LCD Technical FAQ

Contents:

3.2) Character Set

A picture of the character set scanned from a datasheet, chset.gif, can be found in ftp://ftp.armory.com/pub/user/rstevew/LCD/lcdfaq.zip or, on the web, visit Ian Harries' LCD tutorial and see the character set at http://www.doc.ic.ac.uk/~ih/doc/lcd/display.html

None of the standard ASCII control codes [chr(1)-chr(31), chr(127)] are implemented.

Standard ASCII is used for chr(32) <space> through chr(125) '}' (<00100000b..01111101b>), except that tilde (~) is replaced by a yen symbol.

The lower case characters do not have decenders. This is because some LCD's (those with 5x7 dots or 5x8 dots) would chop off the bottom. Lower case characters with decenders appear near the top of the character table for use on modules which have 5x11 dot matrices. You can access them readily by adding 128 (<10000000b>, 80x) if you want decenders and have a 5x11 dot unit that will properly display them. You can create squished "sort-of" decenders of your own in character ram to eliminate the awkwardlooking "tall" lowercase letters provided.

Eight user-defined characters are displayed by chr(0) through chr(7), and redundantly with chr(8) through chr(15). These are sometimes used to implement a comma and lower case characters g, j, p, q, and y with a decender into the cursor row on 5x8 units.

```
                               Font Table
                               ---- -----
LSN   x0  x1  x2  x3  x4  x5  x6  x7  x8  x9  xA  xB  xC  xD  xE  xF
```

```
MSN +----------------------------------------------------------------
0x  | cg0 cg1 cg2 cg3 cg4 cg5 cg6 cg7 cg0 cg1 cg2 cg3 cg4 cg5 cg6 cg7
1x  | <-------------------------- UNDEFINED -------------------------->
2x  |      !   "   #   $   %   &   '   (   )   *   +   ,   -   .   /
3x  |  0   1   2   3   4   5   6   7   8   9   :   ;   <   =   >   ?
4x  |  @   A   B   C   D   E   F   G   H   I   J   K   L   M   N   O
5x  |      Q   R   S   T   U   V   W   X   Y   Z   [  (*)  ]   ^   _
6x  |  `   a   b   c   d   e   f   g   h   i   j   k   l   m   n   o
7x  |      q   r   s   t   u   v   w   x   y   z   {   |   }  --> <--
8x  | <-------------------------- UNDEFINED -------------------------->
9x  | <-------------------------- UNDEFINED -------------------------->
Ax  |     (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*)
Bx  |  -  (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*)
Cx  | (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*)
Dx  | (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*) (*)
Ex  | (*) (*) (*) (*) (*) (*) (*) (g) (*) (*) (j) (*) (*) (*) (*) (*)
Fx  | () (q) (*) (*) (*) (*) (*) (*) (*) (y) (*) (*) (*) (*) (*) (*)
```

Notes:

cg0-cg7
: User-definable characters

(chr)
: Lower-case character with descenders

(*)
: See list below:

Hex	Decimal	Meaning
5C	92	Yen
A0	160	(blank)
A1	161	katakana
:	:	:
A5	165	katakana (looks like dot in center)
:	:	:
DF	223	katakana (looks like degree symbol)
E0	224	LC alpha
E1	225	"a" with umlaut ("a) (German)
E2	226	LC beta (or German es-szet) with descender
E3	227	LC epsilon
E4	228	LC mu with descender
E5	229	LC sigma

E6	230	LC rho
E7	231	"g" with descender
E8	232	radical (square root sign)
E9	233	"- |" (Katakana?)
EA	234	"j" with descender
EB	235	tiny 3 by 3 'x' in upper left corner
EC	236	cent sign
ED	237	pounds sterling
EE	238	"n" with tilde (Spanish) or overbar
EF	239	"o" with umlaut ("o) (German)
F0	240	"" with descender
F1	241	"q" with descender
F2	242	LC theta
F3	243	infinity (lazy-8)
F4	244	LC omega
F5	245	"u" with umlaut ("u) (German)
F6	246	UC sigma
F7	247	LC pi
F8	248	"x" with overbar
F9	249	"y" with descender
FA	250	katakana
FB	251	katakana
FC	252	katakana
FD	253	division symbol (:-)
FE	254	(blank)
FF	255	solid black cursor

(Font table by Doug Girling)

3.3) Instruction Set

Binary data from bit7 to bit 0. If using a 4-bit interface, bit7-bit4 of data are sent first, then bit3-bit0, on sucessive enable cycles. No delay is required between these cycles (except that the minimum E time (450ns) and TcycE time (1us) must be met).

3.3.1) Write Operations

For all write procedures, RS=0 (except as noted), R/~W=0, command/data from CPU to LCD on bit7 to bit0.

Clear display (00000001)

Clears display and returns cursor to home position (address 0). Execution time: 1.64ms

Home cursor (0000001x)

Returns cursor to home position, returns a shifted display to original position. Display data RAM (DD RAM) is unaffected. Execution time: 40us to 1.64ms

x
: don't care

Entry mode set

000001is

Sets cursor move direction and specifies whether or not to shift display. Execution time: 40us

i=1
: Increment

i=0
: Decrement DD RAM address by 1 after each DD RAM write or read.

s=1
: Display scrolls in the direction specified by the "i" bit when the cursor is at the edge of the display window

On/off control

00001dcb

Turn display on or off, turn cursor on or off, blink character at cursor on or off. The contents of Display Data RAM are not altered when the display is turned off. If it is turned on again, the previously displayed text will reappear. When the cursor is on and blink is off, the cursor is an underscore in line 8 (or line 11 for 5x11 displays). When blink is on, the cursor is a solid box alternating with the character in DD RAM at that position. Execution time: 40us

d=1
: Display on.

c=1
: Cursor on

b=1
: blink character at cursor position

The contents of Display Data RAM are not altered when the display is turned off. If the display is turned on again, information displayed previously will reappear.

Cursor/shift

0001srxx

Move cursor or scroll display without changing display data RAM. Execution time: 40us

s=1
: scroll display

s=0
: move cursor.

r=1
: to the right

r=0
: to the left.

x=
: don't care

Function set (001dnfxx)

Set interface data length, mode, font. Execution time: 40us

d=1
: 8-bit interface, d=0: 4-bit interface.

n=1
: 1/16 duty, n=0: 1/8 or 1/11 duty (multiplex ratio). For 2-line displays, this can be thought of as controlling the number of lines displayed (n=0: 1-line, n=1: 2 or more lines) except for 1x16 displays which are addressed as if they were 2x8 displays--two 8-character lines side by side.

f=1
: 5x11 dots, f=0: 5x8 dots.

Character RAM Address Set (01aaaaaa)

To read or write custom characters. Character generator (CG) RAM occupies a separate address space from the DD RAM. Data written to, or read from the LCD after this command will be to/from the CG RAM. The address pointer is incremented after each write, so consecutive bytes (rows) can be sent. Execution time: 40us

aaaaaa
: 6-bit CG RAM address to point to.

Display RAM Address Set (1aaaaaaa)

Reposition cursor. Display Data (DD) RAM occupies a separate address space from the CG

RAM. Data written to, or read from the LCD after this command will be to/from the DD RAM. The address pointer is incremented after each write, so consecutive bytes can be sent. Execution time: 40us

aaaaaaa

7-bit DD RAM address to point to. On two line models (and most 16x1), the command can be interpreted this way:
1laaaaaa
l=line #
a=6-bit column #

Write Data to CG or DD RAM (dddddddd (RS=1))

Data is written to current cursor position and (DD/CG) RAM address (which RAM space depends on the most recent CG_RAM_ Address_Set or DD_RAM_Address_Set command). The (DD/CG) RAM address is incremented/decremented by 1 as determined by the "entry mode set" command. Execution time: 40us for display write, 120us for character generator ram write. Note that RS=1 for this command.

dddddddd

8-bit character code

3.3.2) Read Operations

For all read procedures, RS as noted, R/~W=1, bit7 to bit0 output from LCD to CPU.

Read Busy Flag (baaaaaaa (RS=0))

Read the status of the busy flag, and the value of the RAM address currently being pointed at. Execution time: 1 cycle

b=1

busy
b=0
OK to send

aaaaaaa

7-bit current (DD/CG) RAM address counter (as in "character RAM address set" or "display RAM address set"). DB lines must be inputs (or pulled high with pullup resistors) to be driven by the module.

Read Data from CG or DD RAM (dddddddd (RS=1))

Data is read from current (DD/CG) RAM address position, and the RAM address is automatically incremented/decremented by 1 as determined by the "entry mode set" command. NOTE that the display/cursor is not shifted on data reads. Execution time: 40us for display reads, 120us for

character generator ram reads
dddddddd
DB lines must be inputs (or pulled high with pullup resistors) to be driven by the module. An 8-bit character code will be read back from LCD RAM.
