DFRobot project

# Raspberry Pi B+ configuration

Backing up the SD card image of the Raspberry Pi:

To create an image of the Pi SD card the dd command can be used. Put the SD card in the Mac and identify the 8 GB SD card disk with ‘diskutil list’. The Identifier will be something like ‘disk1’. Then in the dd command one can use /dev/disk1 for the device or for higher speed the (unbuffered) raw disk /dev/rdisk1 which is ok when using the dd command.

sudo dd if=/dev/rdisk1 of=/Users/fhict/ReneB/GitHub/rbakx/DFRobot\_SDcardBackup\_TooLargeForGithub/SDcardBackup20150124.img bs=1m

or to save space:

sudo dd if=/dev/rdisk1 bs=1m | gzip > /Users/fhict/ReneB/GitHub/rbakx/DFRobot\_SDcardBackup\_TooLargeForGithub/SDcardBackup20150124.gz

This will create an image you can use to create a new SD card. This image can be restored on an SD card using again the dd command:  
sudo dd bs=1m if=/Users/fhict/ReneB/GitHub/rbakx/ DFRobot\_SDcardBackup\_TooLargeForGithub/SDcardBackup20150124.img of=/dev/rdisk1

Connecting the Raspberry Pi to the Macbook via the ethernet cable:We want to connect the Pi to the Macbook through the Ethernet cable so we can control the Pi with the keyboard / screen of the Macbook and also provide internet access to the Pi. The latter is done by enabling internet sharing on the Macbook.

* First configure the Ethernet network settings to static IP 192.168.2.1. It must be this address because this is the fixed IP address which the OS X Mavericks uses when enabling internet sharing. The address 192.168.2.1 address means thet the Pi must have a static address in the same subnet, e.g. 192.168.2.2. This is accomplished by setting this in **/etc/network/interfaces**. Also the gateway of the Pi has to be set there and this must be the IP of the Macbook: IP 192.168.2.1.
* Internet sharing from Macbook:  
  When enabling internet sharing on the Macbook the Macbook apparently always uses 192.168.2.1. This is on a different subnet then the Wifi, so the Macbook acts a a router.
* A reboot might be needed before internet sharing is really working. Still the setup seems not very reliable.

## Network configuration in /etc/network/interfaces:

auto lo

iface lo inet loopback

iface eth0 inet static

address 192.168.2.2

netmask 255.255.255.0

gateway 192.168.2.1

allow-hotplug wlan0

iface wlan0 inet dhcp

wpa-roam /etc/wpa\_supplicant/wpa\_supplicant.conf

iface default inet dhcp

To get a static wireless IP address the wlan0 part can be changed to:

allow-hotplug wlan0

iface wlan0 inet manual

wpa-roam /etc/wpa\_supplicant/wpa\_supplicant.conf

iface default inet static

address 192.168.1.42

netmask 255.255.255.0

gateway 192.168.1.254

## Wifi configuration in /etc/wpa\_supplicant/wpa\_supplicant.conf:

ctrl\_interface=DIR=/var/run/wpa\_supplicant GROUP=netdev

update\_config=1

network={

ssid="wifiwifiwifi"

psk="xxx"

# Protocol type can be: RSN (for WP2) and WPA (for WPA1)

proto=RSN

# Key management type can be: WPA-PSK or WPA-EAP (Pre-Shared or Enterprise)

key\_mgmt=WPA-PSK

# Pairwise can be CCMP or TKIP (for WPA2 or WPA1)

pairwise=CCMP

#Authorization option should be OPEN for both WPA1/WPA2 (in less commonly used $

auth\_alg=OPEN

}

Testing internet speed from command line:  
wget -O /dev/null http://speedtest.wdc01.softlayer.com/downloads/test10.zip

Connecting the Raspberry Pi to the wireless network (no Ethernet cable):

* Configure the file /etc/network/interfaces and the /etc/wpa\_supplicant/wpa\_supplicant.conf like above.
* The Pi will get an IP address assigned by DHCP: 192.168.1.42.

Logging into the Raspberry Pi:

* Connect to Pi via SSH: ssh [pi@192.168.1.42](mailto:pi@192.168.1.42), password ‘raspberry’.
* Install VNC server on Pi and VNC client on Macbook.
* Start up VNC server on Pi with: ‘vncserver: 1’.
* Start up VNC client on Macbook and connect to 192.168.1.42, password ‘raspberr’.

Raspberry Pi file sharing:

At http://4dc5.com/2012/06/12/setting-up-vnc-on-raspberry-pi-for-mac-access/ it is described how to set up file sharing. Netatalk is installed which is an open source AFP (Apple Filing Protocol) file server. Then Avahi is installed which is a zero configuration service discovery protocol. The file ‘/etc/avahi/services/afpd.service’ is created with the avahi settings (a.o. TCP/IP port 548 is specified). Avahi is then started as a deamon which will start automatically after each reboot.

In /etc/netatalk/AppleVolumes.default at the end there is:

~/ "Home Directory"

which is the default folder to share. There you can add an extra path to share like:

/etc "/etc"

/lib “/lib”

Then on the Mac, in finder, press cmd-K (connect to server) and enter the Pi IP address and password. Then the Pi shared folder shows up in the shared section in the Finder sidebar.  
When writing to a Pi folder like ‘/etc’ from the Mac the permission is denied. This is because the user ‘pi’ is not the owner or member of the group of that folder (root is) and the permission is on 755. When writing to ‘/etc’ is needed, on the Pi do: ‘sudo chmod 777 /etc’.

After writing put back permission to 755 with ‘sudo chmod 755 /etc’.

Raspberry Pi camera module:

Follow the instructions at ‘https://miguelmota.com/blog/raspberry-pi-camera-board-video-streaming/’ and follow the instructions at ‘Update: 19 Jan 2014 - Easier way to stream’ or ‘http://petrkout.com/electronics/low-latency-0-4-s-video-streaming-from-raspberry-pi-mjpeg-streamer-opencv/’ to install the MJPG Streamer.

Start the camera and MJPG stream:

$ LD\_LIBRARY\_PATH=/opt/mjpg-streamer/ /opt/mjpg-streamer/mjpg\_streamer -i "input\_raspicam.so -fps 15 -q 50 –ex sports -x 640 -y 480" -o "output\_http.so -p 9000 -w /opt/mjpg-streamer/www" > /dev/null 2>&1 &

Then the stream can be viewed at 192.168.1.42:9000.

The extra ‘> /dev/null 2>&1’ (redirect stderr to stdout) at the end is to make sure the mjpg\_streamer command returns with a prompt. Otherwise it seems to hang (despite the ‘run as background task symbol ‘&’) and when called from CGI Python the Python script hangs.

Enable .py scripts with CGI in Apache 2:

Open ‘/etc/apache2/sites-enabled/000-default’ and add  
’AddHandler cgi-script .py’ to the section

<Directory "/usr/lib/cgi-bin">  
...  
</Directory>

Then reload Apache’s configuration using ‘sudo service apache2 reload’.

When running Python scripts from Apache2 through CGI the user is ‘www-data’. So scripts and external program run must have the proper group permissions.  
When using the camera module the VideoCore device ‘/dev/vchiq’ is used. The default permissions are 660. To be able to run from CGI Python: ‘sudo chmod 666 /dev/vchiq’.