AEROJET PAYLOAD: BLUE RC Rover Critical Design Review

Group 28

Justice Cordova	EE
James Ellison	CPE
Wesley Fletcher	CPE
Joshua Kissoon	CPE

Project Overview

Presenters: Wesley Fletcher

Project Overview/Goals

- Creation of ground-based rover payload and canister for integration into 10K rocket to be deployed at apogee

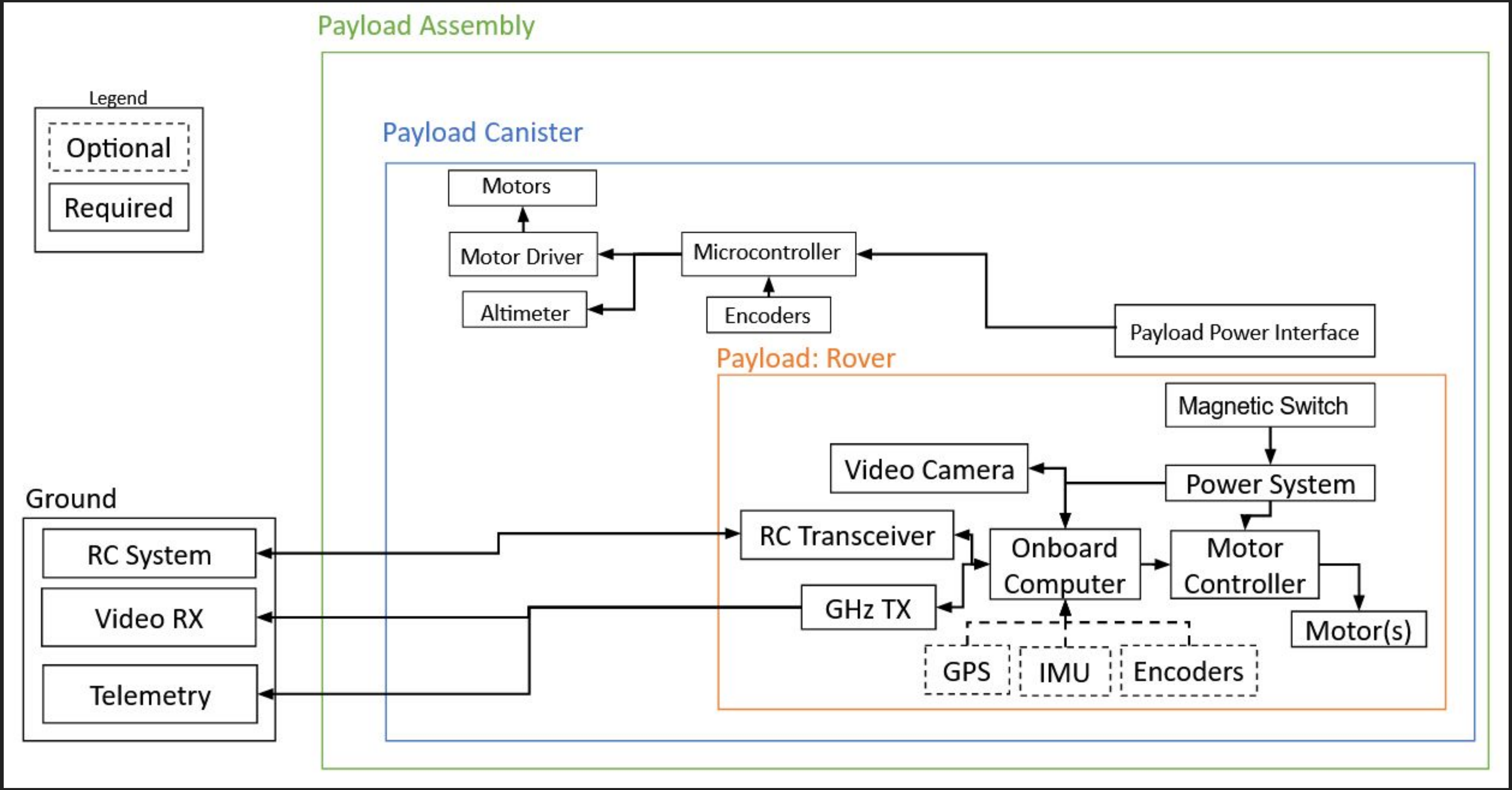
- Must satisfy FAR requirements:
  - Capable of remote control from ground station and live video transmission
  - Travel at least 10 feet

- Requires 3 subsystems:
  - rover payload: ground-based robot
  - payload canister: protects rover during deployment, descent, and landing
  - the ground station: allows communication with rover at distance

System Requirements/Constraints

Trace #	Requirement	Value (if applicable)
R1.0	Mass and Dimensions	
R1.1	Payload assembly max weight (sled, canister, and payload combined)	4.31kg
R1.2	Payload (rover) minimum weight	1kg
R1.3	Payload Canister dimensions	12.7cm diameter 40.64cm length
R1.4	Payload Sled Dimensions	Maximum 1.27cm depth on either side of payload
R2.0	Payload Sled	
R2.1	Payload Sled shall release Payload Canister at deployment phase via force generated by parachute.	-1.0 G
R3.0	Payload Canister	
R3.1	Payload Canister shall contain the Payload during launch, deployment, descent, and landing phases.	N/A
R3.2	Canister shall provide power-on signal to Payload during deployment phase.	N/A
R3.3	Canister shall act as RC signal relay for payload after landing.	N/A
R3.4	Canister shall open under its own power on command to release the rover.	N/A
R3.5	Canister shall be reusable for multiple potential launches.	N/A
R4.1	Rover shall travel the specified distance from the Payload Canister landing site under its own power.	10ft
R4.2	Rover shall be radio-controlled.	N/A
R4.3	Rover shall transmit a live video feed back to the Ground Station at given range.	<=600meters
R4.4	Rover shall be dust-resistant.	IP60
R4.5	Rover shall be able to determine position and orientation relative to Payload Canister.	N/A
R4.6	Rover shall be able to generate and store a rough map of immediate surrounds	N/A
R4.7	Rover shall create a 360 degree horizontal panorama on command.	N/A
R4.8	Rover shall log all relevant sensor and control data during operation for later retrieval.	N/A
R4.9	Rover shall transmit real-time telemetry data back to ground station at defined intervals.	N/A

Overall System Block Diagram



Payload Canister Subsystem

Presenters: Joshua Kissoon

Payload Canister Overview

- Canister keeps the rover secure and protected, and initiates rover deployment
- Aluminum 6061-T6 exterior, stainless steel fasteners
- Deployment consists of a linear actuator design using bipolar stepper motors, lead screws, and shaft couplings

Canister Subsystem Requirements

Trace #	Requirement	Value (if applicable)
R1.0	Mass and Dimensions	
R1.1	Payload assembly max weight (sled, canister, and payload combined)	4.31kg
R1.2	Payload (rover) minimum weight	1kg
R1.3	Payload Canister dimensions	12.7cm diameter 40.64cm length
R1.4	Payload Sled Dimensions	Maximum 1.27cm depth on either side of payload
R2.0	Payload Sled	
R2.1	Payload Sled shall release Payload Canister at deployment phase via force generated by parachute.	-1.0 G
R3.0	Payload Canister	
R3.1	Payload Canister shall contain the Payload during launch, deployment, descent, and landing phases.	N/A
R3.2	Canister shall provide power-on signal to Payload during deployment phase.	N/A
R3.4	Canister shall open under its own power on command to release the rover.	N/A
R3.5	Canister shall be reusable for multiple potential launches.	N/A

FAR 1030 Requirements

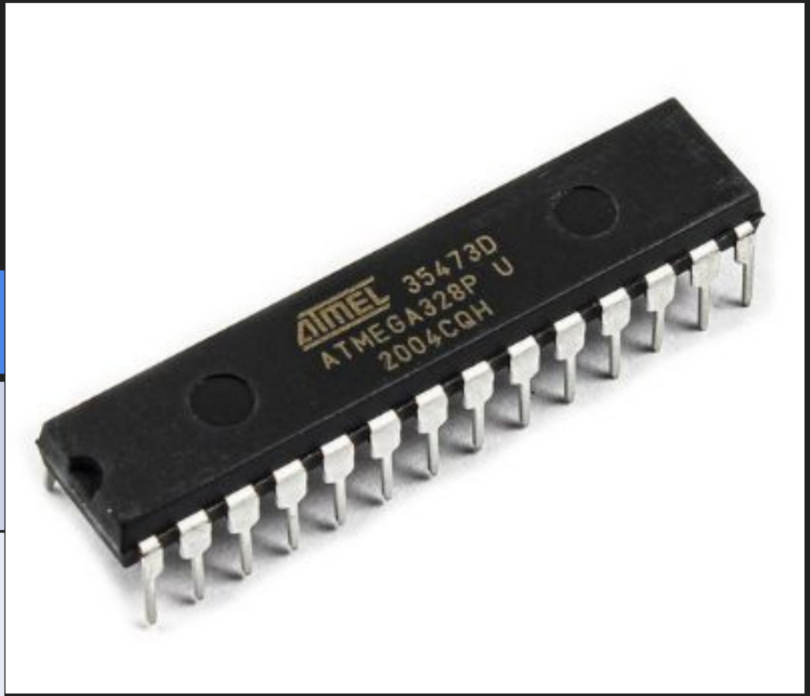
Implementation Specifications

Canister Parts Selections

Parts Selection - Dev. Board/MCU

- Integrates the canister's electronics
- Needs to be compatible with all parts
- Selected Arduino Nano(ATmega328P)
  - Both are very similar
  - Teensy pins are not 5V tolerant

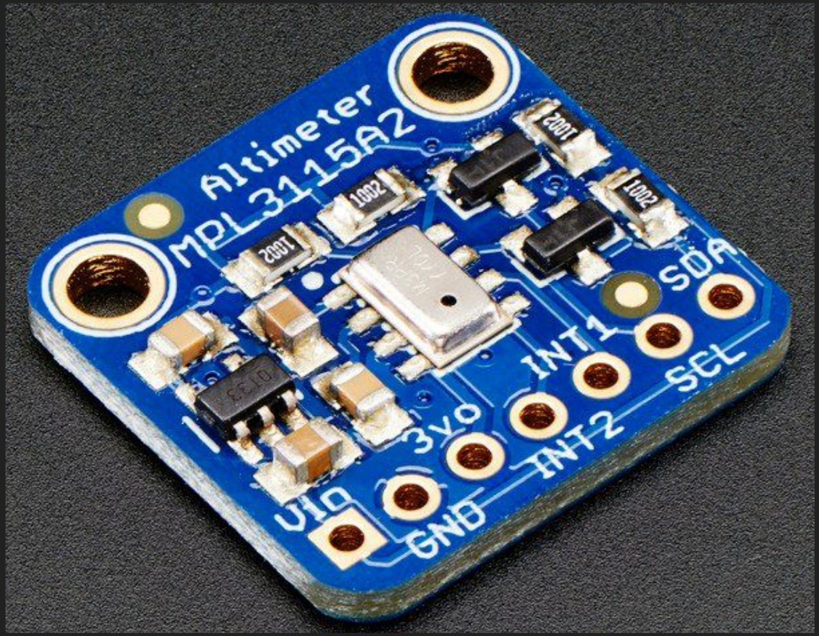
Option	Price	Comparison
Arduino Nano (ATmega328P)	$20.70	● 16 MHz clock speed ● 5V logic voltage ● I2C/SPI
Teensy 4.0 (ARM Cortex-M7)	$19.95	● 600 MHz clock speed ● 3.3V logic voltage ● I2C/SPI ● More GPIO pins/functionality



Parts Selection - Pressure Sensor

- Canister needs to know when to deploy the payload
- Almost identical specifications
- Freescale MPL3115A2 selected
  - More accurate

Option	Price	Comparison
Freescale MPL3115A2	$9.95	● I2C ● 3.3V operating voltage ● 2mA peak current draw ● 1.5 Pascal resolution (±0.3 meters)
Bosch BMP-388	$9.95	● I2C/SPI ● 3.3V operating voltage ● 0.8mA peak current draw ● 8 Pascal resolution (±0.5 meters)



Parts Selection - Stepper Motors

- Canister needs a mechanism to deploy the rover
- 3.2 oz-in per motor required for 4.31kg load
- Selected NEMA 11
  - Smaller, lightweight, less current, fulfills requirement
  - Slightly more expensive

Option	Price	Comparison
NEMA 11 (11HS20-0674D)	$14.27	● 9.91 oz-in holding torque ● 0.67A peak current draw ● 110g ● Half the volume of the NEMA 17
NEMA 17 (17HS13-1334D)	$9.52	● 31 oz-in holding torque ● 1.33A peak current draw ● 230g ● Twice the volume of the NEMA 11

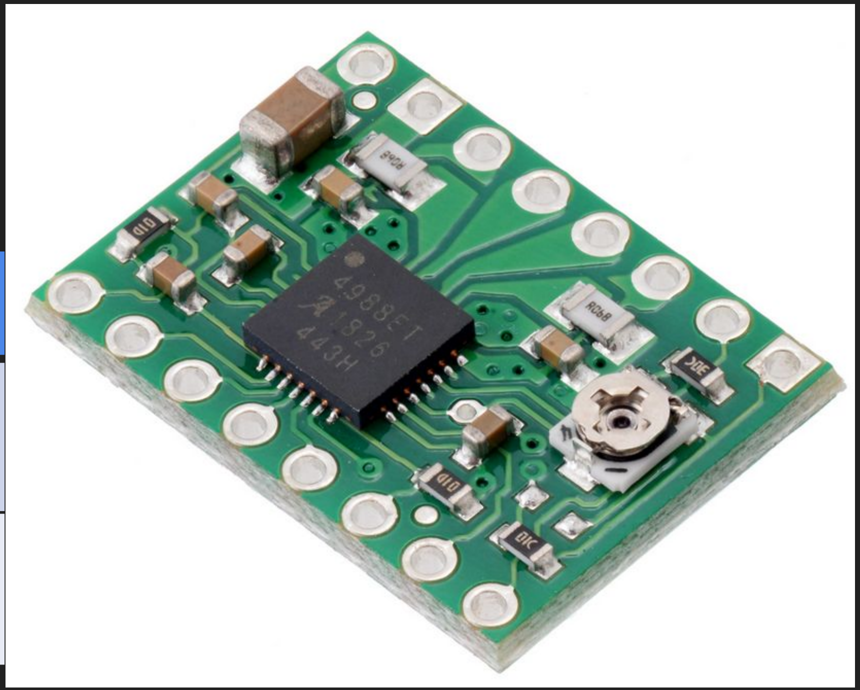
Parts Selection - Motor Encoders

- Measure motor RPM
- MCU needs to know when to terminate execution
- CUI Devices AMT102-V selected
  - Incremental Capacitive
  - Inexpensive compared to absolute

Parts Selection - Motor Drivers

- Motors need to be supplied current
- 0.67A per phase required
- Allegro A4988 selected
  - Cheaper
  - Small motors, low current

Option	Price	Comparison
TI DRV8825	$11.95	● 8.2V - 45V supply voltage ● 3.3V/5V logic voltage ● 1.5A per phase
Allegro A4988	$5.95	● 8V - 35V supply voltage ● 3.3V/5V logic voltage ● 1A per phase



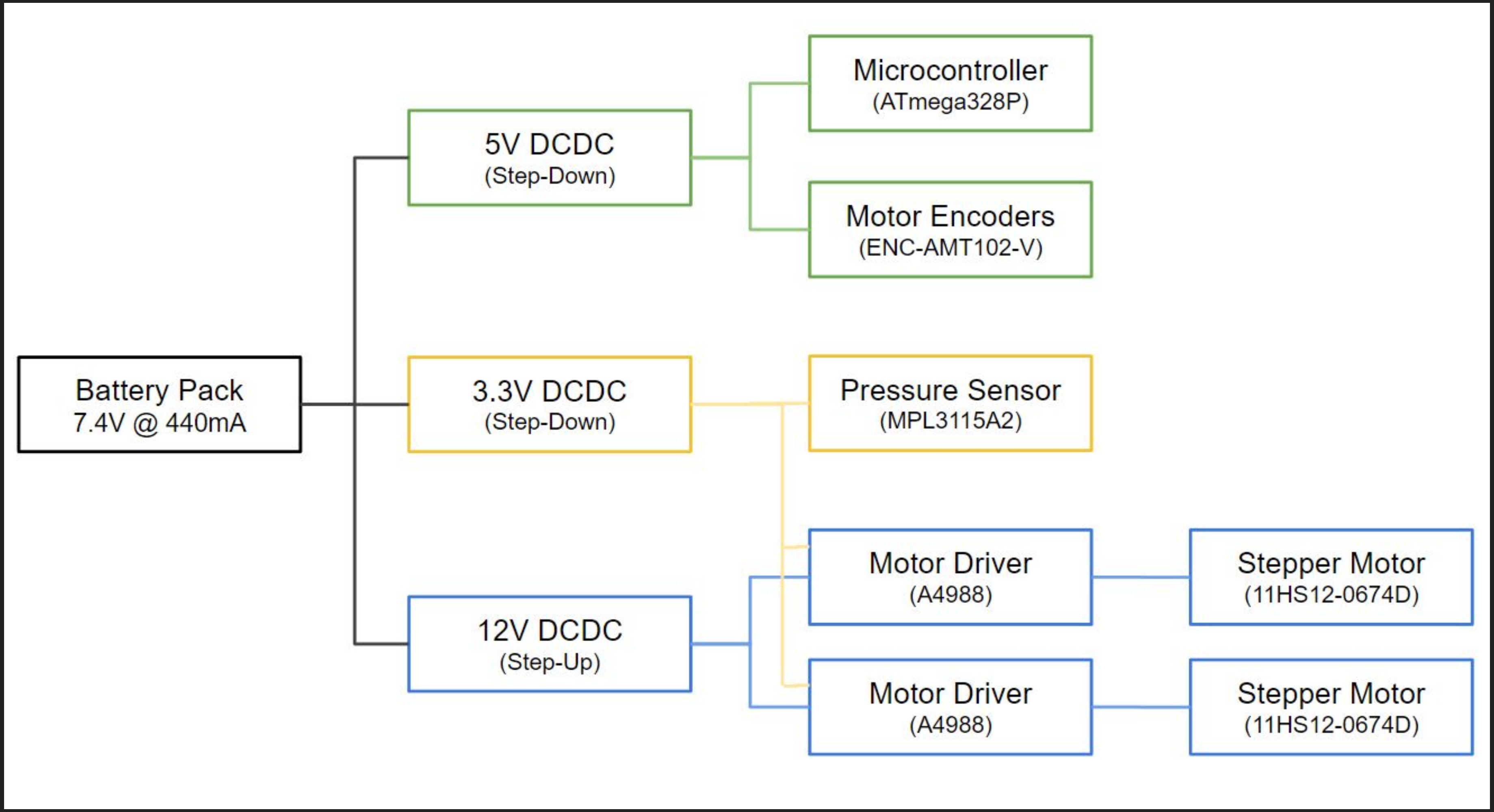
Parts Selection - Power supply

- External power source for our electronics
- Will need to last until rover is deployed
- Selected Li-ion battery pack
  - Lightweight
  - More efficient, cheaper

Option	Price	Comparison
9V(600mAh) + 9V(3000mAh)	$6.50 + $34.99 $41.49	● 345g ● Longer battery life ● More expensive
Li-ion battery pack(2200mAh) + voltage regulators	$15.99 $1.38 $2.25 + $6.32 $25.94	● 96.3g ● Requires voltage regulators ● Less expensive

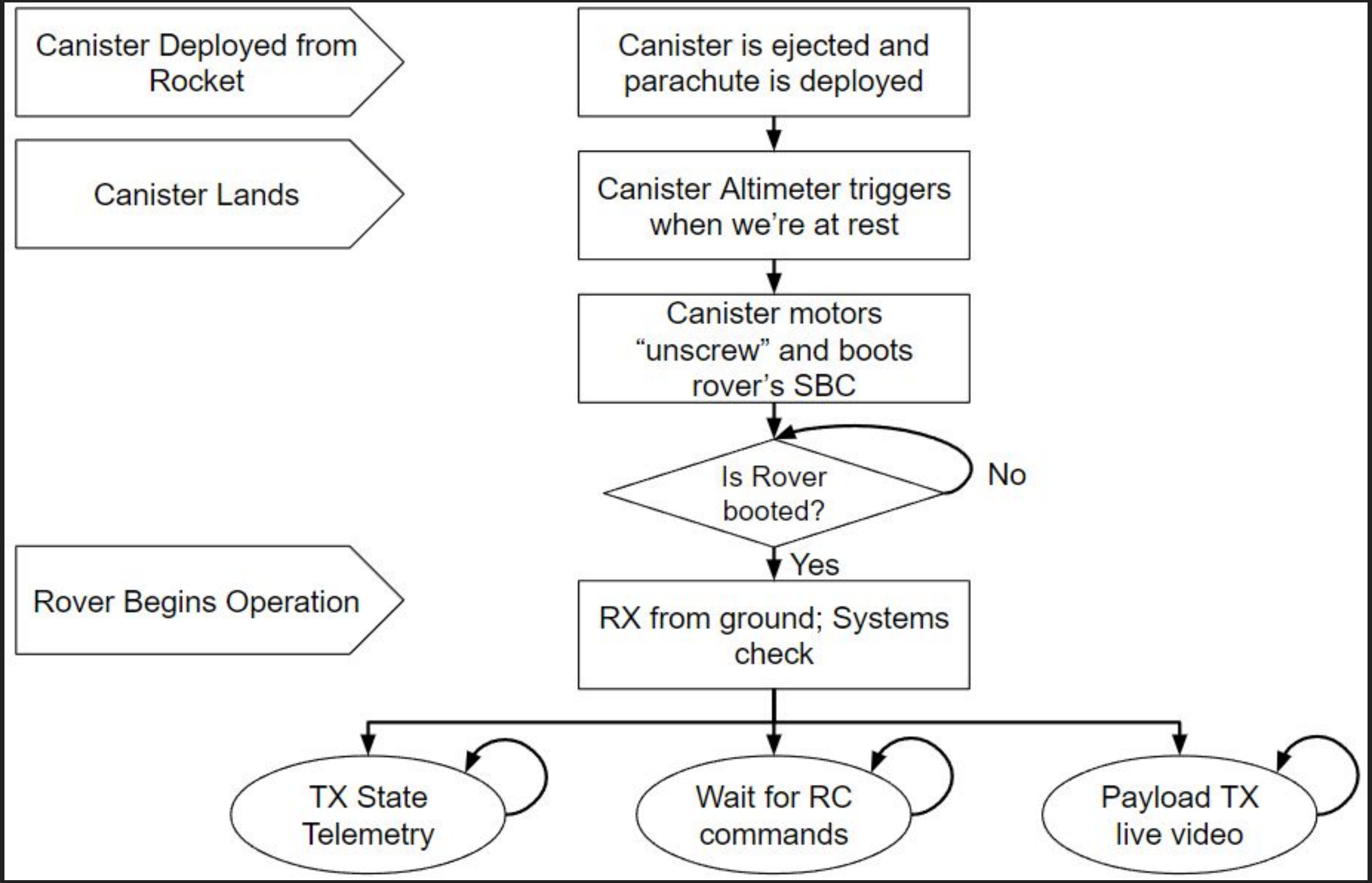
Canister Electrical Design

Canister Electrical Design - Power Tree



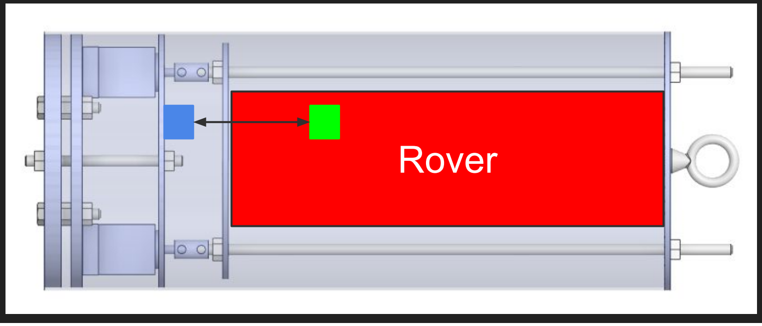
Deployment

Sequence Diagram

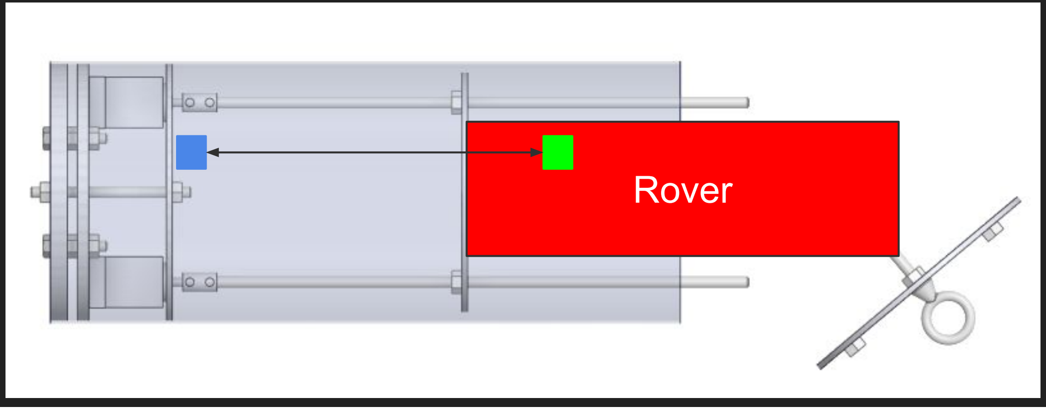


Deployment

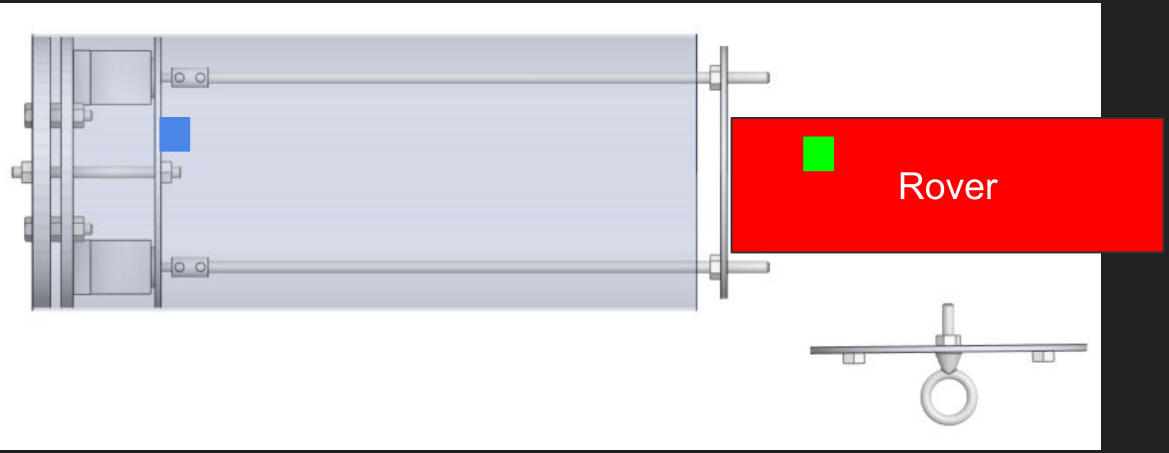
1



2



3



Rover Subsystem Overview

Presenters: Wesley Fletcher + Justice Cordova

Rover Subsystem Overview

- Rover is a ground-based robot designed to traverse short distances on uncontrolled terrain

- Live video transmission via 5.8GHz transceiver, radio control via LoRa (915MHz) transceiver

- Semi-autonomous waypoint navigation aided by:
  - IMU + GPS + Encoders for state estimation
  - Stereo camera depth clouds for environment mapping

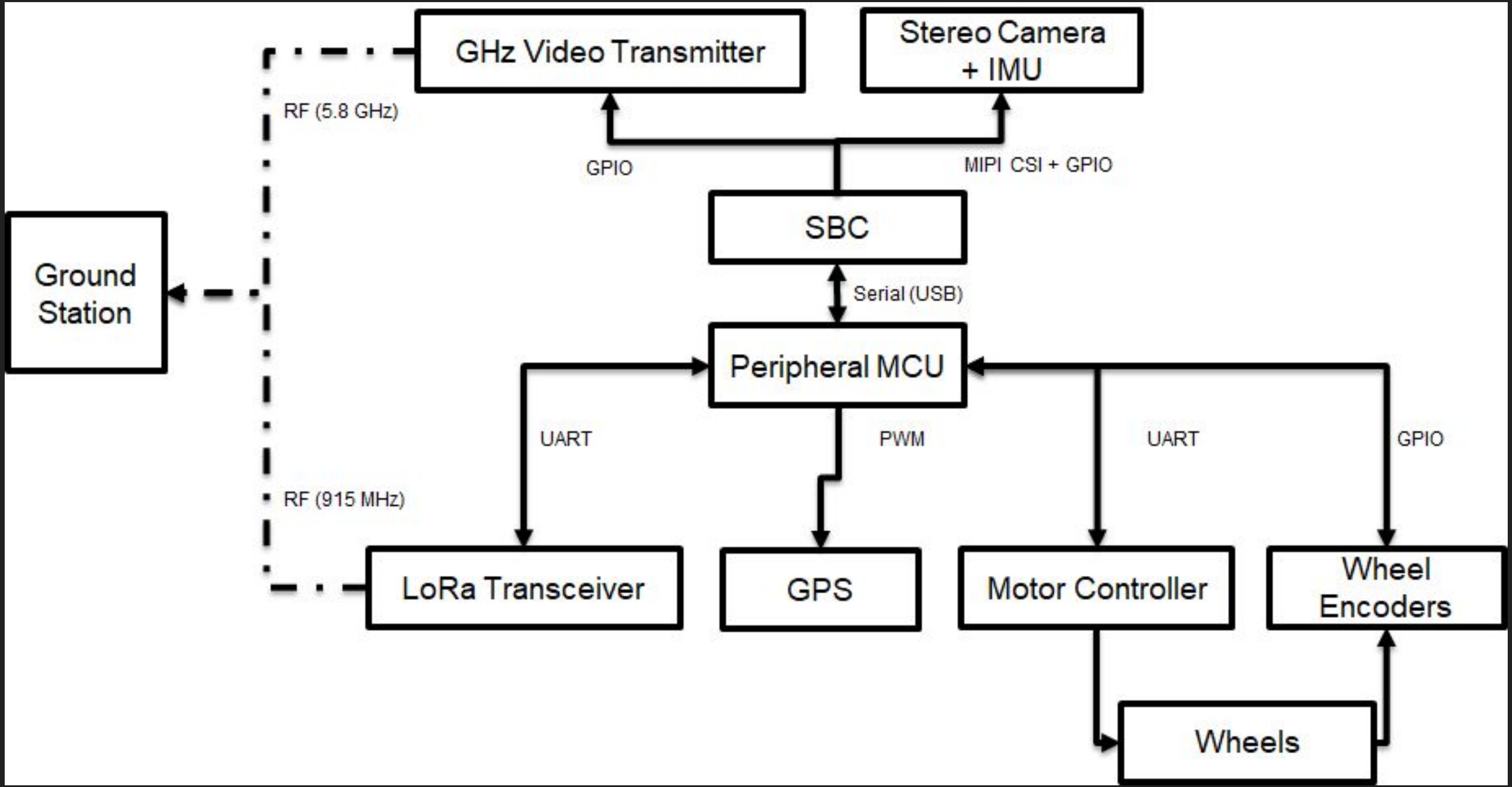
Rover Subsystem Requirements

R4.0	Payload (Rover)	
R4.1	Rover shall travel the specified distance from the Payload Canister landing site under its own power.	10ft
R4.2	Rover shall be radio-controlled.	N/A
R4.3	Rover shall transmit a live video feed back to the Ground Station at given range.	<=600meters
R4.4	Rover shall be dust resistant.	IP50
R4.5	Rover shall be able to determine position and orientation relative to Payload Canister.	N/A
R4.6	Rover shall be able to generate and store a rough map of immediate surrounds	N/A
R4.7	Rover shall create a 360-degree horizontal panorama on command.	N/A
R4.8	Rover shall log all relevant sensor and control data during operation for later retrieval.	N/A
R4.9	Rover shall transmit real-time telemetry data back to ground station at defined intervals.	N/A

Core Functionality

Advanced Functionality

Rover Hardware Block Diagram



Rover Parts Selections

Wesley Fletcher + Justice Cordova

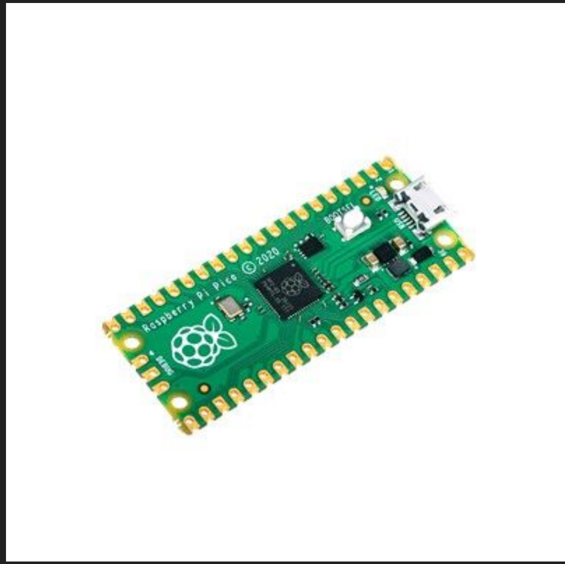
Parts Selection - SBC

- Single-board computer (SBC) on rover had to have sufficient performance for compute-intensive tasks like image processing, mapping, and ROS
- Selected **Jetson Nano** due to compute power, I/O, and immediate availability

Option	Price (MSRP)	Comparison
Jetson Nano Dev. Kit	$99	• More powerful, potentially less energy-efficient, larger footprint • Cheaper (since we already had one) • Immediately available
Raspberry Pi 4 B (4GB)	$55	• Smaller footprint, more power efficient, • More expensive (since we didn't already have one) • Potential back-order

Parts Selection - Peripheral MCU

- Need peripheral MCU to:
  - increase I/O options and availability,
  - decouple I/O from Jetson to increase modularity/flexibility of system and avoid CPU scheduling concerns (Jetson is not real-time)
  - keep Jetson GPIO available for last minute adjustments
- Selected **Raspberry Pi Pico** due to high clock speed, low cost, and easy availability

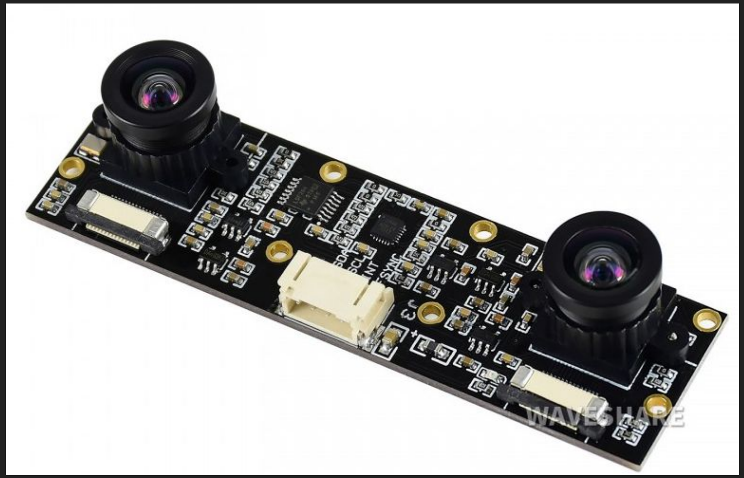


Parts Selection - Camera

- Directly satisfies requirement R4.3 (in combination w/ 5.8GHz TX/RX)

- Selected **Waveshare Stereo Camera** due to:
  - easy interface w/ Jetson Nano (2x MIPI CSI-2 connectors on dev. kit)
  - possibility of depth cloud from stereo image disparities (bonus for mapping)
  - integrated IMU sensor (bonus for state estimation)

Option	Price (MSRP)	Comparison
Waveshare Stereo Camera	$44.99	• Possibility of depth cloud through image processing (good for mapping environment) • Requires 2 MIPI CSI-2 connectors, only Jetson provides 2 • Integrated IMU • Most expensive option
Raspberry Pi Camera Module V2	$25.00	• Requires only 1 MIPI CSI-2 connector (can be used with many SBCs) • Single image stream, no depth cloud options
Leopard Imaging Camera - 136 degree	$29.00	• Requires only 1 MIPI CSI-2 connector (can be used with many SBCs) • Single image stream, no depth cloud options



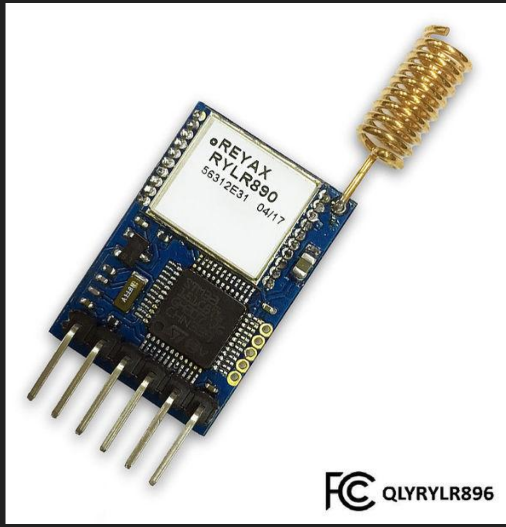
* each of these options uses the IMX-219 sensor, no difference in resolution

Parts Selection - Radio Transceivers

- Most important constraints are size and range
- ESP32 and LoRa considered
- LoRa selected to ensure rover meets range requirements

Option	Price (USD)	Comparison
Reyax RYLR896 LoRa RC Transceivers (2)	$48.93	-UART -Uses LoRa modulation technique, based on chirp spread spectrum technology -operating voltage 3.3 V -max current draw of 43 mA -Range >1 km
LILYGO TTGO dev module WiFi + Bluetooth (2)	$26.56	-I2C -Bluetooth and Wi-Fi protocols -ESP32 Commonly used in projects: lots of resources available online -Graphical Display for neat presentation of transmitted/received data -operating voltage of 3.3 V -max current draw of 67 mA -Range is limited to 300 m

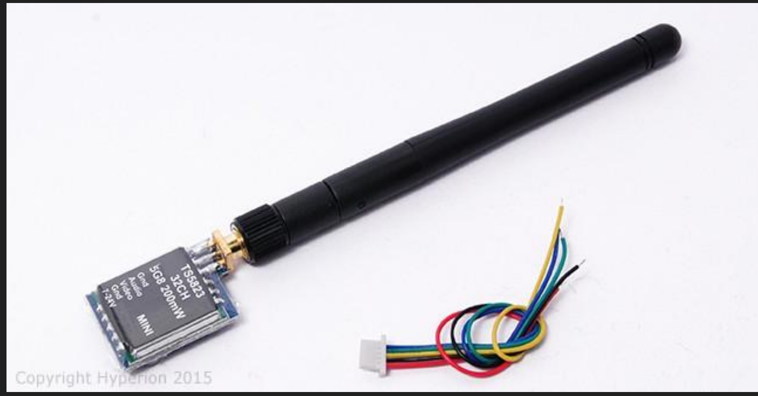
Parts Selection - Radio Transceivers (continued)



- Lora configured for 915 MHz operation: chosen to avoid operation on licensed bands and interference with rocket telemetry (430 MHz)
- Communicates using AT commands (AT commands translated by rover, ground station software)
- Receives RC commands, sends GPS Data
- Extensive field testing confirms range of >600m

Parts Selection - Video TX/RX

- Video TX cannot be achieved using LoRa
- With size constraints, options are severely limited for video TX
- Video TX will be biggest constraint on range of rover (~600m)
- Hyperion TS5823 is selected- only available option that meets size, range constraints-operates at 5.8 GHz
- Skydroid receiver-connects to ground station, broadcast to GUI



Parts Selection - Batteries

- LiFePO4 vs LiPo
- Major considerations are safety and battery capacity
- Decision was made to go with LiPo
  - Protective casing should reduce risk of thermal runaway
  - significantly higher energy density- LiFePo batteries could not be trusted to power all of the rover modules

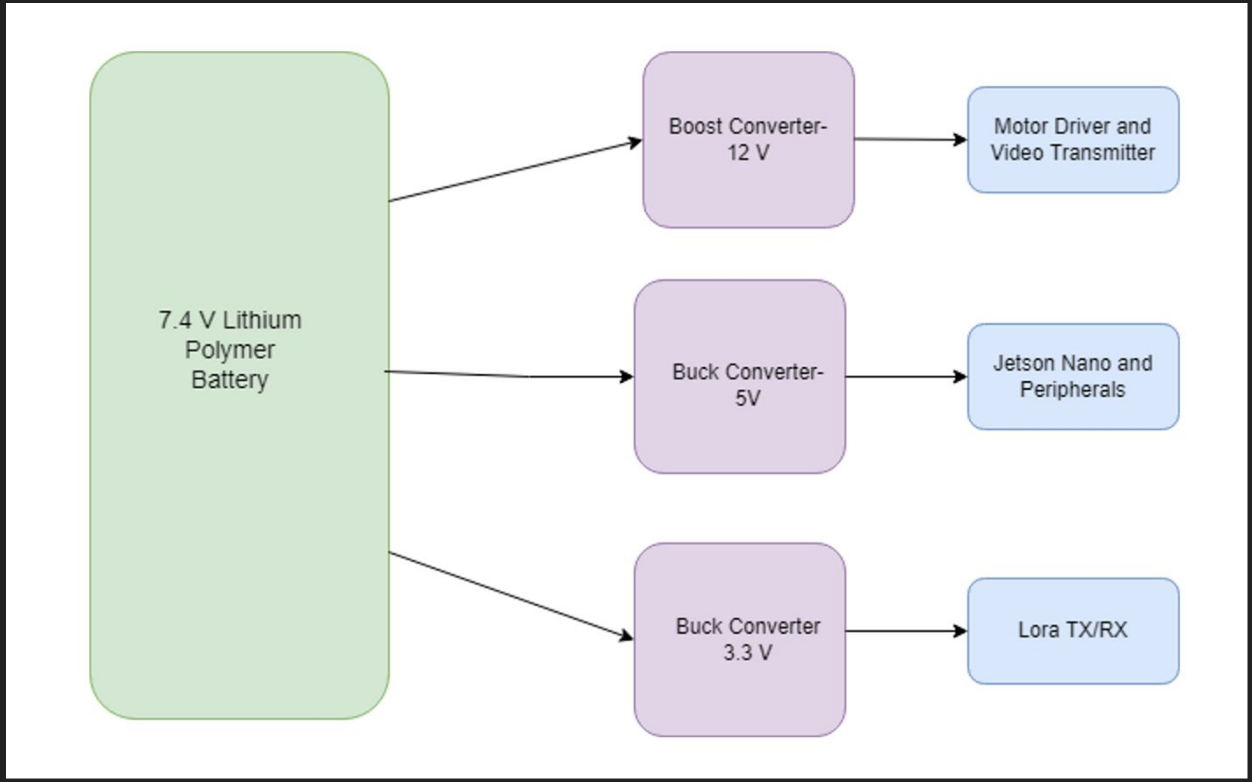
Option	Price (USD)	Comparison
LiFePo4 Battery (2s)	$23.93	-Size is identical to LiPo -Voltage 6.4 V -Current capacity 900 mAh -Lower Risk of thermal runaway
LiPo Battery (2s)	$27.89	-Identical size -voltage 7.4 V -Current capacity 4600 mAh -increased risk of thermal runaway

Rover Electrical Design

Justice Cordova

Rover Electrical Design - Power

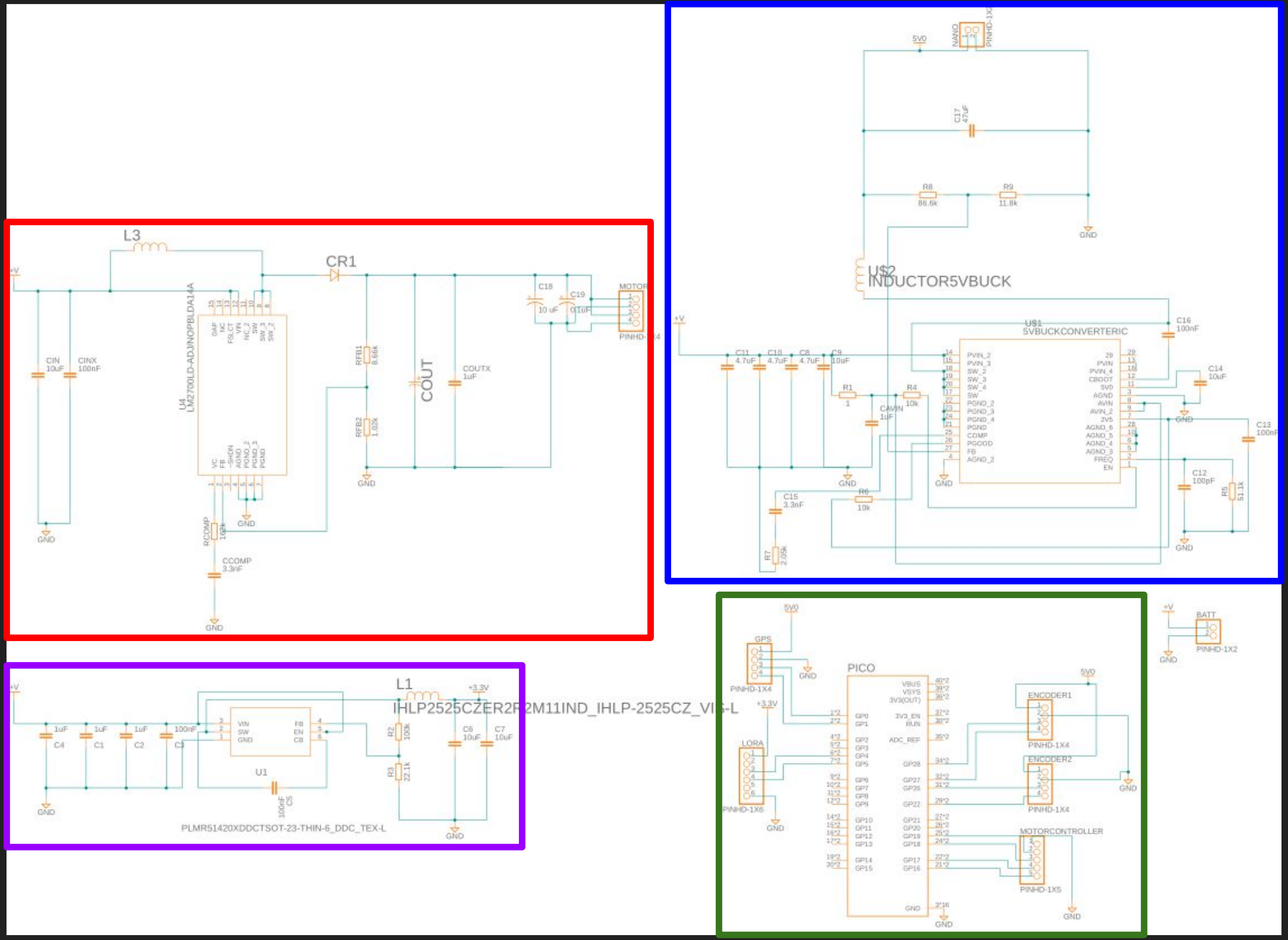
Power Electronics Hierarchy



Rover Electrical Design - PCB Schematic

Boost Converter-Motor and Video TX

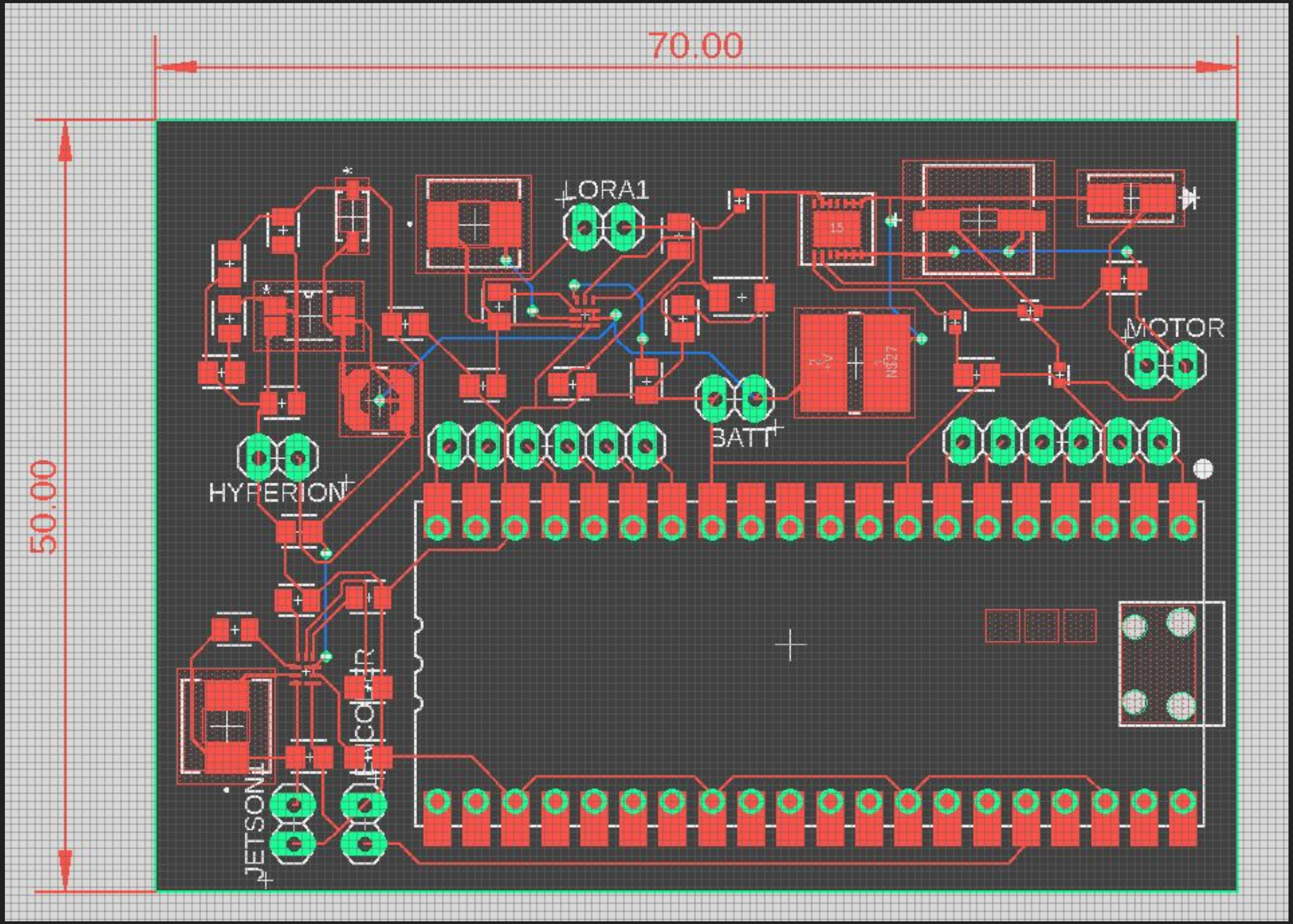
Buck Converter-Jetson/Peripherals



Buck Converter-LoRa

Raspberry Pi Pico

Rover Electrical Design - PCB



Rover Software Design

Wesley Fletcher

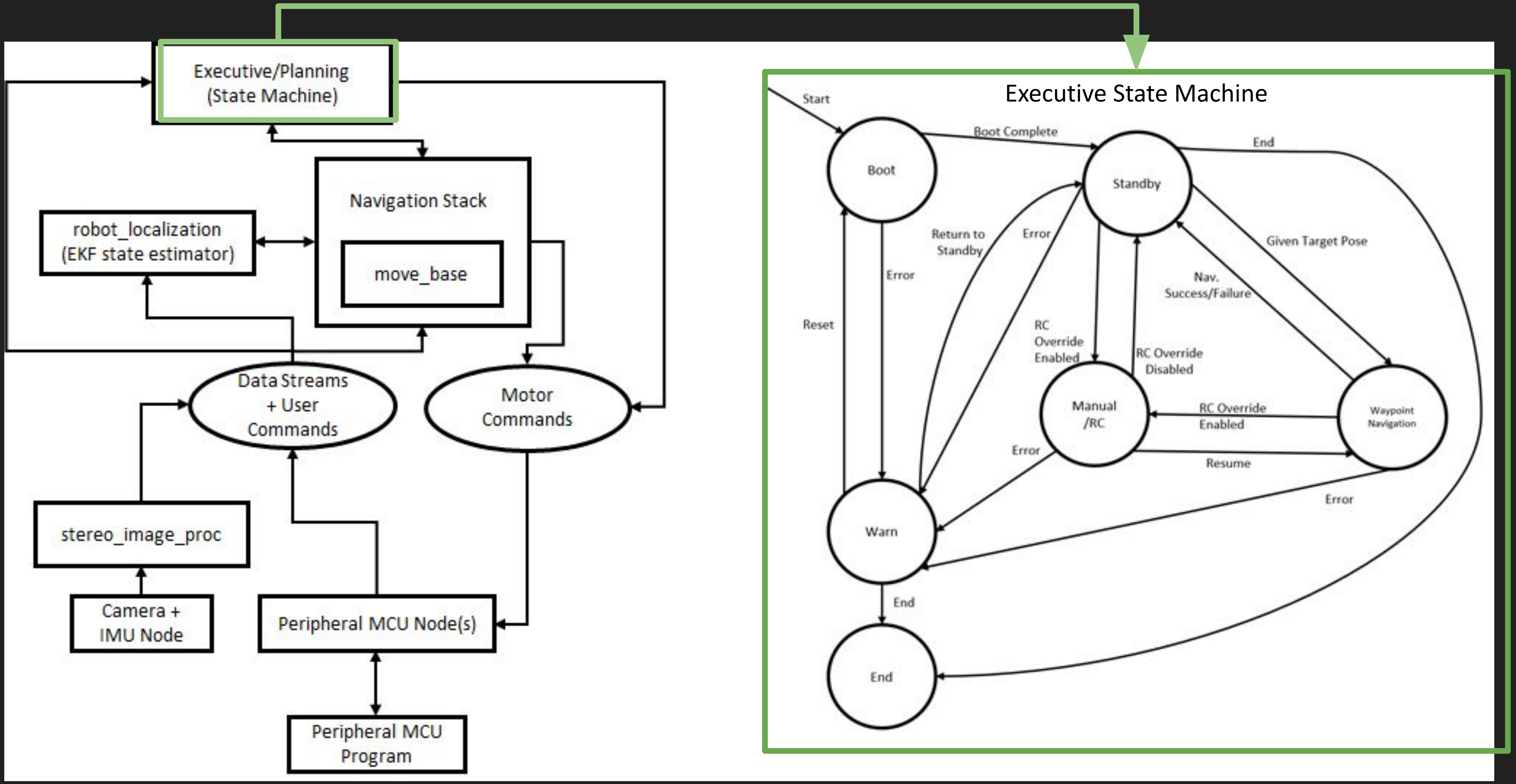
Rover Software Overview

- Using the Robot Operating System (ROS) to enable communication between nodes + provide access to off-the-shelf robotics functionality

- “Executive” program will be a finite state machine: clearly defined states and transitions that can easily be modified and extended

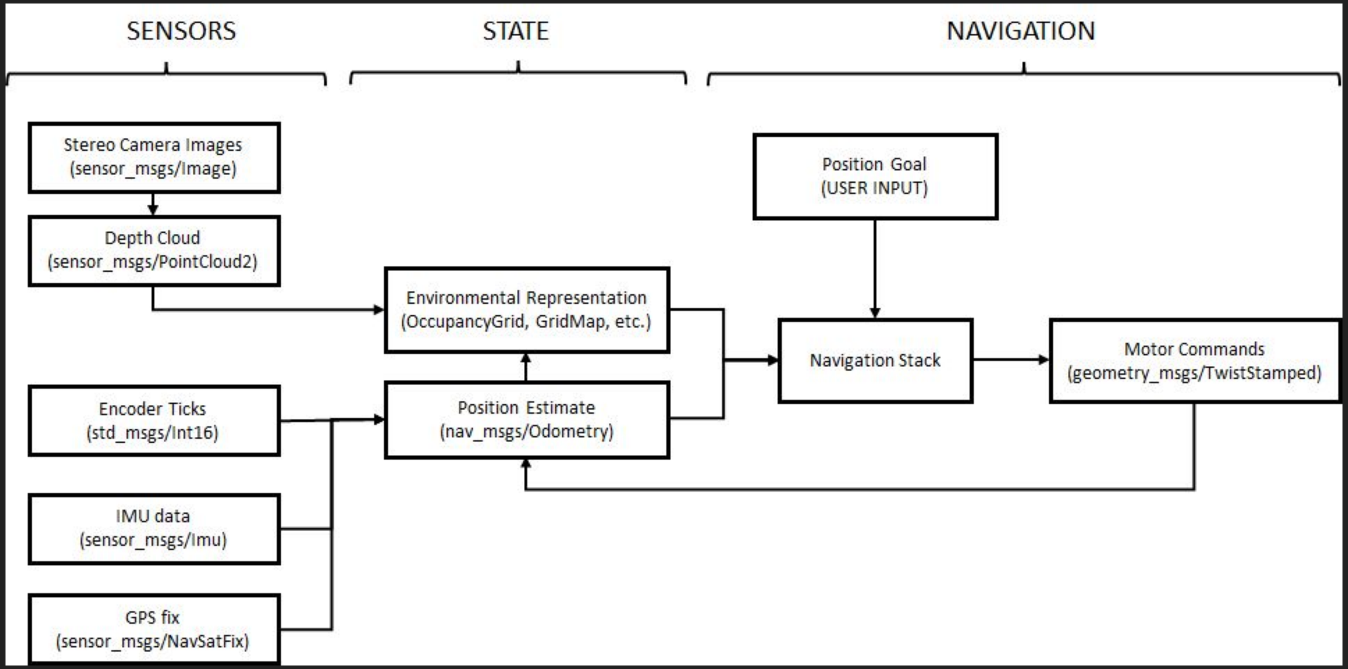
- Semi-autonomous waypoint navigation provided by ROS Navigation Stack, localization modules, and depth cloud processing

Rover Software Block Diagram



Rover Software - Waypoint Navigation

- Strategy to deal with communication latency - provide waypoints rather than real-time motor commands
- Rover in WAYPOINT state handles travelling to goals
  - utilizes move_base ROS package, translates high-level position goal to low-level motor commands



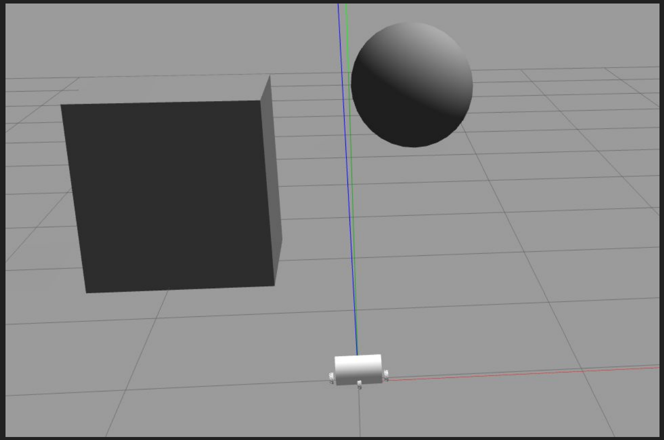
Rover Software - Peripheral Interface

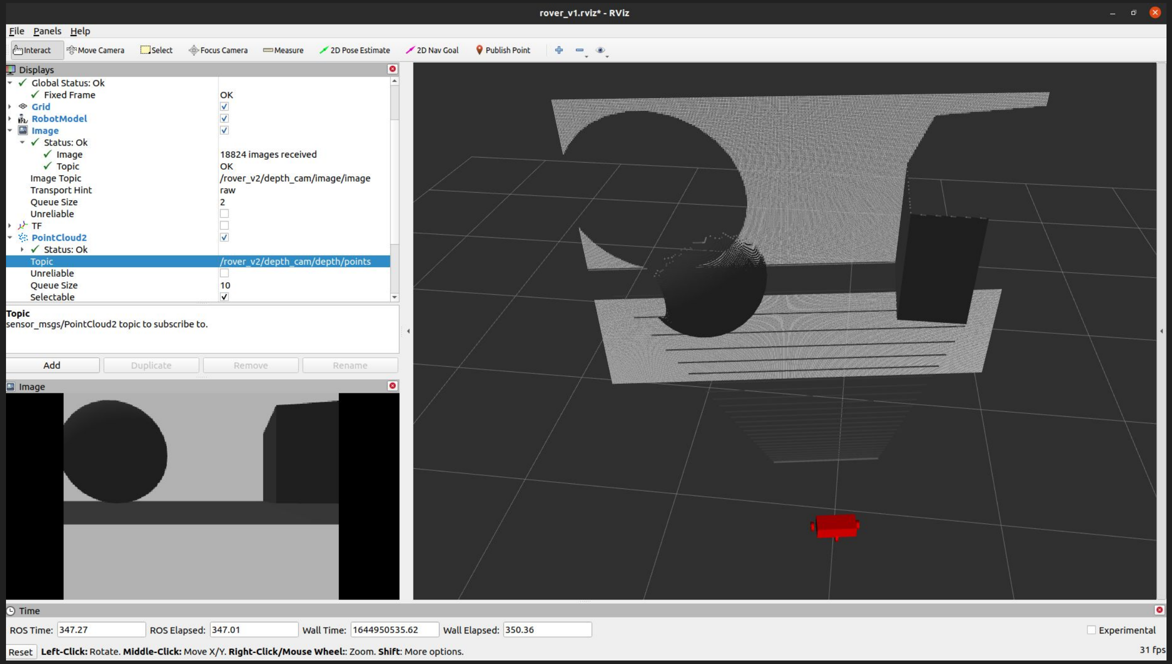
- Peripheral MCU is responsible for:
  - interfacing with sensors + LoRa transceiver
  - transmitting sensor data and RC commands TO the Jetson SBC
  - receiving/transmitting motor controls and telemetry FROM Jetson SBC

- Communication between paired software modules: the Peripheral MCU module and Peripheral Abstraction module

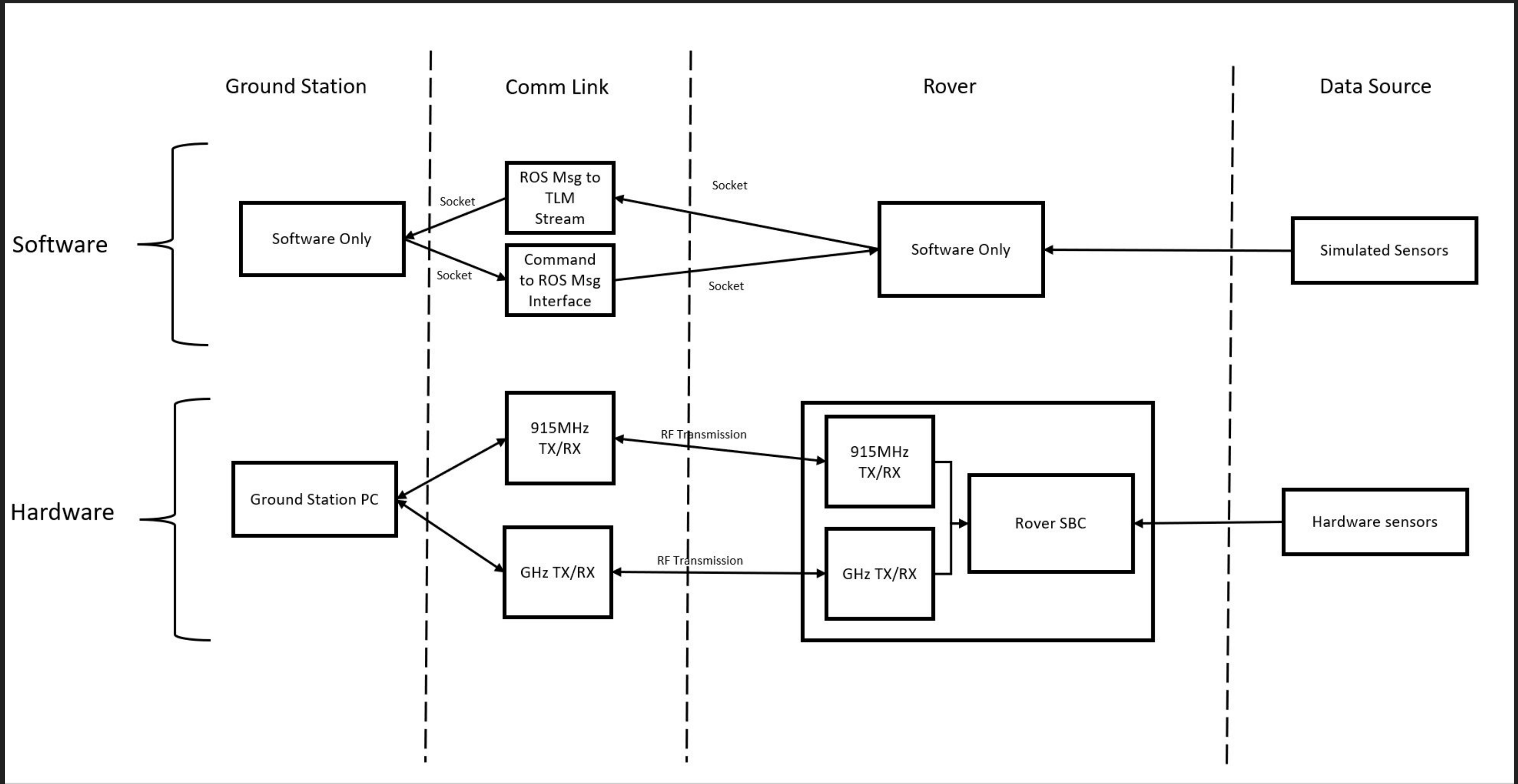
Rover Software Simulation

- Leverage Gazebo simulator and ROS integration to develop and test rover software without hardware considerations
- Past experience with this approach: nearly 1:1 parity between simulation and “real” software configurations



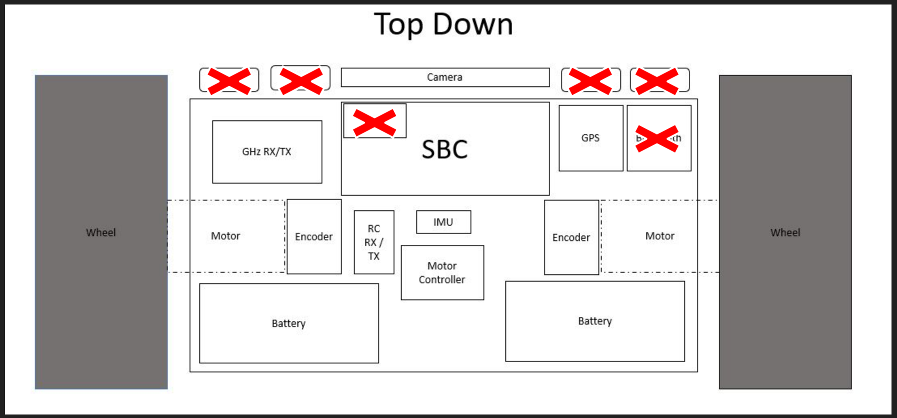


Rover Software Simulation Test Plan

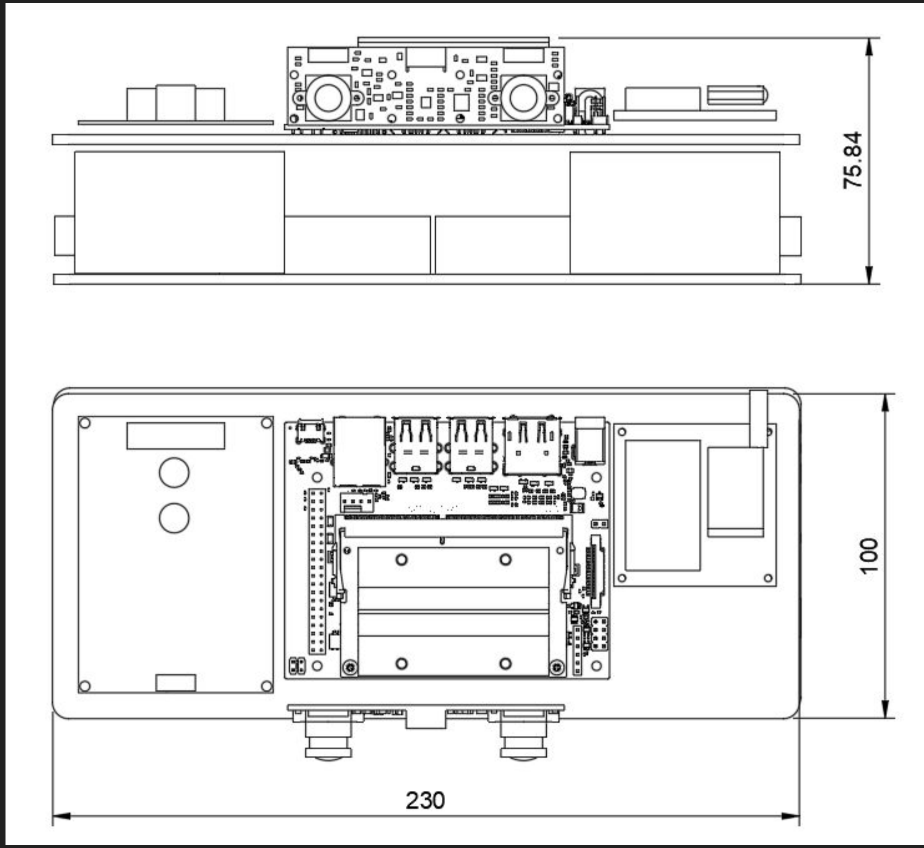


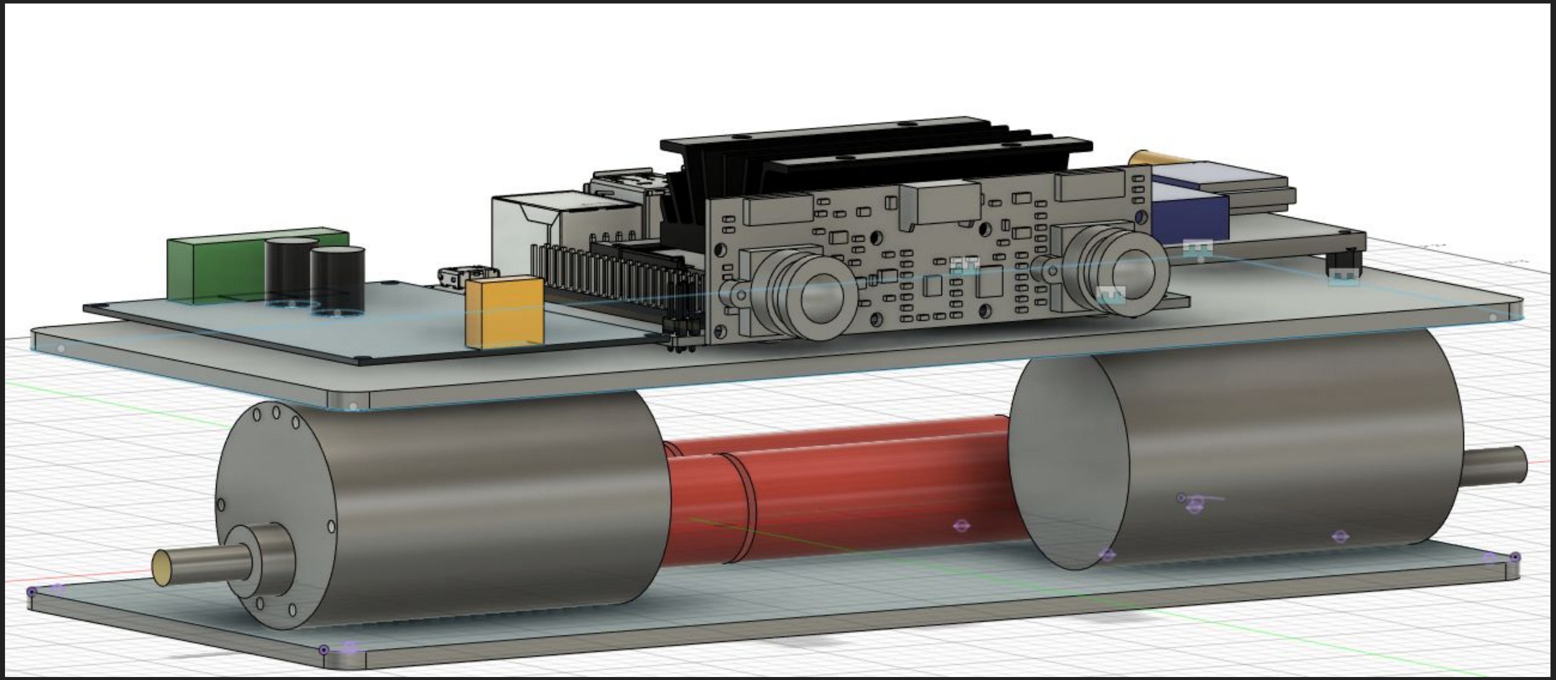
Software Test Plan

- Simulation Verifications:
  - Robot Model
  - Sensor streams
  - Motor Commands
- Software-In-the-Loop (SITL):
  - Localization
  - Mapping
  - Autonomous Nav.
- Communications:
  - Translation Nodes
  - Telemetry/Command messages
- Executive Finite State Machine (FSM):
  - States
  - Transitions



Rover Mechanical Design

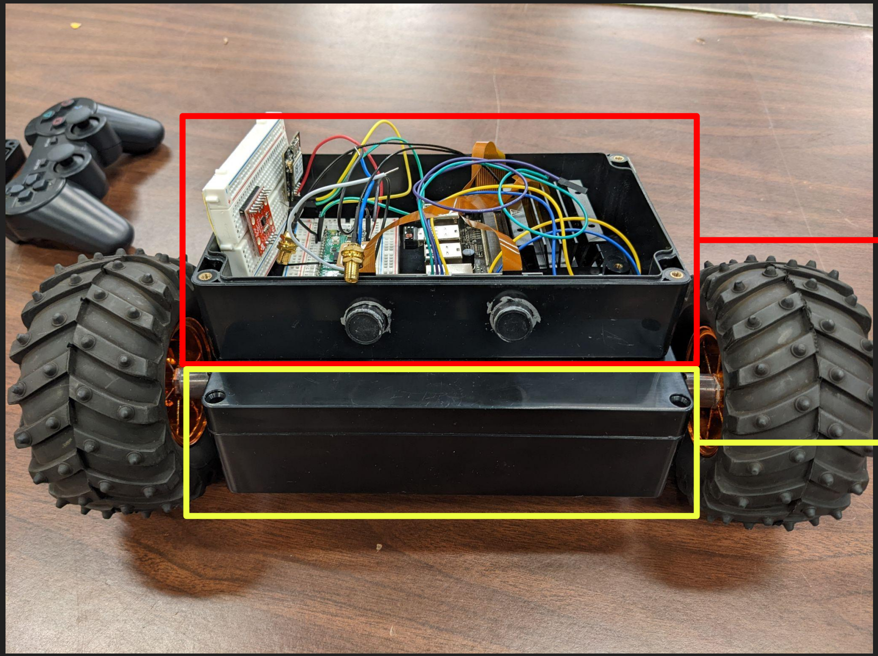




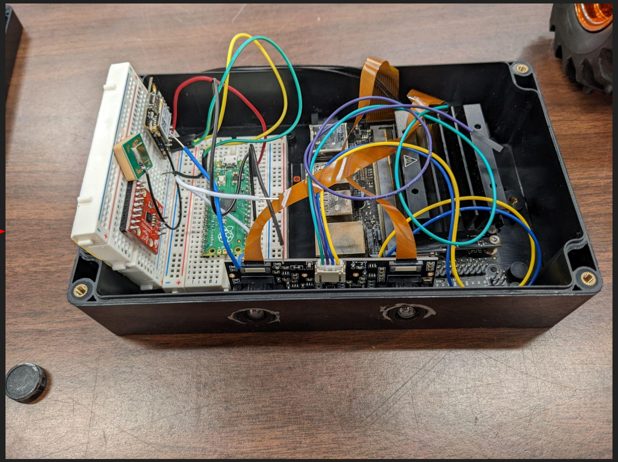
Action Items

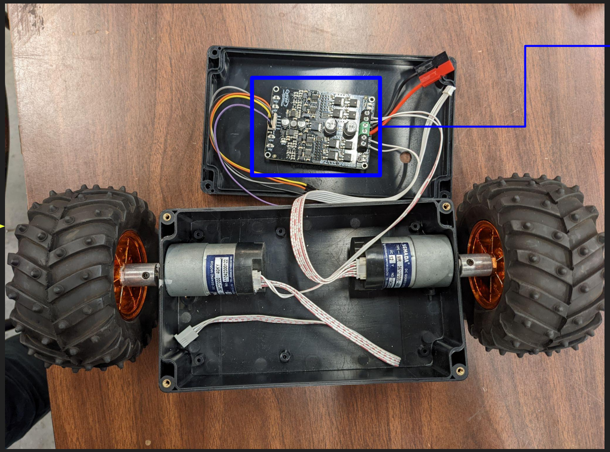
- Enclosure (Box w/ removable lid)
- Wheel selection
- Foot/sled for trailing end
- GPS and IMU placement

(In Progress) Rover Prototype



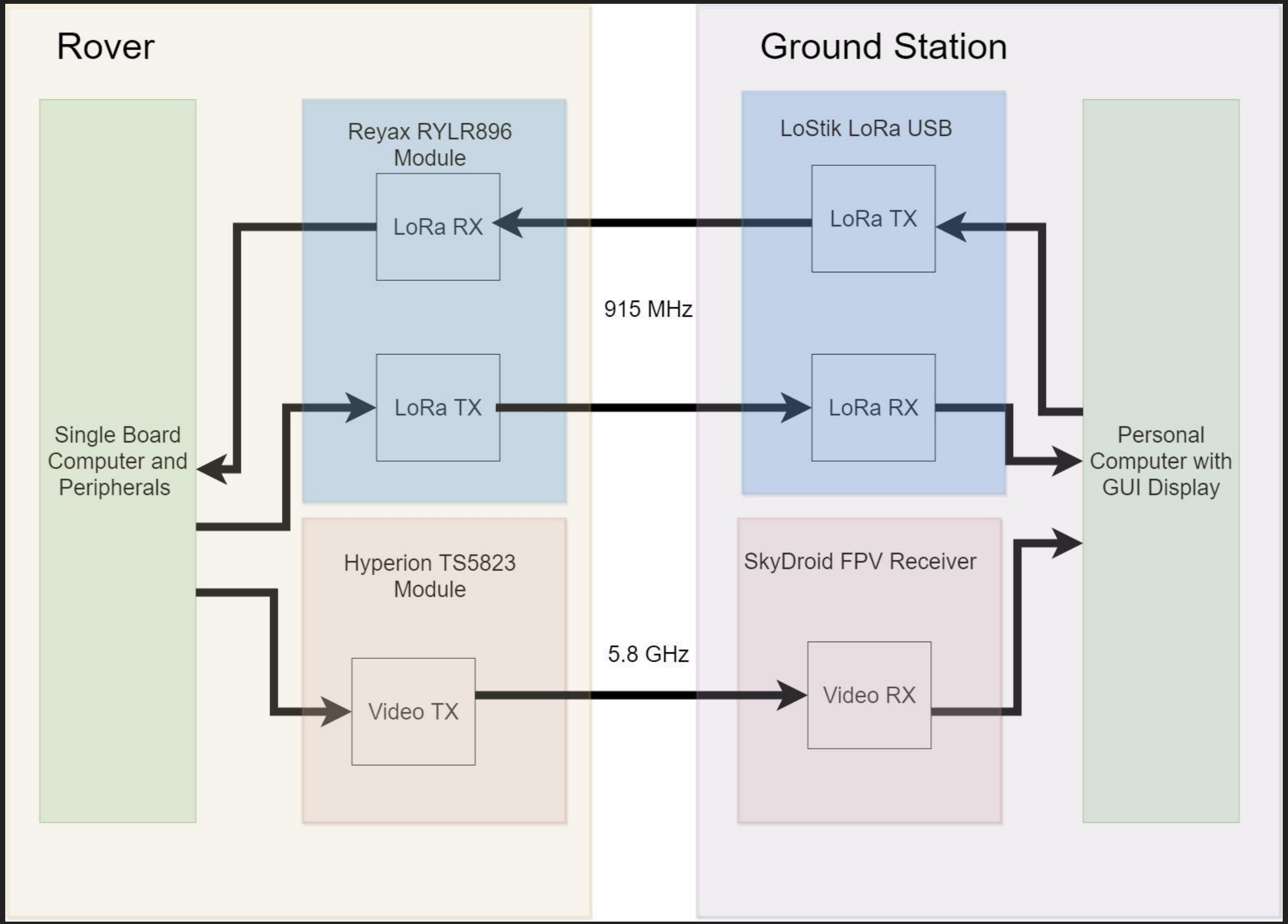
Top Level:
SBC, Stereo Camera, Power Dist., Peripheral MCU





Motor Controller

RF Communications



LoRa Messages Definitions

The following message formats will be used for the 915MHz LoRa communication link. Standardized message formats allow for modular design of this link and make HWIL and SITL simulations easier to develop.

Telemetry Message

- From the rover to the ground station. Contains information on robot state like GPS location, odometry, status info, etc.

Heartbeat Message

- Between the rover and the ground station. Small messages sent at reasonable frequency to ensure that our rover and ground station are still in communication.

Command Message

- From the ground station to the rover. Contains commands for the rover controller, such as RC controls for manual drive, waypoints for autonomous drive, info requests, etc.

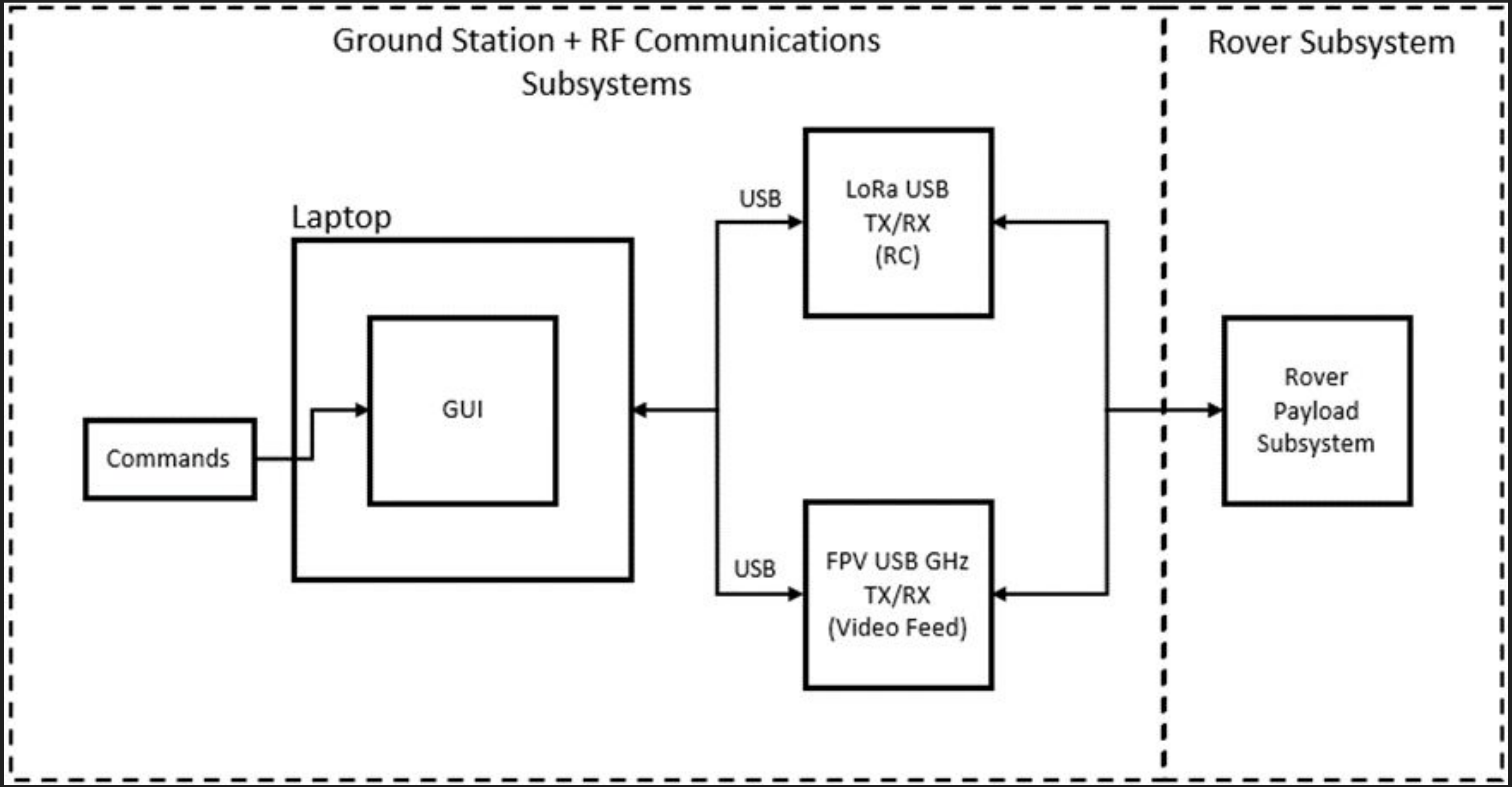
Ground Station

Presenters: James Ellison

Ground Station

- The Ground Station is the command center. It will send instruction to the rover and monitor its video and location.
- The Ground Station will use Ubuntu 20.04 as a host machine.
- A 5.8 Ghz video receiver and LoRa transceiver will be connect via USB.
- We will use a GUI for rover controls and GPS.
  - Used PyQt5 for creating and editing the base of the GUI.
  - The GPS data in the GUI will use PyQtlet.
- A third party application will be used for video feed.

Ground Station Block Diagram



GUI Prototype (Controls/Map)

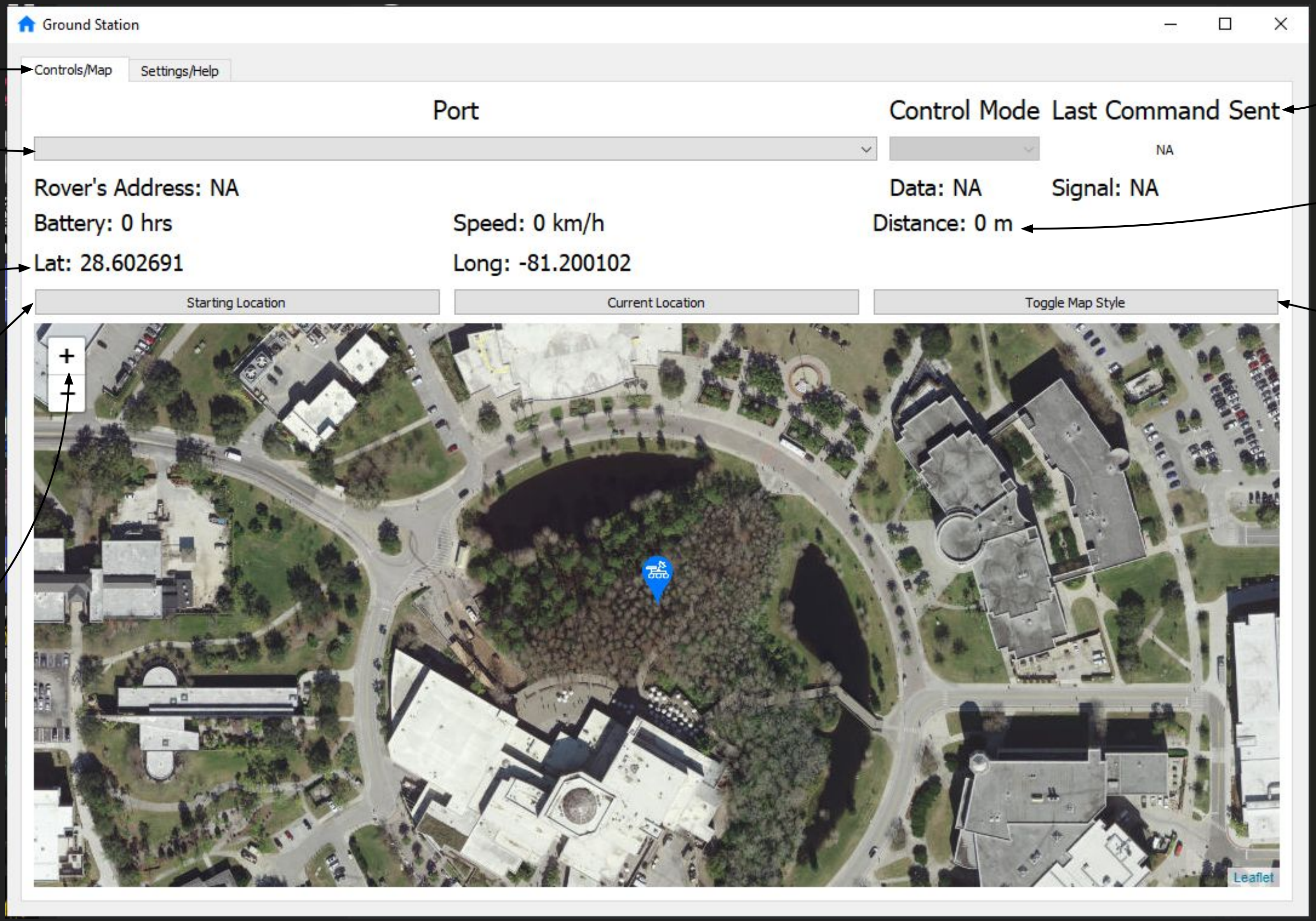
Switch between the Controls and Map or Settings

Port Selection for USB

Latitude and Longitude for specific location

Set Starting and Current Location

Zoom In or Out



Text Command: used to send command manually

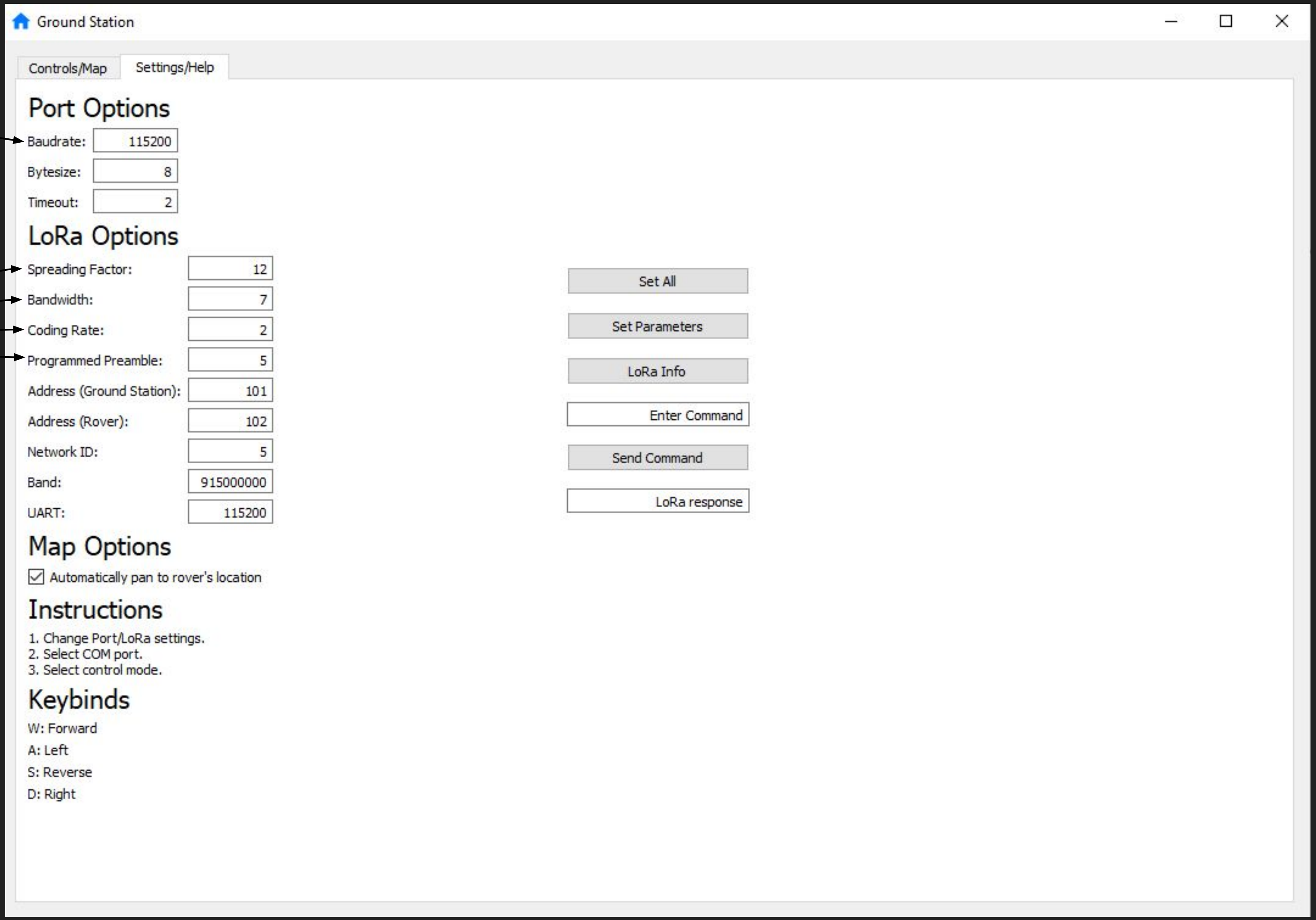
Distance from the starting location

Toggle between different map styles

GUI Prototype (Settings and Help)

Baudrate of 115200 is the default data transmission rate for LoRa RYLR896.

These 4 parameters can be adjusted, with a tradeoff between range and accurate data transmission



Administrative Content

Team Structure and Responsibilities

Justice Cordova	Joshua Kissoon	Wesley Fletcher	James Ellison
• RF Comms • Power Systems • PCB Design	• Canister Design • Canister Electronics • PCB Design	• Rover Design • Rover Electronics • Rover Software	• Ground Station • Rover Software

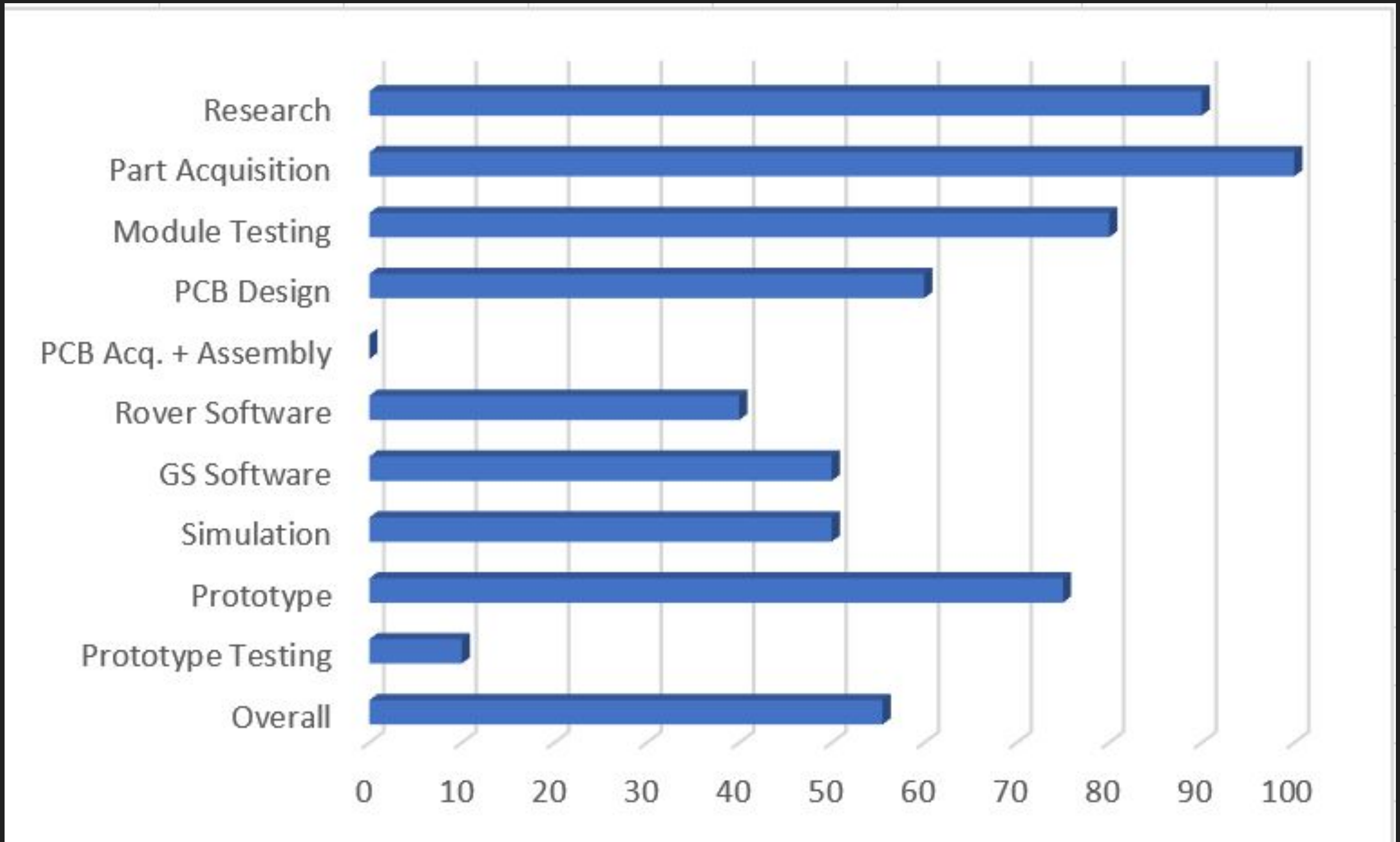
Budget + Financing

Rover				
Function	**Name**	**Price**	**Actual Cost**	**Acq. Status**
SBC	Jetson Nano 4GB Dev. Kit	$99.00	$0.00	ACQUIRED
MCU	Raspberry Pi Pico	$4.00	$4.00	ACQUIRED
Camera	Waveshare Stereo Camera	$44.99	$44.99	ACQUIRED
GPS	GPS NEO-6M	$11.59	$11.59	ACQUIRED
Motor Driver	Cytron Dual Channel 10A	$21.18	$21.18	ACQUIRED
Motors	SGMADA 12V Brushed x 2	$16.00	$0.00	ACQUIRED
Wheels	Dagu Wild Thumper Wheels 120x60mm Pair	$17.95	$0.00	ACQUIRED
Battery	Zeee 2S 7.4V 460mAh Lipo	$27.89	$27.89	ACQUIRED
LoRa Transceiver	RYLR896 LoRa (2)	$48.93	$48.93	ACQUIRED
5.8GHz Video TX/RX	TS5823 Transmitter	$23.00	$23.00	ACQUIRED
Enclosure	Zulkit Project Box IP65 x 2	$16.99	$16.99	ACQUIRED
	Total	$331.52	$198.57	

Canister				
Function	**Name**	**Price**	**Actual Cost**	**Status**
MCU	Arduino Nano	$17.59	$17.59	ACQUIRED
Pressure Sensor	MPL3115A2	$9.95	$9.95	ACQUIRED
Encoders	ENC-AMT102-V	$46.00	$46.00	ACQUIRED
Motor Drivers	A4988	$11.90	$11.90	ACQUIRED
Batteries	LiFePO4 18650	$12	$12	ACQUIRED
Battery Holder	3 Cell 18650	$1.59	$1.59	ACQUIRED
	Total	$99.03	$99.03	

- Financing comes from a sponsor - ARCA, and is being shared with 5 other payload teams (and a rocket team)
- Budget Goal: <$500

Current Progress



Project Difficulties/Challenges

- Size constraints present a challenge for PCB design
  - Modules require three voltage levels, and a significant amount of power electronics
- Supply Chain issues make sourcing parts difficult
  - Choosing DC-DC converter ICs based on availability, not on efficiency or smallest footprint
- Challenges involved in working in a large, interdisciplinary project involving multiple teams, including Mechanical and Aerospace Engineers
  - Coordination has been a challenge
- Significant mechanical engineering challenges involved in our design, with no dedicated Mech Es to solve them

Tasks Remaining

- Rover
  - CRITICAL: Finish software (MCU code + ROS node) to interface the SBC with its sensors and motor controller -> basic functionality
  - Finish state machine and autonomous navigation tuning
  - Finish prototype construction

- Finalize canister deployment strategy and vehicle integration

- Implement LoRa comms protocol for transmission between rover and GS