NASA Challenge Team

High Level Design

Ben Bernhard, Riley Egan, Brendan Kopf, George O’Sullivan, & Cara Ravasio

Table of Contents:

1 Introduction

Our project is loosely based off of the NASA Centennial Challenges Program (CCP) Space Robotics Phase 2 (SRC2). Though the original challenge is for virtual simulations of robotic systems, we will be building physical rovers. Essentially, our modified version of this project involves designing and constructing a system of two autonomous rovers capable of navigating through a partially unknown lunar environment, finding a predetermined object, and estimating the position of the object relative to the rovers’ starting location.

2 Problem Statement and Proposed Solution

As humans begin to spend more time in harsh, extraterrestrial environments, new technologies for applications often reserved for Earthbound missions must emerge. One such mission is the search and retrieval of some known object by a team of robots. We will focus on only the first segment of this mission: the search. This involves identifying the location of the object of interest and reporting the location back to Earth. Moreover, we focus specifically on the electrical engineering aspects of the mission and only require a trivial rover mechanical design. This simplified mission is largely directed by a couple of main design constraints.

Firstly, communication latencies between Earth and the rovers and the limited bandwidth that exist in such a channel dictate that the robotic system be able to function completely autonomously for long periods of time. While long distance communication between the Earth and Moon is outside the scope and budget of this project, we still require that the rovers be autonomous.

Additionally, the location of the object will likely be in a dimly lit or completely dark area (e.g. in a crater or on the dark side of the moon). So, the rovers must be able to function in these lighting conditions.

Furthermore, we consider a time sensitive mission, in which the object must be located as quickly as possible.

Lastly, NASA pays special attention to the radiation and thermal environments in which the device must operate. Both of these environmental hurdles will affect the processing and sensing capabilities of the solution. While thermal concerns are well within our scope, we do not have the resources to properly address radiation concerns, so we simplify this constraint by only requiring simple shielding.

We propose a two rover system to accomplish the mission outlined in the problem description. These two rovers work together to find and report the location of the object of interest in the following way.

First, each rover searches a different area of the moon for the target object while remaining within communication range of one another. Each rover navigates the environment, keeping track of its location relative to its starting location and avoiding obstacles detectable by a range finder.

Once the rover detects the object using object recognition technology, the rover reports the picture to Earth or a nearby lunar gateway for confirmation. Once the object in the image is confirmed to be the object of interest, the rover estimates the object's position using the rangefinder data and motor encoder data and transmits this location to the other rover.

When the other rover receives this signal, it travels to the communicated location, finds the object of interest, and makes its own estimate of the object's location. Using these two independent estimates of the object's location, a final estimate will be made and recorded. This concludes the mission.

An overview of a given rover in the system is illustrated in Figure 1.

Usually, lunar missions would require the rovers to have a long term power supply (generally nuclear). However, we consider the question of a reliable long term power source outside the scope of our project since our focus is on object identification, autonomy, and communication.

We will address radiation and temperature concerns by operating our rovers only on the dark side of the moon. Since the rovers are already expected to operate in dark environments, this is a reasonable simplification. The dark side of the moon will shield us from the moon's higher temperatures and most of the strongest radiation from the sun. With only low temperatures and ambient radiation to worry about, we can reasonably use simple radiation shielding and escape the need for electronics working at high temperatures.

We will demonstrate basic swarm communication by creating two robots to communicate with each other. In theory if two robots can work together, a larger swarm could communicate in the same general way. We will be using Zigbee for our communication protocol.

To successfully identify and locate the prespecified object of interest, the rovers will patrol their paths and use a camera and flash light to identify obstacles and potential objects. The distance between the rover and obstacle or object will be determined by implementing a laser rangefinder (see figure 1). If a potential candidate for the object of interest is found, a picture will be taken for identification purposes. A convolutional neural net will be taught to identify the specific object of interest; we plan to use OpenCV for this step and train it using simulated pictures. If the target is determined to be the object, then it’s location will be recorded and communicated to the other rover.

While the rover is travelling through the partially known terrain, there may be some objects in the rovers path that the rover needs to avoid. We plan on using the laser rangefinder as the obstacle detection for the rover. The laser rangefinder will be directed at obstacles in the rover’s path so the rover knows the location of the obstacles and can take the proper measures to avoid unknown obstacles.

We plan on having the rovers cover a partially known terrain which means the rover can have a predetermined path programmed in. The rovers will be deployed from a known starting location. The partially known environment will have the basic topology of the moon so we can avoid mountains and craters for our rover paths. We will have the rover traverse this terrain while avoiding any obstacles that may come up and then moving back to the programmed path.

3 System Requirements

This system will be built around a PIC32 processor, which will manage communications between rovers, coordinate movement, and interface with a laser rangefinder. In addition to this processor, we intend to use a Raspberry Pi running ROS to handle path planning, as well as to host computer vision computation and location acquisition. We will have to take advantage of SPI, I2C, and UART interfaces on the PIC32 in addition to SPI, UART, I2C, and CSI interfaces on the Pi.

The system will be powered using multi-cell LiPo rechargeable batteries, which we hope will supply power for up to three hours. This is more than enough time to find an object within our demo search area and will allow us to scale up the search area as we see fit.

The system will have multiple wireless interfaces - one for self-location (DecaWave UWB) and one for inter-rover communications (Zigbee). We must first ensure that these do not interfere with each other. The Zigbee module we intend to use provides a maximum range of 400 ft. This is plenty, because a mesh network allows us to relay messages throughout the swarm. To conserve power, we would like to operate at a maximum range of around 200 ft.

This system is intended to be autonomous, so the UI will be limited. We would like to incorporate a configuration button that kicks off auto-configuration and then starts the object search protocol. This configuration process must be able to be carried out independently on each rover so we can dynamically add nodes to the swarm.

This system will be deployed as follows: the anchor points will be placed in known positions and configured. The rovers will then be placed within the search field in known positions. The configuration button on each rover will be depressed and the two bots will establish their respective communication links and begin the search process.

We intend to use two servo motors with a maximum torque of 2.5 kg-cm and wheels with a 3.3 cm radius, leading us to a maximum total rover weight of 3.4 pounds. Thus, we would like to limit the weight of each rover to 3 pounds to maintain a factor of safety of 0.4 pounds.

4 System Block Diagram

4.1 Overall System:

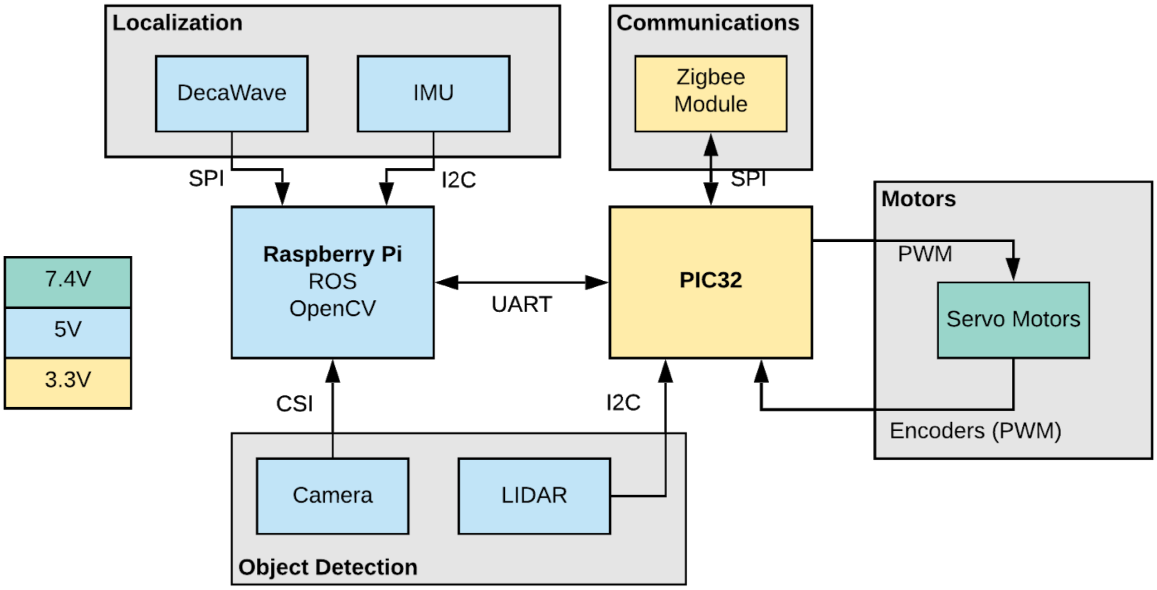

Figure 1. Proposed electrical system overview of a single rover in a two-rover system

4.2 Subsystem and Interface Requirements:

4.2.1 Localization:

The location subsystem will consist of a collection of DecaWave modules and an inertial measurement unit (IMU). We will have four DecaWave DWM1001 modules at fixed anchor points as well as one module per mobile rover. At the anchor points, no interface is required, and we will provide power via micro USB. On each rover, we will read location information from the DecaWave with a Raspberry Pi via an SPI interface. The location calculations are handled onboard the DWM1001. We will execute information reads using Python and the SPIDEV library. We will interface with the IMU via I2C to get information needed to determine rover orientation. Ultimately, this will be used for path planning within ROS.

4.2.2 Object Detection:

The object detection subsystem will rely on both a camera and a laser rangefinder. We will interface the camera with a Raspberry Pi programmed using the OpenCV library.

This connection will be made via the CSI (camera serial interface) onboard the Pi. When the camera detects the target object, we will send a message from the Pi to the PIC32 to begin polling the laser rangefinder sensor. The laser rangefinder will be connected to the PIC32 via an I2C interface. When a measurement is received, we will have to initiate an interrupt to begin communicating this information to the other rover.

4.2.3 Local Communication:

The local rover-to-rover communication subsystem will be built on the Zigbee mesh networking protocol. We will interface a Microchip Zigbee module with the PIC32 via an SPI interface. This will require a full Zigbee stack running on the PIC32.

4.2.4 Motor Control:

The motors are controlled with a PWM signal which will come from our PCB. The Pic32 on our board will receive instructions from the Raspberry Pi and then send the appropriate signal to the motors. While the rovers are on their paths looking for the objects of interest, they will maintain a fixed speed which will be set and monitored by the PIC32 to ensure the motor is travelling at the desired speed.

4.3 Future Enhancement Requirements

In the future, we would like to incorporate all sensors onto one PCB and do away with the Raspberry Pi altogether. This would allow us to reduce the size and weight of the rovers, making them more suitable to be the payload of a rocket. Obviously, we would also like to improve the quality of the mechanical components for this project to make it more suited for a lunar environment.

Additionally, we would like to extend the range of the rover-to-rover communication network using sub-GHz Zigbee communication. This allows for a range of up to 1 km.

Also, we would like to increase the number of agents in the swarm for a future release of the product.

Furthermore, it would be ideal if the rovers were more resistant to radiation and temperature ranges on the moon. The rovers should also be able to handle alien interactions (hostile and non-hostile).

5 High Level Design Decisions

5.1 Subsystem 1: Localization

DecaWave:

Decawave 1001 Module gives the position of each rover relative to 4 known and fixed anchors. According to decawave, the x-y position is usually accurate to within 20cm of the actual location. So, we should be able to meet our requirement of determining an object's position to within 2 meters of its actual location. The decawave module uses time of flight sensing to determine the position. This meets the constraint being feasible in a lunar environment, not relying on GPS.

The DecaWave will receive power of 5V from the Raspberry Pi.

The DecaWave will receive a clock signal from the Pi for SPI.

IMU:

An IMU, such as the Adafruit 9-DOF Accel/Mag/Gyro+Temp Breakout Board - LSM9DS1, will provide estimation for the angle of the rover. The IMU will include the 3 standard IMU sensors: gyroscope, accelerometer, and magnetometer. However, a magnetometer would not work on the moon, since there is no magnetic field on the moon. Therefore, we will first attempt to obtain accurate angle estimation through the gyroscope and accelerometer, only. A board like the Adafruit 9-DOF Accel/Mag/Gyro+Temp Breakout Board provides the output of all three sensors separately. This way, we are able to choose between using all 3 sensors or only 2 sensors.

The IMU will receive power from the Raspberry Pi.

The IMU will get its clock signal from the Pi for I2C.

5.2 Subsystem 2: Object Detection

Range Finder:

The range finder is used to find the distance from the rover to the object of interest and for obstacle avoidance. Once the camera identifies an object, the camera will analyze the object to determine if it is an object of interest. If it is not an object of interest, the rover's path will adjust accordingly. If the object is the object of interest, then the range finder will be pointed at the object and the distance measured. The exact location of the rover will be known with our location system. With the location of the rover and the distance from the rover to the object, the location of the object of interest will be easy to find. The range finder is able to communicate with the PIC32 via I2C.

The range finder will receive 5V power from the PIC32.

The range finder will get its clock signal from the PIC32 for I2C.

Camera:

The camera will be used to identify whether an object is an obstacle or an object of interest. The camera will be constantly scanning the area in front of the rover and using OpenCV to locate objects. The camera will be attached to the Raspberry Pi which will run OpenCV for real time computer vision.

The camera will receive 5V power from the Pi.

The camera will receive a clock signal from the Pi for CSI.

5.3 Subsystem 3: Raspberry Pi

Raspberry Pi:

The Raspberry Pi will be reading and interpreting the data from the IMU, Decawave and Camera subsystems. The Pi will communicate via UART with our custom PCB. The PCB will rely the range finder information, information from the servo motors and communication from the zigbee module. The Pi will communicate with the IMU through I2C and the Decawave module through SPI. The camera will communicate by CSI. The camera module we are using comes with the Pi and it works natively with the Pi. The Pi is also in charge of doing the math for finding the location of the rovers. The Pi will receive information from the Decawave and calculate the location of the rovers based on this information.

The Raspberry Pi will receive power from the PIC32 Custom Board. The on board voltage converter will step down our input voltage from 5V to 3.3V for the logic.

The Raspberry Pi has an on-board clock which will be used to control the Decawave and IMU.

5.4 Subsystem 4: PIC32 Custom Board

Custom Board:

The custom board will be responsible for low-level motor control, stepping down the power source voltage, and low-level communication between the two robots. The low-level motor control will consist of applying PWM motor drive signals to individual motors according to high-level commands sent from the Raspberry Pi via serial communication. It will also receive velocity feedback from each motor to perform simple PID speed regulation. The power will be received from the LiPo battery and stepped down to various voltages according to Figure 1 by using DC-DC converters.

The custom board will get its clock signal from the internal PIC32 clock.

5.5 Subsystem 5: Communication

Zigbee:

Zigbee is a local communication protocol that uses mesh networking. Although we are only demonstrating two rovers, the mesh networking provides a scalable solution. The Microchip MRF24J4DMA Zigbee module has a max range of 400 ft and operates at 2.4 GHz. The Zigbee module communicates over SPI with the PIC32.

The Local Communication will receive 3.3 V from a DC to DC converter on the PIC32 custom board which will step the original 7.4V from the LiPo battery to 3.3V.

The Local Communication will receive a clock signal from the PIC32 internal clock and will communicate over SPI with the PIC32.

5.6 Subsystem 6: Motors

Motors:

> Our bots will use servo motors for the drive train. The servo motors are 360 degree continuous rotation servo motors with encoders included in the servo package. The information from the encoders breaks out from a single wire on the unit and is sent to a GPIO pin on the PIC32 custom board. The encoder information can be used to verify the performance of the motors and the distance traveled. The motors are controlled with a simple PWM signal that will be sent from the PIC32. The velocity of the servo motors can be varied and controlled with the PWM signal.
>
> The servo motors will receive power directly from the 7.4V LiPo battery.
>
> The servo motors do not need clocks.

6 Open Questions

Path planning and obstacle avoidance

> Right now, we are assuming we will be using a ROS package for high-level path planning; however, we are uncertain about the details. We know that we will need a map of our environment, position and angle feedback, and some way to detect obstacles. Our current high level design provides a means of position and angle feedback through the Decawave and IMU, respectively. We also provide a way to detect obstacles, through Lidar. Lastly, we assume there will be a way to input our own map of the environment to define major boundaries of the test site.

Computer vision

> Additionally, we are uncertain about object recognition. We will rely on object recognition to autonomously find the object of interest. Currently, we assume we will use OpenCV or some neural network software. However, we are not extremely familiar with this software.

7 Major Component Costs

To preface our costs, it is important to note that we have developed a partnership with the Notre Dame Robotic Football team as they are invested in the implementation of DecaWave modules with their bots for positioning capabilities.

Given our current partnership with the Notre Dame Robotic Football Club, many of our components are being provided for us by the club's existing resources.

- 2 Cameras with flash ($25 each, $50 total)
- 2 Raspberry Pis (Robotic Football Club)
- 2 Zigbee module ($10 each, $20 total)
- 2 Chassis (3D Printed)
- 2 IMUs ($15 each, $30 total)
- Assuming 3 custom boards cost $75 total
- 2 Caster Wheels ($2 each, $4 total)
- 4 Servos ($30 each, $120 total)
- 4 Wheels ($4 each, $16 total)
- 2 Two-Cell LiPoBatteries ($41 each, $82 total)
- 5 Decawave Modules 1001(Robotic Football Club)
- 2 LIDAR modules (Senior Design Donated)
- Total cost ~$397

8 Conclusions

We believe that adhering to this high level design plan will allow us to move smoothly into an adequate lower level design that will enable us to fulfill our mission objective.

References:

Official NASA Challenge Rules

DWM1001-DEV Product Datasheet

MDEK1001 Kit User Manual

Servo Link

Wheel Link

Caster Wheel Link

Adafruit IMU

Camera with Flash Link

2S 5000 mAh Lipo Battery Link

LIDAR manuals Link

MicroChip Zigbee Module Link