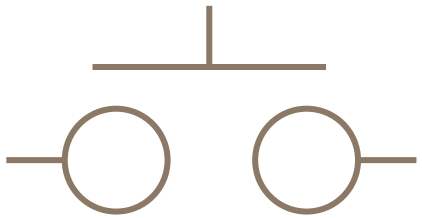
INDUSTRIAL UNIVERSITY OF HOCHIMINH CITY

Modern IoT Technology

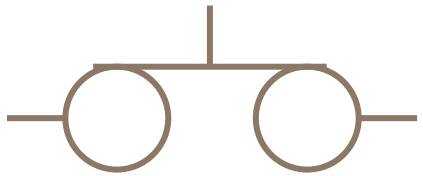
Arduino Sensors

Pull-Up Resistors

- A pin that's not connected to anything
- *floating pin*
  - *digitalRead(),*
  - *Program reads the state of the pin, will it be high (pulled to VCC) or low (pulled to ground)*
  - *Unpredictable answer.*



Normally Open
(NO)

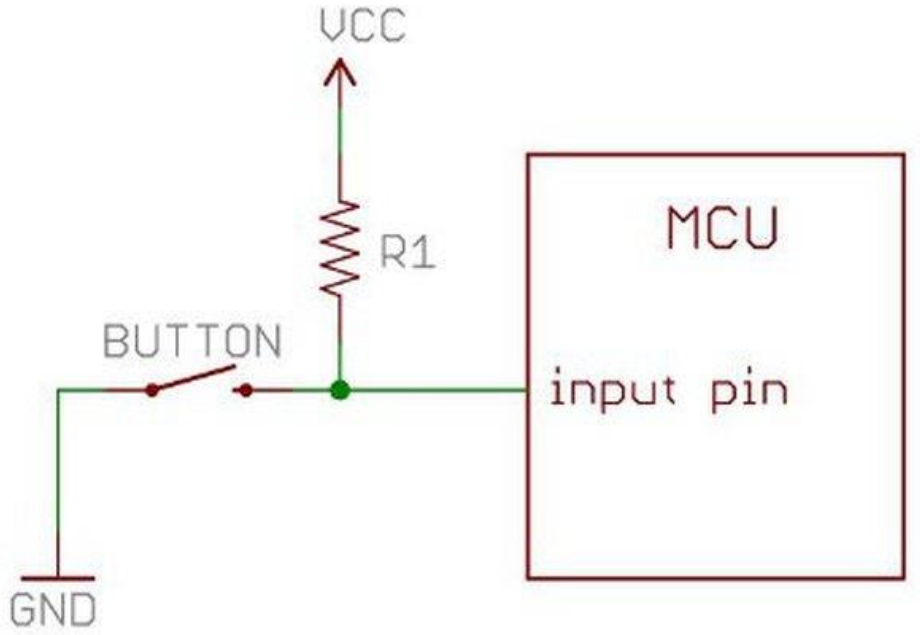


Open Switch
(At Rest)

Closed Switch
(Depressed)

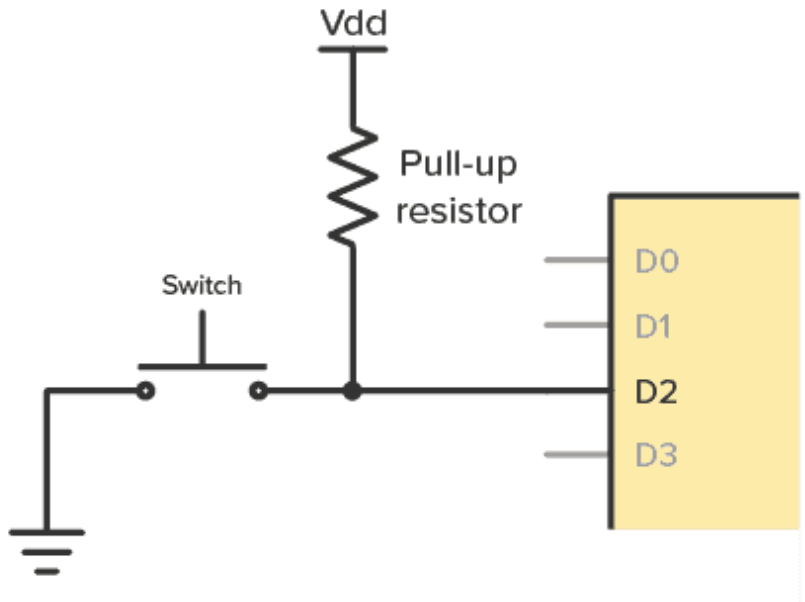
Pull-Up Resistors

- Pull-up resistor is connected to the high voltage (this is usually 3.3V or 5V and is often refereed to as VCC)
- *Pull-up resistor in the Arduino circuitry*



Pull-Up Resistors

- When you press the button?
  - D2 and GND are now connected
  - D2 goes LOW (zero volts, ground)
- Release the button (button up)
  - Pull-up resistor pulls it up to HIGH
  - D2 is only connected to +5 V

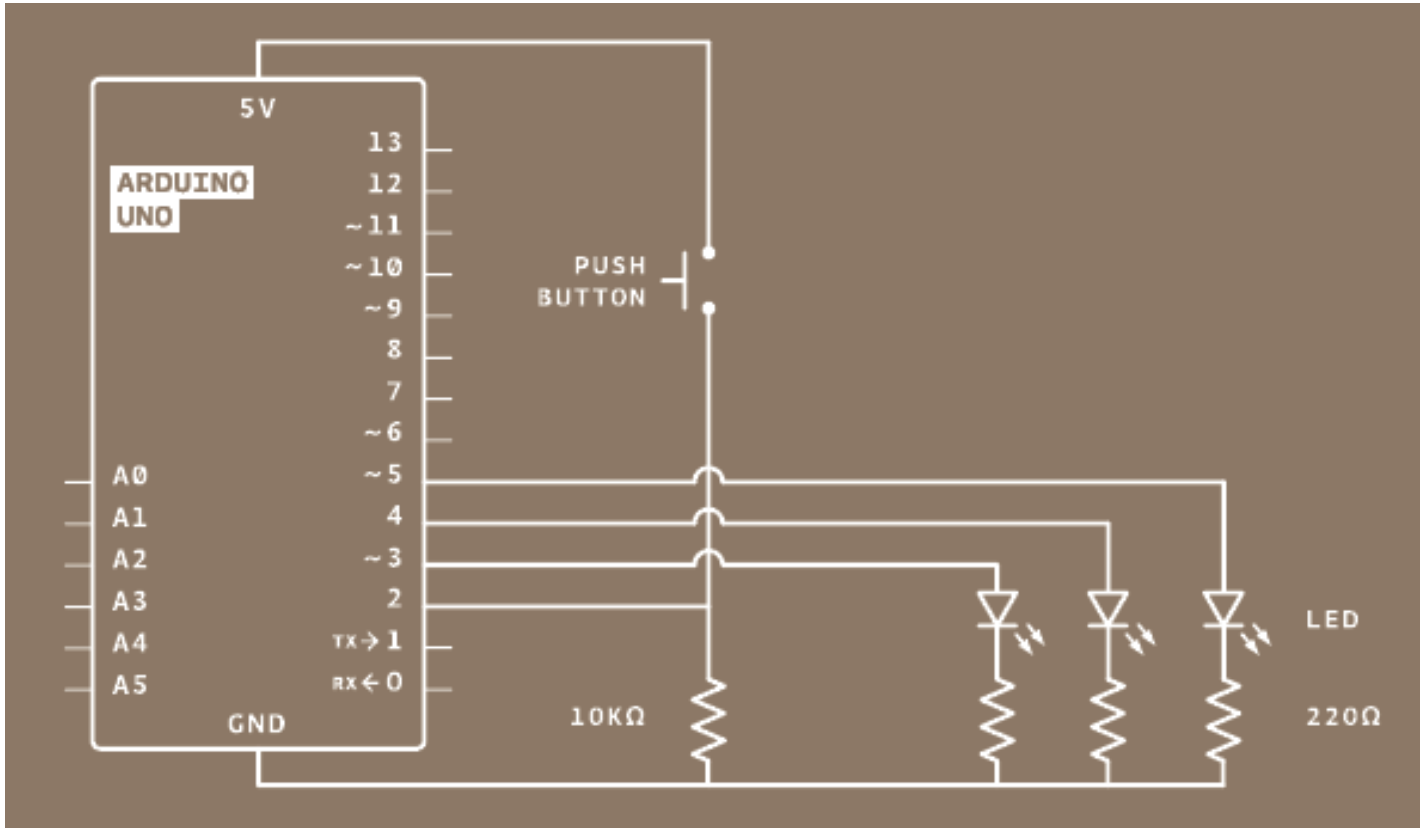


- Value are tens of thousands of ohms to a few million ohms
- e.g., 20 kΩ to 2 MΩ.

Pull-Down Resistors

- A pull-down to hold the pin LOW
- HIGH indicates a button press
- LOW indicates that the button is open

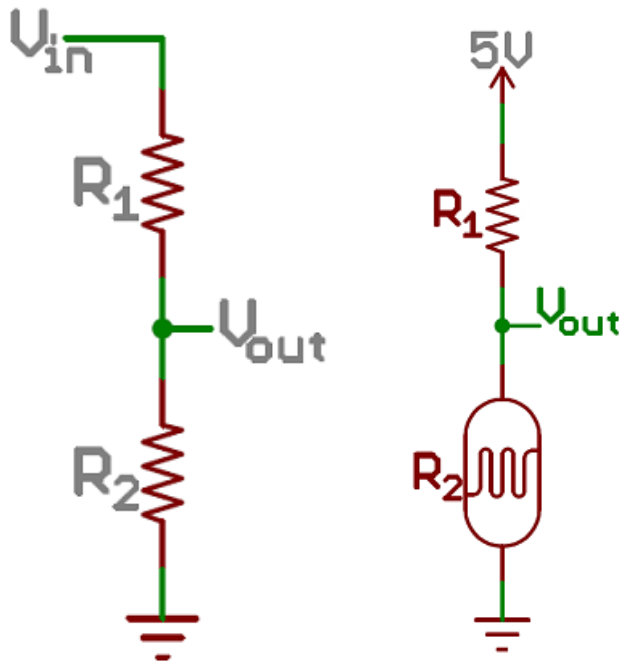
Pull-Down Resistors



A voltage divider

- A voltage divider is used to scale down a large voltage into a smaller one
- In the real world, deal with much larger voltages at times of 5V
- Arduino, maximum of 5 V of analog voltage

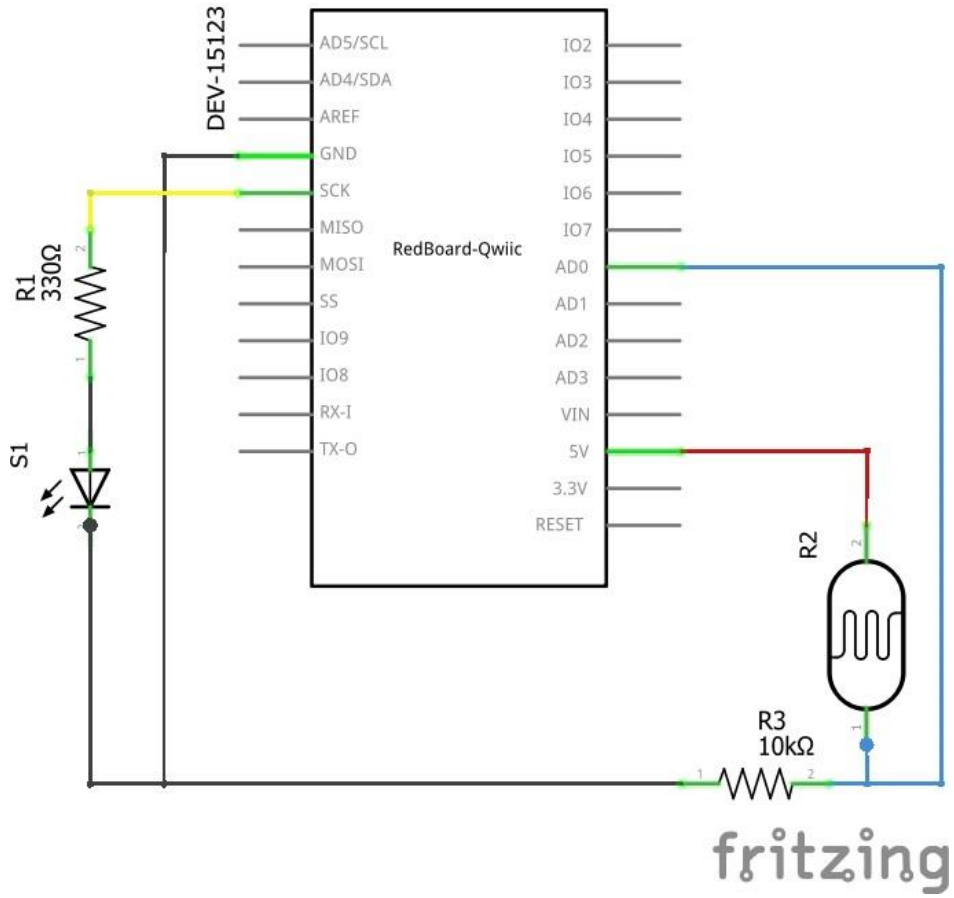
$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

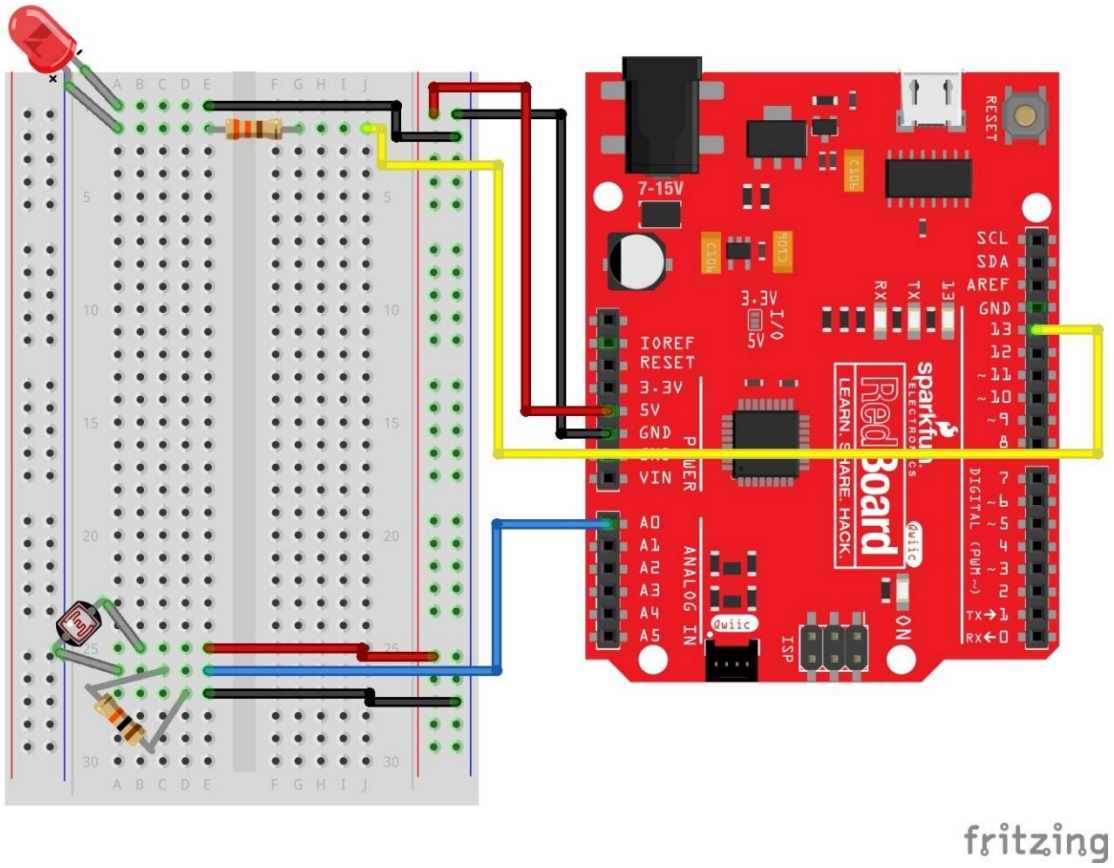


A voltage divider

- Analog to Digital Converter (or ADC)
- The six analog inputs (A0--A5)
- Create a digital signal based on resolution is 10-bit, get 1024 possible values.
- A photo-resistor or Light Dependent Resistor (LDR) is a component that is sensitive to light

A voltage divider





A voltage divider

```
int photoresistor = 0;              //hold a value
based on the brightness of the ambient light
int threshold = 750;    //if the photoresistor
reading is below this value the the light will
turn on
```

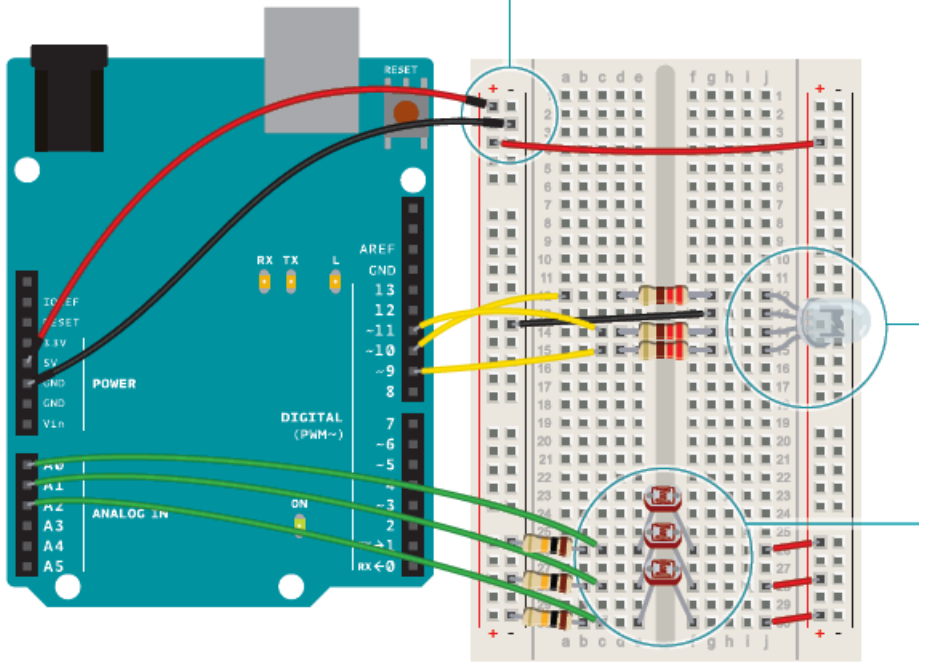
```
void setup()
{
  Serial.begin(9600);      //start a serial
connection with the computer

  pinMode(13, OUTPUT);    //set pin 13 as an
output that can be      set to HIGH or LOW
}
```

```
void loop(){
  //read the brightness of the ambient light
  photoresistor = analogRead(A0);   //between 0 and 1023
based on how bright the ambient light is
  Serial.println(photoresistor);
  //if the photoresistor value is below the threshold turn the
light on, otherwise turn it off
  if (photoresistor < threshold) {
    digitalWrite(13, HIGH);        // Turn on the LED
  } else {
    digitalWrite(13, LOW);         // Turn off the LED
  }
  delay(100);
}
```

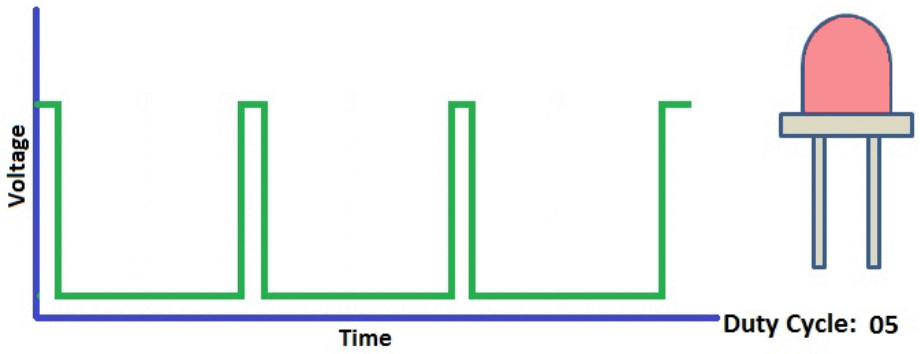
Exercise A voltage divider

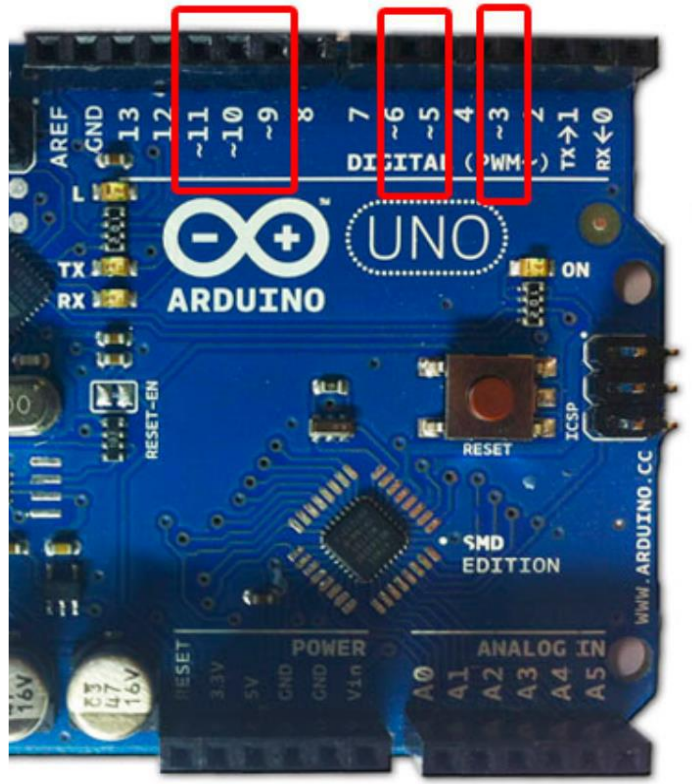
Print out the sensor values on one line



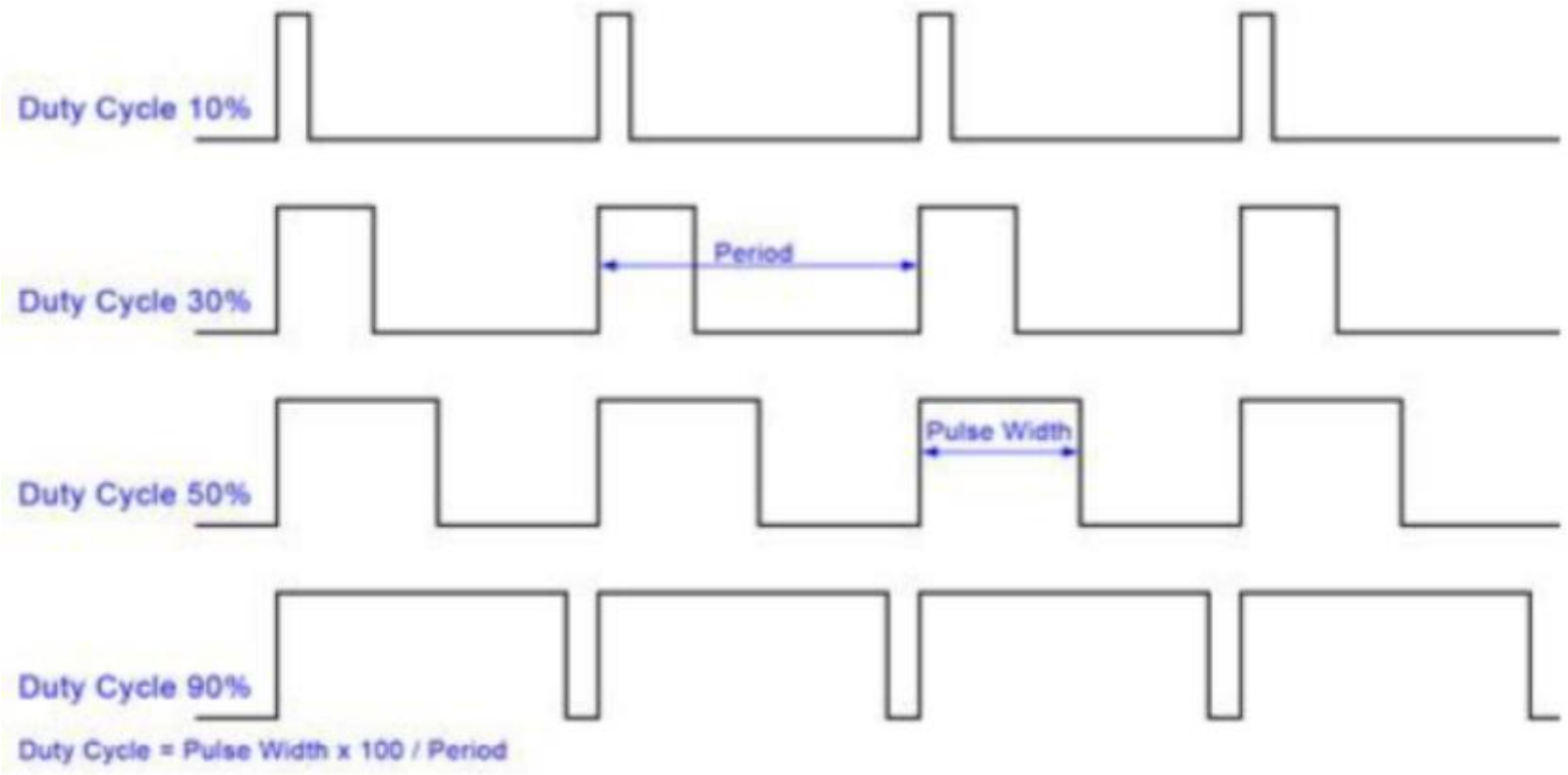
Pulse Width Modulation (PWM)

- Converts a digital value to analog output
- Allows Arduino to generate a **series of pulses**
- When A pin is **output high** the apparent voltage at that pin will be **close to 5 V**
- When the pin is made **output low** it is close to **0 V**





Pulse Width Modulation



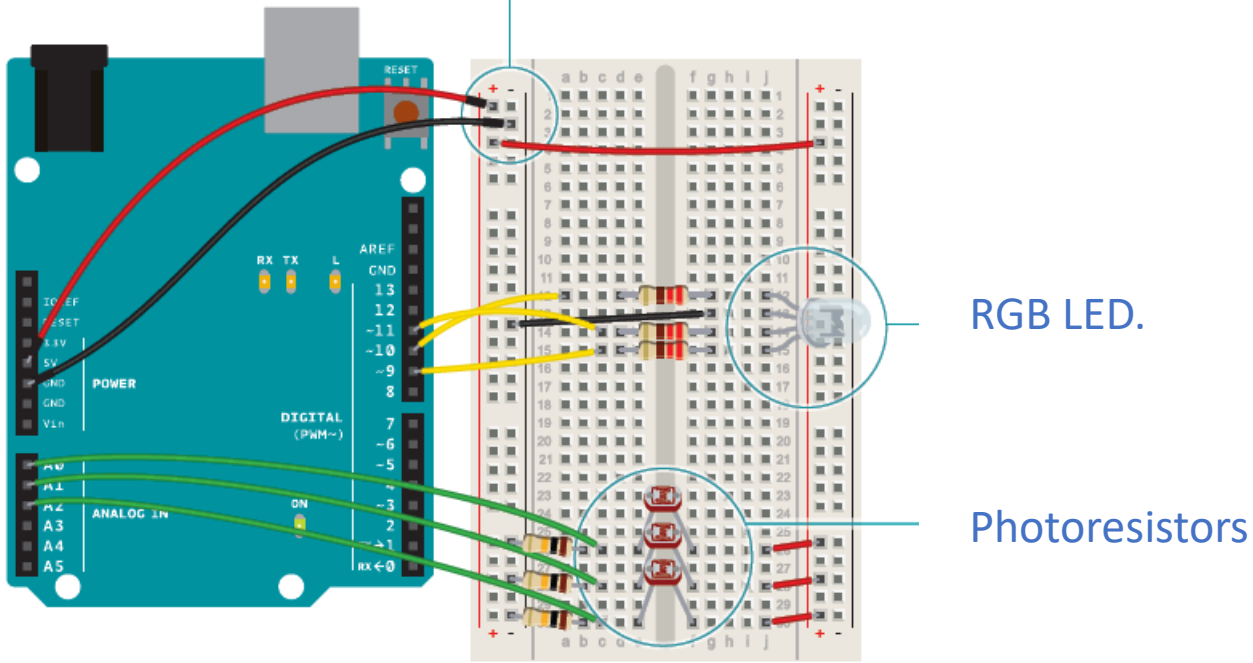
Exercise PWM

- Attach one LED to your Arduino and write a program to change its brightness
  - The highest
  - Medium
  - The lowest

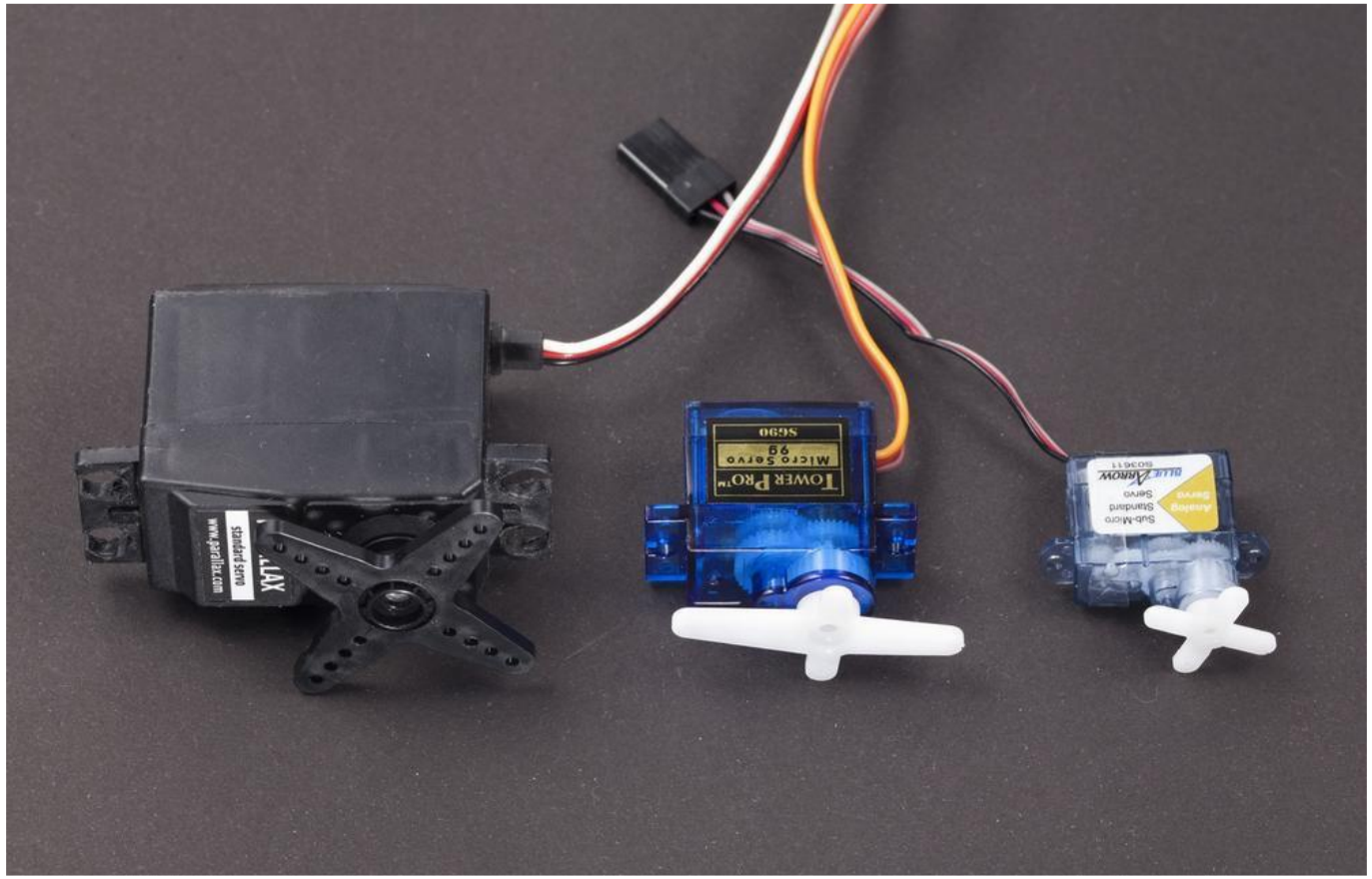
- AnalogWrite(): two arguments: the pin to write to, and a value between 0-255
- Using map() function

Exercise PWM

Print out to change the LED's brightness via PWM

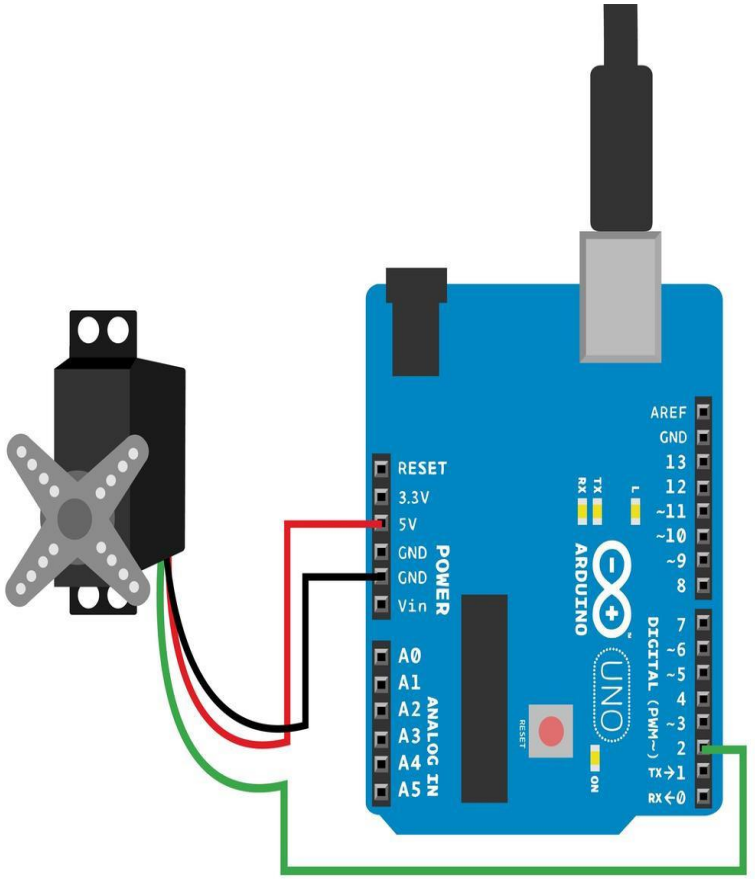


Servo Motors



Servo Motors

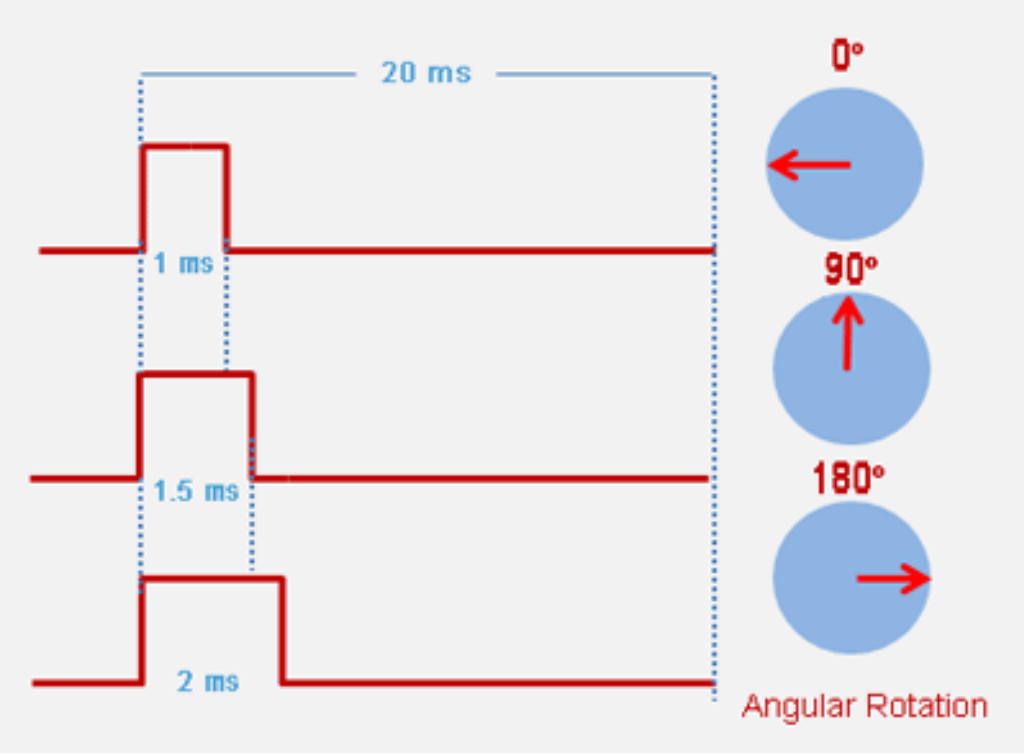
- A servo is a motor you can precisely control
- Servo to turn to a specific angle, such as 90 degrees.
- A servo has three wires: black ground (0 V), red positive (+5 V), and control (yellow or white).
- The length of these pulses tell the servo which angle to move to



Pulse length controls servo angle

Pulse length ms	Pulse length µs	Angle	Comment
0.5 ms	500 µs	< -90 deg	Trying to turn over range, ugly sound from gears
1 ms	1000 µs	-90 deg	Extreme left
1.5 ms	1500 µs	0%	Centered
2 ms	2000 µs	90%	Extreme right
2.5 ms	2500 µs	> 90 deg	Over range, ugly sound

Angle and Stop



Servo Motors