EE337: Implementing SPI for an ADC

B Yaswanth Ram Kumar - 23B1277

April 02, 2025

Introduction

The objective of this laboratory exercise is to implement and test serial communication between an 8051 microcontroller and the MCP3008 analog-to-digital converter (ADC) using the Serial Peripheral Interface (SPI) protocol. The MCP3008 is utilized to convert analog signals into digital values, enabling the microcontroller to read and process sensor outputs. In this lab, a potential divider circuit is initially used to validate the ADC's functionality by measuring varying voltage levels, which are then displayed on an LCD. Following successful verification, the system is extended to measure and display ambient temperature using an LM35 temperature sensor. This lab reinforces fundamental concepts of SPI communication, ADC interfacing, and sensor-based data acquisition on embedded systems.

SPCON Register Configuration

From the MCP3008 datasheet `https://ww1.microchip.com/downloads/en/DeviceDoc/doc4337.pdf`, the SPCON register is configured with the hexadecimal value `0x3F`, based on the following bit settings:

Bit	Name	Description	Value
7	SPR2	Reserved (must be 0)	0
6	SPEN	0 = SPI initially disabled (enabled later)	0
5	SSDIS	1 = Disable Slave Select (SSBAR), free P1.1	1
4	MSTR	1 = Set MCU as SPI Master	1
3	CPOL	1 = SCK idle high	1
2	CPHA	1 = Data sampled on the second edge (trailing)	1
1–0	SPR1:0	11 = SPI Clock = $F_{clk}/16$ (750 kHz @ 12 MHz)	11

Binary Representation: `0b00111111`
Hexadecimal Value: `0x3F`

Establishing Communication between ADC and PT51 Board with `SPI.h`

```
void spi_init(void)
{

    SPCON = 0x3f;  // Configure SPI Control Register 0x3F, Mode 3 SPI, idle state is HIGH and data
        sampled at rising edge.

    IEN1 |= 0x04;  // Enable SPI interrupt (ESPI = 1)

    EA = 1;  // Enable global interrupts

    SPCON |= 0x40;  // Enable SPI module (SPEN = 1)
}
```

- Note that the code for the temperature sensing part in the `main.C` has been added directly to the only pet voltage sensing code. Separate files have not been created.

Configuring `main.C` for receiving data from both Potentiometer voltage and Temperature Sensor LM35

```
void main(void)
{
        int j=0;
        unsigned int adc_data=0;

        spi_init();

        adc_init();
    lcd_init();



        while(1)
        {
                unsigned int x,y,temp_data, pot_data;
                unsigned char str_data[6];

                x = adc(4);  // Read analog value from 4th channel of ADC MCP3008
                pot_data = (unsigned int) (x * 3.2258);  // Convert to millivolts

                y = adc(3);  // Read analog value from 3rd channel of ADC MCP3008
                temp_data = (unsigned int) (y * 3.2258) / 10;  // Convert to temperature ( C )

                // Display "Volt.: " on the first line of LCD
                lcd_cmd(0x80);  // Move cursor to first line
                lcd_write_string(display_msg1);

                // Convert pot_data to string (5-digit format) and print it
                int_to_string(pot_data, str_data);
                lcd_write_string(str_data);
                lcd_write_string(display_msg2);  // " mV"

                // Move cursor to the second line
                lcd_cmd(0xC0);  // Move cursor to second line
                lcd_write_string("Temp.:␣");

                // Convert temp_data to string and print it
                int_to_string(temp_data, str_data);
                lcd_write_string(str_data);
                lcd_write_string("␣oC");

        }

}
```

Thank You

23B1277