Electronics 1

Electronics for Interactive Media Design

Les 3

Emma Pareschi

From the last class

Theory
- Ohm's Law: V = RI
- Power: P = VI

- pull-up resistor
- Voltage divider

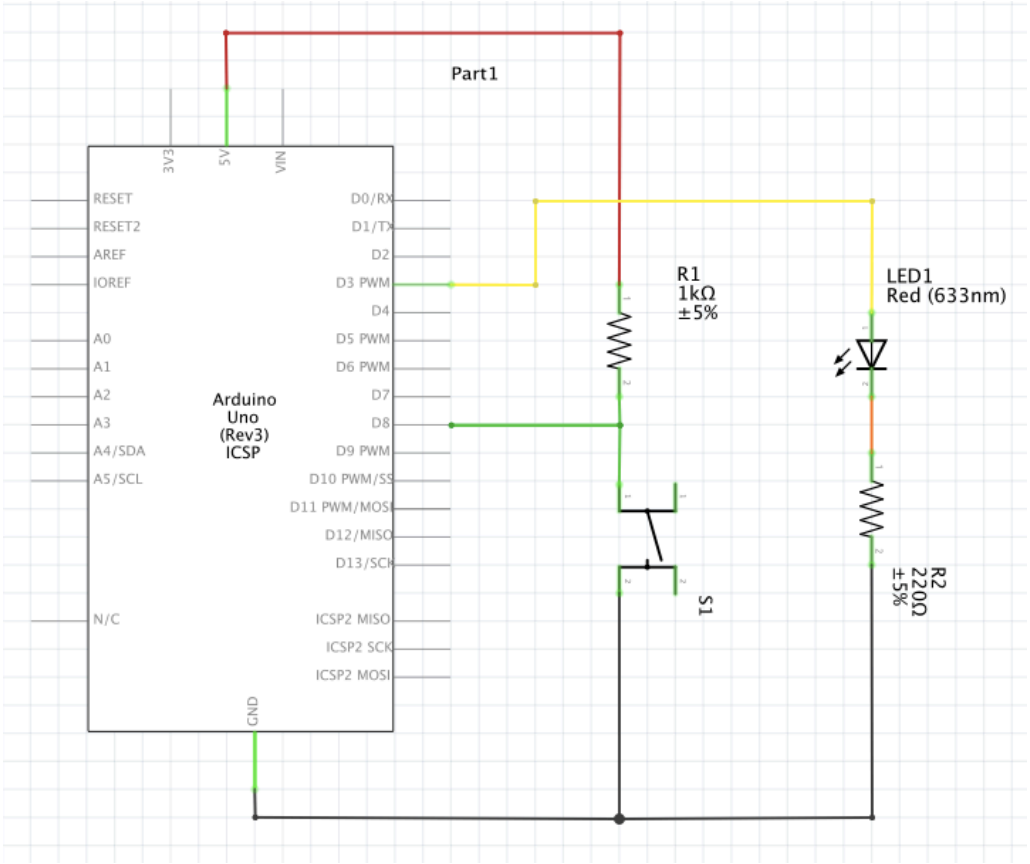
Hardware
- Breadboard
- Arduino:
  - Power pins
  - Digital pins
  - PWM pins
- Input Device:
  - Tact switch
- Output Device:
  - Led

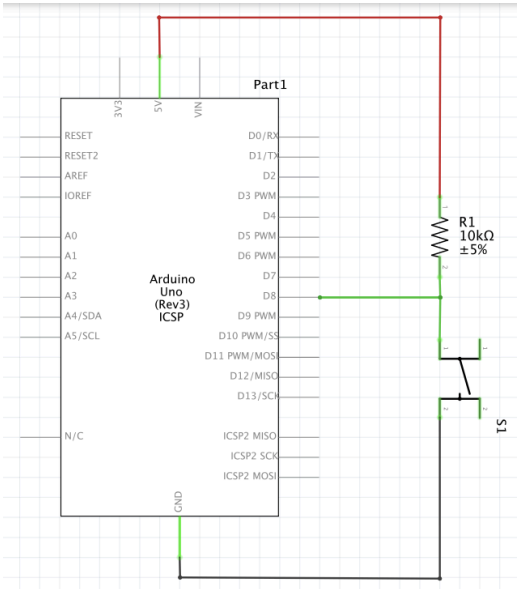
- LDR
- Potentiometer

Software
- structure of Arduino sketch
  - Setup
  - Loop
- Function
  - pinMode
  - digitalWrite / analogWrite
  - digitalRead
  - Serial
  - If…else

- int / float
- if / if…else
- While
- Millis
- Map
- Constrain
(calibration)

Turn on/off the Led based on the switch





Sketch - Led and Switch

```
// constants won't change. They're used here to
// set pin numbers:
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin

// variables will change:
int buttonState = 0;         // variable for reading the pushbutton sta

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  } else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

Control Structure ‘if..else’:

```
if (this condition happens)
{
  // action A
}
else
{
  // action B
}
```

IF

```
if (someVariable > 50)
{
  // do something here
}
```

```
if (x > 120) digitalWrite(LEDpin, HIGH);

if (x > 120)
digitalWrite(LEDpin, HIGH);

if (x > 120){ digitalWrite(LEDpin, HIGH); }

if (x > 120){
  digitalWrite(LEDpin1, HIGH);
  digitalWrite(LEDpin2, HIGH);
}                                  // all are correct
```

If…else

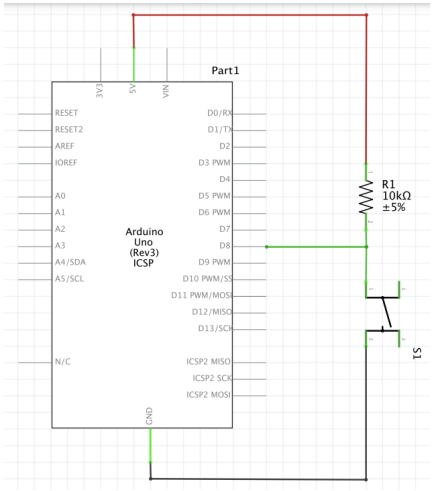
```
if (pinFiveInput < 500)
{
  // action A
}
else
{
  // action B
}
```

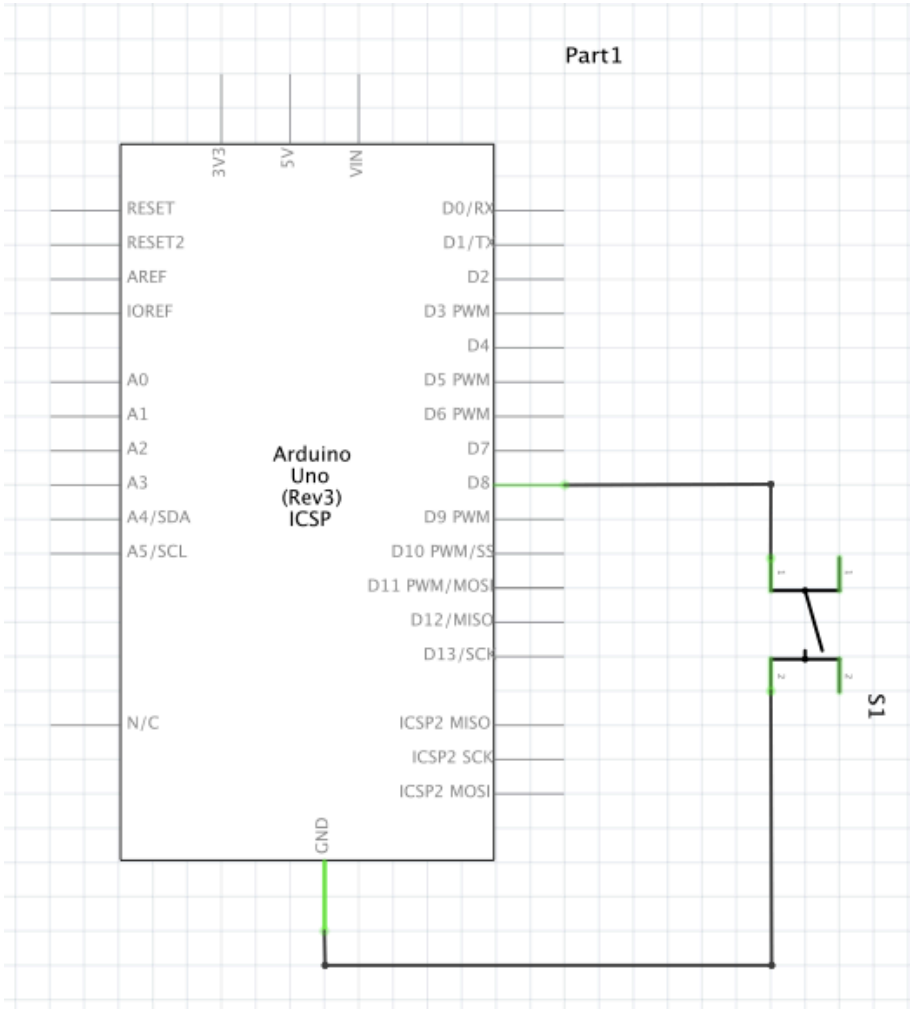
```
if (pinFiveInput < 500)
{
  // do Thing A
}
else if (pinFiveInput >= 1000)
{
  // do Thing B
}
else
{
  // do Thing C
}
```

Comparison Operators

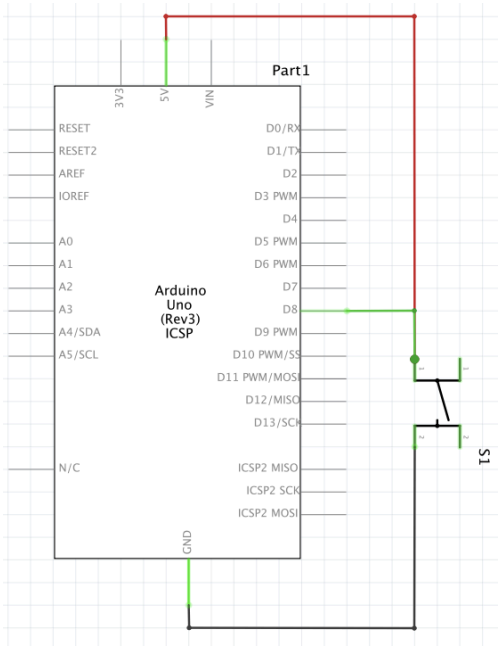
```
x == y (x is equal to y)
x != y (x is not equal to y)
x < y (x is less than y)
x > y (x is greater than y)
x <= y (x is less than or equal to y)
x >= y (x is greater than or equal to y)
```

Pull-up Resistor and Floating Pin

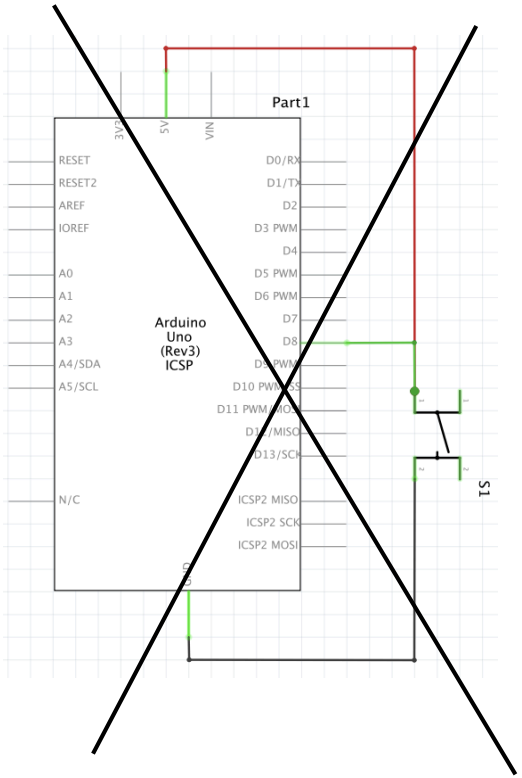




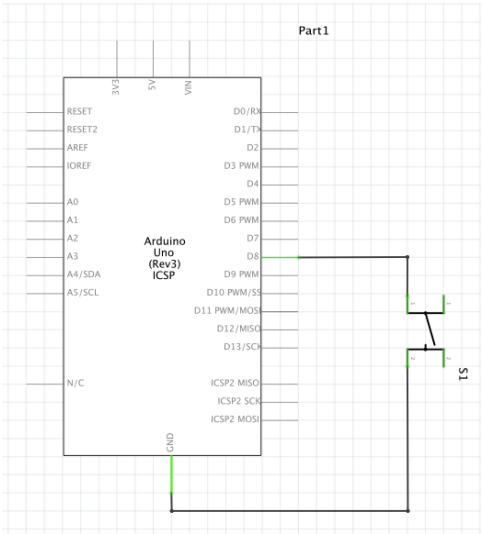
Pull-Up Resistor

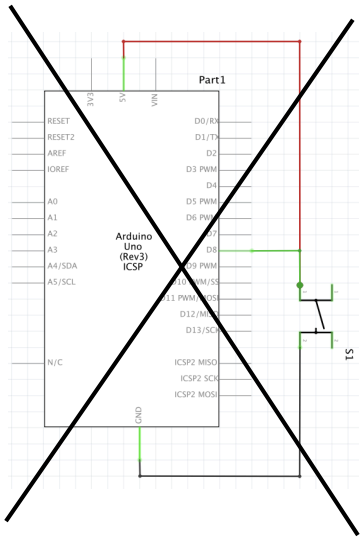


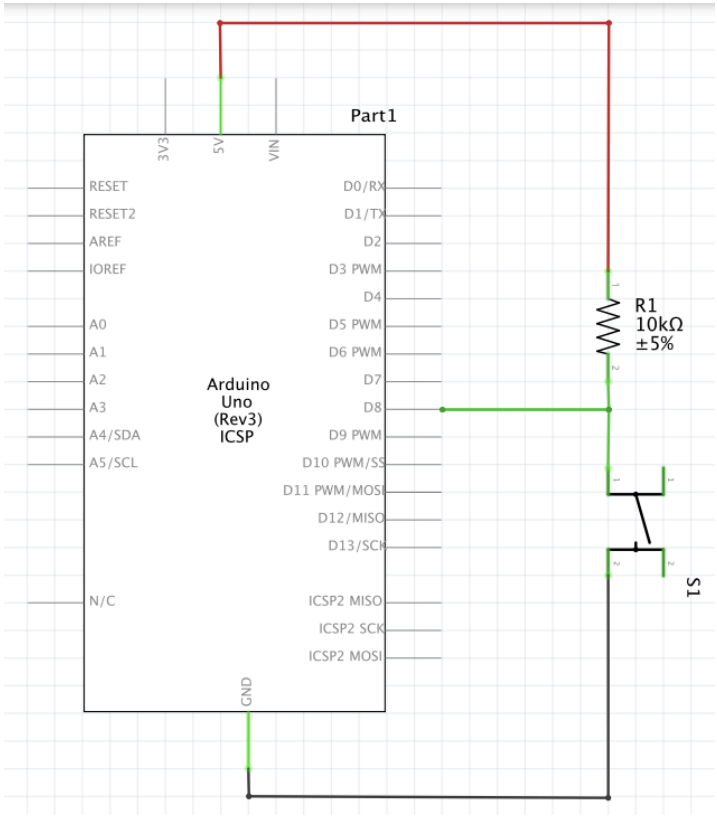
Pull-Up Resistor



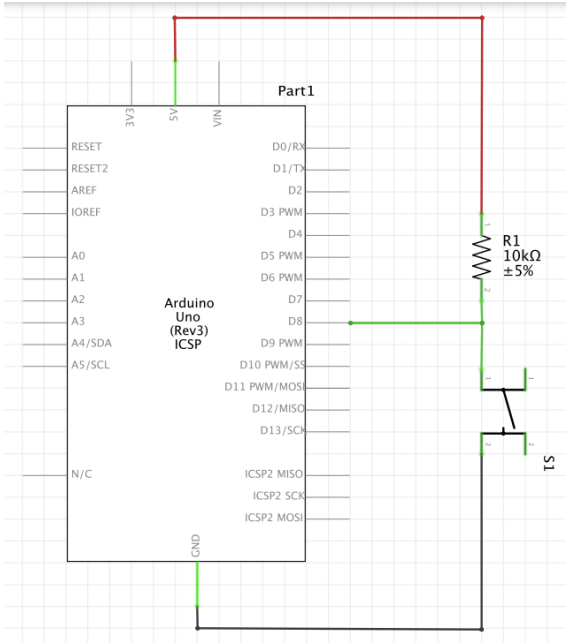
Pull-Up Resistor

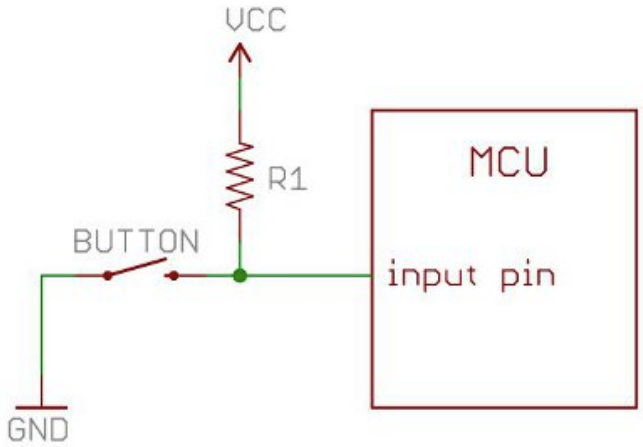






Pull-Up Resistor



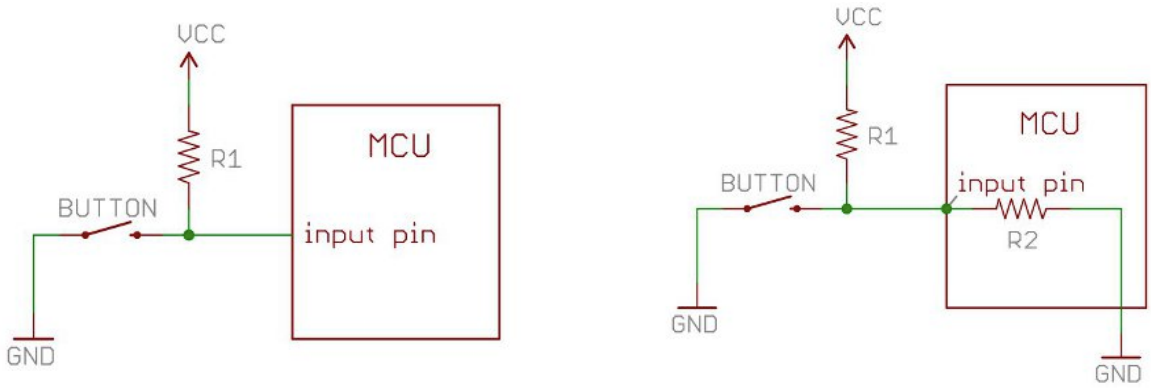


Pull-Up Resistor

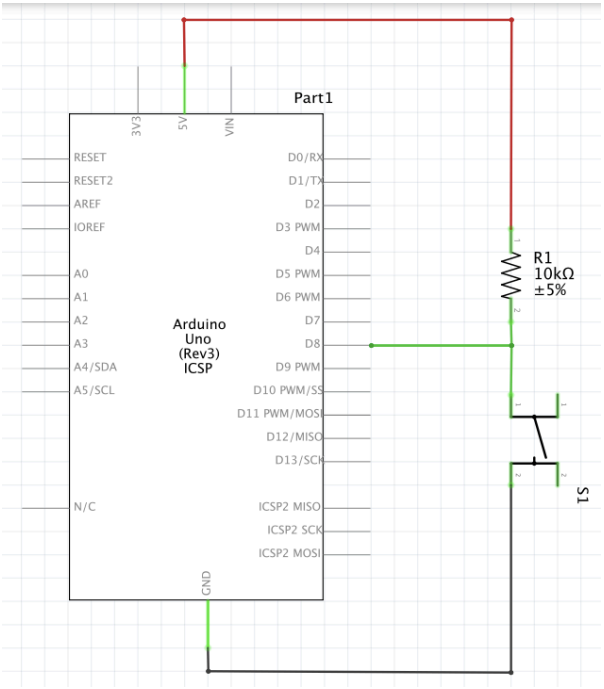
The value of the pull-up resistor needs to be chosen to satisfy two conditions:

1. When the button is pressed, the input pin is pulled low. The value of resistor R1 controls how much current you want to flow from VCC, through the button, and then to ground.

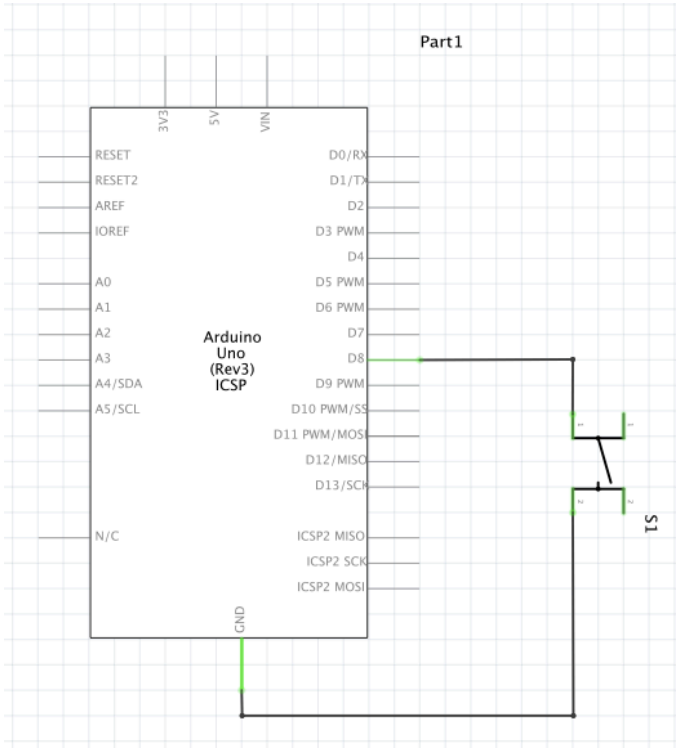
2. When the button is not pressed, the input pin is pulled high. The value of the pull-up resistor controls the voltage on the input pin.



Pull-Up Resistor and Internal Pull-up

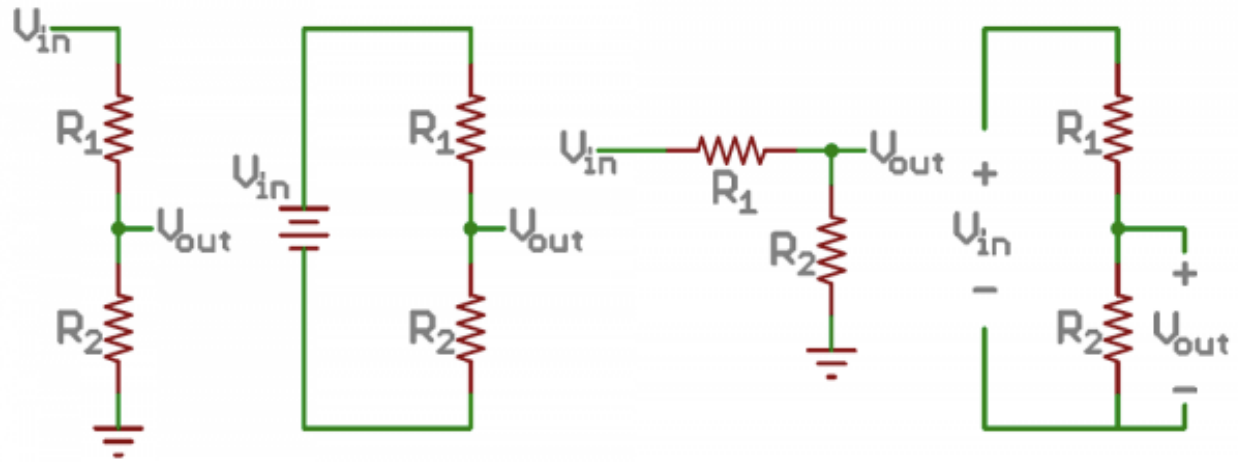


pinMode(3, INPUT);



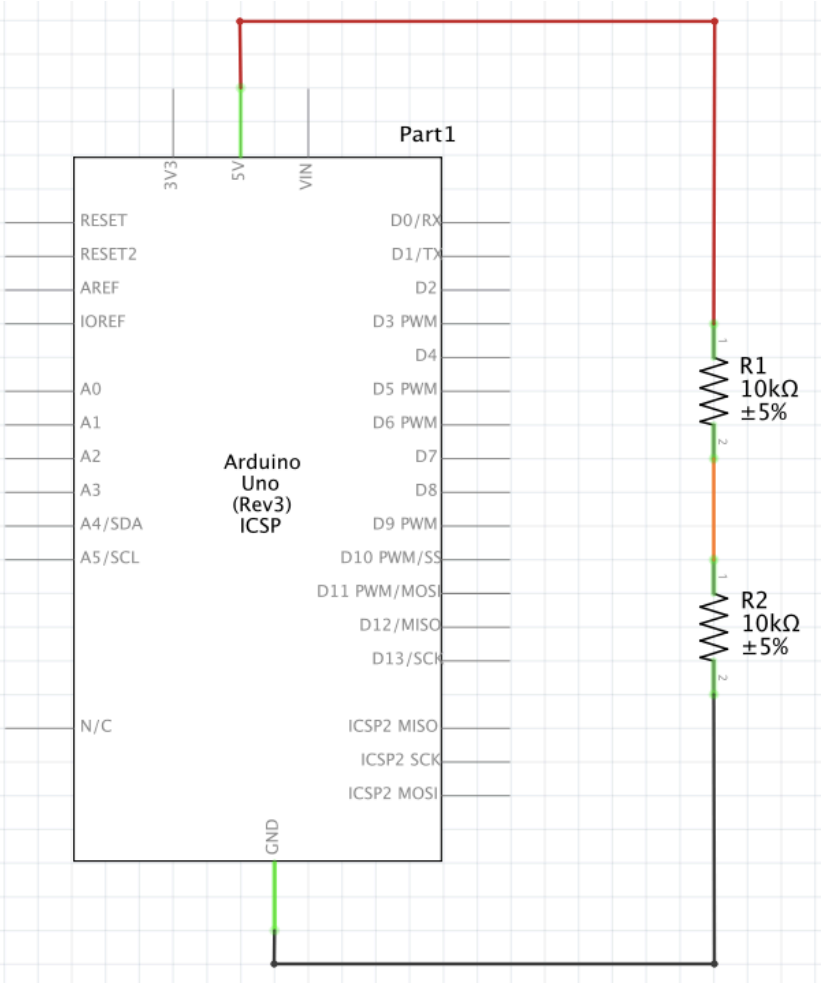
pinMode(3, INPUT_PULLUP);

Voltage Divider for analog sensors



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

Voltage Divider

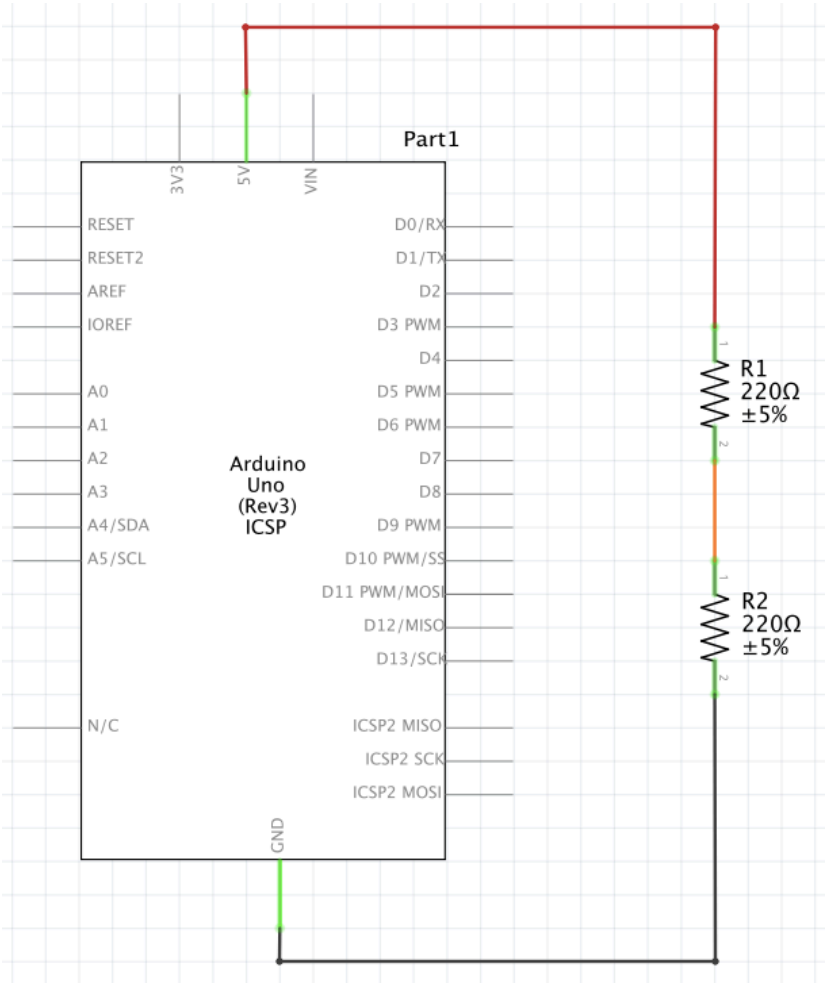


$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

R1 = 10KOhm
R2 = 10KOhm

Vout = ?

Voltage Divider

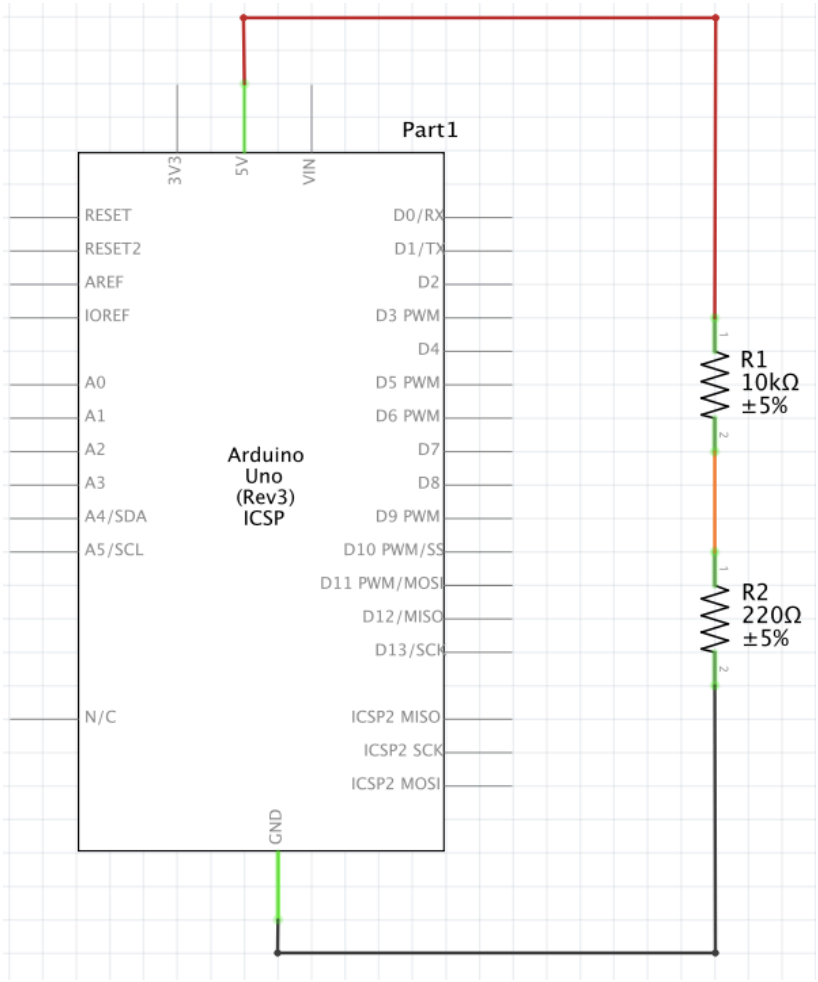


$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

R1 = 220Ohm
R2 = 220Ohm

Vout = ?

Voltage Divider

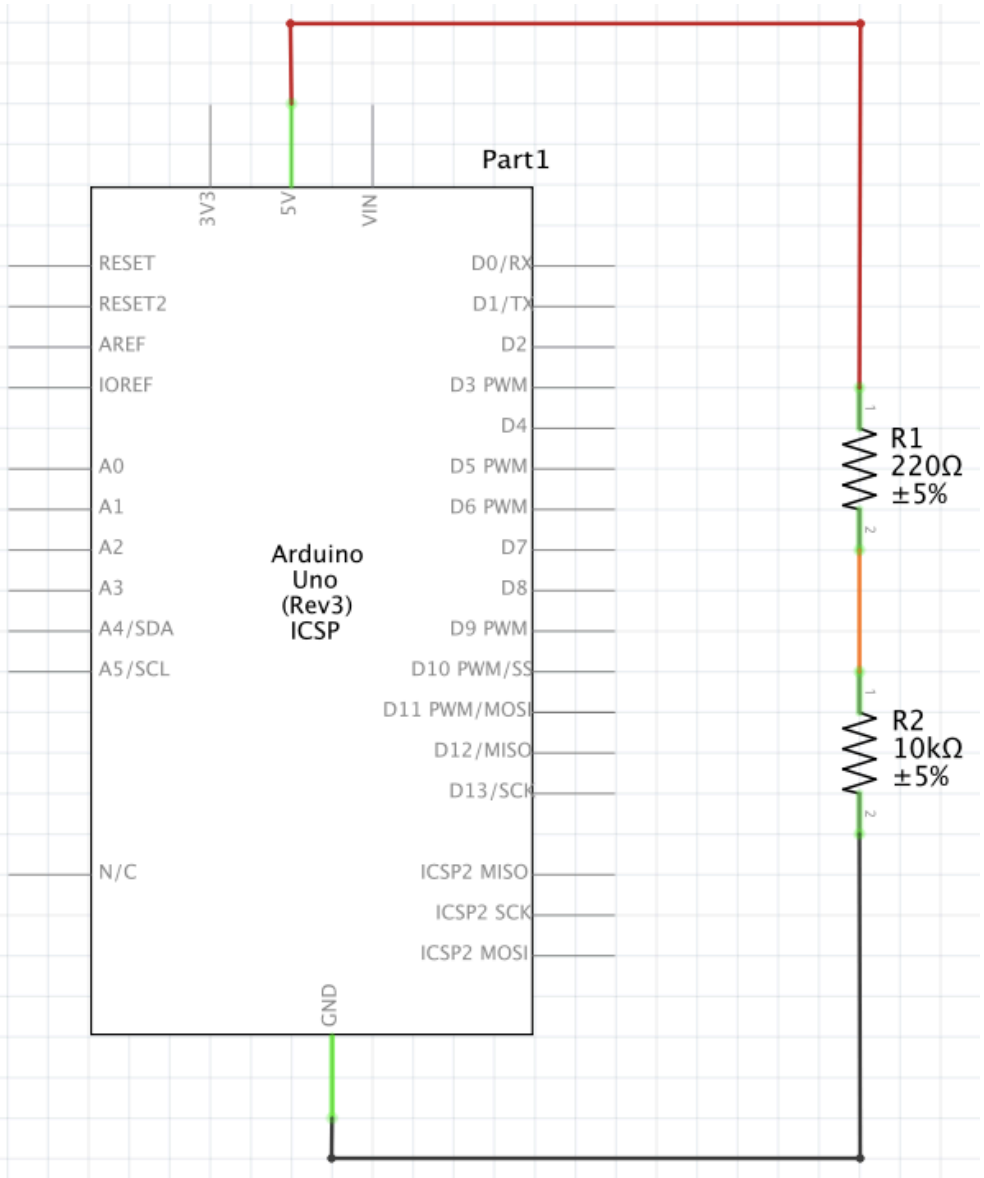


$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

R1 = 10K Ohm
R2 = 220 Ohm

Vout = ?

Voltage Divider



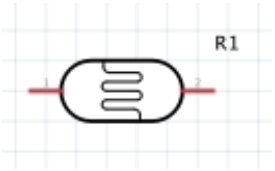
$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

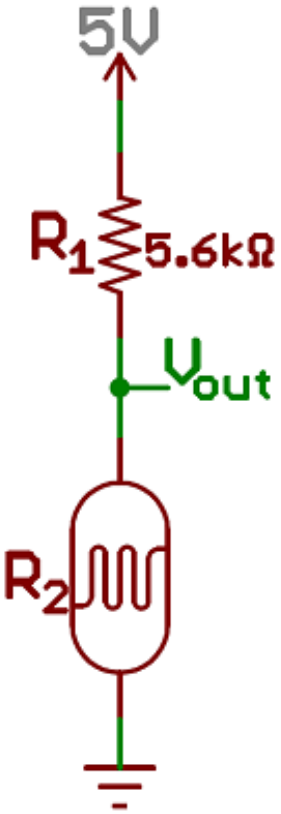
R1 = 220Ohm
R2 = 10KOhm

Vout = ?

Resistor divider - Application

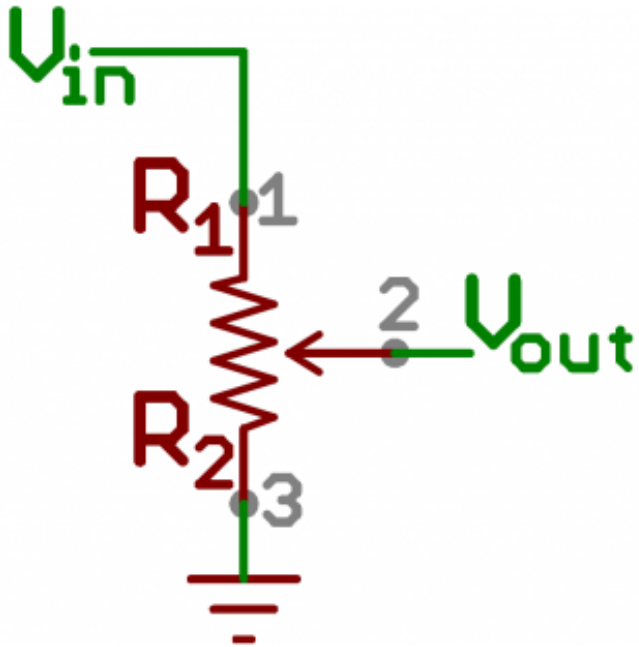
Resistive Sensors



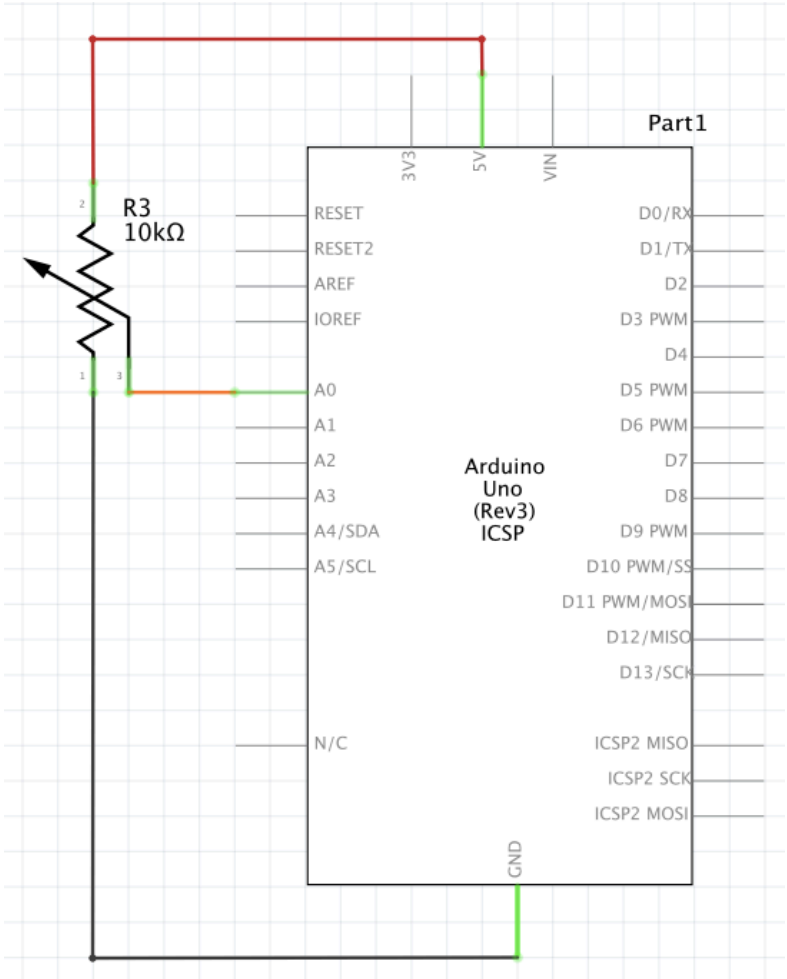


Light Level	R_2 (Sensor)	R_1 (Fixed)	Ratio $R_2/(R_1+R_2)$	V_{out}
Light	1kΩ	5.6kΩ	0.15	0.76 V
Dim	7kΩ	5.6kΩ	0.56	2.78 V
Dark	10kΩ	5.6kΩ	0.67	3.21 V

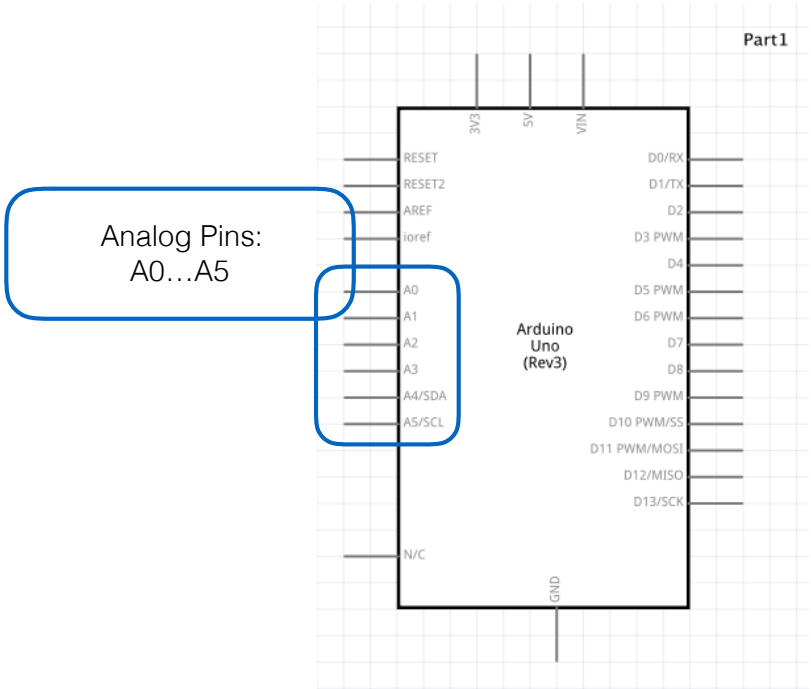
Resistor divider - Application



Potentiometer - schematic



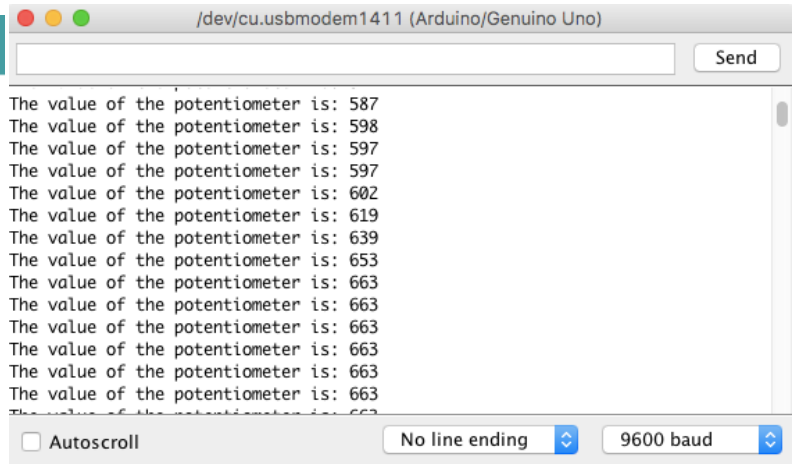
To read a potentiometer or any sensors that generate an analog voltage => you need to use the analog pins.



Potentiometer - Sketch

pot_serial

```
/* Emma Pareschi
 *  October 2017
 *  I read the output voltage of a potenziometer and
 *  I print it on the Serial Monitor
 */

const int pot_pin = A0; //set the variable of the pot pin, it's a constant
int pot_value = 0;     //set the variable to save the status of the pot

void setup() {
  pinMode(pot_pin, INPUT); //define the function of the pin
  Serial.begin(9700);     //open communication
}

void loop() {

  pot_value = analogRead(pot_pin); //read the push button status

  Serial.print("The value of the potentiometer is: ");
  Serial.println(pot_value);

  delay(100);                      // wait
}
```

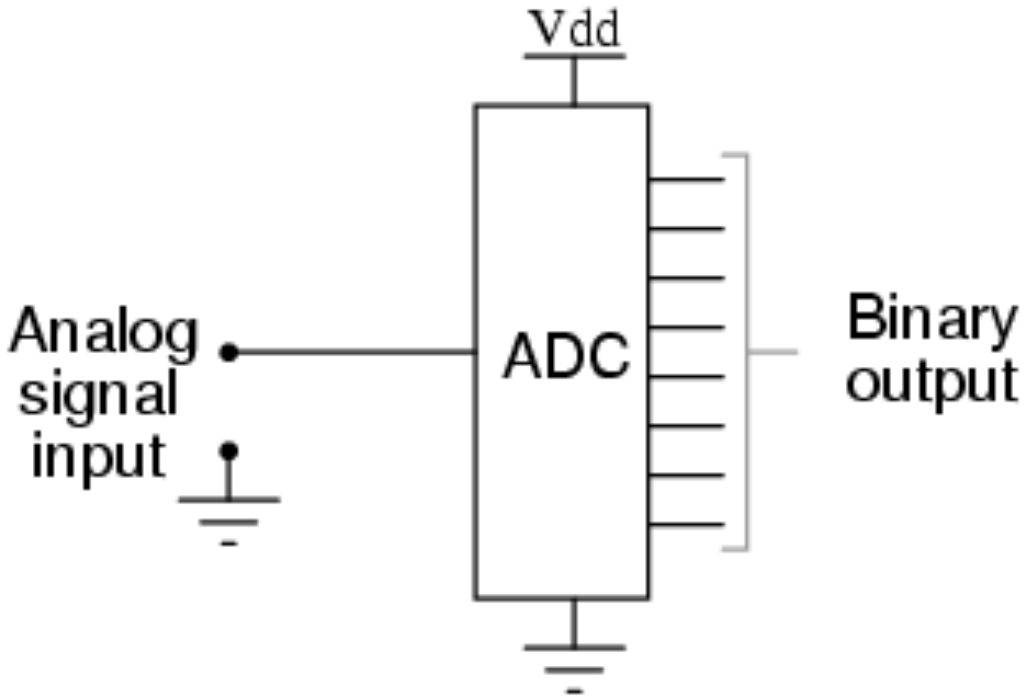


Function to read an analog voltage is:

analogRead(pin);

It will return a number between 0 and 1023

Analog to Digital Converter

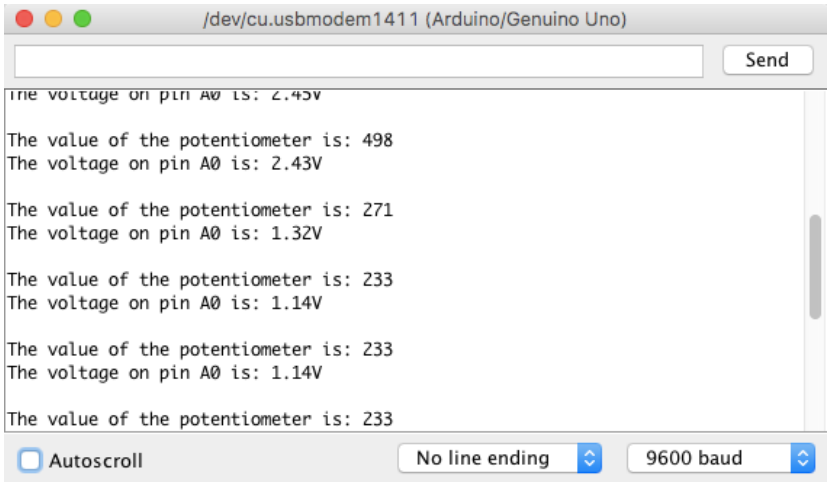


Potentiometer - Sketch

pot_adc

```
/* Emma Pareschi
 *  October 2017
 *  I read the output voltage of a potenziometer and
 *  I print it on the Serial Monitor
 */

const int pot_pin = A0; //set the variable of the pot pin, it's a constant
int pot_value = 0;     //set the variable to save the status of the pot

float pot_voltage = 0; //set the variable to express the reading in analog voltage

void setup() {
  pinMode(pot_pin, INPUT); //define the function of the pin
  Serial.begin(9700);     //open communication
}

void loop() {

  pot_value = analogRead(pot_pin);  //read the push button status

  pot_voltage = pot_value * (5.0 / 1023.0);

  Serial.print("The value of the potentiometer is: ");
  Serial.println(pot_value);
  Serial.print("The voltage on pin A0 is: ");
  Serial.println(pot_voltage);
  Serial.println("");

  delay(1000);                      // wait
}
```

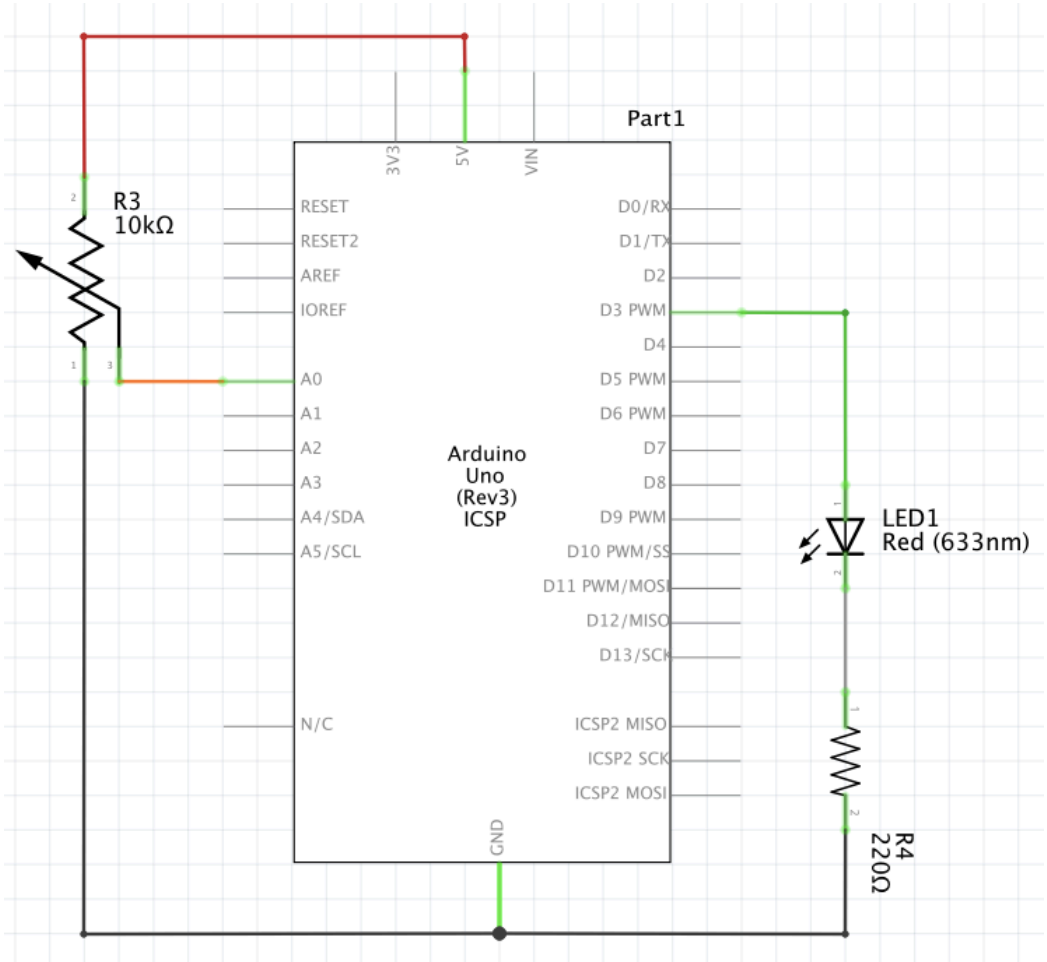
DATA TYPE:
INT vs FLOAT
INT: INTEGER NUMBER
FLOAT: DECIMAL NUMBER



Data Type

Numeric type	Bytes	Range	Use
int	2	-32768 to 32767	Represents positive and negative integer values.
float	4	3.4028235E+38 to -3.4028235E+38	Represents numbers with fractions; use to approximate real-

Potentiometer and LED - Schematic



Potentiometer and LED - Sketch

pot_led_1

```
/*
 * Emma Pareschi
 * October 2017
  Analog Input
 Demonstrates analog input by reading an analog sensor on analog pin 0 and
 turning on and off a light emitting diode(LED)  connected to digital pin 3.
 The amount of time the LED will be on and off depends on
 the value obtained by analogRead().
 */

int pot_pin = A0;    // select the input pin for the potentiometer
int led_pin = 3;      // select the pin for the LED
int pot_value = 0;  // variable to store the value coming from the sensor

void setup() {
  // declare the ledPin as an OUTPUT and the potentiometer as INPUT
  pinMode(led_pin, OUTPUT);
  pinMode(pot_pin, INPUT);
}

void loop() {
  // read the value from the sensor:
  pot_value = analogRead(pot_pin);
  // turn the ledPin on
  digitalWrite(led_pin, HIGH);
  // stop the program for <sensorValue> milliseconds:
  delay(pot_value);
  // turn the ledPin off:
  digitalWrite(led_pin, LOW);
  // stop the program for for <sensorValue> milliseconds:
  delay(pot_value);
}
```

Potentiometer and LED - Sketch

pot_led_2

```
/*
 * Emma Pareschi
 * October 2017
 */

const int pot_pin = A0;    // select the input pin for the potentiometer
int led_pin = 3;      // select the pin for the LED
int pot_value = 0;  // variable to store the value coming from the sensor

void setup() {
  // declare the ledPin as an OUTPUT and the potentiometer as INPUT
  pinMode(led_pin, OUTPUT);
  pinMode(pot_pin, INPUT);
}

void loop() {
  // read the value from the sensor:
  pot_value = analogRead(pot_pin);

  pot_value = map(pot_value,0,1023,0,255); //Convert from 0-1023 to 0-255

  analogWrite(led_pin,pot_value);//turn the led on depend on the output value

}
```

Potentiometer and LED - Sketch

pot_led_2

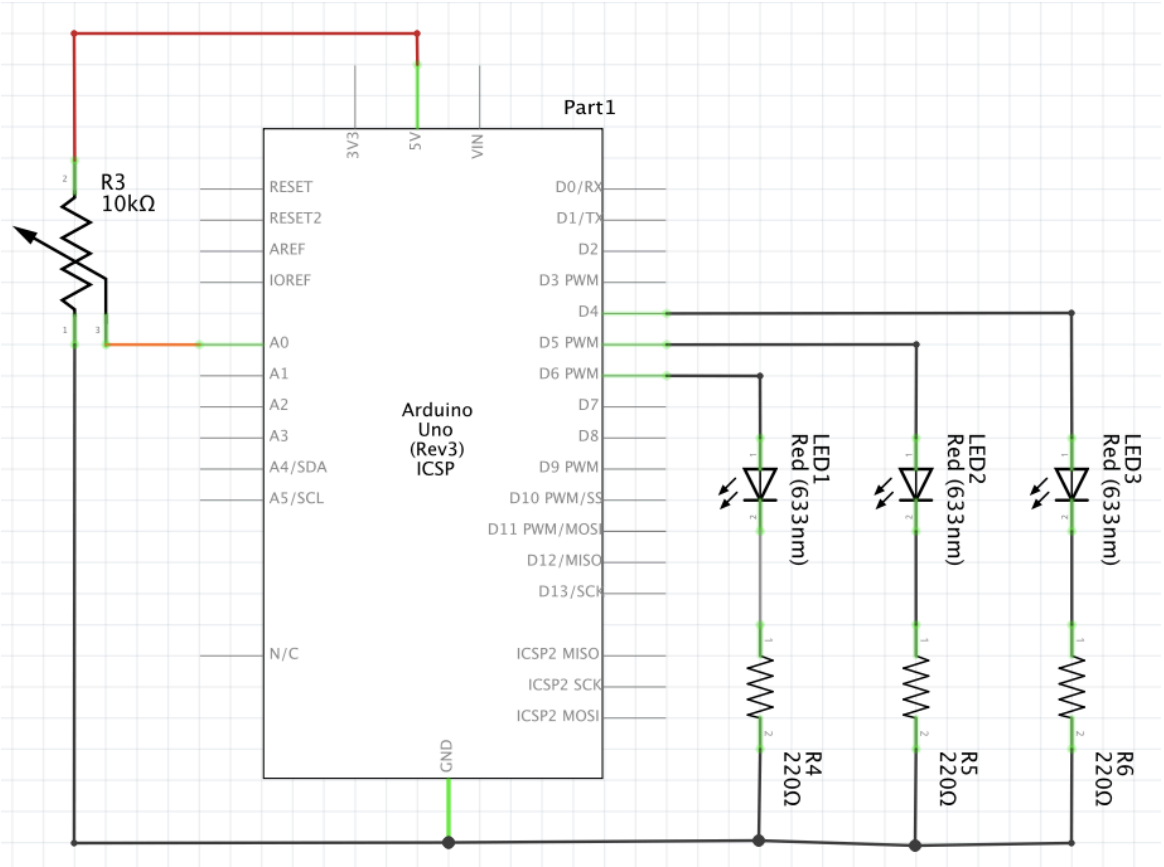
```
/*
 * Emma Pareschi
 * October 2017
 */

const int pot_pin = A0;    // select the input pin for the potentiometer
int led_pin = 3;      // select the pin for the LED
int pot_value = 0;  // variable to store the value coming from the sensor

void setup() {
  // declare the ledPin as an OUTPUT and the potentiometer as INPUT
  pinMode(led_pin, OUTPUT);
  pinMode(pot_pin, INPUT);
}

void loop() {
  // read the value from the sensor:
  pot_value = analogRead(pot_pin);

  pot_value = map(pot_value,0,1023,0,255); //Convert from 0-1023 to 0-255

  analogWrite(led_pin,pot_value);//turn the led on depend on the output value

}
```

Potentiometer and LED - Sketch

pot_led_3

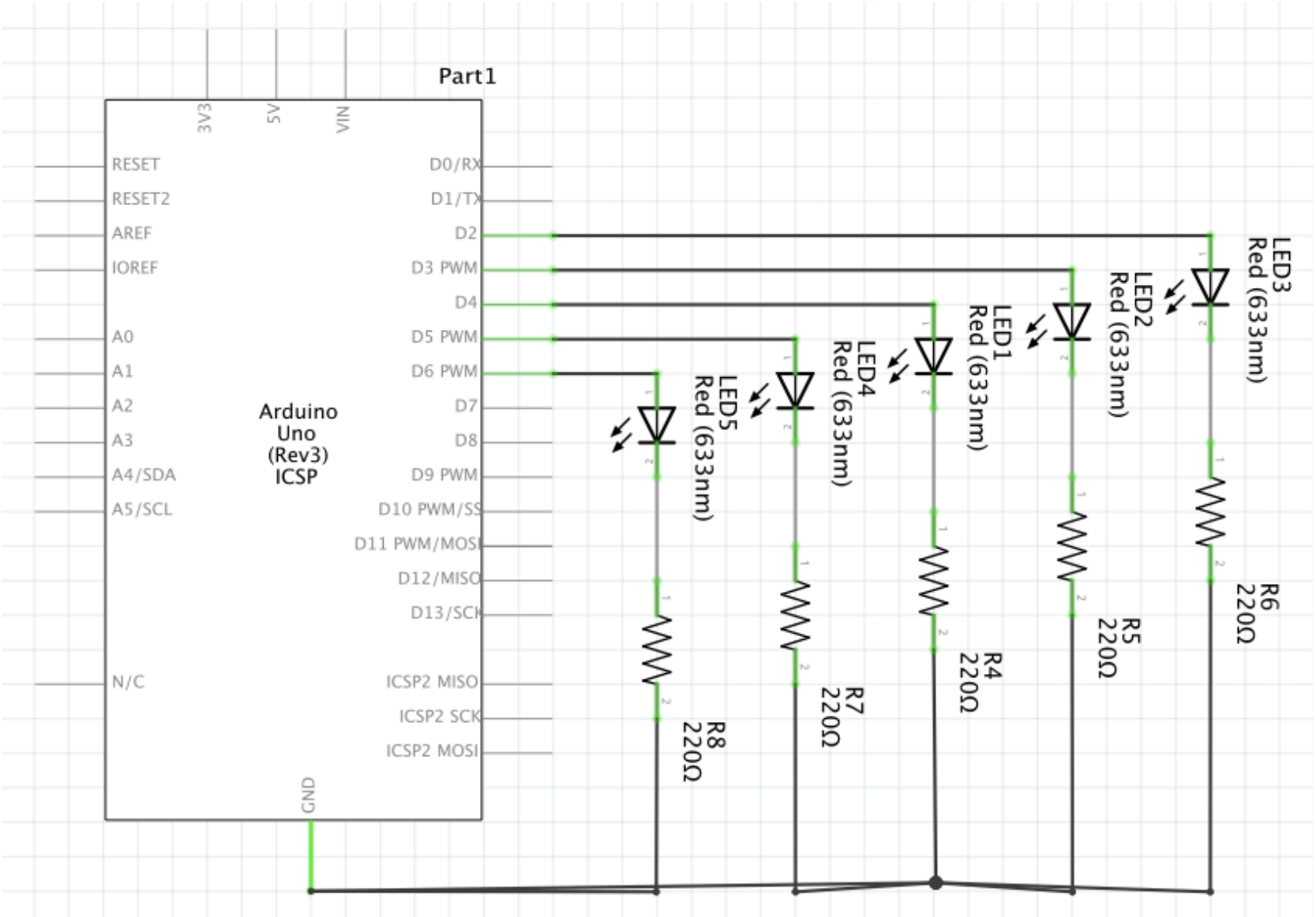
```
/*
 * Emma Pareschi
 * October 2017
 * I control the intensity of a LED using a potentiometer
 */

const int pot_pin = A0;    // select the input pin for the potentiometer
int led_pin = 3;      // select the pin for the LED
int pot_value = 0;  // variable to store the value coming from the sensor

void setup() {
  // declare the ledPin as an OUTPUT and the potentiometer as INPUT
  pinMode(led_pin, OUTPUT);
  pinMode(pot_pin, INPUT);
}

void loop() {
  // read the value from the sensor:
  pot_value = analogRead(pot_pin);

  pot_value = map(pot_value,0,1023,0,255); //Convert from 0-1023 to 0-255

  pot_value = constrain(pot_value, 0, 255); //limit within the range

  analogWrite(led_pin,pot_value);//turn the led on depend on the output value

  delay(100);
}
```

Potentiometer and LED - Schematic



Use the potentiometer to select which Led will turn ON

LED ARRAY- SCHEMATIC



LED ARRAY- SKETCH 2

led_no_array

```
const int Led_pin_1 = 2;
const int Led_pin_2 = 3;
const int Led_pin_3 = 4;
const int Led_pin_4 = 5;
const int Led_pin_5 = 6;

void setup() {
  pinMode(Led_pin_1, OUTPUT);
  pinMode(Led_pin_2, OUTPUT);
  pinMode(Led_pin_3, OUTPUT);
  pinMode(Led_pin_4, OUTPUT);
  pinMode(Led_pin_5, OUTPUT);
}

void loop() {
  digitalWrite(Led_pin_1, HIGH);
  delay(100);
  digitalWrite(Led_pin_1, LOW);
  digitalWrite(Led_pin_2, HIGH);
  delay(100);
  digitalWrite(Led_pin_2, LOW);
  digitalWrite(Led_pin_3, HIGH);
  delay(100);
  digitalWrite(Led_pin_3, LOW);
  digitalWrite(Led_pin_4, HIGH);
  delay(100);
  digitalWrite(Led_pin_4, LOW);
  digitalWrite(Led_pin_5, HIGH);
  delay(100);
  digitalWrite(Led_pin_5, LOW);
}
```

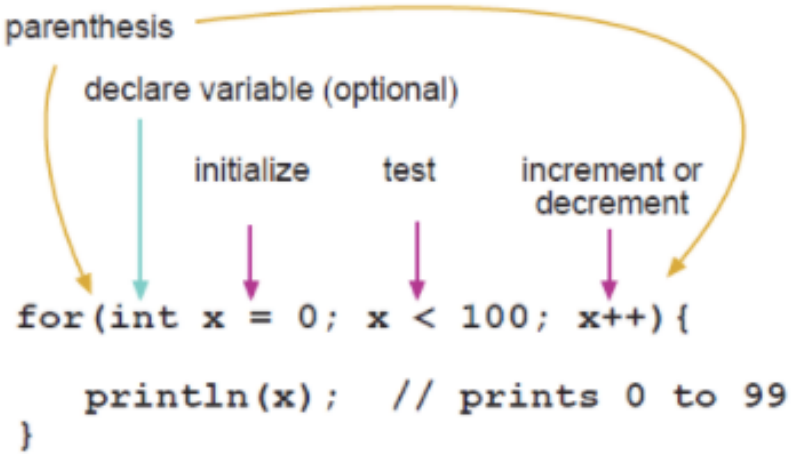
LED ARRAY- SKETCH 2

led_array

```
/*
 * Emma Pareschi
 * Ocotber 2017
 * LED bar graph
*/

// these constants won't change:
const int ledCount = 5;    // the number of LEDs in the bar graph

int ledPins[] = {
  2, 3, 4, 5, 6
};   // an array of pin numbers to which LEDs are attached


void setup() {
  // loop over the pin array and set them all to output:
  for (int i = 0; i < ledCount; i++) {
    pinMode(ledPins[i], OUTPUT);
  }
}

void loop() {
  // loop over the LED array:
  for (int j = 0; j < ledCount; j++) {
    // if the array element's index is less than ledLevel,
    // turn the pin for this element on:
      digitalWrite(ledPins[j], HIGH);
      delay(100);
      digitalWrite(ledPins[j], LOW);
    }
  }
```

Loop ‘FOR’

```
for (initialization; condition; increment) {

//statement(s);

}
```



Loop 'FOR' - ex: dim a LED

for_loop_dim

```
/*
 * Emma Pareschi
 * October 2017
 * Dim an LED using the PWM pin 3 using the loop 'for'
*/

int PWMpin = 3; // LED in series with 470 ohm resistor on pin 10

void setup()
{
  pinMode(PWMpin, OUTPUT);
}

void loop()
{
   for (int i=0; i <= 255; i++){
      analogWrite(PWMpin, i);
      delay(10);
   }

   for (int i=255; i >= 0; i--){
    analogWrite(PWMpin, i);
      delay(10);
   }
}
```

LED ARRAY- SCHEMATIC