WCF Country
West Central Florida Section
Published Quarterly
WCF EXPERIMENTER
Spring 2020

From the Workbench

This issue's thoughts from the Section Manager's workbench

By Darrell Davis, KT4WX
Section Manager - ARRL West Central Florida Section

Welcome to the Spring issue of the WCF EXPERIMENTER. At least it is still technically Spring when we come out with this issue even though in reality it is early summer and the 2020 Atlantic Hurricane Season is due to start a few days from the time that I write this.

Once again we are blessed with really good content for this issue and three of these articles are by local authors. Here are the feature articles for this issue:

- That's One Ugly Microphone - Randy Payne K4EZM
- The Current Status of FT8 on Two Meters and the Higher Bands - Buddy Morgan WB4OMG
- Using your Weather Station With Raspberry PI - Burt Wizeman K4SRQ.
- What's the difference between FDD and TDD? - Lou Freznel W5LEF - A reprint of an article that appears on the Electronic Design website from 2015.

In order for the WCF EXPERIMENTER to be a success it depends upon articles written by you. You may be thinking, "How come you do not have articles on the following topic ________?". We print the articles we receive so if you would like an article on a particular topic, consider writing that article for the WCF EXPERIMENTER. Jim Weslager K3WR, our Assistant Section Manager for Publications work with almost any format. Please send submissions to Jim via email at weslager@gmail.com.

If you know of someone who has written an article for a local club newsletter and you think it would be a good article for the WCF EXPERIMENTER, ask them to submit it for publication in the WCF EXPERIMENTER.

Enjoy this issue, keep experimenting, keep building, and keep writing down your results so others can benefit from your knowledge.

FT8 Update

The Current Status of FT8 on Two Meters and the Higher Bands

By Buddy Morgan, WB4OMG
ARRL Technical Specialist &
Florida Weak Signal Society
Lakeland, FL

This article was originally written for the Southeastern VHF Society Conference, which was canceled due to pandemic issues.

WSJT has been around since 2001. Although Tropo is mentioned, all of the early work seems to be centered around EME. I first noticed WSJT around 2009? I had little interest in “Radio Audio Interfaces”, so it was not until radios with USB Ports became common, that I began to pay real attention to WSJT.

FT8 has been around since 2017. I first saw FT8 in operation, on 6M, a year or so later. It looked pretty good to me. I thought It would have great potential for the higher bands. At Field Day, 2019, I saw a FT 991 connected to a Laptop Computer with a USB cable, running FT8. My interest was finally sparked. I never do anything fast. It took me a few months to implement the WSJT software. I spent four frustrating hours interfacing a radio to a computer. At the conclusion of those four frustrating hours, I made my first FT8 contact, on Six Meters.

There was a Two-Meter frequency, preloaded in the software – 144.174. So, I started to solicit QSOs on Two-Meter FT8, over the Florida Weak Signal Society (FLWSS) Reflector. At first it was kind of like talking to the wall. I seemed to be the only person interested. I persevered. A few people humored me and agreed to work me, on Two-Meter FT8. Over time, more people got on. FT8 seemed to get the tacit approval of FLWSS. After a while, Two-Meter FT8 reached critical mass and be self-sustaining.

Somebody asked me what I did to promote VHF FT8. Really, not a whole lot. In January 2019, I posted to the FLWSS reflector that I wanted to try an FT8 activity night, on Two Meters. Other than that, about all I have done is answer questions and send information to people, when asked.

The Two-Meter frequency – 144.174 MHz would not have been my first choice for an FT8 frequency - it is too high in the band. I felt like there was interference potential with SSB QSOs. It was already preloaded in the software. Changing to another frequency probably would not have worked.

Using .174 on all the higher bands appeared to work. That would keep FT8 out of everybody’s way. In California, 222.174 is in the repeater range of frequencies. So, it would not work on 222, at least not in California. But, on all the higher bands it looked good to me. 432.174 and 1296.174 seem to be pretty well established as FT8, frequencies.

This document was written, in February of 2020. The information contained herein, is current as of then.

Look at PSK Reporter, as of this writing, FT8 is getting about twice the use of the next highest mode (FT4). So, why is FT8, so popular?

One reason is time. You can complete a QSO, in ¼ the time of the one-minute WSJT modes. Another reason is that FT8 is one big party line. You can easily have 20+ stations operating on the same frequency, at the same time.

FT8 was written for Six-Meter Sporadic E Skip. But, it seems to work very well for Tropo, on Two-Meters and the higher bands. In fact, FT8 appears to work just as well or better as JT65 for Troposcatter paths. One last advantage of FT8, it won't heat up your PA, like the "one minute" modes.

So, where are we today, with FT8?

Six Meters: If you have not noticed, FT8 has pretty well taken over the band. CW activity has all but disappeared. The amount of SSB activity is down by at least a factor of 10. Six Meters just doesn't sound like it used to. Is this good or bad? I dunno. But it has happened. All I know is that in my first 49 years of Amateur Radio, I made two contacts with Europeans on Six Meters. These were CW QSOs. The signals were just above the noise. One morning, using FT8, I made three contacts into Europe. There is no going back.

I am going to mention Ten-Meter FT8 activity, because 10M propagation is so similar to Six-Meters. There is FT8 activity on 28.074. There is even local, ground wave activity. Ten-Meter FT8 is showing us that the band is open more often than we thought. I have made 5000-mile QSOs, with 200 watts and a dipole, using FT8, when no SSB or CW signals were heard, on the band. I have heard 6000-mile contacts. I think we are going to find that the same situation exists on Six Meters.

Two Meters and higher in frequency, the crux of my presentation. Right now, FT8 is the dominant digital mode. A few people have tried FT4 and found that it did not work as well as FT8 – which it shouldn't. Actually, the only advantage to FT4 is that it is a 7.5 second cycle. This means that you should be able to make contacts twice as fast as FT8. In an HF or Six Meters Contest situation, this could be decisive. But, on Two Meters, I think not. Actually, FT8 seems to work very well, on Two Meters, 432 and 1296, in a Contest. We will get to that later.

JS8 is another digital mode. The short description is that JS8 is FT8, with some additional information exchange capability. Although JS8 is popular on HF, there is very little activity, on Two Meters and higher in frequency.

A few people report JT65 working better than FT8 for Tropo. One person told me it is 10 db better. But, for most people JT65 doesn't seem to work any better than FT8. Why? JT65 takes four times as long. So that is +6db. But it is close to four times as wide. Minus 6 db. The JT65 CODEC was written for EME. In my opinion, it doesn't handle Tropo as well as the FT8 CODEC. Very few times, have I heard of someone failing to make a Tropo QSO, using FT8 that then make it using JT65. We need to run some more tests.

I think JT9 and JT4 have great promise for Tropo work, on VHF and UHF. Looking at these modes on a Spectrum Analyzer, these modes look similar to FT8. Here again, we need to do some more work.

MSK144, on Six Meters and Two Meters has a good following. Really MSK144 has made a lot of people "casual" Meteor Scatter operators. To the people, that M/S is their prime interest, us casual types are welcome. But, us casual types, just don't give proper reverence to Meteors.

A few people have tried MSK144, for Tropo, with some success. For Tropo, using both time slots, the most stations you could have on 144.150 MHz would be four – two in each QSO. So, MSK144 just is not as social as FT8. MSK144 is great for what it was designed for. But, for Tropo, in my humble opinion, there are better modes.

A typical day, in the (ham radio) life of WB4OMG consists of getting on Two Meters, around 0630. I call CQ on SSB, in multiple directions. Then I get on FT8 and repeat the process. I do the same thing, most evenings. I make about as many contacts on FT8 as I do SSB. We usually have more people on for our Florida Weak Signal Society FT8 activity night, on Two Meters than we do our 432 SSB net. FT8 is not the predominate mode, on Two Meters, like it is on Six Meters. But, when I got on, in the Summer of 2018, there was zero activity, at least in the Metro EL 98 area.

A few days before the January Contest, WA4GPM suggested that we all get on Two Meter FT8, at 8 pm, on Saturday and Sunday, of the contest. It worked! This got everybody on the same frequency, at the same time. Well… Not really on the same frequency, but on the same Waterfall. You could see everybody. We took people "up the bands". When you worked somebody, you sent a message: 432.174 or 1296.174. We did not have to hunt people down. We made contacts faster than we could have on 144.200 or 432.100. I am sure we will do it, again, in June.

Where are we, today? Here in the South East, we get on 144.174 MHz, FT8, every Monday night. The official start time is 2030. People get on as early as Eight pm and stay on, after Ten pm. We started with four stations, on January 21, 2019 – one of which was using a vertical antenna. All these stations were in EL 88 or EL 98.

The next week, W4DEX checked in, from EM 95! Now days we seem to average about 10~12 stations. The most we ever had was 22 stations. Eight stations is a slow night. It started in Florida, but we have stations regularly from Georgia and the Carolinas. Occasionally Alabama and further West. Since January of 2019, I have worked 61 unique stations, using FT8. Most of these contacts have been on our Activity night.

The North Texas Microwave Society sponsors a similar activity. They get on at 2000 Central Time, also on 144.174. The Guys in Texas have done even better, than FLWSS, with 83 unique callsigns, heard on their FT8 night. They also get on 222.174 at 1900 Central Time, on Tuesdays and 432.174 at 1900 Central Time, on Wednesdays.

My most memorable Two Meter FT8 contact was W5LUA, EM 13, 933 miles, around Ten pm Eastern Time, May 27, 2019. We decoded at -18. Other than the FLWSS Beacon, 35 miles, away, Al was the only signal on the band. No beacons from the West - nothing. Called CQ, on SSB and FT8, to no avail. I will always wonder as to the propagation mode. Best guess is that we had our own personal duct.

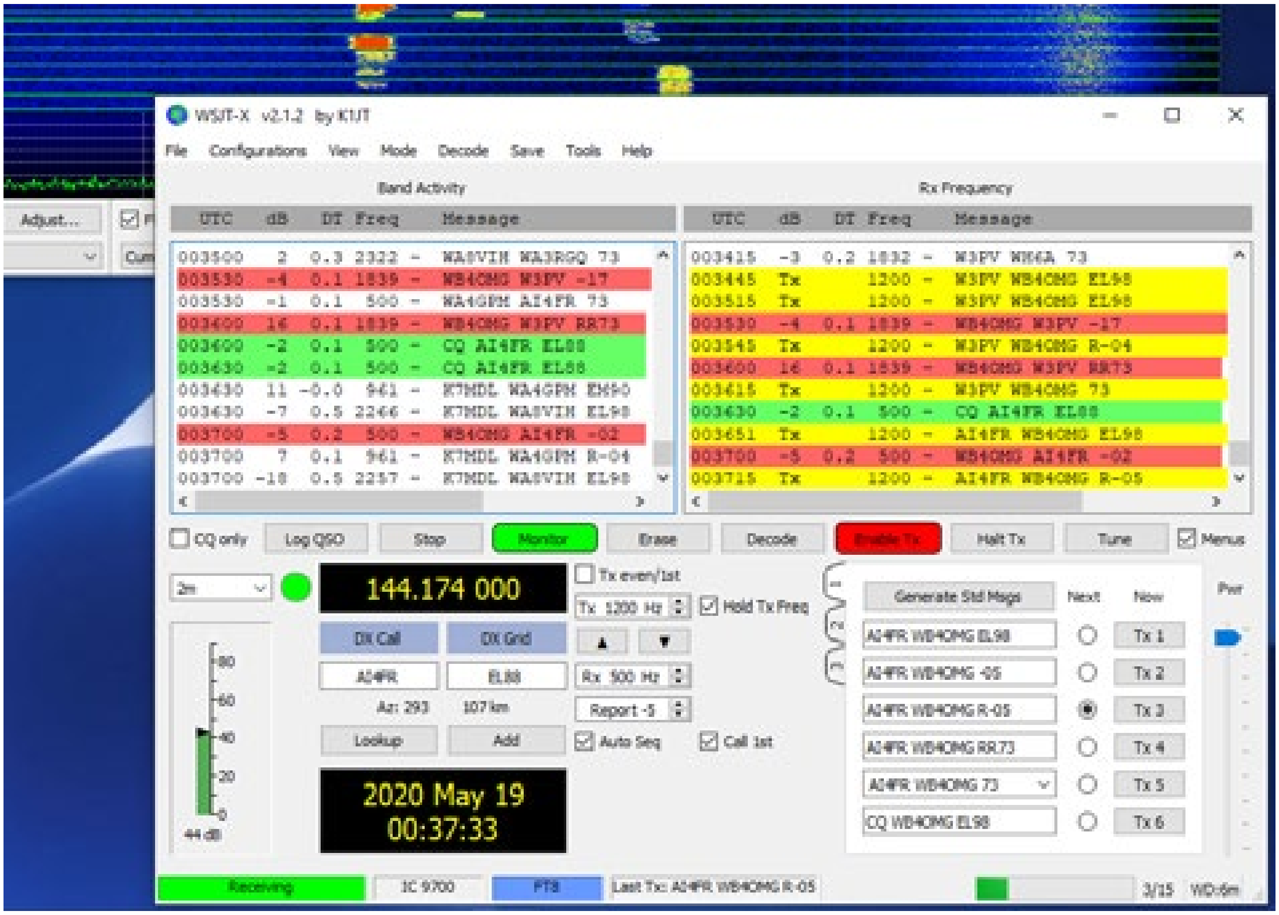
Can you communicate over longer distances using FT8? Under dead band conditions, most certainly. With FT8, modest stations can communicate over distances that, using SSB would take a legal limit amplifier. For a 50 watt and a small yagi station, 400+ mile QSOs are pretty much an every-day affair, using FT8.

During periods of Tropospheric enhancement, sometimes you can stretch the opening, a bit. One night last summer, everybody in Central Florida was working into Virginia, on SSB. Using FT8, I worked Philadelphia. When you hear un-workable SSB signals, down in the noise, you can make contacts, using FT8.

Skipping to the end of the story: On the bands Two Meters and higher, you can work with much weaker signals, using the WSJT software, than you can on SSB or CW. People are having fun. There is more fun to be had.

So, where are we headed? My crystal ball is in the repair shop, but I think we will see more digital activity. Will FT4, take over during contests? It could on Six Meters. Somehow I don't see it happening on Two Meters and up. Contests on Two Meters just are not that fast paced. Will JS8 catch on? Maybe. Being able to chat has a lot of appeal to some people. Will we wind up with six competing digital modes, like in the repeater part of the bands? Could happen. Will people make more use of the "one minute" modes for Tropo? I think so. There is one thing that is certain: things will change!

If you live in the Southeastern United States, join us for FT8 Activity Night, 144.174 MHz, Mondays at 2030 Eastern Time. After you work everybody on Two Meters, switch to 432.174 or 1296.174. At Nine pm, point towards Texas.



Band Activity

```
  UTC     dB   DT Freq    Message
003500     2  0.3 2322 ~  WA8VIH WA3RGQ 73
003530    -4  0.1 1839 ~  WB4OMG W3PV -17
003530    -1  0.1  500 ~  WA4GPM AI4FR 73
003600    16  0.1 1839 ~  WB4OMG W3PV RR73
003600    -2  0.1  500 ~  CQ AI4FR EL88
003630    -2  0.1  500 ~  CQ AI4FR EL88
003630    11 -0.0  961 ~  K7MDL WA4GPM EM90
003630    -7  0.5 2246 ~  K7MDL WA8VIH EL98
003700    -5  0.2  500 ~  WB4OMG AI4FR -02
003700     7  0.1  961 ~  K7MDL WA4GPM R-04
003700   -18  0.5 2257 ~  K7MDL WA8VIH EL98
```

Rx Frequency

```
  UTC     dB   DT Freq    Message
003415    -3  0.2 1832 ~  W3PV WK4A 73
003445    Tx      1200 ~  W3PV WB4OMG EL98
003515    Tx      1200 ~  W3PV WB4OMG EL98
003530    -4  0.1 1839 ~  WB4OMG W3PV -17
003545    Tx      1200 ~  W3PV WB4OMG R-04
003600    16  0.1 1839 ~  WB4OMG W3PV RR73
003615    Tx      1200 ~  W3PV WB4OMG 73
003630    -2  0.1  500 ~  CQ AI4FR EL88
003651    Tx      1200 ~  AI4FR WB4OMG EL98
003700    -5  0.2  500 ~  WB4OMG AI4FR -02
003715    Tx      1200 ~  AI4FR WB4OMG R-05
```

2m

144.174 000

☐ Tx even/1st

Tx 1200 Hz ☑ Hold Tx Freq

AI4FR EL88

Az: 293 107 km

Rx 500 Hz

Report -5

☑ Auto Seq ☑ Call 1st

2020 May 19
00:37:33

Message	Next	Now
AI4FR WB4OMG EL98	○	Tx 1
AI4FR WB4OMG -05	○	Tx 2
AI4FR WB4OMG R-05	●	Tx 3
AI4FR WB4OMG RR73	○	Tx 4
AI4FR WB4OMG 73	○	Tx 5
CQ WB4OMG EL98	○	Tx 6

Receiving | IC 9700 | FT8 | Last Tx: AI4FR WB4OMG R-05 | 3/15 WD:6m

That’s One Ugly Microphone

Crafting a Boom Mike for a Drake TR-7 Transceiver

By Randy Payne, K4EZM
ARRL Assistant Section Manager &
ARRL Technical Specialist

I’m a fan of Drake TR-7 transceivers. The first one I bought worked briefly after going through an expensive repair session. I bought another one that is doing well for me. That put me in the market for a suitable microphone to go with it. I went through several before finally obtaining a Shure 444 desk mic that works well with the Drake. However, I wanted to get a boom mic and use a foot PTT switch to give me hands free operation.

I had an old magnifying lamp mounted on a boom that could be used for the microphone as per the QST article by Geoff Haynes NIGY (“A Homebrewed Shock-Mounted Boom Microphone,” QST, February, 2015, p.33).

The following steps describe how I built the microphone.

1. I first tried to buy an inexpensive microphone for the project. I bought several microphones at hamfests and our local thrift stores, but none of them worked properly. I bought one on eBay, but again it did not work well. I finally found another Shure 444 buried away in one of the multitude of boxes that clutter my work room floor and shelves. I was able to get the cartridge out of it. I tested it and found it worked well.

2. Following Geoff’s ideas, I dug in my box of PVC pipes and fittings (I never throw any of these away) and found the pieces of pipe needed for mounting the cartridge. The pictures show how one piece of pipe is mounted to the boom. A second piece, suitable in size to hold the mike cartridge, was threaded on one end and I had the fitting that threaded onto this. The cartridge fit like it was made to go there.

3. I found a cap for this pipe and drilled a hole in it for the mike cord to go through. After putting the cartridge into the pipe and running the mike cord through the cap, I stuffed in some cotton balls. I then screwed on the fitting that secured the cartridge in the pipe.

4. The piece that mounts to the boom is actually two caps glued together and the closed ends are cut off. I would have used a suitable piece of PVC pipe, but there was not one in my box of PVC pieces. The completed microphone is shock-mounted in this larger piece that is mounted on the boom. Like Geoff, I used hair bands on small clips to do this. This part of the mike will be replaced with a piece of pipe with larger diameter when I can get it. As is, there is really little or no shock proofing.

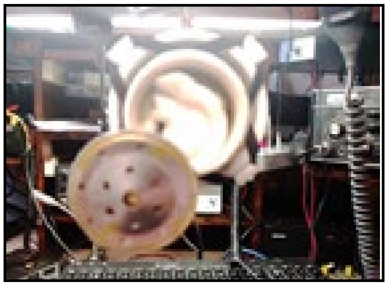
5. As a finishing touch, I took the wind screen from the 444, put some material in it to help prevent wind

blast, and taped it onto the pipe that holds the mike cartridge. I now had a working microphone.

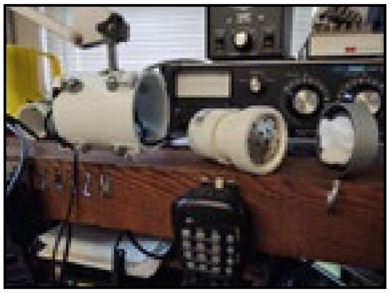
6. I used a small metal box as a junction box for the microphone, a foot switch and a cord to the TR-7 microphone jack. I mounted three jacks on the box for these. I also mounted a PTT switch on the box as a backup to the foot switch. Having the mike plug enables easy testing of other microphones.

This has worked very satisfactorily for me. I can now have both hands available during my sessions as net control. A final step will be painting the mike and the boom mount so they look better.

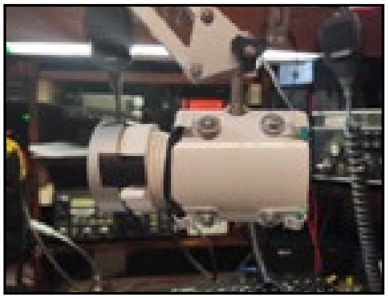
Total cost, not counting money spent of mikes that did not work (total was around $10 for these), was $2.00 for the hair bands! All of the other parts came from junk boxes. Maybe it pays to save!



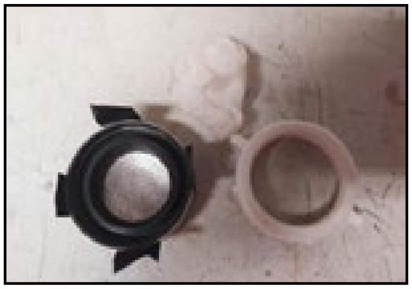
Here is the Shure 444 cartridge ready to be mounted in the pipe that is used for the mike body. The material I used to deaden sound waves behind the cartridge has been put into the mike body.



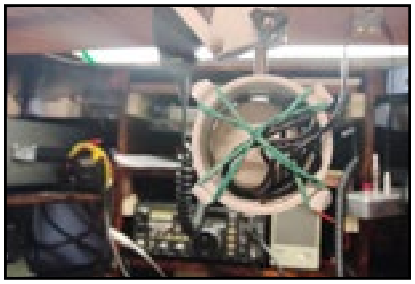
Here is the mike prior to taping on the front cover and mounting it in the boom mount. You can see how the 444 cartridge fits in the front of the mike body and the mike cover from the 444.



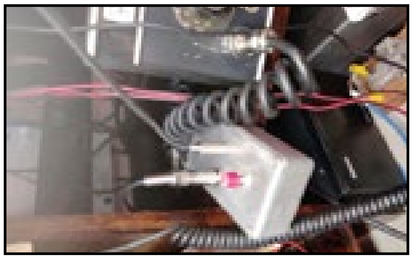
The mike in the boom shock mount. The clips hold the hair bands that provide a measure of shock mounting for the mike.



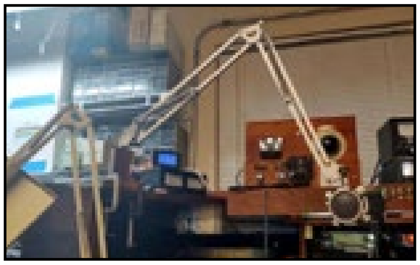
Here is the front cover from the Shure 444. The four pieces of tape are used to secure this cover to the PVC fitting that screws on the front of my boom mike. I later wrapped tape around the joint to better secure the cover.



Rear view of the mike mounted in the boom holder.



This is the junction box with a PTT switch. The coiled cable connects the junction box to the TR-7.



The completed mike in place and ready to use. The clamp holding the boom on the desk will be replaced with screws.

Using Your Weather Station With Raspberry Pi

A Very Affordable Way to Take Your Weather Station

by Burt Wizeman, K4SRQ
ARRL Technical Specialist &ARRL Net Manager - WCF Section Technical Net

With the pandemic I am glad I do not need to go out to play! Wanting to dedicate a PC to my weather station, a Peet Bros Ultimeter 2100, was being met with headwinds from the accounting department at K4SRQ. Already I was using my THD-710G to send the packet stream to CWOP (Citizens Weather Observer Program [1],) via findu, a lot of radio for a basic task. Thus, in my stacks of 'things to redo' that are strewn throughout the K4SRQ shack, I eyed the Raspberry Pi. I wonder if....

I needed:
- CWOP, Weather Underground, PWS.com (Personal Weather Station) Uploads at the least.
- Local Display of the Current Output
- Ability to upload to a webserver.
- Nearly FREE

After researching, installing, configuring, and testing several apps, I found WeeWx to be the best available in the Linux domain.
From the Weewx Website the application is :

a free, open source, software program, written in Python, which interacts with your weather station to produce graphs, reports, and HTML pages. It can optionally publish to weather sites or web servers. It uses modern software concepts, making it simple, robust, and easy to extend. It includes extensive documentation. [2, p. Keffer & Wall]

Weewx is a most excellent application written in Python by Tom Keffer during the winter of 2008-2009 as a sever or daemon to run his weather station. It is designed to be simple to install and configure. Yet, you can get as deep as you want with the workings of the application. WeeWx supports all of the most popular and many of the obscure manufacturers of weather stations.

Getting Started

Assumptions
- You are somewhat familiar with Linux
- You know how to acquire a Raspberry Pi
- You have a weather station able to provide serial data output.
- You can open a Terminal in Linux
- Home directory is **`/home/pi`**
- You know that **`<RETURN>`** means to press the Return or Enter Key on your keyboard

You will need
- Your weather station
- Raspberry Pi 3 or better (Older models work but things are moving ahead.)
- A Micro SD Card
- Serial or Serial to USB Cable for your weather station

- A power source for your Pi
- The baud, parity, and bits that your weather station uses to output data. [3]
- What the name of the serial port is. If you are using a Serial to USB converter the serial name will be **ttyUSB<number>** probably ttyUSB0. [4]

The Process

1. Install Raspbian (a Linux Distribution)
2. Update Raspbian
3. Install minicom (For testing serial communications)
4. Install Python 3.8.3 or later.
5. Install WeeWx
6. Configure WeeWx

Install Raspbian

You will need to install Linux on the Raspberry Pi. The Distribution is called Raspbian. For this you will need a micro SD Card that you can use on your PC. You will most likely use an adapter. Information on how to install Raspbian, can be found at:

https://www.raspberrypi.org/documentation/installation/installing-images/

You will want to update your install of Linux before we start. At the command prompt type:

```
apt-get update<RETURN>
```

Install minicom

You can use this to test your serial connection for troubleshooting.

```
sudo apt-get install minicom<RETURN>
sudo minicom -s<RETURN>
```

Once minicom is installed you can run it and set your default baud rate to that of your weather station to verify the data is flowing from your weather station. In the context of our install, it is simply a troubleshooting tool.

1. When minicom is loaded, typing Ctrl+A O will bring up the Configuration screen.

a. Select Serial port setup Press A for the Serial Device. If you have a standard serial port, start with **/dev/tty0**. For USB to Serial Interface it will be **/dev/ttyUSB**<number> which is usually **/dev/ttyUSB0**. Standard serial ports usually follow a series:

COM1: = tty0
COM2: = tty1
COM3: = tty2. You get the idea.

b. Press E and set the Baud, Data Bits, Stop Bits and Parity for your weather station. The Ultimeter 2100 is 2400 N 8 1.

c. Press Escape when done and select 'Save setup as dfl'

d. Select Exit when saved and you should see data flying across the screen.

Your data may be human readable text or it may be in hex. Refer to your weather station's manual on what data to expect so you can verify proper configuration of the serial port.

Install Python 3.8.3 or later.

WeeWx 4.0, the latest release, uses Python 3.8.

You will need to install Python 3.8.3 (it is the latest at the time of this writing.) and tell the operating system where to look for the Python source code and

WeeWx.

Though all information is covered here, there is a video as well [5] .

All programs and tools have Dependencies. These are pieces of software that are required for the application to run. Python 3.8.3 has several Dependencies that first need to be installed before we continue. Type the following on the command line.

```
sudo apt-get install -y build-essential tk-dev libncurses5-dev libncursesw5-dev libreadline6-dev libdb5.3-dev libgdbm-dev libsqlite3-dev libssl-dev libbz2-dev libexpat1-dev liblzma-dev zlib1g-dev libffi-dev tar wget vim <RETURN>
```

Once these commands run, it would be a good time to invite a family member into the shack as the data will be flying across the screen and you will look impressively smart!

Now that the Dependencies have been installed it is time to install Python 3.8.3.

Like the Store on Microsoft Windows® or the App Store from Apple®, Linux software installs are kept in Repositories, a fancy name for a store. You must tell your operating system where to look for these items. If you get an error, you may need to jump to superuser, do this by prefacing the command with sudo Type the following on the command line.

```
cd /home/pi <RETURN>
wget https://www.python.org/ftp/python/3.8.3/Python-3.8.3.tgz<RETURN>
```

THIS WILL TAKE A LONG TIME. Maybe up to 20 minutes or so. If it seems like it is hanging up, WAIT!
You can substitute the version part – **3.8.3** for the latest version.

With Python 3.8.3 in your home directory you will now begin the installation process! First Python is converted to computer readable code then you create the install. So here we go, type the following:

```
sudo tar zxf Python-3.8.0.tgz<RETURN>
cd Python-3.8.0<RETURN>
sudo ./configure --enable-optimizations<RETURN>
sudo make -j 4<RETURN>
sudo make altinstall<RETURN>
```

Each of these commands will take some time. Once finished, you will have Python 3.8.3 working on your computer. There is one last step to take to make it your default version of Python 3.8.3 when you run Python programs:

```
echo "alias python=/usr/local/bin/python3.8" >> ~/.bashrc<RETURN>
```

Now when you simply type python your operating system will default to running Python 3.8.3.
Let's get WeeWx installed!

Install WeeWx

As I mentioned earlier, a Repository in the Linux domain is a place where source code and application installs are kept. *Advanced Package Tool* or **apt** as it is called when on the command line, is a tool in Linux to install applications. **apt** interacts with the low level system tool *Debian Package Manager* or **dpkg** to be sure all the dependencies are available, downloading them if not and passing this info to the **dpkg** app for install.

First we have to tell Raspbian where the store is and that we can trust the source of the files, that the 'store' is safe to visit:

```
wget -qO - http://weewx.com/keys.html | sudo apt-key add -<RETURN>
```

In human terms this is saying "get the security key from weewx.com and pass it to Advanced Package Tool to be added to the database of authenticated sources"

To be on the side of caution, duplicate this effort by typing the following:

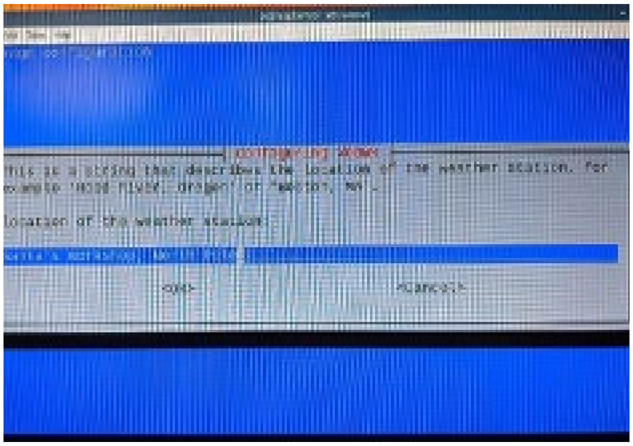
```
wget -qO - http://weewx.com/apt/weewx-python3.list | sudo tee /etc/apt/sources.list.d/weewx.list<RETURN>
```

Now it is time my friend to again update Raspbian and then install WeeWx

```
sudo apt-get update<RETURN>
sudo apt-get install weewx<RETURN>
```

Configure WeeWx

Once WeeWx installs, a screen like the one below prompting for specific information will come up. The first prompt is for the name of your weather station. Once entered, press Tab to move to select <Ok>



```
Configuring weewx
This is a string that describes the location of the weather station, for
example 'Hood River, Oregon' or 'Boston, MA'.

location of the weather station:

<Ok>                                    <Cancel>
```

Next you will need to enter your latitude and longitude. I had an accurate result with

http://www.latlong.net

WeeWx wants the decimal version as provided by latlong.net.

Next you will provide your weather station's altitude above sea level. You may need to check

https://whataltitude.com/

Enter that in either foot or meter. *Don't use 'feet' by mistake, use 'foot'.* Your altitude is important for accurate barometer readings. It is surprising what goes into calculating air pressure. Here is some info to help understand this measurement:

https://www.ndbc.noaa.gov/educate/pressure.shtml

After entering your altitude, you will be prompted for the Units, either Metric or US.

Your next option is to choose your weather station.

Next is choosing your serial port. Using **minicom** you should already have this information

Once chosen, the application will continue the install and write the data you just entered to the **/etc/weewx/weewx.conf** file.

Before we continue to configure WeeWx, I want to point out a few locations that you will get to know by heart. WeeWx keeps configuration files in the directory

/etc/weewx/

Logs are kept in
/var/log/

Configuring WeeWx for your install is done through a text file found at
/etc/weewx/weewx.conf
You might have a different text editor but I prefer **nano** it is easy to use. When changes are made to the configuration file, WeeWx will need to be restarted. To start and stop WeeWx type the following:

To stop:

```
sudo /etc/init.d/weewx stop<RETURN>
```

To Start:

```
sudo /etc/init.d/weewx start<RETURN>
```

To monitor the impact of changes or to verify the function of your install, you will want to use the Linux **tail** command. This command takes the latest entries to a file and displays them on a screen or to another device (think file, drive, other app, ect.) Tail is a great tool to use for seeing changes that occur in a log file.

With this basic information, lets get this app configured. From the command line type:

```
sudo /etc/init.d/weewx stop<RETURN>
```

Using nano to edit the configuration file:

```
sudo nano /etc/weewx/weewx.conf<RETURN>
```

The first item to change will be under the General configuration information section. Under

```
#Set to 1 for extra debug info, otherwise comment it out or set to zero
```

Enter this setting:

```
Debug=2
```

This will set your debug level to more information so that it will be easier to see what is happening if a problem exists. This debug info is written to

/var/log/syslog

As you scroll through the file you will see the entries you just made during install. You will need to first register with CWOP to get your id if you do not have one. You can register at:

http://www.wxqa.com/SIGN-UP.html

To enable the upload to CWOP scroll to the section
[[CWOP]]

```
enable=true
station=<your CWOP station name>
post_interval = 600
#this is the number of seconds between posts
passcode= -1 WeeWx 4.0
```

Follow the same under [[PWSweather]] for your PWS login information. You can register with PWS at:

https://www.pwsweather.com/

There is also a section for Weather Underground:
[[Wunderground]]

The options are self-explanatory.

To configure WeeWx for sending to your website, look under

[[FTP]]

Follow the instructions for uploading to your website. The images used for the meters to display your weather data can be modified. That is outside the scope of this How-To but information is available via the WeeWx website.

Where is The Data?

Data from your weather station is stored in a SQL database to be used by WeeWx. The data is first read from your weather station, written to the database then the data is massaged to be published for the different services you setup in weewx.conf.

In WeeWx you will have full access to the database. Refer to the documentation for further info. Normally you will not have a need to go to the database. The database is located at

```
/var/lib/weewx/
```

How do I know if the data is moving?
How do I know if it is working?

Type

```
sudo tail -f /var/log/syslog
```

Watch as the info is displayed. Success and Failure will be displayed. You should be running an NTP daemon on your Raspberry Pi. This keeps the time accurate on your Raspberry Pi. Timing is important for the uploads.

I also suggest that you setup the Apache2 web server onto your Raspberry Pi. Once installed, you can see your weather station readings from the default meters provided by WeeWx.

http://localhost/weewx

should launch the WeeWx weather page on your server.

I suggest you read the excellent documentation that is provided by the WeeWx team. The complete documentation (some of the best I have seen for any Linux application) is online to read at:

www.weewx.com/docs/userguide.htm

That will get you started. Please read it before digging too deep into the application.

For support there is an excellent forum, monitored by the author of the application and is found at:

https://groups.google.com/forum/#!forum/weewx-user

In the future I hope to setup the Raspberry Pi to send the APRS weather packet via a budget HT to the local APRS network and sending to CWOP via findu. That is the next WX project.

The data you publish to CWOP will go further up the line and be used as part of the overall weather data input for the National Weather Service. The more weather stations we have on the ground, the more accurate the data and forecasting. In Florida, we think our weather is predictable and mundane. It is not. Our weather is quite challenging due to the many variables that can impact a peninsula. By having an accurate weather station, you can help with the accuracy of our forecasts, higher resolution of current conditions and might just be a small part in saving lives. Not bad for Amateurs!

Bibliography

[1] K. Rus Chadwick, "APRSWXNET - the Automatic Position Reporting System as a WX Network," [Online]. Available: http://www.wxqa.com/. [Accessed 14 05 2020].

[2] T. K. &. M. Wall, "index," 2020. [Online]. Available: http://www.weewx.com.

[3] W. Lou Frenzel, "What's The Difference Between Bit Rate And Baud Rate?," WCF Experimenter Winter 2020, 2020.

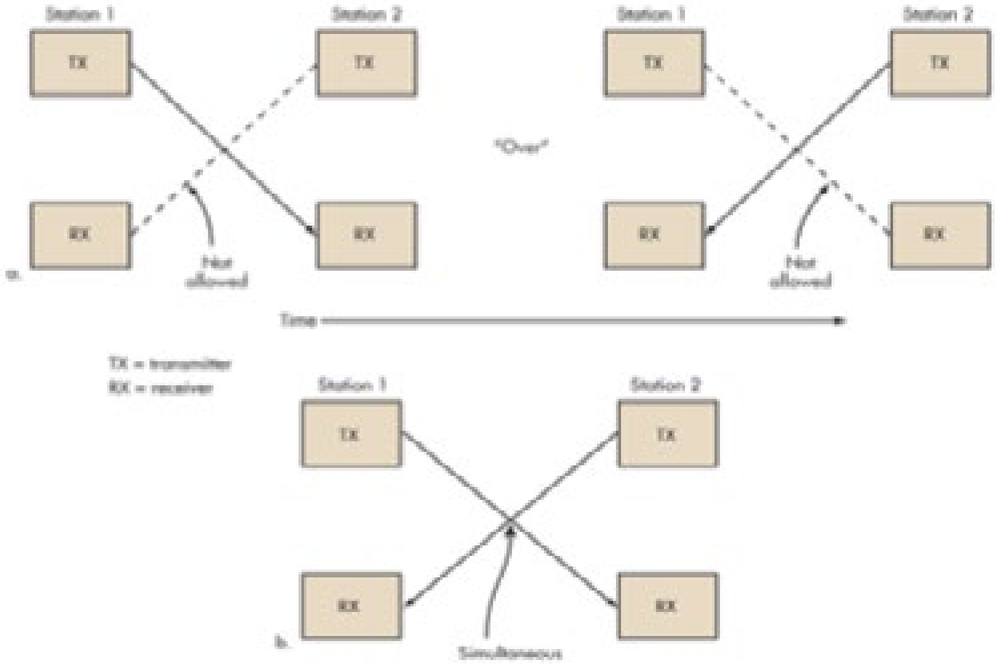
[4] S. Shovon, "List USB Devices Linux," linuxhint, 2018. [Online]. Available: https://linuxhint.com/list-usb-devices-linux/. [Accessed 20 5 2020].

[5] C. Kapoor, "Install Python 3.8 on Raspberry PI (Raspian)," 23 April 2020. [Online]. Available: https://youtu.be/iDqyzVCpk5Y. [Accessed 20 May 2020].

Raspberry Pi

What’s The Difference Between FDD And TDD?

By Lou Frenzel, W5LEF
Reprint from Electronic Design

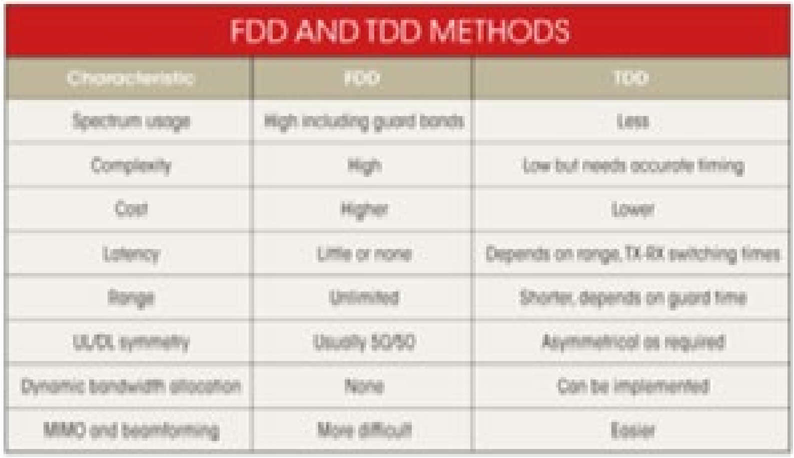


1. There are two modes of duplex operation: (a) half duplex and (b) full duplex.

Duplexing is the process of achieving two-way communications over a communications channel. It takes two forms: half duplex and full duplex (*Fig.* 1).

In half duplex, the two communicating parties take turns transmitting over a shared channel. Two-way radios work this way. As one party talks, the other listens. Speaking parties often say “Over” to indicate that they’re finished and it’s time for the other party to speak. In networking, a single cable is shared as the two computers communicating take turns sending and receiving data.

Full duplex refers to simultaneous two-way communications. The two communicating stations can send and receive at the same time. Landline telephones and cell phones work this way. Some forms of networking per-



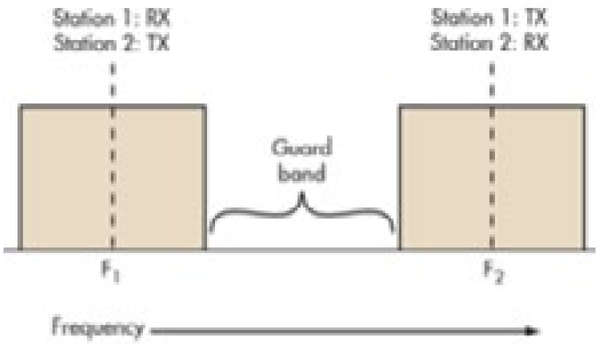
FDD AND TDD METHODS

Characteristic	FDD	TDD
Spectrum usage	High including guard bands	Less
Complexity	High	Low but needs accurate timing
Cost	Higher	Lower
Latency	Little or none	Depends on range, TX-RX switching times
Range	Unlimited	Shorter, depends on guard time
UL/DL symmetry	Usually 50/50	Asymmetrical as required
Dynamic bandwidth allocation	None	Can be implemented
MIMO and beamforming	More difficult	Easier

mit simultaneous transmit and receive operations to occur. This is the more desirable form of duplexing, but it is more complex and expensive than half duplexing. There are two basic forms of full duplexing: frequency division duplex (FDD) and time division duplex (TDD) (*see the table*).

Frequency Division Duplex

FDD requires two separate communications channels. In networking, there are two cables. Full duplex Ethernet uses two twisted pairs inside the CAT5 cable for simultaneous send and receive operations.



2. FDD requires two symmetrical segments of spectrum for the uplink and downlink channels

Wireless systems need two separate frequency bands or channels (Fig. 2). A sufficient amount of guard band separates the two bands so the transmitter and receiver don't interfere with one another. Good filtering or duplexers and possibly shielding are a must to ensure the transmitter does not desensitize the adjacent receiver.

In a cell phone with a transmitter and receiver operating simultaneously within such close proximity, the receiver must filter out as much of the transmitter signal as possible. The greater the spectrum separation, the more effective the filters.

FDD uses lots of frequency spectrum, though, generally at least twice the spectrum needed by TDD. In addition, there must be adequate spectrum separation between the transmit and receive channels. These so called guard bands aren't usable, so they're wasteful. Given the scarcity and expense of spectrum, these are real disadvantages.

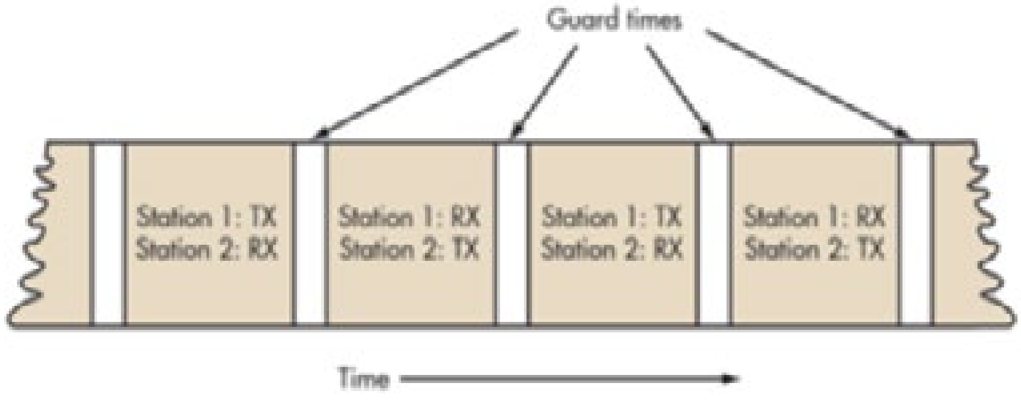
However, FDD is very widely used in cellular telephone systems, such as the widely used GSM system. In some systems the 25MHz band from 869 to 894 MHz is used as the downlink (DL) spectrum from the cell site tower to the handset, and the 25MHz band from 824 to 849 MHz is used as the uplink (UL) spectrum from the handset to cell site.

Another disadvantage with FDD is the difficulty of using special antenna techniques like multiple-input multiple-output (MIMO) and beamforming. These technologies are a core part of the new LongTerm Evolution (LTE) 4G cell phone strategies for increasing data rates. It is difficult to make antenna bandwidths broad enough to cover both sets of spectrum. More complex dynamic tuning circuitry is required.

FDD also works on a cable where transmit and receive channels are given different parts of the cable spectrum, as in cable TV systems. Again, filters are used to keep the channels separate.

Time Division Duplex

TDD uses a single frequency band for both transmit and receive. Then it shares that band by assigning alternating time slots to transmit and receive operations (Fig. 3). The information to be transmitted– whether it's voice, video, or computer data–is in serial binary format. Each time slot may be 1 byte long or could be a frame of multiple bytes.



3. TDD alternates the transmissions and reception of data over time. Time slots may be variable in length

Because of the high-speed nature of the data, the communicating parties cannot tell that the transmissions are intermittent. The transmissions are concurrent rather than simultaneous. For digital voice converted back to analog, no one can tell it isn't full duplex.

In some TDD systems, the alternating time slots are of the same duration or have equal DL and UL times. However, the system doesn't have to be 50/50 symmetrical. The system can be asymmetrical as required.

For instance, in Internet access, download times are usually much longer than upload times so more or fewer frame time slots are assigned as needed. Some TDD formats offer dynamic bandwidth allocation where timeslot numbers or durations are changed on the fly as required.

WThe real advantage of TDD is that it only needs a single channel of frequency spectrum. Furthermore, no spectrum-wasteful guard bands or channel separations are needed. The downside is that successful implementation of TDD needs a very precise timing and synchronization system at both the transmitter and receiver to make sure time slots don't overlap or otherwise interfere with one another.

Timing is often synched to precise GPS--derived atomic clock standards. Guard times are also needed between time slots to prevent overlap. This time is generally equal to the send-receive turnaround time (transmit-receive switching time) and any transmission delays (latency) over the communications path.

Application Examples

Most cellphone systems use FDD. The newer LTE and 4G systems use FDD. Cable TV systems are fully FDD.

Most wireless data transmissions are TDD. WiMAX and WiFi use TDD. So does Bluetooth when piconets are deployed. ZigBee is TDD. Most digital cordless telephones use TDD. Because of the spectrum shortage and expense, TDD is also being adopted in some cellular systems, such as China's TDSCDMA and TDLTE systems. Other TDLTE cellular systems are expected to be deployed where spectrum shortages occur.

Conclusion

TDD appears to be the better overall choice, but FDD is far more widely implemented because of prior frequency spectrum assignments and earlier technologies. FDD will continue to dominate the cellular business for now. Yet as spectrum becomes more costly and scarce, TDD will become more widely adopted as spectrum is reallocated and re-purposed.

Related

What's The Difference Between GaAs And GaN RF Power Amplifiers?

What's The Difference Between Satellite Radio And HD Radio?

What's The Difference Between Bluetooth Low Energy And ANT?

What's The Difference Between Cable And DSL Broadband Access?

Lou Frenzel writes articles and blogs on the wireless, communications and networking sectors for Electronic Design. Formerly, Lou was professor and department head at Austin Community College where he taught electronics for 5 years and occasionally teaches an Adjunct Professor. Lou has 25+ years experience in the electronics industry. He held VP positions at Heathkit and McGraw Hill. He holds a bachelor's degree from the University of Houston and a master's degree from the University of Maryland. He is the author of 20 books on computer and electronics subjects.

Lou Frenzel was born in Galveston, Texas and currently lives with his wife Joan in Austin, Texas. He is a long-time amateur radio operator (W5LEF).