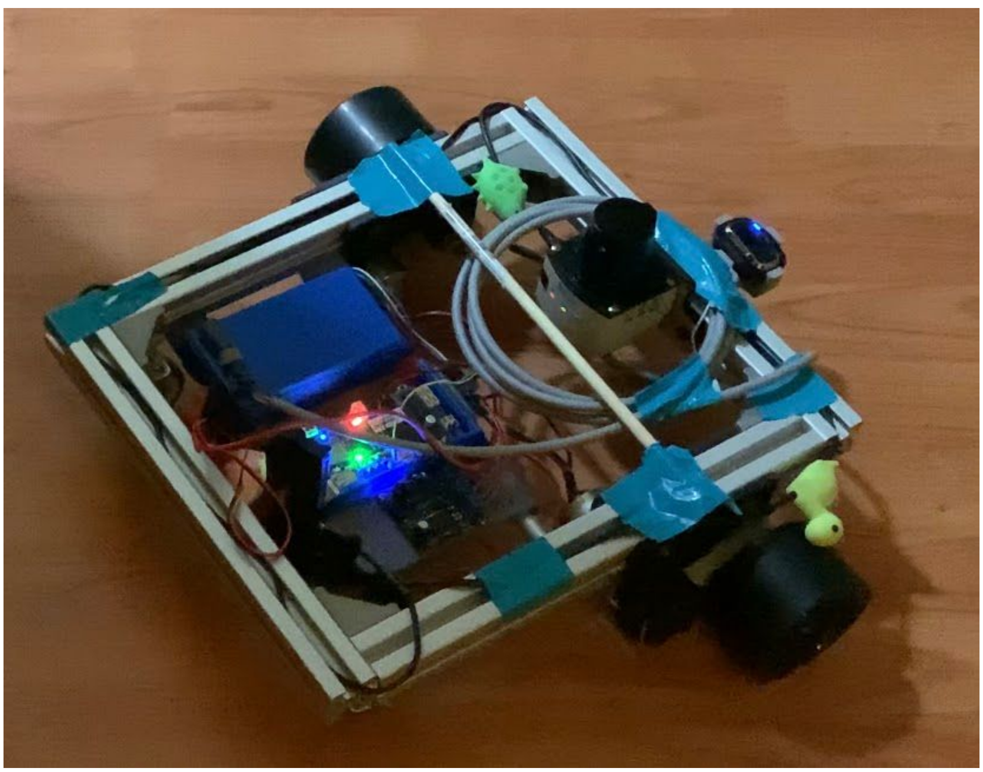
MXET 300
VALE: Personal Assistant

Shanley Mullen

Jesse Rosart-Brodnitz



Introduction:

VALE is one's very own map generating robot with speech recognition. Using the Lidar it can scan a room and save certain points that can be plotted into excel to see where different objects and walls can be saved and seen by the user. From there we can build upon this scope in the future to have the robot know where to go, but that was not a part of our initial goals this semester.

To create this map layout we had to implement a lot of different concepts, those covered in class and those that were not. This meant quite the learning curve at the beginning, but overtime we were able to deduce how to combine multiple elements together to make a fully functioning map. Although, one major issue that will be discussed later is one involving power to the system. Other than using the lidar to scan the environment for usable points within a bounded region we also found the distance traveled by the robot using the encoder values. Plus, we utilized the camera to pick up audio input from the user if a certain area should be saved to the map program such as a chair or movable object. We also allowed the robot to speak to us by using the speaker system that we placed onto our SCUTTLE bot. Overall, this project was quite difficult at the start since we spent a lot of time trying to get programs such as Breezy SLAM working instead of just making our own.

Deliverables and Progress:

The main deliverables set out for this project included map creation, and localization. However, there were some sub deliverables set out as way points to make sure the goal could be achieved. The first deliverable was motion which included driving around the room observing the environment at a safe distance. This involved computing the location of the object relative to the robot. The second task took into account the odometry of the robot. This took the information given off by the encoders to track the robot relative to the starting location. The third deliverable was prompting the user to answer whether an object seen was part of the environment or something temporary the Lidar scanned. For this portion, through the speakers, the robot asked the user whether or not the object scanned should be added to the map. A yes or no input was expected and depending on your response, the robot would respond accordingly and the point recorded or not. This used both speech to text and text to speech capabilities of the robot. A sub deliverable was set out for each of these as well, however for the purposes here, they can be combined into one larger deliverable. The next task involved the output file. The goal here was to output a file listing all the X and Y data points of scanned features in the global environment. This used a log function and output a list of data points to be graphed to see the map. The last deliverable and the most difficult one was getting the lidar to properly output all the data points for the global map.

Where this deviated from the new plan was the map creation. Instead of directly outputting the map for the user with the robot's location, the robot now outputs a list of X and Y components for the user to graph on their own. This also slightly negates the use of the odometry with respect to the map but still outputs information just as if not more important than the map because in this format, the data can be used and manipulated. The original plan for this project was the use and proper implementation of a SLAM algorithm done mostly by the lidar with the motion being controlled by the algorithm. The goals to go about this were shifted however, to localize using odometry and use the lidar for obstacle avoidance instead.

One of the reasons we had to shift from using a traditional SLAM algorithm already out there was the lack of a GUI. A new method has been created to make a GUI out of a client computer but has not been tested yet. Through this method, ideally, one would be able to run the master program and a scatter plot depicting the global coordinates of each data point would be recorded on a continuously updating chart. This chart could also take in the global position of the robot and tracks its progress through the map. This would be the next step for this project.

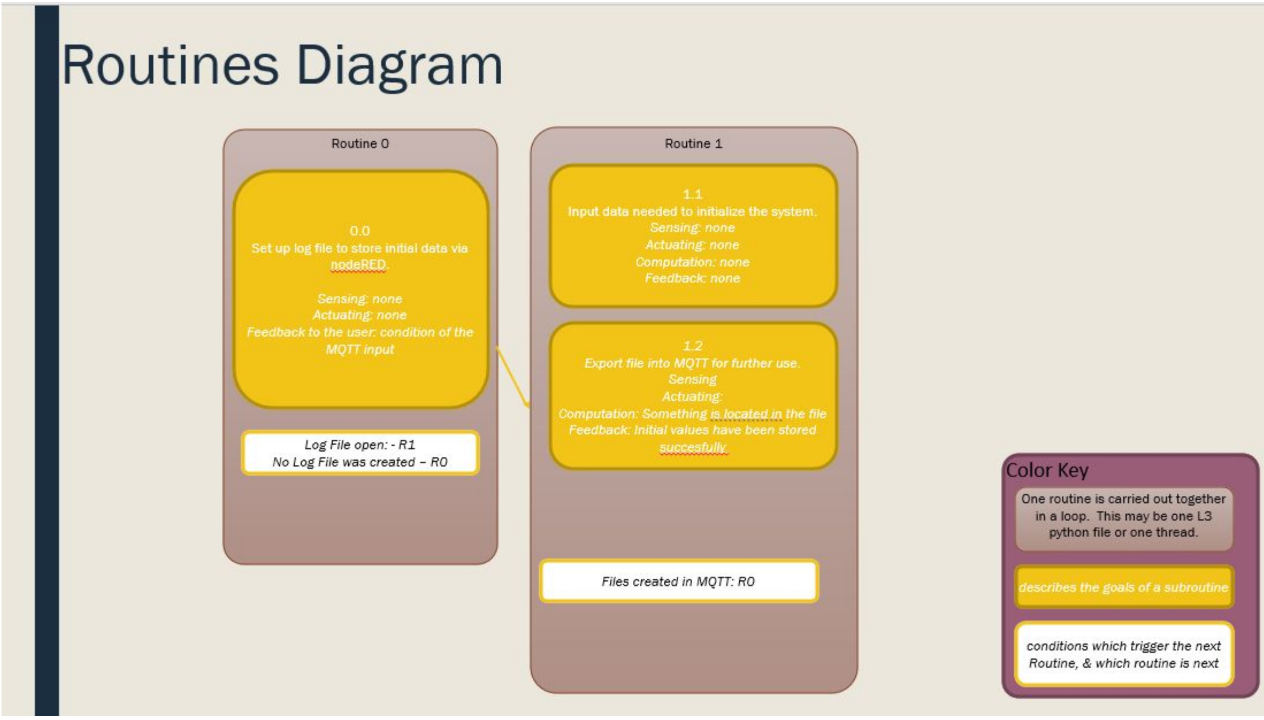


Figure 0.1. Initialization of Subroutines

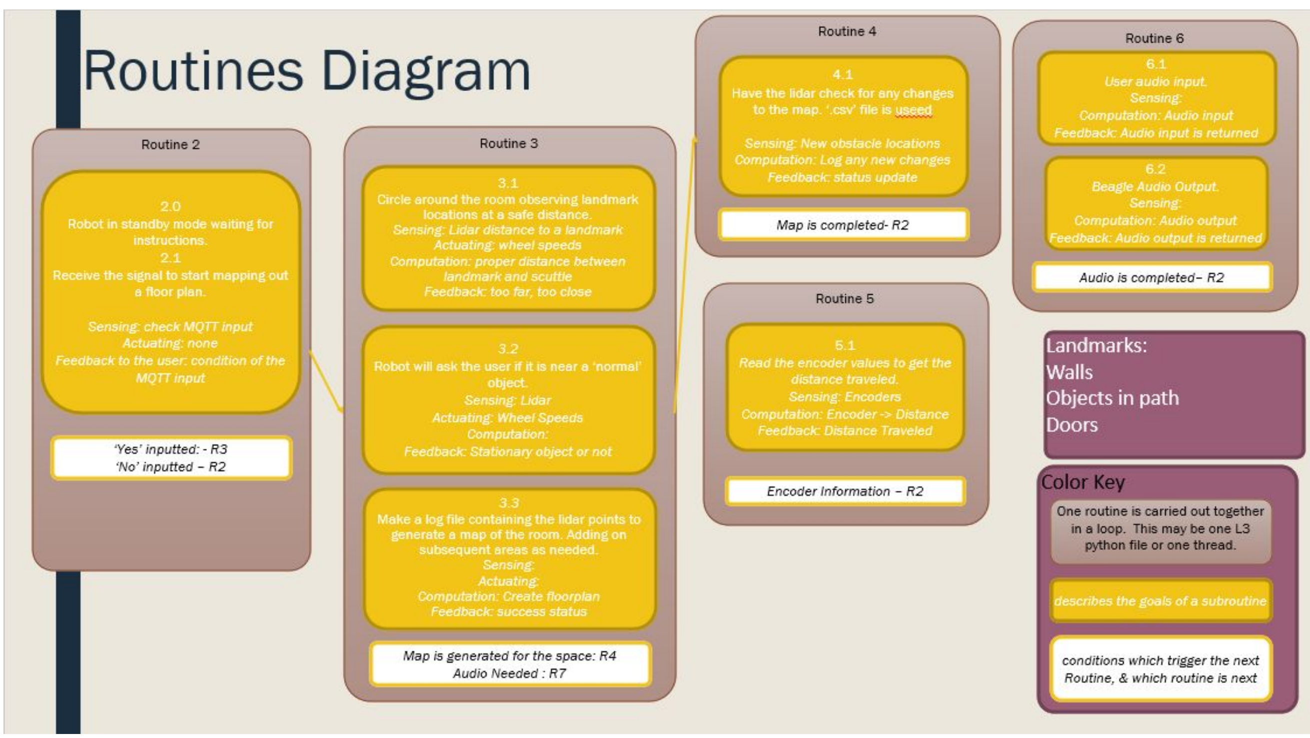


Figure 0.2. Main Subroutines Diagram

Results:

To get our results from the project we first had to run our code on the robot and then have it travel around the space reading points. From there we could tell it to save certain points as opposed to others. Then we exported those points into an excel file and plotted them. This was done since Cloud 9 does not have a GUI that we could use for plotting data in real time. The figure below shows the plotting of test points while this video link, [Robot Demonstration](), shows the process as it happens. We found the map to be fairly accurate when compared to the actual floorplan it was reading.

Table 1

-0.6545285877	0.02692675			
-0.6545285877	0.02692675			
-0.6455609751	-0.003222850457			
-0.6455609751	-0.003222850457			
-0.6566313862	0.03371425824			
-0.6554259941	0.02983672818			
-0.6554259941	0.02983672818			
-0.6527503334	0.02110231462			
-0.6551262339	0.02886690646			
-0.6551262339	0.02886690646			
-0.6563290765	0.03274514333			
-0.6563290765	0.03274514333			
-1.513716544	-0.566392723			

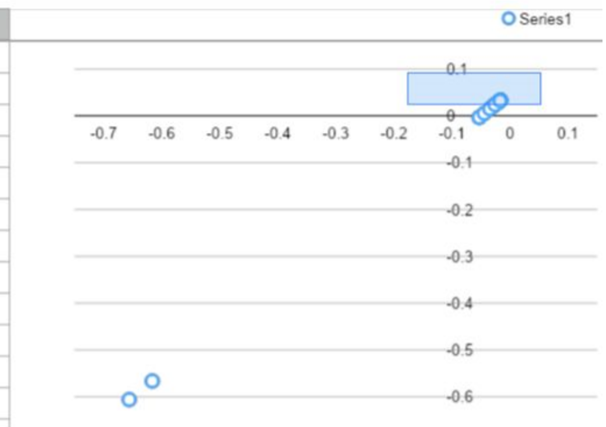


Figure 1. Points plotted in room

Summary:

During this project we learned quite a lot. One of the first things that we learned was how to set up the flite add-ons in python to properly use text to speech. From there we were then able to use speech to text for audio response as well. We found the voice recognition and response to surprising be the easiest part of the project. After that was completed we tried working with different SLAM algorithms to no avail. After those did not turn out we decided to create our own code for making a map.

To start that process we first had to learn how to take a frame reading from the lidar then determine the area of movement, the bounding circle surrounding the lidar, and how to shift the points as they were read. It took a lot of troubleshooting to get the proper distances. Also, we had to take readings from the encoder and convert that into a distance traveled for the robot. Once we figured out how to do those components we then learned how to properly export that data, so that it could be used for the creation of the map. Also, one big issue that we ran into was not having enough power for all of our components now that we have this knowledge if we were to redo this project we would provide more power sources to the robot since it only worked when it was plugged in.

Conclusion:

In conclusion, this project was more time consuming then we originally anticipated for it to be. We ran into multiple issues along the way even having to redo some of our initial calculations. Although, we were able to fix these problems eventually. This project was a great way to combine what we have learned in class with other techniques. Overall, this project was successful and could be very useful in real-world applications.