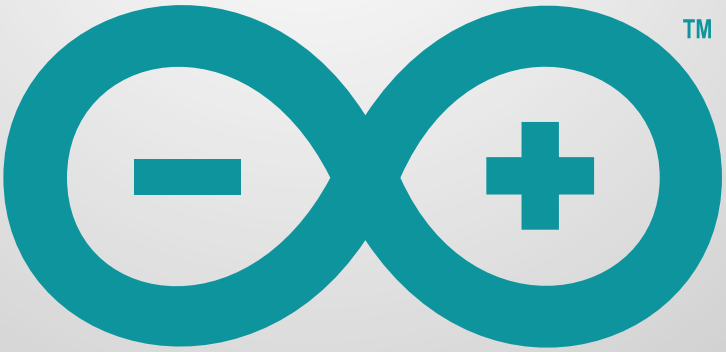
kids @migus labs

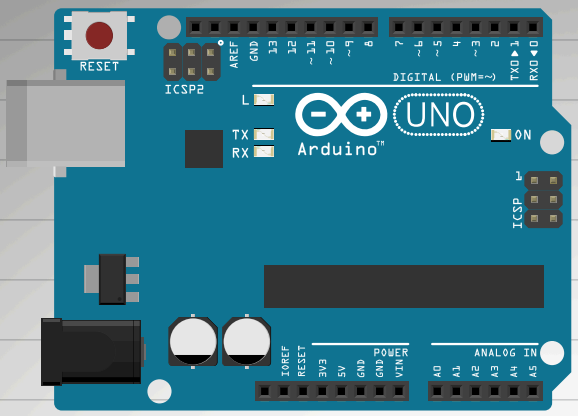
Building prototypes with Arduino



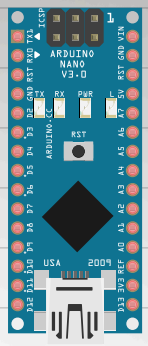
Arduino

Arduino is an open electronics platform for prototyping based on open software and hardware.

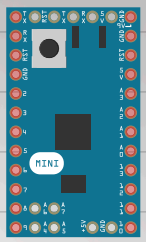
Arduino Uno



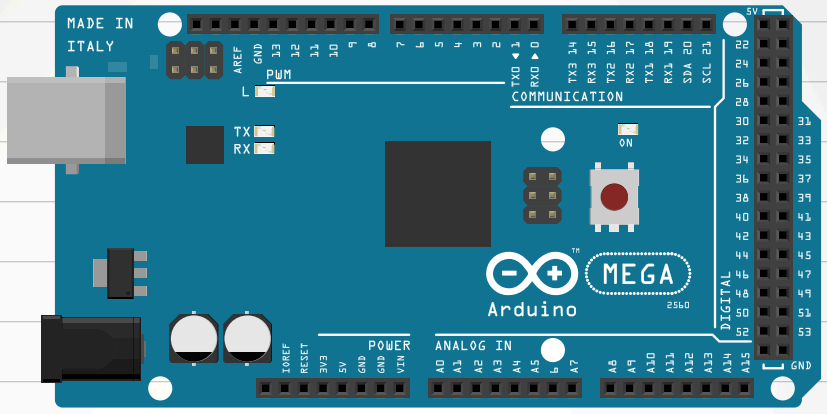
Arduino Nano



Arduino Micro



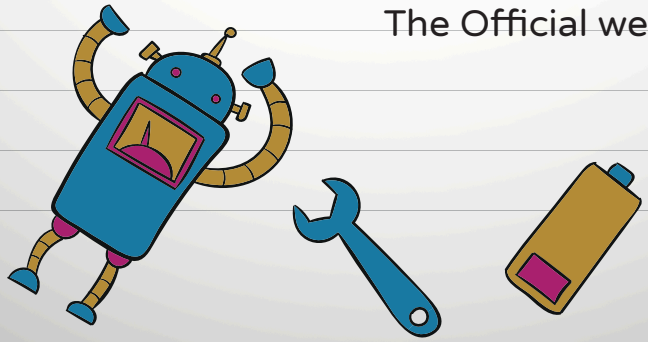
Arduino Mega



fritzing

With Arduino we can take information from the environment by connecting sensors through their input pins and act by controlling lights, motors and other actuators.

The Official website is www.arduino.cc



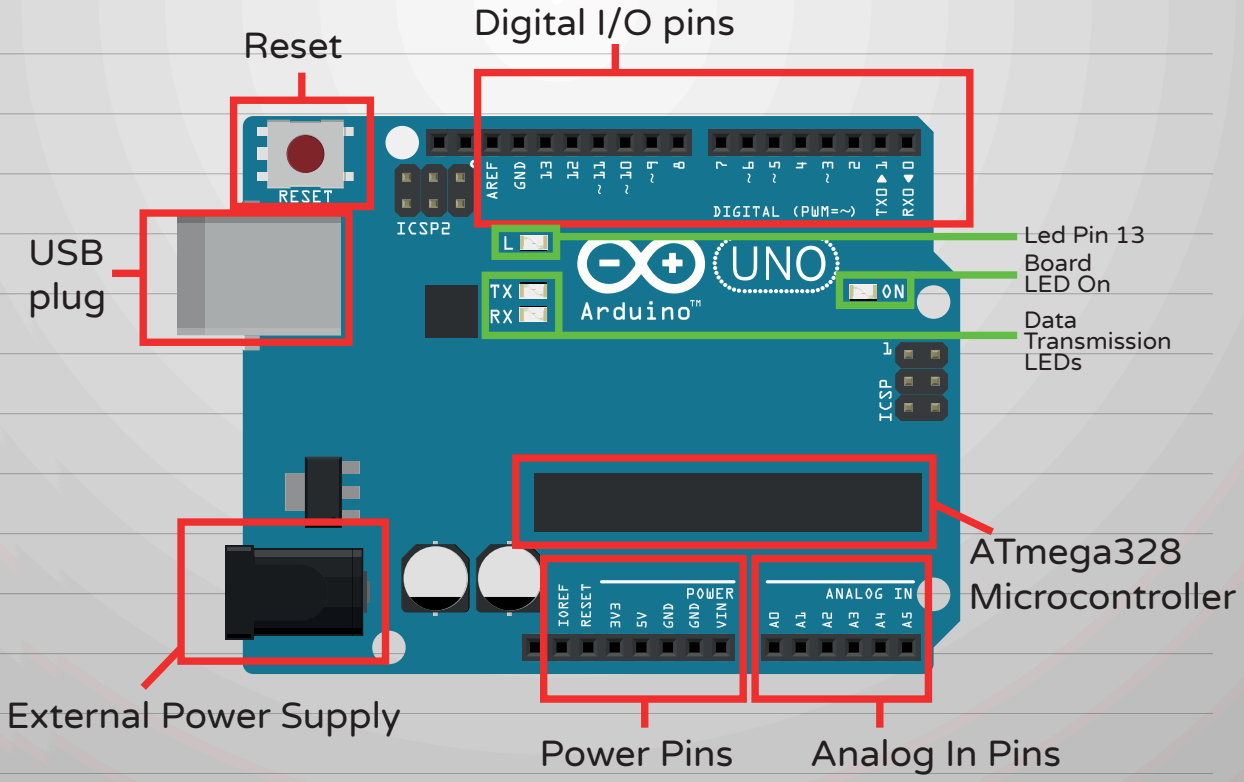
Arduino Uno Board Description

Arduino Uno is an electronic board based on the ATmega328 microcontroller.

It has 14 digital input/output pins, 6 of which can be used as PWM outputs (Pulse-Width Modulation), and other 6 analog inputs.

It also includes a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button.

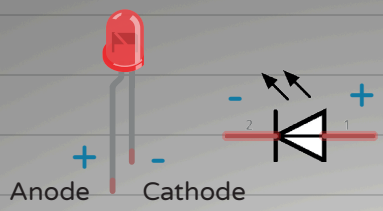
The board contains everything needed for the microcontroller to work. Simply connect it to a computer with a USB cable or power it with an adapter.



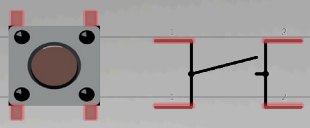
labs @migus
arte e creación dixital

Materials

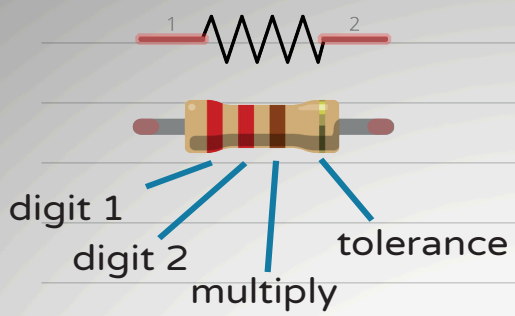
Led



Button



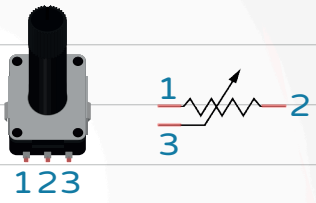
Resistencia



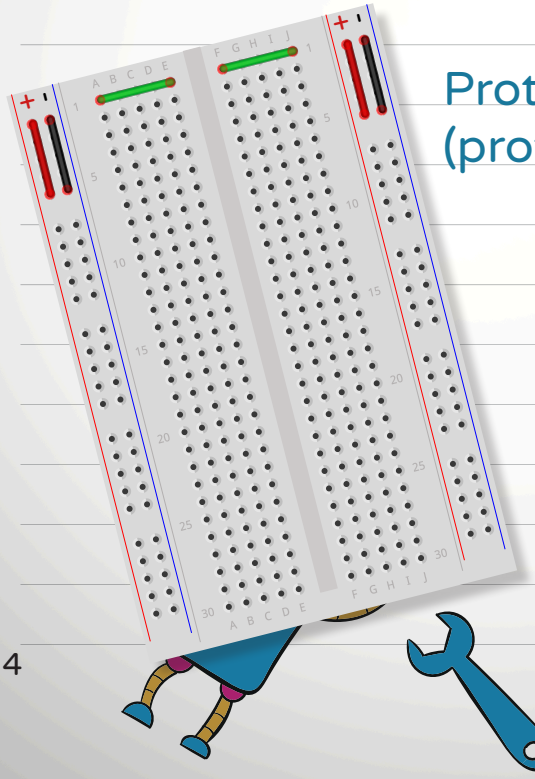
	1st Digit	2st Digit	Multiply	Tolerance
Black	0	0	x 10^0	
Brown	1	1	x 10^1	± 1%
Red	2	2	x 10^2	± 2%
Orange	3	3	x 10^3	
Yellow	4	4	x 10^4	
Green	5	5	x 10^5	± 0,5%
Blue	6	6	x 10^6	± 0,25%
Violet	7	7	x 10^7	± 0,1%
Grey	8	8	x 10^8	± 0,05%
White	9	9	x 10^9	
Gold			x 0,1	± 5%
Silver			x 0,01	± 10%

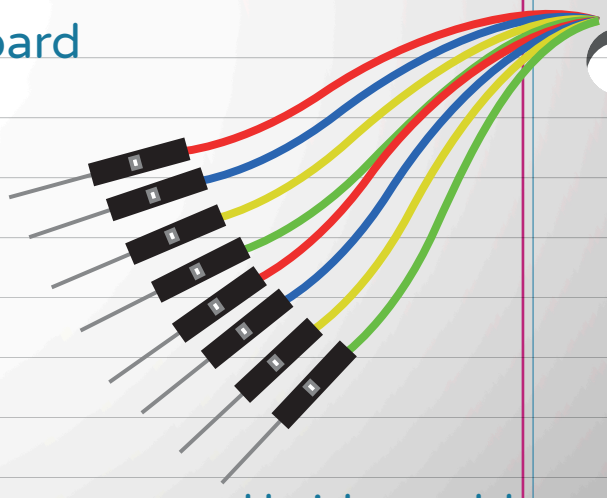
LDR Photoresistance

Potentiometer

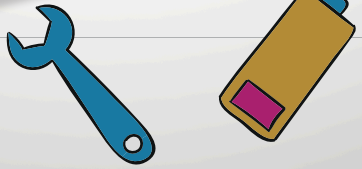


Prototype Board (protoboard)





...and bridge cables

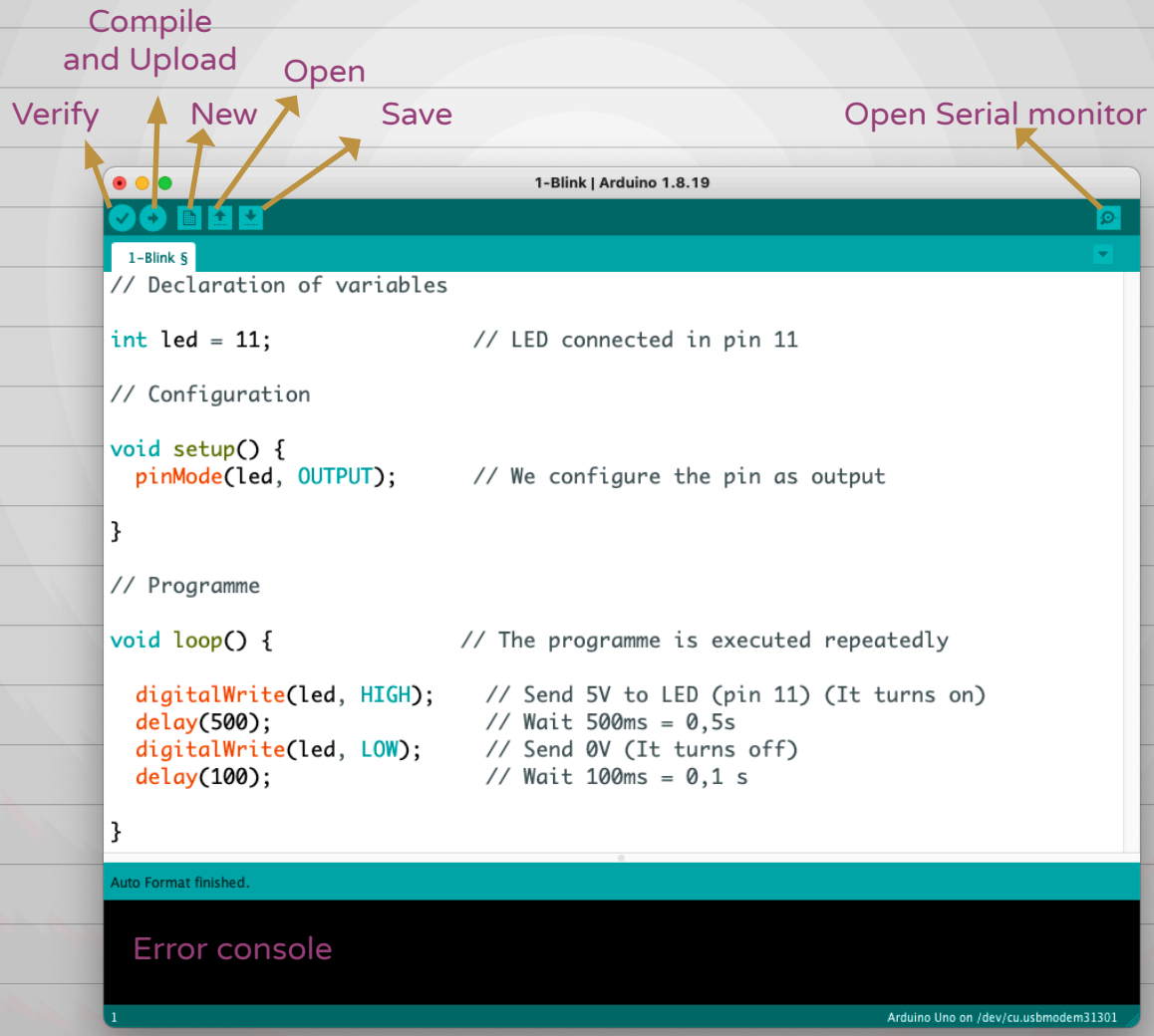


www.amiguslabs.org

Arduino Environment

Arduino's programming environment is called Arduino's IDE, which stands for Integrated Development Environment.

The files used in Arduino's environment are called sketch and they are renamed by default as: sketch+date+letter. The file containing the programme code uses the *.ino extension.

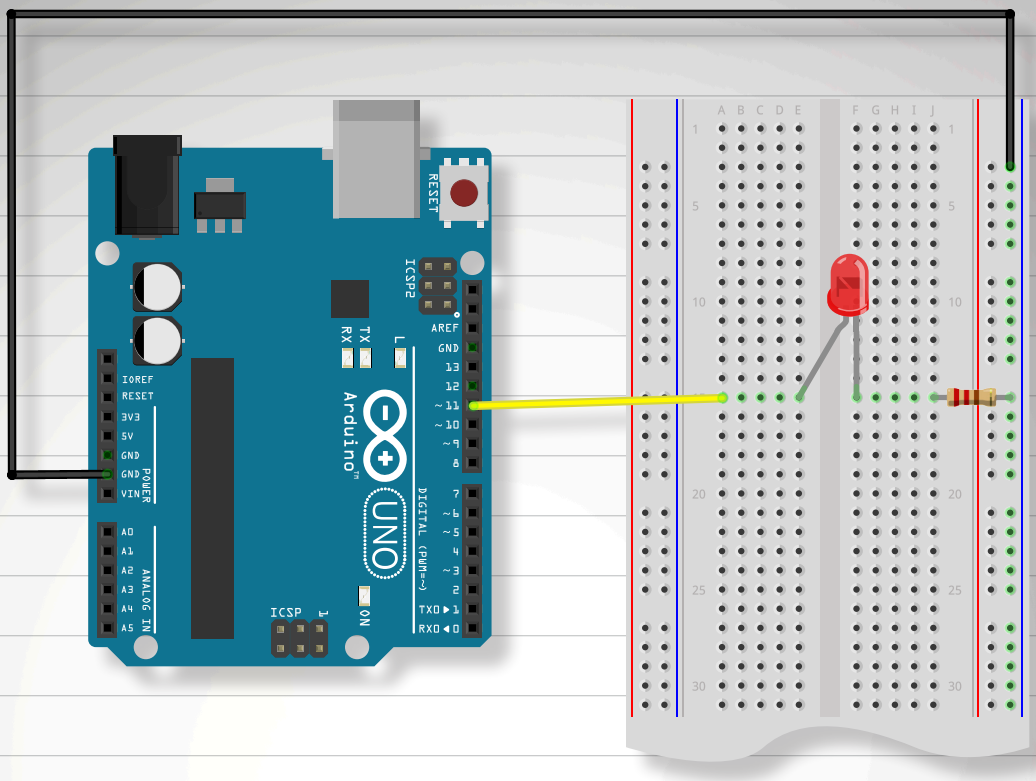


Exercise 1.- Blink

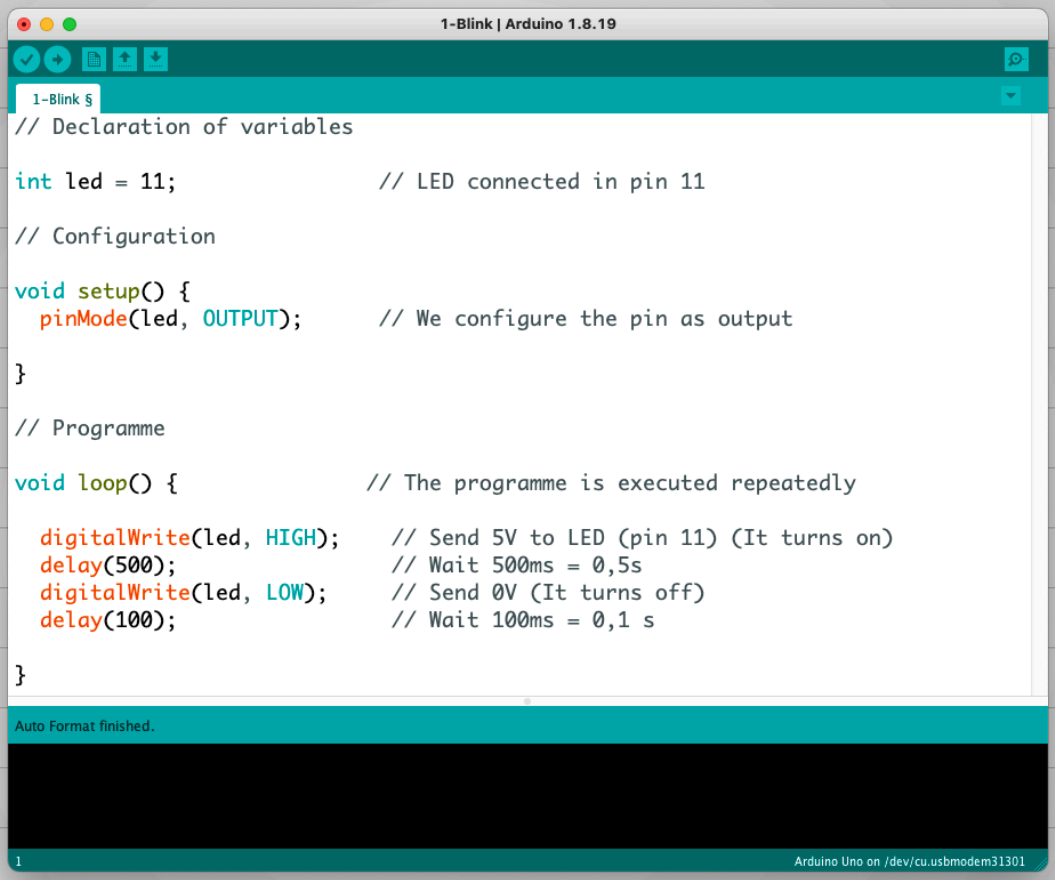
- •Configure Arduino
- Make an LED blink and change the blinking frequency

Assembly

LED in pin 11 with a protection resistance of 220Ω.



Code



Now it is your turn...

1. Try to change the blinking time.

2. Would you be able to make the LED blink just one time?

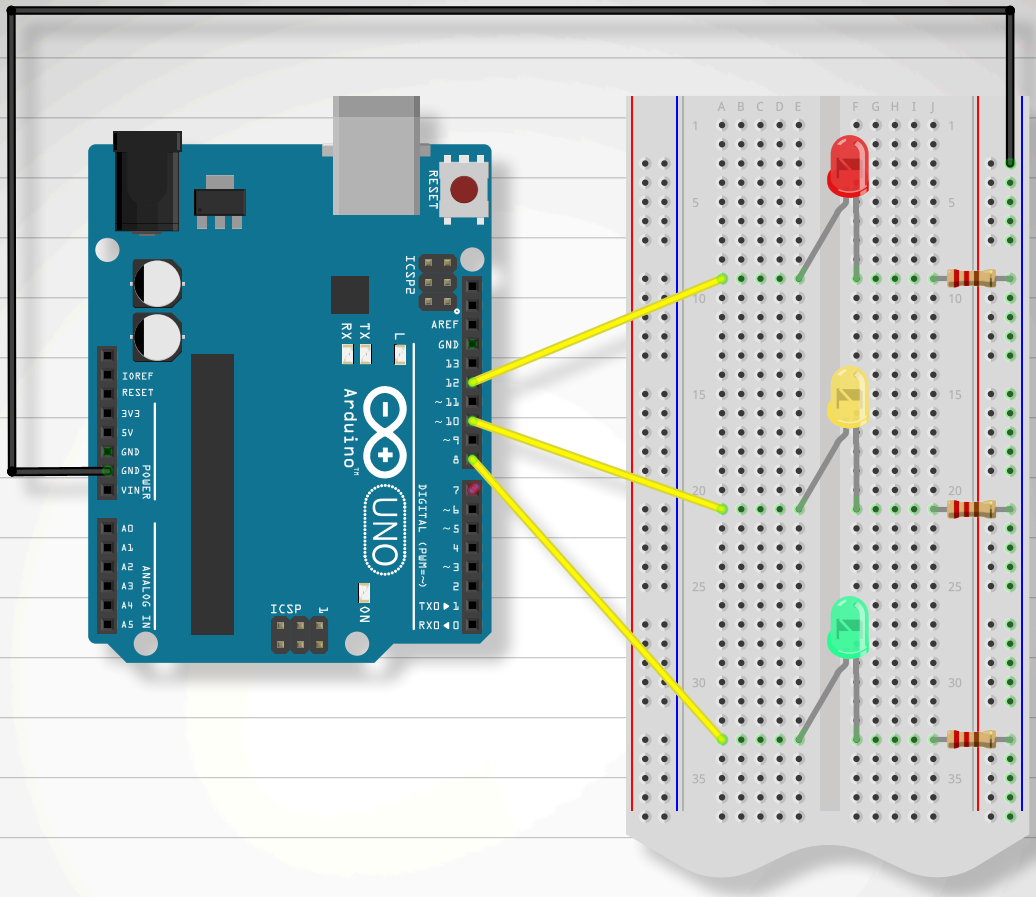
3. Use variables to define the blinking time.

Exercise 2.- Traffic Light

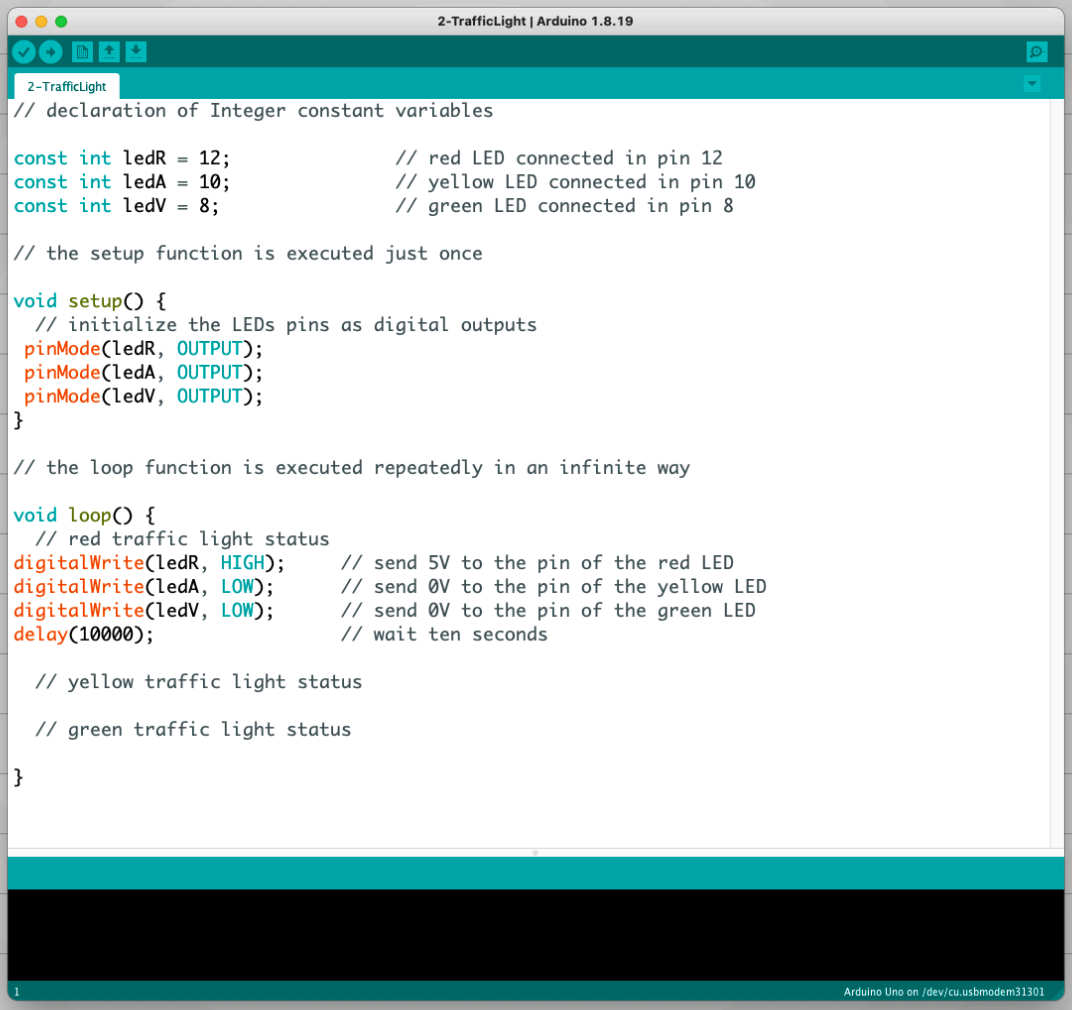
Simulate a car traffic light (red, yellow and green) working in a cyclical way.

Assembly

We have to connect 3 LEDs with their protection resistances to pins 12, 10 and 8.



Code



Now it is your turn...

1. Complete the code with the traffic light in yellow and green.

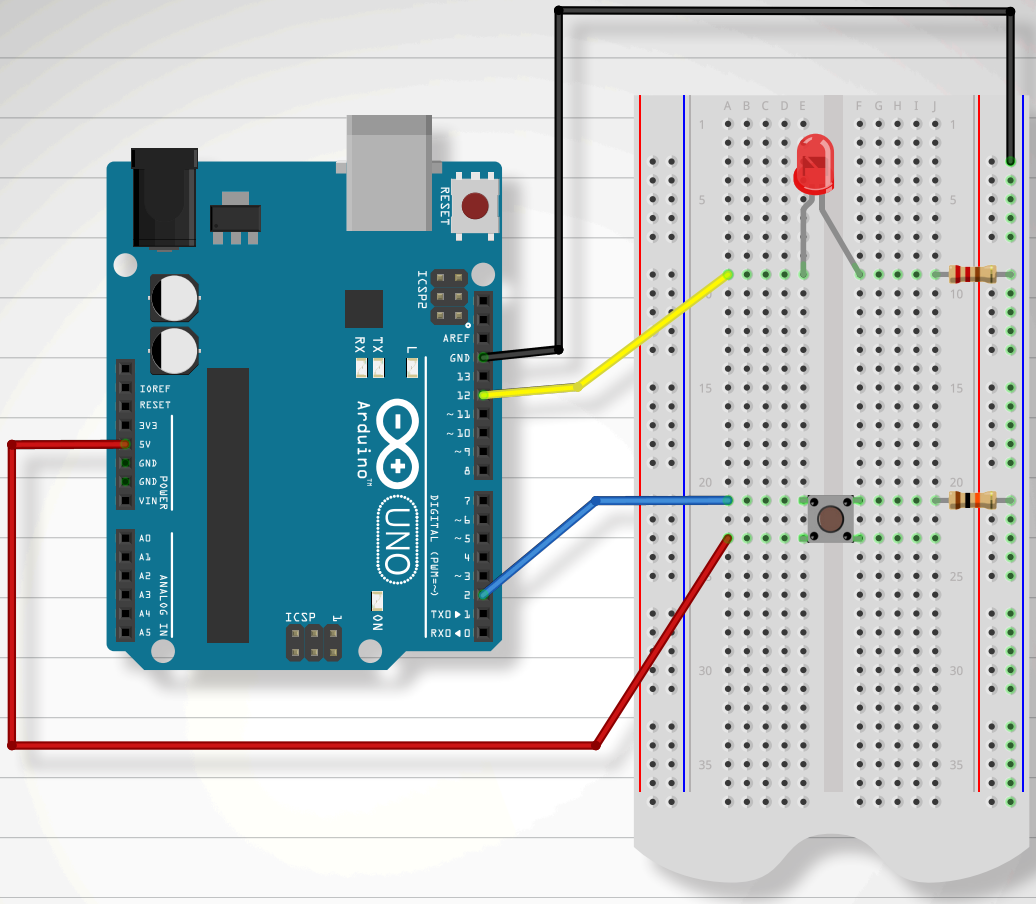
2. Add a pedestrian traffic light (red and green LEDs) synchronised with the car traffic light.

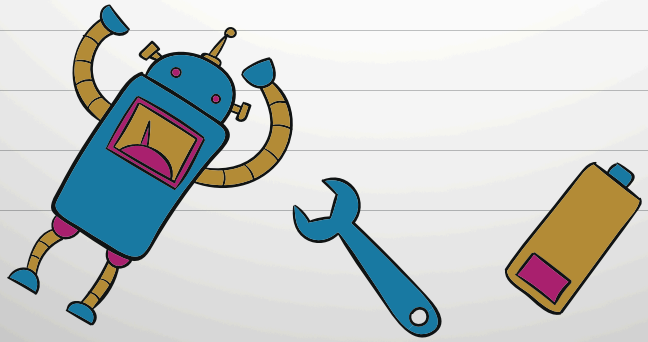
Exercise 3.- Push Button and Led

Control the on and off status of an LED with a push button, so that it turns on or off every time we press the button.

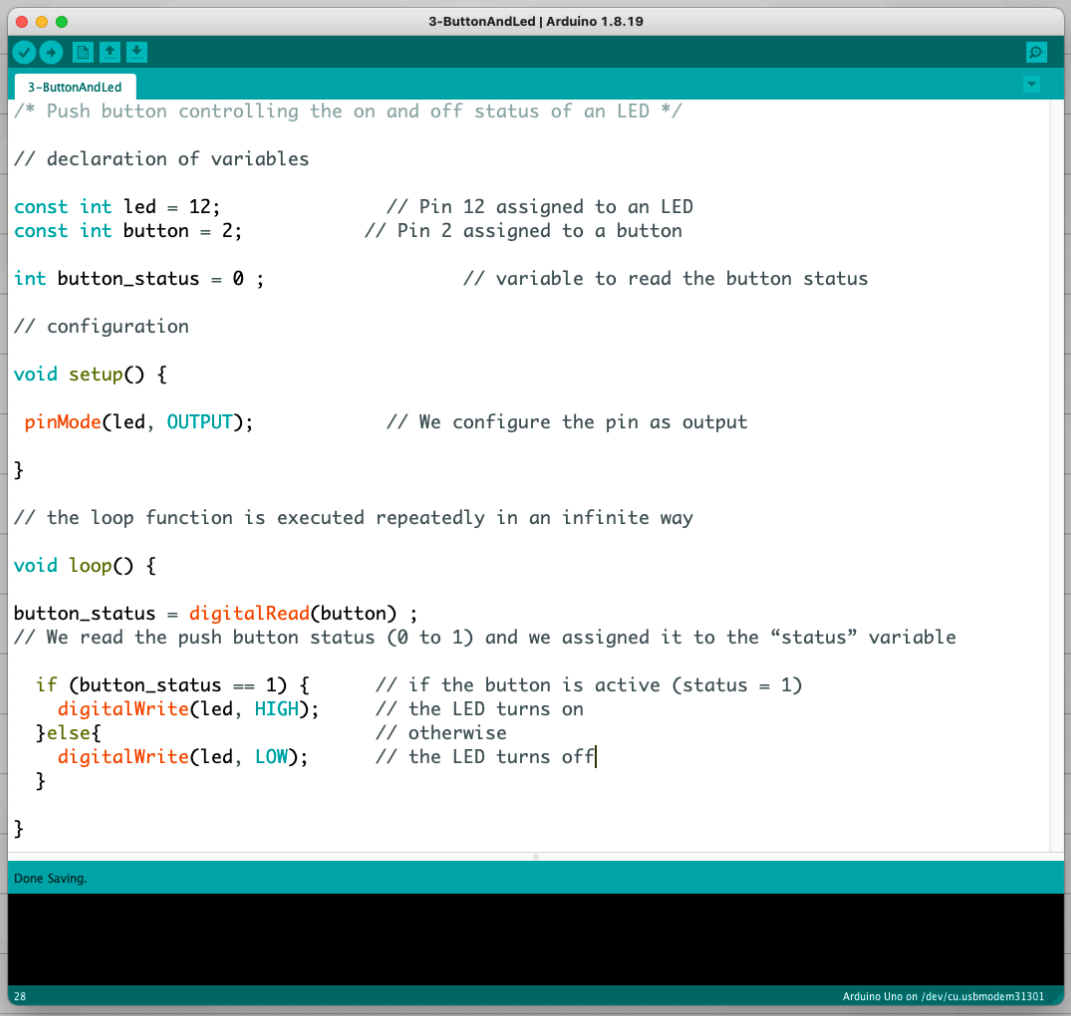
Assembly

We have to connect a push button with pin 2 with a 10 kΩ resistance as shown on the diagram.





Code



Now it is your turn...

1. Add another LED and make them turn on alternatively when pressing and releasing the push button.
2. Every time we press the push button make the LED turn on and off twice.
3. Push button with memory. If the LED was initially on, turn it off, and if it was off, turn it on.

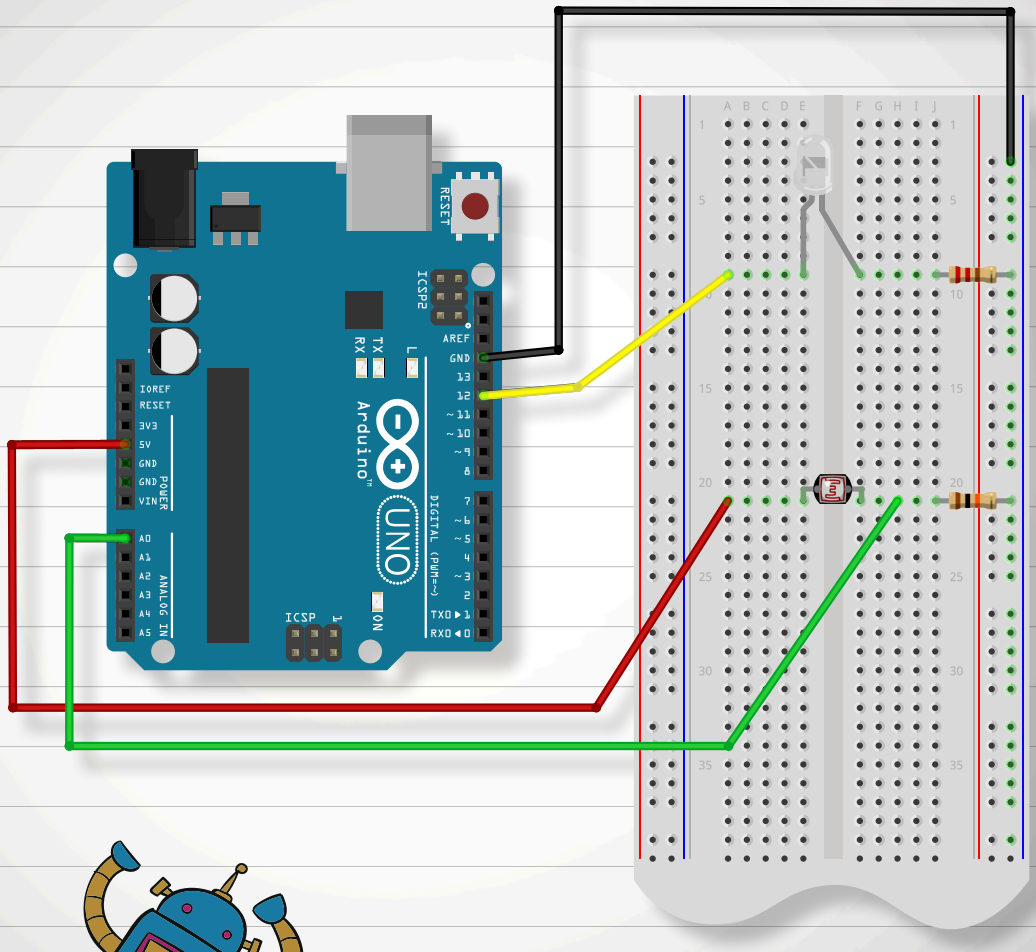
Exercise 4.- Light Switch

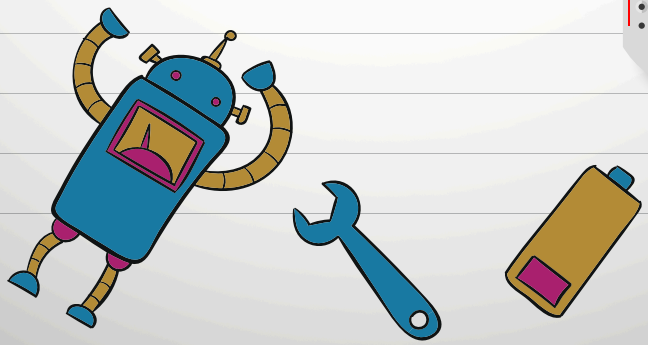
Turn on an LED depending on the light intensity that the light sensor receives. In the dark the LED is on and in high lighting is off.

First, we have to read the values provided by the sensor and secondly the different light conditions.

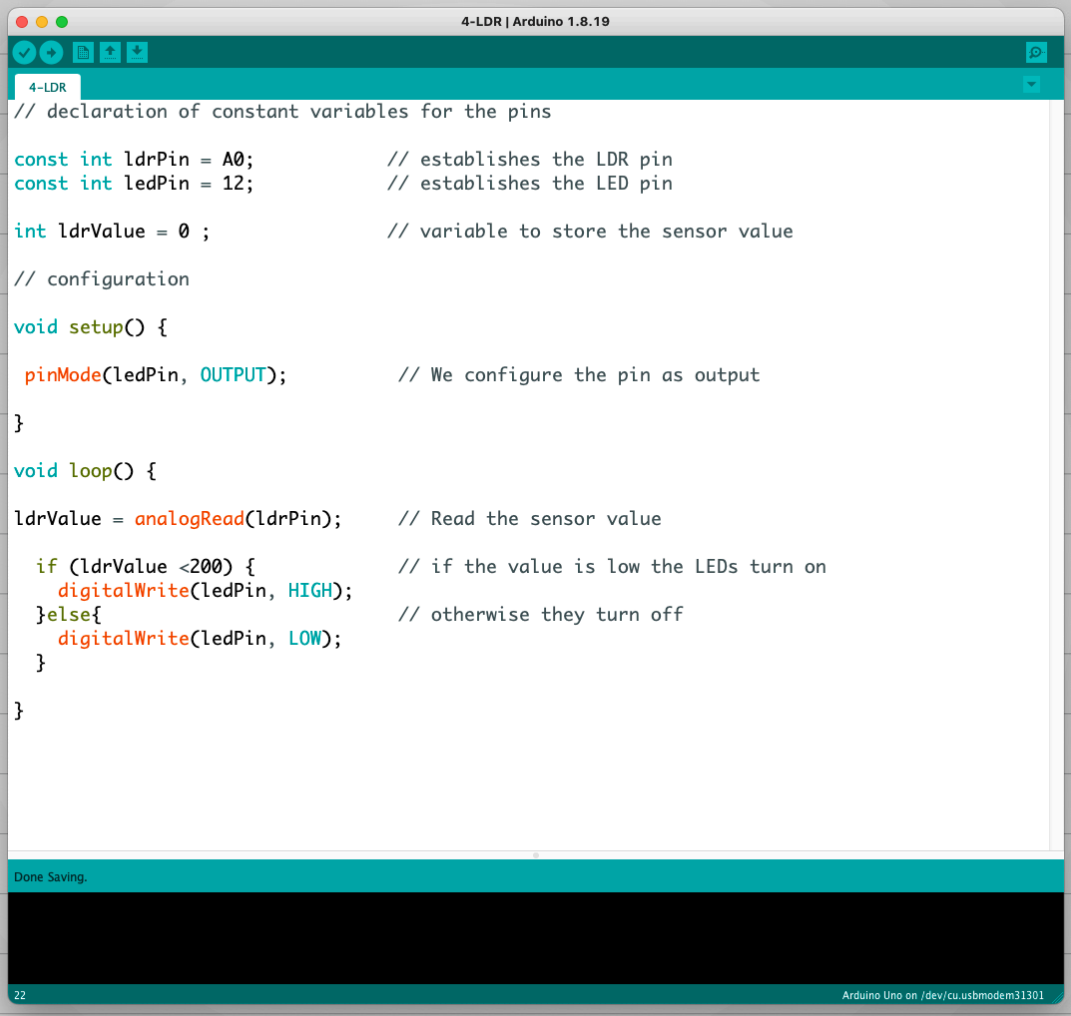
Assembly

Connect an LDR to the analog pin A0 with a 10 kΩ resistance as shown on the diagram.





Code



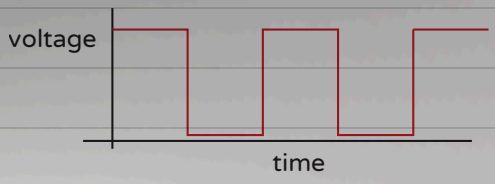
Now it is your turn...

1. Add two more LEDs in a way that you create a light scale. When there is a lot of light, they all can be off, and as soon as the light intensity diminishes more LEDs are being turned on.

2. Based on the previous proposal, make the LEDs blink when the light intensity is very low.

Exercise 5.- PWM Signals

We are going to control the brightness of one LED by sending a PWM signal that varies according to the analog reading of a potentiometer.

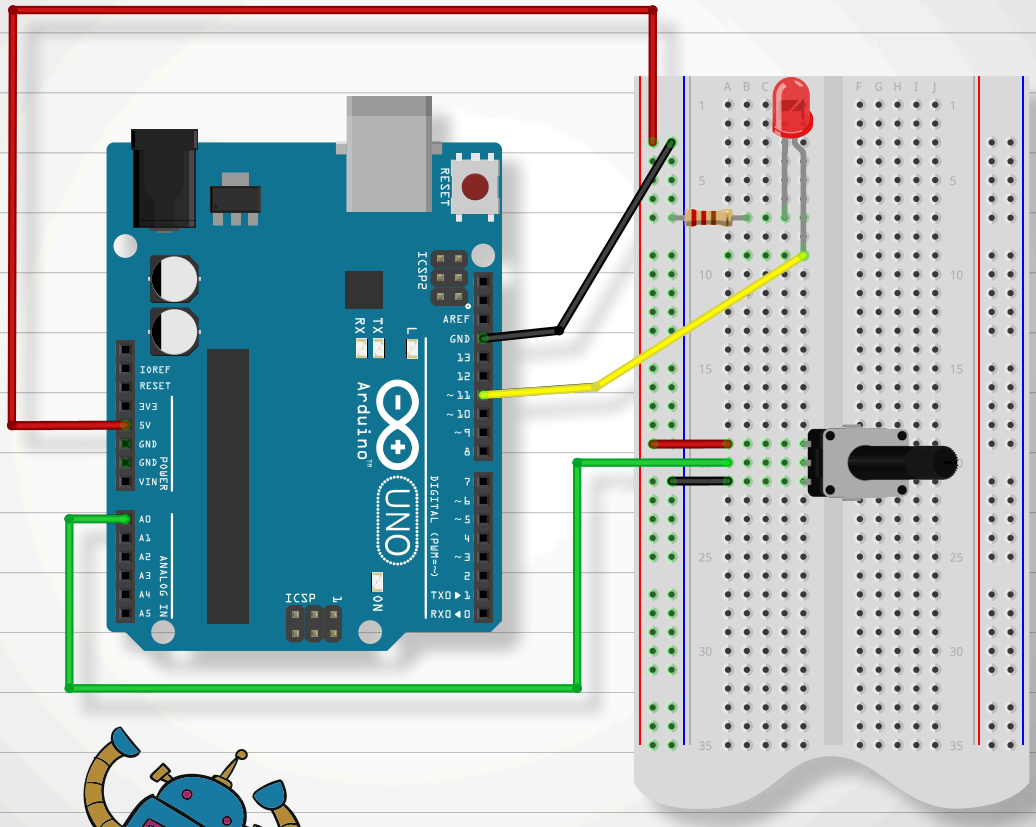


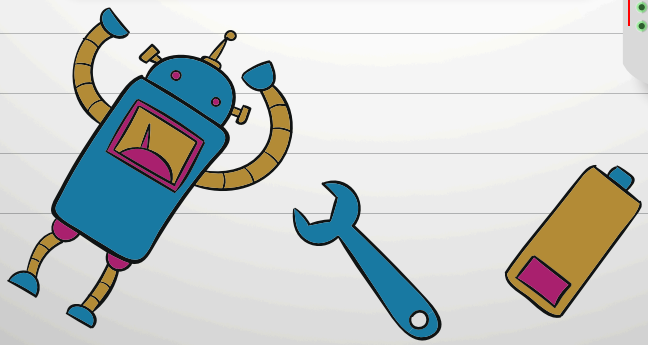
50% cycle

Assembly

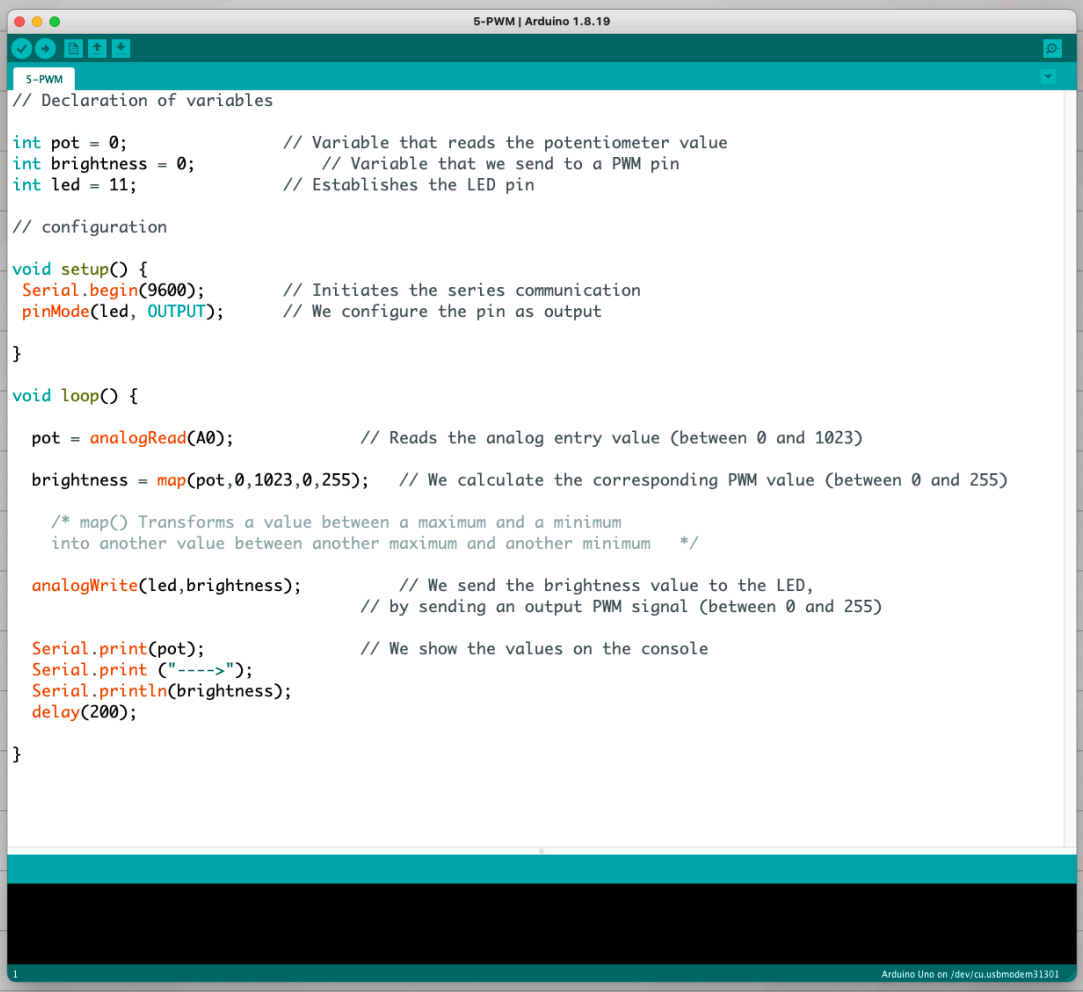
We have to connect a potentiometer with an analog input and make the reading according to the position of the cursor. The reading values will range from 0 to 1023.

We will connect an LED with its protection resistance to a digital output PWM (~). We will send it a value between 0 and 255, getting different brightness levels.





Code



Now it is your turn...

1. Simulate the effect of fire by means of an LED. You can generate an arbitrary brightness and an arbitrary waiting time for an LED.
You can use the random operator, both for the brightness and for the time. Brightness = random (20,255).

2. Create a programme where the LED blinks in a different frequency according to the potentiometer position. For example, for a 0 value in the potentiometer the blinking will be performed with a 100 ms interval, and for a 1023 value in the potentiometer the interval will be 1000 ms. Use a map function in order to transform the analog interval into the time interval.

Exercise 6.- Loops

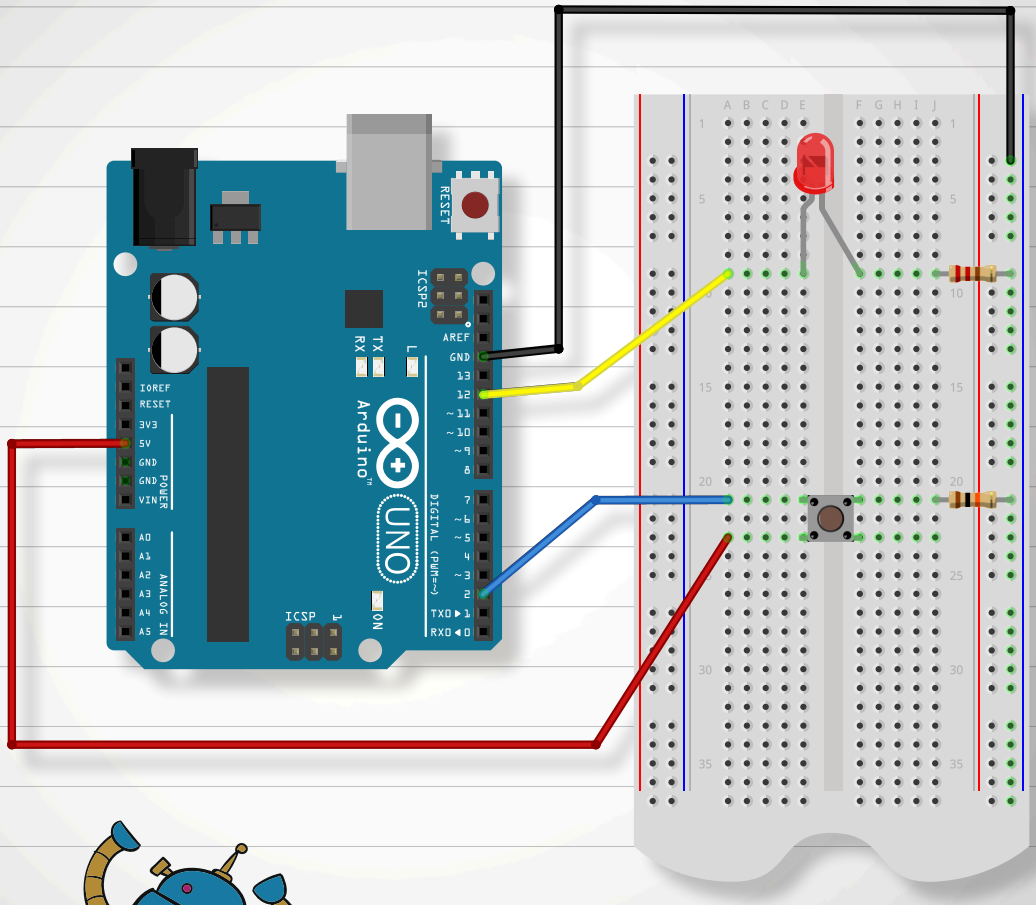
In this example we will make some instructions be repeated a determined number of times.

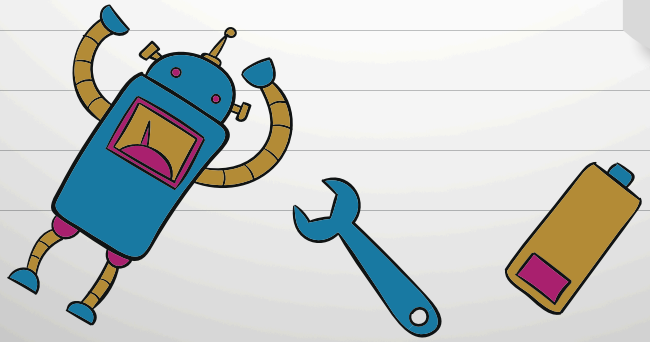
For example, we will make the LED turn on and off several times, we wait some time and we repeat the process.

Assembly

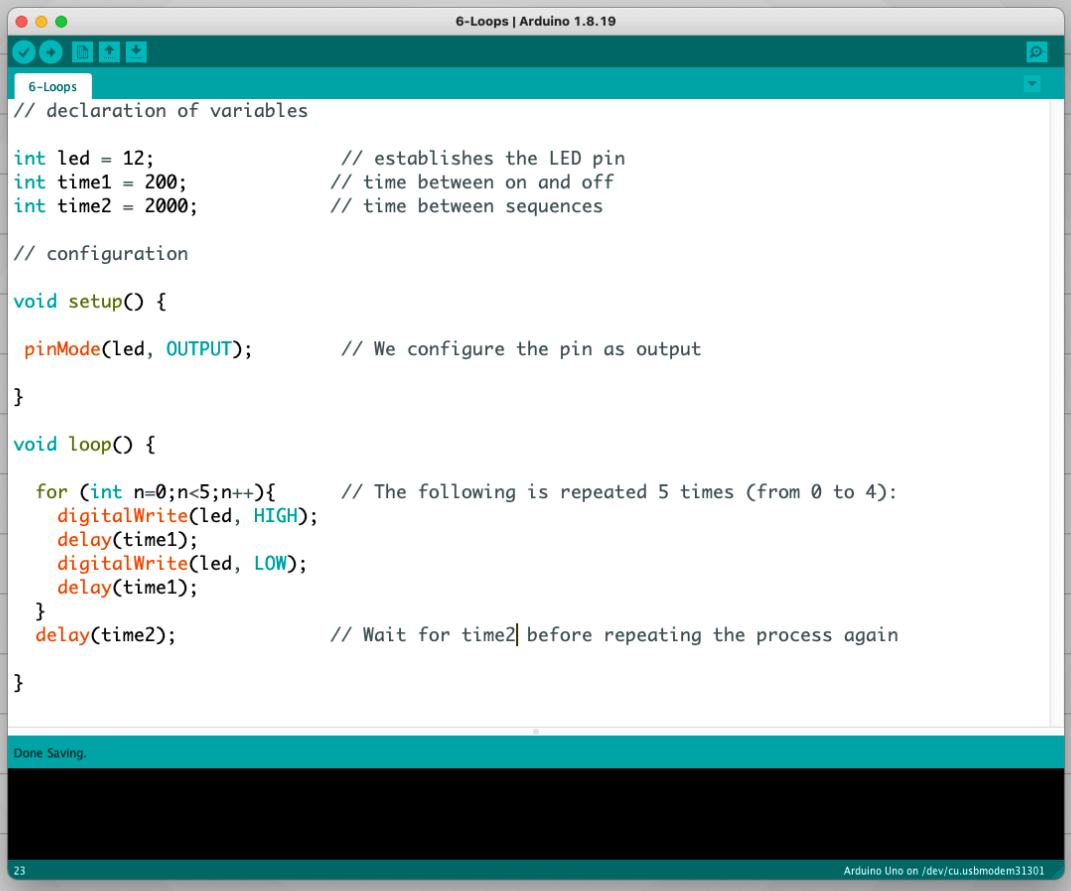
We need an LED and a push button with their respective protection resistances. For this case we will use the function:

for (start, condition, modifier) {process}





Code



Now it is your turn...

1. Connect a second LED and in each cycle make the first LED blink 6 times and the second LED three times.

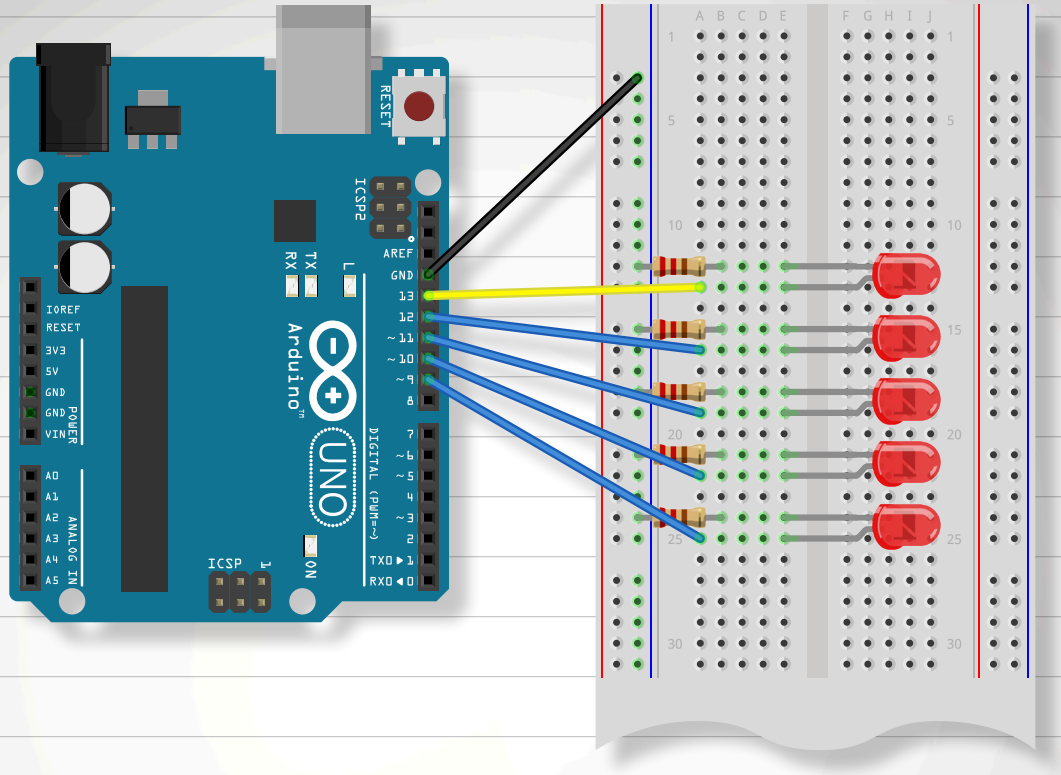
2. Make possible that every time we press the push button a LED turns on and off three times. When the push button is not active, the LED remains off.

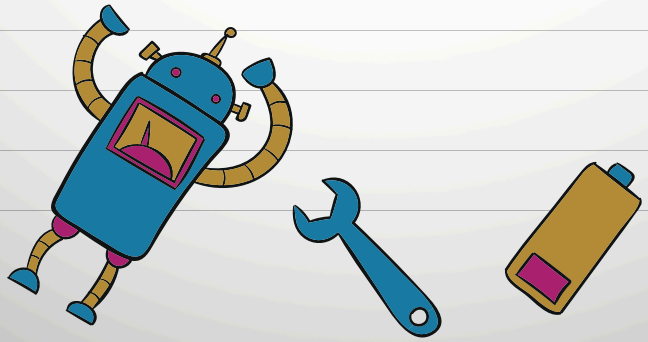
Exercise 7.- Knight Rider

We will simulate the car from “Knight Rider” making a series of LEDs turn on and off in a consecutive order.

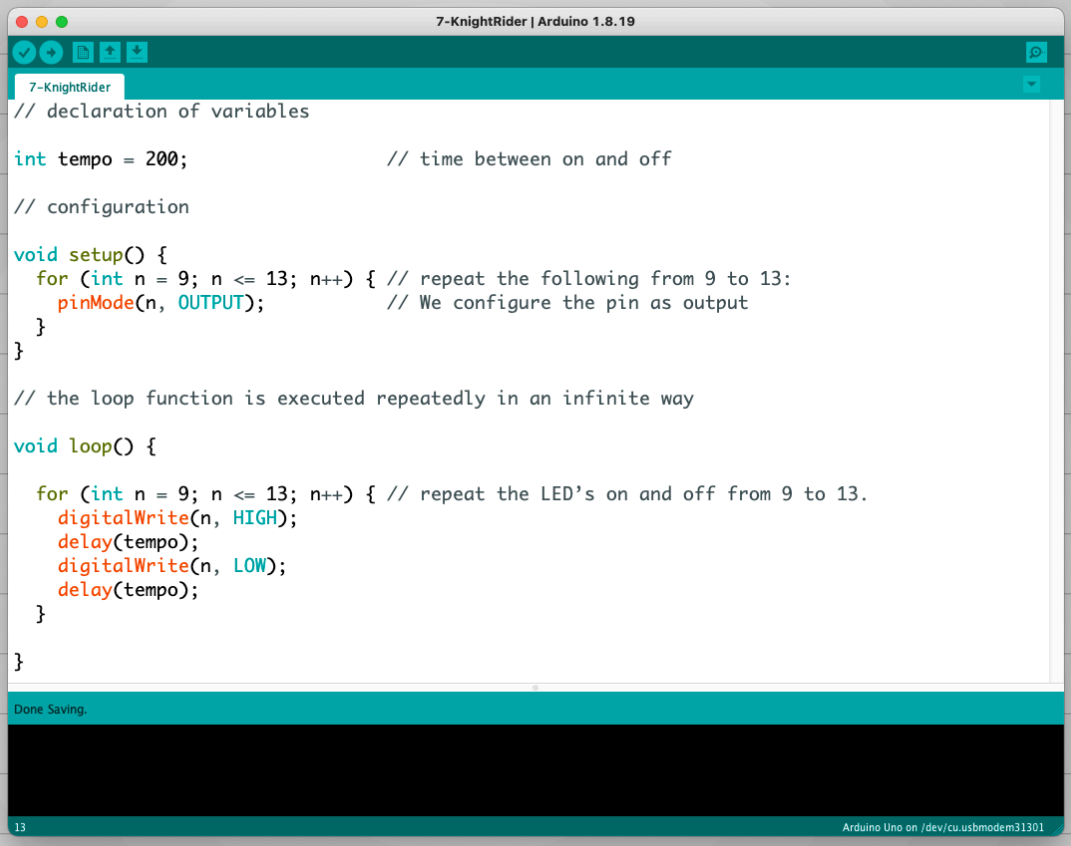
Assembly

We have to connect 5 LEDs with their corresponding protection resistances in pins 9 to 13.





Code



Now it is your turn...

1. Add a descending loop to the previous programme so that the sequence takes place first in one way and then in another.

2. Make the LEDs turn on consecutively without turning off, and then, turn them off one by one in the same way or the opposite one.

3. We can control the sequence speed with a potentiometer connected into an analog input. Use the map function to transform the analog range (from 0 to 1023) into a time range (for example from 50 ms to 1000 ms).

labs @migus

arte e creación dixital

Sponsored by:

XUNTA DE GALICIA

Developed by:

@migus Asociación Informática www.amigus.org

labs @migus www.amiguslabs.org

FAB LAB AS PONTES

Support

Concello de As Pontes · COHEMPO · EuroEume · Nauta · NO PING · nodo outsourcing · TRIADE · SEVERIANO AGROCOMERCIAL · endesa · escourido & asociados · Clip Carpono · Ferretería Marcos · Artes Gráficas CARLIN · grupo Matojoma · YOU CAN SCHOOL of English · fitter · Openor · PONTESA · Desguaces Vilabella · Óptica Carmela · Servicio Móvil · ARECO construimos calidad de vida · ARPROSA · Establecimientos RADIO ETER · Fabián · don Dino · CONPLC · icoEume · PRELUDIO MUSICAL · xestoría coello

www.amiguslabs.org