

Swift Console User's Guide

Overview

The Swift Console is a Graphic User Interface (GUI) for controlling, monitoring and visualizing data from Swift Navigation products. It displays information and allows you to adjust the settings on your hardware. Swift Console can communicate with receivers over serial (RS-232 or USB) or Ethernet interfaces and is available for Windows, Linux and OSX computers.

Refer to this article for Swift Console installation instructions:

<https://support.swiftnav.com/customer/portal/articles/2756825>

Prerequisites

USB to Serial Driver Installation Guide:

<https://support.swiftnav.com/customer/portal/articles/2757197>

Installing Swift Console:

<https://support.swiftnav.com/customer/portal/articles/2756825>

Powering Piksi Multi:

<https://support.swiftnav.com/customer/en/portal/articles/2746937>

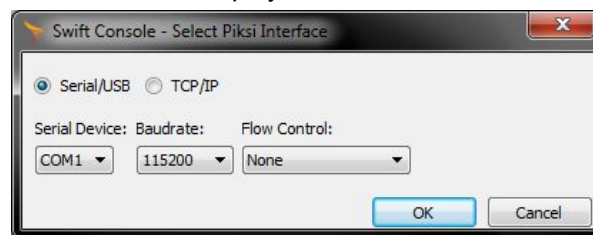
Connecting to Piksi Multi - USB to Serial Adapter:

<https://support.swiftnav.com/customer/en/portal/articles/2747195>

Starting Swift Console

To start the Swift Console double-click on Swift Console icon on the desktop or select it from the programs or applications menu.

When Console is started without parameters it first displays a "Select Piksi Interface" window.



Select a communication interface and its parameter, then press OK button to connect with the device.

Swift Console can be started and connected before the Swift Navigation device has completed its booting process only if RS-232 interface is used. If USB or Ethernet is used, wait for the device complete booting before starting the Console.

Swift Console can also be started with a command line parameters to select interface, enable logging, etc. See Appendix A at the end of this guide for details.

Navigating the GUI

The Swift Console window is divided into four sections as shown on the picture below.



1. **Title Bar:** Displays the Swift Console software version and connection details. Once connected, the communications interface and last six digits of the device serial number are displayed.
2. **Main Window Tabs.** Provides visualization of device data through a series of tabs (described in the the “Main Window Tabs” section).
3. **Console Logging Control:** Displays text messages from the device and Console itself and enables user to start and stop logging:
 - a. Press **CSV Log** button to start recording data from the device in Comma Separated Values (CSV) files. Three CSV files will be saved: baseline, position and velocity.
 - b. Press **JSON Log** button to start recording data from the device in JSON format. For any operation and performance analysis Swift will require a JSON log of the event.

- c. CSV and JSON logs are saved in the folder specified. Press folder button to select a different place to save logs or type it in the folder field.
 - d. Data recording is indicated by depressed button. Press the same button again to stop recording.
4. **Status Bar.** It contains interface information, solution type, number of satellites used in the solution, base station data latency and Device UUID number. Blue icon on the right indicates an active device connection.

The Title Bar, Logging Control and Status Bar are always visible regardless of tab selected.

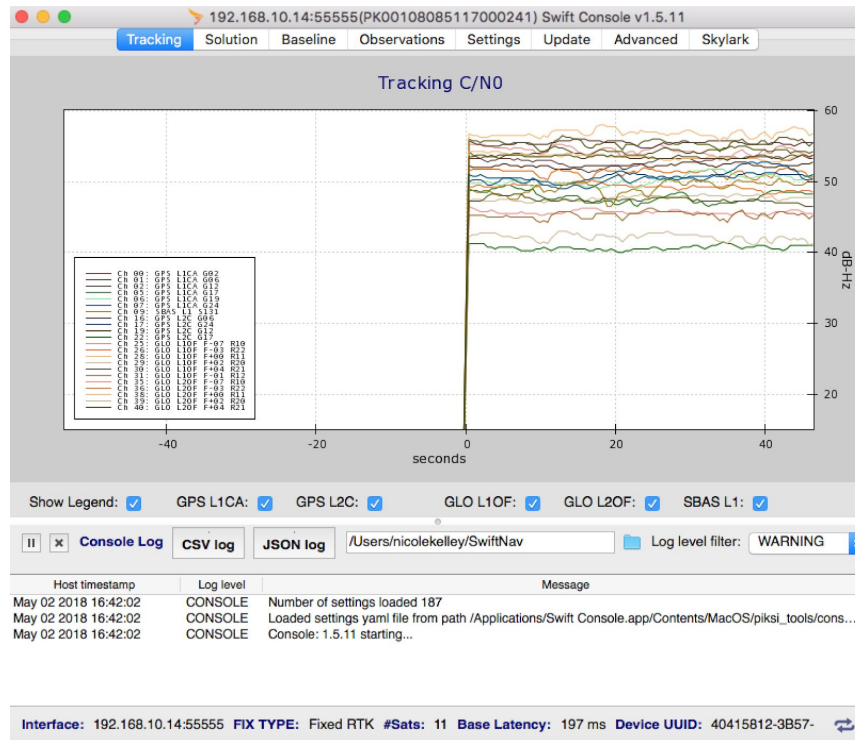
Main Window Tabs

Along the top of the Swift Console window, there are several tabs. When clicked, these tabs provide different views which display device data and controls.

- **Tracking** - displays which satellite signals your device is tracking.
- **Solution** - displays information regarding the absolute position of your device.
- **Baseline** - displays information regarding the high-accuracy RTK position solution of your device.
- **Observations** - displays the GNSS observation data of the local and remote devices
- **Settings** - provides controls to view and modify the settings of your device
- **Update** - provides controls to retrieve and update device firmware and Swift Console
- **Advanced** - provides a few specialized functions for advanced users (System Monitor, IMU, Magnetometer, Networking)
- **Skylark** - provides the unique device UUID information required for skylark activation

Tracking Tab

This tab shows the satellite signals being received (tracked) by the Swift Navigation device.

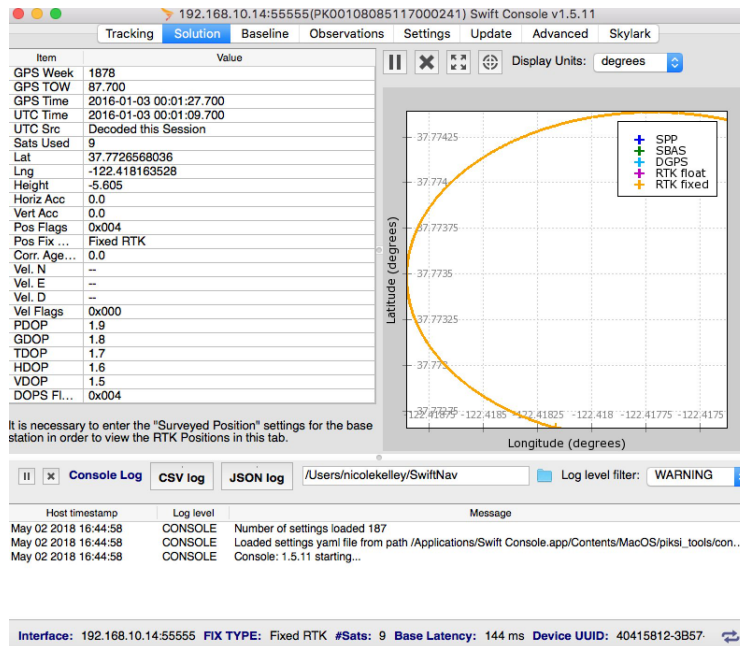


Graph Description

Each signal is represented by a colored line on the graph, and the line's position on the graph represents the strength of the satellite's signal over time. The x-axis is the time and the y-axis is Carrier to Noise Ratio (C/N₀), in dB-Hz, which is the signal strength of the satellite. The most recent time is on the right hand side and the graph scrolls to the left. The higher the value on the y-axis, the stronger the signal.

Solution Tab

This tab shows the device's absolute position, which is the device's location on Earth (latitude, longitude, and altitude/height).



Note: The RTK absolute position of the rover will not be shown unless the base station has entered and is broadcasting its surveyed position.

Graph Description

The Latitude parameter graphed on the vertical axis describes the angular distance north or south of the Earth's equator, in decimal degrees. The Longitude parameter graphed on the horizontal axis describes the angular distance east or west of the Prime Meridian, in decimal degrees.

This graph shows the position solution of the device. The solution type is indicated by color:

- SPP - blue
- SBAS - green
- DGPS - cyan
- RTK float - purple
- RTK fixed - orange

During operation, each receiver will automatically output its single point position, represented as blue points on the graph. The Console will also render an orange line when RTK fixed position is available from the device. An RTK fixed Position Solution is a high precision GPS position solution, with an accuracy of a few centimeters. The blue dots will have less precise positions and will therefore appear as a noisy cluster around the orange RTK fixed line.

Table Description

The table on the left side of the Solution tab displays various metrics regarding the state of the position solution:

Label	Item	Description of Values [units]
GPS Week	GPS Week	Gives the number of whole weeks since the first epoch.
GPS TOW	GPS Time of Week	Gives the number of whole seconds since the beginning of the week.
GPS Time	GPS Time	Displays the current GPS time.
UTC Time	Universal Time Coordinated	Displays the current UTC time
UTC Src	UTC Offset Source	Displays the source of UTC time offset
Sats Used	Number of Satellites	Shows the number of satellites Piksi is tracking.
Lat	Latitude	Represents the angular distance north or south of the Earth's equator.
Lng	Longitude	Represents the angular distance east or west of the Prime Meridian.
Height	Height (Altitude)	Represents height [meters].
Horiz Acc	Horizontal accuracy	Estimated horizontal accuracy [meters]
Vert Acc	Vertical accuracy	Estimated vertical accuracy [meters]
Pos Flags	Position Flags	Describes current solution mode. 0x00 : No fix 0x01 : SPP 0x02 : DGPS 0x03 : RTK float 0x04 : RTK fixed.
Pos Fix Mode	Position Fix Mode	SPP / DGPS / RTK Float / RTK Fixed
Vel. N	Velocity North	Represents the device's Northern velocity [meters per second].
Vel. E	Velocity East	Represents the device's Eastern velocity [meters per second].
Vel. D	Velocity Down	Represents the device's vertical "down" velocity [meters per second].
Vel Flags	Velocity Flags	Describes velocity mode. 0x00 : invalid 0x01 : Measured Doppler derived

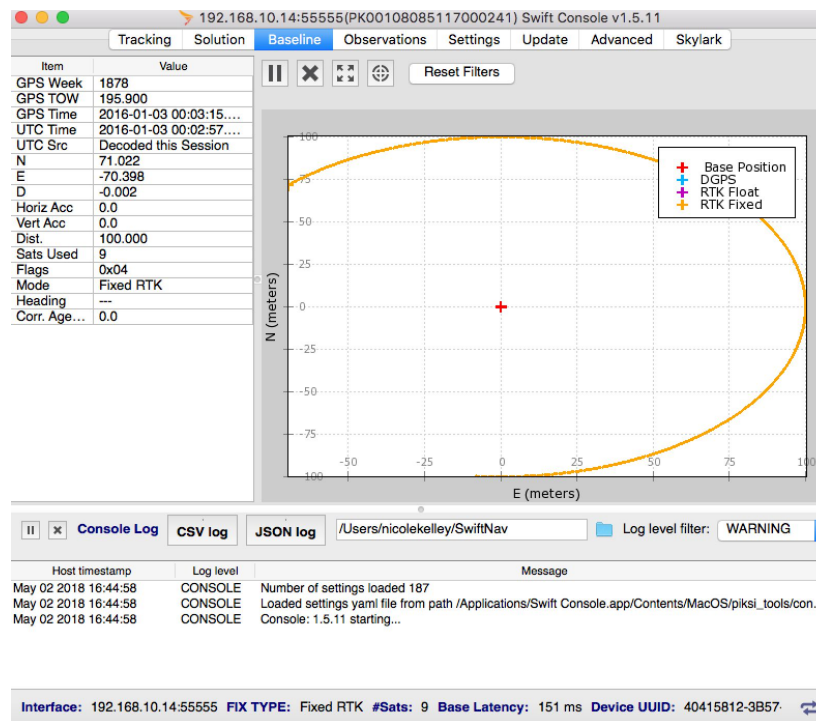
		0x02 : Computed Doppler derived
PDOP	Position Dilution of Precision	Describes the relationship between the error in the device's position and the error in satellite position.
GDOP	Geometric Dilution of Precision	Describes relationship between changes in output position and changes in measured data.
TDOP	Time Dilution of Precision	Represents the measurement of position accuracy through clock offset.
HDOP	Horizontal Dilution of Precision	Represents the measurement of position accuracy through two horizontal coordinates.
VDOP	Vertical Dilution of Precision	Represents the measurement of position accuracy through height.
DOPS Flags	Dilution of Precision Flags	Indicates the position solution with which the DOPS message corresponds. 0x000 : Invalid 0x001 : Single Point Position (SPP) 0x002 : Differential GNSS (DGNSS) 0x003 : Float RTK 0x004 : Fixed RTK

Button Descriptions

Button	Function
Pause	Pauses the output of position solutions on the graph.
X	Erases previous plotted position solutions and starts over plotting new position solutions.
Full Screen	Encompasses all position solutions to be displayed on the graph.
Center Piksi	Centers the current position solution on the center of the graph.

Baseline Tab

This tab shows the relative RTK vector between the base station and a rover.



Graph Description

This graph will show the base station as a red cross, the rover as an orange cross, and the path of the rover as blue dots.

- Within Swift Console, the receiver that is connected to the Console is always the rover and the unit not connected to the Console is always the base.
- The base is always considered to be at coordinate [0,0].
- The rover position data is a relative vector between the base and the rover, given as a distance North (graphed on the vertical axis, in meters), East (graphed on the horizontal axis, in meters), and Down (not graphed).

Table Description

The table on the left side of the Baseline tab displays various metrics regarding the state of the Piksi RTK system:

Label	Item	Description of Values [units]
GPS Week	GPS Week	Gives the number of whole weeks since first epoch
GPS TOW	GPS Time of Week	Gives the number of whole seconds since the beginning of the week
GPS Time	GPS Time	Displays the current GPS time.

UTC Time	Universal Time Coordinated	Displays the current UTC time
UTC Src	UTC Offset Source	Displays the source of UTC time offset
N	North	Displays the north-south relative offset between the base and rover modules of the RTK system [meters].
E	East	Displays the east-west relative offset between the base and rover modules of the RTK system [meters].
D	Down	Displays the up-down relative offset between the base and rover modules of the RTK system [meters].
Horiz Acc	Horizontal accuracy	Estimated horizontal accuracy [meters]
Vert Acc	Vertical accuracy	Estimated vertical accuracy [meters]
Dist.	Distance	Displays the distance between the base and the rover antennas [meters].
Sats Used	Number of Satellites	Gives the number of common satellites that both modules of the RTK system are simultaneously tracking.
Mode	Fix mode	Invalid / DGNSS / Float RTK / Fixed RTK
Flags	Flags	Debug
Corr. Age [s]	Correction Age	Time elapsed since the last correction data received [seconds]

Button Descriptions

The first four buttons are described in the *Solution Tab* section. The remaining button on the right side of the Console is described below:

Button	Function
Reset Filters	Resets both the "float filter" and the "integer ambiguity test filter".

Observations Tab

This tab displays raw GPS observation data in numerical form. The upper table shows data measured by the local device (the one connected directly to your computer). The lower table shows data received from the remote device - usually the base station.

192.168.10.14:55555(PK00108085117000241) Swift Console v1.5.11

TrackingSolutionBaselineObservationsSettingsUpdateAdvancedSkylark

Local

Week: 1878 TOW: 260.000 Total: 9 GPS L1CA: 7 GPS L2C: 2

GPS L1CA: ☒ GPS L2C: ☒

PRN	Pseudorange (m)	Carrier Phase (cycle)	C/N0 (dB-Hz)	Meas. Doppler (Hz)	Comp. Doppler (Hz)	Lock	Flags
203 (GPS ...)	23267645....	12227867...	37.2	2834.84	---	15	0x008F = P...
203 (GPS ...)	23267637....	9528557...	37.2	2208.97	---	15	0x008F = P...
205 (GPS ...)	23753183....	12482820...	37.5	861.12	---	15	0x008F = P...
207 (GPS ...)	20837738....	10950959...	41.5	725.84	---	15	0x008F = P...
207 (GPS ...)	20837734....	8533496...	41.5	565.59	---	15	0x008F = P...
208 (GPS ...)	20843626....	10954276...	42.0	-840.91	---	15	0x008F = P...
209 (GPS ...)	20459804....	10751756...	42.0	-1341.88	---	15	0x008F = P...
210 (GPS ...)	24247645....	12742421...	36.8	2954.00	---	15	0x008F = P...
211 (GPS ...)	24253633....	12746130...	36.5	-2211.63	---	15	0x008F = P...

Remote

Week: 1999 TOW: 344964.000 Total: 22 GPS L1CA: 6 GPS L2P: 6 GLO L1OF: 5 GLO L2OF: 5

GPS L1CA: ☒ GPS L2P: ☒ GLO L1OF: ☒ GLO L2OF: ☒

PRN	Pseudorange (m)	Carrier Phase (cycle)	C/N0 (dB-Hz)	Meas. Doppler (Hz)	Comp. Doppler (Hz)	Lock	Flags
2 (GPS L1...)	20943886....	11006095...	51.0	---	1353.02	15	0x0007 = ...
2 (GPS L2...)	20943882....	8576180...	46.0	---	1054.30	15	0x0007 = ...
6 (GPS L1...)	20739518....	10898698...	51.0	---	-976.47	15	0x0007 = ...
6 (GPS L2...)	20739520....	8492494...	47.0	---	-760.89	15	0x0007 = ...
10 (GLO L...)	20748051....	11059879...	51.0	---	-3241.38	15	0x0007 = ...
10 (GLO L...)	20748059....	8602128...	35.0	---	---	0	0x0007 = ...
11 (GLO L...)	19246544....	10284771...	53.0	---	809.33	15	0x0007 = ...
11 (GLO L...)	19246550....	7999268...	43.0	---	629.48	15	0x0007 = ...
12 (GPS L...)	21030384....	11051546...	53.0	---	1960.89	15	0x0007 = ...
12 (GPS L...)	21030382....	8611595...	44.0	---	1527.96	15	0x0007 = ...
17 (GPS L...)	23533981....	12367194...	45.0	---	-2478.81	15	0x0007 = ...
17 (GPS L...)	23533980....	9636775...	31.0	---	-1931.54	15	0x0007 = ...

Table Description

Each table corresponds to a device in your RTK system. The columns are as follows:

Label	Item	Description of Values [units]
PRN	Pseudo Random Noise code	Represents the Pseudo Random Noise code, which is the unique identifier of each satellite.
Pseudorange (m)	Pseudorange	Displays the raw measurement of the range to the satellite (uncorrected for clock error), measured in meters.
Carrier Phase (cycles)	Carrier Phase	Displays the raw value of the offset of the wavelengths from a nominal value, measured in wavelengths (cycles).
C/N0 (dB-Hz)	Carrier to Noise Ratio	Gives the carrier to noise ratio, measured in dB-Hz.
Meas. Doppler (Hz)	Measured Doppler	
Comp. Doppler (Hz)	Computed Doppler	
Lock	Lock Timer	This value gives an indication of the time for which a signal has maintained continuous phase lock.
Flags	Flags	

Settings Tab

This tab lets you customize various settings of your Swift Navigation device to suit your application. If you change any value in this section, the change will only apply while the device remains powered, unless you save the changes to the flash. Settings can be saved to memory by pressing the "Save to Device" button near the top of the screen.

192.168.10.14:55555(PK00108085117000241) Swift Console v1.5.11

TrackingSolutionBaselineObservations**Settings**UpdateAdvancedSkylark

Name	Value
acquisition	
glonass acquisition enabled	True
sbas acquisition enabled	True
ethernet	
ip config mode	Static
ip address	192.168.10.14
netmask	255.255.255.0
gateway	192.168.0.1
ext event a	
edge trigger	None
sensitivity	0
frontend	
use ext clk	False
antenna selection	Primary
antenna bias	True
imu	
imu raw output	False
imu rate	50
acc range	8g
gyro range	1000
mag raw output	False
mag rate	12.5
nmea	
gpgga msg rate	1
gpgsv msg rate	10
gprmc msg rate	10
gpvtg msg rate	1
gprhdt msg rate	1

Save to Device

Export to File

Import from File

Reset to Defaults

☒ Refresh settings from device

☐ Show Advanced Settings

Clicking any setting on the table to the left on the tab will bring up a description in the right side of the Console. A complete list of the settings - with descriptions - can be found at <http://swiftnav.com/latest/piksi-multi-settings> .

Button Descriptions

The buttons on the right side of the Swift Console are described below:

Button	Function
Save to Device	Saves current set of settings in memory to the persistent storage flash media on device so the configuration will persist between reboots.

Export to File	Exports the complete set of device settings to a configuration file in the .ini format. Note, this will export ALL the settings (even those that are read-only). The file should be hand edited to include only the settings of interest for the application and remove any irrelevant or read-only settings if used to configure the device through the "Import from a file" button.
Import from file	Imports a saved .ini file of device settings. Any errors in the chosen file will abort the operation. It is important to make sure the imported file includes only settings of interest for the intended application. All settings in the file must exist and be writeable to the values in the file for the operation to succeed. Upon success, user will be prompted to choose whether to save settings to device. Read-only settings can be imported without error provided the values in the imported file match exactly the current values.
Reset to Defaults	Erases all settings and resets the device to factory defaults.
Auto Survey	<p>Computes approximate location of the receiver by averaging last 1000 position solutions and stores the resulting values to the Surveyed Position section.</p> <p>This button is visible only when there is a valid SPP fix and after selecting (clicking on) any item from the Surveyed Position section.</p>
Refresh Device Settings	Thus button re-sends the query to discover all settings from device and their current value. Use this button to recover from a situation where the settings table is blank or you wish to confirm the value of a setting.
Show Advanced Settings	Some settings are not intended for typical user interaction with the device and are only visible when the "Show Advanced Settings" checkbox is checked.

Firmware Update Tab

Swift Navigation periodically releases software updates to add features and improve performance. In this tab, you can check if you have the latest version of Swift Console and device firmware. If the firmware on your device is out of date, the Swift Console will automatically prompt you to download and install the latest files.

If you need to update your firmware or Console, follow the instructions here:

Installing Swift Console:

<https://support.swiftnav.com/customer/portal/articles/2756825>

Upgrading Piksi Multi Firmware:

<https://support.swiftnav.com/customer/portal/articles/2757403>

The screenshot shows the 'Update' tab in the Swift Console application. The window title is '192.168.10.14:55555(PK00108085117000241) Swift Console v1.5.11'. The 'Update' tab is selected in the top navigation bar. The 'Hardware Revision' is 'piksi_multi'. The 'Firmware Version' section shows 'Current: v1.4.10' and 'Latest: v1.4.10'. The 'Local File' section has a text input 'Please choose a file' and a button '...'. Below these is an 'Update FW' button. The 'Swift Console Version' section shows 'Current: v1.5.11' and 'Latest: v1.4.6'. The 'Firmware Download' section has a 'Directory' input field with '/Users/nicolekelley/SwiftNav' and a 'Download Latest Firmware' button. The 'Firmware Upgrade Status' section is empty. At the bottom, there is a 'Console Log' section with buttons for 'Console Log', 'CSV log', and 'JSON log'. The log shows three entries: 'May 02 2018 17:09:02 CONSOLE Number of settings loaded 187', 'May 02 2018 17:09:02 CONSOLE Loaded settings yaml file from path /Applications/Swift Console.app/Contents/MacOS/piksi_tools/con...', and 'May 02 2018 17:09:02 CONSOLE Console: 1.5.11 starting...'. The 'Log level filter' is set to 'WARNING'. At the very bottom, a status bar displays: 'Interface: 192.168.10.14:55555 FIX TYPE: Fixed RTK #Sats: 5 Base Latency: 158 ms Device UUID: 40415812-3B57-...'.

Host timestamp	Log level	Message
May 02 2018 17:09:02	CONSOLE	Number of settings loaded 187
May 02 2018 17:09:02	CONSOLE	Loaded settings yaml file from path /Applications/Swift Console.app/Contents/MacOS/piksi_tools/con...
May 02 2018 17:09:02	CONSOLE	Console: 1.5.11 starting...

Interface: 192.168.10.14:55555 FIX TYPE: Fixed RTK #Sats: 5 Base Latency: 158 ms Device UUID: 40415812-3B57-...

Advanced Tab

Additional options for advanced users are detailed below.

- System Monitor - displays overall performance and health of your system.
- IMU - displays raw data from the device's IMU
- Magnetometer - displays raw data from the device's magnetometer
- Networking - presents options for UDP streaming

System Monitor

The *System Monitor* tab displays device system information (CPU and memory utilization), observation latency, and diagnostic information for the communication interfaces.

The *Reset Device* button restarts the device and has the same function as the *RST* button on the Piksi Multi Evaluation board.

The screenshot shows the Swift Console v1.5.11 interface. The top navigation bar includes tabs for Tracking, Solution, Baseline, Observations, Settings, Update, Advanced (selected), and Skylark. The Advanced tab has sub-tabs for System Monitor (selected), IMU, Magnetometer, Networking, and Spectrum Analyzer.

System Monitor

Thread Name	CPU %	Memory
manage acq	64.1	30804
NAP Tracking	25.0	31476
idle	5.0	124
me_calc_pvt	3.0	43020
decode	1.2	1876
starling	0.8	65532
SBP	0.2	65532
rpmsg	0.1	3604
main	0.0	30340

Observation Connection Monitor

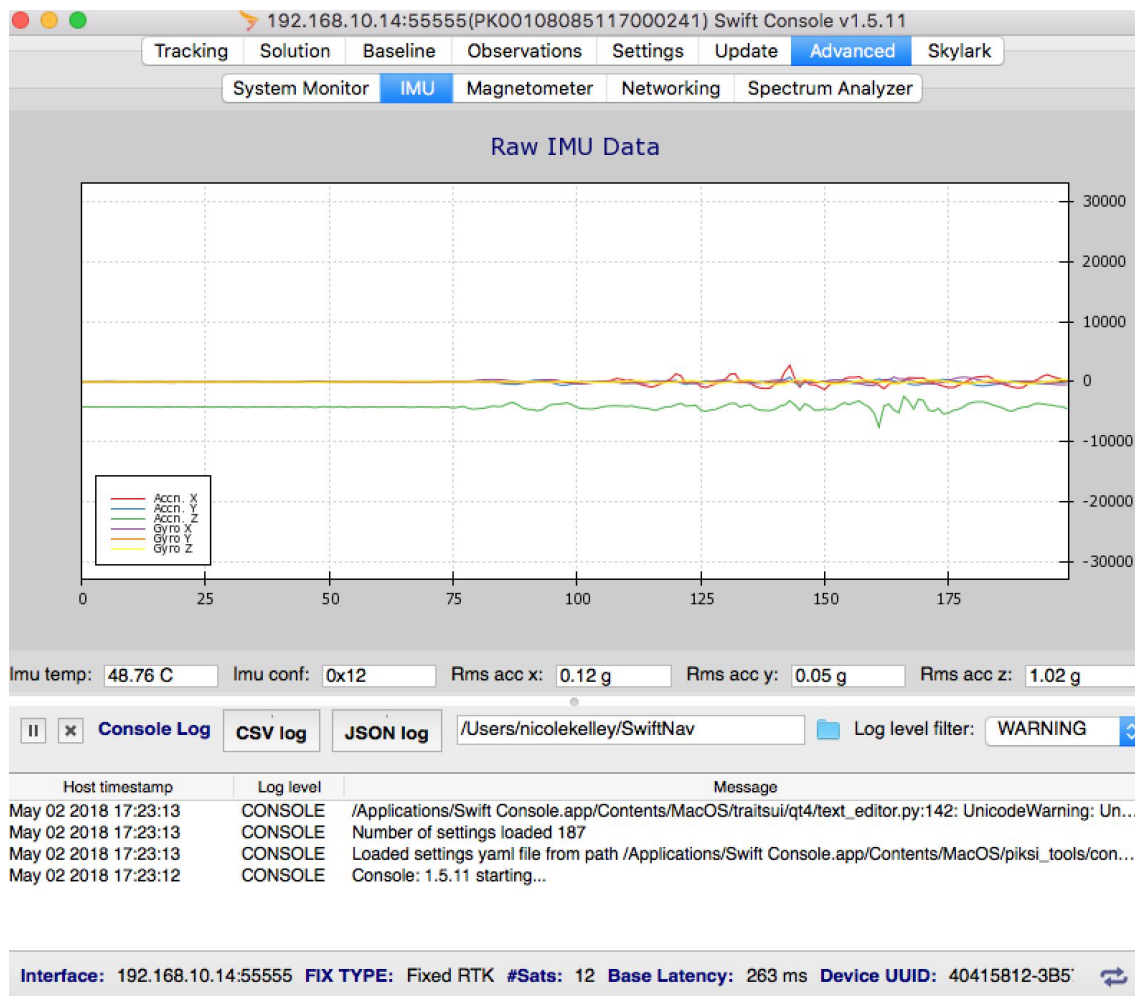
Latency	Period
Curr: 155ms	Curr: 516ms
Avg: 296ms	Avg: 1001ms
Min: 130ms	Min: 92ms
Max: 15322ms	Max: 16011ms

Reset Device button

Console Log

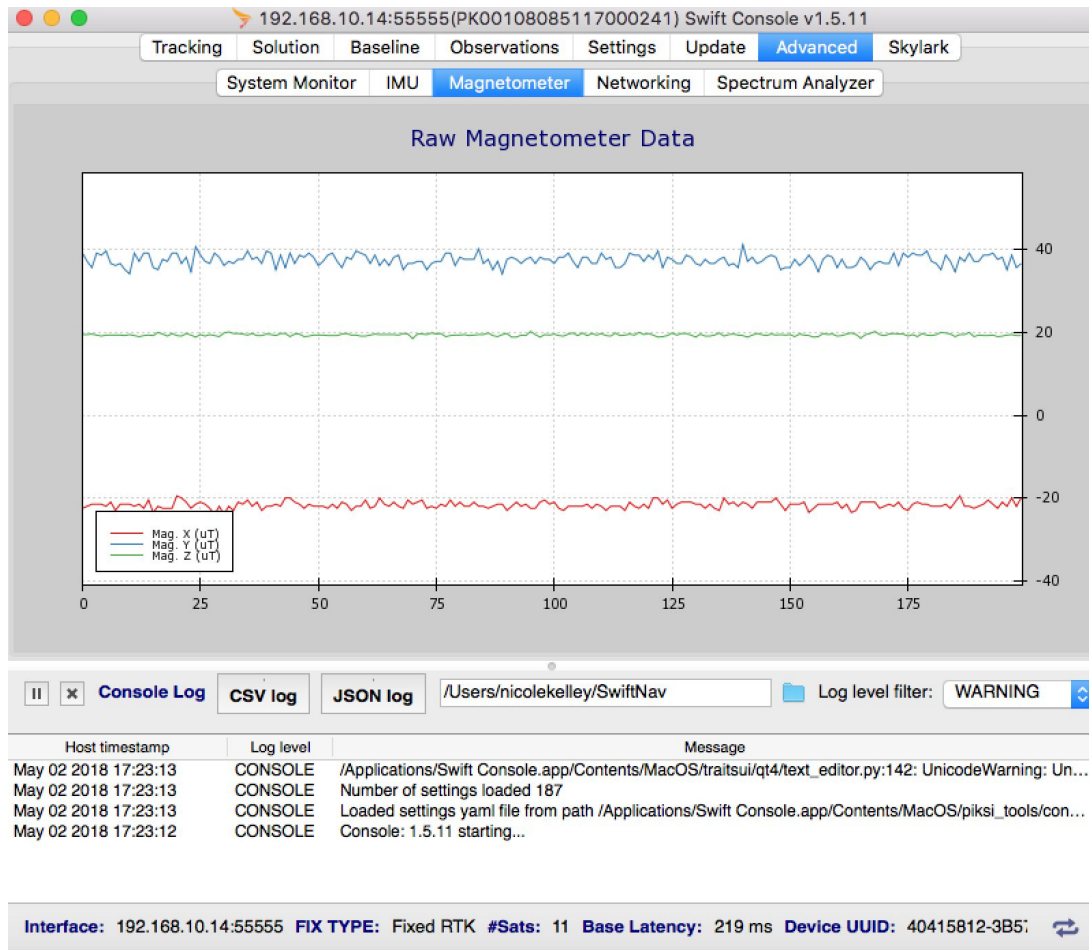
Host timestamp	Log level	Message
May 02 2018 16:44:58	CONSOLE	Number of settings loaded 187
May 02 2018 16:44:58	CONSOLE	Loaded settings yam1 file from path /Applications/Swift Console.app/Contents/MacOS/piksi_tools/con...
May 02 2018 16:44:58	CONSOLE	Console: 1.5.11 starting...

Interface: 192.168.10.14:55555 **FIX TYPE:** Fixed RTK **#Sats:** 9 **Base Latency:** 138 ms **Device UUID:** 40415812-3B5'



Magnetometer

The *magnetometer* tab displays the raw data from device's onboard magnetometer.



Networking

The *Networking* tab displays configuration options for UDP Streaming.

UDP Streaming - Broadcast SBP information received by the Console to other machines or processes over UDP. With the *Observations* radio button selected, the Console will broadcast the necessary information for a rover device to achieve an RTK solution. This can be used to stream observations to a remote device through aircraft telemetry via ground control software such as MAVProxy or Mission Planner.

192.168.10.13:55555(PK00108073817000754) Swift Console v1.5.11

Tracking Solution Baseline Observations Settings Update Advanced Skylark

System Monitor IMU Magnetometer **Networking** Spectrum Analyzer

Messages to broadcast: ☒ Observations ☐ All UDP Streaming

IP Address: 127.0.0.1

Port: 13320

Start Stop

Broadcast SBP information received by the console to other machines or processes over UDP. With the 'Observations' radio button selected, the console will broadcast the necessary information for a rover device to achieve an RTK solution.

This can be used to stream observations to a remote device through aircraft telemetry via ground control software such as MAVProxy or Mission Planner.

Cell Modem Status

Cellular Bandwidth Usage

Data Consumption (Tx/Rx): 33.4KB/47.0KB (ppp0)

Period of Consumption: 0:37:34.400000

Cellular Signal Quality

Signal Strength: -99 dBm

Error Rate (% BER): 0.60

Network

Interface Name	IPv4 Addr	Running	Tx Usage	Rx Usage
eth0	192.168.10.13	True	15.9MB	2.6MB
ppp0	6.201.45.111	True	---	---

Refresh Network Status

Console Log CSV log JSON log /Users/nicolekelley/SwiftNav Log level filter: WARNING

Host timestamp	Log level	Message
May 02 2018 17:43:28	CONSOLE	ppp0
May 02 2018 17:43:28	CONSOLE	sit0
May 02 2018 17:43:28	CONSOLE	eth0
May 02 2018 17:43:28	CONSOLE	lo
May 02 2018 17:43:20	CONSOLE	Number of settings loaded 187
May 02 2018 17:43:20	CONSOLE	Loaded settings yaml file from path /Applications/Swift Console.app/Contents/MacOS/piksi_tools/cons.

Interface: 192.168.10.13:55555 FIX TYPE: Fixed RTK #Sats: 12 Base Latency: 120 ms Device UUID: 0CABBC08-9EFD

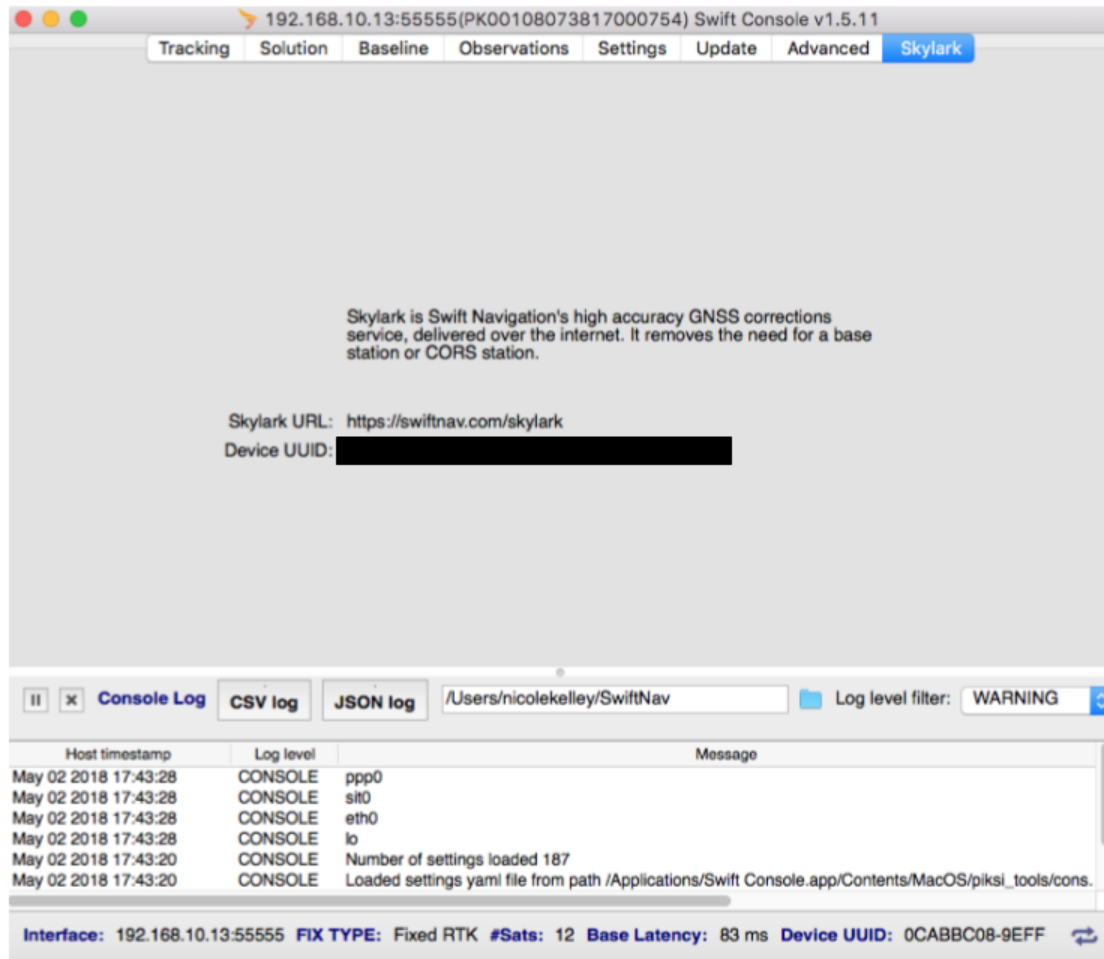
Skylark Tab

Skylark was built from the ground up to deliver high-precision position to autonomous vehicles via a cloud-based corrections service. Its revolutionary approach eliminates the complexity of deploying and maintaining GNSS networks.

To register your Swift device, you will require the UUID shown in this section and sign up for Skylark. For further information, go to <https://www.swiftnav.com/skylark>

To learn more about how to setup and configure your Swift Device on the Skylark Network, we recommend to go through the following guide :

<https://support.swiftnav.com/customer/en/portal/articles/2929676>



Appendix A - Swift Console Command Line Options

```
usage: console [-p PORT] [-b BAUD] [-v] [-f] [-l] [-r] [-o LOG_DIRNAME]
              [--logfilename LOGFILENAME] [--append LOG_FILENAME]
              [--file] [-d TAGS] [-t] [-i INITLOGLEVEL] [-u]
              [--toolkit TOOLKIT] [--error] [--log-console]
              [--skylark SKYLARK] [--serial-upgrade] [-h]
```

optional arguments:

```
-p PORT, --port PORT    specify the serial port to use.
-b BAUD, --baud BAUD    specify the baud rate to use.
-v, --verbose           print extra debugging information.
-f, --ftdi              use pylibftdi instead of pyserial.
-l, --log               serialize SBP messages to autogenerated log file.
-r, --reset             reset device after connection.
-o LOG_DIRNAME, --log-dirname LOG_DIRNAME
                        directory in which to create logfile.
--logfilename LOGFILENAME
                        filename to use for log. Default filename with date
                        and timestamp is used otherwise.
-a APPEND_LOG_FILENAME, --append-log-filename APPEND_LOG_FILENAME
                        file to append log output to.
--file                  Read with a filedriver rather than pyserial.
-d TAGS, --tags TAGS    tags to decorate logs with.
-t, --tcp               Use a TCP connection instead of a local serial port.
                        If TCP is selected, the port is interpreted as
                        host:port
-i INITLOGLEVEL, --initloglevel INITLOGLEVEL
                        Set log level filter.
-u, --update            don't prompt about firmware/console updates.
--toolkit TOOLKIT       specify the TraitsUI toolkit to use, either 'wx' or
                        'qt4'.
--error                 Do not swallow exceptions.
--log-console           Log console stdout/err to file.
--skylark SKYLARK       key value pairs to pass to sbp_relay_view initializer
                        for skylark
--serial-upgrade        Allow software upgrade over serial.
-h, --help              Show usage help in a GUI popup.
```