Andrew Milluzzi, KK4LWR

125 SE 16th Ave., Apt L202, Gainesville, FL 32601; **andy@gatorradio.org**

Getting Started with DMR and DSDPlus

This simple tutorial can help get you started on digital modulation.

Digital modulations are becoming quite popular in Amateur Radio. Technologies like DMR, D-STAR, and System Fusion have made the technology accessible and a new platform for experimentation. Recently there was much buzz about the Tytera MD-380, an inexpensive digital mobile radio (DMR) handheld transceiver. Technologies such as the RTL-SDR — software defined radio based on the RTL chip set — have brought software defined radio to the masses. Combining these two technologies can enable additional experimentation.

Recently the Gator Amateur Radio Club, W4DFU, at the University of Florida installed a DMR repeater for Gainesville and the surrounding area. This repeater is a great resource for students and Technician class operators to talk all over the world using UHF locally. It is easy to get started with DMR by just listening to typical DMR contacts. Depending on the talk group, some are quick, others are well organized nets. One option to decode DMR is to purchase a radio. Another is constructing a receiver from an RTL-SDR, an antenna, and some software.

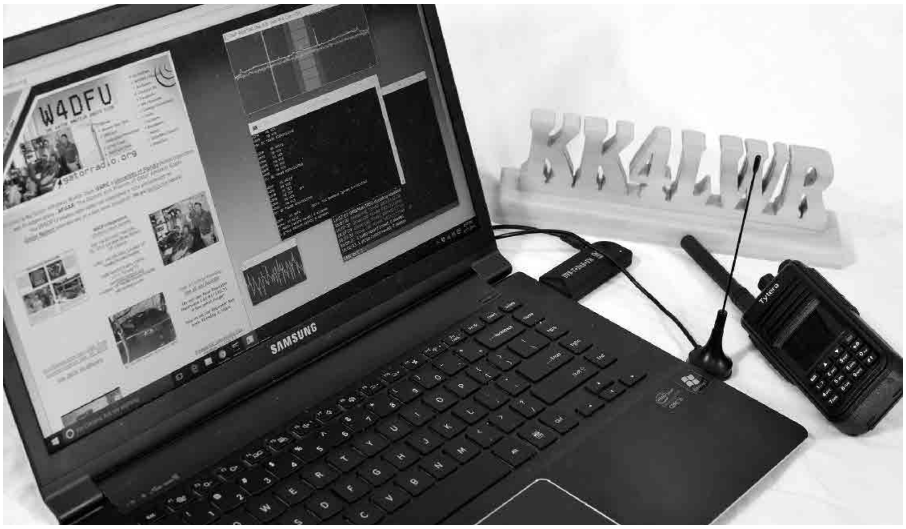


Figure 1 — DMR decoding setup using a Tytera MD-380 handheld transceiver, a laptop computer, and an RTL-SDR dongle. [Andy Milluzzi, KK4LWR, photo]

What is DMR?

DMR stands for Digital Mobile Radio. Sometimes it is called MotoTURBO, the DMR product produced by Motorola Solutions. Like D-STAR or System Fusion, DMR digitizes voice using a vocoder and sends the information via digital packet. Unlike D-STAR and System Fusion, DMR uses TDMA (Time Division Multiple Access) with two time slots in 12.5 kHz. This enables one DMR repeater to act as two on a given frequency.

What is DSDPlus?

DSDPlus is an application that runs on a Windows computer that can decode multiple digital modes.[1] Similar to the open source DSD program, DSDPlus takes an audio stream from a radio and can generate text or audio. The main difference for the scope of this tutorial is that the open source program must be compiled from source code, while DSDPlus offers a Windows binary.

Hardware Setup

My decoding setup requires just a few pieces of physical hardware, seen in Figure 1, and a some virtual hardware. The first thing you need is a DMR source. I already had purchased a Tytera MD-380 handheld transceiver, since I am quite interested in the ongoing firmware experimentation community. You might already have a DMR source, via other hams and perhaps a local repeater. If you do decided to purchase a DMR radio, Motorola Solutions and Hytera each make some outstanding radios. Both Connect Systems and Tytera make some more inexpensive options.

The second needed piece of hardware is a radio to connect to your computer. You could use a VHF/UHF radio and a sound card to interface with your computer. However, I opted for an RTL-SDR for this project. An RTL-SDR is essentially an inexpensive TV tuner USB dongle for your computer. It can be purchased for less than $20 and can cover 50 MHz to 1.7 GHz. Your Windows operating system will most likely need a special driver to use the card as an SDR. You can install the correct driver with assistance from the Zadig web page.[2] The Zadig website has a simple guide for installing the WinUSB driver. This tutorial will assume you are using an RTL-SDR with the appropriate driver.

The final needed hardware component for this tutorial is virtual. The DSD software needs a way to pass the audio. If you are using your computer sound card and an external radio, then you can skip this hardware. I recommend a VB-Audio Virtual Cable as a free solution.[3] When you download and install the software, you should notice a new audio input and output device on your computer that acts as a sound card.

Setting up DSDPlus Suite

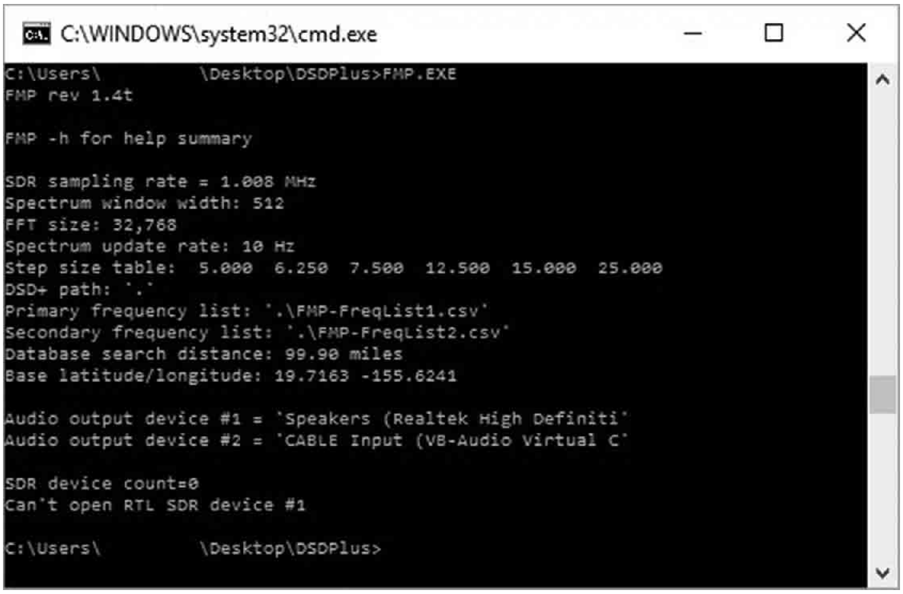
Download the latest version of DSDPlus and DSDPlus DLL package from the DSDPlus web page.[4] Extract the contents of the DSDPlus zip file and the DLL zip file to a folder. You should see a few dozen files. For this tutorial, we will focus on FMP-VC.bat and VC.bat. In DSDPlus two programs must be configured: FMP and DSDPlus. FMP is a basic narrowband FM tuner for an SDR. DSDPlus is the decoder.

With all the files extracted, open a command prompt. In the Windows environment, this can be done by pressing the Windows key on your keyboard and typing CMD. Once open, navigate to the folder containing the DSDPlus files. Another option in Windows 8 or Windows 10 is to click on the File menu in your Explorer window. One of the options listed is open a command prompt. If done from the DSDPlus folder, you will not need to navigate.

Setting up FMP

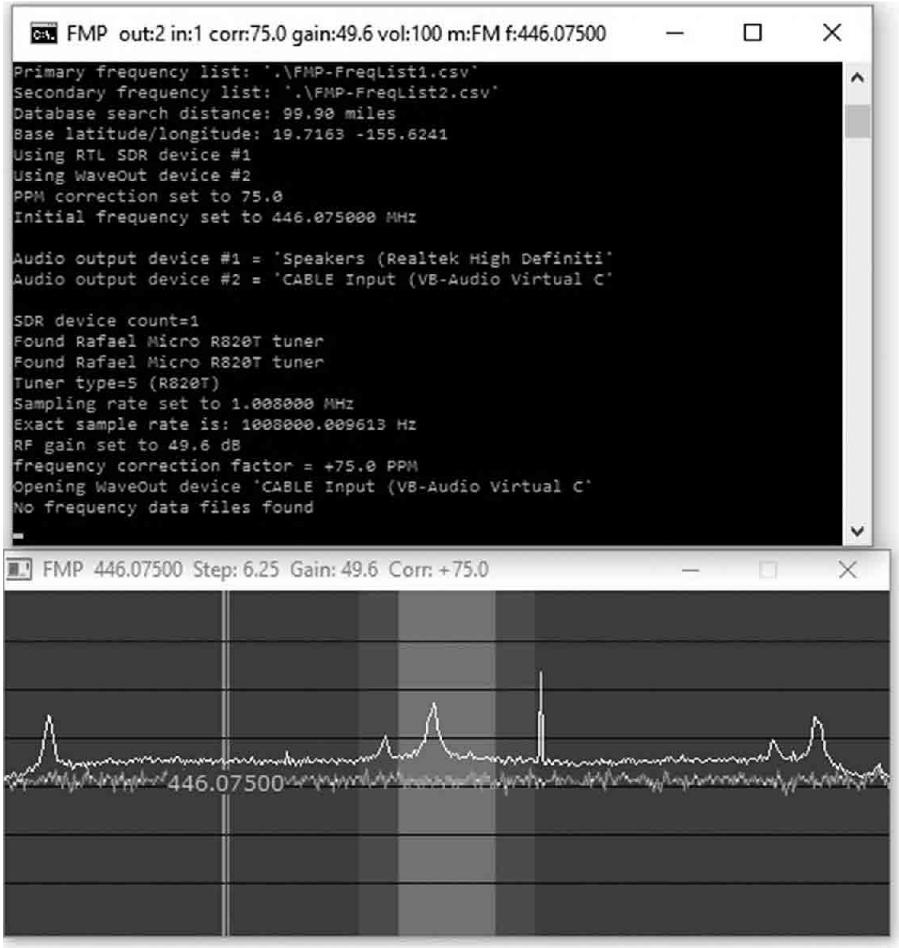
Open the file FMP-VC.bat. You will see several parameters. Modify this script for our use by configuring the four parameters "–I", "–o", "–P", and "–f". The "–i" parameter followed by number (without a space) indicates which SDR to use as an input. For the SDR we can assume it is the only one on your computer, so set it to "–i1". The "–o" parameter followed by a number is the output audio device. The "–P" parameter is the parts-per million (ppm) correction for your SDR. The "–f" parameter is the frequency in megahertz.

To determine the values for the other parameters requires some knowledge about your setup. Let's first tackle the output audio source. Each computer is different. The best way to determine the audio output is to observe what FMP sees. For this next part, you might find it helpful to unplug the SDR from the computer; if it is plugged in the program will launch. Type "FMP.EXE" in your command prompt window. Without an SDR plugged in, the application should fail to launch, but will still list the audio devices. For this program, we need to feed the output to our virtual audio port input. On my computer it is "Audio output device #2" seen in Figure 2.



```
C:\Users\        \Desktop\DSDPlus>FMP.EXE
FMP rev 1.4t

FMP -h for help summary

SDR sampling rate = 1.008 MHz
Spectrum window width: 512
FFT size: 32,768
Spectrum update rate: 10 Hz
Step size table:  5.000  6.250  7.500  12.500  15.000  25.000
DSD+ path: '.'
Primary frequency list: '.\FMP-FreqList1.csv'
Secondary frequency list: '.\FMP-FreqList2.csv'
Database search distance: 99.90 miles
Base latitude/longitude: 19.7163 -155.6241

Audio output device #1 = 'Speakers (Realtek High Definiti'
Audio output device #2 = 'CABLE Input (VB-Audio Virtual C'

SDR device count=0
Can't open RTL SDR device #1

C:\Users\        \Desktop\DSDPlus>
```

Figure 2 — Running FMP.EXE produces this text. Note the listed audio output devices.


FMP out:2 in:1 corr:75.0 gain:49.6 vol:100 m:FM f:446.07500
FMP 446.07500 Step: 6.25 Gain: 49.6 Corr: +75.0
446.07500

```
Primary frequency list: '.\FMP-FreqList1.csv'
Secondary frequency list: '.\FMP-FreqList2.csv'
Database search distance: 99.90 miles
Base latitude/longitude: 19.7163 -155.6241
Using RTL SDR device #1
Using WaveOut device #2
PPM correction set to 75.0
Initial frequency set to 446.075000 MHz

Audio output device #1 = 'Speakers (Realtek High Definiti'
Audio output device #2 = 'CABLE Input (VB-Audio Virtual C'

SDR device count=1
Found Rafael Micro R820T tuner
Found Rafael Micro R820T tuner
Tuner type=5 (R820T)
Sampling rate set to 1.008000 MHz
Exact sample rate is: 1008000.009613 Hz
RF gain set to 49.6 dB
frequency correction factor = +75.0 PPM
Opening WaveOut device 'CABLE Input (VB-Audio Virtual C'
No frequency data files found
_
```

Figure 3 — Screen capture of FMP successfully running, controlling the SDR. Note the command prompt window showing the correct ppm correction and frequency.

Setting the ppm correction for your SDR is specific to your device. You can experiment by leaving it at 0.0. Alternatively you can use another SDR program such as SDR# from the AIRSPY web page to find the ppm correction.[5] My device required a ppm correction of +75.

The last parameter to set is the frequency to monitor. I do not want to clog the larger DMR network with my testing traffic, so I generated my own signals on 446.075 MHz, one of the UHF simplex frequency. DMR is mostly on UHF in the United States. Once the system is working, you can change the frequency to a local repeater.

If done correctly, your FMP-VC.bat should look like:

“FMP –i1 –o2 –P75 –f446.075”.

Plug in your SDR tuner and run the script. You should see several windows pop up. Your computer screen should look like Figure 3. Save the changes to FMP-VC.bat. You can close FMP by pressing the ESC key.

Setting up DSDPlus

Configure the DSDPlus script with a similar process. Open the VC.bat file and observe the various parameters. The parameters that start with a “w”:

“–wsl”, “–wss”, “–wel”, and “–wcl”

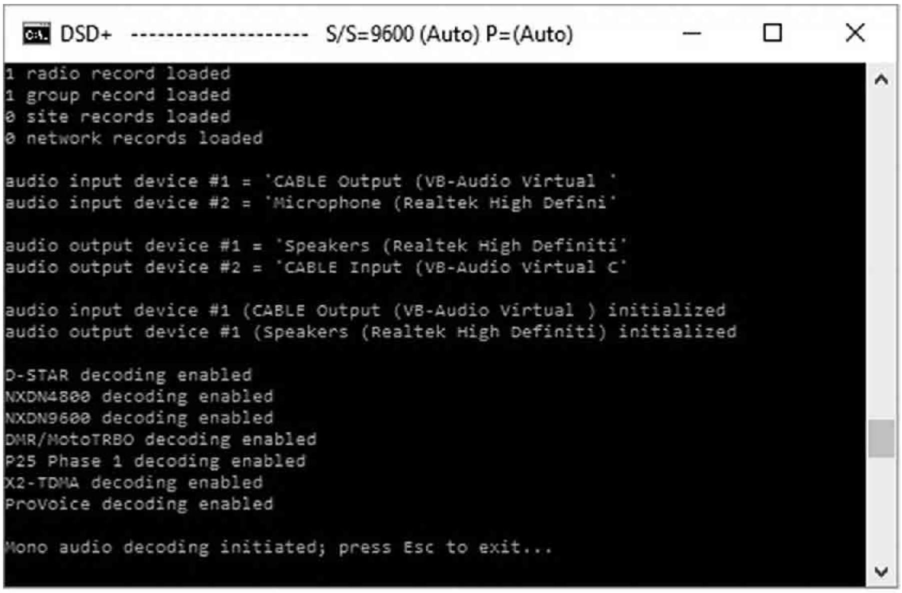
control where the windows are launched. You can set these as you like.

We must configure the audio as we did before by launching DSDPlus from our open command prompt. Unfortunately, there is no way to stop the program from opening all the windows, so you might need to move the various windows out of the way to see the command prompt and get your results. On my computer I got the results shown in Figure 4.

The input should be set to the virtual audio cable output. The output must be your system speakers. In my case this results in input device #1 and output device #1 respectively. This enables me to hear anyone calling.

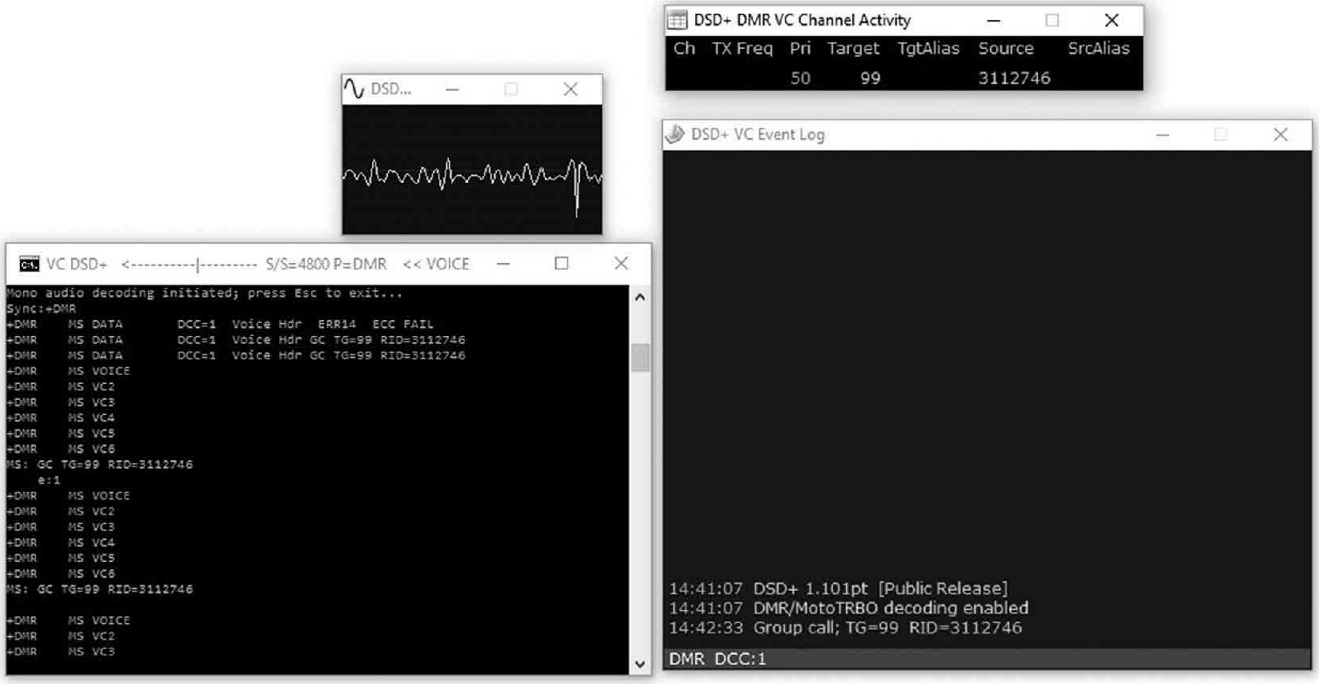
A few other parameters must be set to make DMR work. The first parameter, “–rv”, tells DSDPlus that we expect voice. DSDPlus must also be set to decode DMR/MotoTURBO, since DSDPlus can do much more than just DMR — DSDPlus documentation indicates that it decodes D-STAR, except audio. It also can also decode P25 Phase 1 and NEXEDGE.

To enable DMR we need to pass the “–fr” parameter. We also need to tell DSDPlus which of the two time slots we want to



```
1 radio record loaded
1 group record loaded
0 site records loaded
0 network records loaded

audio input device #1 = 'CABLE Output (VB-Audio Virtual '
audio input device #2 = 'Microphone (Realtek High Defini'

audio output device #1 = 'Speakers (Realtek High Definiti'
audio output device #2 = 'CABLE Input (VB-Audio Virtual C'

audio input device #1 (CABLE Output (VB-Audio Virtual ) initialized
audio output device #1 (Speakers (Realtek High Definiti) initialized

D-STAR decoding enabled
NXDN4800 decoding enabled
NXDN9600 decoding enabled
DMR/MotoTRBO decoding enabled
P25 Phase 1 decoding enabled
X2-TDMA decoding enabled
ProVoice decoding enabled

Mono audio decoding initiated; press Esc to exit...
```

Figure 4 — Running DSDPlus lists both input and output audio devices.


DSD+ DMR VC Channel Activity
DSD...
DSD+ VC Event Log
VC DSD+ <----------|--------- S/S=4800 P=DMR << VOICE

Ch	TX Freq	Pri	Target	TgtAlias	Source	SrcAlias
		50	99		3112746	

```
14:41:07  DSD+ 1.101pt  [Public Release]
14:41:07  DMR/MotoTRBO decoding enabled
14:42:33  Group call; TG=99  RID=3112746
DMR  DCC:1
```

```
Mono audio decoding initiated; press Esc to exit...
Sync:+DMR
+DMR    MS DATA       DCC=1  Voice Hdr  ERR14  ECC FAIL
+DMR    MS DATA       DCC=1  Voice Hdr GC TG=99 RID=3112746
+DMR    MS DATA       DCC=1  Voice Hdr GC TG=99 RID=3112746
+DMR    MS VOICE
+DMR    MS VC2
+DMR    MS VC3
+DMR    MS VC4
+DMR    MS VC5
+DMR    MS VC6
MS: GC TG=99 RID=3112746
    e:1
+DMR    MS VOICE
+DMR    MS VC2
+DMR    MS VC3
+DMR    MS VC4
+DMR    MS VC5
+DMR    MS VC6
MS: GC TG=99 RID=3112746

+DMR    MS VOICE
+DMR    MS VC2
+DMR    MS VC3
```

Figure 5 — A screen capture of several windows of DSDPlus while decoding.

monitor. This is done via passing "–1" or "–2" after the "–fr" parameter. Those are the only required parameters to make it work. You can record what DSDPlus decodes using the "–Pwav" parameter to save the audio as a wave file. You might also discover a "–v3" parameter in the sample provided by DSDPlus. This enables verbose logging and I recommend using it, especially for debugging your scripts.

When you are done, you should have a VC.bat script similar to:

"DSDPlus –fr –1 –rv -Pwav –i1 –o1 –v3 –wsl400.210 –wss100.200 –wel172.522 –wcl528.0 >>VC.log".

Run VC.bat along with FMP-VC.bat and you should be able to decode DMR audio.

Figure 5 is a screen capture containing several windows of information from my test transmission. If you look at the "DSD+ DMR VC Channel Activity" window you can see my target was Talk Group 99. The signal came from my radio programmed with 3112746, my DMR-MARC registered radio number. "The DSD+ VC event log" also reflects this same information. The "DSD..." window shows a trace of the audio signal. The most useful information comes from the command prompt script, lower left window. You can also see DSDPlus initially locked on my signal before getting regular information. This is because my simplex mode has an "always" admit criteria. If we tweak our script to use slot two, the information for my signal would look the same in the command prompt, but no audio would play. This is because DSDPlus is expecting time slot two to provide the audio. We can also see my radio ID and the talk group in this window.

Test Radio Setup

Testing was done on low power and simplex. I programmed my Tytera MD-380 to use the standard Talk Group 99, Time Slot 1, and Color Code 1, as found on the Amateur Radio guide on the DMR-MARC web page.[6] I also have the channel admit criteria set to "always" since there is no signal with which to sync up.

Results

The setup easily decodes my DMR test signals from my MD-380. Figure 5, shows that DSDPlus is successfully decoding the incoming DMR voice packets. The software successfully identifies key packet information, such as talk group, device ID, and so on. The audio output is clear and easy to understand. The software saves a ".wav file" of the audio, enabling me to mute the speakers while testing to avoid feedback.

With the setup working correctly, I have configured an old computer with the RTL-SDR to act as a DMR monitor for W4DFU repeater. I used the time slot selection of DSD+ to limit monitoring to time slot 2 (to avoid hearing the near constant traffic on North America or World Wide talk groups). This lets me hear local traffic and just a few larger area talk groups. The W4DFU repeater is part of the K4USD network. Details for time slots and talk groups can be found at the K4USD web page.[7] I live a few miles from the W4DFU repeater and the small stock antenna that came with the RTL-SDR is easily able to pick up the repeater from my desk.

Next Steps

Like many hams, I am always learning something new. I love the challenge of mastering a new technology. I am far from an expert on DMR, but I am having fun discovering the features of this digital mode. This SDR scanner project is just one way to gain a better understanding of DMR while also enabling a way to connect with other users.

This tutorial was just the tip of the iceberg in terms of software defined radio. While scanning DMR with DSDPlus is nothing new, the software can be intimidating. We have just scratched the surface in what DSDPlus can do and hopefully this will encourage you to experiment with all the features or get started with DMR. Most of the DSDPlus documentation is in text files with the software. Give it a try and see what you can come up with!

Andy Milluzzi, KK4LWR, is an Amateur Extra licensed ham, first licensed in 2012. He is president of the Gator Amateur Radio Club, W4DFU, at the University of Florida. Andy is 2012 alumnus, receiving a BS in Computer Engineering and a BS in Software Engineering, of the Rose-Hulman Institute of Technology in Terre Haute, Indiana. He is a PhD candidate and 2013 alumnus, receiving a MS in Electrical and Computer Engineering, at the University of Florida in Gainesville, Florida. Andy loves how Amateur Radio affords him the ability to tinker and relax, while still incorporating his passion for engineering.

Notes

[1] **www.dsdplus.com/**
[2] **zadig.akeo.ie/**
[3] **vb-audio.pagesperso-orange.fr/**
[4] **www.dsdplus.com/download-2/**
[5] **airspy.com/download/**
[6] **www.dmr-marc.net/media/Amateur_Radio_Guide_to_DMR_Rev_I_20150510.pdf**
[7] **k4usd.org**