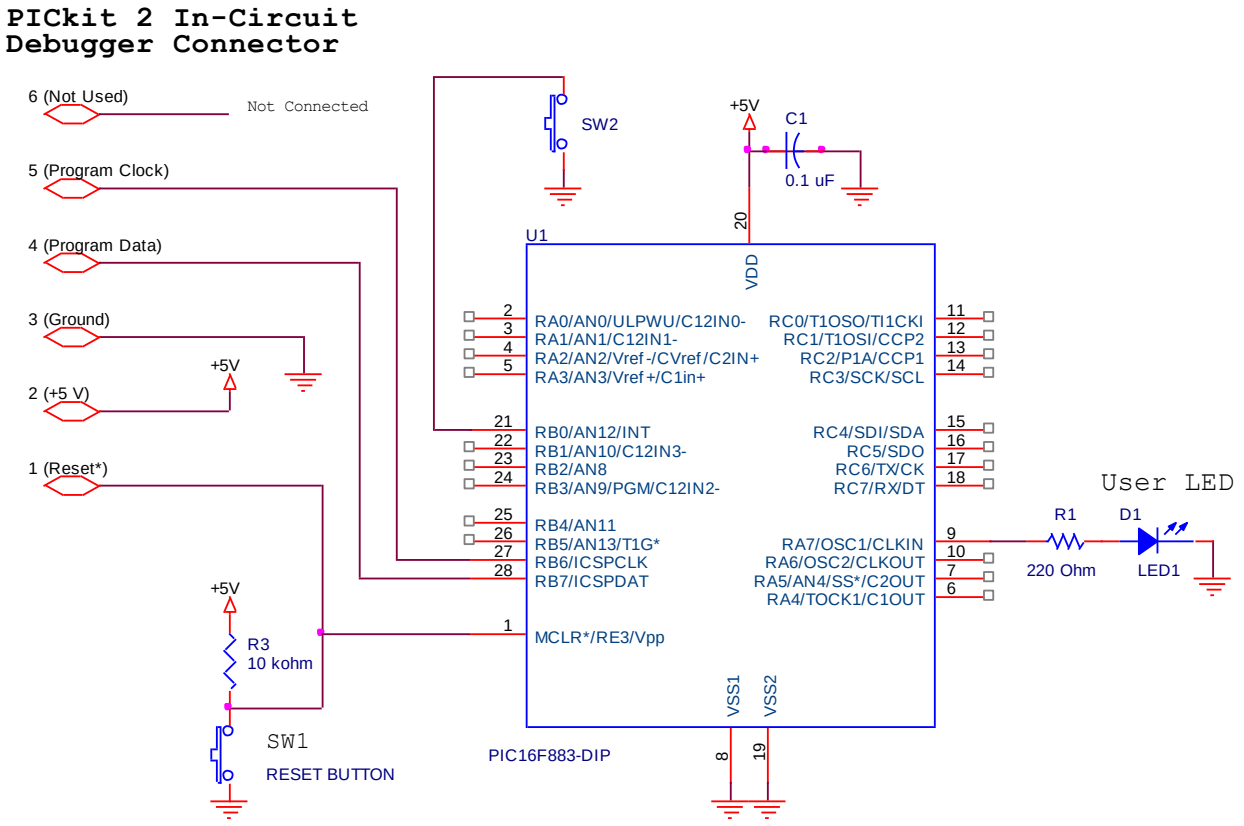
Lab 2 – Programming Test and I/O Pin Toggle Program

In this lab, you will create a simple assembly program to toggle an I/O pin on the PIC16F883. This lab will serve to verify that the programming circuit and program memory are functioning. It will also provide a simple assembly program which we will use again to demonstrate how a program can be loaded on the microcontroller through a test program. We will program the microcontroller and verify the program memory using a PICKit 3. The PICKit 3 may be used for programming a PIC microcontroller, or for performing in-circuit debugging.

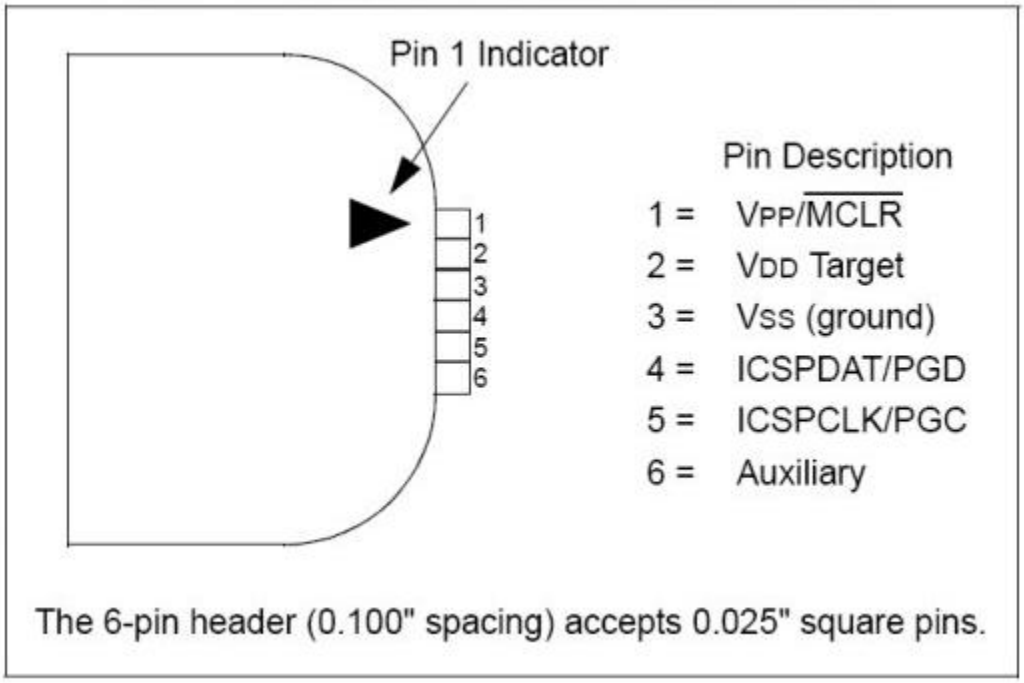
You will program a PIC microcontroller in the circuit shown in Figure 1. This circuit has a single pushbutton input, connected to pin RB0, and a single LED connected to pin RA7. The value output on pin RA7 should mirror the digital input on pin RB0. You will write an assembly program which constantly checks the state of the input RB0, and sets the output RA7 accordingly. Your program will need to perform the necessary initialization steps for the used ports. Port A should be set as an output using the special function register (SFR) TRISA. Port B should be set as a digital input, by setting the port as an input using the TRISB register and disabling analog inputs with the ANSEL and ANSELH registers. The pushbutton connected to RB0 does not have a pull-up resistor, as seen in Figure 1. On the PIC16F883, internal pull-up resistors are available on Port B. Enable the pull-ups using the active low control bit $\overline{RBPU}$ in the OPTION register. (Refer to data sheet for Port A & B and OPTION_REGISTER)

Figure 1. Lab 2 Project Circuit: PIC16F883 with a pushbutton switch and an LED.



The circuit will be made available on a breadboard in lab B104. You will need a PICkit 3 (or PICkit 2) programmer to load the program onto the pic flash memory. As shown in Figure 1, there is a 6-pin connector on the circuit board for the PICkit 3. A pinout for the PICkit programmer is shown in Figure 2. Notice that pin 1 is indicated by an arrow on the PICkit. This should be connected to the connector pin which is wired to pin 1 of the PIC16F883.

Figure 2. PICkit 3 pinout.



Setting up a "Project" in MPLAB X

1. You must have an active project open in order to build or assemble a program using the MPLAB integrated debugging environment (IDE). If you attempt to build a program with no active project, you will encounter errors.

2. To open a new project, click on **File** – **New Project** ... Choose 'Microchip Embedded' and 'Standalone Project' then press Next. Choose Family 'Mid-Range 8-bit MCUs' and 'PIC16F883' (you can just beginning typing PIC16… to speed up finding the chip). Under *Hardware Tools* select PICkit 3. Under *Compiler Toolchains* select MPASM. Choose a Project Location and provide a Project Name (e.g. ***toggleIOtest***). Leave 'Set as main project' checked and click Finish. A project window appears, which initially has empty source folders.

3. To add a source file to the project, right-click on the Source Files folder and choose '*New AssemblyFile.asm*'. Name the file and click Finish.

4. You ***must*** set the configuration bits inside the PIC16F883 before the program is simulated or programmed into the PIC's flash memory. Click on ***Window – PIC Memory Views – Configuration Bit***. Enter the following configuration bit options by clicking on the Setting field entry and then selecting the desired setting:

a.	Oscillator Selection	**INTOSCIO internal oscillator, I/O on RA6 and RA7**
b.	Watchdog Timer	**WDT disabled**
c.	Power-up Timer	**PWRT disabled**
d.	RE3/MCLR pin function	**RE3/MCLR pin function is MCLR**
e.	Code Protection Bit	**Program memory code protection is disabled**
f.	Data Code Protection bit	**Data memory code protection is disabled**
g.	Brown Out Reset Select	**BOR disabled**
h.	Internal/External Switchover	**Internal/External Switchover mode is disabled**
i.	Fail-Safe Clock Monitor	**Fail-Safe Clock Monitor is disabled**
j.	Low Voltage Program	**RB3 pin has digital I/O, HV on MCLR used to program**
k.	Brown Out Selection	**Brown-out Reset set to 2.1 V**
l.	Flash Program Memory Write	**Write protection off**

Note that the Watchdog Timer MUST be turned OFF for a program that does not periodically reset the Watchdog Timer, even if you are only simulating a program using "MPSIM". Otherwise, MPSIM will mysteriously reset itself when the watchdog timer times out.

Click the ***Generate Source Code to Output*** button. The generated code will appear in the *Config Bits Source* window. This is the source code that should be included in our source file to set the appropriate configuration settings on the PIC. Note that it includes the line *#include<xc.h> (or #include "pic16f883.inc")*. This statement includes the header file, which contains many useful labels (so that we can refer to registers and bits by label rather than by address). Highlight all of the code generated, and copy and paste this code into your source file, at the top.

5. You will need a line at the top of your assembly file, above the #include statement, to build for the proper hardware. Add this line to the top of your .asm file.

```
list      p=16F883
```

6. You need to indicate where within the program memory the first instruction in the assembly file should be located. Do this with the line:

```
RESET_VECTOR        code    0               ; Reset vector is at program location 0x00.
```

RESET_VECTOR is just a label here, any label could be used in its place.

7. The next instruction will be placed at program memory location 0x00. This should typically be a goto operation which jumps to the actual beginning point of your program.

Assignment

Due Wednesday, March 23, 2016

Complete the assembly program and build the program in MPLAB using **Run—Build Main Project**. Flash the program onto the circuit located in lab B104 using the PICkit3 that is located with the circuit. Run the program in debug mode, by selecting **Debug—Debug Main Project**, and verify that the LED toggles with the pressing of the push-button connected to RB0 (note that there is a 2nd push-button connected to pin 1, this is the reset button connected to MCLR). Go to **Window–PIC Memory Views—Program Memory** to verify that the program is written to the program memory as expected. Submit your assembly program and a printout from the Program Memory view (only need to include relevant instruction addresses) along with a brief cover page.