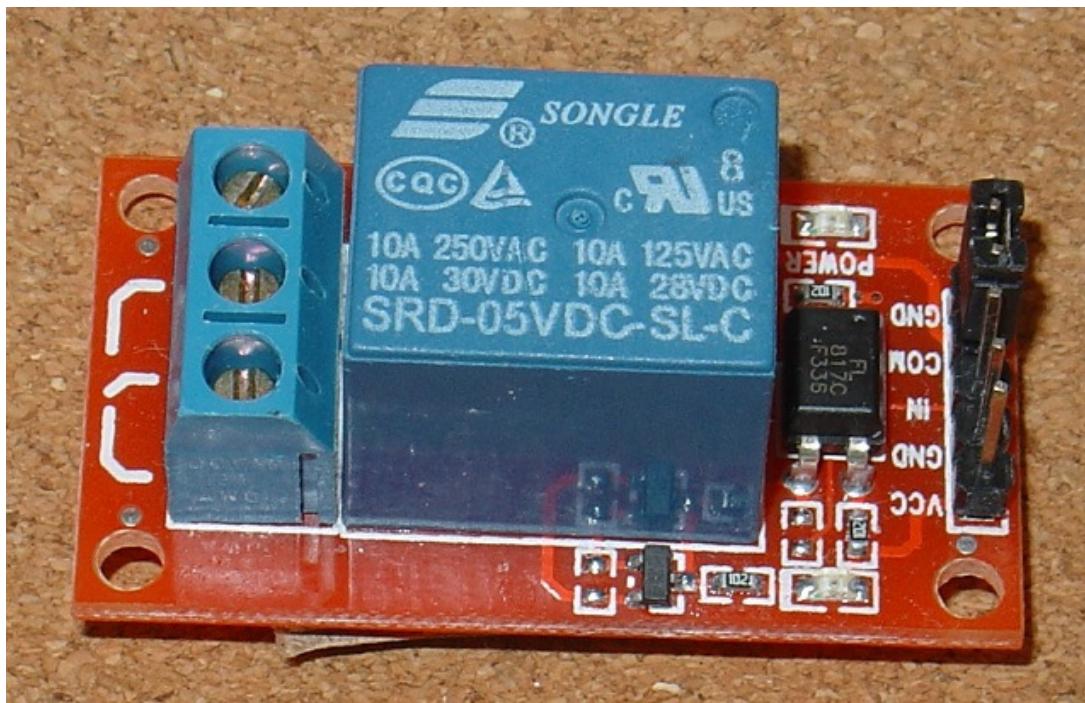




Home Automation How to Add Relays to Arduino

by Matthew Ford 26th January 2015 (originally posted 10th September 2014)
 © Forward Computing and Control Pty. Ltd. NSW Australia
 All rights reserved.



How to select Relays – Its all about Power.

When you are selecting a relay to connect to your Arduino you need to consider
 a) the power (voltage and current) of what you want to switch on and off and
 b) the power (voltage and current) required to drive the relay (i.e. to energise the relay coil)

Relay selection based on what you want to switch on and off.

This is relative straight forward so we will consider it first. While there are many different types of relays (see <http://en.wikipedia.org/wiki/Relay>), for home automation and control there are three main types that would be used.

- i) Reed relay for switching low voltage and low current devices
- ii) 110V Power relay for switching 110V mains powered devices
- iii) 240V Power relays for switching 240V mains powered devices (can also be used for 110V)

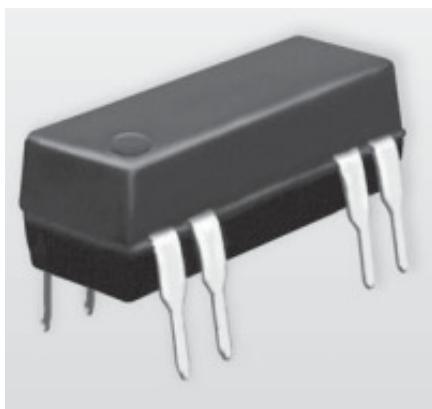
applications)

We will look at a number of configurations for connecting relay modules to Arduino boards:-

- a) Direct driven reed relays
- b) Single 5V power relays powered from the Arduino board's 5V supply
- c) Four (4) relay shields using a separate supply. The shield powers the Arduino board.
- d) Separate multi-relay 5V modules high triggered and low triggered
- e) Separate multi-relay 12V modules high triggered and low triggered
- f) Modifications for Driving a 5V relay module from a 3.3V Arduino board
- g) Solid State Relays

Reed Relays

Reed relays are small low power relays suitable for switching <100V at current <0.5Amps. While designs vary, total switched power is usually less than 10 VA (volts x Amps < 10). Examples of IC form factor reed relays are TRR-1A relays and COTO TECHNOLOGY – 8L01-05-001



Power Relays

Most relay modules sold to connect to Arduino are power relays and are rated for either 110V or 240V. The ratings are usually printed on the relay case.



Notice that the maximum DC voltage that can handled is much less then the AC rating.

This is the type of relay you will need to use of switch mains powered devices. These relays will handle most devices used in homes except the highest powered ones like room heaters, stoves,

motors. Make sure the VA (Volts x Amps) of the device you are switching on/off is less than the relay rating.

Any wiring to the mains power should only be done by a qualified electrician.

A final point, the power relays commonly use silver alloy contacts and are not suitable for switching very low currents like switching Arduino digital inputs. If you are only switching a few millamps, for example by putting a relay across a low voltage garage door push-button, you should use a reed relay (or a relay with gold contacts) as the power relays need a “[wetting current](#)” to break through the surface film resistance and will not switch very small currents reliably.

“*Normally, a switch's wetting current rating is far below its maximum current rating, and well below its normal operating current load in a properly designed system. However, there are applications where a mechanical switch contact may be required to routinely handle currents below normal wetting current limits (for instance, if a mechanical selector switch needs to open or close a digital logic or analog electronic circuit where the current value is extremely small). ... Normal silver or copper alloy contacts will not provide reliable operation if used in such low-current service!*”

(http://www3.eng.cam.ac.uk/DesignOffice/mdp/electric_web/Digital/DIGI_4.html)

Relay Module selection based power required to drive the relay.

Arduino digital outputs can only supply +5V (or +3.3V) and less than 40mA max. (typically <20mA for continuous use). So, except for some reed relays, you cannot drive the relay coil directly from the digital output since 5V power relays typically require ~100mA to operate the relay coil. This means for most (all) relay modules you need to provide a separate supply to the module in addition to the Arduino digital output which is used to control the relay switching.

We will look at a number of configurations for connecting relay modules to Arduino boards:-

- a) Direct driven reed relays
- b) Single 5V power relays powered from the Arduino board's 5V supply
- c) Four (4) relay shields using a separate supply. The shield powers the Arduino board.
- d) Separate multi-relay 5V modules high triggered and low triggered
- e) Separate multi-relay 12V modules high triggered and low triggered

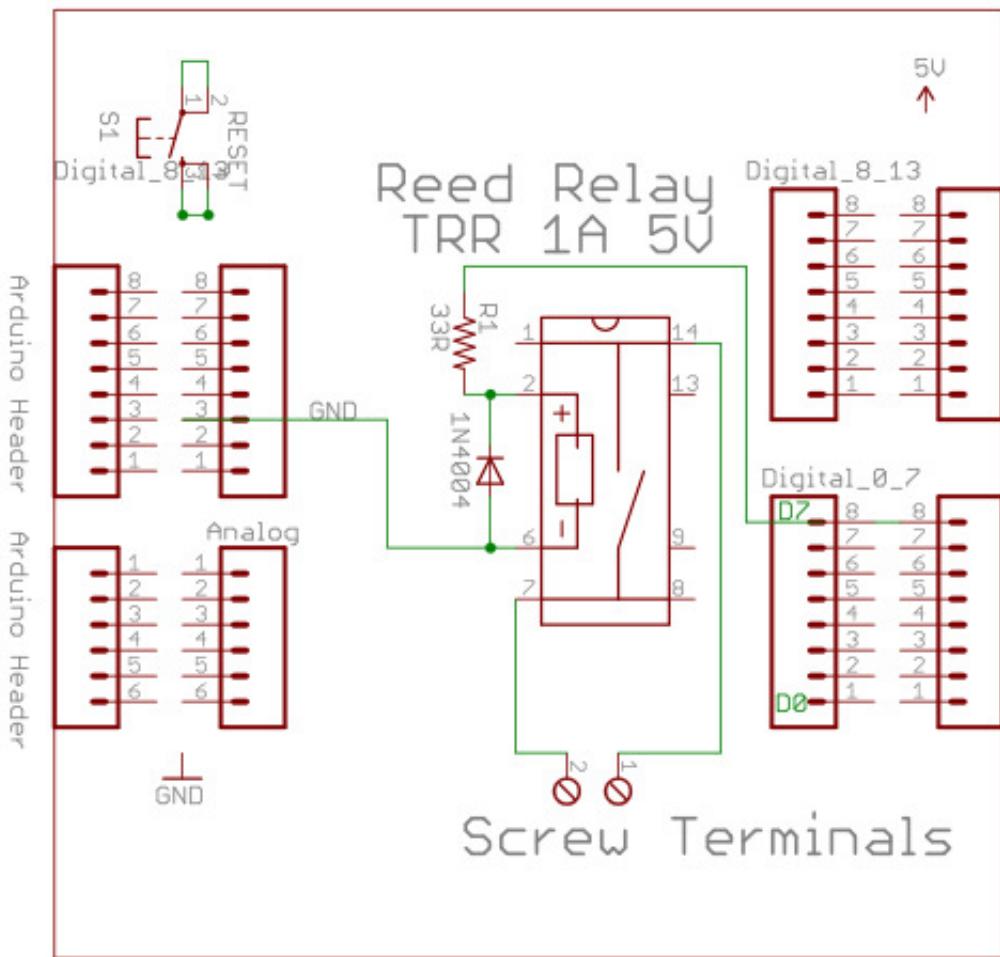
Direct Driven Reed Relays

5V Reed Relays can be driven directly from the Arduino digital output provided their coil current is less than 20mA. TRR-1A relays and COTO TECHNOLOGY – 8L01-05-001 only require 10mA each so you can easily drive 8 or more of these relays from an Arduino board supplied by a USB 500mA power supply plugged into the USB connector (The Arduino Uno has 500mA resetable fuse on the USB input).

Alternatively if the Arduino board is powered via 12V to its power plug, limit the number of 10mA reed relays to less than 6 as the on board 12V to 5V regulator limits the maximum current that can be drawn from the 5V supply to about 70mA or so.

Here is a circuit for a single 5V reed relay being driven directly from D7

D7 Direct Driven 5V Reed Relay



(c)2012 Forward Computing and Control Pty. Ltd.

Direct Drive Clamping Circuit

Two things to note:

- a) The 1N4004 diode across the coil of the relay and
- b) the 33 ohm resistor between D7 and the relay coil +

The diode is there to clamp the voltage/current spike the relay coil produces when it is switched off. Do a search on “flyback diode” for more details on this.

(http://en.wikipedia.org/wiki/Flyback_diode for example)

The 33 ohm resistor protects the D7 output from excessive current and allows the diode to do the clamping. When a digital output is turned off it does not go open circuit rather it connects the output pin to GND via a very low resistance. Without the additional 33 ohm resistance, the current spike from the coil would preferentially go back into the microprocessor via D7 to GND.

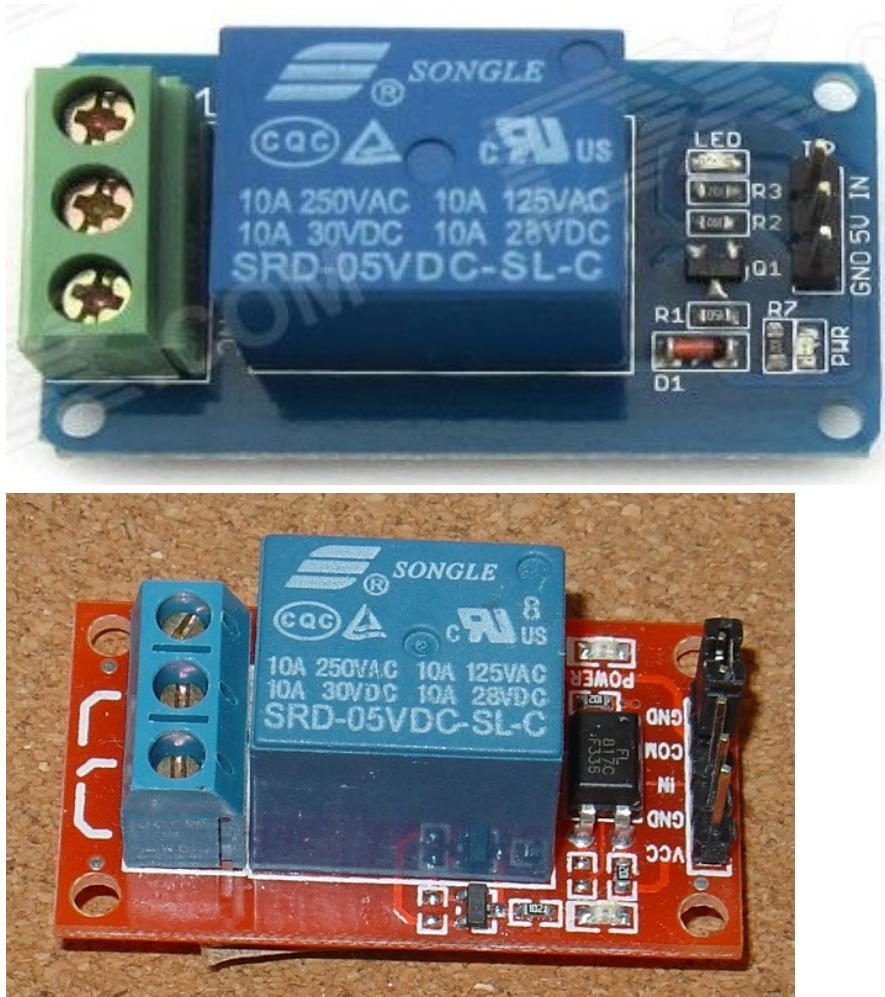
The coil resistance is about 500 ohms so when D7 is high (+5V) most of the voltage appears across the coil (I.e ~4.7V) (search for Ohm's Law or checkout [Sparkfun's tutorial](#)) The “pull in” voltage on the datasheet for the TRR-1A 5V reed relay is 3.75V which means any voltage higher than that must close the contacts so 4.7V is more than enough to ensure the relay operates.

All commercial relay modules should already have the clamping diode fitted across the relay coil.

Single 5V power relay modules powered from the Arduino board's 5V supply

There are a number of different single 5V relay modules available on-line. But most do not have any circuit diagrams available.

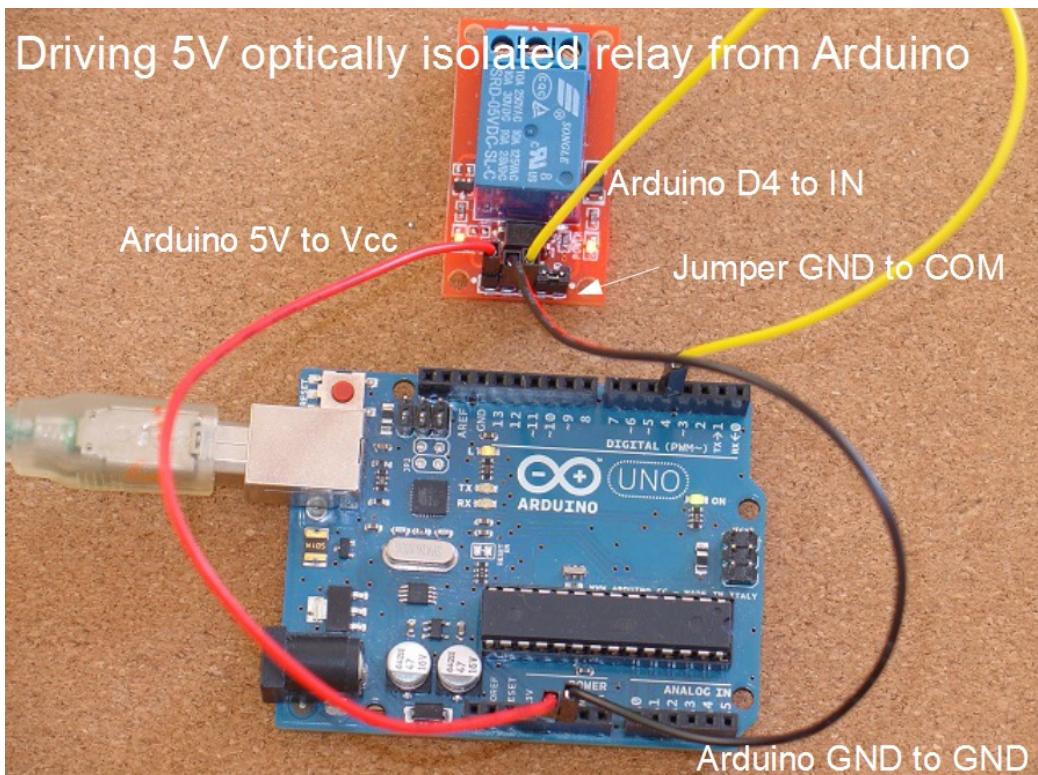
Here are two images (the left hand one is from DX.COM's website).



Both operate the relay when the input (IN) is HIGH (+5V). The right hand one as an optical isolator and separate connectors for the IN ground (COM) and the main 5V supply ground (GND). The jumper is connecting these two grounds together so you can use it just like the left hand board.

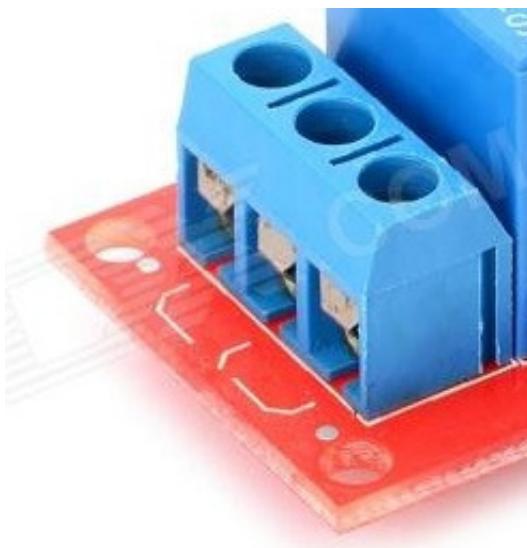
Removing the jumper lets you have isolated supplies, one for the the Arduino and another completely separate one for the relays. Separate supplies provides some extra protection against mains volts getting back into the microprocessor. But this extra protection is mostly illusory. The pcb tracks on the back of the board are quiet close and nothing will protect you against a lightning strike on the power line outside your house.

So for either board, connect the relay GND to one of Arduino's GND pins and the relay 5V or Arduino's 5V pin and the relay IN to one of Arduino digital outputs, say D4. Then when the digital output, D4, is High the relay will operate.



Note the Arduino is being powered by the 5V USB power. The relay is 5V so everything is at the same voltage.

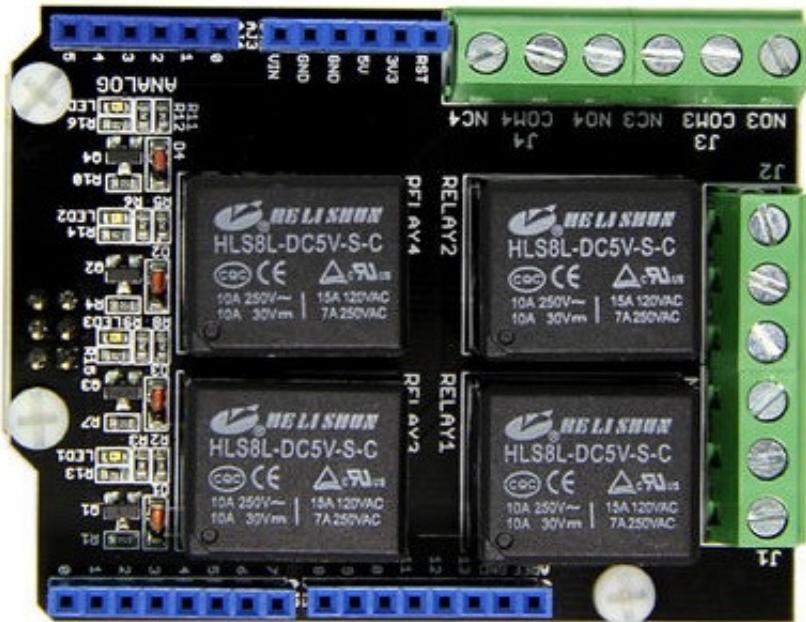
The relay's contacts consist of a Normally Closed (NC) and Normally Open (NO) connection and a COM connection. When the relay is un-powered the NC terminal is connected to the COM terminal and the NO terminal is not connected. When the relay is operated, the COM terminal switches over and is now connected to the NO terminal and NC terminal is not connected. The screw terminals are either marked NO, NC and COM or small drawing is shown like the image below. In the image below, the top terminal is the NO one the centre terminal is the COM and the bottom terminal is the NC one.



Connecting these relay modules to the Arduino's 5V pin means both the trigger current (IN) and the current to drive the relay coil are being supplied by the Arduino board. From the datasheet for the SRD-05VDC-SL-C you can read that the relay coil takes ~72mA to operate. Added to this is the current required by the IN input to trigger the coil. For the optically isolated relay that is about 4mA and perhaps a bit more for the non-isolated one. (Measure it for your module).

In any case each of these relay modules will take <100mA from the Arduino's 5V supply. So if a USB supply is used to power the Arduino, you can attach up to 4 of these modules. Indeed that is

just what this 4 channel relay shield from [Seeed Studio](#) does.

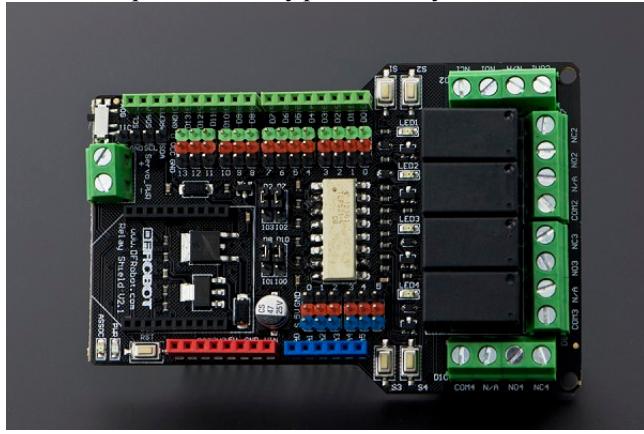


The relays are controlled by D4,D5,D6 and D7 so plugging this relay shield on top the [Bluetooth Shield](#) shown in [Design Custom menus to switch Arduino outputs on and off](#) will give will complete remote control system to switch things on and off controlled by Android using [pfodApp](#). Using the free [pfodDesigner](#), you can easily add the controls for all 4 relays.

This shield is quoted as drawing 250mA, but 300mA would be more realistic based on the datasheet for the relays. In any case it is fine if you are powering the Arduino board via a USB supply. However it is not suitable if you are using a 9V or 12V supply due to the limitations of Arduino's on-board regulator.

Four Channel Relay Shields using a Separate Supply

An example of this type of relay shield is the DFROBOT [Relay Shield for Arduino V2.1](#)



This shield can be configured in a number of ways using the jumper strips. When operating all 4 relays it should be powered by 9V applied to the screw terminal on the left of the picture.

By default the relays are controlled by D2, D7, D8 and D10. The [Itead Bluetooth Shield](#) used in [Design Custom menus to switch Arduino outputs on and off](#) uses D2 as an output so be default only 3 relays can be driven if you use that Bluetooth Shield. However the DFROBOT relay shield can be configured to use any Arduino digital output to drive the relays.

It also includes an Xbee socket so you can plug in an [RN-42XV Bluetooth or RN-XV Wifi](#) module

and configure the jumpers to work with it. **Note in this case you should cut off the reset pin from the relay shield as it is driven by the RS-232 flow control on the Xbee socket.**

Separate multi-relay 5V modules, high triggered and low triggered

If you want more than the 4 relays then there are a number of multi-relay modules available. Typically with 8 relays but some with 16 relays. SainSmart has an 8 channel module.

The specification for the SainSmart say each channel needs 15mA to 20mA to drive it. I expect this is an error as all the modules I have measured need <5mA to drive them. However I suggest you measure the actual current needed to drive the inputs of any 8 channel board you use. The Arduino microprocessor has an **absolute maximum** total digital output current of 200mA, so 8 times 20mA = 160mA is higher than I would like to run continuously.

Since both the Arduino drive voltage and the relay coil voltage is the same, +5V, there is no problem with low triggered inputs since the voltage from the relay board being pulled down by the digital output will not be more than 5V. But as will be discussed below this is not the case for 12V relay modules. It is also not the case if you are driving a 5V relay module from a 3.3V Arduino board like the Due or Fio or SparkFun's FioV3.

Separate multi-relay 12V modules, high triggered and low triggered

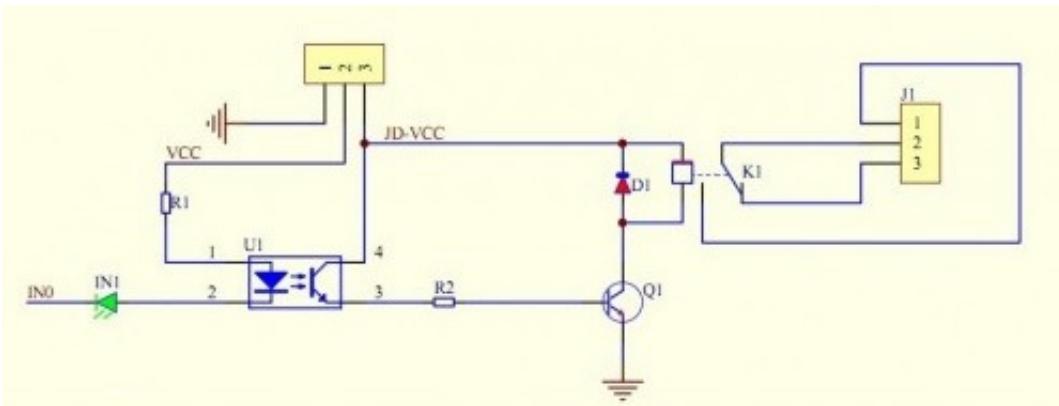
It is sometimes more convenient to use 12V relays driven by a 5V Arduino board. You may have a 12V battery supply or need to use 12V for some other part of the circuit, or the lower current used by the 12V relay coils may be a consideration. In these cases you have to be careful about how the 12V relay coil is driven by the 5V Arduino output.

As mentioned above a lot of the relay modules available do not come with circuit diagrams. Problems can occur if the relay module is low triggered. That is the relay control input is pulled to GND by the Arduino digital output pin. In that case, then when the Arduino output pin is high there may be 12V applied to it from the relay module. This may damage the Arduino microprocessor.

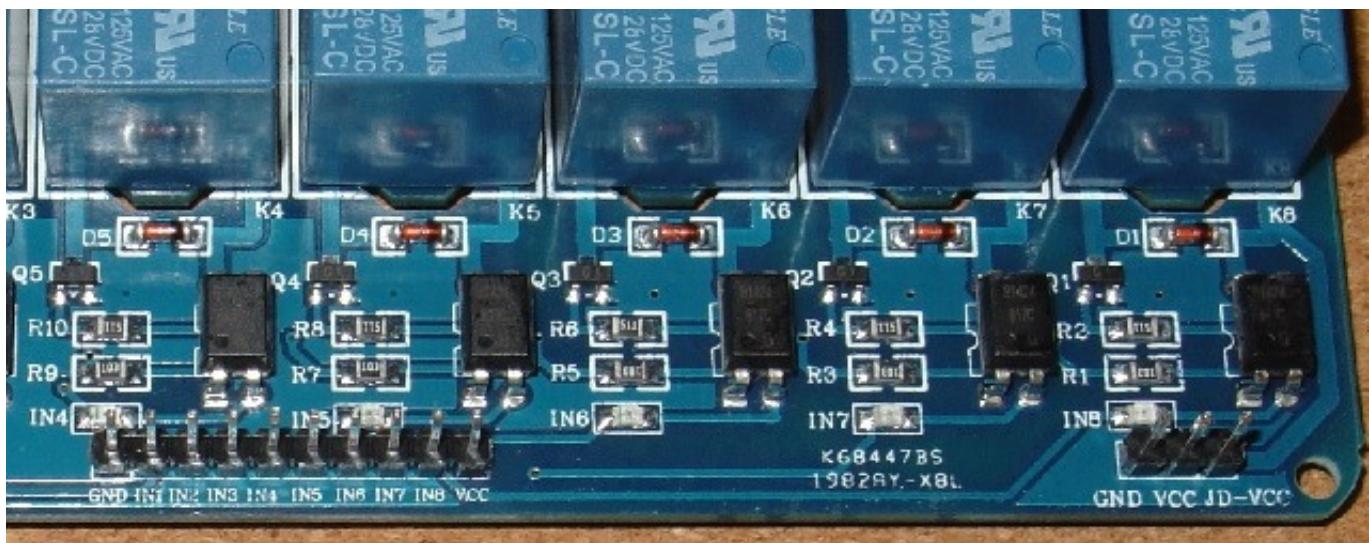
This is also an issue if you are driving 5V relay module from a 3.3V Arduino board like the Due or Fio or SparkFun's FioV3. Again the +5V from the relay board applied to the 3.3V digital output pin may damage the microprocessor.

There are two ways to avoid this problem.

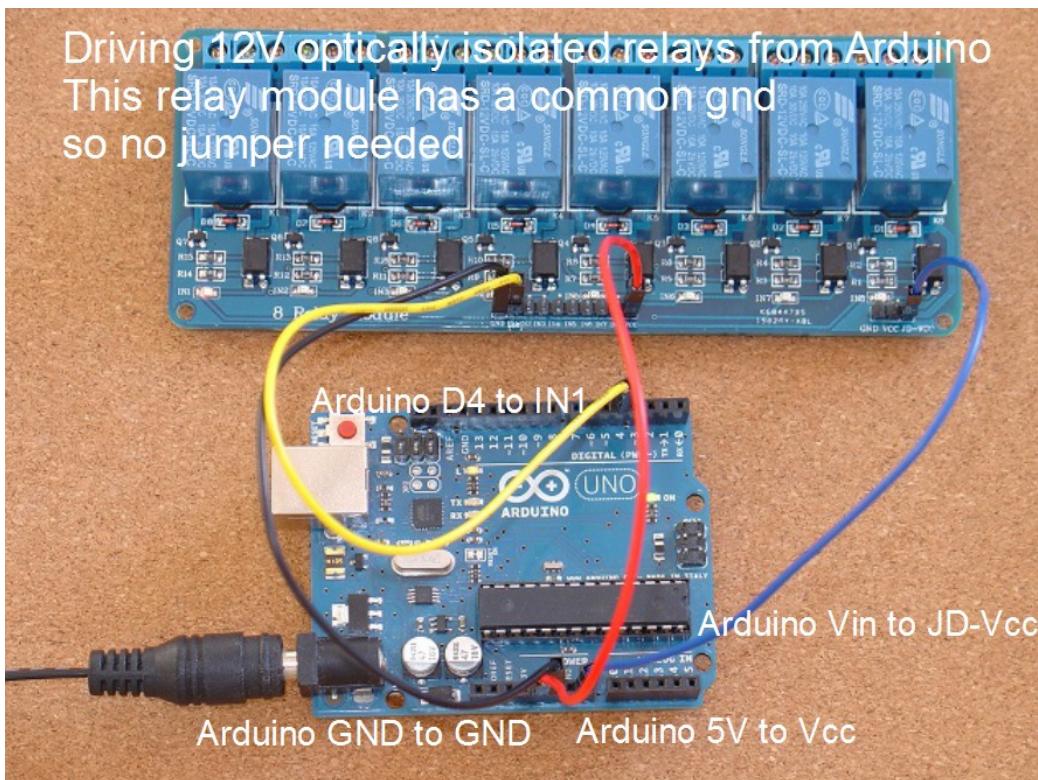
1. Use a module that is high triggered by +5V. In this case there will not be any excess voltage coming back from the module when the Arduino output is low.
2. Use an optically isolated module that has a separate trigger supply terminal in addition to the 12V supply terminals for the relay coils. In this case the Arduino +5V supply goes to the trigger supply terminal and no excess voltage will be applied to the digital outputs. Here is an indicative circuit of one channel of 8 channel relay module (This circuit from [SainSmart](#))



As you can see with an optical isolator and a separate Vcc input (+5V from Arduino) and the Arduino digital inputs are isolated from the 12V relay voltage by the optical isolator. Here is a typical optical isolated 12V 8 relay board. Notice the Gnd and VCC near the IN terminals. That VCC is for the 5V supply from Arduino. At the right hand end is the JD-VCC pin which is for the 12V for the relay coils.



Here is an example connection picture. Note that the Arduino is being powered by 12V via the plug in power. The Vin socket on the Arduino is the same voltage as the power plug, i.e. 12V, so the relay module JD-Vcc is connect to Arduino Vin to supply the 12V for the relay coils. The Arduino Vcc is 5V and is connected to the relay module Vcc to set the trigger voltage. The trigger GND is connected to Arduino GND and D4 is connected to IN1 to drive relay 1. In this module the JD-Vcc and the Vcc connections both have the same GND point so only one ground connection is needed and no jumper is needed.



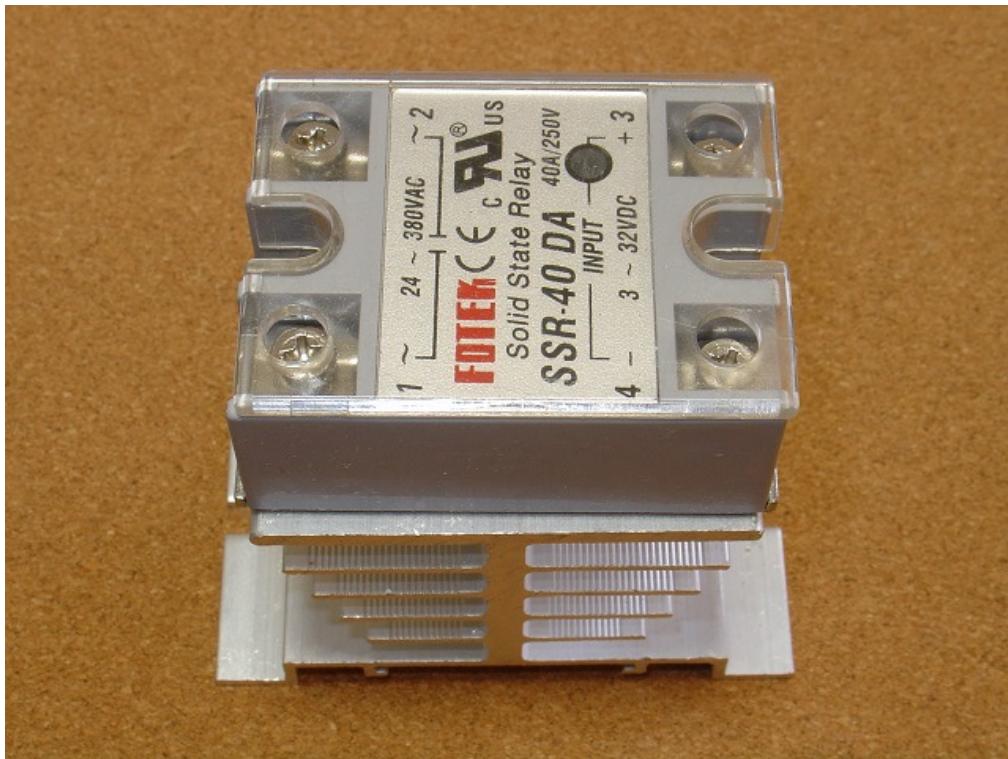
Modifications for Driving a 5V relay module from a 3.3V Arduino board

One final point about drive 5V relay modules from 3.3V Arduino boards. In addition to using one of the above two solutions to prevent the 5V being feed back to your 3.3V digital outputs, you may need to modify relay board, if the 3.3V output does not reliably drive the relay. The problem is the circuit was designed for 5V to drive the optical isolator U1, and U1 may not be getting sufficient current from your 3.3V Arduino's output.

There are two common modifications for 3.3V operation of a 5V relay board:-

1. If the drive circuit has a led in series with it, i.e. IN1 in the circuit above, just short it out by soldering a wire from one side to the other. The led was using up about 2V so shorting it out means now you only need about 3V instead of 5V to drive the relay.
2. If there is not led in series with the drive pin, then add another resistor across R1 to reduce the series resistance and so increase the current available from your 3.3V drive. R1 is typically 10K (102) so adding another 10K resistor across the existing R1 should be sufficient. This will require some careful soldering.

Solid State Relays



Solid State relays are inexpensive and can handle high current loads. The one shown above will switch a load of up to 40Amps from a 24 to 380V AC supply. The control input is DC 3 to 32 Volts. However there are a number of things to look out for when using Solid State Relays with Arduino.

- Make sure the relay is controlled by DC volts 3 volts and up. Some solid state relays AC control inputs of higher voltages.
- Choose a relay that is rated for the current you need to switch. The one in the photo is a 40Amp relay
- Make sure the relay can handle more than 1.414 times the AC line voltage. That is >340VAC for a 240V supply and >180VAC for a 120V supply.
- Always attach the relay to a suitable heat sink using a thin smear of thermal compound that completely covers the base of the relay. Solid State relays can get very hot and burn out if the heat sink is too small. For a 10amp load, the solid relay needs to get rid of about 16Watts of heat.
- Solid State relays are best suited for resistive loads like heating elements. If you are trying to turn a motor on and off a normal relay is probably a better choice.

The main advantage of Solid State relays is that if it still working after 3 months ([infantile failure](#)) it will probably work for many years.

While the relay can be controlled with 3V and so is suitable for the 3.3V arduino boards, it takes about 12mA to switch it do check that your board can supply that amount of current from a digital output. Some 3.3V boards, like LinkIt ONE, will only supply 3mA from a digital output. Arduino Uno boards can supply about 20mA.

Conclusion

There are a number of types of relays you can add to your Arduino to switch real things on and off. You need to choose a relay appropriate to your needs.

The simplest solution for on/off Home Automation is to use [pfodDesigner to design your control](#) for upto 4 switches and then use a Arduino Uno + [Bluetooth module](#) + [4 channel relay board](#) powered by a [USB power supply](#) and controlled by [pfodApp](#) and, if switching mains power, have

an electrician wire it in.

Android™ is a trademark of Google Inc. For use of the Arduino name see
<http://arduino.cc/en/Main/FAQ>

pfodDevice™ and pfodApp™ are trade marks of Forward Computing and Control Pty. Ltd.



Contact Forward Computing and Control by [email](#)

©Copyright 1996-2012 Forward Computing and Control Pty. Ltd. ACN 003 669 994