Moving Company Robots

Osbaldo Vera

R11364327

Texas Tech University

ECE 3334-302

William Ray II

Oluwatobi Oyinloye, Mark Matthews, Seth Larson

October 23, 2018

Abstract

This paper describes the hardware and software implementation of *Moving Company Robots*, a swarm robotics system designed to wirelessly control 4 robots into a formation and navigate through obstacles and turns to a desired destination. Utilizing ESP32's ADC and UART serial communication to send encoders and proximity sensors data through its built in Bluetooth module, the system heavily relies in odometry to successfully maneuver the robots. This paper is written to discuss the GUI, Robot Hardware and Robot Software. The purpose of this system is to help move objects of varying shapes with the use of swarm robotics.

Table of Contents

List of Figures

1. Introduction

Mimicking the way natural systems work has been an effective way to design and implement multiple man-made systems. *Moving Company Robots* is designed to mimic swarm species, most common in nature, ants and bees. These species control multiple moving bodies to accomplish one goal, gather and build to survive. This system is designed to move in a formation to set an object of different shapes, so it can help people move items to a certain destination. To accomplish this the bot will need to accurately maneuver itself into formation and collectively move to the desired destination while keeping its form. Figure 1 shows the general system diagram for accomplishing these tasks.

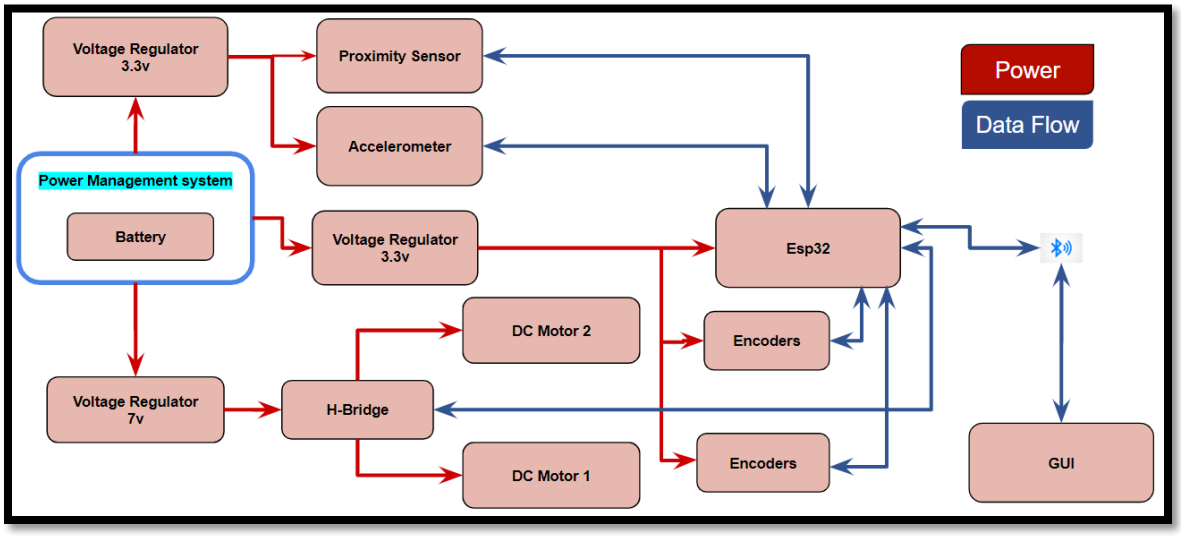


Figure 1: System Diagram of MaxProtect, By Oluwatobi Oyinloye

2. Computer interface

The bots are wirelessly connected to a computer, who's job it is to manage and control each bot's maneuvers and the entire navigation system. The Graphical User Interface is built on MATLAB's App Designer platform. In combination with its build in Instrumentation toolbox, App Designer lets one create multiple Bluetooth sockets and manage them within different objects. This allows the computer and robots to communicate effectively. Figure 2 shows the interface for connecting to the bots themselves. The GUI performs an initial scan of nearby Bluetooth devices and stores them into a dropdown menu, the user can then select which bot to connect to. Attempting to reconnect will result in a "Attempting to Connect..." message, where it either fails and displays a error or connects successfully. When connected successfully the drop-down menu and connect button will be greyed out and the lamp will turn green to signify connectivity. The Rescan button allows the users to scan for available bots within Bluetooth distance. Since the bots rely heavily on odometry, each bot's speed must be calibrated before use. The User will only need to press the Speed Calibration button and the bots will automatically calibrate its speed.

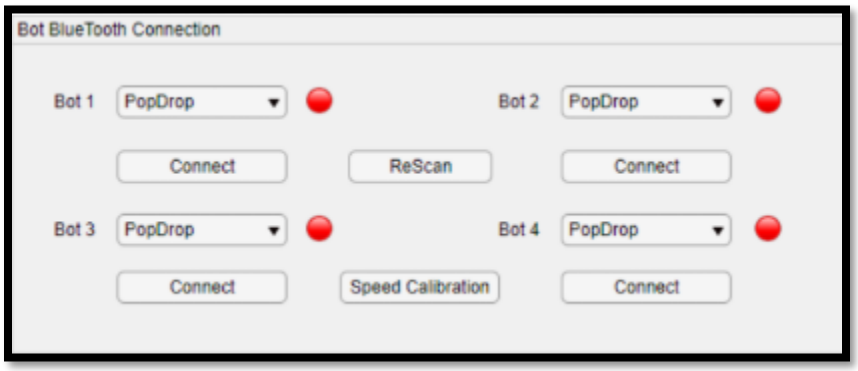


Figure 2: Bluetooth Connection Interface

Messages will be displayed in the panel Section of the GUI shown in Figure 2, The panel consist of two items, a reset button and a textbox. The reset button has the functionality of resetting the entire interface, this includes the Bluetooth sockets and Destination in the navigation Panel. The text box updates text by appending new string variables into a string array. This will display error messages and connectivity updates as well as pathfinding updates.

Figure 3: Panel Interface

A control Panel is accessible in the GUI, containing three tabs: Formation Control, Navigation, Manual. The Formation Control will send and receive feedback data to maneuver each bot into formation, currently this includes a box, cylinder and t formation. Once finished the bots will send data signifying that they have finished, and the textbox will be updated with the messaged “Finished Formation”. The Navigation Tab includes two numeric text fields for a x and y coordinate of one’s desired destination, when clicking the begin button it will plot the destination point and compute a safe path for the bots to fallow using the A* algorithm. The update messages for these will be as fallow: “Creating path”, “Fallowing Path”, “Destination Reached!”. The last tab is the manual tab, which is used to move a specific bot or all the bots in a specific manner. Changing

the WASD State button will allow the user to maneuver the bot with WASD keyboard keys in case of misalignments or one or more bots get stuck. Three other maneuvers are added to the user's or debugger's needs.

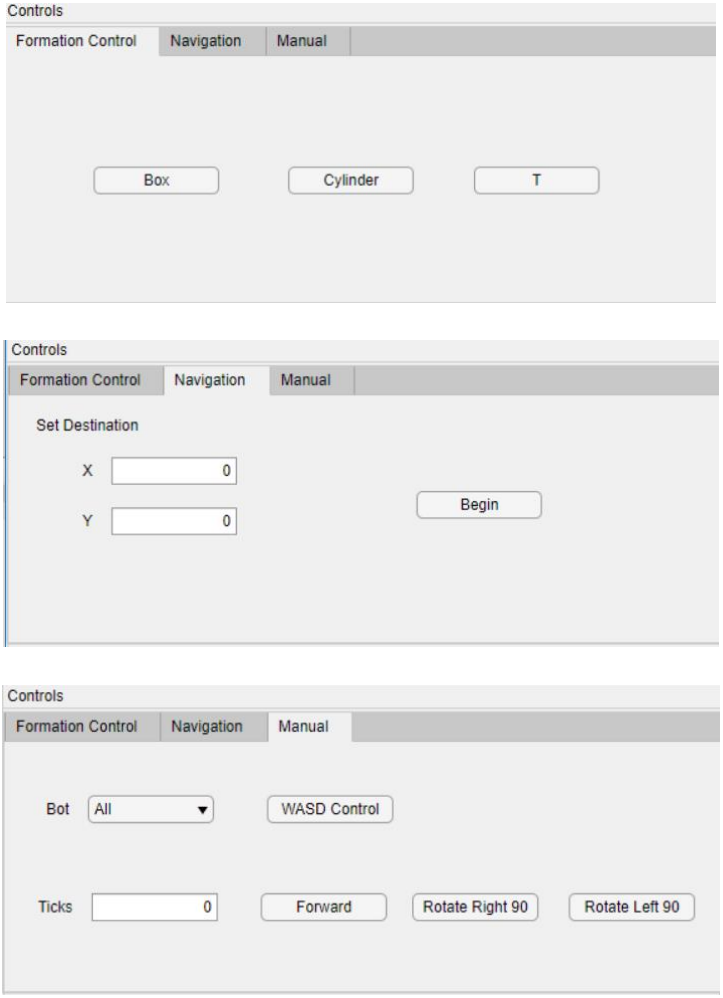


Figure 4: Control Interface and its Tabs

The final section of the GUI is the Binary Occupancy Grid, this is the figure where the pathfinding algorithm will need to execute. MATLAB's built in Robotic System Toolbox allows users to create a occupancy grid using a logical array filled with either 0's or 1's, where 0 are location that are free to pass, and 1's being locations where there are obstructions. Changing the map is made convenient for the users, by changing

the included map.xls excel file one can change the terrain the robot will be navigating through, Figure 5 shows the GUI grid corresponding with the Excel file. The grid operates in a 10m x 10m layout with 10 cells per meter. The Excel file is an array of 100 x 100 cells, each cell corresponding to .1m on the grid. This allows the system to be as precise as possible when executing its path.

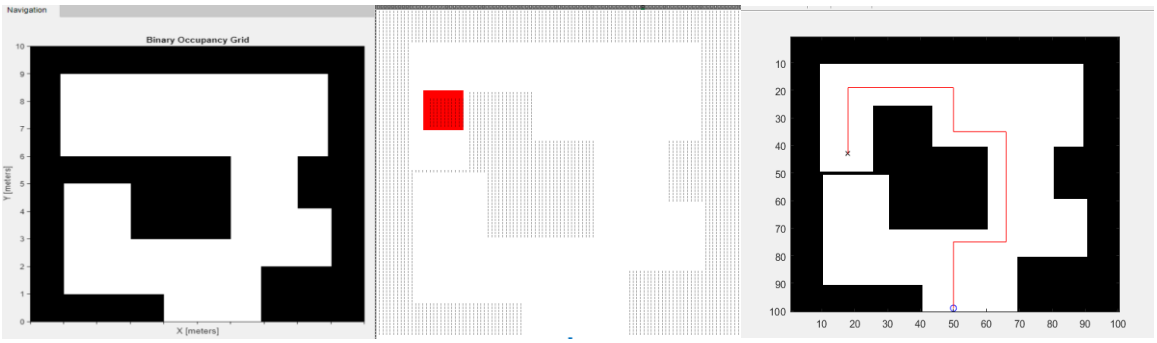


Figure 5: Occupancy Grid with Path Fallowing Elements

With the Binary occupancy map, the A* algorithm checks its neighbors' cells from the starting position from 1 to 8, or .8 meters the size of the robot formation, for any blockage. If it does not it continues with the algorithm leaping 8cells front, right, left position. The Destination area is also changed anytime the user inputs a x or y value to allow for this leap, instead of being one point on the binary map, it translates the point as the middle of a 9x9 array inside the map signified by the red area in Figure 5. When finished as shown in Figure 5, the optimal path is saved into an array derived from the parents of the cells who were in the optimal path, this path is translated into usable instructions for the bots to follow, it does this by running through the optimal path array while keeping track of whether the path has changed to the right or left by comparing the x and y values of the current and previous steps in the path. There are conditional statement to decide whether the bot is going forward in the x direction or the y direction

as well as to decide whether it has made a turn, a turn would result in a change in x ,y direction flag status. If the bot was heading forward then it would subtract the forward direction current, to its previous and that would be the distance the bot would need to go in .1meters.

The communication to the bots was made simple, if it is doing path fallowing it would send two bytes of information first byte would contain “f,r,l”, ‘f’ for forward, ‘r’ rotate right, and ‘l’ for rotate left, followed by the second byte telling it how many encoder values to move the desired distance, sending a 1 would result in .1m movement after the speed calibration. On top of those commands it also included “W,A,D,S” Where “W,A.D” were used for manual movement corresponding to WASD movement without the backwards capability, while “S” was repurposed to the speed calibration.

With all the panels combin makes the most effective experience for the user, shown in the figure below.

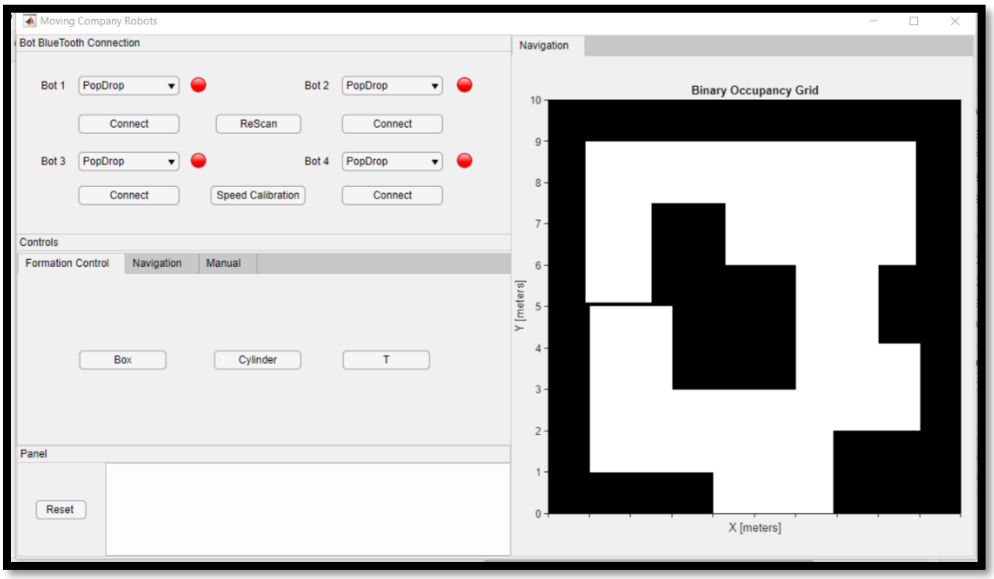


Figure 6: Graphical User Interface

3. Robot Hardware

Each bot will be packaged with 4 IR proximity sensors(gp2y0a60a60szlf), 2 DC motors with encoders (Rover 5), 1 H-bridge Circuit(L298) and an ESP32. The ESP32-wroom will be a system on chip module that contains the ESp32 microcontroller with packages Low powered Bluetooth and Wi-Fi modules with accordance of IEEE 802.15 standards and meeting FCC regulations.[1] The ESP32 is a dual micro pressor with 448kB Rom and 40Mhz Clock frequency [1]. It also includes 18 channel 12bit ADC to use with the 4 proximity sensors. The chip will be mounted onto green board with the given layout Shown in Figure 7.

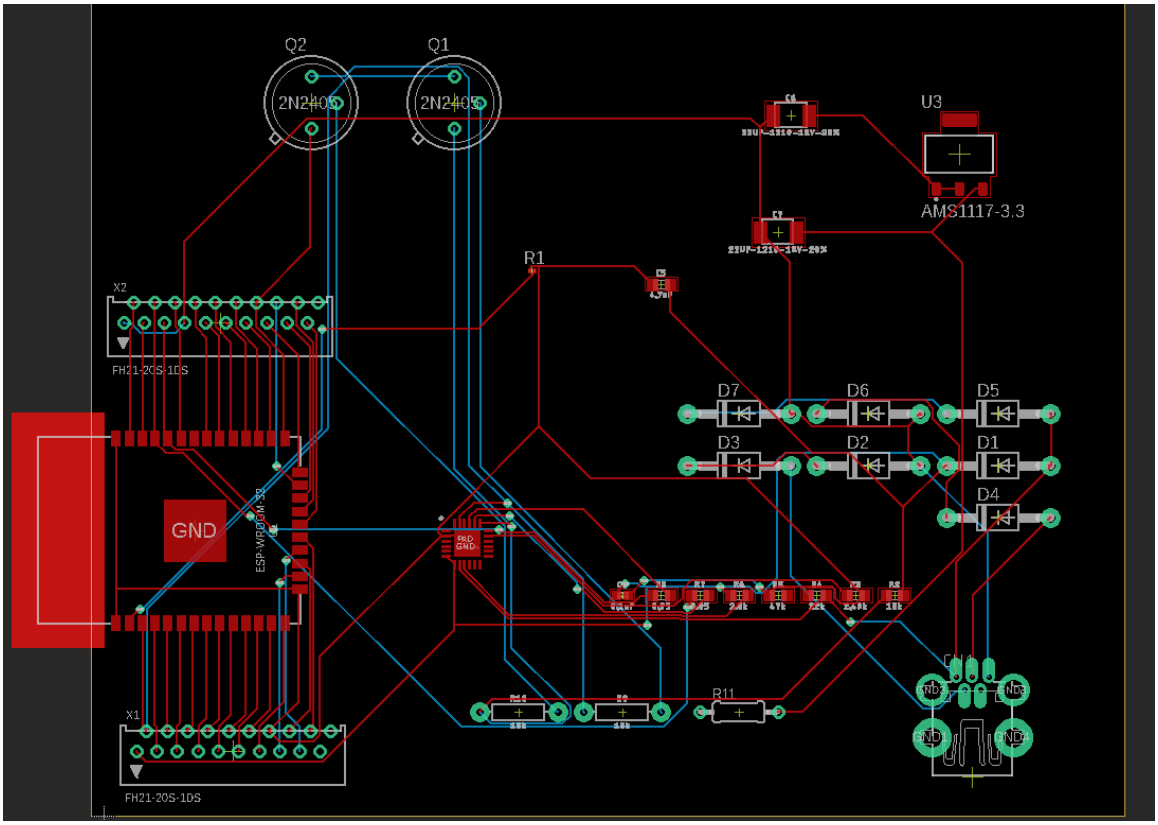


Figure 7: PCB Design

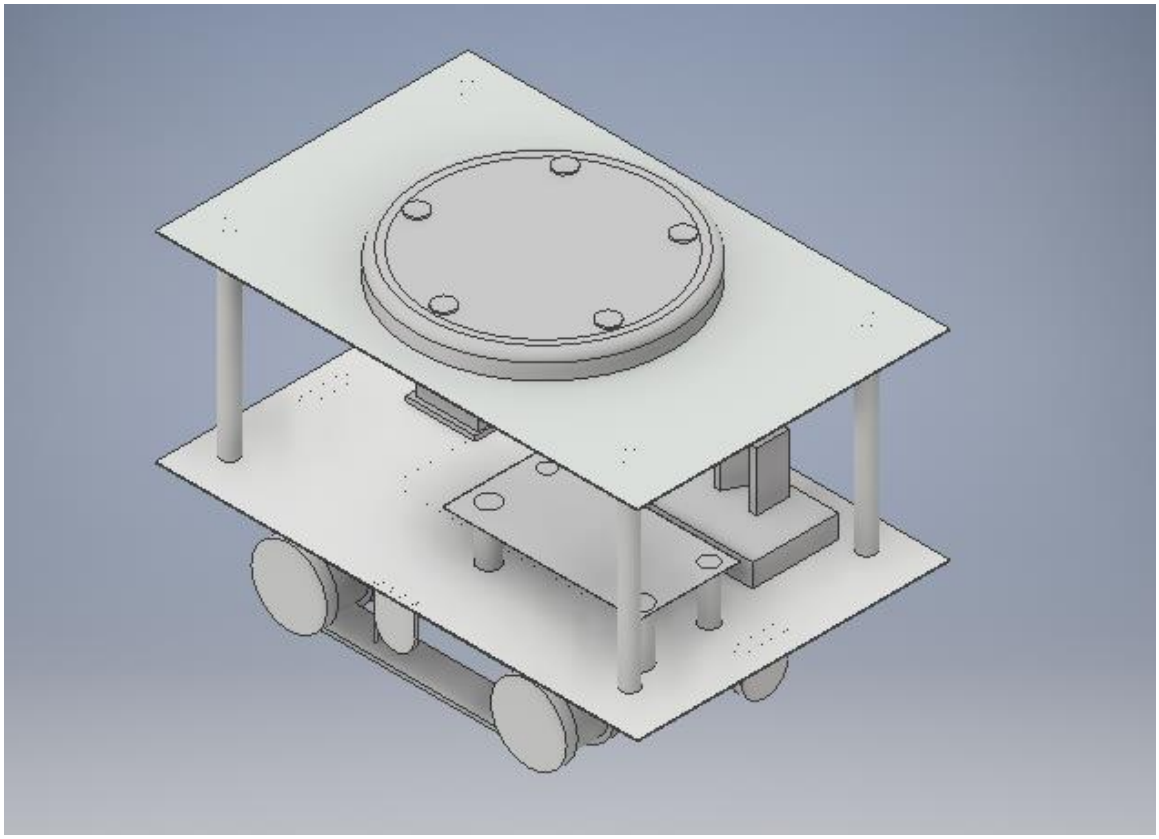


Figure 8: Bot Structure, by Oluwatobi Oyinloye

The bot basic structure is shown in Figure 8, it contains a middle area for all the electronics to seat as well a lazy Suzan bearing to help objects stay put when rotating right or left. The sheets are Plexiglass, which can support up to 33lbs of weight before cracking ensuring a package of 20-30lb range. The proximity sensors where mounted on the four sides of the bot.

4. Robot Software

The Bots Software is needed to do two major functions to complete the path, it must first be able to accurately move a distance forward and in a straight line and then it must keep a constant speed to match all the other bots. The ESP-32 made it simple by including a couple of built in hardware functions, like a pulse counter and PWM generator. The pulse counter allowed use to count the encoders without using valuable interrupts which would

be used for the proximity sensors. The SOC also included a two-core ARM processor which allowed us to constantly check the encoder values on one core. This core was purely checking whether both encoder values where within 3 ticks of each other to keep them in a straight line as much as possible or turning at a consistent rate. It did this by lowering the PWM by ¾ the speed calibrated PWM. The speed calibrated PWM. The Figure below shows the algorithm running in the second core.

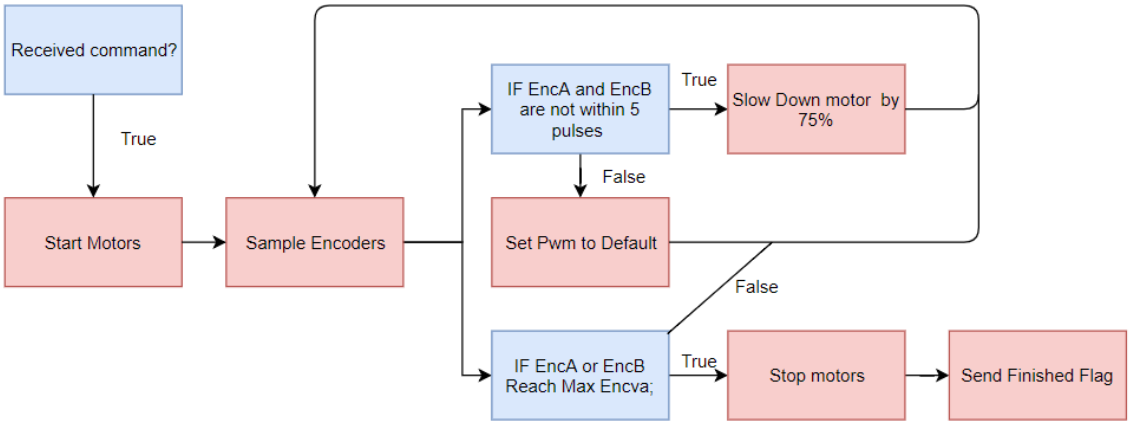


Figure 9: Encoder Flowchart

The Speed calibration relied on counting the amount of pulses per 50ms, making sure it ran 7 encoder ticks per 50ms. This is accomplished by taking increasing the PWM of the each side until they both meet that requirement if it hit above 100% the bot would stop and flash a red light as an error.

5. Conclusion

Overall the swarm will successfully maneuver itself into position and navigate a known terrain to move objects around the specified terrain. The system is tailored to be used by any individual who wishes to move objects around a given room and could see a commercial use with further professional development. As it stands the swarm can only move a maximum of 40kgs and is limited to certain formations but could be developed to be used with an ambiguous formation and terrain.

References

1. "Esp32-Wroom Datasheet", *Expressif,* https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf

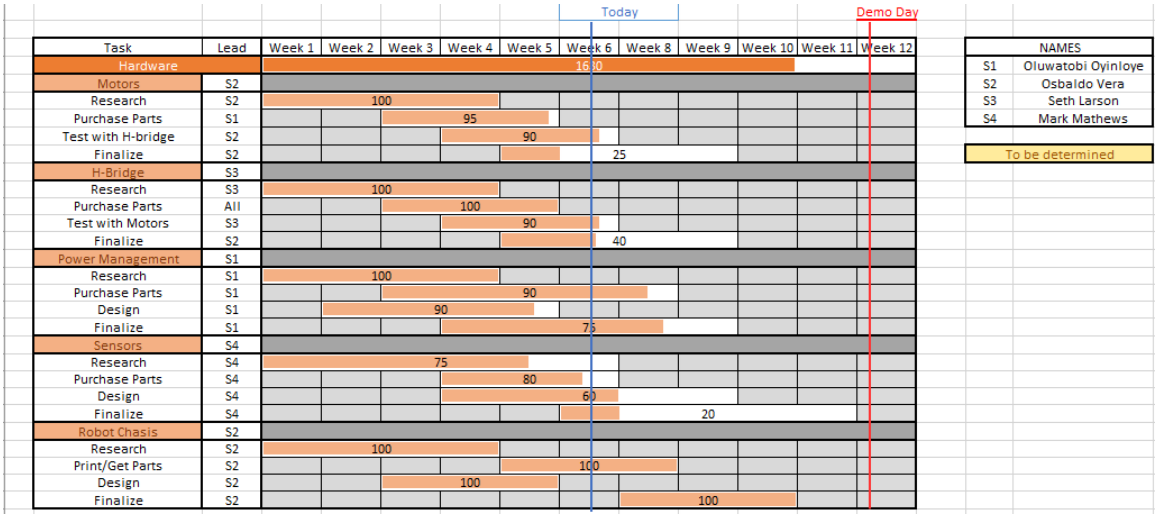
2. "Dagu Rover 5 Tracked Chassis with Encoders", *Pololu Robotics & Electronics.*[Online] https://www.pololu.com/file/0J467/Rover%205.pdf

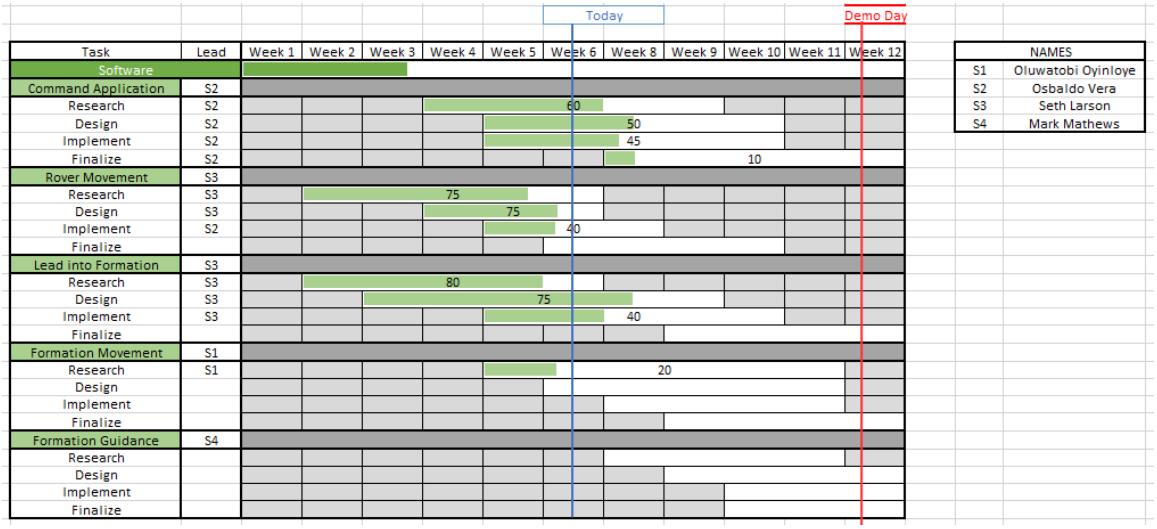
3. STMicroelectronics, "L298 Datasheet | SparkFun", *SparkFun*. [Online]. Available: https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf

4. "Pololu Carrier with Sharp GP2Y0A60SZLF Analog Distance Sensor 10-150cm, 3V," *Pololu Robotics & Electronics*. [Online]. Available: https://www.pololu.com/product/2476. [Accessed: 24-Oct-2018].

Appendix B

Gant Chart





Appendix C

Budget

	Running Total			Total Estimate		
Direct Labor:						
Category or individual:	*Rate/Hr*	*Hrs*		*Rate/Hr*	*Hrs*	
Mark	18	10	$180.00	18	10	$180.00
Tobi	18	10	$180.00	18	10	$180.00
Osbaldo	18	10	$180.00	18	10	$180.00
Seth	18	10	$180.00	18	10	$180.00
Total Labor:			$540.00			$540.00
Consulting Fees:						
Category or individual:	*Rate/Hr*	*Hrs*		*Rate/Hr*	*Hrs*	
Lab I	15	0	$0.00	15	0	$0.00
Lab II	18	0	$0.00	20	0	$0.00
Lab III	18	0	$0.00	25	0	$0.00
Lab IV & V	25	0	$0.00	35	0	$0.00
Lab Tutors	40	0	$0.00	40	0	$0.00
Lab Assistants	40	0	$0.00	40	0	$0.00
Mr. Woodcock	100	0	$0.00	100	0	$0.00
Instructor	200	0	$0.00	200	0	$0.00
Total Contract Labor:			$0.00			$0.00
Direct Labor Cost:		Subtotal:	$540.00		Subtotal:	$540.00
Indirect Labor Cost	*rate:*	100%	$540.00	*rate:*	100%	$540.00
Total Direct Labor:			$1,080.00			$1,080.00
Supplies And Materials:			$432.00			$0.00
(from Materials Cost worksheet)						
Total Direct Material Cost:			$432.00			$0
Equipment Rental:	Value	Rental Rate (Per Week)		Value	Rental Rate	
Oscilloscope	$2,851.00	1.00%	$109.97	$2,851.00	1.00%	$382.85
Function Generator	$292.50	1.00%	$11.28	$292.50	1.00%	$39.28
DMM	$259.00	1.00%	$9.99	$259.00	1.00%	$34.78
Power Supply	$699.00	1.00%	$26.96	$699.00	1.00%	$93.87
Total Rental Costs:			$158.20			$550.77
Total Supplies Materials & Equipment:			$590.20			$550.77
Total TDL+TCL+TDM+TRM			$1,670.20			$1,630.77
Business Overhead:		55%	$918.61		55%	$896.93
Total Cost:		**Current**	$2,588.81		**Estimate**	$2,527.70