ESP32 NOW

WIRELESS COMMUNICATION

ESP NOW

Used to allow communication between devices wirelessly

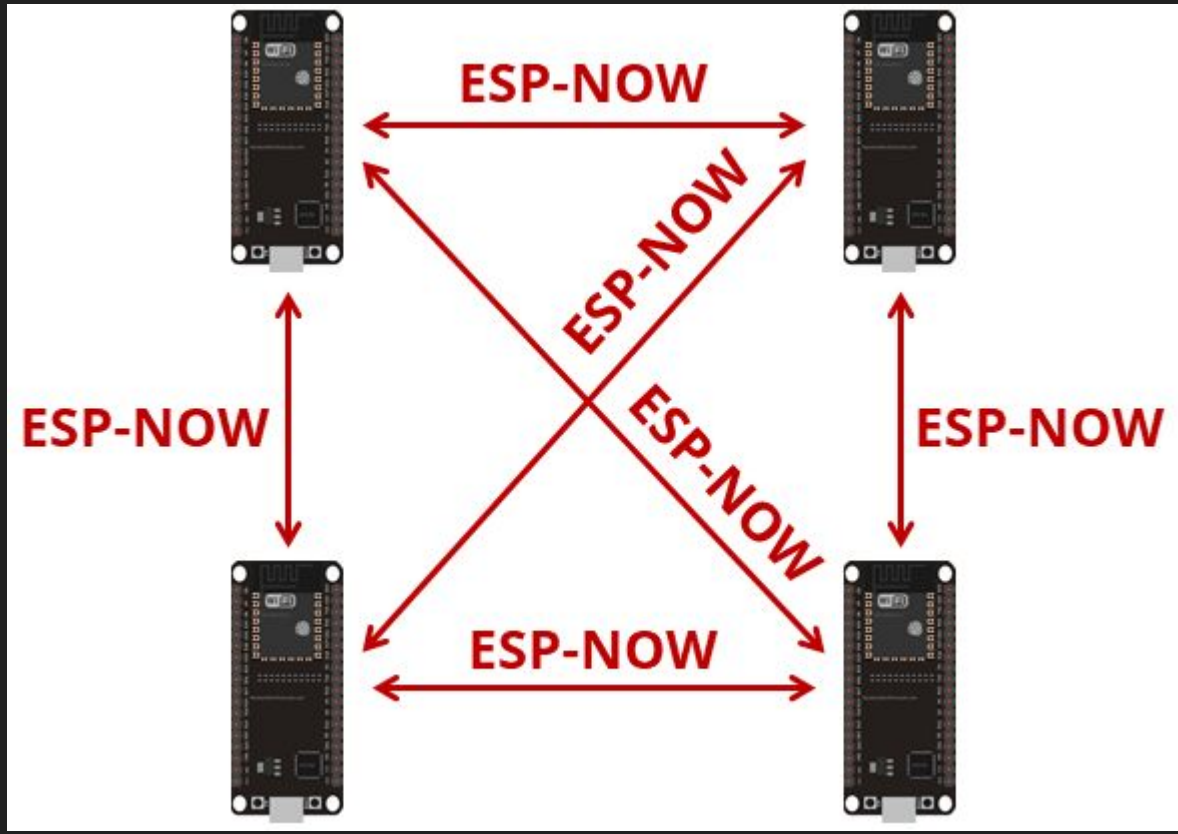
Data is sent from one esp32 to another. Good for sending data from sensor to another esp32. Note that the payload maximum size 250 bytes. The master sends the data/instruction. The slave receives the data/instruction.

Similar to the last one but in this scenario one esp32 is sending data to multiple esp32s. The main sending esp32 is sometimes referred to as the master and the receivers are the slaves. Slaves receive commands from their masters. This is good in scenarios where one esp32 can remote control several others.

In this scenario one esp32 acts as a slave to several masters. Each master could be collecting data and sending it to one esp32 to record the information.

Each esp32 acts as both
a master and slave.

Finally, each esp32 acts as both master and slave to multiple other master and slaves.



ESP32: Getting Board MAC

To communicate via ESP-NOW, you need to know the MAC Address of the ESP32 receiver. That's how you know to which device you'll send the data to. The following displays the code. A link to it is found on the next slide. No code is required in the loop (still need the loop function). Make sure to have your esp32 API (Board Manager) set to 3.x for this to work.

```
#include <WiFi.h>
#include <esp_wifi.h>

void readMacAddress(){
  uint8_t baseMac[6];
  esp_err_t ret = esp_wifi_get_mac(WIFI_IF_STA, baseMac);
  if (ret == ESP_OK) {
    Serial.printf("%02x:%02x:%02x:%02x:%02x:%02x\n",
                  baseMac[0], baseMac[1], baseMac[2],
                  baseMac[3], baseMac[4], baseMac[5]);
  } else {
    Serial.println("Failed to read MAC address");
  }
}

void setup(){
  Serial.begin(115200);

  WiFi.mode(WIFI_STA);
  WiFi.STA.begin();

  Serial.print("[DEFAULT] ESP32 Board MAC Address: ");
  readMacAddress();
}
```

Code to get MAC Addresses

https://raw.githubusercontent.com/RuiSantosdotme/Random-Nerd-Tutorials/master/Projects/ESP32/ESP32_Get_MAC_Address.ino

Upload this code. When done uploading press the reset button and look at the Serial Monitor.

Copy the MAC address for future reference.

ESP-NOW One-way Point to Point Communication

To get you started with ESP-NOW wireless communication, we'll build a simple project that shows how to send a message from one ESP32 to another. One ESP32 will be the "sender/master" and the other ESP32 will be the "receiver/slave".

Sketches

Sender:

https://raw.githubusercontent.com/RuiSantosdotme/Random-Nerd-Tutorials/master/Projects/ESP32/ESP_NOW/ESP_NOW_Sender.ino

Receiver:

https://raw.githubusercontent.com/RuiSantosdotme/Random-Nerd-Tutorials/master/Projects/ESP32/ESP_NOW/ESP_NOW_Receiver.ino

Test the Code

Designate one esp32 as the receiver and the other as the sender. In the sender add the MAC address of the receiver.

After uploading the code take a look at the Serial Monitor of each.

Sample Circuits-Sender

On the Sender ESP32 add a push button to IO4. In the setup() function write:

```
pinMode(4, INPUT_PULLDOWN);
```

The esp32 has internal pulldown resistors so you don't have to add these yourself.

Just connect button to power and to IO4.

Add in a variable to store the state of the button press as follows:

```
int value=0;
```

In the loop() function erase the line myData.b=random(1,20); and replace with

```
value=digitalRead(4);

myData.b=value;
```

When the button is pressed we update the myData.b to reflect its state i.e. 1 or 0 and this will be sent to the receiver. We can speed up this update by reducing the default delay(2000) to delay(100)

Sample Circuit-Receiver

Add in an LED with a resistor to pin 4 of the receiver esp32c6.

In the setup() function add the code:

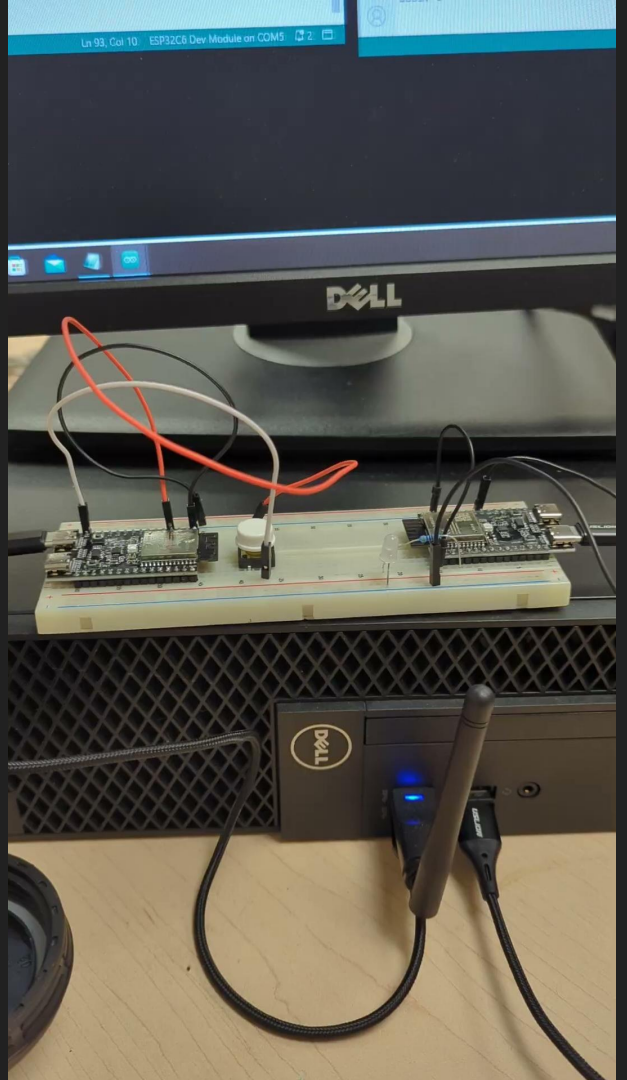
```
pinMode(4,OUTPUT);
```

In the onDataRecv() function add the following code(after the 1st line-memcpy…)

```
if(myData.b==1)
        digitalWrite(4,HIGH);
else
        digitalWrite(4,LOW);
```

Test

Upload the code to both esp32 microcontrollers and then press the button on the sender. The LED should light on the receiver. You have now successfully create a wireless circuit! The possibilities are endless!



Questions and Exercises:

1. Why is there no need for code in the loop() function of the receiver (you still need the loop function)?

ESP-NOW communication on the receiver is handled asynchronously via callback functions (like onDataRecv). The loop() function is still required syntactically in Arduino, but no code is needed inside it unless additional behavior is required. All receiving logic is triggered automatically when data arrives.

2. How could you send more integers at the same time from the sender to the receiver?

You can modify the struct_message structure to include additional integer fields.

3. What might you change/add in the struct_message structure if you were sending to the receiver data from multiple sensors?

```
typedef struct struct_message {
  float temperature;
  float humidity;
  int lightLevel;
} struct_message myData;
```

Assignment:

In your toolkit you should have a purple PCB board with headers for your ESP32 Dev Module. Insert your ESP32 Dev Module into these headers. The LEDs are attached to pins 2,4, 32 and 33. The Piezo Buzzer is attached to pin 15. Working with another group create a program so that the receiver will turn on LED at pin 33 when the random number sent over is 8, pin 32 LED when the number is 18, LED 4 at pin when the number is 7 and LED at pin 2 when the number is 10. When the number sent is 13, play the piezo buzzer for 2 seconds and then turn off all the LEDs. When the touch pin 12 is touched on the sender, flash all the LEDs and buzz the piezo ten times on the receiver. To read inputs from the digital touch pins use the touchRead(GPIO) function. Test the results of this function in your serial monitor first. No need to use pinMode when using the touchRead function for capacitive touch inputs.

/30 TI

Video Example