Terp Rockets Ground Station User Guide

Joseph Hauerstein

August 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 competition rocket, primarily logging and displaying telemetry received via radio transmission. It uses specially formatted APRS messages for receiving data, and will also log APRS specific data, such as callsign, tocall, path, etc. though this is not required. 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 receive telemetry 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 telemetry stream and two live video streams over a single USB cable.

Installation

There are two supported ways to install the ground station: from a precompiled binary, or from source. New releases are infrequent and may not contain a binary for your operating system. While both methods will be explained, it is generally preferred to install a precompiled binary 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 from 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 “Terp Rockets Ground Station.exe”.

From Source

To install the ground station from source, clone the Git repository or download the source code from the Github page. Currently the build process has only been tested to work on Windows, but it should be possible to build on Linux without too much trouble. Before beginning the build process, make sure to install the latest versions of Node.js and NPM.

Windows

Before building on Windows, Mingw-w64 must be installed, which can be found here. 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
```

Once all required software is installed, open a terminal in the same directory as the source code. (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 C:\path\to\mingw help
```

To start the full build, run the command:

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

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

While it has not been thoroughly tested, the build system should be compatible with Linux. As with Windows, the following dependencies will need to be installed via your distro’s package manager:

- GCC, G++
- 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.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 same directory (“build/src”) as on Windows, though it will be named “Terp Rockets Ground Station” (without the “.exe” extension).

Note: The Serial driver is currently only designed to work with Windows.

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

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

Most of the build process does not support multicore builds. However, the ffmpeg build time can be greatly shortened by using multiple cores. To set the number of cores to use, open the “video/build.sh” file, and edit the `CORES` variable to be the number of cores to use in the build.

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, through the debug window, or directly modified in the config.json file. Use of the settings page and the debug window will be covered later in this guide, while this section will explain the different configuration options. Each option is listed with its name in the settings page, followed by the name in the debug window and config.json file in parenthesis.

- Main Window Scale (scale)

- Debug Window Scale (debugScale)
- Debug Mode (debug)
- Disable GUI (noGUI)
- Video Mode (video)
- Map Cache Size (cacheMaxSize)
- Serial Port Baud Rate (baudRate)

Scale Settings

The Main Window Scale and the Debug Window Scale both control the size of each window, since these windows 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, Disable GUI, 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, and reading data from a “test.csv” file in the root application directory to simulate receiving flight data. The “test.csv” file is in the same format as the data output from the ground station.

The Disable GUI setting will make it so that the debug window launches when the application is opened, rather than the main window. This is useful if there is an issue with the GUI because the ground station can still be fully controlled from the debug window. Both windows, main and debug, can also launch the other window, but when the original window is closed, both windows will be. 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.

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.

Device Compatibility

There are no specific hardware requirements, besides the ability to connect via Serial, to use a device with the ground station. 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 three stream multiplexed input, where the first stream contains telemetry, and the second and third contain video. C++ style pseudocode that illustrates the exact formatting for the muxer is available in “docs/examples/groundTransceiver.cpp”. The rest of this section will focus on the formatting for the telemetry and live video streams.

Telemetry

The expected format for Serial communication is shown below.

```
s\r\n

Source:xxx,Destination:xxx,Path:xxx,Type:xxx,Data:xxx,RSSI:xxx\r\n

e\r\n
```

The message begins and ends with a specific sequence of characters to indicate the beginning (using "s\r\n") and end (using "e\r\n") of a message, and indicates the end of the message content with another sequence "\r\n". Note that "\r" and "\n" represent the carriage return and new line escape sequences respectively, but are shown here in text form for clarity.

The message content also has a specific format, with different fields where the "xxx" represents the data. These fields are explained below.

- Source: The source APRS callsign
- Destination: The APRS tocall
- Path: The APRS path setting
- Type: The type of APRS message
- Data: The telemetry recieved from the rocket
- RSSI: The signal strength of the recieved transmission

The Source, Destination, Path, and Type fields are APRS specific values that will not be explained in this guide, and can be set to an arbitrary string (eg. "NA") if APRS is not used. The RSSI setting must be a number. Values greater than -60 represent good signal strength, those between -60 and -90 represent fair signal strength, those between -90 and -120 represent poor signal strength, and values less than -120 repesent extremely low signal strength or loss of signal. Set to 0 if your device does not provide a signal strength. The Data field has a specific format, which is shown below.

```
!DDMM.hhd/DDDMM.hhd[hhh/sss/A=DDDDDD/S[s]/zzz/yyy/xxx/fff
```

The Data field contains latitude, longitude, heading, speed, altitude, orientation, and state flags, the format of this data is explained below.

- Latitude (`DDMM.hhd`): DD is degrees, MM.hh is minutes, and d is North (N) or South (S)
- Longitude (`DDDMM.hhd`): DDD is degrees, MM.hh is minutes, and d is East (E) or West (W)
- Heading (`hhh`): heading as an azimuth (0 to 359 degrees)
- Speed (`sss`): current speed in ft/s
- Altitude (`aaaaaa`): current altitude in ft (-aaaaaa if negative)
- Z orientation (`zzz`): current z angle in degrees (0 to 359 degrees)
- Y orientation (`yyy`): current y angle in degrees (0 to 359 degrees)
- X orientation (`xxx`): current x angle in degrees (0 to 359 degrees)

You can also send other information in the Data field, but it will not be shown in the GUI and will only be logged in the debug window and stored in the log file. To assist in implementing this communication format, sample code is located in `docs/examples/`. It is important that data be sent following the format listed above, or it may be difficult for the ground station to parse.

Commands

The Ground Station can also be used to command the connected radio to send commands to the rocket. While further explanation for the meaning of each flag is left to the Avionics repository, the formatting is listed here so that users who wish to use this feature with the Ground Station can properly develop their radio firmware. The formatting of the commands sent from the ground station is shown below.

```
PDVF
```

This string contains the data entered into each of the four fields in the command popup window. For more information, see the User Interface section. Each letter in the string represents a single byte (number from 0-255), where the value is the value entered into the text box, and 255 represents no input. The meaning of each byte is explained below.

- `P` (0-254 minutes): Minutes until RPI power on
- `D` (0-254 minutes): Minutes until data recording start
- `V` (0-254 minutes): Minutes until video start
- `F` (0-1, True/False): Force the flight computer to detect launch now

Live Video

Each live video stream is expected to be encoded using the AV1 encoding format, so the ffmpeg installation included with the ground station is compiled with support for the dav1d AV1 decoder, which allows for efficient, real-time decoding of the video streams. However, any video encoding format that is supported by ffmpeg out of the box (i.e., not requiring additional binaries to enable, only the ffmpeg source) should be supported. Though it should be noted that the ground station was only tested with decoding AV1 video, so there are no guarantees that it will properly display video encoded in another format in real time. Additionally, the video window will need to be reloaded in the event of signal loss from one of the video transmitters. This will be improved in a future release as live video features are still in active development.

User Interface

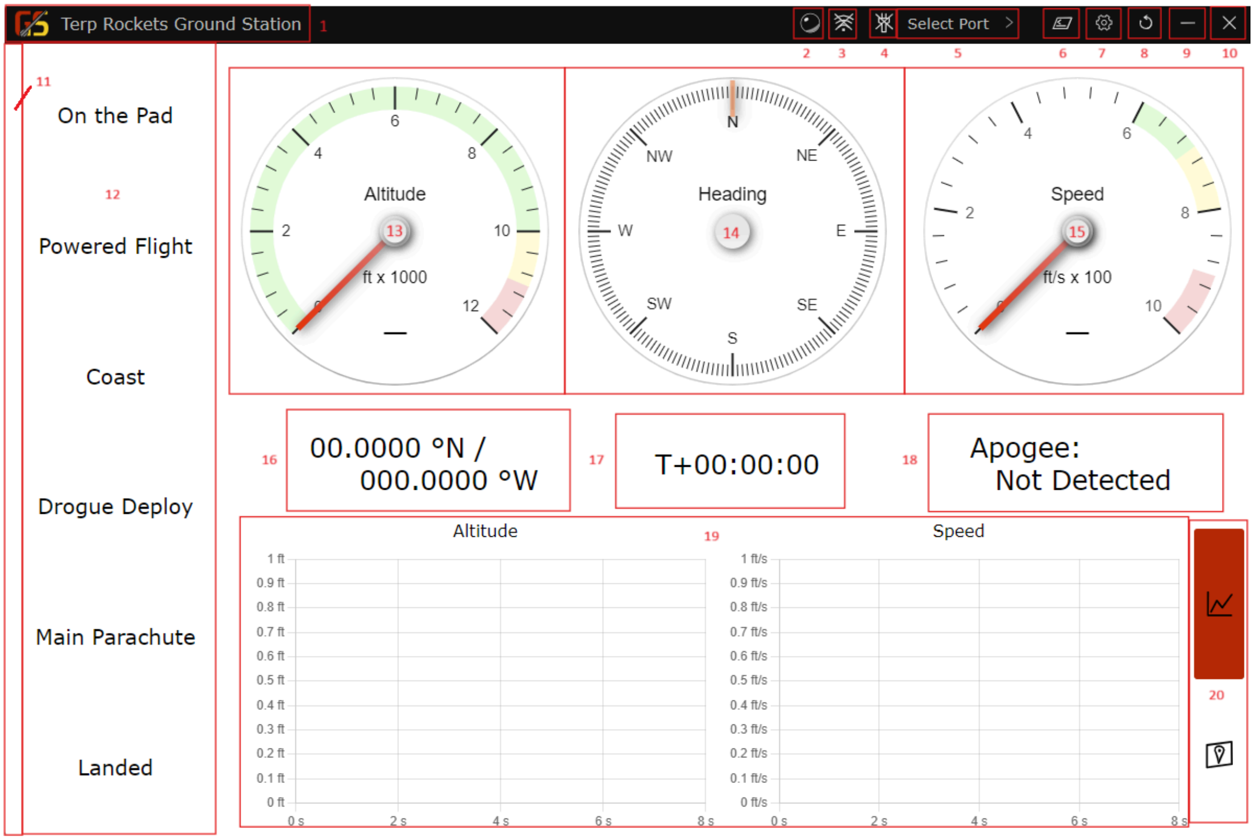
The ground station user interface consists of two main windows: the main application GUI (referred to as the main window) and the debug window, as well as the video window, which is only shown in video mode. The main window also consists of two pages, which are the main page and the settings page. This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

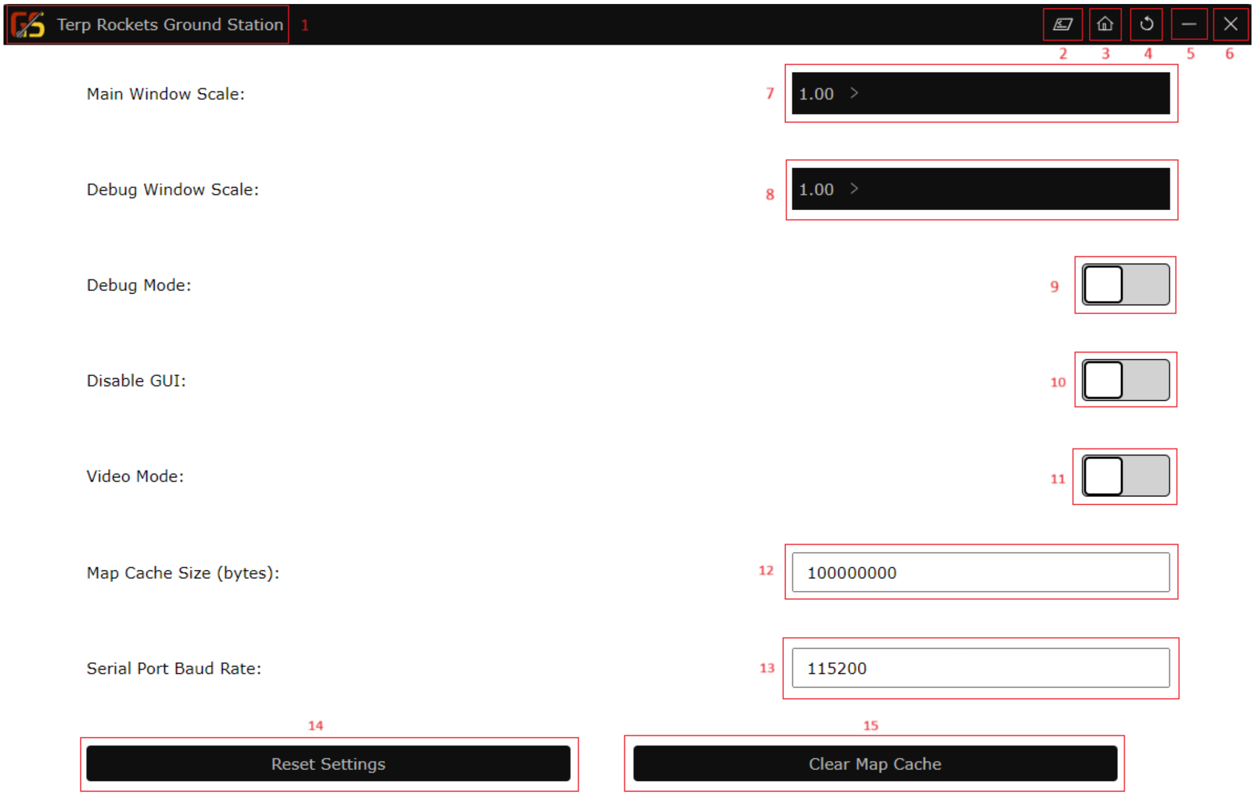
The main page of the main window is the primary page you will interact with when using the application's GUI. It contains various graphical and interactive elements that show the rocket's current state assist with connecting to the receiver device. These elements are highlighted and explained below.



1. The application name and logo
2. The transmission indicator
3. The signal strength indicator
4. The Serial connection indicator
5. The Serial port selector
6. Open the debug window
7. Open the settings page
8. Reload the GUI
9. Minimize the GUI
10. Close the GUI
11. Stage progress bar (not visible)
12. Current stage name
13. Altitude indicator
14. Heading indicator
15. Speed indicator
16. Current latitude and longitude
17. Time since T-0
18. Detected Apogee
19. Map/graph panel
20. Map/graph view switcher

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. The elements on the settings page are highlighted and explained below



Main Window Scale: 1.00

Debug Window Scale: 1.00

Debug Mode:

Disable GUI:

Video Mode:

Map Cache Size (bytes): 100000000

Serial Port Baud Rate: 115200

1. The application name and logo
2. Go back to the main window
3. Open the settings page
4. Reload the GUI
5. Minimize the GUI
6. Close the GUI
7. Main window scale dropdown
8. Debug window scale dropdown
9. Toggle for debug mode
10. Toggle for disabling the GUI
11. Toggle for video mode
12. Input field for the map cache size
13. Input field for the Serial port baud rate
14. Reset settings to defaults
15. Delete the map cache

The Debug Window

As opposed to the main window, the debug window relies on a text based user interface similar to a command window. This interface is set up with a series of base commands, each of which include a set of subcommands that interact with different aspects of the base command. To begin, you can view the full list of base commands by running the command:

```
help
```

As can be seen there are three base commands, which are `window`, `settings`, and `serial`. The window base command allows you to control the debug window and open other windows. The full list of subcommands can be seen by running `window`, and are explained in the list below.

- `-reload`: reloads the UI similar to the reload button in the main window
- `-clear`: clears all command messages from the debug window
- `-devtools`: opens the chromium devtools
- `-opengui`: opens the main window

The settings base command allows you to both view and modify different configuration options from the debug window. It is essentially equivalent to the settings page in the main window. The full list of subcommands can be seen by running `settings`, and are explained in the list below.

- `-set`: allows you to change settings, the subcommands are listed below, text between and including <> should be replaced with the value of the new setting
  - `-scale <new-scale>`: changes the scale setting to <new-scale>
  - `-debugScale <new-scale>`: changes the debugScale setting to <new-scale>
  - `-debug <new-debug>`: changes the debug setting to <new-debug>, should be "true" or "false"
  - `-noGUI <new-noGUI>`: changes the noGUI setting to <new-noGUI>, should be "true" or "false"
  - `-maxCacheSize <new-maxCacheSize>`: changes the maxCacheSize setting to <new-maxCacheSize> in bytes
  - `-baudRate <new-baudRate>`: changes the baudRate setting to <new-baudRate>
- `-save`: saves the current configuration of the settings. Note this does not occur automatically as in the main window
- `-scale`: displays the current scale
- `-debugScale`: displays the current debugScale
- `-debug`: displays the current debug setting
- `-noGUI`: displays the current noGUI setting
- `-maxCacheSize`: displays the current maxCacheSize
- `-baudRate`: displays the current baudRate

The serial base command allows you to manage the application's connection to your receiver device. It serves similar functionality to the Serial port selector in the main window. The available subcommands can be seen using `serial`, and are explained below.

- `-connect <port>`: connects via Serial to the specified <port> using the baud rate selected in settings
- `-disconnect`: disconnects the Serial connection if it is active
- `-status`: displays the current status of the Serial connection
- `-list`: lists the available Serial connections

Note: Control of the live video window is not currently implemented in the debug window.

Video Mode

Activating video mode shows the video window on startup, and makes some modifications to the main page. For this reason, it is recommended to only turn on video mode if the live video features are being used. This section will explain both the modifications to the main page, as well as the layout of the video window.

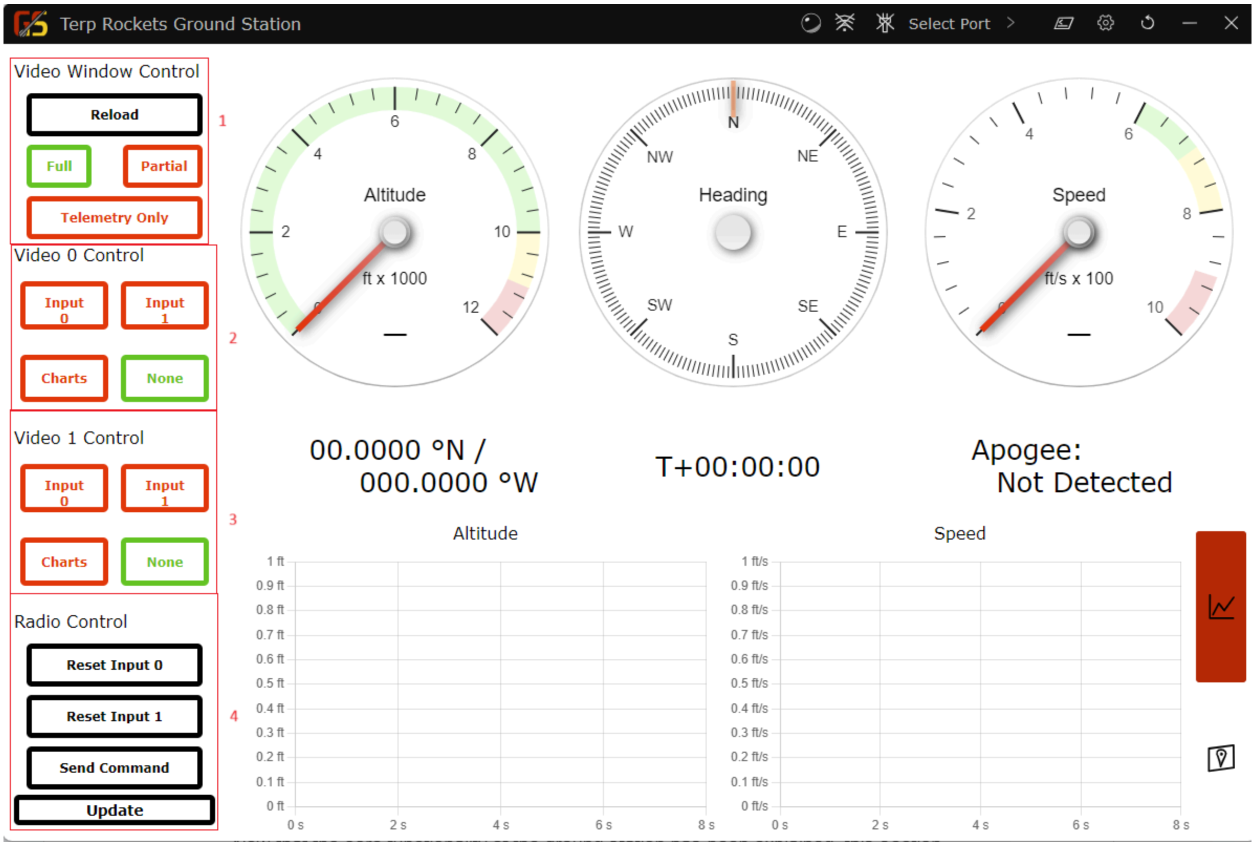
The Main Page

In video mode, most of the main page layout stays the same. However, the panel containing the stage progress bar and the current stage name is replaced with controls for the live video window. The layout of these controls is shown at the end of this section.

There are a few different options for each section of the controls for the video window. 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 “Reload” button in this section allows the user to reload the video window from the main window, but the proper way of updating the video window is by pressing the “Update” button in the radio interface section, which applies all changes selected in the video control panel.

The two video outputs each contain the same options for their source, 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 respectively, and the “Charts” source shows graphs of altitude and speed similar to those in the main page.

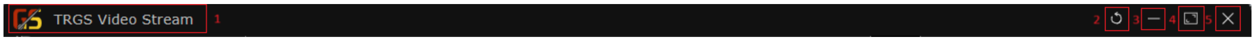
Finally, the radio interface controls provide functionality to send commands using the connected radio. Currently the “Reset Input 0” and “Reset Input 1” buttons are not implemented. The send command button shows a popup window that allows for setting flags used by the Avionics flight computer. An explanation of these flags should be found in the Avionics repository.



1. Video window layout controls
2. First video output source control
3. Second video output source control
4. Radio interface controls

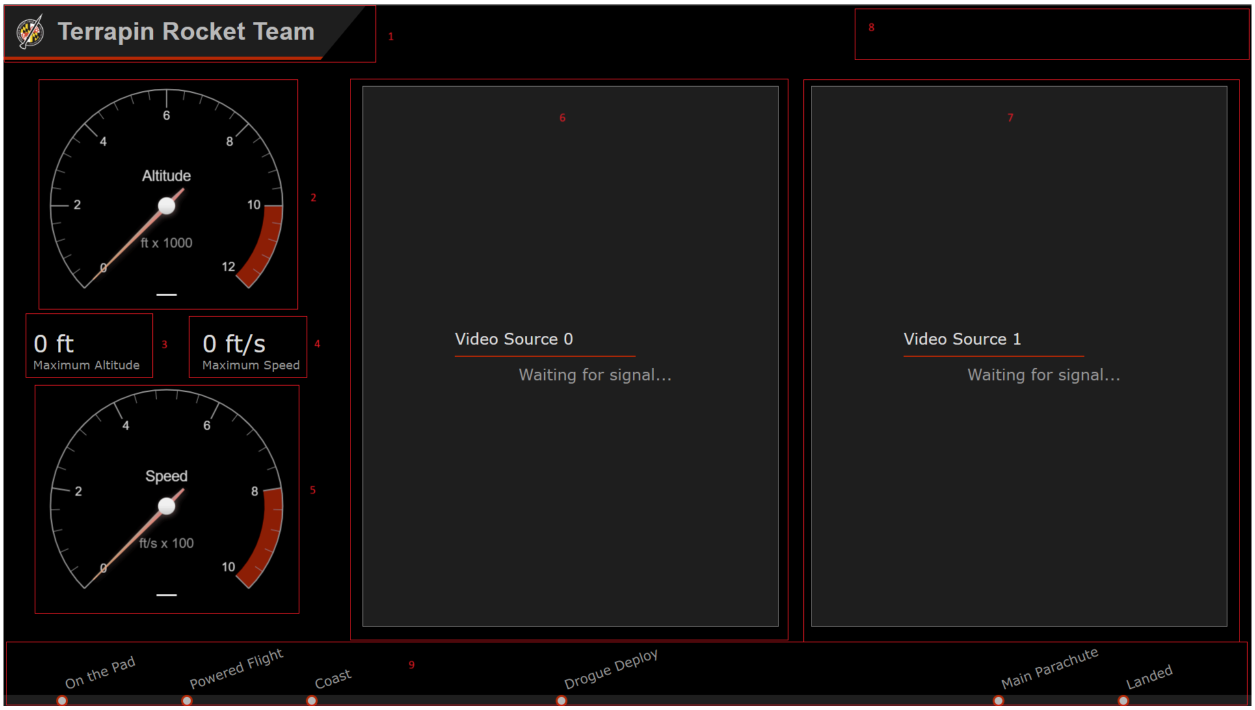
The Video Window

The video window has two possible physical layouts: the full layout and the partial layout. Images of both layouts will be shown, though only the full layout will be annotated with the positions of important elements, since the partial layout is very similar to the full 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 layout of the video window in fullscreen mode.



1. Window title and logo
2. Reload the window
3. Minimize the window
4. Fullscreen the window
5. Close the window

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



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 Workflow

This section aims to serve as a guide on the standard workflows that would be used to operate the ground station during a launch. It will cover the workflow for both the main and debug windows 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.

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, open the debug window and 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 is indicated by the transmission indicator flashing green. Also check the signal strength indicator to ensure you are receiving 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 will likely 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 set the time until data recording starts by using the "Minutes until data recording" command option, or the "Force launch now?" option can be used to force launch detection for the transmitter. 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 ISO date for when the first transmission was received.

The Debug Window

Initial Setup

Before connecting the receiving device to your computer, launch the ground station and run the following command:

```
serial -list
```

This is not required, but will make it easier to identify which Serial connection is your receiver. Connect your device and ensure it is powered on. Rerun the `-list` command and identify the name of the port that corresponds to your device. Then, run the following command to connect to the device:

```
serial -connect <port>
```

To make sure the connection has succeeded, you can run this command:

```
serial -status
```

If it is not connected, look for error messages under the `serial -connect <port>` command. 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. A message will appear in the debug window for each message received. It will contain the received telemetry, along with the received signal strength. Check these values, 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 will likely 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. The ground station does not check for apogee if the main window is not open, so this will have to be done manually.

After Launch

Once the rocket lands, it is likely that you will lose signal due to losing line of sight. However, the last transmission received from the rocket will still be displayed, so you can locate the rocket using its longitude and latitude. 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 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, open the debug window and 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 is indicated by the transmission indicator flashing green. Also check the signal strength indicator to ensure you are receiving 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 have 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 will likely 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, the “Minutes until RPI power on” and “Minutes until video start” options in the command window can assist you in starting the video stream at the right time. If the video stream starts based on launch, and launch is not properly detected, then the “Force launch now?” option can force the hardware to detect launch.

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 ISO date for when the first transmission was received.

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