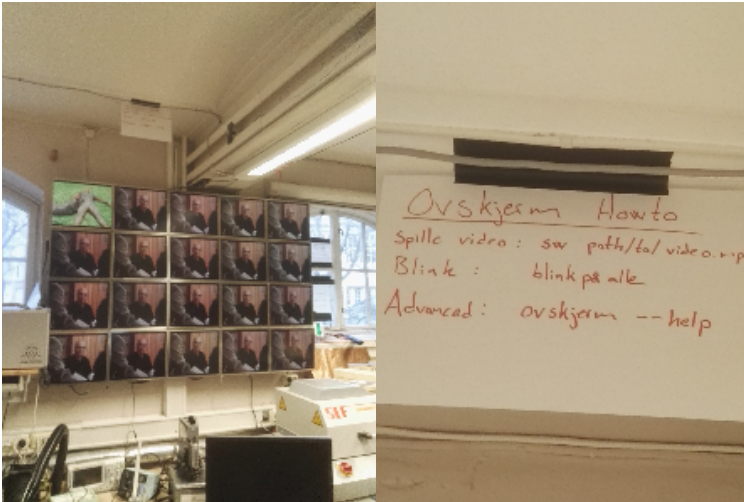


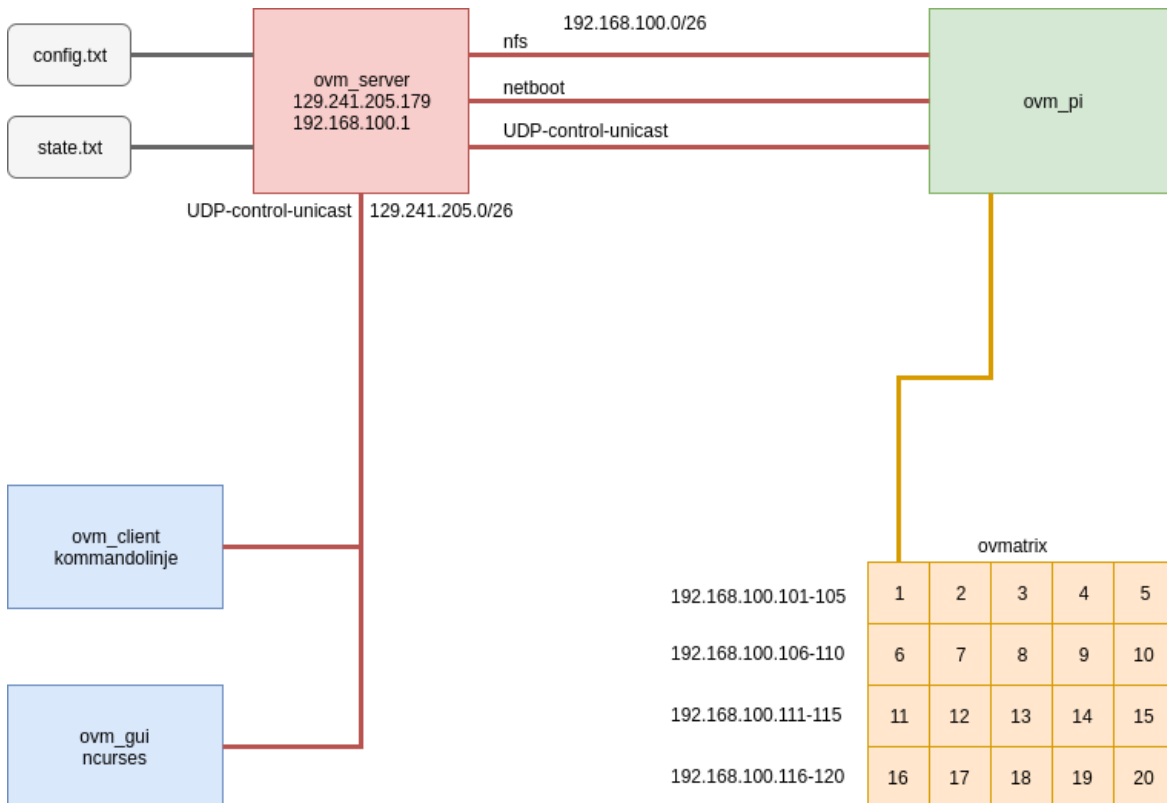
OVskjerm

Ovskjerm er skjermmatrisen på veggen ut mot hovedbygget. Den fungerer vanligvis som en "logg" for Blink, men kan også vise video, nettsider og rutetabeller for buss. Koden finnes på http://git.omegav.no/ovskjerm_scripts og er i all hovedsak skrevet i python. Scriptene krever en del eksterne biblioteker, men alt det er dokumentert på git sammen med koden. Denne wiki siden er i hovedsak for å dokumentere oppsettet av serveren og piene. På serveren ligger hele rpi installasjonen. Hver pi har en lokal versjon av uboot (en bootloader for embedded linux platformer). Når rpi starter uboot vil uboot spørre serveren om diverse filer, helt til den til slutt spør om å få et boot script som vi har lagd. RPI laster ned bootsriptet og kjører det. Bootsriptet laster ned kjernen og "devicetree" (om du ikke vet hva noen av de tingene er kan det absolutt anbefales og lese litt mer om det før du fortsetter i denne guiden). Når rpi har lastet ned de to filene så vil den kjøre bootprosessen som er beskrevet i bootsriptet. Når kjernen starter så velger vi at pi skal hente root filesystemet fra en delt nfs mappe. Når operativsystemet er lastet inn logger pi seg automatisk inn i konsolen, og da blir .bash_profile kjørt, som starter x-serveren. Når x starter med openbox kjøres et autostart script som starter ovskjerm_node_server.py.

20 Rasperry Pi 1 B+. 4 gigabit switcher. 1 x86 maskin.



Systemet følger mester-slave hierarki. Mester finnes på indigo.ed.ntnu.no og denne lager et subnett hvor slavene er koblet til. Se mer informasjon på git eller følgende chart



Software: https://git.omegav.no/ov/ovskjerm_scripts

HW: https://git.omegav.no/ov/ovskjerm_scripts

Setup

Don't follow this wiki guide slavishly. Think when you are doing stuff, because it might have changed after this guide was written and not been documented.

Setup on Server

- Serveren kjører på indigo.ed.ntnu.no på brus
- Set up machine as router with this guide (2 internet interfaces are needed) <http://blog.noviantech.com/2010/12/22/debian-router-gateway-in-15-minutes/> (network/interfaces, iptables, dnsmasq). Set up static ip addresses in /etc/dnsmasq.conf with dhcp-host=mac-addr,ip-addr. /etc/dnsmasq.conf should look something like this:

```
1 # Configuration file for dnsmasq.
2
3 interface=eth0 #The interface we want to delegate ip addresses to the PIwall
4 listen-address=127.0.0.1
5 domain=ed.ntnu.no
6 dhcp-range=192.168.100.100,192.168.100.200,12h
7
8
9 #RAD1
10 dhcp-host=b8:27:eb:f8:e9:5c,192.168.100.101
11 #dhcp-host=86:6d:1f:2b:f2:f9,192.168.100.101
12 dhcp-host=b8:27:eb:1b:b8:ce,192.168.100.102
13 dhcp-host=b8:27:eb:b0:d3:14,192.168.100.103
14 dhcp-host=b8:27:eb:72:49:a4,192.168.100.104
15 dhcp-host=b8:27:eb:24:94:fd,192.168.100.105
16
17 #RAD2
18 dhcp-host=b8:27:eb:1b:15:11,192.168.100.106
19 dhcp-host=b8:27:eb:29:94:6a,192.168.100.107
20 dhcp-host=b8:27:eb:e5:d8:25,192.168.100.108
21 dhcp-host=b8:27:eb:c4:51:1d,192.168.100.109
22 dhcp-host=b8:27:eb:66:0c:83,192.168.100.110
23
24 #RAD3
25 dhcp-host=b8:27:eb:ba:0f:2c,192.168.100.111
26 dhcp-host=b8:27:eb:d9:03:78,192.168.100.112
27 dhcp-host=b8:27:eb:b2:8e:01,192.168.100.113
28 dhcp-host=b8:27:eb:78:39:a8,192.168.100.114
29 dhcp-host=b8:27:eb:be:71:27,192.168.100.115
30
31 #RAD4
32 dhcp-host=b8:27:eb:10:08:d8,192.168.100.116
33 dhcp-host=b8:27:eb:97:97:de,192.168.100.117
34 dhcp-host=b8:27:eb:48:e5:af,192.168.100.118
35 dhcp-host=b8:27:eb:71:69:b3,192.168.100.119
36 dhcp-host=b8:27:eb:34:44:cc,192.168.100.120
37
38 # Set the NTP time server addresses to 192.168.0.4 and 10.10.0.5
39 #dhcp-option=option:ntp-server,192.168.0.4,10.10.0.5
40
41 # Set the boot filename for netboot/PXE. You will only need
42 # this is you want to boot machines over the network and you will need
43 # a TFTP server; either dnsmasq's built in TFTP server or an
44 # external one. (See below for how to enable the TFTP server.)
45 #dhcp-boot=pxelinux.0
46 #dhcp-boot=boot.scr.uimg
47 #dhcp-boot=u-boot.uimg
48 #dhcp-boot=kernel.img
49 #dhcp-boot=arch/boot.scr.uimg
50 dhcp-boot=raspbian/boot.scr.uimg
51
52 # Enable dnsmasq's built-in TFTP server
53 enable-tftp
54
55 # Set the root directory for files available via FTP.
56 tftp-root=/srv/tftp
57
58 # For debugging purposes, log each DNS query as it passes through
59 # dnsmasq.
60 log-queries
```

- Make nfs share <http://www.whaleblubber.ca/boot-raspberry-pi-nfs/>. `/etc/exports` should look something like this:

```

1 # /etc/exports: the access control list for filesystems which may be exported
2 #   to NFS clients.  See exports(5).
3 #
4 # Example for NFSv2 and NFSv3:
5 # /srv/homes          hostname1(rw, sync, no_subtree_check) hostname2(ro, sync, no_subtree_check)
6 #
7 # Example for NFSv4:
8 # /srv/nfs4           gss/krb5i(rw, sync, fsid=0, crossmnt, no_subtree_check)
9 # /srv/nfs4/homes    gss/krb5i(rw, sync, no_subtree_check)
10 #
11 /srv/piwallroot 192.168.100.0/24(rw, sync, no_root_squash, no_subtree_check, insecure)
12 /srv/ovmatrixRoot 192.168.100.0/24(rw, sync, no_root_squash, no_subtree_check)
13 /srv/rasbian 192.168.100.0/24(rw, sync, no_root_squash, no_subtree_check)
14 /srv/archLinux 192.168.100.0/24(rw, sync, no_root_squash, no_subtree_check)
15 /srv/blink 192.168.100.0/24(rw, async, no_root_squash, no_subtree_check) 129.241.205.128/25(rw, async,
no_root_squash, no_subtree_check)

```

- Write a systemd service that launches `ovskjerm_control_server.py` and keep respawning it if it dies.

Setup on RPI

- Make two partitions on the sdcard (see the guide on installing archlinux arm)
- Build uboot from source (use mainline) http://elinux.org/RPi_U-Boot#Mainline. Move the `u-boot.bin` to `kernel.img` on the sdcard's boot partition.
- Install: `sudo apt-get install openbox xorg feh omxplayer python3 python3-pip unclutter`
- Set `XAUTHORITY` to `tmp` as environmental variabel. The `xauthority` is needed because we are mounting `/` as read-only (`/etc/profile` and in `openbox` environmental). `~/xinitrc` should look like this:

```

1 export XAUTHORITY=/tmp/.Xauthority
2 exec openbox-session

```

- Make sure you use `fbset -xres 1280 -yres 1024` some seconds before starting `xorg` because `uboot` is currently a bit fucked (08-10-2016). This might not be needed with later releases of `uboot`. `~/ .bash_profile` should be something like this:

```

1 -f ~/.bashrc && . ~/.bashrc
2 if [ -z "$DISPLAY" ] && [ $(tty) == /dev/tty1 ]; then
3     fbset -xres 1280 -yres 1024
4     sleep 3
5     startx
6 fi

```

- Use `raspi-config` to enable automatic login to console. (then `~/ .bash_profile` gets sourced and `Xorg` starts)
- `Openbox` will autostart a couple of scripts by editing `~/ .config/openbox/autostart` to this:

```

1 xset s off -dpms #Disable blanking (screensaver)
2 feh --bg-scale /home/pi/blink/wallpaper.png #set wallpaper to this img
3 unclutter -idle 1 -jitter 2 -root -display :0 & #hides the mouse
4 /bin/sh /home/pi/ovskjerm_scripts/scripts/startup_ovm_pi_fsm.sh & #starts the ovmatrix script in the
background

```

Software

All kode finnes på https://git.omegav.no/ov/ovskjerm_scripts .

Robotkrig

Scripts from 2017 to generate images from google drive and spreadsheets during Robotkrig. All files are stored in /home/medlem/robotkrig on indigo.ed.ntnu.no. These scripts are executed from crontab every minute.

1. These are used during robowars to automatically update the pictures

- **** /home/medlem/robotkrig/googlesheet-to-image.py
- **** /home/medlem/robotkrig/tournament-to-image.py

googlesheet-to-image.py generates images for final result, fagdom and hinder and stores these in the nfs share /srv/blink/ which the pis can access.
tournament-to-image.py generates the tournament result and stores it as image in /srv/blink

To set up, uncomment the lines in crontab and execute /home/medlem/robotkrig/robotkrig.sh which will tell all the pis what image to show.

2018-Feb, these scripts should be moved to ovskjerm_scripts so that we have version control.