ENGINE

Erasmus+

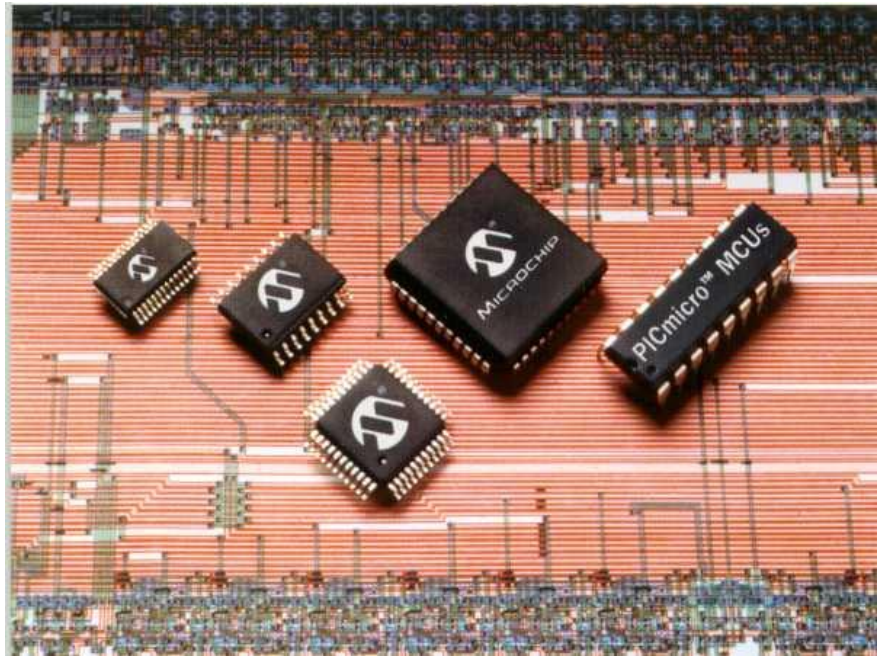
TEACHING ONLINE ELECTRONICS, MICROCONTROLLERS AND PROGRAMMING IN HIGHER EDUCATION

Module_2-7. Push Button

PIC18F4550 with Proteus Simulation

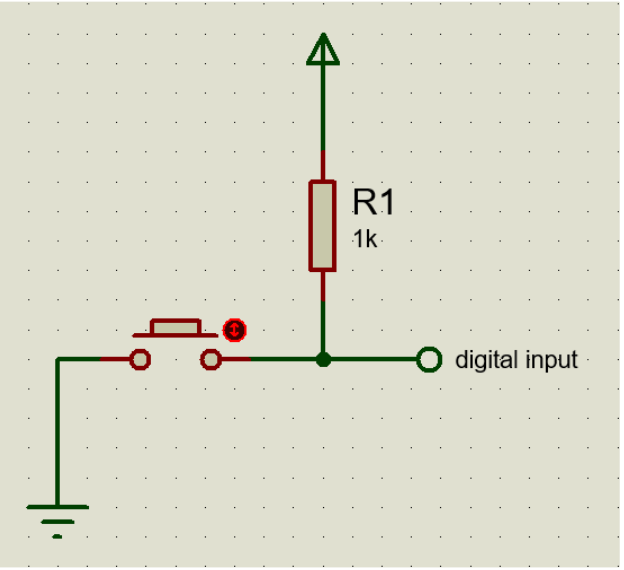
Contents

- Connections
- Example



Module_2-7. Push button Connections

The push-button can be considered as a normally open switch, which closes for as long as we have pressed it. Logical “0” and “1” can be created from the push-button with the appropriate wiring.



Pull-up resistor

The digital input reads “1” as long as the push-button is not pressed.

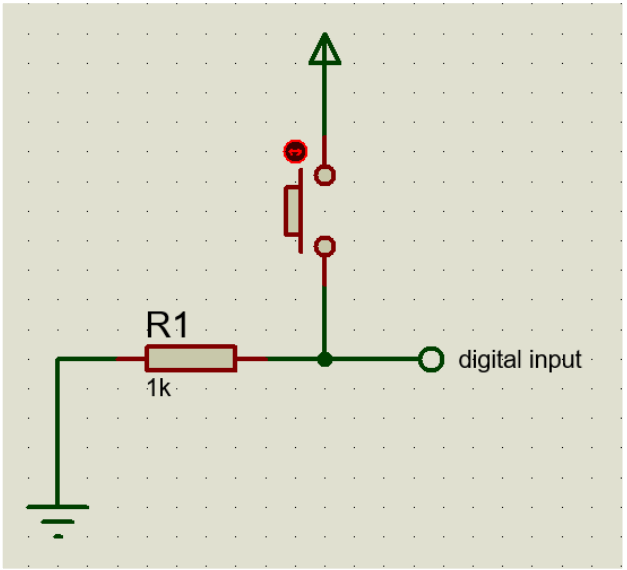
As long as the switch is pressed the input will read “0”.

Module_2-7. Push button Connections

The push-button can be considered as a normally open switch, which closes for as long as we have pressed it. Logical “0” and “1” can be created from the push-button with the appropriate wiring.

The digital input reads “0” as long as the push-button is not pressed.

As long as the switch is pressed the input will read “1”.



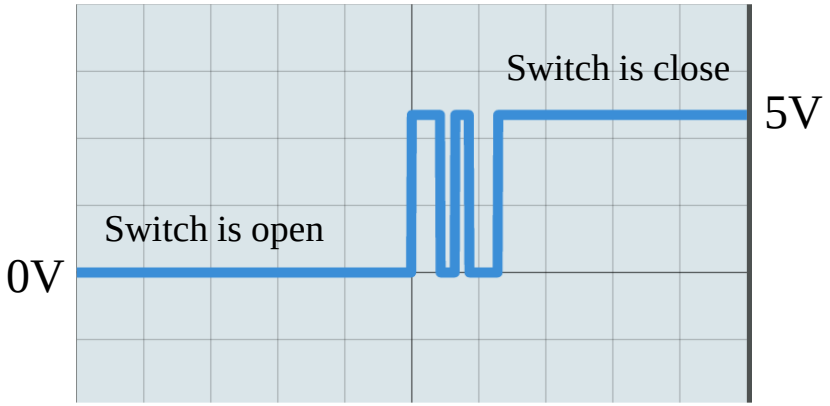
Pull-down resistor

Module_2-7. Push button Connections

Mechanical parts, such as switches and push-buttons, cause the digital input signal to bounce.

These bounces result in the input going back and forth between logical “0” and “1” for a short period of time.

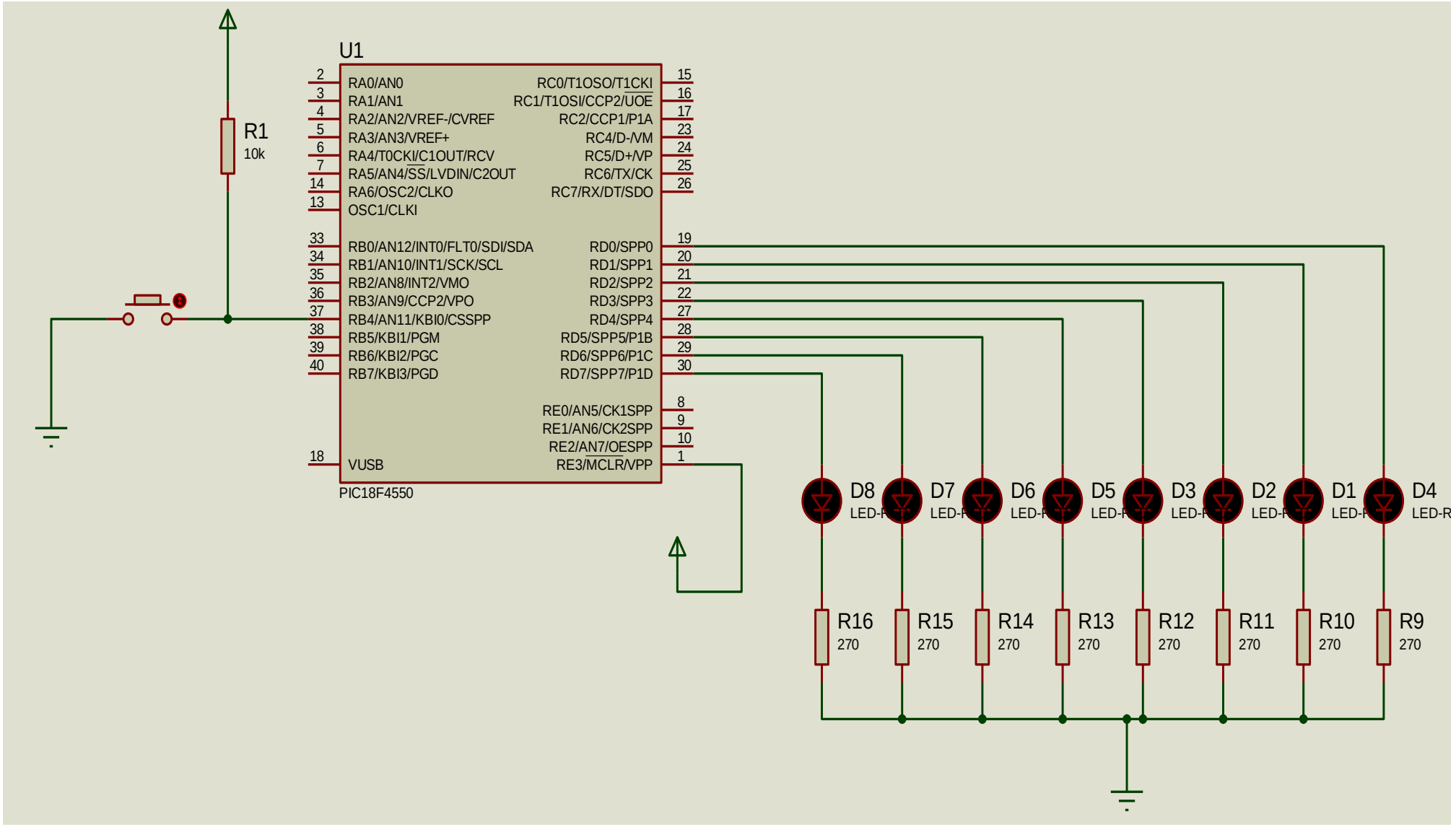
To avoid bounces, an easy and quick solution is to freeze the program for a few milliseconds (approximately 25ms).



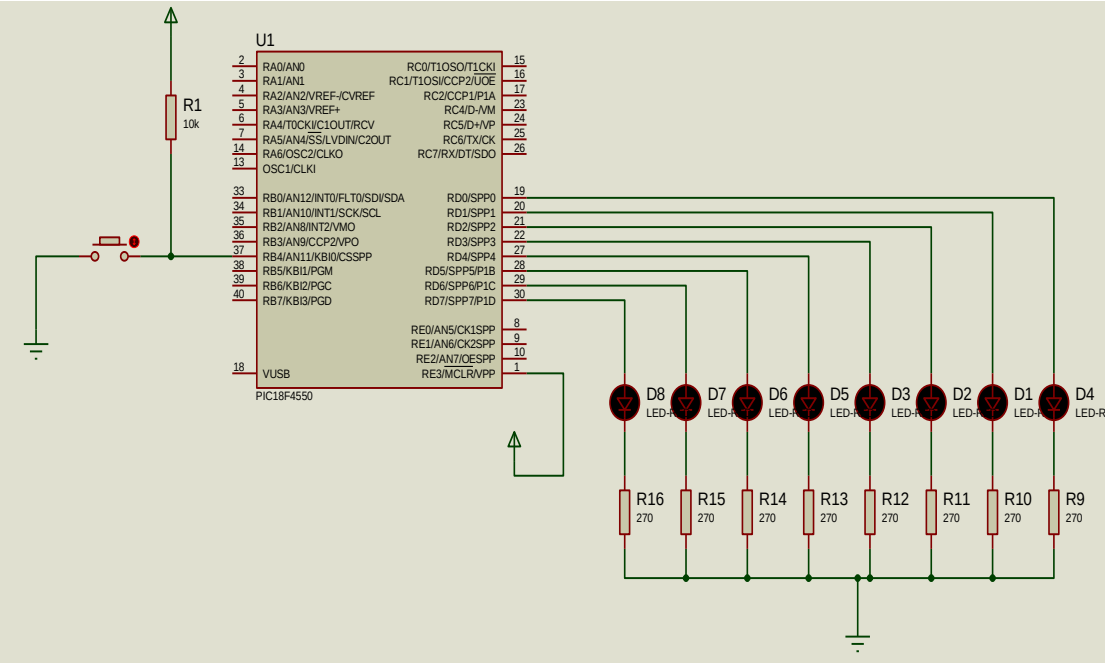
Bounces

1. https://www.allaboutcircuits.com/technical-articles/switch-bounce-how-to-deal-with-it/

Module_2-7. Push button Example



Module_2-7. Push button Example



The purpose is that every time we press and release a button which is connected to pin RB4 the value of PORTD is increased by 1. The result of the increase can be seen on 8 LEDs which are connected to PORTD

Module_2-7. Push button Example

```
#include<main.h>            // This file contains the initial settings
                            // It must be in the same folder with the project
#byte PORTB=0xF81           // F81 Is the position or PORTB data register
                            // at the data memory of the microcontroller
                            // SFR Special Function Register
#byte PORTD=0xF83           // F83 Is the position or PORTB data register
                            // at the data memory of the microcontroller
                            //SFR Special Function Register
// *********   Main program   ************************
void main()
{
  set_tris_d(0x00); //PORTD is set as output port
  set_tris_b(0xFF);  //PORTB is set as input port
  PORTD=0x00;
  while(TRUE) {
     while(input(PIN_B4) == 1) {}
          // Wait until the button is pressed
          // In the wait state, no command is executed
          // When the button is pressed the loop is exited and the next command is executed
    delay_ms(50);
    while(input(PIN_B4) == 0) {}
          // Wait until the button is released
          // In the wait state, no command is executed
          // When the button is pressed the loop is exited and the next command is executed
   delay_ms(50);
   PORTD=PORTD+1;
    }
}
```

ENGINE Partnership

- Warsaw University of Technology (PL) - *coordinator*
- IHU - International Hellenic University (GR)
- EDUMOTIVA - European Lab for Educational Technology (GR)
- University of Padova (IT)
- University of Applied Sciences in Tarnow (PL)

ENGINE ERASMUS+ 2020-1-PL01-KA226-HE-095653

CONTACT:

www.engined.eu

angelika.tefelska@pw.edu.pl

@projectENGINE1

@EUprojectEngine

Erasmus+

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.