American International University- Bangladesh
Department of Electrical and Electronic Engineering
EEE4103: Microprocessor and Embedded Systems Laboratory

Title: Debouncing : Implementation and effects

Introduction:

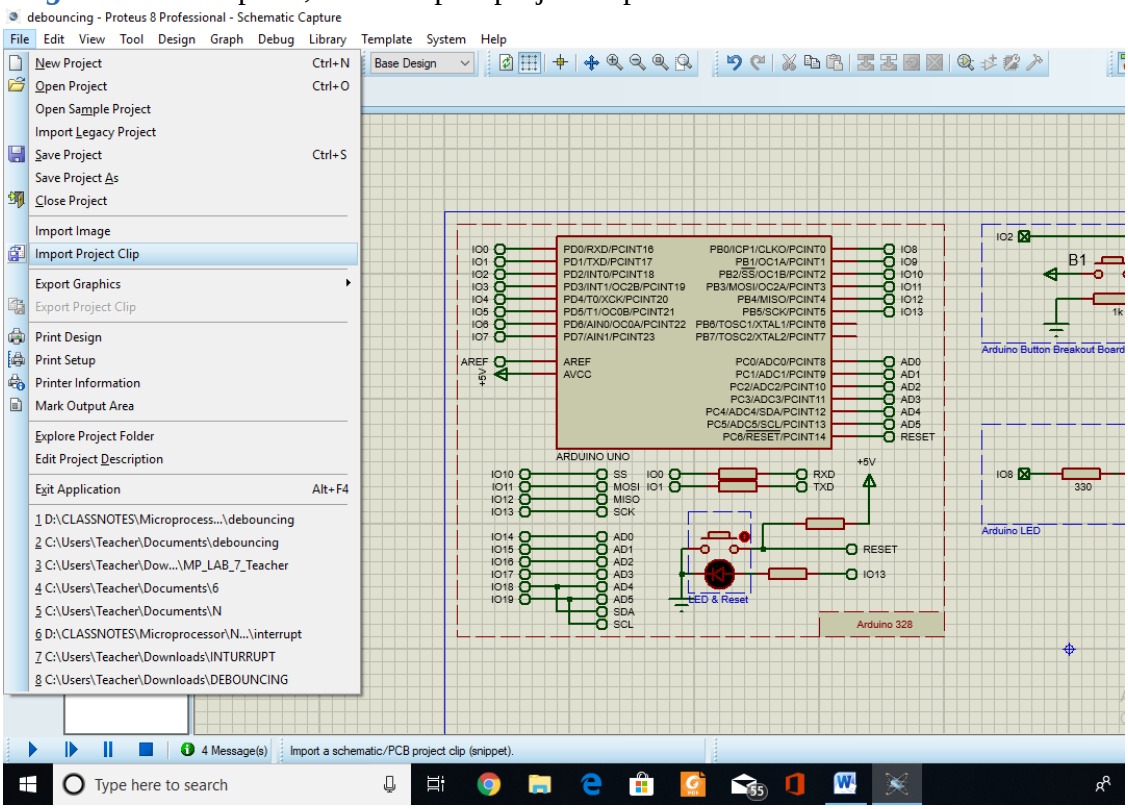
The objective of this experiment is to get familiarized with Debouncing and Interrupts: Implementation and effects in Microcontroller.

Theory and Methodology:

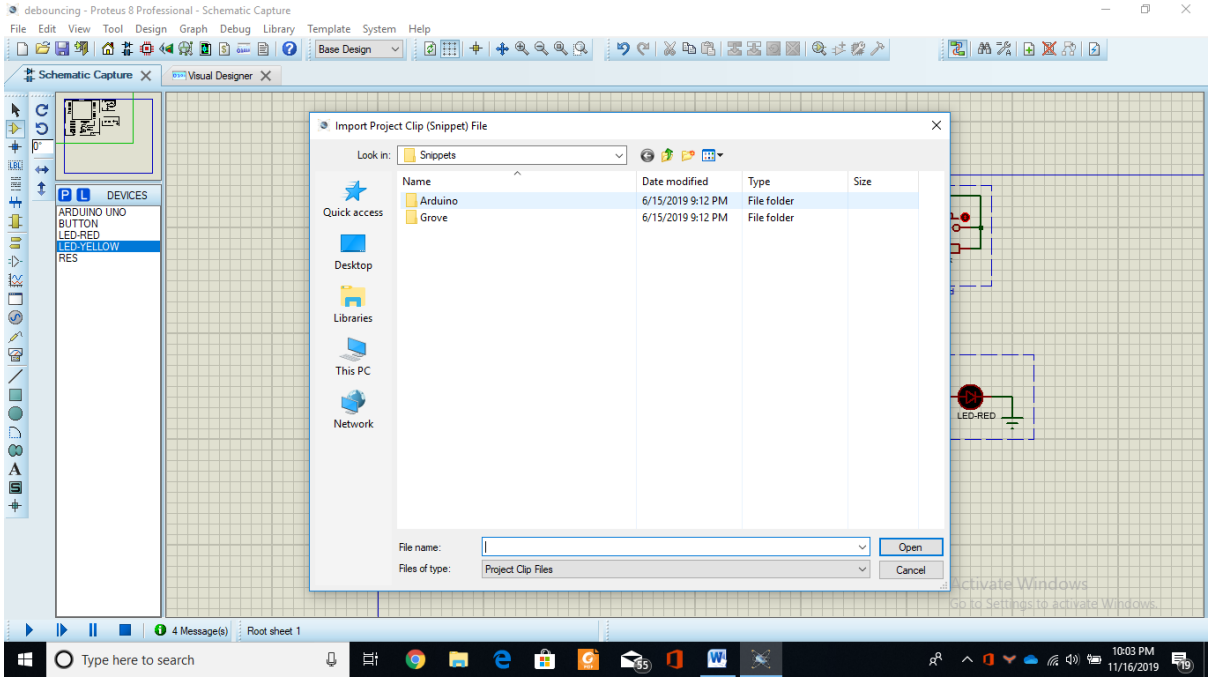
Debouncing: Bouncing is the tendency of any two metal contacts in an electronic device to generate multiple signals as the contacts close or open; debouncing is any kind of hardware device or software that ensures that only a single signal will be acted upon for a single opening or closing of a contact.

Drawing on the Schematic: There is a good deal of documentation on schematic drawing in the main Proteus help files so we'll cover only in brief here. Our task is to draw the following circuitry on the schematic:

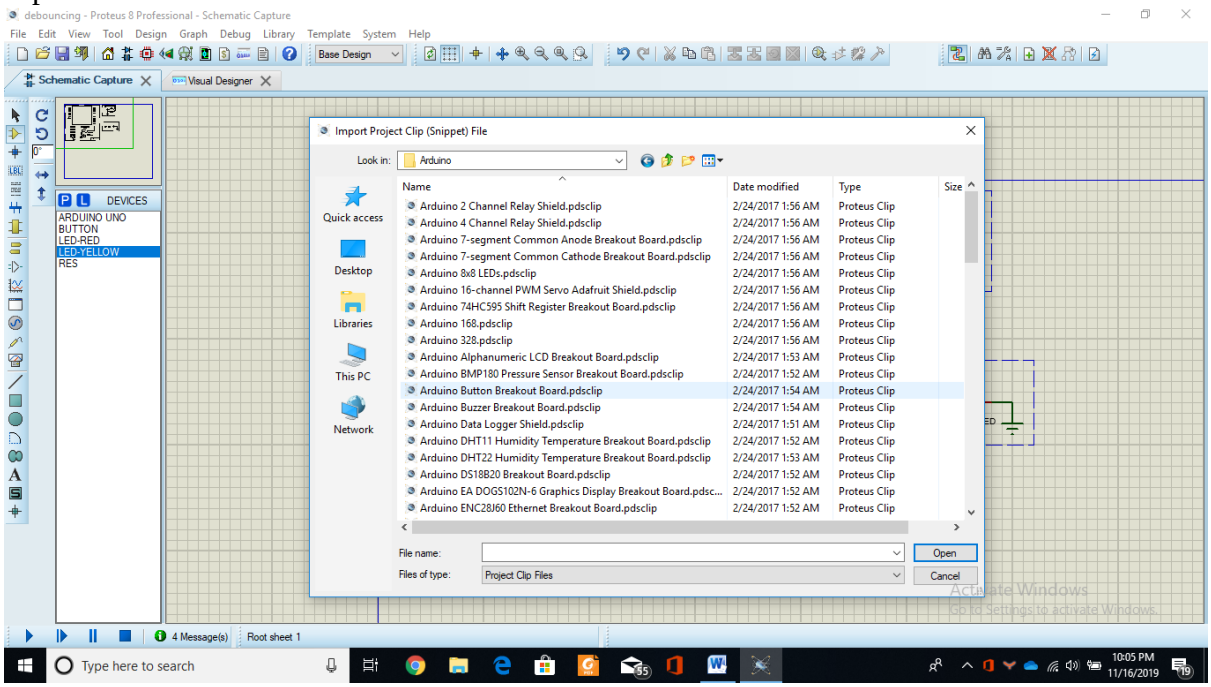
Picking: From files option, select import project clip.

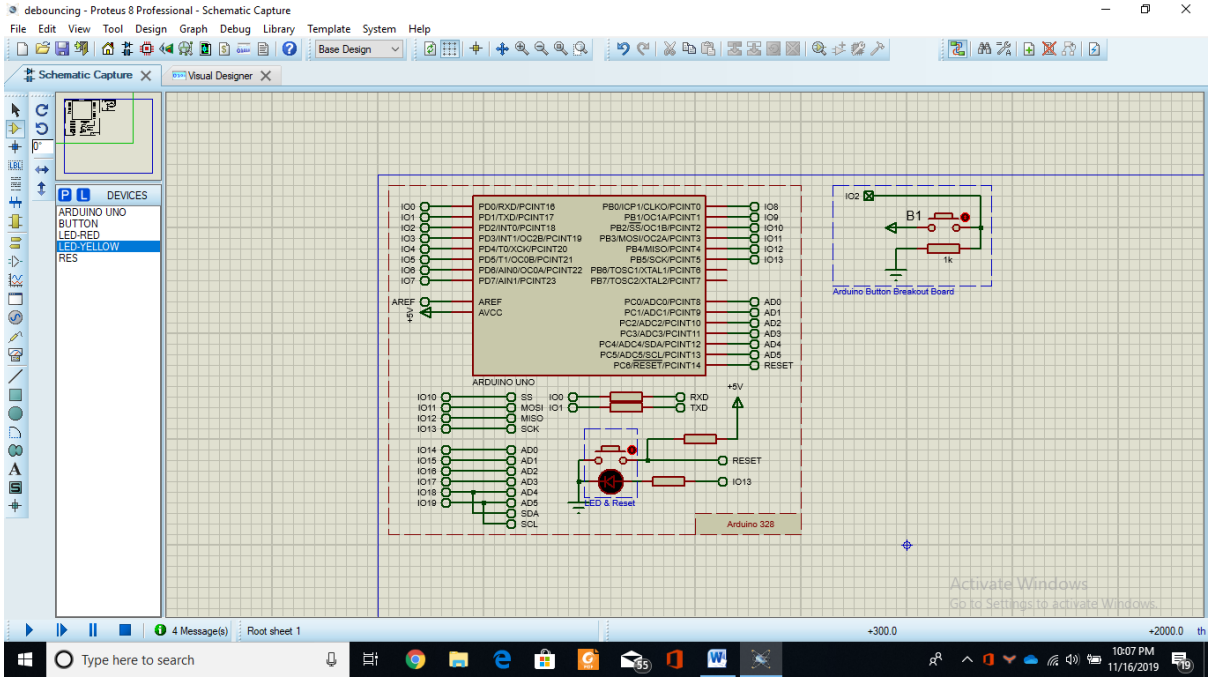


A dialogue box will open as below :

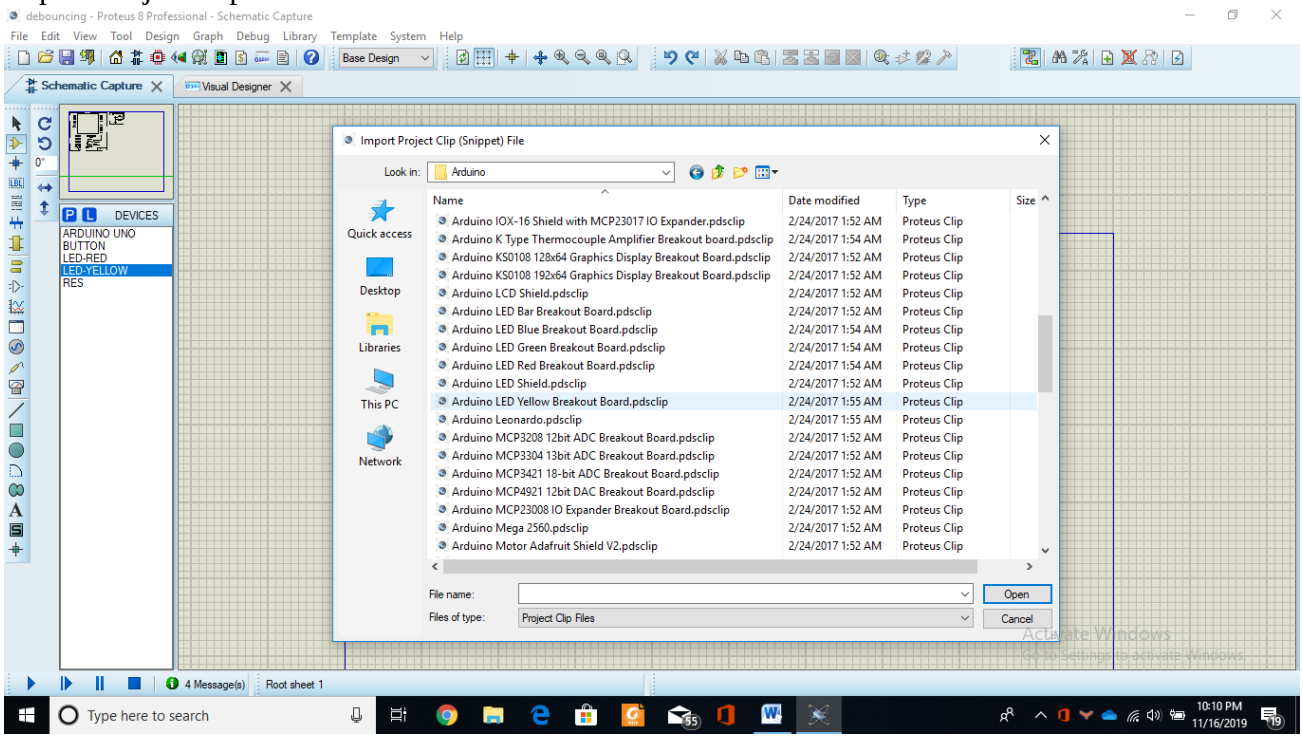


Select Arduino and then select Arduino Button Breakout Board and place it onto schematic capture as below:

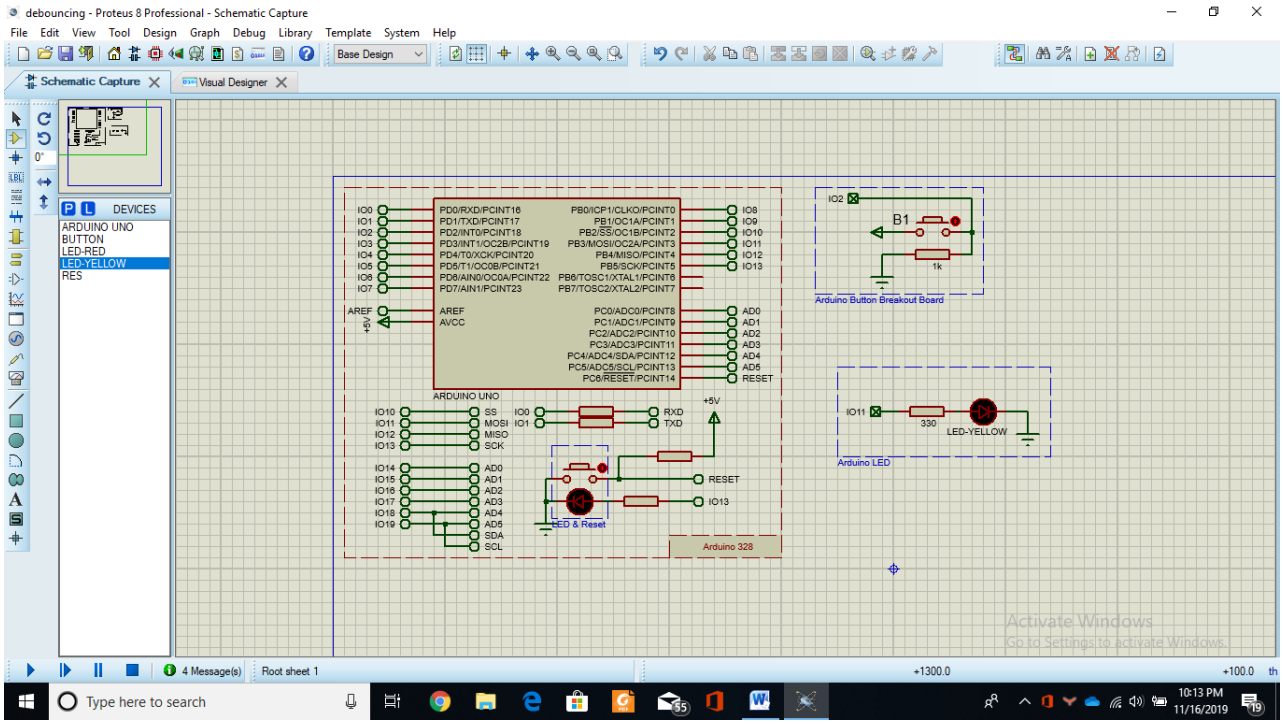




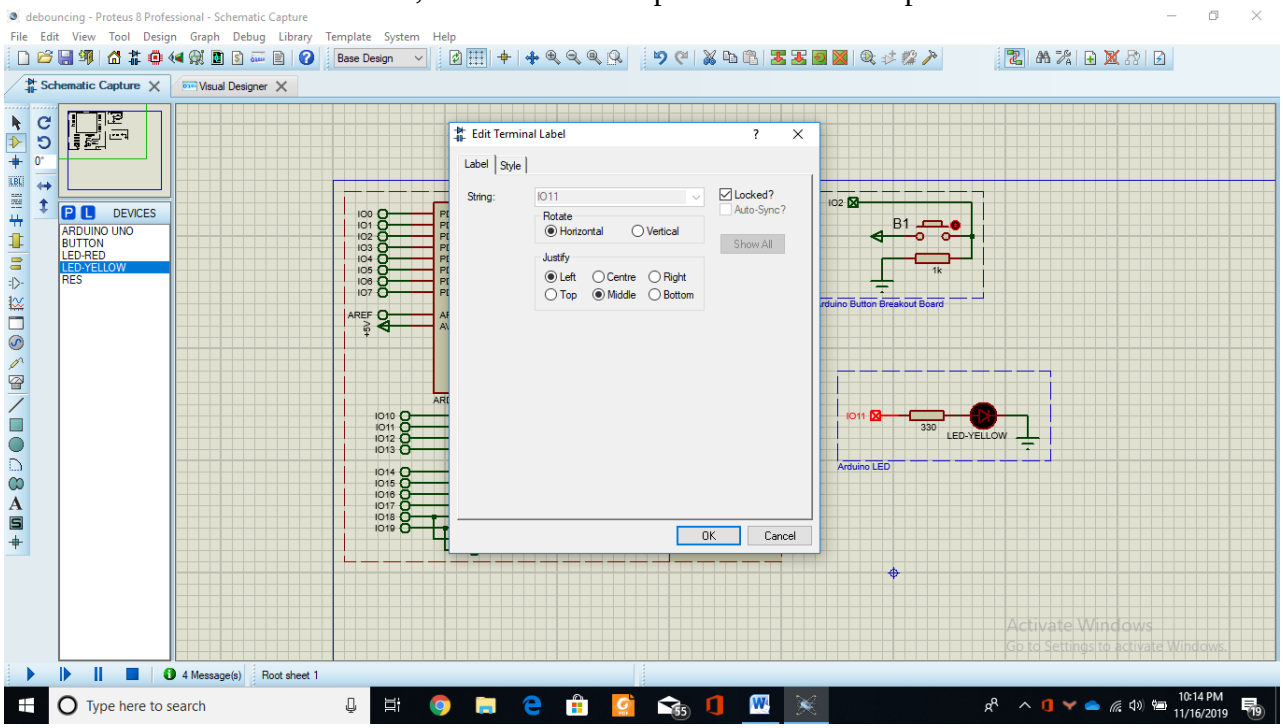
Following same process again select Arduino LED Yellow Breakout Board after selecting Import Project clip from Files .

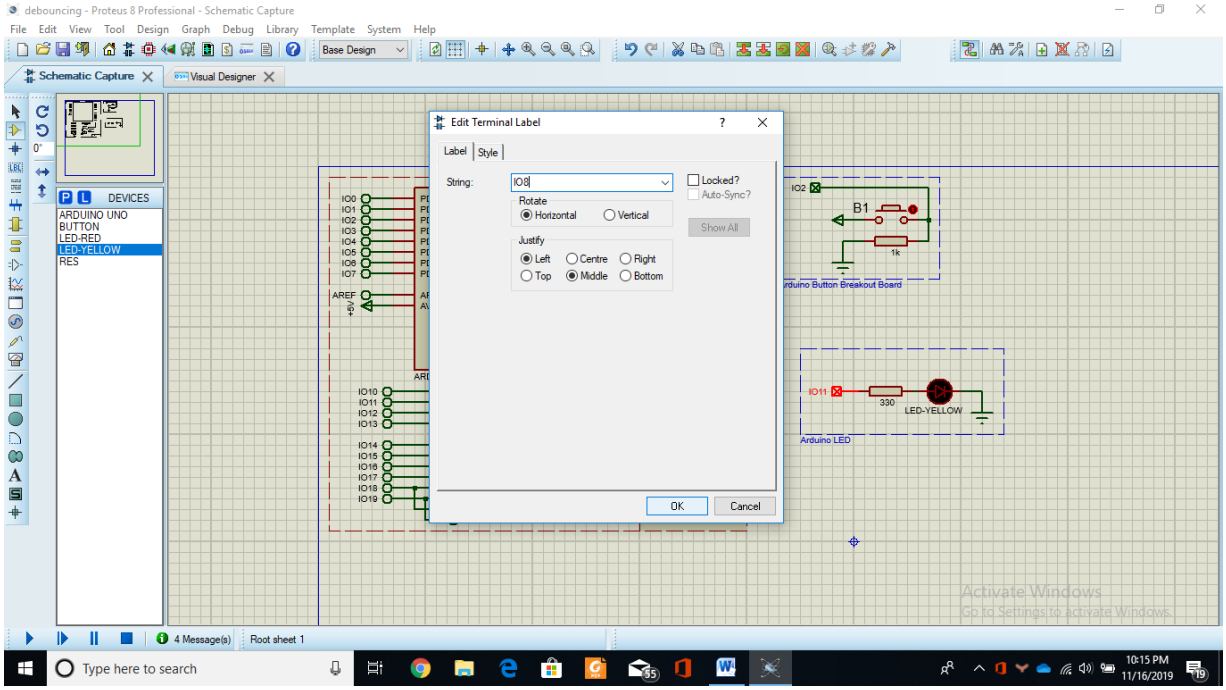


Place it onto Schematic diagram as well.

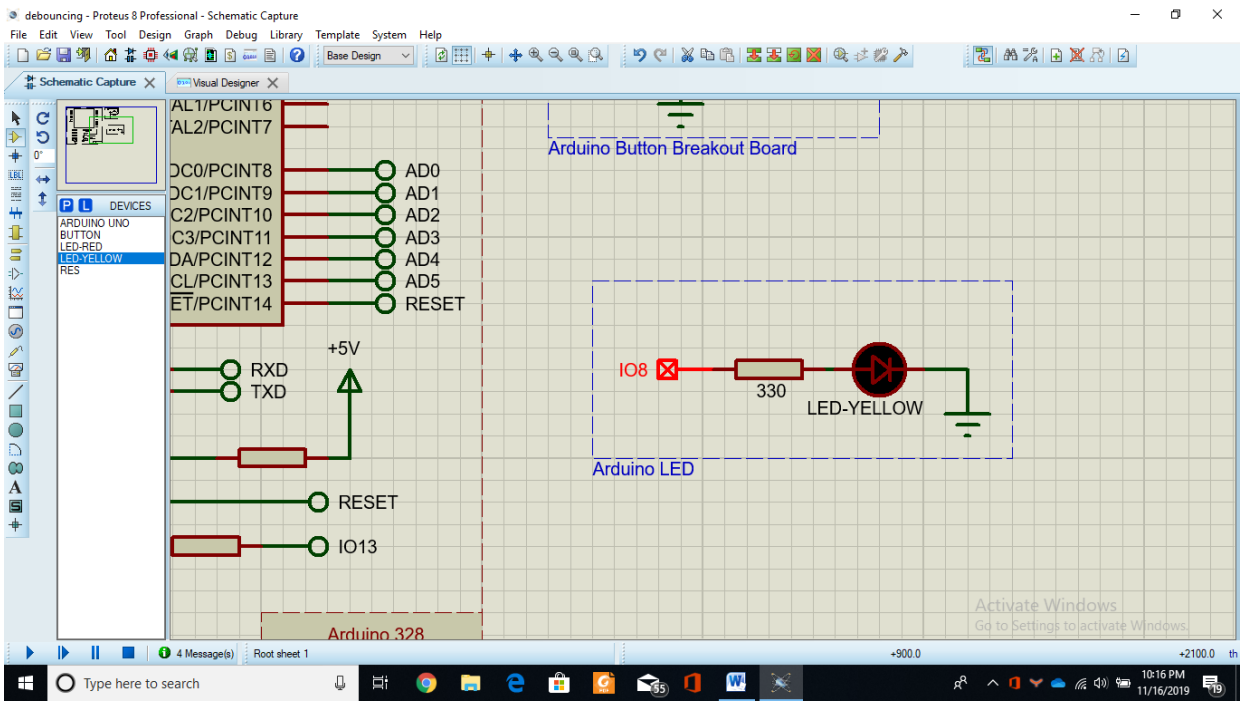


Next double click on label IO11, unselect locked option to edit the output terminal to IO8.

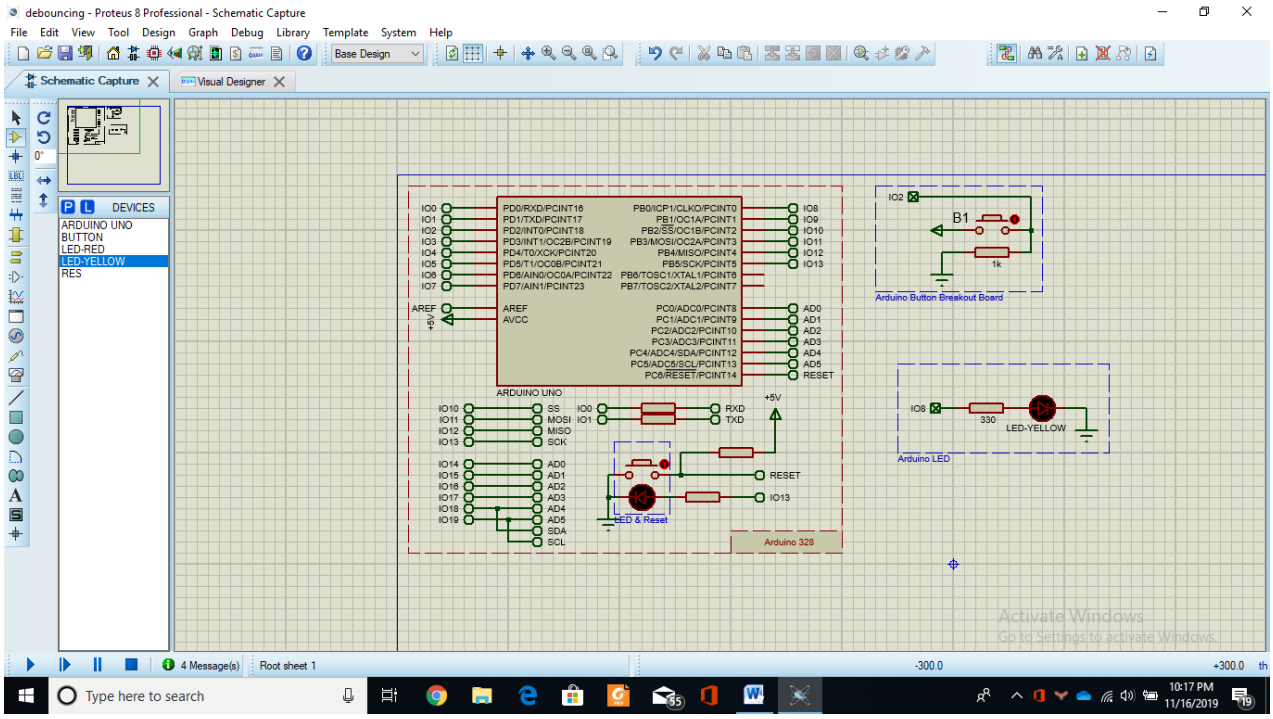




Click OK .



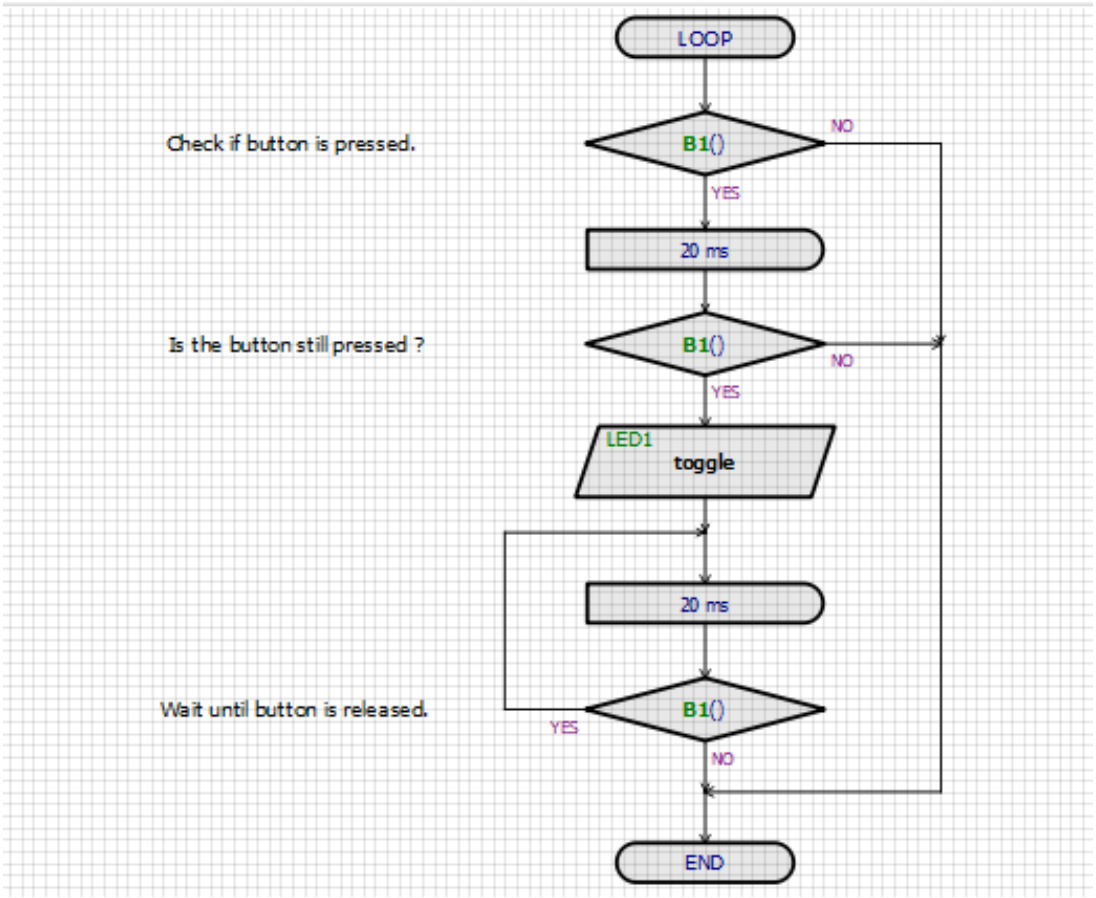
Simulation Setup for Debouncing: Final simulation would look like as below:



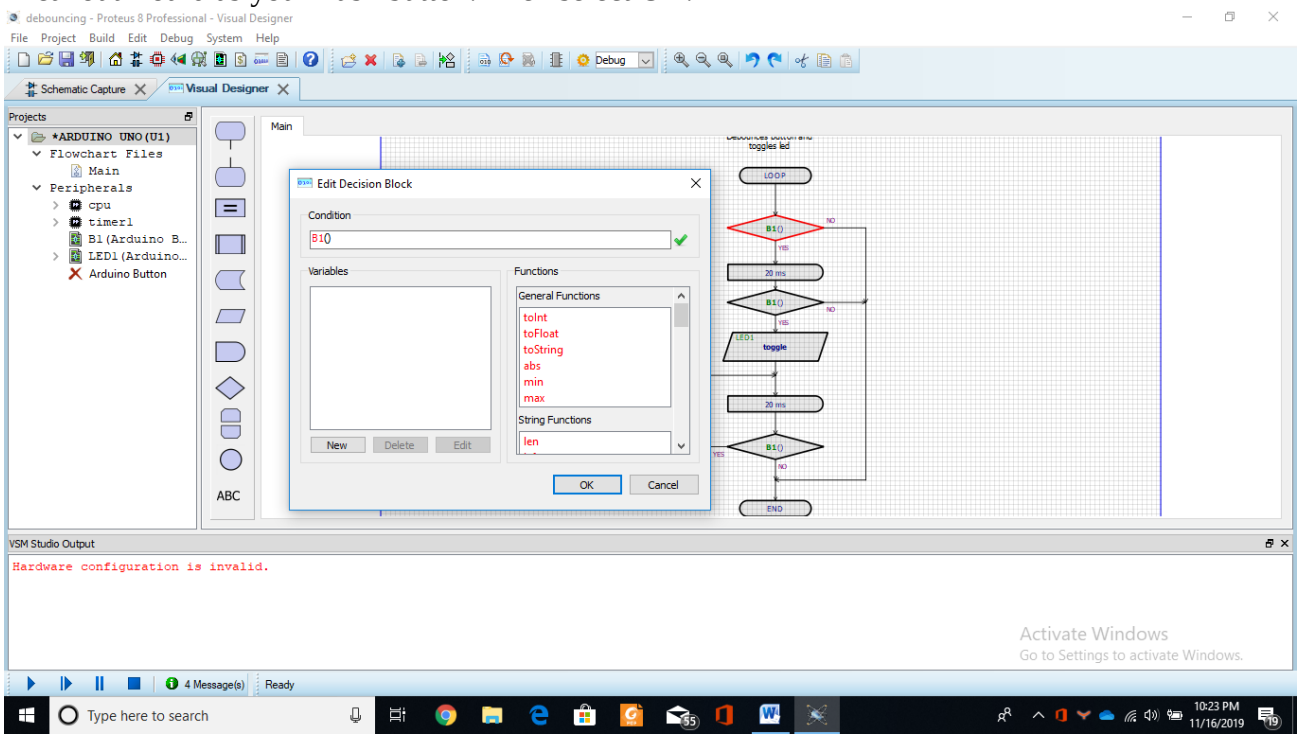
Designing our Program

Everything we have done so far is hardware design. The program design itself takes place on the Visual Designer tab. Unlike our other tutorials we don't have any external peripheral methods in the project tree so we will have to drive the electronics directly using the CPU methods.

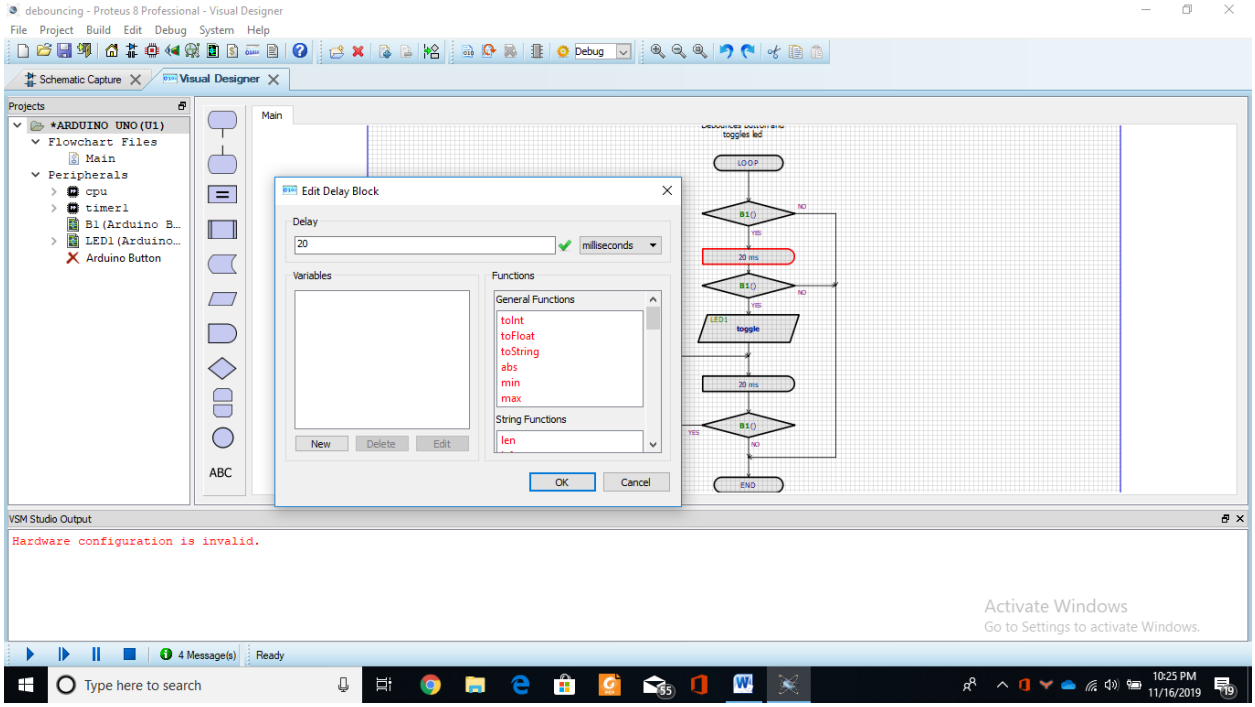
Flowchart for Debouncing:



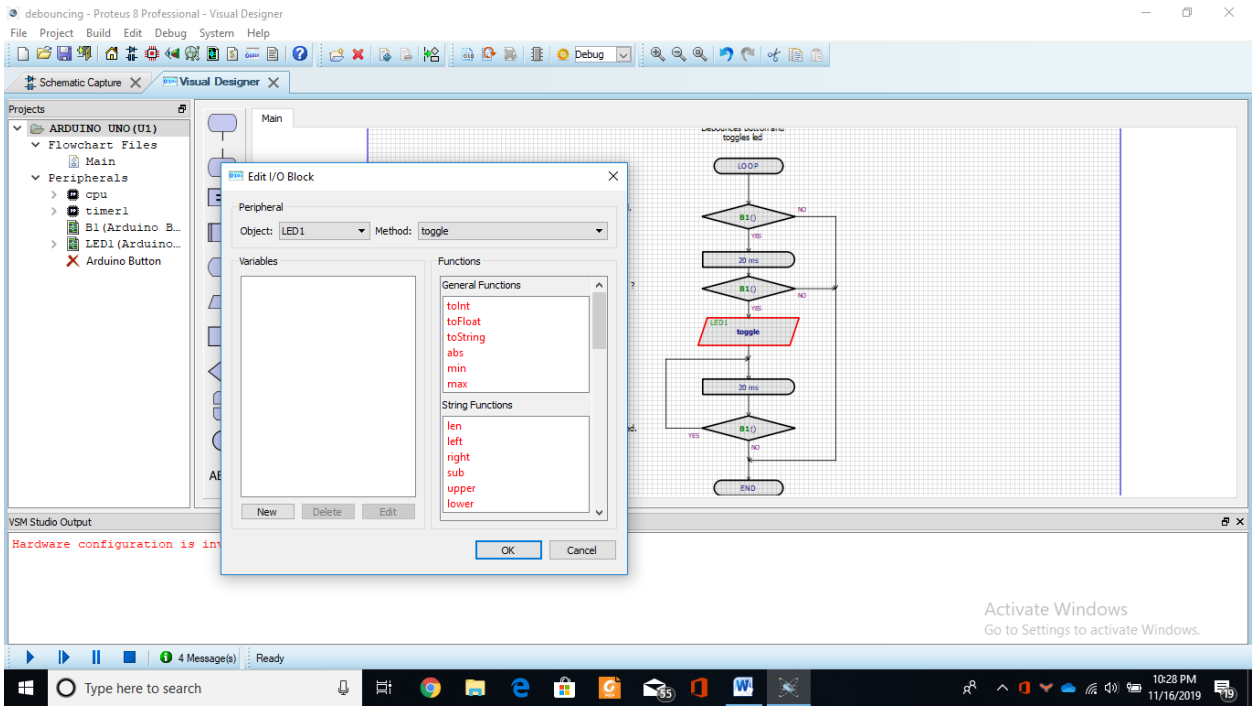
Take a decision block and double click on to edit it, write B1() to select Arduino Button Breakout Board as your Push button. Then select OK.



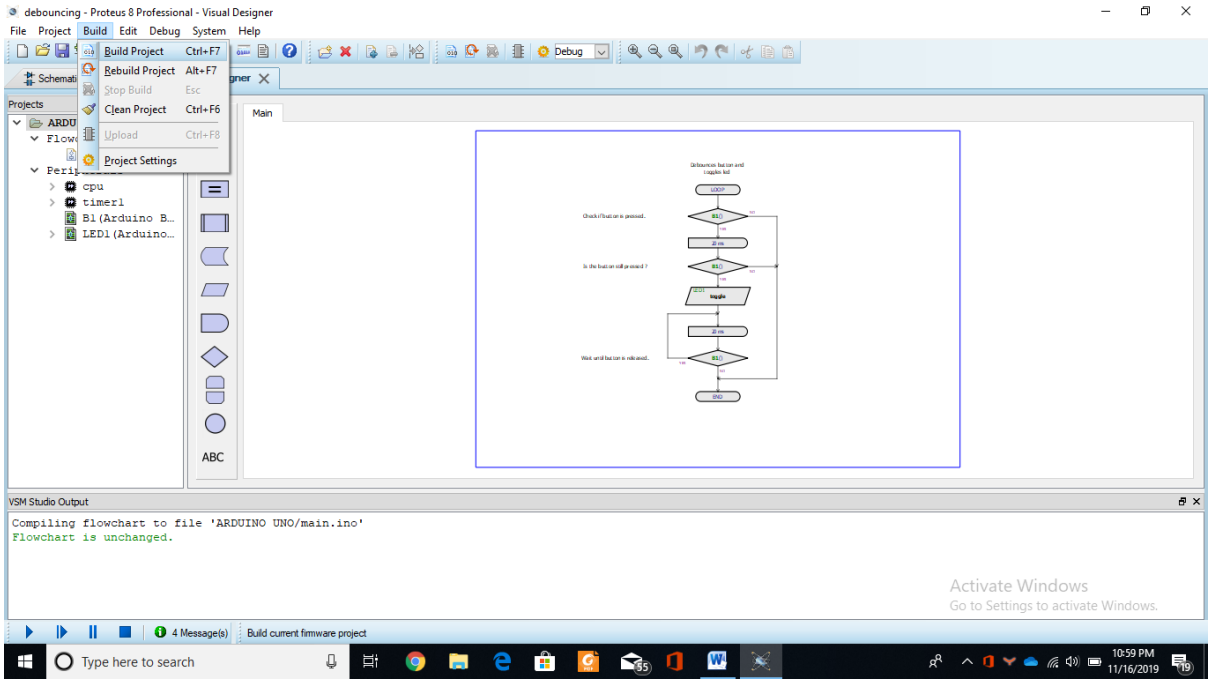
Select your delay time as 20 milliseconds for time delay block and then select OK.



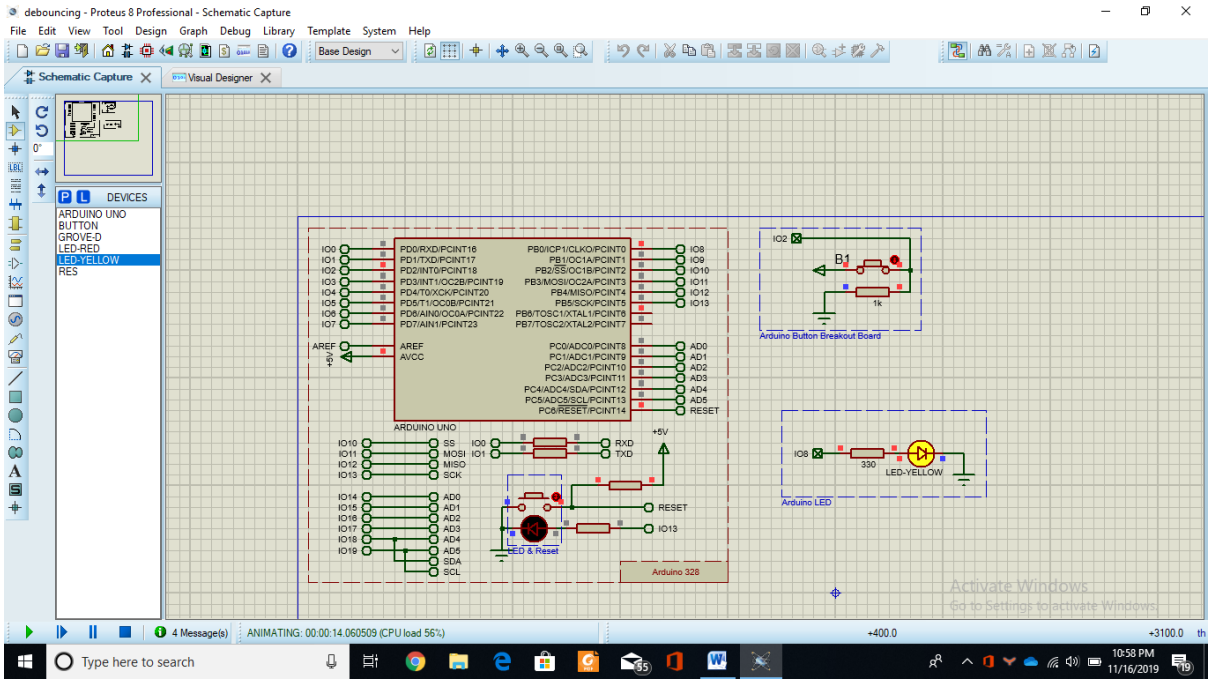
Select B1() for second decision block also and select LED1 of object and toggle as method for I/O peripheral operation as shown below :



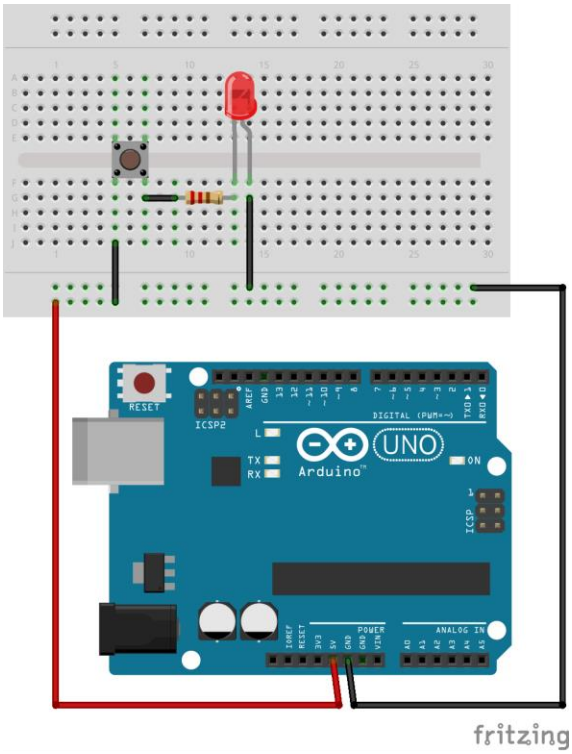
Then select OK and set the time as 20 milliseconds as time delay too before 3rd decision block and then select B1() for 3rd decision block too. Then connect the wires and select build project then in VSM studio, compiled successfully will be shown as below:



Play the button in schematic capture and click on the push button to see the LED on. Thus we understand that debouncing is operated successfully.



Experimental setup:



Apparatus:

- Arduino Uno
- LED
- Push Button

Code for debouncing:

```
int switchPin =8;
int ledPin=13;
boolean lastButton =LOW;
boolean currentButton = LOW;
boolean ledOn = false;

void setup() {
  pinMode(switchPin,INPUT);
  pinMode(ledPin,OUTPUT);
}

boolean debounce(boolean last)
{
  boolean current = digitalRead(switchPin);
  if (last != current)
  {
    delay(5);
    current = digitalRead(switchPin);
  }
```

```
  return current;
}

void loop() {

  currentButton = debounce(lastButton);
  if (lastButton == LOW && currentButton == HIGH)
  {
    ledOn =!ledOn;

    }

 lastButton = currentButton;
digitalWrite(ledPin,ledOn);
}
```

Questions for report writing:

1) Include all codes and scripts into lab report following the writing template mentioned in appendix A of Laboratory Sheet Experiment 1.

Reference(s):

1) https://www.arduino.cc/.
2) https://www.coursera.org/learn/arduino/lecture/ei4ni/1-10-first-glance-at-a-program