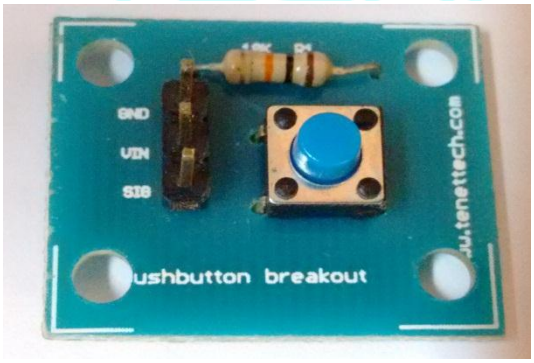
2016

Application Note on Interfacing Arduino with Push-Button



Author: vivek.g.s
Reviewer:
Version1.0

Interfacing Arduino UNO with Push-Button

Introduction

In this application note we will be discussing on interfacing Push-Button with Arduino UNO to blink an LED whenever a button is pressed.

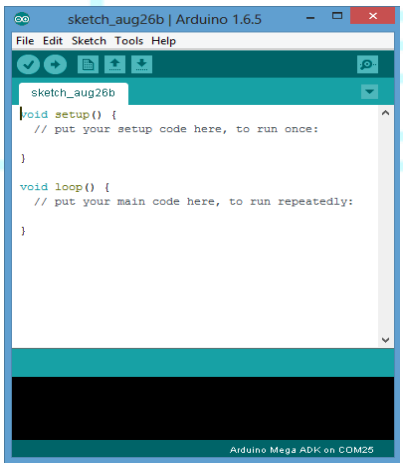
Arduino UNO: Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. All this is defined by a set of instructions programmed through the Arduino Software (IDE).

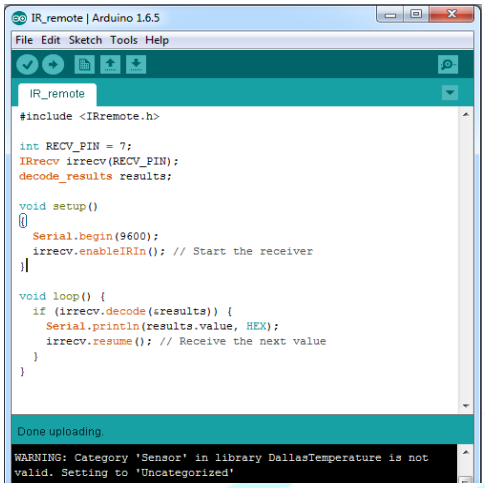
Push-button: A push-button (also spelled pushbutton) or simply button is a simple switch mechanism for controlling some aspect of a machine or a process. Buttons are typically made out of hard material, usually plastic or metal.[1] The surface is usually flat or shaped to accommodate the human finger or hand, so as to be easily depressed or pushed. Buttons are most often biased switches, though even many un-biased buttons (due to their physical nature) require a spring to return to their un-pushed state.

Step1. The Materials required are:

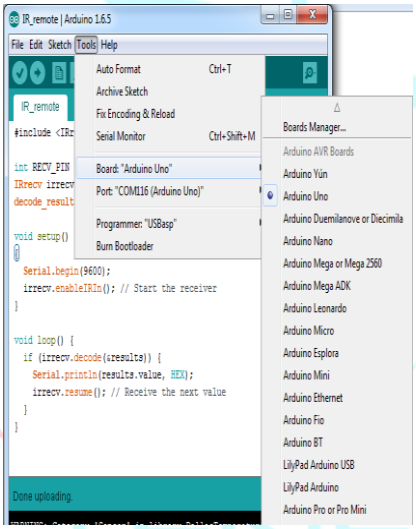
- Arduino UNO
- Push-Button Breakout
- Male to male Jumpers

1. Open Arduino sketch on your PC or Laptop to start the programming.

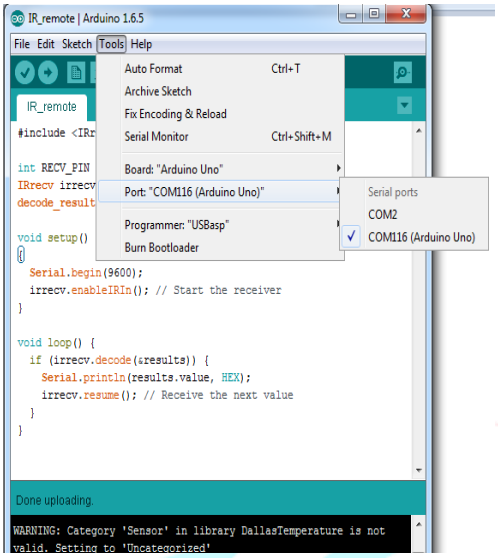




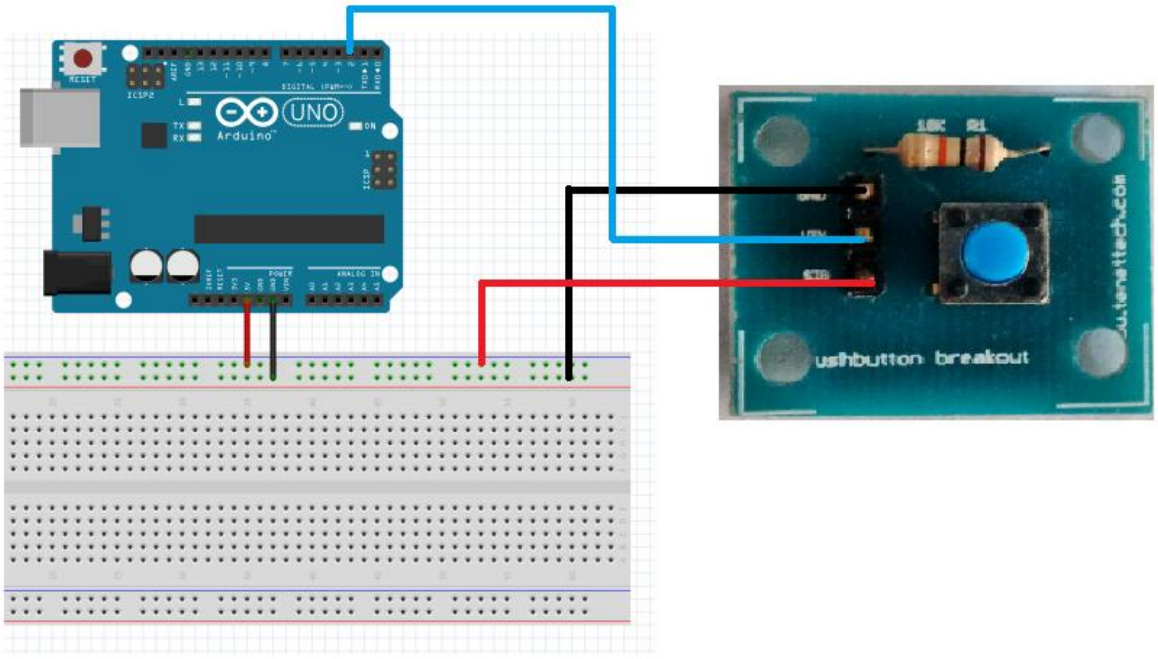
- Type the program for the LED to turn on for 1 sec and turn off 1 sec.
- Click on verify and check for any errors in the program. If no errors are present select the Arduino UNO in IDE. Go to tools> Board> Select Arduino UNO.



- Select port of programming by Tools> Port> Select the port for programming



- Now Upload the program to the arduino

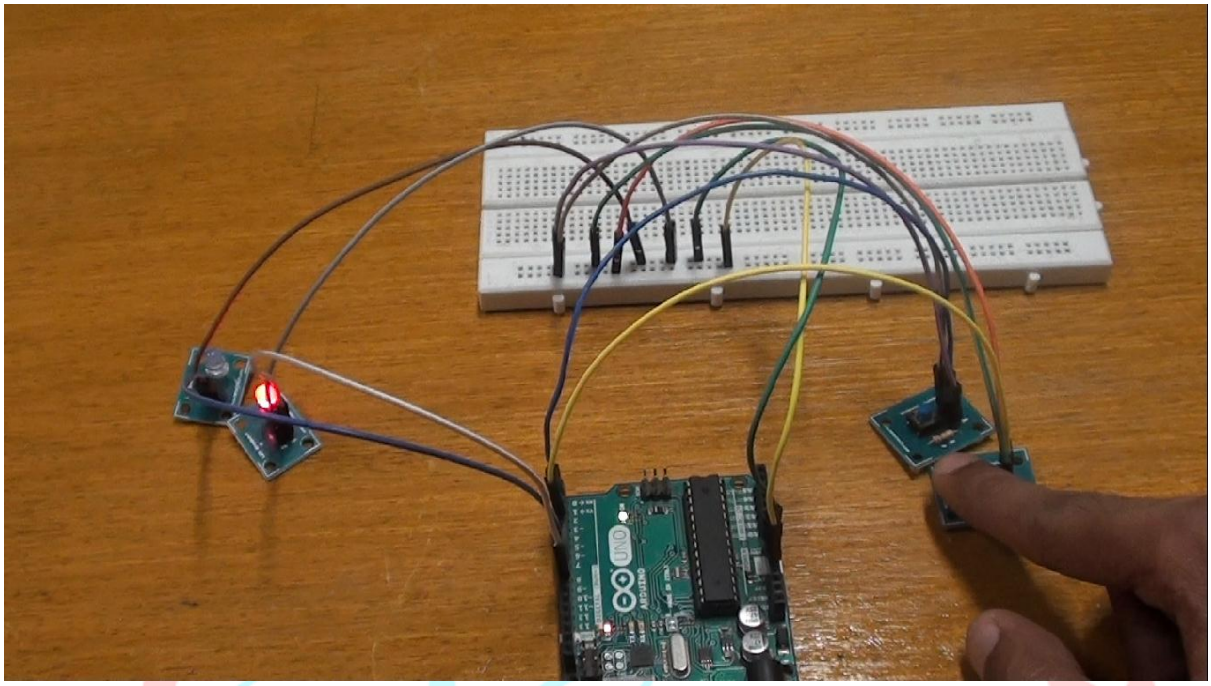


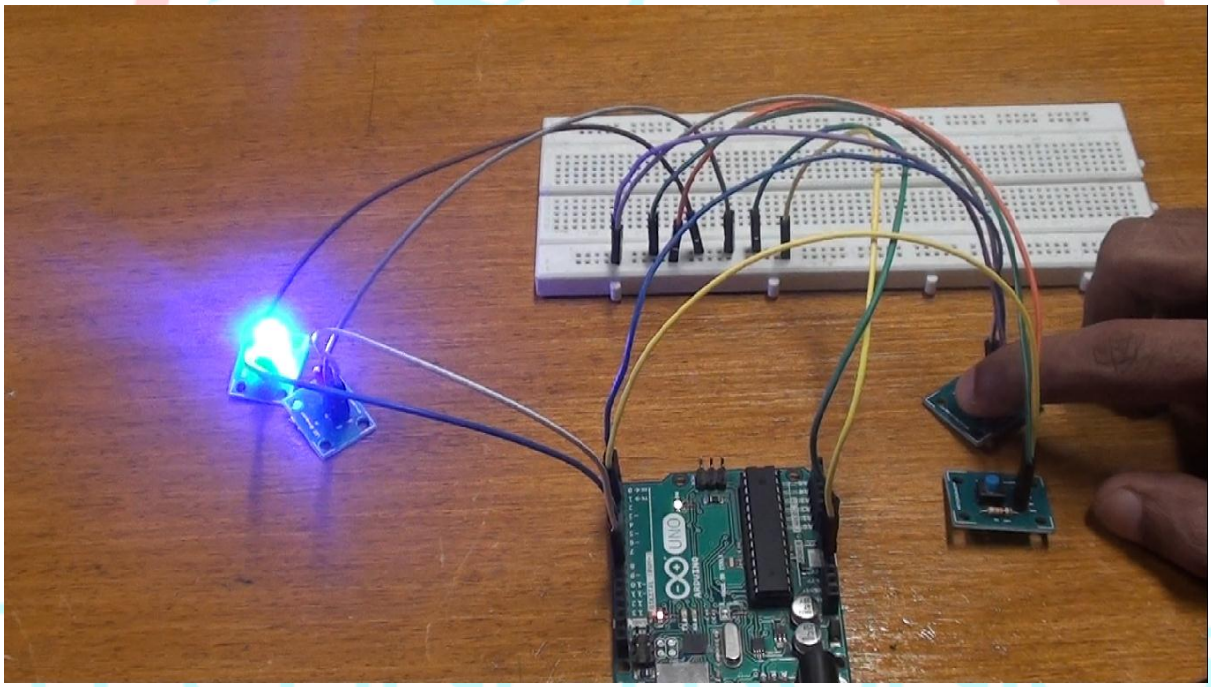
CODE:

```
int pushButton = 2;

int LED = 8;


// the setup routine runs once when you press reset:

void setup() {
```

```
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // make the pushbutton's pin an input:
  pinMode(pushButton, INPUT);
  pinMode(LED, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input pin:
  int buttonState = digitalRead(pushButton);
  // print out the state of the button:
  Serial.println(buttonState);
  if(buttonState == HIGH)
  {
 digitalWrite(LED, HIGH);
  }
  else
  {
    digitalWrite(LED, LOW);
  }
}
```

OUTPUT:





For more information please visit: www.tenettech.com

For technical query please send an e-mail: info@tenettech.com

For product info:

1. **http://www.tenettech.com/search?q=arduino+uno&r1=default**

9/3, 2nd floor, SreeLaksmi Complex, opp, to Vivekananda Park, Girinagar, Bangalore - 560085,

Email: info@tenettech.com, Phone: 080 - 26722726

2.