Department of Electrical Engineering
The Cooper Union for the Advancement of Science and Art

Megamouse

An IEEE Micromouse Competition Submission

Sharang Phadke   Sameer Chauhan
phadke@cooper.edu   chauha3@cooper.edu

Megamouse is an autonomous robot that can explore and solve a maze as specified by the IEEE Micromouse competition. This paper outlines the competition specifications and describes an approach to developing a robot submission to the competition. The elements of a robust locomotion system, such as a kinematic model and the use of encoders, are described. The trade-offs of sensor systems, including speed, coverage, and precision are specified. A control system that uses sensor feedback to implement locomotion control as well as maze-solving algorithms is detailed. The Micromouse competition motivates research such as this project in the field of autonomous mobile robots (AMR), which has numerous household, military, and commercial applications.

Advisor: Prof. T. Cumberbatch   ECE 195A - Fall 2013

Contents

1 Introduction

The IEEE Micromouse competition is an engineering challenge that requires the application of precise mechanical operation as well as computationally efficient algorithms. The competition requires participants to construct an autonomous robotic "mouse" that can map out and solve a maze as quickly as possible. Successful submissions must effectively maneuver through the maze, determine orientation by processing sensor data, and apply maze solving algorithms to find the shortest path to the center of the maze.

The Micromouse competition was made popular in the late 1970's, and remains popular in the UK, U.S., Japan, Singapore, India and South Korea. National and international competitions bring together the top micromouse builders, including Kato-san, Ng Beng Kiat, and Nakashima-san. The current world record for the standard micromouse competition is 3.921 seconds, held by Ng Beng Kiat [1].

The competition motivates research in the larger field of Autonomous Mobile Robots (AMR), which has numerous household, military, and commercial applications. Robotic locomotion, sensing, and intelligence systems are not yet advanced or modular enough to perform arbitrary household tasks, nor can current systems learn complex new tasks that require physical manipulation in addition to computation. However, advances in the field of AMR have led to valuable applications for specific tasks, such as Roomba®, the automated vacuum cleaner, Wheelbarrow, the famous bomb defusing robot, and of course Opportunity and Spirit, NASA's Mars exploration rovers.

This approach to the Micromouse challenge will use rapid prototyping to iteratively improve the mouse design, making small but significant improvements with each new prototype. After implementing a minimal but functional prototype, specific issues within the areas of sensing, locomotion, and maze solving algorithms will be addressed in parallel, using a physical and simulated testing environment.

2 Competition Specifications

The rules for the IEEE Micromouse Competition, as specified in [2], specify constraints that submissions must adhere to, as well as how the competition will run. Each submission must be a self-contained robot no larger than $25cm$ in either length or width. The standard maze is a 16 x 16 grid composed of $18cm$ x $18cm$ squares, with $5cm$ high walls each with a thickness of $1.2cm \pm 5\%$. These specifications are detailed in Table 1:

Specification	**Value**
Maximum Robot Width	$25cm$
Maximum Robot Length	$25cm$
Maze Wall Height	$5cm$
Maze Wall Thickness	$1.2cm \pm 5\%$
Maze Square Size	$18cm$ x $18cm$
Maze Grid Size	16 x 16 squares

Table 1: IEEE Micromouse Competition Specifications

Starting in one of the four corners, each robot attempts to solve the maze alone, with the goal of reaching the four center-most squares, which have no walls separating them and a $20cm$ high post at their center. As the robot traverses the maze, it cannot jump over or damage walls, nor can any of its parts detach from the main chassis.

Each team has a total of 10 minutes to access the maze, during which time it may attempt as many runs as possible. A successful run entails the robot starting in the designated corner of the maze and maneuvering to the destination (center) square. The run with the shortest time is the robot's official time, and the shortest overall time is awarded first place. If no mice reach the destination square, the robots are ranked by the number of unique cells that each of them passes through without being touched by a team member. If a team member does touch the robot (for example in the event of a collision), a 30 second penalty is applied to the total time remaining when the mouse is placed back into the maze.

The standard Micromouse competition strategy is to first explore the maze until the optimal path to the center is discovered. Then, using that information, several attempts can be made to reach the center of the maze as quickly as possible, increasing the robot speed (and the risk of error) each time.

3 Design Considerations

The design elements that must be considered in order to construct a successful Micromouse are laid out in Figure 1. The hierarchy of design elements begins with the sensor systems and proceeds to the locomotion and control systems, because without a robust image of the robot's surroundings, a control system cannot make informed decisions. However, the design process itself is more easily realized in the reverse order - once a physical prototype for the chassis and locomotion system is constructed, sensors can be added and tested, followed by the full control system that accounts for sensor readings. These systems are described in the sequence of the design process that was implemented, with attention given to previously implemented designs.

3.1 Locomotion

Designing a robust locomotion system is the highest priority for making an effective Micromouse. Based on observing previous competition submissions [3] [4], it is clear that a precise system for physically navigating the terrain can prevent the robot from getting stuck in the maze, one of the most common problems in this competition. Several design elements must be considered in order to choose a locomotion system. These include the size and quantity of wheels; the overall structure of the locomotion system; the size, torque, and efficiency of motors; and the types of feedback mechanisms that will guide motion.

3.1.1 Wheels

Although a wheeled locomotion system is not a requirement for the Micromouse competition, almost all submissions use such a system because it is easier to implement and more effective than systems that employ legs, treads or other systems [5]. Three main types of wheels are available: the standard wheel, the caster ball, which can roll in any direction, but cannot be powered, and the omni wheel, which can actually be powered and move in nearly any direction. The choice of wheel type will depend on the choice of locomotion structure. However, certain characteristics of wheels are important regardless of this choice. The drive wheels must be coated in a rubbery material that minimizes the risk of slippage. In addition, the drive wheels must have some degree of built-in suspension in order to maintain a straight course while navigating the maze. Finally, the diameter of the wheels will be an important trade-off: larger wheels will allow for greater torque, but will sacrifice precision of motion, and may increase the risk of slippage [6].

The Tamiya 70111 Sports Tire Set was chosen for the first iteration of the robot. This includes wheels with a 56 mm diameter and 25 mm width that have tacky rubber tires with treads. These tires are very unlikely to slip, and can be compressed by approximately 5 mm, meaning that they will be able to negotiate cracks, ledges, and other artifacts on the maze floor easily.

3.1.2 Locomotion System Structure

The structure of ground robot locomotion systems varies greatly, but can generally be divided into two groups: legged and wheeled locomotion. Given the size and time constraints of this challenge, wheeled locomotion is the clear choice for a fast, precise Micromouse. However, several configurations of wheeled locomotion exist, each of which has distinct advantages. Three different configurations are candidates for this design: a three wheel design composed of omni wheels allowing synchronous drive, as in Figure 4a, a three wheel design composed of two standard wheels and a caster ball in a tricycle configuration, as in Figure 4b, and a traditional design composed of two rows of standard wheels, as in Figure 4c.

The advantage of the omni wheel configuration is that it allows holonomic motion (the ability to move freely in all degrees of freedom), which can prevent the robot from getting stuck in the maze. However, this configuration is also difficult to control, and requires additional sensors to keep track of the orientation of the robot.

The second configuration is the easiest to control, because each of the two wheels are driven by separate motors, and the caster ball rotates freely, eliminating problems of synchronizing three or more wheels. The disadvantage of this structure is the risk of slippage.

The most competitive Micromouse submissions all implement the standard four wheel configuration, which is difficult to control and can cause robots to get stuck in the maze, but allows the fastest movement and precision if implemented well. The first prototype will implement a four-wheel drive system using gears to drive the two wheels on either side of the robot with one motor each. Four-wheel drive is the standard among the most competitive robots, as it provides superior acceleration and greater control, as long as the control system accounts for slippage and drift appropriately. [6] [7].

3.1.3 Motors

The choice of motor for the drive train of the Micromouse entails a trade-off among drive power, power consumption, and weight. Drive power is an especially important consideration for the Micromouse competition, because the robot must be able to reach the destination square quickly in order to be competitive. However, as drive power increases, power consumption increases as well. In order to minimize the weight of the robot, the weight of each component, as well as the weight of batteries, must be minimized. Given these considerations, three types of motors are commonly used in AMR applications: stepper motors, servo motors, and DC motors.

Stepper motors are brush-less DC motors that divide a full rotation into a series of discrete "steps" that are controlled by the polarity of a number of electromagnets around the perimeter of the main rotor. The advantage of using a stepper motor for this project is that stepper motors are very precise, and would eliminate the need for encoders or other feedback sensors. However, these motors are significantly larger, heavier, and more power-hungry than servo or DC motors with equivalent torques, as can be seen in table 2.

Servo motors are similar to stepper motors in that they have a simple built-in control system, however they use potentiometers and a series of gears rather than electromagnets. This allows servo motors to operate much faster than stepper motors, as can be seen below.

DC motors are used in the most competitive Micromouse robots, because they provide by far the highest power-to-weight ratio. A major challenge associated with these motors, however, is that a control system must be implemented to track the drive provided by each motor. This control system usually consists of a pair of IR encoders, as well as an extra layer of computation on the micro-controller to account for encoder inputs.

Specification	RB-Soy-20 Stepper Motor	Hitec HS-55 Servo Motor	Pololu DC Micro Gear-motor
Torque [oz-in]	13	18	15
Weight [g]	140	8	10
Operating Voltage [V]	4.5	5	6
Free Run Current [mA]	670	150	100
Stall Current [mA]	670	150	1600
Free Run RPM	-	61	630

Table 2: Comparison of Common AMR Motors

Despite the challenges associated with DC motors, they are the only choice for a competitive Micromouse robot because of the high power-to-weight ratio they provide. The Pololu DC Micro Gear-motor was chosen for the first iteration of Megamouse. With this choice of motor, the 56 mm Tamiya tires, and a 1:2 gear ratio, Megamouse will be able to achieve a maximum speed of 0.92 m/s. This is a highly competitive speed for undergraduate Micromouse competitions. Moreover, using PWM drive signals with reduced duty cycles, this speed can easily be reduced for turns and other precarious maneuvers.

3.1.4 Initial Prototype

The drive train of the initial prototype was designed around the 56 mm Tamiya Sports Tire Set and Pololu 50:1 Micro Gear-motors, as can be seen in Figure 6. In order to simplify the control system for the drive train, the two wheels on each side of the robot are to be driven by a single motor through a set of plastic gears. The smaller gear has 50 teeth and a pitch diameter of 20 mm, and is attached to the motor shaft with a set screw. The larger gears have 100 teeth and pitch diameters of 40 mm, and are attached directly to the front and rear shafts of the drive train. The shafts are 5 mm diameter brushed steel, and will attach to the Tamiya wheels via custom aluminium hubs, as can be seen in Figure 5a. Finally, the each of the four disjoint shafts will be held to the chassis by aluminium pillow blocks with press fit bearings, as can be seen in Figure 5b. This drive train assembly will provide a high degree of stability and durability, and will be tested thoroughly.

3.2 Chassis

The chassis of the Micromouse must be designed to balance the weight of the power supply, motors, micro-controller, and sensors while preventing the robot from snagging on corners or toppling during turns. In addition, the chassis must account for potentially uneven terrain by having a slightly flexible body. Competitive robots use the same custom PCBs on which their sensors, micro-controller, and other components are mounted as the chassis of the robot, keeping the center of mass low, and maintaining a narrow profile to be able to take wider turns at high speeds. However, acrylic and ABS plastic are also common materials for Micromouse chassis, as they can be cut with common shop tools. The following table compares the mechanical properties of popular material choices.

Material Property	**Acrylic**	**ABS plastic**	**PCB**
Tensile Strength (PSI)	10,000	4,100	45,000
Flexural Modulus (PSI)	480,000	304,000	65,000
Izod Impact ($ft \cdot lb/in$)	0.4	7.7	8
Density (g/cm^3)	1.17	1.06	1.85

Table 3: Comparison of Chassis Materials

The comparison in Table 3 shows that acrylic has a higher tensile strength and flexural strength than ABS, but that it has a lower Izod Impact than ABS. This suggests that acrylic is a stronger material, but that it is brittle and can crack easily. As can be seen in Figure 7, the chassis design requires numerous sharp cuts. Acrylic is likely to crack while fastening components to these areas of the chassis, while ABS plastic is a softer material, and can accommodate the pressure. Compared to acrylic, ABS plastic is a better choice for this design, as it provides the stability required for a stable chassis as well as the flexibility to handle potentially uneven terrain.

For the first prototype, a piece of ABS plastic 13cm x 15cm will be laser cut to the design in Figure 7. Eventually, a custom PCB can be printed to obtain similar specifications and eliminate stray wiring on the robot. Although common PCB materials have nearly double the density of ABS, printing a custom PCB eliminates the weight of a separate micro-controller board.

3.3 Power Source

Micromouse submissions must be free standing, and therefore must be battery powered. Three common options for rechargeable batteries were compared, as can be seen in Table 4:

Specification	**NiCd**	**NiMH**	**Li-Ion**
Relative Power/Weight Ratio	1	1.4	3
Memory Effect	Noticeable	Little Effect	No effect
Self-Discharge (charge/mo)	20%	30%	3%

Table 4: Battery Comparison

Based on these specifications, it is apparent that a lithium-ion battery is the best choice for the Micromouse due to its greater Power/Weight Ratio and low self-discharge rate. An additional advantage is that Li-Ion batteries do not experience the memory effect. The memory effect is the process by which the capacity of a battery is reduced by repeatedly charging the battery before it has been completely discharged. Megamouse will be tested rigorously, and having a battery susceptible to the memory effect will greatly hinder performance.

For the first prototype, the Arduino Uno, two Micrometal gearmotors, and three IR distance sensors will be the only significant power drains. Each of these sets of components can be powered with a 5 V source, and draw up to 50 mA, 2×100 mA, and 3×30 mA respectively. This amounts to a total current of 340 mA. A pair of 3.7 V Lithium Polymer batteries, each with a capacity of 3000 mAh, have been purchased for the first prototype. Since the team is only allowed to access the maze for 10 minutes, two 1000 mAh batteries with of the same voltage may be used to reduce the weight of the robot during the actual competition.

3.4 Sensors

Sensor systems implemented on the Micromouse have the primary purpose of detecting the walls of the maze, and a secondary purpose of tracking the orientation and location of the robot as it moves. These sensors must detect both walls in front of the robot, as well as gaps in walls to either side of the robot as it navigates the maze. Three important parameters must be considered when designing such a system: speed, coverage, and precision, as can be seen in Figure 2. The system must have a high enough sampling rate to prevent the robot from making movement decisions based on incorrect positioning data. The number of sensors and the angles at which they are oriented must cover a wide enough area around the robot to detect walls in the periphery. The sensors must also be powerful enough to minimize the detector error rate (both false alarms and misses) by minimizing SNR. These trade-offs must be addressed as the sensor system for this project is designed [8].

3.4.1 Distance Sensors

Three types of sensors are commonly used for AMR positioning systems: infrared (IR) sensors, ultrasonic sensors, and simple CCD cameras. Ultrasonic sensors are common in AMR, especially in applications with unknown environments, because they are cheap, easy to implement, and are not subject to the high noise conditions that IR sensors are. However, they are not generally as reliable or precise (typically $2-5cm$) as IR sensors. Vision systems, composed of simple cameras, are also not precise enough for this project (typically $5cm$), and require significant processing power to do effective pattern recognition [6] [8].

IR sensors, consisting of an IR LED and an IR sensitive photo-transistor or photo-diode, are the most common among the top Micromouse competitors. The advantage of using IR sensors is that they are cheap, relatively low power, and precise. In addition, the fact that the walls of the maze are white while the floor is black adds to the effectiveness of sensing in the IR range, as IR light is readily absorbed by black paint and reflected by white paint. The first prototype Micromouse will implement simple, off-the-shelf IR sensors, such as the Sharp GP2Y0A21YK IR distance sensor, for the purpose of quickly implementing a design that can be tested. This sensor draws $10mA$ from a 5V source, and operates in the range of $2-10cm$ [9]. However, additional iterations of the prototype will implement higher power LEDs with bandpass filters and a high sampling rate to optimize the three design trade-offs [8]. In order to conserve power, the higher power LEDs can be pulsed without sacrificing SNR. In addition, this scheme, combined with bandpass filters can reduce the effect of noise from ambient sources of IR light, and the high sampling rate can allow the motion control algorithm to make informed decisions on a faster cycle.

3.4.2 Sensor Calibration and Signal Postprocessing

Each Sharp IR distance sensor requires individual calibration due to inherent differences introduced during the manufacturing process. The manufacturer specifications provide the following function to map output voltage to distance:

$$distance = 12343.85 \times {V_{out}}^{-1.15} \tag{1}$$

However, three such sensors were tested, and proved to have significantly different sensor voltages for the same distance. Initial calibration test results showed that if equation 1 is used for two different Sharp IR sensors, there can be errors of 2-3cm when approximating the distance of an object. Since the Micromouse is confined to a small space, an error of a few centimeters could result in a collision with a wall or corner.

In addition to calibrating sensors, sensor data must be post-processed to ensure consistent measurements are obtained. This requires filtering high frequency noise by using a low pass filter. The most simple and space efficient filter to implement is the Exponentially Weighted Moving Average (EWMA) whose output, Y_t, is the weighted average of the current input, X_t and each of the previous inputs exponentially decreasing weight:

$$Y_t = \alpha \sum_{i=0}^{N} (1-\alpha)^i X_{t-i} \tag{2}$$

$$Y_t = \alpha X_t + (1-\alpha) Y_{t-1} \tag{3}$$

Equation 2 requires that all previous data points be stored in memory. However, the filter can be rewritten recursively, as in equation 3. This formulation requires only the previous result to be stored, and can save computational resources as well as memory. In both cases, the value of α can range from 0 to 1, and determines the size of the filter's time window. A value of 1 corresponds to a window of just one point (all previous inputs are ignored,) while a value of 0 weights all observed inputs equally. Thus, a lower value of alpha effectively increases the window of the filter, and increases the SNR of the output at the cost of decreasing the response time of the sensor system. Choosing the correct value of α will depend on the amount of noise and latency the decision algorithm can tolerate.

3.4.3 Detection Schemes

The role of sensors in the context of the Micromouse competition is somewhat simplified, because Micromouse robots need only to detect walls rather than measure how far away walls are. As a result, a reasonable detection scheme to begin with is received signal strength intensity (RSSI). This scheme involves establishing a voltage threshold for detection based on a calibration algorithm that can be run as soon as the robot is placed into the maze. The threshold can automatically be adjusted based on lighting conditions at the time of competition. A more involved way of detecting walls is differential phase shift of arrival (DPSA), which can be seen in Figure 3. DPSA begins with a sinusoidal signal in the MHz range being modulated through the IR emitter. At the receiver end, the signal is passed through a high pass filter to remove noise, passed through an I/Q demodulator, and finally differentiated to find the phase of the real and imaginary channels. The difference in phase, combined with the intensity at the receiver, can be used to detect walls [10].

3.5 Control System

The Micromouse control system must translate sensor readings and knowledge of the maze into a decision of where the robot should move next. The control system must then drive motors to to move the robot, accounting for slippage and drift by integrating encoder and gyroscope readings. These needs can be accommodated by a combination of a relatively inexpensive micro-controller and several canonical AMR control algorithms.

3.5.1 Micro-controller

The micro-controller used to drive motors and process sensor data must be powerful enough to handle high data rates. Sensors may be sampled at a rate between 0.2 and 1 kHz, and the microprocessor must be capable of reading the sensor values (performing an A/D conversion), deciding where to move based on this information and on the location in the maze, and applying signals to each drive motor, all within a short time cycle in the range of 10's of milliseconds. In addition, solving the maze requires significant processing power and memory in order to represent the maze as a graph and quickly apply decision algorithms.

In order to accommodate these processing needs, a micro-controller such as the STMicroelectronics STM32F103RET6 is commonly used in Micromouse competitions. The specifications for this processor can be seen in Table 5: In addition to using (relatively) high powered processors, the most competitive Micromouse robots use custom PCBs to connect the processor to sensors, motors, and other system elements. For the first iteration of the robot, a simple Arduino Uno, with an ATmega38 processor - a simple, well documented micro-controller - can be used to test the functionality of other components. The ATmega38 specifications can also be seen in the table below. After testing the sensor and motion systems with the Arduino, the processing requirements can then be reassessed, and a more appropriate micro-controller can be purchased. Since most of the code will be written in C++, porting to the new controller should be simple.

Specification	**STM32F103RET6**	**ATmega38**
Data Bus	32 bit	16 bit
Clock Rate	45 - 72 MHz	16 MHz
A/D Converters	12 bit	10 bit
D/A Converters	12 bit	10 bit
RAM	64 KB	2 KB
Program Memory	512 KB	32 KB
Supply Voltage	5 V	5 V
Power Consumption	125 - 250 mW	100 - 150 mW
Analog Ports	16	6
Digital Ports	45	14

Table 5: Micro-controller Specifications [11], [12]

3.5.2 Processing Sensor Data

An alternate architecture, involving two processors, may also be used to accommodate the data rates involved with sensor readings. In this scheme, a slave processor can be tasked with sampling sensors at around 1 kHz and reporting post-processed sensor readings to a master processor. For five sensors (three distance sensors and two encoders), this sampling rate translates to a data rate of 160 kb/s, assuming each reading is stored as a C `int`. Simple post-processing, such as a weighted average low pass filter with a 10 sample window, as explained in Section 3.4.2, can also be applied on the slave processor. The master processor can then read aggregated sensor readings at around 16 kb/s, a much lower data rate. This scheme will allow the master processor to allocate more computational resources to controlling locomotion and applying maze solving algorithms. However, a second processor will increase the power consumption of the control system, and take up a significant amount of space on the chassis. These trade-offs will be re-evaluated after the first, simple control system is implemented.

3.5.3 Motor Control Scheme

The standard mechanism for providing current drive to one or more motors in robotics is an H-bridge. An H-bridge is a switching circuit that allows for current to be applied across a load (in this case a motor) in either direction. For the Megamouse drive train, the Texas Instruments L293D Quad Half-H Driver chip was chosen. The specifications of this chip-set can be seen in Table 6. The drive train configuration of the first prototype, as outlined in section 3.1.4, can be expected to require a drive current of 100 mA for each of the two motors. The L293D can easily satisfy this requirement.

Specification	Value
Channel Output Current	600 mA
Peak Output Current	1200 mA
Supply Voltage Range	4.5 - 36 V

Table 6: L293D Quad Half-H Driver Specifications [13]

3.5.4 Kinematic Model

A kinematic model will be implemented in this project to keep track of the direction of motion and distance traveled by the robot travels as it navigates the maze. Based on the choice of locomotion structure, a set of kinematic equations will be derived to relate the signals applied to the driving motors to the motion of the robot. These equations will account for the dimensions of the robot, as well as the size of maze squares, and will be abstracted away from the maze solving algorithm (i.e. the code for the maze solving algorithm will be able to call functions like *turnLeft* and *turnRight*).

A common kinematic model for AMR is an odometry-based model that accounts for error using probabilistic sampling. An odometry-based model implies the use of encoders to measure the movement of each wheel rather than assuming a direct relationship between the power supplied to motors and the movement of the robot. Compared to the latter model, measuring the amount of rotation of each wheel directly can lead to much better estimates of motion. The following equations describe the change in position of the robot between two samples of encoder readings in the global coordinate space. [14]:

$$\Delta s = \frac{\Delta r - \Delta l}{d} \tag{4}$$

$$\Delta \theta = \frac{\Delta r + \Delta l}{2} \tag{5}$$

$$x' = x + \Delta s \cos \theta \tag{6}$$

$$y' = y + \Delta s \sin \theta \tag{7}$$

$$\theta' = \theta + \Delta \theta \tag{8}$$

In this model, Δl is the translation of the robot, Δr is the counter-clockwise rotation of the right wheel relative to the left wheel, $\Delta\theta$ is the angle of this rotation, and d is the distance between the left and right wheels. A depiction of these parameters can be seen in Figure 8. Once this model is implemented and tested, an additional parameter can be added to increase its accuracy. The output of each encoder will naturally be noisy, but this noise can be modeled as a random variable. In the most generic case, the noise that affects each encoder, k, can be modeled as Gaussian, and the parameters μ_k and σ_k can be estimated using maximum likelyhood (ML) estimation during testing. Then, a random sample from $\mathcal{N}(\mu_k, \sigma_k)$ can be added to each reading reading from the kth encoder. This model can allow the control system to train on a more accurate kinematic model, which can improve the overall decision making algorithm. In order to effectively solve the maze, a locomotion control system that is precise to $2 - 3mm$ will be necessary [6] [7], and each of these additions will attempt to bring the precision closer to that value without sacrificing speed.

3.6 Maze Solving Algorithms

The ability of the Micromouse to find the shortest path to the center of the maze is a critical component of the competition. In fact, the problem of solving an arbitrary maze is the simplest component of the Micromouse, as this problem has been solved by several algorithms. The simplest solution to solving a maze is to use a random walk algorithm, which simply makes a random decision at each junction. This algorithm will eventually find the solution to the maze, but can take an infinite amount of time. Clearly, this is not an efficient solution, but may be an easily implemented step during testing. Another simple algorithm is simply follow the right or left wall of the maze, however Micromouse mazes are designed specifically so that this algorithm will not lead to the solution [2]. Two advanced algorithms are used commonly in Micromouse competitions - the Flood Fill and Shortest Path algorithms. These algorithms require memory and additional processing power, but these can easily be accommodated with today's technology.

3.6.1 Flood-fill Algorithm

The flood fill algorithm is actually a graphics method used to fill in connected, similarly colored areas. It can also be used to solve mazes by treating each square of the maze as a pixel. Under this assumption, four cases exist:

1. All four boundary squares are filled - the maze is solved
2. Three of the boundary squares are filled - only one path leads out of the current square
3. Two or less boundary squares are filled - choose one of the paths leading out of the square at random, and mark the current square (in memory) as a square that has been visited

Using these scenarios, a recursive function can be developed to solve a maze by tracing out more and more of the maze until the solution (case 1) is reached. This algorithm is guaranteed to find a solution to the maze, but it may not be the optimal solution.

3.6.2 Shortest Path Algorithm

The Shortest Path algorithm is another solution to the maze problem, and can be implemented with a breadth-first search using the A* algorithm, a common artificial intelligence algorithm. The basic idea of this method is to visit squares of the maze in increasing distance from the starting square until the goal square is reached. Squares can be tracked using a memory structure called a queue.

This algorithm is guaranteed to find the shortest solution to the goal state. Once this solution is found, the robot can return to the starting square, resetting its time, and make a second trip straight to the goal square. This technique of making repeated trips to the goal square is implemented by almost all competitive Micromouse robots, but requires that the robot be fast and have a low probability of getting stuck, because the robot must traverse many paths in the maze repeatedly [15].

The final choice for a maze solving algorithm can be developed during the later stages of the project, after the robot is optimized to turn and move as fast as possible with a low probability of getting caught in the maze. In the mean time, simple algorithms such as random walk may be used to test other systems.

4 Design Process

4.1 Workflow

The design process that is followed in developing a Micromouse with little robotics experience will be especially important to the quality of the final robot. During the early stages of design and testing, as many configurations as possible will be attempted in order to cover a breadth of possibilities for the design of the robot. Then, as design choices are made, a narrower set of components, configurations, and algorithms will be optimized.

4.2 Testing

Testing the functionality of the robot properly will be essential to creating a competitive submission. Several components of the design can be tested alone - the sensing, power, and micro-controller systems can each be benchmarked individually. Sensors can be mounted on a piece of cardboard or plastic in various orientations, and tested at different angles and distances from walls. The I/V characteristics of batteries can be measured against equivalent circuits at idle and maximum loads. The micro-processor can also be benchmarked using simulated sensor data.

However, testing the locomotion system will require the wheels, motors, and chassis to be in place, and the combined system can only be tested after each component has been chosen. A physical maze specifications as close as possible to the competition maze can be constructed to evaluate how the robot will perform as an overall system. During the development phase, software simulations can also be done in C++ to test the efficiency of maze solving algorithms using randomly generated mazes. The average solving time across numerous iterations can be used as a metric to compare algorithms as well as robots themselves.

5 Conclusion

The IEEE Micromouse competition is an engineering challenge that requires the application of precise hardware design as well as efficient software algorithms. This project proposes a design for a robot that solves the competition maze as quickly as possible by integrating a durable and powerful drive train, efficient control systems, and precise sensor systems. The prototype detailed in this paper will be constructed, tested rigorously, and improved over several iterations. The designs proposed by this project may allow advances in AMR and other robotics applications.

6 Acknowledgements

The authors would like to thank several individuals for their advice and support of this project. In particular, Toby Cumberbatch, the advisor for this project, has helped maintain focus of the most crucial components of the project. In addition, Sinisa Janjusevic, Jeff Hackner, Jonothan Ostrander, Daniel Baamonde, and Vasily Kuksenkov have provided invaluable advice on the design of Megamouse.

7 Bibliography

[1] Evan Ackerman, "Meet the New World's Fastest Micromouse Robot," Nov. 2011. [Online; accessed 12-22-2013].

[2] IEEE Region 1 Student Conference, *Micromouse Competition Rules*, 2013.

[3] Jake Edgar, "Micromouse - 1st Place IEEE Region 1 Competition 2013." Youtube, Aug. 2013. `http://www.youtube.com/watch?v=hAHw0VYsi1c` [accessed 11-22-2013].

[4] Peter Boothe, "IEEE Region 1 MicroMouse competition 2012 - winning team." Youtube, Mar. 2012. `http://www.youtube.com/watch?v=DuG-WF9yT5c` [accessed 11-22-2013].

[5] Sven Bottcher, "Principles of Robot Locomotion." Seminar on Human Robot Interaction SS2006, Univ. of Dortmund, 2006A.

[6] Birmingham City University, "Micromouse Guide," 2013. [Online; accessed 12-24-2013].

[7] J. Xiao, "Advanced Mobile Robotics: Locomotion/Motion Planning/Mapping ." Dept. of Electrical Engineering, CUNY City College. [Online; accessed 11-24-2013.

[8] E.M. Gorostiza et al., "Infrared Sensor System for Mobile-Robot Positioning in Intelligent Spaces," *Open Access: Sensors*, 2011.

[9] *GP2Y0A21YK:Optoelectronic Device*, 2005.

[10] H. I. Intersil Americas Inc., Witter D.W., "Distance sensing by iq domain differentiation of time of flight (tof) measurements," Aug 2011.

[11] "STMicroelectronics STM32F103RET6." Mouser Electronics Inc. `http://mouser.com` [Online; accessed 12-10-2013]".

[12] Arduino, "Specs Compare," 2013. [Online; accessed 12-10-2013].

[13] "L293, L293D Quadruple Half-H Drivers." [Online; accessed 12-23-2013].

[14] Kurt Konolige, "Robot Motion: Probabilistic Model; Sampling and Gaussian Implementations; Markov Localization," 2001. SRI International.

[15] W. D. Pullen, "Maze classification," Apr. 2013. [Online; accessed 12-10-2013].

[16] Luzhou Ye, "Green Ye's Personal Website." `http://www.greenye.net/` [Online; accessed 12-14-2013].

[17] "OMNI-WHEEL ROBOT - FUZZY." Society of Robots. http://societyofrobots.com [Online; accessed 12-10-2013].

[18] "DFRobot 2WD Mobile Platform for Arduino." Robotshop Distribution Inc. http://robotshop.com [Online; accessed 12-14-2013].

8 Appendix 1: Diagrams

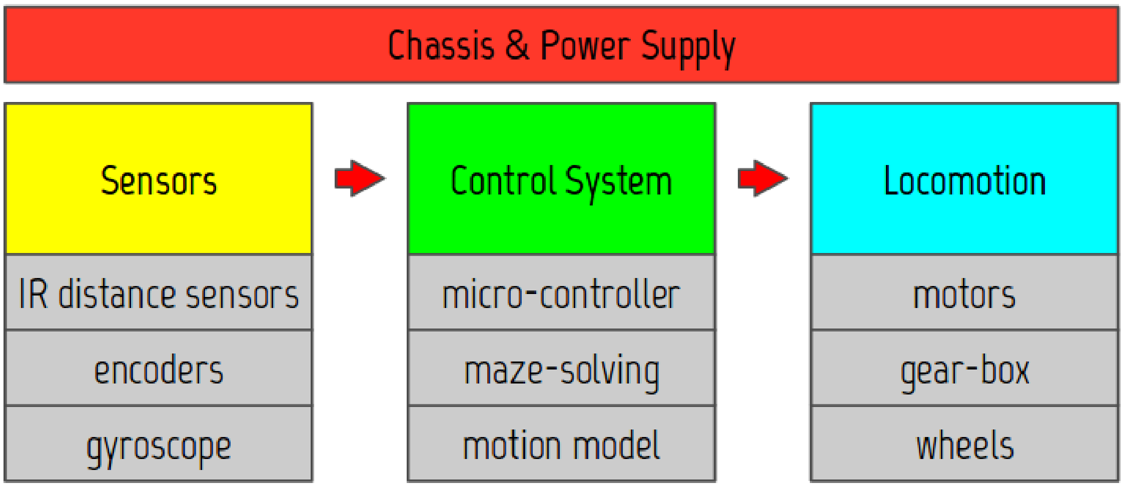


Figure 1: Design Considerations

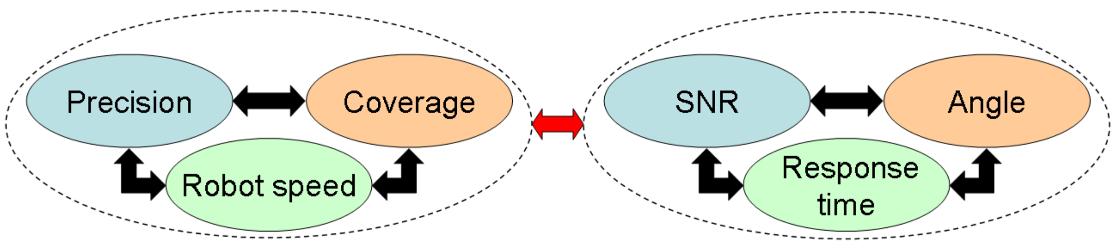


Figure 2: Sensor Trade-offs [16]

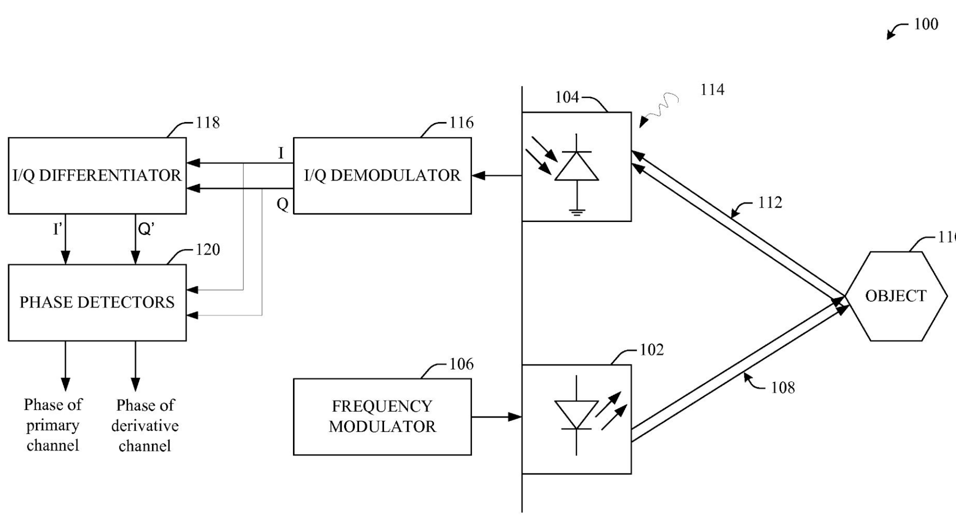
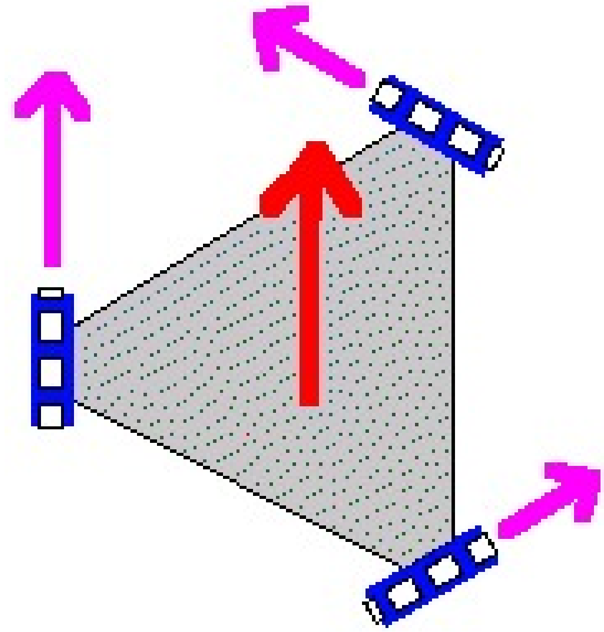
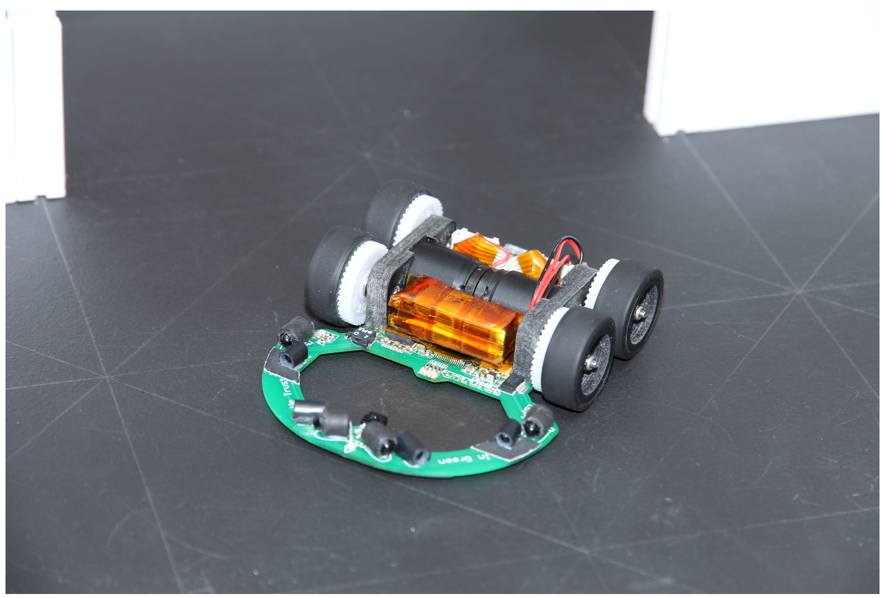


Figure 3: Differential Phase Shift of Arrival [10]



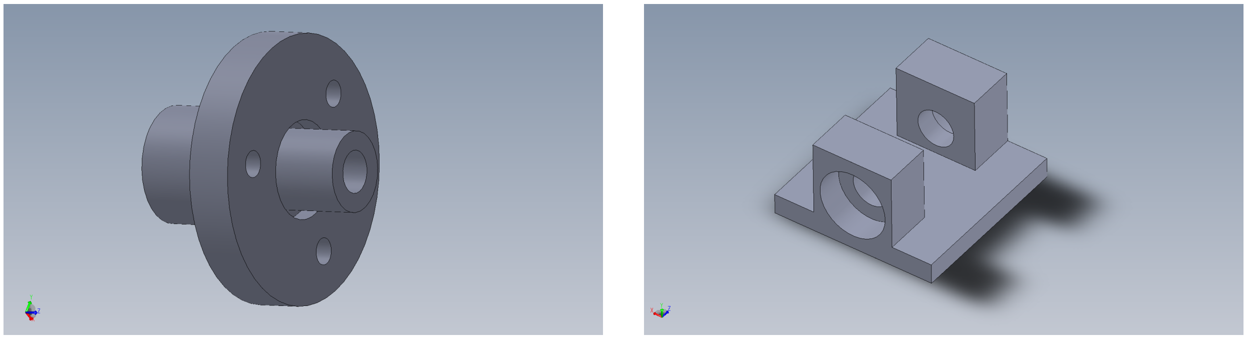
(a) Three Omni Wheel Configuration [17]

(b) Caster Ball Configuration [18]



(c) Standard Four Wheel Configuration [16]

Figure 4: Common AMR Locomotion Structures



(a) Wheel Hub (b) Pillow Block

Figure 5: Drive Train Components

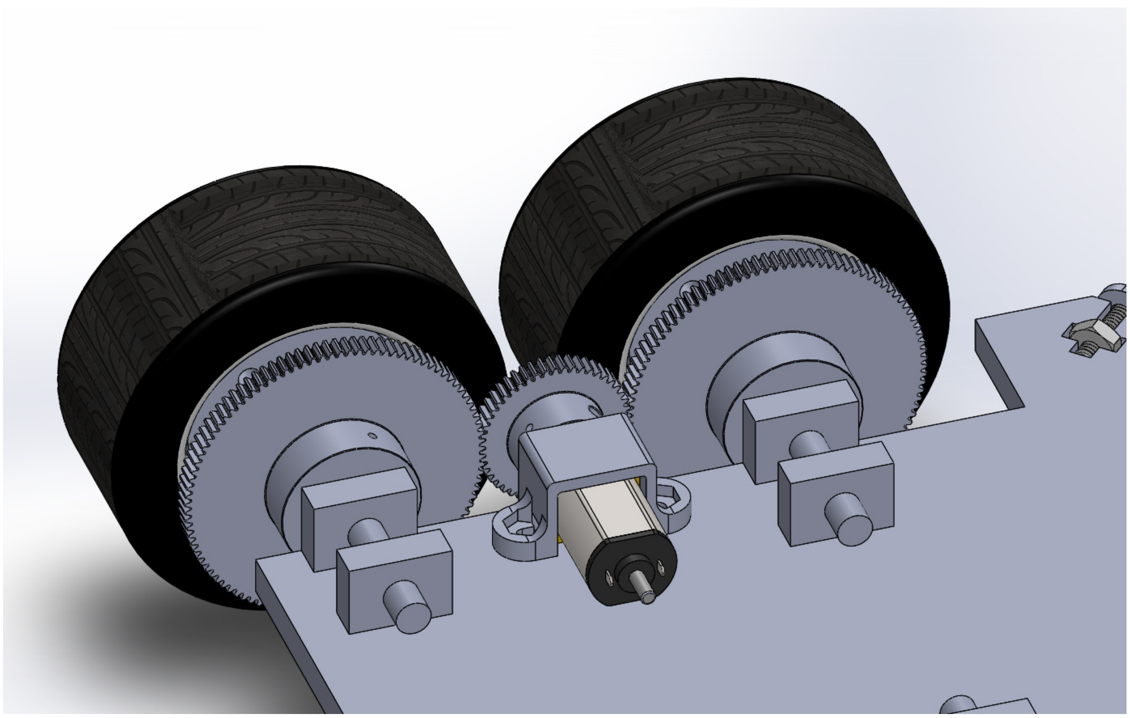


Figure 6: Geared Wheel Design

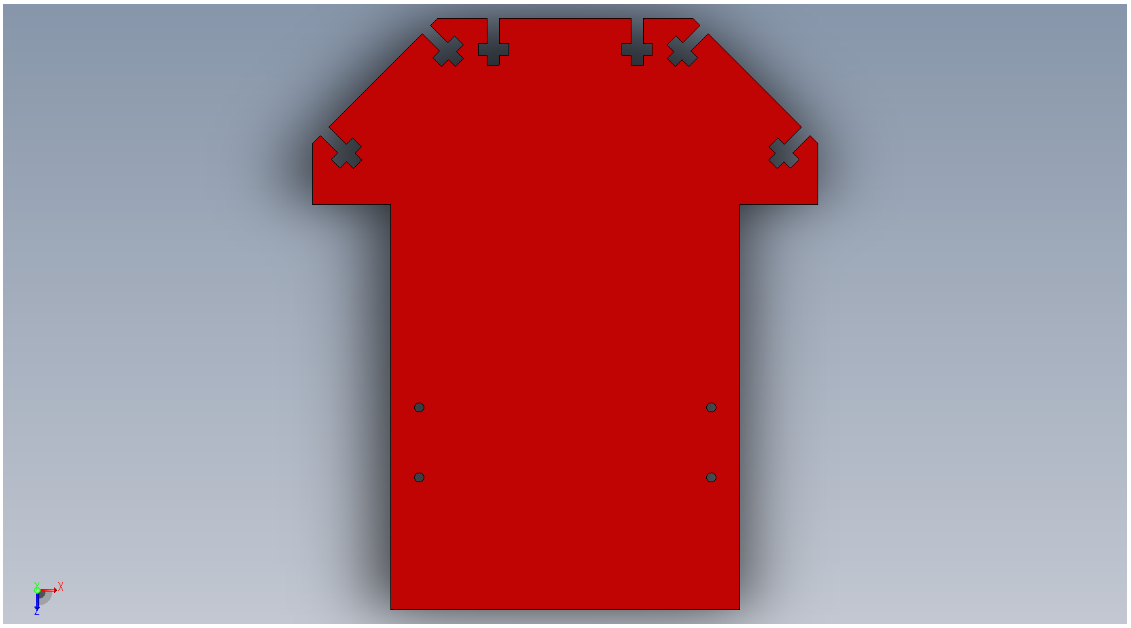


Figure 7: Chassis Design

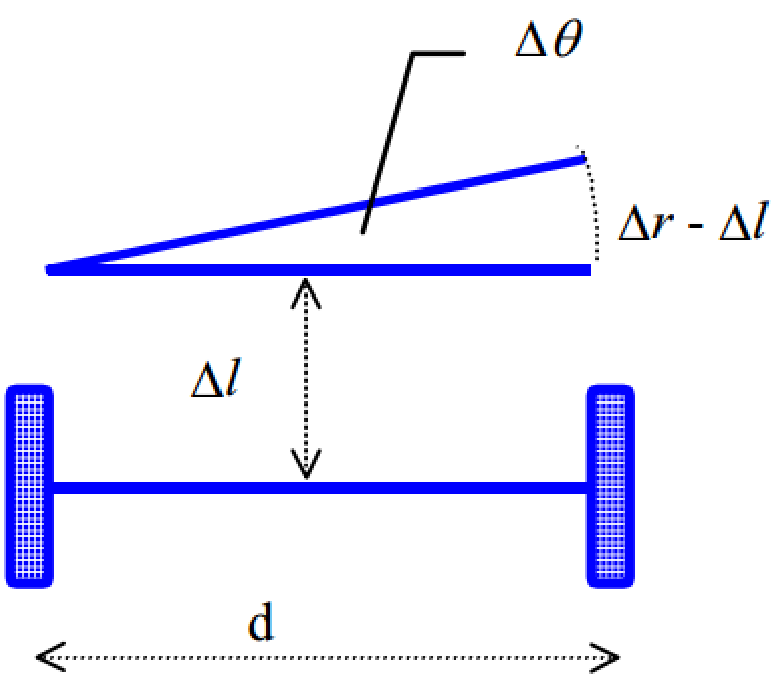


Figure 8: Kinematic Motion Model Parameters [14]

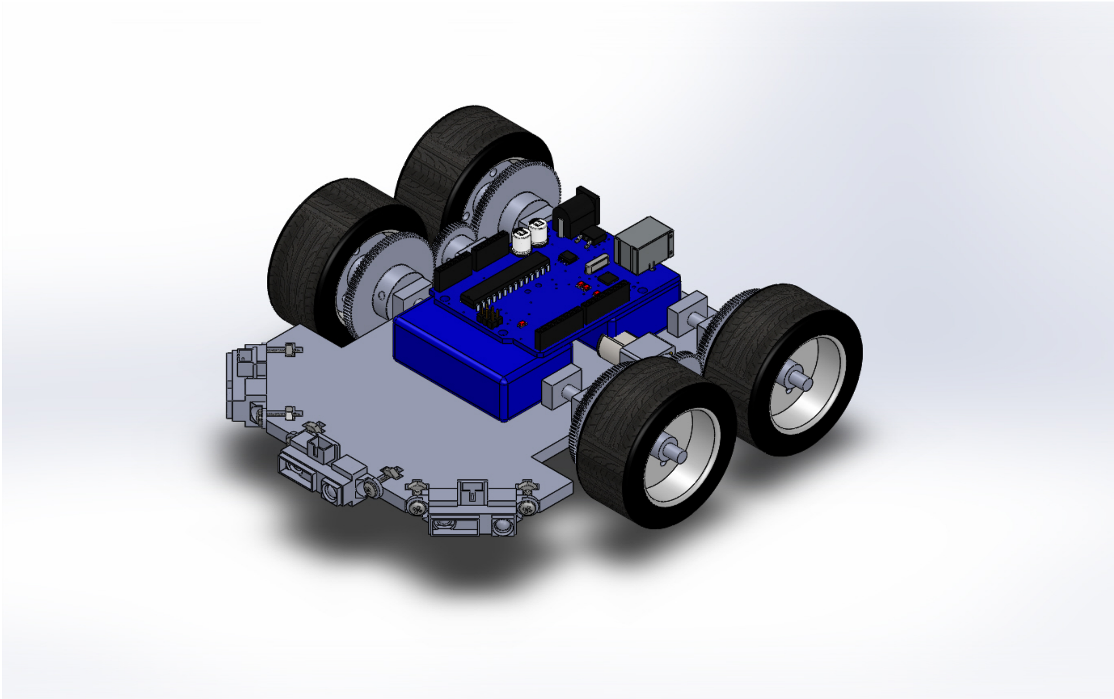


Figure 9: Full Megamouse Design