Kidus Abebe
Yale Exploration Rover
Chief Engineer
8/13/2024

Preliminary Design of the Electronic System

Goal

To have a preliminary control and communication system for the drives and the robotic arm. For simplicity and ease of use, the rover and the arm will use a similar electronic control system.

Literature Review

Stanford Darth Rover (2020):

- Custom power distribution board and (likely) powered by LiPo batteries.
- Easily accessible E-stop
- 2.4 GHz antenna for radio communication (allows for ranges > 1 km)
- They have a modular electronic system with Linux nodes connected through ethernet (whatever any of that even means). Regardless, it looks like these nodes are essentially Raspberry Pi boards running on Linux
- They use a PS4 controller to run this rover

Washington State Rover (2023):

- Custom printed circuit boards + COTS parts for the E&C system
- 3 12V Lithium-ion batteries wired in parallel for power, provide 30 min to 1 hour of run time
- Easily accessible E-stop
- Custom-made PCB houses a microprocessor that handles GPS, atmospheric readings, gyro signals, and GPIO signals. The PCB also contains PWM drivers and motor controllers.

- The primary control system is a Jetson Xavier AI computer, and it handles the bulk of the system processing.
- In terms of software, the robotic operating system (ROS) is used which allows for communication between the rover and the ground station (which also has another ROS device)
- Their software is written in Python, and ROS allows for the implementation of multiple Python scripts.
- For communication, they use 2.4 GHz and 900 MHz antennas and communicate through wifi.

Cornell Hyperion Rover (2024):

- The software transmits commands through a serial communication system
- Moteus R4.11 brushless motor drivers are used for control
- Custom PCBs (yet again) interface with brushed motors, sensors, radio communication, and more.

Stanford Rover (2018):

- Custom sensor and power distribution PCBs
- Easily accessible E-stop
- Long-range radio for low-bandwidth communication
- The main rover computer is the Jetson TX2 running on the ROS.
- COTS 2.4 GHz antenna for radio communication

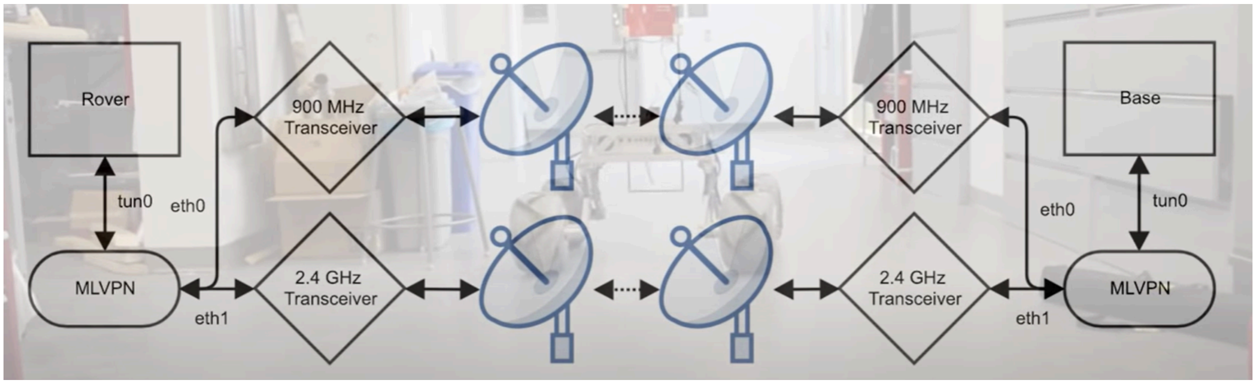
Stanford Rover (2019):

- COTS 2.4 GHz antenna for radio communication
- Each subsystem (the arm, SMS, drives, etc) uses a Linux node to communicate; the nodes are connected through ethernet.

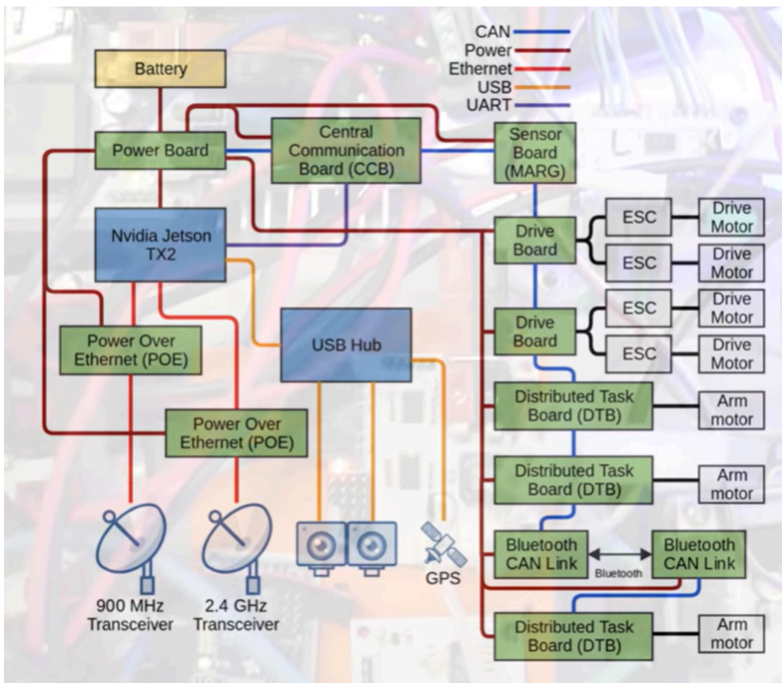
Cornell Artemis Rover (2019):

- The primary onboard computer is the Jetson TX2

- 2.4 GHz and 900 MHz transceivers with omnidirectional antennas for radio communication; capable of maintaining contact even during obstruction of the LOS



- PIC32 microcontrollers are used for the control of the motors and sensors on the rover



Michigan Rover (2020):

- 10 series lithium-ion battery pack built using 18650 cells, with a net output of 36 volts.
- Drives system uses brushless DC motors
- Radio communication with 900 MHz linearly polarized antennas
- Interestingly, the rover contains a remotely deployable radio repeater for extra-long-range communication

Requirement Definitions

Basic design requirements and constraints:

- Long-range communication (up to a mile or more)

  - Standard I/O
  - Low-latency data transmission
  - Low-latency video feed transmission
- Fully remote control of all subsystems
- Image recognition and machine learning ability (for the autonomous navigation challenge)
- High-speed internal communication (i.e. communication between inputs, outputs, and the onboard computer)
- Given the autonomous drive challenge, we would need a powerful onboard computer capable of running image recognition software and machine learning models. This implies the use of some Nvidia computer (like the Jetson used by a lot of other teams)
- Seems like an easily accessible E-stop is mandatory (or at least highly encouraged since every team seems to have one)
- Power systems will be lithium-ion batteries and all drive systems (from the wheels to the robotic arm) should be brushless DC motors.
- On the software end, we can use some combination of an open-source operating system like ROS and Python.
- We will most likely have to produce custom-made PCBs for power distribution and communication between the central computer and the various subsystems on the rover.

Conceptual Design

- Communication system
  - 1 = poor, 3 = acceptable, 5 = excellent

Antenna	Ease of Use	Performance	Literature	Cost	Total
2.4 GHz	5	5	5	3	18
900 MHz	5	3	5	3	16
Both	3	5	3	3	14

  - Based on available literature and system requirements, we will use radio communication along a 2.4 GHz transceiver system
- Primary control system: a short trade study
  - 1 = poor, 3 = acceptable, 5 = excellent

Computer	Ease of Use	Performance	Resources	Cost	Total
Nvidia	3	5	4	1	13
Raspberry Pi	5	3	5	5	18

    - Although an Nvidia processor is the best choice in terms of performance, our budget may restrict us to using a Raspberry Pi.
    - However, if cost is no object, the appropriate choice would be an Nvidia Jetson (nano, Xavier, TX2, etc).
- Required sensors
    - Atmospheric condition sensors
    - GPS
    - Gyroscopic sensors
    - LiDar for obstacle avoidance
    - Cameras (lots of them)
- Power system
    - The rover will be powered by lithium-ion batteries and the power will likely have to be distributed through a custom PCB

Next Steps: Technology Baseline

- Complete any outstanding trade studies for the concept design
    - Communication bandwidth
    - Central control computer
    - Software (Python, C, ROS, Linux, etc)
- Make a PDR and present it to the team and to advisors
- Design the preliminary system architecture
    - Begin designing the custom PCBs for control, power distribution, communication, etc.
    - Begin writing the program for the control system (Python, C, or Linux)