The Positron User Manual

Positioning Partners

May 2022

Table of Contents

Positron Rover Ground Control Station

Bitbucket repository: https://bitbucket.org/positioningpartners/pos-69-controller-ui/src/main/

Created by Positioning Partners for use with our Positron unmanned ground vehicle (UGV).

Visit us at: https://djm1271.uta.cloud/

GUI created with the help of PySimpleGUI: https://github.com/PySimpleGUI/PySimpleGUI

The Positron Rover Ground Control Station is a rover controller, developed in Python, that allows the user to control the Positron UGV.

This ground control station (GCS) allows the user to upload a mission for the Positron Rover to execute. A mission consists of a series of latitude, longitude, and job tuples separated by a comma. We've included an example mission.txt in the above repository. The GCS can activate Positron's paint module as well as track Positron's mission waypoints and *current location.

Prerequisites

Before you begin, ensure you have met the following requirements:

- You've installed the latest version of Python (3.9.0 or higher)
- You're using a Windows 10 machine. MAC OS is not supported.
- You have read this README as well as the Positron user manual.
- You have a telemetry device in your system's USB port that matches the one found on the Positron.
- The 'the_connection = mavutil.mavlink_connection('com7', 57600)' statement on line 37 of controller_UI.py
- will need to be modified to reflect the 'com#' port shown for your telemetry device.

Installation

To install Positron Rover Ground Control Station Software, follow these steps:

Navigate to our code repository given above and download the repository. Unzip the repository to a location of your choice. Open Windows Command Prompt and navigate to the newly installed CONTROLLER UI folder.

Dependencies

Install PySimpleGUI using 'python -m pip install PySimpleGUI --user'

Install folium using 'python -m pip install folium --user'

Install pymavlink using pip 'install pymavlink'

Install pygetwindow using 'pip install pygetwindow'

Execution

To run the program, navigate to the CONTROLLER UI folder using Windows Command Prompt. Once inside this directory use the 'python controller_UI.py' command to execute.

Example Run

When the Positron Ground Control Station starts the user is presented with a window that contains six buttons. The six buttons have a text terminal separating them with three on top and three beneath. There are two drop down menu options on the toolbar of the window. To start an example run of the program use the "Mission" drop down menu and select "Load". Enter 'mission.txt' to load the example mission. Next arm the rover by using the "Rover" drop down menu and selecting "Arm". Once Positron is armed all GCS functions can be executed. To view your mission map, click the "MAP" button. To add a waypoint, click the "WAYPOINTS" button. To remove the last added waypoint, click the "X" button. *To send Positron to the next waypoint click the "GO" button. *To stop Positron, click the "STOP" button. To activate Positron's paint module, click the "PAINT" button. To save a mission to a file use the "Mission" drop down menu and select "Save". Enter 'example.txt' to save your current mission. *These functions were not fully implemented but can be with the addition of the corresponding Mavlink commands.

Contributors

Thanks to the following people who have contributed to this project:

Daniel Marks

Nathan Fusselman

Oscar Mariscal

Deborah Jahaj

Kamal Karkidholi

License

This project uses the following license: GPL3.0 https://www.gnu.org/licenses/gpl-3.0.en.html.

Physical Design

Overview

The rover has been built using the old base of a wheelchair using a differential drive motor setup. This allows for the rover to be driven off roads and provides us with the necessary space and power to drive in all conditions possible.

Power Management

The rover is powered by two 12v deep cycle sealed lead acid batteries that are placed in the base of the rover and held in place between a metal plate and a 3d printed block. These two batteries are connected in series to provide us with the 24v that the motors demand. There is an emergency shutoff switch connected between the two batteries to provide the safest location to ensure that there is no possible short that could result in safety events. This power is then routed through 3 breakers and switches, these control the entire rover for the one marked in red where it will provide power to all systems and the two additional black breakers are connected internally and supply power to the two motors after the motor controller to allow for a direct shutoff for the motors allowing for you to push the rover without issues with it powered on and additional safety. These are connected where if the motor on either side where to draw too much power they would both trip together as to prevent the rover from having one side trip and the other still to work. This 24v power is supplied directly to the motor controller and to the power brick mini to provide battery management information to the rover as well as provide power to the cubepilot.

Motor Controller

The Rover runs on a Sabretooth 32x2 motor controller that provides incredible performance and easy interfacing. This motor controller can be reprogrammed using the DEScribe software that it comes with but is currently setup for independent control with R & L motors and controlled with PWM Servo signals sent to the device from the Cubepilot. This allows for easy communication and is the most common. This controller also supports UART and USB communication, but this is the easiest and way we picked. We have attempted to connect the internal breaking in the motors to the Sabretooth with success but found no advantage and the wiring was more complicated that it was worth, so this was undone and is not recommended.

CubePilot

The cubepilot is the brain of the rover and runs the Ardupilot software. This platform allows for quick integration with many different standards and can be configured quickly once you get an understanding for how it works. The hardware for this is quite simple however the configuration can be anything but at times. It is recommended to use the Mission Planner software as the PX4 firmware does not fully work and the Ardupilot software had issues with the QGroudControl software. This software allows for quick configuring of parameters and other configuration options. There are many online videos and forums that can aid with this if you have issues. Please look over any changes that we have made first and write them down as there is no way to backup these preferences really.

Here 3 RTK GPS

The rover has a Here 3 placed on top that allows the rover to connect to all GPS satellite constellations. This interfaces with the cubepilot using a CAN communication method and can obtain 2.5cm level precision when paired with an RTK base station. We recommend using the RTK station on the roof of Nedderman for this, but we also have a Here + that can be used to create your own base station at your computer however unless you have a Golden Point to reference you must let it survey in and this can take hours of staying in the same place to get comparable accuracy. This RTK data can be provided to the rover using an NTRIP caster like the one provided on Nedderman or the local station and piped to the rover in the Mission Control software. This is a feature we always wanted to add to our base station to send this data from our Python Program and avoid needing Mission Planner for this.

RFD900x-US Telemetry Radios

We use the RFD900x-US telemetry radio system for the best performance when connecting to the rover. This provides up to a theoretical 40km line of site distance and allows for fast connections to the rover. These communicate using standard UART on both sides and can be used as simple radio antennas.

Important Links

Positioning Partners website: https://djm1271.uta.cloud/

GCS source code: https://bitbucket.org/positioningpartners/pos-69-controller-ui/src/main/

Project documentation (Charter, SRS, ADS, Poster): https://djm1271.uta.cloud/documentation/

Project video: https://www.youtube.com/watch?v=eyFeLz77WJ8