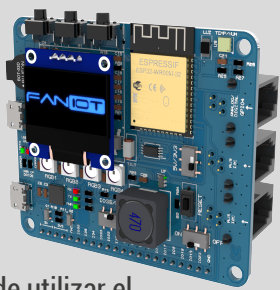


Guía NEOFAN IoT



Lea atentamente esta guía antes de utilizar el producto y guárdela para futuras consultas.

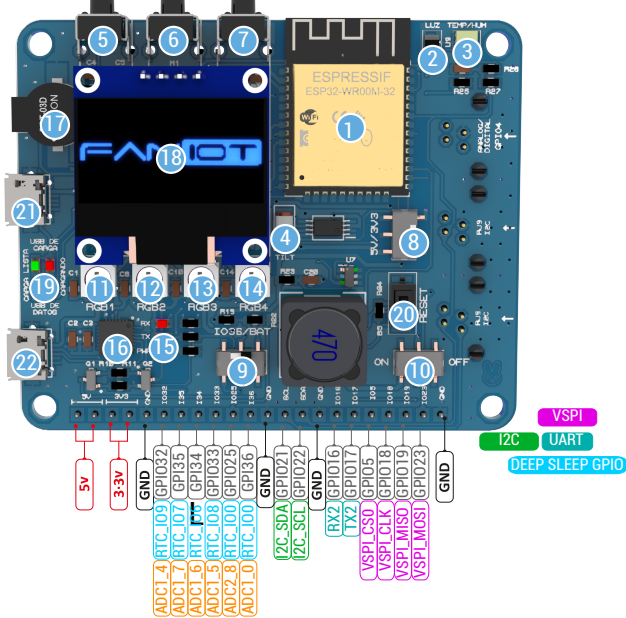
A

La **NEOFAN IoT** posee un microcontrolador con conexión WiFi y Bluetooth, 8 entradas/salidas digitales, 4 de ellas pueden ser utilizadas como entradas analógicas.

Especificaciones Técnicas:

Microcontrolador	ESP32
Reloj	240 MHz
Conectividad	WiFi (802.11 b/g/n) Bluetooth v 4.2
Alcance de Wifi	30 m interiores - 90 m exteriores
Voltaje de Operación	3.3 v
Voltaje de Alimentación	5 v
Pines E/S DIGITALES	8
Entradas Analógicas	4
Conectores RJ-9	3

Peso: 55 gr Alto: 60 mm Ancho: 90mm



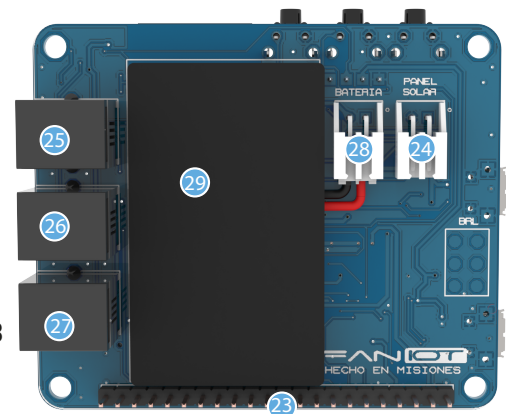
4

- 1 ESP32: Microcontrolador, el cerebro de la placa NEOFAN
- 2 Sensor de Luminosidad TMT6000: GPIO39
- 3 Sensor de Temperatura/Humedad HTU21d: Protocolo I2C
- 4 Sensor de Vibración TILT: GPIO14
- 5 6 7 Botones Programabl-es: GPIO0,GPIO15,GPIO13
- 8 Selección del voltaje para el conector RJ9: V5/ 3V3
- 9 Swich para medir nivel de batería
- 10 ON/OFF switches (Batería Conectada)
- 11 12 13 14 Neopixels RGB Programables: GPIO27
- 15 Programación de Indicadores LED (Rx/Tx) y encendido (PWR).
- 16 Convertidor de USB a TTL CP2102
- 17 Buzzer pasivo programable: GPIO12
- 18 Pantalla OLED 128x64 pixels: Protocolo I2C
- 19 Indicadores de carga de batería
- 20 Botón Reset
- 21 Entrada MicroUSB de carga
- 22 Entrada MicroUSB de programación

A.1 Diagrama de Pines

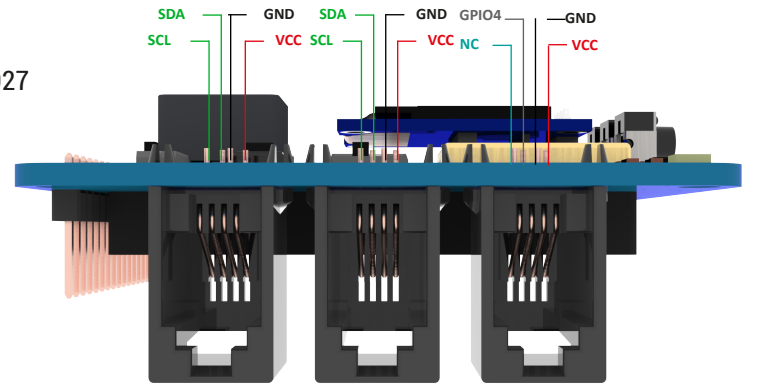
Cada Pin posee un número identificador para utilizarlo dentro del código. Se utilizan las letras para indicar si es una salida digital (D) o analógica (A) respectivamente.

2



- 23 Pines
- 24 Conector JST para panel solar.
- 25 GPIO4 Analog/Digi.
- 26 I2C.
- 27 I2C.
- 28 Conector JST para batería
- 29 Batería Li-Po 3.7V 700mAh

3



A.2 Entornos de Programación

Un Entorno de Desarrollo Integrado (IDE), es un programa informático para la carga de las instrucciones compiladas, es decir traducidas a un lenguaje máquina, en la memoria del módulo.

A.2.1 Descarga, Instalación y Configuración del Código IDE

- Descarga desde: <https://www.arduino.cc/en/software>
- Ejecute el instalador del programa y siga los pasos de instalación
- Finalizado la instalación, debe inicializarlo. Vaya a la pestaña: Archivo - Preferencias.
- En el cuadro de "Gestor de URLs Adicionales de Tarjetas" escriba: https://dl.espressif.com/dl/package_esp32_index.json y de OK.



5

- Cargue el código al Módulo. Para esto debe hacer clic en el botón Subir y esperar unos pocos segundos a que el código se cargue.
- Cuando el código se cargue totalmente se mostrará el mensaje "Subido" en la barra de estado.
- Reinicie la placa presionando el botón Reset y haga clic en el ícono Monitor Serie.
- Se abrirá una ventana con el siguiente mensaje:



A.2.3 Descarga, Instalación y Configuración del Entorno de Programación en Bloque

El Entorno de Programación en Bloque permite estudiante abordar la programación de manera práctica y visual en la que integra conceptos de robótica educativa con IOT.

- Para utilizar el Entorno de Programación en Bloque debe descargarlo desde <https://faniot.com.ar/descarga/FAN-BLOCK.exe>
- Ejecute el instalador del programa y siga los pasos de instalación.
- Una vez instalado debe inicializarlo. Haga click en "Board Manager"



- Busque en Installed la opción de "Generic ESP32 WROOM", haga clic en "Change Board" y confirme.



¡Terminado!

Ya podés comenzar a programar tu Módulo FANIOT

A.2.4 Introducción a la Programación en Bloques

- **Programming Mode:** Cambia el modo de programación de Bloque a Código.
- **Board Manager:** Selecciona el microcontrolador que se desea programar.
- **Package Manager:** Administrador de paquetes.
- **Plugin Manager:** Selecciona y descarga extensiones y para la programación de librerías.

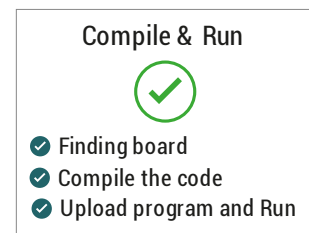
6

- **Example & Tutorials:** Lista ejemplos de código y tutoriales.
- ⚙️ **Setting:** Personaliza visualmente el entorno de programación.
- **Serial Monitor:** Inicia la comunicación entre el Módulo Base y la computadora.
- 📄 **New File:** Crea un nuevo proyecto.
- 📁 **Open File:** Abre un proyecto ya existente.
- 💾 **Save File:** Guarda el proyecto actual.
- ✅ **Just Compile:** Compila el proyecto para la búsqueda de errores.
- ▶️ **Compile and Run:** Compila y carga el proyecto al Módulo Base.
- ⚙️ **Setup Board:** Selecciona el COM y los baudios de comunicación.

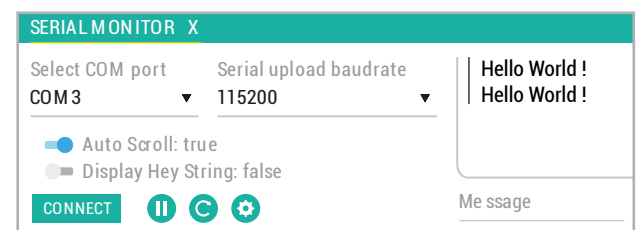
- Para realizar su primer programa debe crear los bloques igual a la siguiente imagen.



- Cargue el bloque a la NeoFan: para esto debe hacer clic en el botón Compile and Run y esperar unos pocos segundos a que el bloque se cargue.
- Cuando se cargue totalmente se mostrará el siguiente mensaje:



- Reinicie la placa presionando el botón RESET y haga clic en el ícono Serial Monitor. Aparecerá el siguiente cuadro:

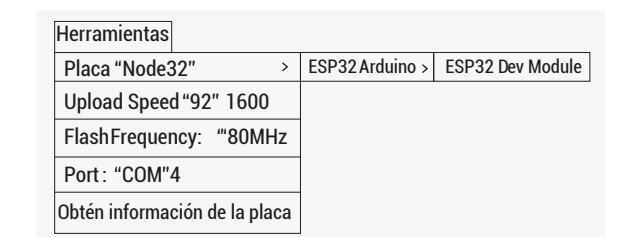


Gestor de URLs Adicionales de Tarjetas https://dl.espressif.com/dl/package_esp32_index.json

- Vaya a la Pestaña: Herramientas - Placa - Gestor de tarjetas.
- En el cuadro de Gestor de tarjetas escriba "esp32" by Espressif Systems y haga clic en Instalar.



- Vaya a la Pestaña Herramientas - Placa - ESP32 Arduino 1.0.6 y elija el Módulo ESP32 Dev Module. También elija el Puerto COM correspondiente.



A.2.2 Introducción al Código IDE

El Código IDE se compone de varias partes:

- 1) Verificar
- 2) Cargar
- 3) Nuevo
- 4) Abrir
- 5) Guardar
- 6) Monitor Serial
- 7) Editor
- 8) Notificaciones



- Para realizar su primer programa debe tipear el siguiente código.

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

void loop(){
  Serial.println("Hola Mundo!");
  delay(5000);
}
```

Este programa enviará un mensaje "Hola mundo" a través de comunicación Serial cada 5 segundos.

- 1 Instalar Python (última versión) from <https://www.python.org/downloads/>



- 2 Abra cmd y ejecute: (en caso de que uno no funcione, intente lo siguiente)

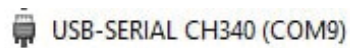
```
pip install esptool
```

```
python -m pip install esptool
```

```
pip2 install esptool
```

- 3 La memoria flash se borra:

```
esptool.py --chip esp32 --port COMX erase_flash  
(Replace the X with the COM port number)
```



- 4 Descargue el último firmware de micropython para ESP32 desde: <https://micropython.org/download/esp32/>

- 5 Copie el comando para cargar el firmware, cambiando el puerto com de linux a COMX

```
From then on program the firmware starting at address 0x1000:  
esptool.py --chip esp32 --port /dev/ttyUSB0 --baud 400000 write_flash -z 0x1000 esp32-20190125-v1.10.bin
```

El firmware en este caso se guarda en la carpeta de descarga.

- 6 Abrir una terminal en descargas

- 7 Copie el nombre del firmware descargado

```
esp32-20220618-v1.19.1.bin
```

- 8 Coloque el comando copiado, colocando el puerto COM correspondiente y cambiando el nombre del firmware descargado.

```
esptool.py --chip esp32 --port COMX --baud 460800 write_flash -z 0x1000 esp32-20220618-v1.19.1.bin
```

```
Windows PowerShell  
Copyright (c) Microsoft Corporation. All rights reserved.  
Try the new cross-platform PowerShell https://aka.ms/powershell  
PS D:\Backup\Usuario\Downloads> esptool.py --chip esp32 --port COM9 --baud 400000 write_flash -z 0x1000 esp32-20220618-v1.19.1.bin
```

Luego, ejecute.

C Programar con MicroPython

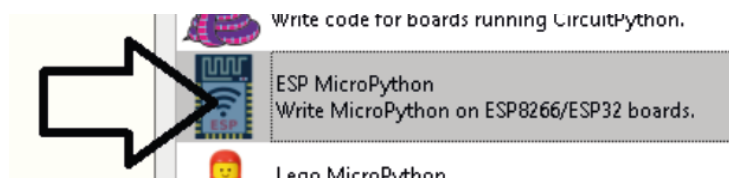
- 1 Ir a la siguiente url: codewith.mu

- 2 Vaya a Descargar y descargue la versión para Windows.

- 3 Abra el instalador, acepte los términos y presione instalar.

- 4 Abra el programa Mu, tardará un tiempo en ejecutarse.

- 5 Seleccionar para programar con ESP MicroPython



- 6 Una vez abierto el programa, vaya a la opción de carga.



- 7 Abra el archivo python con el programa

- 8 Conecte la ESP32

- 9 Haga clic en ejecutar



- 10 Ahora el programa se ejecuta en tiempo real en el ESP32.