The VK4PK Whispering Pi - v1.5

Table of Contents

CHAPTER ONE - Hardware
1. Weak Signal Propagation Reporter (WSPR)
2. Purchasing the Hardware
3. Early Hardware Prototyping and Design
4. QR Pi 20M Raspberry Pi QRP TX Shield Design
5. References

CHAPTER TWO - Installation
1. The Whispering Pi On The Bench
2. Building The Enclosure
3. Fabricating-The-Roof-Bracket
4. Accessing The Roof
5. The 45 Degree Slope
6. The Long Decent. It's A Long Way Down
7. The Bracket Ready to Mount On The Roof
8. Bracket Mounted On The Roof
9. The Storm threatens
10. Antenna Fitted to Roof Bracket
11. QR Pi with Cover Removed
12. The Backside Holds The USB Power Supply
13. QR Pi with Cover Fitted
14. 12 Volt Power and Network Data To The Antenna
15. The Re-purposed Station Master Antenna
16. The Finished Installation
17. The Patch to the Network
18. Surge Protection
19. Is Radio Frequency Isolation Required?

CHAPTER THREE - Software Setup
1. Why do this WSPR project?
2. QR Pi and WSPR - What Do they Stand For?
3. Downloading and Installing The WsprryPi Software Application
4. What Frequencies Am I Transmitting on?
5. Why is my location not accurate?
6. (Mis)Using WSPR fields to encode data
7. Enable ssh on Raspberry Pi
8. Installing System Daemon Service (systemd)
9. Backup Backup Backup
10. Manipulating the qrpi Service
11. The WSPR Protocol
12. Reporting Maps On The Internet
13. Other Applications
14. What Have I Learned?
15. References

CHAPTER FOUR - The Importance of Time
1. Why is super accurate time important?
2. The Jitter Bug
3. NTP or GPS?

Appendices
1. Appendix A - The essential for good decode
2. Publishing Details
CHAPTER ONE - The Whispering Pi Hardware

The Raspberry Pi 2B+ with the HA7DCD add on board for WSPR operation

1.1 Weak Signal Propagation Reporter (WSPR) [Top]

Notes:
- Pronounced "Whisper"
- QRP quasi-beacon mode
- 2-minute TX,
- Randomized Transmit/Rx cycles
- Example Message "VK4PK QG62 20” Note Mainhead Locater is limited to 4 digits (the "qi" is missing from the maidenhead QG62qi)
- 4-FSK modulation, Bandwidth 6Hz
- Spots can be uploaded to WSPRnet.org
- Approx 500 stations participating per day
- Over 200 million spots recorded since 2008

1.2 Purchasing the Hardware [Top]

The HA7DCD add on board to Raspberry Pi is sold by Tucson Amateur Packet Radio Corp. It cost me $29.00 USD and $17.00 postage and took two weeks to arrive.
The TAPR Website for purchasing the Raspberry Pi QRP TX Shield for WSPR on 20 Meters is: https://www.tapr.org/kits_20M-wspr-pi.html

1.3 Early Hardware Prototyping and Design [Top]

QRPi shield was inspired by Joe Taylor's WsprryPi open source program. There were several predecessors. Earlier versions were a little crude.

Early iterations of the WSPR Hardware
1.4 QR Pi 20M Raspberry Pi QRP TX Shield Design  [Top]

The design consists of a Buffer and Band Pass Filter BPF to the RF output from pin (GPIO 4) of the RPi to filter harmonics, an FET amplifier stage and following that, the Low Pass Filter (LPF), and finally an ESD suppressor.

Circuit Schematic with and without functional groups
Stage 1 - Buffer stage to protect the BCM2835 SOC’s clock generator output

Stage 2 - Band Pass Filter
The five section Butterworth band pass filter was developed with reference to "Band Pass Filters for HF Transceivers" by Lew Gordon, K4VX

Stage 3 - A Single FET Power Amplifier

The MMBF170 N-Channel Enhanced Mode MOSFET achieves a 10 dB gain which delivers +20 dBm output power at the end of the Low Pass Filter.

BF170 N Channel FET Amplifier

![BF170 N Channel FET Amplifier](image)

Stage 4 - Chebyshev Filter Low Pass Filter

A brief description of the filter design. The Chebyshev response is a mathematical strategy for achieving a faster roll off by allowing ripple in the frequency response. The mathematical characteristics are derived from Chebyshev polynomials.

Chebyshev Filter in a Cauer topology or an unbalanced ladder topology

![Chebyshev Filter in a Cauer topology or an unbalanced ladder topology](image)

<table>
<thead>
<tr>
<th>Band</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>L1/L3</th>
<th>L2</th>
<th>Toroid</th>
</tr>
</thead>
<tbody>
<tr>
<td>2200m</td>
<td>2.2n/10n</td>
<td>4.7n/22n</td>
<td>4.7n/22n</td>
<td>2.2n/10n</td>
<td>105 (54uH)</td>
<td>105 (54uH)</td>
<td>T50-2  (red)</td>
</tr>
<tr>
<td>600m</td>
<td>2.2n/2.2n</td>
<td>10n</td>
<td>10n</td>
<td>2.2n/2.2n</td>
<td>64 (20uH)</td>
<td>70 (24uH)</td>
<td>T50-2  (red)</td>
</tr>
<tr>
<td>160m</td>
<td>820</td>
<td>2200</td>
<td>2200</td>
<td>820</td>
<td>30 (4.4uH)</td>
<td>34 (5.61uH)</td>
<td>T50-2  (red)</td>
</tr>
<tr>
<td>80m</td>
<td>470</td>
<td>1200</td>
<td>1200</td>
<td>470</td>
<td>25 (2.42uH)</td>
<td>27 (3.01uH)</td>
<td>T37-2  (red)</td>
</tr>
<tr>
<td>60m</td>
<td>680</td>
<td>1200</td>
<td>1200</td>
<td>680</td>
<td>23 (2.12uH)</td>
<td>24 (2.30uH)</td>
<td>T37-2  (red)</td>
</tr>
<tr>
<td>40m</td>
<td>270</td>
<td>680</td>
<td>680</td>
<td>270</td>
<td>21 (1.38uH)</td>
<td>24 (1.70uH)</td>
<td>T37-6  (yellow)</td>
</tr>
<tr>
<td>30m</td>
<td>270</td>
<td>680</td>
<td>680</td>
<td>270</td>
<td>19 (1.09uH)</td>
<td>20 (1.26uH)</td>
<td>T37-6  (yellow)</td>
</tr>
<tr>
<td>20m</td>
<td>160</td>
<td>390</td>
<td>390</td>
<td>180</td>
<td>16 (773nH)</td>
<td>17 (904nH)</td>
<td>T37-6  (yellow)</td>
</tr>
<tr>
<td>17m</td>
<td>100</td>
<td>270</td>
<td>270</td>
<td>100</td>
<td>13 (548nH)</td>
<td>15 (668nH)</td>
<td>T37-6  (yellow)</td>
</tr>
<tr>
<td>15m</td>
<td>82</td>
<td>220</td>
<td>220</td>
<td>82</td>
<td>12 (444nH)</td>
<td>14 (561nH)</td>
<td>T37-6  (yellow)</td>
</tr>
<tr>
<td>12m</td>
<td>100</td>
<td>220</td>
<td>220</td>
<td>100</td>
<td>12 (438nH)</td>
<td>13 (515nH)</td>
<td>T37-6  (yellow)</td>
</tr>
<tr>
<td>10m</td>
<td>56</td>
<td>150</td>
<td>150</td>
<td>56</td>
<td>10 (303nH)</td>
<td>11 (382nH)</td>
<td>T37-6  (yellow)</td>
</tr>
<tr>
<td>6m</td>
<td>22</td>
<td>82</td>
<td>82</td>
<td>22</td>
<td>7 (165nH)</td>
<td>9 (265nH)</td>
<td>T37-6  (yellow)</td>
</tr>
</tbody>
</table>

Stage 5 - The Final Stage is an ESD suppressor diode added to the antenna terminal.

ESD suppression
1.5 References  [Top]

QRPi Whispering Raspberries Zoltán Dóczi, HA7DCD zoltan@rfsparkling.com

The Finished Installation QRPi software installation guide for different HAM TX modes by Zoltan Doczi, HA7DCD
zoltan@rfsparkling.com
This also includes a description of the other modes the HD7DCD add on board

2.2 Building The Enclosure  [Top]

< https://docs.google.com/document/d/1bLKEDmJvKQegqUwDXEy3V7wQTqupHyg2LkN0dOuPY4/edit

Joe Taylor’s K1JT Program Development Page at his Home at Princeton University
http://physics.princeton.edu/pulsar/K1JT/devel.html

CHAPTER TWO - The Whispering Pi Installation

2.1 The Whispering Pi On The Bench  [Top]
2.2 Building The Enclosure  [Top]

2.3 Fabricating The Roof Bracket  [Top]
2.4 Accessing The Roof  [Top]

2.5 The 45 Degree Slope  [Top]
2.6 The Long Descend. It's a Long Way Down  [Top]

2.7 The Bracket Ready to Mount On The Roof  [Top]
2.8 Bracket Mounted On The Roof  [Top]

2.9 The Storm Threatens  [Top]
2.10 Antenna Fitted to Roof Bracket [Top]

2.11 QRPi with Cover Removed [Top]
2.12 The Backside Holds The USB Power Supply  [Top]

2.13 QRPi with Cover Fitted  [Top]
2.14 12 Volt Power and Network Data To The Antenna  [Top]

2.15 The Re-purposed Station Master Antenna  [Top]
2.16 The Finished Installation  [Top]

2.17 The Patch to the Network  [Top]
2.18 Surge Protection

2.19 Is Radio Frequency Isolation Required?
CHAPTER 3 - The Whispering Pi Software Setup

3.1 Why do this WSPR project? [Top]

Why do this project? Firstly it is inexpensive. About $100.00 of kit and some parts from your spares and you are under way. Moreover, it works fine on older models of Raspberry Pi. I had older models not being used because I upgraded to the faster RP3 models so I could improve streaming and general response times. Secondly, it is QRP! I love to concept of transmitting around the globe with a few hundred milliwatts. Thirdly, it is digital. In my view, that is the future of Amateur Radio. Also, there is lots of technology in the project, which means there is a lot to learn and understand. Lastly, but far from least, there are those lovely maps on WSPRnet.org to show off to your friends.

3.2 QRPI and WSPR - What Do they Stand For? [Top]

The "QRPI" is a play on abbreviations. It is the combination of the mnemonic "QRP", the Q-code for low power and the mnemonic "RPi", for Raspberry Pi. Put them together and you have "QRPI". WSPR stands for Weak Signal Propagation Reporter and is pronounced "whisper". Call it the "Whispering Pi" or "QRPI". The choice is up to you.

3.03 Downloading and Installing The WsprryPi Software Application [Top]

The WsprryPi application application is compatible with the original Raspberry Pi, the Raspberry Pi 2/3, and the Pi Zero. To download and compile the WsprryPi application, login to the Raspberry Pi and open a terminal.

The commands are as follows:

```
cd     # change to home folder
sudo apt-get install git
git clone https://github.com/JamesP6000/WsprryPi.git
cd WsprryPi
make
```

If the make process completes without error then it can be installed to /usr/local/bin:

```
sudo make install
```

If you change your mind or want to install another version, it can also be uninstalled from the Raspberry Pi like this:

```
sudo make uninstall
```

Once wspr is successfully installed it can be run manually from the command line prompt as follows:

```
wspr VK4PK QG62qi 20 20m
```

The full explanation of the WSPR application:

```
Usage:
  wspr [options] callsign locator tx_pwr_dBm f1 <f2> <f3> ...
OR
  wspr [options] --test-tone f

Options:
  -h --help
      Print out this help screen.
  -p --ppm ppm
      Known PPM correction to 19.2MHz RPi nominal crystal frequency.
  -s --self-calibration
      Check NTP before every transmission to obtain the PPM error of the
      crystal (default setting?).
  -t --terminate <n>
      Terminate after <n> transmissions have been completed.
  -o --offset
      Add a random frequency offset to each transmission:
      +/- 80 Hz for WSPR
      +/- 8 Hz for WSPR-15
  -t --test-tone freq
      Simply output a test tone at the specified frequency. Only used
      for debugging and to verify calibration.
  -n --no-delay
      Transmit immediately, do not wait for a WSPR TX window. Used
      for testing only.
```

Frequencies can be specified either as an absolute TX carrier frequency, or using one of the following strings. If a string is used, the transmission will happen in the middle of the WSPR region of the selected band.

```
LF LF-15 MF MF-15 160m 160m-15 80m 60m 40m 30m 20m 17m 15m 12m 10m 6m 4m 2m
```

16/28
Transmission gaps can be created by specifying a TX frequency of 0

Note that 'callsign', 'locator', and 'tx_power_dBm' are simply used to fill in the appropriate fields of the WSPR message. Normally, tx_power_dBm should be 10, representing the signal power coming out of the Pi. Set this value appropriately if you are using an external amplifier.

### 3.4 What Frequencies Am I Transmitting on? [Top]

**Available Transmit frequencies in USB data mode are:**

<table>
<thead>
<tr>
<th>Band</th>
<th>Current</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>12m</td>
<td>------</td>
<td>24.9256</td>
</tr>
<tr>
<td>15m</td>
<td>------</td>
<td>21.0956</td>
</tr>
<tr>
<td>17m</td>
<td>18.103</td>
<td>18.1056</td>
</tr>
<tr>
<td>20m</td>
<td>14.0956</td>
<td>same</td>
</tr>
<tr>
<td>30m</td>
<td>10.1387</td>
<td>?</td>
</tr>
<tr>
<td>40m</td>
<td>7.0746</td>
<td>?</td>
</tr>
<tr>
<td>80m</td>
<td>3.5926</td>
<td>same</td>
</tr>
<tr>
<td>160m</td>
<td>1.8086</td>
<td>?</td>
</tr>
</tbody>
</table>

Source: [http://WSPRnet.org/drupal/node/2](http://WSPRnet.org/drupal/node/2)

Determining that actual transmitting frequency takes a bit of working out. The bandwidth of a transmission is about 200Hz. WSPR makes use of the upper side band (USB) and the single side band (SSB) bandwidth allowed is 3000Hz. Say the transmissions falls in the range 1400–1600 Hz above the dial frequency. Quite a small slot of bandwidth, especially when you consider a dozen or more station can simultaneous transmit and be received simultaneously. So, taking the 20 meter band used on the QRPi 14.095600 + 1400 = 14.097000 and, 14.095600 + 1600 = 14.097200, so you are transmitting between 14.097000 and 14.097200. Each successive transmission is moved by software, by a random frequency offset of 80 Hz.

```
pip/wspr:~/logs $ tail -20 wspr.log
TX ended at: UTC 2017-10-11 04:57:51.596 (110.592 s)
Desired center frequency for WSPR transmission: 14.097138 MHz
Waiting for next WSPR transmission window...
TX started at: UTC 2017-10-11 04:58:01.005
Desired center frequency for WSPR transmission: 14.097176 MHz
Waiting for next WSPR transmission window...
TX started at: UTC 2017-10-11 04:59:51.597 (110.592 s)
Desired center frequency for WSPR transmission: 14.097054 MHz
Waiting for next WSPR transmission window...
TX started at: UTC 2017-10-11 05:00:01.004
Desired center frequency for WSPR transmission: 14.097161 MHz
Waiting for next WSPR transmission window...
TX started at: UTC 2017-10-11 05:01:51.596 (110.592 s)
Desired center frequency for WSPR transmission: 14.097046 MHz
Waiting for next WSPR transmission window...
TX started at: UTC 2017-10-11 05:02:01.005
```

### 3.5 Why is my location not accurate? [Top]

Message space is short so only information essential to propagation reporting is included. I guess a four character Maidenhead is considered sufficient for this purpose. The wspr command will accept a six character without error but required two transmission to communicate the last two characters of the location code . More confusing is that the WSPRnet.org site allows and display six character Maidenhead for receiving stations. However, since the intended purpose of WSPR is to report on propagation characteristic, a position within 100k is more than adequate in most cases. The Maidenhead Locator System (named after the town outside London where it was first conceived by a meeting of European VHF managers in 1980), is a grid square that measures 1° latitude by 2° longitude and measures approximately 70 × 100 miles. A grid square is indicated by two letters (the field) and two numbers (the square), as in QG62, and further refined by two more letters to produce "QG62qi". These more precise locators measure 2.5 minutes latitude by 5 minutes longitude, roughly corresponding to 3 × 4 miles in Australian latitudes.

I will leave the last say to Paul VK4ZBV in his post at [http://WSPRnet.org/drupal/node/7109](http://WSPRnet.org/drupal/node/7109)

"WSPR normally only has enough bits allocated to the transmitted message to send a simple callsign, 4 digit locator and the TX power in dBm. This is reported by the receiving station to the WSPRnet.org database where the extra two digits, if available are added to your received locator to provide a more accurate location, distance and bearing.

The database gets the extra two digits from the reports of received stations that your station sends to WSPRnet.org. Stations that run TX only beacons and don’t send reports of received stations are not telling the database where they are, so it can only display their four digit locator which can have errors of many miles from their actual location. If you must send your six digit locator, or you need to send a complex callsign (prefix or suffix), you can force the transmission of the six digit locator by means of a second transmission to convey the extra data required. This requires that the receiving station must copy both...
transmissions to extract all the information, so your chance of being reported is somewhat reduced with poor S/N ratios. You can find more details in the WSPR 2.0 User’s Guide (English) download from http://physics.princeton.edu/pulsar/K1JT/wspr.html”

Mainhead grid squares can be display at http://www.levinecentral.com/ham/grid_square.php.

QG62 grid square measures 1 degrees latitude by 2 degrees longitude

QG62qi grid square measures 2.5 minutes latitude by 5 minutes longitude

Source: http://dxcluster.ha8tks.hu/map/

3.6 (Mis)Using WSPR fields to encode data  [Top]
Ethan Zonca has suggested a hack to increase the information that can be transmitted by WSPR in an article at http://protofusion.org/wordpress/2017/09/tracking-balloons-with-wspr. He states, “We can work around two of these issues with a slight hack on the WSPR network. WSPR packets that have callsigns with ‘0’ or ‘Q’ are considered invalid packets (since no allocated callsign starts with these characters). However, WSPR receivers still process and upload these packets to WSPRnet, and they can be used for balloon telemetry”.

3.7 Enable ssh on Raspberry Pi [Top]

On the raspberry Pi with monitor and keyboard and mouse attached login with the following credentials:
Username: pi
Password: raspberry

Select the terminal application from the menu by selecting the Raspberry Pi logo menu --> Applications --> Terminal
In the terminal type the following command:

```
sudo raspi-config
```

then navigate to ssh, hit Enter and select Enable or Disable ssh server.

```
reboot
```

After Pi is restarted check ip-address with this command and

```
ifconfig
```

**Warning - It is a good idea to change password.**
This can be done by simply typing "passwd" at the command line and following the prompts. Alternatively, any account’s passwords can be changed with these commands:

```
sudo -i
passwd <username> eg "pi"
```

Now return to your linux workstation, or windows workstation with putty installed and remotely login to the Raspberry Pi. After this step you'll not need to use monitor and keyboard to run commands on your Raspberry. All can be performed remotely over a network connection.

**Remotely accessing the WSPR QRPI from a Linux Workstation**

Use the secure shell program (ssh) to remotely access the Raspberry Pi from a workstation.

Running the wspr application

```
------------------------- start of file -------------------------------
#!/bin/sh
# daemon.sh - to run QRPI wspr transmitter
# Glenn N Lyons 20171006
# run from systemd at boot time. See /lib/systemd/system/qrpi.sh
# DO NOT EDIT unless changes are compatible with systemd call
# **** NOTE!!! the #!/bin/sh on line one is essential for systemd to operate
#
CALLSIGN=VK4PK
# QTH=QG62 qi by gnl201701161747
QTH=QG62qi
LOGDIR=/home/pi/logs
LOGFILE=${LOGDIR}/wspr.log
TEMP=/home/pi/logs/temp.log
#
## Rotate the logfile five times
if [ -f ${LOGDIR}/wspr.4.log ]
then
 mv -f ${LOGDIR}/wspr.4.log ${LOGDIR}/wspr.5.log
fi
if [ -f ${LOGDIR}/wspr.3.log ]
then
 mv ${LOGDIR}/wspr.3.log ${LOGDIR}/wspr.4.log
fi
if [ -f ${LOGDIR}/wspr.2.log ]
then
 mv ${LOGDIR}/wspr.2.log ${LOGDIR}/wspr.3.log
fi
if [ -f ${LOGDIR}/wspr.1.log ]
then
 mv ${LOGDIR}/wspr.1.log ${LOGDIR}/wspr.2.log
fi
if [ -f ${LOGDIR}/wspr.log ]
then
 mv ${LOGDIR}/wspr.log ${LOGDIR}/wspr.1.log
fi

vcgenccmd measure_temp >>$TEMP  # sample the RPi's tempreature
vcgenccmd display_power 0 >>$LOGFILE  # turn off video power

------------------------- end of file -------------------------------
```

```
```
3.8 Installing System Daemon Service (systemd)  [Top]

One of several methods to run a program on your Raspberry Pi at startup is to use the systemd files. "systemd" provides a standard process for controlling which programs run when a Linux system boots up. Note that systemd is available only from the Jessie versions of Raspbian OS.

There are four steps to set up systemd:

1. Create a Unit File
2. Setting Up Files and Permissions
3. Re-Initialise Systemd
4. Test to make sure it is working

**Step 1 – Create A Unit File**

Open a new unit file using the command shown below (Feel free to use your favourite editor):

```
sudo vi /lib/systemd/system/qrpi.service
```

Add the following text with your callsign and location:

```
# qrpi.service - Startup QRPi wspr transmitter as daemon service
# Glenn N Lyons VK4PK
# 20171005
[Unit]
Description=QRPi WSPR Transmitter
After=multi-user.target

[Service]
Type=simple
User=pi
WorkingDirectory=/home/pi/WsprryPi
ExecStart=/home/pi/WsprryPi/daemon.sh
Restart=on-abort

[Install]
WantedBy=multi-user.target
```

Save and exit the editor.

You have now configure systemd to run a program called "wspr" on your Raspberry Pi at startup time. This defines a new service called “QRPi WSPR Transmitter” and we are requesting that it is launched once the multi-user environment is available. The “ExecStart” parameter is used to define the absolute path of the startup script daemon.sh in the WsprryPi folder.

**Step 2 - Setting Up Files and Permissions**

There are a few things to do to set up the required files with the correct permissions. Make sure to create the log file folder and logfile. The permission on the unit file, qrpi.service, needs to be set to 644, the permission on logfile directory, logs, is set to 774. Then, use “touch” to create a logfile, wspr-systemd.log, and set it permission to 664, and finally make the “wspr” application is executable by adding “x” to its permission.

The commands are as follows:

```
mkdir /home/pi/logs
sudo chmod 644 /lib/systemd/system/qrpi.service
sudo chmod 774 /home/pi/logs
sudo touch /home/pi/logs/wspr-systemd.log
sudo chmod 664 /home/pi/logs/wspr-systemd.log
sudo chmod +x wspr
```

**Step 3 - Re-Initialise Systemd**

Now the unit and log files have been created and permissions correctly set, we can tell systemd to start:

```
sudo systemctl daemon-reload
sudo systemctl enable qrpi.service
```

You should get the message: "Created symlink /etc/systemd/system/multi-user.target.wants/qrpi.service → /lib/systemd/system/qrpi.service."

And finally to fully test the service is working correctly reboot the Raspberry Pi.

```
sudo reboot
```

The "QRPi" service should be running when the system comes up again. The QRPi service runs up the "wspr" application, so search for that process in memory:
pi@wspr:~ $ ps -eaf | grep wspr
avahi  233  1  0 10:20 ?        300:00:00 avahi-daemon: running [wspr.local]
root  891  890  0 10:59 pts/1    00:00:00 sudo /home/pi/WsprryPi/wspr -r -o -s VK4PK QG62 20 20m
pi  892  890  0 10:59 pts/1    00:00:00 tail -f /home/pi/logs/wspr.log
root  896  891  4 10:59 pts/1    00:00:01 /home/pi/WsprryPi/wspr -r -o -s VK4PK QG62 20 20m
pi  899  652  0 10:59 pts/0    00:00:00 grep --color=auto wspr

Step 4 - Is it Working?

If you have problems and don't see the "wspr" process above, here are some commands to help find the cause:

sudo journalctl | grep qrpi
sudo journalctl -u qrpi.services
sudo systemctl --failed

Help with systemd can be found at these sites:
https://wiki.archlinux.org/index.php/systemd

3.9 Backup Backup Backup [Top]

After all that work a Backup is advisable, so shutdown the Raspberry Pi and take the microSD out and mount it on your linux workstation. On linux use the dd utility to make a copy of your work.

dd if=/dev/sdc status=progress |gzip -c >qrpi-2017-10-11.img.gz

dd is a disk to disk copy utility. It is real quite dump and copies block for block.
"if" is the input file which, in this case, is the raw microSD device.
gzip is the GNU open systems version of the popular Zip archive utility.
The "-c" tell gzip write to the standard output which is then restricted by ">" to a file.

3.10 Manipulating the QRPi Service [Top]

There are several useful command to control the running transmitter, "wsprr" via the qrpi service tag. On some occasions you may want to turn the wspr off or run the process manually from the terminal login.
They are as follows:

"systemd" commands

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reload all the systemd configurations</td>
<td>sudo systemctl daemon-reload</td>
</tr>
<tr>
<td>Stops the service immediately</td>
<td>sudo systemctl stop qrpi.service</td>
</tr>
<tr>
<td>Restarts a service</td>
<td>sudo systemctl restart qrpi.service</td>
</tr>
<tr>
<td>Shows the status the service (including whether it is running or not)</td>
<td>sudo systemctl status qrpi.service</td>
</tr>
<tr>
<td>Enables the service to be started on bootup</td>
<td>sudo systemctl enable qrpi.service</td>
</tr>
<tr>
<td>Disables the service to not start during bootup</td>
<td>sudo systemctl disable qrpi.service</td>
</tr>
</tbody>
</table>

3.11 The WSPR Protocol [Top]

WSPR modulation is 4-FSK (Layland-Lushbaugh polynomials for a K=32, r=1/2 convolutional code) The type of radio emission is classified "F1D", frequency-shift keying. A message contains a station's callsign, Maidenhead grid locator, and transmitter power in dBm. The WSPR protocol compresses the information in the message into 50 binary digits. These are encoded using a convolutional code with constraint length K=32 and a rate of r=1/2. The long constraint length makes undetected decoding errors less probable at the cost, that the highly efficient Viterbi algorithm must be replaced by a simple sequential algorithm for the decoding process.

Source: WSPRing-Around-the-World-QS11-2010-Taylor.pdf (QST magazine December 2001 page 37)

In telecommunication, a convolutional code is a type of error-correcting code that generates parity symbols via the sliding application of a boolean polynomial function to a data stream. A detailed explanation of convolutional code can be found on Wikipedia, https://en.wikipedia.org/wiki/Convolutional_code

The Viterbi Algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states – called the Viterbi path – that results in a sequence of observed events, especially in the context of Markov information sources and hidden Markov models.

Once again, a detail explanation can be found at Wikipedia https://en.wikipedia.org/wiki/Viterbi_algorithm

A detailed description of an "after market" wspr protocol can be found on Andy Talbot, G4JNT site.

Source: G4JNT: The WSPR Coding Process: Non-normative specification of WSPR protocol
3.12 Reporting Maps On The Internet [Top]

Using your Internet browser head to http://WSPRnet.org/drupal/WSPRnet/map. You do not have to login, but scroll down the page and enter the search criteria, and click “Update”. You should be presented with a map of the contacts achieved in the time period selected. If you want more details the “Database” tab provide a tabular history for analysis. Map display can also be had by searching on the http://wspr.aprsinfo.com site.

WSPRnet also provides a database download facility. The day night shading can be switched on or off. The "timestamp" is UTC so the 12:30 hours on the KK1D example is 2:30 hours local. This is when you would expect the ionospheric propagation would enable this hop. “az” is the azimuth in degrees between spotter and spotee.
It is always gratifying to know a little about who is reporting your signals. P. Joshua Rovero, KK1D has an interesting website, http://www.roveroresearch.info

More important his location is "FN31vi" and that is somewhere very close to East Lyme, Connecticut USA. I did the plot on Google Earth and the results were close.

### Database

Specify query parameters

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Call</th>
<th>MHz</th>
<th>SNR</th>
<th>Drift</th>
<th>Grid</th>
<th>Pwr</th>
<th>Reporter</th>
<th>RGrid</th>
<th>km</th>
<th>az</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017-10-24 19:40</td>
<td>VK4PK</td>
<td>14.097084</td>
<td>-20</td>
<td>-3</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VU3KAZ</td>
<td>MK82tu</td>
<td>9309</td>
<td>288</td>
</tr>
<tr>
<td>2017-10-24 20:00</td>
<td>VK4PK</td>
<td>14.097033</td>
<td>-28</td>
<td>-3</td>
<td>QG62qi</td>
<td>0.1</td>
<td>HB9HD</td>
<td>JN47hj</td>
<td>16259</td>
<td>315</td>
</tr>
<tr>
<td>2017-10-24 20:00</td>
<td>VK4PK</td>
<td>14.097032</td>
<td>-21</td>
<td>-3</td>
<td>QG62qi</td>
<td>0.1</td>
<td>EA8BFK</td>
<td>IL38bo</td>
<td>18761</td>
<td>278</td>
</tr>
<tr>
<td>2017-10-24 20:50</td>
<td>VK4PK</td>
<td>14.097085</td>
<td>-24</td>
<td>-1</td>
<td>QG62qi</td>
<td>0.1</td>
<td>ZL2RKL</td>
<td>RE68xd</td>
<td>2424</td>
<td>135</td>
</tr>
<tr>
<td>2017-10-24 21:20</td>
<td>VK4PK</td>
<td>14.097090</td>
<td>-24</td>
<td>-2</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK5MR</td>
<td>PF89kk</td>
<td>1634</td>
<td>255</td>
</tr>
<tr>
<td>2017-10-24 21:20</td>
<td>VK4PK</td>
<td>14.097081</td>
<td>-15</td>
<td>-2</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3WHO</td>
<td>QF22Hl</td>
<td>1324</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-24 21:30</td>
<td>VK4PK</td>
<td>14.097020</td>
<td>-23</td>
<td>-2</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK5MR</td>
<td>PF89kk</td>
<td>1634</td>
<td>255</td>
</tr>
<tr>
<td>2017-10-24 21:30</td>
<td>VK4PK</td>
<td>14.097011</td>
<td>-14</td>
<td>-2</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3WHO</td>
<td>QF22Hl</td>
<td>1324</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-24 21:40</td>
<td>VK4PK</td>
<td>14.097090</td>
<td>-7</td>
<td>-2</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3WHO</td>
<td>QF22Hl</td>
<td>1324</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-24 21:40</td>
<td>VK4PK</td>
<td>14.097079</td>
<td>-22</td>
<td>-1</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3FFB</td>
<td>QF22pe</td>
<td>1359</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-24 21:40</td>
<td>VK4PK</td>
<td>14.097099</td>
<td>-26</td>
<td>-2</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK5MR</td>
<td>PF89kk</td>
<td>1634</td>
<td>255</td>
</tr>
<tr>
<td>2017-10-24 21:50</td>
<td>VK4PK</td>
<td>14.097013</td>
<td>-12</td>
<td>0</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3FFB</td>
<td>QF22pe</td>
<td>1359</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-24 21:50</td>
<td>VK4PK</td>
<td>14.097023</td>
<td>0</td>
<td>-4</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3WHO</td>
<td>QF22Hl</td>
<td>1324</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-24 21:50</td>
<td>VK4PK</td>
<td>14.097032</td>
<td>-24</td>
<td>-2</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK5MR</td>
<td>PF89kk</td>
<td>1634</td>
<td>255</td>
</tr>
<tr>
<td>2017-10-24 22:10</td>
<td>VK4PK</td>
<td>14.097029</td>
<td>-15</td>
<td>-1</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3FFB</td>
<td>QF22pe</td>
<td>1359</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-24 22:10</td>
<td>VK4PK</td>
<td>14.097038</td>
<td>+3</td>
<td>-2</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3WHO</td>
<td>QF22Hl</td>
<td>1324</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-24 22:30</td>
<td>VK4PK</td>
<td>14.097003</td>
<td>-5</td>
<td>-2</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3FFB</td>
<td>QF22pe</td>
<td>1359</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-24 22:30</td>
<td>VK4PK</td>
<td>14.097013</td>
<td>-3</td>
<td>-1</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3WHO</td>
<td>QF22Hl</td>
<td>1324</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-24 23:00</td>
<td>VK4PK</td>
<td>14.097046</td>
<td>+2</td>
<td>-1</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3WHO</td>
<td>QF22Hl</td>
<td>1324</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-24 23:50</td>
<td>VK4PK</td>
<td>14.097188</td>
<td>+2</td>
<td>-1</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3WHO</td>
<td>QF22Hl</td>
<td>1324</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-25 00:30</td>
<td>VK4PK</td>
<td>14.097168</td>
<td>-4</td>
<td>-1</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3WHO</td>
<td>QF22Hl</td>
<td>1324</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-25 00:30</td>
<td>VK4PK</td>
<td>14.097167</td>
<td>-14</td>
<td>0</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3FFB</td>
<td>QF22pe</td>
<td>1359</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-25 00:40</td>
<td>VK4PK</td>
<td>14.097150</td>
<td>-9</td>
<td>-1</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3WHO</td>
<td>QF22Hl</td>
<td>1324</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-25 00:40</td>
<td>VK4PK</td>
<td>14.097141</td>
<td>-12</td>
<td>0</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3FFB</td>
<td>QF22pe</td>
<td>1359</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-25 00:50</td>
<td>VK4PK</td>
<td>14.097100</td>
<td>-8</td>
<td>-2</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3WHO</td>
<td>QF22Hl</td>
<td>1324</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-25 01:00</td>
<td>VK4PK</td>
<td>14.097075</td>
<td>0</td>
<td>0</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3WHO</td>
<td>QF22Hl</td>
<td>1324</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-25 01:10</td>
<td>VK4PK</td>
<td>14.097135</td>
<td>-7</td>
<td>-2</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3WHO</td>
<td>QF22Hl</td>
<td>1324</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-25 01:20</td>
<td>VK4PK</td>
<td>14.097133</td>
<td>-2</td>
<td>-1</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3WHO</td>
<td>QF22Hl</td>
<td>1324</td>
<td>212</td>
</tr>
<tr>
<td>2017-10-25 01:30</td>
<td>VK4PK</td>
<td>14.097038</td>
<td>-11</td>
<td>0</td>
<td>QG62qi</td>
<td>0.1</td>
<td>VK3WHO</td>
<td>QF22Hl</td>
<td>1324</td>
<td>212</td>
</tr>
</tbody>
</table>

Query time: 0.019 sec

It is always gratifying to know a little about who is reporting your signals. P. Joshua Rovero, KK1D has an interesting website, http://www.roveroresearch.info

More important his location is "FN31vi" and that is somewhere very close to East Lyme, Connecticut USA. I did the plot on Google Earth and the results were close.

### Database

Specify query parameters

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Call</th>
<th>MHz</th>
<th>SNR</th>
<th>Drift</th>
<th>Grid</th>
<th>Pwr</th>
<th>Reporter</th>
<th>RGrid</th>
<th>km</th>
<th>az</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017-10-25 12:30</td>
<td>VK4PK</td>
<td>14.097133</td>
<td>-25</td>
<td>-1</td>
<td>QG62qi</td>
<td>0.1</td>
<td>KK1D</td>
<td>FN31vi</td>
<td>15626</td>
<td>58</td>
</tr>
</tbody>
</table>

Query time: 0.174 sec
3.13 Other Applications [Top]

<table>
<thead>
<tr>
<th>Modulation</th>
<th>Program</th>
<th>Jumper: GPIO4</th>
<th>Jumper: GPIO18</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSB</td>
<td>rpitx/qtcsdr</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>FSQ</td>
<td>rpitx</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>NFM</td>
<td>rpitx/qtcsdr</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>AM</td>
<td>rpitx/qtcsdr</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>WSPR</td>
<td>WsprryPi</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CW</td>
<td>PiCW</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SSTV</td>
<td>PySSTV</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Freq. Sweep</td>
<td>FREQ_PI</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Note the add on board comes with the jumper set as default to GPIO4

3.14 What Have I Learned? [Top]

I am surprised how far a 100 milliwatts can get and how well a simple addon board and Rasperry Pi can work. I am very please that such an inexpensive project can achieve such significant results.

There is unresolved understanding of how exactly LOC code works

I have discovered an inconvenience. The WsprryPi "wspr" program needs a switch to introduce a times gap between successive transmissions so signal from other station can be received on other equipment, like my IC-7300 in the quiet periods. The QRPI cannot receive so has no apparent need to pause and listen. Therefore, I cannot monitor the 20 meter band as my own signal is almost continuously drowning out all others. This was later solved by Paul Beet who pointed out there is a facility to transmit as 0 frequency.

It is interesting to watch the propagation differences between day and night and I am looking forward to seeing how they pan out over a longer time frame.

How much time a "quick little project" can take! "Knock it up in a few hours" turned into days then weeks. On the other hand I have learned a lot and been spurred on to greater heights to do more projects.

3.15 References [Top]

Linux systemd daemon management system
https://wiki.archlinux.org/index.php/systemd

Fault finding advise for systemd can be found on the Red Hat site.
CHAPTER FOUR - The Importance of Time

4.01 Why is super accurate time important? [Top]

Horology is the art or science of measuring time and maybe the greatest horologist was John Harrison (1693 – 1776) was a self-educated English carpenter. The building complex instruments to keep increasingly accurate time can arguably be said to begin with Harrison and his timepieces. A brief account of Harrison can be found at Wikipedia https://en.wikipedia.org/wiki/John_Harrison. The book by Dava Sobel, "Longitude: The True Story of a Lone Genius Who Solved the Greatest Scientific Problem of His Time" is a must read.

Now we build temperature compensate crystal locked oscillators and reference GPS satellites and cesium clocks via the Network Time protocol to keep those time pieces very accurate. Timing is very important to the transmitter mode used by WSPR as the the encoding is taken for the transition and not the state of the modulated signal.

4.02 The Jitter Bug [Top]

The Jitterbug is a kind of swing dance popularized in the United States in the early 1900’s, but it is not the type of jitter bug that concerns us. We are talking about the jitter that is the deviation from true periodic signal, in relation to a reference clock signal. Clock period recovery is called timing jitter and is considered an undesirable bug. Some time we transmission to early, some times to late. In electronics and telecommunications, jitter is the deviation from true periodicity of a presumably periodic signal, often in relation to a reference clock signal. In clock recovery applications it is called timing jitter.
So how do we fix this bug? The answer lies with a NTP time or GPS time.

### 4.03 NTP or GPS? [Top]

Somehow we need to constantly correct the drift introduced by tiny inaccuracies in the Pi's local crystal oscillator, so we turn to the internet or the skies above for a more accurate time reference to correct our local time. It not a simple as reading those time references as there are correction and averages to be made. A detailed resource on NTP and the RPI can be found here: [http://www.satsignal.eu/ntp/Raspberry-Pi-NTP.html](http://www.satsignal.eu/ntp/Raspberry-Pi-NTP.html) and is well worth a read. This, of course requires an internet connection or an add-on GPS board to the RPi.

**Stratum 0**

These are high-precision timekeeping devices such as atomic (cesium, rubidium) clocks, GPS clocks or other radio clocks. They generate a very accurate pulse per second signal that triggers an interrupt and timestamp on a connected computer. Stratum 0 devices are also known as reference clocks.

**Stratum 1**

These are computers whose system clocks are synchronized to within a few microseconds of their attached stratum 0 devices. Stratum 1 servers may peer with other stratum 1 servers for sanity checking and backup.[12] They are also referred to as primary time servers.[2][3]

**Stratum 2**

These are computers that are synchronized over a network to stratum 1 servers. Often a stratum 2 computer will query several stratum 1 servers. Stratum 2 computers may also peer with other stratum 2 computers to provide more stable and robust time for all devices in the peer group.

**Stratum 3**

These are computers that are synchronized to stratum 2 servers. They employ the same algorithms for peering and data sampling as stratum 2, and can themselves act as servers for stratum 4 computers, and so on.

**Stratum 4-15**
The upper limit for stratum is 15; stratum 16 is used to indicate that a device is unsynchronized. The NTP algorithms on each computer interact to construct a Bellman-Ford shortest-path spanning tree, to minimize the accumulated round-trip delay to the stratum 1 servers for all the clients.[1]:20

```bash
# ntpq -p
or in interactive mode use the peers option
```

```
-------------------- Man Page ------------------------------------------
peers - Display a list of peers in the form:
[tally] remote refid st t when poll reach delay offset jitter
Variable Description:
[tally] - single-character code indicating current value of the select field
of the .lk.decode.html#peer "peer status word"
remote - host name (or IP number) of peer. The value displayed will be
truncated to 15 characters unless the -w flag is given, in which
case the full value will be displayed on the first line, and the
remaining data is displayed on the next line.
refid - association ID or .lk.decode.html #kiss "'kiss code"
st - stratum
t - u: unicast or manycast client,
b: broadcast or multicast client,
l: local (reference clock),
s: symmetric (peer),
A: manycast server,
B: broadcast server,
M: multicast server
when - sec/min/hr since last received packet
poll - poll interval (log2 s)
reach - reach shift register (octal)
delay - roundtrip delay
offset - offset of server relative to this host
jitter - jitter
-------------------- End Man Page ------------------------------------
```

```
ntpq> peer
remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
0.ubuntu.pool.n .POOL.          16 p    -   64    0    0.000    0.000   0.000
1.ubuntu.pool.n .POOL.          16 p    -   64    0    0.000    0.000   0.000
2.ubuntu.pool.n .POOL.          16 p    -   64    0    0.000    0.000   0.000
3.ubuntu.pool.n .POOL.          16 p    -   64    0    0.000    0.000   0.000
ntp.ubuntu.com  .POOL.          16 p    -   64    0    0.000    0.000   0.000
+x.ns.gin.ntt.ne 249.224.99.213   2 u   49   64  377   30.143    2.053   6.056
+ntp2.ds.network 202.46.191.123   2 u   53   64  377   76.790    5.543  8.259
+103.214.220.220 202.6.131.118   2 u   61   64  377   34.170    1.065  6.044
+joker slash31. .GPS.            1 u   65   64  377   80.810    6.980  8.090
+144.48.166.166 202.6.131.118   2 u   63   64  377   34.213    1.938  2.815
+y.ns.gin.ntt.ne 249.224.99.213   2 u   47   64  377   30.230    4.784  2.108
+ntpl.ursys.com. 283.35.83.242   2 u   46   64  377   30.230    3.698  2.000
+mail1.selcomm. .PPS.            1 u   50   64  377   30.192    0.975 33.566
+mail1.ahsnetwork 187.84.184.106 2 u   44   64  377   42.717    0.317  6.062
+115.187.159.2 10.0.1.252       2 u   58   64  377   81.223   -0.968  7.892
+ntp.2000cn.com. .PPS.            1 u   62   64  377   83.623    5.255  2.302
```

Appendix A - The essential for good decode  [Top]

The essential things required for a good decode are:
1. Your computer clock must be set correctly, errors greater than + or - three seconds
   will prevent both RX and TX decoding). (Quick check are the received signals on the waterfall,
   starting exactly on the two minute markers)?
2. You are using Upper Sideband, the correct sideband for WSPR (LSB won’t decode)
3. The grey box showing Rx Noise should be showing 0 dB but, it is not critical +- 10 dB
   will work ok.
4. Frequency is critical, only to the extent that the WSPR prog will not decode anything
   outside Plus or Minus one hundred hertz of the 1,5kHz audio tone frequency so selecting
   USB setting the tranceiver dial to 10,138,700 should receive the WSPR band
   10,140100 to 10,140,300, a signal on 10,140,200 should produce an audio tone of 1,500 Hz.
   (No RIT or XIT offset)

So assuming Correct Timing, USB Sideband selected, RX Noise Level (0dB) adjusted, and Dial Frequency correct, the
next step is to look at the software and the computer. What Version of WSPR’s, what operating system and brief details of
CPU and memory? A problem that the trouble lies in slow processing of the received data. The WSPR transmission has 50
user data bits, 112 error correcting code bits, combined with another 162 bits of synchronizing symbols fed into the
decoder and error correction process. When the signal is nice and clean a single pass through the decoder may produce a
good decode. When the received signal is noisy, distorted, or suffers interference the data is looped through the decoder
many times, the more times the longer the decoding time. When the signal is badly distorted it may do up to 20,000 loops
through the decoder, which depending on the computer speed and other processing being done can take minutes. You can
see in this printout extract, the 2nd last figure in each line is the number of decode loops each decode required. (1 to
19860)
So a slow decode ties the computer up for a long time, reducing the chance of decoding other signals. If you are hitting 100% for long during a decode you have a problem. A summary from a post by Paul VK4ZBV. Source

http://www.wsprnet.org/drupal/node/2403