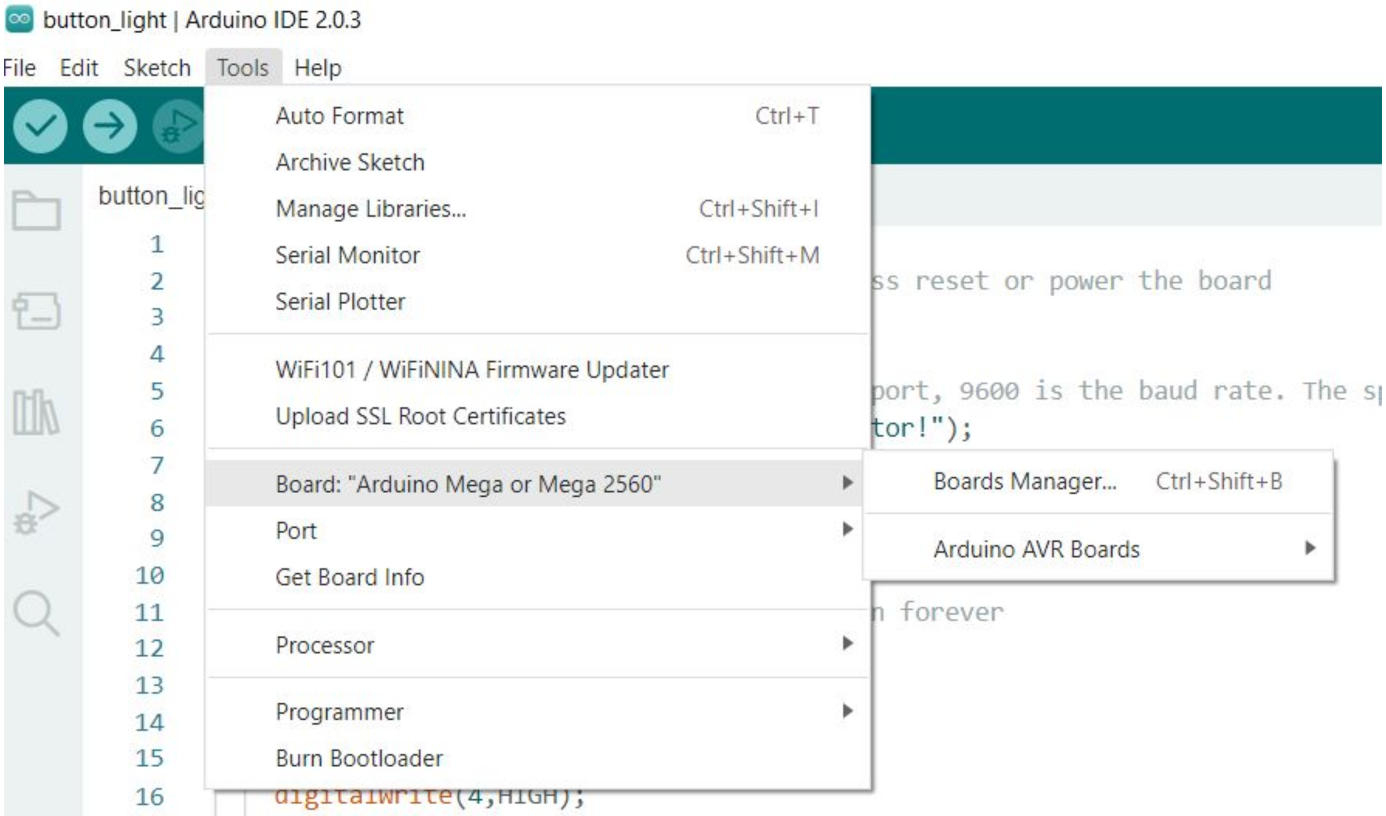
Arduino Coding

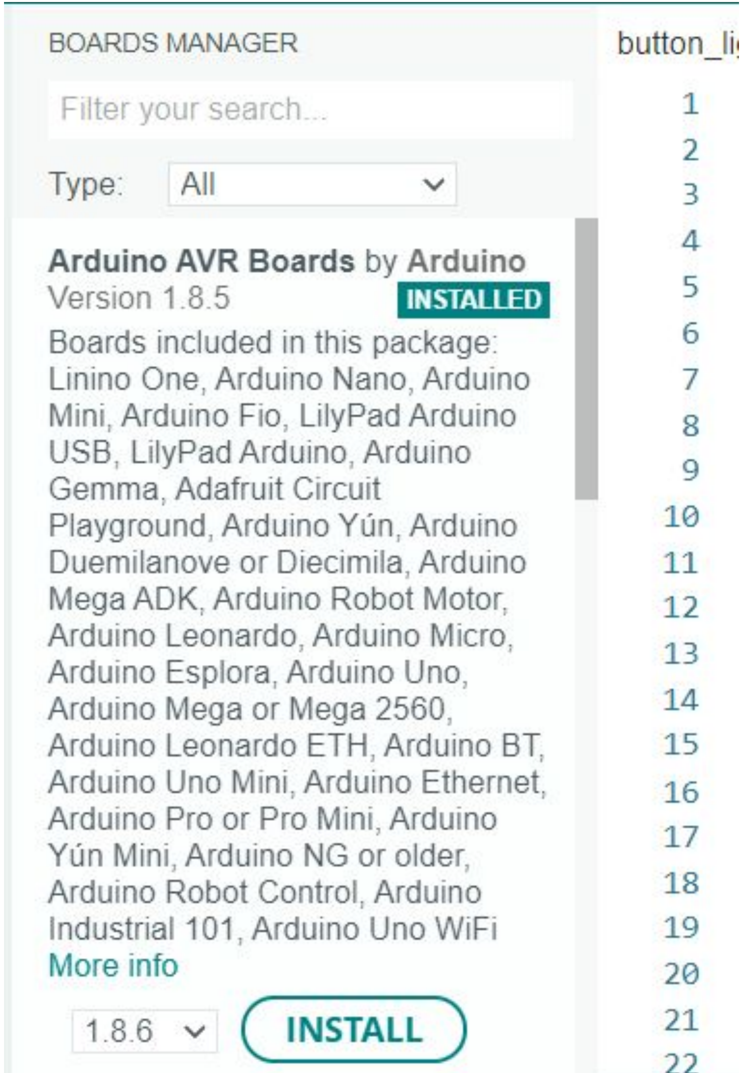
https://www.arduino.cc/en/software

Installing Board driver


button_light | Arduino IDE 2.0.3
File
Edit
Sketch
Tools
Help
button_lig
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
WiFi101 / WiFiNINA Firmware Updater
Upload SSL Root Certificates
Board: "Arduino Mega or Mega 2560"
Port
Get Board Info
Processor
Programmer
Burn Bootloader
Boards Manager... Ctrl+Shift+B
Arduino AVR Boards

```
ss reset or power the board
port, 9600 is the baud rate. The s
tor!");
n forever
digitalWrite(4,HIGH);
```

Install Arduino AVR Boards



Arduino AVR Boards by **Arduino**
Version 1.8.5

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

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

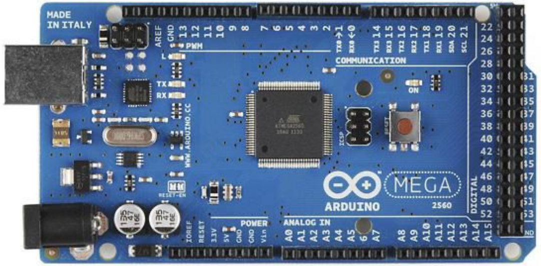
A simple led light that will activate when a button is pressed.

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


}
```

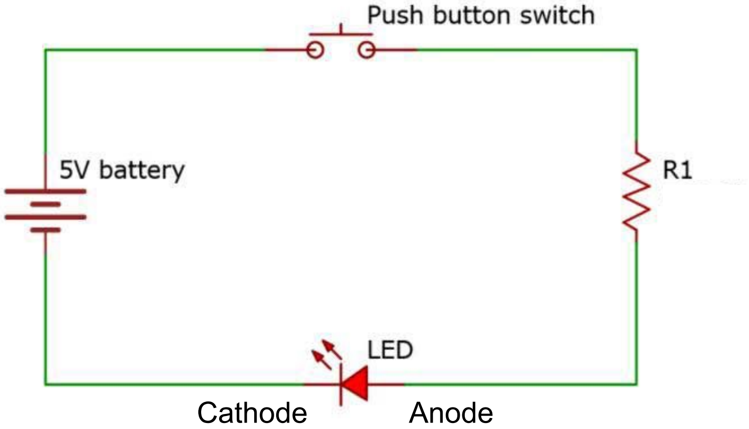
```
// the setup function runs once when you press reset or power the board
void setup() {
  Serial.begin(9600); // sets up the serial port, 9600 is the baud rate. The speed at which a signal can travel
  Serial.println("Hello from the Serial monitor!");
  pinMode(4, OUTPUT); // Output is the macro
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



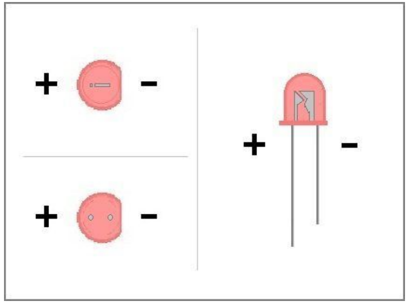
```
// the loop function runs over and over again forever
void loop() {

  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);
  }
}
```

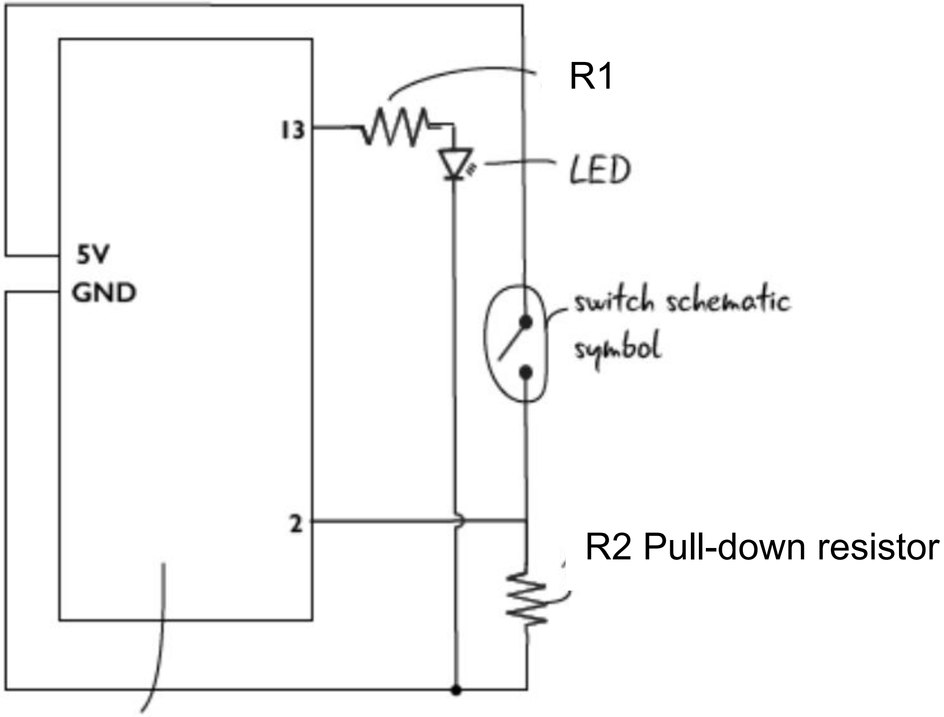
Button press - LED on
Button off - LED off



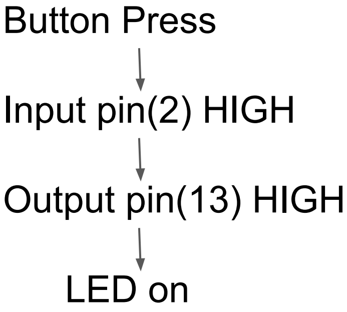
Button forms a circuit



Let’s add Arduino in the picture



Remember, this part of the schematic represents the Arduino



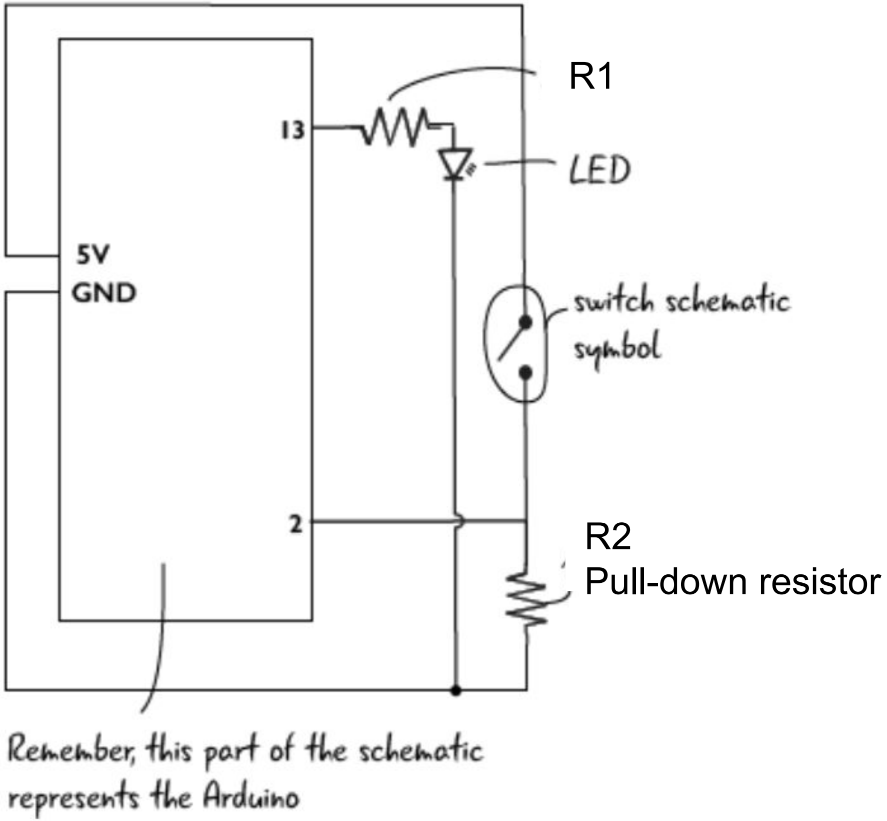
Button Press

Input pin(2) HIGH

Output pin(13) HIGH

LED on

Let’s add Arduino in the picture



Red LED = 2V

Voltage Drop from 5V to 2V

According to Ohm’s law,

V = IR or R = V/I

R_1 = (Vinit - Vfinal)/I_1

= (5-2)/30 mA (If I1 = 30 mA or $30X10^{-3}$A)

= 100 Ohms

If I_1=40 mA then R_1 = 75 Ohm

R_1 = 100 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).

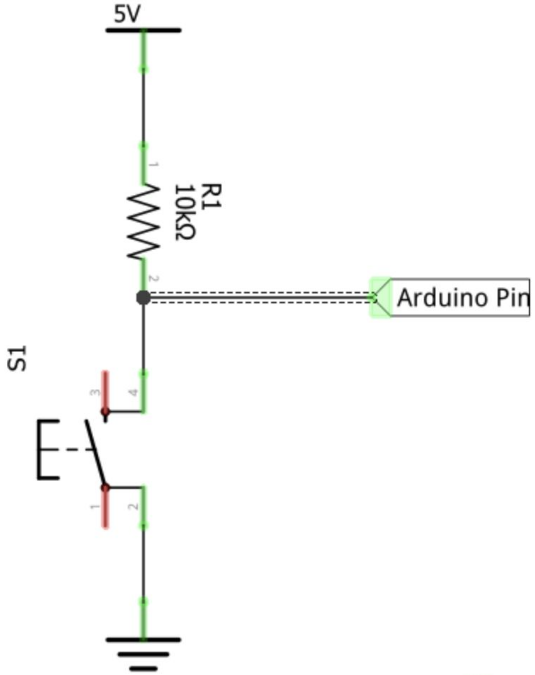
To avoid interpretation errors, the circuits have a **noise margin** that allows them to interpret the signals as HIGH or LOW even though the voltage is not exactly 0v or 5v.

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

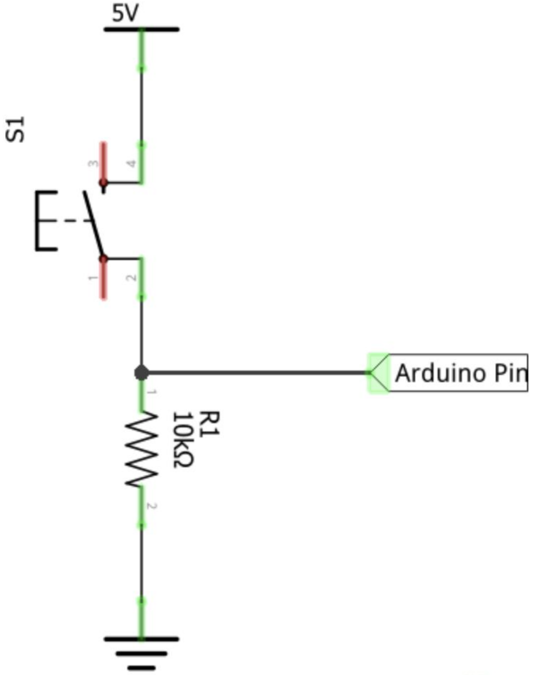
In the case of the pull-up resistor, the Arduino pin is connected to 5v or 3.3v through a resistor.



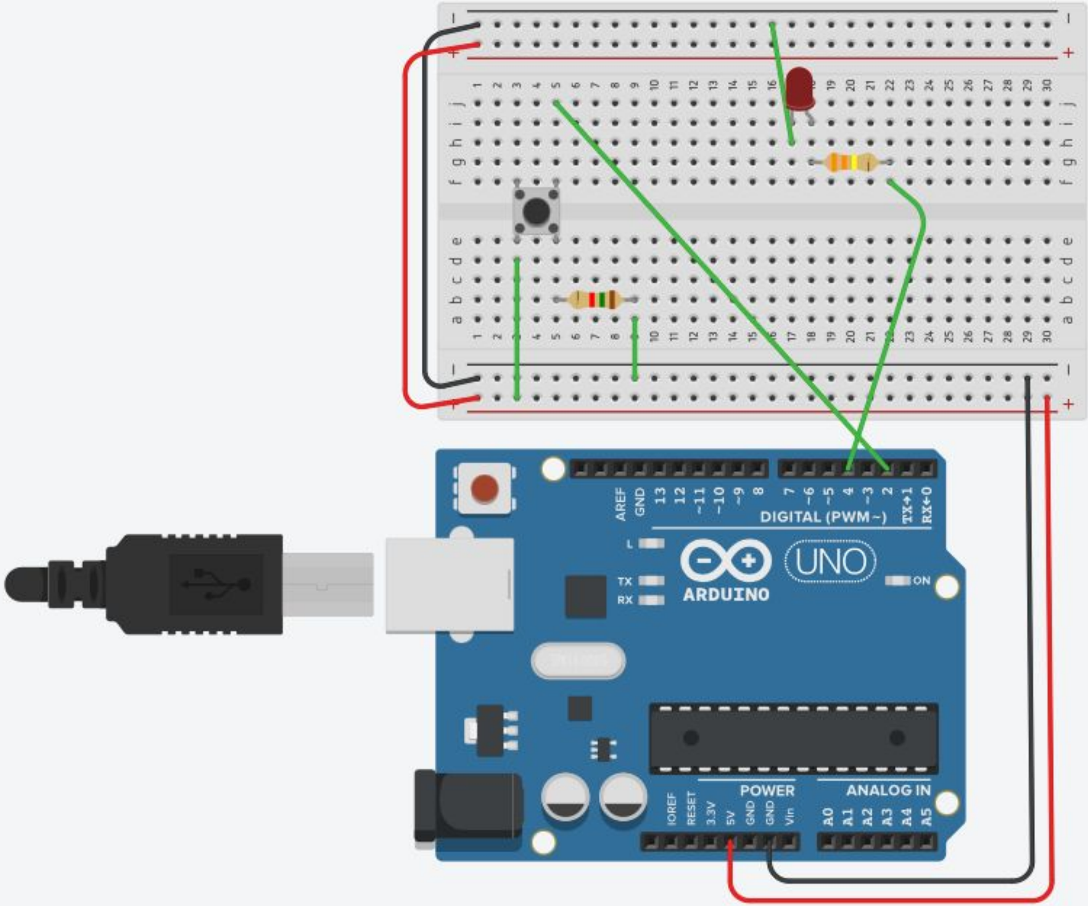
This causes that **by default** it has a **HIGH** value, and when we connect it to 0v by pressing the button, it will have a **LOW** state.

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.


AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5

https://www.tinkercad.com/things/5WPKkJmQndX-arduinobasic1?sharecode=7OB2X_-QVC9ObNfNbCdc_o9id8hrYfW95yLyrKmCHdA

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