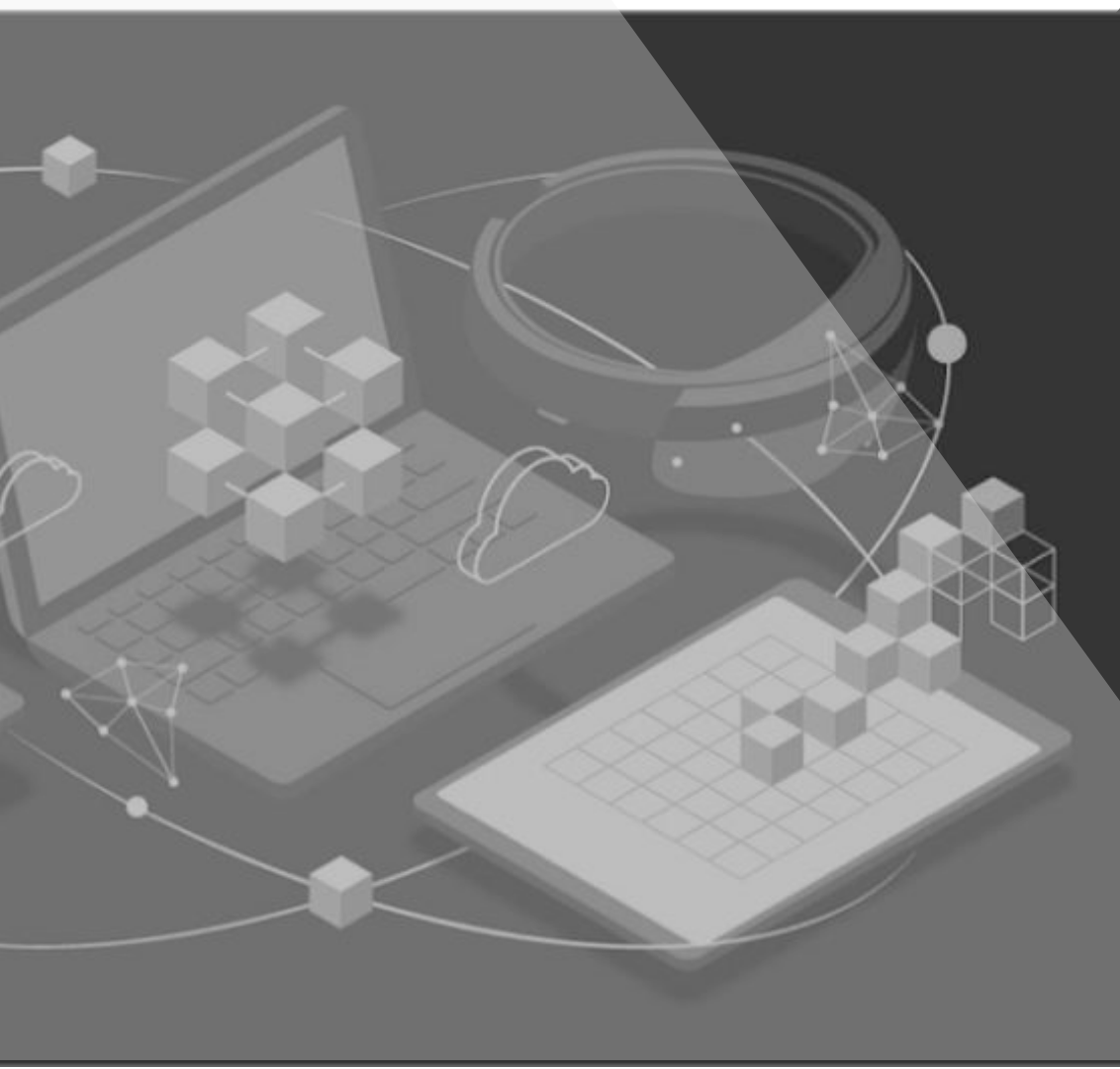




# TABLE OF CONTENTS

---

- 01    **PENGENALAN**  
Pengenalan
- 02    **ESPNOW CONNECTION**  
Memulai dengan ESPNOW  
ESP32
- 03    **TWO WAY CONNECTION**  
Solusi IoT dengan Platform  
Thingsboard
- 04    **PAINLESSMESH**  
Koneksi Esp32 ke dengan  
mesh



# 01

## PENGENALAN

---



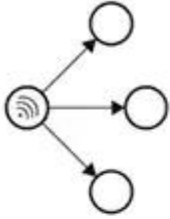
# GOAL

---

- Memahami sistem ESPNow dan metode koneksinya
- Membuat program sensor node dengan master ESPNOW
- Membuat jaringan mesh

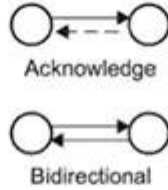


BROADCAST



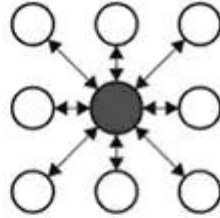
(a)

PEER  
TO  
PEER

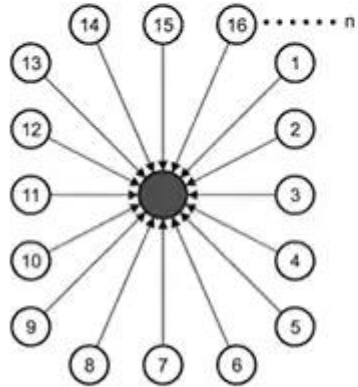


(b)

STAR

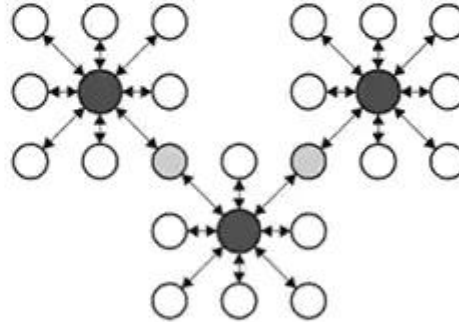


(c)



SCANNING MODE

(d)



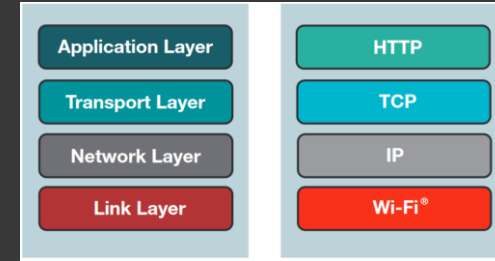
PRACTICAL MESH



(e)

# NETWORK TOPOLOGY

# COMMUNICATION PROTOCOL



	Wi-Fi*	BLE/ Bluetooth 5	Thread	Sub-1 GHz: TI 15.4	Sub-1GHz: Sigfox	Zigbee
<b>Data throughput</b>	Up to 72Mbps	Up to 2Mbps	Up to 250kbps	Up to 200kbps	100bps	Up to 250kbps
<b>Range**</b>	100m	Up to 750m	100m via mesh	4km	25km	130m LOS
<b>Power consumption</b>	Up to 1 year on AA batteries	Up to years on a coin-cell battery	Up to years on a coin-cell battery	Up to years on a coin-cell battery for 1km range	Up to years on a coin-cell battery for limited range	Years on a coin-cell battery
<b>Topology</b>	Star	Point-to-point/Mesh	Mesh & Star	Star	Star	Mesh & Star
<b>IP at the device node</b>	Yes	No	Yes	No	No	No
<b>PC, mobile OS support</b>	Yes	Yes	No	No	No	No
<b>Infrastructure widely deployed</b>	Yes, Access Points	Yes, smart phones	No	No	No	No

\*Single stream 802.11n Wi-Fi MCUs may support lower throughput than peak physical capacity of the network.

\*\*LOS = Line Of Sight. For range, note that maximum data rates are often not available at the longest range.

**Table 1.** Some of the key considerations that will influence the choice of wireless protocols for a specific application, such as data rate, range and power.

# ESPNOW CONNECTION

Tentang ESPNOW

# WROOM32

## PINOUT

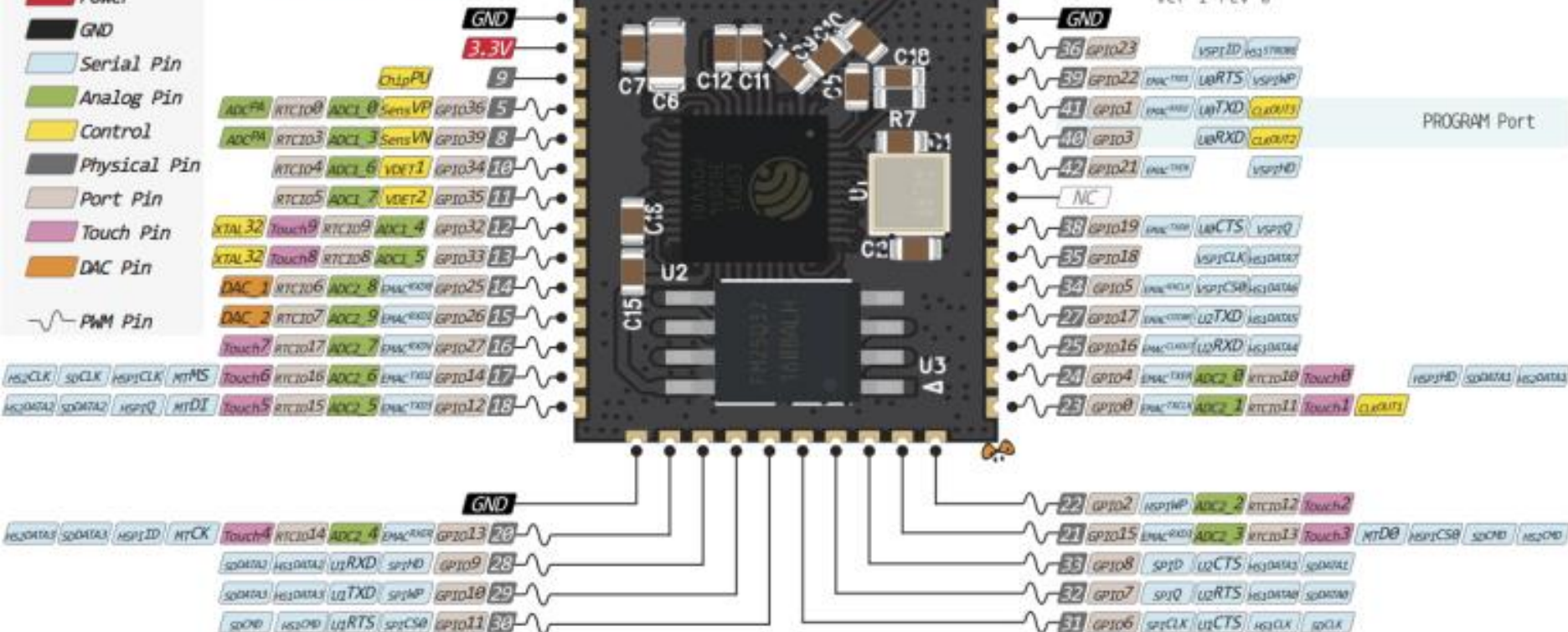
- Power
- GND
- Serial Pin
- Analog Pin
- Control
- Physical Pin
- Port Pin
- Touch Pin
- DAC Pin
- PWM Pin



84 AUG 2016

ver 1 rev 0

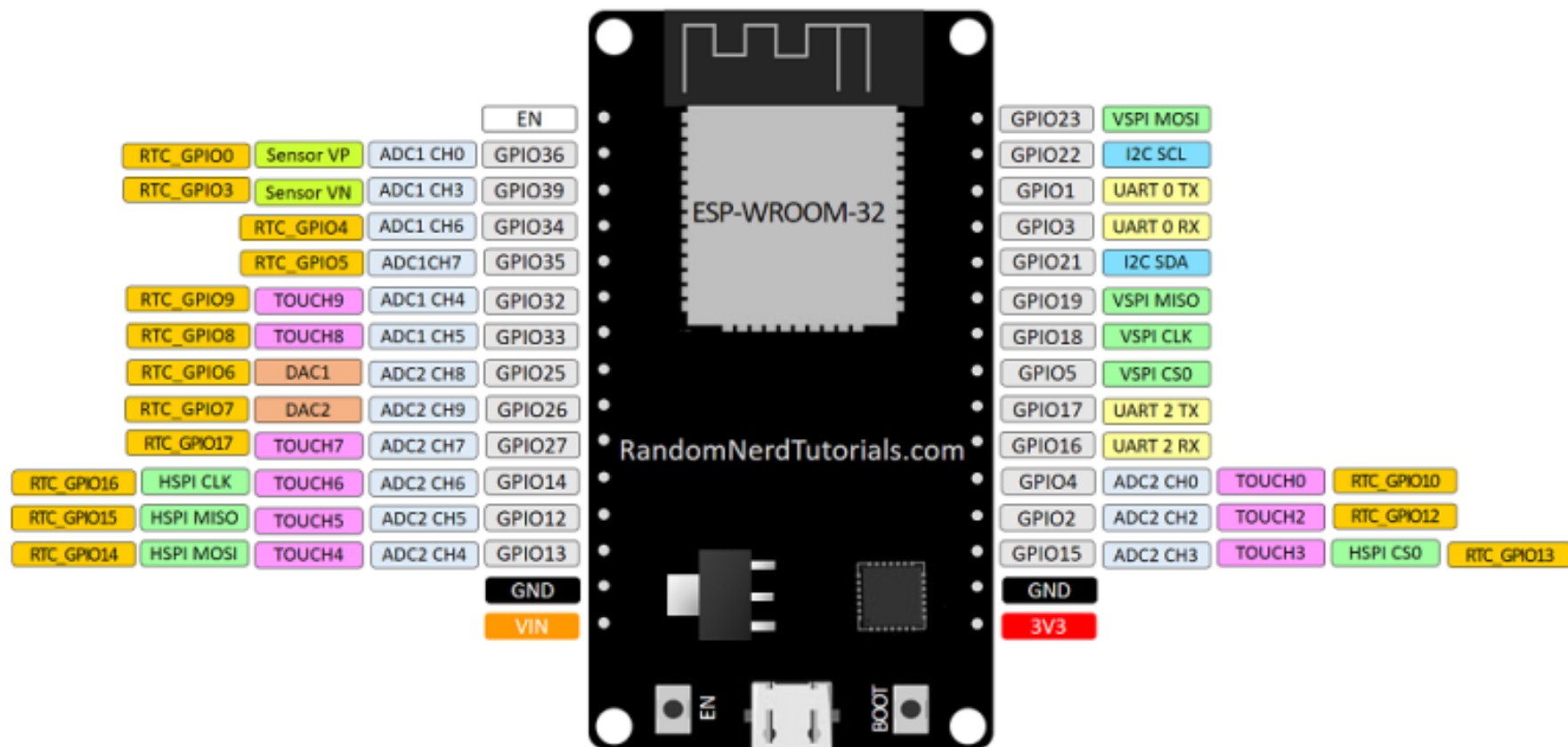
PROGRAM Port





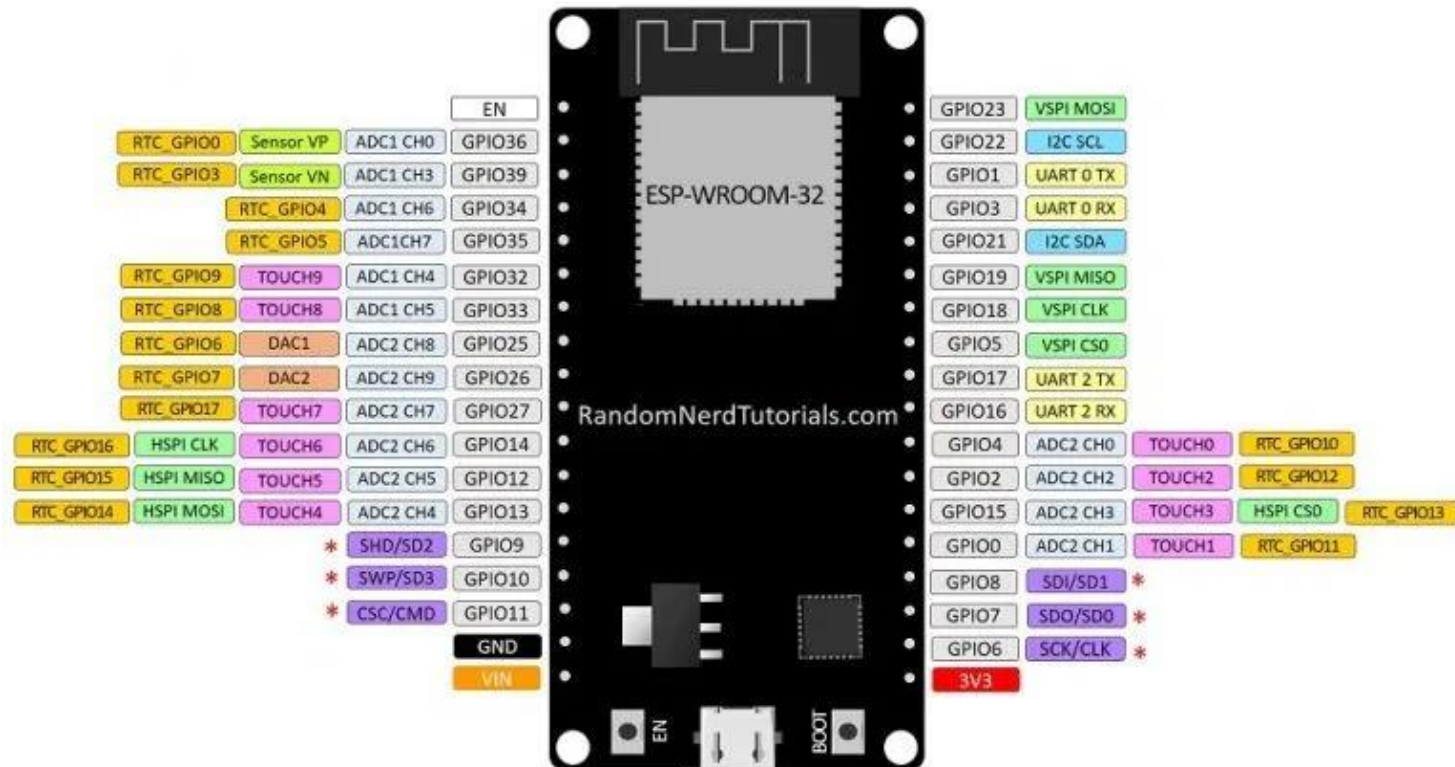
# ESP32 DEVKIT V1 – DOIT

version with 30 GPIOs



# ESP32 DEVKIT V1 – DOIT

version with 36 GPIOs



\* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and CSC/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.

# ESP-NOW

ESP-NOW is a kind of **connectionless** Wi-Fi communication protocol that is defined by Espressif. In ESP-NOW, application data is encapsulated in a vendor-specific action frame and then transmitted from one Wi-Fi device to another without connection. CTR with CBC-MAC Protocol(CCMP) is used to protect the action frame for security. ESP-NOW is widely used in smart light, remote controlling, sensor, etc.

# ESP-NOW

Connectionless communication protocol developed by espressif

Short packet transmission (up to 250 bytes)

Komunikasi tanpa menggunakan Wi-Fi

Mirip komunikasi 2.4Ghz perangkat low power seperti mouse wireless

System menggunakan pairing tanpa hand shake

Maksimum 20 node open dan 10 node jika dengan enkripsi

No router atau dhcp server

No overhead

No lost time to connect

# ESP-NOW ADALAH

Protoocol komunikasi yang cepat yang dapat digunakan untuk pertukaran data kecil (up to 250 bytes) sesama ESP32 board

# HEADER FILE

components/esp\_wifi/include/esp\_now.h

- **esp\_now\_init()** Initializes ESP-NOW. You must initialize Wi-Fi before initializing ESP-NOW.
- **esp\_now\_add\_peer()** Call this function to pair a device and pass as an argument the peer MAC address.
- **esp\_now\_send()** Send data with ESP-NOW.
- **esp\_now\_register\_send\_cb()** Register a callback function that is triggered upon sending data. When a message is sent, a function is called – this function returns whether the delivery was successful or not.
- **esp\_now\_register\_rcv\_cb()** Register a callback function that is triggered upon receiving data. When data is received via ESP-NOW, a function is called.

# CARA MENGIRIM

- Initialize ESP-NOW;
- Register a callback function upon sending data – the OnDataSent function will be executed when a message is sent. This can tell us if the message was successfully delivered or not;
- Add a peer device (the receiver). For this, you need to know the receiver MAC address;
- Send a message to the peer device.

# CARA DARI PENERIMA

- Initialize ESP-NOW;
- Register for a receive callback function (OnDataRecv). This is a function that will be executed when a message is received.
- Inside that callback function, save the message into a variable to execute any task with that information.



# ESP NOW PACKET

No.	Time	Source	Destination	Protocol	Leng	Info	DATA RATE
10	0.857640315	86:f3:eb:73:ca:61	Espressi_73:55:0d	802.11	311	Action, SN=154, ...	6,0
11	0.858121508	86:f3:eb:73:ca:61	Espressi_73:55:0d	802.11	311	Action, SN=154, ...	6,0
12	0.859562409	86:f3:eb:73:ca:61	Espressi_73:55:0d	802.11	311	Action, SN=154, ...	2,0
13	0.860955098	86:f3:eb:73:ca:61	Espressi_73:55:0d	802.11	311	Action, SN=154, ...	2,0
14	0.863632756	86:f3:eb:73:ca:61	Espressi_73:55:0d	802.11	311	Action, SN=154, ...	1,0
15	0.866269693	86:f3:eb:73:ca:61	Espressi_73:55:0d	802.11	311	Action, SN=154, ...	1,0
16	0.868920271	86:f3:eb:73:ca:61	Espressi_73:55:0d	802.11	311	Action, SN=154, ...	1,0
17	0.871582058	86:f3:eb:73:ca:61	Espressi_73:55:0d	802.11	311	Action, SN=154, ...	1,0
18	0.874243732	86:f3:eb:73:ca:61	Espressi_73:55:0d	802.11	311	Action, SN=154, ...	1,0
19	0.876893105	86:f3:eb:73:ca:61	Espressi_73:55:0d	802.11	311	Action, SN=154, ...	1,0
20	0.879536060	86:f3:eb:73:ca:61	Espressi_73:55:0d	802.11	311	Action, SN=154, ...	1,0
37	1.857841049	86:f3:eb:73:ca:61	Espressi_73:55:0d	802.11	311	Action, SN=155, ...	6,0
38	1.858349641	86:f3:eb:73:ca:61	Espressi_73:55:0d	802.11	311	Action, SN=155, ...	6,0
39	1.859748300	86:f3:eb:73:ca:61	Espressi_73:55:0d	802.11	311	Action, SN=155, ...	2,0
40	1.861135593	86:f3:eb:73:ca:61	Espressi_73:55:0d	802.11	311	Action, SN=155, ...	2,0
41	1.863780731	86:f3:eb:73:ca:61	Espressi_73:55:0d	802.11	311	Action, SN=155, ...	1,0

-----  
 | MAC Header | Category Code | Organization Identifier | Random Values | Vendor Specific Content | FCS |  
 -----

24 bytes

1 byte

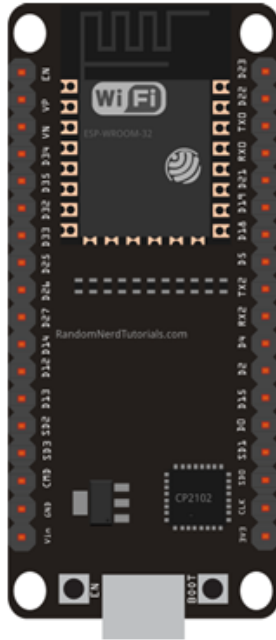
3 bytes

4 bytes

7~255 bytes

4 bytes

# TEST



## ESP-NOW

One-way  
communication



# TOPOLOGY STAR



# PLATFORM IO

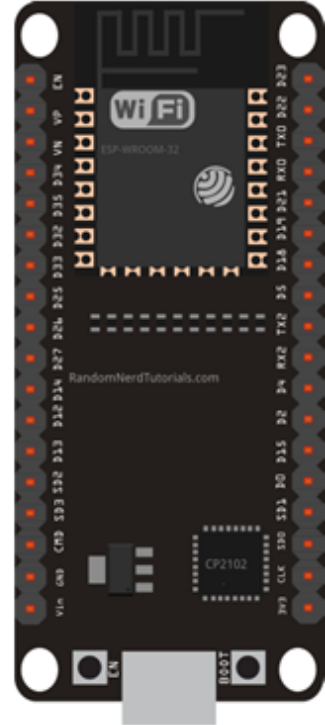
```
monitor_speed = 115200
upload_port = COM6
monitor_port = COM6
lib_deps =
    adafruit/Adafruit Unified Sensor@^1.1.4
    adafruit/Adafruit BMP280 Library@^2.4.2
```

# TEST

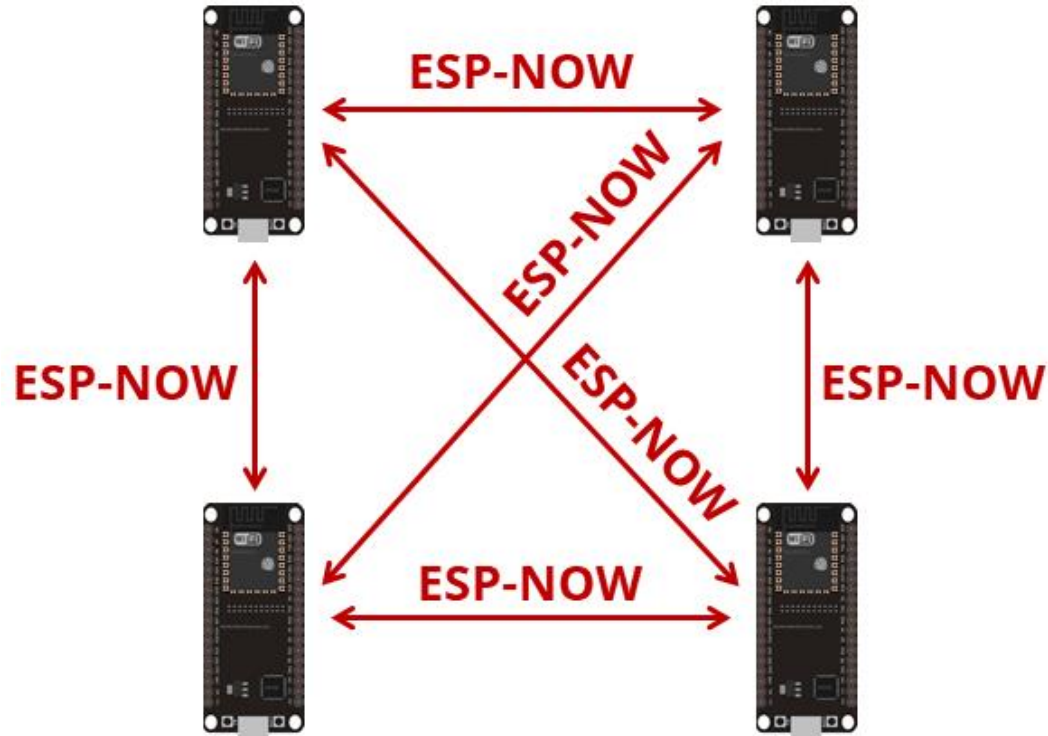


**ESP-NOW**

Two-way  
communication



# TOPOLOGY MESH



# STRUCTURE IN C

```
typedef struct struct_message {  
    uint8_t id;  
    float temp;  
    float pres;  
} struct_message;
```

```
// Create a struct_message to hold incoming sensor readings  
struct_message incomingReadings;
```

# SIZE (TYPE)

Type	Storage size	Value range
char	1 byte	-128 to 127 or 0 to 255
unsigned char	1 byte	0 to 255
signed char	1 byte	-128 to 127
int	2 or 4 bytes	-32,768 to 32,767 or - 2,147,483,648 to 2,147,483,647
unsigned int	2 or 4 bytes	0 to 65,535 or 0 to 4,294,967,295
short	2 bytes	-32,768 to 32,767
unsigned short	2 bytes	0 to 65,535
long	8 bytes or (4bytes for 32 bit OS)	-9223372036854775808 to 9223372036854775807
unsigned long	8 bytes	0 to 18446744073709551615

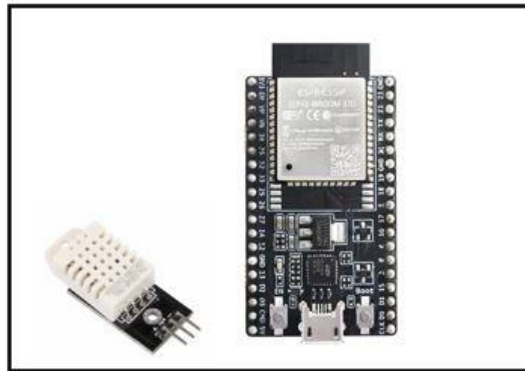


# JSON DATA POINT

```
{  
  "temperature": 42.2,  
  "humidity": 70,  
  "hvacEnabled": true,  
  "hvacState": "IDLE",  
  "configuration": {  
    "someNumber": 42,  
    "someArray": [1,2,3],  
    "someNestedObject": {"key": "value"}  
  }  
}
```

ESP32 Sender #1

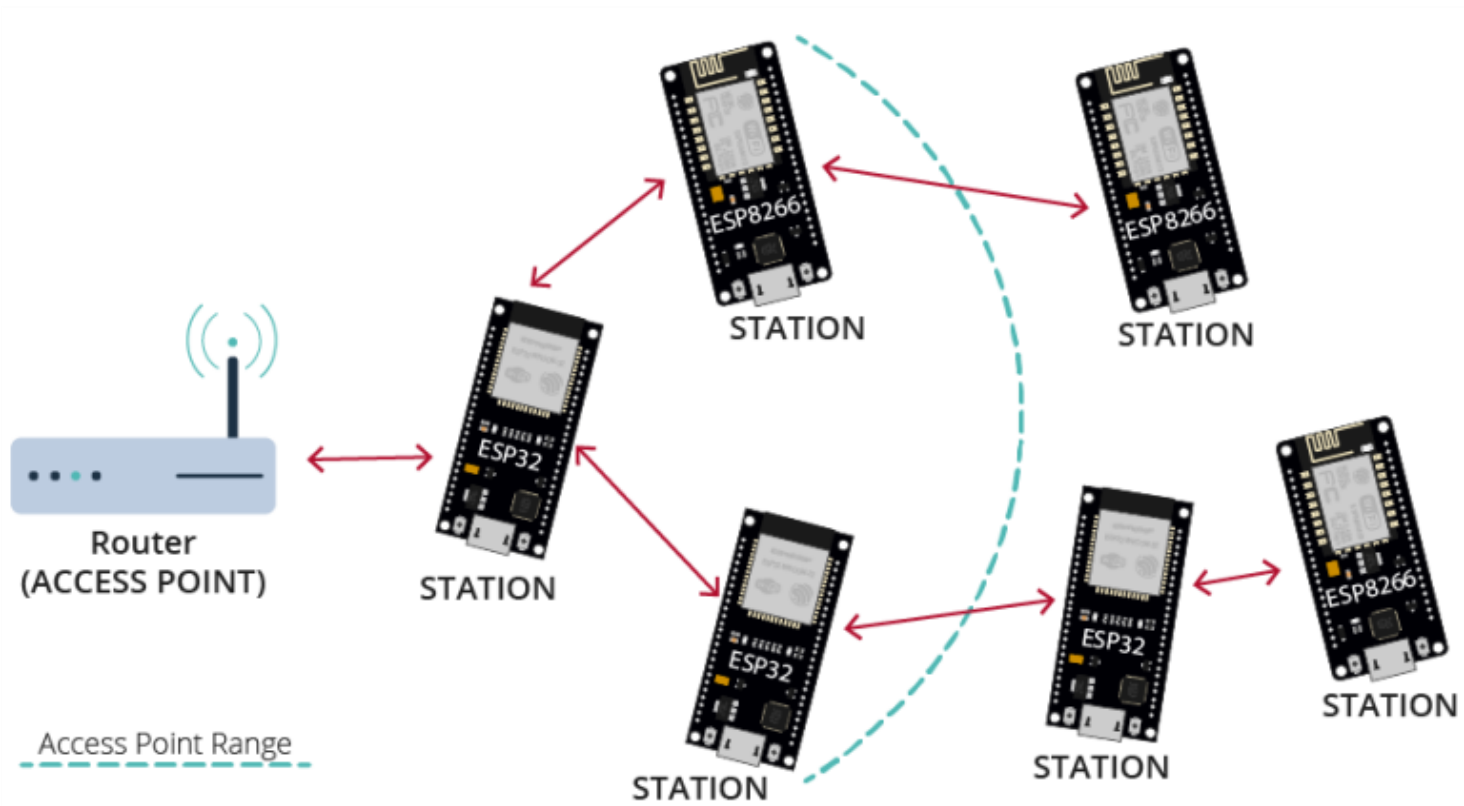
ESP32 Sender #2



ESP32 Receiver  
SERVER



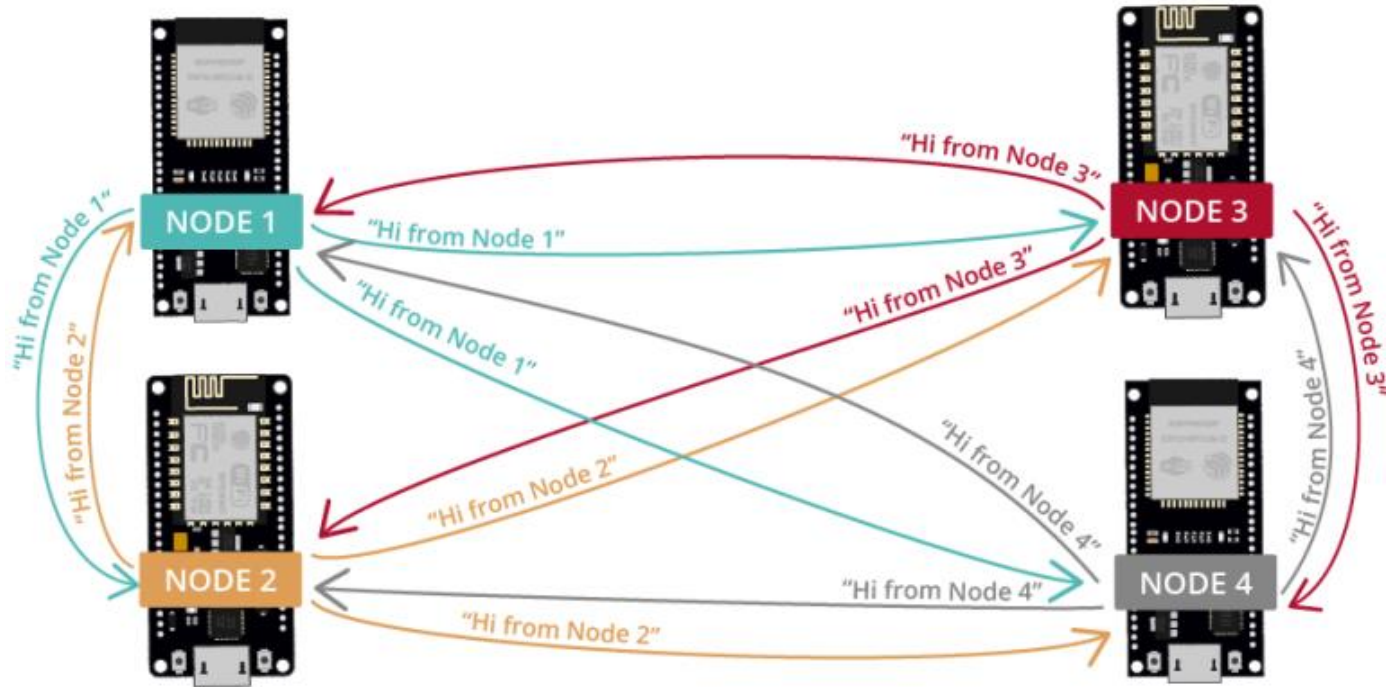
Client



The logo features a large, dark gray triangle pointing upwards, centered on a white background. To the left of the triangle is a light gray triangular area, and to the right is a white triangular area. A thin gray line runs diagonally from the top right corner towards the center. The text "PAINLESSMESH" is centered within the dark gray triangle.

PAINLESSMESH

# TOPOLOGY MESH



# THANKS

Do you have any question?

hasbiida@gmail.com



CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**