Terp Rockets Ground Station User Guide

Joseph Hauerstein
Version 2.0.0-beta
January 2024

Table of Contents

Introduction

Welcome to the Terp Rockets Ground Station User Guide. This guide will explain the main functions of the ground station and how to use them. The ground station was developed to support launch operations for the Terrapin Rocket Team competition rocket, primarily logging and displaying telemetry received via radio transmission. It uses Automatic Packet Reporting System (APRS) messages for receiving telemetry data, and an APRS-like format for sending commands. While it was designed for a specific purpose, the goal was to make it as general as possible so that it could be adapted to handle telemetry and data from many different types of rockets. As a secondary function, the ground station is also able to receive and display live video. Through the use of a multiplexer, it can ingest a configurable number of data streams in different formats over a single USB cable.

Installation

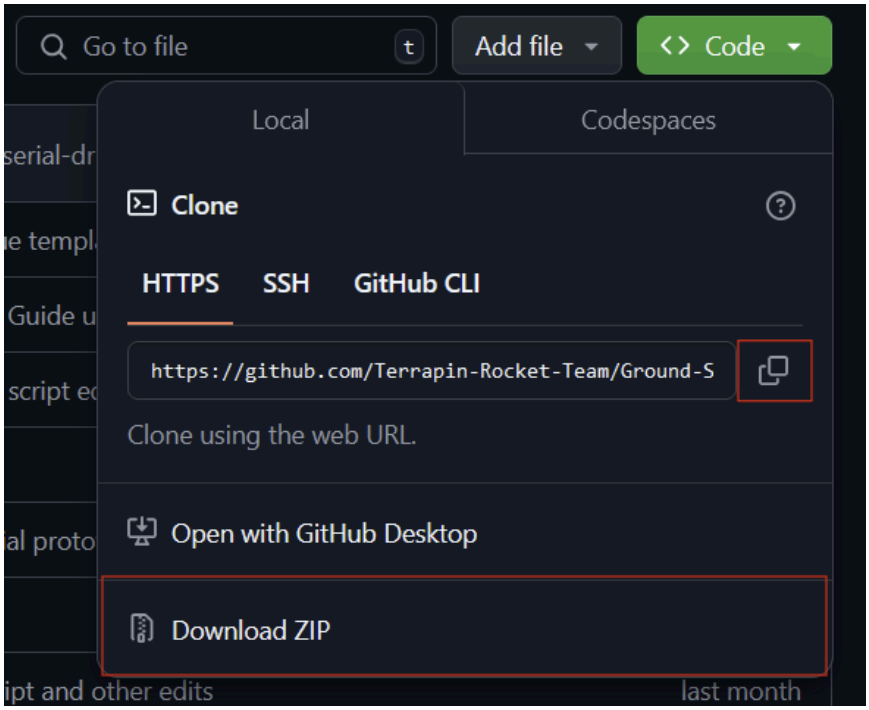
There are two supported ways to install the ground station: from a release, or from source. New releases are infrequent and may not contain a binary for your operating system, so installing from source will often be necessary. While both methods will be explained, it is generally preferred to install from a release if possible.

From a Release

To install from a release navigate to the page for the latest release on the ground station Github page. Here there will be a list of zip files for various platforms. If there is a zip file for your platform, you may download the zip file and extract it to any location on your computer to install the ground station. To start the ground station, simply launch the “Terp Rockets Ground Station” executable.

From Source

To install the ground station from source, clone the Git repository or download the source code from the Github page. You can do this by navigating to the ground station Github page and clicking on the green “Code” button, then copying the clone link, or clicking “Download Zip”. Note which directory you place the ground station source code in, as this will be used later.



The build process has been tested on Windows and Linux, but has not yet been tested on MacOS. Before beginning the build process, make sure to install the latest versions of Node.js and NPM for your operating system.

Windows

Before building on Windows, Mingw-w64 must be installed, which can be found here. Make sure you install it to a directory that your user has full permissions on, or the build scripts will require administrator privileges to run. Note the directory that Mingw/MSYS is installed to (it should contain “msys2.exe” and “mingw64.exe”, among others). This directory will be referred to as `C:\path\to\mingw`. Open “mingw64.exe” and install the required build dependencies using the following command (on one line):

```
pacman -S gcc nasm mingw-w64-x86_64-python3 mingw-w64-x86_64-meson
diffutils make cmake
```

Once all required software is installed, open a terminal in the same directory as the ground station source code. (To do this, you can navigate to the directory in File Explorer and type "cmd" where it shows the file path on Windows.) To see all available build options, you can run the following command:

```
./build.bat C:\path\to\mingw help
```

To start the full build, run the command:

```
./build.bat C:\path\to\mingw all
```

Note that during the first run, the build may take over an hour to complete because FFmpeg and dav1d need to be built. However, after the initial build, rebuilding the ground station if necessary should be much quicker. Once the build command is complete, navigate to the "build/src" directory in the root folder. This directory contains the executable for the ground station, usually called “Terp Rockets Ground Station.exe”. Other folders in “build” will contain the output from building the dependencies. From this point, the ground station is ready to run and the folder containing the executable can be placed anywhere on your computer.

Linux

The ground station requires a Bash environment to execute the build scripts. As with Windows, the following dependencies will need to be installed and available on the PATH via your distro’s package manager:

- GCC, G++
- CMake
- NASM
- Python
- Meson
- Diffutils (e.g., “cmp” and “diff”)
- Make

Once you have installed all required dependencies, you can run the build script. Navigate to the directory where you placed the ground station source in a terminal. You may have to make the build script executable using the command:

```
chmod +x build.sh
```

To see available build options, you can run the command:

```
./build.sh help
```

To start the full build, run the command:

```
./build.sh all
```

The executable will be located in the “build/src” directory, and will usually be named “Terp Rockets Ground Station”. Note that on Linux it may be required for your user to be added to a particular user group in order to access serial ports. This can usually (but not always) be accomplished by running the following command:

```
sudo usermod -a -G dialout $USER
```

Advanced Options

There are some more advanced features of the build system that are not exposed through the command line interface, but could assist in the build process.

Setting the Mingw Path

On Windows, users can manually set `C:/path/to/mingw` so that they do not need to remember it every time the build is run. This can be accomplished by editing the `MINGW_PATH` variable in “build.bat” to be a string representing the full path to `C:/path/to/mingw`. The first argument passed to the build command will still be ignored, so you will have to place all build arguments after the first argument. For example, once a user has modified the `MINGW_PATH` variable, they could run:

```
build 1 all
```

Instead of

```
build C:\path\to\mingw all
```

Where the “1” is ignored by the script since `MINGW_PATH` has been set manually.

Multicore Builds

While most of the build process does not support multicore builds, some parts, especially the FFmpeg/dav1d build, will finish much faster if more than one core can be used. To set the number of cores to use, open the “build.sh” file in one of the ground station subdirectories, and edit the `CORES` variable to be the number of cores to use in that build. This should be less than the number of cores available on your machine. The ground station subdirectories that support multicore builds are:

- coders
- serial
- utils

Setup

Before use, it is important to first set up the ground station so that it is configured properly and is able to communicate with your device. This section will cover the different configuration options available and explain the application's expected input format.

Application Settings

There are a variety of settings available to configure both the behavior of the application and how it communicates with your device. These settings can be changed on the application's settings page. Use of the settings page will be covered later in this guide, while this section will explain the different configuration options. Each option is listed by its name in the settings page, followed by a short description.

- Main Window Scale: Changes the scale of the debug window for different sized screens.

- Debug Mode: Turn on debug mode to log debug statements and open the devtools with each window.
- Data Debug Input: Read input data from local data files. The normal Serial connection will be non-functional with this setting on.
- Serial Driver Debug: Use the debug serial driver to input data from a Ground Station Muxer file. The normal Serial connection will be non-functional with this setting on.
- Video Mode: Turn on to switch the ground station to video streaming mode.
- Toggle Tile Caching: Turn off caching of map tiles to the disk. If disabled, the map may not work when offline.
    - Note: Functionality currently not implemented
- Map Cache Size: The maximum size of the cache for loading tiles offline.
- Serial Port Baud Rate: The baud rate used when connecting to the radio receiver over serial.

Scale Settings

The Main Window Scale controls the size of the main window, since the window cannot be resized manually. These settings will often be used to change the application scale based on your display resolution. It is recommended to set this value to the same value as the OS level window scaling, usually found in the Display options. Otherwise, the recommended range for this value is between 0.5 and 2.0.

Mode Settings

The Debug Mode, Data Debug Input, Serial Driver Debug, and Video Mode settings are toggles that enable or disable specific application functionality. The debug mode setting, when turned on, will activate many functions useful for debugging the ground station. These include opening the Chromium dev tools used by Electron to assist with debugging GUI elements, logging debug statements. The Data Debug Input toggle will enable reading data from CSV files to test ground station functionality. Samples of the CSV flight data files can be found in the “docs/data” folder. Similarly, the Serial Driver Debug toggle will read data from a Ground Station Muxer (GSM) file to simulate reading data from a Serial device. This file can be generated by the “mux-gsm” application found in “utils/gsm”, and a sample file is also included in “docs/data”. The Video Mode setting will place the ground station in video mode, activating the live video features. These features will be fully explained later in this document.

Functionality Settings

The Map Cache Size setting controls the size of the tile cache (in bytes) for the map in the main window. The cache is useful for using the map while your device is not connected to the internet, which is common at launch sites, but you may want to limit the size of the cache so it does not take too much space on your computer. It is recommended to set this to at least 1 MB to ensure the cache has enough space to function properly. However, some users may wish to disable this functionality entirely, which can be done by disabling the Toggle Tile Caching option (Note: Not implemented).

The Serial port baud rate setting controls the baud rate that the application will use when trying to connect to a Serial device. It is important to ensure that this setting is the same in both the application settings and the device settings so that they can communicate properly. It is recommended to set this to one of the standard baud rate settings (9600, 115200, etc.) to ensure the connection functions properly.

Streams Settings

The ground station allows the user to configure a variable number of input and output streams to maximize flexibility so it can be used with a variety of avionics systems configurations. These streams can be configured through the use of the streams page. Use of the settings page will be covered later in this guide, while this section will explain the different configuration options.

Each stream must be configured with three values: the stream name, the stream type, and the stream ID. The name identifies the stream and must be a valid file name on the file system. While the name can be anything, there are some reserved names. Giving a stream these names will connect the stream with that name to specific ground station functionality. These reserved names are explained below.

- telem-avionics: this stream's data is displayed in the Avionics section of the main page when its type is Telemetry
- telem-airbrake: this stream's data is displayed in the Airbrake section of the main page when its type is Telemetry
- telem-payload: this stream's data is displayed in the Payload section of the main page when its type is Telemetry
- video0: this stream's data is made available as Input 0 in the video control panel when its type is Video
- video1: this stream's data is made available as Input 1 in the video control panel when its type is Video

The type of a stream specifies the type of data that will be read from this stream. The stream types come from the RadioMessage library used by the ground station Serial driver. Use of the RadioMessage library to communicate with the Serial driver is explained in the following section. The different stream types and their RadioMessage equivalents are explained below.

- Telemetry (APRSTelem): input stream of telemetry data about the state of a flight computer on the rocket
- Command (APRSCmd): output stream of commands to a flight computer
- Metrics (Metrics): input stream representing performance metrics from a specific radio
- Video (VideoData): input stream of AV1 video data

The stream ID specifies the ID for the demultiplexer to look for when reading data from that stream. This ID should be set by the connected Serial device when multiplexing the stream.

In addition to configuring settings for each stream, the stream page can also be used to configure the available commands for Command streams, and the available stateflags for Telemetry streams by selecting the “Edit Commands and Stateflags” button. A new command can be created by opening the dropdown in the command section and selecting the “New command” option. A command can be edited by selecting it from the dropdown. Each command requires setting the following values:

- Name: the name of the command (also turned into an acronym)
- Function Name: the name of the function to be called on the flight computer (allows for generation of a configuration file that can be loaded by the flight computer)

Each command can optionally take up to 16 arguments totalling up to 16 bits, where each argument requires the following settings:

- Syntax: essentially the same of the argument to help in remembering the order of arguments when sending a command
- Width: how many bits this argument should take up
- Type: the type of argument, can be “bool”, “num”, or “string”
- Conversion List: only available with the type set to “string”, allows for the use of words as arguments in the place of numbers, this is the list of available words

The available stateflags can be set using a similar process in the stateflags section. A new stateflag can be created by selecting “New stateflag” from the dropdown, and stateflags can be edited by selecting them from the dropdown. Each stateflag requires the following values to be set:

- Name: the name of the stateflag, used to identify it later
- Width: how many bits the stateflag takes up
- Function Name: the name of the function to be called to get the stateflag value on the flight computer

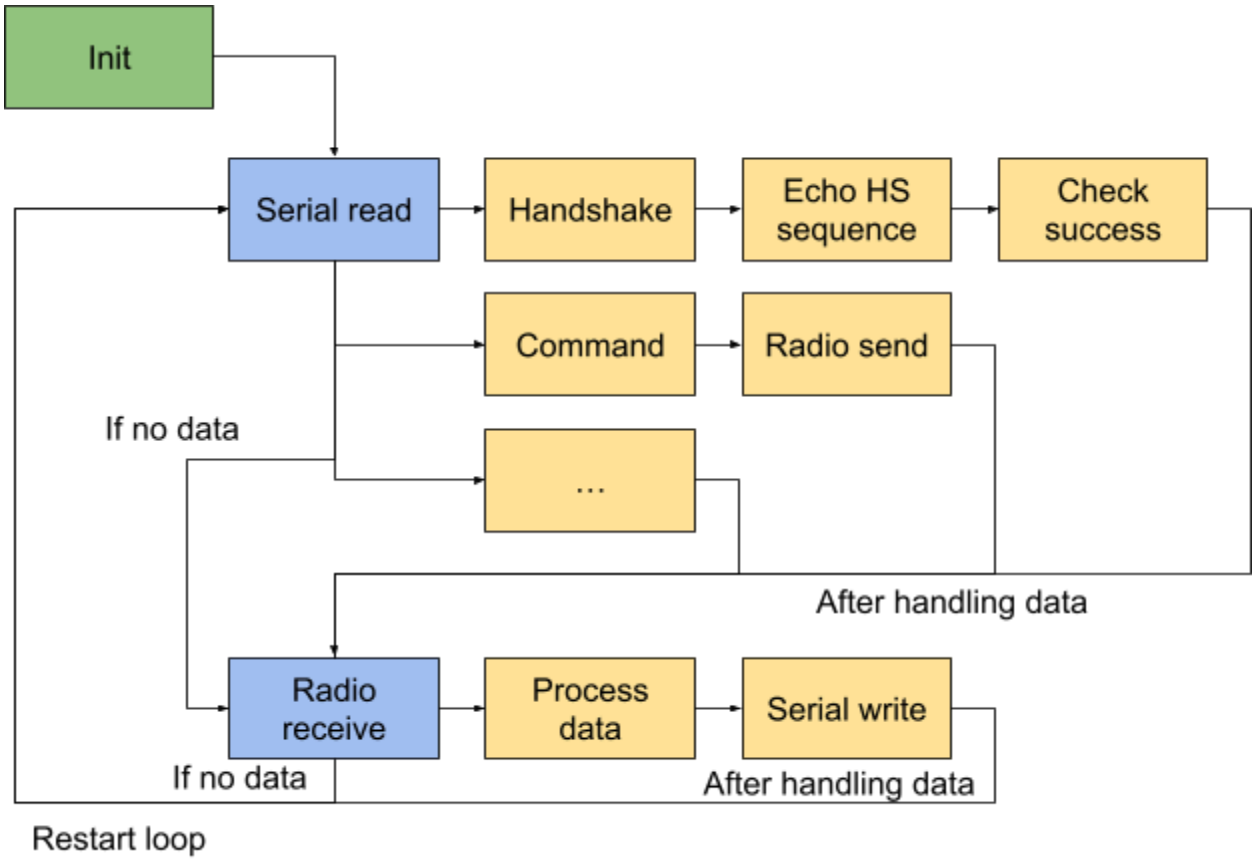
Stateflags for each Telemetry stream can be selected by clicking the “gear” icon for the stream. Note that the menu must be closed by clicking on the “gear” icon again for your selections to save. The selected stateflags for each stream and the list of available commands can be exported to a compressed format to be stored on the flight computer by clicking the “Export Config” button. The exported configs are stored as JSON files in the “data” directory.

Device Compatibility

There are no specific hardware requirements to use a device with the ground station besides the ability to connect via Serial. Rather, this section will focus on ensuring your device's software is compatible. This is important because, while it uses a standard Serial connection to communicate, the ground station expects to receive information in a specific format, or it will not function properly. Specifically, the ground station Serial driver is a demuxer that expects a

multiplexed input in the format configured on the streams page. An example project to program a Serial device to communicate with the ground station Serial driver is available in “docs/examples/serial_v2_base”. The rest of this section will focus on the generalized formatting of the multiplexed data.

Control Flow



The simplified control flow of a device connected to the ground station is shown above. After the device completes its initialization, it should enter a loop where it reads from the Serial port, and then writes any available data. Data should not be written if a successful handshake has not occurred, or the last handshake failed. To complete a handshake, the device should read the handshake command from the Serial port (“handshake\n”), after which it should expect a random sequence of numbers, followed by a newline (“\n”). It should echo this sequence, including the newline, back to the Serial driver by writing to the Serial port. Finally, it should listen for the handshake succeeded (“handshake succeeded\n”) or handshake failed (“handshake failed\n”) command. If the handshake succeeded command was received, the handshake was successful and the device can begin sending data. If the handshake failed command was received, the handshake was not successful and the device cannot start sending data.

The “command\n” command , which allows for the Serial driver to send flight computer commands to the connected device so they can be transmitted, follows a similar process. After the “command\n” command is received, the device should expect the command to be written to the Serial port in the multiplexing format. In addition to the handshake and flight computer command commands, other commands can be added desired by the user, though this would require modifications to the Serial driver.

After available data on the Serial port is handled, the device, assuming a successful handshake, should handle incoming radio data. Once it has received a full transmission, it should encode that transmission into the proper data format (e.g., APRSTelem), and then encode that into the multiplexing format. It is in this final step that the stream ID specified in the streams page must be added to the data. This step is crucial because, without this ID, the Serial driver does not know which stream this data should be sent to.

Message Formatting

As mentioned previously, the ground station uses the RadioMessage library to handle reading data from the Serial port. While a complete explanation of this library is left to another document, a brief overview will be given here to assist in the setup of devices that connect to the ground station.

The RadioMessage library is split into two primary parts Message and Data. Message objects store complete messages and provide an interface for reading, writing, encoding, and decoding them, while Data objects store data that is encoded to, or decoded from a Message, and provides functions to perform encoding and decoding for a particular type of data. These two types of objects work together to provide a straightforward abstraction for handling many different types of messages.

As an example, let’s say a radio connected to a Serial device has received a message encoded as APRSTelem. Note that this must be known beforehand, as there is no way of determining the type of message from its structure at this time. This message is made available by the radio code as a APRSTelem Data object. In order to send this data to the ground station, it must first be encoded. This can be done by passing it to the encode function of a Message. However, this data must now be encoded again into the multiplexing format. To do this, a ground station Data (GSData) object must be created and given the stream Data type (APRSTelem), the stream ID (as configured in the streams page), and the device ID (pertaining to the radio this was received from, only important if multiple radios are being used). Then the GSData object should also be given the encoded APRSTelem data in the Message itself, which can be done by using the GSData fill function. Once this process is complete, the GSData object can be passed to the encode function of the Message, and the message can be sent to the ground station from the Message object.

User Interface

The ground station user interface consists of two main windows: the main application GUI (referred to as the main window) and the debug window, as well as the video window, which is only shown in video mode. The main window also consists of two pages, which are the

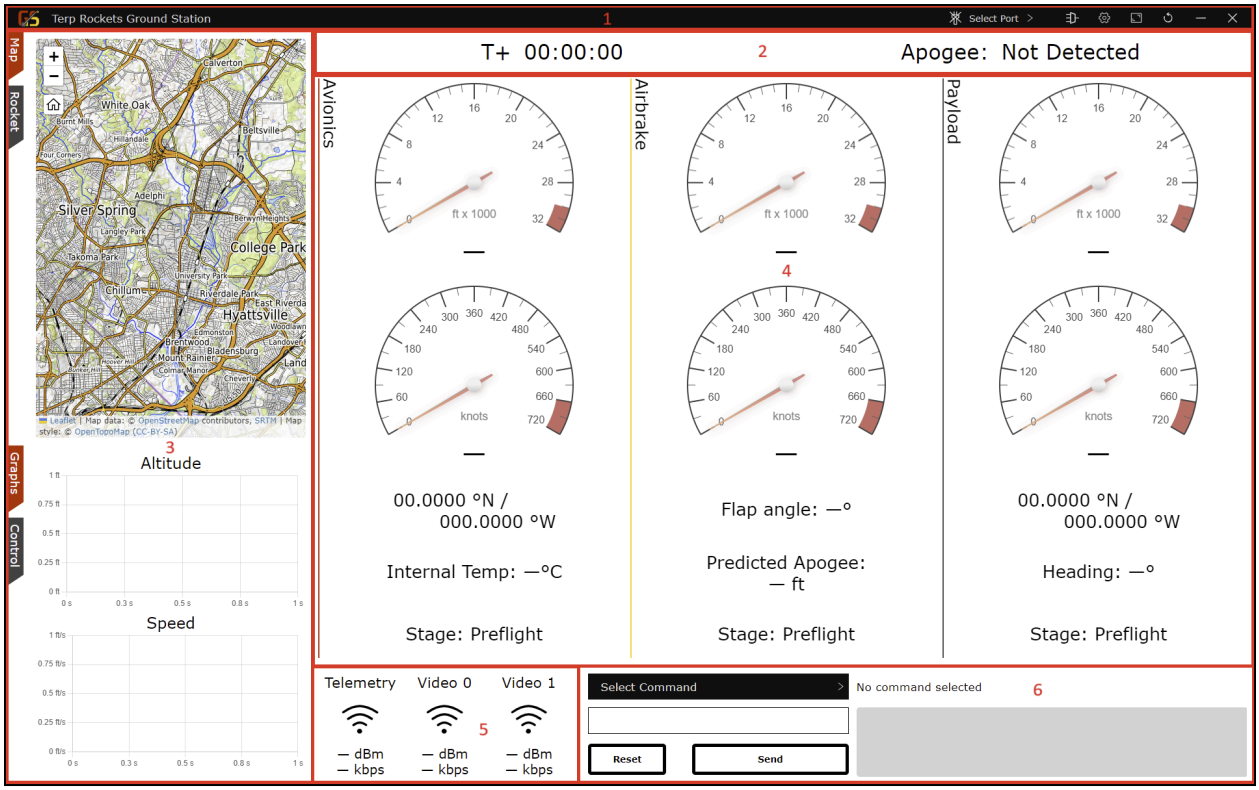
main page and the settings page. This section aims to explain the different elements of each window and how to use them.

The Main Window

The main window is the primary window you will be interacting with to control ground station functionality, change application settings, and view received telemetry. When you first launch the ground station, this should be the first window you see appear.

The Main Page

The main page of the main window is the primary page you will interact with when using the application's GUI. It contains various graphical and interactive elements that show the rocket's current state assist with connecting to the receiver device. These elements are grouped into various numbered sections and are explained below.



Note: Elements are listed from left to right.

1. The top bar:

   Purpose: Application control

   - Application logo and name
   - Serial port selector dropdown
   - Streams page button
   - Settings page button
   - Fullscreen button
   - Reload button

- Minimize button
- Close button

2. The status bar:
   Purpose: Global rocket status information
   - T+ counter
   - Apogee detection

3. The visuals section:
   Purpose: Display and select telemetry visualizations
   - Top visual sidebar (used to select the top visual)
   - Bottom visual sidebar (used to select the bottom visual)
   - Top visual
   - Bottom visual

4. The telemetry section:
   Purpose: Display live telemetry from specific rocket subsystems, with each column being a different subsystem.

Column 1: Avionics
   - Avionics altitude
   - Avionics latitude and longitude
   - Avionics speed
   - Avionics internal temperature
   - Avionics current stage

Column 2: Airbrake
   - Airbrake altitude
   - Airbrake current predicted apogee
   - Airbrake speed
   - Airbrake current stage
   - Airbrake target flap angle

Column 3: Payload
   - Payload altitude
   - Payload latitude and longitude
   - Payload speed
   - Payload ground track heading
   - Payload current stage

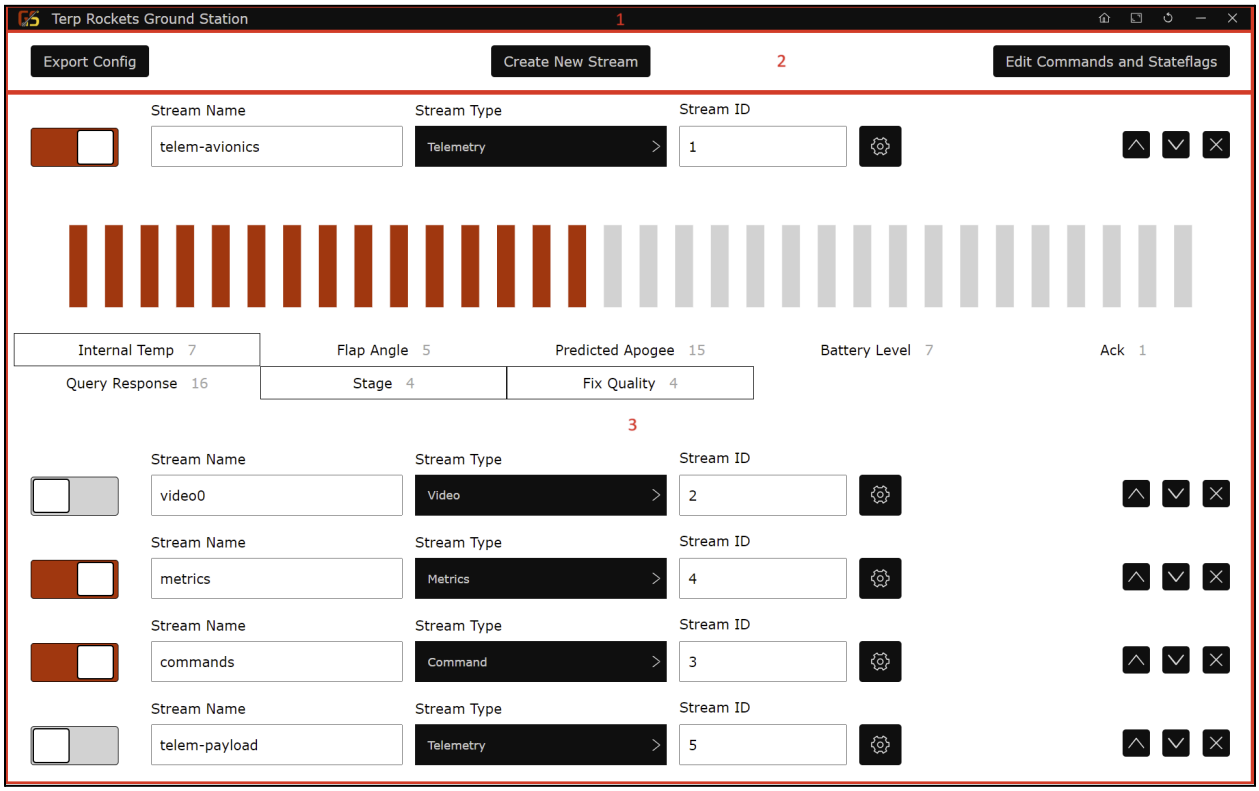
5. The radio status section:
   Purpose: Display signal status of connected radios
   - Telemetry radio status
     - Signal strength graphic
     - Current signal strength
     - Current received bitrate
   - Video 0 radio status
     - Signal strength graphic
     - Current signal strength
     - Current received bitrate
   - Video 1 radio status
     - Signal strength graphic
     - Current signal strength
     - Current received bitrate

6. The command interface:
   Purpose: Send commands to connected rocket subsystems
   - Command dropdown
   - Command entry box
   - Command syntax helper
   - Past commands log

- Reset command entry
- Send entered command

The Streams Page

The streams page allows for setting the available streams, commands, and stateflags. Changes are saved any time you exit the page. The only exception is that stateflag selections for a particular stream are only saved when closing the stateflags selection menu by clicking the “gear” icon. The Serial driver is restarted when exiting this page in order to apply the new settings. The streams page is grouped into two menus: one that allows for the configuration of streams, and one that allows for the configuration of commands and stateflags. These elements are grouped into various numbered sections and are explained below.



Note: Example streams are shown to allow all available inputs to be identified

1. The top bar:

Purpose: Application control

- Application logo and name
- Home page button
- Fullscreen button
- Reload button
- Minimize button
- Close button

2. The button bar

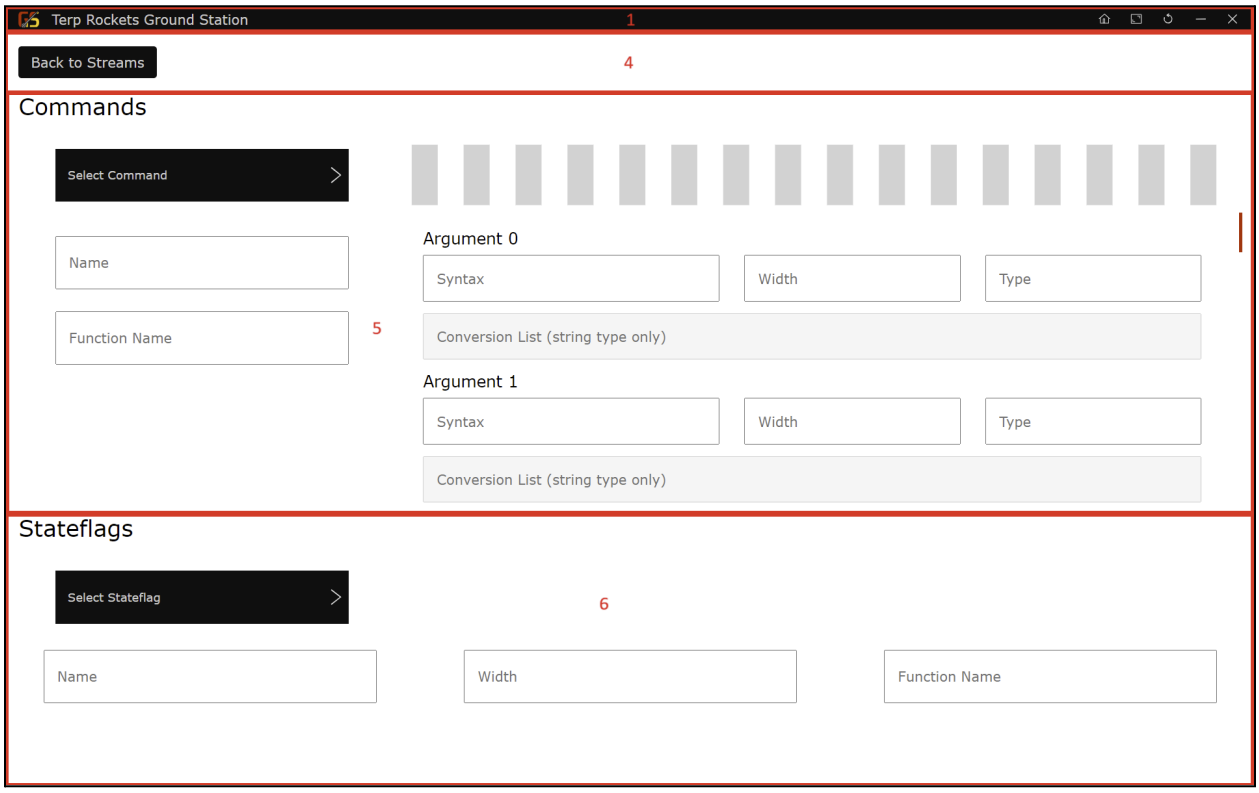
Purpose: Menu actions and controls

- Export config button

- Create new stream button
- Edit commands and stateflags button

3. The stream config section
Purpose: List current streams and allow for configuration
- The stream enable toggle
- The stream name input
- The stream type dropdown
- The stream ID input
- The extra steam configuration button
- The move stream up button
- The move stream down button
- The delete stream button

Note: for Telemetry streams only
- The stateflags bits used bar
- The stateflags selection buttons



4. The button bar (“Edit commands and stateflags” menu)
Purpose: Menu actions and controls
- The back to streams menu button

5. The commands section
Purpose: Configuration of available commands
- The command selection dropdown
- The command bits used bar
- The command name input
- The command function name input

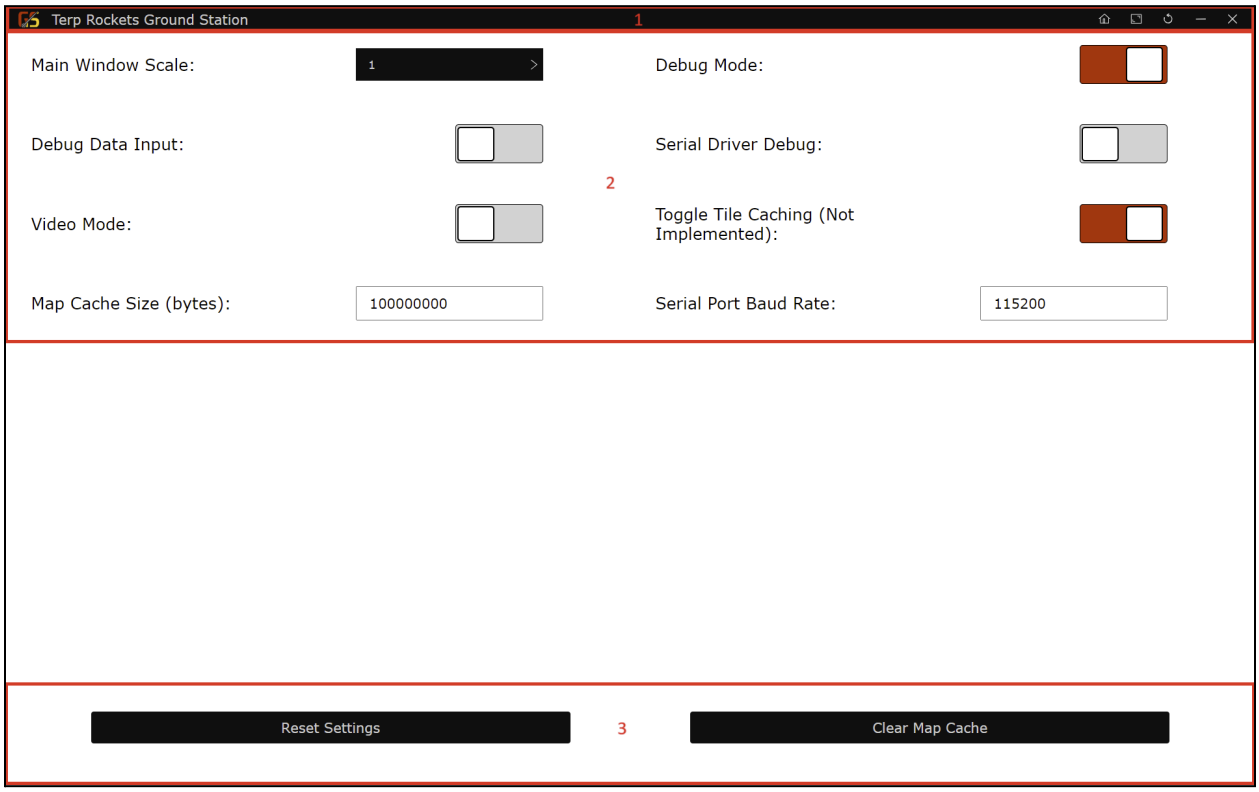
Note: the following elements are repeated for each of the 16 available arguments
- The argument identifier
- The argument syntax input

- The argument width input
- The argument type input
- The argument conversion list input

6. The stateflags section
Purpose: Configuration of available stateflags
- The stateflag selection dropdown
- The stateflag name input
- The stateflag width input
- The stateflag function name input

The Settings Page

The settings page allows for modification of any of the previously discussed application settings. Changes are saved any time you exit the settings page. While some changes may apply immediately, it is recommended you restart the application to ensure all settings take effect. These elements are grouped into various numbered sections and are explained below.



1. The top bar:
Purpose: Application control
- Application logo and name
- Home page button
- Fullscreen button
- Reload button
- Minimize button
- Close button

2. The settings section:
Purpose: Modify application settings

- Main window scale dropdown
- Toggle for debug mode
- Toggle for Debug Data Input
- Toggle for Serial driver debugging
- Toggle for video mode
- Toggle for tile caching
- Input for map cache size
- Input for Serial port baud rate

3. The controls section:
   Purpose: Settings related application controls
   - Reset settings to defaults
   - Delete the map cache

Video Mode

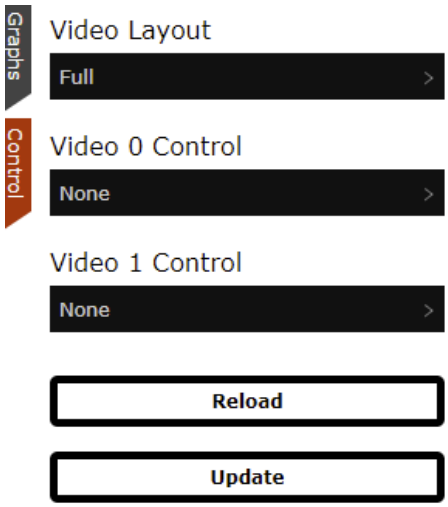
Activating video mode shows the video window on startup. It is recommended to only turn on video mode if the live video features are being used. This section will explain the layout of the video window, and how the video window can be controlled from the main page.

The Main Page

In video mode, the main window can be used to control what is displayed in the video window. These controls can be accessed by selecting the “Control” visualization from the bottom visual sidebar, which will show the video control panel.

There are a few different options for each section of the controls for the video window, available in the “Video Layout” dropdown. For controlling the window layout, you can choose a full, partial, or telemetry only layout. All layouts have a telemetry display on the left side, and a stage progress bar at the bottom. The full layout contains both video outputs, while the partial layout contains only the first. The telemetry only layout is a special case of the partial layout, where the first video is set to the charts option.

The two video outputs each contain the same options for their source in the “Video X Control” dropdowns, and they cannot be set to the same source at the same time, unless it is the “None” source, which shows a placeholder image. The “Input 0” and “Input 1” sources correspond to the first and second live video streams from the multiplexer, and the “Charts” source shows graphs of altitude and speed similar to the “Charts” visualization in the main page. At the bottom of this section, the “Reload” button allows the user to reload the video window from the main window, and the “Update” button applies all changes selected in the video control panel. The default state of the video window control panel is shown below.



The Video Window

The video window has two possible physical layouts: the full layout and the partial layout. The video window is unique, since it is constructed specifically for a 1920x1080 resolution, and will not display properly when the window does not meet or exceed this resolution. Usually the video window is operated in fullscreen mode to meet this requirement. Since the top bar of the video window disappears in fullscreen mode, it will be explained separately, and will be followed by an explanation of the default layout of the video window in fullscreen mode.

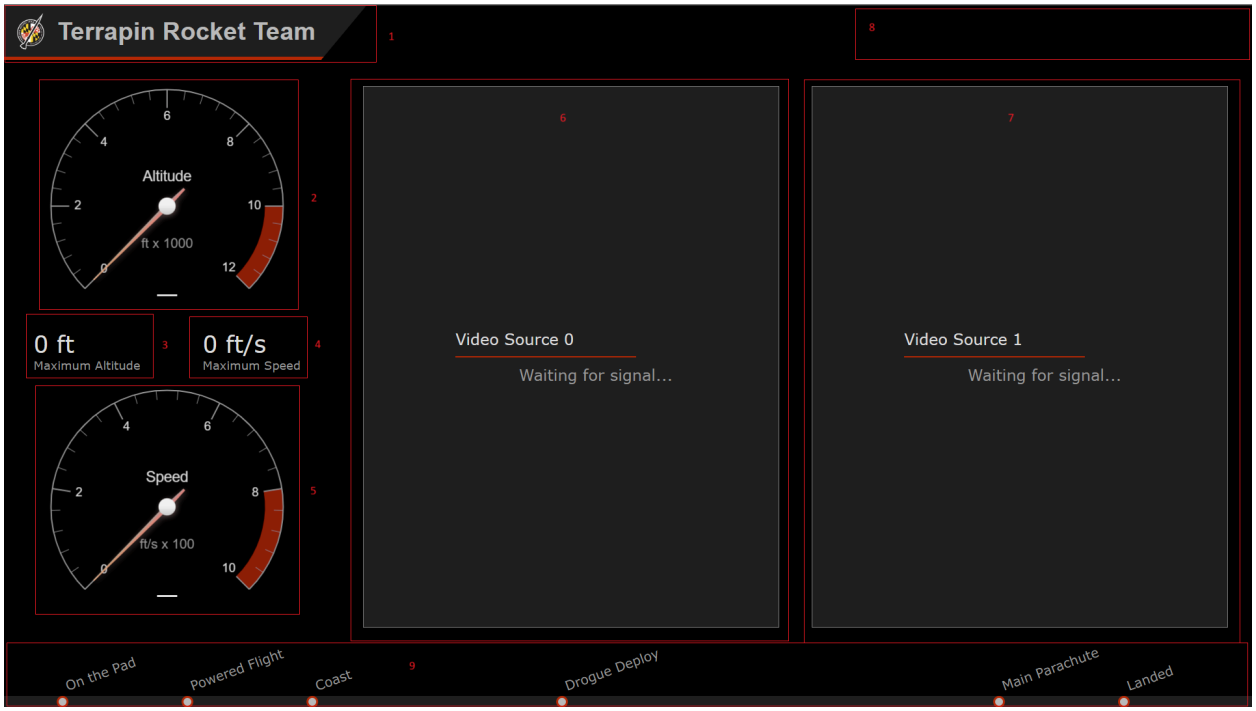
The Video Window top bar:

Note: In order from left to right

- Window title and logo
- Fullscreen the window
- Reload the window
- Minimize the window
- Close the window

Note: The video window can also be fullscreened by pressing the F11 key.

The Video Window default layout:



1. Team name banner
2. Current altitude display
3. Maximum altitude (apogee) display
4. Maximum speed display
5. Current speed display
6. Video 0 output
7. Video 1 output
8. Fun facts banner (not visible)
9. Current stage display

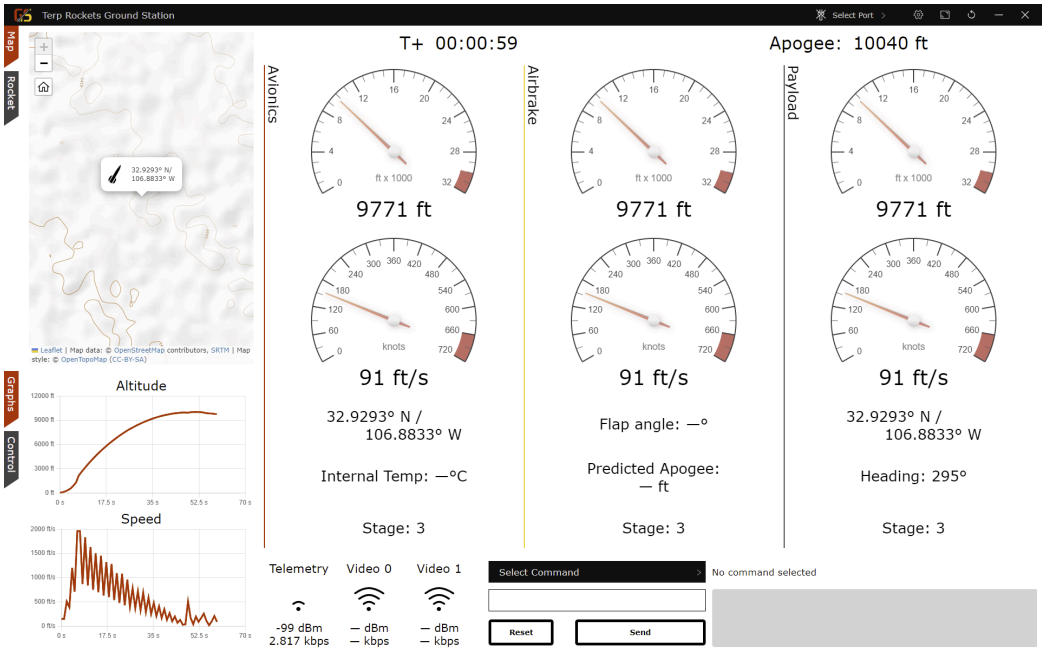
Standard Workflows

This section aims to serve as a guide on the standard workflows that would be used to operate the ground station. It will cover testing the ground station, the workflow for the main window from initial connection to the receiving device up to the rocket's landing, as well as changes in operation of the main window in video mode.

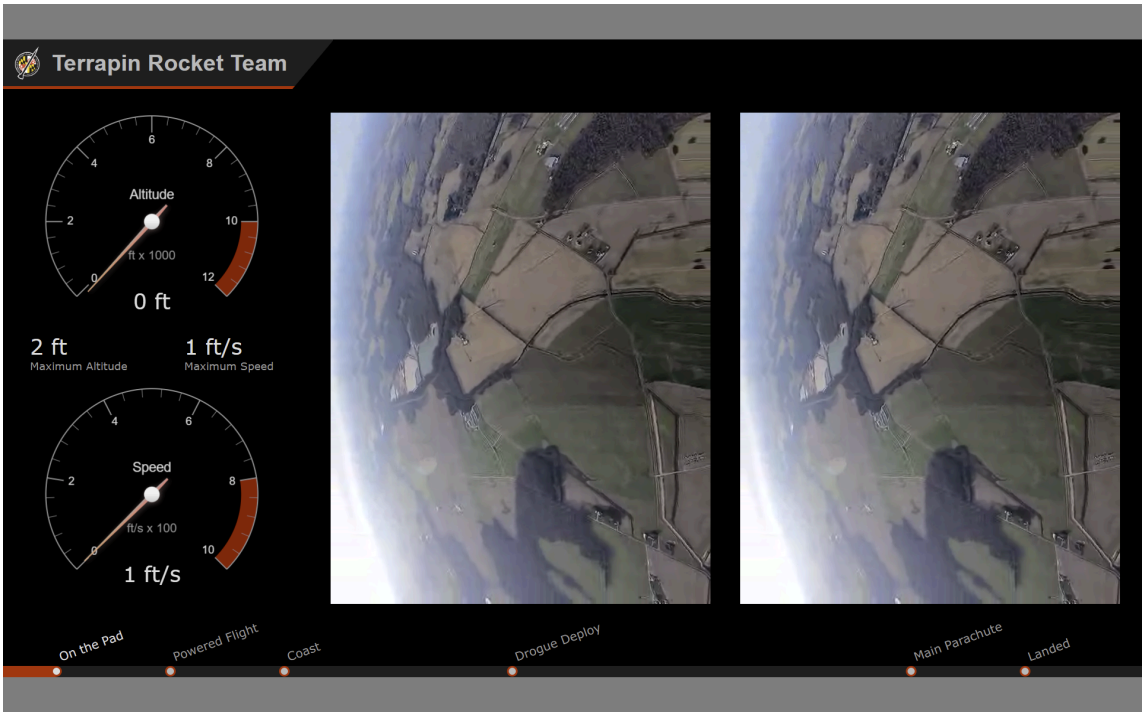
Testing

Often it is useful to be able to test that the ground station is functioning properly, without having to connect actual hardware to it. For this reason, the ground station supports reading flight data from CSV files stored locally on your machine in order to test functionality when in debug mode. Example flight data that can be used to test the frontend and backend systems of the ground station can be found in the “docs/data” directory.

To use this data to test the ground station, copy the CSV and AV1 files to the same directory as the ground station executable. Then, open the ground station by launching the executable, and using the settings page, place the ground station into debug mode and video mode, and toggle on Debug Data Input. Close the ground station and reopen it. You should start receiving data in each of the Avionics, Airbrake, and Payload telemetry displays. The current signal strength and bitrate of the Telemetry radio should also be showing. To test video functionality. Open the video control panel by clicking “Control” in the bottom visual sidebar. Then, for “Video 0 Control” select “Input 0”, and for “Video 1 Control” select “Input 1”. Press the “Update” button, and video should begin being displayed in the video window. Remember that for the video window to be properly formatted, it needs to be fullscreened on at least a 1920x1080 display. An example of the ground station main window during testing is shown below.



An example of the ground station video window during testing is shown below. Note that with the provided testing data, the video and the telemetry do not line up, which is the expected behavior since the test data was taken from two different flights.



Testing the Serial Driver

The Serial driver can also be tested using a similar process as above. The main differences being that only the GSM file needs to be copied to the same directory as the ground station executable, and the ground station should be set to Serial Driver Debug, not Debug Data Input. In order to start reading data, open the Serial dropdown and select the “Being driver debug” option. After this option is clicked, data should begin to appear on the main page. Similarly, video functionality can be tested through the same process as above.

The Main Window

Initial Setup

Before connecting the receiving device to your computer, launch the ground station. Expand the dropdown menu and note any currently available Serial connections. This is not required, but will make it easier to identify which Serial connection is your receiver. Close the dropdown and connect your device, ensuring it is powered on. Reopen the dropdown menu and select the new Serial connection. Check the Serial connection indicator to ensure your device is connected. If it is not connected, you can open the debug log (“debug.log”) or the serial driver debug log (“logs/serial_driver_debug.log”) to check for errors. Sometimes closing the application and unplugging the device, then reopening the application and reconnecting the device can resolve connection issues.

Once your device is connected, turn on your avionics/transmitter, and you should start receiving data. This can be checked by confirming data is appearing on the main page and that the radio status section corresponding to the active radio is indicating the current signal strength

and bitrate. Check the signal strength and signal strength indicator to ensure you have a strong signal. Check all the data readouts, making sure they are displaying the expected values. You now have successfully connected to the ground station, and telemetry is being logged.

Before and During Launch

While connected to the receiving device, it is important to ensure your device does not enter sleep mode, as this can silently disrupt the Serial connection. If your computer goes to sleep and you stop receiving communications, you may need to restart the ground station and reconnect the receiving device to your computer. Monitor the ground station during the launch, making sure it is receiving all expected transmissions. If supported by your hardware, you can send commands to the flight computer(s) on the rocket by using the command interface. The ground station will try to detect apogee, but it is best to monitor received data closely during flight.

After Launch

Once the rocket lands, it is likely that you will lose signal due to losing line of sight. However, the ground station should still display the last known latitude and longitude of the rocket, and will attempt to display its location in the map view. This can be useful in determining the location of the rocket for recovery. Once you no longer need to receive transmissions, you can disconnect the receiving device from the ground station and close the application. A log of the data received during the flight is made available in the data folder in the same directory as the application. The file is in CSV format and the name is the name of the telemetry stream (avionics, airbrake, or payload) followed by the ISO date for when the first transmission was received.

Video Mode

Initial Setup

Before connecting the receiving device to your computer, launch the ground station. Expand the dropdown menu and note any currently available Serial connections. This is not required, but will make it easier to identify which Serial connection is your receiver. Close the dropdown and connect your device, ensuring it is powered on. Reopen the dropdown menu and select the new Serial connection. Check the Serial connection indicator to ensure your device is connected. If it is not connected, you can open the debug log or the serial driver log to check for errors. Sometimes closing the application and unplugging the device, then reopening the application and reconnecting the device can resolve connection issues.

Once your device is connected, turn on your avionics/transmitter, and you should start receiving data. This can be checked by confirming data is appearing on the main page and that the radio status section corresponding to the active radio is indicating the current signal strength and bitrate. Check the signal strength and signal strength indicator to ensure you have a strong signal. Check all the data readouts, making sure they are displaying the expected values. You now have successfully connected to the ground station, and telemetry is being logged.

Next, set up the video window for the flight. Usually this window is fullscreened on a separate monitor, so that the main window can still be used to control it. Set the layout and sources for each video output based on the capabilities of your system. For example, if you have a single video stream and a telemetry stream you could use the full layout with Video 0 source being the video stream and Video 1 source being the charts.

Before and During Launch

While connected to the receiving device, it is important to ensure your device does not enter sleep mode, as this can silently disrupt the Serial connection. If your computer goes to sleep and you stop receiving communications, you may need to restart the ground station and reconnect the receiving device to your computer.

Often video transmitters will initially be powered off to preserve battery before flight, and are only powered on immediately before flight. If supported by your hardware, you can use the command interface to manually power on your video transmission equipment and start transmitting video.

Monitor the ground station during the launch, making sure it is receiving all expected transmissions. The ground station will try to detect apogee, but it is best to monitor received data closely during this phase of flight. The status of the live video streams should also be closely monitored throughout the flight. If you lose signal from one of the video streams, it can be switched off of the video window either by changing the layout or changing the source for the affected video output.

After Launch

Once the rocket lands, it is likely that you will lose signal due to losing line of sight. However, the ground station should still display the last known latitude and longitude of the rocket, and will attempt to display its location in the map view. This can be useful in determining the location of the rocket for recovery. Once you no longer need to receive transmissions, you can disconnect the receiving device from the ground station and close the application. A log of the data received during the flight is made available in the data folder in the same directory as the application. The file is in CSV format and the name is the name of the telemetry stream (avionics, airbrake, or payload) followed by the ISO date for when the first transmission was received. The video received during the flight will also be logged encoded as AV1 data in the same folder using the same naming convention.

Congratulations on reaching the end of the user guide! You should now be able to use the full capabilities of the Terp Rockets Ground Station!