

# **Enhanced Version Advanced Display Protocol (ADP)**

**Users Manual**

Information in this document is subject to change without notice. Companies, names and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without written permission of Electro-Matic Products, Inc.

Copyright © 2011 Electro-Matic Products, Inc. All rights reserved.

V4.0  
WI-0322  
04.20.11 gp&tep/cg

<b>1.0 Introduction</b>	5
<b>2.0 Settings</b>	6
2.1 Initial Power-Up	6
2.2 Communication Ports	6
2.3 Factory Defaults	6
<b>3.0 Document Information</b>	8
<b>4.0 Protocol Overview</b>	9
4.1 Definition	9
4.2 Limitation	9
4.3 Transmission Packet Formats	9
<b>5.0 ADP File Structure</b>	11
5.1 Simple File	11
5.2 Expanded File	11
<b>6.0 Command Codes</b>	13
6.1 Writing Files	13
6.1.1 Simple TEXT Files	13
6.1.2 Simple STRING Files	14
6.1.3 Expanded (" <i>&lt;0X0F&gt;</i> ") TEXT Files (" <i>T</i> ")	15
6.1.4 Expanded (" <i>&lt;0X0F&gt;</i> ") STRING Files (" <i>S</i> ")	16
6.1.5 Simple and Expanded Calls from STRING (" <i>S</i> ") File to TEXT Files (" <i>T</i> ")	17
6.1.6 Nesting for STRING File	18
6.2 Customizing Defaults	18
6.2.1 Default Instructions E# Command Code	18
6.3 Start Mode Commands <i>&lt;0X1B&gt;</i>	22
6.3.1 Vertical Positioning Command Code	22
6.3.2 Character In and Out Command Codes	23
6.4 Controlled Character Display (Special Functions)	24
6.4.1 Character Flash Command Code <i>&lt;07 Hex&gt; &lt;0X07&gt;</i>	24
6.4.2 Display Date and Time Stamp Command Codes <i>&lt;0X0B&gt;</i>	25
6.4.3 Character Frame New Page Command Code <i>&lt;0x0C&gt;</i>	26
6.4.4 Character Line New Line- Word Wrap Command Code <i>&lt;0X0D&gt;</i>	26
6.4.5 Character Horizontal Justification Command Code <i>&lt;0X0A&gt;</i>	26
6.4.6 Character Foreground Color Command Code <i>&lt;0X1C&gt;</i>	26
6.4.7 Character Background Color Command Code <i>&lt;0X1D&gt;</i>	27
6.4.8 Character Speed Command Code	27
6.4.9 Character Font Command Code <i>&lt;0X1A&gt;</i>	27
6.4.10 Character Hold (Stay) Command Code <i>&lt;0X0E&gt;</i>	28

6.4.11 *Play List or Run Sequence*..... 28

6.4.12 *Deletion of Playlist or Run Sequence*..... 29

6.4.13 *Embed Time of Day into Messages* ..... 29

6.4.14 *Embed Date into Messages*..... 30

6.4.15 *Embed String Files in Text* ..... 30

7.0 **Examples of Coding** ..... 34

**Appendix A: Updating an Enhanced 4U2SEE with Firmware 8E04**..... 36

**Appendix B: Delete Sequence**..... 42

**Appendix C: Expanded & Single Valid File Labels** ..... 43

**Appendix D: Examples of FlexSender Code** ..... 44

**Appendix E: Display Mode Descriptions** ..... 49

**Appendix F: Hex-Character Summary**..... 50

**Changelog** ..... 51

## 1.0 Introduction

The 4U2SEE signs are designed to display text messages, graphics, numeric quantities and other information.

This document explains the ASCII (American Standard Code for Information Interchange) communication protocol utilized to communicate with 4U2SEE display boards manufactured by Electro-Matic Products, Inc. with relevance given to Electro-Matic input devices.

The 4U2SEE display is manufactured in two styles:

1. Standard – this version is NOT discussed in this manual and does not support ADP protocol.
2. Enhanced – this version supports ADP protocol and all features detailed in this manual.

There are two (2) methods of communications associated with the 4U2SEE display boards, Serial and Ethernet. The Standard version is limited to Serial communications, while the Enhanced version allows both Serial and Ethernet.

The typical internal file configuration of a 4U2SEE is formatted as: X:\Y\Z

Where the "X:" is the Drive (detailed information given below).

Where the "Y" is the Folder (detailed information given below).

Where the "Z" is the File Name (referenced in Appendix C – "Valid File Labels and Characters" and throughout this manual).

Valid Drive ("X") names are as follows:

- "C" - This is the system drive where default fonts, settings and colors are stored. Currently, this is not accessible with the remote control, but other Electro-Matic communication devices can be utilized.
- "D" - This is flash memory (non-volatile). This drive is used to store TEXT, STRINGS and PICTURES files that need to be remembered after power is removed and restored (retentive memory). Caution must be exercised with this drive. There is a 100,000 file write limitation to this memory. Excessive storage of message files to this drive (more than 100, 000) will cause the memory to fail and render this portion of the display inoperable.
- "E" - This is RAM memory (volatile). This drive is used to store TEXT, STRINGS and PICTURES files that can be lost when power is removed and restored. This drive should be used for TEXT, STRINGS and PICTURES files that are changed frequently. There is no limitation on the number of writes to this drive.

Valid Folder ("Y") names are as follows:

- "P" - This is a picture folder and all files must be in .bmp format in either 2 color, 16 color, 256 color or true color resolution. "P" (PICTURE) folder files can only be displayed by being called from a "T" folder (TEXT) file. (Future capability not discussed in this manual)
- "S" - This is a STRING file folder for variable STRING messages. "S" (STRING) folder files can only be displayed by being called from a "T" (TEXT) folder file.
- "T" - This is a TEXT folder for message files. All main messages to be displayed are stored in this folder.

### 1.1 File Format Introduction

## 2.0 Settings

Within this document are the ASCII commands required to send messages to a 4U2SEE sign via the Ethernet, RS232 or RS485 ports. Alternate devices and software can be used. Caution must be used to make sure the display's port settings and the sending devices settings match, otherwise the devices will not communicate. This document covers 4U2SEE signs with: Firmware Version 8E04 or later versions.

### 2.1 Initial Power-Up

The appropriate settings affiliated with each display board can be viewed with the introduction of electrical power. This information is contained on the 'C' drive (referenced below in section 2.3). The power-up sequence is as follows:

- 1) The IP address – see note below
- 2) The physical size of the display – width and height
- 3) The power setting – selection of on/off enabling or disabling
- 4) The time zone setting – Greenwich Mean Time (GMT)
- 5) The date and time setting – date, time and day of the week
- 6) The firmware version – firmware version and the group and unit numbers (GGUU – referenced in Tables: 2 and 7 ["Sign Address"]).

The previously mentioned alternative input devices for Electro-Matic Products, Inc are the hand held IR remote control, FlexSender software and Sigma 3000 software.

#### PLEASE NOTE:

- 1) The IP (Internet Protocol) address is a set of numbers, separated by periods, that is assigned to every network device.
- 2) The IP address allows you to communicate and control activities on the network to a specific device.
- 3) The IP address of your 'Host' computer must be in the same subnet mask range as your display. Example: If your display's IP address is 192.168.3.250 and the subnet mask for your 'Host' computer is 255.255.255.0, then your 'Host' computer's IP address must be 192.168.3.xx. Where 'xx' can be any number between 1 to 254, except 250 – because it is used by your display. maintenance.
- 4) All above power-up items should be documented and archived for future reference.
- 5) To install or update a Firmware Version (Item 6 in Section 2.1) reference Appendix A: Updating an Enhanced 4U2SEE with Firmware 8E04.

### 2.2 Communication Ports

There are three accessible communication ports available for each enhanced 4U2SEE display sign. The previously mentioned (section 1.0) ethernet port is defined for network devices. If a token ring network is not available, a cross-over cable (pins switched to emulate a network connection) can be used in the Ethernet port to supply proper communications from your 'Host' (transmission) device to your display board. The Serial ports (RS232 and RS485) can have direct non-networked access to the display board. The RS232 cable is for direct communications to the RS232 (gray-connector) port from your 'Host' device. The use of the RS485 (black-connector) port is for linking (daisy chaining) multiple displays and can be utilized by your "Host" device with the use of a converter.

### 2.3 Factory Defaults

When display boards are manufactured and shipped, they are preset with default values. These default values are:

Font color – RED  
 Background color – BLACK  
 Font size – NORMAL 7  
 Line space – 1 PIXEL  
 Stay time – 3 SECONDS  
 Speed – MEDIUM  
 Horizontal justification – CENTER  
 Vertical justification – FILL  
 Wordwrap – ON  
 In Mode- Random  
 Out Mode- Random  
 Flash- OFF  
 Default drive- D:  
 Folder- T:  
 Flash- OFF  
 Default Drive- D:

**PLEASE NOTE:**

- 1) A factory default for Flash is given, but this command can NOT have it's default setting altered with the use of a special function command ("E" – explained in section 6.2)).
- 2) All standard transmission coding conventions are explained in table 2 and throughout this manual.
- 3) All factory defaults can be altered unless otherwise noted.
- 4) When the TEXT is wrapping from one line to the next, the display board will go back from the end of the first line of TEXT and search for the first carriage return space. The display board will wrap from this point so no whole words are broken. If a carriage return is not found then the sign will wrap from the last character fitting on that line.

### 3.0 Document Information

3.1

Document Conventions

Table 1: Document Conventions

Convention	Description
<0x01>	ASCII control character abbreviation
“A”	ASCII character ( in this case the letter A )



## 4.0 Protocol Overview

For proper operation of the 4U2SEE display board, a set of communication rules must be followed, this is a simple definition of protocol. ASCII must be utilized, to communicate in Electro-Matic's ADP format.

Electro-Matic Products has set a restriction regarding the speed of the messages, graphics, numeric or other information sent to the 4U2SEE display. The maximum updating data rate for inputted memorandums is approximately two (2) seconds.

### PLEASE NOTE:

- 1) If data is issued to the display at intervals quicker than the approximate two (2) seconds, the display may not get updated.
- 2) To send messages to the 4U2SEE displays, the run sequence or play list may need to be edited or cleared. To accomplish this via the Electro-Matic input devices, please refer to Appendix B: Delete File Playlist or Sequence or to set or delete a current play list or run sequence please refer to section 6.4.11-12.
- 3) The construction of run sequences or play lists can be referenced in section 6.4.11 and is on a first-in first-out (F.I.F.O.) format.

Messages in this protocol can be transmitted to a sign in the following basic formats:

1. Standard (Figure 1)

**Figure 1: Standard Transmission Packet**

<0x01>	Address Pre-filler	Sign Address	<0x02>	Command Code	Data Field	<0x04>
--------	-----------------------	-----------------	--------	-----------------	---------------	--------

Non-printable control characters like <0x01> are used in the packet:

**Table 2: Standard Transmission Packet Format**

	<0x01>	Address Pre-filler	Sign Address	<0x02>	Command Code	File Label	Data Field	<0x04>
	A	B	C	D	E	F	G	H
Item	Name		Description					
A	<0x01>		The <0x01> is the "Start of Header" ASCII character. Example: <0x01>Z00<0x02>AA This is where you type message <0x04>					
B	Address Pre-filler		Reserved, always an ASCII character (reference Appendix C: Valid File Labels & Characters). All examples throughout this manual use the ASCII character "Z" (this is not required). Example: <0x01>Z00<0x02>A6 This is where you type message <0x04>					
C	Sign Address		The identifier or "address" of the sign represented by two to four ASCII digits as a number between "00" and "99". Address "00" is reserved as a broadcast address. The first two digits represent the group number and the next two digits represent this unit number. (GGUU) When only 2 digits are used in the sign address in this field then the group address is assumed to be 00 and the 2 digits represent the unit number (UU). This change is typical throughout the manual (changing to 00 after the Z) <b>PLEASE NOTE:</b> The unit number (UU) of your display board can be changed with Electro-Matic's input device Sigma 3000 software. Example: <0x01>Z00<0x02>Ac This is where you type message <0x04>					
D	<0x02>		"Start of Text" (ASCII) character. <0x02> always precedes a Command Code. Example: <0x01>Z00<0x02>AA This is where you type message <0x04>					

### 4.1 Definition

### 4.2 Limitation

### 4.3 Transmission Packet Formats

E	Command Code	ASCII character(s) that defines the transmission and data PLEASE NOTE: For examples on TEXT and STRING files refer to sections 6.
F	File Label	ASCII character(s) that indicates the TEXT file being accessed. See "Appendix C: Expanded & Single Valid File Labels & Characters" <b>PLEASE NOTE:</b> For examples on TEXT and STRING files refer to section 6.
G	Data Field	Made up of ASCII characters. The Data Field format is dependent on the preceding Command Code. Example: <0x01>Z00<0x02>Aq <b>This is where you type message</b> <0x04>
H	<0x04>	"End of Transmission"(ASCII) character Example: <0x01>Z00<0x02>A9 This is where you type message <b>&lt;0x04&gt;</b>

**PLEASE NOTE:** In the examples above no carriage return spaces need to be between the "File Label" ("F" in Table 2) and the beginning of your message or between the end of your message and the "End of Transmission" ("H" in Table 2). Also all bold text in the above ASCII code examples are supplied to aid each circumstance and all non-bolded format text is explained in Table 2.

## 5.0 ADP File Structure

There are two, Simple and Expanded, format conventions for addressing ASCII files for the 4U2SEE display board. Each File format and examples are shown below:

The Simple File addressing convention uses a single “Z” file character on the default drive and in the default folder.

### 5.1 Simple File

The format is: X:\YZ

Where “X:” = the default Drive

Where “Y” = the default Folder

Where “Z” = File Name (reference Appendix C Expanded & Single Valid File Labels and Characters)

- 1) <0x01>Z00<0x02>**AA** Simple Text Write File <0x04>
- 2) <0x01>Z00<0x02>**A\$** Simple Text Write File <0x04>

The above examples illustrate the format for Simple Files. All non-bold coding is explained in table 2. Both examples will be stored on the default Drive in the default Folder. The first example with the bold “A” will designate “A” as the file name and the second example with the bold “\$” will designate “\$” as the file name.

#### PLEASE NOTE:

- 1) In the above examples no carriage return spaces need to be between the “File Label (F” in table 2) and the beginning of your message or between the end of your message and the “End of Transmission” (“H” in table 2).
- 2) There is a possibility of ninety-four (94) messages to be stored and displayed with this format.

The Expanded file addressing convention is a variance of the Simple File. This format will allow many more messages to be stored and displayed.

### 5.2 Expanded File

The format is: <0x0F>XFAA (Drive: X: Folder: F File: AA)

Where “X:” = Drive (see explanation)

Where “F” = Folder (see explanation)

Where “AA” = File Name [combination of any two valid file labels (reference Appendix C Valid File Labels and Characters)]

The Folder (“F”) can be comprised of a “P” (PICTURE folder – future capability not discussed in this manual), “S” (STRING folder) or “T” (TEXT folder – factory default)

The Drive (“X”) can be comprised of a “C” (system drive – for defaults ONLY), “D” (flash drive – factory default) or “E” (RAM drive – random access memory)

- 1) <0x01>Z00<0x02>**A<0x0F>**DTAB Expanded Text Write File <0x04>
- 2) <0x01>Z00<0x02>**A<0x0F>**ETQr Expanded Text Write File <0x04>

The above examples illustrate the format for Expanded Files. All non-bold coding is explained in table 2, the bold <0x0F> signifies Expanded Files. The first example, writes (“A” – before <0x0F>) to the “D” Drive and places File “AB” into the “T” Folder. The second example writes (“A” – before <0x0F>) to the “E” Drive and places File “Qr” into the “T” Folder.

#### PLEASE NOTE:

- 1) In the above examples no carriage return spaces need to be between the “AA” and the beginning of your message or between the end of your message and the “End of Transmission” (“H” in table 2).
- 2) There is a possibility of seventeen thousand six hundred and seventy-two (17,672) combination Expanded TEXT file labels (reference Appendix C: Expanded & Single Valid File Labels and Characters), which is calculated by multiplying valid File label combinations and then multiplying by the amount of valid Drives for the specific Folder ([94 x 94] x 2).
- 3) If an Expanded TEXT Write command (“A”) is used then the Expanded TEXT Folder must be a (“T”).

- 4) The expansion of TEXT files can not be accomplished with FlexSender prior to 4U2SEE firmware version 8E04.

## 6.0 Command Codes

A Command Code is used to determine whether information is read from or written to signs.

Figure 2: Comand Code location in the Standard transmission packet

<0x01>	Address Pre-filler	Sign Address	<0x02>	Command Code	Data Field	<0x04>
--------	-----------------------	-----------------	--------	-----------------	---------------	--------

In addition to determining whether information is written or read, the Command Code determines the contents of the Data Field in the protocol transmission packet formats.

Command Codes fall into four, general categories:

- TEXT file commands
- SPECIAL FUNCTION commands
- STRING file commands
- PICTURE file command (future capability not discussed in this manual)

The conventional formats for writing files to a 4U2SEE display board are given below.

6.1

Writing Files

### 6.1.1 Simple TEXT Files

The Simple TEXT files are used to store ASCII characters only (reference Appendix C: Expanded & Single Valid File Labels and Characters). The ASCII message data and display mode information, along with various other control codes, are stored in Simple TEXT files on the default Drive in the default Folder. On initial power-up the sign’s memory is configured with one Simple TEXT file (file label “A”). When writing to a TEXT file and the transmission is over the unit will begin displaying the last received TEXT file. The last received message is the current play list (sequence). Deleting and setting up play lists (sequences) is outlined in section 6.4.11-12.

Example of writing a Simple TEXT file format:  
<0x01>Z00<0x02>**A**a Writing a Simple TEXT file <0x04>

The bold “A” is the command (referenced in table 2) for the writing of a TEXT file (“a”) to be stored on the factory default drive (“D”) and in the default folder (“T”).

Table 3: Write TEXT File Transmission Packet with Default

Item	Name	Description
A	Command Code	“A” = Write TEXT file Example: <0x01>Z00<0x02> <b>A</b> # This is where you type message <0x04>
B	File Label	One ASCII character that indicates the TEXT file being accessed. See “Appendix C: Expanded & Single Valid File Labels & Characters” Example: <0x01>Z00<0x02> <b>A</b> % This is where you type message <0x04>

**PLEASE NOTE:**

- 1) TEXT files are the only files that can be directly shown on the 'LED' (light emitting diode) display.
- 2) In the above examples no carriage return spaces need to be between the "File Label" ("F" in table 2) and the beginning of your message or between the end of your message and the "End of Transmission" ("H" in table 2). Also all bold text in the above ASCII code are supplied to aid each circumstance and all non bolded format is explained in table 2.
- 3) There is a possibility of ninety-four (94) single Simple TEXT file labels (reference Appendix C: Expanded & Single Valid File Labels and Characters), located on the factory default Drive in the default Folder.

**6.1.2 Simple STRING Files**

The Simple STRING files are used to store ASCII characters only (reference Appendix C: Expanded & Single Valid File Labels and Characters). STRING files are used in applications where a group (string) of frequently changing data must be transmitted to and displayed by the sign. Applications include the storage of a number that changes often, such as temperature, a quantity or a timer. Applications can also include frequently changing faults, maintenance items or other industrial and residential formations.

**Table 4: Write STRING File Transmission Packet to Default Drive**

Item	Name	Description
A	Command Code	"G" = Write STRING file Example: <0x01>Z00<0x02> <b>GV</b> 500 <0x04>
B	File Label	One ASCII character that indicates the STRING file being sent. See "Appendix C: Valid File Labels & Characters". Example: <0x01>Z00<0x02> <b>GV</b> 600 <0x04>

**EXAMPLE NOTE:** The coding examples in "Table 4" get called from the TEXT, with embedded STRING, file: <0x01>Z00<0x02>AAGoal:<0x10>U Actual:<0x10>V<0x04> and when the example in Item "A" (Table 4) is sent to the display (populating "V" variable) – it will read:  
Goal: U Actual: 500

The "Goal" quantity has not yet been populated so "U" will stay the default – the last value sent. Then when the example in Item "B" (Table 4) is sent to the display (populating "V" variable) – it will read:  
Goal: U Actual: 600

The "Goal" quantity has not yet been populated so "U" will stay the default – the last value sent. If you populate "U" variable with: <0x01>Z00<0x02>GU 502 <0x04> and when this example is sent to the display (populating "U" variable) – it will read:  
Goal:502 Actual: 600

The "Actual" quantity from Item "B" will stay the default – the last value sent for "V".  
<0x01>Z00<0x02>GU 602 <0x04> and when this example is sent to the display (populating "U" variable) – it will read:  
Goal: 602 Actual: 600

Example of writing a Simple STRING file format:  
<0x01>Z00<0x02>**Gb** Writing a Simple STRING file <0x04>

The bold "G" signifies (referenced in table 2) the writing of a Simple STRING file ("b") to be stored on the factory default drive ("D") and in the default folder ("T").

**PLEASE NOTE:**

- 1) Simple STRING files must be added to TEXT files to be shown on the ‘LED’ display.
- 2) If Simple STRING Write command (“G”) is used then the STRING Folder must be default (“T”).
- 3) In the above examples no carriage return spaces need to be between the “File Label and the beginning of your message or between the end of your message and the “End of Transmission” (“H” in table 2). Also all bold text in the above ASCII code are supplied to aid each circumstance and all non bolded format is explained in table 2.
- 4) There is a possibility of ninety-four (94) single Simple STRING file labels (reference Appendix C: Expanded & Single Valid File Labels and Characters), located on the default Drive in the default Folder.
- 5) Calls (<0x10>) from Simple STRING files to TEXT files are explained in section 6.1.5.
- 6) Nested STRING Files (reference section 6.1.6) can consolidate (group together) Call(s) to STRING File(s). This command can not be utilized in 4U2SEE Firmware versions prior to 8E04.

**6.1.3 Expanded (“<0x0F”>) TEXT Files (“T”)**

The Expanded TEXT files are used to store ASCII characters only (reference Appendix C: Valid File Labels and Characters). The ASCII message data and display mode information, along with various other control codes, are stored in Expanded TEXT files on a specified Drive in the TEXT (“T”) Folder. When writing to a TEXT file and after the transmission is over the unit will begin displaying the last received TEXT file. The last received message is the current play list (sequence). Deleting and setting up play lists (sequences) is outlined in section 6.4.11-12.

**Table 5: Write Expanded TEXT File Transmission Packet**

Item	Name	Description
A	Command Code	“A” = Write TEXT file Examples: <0x01>Z00<0x02> <b>A</b> <0x0F>DTDJThis is where you type message<0x04> <0x01>Z00<0x02> <b>A</b> <0x0F>ETDJThis is where you type message<0x04>
B	Expansion	ASCII character(s) that indicate the TEXT file being accessed. See “Appendix C: Expanded & Single Valid File Labels & Characters” Examples: <0x01>Z00<0x02> <b>A&lt;0x0F&gt;</b> DTZxThis is where you type message<0x04> <0x01>Z00<0x02> <b>A&lt;0x0F&gt;</b> ETZxThis is where you type message<0x04>
C	Drive	Examples: <0x01>Z00<0x02> <b>A&lt;0x0F&gt;</b> <b>DT</b> 12This is where you type message<0x04> <0x01>Z00<0x02> <b>A&lt;0x0F&gt;</b> <b>ET</b> 12This is where you type message<0x04>
D	Folder	Examples: <0x01>Z00<0x02> <b>A&lt;0x0F&gt;</b> <b>ETX</b> xThis is where you type message<0x04> <0x01>Z00<0x02> <b>A&lt;0x0F&gt;</b> <b>DTX</b> xThis is where you type message<0x04>

E	File Label	Examples: <0x01>Z00<0x02>A<0x0F>DT <b>6R</b> This is where you type message<0x04> <0x01>Z00<0x02>A<0x0F>ET <b>6R</b> This is where you type message<0x04>
---	------------	---

**PLEASE NOTE:**

- 1) TEXT files are the only files that can be directly shown on the ‘LED’ display.
- 2) In the above examples no carriage return spaces need to be between the “File Label (“E” in table 5) and the beginning of your message or between the end of your message and the “End of Transmission” (“H” in table 2). Also all bold text in the above ASCII code are supplied to aid each circumstance and all non bolded format is explained in table 2.
- 3) There is a possibility of seventeen thousand six hundred and seventy-two (17,672) combination Expanded TEXT file labels (reference Appendix C - Expanded & Single Valid File Labels and Characters), which is calculated by multiplying valid File label combinations and then multiplying by the amount of valid Drives for the specific Folder ([94 x 94] x 2).
- 4) Valid Drives can be “D:” or “E:” (illustrated in the above table).
- 5) If an Expanded TEXT Write command (“A”) is used then the Expanded TEXT Folder must be a “T”)
- 6) The expansion of TEXT files can not be accomplished prior to 4U2SEE firmware version 8E04.

**6.1.4 Expanded (“<0x0F>”) STRING Files (“S”)**

The Expanded STRING files are used to store ASCII characters only (reference Appendix C: Expanded & Single Valid File Labels and Characters). The ASCII message data and display mode information, along with various other control codes, are stored in Expanded STRING files on a specified Drive in the STRING (“S”) Folder.

**Table 6: Write STRING File Transmission Packet**

		Command Code	Data Field		
"G"		<0X0F>	Drive	Folder	File Label
A		B	C	D	E

Item	Name	Description
A	Command Code	“G” = Write TEXT file Example: <0x01>Z00<0x02> <b>G</b> <0x0F>DSDM 500<0x04>
B	Expansion	One ASCII character that indicates the TEXT file being accessed. See “Appendix C: Expanded & Single Valid File Labels & Characters” Example: <0x01>Zoo<0x02>G< <b>0x0F</b> >ESDH 977<0x04>
C	Drive	Example: <0x01>Z00<0x02>G<0x0F> <b>D</b> SDM 676<0x04>
D	Folder	Example: <0x01>Z00<0x02>G<0x0F> <b>E</b> SDH 899<0x04>
E	File Label	Example: <0x01>Z00<0x02>G<0x0F>DSD <b>M</b> 566<0x04>

**EXAMPLE NOTE:** The coding examples in “Table 6” get called from the Expanded TEXT, with embedded Expanded STRING, file: <0x01>Z00<0x02>A<0x0F>DTfAGoal:<0x10><0x0F>ESDH Actual:<0x10><0x0F>DSDM<0x04> and when the example in Item “A” (Table 6) is sent to the display (populating “<0x10>DSDM” variable) – it will read:  
Goal: DH                      Actual: 500

The “Goal” quantity has not yet been populated so “DH” will stay the default – the last value sent.

Then when the example in Item “B” (Table 6) is sent to the display (populating “<0x10>ESDH” variable) – it will read:  
Goal: 977                      Actual: 500



The “Actual” quantity from Item “A” will stay the default – the last value sent for “<0x10>DSDM”. Then when the example in Item “C” (Table 6) is sent to the display (populating “<0x10>DSDM” variable) – it will read:

Goal: 977      Actual: 676

The “Goal” quantity from Item “B” will stay the default – the last value sent for “<0x10>ESDH”. Then when the example in Item “D” (Table 6) is sent to the display (populating “<0x10>ESDH” variable) – it will read:

Goal: 899      Actual: 676

The “Actual” quantity from Item “C” will stay the default – the last value sent for “<0x10>DSDM”. Then when the example in Item “E” (Table 6) is sent to the display (populating “<0x10>DSDM” variable) – it will read:

Goal: 899      Actual: 566

The “Goal” quantity from Item “D” will stay the default – the last value sent for “<0x10>ESDH”.

Example of writing an Expanded STRING file format:

<0x01>Z00<0x02>**G<0x0F>ES\$b** Writing a Expanded STRING file <0x04>

The bold “G” is the command (referenced in table 2) for the writing of a STRING file.

The bold “<0x0F>” is the ASCII command for file expansion .

The bold “E” specifies the Drive.

The bold “S” specifies the Folder.

The bold “\$b” is the File Name (reference Appendix C: Valid File Labels and Characters).

The above example will write (“G”) an expanded file (“\$b”) to Drive E (“E”) and place in Folder (“S”).

#### PLEASE NOTE:

- 1) Expanded STRING files must be added to TEXT files to be shown on the ‘LED’ display.
- 2) In the above examples no carriage return spaces need to be between the “File Label (“E” in table 6) and the beginning of your message or between the end of your message and the “End of Transmission” (“H” in table 2). Also all bold text in the above ASCII code are supplied to aid each circumstance and all non bolded format is explained in table 2.
- 3) There is a possibility of seventeen thousand six hundred and seventy-two (17,672) combination Expanded STRING file labels (reference Appendix C: Expanded & Single Valid File Labels and Characters), which is calculated by multiplying valid File label combinations and then multiplying by the amount of valid Drives for the specific Folder ([94 x 94] x 2).
- 4) Calls (<0x10>) from Expanded STRING files to TEXT files are explained in section 6.1.5.
- 5) If a Expanded STRING Write command (“G”) is used then the STRING Folder must be a (“S”).
- 6) Valid Drives can be “D:” or “E:” (illustrated in the above table).
- 7) Nested STRING Files (reference section 6.1.6) can consolidate (group together) Calls to STRING File(s). This command can not be utilized in 4U2SEE Firmware versions prior to 8E04.
- 8) The expansion of STRING files can **NOT** be accomplished prior to 4U2SEE firmware version 8E04.

#### 6.1.5 Simple and Expanded Calls from STRING (“S”) Files to TEXT Files (“T”)

Simple and Expanded STRING files are embedded in Simple and Expanded TEXT files. This internal addition must be executed properly to have messages displayed on the ‘LED’ display. A **Call** (“<0x10>”) or multiple **Calls** can be made from a TEXT file to a STRING file, but one (1) to ten (10) STRING file(s) can update at a time (reference section 6.1.6).

The example below illustrates the multiple Simple STRING **Call** to Simple Text files:

1) <0x01>Z00<0x02>AA Goal:<0x10>U Actual:<0x10>V<0x04>

The above example (example 1) is a Simple TEXT file, writing (“A”) to File “A”, being stored in the default Folder (“T”) on the default Drive (“D:”). **Calling** (“<0x10>”) Simple STRING files “U” and “V” to populate TEXT file.

The example below illustrates the multiple Expanded STRING **Call** to Expanded TEXT files:

2) <0x01>Z00<0x02>A<0x0F>DT%% Goal:<0x10><0x0F>DSBc Actual:<0x10><0x0F>ESIE <0x04>

The above example (example 2) is an Expanded TEXT file, writing (“A”) to File “%%”, being stored in

the Folder (“T”) on the Drive (“D:”). **Calling** (“<0x10>”) Expanded STRING File “Bc” on Drive D (“D”) stored in Folder S (“S”) and File “IE” on Drive E (“E”) stored in Folder S (“S”).

The example below illustrates the multiple Simple STRING **Call** to Expanded TEXT files:

3) <0x01>Z00<0x02>AA Goal:<0x10><0x0F>DSBc Actual:<0x10><0x0F>ESIE <0x04>

The above example (example 3) is a Simple TEXT file, writing (“A”) to File “A”, being stored in the default Folder (“T”) on the default Drive (“D:”). **Calling** (“<0x10>”) Expanded STRING File “Bc” on Drive D (“D”) stored in Folder S (“S”) and File “IE” on Drive E (“E”) stored in Folder S (“S”).

The example below illustrates the multiple Expanded STRING **Call** to Simple TEXT files:

4) <0x01>Z00<0x02>A<0x0F>DT%% Goal:<0x10>U Actual:<0x10>V<0x04>

The above example (example 4) is a Expanded TEXT file, writing (“A”) to File “%%”, being stored in the Folder (“T”) on the Drive (“D:”). **Calling** (“<0x10>”) Simple STRING Files “U” and “V” on default Drive D (“D”) and stored in default Folder S (“S”).

#### PLEASE NOTE:

- 1) All bold text in the above ASCII code examples are supplied to aid each circumstance and all non bolded format text is explained in Table 2.

#### 6.1.6 Nesting for STRING Files

This command is used to update multiple STRING Files located on the same sign. Utilizing the following Nesting format, a single command can update up to ten (10) STRING Files on the same display board.

<0x01>Z00<0x02>**Gxyyyy**<0x02>**Gxyyyy**.....<0x02>**Gxyyyy**<0x04>

where “G” and <0x02> (bold) is explained previously

where “x”(bold) represents the file label to write, either Simple or Expanded format.

where “y” (bold) represents data to be written to file.

The “Start of Text” (<0x02> - item “D” in table 2) denotes the beginning of a new File to be updated on the same display.

**EXAMPLE:** <0x01>Z00<0x02>GaThis is Simple STRING file a<0x02>G<0x0F>ESaaThis is Expanded STRING file aa<0x02>G<0x0F>ESVgThis is Expanded STRING file Vg<0x02>GcThis is a Simple STRING file c<0x04>

The above command is the same as sending four (4) separate commands:

<0x01>Z00<0x02>GaThis is a Simple STRING file a<0x04>

<0x01>Z00<0x02>G<0x0F>ESaaThis is a Expanded STRING file aa<0x04>

<0x01>Z00<0x02>G<0x0F>ESVgThis is a Expanded STRING file Vg<0x04>

<0x01>Z00<0x02>GcThis is a Simple STRING file c<0x04>

**PLEASE NOTE:** All bold text in the above ASCII code format example is supplied to aid each circumstance and all non-bolded format text is explained in Table 2.

#### 6.1.7 Special Functions

Electro-Matic Products network supports a range of Special Function commands which give you access to internal registers, diagnostics and other items.

## 6.2 Customizing Defaults

The factory defaults listed in section 2.3 can be customized (altered) to accommodate your specific needs, unless otherwise noted. You can customize a solitary value or alter all the default settings at once.

#### 6.2.1 Default Instructions E# Command

The ASCII instruction code for setting custom default values is:

<0x01>Z00<0x02>**E#XX**<0x04>

The 'E' (explained below) is a special function command

The '#' (explained below) is the default set command

The 'Xx' (explained below) represent the default variable commands with their associated attributes (Bb, Cc, Dd, Ff, Hh, Ll, Mm, Oo, Ss, Tt, Vv and Ww.)

#### 1. Standard (Figure 3)

**Figure 3: Standard Transmission Packet**

<0x01>	Address Pre-filler	Sign Address	<0x02>	Command Code	Data Field	<0x04>
--------	-----------------------	-----------------	--------	-----------------	---------------	--------

Non-printable control characters like <0x01> are used in the packet:

**Table 7: Standard Transmission Packet Format for Defaults**

<0x01>	Address Pre-filler	Sign Address	<0x02>	Command Code	Data Field	<0x04>
A	B	C	D	E	F	G
Item	Name	Description				
A	<0x01>	The <0x01> is the “Start of Header” ASCII character. <b>&lt;0x01&gt;Z00&lt;0x02&gt;E#</b> enter custom default variable(s) and attributes here<0x04>				
B	Address Pre-filler	Reserved, always an ACSII character (reference Appendix C: Valid File Labels & Characters). All examples throughout this manual use the ASCII character "Z" (this is not required). Example: <0x01> <b>Z00&lt;0x02&gt;E#</b> enter customer default variable(s) and attributes here <0x04>				
C	Sign Address	The identifier or “address” of the sign represented by two to four ASCII digits as a number between “00”and “99”. Address “00”is reserved as a broadcast address. The first two digits represent the group number and the next two digits represent this unit number. (GGUU) When only 2 digits are used in the sign address in this field then the group address is assumed to be 00 and the 2 digits represent the unit number (UU). This change is typical throughout the manual (changing to 00 after the Z) <b>PLEASE NOTE:</b> The Group & Unit number (GGUU) of your display board can be changed with Electro-Matc's input device Sigma 3000 Software.. Example: <0x01> <b>Z00&lt;0x02&gt;E#</b> enter custom default variable(s) and attributes here<0x04>				
D	<0x02>	“Start of Text “(ASCII) character. <0x02> always precedes a Command Code. Example: <0x01>Z00 <b>&lt;0x02&gt;E#</b> enter custom default variable(s) and attributes here<0x04>				
E	Command Code: "E#"	One ASCII character that defines the transmission and data <0x01>Z00<0x02> <b>E#</b> enter custom default variable(s) and attributes here<0x04>				
F	Data Field	Made up of ASCII characters. The Data Field format is dependent on the preceding Command Code and File Label. Example: <0x01>Z00<0x02>E# <b>enter custom default variable(s) and attributes here&lt;0x04&gt;</b>				
G	<0x04>	“End of Transmission”(ASCII) character Example: <0x01>Z00<0x02>E#enter custom default variable(s) and attributes here <b>&lt;0x04&gt;</b>				

The variables below are for setting the defaults (Item "F" in Table 7):

B – Specifies background color of display

b – Sets the background color attribute (setting).

0 = Black (Factory default)	<0x01>Z00<0x02>E#B0<0x04>
1 = Red	<0x01>Z00<0x02>E#B1<0x04>
2 = Green	<0x01>Z00<0x02>E#B2<0x04>
3 = Yellow	<0x01>Z00<0x02>E#B3<0x04>

C – Specifies Text character color

c – Sets the color attribute (setting).

0 = Black	<0x01>Z00<0x02>E#C0<0x04>
1 = Red (Factory default)	<0x01>Z00<0x02>E#C1<0x04>
2 = Green	<0x01>Z00<0x02>E#C2<0x04>
3 = Yellow	<0x01>Z00<0x02>E#C3<0x04>

D – Sets the default drive

d – Sets the drive attribute

E = RAM (random access drive – volatile)	<0x01>Z00<0x02>E#DE<0x04>
D = Flash (non-volatile) memory (Factory default)	<0x01>Z00<0x02>E#DD<0x04>

F – Specifies font

f – Sets the font attribute (setting).

0 = Normal 5	<0x01>Z00<0x02>E#F0<0x04>
1 = Normal 7 (Factory default)	<0x01>Z00<0x02>E#F1<0x04>
2 = Normal 14	<0x01>Z00<0x02>E#F2<0x04>
3 = Normal 15	<0x01>Z00<0x02>E#F3<0x04>
4 = Normal 16	<0x01>Z00<0x02>E#F4<0x04>
5 = Bold 14	<0x01>Z00<0x02>E#F5<0x04>
6 = Bold 15	<0x01>Z00<0x02>E#F6<0x04>
7 = Bold 16	<0x01>Z00<0x02>E#F7<0x04>

H – Specifies Vertical Position

h – Sets the Vertical Position attribute (setting).

C= Center (middle)	<0x01>Z00<0x02>E#HC<0x04>
T= Top	<0x01>Z00<0x02>E#HT<0x04>
B= Bottom	<0x01>Z00<0x02>E#HB<0x04>
F= Fill (Factory default)	<0x01>Z00<0x02>E#HF<0x04>

L – Specifies Line Spacing

l – Sets the line spacing between messages attribute (setting).

1 = 1 pixel (Factory default)	<0x01>Z00<0x02>E#L1<0x04>
2 = 2 pixel	<0x01>Z00<0x02>E#L2<0x04>
3 = 3 pixel	<0x01>Z00<0x02>E#L3<0x04>
4 = 4 pixel	<0x01>Z00<0x02>E#L4<0x04>
5 = 5 pixel	<0x01>Z00<0x02>E#L5<0x04>
6 = 6 pixel	<0x01>Z00<0x02>E#L6<0x04>
7 = 7 pixel	<0x01>Z00<0x02>E#L7<0x04>
8 = 8 pixel	<0x01>Z00<0x02>E#L8<0x04>
9 = 9 pixel	<0x01>Z00<0x02>E#L9<0x04>

M – Specifies IN mode

m – Sets the IN mode attribute (setting).

(Reference table 8)

O – Specifies OUT mode

o – Sets the OUT mode attribute (setting).

(Reference table 8)

S - Specifies Speed of characters moving across face of sign

s - Sets the Speed of characters moving across the sign face attribute (setting).

1= Slowest Speed	<0x01>Z00<0x02>E#S1<0x04>
2= Slow Speed	<0x01>Z00<0x02>E#S2<0x04>
3= Medium Speed (Factory default)	<0x01>Z00<0x02>E#S3<0x04>
4= Fast Speed	<0x01>Z00<0x02>E#S4<0x04>
5= Fastest Speed	<0x01>Z00<0x02>E#S5<0x04>

T- Specifies Stay Time between Frames of messages

t- Sets the Stay Time between Frames of messages attribute (setting).

1 second	<0x01>Z00<0x02>E#T1<0x04>
2 seconds	<0x01>Z00<0x02>E#T2<0x04>
3 seconds (Factory Default)	<0x01>Z00<0x02>E#T3<0x04>
4 seconds	<0x01>Z00<0x02>E#T4<0x04>
5 seconds	<0x01>Z00<0x02>E#T5<0x04>
6 seconds	<0x01>Z00<0x02>E#T6<0x04>
7 seconds	<0x01>Z00<0x02>E#T7<0x04>
8 seconds	<0x01>Z00<0x02>E#T8<0x04>
9 seconds	<0x01>Z00<0x02>E#T9<0x04>

V- Specifies Horizontal Justification

v - Sets the Horizontal Justification attribute (setting).

R= Right	<0x01>Z00<0x02>E#VR<0x04>
L= Left	<0x01>Z00<0x02>E#VL<0x04>
C= Center (Factory default)	<0x01>Z00<0x02>E#VC<0x04>

W- Specifies Word Wrap

w - Sets the Word wrap attribute (setting).

0= OFF Text will move across face from left to right	<0x01>Z00<0x02>E#W0<0x04>
1= On (Factory default)	<0x01>Z00<0x02>E#W1<0x04>

Examples of setting defaults are shown below:

<0x01>Z00<0x02>E#C1DEF1Mb<0x04>

In the above example the default character color would be red, the default drive being "E:" (flash), the default font size being normal 7 and the default in mode would hold.

<0x01>Z01<0x02>E#C1DEF1<0x04>

In the above example the default character color would red, the default drive being "E:" (flash) and the default font size being normal 7.

<0x01>Z00<0x02>E#C3B2F0L2T3S5VCHFW1MaObDE<0x04>

This above default customization example would alter all factory default settings, some are not changed – next example will display the same as above example, because the unspecified defaults do not change from the factory settings. In the above and below examples, text character color is yellow, the background color is green, the font is normal 5, the line spacing is 2 pixels, the stay time is 3 (shown in above example, but not shown in following example), the speed is 3 (shown in above example, but not shown in following example), the horizontal justification is center (shown in above example, but not shown in following example), the vertical justification is fill (shown in above example, but not shown in following example), the wordwrap is 1 (shown in above example, but not shown in following example), The inward action mode is a (rotate - see section 6.3.2 and table 8), The outgoing action mode is b (hold – see section 6.3.2 and table 8) and the default drive is set to E.

<0x01>Z00<0x02>E#C3B2F0L2MaObDE<0x04>

#### PLEASE NOTE:

- 1) Same foreground and background color will make the 4U2SEE virtually unreadable.  
Example: <0x01>Z00<0x02>E#C2B2<0x04>
- 2) All non-prescribed default customization variables with attributes use factory settings – meaning all unset default variables will revert to factory settings.
- 3) Default customization can not be accomplished in 4U2SEE firmware versions prior to version 8E04.

- 4) Setting the Font attribute ('f') is based on the display's pixel height.
- 5) Setting the Vertical Position attribute ('h') is based on imagining the horizontal division of the display into three (3) segments – 'top', 'center' and 'bottom'. When 'top' is selected then the message sent to the display will appear on upper line. When 'fill' is selected it will occupy previously defined spaces. On a one-line display, all attributes appear as center attribute.
- 6) Setting the pixel (horizontal) is based on the LED's spacing between lines of text.
- 7) Setting the In mode attribute ('m') is based on the action of the message entering the display.
- 8) Setting the Out mode attribute ('o') is based on the action of the message when exiting the display.
- 9) Setting the Horizontal Justification attribute ('v') is based on imagining the vertical division of the display in to three (3) segments – 'right', 'left' and 'center'.
- 10) Setting the Word Wrap attribute ('w') will determine the fields of a message. Either continuous or paragraphed.

### 6.2.2 Restoring Sign to Factory Defaults

If <0x01>Z00<0x02>E#<0x04> is sent to the display, all factory defaults will be restored.

The ASCII instruction code for restoring factory defaults is:

<0x01>Z00<0x02>E#<0x04>

where "E" is a special function command.

where "#" is the default set command

#### PLEASE NOTE:

- 1) There is no specification for variables and related attributes.
- 2) Default customization can not be accomplished in 4U2SEE firmware versions previous to 8E04.

## 6.3 Start Mode Commands <0x1B>

Mode commands <0x1B> can be embedded into TEXT messages, being used for positioning and incoming/outgoing activities.

### 6.3.1 Vertical Positioning Command Code

This command will allow position specification on the display.

Middle	<0x01>Z00<0x02>AA<0x1B>a This is where to type message <0x04>
Top	<0x01>Z00<0x02>AA<0x1B>"a This is where to type message <0x04>
Bottom	<0x01>Z00<0x02>AA<0x1B>&a This is where to type message <0x04>
Fill	<0x01>Z00<0x02>AA<0x1B>0a This is where to type message <0x04>

#### PLEASE NOTE:

- 1) The example for "Middle" has a carriage return (space) between the Mode ("<0x1B>") command and the rotate command ("a"). Specifying "a" will automatically create a No Wrap situation.
- 2) FlexSender (Electro-Matic input device) does not acknowledge the ASCII coding for "Top", you must utilize hexadecimal code of <0x22> when using this format.  
Example: <0x01>Z00 0x02>AA<0x1B><0x22>a This is where to type message <0x04>
- 3) In the above examples no carriage return spaces need to be between the "Positioning Mode" command and the beginning of your message or between the end of your message and the "End of Transmission" ("H" in Table 2 ). Also all bold text in the above ASCII code examples are supplied to aid each circumstance and all non-bolded format text is explained in Table 2. In the TEXT examples above, all Files ("A") are on the default Drive in the default Folder.
- 4) Mode vertical positioning attributes can be referenced in section 6.3.2 and table 8. In the above examples "a" (rotate) was selected, but numerous options are available.
- 5) If a message is sent to the display with the rotate ("a") option and has a combined character TEXT width less then the signs width limitation (message character count of seventy-nine (79) and sign width is eighty (80)), then message will stay stationary, if message exceeds width limitation the text will move from right to left across the screen and repeat.
- 6) If a Mode ("<0x1B>") is not specified then display feature will resort to default settings.
- 7) Vertical positioning command codes are irrelevant on one line displays.
- 8) For an example of vertical positioning for expanded files reference lines 1-4 of Appendix D: Examples of FlexSender Code.



### 6.3.2 Character In and Out Command Codes

This command will allow activity specifications for incoming and outgoing messages on the display. When an In mode is specified, the message will perform that action when entering the display. When an Out mode is specified, the message will perform that action when exiting the display.

In Mode command format - <0x1B>I[mode]

Out Mode command format - <0x1B>O[mode]

**Table 8: Display Modes**

Display Modes	“z” value	Display Description
Rotate	a	Moves Right to Left
Hold	b	Stays Stationary
Flash	c	Flashes
*Random	d	Displays all modes
Move left	e	Moves Left to Right
Move right	f	Moves Right to Left
Scroll O/L	g	Scroll Out to Left
Scroll O/R	h	Scroll Out to Right
Move up	i	Moves Upward
Move down	j	Moves Downward
Scroll O/C	k	Scrolls Out to Center
Unveil up	l	Unveils Upwards
Unveil down	m	Unveils Downwards
Unveil in	n	Unveils Inwards
Unveil up/in	o	Unveils Upwards and Inwards
Unveil up/out	p	Unveils Upwards and Outwards
Splice across	q	Splice Horizontally
Splice verti	r	Splice Vertically
Fall left	s	Falling to Left
Fall right	t	Falling to Right
Veretian hor	u	Horizontal Venetian Blind
Veretian Ver	v	Vertical Venetian Blind
Rain	w	Raining
Materialize	x	Appears/Disappears in Dots
Twinkle	z	Appears/Disappears in Stars
Squiggle	1	Appears/Disappears in Squiggle
Radar	2	Appears/Disappears in Sonar
FanOpen	3	Fan Opening Up
FanClose	4	Fan Closing Up
RotateRight	5	Rotation to Right
RotateLeft	6	Rotation to Left
Center2Corner	7	Displays Center to Corner
Corner2center	8	Displays Corner to Center
Center2Allsz	9	All Sides From Center
Alls2Center	A	All Sides to Center
FourBlock2Cor	B	Block From Center
FourBlock2Cen	C	Block to Center
FourBlockOut	D	Block Out of Display
FourBlockIn	E	Block In to Display
LeftCorRectIn	F	Left In From Corner

RightCorRectIn	G	Right In From Corner
LBottomRectI	H	Left In From Bottom
RBottomRectI	I	Right In From Bottom
LftCoDiagonal	J	Displays Left Diagonal
RtCoDiagonalI	K	Displays Right Diagonal
LBtmDiagonal	L	Displays Bottom Left Diagonal
RBtmDiagonal	M	Displays Bottom Right Diagonal
Lft2RtDownCor	N	Left to Right Corner in Downwards
Rt2LftDownCor	O	Right to Left Corner in downwards
Left2RtUpCor	P	Left to Right Corner in Upwards
Rit2LftUpCor	Q	Right to Left Corner in Upwards
GrowUp	R	Growing Up

Command format <0x1B>X/Yz

where “X” = incoming display modes

where “Y” = outgoing display modes

where “z” = display appearance (reference above chart)

Examples TEXT files:

Displays incoming message scrolling out to left (“g”) and outgoing message rains (“w”).

<0x01>Z00<0x02>AA<0x1B>lg<0x1B>OwThis is where to type TEXT message<0x04> storing on the default Drive in the default Folder.

Displays incoming message rotates (“a”) and outgoing message squiggles (“1”).

<0x01>Z00<0x02>AA<0x1B>la<0x1B>O1This is where to type TEXT message<0x04> storing on the default Drive in the default Folder.

Displays incoming message scrolling out to left (“g”) and outgoing message rains (“w”).

<0x01>Z00<0x02>A<0x0F>ETag<0x1B>lg<0x1B>OwThis is where to type TEXT message<0x04>storing on Drive E in the TEXT Folder

Displays incoming message rotates (“a”) and outgoing message squiggles (“1”).

<0x01>Z00<0x02>A<0x0F>ETag <0x1B>la<0x1B>O1This is where to type TEXT message<0x04>storing on Drive E in the TEXT Folder

#### PLEASE NOTE:

- 1) Setting either the In or Out Mode to No Wrap (“a”) will apply No Wrap to both modes.
- 2) When a mode (In/Out) is not set, then the default will be used.
- 3) In the above example no carriage return spaces need to be between the “Out Mode “placement and the beginning of your message or between the end of your message and the “End of Transmission”(“H” in Table 2 ). Also all bold text in the above ASCII code examples are supplied to aid each circumstance and all non-bolded format text is explained in Table 2.
- 4) The ASCII character “y” is reserved.
- 5) All Simple and Expanded TEXT files are designated for the TEXT (“T”) Folder.
- 6) The In/Out command codes shown in table 8 are not applicable to 4U2SEE firmware versions prior to 8E04.
- 7) For examples of IN/OUT modes for expanded files reference lines (in) 5-56 and lines (out) 57-108 of Appendix D: Examples of FlexSender Code.

## 6.4 Controlled Character Display (Special Functions)

Various control functions are available for controlling how messages are displayed on the 4U2SEE displays. The various controlled character display functions and the commands associated with them are explained below:

### 6.4.1 Character Flash Command Code (07 Hex) <0x07>

Character Flash Command Code ( 07 Hex)

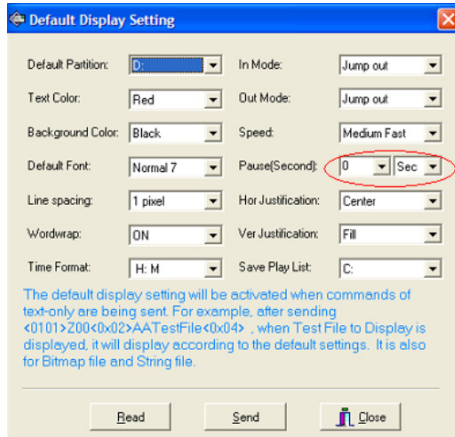
This command will enable text to flash.

‘0’ Flash off (Default)

‘1’Flash on



This command was formerly dependent on the Pause value in the “Default Display Setting” and did not work if the Pause is set to 0 seconds as shown in the screen below. The firmware has been modified (9721-01 or newer) so that the flashing feature is independent of the pause time and flashing will work even if the pause is set to 0 seconds.



For example, the message below updates text file A with a message that flashes.  
 <0x01>Z00<0x02>AA<0x1B>0b<0x07>1This message will flash<0x04>

The message below updates text file A with a message that does not flash.  
 <0x01>Z00<0x02>AA<0x1B>0b<0x07>0This message will not flash<0x04>

The flash command can also be called in a string file.

For example, the message below updates string file U with a message that flashes.  
 <0x01>Z00<0x02>GU<0x07>1This message will flash<0x04>

The message below updates string file U with a message that does not flash.  
 <0x01>Z00<0x02>GU<0x07>0This message will not flash <0x04>

#### 6.4.2 Display Date and Time Stamp Command Codes <0x0B>

The available command display formats are given below.

MM/DD/YY	<0x01>Z00<0x02>AA<0x0B>0<0x04>
DD/MM/YY	<0x01>Z00<0x02>AA<0x0B>1<0x04>
MM-DD-YY	<0x01>Z00<0x02>AA<0x0B>2<0x04>
DD-MM-YY	<0x01>Z00<0x02>AA<0x0B>3<0x04>
MM.DD.YYYY	<0x01>Z00<0x02>AA<0x0B>4<0x04>
YY(year)	<0x01>Z00<0x02>AA<0x0B>5<0x04>
YYYY	<0x01>Z00<0x02>AA<0x0B>6<0x04>
MM(number)	<0x01>Z00<0x02>AA<0x0B>7<0x04>
MMM(character)	<0x01>Z00<0x02>AA<0x0B>8<0x04>
DD(day)	<0x01>Z00<0x02>AA<0x0B>A<0x04>
Day of week-No.	<0x01>Z00<0x02>AA<0x0B>B<0x04>
Day of week-CH.	<0x01>Z00<0x02>AA<0x0B>9<0x04>
HH(hour)	<0x01>Z00<0x02>AA<0x0B>C<0x04>
Min(minute)	<0x01>Z00<0x02>AA<0x0B>D<0x04>
SS(second)	<0x01>Z00<0x02>AA<0x0B>E<0x04>
HH:MIN	<0x01>Z00<0x02>AA<0x0B>F<0x04>
HH:MIN(AM/PM)	<0x01>Z00<0x02>AA<0x0B>a<0x04>
HH:MIN(zone)	<0x01>Z00<0x02>AA<0x0B>b<0x04>

#### PLEASE NOTE:

- 1) The shown date and time commands are affiliated with the display board. The “Host” computer time can be accessed by sending the FlexSender command: <0x01>Z00<0x02>AA<T><0x04>. The command for input device time is (“<T>”)
- 2) All bold text in the above ASCII code examples are supplied to aid each circumstance and all non-bold format text is explained in Table 2.

- 3) There are limitations to the execution of this command in 4U2SEE firmware versions prior to version 8E04.
- 4) For examples of expanded files display date and time stand command code reference lines 111 128 of Appendix D: Examples of FlexSender Code.

#### 6.4.3 Character Frame New Page Command Code <0x0C>

This command turns part of the message following it into a new frame.

An example of the Frame command: <0x01>Z00<0x02>AA<0x1B>"bThis message will hold on top of display<b><0x0C><0x1B>&athis portion of message will rotate on the bottom and be on another page<0x04>

##### PLEASE NOTE:

- 1) Any command that follows the Frame ("**<0x0C>**") will display message on a new page.
- 2) All bold text in the above ASCII code examples are supplied to aid each circumstance and all non-bolded format text is explained in Table 2.
- 3) This command is not applicable to 4U2SEE firmware versions prior to version 8E04.
- 4) For examples of expanded files character frame (new page) command code reference line 128 of Appendix D: Examples of FlexSender Code.

#### 6.4.4 Character Line New Line - Word Wrap Command Code <0x0D>

This command allows messages to be placed on the next line.

An example of the Line command: <0x01>Z00<0x02>AA<0x1B>"bThis message will hold on top of display<b><0x0D><0x1B>&athis portion of message will rotate on the bottom and be on another line<0x04>

##### PLEASE NOTE:

- 1) Any command that follows ("**<0x0D>**") will display text on a new line
- 2) All bold text in the above ASCII code examples are supplied to aid each circumstance and all non-bolded format text is explained in Table 2.
- 3) When the TEXT is wrapping from one line to the next, the display board will go back from the end of the first line of TEXT and search for the first carriage return space. The display board will wrap from this point so no whole words are broken. If a carriage return is not found then the sign will wrap from the last character fitting on that line.
- 4) This command is not applicable to 4U2SEE firmware versions prior to version 8E04.
- 5) For examples of expanded files character line (new line-word wrap) command code reference line 130 of of Appendix D: Examples of FlexSender Code.

#### 6.4.5 Character Horizontal Justification Command Code <0x0A>

Horizontally aligns your message.

Left	<0x01>Z00<0x02>AA<b><0x0A>0 This is where you type message <0x04>
Center (default)	<0x01>Z00<0x02>AA<b><0x0A>1 This is where you type message <0x04>
Right	<0x01>Z00<0x02>AA<b><0x0A>2 This is where you type message <0x04>

##### PLEASE NOTE:

- 1) In the above example no carriage return spaces need to be between the "Horizontal Justification" placement and the beginning of your message or between the end of your message and the "End of Transmission" ("H" in Table 2 ). Also all bold text in the above ASCII code examples are supplied to aid each circumstance and all non-bolded format text is explained in Table 2.
- 2) For examples of expanded files character horizontal justification reference line 131-133 of Appendix D: Examples of FlexSender Code.

#### 6.4.6 Character Foreground Color Command Code <0x1C>

This command allows the selection of text (character) color of your message.

Black	<0x01>Z00<0x02>AA<b><0x1C>0This is where you type message<0x04>
Red	<0x01>Z00<0x02>AA<b><0x1C>1This is where you type message<0x04>
Green	<0x01>Z00<0x02>AA<b><0x1C>2This is where you type message<0x04>
Yellow	<0x01>Z00<0x02>AA<b><0x1C>3This is where you type message<0x04>

Mix color 1	<0x01>Z00<0x02>AA<0x1C> <b>4</b> This is where you type message<0x04>
Mix color 2	<0x01>Z00<0x02>AA<0x1C> <b>5</b> This is where you type message<0x04>
Mix color 3	<0x01>Z00<0x02>AA<0x1C> <b>6</b> This is where you type message<0x04>
Mix color 4	<0x01>Z00<0x02>AA<0x1C> <b>7</b> This is where you type message<0x04>

**PLEASE NOTE:**

- 1) Same foreground and background color will make the 4U2SEE virtually unreadable.
- 2) For examples of expanded files character (foreground) color command code reference lines 134-141 of Appendix D: Examples of FlexSender Code.

**6.4.7 Character Background Color Command Code <0x1D>**

This command allows the selection of background colors of your message.

Black	<0x01>Z00<0x02>AA<0x1D> <b>B0</b> This is where you type message<0x04>
Red	<0x01>Z00<0x02>AA<0x1D> <b>B1</b> This is where you type message<0x04>
Green	<0x01>Z00<0x02>AA<0x1D> <b>B2</b> This is where you type message<0x04>
Yellow	<0x01>Z00<0x02>AA<0x1D> <b>B3</b> This is where you type message<0x04>

**PLEASE NOTE:**

- 1) Same foreground and background color will make the 4U2SEE virtually unreadable.
- 2) For examples of expanded files character (background) color command code reference lines 142-145 of Appendix D: Examples of FlexSender Code.

**6.4.8 Character Speed Command Code**

This command allows the selection of five (5) different speed grades, which determine the speed the characters move across the face of your display.

Slowest Speed	<0x01>Z00<0x02>AA<0x15> This is where you type message <0x04>
Slow Speed	<0x01>Z00<0x02>AA<0x16> This is where you type message <0x04>
Medium Speed	<0x01>Z00<0x02>AA<0x17> This is where you type message <0x04>
Fast Speed	<0x01>Z00<0x02>AA<0x18> This is where you type message <0x04>
Fastest Speed	<0x01>Z00<0x02>AA<0x19> This is where you type message <0x04>

**PLEASE NOTE:**

- 1) In the above examples no carriage return spaces need to be between the Speed Selection and the beginning of your message or between the end of your message and the "End of Transmission" ("H" in Table:2) also all bold text in the above ASCII examples are supplied to aid each circumstance and all non-bolded format text is explained in Table 2.
- 2) This command is not applicable to 4U2SEE firmware versions prior to version 8E04.
- 3) For examples of expanded files character speed command code reference lines 146-150 of Appendix D: Examples of FlexSender Code.

**6.4.9 Character Font Command Code <0x1A>**

This command allows the selection of character font height.

Font Normal 5	<0x01>Z00<0x02>AA<0x1A> <b>0</b> This is where you type your message<0x04>
Font Normal 7	<0x01>Z00<0x02>AA<0x1A> <b>1</b> This is where you type your message<0x04>
Font Normal 14	<0x01>Z00<0x02>AA<0x1A> <b>2</b> This is where you type your message<0x04>
Font Normal 15	<0x01>Z00<0x02>AA<0x1A> <b>3</b> This is where you type your message<0x04>
Font Normal 16	<0x01>Z00<0x02>AA<0x1A> <b>4</b> This is where you type your message<0x04>
Font Bold 14	<0x01>Z00<0x02>AA<0x1A> <b>5</b> This is where you type your message<0x04>
Font Bold 15	<0x01>Z00<0x02>AA<0x1A> <b>6</b> This is where you type your message<0x04>
Font Bold 16	<0x01>Z00<0x02>AA<0x1A> <b>7</b> This is where you type your message<0x04>

**PLEASE NOTE:**

- 1) Based on the physical properties of your display, the transfer of some ASCII code may render your display inoperable. To rectify this situation (assuming display board can be salvaged), cycle power (electricity) to the message controller and re-start your computer. If still unable to communicate, cycle power (electricity) to the sign - then during "Initial Power-up" (section 2.1) send the ASCII command <0x01>Z00<0x02>E\$<0x04>. The transfer of this command, should resolve your operation issue, but delete **ALL** files in RAM (E:) and Flash (D:) drives.
- 2) For examples of expanded files character font command code reference Appendix D: Examples of FlexSender Code.

**6.4.10 Character Hold (Stay) Command Code <0x0E>**

This command determines the available pause rate of the display. This rate of pause is directly proportional to the In/Out modes. This command is based on seconds, from 00 – 99 (must be a two (2) digit code). For simplicity not all code possibilities are shown.

00 seconds	<0x01>Z00<0x02>AA <b>&lt;0x0E&gt;00</b> This is where you type your message <0x04>
01 seconds	<0x01>Z00<0x02>AA <b>&lt;0x0E&gt;01</b> This is where you type your message <0x04>
.	.
05 seconds	<0x01>Z00<0x02>AA <b>&lt;0x0E&gt;05</b> This is where you type your message <0x04>
.	.
10 seconds	<0x01>Z00<0x02>AA <b>&lt;0x0E&gt;10</b> This is where you type your message <0x04>
.	.
99 seconds	<0x01>Z00<0x02>AA <b>&lt;0x0E&gt;99</b> This is where you type your message <0x04>

**PLEASE NOTE:**

- 1) In the above examples no carriage return spaces need to be between the Hold Time and the beginning of your message or between the end of your message and the “End of Transmission” (“H” in Table 2 ). Also all bold text in the above ASCII code examples are supplied to aid each circumstance and all non-bolded format text is explained in Table 2.
- 2) This command is not applicable to 4U2SEE firmware versions prior to version 8E04.
- 3) For examples of expanded files character hold (stay) command code reference Appendix D: Examples of FlexSender Code.

**6.4.11 Play List or Run Sequence**

Existing or creating messages you would like to appear on the display, can be apart of a play list or run sequence.

To construct a play list or a run sequence the following ASCII TEXT (Simple or Expanded) File format must be followed:

<0x01>Z00<0x02>**E.SLXY**<0x04>

To further explain setting up the play list (run sequence) the ‘Command Code’ has been bolded. See below for an explanation:

The ‘E’ is a special function command

The ‘.’ (period) is the set play list command

The ‘SL’ is an extension of setting the list of messages

The ‘XY’ is a designation for the list of TEXT (Simple or Expanded) Files you would like to appear in your play list (sequence).

To construct a play list (run sequence) in ASCII TEXT (Simple or Expanded) Files the following steps must be adhered to:

- 1) Apply (existing) or write a series of TEXT (Simple or Expanded) Files to the display, making resident in memory:

<0x01>Z00<0x02>AA Sample A <0x04>	(Simple TEXT File)
<0x01>Z00<0x02>Aq Sample B <0x04>	(Simple TEXT File)
<0x01>Z00<0x02>A<0x0F>DTxf Sample C <0x04>	(Expanded TEXT File)
<0x01>Z00<0x02>A<0x0F>DTCb Sample E <0x04>	(Expanded TEXT File)
<0x01>Z00<0x02>Ag Sample F1 <0x04>	(Simple TEXT File)
<0x01>Z00<0x02>A<0x0F>ETAS Sample F <0x04>	(Expanded TEXT File)
<0x01>Z00<0x02>Aa Sample G <0x04>	(Simple TEXT File)
<0x01>Z00<0x02>A6 Sample H <0x04>	(Simple TEXT File)

Based on the explanation above, create a play list (sequence) for TEXT Files:

<0x01>Z00<0x02>**E.SLA6**<0x0F>ETAS<0x0F>DTCb<0x04>

The display will read and display four (4) files - separately:

Sample A  
Sample H  
Sample F  
Sample E

Another play list (sequence) creation, based on the previous example is:  
<0x01>Z00<0x02>**E.SLa**<0x0F>DTxfqg<0x04>

The display will read and display four (4) files - separately:  
Sample G  
Sample C  
Sample B  
Sample F1

**PLEASE NOTE:**

- 1) The display board will individually present each text message.
- 2) Message truncation is based on display's physical size and variable options.
- 3) All bold text in the above ASCII code examples are supplied to aid each circumstance and all non bolded text is explained in tables 2 and 7.
- 4) All play lists (sequences) must be set using the same Folder.
- 5) This command is not applicable to 4U2SEE firmware versions prior version 8E04.

#### 6.4.12 Deletion of Play List or Run Sequence

Play lists (sequences) must occasionally be edited to allow further communications to the 4U2SEE display board. This is accomplished by deleting the existing playlist and then creating a new playlist (Section 6.4.11).

The ASCII command of : <0x01>Z00<0x02>**E.**<0x04>  
where "E" represents a special function  
where "." (period) represents the delete delimiter

**PLEASE NOTE:**

- 1) This command will delete the current play list (sequence) without effecting the contents of individual files.
- 2) All bold text in the above ASCII code examples are supplied to aid each circumstance and all non bolded text is explained in Table: 2 & 7.
- 3) This modification of play lists (sequences) can not be accomplished with 4U2SEE prior to firmware version 8E04.
- 4) Deleting the play list will cause the display to show every text file stored in the unit until a new play list is established.

#### 6.4.13 Embed Time of Day into Messages

1. Call Time (13 Hex) Time of day will be displayed.

The command for calling time is  
<0x01>Z00<0x02>AA<0x13><0x04>

The command for calling time with in a string file is  
<0x01>Z00<0x02>GS<0x13><0x04>

Where "G"(47 Hex) is the Write STRING File Command

S = One ASCII character that indicates the STRING file being accessed

The time format would be the default format set by 'Set Time Format' Special function command. (""" 27 Hex) See Section 5.e

For Example,

The following command calls the string file U in text file A  
<0x01>Z00<0x02>AACURRENT TIME:<0x10>U<0x04>

The following command calls time with in the string file U.  
<0x01>Z00<0x02>GU<0x13><0x04>

**6.4.14 Embed Date into Messages**

2. Call date (0B Hex) The date will be displayed.

The command for calling date is

```
<0x01>Z00<0x02>AA<0x0B>D<0x04>
```

Where D is the date format as given below.

"0"(30H)=MM/DD/YY

"1"(31H)=DD/MM/YY

"2"(32H)=MM-DD-YY

"3"(33H)=DD-MM-YY

"4"(34H)=MM.DD.YY

"5"(35H)=DD.MM.YY

"6"(36H)=MM DD YY

"7"(37H)=DD MM YY

"8"(38H)=MMM.DD.YYYY

"9"(39H)=Day of week

Where DD=date, MM=month, YY=year, MMM=month abbreviation, and YYYY=year

For example,

The following command calls the date in MM/DD/YY format in text file A

```
<0x01>Z00<0x02>AA<0x0B>0<0x04>
```

The command for calling date with in a string file is

```
<0x01>Z00<0x02>GS<0x0B>D<0x04>
```

Where "G"(47 Hex) is the Write STRING File Command

S = One ASCII character that indicates the STRING file being accessed

D is the date format as given above

For Example,

The following command calls the string file U in text file A.

```
<0x01>Z00<0x02> AACURRENT DATE:<0x10>U<0x04>
```

The following command calls the date in MM/DD/YY format with in the string file U.

```
<0x01>Z00<0x02> GU<0x0B>0<0x04>
```

**6.4.15 Embed String File within Text**

When string files are called with in a text file, the string file names were previously displayed with an underscore on the display. The firmware has been modified (9721-01 or newer) so that the string file names are no longer displayed.

The command below sets two string files on the display. String file B for top line and F for the bottom line.

```
<0x01>Z00<0x02>AA<0x1B>"b<0x10>B<0x0D><0x1B>&a<0x10>F<0x04>
```

The display will be blank until the string files are updated.

The command below updates the string file values. <0x01>Z00<0x02>GB<0x1C>3<0x1D>0<0x1A>1<0x07>0<0x0A>1Top line Static<0x02>GF<0x1C>3<0x1D>0<0x1A>1<0x07>0<0x19><0x0A>1Bottom Line Scrolling<0x04>

**PLEASE NOTE: Firmware 9721-01 or newer**

**When string file values are updated, 4U2SEE 6205 Controllers for Tricolor Displays used**



to flash and display vertical lines for a second. The firmware has been modified (9721-01 or newer) so that when string file values are updated, the vertical lines are no longer displayed.

When string files are updated, scrolling of the second line in a two-line message used to skip. The firmware (9721-01 or newer) has been modified so that when string files are updated, the display gets updated only after the current message has completely scrolled out of the display to prevent skipping.

The following Write Special Function commands were added to the 4U2SEE ADP Protocol for firmware 9721-01 or newer.

## 6.5 Write Special Function Command Code- "E" (45 Hex)

### 6.5.1 Set Time of Day

a. Set Time of Day (" " (Space) 20 Hex)

The command for updating the time of the day (24-hour format) is  
<0x01>Z00<0x02>E HhMmSs<0x04>

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)

S = ASCII digit representing Seconds(10,s digit)

s = ASCII digit representing Seconds(1,s digit)

For example,

The following command will set the time on the display to 4:55:50 PM

<0x01>Z00<0x02>E 165550<0x04>

The time update command is also supported in a shorter format as given below

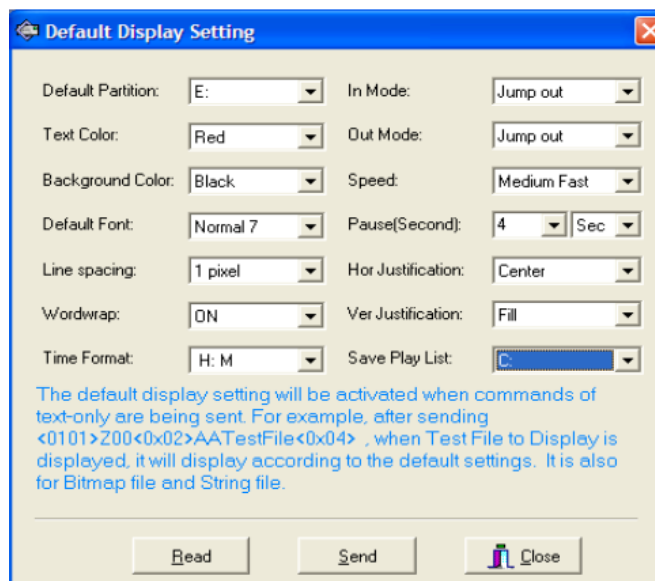
<0x01>Z00<0x02>E HhMm<0x04>

When the command is sent using just the HhMm, the seconds are set to '00'.

For example,

The following command will set the time on the display to 6:35 PM

<0x01>Z00<0x02>E 1835<0x04>



**6.5.2 Clear RAM Memory**

Clear RAM Memory (“\$” 24H)

The command clears the messages stored in the RAM memory (E Drive in 4U2SEE). It will also clear the play list from the default play list location based on the Default Display Setting (Either C or E Drive in 4U2SEE).

This command has been modified from the previous version of protocol. The command used to clear messages from both the Flash and RAM memory (D and E Drives in 4U2SEE).

For Example,

In the Default Display Settings screen below, the play list is saved in C. The following command will clear the RAM memory of the display and the play list saved in the C drive in this case.

```
<0x01>Z00<0x02>E$<0x04>
```

**6.5.3 Clear Flash and RAM**

Clear Flash and RAM Memory (“\$\$\$\$” 24H (four))

The command clears the messages stored in both the Flash and RAM memory (D and E drives in 4U2SEE). It will also clear the play list from both the Flash and RAM memory (C and E drives in 4U2SEE) regardless of where the play list is saved based on the Default Display Setting.

For Example,

The following command will clear the Flash and RAM memory and the play list from the display

```
<0x01>Z00<0x02>E$$$$<0x04>
```

The command clears the messages stored in both the Flash and RAM memory but does not delete the playlist from the Flash memory (C drive in 4U2SEE) on firmware 9721-01. The Playlist will be deleted with this command in next firmware release.

**6.5.4 Set Day of Week**

Set Day of Week (“&” 26 Hex)

The command for setting day of week is

```
<0x01>Z00<0x02>E&W<0x04>
```

Where W is the day of week as given below.

“1” 31 Hex = Sunday

“2” 32 Hex = Monday

“3” 33 Hex = Tuesday

“4” 34 Hex = Wednesday

“5” 35 Hex = Thursday

“6” 36 Hex = Friday

“7” 37 Hex = Saturday

For example,

The following command calls the day of week in text file A.

```
<0x01>Z00<0x02>AA<0x0B>9<0x04>
```

The following command sets the day of week to Sunday.

```
<0x01>Z00<0x02>E&1<0x04>
```

**6.5.5 Set Time Format**

Set Time Format (“”” 27 Hex)

The command for setting time format is



<0x01>Z00<0x02>E'T<0x04>

Where

T = ASCII character that represents how time is displayed on the sign.

Valid entries for T are

“S” 53 Hex = Standard AM/PM format (default)

“M” 4D Hex = 24-Hour (military) time

For example,

The following command will set the default time format on the display to 24-Hour format.

<0x01>Z00<0x02>E'M<0x04>

### 6.5.6 Set Date

Set Date (“;”3B Hex)

The command for setting date is

<0x01>Z00<0x02>E;MMDDYY<0x04>

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

For Example,

The following command calls the date on the display in the MM/DD/YY format

<0x01>Z00<0x02>AADATE:<0x0B>0<0x04>

The following command will set the date on the display to July 21, 2009

<0x01>Z00<0x02>E;072109<0x04>

The command sets the date wrong on firmware 9721-01. It will be fixed on the next firmware release.

## 7.0 Examples of Coding

Below are examples and explanations of combined categories:

<0x01>Z00<0x02>AA<0x1C>1<0x1D>B3 This is where you type message.<0x04>

This Simple TEXT example above writes a message to File “A” on the default Drive in the default Folder with the character color being red, the background color yellow and the message reads This is where you type message.

<0x01>Z22<0x02>Ag<0x1C>4<0x1B>I1<0x1B>O5 This is where you type message.<0x04>

This Simple TEXT example above writes a message to File “g” on default Drive in the default Folder with the character color being mixed, the in-coming activity of the message is squiggling, the out-going activity of the message is rotating to the right and the message reads This is where you type message.

<0x01>Z39<0x02>A6<0x1B>0b<0x1D>B2 This is where you type message.<0x04>

This Simple TEXT example above writes a message to File “6” on the default Drive in the default Folder with the mode command displaying the stationary message in the fill of the display, the background color is green and the message reads This is where you type message.

<0x01>Z26<0x02>A<0x0F>DTAA<0x1B>lg<0x1B>Ow This is where you type message <0x04>

This Expanded TEXT example above writes a message to File “AA” on the “D” Drive in the “T” Folder with the mode command displaying the incoming mode of scroll o/l and an outgoing mode of raining and the message reads This is where you type message.

<0x01>Z45<0x02>A8<0x1B> a<0x1C>1 This is where you type message <0x04>

This Simple TEXT example above writes a message to File “8” on the default Drive in the default Folder with the mode command displaying the rotating message in the center of the display in red text. This example does not default to wrap and the message reads This is where you type message.

<0x01>Z00<0x02>A<0x0F>ETDD<0x1B>”b<0x1C>2<0x1D>B1<0x1B>lw This is where you type message <0x04>

This Expanded TEXT example above writes a message to File “DD” on the “E” Drive in the “T” Folder with the mode command displaying the stationary message on top of the display in green text and a red background . This example defaults to wrap, the message enters the display raining and reads This is where you type message.

<0x01>Z45<0x02>Az<0x1B>”b<0x1C>3 This is where you type first message  
<0x0D><0x1B>&a<0x1C>1 This is where you type second message <0x04>

This Simple TEXT example above writes a message to File “z” on the default Drive in the default Folder with the position mode displaying the stationary message (This is where you type first message) on top of the display in yellow text (wrapping), the next line (“<0x0D>”) has the mode command displaying the rotating message on bottom of the display in red text. The new line does not default to wrap and the message reads This is where you type second message.

<0x01>Z00<0x02>A0<0x1B>0b<0x1C>3 This is where you type first page message <0x0C>  
<0x1B> a<0x1C>1 This is where you type second page message <0x04>

This Simple TEXT example above writes a message to File “0” on the default Drive in the default Folder with the mode command displaying the stationary message (This is where you type first page message) in fill of the display in yellow text (wrapping), then the next page (“<0x0C>”) has the mode command displaying the rotating message in center of the display in red text. The new page does not default to wrap and the message reads This is where you type second page message.

<0x01>**Z69**<0x02>Ap<0x1B>0b<0x0A>1<0x1C>1 This is where you type first page message <0x04>

This Simple TEXT example above writes a message to File “p” on the default Drive in the default Folder with the mode command displaying the stationary message in the center (horizontal) of the display in red text.

<0x01>**Z49**<0x02>AQ<0x1B>0w<0x0E>10<0x1C>1 This is where your message <0x04>

This Simple TEXT example above writes a message to File “Q” on the default Drive in the default Folder with the mode command displaying the in-coming message as rain in fill on the display in red text. When the entire message appears it will stay for ten (10) seconds.

<0x01>**Z98**<0x02>A<0x0F>ETAU<0x1B>0a<0x19><0x1C>1 This is where you type message <0x04>

This Expanded TEXT example above writes a message to File “AU” on the “E” Drive in the “T” Folder with the mode command displaying the rotating message in red text at the fastest speed reading This is where you type message.

<0x01>**Z01**<0x02>Gb This is where you type first page message <0x04>

This Simple STRING example above writes a message to File “b” on the default Drive in the default Folder.

<0x01>**Z09**<0x02>G<0x0F>DSab This is where you type first page message <0x04>

This Expanded STRING example above writes a message to File “ab” on the “D” Drive in the “S” Folder.

<0x01>**Z00**<0x02>AA<0x10>b This is where you type message <0x04>

This Simple TEXT example above writes a message to File “A” on the default Drive in the default Folder and calls STRING file “b”.

#### PLEASE NOTE:

- 1) No carriage return spaces need to be between the selection attribute and the beginning of your message or between the end of your message and the “End of Transmission” (“H” in Table 2 ). Also format characters are explained in Table 2 and throughout this manual.
- 2) The character color and the background color must not be the same, otherwise your message will be altered and certain variables can not be accessed on certain 4U2SEE displays
- 3) The digits (made bold) representing the sign address (item C in tables 2 or 7) in the above examples, illustrate the versatility of this addressing structure. In order to deviate from the broadcast (global) address (00), your display board must have it’s Group & Unit Number (GGUU) the same as your commands sent by your “Host” computer. To accomplish this use Sigma 3000 software (previously mentioned Electro-Matic input device). For more information, please refer to Electro-Matic’s Sigma 3000 manual, posted on the portal.
- 4) The ASCII character “Z” (item B in tables 2 or 7) can be replaced with any other valid ASCII character.
- 5) The example for “Middle” has a carriage return (space) between the Mode (“<0x1B>”) command and the rotate command (“a”). Specifying “a” will automatically create a No Wrap situation.
- 6) If a Mode (“<0x1B>”) is not specified then display feature will resort to default settings.
- 7) FlexSender (Electro-Matic input device) does not acknowledge the ASCII coding for “Top”, you must utilize hexadecimal code of <0x22> when using this format.  
Example: <0x01>**Z00**<0x02>AA<0x1B><0x22>aThis is where you type message<0x04>
- 8) If an Expanded STRING Write command (“G”) is used then the STRING Folder must be a (“S”).
- 9) If a Simple STRING Write command (“G”) is used then the STRING Folder must be a default (“T”).
- 10) If an Expanded or Simple TEXT Write command (“A”) is used then the TEXT Folder must be a default (“T”).
- 11) Expanded TEXT files are not applicable to 4U2SEE firmware versions prior to version 8E04.

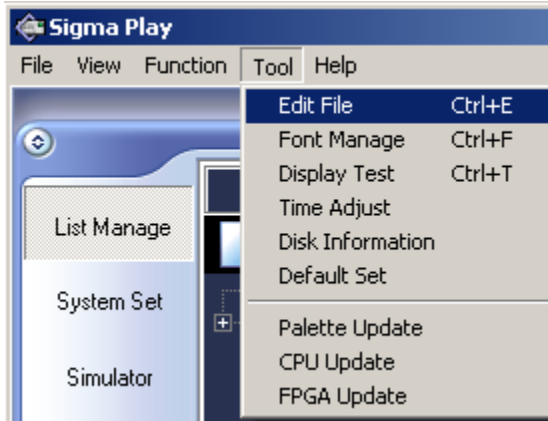
## Appendix A: Updating An Enhanced 4U2SEE With Firmware 8E04

**CAUTION:** The following procedure should only be performed by properly trained technicians. This procedure can only be utilized on enhanced display boards, utilizing Sigma 3000 Version 2.14 software, which can be purchased through Electro-matic Products Inc. Warranty will be voided if not performed by trained technicians.

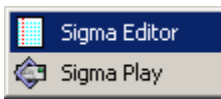
Checking the version of Sigma 3000:

There is multiple avenues to open 'Sigma Editor' and check on the version.

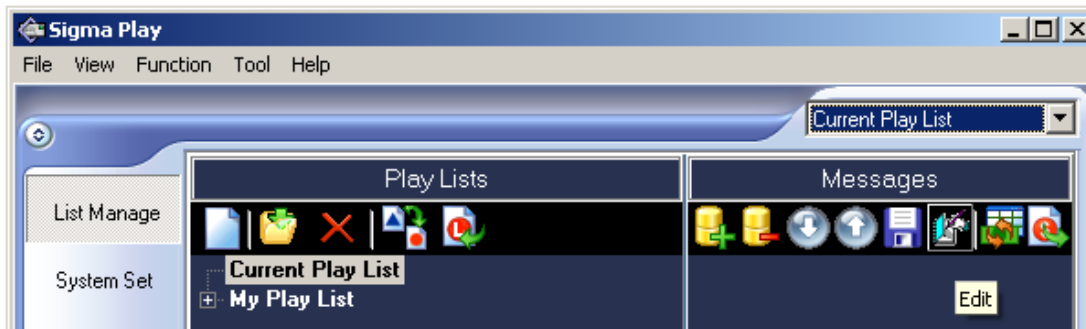
A) Open 'Sigma Play' > select 'Tool' > select 'Edit File' (as shown below)



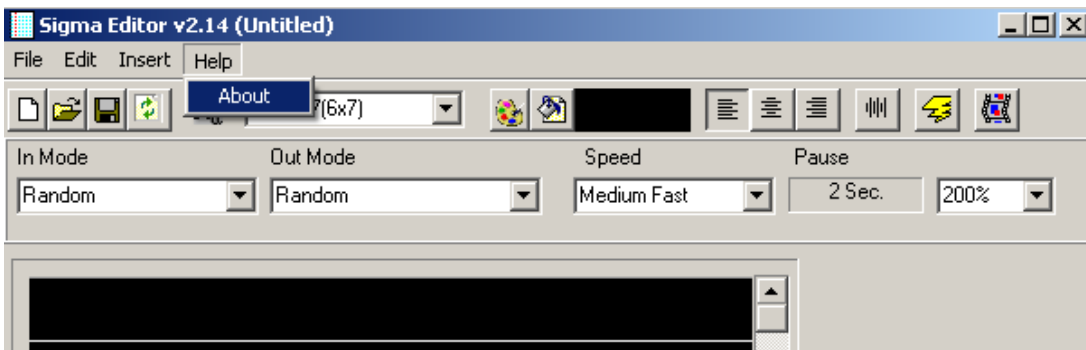
B) Open 'Start' < select 'Sigma 3000 (2.14)' <select 'Sigma Editor' (as shown below)



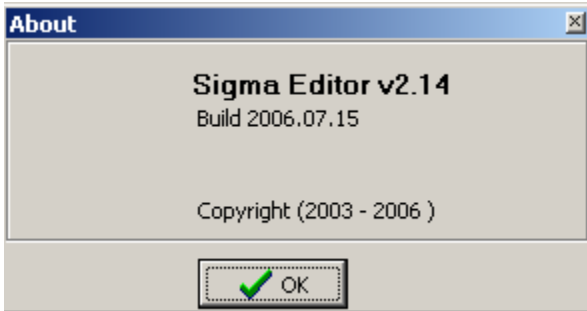
C) Open 'Sigma Play' > select 'Edit' icon (as shown below).



D) Once 'Sigma Editor' is open the version number will appear in the header (title – dark blue) or in the 'Help' pull-down; select 'Help' < select 'About' (as shown below).



E) When 'About' is chosen the dialog box below will appear (as shown below).

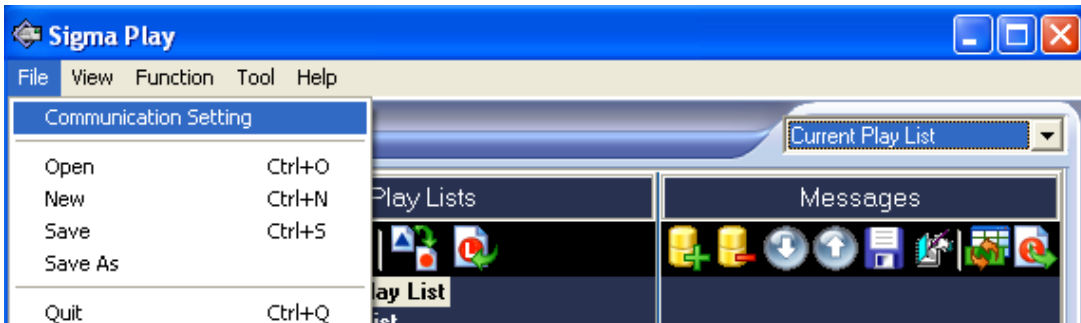


There are three (3) communication ports on each enhanced 4U2SEE sign. The communication ports can be classified as two (2) types, Ethernet and Serial (RS232 – gray connector RS-485 – black connector).

This document illustrates the loading of firmware based on an Ethernet connection, although a Serial connection can also be utilized (RS232 – gray connector is used to communicate with sign and the RS485 – black connector is used to operate slave signs, an Ethernet connection expedites the transfer of data. All of the following steps are required for both classifications, but some steps, which are referenced, are not required if a Serial connection is the elected means of communication.

To update, install or revise a version of 4U2SEE firmware 8E04 the following steps must be followed:

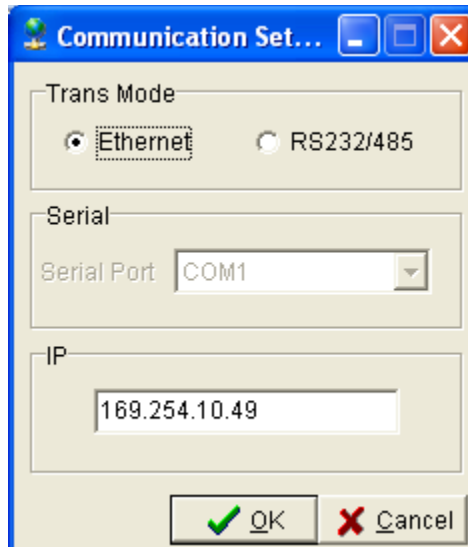
- 1) Install Sigma 3000 Version 2.14 (Sigma 3000 Version 1.0.0 ['Sigma Editor'] is not appropriate)
- 2) Cycle electrical power (initial users introduce electrical power) and copy down all information displayed on the sign. This system information will be utilized later in this document and should be archived for future use.
- 3) Open 'Sigma Play' (as illustrated above).
- 4) Once 'Sigma Play' is opened. You must configure your computer to the display < Select the 'File' tab < 'Communication Setting' (as shown below).



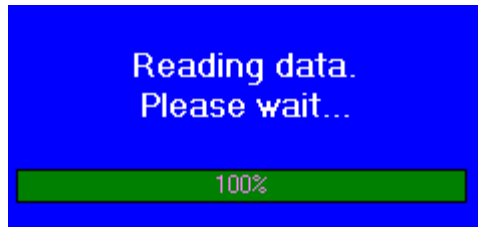
- 5) After 'Communication Setting' is selected. < choose 'Ethernet' selection < Set the proper IP (Internet Protocol) address of the display (viewed previously in step 2), so the display and 'Sigma 3000 Version 2.14' can effectively communicate (as shown below).

**PLEASE NOTE:**

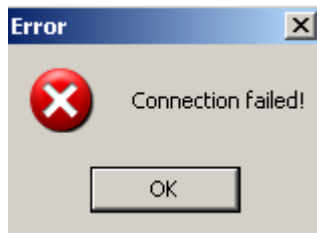
- A) The computer being used IP address must be in the same subnet mask range of the display.  
Example: the 'Host' computer IP address must be 169.254.10.XXX (based on next example).  
Where XXX can be any number from 1 to 254, except for 49 (used by the display board).
- B) The IP address shown in this illustration is the factory default setting, although the IP address can be altered to suit your specific networking needs – the factory default IP address is required to perform this procedure.
- C) If the 'Host' computer's Ethernet port is directly plugged into the display's Ethernet port, then a crossover Ethernet cable must be used.



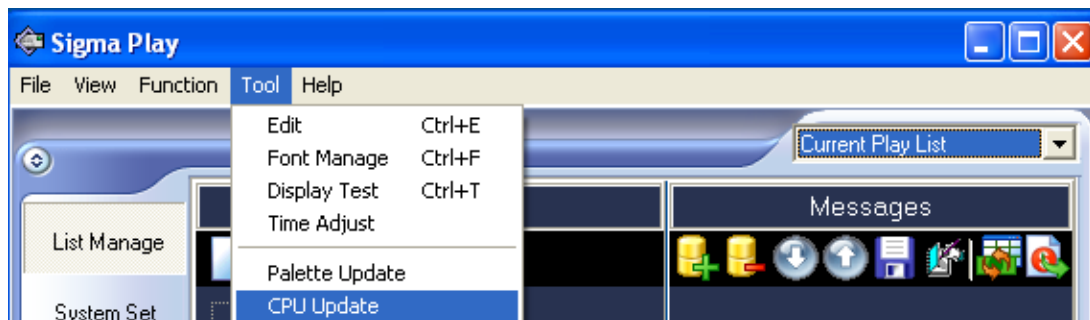
- 6) After 'OK' in the above illustration is selected and communication with the sign is established correctly, then a bar chart will appear (as shown below).



- 7) If connection fails then you will receive (as shown below).



- 8) When connection is successful, select 'Tool' < select 'CPU Update' (as shown below).

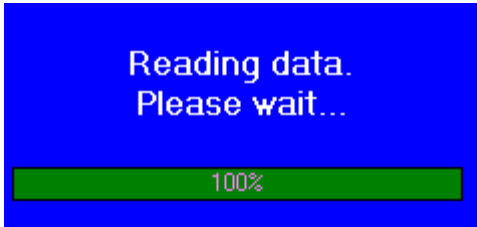


- 9) When 'CPU Update' is chosen, a file dialog box will appear. Allowing you to browse to search for the proper file. The file, which can be obtained from Electro-Matic Products Inc., will be named Sign\_VER YYYY .cpu. The YYYY represents the firmware version number and the cpu (following a period) is referred to as an file extension.



**PLEASE NOTE:**

- A) The little square box to the right of the input area (on the 'CPU Update' screen), can be used as a 'Browse' (search) button.
- 10) Once the file is located > double click the file, which populates the input field < then choose the 'Send' button (illustrated above). A bar chart will appear when sending the firmware file (as shown below).



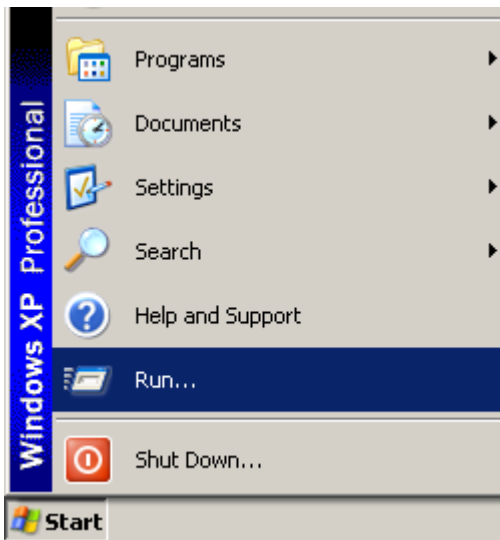
- 11) When file has been sent, you will receive a confirmation (as shown below).



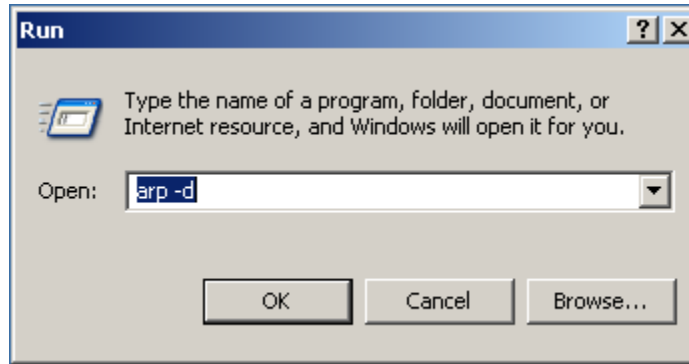
- 12) When the 'Information' dialog box (step 11) is received, please wait for the control board in the sign to make an audio response (beep).
- 13) After the audio response (beep) for successful transfer has been received, unplug from electricity, wait approximately thirty (30) seconds and introduce (plug in) electricity. At this point, you should receive an audio response (beep) from the sign – signifying the system is ready to proceed.

**PLEASE NOTE:**

- A) This next step (step 14) is not required if you've elected to accomplish the previous steps Serially. Continue with step 16.
- 14) When you receive the acknowledgement to proceed, select Windows 'Start' < select 'Run' (as shown below).



15) After selecting 'Run' enter 'arp -d' in the 'Open' field < then select 'OK' (as shown below).

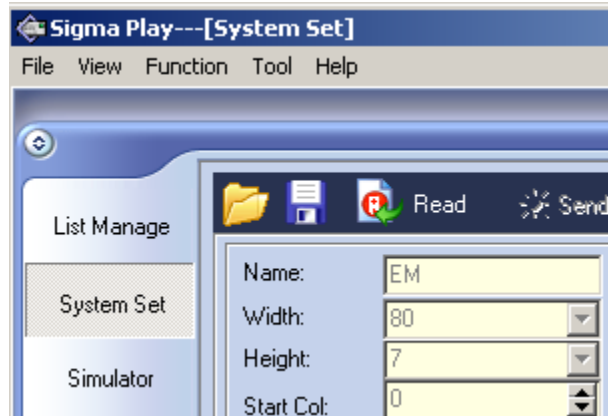


16) Re-establish communications with the display board > repeat steps 3 – 7.

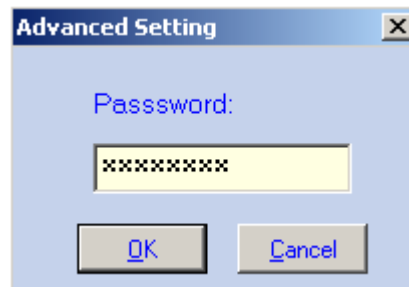
**PLEASE NOTE:**

A) The factory default IP address must be used for this step (reference the Please Note: [item B] affiliated with step 5).

17) Once communications are successfully obtained > select 'System Set' (as shown below).

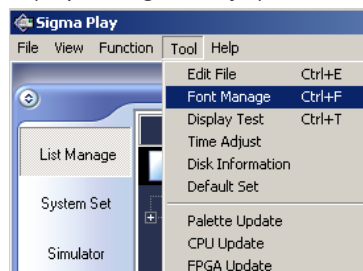


18) To edit (un-gray) variables in 'System Set' simultaneously press the Ctrl-Alt-A keys < enter password 12345678 in the password field (as shown below).



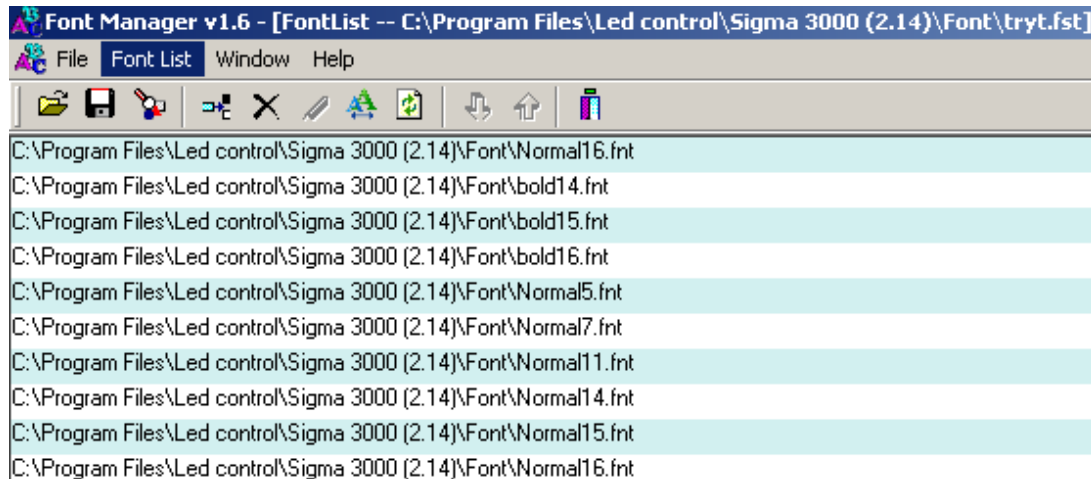
19) When variables in 'System Set' appear as blackened text, adjust the 'Width:' and 'Height:' to the appropriate settings (information copied in step 2)

20) Open 'Sigma Play' (as illustrated above) < select "Tool" < 'Font Manage' (as shown below).





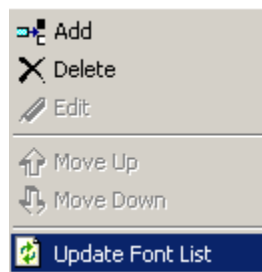
21) When 'Font Manager' is open < select 'Font List' (as shown below).



**PLEASE NOTE:**

A) The above files will be installed in a newly created directory, once Sigma 3000 Version 2.14 is installed.

22) After 'Font List' is opened < select 'Update Font List' (as shown below).



23) A 'Communication Setting' dialog box will appear – follow steps 5-7. If successful execution of steps 5-7, then a 'Sending' bar chart (like step 10) will appear. When successful transmission an 'Information' dialog box (step 11) will appear.

24) Cycle power and System Information will appear on the display – you are ready to send messages to the display. Update your archival file established in step 2.

**CAUTION:** It is very important not to interrupt power during this procedure, unless specified otherwise. If any power interruptions occur, then the display may be unusable and non-repairable. Damaged 4U2SEE display boards can be shipped to Electro-Matic Products, Inc. in the attempt to salvage at a nominal fee.

## Appendix B: Delete Sequence

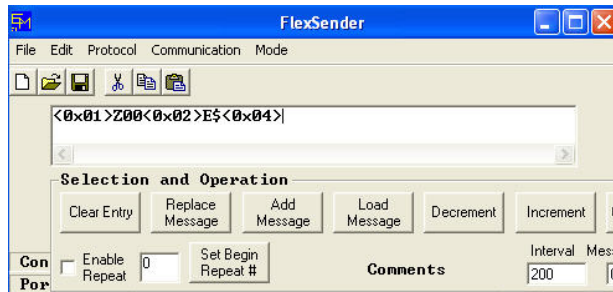
With the three (3) previously mentioned input methods, the following deletion instructions will allow transmission of other messages.

1. The first method is with the remote control. This procedure is outlined below.

**Table 9: Deleting Sequence**

Step	Function	Display Will Read...
1	Press "PROGRAM"	File Operate
2	Press "ENTER"	Delete File
3	Press "DOWN" arrow until it gets to...	Delete Sequence
4	Press "ENTER"	DelSure?(Y/N) (Choices will blink when selected)
5	Use your "LEFT" and "RIGHT" arrows to select either Y=Yes or N=No	DelSure?(Y/N) (Choices will blink when selected)
6	Press "ESC" until you get back to the display mode	
7	Press "ENTER"	
8	Press "ESC" until you get back to the display	

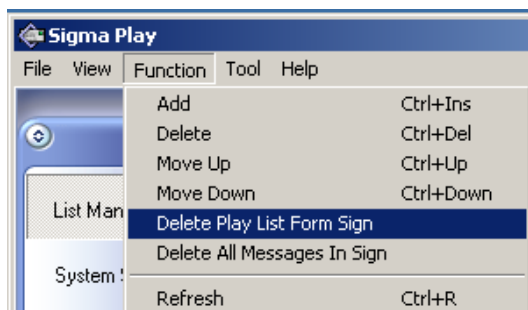
2. The second method is utilizing FlexSender software. Enter <0x01>Z00<0x02>E\$<0x04> in the message input field. This command will delete **ALL** files on the display, as referenced in Section 6.3.13 and throughout this manual.



**PLEASE NOTE:** All files volatile and non-volatile messages will be deleted, when this FlexSender code is used.

**PLEASE NOTE:** If deletion of current play list (sequence) is required reference Section 6.3.13.

3. The third method is utilizing Sigma 3000 software and click on the "Function" tab and then click on the "Delete Play List From Sign".



**PLEASE NOTE:** All devices and software methods can be purchased from Electro-Matic Products, Inc. Methods 2 & 3 (FlexSender and Sigma 3000) must have proper communications between the "host" computer, operating the software and the display.

## Appendix C: Expanded & Single Valid File Labels

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.

### Valid 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-" "
2DH-"-"	3DH-"="	4DH-"M"	5DH-"J"	6DH-"m"	7DH-"}"
2EH-"."	3EH-">"	4EH-"N"	5EH-"€"	6EH-"n"	
2FH-"/"	3FH-"?"	4FH-"O"	5FH-"_"	6FH-"o"	
NOTE: sp=space					

**PLEASE NOTE:** Single valid files constitute ninety-four (94) valid file labels and expanded valid files constitute eight thousand eight hundred and forty-six (8846) valid files. This is calculated by multiplying ninety-four by ninety-four (94x94).

**PLEASE NOTE:** Expanded files are designated with <0x15>x, where x can be any valid file label.

**PLEASE NOTE:** See Section 6.1.2 for examples of expanded and single valid file labels.

## Appendix D: Examples of FlexSender Code

```

1) <0x01>Z00<0x02>AArandom-displays all modes in center<0x04>
2) <0x01>Z00<0x02>AA<0x1B>0arotates in center<0x04>
3) <0x01>Z00<0x02>AA<0x1B>0bholds in center(horizontal)<0x04>
4) <0x01>Z00<0x02>AA<0x1B>1bholds on top (horizontal)<0x04>
5) <0x01>Z00<0x02>AA<0x1B>2bholds on bottom (horizontal)<0x04>
6) <0x01>Z00<0x02>AA<0x1B>3arotates in fill (horizontal)<0x04>
7) <0x01>Z00<0x02>AA<0x1B>0arotates in center<0x04>
8) <0x01>Z00<0x02>AA<0x1B>3bholds in fill (horizontal)<0x04>
9) <0x01>Z00<0x02>AA<0x0a>I-in mode auto (stationary)<0x04>
10) <0x01>Z00<0x02>AA<0x0a>I1in mode move left<0x04>
11) <0x01>Z00<0x02>AA<0x0a>I2in mode move right<0x04>
12) <0x01>Z00<0x02>AA<0x0a>I3in mode scroll o/l<0x04>
13) <0x01>Z00<0x02>AA<0x0a>I4in mode scroll o/r<0x04>
14) <0x01>Z00<0x02>AA<0x0a>I5in mode move up<0x04>
15) <0x01>Z00<0x02>AA<0x0a>I6in mode move down<0x04>
16) <0x01>Z00<0x02>AA<0x0a>I7in mode scroll o/c<0x04>
17) <0x01>Z00<0x02>AA<0x0a>I8in mode unveil up<0x04>
18) <0x01>Z00<0x02>AA<0x0a>I9in mode unveil down<0x04>
19) <0x01>Z00<0x02>AA<0x0a>IAin mode veretian hor<0x04>
20) <0x01>Z00<0x02>AA<0x0a>IBin mode veretian vert<0x04>
21) <0x01>Z00<0x02>AA<0x0a>ICin mode rain<0x04>
22) <0x01>Z00<0x02>AA<0x0a>IDin mode materialize<0x04>
23) <0x01>Z00<0x02>AA<0x0a>IEin mode twinkle<0x04>
24) <0x01>Z00<0x02>AA<0x0a>IFin mode squigle<0x04>
25) <0x01>Z00<0x02>AA<0x0a>IGin mode radar<0x04>
26) <0x01>Z00<0x02>AA<0x0a>IHin mode fanopen<0x04>
27) <0x01>Z00<0x02>AA<0x0a>Ilin mode fanclose<0x04>
28) <0x01>Z00<0x02>AA<0x0a>Ijin mode rotateright<0x04>
29) <0x01>Z00<0x02>AA<0x0a>IKin mode rotateleft<0x04>
30) <0x01>Z00<0x02>AA<0x0a>ILin mode center2corner<0x04>
31) <0x01>Z00<0x02>AA<0x0a>IMin mode corner2center<0x04>
32) <0x01>Z00<0x02>AA<0x0a>INin mode center2allsz<0x04>
33) <0x01>Z00<0x02>AA<0x0a>IOin mode allsz2center<0x04>
34) <0x01>Z00<0x02>AA<0x0a>IPin mode fourblock2cor<0x04>
35) <0x01>Z00<0x02>AA<0x0a>IQin mode fourblock2cen<0x04>
36) <0x01>Z00<0x02>AA<0x0a>IRin mode fourblockout<0x04>
37) <0x01>Z00<0x02>AA<0x0a>ISin mode fourblockin<0x04>
38) <0x01>Z00<0x02>AA<0x0a>ITin mode leftcorrectin<0x04>
39) <0x01>Z00<0x02>AA<0x0a>IUin mode RightCoRectIn<0x04>
40) <0x01>Z00<0x02>AA<0x0a>IVin mode LbottomRectl<0x04>
41) <0x01>Z00<0x02>AA<0x0a>IWin mode RBottomRectl<0x04>
42) <0x01>Z00<0x02>AA<0x0a>IXin mode LftCoDiagonal<0x04>
43) <0x01>Z00<0x02>AA<0x0a>IYin mode RtCoDiagonal<0x04>
44) <0x01>Z00<0x02>AA<0x0a>IZin mode LBtmDiagonal<0x04>
45) <0x01>Z00<0x02>AA<0x0a>Ijin mode RBtmDiagonal<0x04>
46) <0x01>Z00<0x02>AA<0x0a>I\in mode Lft2RtDownCor<0x04>
47) <0x01>Z00<0x02>AA<0x0a>I]in mode Rt2LftDownCor<0x04>
48) <0x01>Z00<0x02>AA<0x0a>I^in mode Left2RtUpCor<0x04>
49) <0x01>Z00<0x02>AA<0x0a>I_in mode Rt2LftUpCor<0x04>
50) <0x01>Z00<0x02>AA<0x0a>O-out mode auto (stationary)<0x04>
51) <0x01>Z00<0x02>AA<0x0a>O1out mode move left<0x04>
52) <0x01>Z00<0x02>AA<0x0a>O2out mode move right<0x04>
53) <0x01>Z00<0x02>AA<0x0a>O3out mode scroll o/l<0x04>
54) <0x01>Z00<0x02>AA<0x0a>O4out mode scroll o/r<0x04>
55) <0x01>Z00<0x02>AA<0x0a>O5out mode move up<0x04>
56) <0x01>Z00<0x02>AA<0x0a>O6out mode move down<0x04>
57) <0x01>Z00<0x02>AA<0x0a>O7out mode scroll o/c<0x04>
58) <0x01>Z00<0x02>AA<0x0a>O8out mode unveil up<0x04>
59) <0x01>Z00<0x02>AA<0x0a>O9out mode unveil down<0x04>
60) <0x01>Z00<0x02>AA<0x0a>OAout mode veretian hor<0x04>

```

```

61) <0x01>Z00<0x02>AA<0x0a>OBout mode veretian vert<0x04>
62) <0x01>Z00<0x02>AA<0x0a>OCout mode rain<0x04>
63) <0x01>Z00<0x02>AA<0x0a>ODout mode materialize<0x04>
64) <0x01>Z00<0x02>AA<0x0a>OEout mode twinkle<0x04>
65) <0x01>Z00<0x02>AA<0x0a>OFout mode squigle<0x04>
66) <0x01>Z00<0x02>AA<0x0a>OGout mode radar<0x04>
67) <0x01>Z00<0x02>AA<0x0a>OHout mode fanopen<0x04>
68) <0x01>Z00<0x02>AA<0x0a>Olout mode fanclose<0x04>
69) <0x01>Z00<0x02>AA<0x0a>OJout mode rotateright<0x04>
70) <0x01>Z00<0x02>AA<0x0a>OKout mode rotateleft<0x04>
71) <0x01>Z00<0x02>AA<0x0a>OLout mode center2corner<0x04>
72) <0x01>Z00<0x02>AA<0x0a>OMout mode corner2center<0x04>
73) <0x01>Z00<0x02>AA<0x0a>ONout mode center2allsz<0x04>
74) <0x01>Z00<0x02>AA<0x0a>OOout mode allsz2center<0x04>
75) <0x01>Z00<0x02>AA<0x0a>OPout mode fourblock2cor<0x04>
76) <0x01>Z00<0x02>AA<0x0a>OQout mode fourblock2cen<0x04>
77) <0x01>Z00<0x02>AA<0x0a>ORout mode fourblockout<0x04>
78) <0x01>Z00<0x02>AA<0x0a>OSout mode fourblockin<0x04>
79) <0x01>Z00<0x02>AA<0x0a>OTout mode leftcorrectin<0x04>
80) <0x01>Z00<0x02>AA<0x0a>OUout mode RightCoRectIn<0x04>
81) <0x01>Z00<0x02>AA<0x0a>OVout mode LbottomRectl<0x04>
82) <0x01>Z00<0x02>AA<0x0a>OWout mode RBottomRectl<0x04>
83) <0x01>Z00<0x02>AA<0x0a>OXout mode LftCoDiagonal<0x04>
84) <0x01>Z00<0x02>AA<0x0a>OYout mode RtCoDiagonal<0x04>
85) <0x01>Z00<0x02>AA<0x0a>OZout mode LBtmDiagonal<0x04>
86) <0x01>Z00<0x02>AA<0x0a>O[out mode RBtmDiagonal<0x04>
87) <0x01>Z00<0x02>AA<0x0a>O\out mode Lft2RtDownCor<0x04>
88) <0x01>Z00<0x02>AA<0x0a>O]out mode Rt2LftDownCor<0x04>
89) <0x01>Z00<0x02>AA<0x0a>O^out mode Left2RtUpCor<0x04>
90) <0x01>Z00<0x02>AA<0x0a>O_in mode Rt2LftUpCor<0x04>
91) <0x01>Z00<0x02>AA<0x1e>0center justify<0x04>
92) <0x01>Z00<0x02>AA<0x1e>1left justify<0x04>
93) <0x01>Z00<0x02>AA<0x1e>2right justify<0x04>
94) <0x01>Z00<0x02>AA<0x0f>0fastest speed<0x04>
95) <0x01>Z00<0x02>AA<0x0f>1fast speed<0x04>
96) <0x01>Z00<0x02>AA<0x0f>2medium fast speed<0x04>
97) <0x01>Z00<0x02>AA<0x0f>3medium speed<0x04>
98) <0x01>Z00<0x02>AA<0x0f>4medium slow speed<0x04>
99) <0x01>Z00<0x02>AA<0x0f>5slow speed<0x04>
100) <0x01>Z00<0x02>AA<0x0f>6slowest speed<0x04>
101) <0x01>Z00<0x02>AA<0x0b>1<0x04>
102) <0x01>Z00<0x02>AA<0x0b>2<0x04>
103) <0x01>Z00<0x02>AA<0x0b>3<0x04>
104) <0x01>Z00<0x02>AA<0x0b>4<0x04>
105) <0x01>Z00<0x02>AA<0x07>0flash off<0x04>
106) <0x01>Z00<0x02>AA<0x07>1flash on<0x04>
107) <0x01>Z00<0x02>AA<0x1a>0font normal 5<0x04>
108) <0x01>Z00<0x02>AA<0x1a>1font normal 7<0x04>
109) <0x01>Z00<0x02>AA<0x1a>2font normal 14<0x04>
110) <0x01>Z00<0x02>AA<0x1a>3font normal 15<0x04>
111) <0x01>Z00<0x02>AA<0x1a>4font normal 16<0x04>
112) <0x01>Z00<0x02>AA<0x1a>5font bold 14 (5 on font list)<0x04>
113) <0x01>Z00<0x02>AA<0x1C>1color-red<0x04>
114) <0x01>Z00<0x02>AA<0x1C>2color-green<0x04>
115) <0x01>Z00<0x02>AA<0x1C>3color-yellow<0x04>
116) <0x01>Z00<0x02>AA<0x1C>4color-mix color1<0x04>
117) <0x01>Z00<0x02>AA<0x1C>5color-mix color2<0x04>
118) <0x01>Z00<0x02>AA<0x1C>6color-mix color3<0x04>
119) <0x01>Z00<0x02>AA<0x1C>7color-mix color4<0x04>
120) <0x01>Z00<0x02>AA<0x1C>8color-black<0x04>
121) <0x01>Z00<0x02>AA<0x1D>B1backcolor-red<0x04>
122) <0x01>Z00<0x02>AA<0x1D>B2backcolor-green<0x04>
123) <0x01>Z00<0x02>AA<0x1D>B3backcolor-yellow<0x04>

```

```

124) <0x01>Z00<0x02>AA<0x1C>3<0x1D>B3fore & back color<0x04>
125) <0x01>Z00<0x02>AA<0x1a>2<0x1C>1<0x1D>B1text-fore & back color<0x04>
126) <0x01>Z00<0x02>AA<0x1C>3<0x0a>l2yellow color-move left<0x04>
127) <0x01>Z00<0x02>AA<0x1C>3<0x0a>l2<0x0a>ODcolor-in-out moving<0x04>
128) <0x01>Z00<0x02>AA<0x1B>0b<0x1D>B3hold-backgrd test<0x04>
129) <0x01>Z00<0x02>AA<0x0b>0<0x04>
130) <0x01>Z00<0x02>AA<0x0b>1<0x04>
131) <0x01>Z00<0x02>AA<0x0b>2<0x04>
132) <0x01>Z00<0x02>AA<0x0b>3<0x04>
133) <0x01>Z00<0x02>AA<0x0b>4<0x04>
134) <0x01>Z00<0x02>AA<0x0b>8<0x04>
135) <0x01>Z00<0x02>AA<0x0b>9<0x04>
136) <0x01>Z00<0x02>AA<0x1B>0bGoal:<0x10>A Actual:<0x10>B<0x04>
137) <0x01>Z00<0x02>GA500<0x04>
138) <0x01>Z00<0x02>GB400<0x04>
139) <0x01>Z00<0x02>GB401<0x04>
140) <0x01>Z00<0x02>GB402<0x04>
141) <0x01>Z00<0x02>GB403<0x04>
142) <0x01>Z00<0x02>GB404<0x04>
143) <0x01>Z00<0x02>GB405<0x04>
144) <0x01>Z00<0x02>GB406<0x04>
145) <0x01>Z00<0x02>GB407<0x04>
146) <0x01>Z00<0x02>GB408<0x04>
147) <0x01>Z00<0x02>GB409<0x04>
148) <0x01>Z00<0x02>GB410<0x04>
149) <0x01>Z00<0x02>E$<0x04>
150) <0x01>Z00<0x02>AA<0x0a>la in mode unveil in <0x04>
151) <0x01>Z00<0x02>AA<0x0a>lb in mode unveil up/in <0x04>
152) <0x01>Z00<0x02>AA<0x0a>lc in mode unveil up/out <0x04>
153) <0x01>Z00<0x02>AA<0x0a>ld in mode splice across <0x04>
154) <0x01>Z00<0x02>AA<0x0a>le in mode splice verti <0x04>
155) <0x01>Z00<0x02>AA<0x0a>lf in mode fall left <0x04>
156) <0x01>Z00<0x02>AA<0x0a>lg in mode fall right <0x04>
157) <0x01>Z00<0x02>AA<0x0a>lh in mode grow up <0x04>
158) <0x01>Z00<0x02>AA<0x0a>Oa out mode unveil in <0x04>
159) <0x01>Z00<0x02>AA<0x0a>Ob out mode unveil up/in <0x04>
160) <0x01>Z00<0x02>AA<0x0a>Oc out mode unveil up/out <0x04>
161) <0x01>Z00<0x02>AA<0x0a>Od out mode splice across <0x04>
162) <0x01>Z00<0x02>AA<0x0a>Oe out mode splice verti <0x04>
163) <0x01>Z00<0x02>AA<0x0a>Of out mode fall left <0x04>
164) <0x01>Z00<0x02>AA<0x0a>Og out mode fall right <0x04>
165) <0x01>Z00<0x02>AA<0x0a>Oh out mode grow up <0x04>
166) <0x01>Z00<0x02>AA<0x0b>5<0x04>
167) <0x01>Z00<0x02>AA<0x0b>6<0x04>
168) <0x01>Z00<0x02>AA<0x0b>7<0x04>
169) <0x01>Z00<0x02>AA<0x0b>A<0x04>
170) <0x01>Z00<0x02>AA<0x0b>B<0x04>
171) <0x01>Z00<0x02>AA<0x0b>C<0x04>
172) <0x01>Z00<0x02>AA<0x0b>D<0x04>
173) <0x01>Z00<0x02>AA<0x0b>E<0x04>
174) <0x01>Z00<0x02>AA<0x0b>F<0x04>
175) <0x01>Z00<0x02>AA<0x0b>a<0x04>
176) <0x01>Z00<0x02>AA<0x0b>b<0x04>
177) <0x01>Z00<0x02>AA<0x1D>B4 background-black<0x04>
178) <0x01>Z00<0x02>E#B1<0x04>
179) <0x01>Z00<0x02>E#B2<0x04>
180) <0x01>Z00<0x02>E#B3<0x04>
181) <0x01>Z00<0x02>E#B4<0x04>
182) <0x01>Z00<0x02>E#C1<0x04>
183) <0x01>Z00<0x02>E#C2<0x04>
184) <0x01>Z00<0x02>E#C3<0x04>
185) <0x01>Z00<0x02>E#C4<0x04>

```



```

186) <0x01>Z00<0x02>E#DE<0x04>
187) <0x01>Z00<0x02>E#DD<0x04>
188) <0x01>Z00<0x02>E#F0<0x04>
189) <0x01>Z00<0x02>E#F1<0x04>
190) <0x01>Z00<0x02>E#F2<0x04>
191) <0x01>Z00<0x02>E#F3<0x04>
192) <0x01>Z00<0x02>E#F4<0x04>
193) <0x01>Z00<0x02>E#F5<0x04>
194) <0x01>Z00<0x02>E#F6<0x04>
195) <0x01>Z00<0x02>E#F7<0x04>
200) <0x01>Z00<0x02>E#HC<0x04>
201) <0x01>Z00<0x02>E#HT<0x04>
202) <0x01>Z00<0x02>E#HB<0x04>
203) <0x01>Z00<0x02>E#HF<0x04>
204) <0x01>Z00<0x02>E#L1<0x04>
205) <0x01>Z00<0x02>E#L2<0x04>
206) <0x01>Z00<0x02>E#L3<0x04>
207) <0x01>Z00<0x02>E#L4<0x04>
208) <0x01>Z00<0x02>E#L5<0x04>
209) <0x01>Z00<0x02>E#L6<0x04>
210) <0x01>Z00<0x02>E#L7<0x04>
211) <0x01>Z00<0x02>E#L8<0x04>
212) <0x01>Z00<0x02>E#L9<0x04>
213) <0x01>Z00<0x02>E#S1<0x04>
214) <0x01>Z00<0x02>E#S2<0x04>
215) <0x01>Z00<0x02>E#S3<0x04>
216) <0x01>Z00<0x02>E#S4<0x04>
217) <0x01>Z00<0x02>E#S5<0x04>
218) <0x01>Z00<0x02>E#S6<0x04>
219) <0x01>Z00<0x02>E#S7<0x04>
220) <0x01>Z00<0x02>E#T1<0x04>
221) <0x01>Z00<0x02>E#T2<0x04>
222) <0x01>Z00<0x02>E#T3<0x04>
223) <0x01>Z00<0x02>E#T4<0x04>
224) <0x01>Z00<0x02>E#T5<0x04>
225) <0x01>Z00<0x02>E#T6<0x04>
226) <0x01>Z00<0x02>E#T7<0x04>
227) <0x01>Z00<0x02>E#T8<0x04>
228) <0x01>Z00<0x02>E#T9<0x04>
229) <0x01>Z00<0x02>E#VR<0x04>
230) <0x01>Z00<0x02>E#VL<0x04>
231) <0x01>Z00<0x02>E#VC<0x04>
232) <0x01>Z00<0x02>E#W0<0x04>
233) <0x01>Z00<0x02>E#W1<0x04>
234) Please Note: Reference table ___ for a complete listing and examples of customizing default In
and Out Modes
235) <0x01>Z00<0x02>AA Enter text here - this is basic single file label <0x04>
236) <0x01>Z98<0x02>A/ Enter text here - this is basic single file label <0x04>
237) <0x01>Z22<0x02>aDB Enter text here - this is basic single file label <0x04>
238) <0x01>Z86<0x02>aE* Enter text here - this is basic single file label <0x04>
239) <0x01>Z11<0x02>GA Enter text here - this is basic single file label <0x04>
240) <0x01>Z66<0x02>G6 Enter text here - this is basic single file label <0x04>
241) <0x01>Z32<0x02>gEw Enter text here - this is basic single file label <0x04>
242) <0x01>Z96<0x02>gD$ Enter text here - this is basic single file label <0x04>
243) <0x01>Z00<0x02>AB<0x15>5 Enter text here - this is an expanded file label <0x04>
244) <0x01>Z98<0x02>A/<0x15>C Enter text here - this is an expanded file label <0x04>
245) <0x01>Z22<0x02>aDB<0x15>2 Enter text here - this is an expanded file label <0x04>
246) <0x01>Z86<0x02>aE6<0x15>A Enter text here - this is an expanded file label <0x04>
247) <0x01>Z11<0x02>GA<0x15>8 Enter text here - this is an expanded file label <0x04>
248) <0x01>Z66<0x02>G6<0x15>a Enter text here - this is an expanded file label <0x04>
249) <0x01>Z32<0x02>gDw<0x15>G Enter text here - this is an expanded file label<0x04>
250) <0x01>Z96<0x02>gE9<0x15>6 Enter text here - this is an expanded file label<0x04>

```

```

251) <0x01>Z00<0x02>E#C1B4F1L1T1S5VCHFW1MdOdDD<0x04>
252) <0x01>Z00<0x02>E#<0x04>
253) <0x01>Z00<0x02>E#C2B2<0x04>
254) <0x01>Z00<0x02>E#C3B2F0L2T1S5VCHFW1MaObDE<0x04>
255) <0x01>Z00<0x02>E#C3B2F0L2MaObDE<0x04>
256) <0x01>Z00<0x02>AA Sample A <0x04>
257) <0x01>Z00<0x02>Aq Sample B <0x04>
258) <0x01>Z00<0x02>Ar<0x15>5 Sample C <0x04>
259) <0x01>Z00<0x02>GC Sample D <0x04>
260) <0x01>Z00<0x02>aDp Sample E <0x04>
261) <0x01>Z00<0x02>gEp<0x15>g Sample F <0x04>
262) <0x01>Z00<0x02>A/<0x15>a Sample G <0x04>
263) <0x01>Z00<0x02>aDJ Sample H <0x04>
264) <0x01>Z39<0x02>E.SLq/<0x15>aCr<0x15>5<0x04>
265) <0x01>Z39<0x02>E.SLJpA<0x04>
266) <0x01>Z23<0x02>E. <0x04>

```

### INDEX SUMMARY

Lines 1 – 8 are Mode Start commands and referenced in section 6.3.1  
 Lines 10-49, 150-157 are In Mode commands and referenced in section 6.3.2  
 Lines 50-90, 158-165 are Out Mode commands and referenced in section 6.3.2  
 Lines 91-93 are Vertical Justification commands and referenced in section 6.3.10  
 Lines 94-100 are Speed commands and can be referenced in section 6.3.6  
 Lines 101-104, 129-135, 166-176 are Date and Time Stamp commands and can be referenced in section 6.3.3  
 Lines 105-106 are Flash commands and can be referenced in section 6.3.1  
 Lines 107-112 are Font commands and can be referenced in section 6.3.7  
 Lines 113-120 are Color commands and can be referenced in section 6.3.8  
 Lines 121-123 & 177 are Background commands and can be referenced in section 6.3.9  
 Lines 124-128 are miscellaneous combination commands that can be referenced throughout  
 Lines 136-148 are example coding for STRING files and can be referenced 6.1.3  
 Line 149 is the Global Delete command, referenced throughout  
 Lines 178-234 are example Default commands, referenced in section 6.2  
 Lines 235-250 are examples of Expanded and Single Valid Files, referenced throughout and in Appendix –  
 Lines 251-255 are examples of Customizing Defaults and referenced in section 6.2  
 Lines 256-265 are examples of setting a Play List (sequence) and referenced in section 6.3.13  
 Line 266 Deletes a Play List (sequence) and referenced in section 6.3.13



## Appendix E: Display Mode Descriptions

**Table 8: Display Mode Descriptions**

Mode Option	FlexSender Code	Description
Auto	<0x01>Z00<0x02>AA<0x0a>I<0x04>	Stays stationary
Move Left	<0x01>Z00<0x02>AA<0x0a>I1<0x04>	Moves left to right
Move Right	<0x01>Z00<0x02>AA<0x0a>I2<0x04>	Moves right to left
Scroll O/L	<0x01>Z00<0x02>AA<0x0a>I3<0x04>	Scrolls out to left
Scroll O/R	<0x01>Z00<0x02>AA<0x0a>I4<0x04>	Scrolls out to right
Move up	<0x01>Z00<0x02>AA<0x0a>I5<0x04>	Moves upward
Move down	<0x01>Z00<0x02>AA<0x0a>I6<0x04>	Moves downward
Scroll O/C	<0x01>Z00<0x02>AA<0x0a>I7<0x04>	Scrolls out from center
Unveil up	<0x01>Z00<0x02>AA<0x0a>I8<0x04>	Unveils upwards
Unveil down	<0x01>Z00<0x02>AA<0x0a>I9<0x04>	Unveils downwards
Veretian hor	<0x01>Z00<0x02>AA<0x0a>IA<0x04>	Horizontal venetian blind
Veretian ver	<0x01>Z00<0x02>AA<0x0a>IB<0x04>	Vertical venetian blind
Rain	<0x01>Z00<0x02>AA<0x0a>IC<0x04>	Raining
Materialize	<0x01>Z00<0x02>AA<0x0a>ID<0x04>	Appear in dots
Twinkle	<0x01>Z00<0x02>AA<0x0a>IE<0x04>	Appears in stars
Squigle	<0x01>Z00<0x02>AA<0x0a>IF<0x04>	Appears in squiggle
Radar	<0x01>Z00<0x02>AA<0x0a>IG<0x04>	Appears as sonar
FanOpen	<0x01>Z00<0x02>AA<0x0a>IH<0x04>	Fan opening up
FanClose	<0x01>Z00<0x02>AA<0x0a>II<0x04>	Fan closing up
RotateRight	<0x01>Z00<0x02>AA<0x0a>IJ<0x04>	Rotation to right
RotateLeft	<0x01>Z00<0x02>AA<0x0a>IK<0x04>	Rotation to left
Center2Corner	<0x01>Z00<0x02>AA<0x0a>IL<0x04>	Displays center to corner
Corner2Center	<0x01>Z00<0x02>AA<0x0a>IM<0x04>	Displays corner to center
Center2Allsz	<0x01>Z00<0x02>AA<0x0a>IN<0x04>	All sides from center
Allsz2Center	<0x01>Z00<0x02>AA<0x0a>IO<0x04>	All sides to center
FourBlock2Cor	<0x01>Z00<0x02>AA<0x0a>IP<0x04>	Block from center
FourBlock2Cen	<0x01>Z00<0x02>AA<0x0a>IQ<0x04>	Block to center
FourBlockOut	<0x01>Z00<0x02>AA<0x0a>IR<0x04>	Block out of display
FourBlockIn	<0x01>Z00<0x02>AA<0x0a>IS<0x04>	Block in to display
LeftCorRectIn	<0x01>Z00<0x02>AA<0x0a>IT<0x04>	Left in from corner
RightCoRectIn	<0x01>Z00<0x02>AA<0x0a>IU<0x04>	Right in from corner
LbottomRectI	<0x01>Z00<0x02>AA<0x0a>IV<0x04>	Left in from bottom
RbottomRectI	<0x01>Z00<0x02>AA<0x0a>IW<0x04>	Right in from bottom
LftCoDiagonal	<0x01>Z00<0x02>AA<0x0a>IX<0x04>	Displays left diagonal
RtCoDiagonalI	<0x01>Z00<0x02>AA<0x0a>IY<0x04>	Displays right diagonal
LBtmDiagonal	<0x01>Z00<0x02>AA<0x0a>IZ<0x04>	Displays bottom left diagonal
RBtmDiagonal	<0x01>Z00<0x02>AA<0x0a>I[<0x04>	Displays bottom right diagonal
Lft2RtDownCor	<0x01>Z00<0x02>AA<0x0a>I\<0x04>	Left to right corner in downwards
Rt2LftDownCor	<0x01>Z00<0x02>AA<0x0a>I]<0x04>	Right to left corner in downwards
Left2RtUpCor	<0x01>Z00<0x02>AA<0x0a>I^<0x04>	Left to right corner in upwards
Rit2LftUpCor	<0x01>Z00<0x02>AA<0x0a>I_<0x04>	Right to left corner in upwards

## Appendix F: Hex-Character Summary

Hex	Character	Description
0	<0x00>	NUL (Table 2)
1	<0x01>	Start Of Header (SOH - Table 2)
2	<0x02>	Start Of Text (STX - Table 2)
3	<0x03>	End Of Text (ETX - Table 2)
4	<0x04>	End Of Transmission (EOT - Table 2)
5	Not used	N/A
6	Not used	N/A
7	<0x07>	Flash (section 6.1.4)
8	Not used	N/A
9	Not used	N/A
0A	<0x0A>	Horizontal Positioning
0B	<0x0B>	Stamp (section 6.4.2)
0C	<0x0C>	New Page (section 6.4.3)
0D	<0x0D>	New Line (section 6.4.4)
0E	<0x0E>	Hold (section 6.4.10)
0F	<0x0F>	Extended File Addressing (section 6)
10	<0x10>	Call STRING File (sections 6.1.2, 6.1.4-6.1.6)
11	Not used	Was In-Modes - Replaced with (<0x1B>I_
12	Not used	Was Out-Modes - Replaced with (<0x1B>O_)
13	Not used	Was Call Time - N/A
14	Not used	Was Call PICTURE - N/A
15	<0x15>	Slowest Speed (section 6.4.8)
16	<0x16>	Slow Speed (section 6.4.8)
17	<0x17>	Medium Speed (section 6.4.8)
18	<0x18>	Fast Speed (section 6.4.8)
19	<0x19>	Fastest Speed (section 6.4.8)
1A	<0x1A>	Fonts (section 6.4.9)
1B	<0x1B>	Start Mode (section 6)
1C	<0x1C>	Foreground Color (section 6.4.6)
1D	<0x1D>	Background Color (section 6.4.7)
1E	Not used	N/A
1F	Not used	N/A

## Change Log

---

### 4/11/11 Changes made to ADP Protocol • Firmware 9721-01 or higher

- 6.4.1 – (Character Flash Command Code) Replaced previous with new version for firmware: 9721-01.
- 6.4.13 – (Embed Time of Day into Messages) Added new command
- 6.4.14 – (Embed Date into Messages) Added new command
- 6.4.15 – (Embed a string file within text) Added new command
- Page 30 (Note) Please note: improvements to string file update with firmware 9721-01
- 6.5 (Write Special Function) New special command added “E”
- 6.5.1 (Set Time of Day) Added new command
- 6.5.2 (Clear RAM Memory) Added new command
- 6.5.3 (Clear Flash and RAM memory) Added new command
- 6.5.4 (Set Day of Week) Added new command
- 6.5.5 (Set Time Format) Added new command
- 6.5.6 (Set Date) Added new command



