



NYU

**TANDON SCHOOL
OF ENGINEERING**



Promoting robotic design and entrepreneurship
experiences among students and teachers

Lesson 14: Arduino Session – Digital/Analog Input

Innovative Technology Experiences for Students and Teachers (ITEST), Professional Development Program, July 2017-19

Mechatronics, Controls, and Robotics Laboratory, Department of Mechanical and Aerospace Engineering, NYU Tandon School of Engineering

CONTENTS

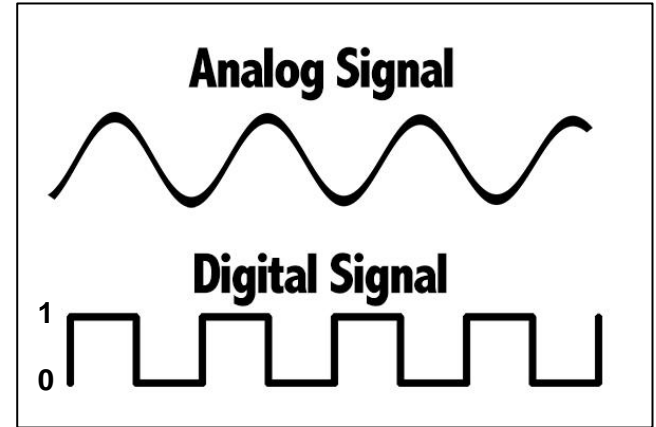


- Analog and digital signal
- Pull-up and pull-down resistance
- Reading a pushbutton
- Joystick

- **TASK/ACTIVITY:**
 - Buttons
 - Joystick

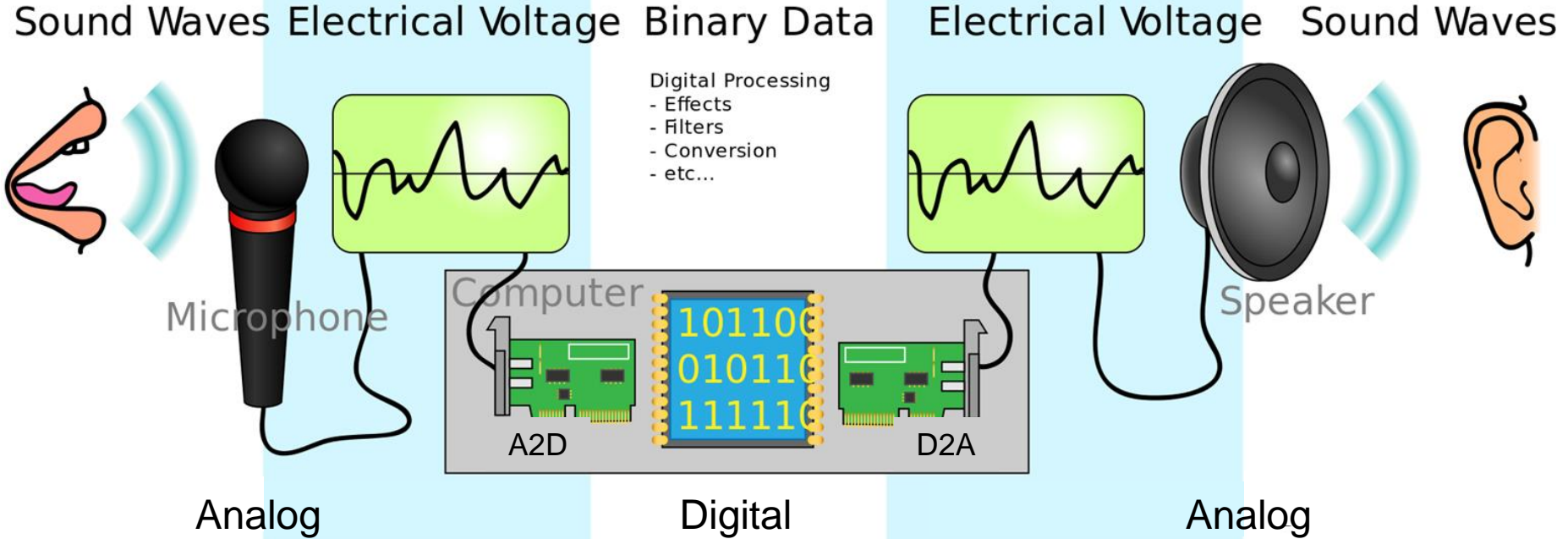
ANALOG AND DIGITAL SIGNAL

- **Analog signals:** Continuous signal varying in amplitude and frequency characteristics
- **Digital signals:** A pulse, either on or off, that conveys information in binary form (1's and 0's)



[Source](#)

ANALOG AND DIGITAL SIGNAL



[Source](#)



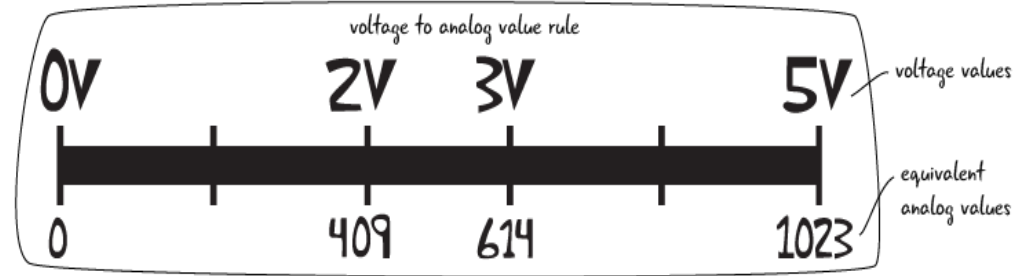
Digital input

In digital inputs,
there are only 2 possible states

HIGH = On = 1 = true

LOW = Off = 0 = false

Analog input



[Source](#)

Name of function	What it does	Parameters it requires	Range of values
digitalRead()	Reads the value of digital input pin	The number of the pin it is assigned to read	Read either 1 or 0 from pin
analogRead()	Reads the value of analog input pin	The number of the pin it is assigned to read	Reads an integer between 0 and 1023 from pin



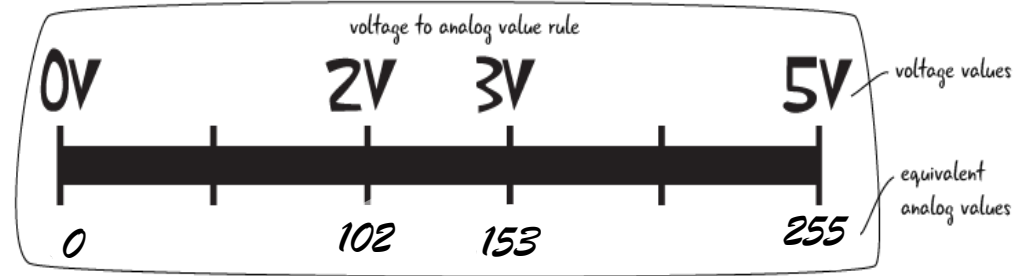
Digital output

In digital outputs, there are only 2 possible states

HIGH = On = 1 = true

LOW = Off = 0 = false

Analog output



Name of function	What it does	Parameters it requires	Range of values
digitalWrite()	Writes the value to a digital pin	The number of the pin it is assigned to write	Write either 1 or 0 to a pin
analogWrite()	Writes a PWM signal to a specified output pin	The number of the pin it is assigned to write (Analog and PWM)	Writes an integer between 0 and 255 to a pin



MAP ANALOG INPUT TO ANALOG OUTPUTS

Analog Input – 0 to 1023

BUT

Analog Output – 0 to 255

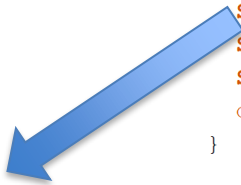
```

//Map an analog value to 8 bits (0 to 255)
void setup() {
  Serial.begin(9600);
}

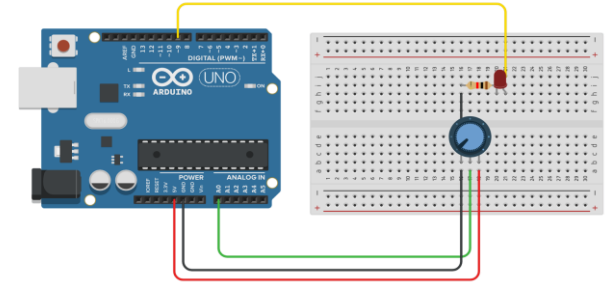
void loop()
{
  int sensorValue = analogRead(A0);
  int outputValue = map(sensorValue, 0, 1023, 0, 255);
  analogWrite(9, outputValue);

  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);
  delay(500);
}

```



$$\text{voltage} = \text{sensorValue} * (5.0 / 1023.0);$$



COM8

```

sensor = 11      output = 2
sensor = 8       output = 1
sensor = 2       output = 0
sensor = 1       output = 0
sensor = 1       output = 0
sensor = 2       output = 0

```

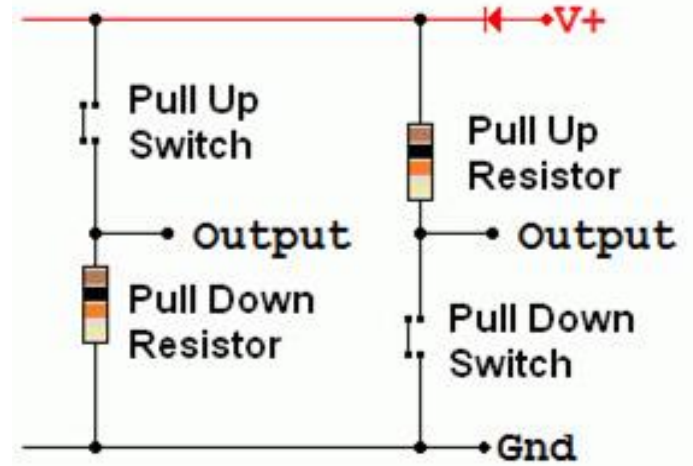
Autoscroll Show timestamp

[Program](#)

[Video](#)

PULL-UP AND PULL-DOWN RESISTANCE

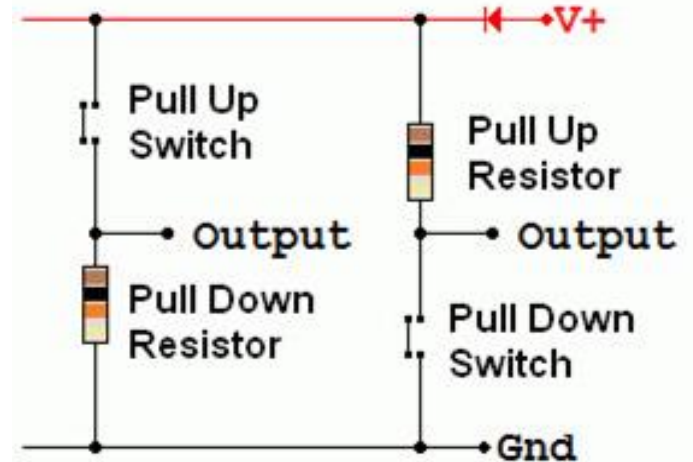
- Pull-up or pull-down resistors are used in electronic logic circuits to ensure that inputs to the digital device settle at expected logic levels if external components are disconnected
- If a pin configured as input with `pinMode(pin, INPUT)` is not given a definite state (left floating), then it will experience random changes in pin state, pick up electrical noises



Source

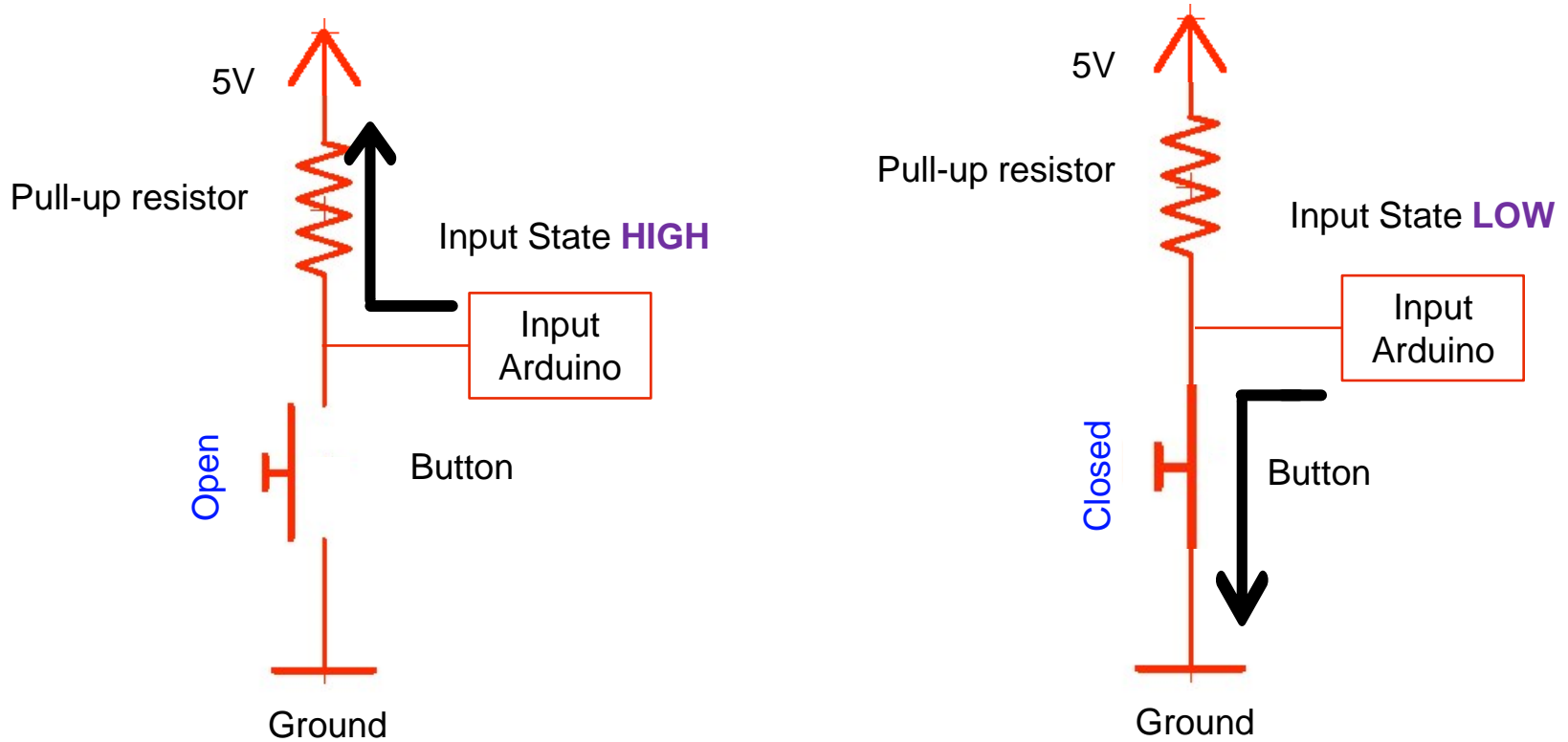
PULL-UP AND PULL-DOWN RESISTANCE

- To prevent the floating condition, a pull-up (or pull-down) resistor connects the pin to 5V (or Gnd)
- Just because you have nothing at all connected to an input pin doesn't mean it is a logical zero



Source

PULL-UP RESISTANCE



READING A PUSH-BUTTON

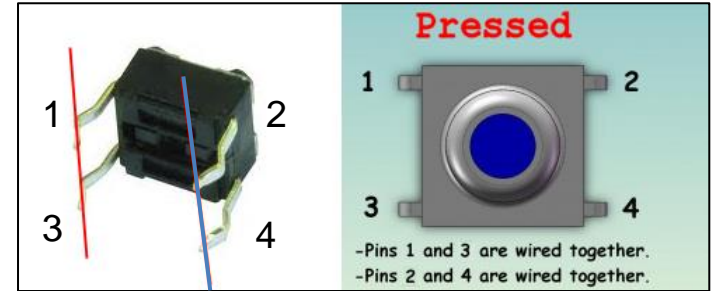
Push-button:

- Connects two points in a circuit when pressed
- Pressed: the connection between its two legs, connecting the pin to the ground, read a LOW (in pull-up circuit)

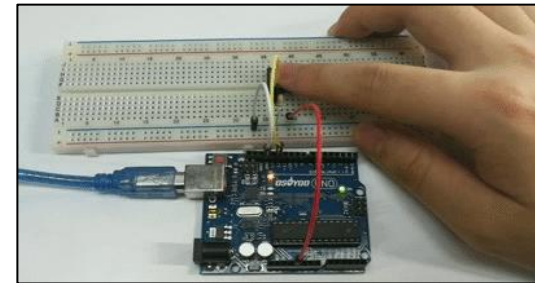


A closed switch:
electricity **can** flow
through the circuit

An open switch:
electricity **cannot** flow
through the circuit



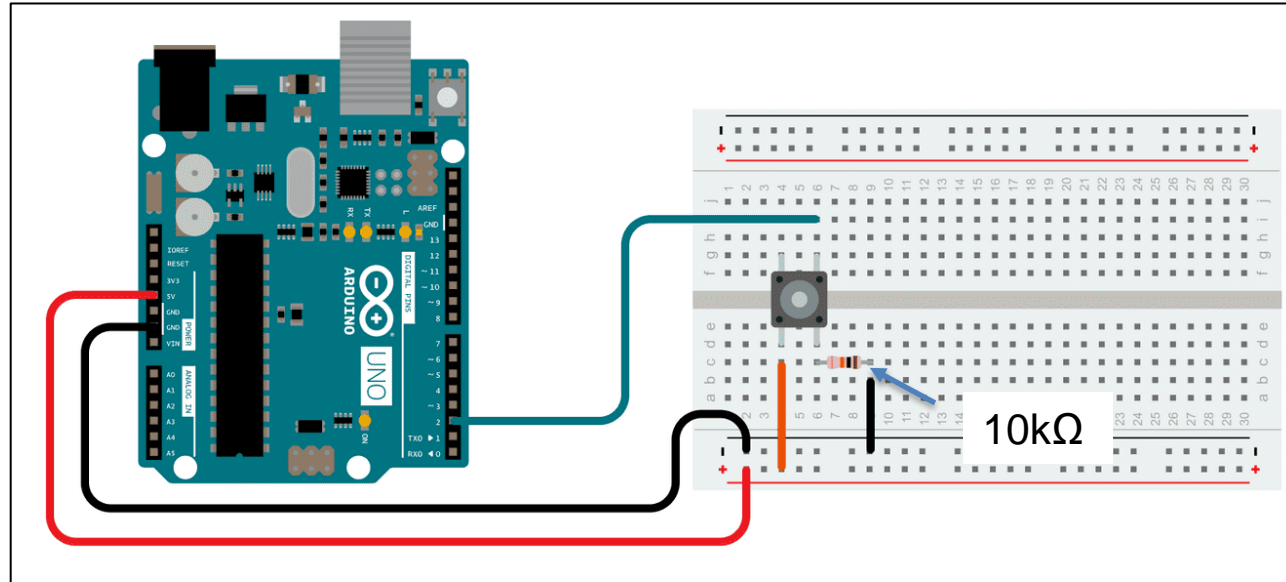
[Source](#)



[Source](#)

ACTIVITY - 1

Is this a pull-up or a pull-down circuit?

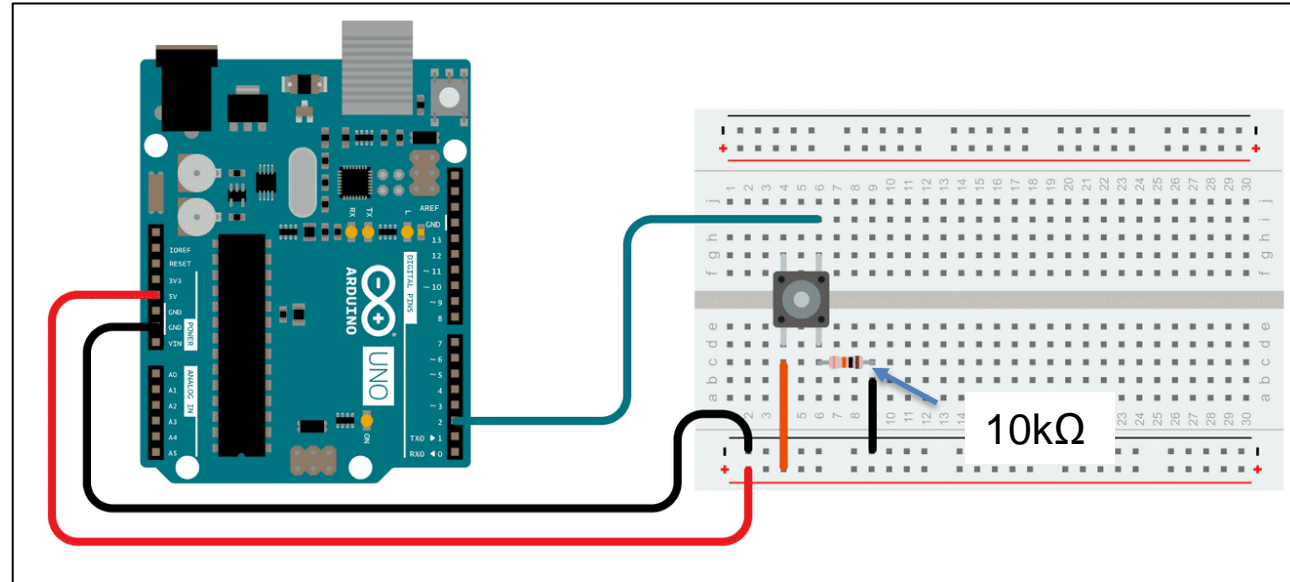


[Source](#)

ACTIVITY -1 SOLUTION

Is this a pull-up or a pull-down circuit?

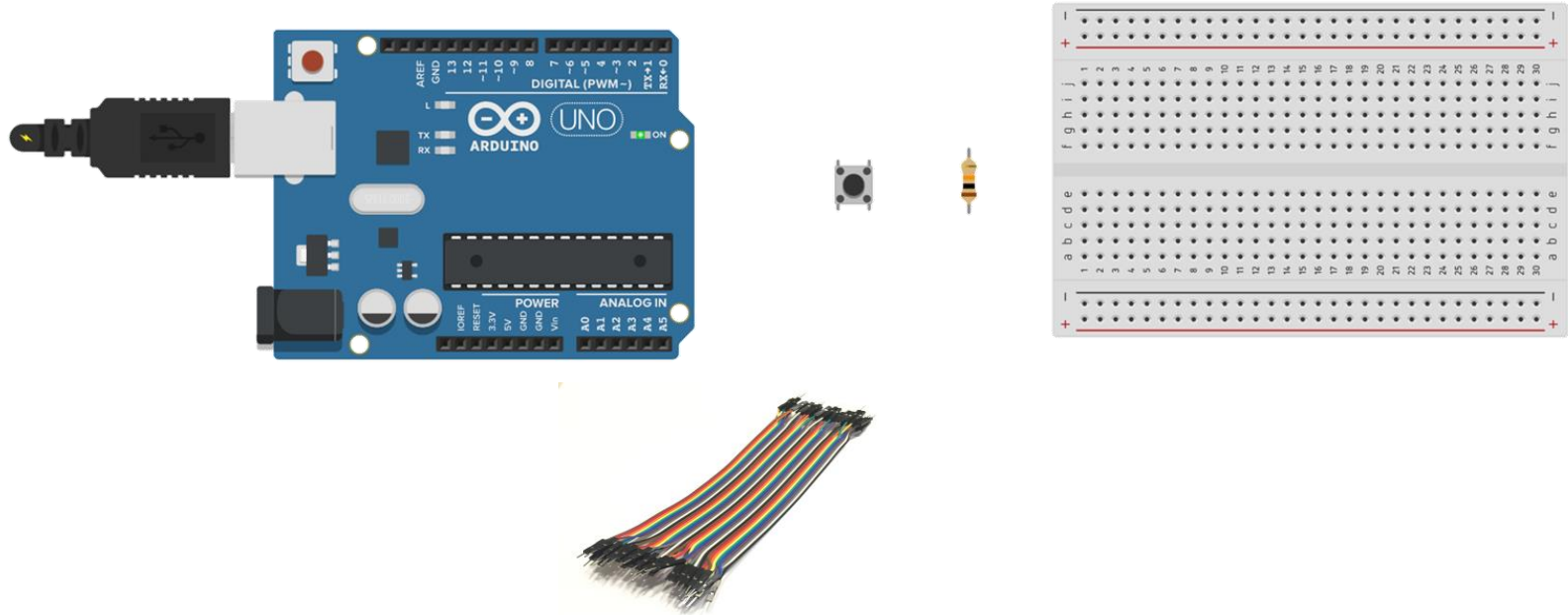
It is a **pull-down** circuit



[Source](#)

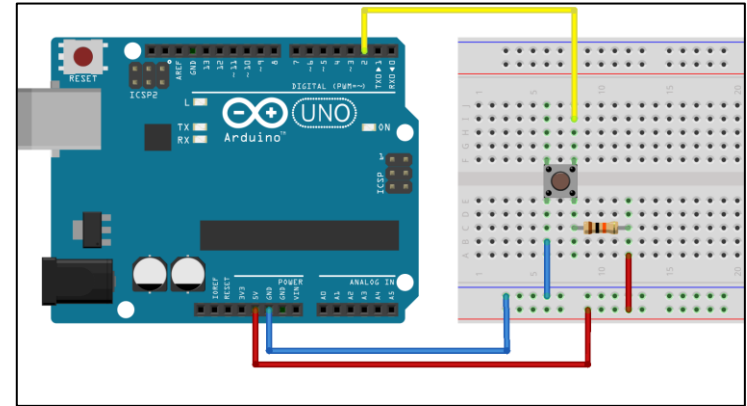
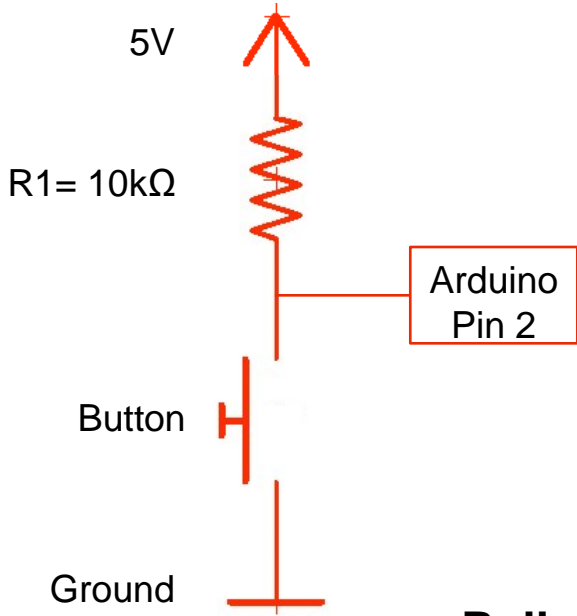
ACTIVITY - 2

- Create a pull-up circuit using these components and write a program to turn on the internal LED on P13 when the button is pressed



ACTIVITY - 2 SOLUTION

- Create a pull-up circuit using these components and write a program to turn on an internal LED when the button is pressed



Source

Pull-up Circuit Diagram



HOW TO READ A PUSHBUTTON (CODE)

```
int LEDpin = 13;
int ButtonPin = 2;
//Initialise LED pin and Button pin

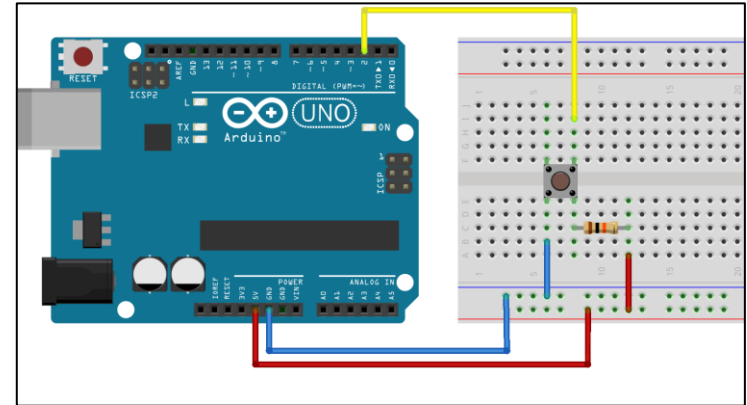
void setup() {
  pinMode(LEDpin, OUTPUT); //setup LED as output
  pinMode(ButtonPin, INPUT); //setup Button input
}

void loop() {
  int buttonValue = digitalRead(ButtonPin); //read value of button

  if (buttonValue == HIGH) {
    digitalWrite(LEDpin, LOW); //Turn LED OFF
  }

  else {
    digitalWrite(LEDpin, HIGH); //Turn LED ON
  }
}
```

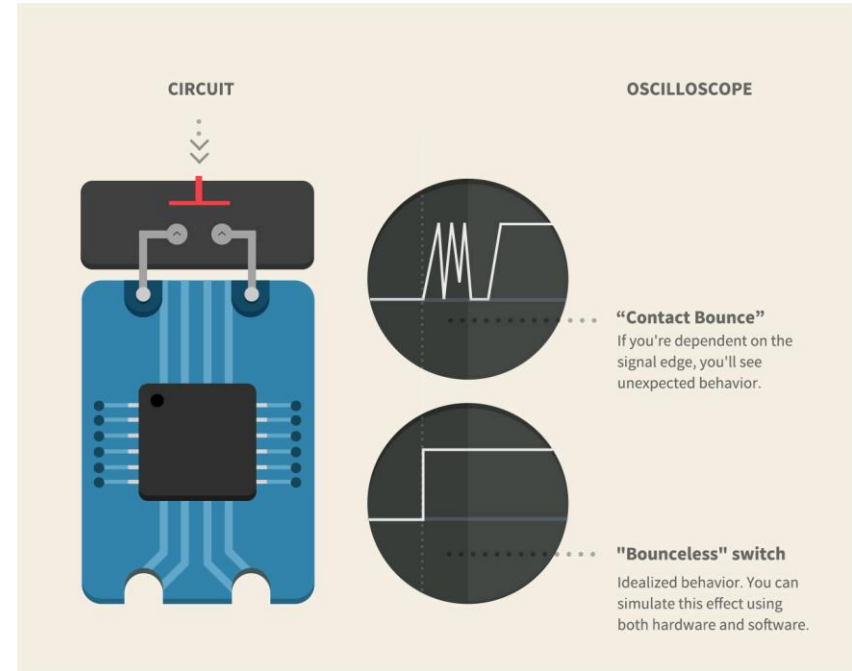
[Program](#)



[Source](#)

DEBOUNCING IN A PUSHBUTTON

- Mechanical and physical issues may cause spurious open/close transitions in the push button.
- This causes the Arduino to believe even a single press as multiple presses



[Source](#)



NYU DEBOUNCING IN A PUSHBUTTON (CODE)

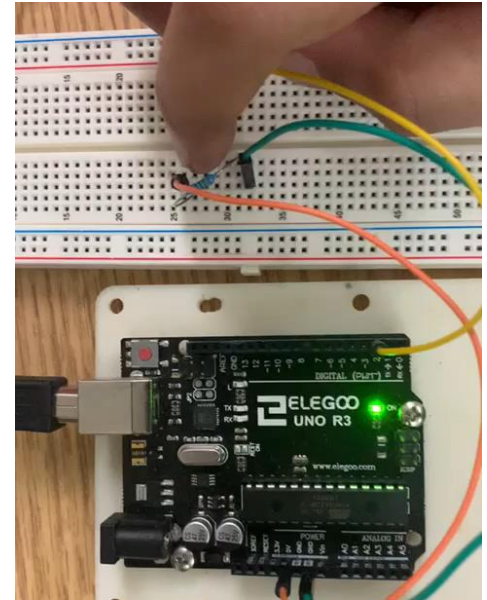
Debouncing:

- So, a single press doesn't appear as multiple presses
- Adding a delay of 100ms at the input read avoids the issue

```
int LEDpin = 13;
int ButtonPin = 2;
//Initialise LED pin and Button pin
void setup() {
  pinMode(LEDpin, OUTPUT);
  pinMode(ButtonPin, INPUT);
  //setup output and input types
}

void loop() {
  int buttonValue = digitalRead(ButtonPin);
  //read value of button
  delay(100);
  //delay to avoid debouncing
  digitalWrite(LEDpin, buttonValue);
  //light LED when button is pressed
}
```

[Program](#)

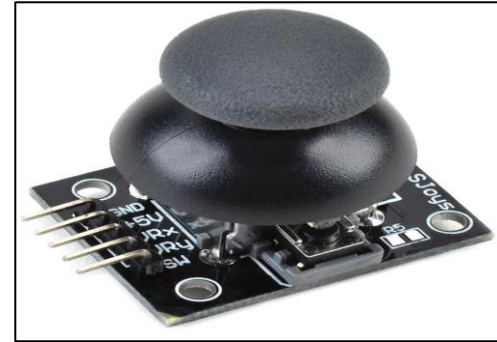


[Video](#)

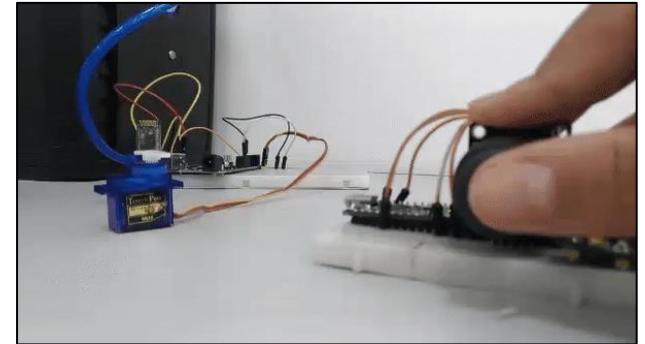
JOYSTICK



[Source](#)



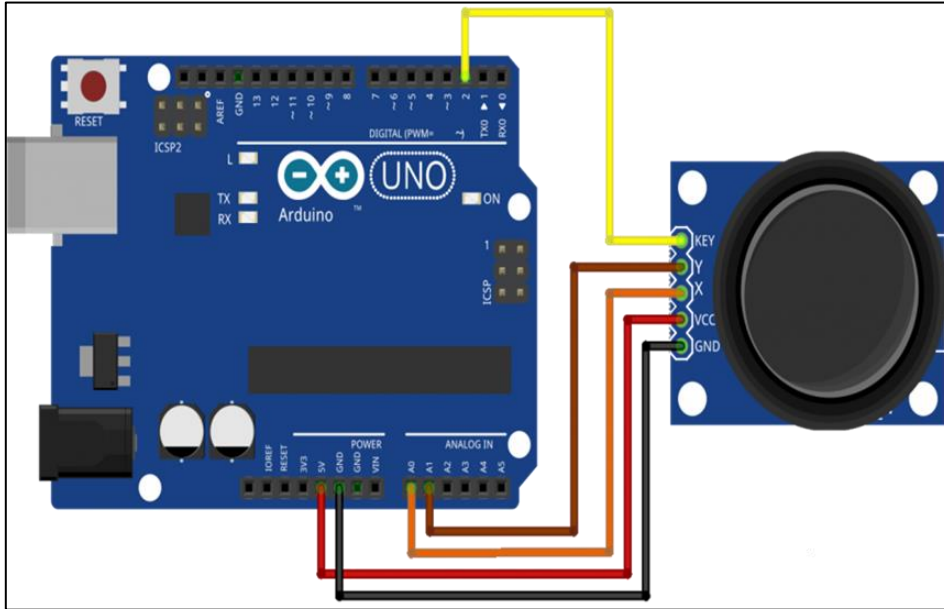
[Source](#)



[Source](#)

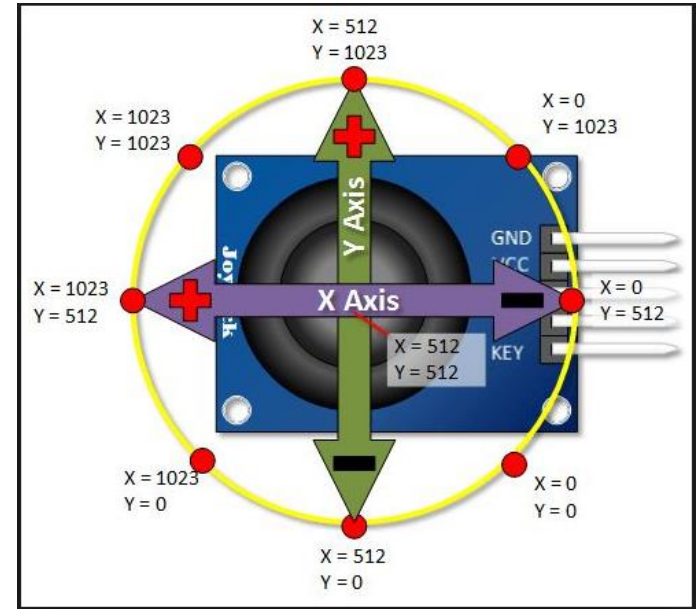
ACTIVITY - 3

- Read X and Y values from the Joystick



[Source](#)

Circuit Diagram



[Source](#)

ACTIVITY – 3 SOLUTION

```

const int SW_pin = 2; // digital pin connected to switch output
const int X_pin = A0; // analog pin connected to X output
const int Y_pin = A1; // analog pin connected to Y output
void setup() {
  pinMode(SW_pin, INPUT);
  digitalWrite(SW_pin, HIGH);
  Serial.begin(9600);
}
void loop() {
  Serial.print("Switch: ");
  Serial.print(digitalRead(SW_pin));
  Serial.print("\n");
  Serial.print("X-axis: ");
  Serial.print(analogRead(X_pin));
  Serial.print("\n");
  Serial.print("Y-axis: ");
  Serial.println(analogRead(Y_pin));
  Serial.print("\n\n");
  delay(500);
}

```

[Program](#)

COM8

Y-axis: 521

Switch: 1

X-axis: 508

Y-axis: 521

Switch: 1

X-axis: 507

Y-axis: 521

Serial

[Video](#)



read value from the potentiometer attached to pin A0

sensorValue = analogRead(analogInPin);

Task 1: Use all the sensors taught in the lecture

Task 2: Sense the digital input from the push button switch, and turn on-off the LED using the switch (using pullup and pulldown circuit)

Task 3: Using the joystick, move the arm up and down, and when you press the joystick, the arm should open

Task 4: Use a potentiometer and change the brightness of the LED

Task 5 (intermediate level): Use a color sensor and show the color shown to it on the serial monitor

Task 6 (advanced level): Build a counter using a push button and show the output on the serial monitor (the concept to be used is debouncing)



NYU

**TANDON SCHOOL
OF ENGINEERING**



Thank You!

Questions and Feedback?

Innovative Technology Experiences for Students and Teachers (ITEST), Professional Development Program, July 2017-19

Mechatronics, Controls, and Robotics Laboratory, Department of Mechanical and Aerospace Engineering, NYU Tandon School of Engineering