JUN
27

Interface MSP430 Launchpad with LCD Module (LCM) in 4 bit mode

A quick overview on interfacing MSP430 with an LCD module. In this case a typical 16x2 parallel LCD module with the Texas Instruments' MSP430 Launchpad development board.

LCDs are very easy to get, adafruit has a nice one, as does sparkfun. If you are going to buy one it would be useful consider these two characteristics:

- Supply Voltage

  The MSP430 (like most MCUs today) runs on 3.3-3.5V. If you get an LCD module that takes 5V then be prepared to either have separate voltage sources for your MCU and LCM or have the ability to step up/down your voltage source. Here's a link that has some ideas on how to do this. Texas Iinstruments has a document that describes 1.5V to 5V boost conversion mechanisms. For my own setup I just used approx 4.3V voltage source (from 3 AA batteries) to power the LCM and used a simple diode on the same source to achieve a forward drop to source the MSP430. This is ok for prototyping but hardly a long term solution.

- Serial Interface

  Some LCD modules also support a serial interface. This uses much fewer pins and is therefore great to use with MCUs that have a low pin count. However the interfacing for that is quite different than the and will not be discussed here.

This LCD for example takes a 3.3V supply source and has both a serial and parallel interface!

Useful Specs

HD44780 LCD Specification:

Here's a link to the full specification for the HD44780 protocol that LCMs follow. It has all the command/data formats and pin specifications. Sometimes the LCM will have it's own spec in which case that takes precedence.

MSP430 Pin Interface to LCD

Here's how the MCU pins are interfaced to the LCM. If you use different connections then update the #define's in the code to reflect your connections.

```
//
// MSP430 LCD interface
```

```
// | LCM |
// LCM Pin # | Function | MSP Pin #
//---------------------------
// PIN 5 | RW | GND
// PIN 4 | RS | P1.0
// PIN 6 | EN | P1.1
// PIN 14 | DB7 | P1.7
// PIN 13 | DB6 | P1.6
// PIN 12 | DB5 | P1.5
// PIN 11 | DB4 | P1.4
//
```

Notice how we kept an identity mapped data bus number to MSP pin translation. It just makes it easier to program the send data as can be seen below.

The other LCM pins connections are as follows:

Pin #16 | GND
Pin #15 | Vcc
Pin #3 | Outout from potentiometer for LCD contrast
Pin #1 | GND
Pin #2 | Vcc

Adafruit also has a great tutorial here on how to connect the pins of the LCD.

MSP430 LCD Driver Code

This is all the code for the MSP430 LCD driver. A simple flow is demonstrated in the main() function. This code should be copy-paste runnable on the value line part plugged into the shipping launchpad. If things aren't working it might be because your LCD module is slower. Try extending the __delay_cycles, first in the Initialization routine. If that doesn't fix it extend the delay time in the pulse module as well.

```
//
// MSP430 LCD Code
//
#include "msp430x20x2.h"

#define LCM_DIR P1DIR
#define LCM_OUT P1OUT

//
// Define symbolic LCM - MCU pin mappings
// We've set DATA PIN TO 4,5,6,7 for easy translation
//
#define LCM_PIN_RS BIT0 // P1.0
#define LCM_PIN_EN BIT1 // P1.1
#define LCM_PIN_D7 BIT7 // P1.7
#define LCM_PIN_D6 BIT6 // P1.6
#define LCM_PIN_D5 BIT5 // P1.5
#define LCM_PIN_D4 BIT4 // P1.4
```

```
#define LCM_PIN_MASK ((LCM_PIN_RS | LCM_PIN_EN | LCM_PIN_D7 |
LCM_PIN_D6 | LCM_PIN_D5 | LCM_PIN_D4))

#define FALSE 0
#define TRUE 1

//
// Routine Desc:
//
// This is the function that must be called
// whenever the LCM needs to be told to
// scan it's data bus.
//
// Parameters:
//
// void.
//
// Return
//
// void.
//
void PulseLcm()
{
    //
    // pull EN bit low
    //
    LCM_OUT &= ~LCM_PIN_EN;
    __delay_cycles(200);

    //
    // pull EN bit high
    //
    LCM_OUT |= LCM_PIN_EN;
    __delay_cycles(200);

    //
    // pull EN bit low again
    //
    LCM_OUT &= (~LCM_PIN_EN);
    __delay_cycles(200);
}

//
// Routine Desc:
//
// Send a byte on the data bus in the 4 bit mode
// This requires sending the data in two chunks.
// The high nibble first and then the low nible
//
// Parameters:
//
// ByteToSend - the single byte to send
//
// IsData - set to TRUE if the byte is character data
```

```
// FALSE if its a command
//
// Return
//
// void.
//
void SendByte(char ByteToSend, int IsData)
{
  //
  // clear out all pins
  //
  LCM_OUT &= (~LCM_PIN_MASK);

  //
  // set High Nibble (HN) -
  // usefulness of the identity mapping
  // apparent here. We can set the
  // DB7 - DB4 just by setting P1.7 - P1.4
  // using a simple assignment
  //
  LCM_OUT |= (ByteToSend & 0xF0);

  if (IsData == TRUE)
  {
    LCM_OUT |= LCM_PIN_RS;
  }
  else
  {
    LCM_OUT &= ~LCM_PIN_RS;
  }
  //
  // we've set up the input voltages to the LCM.
  // Now tell it to read them.
  //
  PulseLcm();

  //
  // set Low Nibble (LN) -
  // usefulness of the identity mapping
  // apparent here. We can set the
  // DB7 - DB4 just by setting P1.7 - P1.4
  // using a simple assignment
  //
  LCM_OUT &= (~LCM_PIN_MASK);
  LCM_OUT |= ((ByteToSend & 0x0F) << 4);

  if (IsData == TRUE)
  {
    LCM_OUT |= LCM_PIN_RS;
  }
  else
  {
    LCM_OUT &= ~LCM_PIN_RS;
  }

  //
```

```
    // we've set up the input voltages to the LCM.
    // Now tell it to read them.
    //
    PulseLcm();
}

//
// Routine Desc:
//
// Set the position of the cursor on the screen
//
// Parameters:
//
// Row - zero based row number
//
// Col - zero based col number
//
// Return
//
// void.
//
void LcmSetCursorPosition(char Row, char Col)
{
    char address;

    //
    // construct address from (Row, Col) pair
    //

    if (Row == 0)
    {
        address = 0;
    }
    else
    {
        address = 0x40;
    }

    address |= Col;
    SendByte(0x80 | address, FALSE);
}

//
// Routine Desc:
//
// Clear the screen data and return the
// cursor to home position
//
// Parameters:
//
// void.
//
// Return
//
// void.
//
```

```
void ClearLcmScreen()
{
    //
    // Clear display, return home
    //
    SendByte(0x01, FALSE);
    SendByte(0x02, FALSE);
}

//
// Routine Desc:
//
// Initialize the LCM after power-up.
//
// Note: This routine must not be called twice on the
// LCM. This is not so uncommon when the power
// for the MCU and LCM are separate.
//
// Parameters:
//
// void.
//
// Return
//
// void.
//
void InitializeLcm(void)
{
    //
    // set the MSP pin configurations
    // and bring them to low
    //
    LCM_DIR |= LCM_PIN_MASK;
    LCM_OUT &= ~(LCM_PIN_MASK);
    //
    // wait for the LCM to warm up and reach
    // active regions. Remember MSPs can power
    // up much faster than the LCM.
    //
    __delay_cycles(100000);

    //
    // initialize the LCM module
    //
    // 1. Set 4-bit input
    //
    LCM_OUT &= ~LCM_PIN_RS;
    LCM_OUT &= ~LCM_PIN_EN;

    LCM_OUT = 0x20;
    PulseLcm();

    //
    // set 4-bit input - second time.
    // (as reqd by the spec.)
    //
```

```
    SendByte(0x28, FALSE);

    //
    // 2. Display on, cursor on, blink cursor
    //
    SendByte(0x0E, FALSE);

    //
    // 3. Cursor move auto-increment
    //
    SendByte(0x06, FALSE);
}

//
// Routine Desc
//
// Print a string of characters to the screen
//
// Parameters:
//
// Text - null terminated string of chars
//
// Returns
//
// void.
//
void PrintStr(char *Text)
{
    char *c;
    c = Text;

    while ((c != 0) && (*c != 0))
    {
        SendByte(*c, TRUE);
        c++;
    }
}

//
// Routine Desc
//
// main entry point to the sketch
//
// Parameters
//
// void.
//
// Returns
//
// void.
//
void main(void)
{
    WDTCTL = WDTPW + WDTHOLD; // Stop watchdog timer

    InitializeLcm();
```

```
  ClearLcmScreen();

  PrintStr("Hello World!");

  while (1)
  {
    __delay_cycles(1000);
  }
}
```

Useful Links from this page

LCD Modules:

- Buying from adafruit and sparkfun
- HD447800 protocol specification

Power supply issues:

- Tips and tricks for converting between 3V and 5V
- Texas Instruments document on power for their MCUs

More Resources

- Other MSP430 LCD example can be found at www.43oh.com on this page.
- Texas Instruments (TI) MSP430 code example

Posted 27th June 2011 by Cache

Labels: LCD, MSP430 Launchpad

29 View comments