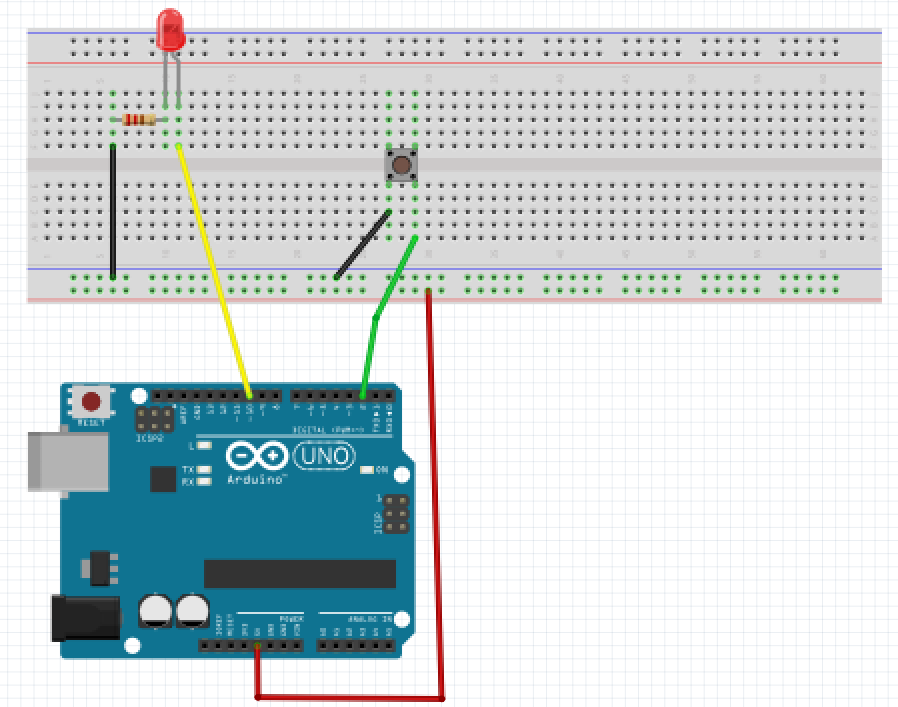
**ARDUINO LED PROJECT**

**Button without external resistors**

For this project we will need:

* Arduino board.
* Button switch
* LED
* Breadboard.
* 470 Ohm resistor.
* 5 Male to Male wires.

Circuit Design:

1. First make sure that the Arduino is powered off (no USB cable plugged to power).
2. Check the mini push switch.
3. Plug either side to a horizontal line on the breadboard.
4. One side plug to the common vertical. ground of the breadboard(+)
5. The other end plug a 330 Ohm resistor and connect to the common 5V of the breadboard.
6. In between the resistor and the 2nd end of the button switch plug a green wire to pin 2 of the Arduino.
7. Plug the longer leg of the led to the breadboard on a separate horizontal line .Using a yellow jumper wire connect from this line to pin 10 of the Arduino.
8. Plug the shorter leg of the led to the breadboard on a separate horizontal line. Plug a 330 Ohm resistor connect from this line to the common ground of the breadboard.
9. Using a red jumper wire to connect the common 5V of the breadboard to 5V of the Arduino.
10. Using a black jumper wire to connect the common ground of the breadboard to GND of the Arduino

|  |
| --- |
| const int buttonPin=2;  const int ledPin = 10;  int sensorValue;  void setup(){  Serial.begin(9600);  pinMode(buttonPin, INPUT\_PULLUP);  pinMode(ledPin,OUTPUT);}  void loop(){  sensorValue=digitalRead (buttonPin);  if( sensorValue == LOW){  digitalWrite(ledPin, HIGH);  delay(100);  }  else{  digitalWrite(ledPin, LOW);  delay(100);  }  } |

const int buttonPin=2;

const int ledPin = 10;

int sensorValue;

This declares the pins we will use to connect to the Arduino.

We create a value to store our input from the switch.

const shows that the values are not expected to change during program execution.

int shows that the value to be stored is of numeric type

Serial.begin(9600);

pinMode(buttonPin, INPUT\_PULLUP);

pinMode(ledPin,OUTPUT);

We declare the pin functionalities. This instructs the Arduino to utilize these pins as either send or receive input or output.

INPUT\_PULLUP instructs the arduino to use it internal resistor.

This in turn reverses the effect of current on the switch

sensorValue=digitalRead (buttonPin);

This reads the values received from the pin attached to sensor and stores it in number form.

if( sensorValue == LOW){

digitalWrite(ledPin, HIGH);

The button switch will send a HIGH signal only if the switch is pressed. This completes the circuit.

This will turn instruct the led to turn on

The internal resistor will be low since the outer circuit through the button is complete compared to the inner circuit which has high resistance thus a low output.

else{

digitalWrite(ledPin, LOW);

if the input value did not meet the condition the program will send a signal LOW to the led.

This will turn it off.

void setup (){}

This initializes the arduino and assigns functionality to its pins.

This also provides required resources for monitoring.

void loop(){}

After executing the void setup() function, we enter the void loop() and this function is executed continuously and repeatedly, until you Arduino is powered off.