Roger Clark

Freelance IT Consultant & Developer,
Melbourne Australia

OpenGD77 – Hotspot mode

4
SEP 2019

by Roger Clark | posted in: DMR, GD-77, Ham radio | 20

For the last 6 weeks, I've been intensively working on the Hotspot mode as part of the OpenGD77 firmware, and it is now finally with the Beta testers in Australia, Europe and the USA.

Note. Currently firmware seems to be stable but not completely bug free, so I am limiting its use to the Beta testers.

Hotspot mode converts a Radioddity GD-77, running the OpenGD77 firmware, into a **DMR ONLY** high power hotspot when its connected to a Raspberry Pi running PiStar.

No other hardware is required, just the RPi and the GD-77

In this photo I'm using a Raspberry Pi 3B+, but I've also tested with a RPi Zero W and it works just as well.

Search

Top Posts & Pages


OpenGD77 - Hotspot mode

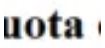

Announcing - OpenGD77 firmware - Phase 1


DMR MMDVM duplex hotspot


Investigating a RCWL 9196 / RCWL-0516 "Radar" motion detector module


Beware dodgy Vector Network Analysers


GD-77 Community CPS - initial OpenGD77 support features


Blog offline due to server quota problems


STM32F103 and Maple / Maple Mini with Arduino 1.5.x IDE


Arduino on the nRF51822 Bluetooth Low Energy microcontroller


Arduino STM32 - USB Serial and DFU

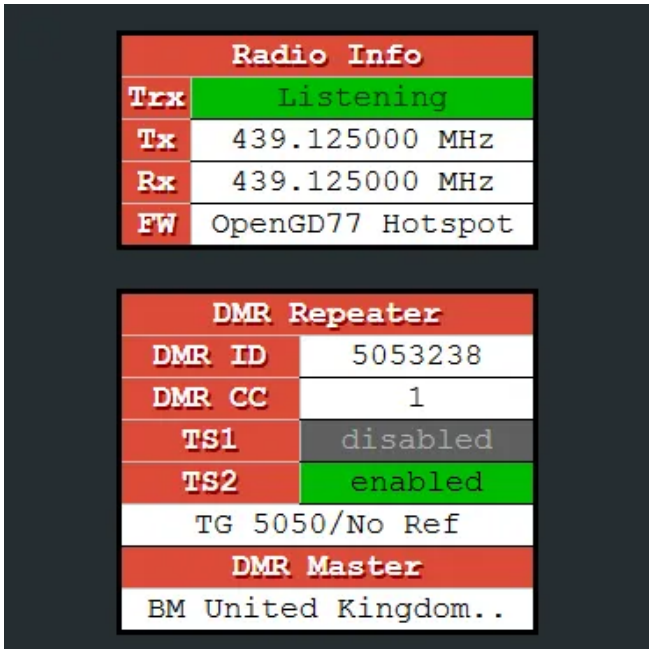
Categories

Select Category

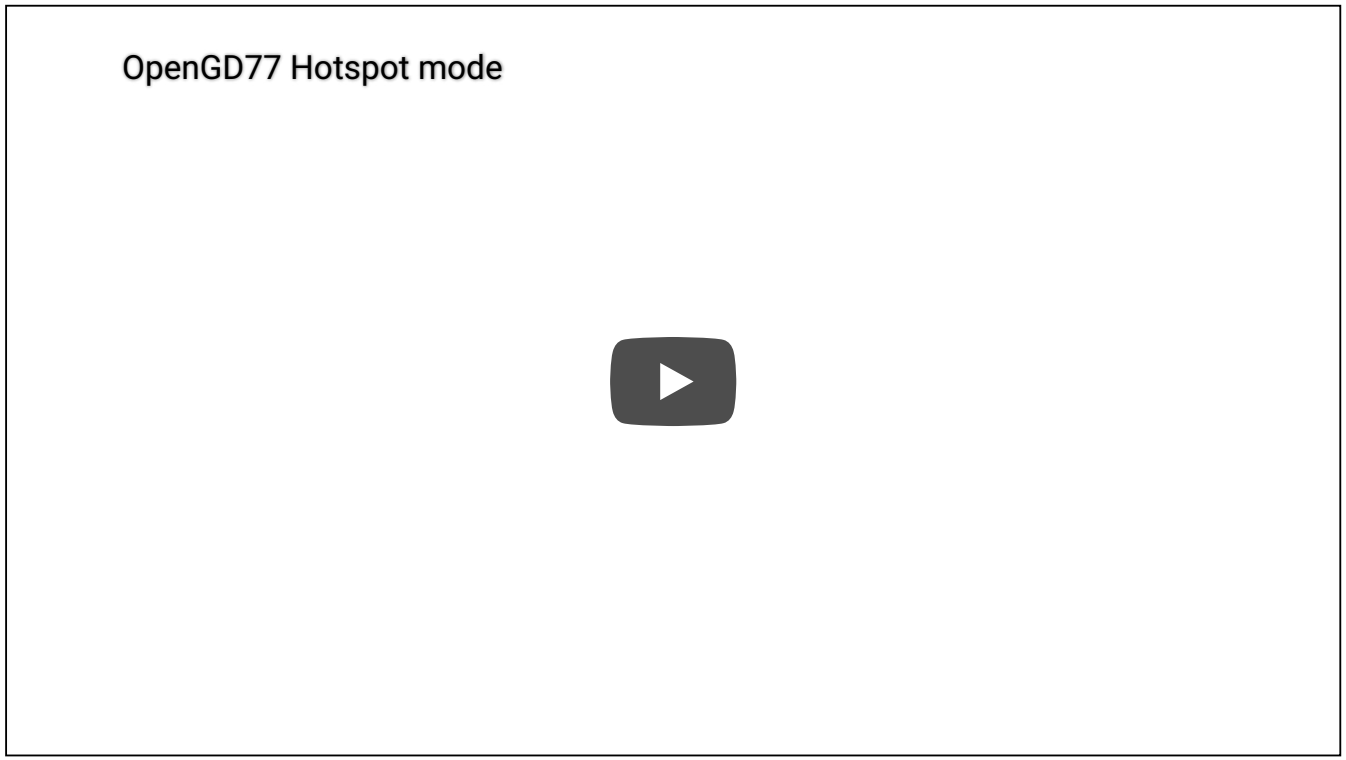
The GD-77 is connected to the RPi via its USB programming cable, and the only change that needs to be made in PiStar, is to change the modem type to any of the “Zumspot” compatible modem types e.g. the “Zumspot Libre”

One thing however you won’t see unless you modify you copy of PiStar is the “modem” name being displayed in the PiStar dashboard, because PiStar only supports displaying the firmware version for the specific “modems” in the list

I have modified my copy of PiStar to add the OpenGD77 Hotspot, and I presume if enough people eventually use their GD-77’s as hotspots, Andy may add the OpenGD77 Hotspot as a new modem type in PiStar, so that it will display correctly.



I’ve done rather shaky video showing the OpenGD77 Hotspot mode working with a Rasperry Pi Zero.



For those of you interested in the technical details, please read on..

Conceptually both Kai and I assumed that adding Hotspot mode functionality to the OpenGD77 firmware should not be too complicated, because we can access the DMR data from any transmission which the GD-77 receives, and can also transmit and audio and other data via DMR RF.

Unfortunately our assumptions about the difficulty in writing Hotspot mode were not correct, and a great deal of new code needed to be added to the OpenGD77 firmware to support Hotspot mode.

Before I continue, I think its worthwhile explaining how a normal PiStar hotspot works.

At the core of PiStar is a program called MMDVMHost, which was written my Jonathan G4LKX, which interfaces the original MMDVM modem hardware, to the Internet servers for DMR and other protocols.

MMDVMHost communicates with the “modem” using its own serial protocol, originally via USB Serial but now more commonly via GPIO serial on a Raspberry Pi to a MMDVM_HS Hat board, aka Jumbospot.

MMDVMHost handles the server connectivity and a loads of other functions, to seamlessly transfer the data, to and from the “modem” board.

I had assumed that the data being set between the “modem” hardware and MMDVMHost contained high level DMR data, including the transmitting stations DMR ID number and the Talkgroup number as well as the AMBE compressed audio data bytes. However, when intercepted some of the serial data being sent between the Raspberry Pi and my duplex hat board, (running in simplex mode), I realised that I was at least partially wrong about how MMDVMHost and MMDVM (or MMDVM_HS) work.

I expected to see data bytes containing my DMR ID (5053238) , as well as bytes containing the Talkgroup I was using (505), but the data between the hotspot board and the RPi did not seem to contain that data at all.

After some research, I found out that the data is in the “On Air” format, and that modem (Hat etc) boards only do minimal processing of the DMR 4FSK data which they receive before sending it to MMDVMHost

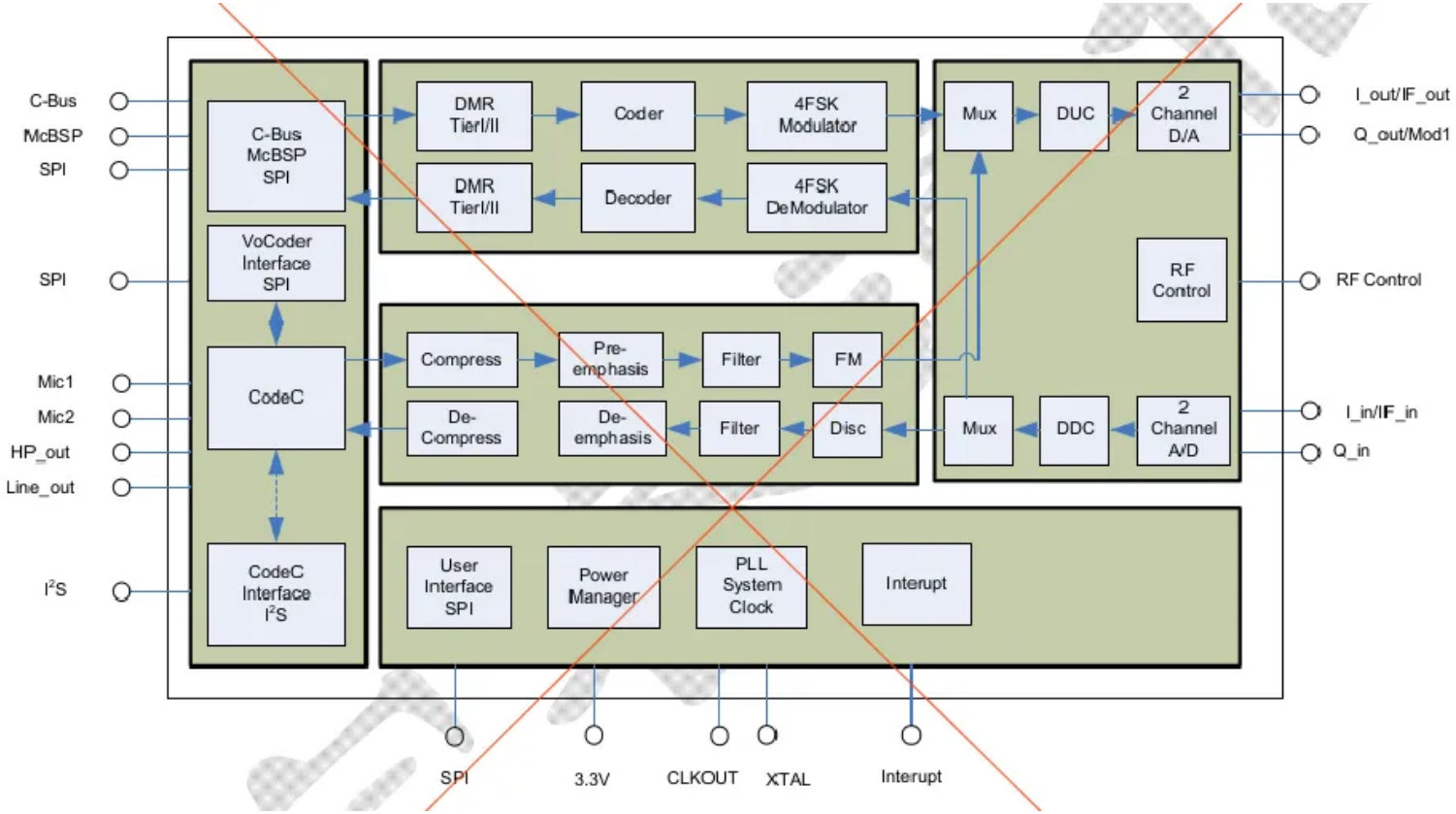
MMDVMHost then creates a packet of network data, which is a header containing the Source and Destination ID's etc, followed by the raw "On Air" data.

At the other end of the system, e.g. any hotspots connected to the same Talkgroup; MMDVMHost, sends the On Air portion of the network packet of data to the modem, and the modem basically just remodulates the On Air data to 4FSK and transmits it.

The GD-77 hardware however, works completely differently….

The GD-77 has a dedicated IC (Hongrui HR-C6000) which performs both the 4FSK demodulation and also completely decodes the DMR "On Air" protocol, so that the CPU in the GD-77 does not need to do this processing.

HR-C6000



Note. This is the same IC that is used in radios like the TYT MD-380 etc and I Baofeng RD-5R and Baofeng DM-1801 etc

Except the MD-380 uses the older version, called the HR-C5000

Hence the only way to interface the OpenGD77 firmware to MMDVMHost was for the OpenGD77 firmware to generate the “On Air” format data that MMDVMHost and every other hotspot expected to receive.

This would have been an incredibly difficult task, except that MMDVMHost contains all the functionality that I needed to both encode to the “On Air” format and also to decode the “On Air” format, and because MMDVMHost is open source, I was able to incorporate the majority of its DMR processing functionality, into the OpenGD77 firmware.

Unfortunately, even though MMDVMHost contained all the necessary functionality, MMDVMHost does not use the functionality in the way that I needed to. For example, It normally decodes the “On Air” data, rather than encoding it, so I had to work out which bits of MMDVMHost I needed to encorporate in the OpenGD77 firmware and also how to use those parts, to do almost the reverse of what they normally do.

And just to complicate things, MMDVMHost is written in the C++ language in an Object Orientated way, whereas the OpenGD77 firmware uses the C programming language which does not support Objects or a number of other features of C++, like function overloading which MMDVMHost makes use of.

So all the source code files needed to be “ported” from C++ to C by making changes to around 50% of the code.

Consequentially, it took me 3 or 4 weeks to find the parts of the functionality in MMDVMHost which I needed to port, and to port them and integrate them into the OpenGD77 firmware.

Additionally, I had to implement the functionality to communicate via MMDVHost’s serial protocol, as well as integrate the new functionality into the existing OpenGD77 Tx and Rx functionality.

Needless to say this was quite a complex task, but after 50 hours of work, I was able to receive network data from MMDVMHost and get the GD-77 to transmit it via RF.

It took me another week to write the functionality so that the received RF DMR signal was sent to MMDVHost, and yet another week to add some missing functionality like Private Calling, and also the transmission of Talker Alias data when received from MMDVMHost as embedded data in the Voice audio frames.

Various versions of the firmware have been with the Beta testers for about a week, and they are still uncovering a few bugs, so currently I don’t feel its stable enough for general release. However I plan to make it part of the OpenGD77 firmware in the fullness of time.

BTW.

If anyone has some specific questions, please post comments, as I have glossed over a lot of the complex technical details of how I made this work, as that would probably be incredibly tedious and also take many hours to document 😉

Previous Post

20 Responses

Scott Evans 04/09/2019 |

Hi Roger, you should be able to push your pistar code edits directly into Andy's main branch and then request a pull request, then he can review and then have the opengd77 added as a separate modem type.

Bob 04/09/2019 |

Hi Roger very interested in your gd77 hotspot project. When will the firmware be released or is it possible to become a beta tester

73 Bob g1saa

Anonymous 04/09/2019 |

Great work Roger!!

mi0ceo 04/09/2019 |

Excellent article on the Hotspot mode looking froward to using it (have a spare GD77). Will I have to use the beta firmware that you have developed for the GD77 on can it run with the stock firmware.
Best George

Mike

05/09/2019 |

Hi Roger,
really impressive progress and results you and Kai reached with the OpenGD77 firmware.
Congrats, this is a huge step of a free firmware and software for this neat radio.
It might also could have impact to some other projects.
73 Mike, DL2MF

Anonymous

05/09/2019 |

Great work Roger!! It's a big progress and a very interesting option to setup a very simple, cost effective, mid power and full digital DMR node. Is it possible to use both TS (at the same time or not)?

Pedro

05/09/2019 |

Thank you very much for your great work!
I just wonder: once you got up to this point of high level development… why not do the final jump of DMR evolution and implement the Codec2 instead of AMBE? The DMR protocol is codec agnostic.
I am pretty sure David Rowe would be happy to adjust word lengths or do the necessary adjustments to match the payload.

73 de M0IEI/EA1CRF

Andres

05/09/2019 |

Totally awesome work… I will keep an eye on you guys, going to test it as soon as it is public. Thanks for that work!
73 de HJ3BUA

Roberto

05/09/2019 |

Felicitaciones, genial tu trabajo, un abrazo de Chile

Rick

05/09/2019 |

Hello, can you send me a copy of the firmware for test in my unused MD-380? Many thanks in advance!

Hardy Wounlund

05/09/2019 |

Great job Roger, looking forward to try it 😉

Fernando S

05/09/2019 |

Great work Roger! A very simple, cost effective and reliable way to build a DMR node. Is it possible (now or in future) to use both TS at the same time?

Stefan

05/09/2019 |

Hi Roger, great work. One Question: If it is used like a ZUM Spot, is it then also usable as DStar Hotspot?

Roger Clark 05/09/2019 |

Codec2 could be used instead of AMBE, but it would be impossible to use the radio to communicate with normal DMR users.

Unfortunately, I don't think there is enough room in the MCU ROM for both the AMBE and Codec2 codecs unless Codec2 is quite small, and perhaps I removed some other functionality to make more space

Roger Clark 05/09/2019 |

I am not sure quite what you mean about using both TimeSlots at the same time.

The radio DMR DSP chip is set to Tier 2 mode, but the firmware does not support everything needed to communicate with a Tier 2 Repeater.

If you mean, can it operate as s Duplex Hotspot, then the answer is No, because a Duplex Hotspot transmits and receives at the same time and requires 2 separate antenna connectors as well as all the internal RF Hardware to be able to do that.

Roger Clark 05/09/2019 |

Yes.

I could do that. The changes to PStar are minimal. As the functionality is the same as a Zumspot Libre, except it only supports DMR mode.

When I have time I will fork the PiStar desktop repo, and make the necessary changes.

However I think a few people would need to lobby for its inclusion.

Roger Clark 05/09/2019 |

No. It can only do DMR because the hardware in the radio has a dedicated DMR IC, and currently we don't know how to use the IC to access the output of its 4FSK decoder, rather than the DMR data output.

The block diagram of the IC seems to suggest that there is no way to access the 4FSK data, but the diagram may be misleading, because there is a way to access the FM audio according to the the data sheet, but there is no path for FM data on the block diagram either.

Mike G4KXQ

05/09/2019 |

Roger I've got one handset with the display problem that Radioddity replaced. Could I load and operate this firmware without any screen? That would be a great use of a defunct handheld.

Roger Clark

05/09/2019 |

Yes.

It doesn't matter if you don't have a screen, however the power will default to about 1W, because the radio is not on a specific channel since its totally controlled by PiStar, and the default power that the firmware uses is 1W.

Currently I don't take any notice of the power value being sent by PiStar (MMDVMHost), because it always defaults to 100% which would be 5W, and the PA in the radio would overheat if used at 5W for a hotspot, for a sustained amount of time.

Currently there is no way to adjust the default power without being able to see the screen. Though I'm thinking about putting a control on the Hotspot screen, to allow the power to be adjusted by pressing the Left and Right arrow keys. And if I did that, you could adjust the power even if you can't see whats on the display.
However it would be a bit tricky to know what value you have selected, because most power meters can't measure DMR power, because of the way it pulses on and off.

BTW.
Problems with the screen are usually quite easy to fix. The majority of the problems are just that the ribbon cable connecting the display is not completely pushed into the connector on the main PCB.

On rare occasions the display LCD is not making good contact with the rubber conductive strips which connect it to the display PCB, but this can often be cured by cleaning the connections and re-seating them.

Very rarely is the display completely broken.

PPS.

I'm thinking of how to either measure the internal temperature in the radio, if the CPU in the radio has an internal thermometer (some CPUs have this but I'm not sure if the MK22 CPU in the GD-77 does).
And I could possibly use the temperature to determine whether the whole radio is getting too hot.

Alternatively, I thought about varying the power when in hotspot mode, based on the duty cycle of the PA. i.e allow it to transmit a 5W for perhaps 5 minutes at a maximum, and then if the duty cycle was high, with very little breaks between long transmissions, I would start to automatically decrease the power level, until the duty cycle decreased again.

Roger Clark 05/09/2019 |

Hi Scott

I've now forked the Pi-Star_DV_dash and updated it to support the OpenGD77

https://github.com/rogerclarkmelbourne/Pi-Star_DV_Dash

I'll send it as a PR, but mainly to let Andy take a look at it to see if what I've done is correct.

Leave a Reply

Enter your comment here...

Archives

Categories

Recent Posts

Select Month

Select Category