Since these <NUL>s were used to determine the baud rate, the ASCII Printable baud rate must be preset to 9600.
B	Exception Code	Exception Code = 5DH = "]"
	<SOH> + 20H	<SOH> + 20H = 21H = "!". The <SOH> (01H) ASCII control code is converted to a printable ASCII character by adding the 20H offset.
C	Type Code	See Table 4, "Standard transmission frame format," on page 5.
D	Sign Address	
E	Exception Code	Exception Code = 5DH = "]"
	<STX> + 20H	<STX> + 20H = 22H = """. The <STX> (02H) ASCII control code is converted to a printable ASCII character by adding the 20H offset.
F	Command Code	See Table 4, "Standard transmission frame format," on page 5.
G	Data Field	
H	Exception Code	Exception Code = 5DH = "]"
	<EOT> + 20H	<EOT> + 20H = 24H = "$". The <EOT> (04H) ASCII control code is converted to a printable ASCII character by adding the 20H offset.

3.1.3.2 ASCII Printable 5FH Exception Code transmission frame format ("3-byte" code)

Table 9: Standard transmission frame compared with ASCII Printable 5FH Exception Code transmission frame



Item	Name		Description
A	<NUL>		The five <NUL>s (00H) used as frame synchronization are not used in the ASCII Printable transmission format. Since these <NUL>s were used to determine the baud rate, the ASCII Printable baud rate must be preset to 9600.
B	Exception Code		Exception Code = 5FH = "_"
	<SOH> =	"0" + "1"	The <SOH> (01H) is broken down into two ASCII bytes "0" and "1".
C	Type Code		See Table 4, "Standard transmission frame format," on page 5.
D	Sign Address		
E	Exception Code		Exception Code = 5FH = "_"
	<STX> =	"0" + "2"	The <STX> (02H) is broken down into two ASCII bytes "0" and "2".
F	Command Code		See Table 4, "Standard transmission frame format," on page 5.
G	Data Field		
H	Exception Code		Exception Code = 5FH = "_"
	<EOT> =	"0" + "4"	The <EOT> (04H) is broken down into two ASCII bytes "0" and "4".

3.2 Command Codes

A Command Code is a single-byte field in a EZ KEY II / EZ95 protocol transmission frame and determines whether information is read or written to signs.

Command Codes not only determine the contents of the Command Code field, but also the Data Field in the EZ KEY II / EZ95 protocol transmission frame formats (see “Transmission frame formats” on page 4).

Command Codes fall into six, general categories:

- TEXT file commands
- SPECIAL FUNCTIONs
- STRING file commands
- DOTS PICTURE file commands
- ALPHAVISION DOTS PICTURE file commands
- ALPHAVISION BULLETIN MESSAGE file commands

3.2.1 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 sign’s 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 pause 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.

3.2.1.1 Write TEXT file Command Code — "A" (41H)

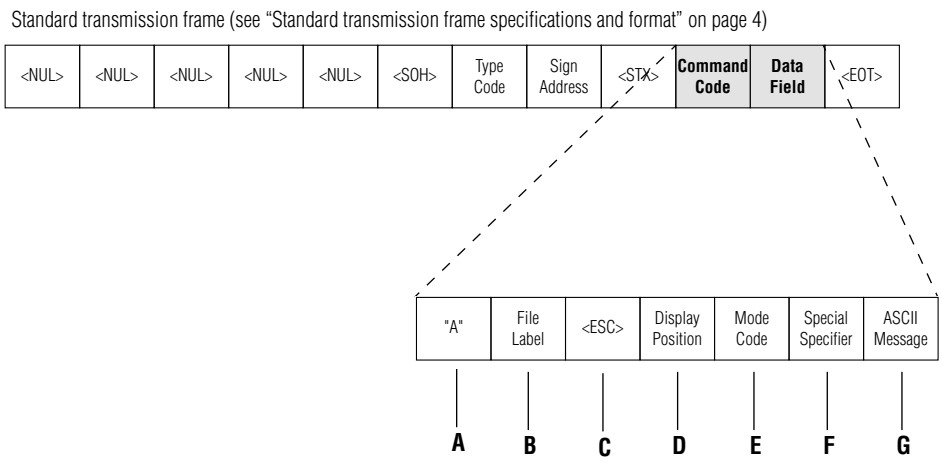
When writing to a TEXT file, the display will blank. After the transmission is over, the unit will begin displaying the last received TEXT file.

SHOW ME

An example of the Write TEXT Command Code is on page 52.

This example also shows how to use multiple Type Codes and Sign Addresses.

Table 10: Write TEXT file transmission frame format



Item	Name			Description
A	Command Code			"A" (41H) = Write TEXT file
B	Data Field	File Label		One ASCII character that indicates the TEXT file being accessed. See "Appendix A: File Label format" on page 37. If the File Label = "0" (30H), then a Priority TEXT file will be written (see "Priority TEXT files" on page 14).
C		[1]TEXT file data format	<ESC>	<ESC> (1BH) always starts the Mode Field.
D			Display Position	A single ASCII character that defines the line position on a multi-line sign: " " 20H Middle Line — text centered vertically. "" 22H Top Line — Text begins on the top line of the sign and the sign will use all its lines minus 1 in order to display the text. For example, a 6-line sign will allow a maximum of 5 lines (6 minus 1) 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 Line. For example, a 6-line sign with 3 lines of text associated with the Top Line would start the Bottom Line text on the 4th line of the sign. "0" 30H Fill — The sign will fill all available lines, centering the lines vertically. **NOTE:** On one-line signs, the Display Position is irrelevant, but it still must be included.
E			Mode Code	A single ASCII character that represents a "mode" which is a way of displaying an ASCII message. See "Standard Modes" on page 66.
F			Special Specifier	(Only required when preceding Mode Code is "n" for SPECIAL.) See "Special Modes" on page 66 and "Special Graphics" on page 67.
G			ASCII Message	The actual text to be displayed on a sign. This can also include special Control Characters (see "Appendix I: ALPHA protocol ASCII table" on page 68). **NOTE:** An ASCII Message cannot be displayed if the previous field (Special Specifier) is a Special Graphic. To display text after a Special Graphic, another Mode Field must be used.

NOTE: [1]This can be repeated until the sign's internal memory limit is reached. This limit is dependent on the individual sign.

3.2.1.2 Read TEXT file Command Code — "B" (42H)

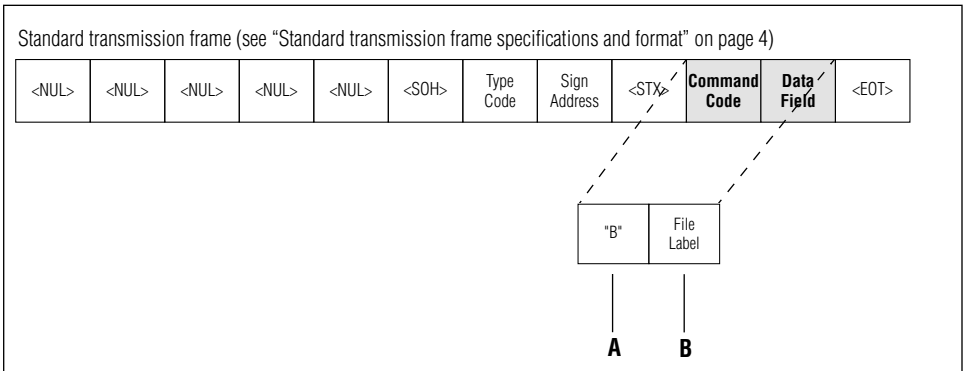
This command asks a sign to send back a TEXT file.

NOTE: Whenever doing a "Read" command on a network with multiple signs, it's important that each sign has a unique Serial Address. Also, only one sign at a time should be written to or read from.

SHOW ME

An example of the Read TEXT file frame is on page 53.

Table 11: Read TEXT file transmission frame format



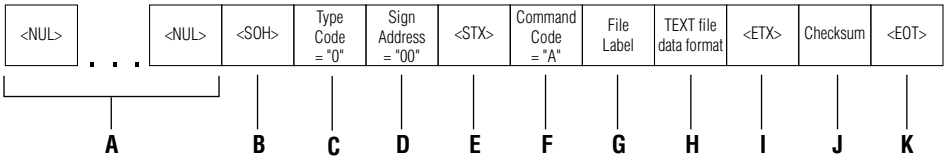
Item	Name		Description
A	Command Code		"B" (42H) = Read TEXT file
B	Data Field	File Label	One ASCII character that indicates the TEXT file being accessed. See "Appendix A: File Label format" on page 37. If the File Label = "0" (30H), then the Priority TEXT file will be read (see "Priority TEXT files" on page 14).

Following the Read TEXT file Command Code, a sign will respond with the following:

SHOW ME

An example of the Read TEXT file sign response frame is on page 53.

Table 12: Read TEXT file sign response frame format



Item	Name	Description
A	<NUL>	Twenty <NUL>s (00H) characters
B	<SOH>	<SOH> (01H) character
C	Type Code	"0" (30H) is the Response code
D	Sign Address	"00" (30H + 30H) is sent regardless of the sign's actual address.
E	<STX>	<STX> (02H) character
F	Command Code	"A" is returned by the sign. (This is the Write TEXT Command Code.)
G	File Label	One ASCII character that indicates the TEXT file being accessed. See "Appendix A: File Label format" on page 37.
H	TEXT file data format	See Table 10, "Write TEXT file transmission frame format," on page 12.
I	<ETX>	<ETX> (03H) character
J	Checksum	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
K	<EOT>	<EOT> (04H) character

3.2.1.3 Priority TEXT files

A Priority TEXT file is a special 125-byte message that does not need to be *configured* because it always exists on a sign. When data is written to a Priority TEXT file, all other TEXT files that are currently running will stop being displayed. A Priority TEXT file is created when a File Label = "0" (30H).

SHOW ME

Examples of Priority TEXT file frames are on page 57.

The Priority TEXT file will run all by itself until:

- a Write Priority TEXT file without any ASCII Message is sent
- a serial write to the Run Time table takes place
- a serial write to the Run Day table takes place
- an IR keyboard is pointed at the sign and the **PROG** key is pressed

Once a Priority TEXT file stops running, the sign will begin running the other TEXT files.

3.2.2 SPECIAL FUNCTIONS files

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

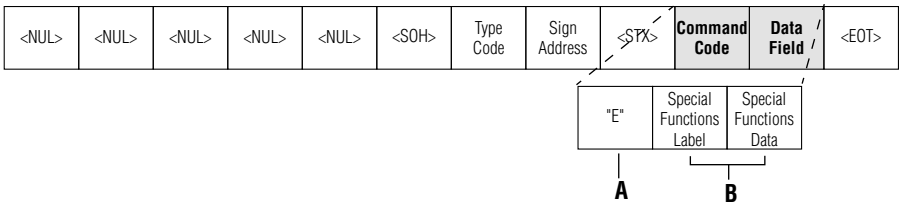
3.2.2.1 Write SPECIAL FUNCTIONS Command Code — "E" (45H)

SHOW ME

An example of the Write SPECIAL FUNCTIONS frame is on page 58.

Table 13: Write SPECIAL FUNCTIONS file transmission frame format

Standard transmission frame (see "Standard transmission frame specifications and format" on page 4)



Item	Name	Description
A	Command Code	"E" (45H) = Write SPECIAL FUNCTIONS file
B	Special Functions Label (one ASCII character)	Special Functions Data
	" " 20H	**Set Time of Day** — four ASCII digits used to set the time of day (24-hour format) clock in a sign. The following format is used: HhMm where: H = ASCII digit representing hours (10's digit) h = ASCII digit representing hours (1's digit) M = ASCII digit representing minutes (10's digit) m = ASCII digit representing minutes (1's digit) To display the time, see the "Control characters" in "Appendix I: ALPHA protocol ASCII table" on page 68.
	"!" 21H	**Enable/Disable a Sign's Speaker** — two ASCII characters: "00" 30H + 30H = enable speaker "FF" 46H + 46H = disable speaker **(default)**

	"$" 24H	**Set Memory Configuration** — eleven (or multiples thereof) ASCII characters used to set a sign's Memory Configuration table. Memory Configuration is a sign's internal battery-backed up RAM directory. A message file cannot be written until a Memory Configuration is written first — *unless the file is a Priority TEXT file or the default TEXT file "A"*. Also, whenever a Memory Configuration is written, the previous table is overwritten. Memory Configuration uses the following format:
		FTPSIZEQQQQ where:
		Repeat for *each* file to be configured.
		F = One ASCII character that represents the File Label. For valid File Labels, see "Appendix A: File Label format" on page 37.
		T = One ASCII character that represents the file type. Valid file types are:
		"A" 41H = TEXT file
		"B" 42H = STRING file
		"D" 43H = DOTS PICTURE file
		P = One ASCII character that presents the keyboard protection status, either
		"U" 55H = Unlocked. Means that the file can be accessed via an IR keyboard.
		"L" 4CH = Locked. Means that the file can not be accessed via an IR keyboard. (For a STRING file, "L" *must* be selected.)
		SIZE = Four ASCII characters that represent the hexadecimal file size in bytes of a TEXT or STRING file. For a DOTS PICTURE file, the first two bytes = # pixel rows and the last two bytes = the # of pixel columns in the picture.
		QQQQ = Four ASCII hexadecimal characters whose format depends on file type used:
		• For a TEXT file, the first two characters represent the file's Start Time and the last two characters represent the Stop Time. For valid entries, see "Appendix B: TEXT file Start and Stop times" on page 38.
		• For a STRING file, use "0000" as place holders because these four characters have no special meaning
		• For a DOTS PICTURE file, this represents the Color Status. Valid entries are:
		"1000" = monochrome, "2000" = 3-color, "4000" = 8-color

SHOW ME

Examples of Set Memory Configuration start on page 60.

Table 13: Write SPECIAL FUNCTIONS file transmission frame format

B (cont)	"&"	26H	**Set Day of Week** — one ASCII digit that represents the day of the week. A sign will automatically update the day of the week at 12:00 am every day. Valid entries are "1" 31H = Sunday "2" 32H = Monday "3" 33H = Tuesday "4" 34H = Wednesday "5" 35H = Thursday "6" 36H = Friday "7" 37H = Saturday
	"'"	27H	**Set Time Format** — one ASCII character that represents how time is shown on a sign. Valid entries are "S" 53H = Standard am/pm format **(default)** "M" 4DH = 24-hour (military) time
	"("	28H	**Generate Speaker Tone** — [2]one to five ASCII characters which generate a tone from a sign's speaker. Valid entries are [3]"A" 41H = Turn sign speaker on. [3]"B" 42H = Turn sign speaker off. [4]"0" 30H = Generate a continuous tone for about 2 seconds [4]"1" 31H = Generate three, short beeps (total time about 2 seconds) [5]"2" 32H = Generate a programmable tone according to this format: FFDR where FF = Two ASCII hexadecimal characters that represent a speaker frequency. Valid entries are from "00" through "FE". D = One ASCII hexadecimal character that represents the duration of a tone in 0.1 second increments. Valid entries are from "1" through "F". R = One ASCII hexadecimal character that represents the number of times a tone is repeated. Valid entries are from "0" through "F".
	")"	29H	**Set Run Time Table** — [6]five ASCII characters used to set the start and stop times in the Run Time table in the following format: FQQQQ where F = One ASCII character that represents a TEXT File Label. QQQQ = Four ASCII hexadecimal characters. The first two characters represent a file's Start Time and the last two characters represent a file's Stop Time. For valid entries, see "Appendix B: TEXT file Start and Stop times" on page 38. These values overwrite the values currently stored in the Memory Configuration table.
	","	2CH	**Soft Reset** — causes a soft reset of the sign. There is no data in this field. A soft reset causes the sign to go through its power-up diagnostics. Memory will not be cleared (non-destructive).
	"."	2EH	**Set Run Sequence** — from 3 to 130 ASCII characters that specify the Run Sequence. From 1 to 128 TEXT files can be set using the following format: KPF where: *F* repeats for *each* file to be configured. K = One ASCII character that represents the type of Run Sequence order: "T" 54H = All subsequent TEXT File Labels in the Run Sequence will run according to their associated *times* **(default)**. "S" 53H = All subsequent TEXT File Labels in the Run Sequence will run *in order* regardless of each file's run time. "D" 44H = All subsequent TEXT file labels in the Run Sequence will run according to their associated times. Then when the file reaches an "off time", the file will be deleted. P = One ASCII character that represents the keyboard protection status: "U" 55H = Unlocked. This allows the Run Sequence to be changed from a hand-held IR keyboard **(default)**. "L" 4CH = Locked. This makes the Run Sequence inaccessible from a hand-held IR keyboard. F = One ASCII character that represents a valid TEXT File Label (See "Appendix A: File Label format" on page 37). If a File Label is invalid or does not exist, the next File Label will be processed. Up to 128 File Labels can be in a Run Sequence.

Table 13: Write SPECIAL FUNCTIONS file transmission frame format

B (cont)	"/"	2FH	**Set Dimming Times** — four ASCII characters that are used to control sign dimming in the following format: WWww where WW = Two ASCII hexadecimal characters that represent the Start Time when a sign should dim. ww = Two ASCII hexadecimal characters that represent the Stop Time when a sign should stop dimming. **NOTE:** For a valid Start Time and Stop Time, see "Appendix B: TEXT file Start and Stop times" on page 38. **NOTE:** If dimming is not desired, set WWww = "0000" **(default)**. **NOTE:** Dimming is only available on the Solar, 790i, 460i, 440i, and 430i signs.
	"2"	32H	**Set Run Day Table** — three ASCII characters that are used for each TEXT File Label to set the start and stop days in the Run Day Table in the following format: FSs where F = One ASCII character that represents the TEXT File Label. For valid File Labels, see "Appendix A: File Label format" on page 37. S = One ASCII hexadecimal character that represents run start day for the TEXT file specified by F. Valid start day characters are: "0" 30H = Daily "1" 31H = Sunday "2" 32H = Monday "3" 33H = Tuesday "4" 34H = Wednesday "5" 35H = Thursday "6" 36H = Friday "7" 37H = Saturday "8" 38H = Monday-Friday "9" 39H = Weekends "A" 41H= Always "B" 42H = Never s = One hexadecimal character that represents the run stop day for the TEXT file specified by F. Valid stop day characters are: "1" 31H = Sunday "2" 32H = Monday "3" 33H = Tuesday "4" 34H = Wednesday "5" 35H = Thursday "6" 36H = Friday "7" 37H = Saturday **NOTE:** The stop day is required even though the start day may cover multiple days (e.g., Daily, Never, etc.) In this case, the stop day is ignored.
	"4"	34H	**Clear Serial Error Status Register** — one ASCII character that is used to clear the Serial Error Status Register to its default value of 40H. This register is set to its default value (40H or 01000000B) for the following Command Codes: (1) Read Serial Error Status Register, (2) Network Query, or (3) Clear Serial Error Status Register. **NOTE:** This command should be used 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 the nested frame. Also, the *last command in a nested transmission frame* should be a Serial Error Status read (see the "*" command in Table 14, "Read SPECIAL FUNCTIONS file transmission frame format," on page 21).
	";"	3BH	**Set Date** — six ASCII characters that are used to set the date in the following format: mmddyy where mm = Two ASCII digits that represent the month dd = Two ASCII digits that represent the day yy = Two ASCII digits that represent the year (see "Y2K NOTE")

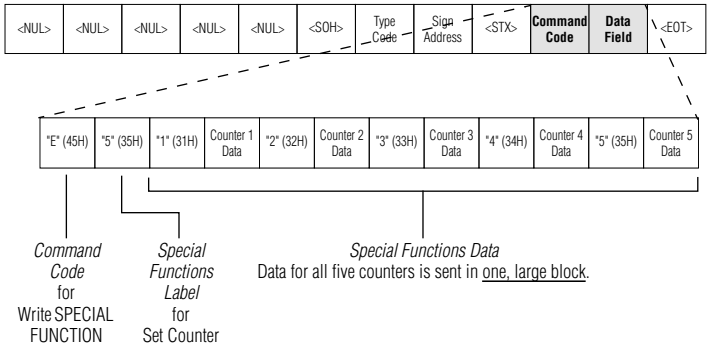
Y2K NOTE

Adaptive uses year "windowing" for Y2K date correction.

For ALPHA firmware version 2.0 and greater, the year (yy) is windowed as follows:
"00" to "96" = 2000 to 2096
"97" to "99" = 1997 to 1999

Table 13: Write SPECIAL FUNCTIONS file transmission frame format

<table>
<tr>
<td>B
(cont)</td>
<td>“5”</td>
<td>35H</td>
<td>
Set Counter — used to set one or more of the five internal timers available on counter-equipped signs. Data for all five counters must be sent as one, large block, in the following format:

NOTE: Even if you are only setting one counter, data must be sent to the other counters as well.
Standard transmission frame (see “Standard transmission frame specifications and format” on page 4)


The format of Counter 1 Data, Counter 2 Data, etc from above is as follows:
BBTTttSSSSSSSSiiiiiiiiVVVVVVVVttttttttFFmmHH
where
BB = Two ASCII hexadecimal characters that set the 8 bits of the Counter Control Byte, whose default value is 01100100B (64H). The first ASCII character sets bits 4 - 7 and the second ASCII character sets bits 0 - 3 of the Counter Control Byte. For example, to set the Counter Control Byte to its default value of 64H, an ASCII “6” (36H) and an ASCII “4” (34H) would be sent. Here’s what the 8 bits of the Counter Control Byte mean:
bit 7 — 1 = counter on, 0 = counter off (default = 0)
bit 6 — 1 = increment, 0 = decrement (default = 1)
bit 5 — 1 = count minutes, 0 = don’t count minutes (default = 1)
bit 4 — 1 = count hours, 0 = don’t count hours (default = 0)
bit 3 — 1 = count days, 0 = don’t count days (default = 0)
bit 2 — 1 = weekends on, 0 = weekends off (default = 1)
[7]bit 1 — 1 = Auto Reload ON, Auto Reload OFF (default = 0)
bit 0 — 0 (default = 0)
[8]TT = Two ASCII hexadecimal characters representing the Counter Start Time. See “Appendix B: TEXT file Start and Stop times” on page 38. (default = “FF” for Always)
[9]tt = Two ASCII hexadecimal characters representing the Counter Stop Time. See “Appendix B: TEXT file Start and Stop times” on page 38. The Counter Stop Time is ignored when the Counter Start Time = “FF” for Always. (default = “00”)
[10]SSSSSSSS = Eight ASCII characters that represent an 8-digit BCD Counter Start Value. Valid values are from “00000000” to “99999999”. (default = “00000000”)
[10]iiiiiiii = Eight ASCII characters that represent an 8-digit BCD Counter Change Value. This is the number that is either incremented or decremented according to bit 6 of the Counter Control Byte. Valid values are from “00000000” to “99999999”. (default = “00000001”)
[10]VVVVVVVV = Eight ASCII characters that represent an 8-digit BCD Current Counter Value. Valid values are from “00000000” to “99999999”. (default = “00000000”)
[10]tttttttt = Eight ASCII characters that represent an 8-digit BCD Counter Target Value. When this value equals the Current Counter Value, from 0 to 5 Target file messages will be sent according to parameter FF (below). Valid values are from “00000000” to “99999999”. (default = “00000000”)
FF = Two ASCII hexadecimal characters that represent the Target File Byte whose default value is 00000000 (00H). The first ASCII character sets bits 4 - 7 and the second ASCII character sets bits 0 - 3 of the Target File Byte. For example, to set a value of 1FH, an ASCII “1” (31H) and an ASCII “F” (46H) would be sent. Here’s what the 8 bits of the Target File Byte mean:
bit 7 — 0 (default = 0)
bit 6 — 0 (default = 0)
bit 5 — 0 (default = 0)
bit 4 — Target File 1: 1 = enabled, 0 = disabled (default = 0)
bit 3 — Target File 2: 1 = enabled, 0 = disabled (default = 0)
bit 2 — Target File 3: 1 = enabled, 0 = disabled (default = 0)
bit 1 — Target File 4: 1 = enabled, 0 = disabled (default = 0)
bit 0 — Target File 5: 1 = enabled, 0 = disabled (default = 0)
</td>
</tr>
</table>

Table 13: Write SPECIAL FUNCTIONS file transmission frame format

B (cont)	"5"	35H	**Set Counter** (cont) [11]mm = Two ASCII hexadecimal characters that set the Counter Change Minutes Synchronization. Valid values are from "00" to "3B" (00 - 59). **(default = "00")** [12]HH = Two ASCII hexadecimal characters that set the Counter Change Hours Synchronization. Valid values are from "00" to "17" (00 - 23) where "00" = 12 am, "01" = 1 am, and so on. **(default = "00")**
	"7"	37H	**Set Serial Address** — Two ASCII hexadecimal characters used to set a sign's serial address. Valid values are from "00" through "FF". **(default = "00")** **NOTE:** If the serial address has been set using a hardware DIP switch to an address other than "00", the DIP switch address will override the address set here — once power to the sign has been cycled.
	"8"	38H	[13]**Set ALPHAVISION DOTS PICTURE Memory Configuration** — a data stream of 24 ASCII characters that repeats for each file configured in a sign. The format for this data stream is as follows: FFFFFFFFFPRRRRCCCCccrrrr where [14]FFFFFFFFF = A 9-character file name P = One ASCII character that represents the keyboard protection status. Valid values are: "U" 55H = Unlocked. This allows the DOTS PICTURE file to be changed from a hand-held IR keyboard **(default)**. "L" 4CH = Locked. This makes the DOTS PICTURE file inaccessible from a hand-held IR keyboard. RRRR = Four ASCII hexadecimal digits that represent the number of pixel rows. Leading zeroes are required (e.g., "0040" = 64 rows). CCCC = Four ASCII hexadecimal digits that represent the number of pixel rows. Leading zeroes are required (e.g., "0060" = 96 columns). cc = Two ASCII hexadecimal digits representing the number of colors in the FAR DOTS PICTURE. Valid values are: "01" = a monochrome DOTS PICTURE "02" = a tricolor DOTS PICTURE rrrr = reserved for future use. Four ASCII zeroes are required — "0000".
	"9"	39H	**Append to ALPHAVISION DOTS PICTURE file Memory Configuration** — 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	**Set Temperature Offset** — allows for improvement in temperature accuracy as displayed on message centers which support temperature display (790i, 460i, 440i, and 430i). The data format is as follows: SO where: S = One ASCII character that stands for the sign of the temperature offset. Valid values are: "+" 2BH = a positive offset "-" 2DH = a negative offset O = One ASCII hexadecimal character that stands for the temperature offset. Valid values are from "0" through "9". *For a Solar sign*, an actual temperature is sent, not an offset. The Solar sign itself computes the offset. The data format for a Solar sign is as follows: SO where: S = One ASCII character that stands for the sign of the temperature. Valid values are: "+" 2BH = a positive temperature "-" 2DH = a negative temperature O = Three ASCII hexadecimal characters that stand for an actual temperature.

Table 13: Write SPECIAL FUNCTIONS file transmission frame format

NOTE: [1]The sum of all the file sizes (except for DOTS PICTURE and ALPHAVISION DOTS PICTURE files) plus 11 bytes of overhead for each file should not exceed the total amount of available memory in the pool. A value of "0000" is a valid SIZE for the last file in the Memory Configuration only if this last file is a TEXT file. This assigns all remaining memory to the file.
[2]When sending nested frames, the tone generation command must be the last transmission frame because the sign's serial port is disabled (and cannot receive any data) while a tone is generated. A tone generation command can never be part of any type of READ command.
[3]This command should not be used with the standard speaker/piezo alarm provided in the sign as it may damage the sign.
[4]Wait a minimum of 3 seconds before transmitting more data to the sign.
[5]Wait until the programmable tone has finished before transmitting more data to the sign.
[6]This 5-byte field repeats for each TEXT configured in the sign. Not all TEXT files need to be updated, only those that require modification.
[7]When the Counter Target Value has been reached, Auto Reload ON will put into the Counter Start Value in Current Counter Value.
[8]Time codes "FD" and "FE" are not valid as Counter Start Times.
[9]Time codes "FD", "FE", and "FF" are not valid as Counter Stop Times.
[10]Leading 0's must be sent if the value is less than 8 digits long. For example, "256" would be sent as "00000256".
[11]This value is used when the Counter Control Byte is set to count hours or days. If minutes are being counted, this value is ignored. However, a value must still be supplied.
[12]This value is used when the Counter Control Byte is set to count days. If minutes or hours are being counted, this value is ignored. However, a value must still be supplied.
[13]Set ALPHAVISION DOTS PICTURE Memory Configuration *only* applies to Full Matrix ALPHAVISION signs and Series 7000 signs.
[14]If a file name is less than 9 characters, it must be padded with leading spaces (20H) so that the total number of characters is always nine.

3.2.2.2 Read SPECIAL FUNCTIONS Command Code — "F" (46H)

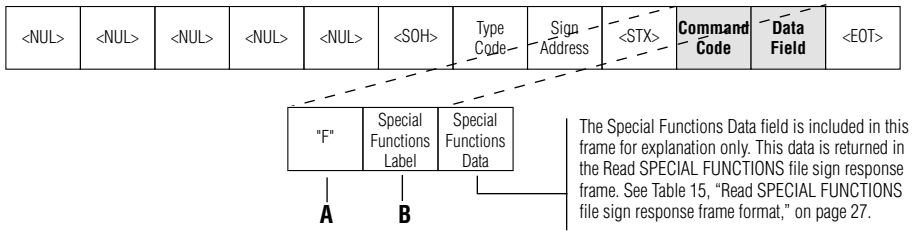
NOTE: Whenever doing a "Read" command on a network with multiple signs, it's important that each sign has a unique Serial Address. Also, only one sign at a time should be accessed or read from.

SHOW ME

An example of the Read SPECIAL FUNCTIONS frame is on page 59.

Table 14: Read SPECIAL FUNCTIONS file transmission frame format

Standard transmission frame (see "Standard transmission frame specifications and format" on page 4)



Item	Name	Description
A	Command Code	"F" (45H) = Read SPECIAL FUNCTIONS file
B	Special Functions Label (one ASCII character)	Special Functions Data (This data is returned in a Read SPECIAL FUNCTIONS file sign response. See Table 15, "Read SPECIAL FUNCTIONS file sign response frame format," on page 27)
	" " 20H	**Read Time of Day** — returns four ASCII digits that represent the time of day (24-hour format) clock in a sign. The following format is used: HhMm where: H = ASCII digit representing hours (10's digit) h = ASCII digit representing hours (1's digit) M = ASCII digit representing minutes (10's digit) m = ASCII digit representing minutes (1's digit) To display the time on a sign, see the "Control characters" in "Appendix I: ALPHA protocol ASCII table" on page 68.
	"!" 21H	**Read Speaker Status** — returns two ASCII characters: "00" 30H + 30H = speaker enabled "FF" 46H + 46H = speaker disabled **(default)**
	""" 22H	**Read General Information** — returns 28 or 29 ASCII characters in the following format: <NUL>FFFFFFFFfMmYyHhNnRSSPOOL,pool where [1]<NUL> = 00H FFFFFFFF = Eight ASCII characters that stand for the firmware installed in the sign f = One ASCII character that stands for the firmware revision letter MmYy = Four ASCII digits that stand for the release date of the firmware. For example, firmware released in January 1993 would be represented as "0193". HhNn = Four ASCII digits that represent the time of day (24-hour format) clock in a sign. The format is the same used for **Read Time of Day** above. R = One ASCII character that represents how time is displayed on a sign where: "S" 53H = standard am/pm format **(default)** "M" 4DH = 24-hour (or military) time SS = Speaker status where: "00" 30H + 30H = speaker enabled "FF" 45H + 45H = speaker disabled **(default)** POOL,pool = Memory Pool where: POOL = Four-digit ASCII hexadecimal number that represents the *total size* of the Memory Pool in bytes. The most significant digit is first. , = 2CH (a comma) pool = Four-digit ASCII hexadecimal number that represents the *unused* portion of the Memory Pool in bytes. The most significant digit is first. **NOTE:** General Information is most useful as a source of troubleshooting information.

Table 14: Read SPECIAL FUNCTIONS file transmission frame format

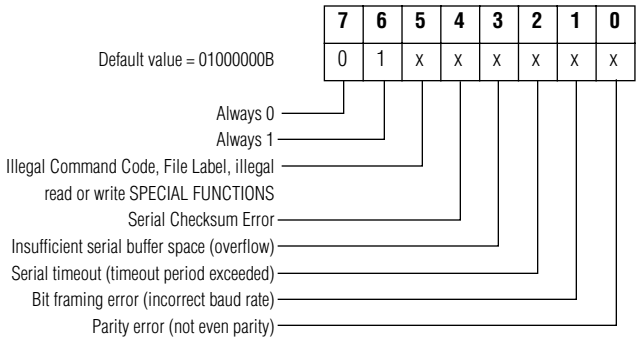
B (cont)	"#" 23H	**Read Memory Pool Size** — returns nine ASCII characters that indicate the total size and available amount of the Memory Pool. The Memory Pool is a sign's internal battery-backed up RAM that is available for file storage. Any unused memory is assigned to the first TEXT file listed in the Memory Configuration once the sign starts running. The Memory Pool is in the following format: POOL,pool. The format is the same used in **Read General Information** above.
	"$" 24H	**Read Memory Configuration** — returns eleven ASCII characters that represent a sign's Memory Configuration table. Memory Configuration is a sign's internal battery-backed up RAM directory. Memory Configuration uses the following format: FTPSIZEQQQQ where: F = One ASCII character that represents the File Label. For valid File Labels, see "Appendix A: File Label format" on page 37. T = One ASCII character that represents the file type. Valid file types are: "A" 41H = TEXT file "B" 42H = STRING file "D" 43H = DOTS PICTURE file P = One ASCII character that presents the keyboard protection status, either "U" 55H = Unlocked. Means that the file can be accessed via an IR keyboard. "L" 4CH = Locked. Means that the file can not be accessed via an IR keyboard. [2]SIZE = Four ASCII characters that represent the hexadecimal file size in bytes of a TEXT or STRING file. QQQQ = Four ASCII hexadecimal characters whose format depends on file type used: • For a TEXT file, the first two characters represent the file's Start Time and the last two characters represent the Stop Time. For valid entries, see "Appendix B: TEXT file Start and Stop times" on page 38. • For a STRING file, "0000" is used as place holders because these four characters have no special meaning. • For a DOTS PICTURE file, this represents the Color Status. Valid entires are "1000" = monochrome DOTS PICTURE "2000" = 3-color DOTS PICTURE "4000" = 8-color DOTS PICTURE
	"%" 25H	**Memory Dump** — returns multiple nested transmission frames with checksums (see "Nesting with Checksums" on page 7) in the following order: 1. Time-of-day setting (see **Read Time of Day** above) 2. Memory Configuration (see **Read Memory Configuration** above) 3. Transmission frame of each file (Write TEXT, STRING, or DOTS PICTURE file) in the order it appears in Memory Configuration 4. Run Sequence (see **Read Run Sequence** below) 5. Run Day Table (see **Read Run Day Table** below) 6. Day-of-Week setting (see **Read Day-of-Week** below) 7. Counter Functions (see **Read Counter Functions** below)
	"&" 26H	**Read Day of Week** — returns one ASCII digit that represents the day of the week. A sign will automatically update the day of the week at 12:00 am every day. Valid entries are "1" 31H = Sunday "2" 32H = Monday "3" 33H = Tuesday "4" 34H = Wednesday "5" 35H = Thursday "6" 36H = Friday "7" 37H = Saturday
	"'" 27H	**Read Time Format** — returns one ASCII character that represents how time is shown on a sign. Valid entries are "S" 53H = Standard am/pm format **(default)** "M" 4DH = 24-hour (military) time
	")" 29H	**Read Run Time Table** — returns [3]five ASCII characters that represent the start and stop times in the Run Time table in the following format: FQQQQ where F = One ASCII character that represents a TEXT File Label. QQQQ = Four ASCII hexadecimal characters. The first two characters represent a file's Start Time and the last two characters represent a file's Stop Time. For valid entries, see "Appendix B: TEXT file Start and Stop times" on page 38. These values overwrite the values currently stored in the Memory Configuration table.

Table 14: Read SPECIAL FUNCTIONS file transmission frame format

<table>
<tr>
<td rowspan="3">B
(cont)</td>
<td>“*” 2AH</td>
<td>Read Serial Error Status Register — returns one bitmapped ASCII character read from a sign’s Serial Error Status Register that represents serial errors recorded by a sign.

This register is set to its default value (40H or 01000000B) for the following Command Codes: (1) Read Serial Error Status Register, (2) Network Query, or (3) Clear Serial Error Status Register.

The sign begins error checking following a valid <SOH> (01H).

The Serial Error Status Register is bitmapped as follows:
Serial Error Status Register
 

NOTE: Errors are OR’d into the Serial Error Status Register. That is, more than one error at a time can be recorded in the register.</td>
</tr>
<tr>
<td>“-” 2DH</td>
<td>Network Query — returns the unit type, Serial Address, and Serial Error Status Register for each sign on the network. The response from each sign is in the following format: UAAZ where:
U = One ASCII character that stands for the unit type of a sign. For valid entries, see “Type Code” in “Standard transmission frame format” on page 5.
AA = Two ASCII hexadecimal characters that represent a sign’s serial address
Z = One ASCII character that represents the Serial Error Status Register of a sign (above)

NOTE: Normally, a Network Query is broadcast to all signs using a “00” in the Sign Address field. When a Network Query is broadcast like this, all signs on the network respond in the following manner: Once the <EOT> is received by a sign, it will respond to the Network Query after a timed interval. This interval is a sum of 1 second plus the product of a sign’s address and 0.5 seconds. For example, a sign with an address of 0FH (15), would reply after 1 + (15 x 0.5) = 8.5 seconds.
NOTE: If there are two or more signs on a network with the same Serial Address, then a Network Query will produce unpredictable results. A response from one of these signs may be garbled because there is no collision detection.</td>
</tr>
<tr>
<td>“.” 2EH</td>
<td>Read Run Sequence — returns from 3 to 130 ASCII characters that specify the Run Sequence. From 1 to 128 TEXT files will be read in the following format: KPF where:
F repeats for each file to be configured.
K = One ASCII character that represents the type of Run Sequence order:
“T” 54H = All subsequent TEXT File Labels in the Run Sequence will run according to their associated times (default).
“S” 53H = All subsequent TEXT File Labels in the Run Sequence will run in order regardless of each file’s run time.
P = One ASCII character that represents the keyboard protection status:
“U” 55H = Unlocked. This allows the Run Sequence to be changed from a hand-held IR keyboard (default).
“L” 4CH = Locked. This makes the Run Sequence inaccessible from a hand-held IR keyboard.
F = One ASCII character that represents a valid TEXT File Label
(See “Appendix A: File Label format” on page 37). If a File Label is invalid or does not exist, the next File Label will be processed. Up to 128 File Labels can be in a Run Sequence.</td>
</tr>
</table>

Table 14: Read SPECIAL FUNCTIONS file transmission frame format

B (cont)	"2"	32H	**Read Run Day Table** — returns three ASCII characters that are used for each TEXT File Label to read the start and stop days in the Run Day Table in the following format: FSs where F = One ASCII character that represents the TEXT File Label. For valid File Labels, see "Appendix A: File Label format" on page 37. S = One ASCII hexadecimal character that represents run start day for the TEXT file specified by F. Valid start day characters are: "0" 30H = Daily "1" 31H = Sunday "2" 32H = Monday "3" 33H = Tuesday "4" 34H = Wednesday "5" 35H = Thursday "6" 36H = Friday "7" 37H = Saturday "8" 38H = Monday-Friday "9" 39H = Weekends "A" 41H= Always "B" 42H = Never s = One hexadecimal character that represents the run stop day for the TEXT file specified by F. Valid stop day characters are: "1" 31H = Sunday "2" 32H = Monday "3" 33H = Tuesday "4" 34H = Wednesday "5" 35H = Thursday "6" 36H = Friday "7" 37H = Saturday **NOTE:** The stop day is required even though the start day may cover multiple days (e.g., Daily, Never, etc.) In this case, the stop day is ignored.
	";"	3BH	**Read Date** — returns six ASCII characters that are used to set the date in the following format: mmddyy where mm = Two ASCII digits that represent the month dd = Two ASCII digits that represent the day yy = Two ASCII digits that represent the year (see "Y2K NOTE" on page 17)

Table 14: Read SPECIAL FUNCTIONS file transmission frame format

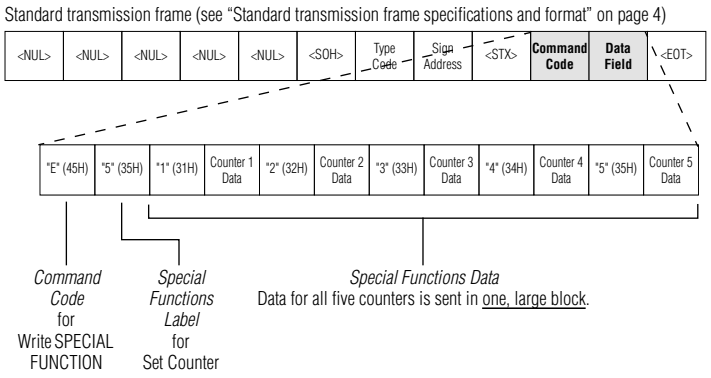
B (cont)	"5"	35H	**Read Counter** — returns data for all five counters is received as *one, large block*, in the following format:  The format of *Counter 1 Data*, *Counter 2 Data*, etc from above is as follows: BBTTttSSSSSSSSiiiiiiiiVVVVVVVVttttttttFFmmHH where BB = Two ASCII hexadecimal characters that stand for the 8 bits of the Counter Control Byte, whose default value is 01100100B (64H). The first ASCII character sets bits 4 - 7 and the second ASCII character sets bits 0 - 3 of the Counter Control Byte. Here's what the 8 bits of the Counter Control Byte mean: bit 7 — 1 = counter on, 0 = counter off **(default = 0)** bit 6 — 1 = increment, 0 = decrement **(default = 1)** bit 5 — 1 = count minutes, 0 = don't count minutes **(default = 1)** bit 4 — 1 = count hours, 0 = don't count hours **(default = 0)** bit 3 — 1 = count days, 0 = don't count days **(default = 0)** bit 2 — 1 = weekends on, 0 = weekends off **(default = 1)** [4]bit 1 — 1 = Auto Reload ON, Auto Reload OFF **(default = 0)** bit 0 — 0 **(default = 0)** [5]TT = Two ASCII hexadecimal characters representing the Counter Start Time. See "Appendix B: TEXT file Start and Stop times" on page 38. **(default = "FF" for Always)** [6]tt = Two ASCII hexadecimal characters representing the Counter Stop Time. See "Appendix B: TEXT file Start and Stop times" on page 38. The Counter Stop Time is ignored when the Counter Start Time = "FF" for Always. **(default = "00")** [7]SSSSSSSS = Eight ASCII characters that represent an 8-digit BCD Counter Start Value. Valid values are from "00000000" to "99999999". **(default = "00000000")** [7]iiiiiiii = Eight ASCII characters that represent an 8-digit BCD Counter Change Value. This is the number that is either incremented or decremented according to bit 6 of the Counter Control Byte. Valid values are from "00000000" to "99999999". **(default = "00000001")** [7]VVVVVVVV = Eight ASCII characters that represent an 8-digit BCD Current Counter Value. Valid values are from "00000000" to "99999999". **(default = "00000000")** [7]tttttttt = Eight ASCII characters that represent an 8-digit BCD Counter Target Value. When this value equals the Current Counter Value, from 0 to 5 Target file messages will be sent according to parameter FF (below). Valid values are from "00000000" to "99999999". **(default = "00000000")** FF = Two ASCII hexadecimal characters that represent the Target File Byte whose default value is 00000000 (00H). The first ASCII character sets bits 4 - 7 and the second ASCII character sets bits 0 - 3 of the Target File Byte. For example, to set a value of 1FH, an ASCII "1" (31H) and an ASCII "F" (46H) would be sent. Here's what the 8 bits of the Target File Byte mean: bit 7 — 0 **(default = 0)** bit 6 — 0 **(default = 0)** bit 5 — 0 **(default = 0)** bit 4 — Target File 1: 1 = enabled, 0 = disabled **(default = 0)** bit 3 — Target File 2: 1 = enabled, 0 = disabled **(default = 0)** bit 2 — Target File 3: 1 = enabled, 0 = disabled **(default = 0)** bit 1 — Target File 4: 1 = enabled, 0 = disabled **(default = 0)** bit 0 — Target File 5: 1 = enabled, 0 = disabled **(default = 0)**

Table 14: Read SPECIAL FUNCTIONS file transmission frame format

B (cont)	"5"	35H	**Read Counter** (cont) [8]mm = Two ASCII hexadecimal characters that set the Counter Change Minutes Synchronization. Valid values are from "00" to "3B" (00 - 59). **(default = "00")** [9]HH = Two ASCII hexadecimal characters that set the Counter Change Hours Synchronization. Valid values are from "00" to "17" (00 - 23) where "00" = 12 am, "01" = 1 am, and so on. **(default = "00")**
	"8"	38H	[10]**Read ALPHAVISION DOTS PICTURE Memory Configuration** — returns a data stream of 24 ASCII characters that repeats for each file configured in a sign. The format for this data stream is as follows: FFFFFFFFFPRRRRCCCCccrrrr where [11]FFFFFFFFF = A 9-character file name P = One ASCII character that represents the keyboard protection status. Valid values are: "U" 55H = Unlocked. This allows the DOTS PICTURE file to be changed from a hand-held IR keyboard **(default)**. "L" 4CH = Locked. This makes the DOTS PICTURE file inaccessible from a hand-held IR keyboard. RRRR = Four ASCII hexadecimal digits that represent the number of pixel rows. Leading zeroes are required (e.g., "0040" = 64 rows). CCCC = Four ASCII hexadecimal digits that represent the number of pixel rows. Leading zeroes are required (e.g., "0060" = 96 columns). cc = Two ASCII hexadecimal digits representing the number of colors in the Far Dots Picture. Valid values are: "01" = a monochrome DOTS PICTURE "02" = a tricolor DOTS PICTURE rrrr = reserved for future use. Four ASCII zeroes are required — "0000".
	"T"	54H	**Read Temperature Offset** — returns two ASCII characters in the following format: SO where: S = One ASCII character that stands for the sign of the temperature offset. Valid values are: "+" 2BH = a positive offset "-" 2DH = a negative offset O = One ASCII hexadecimal character that stands for the temperature offset. Valid values are from "0" through "9". *For a Solar sign*, an actual temperature is read, not an offset. The Solar sign itself computes the offset. The data format for a Solar sign is as follows: SO where: S = One ASCII character that stands for the sign of the temperature. Valid values are: "+" 2BH = a positive temperature "-" 2DH = a negative temperature O = Three ASCII hexadecimal characters that stand for an actual temperature.

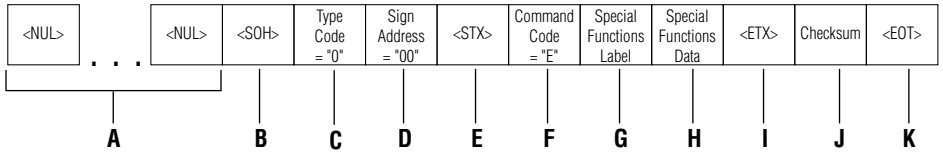
NOTE:
[1]This byte is transmitted only on some signs.
[2]The sum of all the file sizes (except for DOTS PICTURE and FAR DOTS PICTURE files) plus 11 bytes of overhead for each file should not exceed the total amount of available memory in the pool. A value of "0000" is a valid SIZE for the last file in the Memory Configuration only if this last file is a TEXT file. This assigns all remaining memory to the file.
[3]This 5-byte field repeats for each TEXT configured in the sign. Not all TEXT files need to be updated, only those that require modification.
[4]When the Counter Target Value has been reached, Auto Reload ON will put into the Counter Start Value in Current Counter Value.
[5]Time codes "FD" and "FE" are not valid as Counter Start Times.
[6]Time codes "FD", "FE", and "FF" are not valid as Counter Stop Times.
[7]Leading 0's must be sent if the value is less than 8 digits long. For example, "256" would be sent as "00000256".
[8]This value is used when the Counter Control Byte is set to count hours or days. If minutes are being counted, this value is ignored. However, a value must still be supplied.
[9]This value is used when the Counter Control Byte is set to count days. If minutes or hours are being counted, this value is ignored. However, a value must still be supplied.
[10]Read ALPHAVISION DOTS PICTURE Memory Configuration *only* applies to Full Matrix ALPHAVISION signs and Series 7000 signs.
[11]f a file name is less than 9 characters, it must be padded with leading spaces (20H) so that the total number of characters is always nine.

Following the Read SPECIAL FUNCTIONS file Command Code, a sign will respond with the following:

SHOW ME

An example of the Read SPECIAL FUNCTIONS file response frame is on page 59.

Table 15: Read SPECIAL FUNCTIONS file sign response frame format



Item	Name	Description
A	<NUL>	Twenty <NUL> (00H) characters
B	<SOH>	<SOH> (01H) character
C	Type Code	"0" (30H) is the Response code
D	Sign Address	"00" (30H + 30H) is sent regardless of the sign's actual address.
E	<STX>	<STX> (02H) character
F	Command Code	"E" is returned by the sign. (The Write SPECIAL FUNCTIONS Command Code.)
G	Special Functions Label	One ASCII character that indicates the SPECIAL FUNCTION being accessed. See Table 13, "Write SPECIAL FUNCTIONS file transmission frame format," on page 15 and Table 14, "Read SPECIAL FUNCTIONS file transmission frame format," on page 21.
H	Special Functions Data	See Table 13, "Write SPECIAL FUNCTIONS file transmission frame format," on page 15. and Table 14, "Read SPECIAL FUNCTIONS file transmission frame format," on page 21.
I	<ETX>	<ETX> (03H) character
J	Checksum	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
K	<EOT>	<EOT> (04H) character

3.2.3 STRING files

SPECIAL NOTE

For more information on using STRING files, see "Appendix D: STRING file notes" on page 40.

STRING files are used to store short ASCII sets of characters which may be "called up" from a TEXT file. The main purpose of a STRING file 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, the size of a STRING file is limited to 125 bytes.

Before writing to a STRING file, memory must be allocated for the STRING file in the sign. (For further information, see "Set Memory Configuration" in Table 13, "Write SPECIAL FUNCTIONS file transmission frame format," on page 15.)

STRING files are called from a TEXT file using the TEXT file Control character designated for a "Call STRING file". (For further information, see "Control characters" in "Appendix I: ALPHA protocol ASCII table" on page 68).

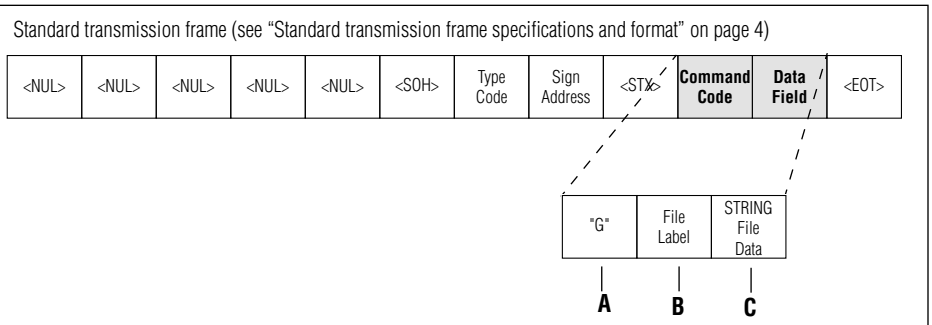
When reading from a STRING file, once the transmission frame has been sent, a sign will either pause or blank, depending on the sign type. Once a sign has transmitted the file, the sign will continue displaying the message from where it was interrupted.

3.2.3.1 Write STRING file Command Code — "G" (47H)

SHOW ME

An example of the Write STRING file frame is on page 63.

Table 16: Write STRING file transmission frame format



Item	Name		Description
A	Command Code		"G" (47H) = Write STRING file
B	Data Field	File Label	One ASCII character that indicates the STRING file being accessed. See "Appendix A: File Label format" on page 37.
C	Data Field	STRING File Data	This data can be ASCII characters 20H through 7FH and the following Control characters (for more information, see "Appendix I: ALPHA protocol ASCII table" on page 68) : 09H = No Hold speed 0DH = New line 11H = Disable wide characters **(default)** 12H = Enable wide characters 13H = Call Time (time of day will be called up) 15H = Speed 1 (slowest) 16H = Speed 2 17H = Speed 3 18H = Speed 4 **(default)** 19H = Speed 5 (fastest) 1AH = Select character set 1CH = Select character color (Rainbow 1 and 2 colors do not work in STRING files) 1EH = Select character spacing

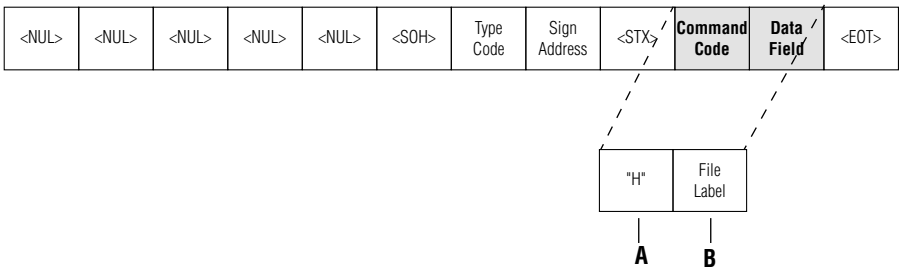
3.2.3.2 Read STRING file Command Code — "H" (48H)

NOTE: Whenever doing a "Read" command on a network with multiple signs, it's important that each sign has a unique Serial Address. Also, only one sign at a time should be written to or read from.

SHOW ME

An example of the Read STRING file frame is on page 64.

Table 17: Read STRING file transmission frame format



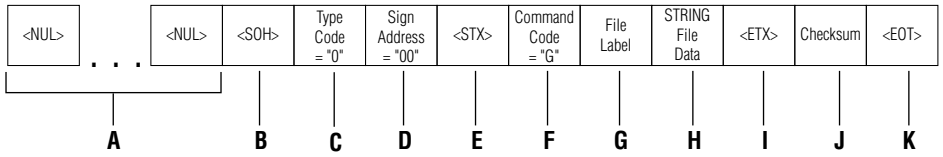
Item	Name		Description
A	Command Code		"H" (47H) = Read STRING file
B	Data Field	File Label	One ASCII character that indicates the STRING file being accessed. See "Appendix A: File Label format" on page 37.

Following the Read STRING file Command Code, a sign will respond with the following:

SHOW ME

An example of the Read STRING file sign response frame is on page 64.

Table 18: Read STRING file sign response frame format



Item	Name	Description
A	<NUL>	Twenty <NUL>s (00H) characters
B	<SOH>	<SOH> (01H) character
C	Type Code	"0" (30H) is the Response code
D	Sign Address	"00" (30H + 30H) is sent regardless of the sign's actual address.
E	<STX>	<STX> (02H) character
F	Command Code	"G" is returned by the sign. (The Write STRING file Command Code.)
G	File Label	One ASCII character that indicates the STRING file being accessed. See "Appendix A: File Label format" on page 37.
H	STRING File Data	See Table 16, "Write STRING file transmission frame format," on page 28.
I	<ETX>	<ETX> (03H) character
J	Checksum	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
K	<EOT>	<EOT> (04H) character

3.2.4 DOTS PICTURE files

DOTS PICTURE files are used to store dot patterns which may be "called" from a TEXT file. The main purpose of DOTS PICTURE files are to allow users to display custom graphics, such as logos.

SPECIAL NOTE

If a graphic's height and width are *greater* than 31 x 255 pixels, then an ALPHAVISION DOTS PICTURE file format must be used.

ALPHAVISION DOTS PICTURE files can only be used on Full Matrix ALPHAVISION signs and Series 7000 signs.

When a DOTS PICTURE exceeds a pixel height of 16 rows or a pixel width of 255 columns, the ALPHAVISION DOTS PICTURE file must be used.

ALPHAVISION signs support both DOTS PICTURE and ALPHAVISION DOTS PICTURE files.

DOTS PICTURE files are "called" from TEXT files using the TEXT file Control character for a "Call DOTS PICTURE" file. For further information, see "Appendix I: ALPHA protocol ASCII table" on page 68.

When a DOTS PICTURE file is sent to a sign, the sign will go blank until the transmission is complete.

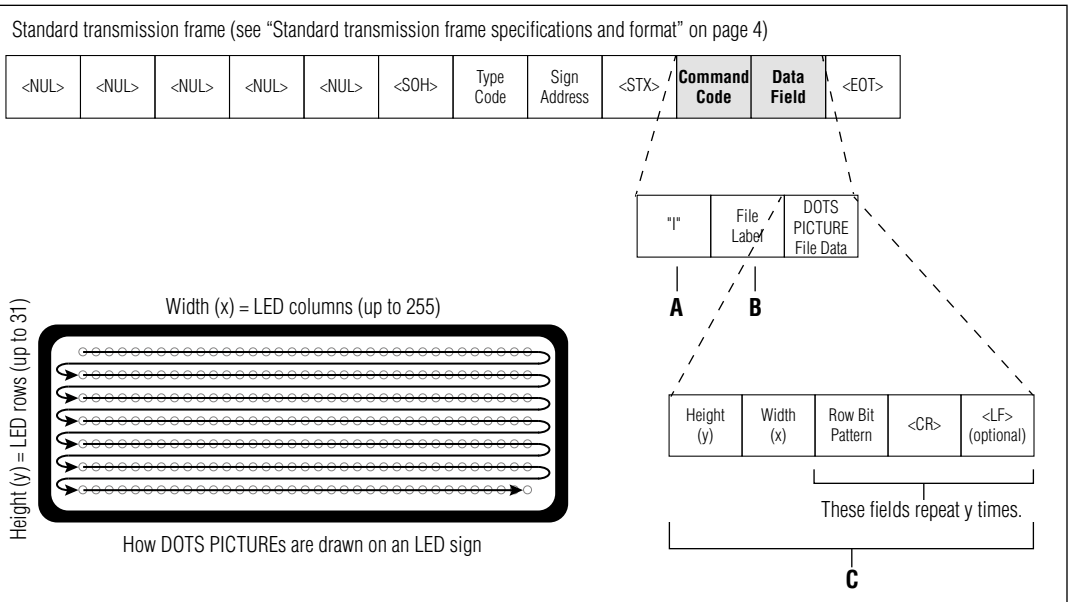
When reading from a DOTS PICTURE file, once the transmission frame has been sent, a sign will either pause or blank, depending on the type of sign. Once a sign has completely transmitted the file, the sign will continue displaying the message from where it was interrupted.

3.2.4.1 Write DOTS PICTURE file Command Code — "I" (49H)

SHOW ME

An example of the Write DOTS PICTURE file frame is on page 65.

Table 19: Write DOTS PICTURE file transmission frame format



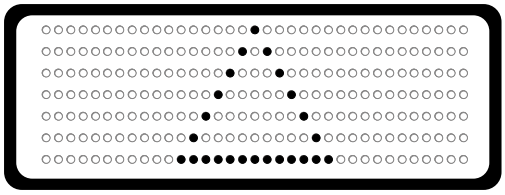
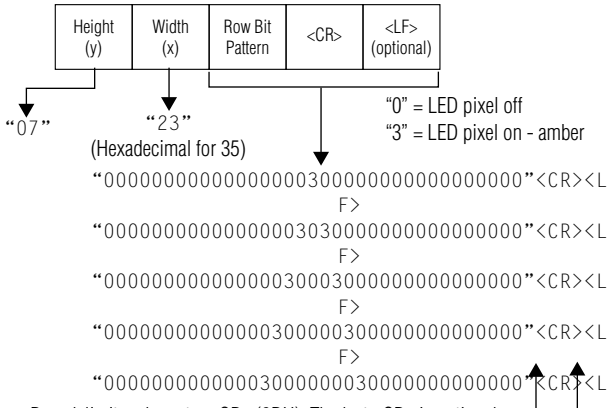
Item	Name			Description
A	Command Code			"I" (49H) = Write DOTS PICTURE file
B	File Label			One ASCII character that indicates the DOTS PICTURE file being accessed. See "Appendix A: File Label format" on page 37.
C	Data Field	DOTS PICTURE File Data	Height (y)	Two ASCII hexadecimal bytes that represent the number of pixel rows in a DOTS PICTURE (a value from 0 to 31). This must match the pixel row bytes set up in "Set ALPHAVISION DOTS PICTURE Memory Configuration" in Table 13, "Write SPECIAL FUNCTIONS file transmission frame format," on page 15. **NOTE:** In an ALPHAVISION DOTS PICTURE, four ASCII hexadecimal bytes are used to represent the number of pixel rows (a value from 0 to 65535).

Table 19: Write DOTS PICTURE file transmission frame format

<table>
<tr><td rowspan="2">C
(cont)</td><td rowspan="2">Data
Field
(cont)</td><td rowspan="2">DOTS
PICTURE
File Data
(cont)</td><td>Width
(x)</td><td>Two ASCII hexadecimal bytes that represent the number of pixel columns in a DOTS PICTURE (a value from 0 to 255). This must match the pixel column bytes set up in “Set ALPHAVISION DOTS PICTURE Memory Configuration” in Table 13, “Write SPECIAL FUNCTIONS file transmission frame format,” on page 15.

NOTE: In an ALPHAVISION DOTS PICTURE, four ASCII hexadecimal bytes are used to represent the number of pixel columns (a value from 0 to 65535).
NOTE: When sending a Write DOTS PICTURE file, the sign receiving the file will clear the current DOTS PICTURE file in memory immediately following the Width information.
NOTE: Following the Width bytes, there should be at least a 100 millisecond delay (not to exceed the timeout period) before sending the Row Bit Pattern.</td></tr>
<tr><td>[1]Row
Bit
Pattern</td><td>The Width (x) number of ASCII characters which represent all the pixels in a row. The first ASCII character = the leftmost pixel in the row, the 2nd ASCII character = the next pixel in the row, etc. (see example below). Valid values are:
“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 signs do not support the full range of colors.

To draw an amber DOTS PICTURE like this (on the 7 x 35 pixel sign shown below) . . .
 
. . . the DOTS PICTURE File Data would look like this:
 

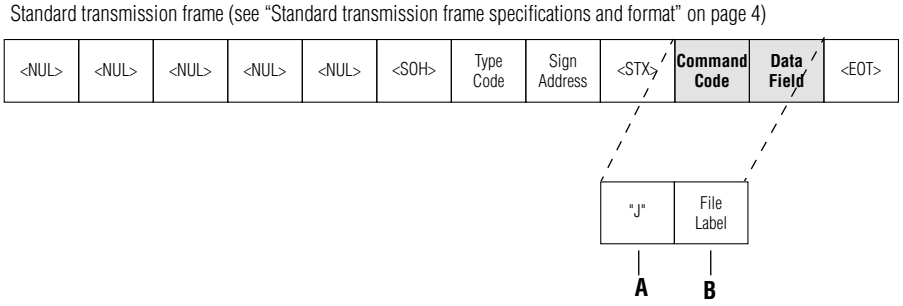
(See “Read DOTS PICTURE file Command Code — “J” (4AH)” on page 32.)

NOTE: If the number of row pixel characters is greater than the Width (x), then the extra row pixel characters will be ignored.
If the number of row pixel characters is less than the Width (x), then the remaining row pixel characters will be turned off (“0”).</td></tr>
<tr><td colspan="5">NOTE: [1]DATA COMPRESSION — Row Bit Pattern can be data compressed as follows for ALPHAVISION DOTS PICTURE files. Data compression can be done anywhere within the Row Bit Pattern. The format for data compression is: <CTR-Q>XXB where:
<CTR-Q> = 11H
XX = Two ASCII hexadecimal characters from “00” to “FF” that stand for the number of times + 1 to repeat B (the pixel color). For example, a value of “0A” (10) means repeat 10 + 1 = 11 times.
B = Pixel color. Valid values are shown in Row Bit Pattern field above.</td></tr>
</table>

3.2.4.2 Read DOTS PICTURE file Command Code — "J" (4AH)

NOTE: Whenever doing a "Read" command on a network with multiple signs, it's important that each sign has a unique Serial Address. Also, only one sign at a time should be written to or read from.

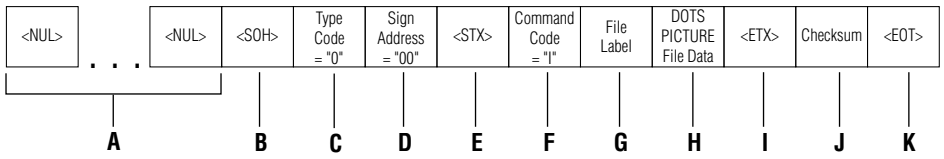
Table 20: Read DOTS PICTURE file transmission frame format



Item	Name		Description
A	Command Code		"J" (4AH) = Read DOTS PICTURE file
B	Data Field	File Label	One ASCII character that indicates the DOTS PICTURE file being accessed. See "Appendix A: File Label format" on page 37.

Following the Read DOTS PICTURE file Command Code, a sign will respond with the following:

Table 21: Read DOTS PICTURE file sign response frame format



Item	Name	Description
A	<NUL>	Twenty <NUL>s (00H) characters
B	<SOH>	<SOH> (01H) character
C	Type Code	"0" (30H) is the Response code
D	Sign Address	"00" (30H + 30H) is sent regardless of the sign's actual address.
E	<STX>	<STX> (02H) character
F	Command Code	"I" is returned by the sign. (The Write DOTS PICTURE file Command Code.)
G	File Label	One ASCII character that indicates the DOTS PICTURE file being accessed. See "Appendix A: File Label format" on page 37.
H	DOTS PICTURE File Data	See Table 19, "Write DOTS PICTURE file transmission frame format," on page 30.
I	<ETX>	<ETX> (03H) character
J	Checksum	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
K	<EOT>	<EOT> (04H) character

3.2.5 ALPHAVISION DOTS PICTURE files

SPECIAL NOTE

An ALPHAVISION DOTS PICTURE file could, in theory, be 65535 pixels high by 65535 pixels wide. On the other hand, a DOTS PICTURE cannot exceed a pixel height of 31 rows and a pixel width of 255 columns.

ALPHAVISION DOTS PICTURE files are used to store dot patterns which may be "called" from a TEXT file. The main purpose of ALPHAVISION DOTS PICTURE files are to allow users to display custom graphics, such as logos.

Full Matrix ALPHAVISION signs and Series 7000 signs support both DOTS PICTURE and ALPHAVISION DOTS PICTURE files.

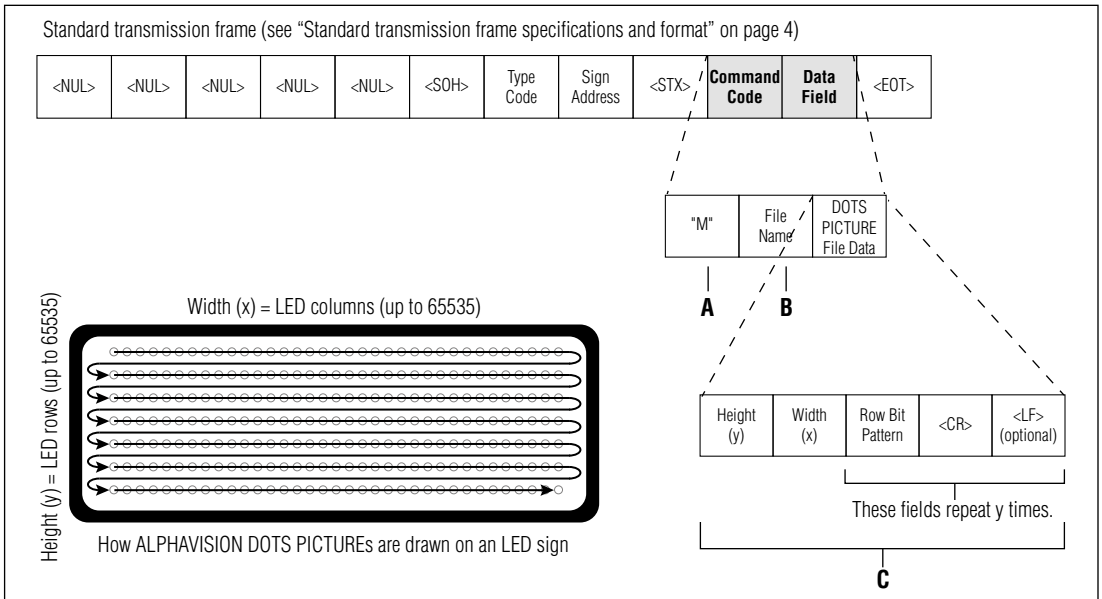
ALPHAVISION DOTS PICTURE files are "called" from TEXT files using the TEXT file Control character for a "Call ALPHAVISION DOTS PICTURE" file. For further information, see "Appendix I: ALPHA protocol ASCII table" on page 68.

When an ALPHAVISION DOTS PICTURE file is sent to a sign, the sign will go blank until the transmission is complete.

When reading from an ALPHAVISION DOTS PICTURE file, once the transmission frame has been sent, a sign will either pause or blank, depending on the type of sign. Once a sign has completely transmitted the file, the sign will continue displaying the message from where it was interrupted.

3.2.5.1 Write ALPHAVISION DOTS PICTURE file Command Code — "M" (4DH)

Table 22: Write ALPHAVISION DOTS PICTURE file transmission frame format

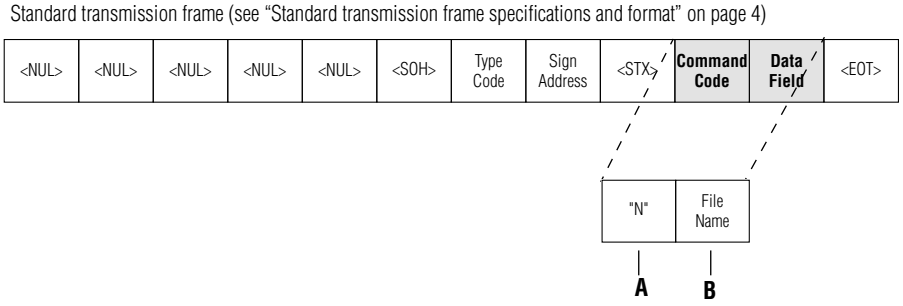


Item	Name			Description
A	Command Code			"M" (4DH) = Write ALPHAVISION DOTS PICTURE file
B	File Name			Nine ASCII characters that indicate the ALPHAVISION DOTS PICTURE file being accessed.
C	Data Field	DOTS PICTURE File Data	Height (y)	See "DOTS PICTURE File Data" in Table 19, "Write DOTS PICTURE file transmission frame format," on page 30.
			Width (x)	
			Row Bit Pattern	
			<CR>	
			<LF>	

3.2.5.2 Read ALPHAVISION DOTS PICTURE file Command Code — "N" (4EH)

NOTE: Whenever doing a "Read" command on a network with multiple signs, it's important that each sign has a unique Serial Address. Also, only one sign at a time should be written to or read from.

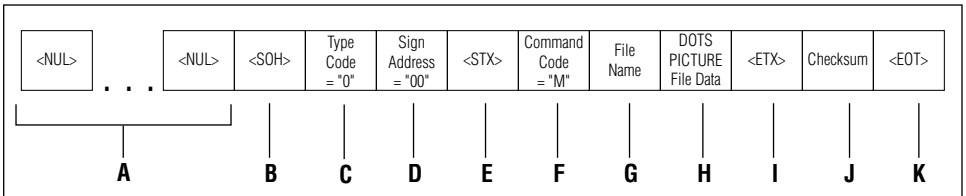
Table 23: Read ALPHAVISION DOTS PICTURE file transmission frame format



Item	Name		Description
A	Command Code		"N" (4EH) = Read ALPHAVISION DOTS PICTURE file
B	Data Field	File Name	Nine ASCII characters that indicate the ALPHAVISION DOTS PICTURE file being accessed.

Following the Read ALPHAVISION DOTS PICTURE file Command Code, a sign will respond with the following:

Table 24: Read ALPHAVISION DOTS PICTURE file sign response frame format



Item	Name	Description
A	<NUL>	Twenty <NUL>s (00H) characters
B	<SOH>	<SOH> (01H) character
C	Type Code	"0" (30H) is the Response code
D	Sign Address	"00" (30H + 30H) is sent regardless of the sign's actual address.
E	<STX>	<STX> (02H) character
F	Command Code	"M" is returned by the sign. (This is the Write ALPHAVISION DOTS PICTURE file Command Code.)
G	File Name	One ASCII character that indicates the DOTS PICTURE file being accessed. See "Appendix A: File Label format" on page 37.
H	DOTS PICTURE File Data	See Table 19, "Write DOTS PICTURE file transmission frame format," on page 30.
I	<ETX>	<ETX> (03H) character
J	Checksum	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
K	<EOT>	<EOT> (04H) character

3.2.6 ALPHAVISION BULLETIN MESSAGE

An ALPHAVISION BULLETIN MESSAGE allows a text message of up to 225 characters to be rotated on a sign's display without interrupting the current operation.

3.2.6.1 Write ALPHAVISION BULLETIN MESSAGE file Command Code — "O" (4FH)

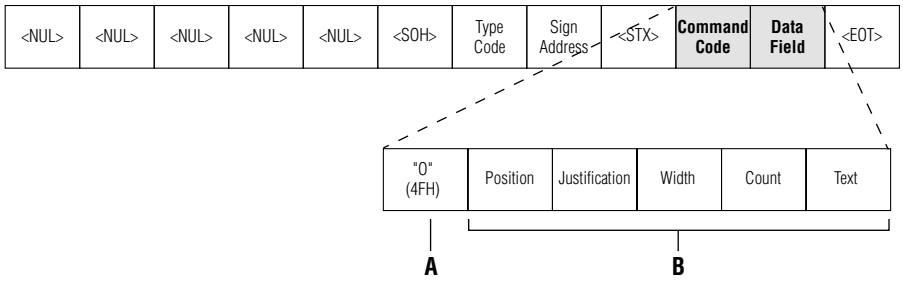
NOTE: Only the size of the ALPHAVISION BULLETIN MESSAGE window is cleared, not the entire line.

NOTE: Only seven high characters are supported.

NOTE: Only ALPHAVISION signs support the ability to vary window Position and Justification. An ALPHA Series 7000 sign displays an ALPHAVISION BULLETIN MESSAGE across the entire width of the sign.

Table 25: Write ALPHAVISION BULLETIN MESSAGE file transmission frame format

Standard transmission frame (see "Standard transmission frame specifications and format" on page 4)



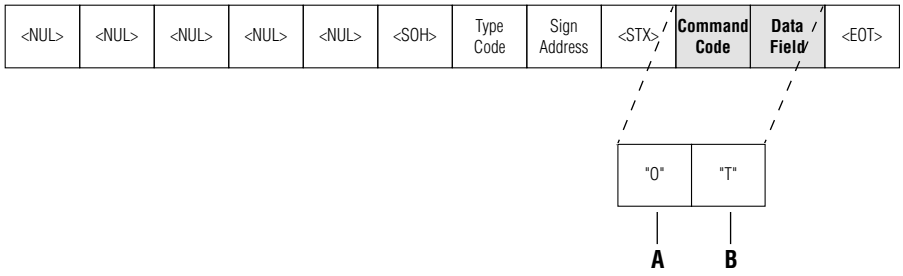
Item	Name		Description
A	Command Code		"O" (4FH) = Write ALPHAVISION BULLETIN MESSAGE file
B	Data Field	Position	One ASCII character that stands for the postion of the bulletin message on a sign. Valid values are: "T" 54H = Top of the display "B" 42H = Bottom of the display
		Justification	One ASCII character that stands for the alignment of the bulletin message on a sign. Valid values are: "L" 4CH = Left side of the display "C" 43H = Center of the display "R" 52H = Right side of the display
		Width	Two ASCII hexadecimal digits that specify the total number of characters in the Text field below. This number will be rounded up to the nearest 32-column width. For example, if the total number of characters = 78, this number would be rounded up to 32 x 3 = 96. The maximum Width is 255 ("FF").
		Count	Two ASCII hexadecimal digits that stand for the number of times the bulletin message should be displayed.
		Text	Up to 225 ASCII characters that comprise the actual bulletin message itself. Messages longer than 225 characters will be truncated. **NOTE:** The only ASCII Control characters allowed in a bulletin message are color codes. (For more information, see "Appendix I: ALPHA protocol ASCII table" on page 68.)

3.2.6.2 Terminate ALPHAVISION BULLETIN MESSAGE file Command Code — “OT” (4F + 54H)

To stop an ALPHAVISION BULLETIN MESSAGE before the Count field (above) has been reached, use this Command Code:

Table 26: Terminate ALPHAVISION BULLETIN MESSAGE file transmission frame format

Standard transmission frame (see “Standard transmission frame specifications and format” on page 4)



Item	Name	Description
A	Command Code	“O” (4FH)
B	Data Field	“T” (54H) is the terminator character.

4.0 APPENDIXES

4.1 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.

File Labels can be anywhere in the range 20H through 7EH inclusive.

The only special case occurs when File Label "0" (30H) is used for a Priority TEXT file (see "Priority TEXT files" on page 14) which is pre-configured as a set portion of memory outside of the Memory Pool.

4.1.1 Valid File Labels

Table 27: Permitted File Labels

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 - "l"
2DH - "-"	3DH - "="	4DH - "M"	5DH - "]"	6DH - "m"	7DH - "}"
2EH - "."	3EH - ">"	4EH - "N"	5EH - "¢"	6EH - "n"	7EH - 1/2 sp
2FH - "/"	3FH - "?"	4FH - "O"	5FH - "_"	6FH - "o"	7FH - reserved

NOTE: File Label "0" (30H) is used for a Priority TEXT file (see "Priority TEXT files" on page 14).

NOTE: If the Counter feature ("Appendix C: Counter information" on page 39) of a sign is used, then File Labels "1" (31H) through "5" (35H) are reserved for Target files.

NOTE: sp = space
1/2 sp = 1/2 space

4.2 Appendix B: TEXT file Start and Stop times

The Start and Stop times are represented in ASCII. For example, a 8:50 am time = 35H = “35” (the ASCII characters 33H and 35H):

4.2.1 Valid Start and Stop times

Table 28: 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 - FDH	NEVER - FEH	ALWAYS - FFH

4.3 Appendix C: Counter information

NOTE: In order to use counters, a sign must have a counter upgrade.

4.3.1 Displaying Counter values

SHOW ME

An example of displaying a Counter value is on page 56.

TEXT files can use Control codes to display counter values. (See "Counters" in the "Extended character set" in "Appendix I: ALPHA protocol ASCII table" on page 68).

4.3.2 Setting up Counters

4.3.2.1 Memory Configuration

The default Memory Configuration on EZ95 signs and all EZII signs *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" through "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). It is important to keep in mind that when writing a new Memory Configuration that TEXT files "1" through "5" need to be included, as these are the TARGET files. (See "Set Memory Configuration" in "Write SPECIAL FUNCTIONS Command Code — "E" (45H)" on page 15.)

4.3.2.2 Memory Dump

A Memory Dump response from a sign equipped with the counter upgrade also contains the counter information. (See "Memory Dump" in "Read SPECIAL FUNCTIONS Command Code — "F" (46H)" on page 21.)

4.3.2.3 Run Sequence

It is important to set up a Run Sequence which runs according to 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. (See "Set Run Sequence" in "Write SPECIAL FUNCTIONS Command Code — "E" (45H)" on page 15.)

4.3.2.4 Run Day Table

It is important to set up a Run Day Table 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). (See "Set Run Day Table" in "Write SPECIAL FUNCTIONS Command Code — "E" (45H)" on page 15.)

4.4 Appendix D: STRING file notes

In the EZII/EZ95 protocol, a STRING file is a short stream of data that is "called" from a TEXT file. A typical use of a STRING file would be to update a count (e.g., a count-down timer) that is continuously displayed on a sign.

4.4.1 Advantages of using STRING files

- When STRING files are used to update data on a sign, the sign won't "blink" or flash during the update. (However, a sign will blink when TEXT files are updated.)
- Using STRING files saves sign memory. For example, if some important data is displayed multiple times within a TEXT file, this data only needs to be stored once in a STRING file, then "called" from the appropriate location within the TEXT file.

4.4.2 Using STRING files example

To use STRING files, there are three basic steps:

STEP 1 — Allocate memory in a sign for the STRING file (and the TEXT file that calls it).

STEP 2 — Write the TEXT file which calls the STRING file.

STEP 3 — Update the STRING file.

SPECIAL NOTE

STEP 1 and STEP 2 are used to initialize a STRING file.

STEP 3 is used to change the information in a STRING file once it has been initialized.

NOTE: The default character spacing is proportional, rather than fixed width. Because of this, a sign's auto-centering will move the displayed data around with the changing character widths in order to keep the data centered.

To avoid this distracting data movement on a sign:

(a) always send the same number of characters in the STRING file data, and

(b) always use fixed width characters by embedding the following 2-byte sequence in your TEXT file *before* the STRING file call: 1EH (Control "^") + 31H ("1").

4.4.2.1 STEP 1 — Allocate memory for a STRING file (and the TEXT file that calls it)

To allocate memory for one STRING file and the TEXT file which calls the STRING file, the following transmission frame could be sent to a network of signs:

Table 29: Using STRING files example: STEP 1

This following is a Standard Transmission Frame (see "Standard transmission frame format" on page 5):

<NUL><NUL><NUL><NUL><NUL><SOH>"Z00"<STX>"E$AAU0400FF001BL00200000"<EOT>

A B C D E F G H

Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>	01H	Start Of Header character

Table 29: Using STRING files example: STEP 1

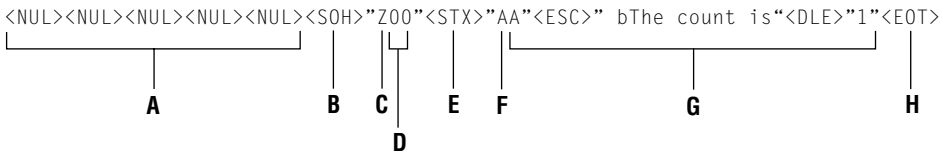
C	Type Code	"Z"	This means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address	"00"	This means all signs on the network should "listen" to this transmission.
E	<STX>	02H	Start of TeXt character
F	Command Code	"E"	This is the "Write SPECIAL FUNCTIONS" Command Code. (See "SPECIAL FUNCTIONS files" on page 15.)
G	Data Field	"$AAU0400FF001BL00200000"	"$" is the Write SPECIAL FUNCTIONS Command Code for **Set Memory Configuration** (see Table 13, "Write SPECIAL FUNCTIONS file transmission frame format," on page 15). The remaining characters have the following meaning: "A" = File Label of the TEXT file which will "call" the STRING file "A" = TEXT File Type "U" = this TEXT file is Unlocked "0400" = the TEXT file size in hexadecimal ("0400" = 1024D) "FF" = the TEXT file's Start Time ("FF" = Always) "00" = the TEXT file's Stop Time (even though the TEXT message will always run, the "00" must be included as padding) "1" = File Label of the STRING file "B" = STRING File Type "L" = this STRING file is Locked "0020" = the STRING file size in hexadecimal ("0020" = 32D). "0000" = padding
H	<EOT>	04H	End Of Transmission character

4.4.2.2 STEP 2 — Write the TEXT file which calls the STRING file

After allocating memory for the TEXT and the STRING files, write the TEXT file which will call the STRING file:

Table 30: Using STRING files example: STEP 2

This following is a Standard Transmission Frame (see "Standard transmission frame format" on page 5):



Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>	01H	Start Of Header character
C	Type Code	"Z"	This means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address	"00"	This means all signs on the network should "listen" to this transmission.
E	<STX>	02H	Start of TeXt character
F	Command Code	"A"	This is the "Write TEXT file" Command Code. (See Table 10, "Write TEXT file transmission frame format," on page 12.)

Table 30: Using STRING files example: STEP 2

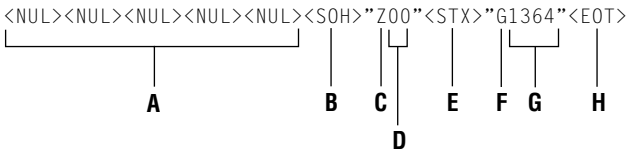
G	Data Field	"A"<ESC>" bThe count is "<DLE>"1"	The characters have the following meaning: "A" = File Label of the TEXT file which will include the STRING file "A" = TEXT File Type <ESC> (1BH) = signals the start of a Mode field " " (20H) = middle line position "b" = Hold Mode "The count is " = the text of this TEXT file <DLE> (10H) = Call STRING file "1" = the STRING File Label to call
H	<EOT>	04H	End Of Transmission character

4.4.2.3 STEP 3 — Update the STRING file

To update the STRING file data (e.g., "The count is 364"), this would be sent:

Table 31: Using STRING files example: STEP 3

This following is a Standard Transmission Frame (see "Standard transmission frame format" on page 5):



Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>	01H	Start Of Header character
C	Type Code	"Z"	This means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address	"00"	This means all signs on the network should "listen" to this transmission.
E	<STX>	02H	Start of TeXt character
F	Command Code	"G"	This is the "Write STRING file" Command Code. (See Table 16, "Write STRING file transmission frame format," on page 28.)
G	Data Field	"1364"	The characters have the following meaning: "1" = the STRING File Label to write to "364" = the STRING file data
H	<EOT>	04H	End Of Transmission character

4.5 Appendix E: Sample programs

Other sample programs will be included at Adaptive's web site: www.ams-i.com.

4.5.1 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 - 1998.  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;
};

unionREGS  {
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);
}
```

4.5.2 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
```

4.6 Appendix F: Network cabling and sign connector pinouts

Following is a list of commonly used cabling and connectors used to network ALPHA signs:

NOTE: For more information on networking signs, see the **Network Configurations** manual (pn 9708-8046).

4.6.1 Network cabling pinouts

Table 32: Network cabling pinouts

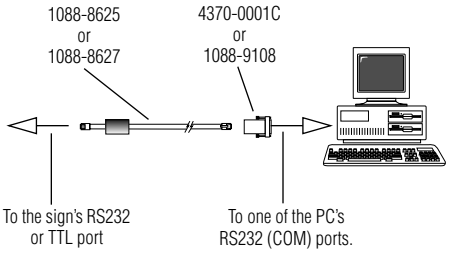
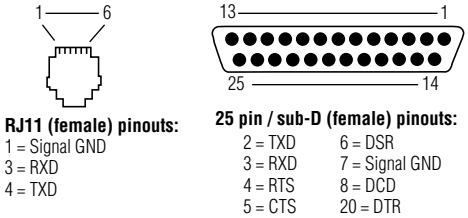
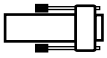
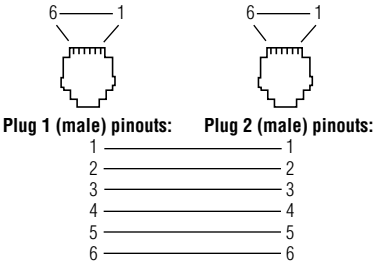
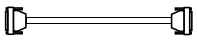
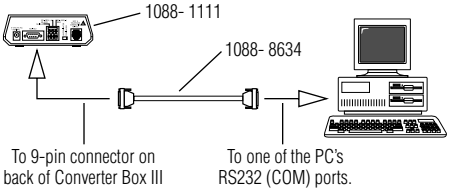
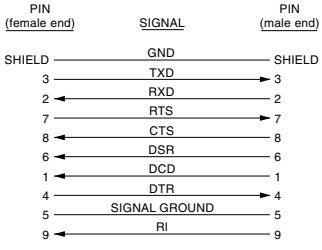
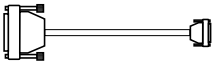
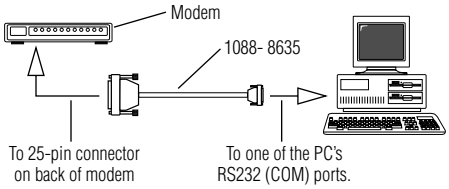
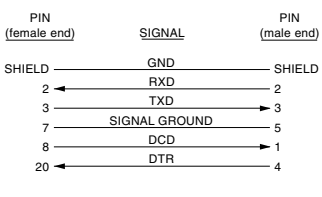
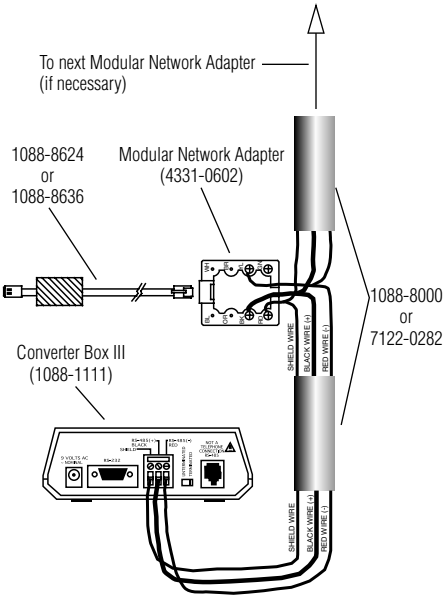
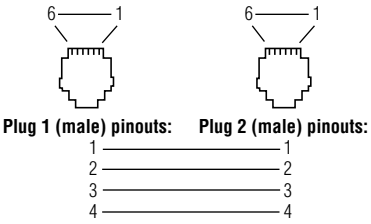
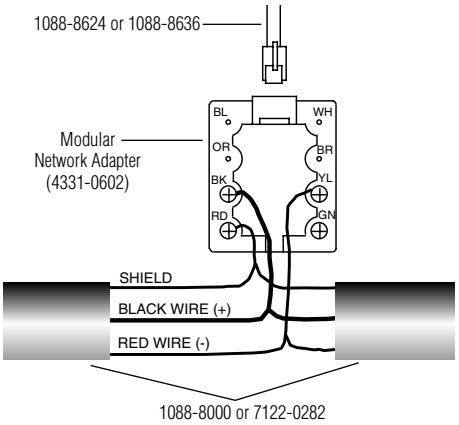
Part #	Item	Function	Pinout
4370-0001C	25 pin sub-D / 6 pos. RJ11: 		 **RJ11 (female) pinouts:** 1 = Signal GND 3 = RXD 4 = TXD **25 pin / sub-D (female) pinouts:** 2 = TXD 3 = RXD 4 = RTS 5 = CTS 6 = DSR 7 = Signal GND 8 = DCD 20 = DTR
1088-9108	9 pin sub-D / 6 pos. RJ11: 		 **RJ11 (female) pinouts:** 1 = Signal GND 3 = RXD 4 = TXD **9 pin / sub-D (female) pinouts:** 1 = DCD 2 = RXD 3 = TXD 4 = DTR 5 = Signal GND 6 = DSR 7 = RTS 8 = CTS
1088-8625 & 1088-8627	6-connector RS232 data cable:  1088-8625 is 25 feet long. 1088-8627 is 50 feet long.		
1088-8634	10 foot, 9 pin-to-9 pin type "A9" RS232 cable: 		
1088-8635	10 foot, 25 pin-to-9 pin type "B9" RS232 cable: 		

Table 32: Network cabling pinouts

1088-8624 & 1088-8636	4-connector RS485 data cable:  1088-8624 is 8 feet long. 1088-8636 is 1 foot long.	
4331-0602	Modular Network Adapter 	
1088-8000 & 7122-0282	RS485 network cabling:  1088-8000 is a 1000 foot spool. 7122-0282 is a 100 foot spool.	

4.6.2 Sign connector pinouts

Table 33: Sign connector pinouts

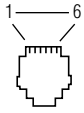
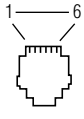
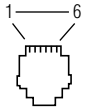
Type	Pinout	Sign type											
		2.1" ALPHAVISION	3.2" ALPHAVISION	7000 Series	4000 Series	300 Series	200 Series	ALPHAVISION (Full Matrix)	BETA-BRITE	Big Dot	Director	PPD	Solar
RS485	 **Pinouts:** 1 = No connect 2 = RS485 (+) 3 = No connect 4 = No connect 5 = RS485 (-) 6 = No connect	●	●	●	●	●	●	●		●	●		
	This connector is inside the sign.  RS485(-) RS485(+) SHIELD												●
RS232 / RS485 / TTL	 **Pinouts:** 1 = GND 2 = RS485 (+) 3 = RXD 4 = TXD 5 = RS485 (-) 6 = +5V (200 mA max)	●	●	●	●	●	●	●		●	●	●	

Table 33: Sign connector pinouts

Type	Pinout	Sign type											
		2.1" ALPHAVISION	3.2" ALPHAVISION	7000 Series	4000 Series	300 Series	200 Series	ALPHAVISION (Full Matrix)	BETA-BRITE	Big Dot	Director	PPD	Solar
RS232	 **Pinouts:** 1 = GND 2 = No connect 3 = RXD 4 = TXD (TTL) 5 = No connect 6 = +5V (200 mA max)								●				
	This connector is inside the sign.  Fused ISO +5V TXD RXD RS232 GND												●

4.7 Appendix G: Protocol examples

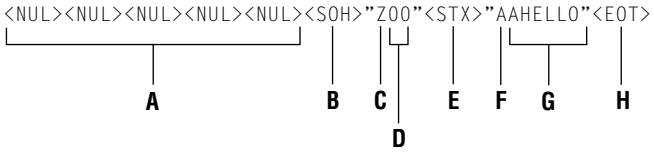
NOTE: In the following examples, it's assumed that the Memory Configuration table (**Table 13** on page 15) in each sign has already been set up properly.

4.7.1 Standard transmission frame examples (page 5)

4.7.1.1 Send a message to all signs on a network example

The following example will display "HELLO" on all signs attached to a network:

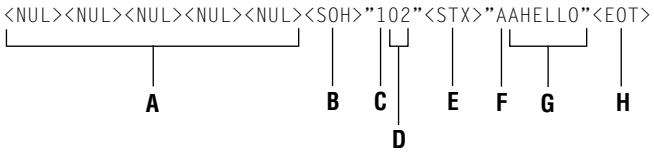
Table 34: Send a message to all signs example



Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"Z"	This means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address		"00"	This means all signs on the network should "listen" to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		"A"	This is the "Write TEXT file" Command Code. (See Table 10, "Write TEXT file transmission frame format," on page 12.)
G	Data Field	File Label	"A"	File Label of the TEXT file
		ASCII Message	"HELLO"	The actual text to be displayed on a sign
H	<EOT>		04H	End Of Transmission character

4.7.1.2 Send a message to all 1-line signs on a network with a Sign Address of 02H example

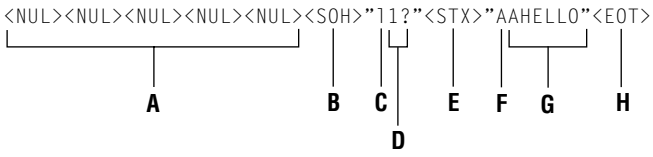
Table 35: Send a message to all 1-line signs on a network with a Sign Address of 02H example



Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"1"	This means that this transmission is directed to all 1-line signs.
D	Sign Address		"02"	This means only 1-line signs with a Sign Address of 02H on the network should "listen" to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		"A"	This is the "Write TEXT file" Command Code. (See Table 10, "Write TEXT file transmission frame format," on page 12.)
G	Data Field	File Label	"A"	File Label of the TEXT file
		ASCII Message	"HELLO"	The actual text to be displayed on a sign
H	<EOT>		04H	End Of Transmission character

4.7.1.3 Send a message to all Series 7000 signs on a network with Sign Addresses 10H through 1FH example

Table 36: Send a message to all Series 7000 signs on a network with Sign Addresses 10H through 1FH example

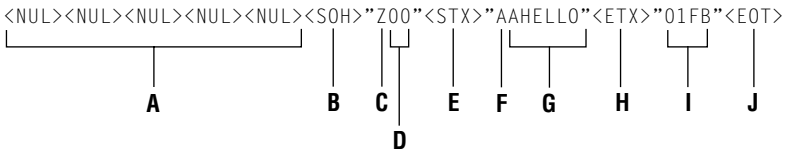


Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>		01H	Start Of Header character
C	Type Code		“I”	This means that this transmission is directed to all Series 7000 signs.
D	Sign Address		“1?”	This means only Series 7000 signs with Sign Addresses between 10H and 1FH inclusive on the network should “listen” to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		“A”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 12.)
G	Data Field	File Label	“A”	File Label of the TEXT file
		ASCII Message	“HELLO”	The actual text to be displayed on a sign
H	<EOT>		04H	End Of Transmission character

4.7.2 Transmission frame with Checksum example (page 6)

This example is identical to the previous example in Table 4.7.1.1, “Send a message to all signs on a network example,” on page 48 except that a Checksum is used in the following example:

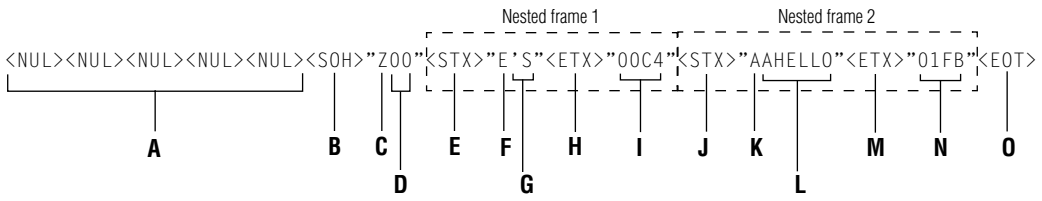
Table 37: Transmission frame with Checksum example



Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>		01H	Start Of Header character
C	Type Code		“Z”	This means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address		“00”	This means all signs on the network should “listen” to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		“A”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 12.)
G	Data Field	File Label	“A”	File Label of the TEXT file
		ASCII Message	“HELLO”	The actual text to be displayed on a sign
H	<ETX>		03H	End of TeXt (03H) character
I	Checksum		“01FB”	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> (item E) through the previous <ETX> (item H) inclusive. The most significant digit is first.
J	<EOT>		04H	End Of Transmission character

4.7.3 Nesting with checksums transmission frame example (page 7)

Table 38: Nesting with checksums transmission frame example

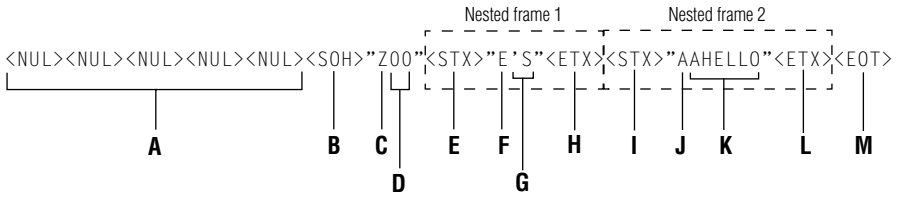


Item	Name			Value	Description
A	<NUL>			00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>			01H	Start Of Header character
C	Type Code			"Z"	This means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address			"00"	This means all signs on the network should "listen" to this transmission.
E	Nested frame 1	<STX>		02H	Start of Nested frame 1
F	Nested frame 1	Command Code		"E"	This is the "Write SPECIAL FUNCTIONS" Command Code. (See Table 13, "Write SPECIAL FUNCTIONS file transmission frame format," on page 15.)
G	Nested frame 1	Data Field	Special Functions Label	""	"" (27H) means Set Time Format
	Nested frame 1	Data Field	Special Functions Data	"S"	This sets the sign's time to the standard am/pm format.
H	Nested frame 1	<ETX>		03H	End of Nested frame 1
I	Nested frame 1	Checksum		"00C4"	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> (item E) through the previous <ETX> (item I) inclusive. The most significant digit is first.
J	Nested frame 2	<STX>		02H	Start of Nested frame 2
K	Nested frame 2	Command Code		"A"	This is the "Write TEXT file" Command Code. (See Table 10, "Write TEXT file transmission frame format," on page 12.)
L	Nested frame 2	Data Field	File Label	"A"	File Label of the TEXT file
	Nested frame 2	Data Field	ASCII Message	"HELLO"	The actual text to be displayed on a sign
M	Nested frame 2	<ETX>		03H	End of Nested frame 2
N	Nested frame 2	Checksum		"01FB"	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> (item E) through the previous <ETX> (item H) inclusive. The most significant digit is first.
O	<EOT>			04H	End Of Transmission character

4.7.4 Nesting without Checksum transmission frame example (page 8)

This frame is identical to the previous frame in **Table 38** on page 50 except that the Checksums are omitted after each nested frame's <ETX>:

Table 39: Nesting without Checksums transmission frame example



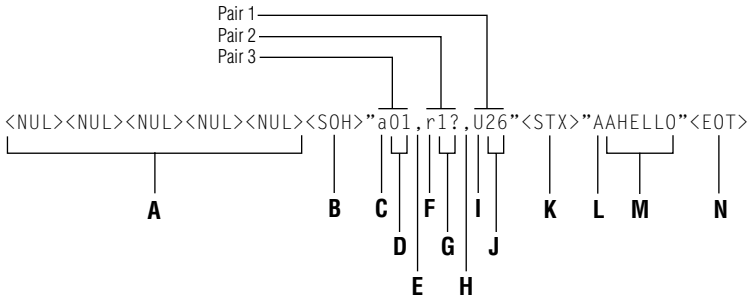
Item	Name			Value	Description
A	<NUL>			00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>			01H	Start Of Header character
C	Type Code			"Z"	This means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address			"00"	This means all signs on the network should "listen" to this transmission.
E	Nested frame 1	<STX>		02H	Start of Nested frame 1
F	Nested frame 1	Command Code		"E"	This is the "Write SPECIAL FUNCTIONS" Command Code. (See Table 13, "Write SPECIAL FUNCTIONS file transmission frame format," on page 15.)
G	Nested frame 1	Data Field	Special Functions Label	""	"" (27H) means Set Time Format
			Special Functions Data	"S"	This sets the sign's time to the standard am/pm format.
H	Nested frame 1	<ETX>		03H	End of Nested frame 1
I	Nested frame 2	<STX>		02H	Start of Nested frame 2
J	Nested frame 2	Command Code		"A"	This is the "Write TEXT file" Command Code. (See Table 10, "Write TEXT file transmission frame format," on page 12.)
K	Nested frame 2	Data Field	File Label	"A"	File Label of the TEXT file
			ASCII Message	"HELLO"	The actual text to be displayed on a sign
L	Nested frame 2	<ETX>		03H	End of Nested frame 2 (Optional when <EOT> is the next character.)
M	<EOT>			04H	End Of Transmission character

4.7.5 Multiple Type Codes / Sign Addresses example (page 5)

In this example three Type Code/Sign Address pairs are shown:

NOTE: The effects of Type Codes are cumulative. For instance, in this example the message would be sent to all 4120C signs and Director signs and 790i signs on the network.

Table 40: Multiple Type Codes / Sign Addresses example



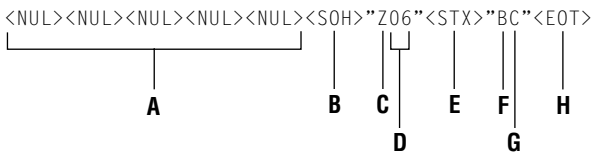
Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Pair 1	Type Code	"a"	This means that this transmission is directed to all 4120C signs.
D	Pair 1	Sign Address	"01"	This means only 4120C signs with a Sign Address of 01H on the network should "listen" to this transmission.
E	Delimiter		","	This separates each Type Code/Sign Address pair.
F	Pair 2	Type Code	"r"	This means that this transmission is directed to all Director signs.
G	Pair 2	Sign Address	"1?"	This means that all signs with a Sign Address between 10H and 1FH inclusive on the network should "listen" to this transmission.
H	Delimiter		","	This separates each Type Code/Sign Address pair.
I	Pair 3	Type Code	"U"	This means that this transmission is directed to all 790i signs.
J	Pair 3	Sign Address	"26"	This means only 790i signs with a Sign Address of 26H on the network should "listen" to this transmission.
K	<STX>		02H	Start of TeXt character
L	Command Code		"A"	This is the "Write TEXT file" Command Code. (See Table 10, "Write TEXT file transmission frame format," on page 12.)
M	Data Field	File Label	"A"	File Label of the TEXT file
M	Data Field	ASCII Message	"HELLO"	The actual text to be displayed on a sign
N	<EOT>		04H	End Of Transmission character

4.7.6 TEXT file examples

4.7.6.1 Read TEXT file example (page 13)

The response to this read file request is shown in **Table 42** on page 53.

Table 41: Read TEXT file example



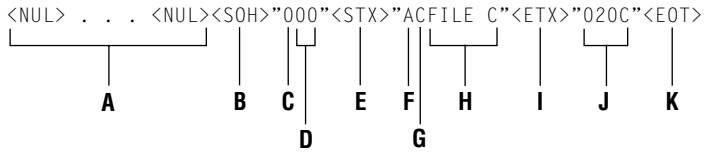
Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>		01H	Start Of Header character
C	Type Code		“Z”	This means that this transmission is directed to all signs.
D	Sign Address		“06”	This means only signs with a Sign Address of 06H on the network should “listen” to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		“B”	This is the “Read TEXT file” Command Code. (See Table 11, “Read TEXT file transmission frame format,” on page 13.)
G	Data Field	File Label	“C”	File Label of the TEXT file to read
H	<EOT>		04H	End Of Transmission character

4.7.6.2 Response to Read TEXT file example (page 13)

This is the response to the read file request shown in the **Table 41** on page 53.

NOTE: For the sake of this example, we’ll assume that the TEXT file with the File Label “C” just contains the text “FILE C”.

Table 42: Response to Read TEXT file example



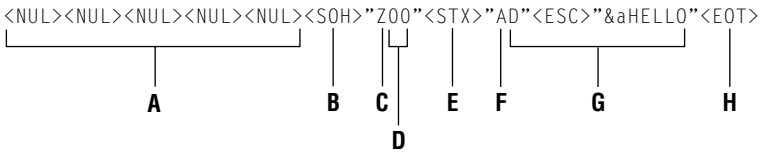
Item	Name	Value	Description
A	<NUL>	00H	Twenty <NUL> characters
B	<SOH>	01H	Start Of Header character
C	Type Code	“0”	The Response Type Code
D	Sign Address	“00”	“00” is always sent.
E	<STX>	02H	Start of TeXt character
F	Command Code	“A”	This is sent in response to the “Read TEXT file” Command Code.
G	File Label	“C”	File Label of the TEXT file that is being read
H	TEXT file data format	“FILE C”	The actual text stored in TEXT file “C”
I	<ETX>	03H	End of TeXt character
J	Checksum	“020C”	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> (item E) through the previous <ETX> (item H) inclusive. The most significant digit is first.
K	<EOT>	04H	End Of Transmission character

4.7.6.3 TEXT file data format examples

4.7.6.3.1 Rotate “Hello” example (page 12)

This example uses the Rotate Mode to move the text “HELLO” on the bottom line of a sign:

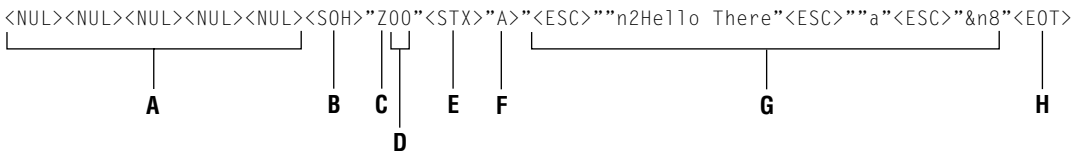
Table 43: Rotate “Hello” example



Item	Name			Value	Description
A	<NUL>			00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>			01H	Start Of Header character
C	Type Code			“Z”	This means that this transmission is directed to all signs.
D	Sign Address			“00”	This means all signs on the network should “listen” to this transmission.
E	<STX>			02H	Start of TeXt character
F	Command Code			“A”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 12.)
G	Data Field	File Label		“D”	File Label of the TEXT file that will be written
		Mode Field	<ESC>	1BH	Escape character
			Display Position	“&”	This means that the ASCII Message should be displayed on the bottom line of a sign.
			Mode Code	“a”	Rotate code.
		ASCII Message		“HELLO”	The actual text to be displayed
H	<EOT>			04H	End Of Transmission character

4.7.6.3.2 Combining text and graphics example (page 12)

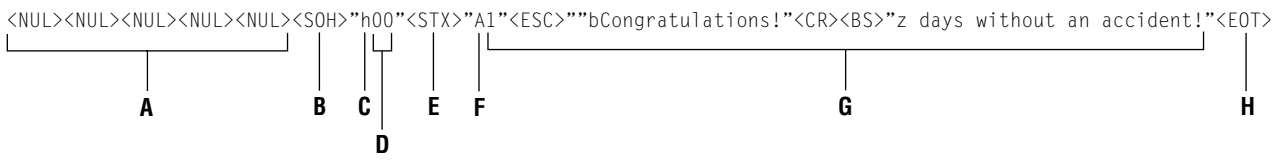
Table 44: Combining text and graphics example



Item	Name				Value	Description
A	<NUL>				00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>				01H	Start Of Header character
C	Type Code				“Z”	This means that this transmission is directed to all signs.
D	Sign Address				“00”	This means all signs on the network should “listen” to this transmission.
E	<STX>				02H	Start of TeXt character
F	Command Code				“A”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 12.)
G	Data Field	File Label			“>”	File Label of the TEXT file that will be written
		TEXT file data format	Mode Field	<ESC>	<ESC>	<ESC> (1BH) always starts the Mode Field
				Display Position	“ “	“ “ (22H) means that the ASCII Message will begin on the Top Line of the sign
				Mode Code	“n”	“n” (6EH) is used in conjunction with the Special Specifier to use the Special Modes (see “Appendix H: Modes” on page 66).
				Special Specifier	“2”	“2” (32H) means that the Special Mode called SNOW will be used.
			ASCII Message		“Hello There”	The actual text to be displayed
		TEXT file data format	Mode Field	<ESC>	<ESC>	<ESC> (1BH) always starts the Mode Field
				Display Position	“ “	“ “ (22H) means the Top Line of the sign.
				Mode Code	“a”	“a” (61H) is the ROTATE Mode Code. This means that the previous ASCII Message (“Hello There”) will be ROTATEd off the Top Line of the sign. This is often referred to as a “Trailing Mode”.
			ASCII Message			In this case, there is no ASCII Message because of the “trailing” ROTATE Mode.
		TEXT file data format	Mode Field	<ESC>	<ESC>	<ESC> (1BH) always starts the Mode Field
				Display Position	“&”	“&“ (22H) means that the ASCII Message will begin on the Bottom Line of the sign
				Mode Code	“n”	“n” (6EH) is used in conjunction with the Special Specifier to use the Special Modes (see “Appendix H: Modes” on page 66).
				Special Specifier	“8”	“8” (38H) means that the Special Mode called WELCOME will be used.
			ASCII Message			In this case, there is no ASCII Message because of the WELCOME animation.
H	<EOT>				04H	End Of Transmission character

4.7.6.3.3 Displaying a Counter value example (page 12)

Table 45: Displaying a Counter value example

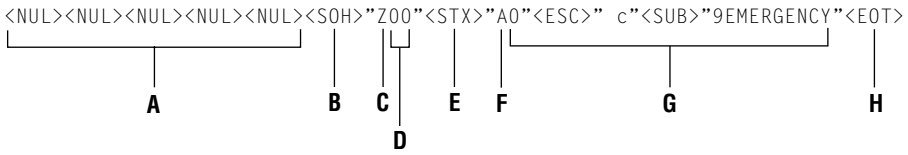


Item	Name				Value	Description
A	<NUL>				00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>				01H	Start Of Header character
C	Type Code				"h"	This means that this transmission is directed to all 4160R signs.
D	Sign Address				"00"	This means all 4160R signs on the network should "listen" to this transmission.
E	<STX>				02H	Start of TeXt character
F	Command Code				"A"	This is the "Write TEXT file" Command Code. (See Table 10, "Write TEXT file transmission frame format," on page 12.)
G	Data Field	File Label			"1"	File Label of the TEXT file
		TEXT file data format	Mode Field	<ESC>	<ESC>	<ESC> (1BH) always starts the Mode Field
				Display Position	"""	""" (22H) means that the ASCII Message will begin on the Top Line of the sign
				Mode Code	"b"	"b" (62H) is the HOLD Mode Code (see page 66)
			ASCII Message		"Congratulations"<CR><BS>"z days without an accident!"	The actual text (with Control Codes) to be displayed on a sign. These Control Codes are used: <CR> (0DH) = means that text after the <CR> will be an the next line of the sign <BS> (08H) + "z" = a 2-byte code used to display a counter, in this case Counter 1 (see "Appendix C: Counter information" on page 39).
H	<EOT>				04H	End Of Transmission character

4.7.7 Priority TEXT file examples (page 14)

4.7.7.1 Write a Priority TEXT file example

Table 46: Write a Priority TEXT file example

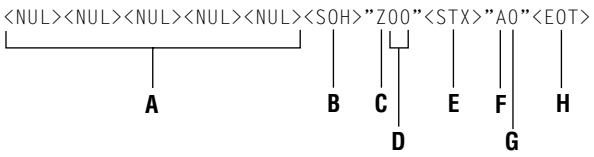


Item	Name				Value	Description
A	<NUL>				00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>				01H	Start Of Header character
C	Type Code				“Z”	This means that this transmission is directed to all signs.
D	Sign Address				“00”	This means all signs on the network should “listen” to this transmission.
E	<STX>				02H	Start of TeXt character
F	Command Code				“A”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 12.)
G	Data Field	File Label			“0”	Priority TEXT File Label
		TEXT file data format	Mode Field	<ESC>	<ESC>	<ESC> (1BH) always starts the Mode Field
				Display Position	“ “	“ “ (20H) means that the ASCII Message will be on the Middle Line of the sign
				Mode Code	“c”	“c” (62H) is the FLASH Mode Code (see page 66)
			ASCII Message		<SUB>“9EMERGENCY”	The actual text (with Control Codes) to be displayed on a sign. These Control Codes are used: <SUB> (1AH) + “9” = a 2-byte code used to select a character set, in this case Full Height Standard (see “Appendix I: ALPHA protocol ASCII table” on page 68).
H	<EOT>				04H	End Of Transmission character

4.7.7.2 Disable a Priority TEXT file example

The following transmission will disable the Priority TEXT file. Whatever was running on a sign *before* the Priority TEXT file was sent will resume running.

Table 47: Disable a Priority TEXT file example



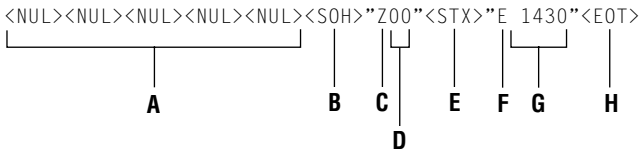
Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>		01H	Start Of Header character
C	Type Code		“Z”	This means that this transmission is directed to all signs.
D	Sign Address		“00”	This means all signs on the network should “listen” to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		“A”	This is the “Write TEXT file” Command Code. (See Table 10, “Write TEXT file transmission frame format,” on page 12.)
G	Data Field	File Label	“0”	Priority TEXT File Label
H	<EOT>		04H	End Of Transmission character

4.7.8 SPECIAL FUNCTIONS examples (page 15)

4.7.8.1 Write SPECIAL FUNCTIONS example (page 15)

The following sets the time on all networked signs to 2:30 pm (1430 in 24-hour format):

Table 48: Write SPECIAL FUNCTIONS example

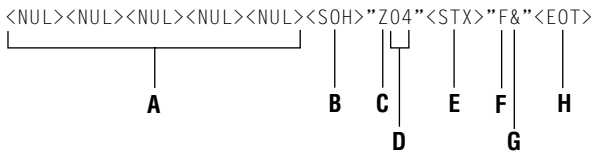


Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>		01H	Start Of Header character
C	Type Code		“Z”	This means that this transmission is directed to all signs.
D	Sign Address		“00”	This means all signs on the network should “listen” to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		“E”	This is the “Write SPECIAL FUNCTIONS file” Command Code. (See Table 13, “Write SPECIAL FUNCTIONS file transmission frame format,” on page 15.)
G	Data Field	Special Functions Label	“ ”	" “ (20H) = Set Time of Day
		Special Functions Data	“1430”	The time to set (in 24-hour format)
H	<EOT>		04H	End Of Transmission character

4.7.8.2 Read SPECIAL FUNCTIONS example (page 21)

The following reads the day of week from a sign with a Sign Address of 4:

Table 49: Read SPECIAL FUNCTIONS example

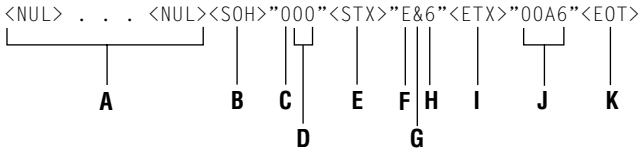


Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>		01H	Start Of Header character
C	Type Code		“Z”	This means that this transmission is directed to all signs.
D	Sign Address		“04”	This means all signs on the network should “listen” to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		“F”	This is the “Read SPECIAL FUNCTIONS file” Command Code. (See Table 14, “Read SPECIAL FUNCTIONS file transmission frame format,” on page 21.)
G	Data Field	Special Functions Label	“&”	"&“ (26H) = Read Day of Week
H	<EOT>		04H	End Of Transmission character

4.7.8.3 Response to Read SPECIAL FUNCTIONS example (page 27)

The following is the response to the Read SPECIAL FUNCTIONS example in **Table 49** above:

Table 50: Response to Read SPECIAL FUNCTIONS example



Item	Name	Value	Description
A	<NUL>	00H	Twenty <NUL> (00H) characters
B	<SOH>	01H	Start Of Header character
C	Type Code	“0”	“0” (30H) is the Response code
D	Sign Address	“00”	“00” (30H + 30H) is sent regardless of the sign’s actual address.
E	<STX>	02H	Start of TeXt character
F	Command Code	“E”	This is the “Read SPECIAL FUNCTIONS file” Command Code. (See Table 14, “Read SPECIAL FUNCTIONS file transmission frame format,” on page 21.)
G	Special Functions Label	“&”	"&“ (26H) = Read Day of Week
H	Special Functions Data	“6”	“6” (36H) stands for Friday
I	<ETX>	03H	End of TeXt character
J	Checksum	“00A6”	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
K	<EOT>	04H	End Of Transmission character

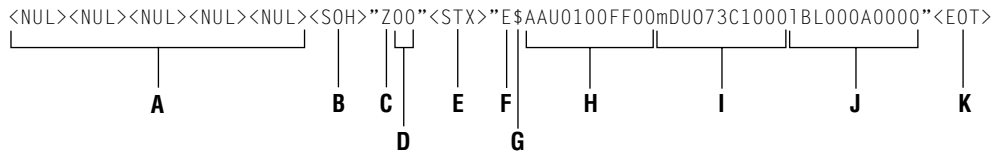
4.7.8.4 SPECIAL FUNCTIONS data formats example (page 15)

4.7.8.4.1 Set Memory Configuration example #1 — Counter data not included

This example writes the following file information to all signs:

- a TEXT file “A”, unlocked, 265 (100H) bytes in length, to run always
- a DOTS PICTURE file “m”, unlocked, 7 x 60 (rows x columns), one color
- a STRING file “l”, locked, 10 bytes in length

Table 51: Set Memory Configuration example #1 — Counter data not included



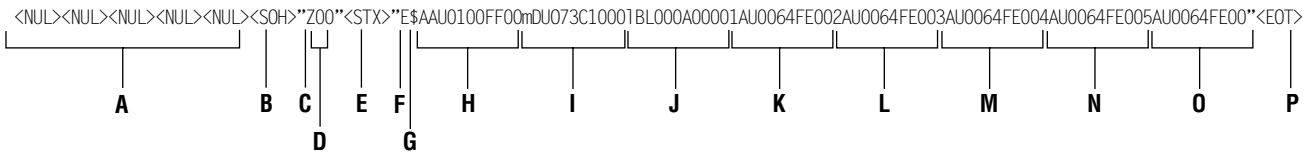
Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>		01H	Start Of Header character
C	Type Code		“Z”	This means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address		“00”	This means all signs on the network should “listen” to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		“E”	This is the “Write SPECIAL FUNCTIONS file” Command Code. (See Table 13, “Write SPECIAL FUNCTIONS file transmission frame format,” on page 15.)
G	Special Functions Label		“$”	“$” (24H) means Set Memory Configuration
H	Special Functions Data	TEXT file	“AAU0100FF00”	These bytes mean the following: “A” = File Label “A” = file type (in this case, a STRING file) “U” = an unlocked file “0100” = the size of this file in bytes (256D) “FF” = the TEXT file’s Start Time (in this case Always) “00” = the TEXT file’s Stop Time (ignored when the Start Time is Always)
I		DOTS PICTURE file	“mDU073C1000”	These bytes mean the following: “m” = File Label “D” = file type (in this case, a DOTS PICTURE file) “U” = an unlocked file “07” = number of pixel rows in the DOTS PICTURE file (7D) “3C” = number of pixel columns in the DOTS PICTURE file (60D) “1000” = a monochrome DOTS PICTURE file
J		STRING file	“lBL000A0000”	These bytes mean the following: “l” = File Label “B” = file type (in this case, a TEXT file) “L” = a locked file “000A” = the size of this file in bytes (10D) “0000” = these are just placeholders for a STRING file
K	<EOT>		04H	End Of Transmission character

4.7.8.4.2 Set Memory Configuration example #2 — Counter data included

The Memory Configuration from the previous example (**Table 51**) is used. However, in this example, in order to use a sign's Counters, the five Target files must be set up. (See also "Appendix C: Counter information" on page 39.)

NOTE: Once a Current Counter Value reaches its Counter Target Value, all Target files are triggered (as set up in the Target File Byte). This means that the Start Times for the appropriate Target files will be automatically set to Always.

Table 52: Set Memory Configuration example #2 — Counter data included



Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called "autobauding".)
B	<SOH>		01H	Start Of Header character
C	Type Code		"Z"	This means that this transmission is directed to all the sign types (i.e., 430i, 4120R, etc.).
D	Sign Address		"00"	This means all signs on the network should "listen" to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		"E"	This is the "Write SPECIAL FUNCTIONS file" Command Code. (See Table 13, "Write SPECIAL FUNCTIONS file transmission frame format," on page 15.)
G	Special Functions Label		"$"	"$" (24H) means Set Memory Configuration
H	Special Functions Data	TEXT file	"AAU0100FF00"	These bytes mean the following: "A" = File Label "A" = file type (in this case, a TEXT file) "U" = an unlocked file "0100" = the size of this file in bytes (256D) "FF" = the TEXT file's Start Time (in this case Always) "00" = the TEXT file's Stop Time (ignored when the Start Time is Always)
I		DOTS PICTURE file	"mDU073C1000"	These bytes mean the following: "m" = File Label "D" = file type (in this case, a DOTS PICTURE file) "U" = an unlocked file "07" = number of pixel rows in the DOTS PICTURE file (7D) "3C" = number of pixel columns in the DOTS PICTURE file (60D) "1000" = a monochrome DOTS PICTURE file
J		STRING file	"IBL000A0000"	These bytes mean the following: "I" = File Label "B" = file type (in this case, a STRING file) "L" = a locked file "000A" = the size of this file in bytes (10D) "0000" = these are just placeholders for a STRING file
K		TEXT file (this is the Target File for Counter 1)	"1AU0064FE00"	These bytes mean the following: "1" = File Label for Counter 1 Target File "A" = file type (in this case, a TEXT file) "U" = an unlocked file "0064" = the size of this file in bytes (100D) "FE" = the TEXT file's Start Time (in this case Never) "00" = the TEXT file's Stop Time (ignored when the Start Time is Never)

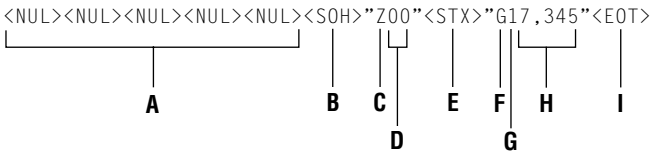
Table 52: Set Memory Configuration example #2 — Counter data included

L	Special Functions Data (continued)	TEXT file (this is the Target File for Counter 2)	"2AU0064FE00"	These bytes mean the following: "2" = File Label for Counter 2 Target File "A" = file type (in this case, a TEXT file) "U" = an unlocked file "0064" = the size of this file in bytes (100D) "FE" = the TEXT file's Start Time (in this case Never) "00" = the TEXT file's Stop Time (ignored when the Start Time is Never)
M		TEXT file (this is the Target File for Counter 3)	"3AU0064FE00"	These bytes mean the following: "3" = File Label for Counter 3 Target File "A" = file type (in this case, a TEXT file) "U" = an unlocked file "0064" = the size of this file in bytes (100D) "FE" = the TEXT file's Start Time (in this case Never) "00" = the TEXT file's Stop Time (ignored when the Start Time is Never)
N		TEXT file (this is the Target File for Counter 4)	"4AU0064FE00"	These bytes mean the following: "4" = File Label for Counter 4 Target File "A" = file type (in this case, a TEXT file) "U" = an unlocked file "0064" = the size of this file in bytes (100D) "FE" = the TEXT file's Start Time (in this case Never) "00" = the TEXT file's Stop Time (ignored when the Start Time is Never)
O		TEXT file (this is the Target File for Counter 5)	"5AU0064FE00"	These bytes mean the following: "5" = File Label for Counter 5 Target File "A" = file type (in this case, a TEXT file) "U" = an unlocked file "0064" = the size of this file in bytes (100D) "FE" = the TEXT file's Start Time (in this case Never) "00" = the TEXT file's Stop Time (ignored when the Start Time is Never)
P	<EOT>		04H	End Of Transmission character

4.7.9 STRING file examples (page 28)

4.7.9.1 Write STRING file example (page 28)

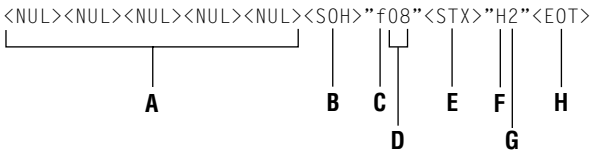
Table 53: Write STRING file example



Item	Name		Value	Description
A	<NUL>		00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>		01H	Start Of Header character
C	Type Code		“Z”	This means that this transmission is directed to all signs.
D	Sign Address		“00”	This means all signs on the network should “listen” to this transmission.
E	<STX>		02H	Start of TeXt character
F	Command Code		“G”	This is the “Write STRING file” Command Code. (See Table 16, “Write STRING file transmission frame format,” on page 28.)
G	Data Field	File Label	“1”	File Label of the STRING file
H		STRING File Data	“7,345”	This is the actual STRING file data.
I	<EOT>		04H	End Of Transmission character

4.7.9.2 Read STRING file example (page 29)

Table 54: Read STRING file example

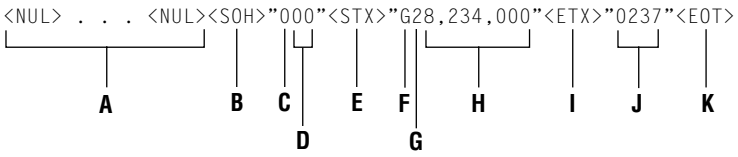


Item	Name	Value	Description
A	<NUL>	00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>	01H	Start Of Header character
C	Type Code	“f”	This means that this transmission is directed to all 215C signs.
D	Sign Address	“08”	This means all 215C signs with an address of 08H on the network should “listen” to this transmission.
E	<STX>	02H	Start of TeXt character
F	Command Code	“H”	This is the “Read STRING file” Command Code. (See Table 17, “Read STRING file transmission frame format,” on page 29.)
G	File Label	“2”	File Label of the STRING file to read
H	<EOT>	04H	End Of Transmission character

4.7.9.3 Response to Read STRING file example (page 29)

The following would be the response from the previous (**Table 54**) example:

Table 55: Response to Read STRING file example



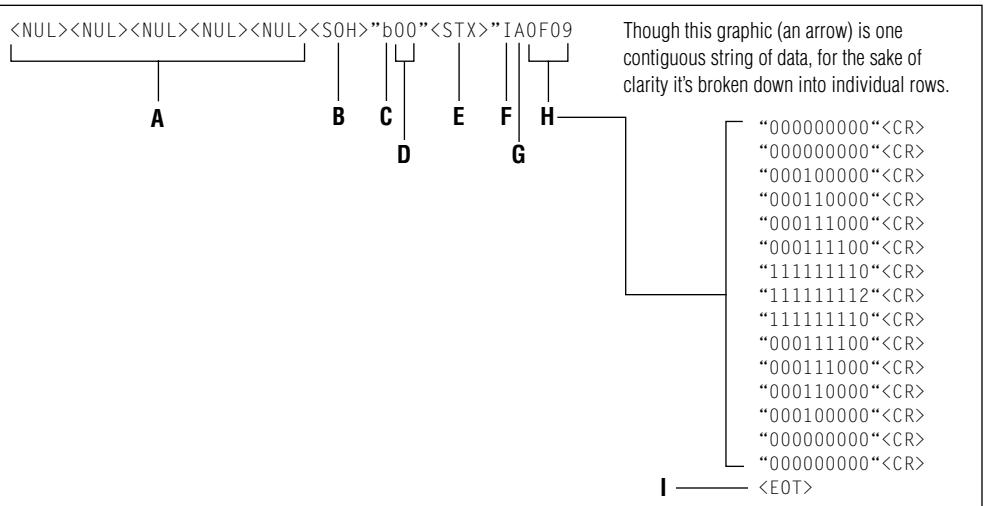
Item	Name	Value	Description
A	<NUL>	00H	Twenty <NUL> (00H) characters
B	<SOH>	01H	Start Of Header character
C	Type Code	“0”	“0” (30H) is the Response code
D	Sign Address	“00”	“00” (30H + 30H) is sent regardless of the sign’s actual address.
E	<STX>	02H	Start of TeXt character
F	Command Code	“G”	“G” is returned by the sign
G	File Label	“2”	"2“ (32H) is the File Label of the STRING file accessed
H	STRING File Data	“8,234,000”	The actual data in the STRING file
I	<ETX>	03H	End of TeXt character
J	Checksum	“0237”	Four ASCII digits that represent a 16-bit hexadecimal summation of all transmitted data from the previous <STX> through the previous <ETX> inclusive. The most significant digit is first.
K	<EOT>	04H	End Of Transmission character

4.7.10 DOTS PICTURE file examples (page 30)

4.7.10.1 Write DOTS PICTURE file example (page 30)

The following would write a DOTS PICTURE file labeled “A”, 15 pixel rows high x 9 pixel columns wide to a 4160C sign:

Table 56: Write DOTS PICTURE file example



Item	Name			Value	Description
A	<NUL>			00H	These five <NUL>s cause a sign to lock onto a baud rate. (This is also called “autobauding”.)
B	<SOH>			01H	Start Of Header character
C	Type Code			“b”	This means that this transmission is directed to all 4160C signs.
D	Sign Address			“00”	This means all 4160C signs on the network should “listen” to this transmission.
E	<STX>			02H	Start of TeXt character
F	Command Code			“I”	This is the “Write STRING file” Command Code. (See Table 16, “Write STRING file transmission frame format,” on page 28.)
G	File Label			“A”	File Label of the STRING file
H	Data Field	DOTS PICTURE File Data	Height (y)	“0F”	“0F” (15D) = pixel height of graphic
			Width (x)	“09”	“09” (9D) = pixel width of graphic
			Row Bit Pattern	“000000000“<CR> “000000000“<CR> “000100000“<CR> “000110000“<CR> “000111000“<CR> “000111100“<CR> “111111110“<CR> “111111112“<CR> “111111110“<CR> “000111100“<CR> “000111000“<CR> “000110000“<CR> “000100000“<CR> “000000000“<CR> “000000000“<CR>	Each row of the graphic is followed by a <CR> (0DH). “0” = sign pixel off “1” = sign pixel on - red “2” = sign pixel on - green
I	<EOT>			04H	End Of Transmission character

4.8 Appendix H: Modes

Modes are ways of displaying information on a sign. For example, the ROTATE Mode makes text or graphics travel from right to left on a sign.

4.8.1 Standard Modes

When a Standard Mode Code of "n" (6EH) is given (see **Table 57**), the following Special Modes (**Table 58**) or Special Graphics (**Table 59**) can be designated in the Special Specifier field (see "TEXT files" on page 11).

Table 57: Standard Modes

Mode name	ASCII code	Hex code	Description
ROTATE	"a"	61H	Message travels right to left.
HOLD	"b"	62H	Message remains stationary.
FLASH	"c"	63H	Message remains stationary and flashes.
reserved	"d"	64H	
ROLL UP	"e"	65H	Previous message is pushed up by a new message.
ROLL DOWN	"f"	66H	Previous message is pushed down by a new message.
ROLL LEFT	"g"	67H	Previous message is pushed left by a new message.
ROLL RIGHT	"h"	68H	Previous message is pushed right by a new message.
WIPE UP	"i"	69H	New message is wiped over the previous message from bottom to top.
WIPE DOWN	"j"	6AH	New message is wiped over the previous message from top to bottom.
WIPE LEFT	"k"	6BH	New message is wiped over the previous message from right to left.
WIPE RIGHT	"l"	6CH	New message is wiped over the previous message from left to right.
SCROLL	"m"	6DH	New message line pushes the bottom line to the top line if 2-line sign.
AUTOMODE	"o"	6FH	Various Modes are called upon to display the message automatically.
ROLL IN	"p"	70H	Previous message is pushed toward the center of the display by the new message.
ROLL OUT	"q"	71H	Previous message is pushed outward from the center by the new message.
WIPE IN	"r"	72H	New message is wiped over the previous message in an inward motion.
WIPE OUT	"s"	73H	New message is wiped over the previous message in an outward motion.
COMPRESSED ROTATE	"t"	74H	Message travels right to left. Characters are appoximately one half their normal width. (Only available on certain sign models.)
SPECIAL	"n"	6EH	This is followed by a Special Specifier ASCII character which defines one of the Special Modes. See "Special Modes" on page 66.

4.8.2 Special Modes

Table 58: Special Modes

Mode name	ASCII code	Hex code	Description
TWINKLE	"0"	30H	Message will twinkle on the sign.
SPARKLE	"1"	31H	New message will sparkle over the current message.
SNOW	"2"	32H	Message will "snow" onto the display.
INTERLOCK	"3"	33H	New message will interlock over the current message in alternating rows of dots from each end.
SWITCH	"4"	34H	Alternating characters "switch" off the sign up and down. New message "switches" on in a similar manner.
SLIDE	"5"	35H	New message slides onto the sign one character at a time from right to left.
SPRAY	"6"	36H	New message sprays across and onto the sign from right to left.
STARBURST	"7"	37H	"Starbursts" explode the new message onto the sign.

Table 58: Special Modes

Mode name	ASCII code	Hex code	Description
WELCOME	“8”	38H	The word “Welcome” is written in script across the sign.
SLOT MACHINE	“9”	39H	Slot machine symbols appear randomly across the sign.

4.8.3 Special Graphics

Table 59: Special Graphics

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

4.9 Appendix I: ALPHA protocol ASCII table

SPECIAL NOTE

This is a variation of the standard ASCII character set that is used in the EZ KEY II / EZ95 protocol.

Variations from standard ISO ASCII are shaded.

	Character		Hex	Dec		Character		Hex	Dec
Control characters	^@	NULL	00	0	Uppercase letters	@		40	64
	^A	SOH	01	1		A		41	65
	^B	STX	02	2		B		42	66
	^C	ETX	03	3		C		43	67
	^D	EOT	04	4		D		44	68
	^E	Double high characters (2-byte format) • 05H + "0" (30H) = Double height off (default) • 05H + "1" (31H) = Double height on	05	5		E		45	69
	^F	True descenders (2-byte format) • 06H + "0" (30H) = True descenders off (default) • 06H + "1" (31H) = True descenders on	06	6		F		46	70
	^G	Character flash (2-byte format) • 07H + "0" (30H) = Character flash off (default) • 07H + "1" (31H) = Character flash on	07	7		G		47	71
	^H	Extended character sets (2-byte format) • 08H + Extended character (see the following "Extended character set")	08	8		H		48	72
	^I	No Hold speed — when used, there will be virtually no pause following the mode presentation. This is not applicable for the Rotate or Compressed Rotate modes.	09	9		I		49	73
	^J		0A	10		J		4A	74
	^K	Call date (2-byte format) — the date will be displayed, where DD = date, MM = month, YY = year, MMM = month abbreviation, and YYYY = year: • 0BH + "0" (30H) = MM/DD/YY • 0BH + "1" (31H) = DD/MM/YY • 0BH + "2" (32H) = MM-DD-YY • 0BH + "3" (33H) = DD-MM-YY • 0BH + "4" (34H) = MM.DD.YY • 0BH + "5" (35H) = DD.MM.YY • 0BH + "6" (36H) = MM DD YY • 0BH + "7" (37H) = DD MM YY • 0BH + "8" (38H) = MMM.DD, YYYY • 0BH + "9" (39H) = Day of week	0B	11		K		4B	75
	^L	New page — start of next display page	0C	12		L		4C	76
	^M	New line — start of new line	0D	13		M		4D	77
	^N		0E	14		N		4E	78
	^O		0F	15		O		4F	79
	^P	Call STRING file (2-byte format) — must be followed by a STRING File Label.	10	16		P		50	80
	^Q	Disable wide characters	11	17		Q		51	81
	^R	Enable wide characters	12	18		R		52	82
	^S	Call Time — time of day will be called up.	13	19		S		53	83
	^T	Call DOTS PICTURE file (2-byte format) — must be followed by a DOTS PICTURE File Label.	14	20		T		54	84
	^U	Speed 1 (slowest)	15	21		U		55	85
	^V	Speed 2	16	22		V		56	86
	^W	Speed 3	17	23		W		57	87
	^X	Speed 4	18	24		X		58	88
	^Y	Speed 5 (fastest)	19	25		Y		59	89
	^Z	Select character set (2-byte format) • 1AH + "1" (31H) = Five high standard • 1AH + "3" (33H) = Seven high standard • 1AH + "5" (35H) = Seven high fancy • 1AH + "6" (36H) = Ten high standard • 1AH + "8" (38H) = Full height fancy • 1AH + "9" (39H) = Full height standard	1A	26		Z		5A	90
	^[	Start of Mode field	1B	27		[		5B	91

	Character		Hex	Dec
Control characters (continued)	^\	Select character color (2-byte format) — some signs do not support all the following colors: • 1CH + “1” (31H) = Red • 1CH + “2” (32H) = Green • 1CH + “3” (33H) = Amber • 1CH + “4” (34H) = Dim red • 1CH + “5” (35H) = Dim green • 1CH + “6” (36H) = Brown • 1CH + “7” (37H) = Orange • 1CH + “8” (38H) = Yellow • 1CH + “9” (39H) = Rainbow 1 • 1CH + “A” (41H) = Rainbow 2 • 1CH + “B” (42H) = Color mix • 1CH + “C” (43H) = Autocolor	1C	28
	^]	Select character attribute (3-byte format) — 1st byte is control code; 2nd byte is the attribute; and 3rd byte specifies either ON [“1” (31H)] or OFF [“0” (30H)]. OFF is the default setting for all of the following: • 1DH + “0” (30H) + “1” or ”0” = Wide ON or OFF • 1DH + “1” (31H) + “1” or ”0” = Double wide ON or OFF • 1DH + “2” (32H) + “1” or ”0” = Double high ON or OFF • 1DH + “3” (33H) + “1” or ”0” = True descenders ON or OFF • 1DH + “4” (34H) + “1” or ”0” = Fixed width ON or OFF • 1DH + “5” (35H) + “1” or ”0” = Fancy ON or OFF	1D	29
	^^	Select character spacing (2-byte format) • 1EH + “0” (30H) = Proportional characters (default) • 1EH + “1” (31H) = Fixed width left justified characters	1E	30
	^_	Call ALPHAVISION DOTS PICTURE file (15-byte format): 1FH + SFFFFFFFFFtttt where • S = “C” (43H) if the file is running as part of a Quick Flick animation. The display is cleared before each ALPHAVISION DOTS PICTURE is shown. S = “L” (4CH) if the file running is 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 used instead. • FFFFFFFFF (9 bytes) = file name. If the file name is less than 9 characters, spaces (20H) should precede the file name, so that the total number of characters is always fixed at 9. • tttt (4 bytes) — display hold time. A 4-digit ASCII hex number indicating tenths of seconds. Leading 0’s are ignored. For example, “0020” = 32 tenths of seconds = 3.2 seconds.	1F	31

	Character	Hex	Dec
Uppercase letters (continued)	\	5C	92
	]	5D	93
	¢	5E	94
	_	5F	95

	Character		Hex	Dec		Character		Hex	Dec
Special characters and numbers		space	20	32	Lowercase letters	`		60	96
	!		21	33		a		61	97
	“		22	34		b		62	98
	#		23	35		c		63	99
	$		24	36		d		64	100
	%		25	37		e		65	101
	&		26	38		f		66	102
	‘		27	39		g		67	103
	(		28	40		h		68	104
	)		29	41		i		69	105
	*		2A	42		j		6A	106
	+		2B	43		k		6B	107
	,		2C	44		l		6C	108
	-		2D	45		m		6D	109
	.		2E	46		n		6E	110
	/		2F	47		o		6F	111
	0		30	48		p		70	112
	1		31	49		q		71	113
	2		32	50		r		72	114
	3		33	51		s		73	115
	4		34	52		t		74	116
	5		35	53		u		75	117
	6		36	54		v		76	118
	7		37	55		w		77	119
	8		38	56		x		78	120
	9		39	57		y		79	121
	:		3A	58		z		7A	122
	;		3B	69		{		7B	123
	<		3C	60		l		7C	124
	=		3D	61		}		7D	125
	>		3E	62		1/2 space		7E	126
	?		3F	63		block		7F	127

Character & Control Code representation			Hex	Dec
Extended character set	Ç	08H + 20H	80H	128
	ü	08H + 21H	81H	129
	é	08H + 22H	82H	130
	â	08H + 23H	83H	131
	ä	08H + 24H	84H	132
	à	08H + 25H	85H	133
	å	08H + 26H	86H	134
	ç	08H + 27H	87H	135
	ê	08H + 28H	88H	136
	ë	08H + 29H	89H	137
	è	08H + 2AH	8AH	138
	ï	08H + 2BH	8BH	139
	î	08H + 2CH	8CH	140
	ì	08H + 2DH	8DH	141
	Ä	08H + 2EH	8EH	142
	Å	08H + 2FH	8FH	143
	É	08H + 30H	90H	144
	æ	08H + 31H	91H	145
	Æ	08H + 32H	92H	146
	ô	08H + 33H	93H	147
	ö	08H + 34H	94H	148
	ò	08H + 35H	95H	149
	û	08H + 36H	96H	150
	ù	08H + 37H	97H	151
	ÿ	08H + 38H	98H	152
	Ö	08H + 39H	99H	153
	Ü	08H + 3AH	9AH	154
	¢	08H + 3BH	9BH	155
	£	08H + 3CH	9CH	156
	¥	08H + 3DH	9DH	157
	%	08H + 3EH	9EH	158
	ƒ	08H + 3FH	9FH	159
	á	08H + 40H	A0H	160
Temperature		08H + "^\" (1CH)	NOTE 2	
		08H + "^]" (1DH)	NOTE 3	

Character & Control Code representation			Hex	Dec
Extended character set	í	08H + 41H	A1H	161
	ó	08H + 42H	A2H	162
	ú	08H + 43H	A3H	163
	ñ	08H + 44H	A4H	164
	Ñ	08H + 45H	A5H	165
	ª	08H + 46H	A6H	166
	º	08H + 47H	A7H	167
	¿	08H + 48H	A8H	168
	°	08H + 49H	A9H	169
	¡	08H + 4AH	AAH	170
	NOTE 1	08H + 4BH	ABH	171
	θ	08H + 4CH	ACH	172
	Θ	08H + 4DH	ADH	173
	c´	08H + 4EH	AEH	174
	C´	08H + 4FH	AFH	175
	c	08H + 50H	BOH	176
	C	08H + 51H	B1H	177
	d	08H + 52H	B2H	178
	D	08H + 53H	B3H	179
	s	08H + 54H	B4H	180
	z	08H + 55H	B5H	181
	Z	08H + 56H	B6H	182
	ß	08H + 57H	B7H	183
	S	08H + 58H	B8H	184
	ß	08H + 59H	B9H	185
	Á	08H + 5AH	BAH	186
	À	08H + 5BH	BBH	187
	Á´	08H + 5CH	BCH	188
	á´	08H + 5DH	BDH	189
	É	08H + 5EH	BEH	190
	Í	08H + 5FH	BFH	191
	Õ	08H + 60H	C0H	192
	õ	08H + 61H	C1H	193
Counters		08H + "z" (7AH)	NOTE 4	
		08H + "{" (7BH)	NOTE 5	
		08H + "|" (7CH)	NOTE 6	
		08H + "}" (7DH)	NOTE 7	
		08H + "~" (7EH)	NOTE 8	

NOTE: 1. A single column space.
2. Displays temperature in Celsius (only on Solar, 790i, 460i, 440i, and 430i)
3. Displays temperature in Fahrenheit (only on Solar, 790i, 460i, 440i, and 430i)
4. Displays the current value in Counter 1
5. Displays the current value in Counter 2
6. Displays the current value in Counter 3
7. Displays the current value in Counter 4
8. Displays the current value in Counter 5

4.10 Appendix J: ISO ASCII table

SPECIAL NOTE

This is the standard ASCII character set.

This is the standard ASCII character set:

	Character			Hex	Dec			Character		Hex	Dec
Control characters	NULL	^@	null	00	0		Uppercase letters	@		40	64
	SOH	^A	start of heading	01	1			A		41	65
	STX	^B	start of text	02	2			B		42	66
	ETX	^C	end of text	03	3			C		43	67
	EOT	^D	end of transmission	04	4			D		44	68
	ENQ	^E	enquiry	05	5			E		45	69
	ACK	^F	acknowledge	06	6			F		46	70
	BEL	^G	bell	07	7			G		47	71
	BS	^H	backspace	08	8			H		48	72
	HT	^I	horizontal tab	09	9			I		49	73
	LF, NL	^J	line feed, new line	0A	10			J		4A	74
	VT	^K	vertical tab	0B	11			K		4B	75
	FF, NP	^L	form feed, new page	0C	12			L		4C	76
	CR	^M	carriage return	0D	13			M		4D	77
	SO	^N	shift out	0E	14			N		4E	78
	SI	^O	shift in	0F	15			O		4F	79
	DLE	^P	data link escape	10	16			P		50	80
	DC1	^Q	device control 1	11	17			Q		51	81
	DC2	^R	device control 2	12	18			R		52	82
	DC3	^S	device control 3	13	19			S		53	83
	DC4	^T	device control 4	14	20			T		54	84
	NAK	^U	negative acknowledge	15	21			U		55	85
	SYN	^V	synchronous idle	16	22			V		56	86
	ETB	^W	end of transmission block	17	23			W		57	87
	CAN	^X	cancel	18	24			X		58	88
	EM	^Y	end of medium	19	25			Y		59	89
	SUB	^Z	substitute	1A	26			Z		5A	90
	ESC	^[	escape	1B	27			[		5B	91
	FS	^\	file seperator	1C	28			\		5C	92
	GS	^]	group seperator	1D	29			]		5D	93
	RS	^^	record seperator	1E	30			^		5E	94
	US	^_	unit seperator	1F	31			_		5F	95
Special characters and numbers			space	20	32		Lowercase letters	`		60	96
		!		21	33			a		61	97
		"		22	34			b		62	98
		#		23	35			c		63	99
		$		24	36			d		64	100
		%		25	37			e		65	101
		&		26	38			f		66	102
		'		27	39			g		67	103
		(		28	40			h		68	104
		)		29	41			i		69	105
		*		2A	42			j		6A	106
		+		2B	43			k		6B	107
		,		2C	44			l		6C	108
		-		2D	45			m		6D	109
		.		2E	46			n		6E	110
		/		2F	47			o		6F	111
		0		30	48			p		70	112
		1		31	49			q		71	113
		2		32	50			r		72	114
		3		33	51			s		73	115
		4		34	52			t		74	116
		5		35	53			u		75	117
		6		36	54			v		76	118
		7		37	55			w		77	119
		8		38	56			x		78	120
		9		39	57			y		79	121
		:		3A	58			z		7A	122
		;		3B	69			{		7B	123
		<		3C	60			|		7C	124
		=		3D	61			}		7D	125
		>		3E	62			~		7E	126
		?		3F	63			DEL		7F	127

4.11 Appendix K: Modes, fonts, colors, and display options available on signs

4.11.1 Modes available on signs

Table 60: Modes available on signs

Sign (FM = Full Matrix, CM = Character Matrix, LM = Line Matrix)		Modes																
								Rotate										
ALPHA sign	Type	Automode	Bulletin	Flash	Hold	Interlock	Roll	Standard	Condensed	Scroll	Slide	Snow	Sparkle	Spray	Starburst	Switch	Twinkle	Wipe
210C	FM	●		●	●		●	●	●	●		●	●	●	●	●	●	●
215	FM	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
220C	FM	●		●	●		●	●	●	●		●	●	●	●	●	●	●
Series 300	FM	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Series 4000	FM	●		●	●	●	●	●		●	●	●	●	●	●	●	●	●
Series 7000	FM	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●
Big Dot	FM	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Alphavision FM	FM	●	●	●	●	●	●	●		●			●				●	●
Alphavision CM	CM	●		●	●													●
790i	FM	●		●	●	●	●	●		●	●	●	●	●	●	●	●	●
Solar	FM	●		●	●	●	●	●		●			●				●	●
Director	CM	●		●	●													●
2.1-inch CM	CM	●		●	●													●
3.2-inch CM	CM	●		●	●													●
PPD	LM	●		●	●	●	●	●		●	●	●	●	●	●	●	●	●
BetaBrite	FM	●		●	●		●	●	●	●		●	●	●	●	●	●	●

4.11.2 Fonts and colors available on signs

Table 61: Fonts and colors available on signs.

Sign (FM = Full Matrix, CM = Character Matrix, LM = Line Matrix)		Fonts & Color													
ALPHA sign	Type	15/16 Row Normal	15/16 Row Fancy	Ten Row	Seven Row Normal	Seven Row Fancy	Five Row	Color (see NOTE)	Normal	Wide	Double Wide	Flashing	Double High	True Descenders	Fixed Width
210C	FM				●	●	●	●	●	●	●	●			●
215	FM				●	●	●	●	●	●	●	●			●
220C	FM				●	●	●	●	●	●	●	●			●
Series 300	FM				●	●	●	●	●	●	●				●
Series 4000	FM	●	●		●	●	●	●	●	●	●	●			●
Series 7000	FM	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Big Dot	FM				●	●	●	●	●	●	●				●
Alphavision FM	FM	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Alphavision CM	CM				●		●	●	●			●			
790i	FM				●		●		●	●	●				●
Solar	FM	●	●		●	●	●	●	●	●	●				●
Director	CM				●		●	●	●			●			
2.1-inch CM	CM				●		●	●	●			●			

Table 61: Fonts and colors available on signs.

Sign (FM = Full Matrix, CM = Character Matrix, LM = Line Matrix)		Fonts & Color													
ALPHA sign	Type	15/16 Row Normal	15/16 Row Fancy	Ten Row	Seven Row Normal	Seven Row Fancy	Five Row	Color (see NOTE)	Normal	Wide	Double Wide	Flashing	Double High	True Descenders	Fixed Width
3.2-inch CM	CM				●		●	●	●			●			
PPD	LM				●	●	●		●	●	●				●
BetaBrite	FM				●	●	●	●	●	●	●	●			●

NOTE: Sign names ending in “C” or “T”, such as 4120C, have color capabilities. Sign names ending in “R”, such as 4120R, can display in red only.

4.11.3 Display options available on signs

Table 62: Options available on signs.

Sign (FM = Full Matrix, CM = Character Matrix, LM = Line Matrix)		Options							
ALPHA sign	Type	Time	Date	Variables	Temperature	Speed	New Line	New Page	Animation
Series 200	FM	●	●	●		●	●		●
Series 300	FM	●	●	●		●	●		●
Series 4000	FM	●	●	●		●	●		●
Series 7000	FM	●	●	●		●	●	●	
Big Dot	FM	●	●	●		●	●		●
Alphavision FM	FM	●	●	●		●	●	●	
Alphavision CM	CM	●	●	●		●	●	●	
790i	FM	●		●	●	●	●		●
Solar	FM	●	●	●	●	●	●		●
Director	CM	●	●	●		●	●	●	
2.1-inch CM	CM	●	●	●		●	●	●	
3.2-inch CM	CM	●	●	●		●	●	●	
PPD	LM	●	●	●		●	●		●