Team: 209 – Micromouse
Expo Presentation

<u>Drivetrain Team</u>

Daxton Barzee (EE) & Carolina Gonzalez (ME)

<u>Mapping Team</u>

Alannys Argandona (EE), Jazmyn Ramirez (EE)

Faculty Advisor: Dr. Curtis Wang

December 6th, 2024

Team Members

Drivetrain Team

Mapping Team

Daxton Barzee

Carolina Gonzalez

Alannys Argandona

Jazmyn Ramirez

Agenda Overview

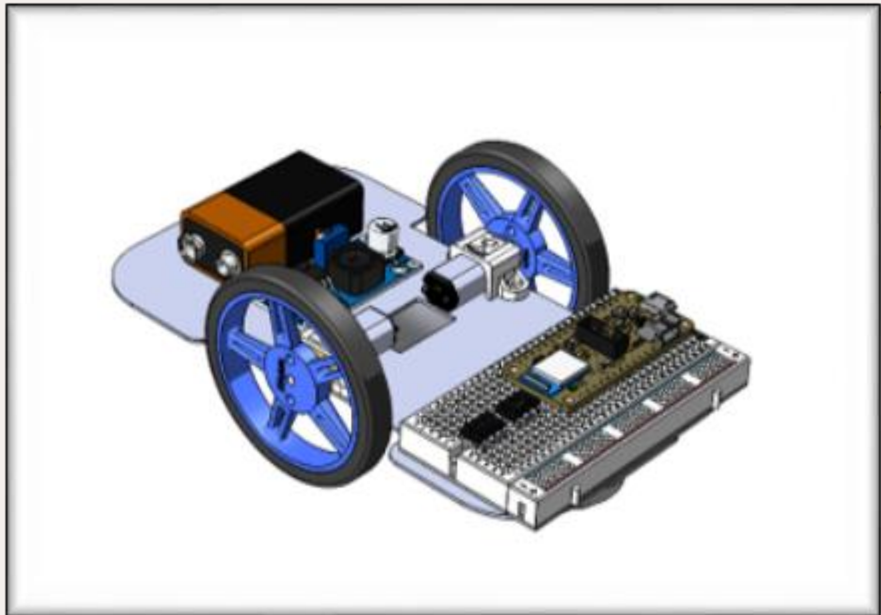
- Project Background
- Project Objective
- System Level Overview
- Drivetrain Design Overview
- Drivetrain testing
- Micromouse Prototype
- Mapping Design Overview
- Mapping Testing
- Maze Testing

Project Background

The IEEE *Micromouse* Competition consists of an annual event challenge in not only designing but also in building a small autonomous robotic mouse that must be capable in solving a maze without any external inputs.

Competition allows for engineering student to test their knowledge in different aspects such as robotics, control systems, machine learning as well more physical components like dynamics.

Project Objective



Micromouse Design

Objective

- ➢ To design and construct an autonomous mouse for the annual IEEE Competition

Team Composition

- ➢ The Mapping Sub-team
- ➢ The Drivetrain Sub-team

Project Management

Drivetrain Team

Sub Team Members	
Carolina Gonzalez	**Daxton Barzee**

Drivetrain Responsibilities

- Design and produce a device that facilitates movement and measurements
- Shall compare viable options for wheels, drivetrain, chassis, distance measurement sensors and path tracing methods for conceptual design.

Mapping Team

Sub Team Members	
Alannys Argandona	**Jazmyn Ramiez**

Mapping Team Responsibilities

- Providing evidence of sensors and data processing method
- Shall create the electrical and computational designs such as sensors and electronic circuits.

Work Breakdown Structure - Drivetrain

Task Team	Task Number	Task	Task Description	Start Date	Finish Date
Phase 1: Input and Output Testing	**1**			**9/2/2024**	**9/20/2024**
Drivetrain Team	1.2	Build all three different mechanical prototypes	In composing all of the mechanical properties such as the chassi and the wheels with the motors.	9/2/2024	9/6/2024
Drivetrain Team	1.3	Test/compare all three mechanical prototypes	Test all three variations and start building on the electrical components on selected chassi	9/8/2024	9/14/2024
Drivetrain Team	1.3b	Test electrical components on chassi	Test all of the bigger electrical components such as the microcontroller, distance sensors, and motors that will be later become perminate to the chassi	9/14/2024	9/20/2024
Drivetrain Team	1.4	Physical Simulation	Having a smaller mock maze completed to test all of the components working together.	9/14/2024	9/20/2024
MILESTONE 1	1.4b	Direction to Motor Output	Complete H-Bridge configuration and software motor control. Test different wheels for accuracy/efficiency.	9/15/2024	9/20/2024
Phase 2: Bridging Mechanical and Electrical Components	**2**			**9/15/2024**	**10/21/2024**
Mapping / Drivetrain Team	2.1	Prototype the Onboard Electrical and Computational Design	Compare the different sensors, circuits, and sensor data processing methods	9/26/2024	9/30/2024
Mapping / DriveTrain Team	2.2	Testing & troubleshooting physical and software architecture	Test for results on hardware and softwareDesigns and work on any issues	9/30/2024	10/4/2024
MILESTONE 2	2.3	Prototype with completed electrical components	Prototype with sensors, motors and working code to move micromouse	10/1/2024	10/11/2024
Drivetrain Team	2.4	Additional trial and errors testing	Create calibration curves and measure turn radius & other drivetrain parameters to solidify testing.	10/11/2024	10/21/2024
MILESTONE 3	3.1	Working Model with Software	Micromouse structure design completed with necessary modifications and working code to run a maze	10/21/2024	11/1/2024
Phase 3: Project Completion	**3**			**10/21/2024**	**11/22/204**
Drivetrain Team	3.1	Present Designs	Begin poster designs for EXPO	11/1/2024	11/8/2024
Mapping / Drivetrain Team	3.2	Optimization	Implement any final improvements in physical / software architecture	11/8/2024	11/15/2024
Mapping / Drivetrain Team	3.3	Feedback	Gain Feedback on Designs and finalize project	11/15/2024	11/22/2024

Work Breakdown Structure - Mapping

Task Team	Task Number	Task	Task Description	Start Date	Finish Date	Deliverable Type
	1	Phase 1: Testing				
Mapping Team	1.2	Create a electrical and computational design	This design will help us figure out how the micromouse will be able to know where it is in terms of the maze and how to move forward	9/2/2024	9/6/2024	Slides
Mapping Team	1.3	Test/compare electronic circuits	Test sensor circuits that would help the mapping of the micromouse	9/8/2024	9/14/2024	Slides
Mapping Team	1.3b	Test sensor data processing methods	Test sensor data processing methods	9/14/2024	9/20/2024	Slides
Mapping team	1.4	Physical Simulation	Create a physical simulation using a lego box, sensors and microcontroller.	9/14/2024	9/20/2024	Slides
MILESTONE 1	1.4b	Sensor Selection	Narrowing down the best sensors for our micromouse to move forward to design	9/14/2024	9/20/2024	Slides
	2	Phase 2: Prototype				
Mapping Team	2.1	Prototype the Electrical and Computational Design	Use the lego box to compare the different sensors, circuits, and sensor data processing methods	9/26/2024	9/30/2024	Slides/Prototype
Mapping Team/DriveTrain Team	2.2	Feedback	Gain Feedback on Designs and work on any issues	9/30/2024	10/4/2024	Slides
MILESTONE 2	2.3	Calculating sensor calibration and calibration curves	Calculating sensor calibration and calibration curves	10/1/2024	10/11/2024	Slides
Mapping Team	2.4	Additional trial and errors testing	Testing all electrical and mechanical components together to find best results	10/11/2024	10/21/2024	Slides
	3	Phase 3: Poster Presentation				
MILESTONE 3	3.1	Working structure design and code	Micromouse structure design completed with necessary modifications and working code to run a maze	10/21/2024	11/1/2024	Slides
Mapping Team	3.1	Present Designs	Begin poster designs for EXPO	11/1/2024	11/8/2024	Slides/Prototype
Mapping Team/DriveTrain Team	3.2	Identify Best Designs and Prepare	See which design works best with one another to be able to then continue preparing for EXPO	11/8/2024	11/15/2024	Slides/Prototype
Mapping Team/DriveTrain Team	3.3	Feedback	Gain Feedback on Designs and finalize project	11/15/2024	11/22/2024	Slides

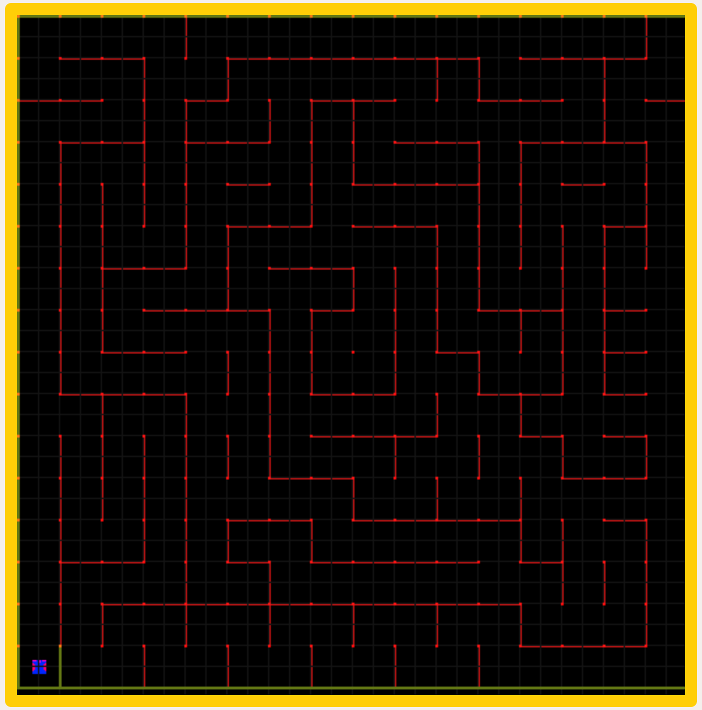
Bill of Materials – Drivetrain Team

Components List					
Iteam Number	Item Name	Price Per Unit	Quantity	Function	Total Price
1	30:1 Micro Metal Gearmotor MP 6V	$18.35	2	Allow for movement to occur within the Micromouse	$91.75
2	Pololu Wheel 32x7mm Pair - Black	$3.95	2	One of the wheel types that undergo testing to determine wether not it is the most efficient set for the Micromouse	$7.90
3	Pololu Wheel 40x7mm Pair - Black	$4.95	2	One of the wheel types that undergo testing to determine wether not it is the most efficient set for the Micromouse	$9.90
4	Pololu Wheel 60x8mm Pair - Blue	$5.75	2	One of the wheel types that undergo testing to determine wether not it is the most efficient set for the Micromouse	$11.50
5	Pololu 400- Point Breadboard with Mounting Holes	$2.95	1	Allows for prototyping and the testing of the electronic circuits without soldering. Eventually, it allows for the solidification of a schematic that can be transferred to. PCB	$2.95
6	Adafruit Feather nRF52840 Express	$24.95	1	Also known as the "brain" of the Micromouse, it allows to received input from various sensors. The use of an algorithim allows for decision making from the pre-programed logic from real-time input.	$24.95
7	L9110H H-Bridge Motor Driver for DC Motor-8Pin- 2.5V-1.2V 800mA	$1.50	2	Allows for direction control as the polarity of the voltage applied to the motor determines whether it will rotate clockwise or counterclockwise.	$24.95
8	Pololu Micro Metal Gearmotor Bracket Extended Pair	$4.95	1	Allows for mounting and support for a gearbox within the motor. It also offeres alignment and stability.	$24.95
9	26 Pcs Self Adhesive Wheel Casters	$8.75	1	Creates stability within the Micromouse's body as it maintains balance as it moves in multiple directions. It has a small contact surface, reducing friction.	$8.75
10	5 Pcs LM2596 DC to DC Buck Converter	$7.99	1	Facilitates the voltage step-down of a 9V battery to a contant 5V voltage required by the motors. This prevents damge from excessive voltage.	$7.99
11	Magnetic Encoder Pair Kit with Side Connector for Micro Metal Gearmotor, 12 CPR, 2.7-18V	$7.89	1	Measures the rotational speed of each wheel and determins the exact location of the Micromouse. It counts the traveled distance based on motor revolutions.	$7.89
12	Lithium Ion Polymer battery - 3.7V 350mAh	$6.95	1	Supplies 3.7 volts to the microcontroller enabaling a direct and stable voltage input for the low voltage interfacing components.	$6.95
13	10 Pcs 5x7" - 0.043" Acrylic Plexiglass	$6.99	1	Due to the custome size and arrangement that the Micromouse has, acrylic was used due to its durability. Since its acrylic, it can be laser cutted to any given size.	$6.99
14	Jumper Wire Kit	$13.99	1	To allow protoyping, jumper wires were used to help finalzie the pin outs and circuitry to each component found within the Micromouse.	$13.99
15	6-Pin Single-Ended Female JST SH-Style Cable 12cm	$1.19	2	Connectors to the encoders to allow be read from the motors to the microcontroller	$2.38
16	PCB Board	$43.00	1	To have a consolidated electrical design	$43.00
Total					$296.79

Bill of Materials – Mapping Team

Item Number	Item Name	Price Per Unit	Quantity	Price	Place of Purchase
1	HC-SR04 (2 PACK)	$6.99	1	$6.99	Amazon
2	MB1030	$29.95	1	$29.95	DigiKey
3	SG90 Micro Servo Motor (2 PACK)	$6.99	1	$6.99	Amazon
4	Adafruit VL6180X Time of Flight Distance Ranging Sensor	$13.95	3	$41.85	Adafruit
5	Adafruit VL53L0X Time of Flight Distance Sensor	$16.95	1	$6.95	RS American
6	Square Wood Sheets	$12.19	1	$12.19	Amazon
7	GP2Y0A21	$12.99	2	$25.98	Amazon
Total				$140.90	

Research Requirements



Automation

- The *Micromouse* robot must be able to navigate and solve the maze autonomously in the shortest amount of time

Hardware

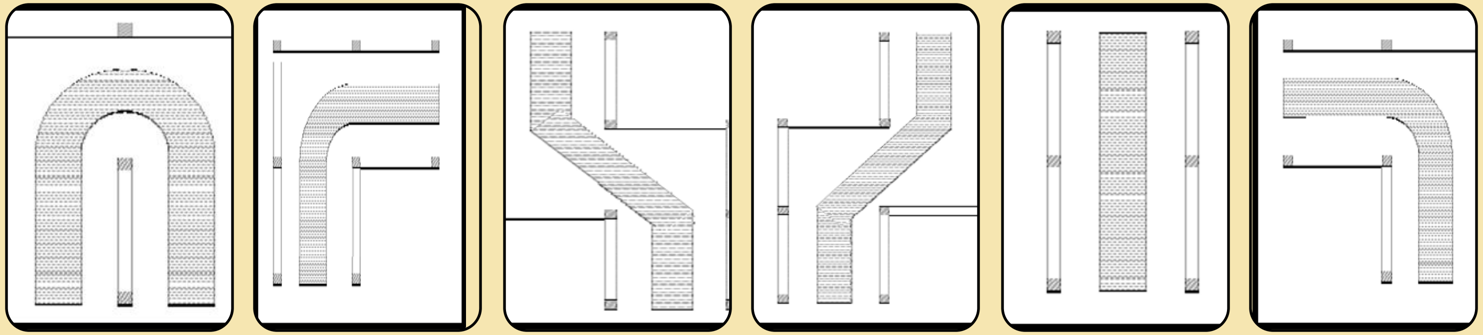
- The *Micromouse* robot must limited to onboard hardware and processing and may not receive external inputs

Maze Restriction

- The *Micromouse* Maze is at a fixed size but the configuration for the correct path isn’t revealed until the day of

Preliminary Research

Micromouse Size Restrictions



Loop U-Turn

Right 90° Turn

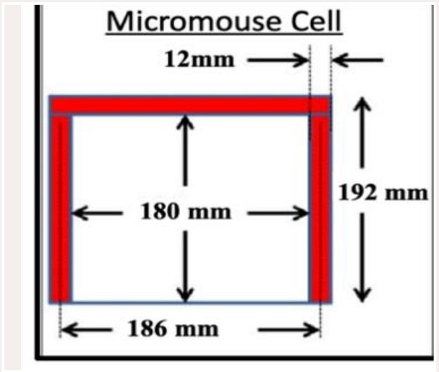
Left Diagonal Maneuver

Right Diagonal Maneuver

Straight Forward Maneuver

Left 90° Turn

Resize

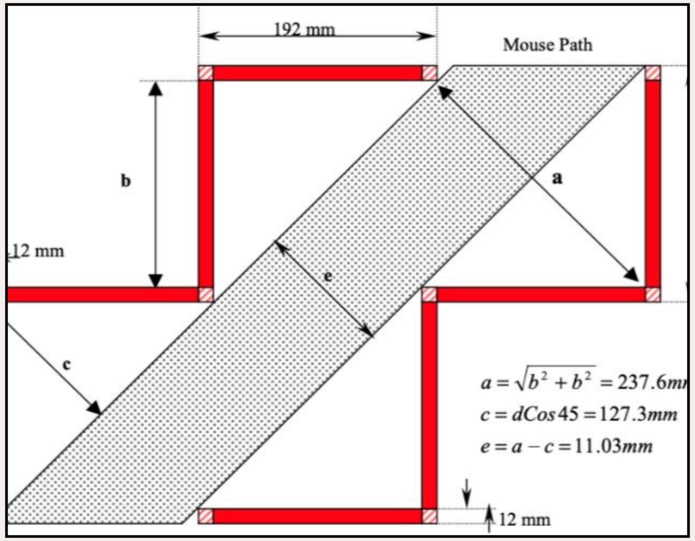


Based on the given parameters from the IEEE Rules, the Unit Cell in Micromouse is illustrated

Preliminary Research

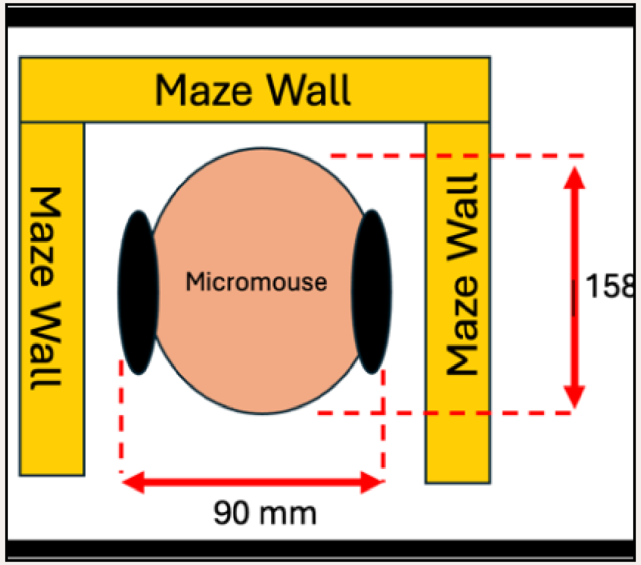
Micromouse Size Restrictions

Resize



To cut across, the "e" path clearance is calculated to create the max length and width of the Micromouse

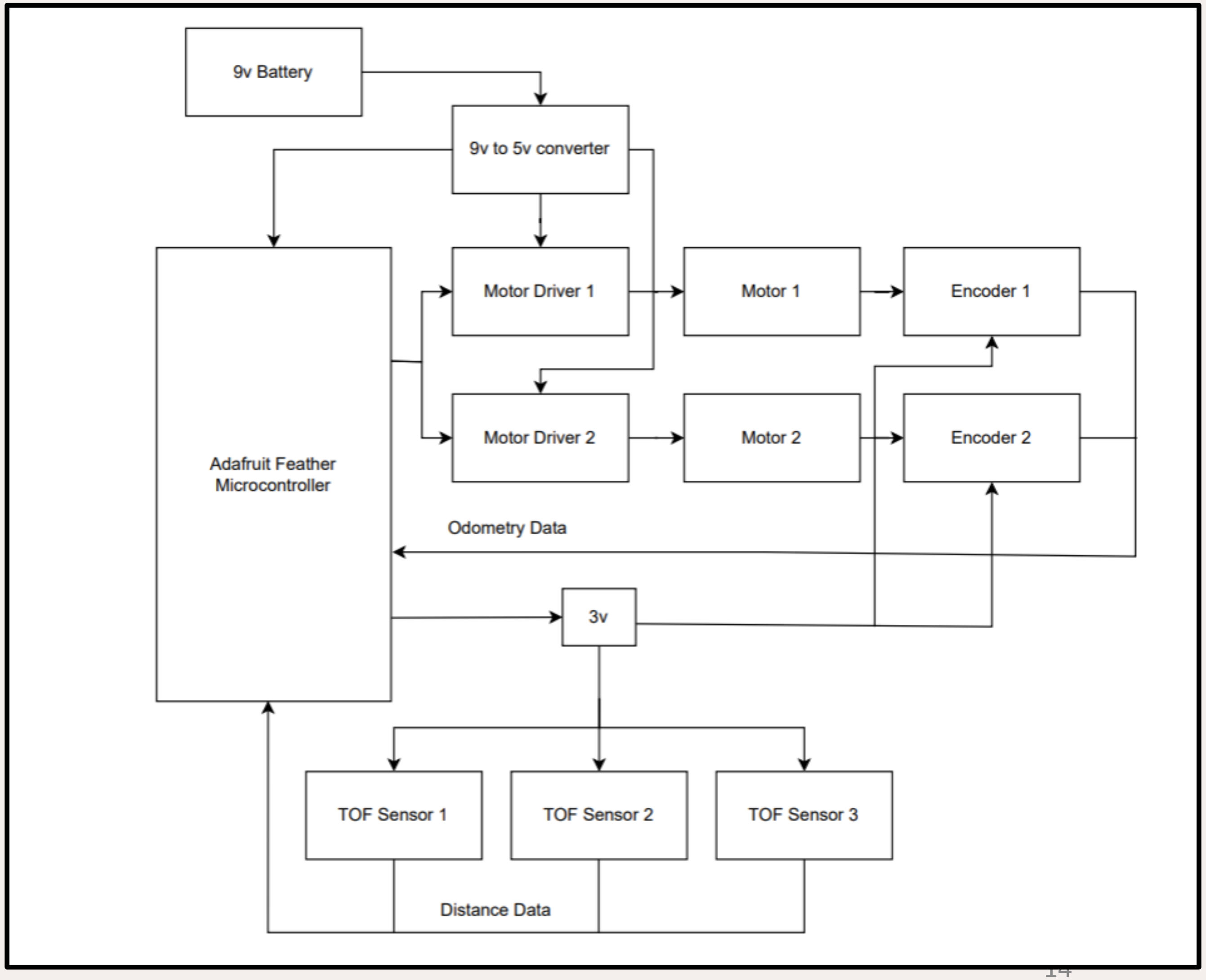
Resize



- **Width:** 90 mm
- **Length:** 158 mm

System Level Overview

- A high-level flowchart was constructed to help connect the different systems
- A PCB will be used as a main circuity hub
- Readings from all sensors will go to and be interpreted in the microcontroller
- Consequential motor speed, direction, and distance will be sent to the motor driver then to all motors.



Mechanical Design Overview

- All 3 Wheels were tested and 60x8 mm Pololu Wheels were chosen as the most efficient

32x7 mm Pololu Wheels

40x7mm Pololu Wheels

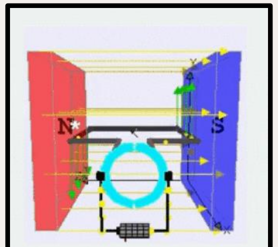
60x8 mm Pololu Wheels

Ball Caster

Due to only having two wheel on the chassis, two ball casters on each end are placed to balance the Micromouse

Mechanical Design Components

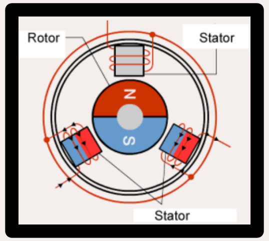
LEGEND
1 = Poor
2 = Fair
3 = Good
4 = Very Good
5 = Excellent


DC Motor


AC Motor

Criteria	DC Motors	AC Motors
Low Cost	2	3
Efficiency	4	2
Speed Control	4	3
Size and Weight	4	3
Power Output	5	4
Total	19	15

Criteria	Brushless DC	Brushed DC	Servo DC
Cost	1	3	2
Efficiency	3	3	2
Speed Control	4	3	5
Total	8	9	9



Dimensions

Size:	10 × 12 × 26 mm[1]
Weight:	9.5 g
Shaft diameter:	3 mm[2]

Although the best choice of a motor is a Brush DC Motor, a greater torque can be added within the motor through a Gearbox Motor.

Mechanical Design Components

Motors- Drivetrain (Motor Case Study) –
30:1 Micro Metal Gearmotor MP 6V

Performance at maximum efficiency	
Max Efficiency @ 6V:	41%
Speed at max efficiency:	830 rmp
Torque at max efficiency	0.10 kg·cm
Current at max efficiency	0.36 A
Output power at max efficiency	0.89 W

General Specifications	
Gear Ratio	29:86:1
No-load Speed @ 6V:	1000 rmp^3
No-load current @ 6V:	0.10A^4
Stall current @ 6V:	1.6 A
Stall Torque @ 6V:	0.57 kg·cm
Max output power @ 6V:	1.5 W
Extended motor shaft?	N
Motor Type:	1.6 A stall @ 6V (HP 6V)

Voltage	No-load Performance	Stall Extrapolation
6 V	1000 RMP, 100 mA	0.57 kg·cm (7.9 oz·in), 1.6 A

The Specs that came with the Pololu Motor

Design Overview – Electrical Components

Odometry – Drivetrain

(Odometry Device Trade Study)

Criteria	Hall Effect Sensor	Optical Sensor
Low Cost	5	2
Accuracy	3	5
Mounting	5	3
Ease of Use/Measurement	4	3
Durability	5	3
Total	17	22

- Two types of odometry sensors were considered
- A Hall Effect sensor could be mounted parallel to a magnet on each wheel of the mouse to measure wheel rotation by magnetic field.
- An optical sensor such as a computer mouse sensor could be mounted on the bottom of the mouse to detect changes between optical captures
- The main drawbacks of the Hall Effect sensor come from solely measuring motor rotations
- The optical sensor could be difficult to configure with the microcontroller to measure odometry
- The Hall Effect sensor was chosen after testing as the ideal odometry sensor

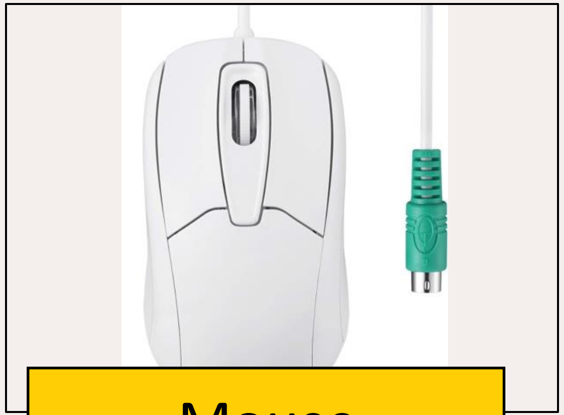
Design Overview – Optical Sensor

Two types of optical sensors were ordered and compared

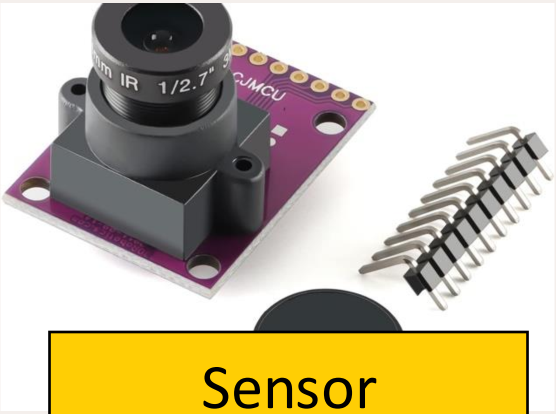
The optical sensor was tested first because it could give full range movement sensing

One sensor was a standalone optical sensor while the other was from a PS2 computer mouse

Sensors would be tested on different floor types to ensure versatility



Mouse



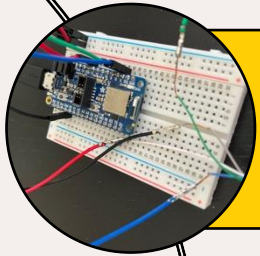
Sensor

Design Components – Optical Testing

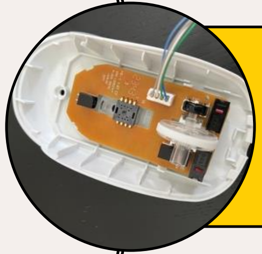
Odometry - Optical Test

- A PS2 optical mouse sensor was found to be more versatile
- This mouse uses a PAW 3515DB Series Optical Mouse Chip
- Accurate measurement over a wide range of surfaces
- Up to 30 inches/sec measurement
- 3300 frames/sec with 1000 DPI
- Easy to calculate real distance travelled read from mouse data pin:
  - Distance per count (inches) = 1 / DPI
  - **Total distance (mm) = ∑(Counts per frame) X 1 / DPI X 25.4**
- Ex: If over 1 second the mouse reports 500 counts
- Total distance (mm)=0.5×25.4=12.7 mm

Odometry – Optical Test Issues



A standard PS2 mouse has VCC, GND, CLK, and DATA pins



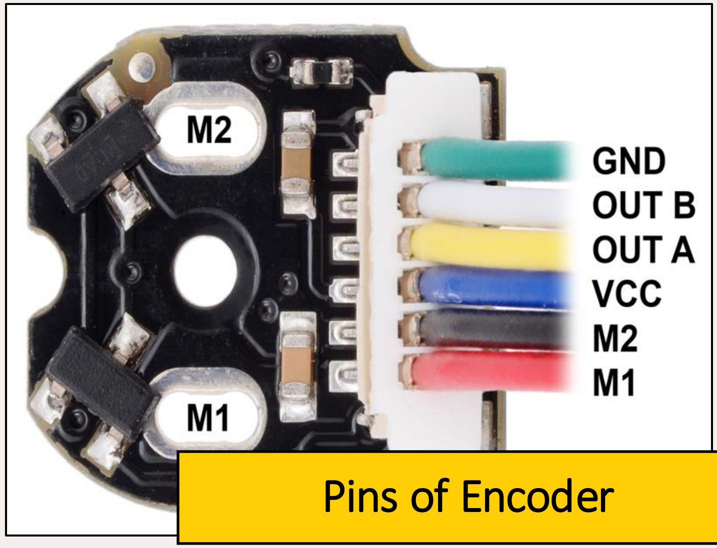
The sensor was tested by sending clock pulses and measuring changes between frames

Unfortunately, inconsistent or no data was read from the PS2 data pin which was an unresolved issue.

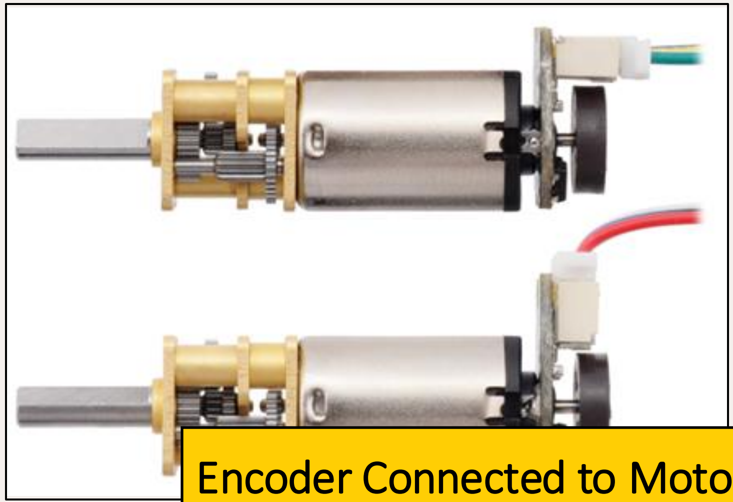
Design Components – Hall Effect

Odometry – Encoder Implementation

- An encoder using the hall-effect principle is how the mouse tracks/controls distance travelled
- "Ticks" are the base unit of encoder rotation and can be directly related to distance travelled using the circumference of the wheel.
- With a full rotation value of 180 ticks and a wheel diameter of 60.0mm gives 188.5mm travelled per rotation.
- This value can be used to tell the motors how far to go/turn as well as store distanced travelled for mapping purposes



Pins of Encoder


Encoder Connected to Motor

Design Components – Odometry Software

```
// Calculate distance based on encoder ticks
float calculateDistance(int ticks) {
  float revolutions = (float)abs(ticks) / ticksPerRevolution;
  return revolutions * pi * wheelDiameter; // Distance in mm
}
```

Odometry Code

- Overall goal is to provide accurate move forward, turn right, and turn left functions for mapping team to use to navigate the maze
- Synching each motor to stop turning at the same time proved to be difficult resulting in a slight left turning forward motion
- This problem was solved temporarily by frontloading software computing before motor control code
- As the overall software to control the robot gets more complex, it is essential to correct robot position using distance sensors.

Design Components – Parametric Chassis

Chassis Design – Battery Implementation: Case Study

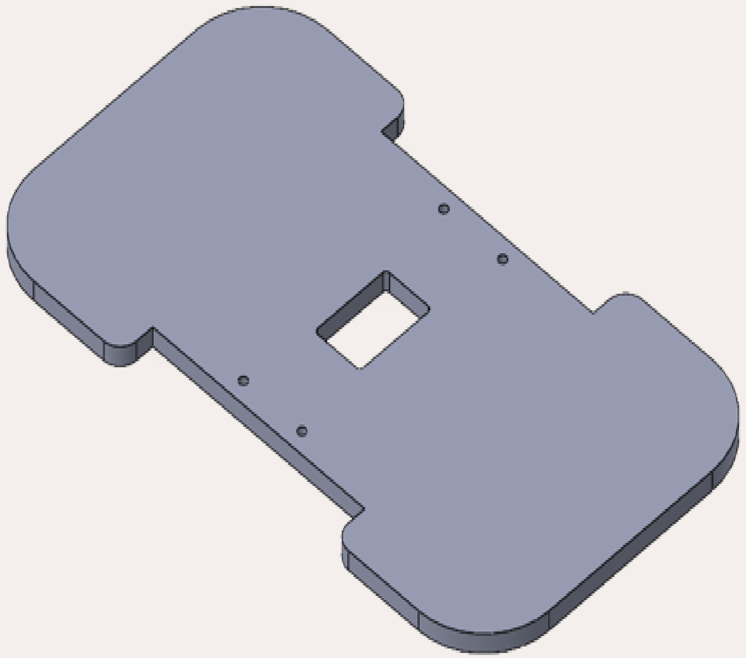
Attribute	9V Battery	4 AAA Batteries	LiPo Battery
Voltage	4 (Compact, fixed 9V)	3 (6V total, modular)	5 (Versatile, scalable packs)
Energy Density	3 (Moderate)	2 (Lower density)	5 (High energy density)
Cost	4 (Affordable)	4 (Affordable per unit)	3 (Higher cost)
Weight	4 (Lightweight)	3 (Comparable weight to 9V)	5 (Lightweight relative to power)
Lifespan (Rechargeable)	3 (Moderate cycles)	3 (Moderate cycles)	4 (Longer lifespan with care)
Applications	3 (Specific uses)	4 (Wide variety of uses)	5 (High versatility)
Advantages	3 (Compact and easy to use)	4 (Modular and versatile)	5 (Powerful and lightweight)
Disadvantages	3 (Lower capacity limits)	3 (Bulky for packs)	2 (Requires careful handling)
Total	27	26	34

» Based on the Case Study, the best option to use is the LiPo Battery. Although it is implemented within the prototype, it is only 3.7 V to avoid a voltage overload within the rest of the components

» It is also important to note that the motor are powered at 5V through the 9V battery directly in the prototype with the use of a buck converter

Design Components – Parametric Chassis

Chassis Design – Overview: Base Chassis



Parametric Base Chassis

- Chassis allows for everything to be mounted on together and move
- Due to not knowing which wheel type would work best up at this point, the best option is to make a **parametric model**
- A parametric model would allow is to make all the parts interchangeable
- In making the Parametric Model, key size restrictions are pointed out to ensure the interchangeability

Design Components – Parametric Chassis

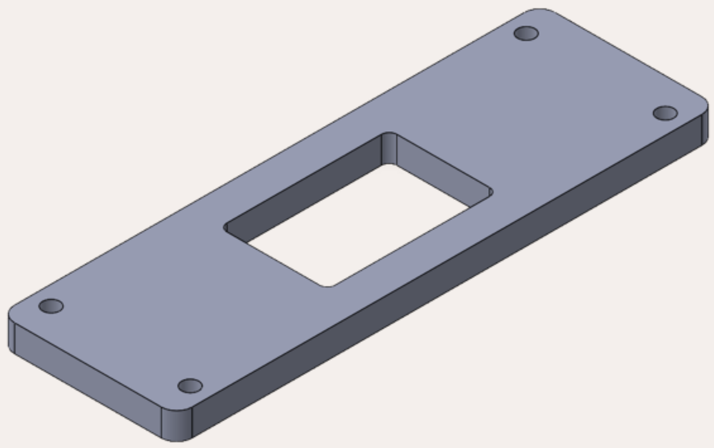
Chassis Design – Key Size Restrictions: Motor Spacer

Size Restrictions

- **Maximum Width: 90 mm**
- **Maximum Length: 158 mm**
- **Minimum Width: 66 mm**
- **Minimum Length: 158 mm**
- **Thickness: 3.05**

Maze Restrictions (Maximum Width, Maximum Length)

Components [Motor and Encoder] Restrictions (Minimum Width, Minimum Length)



Motor Spacer

The Thickness of the acrylic for the chassis is also taken into account as the purpose of this is to create enough offset between the motor's height and the wheel's radius.

The Motor space is placed in the middle of the Base Chassis to allow for encoder wiring to pass through to the microcontroller.

This Motor Spacer is only applicable for the Large Ribbed Wheels

Design Components – Parametric Chassis

Chassis Design – Key Size Restrictions: Wheel Dimensions

Wheel Dimensions			
Model	Parameter	Value	Unit
32 x 7 Pololu Wheels [Small Rigid]	Outer Diameter	32	mm
	Outer Radius	16	mm
	Inner Diameter	23.15	mm
	Inner Radius	23.15	mm
	Thickness	7	mm
	Diamter of adapter	3	mm
40 x 7 Pololu Wheel [Medium Smooth]	Outer Diameter	40	mm
	Outer Radius	20	mm
	Inner Diameter	30	mm
	Inner Radius	15	mm
	Thickness	7	mm
	Diameter of adapter	3	mm
60 x 8 mm Pololu Wheel [Large Rigid]	Outer Diameter	60	mm
	Outer Radius	30	mm
	Inner Diameter	47.25	mm
	Inner Radius	23.625	mm
	Thickness	8	mm
	Diamter of adapter	3	mm

Rigid Smooth Radius: 16 mm

Medium Smooth Radius: 20 mm

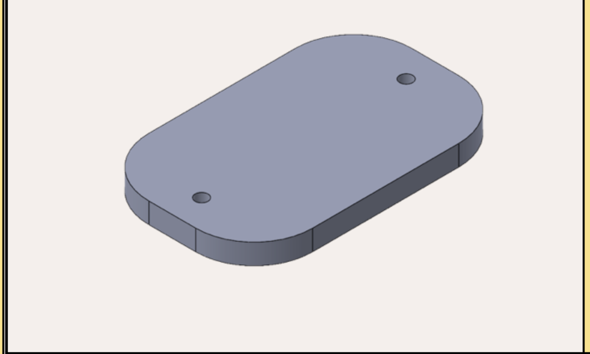
Large Rigid Radius: 30 mm

Based on these measurements, the needed height for each caster is shown on the right

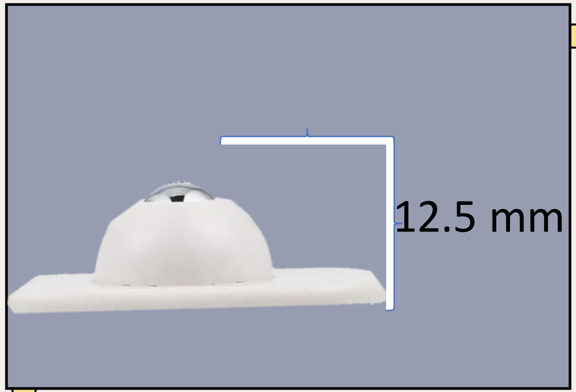
Design Components – Parametric Chassis

Chassis Design – Key Size Restrictions: Caster Spacer Thickness Dimension per Wheel Type

Radius of Wheel Type – Ball Caster Thickness = Thickness of 3D Printed Caster Spacer



Solidworks Model of Caster Spacer



Ball Caster

Rigid Smooth Thickness Needed:
16 mm – 12.5 mm = **4.5 mm**

Medium Smooth Thickness Needed:
20 mm – 12.5 mm = **7.5 mm**

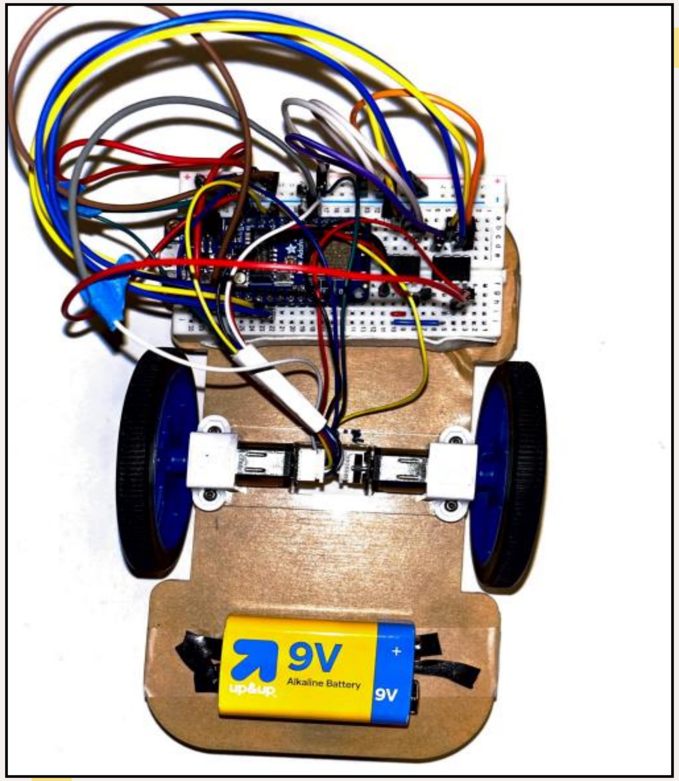
Large Rigid Thickness Needed:
30 mm – 12.5 mm = **16. 5 mm**

With the dimensions need for each wheel type, testing can be performed to start determining which wheel type is the best regarding **Slippage, Traction and Speed**

Design Components – Parametric Chassis

Advantages of Parametric a Chassis

- The advantages of having a parametric model is that it ensures different types of testing regarding the efficiency at a better control, leaving for less room for error.
  - Therefore, continuous testing can occur with the use of this model



Testing with the Parametric Testing

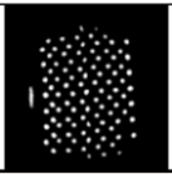
- "One Size Fits All"
- Accommodates mounting points, ensuring stability
- Easily Modifiable

Design Components – Wheel Selection

Wheel Selection – Surface Area Testing: Contact Area

- Greater the diameter of the wheel is, the greater the surface area there is within the *Micromouse*
- In having a greater diameter, there is a greater possibility in loosing traction as speed increases due to the wheel having a high moment of inertia
- In analyzing both tables, it seems as if Large Ribbed Wheels and Medium Sooth Wheels as they offer an in between value where the surface area is great enough to not loose friction, but the diameter is not large where inertia will make a drastic effect in its performance

Contact Area Comparison

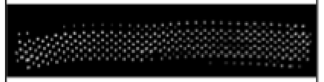
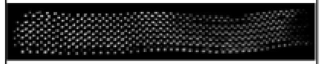
Units	mm					in	mm
Wheel Type	Diameter	Thickness	Length per Rotation (Wheel Circumfernce)	Image Format	ImageJ Processed Image	Actual Area (in)	Actual Area (mm)
Small Rigid	32.15	6.5	100.951	Soft Pressure		0.011	7.09676
				Hard Presure		0.056	36.12896
Small Smooth	40	6.5	125.6	Soft Pressure		0.021	13.54836
				Hard Pressure		0.069	44.51604
Large Rigid	60	7.75	188.4	Soft Pressure		0.03	19.3548
				Hard Pressure		0.109	70.32244
RC Buggy Wheels	62.26	29	195.4964	Soft Pressure		0.045	29.0322
				Hard Pressure		0.178	114.8385

Design Components – Wheel Selection

Wheel Selection – Surface Area Testing: Surface Area of the Circumferance

- Both Large Ribbed Wheels and Medium Smooth Wheels had a smaller Percent Difference from the expected values vs. their experimental values

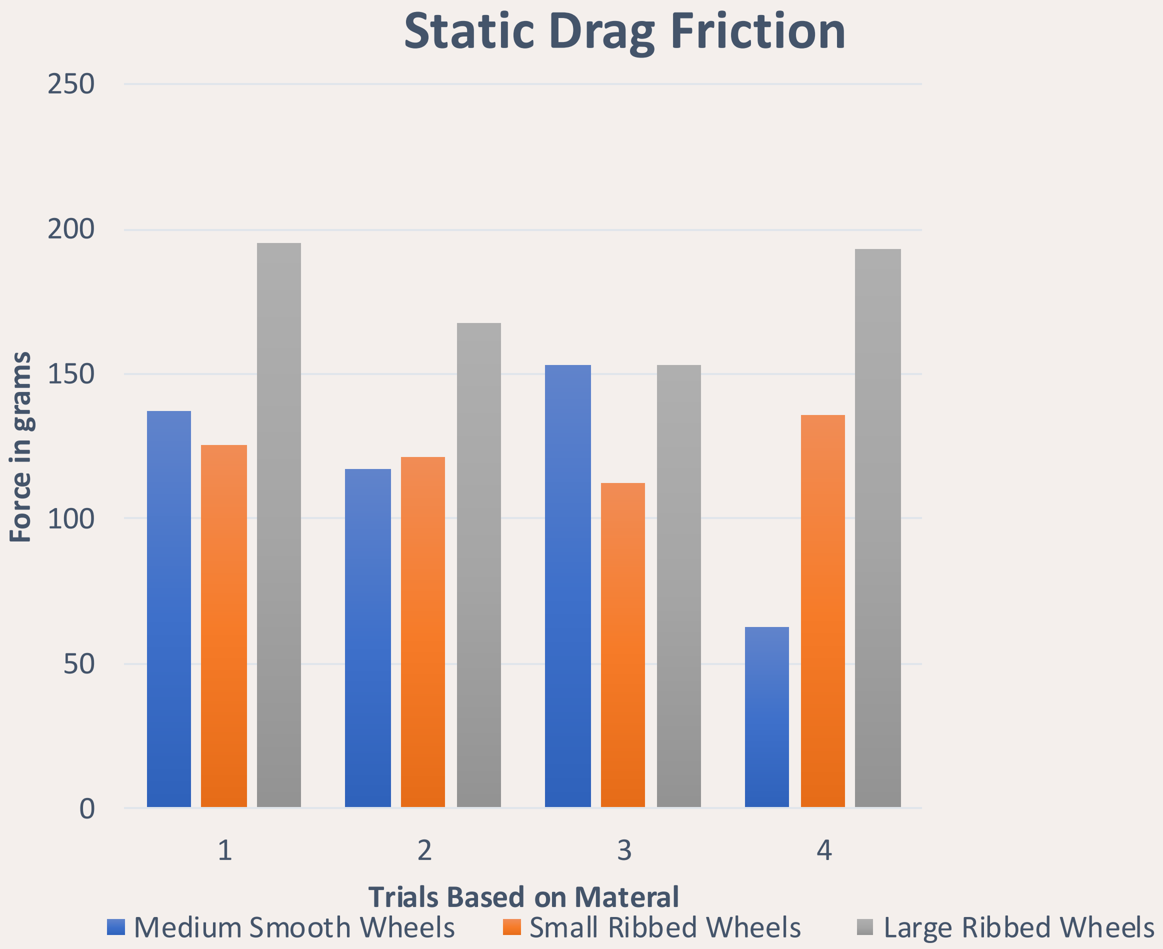
- Based on the information presented within the tables, it seems as if Large Ribbed Wheels to be the most efficient choice but more test need to be run

Surface Area of the Circumferance									
Units	mm					mm	in	mm	mm
Wheel Type	Diameter	Thickness	Length per Rotation (Wheel Circumference)	Image Format Title	ImageJ Processed Image	Expected Area (mm^2)	Actual Area (in)	in mm	Percent Difference
Small Rigid	32.15	6.5	100.951	Soft Pressure, 1Rotation		656.1815	0.183	118.06428	82.01%
				Hard Pressure, 1Rotation		656.1815	0.303	195.48348	70.21%
Small Smooth	40	6.5	125.6	Soft Pressure, 1Rotation		816.4	0.509	328.38644	59.78%
				Hard Pressure, 1Rotation		816.4	0.881	568.38596	30.38%
Large Rigid	60	7.75	188.4	Soft Pressure, 1Rotation		1460.1	0.416	268.38656	81.62%
				Hard Pressure, 1Rotation		1460.1	0.97	625.8052	57.14%
RC Buggy Wheels	62.26	29	195.4964	Soft Pressure, 1Rotation		5669.3956	0.514	331.61224	94.15%
				Hard Pressure, 1Rotation		5669.3956	1.012	652.90192	88.48%

Design Components – Wheel Selection

Wheel Selection – Static Drag Friction Testing

Static Drag friction									
Wheel Type	Trial Number	Trial 1 - Formica		Trial 2 - Melamine		Trial 3 - Paper		Trial 4 - Painted Wood	
		Values		Values		Values		Values	
		lb	g	lb	g	lb	g	lb	g
Large Ribbed Wheels	1	0.32	145.15	0.26	117.93	0.36	163.29	0.16	72.57
	2	0.32	145.15	0.28	127.01	0.36	163.29	0.14	63.50
	3	0.3	136.08	0.28	127.01	0.34	154.22	0.14	63.50
	4	0.32	145.15	0.24	108.86	0.34	154.22	0.12	54.43
	5	0.3	136.08	0.24	108.86	0.32	145.15	0.14	63.50
	6	0.3	136.08	0.28	127.01	0.32	145.15	0.12	54.43
	7	0.28	127.01	0.26	117.93	0.32	145.15	0.14	63.50
	8	0.3	136.08	0.24	108.86	0.34	154.22	0.14	63.50
	9	0.28	127.01	0.26	117.93	0.32	145.15	0.14	63.50
	10	0.3	136.08	0.24	108.86	0.36	163.29	0.14	63.50
	AVG	0.302	136.98	0.258	117.03	0.338	153.31	0.138	62.60
Small Ribbed Wheels	1	0.28	127.01	0.28	127.01	0.26	117.93	0.32	145.15
	2	0.28	127.01	0.26	117.93	0.24	108.86	0.3	136.08
	3	0.28	127.01	0.28	127.01	0.26	117.93	0.3	136.08
	4	0.28	127.01	0.24	108.86	0.26	117.93	0.3	136.08
	5	0.3	136.08	0.26	117.93	0.24	108.86	0.3	136.08
	6	0.28	127.01	0.28	127.01	0.24	108.86	0.3	136.08
	7	0.28	127.01	0.26	117.93	0.24	108.86	0.3	136.08
	8	0.26	117.93	0.26	117.93	0.24	108.86	0.3	136.08
	9	0.26	117.93	0.28	127.01	0.24	108.86	0.28	127.01
	10	0.26	117.93	0.28	127.01	0.26	117.93	0.3	136.08
	AVG	0.276	125.19	0.268	121.56	0.248	112.49	0.3	136.08
Medium Smooth Wheels	1	0.44	199.58	0.3	136.08	0.36	163.29	0.38	172.36
	2	0.44	199.58	0.38	172.36	0.36	163.29	0.4	181.44
	3	0.4	181.44	0.38	172.36	0.34	154.22	0.44	199.58
	4	0.4	181.44	0.38	172.36	0.34	154.22	0.44	199.58
	5	0.44	199.58	0.36	163.29	0.34	154.22	0.46	208.65
	6	0.44	199.58	0.38	172.36	0.34	154.22	0.4	181.44
	7	0.44	199.58	0.36	163.29	0.34	154.22	0.46	208.65
	8	0.46	208.65	0.38	172.36	0.32	145.15	0.44	199.58
	9	0.4	181.44	0.38	172.36	0.32	145.15	0.4	181.44
	10	0.44	199.58	0.4	181.44	0.32	145.15	0.44	199.58
	AVG	0.43	195.04	0.37	167.83	0.338	153.31	0.426	193.23

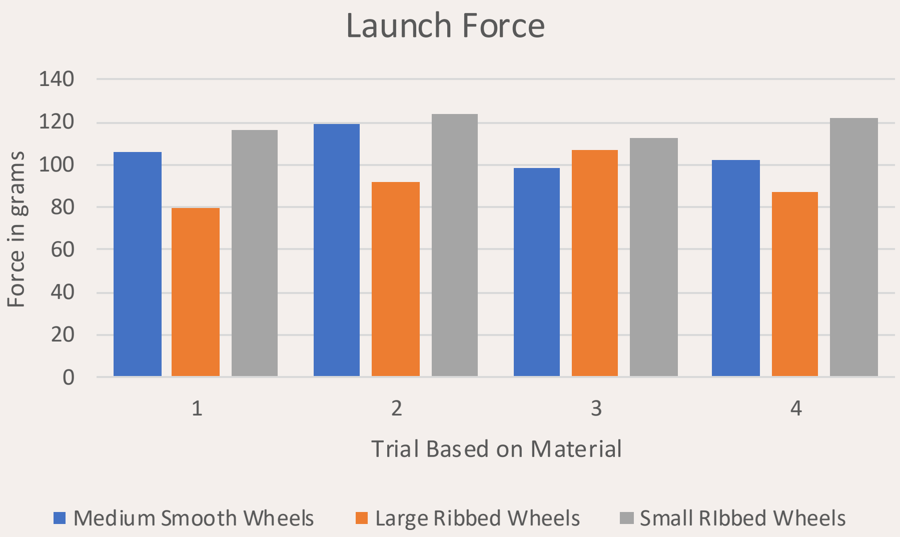


Based on this analysis, Large Ribbed Wheels had the greatest static drag friction, therefore a greater grip/ traction to all 4 surfaces

Design Components – Wheel Selection

Wheel Selection – Launch Force Testing

Launch Force Testing					
Wheel Type	Trial Number	Trial 1 - Formica	Trial 2 - Melamine	Trial 3 - Paper	Trial 4 - Painted Wood
		Values	Values	Values	Values
		g	g	g	g
Medium Smooth Wheels	1	105	110	90	100
	2	115	125	95	100
	3	115	130	100	105
	4	95	115	95	100
	5	105	125	105	105
	6	105	125	100	105
	7	100	110	110	105
	8	105	120	90	100
	9	105	125	105	105
	10	105	110	95	100
	AVG	105.5	119.5	98.5	102.5
Large Ribbed Wheels	1	80	100	110	95
	2	80	90	110	95
	3	70	100	100	90
	4	75	90	110	90
	5	80	90	105	90
	6	80	95	110	90
	7	80	85	105	85
	8	80	85	105	80
	9	90	90	110	80
	10	85	90	100	80
	AVG	80	91.5	106.5	87.5
Small Ribbed Wheels	1	95	130	115	115
	2	105	115	105	105
	3	120	125	120	130
	4	125	120	115	125
	5	115	130	115	125
	6	125	120	115	125
	7	120	135	105	110
	8	115	115	105	130
	9	120	125	110	130
	10	120	125	125	120
	AVG	116	124	113	121.5



This testing allows for the quantification of the difference between wheel diameter and tire differences on acceleration. It is seen that the Small Ribbed Wheels have a higher Launch Force, but they were observed to slip. Although they have better performance, it would ultimately throw off the odometry, causing inaccuracies regarding the autonomy. This loss of traction places Medium Smooth Wheels and Large Ribbed Wheels as the next best options.

Design Components – Wheel Selection

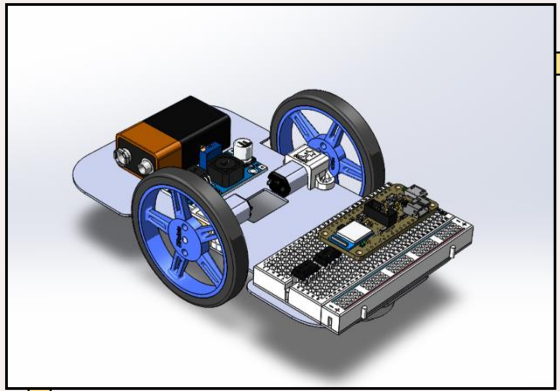
Wheel Selection – Slip Ratio

Slip Ratio (1 Revolution)					
Wheel Type	**Diameter (mm)**	**Trial 1**	**Trial 2**	**Trial 3**	**Average**
Medium Smooth Wheels	40	0.83	0.76	0.84	**0.81**
Large Ribbed Wheels	60	0.87	0.89	0.86	**0.87**

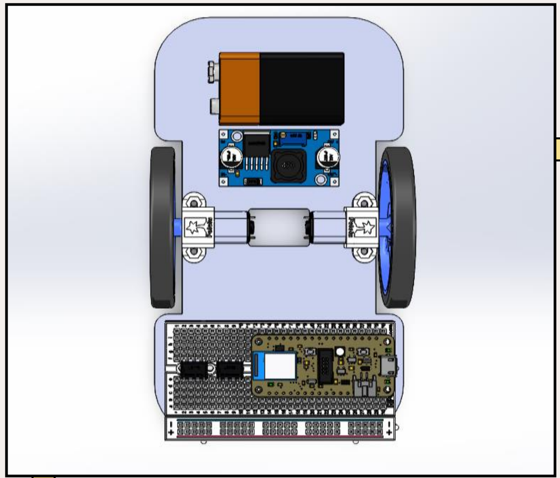
The Slip Ratio was determined by measuring the distance the Micromouse traveled linearly with respect to one full tire revolution. The Large Ribbed Wheels showed the lowest slippage while accelerating in a short distance, simulating the length of 1 Unit Cell a Micromouse moving across the maze.

Design Components – Final CAD Design

Final Prototype Model – Drivetrain Components Only [Chassis and Components Package]

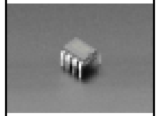
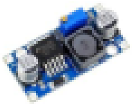
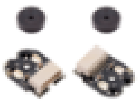
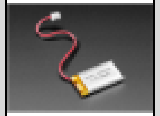


Isotropic View

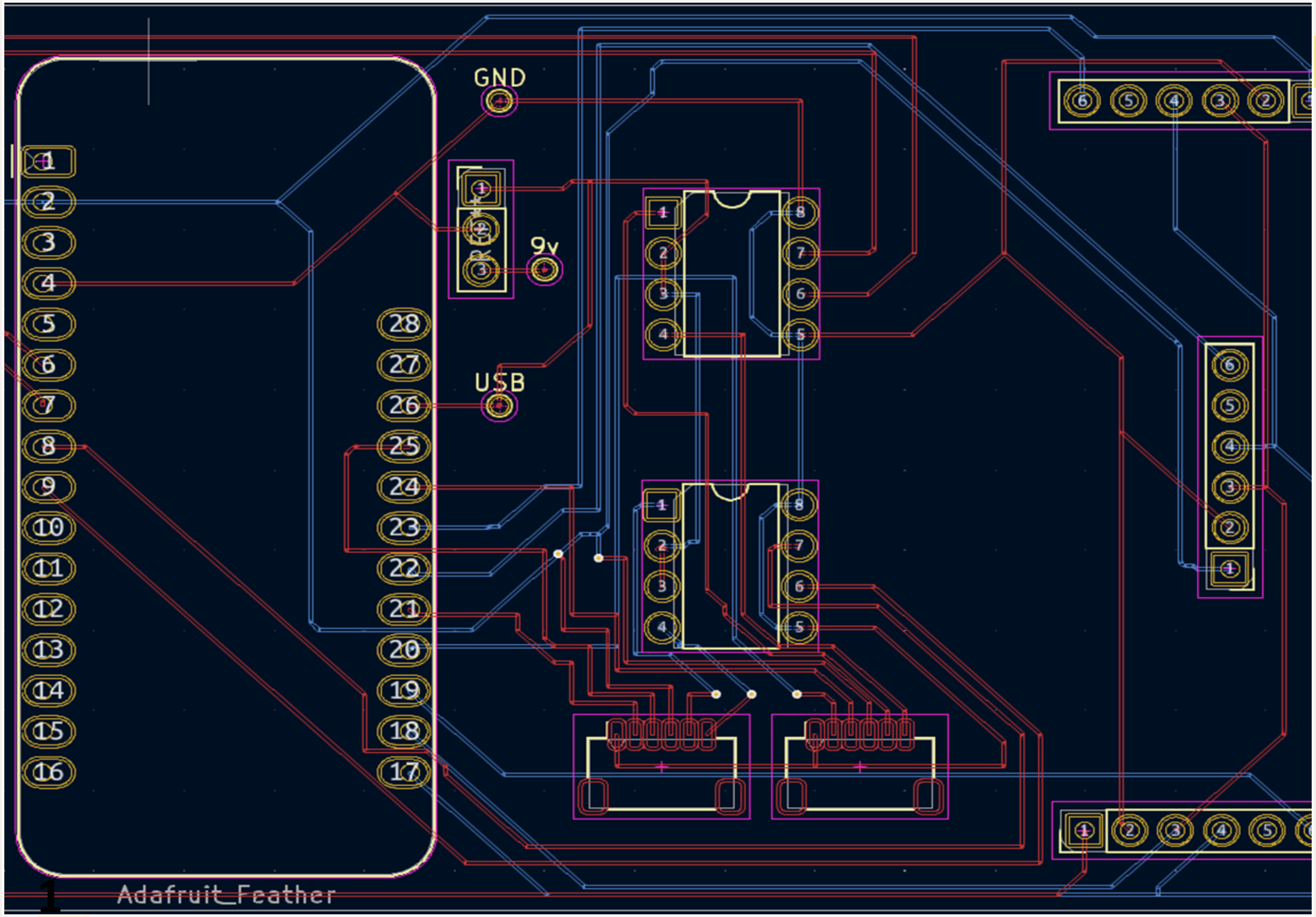


Top View

The components in the table is what is used within the CAD Design

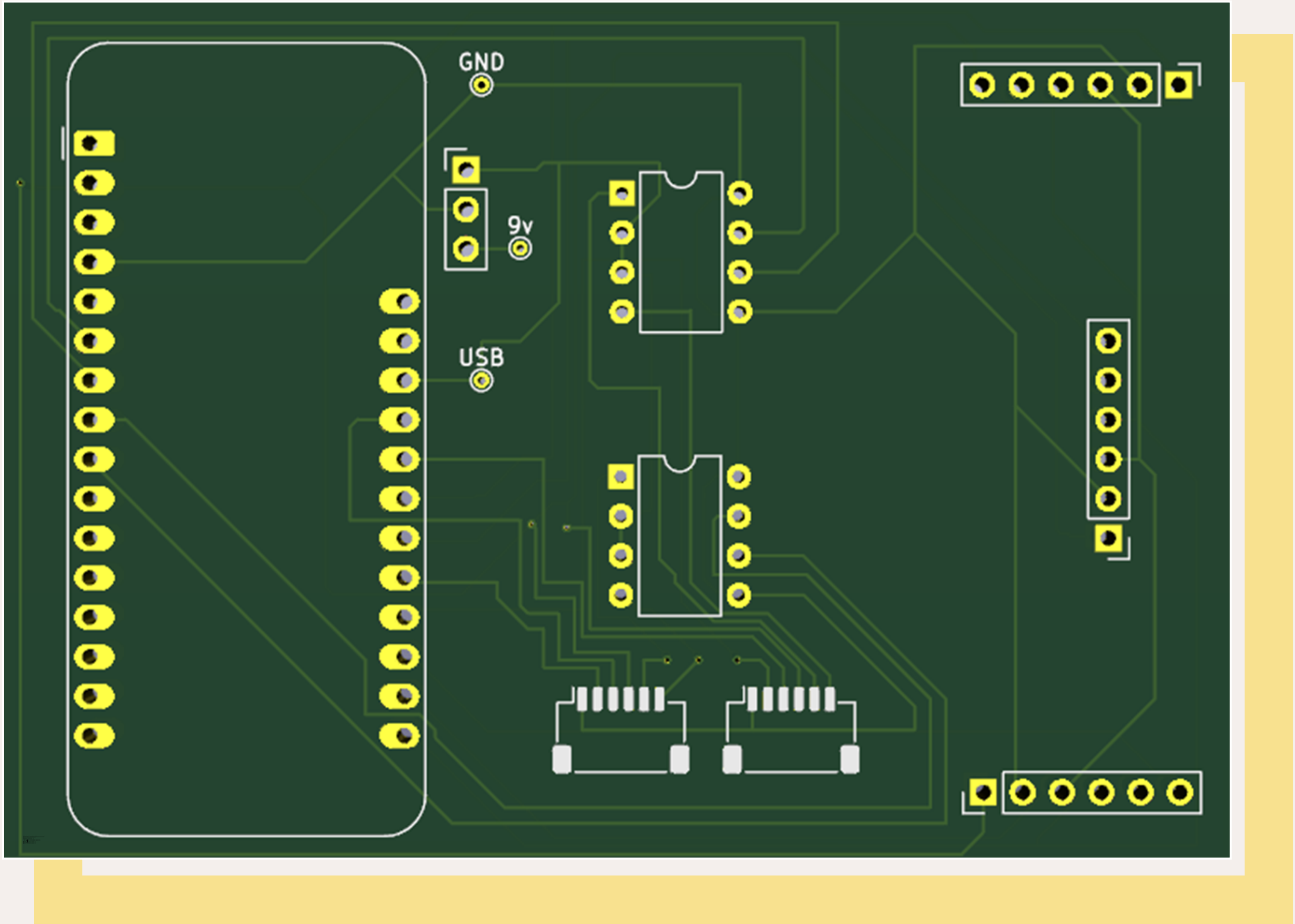
Iteam Number	Item Name	Iteam Image	Quantity	Function
1	30:1 Micro Metal Gearmotor MP 6V		2	Allow for movement to occur within the Micromouse
6	Adafruit Feather nRF52840 Express		1	Also known as the "brain" of the Micromouse, it allows to received input from various sensors. The use of an algorithim allows for decision making from the pre-programed logic from real-time input.
7	L9110H H-Bridge Motor Driver for DC Motor-8Pin- 2.5V-1.2V 800mA		2	Allows for direction control as the polarity of the voltage applied to the motor determines whether it will rotate clockwise or counterclockwise.
10	5 Pcs LM2596 DC to DC Buck Converter		1	Facilitates the voltage step-down of a 9V battery to a contant 5V voltage required by the motors. This prevents damge from excessive voltage.
11	Magnetic Encoder Pair Kit with Side Connector for Micro Metal Gearmotor, 12CPR, 2.7-18V		1	Measures the rotational speed of each wheel and determins the exact location of the Micromouse. It counts the traveled distance based on motor revolutions.
12	Lithium Ion Polymer battery - 3.7V 350mAh		1	Supplies 3.7 volts to the microcontroller enabaling a direct and stable voltage input for the low voltage interfacing components.

Drivetrain Circuit Schematic



- Adafruit Feather Microcontroller
- 9v to 5v Converter
- Motor Driver 1
- Motor Driver 2
- JST 6 pin Encoder Port 1
- JST 6 pin Encoder Port 2
- TOF Sensor Port 1
- TOF Sensor Port 2
- TOF Sensor Port 3

Drivetrain PCB



Electronic Design Components

Sensor Case Study – *Mapping Team*

Approach

- For this project, the mapping team has researched multiple sensors such as infrared sensors, ultrasonic sensors and time of flight sensors. After thoroughly doing research on the pros and cons of these sensors, three sensors have been chosen for the concept design of this project.

Criteria	Infrared Sensors	Ultrasonic Sensors	Time of Flight Sensors
Low Cost	Good	Good	Good
Efficiency	Very Good	Good	Good
Distance	Very Good	Good	Very Good
Size and Weight	Good	Very Good	Very Good
Voltage Power	Good	Very Good	Good

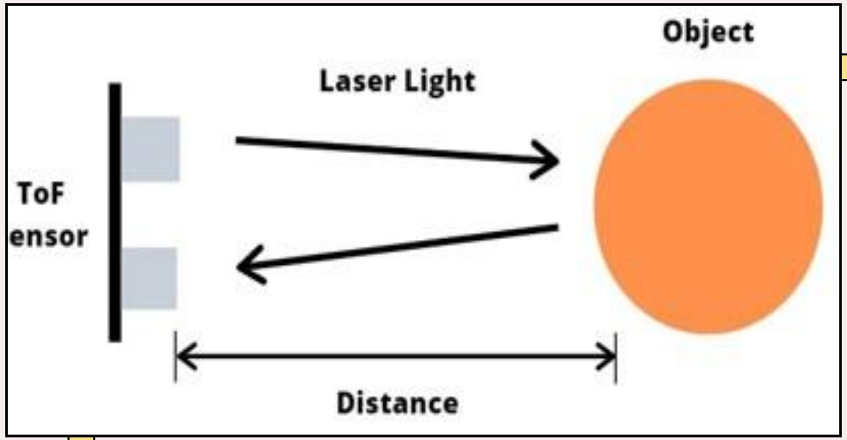
Electronic Design Components

Sensor Research: Time of Flight Sensors – Mapping Team

Approach

- **Two Time-of-Flight (ToF) Sensors: VL6180X and VL53L0X**
- **VL6180X: Proximity Sensor, Ambient Light Sensor, and Vertical Cavity-Emitting Laser (VCSEL) light source**
- **VL53L0X: VCSEL light source with infrared filters**
- **Operate: detects time taken for light to bounce back from a surface**

VL6180 Sensor



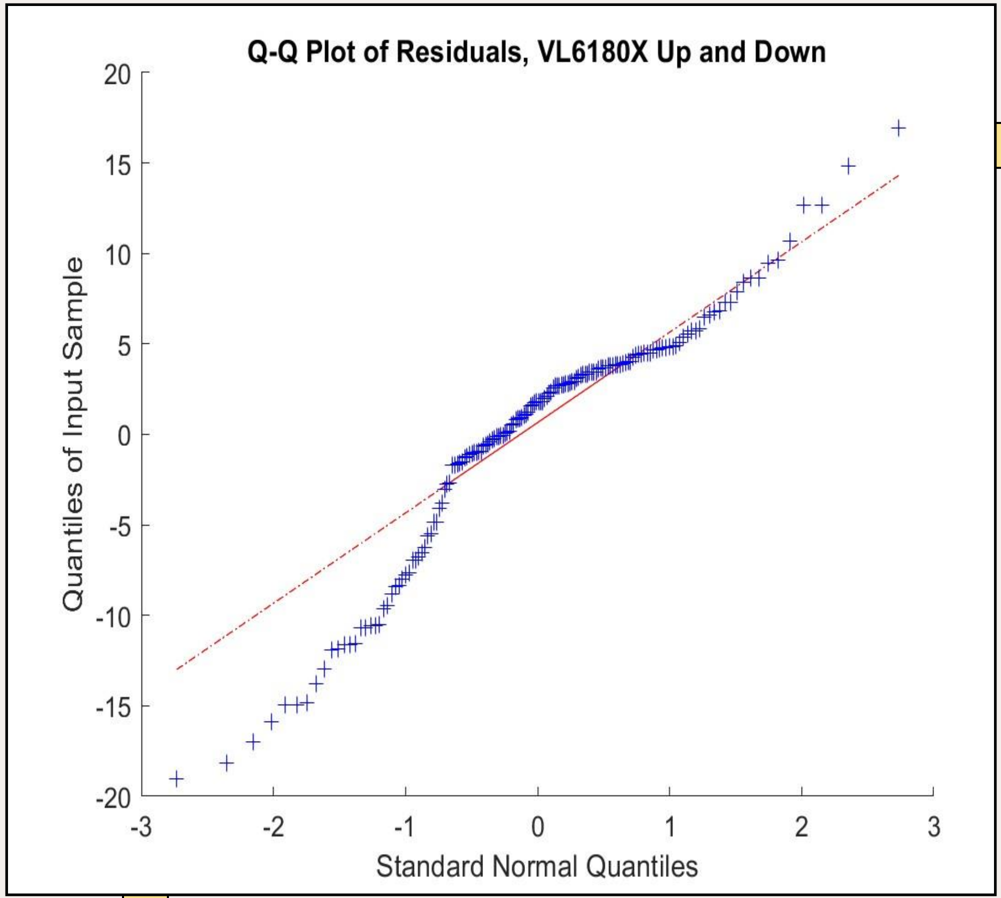
ToF Operation

VL53L0X Sensor

Electronic Design Components

```
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE        tStat       pValue
                   ________    ________    _______    ___________

    (Intercept)    -9.8956       1.0992    -9.0025      6.692e-16
    x1              1.0534     0.009636     109.32    8.6386e-151


Number of observations: 160, Error degrees of freedom: 158
Root Mean Squared Error: 6.73
R-squared: 0.987,  Adjusted R-Squared: 0.987
F-statistic vs. constant model: 1.19e+04, p-value = 8.64e-151
```

VL6180X Sensor

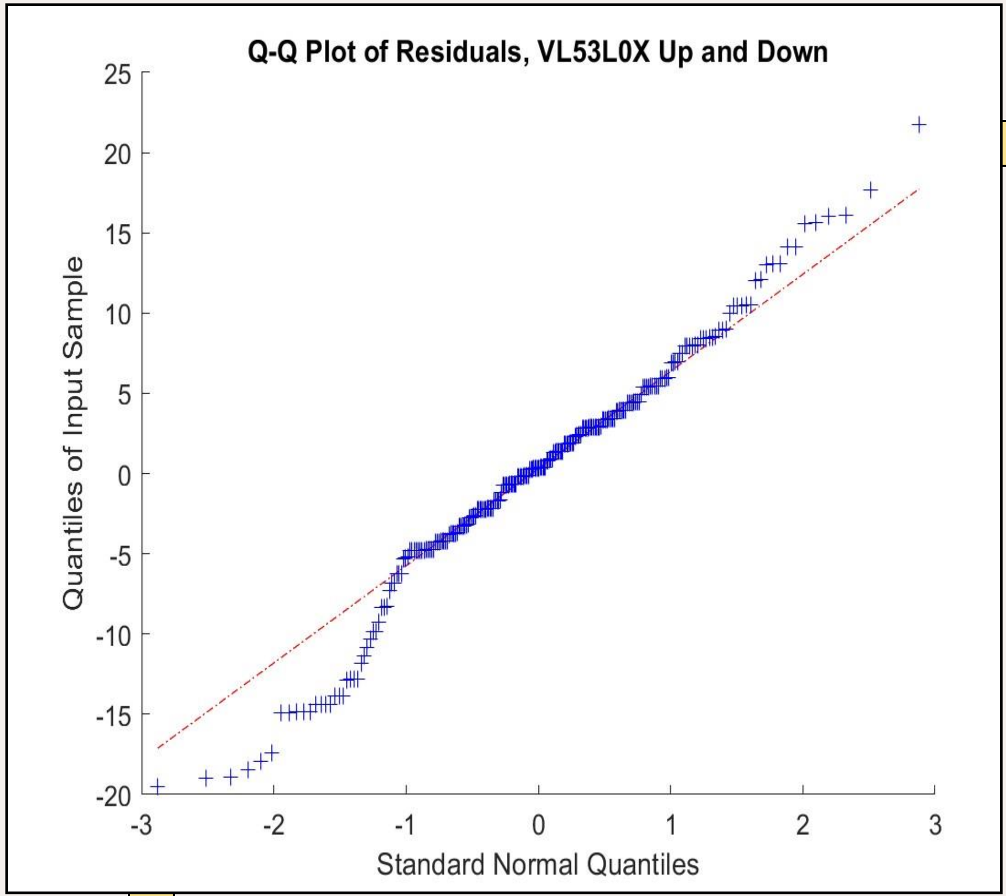


- Root Mean Squared Error: 6.73
- R-squared: 0.987 = 1.3% variance
- Q-Q Plot: sensor readings follow a normal distribution in the short-range region

Electronic Design Components

```
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate        SE         tStat       pValue
                   ________    _________    ______    ___________

    (Intercept)    -5.4848       0.92462    -5.932     1.0006e-08
    x1              1.0164     0.0051529    197.25    1.2978e-274


Number of observations: 250, Error degrees of freedom: 248
Root Mean Squared Error: 7.32
R-squared: 0.994,  Adjusted R-Squared: 0.994
F-statistic vs. constant model: 3.89e+04, p-value = 1.3e-274
```

VL53L0X Sensor



- Root Mean Squared Error: 7.32
- R-squared: 0.994 = 0.6% variance
- Q-Q Plot: residuals follow a normal distribution in the mid-range region

Electronic Design Components

Time of Flight Sensors – Mapping Team

Criteria	VL6180X	VL53L0X
Voltage	2.6-3.0V	2.6-3.5V
Detectable Distance	10-190mm	19-2000mm
Connection	4 pin	4 pin
Pros	Fast accurate distance	Longer detectable distance
Cons	Cannot detect past 100 mm	Readings depend on light absorbed
Price	$13.95	$14.95

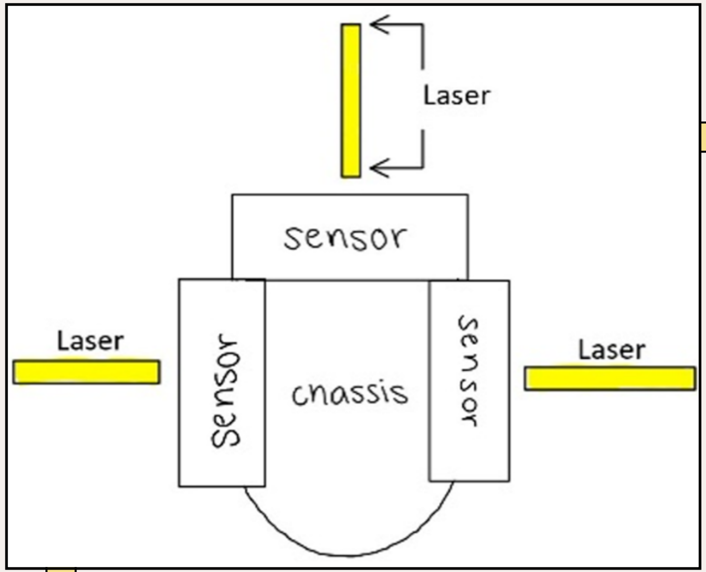
- **VL53L0X detects longer range, VL6180X detects shorter range**
- **The VL6180X can provide precise, close-range measurements, which is useful for detecting obstacles or walls at a short distance**
- **Paired with the Sharp IR sensor, the VL6180X will be the best suited for detecting short distance**
- **Conclusion: When a Sharp IR sensor detects a far-away obstacle, the VL6180X can be used to detect the immediate proximity data, providing a more accurate reading in close spaces**

Electronic Design Components

Sensor Testing: Time of Flight Sensors – Mapping Team

Approach

- VL6180X sensor emits a very narrow light source
- Three VL6180X sensors: front, left, and right
- Faster and accurate maze solving



Sensor Layout Concept Design

Electronic Design Components

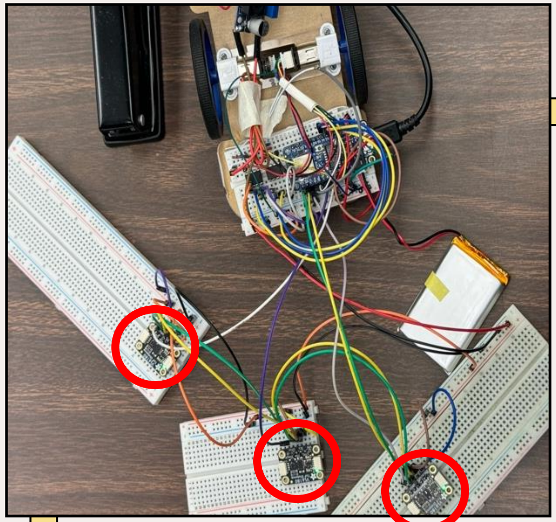
Sensor Software: Time of Flight Sensors VL6180X

1st Testing Arduino Code

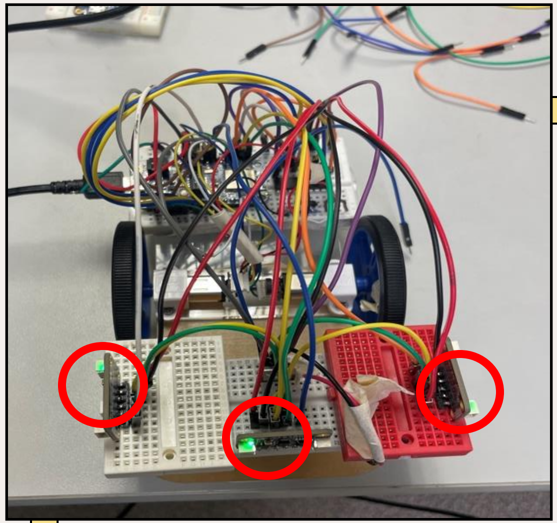
- Three ToF VL6180X sensors are initialized and have their unique addresses to be able to carry out tasks
- Each sensor has its designated XSHUT pin connected to the microcontroller and is outputting its own range onto the Serial Monitor on Arduino

2nd Testing Arduino Code

- Now that we know each sensor is detecting range separately and simultaneously, we need it to be able to detect openings
- Sensors now have a threshold of 30mm, if the sensors detect any range over 30mm then it is an “opening” the Micromouse can then go through



1st Sensor Testing



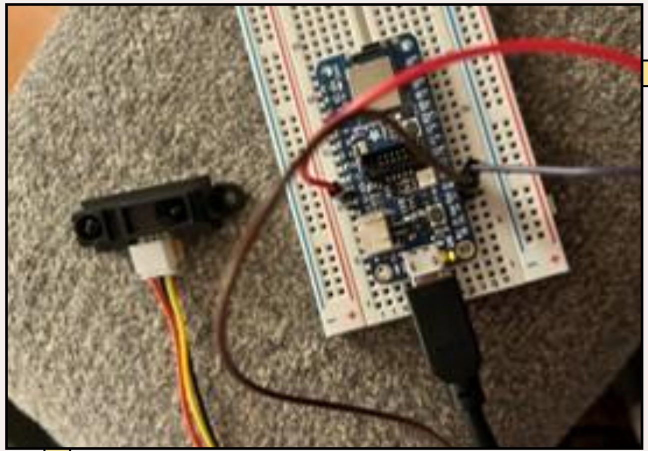
2nd Sensor Testing

Electronic Design Components

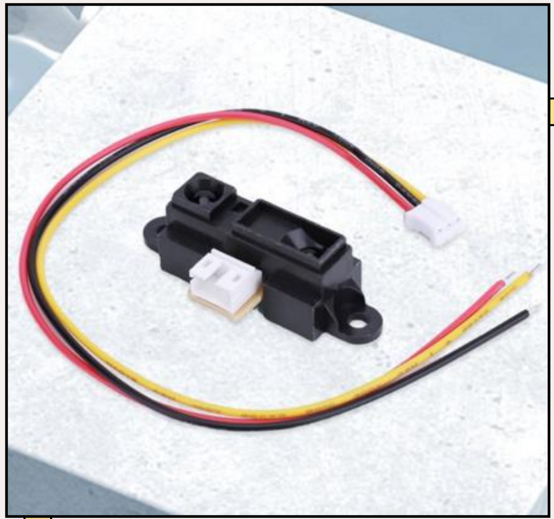
Sensor Research: Sharp IR GP2Y0A21 – Mapping Team

Approach

- The infrared sensor researched is the Sharp IR GP2Y0A21 Distance Sensor works by converting light to electricity or vice versa. This sensor has two lenses which works by emitting a beam of light through one lens and once there is an object detected the beam of light reflects off the object and lands onto the second lens. The pins used to run tests would be GND, USB for 5V and A0.



Testing of Sharp IR GP2Y0A21



Sharp IR GP2Y0A21

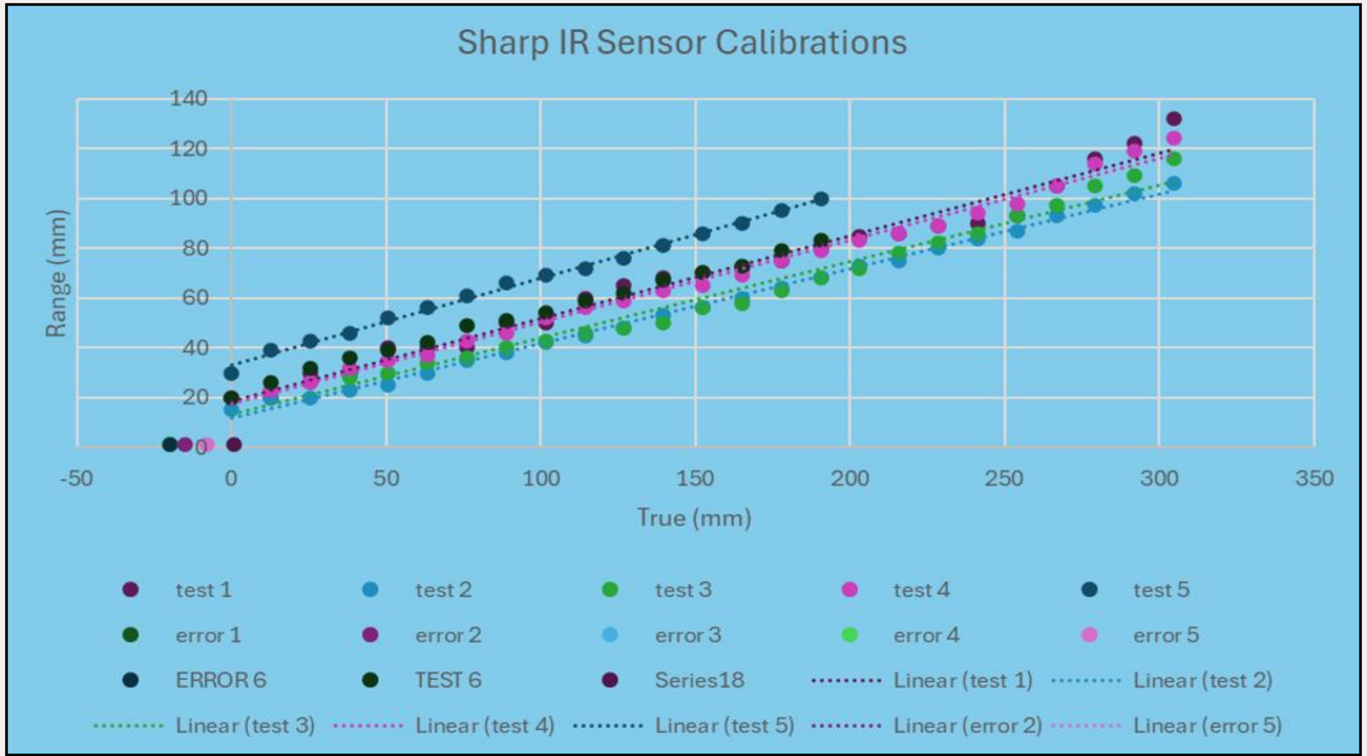
Electronic Design Components

Sensor Testing: Sharp IR GP2Y0A21 – Mapping Team

```
//#define sensor A0 // Sharp IR GP2Y0A41SK0F (4-30cm, analog)
#include <Adafruit_TinyUSB.h>

#include <SharpIR.h>

const int sensorPin = A0;
float distance;

void setup() {
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(sensorPin);

  float voltage = sensorValue * (5.0 / 1023.0);

  distance = 27.86 * pow(voltage, -1.15);

  Serial.print("Sensor Value: ");
  Serial.print(sensorValue);
  Serial.print(" | Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  delay(500);

}
```

Approach

- Within this chart is my sensor calibration curve of the SHARP IR Sensor starting from 0 mm to 132 mm. We ran our code about 5 times to see how each curve would appear. Test 1 through 4 are from the direction of the sensor moving forward and test 5 and 6 are from the direction of the sensor moving backward. The reason for us to do this is to determine the best sensor moving forward on the chassis. The code used is also on this slide.



Electronic Design Components

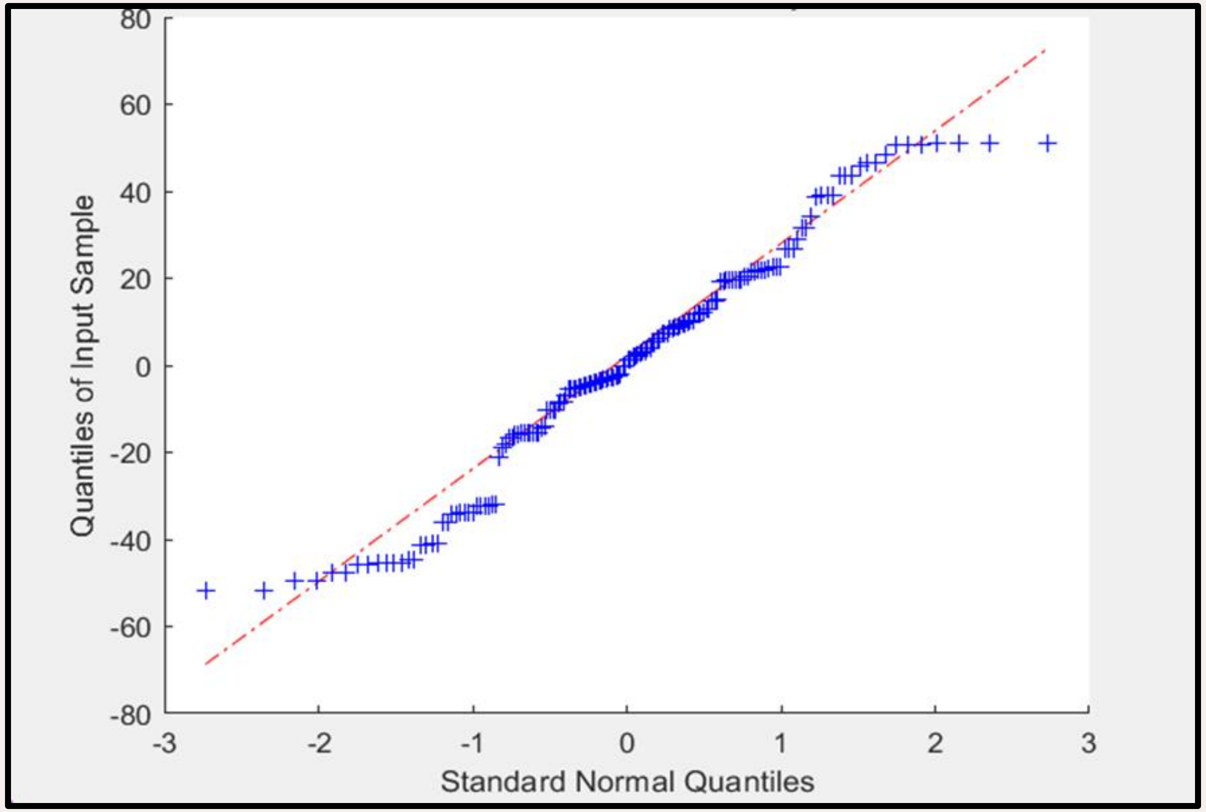
Sensor Testing: Sharp IR GP2Y0A21 – Mapping Team

```
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate      SE        tStat       pValue
                   ________    _______    _______    __________

    (Intercept)    -31.067      8.4849    -3.6615     0.0004546
    x1              2.5499     0.15869     16.068    1.8404e-26


Number of observations: 80, Error degrees of freedom: 78
Root Mean Squared Error: 28.6
R-squared: 0.768,  Adjusted R-Squared: 0.765
F-statistic vs. constant model: 258, p-value = 1.84e-26
```

```
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate      SE        tStat       pValue
                   ________    _______    _______    __________

    (Intercept)    -43.237      7.8724    -5.4923    4.7978e-07
    x1              2.6613     0.14129     18.836    9.4001e-31


Number of observations: 80, Error degrees of freedom: 78
Root Mean Squared Error: 25.2
R-squared: 0.82,  Adjusted R-Squared: 0.817
F-statistic vs. constant model: 355, p-value = 9.4e-31
```

Approach

- Now that I have found the linear regression of my sensor, I was able to make a Q-Q plot to show whether the data set is normally distributed. Doing these tests help us determine which sensors are best for shorter and longer distances.



Electronic Sensor Components

Sensor Testing: Sharp IR GP2Y0A21 – Wall Following Method

Approach

- Within the Sharp IR GP2Y0A21, we have written a code for a wall following method to help our mouse detect when there is a wall too close to our sensors and telling our motors to turn left or continue forward.

```
#include <Adafruit_TinyUSB.h>
#include <SharpIR.h>

const int irSensorPin = A0;       // IR sensor connected to analog pin A0
const int motorLeftForward = 3;    // Motor A forward
const int motorLeftBackward = 4;   // Motor A backward
const int motorRightForward = 5;   // Motor B forward
const int motorRightBackward = 6;  // Motor B backward

// Array to store sensor readings
const int maxReadings = 10;        // Number of readings to store
int sensorValues[maxReadings];     // Array to store sensor readings
int currentIndex = 0;              // Index to keep track of current position

// Thresholds for wall following
const int targetDistance = 200;    // Target distance from wall in sensor val
const int tolerance = 50;          // Tolerance to avoid oscillation in dista

// Variables for sensor value
int sensorValue = 0;

void setup() {
  // Set motor pins as outputs
  pinMode(motorLeftForward, OUTPUT);
  pinMode(motorLeftBackward, OUTPUT);
  pinMode(motorRightForward, OUTPUT);
  pinMode(motorRightBackward, OUTPUT);

  // Start serial communication for debugging
  Serial.begin(9600);
  Serial.println("Wall following with data storage started");
}
```

```
void loop() {
  // Read the value from the IR sensor
  sensorValue = analogRead(irSensorPin);

  // Store the current sensor value in the array
  storeSensorData(sensorValue);

  // Print the stored sensor data every time the array is filled
  if (currentIndex == maxReadings) {
    printSensorData();
    currentIndex = 0;  // Reset index after printing
  }

  // Wall Following Logic
  if (sensorValue < targetDistance - tolerance) {  // Too far from the wall
    moveForward();
  }
  else if (sensorValue > targetDistance + tolerance) {  // Too close to the wall
    turnLeft();
  }
  else {  // Ideal distance from the wall
    moveForward();
  }

  delay(100);  // Small delay for smooth operation
}

// Function to store sensor data in the array
void storeSensorData(int value) {
  // Store the sensor value in the array at the current index
  sensorValues[currentIndex] = value;
```

```
  currentIndex++;
  if (currentIndex >= maxReadings) {
    currentIndex = 0;  // Reset to 0 after storing max readings
  }
}

// Function to print all stored sensor data to the serial monitor
void printSensorData() {
  Serial.println("Stored Sensor Readings:");
  for (int i = 0; i < maxReadings; i++) {
    Serial.print("Reading ");
    Serial.print(i);
    Serial.print(": ");
    Serial.println(sensorValues[i]);
  }
  Serial.println();  // Add a blank line after the readings
}

// Function to move the robot forward
void moveForward() {
  digitalWrite(motorLeftForward, HIGH);
  digitalWrite(motorLeftBackward, LOW);
  digitalWrite(motorRightForward, HIGH);
  digitalWrite(motorRightBackward, LOW);
}

// Function to turn the robot left
void turnLeft() {
  digitalWrite(motorLeftForward, LOW);
  digitalWrite(motorLeftBackward, LOW);
  digitalWrite(motorRightForward, HIGH);
  digitalWrite(motorRightBackward, LOW);
}
```

Electronic Sensor Components

Sensor Testing: Sharp IR GP2Y0A21 – Visited Path Memorizing

Approach

- Within the Sharp IR GP2Y0A21, we have written a code for a visited path memorizing method to help our mouse detect when we have already visited a specific space as well as knowing not to retrace the steps we have already taken.

```
#include <Adafruit_TinyUSB.h>
#include <SharpIR.h>

#define UP    0
#define RIGHT 1
#define DOWN  2
#define LEFT  3

// Define grid size (e.g., 10x10 maze)
#define GRID_WIDTH 12
#define GRID_HEIGHT 12

// Create a 2D array to track visited cells
int visited[GRID_WIDTH][GRID_HEIGHT];

// Sensor pins for the Sharp IR sensor (front detection)
int irPinFront = A0;  // Adjust this to your actual IR sensor

// Motor control pins (adjust based on your hardware setup)
int motorLeftPin = 14;
int motorRightPin = 17;

// Current position of the robot
int currentX = 0;
int currentY = 0;

// Direction the robot is facing (0 = up, 1 = right, 2 = down,
int facingDirection = UP;
```

```
#define DIST_THRESHOLD 15  // cm

// Initialize sensors and motors
void setup() {
  // Set IR sensor pin as input
  pinMode(irPinFront, INPUT);

  // Set motor pins as output
  pinMode(motorLeftPin, OUTPUT);
  pinMode(motorRightPin, OUTPUT);

  // Initialize visited grid to 0 (unvisited)
  memset(visited, 0, sizeof(visited));

  // Mark the starting position as visited
  visited[currentX][currentY] = 1;

  // Start moving
  moveForward();
}

void loop() {
  // Read the Sharp IR sensor to detect the distance to the wall
  int distance = analogRead(irPinFront);  // Analog value from S

  // Convert the sensor reading to a distance in cm (you may nee
  distance = map(distance, 0, 1023, 0, 100); // Simplified mappi

  // If no obstacle in front (sensor reading below threshold)
  if (distance > DIST_THRESHOLD) {
    moveForward();
  } else {
    // Obstacle detected, follow the wall on the left (can be ad
```

```
    if (canTurnLeft() && !isVisited(currentX, currentY, (facingDirection - 1 + 4) % 4)) {
      turnLeft();
      moveForward();
    } else if (canMoveForward() && !isVisited(currentX, currentY, facingDirection)) {
      moveForward();
    } else {
      // If stuck, backtrack or stop (implement backtracking strategy)
      backtrack();
    }
  }
}

// Move forward by one step
void moveForward() {
  // Update current position based on facing direction
  if (facingDirection == UP) {
    currentY--;
  } else if (facingDirection == RIGHT) {
    currentX++;
  } else if (facingDirection == DOWN) {
    currentY++;
  } else if (facingDirection == LEFT) {
    currentX--;
  }

  // Mark the new position as visited
  visited[currentX][currentY] = 1;

  // Move motors forward (this is just a placeholder)
  digitalWrite(motorLeftPin, HIGH);
  digitalWrite(motorRightPin, HIGH);
  delay(500);  // Adjust delay for actual movement speed
  digitalWrite(motorLeftPin, LOW);
```

Electronic Sensor Components

Sensor Testing: Sharp IR GP2Y0A21 – Visited Path Memorizing, Continued Code….

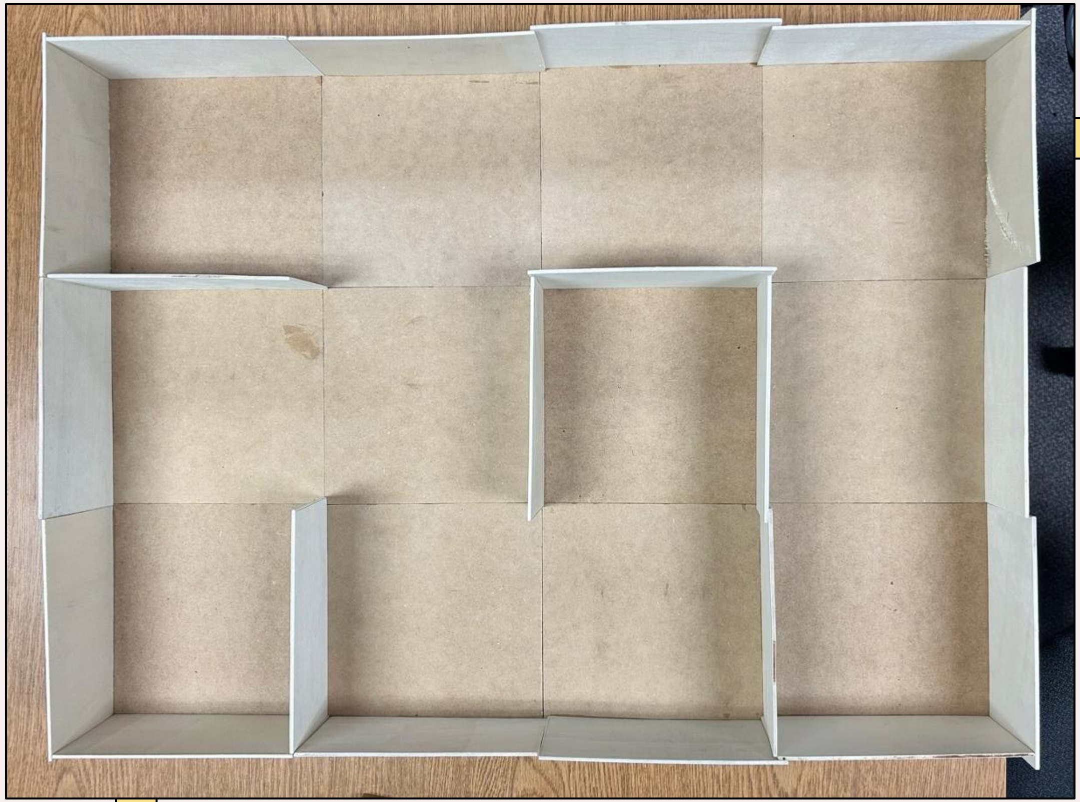
```
  digitalWrite(motorRightPin, LOW);
}

// Check if the robot can turn left (there's no obstacle)
bool canTurnLeft() {
  int newDirection = (facingDirection - 1 + 4) % 4;
  return !isObstacleInDirection(newDirection);
}

// Check if the robot can move forward (there's no obstacle)
bool canMoveForward() {
  return !isObstacleInDirection(facingDirection);
}

// Check if the robot has already visited a position in a giv
bool isVisited(int x, int y, int direction) {
  // You need to check if the next position in the given dire
  if (direction == UP) {
    return visited[x][y - 1] == 1;
  } else if (direction == RIGHT) {
    return visited[x + 1][y] == 1;
  } else if (direction == DOWN) {
    return visited[x][y + 1] == 1;
  } else if (direction == LEFT) {
    return visited[x - 1][y] == 1;
  }
  return false;
}

// Helper function to check if there is an obstacle in the gi
bool isObstacleInDirection(int direction) {
  // Logic to check for obstacles in the given direction usin
```

```
  if (direction == UP) {
    return analogRead(irPinFront) < DIST_THRESHOLD;  // Front s
  } else if (direction == RIGHT) {
    return false;  // Assume no sensor on the right side (can a
  } else if (direction == DOWN) {
    return analogRead(irPinFront) < DIST_THRESHOLD;  // Same se
  } else if (direction == LEFT) {
    return false;  // Assume no sensor on the left side (can ad
  }
  return false;
}

// Turn the robot 90 degrees to the left
void turnLeft() {
  facingDirection = (facingDirection - 1 + 4) % 4;
  // Motor control code to turn left (this is just a placeholde
}

// Turn the robot 90 degrees to the right
void turnRight() {
  facingDirection = (facingDirection + 1) % 4;
  // Motor control code to turn right (this is just a placehold
}

// Backtrack if the robot is stuck
void backtrack() {
  // Simple backtracking: turn 180 degrees and move back to the
  turnLeft();
  turnLeft();
  moveForward();
}
```

Small Scale Testing Maze Design

Mapping Team

- As a team, we have constructed a 3x4 maze using the proper measurement requirements of 8x8in a square. We have exactly 12 square units. We have created the maze to test the *Micromouse*, when it undergoes different types of scenarios. The *Micromouse* can start at any corner and navigate its way to the middle.



References

- *IEEE. "Micromouse Contest Rules." IEEE Region 2 Student Activities Committee*, 2018, https://ewh.ieee.org/reg/2/sac-18/MicromouseRules.pdf.
- Zhen, Gan et al. "Autonomous Maze Solving Robot." (2011).
- *Finan, P. "Mleng in Electronics and Electrical Engineering*." Loughborough University, May 2000.
- Jianping Cai; Zengwei Zheng; Minyi Guo. Inderscience Publishers. "A calibration algorithm for maze micromouse continuous smooth turning
- "Difference Between AC & DC Motor." Electrical Technology, www.electricaltechnology.org/2020/06/difference-between-ac-dc-motor.html.
- Hermitage Automation, "Difference Between AC and DC Motor", https://hermitageautomation.com/difference-between-ac-and-dc-motor/
- Pololu, "Pololu – Item #: 2364", https://www.pololu.com/product/2364