

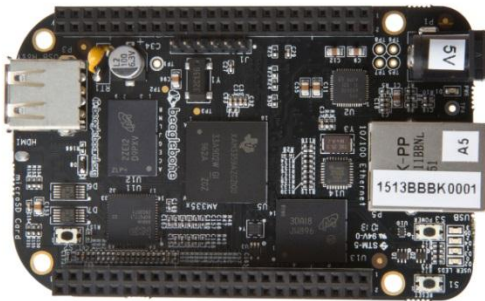


FREE STAR*

BeagleBone, BeagleBone Black and Raspberry Pi

Image Installation and Quick Start Guide for Gateway and

Personal Hotspots



This guide is for users wishing to build an embedded system or appliance, such as a BeagleBone (including the BeagleBone Black) or Raspberry Pi.

So which one: Raspberry Pi or BeagleBone Black?

The following article may help you to choose:

<http://makezine.com/magazine/how-to-choose-the-right-platform-raspberry-pi-or-beaglebone-black/>

Do not use this guide if you intend to install on a PC with an existing CENTOS installation.

The FREE STAR* team has put together three disk images that can be downloaded from here:

For the BeagleBone White:

www.ve2mrc.com/freestar/free-star-beagle-bone-white.7z

For the BeagleBone Black:

www.ve2mrc.com/freestar/free-star-beagle-bone-black.7z

For the Raspberry Pi:

www.ve2mrc.com/freestar/free-star-pi-4gb.7z

Unzip the images in prep' for burning to a SD / μ SD card (these are all 4GB images; some users have had issues getting the Pi image to fit on to a 4GB card, if you find yourself in a similar situation, please try an 8GB card). Please note that these images were created with [7zip](#).

Burning the image to your SD / μ SD card

Option 1:

Download physdiskwrite.

(<http://m0n0.ch/wall/downloads/physdiskwrite-0.5.2.zip>) - usage example here (<http://linhost.info/2010/05/write-a-img-to-a-compact-flash-with-physdiskwrite/>)

Open a DOS window and navigate to where you saved physdiskwrite.

Be sure to know exactly which device your SD / μ SD card is... use physdiskwrite.exe -a to list the devices. On my XP system, the SD card was device 4.

To burn the image:

```
physdiskwrite.exe -u free-star-pi-4gb.img
```

Option 2:

On a Windows box, you can use winimage to burn the image to your micro SD Card (in the case of a BeagleBone) or a full size SD card in the case of the Raspberry Pi (a 4GB card is required in both cases).

Download the image and unzip it, you will have a *.img file.

Greg, VA3OMP, offered the following instructions to help folks to burn the image to a µSD card with Win Image:

Win Image can be downloaded from here:

<http://www.winimage.com/download.htm>

Bare bones instructions for Winimage

1. Download the image file for FreeStar* Beaglebone. Usually with an ext of .img
2. Insert your USB card reader with your 4G microSD card into your PC.
3. Open WinImage (administrator).
4. You'll get the standard "Do you want to allow..." message click on yes.
5. You'll get a nag message to register. Just click on okay.
6. A window will appear "Click the partition to connect to" with a pull-down option with a default of "Partition on disk 0, size:#####MB, FAT".
7. Click on okay.
8. Click on the Disk tool and click on "Restore Virtual Hard Disk image on physical drive..."
9. Click on the disk with the USB 2.0 SD MMC Reader. On the type of file, change to *.*. Navigate to your .img file and click it.
10. You'll get a last chance before you write screen, click on Yes.
11. It'll take a few minutes to write and you'll be finished.

If for some reason the write fails or has errors, do a full format with the HP USB Disk Format Storage Tool and try the above process again.

Option 3:

Download Win32 Disk Imager: <http://sourceforge.net/projects/win32diskimager/>

1. Download the image file of your choice. If this file is a zip archive, unzip it to reveal the .img image file.
2. Insert your USB card reader with your 4GB microSD card into your PC.
3. Open Win32 Disk Imager
4. Using the Win32 Disk Imager browser tool, navigate to and select the .img file.
5. Select the device that you want to write to (i.e., the uSD or SD card).
6. Click on the 'Write' button
7. Once Win32 Disk Imager has finished writing to the card, you will see a confirmation pop-up; click on 'Exit' to exit from Win32 Disk Imager.

At this point you will have a μ SD card or a full-size SD card with the FREE STAR* image, ready to plug into your BeagleBone or Raspberry Pi - go ahead, plug in the card. It is a good idea to plug in your radio/repeater interface at this point (GMSK board, DVRPTR board, or the red DVAP device) before powering up for the first time. Power-up your board (in the case of the BeagleBone White, we recommend an external 5V supply so that the Bone will run at 700MHz). In the case of the Raspberry Pi, power is applied via the micro USB connector (we used a Blackberry USB charger to power the Pi in our Lab).

Note that the Raspberry Pi and USB power is a topic widely discussed on various newsgroups. Many users have found that an externally powered USB hub, with the Pi itself powered from the USB hub offers improved reliability / uptime. The issue is invariably cheap USB cables, we recommend a short cable to minimize voltage drop. An alternate method is to power the Pi via the GPIO connector (Google to find the pin-outs) – please remember that there is no protection on these pins, so you will need to install your own protection (fuse, etc.). The advantage is that you can install a 2.5mm socket and thus power the Pi from a Walwart, etc.

Communicating with the BeagleBone White from your Windows computer via USB:

Connect to a PC via USB - use Windows Control Panel > Device Manager to confirm the COM Port for the Bone.

Use PuTTY to communicate via the appropriate COM port (115200 baud, N, 8, 1)

Press <Enter> a few times to get the login prompt

Note the BeagleBone Black does not (natively) support serial communication via USB.

The Raspberry Pi image has been configured to acquire a dynamic IP address from your router / DHCP server. Since the Pi has a HDMI video output, which may present a challenge for older monitors. On our test system, we determined the IP address from the router and then used an SSH client (such as PuTTY: <http://www.chiark.greenend.org.uk/~sgtatham/putty/>) to login remotely (port 222). *Note the micro USB connector on the PI does not have the data lines connected, so you cannot communicate with the PI via the USB port.*

Login with user: root password: Fr33starAdm1n

Change the root password with the 'passwd' command. Choose a strong password – a combination of letters and numbers do not forget it!

Note there is no graphical user interface – 'startx' on the Pi will have no effect.

Network Setup (BeagleBone)

Execute the following command - confirm the address of your network interface adapter (e.g., eth0, eth1, etc...)

`ip link show`

You may find that your network adapter is something other than eth0 (e.g., eth1)

The image will have my network adapter (MAC address) assigned to eth0.... when you first boot-up, your NIC adapter may be assigned the alias eth1. If your NIC adapter is eth1 (rather than eth0), you may wish to re-assign your eth1 adapter with the alias eth0.

Edit the file `/etc/udev/rules.d/70-persistent-net.rules`

```
# This file was automatically generated by the /lib/udev/write_net_rules
# program, run by the persistent-net-generator.rules rules file.
#
# You can modify it, as long as you keep each rule on a single
# line, and change only the value of the NAME= key.
```

```
# Unknown net device (/devices/platform/cpsw.0/net/eth0) (cpsw)
#SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="00:18:31:8f:e7:4f", ATTR{dev_id}=="0x0", ATTR{type}=="1",
KERNEL=="eth*", NAME="eth0" ← comment out this line
```

```
# Unknown net device (/devices/platform/cpsw.0/net/eth0) (cpsw)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="00:18:31:e1:ae:e1", ATTR{dev_id}=="0x0", ATTR{type}=="1",
KERNEL=="eth*", NAME="eth1" ← Rename eth1 to eth0
```

Comment out the line where eth0 is specified (which has my MAC address).... then rename, your eth1 as eth0.

Save / Exit

The BeagleBone White image has been set with a static IP of 192.168.1.10 - if you need to change this per your own network IP scheme, follow the instructions below.

The BeagleBone Black should have acquired an IP address dynamically from your DHCP server (router, etc.). Some users have reported that certain routers do not show the BeagleBone Black in the list of DHCP clients. A network discovery tool such as PingLite (for Apple iOS devices) and Fing (for Android and Apple iOS devices) can be used to perform a network scan to find the IP address of the BeagleBone Black – it will appear as an ‘Armhf’ device.

Alternatively, per Mike, VA3MW, you may download a product called Advanced IP Scanner <http://www.advanced-ip-scanner.com/> and run this on your network to locate your device (and may be other devices as a bonus!).

Network Setup (Raspberry Pi)

Since the image for the Raspberry Pi was set to acquire a dynamic IP address, we recommend setting a static IP address as described below.

Setting a Static IP (BeagleBone)

in /etc/network/interfaces file:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
```

```
# The loopback network interface
auto lo
iface lo inet loopback
```

```
# The primary network interface
#allow-hotplug eth0
#iface eth0 inet dhcp
```

```
# Static IP address
auto eth0
iface eth0 inet static
    address 192.168.1.10 ← edit this block
```

```
netmask 255.255.255.0
network 192.168.1.0 ← 4th octet is 0
broadcast 192.168.1.255 ← 4th octet is 255
gateway 192.168.1.1 ← your router address
dns-nameservers 192.168.1.1 8.8.8.8 8.8.4.4 ← your router address first
```

Setting a Static IP (Raspberry Pi)

In /etc/network/interfaces file, you will find the following:

```
auto lo

iface lo inet loopback
iface eth0 inet dhcp
```

Modify this file as follows:

```
auto lo ← line 1 same as before
iface lo inet loopback ← line 2 same as before

# Static IP address ← add this block
auto eth0
iface eth0 inet static
    address 192.168.1.10 ← edit address to suit
    netmask 255.255.255.0
    network 192.168.1.0 ← 4th octet is 0
    broadcast 192.168.1.255 ← 4th octet is 255
    gateway 192.168.1.1 ← your router address
    dns-nameservers 192.168.1.1 8.8.8.8 8.8.4.4 ← your router address first
```

Save / Exit

Configuring a Wireless Card

Several people have tried to configure a wireless (WiFi) dongle with their Raspberry Pi without success. Joey, NV0N, shared the following changes to his /etc/network/interfaces file:

(Recommended WiFi dongle can be found [here](#))

```
auto lo
iface lo inet loopback

auto eth0
```

```
iface eth0 inet dhcp

auto wlan0
allow-hotplug wlan0
iface wlan0 inet static
address 10.10.10.73 ← edit address to suit
netmask 255.255.255.0
gateway 10.10.10.1 ← your router address
wpa-ssid "youressidhere" ← your router's WiFi SSID
wpa-psk "yourpasswordhere" ← your passphrase here
dns-nameservers 192.168.1.1 8.8.8.8 8.8.4.4 ← your router address first
```

Reboot your BeagleBone or Raspberry Pi with the `reboot` command.

When it comes back online, check to confirm your network adapter is live:

`ifconfig`

You should see something like this (NOTE in the example below, the IP was set to 192.168.1.4)

```
root@beaglebone:~/g2_link# ifconfig
eth0  Link encap:Ethernet HWaddr 00:18:31:8f:fd:37
      inet addr:192.168.1.4 Bcast:192.168.1.255 Mask:255.255.255.0
      inet6 addr: fe80::218:31ff:fe8f:fd37/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:50179 errors:0 dropped:1 overruns:0 frame:0
      TX packets:9996 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:4010461 (3.8 MiB) TX bytes:696614 (680.2 KiB)

lo    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING MTU:16436 Metric:1
      RX packets:111110 errors:0 dropped:0 overruns:0 frame:0
      TX packets:111110 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:6303110 (6.0 MiB) TX bytes:6303110 (6.0 MiB)
```

Now check for connectivity to the outside world:

`ping www.yahoo.com` (CTRL c to stop)

Firewall Configuration

Forward the following ports on your router / firewall to the static address that you assigned to your BeagleBone or Raspberry Pi:

TCP 222 (default port used for SSH) and port 80 for web / dashboard (optional)
UDP 20001
UDP 30001
UDP 40000

Customizing the Configuration Files

Download the install script from the VA3UV site and execute as follows:

```
cd /tmp
```

```
wget http://www.va3uv.com/hp/debian-setup-new.sh
```

```
chmod +x debian-setup-new.sh
```

```
./debian-setup-new.sh
```

The script will first update your system with any packages / updates since the image was built. This process may take a while.

Next the script will prompt you to select your location info – e.g., geographic zone and then the city so that the local time can be set.

Following customization of the environment, you will be prompted to answer a few questions, please have your ircddb password (the long one!) issued for your Gateway callsign on-hand.

NOTE – if you are setting up a personal hotspot:

Please use your personal callsign when prompted for a Gateway callsign. When prompted for your ircddb password, you can enter anything you like – e.g., 1234.

The install script will prompt you to select your radio interface – options supported are the GMSK board (Dutch Star, NI2O, Star Board, Satoshi, GMSK Micronode, etc.), the DVRPTR v1board from the German team or Bruce in Montreal, the red DVAP device or either the DVRPTR v1 or v2 boards using the XDVRPTR firmware.

We have tested a BeagleBone, BeagleBone Black and Raspberry Pi running both a Satoshi GMSK node adapter and a DVRPTR board (in a dual module setup, with both modules running full-duplex) with excellent results (using a USB hub with external power). The install script allows you to setup one RF module. Please contact us for additional information if you wish to setup a second RF module.

All config files will be updated via the install script. Note if you select the dvrptr board as your radio interface, the script will advise you to look at the dvrptr.log file (/var/log/dvrptr.log or /var/log/dvrptr2.log) to obtain the serial number of your board – example:

```
100313 at 11:36:50:... computed max number of loops 7500, each loop is 400 microseconds
100313 at 11:36:50:Trying to access device /dev/ttyACM0
100313 at 11:36:50:Success in opening device /dev/ttyACM0
100313 at 11:36:50:Device /dev/ttyACM0 now locked for exclusive use
100313 at 11:36:50:Device /dev/ttyACM0 has serial=[12.50.D0.27]
100313 at 11:36:50:Closing device /dev/ttyACM0
100313 at 11:36:50:Cant find any FREE ACMx device that matches
```

In the example above, the dvrptr board has a serial number of 12.50.D0.27 – you now need to manually edit the dvrptr.cfg file (found in /root/dvrptr/dvrptr.cfg) and enter the serial number:

Replace the default:

```
DVRPTR_SERIAL=11.50.D0.63
```

With: DVRPTR_SERIAL=your serial number

Save / exit and then restart the dvrptr service with the `service dvrptr restart` command (or `service dvrptr2 restart` if you are using the XDVRPTR firmware).

The log file should now look something like this:

```
root@raspberrypi:~/dvrptr# tail -f /var/log/dvrptr.log
100313 at 11:48:59:
----- STATUS READ FROM MODEM -----
100313 at 11:48:59:Bit1 : Transmitter enabled      = 1
100313 at 11:48:59:Bit2 : PC Watchdog enabled     = 0
100313 at 11:48:59:Bit3 : Checksum-Calculation enabled = 0
100313 at 11:48:59:Bit4 : I/O 21 Status          = 0
100313 at 11:48:59:Bit5 : I/O 23 Status          = 0
100313 at 11:48:59:Bit6 : Duplex-Capability      = 1
100313 at 11:48:59:Bit7 : (1) physical layer     = 0
100313 at 11:48:59:-----
```

DVAP Access Points

A similar process applies for the red DVAP device – the serial # will need to be retrieved from the /var/log/dvap_rptr.log file and entered into the dvap_rptr.cfg file (found in /root/dvap_rptr/dvap_rptr.cfg).

Thanks to Roland, W9HPX, for providing the following update re., the DVAP calibration offset:

The dvap_rptr.cfg file contains a variable called DVP_OFF. This is the calibration value set at the factory to put the DVAP on the correct frequency.

DVAP users should read the value of this parameter using DvapUtil 1.02 from Robin's web site: <http://www.dvapdongle.com/Downloads.html> and enter it into the dvap_rptr.cfg file. If it is left as 0 it will set the DVAP's calibration value to 0, which may, or may not be correct for your DVAP.

Editing g2_ircddb.cfg

You will need to edit the g2_ircddb.cfg file (in the /root/g2_ircddb folder) to update your Gateway coordinates and frequency - as shown below:

```
# This is for APRS/GPS
SEND_APRS=Y
# The repeater module specifications
#   LATITUDE, LONGITUDE, RANGE, DESCRIPTION
# If you do NOT want to define the aprs parameters for a module,
#   then leave the line BLANK

RPTR_ID_A=
RPTR_ID_B=43.5930,-79.78917,10,FREE STAR* DV 443.8125 +5.00 MHz
RPTR_ID_C=43.5930,-79.78917,10,FREE STAR* DV 147.5400 +0.00 MHz
```

← EDIT the above lines to suit your location / frequencies. If you only have one module, leave the other 2 lines blank (do not delete them!)

```
# Which local address on your Linux box will be used
#   to report to the remote IRC database server(group2, group1,...)
# For most installations it is 0.0.0.0 (which binds to all local IP addresses)
# If you have more than one ethernet card, then you may set it to a specific local IP
#   that is assigned to that specific ethernet card
# If you have one ethernet card, you can use 0.0.0.0
LOCAL_IRC_IP=0.0.0.0
```

```
# This is for the Live IRC site
SEND_QRGS_MAPS=Y
# LATITUDE, LONGITUDE, description1, description2, url
QTH=43.5930, -79.78917, Mississauga ON, BeagleBone, www.freestar.us ← EDIT to suit
# Frequency, shift, range, antenna's height above ground(agl)
QRG_A=
```

QRG_B=443.8125, 5, 10, 20 ←EDIT these 2 lines.
QRG_C=147.540, -0.0, 10, 20 ← If you only have one module, leave the other 2 lines blank
Once you have edited this file, save / exit... then restart g2_ircddb as follows:

NOTE: if this is a personal hotspot, please change the ircddb server setting to use the FREE STAR* IRC cluster:

```
IRC_DDB_HOST=freestar-irc-cluster.ve3lsr.ca
```

```
service g2_ircddb restart
```

Changes to crontab for the web Dashboard

`crontab -e` to get to the crontab editor.

Locate this line:

```
# Webpage updates  
*/2 * * * * /root/g2_link/g2_lh -m "BC" -o /var/run/lighttpd/index.html -a 192.168.1.124 -b  
"VE3URU D-STAR" -t /root/g2_link/g2_lh.tpl VE3URU
```

The -m switch denotes which modules your system has - in the example above, I have module B and C; if your system has (for example, just one module say, module C, you will specify -m "C" or -m "B" for module B.

The -a switch denotes the IP address, replace 192.168.1.124 with your IP address

The -b switch denotes the banner tag - edit to suit

Replace VE3URU with your Gateway Callsign.

Log Files

The image, as supplied, has logging disabled. This is to preserve CPU cycles so that we can get optimal system response, the best audio quality possible and preserve the life of the SD card. For troubleshooting, you may wish to enable logging; this can be done by editing the following:

Package	Enable Logging (By Editing)	Log File
g2_ircddb	/etc/init.d/g2_ircddb	/var/log/g2_ircddb.log
g2_link	/etc/init.d/g2_link	/var/log/g2_link.log
rprr	/root/rprr/rprr.sh	/var/log/rprr.log
dvrprr	/root/dvrprr/dvrprr.sh	/var/log/dvrprr.log
dvrprr2	/root/dvrprr/dvrprr2.sh	/var/log/dvrprr2.log

dvap_rptr	/root/dvap_rptr/dvap_rptr.sh	/var/log/dvap_rptr.sh
-----------	------------------------------	-----------------------

We recommend disabling logging once your system is online and stability has been confirmed.

That's it – welcome to the FREE STAR* project.

Please drop me a note (Ramesh@va3uv.com) so that I can add you to my mailing list for system updates, notices, etc.

Appendix A

Radio Programming

Radio Settings

The following sections are intended to be used as a guide when programming user radios. Of course the repeater callsigns will need to be modified for your particular system.

Simplex Access Points

ICOM radios do not send RPT1 and RPT2 information in the DV data stream when the radio is in simplex mode. We therefore have to program the radio as follows:

Mode = DV

Duplex enabled (set for + or – DUP)

Set the offset frequency to Ø

URCALL = CQCQCQ

RPT1 = callsign of the simplex access point followed by the module (e.g., VE3RSD B)

RPT2 = callsign of the simplex access point followed by 'G' for Gateway (e.g., VE3RSD G).

Note that the ICOM G2 protocol reserves six (6) characters for the callsign. The module / gateway suffix is entered into the eighth (8th) position. Thus a system with a five character callsign will require two spaces in the 6th and 7th position – e.g:

RPT1 = VA3ZZ<space><space>B

RPT2 = VA3ZZ<space><space>G.

Repeater Systems

Mode = DV

Duplex enabled (set for + or – DUP)

Set the offset frequency as required (e.g., 5MHz for UHF) – this should already be set as the default.

URCALL = CQCQCQ

RPT1 = callsign of the repeater followed by the module (e.g., VA3PMO B)

RPT2 = callsign of the repeater followed by 'G' for Gateway (e.g., VA3PMO G).

Appendix B

DTMF Shortcuts

The FREE STAR* solution supports DTMF dialing to connect and disconnect from reflector systems. The table below summarizes the DTMF syntax in comparison to traditional 'URCALL' radio programming:

Action	URCALL	DTMF
Link to XRF005A	XRF005AL	#00501
Link to REF016B	REF016BL	*01602
Link to DCS006B	DCS006BL	D00602
Link to DCS014K	DCS014KL	D01411
Link to ICOM repeater WC7SO module C	WC7SO<space>CL	N/A
Query Link Status	I (in the 8 th position)	99
Disconnect	U (in the 8 th position)	73

The last two digits of the DTMF string represent the numerical equivalent of the remote module that you wish to connect to. This was a necessary change, which accompanied g2_link 3.05 in order to accommodate DCS reflectors, which have 26 modules (five weren't enough!).

To link your repeater module to an ICOM G2 system running dplus, set URCALL = the repeater callsign, followed by the module in the 7th position and the command in the 8th position of URCALL – e.g., to link to WC7SO module C, set URCALL = WC7SO CL. Note that there is a space between the 'O' and 'C' since WC7SO is a five letter call.

Appendix C

Callsign Routing

Callsign Routing

Callsign routing is a powerful feature of the G2 protocol that allows users to establish communication with each other without the need to know which repeater each user is on. This is very handy when a particular user may be taking a long road trip traversing multiple repeater footprints.

Callsign routing – does however, require users to be on the same ircDDB server – whether it be on the same production network, or other test servers. Note the FREE STAR* IRC servers support callsign routing as well as DTMF initiated routing. Please contact us for details.

Callsign (or G2) routing relies on ‘last heard’ information in the ircDDB network (i.e., the last heard position) for each user. Hence when a user traverses multiple repeaters, it is imperative that the user transmits at least once on each repeater so that the repeater can report his / her presence on that repeater to the ircDDB network. This of course relies on each repeater running some form of ircDDB client.

Important Note: The ircDDB Gateways cache the last heard information (user callsign along with gateway callsign and gateway IP address. Callsign routing requests trigger the software to first query the local cache for the remote callsign. If the remote callsign is not available from the local cache, the initial routing request will fail. However, the software queries the ircDDB server for the remote callsign and last heard information making it available for a subsequent routing request – in most cases this will be available within seconds of the first (failed) request.

To contact a user using Callsign (or G2) routing, both users must set each other’s callsign in the URCALL field. The following example illustrates a G2 routing call between VA3UV and WA1SOT:

	VA3UV Radio Settings	WA1SOT Radio Settings
URCALL	WA1SOT	VA3UV
RPT1	VE3RSD C	KB1UAP B
RPT2	VE3RSD G	KB1UAP G

The above settings should remain in effect for the entire QSO. When making the initial call, it is important to state that you are using callsign routing so that the station you are calling can use the ‘one-touch’ (RX>CS) feature on ICOM radios to reply. The ‘RX>CS’ button forces the received callsign to be populated into the URCALL field so that when the remote user replies, his call will be routed back to you.

Please note that callsign routing may be initiated at any time, even when the local

repeater (or access point) module is connected to a reflector. Using a URCALL value other than 'CQCQCQ' blocks data from going out over the reflector. So in the above example, if VA3UV is connected to reflector XRF005A, he can initiate the call to Jim, WA1SOT, using callsign routing as shown above. The call to Jim will not be heard by the users / repeaters connected to the reflector. It will however be broadcast out on the repeater module where Jim was last heard.

Once the call has been established, both parties should disconnect from any reflector or repeater that they may be connected to. This avoids the possibility of clashing / interfering packets coming off the reflector while the repeater is busy.

Callsign routing is very powerful; with the recent addition of ircDDB, it can be used to enable communication between users on home-brewed systems and ICOM G2 systems – providing that the ICOM G2 system is running the ircDDB add-on software.

Appendix D

D-PRS Support and Radio Programming

D-PRS is a process used to convert raw GPS data into TNC2 formatted APRS messages. D-PRS processing capability is included in g2_ircddb. FREE STAR* systems are therefore able to receive, process, and report position reports from mobiles, portables, command posts, etc., to the APRS-IS server network via a TCP socket over the public Internet.

D-PRS support must be enabled by setting the 'SEND_APRS' parameter (in /root/g2_ircddb/g2_ircddb.cfg) to 'Y'.

The example below illustrates the beacon parameters for a simplex system operating on 2m (port 'C')

```
SEND_APRS=Y  
RPTR_ID_A=  
RPTR_ID_B=  
RPTR_ID_C=XX.XXXX,-YY.YYYY,ZZ,FREE STAR* DV 147.540 -0.00 MHz
```

XX.XXXX corresponds to the 'LATITUDE' in decimal degrees, use a negative number for the southern hemisphere.

YY.YYYY corresponds to the 'LONGITUDE' in decimal degrees, use a negative number for the western hemisphere.

ZZ is the coverage area (radius) in miles.

FREE STAR* DV 147.540 -0.00 MHz is the beacon message; replace 147.540 with your repeater frequency, followed by the offset +/- in MHz as shown.

The FREE STAR* team supports the older style GPS and GPS-A format. Some of the earlier D-STAR radios such as the ID-800 and IC-91AD do not support GPS-A format and therefore are confined to the older GPS format. One of the advantages of the older style GPS format is that it continually sends GPS data in the slow speed data accompanying each transmission. This allows remote users to monitor your position and distance, etc., on their radio(s).

The following tables illustrate the configuration of an IC-2820 and IC-92AD for DPRS use:

The table below illustrates the 2820 menu hierarchy:

DV GPS	GPS Sentence	RMC ON
		GGA ON
		Everything else OFF
	GPS Message	See 1 below
	RX GPS Message	Data: _____
	GPS-TX	See 2 below
	GPS Auto TX:	OFF (see 3 below)
GPS-A SET MODE	UNPROTO ADDR	API282,DSTAR* (see 4 below)
	DATA EXTENSION	COURSE/SPEED
	TIME STAMP	HMS
	GPS-A SYMBOL	<Select to suit>
	COMMENT (DATA EXT)	<Your comment>

1. For 'GPS' mode, the GPS-TX message is a combination of the symbol, status text message and a checksum. Please use the [D-PRS Calculator](#) from Peter Loveall (AE5PL) to determine the GPS-TX message.
2. Set to 'GPS' for the older style GPS format, or GPS-A for the newer GPS-A format.
3. GPS Auto TX: By setting GPS Auto TX to 'OFF' this will only send out your position data when you manually transmit. Auto-beaconing should not be enabled when connected to a reflector as the beacon will generate traffic over the reflector causing disruption to all users of that reflector / linked repeaters, etc.
4. UNPROTO: Format is APInnn where 'nnn' is the first 3 digits of your radio model # (e.g., 800, 880, 282). In the case of the IC-80, IC91 and IC92, set the UNPROTO to API80 / API91 / API92.

NOTE: Unlike conventional APRS, SSID's are not supported in DPRS. The solution to this is to use a terminal suffix. For example, a station can have a home-based APRS station with the call VA3XYZ. If the same user were to use a mobile APRS or DPRS station with the same callsign, this would conflict with the home-based station. Thus a second terminal will be required – e.g., 'VE3XYZ M' (Note the 'M' is the terminal, which is specified in the 8th position of the user's 'MyCall' settings).

ICOM IC-92AD Setup – Information supplied by Colin, VE3MSC:

The table below illustrates the 92AD menu hierarchy:

Press "MENU" and then scroll using the red arrows (up ^ on 8 key and down v on 2 key)

to DV SET MODE, and press Enter (5 key).

DV SET MODE	DV DATA TX	PTT - <i>if you wish to send your position/comment when you push to talk. Good for battery saving.</i> AUTO - <i>will send periodically based on GPS AUTO TX setting (described below)</i>
DV SET MODE	GPS TX MODE	GPS-A

Then press Enter (5 key) or RIGHT > (6 key) for GPS-A SET, for the following sub-menu:

GPS-A SET MODE	UNPROTO ADD	API92,DSTAR*
	DATA EXTENS	COURSE/SPEED
	TIME STAMP	HMS
	GPS-A SYMBO	<Select to suit> OTHER: /[for Jogger You will need to hit the 3 key red a/A to get the right character set.
	COMMENT (EXT)	<Your comment>

Then escape back out to the DV SET MODE menu by pressing LEFT < (4 key) a couple of times. Then scroll down to GPS AUTO TX

DV SET MODE	GPS AUTO TX	OFF
-------------	-------------	-----

You will also want to adjust your preferences and time zone offset.

Press "MENU" and then scroll using the red arrows (up ^ on 8 key and down v on 2 key) to MESSAGE/POS and press Enter (5 key)

MESSAGE/POS	GPS SET MOD	
-------------	-------------	--

Scroll to GPS SET MOD and press Enter. The following sub-menu will appear:

GPS SET MODE	GPS SPEED	4800 <i>seems to be default for use with the HM-175GPS SP/MIC</i>
	FORMAT	ddd° mm' ss" <i>Personal preference.</i>
	UNITS	m = Metres in Canada ft/ml = for USA feet miles <i>Personal preference.</i>
	COMPASS DIR	NORTH REF
	UTC OFFSET	-05:00 for Eastern Standard Time. <i>Remember to change it for daylight saving time, e.g. -04:00 for EDT. This is important!</i>

The same comments apply re., use of a terminal 'suffix' to avoid conflict of position updates with (for example) a home based station with the same callsign. The procedure below illustrates the addition of a second terminal 'P' for portable use.

For example:

Press "MENU" and scroll up to CALL SIGN.

CALL SIGN	VA3XYZ P	<i>Do this only if you need to specify a terminal. If this is your only rig on APRS/DPRS then you don't need to do this.</i> <i>Conventions used: P for Portable M for Mobile Note the terminal suffix is specified in the 8th position of the Call Sign field.</i>
-----------	----------	---