Aalto University
School of Engineering

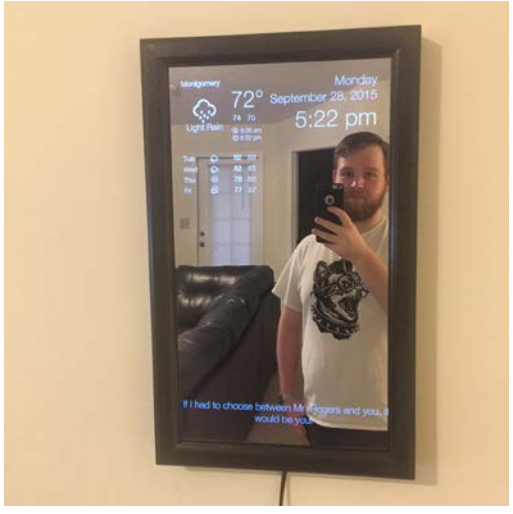
Introduction to Raspberry Pi

Ville Klar
Doctoral Candidate
Machine Design

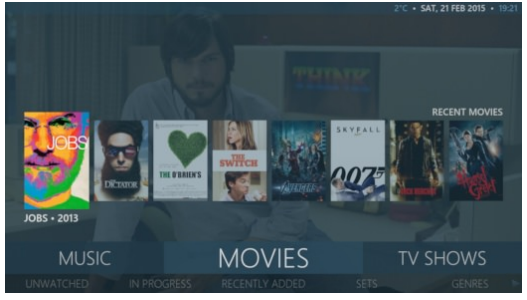
What is Raspberry Pi ?

- Arguably the most popular single board computer (SBC)
  - Easy to get started with because basically every problem is documented
- Add a computer with a OS to practically anything
  - NOTE: Do not expect it to perform as well as your laptop
- Support for a vast array of peripherals (thanks to the Linux kernel)
  - USB devices, networking, displays, cameras, audio etc.

What can you do with a Pi? (not so cool)

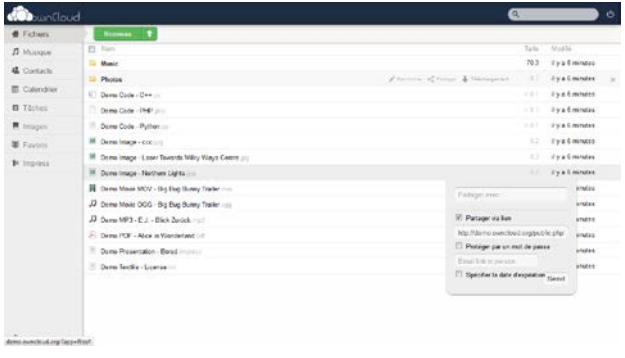


https://www.raspberrypi.org/magpi/magic-mirror/

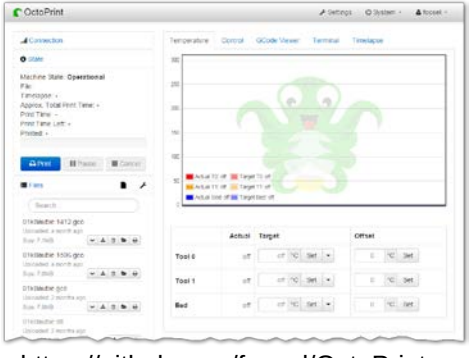


http://mymediaexperience.com/raspberry-pi-xbmc-with-raspbmc/

https://learn.adafruit.com/pigrrl-2/overview

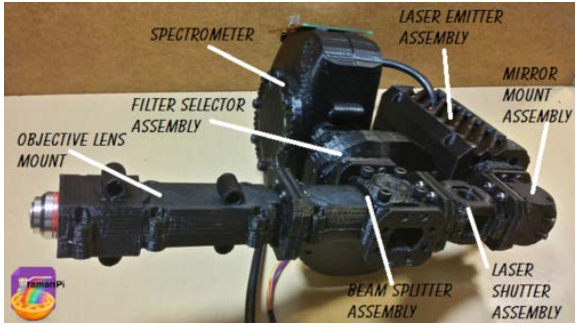


https://vadelmapii.com/blogi/yllapida-omaa-dropbox-kloonia-raspberry-pilla-kayttaen-owncloudia

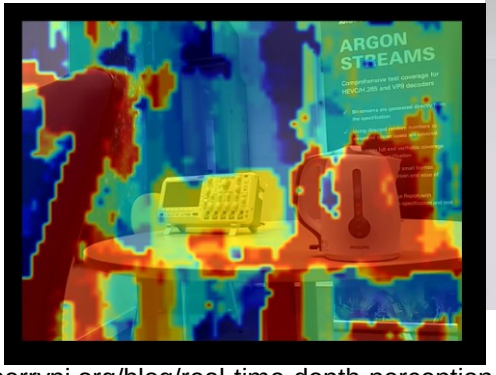


https://github.com/foosel/OctoPrint

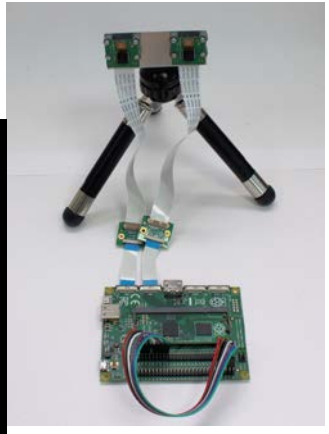
What can you do with a Pi? (cool)



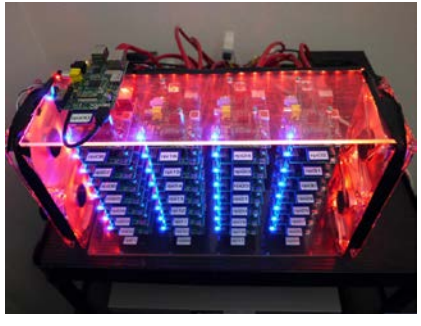
https://hackaday.io/project/1279-ramanpi-raman-spectrometer



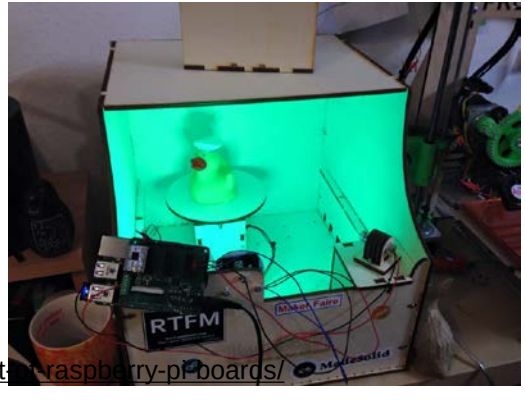
https://www.raspberrypi.org/blog/real-time-depth-perception-with-the-compute-module/



https://hackaday.io/project/1269-mashberry-beer-brewing-with-raspberry-pi

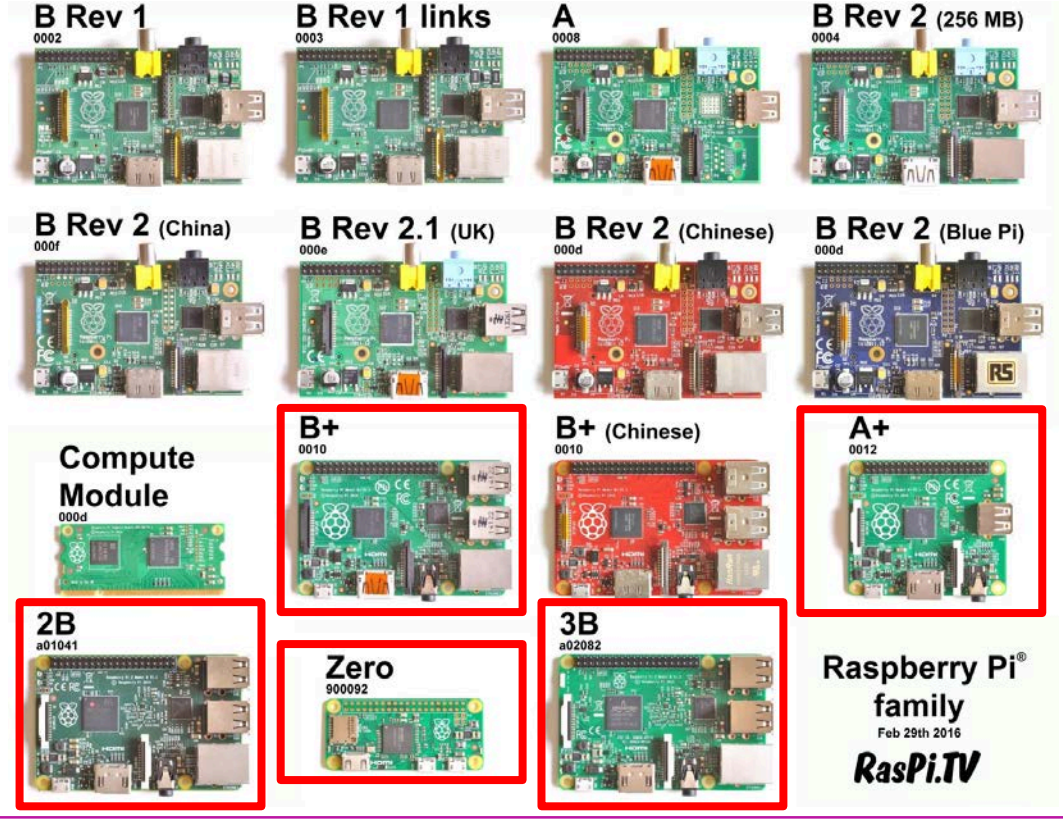


http://www.zdnet.com/article/build-your-own-supercomputer-out-of-raspberry-pi-boards/

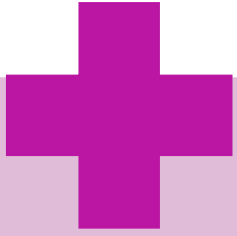


https://www.raspberrypi.org/magpi/fabscan-pi-project-3d-scanning-for-al

Raspberry Pi models



Raspberry Pi advantages and disadvantages



- Cheap (price per performance)
- Well documented
- Availability
  - Also in terms of add-ons (HATs)
- Versatile
- Compact (especially Zero)

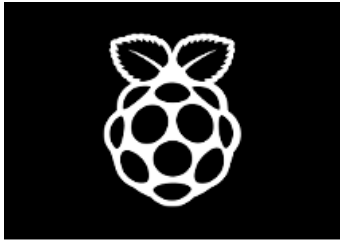
- Scary linux (learning required)
- Not real time*
- No ADC (easy to add though)
- PWM possible but limited frequency

*** It is possible to install a RTOS on the Pi**

Getting started

1. Install OS
2. Configure OS
3. Connect peripherals
4. Write software
5. Profit (?)

OS options



NOOBS

RASPBIAN

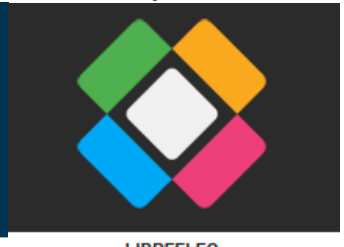
with Pixel or Lite

WINDOWS 10 IOT CORE

RISC OS

A non-Linux distribution

OSMC



LIBREELEC

UBUNTU MATE

SNAPPY UBUNTU CORE

Installation process

- **Get SD card (micro on newer Pi's)**
- **Format to FAT32 (for example with SD card formatter)**

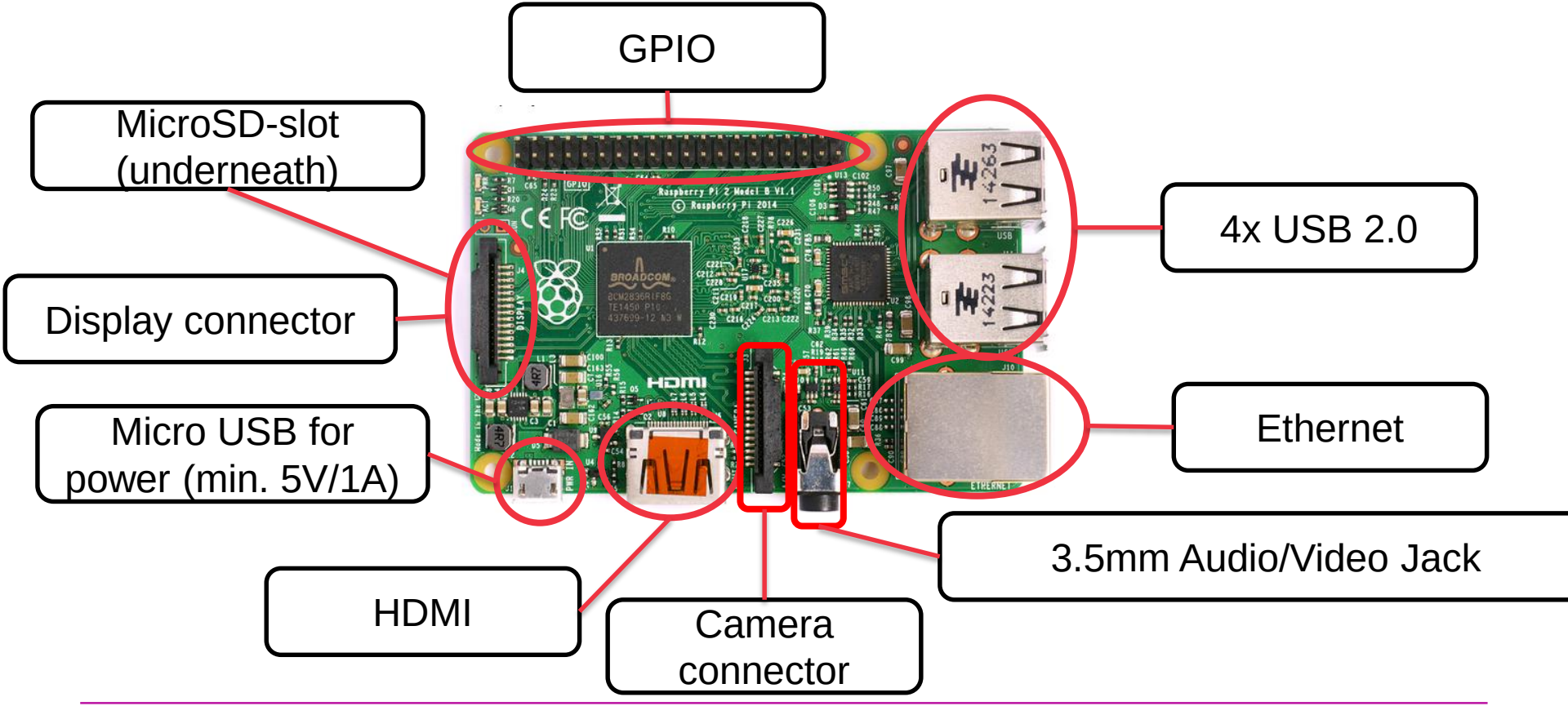
Using NOOBS:

- **Download NOOBS (sd-cards with pre-installed NOOBS can purchased)**
- **Unzip and copy all contents to SD card (takes a while)**
- **Boot**

Using a disk image writer:

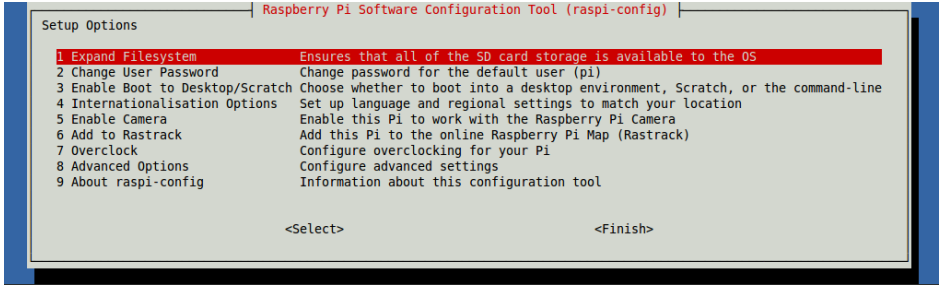
- **Download the disk image (.img)**
- **Write the disk image on the SD card ("hard drive")**
  - Windows: Win32 disk image writer or Etcher
  - MacOSX / Linux: Use dd (or image writer with GUI such as Etcher)
- **Boot**

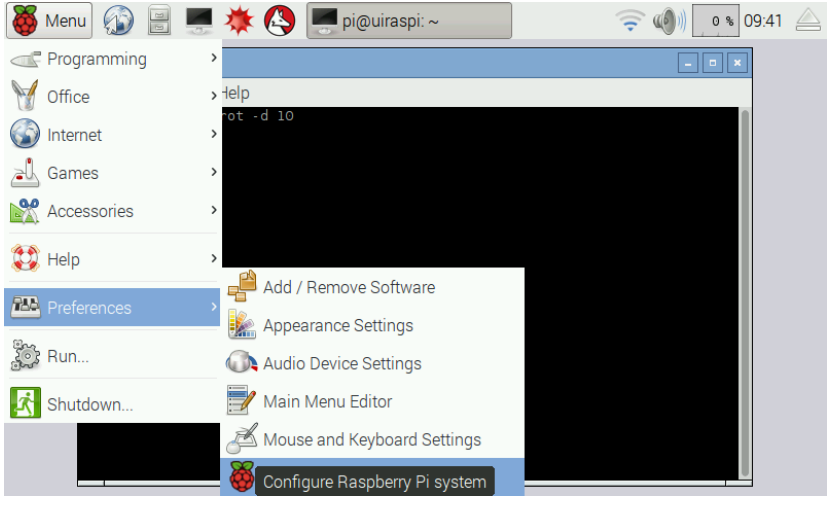
Connections



On first boot…

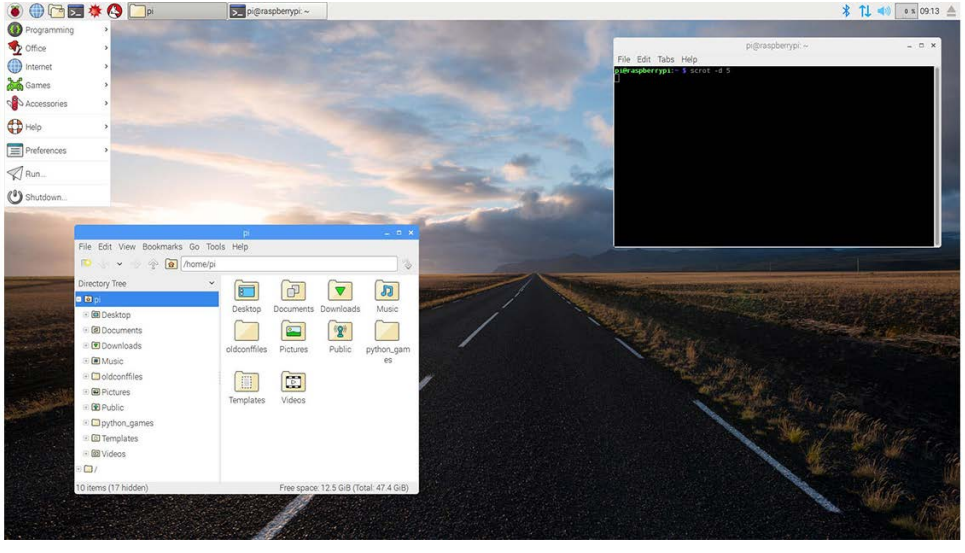
- NOOBS installer has GUI (Raspbian recommended)
- Boots into raspi-config (you can run it with “sudo raspi-config”)
- Expand file system, change password and change keyboard layout, enable ssh etc.





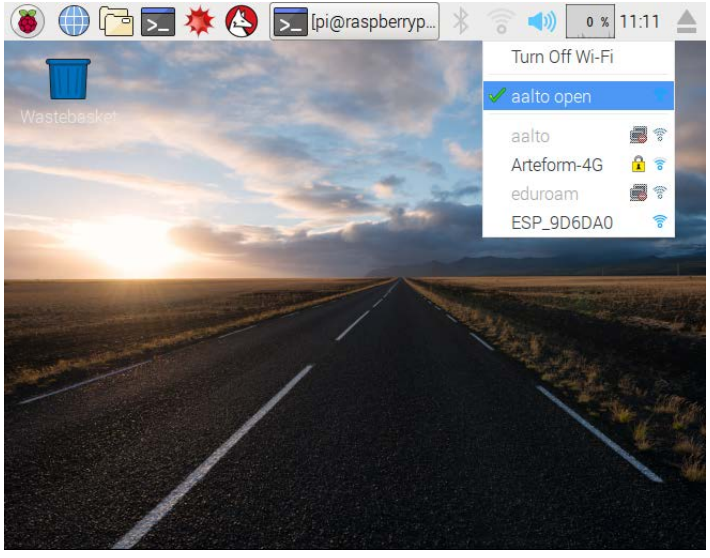
Terminal or Pixel

- **There are two options when booting: shell or Pixel desktop (graphical session)**




```
Debian GNU/Linux wheezy/sid raspberrypi tty1

raspberrypi login: pi
Password:
Last login: Tue Aug 21 21:24:50 EDT 2012 on tty1
Linux raspberrypi 3.1.9+ #168 PREEMPT Sat Jul 14 18:56:31 BST 2012 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Type 'startx' to launch a graphical session

pi@raspberrypi ~ $
```

Connect to WLAN



In GUI:

- **Choose WiFi network and connect**

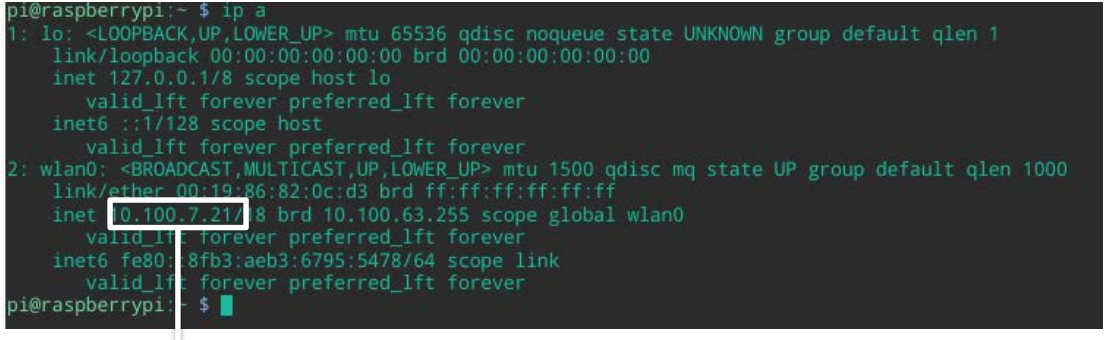
Without GUI :

- **sudo iwlist wlan0 scan** **// scan for networks**
- **sudo nano /etc/wpa_supplicant/wpa_supplicant.conf** **// open wpa_supplicant configuration**
- **Add this to the bottom of the file**

```
network={
        ssid="aalto open"
        proto=RSN
        key_mgmt=NONE
}
```

SSH

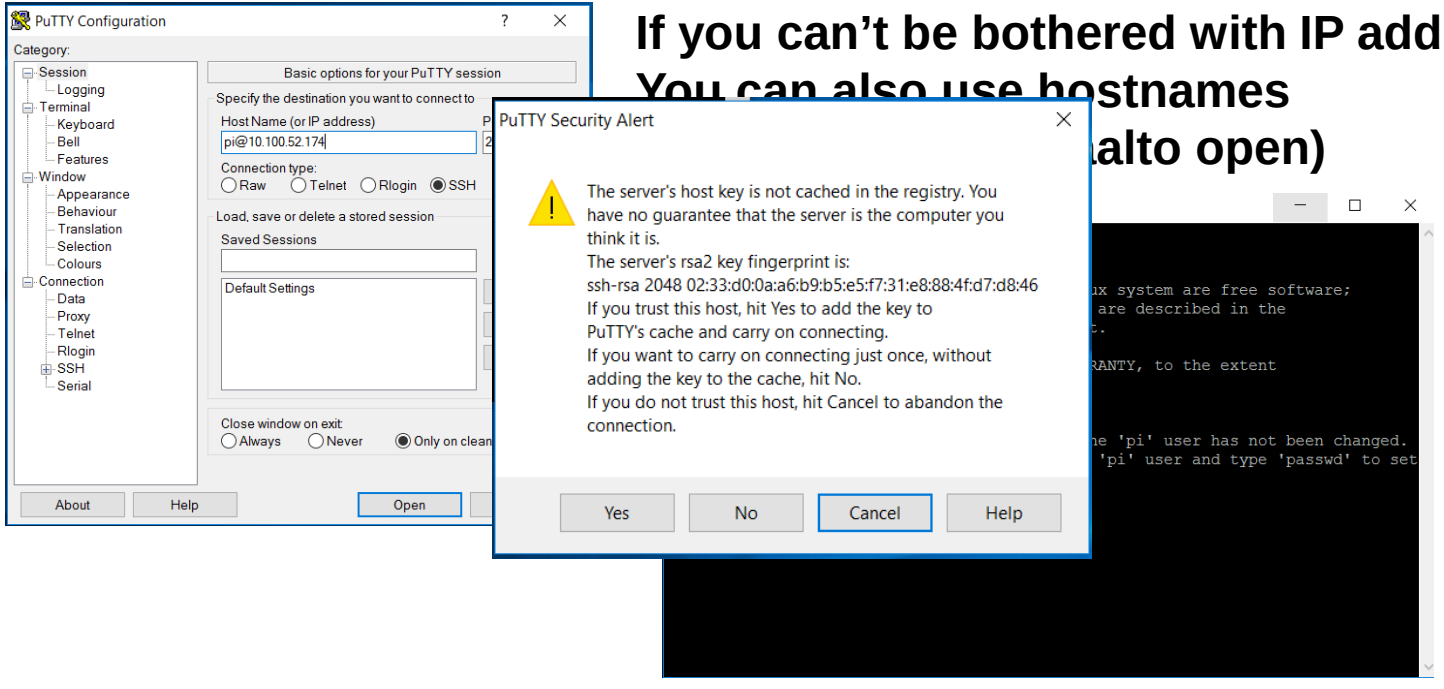
- Secure shell = Easy way to remotely access a Pi
- After connecting to a WLAN we need to find out the Pi's ip address
  - Type 'ip a' in a terminal output is something like this
- If you don't have a monitor connected you can try to nmap the ip
- If the Pi connects to the same router between boots, DHCP should give it the same ip
  - You can also configure a static IP



Local ip address

SSH

When you have obtained the ip-address you can access the Pi with e.g. Putty



If you can't be bothered with IP addresses
You can also use hostnames
(aalto open)

CLI / Shell commands

Command	Description
ls	List contents of current directory (folder)
cd	Change directory
mkdir	Creates a new directory (make directory)
sudo	Execute as super user (“admin” / root privileges)
cp	Copy (syntax is cp src dest)
mv	Move (syntax is mv src dest)
rm	Remove
./	Dot-slash means execute
nano	Launch nano text editor
apt-get update	Update repository list (update list of links to software downloads)
apt-get install	Install a package
reboot	Reboots the system (run with sudo)
poweroff	Turns the Raspberry Pi off (run with sudo)

CLI/ Shell tips

- Use tab to make the terminal autocomplete
- Up and down keys to browse command history
- Ctrl+c to kill a process, Ctrl+z to stop a process
- Ctrl+r to search command history
- Use ‘exit’ to close a terminal window
- Type ‘man’ before the command to see it’s manual
- Github repositories can be cloned (copied) with ‘git clone’

- You can also open an sftp connection to the Pi to view and edit files & folders

Programming language options

- Almost everything
  - C, C++, Python2/3, Javascript (node), Ruby, Lisp, Rust…
- Most projects are either C or Python

GPIO Libraries

- **WiringPi (C and Python)**
  - Blink example : http://wiringpi.com/examples/blink/
- **RPIO (python2/3)**
  - PWM via DMA (up to 1µs resolution)
- **Bcm2835.h (C)**
- **Pigpio (javascript / python/ C)**

Demo 1: Video streaming

- Installation instructions: https://github.com/jacksonliam/mjpg-streamer
  - Already installed in these Pi's
- Connect the Raspicam, make sure it is working with

  raspistill -o cam.jpg
- Run mjpeg-streamer with (first cd into ~/Software/mjpg-streamer/mjpg-streamer-experimental) :

  export LD_LIBRARY_PATH=.

  ./mjpg_streamer -o "output_http.so -w ./www" -i "input_raspicam.so"

- View the stream at http://localhost:8080 or with another computer (on the same network) at http://<IP-address-of-Pi>:8080
- You can also use usb web cameras with mjpeg-streamer
- There are many other options for streaming: clvc, gstreamer,
- Use ctrl+c to kill the stream

Demo 2: WiringPi

- **Open terminal**
- **cd into ~/Software/WiringPi_Blink**
- **Connect LED to gpio 29 and short lead to ground**
  - We should put a current limiting LED but we can break the rules
- **Run blink with sudo ./blink**
- **Try changing the pin to 28 (in the code) and run make to compile**

Raspberry Pi GPIO Header

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

Demo 3: Servo

- Open terminal
- cd into ~/Software/Python_Servo
- Connect servo to gnd, 5V and GPIO 4
- Run sudo pigpiod to start the gpio daemon
- Run the python script with
  python servo_demo.py
- You can give another pin as an argument to the script e.g. python servo_demo.py 18
- Try changing the speed of the servo

Raspberry Pi GPIO Header

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

Tips

- For (prototype) web servers: Python Flask
- For GUIs: Tkinter (Python) or Qt (C++) or Node.js+Express.js
- If you need e.g. read analog voltages and control motors → consider combining an Arduino with a Raspberry Pi

(Raspberry Pi commands Arduino board via serial (USB))

Links

- Help videos (getting started)
- Embedded Linux Wiki
- Raspberry Pi Forums
- Hackaday.io Raspberry Pi projects
- Adafruit learning guides
- Raspberry Pi subreddit