

# README

This archive contains the following –

1. *WSPR\_Locate.exe*: Open Source program written in C# that runs on a user's computer and attempts to locate the IP address of the Raspberry Pi (rPi) in the network. The program starts with the IP address of the PC it's running on and sweeps the addresses on the same subnet.

The rPi listens to port 25197 (I figured that was obscure enough that there is little chance of some other program using it). *WSPR\_Locate.exe* tries to establish a connection to this port. If successful, the rPi sends back the output from an *ifconfig* command which includes the IP address of the rPi.

2. *WSPR\_Data\_Reduce.exe*: Open Source program written in C# that strips off (probably) unwanted data.

You can download a month's worth of wspr report data from the *wsprnet.org* website. This will be a fairly large file (say, 300 MB). When it's uncompressed it turns into a very large file (say, 1.4 GB). This is a *comma separated variable* (csv) file that could in theory be read into *Excel*. *Excel* chokes on files this size. Plus, I expect that you're only interested in reports from people who heard *you*.

*WSPR\_Data\_Reduce.exe* has a textbox to enter your callsign. The program filters out any contacts that aren't you. The resulting file is less than 1 MB (mine are usually ~300 KB). This size is much more manageable in *Excel*.

3. *KiTTY.exe*: *KiTTY.exe* is an Open Source (MIT license) SSH client program. It runs on your PC.

Go to the *KiTTY* website ([www.9bis.net/kitty/?page=Welcome&zone=en](http://www.9bis.net/kitty/?page=Welcome&zone=en)) for more information.

You can use SSH to communicate with your rPi. Once you have the IP address of your rPi (acquired from *WSPR\_Locate.exe*), you can login to the rPi. Run *KiTTY.exe* and enter the IP address and you're talking to your rPi. The login username is **pi** and the password is **wspr**.

You talk *Linux* to your rPi on a command line. There are ways to have a graphical interface, but I'm assuming that you're competent at talking *Linux* and can figure out that stuff by yourself.

4. Source Code: I've included the source code the *WSPR\_Locate.exe* and *WSPR\_Data\_Reduce.exe*. This is code that I (Bruce Raymond/ND8I) have written and placed in the public domain (Open Source).

I used two other libraries in this project (*wiringPi* and *WsprryPi*) that are covered by GNU General Pulic License.

5. *SD\_CardFormatter0500SetupEN.exe*: Program for formatting an SD Card. This is the preferred method for reformatting SD Cards. I didn't see any information on license status. I downloaded it from the following web site. I have to assume that if they didn't want us to have it, they wouldn't have made it easily downloadable.

[https://www.sdcard.org/downloads/formatter\\_4/#](https://www.sdcard.org/downloads/formatter_4/#)

This program is useful for cleaning up a messed up SD Card. I've found that sometimes the rPi get confused and takes it out on the SD Card (well, maybe the card is at fault, who knows?). Before just burning the SD Card image on the card, it's useful to clean everything up by running this program.

*WiringPi* ([wiringpi.com](http://wiringpi.com)) greatly simplifies the process of using the I/O pins on the rPi.

*WsprryPi* ([github.com/JamesP6000/WsprryPi](https://github.com/JamesP6000/WsprryPi)) generates the WSPR signal.

I am grateful to the authors of both packages.

I wrote both *WSPR\_Locate.exe* and *WSPR\_Data\_Reduce.exe* in C# and compiled them using Visual Studio Professional 2013. Both programs appear to compile (and run afterward) using *Sharp Develop 5.1* ([www.icsharpcode.net/opensource/sd/download](http://www.icsharpcode.net/opensource/sd/download)). *Sharp Develop* is Open Source software.

Let me know if you run into trouble.

73,

Bruce Raymond/ND8I

[nd8i@arrl.net](mailto:nd8i@arrl.net)