

# Ejemplos:

## Estación Metereológica:

[https://github.com/NewbieMakerLearning/Servidor\\_Estacion\\_Meteo](https://github.com/NewbieMakerLearning/Servidor_Estacion_Meteo)

## Bateria bien montada

<https://emariete.com/medidor-co2-con-bateria-bien-hecho/>

## ESP32



## ARDUINO IDE

Nos descargamos la última versión de Arduino IDE, tiene que ser por encima de la 1.6

Descargamos un plugin para poder conectar con esp32:

<https://github.com/espressif/arduino-esp32>

[https://github.com/espressif/arduino-esp32/blob/master/docs/arduino-ide/boards\\_manager.md](https://github.com/espressif/arduino-esp32/blob/master/docs/arduino-ide/boards_manager.md)

Ponemos esta URL en file > preferences > Additional Boards Manager URLs:

```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json
```

Tarda un ratillo en descargárselas

Ahora vamos a tools > Board: "Arduino Uno" y seleccionamos Boards Manager. Buscamos esp32 y damos a install

En Tools > Board: "Arduino Uno" > ESP32 Arduino

Reiniciamos Arduino. Nos aseguramos que python apunte a python3 instalando python-is-python3

```
sudo apt install python-is-python3
```

Código de ejemplo. Conecta a wifi y nos da la ip. Una vez subido tenemos que reiniciar esp32 (desconectando usb)

```
#include <WiFi.h>           // Include the Wi-Fi library

const char* ssid      = "pitufina";           // The SSID (name) of the Wi-Fi
network you want to connect to
const char* password = "*****";           // The password of the Wi-Fi network

void setup() {
    Serial.begin(115200);           // Start the Serial communication to send
messages to the computer
    delay(10);
    Serial.println('\n');
    WiFi.begin(ssid, password);           // Connect to the network
    Serial.print("Connecting to ");
    Serial.print(ssid);

    while (WiFi.status() != WL_CONNECTED) { // Wait for the Wi-Fi to connect
        delay(500);
        Serial.print('.');
    }

    Serial.println('\n');
    Serial.println("Connection established!");
    Serial.print("IP address:\t");
    Serial.println(WiFi.localIP());           // Send the IP address of the
ESP8266 to the computer
}

void loop() {
}
```

La salida que nos da es esta:

```
.

Connection established!
IP address: 192.168.1.97
```

## HAcer petición GET y POST

```
#include <WiFi.h>           // Include the Wi-Fi library
#include <HTTPClient.h>

const char* ssid      = "mi_red";           // The SSID (name) of the Wi-Fi
network you want to connect to
const char* password = "*****";          // The password of the Wi-Fi network

//Your Domain name with URL path or IP address with path
String serverName = "http://192.168.1.200/esp32";

// the following variables are unsigned longs because the time, measured in
// milliseconds, will quickly become a bigger number than can be stored in
// an int.
unsigned long lastTime = 0;
// Timer set to 10 minutes (600000)
//unsigned long timerDelay = 600000;
// Set timer to 5 seconds (5000)
unsigned long timerDelay = 5000;

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

  WiFi.begin(ssid, password);
  Serial.println("Connecting");
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to WiFi network with IP Address: ");
  Serial.println(WiFi.localIP());

  Serial.println("Timer set to 5 seconds (timerDelay variable), it will take
5 seconds before publishing the first reading.");
}

void loop() {
  //Send an HTTP POST request every 10 minutes
  if ((millis() - lastTime) > timerDelay) {
    //Check WiFi connection status
    if(WiFi.status()== WL_CONNECTED){
      HTTPClient http;

      String serverPath = serverName + "?temperature=24.37";
      // Your Domain name with URL path or IP address with path
      http.begin(serverPath.c_str());
      // Send HTTP GET request
```

```
int httpResponseCode = http.GET();
if (httpResponseCode>0) {
    Serial.print("HTTP Response code: ");
    Serial.println(httpResponseCode);
    String payload = http.getString();
    Serial.println(payload);
}
else {
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
}
// Free resources
http.end();
}
else {
    Serial.println("WiFi Disconnected");
}
lastTime = millis();
}
}
```

## Python

<https://www.youtube.com/watch?v=fs10aqFTj8I>

```
apt-get install esptool
```

Descargamos el firmware:

<https://micropython.org/download/esp32/>

Descargo este:

```
GENERIC : esp32-idf3-20210202-v1.14.bin
```

Para borrar la memoria. Si no sabemos el puerto, no lo ponemos y lo detecta, es /dev/ttyUSB0:

```
esptool erase_flash
```

```
esptool.py v2.8
Found 1 serial ports
Serial port /dev/ttyUSB0
Connecting....._
Detecting chip type... ESP32
Chip is ESP32D0WDQ6 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
```

```
MAC: f0:08:d1:d3:1e:98
Enabling default SPI flash mode...
Erasing flash (this may take a while)...
```

A fatal error occurred: ESP32 ROM does not support function erase\_flash.

Da error. Subimos el firmware que hemos descargado. Tarda un poco. Debería devolver un prompt pero no hace nada:

```
esptool write_flash 0x1000 esp32-idf3-20210202-v1.14.bin
```

```
esptool.py v2.8
Found 1 serial ports
Serial port /dev/ttyUSB0
Connecting.....
Detecting chip type... ESP32
Chip is ESP32D0WDQ6 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: f0:08:d1:d3:1e:98
Enabling default SPI flash mode...
Configuring flash size...
Auto-detected Flash size: 4MB
Erasing flash...
Took 2.39s to erase flash block
Wrote 1445888 bytes at 0x00001000 in 140.5 seconds (82.3 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

## Arduino IDE

<https://github.com/espressif/arduino-esp32>

[https://github.com/espressif/arduino-esp32/blob/master/docs/arduino-ide/boards\\_manager.md](https://github.com/espressif/arduino-esp32/blob/master/docs/arduino-ide/boards_manager.md)

Ponemos esta URL en file > preferences > Additional Boards Manager URLs:

```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json
```

Ahora vamos a tools > Board: "Arduino Uno" y seleccionamos Boards Manager. Buscamos esp32 y damos a install

En Tools > Board: "Arduino Uno" > ESP32 Arduino

# Energia y consumo

<https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/>

# Pantalla e-paper

<https://www.youtube.com/watch?v=1xQqc6ZCXdw>

From:

<http://wiki.legido.com/> - **Legido Wiki**

Permanent link:

<http://wiki.legido.com/doku.php?id=informatica:arduino:esp32>



Last update: **2022/10/24 22:12**