Raspberry Pi setup

Configuration for the working environment based on the official Raspbian image.

Basic configuration

$ sudo raspi-config

Expand filesystem
Enable camera
Change timezone
Configure locales

To remove the language errors add the following lines to /etc/default/locale

LANG=en_GB.UTF-8
LC_ALL=en_GB.UTF-8
LANGUAGE=en_GB.UTF-8

Then reboot the system

$ sudo reboot

Update the system

$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo reboot

Update the firmware

$ sudo rpi-update
$ sudo reboot

Public key authentication (optional)

On a mac transfer the public key to the RPi

$ cat ~/.ssh/id_rsa.pub | ssh pi@192.168.0.28 "mkdir .ssh;cat >> .ssh/authorized_keys"

Camera driver installation (for C++)

$ wget http://www.linux-projects.org/listing/uv4l_repo/lrkey.asc && sudo apt-key add ./lrkey.asc

Add the following line to the file /etc/apt/sources.list

$ deb http://www.linux-projects.org/listing/uv4l_repo/raspbian/ wheezy main
$ sudo apt-get update
$ sudo apt-get install uv4l uv4l-raspicam
$ sudo reboot

For driver options type

$ uv4l --help --driver raspicam --driver-help

To load the driver

$ uv4l --driver raspicam --auto-video_nr --width 640 --height 480 --nopreview --encoding jpeg

Or

```
$ uv4l --driver raspicam --auto-video_nr --width 640 --height 480 --nopreview --encoding yuv420
```

To load the driver at boot

```
$ sudo apt-get install uv4l-raspicam-extras
```

To kill the driver

```
$ pkill uv4l
```

OpenCV installation

This installs the required packages for C/C++ and Python OpenCV programming, if only C/C++ is required then “libopencv-dev” is enough.

```
$ sudo apt-get install libopencv-dev python-opencv
```

Depending on the use the uv4l driver may need to be loaded and the LD_PRELOAD may need to be defined, but this maybe not, try without these 2 lines first:

```
$ uv4l --driver raspicam --auto-video_nr --width 640 --height 480 --nopreview --encoding jpeg
$ export LD_PRELOAD=/usr/lib/uv4l/uv4lext/armv6l/libuv4lext.so
```

Camera API in Python

This package uses MMAL underneath and is extremely fast in acquiring the preview image, it also doesn’t need the driver and LD_PRELOAD to be defined and is really easy to use and configure.

```
$ sudo apt-get install python-picamera
```

http://picamera.readthedocs.org/en/release-1.3/install2.html#raspbian-installation

Camera API in C++

This works, really fast, but only in grayscale.

See https://github.com/Josh-Larson/CameraBoardAPI

Copy the files “/build/src/libraspicam.so*” into /usr/local/lib/.

Other resources:

- https://github.com/tasanakorn/rpi-mmal-demo
- https://github.com/hcpa/rpi-mmalcam

Access desktop with VNC

```
$ sudo apt-get update
$ sudo apt-get install tightvncserver
```

To start the VNC server type

```
$ vncserver
```
Enter an 8 character password such as “12345678”
Select “n” for view-only password

To connect from a Mac computer open Finder, go to Go→Connect to Server… (⌘K) and type vnc://<IP of the RPi>:5901

I2C communication

Install: http://skpang.co.uk/blog/archives/575

```
$ sudo apt-get update
$ sudo apt-get install i2c-tools
```

It’s normal to get the *“/run/udev or .udevdb or .udev presence implies active udev. Aborting MAKEDEV invocation.”* warning, it doesn’t mean there was an error in the install.

Add “pi” to the group “i2c” to get the permissions to use the i2c-tools:

```
$ sudo adduser pi i2c
$ sudo reboot
```

To detect if something is connected to the i2c lines type:

```
$ sudo i2cdetect -y 1
```

Examples: http://www.robot-electronics.co.uk/htm/raspberry_pi_examples.htm

Make sure not to go over 3.3V on the RPi GPIO pins, if the Arduino Wire library is used it automatically activates the pull-up resistors to its VCC which can be at 5V.

Serial connection in C++

To connect to a USB device via the serial connection there is no need to install anything, the USB device (an Arduino for example) will be visible in /dev/ttyACM0, one should simply connect to it via a serial console or through a C++ program.

To test the connection you can use the screen command:

```
$ sudo apt-get update
$ sudo apt-get install screen
$ screen /dev/ttyACM0 9600
```

Type Alt+a, Alt+k to exit screen.

Serial connection in Python

http://pyserial.sourceforge.net/

```
$ sudo apt-get install python-serial
```

Raspberry Pi LED control

One can control the on-board ACT LED for debugging:
http://raspberrypi.stackexchange.com/questions/697/how-do-i-control-the-system-leds-using-my-software

```
$ sudo su
$ echo none >/sys/class/leds/led0/trigger
$ echo 1 >/sys/class/leds/led0/brightness
$ echo 0 >/sys/class/leds/led0/brightness
$ exit
```

The LED is connected to GPIO16 which can be used to toggle the LED when not used for SD card access indication.

Reset to default ACT function which makes it come on when the SD card is accessed:

```
$ echo mmc0 >/sys/class/leds/led0/trigger
```

Mount the Raspberry Pi filesystem in read-only

To prevent corruption the filesystem can be mounted in read-only mode so that a power failure won't corrupt the system if it was currently writing something on the SD card.

In /etc/fstab:

```
/dev/mmcblk0p2  /           ext4    defaults,noatime,ro  0       1
tmpfs           /tmp        tmpfs   defaults,size=30M    0       0
```

Disable services you don't need (not tested):

```
insserv -r dphys-swapfile
insserv -r rsyslog
```

http://raspberrypi.stackexchange.com/questions/5112/running-on-read-only-sd-card

WiFi configuration from command line

When a WiFi dongle is attached and must be configured via serial, such as Bluetooth, these are the commands to issue: http://harindern.blogspot.ch/2013/02/raspberry-wifi.html

Other good tutorial: http://www.maketecheasier.com/setup-wifi-on-raspberry-pi/

Make sure all lines perfectly match, sometimes some line breaks are skipped when opening a file for some reason, they must be put back before saving.

Ethernet connection directly with the Raspberry Pi

To connect 2 computers via Ethernet cable they have to be assigned IP addresses.

First conigure the Rapsberry Pi with a static address the the following command:

```
$ sudo ifconfig eth0 192.168.0.10
```

Then configure the comptuer (Linux) with

```
$ sudo ifconfig eth0 192.168.0.1
```

The computer will be able to communicate with the Raspberry Pi via ssh now.

Launch python script at startup

Add these lines before “exit 0” in /ets/rc.local

```
# Automatically launch the robot script
python /home/pi/robot/python/control.py &
```

References

- Another C++ API for the camera
  http://www.uco.es/investiga/grupos/ava/node/40
- Acquiring full sensor video with a Raspberry Pi camera module using v4l2
  http://www.robertcudmore.org/blog/?p=200