Battlefield Effects Simulator Robot

UCF Senior Design Project

28 November 2022

Julia Kemper
Computer Engineering

Nicholas Nachowicz
Computer Engineering

Jared Rymkos
Computer Engineering

Michael Rodriguez
Electrical Engineering

PEO STRI

UCF
UNIVERSITY OF CENTRAL FLORIDA

Sponsors/Mentors:

James Todd

Jeremy Lanman

David Howard

Tom Waligora

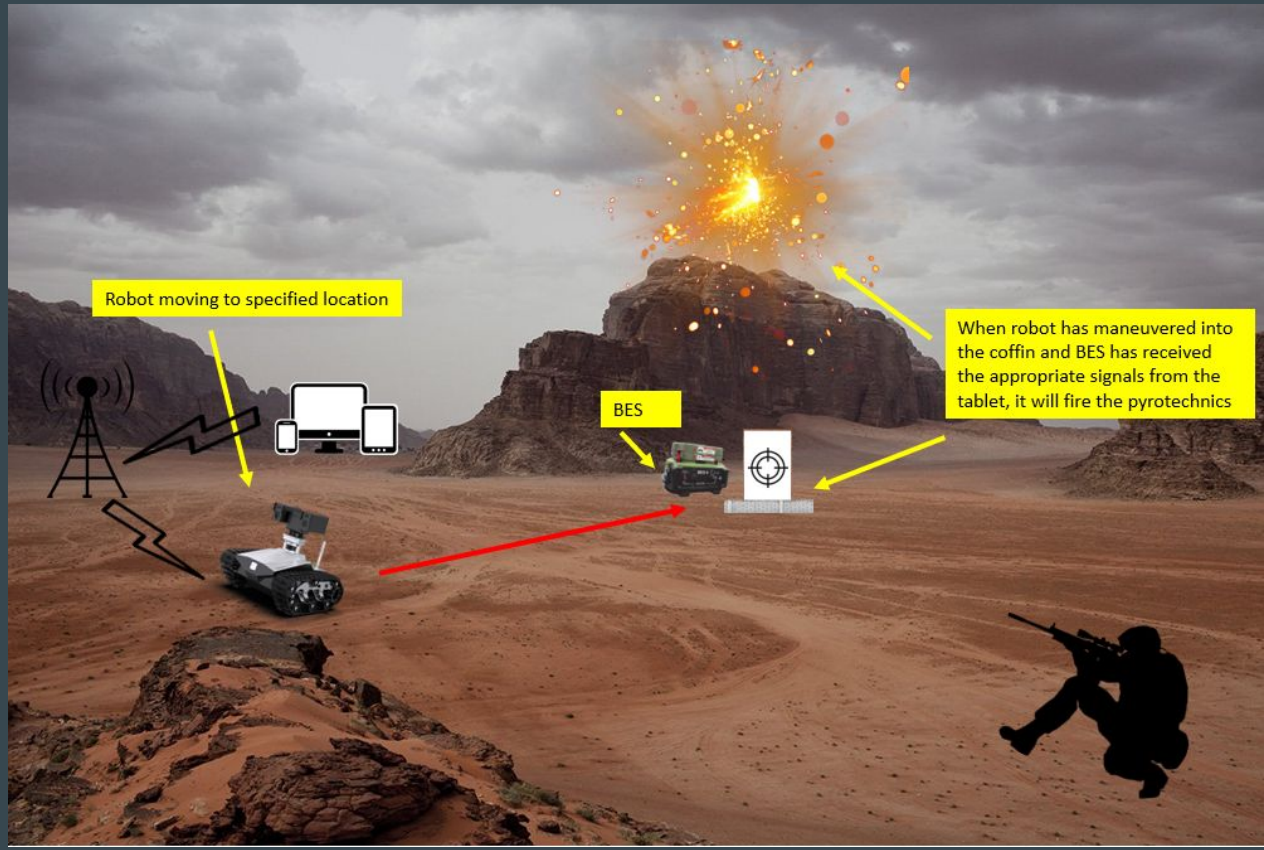
Project Overview

The Battlefield Effects Simulator robot is a small scale version of a robotic system that will be utilized on live fire ranges to support training realism and feedback, threat representation, and pyrotechnical handling.

This version of the robot will maneuver through rough terrain to and from desired locations to accurately fire pyrotechnics when proper safety signals are sent. The BES for this prototype will not be on top of the robot, but will still be in communication with it. The robot will interact with the BES to send the safety conditions for when it is clear to fire the pyrotechnics.

The safety conditions include:

1. The robot has arrived at the correct location on the map
2. The robot is on a flat surface plus or minus ten degrees



Takeaway: This product will provide a rapid prototype and technical data package that can be leveraged by the Army to: **increase training throughput, improve safety,** and **reduce cost** of live fire training

Goals and Objectives

Main goal: For this robotic system to be utilized on live fire ranges to support training realism and feedback, threat representation, and pyrotechnical handling.

Our objectives consist of the following:

- The robot will use lidar data to create a map of its environment
- The robot will correctly navigate to its specified destination
- The robot will detect obstacles in its path and navigate around them
- The robot will ensure it is level +- 10 degrees when it has arrived at its specified destination
- The robot will ensure it has arrived at the correct destination
- The robot will send signal to GUI via ESP32 that it is safe to fire
- GUI will send signal to BES to fire pyrotechnics

Project Accomplishments

General Software/Hardware:

- Creation of custom printed circuit board
- Ubuntu integration with Raspberry Pi
- ROS integration with Ubuntu/Raspberry Pi
- Communication between ROS, Raspberry Pi and Arduino
- Creation of GUI that tracks robots location and

Sensor Data:

- Lidar integration with Raspberry Pi
- IMU integration with Raspberry Pi
- Output of lidar data in terminal
- Visualization of lidar data through ROS RVIZ application
- SLAM (Simultaneous Localization and Mapping) integration
    - Creating a map of robots environment using lidar
    - Creating costmap configurations to detect obstacles in robot's environment
    - Publishing IMU data and visualizing it to RVIZ application

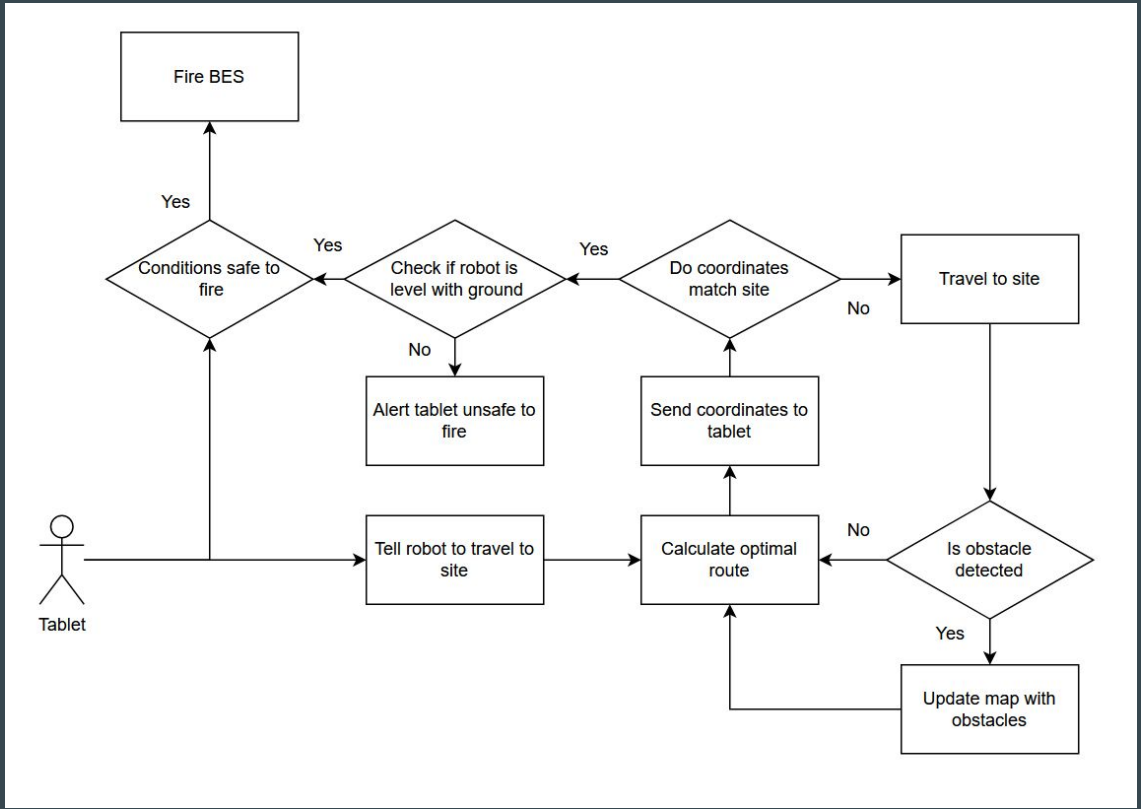
Odometry Data:

- Publishing robot velocity commands through Arduino code to motor controllers
- Remotely controlling robot velocity through ROS steering GUI
- SLAM (Simultaneous Localization and Mapping) integration
    - Creating initial and goal pose publishiner
    - Visualizing robot moving in a mapped environment
    - Autonomous navigation of robot
- Motor encoder, motor controller, and arduino integration

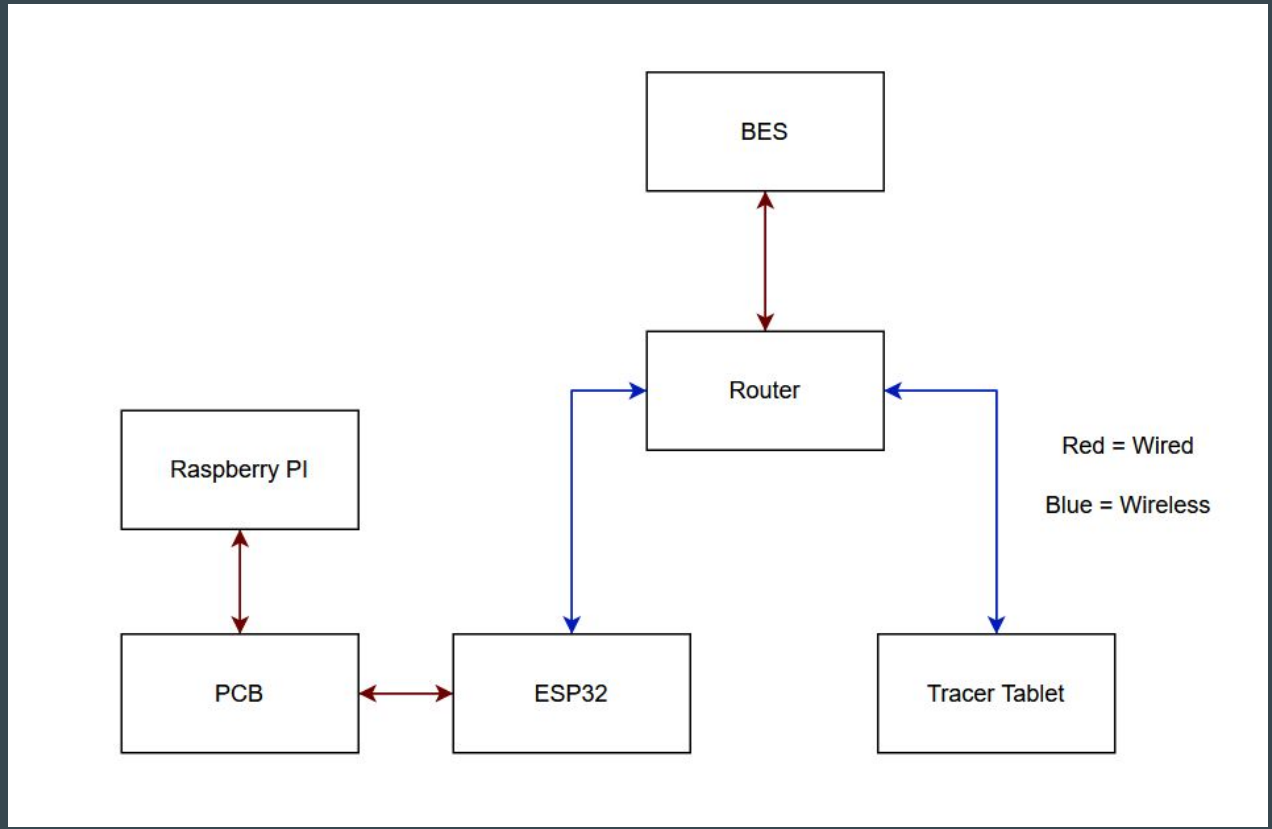
Communication:

- Windows tablet runs GUI and acts as client while ESP32 acts as server
- Tablet and ESP32 are on the same network and communicate through sockets
- BES communicates through wifi with ESP32
- Integration with existing product line software

Overall Movement Flow Chart



Communication Diagram



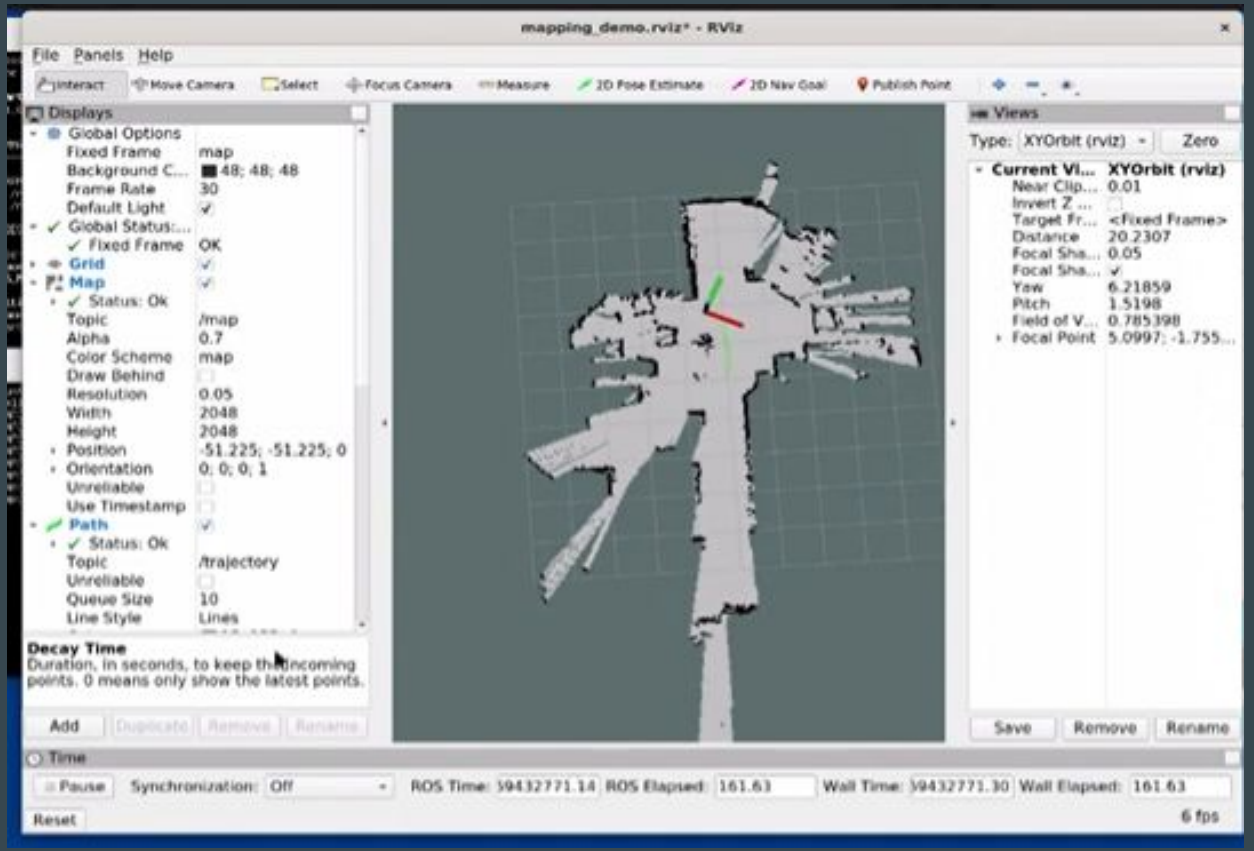
Software Design: ROS Navigation Stack

Environment Mapping, Robot Navigation, and **Object Detection** was performed via **Robot Operating System (ROS)**

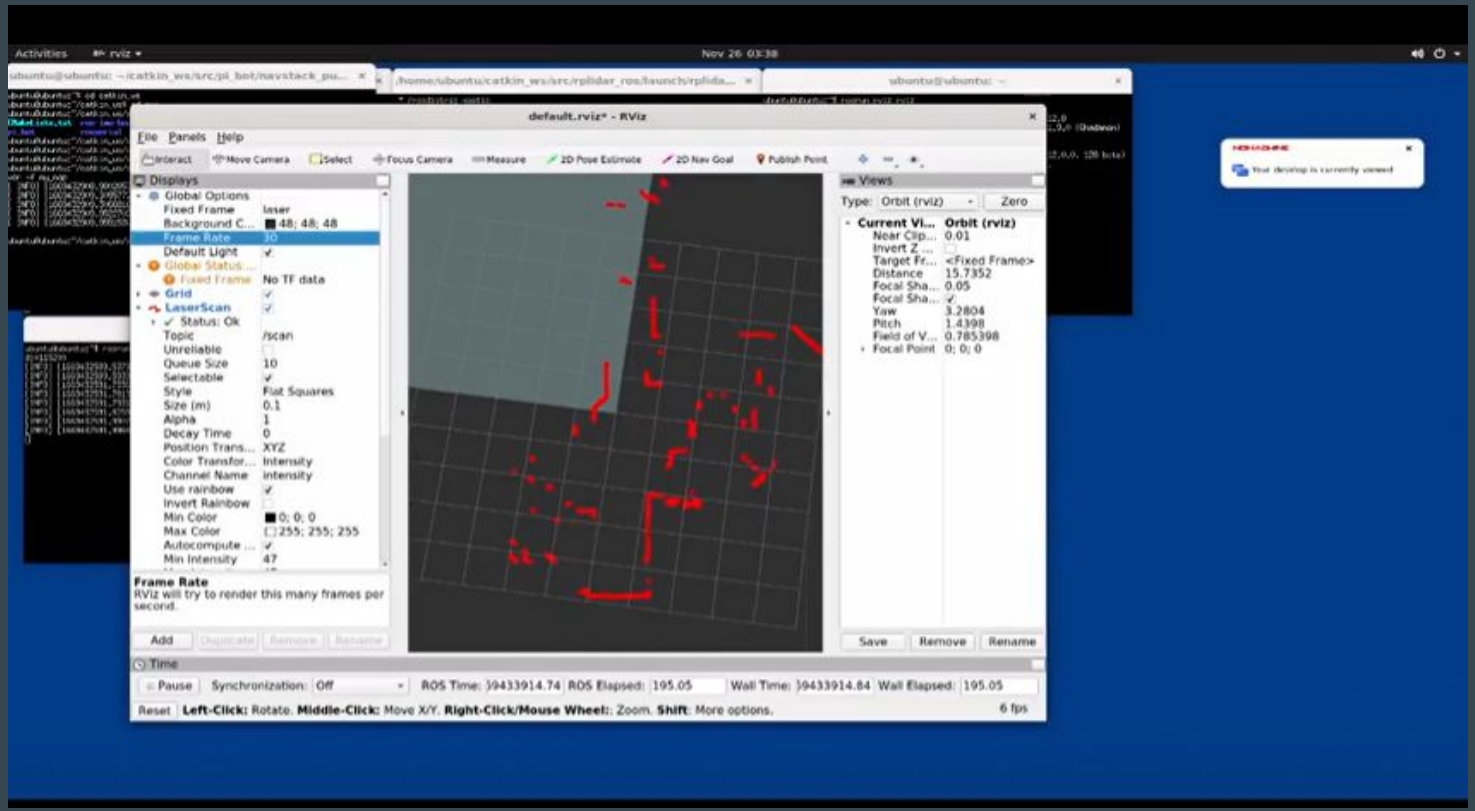
- Robot Operating System (ROS) Noetic
- ROS RPLIDAR package
  - Integration of lidar, mapping and object detection
  - Similar to projects at STRI that use lidar to map environments such as OWT
- ROS Hector SLAM (Simultaneous localization and mapping) package
  - Mapping and navigation
- RVIZ application - display mapping data
- Arduino code written in C++ is uploaded to PCB and sends velocity commands to Raspberry Pi and motor controllers

Software Design: Environment Mapping

- Map created in RVIZ by robot traveling around environment utilizing lidar data and ROS SLAM package



Software Design: Robot Navigation and Object Detection



GUI Demo

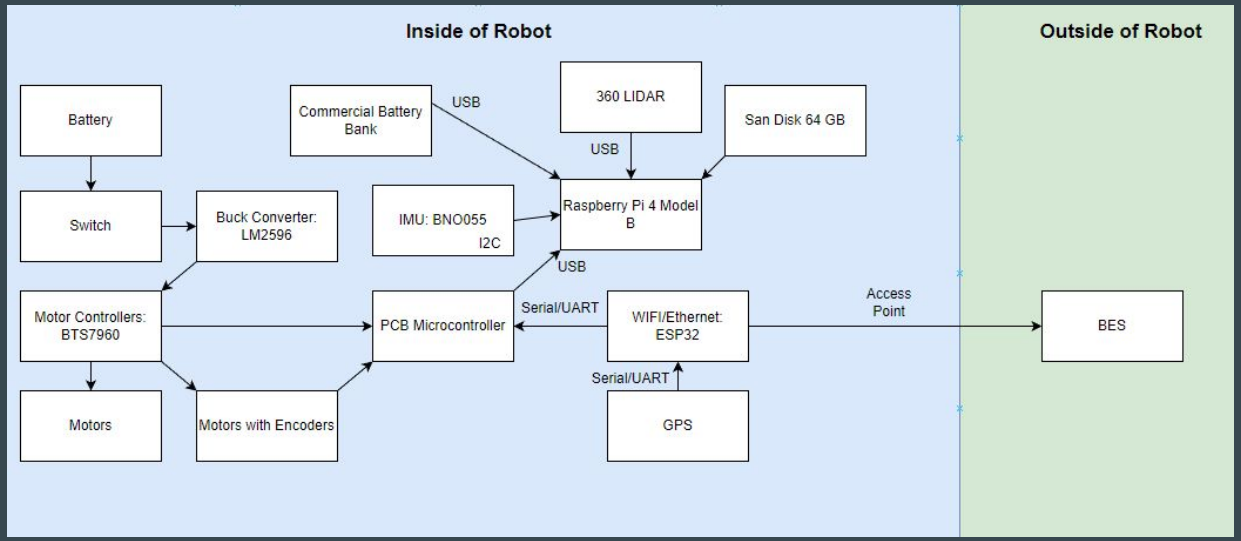
Path Forward

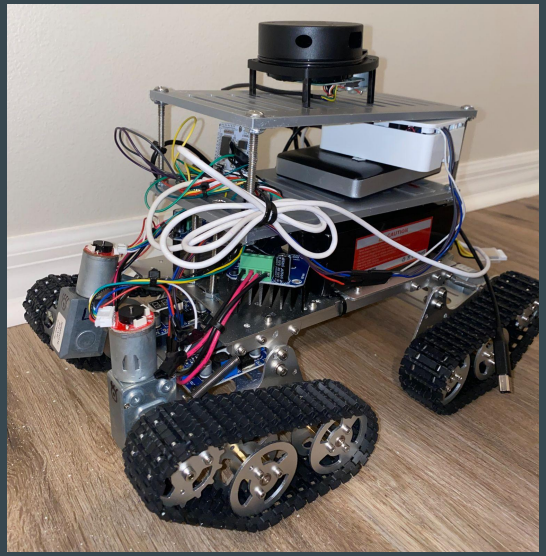
PEO STRI has expressed interest in continuing to sponsor UCF senior design teams to continue and evolve this concept/project over multiple phases.

- Continue to improve accuracy of autonomous navigation
- Improve obstacle detection/clearing of obstacles that are no longer in robot path
- Camera integration to detect humans vs moving objects
- Build larger scale prototype

Backup

Overall Design

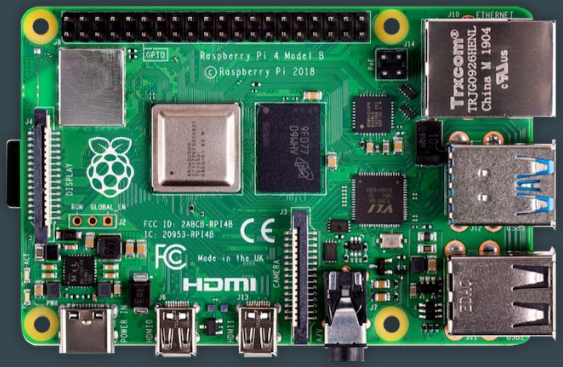




Raspberry Pi 4B Model

- Best performance specs compared to Jetson Nano, Odroid XU4, and Jetson Xaiver
- Lots of documentation regarding Raspberry Pi 4B and ROS integration
- Affordable and available for purchase by reliable vendor

- Ran Ubuntu Desktop
- Used to power LiDAR and process LiDAR data
- Used to power IMU and process IMU data
- Used to power PCB
- Acts as a virtual machine to run all ROS programs



LiDAR - RPLIDAR A1

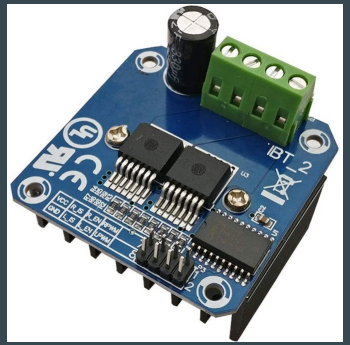
- Used to create map environment
- Used to detect obstacles in environment
- Used to accurately navigate through mapped environment
- Similar to projects at STRI that map environments such as OWT
- Used RPLiDAR Package created by ROS

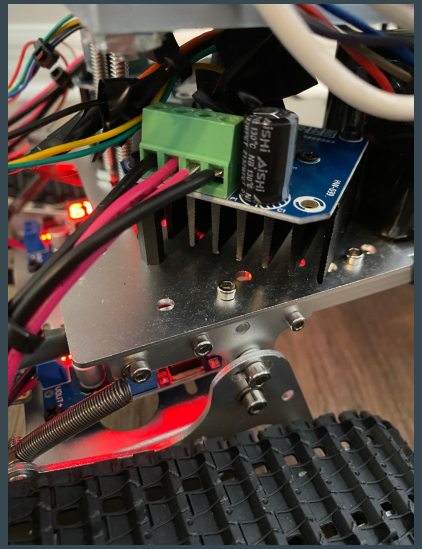
- 360 degree 2D laser scan
- 12 meter laser scan distance
- 5.5 Hz scan rate
- 8000 per time samples

BTS7960 Motor Driver

- Used to power and control the motors
- Reliable for powering both low voltage and high voltage motors
- Low cost and available from a reputable vendor

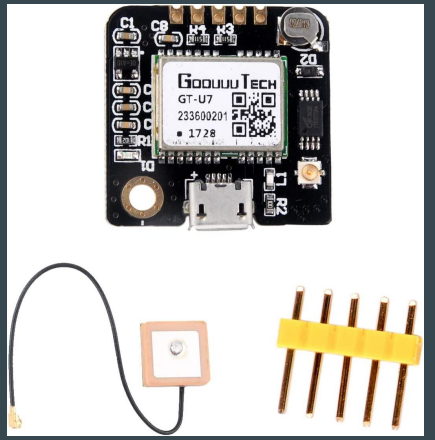
- Capable of powering motors with 6-27 V
- Has a max output of 43 A for motors
- Only needs 5 V to operate logic side





GPS - NEO 6

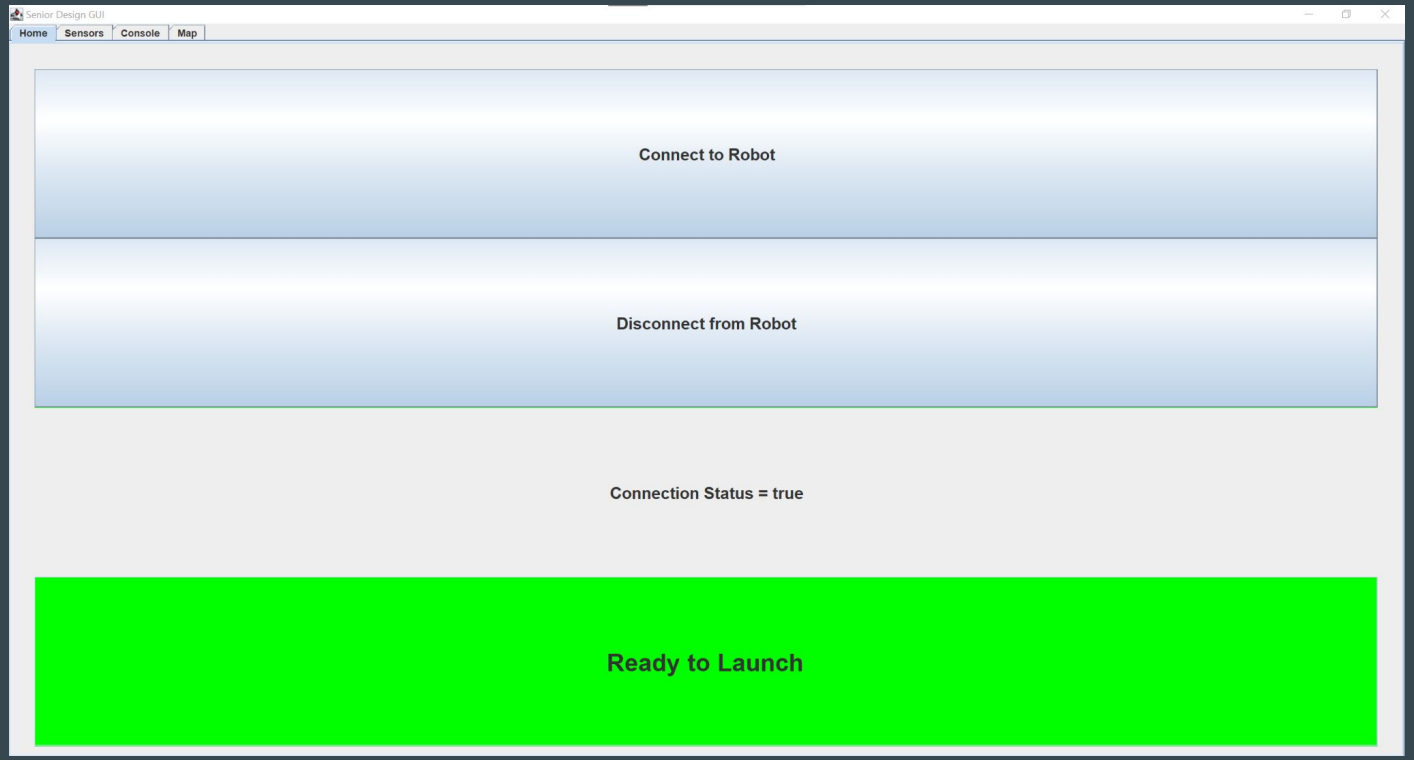
- Used to receive longitude and latitude from satellite
- Needs time to pick up a satellite before its accurate
- Coordinates are transferred to GUI to be displayed
- Connected to ESP32 through UART using GPIO pins



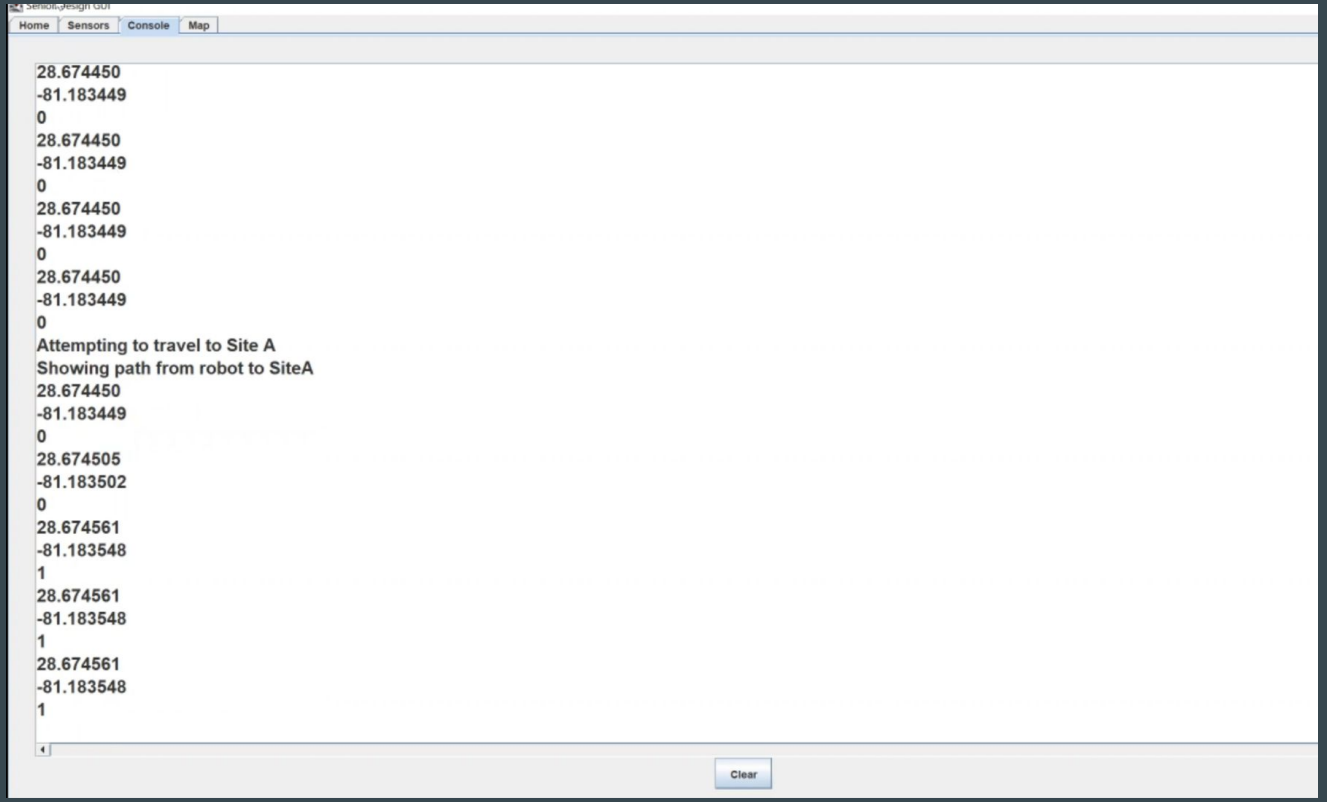
ESP32

- Acts as server for the client (tablet) to connect too when on same network
- GPS coordinates and launch status are sent every 2 seconds to tablet
- Receives a byte (‘A’, ‘B’, ‘C’, or ‘0’) signalling which site to travel too
- Powered by Arduino and code is written in C++
- Link to the code: https://github.com/jrymkos/ESP32_WIFI

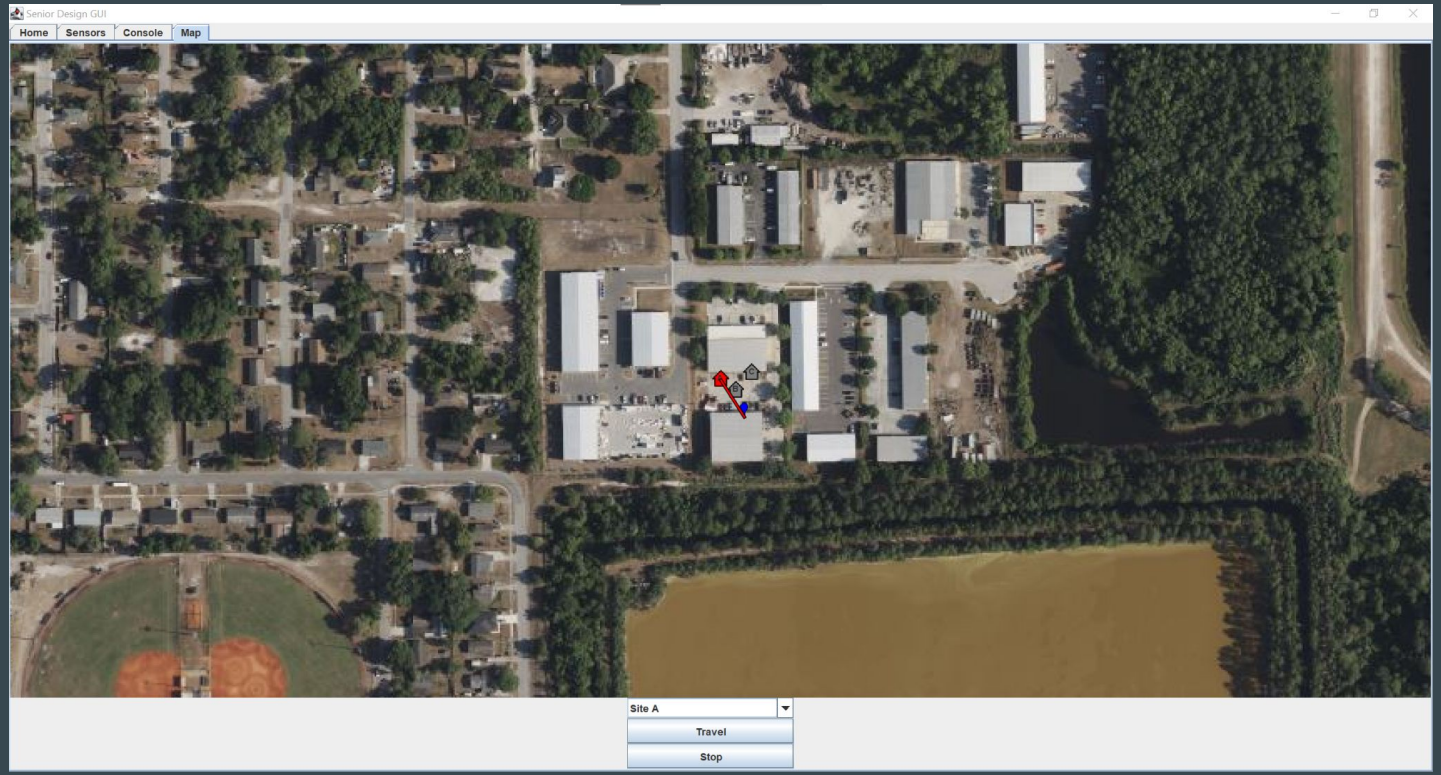
GUI


Senior Design GUI
Home
Sensors
Console
Map
Connect to Robot
Disconnect from Robot
Connection Status = true
Ready to Launch

GUI


Home
Sensors
Console
Map
28.674450
-81.183449
0
28.674450
-81.183449
0
28.674450
-81.183449
0
28.674450
-81.183449
0
Attempting to travel to Site A
Showing path from robot to SiteA
28.674450
-81.183449
0
28.674505
-81.183502
0
28.674561
-81.183548
1
28.674561
-81.183548
1
28.674561
-81.183548
1
Clear

GUI


Senior Design GUI
Home
Sensors
Console
Map
Site A
Travel
Stop