OpenCpn : NMEA Input/Output/Multiplexing

https://opencpn.org/wiki/dokuwiki/doku.php?id=opencpn:supplementary_software:nmea_instruments

Nmea Server using Python Script & Launcher_pi

Contributed by Transmitter Dan, reported by R. Gleason

From the Command Line - Once everything is set up:

from command prompt cd C:/python27
Command: > python VDRServer1.py Hakefjord-Sweden.txt 127.0.0.1 2947 .033
Command: > python VDRServer1.py Hartmut-Netherlands.txt 127.0.0.1 2947 .033

Another way to start the Nmea server is to define all the paths to the various files so you don't have to change directory to C:\python\. This example has the Nmea files under C:\Data-Dart\Nmea\ for example:
Command: > c:\python27\python c:\python27\VDRServer1.py C:\Data-Dart\Nmea\Hakefjord-Sweden.txt 127.0.0.1 2947 .033
Command: > c:\python27\python c:\python27\VDRServer1.py C:\Data-Dart\Nmea\Hartmut-Netherlands.txt 127.0.0.1 2947 .033

Download Attached Files

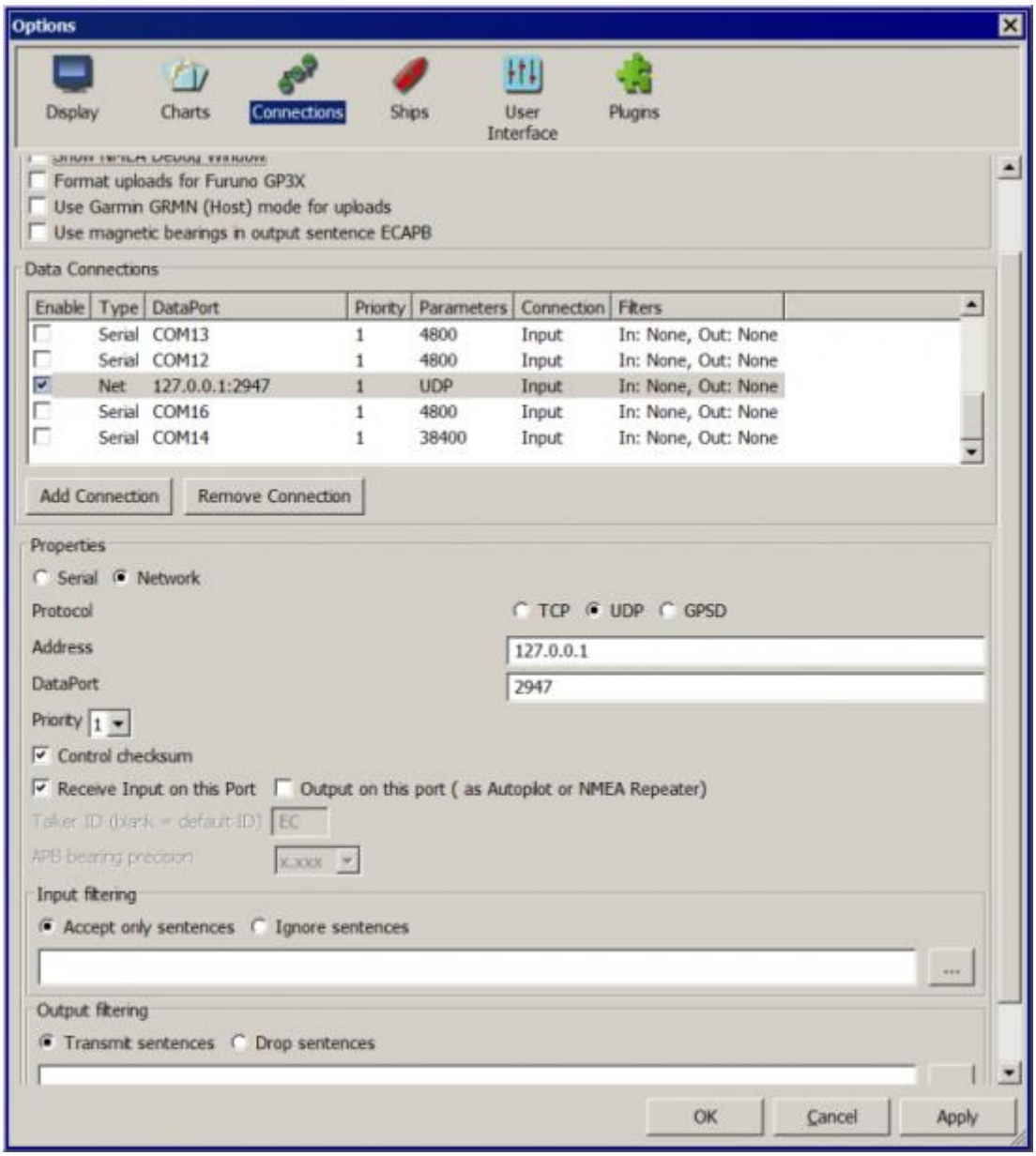
Nmea-Server-Python-Script-README.txt
VDRServer1.py
Hakefjord.txt
Hartmut-Netherlands.txt
These are available at
https://github.com/transmitterdan/VDRplayer
Opencpn Beta File Thingie (Please login with username=rguser, password=rgpass)
Download from the Nmea-Server folder. PLEASE Remove ".TXT" from VDRServer1.py.TXT

Install and Setup:

1. Download and install Python27 Python is a platform independent scripting language interpreter.
2. You can Download Python for Windows here: https://www.python.org/downloads/ "Download Python 2.7.10
3. Execute the python-2-7-10.msi file and install to c:\python27 by default, it will require 95

mb.
4. Copy the file “VDRServer1.py” file into c:\python27
5. Copy the NMEA file Hakefjord-Sweden.txt into into c:\python27
6. Copy the NMEA file Harmut-Netherlands.txt into c:\python27
7. Open a command prompt Start > Run > enter CMD at the prompt type “CD C:\python27”
8. Then enter either command:
Command: python VDRServer1.py Hakefjord-Sweden.txt 127.0.0.1 2947 .033
Command: python VDRServer1.py Harmut-Netherlands.txt 127.0.0.1 2947 .033
9. Leave the command prompt window open to keep the Pyton Server program running.
10. In Opencpn, set up a communications channel for network UDP. This is what the setup screen in O looks like for Windows: In Opencpn Options > Connections > Add connection

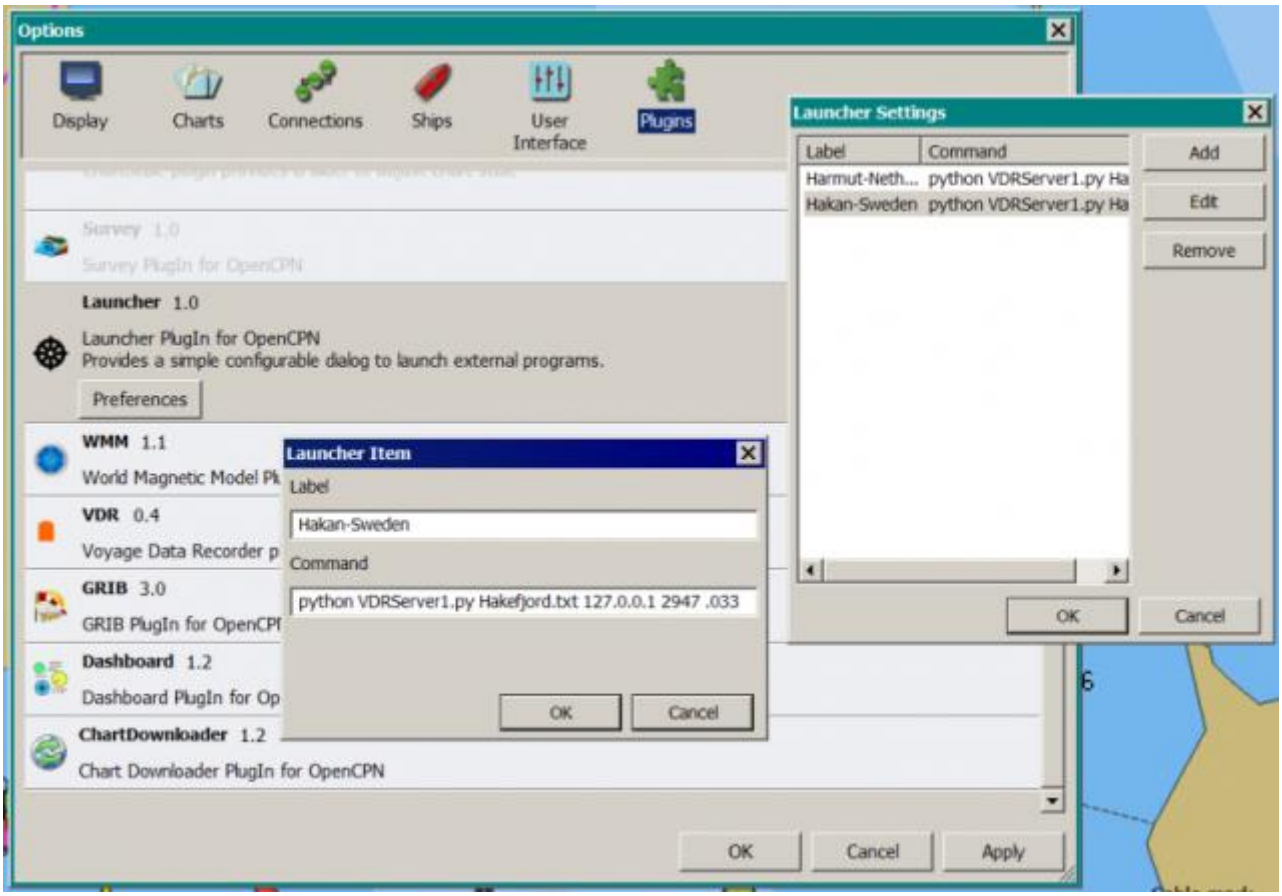
```
Connection Type: Network\\
Protocol: UDP\\
Address: 127.0.0.1\\
Dataport: 2947\\
Priority: 1\\
Control Checksum checked\\
Receive Input checked\\
Output on this port not checked\\
Input Filtering:  Sentences only button\\
```



11. Now you should see the Nmea data being run in Sweden or Netherlands
12. See TransmitterDan's Python script “VDRServer1.py”, which is also below.

Using Launcher_pi to make it easy

If you want to make this even easier, you can download the Launcher_pi plugin and install it. Then under Options > Plugins > Launcher, Enable the plugin.



Then use Launcher > Preferences to make appropriate entries to run various nmea files for testing, eg:

Name : Haken-Sweden
Command: c:\python27\python c:\python27\VDRServer1.py C:\Data-Dart\Nmea\Hakefjord-Sweden.txt 127.0.0.1 2947 .033

Name: Hartmut-Netherlands
Command: c:\python27\python c:\python27\VDRServer1.py C:\Data-Dart\Nmea\Hartmut-Netherlands.txt 127.0.0.1 2947 .033

Under Options > Plugins hit Apply, Ok and then in the main Opencpn screen bring up the Launcher menu.

You should see Hakefjord-Sweden and Hartmut-Netherlands to select. Try one of them. Don't close the command prompt or the nmea file will stop running. Now go setup Opencpn > Options > Connections as in item #10 above.

OpenCPN Connections should have a NET Connection for:

Type: Net **Dataport:** 127.0.0.1 2947 .033 **Parameters:** UDP **Connection:** In/out **Filters:** none

VDRServer1.py

==========

```
import socket
import sys
import time

if len(sys.argv) < 4:
    print("USAGE:")
    print("[python] VDRServer1.py InputFile IP_Address Port# [Sleep time]")
    print("Sleep time is the delay in seconds between UDP messages sent.")
    print("Sleep time defaults to 0.1 seconds")
    sys.exit()

UDP_IP = sys.argv[2]
UDP_PORT = int(sys.argv[3])
filename = sys.argv[1]

if len(sys.argv) > 4:
    delay = float(sys.argv[4])
else:
    delay = 0.1

print(['UDP target IP:', UDP_IP])
print(['UDP target port:', str(UDP_PORT)])

sock = socket.socket(socket.AF_INET, # Internet
    socket.SOCK_DGRAM) # UDP
f = open(filename, 'r')

while True :
    mess = f.readline()
    if len(mess) < 1:
        f.close()
        sys.exit()
    # print(mess)
    mess = mess.strip()
    sock.sendto(mess.encode("utf-8"),(UDP_IP, UDP_PORT))
    time.sleep(delay)
```