LCD Control Made Easy

Introduction

A liquid crystal pixel consists of liquid crystal material which is filled between a metallic back plane and a front electrode. The liquid crystal material rotates the plane of polarization of light passing through it by an amount depending on the electric field between the back plane and the front electrode. However, to prevent degeneration of the material through electrolysis, a DC field is not applied. The back plane is driven by a square wave and the front plane is driven with another square wave which is either in phase with the back plane (leading to zero field across the material) or in phase opposition to the back plane (maximum field). In order to display a character, a set of pixels need to be turned on or off depending on the font.

Obviously, the control for implementing all this is cumbersome. To manage LCD displays, Hitachi introduced a micro-controller (HD44780U) about a decade ago, which was pre-programmed to control such displays. Apart from providing appropriate square waves to pixels, it has a local memory which stores the characters to display and a programmable memory which stores fonts. The interface to this micro-controller has become an industry standard and most small LCD displays use this interface. The display provided with the Pt51 kit (JHD 162A) is no exception.

Thus, controlling the LCD display is just a matter of sending commands and data from the micro-controller on Pt51 (AT 89C5131A) to the micro-controller on the display (HD44780U). JHD 162A can display up to 16 characters per line in two lines (hence its name – 16 2A). When the data to be displayed is sent to the LCD board, it is stored in the RAM of the on-board micro-controller (HD44780U). The LCD uC must be told that new data has arrived and should be given time for it to read this data and to act on it, before further data can be sent to it. The interfacing protocol is built around these requirements.

Hardware interfacing

The 44780 interface uses three control lines. The RS line is the register select signal which decides whether a command or data byte is being sent or received. This line is low for command transfer and high for data transfer. The RW line signals whether a read or write operation is desired. This line is driven high for a read operation and low for writing. The third line is EN or enable. This provides hand shake for data transfer. The actual data can be sent over either 8 data lines, or over 4 data lines in two successive transfers. The Pt51 board uses three lines from Port P0 for the control lines. P0.0 is connected to RS, P0.1 is connected to RW and P0.2 is connected to EN. Port P2 provides 8 lines for data.

The AT89C5131A works on 3.3V, while the display uC works at 5V. Fortunately, the port pins on the AT89C5131A are 5V compliant. Pull up resistors are used on the Pt51 board to

make the voltage levels compatible between the two units.

Software interfacing and Timing

Timing for data transfer is accomplished through the EN line. This line is driven low at the start of data transfer. The RS, RW and data lines are now driven to their desired values. EN is then taken high. This tells the 44780 to look for a command/data transfer. Finally, EN is taken down again. 44780 acts on the command/data only when EN has a downward transition.

To write self documenting programs, it is helpful to define the following equates:

```
RS EQU P0.0
RW EQU P0.1
EN EQU P0.2
Data EQU P2
```

Obviously, most of the time we shall be writing to the LCD. The only read action which is commonly performed is to get the status byte from the LCD uP. If we perform a read with the RS line at 0, it returns a status byte. The most significant bit of this byte indicates the busy status of the LCD uC. Fresh data can be sent to the display only when this bit is zero.

Command Set for 44780

The position of the first non-zero bit (from MSB) determines what kind of command it is. If DB7 = 1, it is the display data RAM address set command, where the rest of the bits specify the address in the display RAM from where the next read/write will be carried out.

Similarly, if DB6 = 1 is the first non zero bit, it is the character generator address set command.

If DB7, DB6 are 0 and DB5 = 1, it is the function set command, where DB4 specifies the number of data bits used by the interface, DB3 gives the number of lines in the display and DB2 picks the font.

If DB4 is the first non zero bit, it is the shift command. DB3 gives whether the display or the cursor will be shifted, while DB2 signifies whether a right or left shift is desired.

If DB3 is the first non zero bit in the command, it is the display switch command. DB2 determines if the whole display will be turned on/off, DB1 is for turning the cursor on/off while DB0 turns blinking of the cursor on or off.

If DB2 is the first non-zero bit, it is the input set command. DB1 is interpreted as increment/decrement mode for entering data while DB0 signifies whether the display should shift after entering the new character by one position.

If DB1 is the first non-zero bit, it returns the cursor to the first position. If DB0 is the only non-zero bit, it clears the screen.

Configuring the LCD

To configure the LCD, we need to send a function set command first. (Without this command, the display will not know whether to interpret the rest of the communication as 8 bit data or 4 bit data). Function set has the following format:

DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	DL	N	F	-	-

Where DB 765 = 001 signifies that it is a function set command, DL is 1 for 8 bit data interface and 0 for 4 bit interface, N is 1 for a 2 line display, 0 for a single line display and F is 1 for a 5x10 font and 0 for a 5x7 font.

For a 2 line display with 5x7 font on an 8 bit interface, we need to send 00111000 = 38H.

We next turn the display on by sending the Display switch command. Display switch has the following format:

DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	D	C	B

Where D stands for Display, C for cursor and B for blink. To turn on the display with a cursor which is not blinking, we send 00001110 or 0EH.

Finally, we can program the display that every time we send a character, the cursor automatically shifts to the right by one position. This is accomplished by the Input Set command. This command has the following format:

DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	I/D	S

Here I/D stands for auto increment/decrement of cursor position while S is for shift control. For just incrementing the cursor position, we send 00000110 or 06H.

Thus configuring/initializing of the display just involves sending 38H, 0EH and 06H to the display to set it to the mode described above. Of course, each command should be sent after ensuring through a status read that the display uC is no more busy. One may optionally clear the screen after this by sending a clear screen command (00000001).

Displaying Text on the LCD

After initialization, displaying text is just a matter of writing each character to the display in data mode. Of course, before sending each character, one must ensure that the LCD is not busy.

If we want to position the text at a particular place in the display, the data address must be set accordingly. The sixteen characters in the first line reside at data address 00H to 0FH. However, the second line does not begin from the address 10H! Addresses for the second line span the address range 40H to 4FH. This is to accommodate longer line displays in future versions of the LCD controller. To set the data address, we use the command where DB7 = 1 and the remaining bits provide the data address to be set.

Looking at the command set, you can figure out how to insert text at the left, at right or in the middle of a line.