

LCD Character Set

Higher 4bit Lower 4bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000		0	a	P	`	P		-	9	E	O	P	
xxxx0001		!	1	A	Q	a	9	.	7	4	3	q	
xxxx0010		"	2	B	R	b	r	"	i	u	x	e	e
xxxx0011		#	3	C	S	c	s	_	o	t	e	e	e
xxxx0100		\$	4	D	T	d	t	\	i	t	t	u	o
xxxx0101		%	5	E	U	e	u	.	o	t	u	e	o
xxxx0110		&	6	F	V	f	v	7	o	u	e	p	z
xxxx0111		'	7	G	W	g	w	7	t	z	u	q	π
xxxx1000		(8	H	X	h	x	4	o	t	u	u	z
xxxx1001)	9	I	Y	i	y	o	t	u	u	u	u
xxxx1010		*	:	J	Z	j	z	u	u	u	u	i	z
xxxx1011		+	:	K	L	k	l	o	t	u	u	u	u
xxxx1100		,	<	L	*	l	l	u	u	u	u	u	u
xxxx1101		-	=	M	I	m	i	u	u	u	u	u	u
xxxx1110		.	>	N	^	n	^	o	t	u	u	u	u
xxxx1111		/	?	O	_	o	*	u	u	u	u	u	u

Custom Characters

The display module has 64 bytes of character generator RAM, which provide 8 user-definable characters. These characters correspond to character codes 00h through 07h. Character codes 08h through 0Fh also correspond to the eight user-defined characters.

Each user-defined character is a 5X8 character in an 8X8 block. The mapping between the CG RAM locations and the displayed character is shown in the following figure. If the 5X10 dot display arrangement is used, only 4 user-defined characters are supported and the characters are defined with 11 bytes each.

CG Address	CG Data	
	D7-----D0	
x+0	. . . 1 1 1 1 1	*****
x+1	. . . 1 0 0 0 0	*
x+2	. . . 1 0 0 0 1	* *
x+3	. . . 1 1 1 1 1	*****
x+4	. . . 1 0 0 0 1	* *
x+5	. . . 1 0 0 0 0	*
x+6	. . . 1 0 0 0 0	*
x+7	. . . 0 0 0 0 0	(cursor line)

Where "x" is the base address for the character: CG0=00h, CG1=08h, CG2=10h, CG3=18h, CG4=20h, CG5=28h, CG6=30h, CG7=38h.

As you can see in the above figure, each byte of CG RAM represents 1 row of pixels in the character, with D0 being the rightmost pixel, and D4 being the leftmost (the upper 3 bits are not displayed). The first address for a character is the topmost row, while the 7th is the bottom. (The 8th is ORed with the cursor underbar). A '1' in a pixel position becomes a dark pixel on the display. CG RAM occupies a separate address space from the data display (DD) RAM. One must set the (first) CG address before writing any CG data. Each write automatically increments/decrements the CG address pointer to the next location. Remember to set the display back to DD addressing mode before trying to write characters to be displayed.

As an example, we'll define character #3 to be the above symbol by writing the CG address to the start of the character's RAM then writing the successive rows of pixel data.

RS	R/W	D7----D0	
0	0	01011000	Set CG RAM address to char #3
1	0	00011111	Write top row of pixels
1	0	00010000	:
1	0	00010001	:
1	0	00011111	:
1	0	00010001	:
1	0	00010000	:
1	0	00010000	Write bottom row of pixels
1	0	00000000	Write cursor row (descender)