WSPR Data Analysis

Gregg Daugherty (WB6YAZ) <gregg-daugherty@comcast.net>

Jon Chin [cjon08@gmail.com](mailto:cjon08@gmail.com)

8/15/18

WSPR (<https://physics.princeton.edu/pulsar/k1jt/wsjtx.html>) provides a fascinating tool for exploring propagation. Presented here are programs that enable graphical examination of the saved WSPR data for study of HF propagation and for antenna efficiency comparisons. Analysis is based on parsing the ALL\_WSPR.TXT file generated by either the WSJT-X program, or the RedPitaya WSPR hardware (<https://www.redpitaya.com>).

The base scripts are in Python, Labview is used as a front end for launching the scripts and viewing text data. The Python scripts can be launched individually via the command line or via Anaconda Spyder. Anaconda (<https://www.anaconda.com>) is free while Labview home edition (<https://store.digilentinc.com/labview-home-bundle/>) can be purchased for $50 from Digilent. In my past life, we used Labview extensively for automated test and data evaluation. I have uploaded the Python and Labview programs to my GitHub account (https://github.com/greggdaug/WSPR\_analysis)

The first program copies the ALL\_WSPR.TXT file generated by remote RedPitaya hardware to a local folder for processing. As the RedPitaya runs Linux, it is not easy to directly access the file from a Windows machine without installing additional packages on the Linux side. The paramiko python class provides secure FTP methods to allow easy file transfer from Linux to Windows.

The second script generates a world plot showing the transmit locations with colors representing the relative receive SNR at each location. Clicking on a dot provides printing of the underlying data in the Labview front panel or IPython console. Variables are provided to enable setting the amount of data scanned, dates, times, etc. It should be noted that if multiple stations are received at the same grid square location, only the last data point ends up being plotted on the world plot. The data from all stations are printed in the console.

The third script uses the same data file to plot SNR vs location over variables as above. The two scripts used together gives one methods of studying which beacons are accessible as well as a way to parse the data to determine propagation conditions over a set of conditions.

Our python environment of choice is Anaconda (<https://www.anaconda.com/download/>). This free collection of tools provides many of the most used packages as well as an integrated design environment (IDE, spyder), a command line interface and the jupyter notebook. Versions are available for Windows, Linux and MacOS.

Several additional packages are required, the installation procedure for these packages is shown in Figure 1. Figures 2 – 6 show Labview front panels and Python graphical output. Figure 7 is a grid-square plot for reference.

/Reference:

http://www.physics.princeton.edu/pulsar/K1JT/wsjtx-doc/wsjtx-main-1.8.0-rc2.html