Digital modes in the field?

It's easy as Pi

12/20/2017

"PRAIRIE PORTABLE" OPERATIONS

IN THIS ISSUE

Digital Modes with RPi3

by Shawn Morgan - WØFW

Hello guys, my name is Shawn Morgan. My callsign is WØFW. I am a cattle rancher living on the prairie in southwestern South Dakota. We raise about 175 head of black Angus cows on roughly 4000 acres. Our ranch boarders the Buffalo Gap National Grasslands which is nearly 560,000 acres. It offers some beautiful scenery to look at while operating portable.

While ranching keeps me busy, I try to set aside a few hours each day for "playing radio". I have a modest shack in the house that consists of a Flex 3000 SDR transceiver and a GAP Challenger DX, a 31-foot vertical that covers 2 meters through 80 meters. I also run an Alinco DR-B185 2-meter base radio with a Hustler G7, a 17-foot VHF only vertical.

I have been wanting to put a field portable station together for quite a while now. My goal was to keep my "shack in a sack" small enough to be able to backpack long distances on foot. My goals include SOTA (Summits On The Air) activations and Emergency Field Communications.

I also want to do long duration field deployments (2-5 days) and packing supplies for such an outing requires a small, lightweight station, especially when you are on foot.

I sold off some unused ham radio equipment to finance my portable station. I am currently running a fully loaded Elecraft KX3 with the PX3 panadapter. My station is powered by a LiFePo4 10Ah battery, or the KX3's internal batteries if I am traveling light.

Since weight is a concern, I decided to forego the bulky laptop for running digital mode software. My original plan was to carry a 10-inch Android tablet, because I had recently purchased one. This is lightweight and relatively small and fit into a compartment in my backpack very nicely.

I went to the Google Play store and downloaded every available app for digital modes and rig control. Well, it didn't take me long to figure out that the digital mode options were few and most of the apps just plain sucked. I gave each one a serious try from the field. Most were unstable, froze up or crashed repeatedly.

The entire experience was very frustrating.

"Shack in a Sack"

My entire portable station fits into this backpack. It's not all that big.... about the size of a decent bug out bag.

Vilros Raspberry Pi 3 Essentials Kit $69

Since my portable station budget is somewhat limited, I did not want to purchase a new laptop to run the digital modes. Aside from the expense, there just wasn't any room in the backpack for the laptop.

While researching my options, I saw an ad for the Raspberry Pi and that got me going for a couple of reasons. One, I am a bit of a tech geek and wanted to play with a Pi just to see what they were all about, and secondly the processing power of the new Raspberry Pi 3 was impressive. A Quad-Core 1.2 GHz processor with 1 GB of RAM. As a bonus, the new Raspberry Pi 3 also comes equipped with onboard Wi-Fi and Bluetooth connectivity.

I explored digital mode software that was compatible with Linux since the Raspberry Pi 3 runs the Raspbian Stretch operating system, which is a Debian derivative. I was quite surprised at all the available software.

This got me to thinking that this could possibly be a great alternative to a bulky laptop and it was cheap. A RPi3 board can be had for around $35 USD.

I ordered the Vilros Raspberry Pi 3 Essentials Kit from Amazon for $69 USD. The kit comes with the RPi3, Samsung 32 GB Evo Plus (Class 10) Micro SD card preloaded with NOOBS. It also included a 2.5-amp USB power supply, a 5-foot HDMI cable and nice plastic case for the Pi.

When my kit arrived, I set to work on building this into the ultimate field portable digital mode machine. My first thought during the unboxing was, while the included plastic case for the RPi3 was nice, it wouldn't do battle in the field very well. The case was a press fit with no screws holding anything in place. I had visions of dropping it and seeing the case fly apart and my new Pi covered in dust, or worse.....remember I am a rancher.

I did some searching on eBay and found a black anodized aluminum case that fit the Raspberry Pi. I thought this would not only add protection when operating out in the field, but may also provide some RF shielding, not that that is a huge problem with a QRP station., but nevertheless. It also looks pretty good sitting next to the KX3.

The Rpi3 comes completely built from the factory. It is about the size of a credit card.

The SD card is loaded with NOOBS which is just a fancy interface for installing an operating system on your new Pi. It also included the Raspbian Stretch operating system along with several other things that I would never use. NOOBS makes the entire process of installing the OS very simple. I am not a very experienced Linux user, although I have loaded several Linux based operating systems onto older laptops and netbooks and dabbled with it a bit.

Once the OS was loaded onto the Pi, I started researching where to find the software packages that I wanted to install for digital modes. Turns out that FLDigi was easy. Raspbian Stretch uses terminal commands like:

sudo apt-get update

sudo apt-get install fldigi

The FLDigi package could also be located by searching "ham radio" under the "Add / Remove Software" function, but I couldn't figure out where to find the software after I added it, so I used the ***sudo apt-get*** method in the terminal.

I also loaded WSJT-X using this method but after running the program, I discovered that it was an old version. Quite possibly the first public release of WSJT-X. It did not include the new FT8 mode. Since all the cool kids are using FT8 these days, it was essential that I figure out how to install a newer version.

I did a lot of researching on how to make this Raspbian Stretch OS find the current version of WSJT-X.

Here is how I did it:

All install actions are performed in a terminal.

sudo apt-get install dirmngr

sudo apt-key adv --recv-keys --keyserver keyserver.ubuntu.com 862549F9

sudo nano /etc/apt/sources.list

Add the following to the end of sources.list

 # -----
deb
http://ppa.launchpad.net/ki7mt/wsjtx-next/ubuntu
trusty main
#

Ctrl+X to save and exit.

sudo apt-get update
sudo apt-get install wsjtx

This gave me the latest version of WSJT-X which includes the new FT8 mode.

I also added DireWolf which is a virtual TNC program for running APRS. It gives you several modes to choose from. Digipeater, beacon and iGate to name a few. My thought was to possibly use this as a field portable iGate when used in conjunction with my LTE enabled tablet. I thought this might come in handy for keeping tabs on a group during a search and rescue mission. The iGate receives the APRS packets and relays them to the internet so you can track using aprs.fi or similar websites.

I followed this link to get DireWolf loaded on the Raspberry Pi.

https://andrewmemory.wordpress.com/2015/03/22/setting-up-direwolfxastir-on-a-raspberry-pi/

I have also used these instructions for loading DireWolf on an old Netbook that I have set up for an iGate here in the shack. My coverage area is an 80-mile radius and it relays several thousand packets to www.aprs.fi each month.

.

Once all the software was loaded onto the Raspberry Pi, it was time to figure out how to run it out in the field. In the shack, I had a keyboard and mouse plugged into the Pi and the HDMI cable running to one of my two 27-inch monitors. Such luxuries would not be possible out on the wide-open prairie. Did I mention that you can see so far out here that you can watch your dog run away for a week?

Since I already had a shiny new Android tablet that proved to be useless for digital modes using the available apps, I decided to get to work on using it as my keyboard, mouse and monitor.

To make this happen, I enabled VNC Server (Virtual Network Connection). It comes with Raspbian Stretch and can be implemented from the configuration section.

To use the tablet, I downloaded VNC Viewer from the Google Play store. To make this work, you must first start the server using a terminal command. This requires logging into the Pi via SSH (Secure Shell). For this, I downloaded JuiceSSH from the Play store. This app cost a couple bucks but it's money well spent.

To use JuiceSSH, you must create a connection using the IP address of your RPi3. Once the connection is established, you are presented with a terminal, just like the one you used to install WSJT-X. In this terminal, you type"

vncserver

This will start VNC Server on your Pi.

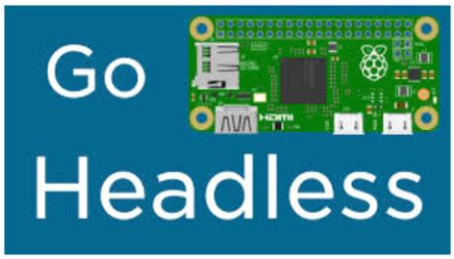
Then type

exit

This command will shut down the terminal and you can exit the JuiceSSH app.

On a side note, this server is supposed to start automatically upon boot up if you enable it in the operating systems configuration section but for some reason I haven't been able to make it work. The research to correct this is still ongoing.

Now that the VNC Server is running on the Pi you can go to VNC Viewer app on your tablet and start it. You must set this up using the Pi's IP address much like you did for the JuiceSSH app.



Once the connection between the tablet and the Pi is made, you are presented with the same screen you would be looking at if the Pi were connected to an actual monitor. On the screen you with see an **X** which is your curser. You navigate the same as you would on a PC, by dragging your cursor wherever you need to go.

So, if you've made it this far, congratulations you now have a **headless Pi.**

The next problem that I needed to solve was connecting the tablet to the Pi **without an internet connection** to simulate a field deployment. It was an easy task on the home Wi-Fi but it's coverage area just barely reaches beyond my yard. I did a fair amount of research on how to accomplish this. I attempted to turn the RPi3 into an Access Point. After 4 failed attempts, which resulted in reloading the operating system and the digital mode software each time, I discovered that the latest Raspbian version, "Stretch" was missing some key components to make this work.

It works perfectly with the slightly older version "Jesse" but unless your extremely proficient with the inner workings of Linux, it's not going to work with Stretch.

Since there is more than one way to slice a Pi, I decided to create the wireless hotspot from my LTE enabled Android tablet. It worked perfectly on the first attempt and it will connect to the Pi with or without cell service.

Now that the field access point issue has been resolved, it was time to make the RPi3 play nice with the KX3.

The first order of business was to get the sound from the Pi to the radio and back. To accomplish this, I used a cheap Sabrent soundcard that I had purchased from Amazon for some earlier experimental project that I had done. It works perfectly and provides great sound quality to both the radio and the software.

The Sabrent soundcard however, is a little fat and it interfered with neighboring USB devices, so I ordered a different cheap Amazon soundcard that had a short cable between the USB connector and the soundcard, but the audio was not usable by the digital mode software. I decided I'd stick with the Sabrent card since I no longer needed a mouse and keyboard plugged in, it wouldn't interfere with anything.

The KX3 is probably the best radio I have ever owned. It's perfectly sized for field portable work.

KX3 CAT control via KXUSB

To make the radio talk with the software, I connected the Elecraft KXUSB cable to the RPi3. The drivers for the USB's chipset were already included in the Raspbian Stretch operating system so no additional work was needed. As soon as the cable was plugged in, the 2 LED lights started blinking so I knew it was making connection to the radio.

With some changes to the configuration in each software program, I had the CAT control of the radio working. I also pointed the audio stream to my USB soundcard and adjusted the levels using the RPi3 sound mixer.

The whole process was simple and trouble-free. Now I am ready to head to the field, so I packed everything up and headed toward the door when suddenly, I felt tugging sensation on my backpack…

The darn AC power cord for the RPi3 was only 5 feet long, no way I am going to make it to the field with that. I don't even have an extension cord that is long enough.

Time to cut the cord….

The RPi3 requires a 5 volt power supply that provides about 2.5 amps of DC power.

I wanted to power the Pi using the same battery that will be powering the field portable station since I was already packing a 10Ah LiFePo4 battery with me.

This problem was solved by adding a UBEC (Universal Battery Eliminating Circuit). I used them back in my radio-controlled rock crawler days to limit the voltage to the 7.2-volt receiver since it was powered by the same 3 cell LiPo battery (11.1 volts) as the crawler.

My rock crawler BEC's were computer programmable and fairly expensive. I didn't feel that I needed anything quite this fancy for the Pi. I ordered $9 USD from Amazon that was built for 5 volts / 3 amps out.

- Output Voltage: 5V / 3A

- Input: 5.5V-26V
- Lightweight and compact
- LED indicator light
- Size: 43x17x7mm, Weight: 11g

The UBEC fits nicely inside the Pi case. I connected the supplied power plug to the correct GPIO pins to power the Pi. After drilling a small hole in the Pi case, I ran the power wires out and attached Anderson Power Pole connectors to them. This will plug directly into my DC power distribution block.

The anodized aluminum case that I bought for the Pi came with a small cooling fan. The fan is probably not needed since I installed the heatsinks on the Pi's board, but I thought what if I were operating in direct sunlight on a hot summer day. It might get warm enough to need some additional cooling. I would hate to see the CPU throttle back and the software hang mid QSO with a once in a lifetime DX station.

When the fan is plugged into the GPIO pins on the Pi, it runs all the time. If the cooling is not needed, then this becomes an unnecessary battery drain.

I decided to solve this problem by wiring in a NPN transistor (SO8050) between the fan and the GPIO pins. This will turn the fan off until a command is sent to power a 3.3v GPIO pin. Once the pin is powered, the fan will run until the command is sent to turn the power off to that pin.

To handle the programming, I turned to Python and typed up a short program that will make the Pi take temperature readings from the CPU chipset. Once the temperature reaches the threshold, the command is sent to power the 3.3v pin, starting the fan. When the temperature has dropped below the threshold, the fan is turned off.

So, there you have it folks, my slice of the Pi.

This whole thing was done as a proof of concept. Now that I have determined that it will work nicely with my field portable station, I will put the finishing touches on the Pi.

Things left to do:

- Tidy up the wiring inside the Pi case.
- Solder longer power leads onto the BEC.
- Permanently attach the BEC to the Pi case.
- Heat shrink and install the NPN transistor and associated wiring. (tested concept from a breadboard)

If this sounds like something useful to you, I encourage you to give it a try. Don't be scared if your unfamiliar with the Raspberry Pi platform or Linux. It was all pretty much a first for me and if I can do it, so can you.

Thanks for sticking with me through this long article. I hope to work you all on the air from my field portable station someday.

73 de WØFW

The last two pictures are field portable ops with the pet antelope and a picture from my favorite operating place on top of the mountain on my ranch. This is where I do multi-day deployments.