

ESP-NOW communication protocol

ESP-NOW is a wireless communication protocol defined by Espressif, which enables the direct, quick and low-power control of smart devices, without the need of a router. ESP-NOW can work with Wi-Fi and Bluetooth LE, and supports the ESP8266, ESP32, ESP32-S and ESP32-C series of SoCs. It's widely used in smart-home appliances, remote controlling, sensors, etc. There two communication protocol for esp-now:

1. One way communication
2. Two way communication

1. **One way communication**: There are three possible way to communicate-

- I. Sending data from a master to a Slave
- II. Sending data from a master to multiple slave
- III. Sending data from multiple slave to one master

- I. Sending data from a master to a slave: For this, here I have built a simple project, I have two TTGO T-call V1.4 esp32. In master esp32 I have set a random number to send the receiver. And in the slave esp32 serial monitor shows the random values. Here some pictures of this communication system-



Figure 1: One way communication

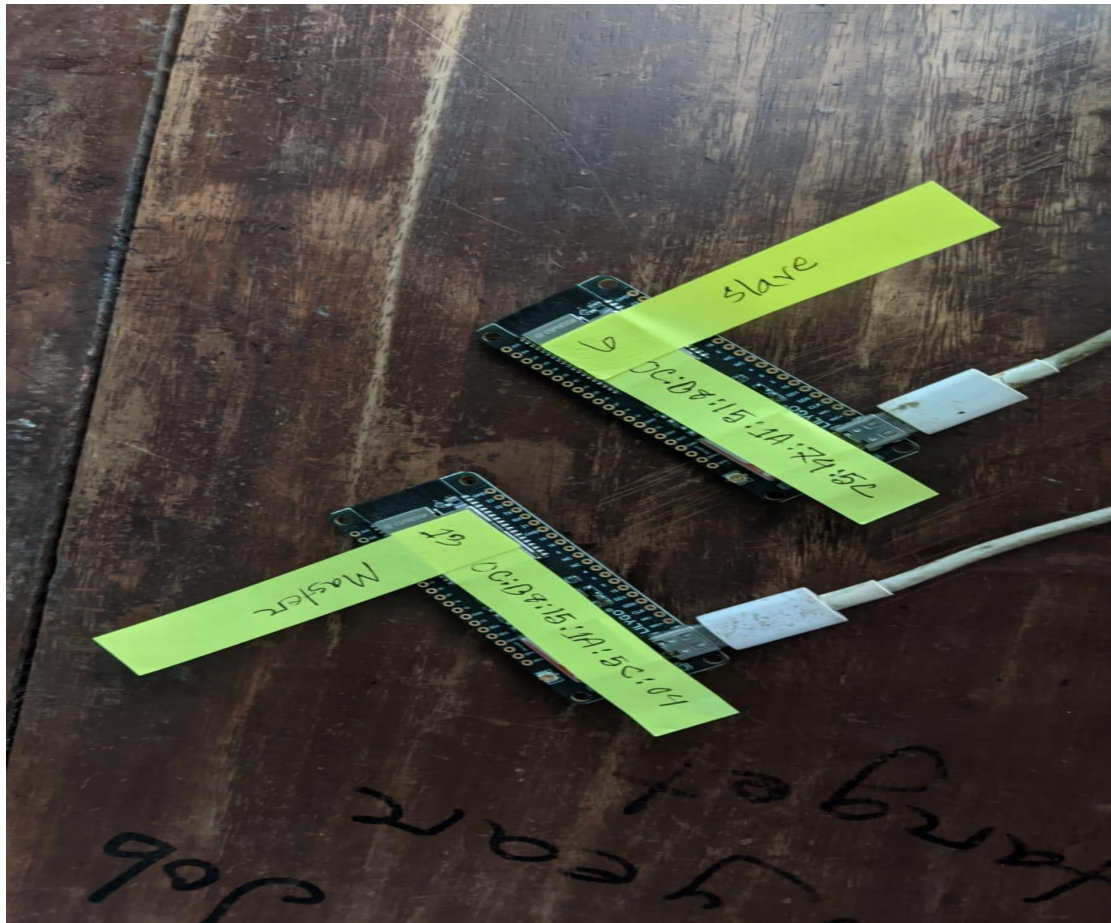


Figure 2: One way communication set-up

In figure 2 from master to slave data is transferred. Here is the serial monitor screen shot where a random data is send to the slave-

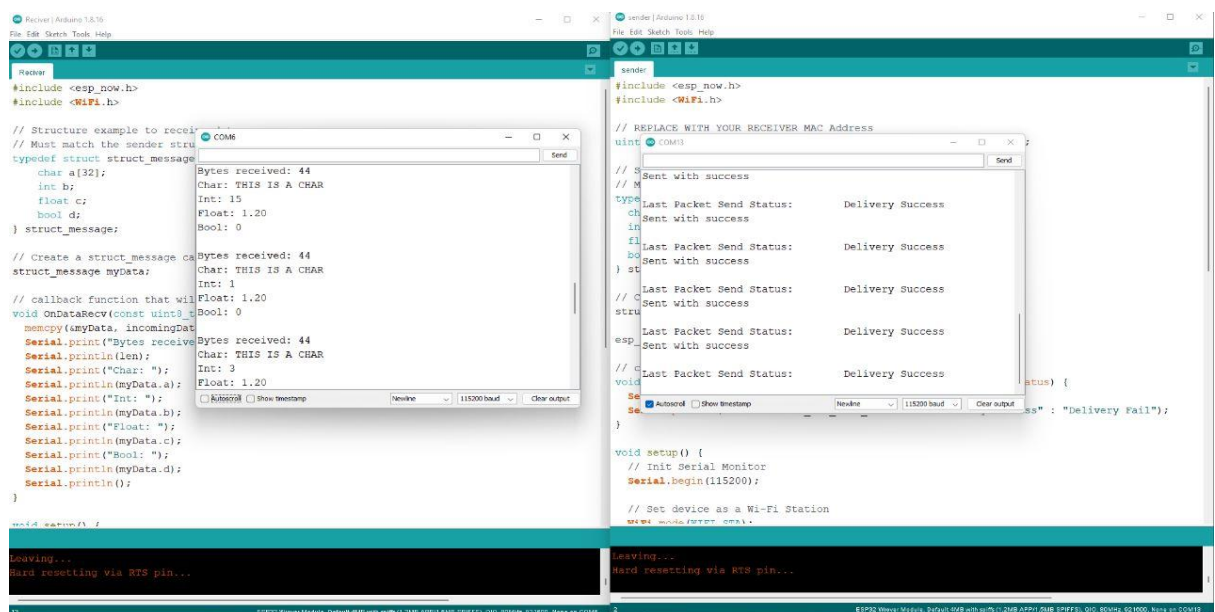


Figure 3: A random data send to slave esp from master(one way communication)

- II. Sending data from a master to multiple slaves: Here I have one master and 5 slaves. In master esp I have set a random number to send it into the slaves esp. Also I have set nine different random numbers to send the slaves esp. Both processes are run successfully. Here some pictures of this communication process:



Figure 4: One way communication (One master to multiple slaves)

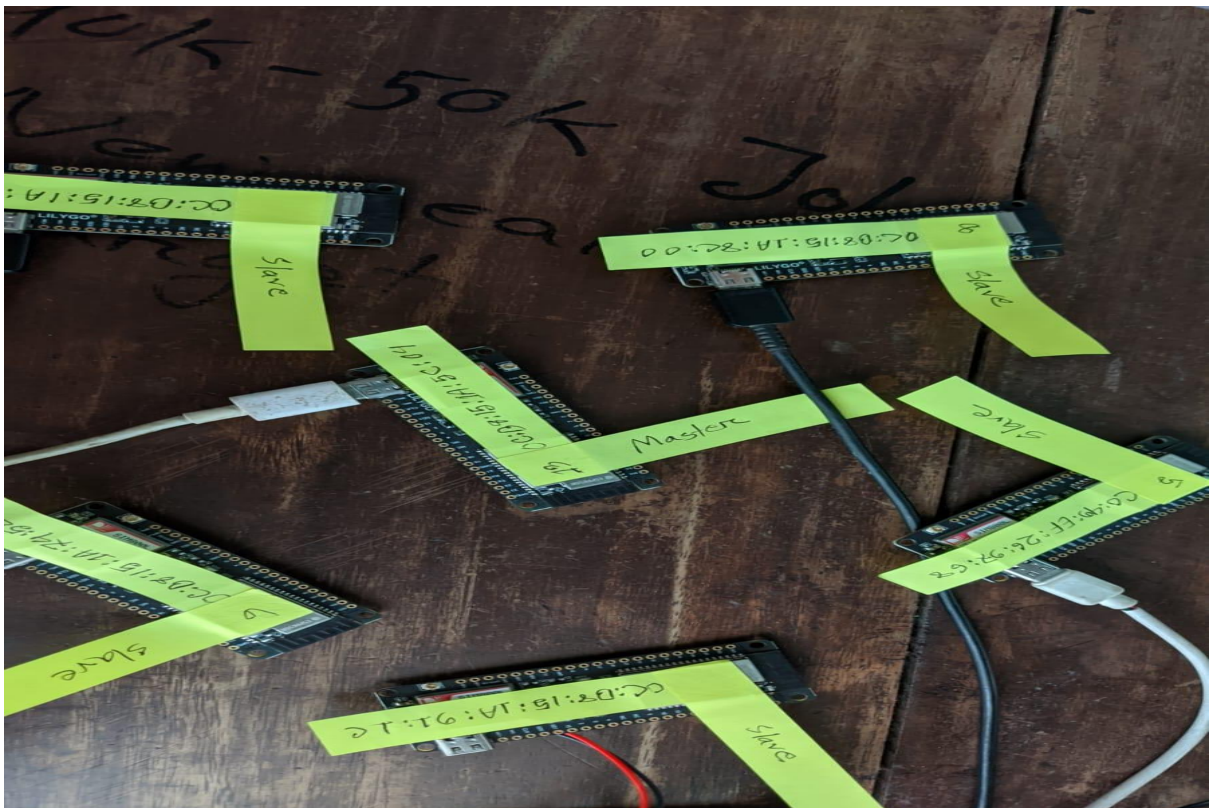


Figure 5: One way communication (One master to multiple slaves) set-up

In figure 5 data sent from master to different slaves address(random same data or different data). Here in serial monitor message-

```

// ESP_NOW_Slave
#include <esp_now.h>
#include <WiFi.h>

//Structure example to receive data
//Must match the sender structure
typedef struct test_struct {
  int x;
  int y;
} test_struct;

//Create a struct_message called myData
test_struct myData;

//callback function that will be executed when data is received
void onDataRecv(const uint8_t * mac, const uint8_t *incomingData,
  memcpy(myData, incomingData, sizeof(myData));
  Serial.print("Bytes received: ");
  Serial.println(len);
  Serial.print("x: ");
  Serial.println(myData.x);
  Serial.print("y: ");
  Serial.println(myData.y);
  Serial.println();
}

void setup() {
  //Initialize Serial Monitor
  Serial.begin(115200);

  //Set device as a Wi-Fi Station
  WiFi.mode(WIFI_STA);
}

//Serial Monitor
Serial.println("leaving...");
Serial.println("Hard resetting via RTS pin...");

// ESP_NOW_Master
#include <esp_now.h>
#include <WiFi.h>

// REPLACE WITH YOUR ESP RECEIVER'S MAC ADDRESS
uint8_t macs[] = {
  0x80, 0x15, 0x1a, 0x8c, 0x00,
  0x49, 0xef, 0x26, 0x97, 0xe0,
  0x80, 0x15, 0x1a, 0x74, 0x5c,
  0x49, 0xef, 0x1d, 0xe0, 0x50,
  0x80, 0x15, 0x1a, 0x82, 0xc0,
  0x80, 0x15, 0x1a, 0x91, 0xc,
  0x80, 0x15, 0x1a, 0x5c, 0x00,
  0x80, 0x15, 0x1a, 0x8c, 0x00,
  0x80, 0x15, 0x1a, 0x8c, 0xe0,
};

void onDataSend(const uint8_t *mac_addr, esp_now_send_status_t status) {
  char macStr[18];
  Serial.print("Packet to: ");
  // Copies the sender mac address to a string
  snprintf(macStr, sizeof(macStr), "%02x:%02x:%02x:%02x:%02x",
    mac_addr[0], mac_addr[1], mac_addr[2], mac_addr[3], mac_addr[4], mac_addr[5]);
  Serial.println(macStr);
}

void setup() {
  //Initialize Serial Monitor
  Serial.begin(115200);

  //Set device as a Wi-Fi Station
  WiFi.mode(WIFI_STA);
}

//Serial Monitor
Serial.println("leaving...");
Serial.println("Hard resetting via RTS pin...");

```

Figure 6: A random data send to multiple slaves esp from one master(one way communication)

III. Sending data from multiple slaves to one master: In this system, nine slaves esp with different messages sent to one master esp. Here also send a unique id number for each slave esp to the master as easily identify which data come from which slave esp. Here some pictures of this communication process:



Figure 3: One way communication(Multiple slaves to One master)

2. **Two way communication:** In two way communication first I have tested two esp send data to each other. Then I have set 3 esp to communicate with each other and it successfully operates. Here some pictures of two way communication of ESP-NOW protocol-



Figure 9: Two esp communicate each other with two way communication protocol

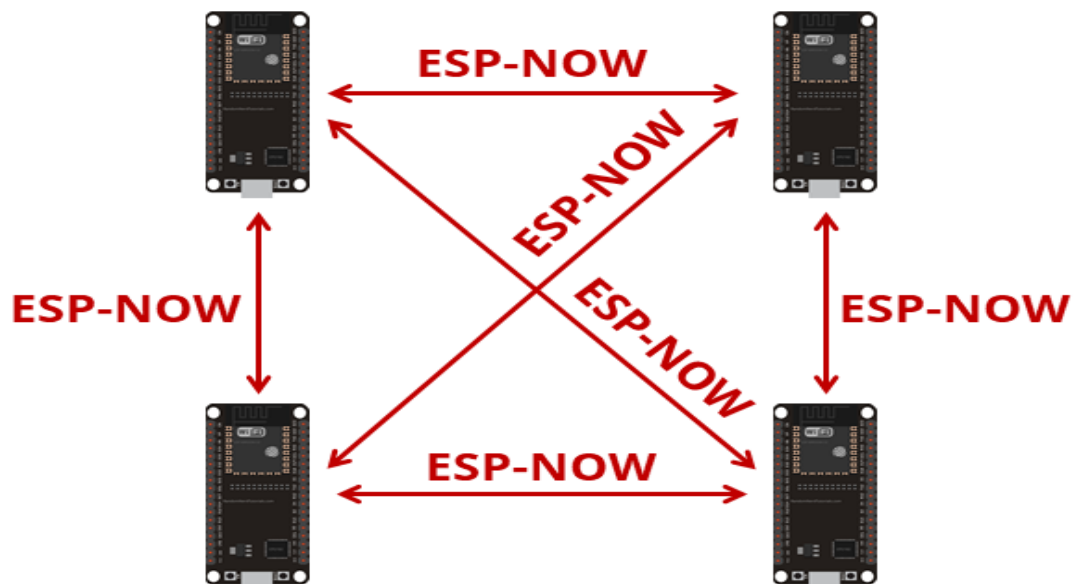


Figure 10: 4 esp communicate each other with two way communication protocol

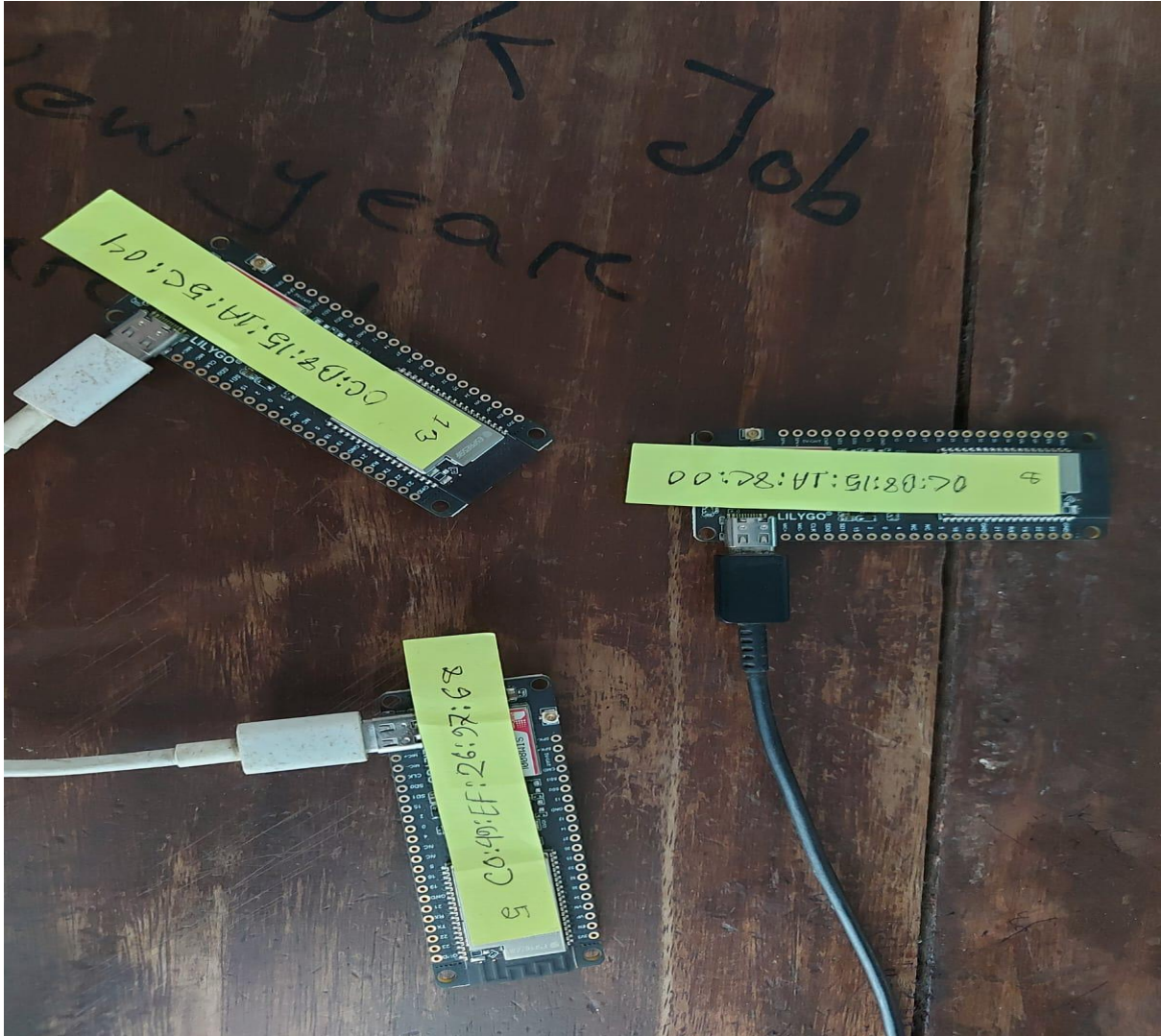


Figure 11: 3 esp communicate each other with two way communication protocol set-up

In Figure 11 each esp communicates with data transfer to each other. Like ID 5 receives data from ID 8 and ID 13 and sends data to them. Same for each esp. Here is the serial monitor message -

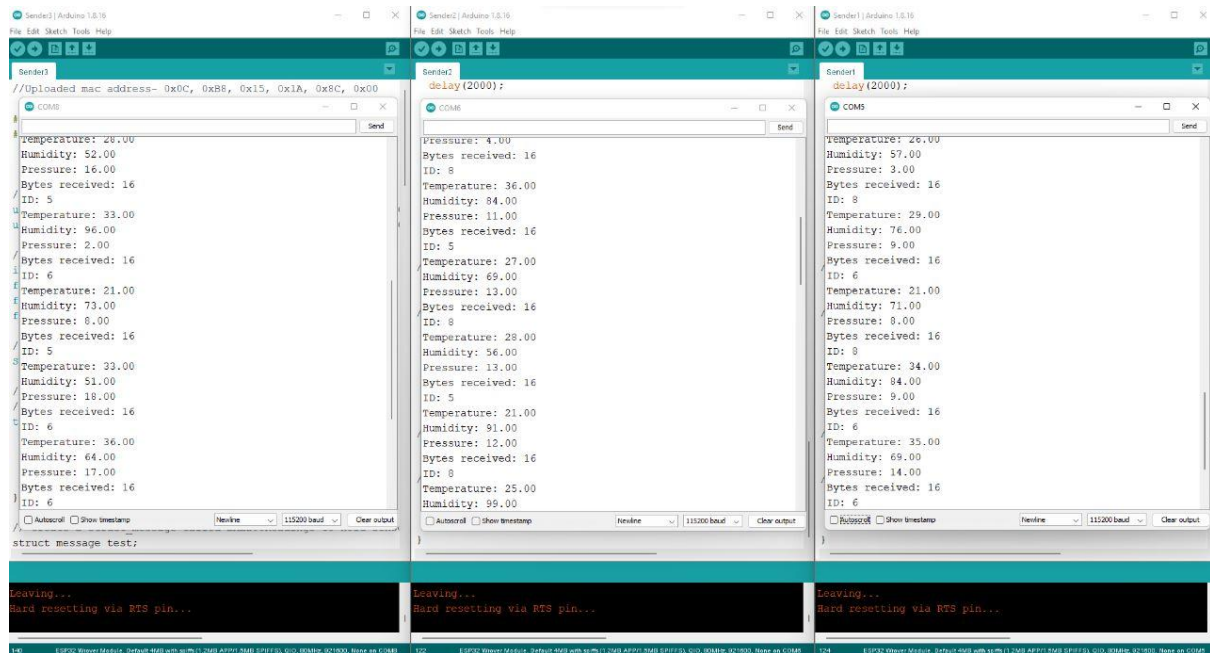


Figure 12: 3 esp communicate each other with two way communication protocol