EE 337: SPI and ADC Interface to 8051
Lab 7

Objectives

After completing this lab, you will be able to:

- write programs in embedded C for Pt-51 board.
- use serial peripheral interface (SPI) protocol for interfacing an ADC (TLV1543) with Pt-51.
- use ADC (TLV1543) for voltage measurement and distance measurement using a Hall effect sensor (DRV5053).

Background

SPI (Serial Peripheral Interface) is a synchronous, serial communication protocol used for communicating between microcontrollers and peripherals (or between 2 or more microcontrollers). In SPI communication, there is a `master` device (typically the μC), and a `slave` device (typically a peripheral or other μC). It uses shift registers for converting parallel data to serial and four lines for communication. They are two data lines *viz* MOSI (Master Out Slave In), MISO (Master In Slave Out), a CLK, and a Select line to choose the device you wish to communicate to. In this experiment, once the ADC is successfully interfaced to the microcontroller, it will be used to measure the distance using Hall effect sensor.

Note that the analog voltage output of the Hall effect sensor has to be amplified before being given to the micro controller via SPI. In this experiment, we are going to send data from the ADC (slave) to the 8051 micro controller (`master`).

The data in this case is the analog voltage at one of the input channels of ADC. It could be the analog voltage from a potentiometer for voltage measurement or output voltage (proportional to distance) from the Hall effect sensor.

We would like to thank Texas Instruments for providing the Hall Effect sensors (DRV5053) and ADC (TLV1543) for this experiment!

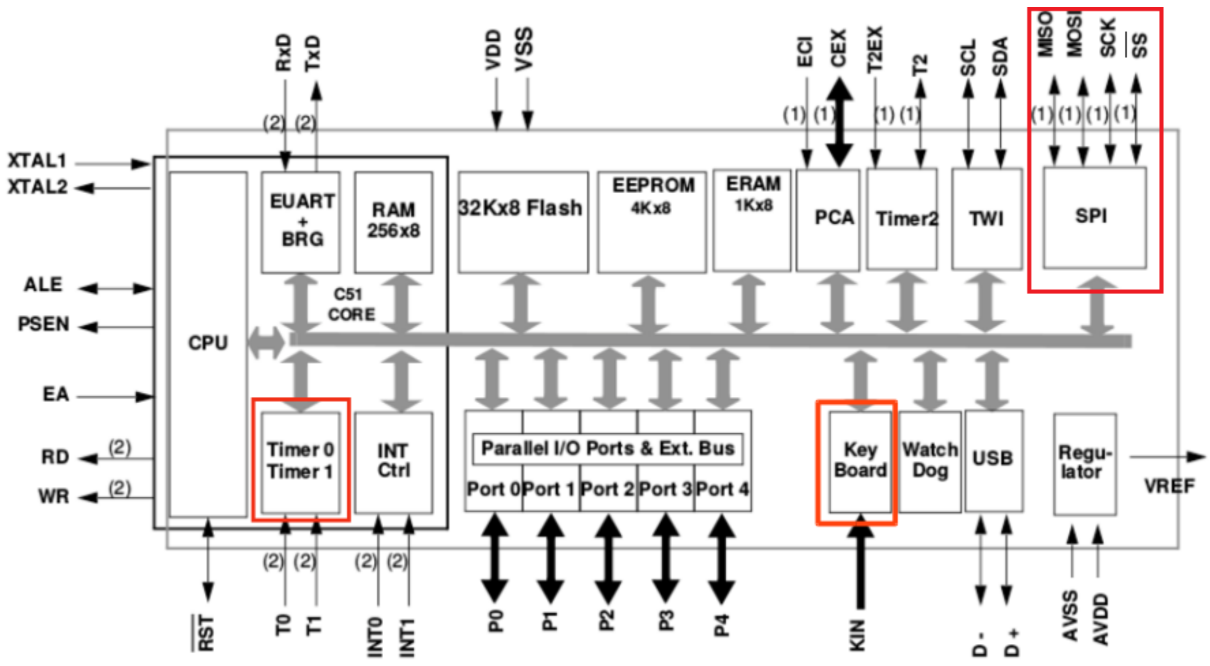

Figure 1: Block diagram

Homework

1. Go through and understand the reference document on *Embedded C programming using 8051 for Keil* uploaded on moodle. Go through the *programming style sheet* for writing better code.

2. Until now you have used assembly language for the LCD subroutines. In this lab use the *lcd.c* file which is written in embedded C.

3. For familiarization and testing of the LCD code, write C code to display “Hello” on the first line of the LCD screen starting from the first position and ”World” on the second line of the LCD starting from mid-position.

4. Read the datasheets of the ADC (TLV1543) and Hall effect sensor (DRV5053).

5. Read through the document *SPI-Intro* for an introduction to SPI. Refer to *spi.c* file. To make your life easy, the micro-controller is already configured for serial communication with the ADC IC (TLV1543) in the code written in *spi.c*.

6. Understand the configuration for doing the same. The code needs to be modified for Labwork part in order to receive data periodically from the serial interface using timers. Read through *ADC Interfacing* for understanding the same.

Lab Work

1. Use the code written in *spi.c* to for this part.
   Connect a potentiometer between 3.3V supply and ground terminals (available on-board). Vary the voltage from 0 to 3.3 volts and display the measured voltage on the LCD in the format: Voltage: xxxx mV
   Use the *delay_ms* subroutine to provide a delay of 100 msecs between every measurement. Verify that the displayed voltage is correct using a DMM or DSO.

   You are to perform these tasks.

   (a) Interface the ADC TLV1543 to the microcontroller in the Pt-51 board

   (b) Modify the Embedded C code written in the file *spi.c* by referring and understanding the *ADCInterfacing.pdf* and datasheet of the TLV1543.

   (c) Set up one of the hardware timers (Timers 0,1 or 2) to raise an event every 100 ms.

   (d) Measure the voltage output from the potentiometer using the ADC every 100 ms using the *delay_ms* sub-routine.

   (e) Display the voltage on the LCD.

2. Once the system is tested to measure variable input voltage, it can be set up for distance measurement.

3. The magnetic field for the Hall effect sensor is set up using a magnet. And the distance between the magnet and the sensor is to be measured and displayed on LCD.

4. Connect the amplifier circuit for measurement of distance using Hall effect sensor circuit shown in Fig. 2.

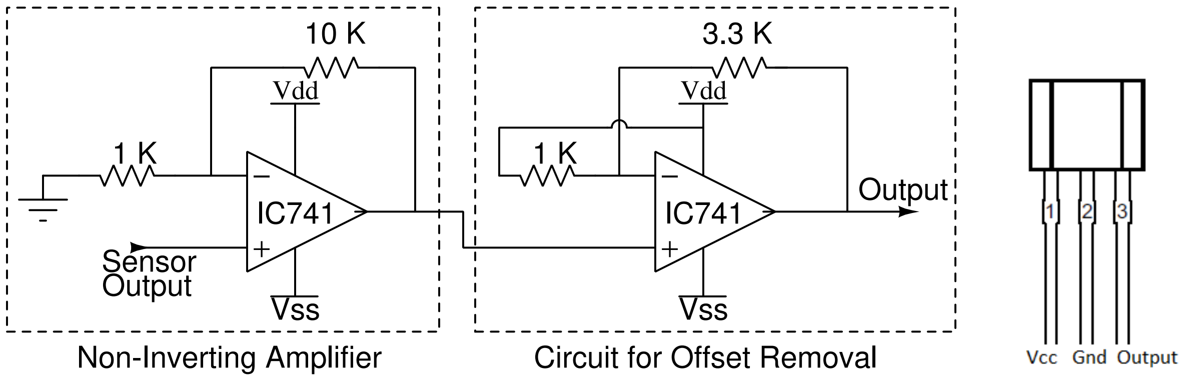

Figure 2: Amplifier circuit for measurement of distance using Hall effect sensor.

5. Test the circuit by placing the magnet at a distance of around 15 cm and taking readings of output voltage while bringing the magnet closer to sensor in small steps. For this part, you don't need the pt-51 board.

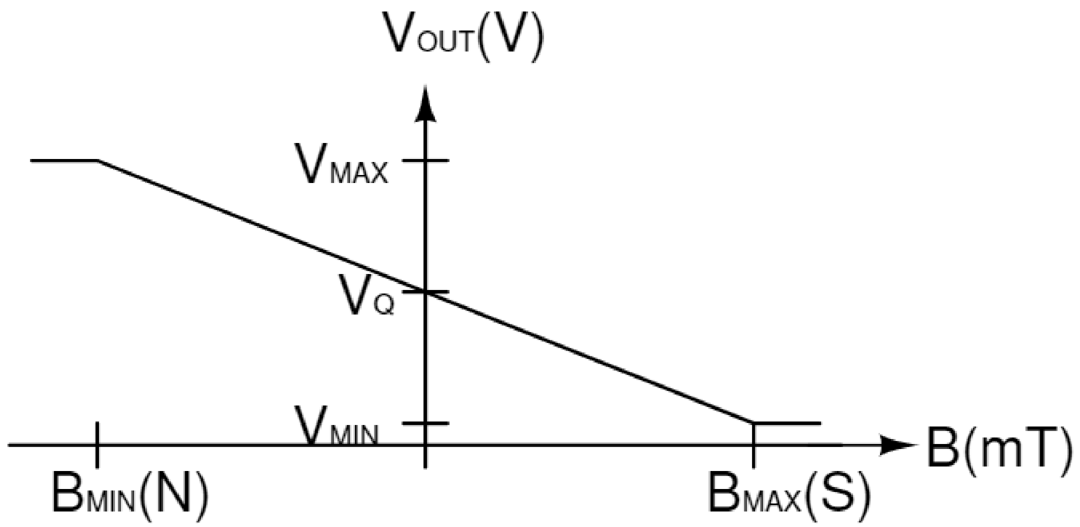

Figure 3: Characteristics of the hall effect sensor

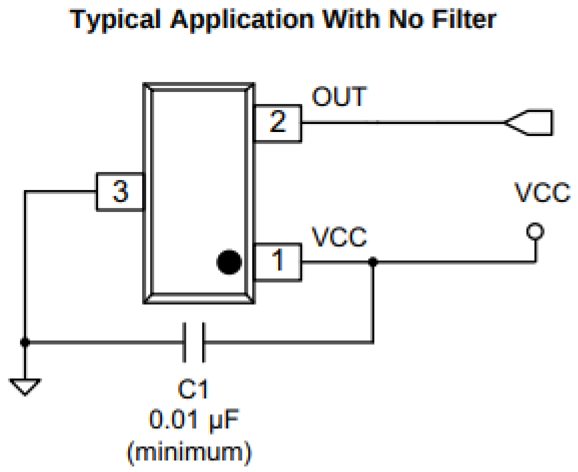

Figure 4: Typical application of the hall effect sensor

6. Now connect output of the sensor amplifier to the input of ADC. Measure the voltage output from the sensor for different positions of the magnet and display the distance on the LCD screen. The distance is to be calculated every 100 ms using a hardware timer (interrupt based) and updated on the LCD screen, so that a consistent and fast updating display is provided to the user. The program should read 10 ADC samples every 100 ms and average these 10 samples to compensate for the fluctuations in the output voltage. The distance measured is then to be displayed on the LCD Screen in the format,

```
Distance: xxxx cm
```

Useful functions in `lcd.c` and `spi.c` files:

- `SPI_Init( )` : Initializes the SPI Module and configures it to interface with the ADC.
- `LCD_Init( )` : Initializes the LCD screen using the same commands used in the assembly language programs written earlier, clears the LCD and sets the cursor position to Line 1 Position 1.
- `LCD_DataWrite(char dat)` : Writes a character on the LCD screen. e.g.,

  ```
  LCD_DataWrite(0x38);
  ```

- `LCD_CmdWrite(char cmd)` : Writes a command to the LCD. e.g.,

  ```
  LCD_CmdWrite(0xC6);
  ```

- `LCD_WriteString(char * str, unsigned char len)` : Writes a string on the LCD Screen. e.g.,

  ```
    LCD_WriteString(Hello, 5);
    char str[10];
    LCD_WriteString(str[3], 4);
  ```

  Please note that while displaying a string, the number of characters should match the number of characters in the string. Otherwise incorrect values will be displayed.

- `LCD_Ready()` : Checks if the LCD is ready to receive commands
- `sdelay(int delay)` : Produces a small delay (15 μs for a 24MHz clock)