YET ANOTHER RASPBERRY PI CLUSTER

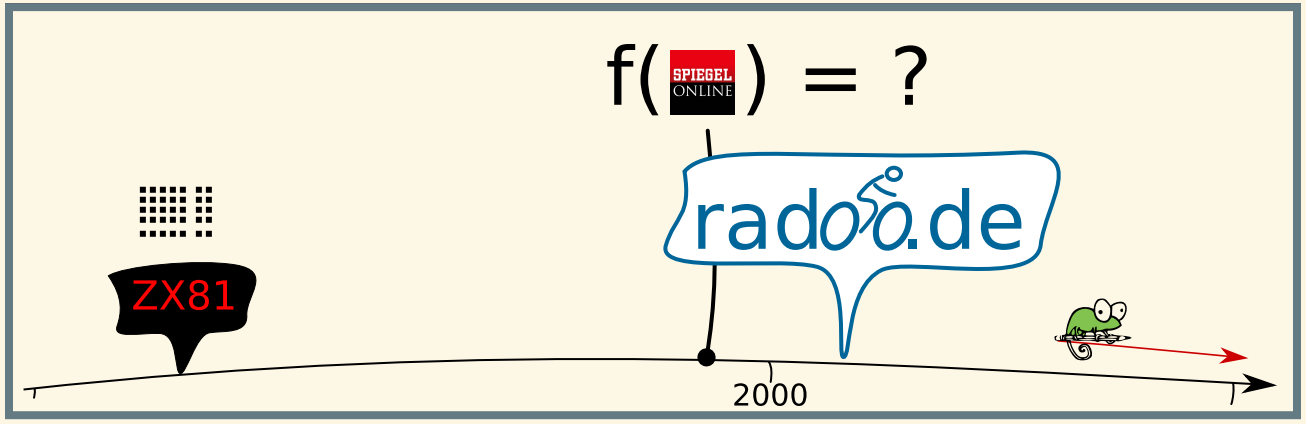
THE SLIDES OF A LECTURE/PRESENTATION ON 2016-03-20 @ CHEMNITZER LINUX TAGE

Christian Prior

SLIDES AVAILABLE

http://www.helotism.de \ ⏎
/business/marketing/presentation/CLT_2016-03-20

ABOUT ME



https://www.facebook.com/profile.php?id=100010639868228

https://github.com/cprior

CONTENT OF TODAY'S PRESENTATION

[Maker] CNC, CAD&CAM development boards

power supply [SysAdmin]

Config Mgt monitoring

[Data Analysis] data provider

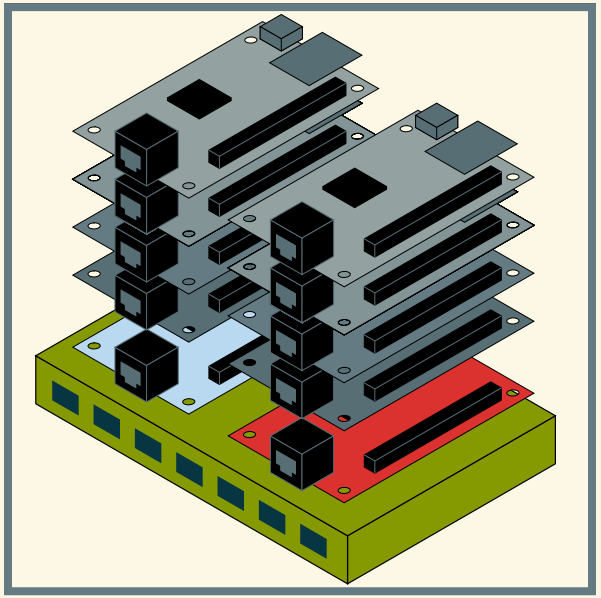
visualization [Documentation]

HELOTISM

Wiktionary.org: Helotism (zoology): A form of mutualism in which one species is forced to perform tasks for another, for their mutual benefit.

THE GOALS

- keeping up with IT changes
- solid Linux sysadmin skills are the foundation for "Big Data"
- getting most out of these boards
- getting ahead of the complexity curve



THE ROADMAP

Show entries Search:

ID	iteration	completion
1	bootstrap	100
2	init	70

Showing 1 to 2 of 7 entries

First Previous 1 2 3 4 Next

Last

THE REPO

https://github.com/helotism

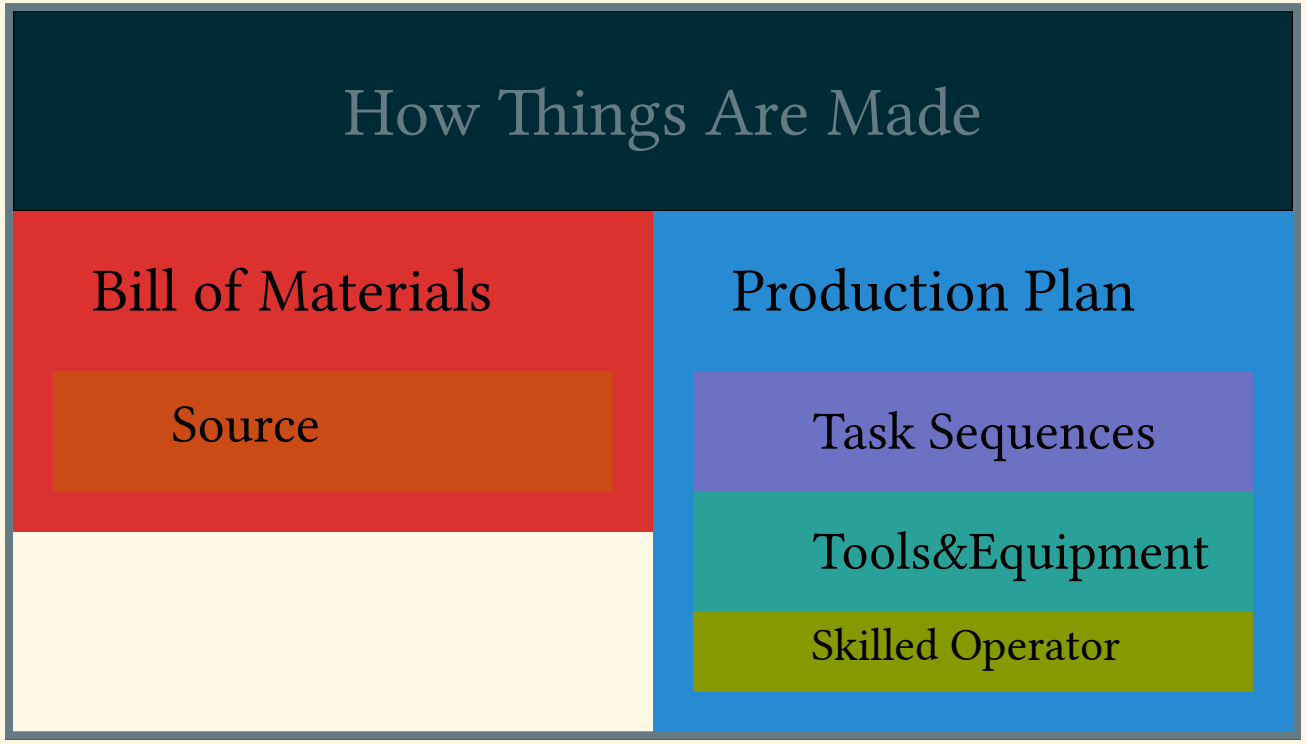
- all-in one repo
  - hardware
  - software
  - promotion
- Issues welcome! ;)

JEKYLL WEBSITE

helotism.de

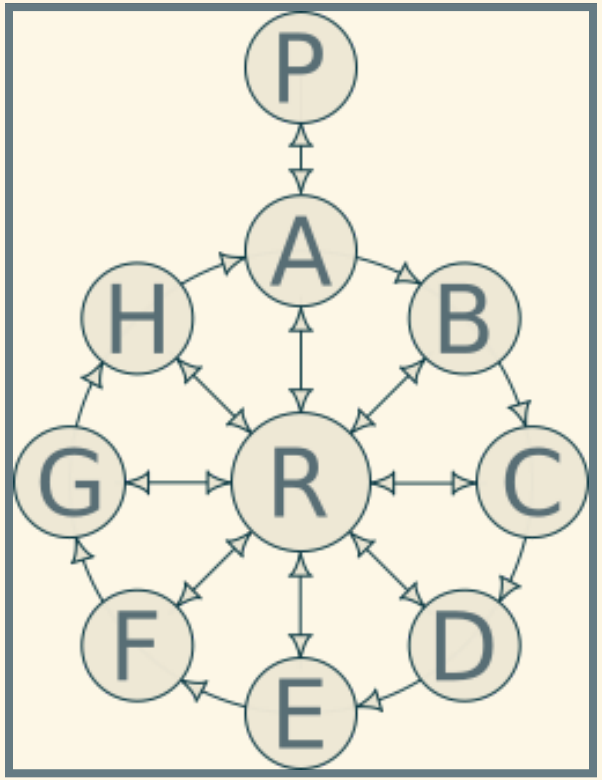
```
#_config.yml
destination:  ./business/marketing/website/_site
```

ERP

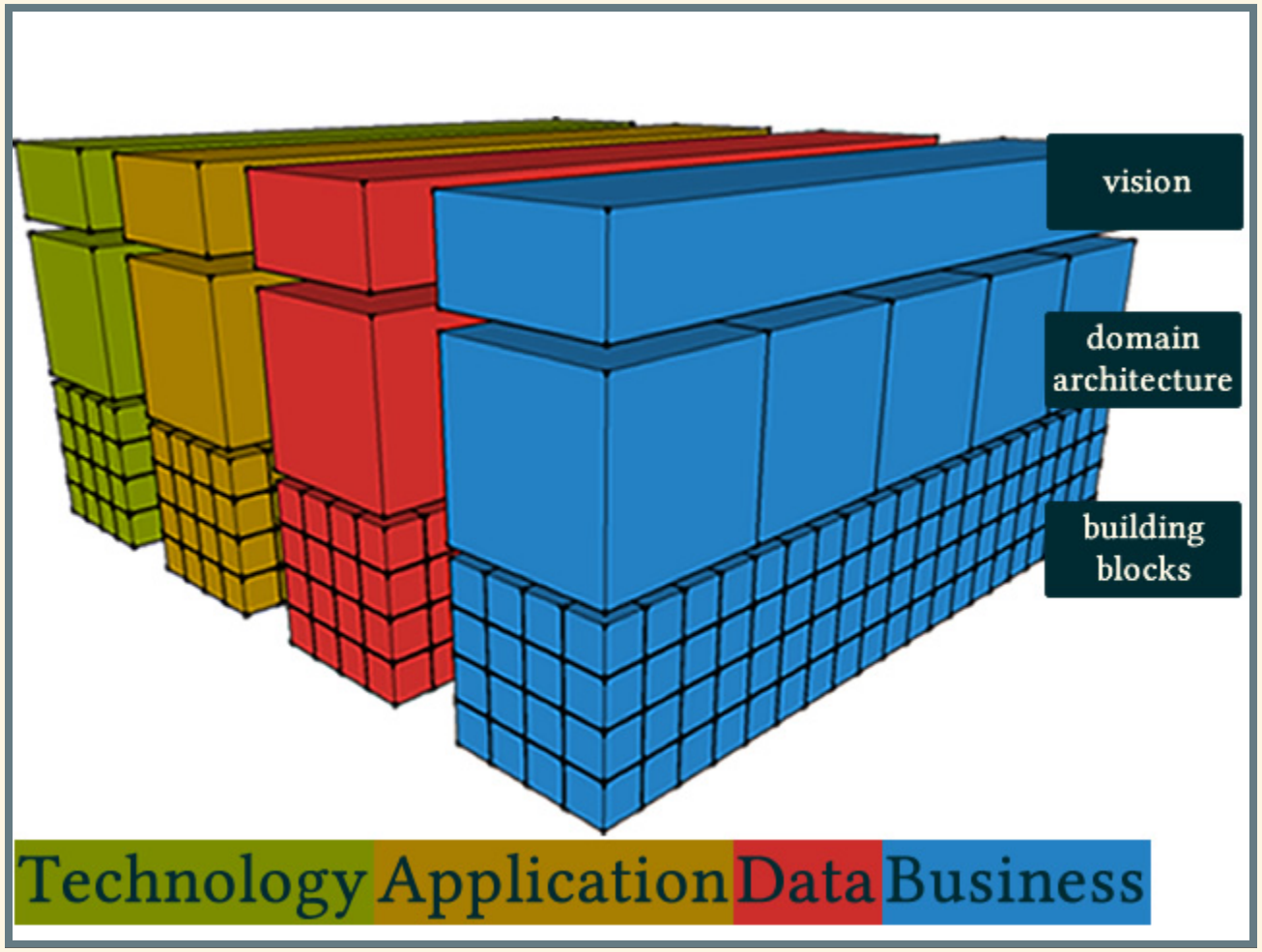

How Things Are Made
Bill of Materials
Source
Production Plan
Task Sequences
Tools&Equipment
Skilled Operator

TOGAF: ARCHITECTURE DEVELOPMENT METHOD

- P Prerequisites
- A Vision
- B Business Plan/Action
- C Information System Plan/Actual
- D Technology Plan/Actual
- E Opportunities and Solutions
- F Migration Planning
- G Implementation Governance
- H Change Management
- R Requirements



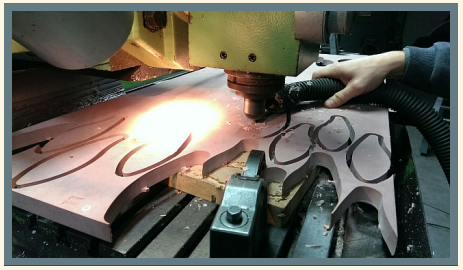
TOGAF: "DOMAINS" & BUILDING BLOCKS



MAKER

- DXF
- chipping vs. 3D-printing
- CAD4c Computer-Aided Design for clamping

ENTHUSIAST'S SMALL BATCH MANUFACTURING






DEVELOPMENT BOARDS

The market for development boards is confusing: https://en.wikipedia.org/wiki/Comparison_of_single-board_computers is a good overview.

DEVELOPMENT BOARDS: COMMON FEATURES

- Computation: Processor and Memory
- Communication through Ethernet/…
- Powersupply: Consumption, buttons
- Interaction via GPIO
- Fixture: Mounting holes and dimensions
- Storage: SD cards and beyond
- Synchronization: RTC time

DIMENSIONS

-> see repo

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

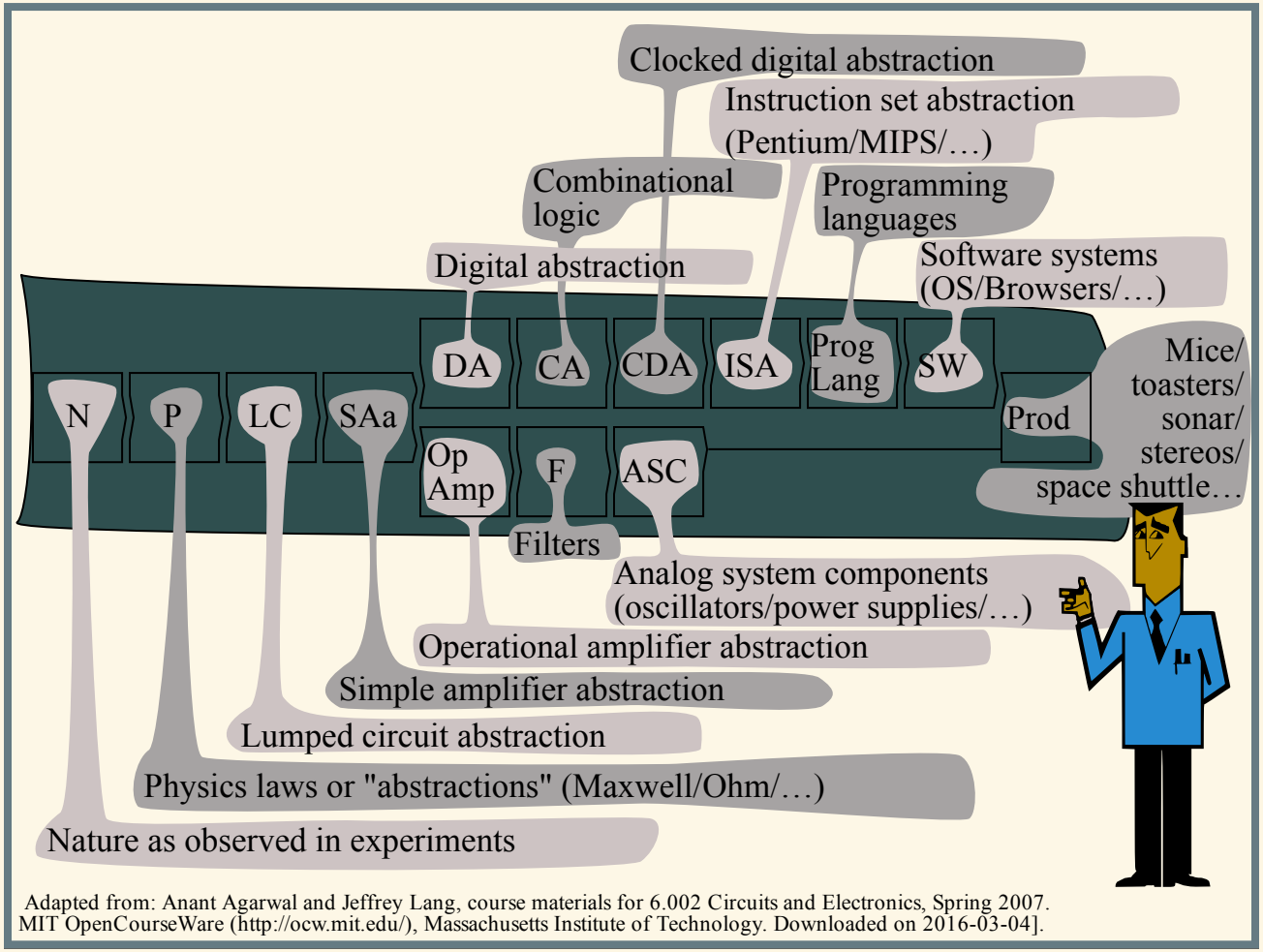
MINNOWBOARD

UEFI

OPERATING SYSTEM

1. archlinuxarm.org
   - all saltstack dependencies met
2. archlinuxarm.org
   - up-to-date systemd (229-3 in march 2016)
3. archlinuxarm.org
   - pre-compiled ;)
4. Raspbian/DietPi/Debian Jessie
   - Raspbian "Jessie" December 2015, systemd "216"

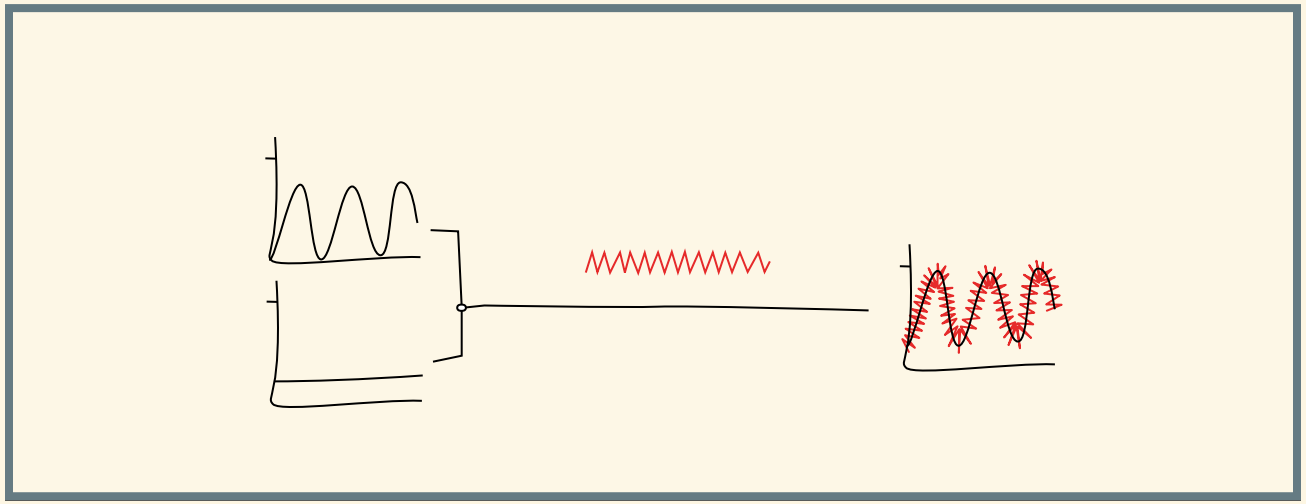
CIRCUITS



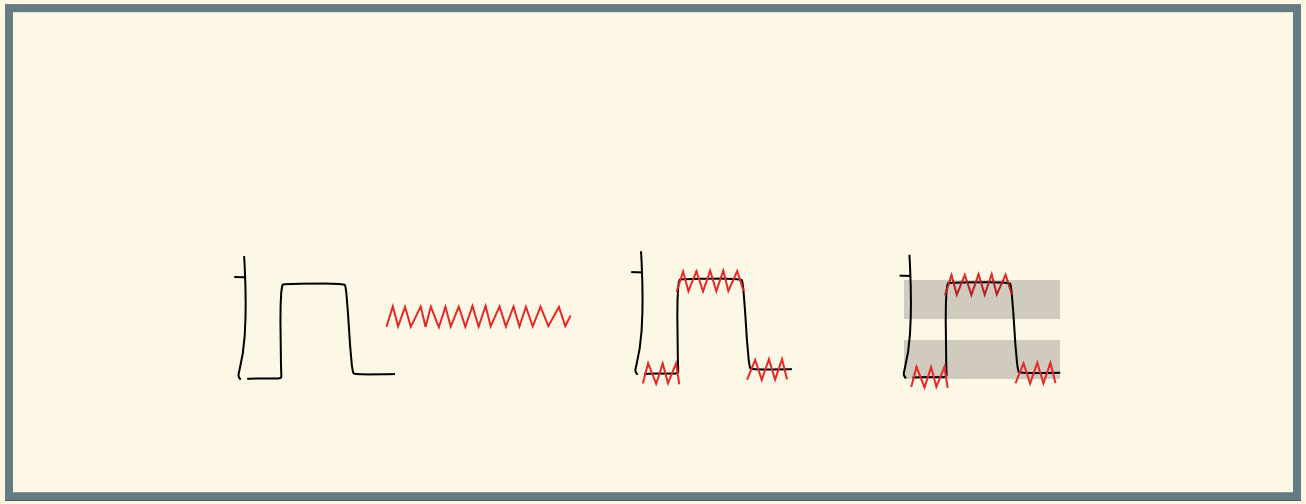
Adapted from: Anant Agarwal and Jeffrey Lang, course materials for 6.002 Circuits and Electronics, Spring 2007.
MIT OpenCourseWare (http://ocw.mit.edu/), Massachusetts Institute of Technology. Downloaded on 2016-03-04].

ALLES WEGABSTRAHIEREN!

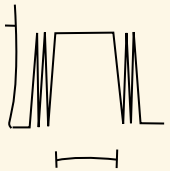
NOISE



DISCRETIZATION



DEBOUNCING

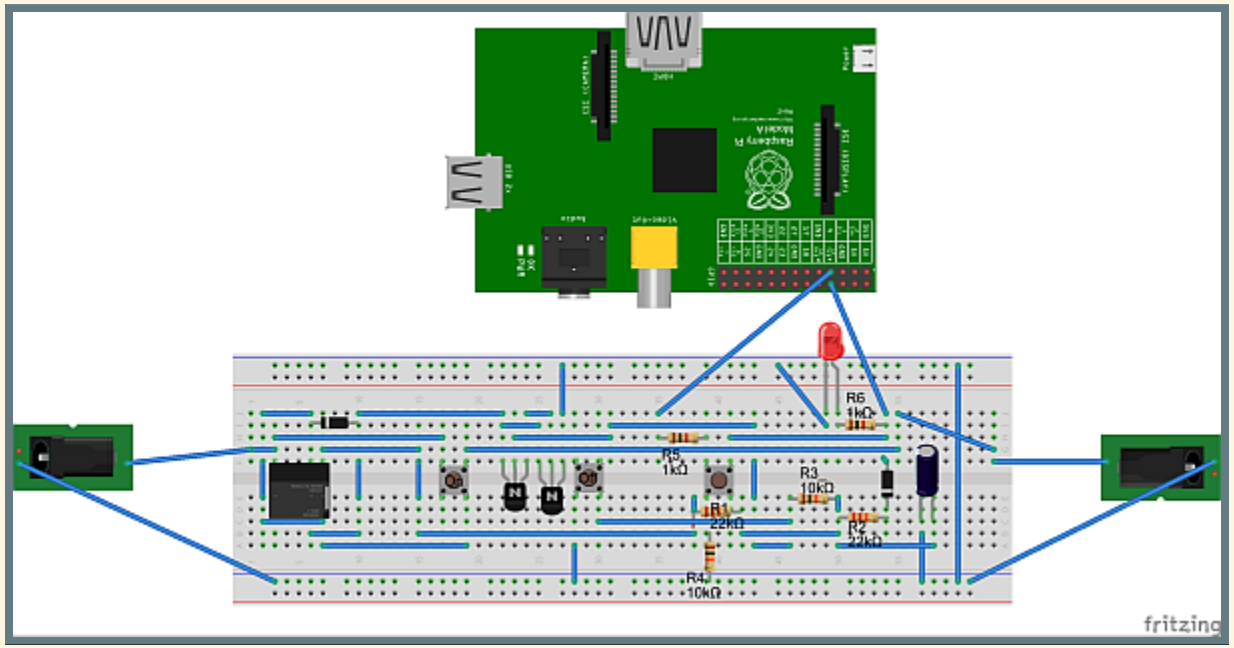


POWERSUPPLY

- prevent brown-out
- one switch for all boards

ON-OFF-SWITCH

- hardware
- code



raspberry-pi-geek.com On-Off-Switch

SCALE OUT

PYTHON LOGIC...

```
b = mraa.Gpio(20)
b.dir(mraa.DIR_IN)
b.isr(mraa.EDGE_FALLING, handleInterrupt, handleInterrupt)
```

```
def handleInterrupt(args):
    #print("handleInterrupt")
    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):
          #print("past debounce")
          journal.send('GPIO20 pressed DEBOUNCED.', FIELD2='GPIO20')
          button20.pressed_debounced = True
          return
      else:
          #print("not debounced yet")
          journal.send('GPIO20 pressed, not debounced', FIELD2='GPIO20')
      time.sleep(0.5) #inside this interrupt handler only
```

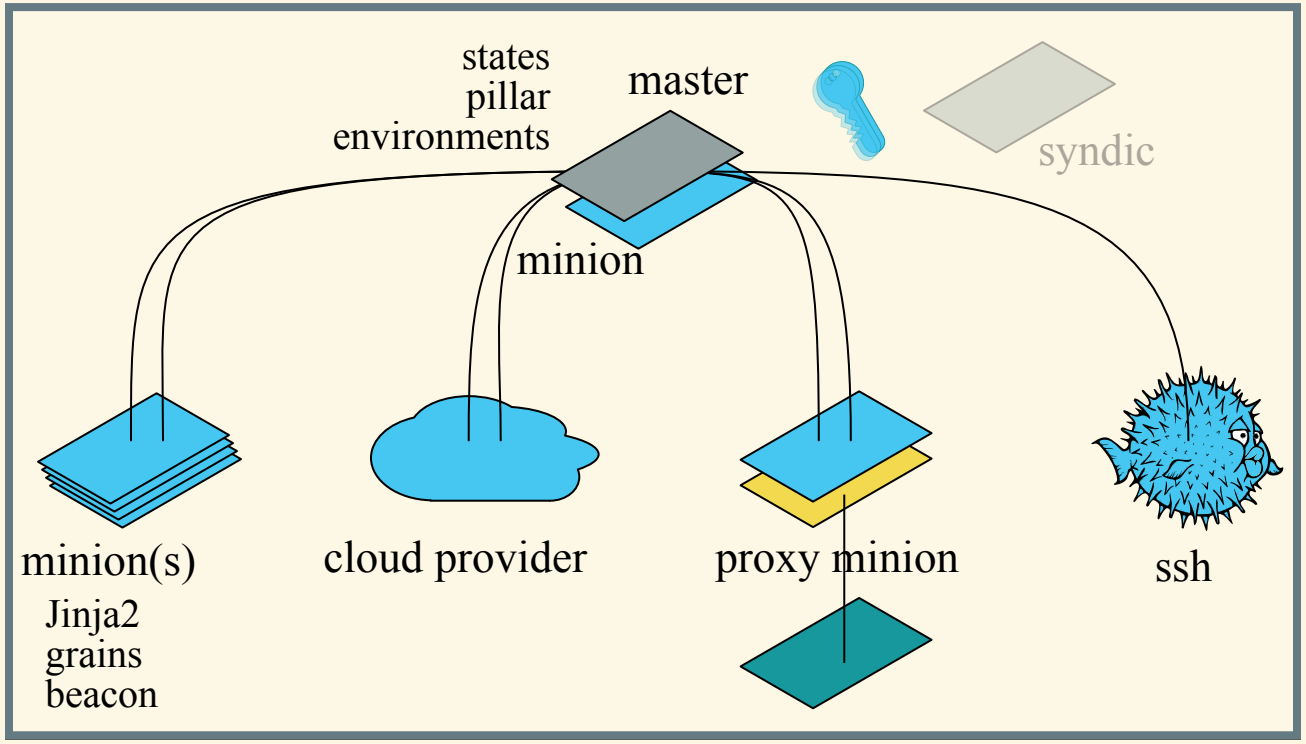
...AND SYSTEMD DAEMONIZATION.

```
[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 fileserver
- "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'
```

TODO: SCALE-DOWN AND SIMPLIFY

- three RPi 2 or RPi3
- bootstrap.sh script from ArchLinux ARM iso to "cluster"
- keep formfactor to a minimum
  - still a switch is needed
  - and a button
  - and a RTC

TODO: MORE HARDWARE DIVERSITY

- PINE64
- GBit ethernet wanted

TODO: CONFIG MGT AND IOT