

Creando una IoT

Un tutorial con Thingspeak, Arduino y ESP32



Taller de Iniciación a los Sistemas Ciber Físicos
MINA - Facultad de Ingeniería - Udelar

Temario

Una Nube: thingspeak

Un microcontrolador: ESP32

Un firmware: Arduino

Programemos nuestro microcontrolador

Conectémonos a la nube

Enviemos datos

Exploremos nuestros datos en la nube



Qué es una nube IoT

thingspeak.com

ThingSpeak™ Channels Apps Devices Support

Public Channels

af104-uwz

Channel ID: 624218
Author: griesu62
Ultraschall-Wasser-Zähler

uwz, lot; water, pulse, l-bus, m-bus

San Diego - Estación...

Channel ID: 1293177
Author: santiago
San Diego, Cerro Largo, Uruguay Estación Meteorológica Solar (Temp, Hum, Presion, Lluvia, Viento). ESP8266, UNO R3, BME 680
Update Interval - 15 seg https://clima.santiago.ovh/

uruguay, cerro largo, san diego, estacion solar, solar, cerro largo uy, uy

Hühnerhof Fischer

Channel ID: 2444711
Author: mwa0000022903516
Wetterdaten aus und um den Hühnerstall des Hühnerhofs Fischer in Hohegeiß

hühnerhof fischer, hohegeiß, knust

WeatherStation

Channel ID: 12397
Author: ewetjen27
MathWorks Weather Station, West Garage, Natick, MA 01760, USA

mathworks weather station, weather, mathworks

Wind Power Smart Mon...

Channel ID: 1785844
Author: mwa0000022903516
This channel is used to monitor the wind speed and carry out analysis on the effect of all these parameters on wind turbine power generation

wind power, wind speed

DTT

Channel ID: 838448
Author: chrisyuk
DTT stream bitrate analysis

dvb-t, freeview, dvb, dtt

Wind and Solar Power...

Pianosa Teltonika

Various variables ...

ThingSpeak™ Channels Apps Devices Support Commercial Use How to Buy JV

Wind Power Smart Monitor

Channel ID: 1785844
Author: mwa0000022903516
Access: Public

This channel is used to monitor the wind speed and carry out analysis on the effect of all these parameters on wind turbine power generation

wind power, wind speed

Export recent data More Information GitHub

MATLAB Analysis MATLAB Visualization

Field 1 Chart

Wind Power Smart Monitor

Wind Speed

10
5
0

15:45 16:00 16:15 16:30 16:45

thingspeak.com

Field 2 Chart

Wind Power Smart Monitor

Wind Power Density (Wattsm^2)

20
0

15:45 16:00 16:15 16:30 16:45

thingspeak.com

Field 3 Chart

Wind Power Smart Monitor

Wind Power (Watts)

200k
0k

15:45 16:00 16:15 16:30 16:45

thingspeak.com

Field 4 Chart

Wind Power Smart Monitor

Air Density (kg/m^3)

0.0506
0.0505

Air Density (kg/m^3): 0.05054455020818259
Fri Mar 01 2024
16:05:34 GMT-0300

15:45 16:00 16:15 16:30 16:45

thingspeak.com



Qué es un microcontrolador (MCU)

Microcontrolador ESP32-PICO-KIT-1

Velocidad de reloj: 80MHz / 160MHz

RAM: 520KB

Flash: 4MB

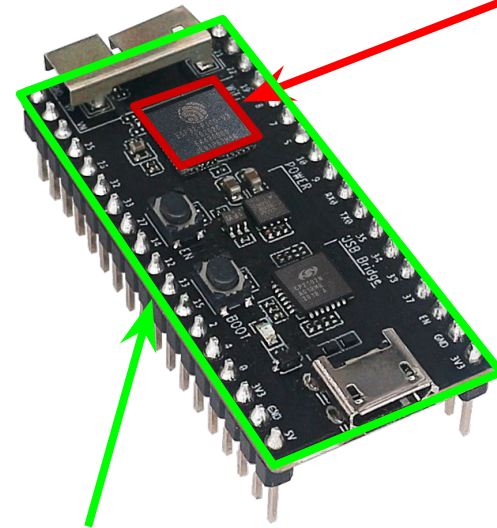
Alimentación: USB (5V)

Voltaje de funcionamiento: 3.3V

WiFi integrada (802.11b/g/n),
Bluetooth (v4, BLE)

Precio: 4-5usd / 10-15usd

ESP32-PICO-V3

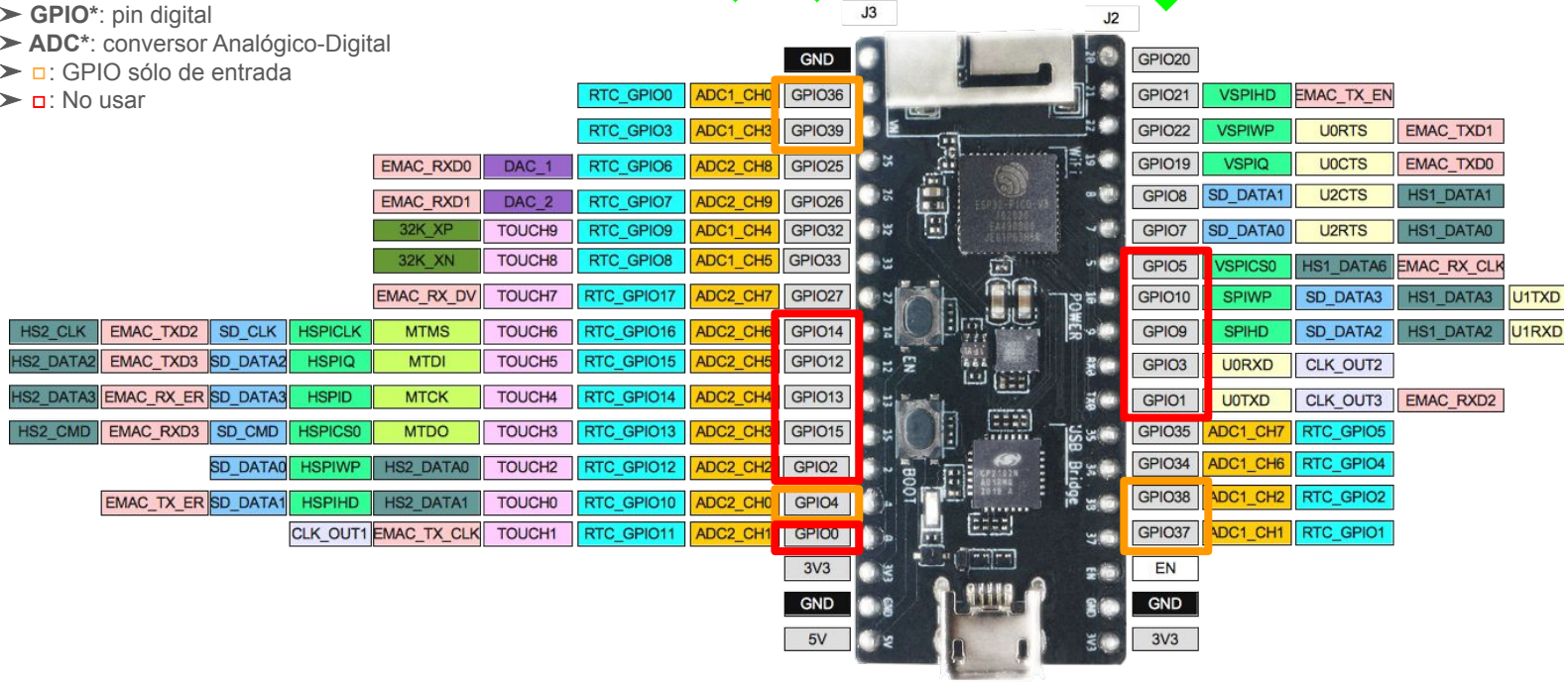


ESP32-PICO-KIT-1



Lo que nos interesa

-



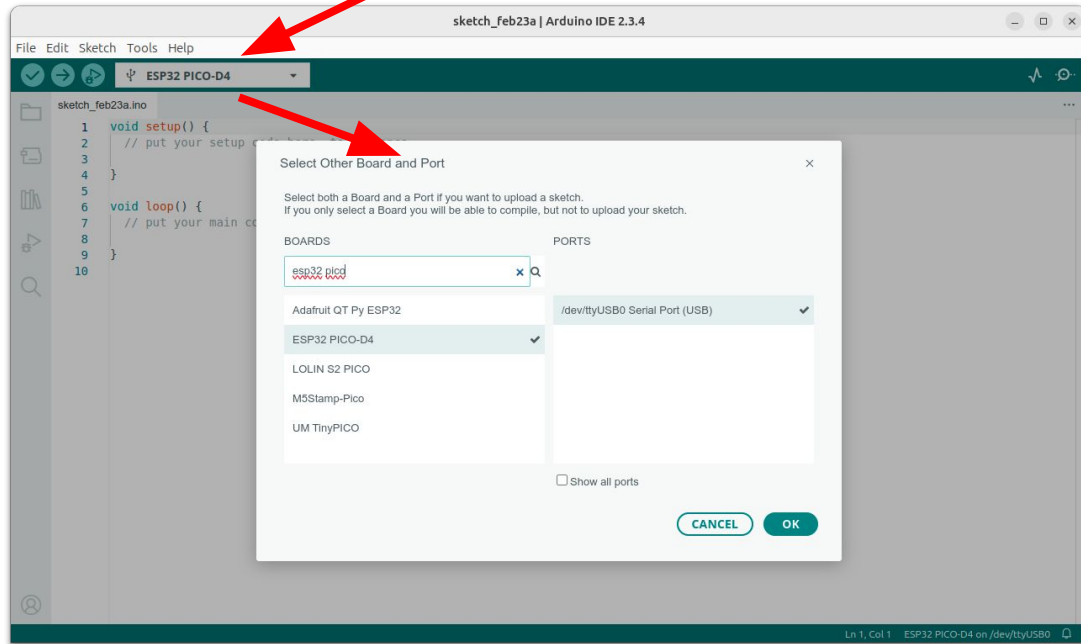
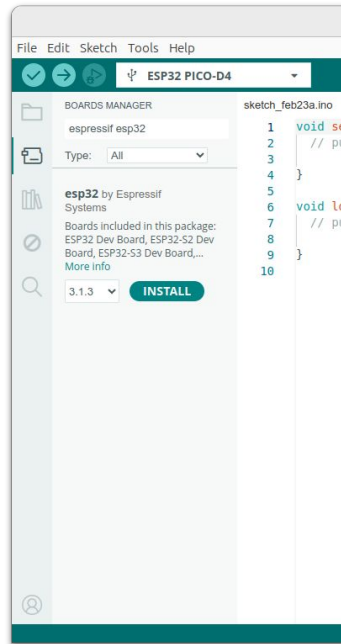
<https://docs.espressif.com/projects/esp-idf/en/v4.4/esp32/hw-reference/esp32/get-started-pico-kit-1.html>



Cómo se programa un MCU

Arduino: agregar soporte para nuestro MCU

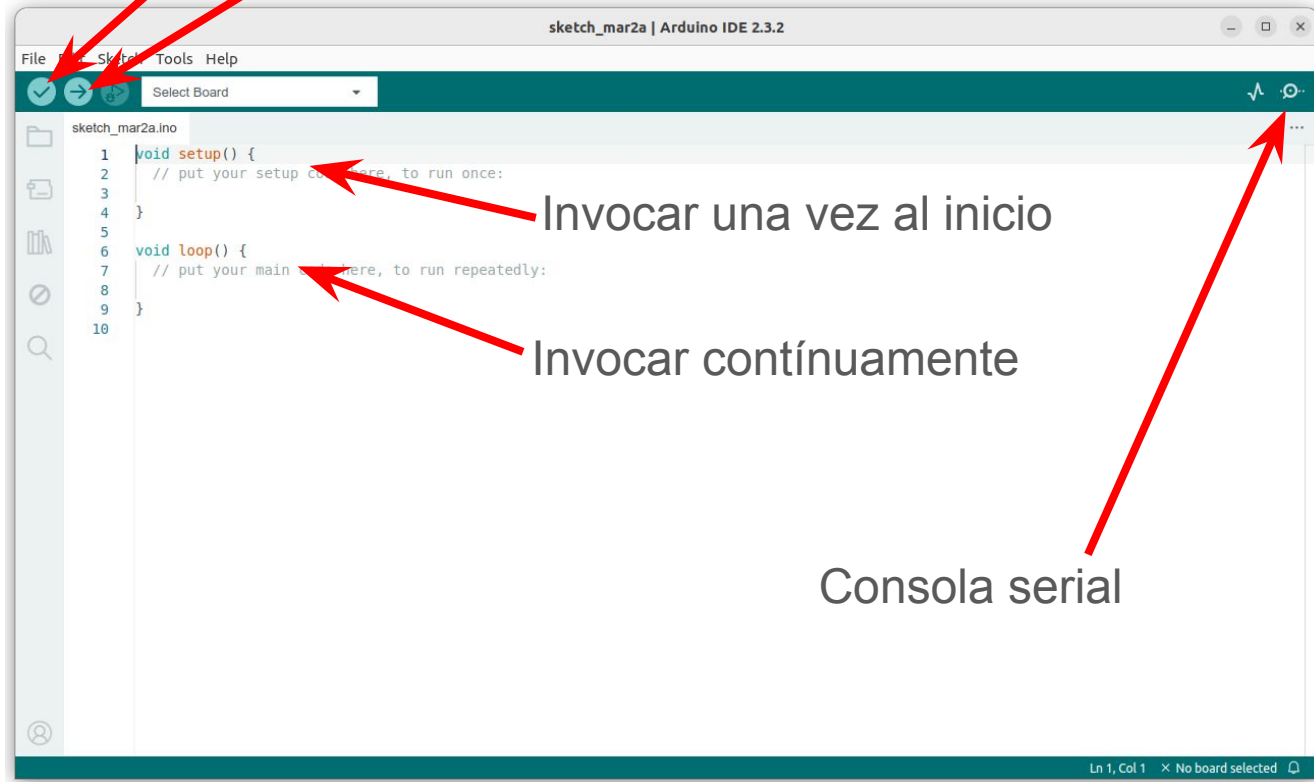
Seleccionar **ESP32 PICO-D4**



Arduino

Verificar (compilar)

Instalar ("flashear")



Arduino: estructura de un programa

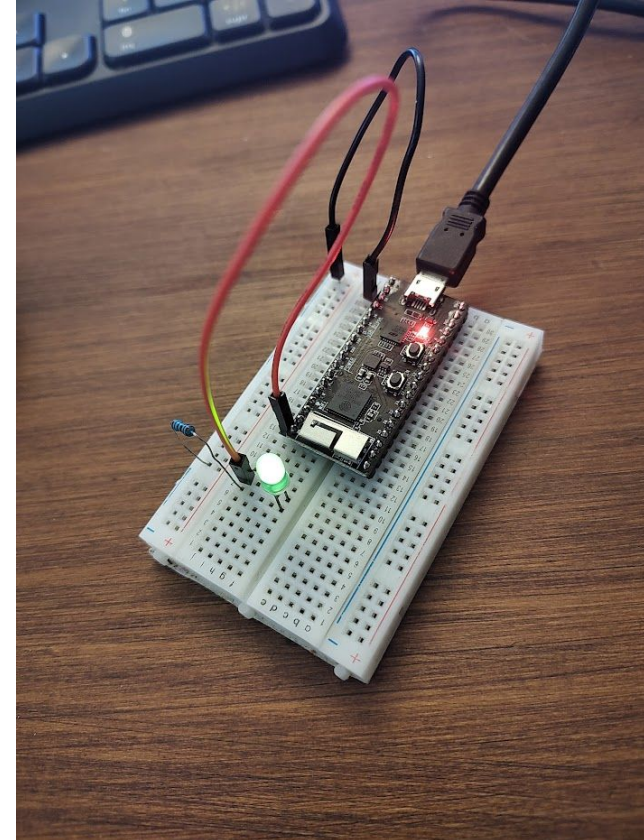
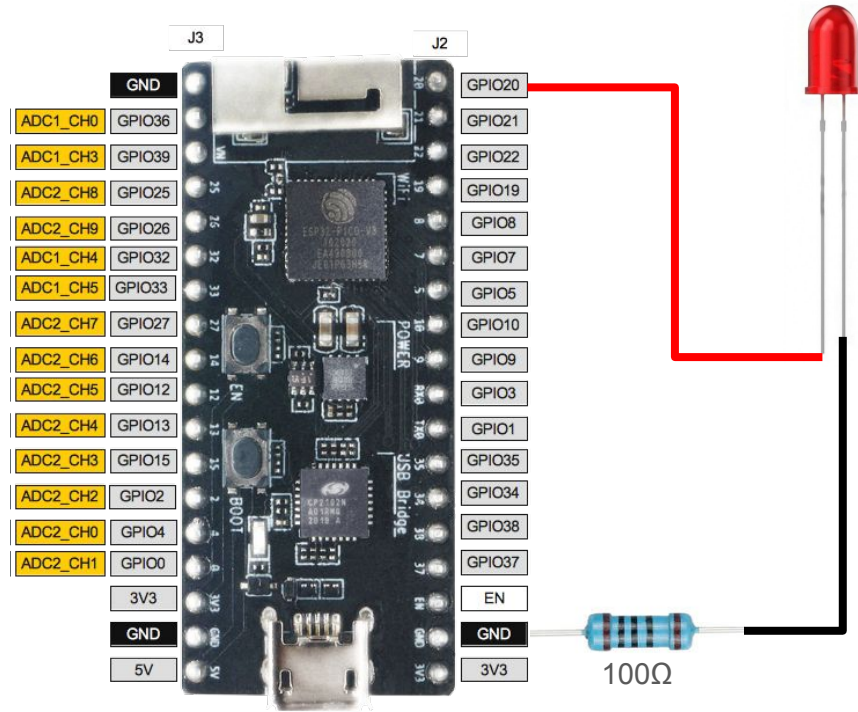
```
void setup() {  
    Serial.begin(115200);  
    Serial.println("Hello World!");  
}  
  
void loop() {  
}
```

```
void setup() {  
    Serial.begin(115200);  
}  
  
void loop() {  
    Serial.println("Hello World!");  
}
```

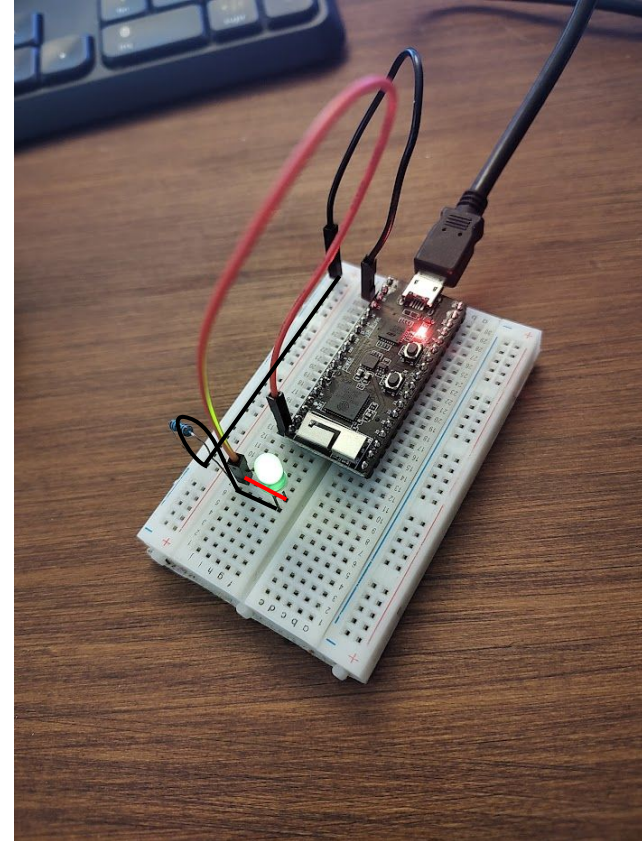
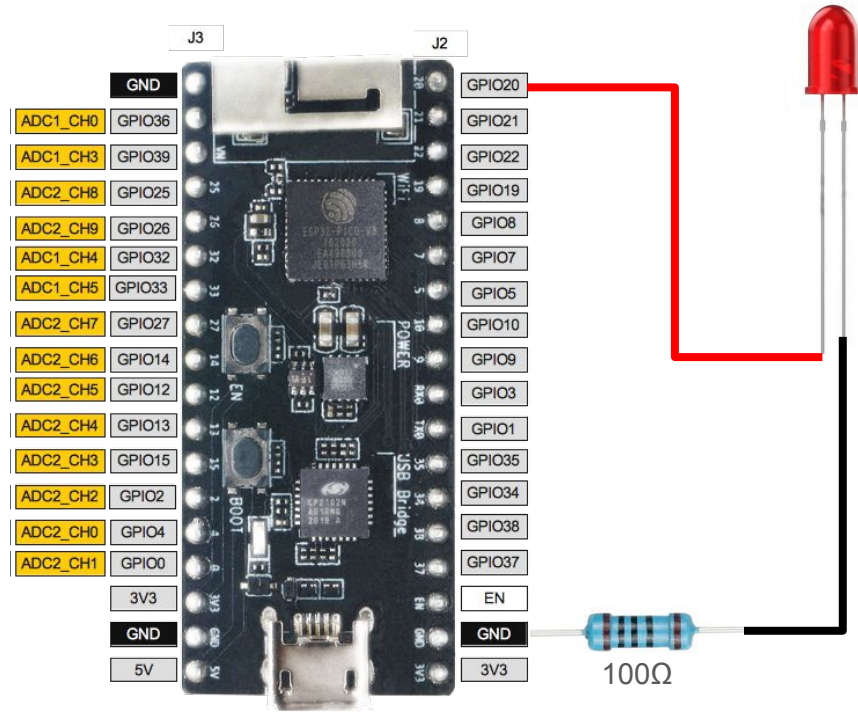


Hola mundo en MCU

Arduino: escribir en pin digital



Arduino: escribir en pin digital



Arduino: escribir en pin digital

```
#define LED_PIN 20

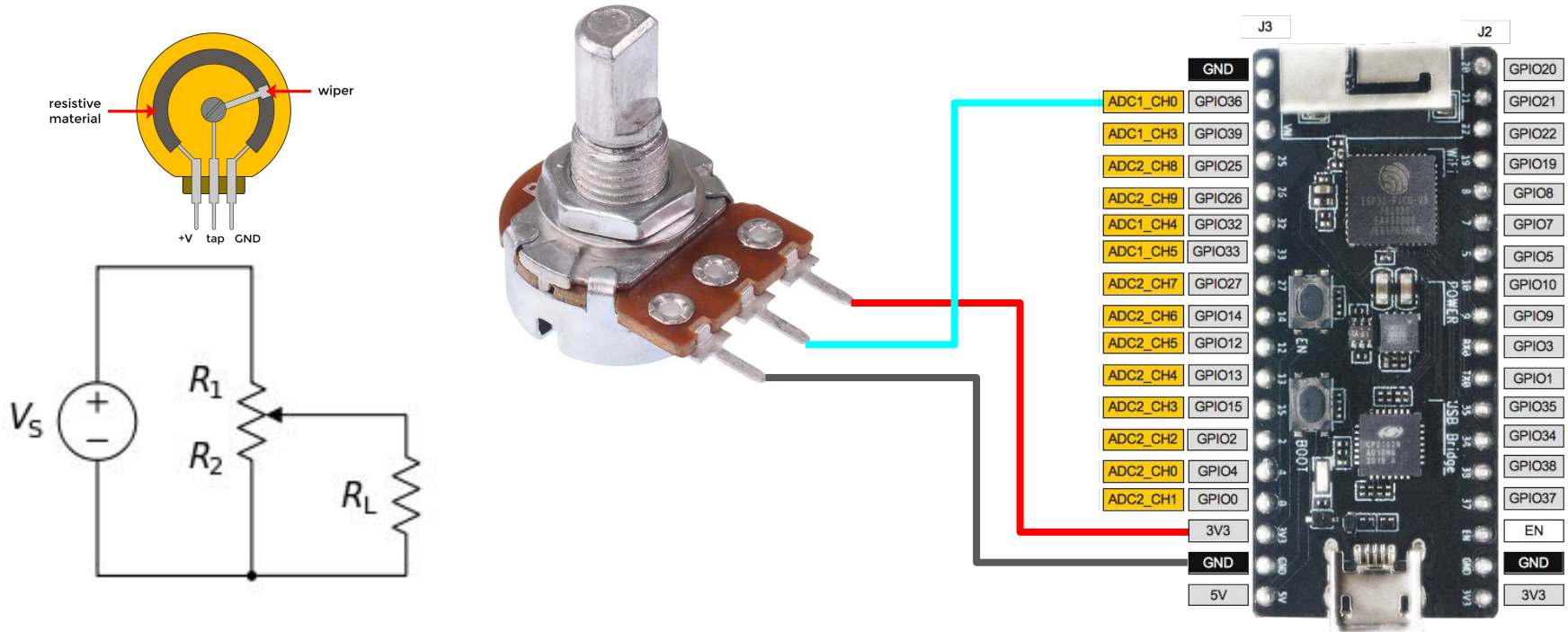
void setup() {
    Serial.begin(115200);
    pinMode(LED_PIN, OUTPUT); // LED pin as output.
    Serial.println("\n\nLED pin: " + String(LED_PIN));
}

void loop() {
    Serial.print("1 ");
    digitalWrite(LED_PIN, HIGH);
    delay(1000);
    Serial.print("0 ");
    digitalWrite(LED_PIN, LOW);
    delay(1000);
}
```

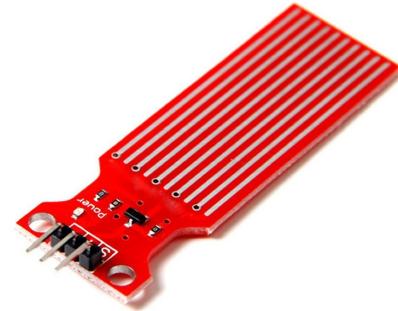
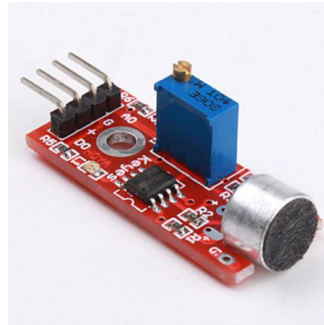
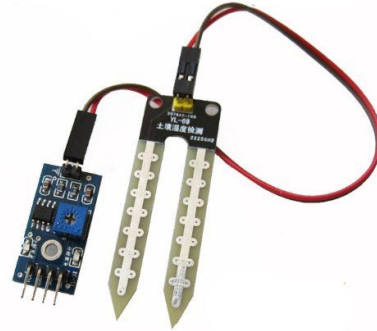
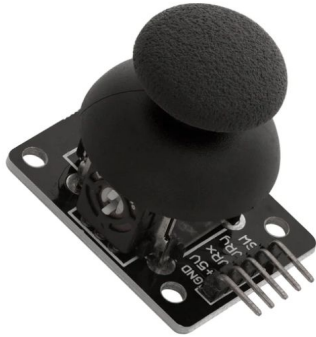


Cómo se lee un sensor analógico

Arduino: cablear una resistencia variable



Sensores analógicos



Arduino: conversor analógico-digital (ADC)

```
#define R1_PIN 36
#define LED_PIN 20

int last_change = 0;
int led_state = 0;

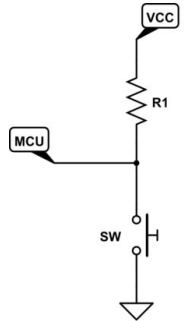
void setup() {
    Serial.begin(115200);
    pinMode(LED_PIN, OUTPUT);
}

void loop() {
    int now = millis();
    int delay = analogRead(R1_PIN);
    if (now - last_change > delay) {
        last_change = now;
        led_state = 1 - led_state;
        Serial.println(String(delay));
        digitalWrite(LED_PIN, led_state);
    }
}
```

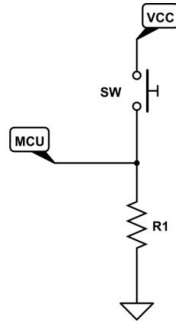


Cómo se lee un sensor digital

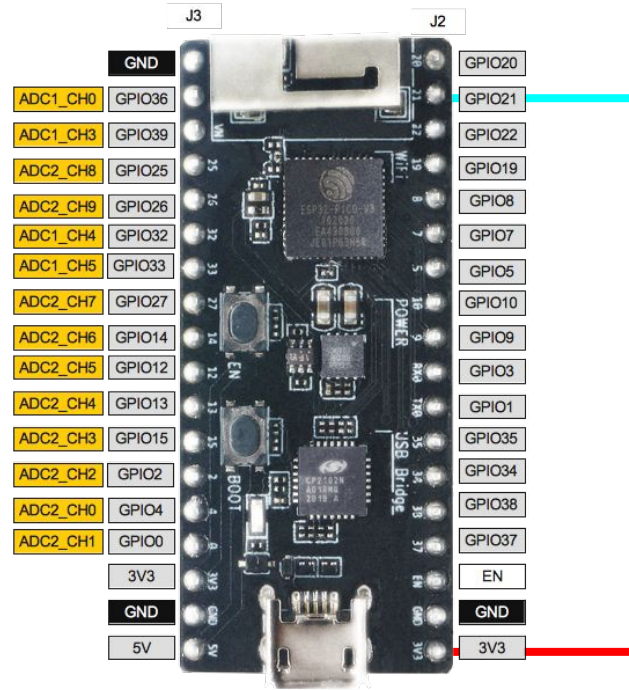
Arduino: cablear un sensor digital



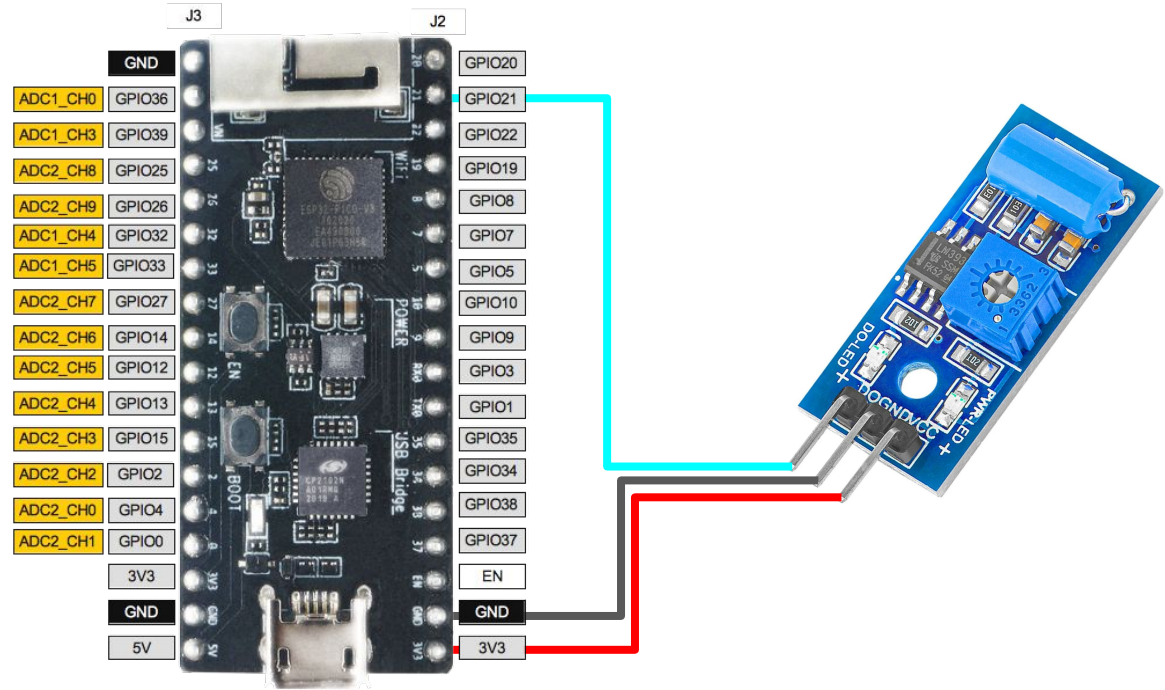
Pull-up method



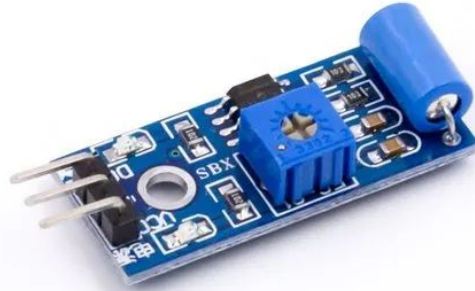
Pull-down method



Arduino: cablear un sensor digital



Sensores digitales



Arduino: leer un sensor digital (polling)

```
const int buttonPin = 21;

void setup() {
  Serial.begin(115200);
  pinMode(buttonPin, INPUT_PULLDOWN);
}

void loop() {
  int buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH) {
    Serial.println("on");
  } else {
    Serial.println("off");
  }
}
```

```
const int buttonPin = 21;
int lastState = LOW;

void setup() {
  Serial.begin(115200);
  pinMode(buttonPin, INPUT_PULLDOWN);
}

void loop() {
  int buttonState = digitalRead(buttonPin);
  if (lastState != buttonState) {
    Serial.println(buttonState);
    lastState = buttonState;
  }
}
```



Arduino: leer un sensor digital (interrupciones)

```
const int buttonPin = 21;
volatile byte clicked = 0;

void setup() {
  Serial.begin(115200);
  pinMode(buttonPin, INPUT_PULLDOWN);
  attachInterrupt(digitalPinToInterrupt(buttonPin), click, RISING);
}

void loop() {
  if (clicked == 1) {
    Serial.println(millis());
    clicked = 0;
  }
}

ICACHE_RAM_ATTR void click() {
  clicked = 1;
}
```



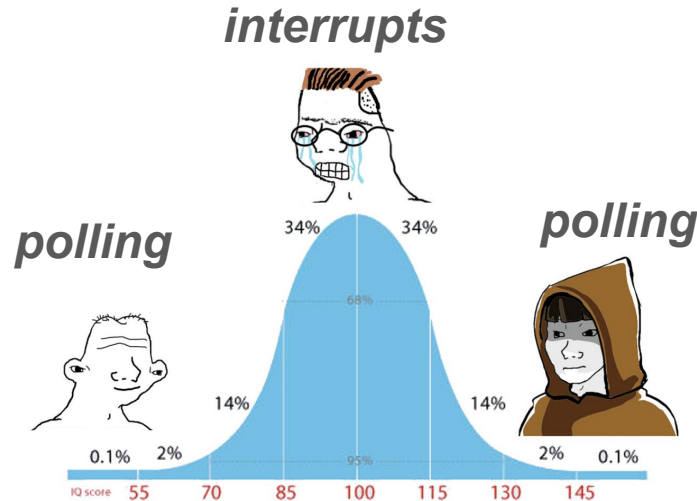
Arduino: leer un sensor digital (interrupciones)

- Interrupciones en `CHANGE` / `FALLING` / `RISING`
- Los handlers de las interrupciones no reciben ni devuelven parámetros
 - Deben comunicarse con el resto del programa mediante variables globales
- Las variables globales que se actualizan en un handler deben ser declaradas `volatile`.
- Los handlers deben ser declarados `ICACHE_RAM_ATTR`.
- Se pueden proteger secciones de código con `noInterrupts()` / `interrupts()`; Usar responsablemente.
- Una interrupción no interrumpe a otra interrupción
 - Durante una interrupción “nada sucede”
 - El tiempo (`millis()`) no avanza
 - El handler debe ser lo más corto y rápido posible
 - No usar `delay()`
 - En lo posible no usar el puerto serial, que incluye la consola

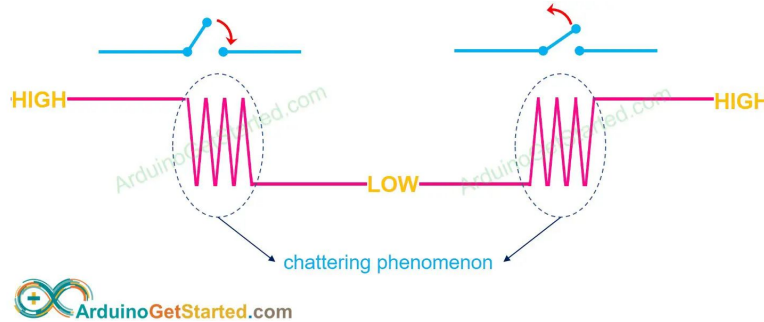


Arduino: leer un sensor digital (interrupciones)

- Polling: fácil, obtuso, tendencia a ineficiente y confuso.
- Interrupciones: elegantes, sutiles, eficientes, peligrosas.



Arduino: leer un sensor digital (*debounce*)



```
#define BOUNCE_TIME 50
static int lastDigitalOn = 0;

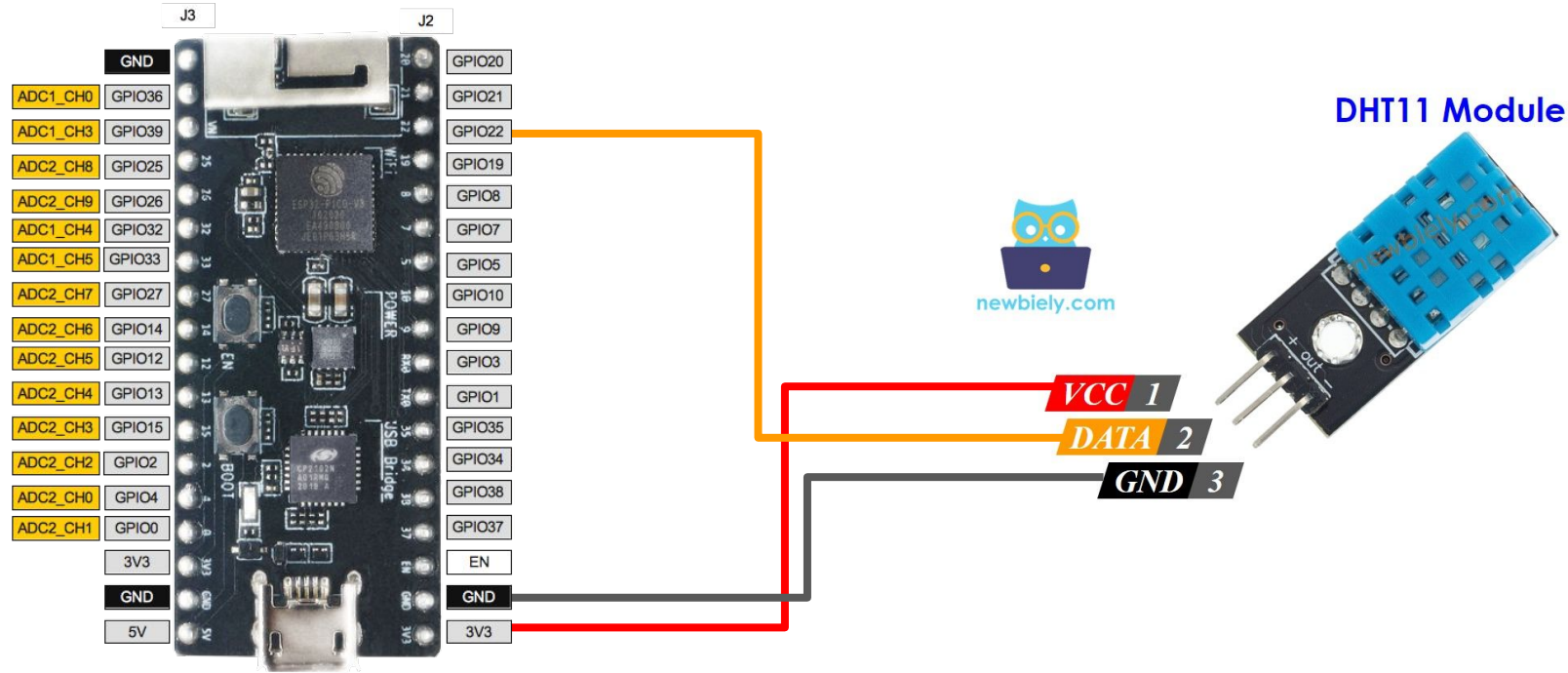
...
if( digitalRead(button) && (millis()-lastDigitalOn > BOUNCE_TIME) ) {
    lastDigitalOn = millis();
    // do something
}

...
```

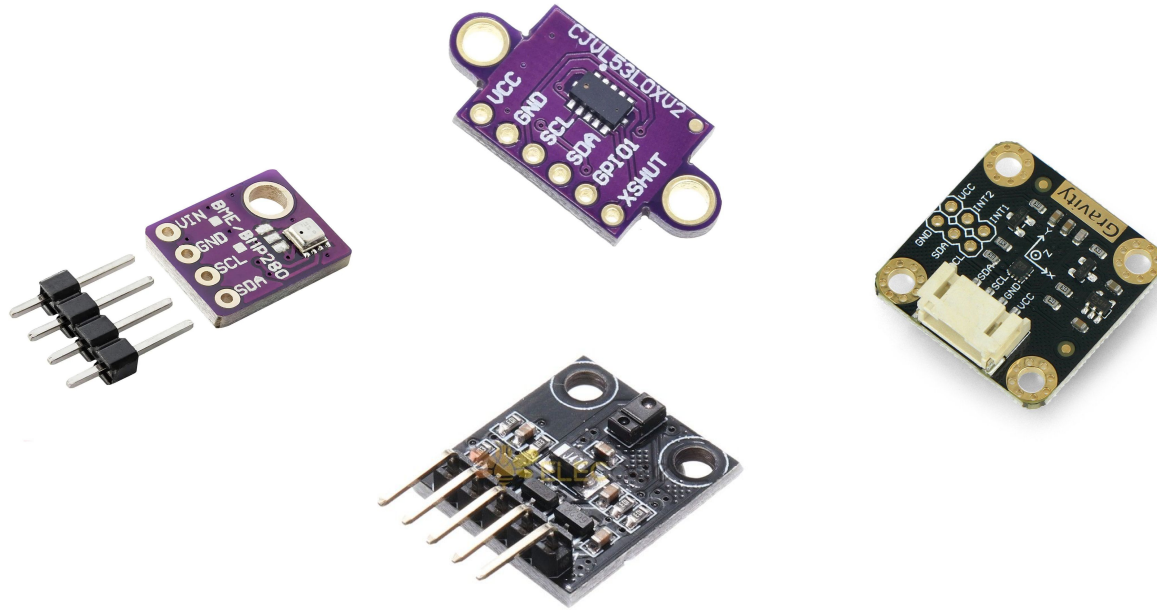


Cómo se lee un sensor basado en
un protocolo (ej: 1-wire)

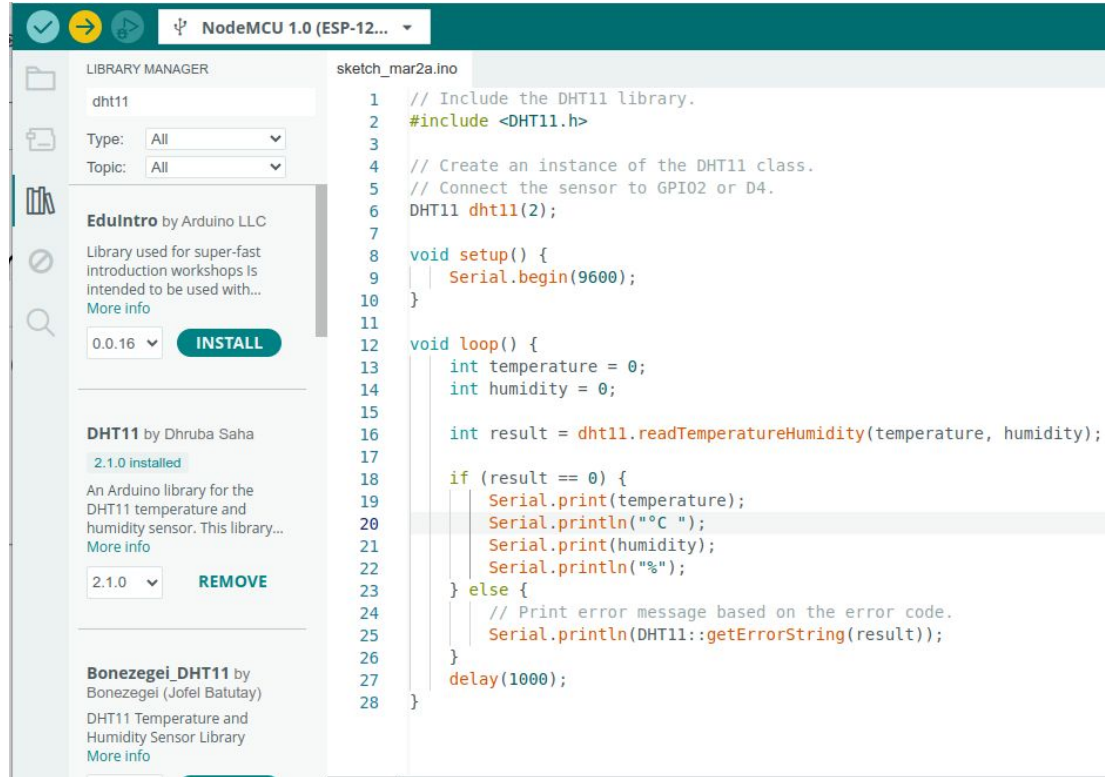
Arduino: cablear un sensor 1-wire



Sensores 1-wire, I²C, SPI...



Arduino: agregar y usar la biblioteca de un sensor



The screenshot displays the Arduino IDE interface. On the left, the 'LIBRARY MANAGER' panel is open, showing a search for 'dht11'. The 'DHT11' library by Dhruba Saha is listed as '2.1.0 installed'. The main editor area shows a sketch named 'sketch_mar2a.ino' with the following code:

```
1 // Include the DHT11 library.
2 #include <DHT11.h>
3
4 // Create an instance of the DHT11 class.
5 // Connect the sensor to GPIO2 or D4.
6 DHT11 dht11(2);
7
8 void setup() {
9   Serial.begin(9600);
10 }
11
12 void loop() {
13   int temperature = 0;
14   int humidity = 0;
15
16   int result = dht11.readTemperatureHumidity(temperature, humidity);
17
18   if (result == 0) {
19     Serial.print(temperature);
20     Serial.println("°C ");
21     Serial.print(humidity);
22     Serial.println("%");
23   } else {
24     // Print error message based on the error code.
25     Serial.println(DHT11::getErrorString(result));
26   }
27   delay(1000);
28 }
```



Arduino: leer sensor DHT11

```
#include <DHT11.h> // Include the DHT11 library.
DHT11 dht11(D4); // Connect the sensor to GPIO2 (D4)

void setup() {
    Serial.begin(115200);
}

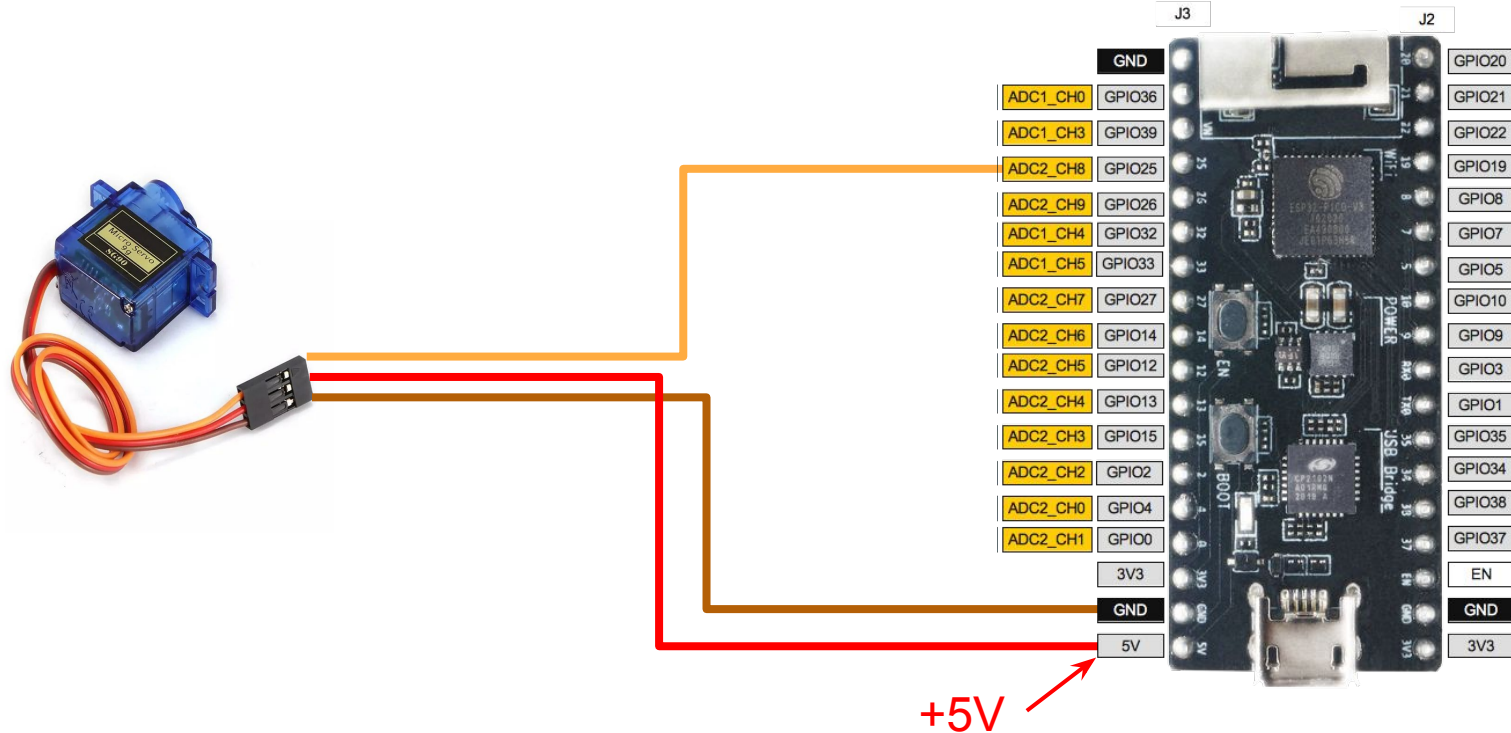
void loop() {
    int temperature = 0;
    int humidity = 0;

    int result = dht11.readTemperatureHumidity(temperature, humidity);
    if (result == 0) {
        Serial.print(temperature); Serial.print("°C ");
        Serial.print(humidity); Serial.println("%");
    } else {
        Serial.println(DHT11::getErrorString(result));
    }
    delay(1000);
}
```



Cómo se controla un servo-motor

Arduino: cablear un servo-motor



Arduino: controlar un servo-motor

```
#include <Servo.h>

Servo servo1;

void setup() {
  Serial.begin(115200);
  servo1.attach(25); // servo attached to GPIO25 pin
}

void loop() {
  Serial.println(0);
  servo1.write(0);
  delay(1000);

  Serial.println(180);
  servo1.write(180);
  delay(1000);
}
```



Cómo se envían datos a la nube

Creemos un canal

1

ThingSpeak™ Channels ▾ Apps ▾ Devices ▾ Support ▾

My Channels

New Channel

Search by tag

2

ThingSpeak™ Channels ▾ Apps ▾ Devices ▾ Support ▾

New Channel

Name

Description

Field 1 ☒

Field 2 ☒

Field 3 ☐

3

ThingSpeak™ Channels ▾ Apps ▾ Devices ▾ Support ▾

Mi sensor

Channel ID: 2454031
Author: mwa0000033127655
Access: Private

Private View Public View Channel Settings Sharing API Keys Data Impo

Write API Key

Key

Generate New Write API Key



Publiquemos en thingspeak

sketch_mar17a | Arduino IDE 2.3.4

File Edit Sketch Tools Help

ESP32 PICO-D4

LIBRARY MANAGER

thingspeak

Type: All

Topic: All

ThingSpeak by MathWorks
<support@thingspeak.com>
2.0.1 installed

ThingSpeak Communication Library for Arduino, ESP8266 & ESP32 ThingSpeak (https://www.thingspeak.com) is an analytic IoT...

More info

2.0.1 REMOVE

ThingSpeak_asukiaaa by Asuki Kono
An API manager for ThingSpeak It writes field

sketch_mar17a.ino

```
1 #include <WiFi.h>
2 #include "ThingSpeak.h"
3
4 unsigned long myChannelNumber = 2454031;
5 const char * myWriteAPIKey = "XXXXXXXXXXXXXXXXX";
6
7 const char* ssid = "wifing"; // your network SSID (name)
8 const char* password = "wifing-pub"; // your network password
9
10 WiFiClient client;
11
12 void setup() {
13   Serial.begin(115200); //Initialize serial
14   delay(100);
15   WiFi.mode(WIFI_STA);
16
17   ThingSpeak.begin(client); // Initialize ThingSpeak
18 }
```



Publiquemos en thingspeak (1 campo)

```
#include "ThingSpeak.h"

unsigned long myChannel = 2454031;
const char* writeAPIKey = "7QYPX39RABNLVJR3";

#include <WiFi.h>
char ssid[] = "wifing";
char pass[] = "wifing-pub";
WiFiClient client;

void setup() {
    Serial.begin(115200);
    delay(100);
    WiFi.mode(WIFI_STA);
    ThingSpeak.begin(client);
}

void loop() {
    if (WiFi.status() != WL_CONNECTED) {
        Serial.print("Attempting to connect to Wifi...");
        while (WiFi.status() != WL_CONNECTED) {
            WiFi.begin(ssid, pass);
            delay(5000);
        }
    }

    long rssi = WiFi.RSSI();
    int httpCode = ThingSpeak.writeField(myChannel, 1, rssi, writeAPIKey);
    if (httpCode == 200) {
        Serial.println("Channel write successful.");
    } else {
        Serial.println("HTTP error code " + String(httpCode));
    }

    delay(20000);
}
```



Publiquemos en thingspeak (varios campos)

```
void loop() {  
    if (WiFi.status() != WL_CONNECTED) {  
        while (WiFi.status() != WL_CONNECTED) {  
            WiFi.begin(ssid, pass);  
            delay(5000);  
        }  
    }  
  
    long rssi = WiFi.RSSI();  
    int valueA0 = analogRead(A0);  
    ThingSpeak.setField(1, rssi);  
    ThingSpeak.setField(2, valueA0);  
  
    int httpCode = ThingSpeak.writeFields(myChannel, myWriteAPIKey);  
    if (httpCode == 200) {  
        Serial.println("Channel write successful.");  
    } else {  
        Serial.println("Problem writing to channel. HTTP error code " + String(httpCode));  
    }  
    delay(20000);  
}
```



Fin