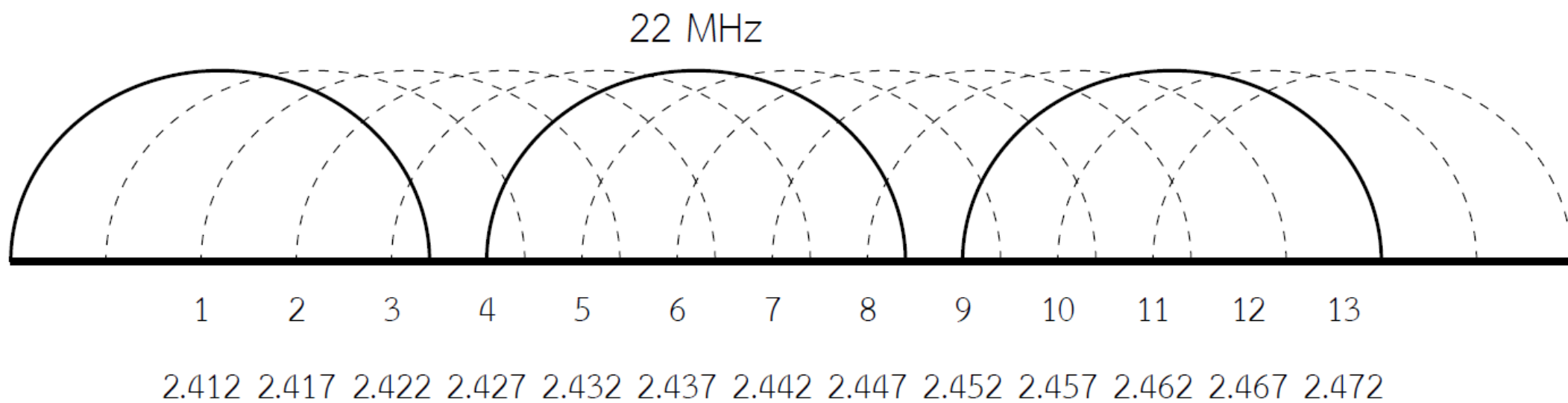


WiFi

# มาตรฐานที่รองรับ

มาตรฐาน	ฟังก์ชัน
b	รองรับการสื่อสารที่ความถี่ 2.4 GHz ที่อัตราเร็ว 11 Mbps
g	รองรับการสื่อสารที่ความถี่ 2.4 GHz ที่อัตราเร็ว 54 Mbps
n	รองรับการสื่อสารที่ความถี่ 2.4 GHz ที่อัตราเร็วมากกว่า 100 Mbps
e	เพื่อให้การสื่อสารใน 802.11 MAC รองรับคุณภาพการให้บริการ (QoS)
i	เพิ่มความปลอดภัยให้ 802.11 MAC และการทำการพิสูจน์ทราบตัวตน

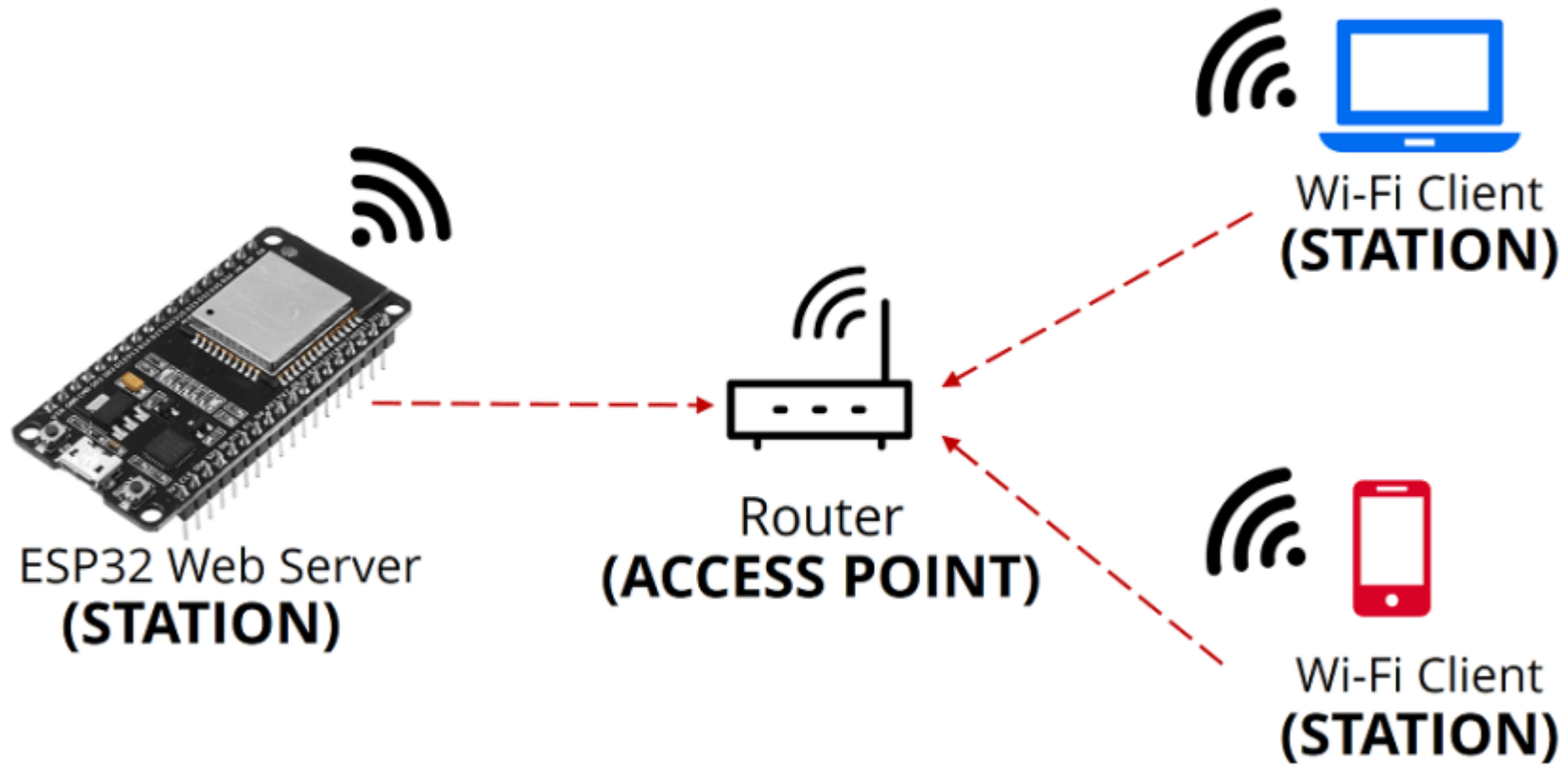
# ช่องสัญญาณ



# Access Point



# Station



# IP Address

- **IPv4 Address = 127.0.0.1 (Localhost)**

- สำหรับทดสอบการใช้งานแอปพลิเคชันที่ทำงานภายใต้การสื่อสารผ่านเน็ตเวิร์ก เพื่อไม่ให้เกิดการสื่อสาร ผ่านข้อมูลไปสู่อินเทอร์เน็ตโดยไม่จำเป็น

- **IP Address = 0.0.0.0**

- กรณีไคลเอนต์ หากอุปกรณ์ดังกล่าวไม่ได้รับการกำหนด IP Address จะทำให้มีแอดเดรสเป็น 0.0.0.0
- กรณีเซิร์ฟเวอร์ เป็น 0.0.0.0 แสดงถึงรองรับทุก IP address สามารถเข้าถึงได้

# การเชื่อมต่อไวไฟของ ESP32

- โหมดสถานี

```
nic = network.WLAN(network.STA_IF)
```

- โหมดแอ็กเซสพอยต์

```
nic = network.WLAN(network.AP_IF)
```

# Station Mode

```
1 import network
2 station = network.WLAN(network.STA_IF)
3 station.active(True)
4 station.connect("<AP_Name>", "<password>")
```



- ทดสอบว่ามีเกิดการเชื่อมต่อเรียบร้อยแล้ว

```
>>> station.isconnected()  
True
```

- ตรวจสอบหมายเลข IP address ที่ได้ผลที่ได้จะเป็น IP address, subnet mask, gateway และ DNS

```
>>> station.ifconfig()  
(192.168.0.64, 255.255.255.0, 192.168.0.1, 192.168.0.1)
```

- ทดสอบความแรงของสัญญาณ

```
>>> station.status(rssi)  
-55
```

- การตรวจสอบหมายเลข mac

```
>>> import ubinascii  
>>> ubinascii.hexlify(network.WLAN().config(mac), :).decode()  
7c:9e:bd:f5:4e:2c
```

# Access Point Mode

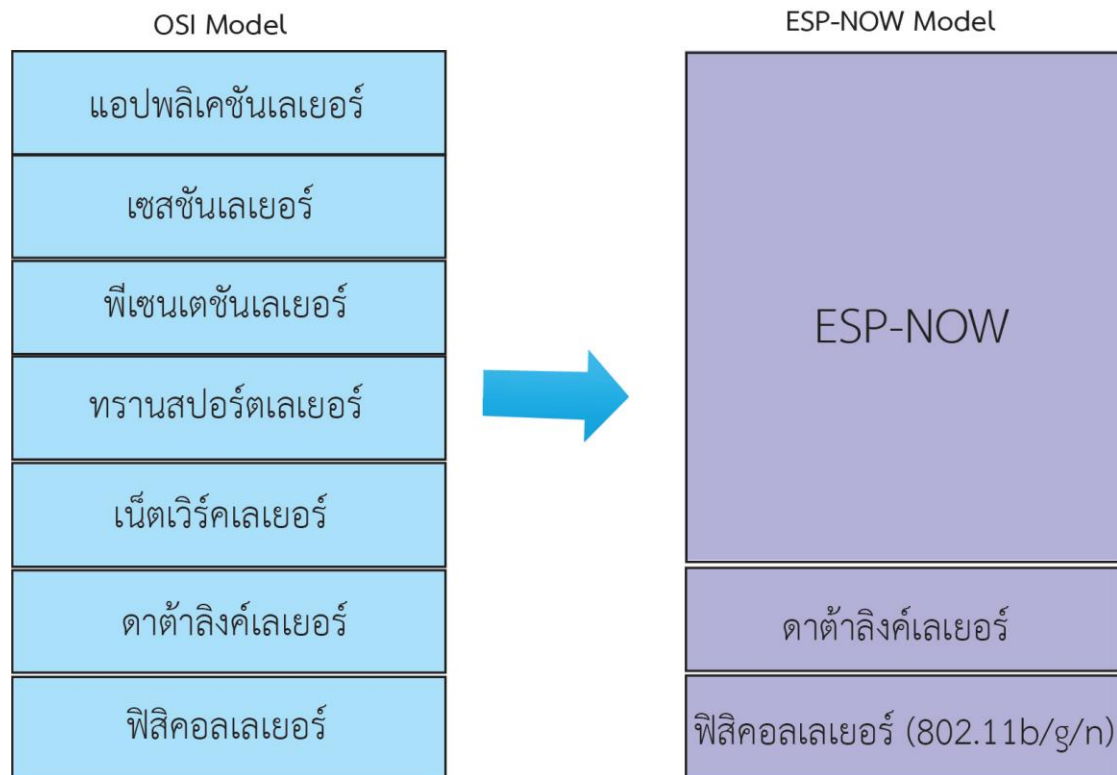
```
1 import network
2 ssid = "mc_ap"
3 password = "123456"
4
5 def create_ap(ssid, password):
6     ap = network.WLAN(network.AP_IF)
7     ap.active(True)
8     ap.config(essid=ssid, password = password)
9
10     while ap.active() == False:
11         password
12
13     print(Acess point ready!)
14     ipinfo = ap.ifconfig()
15     print(ipinfo)
16     return ipinfo[0]
17
18 ip = create_accesspoint(ssid, password)
```

# Assigned IP ให้ Access Point

```
1 import network, socket
2 from time import sleep_ms
3 import gc
4 gc.collect()
5
6 ssid = "myAP"
7 password = "mypassword"
8
9 #assigned ip (IP address, netmask, gateway, DNS)
10 ipassigned = ('192.168.1.1', '255.255.255.0', '192.168.1.1', '8.8.8.8')
11 ap = network.WLAN(network.AP_IF)
12 ap.active(True)
13 ap.config(essid=ssid, password = password)
14 ap.ifconfig(ipassigned)
15
16 while ap.active() == False:
17     pass
18
19
20 print("Access point ready!")
21 print(ap.ifconfig())
22
```

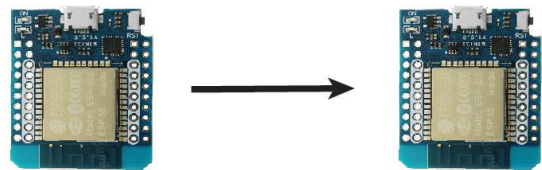
# ESP-NOW

- **ESP-NOW** เป็นโปรโตคอลที่พัฒนาขึ้นโดยบริษัท **Espressif** เพื่ออุปกรณ์ที่มีขนาดเล็กพลังงานต่ำ

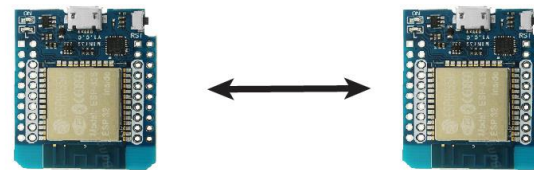


- การสื่อสารสามารถรองรับข้อมูลได้สูงสุดที่ 250 ไบต์
- การสื่อสารสามารถมีเพียร์สูงสุดไม่เกิน 20 อุปกรณ์
- Download firmware  
<https://github.com/glenn20/micropython-espnow-images>

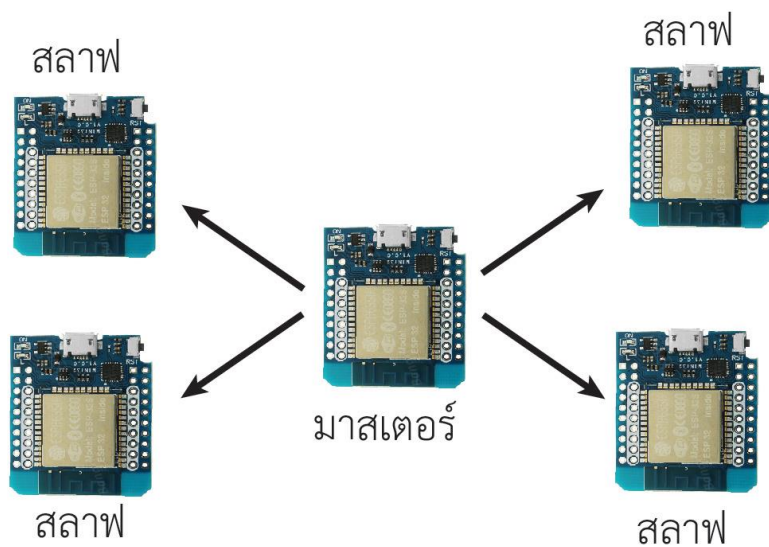
# การสื่อสาร



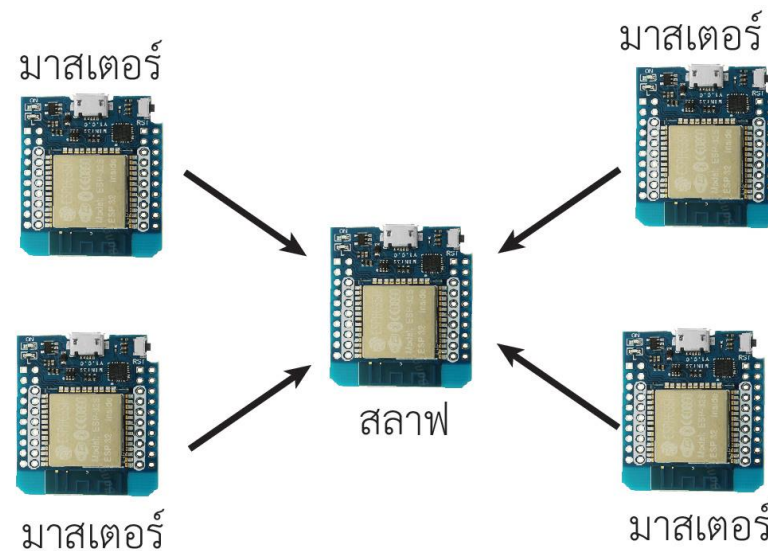
(a)



(b)



(c)



(d)

# MAC Address

```
1 import network, ubinascii
2 wlan = network.WLAN(network.STA_IF)
3 wlan.active(True)
4 wlan_mac = wlan.config('mac')
5 print('MAC Address: {} or {}'.format(wlan_mac, ubinascii.hexlify(wlan_mac).decode().upper()))
6
```

# Sender

```
1 import network, espnow
2 from time import sleep_ms
3
4 sta = network.WLAN(network.STA_IF)
5 sta.active(True)
6
7 # Set up Esp_Now
8 sender = espnow.ESPNow()
9 sender.active(True)
10
11 # Set MAC Address
12 mypeer = b'\xac\x0b\xfb\xce\xbc'
13
14 # add_peer()
15 sender.add_peer(mypeer)
16
17 # Start sending
18 for i in range(10):
19     sleep_ms(100)
20     sender.send(mypeer, 'Hello', True)
21 sender.send(mypeer, b'bye')
22 sleep_ms(100)
23
24 # Show sending Stat
25 print('Send:{}, Response:{}, Failures: {}'.format(sender.stats()[0], sender.stats()[1], sender.stats()[2]) )
```

# Receiver

```
1 import network, espnow
2
3 sta = network.WLAN(network.STA_IF)
4 sta.active(True)
5
6 # Set up Esp_Now
7 receiver = espnow.ESPNow()
8 receiver.active(True)
9
10 # Waiting to receive data
11 while True:
12     host, msg = receiver.recv()
13     print('{} from {}'.format(msg, host))
14     if msg == b'bye':
15         print("END")
16         break
17
18 # Show receiving Stat
19 print('Receive:{}, Drop:{}'.format(receiver.stats()[3], receiver.stats()[4]))
```