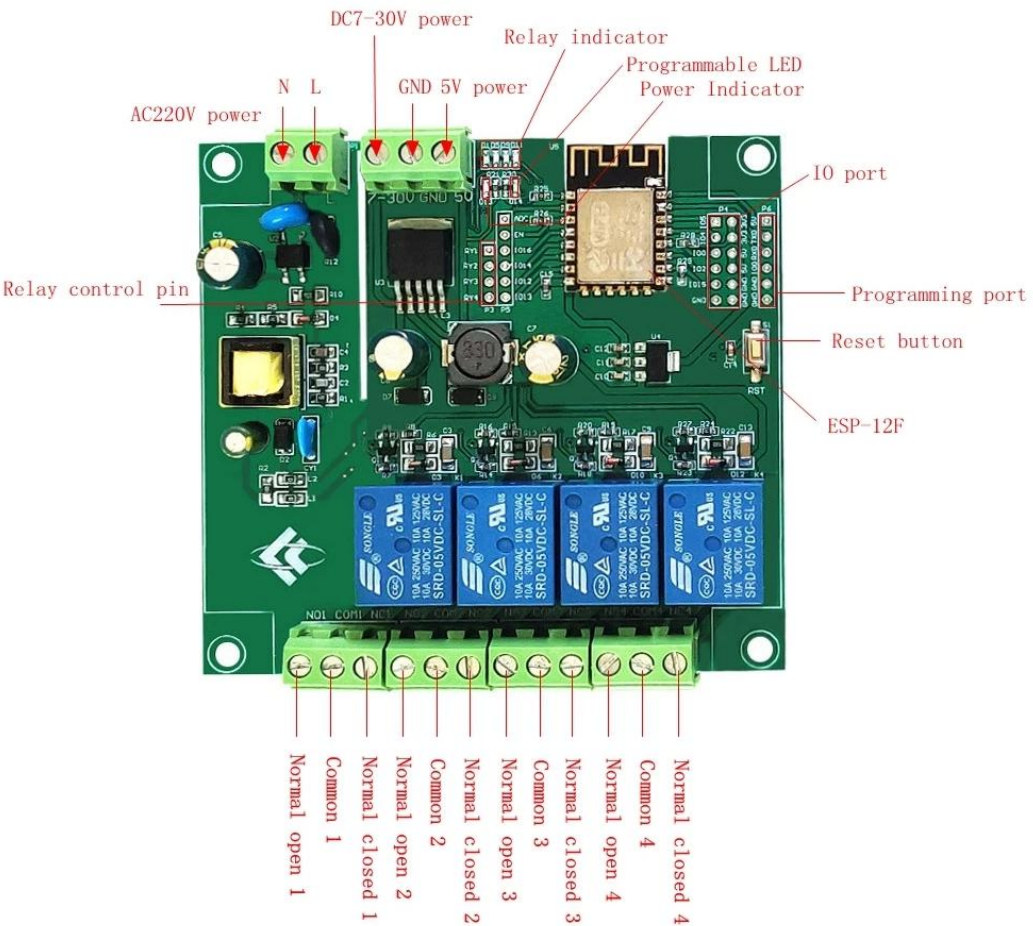
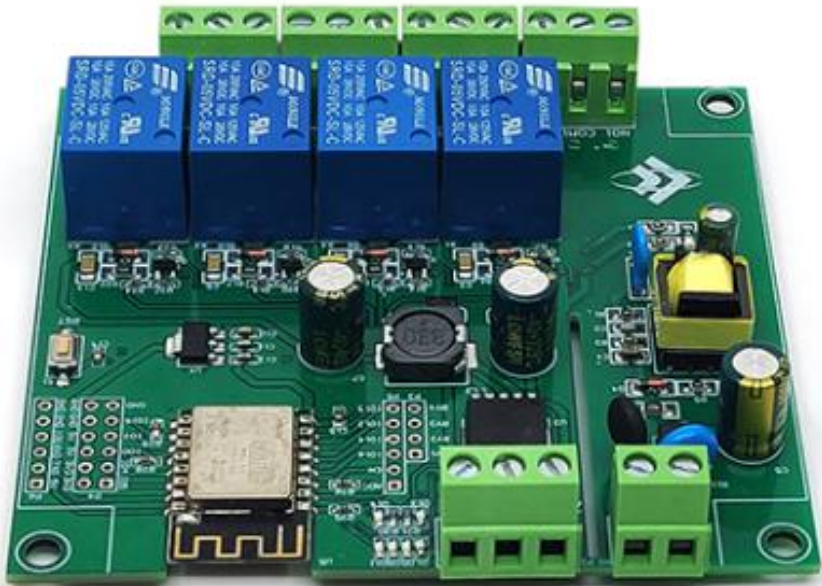


LC-Relay-ESP12-4R-MV

The ESP8266 eight-way relay development board is equipped with ESP-12F WiFi module and supports AC/DC7-28V/5V power supply. Provide for Arduino development environment reference code, suitable for ESP8266 secondary development learning, smart home wireless control and other occasions.



Feature introduction:

1. Onboard mature and stable ESP-12F WIFI module, large capacity 4M BYTE FLASH;
2. The I/O port of the WIFI module and the UART program download port are all led out to facilitate secondary development;
3. Onboard AC-DC switching power supply module, power supply mode supports AC90-250V/DC7-30V/5v;
4. Onboard WIFI module RST reset button;
5. ESP-12F supports the use of development tools such as ECLIPSE/ARDUINO IDE, and provides ARDUINO Reference program in the development environment;
6. Onboard 4ch 5V relays, output switch signal, suitable for controlling the working voltage of AC Load within 250V/DC 30V;
7. On-board power indicator, 1 programmable LED and relay indicator.

Burning port:

The GND, RX, TX, and 5V of ESP8266 are respectively connected to the GND, TX, RX, of the external TTL serial port module

5V, IO0 needs to be connected to GND when downloading, and then disconnect the connection between IO0 and GND after downloading;

Relay output:

NC: Normally closed, the relay is short-connected with COM before it is closed, and it is suspended after being closed;

COM: public end;

NO: Normally open end, the relay is suspended before being closed, and shorted to COM after being closed

Introduction to GPIO port

Num.	Name	Function	Num.	Name	Function
1	ADC	A/D conversion result. Input voltage range 0~1V, value range: 0~1024	11	IO15	GPIO15; MTDO; HSPI_CS; UART0_RTS
2	EN	Enable pin, pull up by default	12	TXD	UART0_TXD; GPIO1
3	IO16	GPIO16	13	RXD	UART0_RXD; GPIO3
4	IO14	GPIO14; HSPI_CLK	14	GND	Power ground
5	IO12	GPIO12; HSPI_MISO	15	5V	5V power supply
6	IO13	GPIO13; HSPI_MOSI; UART0_CTS	16	3.3V	3.3V power supply
7	IO5	GPIO5	17	RY1	One relay drive port, you can use a shorting cap and IO16 to short; if you want to use other I/O drive relays, you can use DuPont wire jumpers
8	IO4	GPIO4	18	RY2	One relay drive port, you can use a shorting cap and IO14 to short; if you want to use other I/O drive relays, you can use DuPont wire jumpers

9	IO0	GPIO0	19	RY3	One relay drive port, you can use a shorting cap and IO12 to short; if you want to use other I/O drive relays, you can use DuPont wire jumpers
10	IO2	GPIO2; UART1_TXD	20	RY4	One relay drive port, you can use a shorting cap and IO13 to short; if you want to use other I/O drive relays, you can use DuPont wire jumpers

Arduino development environment construction

ESP8266 supports development tools such as Eclipse/Arduino IDE. It is relatively simple to use Arduino. Here is how to build Arduino development environment:

1. Install Arduino IDE 1.8.9 or the latest version;
2. Open the Arduino IDE, click File-Preferences in the menu bar, and click Add URL in "Additional Development Board Manager URL" after entering the preferences:
http://arduino.esp8266.com/stable/package_esp8266com_index.json
3. Click Tools-Development Board-Development Board Manager in the menu bar, then search for "ESP8266" to install the Arduino support package 2.5.2 or the latest version of ESP8266

Program download

1. Use a jumper cap to connect the IO0 and GND pins, prepare a TTL serial port module (for example: FT232) to plug into the computer USB, the serial port module and the development board are connected as follows:

TTL serial port module	ESP8266 development board
GND	GND
TX	RX
RX	TX
5V	5V

2. Click Tools-Development Board in the menu bar, select the development board as ESPino (ESP-12 module)
3. Open the program you want to download, click on Tools-Port in the menu bar to select the correct port number
4. After clicking "Upload", the program will be automatically compiled and downloaded to the development board, as follows:



The screenshot shows the Arduino IDE interface with the file 'LED_Relay.ino' open. The code defines two pins, PIN_LED (16) and PIN_RELAY (5). The setup function configures both pins as outputs and sets the LED to HIGH and the relay to HIGH. The loop function toggles the LED state (LOW for 1s, HIGH for 1s) and keeps the relay HIGH. The serial monitor shows the upload was successful and the board is now in COM48.

```
LED_Relay | Arduino 1.8.9
文件 编辑 项目 工具 帮助

LED_Relay

/*
  文件名称: LED_Relay.ino
  功能: LED闪烁+继电器开启
*/

#define PIN_LED 16
#define PIN_RELAY 5

void setup()
{
  pinMode(PIN_LED, OUTPUT); //输出模式
  pinMode(PIN_RELAY, OUTPUT); //输出模式
  digitalWrite(PIN_LED, HIGH); //LED默认关闭
  digitalWrite(PIN_RELAY, HIGH); //继电器默认开启
}

void loop()
{
  digitalWrite(PIN_LED, LOW); //打开LED
  delay(1000); //延时1S
  digitalWrite(PIN_LED, HIGH); //关闭LED
  delay(1000); //延时1S
}

上传成功
Leaving...
Hard resetting via RTS pin...

3 ESPino (ESP-12 Module) 在 COM48
```

5. Finally, disconnect IO0 and GND, power on the development board again or press the reset button to run the program.