KG5E

Four Days in May℠

CONFERENCE PROCEEDINGS

May 19, 2016
Fairborn, Ohio
USA

QRP Amateur Radio Club International

Four Days in May

CONFERENCE PROCEEDINGS

May 19, 2016

— Table of Contents —

*** Advance copies not available for publication*

Four Days In May
2016: 21 Years of QRP Fun at FDIM!

Welcome to FDIM!

This year is the 21st annual Four Days in May QRP Symposium! On behalf of QRP Amateur Radio Club International (QRP ARCI), welcome to FDIM. Once again, QRPers and homebrew enthusiasts from around the world are gathering in the Dayton, Ohio area to learn, have fun, share ideas and talk about their experiences with other like-minded hams.

Our Thursday seminar speakers will help you appreciate and understand a wide range of topics—operating, construction, measurement and repair, using both analog and digital technology. When you add the Buildathon, Vendor Night, Club Night, Show 'n Tell and the grand closing banquet, it's easy to understand why so many QRPers tell us that FDIM is the highlight of the year!

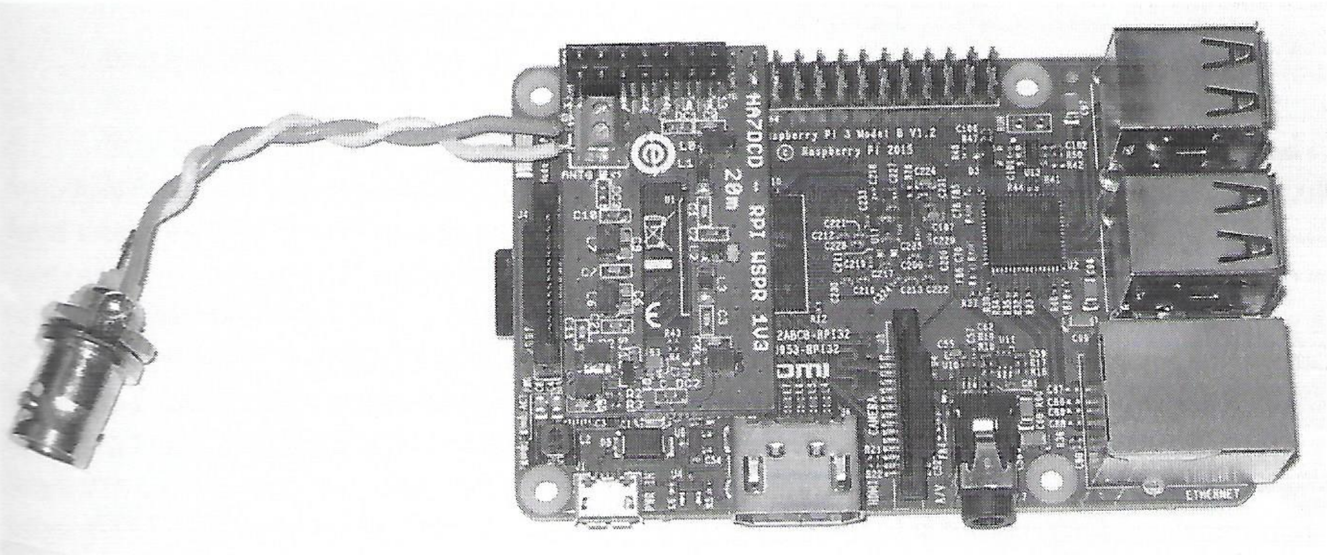
On behalf of everyone on the FDIM 2016 team, and the members of the QRP ARCI Board of Directors, we hope you enjoy this year's Four Days in May!

Steve Fletcher, G4GXL
President, QRP ARCI

Preston Douglas, WJ2V
Vice President, QRP ARCI

WSPR and the Raspberry Pi

by Scotty Cowling WA2DFI



Raspberry Pi version 3 with HA7DCD's QRPi TX shield for 20M from TAPR

WSPR stands for **W**eak **S**ignal **P**ropagation **R**eporter, and is pronounced "whisper". WSPR uses the MEPT-JT mode of transmission (Manned Experimental Propagation Transmitter—Joe Taylor) developed by (you guessed it!) Joe Taylor, K1JT. WSPR is designed to test propagation paths on the MF and HF bands using low-power beacons. While WSPR is technically just the software used to send and receive the MEPT-JT transmissions, things get a lot more interesting when you point your computer browser to WSPRnet, or the **W**eak **S**ignal **P**ropagation **R**eporter **net**work. WSPRnet is the result when you link hundreds of WSPR beacon stations together with the Internet.

Technical Stuff

Each WSPR transmission takes approximately 2 minutes, and contains your callsign, Maidenhead grid locator and your transmitter power in dBm. (Example: "WA2DFI DM43AJ 20") This is all compressed down into 50 bits (binary digits) of data. The keying rate is 1.4648 baud and the signal occupies a bandwidth of about 6 Hz. This is why it takes almost 2 minutes to send one transmission. The benefit, however, is that the receiver decoders can successfully decode a signal at about –28 dB S/N ratio (measured in a 2500 Hz bandwidth). Note that this is *minus* 28 dB, i.e. 28 dB below the noise.

Why would we want to do this? How about these reasons for starters:

- Real time propagation path testing between multiple points
- What bands are open (i.e., what is the MUF)?
- What is my antenna pattern?

While all the technical details are beyond this simple presentation, here are a few links to get you more information before we dive into the nuts and bolts of putting a WSPR station on the air, which we will cover in detail:

- Joe Taylor’s WSPR page: physics.princeton.edu/pulsar/K1JT/wspr.html
- WSPR wiki: en.wikipedia.org/wiki/WSPR_(amateur_radio_software)
- WSPRnet website: wsprnet.org

What Do You Need?

So, what do you need to get on the air with your own beacon? A quick and easy way is with a Raspberry Pi single board computer. “But wait,” you say, “that runs Linux and I don’'t know anything about how to do that!” Never fear, it is easy and you will not have to learn hardly any Linux. The whole process is way easier than learning the menu system on your new 2M HT.

The RPi (as it is known), along with the TAPR QRPi board, will allow you to transmit, but not to receive. You will need to connect to WSPRnet on the Internet to see worldwide spots. To get on the air, you will need the following:

wifi Built-in

- Rasperry Pi version 2 or 3: mcmelectronics.com
- USB power supply (or USB port on a PC) and micro USB cable
- 4GB or larger micro-SD card (8 GB or larger class 10 is best)
- TAPR QRPi TX shield for 20M: tapr.org/kits_20M-wspr-pi.html
- HDMI monitor, USB keyboard and USB mouse
- 20M antenna of some kind
- PC running Windows or Linux that can write a micro-SD card

The first three in the list above can be purchased at Hamvention from MCM Electronics in booths SA03078-SA0311. The TAPR QRPi can be purchased at the TAPR booths BA0451-BA0454. You will likely have a monitor, USB keyboard and mouse around the shack somewhere. What, you don’t have spare monitors (that’s plural)? Visit the flea market for some quality shopping time!

Now that the hardware is out of the way, what do you need in the way of software? Here is the list:

- Ubuntu Mate 16.04 beta for RPi 2 and RPi 3 image: ubuntu-mate.org/raspberry-pi
- 7-zip file archive: 7-zip.org
- Rufus USB utility: rufus.akeo.ie
- WSPR application: github.com/JamesP6000/WsprryPi.git

The 7-zip archive software and the Rufus USB utility are only needed if you are going to program the bootable micro-SD card on a Windows machine. I will also show you how to get the Linux tools to build the SD-card if you want to build it on a Linux machine instead. All of the software is free and downloadable using the links in the above list. Now that we have all of the software and hardware, let’s put it all together and get on the air! You will only have to download the first three ahead of time; the fourth we will download with Ubuntu Mate itself, after we install it.

The Big Picture

Here is quick list of what we need to do:

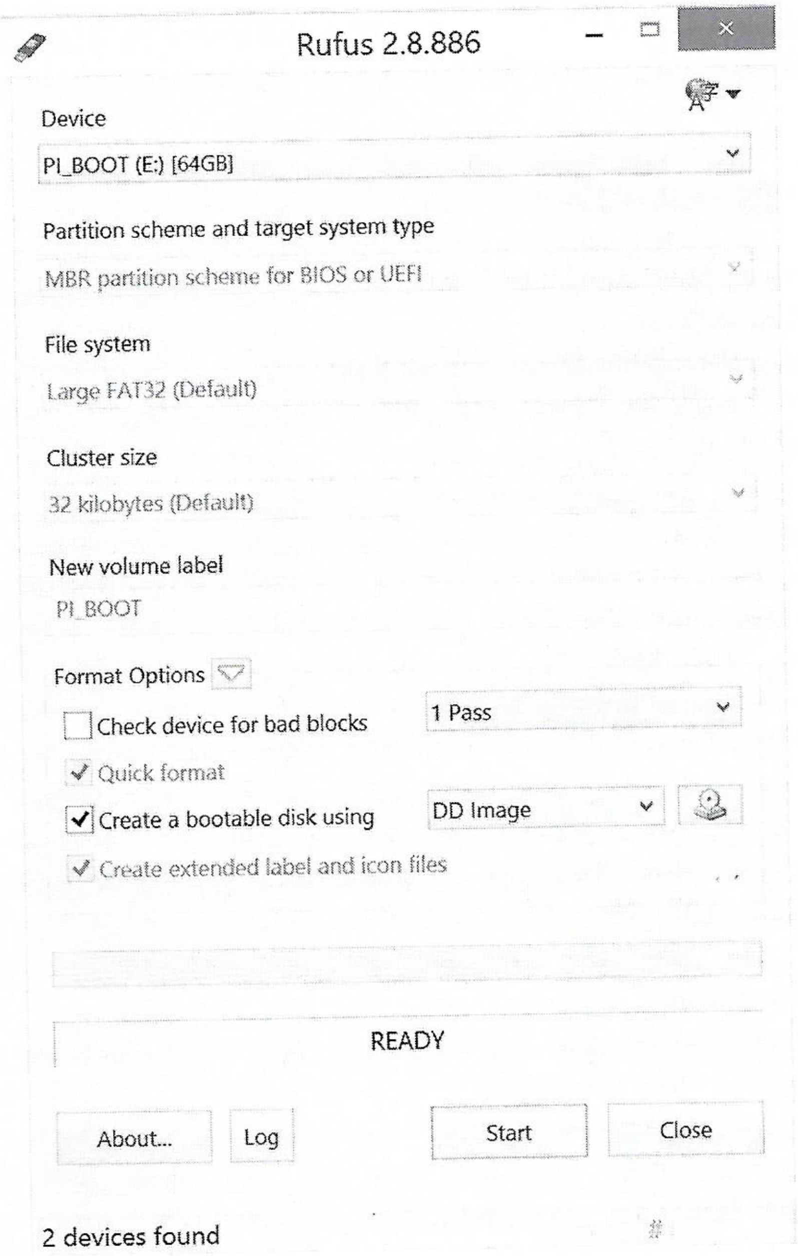
1. Program the micro-SD card with a bootable image
2. Hook up all the hardware
3. Boot Linux from the micro-SD card and set it up
4. Download and compile the WSPR application
5. Run the WSPR application

Making a Bootable SD Card

There are two ways to make a bootable card: one on a Windows system and one on a Linux system. You can find these instructions on the Ubuntu Mate download page (ubuntu-mate.org/raspberry-pi) for both operating systems. If you have a Linux system already, you likely already know how to do this. Since this presentation is geared towards Windows users, I will focus on the Windows procedure and leave the Linux procedure for you to follow on the web page. (I will show you the Linux commands to make a bootable micro-SD card in the next section just for completeness.) Note that this PC (Windows or Linux) is only used to program the micro-SD card once, just to get the bootable image onto the card. After the micro-SD boots, we will not need the PC any more (except maybe to visit WSPRnet, but you can use the RPi system to do that, too.)

SD Card For Windows Users

First, download the Ubuntu MATE 16.04 beta 2 from one of the mirror sites (I used the Canadian site). Make sure you get the 16.04 beta 2 version (there are two versions). Use 7-zip to uncompress it and then use Rufus to write the image to the card. It is about 3.9 GB, and even though a 4 GB card will theoretically work, I recommend an 8 GB card as a minimum. After you start Rufus, *make sure that you select your SD card first, before you do anything else!* All data will be erased on the drive named in the Device box at the top of the screen. Next select Large FAT32 (Default) under File system and DD image in the drop down box next to the small picture of the CD. Click on the CD picture to browse for your Linux MATE image file (it will have a .img extension). You should have Quick format, Create a bootable disk using and Create extended label and icon files all checked. You can also enter a new volume label. I used "PI_BOOT" for my volume name. Click Start to program the image to the SD card. It will take a couple of minutes to program your card.



Rufus 2.8.886

Device

PI_BOOT (E:) [64GB]

Partition scheme and target system type

MBR partition scheme for BIOS or UEFI

File system

Large FAT32 (Default)

Cluster size

32 kilobytes (Default)

New volume label

PI_BOOT

Format Options

☐ Check device for bad blocks — 1 Pass

☑ Quick format

☑ Create a bootable disk using — DD Image

☑ Create extended label and icon files

READY

2 devices found

Rufus set up to make a bootable micro-SD card

SC Card For Linux Users

If you are reading the section, I assume that you are an experienced Linux user, so I will not offer much in the way of explanation for these commands. If you are a Windows user, you can skip this section. Use gparted to remove all the existing partitions from the micro-SD card before installing the image. This will result in one large group of unallocated blocks on the card.

Assuming that the card is on /dev/sdb, execute the following commands:

20dbm ≈ 100mW

14.097100

```
sudo apt-get install gddrescue xz-utils
unxz ubuntu-mate-16.04-beta2-desktop-armhf-raspberry-pi.img.xz
sudo ddrescue -D —force ubuntu-mate-16.04-beta2-desktop-armhf-raspberry-pi.img /dev/sdb
```

Hooking It Up and Booting Linux

Plug the TAPR QRPi board onto the RPi pin header. Connect your 20M antenna to the QRPi screw terminals (observe polarity marked on the board). Plug the HDMI monitor, USB mouse and keyboard into the RPi. Insert your newly programmed micro-SD card into the slot on the bottom of the RPi board. Supply power to the micro-USB port and the RPi will start its boot sequence. Congratulations, you are now an official Linux user! Uh-oh, not so fast; we need to set up the system during the first boot. Here is how to do it.

1. Select your language, then click Continue
2. Choose your WiFi Access Point, then click Continue (note: you need WiFi to download WSPR, so do not skip this step)
3. Choose your time zone, then click Continue
4. Choose your keyboard (default is usually fine), then click Continue
5. Enter your name, PC name (I used WSPR), username and password (I used my call for both)
6. Check the Login Automatically box, then click Continue
7. If you get the low disk space warning, ignore it for now.
8. Be patient, it will eventually boot to the Welcome screen

We have one more thing to do before we download and run the WSPR software. We need to re-size the file system to use the entire micro-SD card. You paid for those bytes, so why not use every one? Click the large Raspberry Pi Information button on the Welcome screen. Then click the Resize button and reboot the RPi. Hint: to reboot, use the power button in the very upper right corner of the screen and select Restart.

Download and Compile WSPR

Open a terminal window (type CTRL-ALT-T all at the same time) and type the following commands:

```
sudo apt-get update
sudo apt-get dist-upgrade
sudo apt-get install git
git clone https://github.com/JamesP6000/WsprryPi.git
cd WsprryPi
make
```

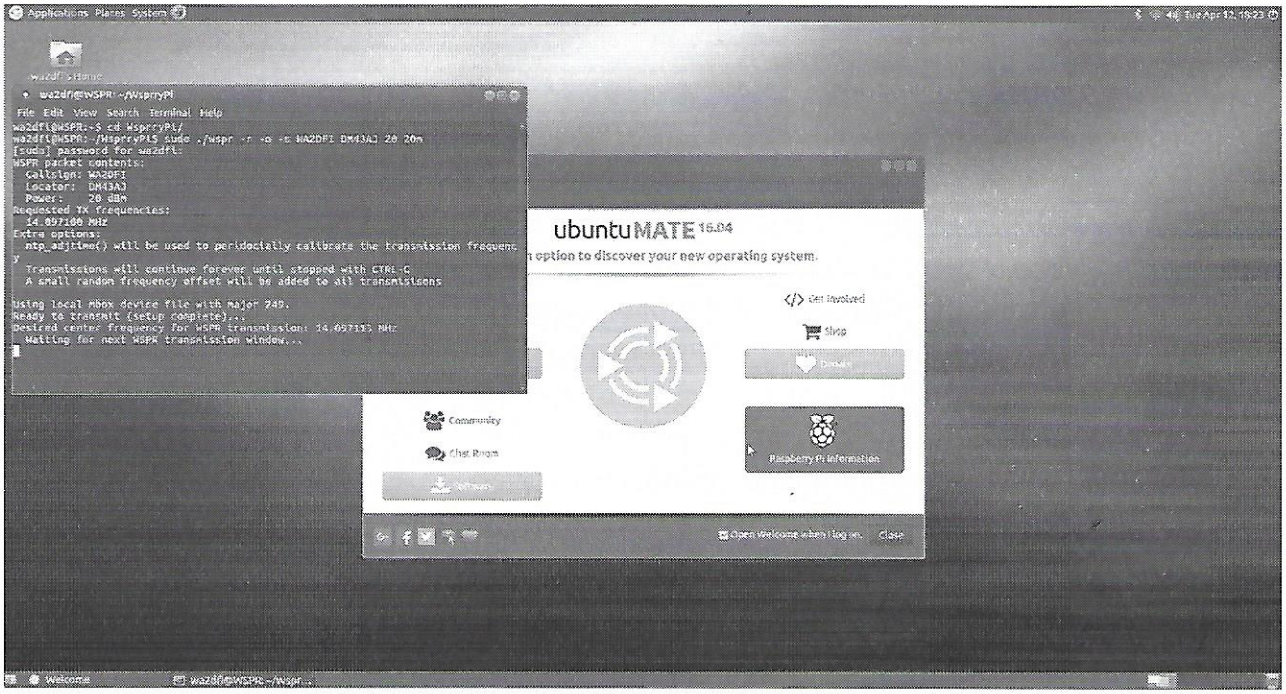
This will compile the WSPR code so you will be able to run it. You will have to enter your password when asked. Answer any questions with the default answers. Note that some of these steps take some time to complete.

Run It!

In the same terminal window, substituting your call for callsign and your 6-digit grid square (e.g., DM43AJ is mine in AZ) for grid, type:

```
sudo ./wspr -r -o -s callsign grid 20 20m
```

Don't be too impatient! Remember, it takes 2 minutes to transmit, and the beacons are time-slotted using the computer clock. Give it some time before you head out to HRO for a new antenna!



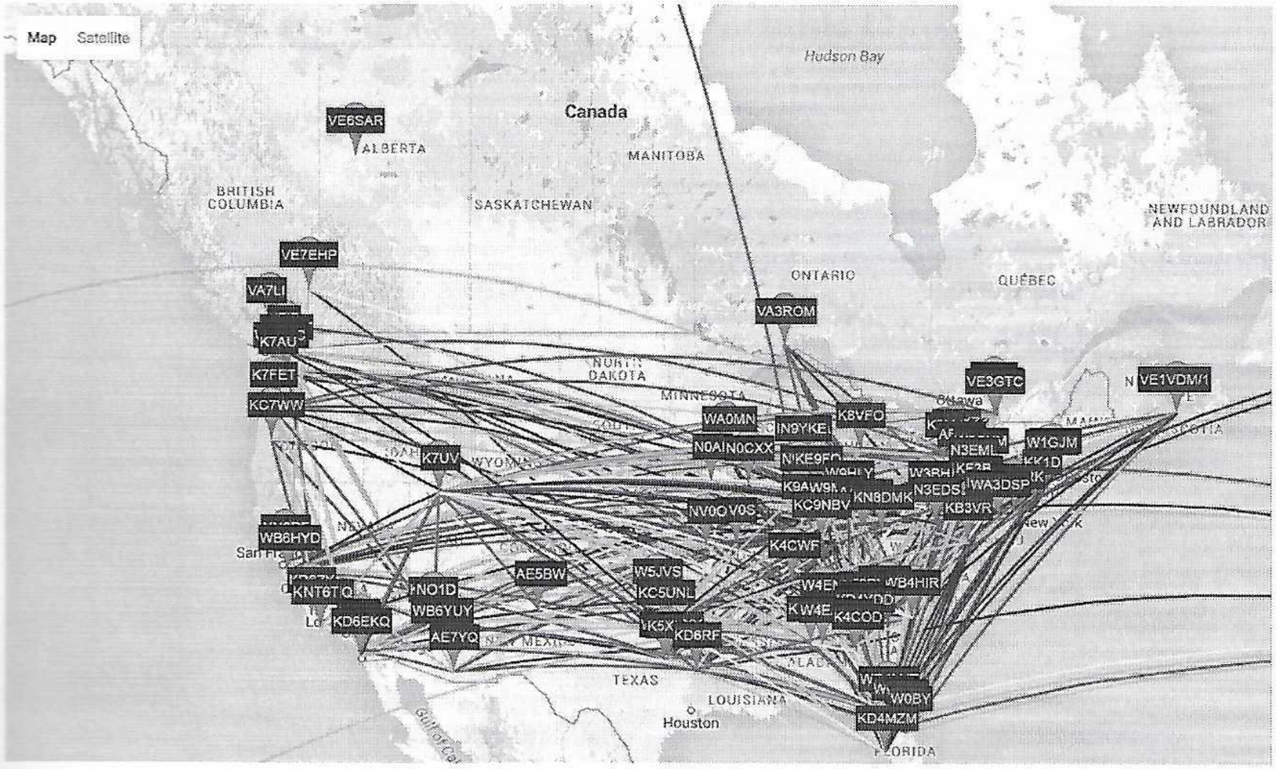
```
wa2dfi@WSPR:~$ cd WsprryPi/
wa2dfi@WSPR:~/WsprryPi$ sudo ./wspr -r -o -s WA2DFI DM43AJ 20 20m
[sudo] password for wa2dfi:
WSPR packet contents:
  Callsign: WA2DFI
  Locator:  DM43AJ
  Power:    20 dBm
Requested TX frequencies:
  14.097100 MHz
Extra options:
  ntp_adjtime() will be used to peridocially calibrate the transmission frequenc
y
  Transmissions will continue forever until stopped with CTRL-C
  A small random frequency offset will be added to all transmisisons

Using local mbox device file with major 249.
Ready to transmit (setup complete)...
Desired center frequency for WSPR transmission: 14.097115 MHz
  Waiting for next WSPR transmission window...
```

Linux MATE running WSPR on the Raspberry Pi

Watching the network

Open a browser window (either on the RPi or another machine) and take a look at wsprnet.org to see what you can see. You can show maps with stations connected by lines (Map tab), you can show tabulated lists of stations by band (Activity tab), you can query a database of stations by band, call, time, etc (Database tab) and more. There is also a forum for discussions and a place to ask questions (Forum tab). Join up and have fun!

Welcome to WSPR and the ranks of Linux users!



WSPRnet.org web site showing propagation paths