CALIFORNIA STATE POLYTECHNIC UNIVERSITY, POMONA
COLLEGE OF ENGINEERING

ECE 3301L Spring 2019 Session 2 Microcontroller Lab Felix Pinai

LAB2
Basic Input/Output using Microcontroller parallel ports

To perform the lab below, you need to download the spec of the processor that we are using in the lab. Go to the following link:

http://ww1.microchip.com/downloads/en/DeviceDoc/39689f.pdf

Download the datasheet and save somewhere (like Desktop) that you can easily have access to it.

MATERIALS:

Provided by instructor:

- 2 RGB LEDs

Student must get the following:

- (10) 220 or 330 ohms resistors
- (4) 1K or higher resistors
- (4) regular LEDs (any color)
- 1 bank of 4 minimum DIP Switches
-

PART 1)

Write a program that will input from a bank of 4 switches (DIP switches) and output each corresponding input to an equivalent LED. There would be a total of 4 LEDs. The inputs are connected from PORT B bits 0-3 while the outputs are from PORT A bits 0-3.

Hint:

a) Use the downloaded datasheet of the PIC18F4321, go to chapter 11 'I/O Ports. Study the definitions of the **TRISx** registers where 'x' represents the PORT name. Set the right value of each bit in the TRIS register to setup whether the corresponding pin is an input or output. A '1' will make the pin an input while a '0' makes it an output.

b) If you use PORTA or PORTB for the inputs and/or outputs, you need to study the definitions of the register **ADCON1** (see the chapter 20 '10-bit Analog-to-Digital Converter A/D Module' in the datasheet) and its PCFG3:PCFG0 bits. You need to make sure that the pins from ports A and B are setup in digital mode and not analog.

c) Write an endless loop to read the input switches and output the values to the LEDs. Based on the schematics shown on 'LAB2_Schematics.pdf', the four inputs should be port B bits 0 through 3 while the four LEDs should be connected to port A bits 0 through 3.

Example:

```
char in;                          // Use variable 'in' as char
        while (1)
        {
                in  = PORTB;       // read data from PORTB and save it
                                   // into 'in'
                in = in & 0x0F;    // Mask out the unused upper four bits
                PORTA = in;        // Output the data to PORTA
        }
```

PART 2)

Connect the provided Common-Cathode RGB LED at D1 to the pins D0 through D2. Pin D0 will be used to turn on the RED color of LED D1, D1 is for the GREEN and D2 is for the BLUE. Don't forget to add the TRIS command for this port D.

Write an endless loop where the lowest three pins of the DIP Switch SW1 (pins connected to port B bits 0 through 2) are read and then these three bits are outputs to the port D whereas the following color will be generated on the LED D1:

Inputs (SW1)			Outputs (D1)			
RB2	RB1	RB0	RD2	RD1	RD0	Color
0	0	0	0	0	0	No light
0	0	1	0	0	1	Red
0	1	0	0	1	0	Green
0	1	1	0	1	1	Yellow
1	0	0	1	0	0	Blue
1	0	1	1	0	1	Purple
1	1	0	1	1	0	Cyan
1	1	1	1	1	1	White

PART 3)

Write an endless loop that will continuously generate the list of colors shown on Part 2) with about 1 second delay between each color. There is no input to read.

Hint:

Write the subroutine Delay_One_Sec() below:

```
void Delay_One_Sec()
{
        for(int  I=0; I <xxxx;  I++);
 }
```

whereas you have to specify the value of 'xxxx' in order to make the desired delay of time. Once this delay routine is written, use it to generate the time delay. You would need an overall endless loop that will output the 8 different colors spaced by 1 second delay.

PART 4)

Based on the code on Part 3), add the Common-Anode RGB D2 to the PORT D bits 3 through 5.

Rewrite the code such that the following sequence is generated:

Color @RGB LED D1	Color @RGB LED D2
No light (off)	White
Red	Cyan
Green	Purple
Blue	Yellow
Yellow	Blue
Purple	Green
Cyan	Red
White	No light (off)

PART 5)

This part will have a different sequence of colors for the LEDs D1 and D2.

Color @RGB LED D1	Color @RGB LED D2
White	Blue
Red	White
Green	Purple
Blue	No light (off)
Yellow	Red
Purple	Cyan
Cyan	Green
No light (off)	Yellow

PART 6)

Use any DIP switch connected to PORT B to select what color sequence to display. If the logic state of the switch is ‘0’, display the sequence on Part 4). Else, display the sequence on Part 5).