

SPECIFICATION

FOR THE

STATIC CONTROLS CORPORATION

Enhanced SCC Multi-Use Protocol Version 2.00

REVISIONS					DRAWN	DATE	STATIC CONTROLS CORPORATION		
REV	ECO	BY	DATE	APP	G. Robertson R. Schupbach D. Schmidt	8/7/02	WIXOM, MICHIGAN		
X0	ORIGINAL RELEASE	G.R.	6/4/97		CHECKED	DATE	(248) 926-4400		
X1	MOD AC1 CMD	G.R.	6/19/97				TITLE		
A0	MOD 3.2 CMDS	R.S.	10/3/97		DESIGN ENGINEER	DATE	SPECIFICATION		
A1	MOD 3.2 CMDS	R.S.	10/6/97				Multi-Use Protocol Version 2.00		
A2	MOD A.2 EXAMPLE	R.S.	1/14/98		APPROVED	DATE	SIZE	DRAWING NUMBER	REV
A3	See revision history ¶ 1.1	G.R.	8/25/98				A	MN-0974-200	A9
A4	See revision history ¶ 1.1	R.S.	1/14/99				NEXT ASSY.	SCALE	SHEET
A5	See revision history ¶ 1.1	G.R.	7/13/00				N/A	NONE	1 of 48
A6	See revision history ¶ 1.1	R.S.	7/20/00						
A7	See revision history ¶ 1.1	R.S.	8/11/00						
A8	See revision history ¶ 1.1	D.S.	5/17/02						
A9	See revision history ¶ 1.1	R.S.	8/7/02						

TABLE OF CONTENTS

Sheet Number

1.0	OVERVIEW.....	4
1.1	REVISION HISTORY.....	4
1.2	SCOPE.....	5
1.3	RELATED DOCUMENTS.....	5
2.0	GENERAL COMMAND STRUCTURE	6
2.1	COMMAND PACKET: INITIATION FIELD (STX).....	6
2.2	COMMAND PACKET: GROUP AND UNIT ADDRESS FIELDS.....	6
2.3	COMMAND PACKET: DATA FIELD.....	7
2.4	COMMAND PACKET: REQUEST SLAVE RESPONSE FIELD (ENQ).....	8
2.5	COMMAND PACKET: TERMINATION FIELD(S) (EOT OR ETX-CHECKSUM).....	8
3.0	HARDWARE SPECIFIC COMMAND STRUCTURE.....	9
3.1	COMMANDS FOR SCC FIRST GENERATION HARDWARE	9
3.1.1	M-950 Marquee Display Control CPU.....	9
3.1.2	M-951 Marquee Display Control CPU.....	9
3.1.3	M-973-A Display Control CPU.....	9
3.1.4	M-973-B Display Control CPU.....	9
3.1.5	M-973-C Display Control CPU.....	9
3.1.6	M-973-D Display Control CPU.....	9
3.1.7	M-973-E Display Control CPU.....	9
3.1.8	M-973-F Display Control CPU.....	9
3.1.9	M-973-G Display Control CPU.....	10
3.1.10	M-973-H Display Control CPU.....	10
3.2	COMMANDS FOR SCC SECOND GENERATION HARDWARE	10
3.3	NAB-6 DISPLAY CONTROL CPU: HARDWARE SETUP	10
3.3.1	System Configuration Commands	10
3.3.1.1	ESC Sequence Commands	11
3.3.1.2	Hardware Setup: Set GROUP and UNIT Addresses.....	11
3.3.1.3	Hardware Setup: Set Fader Level.....	12
3.3.1.4	Hardware Setup: Redisplay Boot Time Information.....	12
3.3.1.5	Hardware Setup: Set Board Type	12
3.3.1.6	Hardware Setup: Set Auto Center Option	13
3.4	NAB-6 DISPLAY CPU: 974 ANDON CELL CONTROL	13
3.4.1	Andon Cell Control: AC1 variant	13
3.4.2	Andon Cell Control: AC2 variant	14
3.4.3	Andon Cell Control: AW1 command	16
3.5	NAB-6 DISPLAY CPU: 952 & 954 ALPHA-NUMERIC MARQUEE CONTROL.....	17
3.6	NAB-6 DISPLAY CPU: MUSIC MAKER CONTROL	17
3.6.1	Music Maker Control: TU1, TU2 Commands.....	17
3.6.2	Music Maker Control: VL1 Command	20
3.7	NAB-6 DISPLAY CPU: 801 NUMERIC/TIME DISPLAY CONTROL	20
3.7.1	801 Numeric Display Control: NUM Command	20
3.7.2	801 Numeric Display Control: TIM Command	22
3.8	NAB-6 DISPLAY CPU: 955 TRI-COLOR ALPHA-NUMERIC MARQUEE CONTROL.....	23
3.8.1	TX1 Command: Text Command	23
3.8.2	RST Command: Restore.....	26
3.8.3	AD1 Command: Column Definition & Sizing	26
3.8.4	AW1 Command: Write Values and Attributes	27

3.8.5	AC1 Commands: Activate Attributes	31
3.8.6	CHR Command: Character Create	36
3.8.7	DEL Command : Delete Command.....	38
3.8.8	TS1 Command: Test Board	39
3.8.9	NRM overview: Normal State	39
3.8.10	CLR Command: Clear.....	40
3.8.11	SL1 Command: Stack Light 1	40
APPENDIX A: COMMAND EXAMPLES		42
A.1:	Command Summary.....	42
A.2:	Summary of Command Syntaxes	43
A.3:	Additional Command Examples.....	45
APPENDIX B: ASCII CONTROL CODES.....		47
APPENDIX C: GLOSSARY OF ACRONYMS AND ABBREVIATIONS		48

SPECIFICATION: Enhanced SCC Multi-Use Protocol Version 2.00

1.0 OVERVIEW

This document presents the specification for the Static Controls Corporation Enhanced SCC Multi-Use Protocol, version 2.00. The Enhanced SCC Multi-Use Protocol is a communications protocol used for controlling Static Controls Corporation (SCC) display devices from a host computer. Display devices may include Andon / bingo boards, alpha-numeric character displays (marquees) and numeric (7-segment) readouts. The host computer and its software are typically supplied by the customer.

The main purpose of this document is to provide the definitions required for the customer to write a software program to run on the host computer such that it can control the SCC display device.

1.1 Revision History

This current document supersedes all earlier versions as noted below:

<u>Name</u>	<u>Rev</u>	<u>Date</u>	<u>Status</u>
MN-0974-200	X0	06-04-97	Original release
MN-0974-200	X1	06-19-97	Modified AC1 command
MN-0974-200	A0	10-03-97	Modified 3.2 commands
MN-0974-200	A1	10-06-97	Modified 3.2 commands
MN-0974-200	A2	01-14-98	Modified A.2 example
MN-0974-200	A3	08-25-98	New AC2 and NUM commands
MN-0974-200	A4	01-14-99	New TIM, expanded NUM commands
MN-0974-200	A5	07-13-00	New AW1, SBT, SVS, 955 commands
MN-0974-200	A6	08-11-00	New TU2, AD1, Mod TU1, AC1, AW1
MN-0974-200	A7	08-11-00	
MN-0974-200	A8	04-23-02	Reinserted SBT command; Various formatting updates
MN-0974-200	A9	08-07-02	Corrected CHR command

SCC uses X0, X1, X2, etc. to indicate the revision level of “pre-production” hardware and documentation.

SCC uses A, B, C, etc. to indicate the revision level of “production” hardware and documentation, where revision “A” indicates the initial production release.

SCC uses 1.00, 2.00, 2.01, etc. to indicate the version number of software and firmware.

The document you are reading is document revision level **A8**, and describes the Enhanced SCC Multi-Use Protocol Version 2.00. It is possible for this document to be revised without necessarily changing the software version number. For example, expanded command descriptions would change the document revision level, but not the software functionality, and hence its version number.

The Enhanced SCC Multi-Use Protocol Version 2.00 is the successor to the original SCC Multi-Use Protocol. The original did not have a version number, but is now known retroactively as Version 1.00.

Version 1.00 is defined by the SCC document, "*STATIC CONTROLS CORP., MULTI-USE PROTOCOL*", SCC document number MN-0973-001 rev H, dated 10-21-96.

The Enhanced SCC Multi-Use Protocol Version 2.00 is designed as a superset (i.e. it is backwards compatible) of the earlier Multi-Use Protocol Version 1.00. However, where conflicts arise between version 2.00 and 1.00, version 2.00 shall take precedence.

1.2 Scope

This document describes the software aspects of the Enhanced SCC Multi-Use Protocol Version 2.00 only. Hardware specifics are discussed only to the extent necessary to describe the software.

Furthermore, the software being described in this document is for the communications protocol that exists at the interface between the SCC display enclosure and the outside world. Details of the internal SCC bus, including its hardware and firmware are not discussed in this document, except as necessary to support the description of the Enhanced SCC Multi-Use Protocol Version 2.00.

The protocol discussed in this document is applicable to an RS-485, half-duplex, multi-display board based environment. While the Ethernet communications option is beyond the scope of this document, please note that the Ethernet based communications are handled as a superset of this protocol. That implies that the entire command sequences described in this document will be contained within the Ethernet packets. The (optional) SCC Ethernet front end will take care of removing the Ethernet protocol layer, and presenting the command packets as described in this document to the display board back-end.

1.3 Related Documents

"*STATIC CONTROLS CORP., MULTI-USE PROTOCOL*" SCC document number MN-0973-001 rev H, dated 10-21-96.

2.0 GENERAL COMMAND STRUCTURE

The following conventions are used throughout this document:

- \$XX A pair of hexadecimal digits (0-9, A-F) preceded by a '\$' indicates a hexadecimal byte value. Values from \$00 to \$FF are allowed.
- x Lower case 'x' indicates a user supplied ASCII HEX byte. ASCII HEX bytes are limited to range of ASCII codes for the characters 0-9 (\$30-\$39) and A-F (\$41 to \$46).
- 'X' A character surrounded by single quotes indicates an ASCII printable character byte (codes \$20 to \$7F). For example, 'G' means \$47 (i.e. the ASCII code for the letter G).
- STX A two or three letter character string indicates the mnemonic for an ASCII control code byte (\$00 to \$1F). For example, STX is Start of TeXt, which is a \$02 in hex.

The protocol uses only ASCII codes (i.e. binary codes are never used).

The Enhanced SCC Multi-Use Protocol is a master-to-slave protocol. The master (host CPU) initiates all communications to the slave (display).

All Enhanced SCC Multi-Use Protocol version 2.00 commands use one of the following two general command structures:

[STX] [GROUP ADDRESS] [UNIT ADDRESS] [DATA] [ENQ] [EOT]
--or--
[STX] [GROUP ADDRESS] [UNIT ADDRESS] [DATA] [ENQ] [ETX] [CHECKSUM]

Command packets are restricted to a maximum length of 255 bytes.

2.1 Command Packet: Initiation Field (STX)

The ASCII start of text character (STX = \$02) is the first byte of all command packets. The STX is used to indicate the start of a packet. It is only used at the beginning of a packet and will never appear anywhere else in a packet. All slaves will assume that a new command packet is starting whenever this character is received.

2.2 Command Packet: GROUP and UNIT Address Fields

Immediately following the STX character is the GROUP address field. This is a two byte field where each byte is an ASCII HEX character. Therefore GROUP addresses from 00 to FF (\$30 \$30 to \$46 \$46) can be represented.

Following the GROUP address field is the UNIT address field. This is also a two byte field where each byte is an ASCII HEX character. Therefore UNIT addresses from 00 to FF (\$30 \$30 to \$46 \$46) can be represented.

The GROUP and UNIT address are together used to select a display element or a grouping of elements. The GROUP and UNIT addresses are assigned at the time the hardware is setup. On first generation SCC hardware the group and unit addresses are set via hardware DIP switches. On second generation SCC hardware the GROUP and UNIT addresses are set via special software commands and are stored in non-volatile memory.

Typically, a single display board (an entire sign containing either a single display element or a group of elements) will have a single GROUP address (e.g. 01) and several UNIT addresses (e.g. 01, 02, 03, etc.). The UNIT addresses are used to select an individual display element (or a collection of similar display elements) within the display board. This collection of display elements is called a DEVICE SET.

It is important that all GROUP and UNIT addresses be unique, if multiple display boards are to exist on a single network cable. This is required in order for the slave display boards to have the ability to respond back to the master (when commanded) and not collide with another slave. Refer to section 3.2.1 of this document for related hardware setup commands.

Address "00" is reserved for use as a broadcast address. This address will automatically match that of all slave devices independent of the address set for the slave devices at setup. Both the GROUP and UNIT address fields support this broadcast feature. Therefore, a GROUP and UNIT address of "0000" will ensure that the command packet which follows will be received by ALL elements of ALL display boards on the network, regardless of their specifically set group and unit addresses. Although all display elements and display boards receive the command packet, only those devices which recognize the command as a valid command will respond.

2.3 Command Packet: DATA Field

The STX, GROUP and UNIT address fields are followed by the DATA field. This field is a collection of (optional) command codes and display data that control the hardware which was selected by the GROUP and UNIT address. The actual specifics of the DATA field varies from one hardware device to the next and is covered in detail in section 3.0 below (HARDWARE SPECIFIC COMMAND STRUCTURE).

In general, the (optional) command portion of the DATA field takes one of two forms. Either the commands are single byte ASCII control codes (\$00 thru \$1F, but excluding STX, EOT, ETX and NAK), or they are multiple byte "escape sequences" where the ASCII control code, "ESC", is followed by one or more additional ASCII characters. These additional bytes for escape sequences are limited to the ASCII printable characters (\$20 to \$7F).

Following the DATA field are either characters to be displayed or, if an attribute is associated with the command, one or more ASCII HEX codes (0-F).

For alpha-numeric displays, the command is optional and ASCII characters to be displayed can be immediately sent. In such a case, they will begin displaying at the current cursor position.

Multiple commands can be sent per command packet. That is, once a slave device has decoded the STX-GROUP-UNIT address header sequence which matches its address, then all additional DATA field sequences will be received by that same slave until an EOT or ETX character is received.

2.4 Command Packet: Request Slave Response Field (ENQ)

This field is optional. If the ASCII control code ENQ follows the DATA field, then the slave will send a response back to the host following the termination of the current command.

The minimum response is either an ACK or a NAK ASCII control byte. However, much longer packets are produced in response to some commands.

The responsibility for controlling the slaves in order to prevent data bus collisions rests with the master (host) computer. Collisions are only a potential problem for multi-drop RS-485 configurations (i.e. two or more slave displays per RS-485 network).

2.5 Command Packet: Termination Field(s) (EOT or ETX-CHECKSUM)

The command packet can be terminated in one of two ways.

The first way is to terminate the command packet with an EOT character which will cause the slave to immediately execute the command.

Alternatively, an ETX character can be sent followed by a two byte ASCII encoded HEX checksum. In this case, the slave will first calculate a checksum of the received command packet and compare the calculated checksum value to the received checksum value. If they are equal, it will then execute the command; otherwise, it will ignore the command. If the DATA field included the ENQ command, then the slave will send a NAK character to indicate an invalid checksum.

The checksum is calculated by summing all of the bytes of the command packet starting with the GROUP address field, through and including the ETX byte. The sum is then truncated to its least significant byte and a two's complement of the remaining byte is performed. The resultant byte is then sent by the master as a pair of ASCII encoded hex digits.

3.0 HARDWARE SPECIFIC COMMAND STRUCTURE

SCC now produces two distinct generations of display products. Among other attributes, the second generation display products are designed to simplify the host computer communications interface.

The Enhanced SCC Multi-Use Protocol Version 2.00 is designed specifically to support the SCC second generation hardware display products, while maintaining backwards compatibility and support for the first generation products.

The first generation SCC display products use multiple display control CPUs per display board. Each CPU has the ability to drive a certain maximum number of display elements. This can sometimes impose awkward addressing requirements on the host master computer.

In contrast, the second generation SCC display hardware uses a single front end CPU which receives the command packets from the outside world (i.e. host computer), parses them, and distributes the appropriate sub-commands to the various display components (circuit boards) over the internal SCC bus.

The initial second generation display controller is the SCC NAB-6.

3.1 Commands for SCC First Generation Hardware

3.1.1 M-950 Marquee Display Control CPU

Please refer to "SCC MULTI-USE PROTOCOL", doc # MN-0973-001 rev H.

3.1.2 M-951 Marquee Display Control CPU

Please refer to "SCC MULTI-USE PROTOCOL", doc # MN-0973-001 rev H.

3.1.3 M-973-A Display Control CPU

Please refer to "SCC MULTI-USE PROTOCOL", doc # MN-0973-001 rev H.

3.1.4 M-973-B Display Control CPU

Please refer to "SCC MULTI-USE PROTOCOL", doc # MN-0973-001 rev H.

3.1.5 M-973-C Display Control CPU

Please refer to "SCC MULTI-USE PROTOCOL", doc # MN-0973-001 rev H.

3.1.6 M-973-D Display Control CPU

Please refer to "SCC MULTI-USE PROTOCOL", doc # MN-0973-001 rev H.

3.1.7 M-973-E Display Control CPU

Please refer to "SCC MULTI-USE PROTOCOL", doc # MN-0973-001 rev H.

3.1.8 M-973-F Display Control CPU

Please refer to "SCC MULTI-USE PROTOCOL", doc # MN-0973-001 rev H.

3.1.9 M-973-G Display Control CPU

Please refer to "SCC MULTI-USE PROTOCOL", doc # MN-0973-001 rev H.

3.1.10 M-973-H Display Control CPU

Please refer to "SCC MULTI-USE PROTOCOL", doc # MN-0973-001 rev H.

3.2 Commands for SCC Second Generation Hardware

Not all commands for SCC second generation hardware are applicable to all enclosures. The commands used for a particular enclosure depend on the configuration of the hardware within that enclosure.

For example: The numeric display control (NUM command of paragraph [3.7.1](#)) has a standard format and two optional formats. One of these command formats will be appropriate for a particular enclosure depending on the row and column configuration of the numeric displays within the enclosure.

For the appropriate command formats of a particular enclosure, refer to the command examples section of the drawing package which accompanies each enclosure.

3.3 NAB-6 Display Control CPU: Hardware Setup

The NAB-6 is the headend controller for the display elements within the display board. It receives the command packets from the (external) host computer, decodes them, and then commands the individual (internal) display elements to perform the appropriate actions.

The NAB-6 also handles the task of grouping multiple individual display elements into larger display sub-systems (called device sets) which the master host computer can then communicate with via a single GROUP and UNIT address. This ability of the NAB-6 to group the display elements is one of the powerful features of the SCC second generation hardware architecture.

At the time of manufacture, each display element within a display board receives a hardcoded internal address known as the SCC bus device number. A table in non-volatile memory within the NAB-6 is then used to equate these device numbers to outside world GROUP and UNIT addresses. This table is configured before the display board is shipped from SCC.

3.3.1 System Configuration Commands

System commands are used to set various parameters and do not directly control the display elements. They are used to set configuration or control variables which are used by the software when initializing or controlling the sign.

These commands set the configuration parameters that determine which sign is supported and the addressing (group, unit, etc.). Note that all system commands are sent to group "00" and unit "00".

3.3.1.1 ESC Sequence Commands

The following commands are used for test and diagnostic purposes and apply to all devices within the display board:

- ESC '0' = reset to power up conditions.
- ESC '1' = do lamp test.
- ESC '2' = do diagnostic test.
- ESC '3' = clear the input buffers, clear all displays, turn all outputs off.

3.3.1.2 Hardware Setup: Set GROUP and UNIT Addresses

The 'SGU' command allows the GROUP and UNIT addresses to be revised from their factory configuration. Each device set (i.e. collection of display board types) within the sign has a group and unit address.

The command syntax is:

ESC 'SSN' sssss ESC 'SGU' gguu gguu gguu gguu

where:

ESC is the ASCII escape control code

'SSN' is the character string 'SSN' (stands for System Serial Number)

sssss is the 5 digit serial number (The serial number is displayed on the marquee at boot time and when the SVN command syntax is used. Refer to section [3.3.1.4](#) of this document)

'SGU' is the character string 'SGU' (stands for Set system Group and Unit addresses)

gguu thru **gguu** fields are the new GROUP and UNIT addresses to be set for each the DEVICE SETS up to the maximum number of possible allowable device sets . GROUP (gg) and UNIT (uu) addresses are specified as 4 ASCII HEX characters. The first "gguu" represents device set 1, the second "gguu" is device set 2, etc. up to the maximum device set supported by the software. A variable number of values are allowed and only those device sets specified are changed. Setting "gguu" to "0000" will disable that device set. These values are saved in a configuration file on the NAB6 CPU.

Refer to section [2.2](#) of this document for additional information regarding the GROUP and UNIT address.

3.3.1.3 Hardware Setup: Set Fader Level

On some display elements the mixture of green and red light that comprise a yellow display is variable and can be controlled by the fader command.

The command syntax is:

ESC 'SSN' sssss ESC 'SFD' gg 'G'

where:

ESC is the ASCII escape control code

'SSN' is the character string 'SSN' (stands for System Serial Number)

sssss is the 5 digit serial number (the serial number is displayed on the marquee at boot time)

'SFD' is the character string 'SFD' (stands for Set system FaDer)

gg is a percent GREEN from 0 to 99. RED is automatically calculated as being the complementary percentage so that RED and GREEN total 100%.

3.3.1.4 Hardware Setup: Redisplay Boot Time Information

At boot time, the alpha-numeric marquee displays a variety of informational messages including, but not limited to the following:

- Current NAB-6 software version
- Serial number of the NAB-6 CPU
- Current GROUP and UNIT settings
- Current IP Address, if using ETHERNET communication

The boot time information can be redisplayed at any time using the SVN command.

The command syntax is:

ESC 'SVN'

where:

ESC is the ASCII escape control code

'SVN' is the character string 'SVN' (stands for System Version Number)

3.3.1.5 Hardware Setup: Set Board Type

Each sign hardware configuration (a.k.a. board configuration) is assigned a unique board type number. This number is set at the factory and typically will only need to be changed in the field if the hardware configuration is changed (e.g. display elements are added or removed from the sign.)

The command syntax is:

ESC 'SSN' sssss ESC 'SBT' nnn where nnn = {0 ... 999}

where:

ESC is the ASCII escape control code;

'SSN' is the character string 'SSN' (stands for System Serial Number);

sssss is the 5 digit serial number (the serial number is displayed on the marquee at boot time);

'SBT' is the character string 'SBT' (stands for Set system Board Type);

nn is a number which represents different sign configurations and is stored as the first line of the NAB6.CFG. The system will reboot after saving this value in the file so that the software will be initialized with the correct sign type.

3.3.1.6 Hardware Setup: Set Auto Center Option

For signs with a marquee display, this command sets the Auto Center Option.

The command syntax is:

ESC SMCnC Where n = 0 or 1

If n = 0, then autocentering is 'OFF' and the displayed text is left-justified on the marquee display (assuming the message is shorter than the maximum length of the display and, therefore, not scrolling).

If n = 1, then autocentering is 'ON' and the display text is centered on the marquee display (assuming the message is shorter than the maximum length of the display and, therefore, not scrolling).

3.4 NAB-6 Display CPU: 974 Andon Cell Control

3.4.1 Andon Cell Control: AC1 variant

For the AC1 command, all of the Andon cells within the display board are accessed via a single GROUP and UNIT address using a [row, column, attribute, color 1, color 2, bin] syntax. Row 1 is the top row. Column 1 is the left column. Row and column numbers increase as you go down, and to the right, respectively.

The command syntax is:

ESC 'AC1' yy 'R' xx 'C' a 'A' p 'P' s 'S' bb 'B'

where:

ESC is the ASCII escape control code;

'AC1' is the character string 'AC1' (stands for Andon Cell type #1);

yy is the Andon cell row number, in ASCII decimal (1 to 99);

'R' is the character 'R' (stands for Row);

xx is the Andon cell column number, in ASCII decimal (1 to 99);
'C' is the character 'C' (stands for Column);

a is the Andon cell display attribute, as follows:

\$30 (0) = off

\$31 (1) = on (default if attribute is not specified)

\$32 (2) = flash

'A' is the character 'A' (stands for Attribute).

p is the Andon cell primary display color, as follows:

\$30 (0) = black

\$31 (1) = red (default if primary color is not specified)

\$32 (2) = amber

\$33 (3) = green

'P' is the character 'P' (stands for Primary color).

s is the Andon cell secondary display color. This is the second color displayed during flashing. The codes are as follows:

\$30 (0) = black (default if secondary color is not specified)

\$31 (1) = red

\$32 (2) = amber

\$33 (3) = green

'S' is the character 'S' (stands for Secondary color).

bb is the Andon cell "bin" number, in ASCII decimal (1 to 16);

The "bin" option is only available on certain software reconfigurable Andon cells (e.g. AD-0974-090 and AD-0974-093) which have on board non-volatile memory (EEPROM) for storing multiple sets of display characters (one set per "bin"). The AW1 command (see 3.2.2.3 below) is used to set the characters stored in each "bin". Default if bin is not specified is 1.

'B' is the character 'B' (stands for Bin);

3.4.2 Andon Cell Control: AC2 variant

The AC2 command is an expanded version of the AC1 command described above. It allows Andon cells or text to be accessed with the same command.

It also allows "sub" rows and columns via a decimal point extension to the main row and column (e.g. M.ss instead of simply M, where M is the number of the main row and ss is the number of the "sub" row).

With the AC2 command, all of the Andon cells within the display board are accessed via a single GROUP and UNIT address using a [row, column, attribute, color 1, color 2] syntax. Row 1 is the top row. Column 1 is the left column. Row and column numbers increase as you go down, and to the right, respectively.

The command syntax is:

ESC 'AC2' yy.y 'R' xx.x 'C' a 'A' p 'P' s 'S' (cell option)

ESC 'AC2' yy.y 'R' xx.x 'C' [wxyz] a 'A' (text option)

where:

ESC is the ASCII escape control code;

'AC2' is the character string 'AC2' (stands for Andon Cell type #2);

yy.y is the Andon cell row number, in ASCII decimal (1 to 99);

The "yy" to the left of the decimal point specifies the "main" row number. This is the row that a display board occupies. The ".y" to the right of the decimal point specifies the "sub" row on the display board. Thus if a display board was in the second row of a sign, and the display board had a column of three Andon cells, the top cell would be in row 2.1, the middle cell in row 2.2 and the bottom cell in row 2.3.

'R' is the character 'R' (stands for Row). It terminates the field;

xx.x is the Andon cell column number, in ASCII decimal (1 to 99);

The "xx" to the left of the decimal point specifies the "main" column number. This is the column that a display board occupies. The ".x" to the right of the decimal point specifies the "sub" column on the display board. Thus if a display board was in the fourth column of a sign, and the display board had a row of three Andon cells, the left most cell would be in column 4.1, the middle cell in column 4.2 and the right most cell in column 4.3.

'C' is the character 'C' (stands for Column). It terminates the field;

a is the Andon cell display attribute, as follows:

\$30 (0) = off
\$31 (1) = on (default if attribute is not specified)
\$32 (2) = flash

'A' is the character 'A' (stands for Attribute). It terminates the field;

p is the Andon cell primary display color, as follows:

\$30 (0) = black
\$31 (1) = red (default if primary color is not specified)
\$32 (2) = amber
\$33 (3) = green

'P' is the character 'P' (stands for Primary color). It terminates the field;

s is the Andon cell secondary display color. This is the second color displayed during flashing. The codes are as follows:

\$30 (0) = black (default if secondary color is not specified)
\$31 (1) = red
\$32 (2) = amber
\$33 (3) = green

'S' is the character 'S' (stands for Secondary color). It terminates the field;

'[' and ']' surround the text characters that are to be displayed.

Text is displayed starting in the left most column of the display board. Messages longer than the available display space are truncated at the right end. Messages shorter than the available space will be padded out with spaces on the right hand end.

The backslash character ('\') must be used before the characters '[', ']' and '\ ' in order to display them as text. For example, to display the string "1\23" on a four character display, the string [1 \ \ 2 3] would be used.

The Cell and Text options may not be mixed in the same command string.

3.4.3 Andon Cell Control: AW1 command

The AW1 command is used to configure the alpha-numeric characters displayed by the software-reconfigurable Andon cells (e.g. AD-0974-090 or AD-0974-093). The characters are stored in on-board, non-volatile memory (EEPROM).

The AW1 command is expected to be used "infrequently". Infrequently is defined by the lifetime of the EEPROM, which is 100,000 cycles (i.e. 100,000 uses of the AW1 command). Once the alpha-numeric characters have been set (typically at the time that the sign is installed), the Andon cell is controlled, thereafter, via the AC1 command.

For the AW1 command, all of the Andon cells within the display board are accessed via a single GROUP and UNIT address using a [row, column, bin, text] syntax. Row 1 is the top row. Column 1 is the left column. Row and column numbers increase as you go down, and to the right, respectively.

The command syntax is:

```
ESC 'AW1' yy 'R' xx 'C' bb 'B' [ xyz ]
```

where:

ESC is the ASCII escape control code;

'AW1' is the character string 'AW1' (stands for Andon cell Write type #1);

yy is the Andon cell row number, in ASCII decimal (1 to 99);

'R' is the character 'R' (stands for Row);

xx is the Andon cell column number, in ASCII decimal (1 to 99);

'C' is the character 'C' (stands for Column);

bb is the Andon cell "bin" number, in ASCII decimal (1 to 16);

A different text string can be stored in each of up to 16 bins. Bins are loaded one at a time with the AW1 command. The currently displayed bin is selected via the AC1 command. Default if bin is not specified is 1. Cell text in the bins defaults to a solid block if nothing has been stored in them.

'B' is the character 'B' (stands for Bin);

'[' and ']' surround the text characters that are to be stored in the currently selected bin.

Text is stored starting in the left most column of the display board. Text strings longer than the available display space are truncated at the right end. Text strings shorter than the available space will be padded out with spaces on the right hand end.

The backslash character ('\') must be used before the characters '[' , ']' and '\ ' in order to store them as text. For example, to store the string "\23" on a four character display, the string [1 \ \ 2 3] would be used.

3.5 NAB-6 Display CPU: 952 & 954 Alpha-Numeric Marquee Control

The 952/954 Marquees are alpha-numeric displays that are a multiple of five 4 inch characters in length (typically 20 or 25). This is an SCC second generation hardware device.

For the case where there is more than one 952/954 Marquee line, each line will have a unique GROUP and UNIT address, and the lines will be controlled independently.

The initial 952/954 command set is based on a subset of the *first generation* 950 command set. The 952/954 marquee commands and data are:

\$08	= destructive backspace cursor one position.
\$09	= non-destructive cursor forward one position.
\$0C	= put cursor to the home position (character position #1) and clear the display.
\$0D	= carriage return, cursor to beginning of the line.
\$10	= flash function on. All characters after this command will flash until a flash off command is issued. A single command will flash characters at the slow rate (once per 2 seconds). A double issued command will flash characters at the fast rate (once per second).
\$12	= flash function off. All characters after this command will not flash.
\$14 (xx)	= set cursor position. Followed by 2 ASCII HEX numbers of character position.
\$20 thru \$7F	= displayable alpha-numeric ASCII encoded characters.

3.6 NAB-6 Display CPU: Music Maker Control

3.6.1 Music Maker Control: TU1, TU2 Commands

TU1 Command Overview

This command is used to play a particular tune on a particular music system device at normal tempo. The TU1 command has the following configurable parameters:

- Tune number - Index number of the tune to be played (0-255, 0 is off).

- Play duration - Length of time in seconds that the specified tune will play.
- Repeat delay - Length of time in seconds between end of play duration and repeat.
- Total duration - Length of time in seconds that play duration+repeat delay loops.
- Priority - Allows tunes to be given value of importance. Priority value is 1 to 5, 1 being highest.

TU2 Command Overview

This command is used to play a particular tune on a particular music system device at fast tempo. The TU2 command has the following configurable parameters:

- Tune number - Index number of the tune to be played (0-255, 0 is off).
- Play duration - Length of time in seconds that the specified tune will play.
- Repeat delay - Length of time in seconds between end of play duration and repeat.
- Total duration - Length of time in seconds that play duration+repeat delay loops.
- Priority - Allows tunes to be given value of importance. Priority value is 1 to 5, 1 being highest.

TU1/TU2 Command Syntax:

```
ESC 'TU1' ttt 'T' sss 'S' rr 'R' ddd 'D' p 'P'
ESC 'TU2' ttt 'T' sss 'S' rr 'R' ddd 'D' p 'P'
```

Where:

- ESC (required) The ASCII escape control code (ESC=\$1B)
- TU1 (required) The character string 'TU1' which stands for TUne type 1, normal tempo.
- TU2 (required) The character string 'TU2' which stands for TUne type 2, fast tempo.
- ttt (required) The tune number in ASCII decimal (0 to 255, 0 is off). A value of 0 stops any tune playing, regardless of tune length and priority. Padding zeros are not necessary (i.e. tune #1 = #001).
- T (required) The character string 'T' which stands for Tune number.
- sss (optional) The play duration in seconds, in ASCII decimal (1 to 999). If this field is not used then the tune will play until stopped or until another tune is selected.

- S (optional) The character string 'S' which stands for Seconds. The specified tune plays for sss seconds.
- rr (optional) The repeat delay time in seconds, in ASCII decimal (0 to 99).
- R (optional) The character string 'R' which stands for Repeat. The specified tune is repeated after a delay = rr for the specified play duration. This cycle continues until the total duration expires.
- ddd (optional) The total duration in seconds, in ASCII decimal (1 to 999). If included, this field limits continuous play established using play duration and repeat delay.
- D (optional) The character string 'D' which stands for Duration.
- p (optional) The tune priority, in ASCII decimal (1 to 5), 1 being highest priority, 5 being lowest. The music system will play the tune with highest priority ignoring tune commands received with lower priority until a tune with higher priority has completed. The tune with lower priority will not be held in a buffer if higher tune priority is playing.
- P (optional) The character string 'P' which stands for Priority. If priority is not included in a particular command, then all subsequent commands received take precedence over that command.

Examples:

0x02 0006 0x1b TU1023T10S5R120D 0x04

Tune # 023 with normal tempo, play for 10 seconds, and repeat with a 5 second delay between playing and a total duration of 2 minutes (120 seconds).

0x02 0006 0x1b TU2013T 0x04

Tune # 13 with fast tempo, will play continuously.

Examples using the Priority Parameter:

0x02 0006 0x1b TU1013T4P 0x04

Tune #13 will play continuously with a number 4 priority.

0x02 0006 0x1b TU1014T5P 0x04

Tune #14 with a 5 priority will not be played previous command has higher priority.

0x02 0006 0x1b TU2022T1P 0x04

Tune #22 with a 1 priority will be played at fast speed continuously, has higher priority than previous command.

0x02 0006 0x1b TU20T 0x04

All tunes will be stopped because tune #0 (no tune) has been selected.

3.6.2 Music Maker Control: VL1 Command

This command is used to remotely set the volume of a particular channel of the music system. All four channels play the same tune but the volume level of each channel is controlled independently. The VL1 command has the following configurable parameters:

- Volume level - Desired volume level (0 to 100, 0 is off).
- Channel - Target channel (1 to 4), labeled "OUT 1" to "OUT 4" on the music system.

VL1 Command Syntax:

ESC 'VL1' vvv 'V' c 'C'

Where:

ESC (required) The ASCII escape control code (ESC=\$1B)

VL1 (required) The character string 'VL1' which stands for VoLume type 1.

vvv (required) The volume level in ASCII decimal (0 to 100, 0 is off).

V (required) The character string 'V' which stands for Volume. This value is maintained until changed by another VL1 command or until modified by the local volume input.

c (required) The channel in ASCII decimal (1 to 4).

C (required) The character 'C' which stands for Channel.

3.7 NAB-6 Display CPU: 801 Numeric/Time Display Control

3.7.1 801 Numeric Display Control: NUM Command

The SCC AD-0801-xxx series of numeric displays are controlled using the NUM command. In most applications, each numeric display will be assigned to a separate device set, and thus will be accessed via a unique GROUP and UNIT address (variant 1 below). However, in some applications, multiple numeric displays will be assigned to a single device set (i.e. one GROUP and UNIT to affect the entire group of numeric displays) and individual displays will be addressed by using variants 2 or 3 of the NUM command which include a row and column parameter.

The command syntax is:

```
ESC 'NUM'          nnnn N   a   'A' (variant 1)
ESC 'NUM' yy 'R'   xx 'C' nnnn N   a   'A' (variant 2)
ESC 'NUM' yy.y 'R' xx.x 'C' nnnn N   a   'A' (variant 3)
```

Row 1 is the top row. Column 1 is the left column. Row and column numbers increase as you go down and to the right, respectively.

where:

ESC is the ASCII escape control code;

'NUM' is the character string 'NUM' (stands for NUMeric);

nnnn is the numeric data field (i.e. the digits that are to be displayed).

The number of digits entered will vary according to the number of digits available on the display. The displayed number is right justified and zero filled on the left. Extra digits on the left side of the field are discarded.

Valid "digits" are 0-9, '-' (hyphen), ' ' (blank) and '.' (decimal point).

The decimal point may follow any of the digits.

'N' is the character 'N' (stands for Number). It terminates the field;

a is the Numeric display attribute, as follows:

\$30 (0) = off

\$31 (1) = on (default if attribute is not specified)

\$32 (2) = flash slow

\$33 (3) = flash fast

'A' is the character 'A' (stands for Attribute). It terminates the field;

NOTE: All of the digits of the numeric display share the same attribute.

For variants 2 and 3, the row and column parameters are used to select one of N numeric display boards in an x by y array of boards. The parameters have the following meanings:

yy is the numeric display row number, in ASCII decimal (1 to 99);

'R' is the character 'R' (stands for Row). It terminates the field;

Row 1 is the top row. Row numbers increase as you go down.

xx is the numeric display column number, in ASCII decimal (1 to 99);

'C' is the character 'C' (stands for Column). It terminates the field;

Column 1 is the left most column. Column numbers increase as you go right.

yy.y is the numeric display row number, in ASCII decimal (1 to 99);

The “yy” to the left of the decimal point specifies the “main” row number. This is the row that a cluster of display boards occupy. The “.y” to the right of the decimal point specifies the “sub” row of a particular numeric display board within the cluster. The row and sub row assignments of a particular enclosure are illustrated in the drawing package that accompanies that enclosure.

‘R’ is the character ‘R’ (stands for Row). It terminates the field;

xx.x is the numeric display column number, in ASCII decimal (1 to 99);

The “xx” to the left of the decimal point specifies the “main” column number.

This is the column that a cluster of numeric display boards occupies. The “.x” to the right of the decimal point specifies the “sub” column of a particular numeric display board within the cluster. The column and sub column assignments of a particular enclosure are illustrated in the drawing package that accompanies that enclosure.

‘C’ is the character ‘C’ (stands for Column). It terminates the field.

3.7.2 801 Numeric Display Control: TIM Command

Some numeric displays of the SCC AD-0801-xxx series are controlled using the TIM command (stands for TIME). This command allows the user to set the current time and date which updates the numeric display. The numeric display will then keep time. In order to ensure that the time displayed remains accurate to the level of precision desired, the user may re-send additional TIM commands, at their discretion, to update the display.

The command syntax is:

ESC ‘TIM’ hh:mm:ss ‘T’ yyyy-mm-dd ‘D’ a ‘A’

where:

ESC is the ASCII escape control code;

‘TIM’ is the character string ‘TIM’ (stands for TIME);

hh is a numeric data field representing the hour to be displayed. This value must be entered in the 24 hour format (00 to 23), although it may be displayed in either the 12-hour or 24-hour format. A colon is used to separate the hh field from the mm field.

mm of the time field is a numeric data field representing the minutes to be displayed (00 to 59). A colon is used to separate the mm field from the ss field.

ss is a numeric data field representing the seconds to be displayed (00 to 59). The colon and ss (‘:ss’) are optional (default=00).

‘T’ is the character ‘T’ (stands for Time). It terminates the field;

yyyy is a numeric data field representing the year. A hyphen is used to separate the yyyy field from the mm field.

mm of the date field is a numeric data field representing the month (01 to 12). A hyphen is used to separate the mm field from the dd field.

dd is a numeric data field representing the day of the month (01 to 31).

'D' is the character 'D' (stands for Date). It terminates the field.

a is the display format (1 = 12-hour, 2 = 24-hour).

'A' is the character 'A' (stands for Attribute). It terminates the field.

Every time the user sends a command packet which includes a valid attribute, that attribute setting is saved to non-volatile memory. This means that the attribute will remain set until the user changes it.

Each of the three fields (Time, Date, Attribute) in the TIM command are optional. This means that you may change any of them without affecting the others, should you choose to do so. (e.g. You may change the attribute from 12-hour format to 24-hour format without affecting the time or date settings. Or, you may change the actual time without affecting the date or time format attribute).

3.8 NAB-6 Display CPU: 955 Tri-Color Alpha-Numeric Marquee Control

The 955 Marquees are alpha-numeric displays. The length of each line is a multiple of five, four-inch character boards (typically 5 or 6 boards, which provide from 25 to 30 characters of display area without scrolling). The 955 uses bi-color LEDs (RED and GREEN) which, by varying the intensity or "mix" of the RED and GREEN, are capable of displaying up to 9 colors. This is an SCC second generation hardware device.

Via software control, the 955 can be configured to respond to either marquee text commands (TX1) or Andon Cell commands (AD1, AW1, AC1, CHR). Many of the following additional commands apply: RST, DEL, TS1, NRM, and CLR.

3.8.1 TX1 Command: Text Command

The TX1 command is issued to display text messages on a specified row in the light board. If the specified row is currently in an ANDON configuration upon receipt of the TX1 command (meaning it is currently displaying cell values), the light board will blank the current cell values and, instead, display the text message with the specified display attributes. The TX1 command will ***not*** remove the cell values that are stored in non-volatile memory (values that have been configured using the AW1 command). The text sent by the TX1 command will be stored in volatile memory and will continue to display until one of the following occurs:

- The row receives a subsequent TX1 command
- The row receives a RST command (reset command)
- The row receives a CLR command (clear command)
- The Row receives a TS1 command (test command)

After receiving an RST, CLR or TS1 command, the row will return to an ANDON configuration.

TX1 command syntax:

ESC 'TX1' yy 'R' a 'A' p 'P' s 'S' f 'F' [xyz]

Where:

ESC = (required) the ASCII control code escape

TX1 = (required) ASCII string 'TX1' (stands for Text Mode type 1)

yy = (required) the Andon cell row number, in ASCII decimal (1 to 99)

'R' = (required) the character 'R' (stands for row)

Note: The following parameters are optional and will be accepted as long as they do not effect the error proofing operation of the board.

a = attribute

Where a is:

- 0 = off
- 1 = hold (default, if a is not supplied)
- 2 = normal flash (1 second cycle)
- 3 = fast flash (1/2 second cycle)

A = Attribute Terminator

p = The primary display color for the text in the cell

Where p is:

- 0 = black (off)
- 1 = red (default)
- 2 = amber
- 3 = green
- 4 = light red
- 5 = yellow
- 6 = light green
- 7 = brown
- 8 = orange

P = Primary Color Terminator

s = The secondary display color for the text in the cell

Where s is:

0 = black (default/off)

1 = red

2 = amber

3 = green

4 = light red

5 = yellow

6 = light green

7 = brown

8 = orange

S = Secondary Color Terminator

f = The font code:

Where f is:

0 = fixed space, 7 pixel high sans serif (default)

1 = proportional spaced, 7 pixel high, serif

2 = proportional spaced, 7 pixel high, sans serif

3 = proportional spaced, 5 pixel high, sans serif

4 = proportional spaced, 7 pixel high, sans serif, double wide

5 = proportional spaced, 7 pixel high, bold, sans serif

6 = user defined font #1 (As defined with the CHR command)

F = Font Terminator

Note: If any of the "optional" parameters above are chosen, then the corresponding parameter terminator (A, P, S or F) MUST also be included.

[xyz] = Text to be displayed (THIS IS A REQUIRED FIELD)

If the backslash character or either of the bracket characters are to be included in the text, they are to be *preceded* by the back-slash character.

Examples: \ \ []

Example of a TX1 command:

0x02 0202 0x1b TX13R1A3P2J[Hello There] 0x04

This will establish that row 3 will change to a text-only row and display "Hello There" (quotes will not be displayed) in a green, holding state, right justified.

3.8.2 RST Command: Restore

The RST command will be used to restore a specified row or an entire light board to previous state. This command will end a previous TX1 command for all rows, or the selected row, and return it to its previous state.

RST command syntax:

ESC 'RST' yy 'R'

Where:

ESC = (required) the ASCII control code escape

RST = (required) ASCII string 'RST' (stands for Andon sign Reset)

yy = (optional) is the Andon cell row number, in ASCII decimal (1 to 99)

R = (optional) the character 'R' (stands for row)

Example:

0x02 0207 0x1b RST3R 0x04

This will establish that row 3 will be restored to its previous display state before command was issued.

3.8.3 AD1 Command: Column Definition & Sizing

The AD1 command defines the column characteristics of a specified row of the light board. The AD1 command will define the width (in pixel columns) of each display cell within the row. A valid cell width may range from 1 to the maximum number of pixel columns in the row. Once the sizes of the light board cells have been configured, no further AD1 commands need to be issued unless it is desirable to change the size configuration. This command will take effect immediately after it is issued.

AD1 command syntax:

ESC 'AD1' yy 'R' [cc,cc,cc,cc,cc...cc]

Where:

ESC = (required) the ASCII control code escape

AD1 = (required) ASCII string 'AD1' (stands for Andon Define type 1)

yy = (required) the Andon cell row number, in ASCII decimal (1 to 99)

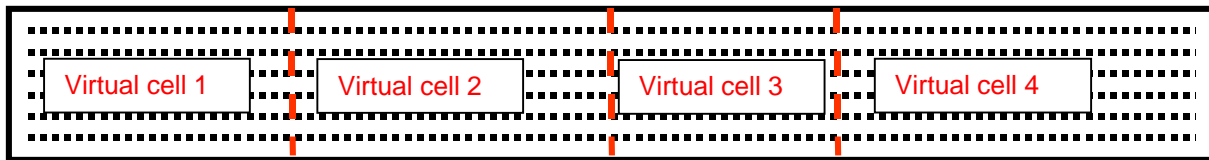
R = (required) the character 'R' (stands for row)

[cc, cc, cc, ...cc] = Cell configurations where: cc = number of pixels (columns) from 1 column of pixels to the max number of pixel columns on light board being configured. One cell required, up to x configurations (x = Max Cells, i.e. 40)

Example:

0x02 0005 0x1b AD13R[25, 30, 21, 34] 0x04

This will establish that: cell 1 in row 3 will be 25 pixel columns
cell 2 in row 3 will be 30 pixel columns
cell 3 in row 3 will be 21 pixel columns
cell 4 in row 3 will be 34 pixel columns



AD1 Command Error Checking

If 'AD1' commands received by the board exceed the column space available for the row, the sign will display "illegal configuration" and remain in that condition until a legal AD1 command is received. This will only appear on the row to which the illegal command was sent. For example, if the maximum pixels for a row is 160 and a total of 200 pixel columns have been issued in the AD1 command, then the error message will be displayed.

The AD1 commands will not error on commands that under-utilize the pixels available on a board. For example, if the max pixels for a row is 160 and a total of 150 pixel columns have been issued in the AD1 command, then the error will not be displayed and the columns will be set even though the row is under-utilized.

When an AD1 command is issued to a board that has been previously defined, all previous column data for the row is removed. If board pixels are under utilized, data from the previous definition will not display. Example: Board had three columns defined, AD1 command redefines two columns, and the data from previous third column is deleted.

3.8.4 AW1 Command: Write Values and Attributes

The AW1 Command is used to store display values and default display attributes for the cells (which were already defined using the AD1 Command -- see previous AD1

definition for more details). The display values and the default display attributes assigned to a cell using the AW1 command will be stored in non-volatile memory and will remain in memory until a subsequent AW1 command is issued to reconfigure the cells.

Commands issued with AW1 take effect immediately and remove previous cell data. As many as 16 display values may be stored per cell. The specific display values for a particular cell will be identified by a bin number (bins 1 to 16).

Note: Cells must be configured using the AD1 command before the AW1 command can be issued.

AW1 command syntax:

```
ESC 'AW1' yy 'R' xx 'C' p 'P' s 'S' f 'F' j 'J' a 'A' [xyz] bb 'B'
```

Where:

- ESC = (required) the ASCII control code escape
- AW1 = (required) ASCII string 'AW1' (stands for Andon Cell Write type 1)
- yy = (required) the Andon cell row number, in ASCII decimal (1 to 99)
- R = (required) the character 'R' (stands for row)
- xx = (required) the Andon cell number, in ASCII decimal (1 to 40)
- C = (required) the Character 'C' (stands for column)
- p = (optional) the primary display color default for the text in the cell

where:

- 0 = black (off)
- 1 = red (default)
- 2 = amber
- 3 = green
- 4 = light red
- 5 = yellow
- 6 = light green
- 7 = brown
- 8 = orange

- P = (optional) the character 'P' (stands for primary color)
- s = (optional) the secondary display color default for the text in the cell

where:

- 0 = black (default/off)
- 1 = red
- 2 = amber
- 3 = green
- 4 = light red
- 5 = yellow
- 6 = light green
- 7 = brown
- 8 = orange

S = (optional) the character 'S' (stands for secondary color)

f = (optional) is the font code:

where:

- 0 = fixed space, 7 pixel high sans serif (default)
- 1 = proportional spaced, 7 pixel high, serif
- 2 = proportional spaced, 7 pixel high, sans serif
- 3 = proportional spaced, 5 pixel high, sans serif
- 4 = proportional spaced, 7 pixel high, sans serif, double wide (see below)
- 5 = proportional spaced, 7 pixel high, bold, sans serif
- 6 = user defined font #1 (As defined with the CHR command)

F = (optional) the character 'F' (stands for Font)

j = (optional) justification

Where:

- 1 = left
- 2 = right
- 3 = center (default)

J = (optional) the character 'J' (stands for justification)

a = (optional) the Andon cell default attribute

where:

- 0 = off
- 1 = on solid (default if 'A' is unspecified)
- 2 = normal flash (1 second cycle)
- 3 = fast flash (1/2 second cycle)

A = (optional) the character 'A' (stands for attribute)

[xyz] = (required) text to be displayed

If the backslash character or either of the bracket characters is to be included in the text, they are to be preceded by the backslash character.

Examples: \ \ []

bb = (optional) the Andon cell number, in ASCII decimal (1 to 16... default is 1 if not specified)

B = (optional) the character 'B' (stands for bin)

Example:

0x02 1202 0x1b AW13R5C [Hello \[There\]] 1B 0x04

This will establish that row 3;cell 5; will be set to the value of "Hello [There]" in bin number 1 with system defined default attributes.

AW1 Command Error Checking

If the text to display exceeds the size of the cell defined with the AD1 command, the data will truncate and display an error notification. The error notification will be a question mark "?" in the right-most pixels of the cell. The text data and the "?" will remain in the affected cell and will not affect the cells around it. The error will only be displayed in the bin that is affected and will affect the other bins that are in that cell.

CAUTION: This error may not be evident until that cell is activated.

AW1 Defaults

All attributes for cells and bins are determined by the AW1 command. If optional attributes are not supplied via the AW1 command, the following default attributes will be added to the cell and bin:

Primary color	Red
Secondary color	Black
Font	7-pixel high Sans Serif
Justification	Center
Display mode	On solid (non-flashing)
Bin number	#1 (First bin location)

NOTE: Double Wide Fonts, such as Font # 4 (proportional spacing, 7 pixel high, Sans Serif, doublewide) will require twice the pixels as a standard size font. For example, the standard size font requires 6 pixels, where the double size of the standard font requires 12 pixels.

3.8.5 AC1 Commands: Activate Attributes

The AC1 command is issued to a light board cell to set or update the display attribute for that cell. Valid display attributes are as follows:

Set display color

Set animation (solid, flash slow, flash fast)

Set bin in Cycle mode (See Cycle Bin Command Options)

Set font (applies a font to the display value)

Set justification (left, right, center)

Set bin number (indicates which bin value to display)

The AC1 command is also used to temporarily display a value in a cell with the capability of setting all of the above attributes. However, the value being displayed will be held in volatile memory and will clear on any subsequent command addressed to the cell in which it is being displayed.

NOTE: AC1 commands that do not include optional attributes will result in a display in accordance with the attributes previously assigned to the cell and bin using the AW1 command.

AC1 Command Syntax:

ESC 'AC1' yy 'R' xx 'C' a 'A' p 'P' f 'F' j 'J' bb 'B' (variant 1)

ESC 'AC1' yy 'R' xx 'C' a 'A' p 'P' [xyz] f 'F' j 'J' (variant 2)

Variant 2: (Note: This command displays text in a cell temporarily rather than writing to a bin. The text, font, and justification parameters will temporarily override those set for the bin.)

Where:

ESC = (required) the ASCII escape control code;

AC1 = (required) the character string 'AC1' which stands for Andon Cell type 1

yy = (required) the Andon cell row number, in ASCII decimal (1 to 99)

R = (required) the character 'R' (stands for row)

Xx = (required) the Andon cell number, in ASCII decimal (1 to 40)

C = (required) the Character 'C' (stands for column)

a = (optional) the Andon cell attribute

where a is:

0 = off

1 = on (default if 'A' is unspecified)

2 = normal flash (1 second cycle)
3 = fast flash (1/2 second cycle)

A = (optional) the character 'A' (stands for attribute)

p = (optional) the primary display color for the text in the cell

where p is:

0 = black (off)
1 = red (default)
2 = amber
3 = green
4 = light red
5 = yellow
6 = light green
7 = brown
8 = orange

P = (optional) the character 'P' (stands for primary color)

s = (optional) the secondary display color for the text in the cell

where s is:

0 = black (default/off)
1 = red
2 = amber
3 = green
4 = light red
5 = yellow
6 = light green
7 = brown
8 = orange

S = (optional) the character 'S' (stands for secondary color)

[xyz] = (optional) text to be displayed (text option not active with bin cycle)

If the backslash character or either of the bracket characters are to be included in the text, they are to be preceded by the backslash character.

Examples: \\ \[\]

f = (optional) is the font code: (font code affects all bins when used with bin cycle)

where f is:

0 = fixed space, 7 pixel high sans serif (default)
1 = proportional spaced, 7 pixel high, serif

- 2 = proportional spaced, 7 pixel high, sans serif
- 3 = proportional spaced, 5 pixel high, sans serif
- 4 = proportional spaced, 7 pixel high, sans serif, double wide
(requires 2x pixel spacing)
- 5 = proportional spaced, 7 pixel high, bold, sans serif
- 6 = user defined font #1 (As defined with the CHR command)

F = (optional) the character 'F' (stands for Font)

j = (optional) justification

where j is:

- 1 = left
- 2 = right
- 3 = center (default)

J = (optional) the character 'J' (stands for justification)

bb = (optional) the Andon cell number, in ASCII decimal (1 to 16 .. default is 1 if not specified, no cycle, stop cycle of bins and displays bin)

'B' = (optional) the character 'B' (stands for bin)

Example1 (variant 1):

0x02 0601 0x1b AC13R5C3A2P3B 0x04

This will establish that row 3, cell 5, will display Flashing Fast in Amber in the third bin.

Example 2 (variant 2):

0x02 1207 0x1b AC13R5C3A2P[hello]2F 0x04

This will establish that row 3, cell 5, bin 1 will display "hello", in a proportional spaced, 7 pixel high, sans serif font, Flashing Fast in Amber.

AC1 Command Error Checking

If the text to be displayed exceeds the size of the cell defined with the AD1 command, the data will truncate and display an error notification. The error notification will be a question mark "?" in the right-most pixels of the cell. The text data and the "?" will remain in the affected cell and will not affect the cells around it.

If utilizing the optional text command causes text to exceed the cell size, then the error will only appear when that command is issued and will clear when the cell has been restored to a correct configuration. For example, utilizing the doublewide font (4 doublewide) requires twice as many pixels as the standard size fonts. The error (?) would only occur when the doublewide font is activated in a cell that is undersized for a doublewide font.

AC1 Cycle Bins Command Options

The AC1 Cycle Bins Command allows a command issued to a light board cell to set, or update, the display attribute for cells that are cycling between bin values.

Valid display attributes are as follows:

- Set bin number (indicates which bin value to display)
- Set bin in Cycle mode (display more than one bin number by specifying bin numbers to toggle between. bins display in the order specified in command)
Note: The AC1 command issued with the bin cycle option will utilize the attributes previously set for those bins with the AW1 command unless color and/or font size are specified.
- Set color value for bin (Red, Yellow, Green....)
- Set Font for bin (Standard, doublewide...)

The AC1 bin commands will be held in volatile memory and will clear on any subsequent command addressed to the cell in which it is being displayed.

AC1 commands that do not define command attributes will utilize the attributes previously assigned to the cell and bin using the AW1 command. The display mode on a cycling bin will be solid. (To assist in readability, flash is not an available option while bin-cycling.)

AC1 Bin Cycle Command Syntax:

- ESC 'AC1' yy 'R' xx 'C' p 'P' bb 'B' (variant 1)
- ESC 'AC1' yy 'R' xx 'C' p 'P' f 'F' [bb...bb] 'B' (variant 2)
- ESC 'AC1' yy 'R' xx 'C' [+bb] 'B' (variant 3)
- ESC 'AC1' yy 'R' xx 'C' [-bb] 'B' (variant 4)

Where:

ESC = (required) the ASCII escape control code;

AC1 = (required) the character string 'AC1' which stands for Andon Cell type 1

yy = (required) the Andon cell row number, in ASCII decimal (1 to 99)

R = (required) the character 'R' (stands for row)

xx = (required) the Andon cell number, in ASCII decimal (1 to 40)

C = (required) the Character 'C' (stands for column)

p = (optional) the primary display color for the text in the cell
where:

0 = black (off)

1 = red (default)

- 2 = amber
- 3 = green
- 4 = light red
- 5 = yellow
- 6 = light green
- 7 = brown
- 8 = orange

P = (optional) the character 'P' (stands for primary color)

s = (optional) the secondary display color for the text in the cell

where:

- 0 = black (default/off)
- 1 = red
- 2 = amber
- 3 = green
- 4 = light red
- 5 = yellow
- 6 = light green
- 7 = brown
- 8 = orange

S = (optional) the character 'S' (stands for secondary color)

f = (optional) is the font code: (font code affects all bins when used with bin cycle)

Where:

- 0 = fixed space, 7 pixel high sans serif (default)
- 1 = proportional spaced, 7 pixel high, serif
- 2 = proportional spaced, 7 pixel high, sans serif
- 3 = proportional spaced, 5 pixel high, sans serif
- 4 = proportional spaced, 7 pixel high, sans serif, double wide (requires 2x pixel spacing)
- 5 = proportional spaced, 7 pixel high, bold, sans serif
- 6 = user defined font #1 (As defined with the CHR command)

F = (optional) the character 'F' (stands for Font)

bb = (optional) the Andon cell number, in ASCII decimal (1 to 16 .. default is 1 if not specified, no cycle) Single Bin command resets cycle mode. See standard AC1 command.

OR Cycle bins

[bb,...,bb] = Bin Cycle from 1 to 16 bin numbers. If more than one bin number is specified then the display will cycle between the specified bins in the order specified (bin cycle attributes for font must be set in the AW1 command)

OR

[+bb] = Bin Cycles from 1 to 16 bin numbers. Bin numbers to cycle are specified and added to list which will cycle between the specified bins in the order specified

[-bb] = Bin Cycle removes the bin number specified, removed from the list which will cycle between the specified bins in the order

'B' = (optional) the character 'B' (stands for bin)

A single command is issued as each bin is added or removed from the cycle list.

Issuing the single bin display command would clear all cycling received in previous commands.

Sending a CLR or RST would stop all cycling and clear bins from the cycling list. Bins will cycle in Last In, First Out (LIFO) order. LIFO order is necessary since an update to the board may be received before the entire cycle has completed. This will ensure that the newest activation is not delayed. Other attributes, defined earlier, will not change.

Bin 1 could be added to the cycle by issuing the following command:

AC1 2 'R' 1 'C' [+1] 'B'

Additional Examples:

AC1 2 'R' 1 'C' [+2] 'B'

AC1 2 'R' 1 'C' 2'P'4'F'[+3] 'B'

AC1 2 'R' 1 'C' [+4] 'B'

AC1 2 'R' 1 'C' [-3] 'B'

AC1 2 'R' 1 'C' [-2] 'B'

AC1 2 'R' 1 'C' [3] 'B'

AC1 2 'R' 1 'C' 1 'B'

Bin 2 is activated with AW1 defaults

Bin 3 is double wide and is added to cycle with Bin 2

Bin 4 is red and is added to cycle with Bin 2 and 3

Bin 3 is removed from the cycle, Bin 4 and 2 continue to cycle

Bin 2 is removed from cycle, Bin 4 continues to display

Bin 3 is added to cycle with Bin 4

Bin 1 is displayed, all cycling is cleared

NOTE: A cell will not flash when in Cycle Bin Mode to ensure readability.

3.8.6 CHR Command: Character Create

The CHR command is used to set user defined fonts and characters. The default user defined set will be initialized to the 7 high sans serif font. This font can be used in the same manner as any other font, and it is also programmable by the user. Using the CHR command, certain or all of the characters in the set can be modified or completely changed. The set will be stored in non-volatile memory indefinitely or until it is reprogrammed. A change to the user defined set applies to all rows of the Andon display.

CHR command syntax:

ESC 'CHR' f 'F' c 'C' b1,b2,b3 ... bn EOT

Where:

ESC = the ASCII control code escape

CHR = ASCII string 'CHR' (stands for character set update)

f = (optional) is the font code:

where:

0 = (reserved/not programmable) fixed space, 7 pixel high sans serif
(default)

1 = (reserved/not programmable) proportional spaced, 7 pixel high, serif

2 = (reserved/not programmable) proportional spaced, 7 pixel high, sans serif

3 = (reserved/not programmable) proportional spaced, 5 pixel high, sans serif

4 = (reserved/not programmable) proportional spaced, 7 pixel high, sans serif, double wide

5 = (reserved/not programmable) proportional spaced, 7 pixel high, bold, sans serif

6 = user defined font (default if not defined)

F = (optional) the character 'F' (stands for Font)

c = 1 byte representing the character number to be programmed
0x20 – 0x7f. Use the ASCII character (e.g. use 'N' for 0x4e)

b1 .. bn = the character data

C = (optional) the character 'C' (stands for Character)

Note: A single character can be defined as any width from 1 pixel wide to the maximum number of pixels in a light board row. The characters will be constructed from byte values. Each byte value will equate to a pixel column. The order of values within the message starts with the left-most pixel column of the character and moves to the right. The lowest order pixel within a pixel column will be the top pixel within the column.

Example: (modifying the N character)

Command:

0x02 0504 0x1bCHR6FNC127,2,4,8,16,32,127 0x04

Send values: 127 - 2 - 4 - 8 - 16 - 32 - 127

Bit Place Value							
1	1	0	0	0	0	0	1
2	1	1	0	0	0	0	1
4	1	0	1	0	0	0	1
8	1	0	0	1	0	0	1
16	1	0	0	0	1	0	1
32	1	0	0	0	0	1	1
64	1	0	0	0	0	0	1
128	0	0	0	0	0	0	0
Dec Value	127	2	4	8	16	32	127

Note: The highest order bit is unused

3.8.7 DEL Command : Delete Command

The 'DEL' delete command is used to clear Lightboard memory.

It will remove all AD1 command formatting and AW1 values from the board. All cell sizes and values held in bins will be removed. After a delete command is issued the board MUST be reformatted with AD1 and AW1 commands.

DEL command syntax:

ESC 'DEL'

Where:

ESC = (required) the ASCII control code escape

DEL = (required) ASCII string 'DEL' (stands for Delete stored values)

EOT = (required) End of transmission.

Example: (Delete Board Setup)

0x02 0501 0x1b DEL 0x04

This will clear all AD1 and AW1 data stored in the board.

3.8.8 TS1 Command: Test Board

This command is used to initiate a test on a particular Andon sign. All rows will begin a test mode, which will start with a video test of the LED by setting them all to Amber. The board will then cycle through all bins 1 –16 displaying bin number and values set in the bin with the AW1 commands. This mode will continue until a CLR, NRM or RST command is received.

TS1 command syntax:

ESC 'TS1' yy 'R'

Where:

ESC = (required) the ASCII control code escape

TS1 = (required) ASCII string 'TS1' (stands for Test Mode 1)

Example:

0x02 0306 0x1b TS12x04

This will put the board into test mode.

3.8.9 NRM overview: Normal State

The NRM command will be used to restore a specified row, or an entire light board, to its Normal State. This command will end a previous CLR or TX1 command for all rows, or a selected row, and return it to its previous state.

Normal State is defined as the value and condition defined for bin1 set by the AW1 Command.

NRM command syntax:

ESC 'NRM' yy 'R'

Where:

ESC = (required) the ASCII control code escape

RST = (required) ASCII string 'RST' (stands for Andon sign Reset)

yy = (optional) is the Andon cell row number, in ASCII decimal (1 to 99)

R = (optional) the character 'R' (stands for row)

Example:

0x02 0207 0x1b NRM3R 0x04

This will establish that row 3 will be restored to its previous display state before command was issued.

3.8.10 CLR Command: Clear

The CLR Command is used to clear the display of a specified row or the entire board. The CLR command will not remove any values from non-volatile memory. The CLR command will only blank the current display of Andon cells or text messages.

CLR command syntax:

ESC 'CLR' yy 'R'

Where:

ESC = (required) the ASCII control code escape

CLR = (required) ASCII string 'CLR' (stands for Andon sign clear)

yy = (optional) is the Andon cell row number, in ASCII decimal (1 to 99)

R = (optional) the character 'R' (stands for row)
If Row "0" is specified the entire board will be cleared.

Examples:

0x02 0904 0x1b CLR3R 0x04

This will establish that row 3 display will be turned blank.

0x02 0904 0x1b CLR 0x04

This will blank the entire board.

3.8.11 SL1 Command: Stack Light 1

This command is used to tell the Andon display to change the status of a stack light.

SL1 command syntax:

ESC 'SL1' yy 'R' a 'A' EOT

Where:

yy = (required) the Stack light row number, in ASCII decimal (values: 1 to max row -- 1 being the top-most row)

R = (required) the character 'R' (stands for row)

a = (required) the Andon cell attribute

where a is:

0 = off

1 = on (default if 'A' is unspecified)

2 = normal flash (1 second cycle)

3 = fast flash (1/2 second cycle)

A = (required) the character 'A' (stands for attribute)

Example: (enabling stack light 2)

0x02 0501 0x1b SL12R1A 0x04

This will establish that row 2 of the stack light will be turned on.

APPENDIXES

APPENDIX A: COMMAND EXAMPLES

A.1: Command Summary

The following list is a summary of the commands covered in this manual:

System Commands

SBT
SGU
SFD
SVN
SMC

ANDON Cell Commands

AC1
AC2
AW1

Numeric/Clock Display

NUM
TIM

Music System Commands

TU1
TU2
VL1

Stack Light Control Command

SL1

955 MARQUEE Commands

TX1
RST
AD1
AW1
AC1
CHR
DEL
TS1
NRM
CLR

A.2: Summary of Command Syntaxes

System Commands

(SBT) Set Board Type

ESC 'SSN' sssss ESC 'SBT' nnn

(SGU) Set Group And Unit

ESC 'SSN' sssss ESC 'SGU' gguu gguu gguu gguu

(SFD) Set Fader Level

ESC 'SSN' sssss ESC 'SFD' gg 'G'

(SVN) Redisplay Boot Time Information

ESC SVN

(SMC) Set Auto-Center Option

ESC 'SMCnC' Where n = 0 or 1

ANDON Cell Control Commands

(AC1) ANDON Cell Control Variant 1

ESC 'AC1' yy 'R' xx 'C' a 'A' p 'P' s 'S' bb 'B'

(AC2) ANDON Cell Control Variant 2

ESC 'AC2' yy.y 'R' xx.x 'C' a 'A' p 'P' s 'S' (cell option)

ESC 'AC2' yy.y 'R' xx.x 'C' [wxyz] a 'A' (text option)

(AW1) ANDON Cell Write Command

ESC 'AW1' yy 'R' xx 'C' bb 'B' [xyz]

Numeric/Clock Commands

(NUM) Numeric Display

ESC 'NUM' nnnn N a 'A' (variant 1)

ESC 'NUM' yy 'R' xx 'C' nnnn N a 'A' (variant 2)

ESC 'NUM' yy.y 'R' xx.x 'C' nnnn N a 'A' (variant 3)

(TIM) Time Display

ESC 'TIM' hh:mm:ss 'T' yyyy-mm-dd 'D' a 'A'

Music System Commands

(TU1) Tune Selection Normal Speed

STX gg uu ESC 'TU1' ttt 'T' sss 'S' rr 'R' ddd 'D' p 'P' EOT

(TU2) Tune Selection Fast Speed
STX gg uu ESC 'TU2' ttt 'T' sss 'S' rr 'R' ddd 'D' p 'P' EOT

(VL1) Volume Control
STX gg uu ESC 'VL1' vvv 'V' c 'C' EOT

Stack Light Command

(SL1) Stack Light 1
STX gg uu ESC 'SL1' yy 'R' a 'A' EOT

MARQUEE Commands

952/954 - Alpha-Numeric Marquee
Refer to section [A.3](#) of the Appendix for 952/954 Marquee command examples.

955 - Tri-Color Alpha-Numeric Marquee

(TX1) Text Command
STX gg uu ESC 'TX1' yy 'R' a 'A' p 'P' s 'S' f 'F' [xyz]

(RST) Restore Command
STX gg uu ESC 'RST' yy 'R'

(AD1) Column Definition & Sizing
STX gg uu ESC 'AD1' yy 'R' [cc,cc,cc,cc,cc...cc]

(AW1) Write Values and Attributes to ANDON Cells
STX gg uu ESC 'AW1' yy 'R' xx 'C' p 'P' s 'S' f 'F' j 'J' a 'A' [xyz] bb 'B'

(AC1) Display/Activate Attributes for Values Stored in ANDON Cells
STX gg uu ESC 'AD1' yy 'R' [cc,cc,cc,cc,cc...cc]

(CHR) Character Create
STX gg uu ESC 'CHR' yy 'R' f 'F' c b1, b2, b3, b4, b5, b6, b7 EOT

(DEL) Delete Command
STX gg uu ESC 'DEL' EOT

(TS1) Test Board
STX gg uu ESC 'TS1' yy 'R'

(NRM) Normal
STX gg uu ESC 'NRM' yy 'R'

(CLR) Clear
STX gg uu ESC 'CLR' yy 'R'

A.3: Additional Command Examples

Command Examples for Using NAB-6 to Control 952/954 Marquee Display

The following command examples are shown with a GROUP address of 01 and a UNIT address of 02.

Example #1:

Sends a string of 5 characters to the alpha-numeric display marquee. This command will display the characters 12345 on the display. A carriage return sent before the characters ensures that the display is at the first character on the line before the characters 12345 are displayed:

```
$02 $30 $31 $30 $32 $0D $31 $32 $33 $34 $35 $04
```

Example #2:

Sends a message to the display that will replace the first 2 characters on the display with the characters 'AB':

```
$02 $30 $31 $30 $32 $0D $41 $42 $04
```

Example #3:

Sends a message to the display that will clear the display and then display the characters 'AB' starting at the fifth character position:

```
$02 $30 $31 $30 $32 $0C $14 $30 $35 $41 $42 $04
```

Example #4:

Sends a message to the display that will clear the display and then display the character 'A' flashing at the fifth character position on the display, and then the character '1', non-flashing, after the 'A':

```
$02 $30 $31 $30 $32 $0C $14 $30 $35 $10 $41 $12 $31 $04
```

Command Examples for Using NAB-6 to Control Andon Cells

The following command examples are shown with a GROUP address of 03 and a UNIT address of 04.

Example #1:

Select the Andon cell in row 3, column 5, set to fast flashing, color to amber.

```
$02 $30 $33 $30 $34 $1B $41 $43 $31 $33 $52 $35 $43 $33 $41 $32 $50 $04  
STX '0' '3' '0' '4' ESC 'A' 'C' '1' '3' 'R' '5' 'C' '3' 'A' '2' 'P' EOT
```

Command Examples for Using NAB-6 to Control Music Maker

The following command examples are shown with a GROUP address of 06 and a UNIT address of 07.

Example #1:

Play tune #23 on the music maker.

```
$02 $30 $36 $30 $37 $1B $54 $55 $31 $32 $33 $54 $04  
STX '0' '6' '0' '7' ESC 'T' 'U' '1' '2' '3' 'T' EOT
```

APPENDIX B: ASCII CONTROL CODES

In the "Key" column in the table below, the caret (^) before each character represents the CTRL key on the keyboard. (Exception: \$1E -- the second caret is the actual symbol. This is entered by simultaneously pressing ctl-shift-6).

Hex	Dec	Key	ASCII Name	ASCII Description	SCC Usage
\$00	00	^@	NUL	Null	
\$01	01	^A	SOH	Start of Header	
\$02	02	^B	STX	Start of Text	STX
\$03	03	^C	ETX	End of Text	ETX
\$04	04	^D	EOT	End of Transmission	EOT
\$05	05	^E	ENQ	Enquiry	ENQ
\$06	06	^F	ACK	Acknowledge	ACK; also set specific cell
\$07	07	^G	BEL	Bell	Sound alarm
\$08	08	^H	BS	Backspace	BS
\$09	09	^I	HT	Horizontal Tab	Cursor forward 1 position
\$0A	10	^J	LF	Line Feed	LF
\$0B	11	^K	VT	Vertical Tab	
\$0C	12	^L	FF	Form Feed	Cursor home & clear display
\$0D	13	^M	CR	Carriage Return	CR
\$0E	14	^N	SO	Shift Out	
\$0F	15	^O	SI	Shift In	Set 16 discrete outputs
\$10	16	^P	DLE	Data Link Escape	Flash ON
\$11	17	^Q	DC1	Device Control 1	Sound secondary alarm
\$12	18	^R	DC2	Device Control 2	Flash OFF
\$13	19	^S	DC3	Device Control 3	Set 1st 4 digit BCD output
\$14	20	^T	DC4	Device Control 4	Set Cursor Position
\$15	21	^U	NAK	Negative Acknowledge	NAK
\$16	22	^V	SYN	Synchronous Idle	Display attribute off
\$17	23	^W	ETB	End Transmission Block	Display attribute on
\$18	24	^X	CAN	Cancel	
\$19	25	^Y	EM	End of Medium	
\$1A	26	^Z	SUB	Substitute	Set 32 discrete outputs
\$1B	27	^[ESC	Escape	ESC
\$1C	28	^\ ^_	FS	File Separator	
\$1D	29	^]	GS	Group Separator	Set specific cell
\$1E	30	^^	RS	Record Separator	Set 2nd 4-digit BCD output
\$1F	31	^_	US	Unit Separator	Set specific cell (2nd of 16)

APPENDIX C: GLOSSARY of ACRONYMS and ABBREVIATIONS

AKA	Also Known As
DRAM	Dynamic Random Access Memory
EEPROM	Electrically Erasable Programmable Read Only Memory
EOT	End of Transmission (ASCII control code)
ESC	Escape (ASCII control code)
ETX	End of Text (ASCII control code)
SCC	Static Controls Corporation
STX	Start of Text (ASCII control code)
TBD	To Be Determined