LAB2: PROGRAMMING PIC18F4550 MICROCONTROLLER WITH BUTTON

1. Objectives:

- Learn how to use an output pin by driving an LED

2. Introduction to switch/button

Push Buttons and Switches are digital inputs and are widely used in electronic projects as most systems need to respond to user commands or sensors. Reading a switch is very useful because a switch is widely used and can also represent a wide range of digital devices in real world like push buttons, limit sensors, level switches, proximity switches, keypads (a combination of switches) etc.

Connecting a switch to a microcontroller is straight forward, all we need is a pull-up or pull-down resistor. The pull-up or pull-down resistor is very important, if there is no resistor it will be difficult to determine the state of the pin, this is called floating.

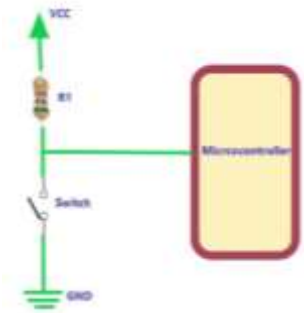


Figure 1: A switch with a Pull-up resistor

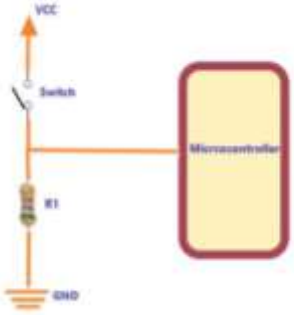


Figure 2: A switch with a Pull-down resistor

The microcontroller pin is configured as an input. If there is nothing connected to the pin and the microcontroller program reads the state of the pin, will it be high (pulled to VCC) or low (pulled to ground)? It is difficult to tell. But with a resistor connected to VCC (pull-up) as in figure 1, or connected to ground (pull-down) as in figure 2, will ensure that the pin is in either a high or low state.

Pull-up resistors are the more common so we will focus on them. In figure 1, if the switch is open, the input of the PIC will be high (+5V) and when the switch is closed, the input of the PIC will be low. If there was no resistor, then it could have been a short circuit. Internal pull-up resistors can also be enabled in software if external resistors are not going to be used, refer to the datasheet to find out more.

The larger the resistance of this pull-up resistor, the slower the pin is to respond to voltage changes, this is because the system that feeds the input pin is essentially a capacitor coupled with the pull-up resistor, thus this forms an RC filter, and RC filters take some time to charge and discharge. So if you have a very fast changing signal (like USB), a high value pull-up resistor can limit the speed at which the pin can reliably change state. And on the other hand if you select a lower resistance, when the switch is closed, more

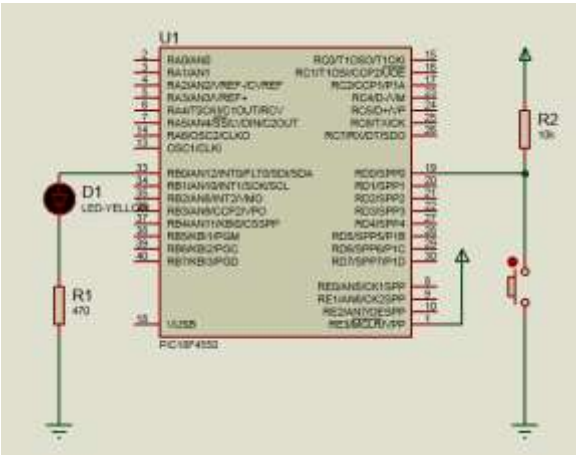
current will be routed to ground which is not a good idea especially if the circuit is battery powered. Generally a value of 4.7KΩ, 10KΩ could do the job.

The **TRISx** register control the direction of the PORT. If the value if the **TRISx register is "1"**, then the PORT/PIN becomes input, any input device like a switch can be read then. When the value if the **TRISx register is "0"**, then the PORT/PIN becomes output, any output device like a diode can be controlled then.

Example:

- Set PORTB (all bits) to input: TRISB = 0xFF
- Set PORTD bit 0 to input: TRISD0 = 1
- Set PORTB bits 0 to 3 to input and bits 4 to 7 to output: TRISB= 0x0F
- To remove the effect of switch bouncing, we can read the switch twice with a short delay in between to make sure the switch has completely changed its status.

Let's try to simulate this circuit below:




```
int main()
{
  TRISB0 = 0; //RB0 as Output PIN
  TRISD0 = 1; //RD0 as Input PIN

  RB0 = 0; //LED Off

  while(1)
  {
    if(RD0 == 0) //If Switch Pressed
    {
      RB0 = 1; //LED ON
      __delay_ms(3000); //3 Second Delay
      RB0 = 0; //LED OFF
    }
  }
  return 0;
}
```

3. **Lab exercise**

a. **Lab 2a.**

- Connect the circuit using Proteus with PIC18F4550, a LED on PORTD and 2 buttons on Port B.
- Write a program using MPLABX to control one LED with two button: firstly, the led is set off. When button 1 is pressed, the light is set on and when button 2 is pressed, the light is set off. This process repeats.
- Simulate the circuit using Proteus ISIS program.

b. **Lab 2b.** Connect the circuit using Proteus with PIC18F4550 with 8 LEDs on PORTD and one button on Port B. Write an C program to glow LEDs in a sequential manner when the push button is pressed and set LEDs off in reverse sequence when the push button is pressed again. Use MPLABX to write our code and Proteus to show the results.

Name: Student Code: Class: Lab:

1. Circuit

2. Algorithm flowchart
3. Code and explanation
4. Summary