pi230vDoc

Control 230v-Relais with Raspberry Pi Model B+

Sebastian Frenger

2016-07-11

Inhaltsverzeichnis

1. preamble

How to switch power over ethernet with very low costs and minimal effort. With 35€, one our of time to setup the basics you got a mighty piece of hardware to „automate your home“.

1.1. idea

Be able to switch power over ethernet based on events like dusk or dawn, temperature, anything. The pi itself just makes the relais available over network, it will be instructed to switch the reley by any (authenticated) member in the network.

1.2. limit

There is no webinterface (feel free to do it yourself), the pi does not know details of the network. It is the other way aroung: Members of the network know the pi to control him.

2. solution

Combining well known hard- and software the way i want, inspired by skiwithpete and my knowledge of linux, cron etc.

2.1. parts

name	€	buy @	why
pi	30,00	amazon	control
5v-relais	5,00	amazon	to control 230v
6x power strip	15,00	amazon	to connect 6 different 230v-devices
cable	1,50	amazon	to connect pi with relais

2.2. setup

connect to your pi over network

```
ssh pi@ipOfPi
```

wiringPi to control gpios very handy

```
pi@raspberryPi: sudo aptitude install wiringpi
```

gpio should now be able to list your current (untouched) gpio-configuration

```
pi@raspberryPi: gpio readall
```

should show you table like this:

```
+-----+-----+---------+------+---+--B Plus--+---+------+---------+-----+-----+
| BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
+-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
|     |     |    3.3v |      |   |  1 || 2  |   |      | 5v      |     |     |
|   2 |   8 |   SDA.1 |   IN | 1 |  3 || 4  |   |      | 5V      |     |     |
|   3 |   9 |   SCL.1 |   IN | 1 |  5 || 6  |   |      | 0v      |     |     |
|   4 |   7 | GPIO. 7 |   IN | 1 |  7 || 8  | 1 | ALT0 | TxD     | 15  | 14  |
|     |     |      0v |      |   |  9 || 10 | 1 | ALT0 | RxD     | 16  | 15  |
|  17 |   0 | GPIO. 0 |   IN | 0 | 11 || 12 | 0 | IN   | GPIO. 1 | 1   | 18  |
|  27 |   2 | GPIO. 2 |   IN | 0 | 13 || 14 |   |      | 0v      |     |     |
|  22 |   3 | GPIO. 3 |   IN | 0 | 15 || 16 | 0 | IN   | GPIO. 4 | 4   | 23  |
|     |     |    3.3v |      |   | 17 || 18 | 0 | IN   | GPIO. 5 | 5   | 24  |
|  10 |  12 |    MOSI |   IN | 0 | 19 || 20 |   |      | 0v      |     |     |
|   9 |  13 |    MISO |   IN | 0 | 21 || 22 | 0 | IN   | GPIO. 6 | 6   | 25  |
|  11 |  14 |    SCLK |   IN | 0 | 23 || 24 | 1 | IN   | CE0     | 10  | 8   |
|     |     |      0v |      |   | 25 || 26 | 1 | IN   | CE1     | 11  | 7   |
|   0 |  30 |   SDA.0 |   IN | 1 | 27 || 28 | 1 | IN   | SCL.0   | 31  | 1   |
|   5 |  21 | GPIO.21 |   IN | 1 | 29 || 30 |   |      | 0v      |     |     |
|   6 |  22 | GPIO.22 |   IN | 1 | 31 || 32 | 0 | IN   | GPIO.26 | 26  | 12  |
|  13 |  23 | GPIO.23 |   IN | 0 | 33 || 34 |   |      | 0v      |     |     |
|  19 |  24 | GPIO.24 |   IN | 0 | 35 || 36 | 0 | IN   | GPIO.27 | 27  | 16  |
|  26 |  25 | GPIO.25 |   IN | 0 | 37 || 38 | 0 | IN   | GPIO.28 | 28  | 20  |
|     |     |      0v |      |   | 39 || 40 | 0 | IN   | GPIO.29 | 29  | 21  |
+-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
| BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
+-----+-----+---------+------+---+--B Plus--+---+------+---------+-----+-----+
```

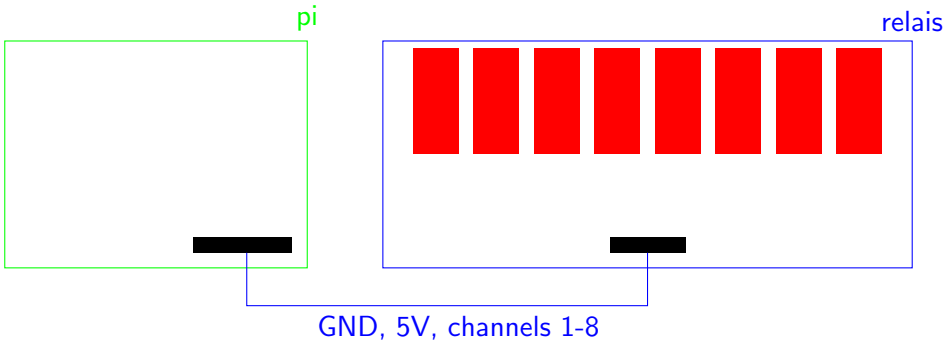
save that table somewhere, you will need it later

shutdown the pi, it's time to plug the relais together with it in the next chapter.

```
pi@raspberryPi: sudo shutdown -h now
```

unplug display, keyboard and mouse and the power from your pi.

3. plug



connect cables: ground to ground, channels to wiringPi-gpio, and 5V to 5V, check your result twize. We use gpios 1 to 8 here[1].

document the connections which channel on the relais is controlled by wich gpio, especially if you are not using wiringPi 1 to 8. You need that later.

4. init

When pi boots up all gpios (1-8) should be set „mode“ to „out“ and „power“ to „off“ (relais are not switched by pi)[2]

power on your pi again, connect by

```
ssh pi@ipOfPi
```

test initialization by

```
pi@raspberryPi: for i in {1..8}; do gpio write $i 1; gpio mode $i out;
```

after that all 8 channels should not glow red - showing they are off, or, in same state as before, not controlled at all. Later, when you switch them on, the red light glows. You will hear the sound each time a relais switched state from on to off or vice versa.

cron is your friend to initialize the gpios each time the pi is powered on.

```
pi@raspberryPi: crontab -e
```

@reboot as magic word is executed - surprise - at reboot. Add a line:

```
@reboot for i in {1..8}; do gpio write $i 1; gpio mode $i out; done
```

and save

5. test switching with 5v

Time to play! Now where the pi and its gpios are initialized correct on each boot, it is time to test switching the channels from your „fat“ client.

power on relais 3 (via channel 3), so it switches

```
ssh pi@ipOfPi 'gpio write 3 0'
```

power off to switch it back

```
ssh pi@ipOfPi 'gpio write 3 0'
```

display all from gpio-table mentioned earlier

```
ssh pi@ipOfPi 'gpio readall'
```

display relais power state on/off (0/1) for the 8 channels.

```
ssh pi@ipOfPi 'for i in {1..8}; do echo -n "$i: "; gpio read $i; done'
```

[1] that makes it easy - wiringPi 3 is channel 3 and so on, there is no more mapping

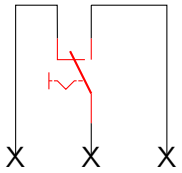
[2] some gpios defaults to mode IN , some gpios are powered by default, just lets put them all in same state.

6. Build 230v-circuit

shutdown the pi, disconnect anything but the relais, it's time to plug the relais together with 230v-circuits.

```
pi@raspberryPi: sudo shutdown -h now
```

6.1. Single relais

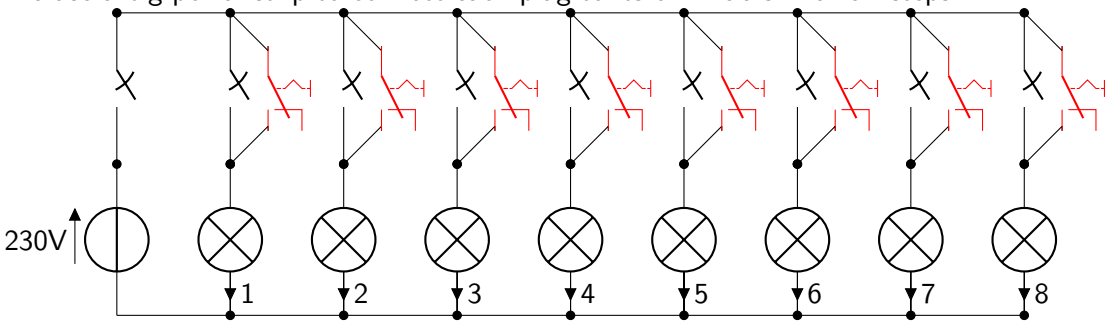


Note there are 3 ways of input on the relais, but just 2 ends of a cable (one coming from the 230v power strip, one going back to 230v power strip). Check the relais onboard documentation, which shows state when relais is powered off. Do you want power enabled if pi is down, or disabled, not controlling anything? Later, when pi is up, you may switch as you want. This can get interesting on reboots or when you - whyever - disconnect pi from relais.

6.2. 8x power strip

Note that the 230v-circuit should have it's own ground, power strips already have them build in when using default plugs.

We use a big power strip to connect each plug to its own relais in a few steps.



7. test switching with 230v

When all connection you want to use are done, reconnect power and ethernet to the pi, boot it up and repeat the test from above. E.g. Connect a desk lamp to test your results. Depending on how you connected the 2 230v-cables from each power strip to the relais, switching the relais „on“ (give 5v power to the relais itself) may result in disconnecting the 230v-circuit, or vice versa. Test it out.

8. building jobs

using cron, at, or whatever event you want it is easy to switch power:

8.1. easy cronjobs

We switch christmas-lights by cronjobs 4 times every day: 4:56 (on) to dawn (off), and dusk (on) to 22:00 (off). In the meantime we can switch as often as we want, as long as we want. There is no job running every minute checking or switching state based on dusk or dawn.

```
me@server: cat lightsOn.bsh
#!/bin/bash
ssh pi@ipOfPi 'gpio write 1 0'
```

```
me@server: cat lightsOff.bsh
#!/bin/bash
ssh pi@ipOfPi 'gpio write 1 1'
```

```
me@server: cat switch.bsh
#!/bin/bash
cd "$( cd "$( dirname "$BASH_SOURCE[0]" )" && pwd )"
on="bash /home/me/daylight/lightsOn.bsh"
off="bash /home/me/daylight/lightsOff.bsh"
dawn=`java -jar daylight.jar -lat '50.12345' -lon '10.12345' -getDawn`
dusk=`java -jar daylight.jar -lat '50.12345' -lon '10.12345' -getDusk`

echo "$on" | at 4:56
echo "$off" | at "$dawn"

echo "$on" | at "$dusk"
echo "$off" | at 22:00
```

```
me@server: crontab -l
0 1 * * * bash /home/me/daylight/switch.bsh
```

8.2. change state for 4 hours

```
ssh pi@ipOfPi 'gpio write 1 0'
atq now + 4 hours > ssh pi@ipOfPi 'gpio write 1 1'
```

8.3. discover more

motion switch state based on motion detected.

noise switch light on when you hear someone, mic

temperature switch vents on when it get't too hot, temp

network-member switch state (unlock door) when your smartphone connects to your wifi

christmas-lightning fine controlled, inspired by sunrise-sunset api

anything feel free to combine with other sensors on adafruit.

A. preparation

A.1. raspbian

Install raspbian on the sdcard from your „fat client“ before pluggin in the pi.

download os-image [3]

write image to sdcard [4] in short[5]:

[3] for pi-sdcard from raspberrypi.org
[4] as described documentation, for linux
[5] sdc(1) is an example!, be sure to chose the right one

```
df -h
umount /dev/sdX1
dd bs=4M if=2015-11-21-raspbian-jessie.img of=/dev/sdX
sync
```

resize sdcard-partition, optionally[6]

first time boot the pi [7].

A.2. network

ssh-key -generation [8], in short:

```
pi@raspberryPi: passwd
pi@raspberryPi: ssh-keygen
```

ip : note the ip the pi has got in your network

```
pi@raspberryPi: ifconfig
```

A.3. remote

Configure the pi remotely from your „fat“ client, set up a single-sign-on with ssh from your „fat“ client:

ssh-copy-id copy the public id of your „fat“ client to the pi

```
ssh-copy-id pi@ipOfPi
```

[6] with the tool of your choice, e.g. gparted
[7] by plugging the sdcard, display, lan, keyboard and mouse and at last power into it
[8] to reach the pi on a secure way with ssh over network. Set a password for the user „pi“ and generate a ssh-key