

MMDVM-Pi rev. 1.0 board

The MMDVM-Pi hardware radio interface is part of the Multi-Mode Digital Voice Modem open source project. It combines the power of an ARM processor and analog radio interface to create a modem to handle all amateur digital voice modes.

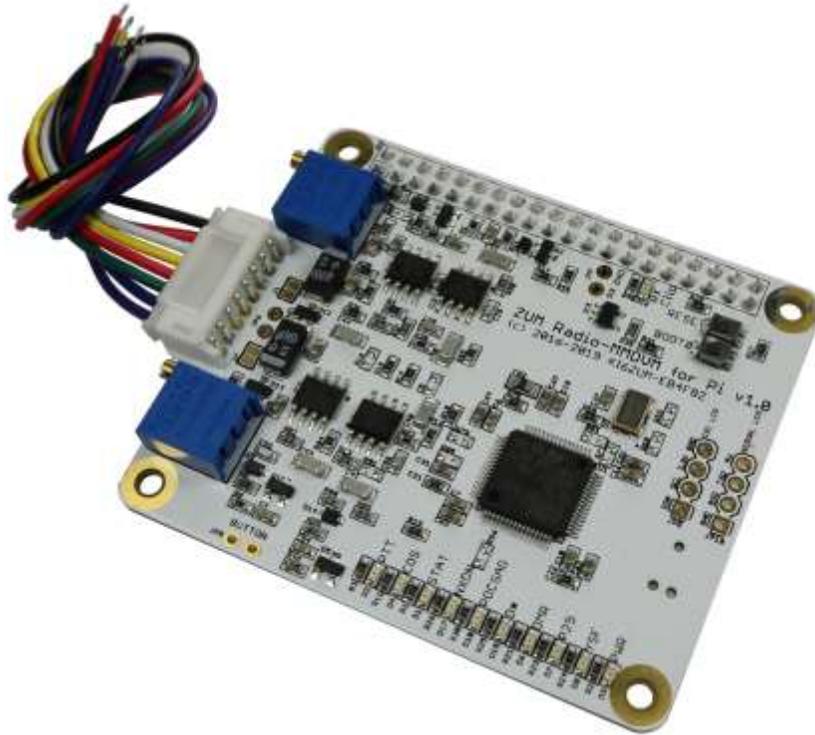


Table of Contents

Specifications	3
Wiring connections	3
Adjusting signal levels (hardware)	4
Adding an external display.....	6
Configuring Pi-Star (including adjusting signal levels via software)	8
Building MMDVM-Pi firmware.....	11
On Windows 10.....	11
On Ubuntu.....	12
On macOS.....	12
On Pi-Star	13
Flashing MMDVM-Pi	15
Windows	15
Linux and macOS.....	18
Pi-Star.....	19
Find IP Address.....	20
Info for use under Windows 10	21
Support	21

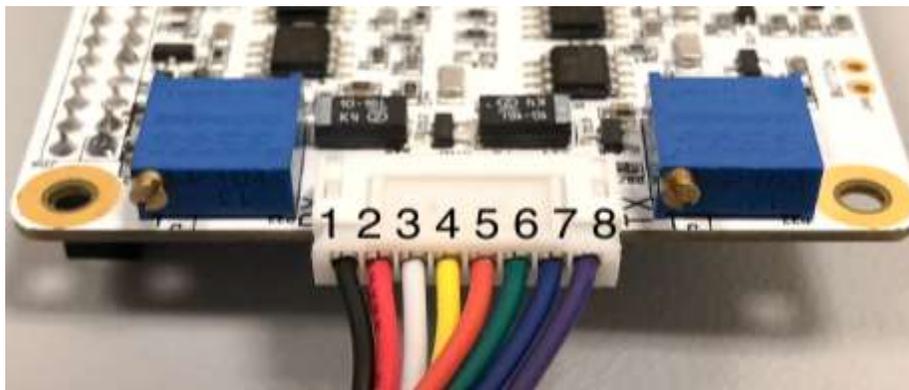
Specifications

The MMDVM-Pi board is designed for repeater and high power hotspot applications – connects to user supplied Raspberry Pi board and radio(s)

- Utilizes a high performance 32bit ARM processor (STM32F722) running at 180Mhz (Room for future upgrades)
- Fifth generation analog filter design that provides an extremely low BER compared to previous generations
- 2 Multi-turn pots for fine RX and TX adjustments
- Onboard LEDs to show status and modes (PTT, COS, Power, D-Star, DMR, P25, Fusion, NXDN and POCSAG)
- Onboard LED to show receive signal level clipping
- Connection for OLED and Nextion LCD screens
- 8 pin JST Connector with pigtail wires
- Open source MMDVM firmware preloaded and easily upgraded by software

Wiring connections

Here is a picture of the 8 pin header with cable showing wire colors:



Here is a table of the pin numbers, names and wire colors:

Pin number	Signal name	Description	Wire color
1	CTRL	Control (output)	Black
2	COS/STAT1	Carrier sense (input)	Red
3	RX audio	Receive audio from radio (input)	White
4	Ground	Signal ground	Yellow
5	Ground	Signal ground	Orange
6	TX audio	Transmit audio to radio (output)	Green
7	PTT	Push to talk (enable transmit) (input)	Blue
8	RSSI	Received signal strength indicator (input)	Purple

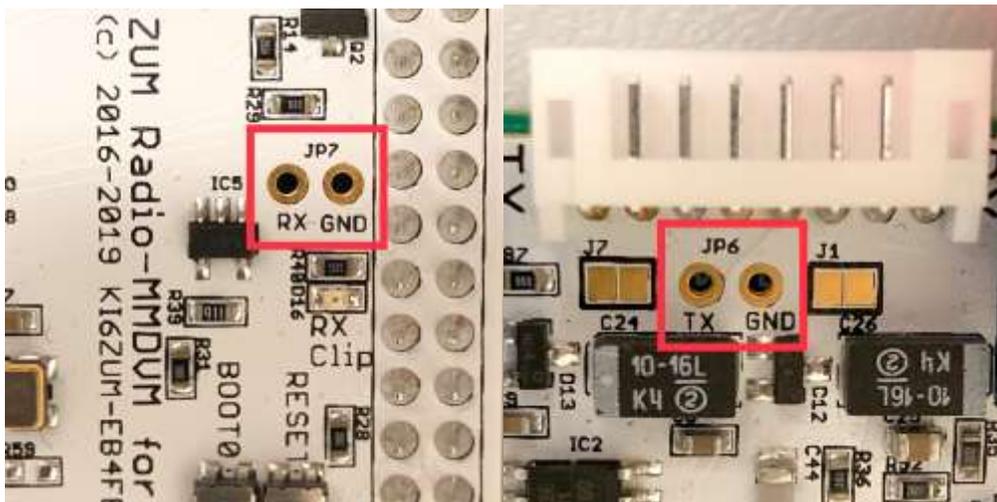
Here's an example radio's (FT-7900) pin connection via the mini DIN 6 connector on the rear of the radio:



Pin number	Signal name	Wire color	Mini DIN 6 connector
3	RX audio	White	9600 from radio
4	Signal ground	Yellow	GND
6	TX audio	Green	Audio to radio
7	PTT	Blue	PTT

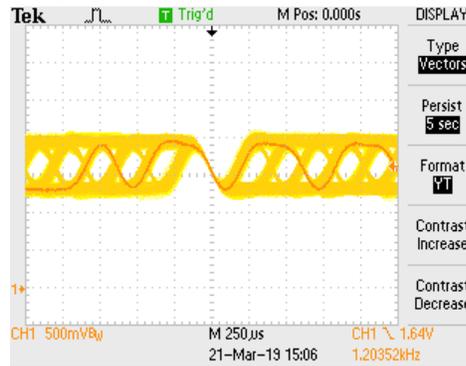
Adjusting signal levels (hardware)

Here is a picture of the test points on the PCB that can be used to look at the RX and TX signal levels on an oscilloscope. The signal on JP7 is the RX signal audio on the ADC input pin of the STM32 chip. JP6 has the TX signal audio on the pin going to the radio.

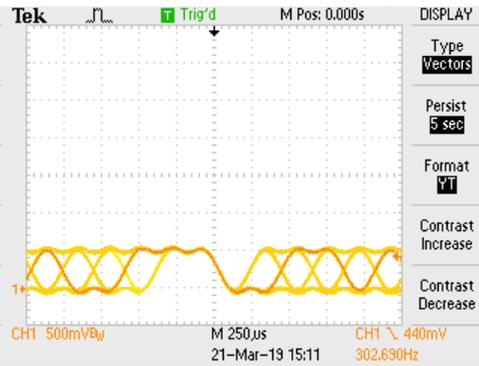


Here are some signal images captured on an Oscilloscope

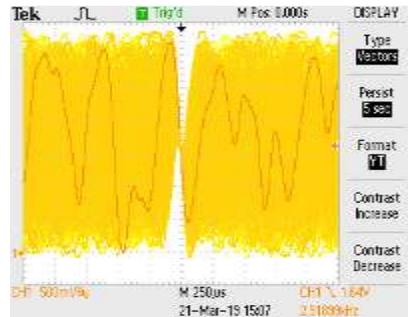
RX-(JP7)



TX-(JP6)



Noise without signal (JP7)



JUMPER DESCRIPTION TABLE

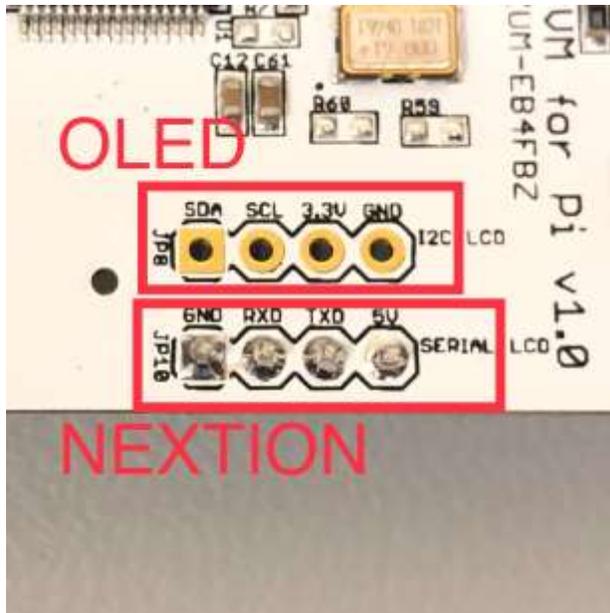
Bypass Pad	ON	OFF	Description
J1	X		Disable DC blocking capacitor on Radio RX end
J1		X	Enable DC blocking capacitor on Radio RX end
J2	X		Bypass 10K variable resistor
J2		X	Enable 10K variable resistor
J7	X		Disable DC blocking capacitor on Radio TX end
J7		X	Enable DC blocking capacitor on Radio TX end

LED Description Table

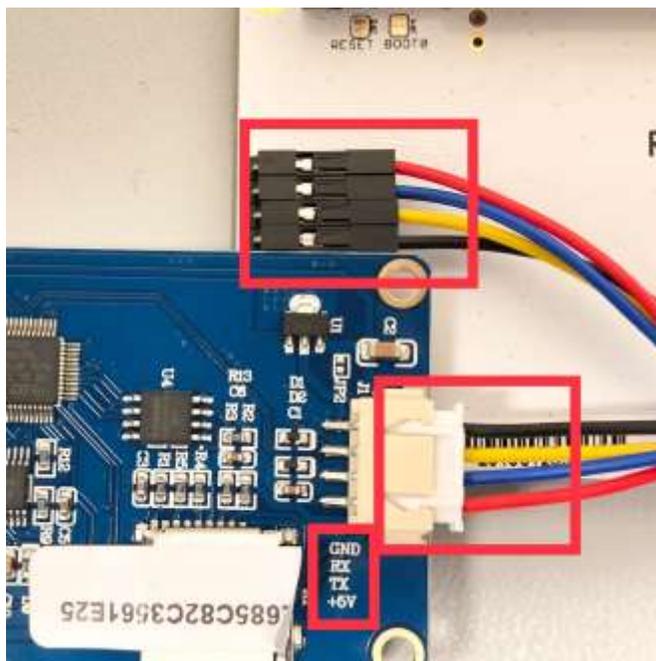
	LED NAME	Description
	PTT	Board is transmitting
	COS	Board is receiving
	STAT	Status
	NXDN	NXDN Enabled
	POCSAG	POCSAG Enabled
	D*	D-Star Enabled
	DMR	DMR Enabled
	P25	P25 Enabled
	YSF	YSF Enabled
PWR	Board is receiving power	

Adding an external display

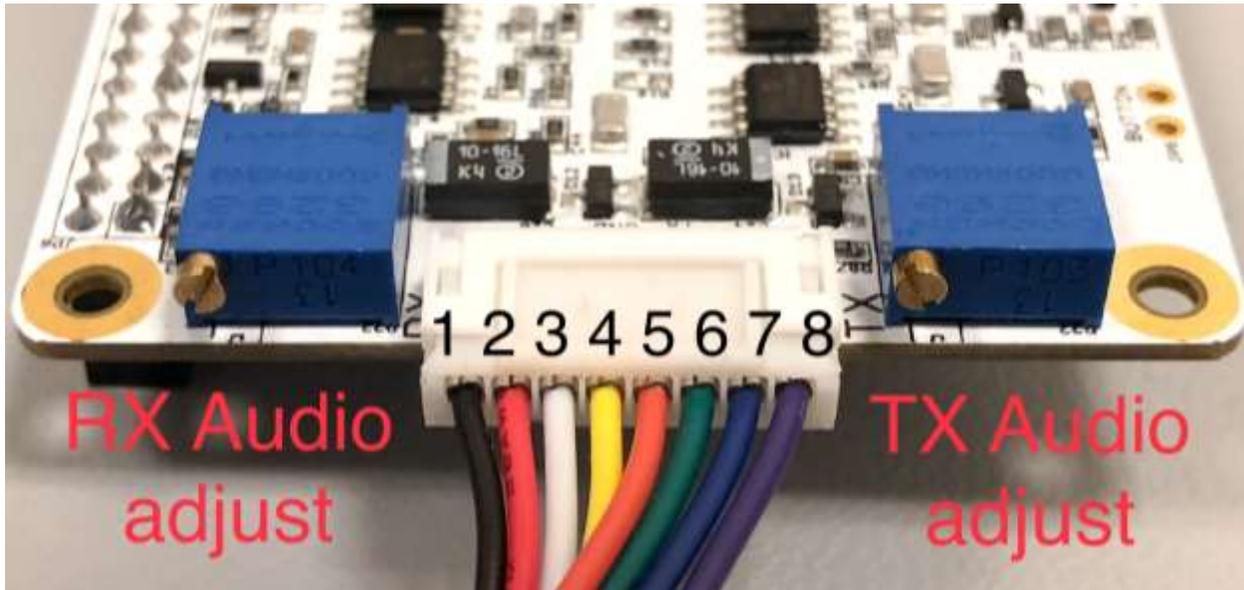
JP8 is where you can solder a header for connecting an OLED display. JP10 is where you can solder a header for connecting a NEXTION display.



For connecting a NEXTION display, make sure that each colored wire matches the connections shown in the image below.



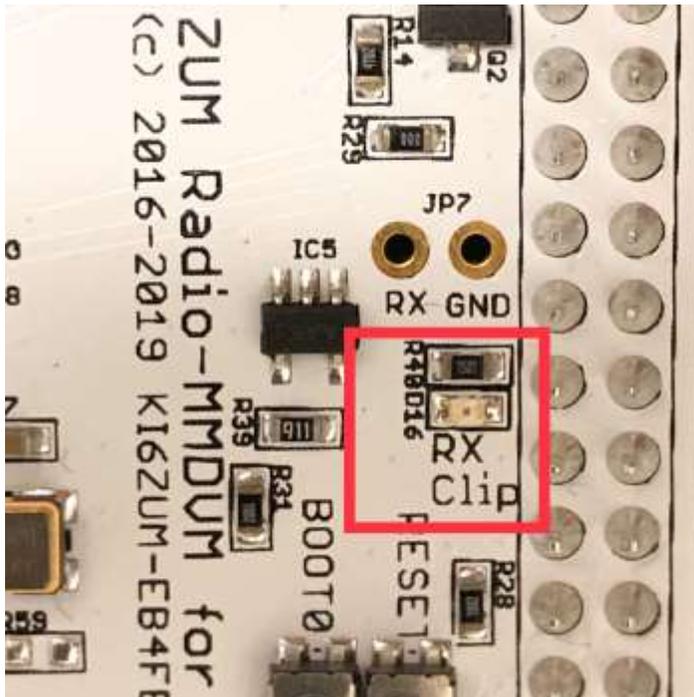
Here is a picture of the two trim pots used to adjust the signal levels of the RX and TX audio



Turning the RX audio trim pot counter-clockwise will increase the signal level, while turning the trim pot clockwise will decrease the signal level.

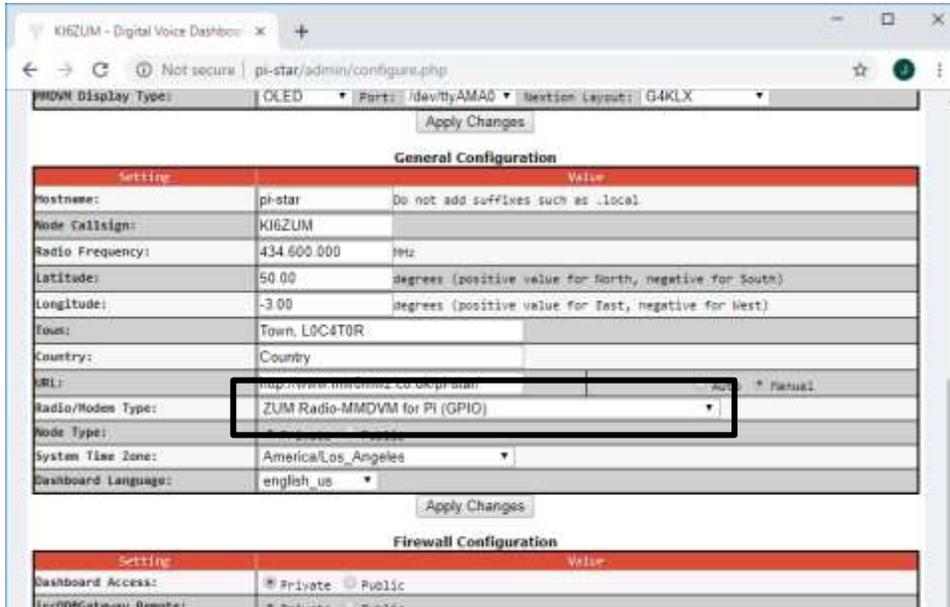
Turning the TX audio trim pot counter clockwise will decrease the signal level, while turning the trim pot clockwise will increase the signal level.

When there is no audio stream coming in, adjust the RX pot until the RX CLIP LED barely lights up.

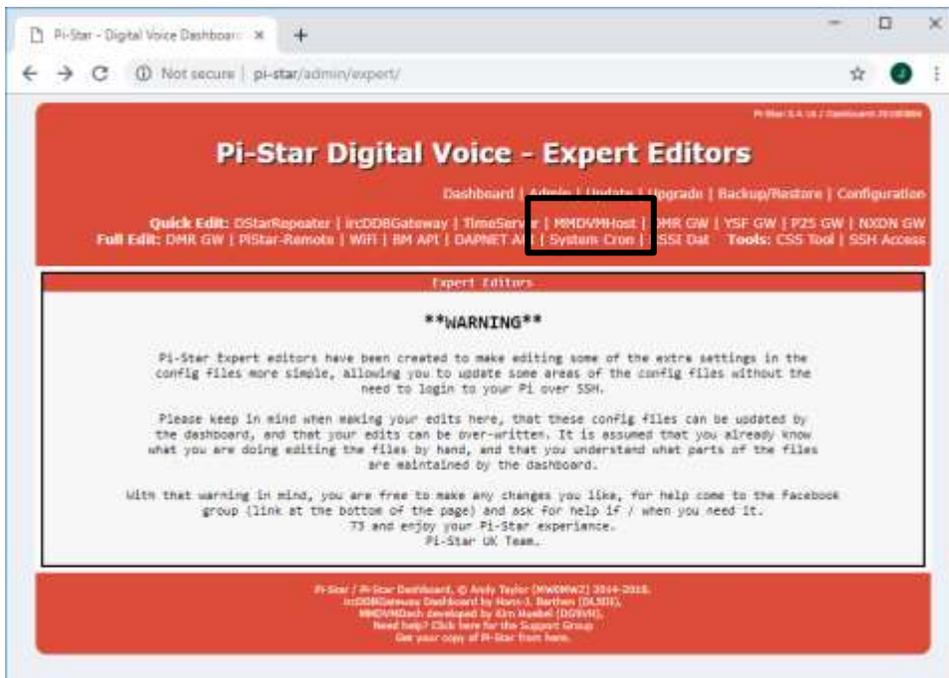


Configuring Pi-Star (including adjusting signal levels via software)

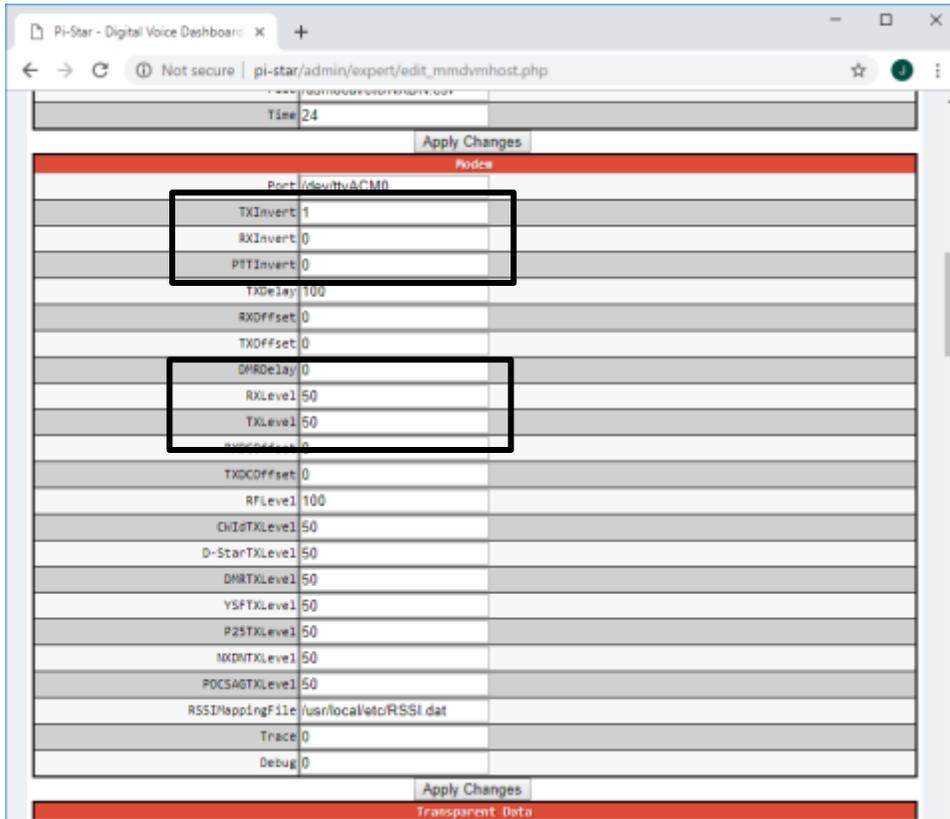
From the “Configuration” menu, set the Radio/Modem type to “ZUM Radio-MMDVM for Pi (GPIO)” and then select Apply Changes.



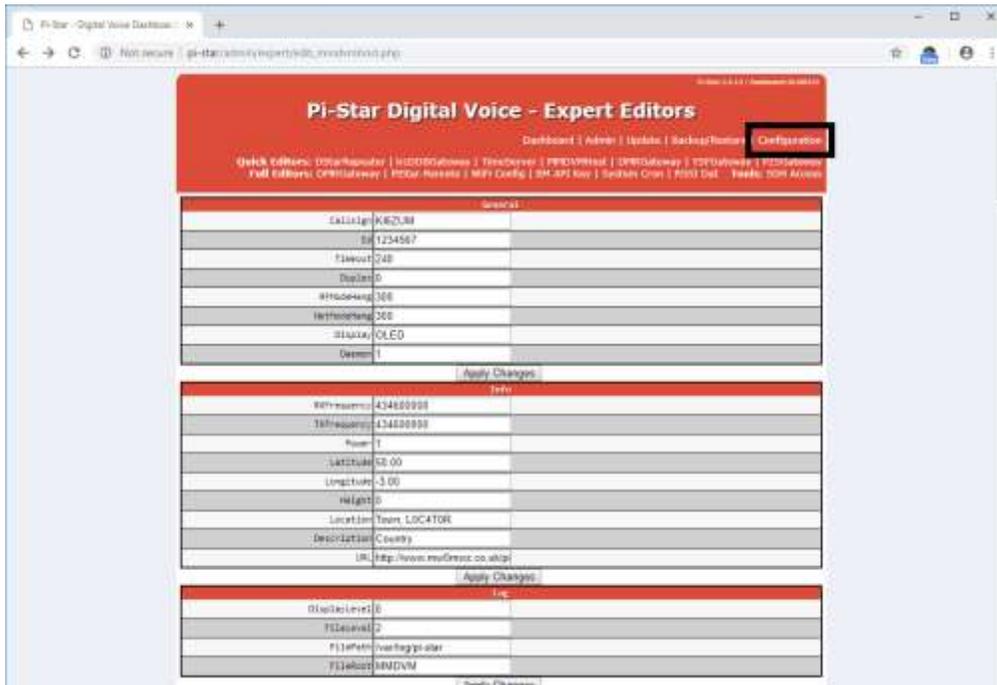
Next select the “expert” menu, and then select “MMDVMHost”



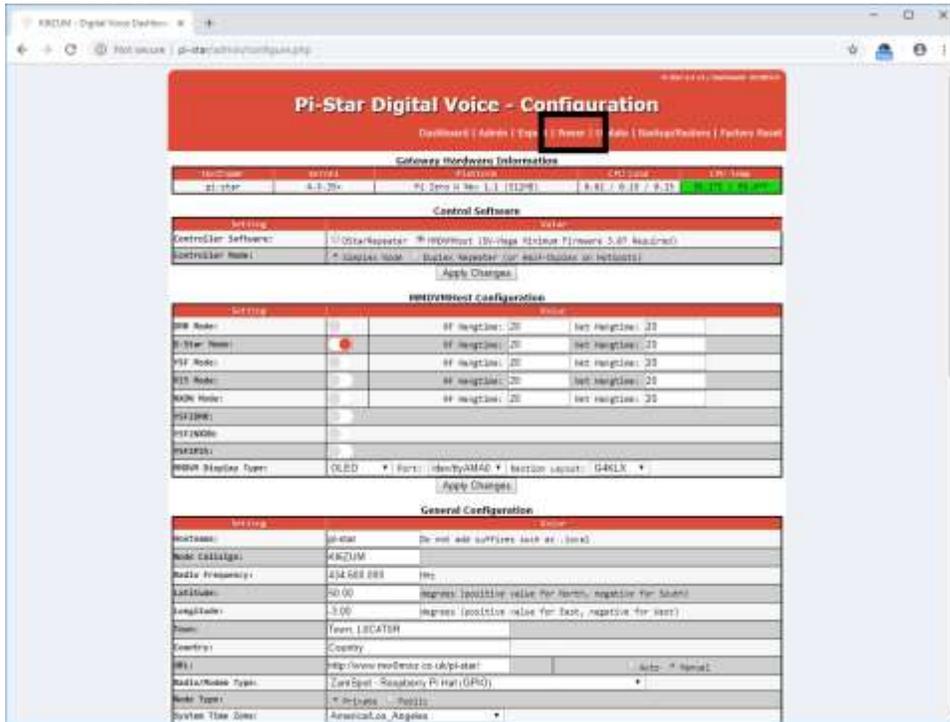
Scroll down to the “Modem” section to access the invert and level settings then select “Apply Changes”



Scroll back up to the top of the page and select “Configuration”.



Next, select "Power"



Next, select "Reboot"



Building MMDVM-Pi firmware

On Windows 10

- Install bash using these instructions: [bash-windows-10](#)
 - Once the components installed on your computer, click the Restart now button to complete the task.
- Install Ubuntu Linux
 - go to the Microsoft app store - <https://aka.ms/wslstore>
 - Click on Ubuntu
 - Click on Get
 - after it installs, click on Launch
 - now enter a new username for linux
 - enter a password for the new linux user
 - enter the password again
- Once bash and Linux are installed, install GCC for ARM. At the bash command prompt, type in the following instructions.
 - `sudo apt-get update`
 - `sudo apt-get upgrade`
 - `sudo apt-get install gcc`
 - `sudo apt-get install make`
 - `sudo apt-get install git gcc-arm-none-eabi libstdc++-arm-none-eabi-newlib autoconf libtool pkg-config libusb-1.0-0 libusb-1.0-0-dev`
- Get the latest source code from GitHub:
 - `git clone https://github.com/q4klx/MMDVM`
 - `cd MMDVM`
 - `git submodule init`
 - `git submodule update`
- Edit Config.h. Uncomment the line: `#define MODE_LEDS`
 - If you want to connect a Nextion display, then also uncomment the line:
`#define SERIAL_REPEATER`
- To start build run: `make pi-f722`
- Binaries will be under the `bin/` folder

On Ubuntu

- Follow the same instructions as **Windows 10** but skip the part about installing bash and Ubuntu

On macOS

- First install Homebrew:
 - REQUIREMENTS FOR HOMEBREW:
 - 64-bit Intel CPU
 - macOS High Sierra (10.13) or higher
 - Open the Terminal
 - Copy and paste the following line and press Enter:
 - `/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`
 - If you would like to read up on using homebrew visit <https://brew.sh/>
- Once Homebrew has been installed, run the following command:
 - `brew install libusb autogen automake wget pkg-config cmake openocd`
- Install the ARM GCC toolchain:
 - Run the following command: `brew tap ArmMbed/homebrew-formulae`
 - Run the following command: `brew install arm-none-eabi-gcc`
- Get the latest source code from GitHub:
 - `git clone https://github.com/q4klx/MMDVM`
 - `cd MMDVM`
 - `git submodule init`
 - `git submodule update`
- Edit Config.h. Uncomment the line: `#define MODE_LEDS`
 - If you want to connect a Nextion display, then also uncomment the line: `#define SERIAL_REPEATER`
- To start build run: `make pi-f722`
- Binaries will be under the `bin/` folder

On Pi-Star

NOTE! You cannot build firmware for the MMDVM-Pi v1.0 under Pi-Star v3.4.17 or below because the compiler doesn't fully support the STM32F7 chips. You must use the latest Pi-Star Beta v4.1.0-RC4 or above.

- Turn on Raspberry Pi
- Go to *Configuration->Expert->SSH Access*
- Login to pi-star
- Run command *rpm-rw*
- Run command *git clone <https://github.com/g4klx/MMDVM>*
- If you get an error saying "Could not resolve host", it likely means that your network is setup for IPV6 and the Pi has not been able to acquire the IPV4 nameserver via DHCP. Try the following. Otherwise skip to the "Install the toolchain" step
 - Run command *sudo vi /etc/resolv.conf*
 - Move cursor to the end of the line that starts with "nameserver" and then press the "a" key on your keyboard in order to move the cursor over
 - Press Enter to start typing on a new line, and then type this in:

■ *nameserver 8.8.8.8*



```
Dashboard | Admin | Update | Upgrade | Backup/Restore | Configuration
Quick Edit: DStarRepeater | mDStarGateway | TrunkServer | MMDVMHost | DMK GW | YSF GW | P25 GW | NCSX GW | DAPNET GW
Full Edit: DMK GW | PStar-Remote | WSP | RM API | DAPNET API | System Open | KISCC Dst | Tealer: CSR Tool | SSH Access

SSH - Pi-Star
# Generated by resolvconf
nameserver fe80::7ad2:94ff:fe73:f0c6%eth0
nameserver 8.8.8.8
```

- Press the ESC key on your keyboard
- Then type the following:

■ *:wq*



```
Dashboard | Admin | Update | Upgrade | Backup/Restore | Configuration
Quick Edit: DStarRepeater | mDStarGateway | TrunkServer | MMDVMHost | DMK GW | YSF GW | P25 GW | NCSX GW | DAPNET GW
Full Edit: DMK GW | PStar-Remote | WSP | RM API | DAPNET API | System Open | KISCC Dst | Tealer: CSR Tool | SSH Access

SSH - Pi-Star
# Generated by resolvconf
nameserver fe80::7ad2:94ff:fe73:f0c6%eth0
nameserver 8.8.8.8
:wq
```

- Then press Enter
- You should now have exited the text editor. You can try the git clone command again and it should work now

```

pi-star@pi-star(rw):~$ git clone https://github.com/g4klx/MMDVM
Cloning into 'MMDVM'...
fatal: unable to access 'https://github.com/g4klx/MMDVM/': Could not resolve host: github.com
pi-star@pi-star(rw):~$ sudo vi /etc/resolv.conf
pi-star@pi-star(rw):~$ git clone https://github.com/g4klx/MMDVM
Cloning into 'MMDVM'...
remote: Enumerating objects: 3746, done.
remote: Total 3746 (delta 0), reused 0 (delta 0), pack-reused 3746
Receiving objects: 100% (3746/3746), 3.77 MiB | 3.64 MiB/s, done.
Resolving deltas: 100% (2532/2532), done.
pi-star@pi-star(rw):~$

```

[Click here for fullscreen SSH client](#)

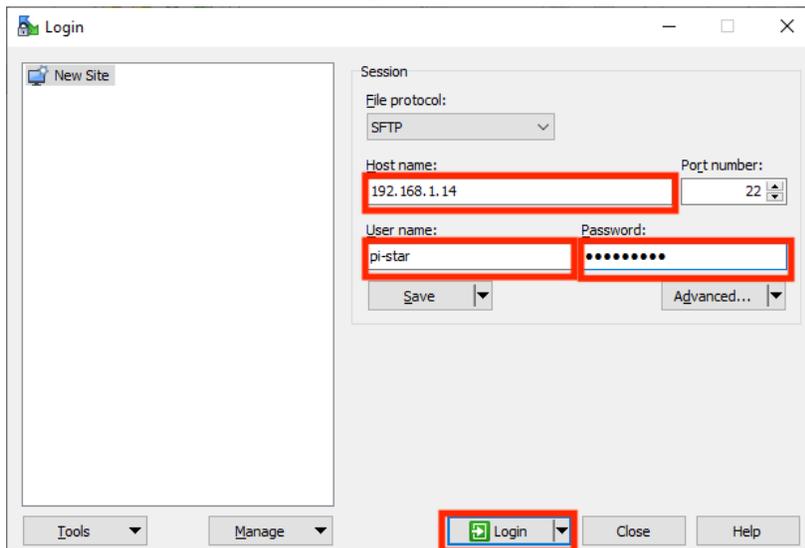
Pi-Star web: [radio](#), © Andy Taylor (G4KLX) 2014-2018
 Read help? [Click here for the Support Docs](#)
 ... get your copy of Pi-Star from here

- Install the toolchain and necessary packages by running the following command:
 - `sudo apt-get install git gcc-arm-none-eabi gdb-arm-none-eabi libstdc++-arm-none-eabi-newlib autoconf libtool pkg-config libusb-1.0-0 libusb-1.0-0-dev`
- Go into source folder: `cd MMDVM`
- Get submodules by running the following commands:
 - `git submodule init`
 - `git submodule update`
- Edit Config.h. Uncomment the line:
 - `#define MODE_LEDS`
 - If you want to connect a Nextion display, then also uncomment the line:
 - `#define SERIAL_REPEATER`
- To start build run: `make pi-f722`
- Binaries will be under the `bin/` folder

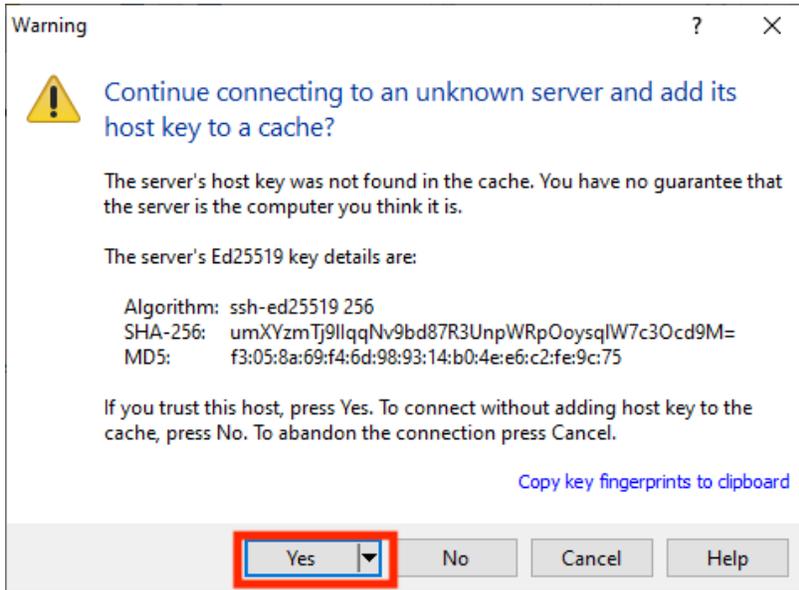
Flashing MMDVM-Pi

Windows

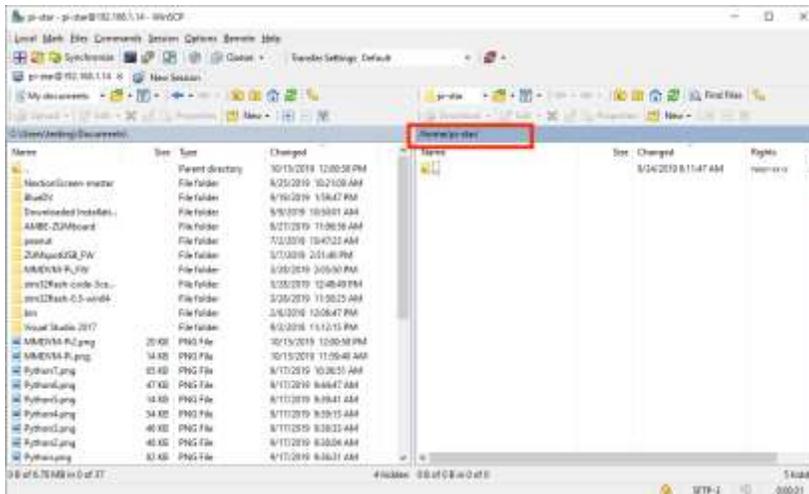
- Turn on Raspberry Pi
- Open your preferred web browser (Chrome) and go to <http://pi-star.local>
 - NOTE: If <http://pi-star.local> does not work for you then you will have to connect to Pi-Star via the IP Address (Ex. <http://192.168.1.14>). In order to find out what is the IP address of your Raspberry Pi, see the section titled **Find IP Address**
- Go to *Configuration->Expert->SSH Access*
- Login to pi-star
- Run command *rpm-rw*
- Transfer binary from the previous section **Building Firmware** to the Raspberry Pi. Should be under *bin -> mmdvm_f7.bin*.
 - You can use an app like WinSCP to transfer the file. You can download it from <https://winscp.net/eng/index.php>
 - Open WinSCP
 - Under “Host name” you can enter <http://pi-star.local> or if in your case you access Pi-Star via the IP address then that is what would go there.
 - Enter username which is *pi-star*
 - Enter password. If you haven’t changed it then it should be: *raspberrypi*
 - Press the “Login” button in order to connect to the Raspberry Pi



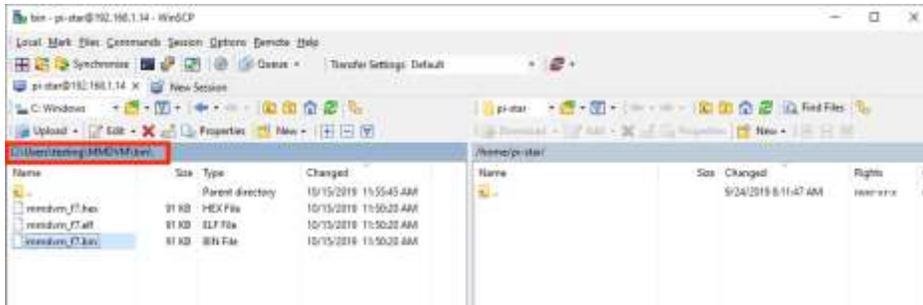
- o If a window pops up asking if you want to continue connecting, select "Yes"



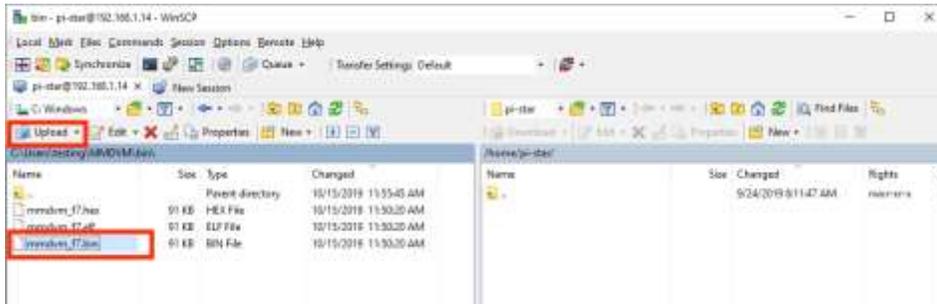
- o Once logged in you should see a screen similar to this



- o Navigate to the folder where you built the firmware



- Transfer the file named mmdvm_f7.bin by pressing the “Upload” button



- File has now been transferred



- Run the following command:
 - `cd ~/`
- Download the flashing script by running the following command:
 - `curl -OL https://raw.githubusercontent.com/veraabad/MMDVM-Pi_Update/master/MMDVM-Pi_FW_Update.sh`
- Run the following command:
 - `ls`
- If you see something like the following image then continue to the “Copy and paste” step, otherwise start back at the beginning of the **Windows** section



- Copy and paste the following command and then press Enter:
 - `sudo chmod +x MMDVM-Pi_FW_Update.sh`
- Run command `./MMDVM-Pi_FW_Update.sh`
- The install script will take care of flashing the MMDVM-Pi

Linux and macOS

- Turn on Raspberry Pi
- Open your preferred web browser (Chrome) and go to <http://pi-star.local>
 - NOTE: If <http://pi-star.local> does not work for you then you will have to connect to Pi-Star via the IP Address (Ex. <http://192.168.1.14>). In order to find out what is the IP address of your Raspberry Pi, see the section titled **Find IP Address**
- Go to *Configuration->Expert->SSH Access*
- Login to pi-star
- Run command *rpi-rw*
- Transfer binary from the previous section **Building Firmware** to the Raspberry Pi. Should be under *bin -> mmdvm_f7.bin*.
 - On macOS and Linux, you can use the Terminal by running the following commands:
 - Go to MMDVM folder: *cd MMDVM*
 - *scp bin/mmdvm_f7.bin pi-star@pi_star_ip_address:~/*
 - You will need to change *pi_star_ip_address* to the IP address off the Raspberry Pi.
- From the Pi-Star SSH window, run the following command:
 - *cd ~/*
- Download the flashing script by running the following command:
 - *curl -OL https://raw.githubusercontent.com/veraabad/MMDVM-Pi_Firmware/master/MMDVM-Pi_FW_Update.sh*
- Run the following command:
 - *ls*
- If you see something like the following image then continue to the “Copy and paste” step, otherwise start back at the beginning of the **Linux and macOS** section



The screenshot shows the Pi-Star web interface at the top with a red header. Below the header is a navigation menu with links: Dashboard | Admin | Update | Upgrade | Backup/Restore | Configuration. Underneath, there are two rows of links for quick and full editing of various components. At the bottom, an SSH terminal window is open, showing the following commands and output:

```
pi-star@pi-star(rw):~$ ls
mmdvm_f7.bin  MMDVM-Pi_FW_Update.sh
pi-star@pi-star(rw):~$
```

- Copy and paste the following command and then press Enter:
 - *sudo chmod +x MMDVM-Pi_FW_Update.sh*
- Run command *./MMDVM-Pi_FW_Update.sh*
- The install script will take care of flashing the MMDVM-Pi

Pi-Star

If you built under the Pi-Star beta follow these instructions

- Turn on Raspberry Pi
- Open your preferred web browser (Chrome). On an iOS/OSX device go to web address <http://pi-star.local> and for other operating systems go to <http://pi-star>
 - NOTE: If <http://pi-star.local> / <http://pi-star> addresses do not work for you then you will have to connect to Pi-Star via the IP Address (Ex. <http://192.168.1.14>). In order to find out what is the IP address of your Raspberry Pi, see the section titled **Find IP Address**
- Go to *Configuration->Expert->SSH Access*
- Login to pi-star
- Run command *rpi-rw*
- Go to the folder where the binary is located. This binary file was created from following the previous section **Building Firmware**.
 - Binary file should be under *MMDVM -> bin -> mmdvm_f7.bin*
- From the Pi-Star SSH window, run the following command:
 - *cd MMDVM/bin/*
- Download the flashing script by running the following command:
 - *curl -OL https://raw.githubusercontent.com/veraabad/MMDVM-Pi_Update/master/MMDVM-Pi_FW_Update.sh*
- Run the following command:
 - *ls*
- If you can see both the *mmdvm_f7.bin* file and the *MMDVM-Pi_FW_Update.sh* file then continue to the “Copy and paste” step, otherwise start back at the beginning of the **Pi-Star** section



- Copy and paste the following command and then press Enter:
 - *sudo chmod +x MMDVM-Pi_FW_Update.sh*
- Run command *./MMDVM-Pi_FW_Update.sh*
- The install script will take care of flashing the MMDVM-Pi

Find IP Address

If you are having issues connecting to Pi-Star via <http://pi-star.local> / <http://pi-star> then you will have to connect via the IP address of the Raspberry Pi. Follow these instructions in order to find out what is the IP address of the Raspberry Pi.

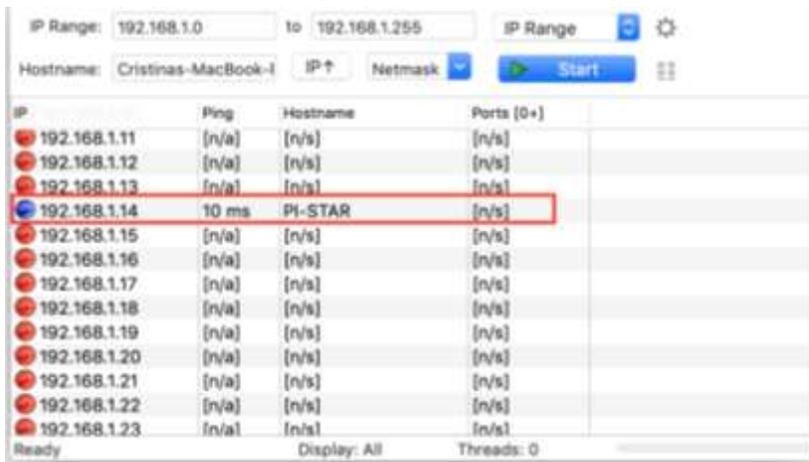
- Download Angry IP Scanner from <https://angryip.org/download>. It's available for Windows, macOS and Linux
- Follow the instructions from the installer
- Once installed open the application and click on the "Start" button



- It will take a few seconds to find out what is connected to your network
- Once it is done a pop-up window will show you how many hosts are connected to your network. Click "Close" to dismiss it.



- Now scroll through the addresses until you find one named PI-STAR.



- Use that IP address to connect to pi-star. In your preferred web browser enter the following: http://IP_ADDRESS
 - Replace IP_ADDRESS with the one you found in Angry IP Scanner. In the example image above the IP address is 192.168.1.14 So in that example, <http://192.168.1.14> would be what you would type into a web browser to access Pi-Star

Info for use under Windows 10

This is a web page with a good tutorial on how to use the mmdvmcal software tool and a spectrum analyzer to adjustment the signal levels for DMR:

<https://www.f5uui.net/en/installation-calibration-adjustment-tunning-mmdvm-mmdvmhost-raspberry-motorola-gm360/5/>

Another document describing the spectrum adjustment process:

<http://www.swedmr.se/wp-content/uploads/2017/08/Justering-av-repeater-med-MMDVM.pdf>

Support

Great video from W1MSG showing getting started with Pi-Star:

<https://www.youtube.com/watch?v=B5G4gYDdJeQ>

MMDVM Yahoo group:

<https://groups.yahoo.com/neo/groups/mmdvm/conversations/messages>

Pi-Star support forum:

<https://forum.pistar.uk/>

Pi-Star Facebook support group:

<https://www.facebook.com/groups/pistar/>

Pi-Star Wiki:

<http://wiki.pistar.uk>

A web page describing the setup of an MMDVM repeater:

<https://sadigitalradio.com/digital-radio-how-tos/make-mmdvm-digital-repeater/>