

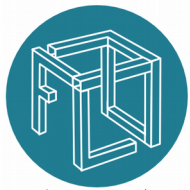
FABLAB

SANTANDER

Taller Arduino modulo 1

Introducción

Fablabsantander.org/files/talleres/arduino/tallerArduino.pdf

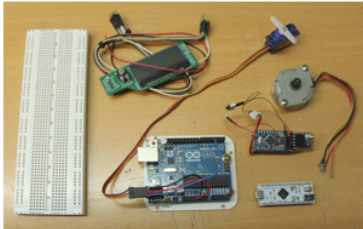


FABLAB
SANTANDER

Los talleres del FabLab

Módulo 1

Los tipos de Arduinos, los pins, el compilador, el ejemplo blink, controlar una LED, montaje en un breadboard, descripción de los pins PWM y ADC

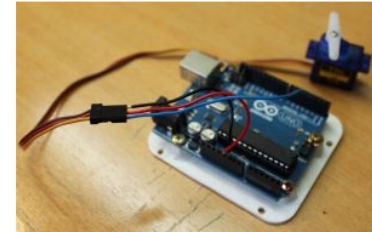


Módulo 2

Salida/entrada digital: ejemplo "State change detection" explicado en detalles: contar cuantas veces actuamos un interruptor y encender una LED cada 4 cuentas. Como cablear un interruptor. Pin de entrada analógica (ADC) : leer la señal de un sensor. Ejemplo de sensor de luz hecho con una LED (utilizada al como sensor y no emisor!). -> comunicación óptica entre 2 arduinos !

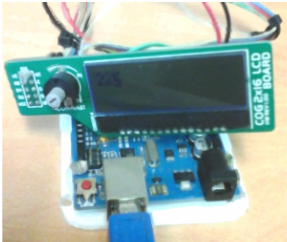
Módulo 3

Pin ADC : ejemplo montaje potenciómetro, pin PWM: modular una señal de salida, montaje LED de brillo variable y de un motor servo.



Módulo 4

Observación la señal PWM con el osciloscopio, programar una pantalla LCD, instalar un sensor de temperatura (termo-resistor) y enseñar esta temperatura en la LCD.

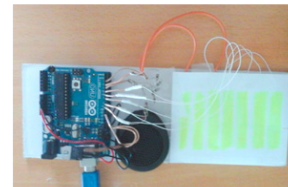


Módulo 5

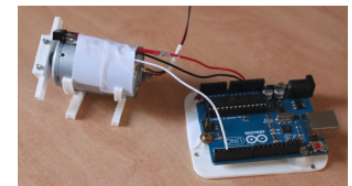
Un poco de potencia: controlar un motor continuo, usar un alta-voz potente.

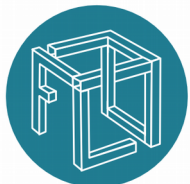
Módulo 6

Un mini piano.



Módulo 7: usar las interrupciones: ejemplo de un codificador óptico para un motor continuo o de un sistema de "Persistence Of Vision" (POV)





FABLAB
SANTANDER

Las placas Arduino



ARDUINO UNO



ARDUINO 101



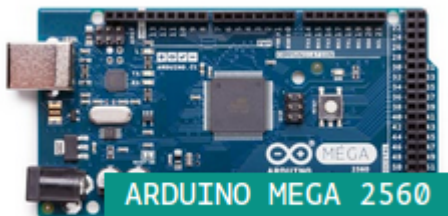
ARDUINO PRO



ARDUINO MICRO



ARDUINO PRO MINI



ARDUINO MEGA 2560



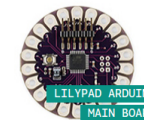
ARDUINO ZERO



GEMMA ARDUINO GEMMA



LILYPAD ARDUINO SIMPLE



LILYPAD ARDUINO MAIN BOARD



ARDUINO MKR1000



ARDUINO WIFI SHIELD 101



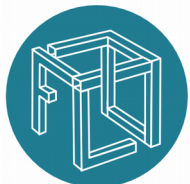
ARDUINO YUN Shield



LILYPAD ARDUINO USB



LILYPAD ARDUINO SIMPLE SNAP



FABLAB

SANTANDER

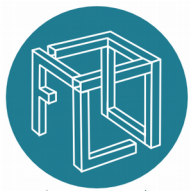
El Arduino IDE

Integrated Development Environment

```
BareMinimum | Arduino 1.6.11
Fichier Édition Croquis Outils Aide
BareMinimum
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
Arduino/Genuino Uno sur /dev/ttyACM0
```

Se ejecuta una vez al arranque

Se ejecuta en bucle



FABLAB

SANTANDER

Conectar la placa

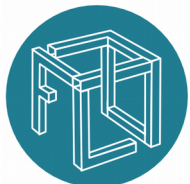
Seleccionar la placa:

Herramientas → placa

Seleccionar el puerto:

Herramientas → puerto

(En windows ver con win+pause si la placa esta vista como periférico en un puerto serie COMxx)



FABLAB
SANTANDER

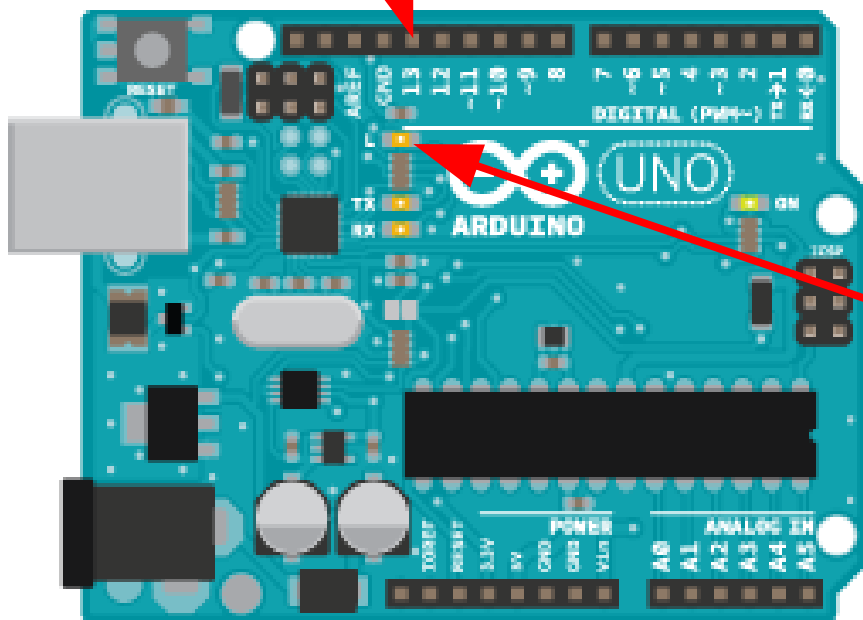
El ejemplo Blink

La función "setup" se ejecuta al inicio (reset y arranque)

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000);           // wait for a second
}
```

PIN 13



La función "loop" se ejecuta de forma continua

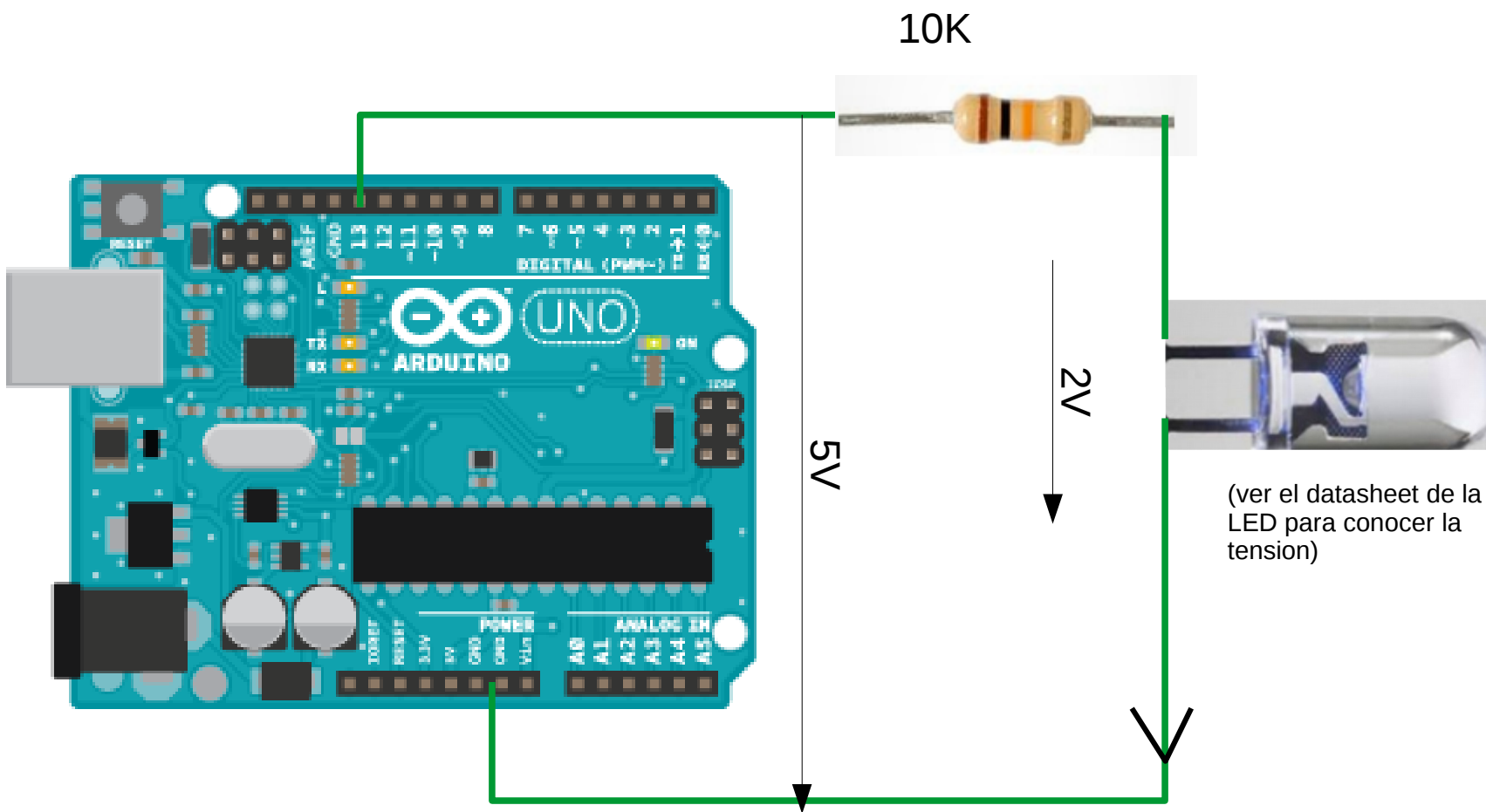
LED conectada al PIN 13



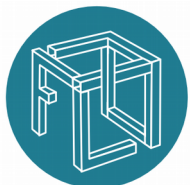
FABLAB

SANTANDER

Un poco mas allá



$$I=U/R=(5-2)/10000=3\text{mA}$$



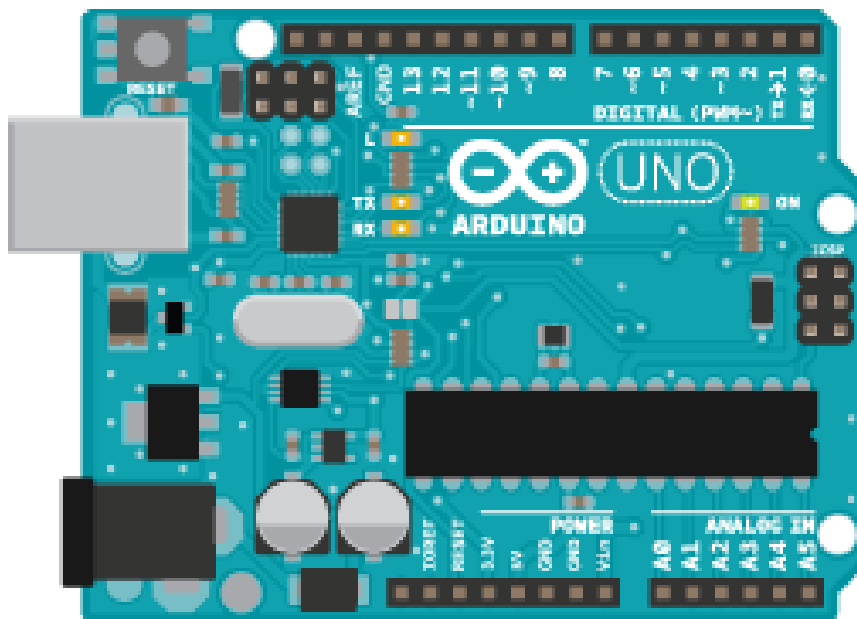
FABLAB

SANTANDER

Las posibilidades

Entradas o salidas digital (0 o 1)

Salidas analógicas PWM (~)



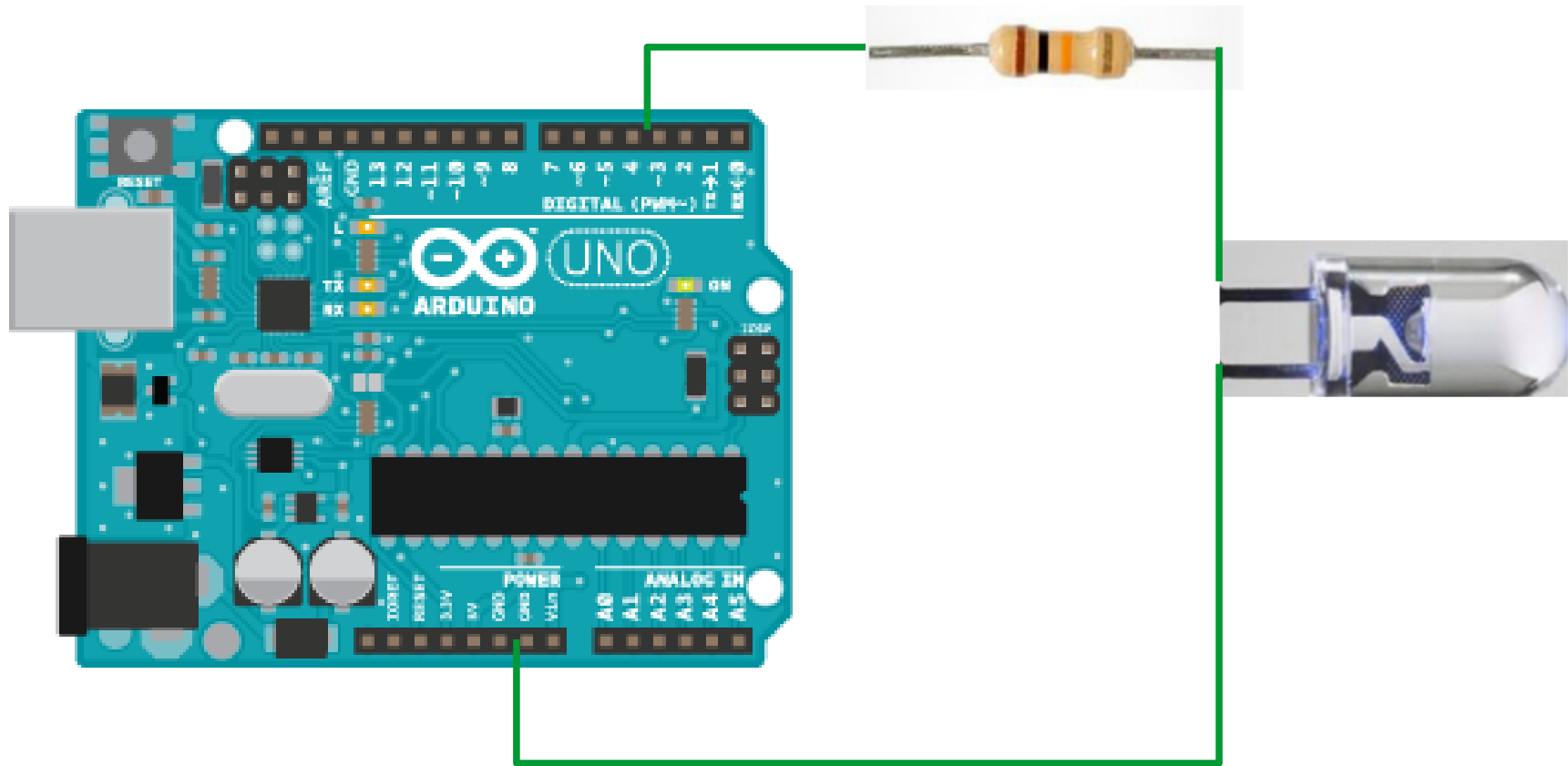
Entradas analógicas: (valor de 0 a 1024)

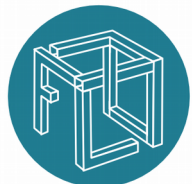


FABLAB
SANTANDER

Usamos la salida analógica PWM

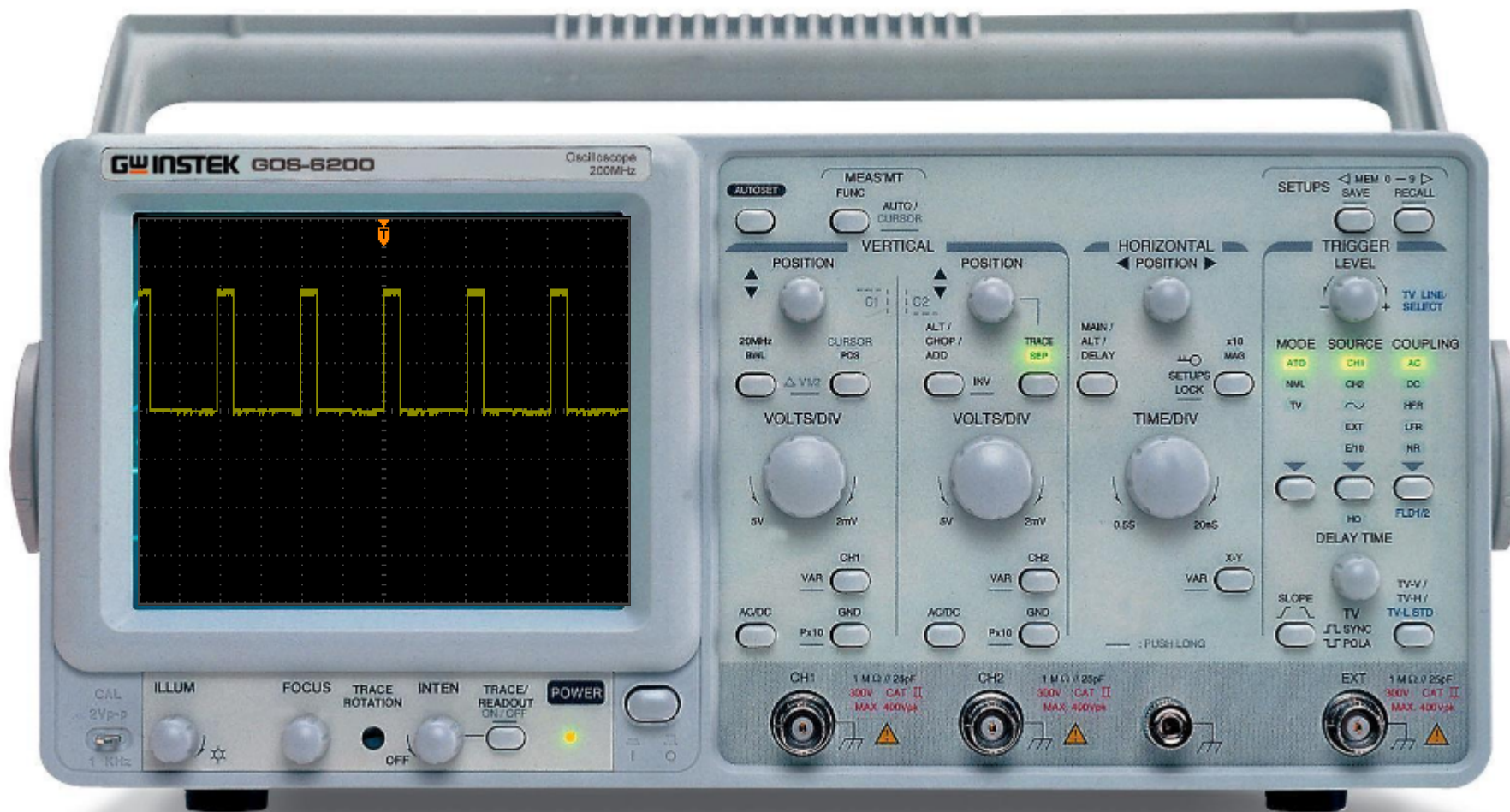
Ejemplo: "fading"

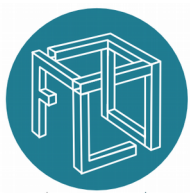




FABLAB
SANTANDER

Ver la salida PWM con el osciloscopio





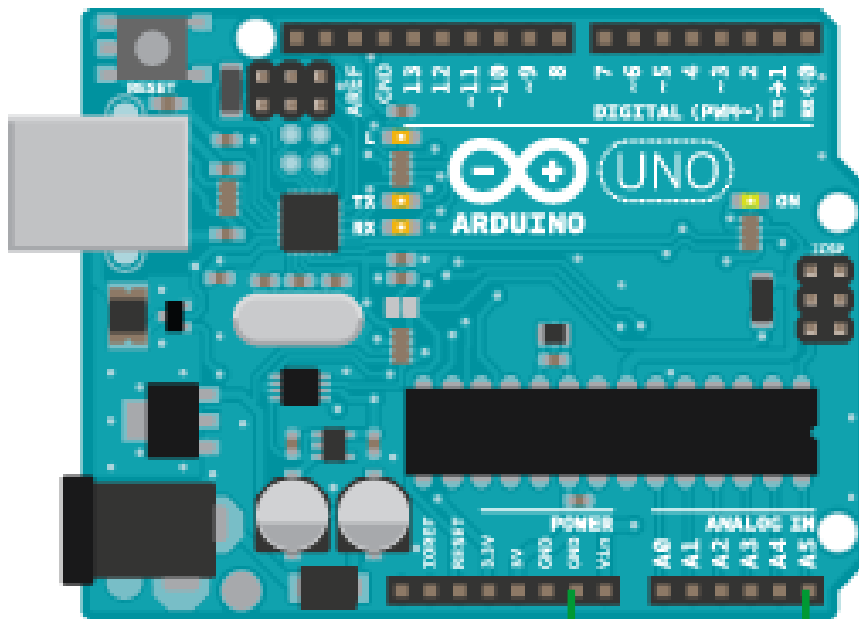
FABLAB

SANTANDER

Usamos la entrada analógica

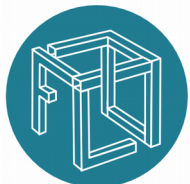
Ejemplo: "AnalogReadSerial"

```
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize serial communication at 9600 bits per second:  
  Serial.begin(9600);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  // read the input on analog pin 0:  
  int sensorValue = analogRead(A0);  
  // print out the value you read:  
  Serial.println(sensorValue);  
  delay(1);      // delay in between reads for stability  
}
```



Un LED es sensible a la luz

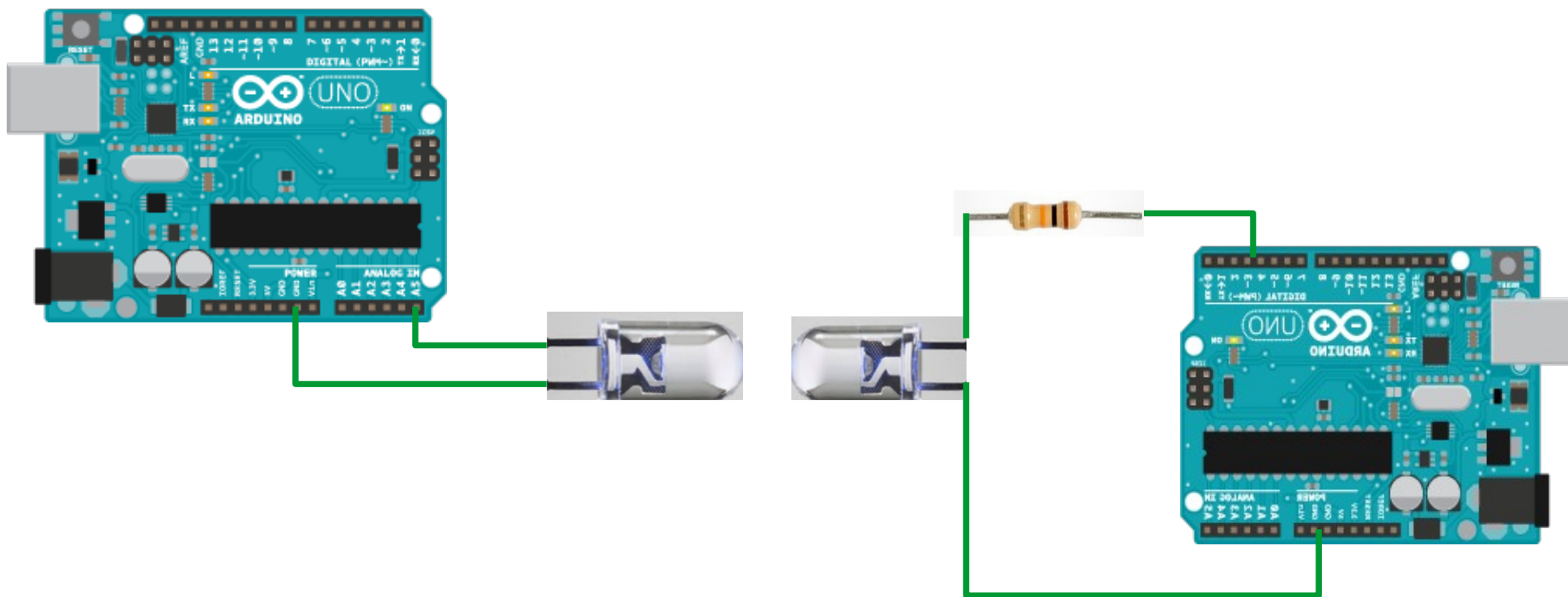


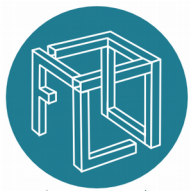


FABLAB

SANTANDER

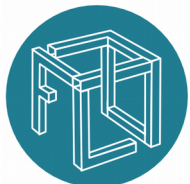
Hacemos comunicación óptica !





FABLAB
SANTANDER

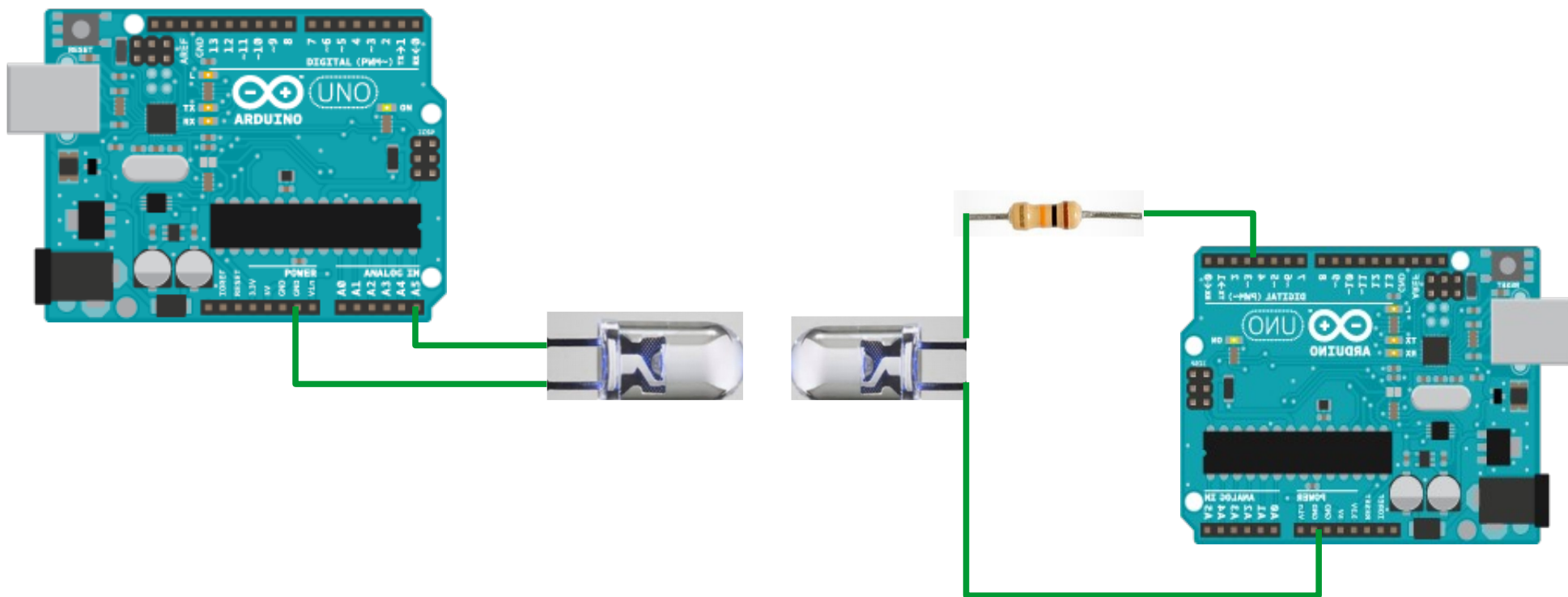
Taller Arduino modulo 2

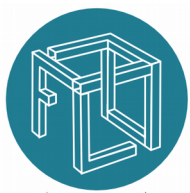


FABLAB

SANTANDER

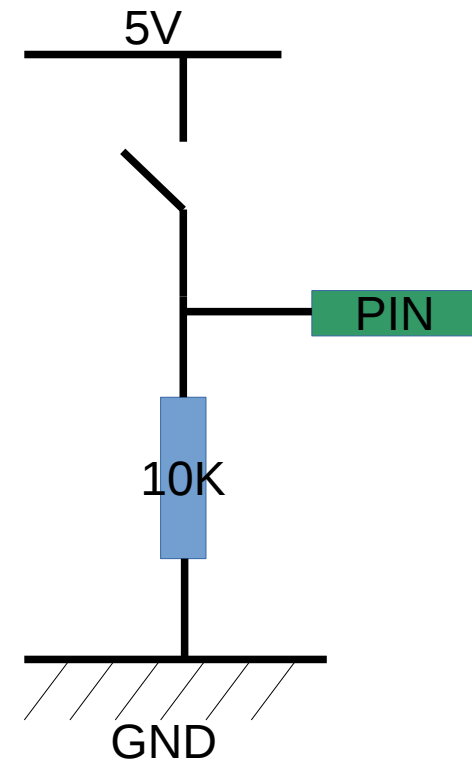
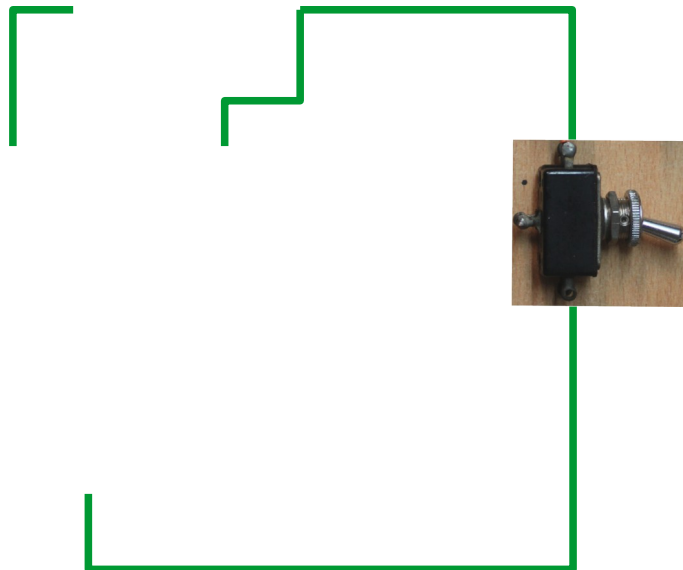
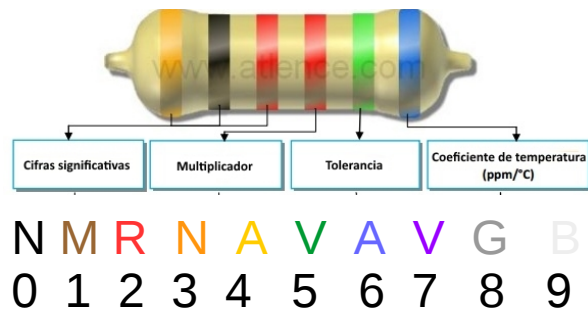
Hacemos comunicación óptica !

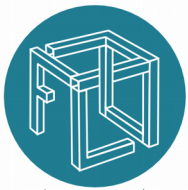




FABLAB
SANTANDER

Montaje de un interruptor

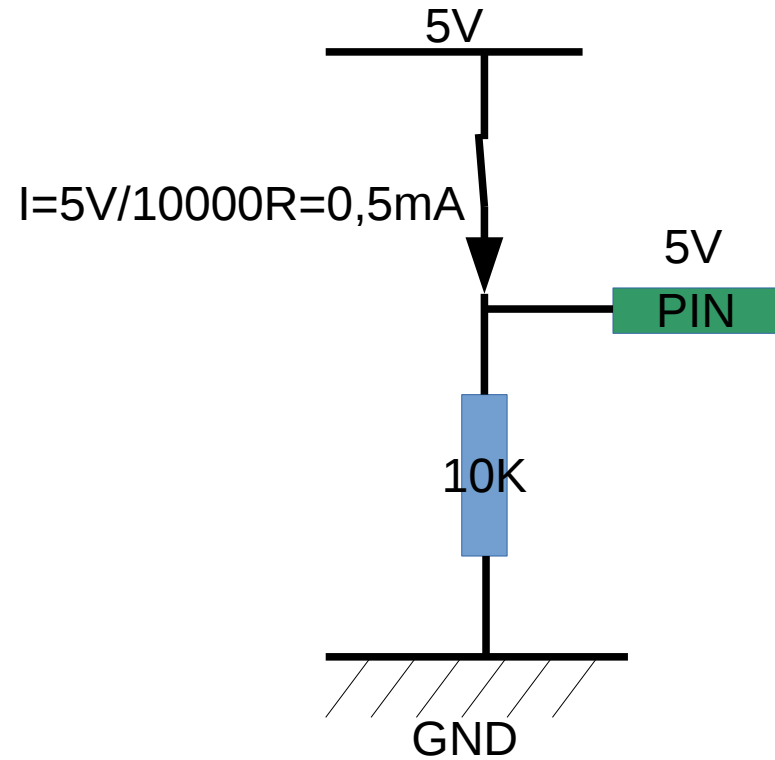
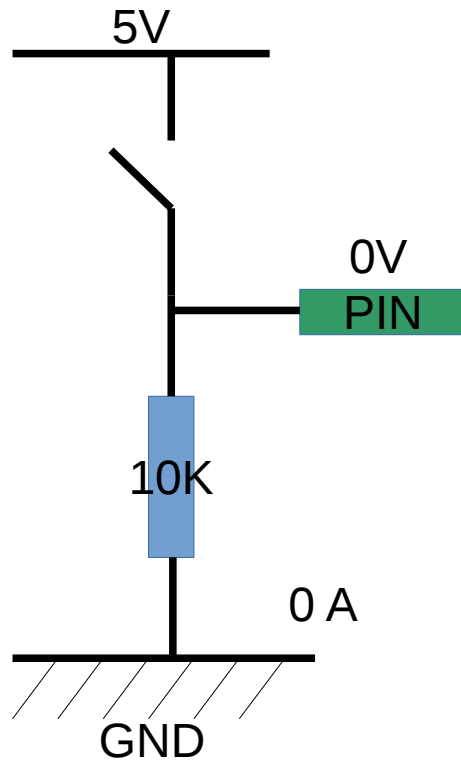


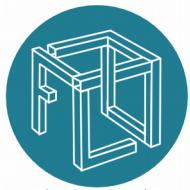


FABLAB

SANTANDER

Las corrientes





FABLAB
SANTANDER

Ejemplo State Change Detection

```
// this constant won't change:
const int buttonPin = 2; // the pin that the pushbutton is attached to
const int ledPin = 13; // the pin that the LED is attached to

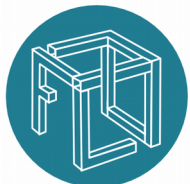
// Variables will change:
int buttonPushCounter = 0; // counter for the number of button presses
int buttonState = 0; // current state of the button
int lastButtonState = 0; // previous state of the button

void setup() {
  // initialize the button pin as a input:
  pinMode(buttonPin, INPUT);
  // initialize the LED as an output:
  pinMode(ledPin, OUTPUT);
  // initialize serial communication:
  Serial.begin(9600);
}
```

```
void loop() {
  // read the pushbutton input pin:
  buttonState = digitalRead(buttonPin);

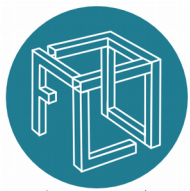
  // compare the buttonState to its previous state
  if (buttonState != lastButtonState) {
    // if the state has changed, increment the counter
    if (buttonState == HIGH) {
      // if the current state is HIGH then the button
      // went from off to on:
      buttonPushCounter++;
      Serial.println("on");
      Serial.print("number of button pushes: ");
      Serial.println(buttonPushCounter);
    } else {
      // if the current state is LOW then the button
      // went from on to off:
      Serial.println("off");
    }
    // Delay a little bit to avoid bouncing
    delay(50);
  }
  // save the current state as the last state,
  //for next time through the loop
  lastButtonState = buttonState;

  // turns on the LED every four button pushes by
  // checking the modulo of the button push counter.
  // the modulo function gives you the remainder of
  // the division of two numbers:
  if (buttonPushCounter % 4 == 0) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```



FABLAB
SANTANDER

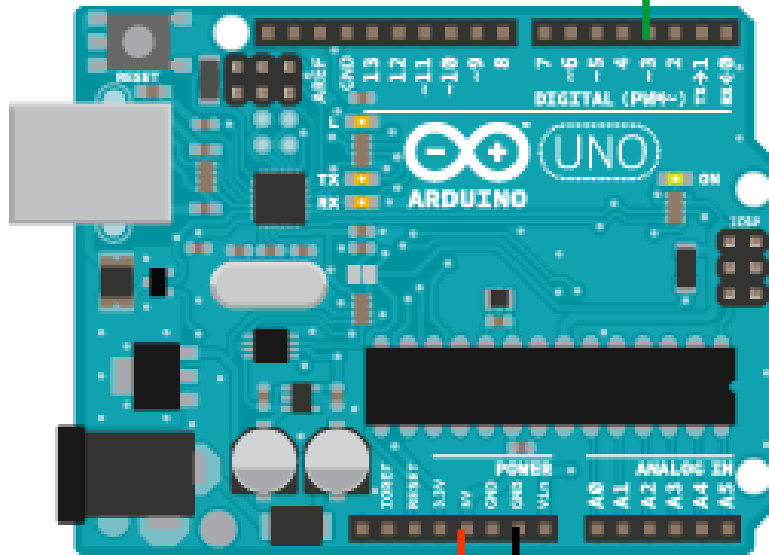
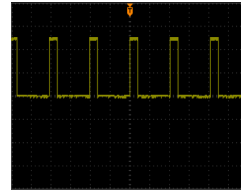
Taller Arduino modulo 3



FABLAB
SANTANDER

Mandar un servo motor

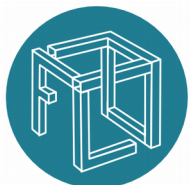
El ángulo del motor es codificado con señal PWM



Ejemplo:
Servo → sweep

Probar también
directamente con
analogWrite

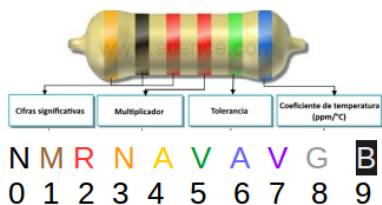
```
#include <Servo.h>  
Servo myservo; // create servo object to control a servo  
myservo.attach(9);  
myservo.write(pos);
```



FABLAB
SANTANDER

ADC. Potenciómetro

El ajuste del potenciómetro cambia el brillo de la LED



$$I = V_{cc} / (R_1 + R)$$

$$V_0 = R I$$

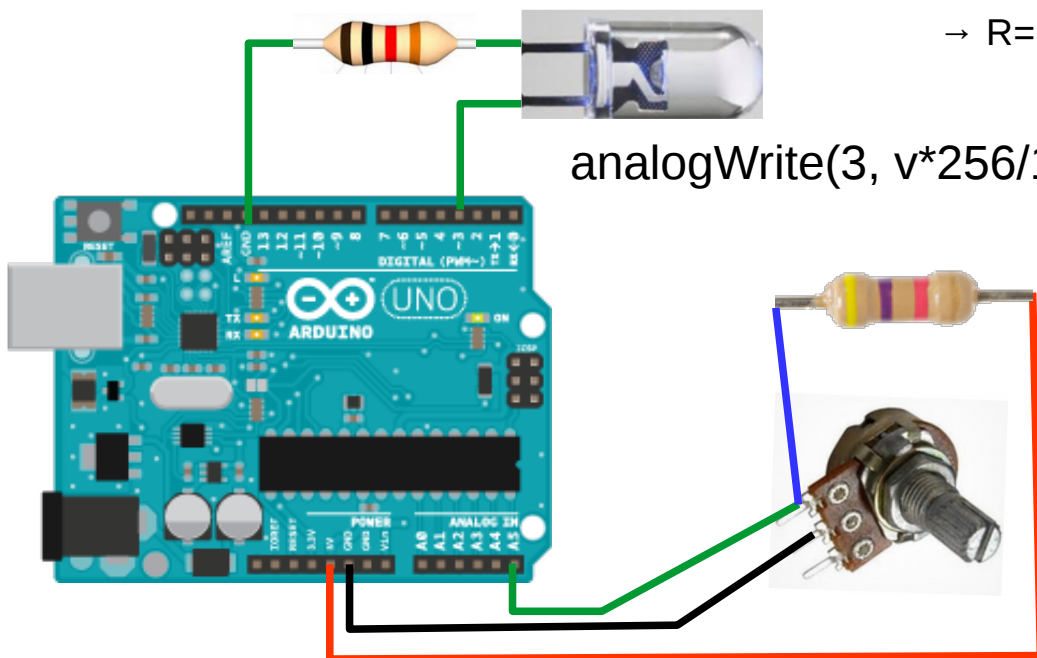
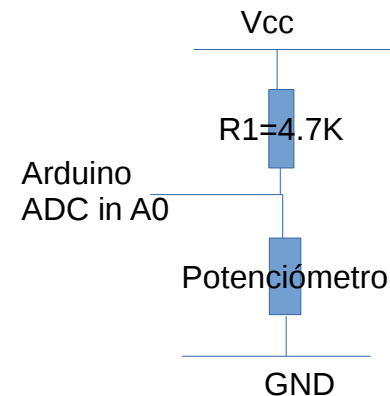
$$\rightarrow V_0 / R = V_{cc} / (R_1 + R)$$

$$\rightarrow R / V_0 = (R_1 + R) / V_{cc}$$

$$\rightarrow R(1/V_0 - 1/V_{cc}) = R_1 / V_{cc}$$

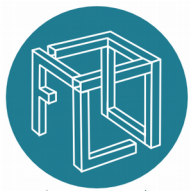
$$\rightarrow R = R_1 / V_{cc} (1/V_0 - 1/V_{cc})$$

$$\rightarrow R = R_1 (V_{cc} / V_0 - 1)$$



```
analogWrite(3, v*256/1024);
```

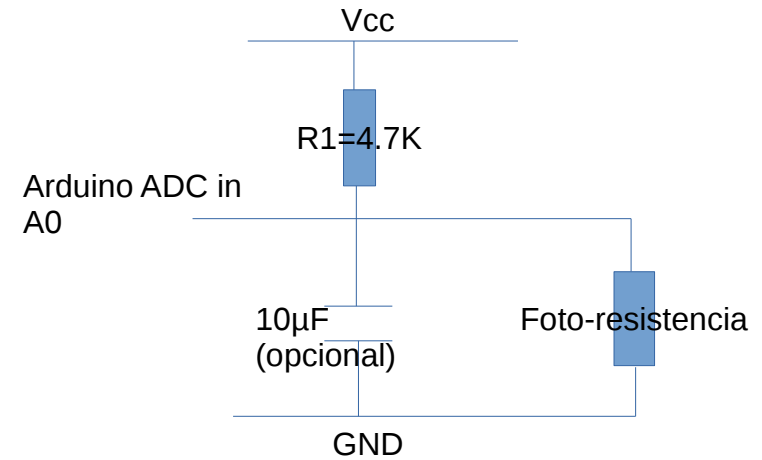
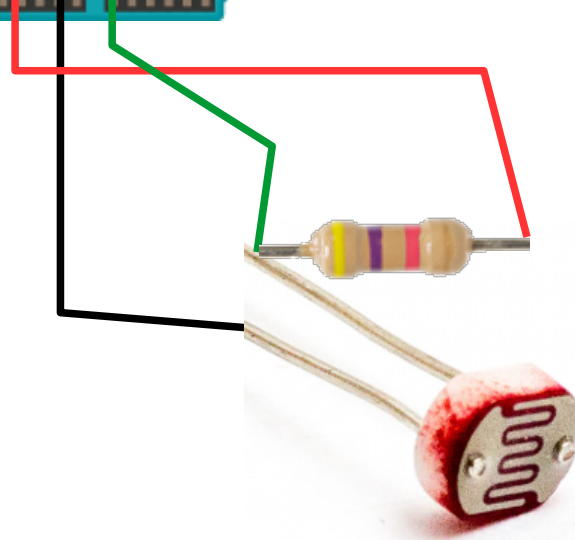
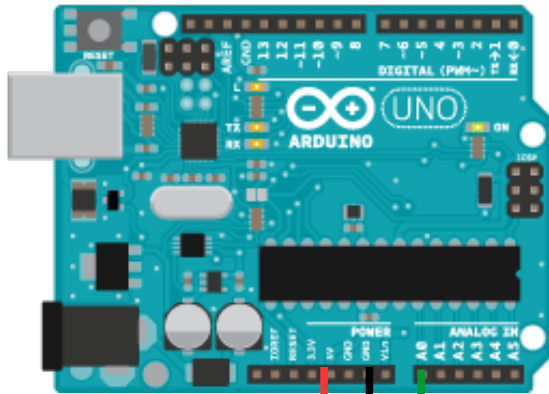
```
v=analogRead(A5);
```



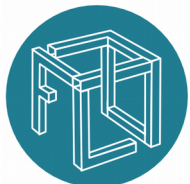
FABLAB

SANTANDER

Foto-resistencia



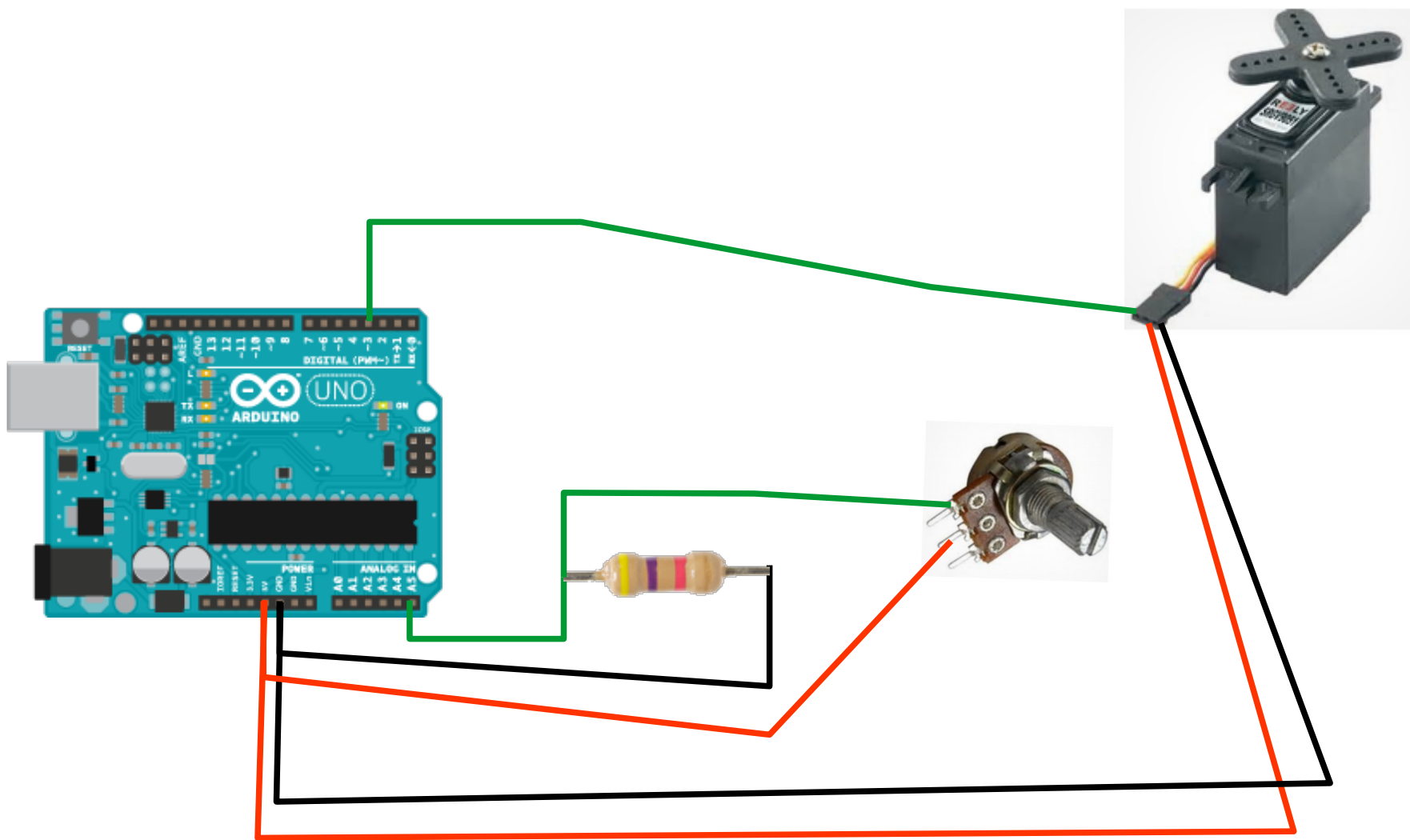
Resistencia (con luz) : ~10k-20k Ohm
Resistencia (oscuridad): ~1M Ohm



FABLAB

SANTANDER

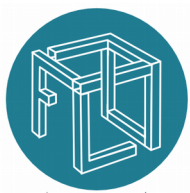
Potenciometro+servo





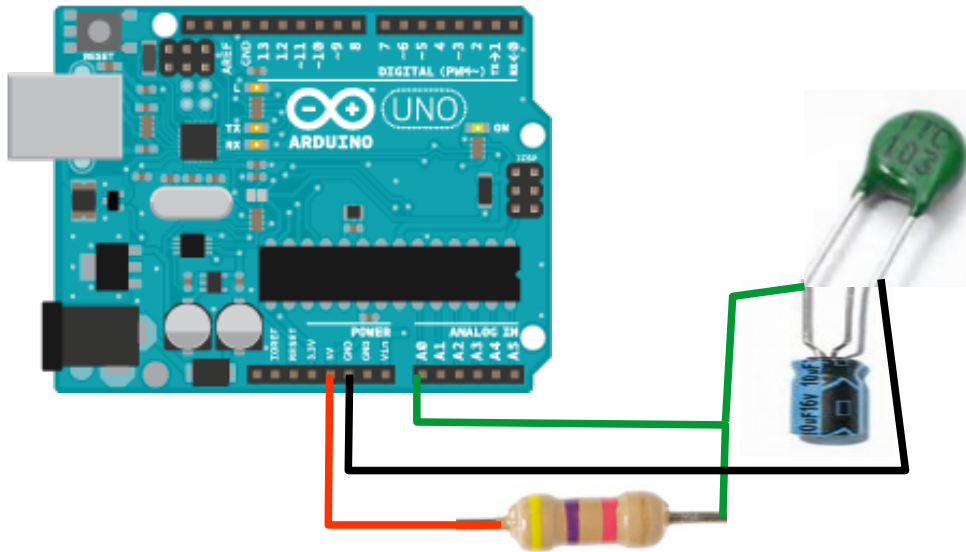
FABLAB
SANTANDER

Taller Arduino modulo 4

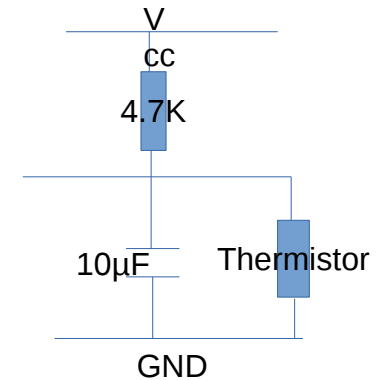


FABLAB
SANTANDER

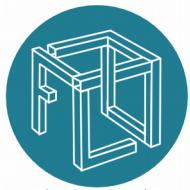
ADC: thermistor (Impresoras 3D)



Arduino
ADC in A0



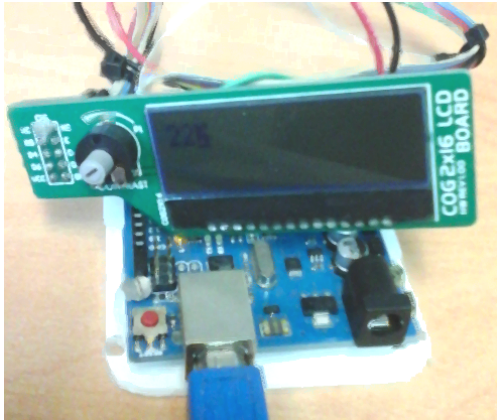
Ver el valor con el puerto serie.
Convertir el valor en temperatura con la tabla "thermistor.h"



FABLAB

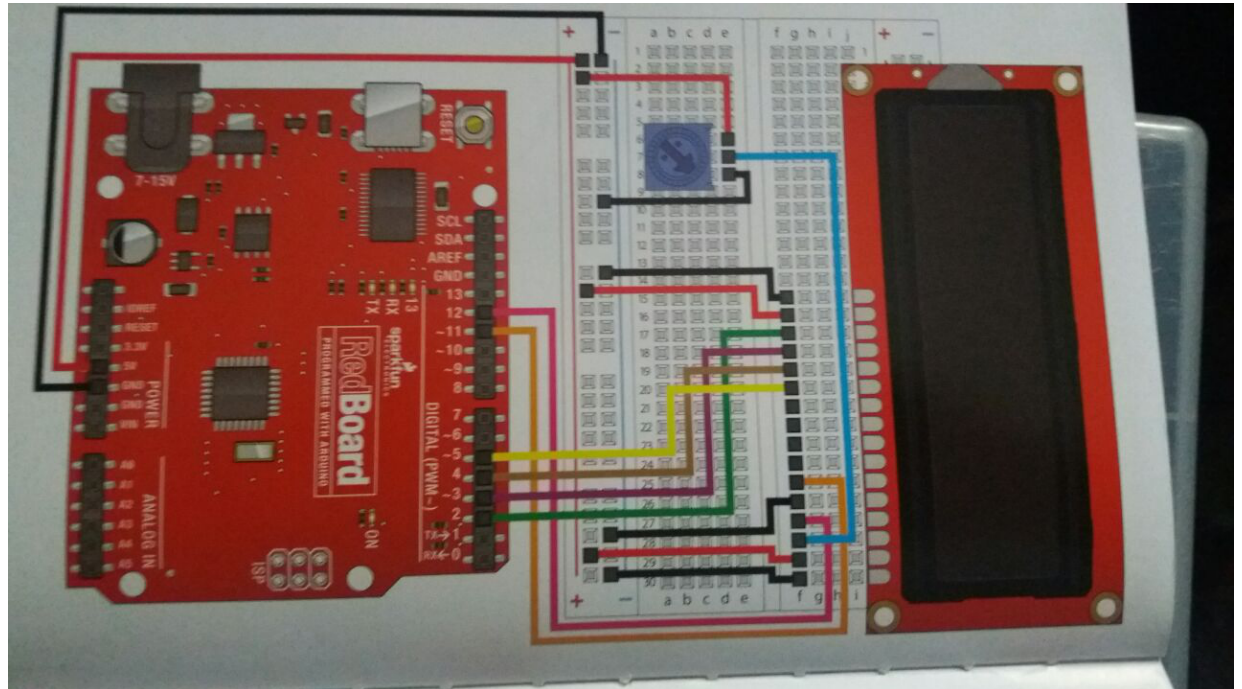
SANTANDER

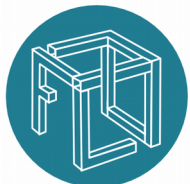
Pantalla LCD



LCD.zip

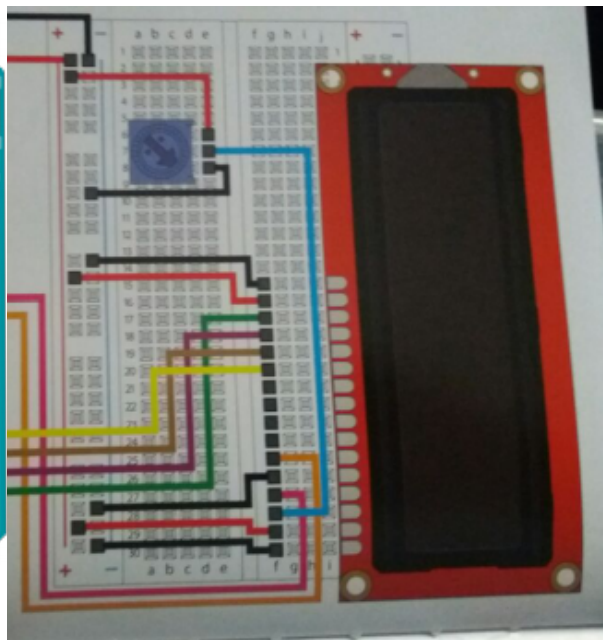
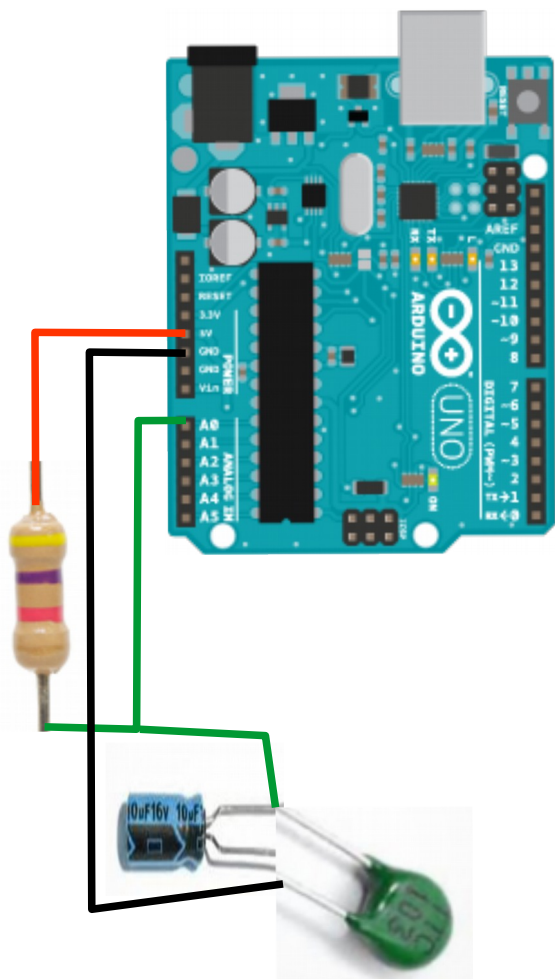
```
// include the library code:
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup(){
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // initialize the serial communications:
  Serial.begin(9600);
}
void loop()
{
  // when characters arrive over the serial port...
  if (Serial.available()) {
    // wait a bit for the entire message to arrive
    delay(100);
    // clear the screen
    lcd.clear();
    // read all the available characters
    while (Serial.available() > 0) {
      //receive unix time
      // display each character to the LCD
      lcd.write(Serial.read());
    }
  }
}
```

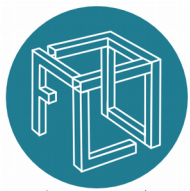




FABLAB
SANTANDER

Temperatura y LCD

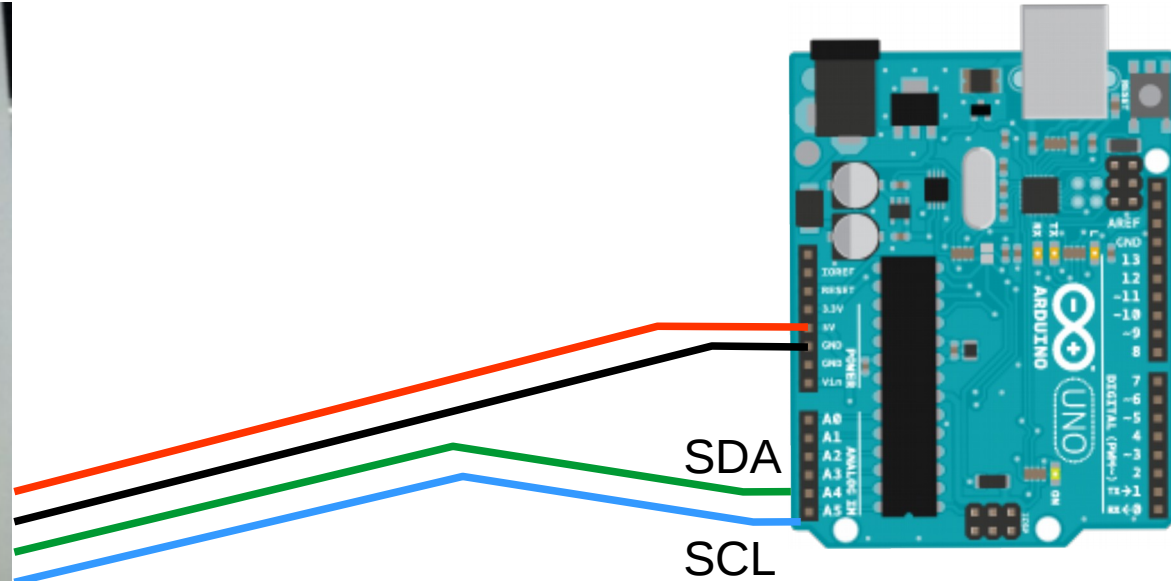




FABLAB

SANTANDER

Pantalla LCD I2C



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

Arduino-LiquidCrystal-I2C-library-master.zip

I2C_scan.zip

LCD_I2C.zip

```

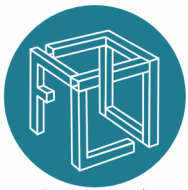
#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>

#define I2C_ADDR 0x27

LiquidCrystal_I2C lcd(I2C_ADDR,2, 1, 0, 4, 5, 6, 7);
void setup()
{
  lcd.begin (16,2); // Inicializar el display con 16 caracteres 2 lineas
  lcd.setBacklightPin(3,POSITIVE);
  lcd.setBacklight(HIGH);

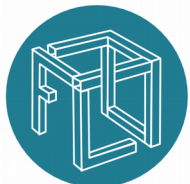
  lcd.home ();      // go home
  lcd.print("FabLab");
  lcd.setCursor ( 0, 1 ); // go to the 2nd line
  lcd.print("Santander");
}
void loop()
{}

```



FABLAB
SANTANDER

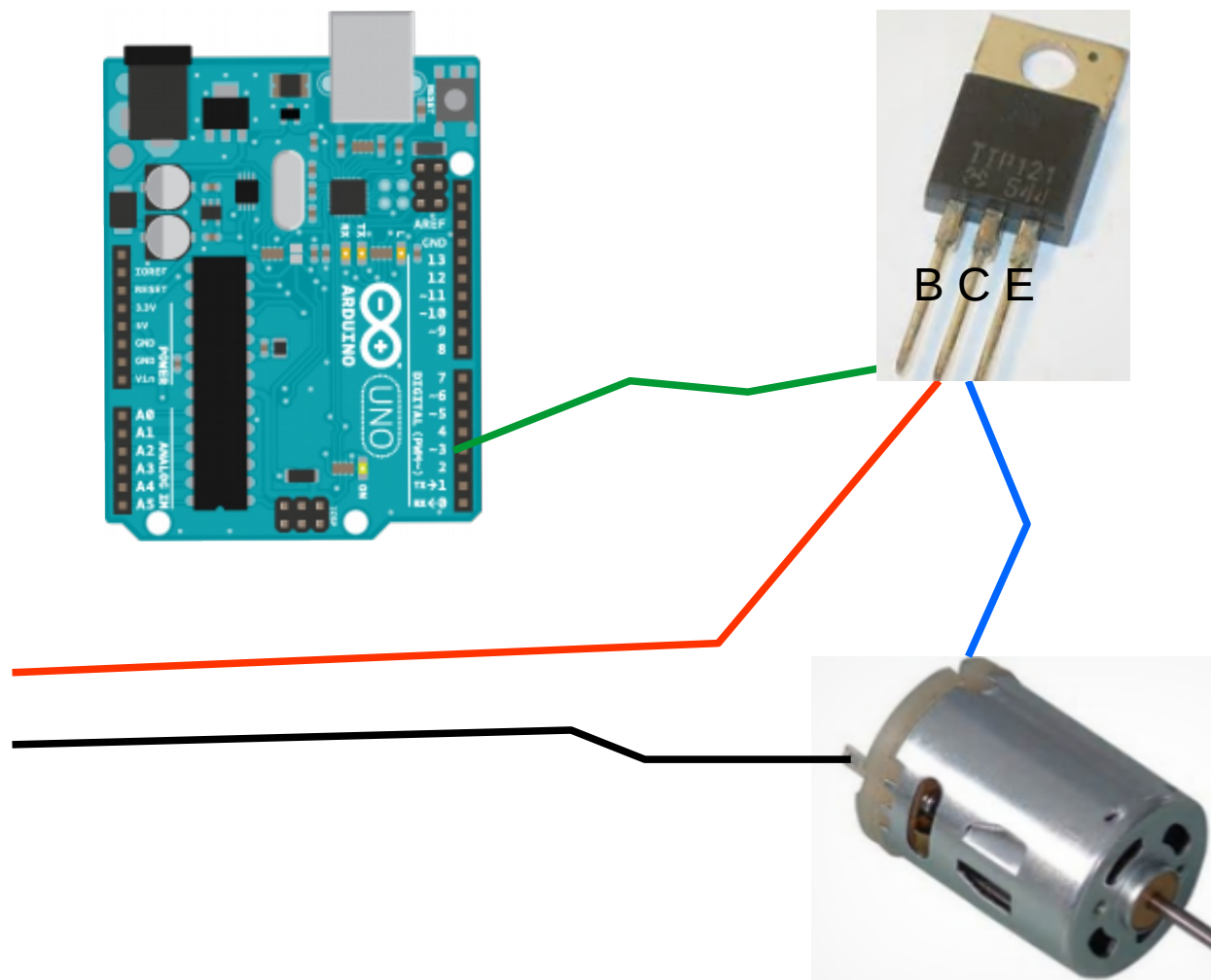
Taller Arduino modulo 5

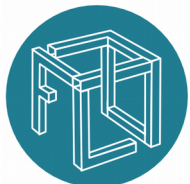


FABLAB

SANTANDER

Potencia: motor DC

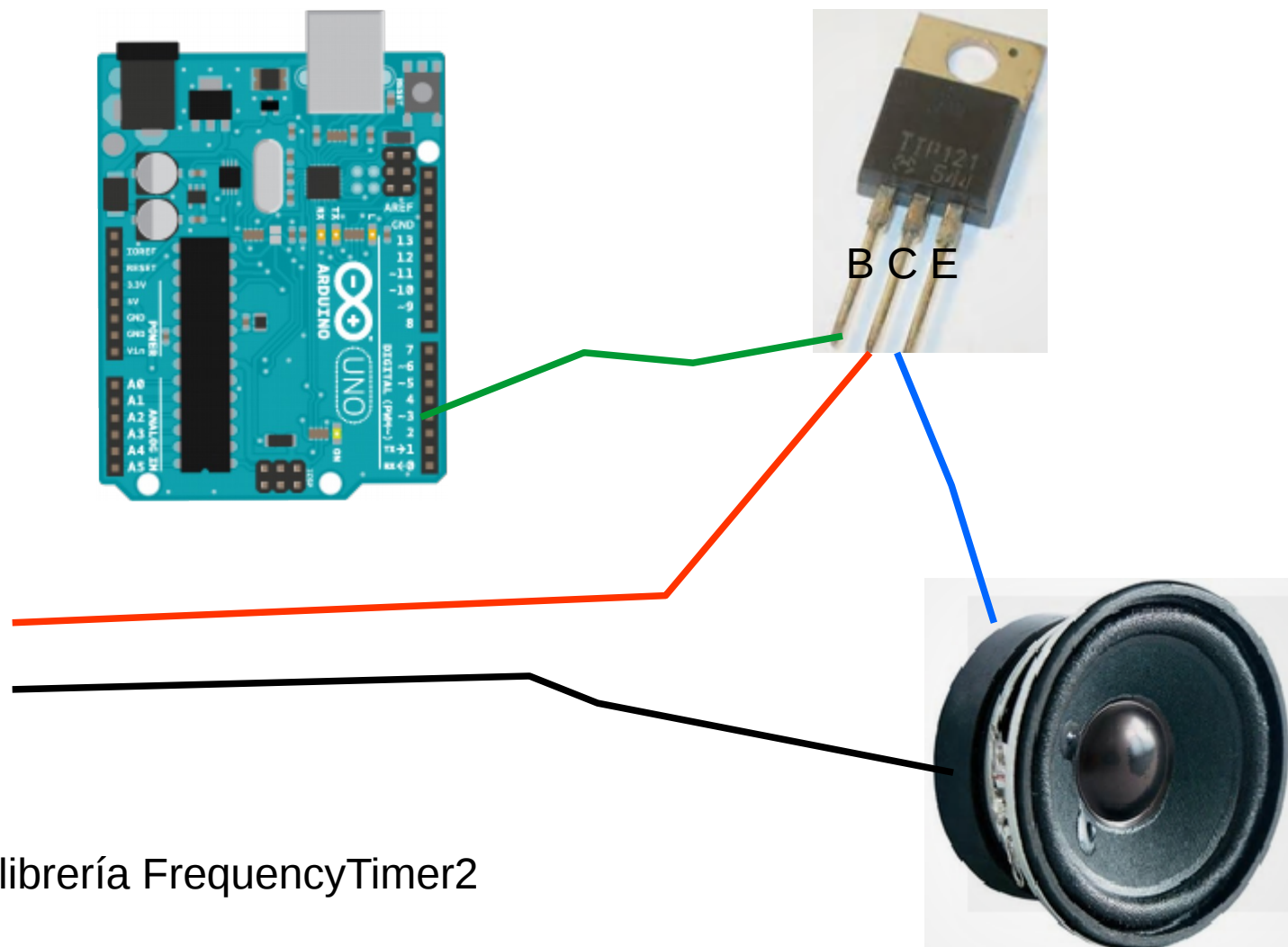




FABLAB

SANTANDER

Potencia: altavoz

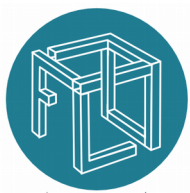


Instalar la librería FrequencyTimer2



FABLAB
SANTANDER

Taller Arduino modulo 6



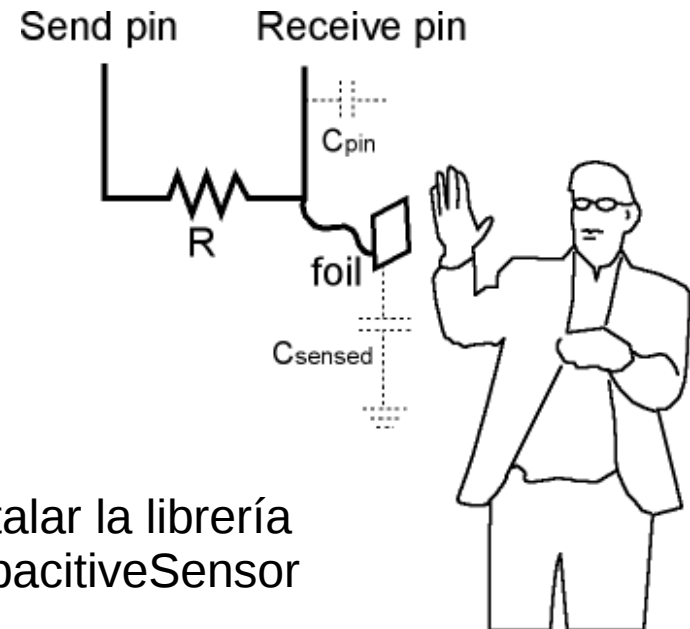
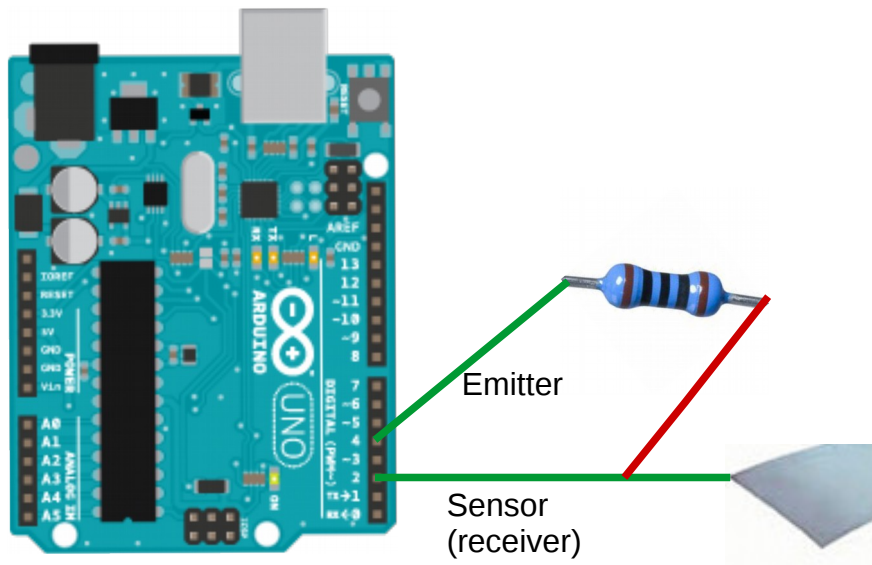
FABLAB

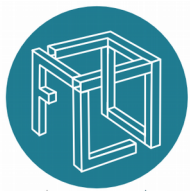
SANTANDER

Sensor capacitivo

El principio del sensor es medir la capacidad es decir la cantidad de electrones que puede acumular el condensador formado por el dedo y la tecla. arduino manda corriente hacia el sensor. Otro pin recibe la corriente. El voltaje medido toma un tiempo para llegar a un umbral: este tiempo es proporcional a la capacidad.

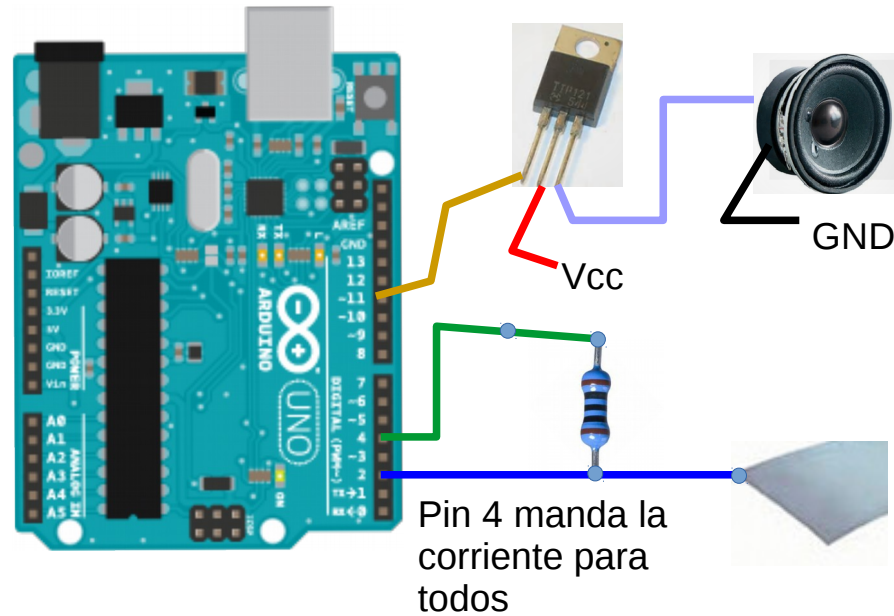
<http://playground.arduino.cc/Main/CapacitiveSensor?f rom=Main.CapSense>





FABLAB
SANTANDER

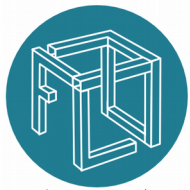
Harpa capacitiva



Instalar las librerías:

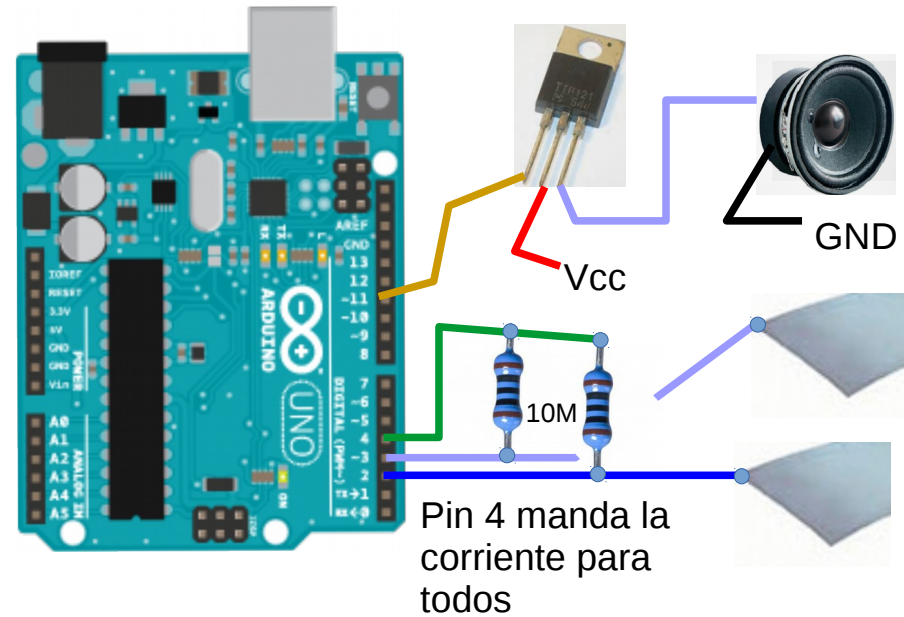
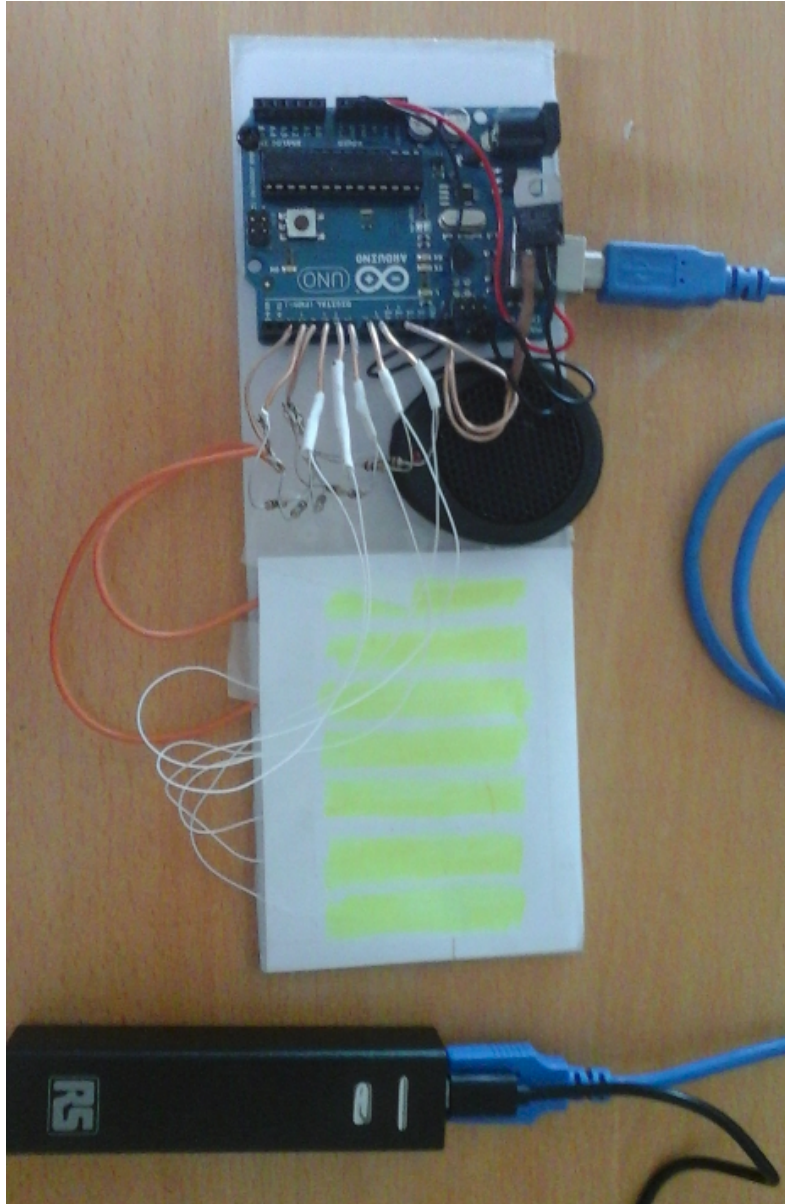
- CapacitiveSensor
- FrequencyTimer2

Modular la frecuencia según la capacidad



FABLAB
SANTANDER

Mini piano



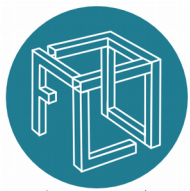
prototipo:

7 sensores
scotch aluminio
resistencias 10Mohms
ampli darlington TIP121

Instalar las librerías:

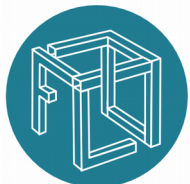
- CapacitiveSensor
- FrequencyTimer2

Ejemplo: miniPiano



FABLAB
SANTANDER

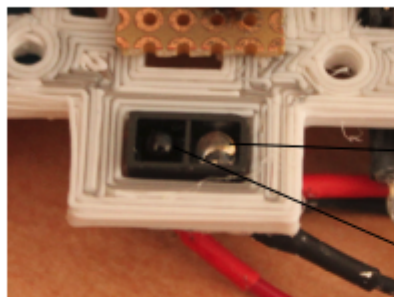
Taller Arduino modulo 7



FABLAB
SANTANDER

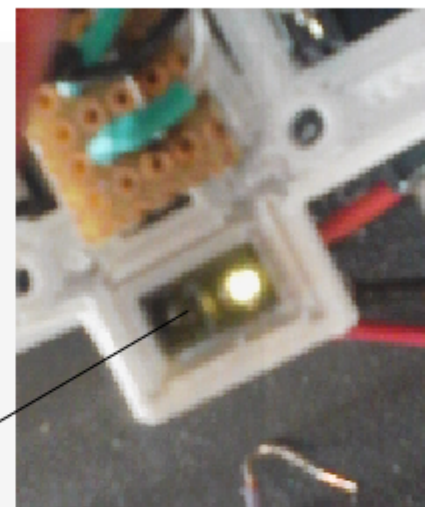
Sensor de proximidad

Mounting of IR proximity sensor:

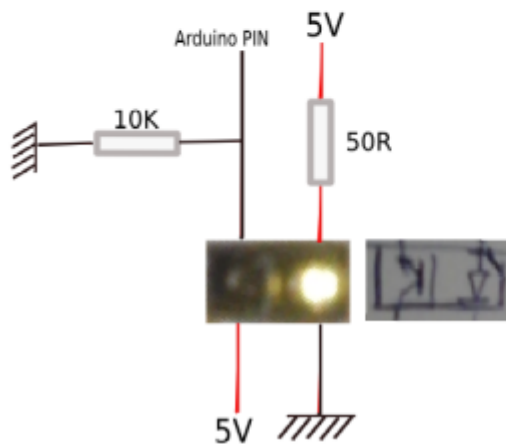


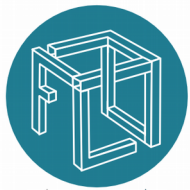
IR LED

Photodiode



For debugging,
IR LED can be seen
by the mobile phone
camera

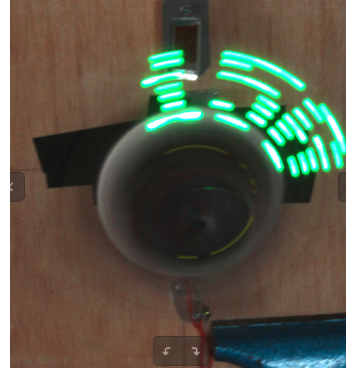




FABLAB

SANTANDER

Interupciones



Permite reaccionar a una accion exterior de manera (casi) inmediata

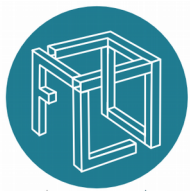
Board

Uno, Nano, Mini, other 328-based
Mega, Mega2560, MegaADK
Micro, Leonardo, other 32u4-based
Zero
MKR1000 Rev.1
Due
101

Digital Pins Usable For Interrupts

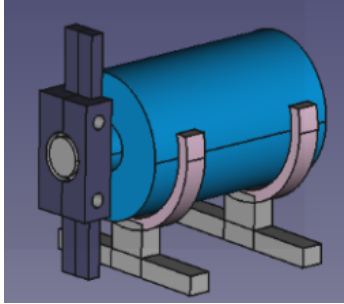
2, 3
2, 3, 18, 19, 20, 21
0, 1, 2, 3, 7
all digital pins, except 4
0, 1, 4, 5, 6, 7, 8, 9, A1, A2
all digital pins
all digital pins

- Una funcion del programa arduino se asocia a un PIN
- Se ejecuta cuando el PIN recibe HIGH
- Arduino deja todo lo que esta haciendo....



FABLAB
SANTANDER

Aplicacion de interrupciones codificador optico



```
int pinSensor=2;//sensor to measure period (or angular speed)  
int c=0;
```

```
void setup()  
{  
  Serial.begin(9600);  
  pinMode(pinSensor,INPUT);  
  pinMode(13,OUTPUT);  
  //configure the interrupt associating  
  //the interrupt 0 (pin 2)  
  //see https://www.arduino.cc/en/Reference/AttachInterrupt )  
  attachInterrupt(0, trig, RISING);  
}
```

```
void trig() { c++; }
```

```
void loop()  
{  
  Serial.println(c);  
  delay(1000);  
}
```

