ComputerVoltageSource LCD Software Guide

June 25, 2006

Overview

The ComputerVoltageSource LCD Support module provides control for a 2 row by 16 character LCD.

Display Commands

Character display operations are controlled through eighteen commands. These commands control the active display position where characters are written, whether the display overwrites previous characters or scrolls, selects the character set to be displayed, and clears the display. A cursor is not displayed since this requires a character position and reduces the number of displayed characters.

At power-on the LCD is set to overwrite mode, character set0, and displays the message:

```
ComputerVoltSrce
(c) D.Brown 2006
```

The first LCD display byte received will clear the display.

Control characters 0x08 to 0x1f implement the following display functions:

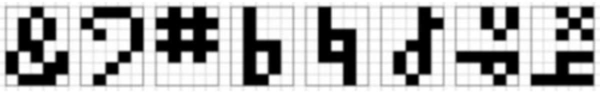
0x08: (backspace) Moves the active display position one backwards
0x09: (tab) Moves the active display position one forward
0x0a: (line feed) Clears the display and sets the active display position to beginning of line 1
0x0b: (vertical tab) Sets the active display position to beginning of line 2
0x0c: (form feed) Clears the display and sets the active display position to beginning of line 2
0x0d: (carriage return) Sets the active display position to beginning of line 1
0x0e: (shift out) Selects the lower characters set for display
0x0f: (shift in) Selects the upper characters set for display
0x10: Program Character0 <data>,<data> … <data> (8 bytes total)
0x11: Program Character1 <data>,<data> … <data> (8 bytes total)
0x12: Program Character2 <data>,<data> … <data> (8 bytes total)
0x13: Program Character3 <data>,<data> … <data> (8 bytes total)
0x14: Program Character4 <data>,<data> … <data> (8 bytes total)
0x15: Program Character5 <data>,<data> … <data> (8 bytes total)
0x16: Program Character6 <data>,<data> … <data> (8 bytes total)
0x17: Program Character7 <data>,<data> … <data> (8 bytes total)
0x18: Sets the display mode to overwrite; new characters overwrite previous
0x19: Sets the display mode to scroll; display scrolls for new characters
0x1a: Sets Absolute Position <data> (2 bytes total)
0x1b: Sets Relative Position <data> (2 bytes total)
0x1c: Reserved
0x1d: Reserved
0x1e: Reserved
0x1f: Write Message <data>,<data> … <data> (33 bytes total)

User-Programmable Characters

The display supports 8 user-programmable characters. These are selected by character codes 0x00 to 0x07. These characters have been pre-programmed with useful music characters and symbols and may be reprogrammed.

Default programmable characters	
0x00	Treble Clef
0x01	Bass Clef
0x02	Sharp symbol
0x03	Flat symbol
0x04	Natural symbol
0x05	Note symbol
0x06	dc (sideways for *decimal*)
0x07	hx (sideways for *hex*)

Default 5x7 programmable character font:



Control characters 0x10 to 0x18 define the user-programmable characters:

Each of the following eight byte control character programs a specific user programmable character independently:

Program Character	Programs character 0x00							
	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Byte 1	0	0	0	1	0	0	0	0
Byte 2	0	n/a	n/a	row7 b4	row7 b3	row7 b2	row7 b1	row7 b0
Byte 3	0	n/a	n/a	row6 b4	row6 b3	row6 b2	row6 b1	row6 b0
Byte 4	0	n/a	n/a	row5 b4	row5 b3	row5 b2	row5 b1	row5 b0
Byte 5	0	n/a	n/a	row4 b4	row4 b3	row4 b2	row4 b1	row4 b0
Byte 6	0	n/a	n/a	row3 b4	row3 b3	row3 b2	row3 b1	row3 b0
Byte 7	0	n/a	n/a	row2 b4	row2 b3	row2 b2	row2 b1	row2 b0
Byte 8	0	n/a	n/a	row1 b4	row1 b3	row1 b2	row1 b1	row1 b0

Program Character	Programs characters 0x00 – 0x07							
	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Byte 1	0	0	0	1	0	cc b2	cc b1	cc b0
Byte 2	0	n/a	n/a	row7 b4	row7 b3	row7 b2	row7 b1	row7 b0
Bytes 3 – 8 repeat as above for each character row								
Note: cc b2 – cc b0 specifies the user-programmed character at 0x00 to 0x07								

The 5x7 programmable character cell format is defined by the following table:

5 x 7 Character Cell Format					
	b4	b3	b2	b1	b0
row7	p	p	p	p	p
row6	p	p	p	p	p
row5	p	p	p	p	p
row4	p	p	p	p	p
row3	p	p	p	p	p
row2	p	p	p	p	p
row1	p	p	p	p	p
Note: pixel p displays if set to 1					

The Display Position command is a two byte sequence that will set an absolute or relative display position for the next character.

The absolute display position (0x0a) sets the position for the next character to be written in from the start of the display. The range is 0 – 31.

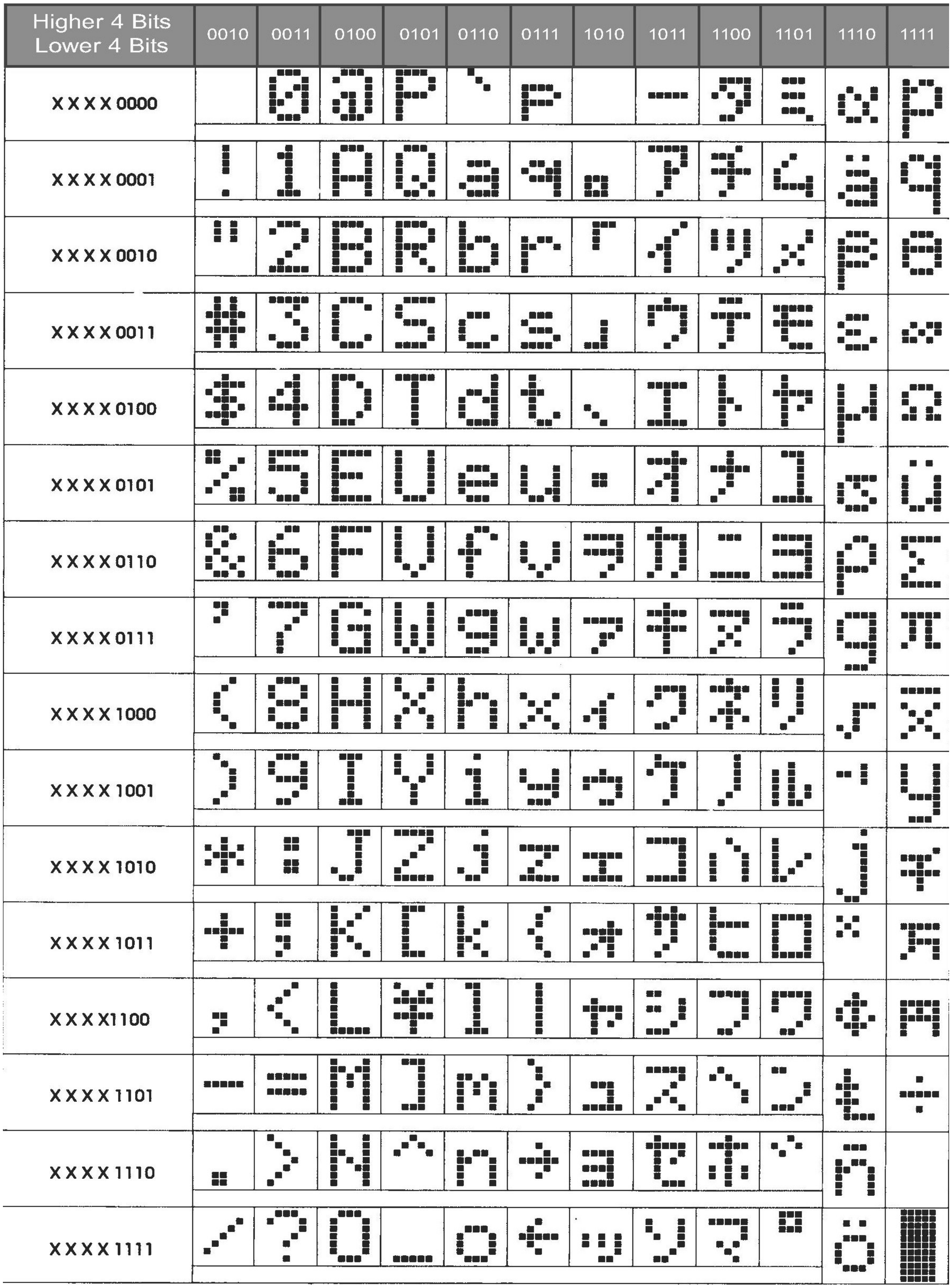
The relative display position (0x1b) sets the position for the next character to be written in from the current position. The range is 0-31. Numbers which are larger than the remaining characters in the display will wrap back to the beginning of the display. Thus 31 will move 1 position backwards to overwrite the last character. This is the same as -1 in two’s compliment. MIDI data is 7 bits so -1 and must be coded as -1&$1f to strip off the upper bits. Scrolling, if enabled, will not occur until characters reach the end of the display.

Display Position		Sets active display position 0x1A, 0x1B						
	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Byte 1	0	0	0	1	1	0	1	d0
Byte 2	0	0	0	b4	b3	b2	b1	b0
d0=0 for absolute position, d0=1 for relative position								
Note: b4 –b0 specifies the display position value 0 - 31								

The Write Message command (0x1f) is a 33 byte sequence that will store a new power-on message in the eeprom. The 32 remaining bytes will be the display character codes starting from the first character. The first character must be non-zero to indicate that a valid message is saved in the eeprom. Programmed characters should not be used since the characters will not have yet been programmed by the microcontroller. Note that 8 bits may be used to access nearly all characters in the ROM font which does not conform to MIDI protocol for 7 bit data. Characters 0xf8 – 0xff are recognized as MIDI real-time messages and are ignored. A credit message will be displayed at the completion of the write message sequence and the microcontroller will restart. The MIDI sysex termination byte (0xf7) does not need to be sent and any data beyond 33 bytes will be ignored. Any message in eeprom may be cleared by writing a new message with a first byte of 0x00.

Characters

The following characters are supported using a Lumex LCM-S01602DSF/C LCD module:



Higher 4 Bits Lower 4 Bits	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
XXXX0000		0	@	P	`	p		ー	タ	ミ	α	p
XXXX0001	!	1	A	Q	a	q	。	ア	チ	ム	ä	q
XXXX0010	"	2	B	R	b	r	「	イ	ツ	メ	β	θ
XXXX0011	#	3	C	S	c	s	」	ウ	テ	モ	ε	∞
XXXX0100	$	4	D	T	d	t	、	エ	ト	ヤ	μ	Ω
XXXX0101	%	5	E	U	e	u	・	オ	ナ	ユ	σ	ü
XXXX0110	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
XXXX0111	'	7	G	W	g	w	ァ	キ	ヌ	ラ	g	π
XXXX1000	(	8	H	X	h	x	ィ	ク	ネ	リ	√	x̄
XXXX1001	)	9	I	Y	i	y	ゥ	ケ	ノ	ル	$^{-1}$	y
XXXX1010	*	:	J	Z	j	z	ェ	コ	ハ	レ	j	千
XXXX1011	+	;	K	[	k	{	ォ	サ	ヒ	ロ	x	万
XXXX1100	,	<	L	¥	l	|	ャ	シ	フ	ワ	¢	円
XXXX1101	-	=	M	]	m	}	ュ	ス	ヘ	ン	£	÷
XXXX1110	.	>	N	^	n	→	ョ	セ	ホ	゛	ñ	
XXXX1111	/	?	O	_	o	←	ッ	ソ	マ	゜	ö	█

Notes:

1. Characters 0x00 to 0x07 are user programmable
2. Characters 0x08 to 0x1f are reserved for LCD commands
3. Characters 0x80 to 0x9f are blank (e.g. space).
4. MIDI data is restricted to 7 bits. Character set0 mode adds the 8th bit as a 0 and character set1 mode adds the 8th bit as a 1. For example, in character set1 mode, 0x00 is changed to 0x80 and displays as a space, 0x74 is changed to 0xf4 and displays as a Ω.

Communications

The CVS communicates to the display through a MIDI sysex message. The MIDI sysex command consist of a message byte (0xf0) followed by the identification of the display (0x7d). All subsequent bytes will be routed to the display until a termination byte (0xf7) is received. Note that if MIDI out will not used the MIDI sysex does not ever need to be terminated. All MIDI messages excluding display messages will be routed to MIDI out. Any connected MIDI device will never be sent a display message. The LED will flash for each MIDI byte sent and can be used as a general MIDI traffic indicator.

Using MIDI sysex messages, the following AtomPro28 commands can be used:

sethserial h31200,h8databits,hnoparity,h1stopbits
This statement initializes the hserial system for MIDI communications.

hserout [0xf0,0x7d,0x1e,0x1b,"Hello World",0xf7]
This statement clears the display, enables scrolling, and displays: `Hello World`