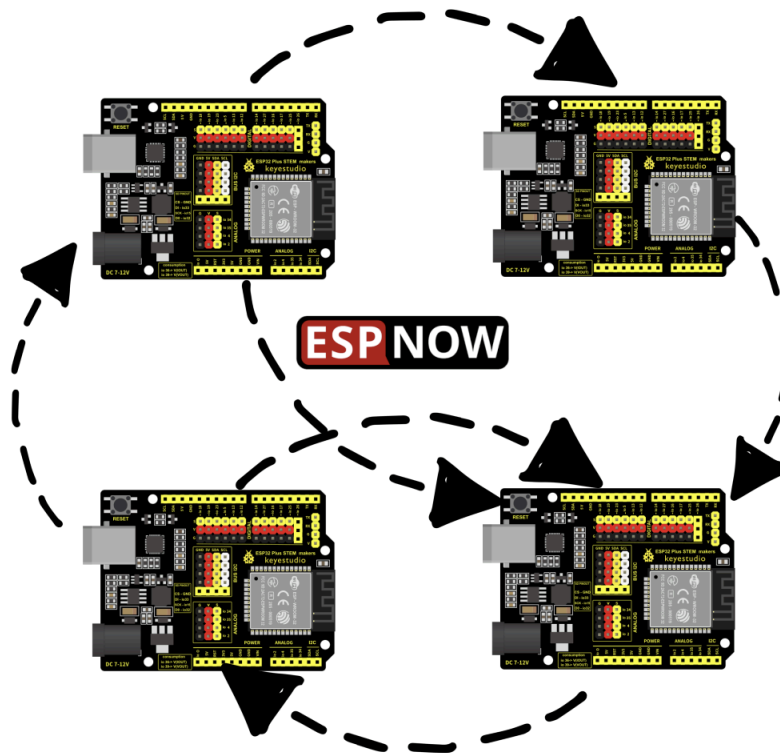




Comunicación ESP-NOW con ESP32 STEAMakers



Juanjo López

22-07-2024

1. Introducción

ESP-NOW es un protocolo de comunicación inalámbrica desarrollado por Espressif para los microcontroladores ESP32, diseñado para operar con baja potencia y baja latencia. Utiliza la banda de 2.4 GHz para transmitir datos directamente entre dispositivos ESP32, sin necesidad de una red Wi-Fi, lo que lo hace ideal para aplicaciones en tiempo real. Su funcionamiento permite la comunicación directa entre múltiples dispositivos, formando una red ad-hoc, y soporta tanto unicast como multicast, permitiendo la transmisión de datos a uno o varios dispositivos simultáneamente.

En términos de seguridad, ESP-NOW ofrece encriptación de datos mediante claves precompartidas, asegurando la comunicación. Este protocolo está diseñado para ser eficiente en términos de consumo de energía, lo que lo hace adecuado para dispositivos alimentados por baterías. Cada dispositivo ESP32 puede comunicarse con hasta 20 dispositivos pares de manera simultánea, utilizando direcciones MAC para identificar y comunicarse entre sí.

ESP-NOW se utiliza en una variedad de aplicaciones, como la automatización del hogar, redes de sensores, robótica y monitoreo industrial. En la automatización del hogar, permite el control de dispositivos y sensores sin necesidad de un router central. En las redes de sensores, facilita la transmisión de datos entre múltiples nodos de manera eficiente. En robótica, es fundamental para la coordinación entre robots y controladores en tareas colaborativas en tiempo real, mientras que en el monitoreo industrial, permite la comunicación entre dispositivos en entornos industriales para la recopilación y transmisión de datos.

Características del Protocolo ESP-NOW

-Baja Latencia: ESP-NOW ofrece una transmisión de datos casi instantánea, con una latencia extremadamente baja, lo que es vital para aplicaciones en tiempo real.

-Comunicación en Red: Permite la interconexión de múltiples dispositivos, formando una red que mejora la cobertura y la robustez de la comunicación.

-Eficiencia Energética: Diseñado para ser eficiente en términos de consumo de energía, lo cual es crucial para dispositivos alimentados por baterías.

-Facilidad de Implementación: La simplicidad de ESP-NOW permite a los estudiantes concentrarse en el diseño y la funcionalidad de sus robots sin preocuparse por la complejidad de las redes de comunicación.

-Interactividad en Tiempo Real: La baja latencia es esencial para la robótica, donde la respuesta rápida a los comandos es fundamental. Esto permite que los robots reaccionen casi instantáneamente a las órdenes del controlador.

-Proyectos Colaborativos: Con ESP-NOW, múltiples robots pueden comunicarse entre sí de manera eficiente, posibilitando proyectos colaborativos y la implementación de sistemas robóticos más complejos y coordinados.

-Portabilidad y Flexibilidad: Al no depender de una infraestructura Wi-Fi, los proyectos pueden ser implementados en cualquier lugar, aumentando la flexibilidad y el alcance de las actividades educativas.

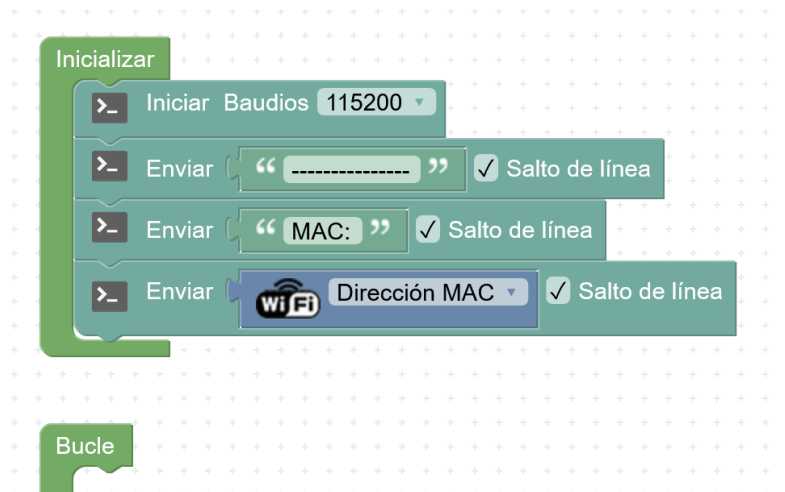
2. Obtener dirección MAC (WiFi) de la placa ESP32 STEAMAKERS

La dirección MAC (Media Access Control) es un identificador único asignado a la interfaz de red de cada dispositivo para su comunicación en una red. En el caso de las placas ESP32, la dirección MAC se utiliza tanto para la conexión Wi-Fi como para la comunicación ESP-NOW.

Formato: La dirección MAC consta de 6 bytes (48 bits), y generalmente se expresa en formato hexadecimal, separado por dos puntos (:) o guiones (-)

2.1 Obtener la dirección MAC por la consola, puedes usar el siguiente programa:

<http://www.arduinoblocks.com/web/project/1889481>



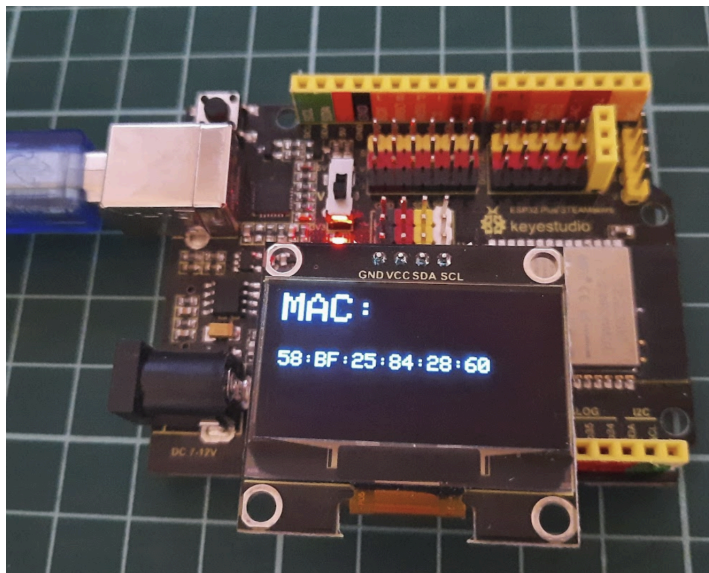
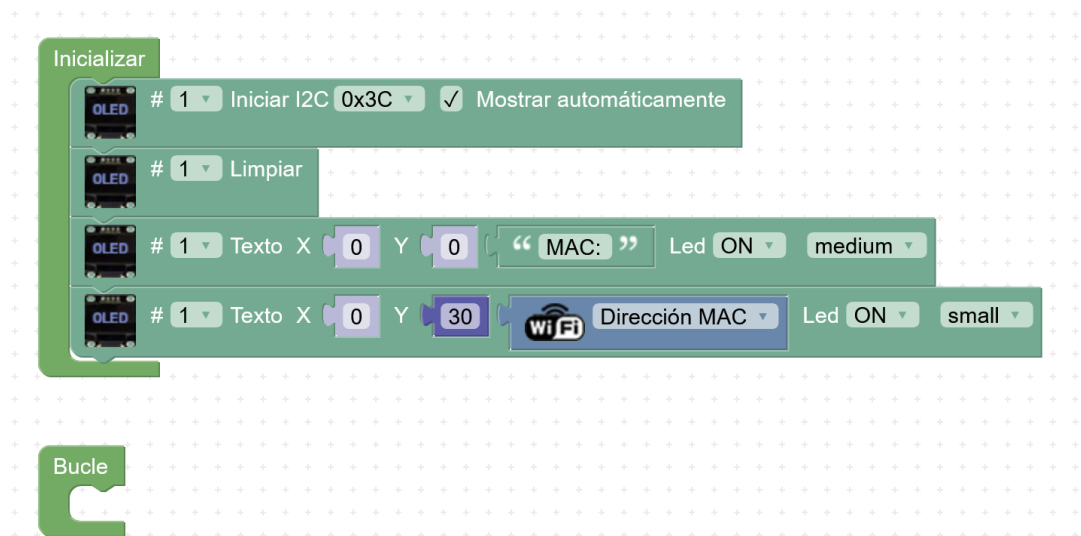
Consola serie:

```
ets Jun 8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:10944
load:0x40080400,len:6388
entry 0x400806b4
```

```
MAC:
58:BF:25:84:28:60
```

2.2 Obtener la dirección MAC y visualizar en pantalla OLED



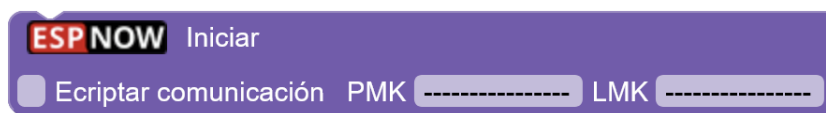
CONSEJO: A la hora de trabajar con proyectos ESP-NOW es recomendable etiquetar físicamente cada placa con la dirección MAC obtenida para poder fácilmente identificar cada una de ellas durante la programación.

3. Bloques ESPNOW en Arduinoblocks

3.1 Iniciar

Inicia el protocolo ESP-NO.

Permite activar la encriptación mediante una clave primaria (PMK) de 16 caracteres y una clave local (LMK) de 16 caracteres. En los dispositivos emparejados en la red ESP-NOW deberán coincidir las claves para poder establecer comunicación.



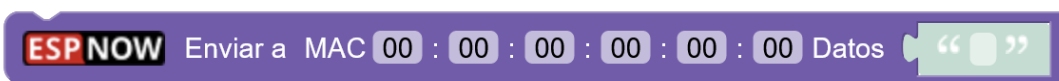
3.2 Añadir nodo remoto (emparejar)

Añade un nuevo nodo a partir de la dirección MAC del dispositivo. Una vez emparejado podremos enviar datos a ese nodo. (según las especificaciones se permiten un máximo 10 nodos aproximadamente o un máximo de 6 en caso de usar encriptación, aunque parece que en la práctica esta limitación es más flexible)



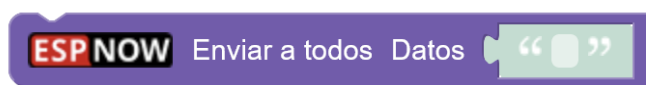
3.3 Enviar a un nodo (previamente emparejado)

Permite enviar datos (en forma de trama de texto) a un nodo remoto. Los datos pueden tener una longitud máxima de 250 caracteres.



3.4 Enviar a todos (los nodos emparejados)

Envía una trama de datos en forma de texto a todos los nodos emparejados. Los datos pueden tener una longitud máxima de 250 caracteres.



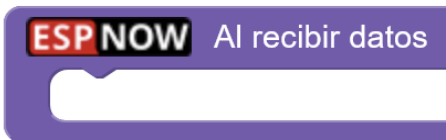
3.5 Comprobación de si el último envío ha sido correcto

Podemos detectar si ha habido acuse de recibo del nodo remoto o no.

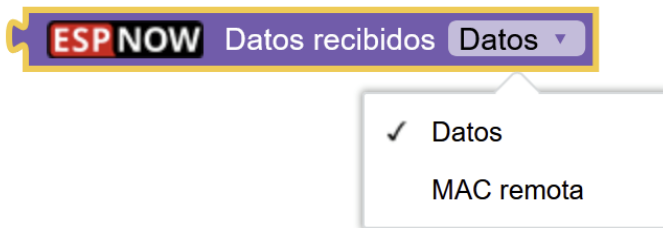


3.6 Al recibir datos

Evento que se produce al recibir nuevos datos desde un nodo remoto



Dentro del evento tendremos los valores válidos para este bloque que nos proporciona tanto los datos recibidos como la dirección MAC del nodo remitente.



:

IMPORTANTE:

El evento "Al recibir datos" se puede bloquear internamente si realizamos esperas bloqueantes en nuestro programa. Por tanto evitar usar el bloque "esperar", y en su lugar utilizar el "ejecutar cada" o otros esquemas de programación similares no bloqueantes.

4. Ejemplos

4.1 Comunicación unidireccional Emisor - Receptor

Una placa envía el valor de una variable "contador" que aumentará cada segundo y se visualizará en la pantalla OLED de la placa receptora.

Placa 1 (emisor): 58:BF:25:84:26:4C

Inicializar

- ESP NOW Iniciar
- Escriptar comunicación PMK _____ LMK _____
- ESP NOW Emparejar nodo remoto MAC 58 : BF : 25 : 84 : 28 : 60
- Establecer cont = 0

Bucle

- ESP NOW Enviar a MAC 58 : BF : 25 : 84 : 28 : 60 Datos cont
- Establecer cont = cont + 1
- Esperar 1000 milisegundos

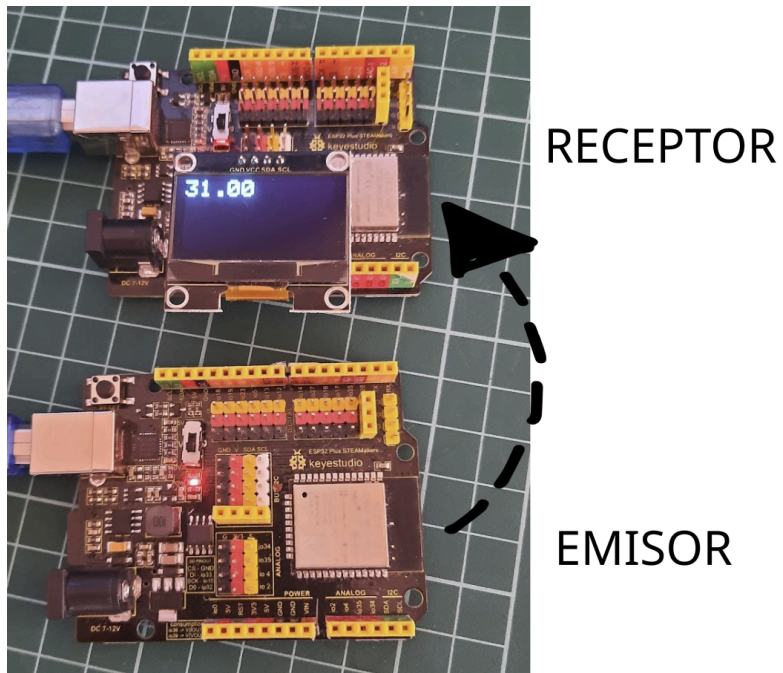
Placa 2 (receptor): 58:BF:25:84:28:60

Inicializar

- OLED # 1 Iniciar I2C 0x3C Mostrar automáticamente
- OLED # 1 Limpiar
- ESP NOW Iniciar
- Escriptar comunicación PMK _____ LMK _____

Bucle

- ESP NOW Al recibir datos
- OLED # 1 Limpiar
- OLED # 1 Texto X 0 Y 0 ESP NOW Datos recibidos Datos Led ON big



IMPORTANTE: Si encriptamos la información, aunque el envío sea en una dirección debemos emparejar también al nodo emisor con el receptor.

Placa 1 (emisor) encriptado:

```

Inicializar
  ESP NOW Iniciar
  Ecriptar comunicación PMK juanjo----- LMK lopez-----
  ESP NOW Emparejar nodo remoto MAC 58 : BF : 25 : 84 : 28 : 60
  Establecer cont = 0
  
```

Placa 2 (receptor) encriptado:

```

Inicializar
  ESP NOW Iniciar
  Ecriptar comunicación PMK juanjo----- LMK lopez-----
  nodo emisor ESP NOW Emparejar nodo remoto MAC 58 : BF : 25 : 84 : 26 : 4C
  # 1 Iniciar I2C 0x3C Mostrar automáticamente
  # 1 Limpiar
  
```


4.2 Comunicación unidireccional Emisor - Receptor + JSON

En la mayoría de ocasiones debemos enviar diferentes datos entre los nodos de la red ESP-NOW, una manera sencilla y efectiva es usando la codificación JSON para encapsular los datos en forma de claves-valor y poder recuperarlos en el receptor de forma sencilla.

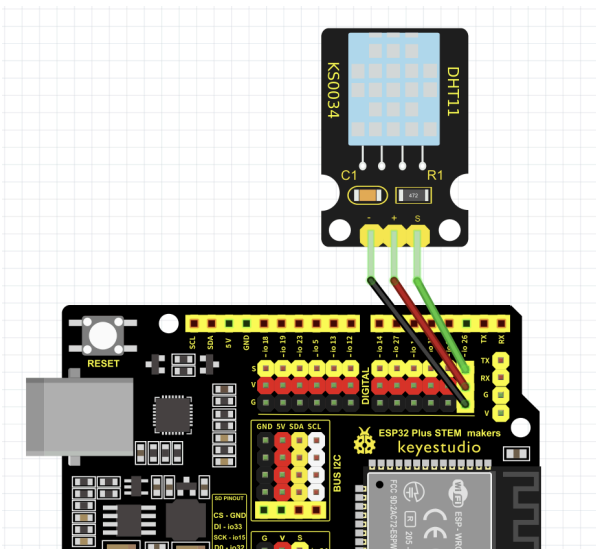
Para más información sobre JSON y arduinoblocks:

<https://drive.google.com/file/d/1r290vEJVZtt8yp4PELnXgTEFWqUUBZk/view>

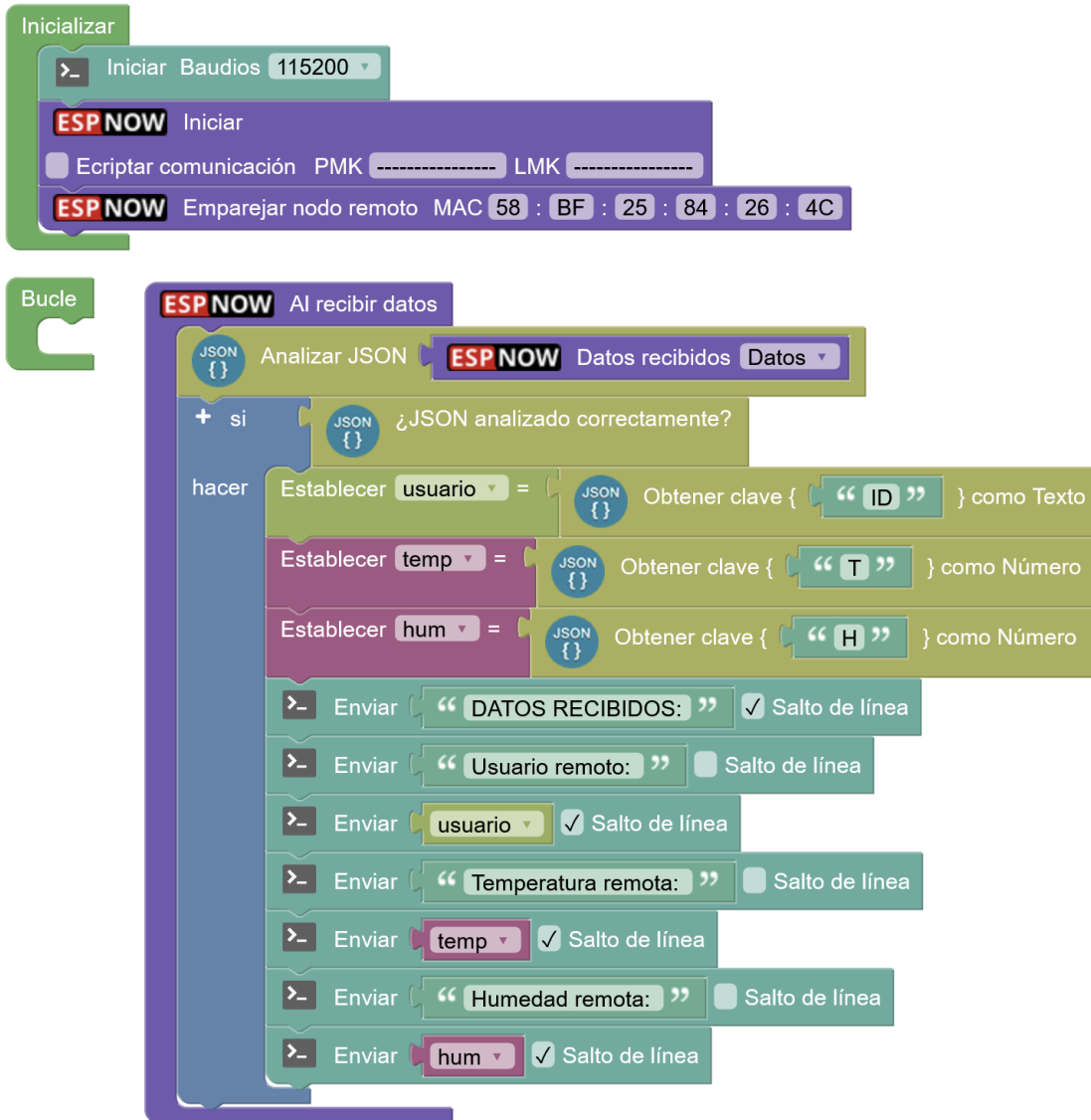
Placa 1 (emisor): Emite valores de temperatura, humedad y nombre de usuario (encapsulados en formato JSON)

The screenshot shows the following code blocks:

- Inicializar:**
 - ESP NOW Iniciar (with checkboxes for 'Ecriptar comunicación', 'PMK', and 'LMK')
 - ESP NOW Emparejar nodo remoto MAC 58 : BF : 25 : 84 : 28 : 60
 - Establecer usuario = "Juanjo"
- Bucle:**
 - Establecer temp = DHT-22 Temperatura °C Pin 26 (D2)
 - Establecer hum = DHT-22 Humedad % Pin 26 (D2)
 - Establecer datos = JSON Codificar {Key:Value} +
 - JSON Key "T" Value temp
 - JSON Key "H" Value hum
 - JSON Key "ID" Value usuario
 - ESP NOW Enviar a MAC 58 : BF : 25 : 84 : 28 : 60 Datos datos
 - Esperar 5000 milisegundos



Placa 2 (receptor): Recibe los datos, analiza los datos en formato JSON para extraer los valores y los envía por consola serie.



Consola serie

Baudrate: 115200 Conectar Desconectar Limpiar

Enviar

```
DATOS RECIBIDOS:
Usuario remoto: Juanjo
Temperatura remota: 26.20
Humedad remota: 56.10
```

4.3 Comunicación Emisor - Receptores múltiples

Una placa ESP32 STEAMakers hará de controlador “maestro” enviando colores RGB generados aleatoriamente, las placas “esclavo” recibirán todas la misma información y controlarán cada una de ellas una tira RGB neopixel ajustada a los valores RGB recibidos.

Programa para emisor:

Aquí añadir todos los nodos remotos con su MAC...

Inicializar

- ESP NOW Iniciar
 - Ecriptar comunicación PMK _____ LMK _____
- ESP NOW Emparejar nodo remoto MAC 58 : BF : 25 : 84 : 28 : 60
- ESP NOW Emparejar nodo remoto MAC 00 : 00 : 00 : 00 : 00 : 00
- ESP NOW Emparejar nodo remoto MAC 00 : 00 : 00 : 00 : 00 : 00
- ESP NOW Emparejar nodo remoto MAC 00 : 00 : 00 : 00 : 00 : 00

Bucle

- Establecer datos = JSON Codificar {Key:Value} +
 - JSON Key "R" Value entero aleatorio de 0 a 255
 - JSON Key "G" Value entero aleatorio de 0 a 255
 - JSON Key "B" Value entero aleatorio de 0 a 255
- ESP NOW Enviar a todos Datos datos
- Esperar 5000 milisegundos

Programa para nodos receptores:

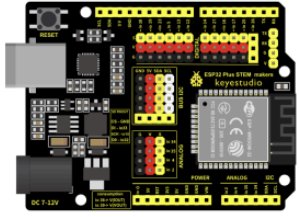
emparejar con el nodo maestro no es necesario si no usamos encriptación

Inicializar

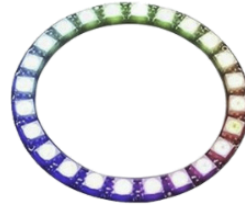
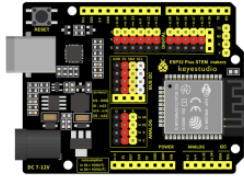
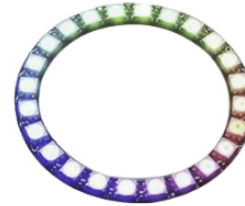
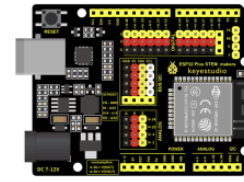
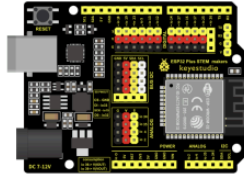
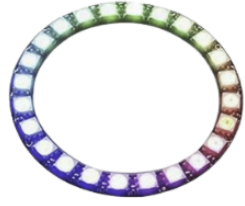
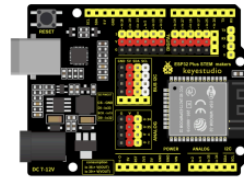
- ESP NOW Iniciar
 - Ecriptar comunicación PMK _____ LMK _____
- ESP NOW Emparejar nodo remoto MAC 58 : BF : 25 : 84 : 26 : 4C
- Iniciar GRB 800Khz Número de píxeles 30 Pin 25 (D3)
- Limpiar
- Mostrar

Bucle

- ESP NOW Al recibir datos
 - JSON Analizar JSON ESP NOW Datos recibidos Datos
 - + si ¿JSON analizado correctamente?
 - hacer
 - Establecer rojo = JSON Obtener clave { "R" } como Número
 - Establecer verde = JSON Obtener clave { "G" } como Número
 - Establecer azul = JSON Obtener clave { "B" } como Número
 - Limpiar
 - contar con i desde 0 hasta 29 de a 1
 - hacer
 - Establecer pixel # i R rojo G verde B azul
 - Mostrar



ESP NOW



4.4 Comunicación Emisores múltiples - Receptor

En este ejemplo se implementa un sencillo sistema de datalogger con nodos distribuidos que envían periódicamente información a un nodo receptor único que recoge toda la información y la registra en una tarjeta microSD en formato CSV.

Los nodos de sensores remotos utilizan el mismo programa, aunque para cada uno habría que asignarles un ID de nodo para poderlos identificar en los datos registrados.

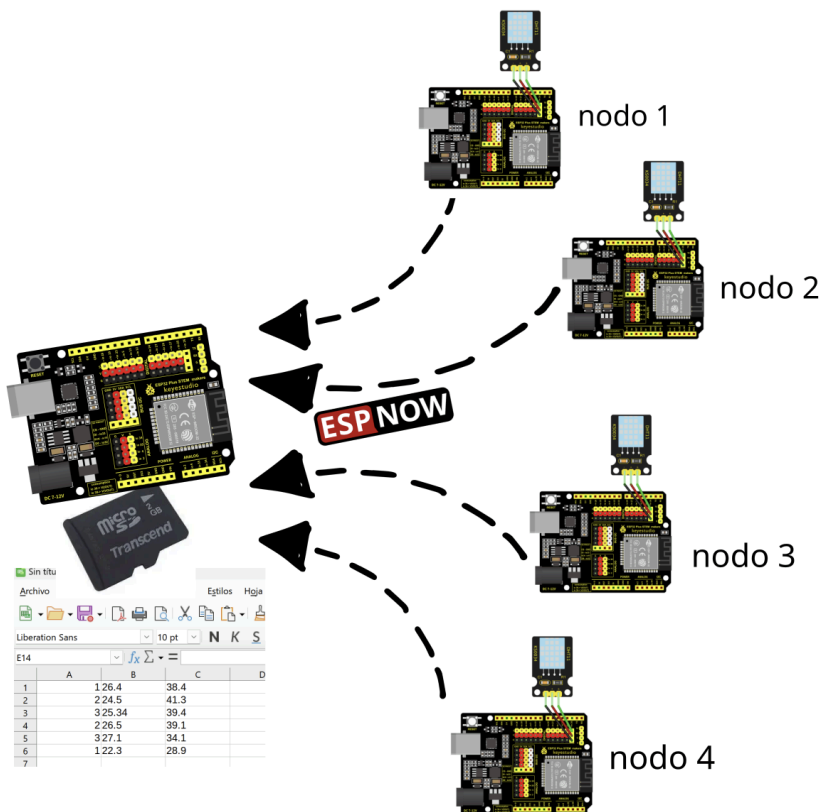
Programa nodos emisores:

The image shows a Scratch-style code editor with the following blocks:

- Inicializar** (Initialize) block:
 - ESP NOW** Iniciar (Start)
 - Ecriptar comunicación (Encrypt communication) with fields for PMK and LMK.
 - ESP NOW** Emparejar nodo remoto (Pair remote node) with MAC address 58 : BF : 25 : 84 : 28 : 60.
 - Establecer (Set) numero de nodo (node number) = 1.
- Bucle** (Loop) block:
 - Establecer (Set) temp = DHT-22 sensor, Temperatura °C, Pin 26 (D2).
 - Establecer (Set) hum = DHT-22 sensor, Humedad %, Pin 26 (D2).
 - Establecer (Set) datos = JSON Codificar {Key:Value} block with three entries:
 - Key "ID", Value numero de nodo.
 - Key "T", Value temp.
 - Key "H", Value hum.
 - ESP NOW** Enviar a MAC 58 : BF : 25 : 84 : 28 : 60 Datos (Send to MAC 58 : BF : 25 : 84 : 28 : 60 Data) with field datos.
 - Esperar (Wait) 15000 milisegundos (milliseconds).

Programa nodo receptor (datalogger):

```
graph TD
    subgraph Inicializar
        A[ESP NOW Iniciar] --> B[Escribir comunicación PMK LMK]
        B --> C[SD Iniciar]
    end
    subgraph Bucle
        D[ESP NOW Al recibir datos] --> E[Analizar JSON Datos recibidos]
        E --> F{¿JSON analizado correctamente?}
        F -- si --> G[Establecer id = Obtener clave { "ID" } como Número]
        G --> H[Establecer temp = Obtener clave { "T" } como Número]
        H --> I[Establecer hum = Obtener clave { "H" } como Número]
        I --> J[Escribir "datos.csv" Texto CSV ; crear texto con id temp hum]
        J --> D
    end
```



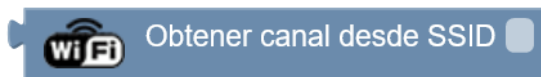
4.5 Comunicación ESP-NOW + WiFi (Telegram)

El microcontrolador ESP32 permite utilizar simultáneamente el módulo interno de radio (2.4GHz) con protocolo ESP-NOW y WiFi, de forma que podemos seguir utilizando la funcionalidad ESP-NOW a la vez que conectamos el módulo a internet.

El siguiente ejemplo es similar al anterior, consta de módulo de sensores remotos que mediante protocolo ESP-NOW todos envían información a un nodo receptor que hará de datalogger, pero en este caso enviando los datos recibidos de los distintos nodos a un chat de Telegram.

La única diferencia en los nodos emisores es que el protocolo ESP-NOW debe configurarse para trabajar en la misma banda de frecuencia que el WiFi al que se va a conectar al módulo receptor (el que se conecta por WiFi a internet). De esta forma los dos módulo emisor-receptor trabajarán en la misma banda para ESP-NOW y WiFi (pues el módulo de radio interno no puede trabajar en dos bandas distintas al mismo tiempo)

Mediante este bloque podemos obtener el canal de la WiFi que vayamos a utilizar simplemente indicando su SSID (nombre del WiFi):

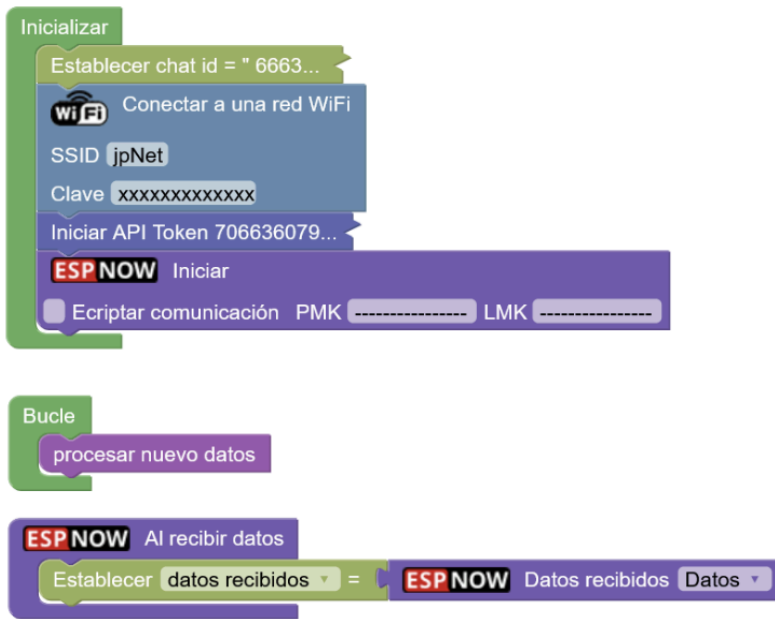


Programa de nodos remotos (sensores):

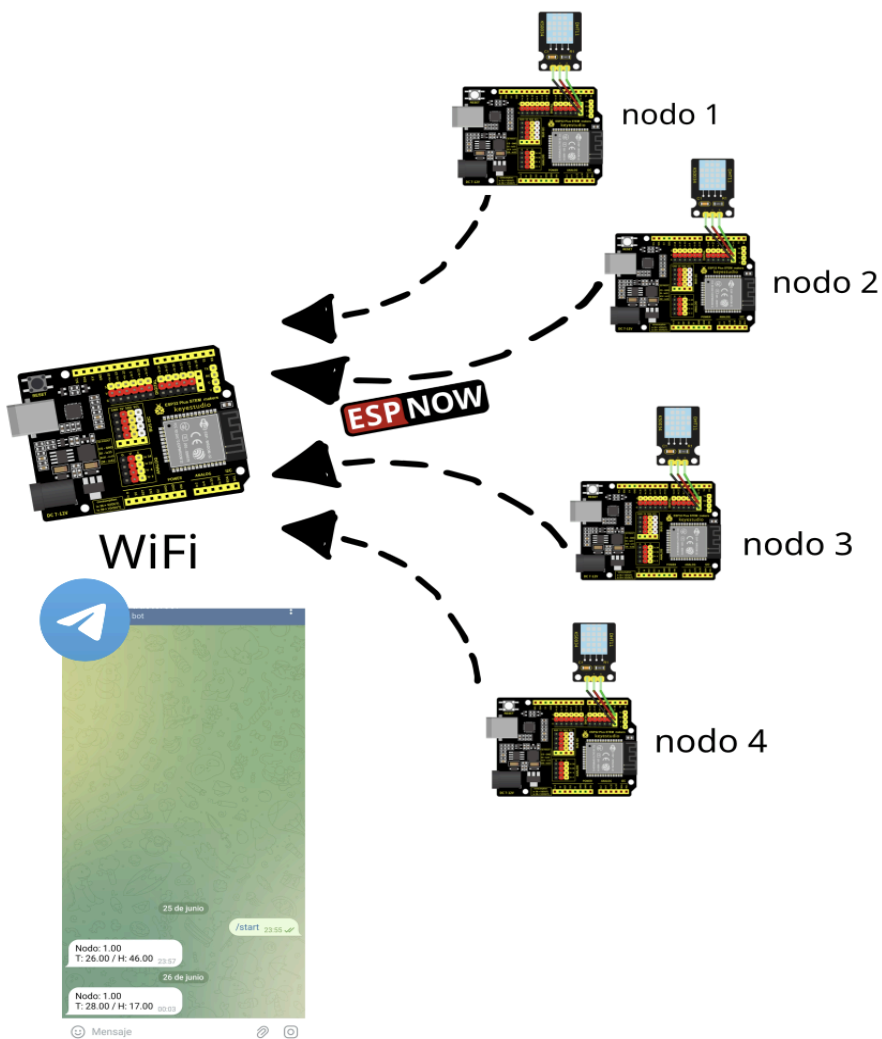
```
void setup() {
  ESP_NOW Establecer canal WiFi Obtener canal desde SSID jpNet
  ESP_NOW Iniciar
  Ecriptar comunicación PMK LMK
  ESP_NOW Emparejar nodo remoto MAC 58 : BF : 25 : 84 : 28 : 60
  Establecer numero de nodo = 1
}

void loop() {
  Ejecutar cada 60000 ms
  Establecer temp = DHT-22 Temperatura °C Pin 17 (D4)
  Establecer hum = DHT-22 Humedad % Pin 17 (D4)
  Establecer datos = Codificar {Key:Value} - +
    JSON {} Key "ID" Value numero de nodo
    JSON {} Key "T" Value temp
    JSON {} Key "H" Value hum
  ESP_NOW Enviar a MAC 58 : BF : 25 : 84 : 28 : 60 Datos datos
}
```

Programa de datalogger Telegram:



Esquema de datalogger Telegram:



Enlaces en información:

<https://www.espressif.com/en/solutions/low-power-solutions/esp-now>

<https://randomnerdtutorials.com/esp-now-esp32-arduino-ide/>

<http://www.arduinoblocks.com>