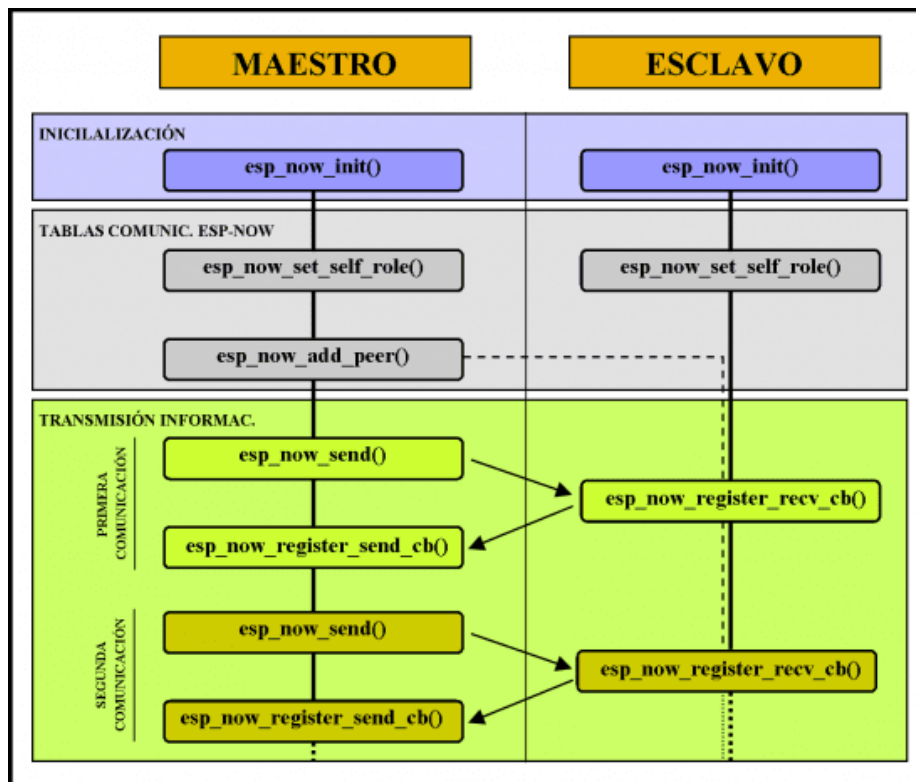


NodeMCU: 04_ESPNOW (Comunicación maestro-esclavo)

agrportfolioeducativo.blogspot.com/2020/03/nodemcu-04espnow-comunicacion.html



Intentando la comunicación ESP-NOW, me propongo hacer el esquema que propone Dani No en su web de **esploradores**. Intentaremos la conexión master + esclavo para después poder intentar hacer la master+esclavo en ambos dispositivos

Primero, averiguamos las MACS.

Según Dani No, cada dispositivo posee dos MACs distintas. Una para las comunicaciones como AP (AP MAC) y otra como station (STA MAC). Para averiguarlas escribimos el siguiente programa:

```
#include <ESP8266WiFi.h>

void setup() {
  Serial.begin(115200); Serial.println();Serial.println();
  Serial.print("AP MAC: "); Serial.println(WiFi.softAPmacAddress()); // MAC como AP
  Serial.print("STA MAC: "); Serial.println(WiFi.macAddress()); // MAC como estacion-station
}

void loop() {}
```

[view raw NodeMCU_averiguar_macs.ino](#) hosted with ♥ by [GitHub](#)

En mi caso, salen estos resultados:

Master	AP MAC: EE:FA:BC:C5:AC:AF
	STA MAC: EC:FA:BC:C5:AC:AF

Slave	AP MAC: A6:CF:12:DF:5A:6B
	STA MAC: A4:CF:12:DF:5A:6B

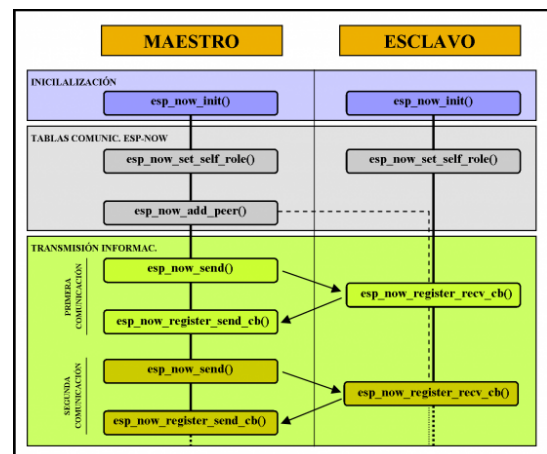
Tal como dicen en la web de Dani No,

La dirección MAC que debemos asignar a un par, depende de la conexión WiFi que tenga establecida y su papel en el par de comunicación. En la siguiente tabla se resumen las posibles combinaciones y la MAC a elegir:

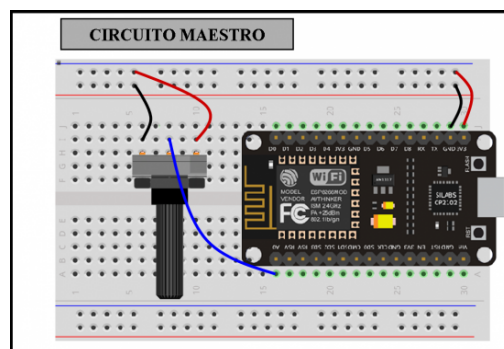
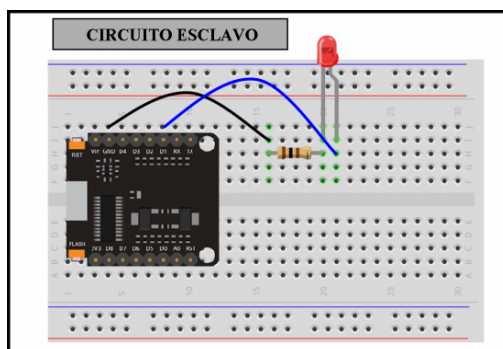
		TIPO DE CONEXIÓN WiFi			
		SIN CONEXIÓN	AP	STA	AP+STA
PAPEL	MAESTRO	STA MAC	AP MAC	STA MAC*	AP MAC
	ESCLAVO	STA MAC	AP MAC*	STA MAC	STA MAC*

*Conexiones WiFi sugeridas por [Spressif](#), en el caso de que se utilicen. (Maestro-STA y Esclavo-AP o AP+STA)

¿Cómo se establece una comunicación? En la web se nos hace un resumen de las funciones a incorporar, divididas en grupos. Podemos verlas mejor en ella. Simplemente me refiero ahora con el esquema:



Ejemplo



Script del maestro

/*

Ejemplo de comunicación ESP-NOW por Dani No www.esploradores.com

ESTE SKETCH CORRESPONDE AL PAR MAESTRO

El sketch permite el envío de dos datos mediante la comunicación ESP-NOW a un par esclavo:

- En la variable -potenciometro- se envía el dato de la lectura de la entrada analógica del ESP. Se leen y envían -valores entre 0 y 1023- que se hacen variar mediante un potenciómetro y se utilizarán para regular la intensidad del LED conectado en el circuito esclavo.

- En la variable -tiempo- se envía el dato del milisegundo en el que el que se envían los datos al circuito esclavo.

!!!IMPORTANTE!!!

Para hacer funcionar la comunicación ESP-NOW si se tiene instalado en Arduino como gestor de tarjetas el: esp8266 by ESP8266 Community versión 2.3.0 (Se puede comprobar buscándolo en: Herramientas->Placa:"NodeMCU 1.0(ESP-12E Module)"->Gestor de tarjetas..., es necesario editar el fichero: ~/Library/Arduino15/packages/esp8266/hardware/esp8266/2.1.0/platform.txt, buscar "compiler.c.elf.libs", y añadir al final de la línea "-lespnow".

*/

```
#include <ESP8266WiFi.h>
```

```
extern "C" {
```

```
#include <espnow.h>
```

```
}
```

```
/**ESTRUCTURA DE LOS DATOS TRANSMITIDOS MAESTRO/ESCLAVO**//
```

```
//Se de establecer IGUAL en el par esclavo
```

```
struct ESTRUCTURA_DATOS {
```

```
uint16_t potenciometro;
```

```
uint32_t tiempo;
```

```
};
```

```
void setup() {
```

```
/**INICIALIZACIÓN DEL PUERTO SERIE**//
```

```
Serial.begin(115200); Serial.println();Serial.println();
```

```
/**INICIALIZACIÓN DEL PROTOCOLO ESP-NOW**/
```

```
if (esp_now_init()!=0) {
```

```
Serial.println("*** ESP_Now init failed");
```

```
ESP.restart();
```

```
delay(1);
```

```
}
```

```
/**DATOS DE LAS MAC (Access Point y Station) del ESP**/
```

```
Serial.print("Access Point MAC de este ESP: "); Serial.println(WiFi.softAPmacAddress());
```

```
Serial.print("Station MAC de este ESP: "); Serial.println(WiFi.macAddress());
```

```
/**DECLARACIÓN DEL PAPEL DEL DISPOSITIVO ESP EN LA COMUNICACIÓN**/
```

```
//0=OCIOSO, 1=MAESTRO, 2=ESCLAVO y 3=MAESTRO+ESCLAVO
```

```
esp_now_set_self_role(1);
```

```
/**EMPAREJAMIENTO CON EL ESCLAVO**/
```

```
// Dirección MAC del ESP con el que se empareja (esclavo)
```

```
// Se debe introducir la STA MAC correspondiente
```

```
uint8_t mac_addr[6] = {0xA4, 0xCF, 0x12, 0xDF, 0x5A, 0x6B}; // STA MAC esclavo
```

```
uint8_t role=2;
```

```
uint8_t channel=3;
```

```
uint8_t key[0]={}; //no hay clave
```

```
//uint8_t key[16] = {0,255,1,1,1,1,1,1,1,1,1,1,1,1,1,1};
```

```
uint8_t key_len=sizeof(key);
```

```
Serial.print("Tamaño de *key: "); Serial.println(key_len);
```

```
esp_now_add_peer(mac_addr,role,channel,key,key_len);
```

```
}
```

```
void loop() {
```

```
/**DATOS A ENVIAR**/
```

```
ESTRUCTURA_DATOS ED;
```

```
ED.potenciometro = analogRead(A0);
```

```
Serial.print("Dato potenciometro: "); Serial.print(ED.potenciometro);
```

```
delay(20);
```

```
ED.tiempo = millis();
```

```
Serial.print(". Dato tiempo: "); Serial.print(ED.tiempo);
```

```
/**ENVÍO DE LOS DATOS**/
```

```
//uint8_t *da=NULL; //NULL envía los datos a todos los ESP con los que está emparejado
```

```
uint8_t da[6] = {0xA4, 0xCF, 0x12, 0xDF, 0x5A, 0x6B}; // ¿mismos datos que STA MAC?
```

```
uint8_t data[sizeof(ED)]; memcpy(data, &ED, sizeof(ED));
```

```
uint8_t len = sizeof(data);
```

```
esp_now_send(da, data, len);
```

```
delay(1); //Si se pierden datos en la recepción se debe subir este valor
```

```
/**VERIFICACIÓN DE LA RECEPCIÓN CORRECTA DE LOS DATOS POR EL ESCLAVO**/
```

```
esp_now_register_send_cb([](uint8_t* mac, uint8_t status) {
```

```
char MACesclavo[6];
```

```
sprintf(MACesclavo,"%02X:%02X:%02X:%02X:%02X:%02X",mac[0],mac[1],mac[2],mac[3],mac[4],mac[5]);
```

```
Serial.print(". Enviado a ESP MAC: "); Serial.print(MACesclavo);
```

```
Serial.print(". Recepcion (0=OK - 1=ERROR): "); Serial.println(status);
```

```
});
```

```
}
```

view raw [NODEMCU ESPNOW par maestro.ino](#) hosted with ♥ by [GitHub](#)

Script del esclavo

```
/*
```

```
Ejemplo de comunicación ESP-NOW por Dani No www.esploradores.com
```

```
***ESTE SKETCH CORRESPONDE AL PAR ESCLAVO***
```

```
El sketch permite la recepción de una comunicación vía ESP-NOW de dos datos enviados
```

```
por el par maestro.
```

```
- En la variable -potenciometro- se envía el dato de la lectura de la entrada
```

```
analógica del circuito maestro -valores entre 0 y 1023-, que se utilizarán
```

```
para regular la intensidad del LED conectado en el circuito esclavo.
```

```
- En la variable -tiempo- se envía el dato del milisegundo en el que el circuito maestro
```

```
realiza la comunicación con el circuito esclavo.
```

```
¡¡¡¡¡IMPORTANTE!!!
```

Para hacer funcionar la comunicación ESP-NOW si se tiene instalado en Arduino como gestor de tarjetas el: esp8266 by ESP8266 Community versión 2.3.0 (Se puede comprobar buscándolo en: Herramientas->Placa:"NodeMCU 1.0(ESP-12E Module)"->Gestor de tarjetas..., es necesario editar el fichero: ~/Library/Arduino15/packages/esp8266/hardware/esp8266/2.1.0/platform.txt, buscar "compiler.c.elf.libs", y añadir al final de la línea "-lespnow".

*/

```
#include <ESP8266WiFi.h>
```

```
extern "C" {
```

```
#include <espnow.h>
```

```
}
```

```
/**ESTRUCTURA DE LOS DATOS TRANSMITIDOS MAESTRO/ESCLAVO**//
```

```
//Se de establecer IGUAL en el par maestro
```

```
struct ESTRUCTURA_DATOS {
```

```
uint16_t potenciómetro = 0;
```

```
uint32_t tiempo = 0;
```

```
};
```

```
/**PIN de conexión del LED a regular con el potenciómetro del ESP MAESTRO**//
```

```
int PinLED = 5; //Pin D1
```

```
void setup() {
```

```
/**INICIALIZACIÓN DEL PUERTO SERIE**//
```

```
Serial.begin(115200); Serial.println();
```

```
/**INICIALIZACIÓN DEL PROTOCOLO ESP-NOW**//
```

```
if (esp_now_init()!=0) {
```

```
Serial.println("Protocolo ESP-NOW no inicializado...");
```

```
ESP.restart();
```

```
delay(1);
```

```
}
```

```
/**DATOS DE LAS MAC (Access Point y Station) del ESP**//
```

```
Serial.print("AP MAC: "); Serial.println(WiFi.softAPmacAddress());
```

```
Serial.print("STA MAC: "); Serial.println(WiFi.macAddress());
```

```
/**DECLARACIÓN DEL PAPEL DEL DISPOSITIVO ESP EN LA COMUNICACIÓN**/
```

```
//0=OCIOSO, 1=MAESTRO, 2=ESCLAVO y 3=MAESTRO+ESCLAVO
```

```
esp_now_set_self_role(2);
```

```
/**DECLARACIÓN del PinLED como SALIDA**/
```

```
pinMode(PinLED, OUTPUT);
```

```
}
```

```
void loop() {
```

```
/**RECEPCIÓN DE LA COMUNICACIÓN ESP-NOW**/
```

```
esp_now_register_recv_cb([(uint8_t *mac, uint8_t *data, uint8_t len) {
```

```
char MACmaestro[6];
```

```
sprintf(MACmaestro,  
"%02X:%02X:%02X:%02X:%02X:%02X",mac[0],mac[1],mac[2],mac[3],mac[4],mac[5]);
```

```
Serial.print("Recepcion desde ESP MAC: "); Serial.print(MACmaestro);
```

```
ESTRUCTURA_DATOS ED;
```

```
memcpy(&ED, data, sizeof(ED));
```

```
Serial.print(". Dato potenciometro: "); Serial.print(ED.potenciometro);
```

```
Serial.print(". Dato tiempo: "); Serial.println(ED.tiempo);
```

```
analogWrite(PinLED,ED.potenciometro);
```

```
});
```

```
}
```

[view raw NODEMCU_ESPNOW_par_esclavo.ino](#) hosted with ♥ by [GitHub](#)

=====

Bibliografía

=====

<https://www.esploradores.com/practica-6-conexion-esp-now/>

<https://randomnerdtutorials.com/esp-now-two-way-communication-esp32/>