RPi2/3 version 1.02 Install Notes

The version 1.02 RPi2/3 release comes up natively the same as the 1.01 RPi2 release with the exception that it has a new kernel and runs equally well on both the RPi2 and RPi3.

To enable the alpha-test version in the 1.02 code and the additional commands listed below, see the "safe_asterisk" command in the file /usr/local/etc/rc.allstar

...Comment out: (put a # at the beginning of the line)
"/usr/bin/safe_asterisk" and
uncomment the line: (remove # at the beginning of the line)
"/usr/local/hamvoip-alpha-test/sbin/safe_asterisk" Then reboot your system.

The new code -should- be stable, but remember this is beta code at this point!

If you need to switch back to the original stable Asterisk release code, it's as simple as reversing the above change and rebooting.

This is just a portion of the code changes which will be in our 2.0 release. If you decide to try this please report back any issues you may fine and if you have any questions don't hesitate to ask.

WIFI on the Raspberry Pi 3. The Pi3 has built-in WIFI. There is no need to add any external device. Tests show it works quite well. The WIFI code is NOT part of the 1.02 release. Here is the information on downloading and installing the WIFI package.

WIFI on the Pi3

A hamvoip package for utilizing wireless on the RPi3 is now available. This package **ONLY** works on the RPi3 and with the current RPi2-3 1.02beta download on the hamvoip.org website. So far tests show that wireless on the Pi3 works very well and has good range. If you would like to use wireless on the RPi3 download this package to your RPi3.

NOTE - You must be using DHCP when using wireless! If not rerun the firsttime.sh script and set the network question to DHCP. DHCP is generally recommended when using both wired and wireless connections. Most all routers have the ability to set an assigned DHCP address as permanent. Once set as permanent you can then do the port forwards to that address. Wire and wireless have different mac addresses so they will be assigned different IP addresses on your router.

Web download -

https://hamvoip.org/downloads/hamvoip-wireless-config-0.1-7.pkg.tar.xz

Or on your Pi3 go to the the root directory

```
cd /root
```

then

```
wget https://hamvoip.org/downloads/hamvoip-wireless-config-0.1-7.pkg.tar.xz
```

Then install the package with the following command

```
pacman -U --force hamvoip-wireless-config-0.1-7.pkg.tar.xz
```

Also an important addition to eliminate CRDA messages and hangups at boot please install the following packages.

Enter at the linux prompt -

```
pacman -Sy crda iw wireless-regdb
```

This will install these three packages. Again this only needs to be installed in the V1.02 RPi2/3 beta version.

What is CRDA?

CRDA domain is relevant for choosing the permitted wireless channels for a given country. Not all channels are permitted in all countries. For Europe has channel 12 and 13, which are not permitted in North America.

Although it does not seem to matter in the US, you should uncomment your country designator in /etc/conf.d/wireless-regdom

```
#WIRELESS_REGDOM="US"    - remove the # in this line for US
```

when done, execute:

```
wireless-setup.sh
```

Select your wireless access point and give the passphrase. It will then give you further instructions and ask you to disconnect the wired connection. When the system is rebooted it will be using the wireless connection.

Note that the wireless connection will be a new local IP address. You should hear this from your radio if you had everything setup in the wired mode previously. You could also get this IP address from your router or by reading the CW from the LED on the board.

Another option is an inexpensive HDMI monitor and wireless keyboard attached to the Pi3. This would allow changing the connections, checking status, and doing everything you would normally do with a console connection to Linux. Here is what I use, it is great for demos also.

HDMI monitor -

https://www.amazon.com/gp/product/B012ZRYDYY/ref=oh_aui_detailpage_o01_s00?ie=UTF8&psc=1

Keyboard -

https://www.amazon.com/gp/product/B00I5SW8MC/ref=oh_aui_detailpage_o01_s00?ie=UTF8&psc=1

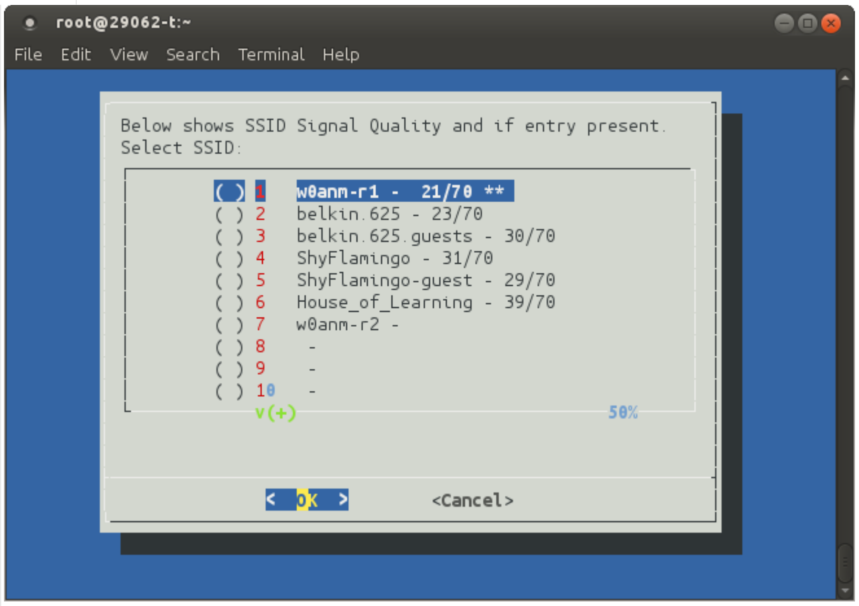
Note when using the monitor you may want to change the framebuffer size in /boot/config.txt

```
framebuffer_width=720 #1280 former value
framebuffer_height=480 #720  former value
```

While port forwarding is not necessary especially in mobile nodes if you don't have it enabled you will only be able to connect outbound. To setup port forwarding first make the assigned wireless IP address permanent in your router. Every router does this differently but most all have the capability based on the Pi boards MAC address to come up on the same IP every time it is booted. Once you have that setup if you desire port forwarding you can set that up pointing to the permanently assigned wireless IP address.

There is a "wireless-control.sh" script to enable and disable the wifi interface as well as showing status.

Example screen for selecting wireless access point -



```
Below shows SSID Signal Quality and if entry present.
Select SSID:

        ( ) 1  w0anm-r1 -  21/70 **
        ( ) 2  belkin.625 - 23/70
        ( ) 3  belkin.625.guests - 30/70
        ( ) 4  ShyFlamingo - 31/70
        ( ) 5  ShyFlamingo-guest - 29/70
        ( ) 6  House_of_Learning - 39/70
        ( ) 7  w0anm-r2 -
        ( ) 8   -
        ( ) 9   -
        ( ) 10  -
            v(+)                          50%

            <  Ok  >        <Cancel>
```

Wireless access point definitions are stored in:

/etc/wpa_supplicant/wpa_supplicant_custom-wlan0.conf

The entries you make with wireless-setup.sh are cumulative. When booting up in wireless mode each entry in order will be tried for connection until a good one is found. This can cause confusion when a local wifi that is always on is first on the list and would always be selected. If you wanted to connect to a phone hotspot for testing and you have the home wifi first in the list it will never connect to the phone. You have two options use a script to delete all entries or manually edit the /etc/wpa_supplicant/wpa_supplicant_custom-wlan0.conf file. If you are not experienced in editing files it would be safer to delete the file and reenter the wifi devices in the order you desire. To clear the file run the following script -

clear_wpa_passwd_file.sh

Then proceed to do the wireless setup again in the order you desire.

You can look at the status of a connection including its IP address by selecting the status option in the wireless-control.sh script.

Thanks to Chris, W0ANM for getting this going on the Pi 3. This code will be included as part of the upcoming V2.0 release.

Tests so far show that the Pi3 WIFI works extremely well.

New Changes in the alpha test code

New app_rpt.c changes:

- Removed obsolete uchameleon code (about 2500 lines).

This is obsolete hardware.

app_rpt.conf

These added commands are on a per node basis and can be added in each node assignment.
Note that in this test mode these commands will have to be added to the appropriate config file.

- Added COS timeout, with COS flap suppression

rx_timeout <time_in _ms>   Default 300000 – 5 minutes

rx_cosflap <time_in_ms> Default 20 - 20 ms

This is an attempt at eliminating single node hangups of connected networks. Often a single node, due to radio or interface failure, can cause a COS hangup tying up an entire network. If the timeout is set COS is monitored and the indication of asserted COS to the network is dropped if the time is exceeded. The COS timeout is set in the same way as a TX timeout. The COS unasserted indication to the network will remain until the hardware COS is cleared.

COSFLAP can also be set and will attempt to catch conditions where a squelch is "flapping" by not resetting the timer during this condition and allowing the timeout period to proceed.
Default rx_timeout is 5 minutes. This should be set to just longer than the maximum expected transmit time into your node. This is typically less than 3 minutes as this is a common repeater timeout value.

- Added telemetry gain levels
t**elemnomdb <level_in_dB> Default 0 – no gain change**

This will allow a level adjustment of telemetry (connect, disconnect, etc.) messages. The level change is in dB like the Echolink level adjustments. This will eliminate the need to change the level of audio files.

- Added telemetry ducking (AKA: Allison ducking)

telemduckdb <level_in_dB> Default -6dB

This optionally reduces telemetry levels when a voice signal is present.
Note these are independent levels not additive. Typical settings might be -9 for telemnomdb and -12 for telemduckdb

- Changed tx timeout variable name from totime to tx_timeout (the old variable is deprecated, but still works after printing a console warning)

tx_timeout <time_in_ms>

This was done to avoid the confusion between TX and RX timeouts. TX timeout is applied only to the local transmitter. It is often in addition to a TX timeout that is set at a longer time in the radio itself ensuring double protection for a hung transmitter.

New core Asterisk changes:

- Removed need to recompile to enable/disable RADIO_RELAX

This command when on reduces the DTMF decode requirements but can also cause DTMF "Falsing" RADIO _RELAX is OFF by default and has been in ALL BBB and RPi2 compiles. This command previously required a menuselect change and recompile to enable. It is now added as a runtime option:

dtmf_relax=[0|1]

in the simpleusb.conf and usbradio.conf files, for a given interface stanza.

DO NOT set this to on unless you absolutely need it. Setting your levels and deemphasis correctly almost always solves the problem of not decoding DTMF. Setting this to on can also cause audio holes due to voice falsing. The default is off if you don't define it.

New chan_simpleusb.c changes:

- Added code to resolve annoying variable transmitted audio delay

This has plagued simpleusb on all platforms since it was added to the ACID code. It is more prevalent on the small boards and especially the RPi2. The symptoms generally only appear when Allstar is used on a repeater. A user would hear their own voice for a few syllables or a word after unkeying. This update totally fixes the problem and the delay through the system is now constant and short in length.

Other updates

- Cleaned up all compiler warnings. in app_rpt.c and chan_simpleusb both now compile cleanly, no warnings at all. Added flags to allow compiles using new gcc for new and old versions of the code

<u>Additional information on the above commands.</u>

Some of this may be redundant to the above but was in a later message and included here for your information.

Note that audio level changing is an individual thing and also depends on the levels already set in the played files. As a for instance there is a level change in the on the hour time routine -

/usr/local/sbin/saytime.pl If you are changing the wholesale levels with telemnomdb and telemduckdb you will want to eliminate the additional reduction in the file.

```
# Following lines process the output file with sox to lower the volume
# negative numbers lower than -1 reduce the volume - see Sox man page
# Other processing could be done if necessary
#
#@proglist = ("nice -19 sox --temp /tmp " . $outdir . "/temp.gsm " . $outdir . "/current-time.gsm vol
-0.35");
#system(@proglist);
```

To do that simply comment out the additional two lines as shown above.

I find that telemnomdb= -9 and telemduckdb= -12 works for me but of course it is a user preference.

Although I have left the annoying connect/disconnect messages turned off, and you may have also, you may find that turning them back on is now acceptable when using the telem levels.

The rx_timeout is a good feature in that it ensures a COS hangup won't tie up a network forever or at least until someone disconnects. The default is 5 minutes but you may want to lower that to something over the maximum time you would talk - maybe 3.5 minutes or rx_timeout=210000

When switching to the beta version of 1.02 you will need to add these commands to your config files to utilize them. Without adding them the default levels apply. When you add or change these values in rpt.conf a simple 'rpt reload' in the Asterisk client will immediately change the setting.

- simpleusb audio delay problem

The audio delay problem prevalent on the RPi2 and to some extent on the BBB has been eliminated with this update. This was always more apparent when the server was used as a repeater controller in full duplex mode but should also give more consistent results in simplex mode.

- Telemetry level control

A long standing issue with the Acid release has been the inability to control telemetry verses user voice levels in Allstar. Many users modified audio files to accomplish this but this was always a crude way to do it. Two new commands have been added to control telemetry audio levels. One controls the overall level and the other the level when a signal is present. Telemetry levels are the voice commands, typically Alison, that you hear when you connect, disconnect, etc. CW and courtesy tone levels are controlled separately within their commands. The telemetry commands are installed in the node stanza of rpt.conf and can be defined in each node of a multi-node system.

Also note that if you have altered the amplitude of any sound files they should be returned to stock level and let these commands change the levels.

- telemnomdb - default 0dB, no change

This command controls the overall level of telemetry in allstar in dB. Typically a user would want to reduce the level somewhat. To reduce the level specify a minus value. Start with

telemnomdb=-6 and increase from there to the desired level.

- telemduckdb - default -9dB reduction

This command controls how much the telemetry audio levels are reduced when a voice signal is present. This is an absolute setting and not relative to telemnomdb. So if telemnomdb is set to -6 and telemduckdb is set to -9 and telemetry was playing and a signal was present the telemetry would be reduced by -9 db and resume back to a -6 reduction with no signal present.

- COS (rx) timeout control

A nagging problem with Allstar is the ability for one node to entirely lockup a multi-node connected system. This often happens when a radio is turned off or disconnected. The COS signal will usually assume a high level when a radio is turned off or disconnected as there is a pullup in the cm108 chip. If you are using carrierfrom=usb this can happen but it would be a good idea to set this timeout in all cases to ensure that you node will release COS after a timeout period. These commands are on a per node basis in the node assignment section of rpt.conf.

- rx_timeout - default 300000 (5 minutes)

This command controls the length of time the hardware COS line can remain active before software COS is released. The timeout is reset when the hardware COS line goes inactive. When the timeout happens it appears to all Allstar connected nodes that you stopped transmitting. Obviously this timeout would also happen if a valid transmission went beyond the timeout. Typically you would set this to some value longer than you would expect to talk in a single transmission. Recommendations are something more than 3 minutes.

- rx_cosflap - default 20 (20 milliseconds)

This command works in conjunction with rx_timeout. Its purpose is to help eliminate repetitive squelch openings from resetting rx_timeout. This would typically happen if a squelch was on the edge of opening and 'flapped' open and closed causing a channel disruption. Set appropriately this would allow the timeout to happen in the rx_timeout period.

- Transmitter timeout
The transmitter timeout command 'totimeout' in rpt.conf has been changed to tx_timeout to make its name more meaningful and to be a companion to rx_timeout. The 'totimeout' command is deprecated but still works. Users should change to the new command syntax. A warning message upon loading the rpt module is given in the asterisk client if you are using the old command. The 1.3 versions have the new command installed so users loading the 1.3 version should not need to make this change.