LCD Board C & Assembly Strategy

Introduction

This document will suggest basic strategies for creating 'C' and Assembly code for the LCD E-Block. Because this E-Block can be used with a variety of upstream boards (e.g. PICmicro multi-programmer board), this document will not provide all of the information required. See the "further reading" section for more complete reference information.

The LCD E-Block uses a Samsung KS0066U controller chip to control the actual LCD display. This controller chip accepts 8-bit commands and data, but only 4-bits of the data bus are connected to the E-Block connector so commands and data must be sent using the controller chip's 4-bit interface mode. This data bus is connected to pins 1-4 of the connector, so (for example) if the E-Block is connected to Port B of a PICmicro multi-programmer E-Block, this data bus would correspond to pins B0 to B3 of the PICmicro.

There are 2 other lines on the connector - "RS" (register select) and "E" (enable). RS is used to indicate whether a command (RS = 0) or data (RS = 1) is being sent. When the controller chip receives a high-to-low transition on the E line, it samples the data bus - thus the E pin is essentially used to clock the data into the LCD controller device. RS and E are connected to pins 5 and 6 of the connector (and so would correspond to B4 and B5 in the PICmicro example above).

Implementing a strategy

The following strategy is specific for a PICmicro microcontroller, but should be adaptable to any upstream device.

Sending a command

Because a 4-bit interface mode is used, one 8-bit data byte needs to be sent as 2 separate nibbles (i.e. half a byte) - first the high nibble and then the low nibble.

We will assume that the "E" line is initially low. Here's the stages required to send the command 0x01 ("display clear") to the LCD controller using a PICmicro with Port B connected to the LCD E-Block.

1) Set Port B to 0x00
    - B5 is the E line and is low
    - B4 is the RS line and is low to indicate that it is a command
    - B0-3 is binary 0000, which is the high nibble of the command 0x01
2) B5 -> 1 (bring the E line high)
3) B5 -> 0 (bring it low again)
4) Set Port B to 0x01
    - B5 is the E line and is low
    - B4 is the RS line and is low to indicate that it is a command
    - B0-3 is binary 0001, which is the low nibble of the command 0x01
5) B5 -> 1
6) B5 -> 0
7) Wait for the command to be processed

The time required to process the command depends on which command is being processed and these timing values can be found in the LCD controller datasheet, but 2ms should be enough.

Sending data

Sending data is almost exactly the same procedure as listed above, except that the RS line must now be kept high to indicate that data rather than a command is being sent.

Initialising the LCD display

When power is first applied to the LCD display E-Block, the LCD controller chip will automatically begin its own initialisation routine that typically takes 30ms - so it is prudent to include this delay in your firmware just to make sure the LCD display is ready to receive commands before your initialisation routine is started.

There are a number of initialisation commands required to set up the LCD display correctly. We will present a typical LCD initialisation routine that allows a simple message to be displayed. Refer to the LCD controller chip's datasheet if you require another configuration (e.g. cursor display, screen shift). The list of 5 commands to be sent is as follows. Remember that the RS line must be maintained low (to indicate that a command is being sent), to toggle the E line (to clock the data) and to wait for an appropriate time to allow the command to be processed.

1) 0x22 - initialises the LCD display
2) 0x2C - display is 2 lines
3) 0x0C - display on, cursor off, blink off
4) 0x01 - clear the display
5) 0x06 - increment mode, shift off

This initialisation is suggested by the KS0066U datasheet, but there does seem to be some discrepancy between other suggested initialisation techniques. Here is an alternative if the one above fails to work:

1) 0x33
2) 0x32
3) 0x2C
4) 0x0C
5) 0x01
6) 0x06

Writing a value to the LCD display

The LCD display uses an ASCII-like character set, so 'C' and Assembly users can typically refer to literal character constants (e.g. 'A') rather than their ASCII equivalents ('A' = 0x41, or 65 in decimal). Characters are sent to the LCD display in the same way as commands, but RS must be kept high to indicate that it is data rather than a command. "Hello!" can be displayed on the LCD display by using the following sequence:

1) 0x48 or 'H'
2) 0x65 or 'e'
3) 0x6C or 'l'
4) 0x6C or 'l'
5) 0x6F or 'o'
6) 0x21 or '!'

Other commands

The command can be found in the KS0066U datasheet. A particularly useful one is the "Set DDRAM Address" command which sets the cursor position for the LCD display.

Further reading

See the KS0066U datasheet for more information on the specific LCD controller commands available.

Another useful source of information is the "Assembly for PICmicros" and "C for PICmicros" CD ROMS available from Matrix Multimedia.