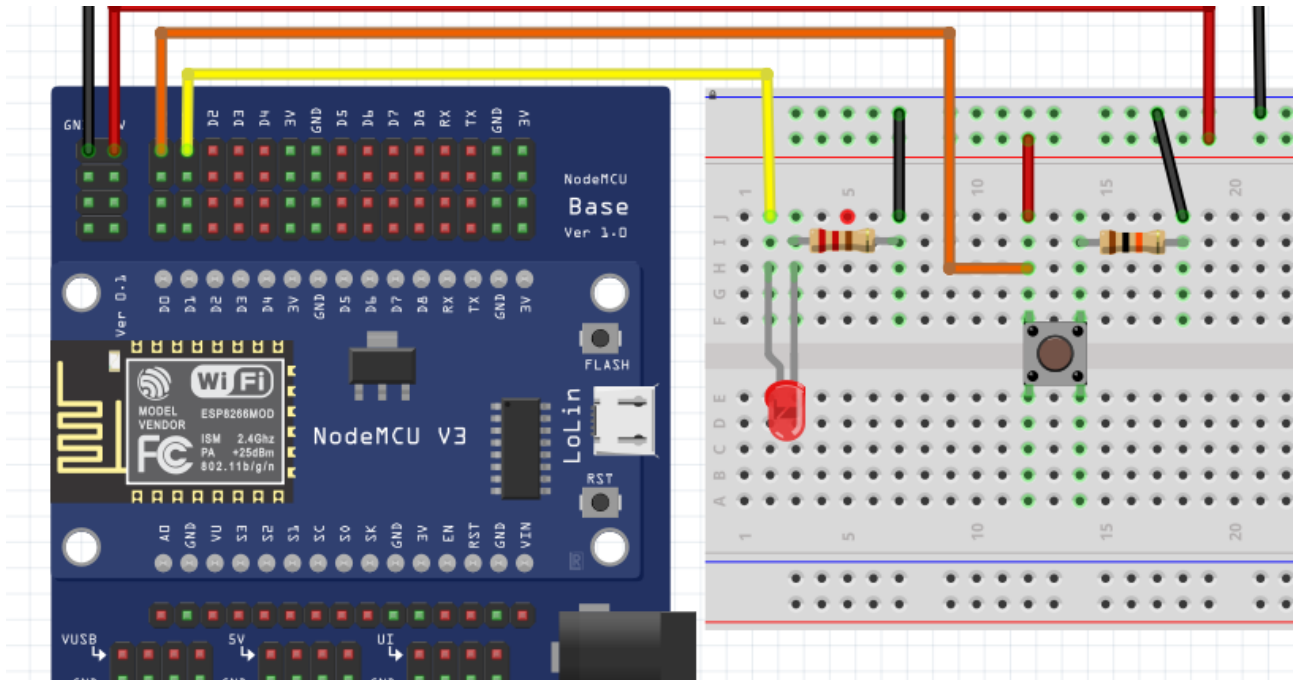


NodeMCU: 08_ESPNOW (Comunicación bidireccional) Mejorando el emisor: FASE 1

agrportfolioeducativo.blogspot.com/2020/04/nodemcu-08espnow-comunicacion.html



ESQUEMA

Constaría del siguiente esquema electrónico: un simple pulsador y un led que parpadea a frecuencias distintas.

El programa mejorado para el botón del pánico, respecto al emisor es el siguiente:

PROGRAMA

```
/* Este ejemplo intenta recopilar lo aprendido de esquemas anteriores
```

```
* y comunicará dos NodeMCU station "hablando" entre ellos.
```

```
* La acción en uno, establecerá una reacción en el otro.
```

```
*/
```

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

```
extern "C" {
```

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

```
}
```

```
/**ESTRUCTURA DE LOS DATOS TRANSMITIDOS ENTRE LAS UNIDADES**/
```

```
// Se establecerá IGUAL en todas
```

```
struct ESTRUCTURA_DATOS {
```

```
int estado; // 0 --> apagado , 1 --> operativo, 2 -->activado
```

```
String estacion;
```

```
};
```

```
// variables de elementos hardware
```

```
int entrada = D0; // pulsador
```

```
int salida = D1; // LED
```

```
int pulsado = 0;
```

```
int pulsadoAntes = 0;
```

```
long tiempo = 0; // variable tiempo
```

```
int cadencia=500; // Cadencia de parpadeo
```

```
// 500 -> medio segundo
```

```
int desbloqueo=0; // veces que es necesario pulsar
```

```
int n = 1; // períodos de apagados (tren de pulsos)
```

```
// el botón para desbloquear la alarma
```

```
ESTRUCTURA_DATOS ED;
```

```
// *****
```

```
// funciones
```

```
// *****
```

```
// *****
```

```
// SETUP
```

```
// *****
```

```
void setup() {
```

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

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

```
  /*** pinMode ***
```

```
  pinMode(entrada,INPUT);
```

```
  pinMode(salida,OUTPUT);
```

```
  /***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(3); // Sería como maestro y esclavo a la vez

/**EMPAREJAMIENTO CON EL ESCLAVO: CENTRAL**/

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

// Se debe introducir la STA MAC correspondiente

uint8_t mac_addr[6] = {0xEC, 0xFA, 0xBC, 0xC5, 0xAC, 0xAF}; // STA MAC esclavo

// Esta es la MAC de la CENTRAL

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);

ED.estado = 1; // Si termina el SETUP está operativo

ED.estacion="A";

}

// *****

// LOOP

// *****

void loop() {

// -----

// Acciones del programa que no suponen envío de datos

```

```

// -----
tiempo = millis();

pulsado = digitalRead(entrada);

// A) caso que ED.estado=1 (operativo y se pulse)
if (pulsado==1 & pulsadoAntes==0 & ED.estado==1) {
    ED.estado = 2; // Activo la alarma
} else if (pulsado==1 & pulsadoAntes==0 & ED.estado==2 & desbloqueo<=2) {
    // O en caso que pulse el estado sea 2 y no haya desbloqueo
    desbloqueo+=1; // Suma 1 a desbloqueo
}

// B) Llega al tercer estado de desbloqueo
if (desbloqueo==3) { //
    desbloqueo=0;
    ED.estado=1; // inactiva la alarma
}

// C) Según el caso cambia el parpadeo
// cadencia = 19900-9900*ED.estado;
n=8*ED.estado+18; // Si estado 1, n=10; si 2, n=2

// Estado 1 --> Cada 10 segundos; Estado 2 --> cada 0.1 seg
digitalWrite(D1,((ED.estado*tiempo/cadencia)%n==0)); // Un parpadeo cada n veces
Serial.print("Alarma: (1-OFF,2-ON): " + (String) ED.estado + " // desbloqueo: " + (String) desbloqueo);
delay(10); // pequeño delay...

// -----

// ENVÍO: como Master

// -----

/**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] = {0xEC, 0xFA, 0xBC, 0xC5, 0xAC, 0xAF}; // ¿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

// *** CALLBACK de datos ENVIADOS ***

// *** Verificamos que los datos se han enviado....

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);

});

// -----

// -----

// Recepción: como Esclavo

// -----

esp_now_register_rcv_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 ED2;

memcpy(&ED2, data, sizeof(ED2));

Serial.print(" # Recibido de "+ ED2.estacion + " // Estoy --> "+ (String) ED2.estado);

// digitalWrite(salida,ED2.estado);

});

// -----

pulsadoAntes=pulsado; // Importante para reconocer el flanco de subida del pulsador

Serial.println();

}

```

[view raw](#) [NodeMCU BOTON PANICO EMISOR.ino](#) hosted with ♥ by [GitHub](#)

El modo de funcionamiento es el siguiente: tenemos dentro de la variable estructurada ED, los dos siguientes valores **ED.estado** y **ED.estacion**. Estos datos se envían constantemente a la central. El valor de ED.estacion es "A".

ED.estado=1 significa estación con alarma desactivada. Al pulsar el botón en Do (detecta flanco de subida) cambia el estado a alarma activada, **ED.estado=2**.

Para desactivar la alarma y volver al estado anterior, debo pulsar tres veces el botón. La variable desbloqueo cuanta de 0 a 2, y, cuando llega a 3 pasa a 0 y desactiva la alarma.

Esto significa que el usuario, al pulsar una vez, activará la alarma. Y necesita pulsarla tres veces para desactivarla.

Por otra parte, la cadencia de parpadeo del piloto de cada estación es lenta (1 parpadeo cada 10 segundos) cuando la alarma está desactivada y rápido si la alarma está activada.

En un primer momento, la recepción de datos de la CENTRAL simplemente pone en el monitor serie el valor recibido. En otra fase, debería, a lo mejor, controlar el estado de la estación.