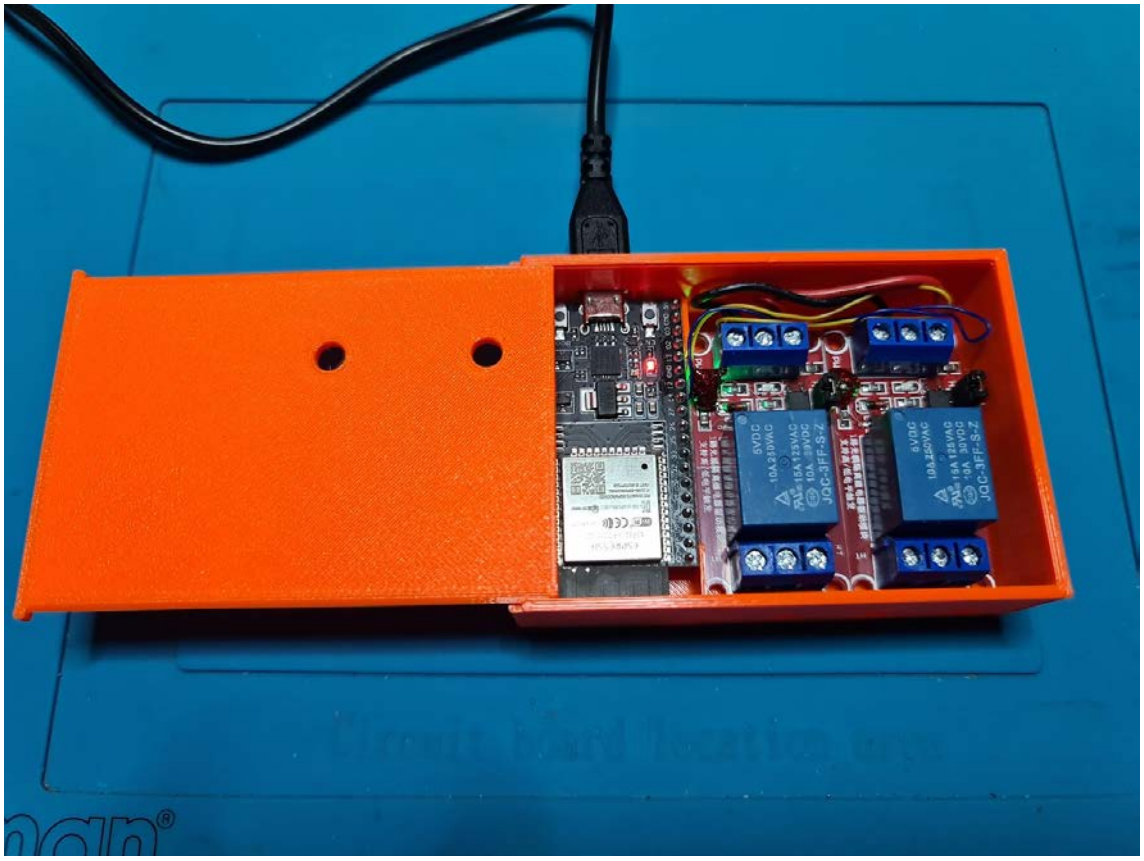
Control de dos relés por Telegram



ESP32 que usé https://www.az-delivery.de/es/products/esp-32-dev-kit-c-v4

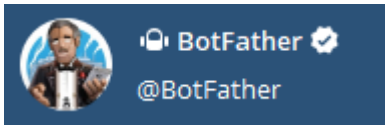
Pero servirá cualquier otro modelo.

Enlace al Relé que yo usé, pero pueden servir muchos otros https://tinyurl.com/2cppuece

El esquema de conexiones es muy simple, la alimentación de los módulos Relé la tomo de los 5v del ESP32 y GND y el control de los pines 12 y 14.

La salida de los Relés según convenga, normalmente Línea al COM y salida por NO

Hay que crear un Bot de Telegram para el control https://t.me/BotFather



El propio Bot te da ayuda de su manejo, pero en el enlace del programa original de Rui Santos hay mucha información al respecto. https://tinyurl.com/243d4237

Si te interesa la caja aquí los STL https://www.thingiverse.com/thing:6741637

Este es el programa:

```
/***
 CONTROLAR DOS RELES POR TELEGRAM JCS666
 Codigo modificado del original de Rui Santos
 https://randomnerdtutorials.com/telegram-control-esp32-esp8266-nodemcu-outputs/#more-97812
***/
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
// Reemplazar con los datos de tu red wifi
#define WIFI_SSID "***SSID***"
#define WIFI_PASSWORD "***CLAVE***"
//Token de Telegram BOT se obteniene desde Botfather en telegram
#define BOT_TOKEN "***TOKEN***"
const unsigned long tiempo = 1000;  //tiempo medio entre mensajes de escaneo
WiFiClientSecure secured_client;
UniversalTelegramBot bot(BOT_TOKEN, secured_client);
unsigned long tiempoAnterior;  //última vez que se realizó el análisis de mensajes
//Definir pines y estado
const int Rele1 = 12;
const int Rele2 = 14;
int estadoRele1 = 0;
int estadoRele2 = 0;
```

```
int inicio = 1;
String chat_id;
#define ID_Chat "***12345678***"  //ID_Chat se obtiene de telegram
void mensajesNuevos(int numerosMensajes) {
  for (int i = 0; i < numerosMensajes; i++) {
    String chat_id = bot.messages[i].chat_id;
    String text = bot.messages[i].text;
    //////////R 1 en el pin 12//////
    if (text == "R1on") {
      digitalWrite(Rele1, HIGH);  //
      estadoRele1 = 1;
      bot.sendMessage(chat_id, "R 1 encendido", "");
    }


    if (text == "R1off") {
      estadoRele1 = 0;
      digitalWrite(Rele1, LOW);  //
      bot.sendMessage(chat_id, "R 1 apagado", "");
    }
    //////////R 2 en el pin 14//////
    if (text == "R2on") {
      digitalWrite(Rele2, HIGH);
      estadoRele2 = 1;
      bot.sendMessage(chat_id, "R 2 encendido", "");
    }


    if (text == "R2off") {
      estadoRele2 = 0;
      digitalWrite(Rele2, LOW);
```

```
  bot.sendMessage(chat_id, "R 2 apagado", "");
 }


 //////// Estado de los Reles  ///////


 if (text == "Estado") {
  ////Estado R 1////
  if (estadoRele1) {
   bot.sendMessage(chat_id, "R 1 encendido", "");
  } else {
   bot.sendMessage(chat_id, "R 1 apagado", "");
  }
  ////Estado R 2////
  if (estadoRele2) {
   bot.sendMessage(chat_id, "R 2 encendido", "");
  } else {
   bot.sendMessage(chat_id, "R 2 apagado", "");
  }
 }


 if (text == "Ayuda") {
  String ayuda = "Bienvenido al sistema de control con Esp32.\n\n";
  ayuda += "Estas son las opciones.\n\n";
  ayuda += "R1on: para encender R 1 \n";
  ayuda += "R1off: para apagar R 1 \n";
  ayuda += "R2on: para encender R 2 \n";
  ayuda += "R2off: para apagar R 2 \n";
  ayuda += "Estado : devuelve el estado actual de Rx\n";
```

```
    ayuda += "Ayuda: Imprime este menú \n";
    ayuda += "Recuerda el sistema distingue entre mayuculas y minusculas \n";
    bot.sendMessage(chat_id, ayuda, "");
   }
  }
}



void setup() {
  Serial.begin(115200);
  pinMode(Rele1, OUTPUT);    //inicializar  pin 12 digital como salida.
  pinMode(Rele2, OUTPUT);    //inicializar pin 14 digital como salida.
  digitalWrite(Rele1, LOW);  //y reles apagados.
  digitalWrite(Rele2, LOW);
  // Intenta conectarse a la red wifi
  Serial.print("Conectando a la red ");
  Serial.print(WIFI_SSID);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  secured_client.setCACert(TELEGRAM_CERTIFICATE_ROOT);  //Agregar certificado raíz para
api.telegram.org
  while (WiFi.status() != WL_CONNECTED) {
   Serial.print(".");
   delay(500);
  }
  Serial.print("\nConectado a la red wifi. Dirección IP: ");
  Serial.println(WiFi.localIP());
  if (inicio == 1) {
   Serial.println("Sistema preparado");
   bot.sendMessage(ID_Chat, "Sistema preparado!!!, escribe Ayuda para ver las opciones", "");
//Enviamos un mensaje a telegram para informar que el sistema está listo
   inicio = 0;
  }
```

```
}



void loop() {
  //Verifica si hay datos nuevos en telegram cada 1 segundo
  if (millis() - tiempoAnterior > tiempo) {
    int numerosMensajes = bot.getUpdates(bot.last_message_received + 1);



    while (numerosMensajes) {
      Serial.println("Comando recibido");
      mensajesNuevos(numerosMensajes);
      numerosMensajes = bot.getUpdates(bot.last_message_received + 1);
    }



    tiempoAnterior = millis();
  }
}
```