WSPR Without Tears

Table of Contents

Introduction

Thank you for purchasing the WSPR Without Tears transmitter (WWoT). We hope you’ll be pleased with it. WWoT is our attempt to put together a WSPR transmitter transmitter that avoids (hopefully) most of the problems associated with building a fairly complicated digital mode transmitter. All of the information, data and files are accessible through the TAPR website:

TAPR - WSPR Webpage

WSPR stands for Weak Signal Propagation Reporter Network. It's a digital mode used by hams. The process is similar to a beacon. WSPR is a wonderful communication mode created by Joe Taylor (K1JT). WWoT is our attempt at simplifying the process of getting an actual transmitter up and running so you can enjoy working with WSPR and avoid dealing with the headaches associated with building a working system.

WSPR Without Tears

What makes it really interesting is that WSPR is optimized for operating at very low power. For example, it's normal to transmit with 200 mW from the US and be received in Australia or Europe. There are hams all over the world with WSPR receivers. They record what they receive in a central database that can be queried over the Internet. So you put your WSPR transmitter on the air and monitor its performance from the Internet. Monitoring WSPR is sort of addictive. It's fascinating to see what effect time of day, seasons, and weather have on propagation.

There is a wide array of choices for setting up a WSPR beacon. The most obvious and possibly the simplest is to use your regular transceiver. This requires a computer and sound card connection. Most of us have at least one computer in the shack and many of us are already using digital modes, so the equipment is already in place to run WSPR. All you have to do is download the software (free) and you're off to the races. This approach works well and the price is right if you're already are set up for digital modes. Even if you're not set up for digital modes, the interface between your transceiver and computer can be had for less than $100.

Using your base station for WSPR is an excellent approach. The downside is that you are tying up your station while WSPR is on, so leaving WSPR on for an extended length of time gets in the way of radio. A second point, the output power of many transceivers can't be adjusted below 5 watts. An output power of 5 watts will certainly work, but it's interesting to see how little power you can get away with and the useful power range is a lot less than 5 watts.

The alternative to using your regular station for WSPR is to have a completely separate WSPR station. That's a lot more realistic when you're dealing with output powers in the range of milliwatts. When we started down this path we looked at different approaches. One approach we looked at was high quality and well done, but there are some operational complaints we had with it that we wanted to fix. This is our version for a fix.

Approach

We found an alternate approach using a Raspberry Pi (Pi) computer to generate WSPR transmissions. A Pi is about the size of a deck of playing cards and costs about $40-$50. It has an amazing amount of computing power for something so small and so cheap. It runs the Linux operating system.

The Pi can be configured to generate the WSPR signal without any external components! That's just amazing to me. Somebody figured out how to modulate the clock/oscillator and generate a WSPR signal. The Pi will also set the time by calling an NTP server and use the time information to adjust the frequency, eliminating the need for any calibration or timing setup.

There are two problems with using the Pi for WSPR. First, the output signal is a square wave with harmonics from here out to next Tuesday. A low pass filter is necessary to keep other hams and the

FCC happy. Second, the output power is around 10 mW. We talked about the thrill of using low power to make contacts, but it's nice to start with a little more power than that.

Our WSPR Pi is a circuit board (pcb) that plugs into the Pi 40-pin header. It boosts the signal level to about 200 mW and contains a seven pole low pass filter to keep out-of-band transmissions down by 43 dB. It also contains an LED that lights when transmitting and a crude output power indicator. (We found it annoying to not know when and how well our system was working without connecting a 'scope or power meter and wanted something simple as a sanity check.)

The Pi is a computer, and most computers need to be shutdown properly rather than just have the power turned off. We added a push button to the circuit board to initiate a shutdown sequence in the Pi. We wrote a program to run in the background that monitors the push button and shuts the Pi down when it's pressed.

Raspberry Pi

The Pi is a computer and is subject to all of the headaches associated with computers. We wanted to make life easier for anybody trying to duplicate our WSPR experience by limiting the amount of computer exposure you're subjected to. Loading the software into the Pi is not difficult, but there are quite a few areas that can go wrong and it can be frustrating if you've never done it before. It only takes one problem to kill the whole process. There are a lot of steps involved in getting to a working WSPR transmitter.

We created a version of the software that runs automatically when the Pi is turned on. There are several parameters that you (the user) have to set: callsign, grid square and WiFi (if you're using wifi). We programmed a web server in the Pi (we told you it's a computer :-) that lets you fill this stuff in from a web page onyour PC. In other words, you bring up Internet Explorer or Chrome or Firefox and type in the Pi's address (we'll get to that) and the web page comes up and you enter the data. This means no programming of the Pi at all. Bring up the web page, type in your callsign, grid square, and WiFi parameters and you're up and running.

The operating system (Linux) and all of the software are stored on a micro SD card. The SD card is plugged into the Pi. Set it and forget it. You need a card reader (<$10) and software (from the TAPR website) and an SD card to do this. We'll repeat the important part: you have to burn the image onto the SD card (similar to burning a CD/DVD) and then plug it into the Pi. There's *no* programming.

What you'll need to complete the assembly

1. Raspberry Pi 2.0 Model B or Raspberry Pi 3.0
2. *2 amp (minimum) power supply w/micro USB connector.*

3. Ethernet cable.
4. WiFi USB dongle. You only need this if you're using an Pi 2.0 and you want to use WiFi. WiFi is built into the Pi 3.0.
5. SD card (4 GB min, grade 10)
6. SD card reader (micro SD card to USB)
7. antenna (mate to BNC connector on WWoT)

Programming an SD Card

Setup

1. You need an SD Card; at least 4GB, 8GB is better. Class 10.
2. SD Card reader/writer. We've got a Targus Micro SD/T-F. It doesn't really matter what card reader/writer you use. It connects your SD card to a USB port.
1. Win32DiskImager – It's a freebee. Download it from: *https://sourceforge.net/projects/win32diskimager/* or *sourceforge.net/projects/win32diskimager/.*
2. Install Win32DiskImager.
3. Disk Image – Download it from: WSPR SD Card Image for Pi 2, 3 or WSPR SD Card Image for Pi 4.
4. Unzip the file after it's finished downloading (it will take a while).

Program SD Card

1. Plug the SD Card reader/writer into a USB port.
2. Plug the SD Card into the reader/writer.

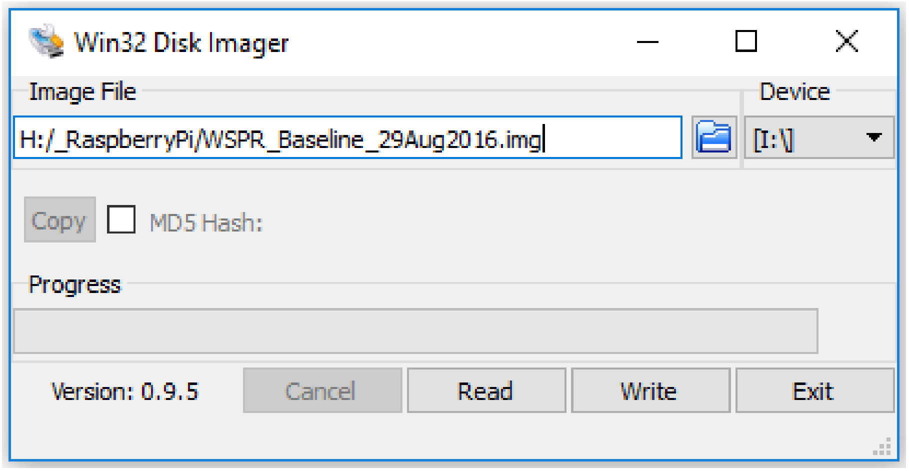


Image File: H:/_RaspberryPi/WSPR_Baseline_29Aug2016.img
Device: [I:\]
Version: 0.9.5

Figure 1: Win32DiskImager

3. Start Win32DiskImager.
4. Click on the folder icon in the upper right and find the disk image you downloaded and unzipped.
5. Click *Write* and go get a cup of coffee. Note: Win32DiskImager shows a small message box when it's done writing the SD card. Unfortunately, the message box likes to hide behind anything else you have up on the screen so you can't see it. Be forewarned that you might have to go looking for it.
6. Click *Exit*.
7. Remove the SD card and install it into the Pi. Make sure that power is disconnected from the Pi before doing this.

If you haven't annoyed the gods of the Internet too much today you should be in luck and everything will work!

Install SD Card into Pi

First, insert the SD card into the Pi (Figure 2: Inserting SD Card into Pi and Figure 3: SD Card Installed in Pi). The SD card sorta slides in up side down. On Pi 2.0 it ratchets in (pops in then out a little and sorta locks in place). On Pi 3.0 there's a friction fit, so there's no locking.

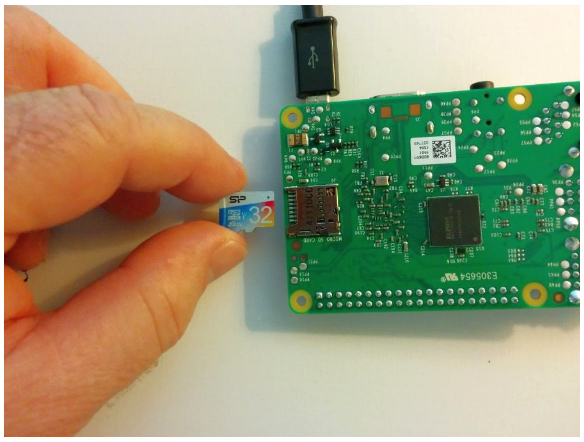


Figure 2: Inserting SD Card into Pi

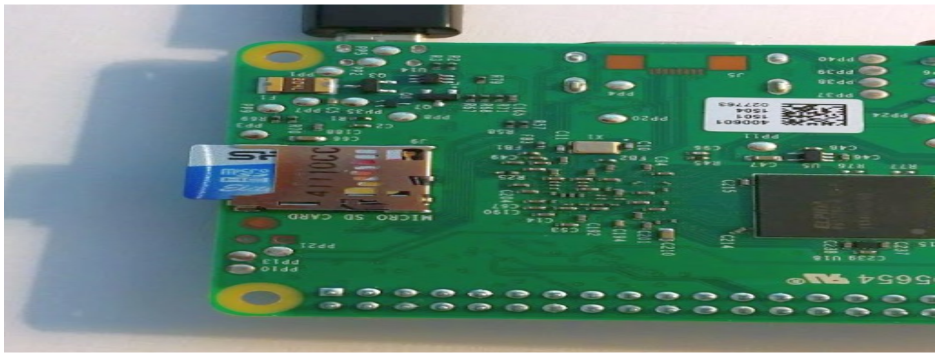


Figure 3: SD Card Installed in Pi

Assemble WWoT and Pi

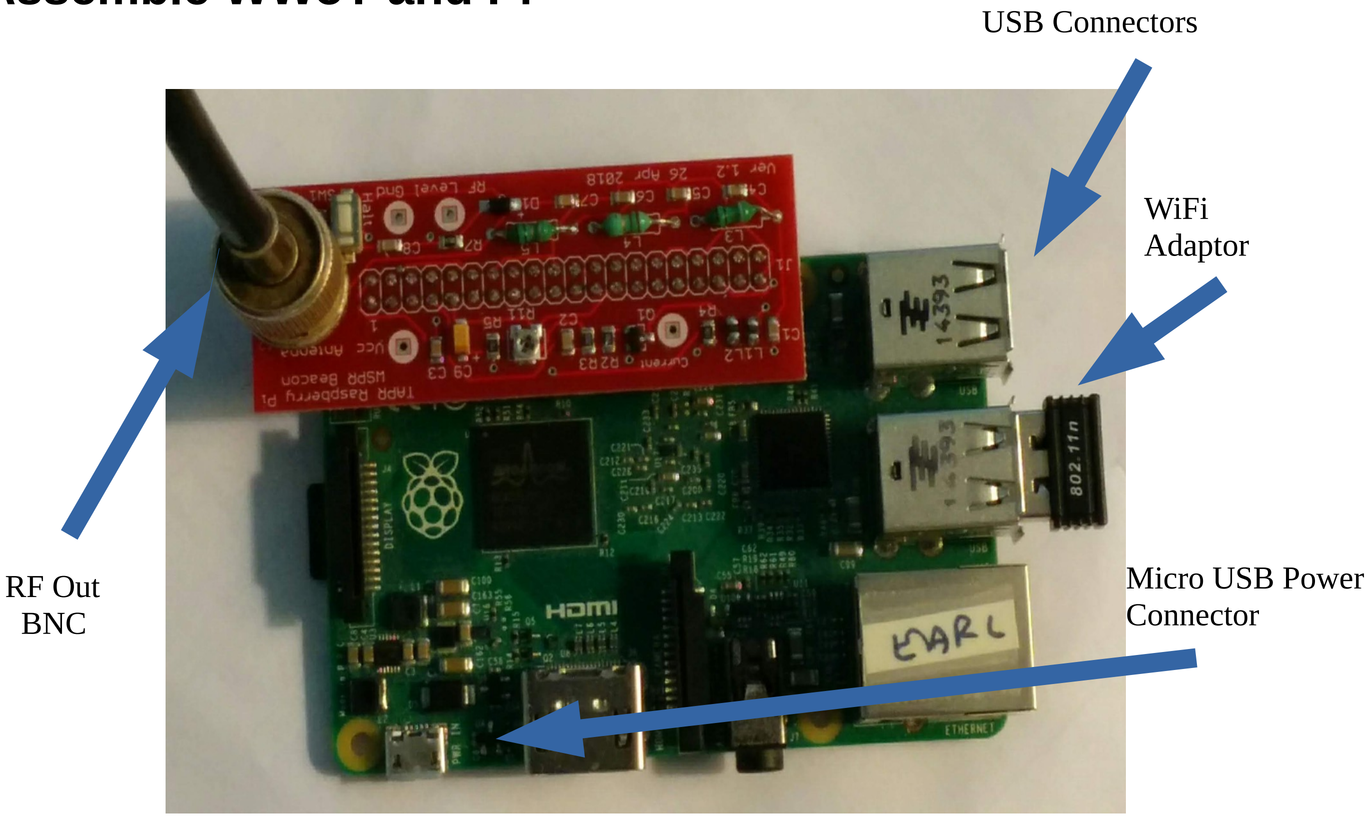


Figure 4: WWoT and Pi Assembly

Attach the WWoT board to the Pi. The BNC connector is on the opposite side from the Pi USB connectors(Figure 4: WWoT and Pi Assembly). The micro USB power connector is coming out of the lower left. RF output comes through a BNC connector shown in the upper left. A WiFi dongle is plugged into one of the USB ports (pi 2.0 only - it's built it to Pi 3.0). The Pi LEDs are on the left hand side of the Pi board below the WWoT board. They are not visible in the picture.

The side view of the completed assembly (Figure 5: WSPR Beacon (Side View)) shows how the WWoT is plugged into the Pi. It's definitely possible to connect them incorrectly, so don't do that. Notice from this view that the Pi USB connectors are on the right hand side and the WWoT BNC connector is on the left. That's how it should be. Also, make sure that the Pi pins and the WWoT plug are lined up. It is possible to offset one from the other, so you only have one row of pins connected.

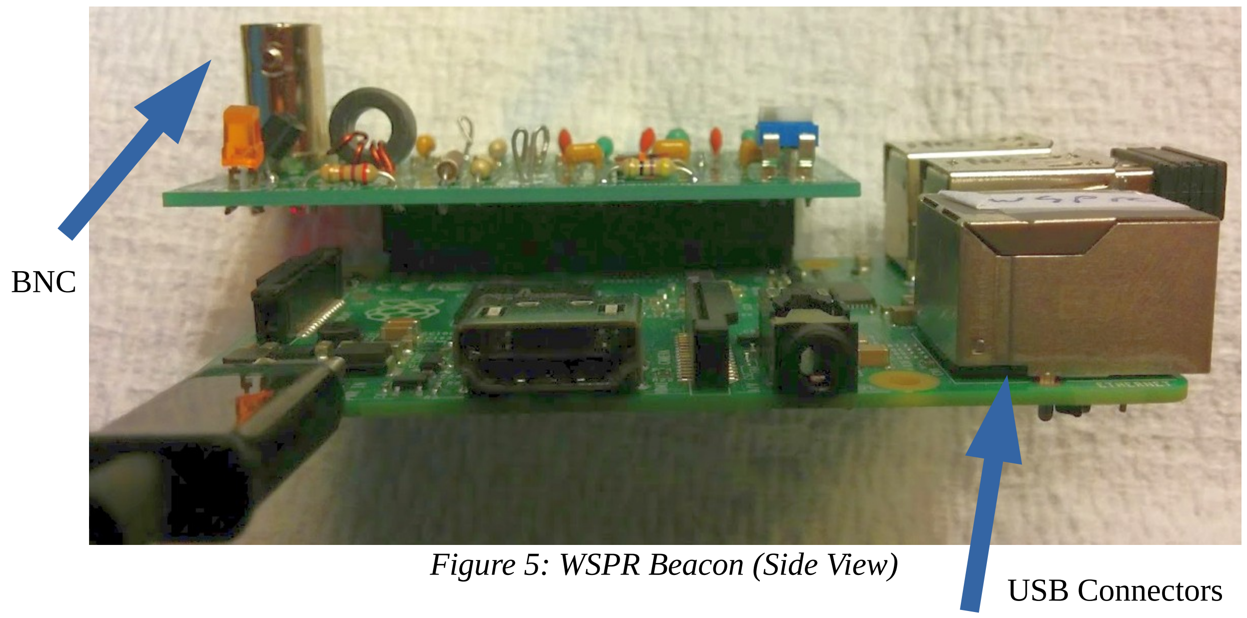


Figure 5: WSPR Beacon (Side View)

Setup and Adjustment

What You Need

I'm assuming that you have -

1. Assembled Pi/WWoT
2. 2+ amp power supply for the Pi
3. Ethernet cable to connect Pi to your router
4. SSID and passkey for your router (see below if you don't have these)
5. PC connected to your network
6. Optional: USB WiFi adapter for Pi 2 (Pi 3 and 4 have WiFi built in)

WSPR Without Tears

Getting SSID and Passkey

Start by looking on the router for login information. Also, try the Internet for a router manual (we know, we hate to read directions, too).

Next, you're looking for your router's web page. Open a browser and type the IP address *192.168.0.1* into the address textbox at the top (just as if you were typing www.google.com). If nothing happens try *192.168.1.1* and *192.168.2.1*.

Log into the router using the username and password printed on the side of the router or in the manual.

Look through the various wireless setting for *SSID* and passkey (or something similar). We're sorry, but we can't offer too much more because each router will be different.

Initial Setup

The power is ***off*** at this point.

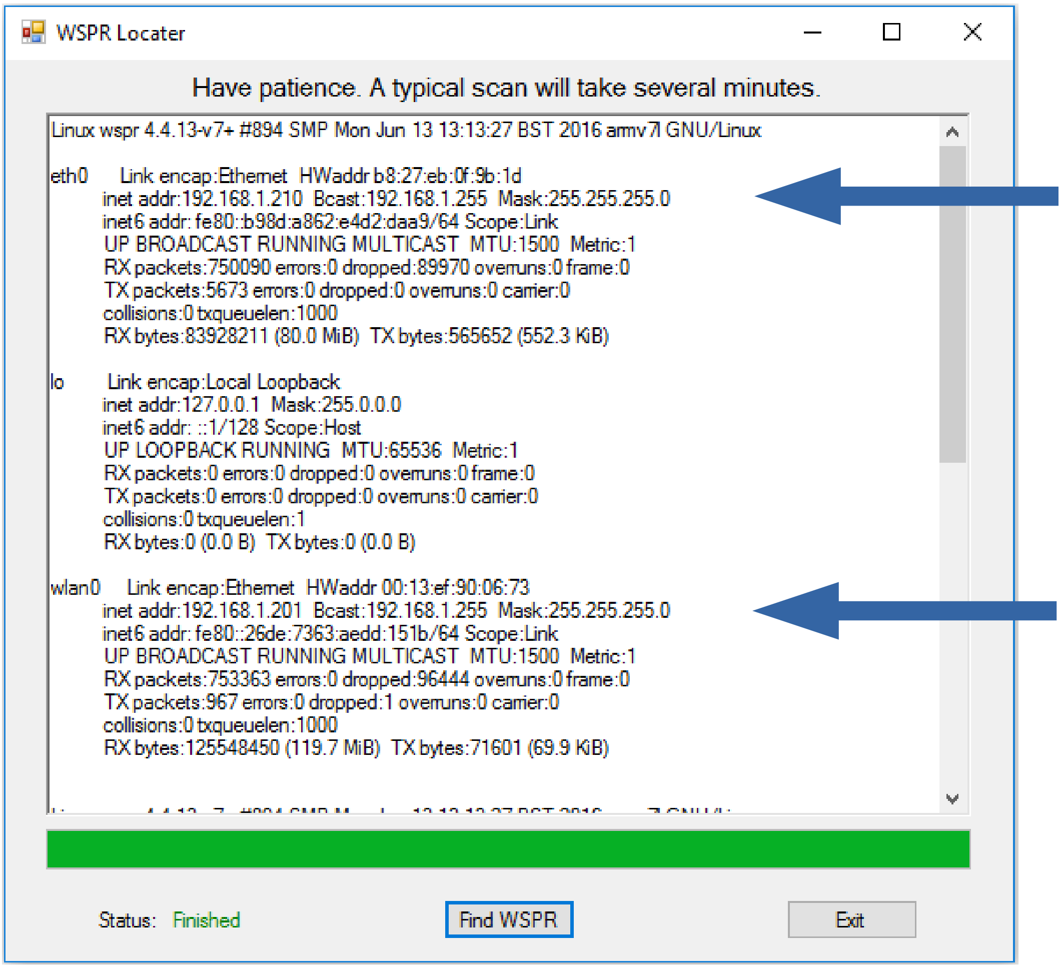
1. Make sure that the power is turned **off**.
2. Connect the Pi to your router with the Ethernet cable. This part is a little tricky. The Pi has to be connected to a DHCP server. That's normally your router. *It won't work if you plug the Pi into your laptop, at least not without some advanced setup.*

Locate the Pi IP Address

1. Working from your PC, download *WSPR Executables* (zip file) from the web site (*WSPR-Executables*), unzip the zip file, and extract the program WSPR_Locate.exe to a suitable work area.
2. Execute WSPR_Locate.exe. The program will take about five minutes to execute and will (hopefully) give you the IP address of the Pi.

   Some assumptions have been made: 1) your network is a private Class C network (IP addresses begin with 192.168..), and 2) you have DHCP enabled on your router (router hands out IP addresses). If you have no idea what we're talking about then don't worry about it – these are typical settings for a home network.

WSPR Without Tears



```
Linux wspr 4.4.13-v7+ #894 SMP Mon Jun 13 13:13:27 BST 2016 armv7l GNU/Linux

eth0      Link encap:Ethernet  HWaddr b8:27:eb:0f:9b:1d
          inet addr:192.168.1.210  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::b98d:a862:e4d2:daa9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:750090 errors:0 dropped:89970 overruns:0 frame:0
          TX packets:5673 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:83928211 (80.0 MiB)  TX bytes:565652 (552.3 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0     Link encap:Ethernet  HWaddr 00:13:ef:90:06:73
          inet addr:192.168.1.201  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::26de:7363:aedd:151b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:753363 errors:0 dropped:96444 overruns:0 frame:0
          TX packets:967 errors:0 dropped:1 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:125548450 (119.7 MiB)  TX bytes:71601 (69.9 KiB)
```

Status: Finished

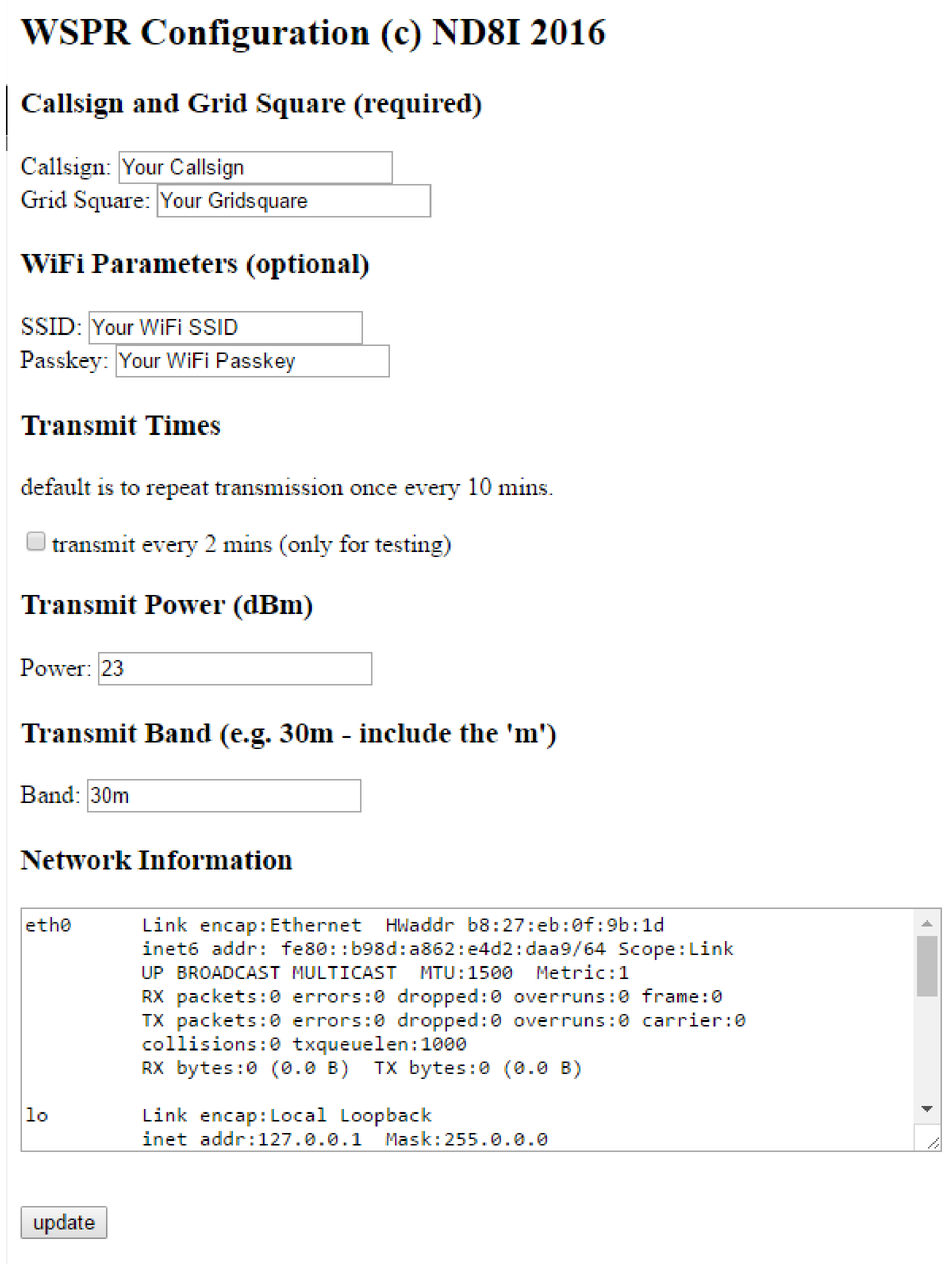
Figure 6: WSPR Locater

With any luck, the display will look something like Figure 6: WSPR Locater. Look for the line starting with *eth0*. The line below it shows *inet addr: 192.168.1.210*. Your setup should be similar except for the last two numbers (it will begin with 192.168, but the last two numbers will probably be different). Record the full number (*192.168.aaa.bbb* – aaa and bbb are the numbers for your Pi) for the next step. That's the IP address of your Pi. If the program fails to find the Pi, try it again. If that fails, go to the Troubleshooting section.

Note: The first time you run WSPR_Locater *inet addr* will be displayed for *eth0*, the Ethernet connection. After setting the WiFi portion of the WSPR configuration (see Figure 7: WSPR Configuration Screen), WSPR_Locater will display *inet addr* for *eth0* if the Ethernet cable is still connected and *inet addr* for *wlan0*, the WiFi connection.

WSPR Without Tears

Setting WSPR Configuration



WSPR Configuration (c) ND8I 2016

Callsign and Grid Square (required)

Callsign: Your Callsign
Grid Square: Your Gridsquare

WiFi Parameters (optional)

SSID: Your WiFi SSID
Passkey: Your WiFi Passkey

Transmit Times

default is to repeat transmission once every 10 mins.

☐ transmit every 2 mins (only for testing)

Transmit Power (dBm)

Power: 23

Transmit Band (e.g. 30m - include the 'm')

Band: 30m

Network Information

```
eth0      Link encap:Ethernet  HWaddr b8:27:eb:0f:9b:1d
          inet6 addr: fe80::b98d:a862:e4d2:daa9/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
```

Figure 7: WSPR Configuration Screen

WSPR Without Tears

1. Open a web browser (Internet Explorer, Firefox, Chrome, …) and type in the IP address you just recorded (192.168.0.67 for example). This will bring up the configuration screen shown in Figure 7: WSPR Configuration Screen -
2. Enter your *callsign* and *gridsquare parameters*.
3. Enter the *SSID* and *passkey* (password) for your WiFi network. These are what you use to enable your cell phone and things like Roku to use your home network. If you don't know what they are you'll have to go digging through your router to figure them out (see the *Troubleshooting* section).

   If you elect to not use WiFi then you don't need to enter SSID and Passkey. You will have to keep the Pi connected to the network by Ethernet cable because the Pi uses the Internet for time synchronization.
4. Check/uncheck the *transmit every 2 mins* checkbox. (I recommend leaving it unchecked.)

   A WSPR message takes just under two minutes to send. The fastest you can send WSPR messages is every two minutes. It's considered bad form to flood the WSPR servers with messages. The default is to send a WSPR message every ten minutes. If you check the *transmit every 2 mins* checkbox then messages will be sent every two minutes. This setting is useful for testing and short term experiments.
5. Set the *Transmit Power* and *Transmit Band* to *23 dBm* and *30m for 30m and 20 dBm and 40m for 40m.*
6. When all done click on the *update* button. This will store the data on the Pi.

Restart Pi

1. Press the pushbutton on the WWoT. Hold for two seconds and release. This commands the Pi to enter shutdown mode (you're always supposed to shutdown a computer properly or the file system can be damaged – pulling the plug is bad).
2. Unplug the power supply, wait 10 sec, then plug the power supply back in and wait until the LEDs on the Pi stop flashing.

Measure RF Output

The LED on the WWoT turns on when the WWoT is transmitting. Wait until the LED is on and measure the voltage across the test points labeled *RF Level* and *Gnd*. This should be in the range of 3-5 volts, indicating the WSPR transmitter is generating full power.

IP Addresses

Ignore this section if you’re not using WiFi.

We have made the assumption that you are using WiFi to connect to the Pi. This is our preference, but it is by no means necessary. What may be confusing is that initially you have to use an Ethernet (wired) connection to talk to the Pi. This is because the WiFi login information has not yet been programmed into the Pi.

After you have set up WiFi (see Figure 7: WSPR Configuration Screen) you will have two valid IP addresses assigned to the Pi. Either one will work. You probably wouldn’t have gone to the trouble of setting up WiFi if you were just going to use Ethernet. So now is the time to remove the Ethernet cable. Your Pi will now have one IP address. There is no advantage to having two IP addresses, so you should remove the Ethernet cable if you’re planning on using WiFi.

WWoT Schematic

The WWoT schematic is shown in Figure 8: WWoT Schematic.

Congratulations! Connect an antenna and you're done.

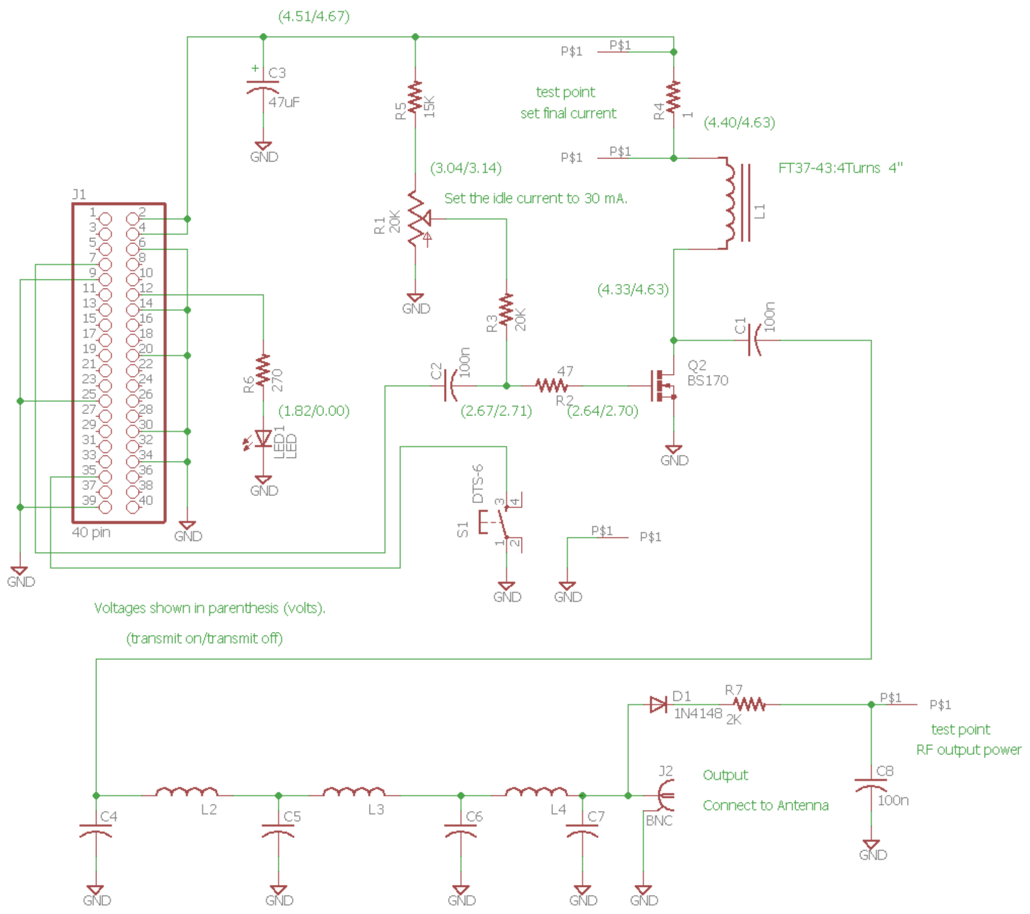


Figure 8: WWoT Schematic

Troubleshooting - It Doesn't Work ...

Well, we'd hoped that you wouldn't ever have to get to this page, but …

We've tried very hard to make sure that everything is correct, however sometimes stuff happens ;-) The good news is it's all fixable. Follow the steps in the order given. We're assuming that something isn't working right at this point, so let's find out what it is.

1. Check that the pcb is plugged into the Pi correctly.
2. Check that you've got the Pi Ethernet cable plugged into the Pi and that the other end of the Ethernet cable is going to your router (not your laptop).
3. Check that the red LED on the **Pi** turns on when power is applied.
4. Verify that the software is working by bringing up the web page (type the IP address into the address line of your browser on your PC/Mac/Linux box: 192.168.aaa.bbb). If the web page displays then the software is probably working okay. The Pi should have power for this test.

   You may have to run the WSPR_Locater program to find the IP address of the Pi and then enter the IP address into the address bar of the browser.

   If the web page doesn't display or WSPR_Locater can't find the Pi then press and hold the shutdown button for 2 sec and unplug the power supply. Wait for 30 sec and plug in the power supply. Wait for the LED's to stop blinking and try to access the web page again. If this fails to work there may be a problem with the software load on the SD card.

5. Press the pushbutton on the WWoT and hold for two seconds. Wait until the LED's on the Pi stop flashing and unplug the power to the Pi.
6. Unplug the power supply and remove the WWoT from the Pi. Look at the board. Is anything odd looking or out of place? Look at both sides of the board.
7. Okay, it's time to reassemble the circuit board to the Pi header, get our your voltmeter and turn on the power. The voltmeter ground lead should be connected to the ground test (*Gnd*) point next to the push button.

We've listed voltages in the schematic (above) as *transmit/idle*, so for example, the voltage on the power supply rail (top wire in the schematic) is shown as (4.51/4.67). This means that we measured 4.51 volts between the power supply rail and ground (*Gnd*) while the unit was transmitting, and 4.69 volts when is was not transmitting (idle).

Keep in mind that we're using a cheap DVM, you're probably using a cheap DVM, and component tolerances could be all over the place. This means that if your readings are within, say 10%-20% of ours that they're probably okay.

We'd suggest printing out the schematic and measuring and recording the voltages listed. Actually, it's probably simpler to just measure the voltages (and write them down) on both sides of R5, both sides of L1, and both sides of R2.

The higher voltage across R5 is the power supply rail. It should be between 4.5-5.0 volts. The voltages on both sides of L1 should be about the same when not transmitting and one significantly lower than the other when transmitting (LED on). The voltages on both sides of R2 should be about the same whether transmitting or not. The voltages at R2 will vary depending on how pot R1 is set. The important part is that they're above 0.0 volts and have pretty much the same voltage on either side of R2.

If you find voltages that differ significantly from what's listed on the schematic, start by looking for bad solder joints. If that fails, let us know because we're probably looking at a bad component, which we will be happy to replace.

Appendix 1 – Communicating With the Pi over SSH

This sorta seems like the antithesis of what we've been trying to accomplish. We wanted to hide the hard details so that you could concentrate on getting a WSPR signal on the air. Okay, you're not afraid of Linux and you want to talk to your Pi over SSH. Actually, we sympathize – that's our favorite mode for talking to the Pi.

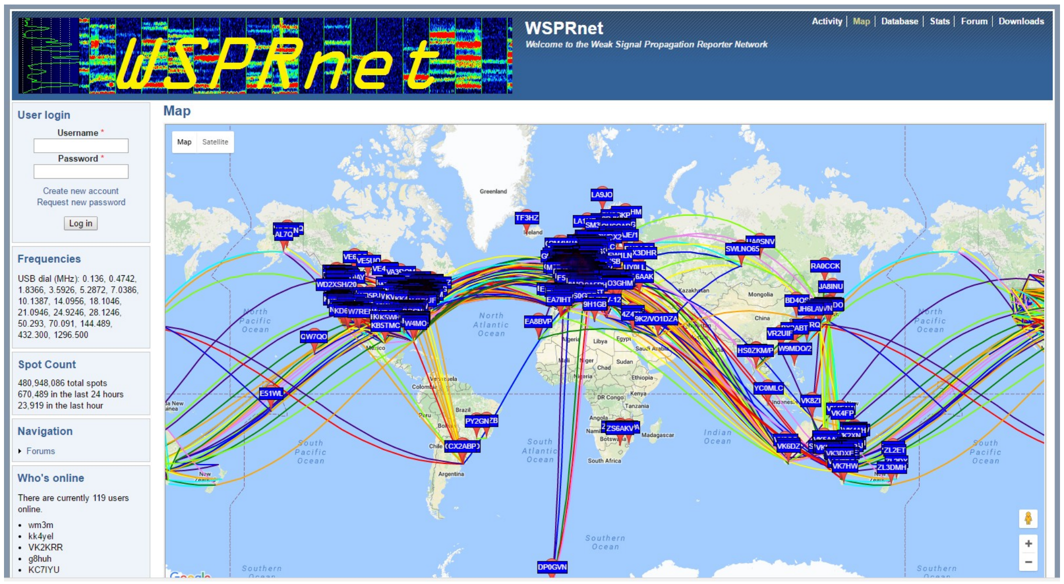
1. Download the program KiTTY.exe (no, not a cat) as part of the *WSPR Executables* from *WSPR-Executables* or from KiTTY.
2. Get the IP address of the Pi from WSPR_Locater.
3. Run the KiTTY and enter the IP address for the Pi. This will open a command window.
4. Username: ***pi***   Password: ***wspr***

WSPR Without Tears

Appendix 2 – Postprocessing Data

The first thing you're going to want to do once your WSPR transmitter is up and running is to see how your signal is getting out. Go to the website *WSPR Net* and click on *Map* in the upper right hand corner.

WSPRnet Map


WSPRnet
WSPRnet
Welcome to the Weak Signal Propagation Reporter Network
Activity | Map | Database | Stats | Forum | Downloads
User login
Username *
Password *
Create new account
Request new password
Log in
Map
Map
Satellite

Frequencies

USB dial (MHz): 0.136, 0.4742, 1.8366, 3.5926, 5.2872, 7.0386, 10.1387, 14.0956, 18.1046, 21.0946, 24.9246, 28.1246, 50.293, 70.091, 144.489, 432.300, 1296.500

Spot Count

480,948,086 total spots
670,489 in the last 24 hours
23,919 in the last hour

Navigation

- Forums

Who's online

There are currently 119 users online.

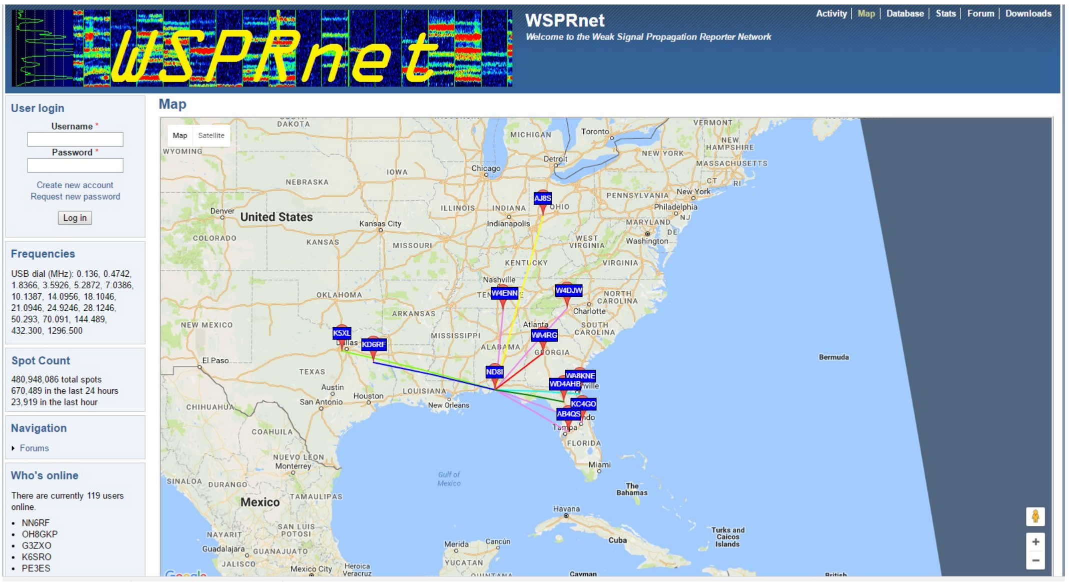
- wm3m
- kk4yel
- VK2KRR
- g8huh
- KC7IYU

Figure 9: WSPRnet Display

Scroll down to the bottom of the graph and change Band as appropriate (30m, 40m, etc), insert your *callsign*, change the *Time Period* to 24 hours, click on the *Day/Night* overlay checkbox if you want to see it and then click on the *Update* button. You can pan and zoom to see how far your signal has gotten.

WSPR Without Tears

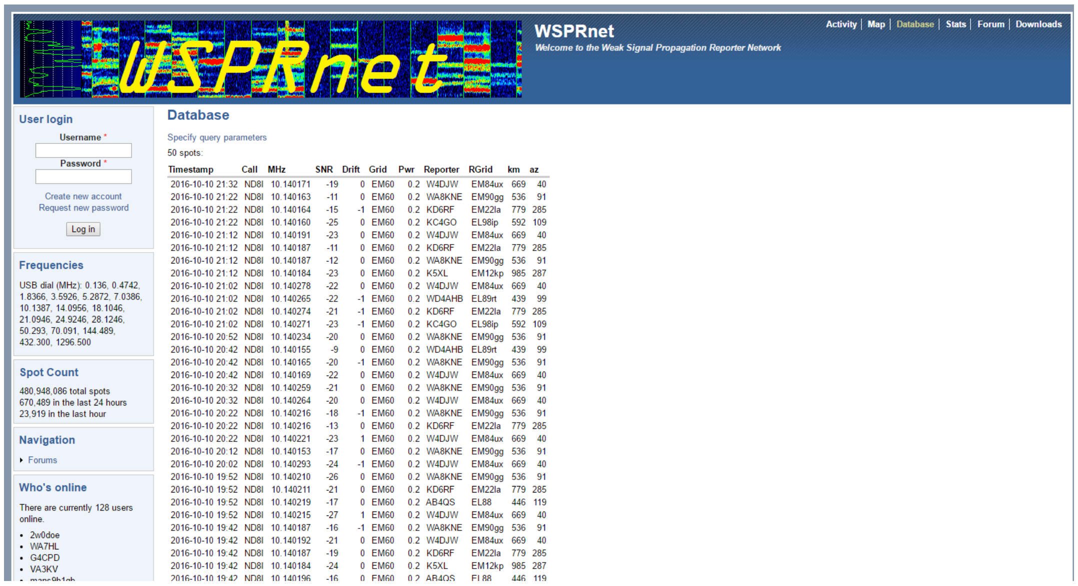
Here's a typical view.



WSPRnet
Welcome to the Weak Signal Propagation Reporter Network
Activity | Map | Database | Stats | Forum | Downloads
User login
Username *
Password *
Create new account
Request new password
Log in
Frequencies
USB dial (MHz): 0.136, 0.4742, 1.8366, 3.5926, 5.2872, 7.0386, 10.1387, 14.0956, 18.1046, 21.0946, 24.9246, 28.1246, 50.293, 70.091, 144.489, 432.300, 1296.500
Spot Count
480,948,086 total spots
670,489 in the last 24 hours
23,919 in the last hour
Navigation
Forums
Who's online
There are currently 119 users online.
NN6RF
OH8GKP
G3ZXO
K6SRO
PE3ES
Map
AJ8S
W4DJW
W4ENN
W4IRG
K5XL
KD6RF
ND8I
WA8KNE
WD4AHB
KC4GO
AB4QS

Figure 10: WSPRnet Display - Zoom

Now click on the *Database* button (next to *Map*) in the upper right hand corner. Click on *Specify query parameters* just below Database in the upper left and change *Band* to the appropriate band (30m, 40m, etc) and add your *callsign*. Click the *Update* button.


WSPRnet

Welcome to the Weak Signal Propagation Reporter Network

Activity | Map | Database | Stats | Forum | Downloads

User login

Username *

Password *

Create new account
Request new password

Log in

Frequencies

USB dial (MHz): 0.136, 0.4742, 1.8366, 3.5926, 5.2872, 7.0386, 10.1387, 14.0956, 18.1046, 21.0946, 24.9246, 28.1246, 50.293, 70.091, 144.489, 432.300, 1296.500

Spot Count

480,948,086 total spots
670,489 in the last 24 hours
23,919 in the last hour

Navigation

- Forums

Who's online

There are currently 128 users online.

- 2w0doe
- WA7HL
- G4CPD
- VA3KV

Database

Specify query parameters

50 spots:

Timestamp	Call	MHz	SNR	Drift	Grid	Pwr	Reporter	RGrid	km	az
2016-10-10 21:32	ND8I	10.140171	-19	0	EM60	0.2	W4DJW	EM84ux	669	40
2016-10-10 21:22	ND8I	10.140163	-11	0	EM60	0.2	WA8KNE	EM90gg	536	91
2016-10-10 21:22	ND8I	10.140164	-15	-1	EM60	0.2	KD6RF	EM22la	779	285
2016-10-10 21:22	ND8I	10.140160	-25	0	EM60	0.2	KC4GO	EL98ip	592	109
2016-10-10 21:12	ND8I	10.140191	-23	0	EM60	0.2	W4DJW	EM84ux	669	40
2016-10-10 21:12	ND8I	10.140187	-11	0	EM60	0.2	KD6RF	EM22la	779	285
2016-10-10 21:12	ND8I	10.140187	-12	0	EM60	0.2	WA8KNE	EM90gg	536	91
2016-10-10 21:12	ND8I	10.140184	-23	0	EM60	0.2	K5XL	EM12kp	985	287
2016-10-10 21:02	ND8I	10.140278	-22	0	EM60	0.2	W4DJW	EM84ux	669	40
2016-10-10 21:02	ND8I	10.140265	-22	-1	EM60	0.2	WD4AHB	EL89rt	439	99
2016-10-10 21:02	ND8I	10.140274	-21	-1	EM60	0.2	KD6RF	EM22la	779	285
2016-10-10 21:02	ND8I	10.140271	-23	-1	EM60	0.2	KC4GO	EL98ip	592	109
2016-10-10 20:52	ND8I	10.140234	-20	0	EM60	0.2	WA8KNE	EM90gg	536	91
2016-10-10 20:42	ND8I	10.140155	-9	0	EM60	0.2	WD4AHB	EL89rt	439	99
2016-10-10 20:42	ND8I	10.140165	-20	-1	EM60	0.2	WA8KNE	EM90gg	536	91
2016-10-10 20:42	ND8I	10.140169	-22	0	EM60	0.2	W4DJW	EM84ux	669	40
2016-10-10 20:32	ND8I	10.140259	-21	0	EM60	0.2	WA8KNE	EM90gg	536	91
2016-10-10 20:32	ND8I	10.140264	-20	0	EM60	0.2	W4DJW	EM84ux	669	40
2016-10-10 20:22	ND8I	10.140216	-18	-1	EM60	0.2	WA8KNE	EM90gg	536	91
2016-10-10 20:22	ND8I	10.140216	-13	0	EM60	0.2	KD6RF	EM22la	779	285
2016-10-10 20:22	ND8I	10.140221	-23	1	EM60	0.2	W4DJW	EM84ux	669	40
2016-10-10 20:12	ND8I	10.140153	-17	0	EM60	0.2	WA8KNE	EM90gg	536	91
2016-10-10 20:02	ND8I	10.140293	-24	-1	EM60	0.2	W4DJW	EM84ux	669	40
2016-10-10 19:52	ND8I	10.140210	-26	0	EM60	0.2	WA8KNE	EM90gg	536	91
2016-10-10 19:52	ND8I	10.140211	-21	0	EM60	0.2	KD6RF	EM22la	779	285
2016-10-10 19:52	ND8I	10.140219	-17	0	EM60	0.2	AB4QS	EL88	446	119
2016-10-10 19:52	ND8I	10.140215	-27	1	EM60	0.2	W4DJW	EM84ux	669	40
2016-10-10 19:42	ND8I	10.140187	-16	-1	EM60	0.2	WA8KNE	EM90gg	536	91
2016-10-10 19:42	ND8I	10.140192	-21	0	EM60	0.2	W4DJW	EM84ux	669	40
2016-10-10 19:42	ND8I	10.140187	-19	0	EM60	0.2	KD6RF	EM22la	779	285
2016-10-10 19:42	ND8I	10.140184	-24	0	EM60	0.2	K5XL	EM12kp	985	287
2016-10-10 19:42	ND8I	10.140196	-16	0	EM60	0.2	AB4QS	EL88	446	119

Figure 11: WSPRnet Database

This gives the particulars for the contacts – their gridsquares and the distance traveled. (Multiply km by 0.62 to get miles.)

Downloading Data

We're fascinated to see how far our signal goes using just a piece of wire strung around the room (that's all we've done with our antenna). It's also interesting to see the effect of daytime vs nighttime on propagation.

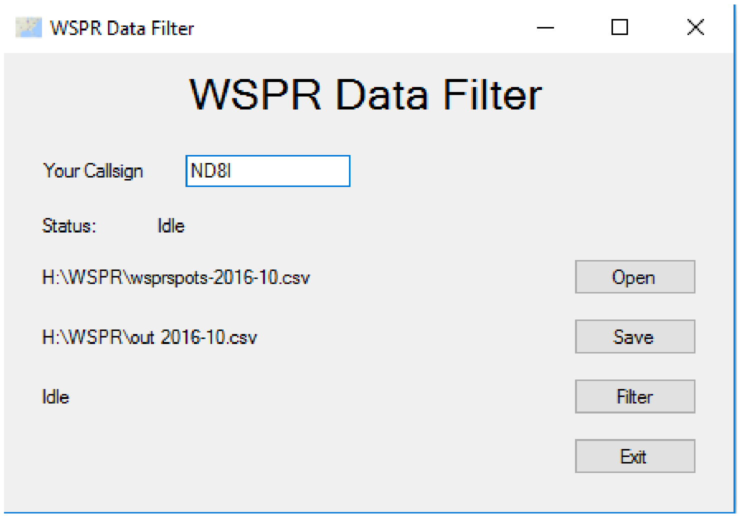
After you've run your transmitter for a while you might be tempted to play with the data some more. The WSPRnet people have stored information on all of the transmissions in files that you can download and process. This is under the *Downloads* tab. The files are stored by *year-month*. They are available in gzip or zip format.

Pick a month that you'd like to investigate and click on the word *zip*. We're amused by what can only be described as wishful thinking on the part of the people who put together the web page. The writeup at the top says "Compressed file sizes range from 1-20MB." The compressed files will be much closer to 300 MB and uncompressed go to about 1.4 GB.

Most of the entries were not information we cared about – they list *all* of the contacts in a particular month. We only want to see our contacts.

WSPR Data Filter

We wrote a program to extract only our contacts. The program is called *WSPR_Data_Filter.exe* and can be downloaded from the web site (part of the executables file). The program removes everything but your contacts from the downloaded data making it a *much* more reasonably-sized file.


WSPR Data Filter

WSPR Data Filter

Your Callsign: ND8I

Status: Idle

H:\WSPR\wsprspots-2016-10.csv

H:\WSPR\out 2016-10.csv

Idle

Figure 12: WSPR Data Filter

WSPR Without Tears

Here is how you use it -

1. Download a month's worth of data from WSPRnet.org.
2. Unzip the file (these are big files – it will take some time).
3. Download Run WSPR_Data_Filter (download from WSPR-Executables, click on *WSPR Executables* and extract the program from the .zip file).
4. Run WSPR_Data_Filter.
5. Enter your *callsign*
6. Click the *Open* button, select the file you unpacked (not the .zip file), and click *Open*.
7. Click the *Save* button, enter a *filename* to save the output as, and click *Save*.
8. Click the *Filter* button.
9. Click the *Exit* button when processing is done.

The output can be read into any spreadsheet program (e.g. Excel) directly for processing and graphing.