
Documentación técnica de programación

Introducción

Esta guía ayudará a los usuarios a configurar todo lo necesario para la completa utilización del sistema. Se divide en dos apartados, configuración para el nodo central y configuración para el resto de nodos.

Estructura de directorios

Enlace a la descarga de material (Github) <https://github.com/turbolargoo/Free-Connect>

Estructura general de los materiales disponibles en la plataforma:

- **Ejemplos de la comunidad:** Aquí se guardarán los proyectos enviados a través del foro.
- **Ejemplos proporcionados:** Código de los ejemplos utilizados en los manuales.
 - **detector_presencia:** Código para el ejemplo *Detector de presencia*.
 - **planta:** Primera parte del código para el ejemplo *Centinela para plantas*.
 - **planta_eventos:** Segunda parte del código para el ejemplo *Centinela para plantas*.
 - **planta_telegram:** Tercera parte del código para el ejemplo *Centinela para plantas*.

- **termostato:** Código para el ejemplo *Termostato*.
- **Manuales:** Versión de los manuales en PDF.
- **Software útil:** Programas complementarios.
 - **oled_test:** Ayuda para configurar la pantalla integrada de algunos módulos.
 - **scannerI2c:** Programa para averiguar la dirección I2C de los dispositivos conectados (obtenido en Arduino forum).
 - **soporte oled:** Funciones para el manejo de pantallas.
 - **vanilla:** Código base sin ninguna modificación funcional.
- **librerías:** Recopilación de librerías utilizadas.

Nodo Central

Es el encargado de gestionar las comunicaciones y proporciona la interfaz. Su arquitectura es la propia de un servidor. Puede ser instalado en una máquina Linux cualquiera o sobre una Raspberry Pi dedicada.

Instalación y ejecución del proyecto

- **Instalación en una raspberry pi dedicada.** Esta es la opción mas sencilla y la recomendada para la mayoría de los usuarios. La raspberry pi quedará dedicada en exclusiva y los requisitos de recursos son bajos. Escogemos el modelo Zero versión w (wifi integrado) por este motivo.

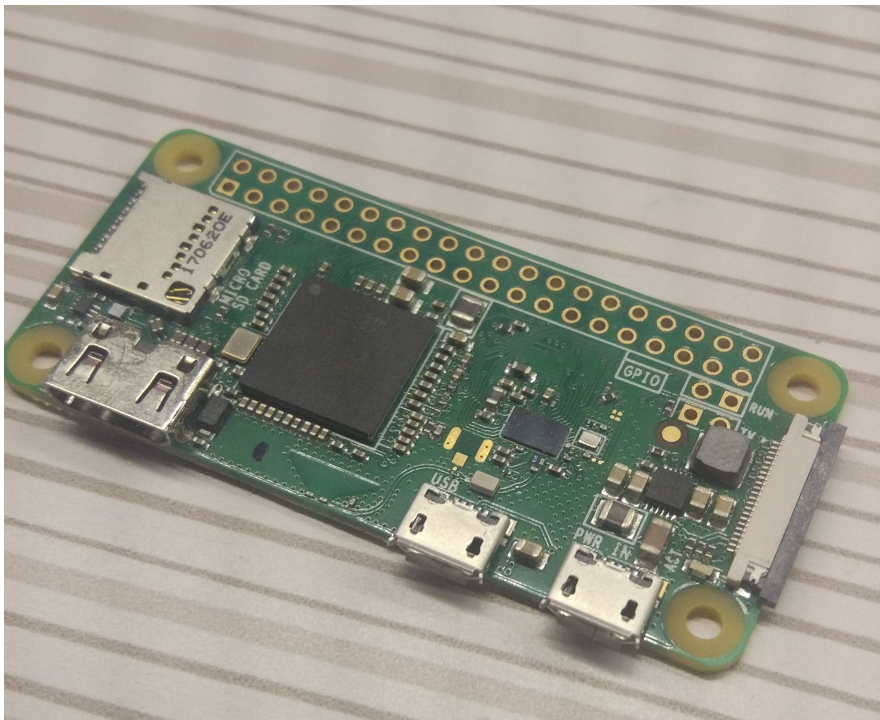


Figura 1: Raspberry pi zero w. Vista superior

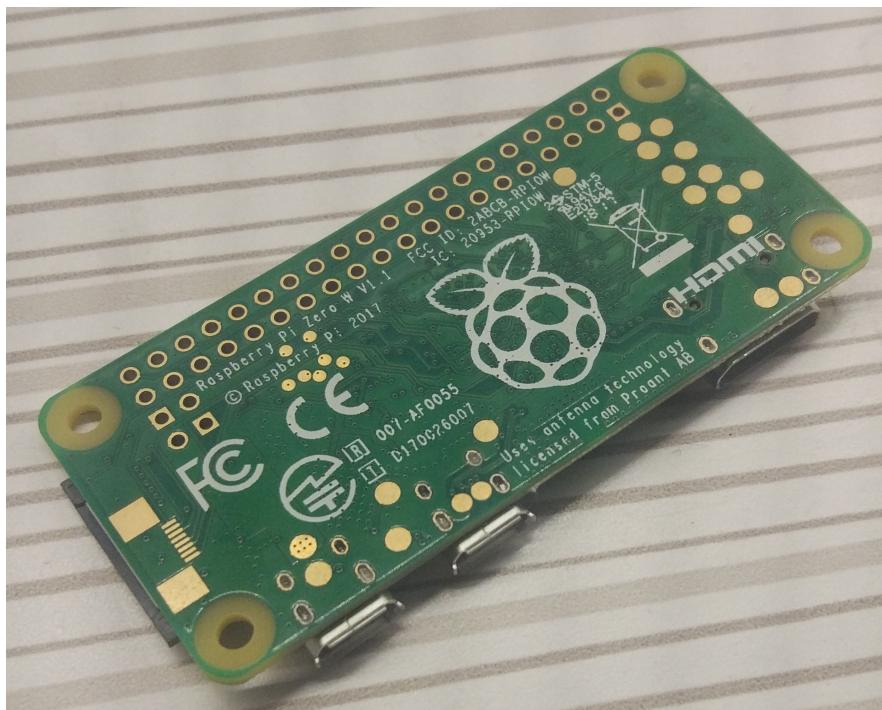


Figura 2: Raspberry pi zero w. Vista inferior



Figura 3: Raspberry pi zero w. Dentro de carcasa impresa en 3D

- **Paso 1: Descarga de contenido**

Descargamos la imagen del sistema desde la web oficial: <https://iot.mozilla.org/gateway/>

La versión utilizada en todos los ejemplos es la más reciente en Marzo de 2020. Versión 0.12

- **Paso 2: Grabar tarjeta SD**

Descomprimos el archivo y lo grabamos en la tarjeta SD utilizando un programa como *rufus* [?].

- **Paso 3: Conexión con el nodo**

Esperamos unos minutos a que el módulo se inicie y escaneamos con nuestro ordenador en busca de redes wifi. El nodo habrá creado su propia red. Nos conectamos a ella.

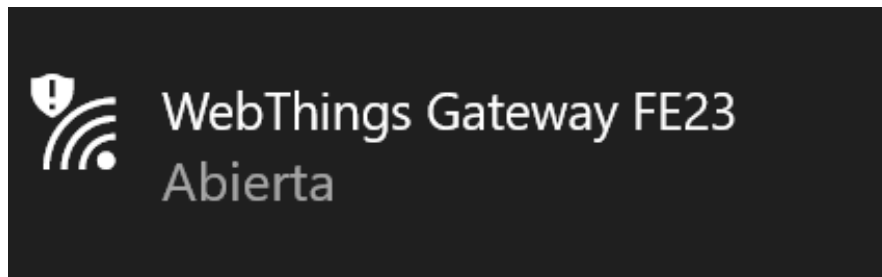


Figura 4: Red wifi creada por el modulo central

- **Paso 4: Acceso a la interfaz**

Navegamos a la ip de la puerta de enlace (192.168.1.1)

Seleccionamos nuestra red wifi e introducimos las credenciales.



Figura 5: Red wifi creada por el modulo central

Una vez se haya establecido la conexión, volvemos a conectar nuestro ordenador a la red wifi habitual y navegamos hacia <http://gateway>.

- **Paso 5: Configuración del dominio**

Este dominio será nuestra dirección para acceder remotamente al nodo central.



Bienvenido

Elije una dirección web segura para tu puerta de enlace:

subdominio .mozilla-iot.org

Correo electrónico

☒ Mantenedme informado sobre las nuevas funciones y oportunidades de colaboración. [Política de privacidad](#)

Crear

Omitir

Figura 6: Configuración del subdominio

Si ya hemos creado un dominio anteriormente y queremos conectarnos a él en vez de crear uno nuevo, lo que debemos hacer es introducir los datos de ese dominio y pulsar el botón crear. Esto no creará otro dominio sino que enviará un token al correo electrónico usado para poder acceder a nuestro dominio ya existente.

Bienvenido

Elije una dirección web segura para tu puerta de enlace:

! .mozilla-iot.org

c

4

Revisa el correo electrónico para obtener un token de recuperación y pégalo arriba.

Crear

Omitir

Figura 7: Configuración del subdominio

Todo listo para añadir los demás nodos.

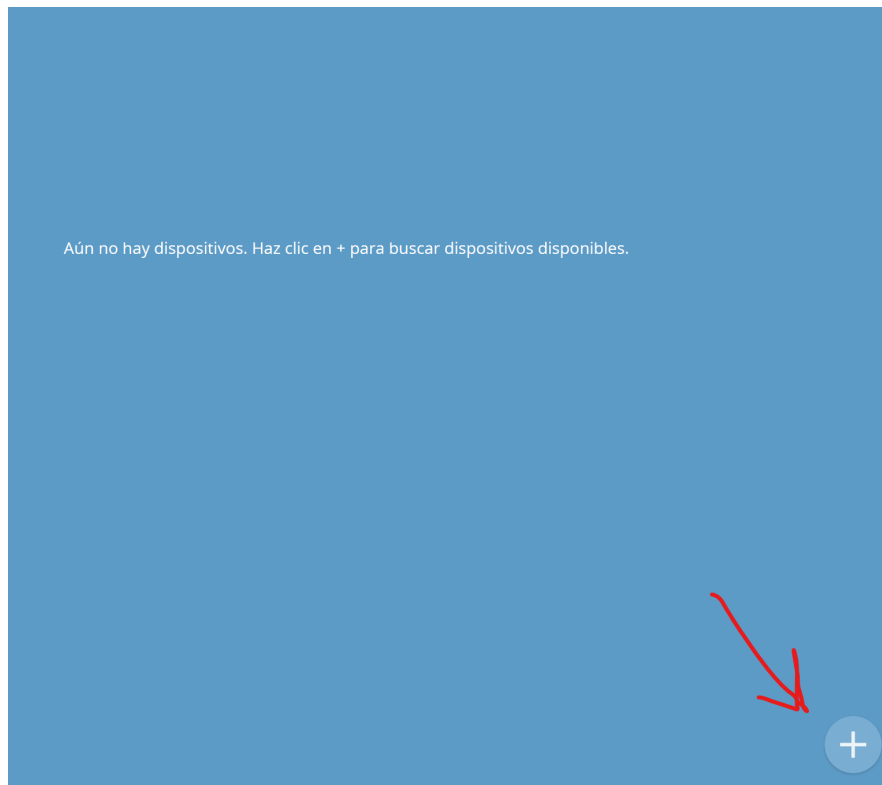


Figura 8: Configuración del nodo central

■ Instalación en un servidor linux.

Es posible instalar el software en cualquier servidor que utilice linux. Si ya tenemos una máquina dedicada, esta opción es muy conveniente.

Estos son los comandos en orden de instalación:

```
sudo apt update
sudo apt install pkg-config
sudo apt install curl
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.35.2/install.sh
— bash
. ~/.bashrc
sudo apt install libusb-1.0-0-dev libudev-dev
sudo apt install autoconf
sudo apt install libpng16-16 o sudo apt install libpng-dev (para arm)
sudo apt install libffi-dev
sudo apt install git
```

```
sudo apt install build-essential
sudo apt install python3-pip
sudo pip3 install six
sudo python3 -m pip install git+https://github.com/mozilla-iot/gateway-
addon-python#egg=gateway-addon
git clone https://github.com/mozilla-iot/gateway.git
cd gateway
nvm install
nvm use
nvm alias default $(node -v)
npm ci
```

Una vez haya finalizado la instalación pararemos cualquier proceso que pueda interferir en la instalación (servidores apache o similares) y procedemos de la misma manera que en la instalación desde imagen.

Otros nodos

A continuación encontramos una guía sobre la configuración necesaria para la creación de los demás nodos.

Requisitos previos

Es necesario tener instalado el entorno de desarrollo Arduino IDE. La versión utilizada es la 1.8.13 <https://www.arduino.cc/en/Main/Software>

Primero debemos añadir el paquete de placas ESP32 ya que no se encuentra incluido por defecto. El enlace para el gestor de placas es: https://dl.espressif.com/dl/package_esp32_index.json

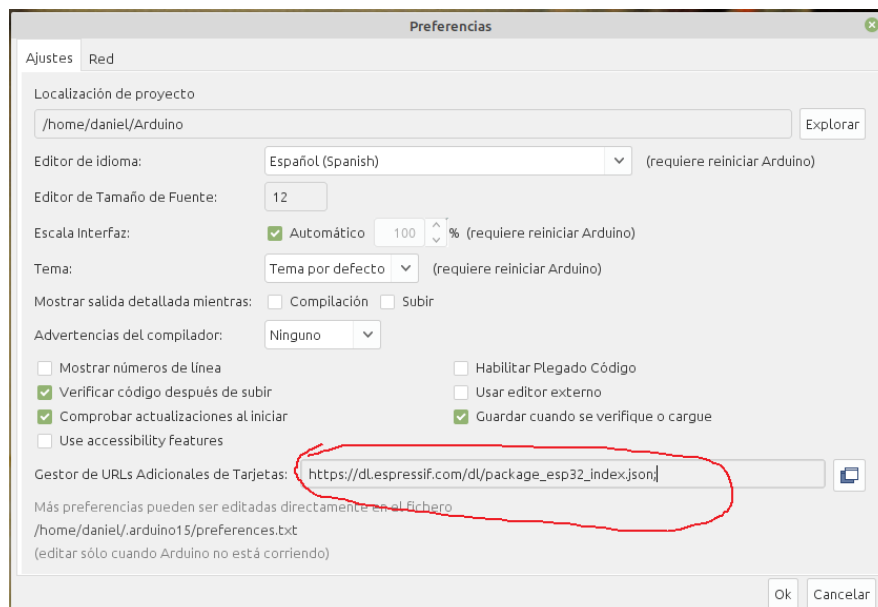


Figura 9: Instalación placas esp32

El siguiente paso es instalar el paquete desde el gestor de placas. La versión utilizada es la 1.0.4 de Espressif Systems

Gestor de tarjetas

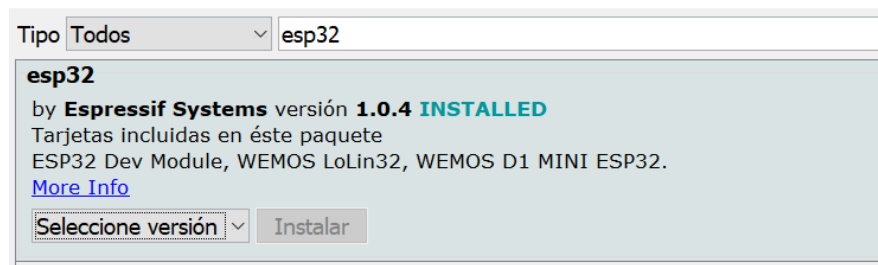


Figura 10: Instalación del paquete de placas esp32

Ahora ya podemos seleccionar la placa:

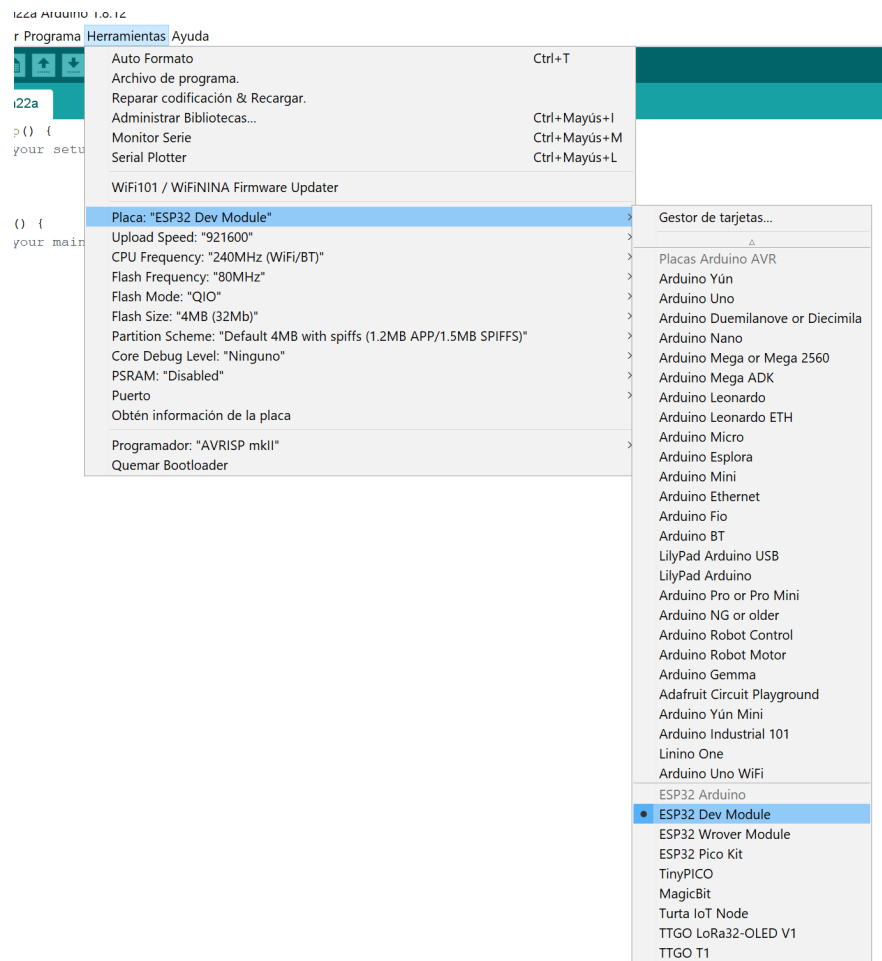


Figura 11: Placa a seleccionar

Librerías necesarias

- WebThingAdapter. Instalable desde el gestor de librerías.

Versión 0.11 de Mozilla Iot

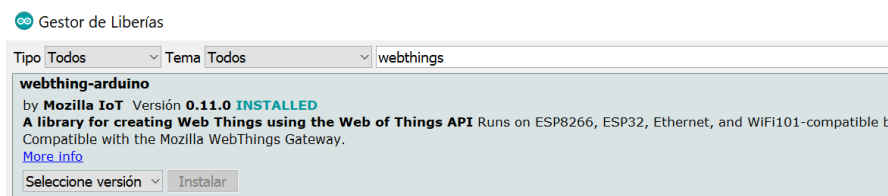


Figura 12: Librería WebThingAdapter

- ArduinoJson. Instalable desde el gestor de librerías.
Versión 6.14.1 de Benoit Blanchon
- ESPAsyncWebServer. Descargamos de <https://github.com/me-no-dev/ESPAsyncWebServer.git>
- AsyncTCP. Descargamos de <https://github.com/me-no-dev/AsyncTCP.git>
- ArduinoOTA. Está instalada por defecto en Arduino IDE
- utlgbotlib Instalable desde el gestor de librerías.
Versión 1.0.0 de JRios

Librerías adicionales

- SSD1306Wire. Pantalla oled. Descargamos de <https://github.com/ThingPulse/esp8266-oled-ssd1306.git>
- DHTesp. Sensor DHT11. Disponible desde el gestor de librerías.
Versión 1.17.0 de beegee_tokyo

Compilación, instalación y ejecución del proyecto

Una vez estén satisfechos los requisitos expuestos anteriormente, podremos compilar el código de cualquier nodo. Para ello utilizaremos los botones correspondientes del entorno de desarrollo.

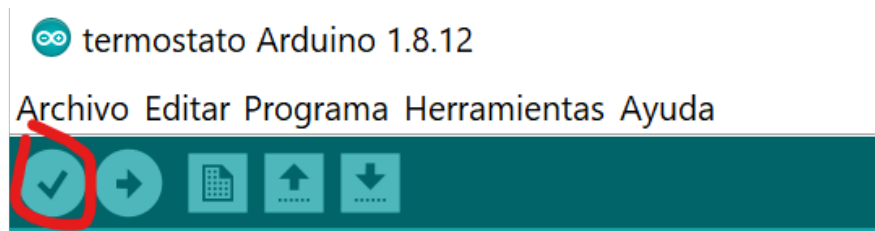


Figura 13: Entorno de desarrollo