

```

1  # define RELAISPIN 16
2  # define POTI A0
3
4  # include < ESP8266WiFi .h >
5  # include < espnow .h >
6
7  uint8_t newMacAddress [] = { 0 x96 , 0 x3B , 0 xC7 , 0 x34 , 0 x69 , 0 x02 };
8  uint8_t receiverAddress [] = { 0 x94 , 0 x3C , 0 xC6 , 0 x33 , 0 x68 , 0 x01 };
9
10 // Functions
11 ///////////////////////////////////////////////////////////////////
12 void onoff ( uint8_t* macAddr , uint8_t* incomingData , uint8_t len ) { // Nimmt empfangene Nachricht als Argument schaltet je nach Inhalt
13   das Relais HIGH oder LOW
14   if ( incomingData [0] == 'E' ) {
15     digitalWrite ( RELAISPIN , HIGH );
16   }
17   if ( incomingData [0] == 'A' ) {
18     digitalWrite ( RELAISPIN , LOW );
19   }
20 }
21
22 void peercheck ( uint8_t* mac_addr , uint8_t sendStatus ) { // ueberprueft , ob der Teilnehmer erreicht werden konnte . Schaltet das Relais ab
23   , wenn dies nicht der Fall ist
24   if ( sendStatus != 0 ) {
25     digitalWrite ( RELAISPIN , LOW );
26   }
27 }
28 //
29 ///////////////////////////////////////////////////////////////////
30
31 void setup () {
32   pinMode ( RELAISPIN , OUTPUT ); // Definieren von Relaispin als Output
33   pinMode ( LED_BUILTIN , OUTPUT ); // Definieren von eingebauter LED als Output
34   digitalWrite ( RELAISPIN , LOW ); // Zusperrern im Fall eines Stromausfalls und Wiederanlaufs
35   WiFi . mode ( WIFI_STA ); // Initialisieren des Boards als Wifi - Station
36   wifi_set_macaddr ( STATION_IF , newMacAddress ); // Vergeben einer MAC an das Board
37   WiFi . disconnect (); // Deinitialisieren des Boards als Standard - Wifi - Device
38
39   if ( esp_now_init () != 0 ) { // Initialisieren des Boards als ESPnow - Wifi - Device und Abfrage ob erfolgreich durch Ansteuern der
40     eingebauten LED
41     digitalWrite ( LED_BUILTIN , HIGH );
42   } else {
43     digitalWrite ( LED_BUILTIN , LOW );
44   }
45 }

```

```

40  esp_now_add_peer ( receiverAddress ,    ESP_NOW_ROLE_COMBO 0 , NULL , 0); // ESP32 als Kommunikationspartner hinzufügen . Dazu werden MAC ,
41      Rolle , Channel , Schlüssel und Schlüssellaenge uebergeben
42
43  esp_now_set_self_role ( ESP_NOW_ROLE_COMBO ); // ESP8266 zum Transceiver machen
44
45  esp_now_register_rcv_cb ( onoff ); // Fuehrt bei Empfangen einer Nachricht die Interrupt - Funktion onoff aus . Definiert in ESPnow
46      erhaelt diese als Argumente die Sender - MAC , die Nachricht und die Laenge der Nachricht
47  esp_now_register_send_cb ( peercheck ); // Fuehrt bei Senden einer Nachricht die Interrupt - Funktion peercheck aus . Definiert in ESPnow
48      erhaelt diese als Argumente die Empfaenger - MAC , und den Erfolgsstatus des Versendens
49  }
50
51  void loop () {
52      uint8_t potiwert = analogRead ( POT1 ) / 5; // Einlesen des Potiwertes an Analogpin und runterskalieren des Wertes
53
54      esp_now_send ( receiverAddress , ( uint8_t *) & potiwert , sizeof ( potiwert )); // Versenden des Potiwertes
55
56      delay (200) ;
57  }

```

Anhang B