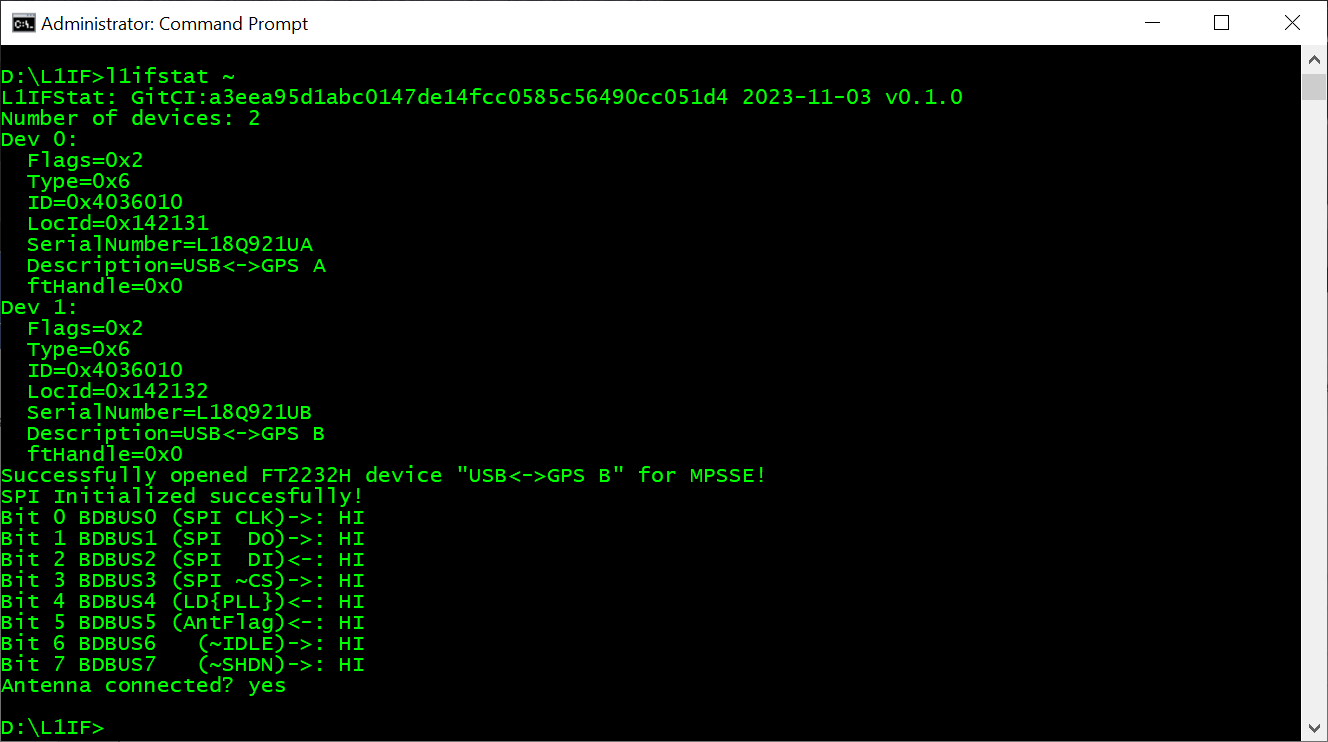
Functional Board Checkout

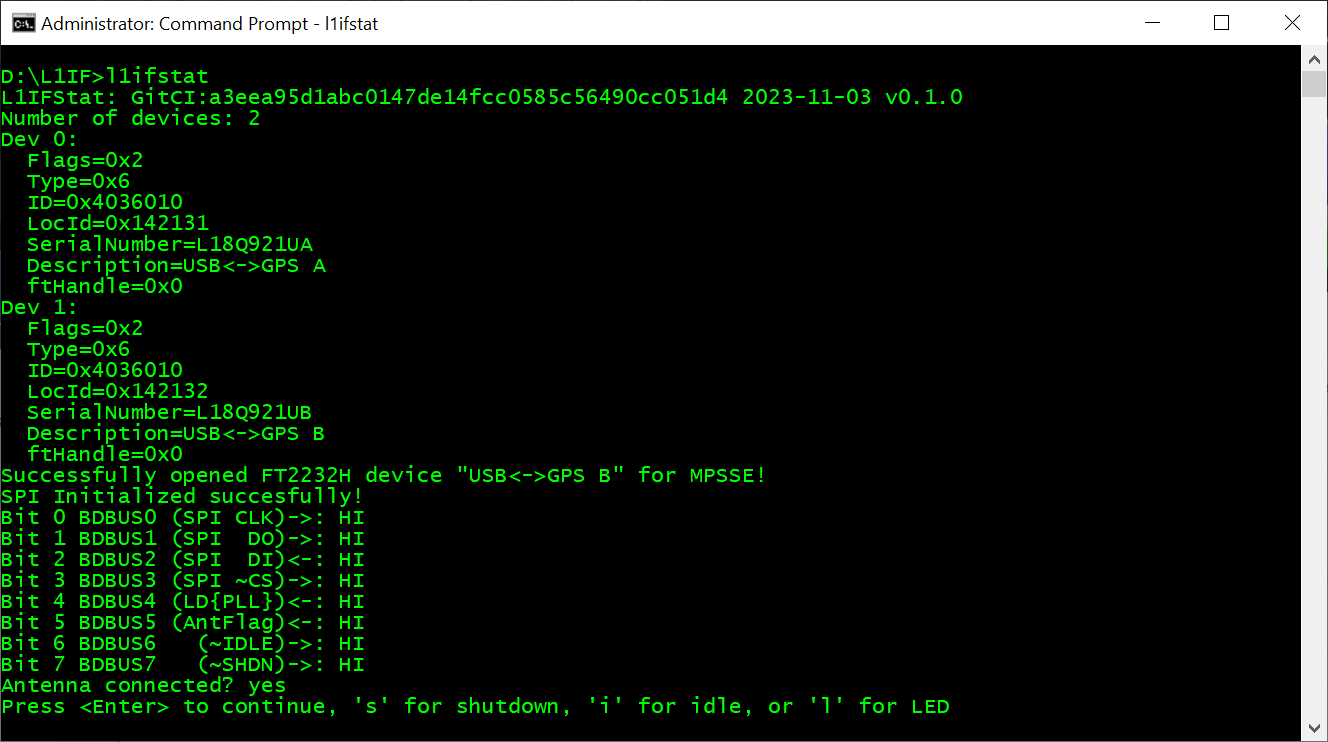
There are three applications that one may use to verify the functionality of the L1IFStream device. Those three applications are L1IFStat.exe, L1IFtap.exe, and L1IFPFSS.exe. The L1IFStat program uses the FTDI D2XX driver to communicate with the MPSSE portion of the FT2232H device. The L1IFtap program uses the FTDI D2XX driver to pull data from the asynchronous FIFO portion of the FT2232H device. The L1IFPFSS program is a Parallel Frequency Search Space application meant to verify the existence of PRN signal data with the collected sample data. The purpose of this document is to illustrate how one may use these three programs to verify the L1IFStream device is working properly.

L1IFStat:

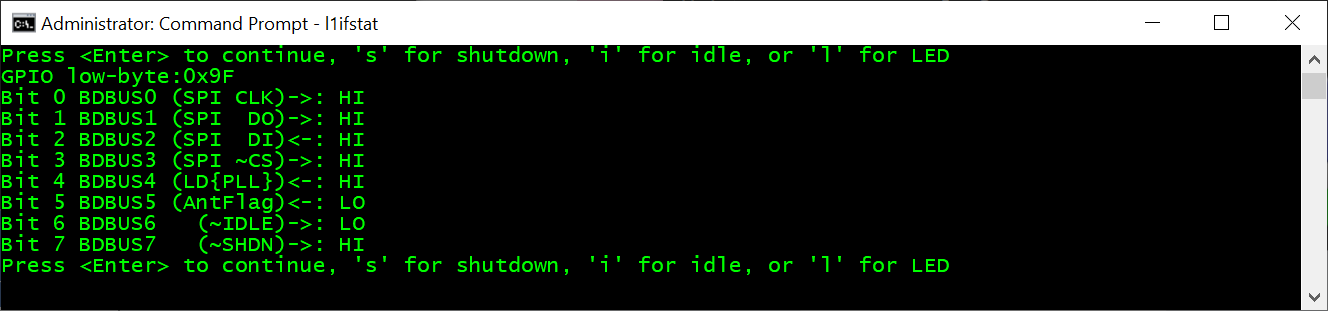
The L1IFStat program may be executed with the ‘~’ argument if interactive mode is not desired. Such instantiation should look like the following:

One can see two devices, and the state of the lower GPIO byte.

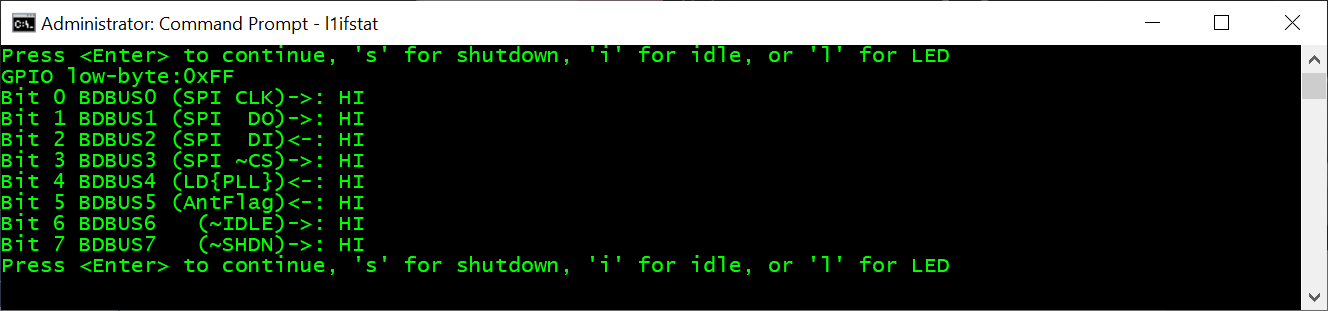
One may also choose to use the L1IFStat program to modify the ~IDLE, ~SHDN, and ~LED state by running L1IFStat.exe with no arguments like the following:



Typing any of the keys shown toggles the GPIO pin for the selected I/O. For example, typing ‘I’:



Then type ‘I’ again to bring the MAX2769B out of Idle:

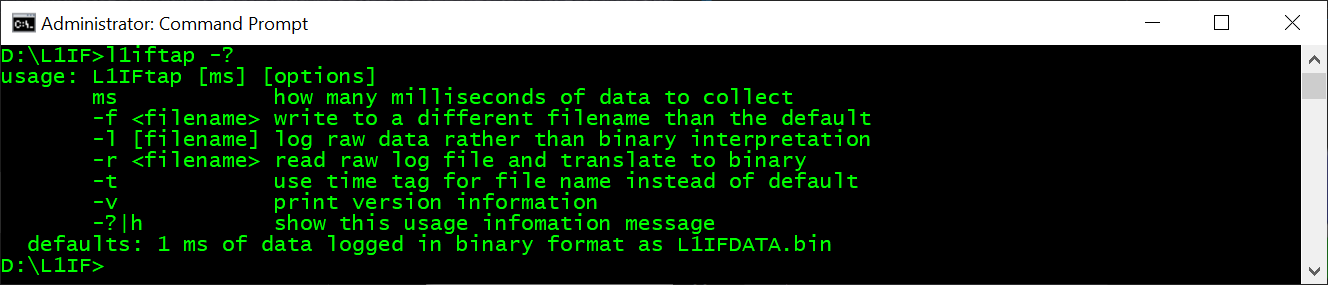


The same may be repeated for shutdown as well as the LED. It may be interesting to have Tera Term open and stream the FIFO data to see the difference between idle and shutdown.

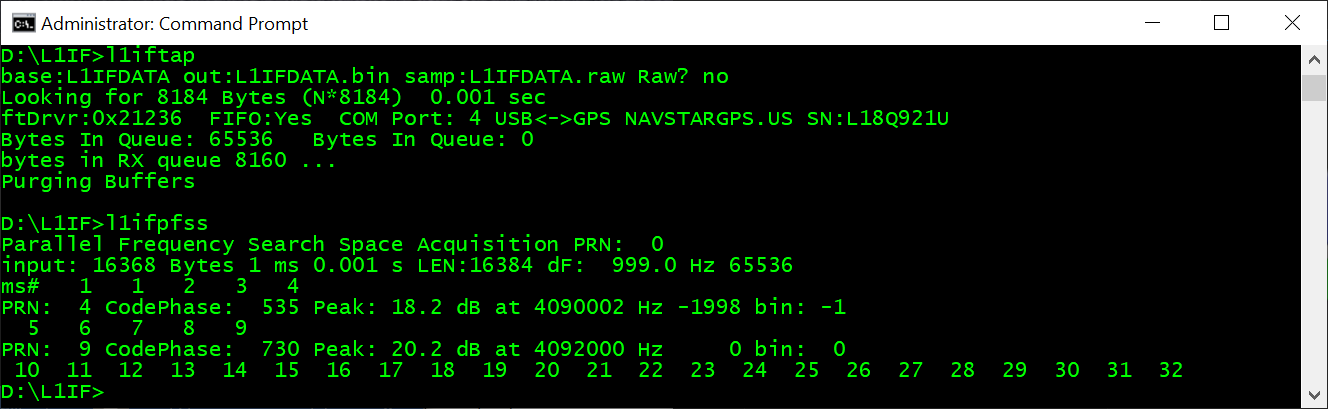
There is a (presently) hidden feature where a key press of ‘p’ will configure the MPSSE for SPI, and send a command to change the clock frequency of the MAX2769. This feature could be useful for developing nonstandard device configurations with the board setup for serial interface mode. The feature could be expanded to read user-supplied configuration files for sending to the device.

L1IFtap:

The L1IFtap program has some user-selectable options. The primary purpose of the L1IFtap program is to collect data for the user. The program may be run with the -? Or -h option to see the available options.



This program may be run with no option at all to collect 1 ms of data in binary format to a file named L1IFDATA.bin. Doing so provides a data file compatible with the parallel frequency search space program’s defaults such that running each one in succession with no arguments is a valid approach to verify whether the board is working as shown below:



In the example one can see 1 ms of data was collected. In this 1 ms of data, PRNs 4 and 9 had sufficient energy to pass an acquisition existence test.

To collect more than 1 millisecond of data with L1IFtap, specify the number of milliseconds worth of data you’d like collected. No command switches are required.

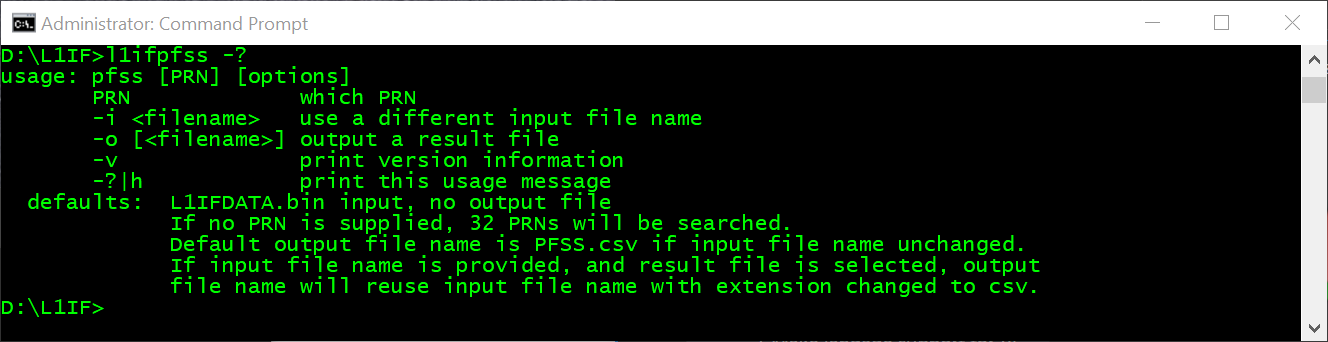
The L1IFtap program also allows one to collect raw data which is the data that has not been expanded into binary format. To collect raw data, provide the number of milliseconds you’d like collected. Using a large number like 60000 is a good test to see whether the buffer can keep up with the device. The program reports how many bytes it expects to collect based on the time supplied which one can verify after the run has completed. Additionally, if 60000 is given, one can watch a clock to see that the program only took 1 minute to run. If the program terminates at about the time requested (after 1 minute,) and the file size is correct, the buffer likely kept up with the device.

The other options enable logging raw data rather than data converted to binary, as well as changing the default filename to a user-defined value. The -t option instructs the application to use a time tag for the filename rather than using either the default or a supplied file name. The time tag filename is an ISO8601 formatted time tag.

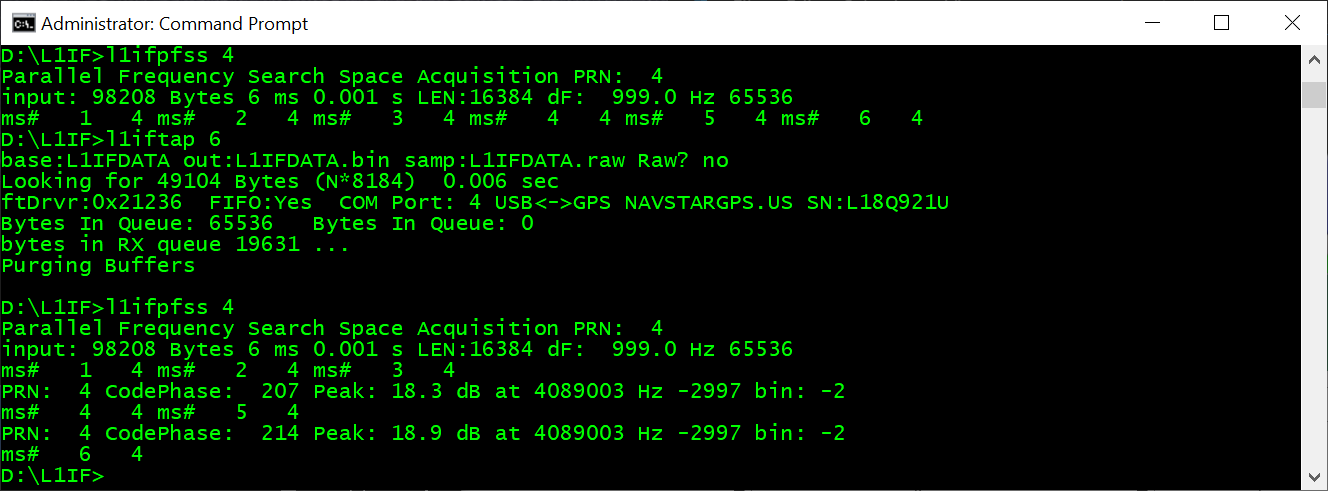
The final option, -r, reads a raw file and converts it to binary. A raw file contains the data as it was received from the MAX2769 unaltered. This format is not compatible with any downstream tools, but is the likely candidate for what is likely to end up in a UDP packet in the not too distant future.

L1IFPFSS:

L1IFPFSS implements a Parallel Frequency Search Space algorithm for inspecting binary collected data. Given no arguments, the program will look for a file called L1IFDATA.bin, and will search 1 PRN at a time through 1 ms of data at a time looking for correlation with PRNs 1 through 32. The defaults may be altered with command line switches which may be seen with the -? Or -h options:

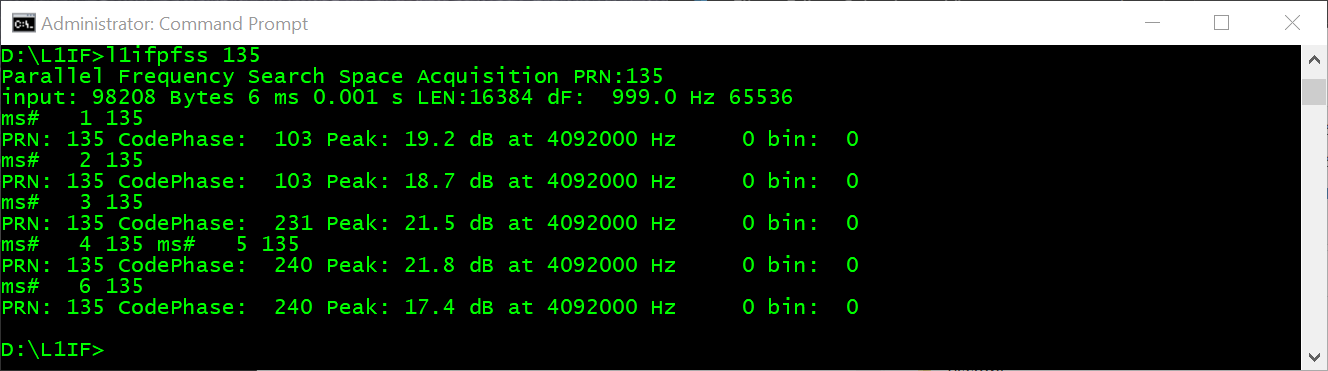


If a PRN is known or suspected to be present, and more than one millisecond of data is collected, the users does not have to endure a search of all the other PRNs, one can simply supply the PRN of the desired satellite to see whether it is present as shown:



If an output file is requested, the result is a csv with PRN in the first column, Doppler bin in the second, Power Spectral Desity (PSD) in the third, the frequency of the highest peak in the bin in the fourth, the frequency difference from IF in the fifth, and the index of the maximum correlation bin in the sixth.

Finally, L1IFPFSS can search for SBAS as shown:



It cannot presently integrate longer than 1 ms.