

Raspberry Pi B+ Setup & Configuration

Environment

Setup Platform:

Windows 8.1

Raspberry Pi:

Board:

Wireless Adapter: Edimax 150Mbps Wireless 802.11b/g/n nano USB Adapter, Model EW-7811Un

Software: Raspbian OS

Caveats

Without a properly configured & booting sd-card, no video will be put out at all!

Install OS

First Boot & Initial Configuration

Wireless Network Configuration

1. First, let's make sure your wireless hardware is correctly working

Perform a wireless survey via commandline:

```
iwlist wlan0 scan
```

If the scan fails, try it a few more times.

In my experience, I've had to scan more than once to get a

good survey.

2. Edit the interfaces config file

```
sudo vi /etc/network/interfaces
#Change as follows:
allow-hotplug wlan0 #Applies to plug-n-play hardware such as
USB, specify the interface name (e.g. wlan0)
iface wlan0 inet dhcp #Specify dhcp for the interface
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf #Applies to
Wi-Fi Protected Access client wpa_supplicant/
pre-up wpa_supplicant -B w -D wext -i wlan0 -
c/etc/wpa_supplicant/wpa_supplicant.conf #Specify preflight
command for wpa initialization
post-down killall -q wpa_supplicant #Specify post command for
wpa stop
```

Important: Notice the lack of a space after the -c switch in the wpa_supplicant command. That is purposeful!

3. Next, we are going to be using the wpa_passphrase command to generate the encrypted passphrase that will be used in the wpa_supplicant.conf file referenced in the interface config

```
#Call the wpa_passphrase command with the Wireless SSID and the
Password Phrase (passphrase)
#usage: wpa_passphrase <ssid> [passphrase]
#If passphrase is left out, it will be read from stdin
#Change directory to the relevant path
cd /etc/wpa_supplicant
#issue the wpa_passphrase command to generate the encrypted
psk passphrase
wpa_passphrase {ESSID} {pass}
#where {ESSID} and {pass} are your Wireless SSID and Passwd,
respectively
```

4. Copy the encrypted key. You'll need this for the next step

5. Modify the wpa supplicant configuration file according to your wireless network configuration

Here is my file:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="MyNetworkName"
    proto=RSN
    key_mgmt=WPA-PSK
    pairwise=CCMP TKIP
    group=CCMP TKIP
    psk=134b516a855384ea15429da72033ec2725934ed3ff5e3bc2fff64b8ad7
    8c2dfc
}
```

Important: The above psk passphrase was randomly generated! You must change it to match your settings. Also, do not enclose the value in double quotes.

6. Restart your wireless network client

```
sudo wpa_action wlan0 stop
sudo ifup wlan0
#--or--#
sudo wpa_action wlan0 stop && sudo ifup wlan0
```

7. Verify your wireless connectivity

```
iwconfig
#-- or --#
sudo wpa_cli status
#-- or --#
ifconfig
#Optionally, you can query your dhcp lease records file
sudo cat /var/lib/dhcp/dhclient.leases
```

If problems, see the Troubleshooting section

[divider]

Appendix

[divider]

Command Cheatsheet

Handy Commands

Command	<pre> Description iwconfig wlan0 essid NETWORK_ID key WIRELESS_KEY </pre>	<pre> Call the iwconfig command to generate the wireless configuration sudo wpa_action wlan0 stop </pre>	<pre> Stop the wireless network interface sudo ifup wlan0 </pre>	<pre> Start the wireless network interface sudo wpa_action wlan0 stop && sudo ifup wlan0 </pre>	<pre> Previous two commands, second command runs if first is successful sudo find /somepath - type f - name '*' - executable </pre>	<pre> Search for executables under the specified folder </pre>
---------	---	--	--	---	---	--

Customizations

1. Enable filesystem check (fsck) after reboot

Because we're working with flash media (SD Cards, etc), I wanted a way to force the Raspberry Pi to check its filesystem after every reboot.

The Raspbian OS will perform a filesystem check if a special file semaphore is present in the root (/) path

```
/forcefsck
```

You can run a one-time fsck on the system simply by creating this file with the touch command, as in:

```
sudo touch /forcefsck
```

To ensure this happens after **every** reboot, you can add this command sequence to the Raspbian OS system startup file:

```
/etc/rc.local
```

```
sudo vi /etc/rc.local
```

Add the below line:

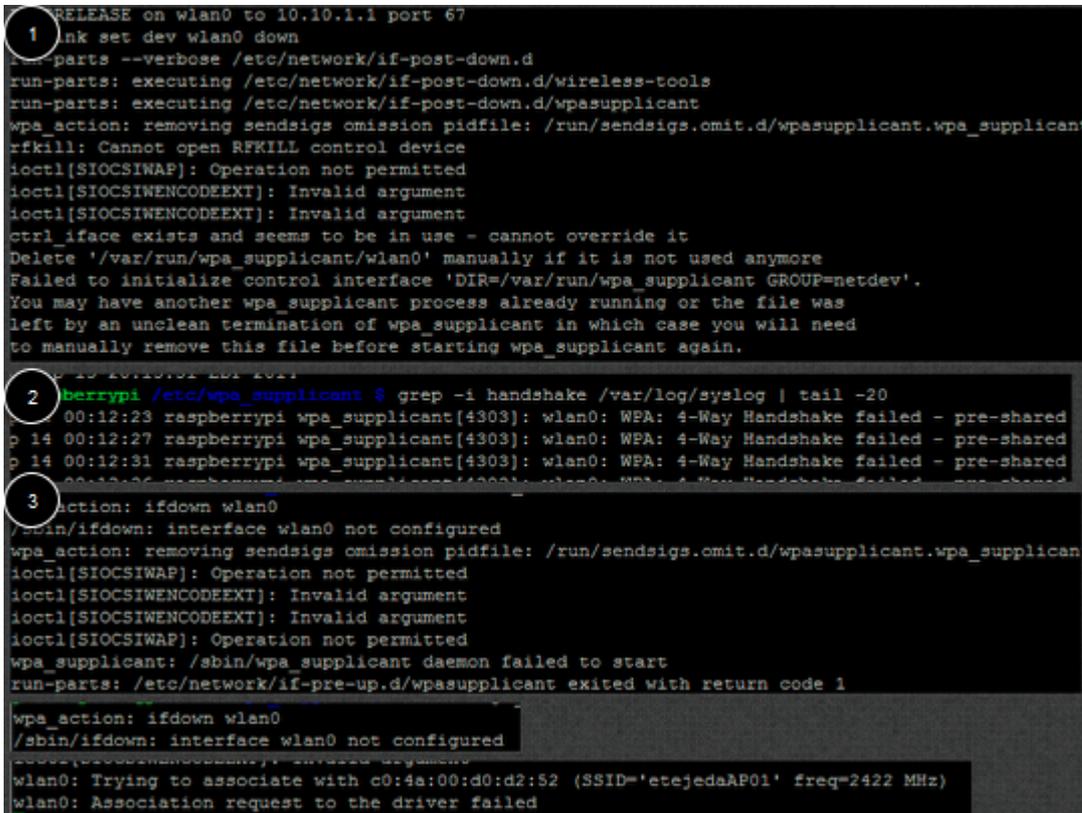
```
/usr/bin/touch /forcefsck
```

Soldering Inputs & Interfaces

The purpose of the P6 header is to enable a reset of the

Raspberry Pi's CPU. The reset switch can restart the system, but it does not provide the safe shutdown

Troubleshooting



```
RELEASE on wlan0 to 10.10.1.1 port 67
1 link set dev wlan0 down
run-parts --verbose /etc/network/if-post-down.d
run-parts: executing /etc/network/if-post-down.d/wireless-tools
run-parts: executing /etc/network/if-post-down.d/wpa_supplicant
wpa_action: removing sendsigs omission pidfile: /run/sendsigs.omit.d/wpa_supplicant.wpa_supplicant
rfkill: Cannot open RFKILL control device
ioctl[SIOCSINAP]: Operation not permitted
ioctl[SIOCSINENCODEEXT]: Invalid argument
ioctl[SIOCSINENCODEEXT]: Invalid argument
ctrl_iface exists and seems to be in use - cannot override it
Delete '/var/run/wpa_supplicant/wlan0' manually if it is not used anymore
Failed to initialize control interface 'DIR=/var/run/wpa_supplicant GROUP=netdev'.
You may have another wpa_supplicant process already running or the file was
left by an unclean termination of wpa_supplicant in which case you will need
to manually remove this file before starting wpa_supplicant again.

2 berrypi /etc/wpa_supplicant $ grep -i handshake /var/log/syslog | tail -20
00:12:23 raspberrypi wpa_supplicant[4303]: wlan0: WPA: 4-Way Handshake failed - pre-shared
0 14 00:12:27 raspberrypi wpa_supplicant[4303]: wlan0: WPA: 4-Way Handshake failed - pre-shared
0 14 00:12:31 raspberrypi wpa_supplicant[4303]: wlan0: WPA: 4-Way Handshake failed - pre-shared
00:12:36 raspberrypi wpa_supplicant[4303]: wlan0: WPA: 4-Way Handshake failed - pre-shared

3 action: ifdown wlan0
/sbin/ifdown: interface wlan0 not configured
wpa_action: removing sendsigs omission pidfile: /run/sendsigs.omit.d/wpa_supplicant.wpa_supplicant
ioctl[SIOCSIWAP]: Operation not permitted
ioctl[SIOCSIWENCODEEXT]: Invalid argument
ioctl[SIOCSIWENCODEEXT]: Invalid argument
ioctl[SIOCSIWAP]: Operation not permitted
wpa_supplicant: /sbin/wpa_supplicant daemon failed to start
run-parts: /etc/network/if-pre-up.d/wpa_supplicant exited with return code 1

wpa_action: ifdown wlan0
/sbin/ifdown: interface wlan0 not configured
wlan0: Trying to associate with c0:4a:00:d0:d2:52 (SSID='etejedaAP01' freq=2422 MHz)
wlan0: Association request to the driver failed
```

1. Search the system log for any errors

```
grep -i handshake /var/log/syslog | tail -20
```

```
Sep 14 00:13:32 raspberrypi wpa_supplicant[4303]: wlan0:
WPA: 4-Way Handshake failed - pre-shared key may be incorrect
Sep 14 00:13:36 raspberrypi wpa_supplicant[4303]: wlan0: WPA:
4-Way Handshake failed - pre-shared key may be incorrect
Sep 14 00:13:40 raspberrypi wpa_supplicant[4303]: wlan0: WPA:
4-Way Handshake failed - pre-shared key may be incorrect
Sep 14 00:13:45 raspberrypi wpa_supplicant[4303]: wlan0: WPA:
4-Way Handshake failed - pre-shared key may be incorrect
Correct the passphrase
```

```
sudo rm -f /var/run/wpa_supplicant/wlan0
```

```
sudo rm -f /var/run/wpa_supplicant/wlan0
```

4. Try to manually run the connection command:

```
sudo /sbin/wpa_supplicant -P
/var/run/wpa_supplicant.wlan0.pid -i wlan0 -D nl80211,wext -
```


Term	Notes dupont wires	http://lmgty.com/?q=dupont
wires		

Difference between the SET and SELECT statements when assigning variables in T-SQL

see: {http://vyaskn.tripod.com/differences_between_set_and_select.htm}

see: {<http://stackoverflow.com/questions/3945361/t-sql-set-versus-select-when-assigning-variables>}

Quick Summary of Differences

1. SET is the ANSI standard for variable assignment, SELECT is not.
2. SET can only assign one variable at a time, SELECT can make multiple assignments at once.
3. If assigning from a query, SET can only assign a scalar value. If the query returns multiple values/rows then SET will raise an error. SELECT will assign one of the values to the variable and hide the fact that multiple values were returned (so you'd likely never know why something was going wrong elsewhere – have fun troubleshooting that one)
4. When assigning from a query if there is no value returned then SET will assign NULL, where SELECT will not make the assignment at all (so the variable will not be changed from its previous value)

5. As far as speed differences – there are no direct differences between SET and SELECT. However SELECT's ability to make multiple assignments in one shot does give it a slight speed advantage over SET.