Using UDP_Sniffer to monitor data being broadcast to a network

With the issues we have been having with SCS data not passing between the mission and science networks, it can be helpful to verify that a computer can in fact see the SCS broadcast data. I wrote a simple command line python script that will open a UDP port, listen for data, and when data is received, print out some information about the sender, the port the data was received on, and the data itself (if the data can be converted to ASCII.)

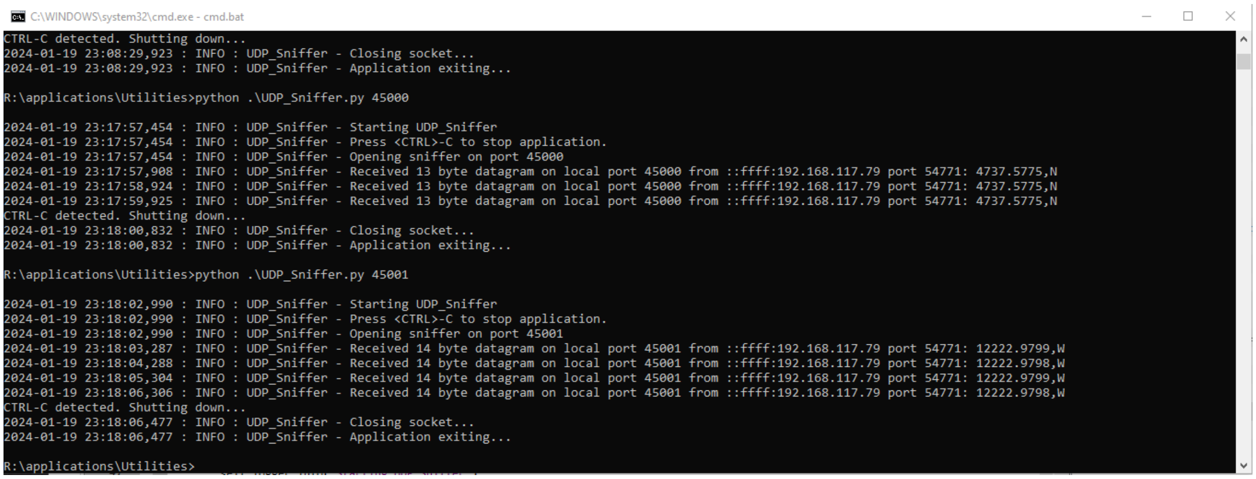
This version of instructions is for use with the compiled version of UDP_Sniffer. This version will run without an installation of Python. Note that the screenshot below shows how to invoke the non-compiled version so the command line will be slightly different. Please follow the text instructions on how to launch the program. The output of the program will be the same which is why I have kept this screenshot.

Open a command prompt and navigate to the `UDP_Sniffer` folder.

Start the UDP_Sniffer program and pass it the UDP port you want to listen to. In this example the program is located in C:\UDP_Sniffer and we will listen to port 45001.

```
C:\UDP_Sniffer>UDP_Sniffer 45001
```

The output will look something like this if you are receiving data:



```
CTRL-C detected. Shutting down...
2024-01-19 23:08:29,923 : INFO : UDP_Sniffer - Closing socket...
2024-01-19 23:08:29,923 : INFO : UDP_Sniffer - Application exiting...

R:\applications\Utilities>python .\UDP_Sniffer.py 45000

2024-01-19 23:17:57,454 : INFO : UDP_Sniffer - Starting UDP_Sniffer
2024-01-19 23:17:57,454 : INFO : UDP_Sniffer - Press <CTRL>-C to stop application.
2024-01-19 23:17:57,454 : INFO : UDP_Sniffer - Opening sniffer on port 45000
2024-01-19 23:17:57,908 : INFO : UDP_Sniffer - Received 13 byte datagram on local port 45000 from ::ffff:192.168.117.79 port 54771: 4737.5775,N
2024-01-19 23:17:58,924 : INFO : UDP_Sniffer - Received 13 byte datagram on local port 45000 from ::ffff:192.168.117.79 port 54771: 4737.5775,N
2024-01-19 23:17:59,925 : INFO : UDP_Sniffer - Received 13 byte datagram on local port 45000 from ::ffff:192.168.117.79 port 54771: 4737.5775,N
CTRL-C detected. Shutting down...
2024-01-19 23:18:00,832 : INFO : UDP_Sniffer - Closing socket...
2024-01-19 23:18:00,832 : INFO : UDP_Sniffer - Application exiting...

R:\applications\Utilities>python .\UDP_Sniffer.py 45001

2024-01-19 23:18:02,990 : INFO : UDP_Sniffer - Starting UDP_Sniffer
2024-01-19 23:18:02,990 : INFO : UDP_Sniffer - Press <CTRL>-C to stop application.
2024-01-19 23:18:02,990 : INFO : UDP_Sniffer - Opening sniffer on port 45001
2024-01-19 23:18:03,287 : INFO : UDP_Sniffer - Received 14 byte datagram on local port 45001 from ::ffff:192.168.117.79 port 54771: 12222.9799,W
2024-01-19 23:18:04,288 : INFO : UDP_Sniffer - Received 14 byte datagram on local port 45001 from ::ffff:192.168.117.79 port 54771: 12222.9798,W
2024-01-19 23:18:05,304 : INFO : UDP_Sniffer - Received 14 byte datagram on local port 45001 from ::ffff:192.168.117.79 port 54771: 12222.9799,W
2024-01-19 23:18:06,306 : INFO : UDP_Sniffer - Received 14 byte datagram on local port 45001 from ::ffff:192.168.117.79 port 54771: 12222.9798,W
CTRL-C detected. Shutting down...
2024-01-19 23:18:06,477 : INFO : UDP_Sniffer - Closing socket...
2024-01-19 23:18:06,477 : INFO : UDP_Sniffer - Application exiting...

R:\applications\Utilities>
```

Currently, SCS is configured to send data from ports 45000-45014

The list of sensor data and port numbers can be found here.