17.383 Microprocessors A
Lab 2
Fall 2010

PIC16F684 Digital Input/Output (I/O)

Objectives

1) To use MPLAB IDE software and PICC Compiler to demonstrate the following:

    a. Digital Input
    b. Debouncing switches via code development
    c. Digital Input and Output control
    d. Numbering Systems

Materials

- 1 Computer (PC) with an available USB Port
- MPLAB IDE and HI-Tech PICC software installed on PC
- 1 PICkit ™ 1 with USB cable
- 1 PIC16F684
- 1 Oscilloscope

WARNINGS AND PRECAUTIONS

1) Never remove the PIC16F684 from an energized circuit

2) Do not construct circuits while energized

3) Follow electrical safety precautions

Source File Locations

1) HI-TECH C PRO Toolsuite

    □ HI-TECH C Compiler for PIC10/12/16 MCUs

    C:\Program Files\HI-TECH Software\PICC\9.70\bin\picc.exe

2) Microchip MPLASM Toolsuite

- □ MPASM Assembler (mpasmwin.exe)

  C:\Program Files\Microchip\MPASM Suite\mpasmwin.exe

- □ MPLINK Linker (mplink.exe)

  C:\Program Files\Microchip\MPASM Suite\mplink.exe

- □ MPLIB Librarian (mplib.exe)

  C:\Program Files\Microchip\MPASM Suite\mplib.exe

Background Information

The PICkit 1 Starter Kit has D0 through D7 LEDs connected to the PIC16F684 microcontroller via PORTA pins RA1, RA2, RA4, and RA5. In addition, the starter kit has a switch (SW1) connected to PORTA pin RA3 which will be utilized for microcontroller interfacing experimenting. Figures 1 and 2 of this lab show the schematics for these connections. Utilizing Microchip's MPLAB and HI-TECH C Compiler, one can access PORTA to read and write to the pins, thus enabling the use of these digital connections.

The switch is connected across a +5 volt supply to the data input pin of the microcontroller via a resister. An issue arises with mechanical switches, in that they do not cleanly open and close. They can produce spikes which can be interpreted as false switch actions. There are a variety of methods to eliminate switch bouncing. One method is the addition of a capacitor across the switch. Another method is a software approach. In this lab, we will take the software approach by writing code which can be incorporated into your application which will eliminate switch bouncing.

We will introduce some methods of representing information within microcontroller systems. Decimal, hexadecimal, and binary numbering systems will be outlined. Programming and data information will use the various numbering systems.

Minimal code will be provided to the student during this lab. Utilizing the fundamentals presented thus far, the student can write code to perform the requested routines. This will reinforce the concepts learned to date.

17.383 Microprocessors A
Lab 2
Fall 2010

Pre-Lab Preparation

1) Develop the required code as outlined in the procedural steps.

Procedure

Experiment 1. DIGITAL SWITCH INTERFACE (Input to the microcontroller)

a. Review the schematic in figure 1 and determine how the PICKIT1 Starter kit pushbutton switch (SW1) is connected to the PIC16F684.

b. From the schematic determine what voltage will be seen when the switch is off.

c. From the schematic determine what voltage will be seen when the switch is on.

d. Utilizing the information acquired from Experiment 1 a thru c above, create code that will perform as follows:

1. When the switch is off, all of the LEDs will be off.

2. When the switch is on, D5 LED will be on. Schematic for the LEDs is shown in figure 2.

3. When the switch is release, D5 LED will go out.

4. The code will contain the following:

    a. Write code in C. Use the MPLAB/PICC software to produce the code files.
    b. Include header information as shown in figure 3
    c. Include the pic.h header file
    d. Include the proper configuration word at the proper location
    e. Initialize PORTA to zero
    f. Turn off the comparators
    g. Turn off the ADC

e. Save the program as Lab_2A.c.

f. Burn the code into your PIC16F684 microcontroller and verify that your hardware is functioning as planned.

g. If required, when the software and hardware are operating properly, notify the instructor so it can be observed. Annotate observance in your Lab Notebook.

h. When the hardware is functioning as desired, printout a copy of Lab_2A.c. Submit the printout or paste a copy in your Lab Report.

Experiment 2. DEBOUNCING SWITCHES

a. Due to the bouncing action as a result of activating a mechanical switch, the LED turned on in Experiment 1 may in fact have turned on roughly 50 times per second, rather than once as we originally thought. In order to eliminate these false switch actions, code can be incorporated into your programs which allows time for the switch to settle out by means of adding a 20 msec delay. An example of such code is shown in figure 4.

b. Create a new file as follows:

1. Write the code in C. Use the MPLAB and PICC software to produce the code files.
2. Include header information as shown in figure 3.
3. Add the debouncing code contained in figure 4 to your code developed in Experiment 1.

c. Review the debouncing code (figure 4) and determine how it is functioning.

d. In the lab write-up, indicate the results of your review of the code you accomplished in the prior step.

e. Save the program as Lab_2B.c.

f. Burn the code into your PIC16F684 microcontroller and verify that your hardware is functioning as planned.

g. Use an oscilloscope to watch the transition when a switch is pushed/released. Record what you observe. Is there any reason for concern?

h. If required, when the software and hardware are operating properly, notify the instructor so it can be observed. Annotate observance in your Lab Notebook.

i. When the hardware is functioning as desired, printout a copy of Lab_2B.c. Submit the printout with your report.

Experiment 3. DIGITAL SWITCH INTERFACE – Controlling LEDs

a. Modify the code developed in Experiment 1 to incorporate the required features:

1. Initial power on/programming, all LEDs are off.

2. When the switch is depressed for the first time, LED D0 will come on.

3. The LED turned on in Experiment 3 a.2 will stay energized until the switch is depressed a second time. At that time, LED D0 will go out and LED D1 will come on.

4. The process above will continue until LED D7 is turned on. The next time the switch is depressed, all LEDs will be off and the sequence will start all over.

   SUGGESTIONS: Use the ***switch*** or the ***if*** – ***if else*** statements.

5. At this time, *do not incorporate* the switch debouncing code into your file. Do you think that the circuit will operate as desired when the debouncing code is not included in your code?

b. Save the program as Lab_2C.c.

c. Burn the code into your PIC16F684 microcontroller and verify that your hardware is functioning as planned.

d. If required, when the software and hardware are operating properly, notify the instructor so it can be observed. Annotate observance in your Lab Notebook.

e. When the hardware is functioning as desired, printout a copy of Lab_2C.c. Submit the printout with your report.

f. Repeat Experiment 3 a thru e and now add the switch debouncing code into your file. Your new filename will be Lab_2D.c. Does this enhance the operation of the circuit?

g. When the hardware is functioning as desired, printout a copy of Lab_2D.c. Submit the printout with your report.

Experiment 4. NUMBERING SYSTEMS (If there is Time Permitting)

a. When producing code, we have, for the most part, been entering numbers using the decimal format. In some cases it may be desirable to develop your code using other numbering systems (binary, hexadecimal). The format for the various numbering systems are as follows:

System	Format	Example
Decimal	##	0
Hexadecimal	0x##	0x3F
Binary	0b########	0b00111111

b. Rewrite the code developed in Experiment 3 utilizing hexadecimal numbers in place of the decimal values.

c. Save the program as Lab_2E.c.

d. Burn the code into your PIC16F684 microcontroller and verify that your hardware is functioning as planned.

e. If required, when the software and hardware are operating properly, notify the instructor so it can be observed. Annotate observance in your Lab Notebook.

f. When the hardware is functioning as desired, printout a copy of Lab_2E.c. Submit the printout with your report.

g. Repeat Experiment 4 b thru f using binary numbers. When saving and printing your file, use the file name of “Lab_2F.c”.

17.383 Microprocessors A
Lab 2
Fall 2010

Summary:

We have covered the following topics during this lab:

Digital Input interfacing with the microcontroller
Debouncing switches using code within your file
Controlling LEDs and switches (Digital Input and Output control) for desired results
Explored the use of a variety of numbering systems when writing code

Many applications will require the control of the digital microcontroller ports. By understanding how to control these ports, any digital device can be connected and controlled by the microcontroller.

Question:

1. None

Figure 1 - PICkit 1 Starter Kit - Switch (SW1) Schematic

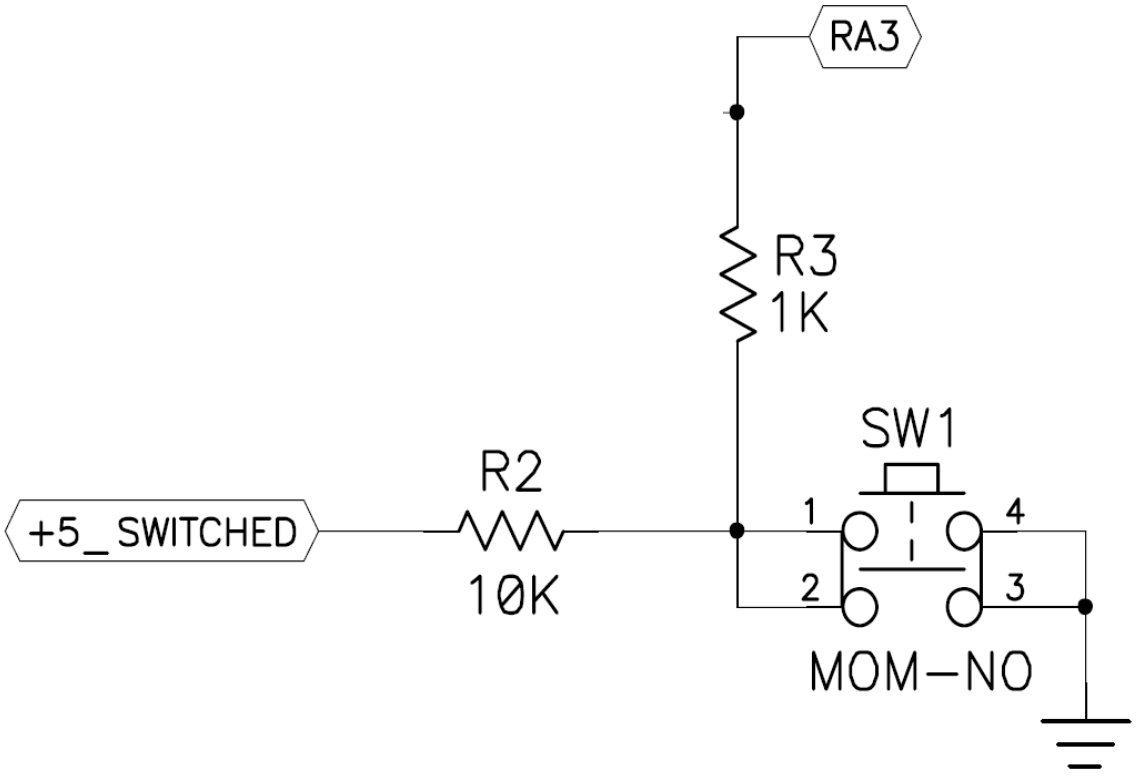

Figure 2 - PICkit 1 Starter Kit LED Layout

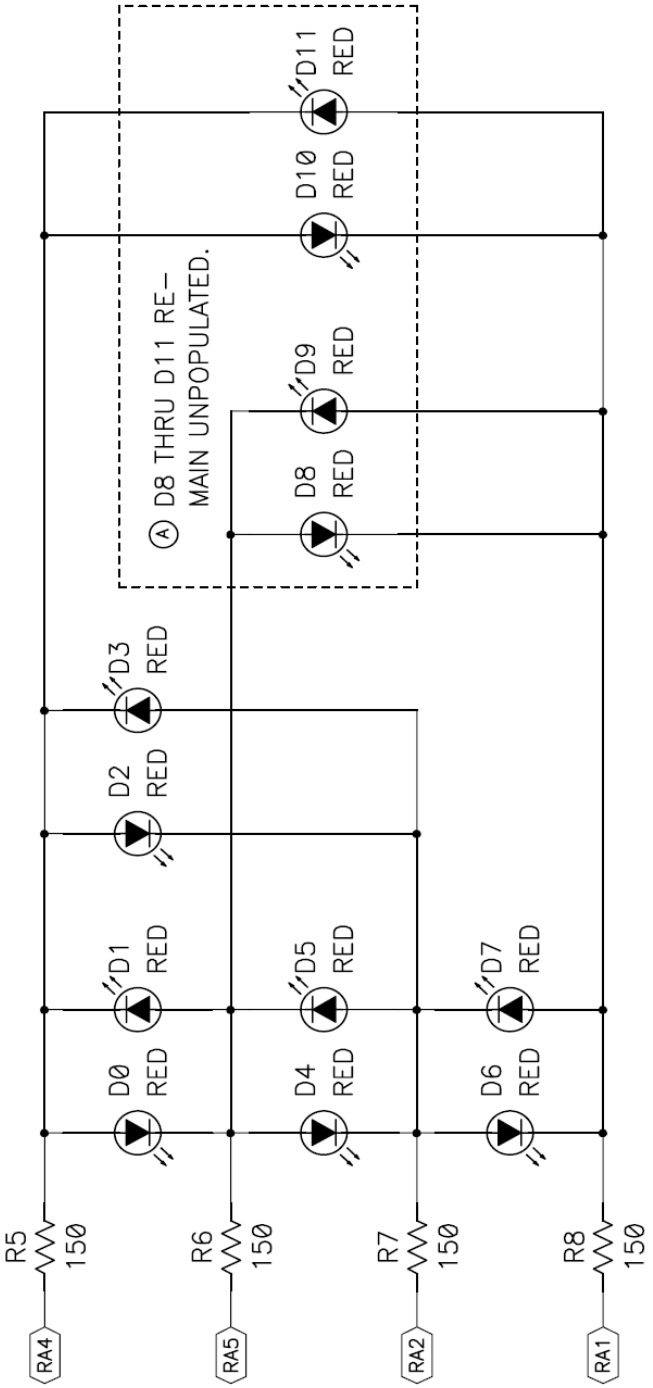

Figure 3 - Required Header Code

```
/*******************************************************
*
* Program Title:     xxxx
*
* Program File Name: Lab_XA.c
*
* Microprocessors A 17.383
*
* xxxxxxxx - Put in Semester (i.e. Fall 2010) here
*
* xxxxxxxx - Put in your name here
*
* xx/xx/xx - Put date here
*
*******************************************************/
```

Figure 4 - Switch Debouncing Code

```
/*******************************************************
* Function: delay_20ms
*
* Description:  Causes a delay in program execution
*
* Notes:
*
* Delay was determined through trial and error
*
* Returns:  None
*
*******************************************************/
void delay_20ms(void)
{
      const int D20ms = 1150;      //  Declare a Constant for 20 ms Delay
      int i;

            for (i = 0; i < D20ms; i++)

      return;

}
/******** END OF delay_20ms ************************/
```