

Anites

All Things: Home Automation, Programming, Raspberry Pi, Arduino, Robotics and Linux

42 Anites Omnia :: Geeky Calendar :: What are Anites?

SUNDAY, SEPTEMBER 8, 2013

Driving a Relay using GPIO

The next Raspberry Pi project I am working on is controlling my garage door. To control the garage door I will need to use a relay. Unfortunately, I used my one and only, easy to use, and very compact, "[Relay Module](#)" with my [sprinkler project](#). I could order another, but that would take time, and I wanted to get something working over the weekend.

Luckily, I had a pack of 5V relays in my electronics component collection. I had ordered a 10 pack for \$6 off of [Amazon](#) some time ago. (*do a quick search for "5V Relay", there are numerous options*)



QIANJI JQC-3F(T73) Relay

Controlling a relay from a Raspberry Pi is not as easy as simply connecting it to a GPIO port.

SEARCH

ABOUT ME



Kenneth Burgener

Follow 133

[View my complete profile](#)

FOLLOW BY EMAIL

ARCHIVE

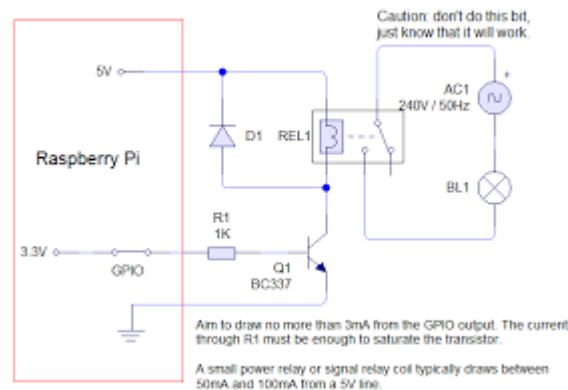
► [2017](#) (4)

► [2016](#) (1)

First, the GPIO port is only 3.3V and most small relays require 5V. Luckily the Raspberry Pi does have a 5V rail we can use, but we need a way to switch that on and off. That is where the transistor comes in. Our GPIO port will turn the transistor on and off, which will in turn turn the 5V lead to the relay on and off.

In theory this GPIO -> Transistor -> Relay are all that is needed, but of course we need a bit of protection to not fry the components (or the Pi). This involves a few resistors and a diode. The resistors will limit the current, and the diode will wrap the relay to dissipate the back current (a relay is just a coil with a magnet, and coils store energy). This mess is why the compact pre-built Relay Modules are so very nice!

I found an excellent article, by [Kevin Sangeelee](#), that provides a great explanation and a sample schematic for driving a simple relay. Kevin describes the use of the diode as: *"The diode in the circuit is there to conduct the current generated by the de-energising coil back across the coil (e.g. when switched off), allowing the power to dissipate more gradually, avoiding a voltage spike."* ([source](#))



Schematic for a relay via GPIO on the Raspberry Pi ([source](#))

I did not have all of the exact components listed in Kevin's schematic, but I did have several close components that I traded with, from my electronic component box. *(If you don't have a box for all these various small components, you really should, as you never know when you will need one).* For the transistor I used a 2N2222 A338. For the relay I used the 5V QIANJI JQC-3F(T73) relay

- [2015](#) (10)
- [2014](#) (5)
- ▼ [2013](#) (23)

► [December](#) (5)

► [November](#) (5)

► [October](#) (2)

▼ [September](#) (7)

[Garage Door Part 1 - Open Door \(GPIO Output\)](#)

[Monitoring a UPS](#)

[Driving a Relay using GPIO](#)

[Data Logging with the Internet of Things - Xively](#)

[Automated Sprinkler System First Look](#)

[Remote Control Machines](#)

[The MagPi](#)

► [August](#) (4)

LABELS

[achievement](#) (2)

[arduino](#) (2)

[automation](#) (10)

[coding](#) (1)

[data-logging](#) (2)

[development](#) (1)

[distribution](#) (2)

[electronics](#) (6)

[events](#) (2)

from Amazon. For the diode I used a generic, nondescript, one I also had in the component box. With my setup, the 1K resistor was too high to drive my transistor, so I reduced it to a 100 Ohm resistor, and this seemed to work fine. I also connected up a resistor and LED on the load side, to verify that the relay was functioning, as if the noisy relay clicking wasn't enough.

I then put together a little bit of Python code to produced this fun clicking light show:

```
#!/usr/bin/env python

import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(25, GPIO.OUT)

try:
    while True:
        print "On"
        GPIO.output(25, GPIO.HIGH)
        time.sleep(.1)
        print "Off"
```

[games](#) (1)
[google](#) (1)
[hardware](#) (1)
[howto](#) (1)
[humor](#) (1)
[ideas](#) (2)
[installation](#) (2)
[keyboard](#) (1)
[lala-pi](#) (4)
[linux](#) (2)
[magazine](#) (1)
[magpi](#) (1)
[math](#) (2)
[news](#) (3)
[power](#) (1)
[programming](#) (5)
[pytemper](#) (1)
[python](#) (6)
[raspberrypi](#) (9)
[raspbian](#) (6)
[raspbmc](#) (1)
[robots](#) (1)
[science](#) (2)
[scouting](#) (2)
[security](#) (2)
[sprinkler](#) (1)
[temperature](#) (1)
[wifi](#) (1)
[x10](#) (1)
[xively](#) (1)

```
GPIO.output(25, GPIO.LOW)
time.sleep(.1)
finally:
    print "Cleaning up..."
    GPIO.cleanup()
```

The next step is to connect it to the garage door opener, and then get some open/close sensors so I can tell when the door is open or closed. Eventually, I am going to write up a proximity application for my cell phone to automatically open the garage door on my way home from work.

Posted by [Kenneth Burgener](#) at [11:07 AM](#)

[Print](#)

Labels: [electronics](#)

No comments:

[Post a Comment](#)

Links to this post

[Create a Link](#)

[zeromq](#) (1)

CONTACT

Name

Email *

Message *

[Send](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

GLOBE



By [Kenneth Burgener](#) © 2013-2014. Picture Window theme. Powered by [Blogger](#).