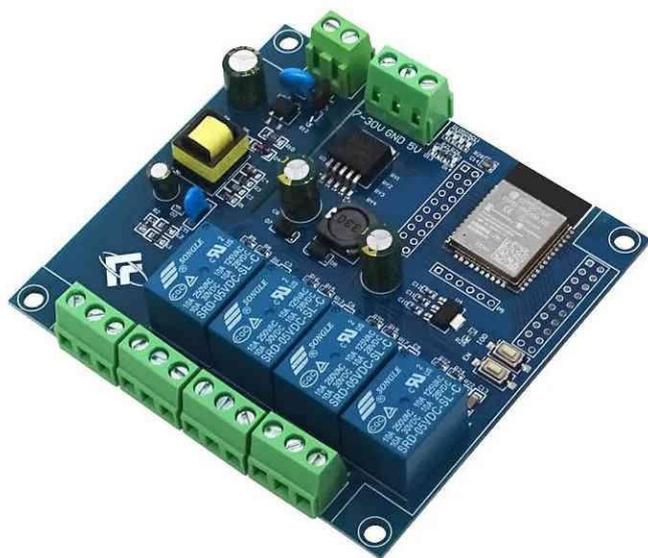


Datenblatt

Artikel RBS16467

4 Kanal Relais Modul ESP32 Wifi Bluetooth



Beschreibung.....	1
Details.....	2
Pinout.....	2
ESPHome Standardkonfiguration.....	3
Beispielsketch für Arduino IDE.....	4

Beschreibung

Entdecke die grenzenlose Welt der drahtlosen Steuerung mit unserem 4-Kanal Relais Modul ESP32 WiFi Bluetooth! Dieses Entwicklungsboard ist mit dem leistungsstarken ESP32-WROOM-32E WiFi&BLE Modul ausgestattet und bietet eine Vielzahl von Funktionen, die perfekt auf deine DIY-Projekte zugeschnitten sind.

Merkmale:

- Leistungsfähiger ESP32-WROOM-32E Module: Das Board verfügt über einen ausgereiften und stabilen ESP32-WROOM-32E Module mit großzügigen 4M Byte Flash-Speicher.
- Vielseitige I/O-Ports: Alle I/O-Ports des ESP32-Moduls sind herausgeführt, einschließlich des UART-Programm-Download-Ports, um die Sekundärentwicklung zu erleichtern.
- Praktische Bedienelemente: Ein programmierbarer Taster (IO0) und ein Reset-Taster auf dem Board erleichtern die Handhabung und Entwicklung.
- Arduino IDE Unterstützung: Die Verwendung von Arduino IDE Entwicklungstools wird vollständig unterstützt, um die Programmierung so einfach wie möglich zu gestalten.
- 4-Kanal 5V Relais: Mit onboard 4-Weg 5V Relais kannst du Schaltaufgaben steuern, ideal für Lasten mit Arbeitsspannungen bis AC 250V/DC30V.
- Integrierte LED-Anzeige: Eine programmierbare LED auf GPIO23 und vier LEDs als Relaisanzeige ermöglichen eine einfache Überwachung des Modulstatus.

Anwendungen:

Dieses Entwicklungsboard eignet sich perfekt für ESP32-Sekundärentwicklung, Lernprojekte und drahtlose Steuerung im Smart Home-Bereich. Entfessele deine kreativen Ideen und gestalte innovative Projekte!

Schnittstellen:

Programmierschnittstelle: GND, RX, TX, 5V sind für die Verbindung mit einem externen TTL-Serial-Port-Modul vorgesehen. IO0 wird während des Downloads mit GND verbunden und nach Abschluss des Downloads getrennt.

Relaisausgang:

NC (Normalerweise Geschlossen): Der Relay ist vor dem Einschalten mit COM kurzgeschlossen und bleibt nach dem Einschalten in der Luft.

COM (Gemeinsamer Anschluss): Der öffentliche Terminal.

NO (Normalerweise Offen): Der Relay ist vor dem Schließen aufgehängt und nach dem Schließen mit COM kurzgeschlossen.

Details

- AC/DC-betriebenes ESP32 4-Kanal Relais Entwicklungsboard
- Ausgestattet mit ESP32-WROOM-32E WiFi&BLE Modul
- Alle I/O-Ports herausgeführt
- Unterstützt AC220V/DC5-30V Stromversorgung
- Für ESP32 Sekundärentwicklung und Lernen
- Smart Home Drahtlose Steuerung und andere Anwendungen
- Boardgröße: 93*87mm
- Gewicht: 92g
- Vielseitige Anwendungsmöglichkeiten für innovative Projekte

Pinout

Das Board besitzt zwei zweireihige Header für jeden GPIO des ESP32 und einen einreihigen Pinheader zum Flashen des ESP32. Die Programmierung kann mit 3,3 V erfolgen, die mit dem 5V-Pin verbunden sind, und GPIO0 kann mit GND verbunden werden (über einen Jumper oder einen Taster).

Nach dem Flashen, wenn Sie das Board testen möchten, muss es ordnungsgemäß über dedizierte Anschlüsse oder über 5V Gleichstrom, der am 5V-Pin angelegt ist, mit Strom versorgt werden. Wenn Sie es mit 3,3 V am 5V-Pin einschalten, wird der ESP32 unter Spannungsausfallbedingungen stehen.

Header zum Programmieren

Pin	Comment
5V	Nicht zum Programmieren nutzen
TX	3.3.V Logikpegel!
RX	3.3.V Logikpegel!
GND	Masse
GND	Masse
GPIO0	3.3.V Logikpegel! (Mit einem Taster zum Programmieren verbunden)

Internes Pinout

Pin	Function
GPIO23	Status LED
GPIO32	Relais 1
GPIO33	Relais 2
GPIO25	Relais 3
GPIO26	Relais 4

ESPHome Standardkonfiguration

```
esphome:
  name: ac-dc-relay-x4
  friendly_name: ac-dc-relay-x4

esp32:
  board: esp32dev
  framework:
    type: arduino

output:
  - platform: gpio
    pin: GPIO23
    id: led
  - platform: gpio
    pin: GPIO32
    id: relay_pin_1
  - platform: gpio
    pin: GPIO33
    id: relay_pin_2
  - platform: gpio
    pin: GPIO25
    id: relay_pin_3
  - platform: gpio
    pin: GPIO26
    id: relay_pin_4

# Define the switches based on the relay pins
switch:
  - platform: output
    id: relay_1
    name: "Relay 1"
    output: relay_pin_1
  - platform: output
    id: relay_2
    name: "Relay 2"
    output: relay_pin_2
  - platform: output
    id: relay_3
    name: "Relay 3"
    output: relay_pin_3
  - platform: output
    id: relay_4
    name: "Relay 4"
    output: relay_pin_4

interval:
  - interval: 1000ms
    then:
      - output.turn_on: led
      - delay: 500ms
      - output.turn_off: led
```

Beispielsketch für Arduino IDE

```
#define RELAIS_1 32 // Relais Pins werden deklariert
#define RELAIS_2 33
#define RELAIS_3 25
#define RELAIS_4 26

void setup() {
  pinMode(RELAIS_1 ,OUTPUT); // Relais-Pins werden als Output deklariert
  pinMode(RELAIS_2 ,OUTPUT);
  pinMode(RELAIS_3 ,OUTPUT);
  pinMode(RELAIS_4 ,OUTPUT);
}

void loop() {
  digitalWrite(RELAIS_1 , HIGH); // Relais 1 für 5 Sekunden an
  digitalWrite(RELAIS_2 , HIGH); // Relais 2 für 5 Sekunden an
  delay(5000);
  digitalWrite(RELAIS_1 ,LOW); // Relais 1 für 5 Sekunden aus
  digitalWrite(RELAIS_2 ,LOW); // Relais 2 für 5 Sekunden aus
  delay(5000);
  digitalWrite(RELAIS_3 , HIGH); //Relais 3 für 5 Sekunden an
  digitalWrite(RELAIS_4 , HIGH); // Relais 4 für 5 Sekunden an
  delay(5000);
  digitalWrite(RELAIS_3 ,LOW); // Relais 3 für 5 Sekunden aus
  digitalWrite(RELAIS_4 ,LOW); // Relais 4 für 5 Sekunden aus
  delay(5000);
}
```