

The image to be loaded to the raspberry Pi is provided as a .7z compressed file
uncompress it and load to relevant sd card. There are many articles on the net about how to
do this

<http://imgtfy.com/?q=load+raspberry+img+to+card>

The uncompressed image is 1.1 GB (Nov 15) with the standard 2 partition structure for
raspbian linux.

So it will fit on most SD cards, e.g. 2, 4, or 8Gb.

There is no particular need to expand the filesystem. Running as a straight igate it has
logrotation and other maintenance scripts with 150+ MB free space available.

Note:

I usually initially boot with just the sound card and Ethernet cable
plugged in and run through the first part of the setup.
Power up the pi and it should appear via dhcp on local net shortly,
check your router or other methods to find it's IP address.

ssh with the usual **pi** and **raspberry**. For windows users putty is probably the best option for
ssh access.

After login the build starts using reading the *\$HOME/.profile* to run some scripts, the first
pass runs the *aprx/configaprx*, then recommends reboot.

The second pass runs the *aprx/testaprx* (which is the step to listen to the radio, adjust sound
card and then commit the startup).

The configuration script will need some information for the first setup.

Note:

The script will now calculate your APRSIS Passcode based on your callsign.

Note:

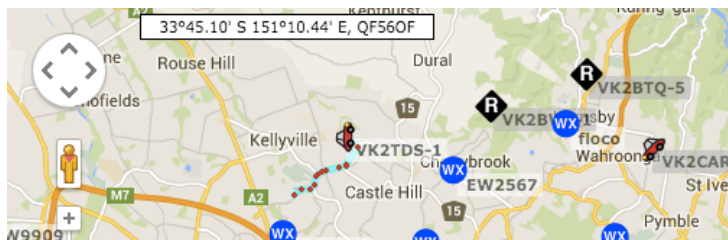
The step that most often goes wrong and causes later problem is the lat and long.
Note the format degMM.MMD This is Degrees Minutes and decimal minutes with Direction.

Location formats:

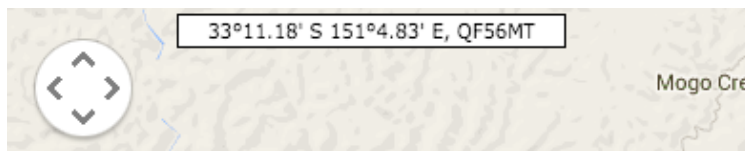
dd.dddddddd (Degrees and decimals of degree)

dd.MM.ss (Degrees, Minutes and Seconds)

dd.MM.MMMM (Degrees, Minutes and Decimals of Minutes)



The format provided on aprs.is in the top right hand corner
This example would convert to 3345.10S and 15110.44E Particularly note the placement of the decimal point. Also you may need a leading 0 in the MM.



This location is 3311.18S 15104.83E

```
If the values are correct enter Y [n]
Valid Callsign [e.g. VK2ABC] [VK2RF]VK2ABC
Terminal number [e.g. 1] [1]
Passcode is unique for your Callsign
Passcode for [VK2ABC] calculated as [21931]
APRSIS Passcode number [21931]
lat and long for your igate location
Format is degMM.MMS use the aprs.is google map to get the value
lat value [e.g. 3351.12S] [3351.43S]
lon value [e.g. 15151.12E] [15112.91E]
Params are Callsign VK2ABC Terminal 1 creating VK2ABC-1
APRSIS passcode 21931
lat is 3351.43S lon is 15112.91E
If the values are correct enter Y [n]
```

Typical answers:

Your personal/club call e.g. VK2ABC

Terminal Number is a unique identifier for this device in the APRS-IS network. Common answers for igate "3"

it shows the values then

" If the values are correct enter "Y"

it is looking for "Y" otherwise it will just loop defaulting to the values you last typed for correction.

Basically the bit I need to expand on... is the steps in testaprx

echo "a) start soundmodem/ax25 (no aprx)"

this starts the soundmodem process to listen to the radio without running the aprx which decodes and sends to the aprx network servers

echo "b) stop soundmodem/ax25 (no aprx)"

this stops the soundmodem process to listen to the radio without running the aprx which decodes and sends to the aprx network servers

echo "c) start alsamixer"

alsamixer is used to adjust the sound levels.... in particular we need to switch to the input group and adjust the MIC to about 50%

echo "d) tail syslog"

echo "e) axlisten (for signal or test track)"

axlisten will "decode" whatever soundmodem gives it, when I was testing I actually kept a looped wav file from the net that I played direct from notebook to soundcard.

echo "f) mheard"

mheard will display a log of all decoded packets it has received (they don't appear where you would expect!! in aprx.log or aprx-rf.log)

echo "g) tail aprx.log"

echo "h) tail aprx-rf.log"

These are useful after you have committed to see messages to from aprx servers

echo "i) Commit soundmodem/ax25/aprx to startup "

This commits the autostartup

echo "X) Clear All values and start again"

literally.... it will try and reset the scripts, setups , hostnames ... everything but the passwd so that you can start from the first menu cleanly.

echo "q) quit this menu"

for a new build I use

c) alsamixer *to adjust the MIC in to 50%*

a) start soundmodem noaprx this is not yet sending to internet

e) start axlisten

then try to generate and/or listen on radio or listen to a wav file.

then

i) Commit soundmodem....

and reboot.

I only use the other options if somethings not working.

To restart the testaprx menu

\$ aprx/testaprx

If you want to use with wifi plug in your usb wifi, and then at the command line you can use wicd-curses which will run you through an old curses based setup for wifi.