

Control de rele con Telegram

En este proyecto se trata de activar y desactivar un rele (puede controlar cargas en continua y alternar hasta 10A), mediante ESP8266 -01 y Telegram con los comandos **/on** y **/off**.

Crear Bot en Telegram:

Ver pagina oficial: <https://core.telegram.org/bots#6-botfather>

En resumen una vez descargada la APP de Telegram y creada nuestra cuenta:

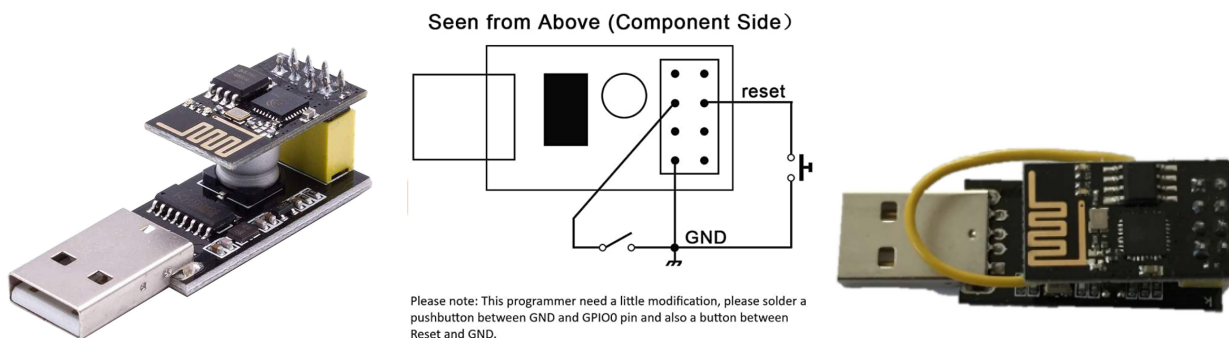
- Buscar BotFather. Con **/start**, nos muestra los comandos utilizables.
- Mandar mensaje **/newbot**
- Nos pide nombre del Bot (ejemplo esp32Cam)
- Nos pide usuario del Bot, que debe terminar en bot (ejemplo **j1234_bot**).
- Nos crea el bot y no muestra el token para acceder mediante HTTP
Api: **12732xxxxx:xxxxxxxxxxW9pFjxxxxxxxxgx4** (que utilizaremos en el scrip)
- Para conocer el chatId buscar Bot **IDBot** y mandar mensaje **/getid**
- Añadimos el Bot creado **t.me/j1234_bot**

Para programar ESP -01

Chip de interfaz serial CH340

comprar

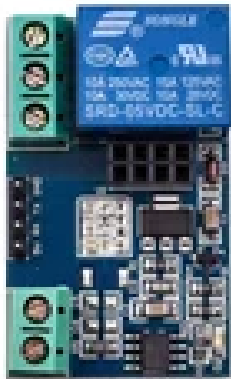
https://www.amazon.es/azdelivery-ESP8266-ESP-adaptador-Arduino-Raspberry/dp/B078J7LDLY/ref=sr_1_6?_mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&dchild=1&keywords=esp01+relay&qid=1607198384&sr=8-6



Vamos a utilizar dos tipos de placas:

- Rele de AZ-DELIVERY

Comprar: <https://www.az-delivery.de/products/esp8266-01s-mit-relais?ls=de>



La placa hay que alimentarla con 5vcc.

Puede soportar cargas de hasta 10A a 220Vac.

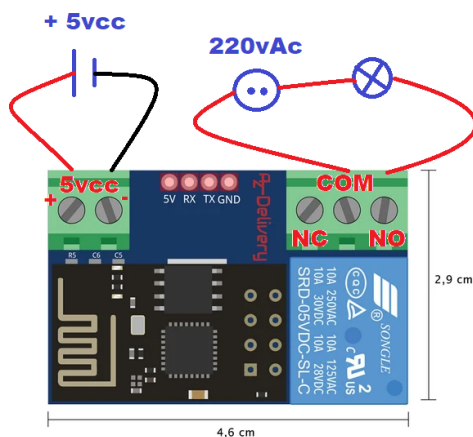
Para activar y desactivar el rele se utiliza el puerto serie con los siguientes comandos:

Serial.begin(9600); // No cambiar ya que solo funciona a 9600baudios

```
/on:      byte open[] = {0xA0, 0x01, 0x01, 0xA2};  
          Serial.write(open, sizeof(open));
```

```
/off      byte close[] = {0xA0, 0x01, 0x00, 0xA1};  
          Serial.write(close, sizeof(close));
```

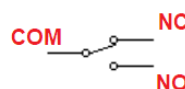
Consumo off: 90mA on: 160mA



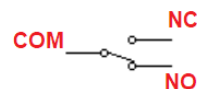
Cuando el rele esta desactivado, hay continuidad entre NC y COM

Cuando el rele se activa hay continuidad entre NO y COM

Rele desactivado



Rele activado



ESP8266_rele_az-delivery_telegram.ino

```
/*
 * jarp 2020
 * Control placa rele desde Telegram
 * az-delivery
 * Consumo off: 90mA on: 160mA
 *
 * apaga: /off
 * enciende: /on
 * Serial.begin(9600); // No cambiar ya que solo funciona a 9600baudios
 * ESP8266_rele_az-delivery_telegram
 */
#include <ESP8266WiFi.h>

const char * ssid = "xxxxxxxx"; // Actualizar dato
const char * password = "xxxxxxxx"; // Actualizar dato
const char* token = "xxxxxxxxxxxxxxxxxxxx"; // Actualizar dato
#include <AsyncTelegram.h>
AsyncTelegram myBot;

void setup(){
  Serial.begin(9600); // No cambiar ya que solo funciona a 9600baudios
  Serial.println("Starten");
  delay(10);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }

  Serial.println("IP: ");
  Serial.println(WiFi.localIP());
  //convierte Ip a cadena
  IPAddress ipAddress = WiFi.localIP();
  String miip =String(ipAddress[0]) + String(".") + String(ipAddress[1]) + String(".") +
String(ipAddress[2]) + String(".") + String(ipAddress[3]) ;

  // Establecer las propiedades del bot de Telegram
  myBot.setUpdateTime(1000);
  myBot.setTelegramToken(token);
  // Comprueba conexión de Telegram
  Serial.print("\nProbar conexión de Telegram... ");

  myBot.begin() ? Serial.println("OK") : Serial.println("No OK");
  Serial.print("Bot name: @");
  Serial.println(myBot.userName);
  TBMessage msg;
  msg.sender.id = xxxxxxxx; // Actualizar dato
  myBot.sendMessage(msg, "ESP32-Cam Conectado- IP:" + miip , "");
}

void loop(){
  TBMessage msg;
```

```

msg.sender.id = xxxxxxxx; // Actualizar dato
// si hay un mensaje entrante .....
if (myBot.getNewMessage(msg)) {
  Serial.print("Nuevo mensaje de chat_id: ");
  Serial.println(msg.sender.id);
  MessageType msgType = msg.messageType;

  if (msgType == MessageType){
    // Recibí un mensaje de texto
    Serial.println(msg.text);
    if (msg.text.equalsIgnoreCase("/on")){
      Serial.println("ON");
      delay(10);
      byte open[] = {0xA0, 0x01, 0x01, 0xA2};
      Serial.write(open, sizeof(open));
      myBot.sendMessage(msg, "Rele ON", "");
    }
  }
  if (msgType == MessageType){
    // Recibí un mensaje de texto
    Serial.println(msg.text);
    if (msg.text.equalsIgnoreCase("/off")){
      Serial.println("OFF");
      delay(10);
      byte close[] = {0xA0, 0x01, 0x00, 0xA1};
      Serial.write(close, sizeof(close));
      myBot.sendMessage(msg, "Rele OFF", "");
    }
  }
}
}
}
}

```

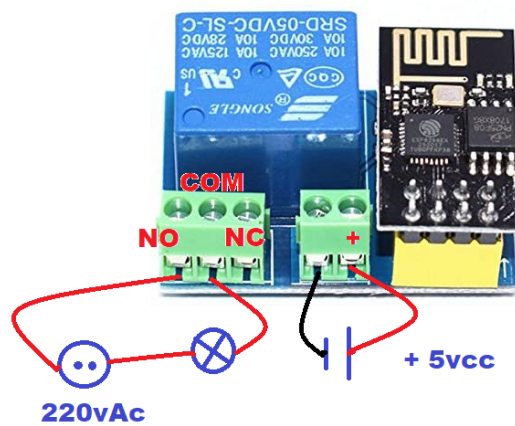
Mas información:

https://www.az-delivery.de/es/blogs/azdelivery-blog-fur-arduino-und-raspberry-pi/esp8266-01-relais-mit-website-steuern?_pos=2&_sid=3a11a6611&_ss=r

● Rele TB:IOTMCU

Comprar:

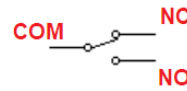
https://www.amazon.es/REFURBISHHOUSE-Interruptor-Inteligente-Aplicacion-inalambrico/dp/B07JKM517H/ref=sr_1_3?_mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&dchild=1&keywords=esp01+relay&qid=1607198384&sr=8-3



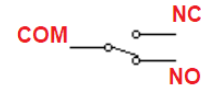
Cuando el rele esta desactivado, hay continuidad entre NC y COM

Cuando el rele se activa hay continuidad entre NO y COM

Rele desactivado



Rele activado



La placa hay que alimentarla con 5vcc
Puede soportar cargas de hasta 10A a 220Vac
Para activar y desactivar el rele se utiliza el GPIO0

```
#define RELAY 0 // rele conectado a GPIO0
GPIO0, Activa rele con LOW, desactiva rele con HIGH
```

```
/on:      digitalWrite(RELAY,HIGH);
```

```
/off      digitalWrite(RELAY,HIGH);;
```

Consumo Off: 90mA ON: 160mA

ESP8266_rele_telegram.ino

```
/* ESP8266_rele_telegram
 * jarp 2020
 * Control placa rele desde Telegram por GPIO0
 * Consumo off: 90mA on: 160mA
 *
 * apaga: /off
 * enciende: /on
 *
 */

#include <ESP8266WiFi.h>

const char * ssid = "xxxxxxxx"; // Actualizar dato
const char * password = "xxxxxxxx"; // Actualizar dato
const char* token = "xxxxxxxxxxxxxxxxxxxx"; // Actualizar dato
#include <AsyncTelegram.h>
AsyncTelegram myBot;

#define RELAY 0 // rele conectado a GPIO0, Activa con LOW, desactiva con HIGH
void setup(){
  Serial.begin(115200);
  pinMode(RELAY,OUTPUT);
  digitalWrite(RELAY, LOW); // Activa rele

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
  Serial.println("IP: ");
  Serial.println(WiFi.localIP());
  //convierte Ip a cadena
  IPAddress ipAddress = WiFi.localIP();
  String miip =String(ipAddress[0]) + String(".") + String(ipAddress[1]) + String(".") +
String(ipAddress[2]) + String(".") + String(ipAddress[3]) ;

  // Establecer las propiedades del bot de Telegram
  myBot.setUpdateTime(1000);
  myBot.setTelegramToken(token);
  // Comprueba conexión de Telegram
  Serial.print("\nProbar conexión de Telegram... ");

  myBot.begin() ? Serial.println("OK") : Serial.println("No OK");
  Serial.print("Bot name: @");
  Serial.println(myBot.userName);
  TBMessage msg;
  msg.sender.id = xxxxxxxx; // Actualizar dato
  myBot.sendMessage(msg, "ESP8266-rele Conectado - IP:" + miip , "");
}

void loop(){
  TBMessage msg;
  msg.sender.id = xxxxxxxx; // Actualizar dato
```

```

// si hay un mensaje entrante .....
if (myBot.getNewMessage(msg)) {
  Serial.print("Nuevo mensaje de chat_id: ");
  Serial.println(msg.sender.id);
  MessageType msgType = msg.messageType;

  if (msgType == MessageType){
    // Recibí un mensaje de texto
    Serial.println(msg.text);
    if (msg.text.equalsIgnoreCase("/on")){
      Serial.println("ON");
      digitalWrite(RELAY,LOW); //GPIO0, Activa con LOW, desactiva con HIGH
      myBot.sendMessage(msg, "Rele ON", "");
    }
  }
  if (msgType == MessageType){
    // Recibí un mensaje de texto
    Serial.println(msg.text);
    if (msg.text.equalsIgnoreCase("/off")){
      Serial.println("OFF");
      digitalWrite(RELAY,HIGH); //GPIO0, Activa con LOW, desactiva con HIGH
      myBot.sendMessage(msg, "Rele OFF", "");
    }
  }
}
}
}
}

```

Mas información:

<https://www.instructables.com/ESP0101S-RELAY-MODULE-TUTORIAL/>

<https://randomnerdtutorials.com/esp8266-relay-module-ac-web-server/>