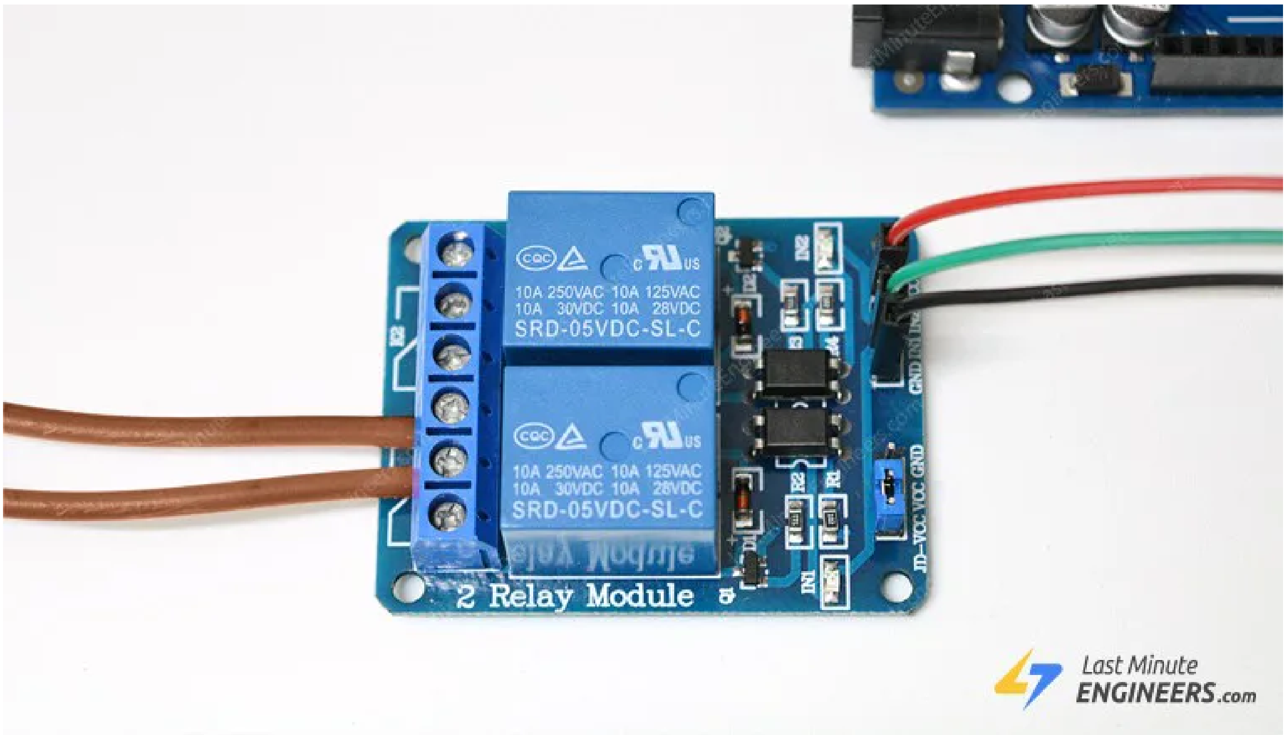
Interface Two Channel Relay Module with Arduino



In your upcoming project, you might want to use your Arduino to control a high-voltage device, like a lamp, fan, or other household appliance. However, because the Arduino runs on 5 volts, it cannot directly control these high-voltage appliances.

This is where relay modules come into play. These well-contained modules are inexpensive, simple to connect, and ideal for home-brew projects that require switching modest amounts of AC or DC power. The only downside is that, because these are electro-mechanical devices, they are more prone to wear and tear over time.

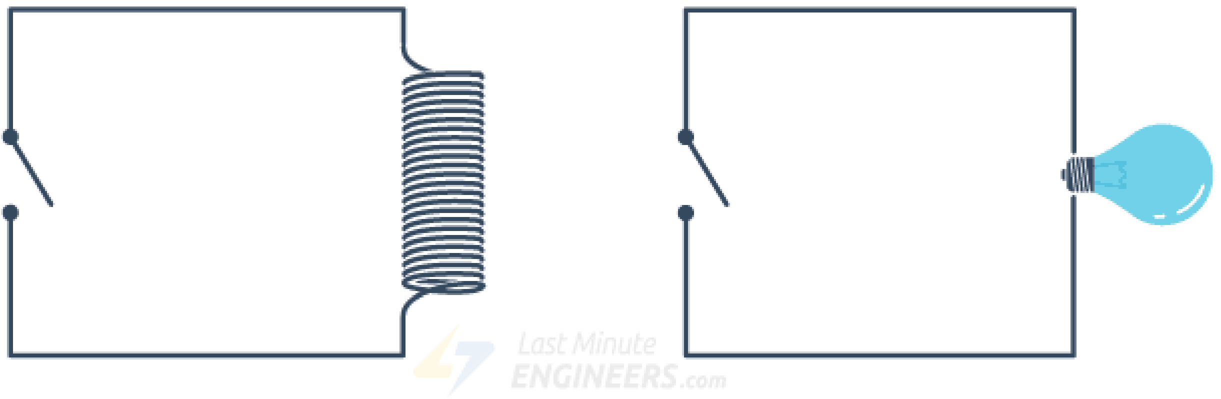
This tutorial will walk you through setting up the relay module to turn on a lamp or other device, but first, a quick primer on relays.

How Do Relays Work?

At the core of a relay is an electromagnet (a wire coil that becomes a temporary magnet when electricity is passed through it). A relay can be thought of as an electric lever; you turn it on with a relatively small current, and it turns on another device with a much larger current.

Relay Basics

Here’s a small animation showing how a relay links two circuits together.



To illustrate, think about two simple circuits: one with an electromagnet and a switch or sensor, and the other with a magnetic switch and a light bulb.

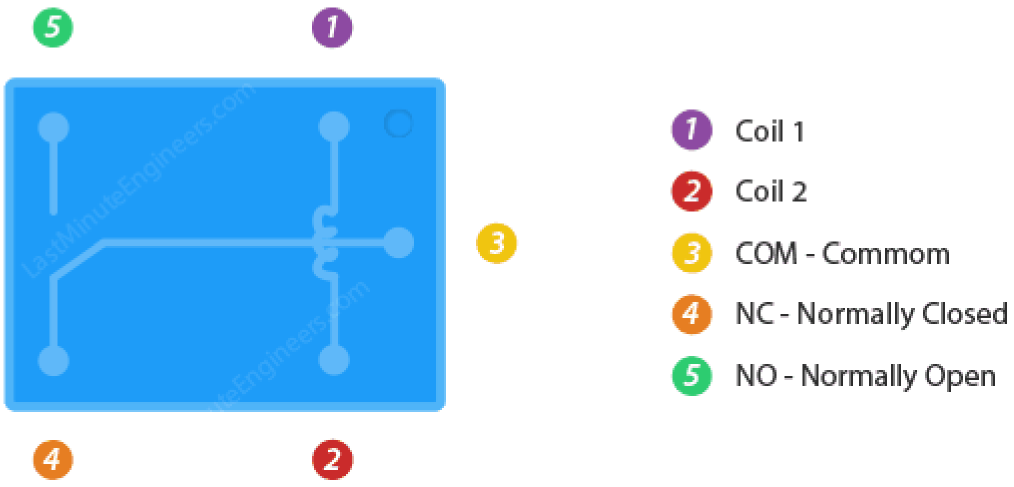
Initially, both circuits are open, with no current flowing through them.

When a small current flows through the first circuit, the electromagnet is energized, creating a magnetic field around it. The energized electromagnet attracts the second circuit's contact, closing the switch and allowing a large current to flow.

When the current in the first circuit stops flowing, the contact returns to its original position, reopening the second circuit.

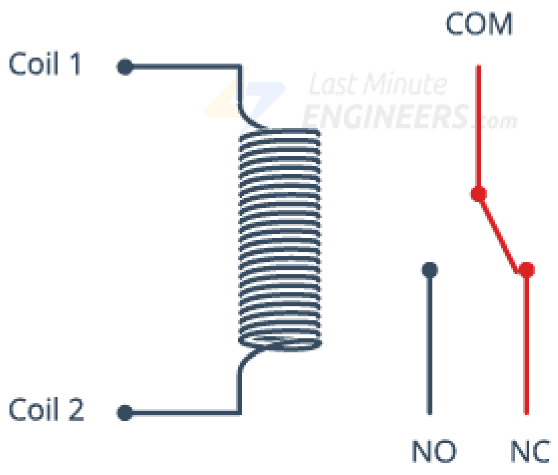
Relay Operation

A relay typically has five pins, three of which are high voltage terminals (NC, COM, and NO) that connect to the device being controlled.



The device is connected between the COM (common) terminal and either the NC (normally closed) or NO (normally open) terminal, depending on whether the device should remain normally on or off.

Between the remaining two pins (coil1 and coil2) is a coil that acts as an electromagnet.



Normally (initial position), the COM terminal is connected to the NC terminal and the NO terminal is open.

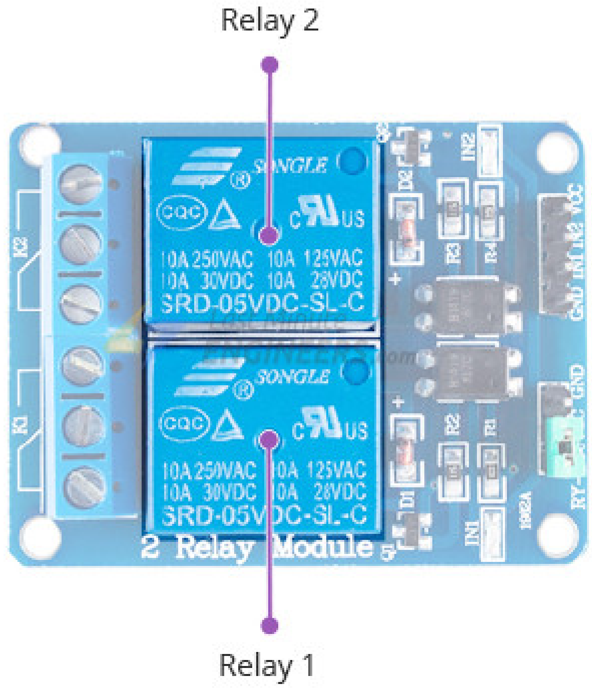
When current flows through the coil, the electromagnet becomes energized, causing the switch's internal contact to move. The COM then connects to the NO terminal, disconnecting from the NC terminal.

When the current stops flowing through the coil, the internal contact is returned to its initial position, re-connecting the NC terminal to the COM and re-opening the NO terminal.

To put it another way, the relay functions as a single-pole-double-throw switch (SPDT).

Two-Channel Relay Module Hardware Overview

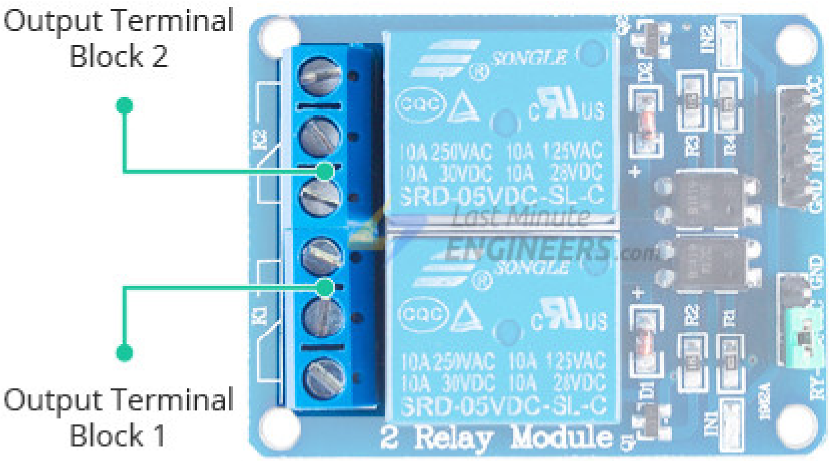
The two-channel relay module is designed to allow your Arduino to control two high-powered devices. It has two relays, each with a maximum current rating of 10A at 250VAC or 30VDC.



Modules with one, four, and eight channels are also available. You can choose the one that best meets your needs.

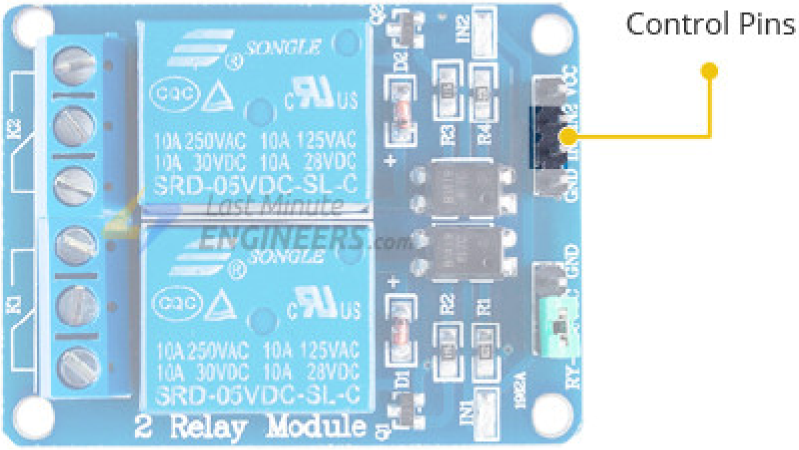
Output Terminal Blocks

The high voltage terminals (NC, COM, and NO) of each relay are broken out to two screw terminals. The device you want to control can be connected across them.



Module Control

On the other side of the module, there are two input pins, IN1 and IN2, for controlling the relay. These pins are 5V logic compatible, so if you have a microcontroller like an Arduino, you can drive a relay with any digital output pin.

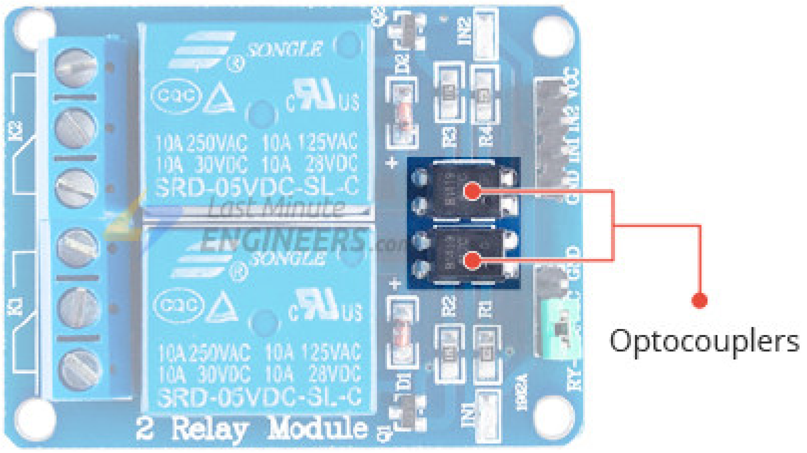


The input pins are active low, which means that a logic LOW activates the relay and a logic HIGH deactivates it.

The relay module has two LEDs that indicate the status of the relay. When a relay is activated, the corresponding LED lights up.

Built-in Optocouplers

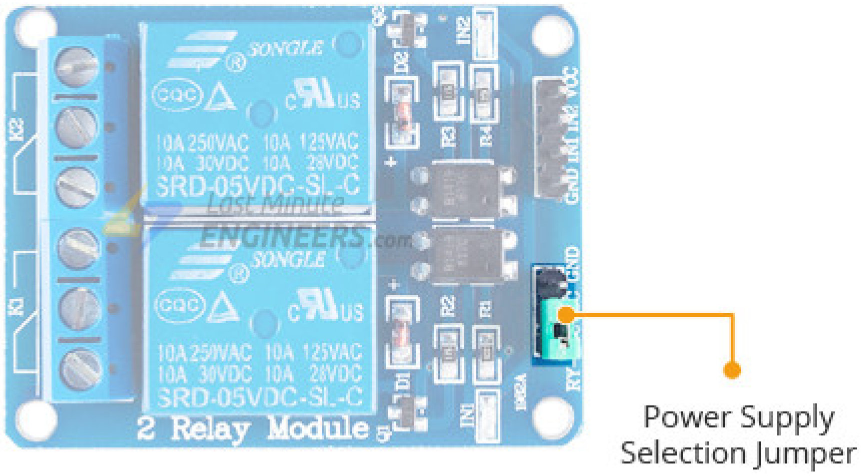
One of the best features of these modules is the inclusion of two optocouplers on the logic inputs.



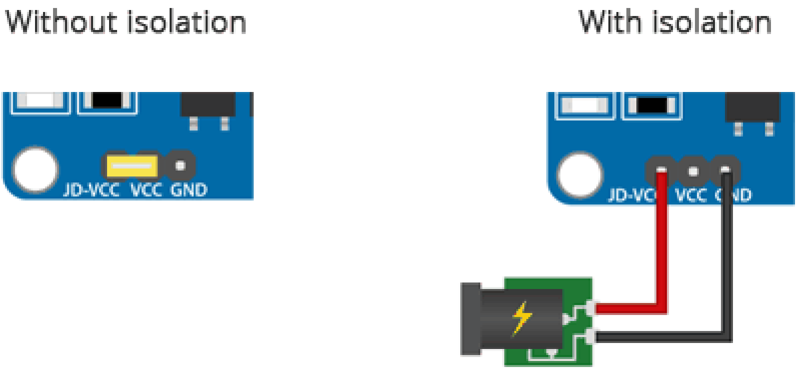
Optocouplers offer complete electrical isolation between the logic control input and the relay power as an extra layer of protection in the event of a major failure on the relay's AC load, such as a lightning strike.

Power Supply Selection Jumper

The module has a jumper between the header pins JD-VCC and VCC (the relay power is connected to the logic power). This jumper determines whether or not the relay module is physically connected to the Arduino.



When the jumper is in place, the Arduino powers the relay's electromagnet directly. In this case, the relay module and the Arduino are not physically isolated from each other, but this makes the device easier to use because it only needs one power supply.

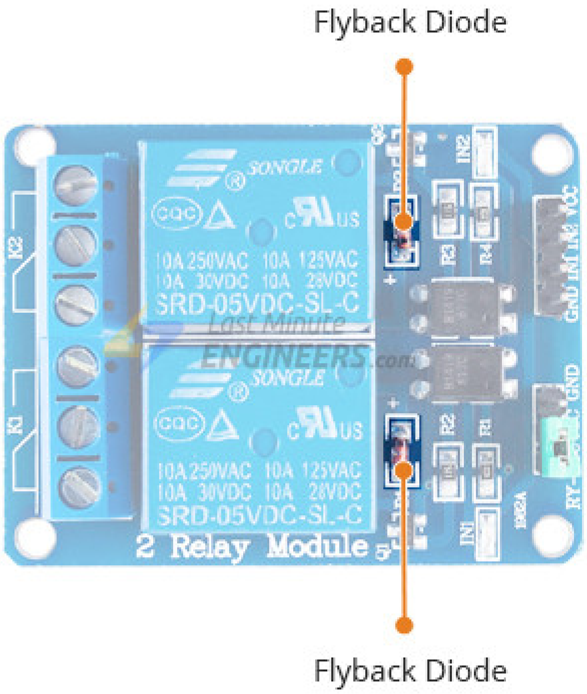


When you remove the jumper, the relay module is physically isolated from the Arduino. However, in this case, you must provide a separate 5V power supply voltage to the JD-VCC and GND.

Module Power

The module operates on 5 volts and draws approximately 140mA when both relays are activated (70mA each).

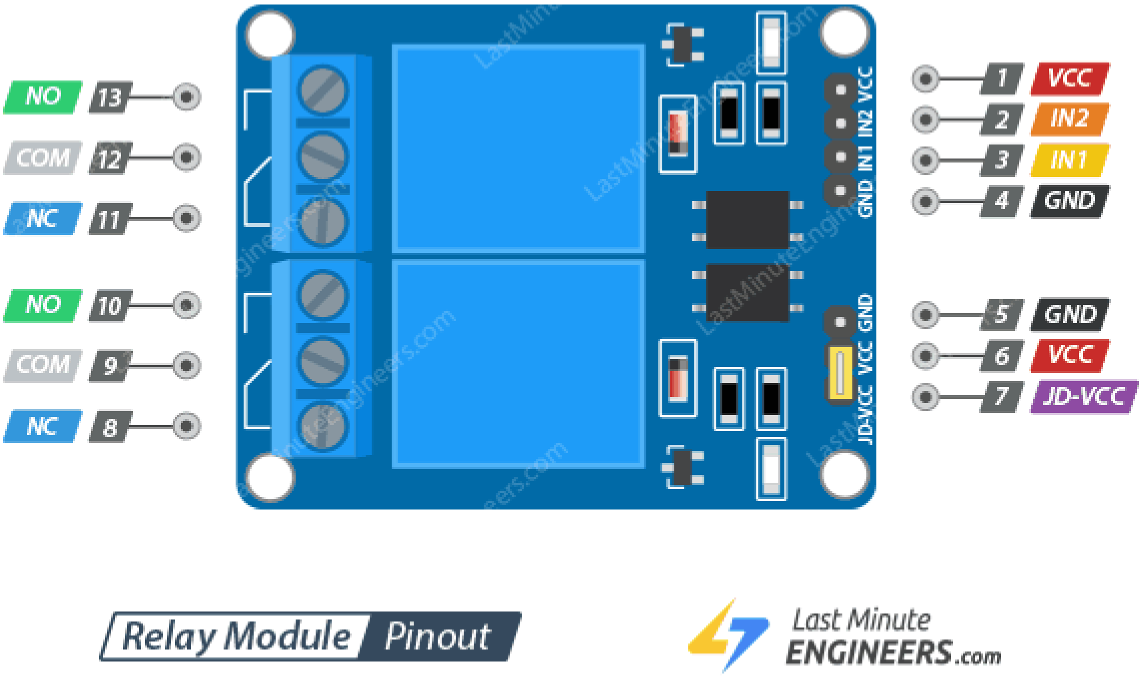
The module also includes flyback diodes that are connected in parallel with the relay coils to safely shunt current when the relay coil is de-energized.



Remember that if optocouplers are enabled, two separate 5V power sources are required.

Two-Channel Relay Module Pinout

Let's take a look at the pinout.



Control Pins:

VCC pin provides power to the built-in optocouplers and, optionally, the relay's electromagnet (if you keep the jumper in place). Connect it to the 5V pin on the Arduino.

GND is the common ground pin.

IN1 & IN2 pins control the relay. These are active low pins, which means that pulling them LOW activates the relay and pulling them HIGH deactivates it.

Power Supply Selection Pins:

JD-VCC provides power to the relay's electromagnet. When the jumper is in place, JD-VCC is shorted to VCC, allowing the electromagnets to be powered by the Arduino's 5V line. Without the jumper cap, you'd have to connect it to a separate 5V power source.

VCC pin is shorted to the JD-VCC pin with the jumper cap on. Keep this pin disconnected if you remove the jumper.

GND is the common ground pin.

Output Terminals:

COM terminal connects to the device you intend to control.

NC terminal is normally connected to the COM terminal, unless you activate the relay, which breaks the connection.

NO terminal is normally open, unless you activate the relay that connects it to the COM terminal.

Wiring a Two-Channel Relay Module to an Arduino

Now that we know everything about the relay module, it's time to put it to use! Let's wire up our relay module to operate a lamp.

> **Warning:**
> This board interacts with HIGH AC voltage. Improper or incorrect use could result in serious injury or death. Therefore, it is intended for people who are familiar with and knowledgeable about HIGH AC voltage.

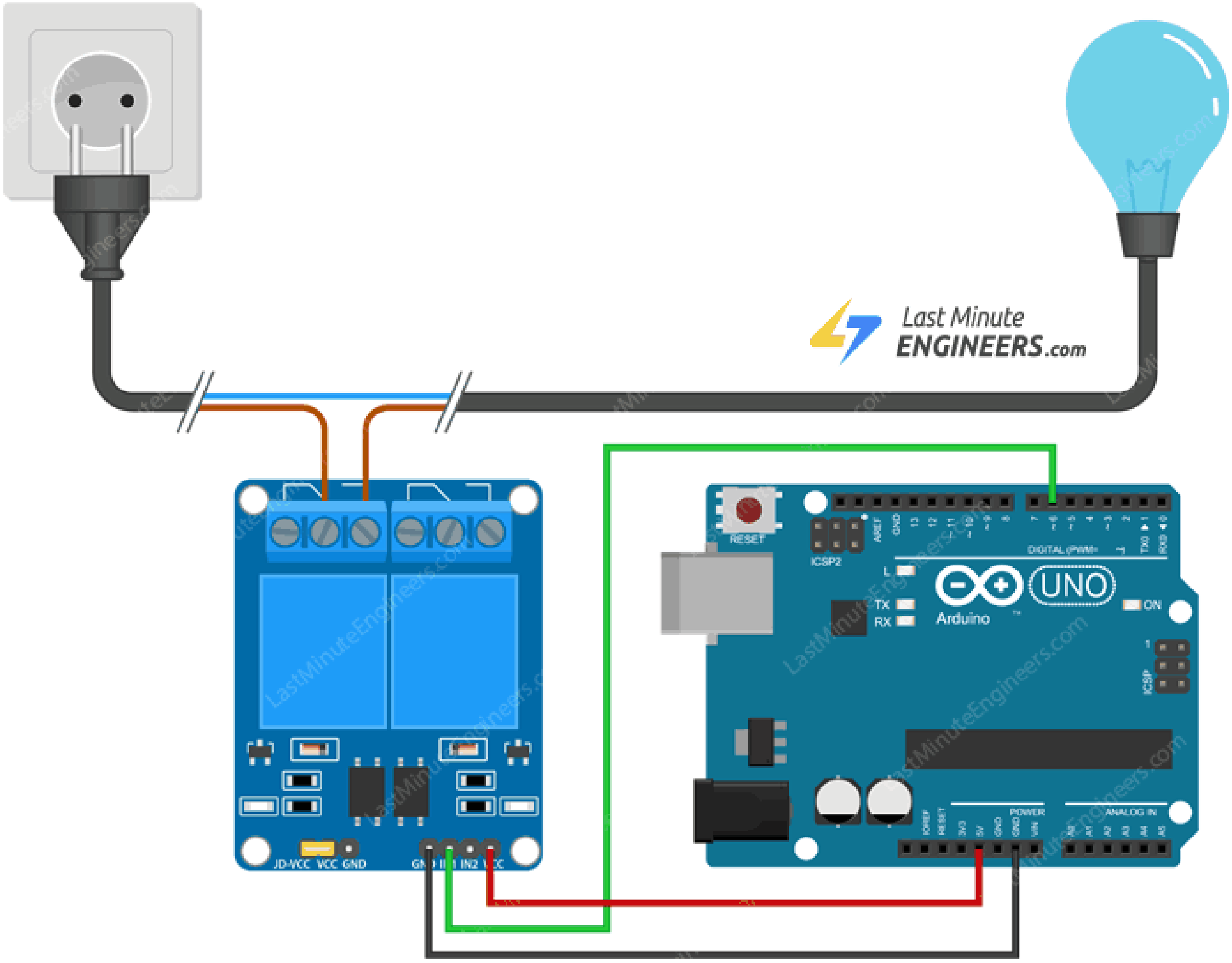
Begin by connecting the module's VCC pin to the Arduino's 5V pin and the GND pin to ground. We will only be using one relay for our experiment, so connect digital pin #6 to the IN1 input pin.

You'll also need to connect the relay module to the AC-powered device you want to control, in this case, a lamp. You'll need to cut your live AC line and connect one end of the cut wire (coming from the wall) to COM and the other to NC or NO, depending on what you want your device's initial state to be.

If you want to keep your device off most of the time and turn it on occasionally, connect the other end of the wire to NO. Otherwise, connect it to NC.

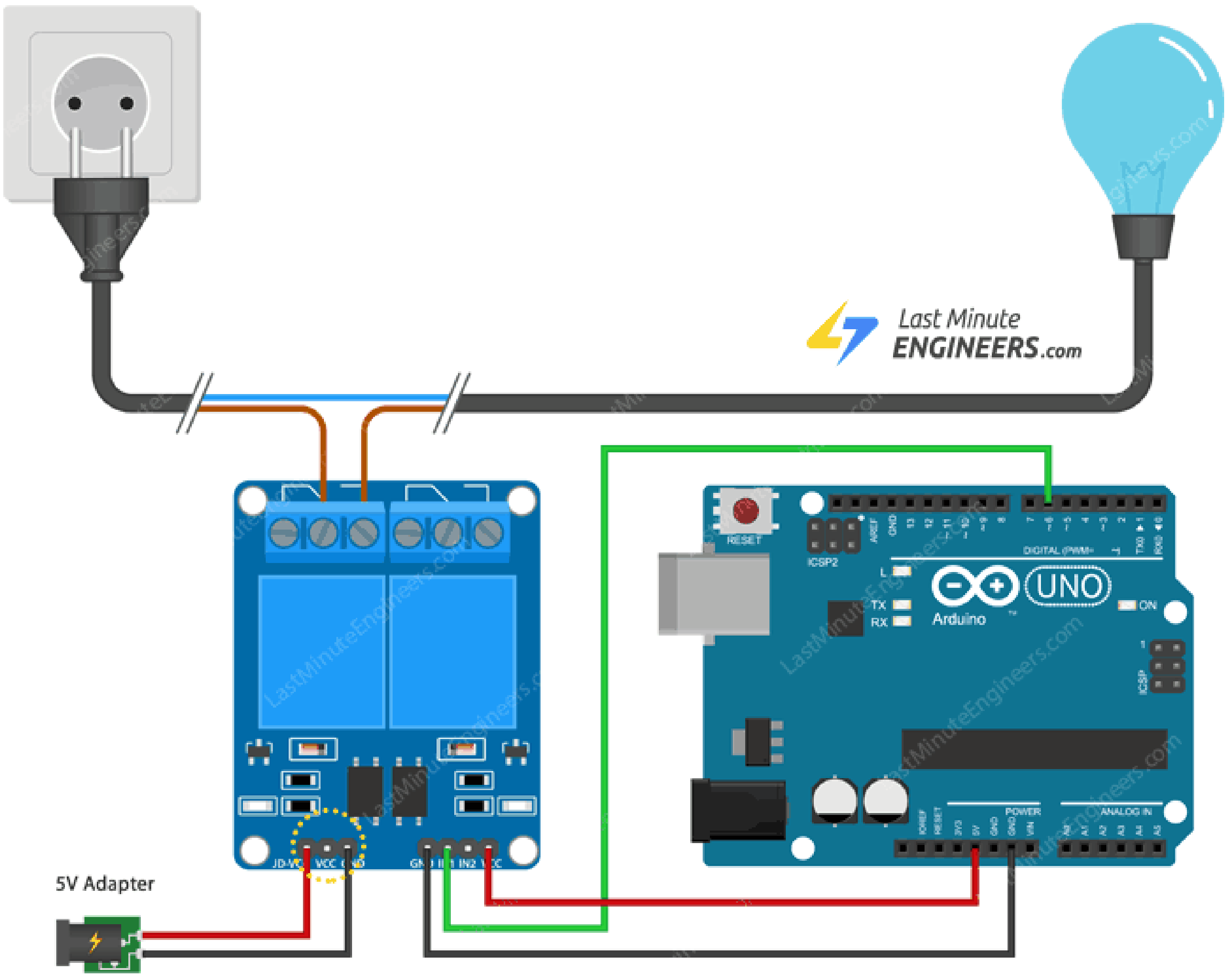
For this project, we want our lamp to be off at first and then turn on when we activate the relay, so we will connect one end of the wire to COM and the other to NO.

The following illustration shows the wiring.



We left the jumper in place in the above wiring diagram, so the relay's electromagnet will be driven directly from the Arduino. In this case, the relay module and the Arduino will not be physically isolated.

If you want to keep them isolated, you must provide a separate 5V power supply voltage to the JD-VCC and GND. The wiring below shows how to accomplish this.



Remove jumper when external power suppply is connected

Arduino Example Code

Controlling a relay module with the Arduino is as easy as controlling an LED. Here's a simple code that will activate the relay for 3 seconds and then deactivate it for 3 seconds.

```
int RelayPin = 6;

void setup() {
	// Set RelayPin as an output pin
	pinMode(RelayPin, OUTPUT);
}

void loop() {
	// Let's turn on the relay...
	digitalWrite(RelayPin, LOW);
	delay(3000);
	
	// Let's turn off the relay...
	digitalWrite(RelayPin, HIGH);
	delay(3000);
}
```

Code Explanation:

The sketch begins by declaring the pin to which the relay module's input pin is connected.

```
int RelayPin = 6;
```

In the setup function, we configure the input pin to behave as an output.

```
pinMode(RelayPin, OUTPUT);
```

In the loop function, we turn the device ON/OFF by pulling the relay pin LOW/HIGH.

`digitalWrite(RelayPin, LOW)` pulls the pin LOW, whereas `digitalWrite(RelayPin, HIGH)` pulls the pin HIGH.

```
digitalWrite(RelayPin, LOW);
delay(3000);

digitalWrite(RelayPin, HIGH);
delay(3000);
```