ESP-NOW Communication Guide

What is ESP-NOW?

ESP-NOW is a wireless communication protocol developed by Espressif for ESP32 and ESP8266 devices. It allows direct device-to-device communication without needing a WiFi router or access point.

Key Features

- **Fast**: Low latency communication
- **Simple**: No router or WiFi network needed
- **Efficient**: Low power consumption
- **Range**: Up to 200+ meters in open space
- **Peer-to-Peer**: Devices talk directly to each other

How ESP-NOW Works

MAC Address

Every ESP32/ESP8266 has a unique MAC address (like a hardware ID). Devices use MAC addresses to identify and communicate with each other.

Format: `AA:BB:CC:DD:EE:FF` (six pairs of hexadecimal numbers)

Peers

A "peer" is another device you want to communicate with. You must know the peer's MAC address to send messages to it.

Basic Workflow

For Sender Device:

1. **Initialize** ESP-NOW
2. **Get** your MAC address (to share with receiver)
3. **Add peer** using receiver's MAC address
4. **Send** messages to peer

For Receiver Device:

1. **Initialize** ESP-NOW
2. **Get** your MAC address (to share with sender)
3. **Receive** messages in a loop

Block Reference

1. Import ESP-NOW Library

```
[import ESP-NOW library]
```

Place this at the very beginning of your program.

2. Initialize ESP-NOW

```
[ESP-NOW initialize]
```

- Starts ESP-NOW
- Prints your device's MAC address to the console
- **Important**: Copy this MAC address to use on other devices!

3. Get MAC Address

```
[ESP-NOW get MAC address] → returns string
```

Returns your device's MAC address as a string (e.g., "AA:BB:CC:DD:EE:FF")

4. Add Peer

```
[ESP-NOW add peer] [MAC address]
```

- Adds another device as a communication partner
- You need the other device's MAC address
- Format: "AA:BB:CC:DD:EE:FF"

5. Send Message (Simple)

```
[ESP-NOW send] [message] [to first peer]
```

Sends a message to the first peer you added (peer 0)

6. Send Message (Advanced)

```
[ESP-NOW send] [message] [to peer] [number]
```

Sends to a specific peer by index:

- 0 = first peer added
- 1 = second peer added
- etc.

7. Receive Message (Simple)

```
[ESP-NOW receive] → returns string or None
```

- Waits 1 second (1000ms) for a message
- Returns the message text if received
- Returns `None` if no message

8. Receive Message (Advanced)

```
[ESP-NOW receive (timeout] [milliseconds] [ms)] → returns string or None
```

- Custom timeout period
- 1000 = 1 second, 5000 = 5 seconds, etc.

Example Programs

Example 1: Simple Sender

Steps:

1. Run this program first to get your MAC address
2. Copy the MAC address from the console
3. Give this MAC to the receiver device

Program:

```
import ESP-NOW library
ESP-NOW initialize
wait 2 seconds
add peer "BB:BB:BB:BB:BB:BB"  ← Replace with receiver's MAC
forever:
    send "Hello!" to first peer
    wait 2 seconds
```

Example 2: Simple Receiver

Steps:

1. Run this program first to get your MAC address
2. Copy the MAC address from the console
3. Use this MAC on the sender device

Program:

```
import ESP-NOW library
ESP-NOW initialize
forever:
    set [message] to [ESP-NOW receive]
    if message ≠ None:
        print message
```

Example 3: Two-Way Communication

Both devices run similar code and act as sender AND receiver:

```
import ESP-NOW library
ESP-NOW initialize
add peer "AA:AA:AA:AA:AA:AA"  ← Other device's MAC

forever:
    // Try to receive
    set [message] to [ESP-NOW receive (timeout 100 ms)]
    if message ≠ None:
        print message

    // Send periodically
    if time % 3000 < 100:
        send "Hello from Device 1" to first peer
```

Common Patterns

Pattern 1: Sensor Data Broadcasting

One device reads sensor data and broadcasts to multiple receivers:

```
Sender:
- Read sensor
- Send data to all peers
- Wait and repeat

Receivers:
- Continuously listen
- Process received data
- Display or log
```

Pattern 2: Remote Control

One device controls another (like a remote):

```
Controller:
- Read button presses
- Send commands ("UP", "DOWN", "LEFT", "RIGHT")

Controlled Device:
- Receive commands
- Execute actions based on command
```

Pattern 3: Mesh Network

Multiple devices can all talk to each other:

```
Each device:
- Adds all other devices as peers
- Can send to any peer
- Receives from all peers
```

Important Tips

📍 Getting MAC Addresses

1. Upload a simple program with just `ESP-NOW initialize`
2. Open the serial console
3. Copy the MAC address that appears
4. Share this MAC with other devices

⚠ Common Issues

"No message received"

- Check both devices are initialized
- Verify MAC addresses are correct
- Make sure devices are in range (< 200m)
- Check for typos in MAC address format

"Peer not added"

- Verify MAC address format: `AA:BB:CC:DD:EE:FF`
- Use colons `:` between each pair
- Use uppercase letters for A-F

Messages not appearing

- Make sure receiver is continuously checking (use a loop)
- Add a small delay in loops to avoid overwhelming the CPU
- Check if `None` is returned (no message available)

💡 Best Practices

1. **Always initialize first** before any other ESP-NOW operation
2. **Copy MAC addresses carefully** - one wrong character breaks communication
3. **Use loops for receiving** - messages arrive at unpredictable times
4. **Check for None** - not every receive call will have a message
5. **Add delays** - don't send/receive too fast (at least 10-100ms between operations)
6. **Keep messages short** - maximum 250 bytes per message

Range and Limitations

Range

- **Indoor**: 50-100 meters (depending on walls)
- **Outdoor**: 200+ meters in open space
- Affected by obstacles, interference, antenna quality

Limitations

- Maximum 250 bytes per message
- Up to 20 peers can be added
- No automatic acknowledgment (you don't know if message was received)
- No encryption by default (messages are sent in plain text)

Quick Start Checklist

For Device 1 (Sender):

- [ ] Import ESP-NOW library
- [ ] Initialize ESP-NOW
- [ ] Note your MAC address from console
- [ ] Add Device 2's MAC as peer
- [ ] Send messages

For Device 2 (Receiver):

- [ ] Import ESP-NOW library

- [ ] Initialize ESP-NOW
- [ ] Note your MAC address from console
- [ ] Give your MAC to Device 1
- [ ] Loop to receive messages

Connect them:

- [ ] Device 1 uses Device 2's MAC address
- [ ] Both devices are powered on
- [ ] Both devices are within range
- [ ] Test communication!

Next Steps

Once you master the basics:

- Try bidirectional communication (both send and receive)
- Add multiple peers for mesh networks
- Send sensor data wirelessly
- Create remote control projects
- Build IoT networks without WiFi routers

Happy making! 🚀