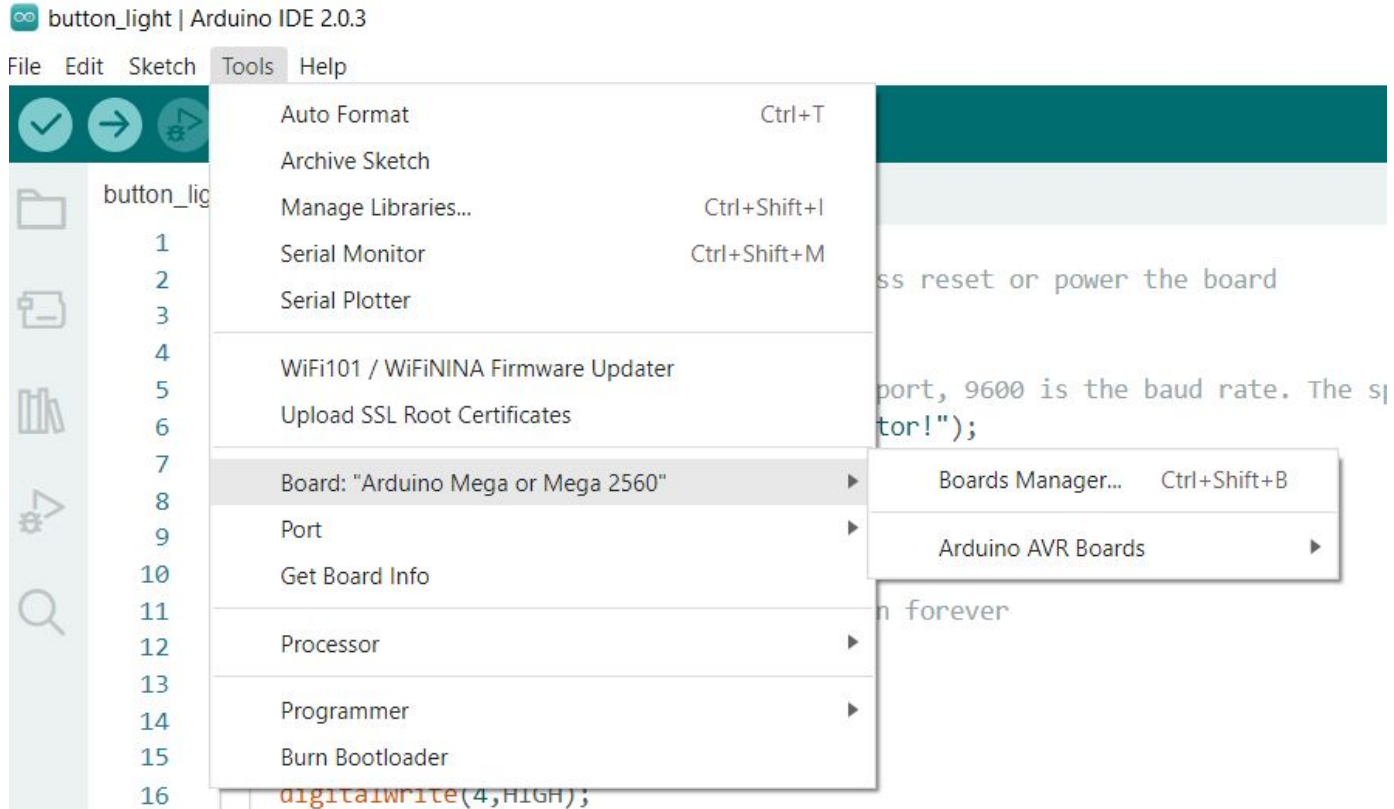


Arduino Coding

<https://www.arduino.cc/en/software>

Installing Board driver



Install Arduino AVR Boards

BOARDS MANAGER

Filter your search...

Type: All

Arduino AVR Boards by **Arduino**
Version 1.8.5 **INSTALLED**

Boards included in this package:
Linino One, Arduino Nano, Arduino
Mini, Arduino Fio, LilyPad Arduino
USB, LilyPad Arduino, Arduino
Gemma, Adafruit Circuit
Playground, Arduino Yún, Arduino
Duemilanove or Diecimila, Arduino
Mega ADK, Arduino Robot Motor,
Arduino Leonardo, Arduino Micro,
Arduino Esplora, Arduino Uno,
Arduino Mega or Mega 2560,
Arduino Leonardo ETH, Arduino BT,
Arduino Uno Mini, Arduino Ethernet,
Arduino Pro or Pro Mini, Arduino
Yún Mini, Arduino NG or older,
Arduino Robot Control, Arduino
Industrial 101, Arduino Uno WiFi
[More info](#)

1.8.6

INSTALL

button_li
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

Select Board: Tools -> Board -> Arduino AVR Boards -> arduino mega or mega 2560

Select port: You will see the arduino port when are connected

Verify : only compile your code

Upload : compile your code and upload it to arduino board

Serial monitor -> tools -> serial monitor

Task

A simple led light that will activate when a button is pressed.

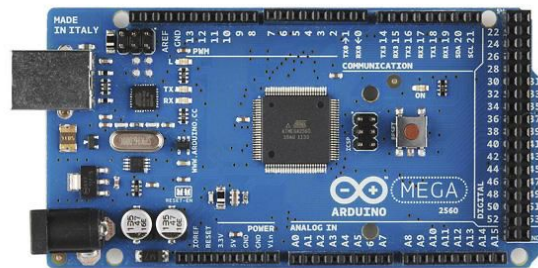
Button pressed -> Light on

Button off -> light off

```
void setup() {  
    // the setup function runs once when you press reset or power the  
board  
  
}
```

```
void loop() {  
    // the loop function runs over and over again forever  
  
}
```

```
// the setup function runs once when you press  
reset or power the board  
  
void setup() {  
    Serial.begin(9600); // sets up the serial  
    port, 9600 is the baud rate. The speed at  
    which a signal can travel  
  
    Serial.println("Hello from the Serial  
    monitor!");  
  
    pinMode(4, OUTPUT); // Output is the macro  
    pinMode(2, INPUT);  
  
}
```




```
// the loop function runs over and over again forever
```

```
void loop() {
```

```
    int input = digitalRead(2);
```

```
    if(input == HIGH) {
```

```
        digitalWrite(4,HIGH);
```

```
        Serial.println("Button has been pressed");
```

```
        delay(1000);
```

```
    }else{
```

```
        digitalWrite(4,LOW);
```

```
    }
```

```
}
```

```
// the loop function runs over and over again forever
```

```
void loop() {
```

```
  char input = digitalRead(2);
```

```
  if(input == HIGH){
```

```
    digitalWrite(4,HIGH);
```

```
    Serial.println("Button has been pressed");
```

```
    delay(1000);
```

```
  }else{
```

```
    digitalWrite(4,LOW);
```

```
  }
```

```
char serialInput = Serial.read(); //asking for input
```

```
  if(serialInput == 'a'){ //what is happening here?
```

```
    digitalWrite(4,HIGH);
```

```
    delay(1000);
```

```
    digitalWrite(4,LOW);
```

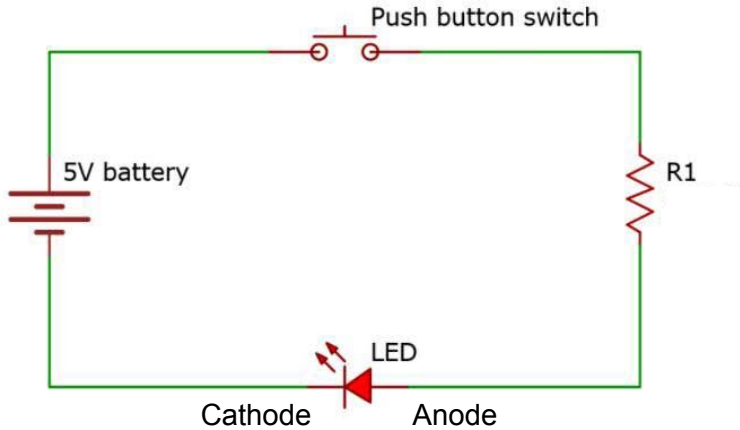
```
  }else if(serialInput == 'b'){
```

```
    Serial.println(serialInput);
```

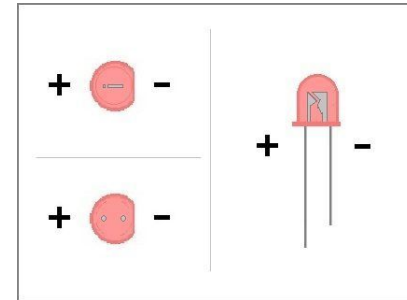
```
  }
```

```
}
```

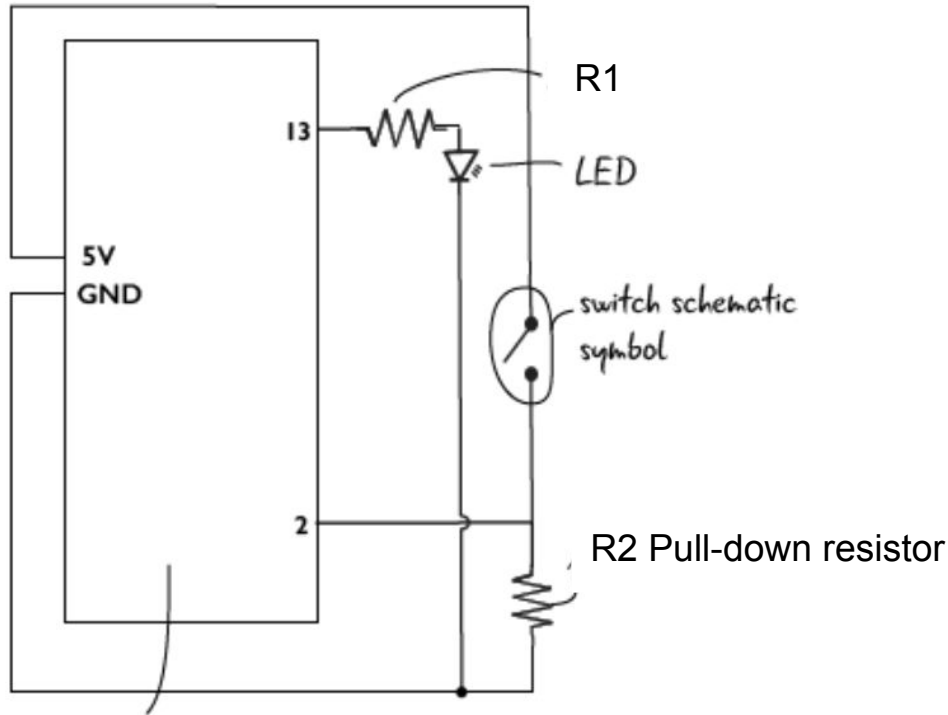
Button press - LED on
Button off - LED off



Button forms a circuit



Let's add Arduino in the picture



Remember, this part of the schematic represents the Arduino

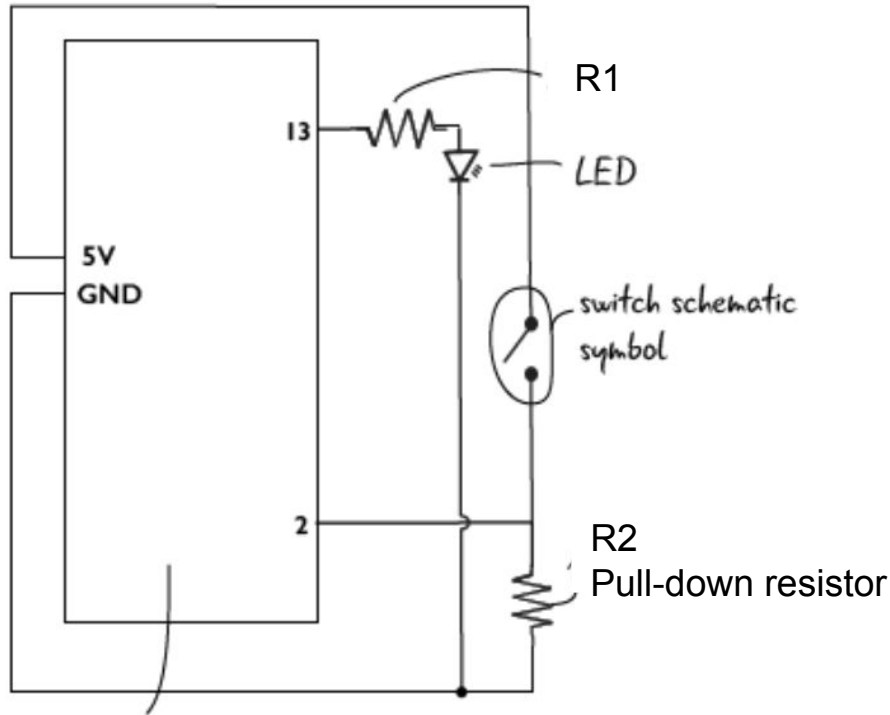
Button Press

↓
Input pin(2) HIGH

↓
Output pin(13) HIGH

↓
LED on

Let's add Arduino in the picture



Remember, this part of the schematic represents the Arduino

Red LED = 2V

Voltage Drop from 5V to 2V

According to Ohm's law,

$$V = IR \text{ or } R = V/I$$

$$\begin{aligned} R_1 &= (V_{\text{init}} - V_{\text{final}})/I_1 \\ &= (5-2)/30 \text{ mA (If } I_1 = 30 \text{ mA or } 30 \times 10^{-3} \text{ A)} \\ &= 100 \text{ Ohms} \end{aligned}$$

If $I_1 = 40 \text{ mA}$ then $R_1 = 75 \text{ Ohm}$

$R_1 = 100 \text{ ohms}$ is the minimum value needed

For the **HIGH** or 1 state a value of 5v will be used, and for **LOW** or 0 a value of 0v will be used.

Nothing is perfect, and the signal is not going to be exactly at 0v or 5v (resistance and noise of the cables for example).

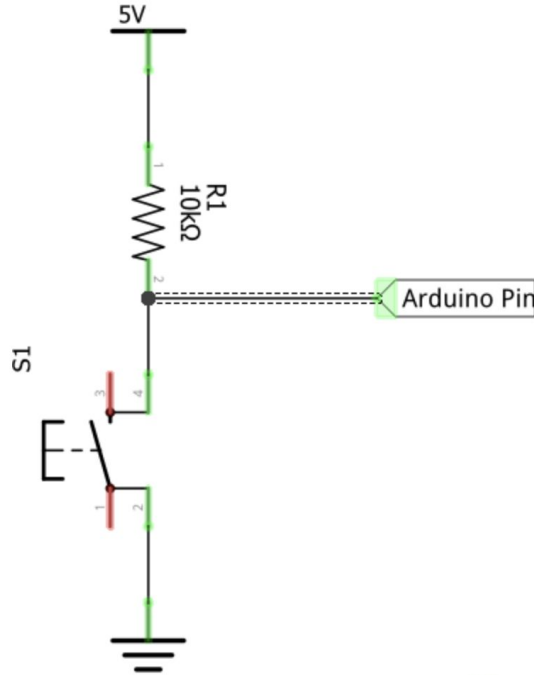
To avoid interpretation errors, the circuits have a **noise margin** that allows them to interpret the signals as HIGH or LOW even though the voltage is not exactly 0v or 5v.

FLOAT, is when a pin is not in a fixed state and fluctuates (for example a loose cable connected to an Arduino pin). This can cause the state of our pin to vary between 0 and 1, and end up giving false positives.

<https://www.electrosoftcloud.com/en/arduino-pull-up-pull-down-resistors/>

pull-up resistor

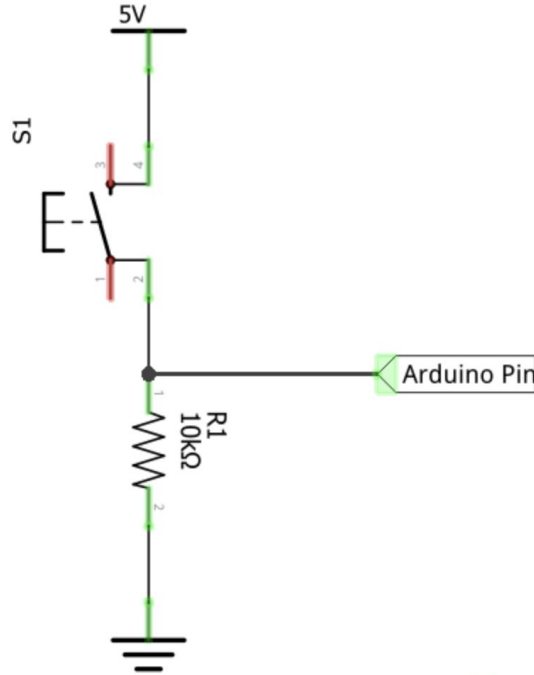
In the case of the pull-up resistor, the Arduino pin is connected to 5v or 3.3v through a resistor.



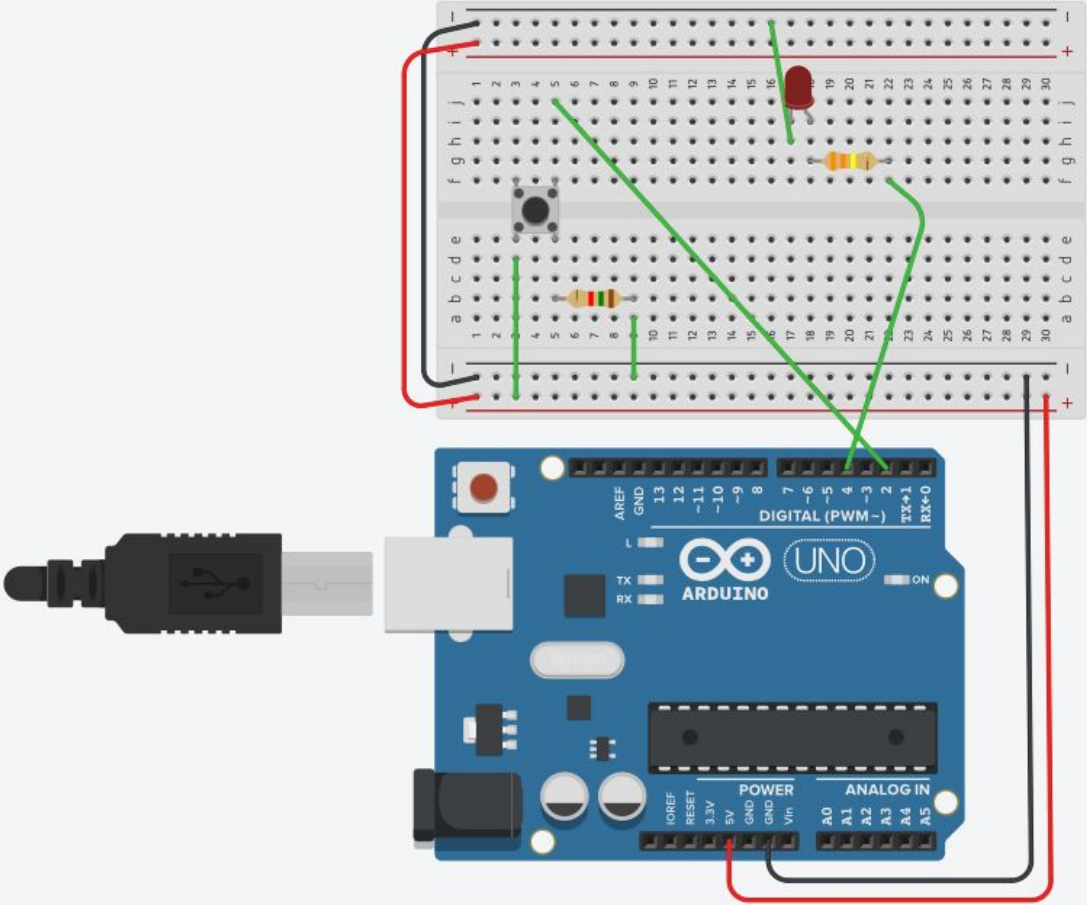
This causes that **by default** it has a **HIGH** value, and when we connect it to 0v by pressing the button, it will have a **LOW** state.

pull-down resistor

In this case this resistance is connected to 0v, so **by default** it will have a **LOW** state.



When the button is pressed, our pin will change from a LOW state to a HIGH state.



https://www.tinkercad.com/things/5WPKkJmQndX-arduino-basic1?sharecode=7OB2X_-QVC9ObNfNbCdc_o9id8hrYfW95yLyrKmCHdA

The simulation has Arduino UNO but you are going to use Arduino ATmega2560