Darius Emrani, Alazar Hailemariam
Ben Schoeler, Thomas Yu

Project Requirements: Autonomous Area Calculating Robot

While many of the requirements for our project have been given to us in the project description document, this document will specify fully what we plan to have the rover actually do. It will also describe what the rover will need in order for it to meet our stipulations. Furthermore, the web-pages and necessary additional requirements will be specified and explored in greater detail.

Functional Requirements

The rover will require some way to detect both the distance from the rover to the enclosure it resides in, and the direction that the rover is pointing. The proximity and orientation data needs to be accurate and precise enough so that the rover will be able to calculate the area of the enclosure within a reasonable error margin (+/- 10%). The exact error bound will change depending on the enclosure, but will be specified when the algorithm for calculating the data has completed. Additionally, the rover needs to be able to complete this task within a reasonable time-frame (<5 minutes).

Specifically, the rover will be placed near the center of the enclosure, and the rover will begin to rotate about its axis while taking proximity and orientation data. If a discontinuity is detected from the point of view of the rover, it will then move about the enclosure in order to get a clearer picture of the non-regular structure. Depending on the data that the rover has received, it will either begin rotating and scanning for data as it was previously, or go towards the wall to get more accurate data. Eventually, all corners will be visited as this is required by Mandatory Tasks #3 and #4.

The raw data provided by the sensors is first processed by the PIC microprocessors and eventually makes its way to the ARM processor where the majority of calculations are done and a “map” will be generated. Using this generated map, the rover has the following capabilities:

Additional task #4 – *Autonomously create a graphical map of the enclosure on the web interface*

Additional task #5 – *Use the graphical map to allow a user to position the rover at any location within the enclosure.*

The generated map’s points of interest (the sensor readings) are used to create a “mock enclosure.” The “mock enclosure” is displayed on the web page and should ideally look like the real enclosure from a top down view. The map enclosure is overlaid on top of a coordinate system. A user interface allows a user to input a desired location and have the rover move to that spot. The rover will not be able to navigate to coordinates that are outside the bounds of the enclosure. If a user specifies such a coordinate, an error message will pop up.

Web-Page Requirements

Darius Emrani, Alazar Hailemariam
Ben Schoeler, Thomas Yu

The web-page will be used for command and control of the rover. This web page as seen in *Figure 1* will be updated by the rover's ARM board. Sensor status information and the current value detected by each sensor (in terms of a distance) will be available on the right half of the screen.

On the left half of the screen, the data received will then be used to create a graphical map of the enclosure which will display what the area looks like as the rover sees it. Below this there will be a field that outputs the calculated area along with a calculated error bound. This will fulfill one of our team's additional requirements.

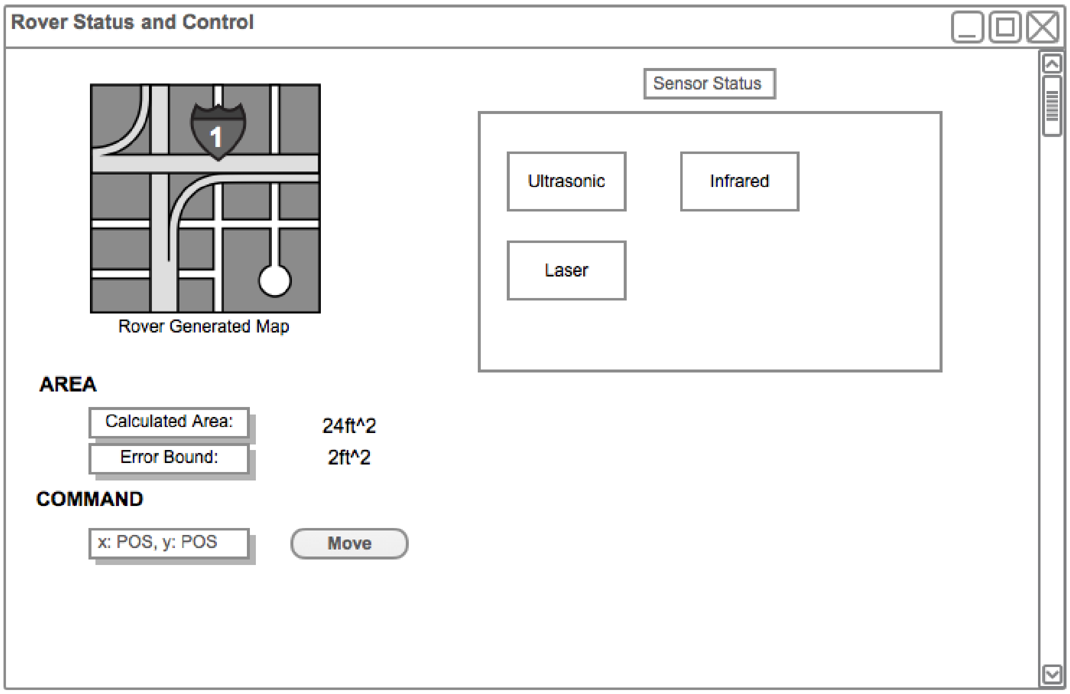

Figure 1: The Rover Status and Control Page

At the bottom of the screen there will be a command field where x and y coordinates with relation to the coordinate system on the map can be inputted and the rover will move to that position on the map. You will also be able to click start and stop buttons.

Algorithm/Decision-Making Requirements

The code residing on the ARM is divided into two categories, robot interaction and computational code. Robot interaction will consist of functions that communicate with the PIC to get sensor readings and physical robot navigation instructions like turning and moving forward. The computational aspect will consist of functions that create the map, calculate the area of the map, and calculate an efficient path to a destination. The map will assume the form of a two dimensional grid of cells. The more cells are in the grid results in a higher resolution map from which a better estimate of the area can be made. Further, the map will be designed such that newer more accurate readings from the infrared can be incorporated into any gaps missed by the laser. This is an extension of the map creation algorithm. The added advantage of the two dimensional grid form is that it makes finding

Darius Emrani, Alazar Hailemariam
Ben Schoeler, Thomas Yu

an efficient path to a destination given the source location easier to achieve. These two major aspects of the code will then be united in a one executable main where the logical behavior of the robot will be modeled.