Terp Rockets Ground Station User Guide

July 2023

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 including logging and displaying telemetry received via radio transmission. It supports the use of APRS messages for transmitting telemetry, and will log APRS specific data, such as callsign, tocall, path, etc. but 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 recieve telemetry from many different types of rockets.

Installation

There are two supported ways to install the ground station: from a zip file generated for a new release, or from source. New releases are infrequent and many not contain a binary for your operating system. While both methods will be explained, it is generally preferred to install from source.

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.

From Source

To install the ground station from source, clone the Git repository or download the source code from the Github page. Before beginning the build process, make sure to install the latest versions of Node.js and NPM. Once these are 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) and run the command:

```
npm install
```

This will install the depencies required to build the ground station.

If you would like to generate an installer along with the executable, open the makers.txt file in the root directory of the source code and copy the maker for your platform into the config.makers section of the package.json file. This step is not required to generate the executable required to run the ground station on your machine, and is only necessary if you want a portable installer. By default, building the ground station will generate a folder with the executable and a zip file.

```
  "author": "Joseph Hauerstein",
  "license": "ISC",
  "dependencies": {
    "electron-squirrel-startup": "^1.0.0",
    "serialport": "^10.5.0"
  },
  "config": {
    "forge": {
      "packagerConfig": {
        "icon": "assets/icon"
      },
      "makers": [
        {
          "name": "@electron-forge/maker-zip"
        }
      ]
    }
  },
  "devDependencies": {
    "@electron-forge/cli": "^6.0.4",
    "@electron-forge/maker-deb": "^6.0.4",
    "@electron-forge/maker-rpm": "^6.0.4",
    "@electron-forge/maker-squirrel": "^6.0.4",
    "@electron-forge/maker-zip": "^6.0.4",
    "electron": "^22.1.0"
  }
}
```

To build the application, return to the previously used terminal window and run the command:

```
npm run make
```

Once this command is complete, navigate to the "out" directory in the root folder. In this directory will be two folders named "Terp Rockets Ground Station" and "makers". The "Terp Rockets Ground Station" folder contains the executable for the ground station. Running this should cause the ground station main window to launch. the other folder will contain the output of any makers in config.makers in package.json. By default, it should contain a zipped copy of the previously mentioned "Terp Rockets Ground Station" folder. From this point, the ground station is ready to run and the folder containing the executable can be placed anywhere on your computer.

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, 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)

- Map Cache Size (cacheMaxSize)

- Serial Port Baud Rate (baudRate)

The main window scale and the debug window scale both control the size of each window, as resizing the window manually is currently not available. The most likely use case for these settings is on lower or higher resolution displays, which may cause the windows to appear too large or too small respectively. It is recommended to set this to a value between 0.5 and 2.0 if necessary.

The debug mode and disable GUI settings are both 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 logging the last recieved transmission from an attached device to use when a device is not attached to test GUI functionality, opening the Chromium dev tools used by Electron to assist with debugging GUI elements, and logging debug statements. 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 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 recieve information in a specific format, or it will not function properly. 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=aaaaaa/Sx/HH:MM:SS
```

The Data field contains latitude, longitude, heading, speed, altitude, stage, and T-0 information, the format of which 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)

- Stage (Sx): current rocket stage as a number (eg. S0, S1, S2 etc.)

- T-0 (HH:MM:SS): the liftoff time in 24 hour format in hours, minutes, and seconds

You can also send other information in the Data field, it will not be shown in the GUI, but it will 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/dataformat.cpp. It is important that data be sent following the format listed above, or it may be difficult for the ground station to parse.

User Interface

The ground station user interface consists of two main windows: the main application GUI (refered to as the main window) and the debug window. The main window consists of two pages, which are the main page and the settings page. This sections aims to explain the different elements of each window and how to use them.

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 reciever device. These elements are highlighted and explained below.

1. The application name and logo

2. The transimission indicator

3. The Serial port selector

4. The Serial connection indicator

5. The signal strength indicator

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

12. Current stage name

13. Altitude indicator

14. Heading indicator

15. Speed indicator

16. Current latitude and longitude

17. Time since T-0

18. Altitude graph

19. Speed graph

20. Switch to graph view

21. Switch to map view

<Add image and more information for some numbers later>

The Settings Page

The settings page allows you to modify 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.

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. Dropdown menu for the main window scale

8. Dropdown menu for the debug window scale

9. Toggle for debug mode

10. Toggle for disabling the GUI

11. Input field for the map cache size

12. Input field for the serial port baud rate

13. Reset settings to defaults

14. Delete the map cache

<Add image and more information for some numbers later>

The Debug Window

As opposed to the main window, the debug window relies on a text based user interface similar to a command window. The basis of this interface is 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" allows you to control the debug window and open other windows. The full list of subcommands can be seen by running the "window" command, 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" 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 the "settings" command, and are explained in the list below.

- "-set": allows you to change settings, the subcommands are listed below, text between <> 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" command allows you to manage the application's connection

to your reciever device. It serves similar functionality to the Serial port selector in the main window. The available subcommands can be seen using the "serial" command, 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

Standard Workflow

Now that the core functionality of the ground station has been explained, this sections 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 intial connection to the recieving device up to the rocket's landing.

The Main Window

Intial Setup:

Before connecting the recieving 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 reciever. 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 recieving data. This is indicated by the transmission indicator flashing

green. Also check the signal strength indicator to ensure you are recieving 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 recieving 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 recieving communications, you will likely need to restart the ground station and reconnect the recieving device to your computer. Monitor the ground station during the launch, making sure it is recieving all expected transmissions. The ground station will try to detect apogee, but this may not be reliable, so monitor recieved data closely during this phase of 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. Once you no longer need to recieve transmissions, you can disconnect the receiving device from the ground station and close the application. A log of the data recieved during the flight is made available in the data folder in the same directory as the application. The file is in .csv format and named as the time the first transmission was recieved.

The Debug Window

Intial Setup:

Before connecting the recieving 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 reciever. 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 recieving data. A message will appear in the debug window for each message recieved. It will contain the recieved telemetry, along with the recieved 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 recieving 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 recieving communications, you will likely need to restart the ground station and reconnect the recieving device to your computer. Monitor the ground station during the launch, making sure it is recieving 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 recieved from the rocket will still be displayed, so you can locate the rocket using its longitude and latitude. Once you

no longer need to recieve transmissions, you can disconnect the receiving device from the ground station and close the application. A log of the data recieved during the flight is made available in the data folder in the same directory as the application. The file is in .csv format and named as the time the first transmission was recieved.

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