**Raspberry pi Notes**

**Ehsan Shafiei**

**2016, 2017**

# Starting it up and running

Running using laptop:

* Make OS up
* Download the image file of Raspbian Lite
* Write the image file to the SD card using Win32DisImager
* Set up IP address:
* In the SD card folder find cmdline.text
* At the end of the file write ip=169.254.0.2 (for direct access one can write ip=196.168.0.2)
* Connect keyboard and monitor to the raspi and enter

sudo raspi-config

and enable the SSH connection in the advanced config.

* Internet Protocol Version 4 Properties may set to a static ip address.
* Setup a terminal emulator :

Using PuTTy

* Install Xming
* Download and run putty.exe
* In Session category enter the IP address 169.254.0.2 in the Host Name field
* In Connection category under SSH->X11 make sure that Enable X11 forwarding is checked
* Save the session with an appropriate name (RaspberryPiAuto)

Using SmarTTY

* Run SmarTTY.exe
* Host name: 192.168.137.2
* Username: pi
* Password: raspberry
* Run raspi
* Insert the micro SD into it
* Connect the LAN cable to the PC
* Start the system
* Configure raspi
* Run putty
* Load RaspberryPiAuto session and open it
* At start time the consul ask for the log in info. Log in as “pi” with password “raspberry”
* Write *sudo raspi-config*
* Apply your preferred configurtions

# Simulink code generation

Set compiler when running mex:

mex -v GCC='arm-linux-gnueabihf-gcc' cfile.c

For Simulink build use the following options:

* System target file: ert\_linux.tlc
* Make command: make\_rtw CC=arm-linux-gnueabihf-gcc
* Template makefile: ert\_linux.tmf

# Run Binary/Executable File

To run the generated binary/executable file on RPi, take the following steps:

1. Upload the file to /home/pi
2. chmod +x FileName
3. sudo ./FileName

# UDP communication

*Matlab/Simulink setup*

For UDP send from RPi, use the “UDP Send” block from “Embedded coder -> Embedded Targets -> Operating Systems -> Embedded Linux”.

The problem is it would generate code using a shared library “libmwnetworkdevice.so” which is not available in the RPi Linux.

The following steps should be taken to be able to compile for the target:

1. In *Hotlib\_Network.c* (C:\Matlab2014a\MATLAB\R2014a\toolbox\shared\dspblks\extern\src) add #define \_USE\_TARGET\_UDP\_
2. Copy the content of *linuxUDP.c* from

C:\Matlab2014a\MATLAB\R2014a\toolbox\target\extensions\operatingsystem\linux\src\linuxUDP.c

and add it to the bottom of *Hostlib\_rtw.c* existed in

(C:\Matlab2014a\MATLAB\R2014a\toolbox\shared\dspblks\extern\src)

*IP address*

In the command prompt enter the command

ipconig

to get information about the **local IP addresses** in the machine. In this case it was 192.168.0.11.

# I2C

I2C stuff

## Repeated Start I2C

To enable repeated start I2C in rpi:

$ sudo su –

$ echo -n 1 > /sys/module/i2c\_bcm2708/parameters/combined  
$ exit

To ensure that the combined remains set after reboot, create a file using nano (e.g. i2c.conf) in ‘/etc/modprobe.d/’ and write the following line into it:

options i2c-bcm2708 combined=1

To check that, reboot the rpi and write:

$ sudo cat /sys/module/i2c\_bcm2708/parameters/combined

Then it should return ‘Y’.

# WiringPi

To install WiringPi library follow the instruction in [www.wiringpi.com](http://www.wiringpi.com)

To cross-compile in eclipse, copy

libwiringPi.so

from rpi folder

usr/local/lib/

and paste it to the wiringpi folder (e.g. C:\workspace\RaspberryPi\wiringPi-b0a60c3\wiringPi).

Then add the folder path to the “Includes” and “Library Paths” in C/C++ General => Path and Symbols.

# OpenCV

<http://www.pyimagesearch.com/2016/04/18/install-guide-raspberry-pi-3-raspbian-jessie-opencv-3/>

<http://docs.opencv.org/trunk/d7/d9f/tutorial_linux_install.html>

* sudo raspi-config

Once prompted, you should select the first option, “1. Expand File System”, hit Enter on your keyboard, arrow down to the “<Finish>” button, and then reboot your Pi:

* sudo reboot
* sudo apt-get update
* sudo apt-get upgrade
* sudo apt-get install build-essential cmake pkg-config
* sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
* sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
* sudo apt-get install libxvidcore-dev libx264-dev
* sudo apt-get install libgtk2.0-dev
* sudo apt-get install libatlas-base-dev gfortran
* mkdir wd
* cd wd

copy opencv and opencv\_contrib directories here.

* cd opencv
* mkdir build
* cd build
* cmake -D CMAKE\_BUILD\_TYPE=RELEASE \

-D CMAKE\_INSTALL\_PREFIX=/usr/local \

-D OPENCV\_EXTRA\_MODULES\_PATH=~/wd/opencv\_contrib/modules \

-D BUILD\_EXAMPLES=ON ..

* make –j4
* sudo make install
* sudo ldconfig

Check for installation

pkg-config --modversion opencv