

#FABRICADEMY2020 TUTORIALS

Arduino and friends

ELECTRONIC BASICS TUTORIAL

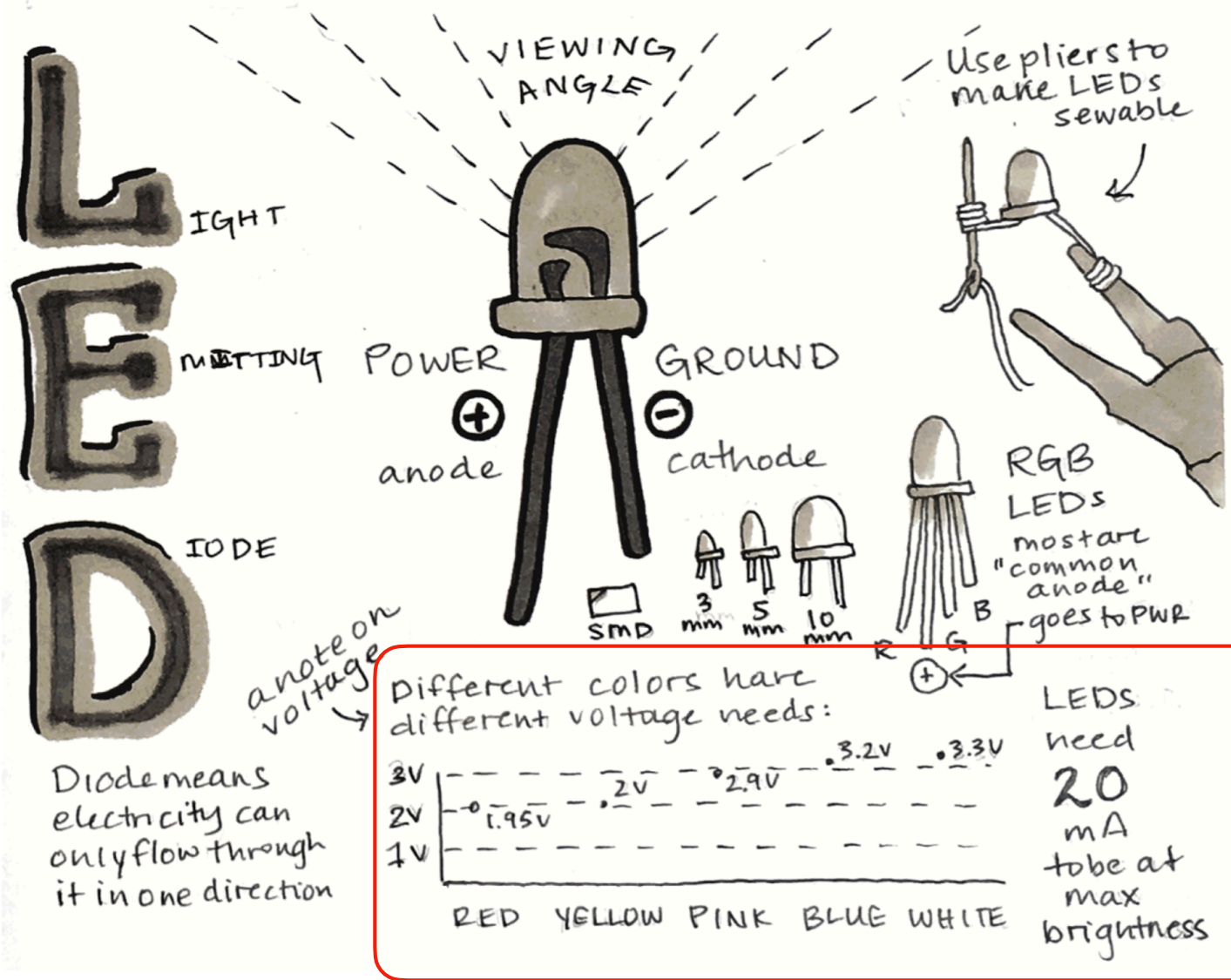
EMMA PARESCHI
21-22 / 10 / 2020

Fabricademy | textile & technology academy 2020



LED

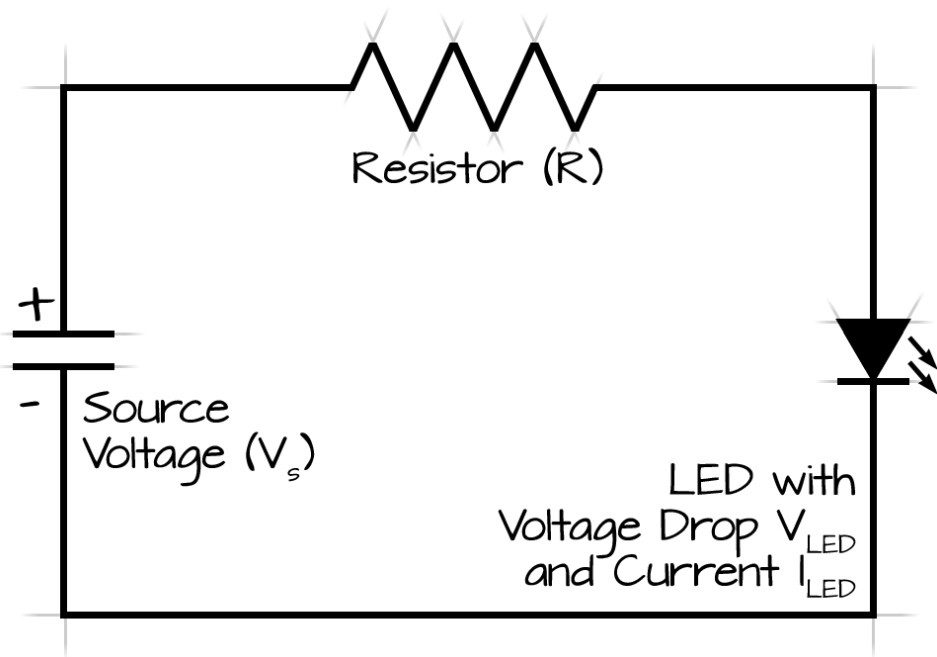
COMPONENT



<https://learn.sparkfun.com/tutorials/light-emitting-diodes-leds/all>

Led

$$R = \frac{(V_s - V_{LED})}{I_{LED}}$$



Different colors have different voltage needs:

3V	2V	1V	1.95V	2V	2.9V	3.2V	3.3V
RED	YELLOW	PINK	BLUE	WHITE			

LEDs need 20 mA to be at max brightness

Voltage Source (V_s)	<input type="text" value="source voltage"/>	Volts (V)
Voltage drop from LED (V_{LED})	<input type="text" value="LED voltage"/>	Volts (V)
Current through the LED (I_{LED})	<input type="text" value="LED current"/>	milliamps (mA) ▾
Resistance (R)	<input type="text" value="resistance"/>	ohms (Ω) ▾
LED Power (P)	<input type="text" value="LED power"/>	Watts (W)
<input type="button" value="Calculate"/>		Click "Calculate" to update the fields with orange borders.

<https://ohmslawcalculator.com/led-resistor-calculator>

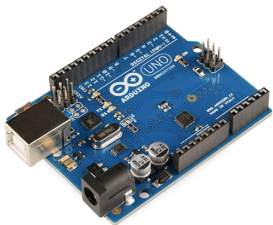
Arduino

“**Physical Computing** is about prototyping with electronics, turning sensors, actuators and microcontrollers into materials for designers and artists.”

“It involves the design of interactive objects that can communicate with humans using sensors and actuators controlled by a behaviour implemented as software running inside a microcontroller.”

Massimo Banzi, Tinker.it & Arduino Co-Founder

Hardware



Software

```
Arduino IDE - Blink | Arduino 1.6.5
Blink
pin -> [pin]
This example code is in the public domain.
modified 8 May 2014
by Simon St Laurent

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize the LED pin as: OUTPUT
  pinMode(LED_PIN, OUTPUT);
}

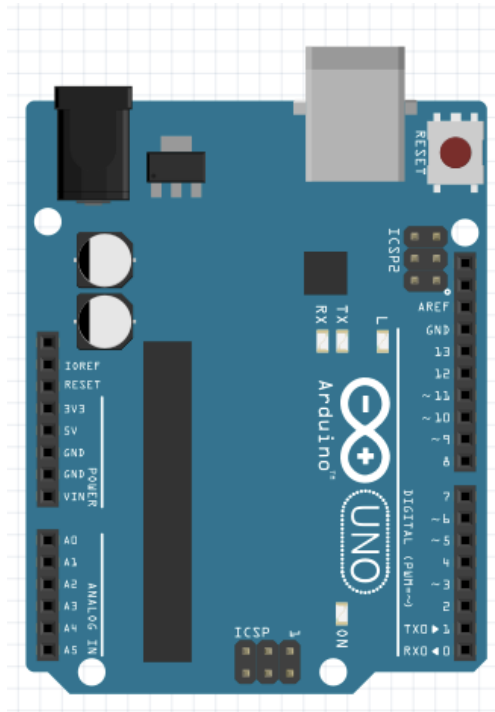
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_PIN, HIGH); // turn the LED on (HIGH is the voltage
  digitalWrite(LED_PIN, LOW); // turn the LED off by making the voltage
  delay(1000); // wait for a second.
  digitalWrite(LED_PIN, HIGH); // turn the LED on by making the voltage
  delay(1000); // wait for a second.
}
```

Community

Category	Posts	Topics
Installation & Troubleshooting For problems with Arduino itself, NOT your project Last post: Today at 07:20 am by topshark	67,313	15,896
Project Guidance Advice on general approaches or feasibility Last post: Today at 09:33 am by ArduinoShieldL293D by thummer100	328,263	45,241
Programming Questions Understanding the language, error messages, etc. Last post: Today at 08:54 am by Problem with EEPROM... by ym_65	455,119	55,397
General Electronics Resistors, capacitors, breadboards, soldering, etc. Last post: Today at 09:20 am by Arduino the right one... by democox	159,694	16,984

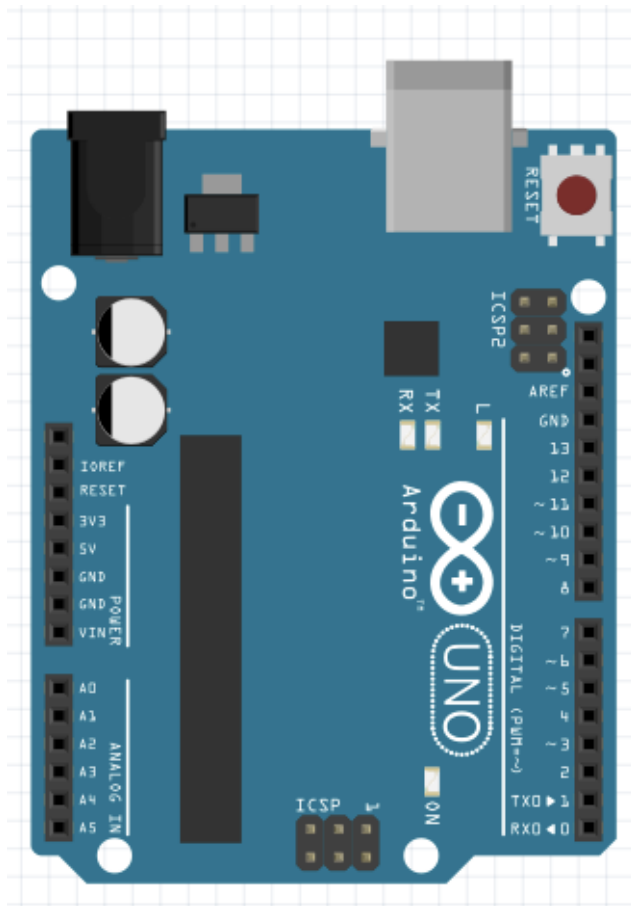
Arduino

Sensor
(input device)



Led
(output device)

Arduino today



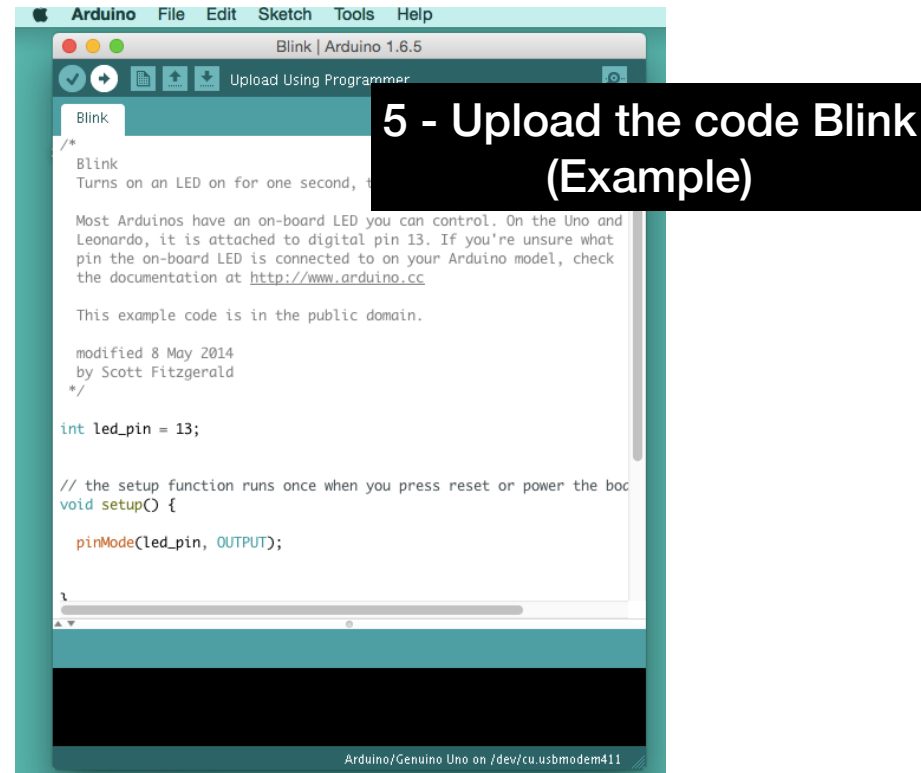
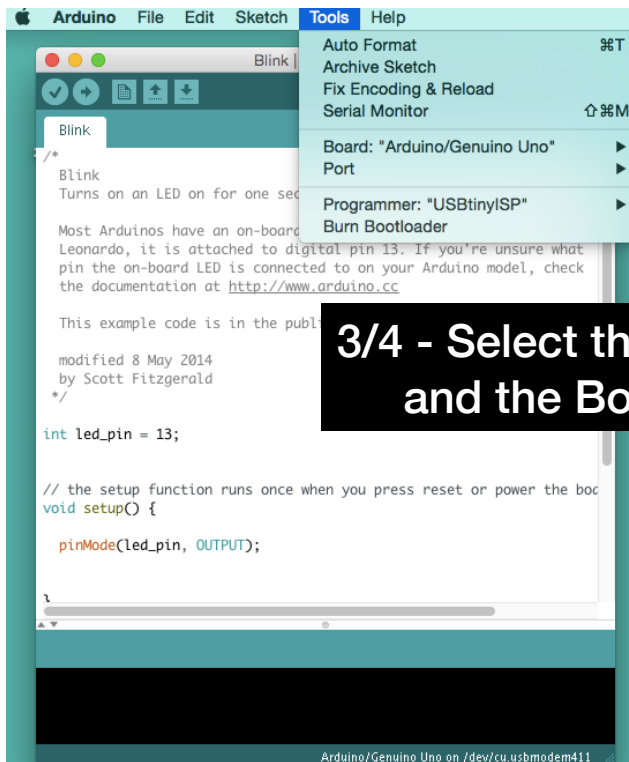
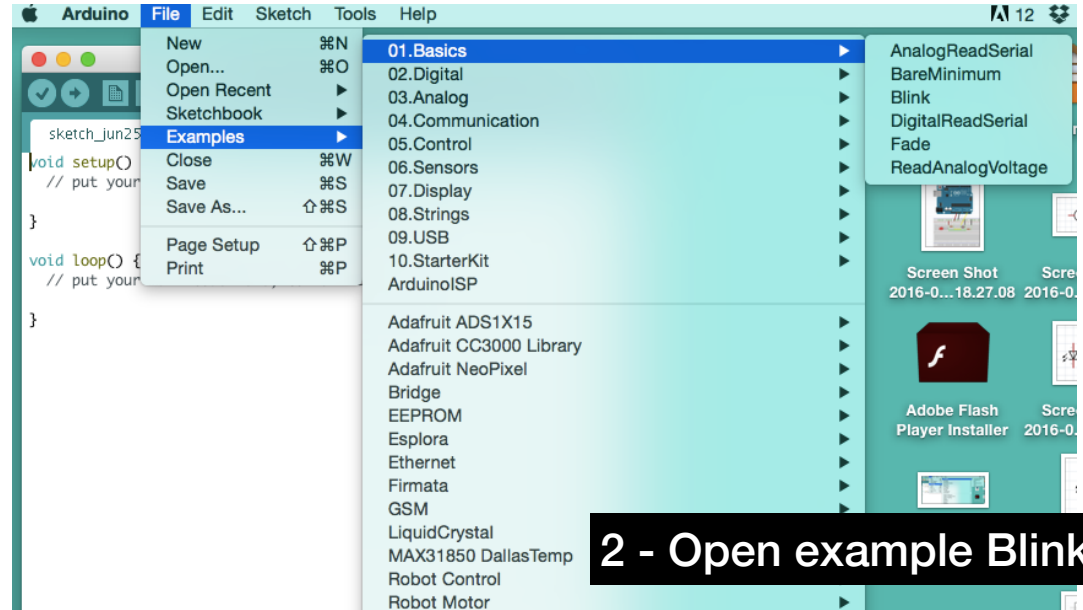
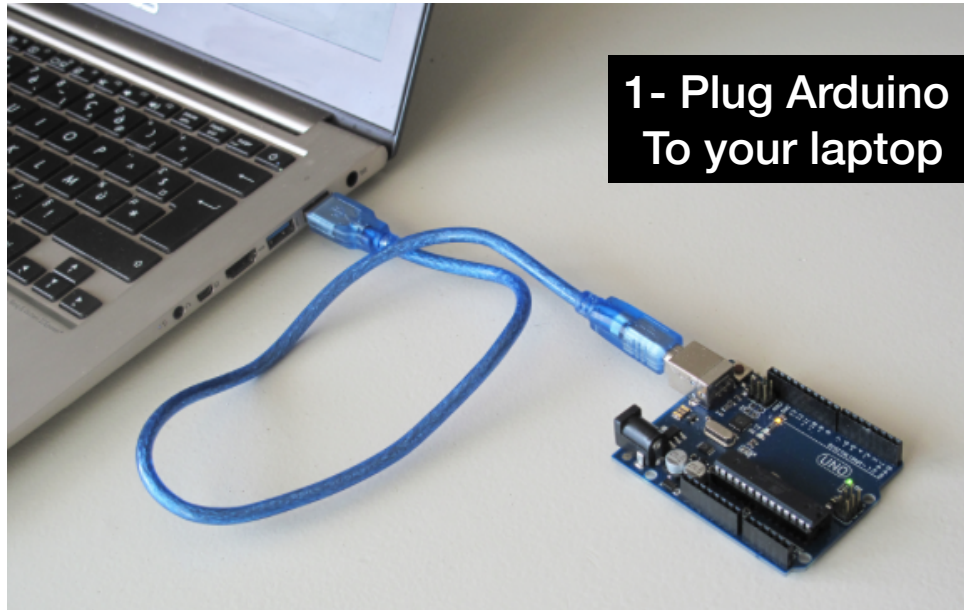
How we will use Arduino

- control a Led
- read a sensor
- to control a Led based on the interaction with the sensor

What do we need:

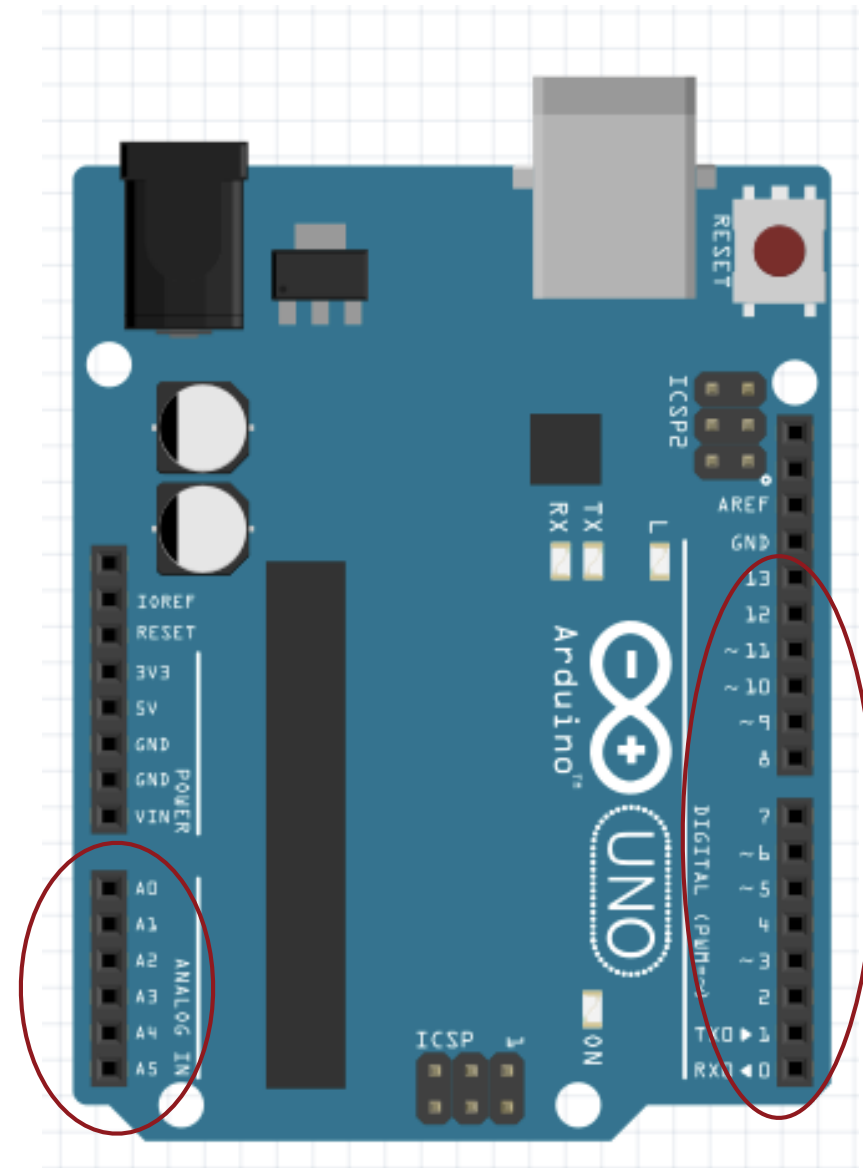
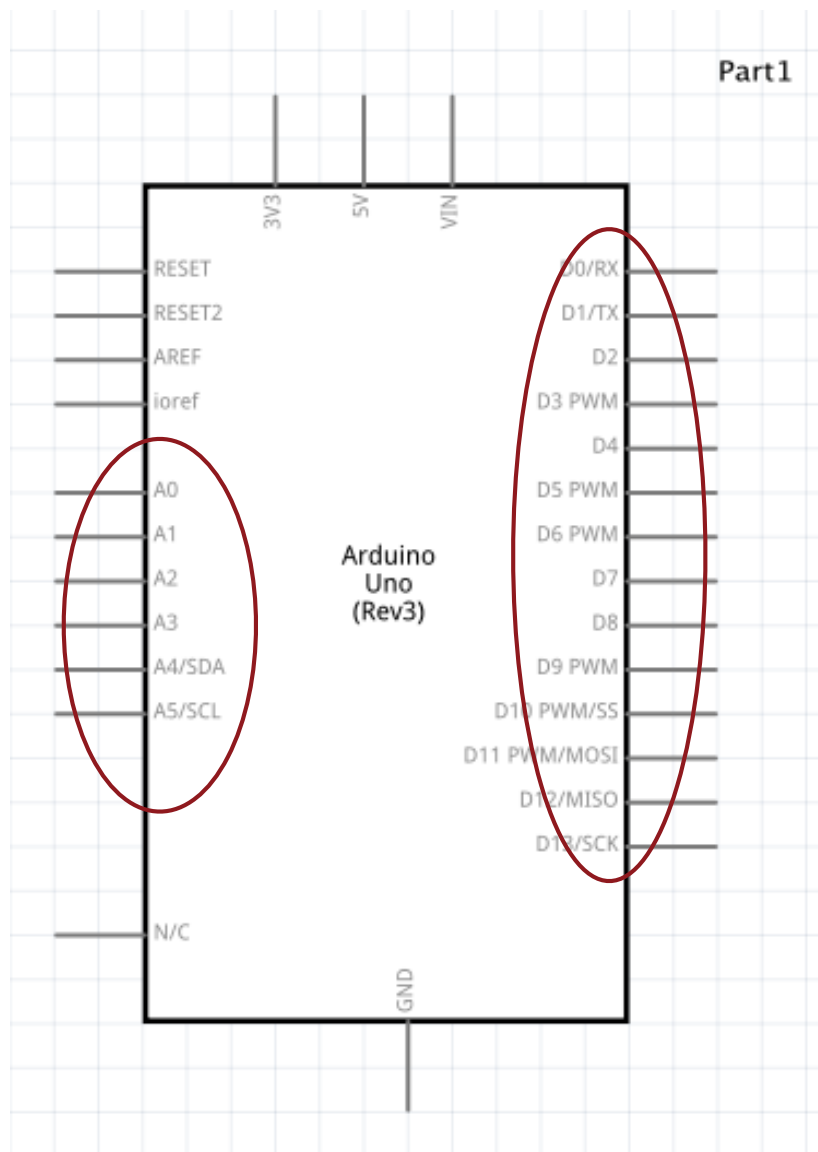
- make a circuit (with Arduino)
- 'write' (use the right) the code for the circuit

...check if the environment works

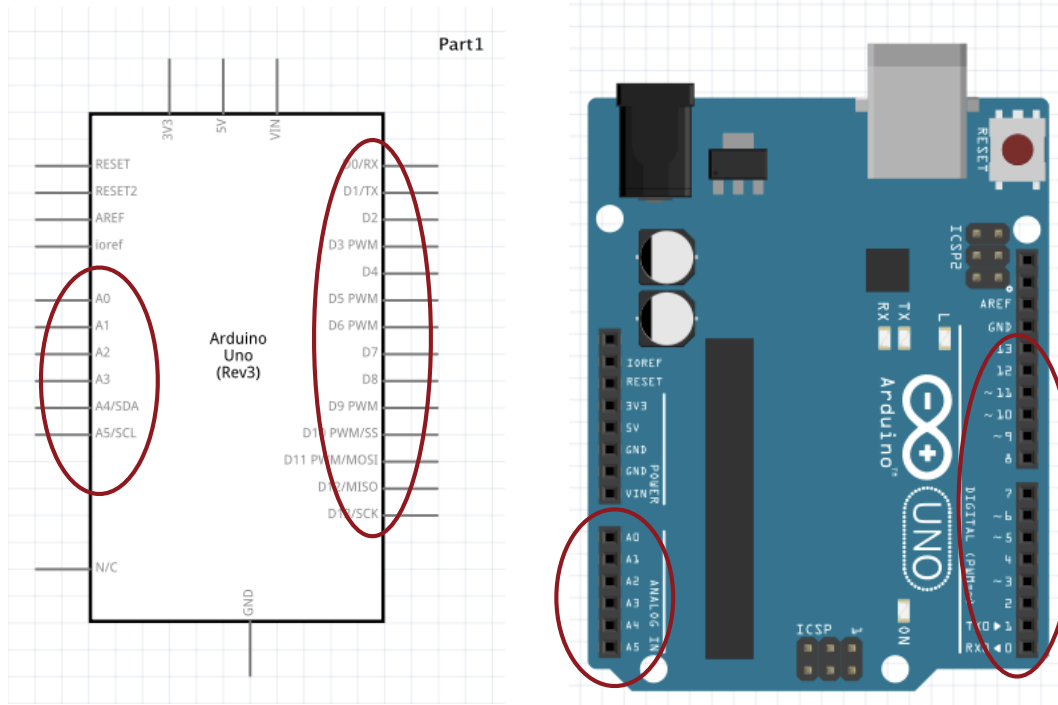


If it works, the led embedded on the board blinks

Arduino - Input/Output pins



Arduino - Input/Output pins



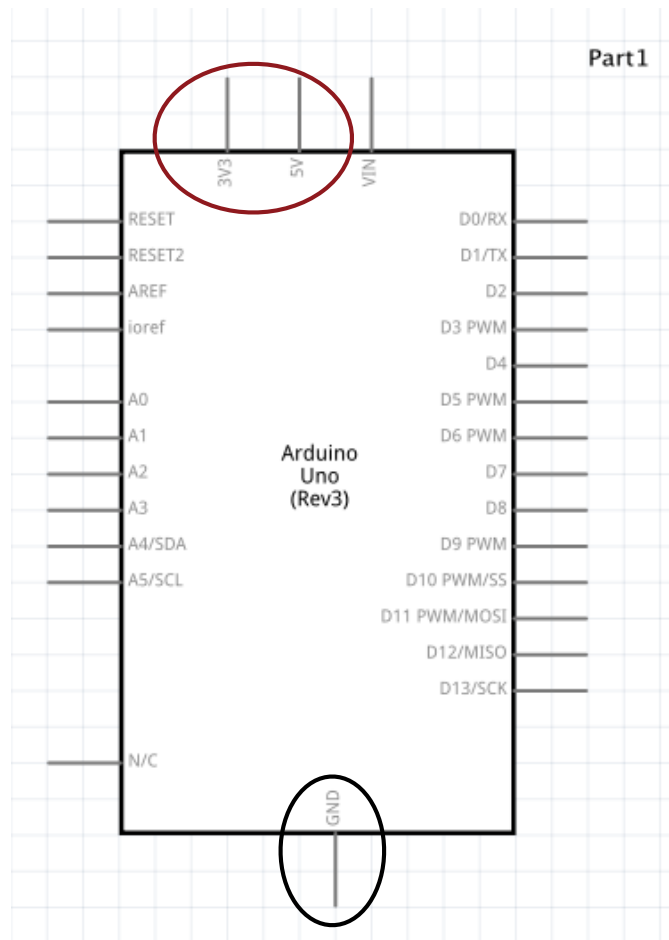
With the INPUT - OUTPUT pins you can:

- Read Voltage
- Control Voltage

Like we did yesterday with the MULTIMETER!!!!

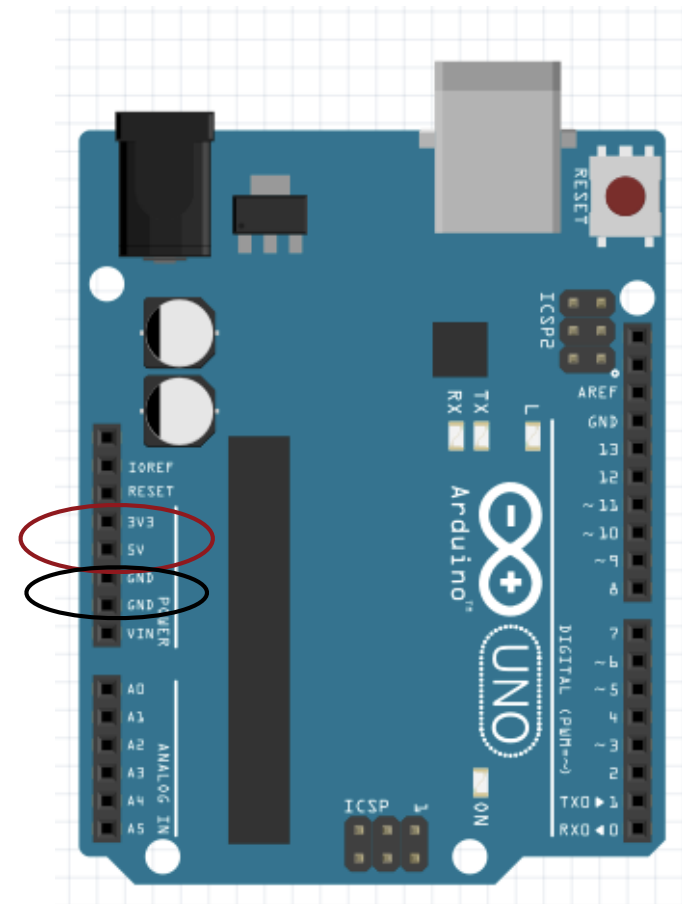
Arduino - Power Pins

Power Pins

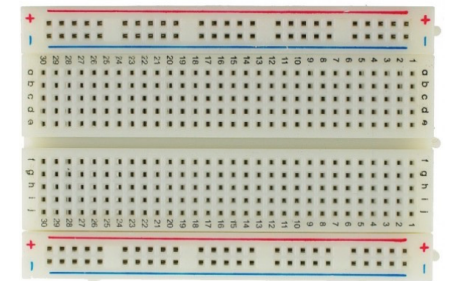
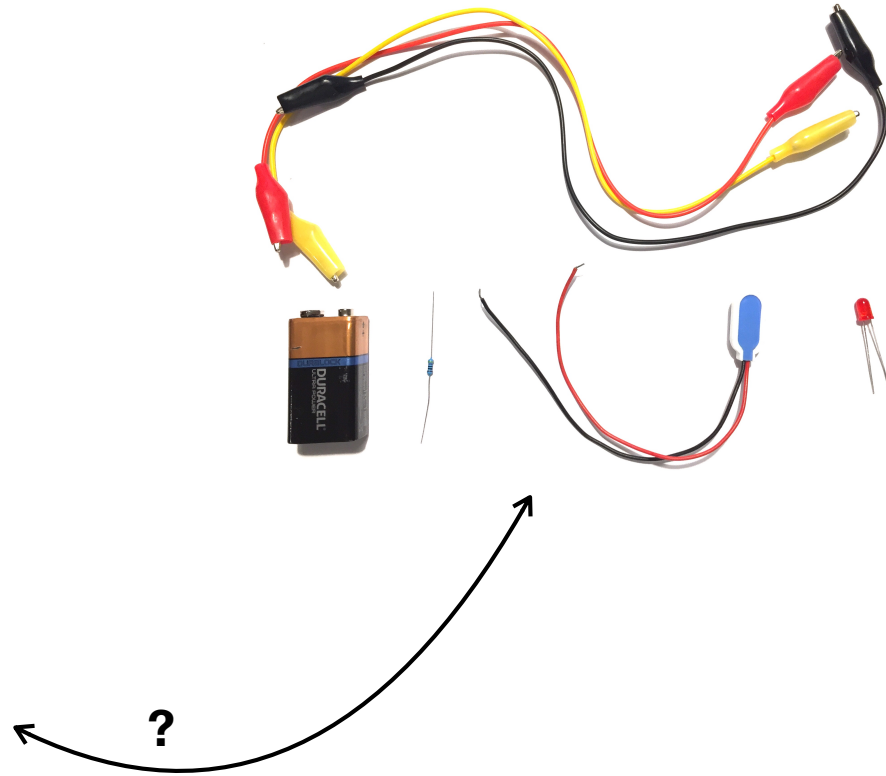
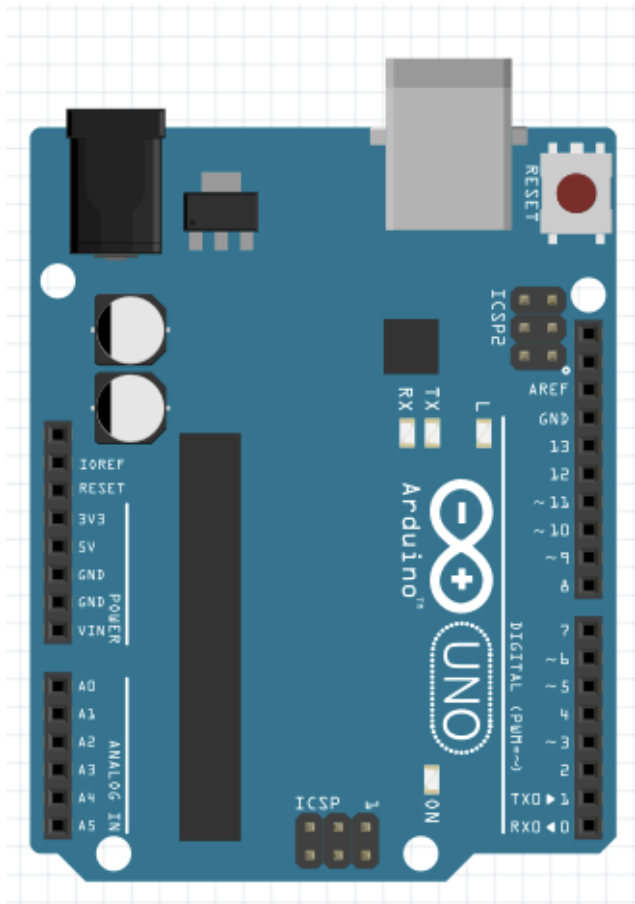


5V / 3.3V:
Power

GND:
Ground Pin

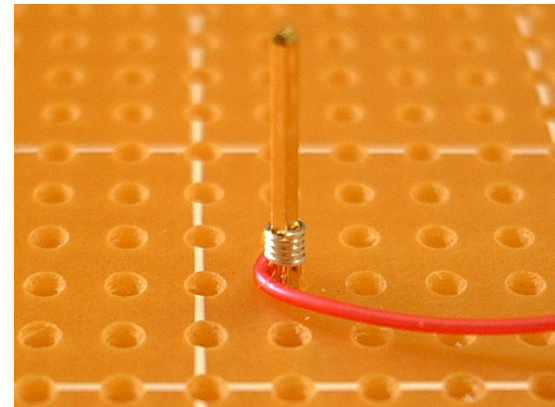
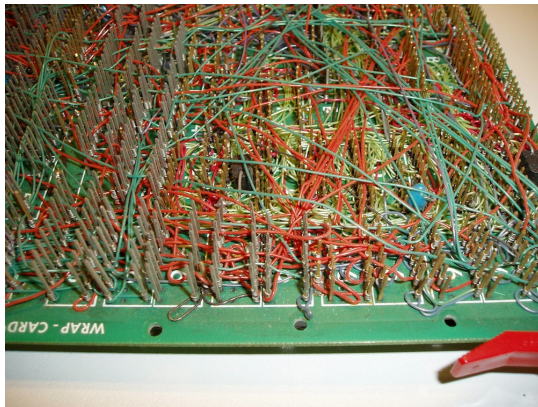


To make the circuit: tools

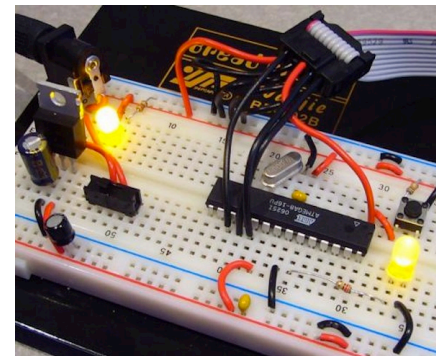
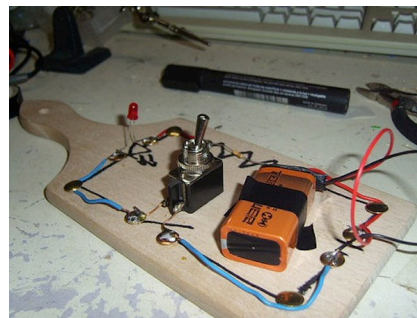


BreadBoard

Wire - Wrap



Breadboard



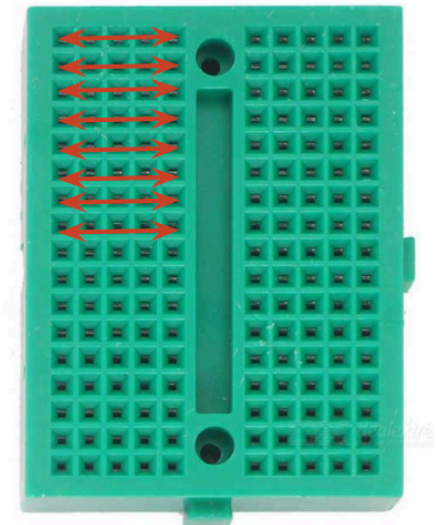
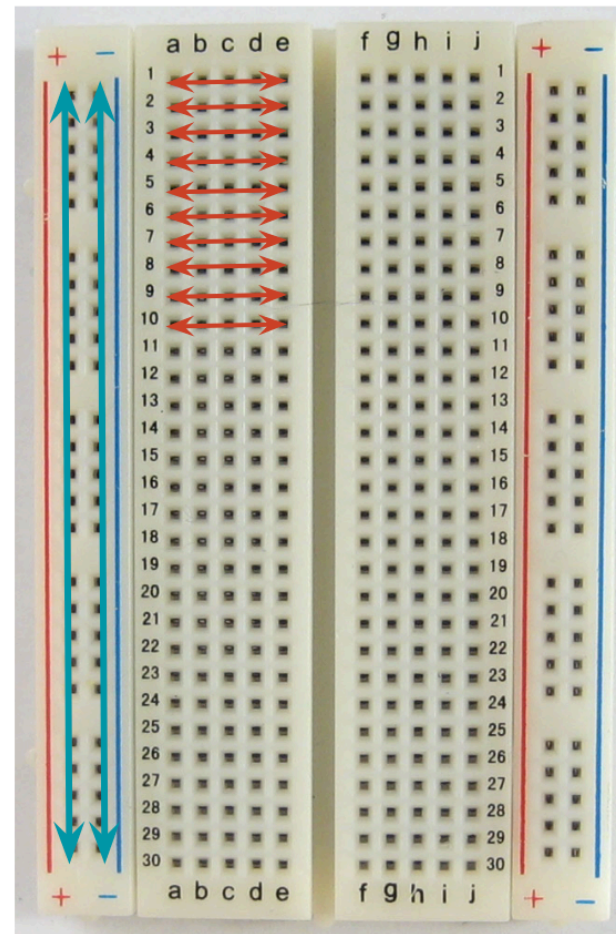
Use to prototype your circuit first.

Vertical lines are connected

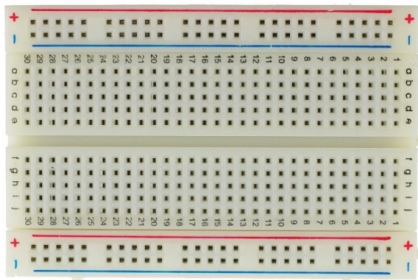
Horizontal lines are connected

BREADBOARDS

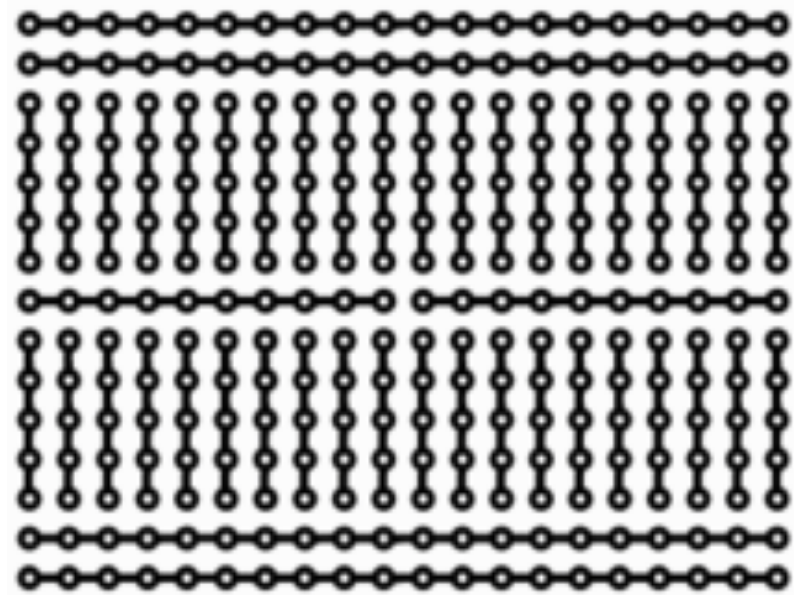
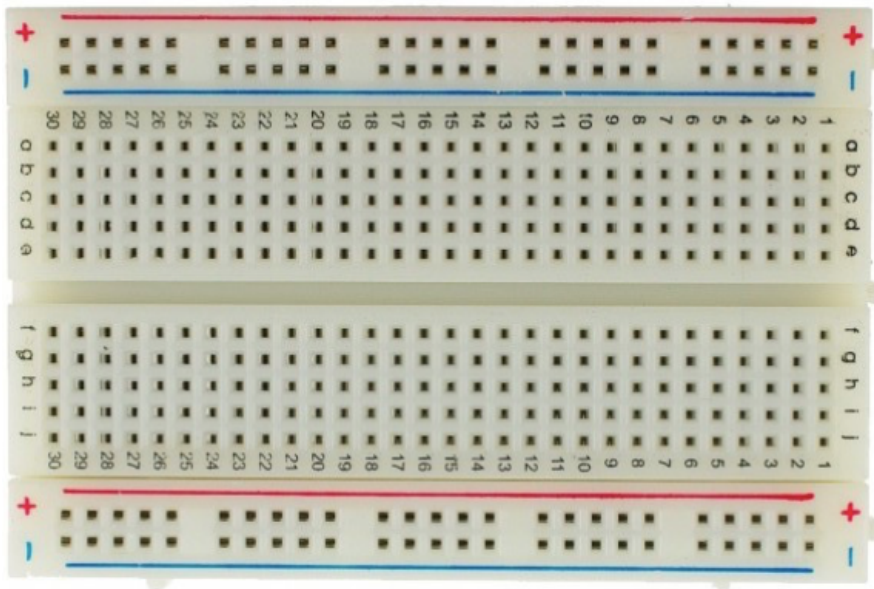
COMPONENTS+TOOLS



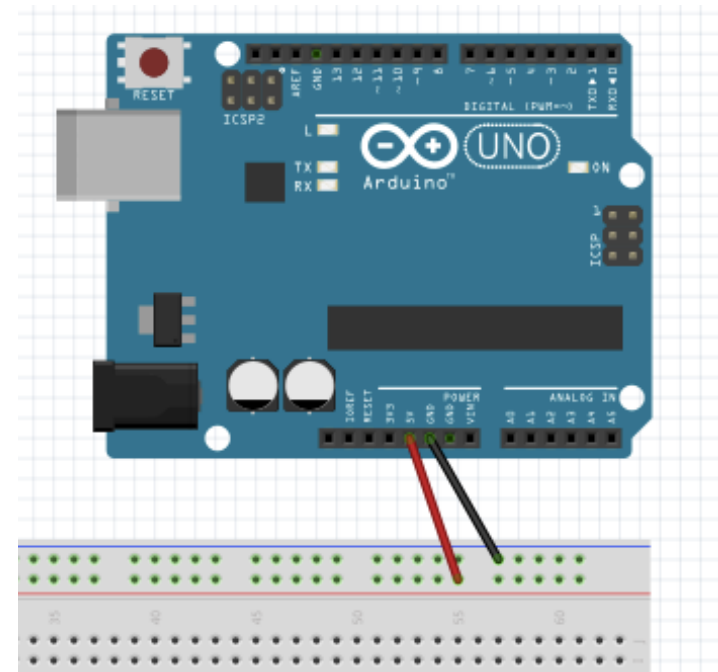
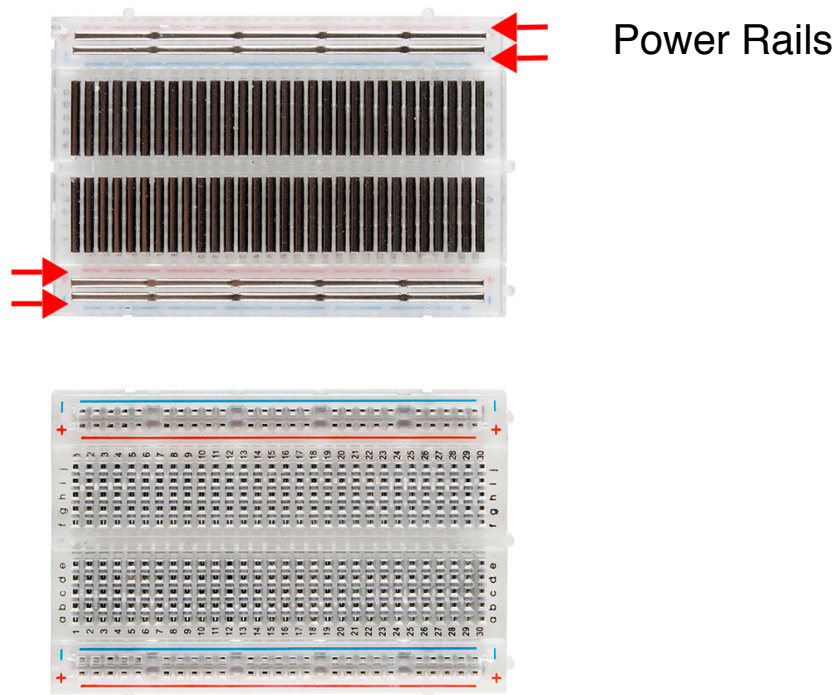
BreadBoard



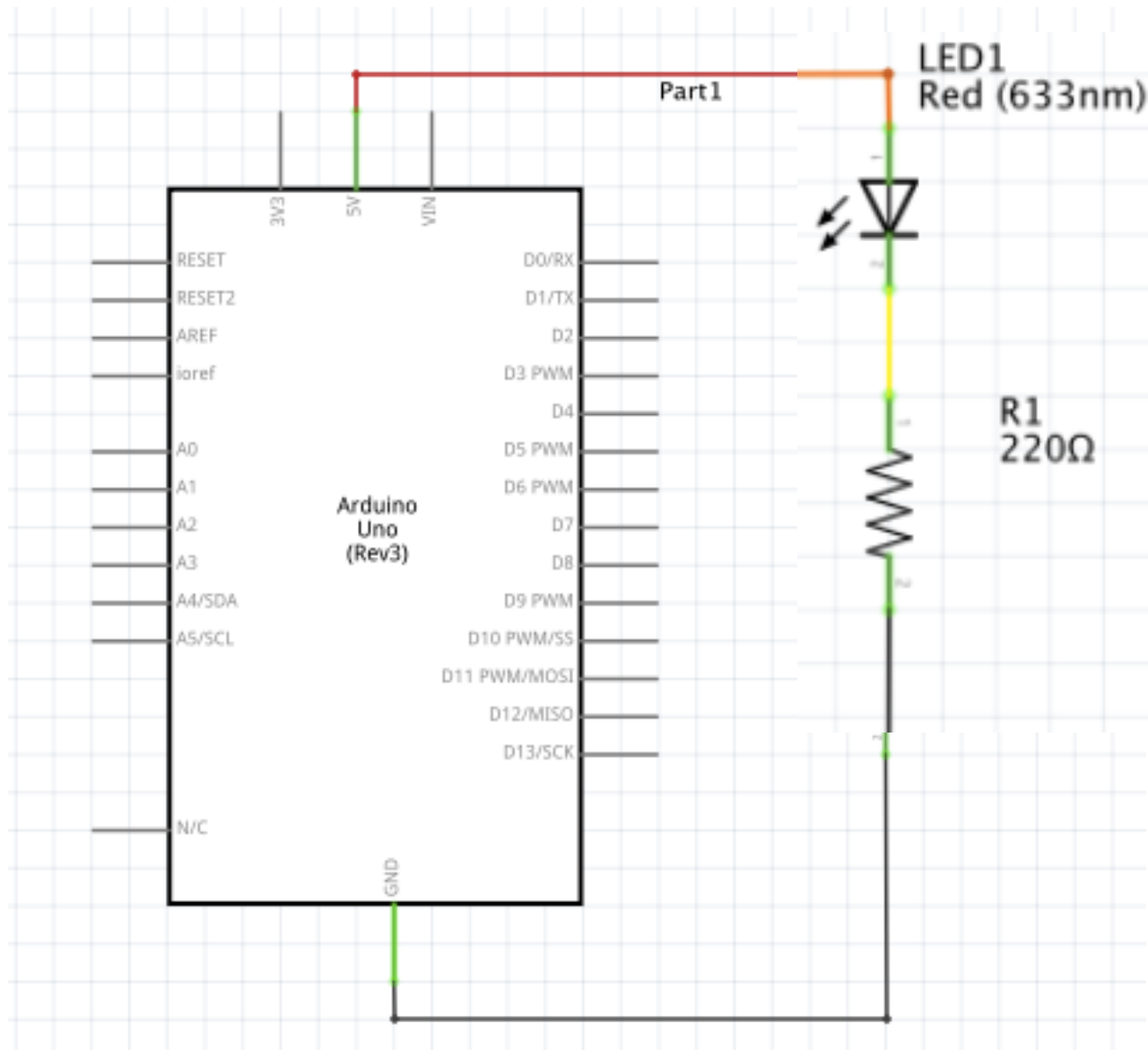
It is a physical support for making temporary circuits and prototyping, and they require absolutely no soldering.



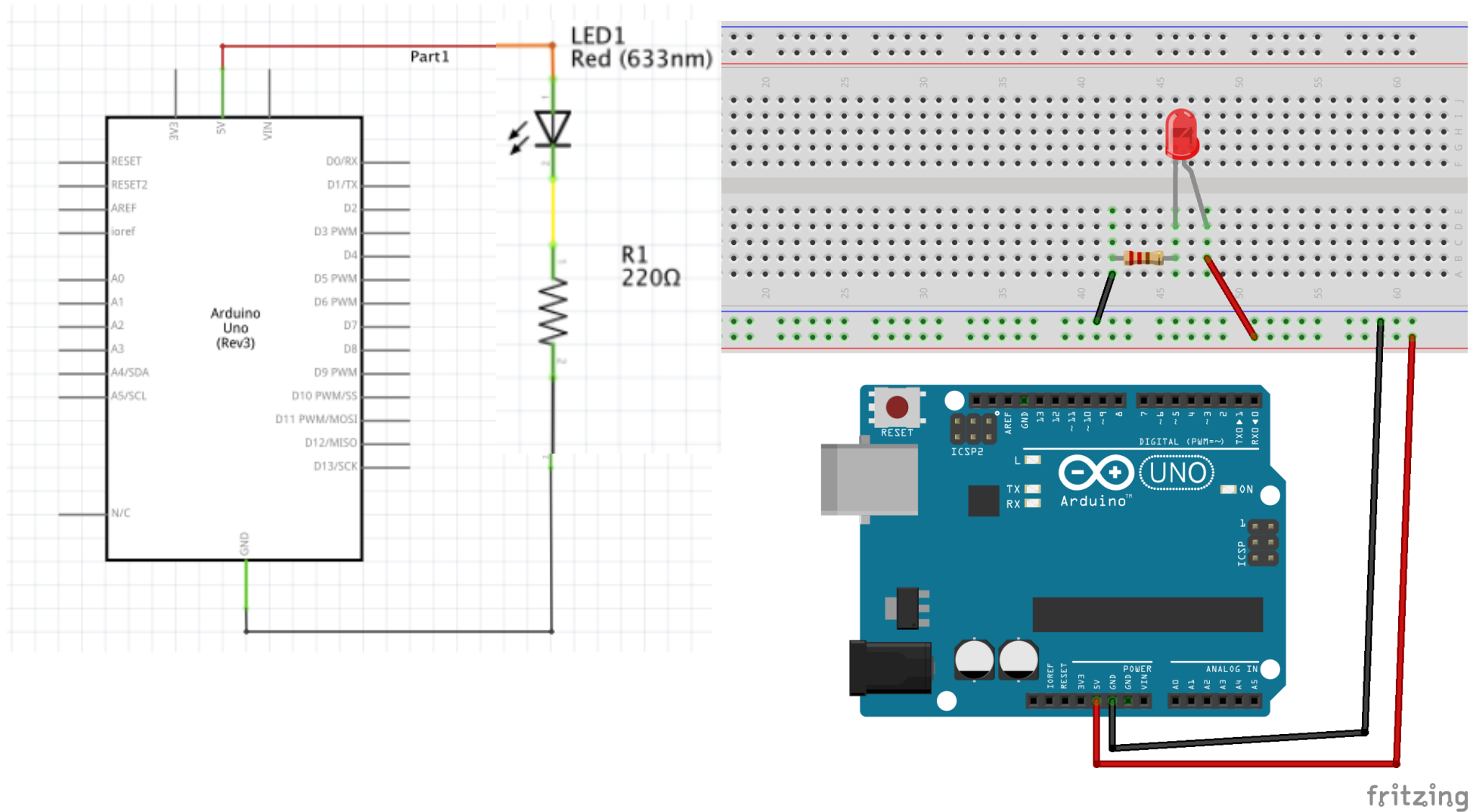
BreadBoard - best practice



Exercise: Led ON



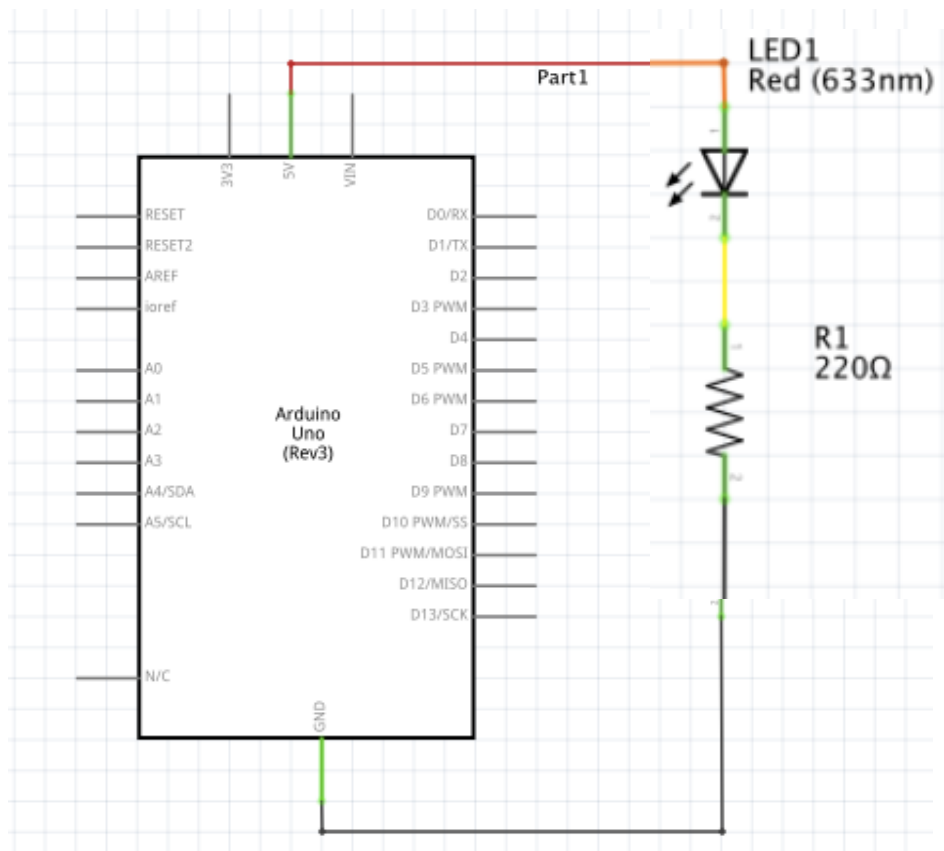
Exercise: Led ON



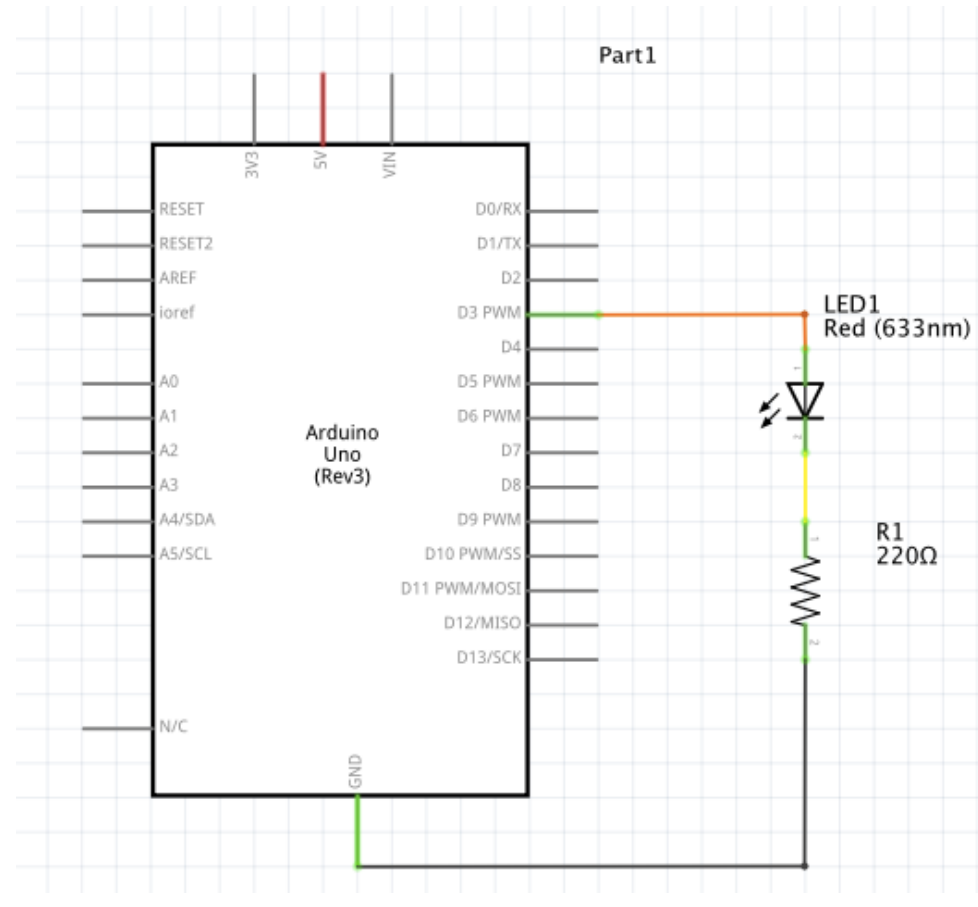
Arduino: Led

Make a Led blinks

Arduino and a Led

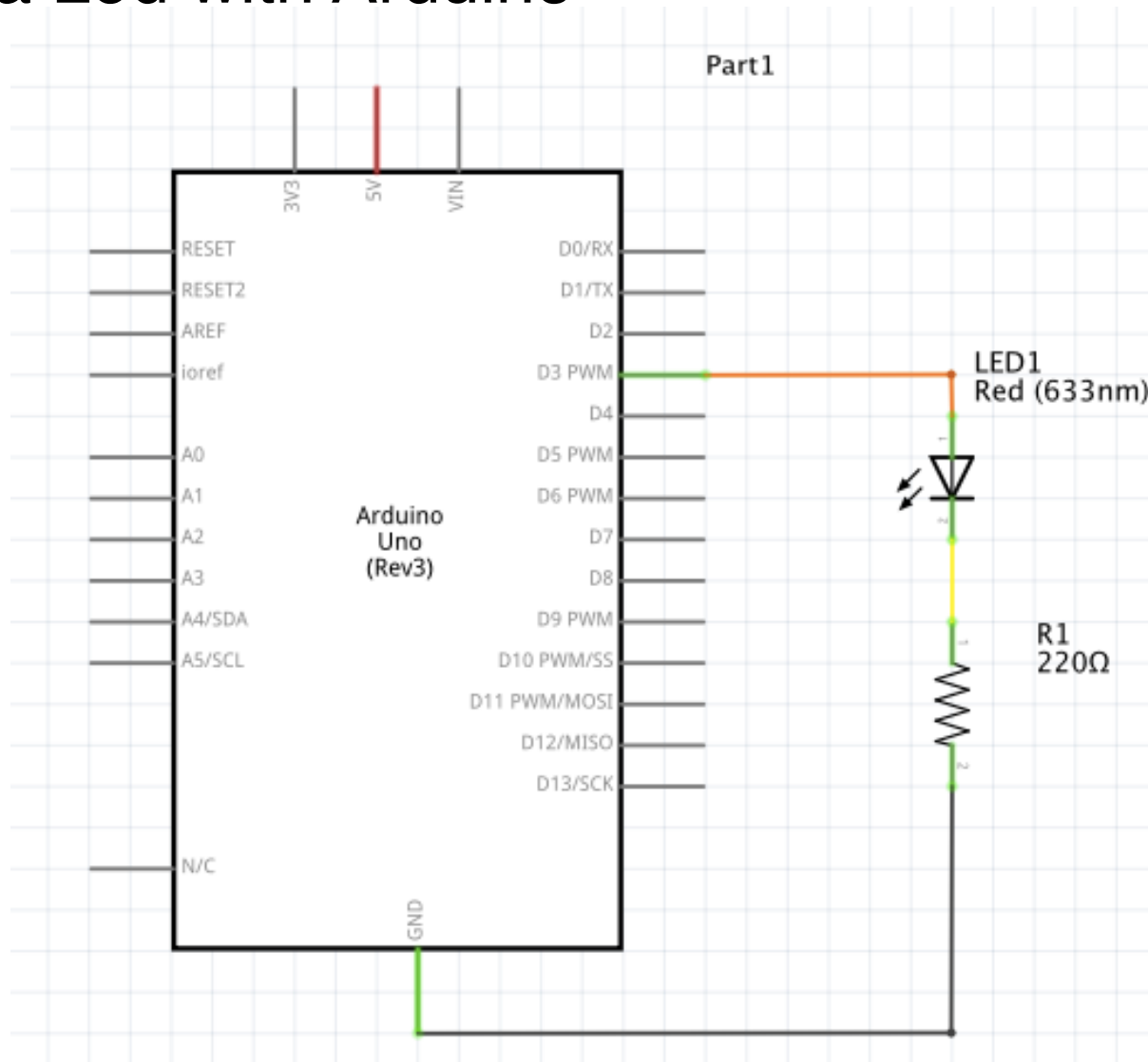


We can't control the power pins.



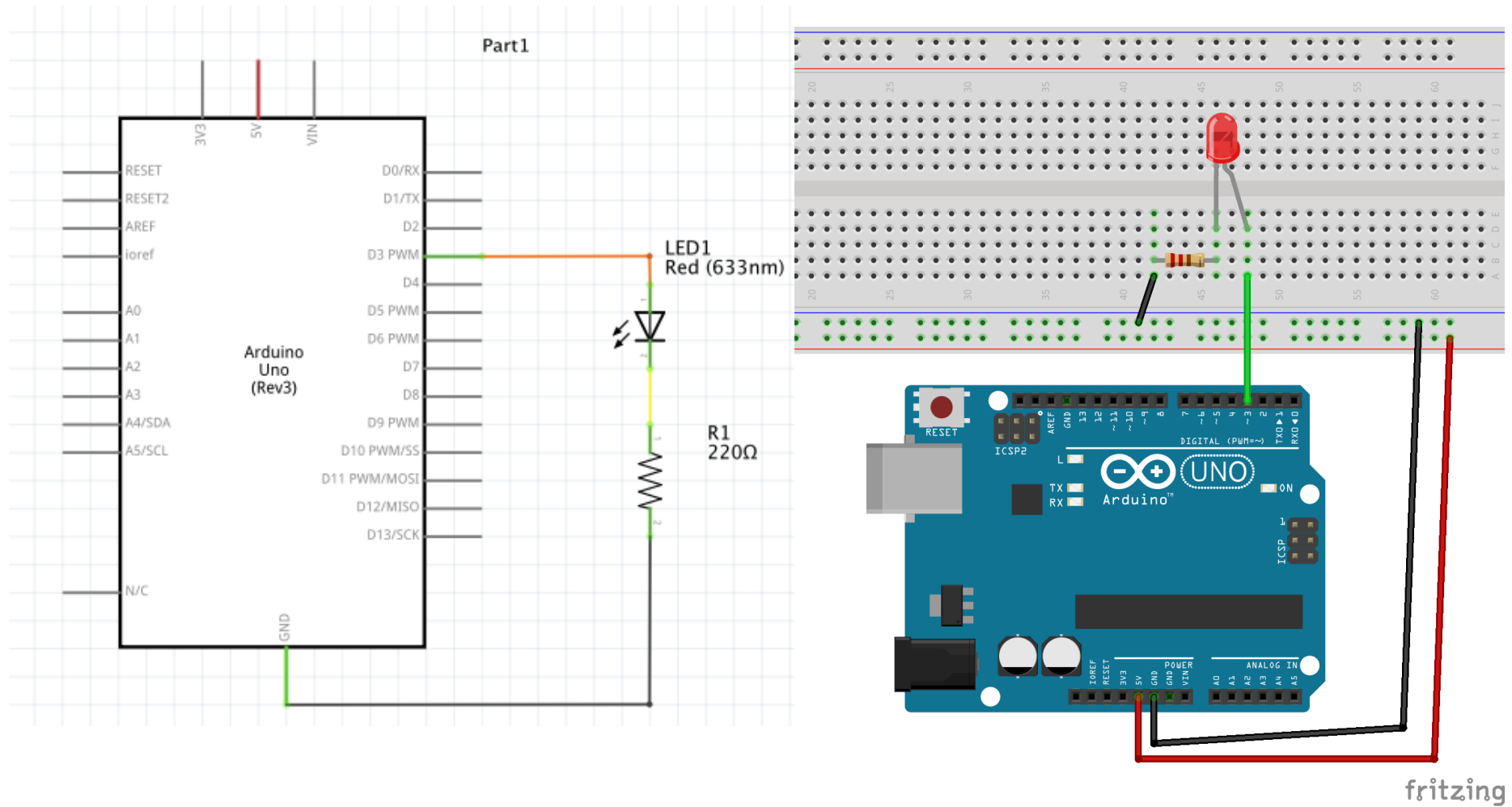
Connect the LED to pin D3

Control a Led with Arduino

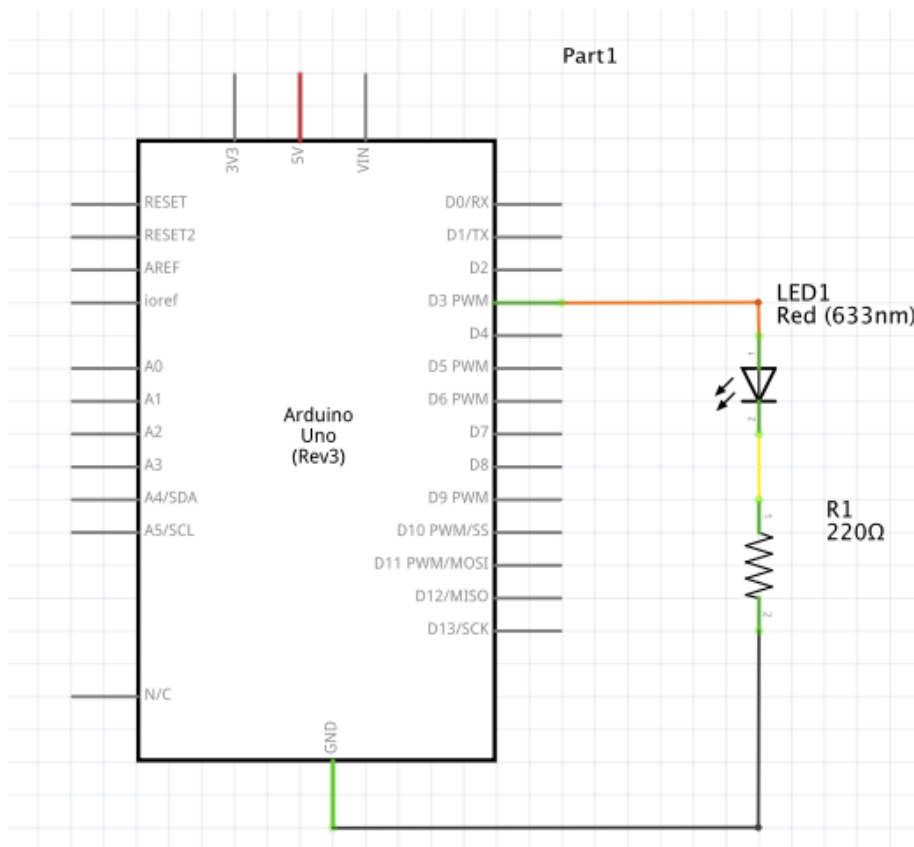


Connect the LED to pin D3

Control a Led with Arduino



Control a Led with Arduino



```
01_blink_led | Arduino 1.8.5
01_blink_led
/*Emma Pareschi 25 Spetmber 2017
 * I turn on a led and I turn it off
 */

int led_pin = 3; //defin the pin where the Led is connected

void setup() {

    pinMode(led_pin, OUTPUT); //define pin of the Led as an output

}

void loop() {

    digitalWrite(led_pin, HIGH); //turn the Led on
    delay(100); //wait 100millisecond
    digitalWrite(led_pin, LOW); //turn the Led off
    delay(100); //wait 100millisecond

}
```

01_blink_led

Time to Program

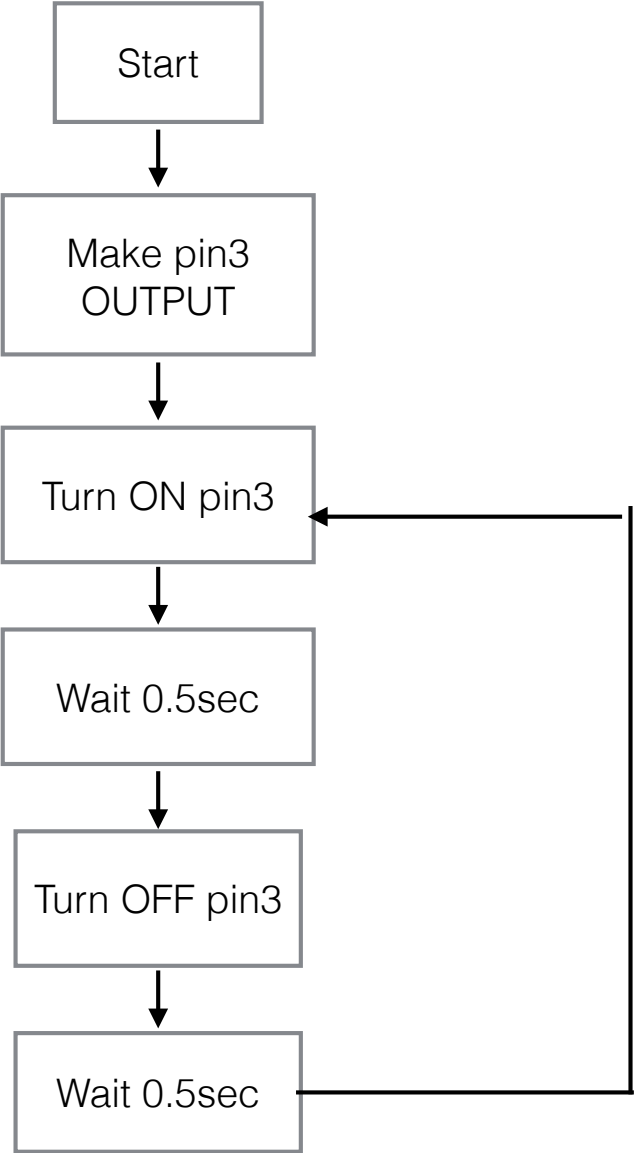
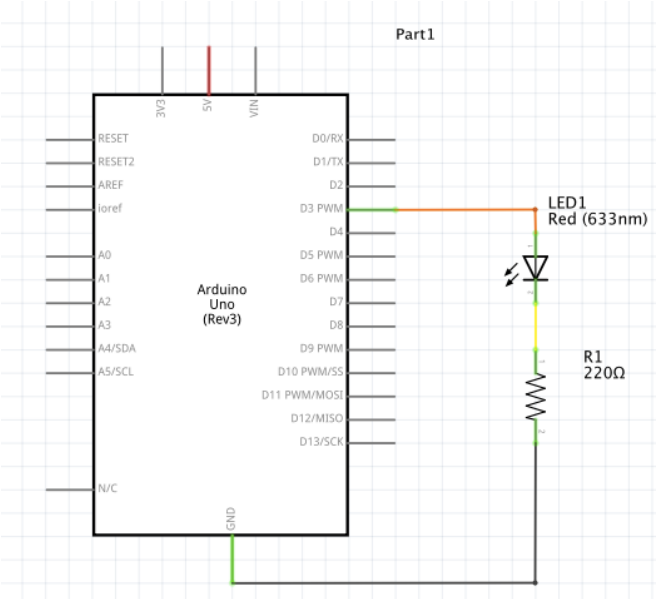
```
sketch_oct01a | Arduino 1.8.5  
sketch_oct01a §  
void setup() {  
  // put your setup code here, to run once:  
  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```

The diagram shows a flowchart with two rounded rectangular boxes. The top box is light blue and labeled 'setup()'. An arrow points from the bottom of the 'setup()' box to the top of a pink box labeled 'Loop()'. A feedback arrow starts from the bottom of the 'Loop()' box, goes down, then left, then up, and finally right to point back into the top of the 'Loop()' box, indicating a repeating cycle.

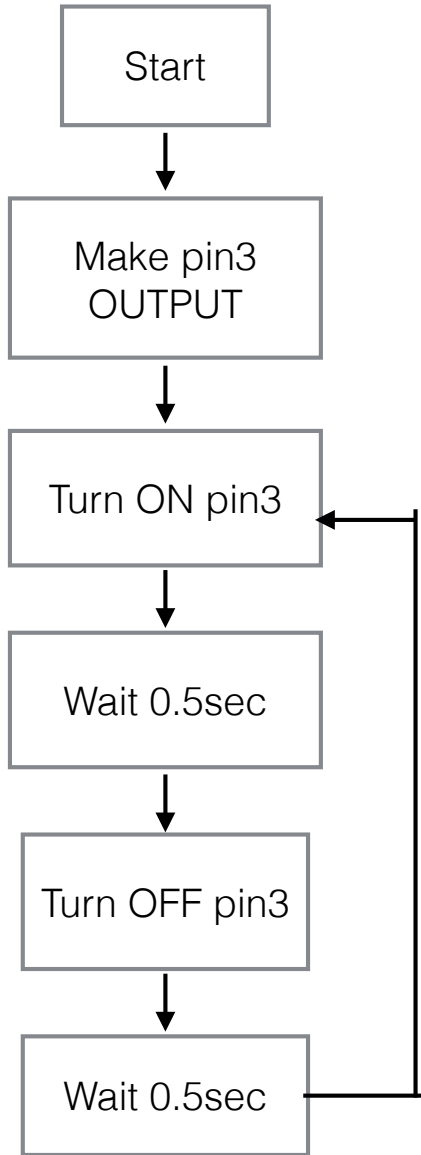
Initialise
Runs once, at the beginning when the Arduino is powered on.
Define the pins.

Running
Run repeatedly, after setup.

Control Flow



Control Flow

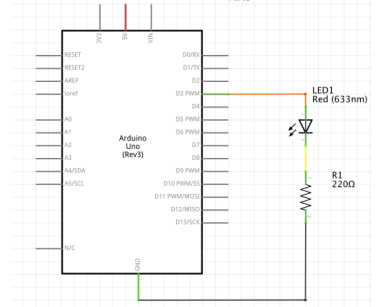


In human language

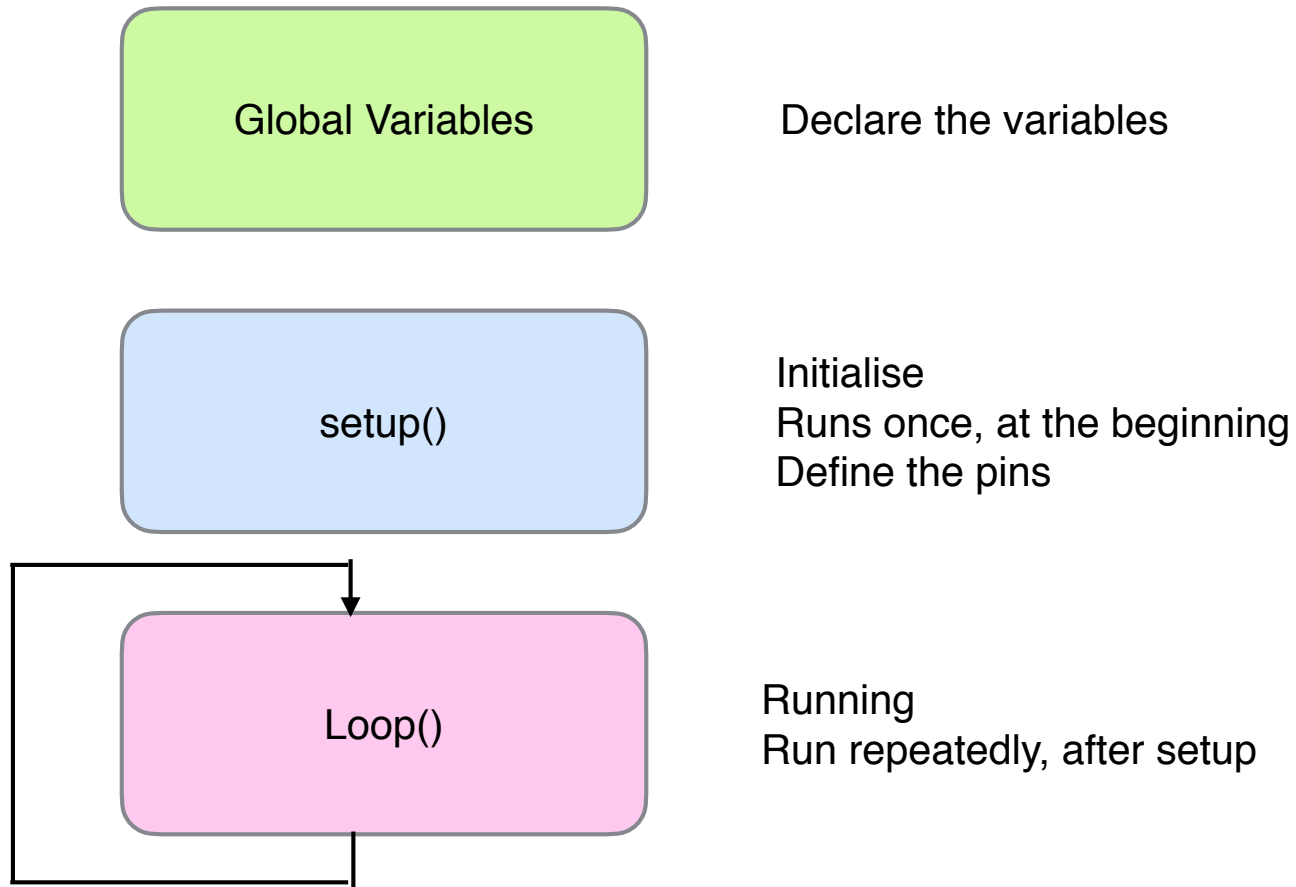
```
void setup() {  
  make the pin 3 an OUTPUT  
}  
  
void loop() {  
  turn Led ON  
  wait 0.5 sec  
  turn Led OFF  
  wait 0.5 sec  
}
```

In Arduino language

```
void setup() {  
  // put your setup code here, to run once:  
  pinMode(3, OUTPUT);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  digitalWrite(3, HIGH);  
  delay(500);  
  digitalWrite(3, LOW);  
  delay(500);  
}
```



SKETCH



01_blink_led Sketch

Global variables	<pre>int led_pin = 3; //defin the pin where the Led is connected</pre>
setup()	<pre>void setup() { pinMode(led_pin, OUTPUT); //define pin of the Led as an output }</pre>
loop()	<pre>void loop() { digitalWrite(led_pin, HIGH); //turn the Led on delay(100); //wait 100millisecond digitalWrite(led_pin, LOW); //turn the Led off delay(100); //wait 100millisecond }</pre>

Done Saving.

The sketch name had to be modified. Sketch names can only consist of ASCII characters and numbers (but cannot start with a number). They should also be less than 64 characters long.

20 ATtiny25/45/85, ATtiny85, Internal 16 MHz on /dev/cu.usbserial-FTH9HXH0

Blink Sketch

setup()
Define the function of the pin
`pinMode(pin, function);`

The pin is 'led_pin'
The functionality is Output

```
_01_blink | Arduino 1.6.9
_01_blink
/*Emma Pareschi 25 Spetmber 2017
 * I turn on a led and I turn it off
 */

int led_pin = 3; //defin the pin where the Led is connected

void setup() {
  pinMode(led_pin, OUTPUT); //define pin of the Led as an output
}

void loop() {
  digitalWrite(led_pin, HIGH); //turn the Led on
  delay(100); //wait 100millisecond
  digitalWrite(led_pin, LOW); //turn the Led off
  delay(100); //wait 100millisecond
}

Done Saving.
The sketch name had to be modified. Sketch names can only consist
of ASCII characters and numbers (but cannot start with a number).
They should also be less than 64 characters long.

20 ATtiny25/45/85, ATtiny85, Internal 16 MHz on /dev/cu.usbserial-FTH9HXH0
```

Blink Sketch

Loop()

The list of commands that arduino runs repeatedly.

DigitalWrite(pin, level);

we control the voltage on the pin

HIGH: high voltage (5V)

LOW: low voltage (ground)

Delay(time);

We add a delay in millisecond



```
/*Emma Pareschi 25 Spetmber 2017
 * I turn on a led and I turn it off
 */

int led_pin = 3; //defin the pin where the Led is connected

void setup() {

  pinMode(led_pin, OUTPUT); //define pin of the Led as an output
}

void loop() {

  digitalWrite(led_pin, HIGH); //turn the Led on
  delay(100); //wait 100millisecond
  digitalWrite(led_pin, LOW); //turn the Led off
  delay(100); //wait 100millisecond
}

Done Saving.

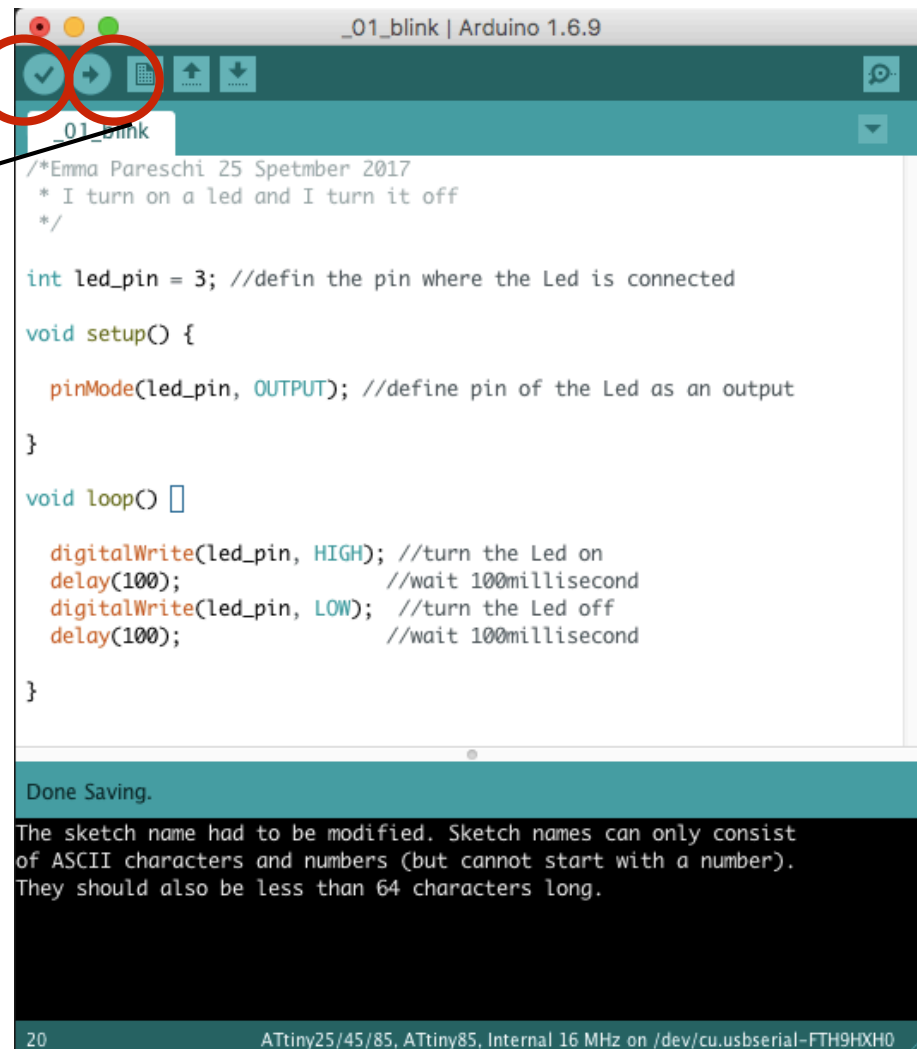
The sketch name had to be modified. Sketch names can only consist
of ASCII characters and numbers (but cannot start with a number).
They should also be less than 64 characters long.
```

20 ATtiny25/45/85, ATtiny85, Internal 16 MHz on /dev/cu.usbserial-FTH9HXH0

Connect, Compile and Upload

1. Compile

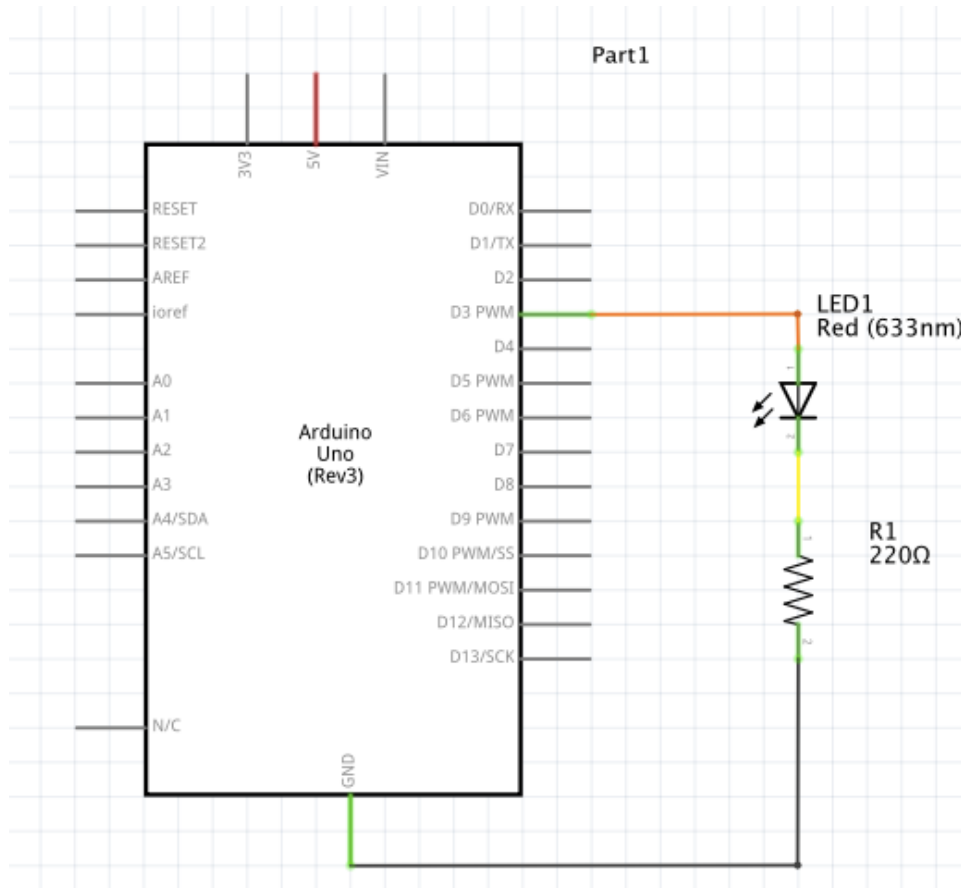
2. Upload



Arduino: Led

Make a Led fades

Arduino and a Led Fade



02_led_fade

```
/*  
 * Fade  
 *  
 * This example shows how to fade an LED on pin 9 using the analogWrite() function.  
 *  
 * The analogWrite() function uses PWM, so if you want to change the pin you're using, be sure to use another PWM capable pin. On most Arduino, the PWM pins are identified with a "~" sign, like ~3, ~5, ~6, ~9, ~10 and ~11.  
 *  
 * This example code is in the public domain.  
 *  
 * http://www.arduino.cc/en/Tutorial/Fade  
 */  
  
int led = 3;           // the PWM pin the LED is attached to  
int brightness = 0;   // how bright the LED is  
int fadeAmount = 5;   // how many points to fade the LED by  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // declare pin 9 to be an output:  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  // set the brightness of pin 9:  
  analogWrite(led, brightness);  
  
  // change the brightness for next time through the loop:  
  brightness = brightness + fadeAmount;  
  
  // reverse the direction of the fading at the ends of the fade:  
  if (brightness <= 0 || brightness >= 255) {  
    fadeAmount = -fadeAmount;  
  }  
}
```

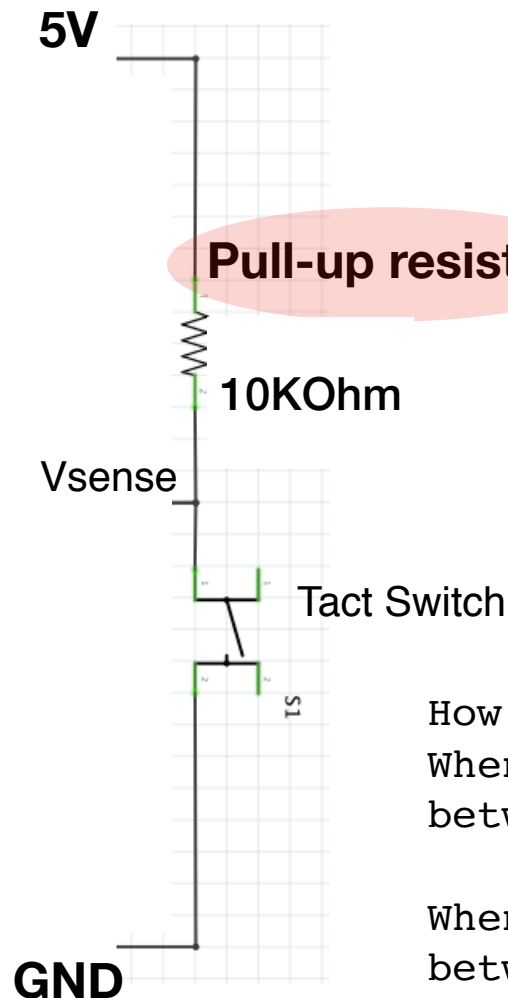
02_led_fade

From Example -> Basic -> Fade

Arduino: Digital Sensors

Read a Digital Sensor and print the value on the Serial Monitor

How the digital sensor change the Voltage

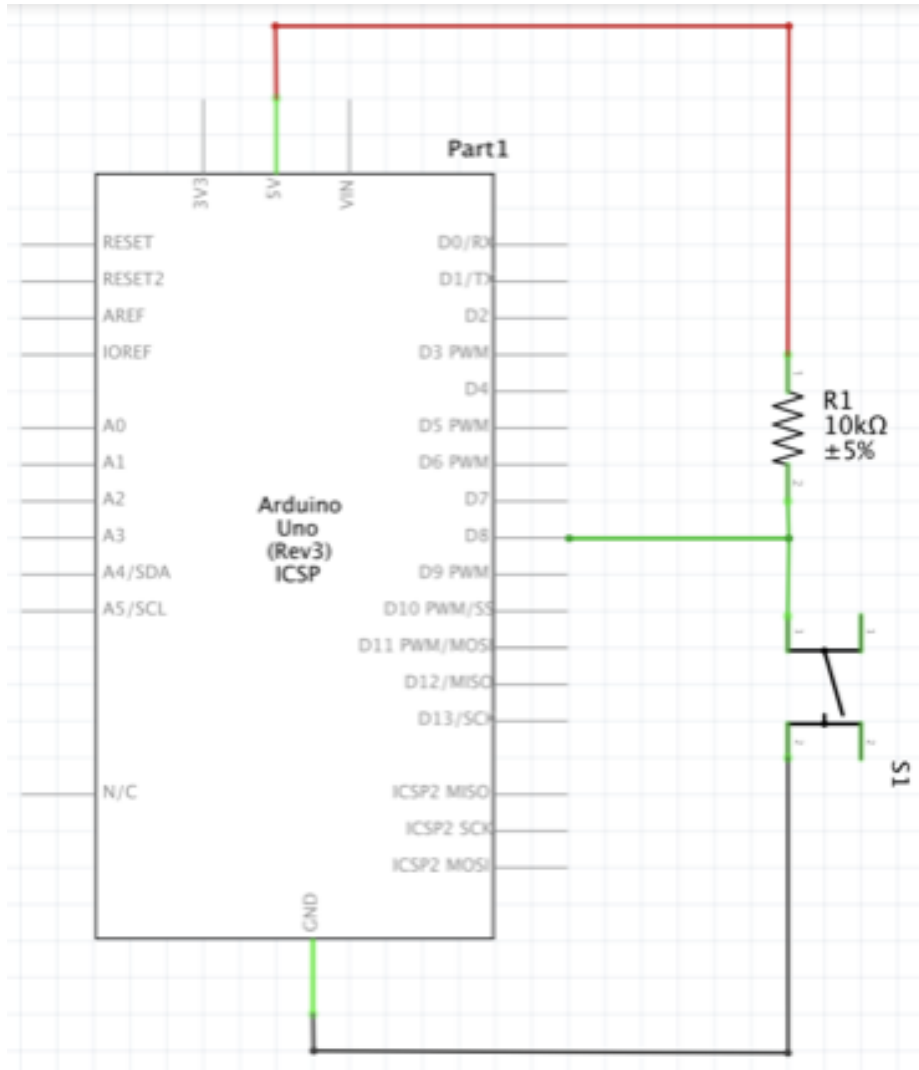


How it works:

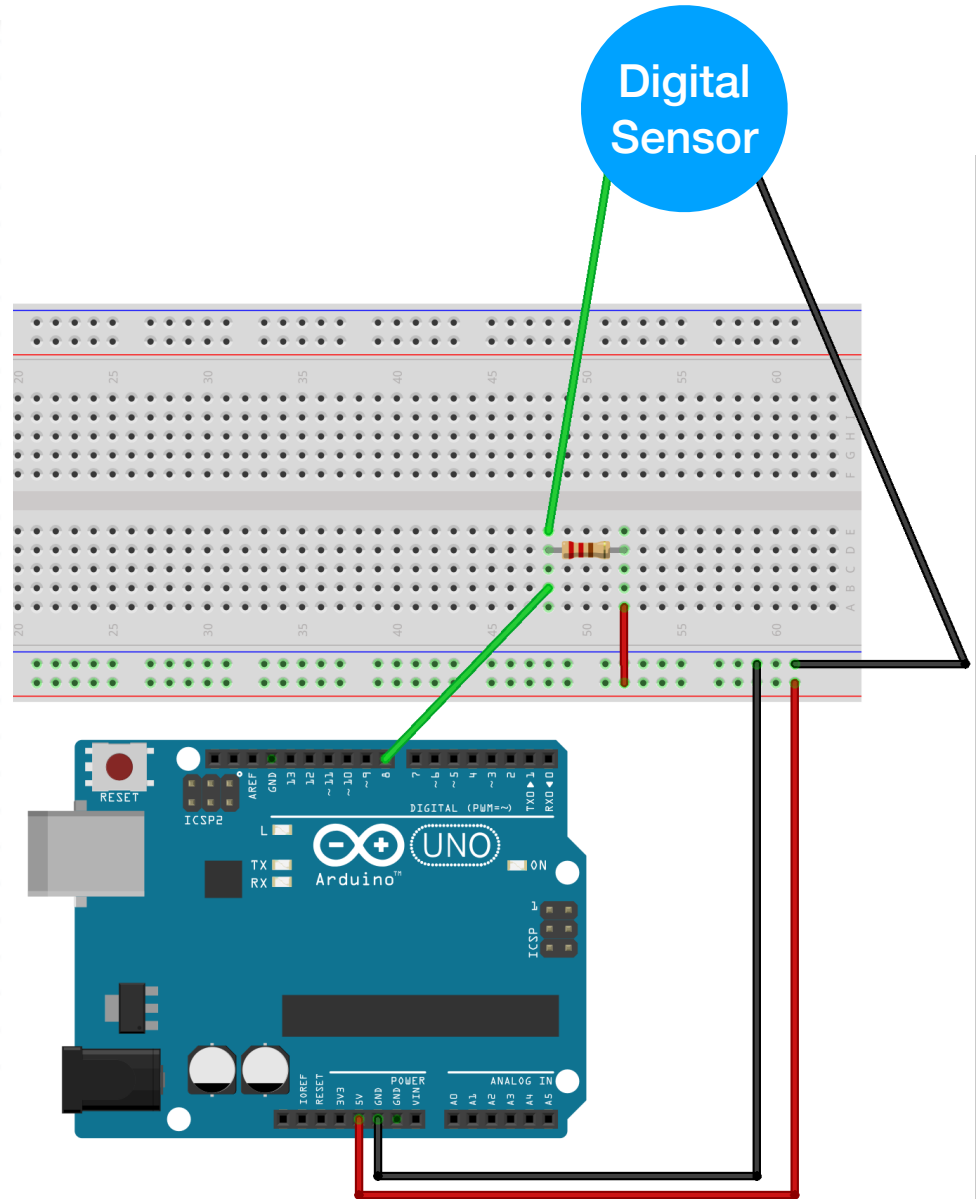
When the switch is NOT pushed the voltage between Vsense and GND is 5 Volt

When the switch is pushed the voltage between Vsense and GND is 0

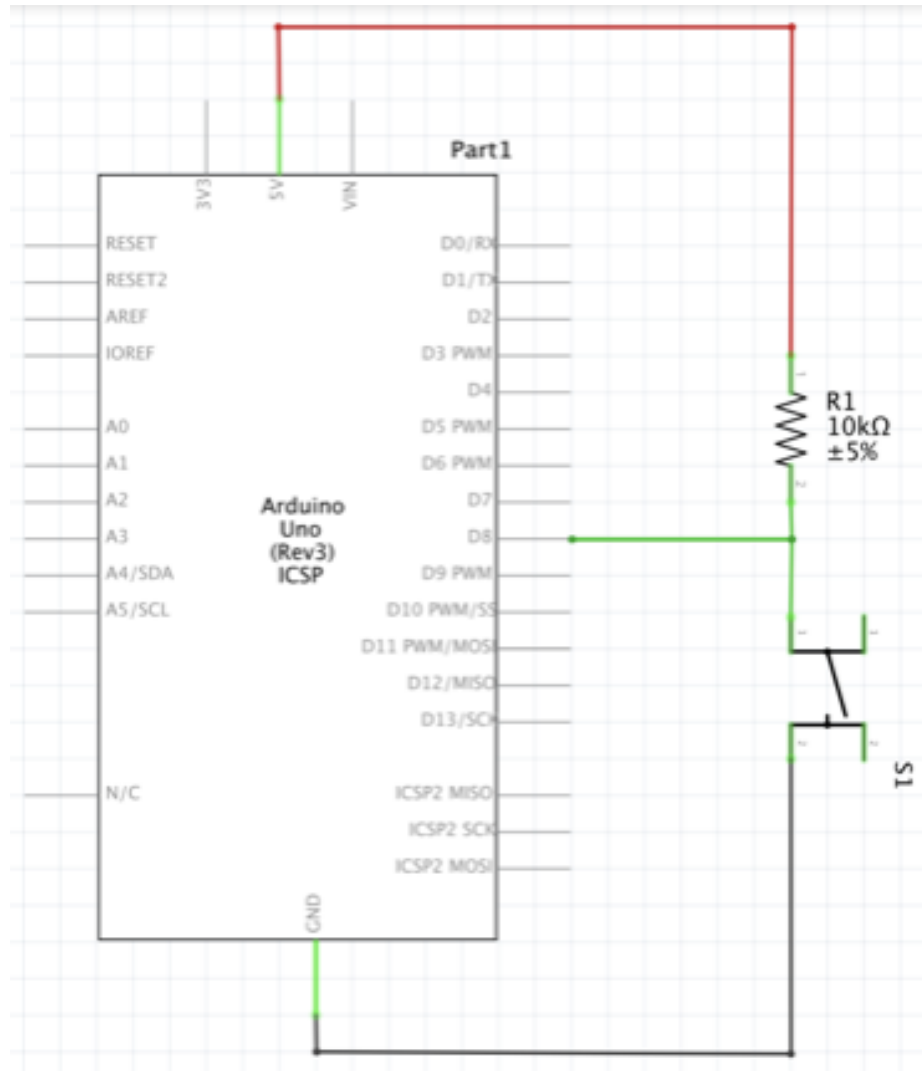
Digital switch and Arduino



`pinMode(8, INPUT);`



Digital switch and Arduino



```
pinMode(8, INPUT);
```

03_read_DigitalSensor

```
/*Emma Pareschi  
 * we read the value of a digital sensor connected to pin digital_sensor_pin and  
 * we print it on the Serial Monitor  
 */
```

```
int digital_sensor_pin = 8; //change the pin, where the sensor is connected?  
int digital_sensor_value = 0;
```

```
void setup() {
```

```
  // put your setup code here, to run once:
```

```
  pinMode(digital_sensor_pin, INPUT); //define the pin as INPUT
```

```
  Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
  // put your main code here, to run repeatedly:
```

```
  digital_sensor_value = digitalRead(digital_sensor_pin); // read the sensor
```

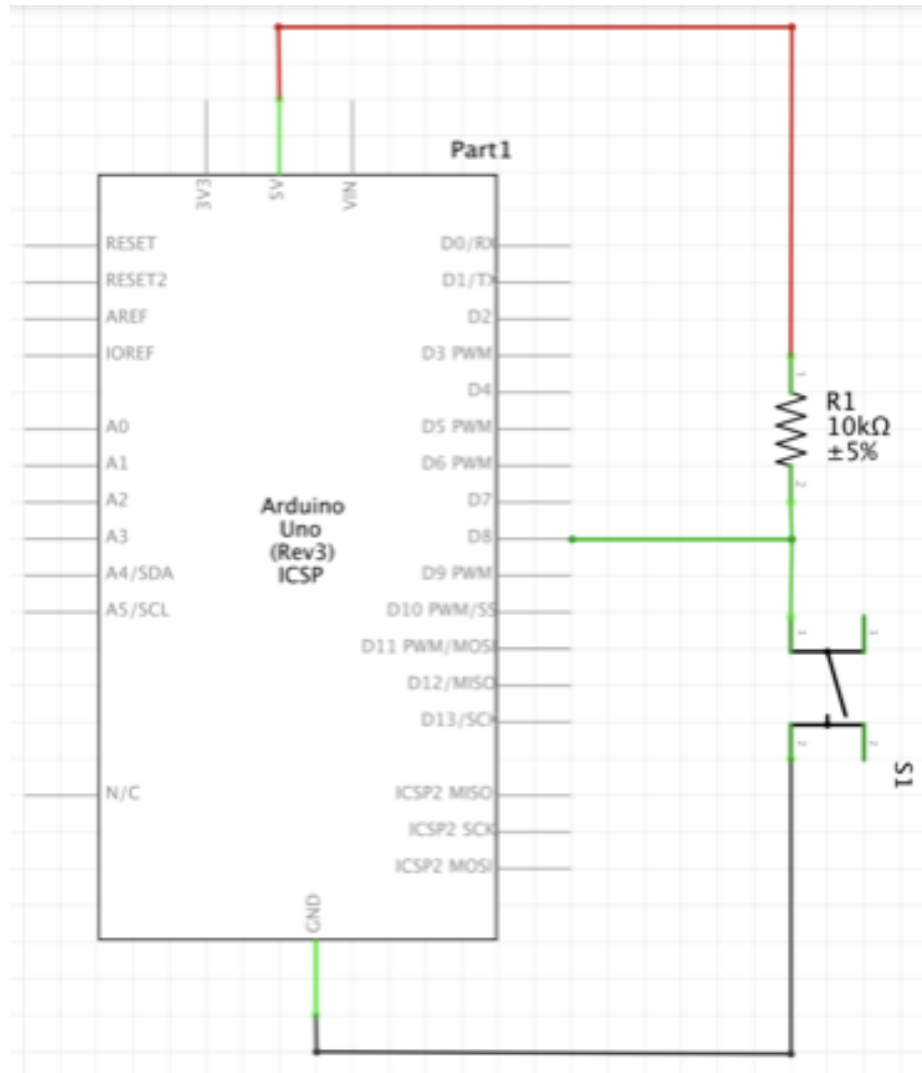
```
  Serial.println(digital_sensor_value); //print the value
```

```
  delay(100);
```

```
}
```

03_read_DigitalSensor

Digital switch and Arduino - Serial Monitor

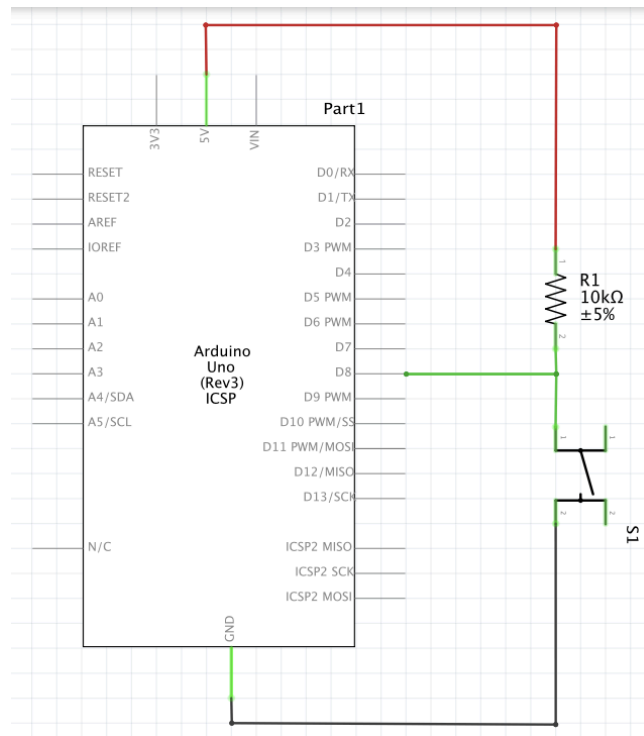


`pinMode(8, INPUT);`

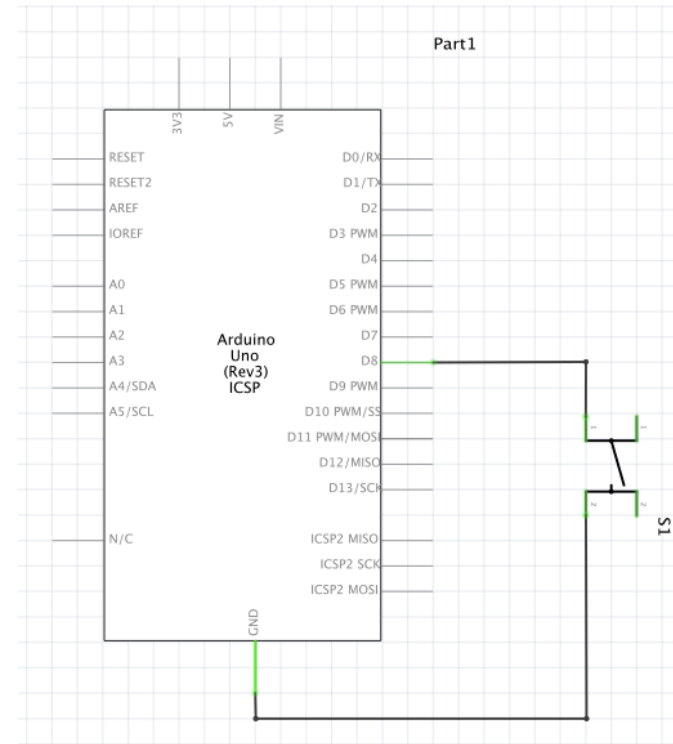
```
03_read_DigitalSensor | Arduino 1.8.5  
03_read_DigitalSensor  
/*Emma Pareschi  
 * we read the value of a digital sensor connected to pin digital_sensor_pin and  
 * we print it on the Serial Monitor  
 */  
  
int digital_sensor_pin = 8; //change the pin, where the sensor is connected?  
int digital_sensor_value = 0;  
  
void setup() {  
 // put your setup code  
 pinMode(digital_sensor_pin, INPUT);  
 Serial.begin(9600);  
}  
  
void loop() {  
 // put your main code  
 digital_sensor_value = digitalRead(digital_sensor_pin);  
 Serial.println(digital_sensor_value);  
 delay(100);  
}  
  
Done Saving.  
avrdude done. Thank you.  
14  
Autoscroll No line ending 9600 baud
```

`03_read_DigitalSensor`

Digital switch and Arduino



```
pinMode(8, INPUT);
```

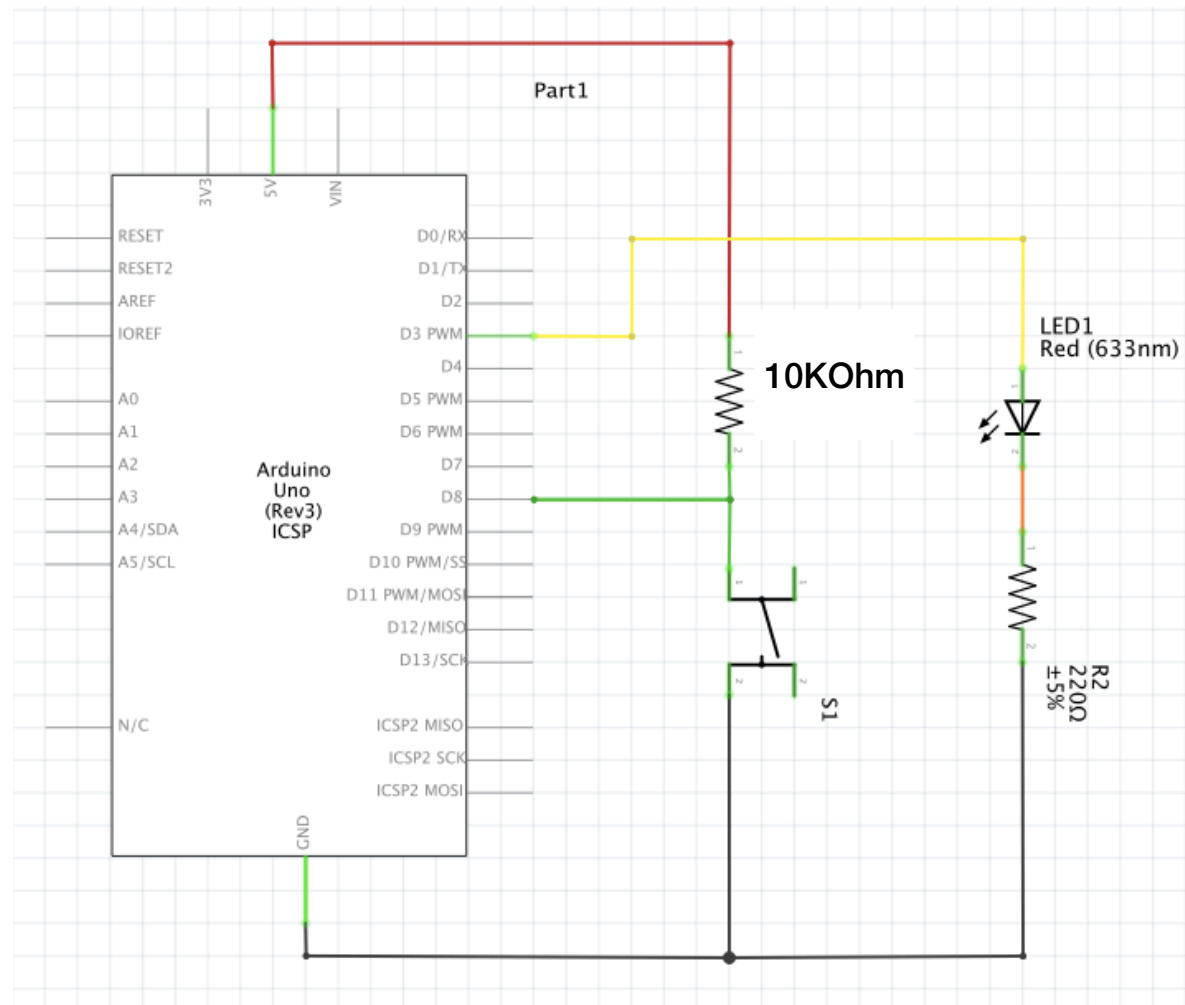


```
pinMode(8, INPUT_PULLUP);
```

Arduino: Digital Sensors - Led

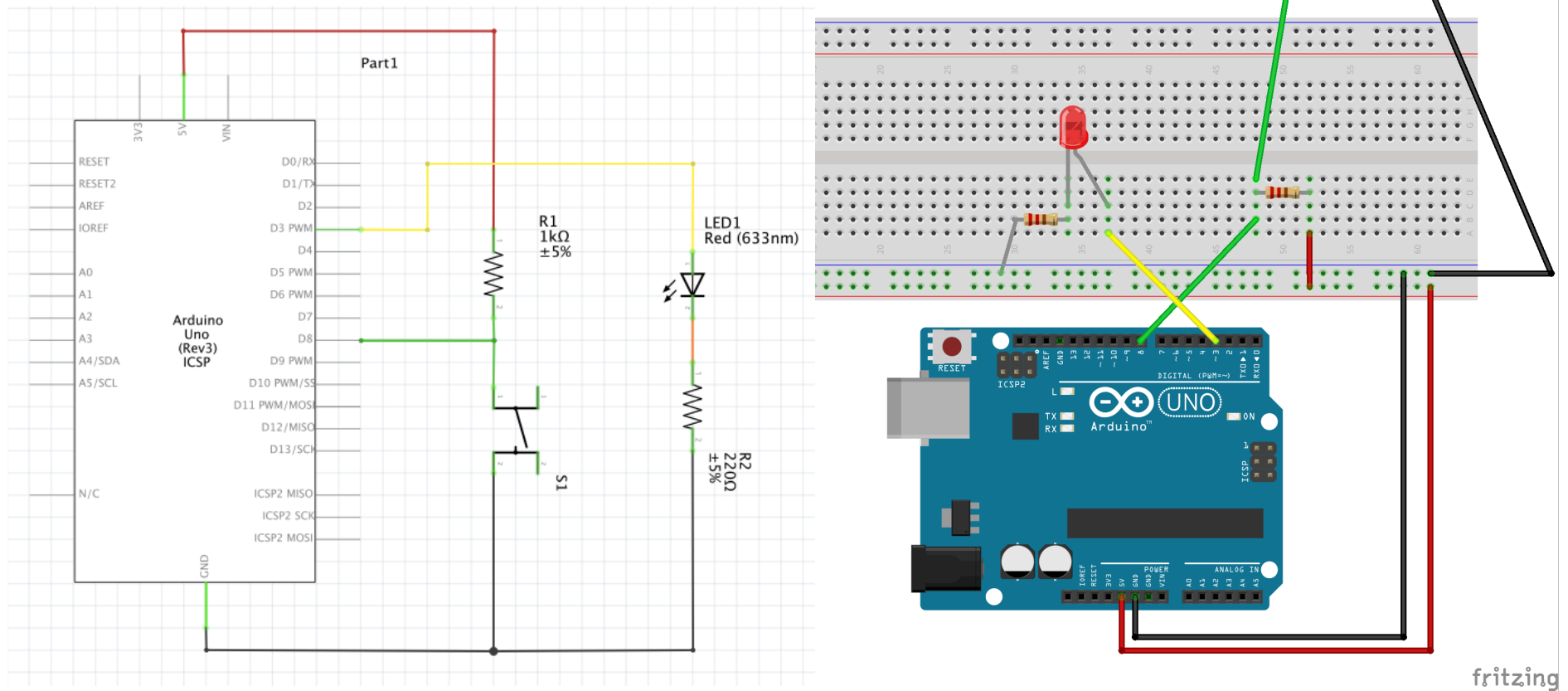
Control a Led with a Digital Sensor

Digital switch and a Led



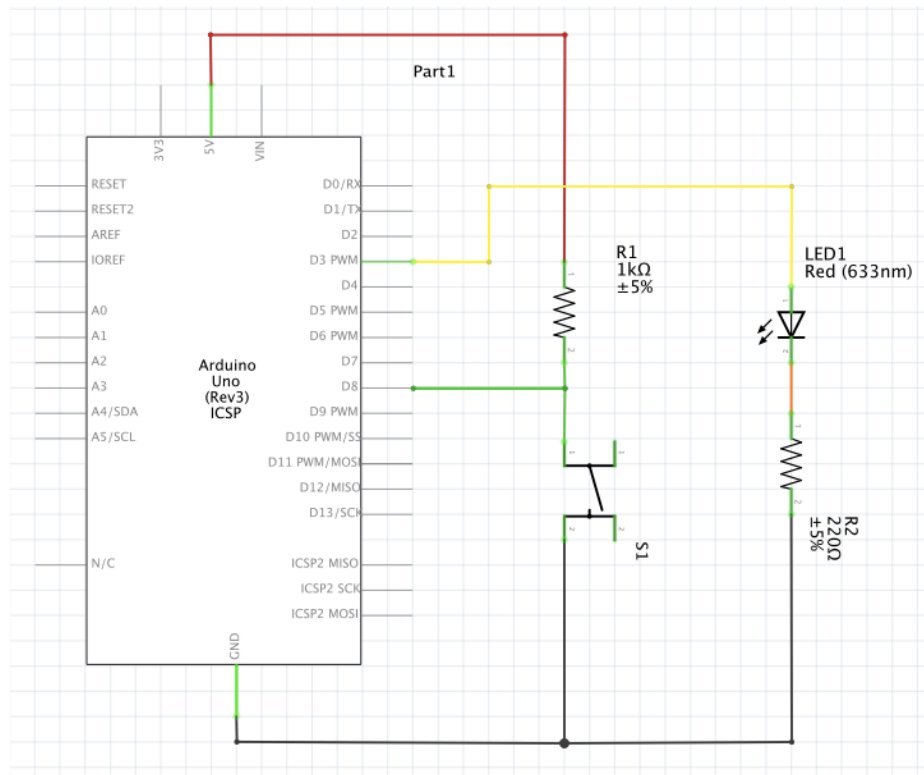
You already have the circuit of the sensor, you have to add the Led!

Digital switch and a Led



You already have the circuit of the sensor, you have to add the Led!

Digital switch and a Led



03_button_led

```
/*Emma Pareschi,
 * this sketch is a modification of the example button!!
 */

int digital_sensor_pin = 7; //change the pin, where the sensor is connected?
int digital_sensor_value = 0;
int led_pin = 3; //change the pin of the Led

void setup() {
  // put your setup code here, to run once:
  pinMode(digital_sensor_pin, INPUT);
  Serial.begin(9600);
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(led_pin, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digital_sensor_value = digitalRead(digital_sensor_pin);

  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
  if(digital_sensor_value == HIGH){
    // turn LED on:
    digitalWrite(led_pin, HIGH);
  } else {
    // turn LED off:
    digitalWrite(led_pin, LOW); // turn the LED off by making the voltage LOW
  }
}
```

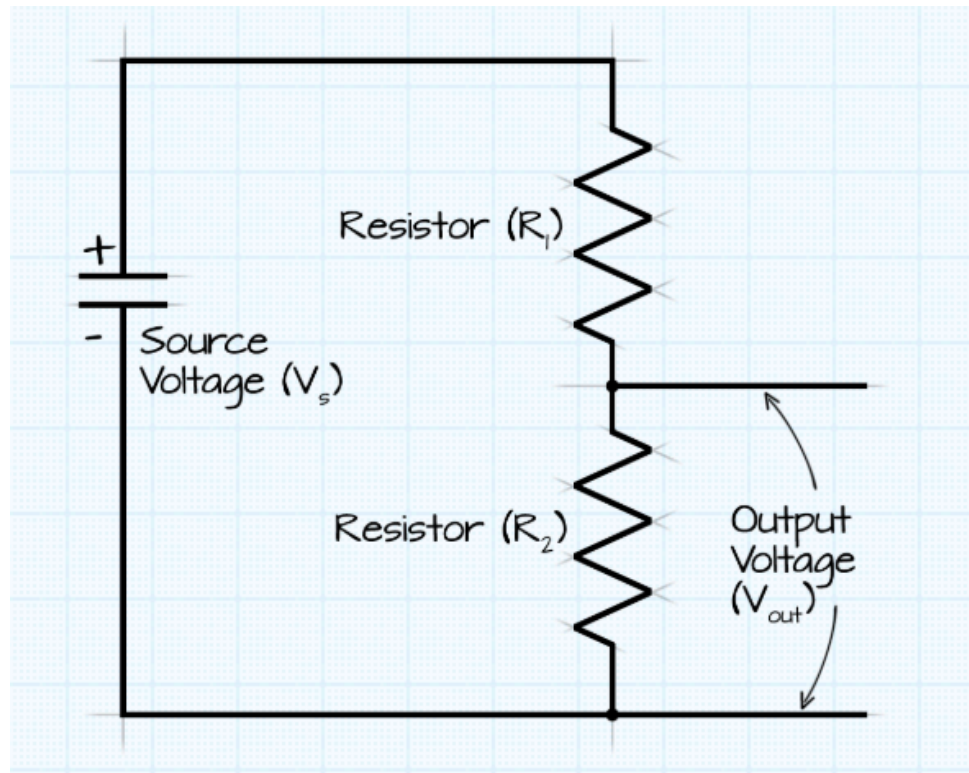
03_button_led

From Example -> Digital -> Button

Arduino: Analog Sensor

Read and print the value of an Analog Sensor connected to Arduino

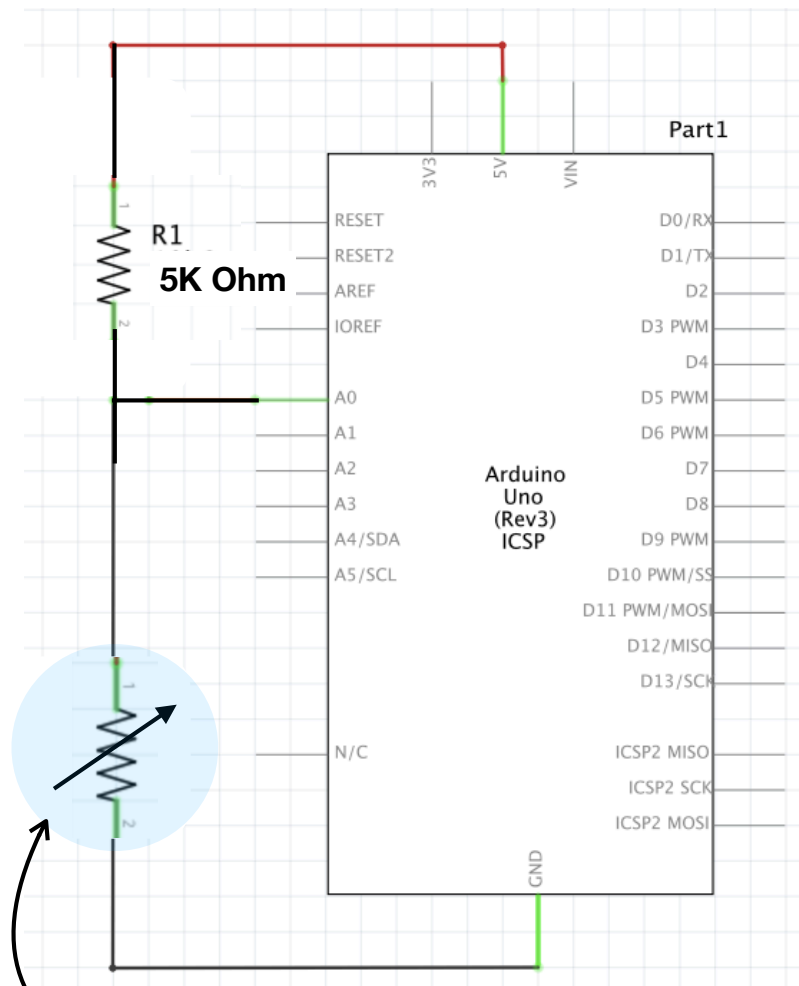
Voltage divider (analog sensor)



$$V_{out} = \frac{V_s \times R_2}{(R_1 + R_2)}$$

<http://www.ohmslawcalculator.com/voltage-divider-calculator>

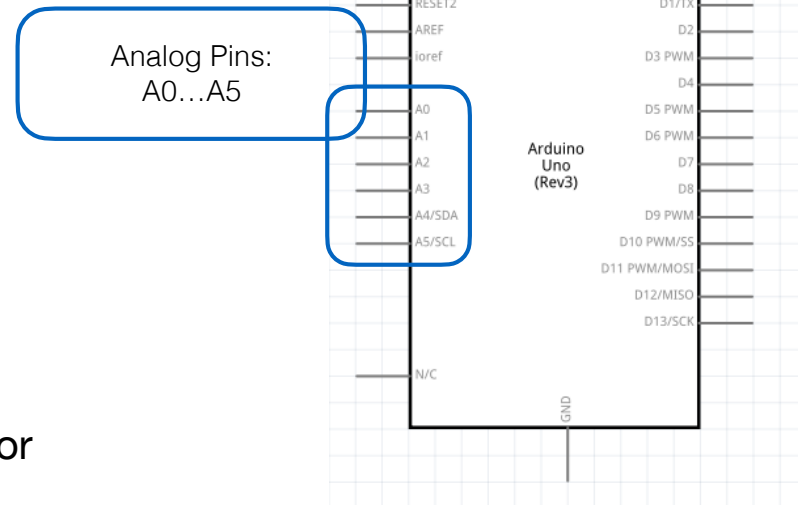
Analog sensor - schematic



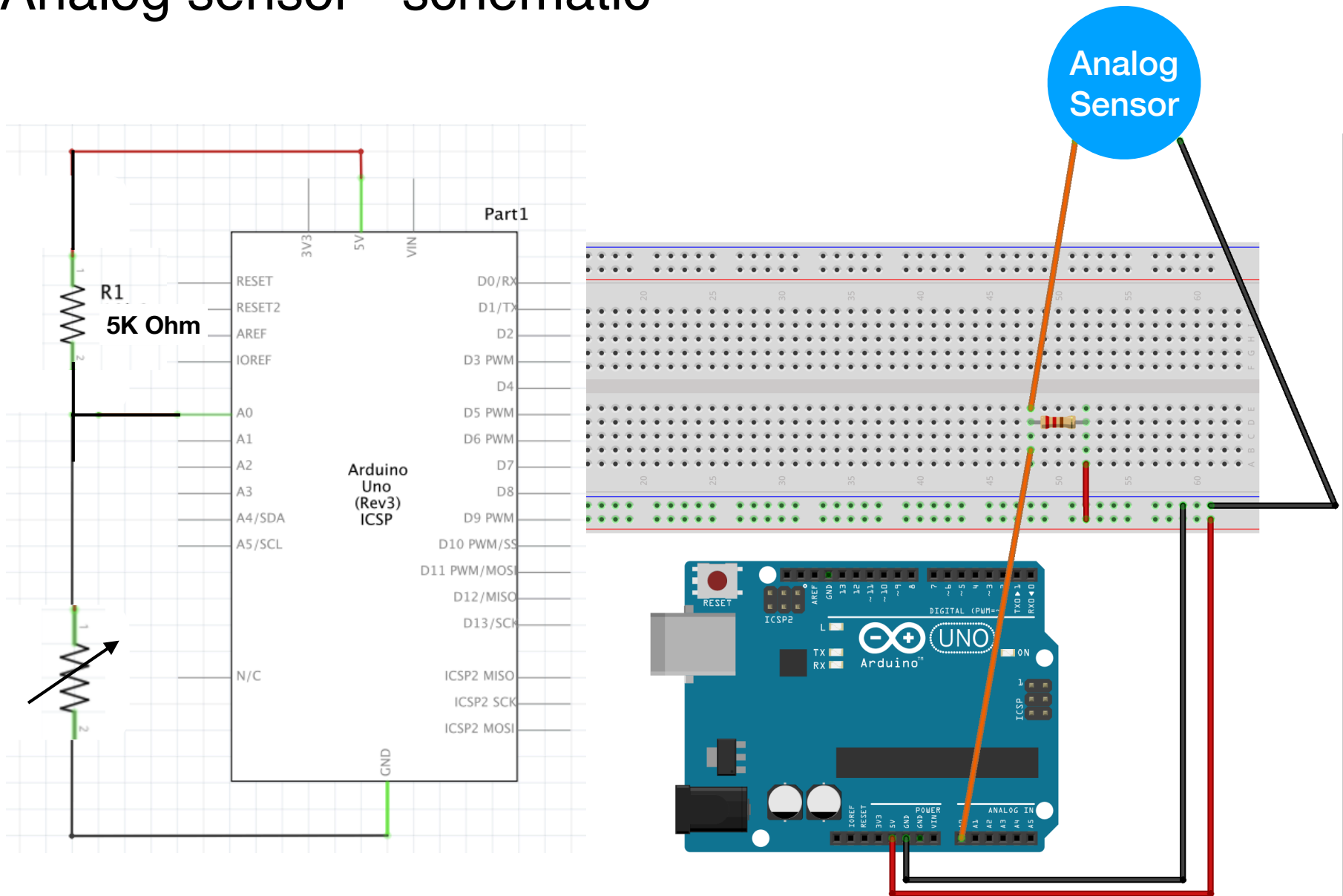
This is the symbol of a Variable Resistor

An Analog sensor is a variable resistor!! Change the resistance you change the Voltage V_{sense}the Arduino will read this Voltage.

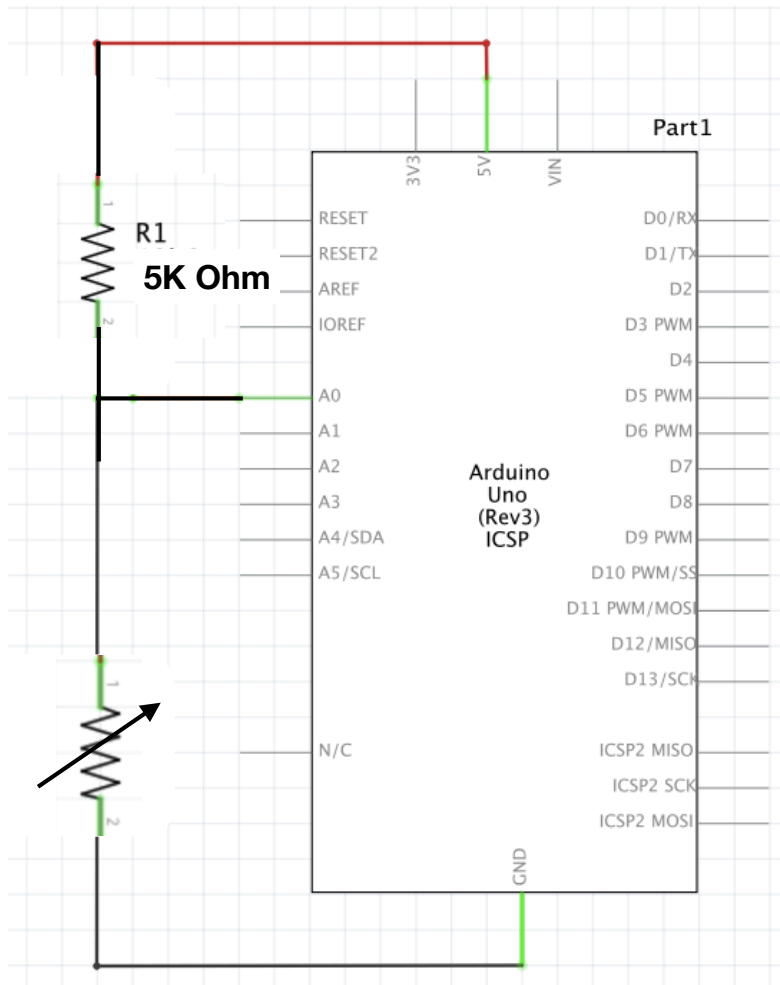
To read an Analog sensor that generates an analog voltage => you need to use the ANALOG PINS.



Analog sensor - schematic



Read an Analog sensor

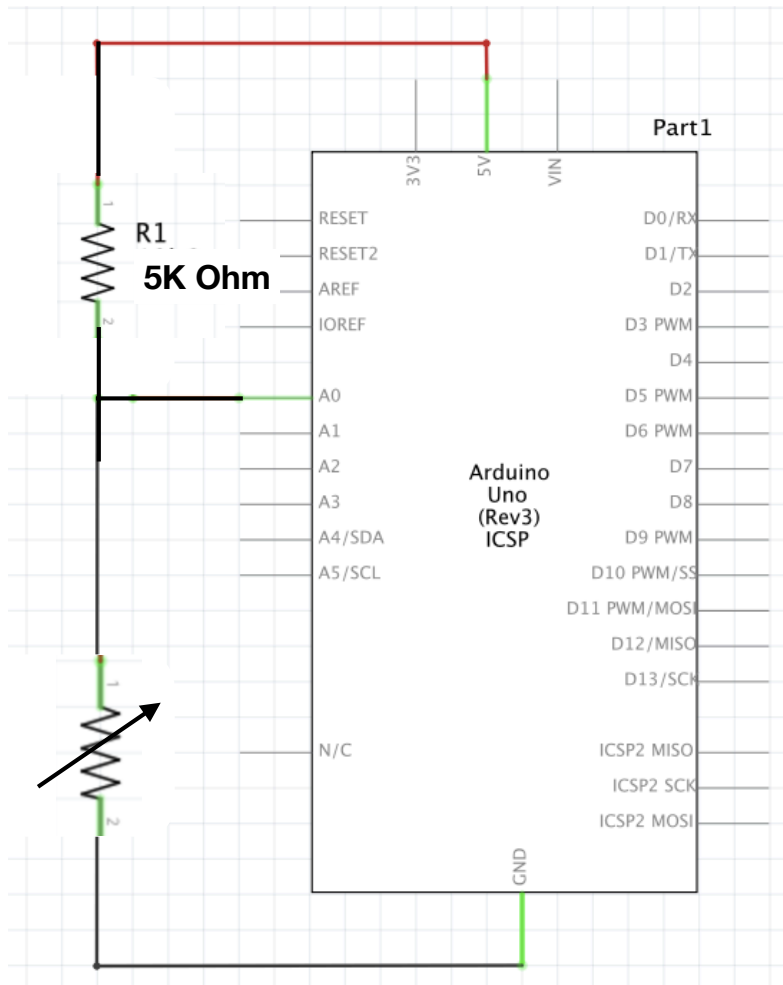


04_fabricademy_analog_sensor

```
/*Emma Pareschi,  
 * with this sketch we read the analog sensor connected to pin analog_sensor_pin  
 */  
  
int analog_sensor_pin = A0; //change the pin, where the sensor is connected?  
int analog_sensor_value = 0;  
  
void setup() {  
  // put your setup code here, to run once:  
  pinMode(analog_sensor_pin, INPUT);  
  Serial.begin(9600);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  analog_sensor_value = analogRead(analog_sensor_pin); //read the Voltage of the pin sensor  
  
  Serial.println(analog_sensor_value); // print the value on the Serial monitor  
  delay(100);  
}
```

05_read_AnalogSensor

Read and map an Analog sensor



```
05_read_AnalogSensor_map
/*Emma Pareschi,
 * with this sketch we read the analog sensor connected to pin analog_sensor_pin
 */

int analog_sensor_pin = A0; //change the pin, where the sensor is connected?
int analog_sensor_value = 0;

void setup() {
  // put your setup code here, to run once:
  pinMode(analog_sensor_pin, INPUT);
  Serial.begin(9600);
}

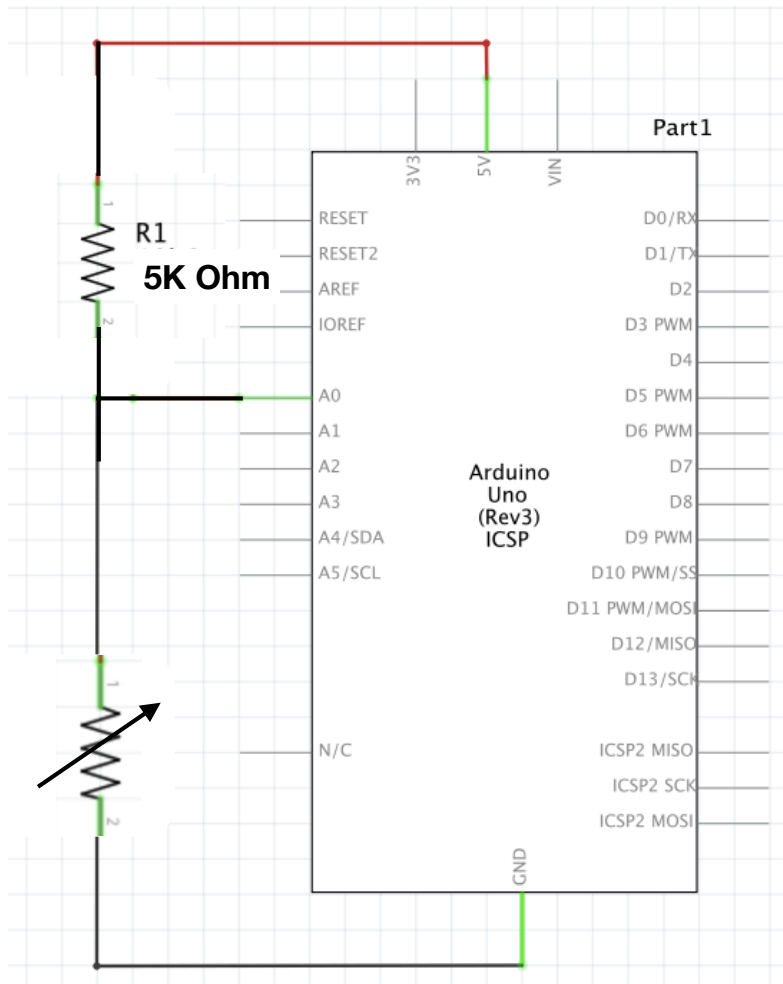
void loop() {
  // put your main code here, to run repeatedly:
  analog_sensor_value = analogRead(analog_sensor_pin); //read the Voltage of the pin sensor

  analog_sensor_value = map(analog_sensor_value, 230, 130, 0, 255); //we change the range
  analog_sensor_value = constrain(analog_sensor_value, 0, 255); //we apply the limits

  Serial.println(analog_sensor_value); // print the value on the Serial monitor
  delay(100);
}
```

06_read_AnalogSensor_map

Smooth the value of an Analog sensor



Smoothing

```
// Define the number of samples to keep track of. The higher the number, the
// more the readings will be smoothed, but the slower the output will respond to
// the input. Using a constant rather than a normal variable lets us use this
// value to determine the size of the readings array.
const int numReadings = 10;
```

```
int readings[numReadings]; // the readings from the analog input
int readIndex = 0;         // the index of the current reading
int total = 0;             // the running total
int average = 0;          // the average
```

```
int inputPin = A0;
```

```
void setup() {
  // initialize serial communication with computer:
  Serial.begin(9600);
  // initialize all the readings to 0:
  for (int thisReading = 0; thisReading < numReadings; thisReading++) {
    readings[thisReading] = 0;
  }
}
```

```
void loop() {
  // subtract the last reading:
  total = total - readings[readIndex];
  // read from the sensor:
  readings[readIndex] = analogRead(inputPin);
  // add the reading to the total:
  total = total + readings[readIndex];
  // advance to the next position in the array:
  readIndex = readIndex + 1;

  // if we're at the end of the array...
  if (readIndex >= numReadings) {
    // ..wrap around to the beginning:
    readIndex = 0;
  }
}
```

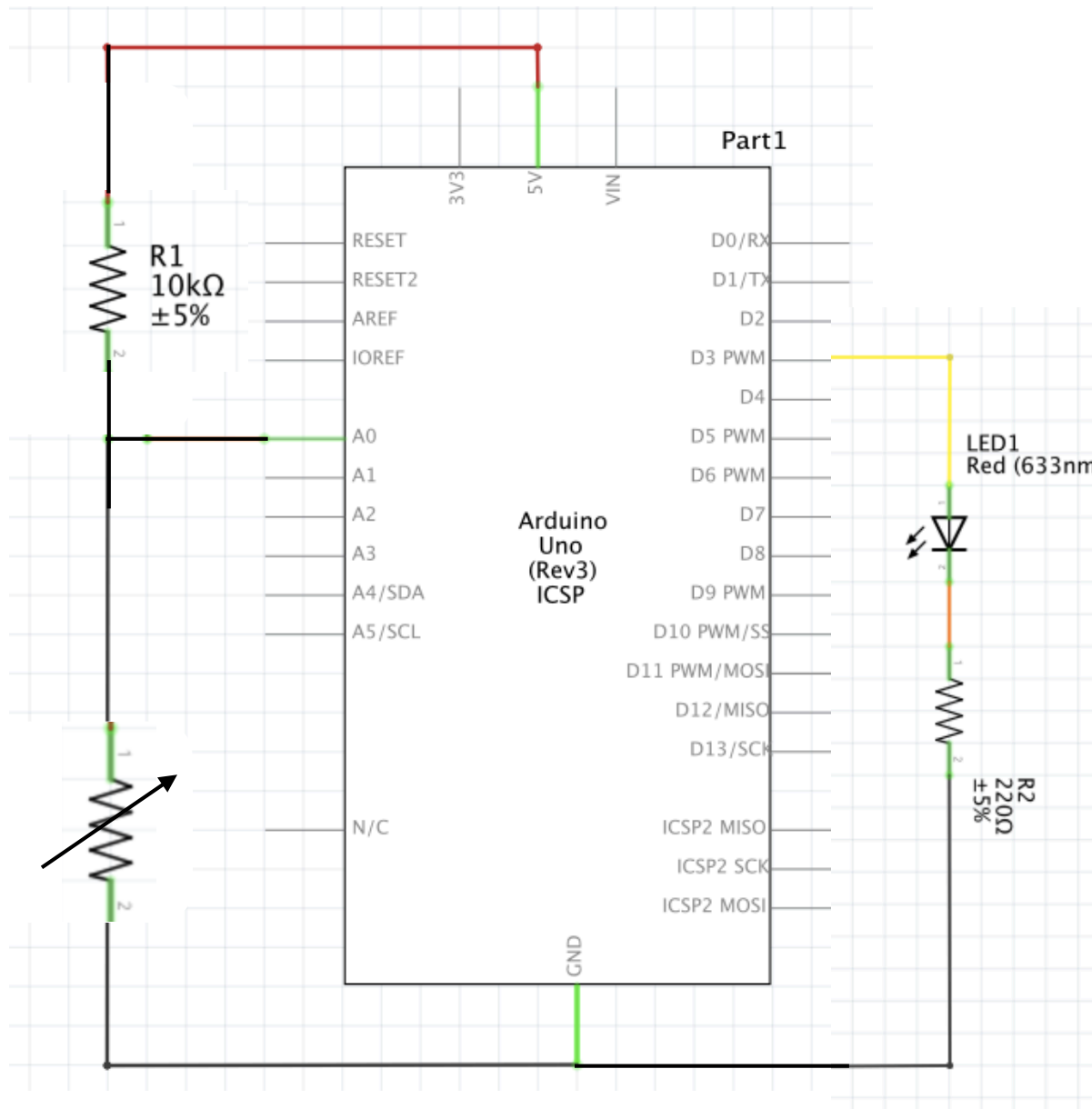
Example -> Analog -> Smoothing

Open Tools -> Serial Plotter and you will see how much smooth is the signal of the sensor... nice :)

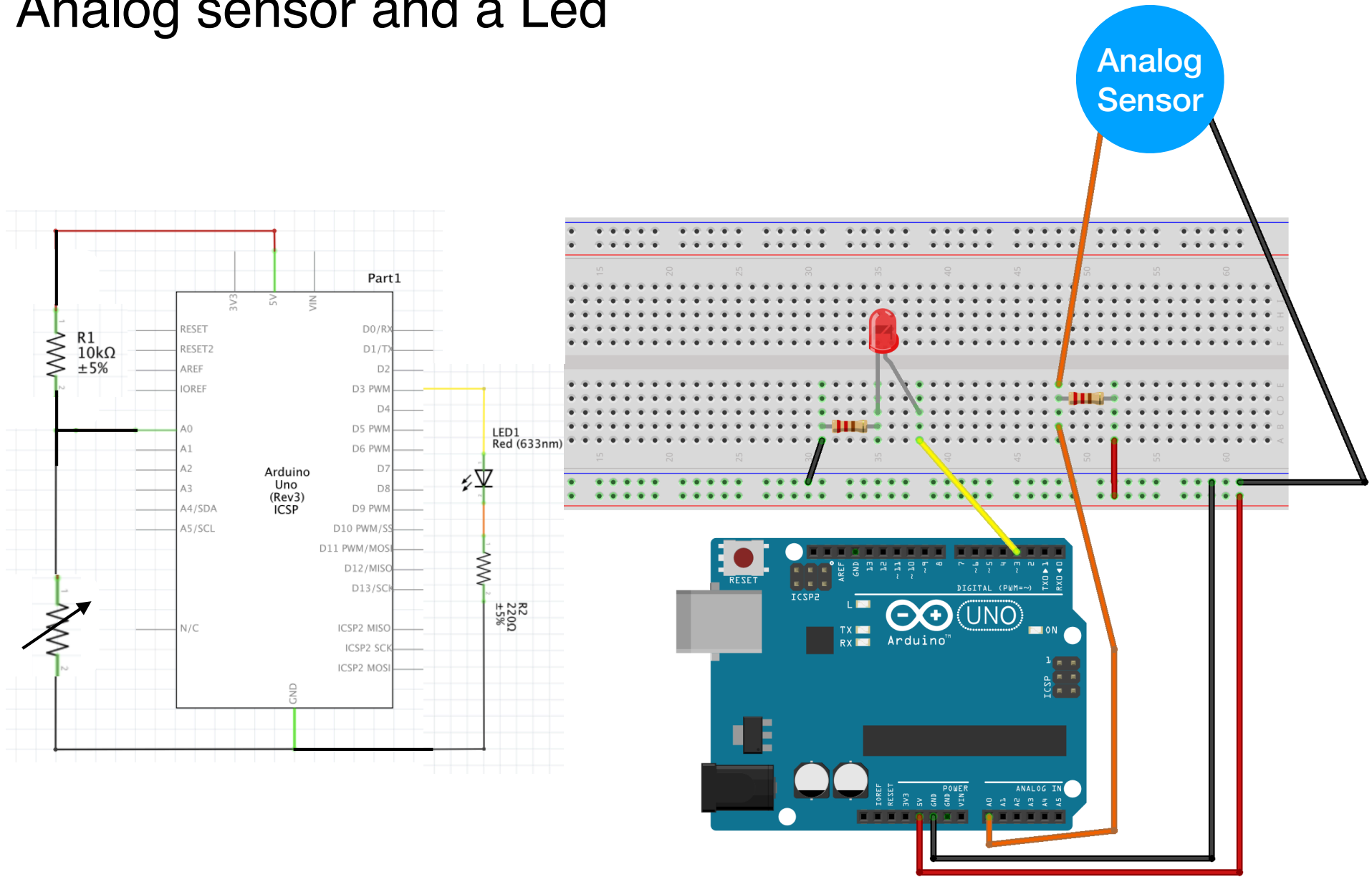
Arduino: Analog Sensor and Led

Read and print the value of an Analog Sensor connected to Arduino

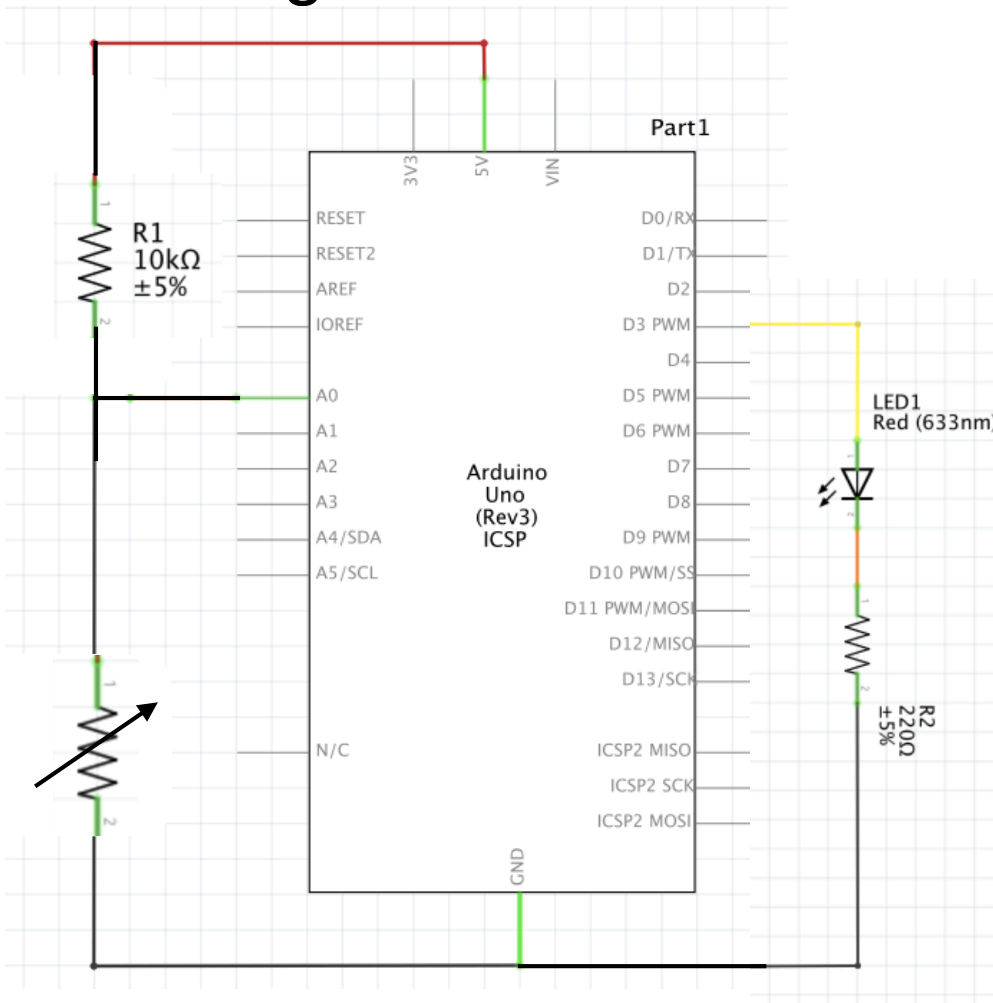
Analog sensor and a Led



Analog sensor and a Led



Analog sensor and a Led



07_pressure_led

```

/*Emma Pareschi,
 * with this sketch we read the analog sensor connected to pin analog_sensor_pin,
 * we map the value and we use it to control the Led
 */

int analog_sensor_pin = A0; //change the pin, where the sensor is connected?
int analog_sensor_value = 0;
int led_pin = 3;

void setup() {
  // put your setup code here, to run once:
  pinMode(analog_sensor_pin, INPUT);
  pinMode(led_pin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  analog_sensor_value = analogRead(analog_sensor_pin); //read the Voltage of the pin sensor

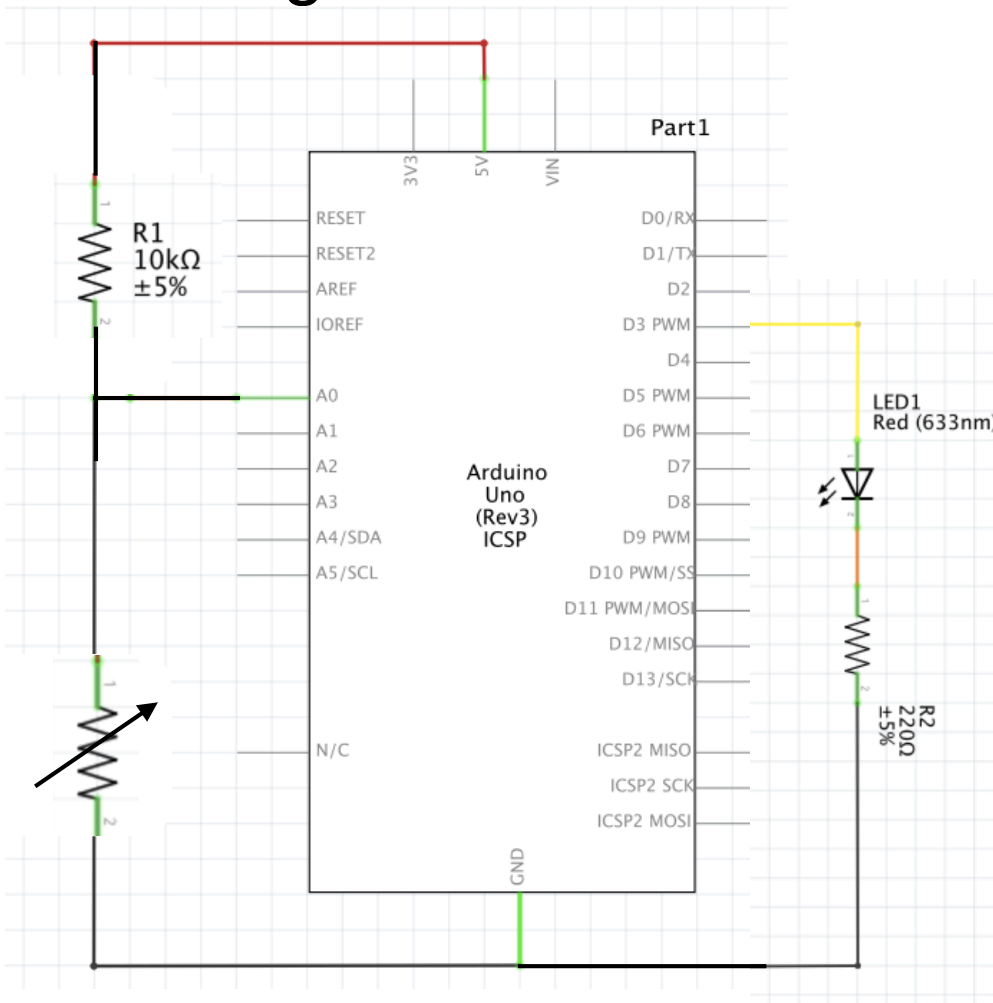
  analog_sensor_value = map(analog_sensor_value, 230, 130, 0, 255); //we change the range
  analog_sensor_value = constrain(analog_sensor_value, 0, 255); //we apply the limits

  analogWrite(led_pin, analog_sensor_value); //we use the mapped value to control the Led
  Serial.println(analog_sensor_value); // print the value on the Serial monitor
  delay(10);
}

```

07_pressure_led

Analog sensor and a Led



Calibration

/*
Calibration

Demonstrates one technique for calibrating sensor input. The sensor readings during the first five seconds of the sketch execution define the minimum and maximum of expected values attached to the sensor pin.

The sensor minimum and maximum initial values may seem backwards. Initially, you set the minimum high and listen for anything lower, saving it as the new minimum. Likewise, you set the maximum low and listen for anything higher as the new maximum.

The circuit:

- analog sensor (potentiometer will do) attached to analog input 0
- LED attached from digital pin 9 to ground

created 29 Oct 2008

by David A Mellis

modified 30 Aug 2011

by Tom Igoe

This example code is in the public domain.

<http://www.arduino.cc/en/Tutorial/Calibration>

*/

// These constants won't change:

`const int sensorPin = A0;` // pin that the sensor is attached to

`const int ledPin = 9;` // pin that the LED is attached to

// variables:

`int sensorValue = 0;` // the sensor value

`int sensorMin = 1023;` // minimum sensor value

`int sensorMax = 0;` // maximum sensor value

`void setup() {`

`// turn on LED to signal the start of the calibration period:`

`pinMode(13, OUTPUT);`

Example -> Analog -> Calibration

When you use the example don't forget to set the right pins that you are actually using in your circuit.

Sketches

This is the list of sketches we used in class, some are provided with this presentation, some come from the Example of Arduino IDE.

Control a Led

- 01_blink_Led
- 02_fade_Led

Read a Digital Sensor and print the value on the Serial Monitor

- 03_read_DigitalSensor

Control a Led with a Digital Sensor

- Button (from Example)

Read an Analog Sensor and print the value on the Serial Monitor

- 05_read_AnalogSensor
- 06_read_AnalogSensor_map
- Smoothing (from Example)

Control a Led with an Analog Sensor

- 07_pressure_led
- Calibration (from Example)

Workflow

1- decide which sensor to make, if it is your first sensor, follow a tutorial. If you already have experience, have fun and try something new or change a sensor already documented.

2- Once you have the sensor, test it with the multimeter.

3- Once the sensor looks ok (with the multimeter), connect it to the Arduino and read the value on the Serial Monitor or Serial Plotter.

4- Once you can read the values of the sensor with the Arduino, Connect also the Led and try to control the Led based on the interaction of the sensor.

Arduino Language Reference



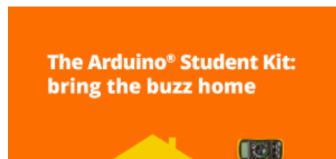
- LANGUAGE
- FUNCTIONS
- VARIABLES
- STRUCTURE

- LIBRARIES
- IOT CLOUD API
- GLOSSARY

The Arduino Reference text is licensed under a [Creative Commons Attribution-Share Alike 3.0 License](#).

Find anything that can be improved? [Suggest corrections and new documentation via GitHub](#).

Doubts on how to use Github? Learn everything you need to know in [this tutorial](#).



Language Reference

Arduino programming language can be divided in three main parts: functions, values (variables and constants), and structure.

FUNCTIONS

For controlling the Arduino board and performing computations.

Digital I/O

[digitalRead\(\)](#)
[digitalWrite\(\)](#)
[pinMode\(\)](#)

Analog I/O

[analogRead\(\)](#)
[analogReference\(\)](#)
[analogWrite\(\)](#)

Math

[abs\(\)](#)
[constrain\(\)](#)
[map\(\)](#)
[max\(\)](#)
[min\(\)](#)
[pow\(\)](#)
[sq\(\)](#)
[sqrt\(\)](#)

Trigonometry

Random Numbers

[random\(\)](#)
[randomSeed\(\)](#)

Bits and Bytes

[bit\(\)](#)
[bitClear\(\)](#)
[bitRead\(\)](#)
[bitSet\(\)](#)
[bitWrite\(\)](#)
[highByte\(\)](#)

[? Help](#)

<https://www.arduino.cc/reference/en/>

ASSIGNMENT

FOR NEXT WEEK

- Create one digital soft switch AND one analogue soft sensor
- Connect the switch AND sensor to Arduino. Read the values using the serial monitor, then use the map() or other functions to get more meaningful data.
- Connect your switch OR sensor to the Arduino to control a LED or mini vibration motor
- Integrate the switch OR sensor and the output you used above in a swatch using hard-soft connections

Question?

You can contact me: emma.pareschi@gmail.com

GOOD LUCK EVERYONE!!!!!!!!!!