SDSU Mighty Mouse

A Project Proposal

MicroMouse Competition

Team Members:

Mai Chao Her **Tareq Esmail**
Albert Fruner **Brian Lieu**
David Phan **Brian Snow**

Sponsored by:

SAN DIEGO STATE UNIVERSITY

Introduction.

Abstract:

Micromouse is a robotic maze solving competition held by IEEE since the 1980s. The robot must traverse through the maze and find the center in the shortest time possible. The robot must conform to the IEEE Region 6 Southwest's official rules of design constraints. Team SDSU Mighty Mouse's objective is to design and build a robot capable of finding the center of the maze in the shortest time and compete against other SDSU micromouse senior design teams on May 8, 2015.

Project Description:

SDSU Mighty Mouse's main goals are to design a robot with speed, accuracy, and precision. We will use the STM32F407VGT6 microcontroller with a processor speed of 168 MHz to control all the required peripherals. To achieve our goals and design a functioning robot, we will incorporate feedback control using high resolution motor encoders, infrared sensors for wall detection, and a gyro for velocity on turns.

To acquire a speed of about 0.3 meters per second, the robot will be about 10 ounces with dc motors for speed. Our robot will traverse the maze and find the center within an approximate time of 1 minute. Encoders will be used to monitor the speed of each motor's wheel. The gyro will feed back orientation data to the microcontroller which will help control the rate of rotation around a particular turn in the maze.

Infrared light detectors will be used to provide accuracy and precision. We will be using photo transmitters and receivers to sense if there is a wall. The detectors will feed back the information to the microcontroller. The feedback information will be used to help avoid collisions while the robot is traversing through the maze.

The robot will be running on the flood fill algorithm when exploring and finding the center of the maze. Fine tuning and testing will be done to perfect the algorithm. We will also incorporate a decision making process which the robot will always make a right turn first if there is an intersection.

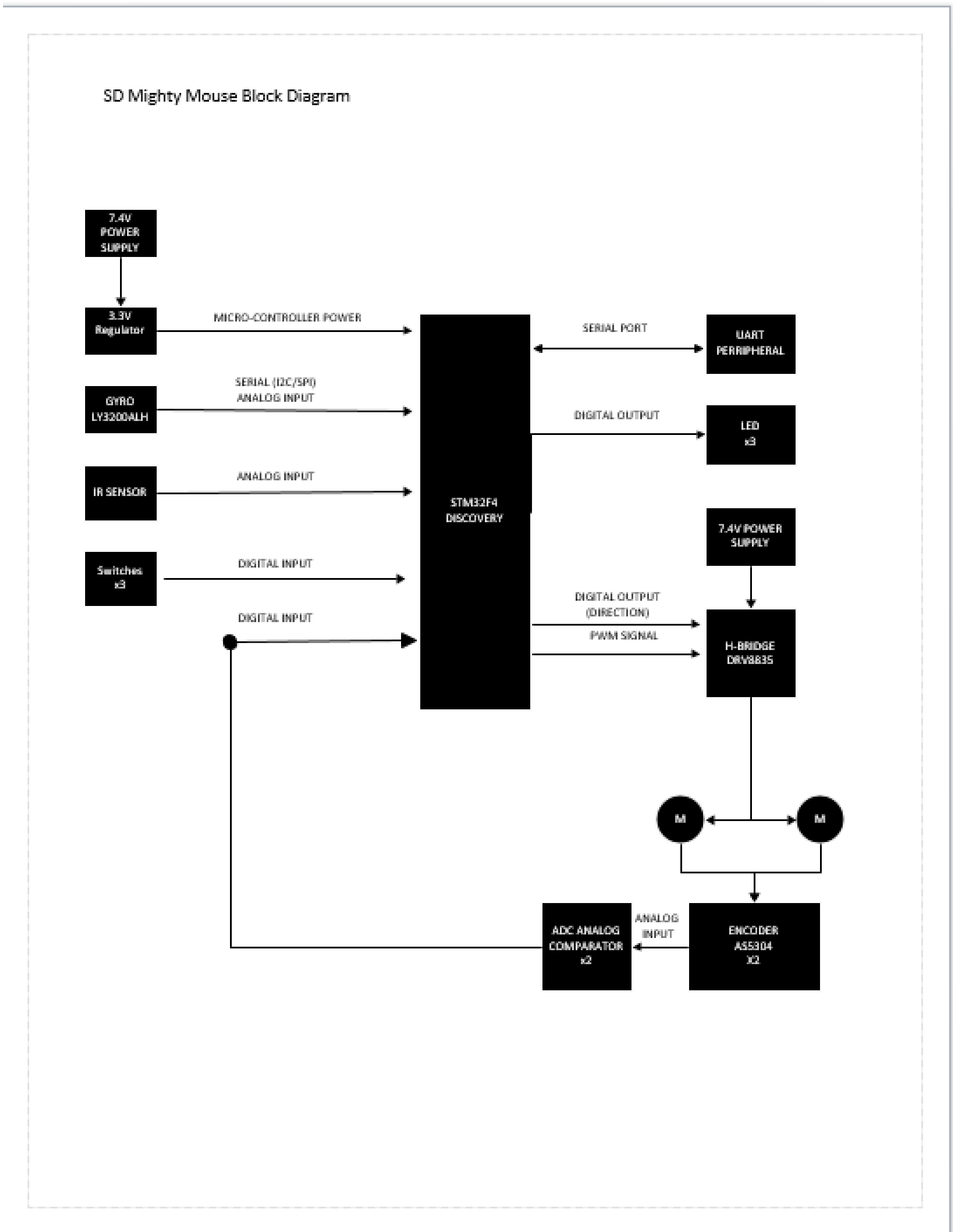
Since each MicroMouse contest is allocated a total of 10 minutes of access to the maze, our robotic mouse will consist of two modes:

Modes:

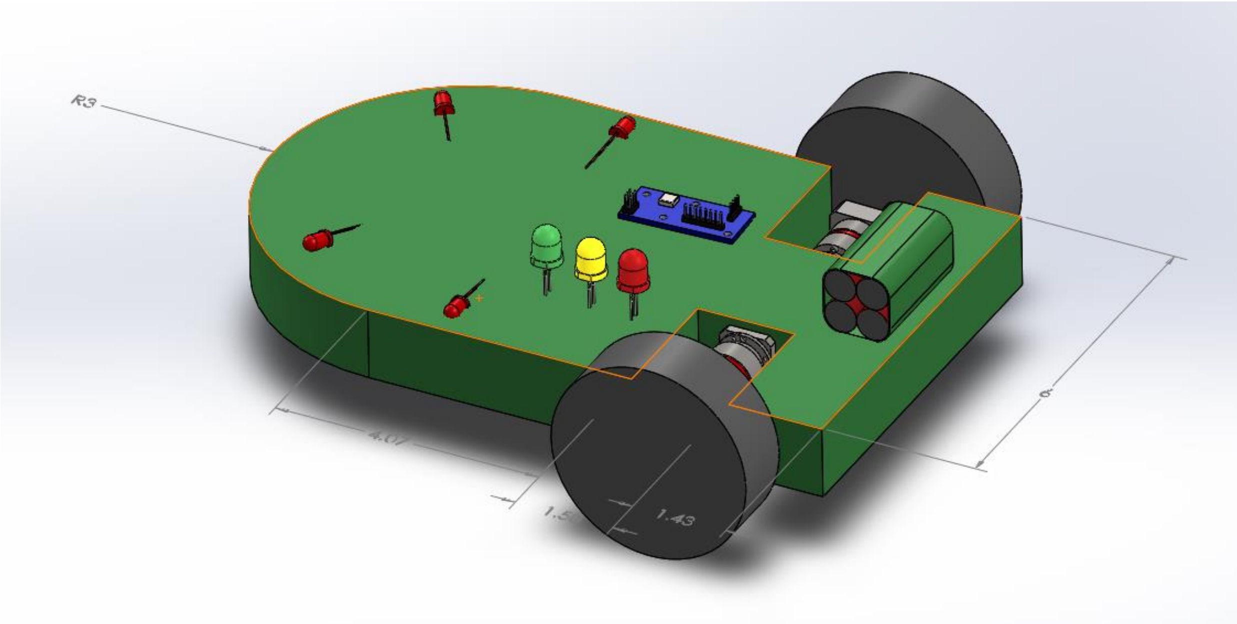
- Mode 1 will map the the whole maze, traversing through every square (max time of 7 minutes).
- Mode 2 will find the center in the shortest time with three different speeds and turn profiles: slow, medium, and fast.

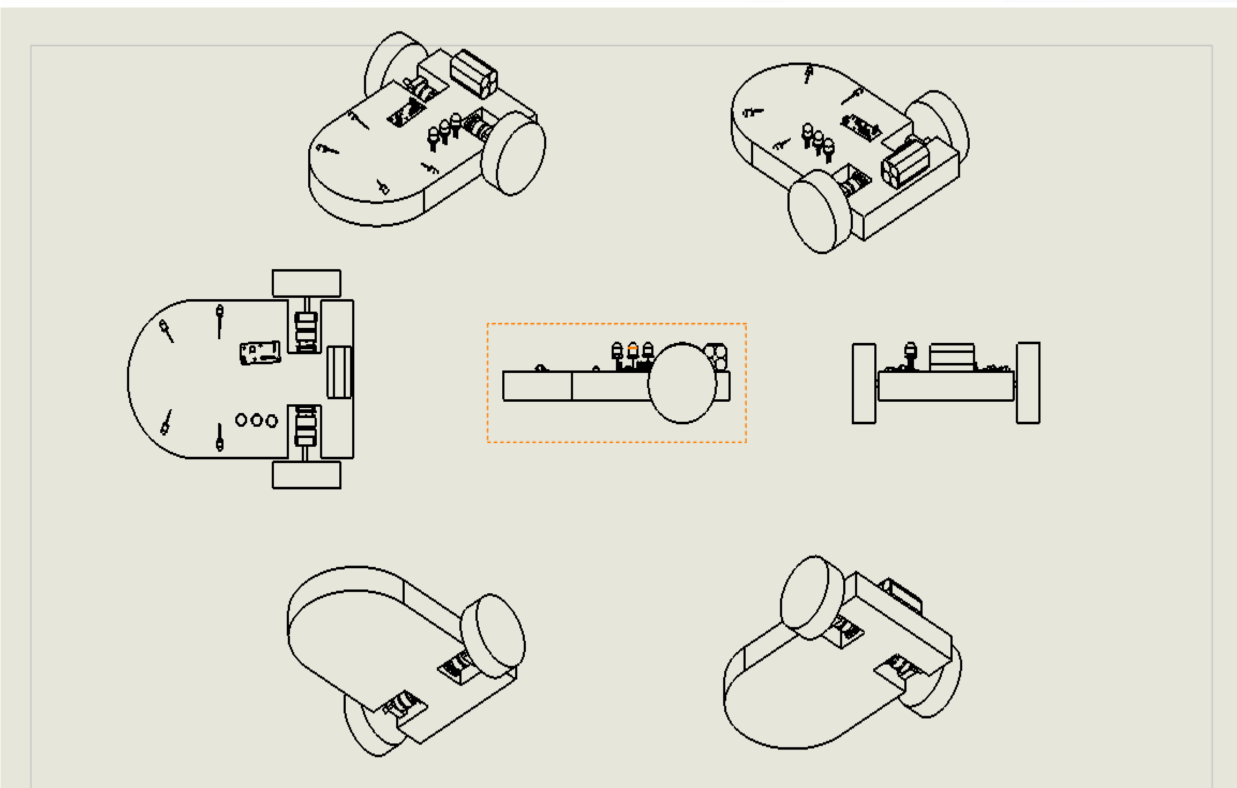
Design.

Block Diagram:



Mock-Up Illustrations:





Performance Requirement:

- Micromouse must be able to detect walls to avoid colliding with maze walls that will be the dimensions of 18 x 18 cm.
- The robot will map and store the maze.
- The robot will find the quickest way to the center.
- The robot has 10 minutes to solve the maze by getting to the middle.
- Once the robot has mapped the middle, it must return to the initial starting point.
- A Micromouse cannot be remote controlled.
- The robot must not use a combustion engine.
- The robot can be interacted with while in the starting square, so it may use a manual switch to change modes.
- The robot must fit within a 25 x 25 cm footprint.
- The robot may not leave pieces of itself behind while traversing the maze.

Testing and Verification.

Testing Procedure:

By testing the micromouse robot, it should be easier to fine tune to achieve the project goals. The results of testing the micromouse mouse should verify how reliable the robot is in finding the center of the maze in a precise and fast way.

Hardware Testing:

To test the hardware, we will test each component individually. We will test each component to see if they function properly first then slowly tune them to our specifications. The first component we will test is the DC motor. To do this, we will connect it to a power supply and apply voltage to it. Because our battery supply is 7.4, this is the voltage we will use with the power supply. We will then go on to measure the no load voltage and current and the stall current. By doing this, it will tell us how much current is being drawn through the motors. To measure the speed of the motor, we will attach the wheels and measure their velocity using the tachometer used in the 380L. This will tell us how quickly the robot will move under our chosen voltage, and will tell us if there are any discrepancies in the two motors.

For both the IR Emitter and receiver, we will test them first as the manual data specifications and then slowly adjust it to our design specifications. We will mainly test them to see how fast they store/process data on distance from walls.

Separate from testing the motors, we must test the encoders after testing the motor. The encoders will be fabricated separately on the side of the PCB board to measure the 44 poles on the wheel.

Lastly, we will test the gyro. To do this, we will fabricate a separate PCB board and test it. We will test them first as the manual data specifications and then slowly fine tune them to our needs.

At the end, we will test all the extra components and modules to our design requirements. All data will be recorded and saved for future reference. Once everything is tested and tuned, we will solder and combine all the components to the PCB board.

Software Testing:

A software simulator will be used to test and fine tune the flood fill algorithms and decision making process with the following specifications:

1. The robot will turn right if there is an intersection.
2. The robot will only turn left if there are walls to in front and right of it.
3. The robot will turn 180 degree if it reaches a dead end
4. The robot will go straight until it comes to a wall in front of it or opening to its right.

The microcontroller will be used to control and test the peripherals such as DC motors, encoders, IR sensors. An oscilloscope will be used to test the pulse width modulation (PWM) output signal the microcontroller and H-Bridge drive will be supplying the motor. UART will output the readings received from each peripheral for testing and debugging purposes. The following tests will be done to meet the specifications:

1. Test motor drive functions (pivot turn left and right, constant velocity curve turn left and right, find optimal acceleration and deceleration profiles on straightaways, implement diagonal turns if we have time).
2. Test IR sensor voltage for given distance and find linear range to operate within.
3. Test interrupts for motor driver.
4. Test push buttons to switch between “mapping” and “speed run” modes.
5. Test switch to give the correct motor speed and turn profiles: slow, medium, fast.

Micromouse Testing:

Once the hardware and software testing is finished, we will combine the two to test the actual micromouse robot. The purpose of this is to sustain the max amount of speed and accuracy given the final design’s weight and dimensions. The steps for this testing are given below us:

- import motor function softwares to the STM32F407 Discovery microcontroller and adjust till microcontroller is able to process data from input and output ports

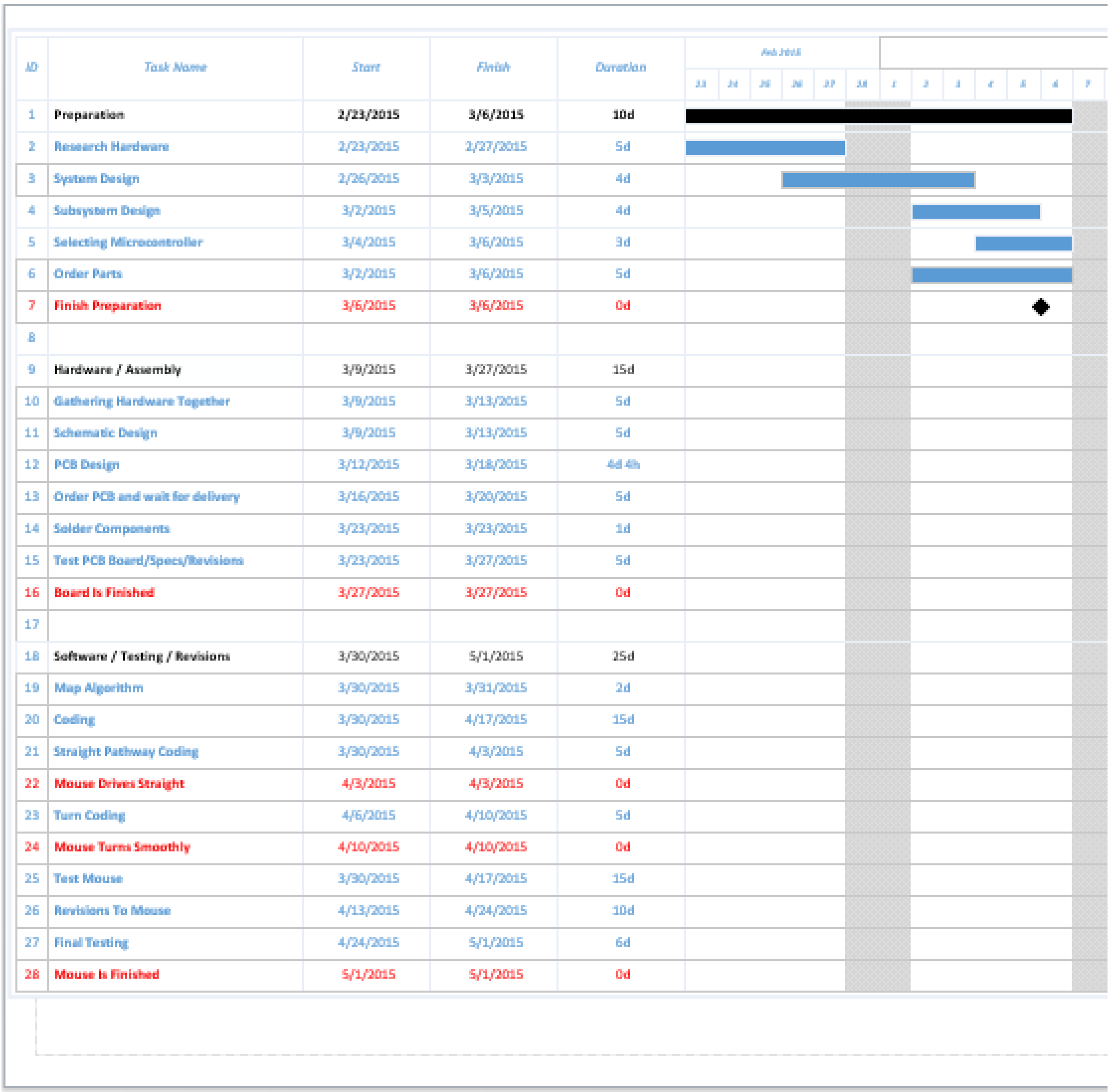
- Import navigation functions to the STM32F407 Discovery microcontroller and adjust till processed data can be achieved from input and output ports
- Test the robot to see if it is able to speed up on straight aways or slow down when needed
- Test the robot to see if it is able to make left and right turns and is able to drive straight without touching any walls
- Test robot to see if it is able to make curve turns compared to pivot turns
- Test robot’s first mode (mapping)
- Test robot’s second mode (speed run: slow, medium, and fast)

Benchmarks:

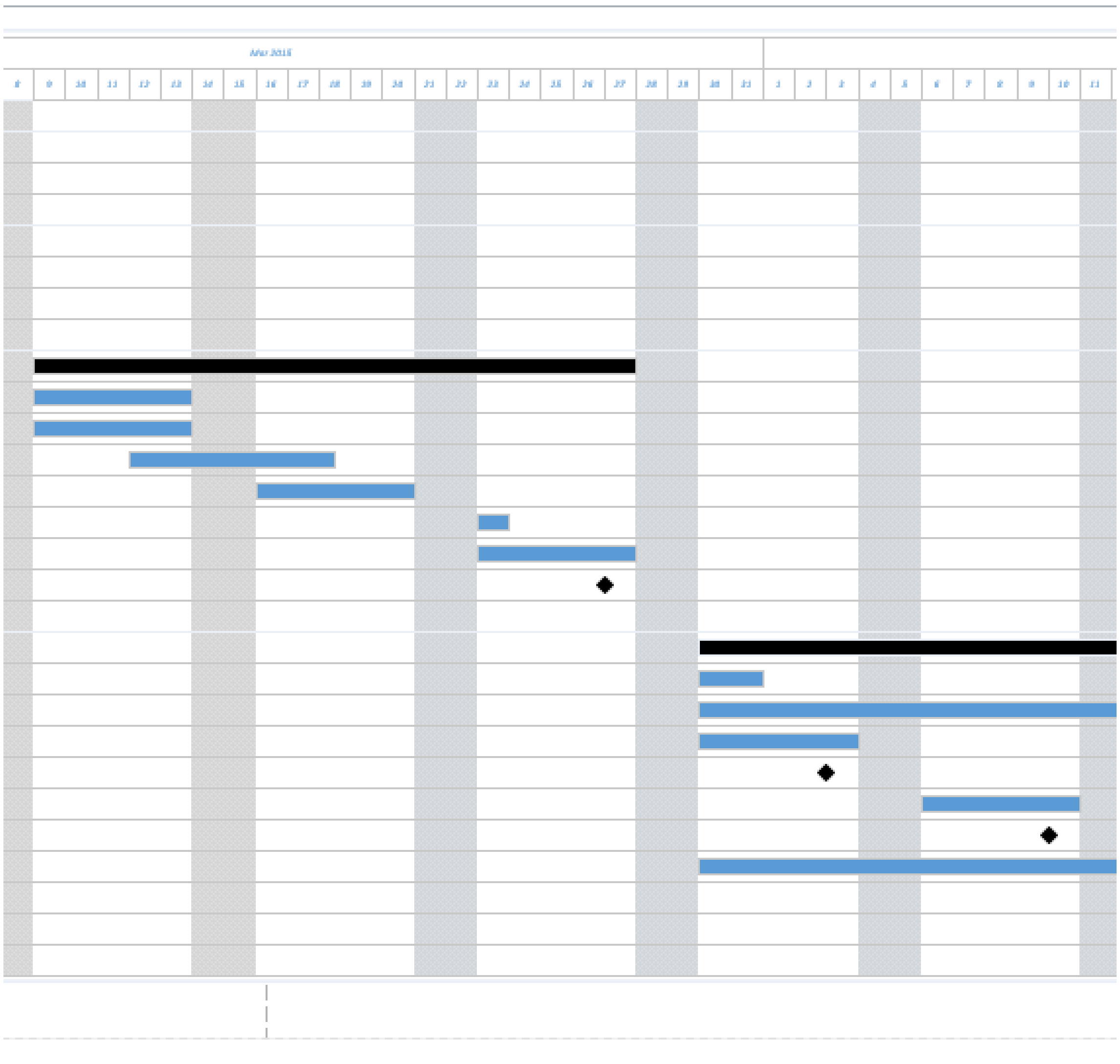
1) Top speed of 0.3 m/sec.
2) Pivot turn (90 degrees right and left, 180 degrees).
3) 90 degree curve turn at constant velocity - right and left.
4) Detect oncoming wall and implement deceleration at X distance. (Test IR sensors for distance and account for processing time)
5) Weigh less than 10 ounces.
6) Fastest run under 1 minute for any solvable maze configuration.
7) Robot will map to the center and reroute back to starting point within 10 minute.
8) Robot must avoid collision with walls during mapping mode and speed run modes slow and medium (not necessarily during fast).

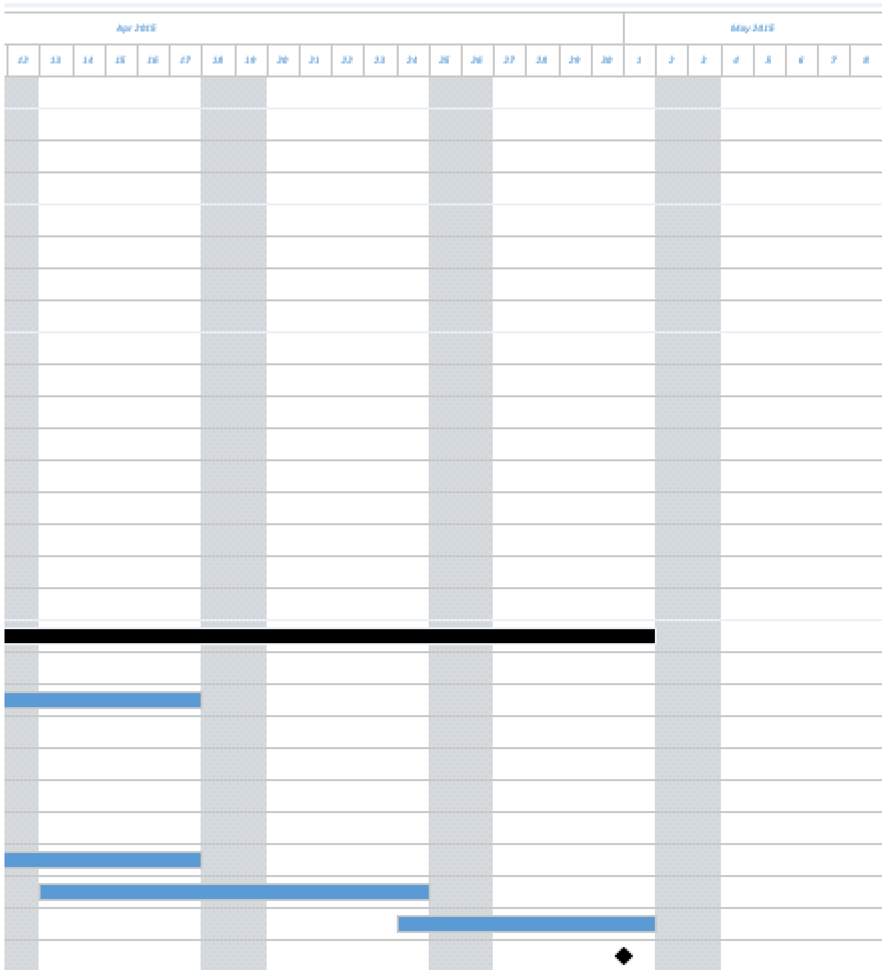
Project Management.

Gantt Chart:



ID	Task Name	Start	Finish	Duration
1	Preparation	2/23/2015	3/6/2015	10d
2	Research Hardware	2/23/2015	2/27/2015	5d
3	System Design	2/26/2015	3/3/2015	4d
4	Subsystem Design	3/2/2015	3/5/2015	4d
5	Selecting Microcontroller	3/4/2015	3/6/2015	3d
6	Order Parts	3/2/2015	3/6/2015	5d
7	Finish Preparation	3/6/2015	3/6/2015	0d
8				
9	Hardware / Assembly	3/9/2015	3/27/2015	15d
10	Gathering Hardware Together	3/9/2015	3/13/2015	5d
11	Schematic Design	3/9/2015	3/13/2015	5d
12	PCB Design	3/12/2015	3/18/2015	4d 4h
13	Order PCB and wait for delivery	3/16/2015	3/20/2015	5d
14	Solder Components	3/23/2015	3/23/2015	1d
15	Test PCB Board/Specs/Revisions	3/23/2015	3/27/2015	5d
16	Board Is Finished	3/27/2015	3/27/2015	0d
17				
18	Software / Testing / Revisions	3/30/2015	5/1/2015	25d
19	Map Algorithm	3/30/2015	3/31/2015	2d
20	Coding	3/30/2015	4/17/2015	15d
21	Straight Pathway Coding	3/30/2015	4/3/2015	5d
22	Mouse Drives Straight	4/3/2015	4/3/2015	0d
23	Turn Coding	4/6/2015	4/10/2015	5d
24	Mouse Turns Smoothly	4/10/2015	4/10/2015	0d
25	Test Mouse	3/30/2015	4/17/2015	15d
26	Revisions To Mouse	4/13/2015	4/24/2015	10d
27	Final Testing	4/24/2015	5/1/2015	6d
28	Mouse Is Finished	5/1/2015	5/1/2015	0d


Mar 2015
8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8 9 10 11



Apr
May
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
1 2 3 4 5 6 7 8

Budget.

Parts	Cost
Motors/Motor Driver	$100.00
Microcontroller	$100.00
Chassis	$100.00
Replacement Parts	$100.00
Encoders	$40.00
Wheels	$30.00
IR Sensors	$30.00
Gyro	$10.00
Batteries	$40.00
Miscellaneous	$50.00
Total	$600.00