

Amateur Radio Notes ♪

by Toshen, KE0FHS

[CQ](#) · [Base](#) · [D-STAR](#) · [DMR](#) · [Hotspots](#) · [Pi-Star](#)

Pi-Star notes

Revised: Apr 2023; CC BY-SA [↗](#); PDFs [↗](#)

As of Jan 1, 2024, this site is no longer being updated. It will remain available for some time for legacy reference.

Most up-to-date version: amateurradionotes.com/pi-star-notes.htm

[Playing with Pi-Star](#)

[Pi-Star cross modes](#)

[Pi-Star notes](#) [[Quick links](#) v]

Jump to: [Links to helpful resources](#)

Jump to: [A few other notes ...](#)

[Pi-Star troubleshooting](#)

Links to helpful resources

- a. [Pi-Star: sites](#)
- b. [Pi-Star: forums & groups](#)
- c. [Pi-Star: articles](#)
- d. [Pi-Star: presentations](#)
- e. [Pi-Star: videos](#)
- f. [Pi-Star: hardware and software](#)
- g. [Pi-Star: tools](#)
- h. [Pi-Star: related GitHub repositories](#)

A few other notes ...

[^ Top](#) | [Quick links v](#)

a) Pi-Star: sites

- [Pi-Star website](#)
- [BrandMeister talkgroup list](#)
- [DMR+ reflector list](#)
- [YSF reflector list](#)
- [FCS reflector list](#)
- [P25 reflector list](#)
- [NXDN reflector list](#)
- [NXDN Forum](#)

[^ Top](#) | [Quick links v](#)

b) Pi-Star: forums & groups

- [Pi-Star User Forum](#) and [Pi-Star Support Group](#)
- [Nextion Screens group](#) and [Nextion Ham-Radio Screens group](#)
- [Official ZUMspot and ZUM Radio Products Group](#)

[^ Top](#) | [Quick links v](#)

c) Pi-Star: articles

- [Manually adding WiFi settings to RPi \(below\)](#), by Andy Taylor, MW0MWZ
- [Raspbian explained](#), a good article about Raspbian, Debian, and Linux
- [Pi-Star Configuration](#), 2017, by Barry, K0BSB
- [Secure Shell \(SSH\)](#), Wikipedia
- [ZUMspot & MMDVM Demo page](#), 2017, by Ron, VE1AIC
- [U.S. Band Plan](#), by ARRL

- [Colorado Frequency Use Plans](#) (especially, [CCARC Guidelines on Hotspots](#))
- For the frequency plans for other U.S. States, see [Amateur Radio: Repeater Coordinators](#), a list by Tom, W2XQ
- [Digital Voice frequencies](#), by Ron, VE1AIC

[^ Top](#) | [Quick links](#) v

d) Pi-Star: presentations

- The [PAPA System](#) provides its members with excellent guides and tutorials related to digital radio including two by Dave, KC6N: ZUMspot/Pi-Star presentation, 2018, and Digital Voice for Amateur Radio, 2018 (the PAPA System, which is a 501(c)(3) organization, and operates for the benefit of the amateur radio community, offers trial guest memberships)
- [ZUMspot with Pi-Star](#), 2017, by David Cappello, KG5EIU

[^ Top](#) | [Quick links](#) v

e) Pi-Star: videos

- [The official Pi-Star videos](#), by Craig, W1MSG
- [Pi-Star - 12 Months On](#), ICQ Podcast Episode 274, Aug 2018 (interview with Andy Taylor, MW0MWZ begins at 1:10:00 and lasts 35 minutes)
- [Raspberry Pi/DVMEGA Radio Hotspot with Pi-Star Image](#), 2017, by N0NO
- [The Road to the MMDVM](#), Jonathan Naylor, G4KLX, Pacificon 2018
- [Multimode Digital Voice Modem](#), Jim McLaughlin, KI6ZUM, Pacificon 2018
- [MMDVM](#), 2017, Jonathan Naylor, G4KLX
- [irdDDB Remote Android app](#), by Michael Carey, VK5ZEA
- [ircDDB Remote Control Pi-Star \(for iOS\)](#), by David Cappello, KG5EIU
- [How to SSH into Pi-Star hotspot](#), 2017, by David Cappello, KG5EIU

[^ Top](#) | [Quick links](#) v

f) Pi-Star: hardware & software

- [Pi-Star Downloads](#), by Andy Taylor, MW0MWZ
- [Pi-Star Beta Downloads \(including Release Candidates\)](#), by Andy Taylor, MW0MWZ

- If you want to have some fun experimenting, check out the Pi-Star fork being developed by Daniel, F1RMB: [GitHub: f1rmb/Pi-Star_DV_Dash](#) ↗.
- [Nextion screen layouts](#) ↗, by Jonathan Naylor, G4KLX
- [Raspberry Pi Pinout](#) ↗
- [Ham Radio Outlet \(HRO\)](#) ↗, hotspots and digital radios
- [FriendlyARM NanoPi](#) ↗ and [NanoPi Air](#) ↗
- [Hardkernel ODROID](#) ↗
- [Xunlong Software Orange Pi](#) ↗

[^ Top](#) | [Quick links](#) v

g) Pi-Star: tools

- [Pi-Star WiFi Builder](#) ↗
- CCS7 ID registrations:
 - Registrations for everywhere—including North America, South America, Asia, and Oceania, Europe, and Africa—are now all handled by [RadioID.net](#) ↗.
- [QuadNet2 USA IRC Network](#) ↗ – open style D-STAR routing
- [Etcher](#) ↗ - an open source SD Card Imager app by balena that's available for Windows, Mac, and Linux
- [SD Memory Card Formatter](#) ↗ - single purpose app for Windows and Mac.
- [Terminus](#) ↗ - SSH client for Android, iOS, Mac, Linux, Windows
- [PuTTY](#) ↗ - SSH client for Windows
- To find QTH locator: [APRS.fi](#) ↗
- [Online YSF Reflectors and YSFReflector Registry](#) ↗

[^ Top](#) | [Quick links](#) v

g) Pi-Star: related GitHub repositories

This is a list of GitHub repositories related to Pi-Star, which can be interesting to watch if you want to stay tuned into the latest developments.

Thanks to the team at [Colorado Digital Multiprotocol](#) and the participants of the [Colorado HD \(Hotspot Discussion\) net](#) for putting this list together.

- Andy Taylor, MW0MWZ, the developer of Pi-Star, maintains the [AndyTaylorTweet profile](#). It's fun to keep an eye on the [The main repositories page](#), where you can see the recent activity in all his repositories. When there is activity in a repository, you can open its page and from there open its Commits listing, and then you drill down into individual commits to view the actual code changes.
- Jonathan Naylor, G4KLX, maintains the [g4klx profile](#), where you can find many of the key projects that make up the foundation of Pi-Star, for example, [MMDVMHost](#) and [DMRGateway client](#), among others.
- Andy, CA6JAU, maintains the [juribeparada](#) profile, where you can find the [MMDVM_HS firmware project](#), as well as the [MMDVM_CM cross modes project](#) (DMR2YSF, YSF2DMR, etc.).
- Rob van Rheenen, PD0DIB, maintains the [PD0DIB profile](#), where you can find his work on the [Nextion Ham Radio Screens project](#).
- You can find more profiles and repositories listed on the document by Colorado Digital Multiprotocol: [GitHub and Amateur Radio Digital](#)

[^ Top](#) | [Quick links v](#)

A few other notes

1. [SSH Access to Pi-Star](#)
2. [Port forwarding](#)
3. [Radio/modem types supported](#)
4. [Entering read-write mode](#)
5. [Running commands as the root user](#)
6. [Expanding the filesystem](#)
7. [Manually adding or overriding repeaters and reflectors](#)
 - a. [Method for overriding default XLX rooms](#)
8. [Updating hotspot firmware via Pi-Star](#)
 - a. [Updating ZUMspot/MMDVM_HS, etc.](#)
 - b. [Updating DVMEGA mounted on an RPi](#)
9. [Fine tuning to reduce high BER](#)
10. [Manually adding WiFi settings to RPi](#)
11. [Connecting a Pi-Star hotspot via cell phone](#)
 - a. [Connecting via WiFi](#)
 - b. [Connecting via USB tethering](#)
12. [Re-syncing system time and changing date format](#)
13. [DMRGateway notes](#)
14. [D-STAR routing](#)
15. [Watchdogs, Remote RF commands, and Keepers](#)
16. [Pi-Star's BrandMeister Manager module](#)
17. [Adding a custom NXDN reflector to Pi-Star](#)
18. [CSS Tool](#)
19. [RSSI Expert Editor](#)
20. [Prevent sending APRS info via BrandMeister](#)
21. [Upgrading to Pi-Star V4](#)

22. [Linking to reflectors from the command line](#)
23. [A few Raspberry Pi OS commands](#)
24. [Pi-Star fork by F1RMB](#)
25. [DMRGateway rewrite rules](#)

[^ Top](#) | [Quick links v](#)

1) SSH Access to Pi-Star



You can use SSH to access Pi-Star directly from the Pi-Star Expert Editor:

1. Enter the Expert Editor:
 - On the Configuration page, click the Expert link.
 - Or, for macOS, Linux, and Windows 10, you can use the URL: <http://pi-star.local/admin/expert>.
For older versions of Windows: <http://pi-star/admin/expert>.
2. Click the SSH Access link and log in.

To use the full screen version of the SSH Access client, click the "fullscreen SSH client" link at the bottom of the window and log in.

Alternatively, use a standalone SSH app

Using a standalone SSH app is recommended for running Pi-Star updates and upgrades:

- SSH client for Android, iOS, Mac, Linux, Windows: [Termius](#) .
- Note:* You can find the IP address that your Pi-Star hotspot is using in the Wireless Configuration section of Pi-Star.
- SSH client for Windows: [PuTTY](#) .
- Built-in macOS SSH access: open the Terminal.app and type:

```
ssh pi-star@pi-star.local
```

Some SSH commands that can be run

- Run a Pi-Star Update:

```
sudo pistar-update
```

Updates are run overnight automatically, but you can run this command as the root user to force an update at anytime. Must be logged into Pi-Star. Running Pi-Star Update from the dashboard

updates the dashboard binaries, and hostfiles. Running an update from an SSH client (`sudo pistar-update`) also updates the OS.

- Run a Pi-Star Upgrade:

```
sudo pistar-upgrade
```

Upgrades the base system services and packages. Upgrades are available less often than updates.

- Initiate a system halt (-h) and shutdown immediately (now):

```
sudo shutdown -h now
```

- Initiate a shutdown immediately (now), and then restart (-r)

```
sudo shutdown -r now
```

- Initiate a reboot:

```
sudo reboot
```

- Find a file:

```
sudo find / -name file.txt
```

for example:

```
sudo find / -name DPlus_Hosts.txt
```

- View CPU info:

```
cat /proc/cpuinfo
```

- View free memory info:

```
free -m
```

- View disk space info:

```
df -h
```

- Restart the webserver that hosts the dashboard:

```
sudo service nginx restart
```

- Check whether WiFi Power Save is on:

```
iw wlan0 get power_save
```

Note 1: This can be helpful if you're losing WiFi connectivity when the hotspot has been running for awhile.

- Find information about your current radio/modem board and firmware version:

```
sudo pistar-findmodem
```

- Run a jitter test (requires a single argument, the prefix of the group of hosts you want to check as used in the `/usr/local/etc/DMR_Hosts.txt` file), for example:

```
sudo pistar-jittertest BM_Unites_States
```


The results look like this:

```
pi-star@pi-star(ro):~$ sudo pistar-jittertest BM_United_States
BM_United_States_3102      Avg RTT: 43.83ms      Peak Deviation: 8.30ms
BM_United_States_3101      Avg RTT: 59.83ms      Peak Deviation: 4.30ms
BM_United_States_3103      Avg RTT: 45.40ms      Peak Deviation: 1.20ms
```

- For other Pi-Star script commands, see: `/usr/local/sbin`
- See also: [23\) A few Raspberry Pi OS commands](#)

See also:

- [4\) Entering read-write mode](#)
- [5\) Running commands as the root user](#)
- [6\) Expanding the filesystem](#)
- [12\) Re-syncing system time and changing date format](#)
- [22\) Linking to reflectors from the command line](#)
- [How to SSH into Pi-Star hotspot](#) [🔗](#) (video), by David Cappello, KG5EIU.

[^ Top](#) | [Quick links](#) [v](#)

2) Port forwarding

Port forwarding is becoming less necessary for hotspots, but may be needed for QuadNet Smart Group and some X-Reflectors (XRF).

- **For QuadNet Smart Groups** – In order to use Smart Groups with a home hotspot, you need to set up port forwarding of port 40000. For this purpose, it's necessary to forward public and private UDP.
- **For XRF** – If your router doesn't support uPNP and you don't want to enable "Use DPlus for XRF" in Pi-Star's D-STAR Configuration section, you can set up port forwarding for XRF. Per Andy Taylor in the Pi-Star Users Support Group: "If your router supports uPNP, Pi-Star will request the ports required to be forwarded, however many people disable uPNP and so you may have to do some manual port forwarding. REF/DCS reflectors can be connected without any port forwards, but XRF reflectors require port UDP/30001 to be forwarded to your Pi-Star's IP address."

Note: Apple routers don't support uPNP. For info about setting up port forwarding, see: [AirPort – Port Mapping Basics using AirPort Utility v6.x](#) [🔗](#).

3) Radio/modem types supported

1. Icom Repeater Controller ID-RP2C (DStarRepeater Only)
2. DVMEGA Raspberry Pi Hat (GPIO) - Single Band (70cm)
3. DVMEGA Raspberry Pi Hat (GPIO) - Dual Band
4. DVMEGA on Arduino (USB - /dev/ttyUSB0) - Dual Band
5. DVMEGA on Arduino (USB - /dev/ttyACM0) - Dual Band
6. DVMEGA on Arduino (USB - /dev/ttyUSB0) - GMSK Modem
7. DVMEGA on Arduino (USB - /dev/ttyACM0) - GMSK Modem
8. DVMEGA on Bluestack - Single Band (70cm)
9. DVMEGA on Bluestack - Dual Band
10. GMSK Modem (USB DStarRepeater Only)
11. DV-RPTR V1 (USB)
12. DV-RPTR V2 (USB)
13. DV-RPTR V3 (USB)
14. DVAP (USB)
15. MMDVM / MMDVM_HS / Teensy / ZUM (USB)
16. STM32-DVM / MMDVM_HS - Raspberry Pi Hat (GPIO)
17. STM32-DVM (USB)
18. ZUMspot Libre (USB)
19. ZUMspot - USB stick
20. ZUMspot - Single Band Raspberry Pi Hat (GPIO)
21. ZUMspot - Dual Band Raspberry Pi Hat (GPIO)
22. ZUMspot - Duplex Raspberry Pi (GPIO)
23. ZUM Radio-MMDVM for Pi (GPIO)
24. ZUM Radio-MMDVM-Nucleo (USB)
Note: The Nucleo is for repeater and high power hotspot applications.
25. MicroNode Nano-Spot (Built In)
26. MicroNode Teensy (USB)
27. MMDVM F4M-GPIO (GPIO)
28. MMDVM F4M/F7M (F0DEI) for USB
29. MMDVM_HS_Dual_Band for Pi (GPIO)
30. MMDVM_HS_Hat (DB9MAT & DF2ET) for Pi (GPIO)

31. MMDVM_HS_Hat Dual (DB9MAT, DF2ET & DO7EN) for Pi (GPIO)
32. MMDVM_HS_Hat Dual (DB9MAT, DF2ET & DO7EN) for Pi (USB)
33. MMDVM_HS_AMBE (D2RG HS_AMBE) for Pi (GPIO)
34. MMDVM_RPT_Hat (DB9MAT, DF2ET & DO7EN) for Pi (GPIO)
35. MMDVM_HS_MDO Hat (BG3MDO) for Pi (GPIO)
36. MMDVM_HS_NPi Hat (VR2VYE) for Nano Pi (GPIO)
37. MMDVM_HS_Hat Dual (VR2VYE) for Pi (GPIO)
38. Genesis - HHDVM_HS_Hat for Pi (GPIO)
39. Genesis Dual - HHDVM_HS_Hat_Dual for Pi (GPIO)
40. LoneStar - MMDVM_HS_Hat for Pi (GPIO)
41. LoneStar - MMDVM_HS_Dual_Hat for Pi (GPIO)
42. LoneStar - USB Stick
43. SkyBridge - MMDVM_HS_Dual_Hat for Pi (GPIO)
44. MMDVM_NANO_DV (BG4TGO) for NanoPi Air (GPIO)
45. MMDVM_NANO_DV (BG4TGO) for NanoPi Air (USB)
46. OpenGD77 DMR hotspot (USB)
For more info, see: [OpenGD77](#)

[^ Top](#) | [Quick links v](#)

4) Entering read-write mode

By default, Pi-Star is in read-only mode. To toggle:

1. Log into Pi-Star via SSH.
2. To enable read-write mode:

```
rpi-rw
```

The command line prompt will change

```
from: pi-star@pi-star(ro):tmp$
```

```
to: pi-star@pi-star(rw):tmp$
```

3. To re-enable read-only mode:

```
rpi-ro
```

[^ Top](#) | [Quick links v](#)

5) Running commands as the root user

To run commands as the root user:

1. Log into Pi-Star via SSH.
2. Run a single command as the root user:

```
sudo [cmd]
```

for example:

```
sudo pistar-update
```

To switch user (su) to the root user (proceed carefully when using the root user; you can easily mess things up):

1. Log into Pi-Star via SSH.
2. Switch to the root user:

```
sudo su
```

The command line prompt will change

from: `pi-star@pi-star(ro):tmp$`

to: `root@pi-star-m(ro):pi-star#`

– or –

To switch to the root user and to the root user's home directory (/root):

```
sudo su -
```

The command line prompt will change

to: `root@pi-star-m(ro):~#`

3. To exit the root user:

```
exit
```

There's a good answer about sudo su and related commands on [AskUbuntu.com](https://askubuntu.com) .

[^ Top](#) | [Quick links v](#)

6) Expanding the filesystem

Note: For Pi-Star version 4 and above, this happens automatically as part of the first boot up.

Pi-Star runs in read-only mode and takes up very little space on the microSD card. If Pi-Star is used for a hotspot, 4 GB usually is more than enough space. However, if Pi-Star is used for a repeater or you run it hard, 8 GB is probably better, and you should expand the filesystem to take up all the available space.

In Raspberry Pi OS, the command to expand the filesystem to fill the SD card is part of the raspi-config utility, but that utility isn't available Pi-Star; instead, run the Pi-Star expand command:

1. Log into Pi-Star via SSH.

2. Optionally, look at the pre-expansion disk space used by the filesystem in human-readable format (pay attention to the `/dev/root` numbers):

```
df -h
```

3. Expand the root partition:

```
sudo pistar-expand
```

(*Note:* "pistar" with no hyphen)

4. *Important!* Reboot Pi-Star.
5. Optionally, look at the post-expansion numbers.

Example: When I ran `pistar-expand` on the filesystem on my 16 GB microSD card to test it out, I ended up with 1.28 GB used (9%) and 13.37 GB free.

[^ Top](#) | [Quick links v](#)

7) Manually adding or overriding repeaters and reflectors

Pi-Star updates the lists of reflectors and repeaters stored in the hosts files (`/usr/local/etc`) nightly, but there may be times when you want to use one that isn't listed or override an incorrect entry. If you add the changes to the regular hosts file, they will be overwritten the next time a Pi-Star Update is run, either by the automatic overnight method or a manual update. To get around this, you can add the changes manually to the override hosts file in the `/root` directory, which is the root user's home directory. Once you've done this, you can run a regular Pi-Star Update and your changes will be appended to the regular hosts file.

1. Log into Pi-Star via SSH.
2. Enable read-write mode:

```
rpi-rw
```

3. Switch user so that you're in root's home directory (`/root`) with root's privileges:

```
sudo su -
```

4. Edit the appropriate hosts file, for example:

```
nano DExtra_Hosts.txt
```

Note 1: The filename, including capitalization, must be exact. Hosts files that are available: `DCS_Hosts.txt`, `DMR_Hosts.txt`, `DPlus_Hosts.txt`, `NXDNHosts.txt`, `DExtra_Hosts.txt`, and `P25Hosts.txt`.

Note 2: There isn't a file for YSF Hosts. According to Andrew, M1DNS, in the Pi-Star User Forum, "The list is generated from the online registry. Register at <https://register.ysfreflector.de/> and Ur

reflector will automatically get added to the server that updates pistar."

Note 3: As of Pi-Star 4.1.0, there also isn't an available file for XLXHosts.txt anymore. I think that's because it was used for overriding default XLX rooms, and that override functionality is now handled via Configuration > DMR Configuration > DMR Master = DMRGateway > XLX Startup Module. **However**, there were [changes made to Pi-Star Dashboard 20200406](#) that might make it possible to use an override entry in the XLXHosts.txt file in the root folder. I haven't tested this, but it might be worth trying.

- 5. Make your additions or changes.

Note: For the DPlus_Hosts.txt file, you can add a capital L to the end of the line to force the entry in this file to override any duplicates in the main Pi-Star host file.

- 6. Write out the changes (^o), and exit (^x).

- 7. Exit root user:

```
exit
```

- 8. Re-enable read-only mode:

```
rpi-ro
```

- 9. Run a Pi-Star update so your changes get appended to the files in /usr/local/etc:

```
sudo pistar-update
```

Note: From what I've seen, the P25 and NXDN Hosts files work a bit different from the other modes. The other modes append the addition in the root hosts files to the hosts files with the same name in /usr/local/etc. But for P25 and NXDN, they are added to a different file in /usr/local/etc: P25HostsLocal.txt and NXDNHostsLocal.txt. Regardless, the outcome is the same. After you run the Pi-Star Update, you should find your added reflector in the P25 or NXDN Startup Host drop-down list on the Configuration page, and it won't get deleted by the automatic overnight update.

Here's an example of the file for adding a DPlus entry.

```
#####
# #
# DPlus_Hosts.txt - Overrides #
# #
#####
# #
# Add your over-ride lines below, #
# following the example #
# please make sure that lines are tab #
```

```
# delimited and NOT space delimited. #
# #
# Adding a capital L to the end of the #
# line will force the entry in this #
# file to override any duplicates #
# in the Pi-Star host files.#
# #
# #
#####
# Reflector IP/Hostname Lock #
#####
#REF000 1.2.3.4 L
XRF005 216.16.240.236 L
```

7a) Method for overriding default XLX rooms

As of Pi-Star Dashboard 20190115, this is now handled via Configuration > DMR Configuration > DMR Master = DMRGateway > XLX Startup Module.

[^ Top](#) | [Quick links v](#)

8) Updating hotspot firmware via Pi-Star


It's possible to update the firmware of several hotspot boards via Pi-Star.

Standalone article: [Updating hotspot firmware via Pi-Star](#) 

Related article: [Updating and upgrading Pi-Star](#) 

Note: These instructions are for hotspots, not repeaters.

Important! These instructions may not work for cloned boards made in China (thanks to Tim, VA7TLS, for pointing this out). For a possible workaround, see these posts in the Pi-Star User Forum (not tested by me, so proceed at your own risk):

- [HS Hat not working properly](#) 
- [Jumbospot - can't update/downgrade firmware](#) 

There are two different sets of instructions:

[a\) Updating ZUMspot/MMDVM_HS, etc.](#)

b) Updating DVMEGA mounted on an RPi

8a) Updating ZUMspot/MMDVM_HS, etc.

Current version as of Nov 8, 2020: [v1.5.2](#).

Beginning with Pi-Star dashboard v20181214, you can see in the Radio Info section which TCXO chip (12.288 or 14.7456 MHz) a ZUMspot/MMDVM_HS board is running. This info is needed to determine which firmware update to apply, for example, with the MMDVM_HS_HAT or MMDVM_HS_DUAL_HAT, there are separate firmware update script commands for the different chips.

Radio Info	
Trx	Listening
Tx	438.550000 MHz
Rx	438.550000 MHz
FW	ZUMspot:v1.4.14
TCXO	14.7456MHz

You also can find information about your current radio/modem board and firmware version by running the `pistar-findmodem` script command via SSH:

```
sudo pistar-findmodem
```

For more info about this script command, see [Find info about the radio/modem board and about what is using the hotspot's modem or USB port](#).

Firmware release notes on GitHub: [ZUMspot/MMDVM_HS](#).

Hint: It's possible to operate a hotspot with both WiFi enabled and an Ethernet cable connected (one or the other will take precedence). However, for some reason, having both might interfere with firmware upgrades. For best results, use one or the other, but not both. Thanks to Emory, WM3M, for this hint.

1. Log into Pi-Star via SSH.

2. Run a Pi-Star Update:

```
sudo pistar-update
```

Note: If there's anything unusual about the update process, for example, if there are unusual error messages or it finishes with a "mount: / is busy" message, it might be a good idea to rerun the update and also to reboot before proceeding:

```
sudo reboot
```

3. Run one of the following commands:

Note 1: It isn't necessary to first put Pi-Star into read-write mode (`rpi-rw`); the script takes care of that.

Note 2: When you first enter one of the following commands, you'll see a message showing the current version of firmware that is installed on your hotspot, for example:

```
MMDVM_HS_Hat-v1.3.3 20180224 ADF7021 FW by CA6JAU GitID #d90263f
```

Press any key to write the hs_hat firmware to this modem or Ctrl-C to quit...

If you press any key to continue, the next thing you'll see is a message about the new version of firmware that is being installed, for example:

```
Flashing your hs_hat modem to the latest version
```

```
--2018-05-23 08:41:20-- http://github.com/juribeparada/MMDVM_HS/releases/download/v1.3.6/  
install_fw_hshat.sh
```

- ZUMspot board connected to GPIO:

```
sudo pistol-zumspotflash rpi
```
- MMDVM_HS_Hat board with 14.7456 MHz TCXO¹ connected to GPIO:

```
sudo pistol-mmdvmhshatflash hs_hat
```
- MMDVM_HS_Hat board with 12.288 MHz TCXO connected to GPIO:

```
sudo pistol-mmdvmhshatflash hs_hat-12mhz
```
- MMDVM_HS_DUAL_Hat board with 14.7456 MHz TCXO connected to GPIO:

```
sudo pistol-mmdvmhshatflash hs_dual_hat
```
- MMDVM_HS_DUAL_Hat board with 12.288 MHz TCXO connected to GPIO:

```
sudo pistol-mmdvmhshatflash hs_dual_hat-12mhz
```
- Nano Hat board connected to GPIO:

```
sudo pistol-vyehsflash nano_hs
```
- HS_DUAL_HAT (VR2VYE) connected to GPIO:

```
pistol-vyehsflash hs_dual_hat
```
- NanoDV NPi board:

```
sudo pistol-nanodvflash pi
```
- NanoDV USB board:

```
sudo pistol-nanodvflash usb
```
- BD7KLE/BG3MDO devices:

```
sudo pistol-mdoflash
```
- ZUMspot duplex board connected to GPIO:

```
sudo pistol-zumspotflash rpi_duplex
```
- ZUMspot USB key:

```
sudo pistol-zumspotflash usb
```

- USB-connected Libre Modem:

```
sudo pistar-zumspotflash libre
```

4. A prompt will tell you to press any key to begin.
5. When the update is finished, another prompt will tell you to press any key to reboot Pi-Star. Be patient, this step takes a bit of time during which it doesn't look like anything is happening.
6. Once the firmware update is complete, you can see the new version in the Radio Info module on the Pi-Star Dashboard or Admin pages.

[1] TCXO = Temperature Compensated Crystal Oscillator

[2] The script for updating a ZUM Radio MMDVM-Pi board can be found here: [GitHub: juribeparada/MMDVM_man](#) .

Trouble re-opening the dashboard

Sometimes after performing a Pi-Star or firmware update, for some reason the dashboard won't re-open in the browser on my Windows 10 laptop when using the standard `http://pi-star/` or `http://pi-star.local/` URLs. When that happens, find the hotspot's IP address and use that. After the dashboard opens the first time using the IP address, the URLs should work again.

Downgrading firmware

For those rare situations when a firmware update breaks things, Michael Eckhoff, K5MRE, posted instructions for downgrading ZUMspot/MMDVM_HS firmware in the Pi-Star Users Support Group.

Followup note: I found out there is a script for downgrading the MMDVM_HS_Hat boards, but haven't tried it. **Use at your risk!**

- To downgrade MMDVM_HS_Hat with 14.7456 MHz TCXO connected to GPIO:

```
sudo pistar-mmdvmhshatdowngrade hs_hat
```
- To downgrade MMDVM_HS_Hat with 12.288 MHz TCXO connected to GPIO:

```
sudo pistar-mmdvmhshatdowngrade hs_hat-12mhz
```
- To downgrade MMDVM_HS_DUAL_Hat with 14.7456 MHz TCXO connected to GPIO:

```
sudo pistar-mmdvmhshatdowngrade hs_dual_hat
```
- To downgrade MMDVM_HS_DUAL_Hat with 12.2880 MHz TCXO connected to GPIO:

```
sudo pistar-mmdvmhshatdowngrade hs_dual_hat-12mhz
```

Here's the manual script by K5MRE. I've tried this and it worked, but there are no guarantees. **Use at your own risk!**

1. Log into Pi-Star via SSH.

2. Enable read-write mode:

```
rpi-rw
```

3. Download the firmware you want to use (see [juribeparada/MMDVM_HS](#)). Note, the URL in the following command is split into two lines, but enter it with no spaces:

```
wget https://github.com/juribeparada/MMDVM_HS/  
releases/download/v#.#.#/modem_fw.bin
```

where #.#.# is the version of the firmware you want to downgrade to and modem_fw.bin is the bin file that matches your board:

```
mmdvm_hs_dual_hat_fw-12mhz.bin
```

```
mmdvm_hs_dual_hat_fw.bin
```

```
mmdvm_hs_hat_fw-12mhz.bin
```

```
mmdvm_hs_hat_fw.bin
```

```
nanodv_npi_fw.bin
```

```
nanodv_usb_fw.bin
```

```
nano_hotspot_fw.bin
```

```
zumspot_duplex_fw.bin
```

```
zumspot_libre_fw.bin
```

```
zumspot_rpi_fw.bin
```

```
zumspot_usb_fw.bin
```

4. Flash the downloaded firmware:

```
sudo pistar-mmdvmhshatflash-custom modem_fw.bin
```

again, where modem_fw.bin is the bin file that matches your board.

5. Reboot

After rebooting, you optionally can clean up by removing the bin file:

1. Log into Pi-Star via SSH.

2. Enable read-write mode:

```
rpi-rw
```


3. Remove the bin file:

```
modem_fw.bin
```

where modem_fw.bin is the bin file that matches your board.

8b) Updating DVMEGA mounted on an RPi

ALERT! This method does not work with Pi-Star 4.0.0-RC3 or RC4. It likely also does not work with Pi-Star 4.1.x. Basically, this is a method that might work only if you are running a very old version of Pi-Star.

Note: This method, which was figured out by Tom, ON4TOP. Tony Corbett, G0WFFV, discusses it in detail on his Radio Geek blog, [HOW-TO Update DVMega firmware without a programmer or an Arduino](#) 


Tom, ON4TOP, has come to your rescue. He has figured out that it is possible to flash the DVMega firmware using the Rpi itself with the DVMega still in situ by wiring the reset pin of the ATMEGA328 directly to a spare GPIO port and then using (a slightly modified version of) avrdude to flash the hex file straight over /dev/ttyAMA0. The same procedure works just as well on a single band DVMega....

Subsequently, Andy Taylor, MW0MWZ, adapted this method for use with Pi-Star. What follows is a summary of Andy's adaption.

This process works in Pi-Star's default Read-Only mode. These are the instructions for the non-"Japan" firmware update. (Sorry Japanese hams but I don't know the process for the "Japan" firmware update.)

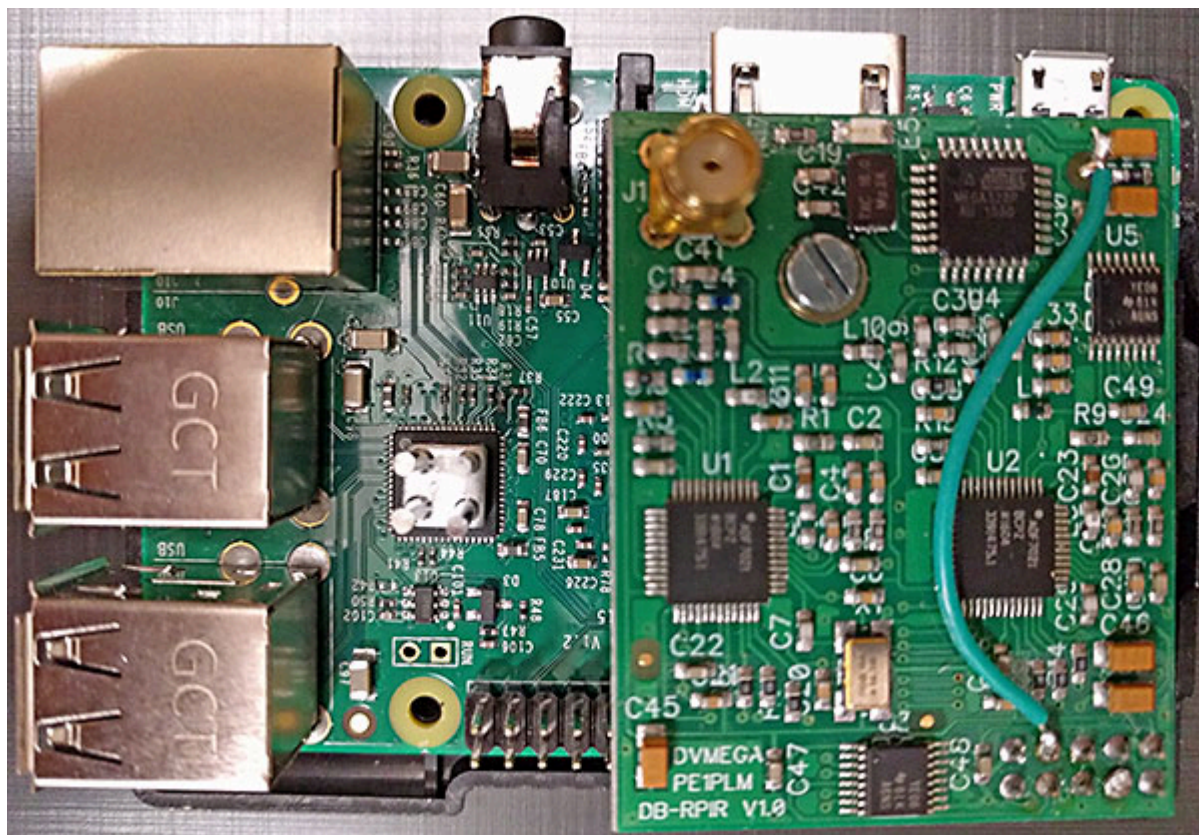
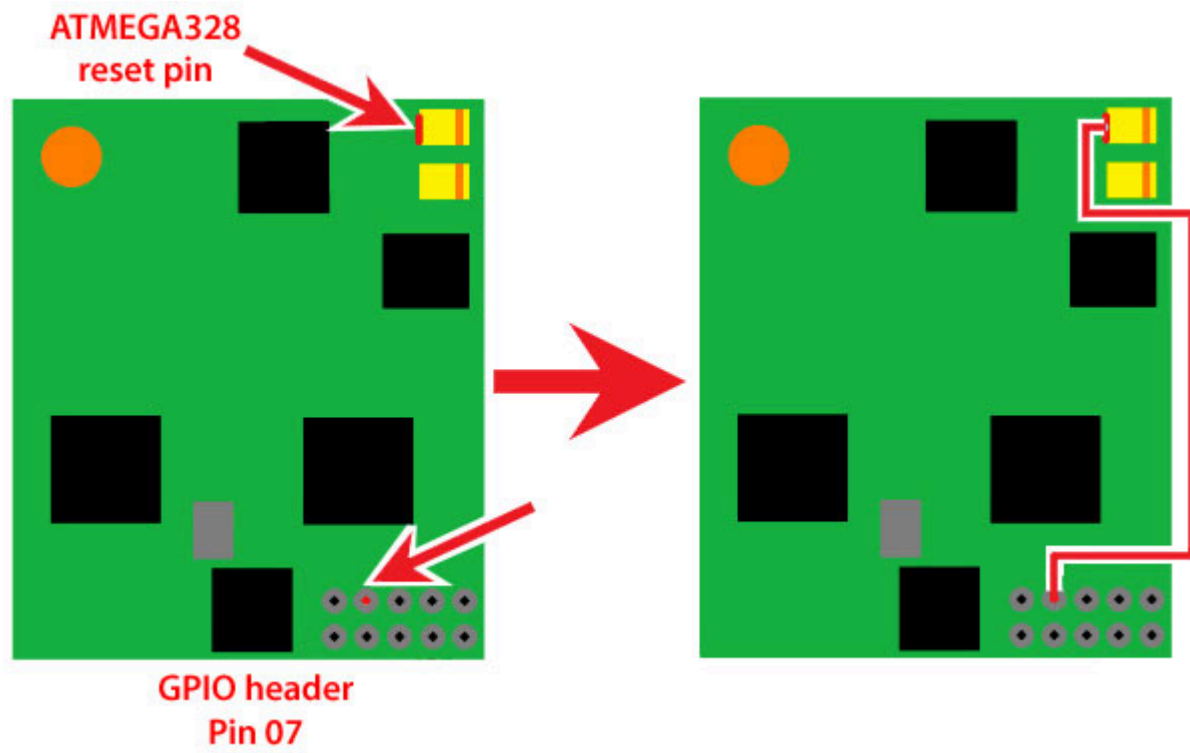
1. For updating the DVMEGA firmware using a Raspberry Pi 3, solder a jumper wire between the ATMEGA328 reset pin and GPIO header Pin 7.

Important! Soldering a DVMEGA and updating its firmware are at your own risk. The only thing that is guaranteed is that if you wreck your DVMEGA doing this, then you own a wrecked DVMEGA.

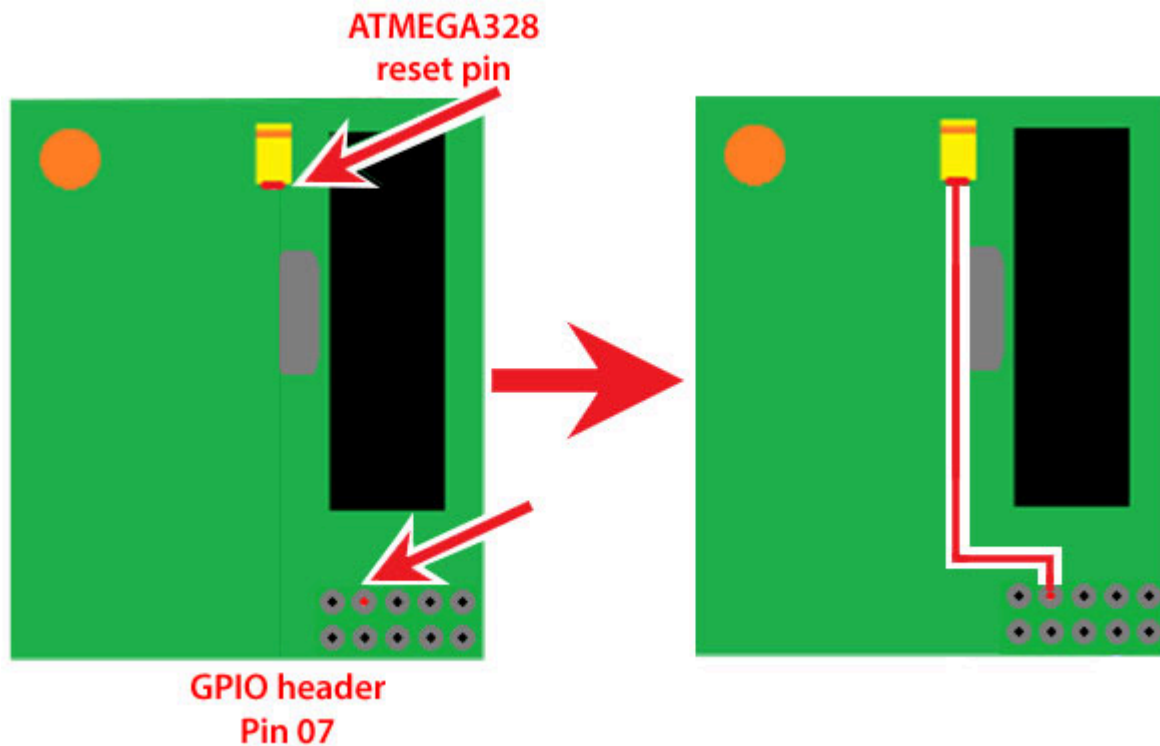
Important! This jumper configuration is for use only when using the Raspberry Pi for updating the firmware. When using the BlueStack board, there is a different configuration and method (see [Updating DVMEGA firmware using a BlueStack board](#) 

Note: You can leave the soldered jumper wire in place after updating the firmware, making subsequent updates much easier.

DVMEGA-DUAL



DVMEGA-SINGLE:



2. SSH into your Pi-Star.
3. Change directory to tmp:
`pi-star@pi-star(ro):~$ cd /tmp`
4. Download the imaging tools:
`pi-star@pi-star(ro):tmp$ wget --tries=3 https://www.pistar.uk/downloads/dvmega-flash-tools.zip`
5. Download the DVMega firmware update file:

- a. To find the latest firmware version, visit the [DVMEGA Downloads](#) page and look in the section for BlueDV-MMDVMhost for Raspberry Pi.

Where ### is the firmware update version:

```
pi-star@pi-star(ro):tmp$ wget --tries=3 http://www.dvmega.nl/wp-content/uploads/2018/11/DVMEGA_RH_V###_UN0.zip
```

Note: I added `--tries=3` after I ran into a problem with the DVMEGA site being unavailable the first time I tried this. The retries kept happening and each one took a long time, so this step seemed to be in an unending loop. I ended up rebooting the device to break the looping. Turns out the default for `wget` (web get) is 20 retries, which seems excessively high to me for this operation.

- b. Unzip both imaging tools and firmware files to the dvmega directory:

```
pi-star@pi-star(ro):tmp$ unzip -d dvmega dvmega-flash-tools.zip
```

- c. Where ### is the firmware update version:

```
pi-star@pi-star(ro):tmp$ unzip -j -d dvmega DVMEGA_RH_V###_UNO.zip
```

Note: The -d switch causes the files to be extracted into the dvmega folder. The -j switch extracts files into the current folder without the folder structure in zip file being recreated.

- d. Change directory to dvmega:

```
pi-star@pi-star(ro):tmp$ cd dvmega/
```

- e. In version 3.26, two hex files are inflated: dvmega/DVMEGA_RH_V326_UNO.hex and dvmega/DVMEGA_RH_V326_UNO_JAPAN.hex. Remove the one you don't need. In my case, I'm installing the non-JAPAN firmware, so:

```
pi-star@pi-star(ro):dvmega$ rm DVMEGA_RH_V326_UNO_JAPAN.hex
```

- f. Run the script:

```
pi-star@pi-star(ro):dvmega$ sudo ./flash_gpio.sh
```

- g. When the script is finished, you'll see:

```
strace: |autoreset: Broken pipe
```

```
Once flashing has been completed, you should reboot your Pi-Star
```

- h. Reboot Pi-Star:

```
pi-star@pi-star(ro):dvmega$ sudo reboot
```

(or you can exit the SSH session and reboot from the Pi-Star dashboard).


[^ Top](#) | [Quick links v](#)

9) Fine tuning to reduce high BER

If you're experiencing high (greater than 1%) Bit Error Rate (BER) with your radio, you can try reducing it by adjusting the RX Offset:

1. Set up Pi-Star with the MMDVMHost Configuration Mode you want to use. Be sure to apply changes.
2. The best way to work on this is to have Pi-Star open in two browsers:
 - a. Set up the first to view the Expert Editor, where you'll do the adjustments:
 - i. Enter Expert Editor: pi-star.local/admin/expert.


- ii. Click MMDVMHost.
 - iii. Scroll down to the Modem section and find the RXOffset setting.
- b. Set up the second to view the dashboard so you can view the resulting BER readings.
3. The easiest thing to do is to make calls lasting a few seconds to a Parrot talkgroup or reflector.
4. In the Expert Editor, adjust the RXOffset setting, make a parrot call, and observe the readings. You're aiming for a BER of less than 1% preferably less the 0.5%.
 - a. Begin with +/-100 offsets, and see how that affects the BER.
 - b. Once you find the lowest BER with the +/-100-step offsets, repeat the process with +/-50-step offsets, and finally +/-10-step offsets until you find the lowest BER.
5. Repeat this process in each MMDVMHost Configuration Mode in which you're experiencing high BER, until you achieve the optimal BER. If you can't find a good balance for the BER between the different modes, it may be better to use different hotspots for each different mode.

Alternative and more exact method to calculate offsets: Daniel, K9NPX, has a good article on using MMDVMCAL to get more exact RXOffset (and TXOffset) settings for a DMR radio: [Hotspot Offset Calibration](#) .

[^ Top](#) | [Quick links](#) 

10) Manually adding WiFi settings to RPi

After flashing the Pi-Star image to microSD card, you can manually add the WiFi settings you want used so that they'll auto-install on first boot up:


1. Create a wpa_supplicant.conf file. I used [NotePad++](#)  for this. The file should contain the following lines and should be saved as a plain txt file with the name wpa_supplicant.conf (add your own country code, and replace WirelessNetworkName and WirelessNetworkPassword with your own network's name and password):

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
ap_scan=1
fast_reauth=1
country=US
network={
  ssid="WirelessNetworkName"
  psk="WirelessNetworkPassword"
```



```
id_str="0"  
priority=100  
}
```

Important: Make sure you set country equal to your own country code as this controls which WiFi channels can be used. This is especially important if your router supports 5 GHz.

Note: In newer versions of Pi-Star, if you use Pi-Star's tools to create the wpa_supplicant.conf file—by entering the wireless network credentials on the Configuration page or by using the [Pi-Star WiFi Builder](#) —you'll get a version of the conf file that comments out the actual credentials and inserts HEX versions of them, which makes them work more seamlessly with certain devices, like iPhones. In that case, it'll look a bit different:

```
network={  
#ssid="WirelessNetworkName"  
ssid=576972656c6573734e6574776f726b4e616d65  
#psk="WirelessNetworkPassword"  
psk=364b8070f605f9222cca4e21173ad266e6a54e1  
495a13d38280c67e78a3d5ff2  
id_str="0"  
priority=100  
}
```

- Optionally, you can add additional WiFi connections in the same way:
 - a. The first WiFi connection should have an ID of 0 and a priority of 100:


```
id_str="0"  
priority=100
```

- b. For each additional WiFi connection, increment the ID by one and decrement the priority by one, for example, for the second WiFi connection:

```
id_str="1"  
priority=99
```

2. Insert into your computer a microSD card that was preloaded with the Pi-Star image (such as the one that comes with a ZUMspot kit) or flash a fresh Pi-Star image to a microSD card.

Note: You can ignore any system messages about needing to format the card that are displayed when you first insert the card or after you finish flashing the image. After flashing the Pi-Star image to the microSD card, eject it, and then reinsert it.

3. Copy the `wpa_supplicant.conf` file to the boot partition of the freshly imaged microSD card, and then eject it again.
4. Insert the microSD card into the RPi. Upon boot-up, the file is moved to:
`/etc/wpa_supplicant/wpa_supplicant.conf`
and becomes the installed wireless network settings.
5. If necessary, you can subsequently change the wireless network settings (add, edit, or delete) via [Wireless Configuration](#)  in Pi-Star's Configuration view, or via WiFi Config in Pi-Star's Expert Editor.

[^ Top](#) | [Quick links](#) 

11) Connecting a Pi-Star hotspot via cell phone

You also can connect a Pi-Star hotspot to a cell phone, wirelessly or via USB tethering. This makes it possible to easily create a mobile hotspot.

Important! Make sure your battery pack or power supply provides enough amperage to your hotspot; otherwise, it may not work or may work erratically. This is always the case with hotspots, but since connecting a hotspot to a cell phone is often done to make the hotspot mobile, and a battery pack is often used in that case, it's worth emphasizing this.

Note: I had to turn off the VPN service that was running on my Android phone before I was able to connect to the Pi-Star dashboard via the phone's hotspot connection, whether the connection was via WiFi or USB tethering.

11a) Connecting via WiFi

1. If you haven't already, set up your phone's hotspot with a name and password. *Important!* Don't use special characters in your phone's WiFi Hotspot network name (SSID). They can prevent connection.
Hint: In general, avoid using special characters in the password that may cause issues, such as: `{ } & ~ ! () ^ " ' ``
2. Add your phone's hotspot name and password to Pi-Star.
 - You can add more than one wireless network connection. If you have multiple wireless network connections, when you boot up Pi-Star, it will scan each one in turn based on its priority until it finds one to connect to:
 - Be patient, each connection attempt takes about 40 seconds.

- The first wireless network connection you add is given an ID of 0 and a priority of 100. For each additional wireless network connection you add, the ID is increased by one and the priority is decreased by one. Thanks to Bob, NOYWB, for his post in the Pi-Star Users Support Group explaining how this works.
- Turn on your phone's portable hotspot.
- Turn on your Pi-Star hotspot and give it time to connect wirelessly to your phone's hotspot.
- In your phone's browser, open the Pi-Star dashboard. On some phones you may be able to use `http://pi-star/` or `http://pi-star.local/` to connect, but on many phones, like my Android device, you must use the Pi-Star hotspot's IP address.

Note 1: I run a VPN on my Android device and have found that I need to disconnect from the VPN in order to connect my Pi-Star hotspot to my Android device's portable hotspot.

Note 2: Some Android devices show a list of connected devices on the Android device's hotspot settings page, and you can tap a listed device to discover its IP address (thanks to Mike, K2GC, for this hint).

- My Android device doesn't have this feature, so I use an Android LAN scanner app, and then use that IP address in my Android browser to connect to the Pi-Star dashboard.
- If your RPi is connected to a monitor and keyboard, you can also run `ifconfig`. In the returned results, look for the IP address for USB 0.

Note 3: Setting up Pi-Star's wireless configuration using a phone and Auto AP is a bit like juggling, but it is possible:

1. Again, on my Android device, I have to start by turning off the VPN.
2. Then I turn on the Pi-Star hotspot, wait 3 minutes until it goes into Pi-Star Auto AP mode, and then select it in my Android device's list of available WiFi networks. Android freaks out a bit about being connected to a network that doesn't have internet connectivity, but I tell it to calm down and carry on.
3. Next, I use the standard Pi-Star Auto AP mobile IP address (192.168.50.1) as the URL in my browser to connect to the Pi-Star dashboard.
4. Once the Pi-Star dashboard opens, I navigate to the Configuration page and add the wireless network credentials in the Wireless Configuration section. Scan for networks does work. After adding each wireless network, I click Save (and connect), but of course, it won't connect, because the Pi-Star hotspot is locked into the private Pi-Star Auto AP network. After I add all the networks I want to, I navigate to the Power and page and reboot.

5. Once the Pi-Star hotspot has reconnected to my Android device's portable hotspot via regular wireless connection, I use the Android LAN scanner app to find its IP address (which will be different from the Pi-Star Auto AP mobile IP address) and use it as the URL in my browser to connect to the Pi-Star dashboard and begin using it.

Note 4: If your RPi is connected to a monitor and keyboard, you can also run `ifconfig`. In the returned results, look for the IP address for USB 0.

11b) Connecting via USB tethering

Note: John, K1WIZ, posted in the Pi-Star Users Support Group introducing the idea of using USB tethering with a hotspot running Pi-Star: "This allows your hotspot to use the USB connection (instead of WiFi) so that your battery stays charged and isn't drained. Also, using USB is far superior to using WiFi to tether to a phone with a data plan—WiFi is half duplex, whereas the 'Ethernet over USB' is full duplex."


1. Plug your Pi-Star hotspot into your phone via USB, and then turn on the Pi-Star hotspot.
2. Different models of phones will behave differently. On my Android phone, I get a notification that the phone is being powered via USB. Once I see that, message, the USB tethering option is enabled in Settings > Network & Internet > Hotspot & tethering, and I can turn it on.
3. In your phone's browser, open the Pi-Star dashboard. On some phones you may be able to use `http://pi-star/` or `http://pi-star.local/` to connect, but on many phones, like my Android phone, you must use the connected hotspot's IP address, which you can find using a LAN scanner app.

Note: Thanks to Carty, KA2Y, for sharing his experiences setting up USB tethering with me, which helped me improve these notes.

[^ Top](#) | [Quick links v](#)

12) Re-syncing system time and changing date format

Forcing RPi system time re-sync

The RPi uses Network Time Protocol (NTP) to synchronize its clock to UTC. In some cases, the time may get out of sync. In that case, you can simply restart the RPi and the NTP daemon should re-synchronize to UTC. If necessary, you can force NTPD to update the date/time. Here is the force process for Pi-Star 4.1.x and newer (thanks to Craig, KC7NGC for [pointing this out in the Pi-Star User Forum](#) 

1. Log into Pi-Star via SSH.

2. `sudo systemctl stop ntp.service`
`sudo ntpd -gq`
`sudo systemctl start ntp.service`

Here is the force process for Pi-Star 3.4.x:

1. Log into Pi-Star via SSH.
2. `sudo systemctl stop ntp.service`
`sudo ntpdate pool.ntp.org`
`sudo systemctl start ntp.service`

Changing the date format

Warning! It's possible that changing the date format can cause problems with running some scripts, for example, a Pi-Star Update. Proceed at your own risk!

There may be times when you want to change the way a date is displayed, for example, if you're using a Nextion screen. This can't be done via Pi-Star Configuration; instead, you must configure the system Locales setting.

1. Log into Pi-Star via SSH.
Note: I tested this a lot of times in various ways. If I used Pi-Star's built-in SSH Access client, it all worked fine. If I used an external SSH client, after rebooting, when I returned to the Pi-Star dashboard in my browser, MMDVMHost would stop running, and I needed to power off the hotspot and then restart it to get things working properly, including the new locale.
2. Enable read-write mode:
`rpi-rw`
3. Open the locales configuration utility
`sudo dpkg-reconfigure locales`
4. When the Configuring locales window opens an introductory message is displayed. (If your screen is small, proceed to the next screen by pressing Enter to select Ok.)
5. A list of locales is displayed. Use the arrow keys to navigate to the locale(s) you want generated for the system. Press the Space Bar to select or deselect a locale, for example, I deselected en_GB.UTF-8 and selected en_US.UTF-8.
6. Once you have selected the locale(s) you want generated for the system, press Tab to select Ok (or Tab again to select Cancel), and then press Enter.

7. A message about default system locales is displayed. (If your screen is small, proceed to the next screen by pressing Enter to select Ok.)
8. A list of your selected locales is displayed. Use the arrow keys to highlight the locale you want to use as the default system locale, press Tab to select Ok (or Tab again to select Cancel), and then press Enter.
9. You'll be returned to the SSH command line. Wait for the system locales to be generated.
10. When the process is complete, reboot Pi-Star:
`sudo reboot`

[^ Top](#) | [Quick links v](#)

13) DMRGateway notes

Setting up and using DMRGateway is significantly more complex than simply setting up a single DMR Master for one of the systems: BrandMeister, DMRPlus (a.k.a., DMR+), and XLX.

Auto-connecting to DMR talkgroups

Per Andy Taylor in the Pi-Star Users Support Group: "DMR Talkgroups - those are controlled by the master (rather than the client). **On DMR+**, you need to use the options line; **on BrandMeister**, you need to sign up for self-care and set them using their nifty app [ed note: the app is named: My hotspots]." For related info, see the following note: [Pi-Star's BrandMeister Manager module](#).

A DMR+ example per Andy Taylor in an answer in the Pi-Star Users Support Group – If you don't want to auto-connect to a talkgroup (when Pi-Star is set up as a hotspot):

```
StartRef=4000;RelinkTime=60;UserLink=1;TS1_1=9;
```

- `StartRef=4000` = sets the startup reflector to unlink.
- `RelinkTime` (optional) = # of mins of idle time before relinking to StartRef.
- `TS1_1=9` = sets time slot 1 talkgroup 1 to talkgroup 9, which in the DMR+ world sets Direct Mode Operation (DMO) mode as un-linkable.

And another example – To set up a DMR+ auto-connect, for example, to 4409:

```
StartRef=4409;RelinkTime=60;UserLink=1;
```

DMR reflectors, talkgroups 6, 8, and 9

On the DMR network, DMR reflectors enable DMR repeaters to be dynamically connected, similar to how reflectors work on D-STAR. When DMRGateway is enabled, XLX reflector traffic uses talkgroup 6, DMR+

reflector traffic uses talkgroup 8, and BrandMeister uses talkgroup 9.

- *Dec 2018* – BrandMeister USA no longer supports the use of reflectors via its network.
- XLX reflectors use prefix 6 = 64001 - 64999. Link using a private call, then talk via a group call to TG 6.
 - To change XLX modules, 64001 = module A, 64002 = B, etc.
 - *As of Aug 2017*: Private calls to 8001 - 8999 can be used to select XLX masters, for example, 68313 = XLX313, 68950 = XLX950.
 - *As of Jan 2018*: Private calls to 8000 - 8999 can be used **with DMRGateway** to link to XLX reflectors, for example, 68313 links to XLX313's default module (and drops any previously linked reflector), 68950 = XLX950 (on its default E module/4005).
 - *As of Jan 2018*: A private call to 64000 will drop the link to an XLX module, but not to the XLX master.
 - Per Andy Taylor in the Pi-Star Users Support Group: "If you need to override the XLX default room/module numbers, make a file, /root/XLXHosts.txt, with only the XLX masters that you need to override in there, using the same format as the file downloaded from the update server. Set the room to whatever you want. This will stay put through the update process." For more about this, see the note: [Method for overriding default XLX rooms](#).
- DMR+ reflectors (use prefix 8) = 84001 - 84999. Link using a private call, then talk via a group call to TG 8.

DMR reflector XLX950

[XLX950](#) is a Multiprotocol Gateway developer platform hosted by DVC, and provides a bridge between DMR and D-STAR. XLX950 talkgroups 4001 - 4026 are bridged to XRF950 modules A - Z:

- XLX950 TG x4001 = XRF950A
- XLX950 TG x4002 = XRF950B
- •••
- XLX950 TG x4026 = XRF950Z


Notes:

- Craig, W1MSG, has a good video about all of this: [DMRGateway](#).
- See also the DMR notes:
 - [Constellation reflectors](#)

- [Dynamic, static, and auto-static talkgroups](#) 

[^ Top](#) | [Quick links](#) 

14) D-STAR routing (QuadNet)

The [QuadNet2 USA IRC Network](#)  team is doing some good work on D-STAR routing, and Pi-Star works well with QuadNet. An excerpt from the QuadNet Mission Statement:

We are an open ircDDB routing network that doesn't require registration. We contend that valid Amateur Radio license is enough to operate on the QuadNet2 USA IRC Network. We do not require a special club or repeater call sign for use on the network. As long as the call sign is valid and being legally used, we fully support your use of dongles, hotspots and repeaters (both homebrew and ICOM) on the QuadNet network. The purpose of QuadNet is to support ROUTING. Routing is an alternate method of connecting to other D-Star users.

An excerpt from the QuadNet [Routing FAQ](#) :

Routing is a powerful method to connect to individuals, repeaters or groups on a D-STAR network. Routing is a connectionless protocol that does not require a Hosts.txt file to connect. The QuadNet network server does all the heavy lifting for you. You just have to program your radio with the routes you want to use. If there is an individual, repeater or STARnet Group on the QuadNet network, you should be able to route there just by knowing the call sign, or in the case of Group Routing, a "subscribe call."

There are three kinds of routing:

1. **Call Sign Routing** is used to make a person to person connection....
2. **Zone Routing** or sometimes called Repeater Routing will route you radio directly to a repeater on the network. It's just like linking to a repeater, but it's easier to do....
3. **Group Routing** means routing to a STARnet Digital Group. A STARnet Group is kind of like a reflector, but it is actually more like a repeater without the RF transceiver. A STARnet Group can have many individual users "subscribed" to it. Anyone subscribed to a STARnet Group will hear all traffic on the Group.

From a post related to D-STAR to DMR cross-mode reflectors by Jeff, VE6DV, on the Pi-Star User Forum:

Another very popular cross mode reflector would be the Quadnet Array. This is one of the most active systems outside of REF001C and REF030C.

On the D-STAR side you can access the Array by going to:

D-STAR Reflectors:

- XRF757A in Atlanta
- XLX735A in New York
- XLX307D in Wyoming
- XLX049D in Northern Ireland

QuadNet Smart Groups

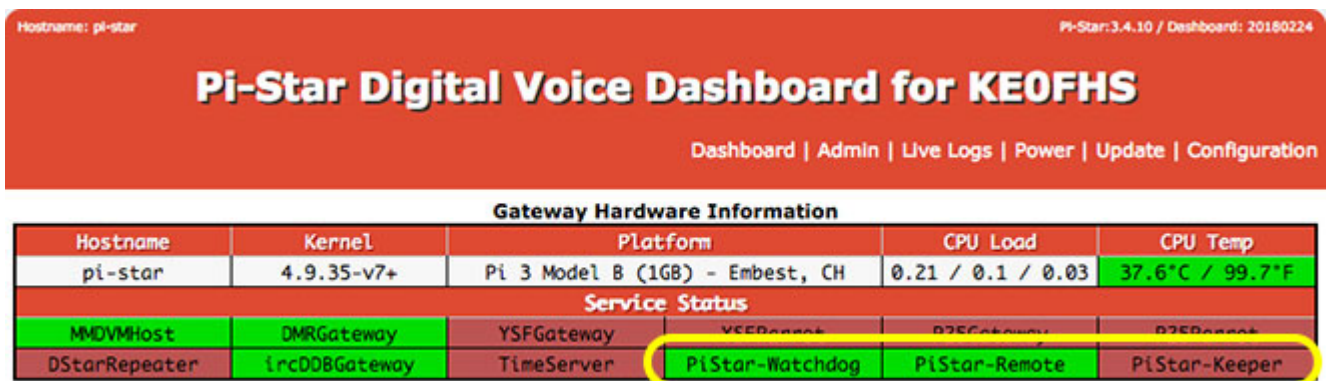
- DSTAR1 located in New York
- DSTAR2 located in San Francisco
- DSTAR3 located in Ohio
- DSTAR4 located in Alabama

On DMR you can use Brandmeister TG 31012. Alternatively you can access the transcoder directly by using XLX307 as your DMR master and then use talkgroup 6.

[^ Top](#) | [Quick links v](#)

15) Watchdogs, Remote RF commands, and Keepers

What do these services do, anyway?



Hostname: pi-star Pi-Star:3.4.10 / Dashboard: 20180224

Pi-Star Digital Voice Dashboard for KE0FHS

[Dashboard](#) | [Admin](#) | [Live Logs](#) | [Power](#) | [Update](#) | [Configuration](#)

Gateway Hardware Information					
Hostname	Kernel	Platform	CPU Load	CPU Temp	
pi-star	4.9.35-v7+	Pi 3 Model B (1GB) - Embest, CH	0.21 / 0.1 / 0.03	37.6°C / 99.7°F	
Service Status					
MMDVMHost	DMRGateway	YSFGateway	YSFRepeater	D2EGateway	D2ERepeater
DStarRepeater	LircDDBGateway	TimeServer	PiStar-Watchdog	PiStar-Remote	PiStar-Keeper

Per Andy Taylor in the Pi-Star Users Support Group:

PiStar-Watchdog – This does what most watchdogs do, it watches; specifically, it watches the services that relate to radio functions and keeps them running. If you stop say MMDVMHost, the watchdog will see that, and will restart it again within 5 mins. This is less handy than it

used to be. I could migrate this functionality to SystemD, but the PiStar-Watchdog has been around for a while, it's an old and trusted friend :)

PiStar-Remote – In the old days, before MMDVMHost was born, ircDDBGateway (D-Star) had the ability to listen for commands via RF; you could command your system to reboot, shutdown, etc., from your radio. Then came MMDVMHost – this feature was missing, and so a need became apparent. PiStar-Remote is my answer to that need. It's far from perfect, but gives you some basic remote control ability via RF, including RF kill switch.

PiStar-Keeper – This service has a VERY specific use... In the UK there is a special class of repeater license that requires the keeper to be present. This service uses bluetooth to watch for the keeper's cell phone, and disables the services when the keeper moves too far away. Like I said, REALLY specific!

Note: Configuration settings for PiStar-Keeper are in the /etc/pistar-keeper file.

Remote Control RF commands

You can access these settings via the Expert Editor's PiStar-Remote tab. Once you have set this feature up the way you want, you can call these codes from your radio to invoke the associated command. For example, to reboot your hospot: if you're using a DMR radio, you can make a **private call** to DMR ID 9999997; if you're using D-STAR, you can make a call to URCALL=REBOOTPI.

Important:

1. Enabled **MUST** be set to true.
2. Callsign **MUST** be set to your CALLSIGN, which **MUST** be UPPERCASE.
3. For the mode(s) you want to use, if you want to enable the shutdown, hostfiles, getip, or 8Ball commands, you **MUST** remove the leading # (comment symbol).
4. *APR 2019* – In previous versions of Pi-Star, the default commands for DMR were in the 899999x range; however, in version 4 of Pi-Star, the default commands for DMR are in the 999999x range. This might conflict a bit with BrandMeister's AirSecurity feature, which uses that range. One consequence of this is that if you use a PiStar-Remote command in the 999999x range for BrandMeister, you might hear a strange announcement, "Secret not found."

[banner]

```
# Pi-Star Remote config file
```

```
# This config file is designed for the Pi-Star Keeper remote control
```

```
# The remote control system is designed to give repeater keepers an
```

RF KillSwitch for their repeaters.

[enable]

Should PiStar-Remote be Enabled? (true|false)

enabled=false

[keeper]

Keepers Information

callsign=YOURCALLSIGN

[d-star]

UR fields

svckill=SVCKILL

svcrestart=SVCRSTRT

reboot=REBOOTPI

#shutdown=SHUTDOWN

getip=GETIP

#hostfiles=HOSTFILE

#8Ball=8BALL

[dmr]

TG commands [*Ed note: Use private calls*]

svckill=9999999

svcrestart=9999998

reboot=9999997

#shutdown=9999996

#hostfiles=9999995

[ysf]

ROOM commands

svckill=99999

svcrestart=99998

reboot=99997

#shutdown=99996

#hostfiles=99995

[p25]

P25 Talkgroups are limited to 1->65535

svckill=65531

svcrestart=65532

reboot=65533

#shutdown=65534

#hostfiles=65530

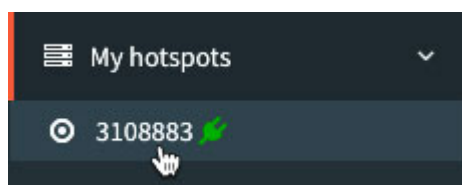
Some notes about PiStar-Remote

- **getip** – As of the 20190127 dashboard update, the GETIP command is working. Be patient, it takes awhile for the text message to be sent to your radio with the IP address.
- **svckill and svcrestart** – Per Andy Taylor in the [Pi-Star User Forum](#), the svckill command is "an RF kill-switch for repeater keepers, there for them to cause the repeater to go offline and stay down." It kills the following: "if you are in MMDVMHost mode, it restarts MMDVMHost and ircDDBGateway, in DStarRepeater mode it restarts DStarRepeater and ircDDBGateway." And according to Andrew, M1DNS, in the same topic, the svcrestart command restarts the services used by Pi-Star without restarting the RPi the way a reboot will.
- **hostfiles** – The hostfiles command triggers a request for a host file update.
- **reboot and shutdown** – When rebooting or shutting down, the behavior is quite different compared to using the software Reboot and Shutdown buttons in the Pi-Star Power view. After keying up the remote RF command, the call is logged in the Gateway Activity and Local Activity sections, but then the Pi-Star dashboard just freezes; it doesn't look like anything more is happening. If you then try to navigate to a different view like Configuration or Dashboard, you'll get a "Site can't be reached" error message, which tells you the command has been executed.
- 8BALL is simply a play on the old Magic 8 Ball toy. Think of a question, transmit, and a text answer is sent back to your phone after a bit. (Apparently, some quantities of beer were involved in the genesis of this feature.)

[^ Top](#) | [Quick links v](#)

16) Pi-Star's BrandMeister Manager module

If you choose to use BrandMeister, you can register for an account on the [BrandMeister Network](#), which gives you access to a variety of "SelfCare" information and settings, including access to settings specific to your hotspot via the "My hotspots" menu.



If you use that feature, you can turn on Pi-Star's BrandMeister Manager module in order to access some of the key hotspot settings from Pi-Star's Admin view, including: add/remove static talkgroups, and drop

calls (QSO) and dynamic talkgroups.

BrandMeister Manager			
Static Talkgroup	Slot	Add / Remove	Action
<input type="text"/>	<input type="radio"/> TS1 <input checked="" type="radio"/> TS2	<input checked="" type="radio"/> Add <input type="radio"/> Delete	<input type="button" value="Modify Static"/>
<input type="button" value="Drop QSO"/> <input type="button" value="Drop All Dynamic"/>			

To more information about enabling the BrandMeister Manager module, see: [BrandMeister User API keys](#) .


Once you have generated the BM API v2 key, in Pi-Star, open the Configuration tab, click the Expert link to open the Expert Editor, click BM API, and enter the key there.

Note 1: BrandMeister views simplex hotspots as having only one time slot: TS0. So for simplex hotspots, the BrandMeister Manager time slot options are ignored, and when you make changes, you'll see the status update message refer to "TS0." For duplex hotspots, time slots 1 and 2 are used.

BrandMeister Manager	
Command Output	
BrandMeister API: Talkgroup 31672 added on slot 0.	

Note 2: As of Pi-Star dashboard version 201901014, the BrandMeister Manager tools support duplex hotspots; the Slot selection now applies to Drop QSO and Drop All Dynamic.

Note 3: If Pi-Star is configured for a repeater, the time slot options are used.

Note 4: For advanced BrandMeister API use, see the article [Using the BrandMeister API by Vladimir, AC2F](#) .

[^ Top](#) | [Quick links v](#)

17) Adding a custom NXDN reflector to Pi-Star

NXDNGateway was added in Pi-Star v3.4.12.

To add a custom NXDN reflector to Pi-Star, for example, 31672 using port 41500, watch the [Pi-Star NXDN Upgrades video](#) by Craig, W1MSG.

For reference, here are the steps:

1. Log into Pi-Star via SSH.
2. Put Pi-Star into read-write mode:

```
pi-star@pi-star(ro):~$ rpi-rw
```

3. Add an NXDNHosts.txt (note capitalization) override file to the root user's home directory (/root):

```
pi-star@pi-star(rw):~$ sudo nano root/NXDNHosts.txt
```

4. Add the following tab-separated entry to NXDNHosts.txt:

```
# Ref URL Port
31672 pi-nxdn.trianglenc.net 41500
```

Write out the changes and exit:

```
Ctrl+o
```

```
Ctrl+x
```

Note: This info, which won't be overwritten by an update, will be appended to the main NXDNHosts.txt file that is stored in the /usr/local/etc folder whenever an update is run.

5. Create custom firewall rules in the ipv4.fw file:

```
pi-star@pi-star(rw):~$ sudo nano root/ipv4.fw
```

Add the following two lines to the ipv4.fw file:

```
iptables -A OUTPUT -p udp --dport 41500 -j ACCEPT
iptables -t mangle -A POSTROUTING -p udp --dport 41500 -j DSCP --set-dscp 46
```

Write out the changes and exit:

```
Ctrl+o
```

```
Ctrl+x
```

6. Update the firewall to add your changes:

```
pi-star@pi-star(rw):~$ sudo pistar-firewall
```

7. Put Pi-Star back into read-only mode:

```
pi-star@pi-star(rw):~$ rpi-ro
```

8. Run an update:

```
pi-star@pi-star(ro):~$ sudo pistar-update
```

9. Exit SSH mode:

```
exit
```

[^ Top](#) | [Quick links v](#)

18) CSS Tool (color party!)

The Expert Editor CSS Tool let's you play around with the color scheme.

(*June 2018:* This is a beta feature. Not all UI colors can be changed, and it doesn't yet apply to mobile view.)

Hostname: pi-star-m Pi-Star:3.4.15 / Dashboard: 20180530

Pi-Star Digital Voice Dashboard for KE0FHS

Dashboard | Admin | Configuration

Modes Enabled		Gateway Activity							
D-Star	DNR	Time (MDT)	Mode	Callsign	Target	Src	Dur(s)	Loss	BER
YSF	P25	09:54:07 May 31st	D-Star	KE0FHS/INFO	CQCQCQ	Net	2.6	0%	0.0%

Note: In this example, I changed Background Banners to the web color royalblue (hex value 4169E1) and the Tables HeadDrop to dark gray (hex value 303030).

If you do create custom CSS settings, it's a good idea to write them down because they aren't included in backups.

The default values:

- Background
 - Page (surrounding blank area) = edf0f5h (bluish white)
 - Content = fffffff (white)
 - Banners (Pi-Star's main color) = dd4b39 (slightly grayish red)
- Text
 - Banners = fffffff (white)
 - BannersDrop (text drop shadow) = 303030 (dark gray)
- Tables
 - HeadDrop (drop shadow) = 8b0000 (dark red)
 - BgEven (lighter table row) = f7f7f7 (off white)
 - BgOdd (darker table row) = d0d0d0 (light gray)
- Content
 - Text = 000000 (black)
- BannerH2 [This is not yet working.]
 - Enabled = 0 (off)
 - Text = Some Text
- BannerExtText [This is not yet working.]
 - Enabled = 0 (off)
 - Text = Some long text entry
- Lookup

Enabled = 0 (off)

Service = RadioID or QRZ

Notes: Service name must be entered exactly as shown. RadioID is the default. You may need to click the Factory Reset button to make the Lookup option display. If you previously entered custom CSS settings, make a note of them so that you can restore them after resetting.

Note: "If you took it all too far and now it makes you feel sick," the CCS Tool's Factory Reset button resets the CSS options back to their original settings. This Factory Reset affects only the CSS options.

[^ Top](#) | [Quick links v](#)

19) RSSI Expert Editor

Per Andy Taylor in the [Pi-Star User Forum](#): "RSSI values depend on the radio—the stock values are there for MMDVM_HS units that support RSSI; if you use anything else, you may well need to tune it."

```
# This file maps the raw RSSI values to dBm values
# to send to the DMR network.
# A number of data points should be entered and
# the software will use those to work out the in-between values.
#
# The format of the file is [tab separated]:
# Raw RSSI Value    dBm Value
#
# RSSI Default Values for MMDVM_HS:
#
# 43    -43
# 53    -53
# 63    -63
# 73    -73
# 83    -83
# 93    -93
# 99    -99
# 105   -105
# 111   -111
# 117   -117
# 123   -123
# 129   -129
```

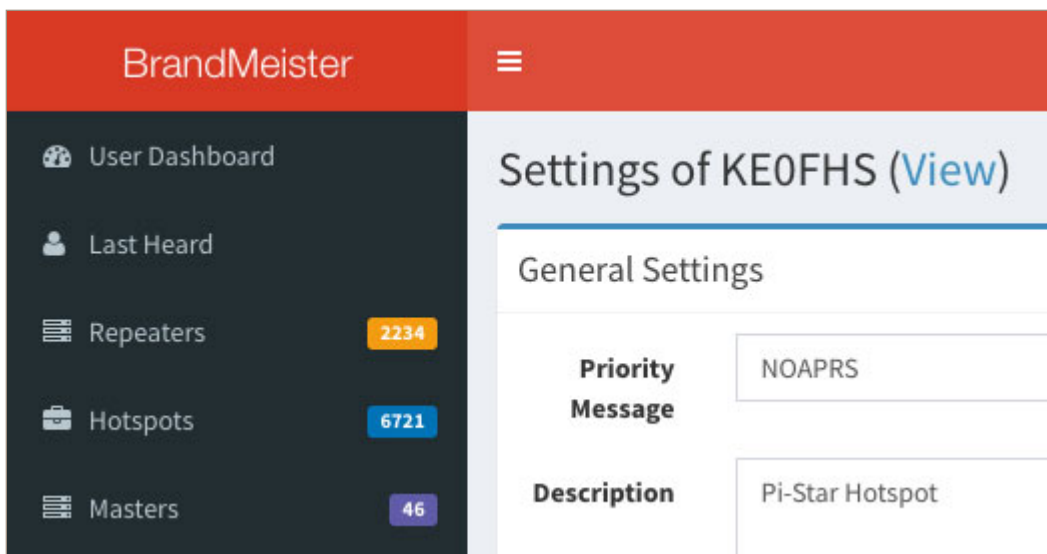

135 -135

141 -141

[^ Top](#) | [Quick links v](#)

20) Prevent sending APRS info via BrandMeister

Thanks to KH6HZ for this hint posted in the Pi-Star User Forum: A sysop of a repeater or a hotspot can prevent APRS reporting by adding the tag NOGATE or NOAPRS to the Priority Message field in the BrandMeister dashboard. For a hotspot, you can find that field in the My hotspots settings.



Note: There's also an APRS-related setting in the Expert Editor > ircDDBgateway Quick Editor: aprsEnabled (0 = disabled; 1 = enabled).

[^ Top](#) | [Quick links v](#)

21) Upgrading to Pi-Star V4

Standalone article: [Upgrading to Pi-Star V4](#) [↗](#)




If you're upgrading an existing hotspot from Pi-Star V3.# or V4.0 to Pi-Star V4.1 or newer: You can't just click the Upgrade link or run `sudo pistar-upgrade`. However, there's a shortcut that makes the upgrade easier.

I've tested this on Mac and Windows 10, and it worked well. Here are the steps:

1. Backup the Pi-Star configuration on your existing hotspot. See [Backing up and restoring Pi-Star](#) [↗](#).

Hint: Be sure to keep the original name of the backup file; changing the name can cause it to be

not unpacked and restored.

2. Download the Pi-Star V4.1 or newer image:
 - Release candidates (RC) are available from [Pi-Star Beta Downloads](#) .
 - Full releases, when available, will be available from [Pi-Star Downloads](#) .
3. Insert a microSD card into your computer and flash a download of the Pi-Star V4.1 or newer image to a microSD card. See [Flashing Pi-Star](#) .
- Note:* If using Windows, dismiss the system Format Disk message.
4. After flashing the Pi-Star image to the microSD card, eject it, and then reinsert it.
Note: If using Windows, once again dismiss the Format Disk message.
5. Copy the backup zip file to the boot partition of the freshly imaged microSD card, and then eject it again.
6. Insert the microSD card into the RPi and power it up. The first boot up will take a bit longer than usual because V4.1 or newer now automatically expands the filesystem (sweet!) and then unpacks and restores your backed up configuration. The good news is that subsequent boot ups are even quicker than they were previously.

Hint (thanks to Larry, KF7L): If the image doesn't connect to your WiFi network and instead goes into Auto AP mode, the issue could be because your router doesn't broadcast SSIDs. In that case, you might want to try adding:

```
scan_ssid=1
```

to the `/etc/wpa_supplicant/wpa_supplicant.conf` file, between `ssid` and `psk`:

```
network={
    #ssid="WirelessNetworkName"
    ssid=576972656c ...
    scan_ssid=1
    #psk="WirelessNetworkPassword"
    psk=364b8070f60 ...
    id_str=""
    priority=100
```

7. Log into Pi-Star in order to open the Configuration page.
 - *User Name* – This your hotspot's hostname, either the default, pi-star, or whatever you changed it to in your backed up configuration.

- *Password* – Even if you used your own Remote Access Password in the configuration you backed up, for security reasons, the Pi-Star backup does not back up this password, so you must use the default password to log in: raspberry.
8. It's a good idea to double-check your configuration settings. Some settings are not backed up, so if you changed them to something other than the default, you'll find that they are back on the default settings of a new install, for example:
- *General Configuration*:
 - System Time Zone (default Europe/London)
 - Dashboard language (default English_uk)
 - *D-Star Configuration* – Time Announcements (default: On) and Use DPlus for XRF (default: Off). If you change the Use DPlus for XRF setting, you must run a Pi-Star Update to put the new setting into effect.
 - *Firewall Configuration* – Auto AP (default: On)
 - *Remote Access Password* – Since Pi-Star backup does not back up this password, if you want to use your own password, **which is a good idea**, you'll need to enter your own password.
9. It's also a good idea to make a backup of your new Pi-Star V4.1 or newer configuration. For more info, see [Backing up and restoring Pi-Star](#) ↗.

[^ Top](#) | [Quick links](#) v

22) Linking to reflectors from the command line

There is a script for linking to and unlinking from reflectors from the command line, for example, in an SSH session.

- To link to a reflector, for example, REF001 C:
`sudo pistar-link ref001_c`
- To unlink from a reflector:
`sudo pistar-link unlink`
- To view other commands:
`sudo pistar-link`

[^ Top](#) | [Quick links](#) v | [Full TOC](#)

23) A few Raspberry Pi OS commands

- [Download Raspberry Pi OS](#) from Raspberry Pi Foundation.
- [Installation instructions](#).
- To download updates to installed components and perform an upgrade to install them, in Terminal:

Code:

```
sudo apt-get update
```

Code:

```
sudo apt-get upgrade
```

- To change the default password, use the Change User Password setup option in the Raspberry Pi Configuration application ([instructions](#)):

Code:

```
sudo raspi-config
```

- To create a separate root password, run:

```
sudo passwd root
```

- Change the date and time:

Code:

```
sudo raspi-config
```

Code:

```
Internationalization options
```

- Check CPU temperature:

Code:

```
vcgencmd measure_temp
```

Note: There's also a panel applet called CPU Temperature Monitor. It puts a little box in the task bar that shows the current temp as well as little graph at the bottom that shows it over time. To add it, right-click the task bar to access Add / Remove Panel Items.

- Customize the clock panel applet in the task bar:

As with most of the panel applets, you can right-click the clock to access the customization options. It's possible to specify how you want the day, date, and time to appear, as well as to add other info like the time zone and the offset from UTC. For example, mine displays:

```
Mon Sep 16, 2019 8:39 PM MST -06:00
```

- View a weather snapshot:

There's also a panel applet for the weather. After you select your city, it will display the local temperature and a little graphic that indicates conditions (sunny, cloudy, etc.).

24) Pi-Star fork by F1RMB

If you want to experiment, check out the Pi-Star fork being developed by Daniel, F1RMB: [GitHub: f1rmb/Pi-Star_DV_Dash](#). Read the lower half of the page.



Hostname: pi-star-2 Pi-Star:4.1.0-RC10 / Dashboard: 20200218-f1rmb

Pi-Star Digital Voice Dashboard for KE0FH5

Dashboard Admin Live Logs Sysinfo Power Update Configuration

Gateway Hardware Information

Hostname	Kernel	Platform	CPU Load		CPU Temp	
pi-star-2	4.19.97-v7l+	Pi 4 Model B (1GB) - Sony, UK	1m:0	5m:0.02	15m:0	39.4°C/102.9°F

Service Status

MMDVMHost	DMRGateway	YSFGateway	YSFParrot	P25Gateway	P25Parrot	DAPNETGateway
DStarRepeater	ircDDBGateway	TimeServer	PiStar-Watchdog	PiStar-Remote	PiStar-Keeper	MobileGPS

A few notes about the F1RMB fork:

- It is an experimental fork and is **NOT** an official version of Pi-Star, and you **CANNOT** get support for it on the official Pi-Star forum. That said, I find that it works very well, and it is updated often with new features and fixes.
- It supports the Open GD77 hotspot very well.
- It incorporates some nice enhancements and fixes.
- It's easy to install. See the notes on the lower half of the main GitHub page: [f1rmb / Pi-Star_DV_Dash](#).
- You can switch back and forth between the F1RMB fork and the official version of Pi-Star using the script Daniel provides.

PROCEED AT YOUR OWN RISK:

1. **Before getting started, make sure the official version of Pi-Star is fully updated and upgraded.** Do this via SSH to get all of the OS updates as well (you won't get these when updating within the dashboard itself). I've found with my RPi 3A+ and 4B hotspots running the official beta 4.1.x branch that updating fully to the most recent version can take persistence and outright stubbornness. First I run an update, then an upgrade. Then I power off and back on and run the update process via SSH again. I keep doing this until the update runs and finishes normally with Pi-Star in the regular read-only mode. Sometimes, the update process appears to get stuck. When that happens, I let it sit for a long time, then power off and back on and run the update via SSH again. Other times, some directories are locked and other errors happen. I just keep powering off and back on and rerunning the update via SSH until it runs flawlessly and I end up with Pi-Star in the regular read-only mode. That's when I know I'm golden.

2. After updating and upgrading successfully, make a backup of the official Pi-Star configuration before you start this experiment!

3. Log into Pi-Star via SSH.

4. Run the following commands to be able to switch between the official version and the fork:

```
sudo su
rpi-rw
cd
wget http://tinyurl.com/f1rmb-pistar-ng -O f1rmb-pistar
chmod +x f1rmb-pistar
```

5. Then, to switch to the F1RMB fork:

```
./f1rmb-pistar -ia
```

After switching to the F1RMB fork:

- a. When switching, the callsign displayed by the dashboard gets reset to the default M1ABC. This is normal. To set it to your own callsign, go to the Configuration page and click an Apply Changes button.
- b. You also may need to reset a couple other configuration options back to what you normally use, at the very least, General Configuration > Dashboard Language. After making any other changes, click Apply Changes again.
- c. Run Pi-Star Update once.
- d. Run Pi-Star Upgrade as many times as necessary to get to the latest version.
- e. If you were using custom CSS colors in the official Pi-Star dashboard, you may need to go into Expert Editor > Tools > CSS Tool to reset the colors back to the defaults for the F1RMB fork. Once you've done that, you can easily experiment with colors because the F1RMB fork's version of the CSS Tool includes a really nice color picker.

6. To switch back to the official Pi-Star dashboard:

```
./f1rmb-pistar -ra
```

After switching back to the official version of Pi-Star:

- a. When switching, the callsign displayed by the dashboard gets reset to the default M1ABC. This is normal. To set it to your own callsign, go to the Configuration page and click an Apply Changes button.
- b. You also may need to reset a couple other configuration options back to what you normally use, at the very least, General Configuration > Dashboard Language. After making any other changes, click Apply Changes again.

25) DMRGateway rewrite rules

Andy Taylor, MW0MWZ, has done some interesting work on the DMR rewrite rules, posting an example and explanation in the Pi-Star User Forum: [New Overlay for DMRGateway](#) [🔗](#). Here are a few extracts from that topic:

General concept: The idea here is to be able to use TS1 or TS2 on the repeater, you can tweak this to your desires with some caveats:

- Any Private call, or Group call that is not specifically routed, will go out via BrandMeister. So GPS updates, private contacts, SMS etc will go that way.
- BrandMeister works as normal without any changes, they have the most TGs and this causes the least frustration to do it this way.
- DMR+ this has had quite some revisions, steering the output Group calls to specific TimeSlots to make them work properly.
- TGIF Talkgroups are all extended to 7 digits total, with a "4" prefix, and can be used on either time slot. (No conflict with BM or DMR+).
- And there is still provision for DMR Network 5 to link to whatever custom thing you want too.

Andy's example from the topic:

```
[DMR Network 1]
Enabled=1
Address=87.117.229.173
Port=62031
TGRewrite0=2,9,2,9,1
TypeRewrite1=1,9990,1,9990
TypeRewrite2=2,9990,2,9990
SrcRewrite0=2,4000,2,9,1001
PassAllPC1=1
PassAllPC2=2
PassAllTG1=1
PassAllTG2=2
Password="passw0rd"
```

Debug=0

Name=BM_United_Kingdom_2341

Id=123456701

[DMR Network 2]

Enabled=1

Address=168.235.109.210

Port=55555

TGRewrite0=2,8,2,9,1

TGRewrite1=2,80505,2,505,1

TGRewrite2=2,80800,2,800,100

TGRewrite3=2,83801,2,3801,8

TGRewrite4=2,89990,2,9990,1

TGRewrite5=2,80001,1,1,9999

TGRewrite6=2,80001,2,1,9999

PCRewrite0=2,84000,2,4000,1001

Password="PASSWORD"

Debug=0

Id=123456701

Name=DMR+_IPSC2-QUADNET

Options="Userlink=1;"

[DMR Network 4]

Enabled=1

Name=TGIF_Network

PCRewrite1=1,4009990,1,9990,1

PCRewrite2=2,4009990,2,9990,1

TypeRewrite1=1,4009990,1,9990

TypeRewrite2=2,4009990,2,9990

TGRewrite1=1,4000001,1,1,999999

TGRewrite2=2,4000001,2,1,999999

SrcRewrite1=1,9990,1,4009990,1

SrcRewrite2=2,9990,2,4009990,1

SrcRewrite3=1,1,1,4000001,999999

SrcRewrite4=2,1,2,4000001,999999

Address=tgif.network

Password="passw0rd"

Port=62031

Location=0

Debug=0

Id=123456701

For more information about some of the individual rewrite rules, see [this specific post](#) in the topic.

If you use an AnyTone radio, you also may be interested in this post: [DMRGateway Overlay - TalkGroup List Generator for AnyTone](#)

See also the helpful downloads made available by Andy on his Pi-Star site: [AnyTone DMR Downloads](#). Includes the following full lists: DMR User, BrandMeister TalkGroup, DMR+ TalkGroup, TGIF TalkGroup, and UK Repeater. The files are generated when you download them, so they're always up to date.

[^ Top](#) | [Quick links v](#)

[< Pi-Star cross modes · Troubleshooting >](#)

Quick links to this page's content

[^ Top](#)

Links to helpful resources

- a. [Pi-Star: sites](#)
- b. [Pi-Star: forums & groups](#)
- c. [Pi-Star: articles](#)
- d. [Pi-Star: presentations](#)
- e. [Pi-Star: videos](#)
- f. [Pi-Star: hardware and software](#)
- g. [Pi-Star: tools](#)
- h. [Pi-Star: related GitHub repositories](#)

A few other notes

1. SSH Access to Pi-Star
2. Port forwarding
3. Radio/modem types supported
4. Entering read-write mode
5. Running commands as the root user
6. Expanding the filesystem
7. Manually adding or overriding repeaters and reflectors
 - a. Method for overriding default XLX rooms
8. Updating hotspot firmware via Pi-Star
 - a. Updating ZUMspot/MMDVM_HS, etc.
 - b. Updating DVMEGA mounted on an RPi
9. Fine tuning to reduce high BER
10. Manually adding WiFi settings to RPi
11. Connecting a Pi-Star hotspot via cell phone
 - a. Connecting via WiFi
 - b. Connecting via USB tethering
12. Re-syncing system time and changing date format
13. DMRGateway notes
14. D-STAR routing
15. Watchdogs, Remote RF commands, and Keepers
16. Pi-Star's BrandMeister Manager module
17. Adding a custom NXDN reflector to Pi-Star
18. CSS Tool (color party)
19. RSSI Expert Editor
20. Prevent sending APRS info via BrandMeister
21. Upgrading to Pi-Star V4
22. Linking to reflectors from the command line
23. A few Raspberry Pi OS commands

24. [Pi-Star fork by F1RMB](#)

25. [DMRGateway rewrite rules](#)

[Playing with Pi-Star](#)

[Pi-Star cross modes](#)

[Pi-Star troubleshooting](#)

[^ Top](#)

[< Pi-Star cross modes · Troubleshooting >](#)

Amateur radio notes by Toshen, KE0FHS:

[CQ](#) · [Base](#) · [D-STAR](#) · [DMR](#) · [Hotspots](#) · [Pi-Star](#)

Comments or feedback?

[Please let me know](#) [↗](#)



This work by [KE0FHS](#) [↗](#)

is licensed under a

[Creative Commons Attribution-ShareAlike 4.0 International License](#) [↗](#)