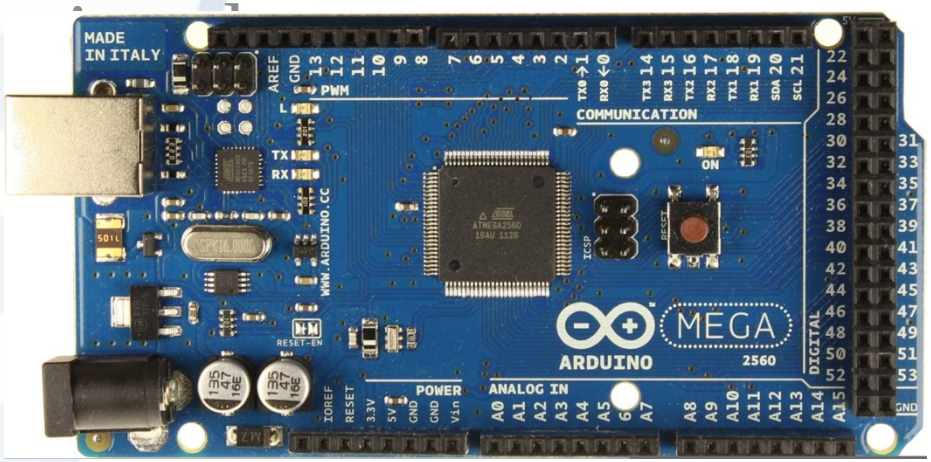
Arduino

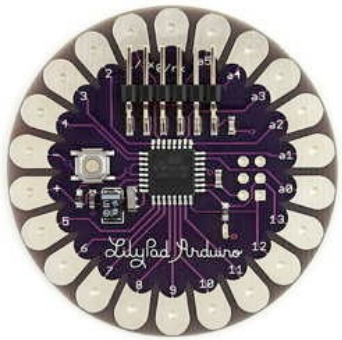
Introducción

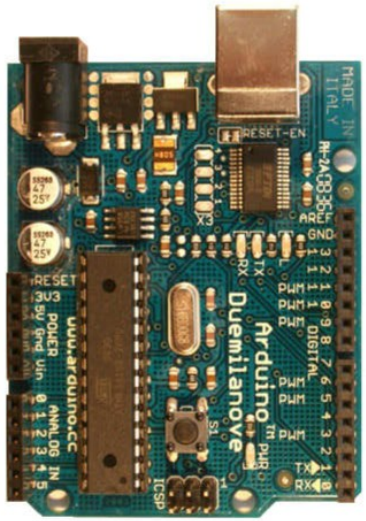
¿Qué es arduino?

Es una plataforma de hardware de código abierto, basada en una sencilla placa con entradas y salidas, analógicas y digitales.

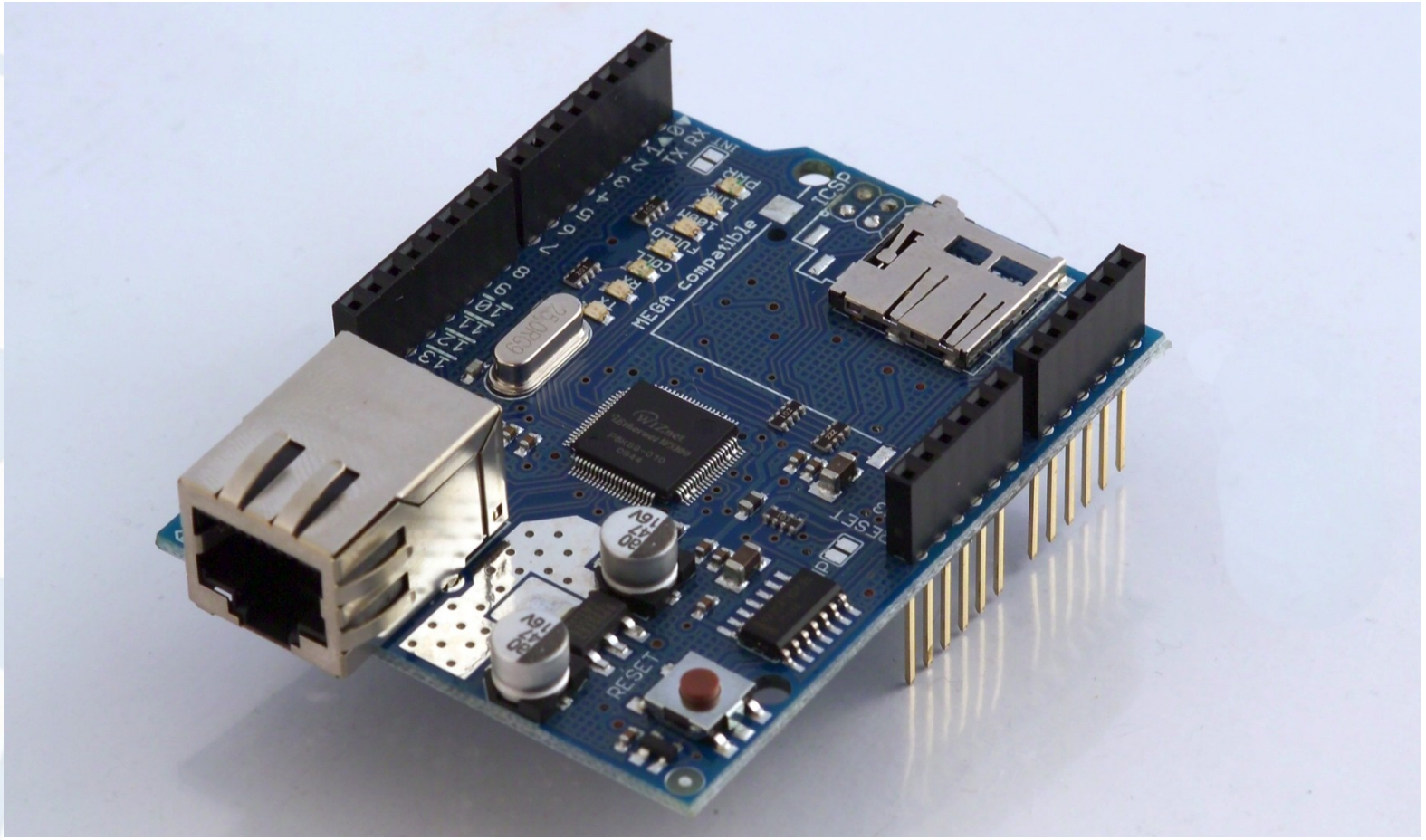
En resumen es un dispositivo que conecta el mundo físico con el mundo



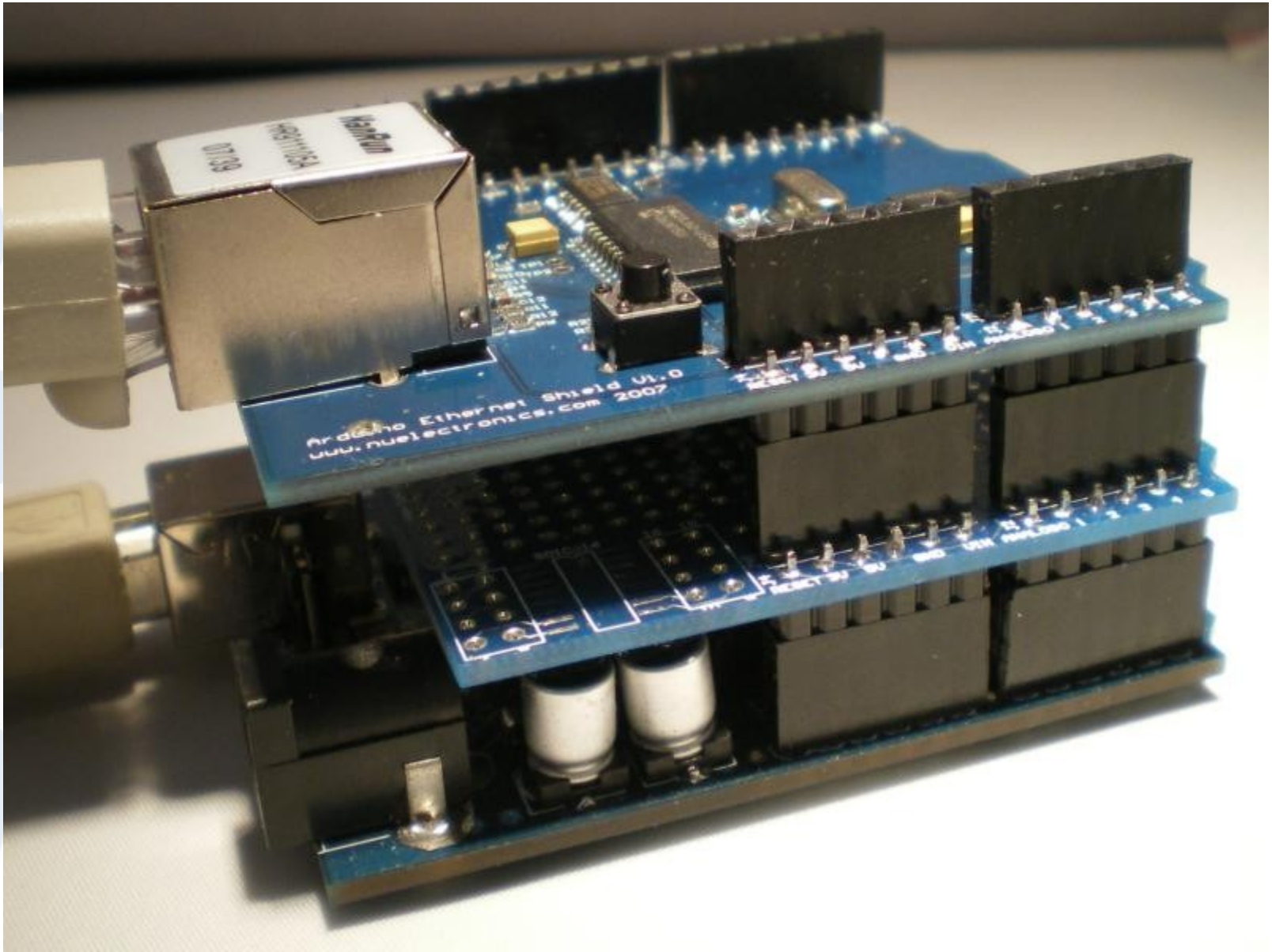




Shields

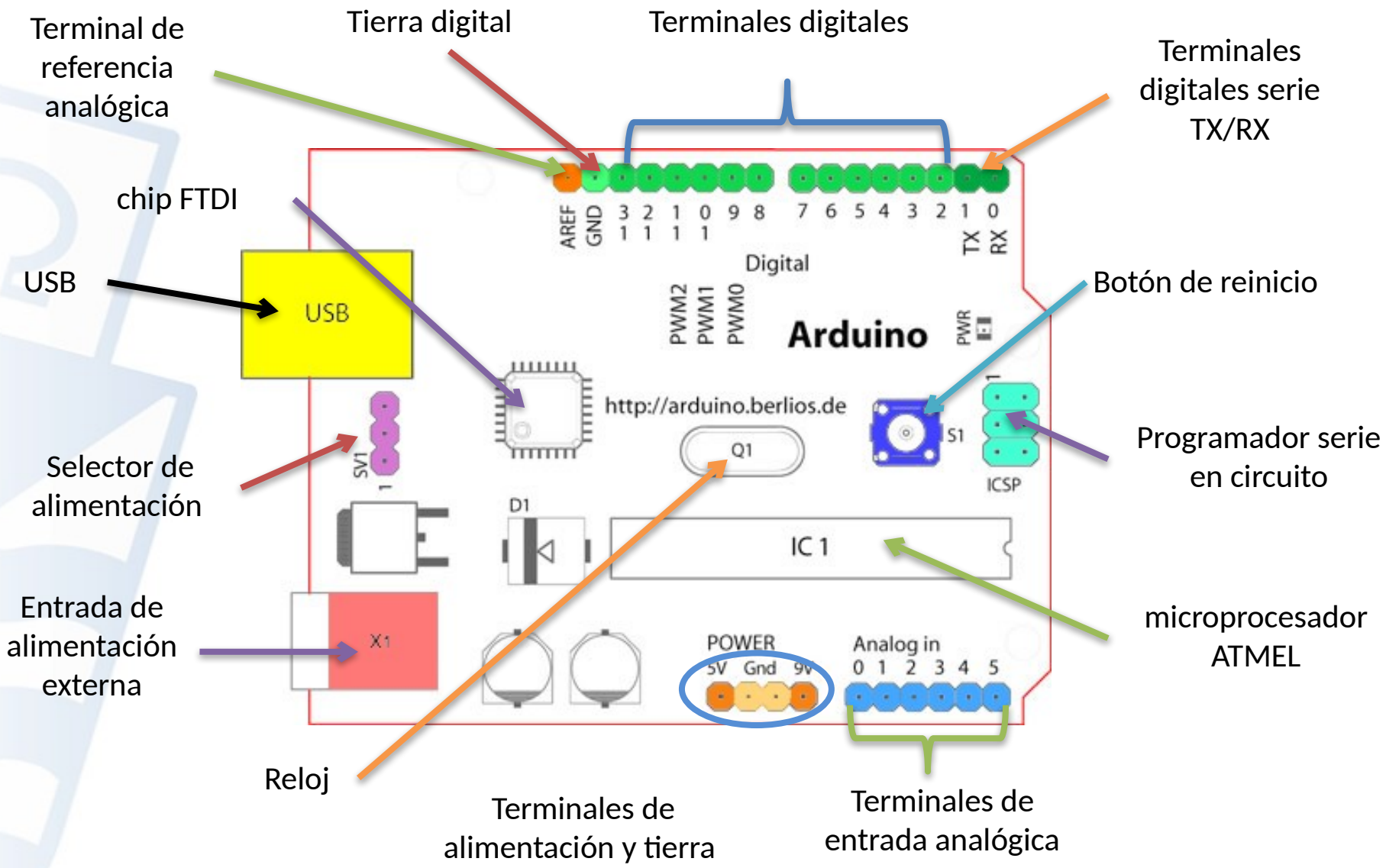


Shields

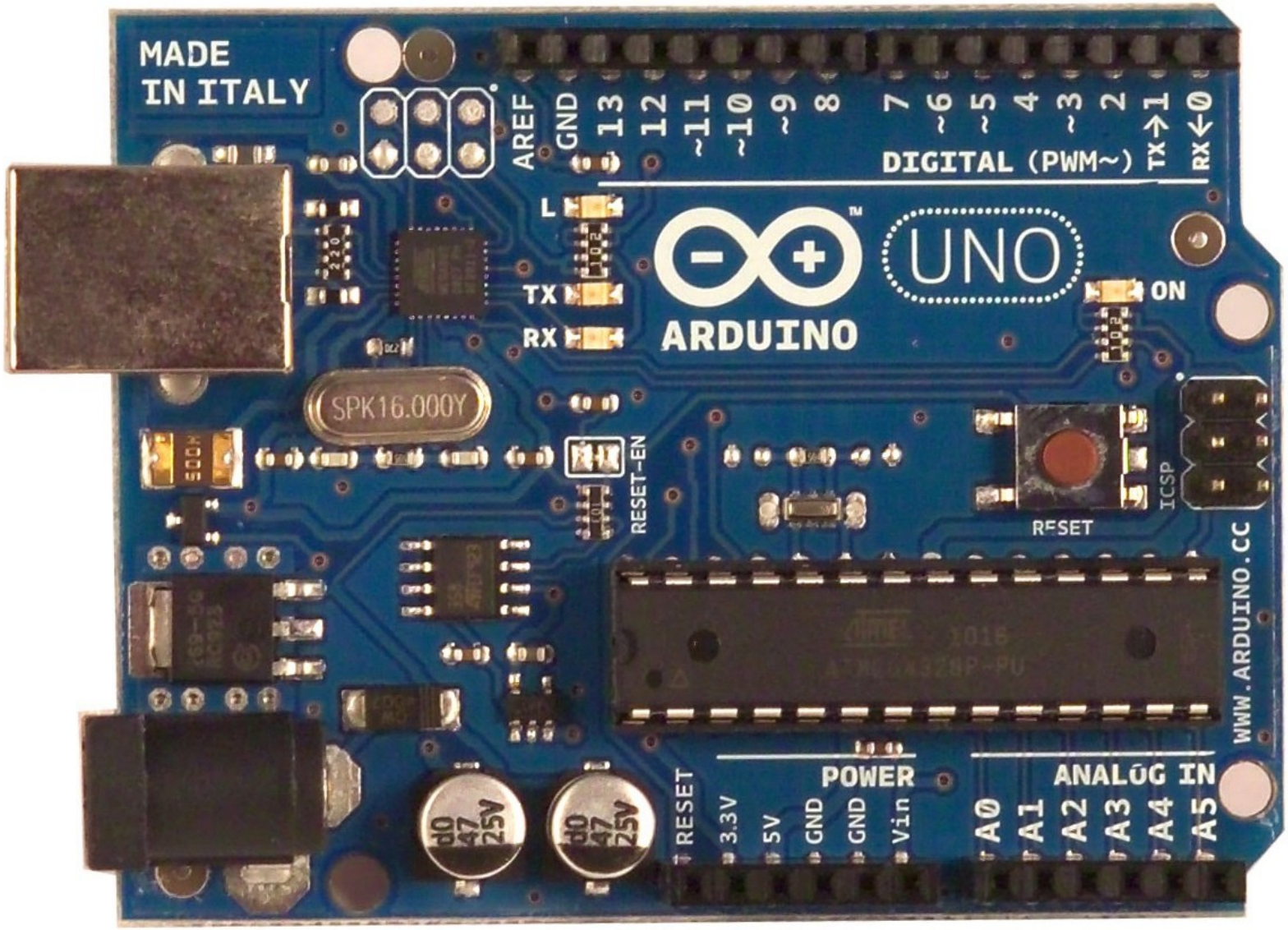


Hardware del arduino

Estructura


Terminal de referencia analógica
Tierra digital
Terminales digitales
Terminales digitales serie TX/RX
chip FTDI
USB
USB
Botón de reinicio
Selector de alimentación
Programador serie en circuito
Entrada de alimentación externa
microprocesador ATMEL
Reloj
Terminales de alimentación y tierra
Terminales de entrada analógica
AREF GND 3 2 1 0 9 8 7 6 5 4 3 2 1 0
TX RX
Digital
PWM2 PWM1 PWM0
Arduino
PWR
http://arduino.berlios.de
Q1
S1
ICSP
SV1
D1
IC 1
X1
POWER
5V Gnd 9V
Analog in
0 1 2 3 4 5

Estructura

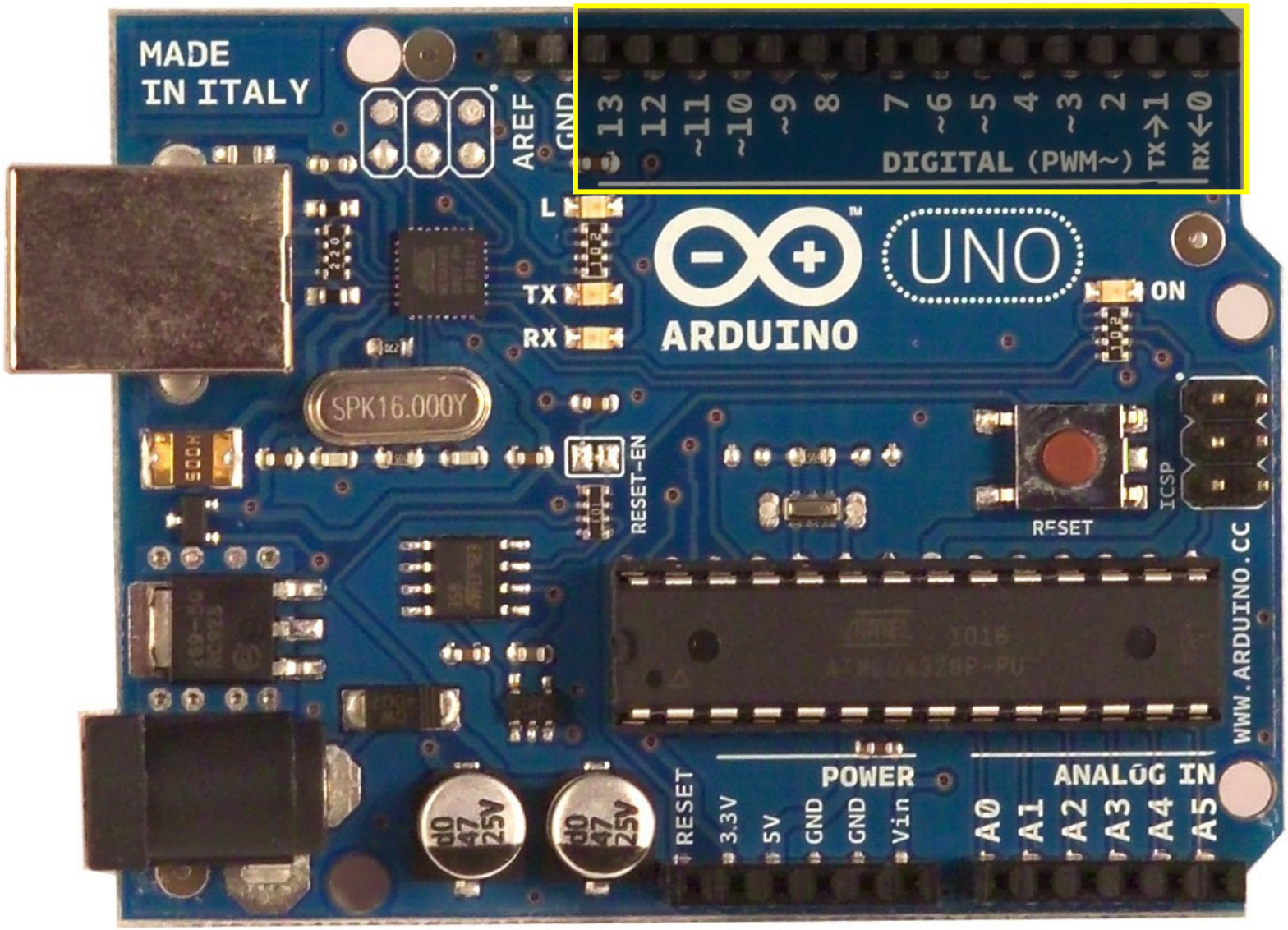

MADE
IN ITALY
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
SPK16.000Y
RESET-EN
RESET
ICSP
WWW.ARDUINO.CC
POWER
ANALOG IN
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

Codificar en Arduino

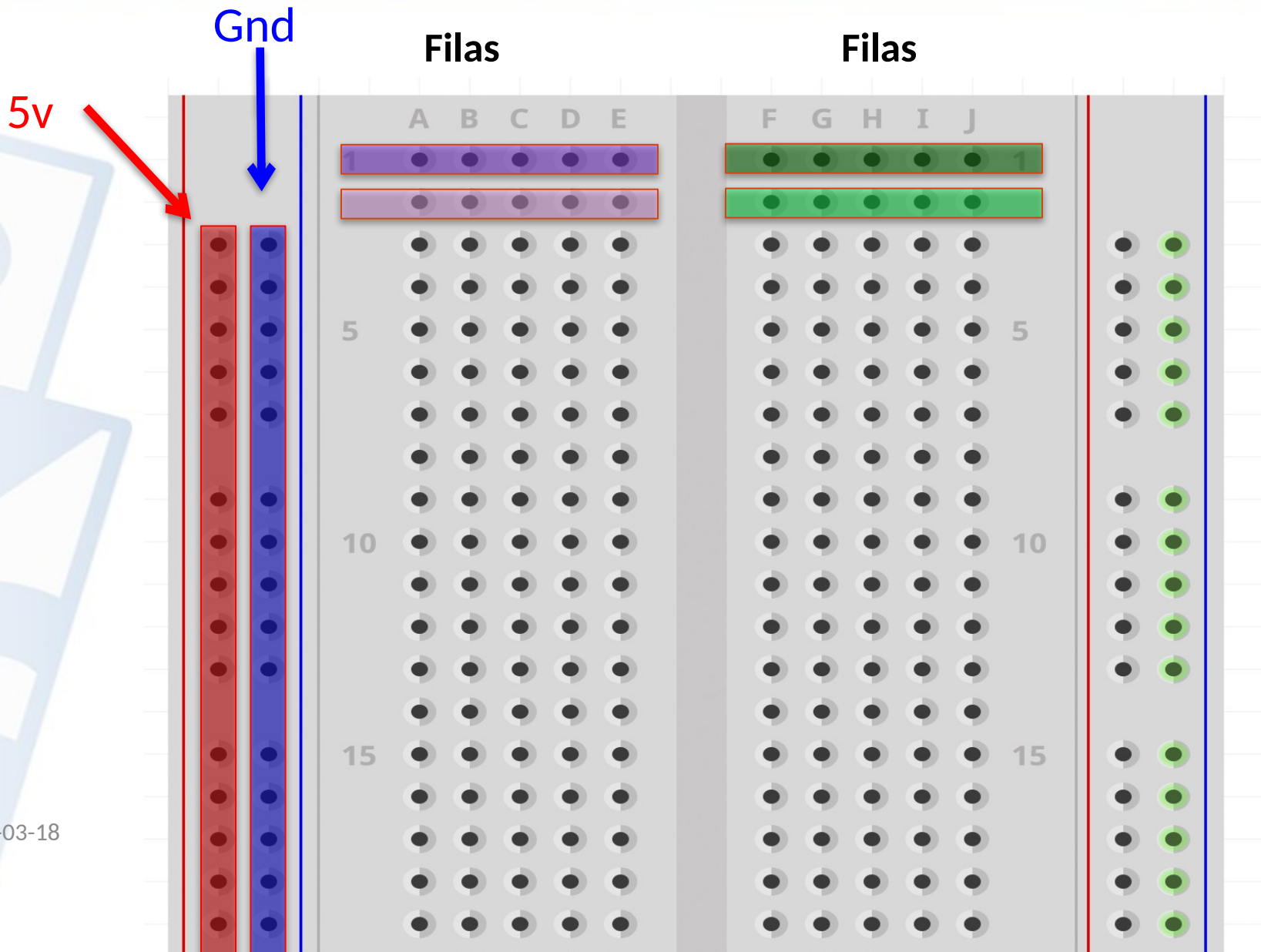
```
//Set variable

void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Ejemplos

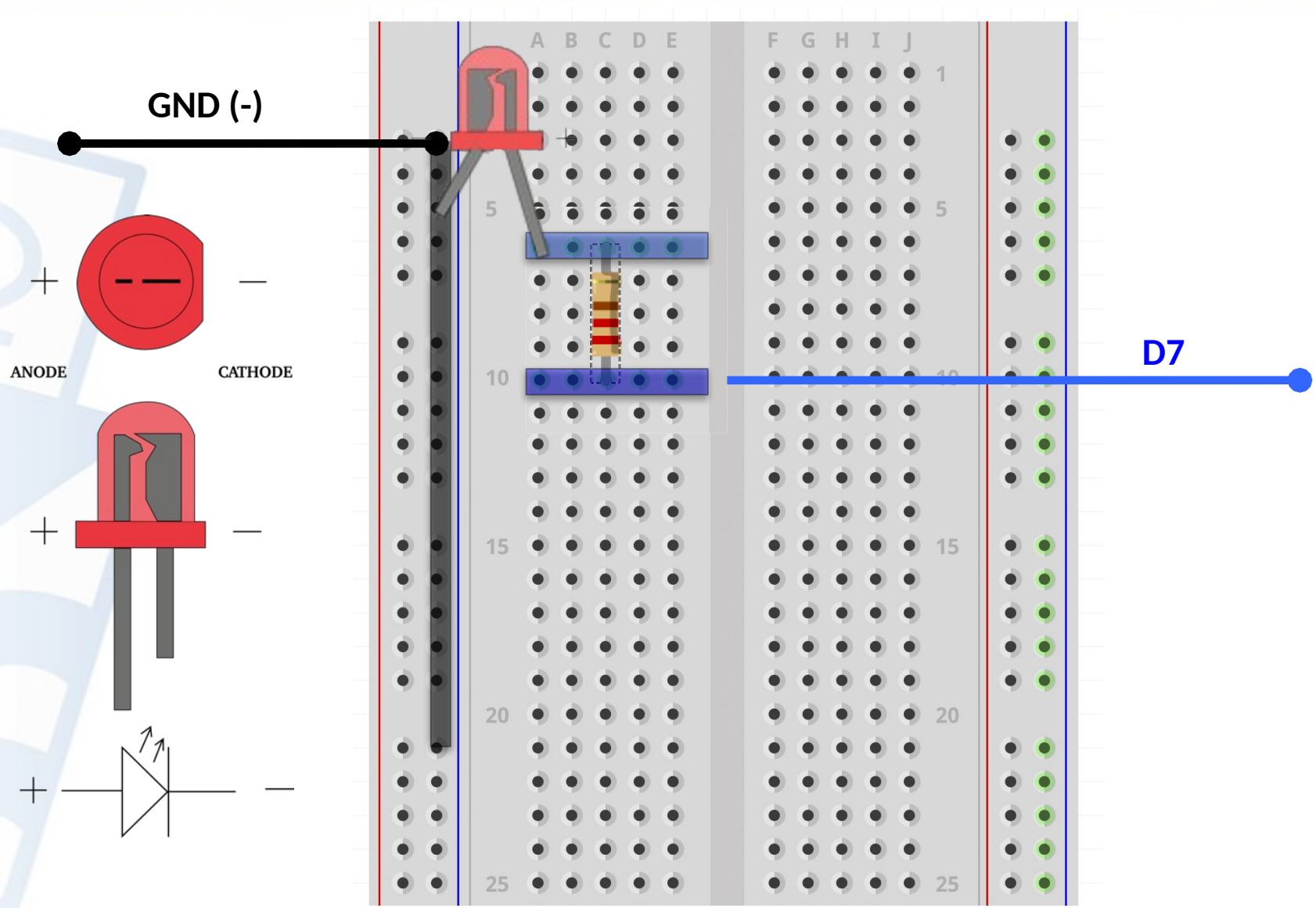
Entradas y salidas digitales



Breadboard



Salida Digital

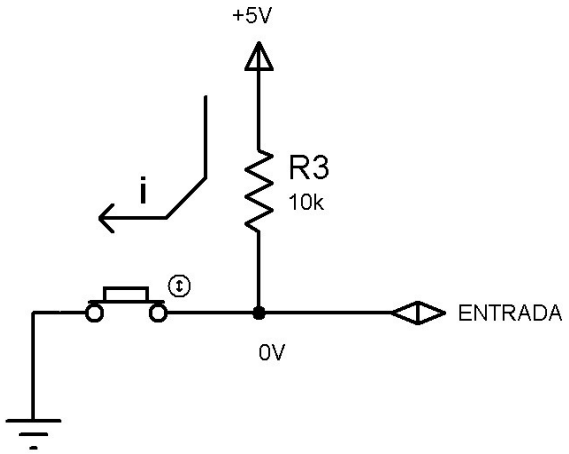


Codigo para Salida Digital

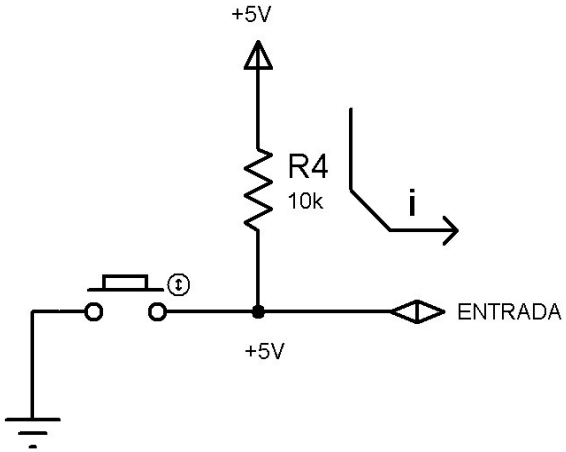
```
void setup() {
  // Iniciar el PIN en modeo de Salida
  pinMode(7,OUTPUT);
}

void loop() {
  // Poner un HIGH (True, 1) en la salida (5v)
  digitalWrite(7,HIGH);
  delay(1000);
  // Poner un LOW (False, 0) en la salida.
  digitalWrite(7, LOW);
  delay(1000);
}
```

Entrada Digital (Pull_UP)

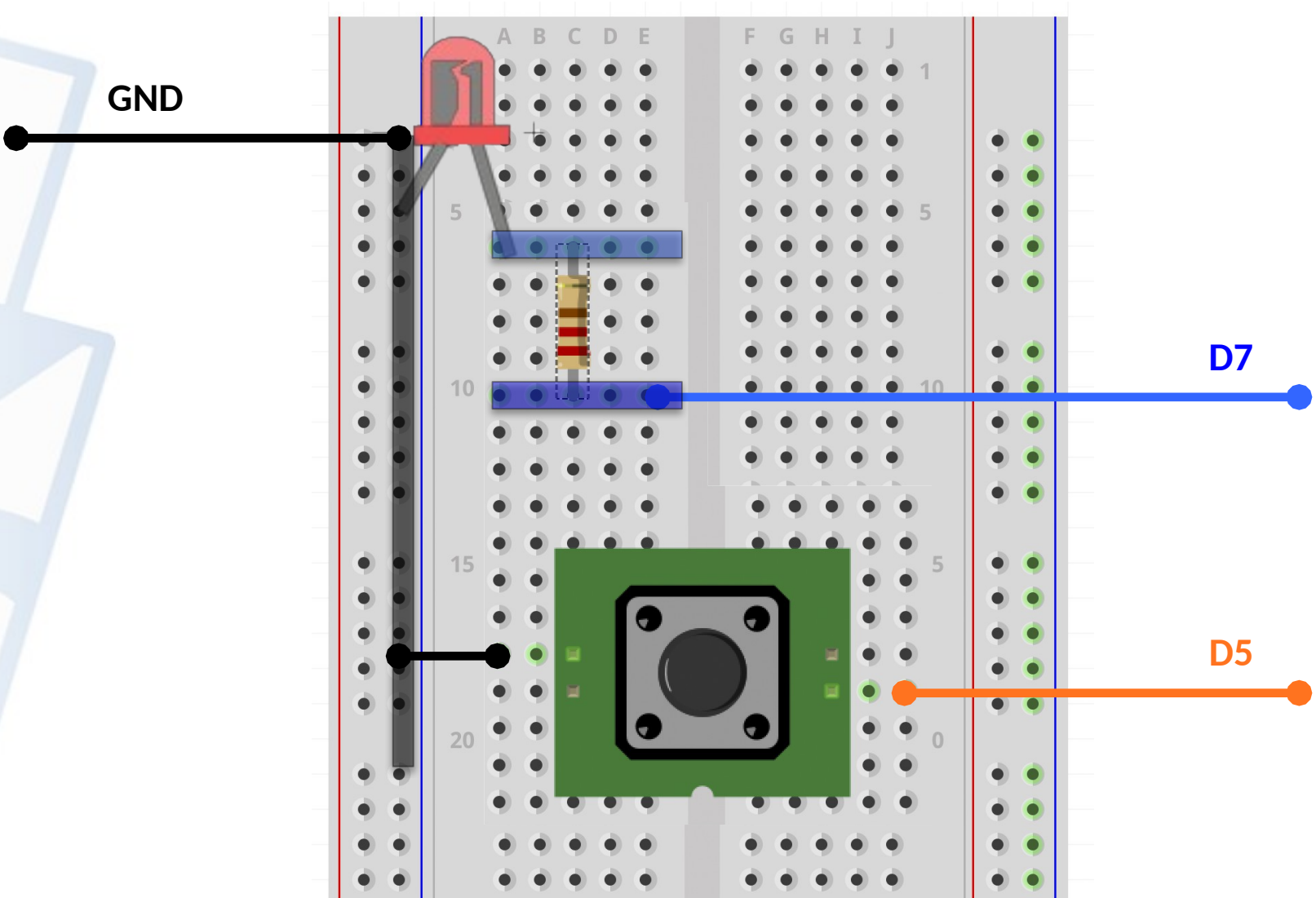
Close



Open



Entrada Digital (Pull_UP)

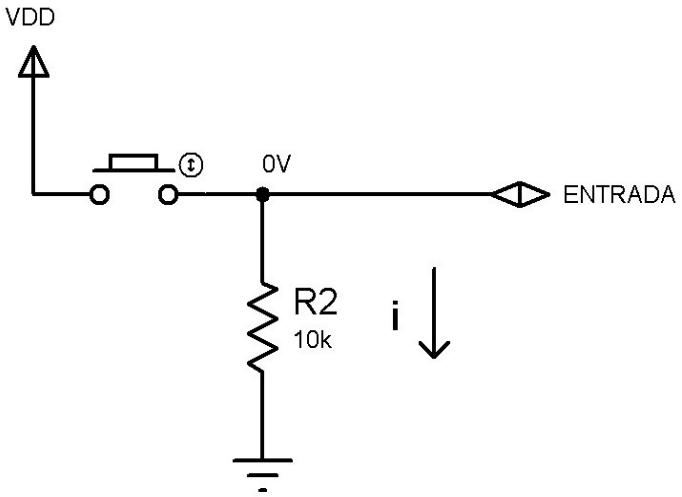


Codigo para Entrada Digital

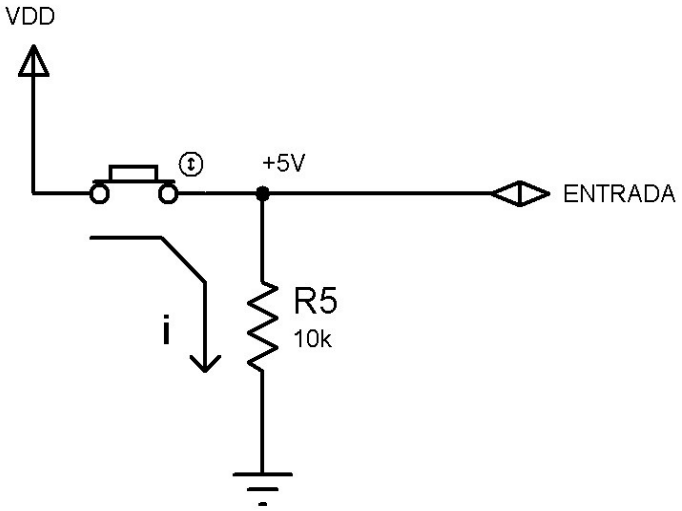
```
void setup() {
  // Poner el PIN 7 en Salida
  pinMode(7,OUTPUT);
  // Poner el PIN 5 en Entrada con Pull_Up
  // Pull_Up pone la entrada por defecto en HIGH
  pinMode(5, INPUT_PULLUP);

}

void loop() {
  // Leer la entrada del PIN 5
  bool lectura = digitalRead(5);
  // La salida del led depedende de la entrada
 digitalWrite(7,lectura);
}
```

Entradas digital – Pull-down

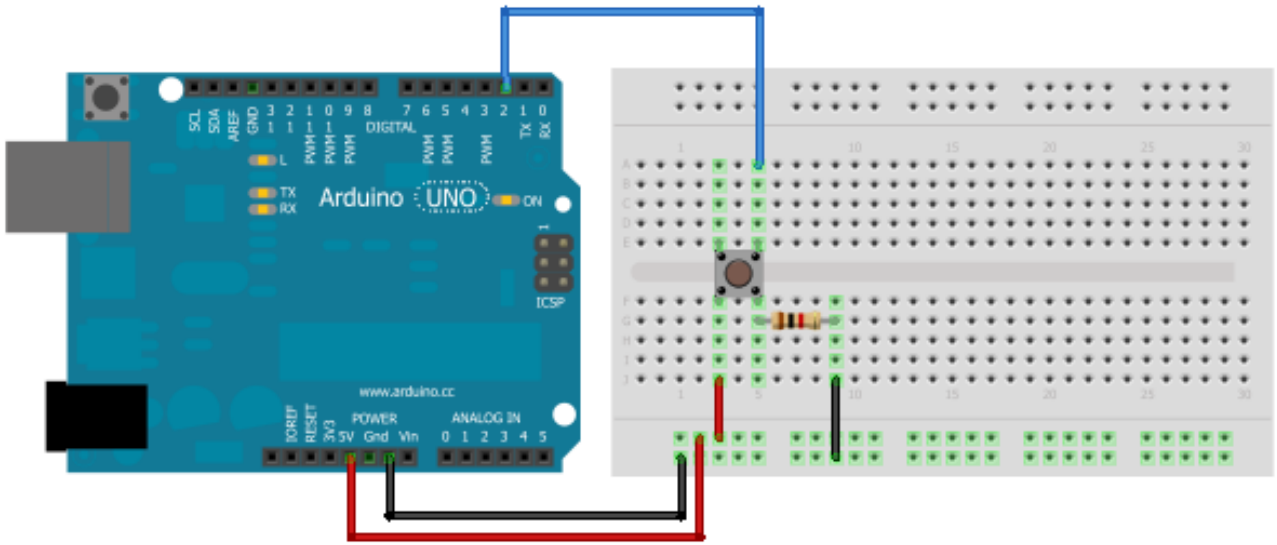
Close



Open



Entradas digital – Pull-down

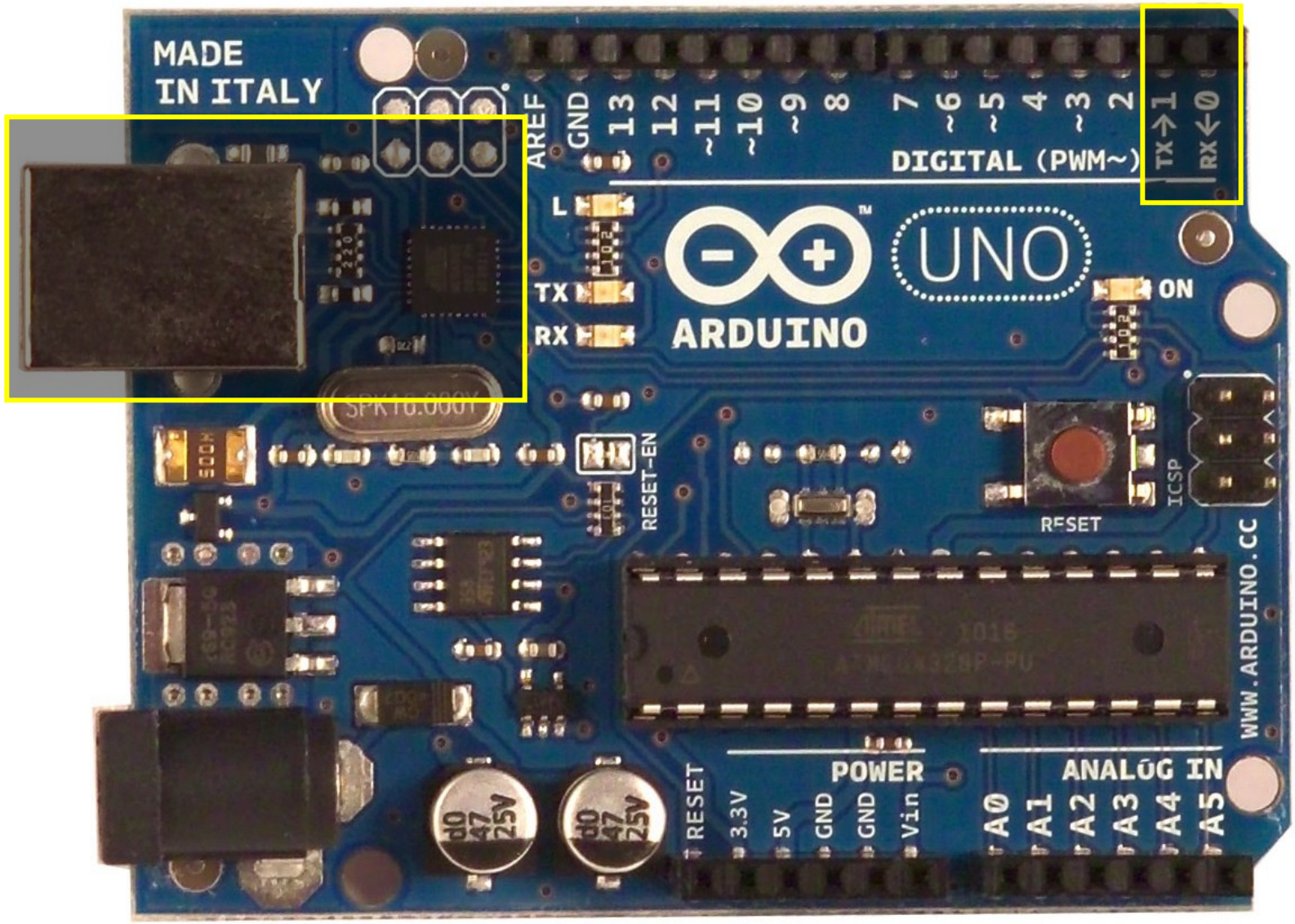


Una pull-down resistor, obliga a que una entrada digital cierre el circuito con el ground. Normalmente, estás resistencias son de 10 KOhms.

¿Cómo es posible tener conectado entonces el ground y el VCC al mismo tiempo?
La corriente siempre sigue el camino con menos resistencia, por lo tanto, cuando conectamos directamente el VCC, este camino no ofrece resistencia, y el circuito se cierra en este sentido .

También existe el PULL_UP que cierra el circuito con VCC en ausencia de GND.

Comunicación serial con UART

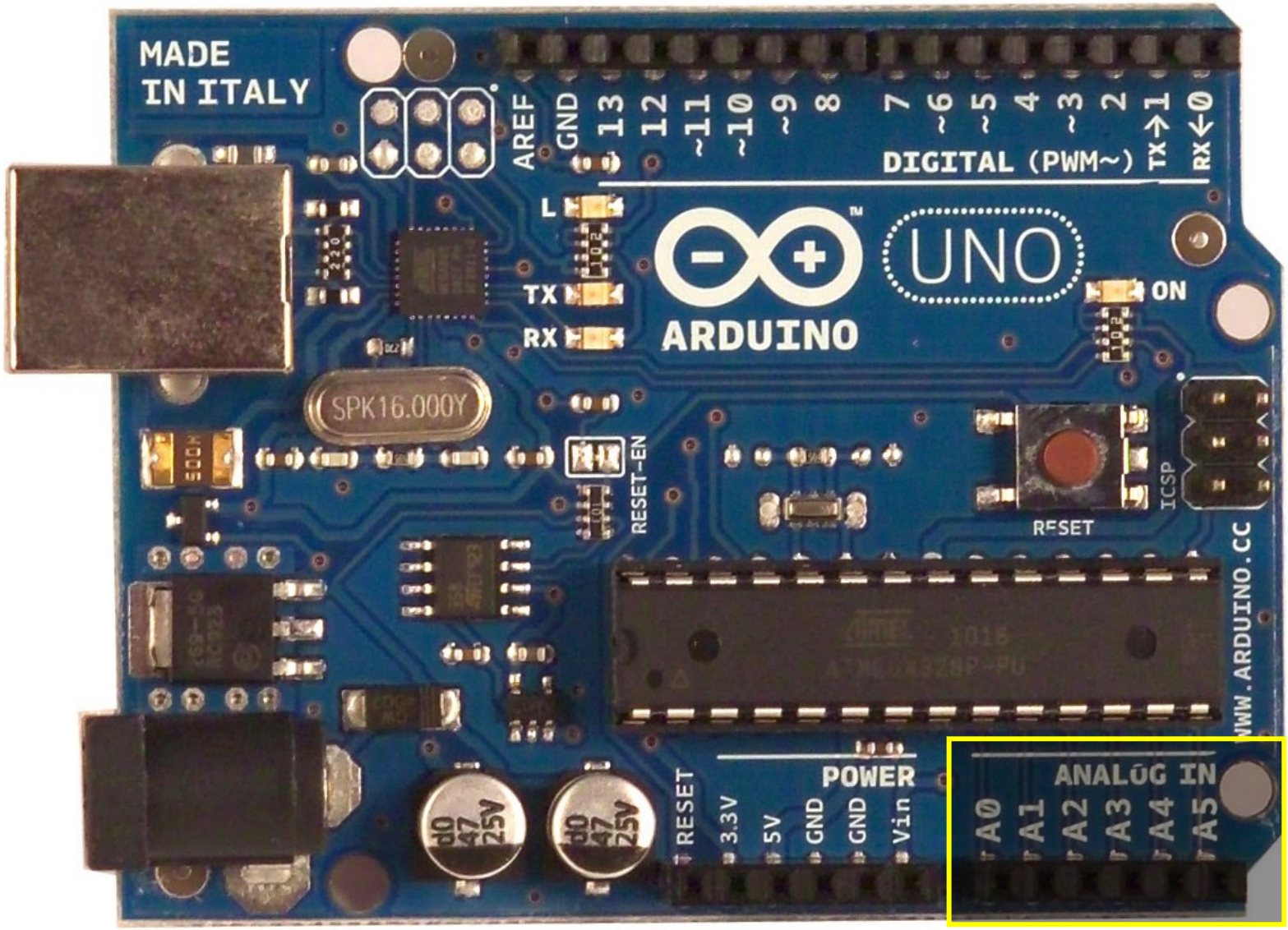


Codificar en Arduino

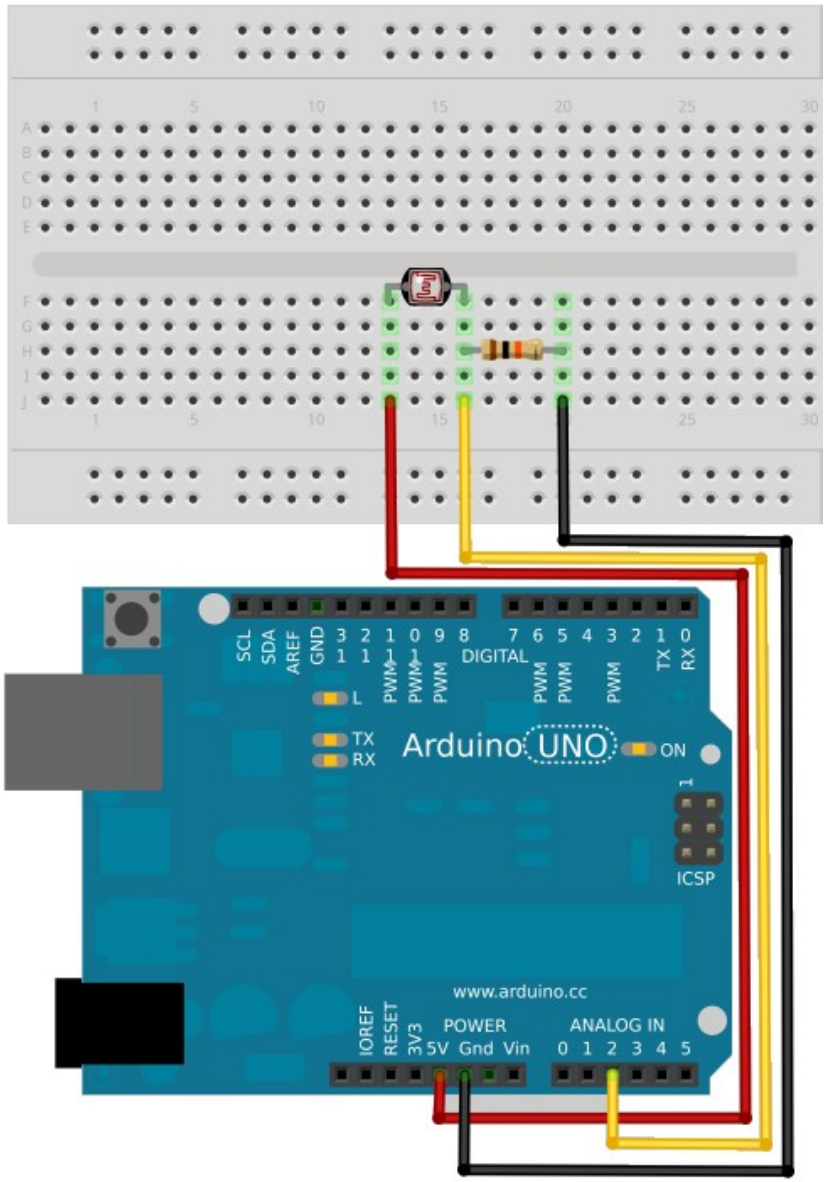
```
int pushButton = 2;

void setup() {
  // initialize serial communication at 9600 bps
  Serial.begin(9600);
  // make the pushbutton's pin an input:
  pinMode(pushButton, INPUT);
}

void loop() {
  // read the input pin:
  int buttonState = digitalRead(pushButton);
  // print out the state of the button:
  Serial.println(buttonState);
  delay(1);// for reads for stability
}
```

- UART son las siglas de Universal Asynchronus Receiver-Transmiter, y normalmente hacer referencia a un puerto de comunicación Serial.
- Usaremos el UART (o puerto Serial), para enviar y recibir información entre el arduino, y el computador (o cualquier dispositivo HOST).
- Serial.begin(9600) hace referencia a la velocidad de comunicación del puerto: 9600 bps -> 9600 bits por segundo.

Entradas analógicas



Ejemplo: FotoResitencia + UART

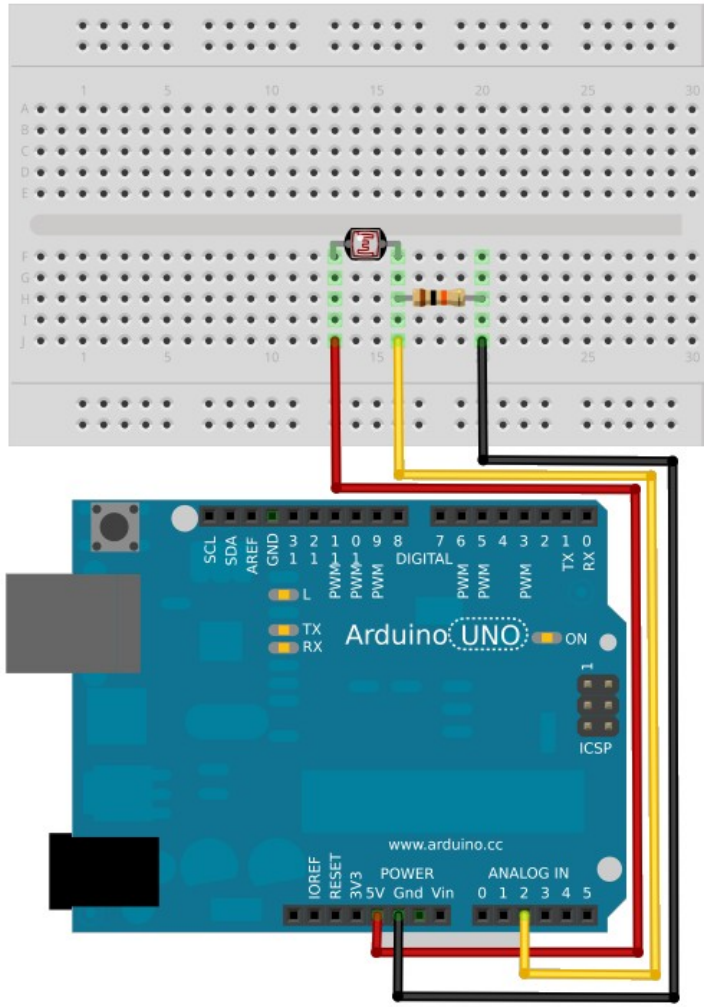


En el circuito donde tenemos a la resistencia y a la fotocelda en serie, esto con el fin de generar en el punto medio el voltaje de salida el cual leeremos en el pin A0 de nuestro arduino (pulldown)*.

Al momento de aumentar la intensidad de la luz, disminuye la resistencia generada, en nuestro circuito.
Al momento de disminuir la resistencia aumenta el voltaje, es decir si hay poca luz el voltaje será mínimo, pero si la luz aumenta el voltaje lo hace.

* La resistencia va entre 1K hasta los 10K, dependiendo de la intensidad de la luz y los resultados esperados.

Ejemplo: FotoResitencia + UART

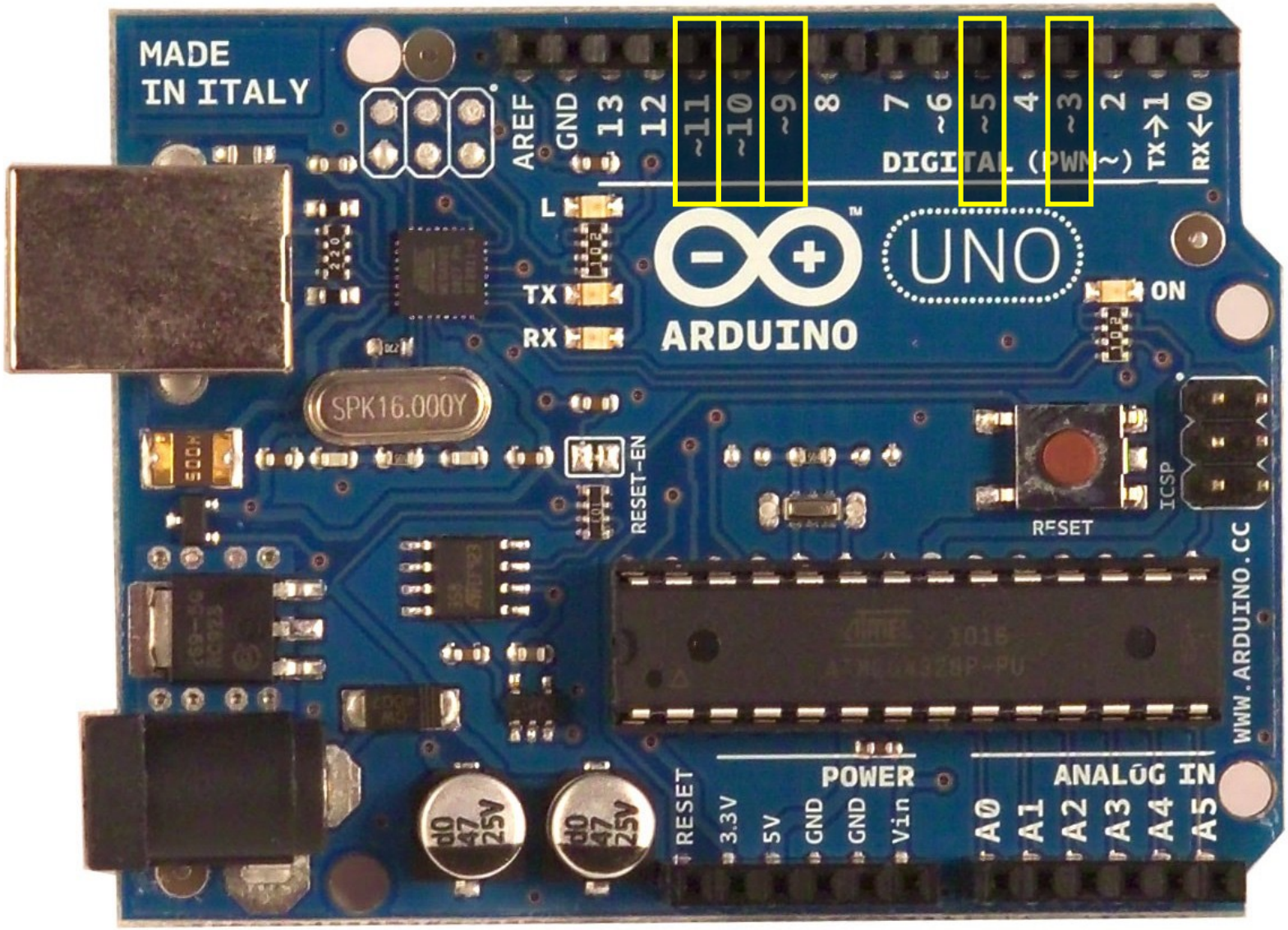



```
#define sensor A2

void setup() {
  Serial.begin(9600);
  pinMode(sensor, INPUT);
}

void loop() {
  int valueAnalog =
    analogRead(sensor);
  Serial.println(valueAnalog);
}
```

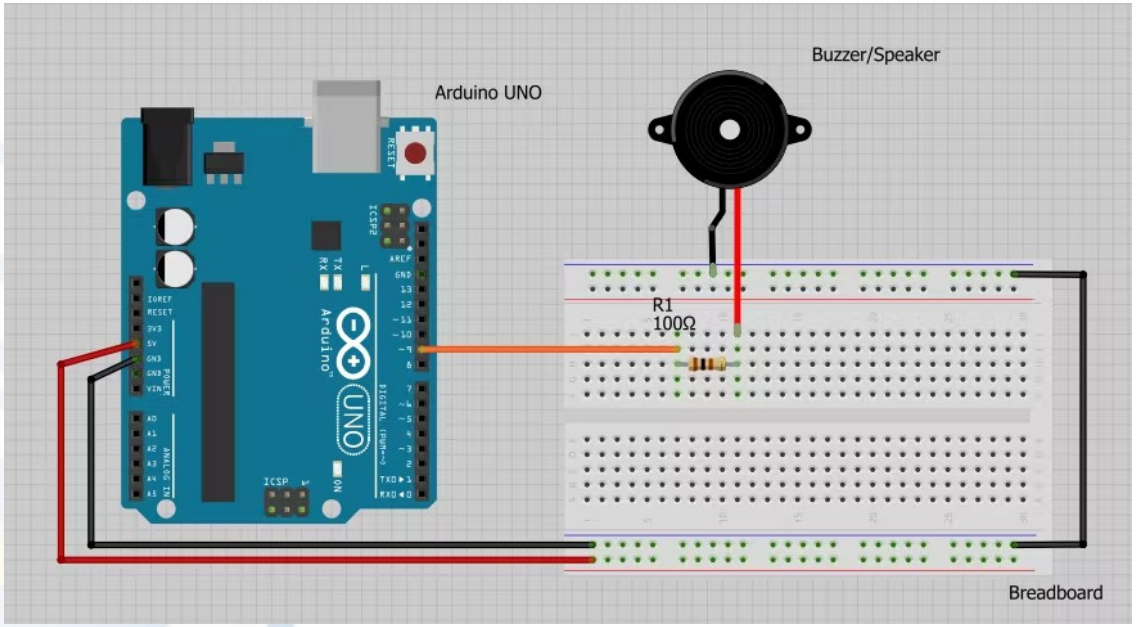
threshold

Desafío:
Crea un threshold/umbral para encender un led.

Modulacion por ancho de pulso PWM

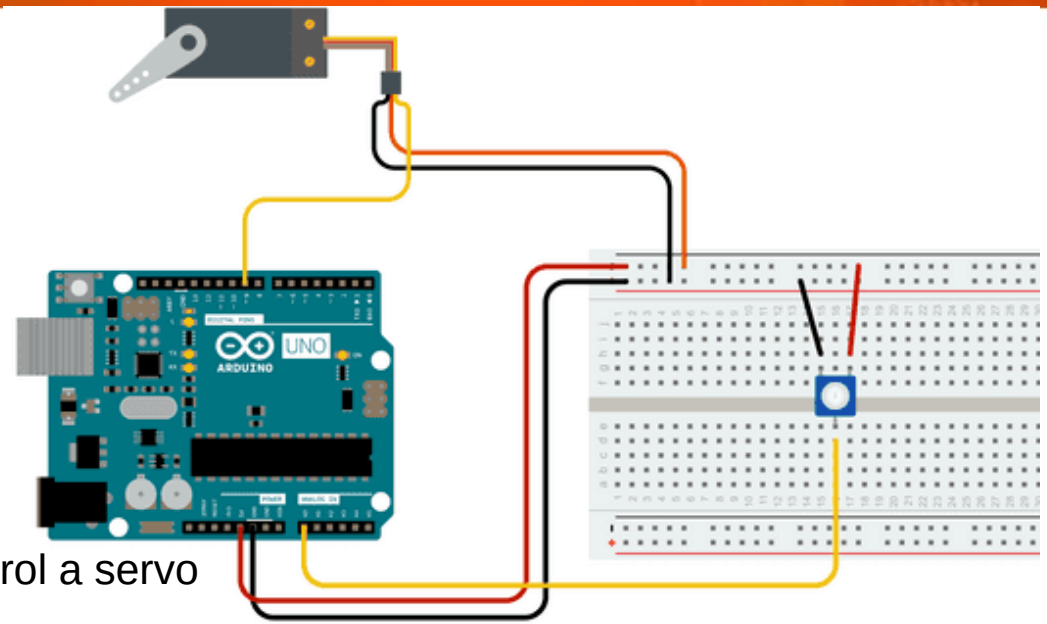


Ejemplo: Buzzer




```
const int buzzer = 9; //buzzer to arduino pin 9

void setup(){
  pinMode(buzzer, OUTPUT); //Set buzzer as an output pin
}

void loop(){
  tone(buzzer, 1000); // Send 1KHz sound signal...
  delay(1000);        // Wait for 1 sec
  noTone(buzzer);     // Stop sound...
  delay(1000);        // Waitfor 1sec

}
```

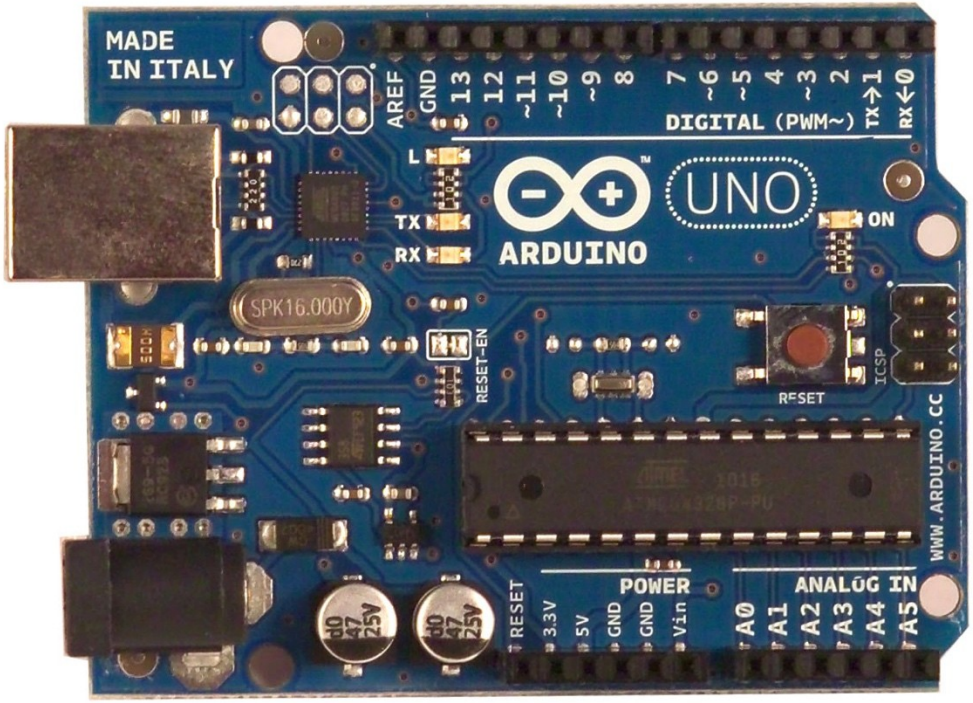
Ejemplo: Servo motor




```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo

int potpin = 0;  // analog pin used to connect the potentiometer
int val;    // variable to read the value from the analog pin

void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop() {
  val = analogRead(potpin);            // reads the value of the potentiometer (value between 0 and 1023)
  val = map(val, 0, 1023, 0, 180);     // scale it to use it with the servo (value between 0 and 180)
  myservo.write(val);                  // sets the servo position according to the scaled value
  delay(15);                           // waits for the servo to get there
```

EEPROM

- EEPROM es la memoria que nos permitirá persistencia




```
#include <EEPROM.h>
int v = EEPROM.read(posicion);
EEPROM.write(posicion, valor);
Rango: 0 - 255
```