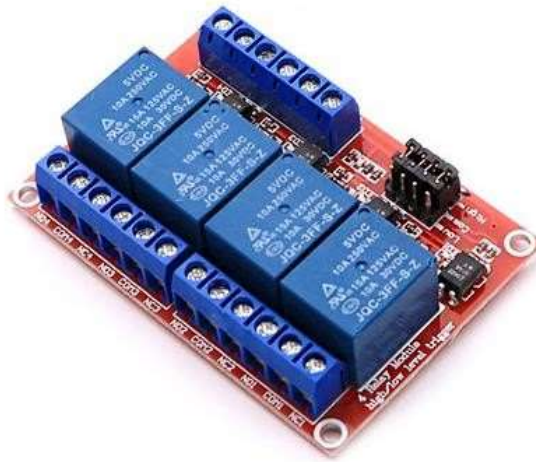


Arduino 4 Channel Relay Module 5V

User's Manual



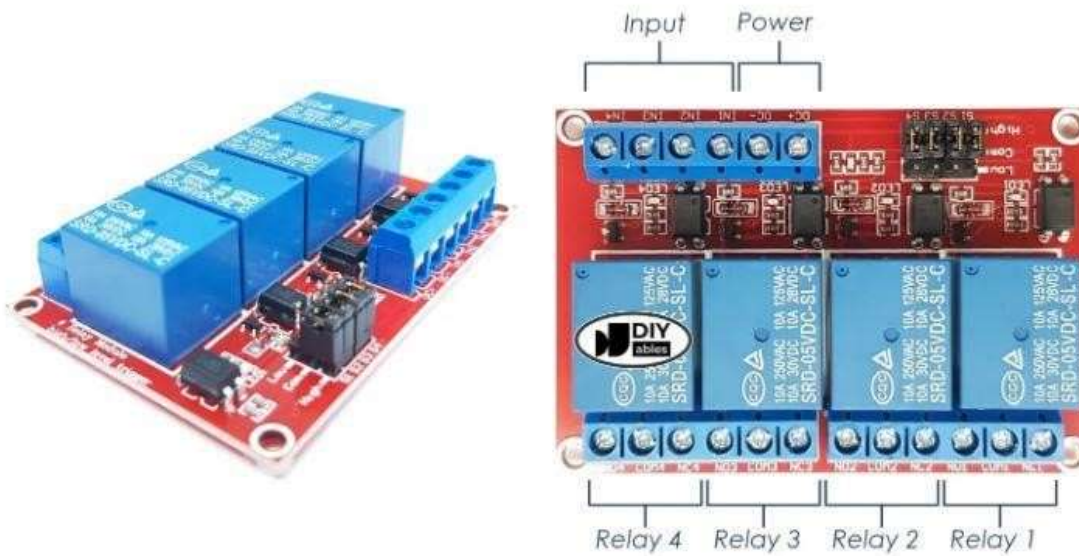
Arduino 4 Channel Relay Module:

When we want to control 4 high-voltage devices such as pumps, fans, actuators... We can use multiple relay modules. However, there is a simpler way is to use a 4-channel relay module. A 4-channel relay module is a combination of 4 relays on a single board.

A 4-channel relay module vs 4 x 1-channel relay modules:

- ◆— A 4-channel relay module has simpler wiring.
- ◆— A 4-channel relay module uses less space.
- ◆— A 4-channel relay module is cheaper.
- ◆— The programming is the same.

About Arduino 4 Channel Relay Module:



A 4-channel relay module has the following pins:

◆ Power pins for relay boards

- DC+: connect this pin to 5V pin of power supply
- DC-: connect this pin to the GND pin of the power supply and also to the GND pin of the Arduino

◆ Signal pins:

- IN1: this pin receives the control signal from Arduino to control relay 1 on the module
- IN2: this pin receives the control signal from Arduino to control relay 2 on the module
- IN3: this pin receives the control signal from Arduino to control relay 3 on the module
- IN4: this pin receives the control signal from Arduino to control relay 4 on the module

◆ Output pins: NCx (**normally closed** pin), NOx (**normally open** pin), COMx (**common** pin),

- NC1, NO1, COM1: These pins connect to a high-voltage device that is controlled by relay 1
- NC2, NO2, COM2: These pins connect to a high-voltage device that is controlled by relay 2
- NC3, NO3, COM3: These pins connect to a high-voltage device that is controlled by relay 3
- NC4, NO4, COM4: These pins connect to a high-voltage device that is controlled by relay 4

For detail of how to connect relay to high-voltage, what are differences between **normally closed** and **normally open**

It also has 4 jumpers to select between the low trigger and the high trigger for each relay individually.

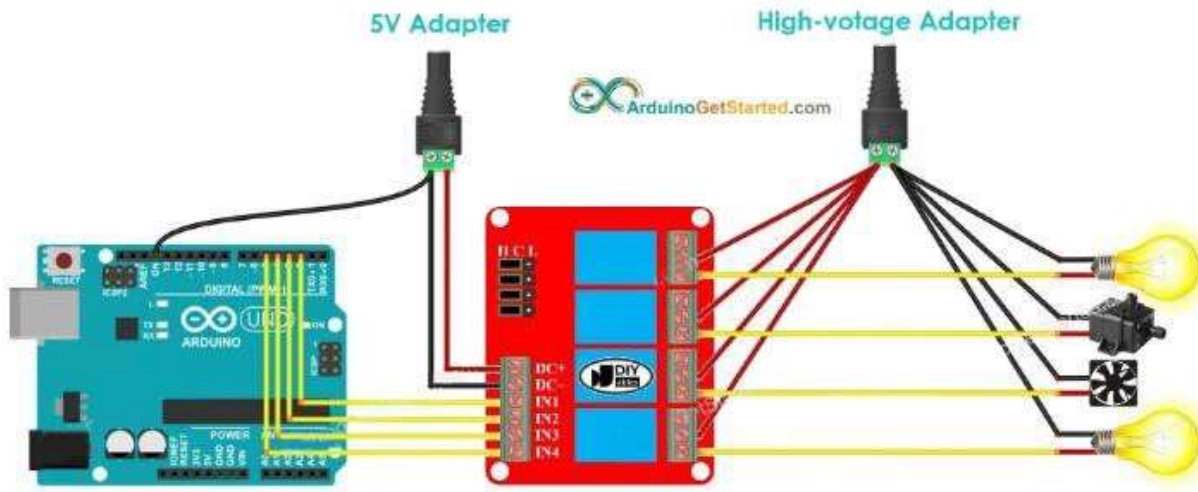
Wiring Diagram:

4-channel relay module consumes considerable power. Therefore, we should NOT power the module directly from the 5V pin of Arduino. We need to use external 5V power for the module instead.

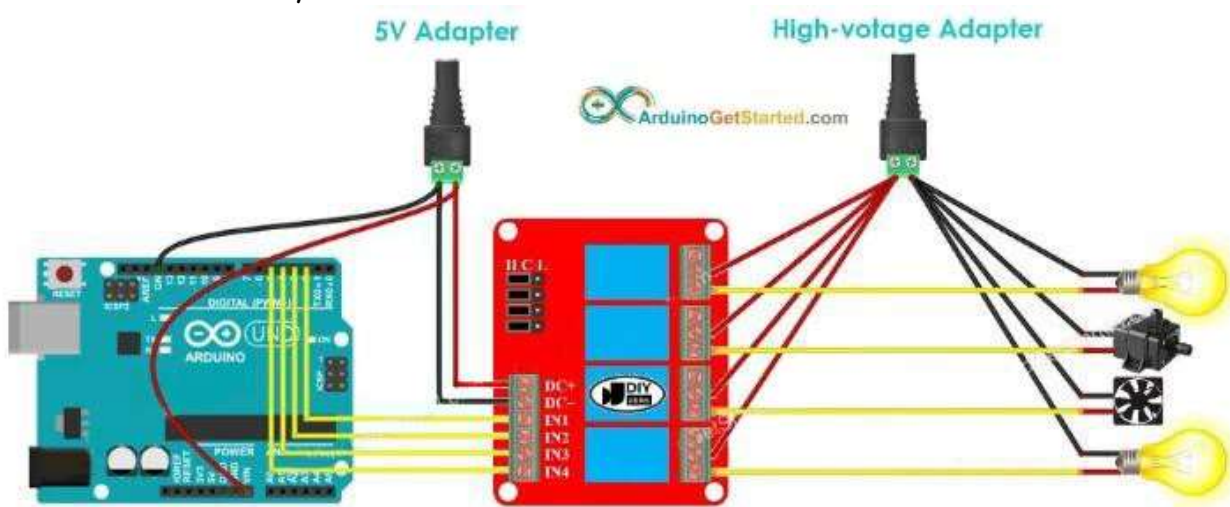
So, we need to use three power sources:

- ◆ — A 5V power adapter for Arduino
- ◆ — A 5V power adapter for the 4-channel relay module
- ◆ — A higher-voltage power adapter (12VDC, 24VDC, 48VDC, 220AC...) for things that are controlled by the 4-channel relay module

- ◆— Wiring diagram with three power sources. Power supply for Arduino (not included in the image) can be via either USB cable or power jack.



- ◆— We can reduce the number of power adapters by using a single 5V power source for both Arduino and the 4-channel relay module.



※ **NOTE THAT:**

If 4 devices that are controlled by a 4-channel relay module use the same voltage, we can use a single high-voltage power adapter for all. However, if they use different voltages, we can use different high-voltage power adapters independently.

How to Program for 4-Channel Relay Module:

- ◆— Initialize the Arduino pin to the digital output mode by using pinMode() function.

```
pinMode(PIN_RELAY_1, OUTPUT);
pinMode(PIN_RELAY_2, OUTPUT);
pinMode(PIN_RELAY_3, OUTPUT);
pinMode(PIN_RELAY_4, OUTPUT);
```

- ◆— Control the relay's state by using digitalWrite() function.

```
digitalWrite(PIN_RELAY_1, HIGH);
digitalWrite(PIN_RELAY_2, HIGH);
digitalWrite(PIN_RELAY_3, HIGH);
digitalWrite(PIN_RELAY_4, HIGH);
```

Arduino Code:

```
/*
 * Created by ArduinoGetStarted.com
 *
 * This example code is in the public domain
 *
 * Tutorial page: https://arduinogetstarted.com/tutorials/arduino-4-channel-
relaymodule */

#define PIN_RELAY_1 2 // the Arduino pin, which connects to the IN1 pin of relay module
#define PIN_RELAY_2 3 // the Arduino pin, which connects to the IN2 pin of relay module
#define PIN_RELAY_3 4 // the Arduino pin, which connects to the IN3 pin of relay module
#define PIN_RELAY_4 5 // the Arduino pin, which connects to the IN4 pin of relay module

// the setup function runs once when you press reset or power the board void
setup() {
  Serial.begin(9600);

  // initialize digital pin as an output.
  pinMode(PIN_RELAY_1, OUTPUT);
  pinMode(PIN_RELAY_2, OUTPUT);
  pinMode(PIN_RELAY_3, OUTPUT);
  pinMode(PIN_RELAY_4, OUTPUT); }

// the loop function runs over and over again forever void
loop() {
  Serial.println("Turn on all");
  digitalWrite(PIN_RELAY_1, HIGH);
  digitalWrite(PIN_RELAY_2, HIGH);
  digitalWrite(PIN_RELAY_3, HIGH);
  digitalWrite(PIN_RELAY_4, HIGH); delay(1000);

  Serial.println("Turn off all");
  digitalWrite(PIN_RELAY_1, LOW);
  digitalWrite(PIN_RELAY_2, LOW);
  digitalWrite(PIN_RELAY_3, LOW);
  digitalWrite(PIN_RELAY_4, LOW);
  delay(1000); }
```

Quick Steps:

- ◆— Copy the above code and open with Arduino IDE
- ◆— Click Upload button on Arduino IDE to upload code to Arduino
- ◆— Listen the click sound on relays.
- ◆— See the result on Serial Monitor.



The screenshot shows the Serial Monitor window for COM6. The output text is as follows:

```
Turn on all
Turn off all
Turn on all
Turn off all
Turn on all
Turn off all
Turn on all
Turn off all
```

At the bottom of the window, there are several controls: Autoscroll, Show timestamp, a Newline dropdown menu, a baud rate dropdown menu set to 9600, and a Clear output button.