

Manual de Usuario



Introducción

Este manual está diseñado para guiarte en el uso de un entrenador de ESP32, que permite la medición de variables ambientales como luz, humedad y temperatura. Aquí encontrarás tanto un ejemplo básico de hardware y programación. Este manual esta orientado a programadores principiantes entusiastas, se te ayudará a entender las capacidades del ESP32, las diferencias entre sus pines y su uso practico.

Objetivo

Demostrar de forma sencilla el funcionamiento y las capacidades del ESP32 a través de un ejemplo práctico de medición de variables ambientales.

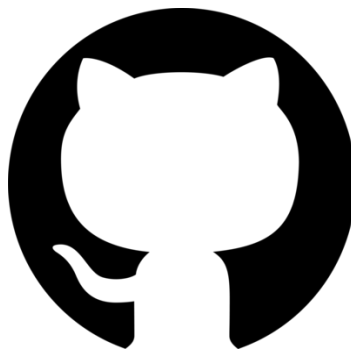
Lista de materiales

- Un ESP32 DevKit V1.
- Un DHT11.
- Una fotorresistencia (De manera opcional de referencia LDR 5537).
- Una placa de circuito impreso (PCB).
- Un cable de datos de USB a micro USB.
- Cautín.
- Estaño.
- Un L7805CV.
- Un computador con Arduino IDE instalado.
- PLA para impresora 3D.
- Headers Hembra.
- Una bateria de 9v.

Elaboración del proyecto

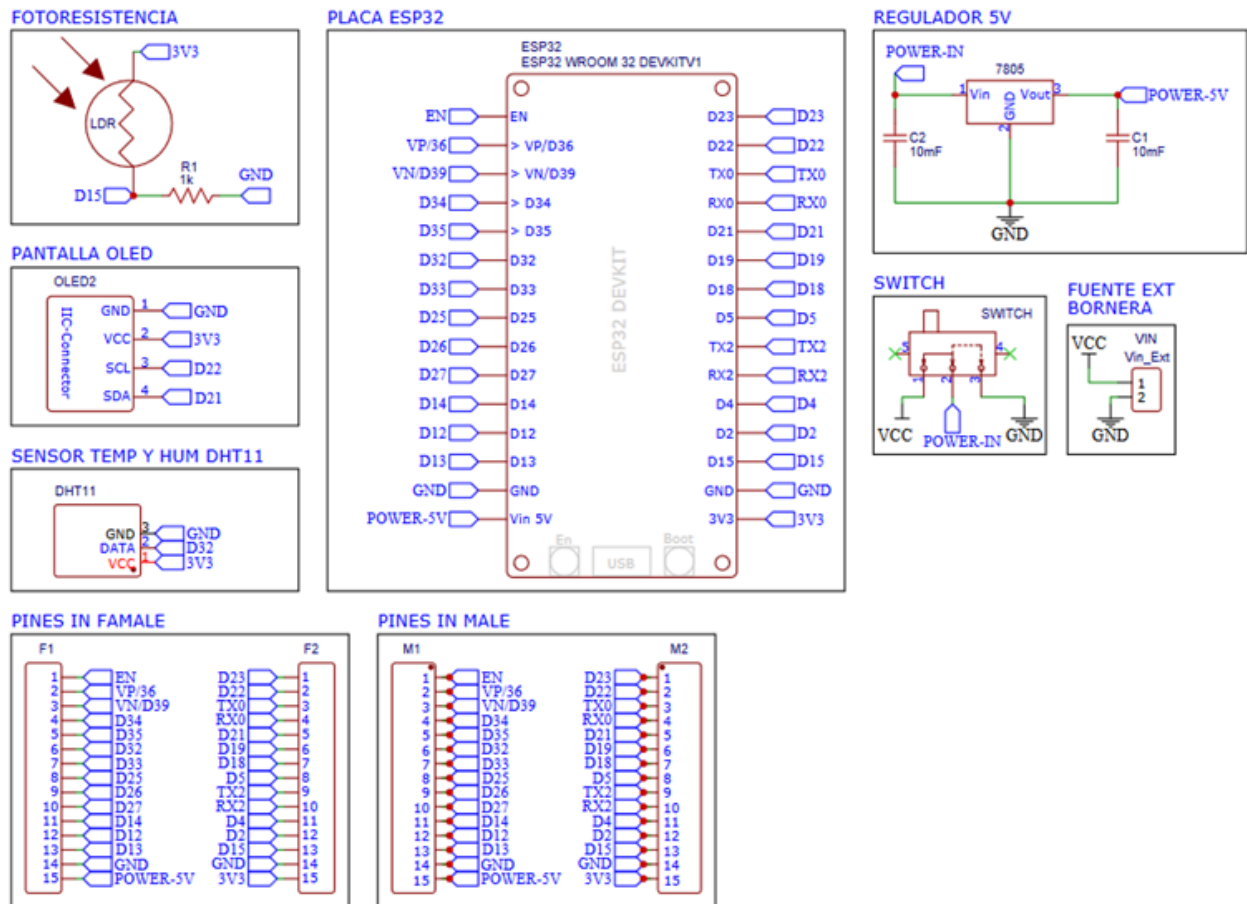
Paso 1. Descarga de los documentos

El primer paso es descargar la carpeta ubicada en el repositorio de GitHub, la cual esta comprimida en un archivo de tipo .zip. Una vez descargada se debe descomprimir para tener acceso a todos los documentos necesarios para la realización del prototipo.

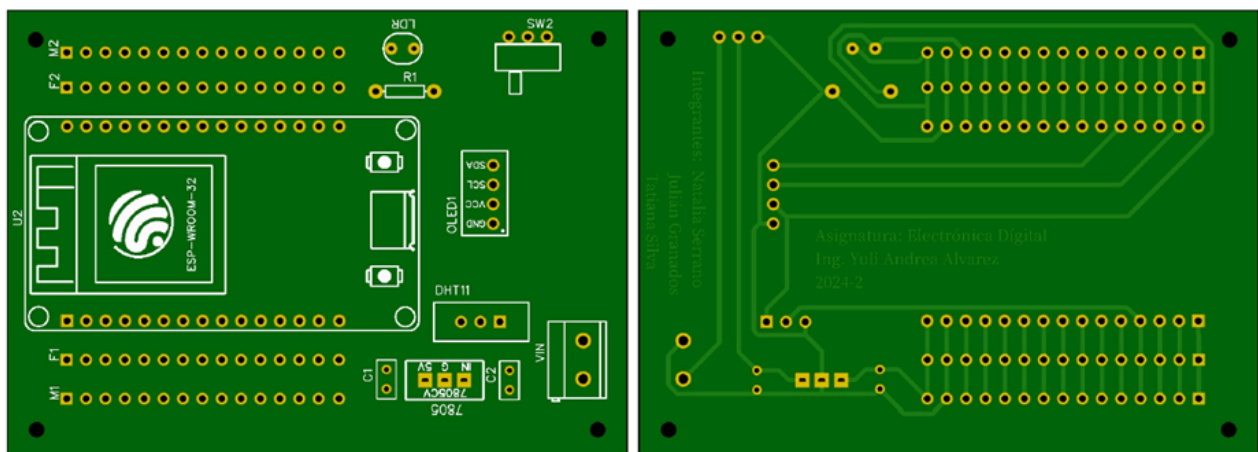


Paso 2. Elaboración de la PCB

Dentro de la carpeta descomprimida se puede encontrar un diseño de EasyEDA del circuito impreso, el cual es necesario lo envíes a un sitio especializado para su impresión.

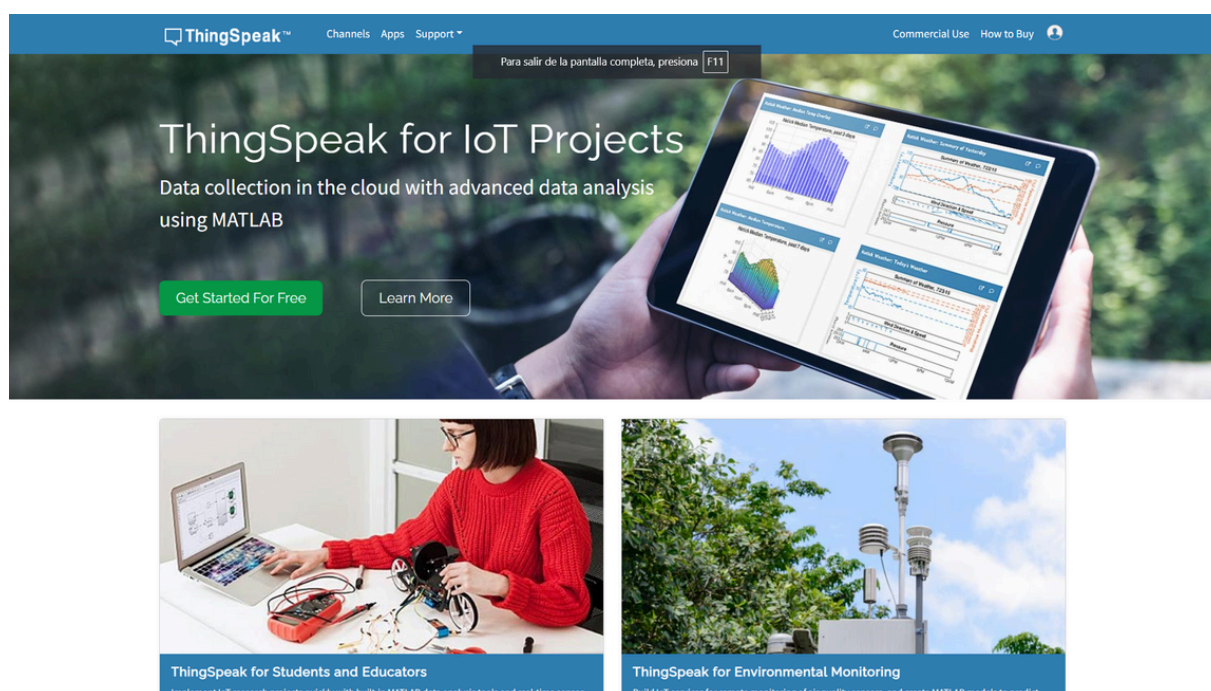


Cuando ya se tenga el circuito impreso es necesario soldar, con el cautín y estaño, los componentes a la placa, recomendamos el uso de headers tipo hembra para proteger los sensores y el esp32, para su uso posteriormente en otros proyectos. Todos los componentes tienen una ubicación asignada y marcada dentro de la placa.

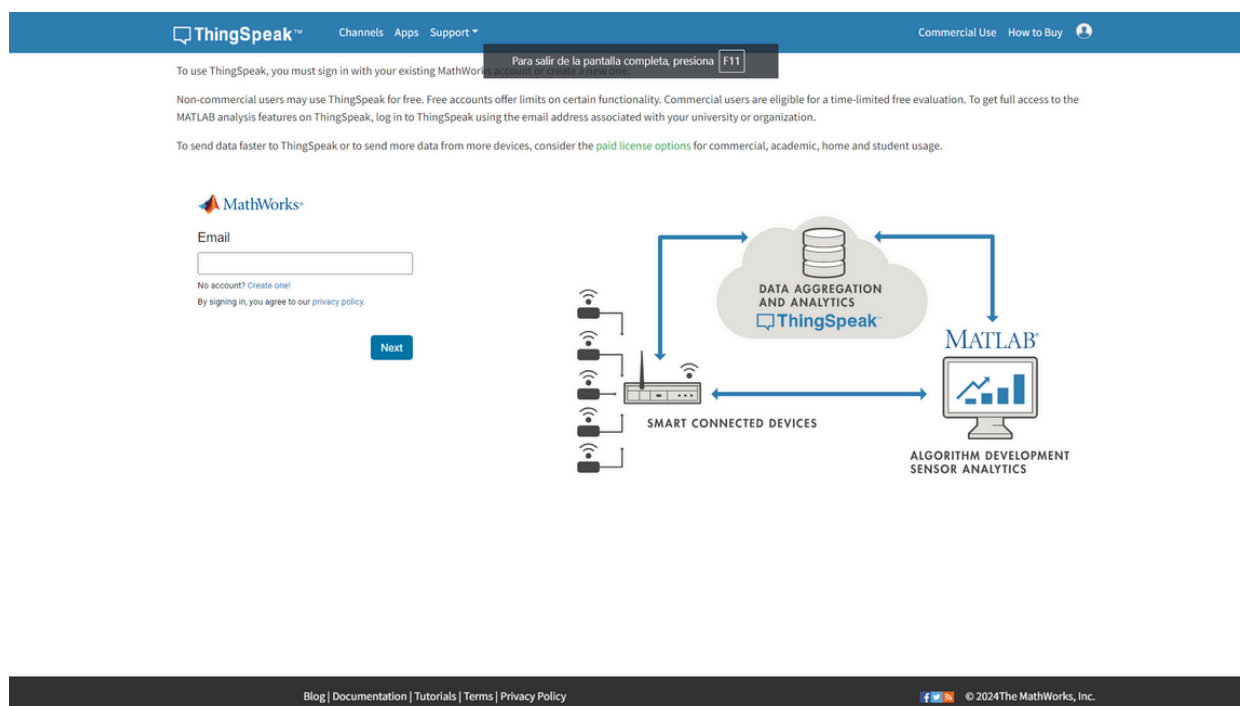


Paso 3. Creación de la plataforma en ThingSpeak

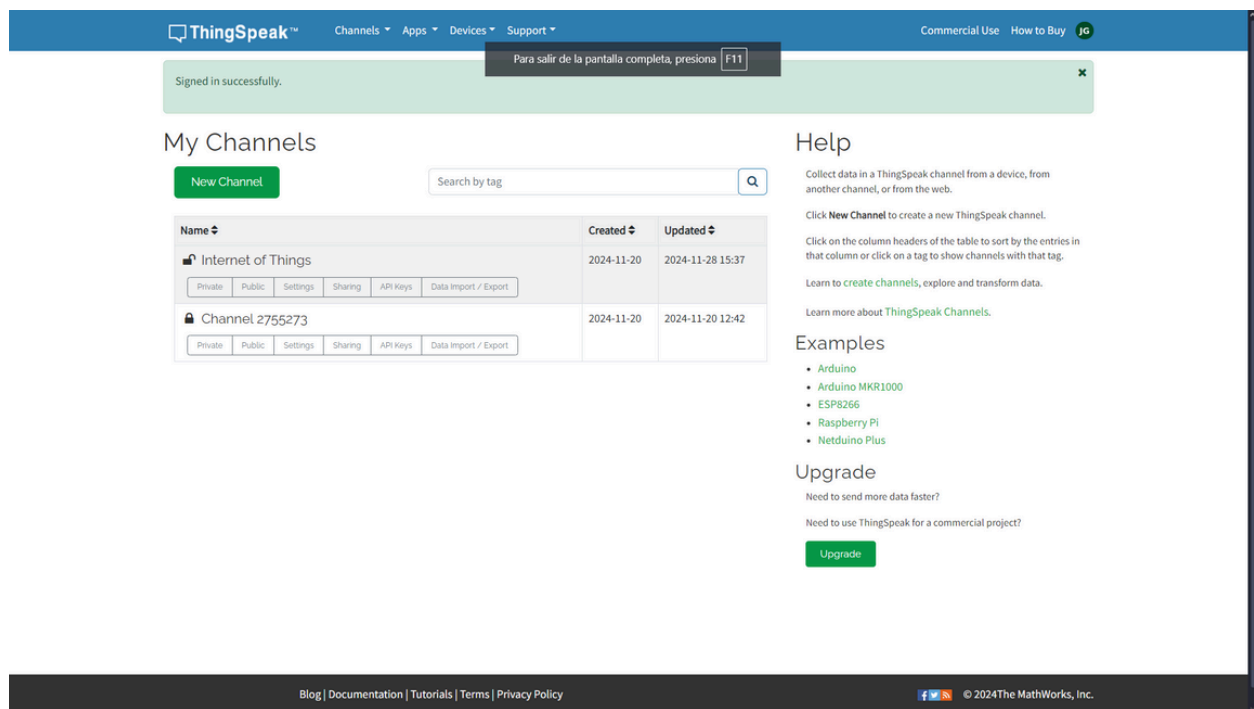
ThingSpeak es una plataforma web que permite almacenar datos en un servidor en la nube y su visualización desde cualquier dispositivo, para utilizarla es necesario que primero te registres de la siguiente manera.



Al entrar a la pagina principal de ThinSpeak debes clickear el botón verde “Get Started For Free”, donde se solicitara ingresar un correo asociado a una cuenta de MathWorks, si no tiene una puede crearla clickeando el apartado de “Create one!”.



Una vez ya tengas la cuenta de MathWorks accederás al siguiente menú, donde se puede visualizar los canales de transmisión de datos que hayas creado anteriormente. En este caso al ser tu primer canal deberás tocar el botón “New Channel”.



Al crear un nuevo canal se pide rellenar los siguientes campos, en este caso solo es necesario rellenar los campos mostrados a continuación:

Name

Description

Field 1 ☒

Field 2 ☒

Field 3 ☒

Field 4 ☐

Field 5 ☐

Field 6 ☐

Field 7 ☐

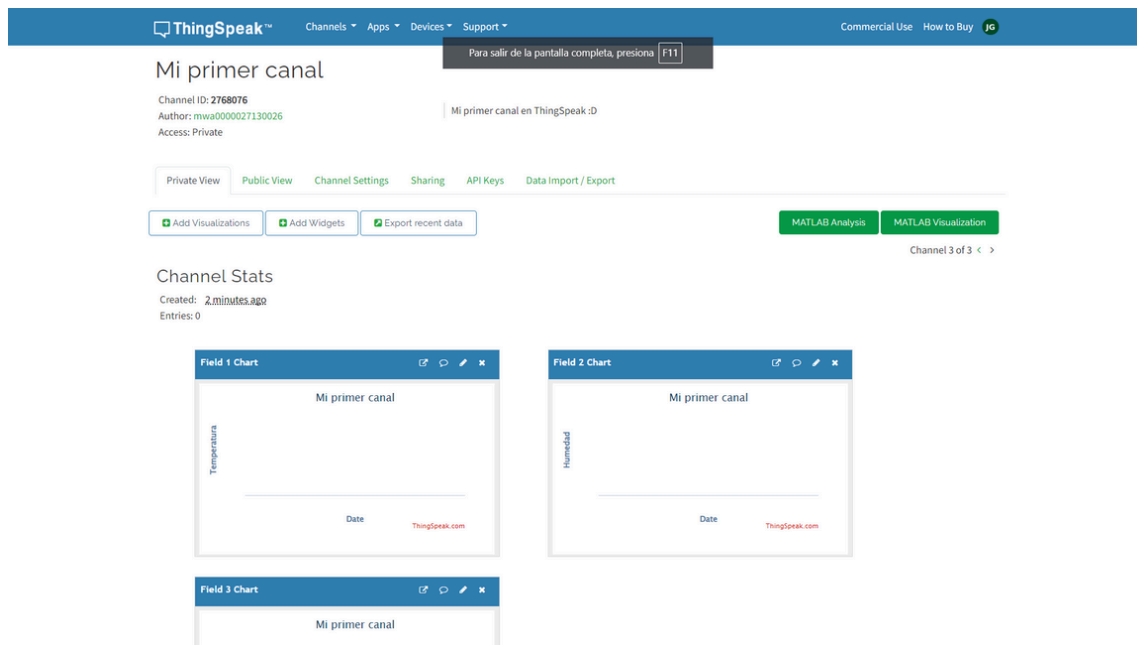
Field 8 ☐

Metadata

Tags

(Tags are comma separated)

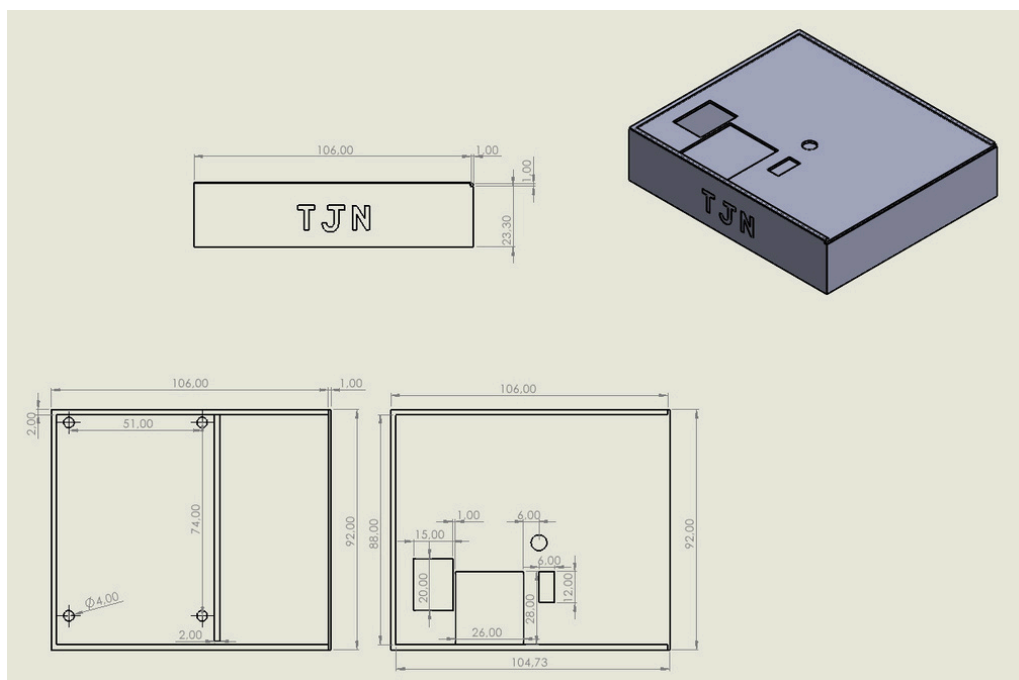
Al guardar el canal debe aparecer lo siguiente, en donde se aprecian las graficas de cada dato, el numero de identificación del canal o Channel ID y e apartado de API Keys, el cual será necesario para el código mas adelante.



Esa API key es secreta y es la llave que permite a elementos externos, como el ESP conectarse y enviar datos al canal, es necesario que no se comparta esta llave para evitar el envío o recolección de información por parte de intrusos no autorizados.

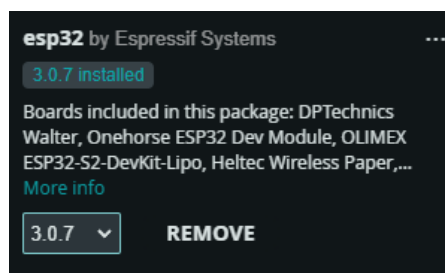
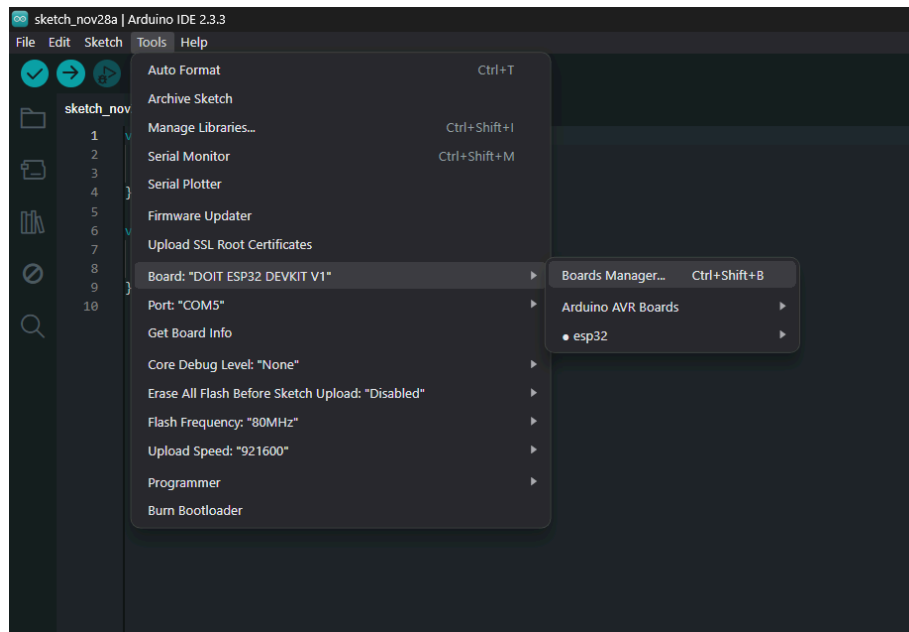
Paso 4. Impresión de la carcaza

Dentro de la misma carpeta descomprimida se encuentra el archivo de Stl, con el cual puedes realizar la impresión de la carcaza en una impresora 3D.

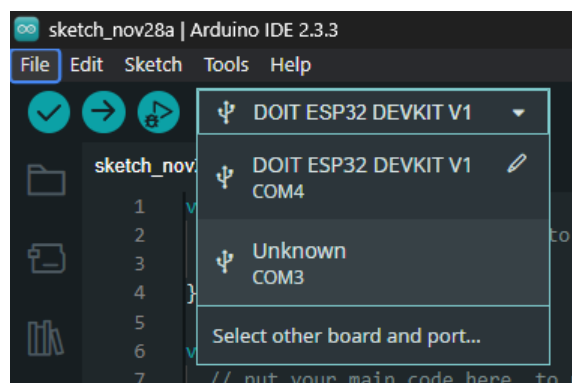


Paso 5. Explicación del código

Dentro de la carpeta se encuentra un archivo .ino, en el esta el código de ArduinoIDE que será necesario cargar al ESP32 para el funcionamiento del entrenador. El primer paso para subirlo es instalar el paquete de placas esp32 de Espressif Systems.

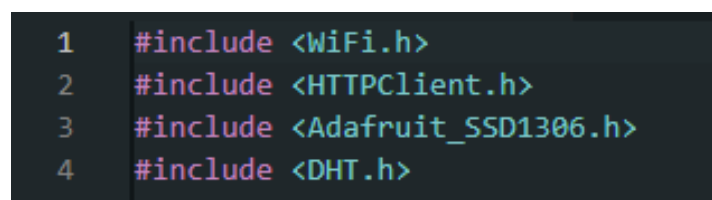


Luego de descargar el paquete de placas es necesario seleccionar la placa y el puerto en el cual esta conectado el ESP 32.



Ahora se explicara paso a paso cada parte del código:

- Primero se incluyen las librerías a utilizar, éstas se pueden descargar desde Github o desde el apartado de librerías de Arduino.



- Se determina el pin de datos del sensor DHT11. Además de crear un objeto dem tipo DHT, esto se utilizará más adelante.

```
6   #define PIN_DHT 32
7   #define TIPO_DHT DHT11
8   DHT dht(PIN_DHT, TIPO_DHT);
9
```

- Se definen los parámetros de la pantalla OLED y el pin de entrada de los datos analógicos de la foto resistencia.

```
10  #define ANCHO_PANTALLA 128
11  #define ALTO_PANTALLA 64
12  Adafruit_SSD1306 pantalla(ANCHO_PANTALLA, ALTO_PANTALLA, &Wire, -1);
13
14  #define PIN_ANALOGICO 15
```

- Se coloca el nombre o ssid de la red Wifi a la cual se va a conectar el Esp32 y su respectiva contraseña, además se incluye el link del servidor de Thingspeak y la clave Api del canal al cual se van a enviar los datos.

```
16  const char* ssid = "Tatiana";
17  const char* contrasena = "12345678";
18
19  const char* servidor = "http://api.thingspeak.com/update";
20  const char* claveApi = "8K20X60F251TR04C";
21
```

- Dentro del void setup() se inicia la comunicación serial en 115200 baudios de velocidad, se inicializa el método begin() del objeto DHT, se limpian los datos en pantalla de la Oled y se llama a la función wifi.begin() para conectar el Esp32 a internet.

```
22  void setup() {
23      Serial.begin(115200);
24      dht.begin();
25      if (!pantalla.begin(0x3C)) {
26          for (;;);
27      }
28      pantalla.clearDisplay();
29      pantalla.display();
30      WiFi.begin(ssid, contrasena);
31      while (WiFi.status() != WL_CONNECTED) {
32          delay(1000);
33      }
34  }
```

- Se leen los valores de los sensores, en este caso humedad y temperatura del DHT y el valor analógico de la foto resistencia. Este último valor lo daremos como porcentaje de luminosidad.

```
37  float temperatura = dht.readTemperature();
38  float humedad = dht.readHumidity();
39  int valorAnalogico = analogRead(PIN_ANALOGICO);
40  float luz = (valorAnalogico / 4095.0) * 100.0;
41
```

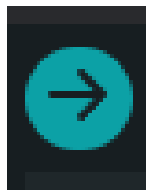

- Enviamos los datos a la pantalla OLED para mostrarlos de manera ordenada.

```
42 pantalla.clearDisplay();
43 pantalla.setTextSize(1);
44 pantalla.setTextColor(SSD1306_WHITE);
45 pantalla.setCursor(0, 0);
46 pantalla.print("Temp: ");
47 pantalla.print(temperatura);
48 pantalla.println(" C");
49 pantalla.print("Hum: ");
50 pantalla.print(humedad);
51 pantalla.println(" %");
52 pantalla.print("Luz: ");
53 pantalla.print(luz, 1);
54 pantalla.println(" %");
55 pantalla.display();
```

- Y utilizamos esta función para enviar los datos por medio del protocolo de comunicación Http.

```
57 if (WiFi.status() == WL_CONNECTED) {
58     HTTPClient http;
59     String url = String(servidor) + "?api_key=" + claveApi +
60     "&field1=" + String(temperatura) +
61     "&field2=" + String(humedad) +
62     "&field3=" + String(luz, 1);
63     http.begin(url);
64     http.GET();
65     http.end();
66 }
67
68 delay(15000);
69 }
```

Por ultimo pulsas el botón de la flecha a la derecha "Upload".



Paso 6. Arma el prototipo

Ubica los componentes del prototipo con el código adentro y añade una batería de 9v para poder alimentar todo el sistema.

