

ELECTRONICS 1

ELECTRONICS FOR INTERACTIVE MEDIA DESIGN
LES 3

EMMA PARESCHI

FROM THE LAST CLASS

THEORY

- OHM'S LAW: $V = RI$
- POWER: $P = VI$

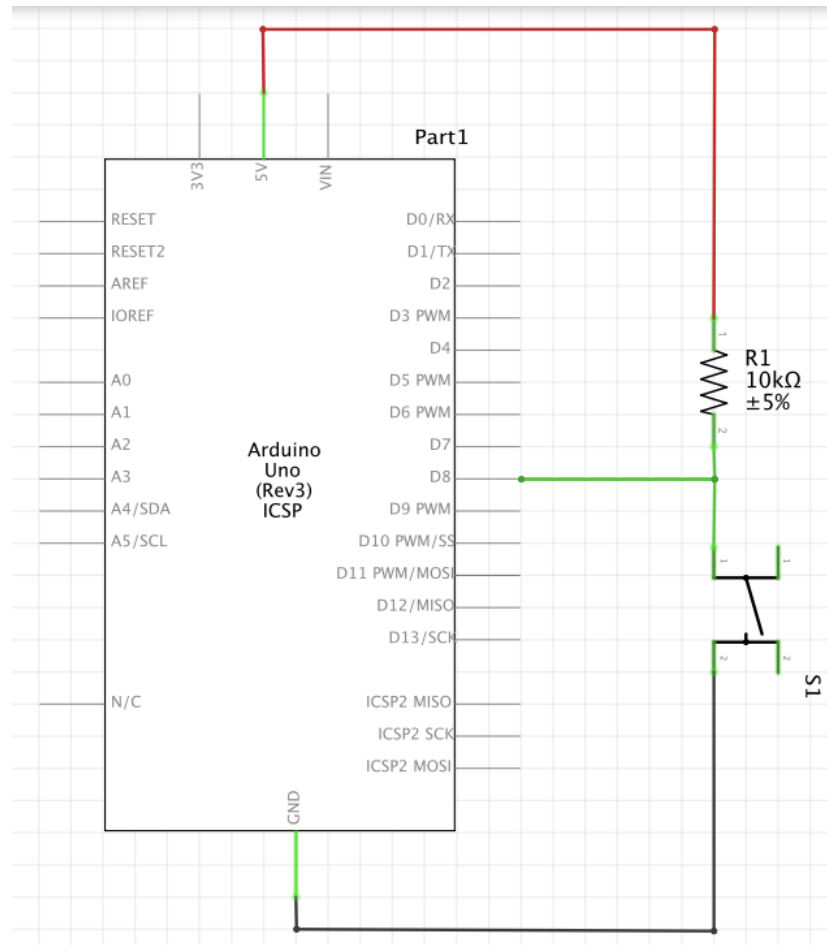
HARDWARE

- BREADBOARD
- ARDUINO:
 - POWER PINS
 - DIGITAL PINS
 - PWM PINS
- INPUT DEVICE:
 - TACT SWITCH
- OUTPUT DEVICE:
 - LED

SOFTWARE

- STRUCTURE OF ARDUINO SKETCH
 - SETUP
 - LOOP
- FUNCTION
 - PINMODE
 - DIGITALWRITE / ANALOGWRITE
 - DIGITALREAD
 - SERIAL

FROM THE LAST CLASS



HOW TO READ THE BUTTON AND PRINT
THE STATUS ON THE SERIAL MONITOR.

THE PUSH BUTTON IS READ BY THE
DIGITAL PIN 8.

SKETCH

```
tact_sw
/* Emma Pareschi
 * September 2017
 * I read the status of a push button and I print it on a Serial Monitor.
 */

const int sw_pin = 8; //set the variable of the switch pin, it's a constant
int sw_state = 0;      //set the variable to save the status of the switch

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

void loop() {
  sw_state = digitalRead(sw_pin); //read the push button status
  Serial.println(sw_state);        // print the status on the Serial Monitor
  delay(100);                      // wait
}
```

CONTROL STRUCTURE 'IF'

```
tact_sw_simpleIF_serial
/* Emma Pareschi
 * Septmber 2017
 * I read the status of a push button and
 * IF the push button is pressed, I print "CIAO" on the Serial Monitor
 */

const int sw_pin = 8; //set the variable of the switch pin, it's a constant
int sw_state = 0;      //set the variable to save the status of the switch

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

void loop() {

  sw_state = digitalRead(sw_pin); //read the push button status

  if (sw_state == LOW){
    Serial.println("CIAO");
  }

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

Comparison operator
equal ==
not equal !=

Control Structure 'if':

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

IF AND COMPARISON COMPARATOR

```
//all correct|
if (sw_state == LOW){
    Serial.println("CIAO");
}

if (sw_state == LOW){ Serial.println("CIAO"); }

if (sw_state == LOW) Serial.println("CIAO");
```

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)
```

CONTROL STRUCTURE 'IF...ELSE'

tact_sw_if_serial

```
/* Emma Pareschi
 * September 2017
 * I read the status of a push button and
 * I print a question on a Serial Monitor when the button is pressed
 */

const int sw_pin = 8; //set the variable of the switch pin, it's a constant
int sw_state = 0;      //set the variable to save the status of the switch

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

void loop() {
  sw_state = digitalRead(sw_pin); //read the push button status

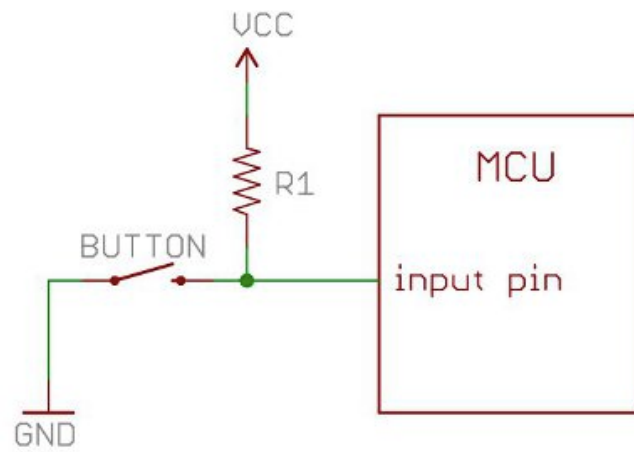
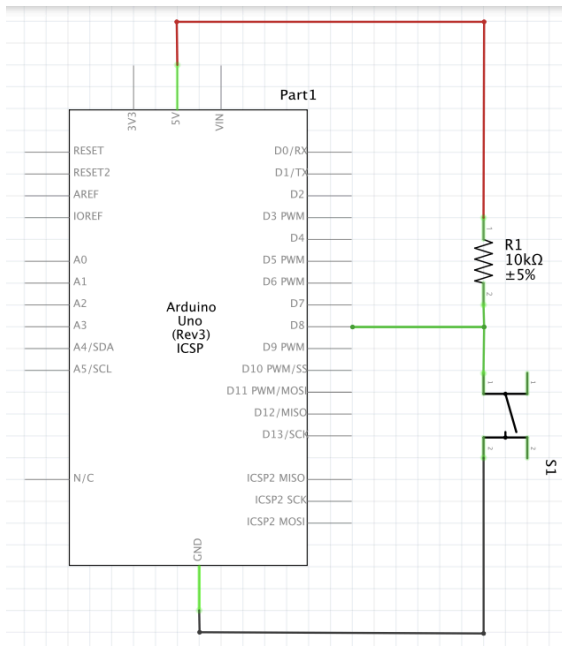
  if (sw_state == HIGH){
    Serial.println("...zzz...");
  } else {
    Serial.println("");
    Serial.println("did you call me?");
    Serial.println("");
    delay(1000);
  }

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

Control Structure 'if..else'

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

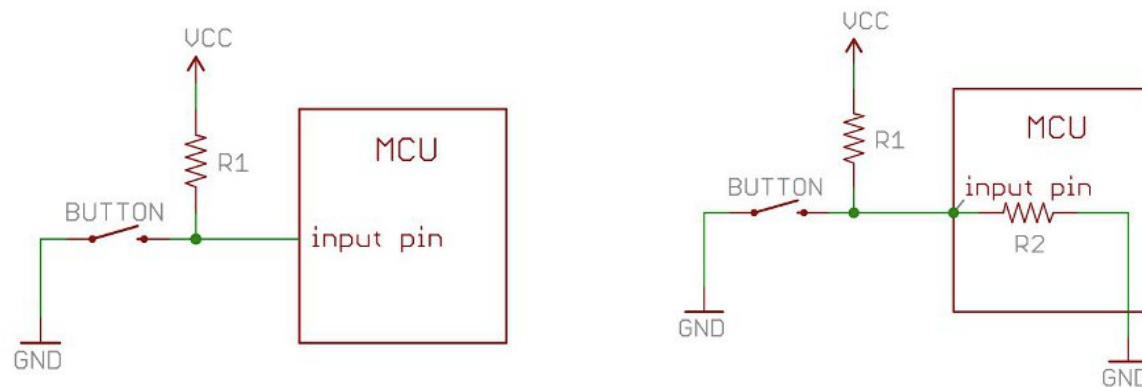
PULL-UP RESISTOR



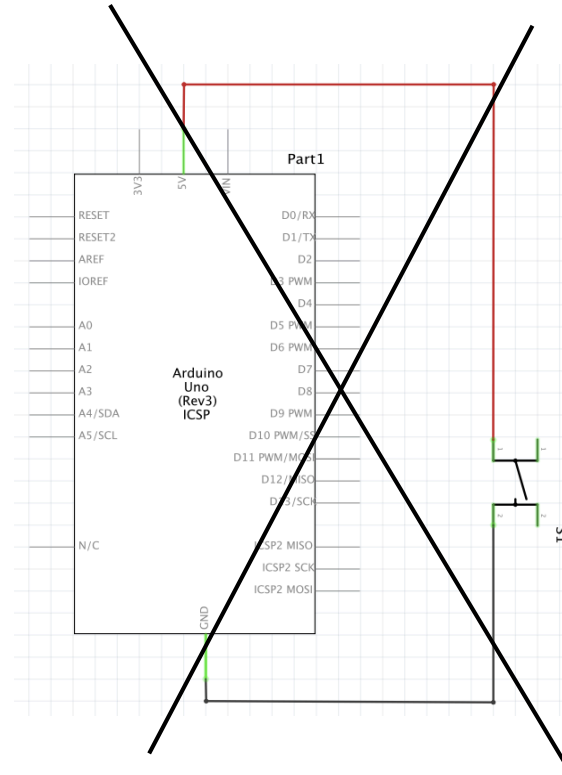
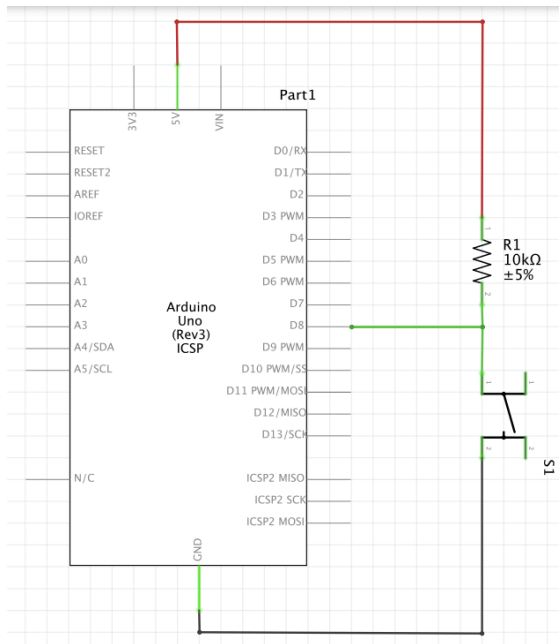
PULL-UP RESISTOR

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

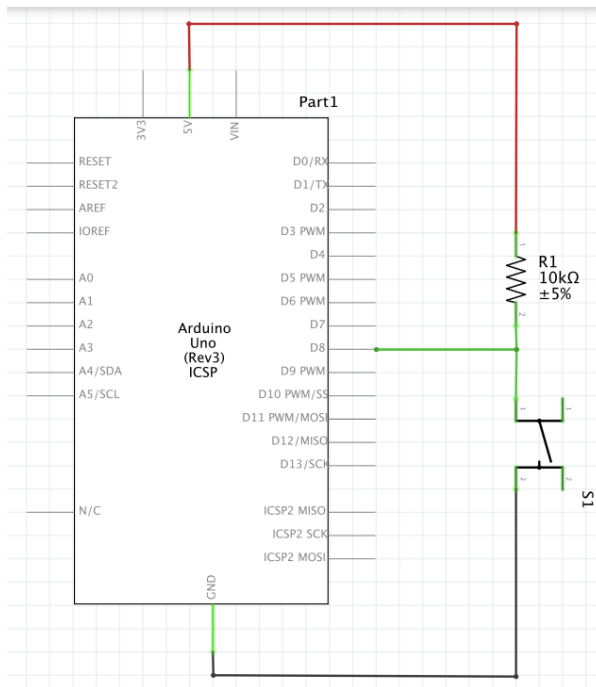
1. When the button is pressed, the input pin is pulled low. The value of resistor $R1$ controls how much current you want to flow from VCC, through the button, and then to ground.
2. When the button is not pressed, the input pin is pulled high. The value of the pull-up resistor controls the voltage on the input pin.



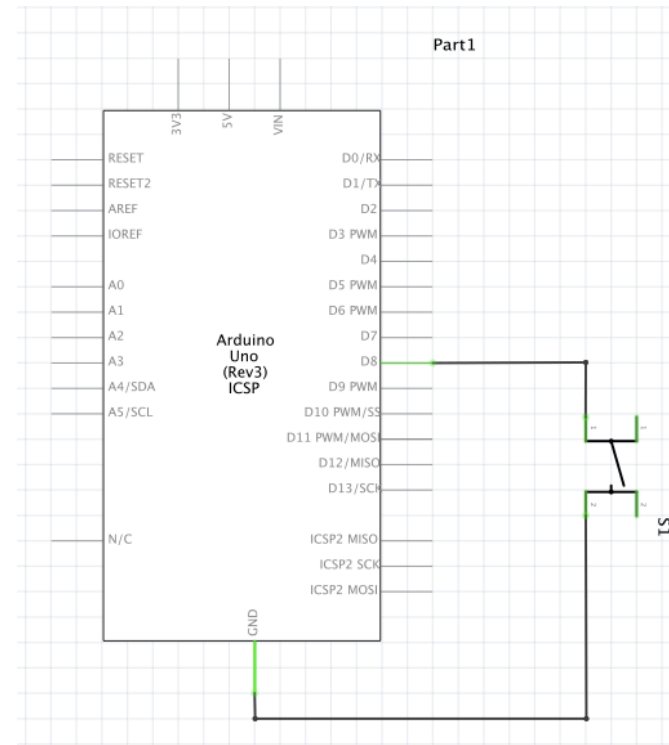
PULL-UP RESISTOR



PULL-UP RESISTOR

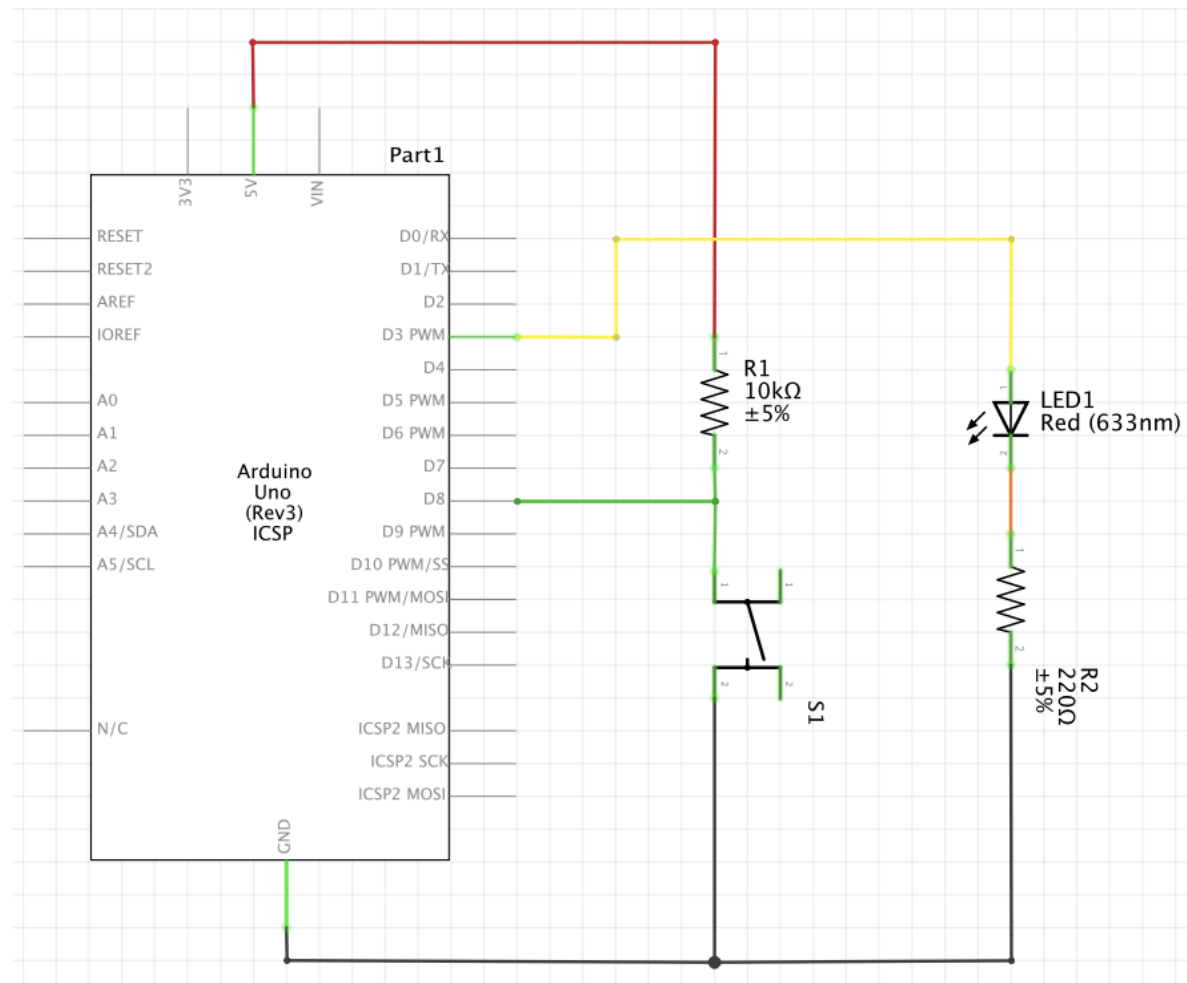


```
PINMODE(3, INPUT);
```



```
pinMode(3, INPUT_PULLUP);
```

CONTROL A LED WITH A SWITCH



CONTROL A LED WITH A SWITCH - SKETCH

```
tact_sw_led
/* Emma Pareschi
 * Septmber 2017
 * I read the status of a push button and
 * turn on the LED if the push button is NOT presses
 */

const int sw_pin = 8; //set the variable of the switch pin, it's a constant
int sw_state = 0;      //set the variable to save the status of the switch

const int led_pin = 3; //the pin of the LED

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

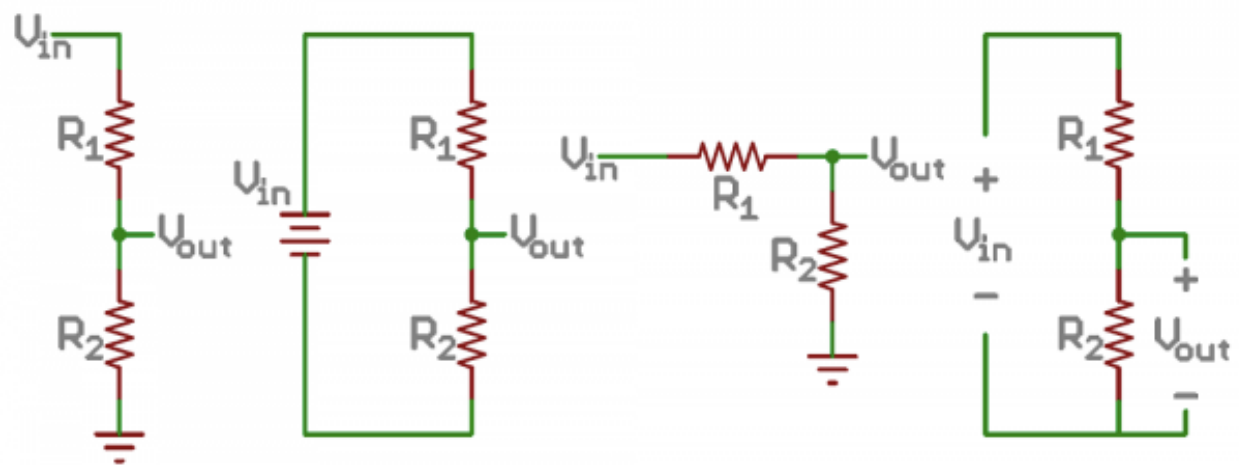
void loop() {

  sw_state = digitalRead(sw_pin); //read the push button status

  if (sw_state == HIGH){          //if the sw is NOT pushed
    digitalWrite(led_pin, HIGH); //LED ON
  } else {                        //if the sw is pushed
    digitalWrite(led_pin, LOW);  //LED OFF
  }

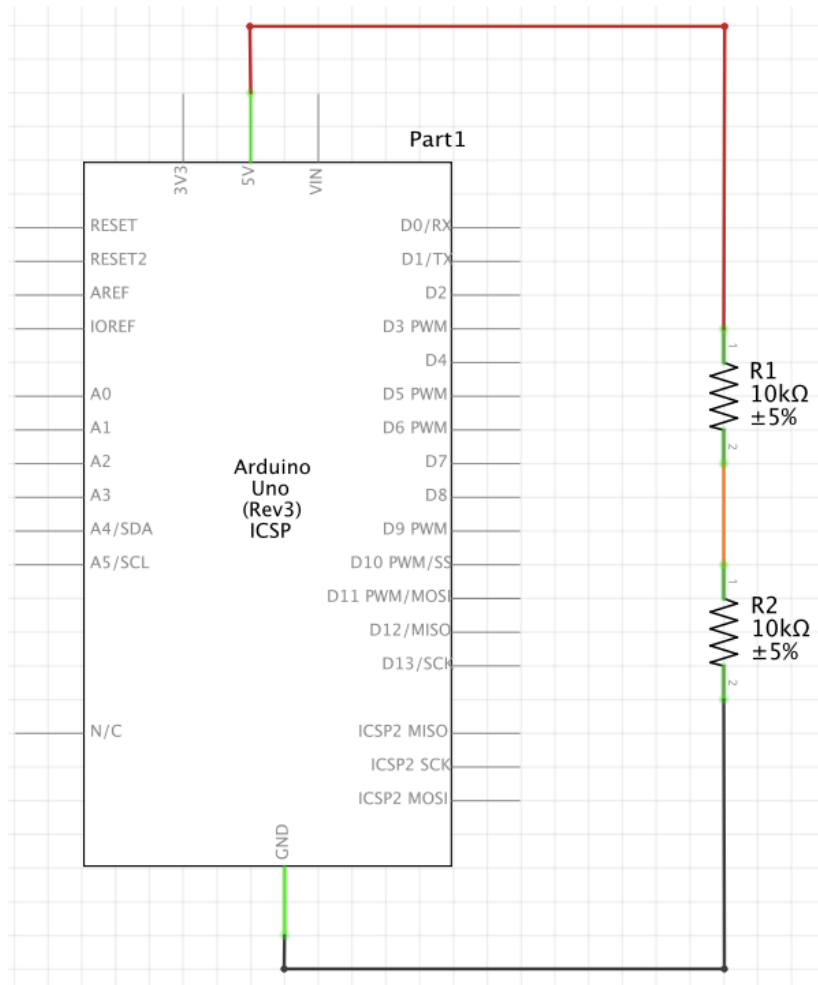
  delay(100);                    // wait
}
```

VOLTAGE DIVIDER



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

VOLTAGE DIVIDER



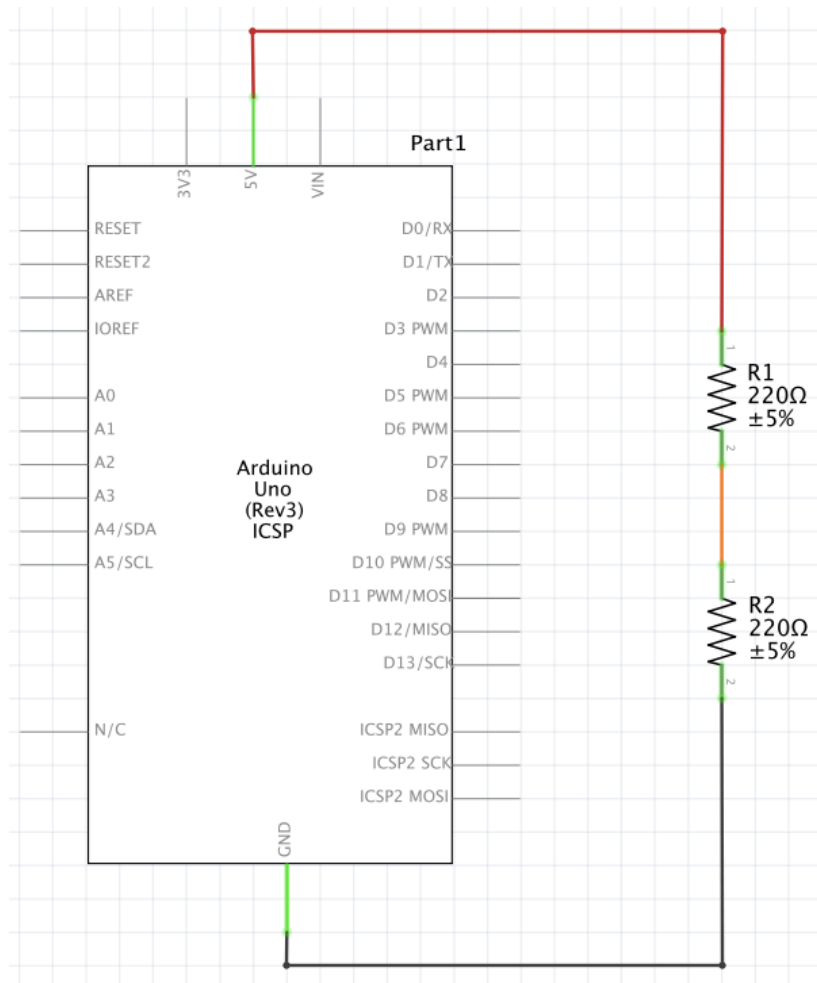
$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

$$R_1 = 10K0HM$$

$$R_2 = 10K0HM$$

$$V_{OUT} = ?$$

VOLTAGE DIVIDER



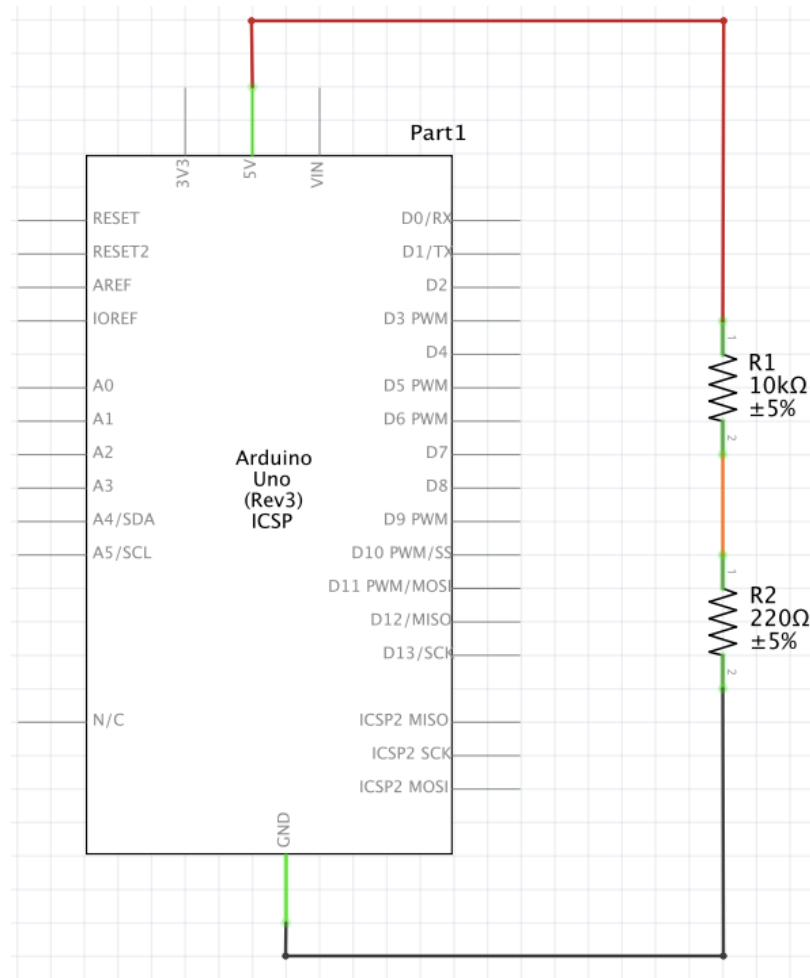
$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

$$R_1 = 2200\text{H}\Omega$$

$$R_2 = 2200\text{H}\Omega$$

$$V_{out} = ?$$

VOLTAGE DIVIDER



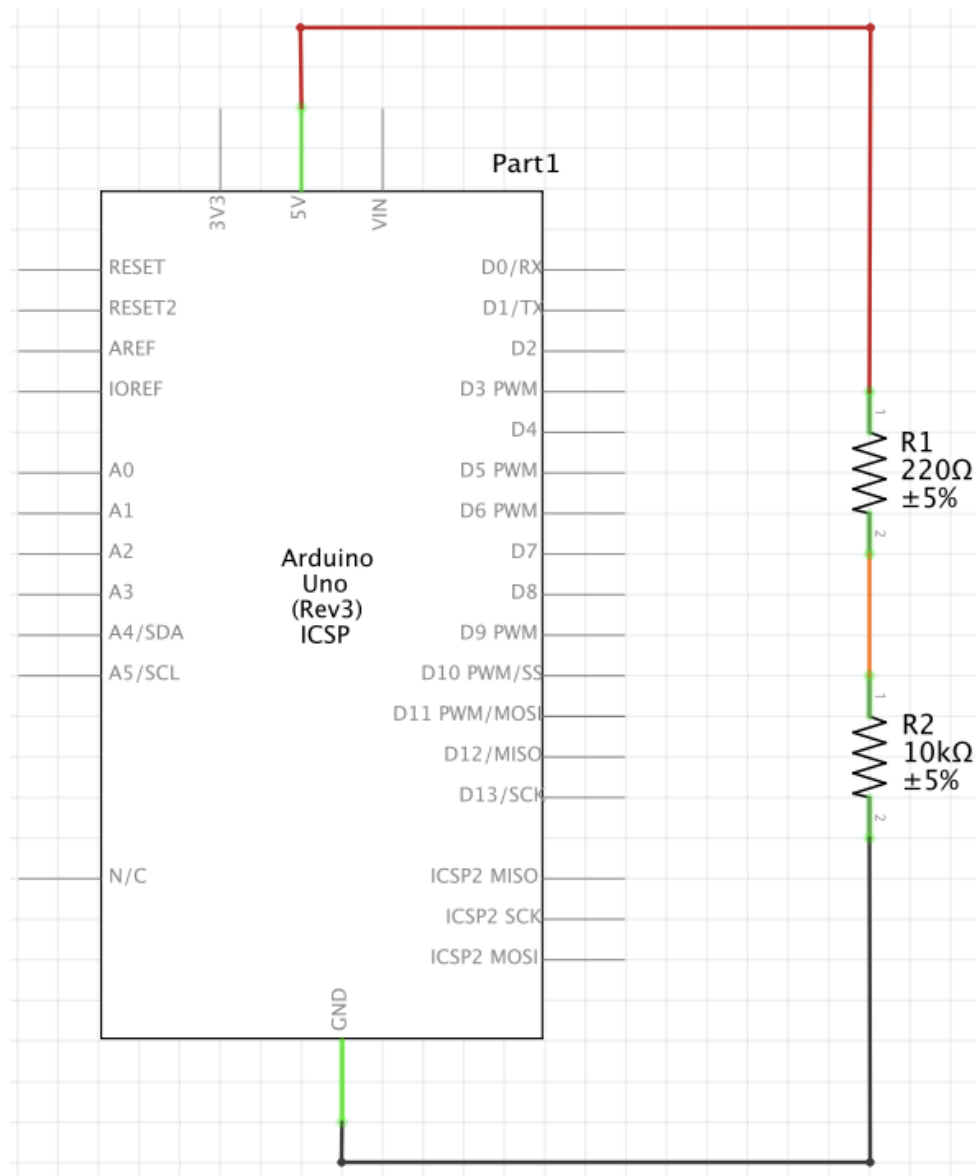
$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

$$R_1 = 10\text{K}\Omega$$

$$R_2 = 220\Omega$$

$$V_{out} = ?$$

VOLTAGE DIVIDER



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

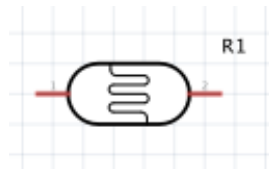
$$R_1 = 2200\text{HM}$$

$$R_2 = 10\text{K}0\text{HM}$$

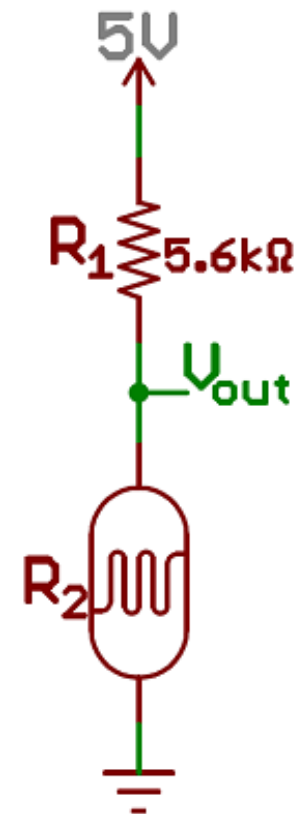
$$V_{OUT} = ?$$

RESISTOR DIVIDER - APPLICATION

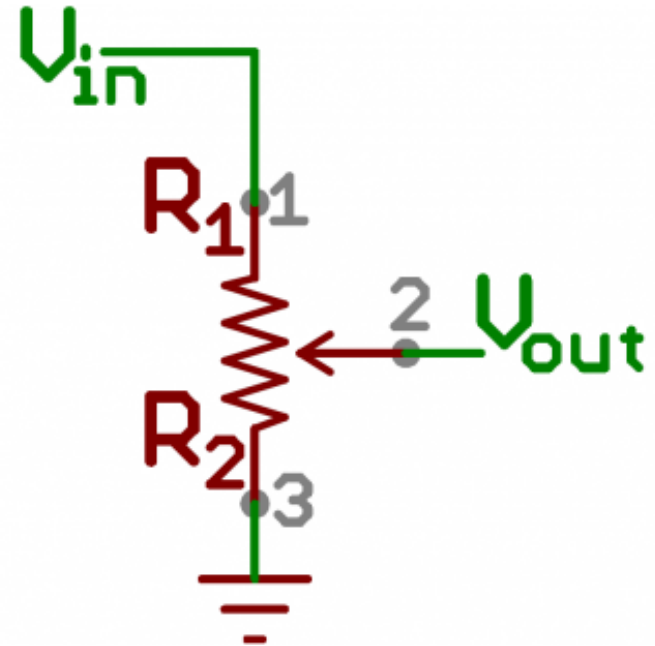
Resistive Sensors



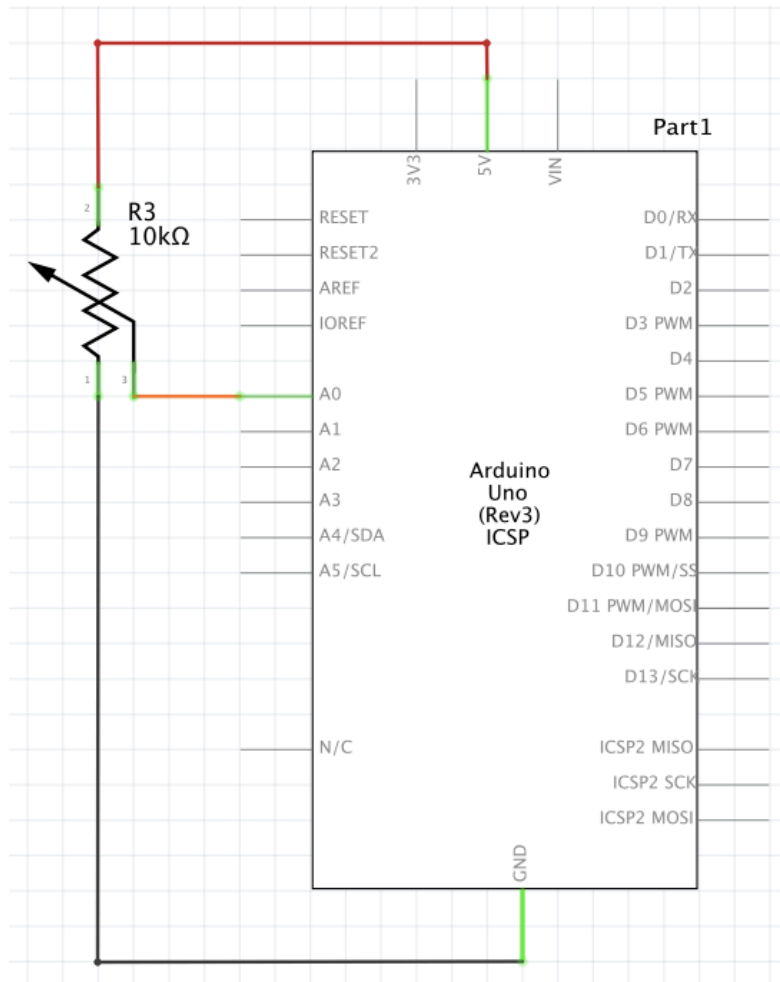
Light Level	R_2 (Sensor)	R_1 (Fixed)	Ratio $R_2/(R_1+R_2)$	V_{out}
Light	$1k\Omega$	$5.6k\Omega$	0.15	0.76 V
Dim	$7k\Omega$	$5.6k\Omega$	0.56	2.78 V
Dark	$10k\Omega$	$5.6k\Omega$	0.67	3.21 V



RESISTOR DIVIDER - APPLICATION

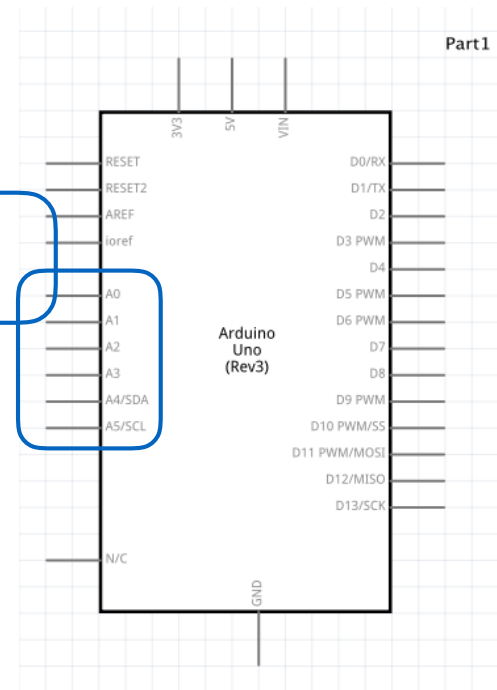


POTENTIOMETER - SCHEMATIC



TO READ A POTENTIOMETER OR ANY SENSOR THAT GENERATE AN ANALOG VOLTAGE => YOU NEED TO USE THE ANALOG PINS.

Analog Pins:
A0...A5



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
}
```

/dev/cu.usbmodem1411 (Arduino/Genuino Uno)

Send

The value of the potentiometer is: 587
The value of the potentiometer is: 598
The value of the potentiometer is: 597
The value of the potentiometer is: 597
The value of the potentiometer is: 602
The value of the potentiometer is: 619
The value of the potentiometer is: 639
The value of the potentiometer is: 653
The value of the potentiometer is: 663
The value of the potentiometer is: 663
The value of the potentiometer is: 663
The value of the potentiometer is: 663
The value of the potentiometer is: 663
The value of the potentiometer is: 663
The value of the potentiometer is: 663

☐ Autoscroll

No line ending

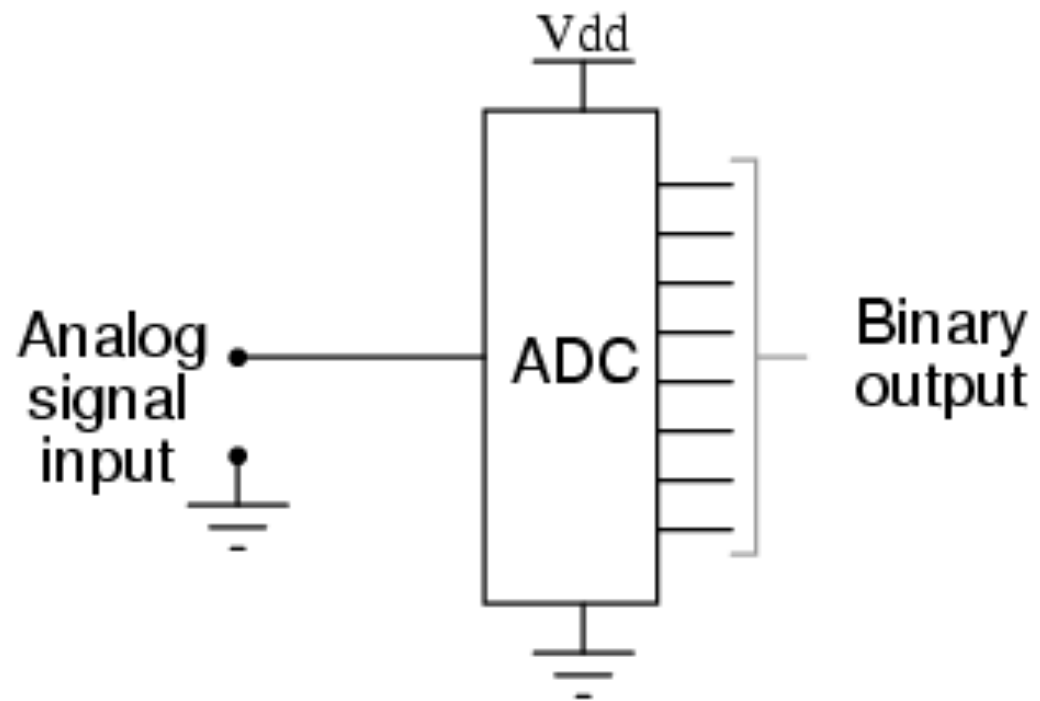
9600 baud

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() {

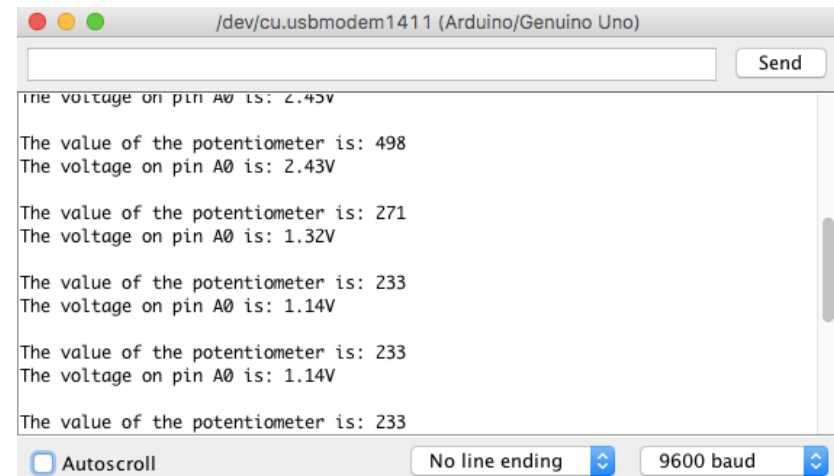
  pot_value = analogRead(pot_pin); //read the push button status

  pot_voltage = pot_value * (5.0 / 1023.0);

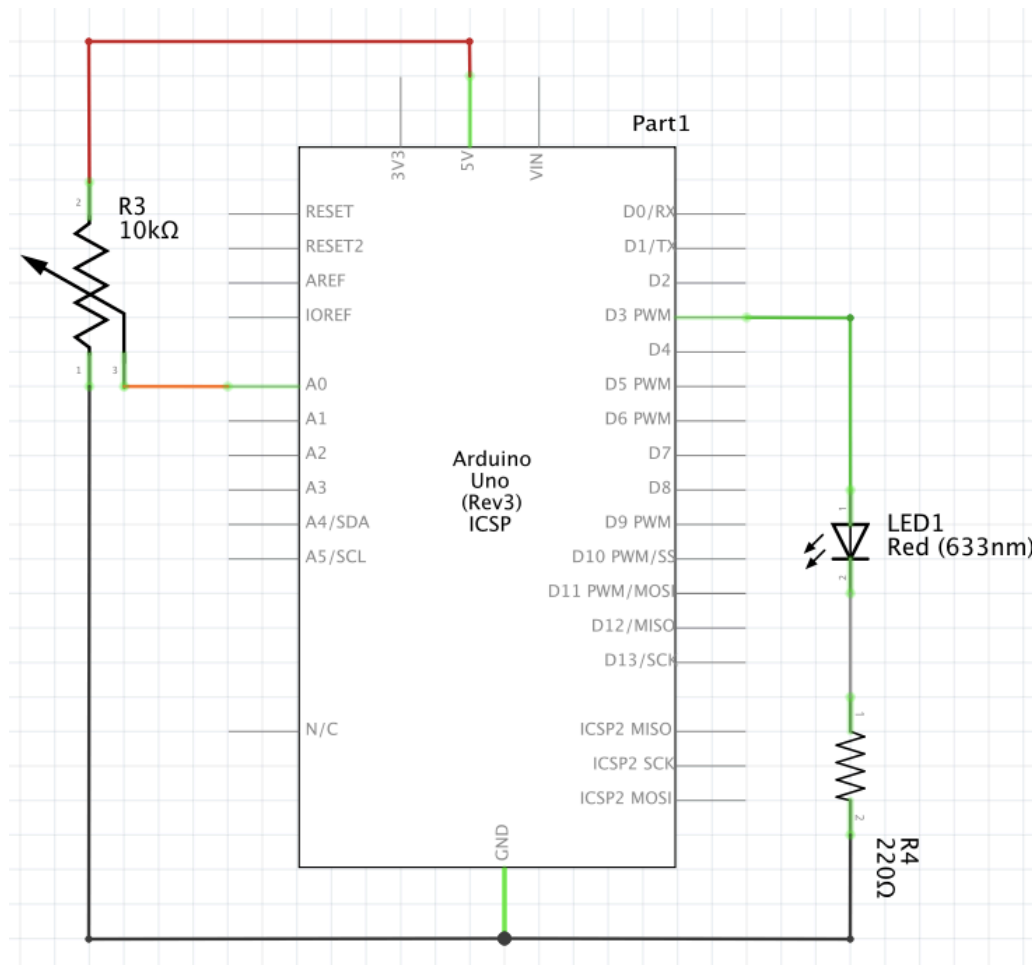
  Serial.print("The value of the potentiometer is: ");
  Serial.println(pot_value);
  Serial.print("The voltage on pin A0 is: ");
  Serial.println(pot_voltage);
  Serial.println("");

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

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



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

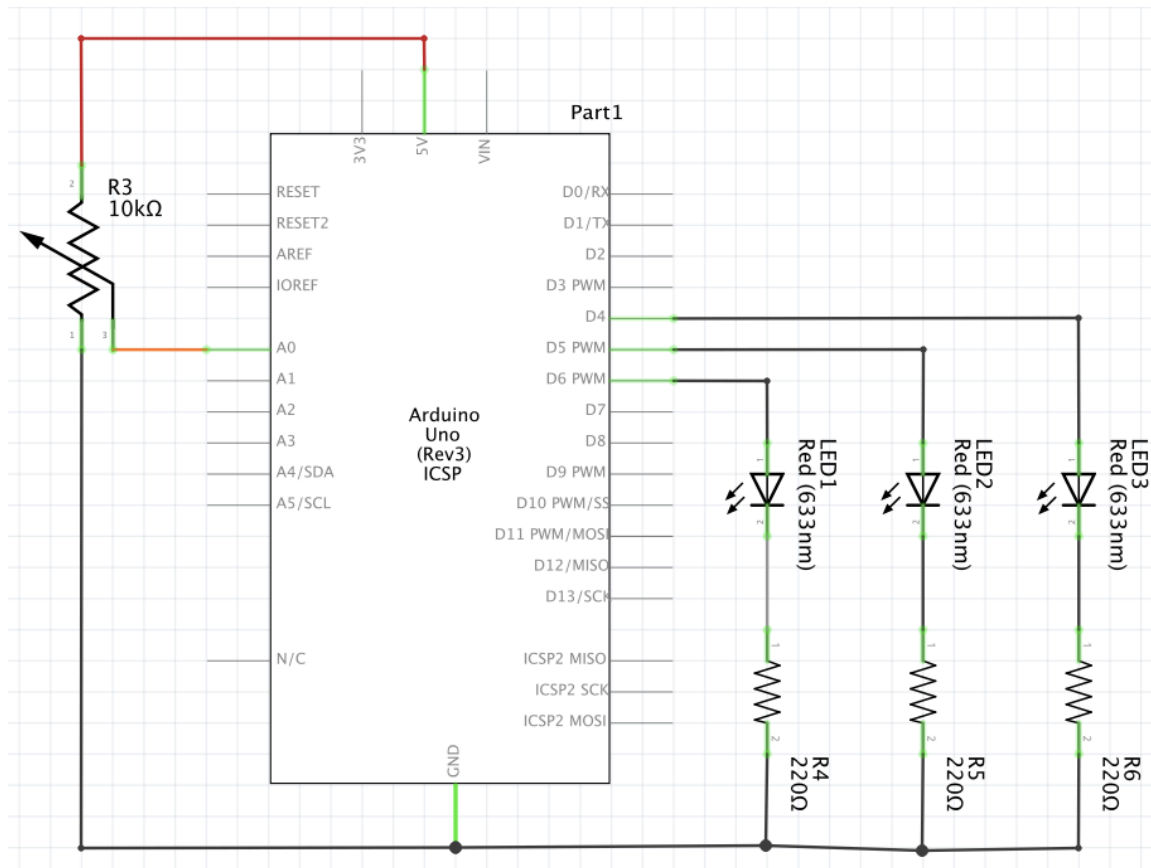
void setup() {
    // declare the ledPin as an OUTPUT and the potentiometer as INPUT
    pinMode(led_pin, OUTPUT);
    pinMode(pot_pin, INPUT);
}

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 - SCHEMATIC



USE THE POTENTIOMETER
TO SELECT WHICH LED
WILL TURN ON

ASSIGNMENTS

1) Read and understand the different kinds of potentiometers. Select two different projects/products that implement two different potentiometers. Write a post about them and explain how the input devices are used.

2) Design and program a circuit with the following specifications:

- at least one tact switch
- at least one potentiometer or a LDR
- at least four LEDs
- use the functions digitalRead/analogRead

If you need ideas about the functionality of the circuit, look at the following slide.

3) Document the project of point 2) in your blog with, the documentation includes:

- a short description of the functionality of the circuit
- schematic of the circuit
- sketch (arduino code)
- if it works: upload a video of the circuit that works
- if it doesn't work: explain what you did to debug hardware and software

ASSIGNMENTS

If you need inspiration about the functionality of the circuit of assignment 2, you can implement the following examples:

- with the potentiometer you select the LED that has to be ON. You turn OFF all the LED only when the push button is pressed.
- all the LEDs blink only if the tact switch is pressed and with the potentiometer you select how fast the LED blinks.

SOURCES AND LICENCE

Arduino software, If, If...else, comparison operators, analogRead, map:

<https://www.arduino.cc/en/Reference/If>

<https://www.arduino.cc/en/Reference/Else>

<https://www.arduino.cc/en/Reference/AnalogRead>

<https://www.arduino.cc/en/Reference/Map>

Pull-UP:

<https://learn.sparkfun.com/tutorials/pull-up-resistors>

Voltage Divider:

<https://learn.sparkfun.com/tutorials/voltage-dividers>

ADC:

<https://learn.sparkfun.com/tutorials/analog-to-digital-conversion>

Potentiometer (pg. 89)

Encyclopedia of Electronic Components Volume 1, 1st Edition,
Charles Platt.

LDR:

<https://learn.sparkfun.com/tutorials/photocell-hookup-guide>
<https://learn.adafruit.com/photocells/using-a-photocell>

L I C E N C E

L I C E N C E

EXCEPT WHERE OTHERWISE NOTED, THIS WORK IS LICENSED UNDER:
[HTTPS://CREATIVECOMMONS.ORG/LICENSES/BY/4.0/](https://creativecommons.org/licenses/by/4.0/)

