EC336: Embedded Systems

Interfacing Display Devices

Arvind S Kumar (15EC106), Pavan M (15EC137),
Samarth B (15EC143), Sripathi M (15EC149)

15 August 2018

Submitted to:

Prof. Ramesh Kini M, Prof. Arulalan Rajan
Department of Electronics and Communication Engineering

NITK, Surathkal

1 Goal of the Lab

The aim of this lab was to interface the chosen microcontroller with display devices such as LEDs, 7-segment displays and LCD. The micro controller chosen by our group is TI's MSP430G2.

2 Components Used

For this lab the components used are :

- MSP430 microcontroller
- LEDs
- 2.2 kΩ resistors
- 2 7-segment displays
- LCD

3 Board Details

Some of the features of the MSP430 are - (Refer Figure 1 and Figure 2 for more information)

- 16-bit RISC architecture
- Von-Neumann architecture
- Upto 16 MHz clock frequency
- Low power consumption and five power saving modes
- 16 general purpose registers

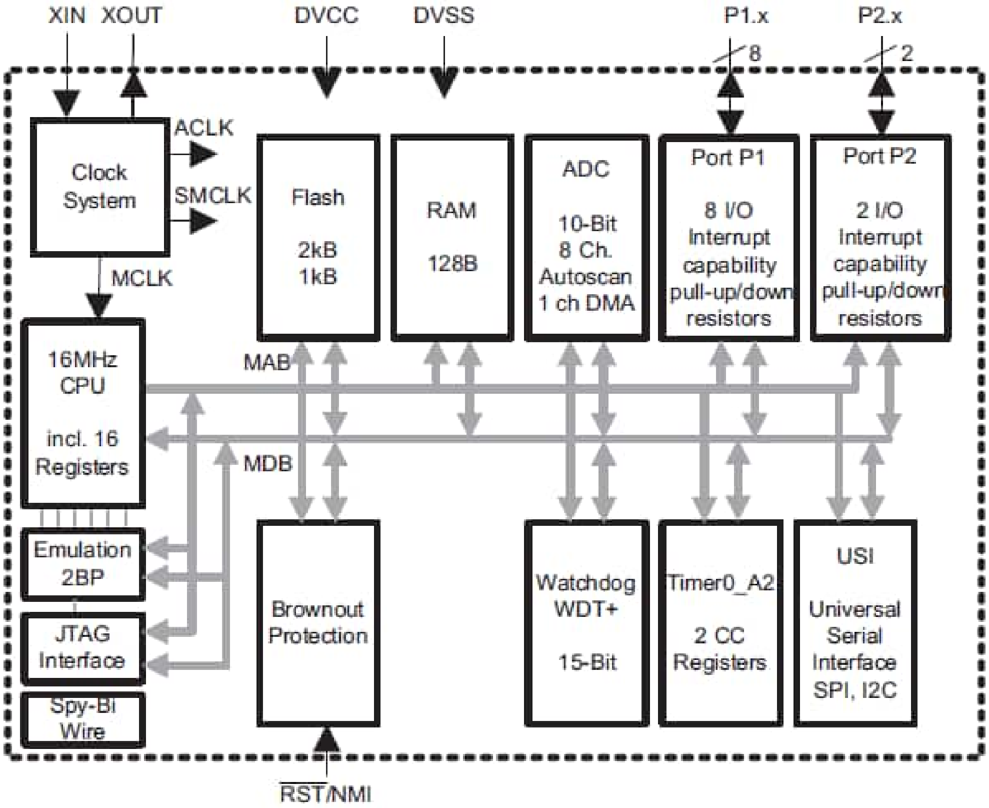
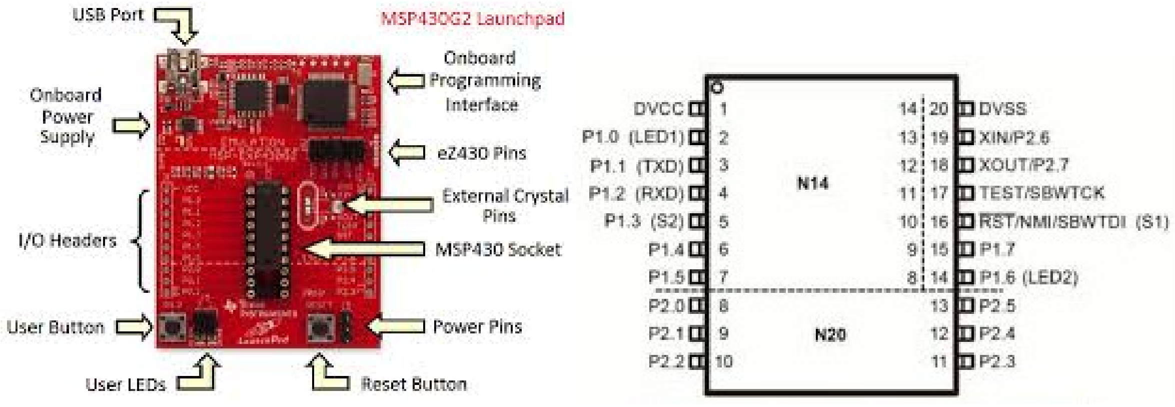


Figure 1: MSP430 Architecture

- 10-bit ADC
- Supports SPI, I2C, UART and USB interfaces
- Has 16 GPIO pins divided as two ports (8-bit each)

4 Interfacing Details

Three display devices - LEDs, 7-segment displays and an LCD display were interfaced. The details for each of them are as follows -



(a) Board

(b) Pin Diagram

Figure 2: Board details

4.1 LEDs

To do this, first the on-board LEDs (LED1 and LED2) were made to blink using the given example program. LED1 is connected to Port 1 BIT0 and LED2 is connected to Port 2 BIT6.

To blink an external LED, we connected an LED through a 2.2k resistor to Port 1 BIT4.

The delays were controlled using do...while loops

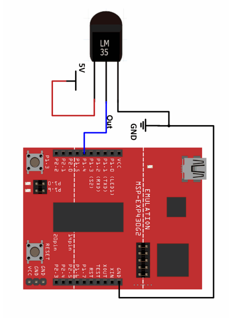


Figure 3: LED interface

4.2 7-Segment Display

To interface the 7-segment display, an array of the 7-segment 8 bit patterns were created and stored. A loop is run through these values and a delay is provided using do...while loops to be able to observe the transition. All 8 pins of Port 1 were used to interface the 7-segment display (including the decimal point)
Next, a two digit counter was made to count from 0 to 99. Both the 7-segment displays were connected to the same data lines (i.e Port 1) and to switch between the two displays the COM ports of the displays were controlled using a signal. This signal was generated using multiplexing between two bits on Port 2. By controlling the amount of delay, we were able to generate the illusion of a two digit counter.

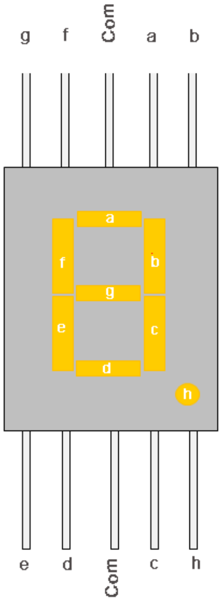
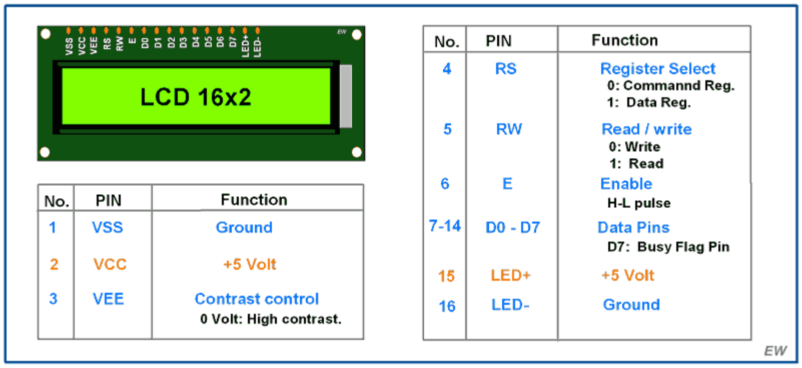


Figure 4: 7-Segment Display

4.3 LCD

The LCD was interfaced in the 4-bit 2-line configuration. The LCD was connected to the MSP430 as shown. 4 pins from Port 1 (BIT0 to BIT3) were connected to the LCD data pins D7-D4. The EN and RS pins of the LCD were connected to BIT4 and BIT5 of Port 1 respectively. Since we only want to write to the LCD, the RW pin of the LCD was grounded. To control the contrast, the contrast pin of the LCD was connected to a potential divider created using 4 2.2 kΩ resistors in a 3:1 configuration. The VCC and VSS pins of the LCD were connected to the 5V supply of the MSP430 (T1) and ground respectively. These help control the intensity of the back light.



No.	PIN	Function
1	VSS	Ground
2	VCC	+5 Volt
3	VEE	Contrast control 0 Volt: High contrast.

No.	PIN	Function
4	RS	Register Select 0: Commannd Reg. 1: Data Reg.
5	RW	Read / write 0: Write 1: Read
6	E	Enable H-L pulse
7-14	D0 - D7	Data Pins D7: Busy Flag Pin
15	LED+	+5 Volt
16	LED-	Ground

Figure 5: LCD PIN diagram

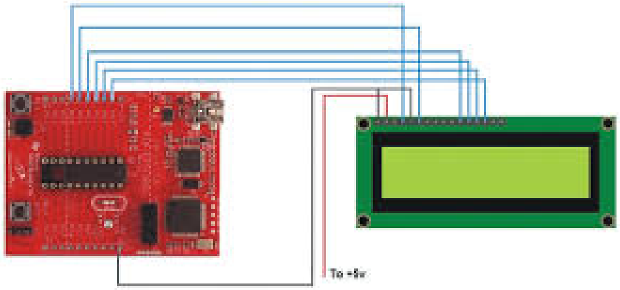


Figure 6: LCD interface

In order to print a message that is in infinite scroll on the LCD screen, a string with the message was used. Each chunk of 16 characters was sent to the LCD after a specific delay which was controlled using the delaycycles() command. At each iteration the first character of the 16 character chunk was removed and added to the end of the string.