Design Revision

CEN3907C: Senior Design for Computer Engineering

Red Horizon Team
Kayleigh Beron, Andrea Chacon, Morgan Cobb, Kayla Dunnuck, Quintin Lopez-Scarim
Machine Intelligence Lab - University of Florida

Table of Contents

Introduction

The Red Horizon project aims to develop a prototype of a Mars rover, integrating advanced features to enhance its autonomy and functionality.

Purpose/Need

The purpose of the Red Horizon project lies in providing an educational and research platform for exploring the capabilities of machine intelligence in autonomous robotic systems. This replica serves as a valuable tool for students, researchers, and educators at the Machine Intelligence Lab to study and experiment with autonomous navigation, object detection, and sensor integration. By simulating the functionalities of a Mars rover, the project fulfills the need for practical hands-on experience in developing and deploying intelligent robotic systems.

Red Horizon will be used by the Machine Intelligence Lab to show off the kinds of autonomous projects students of all ages could work on if they choose to pursue STEM and join MIL. However, there are many possible applications as well, as its mechanical chassis design and motor system allow for movement across all kinds of terrains. It's strong ability to go over things in all kinds of terrain, make it possible to be used in emergency situations, where it could use the camera to detect victims in areas of natural disasters or cramped spaces where it would be too dangerous to send a person without confirmation of victims.

Domain & Prior Art

The Red Horizon project operates within the field of autonomous robotics and control systems. While there have been significant achievements in this domain, including successful Mars rover missions conducted by various countries, our project aims to recreate this to not only help Machine Intelligence Lab bring a stronger attraction to STEM education with something of great historical significance, but to create something that could be applied in emergencies.

Some prior art would be the original Mars Rover that we would be imitating from, as the wheels and mechanical system are nearly identical. As for educational prior art, Makeblock creates "mBots" that are robots that can easily be built and programmed by young teens, to get introduced to the basics of robotics and allow young engineers to both learn and play with the robot they've created. The FIRST Competition is also a robotics competition aimed at middle schoolers and high schoolers of different levels to create a robot with autonomous and telemetry modes with computer vision, that allow young kids to learn about the basics of robotics and gain interest.

Differentiation

As for how Red Horizon will be different, Red Horizon will be more complex than those educational tools like FIRST and mBots, as it is a project that wouldn't be built by the young kids it is aiming to inspire as it would be too difficult for their level. It is a chance for those kids who are working in FIRST Robotics, to see a more complex project and the real world applications of what they could be working on. This is a project we could bring to local schools with FIRST robotics teams and allow them to play with it while seeing how they can apply robotics outside of a high-school competition.

Impact & Risk Assessment

The impact of the Red Horizon project extends primarily to the educational and research domains within the university community. By providing a hands-on learning experience, the project enhances students' understanding of machine intelligence, robotics, and computer engineering.
The project also raises ethical and privacy concerns, particularly regarding the use of onboard cameras for object detection and navigation. The potential for unintended surveillance or privacy breaches requires careful consideration and implementation of privacy safeguards. As for how we will address this, we will ensure that the robot cannot leave MIL room by trying to make a docking station for it, and ensure the camera is at an angle it can only see directly in-front from a low height. The controllers for it will be kept by Dr. Schwartz as well.

Drawbacks and Limitations

While creating a Mars rover replica like Red Horizon for educational purposes is exciting, it comes with several drawbacks and limitations. Firstly, reliance on computer vision for navigation may be prone to errors in identifying colors and shapes accurately, especially in challenging environments. Additionally, the rover's autonomy may lead to unpredictable behavior, potentially causing collisions or getting stuck in obstacles. Moreover, replicating the complex terrain and environmental conditions of Mars accurately can be difficult, limiting the realism and educational value of the experience. Overall however, these limitations shouldn't be too large to affect the educational experience.

Statement of Work

By the pre-alpha build (03/27) the Red Horizon team plans to have drivers for the Motors and the controller, along with our high level ROS (Robot Operating System) environment. For the drivers we plan to use C++,C, and potentially some assembly for speed and efficiency. C++ and C will be the default for most drivers, and assembly will be used for some reach goal by utilizing some cutting edge communication protocols. For the specific communication protocols we plan to use for each peripheral please see either the tables below, or the diagram included in the *systems* section. We will use Cmake as our build system since it meshes well with C++, C, assembly, and ROS. The motor board and kill board will be completed and most parts should be ordered and will continue until the end of the alpha build. We also hope to order sonar kits, and to have a better foundational understanding of the current mechanical state of the Red Horizon.

By the completion of our prototype (04/16) development, we hope to have drivers for the gyroscope, camera, and interfacing with our external computer. Once again, we will be utilizing C++,C, and assembly for that, along with Cmake as our build system. The voltage divider board and battery monitor board will be completed and tested to ensure functionality. Some of our secondary features that we hope to get done include drivers for an accelerometer, and patching up any mechanical issues.

As for our alpha build/demo (08/27) we plan to have attached sonar with drivers in C and C++ using Cmake, and hope to utilize daisy-chaining to increase their accuracy. With those sonar attachments come more information that we can use to increase autonomy, which we hope to have at a sufficient level to roam on its own at this point without destroying itself. . Bluetooth and wifi will also be connected to the controllers to the kill board and raspberry pi. LCD output will be created and tested to ensure data from the screen outputs all current data correctly.

Pre-Alpha build

Core features	Done by:	Secondary features	Timeline (Starting March 11th)
Controller Drivers (wifi)	Quintin and Andrea	Order sonar kits/start researching how to daisy chain them	Week 2
Motor/DAC Drivers	Quintin and Andrea		Week 3
Configure ROS	Morgan		Week 1-2
Begin teleop code	Morgan and Kayla		Week 2

Determine the current mechanical state of Red Horizon	Quintin		Week 1
Motor and Kill Board	Kayleigh and Kayla		Weeks 1-3
Initialize Raspberry Pi	Andrea		Weeks 1-3

Prototype

Core features	Done by:	Secondary features	Timeline (Week 4 starting 04/01/2024)
Motors attached with minimal functionality	Everyone	Accelerometer drivers	Weeks 4-6
Gyroscope drivers (I2C)	Quintin	Mechanical fixes	Week 4-5
UI for data/connectivity with computer	Morgan and Kayla		Week 4
Begin high level data analysis (basic autonomy)	Morgan and Kayla		Week 5-6
Add camera with base drivers (USB)	Andrea		Week 6
Configure LCD to Raspberry Pi (SPI)	Andrea		Week 4-5
Configure remote controller (WIFI)	Andrea and Kayleigh		Week 5-6
Voltage Divider and Battery Monitor Board	Kayleigh and Kayla		Week 4-6

Alpha build/demo

Core features	Done by:	Secondary features	Timeline (Continuing on starting 08/12/2024 with some work done over summer)
Attach functioning sonar	Quintin and Andrea	Path mapping with an accelerometer	Week 7-8
Basic autonomy through high level interactions finished	Quintin and Andrea	Mechanical fixes	Week 9-10
Finish part ordering	Kayleigh		Week 7
Start testing and integration	Everyone		Week 7-10

Deliverable Artifacts

We intend to deliver the software for the control, as well as drivers/libraries created for the firmware in a github repository. Documentation in the form of jupyter notebooks for these drivers will be provided. Additionally, we will deliver the PCB's used for the kill board, voltage divider board, motors board, and battery monitor board with their schematics. The final product will be all of these features combined onto the working rover, and delivered to the faculty supervisor, to be used for future research lab demos.

As for how this will be maintained for long-term use there are several plans. The robot will stay in the Machine Intelligence Lab, and the controllers will be kept by Dr. Schwartz. We will provide the CUI to Dr. Schwartz, and keep it on one of the computers in the MIL lab. On one of the computers in the MIL lab, students should just receive the controllers by Dr. Schwartz, run the .exe file on the MIL computer to open the CUI, and once the controllers are connected and the CUI is booted establishing connection to the rover, students should be free to use the rover around the room as pleased. This software will also be available on one of the MIL laptops that can be used at outreach events with local schools for educational purposes, and documentation will be provided on how to use and how to fix some possible issues. Lastly, we will provide a final paper that describes how it works that can be studied by any MIL member.

Mockups

Interfaces

The main interface of this project would be the Bluetooth Logitech Controllers and the LCD Screen. The main inputs from humans come from the kill button (a large physical button on the exterior of the robot to emergency kill), as well as the bluetooth controllers that would change the rover from autonomous to teleop mode (driving by human control) by moving a joystick, and reset it based on a button. We could possibly also try a phone app, not sure if we will implement that yet. The LCD screen would display the status of everything (kill/not kill mode, teleop or autonomous mode, temperature, humidity, altitude, orientation, and if detecting anything on the sensors, all based on the different sensors).
Below is a possible LCD Screen output:

Figure 1: Potential LCD Screen output

Below is a possible remote controller mapping. The soil test is specifically for activating the soil test state that would push a servo stick into the dirt, and output the results from it into the LCD display.

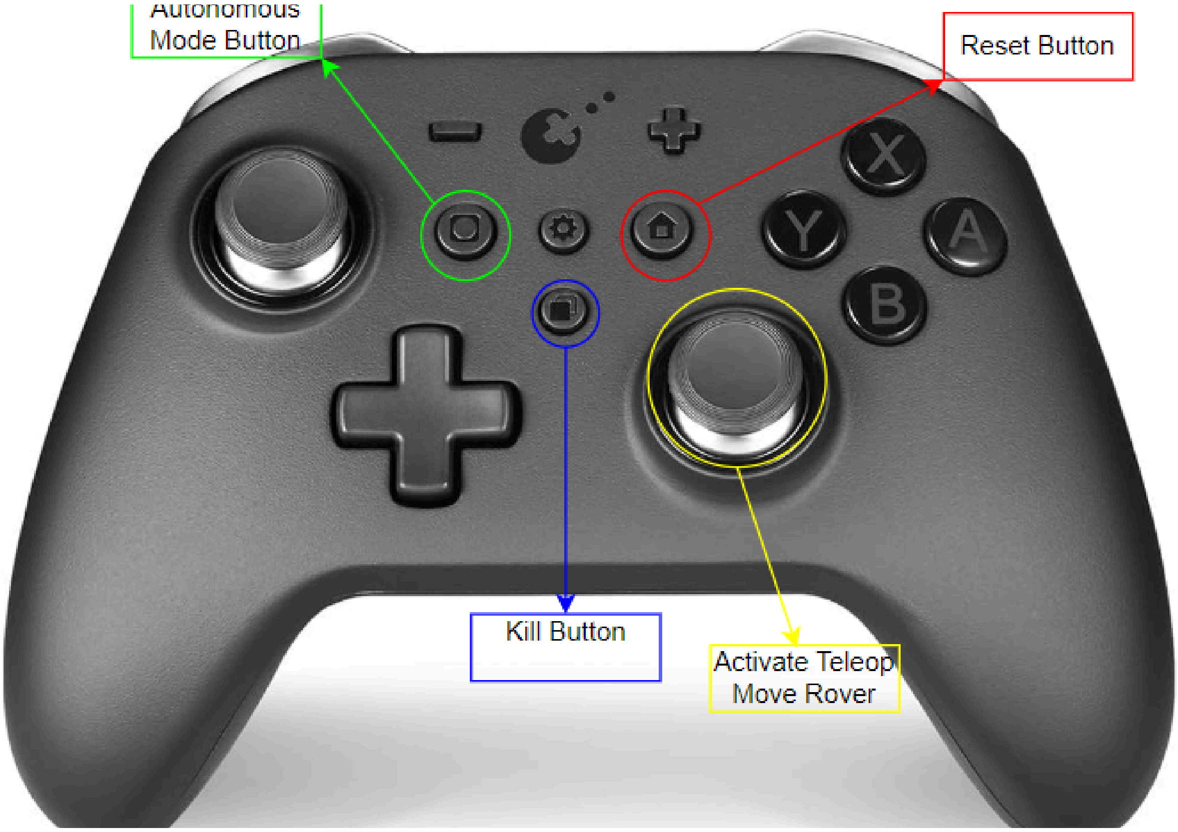
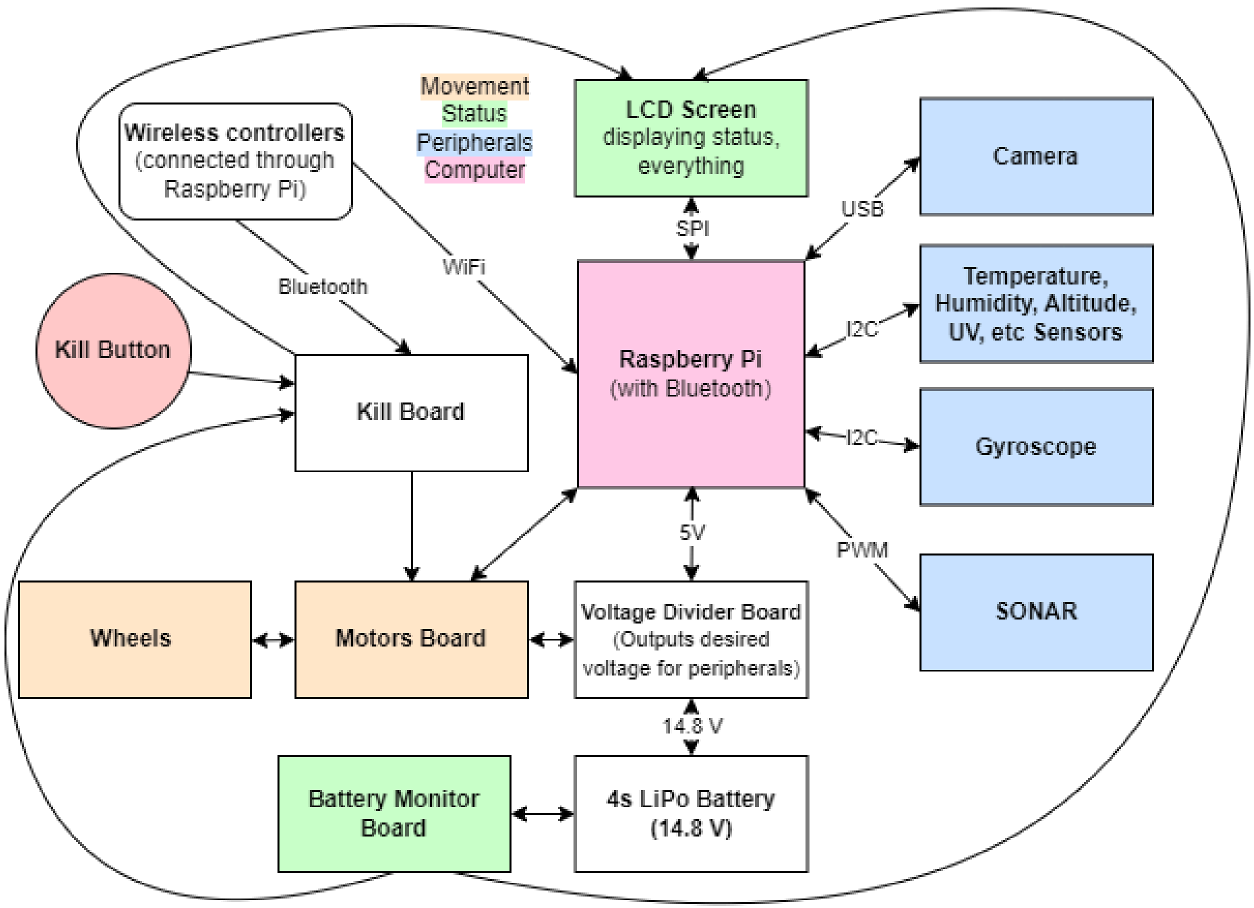


Figure 2: Remote Controller Map

Systems



Above is the general hardware system diagram for our project. Below is the block diagram for the high-level software (computer vision) algorithm. Our main computer will be a Raspberry Pi with Bluetooth functionality that would connect via Bluetooth to the Kill Board (kills all motors/servos when necessary).

We would have a motor system with different motors for each wheel, (possibly mecanum wheels), an arm system involving a servo that just sticks a soil quality sensor into the dirt, and output displayed on the LCD. We would also have a camera connected to the computer searching for shapes and colors (e.g: red circle detected in the distance). Lastly there should be a status system for killed, unkilled, autonomous, teleop, etc.

Networking

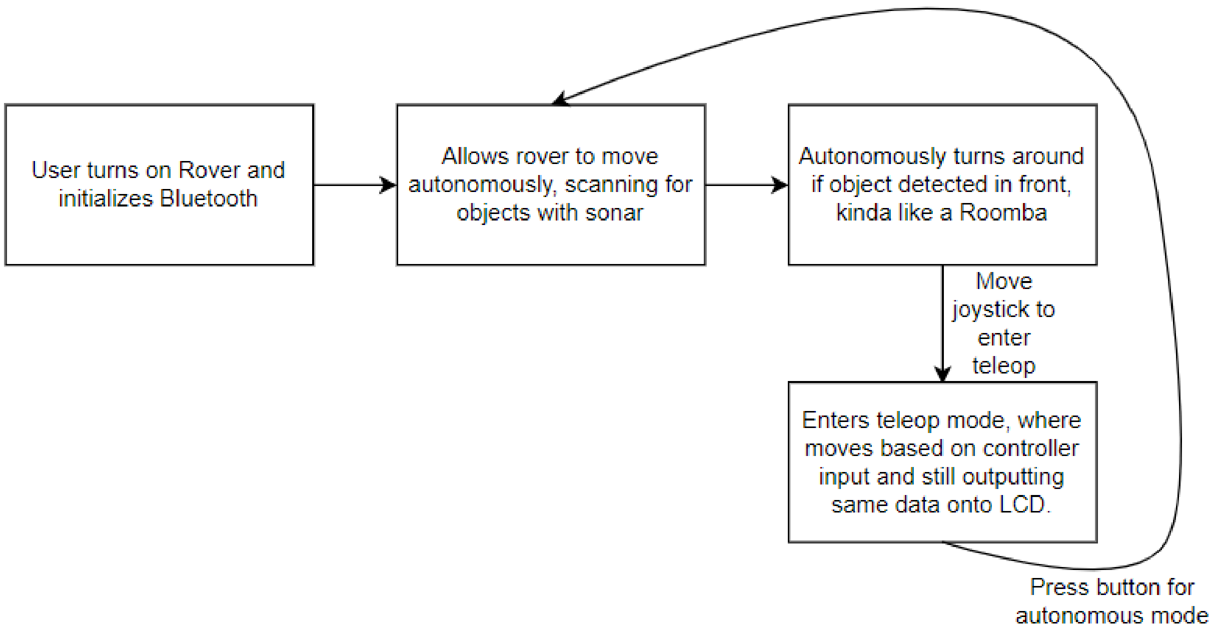
There are several necessary communication protocols to fully implement our rover. We would use Serial Peripheral Interface (SPI) communication from our Pi to the screen, to be able to display our sensor and critical information quickly. Additionally, the sensors will use Inter-Integrated Circuits (I2C) or SPI - depending on how critical the data transfers are, as well as the physical location of the hardware components. Additionally, we will use Pulse Width Modulation (PWM) to communicate with the sonar sensors, encoding analog signals into digital signals.

In order to communicate with our Pi from our controller, we will use a CUI that will send instructions to the Pi via Wifi for high data transfer, wide coverage range, and flexibility. We will also use bluetooth or wifi to communicate with the kill board.

Storyboards

The LCD, which is the biggest output, will just look like this most of the time, with the values not in bold, changing every millisecond.

Storyboard example:



Draft Schematics