

15 MINUTES FROM 0 TO RPI CLUSTER

2016-06-11 @ PI AND MORE, TRIER

Christian Prior

SLIDES AVAILABLE

[http://www.helotism.de \ ↵
/business/marketing/presentation/piandmore_2016-06-11](http://www.helotism.de \ ↵ /business/marketing/presentation/piandmore_2016-06-11)

THIS TALK

Fast-forward through an installation

Connect to an elasticsearch cluster

ABOUT ME

business administration guy

20 years of Linux enthusiasm

Currently working as a Backlevel Software Support Engineer
for a retail solution

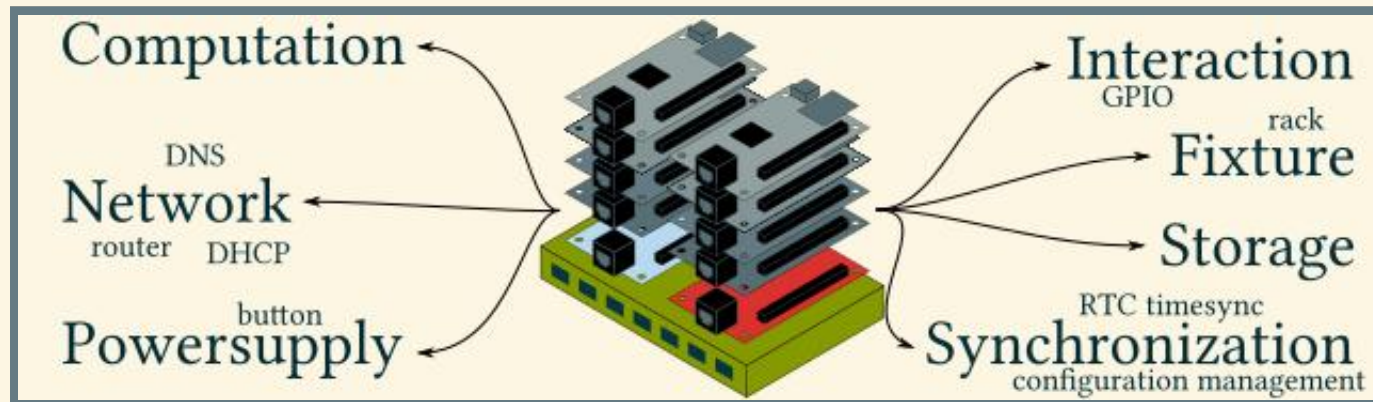
/me ♡♡♡ complexity!

<https://github.com/cprior>

HELOTISM

<https://github.com/helotism/helotism>

Installer script for a 250€-cluster solving basic sysadmin requirements



BILL OF MATERIALS

Show 4 ▼ entries

Search:

count ▲	name	price ▼
1 pcs	USB power hub	40,57 EUR
1 pcs	Switch	24,75 EUR
1 pcs	USB Ethernet adapter	23,37 EUR
1 pcs	Power Switch	15 GBP

Showing 1 to 4 of 9 entries

Previous

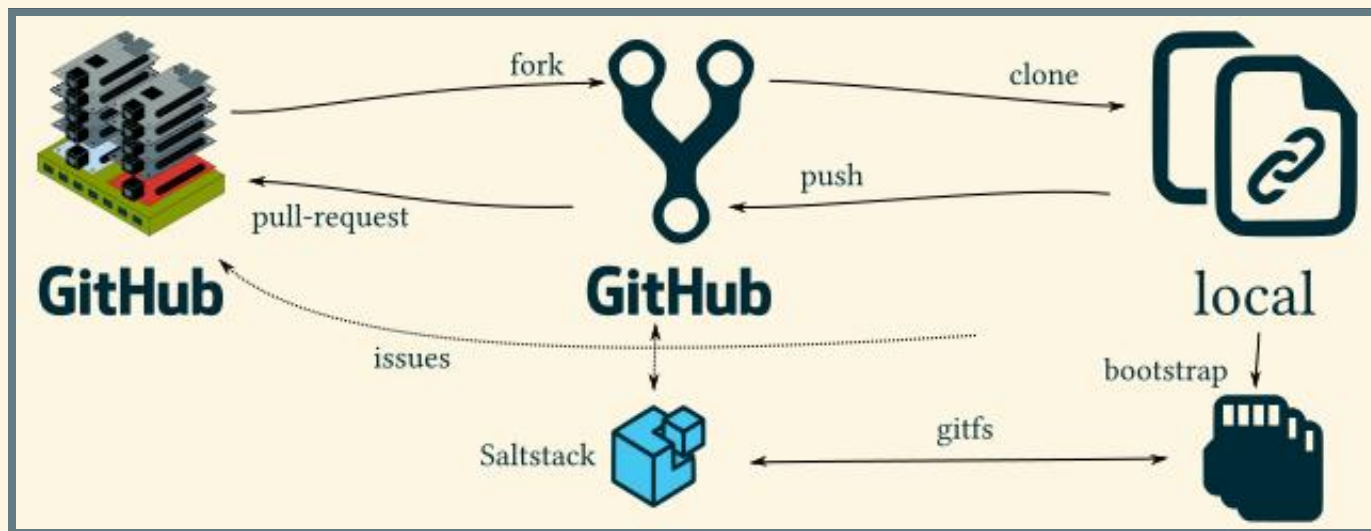
1

2

3

Next

BOOTSTRAP SCRIPT



BASIC CONFIG ITEMS

variable name	getopts parameter
__GITREMOTEORIGINURL	-r
__GITREMOTEORIGINBRANCH	-g
__COUNT	-c
__MASTERHOSTNAME	-m
__HOSTNAMEPREFIX	-n
__NETWORKSEGMENTCIDR	-s
__FQDNNAME	-d

INNER WORKINGS

getopts	default
-r	https://github.com/helotism/helotism.git
-g	master
-c	2
-m	axle
-n	spoke0
-s	10.16.6.1/24
-d	wheel.example.com
-b	/dev/null

POWERSUPPLY

- prevent brown-out
- one switch for all boards

POWER CONSUMPTION

Rule of thumb:

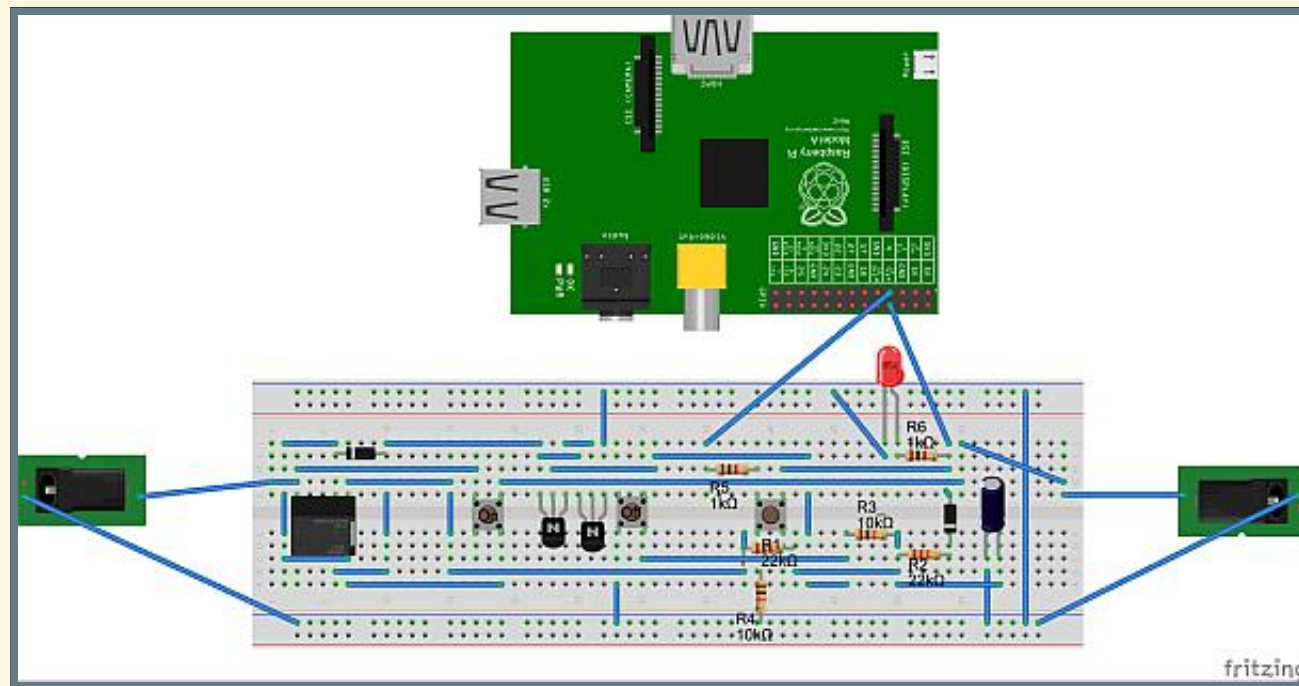
- 1 Pi idle == 2.5W ($5V * 0,5 A$)
- 1 Pi under load, no USB == 5W ($5V * 1 A$)

Caveat: GPIO-pins are no USB ports ;)

- 5V passed straight through from USB
- 3.3V rail max 50mA
- GPIO pins 16ma in total

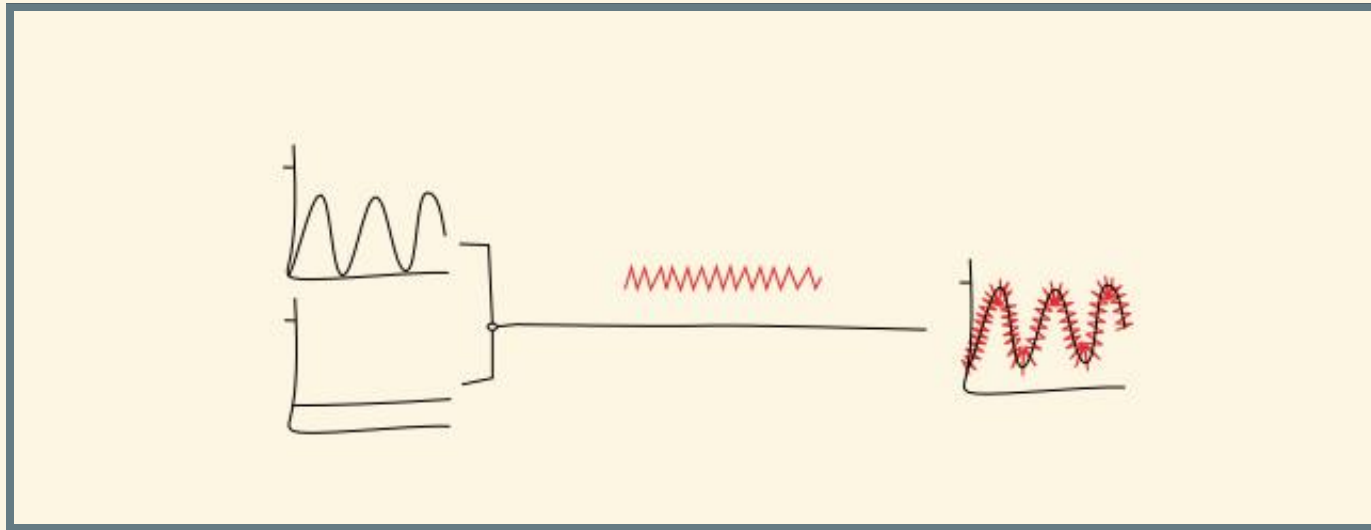
ON-OFF-SWITCH

- hardware
- code

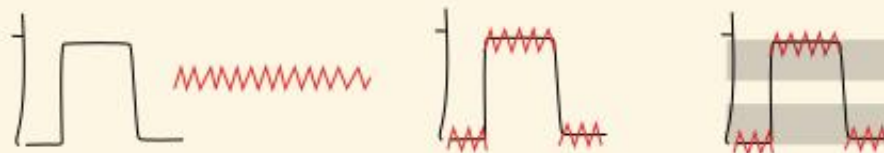


raspberrypi-geek.com On-Off-Switch

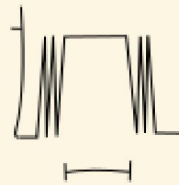
NOISE



DISCRETIZATION



DEBOUNCING



PYTHON LOGIC...

```
def handleInterrupt(args):
    button20.pressed = True #that's a fact, but...
    button20.pressed_debounced = False
    interrupted_at = datetime.datetime.now()
    debounce_until = interrupted_at + datetime.timedelta(0,3)

    while True:
        if (datetime.datetime.now() > debounce_until):
            journal.send('GPIO20 pressed DEBOUNCED.', FIELD2='GPIO20')
            //do something
            button20.pressed_debounced = True
            return
        else:
            journal.send('GPIO20 pressed, not debounced', FIELD2='GPIO20')
            time.sleep(0.5) #inside this interrupt handler only
```

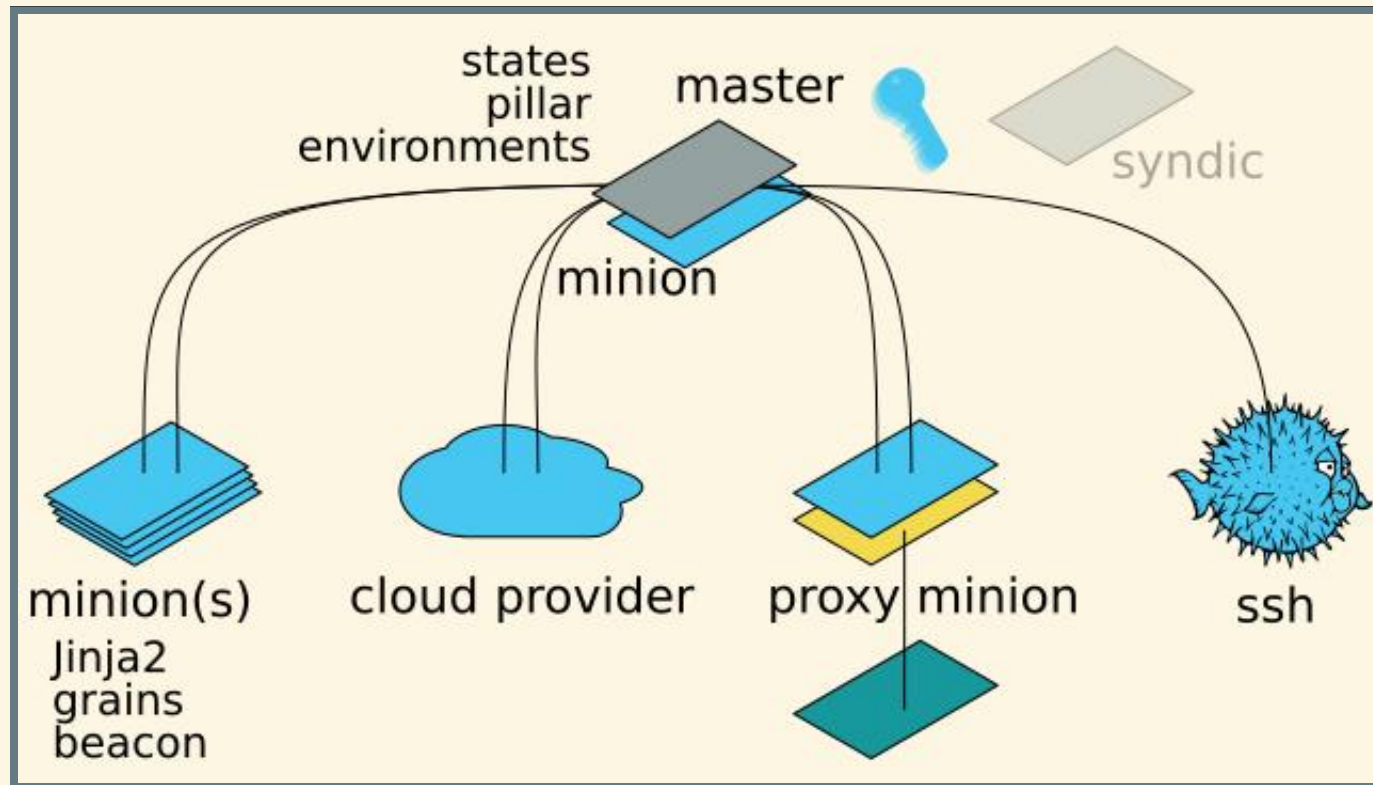

...AND SYSTEMD DAEMONIZATION.

```
[Unit]
Description=An GPIO interrupt listener
After=local-fs.target

[Service]
Type=simple
ExecStart=/bin/bash -c 'cd /opt/helotism/powersupply-env; \
                        source bin/activate; \
                        python ./mraa_interrupt.py'

[Install]
WantedBy=multi-user.target
```

SALTSTACK ECOSYSTEM



SALTSTACK TOP FILE

```
base:                # environment
  'web*':            # targeted minions
    - apache         # state file 'apache.sls'
```

- the top.sls is a special state file as "entry point" into the fileservers
- "apache" references ./apache.sls file

SALTSTACK STATE FILE

```
#apache.sls
{% if grains['os'] == 'Debian' %}
apache: apache2
{% elif grains['os'] == 'RedHat' %}
apache: httpd
{% endif %}
```

- Jinja2 template language
- one should read the fine manual:
<http://jinja.pocoo.org/docs/dev/>

SALTSTACK ENVIRONMENTS

```
file_roots:  
  dev:  
    - /srv/salt/dev  
  base:  
    - /srv/salt
```

- environments are configured in the master config file

SALTSTACK FILESERVER

```
fileserver_backend: #first filename match wins
  - roots
  - git

gitfs_remotes:
  - git://github.com/example/first.git
  - https://github.com/example/second.git
    - root: salt          #subdirectory
    - mountpoint: salt://sub/dir
    - base: myTag05        #git branch
  - file:///root/third

#top_file_merging_strategy: merge #same
#env_order: ['base', 'dev', 'prod']
```

- these are powerful configuration mechanisms:
"infrastructure as code" served from a Git repo
- many ways to segment or override

SAMPLE SALT USAGE

```
#remote execution?  
salt '*' cmd.run 'uname -a'
```

```
#listing and accepting keys  
salt-key -L  
salt-key -A
```

```
#salt.modules.test.ping  
salt '*' test.ping
```

```
#targeting by grains  
salt -G 'os:(RedHat|Debian)' test.ping
```

```
#more sound than test.ping  
salt-run manage.up
```

```
#apply common.sls on all (accepted) minions  
salt '*' state.sls common  
#This is the "endgame" in salt  
salt '*' state.highstate  
#remote execution!  
salt '*' cmd.run 'uname -a'
```

DEMO

...

QUESTIONS?

Fork it on GitHub!

Issues welcome.