ECE 3720

SPI (Serial Peripheral Interface)

Section 2

Brandon Wider

Date Performed: March / 26 / 2020

Lab 9

ABSTRACT:

This lab discusses how to use the SPI and shift registers to run the 8-bit shift register with LEDs.

INTRODUCTION:

The purpose of this lab was to use and understand the SPI and show how the signals is produced by it can control an external peripheral using an interrupt to increment an array of numbers. For this lab it'll be lighting up the LED down the line from LED 0 - 7 according to the binary value of the program's array.

EXPERIMENTAL PROCEDURES:

1. Configure the SPI register
2. Set data direction with TRISx
3. Disable the interrupt
4. Set edge polarity
5. Create a while loop to continuously run the code
6. Create an __ISR function to write the spiChar[] to SPI1BUF
7. Use the PPS to select SS1 and SD01
8. Enable interrupts to display the values through the LEDs
9. Setup and wire up the circuit diagram

RESULTS and DISCUSSION:

If done in the lab, it should light up the LEDs based on the increment of the numbers in the array using the interrupt by going from array position 0 to the end of the array. From there, the LEDs will light up as the binary equivalent of the decimal value in the array. A delay function was used to help ensure the LEDs ran correctly.

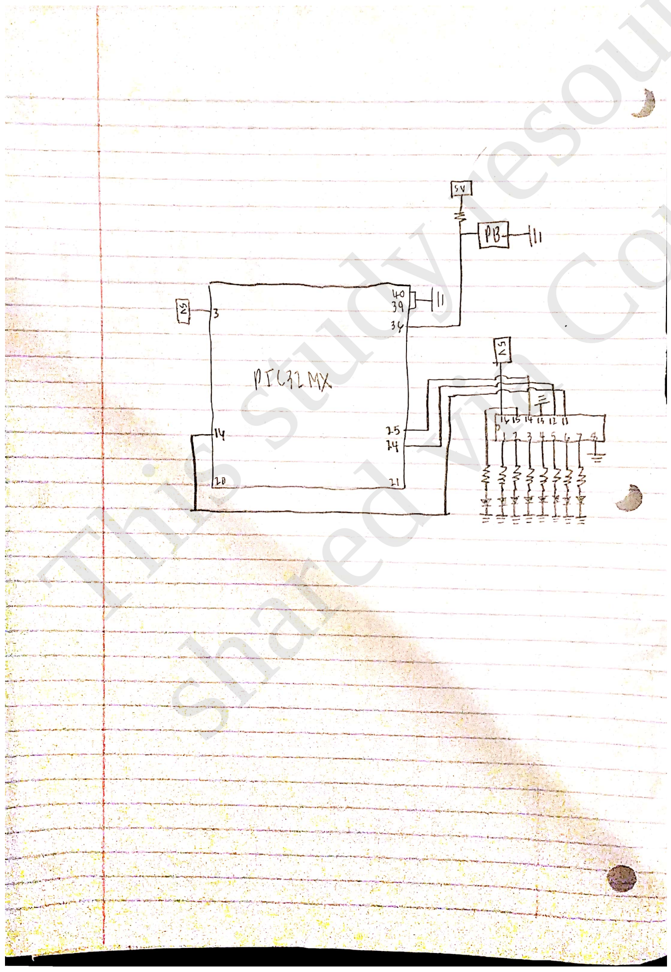
CONCLUSSION:

The purpose of this lab was to implement and understand SPI. This is done with SN74HC959, which is used to shift the written bits from the SPI buffer to the corresponding LED output. The circuit itself receives different signals from the PIC32, with the SDO (serial data out) and write out each of the 8 bits to the chip. This component for the lab could be used for general purpose of different electronic devices that works with different sequential and or repeat a cycle like radio stations.

REFERENCES:

Clemson University’s ECE 372 Lab 9 Powerpoint.

FIGURES AND TABLES:



CODE:

```
#include<plib.h>

int i = 0;

char spiChars[18] = {0, 1, 2, 4, 8, 16, 32, 64, 128, 255,
                     254, 253, 251, 247, 239, 223, 191, 127};

// Necessary for the LEDs to light up properly in the right order
delay(){
    int i,j;
    for(i=0;i<250;i++)
        for(j=0;j<250;j++);
}

//InterruptFunction
void __ISR(3) other(void)
{
    if(i == 18)
    {
        i = 0;
    }
    SPI1BUF = spiChars[i];
    i++;
    while(SPI1STATbits.SPIBUSY);
    delay();
    IFS0bits.INT0IF = 0;
}

main()
{
    INTEnableSystemMultiVectoredInt();

    ANSELB = 0;
    ANSELC = 0;
    TRISB = 0;
    TRISC = 0;
    TRISBbits.TRISB7 = 1;
```

```
        //Setup Interrupt
    INTCONbits.INT0EP = 1;
    IPC0bits.INT0IP = 1;
    IFS0bits.INT0IF = 0;
    IEC0bits.INT0IE = 1;

    PPSOutput(1,RPC0,SS1);
    PPSOutput(3,RPC1,SDO1);

    IEC1bits.SPI1EIE = 0;
    IEC1bits.SPI1RXIE = 0;
    IEC1bits.SPI1TXIE = 0;
    SPI1CONbits.ON = 0;

    int rData = SPI1BUF;

    SPI1CONbits.ENHBUF = 0;
    SPI1BRG = 5;
    SPI1STATbits.SPIROV = 0;
    SPI1CONbits.MSTEN = 1;
    SPI1CONbits.MSSEN = 1;
    SPI1CONbits.SSEN = 0;
    SPI1CONbits.MCLKSEL = 0;
    SPI1CON2bits.AUDEN = 0;
    SPI1CONbits.MODE32 = 0;
    SPI1CONbits.MODE16 = 0;
    SPI1CONbits.CKP = 0;
    SPI1CONbits.CKE = 1;
    SPI1CONbits.ON = 1;
    SPI1BUF = 0;

    while(1);
}
```