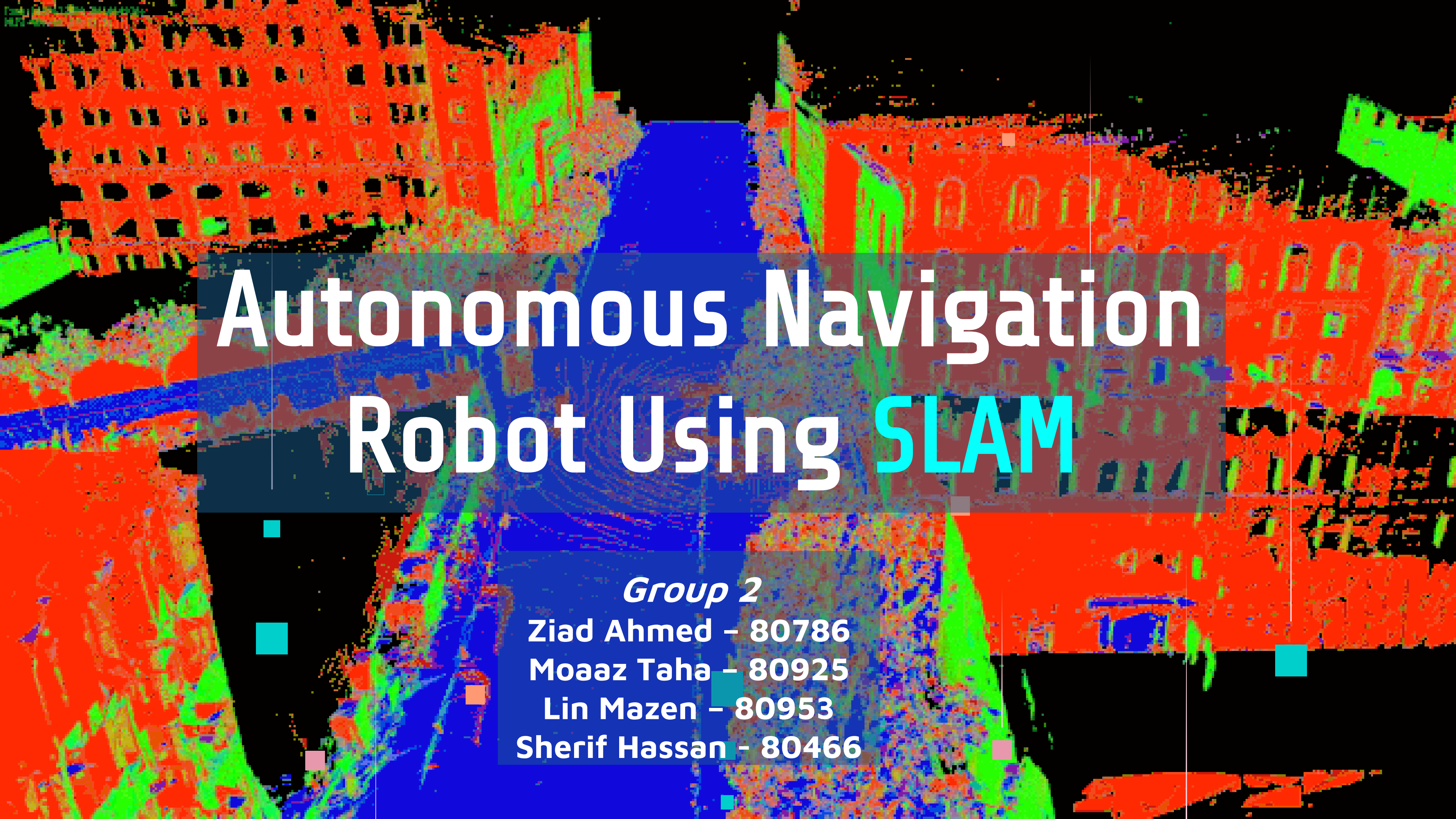


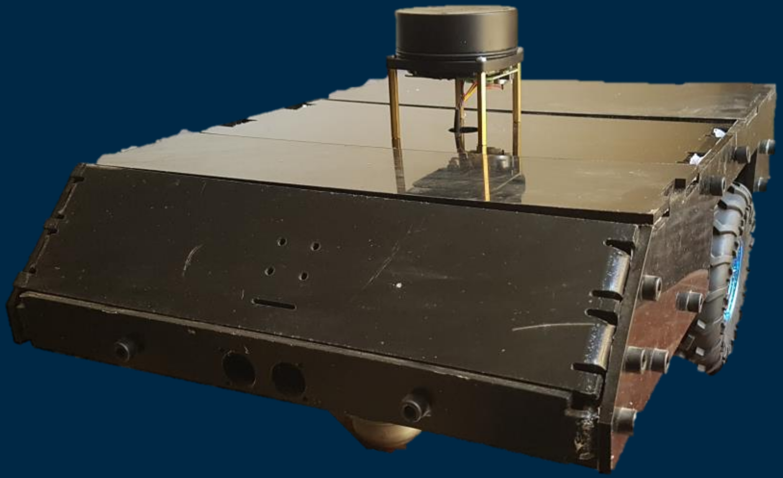
Autonomous Navigation Robot Using SLAM

Group 2
Ziad Ahmed - 80786
Moaaz Taha - 80925
Lin Mazen - 80953
Sherif Hassan - 80466

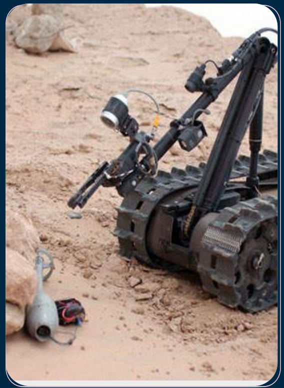
INTRODUCTION

- Our project is an Autonomous Navigation Robot using ROS to explore indoor environments by creating a map using SLAM and use path planning according to the created map.

- Autonomous Navigation Robots are used in many applications, such as field scanning, warehouse management, and transportation.



ADVANTAGES



Safety

Efficiency

Convenience

Innovation

AGENDA

01

MECHANICAL DESIGN

02

ELECTRICAL DESIGN

Sensors &
Schematics

03

SOFTWARE

ROS –
Ros2_control –
SLAM
- Path Planning

04

ADDITIONS

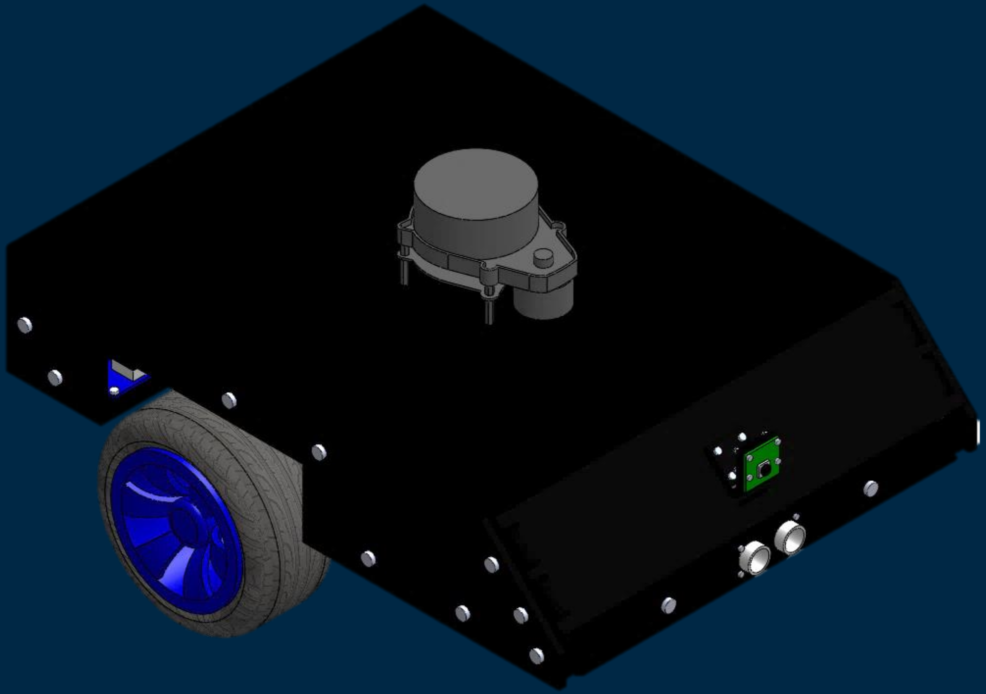
Timeline –
Business
Opportunities

MECHANICAL DESIGN

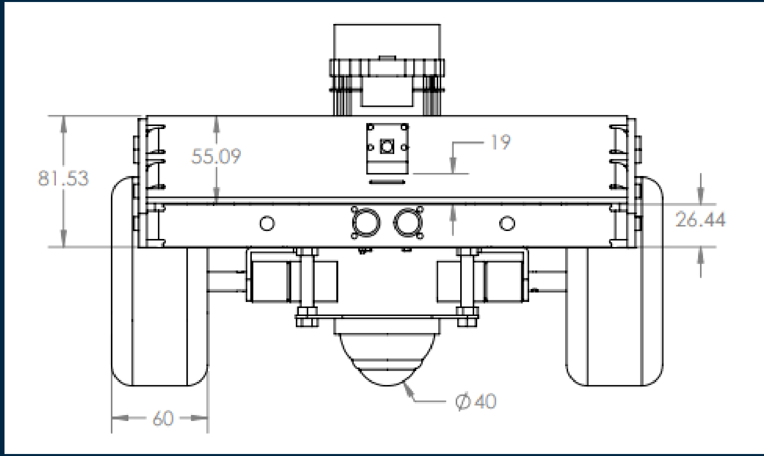
01

CAD Model

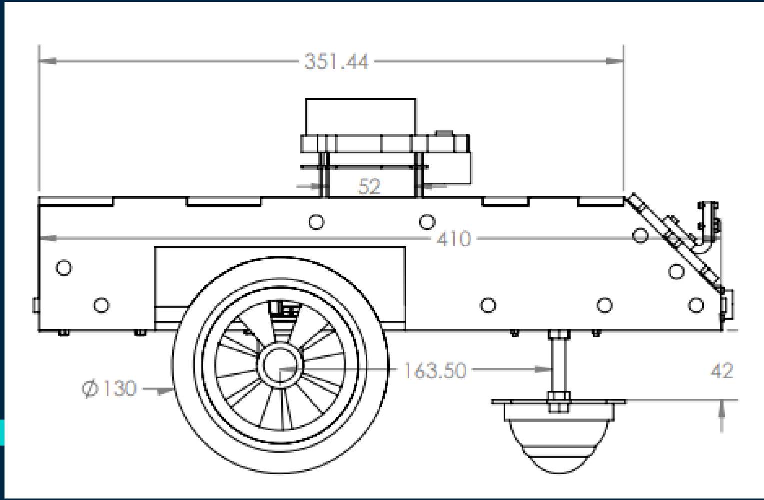
ISOMETRIC



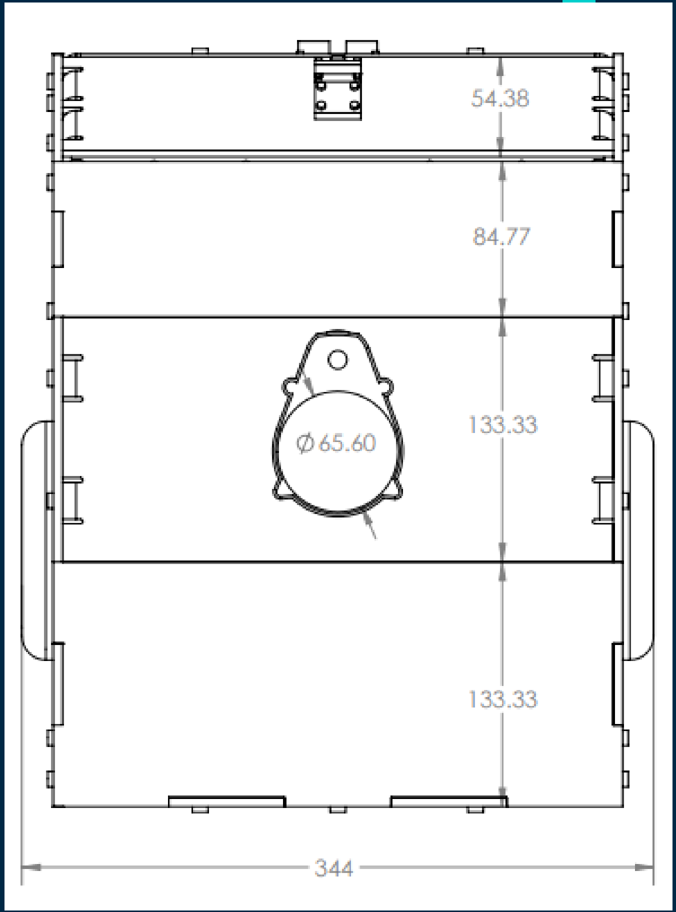
VIEWS



Front view



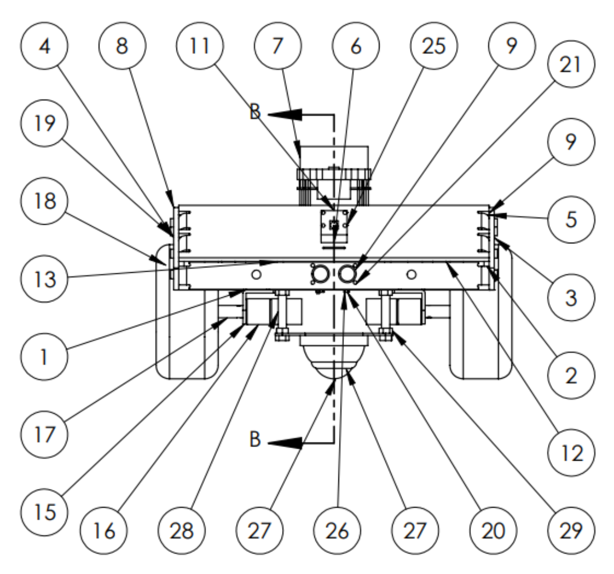
Side view

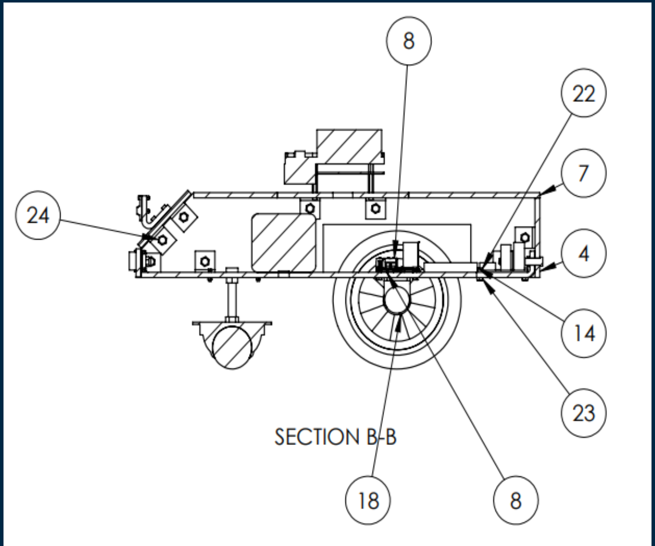


Top view

PARTS

ITEM NO.	PART NUMBER	QTY.
1	Bottom chassis	1
2	Front chassis	1
3	Side chassis	2
4	Back chassis	1
5	Top chassis	4
6	Camera holder	1
7	LIDAR	1
8	L298N Stepper Motor Driver	1
9	Ultrasonic	1
10	Raspberrypi	1
11	Camera	1
12	Converter	2
13	Battery	1
14	breadboard	2
15	Motor_Mount	2
16	motor	2
17	Coupler	2
18	Wheel	2
19	Screw6mm	23
20	Screw3mm	16
21	Screw2mm	12
22	Screw4mm	6
23	Bolt4mm	6
24	Bolt6mm	23
25	Bolt2mm	10
26	Bolt3mm	8
27	CastorWheel	1
28	CastorScrew	2
29	CastorBolt	6





CALCULATIONS

CALCULATIONS

$$\sum Moment_y = zero$$

$$\therefore 9.8(70.5) + 15.327(136.4) - (2X - 9.8)(184) = zero$$

$$\therefore X = 12.4584N\#$$

$$\sum Forces = zero$$

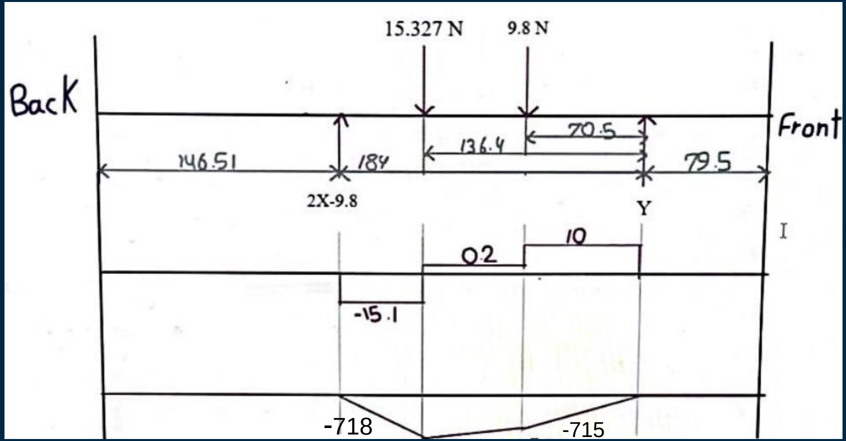
$$\therefore 2(12.4584) - 9.8 + Y = 25.127\ N$$

$$\therefore Y = 10\ N\#$$

$$\sigma = \frac{F}{A} = \frac{34.927}{127100} = 2.75 \times 10^{-4}\ N/mm^2$$

$$\varepsilon = \frac{\sigma}{E} = \frac{2.75 \times 10^{-4}}{2760} = 9.957 \times 10^{-8}$$

$$\delta = \varepsilon l = 9.957 \times 10^{-8} \times 5 = 4.976 \times 10^{-7} mm$$



Loads Distribution

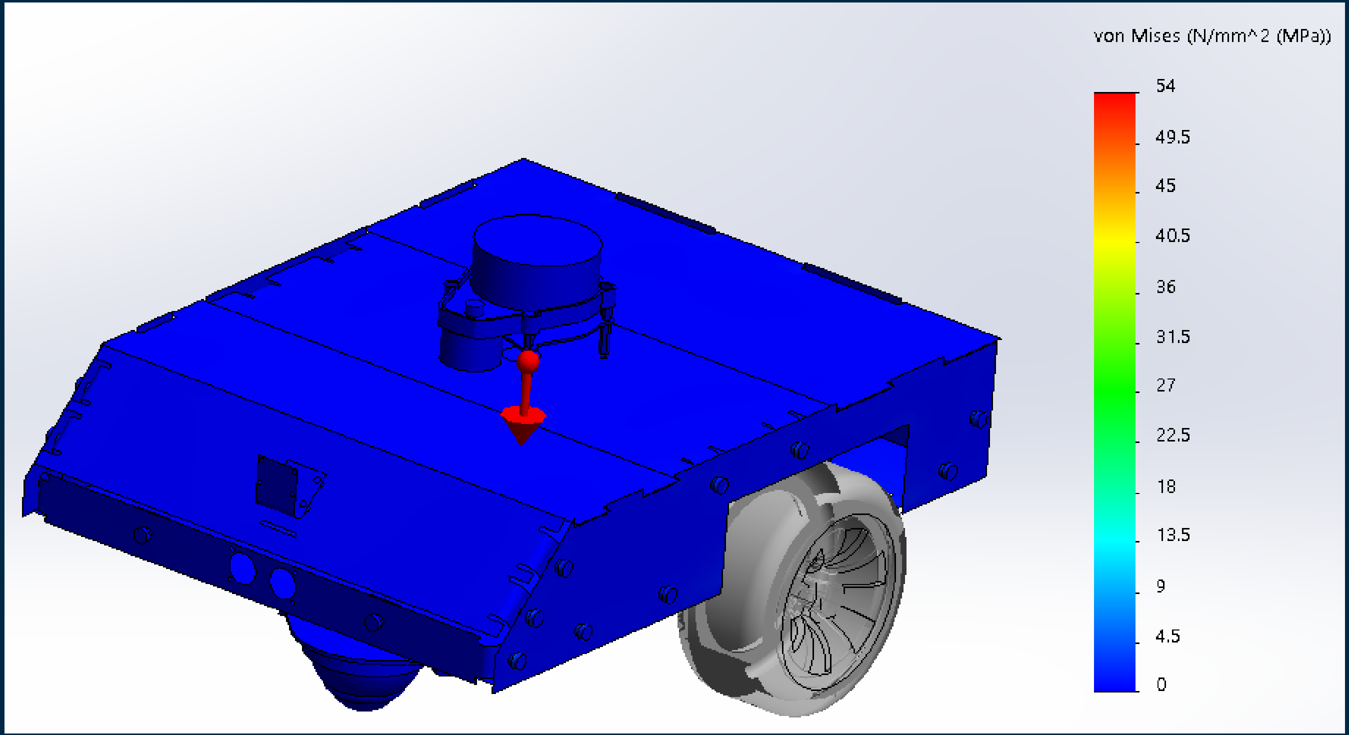
MATERIAL SELECTION

Acrylic Properties

Property	Value
Technical name	Acrylic (PMMA)
Relative Density	1.19 g/cm3
Minimum Service Temperature	-40ºC
Maximum Service Temperature	80ºC
Melt Temperature	130°C (266°F)
Chemical Formula	$(C_5H_8O_2)_n$
Tensile Strength	65 MPa (9400 PSI)
Shrink Rate	0.2 – 1% (.002 – .01 in/in)
Flexural Strength	90 MPa (13000 PSI)
Specific Gravity	1.18
Typical Injection Mold Temperature	79-107°C (175-225°F)
Water absorption (immersion 24 hours)	0.20%
Light Transmittance	92%
Heat deflection temperature	95°C (203°F) at 0.46 MPa (66 PSI)

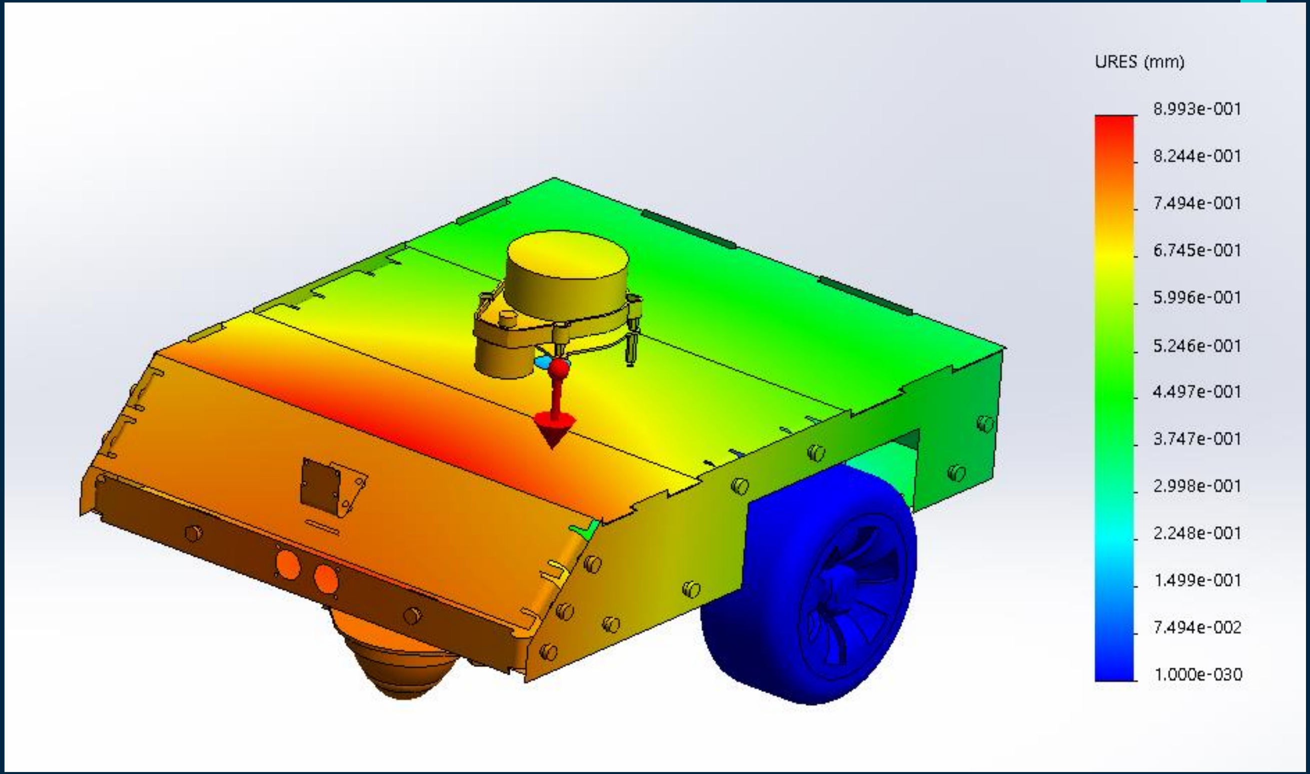
CAD ANALYSIS

STRESS ANALYSIS



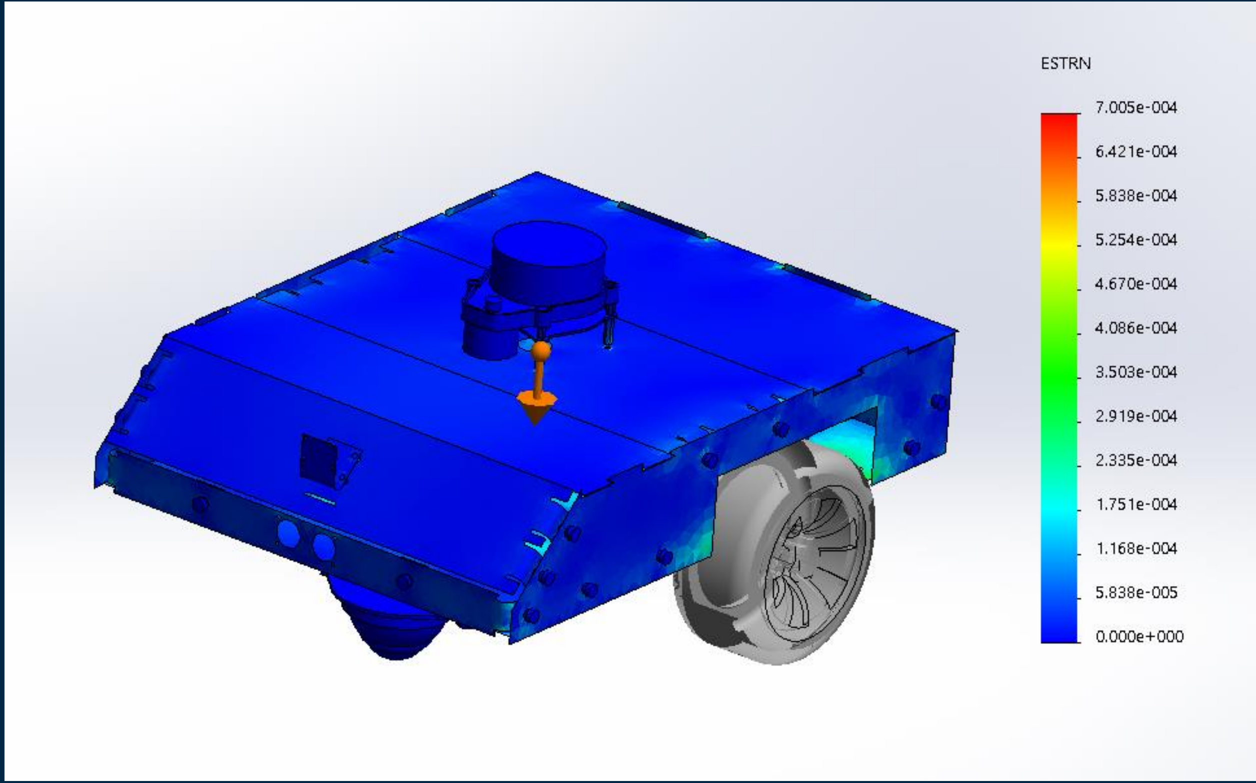
Type	Min	Max
VON: von Mises Stress	0 N/mm^2 (MPa) Node: 47794	54 N/mm^2 (MPa) Node: 8603

DISPLACEMENT ANALYSIS



Type	Min	Max
URES: Resultant Displacement	0.000e+000mm Node: 47794	8.993e-001mm Node: 94044

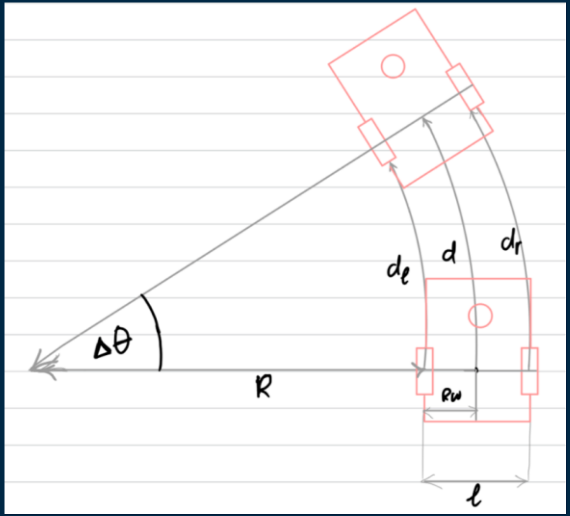
STRAIN ANALYSIS



Type	Min	Max
ESTRN: Equivalent Strain	0.000e+000 Element: 25237	7.005e-004 Element: 33195

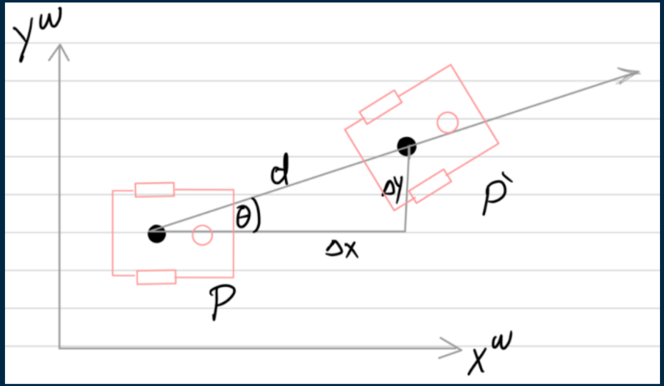
KINAMATICS

MODELING: FOR A WHEELED ROBOT



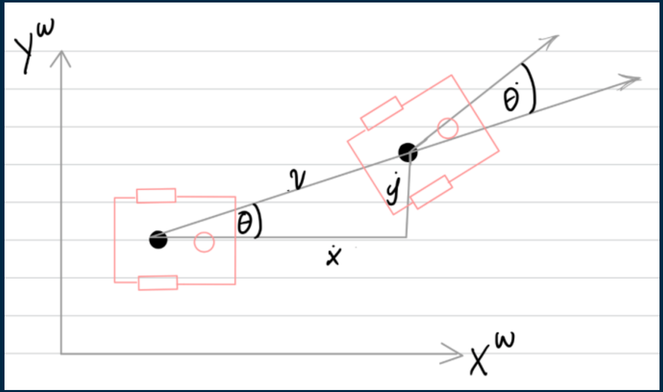
$$\Delta\theta = \frac{d_r - d_l}{2R_W}$$

$$d = \frac{d_r + d_l}{2}$$



$$\grave{P} =$$

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} d\cos\theta \\ d\sin\theta \\ \Delta\theta \end{bmatrix}$$



$$\dot{v} =$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \dfrac{r(v_r - v_l)}{2R_W} \end{bmatrix}$$

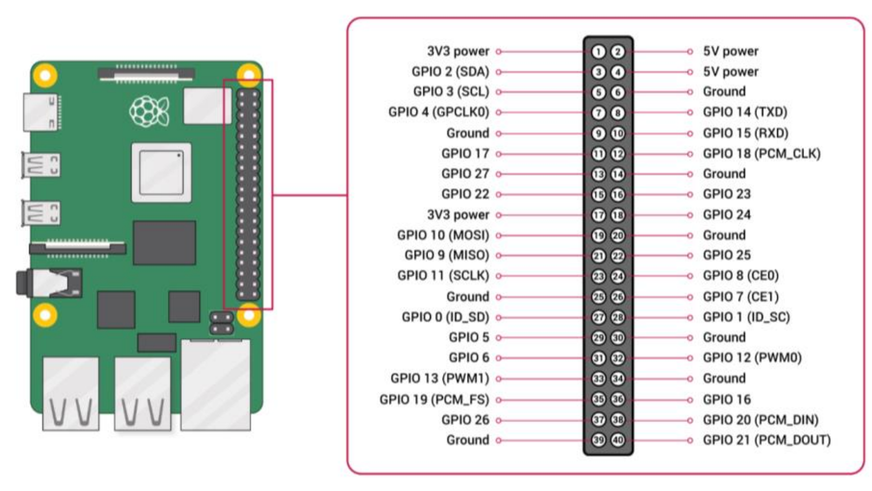
PREVIOUS PROJECT

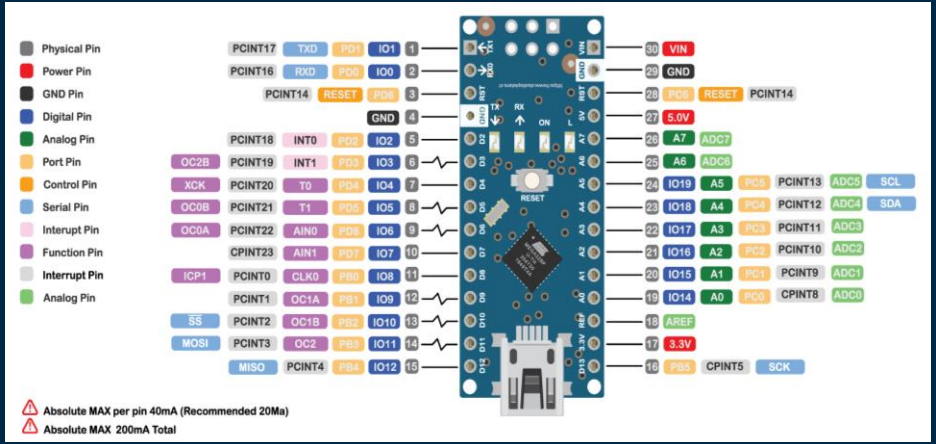
SWARM ROBOTS

ELECTRICAL DESIGN

02

CONTROLLER





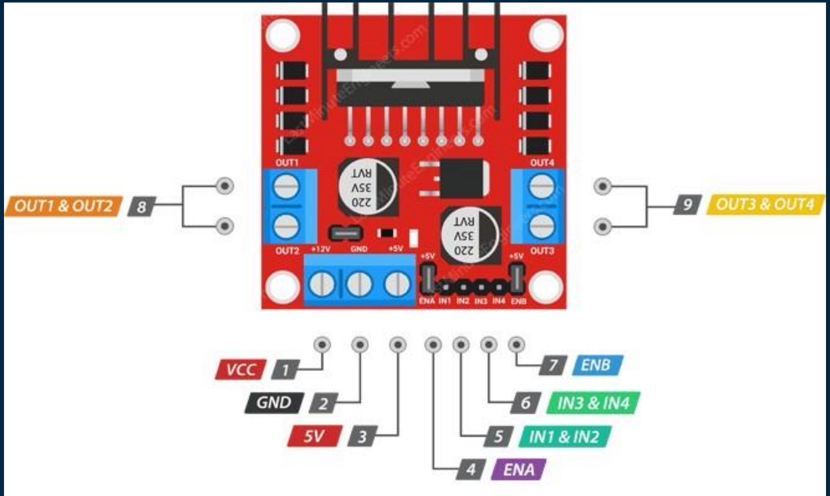
- Raspberry Pi 4 on-board computer

Arduino Nano Microcontroller

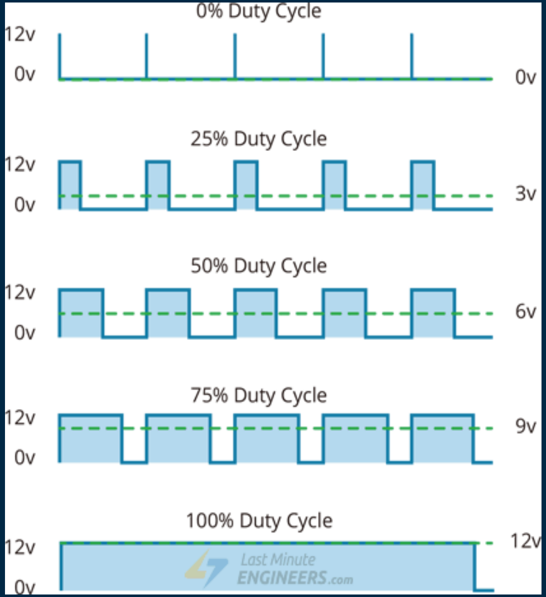
DC MOTOR

Specification	Value
Operating Voltage	+12V
No Load Current	100mA
No Load Speed	200 rpm
Rated Load Current	330mA
Rated Load Speed	140 rpm
Rated Torque	1.43 kg.cm
Stall Torque	5 kg.cm
Output Power	2 W
Stall Current	1.5A

MOTOR CONTROL



L298N Motor Driver

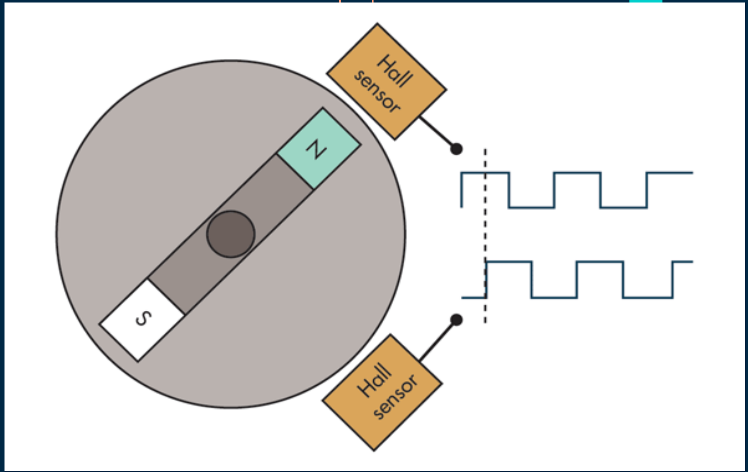


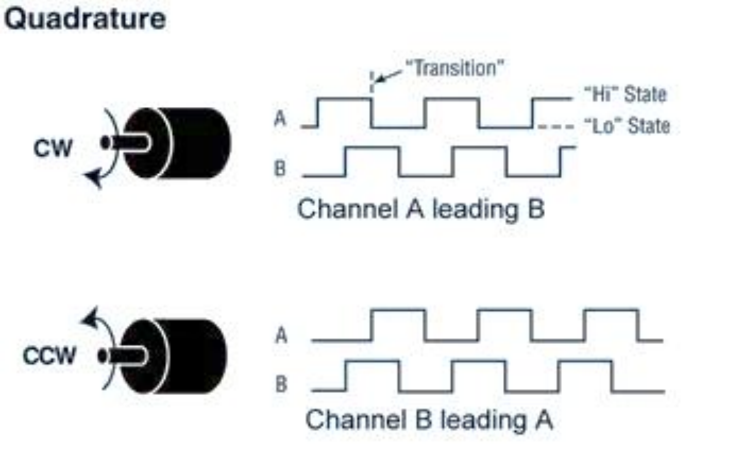
PWM: Speed Control

ENCODER

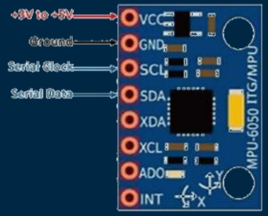
- We used encoder for tracking the robot position by calculating the number of revolution each motor made.

- Two channel output from encoder
  - One is connected to an external interrupt pin .
  - Another is connected to a digital input pin .
  - The two channels are responsible for detecting the rotation according to the lead of each channel's signal

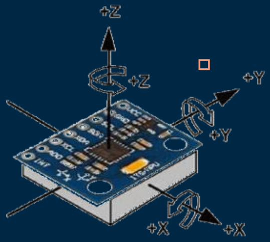




INERTIAL MEASUREMENT UNIT (IMU)



MPU6050 Pinout



MEMS

Micro Electro-Mechanical System.

GYROSCOPE

Provides a measure for the angular rate.

ACCELEROMETERS

Provides a measure for force/acceleration.

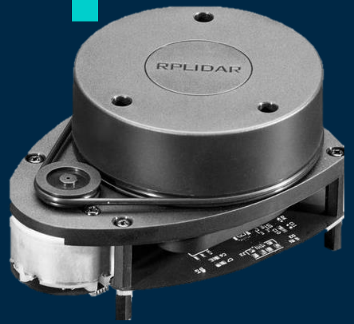
MAGNETOMETERS

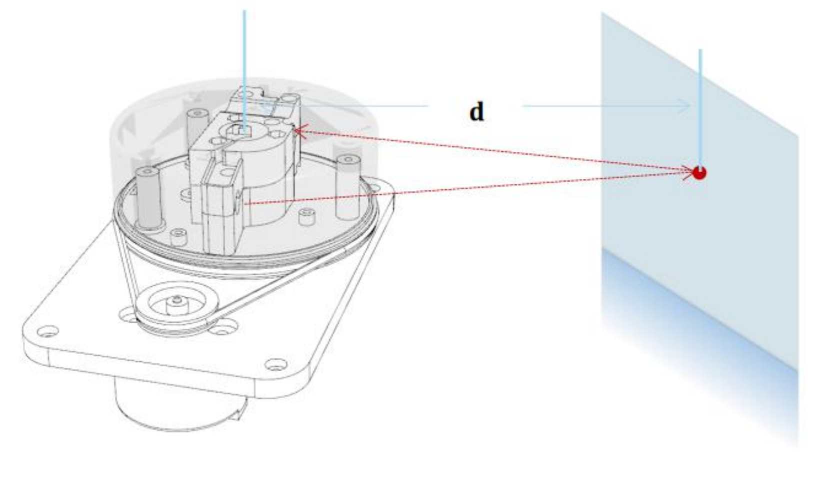
Provides a measure for the magnatic field forces.

LIDAR

- RPLIDAR A1 is a 2D lidar used to sense environment using laser signals.

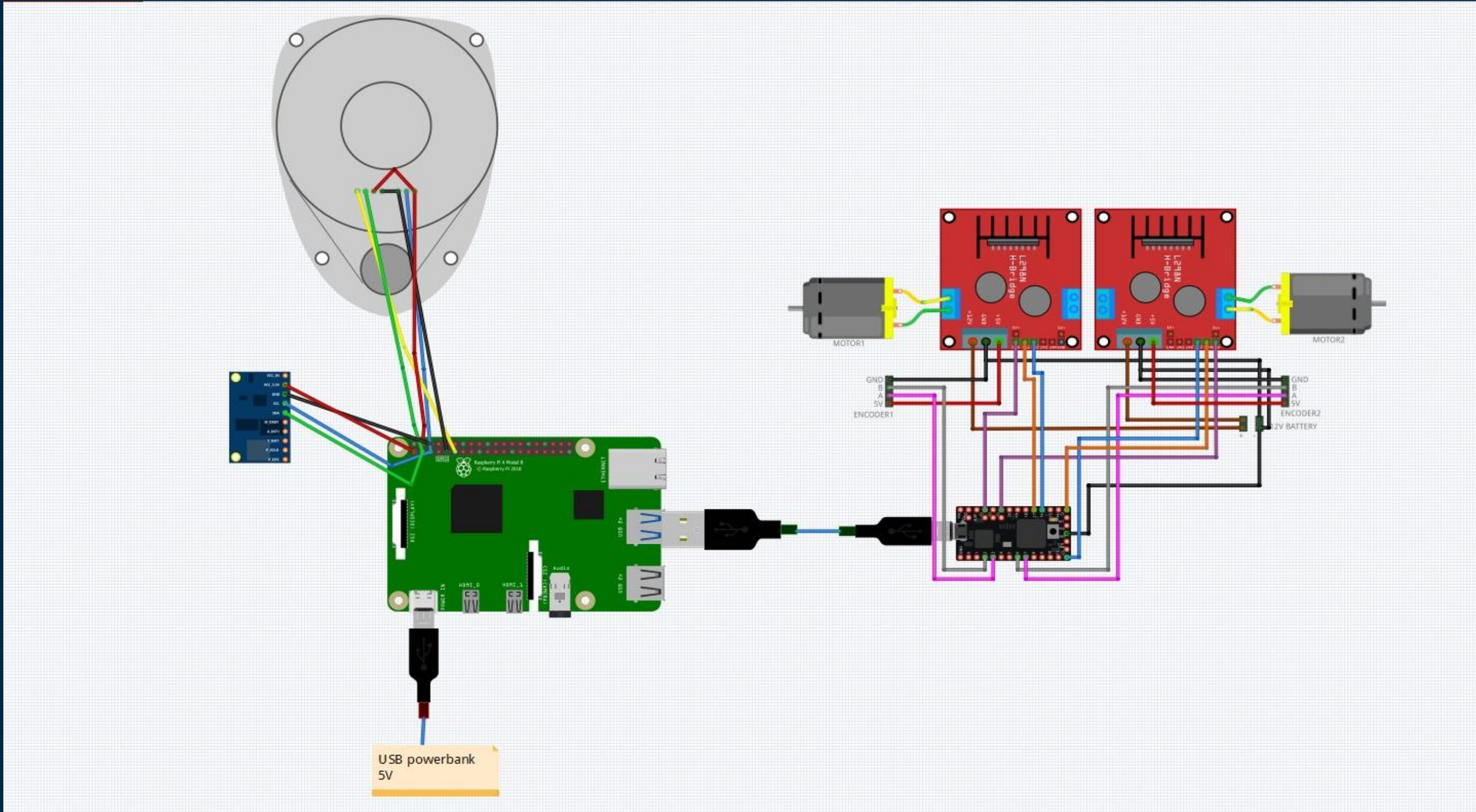
- The Lidar is essential in the process of constructing the map with SLAM.



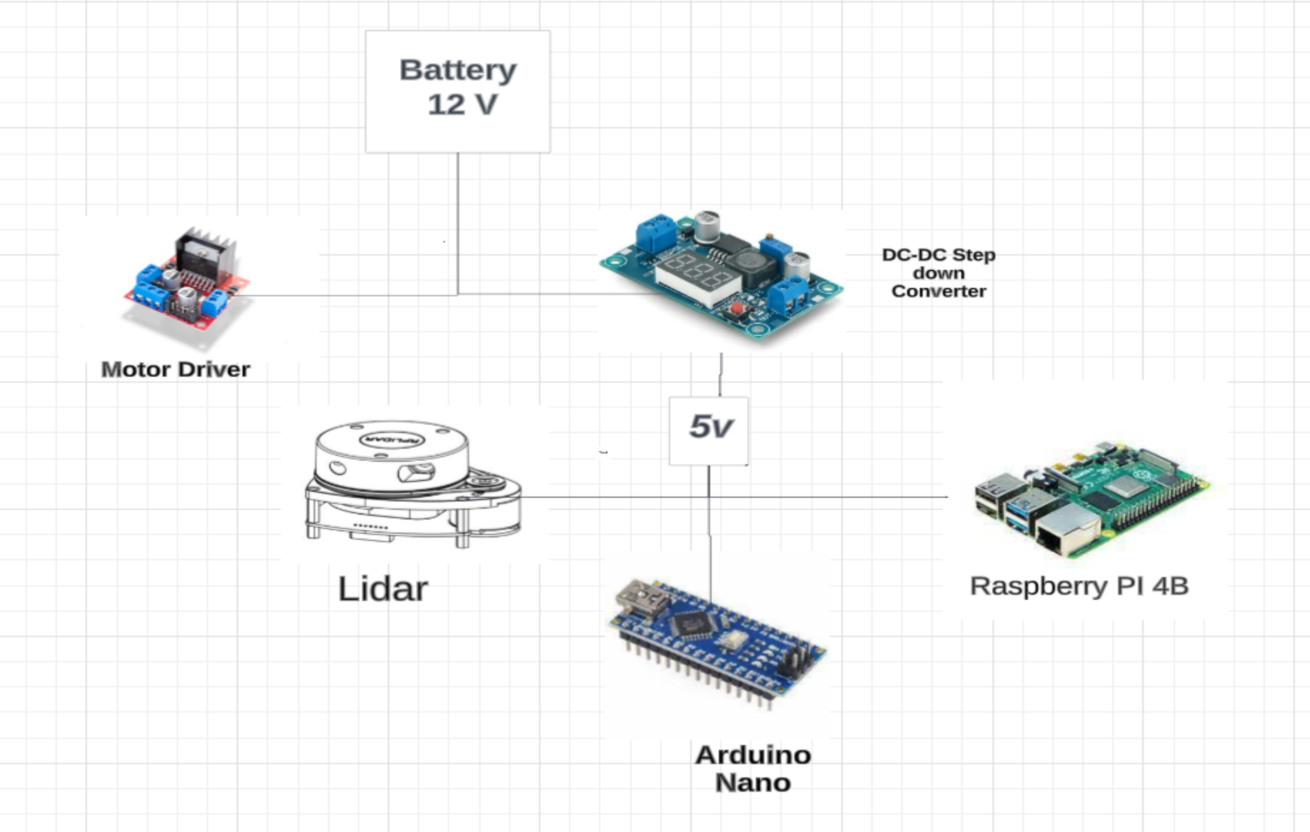


Typical Distance Range	**0.15~12m**
Sample Frequency	8000 Hz
Communication Interface	UART @ 115200bps
Power	5V
Working Current	100mA
Power Consumption	0.5W
Scan Range	360°

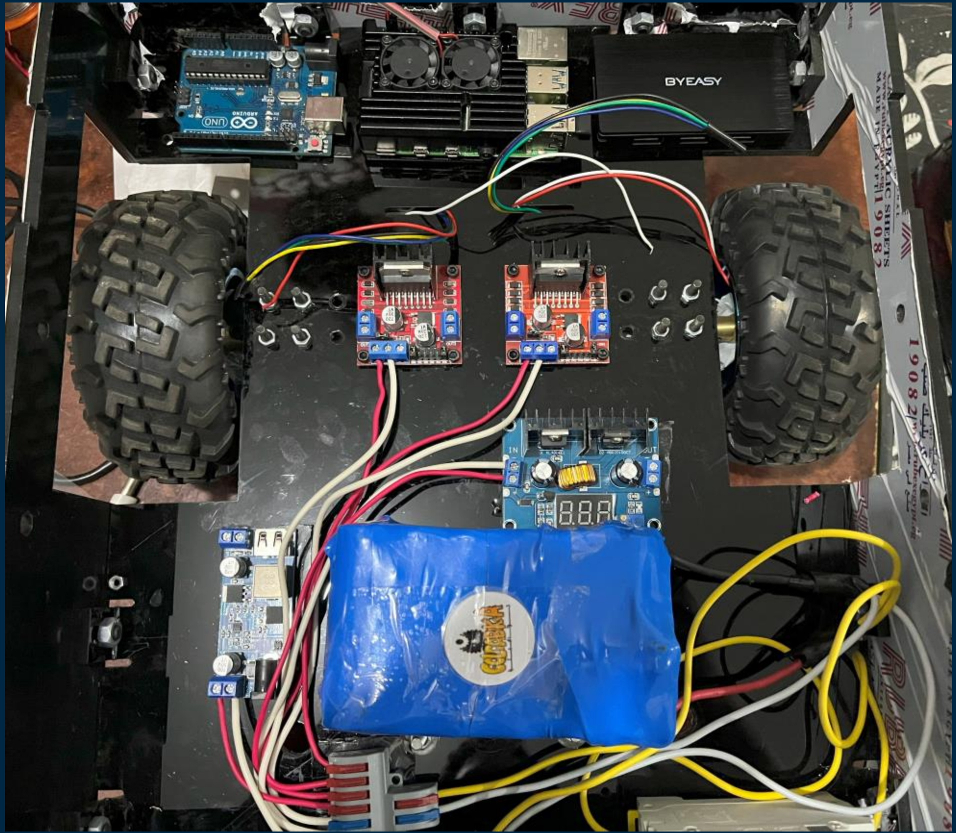
CIRCUIT SCHEMATIC



POWER CIRCUIT



ROBOT WIRING



POWER CONSUMPTION

Components	Description	Max(mAh)
2 Motor	$E_{total} = \frac{2 \times 12V \times 1500mA \times 1h \times 1.3}{12V}$	3900
Arduino Nano	$E_{total} = \frac{5V \times 500mA \times 1h \times 1.3}{5V}$	650
MPU 6050	$E_{total} = \frac{3.5V \times 39mA \times 1h \times 1.3}{5V}$	36.50
Raspberry PI 4B	$E_{total} = \frac{5.25V \times 3000mA \times 1h \times 1.3}{5V}$	4305.50
Raspberry PI Camera	$E_{total} = \frac{1.8V \times 100mA \times 1h \times 1.3}{5V}$	47
Lidar	$E_{total} = \frac{5V \times 600mA \times 1h \times 1.3}{5V}$	780
Total		9719

BOM

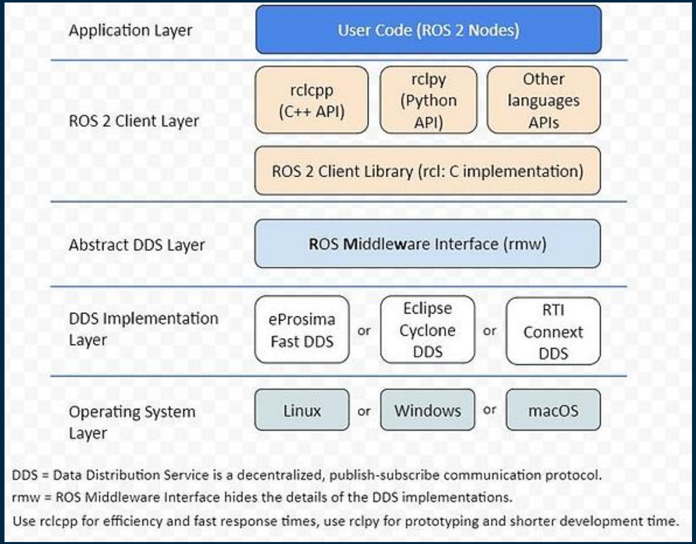
Component	No. of items	Price (L.E.)
Arduino Nano	1	220
Raspberry PI 4B	1	5000
Lidar	1	6000
DC Motor (12V. 200 RPM, 1.43 kg.cm torque)	2	770
Motor Driver	2	130
Battery **(Lithium – Ion Battery 12 Volt and 10000 mAh)**	1	800
XL4016 DC-DC Step down Converter Module	1	75
Mech & electrical components		300
IMU (MPU 6050Acce & Gyro)	1	120
Raspberry PI Camera	1	850
Total		***14265***

SOFTWARE

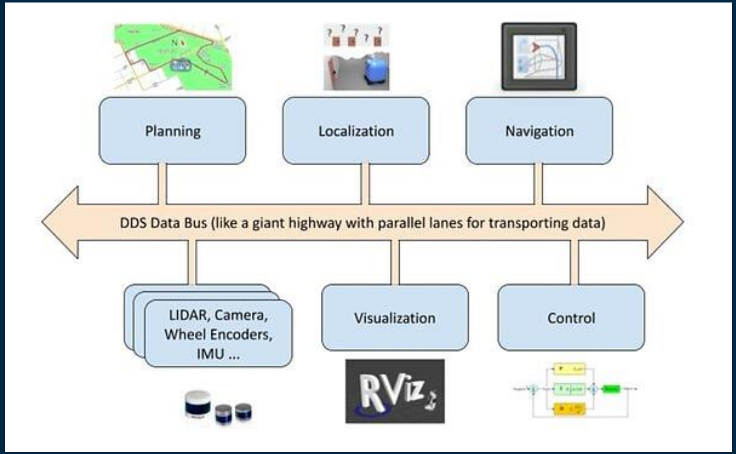
03

ROBOT OPERATING SYSTEM

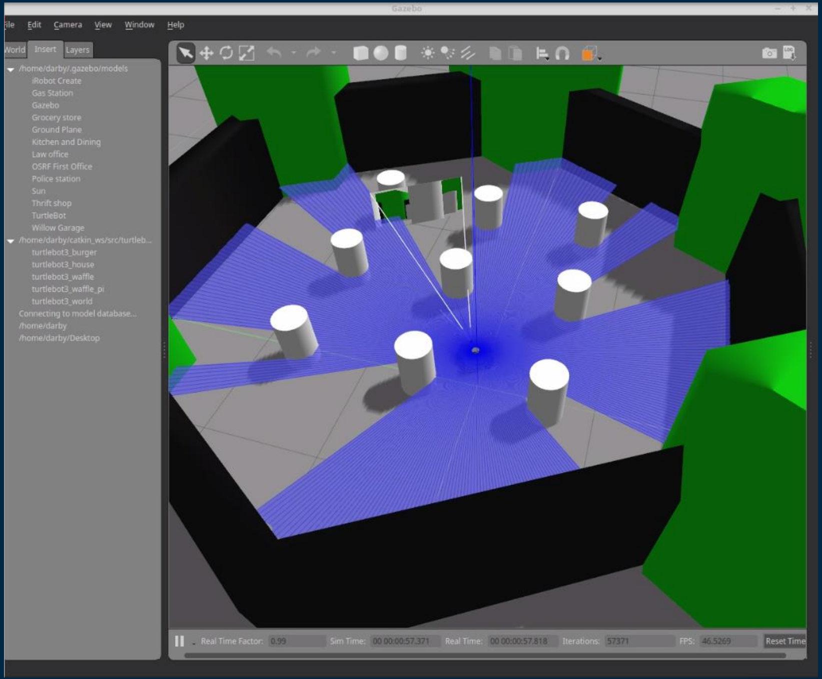
ROS FOXY

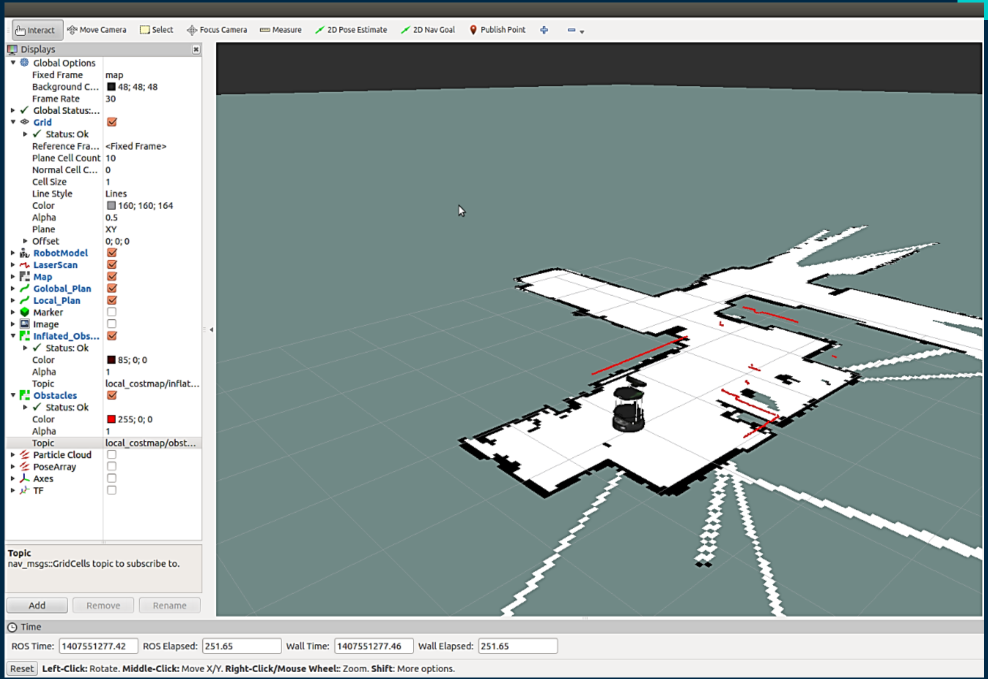

Application Layer
User Code (ROS 2 Nodes)
ROS 2 Client Layer
rclcpp (C++ API)
rclpy (Python API)
Other languages APIs
ROS 2 Client Library (rcl: C implementation)
Abstract DDS Layer
ROS Middleware Interface (rmw)
DDS Implementation Layer
eProsima Fast DDS
or
Eclipse Cyclone DDS
or
RTI Connext DDS
Operating System Layer
Linux
or
Windows
or
macOS
DDS = Data Distribution Service is a decentralized, publish-subscribe communication protocol.
rmw = ROS Middleware Interface hides the details of the DDS implementations.
Use rclcpp for efficiency and fast response times, use rclpy for prototyping and shorter development time.


ROS
FOXY FITZROY


Planning
Localization
Navigation
DDS Data Bus (like a giant highway with parallel lanes for transporting data)
LIDAR, Camera, Wheel Encoders, IMU ...
Visualization
Control
RViz

TOOLS



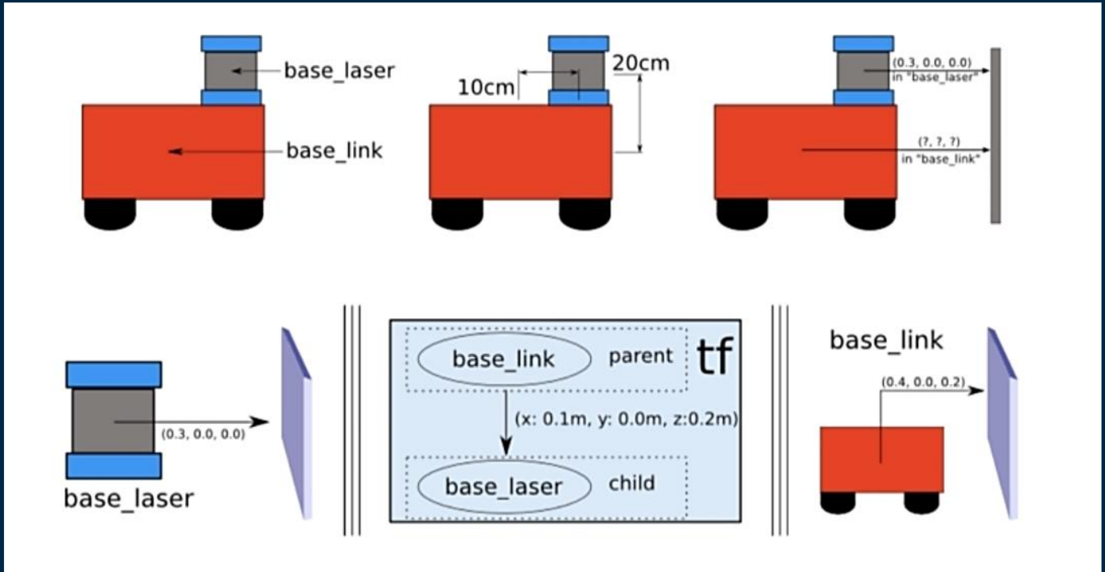


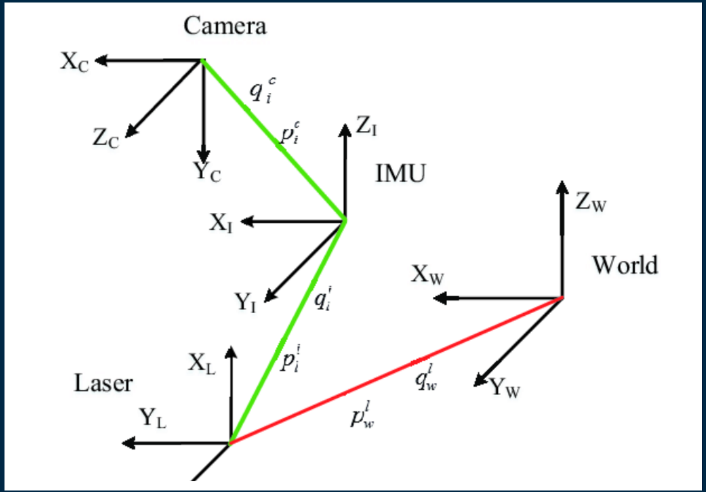
GAZEBO

RViz

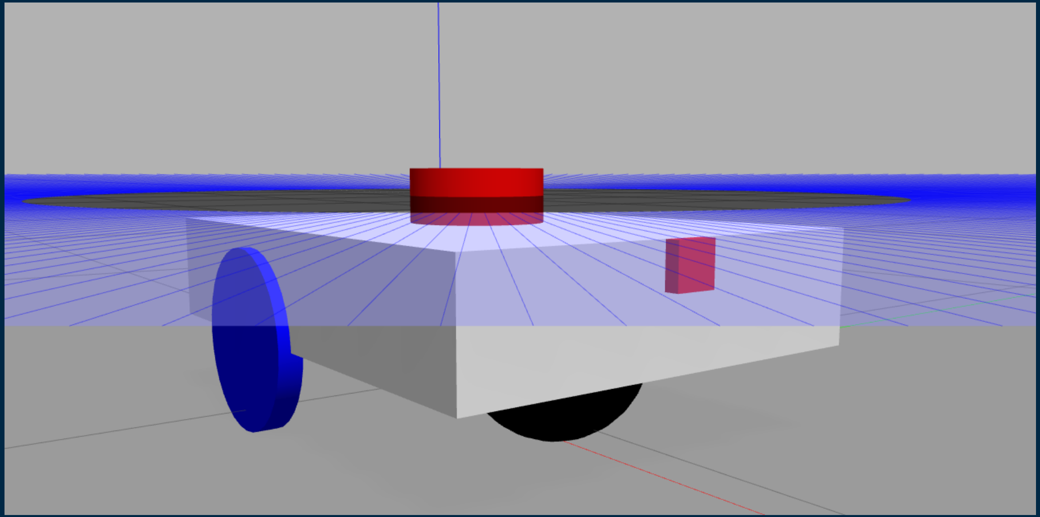
TRANSFORM CONFIGURATION

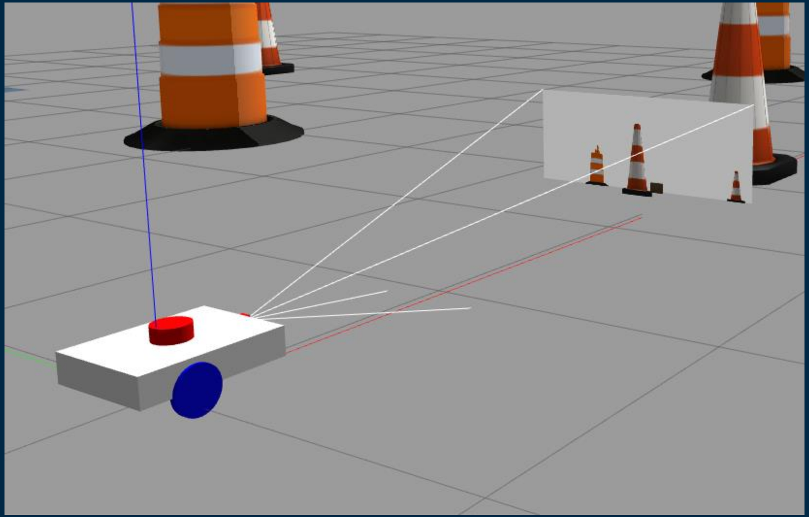
- A robot must publish tf transform to describe the relation between coordinate frames.



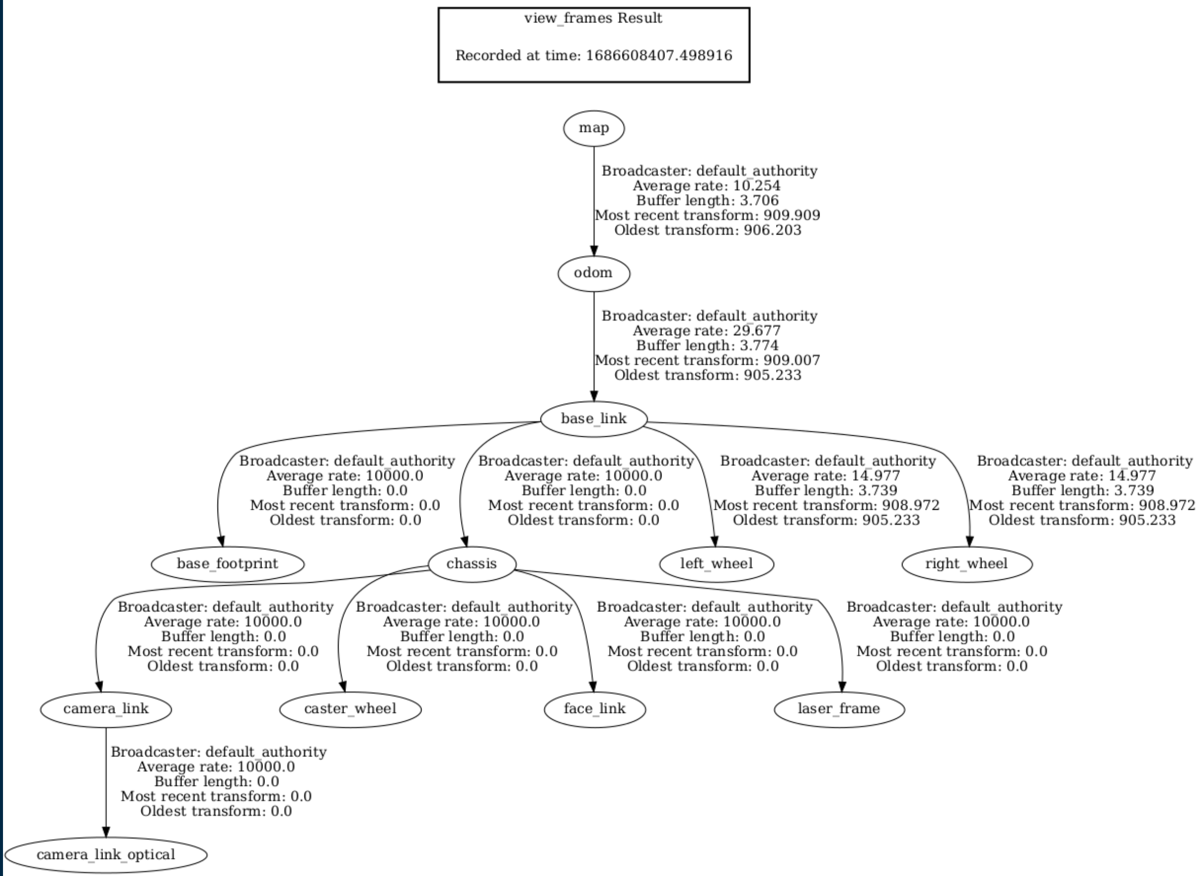


UNIFIED ROBOTICS DESCRIPTION FORMAT (URDF)



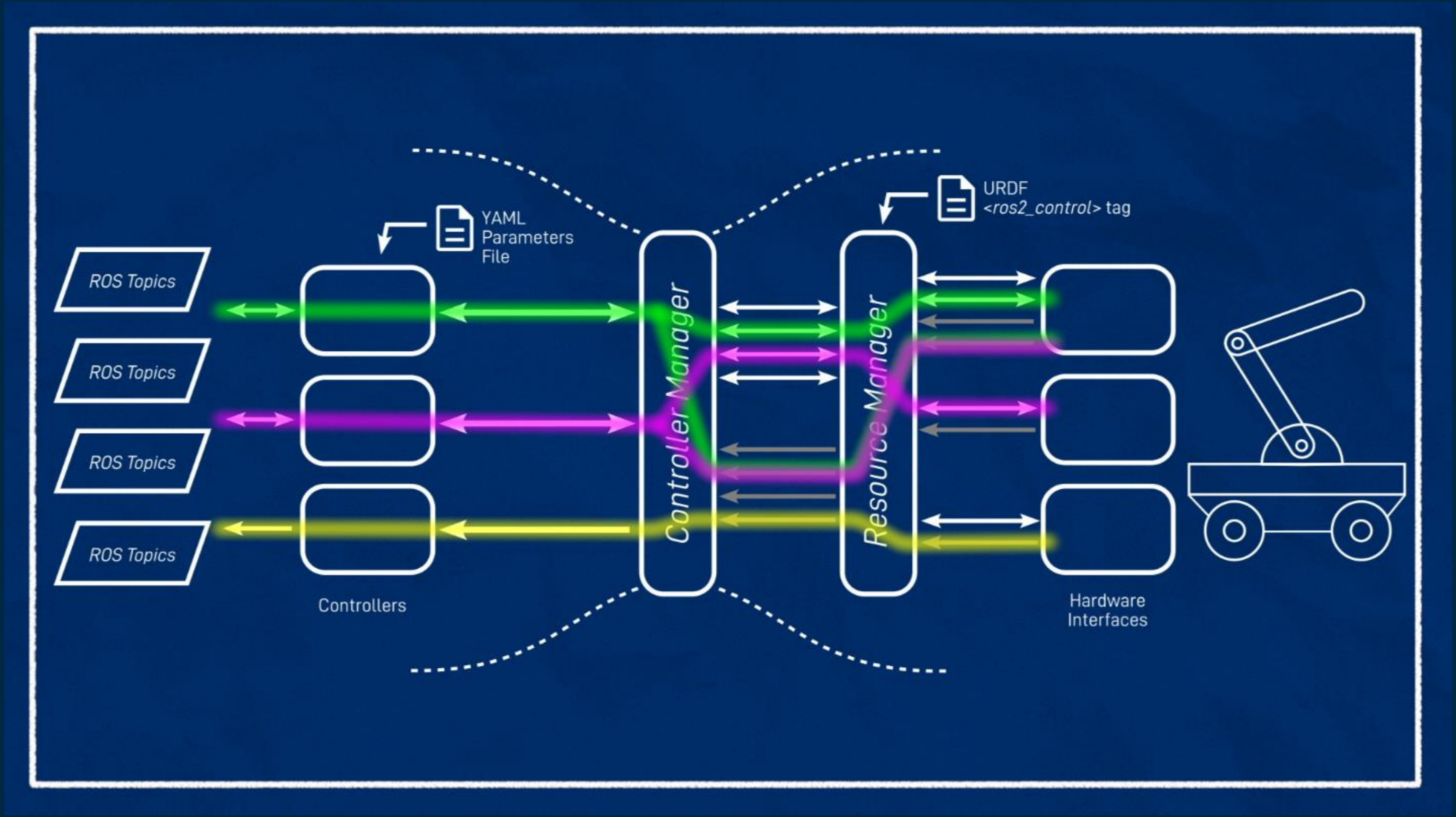


FRAMES & NODES

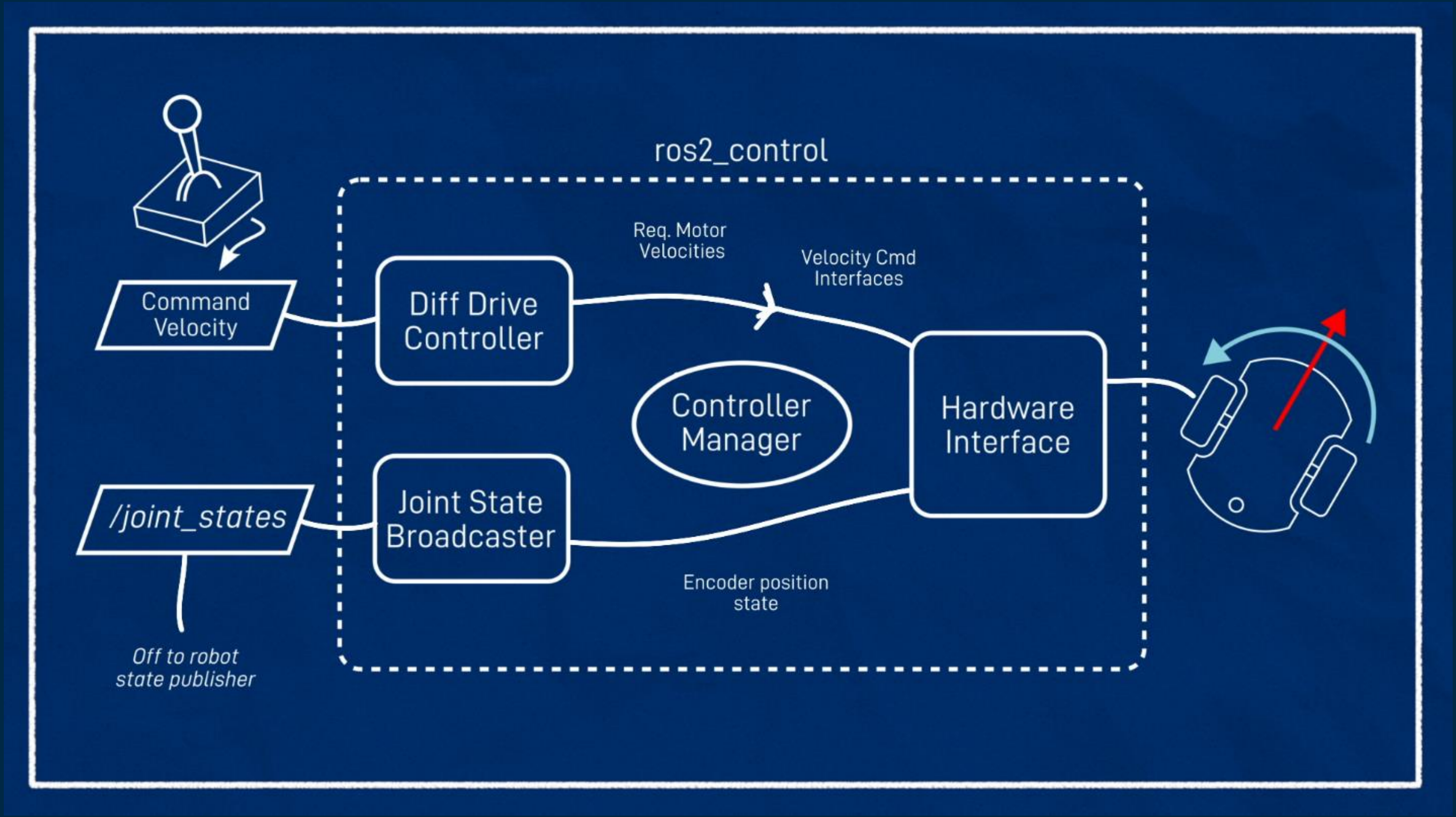


Robot Motion Control

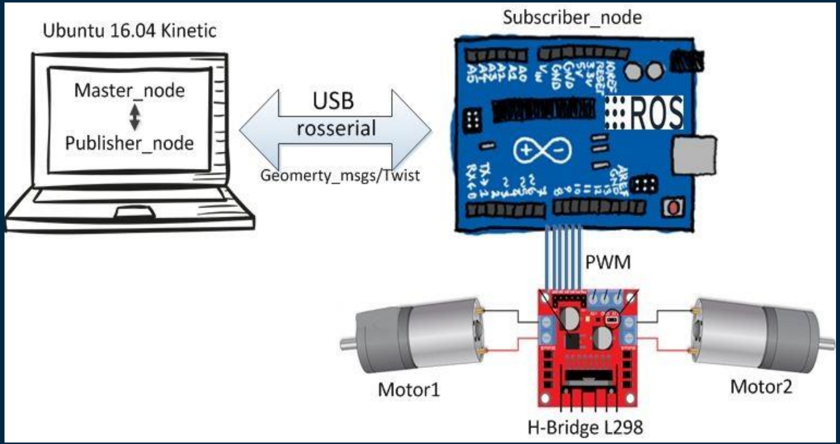
ROS2_CONTROL OVERVIEW



ROS2_CONTROL: DIFFERENTIAL DRIVE

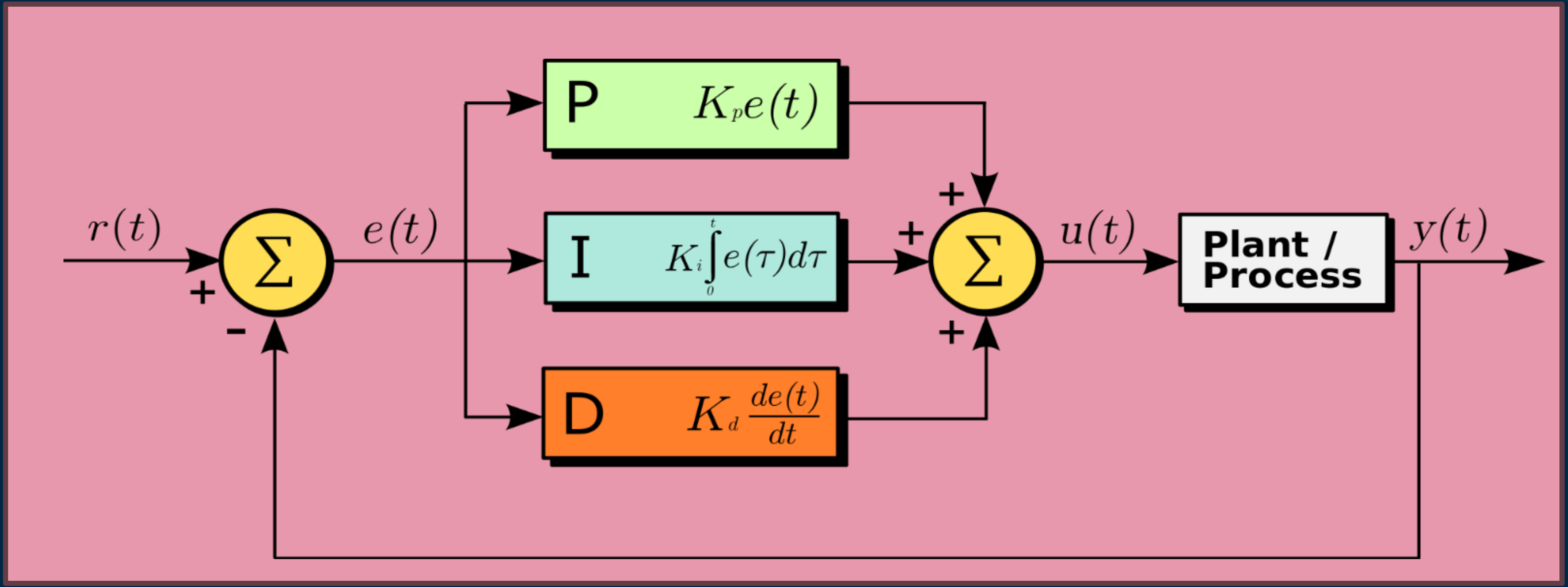


RASPBERRY PI & ARDUINO COMMUNICATION




```xml
<xacro:unless value="$(arg sim_mode)">
        <ros2_control name="RealRobot" type="system">
            <hardware>
                <plugin>diffdrive_arduino/DiffDriveArduino</plugin>
                <param name="left_wheel_name">left_wheel_joint</param>
                <param name="right_wheel_name">right_wheel_joint</param>
                <param name="loop_rate">30</param>
                <param name="device">/dev/ttyACM0</param>
                <param name="baud_rate">57600</param>
                <param name="timeout">1000</param>
                <param name="enc_counts_per_rev">1910</param>
            </hardware>
```

PID Controller

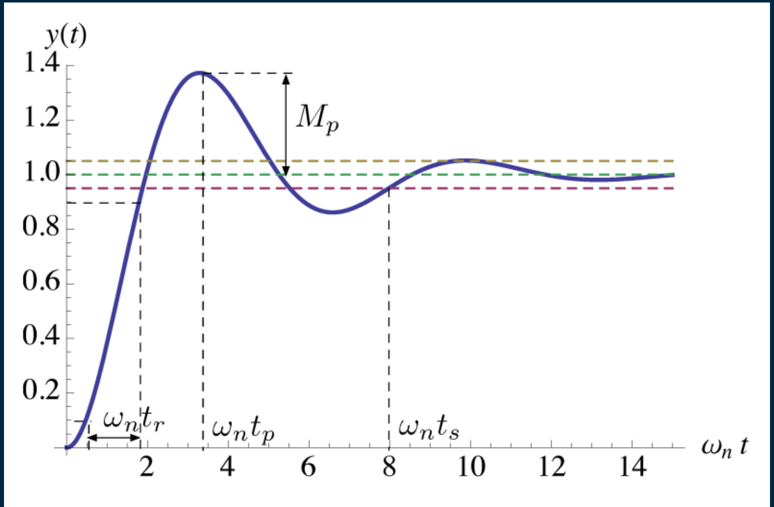


$$u(t) = K_p e(t) + K_i \int_0^t e(t')\,dt' + K_d \frac{de(t)}{dt},$$

$$K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}$$

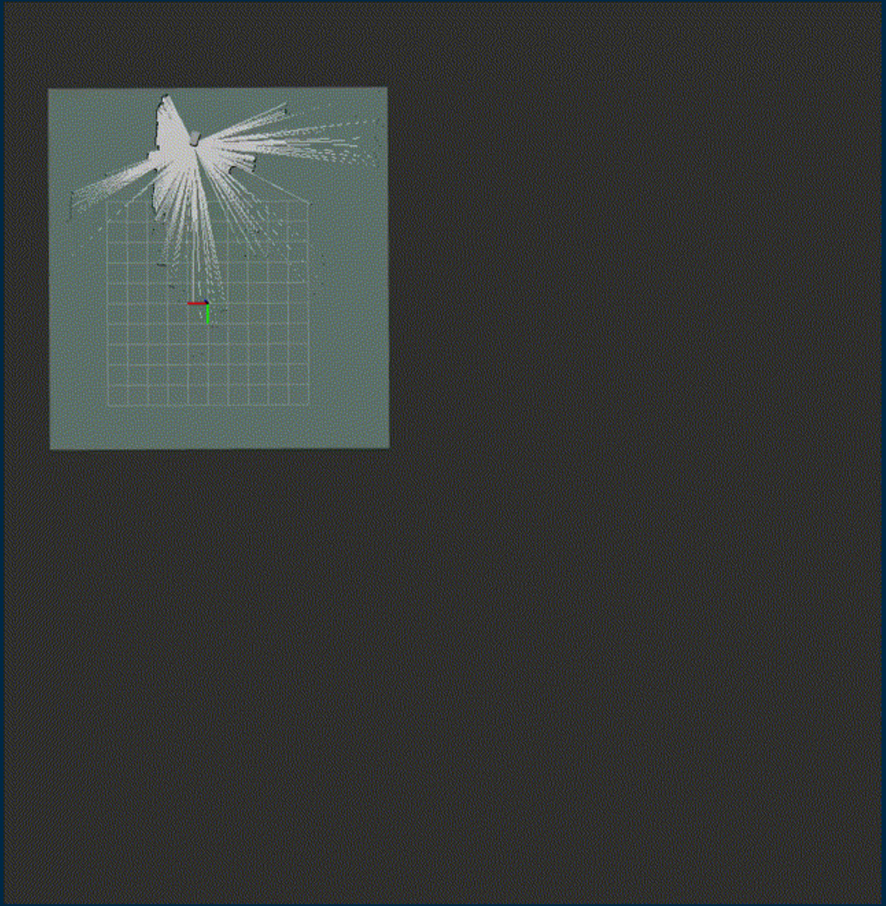
PID PARAMETERS

Controller parameter	Rise time	Overshoot	Stelling time	S – S Error
K_P	Decrease	Increase	Small change	Decrease
K_i	Decrease	Increase	Increase	Decrease
K_d	Small change	Decrease	Decrease	No change

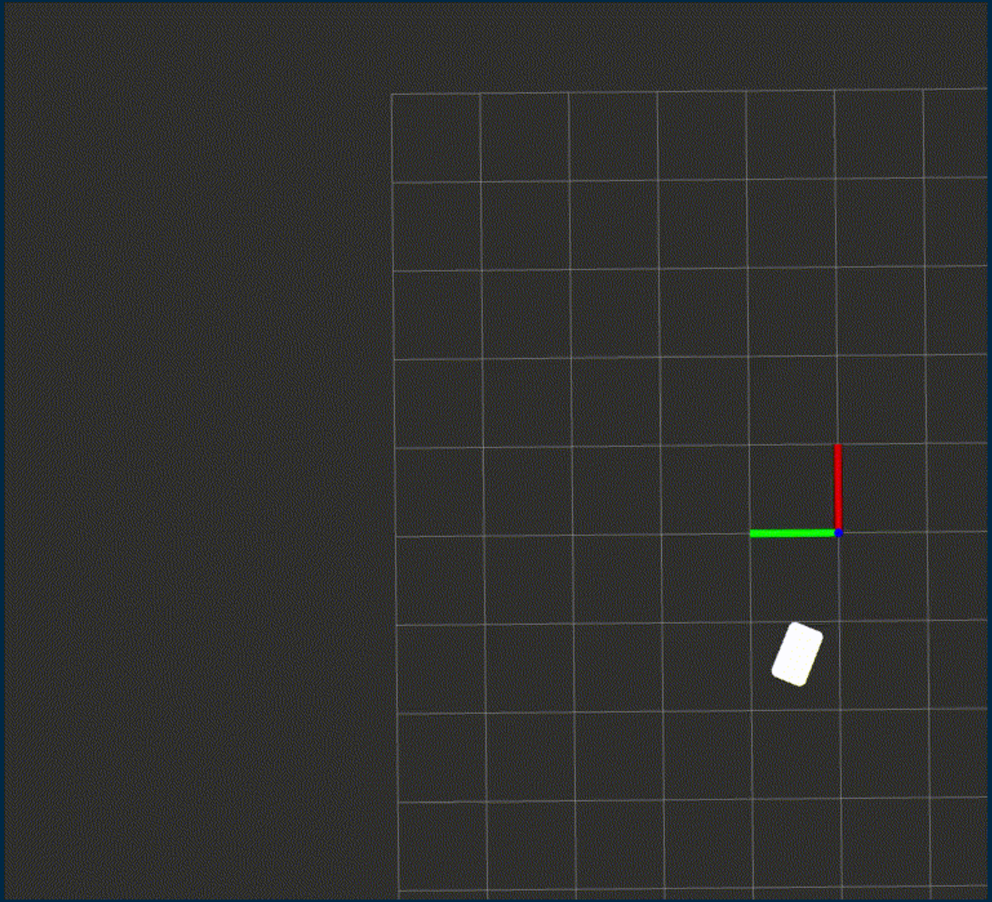


SIMULTANEOUS LOCALIZATION & MAPPING (SLAM)

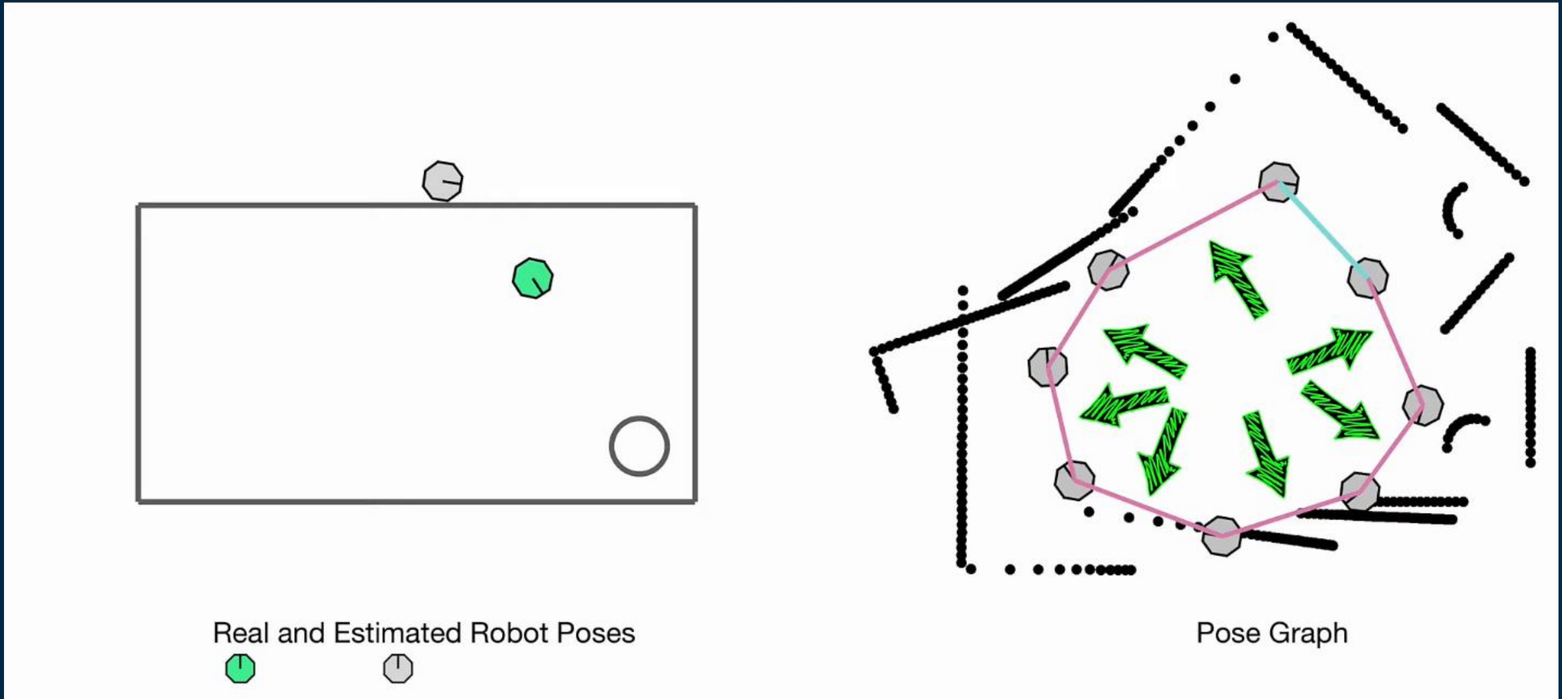
SLAM DEMONSTRATION



SLAM DEMONSTRATION



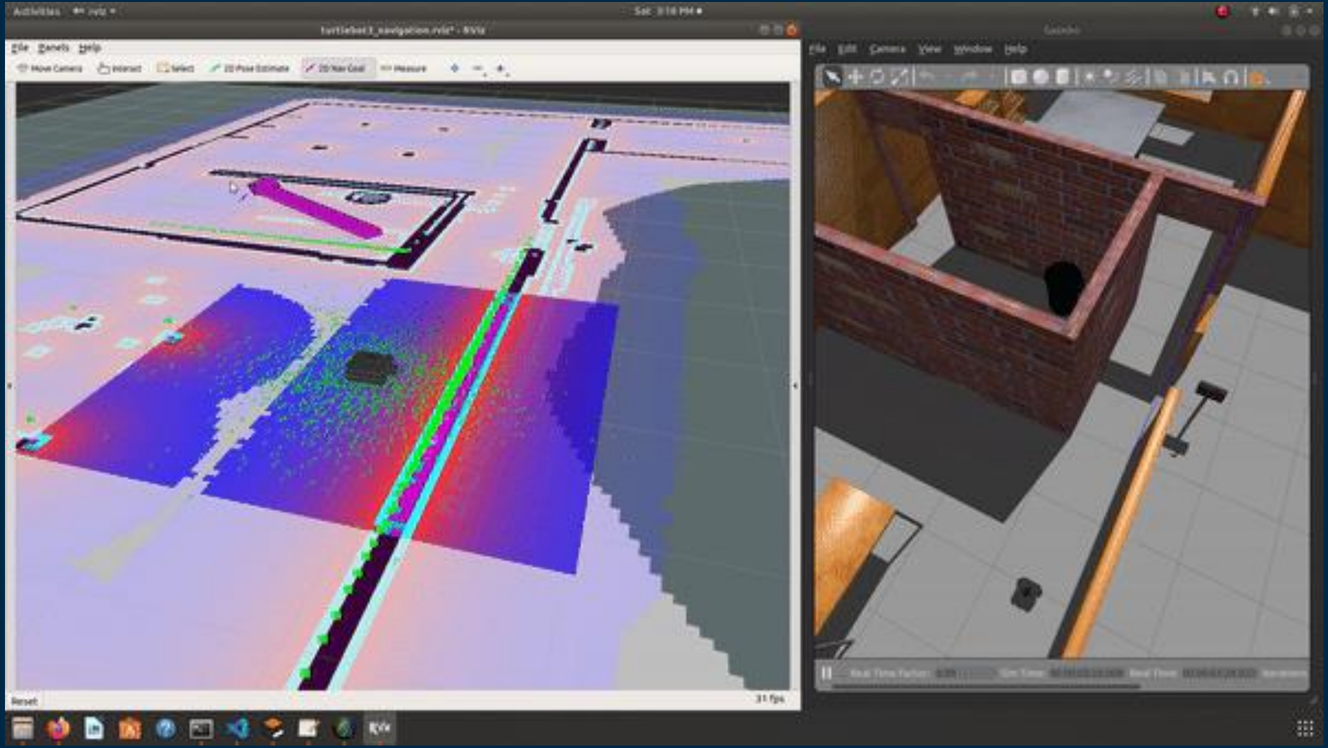
POSE GRAPH OPTIMIZATION



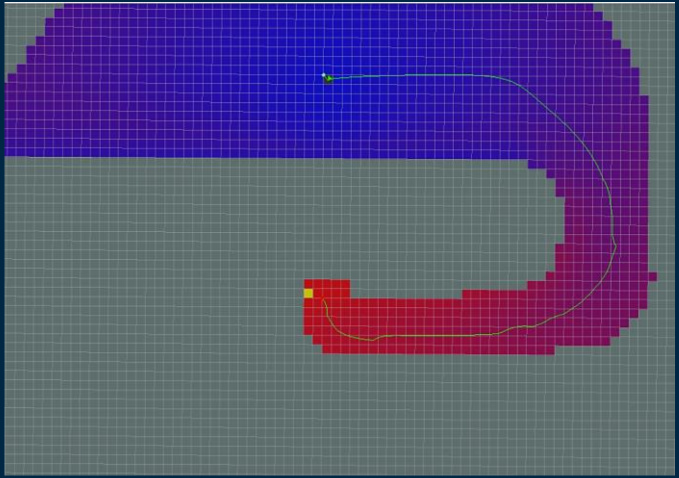
PATH PLANNING


Environment Geometry
Robot Geometry
Start Pose of Robot
Goal Pose of Robot
Path Planning Algorithm
Path from Start to Goal

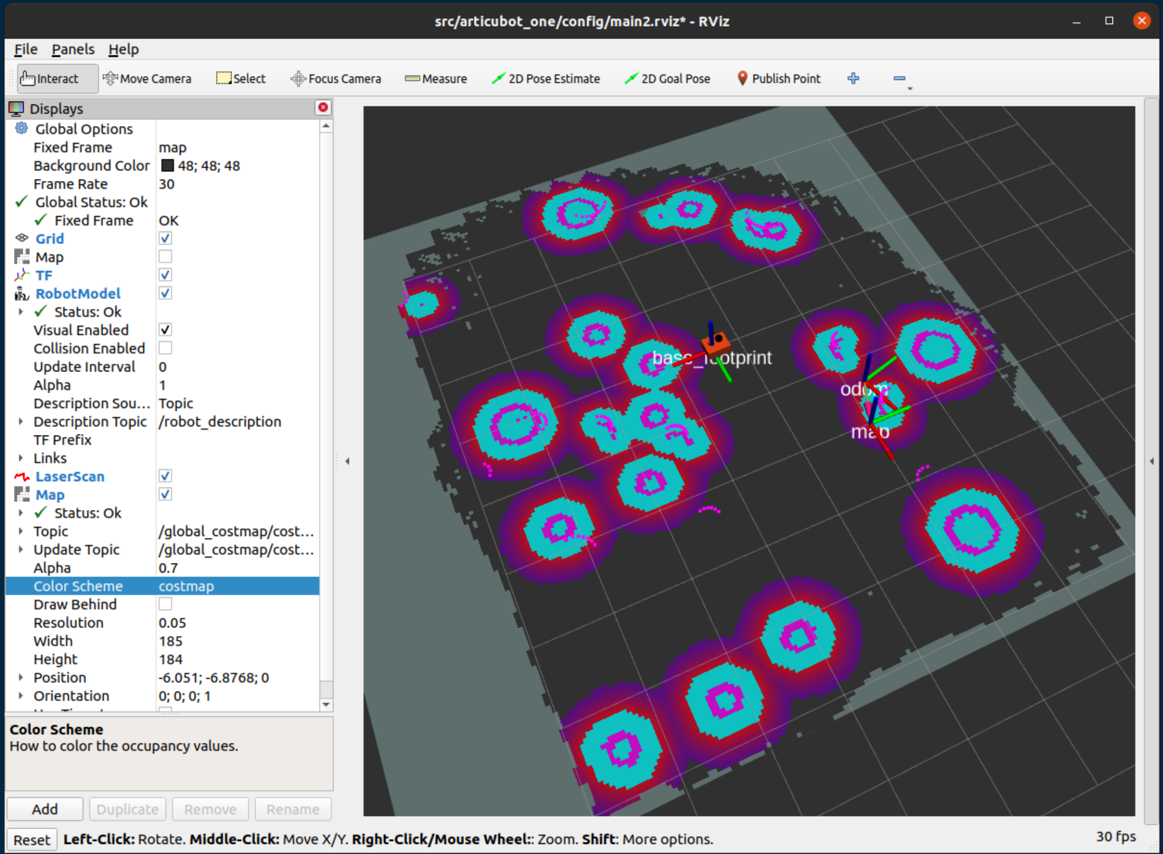
PATH PLANNING DEMONSTRATION



A* ALGORITHM

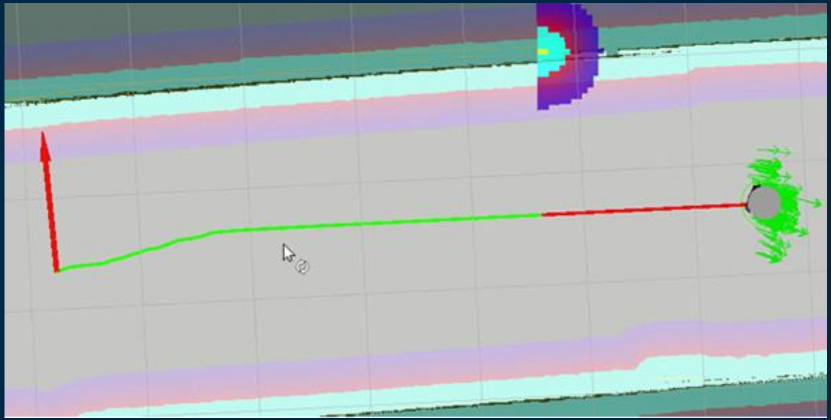


GLOBAL PLANNER

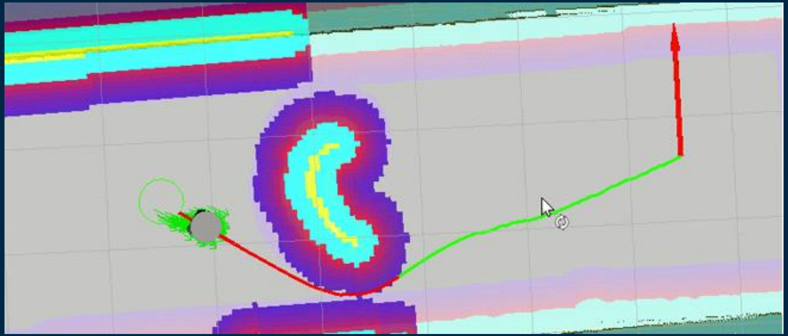


LOCAL PLANER

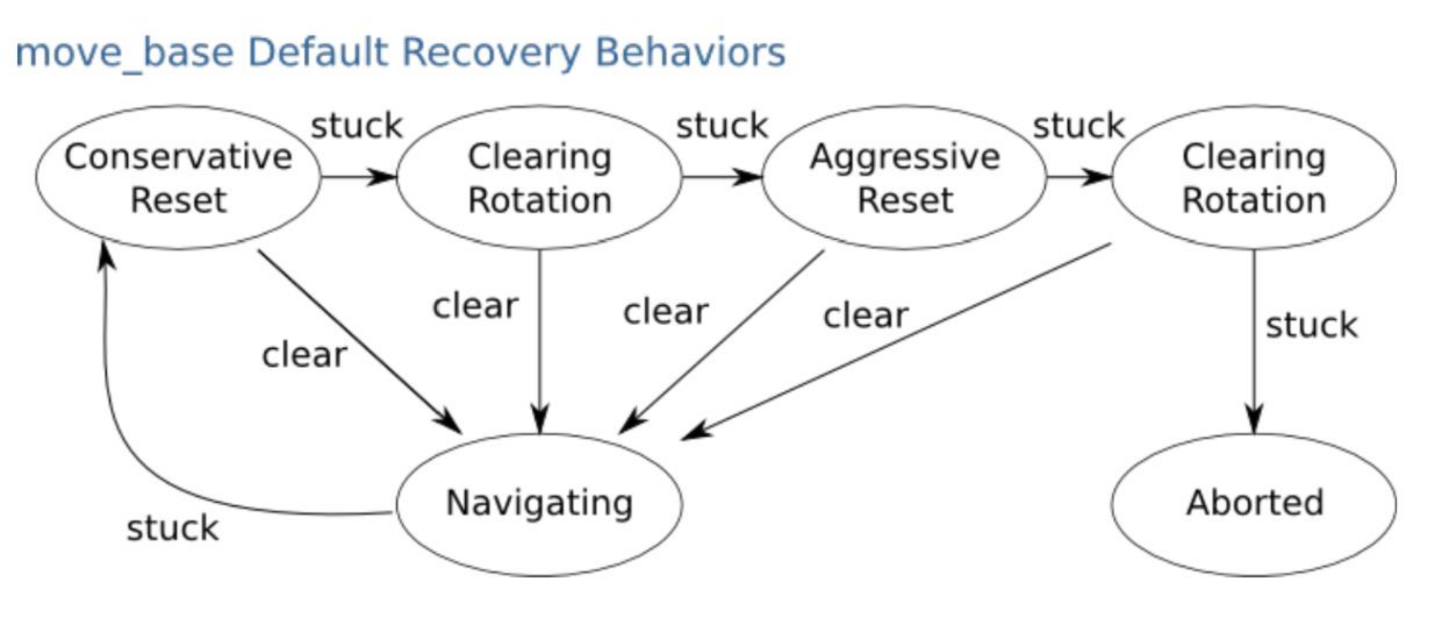
NO OBSTACLES



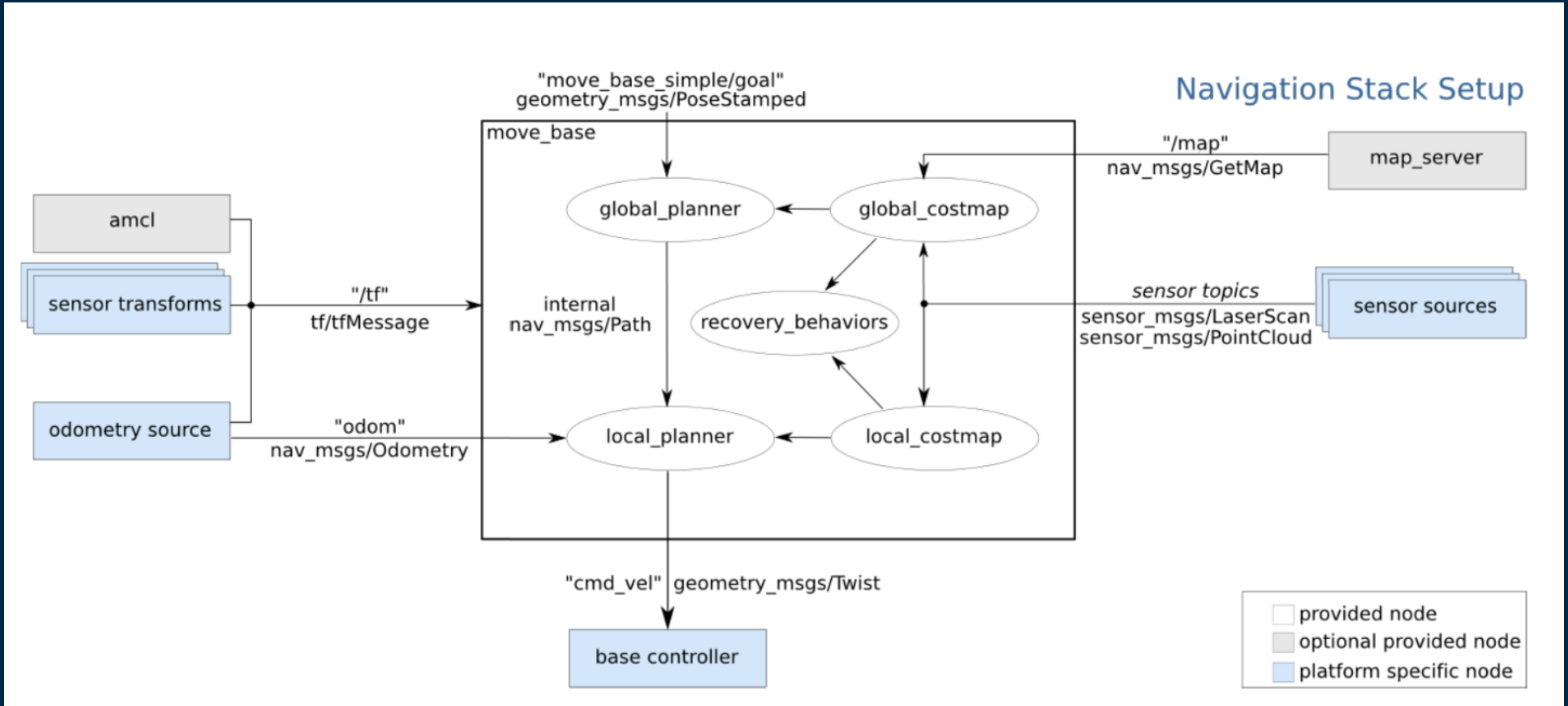
WITH OBSTACLES



RECOVERY BEHAVIOUR



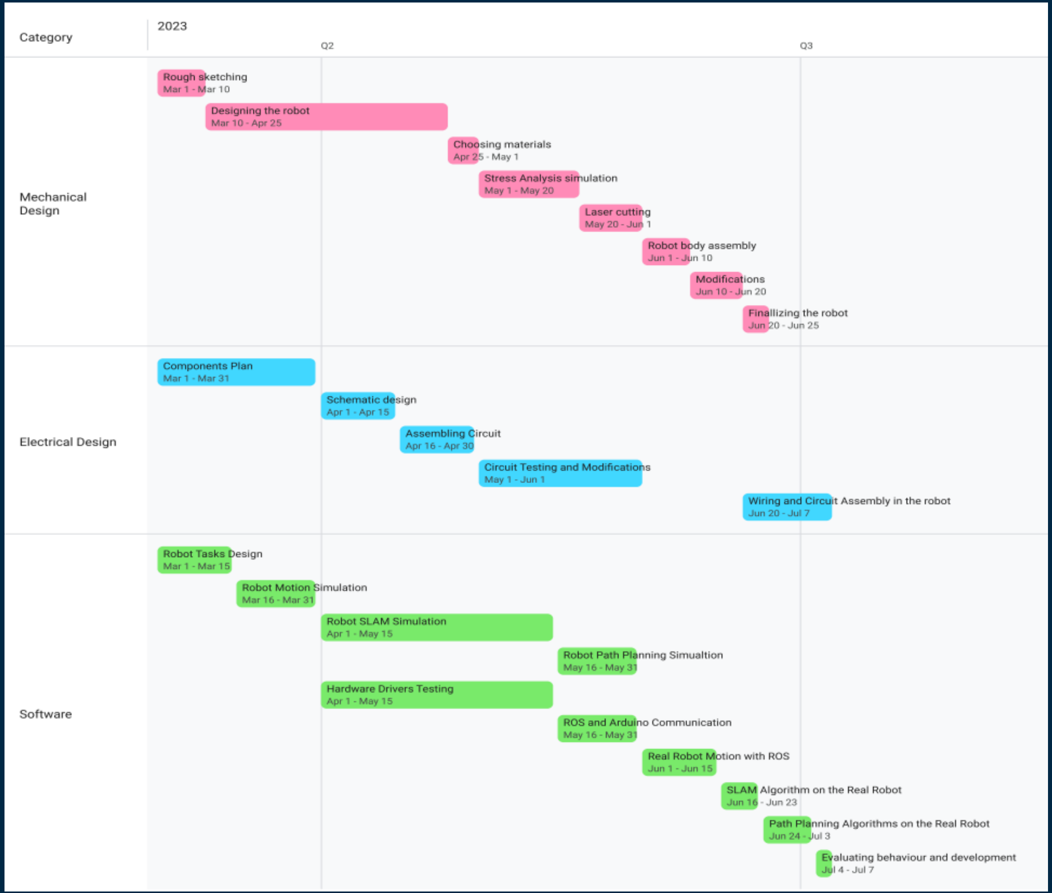
NAV2 STACK


Navigation Stack Setup
"move_base_simple/goal"
geometry_msgs/PoseStamped
move_base
amcl
sensor transforms
odometry source
"/tf"
tf/tfMessage
"odom"
nav_msgs/Odometry
global_planner
global_costmap
internal
nav_msgs/Path
recovery_behaviors
local_planner
local_costmap
"/map"
nav_msgs/GetMap
map_server
sensor topics
sensor_msgs/LaserScan
sensor_msgs/PointCloud
sensor sources
"cmd_vel"
geometry_msgs/Twist
base controller
provided node
optional provided node
platform specific node

ADDITIONS

04

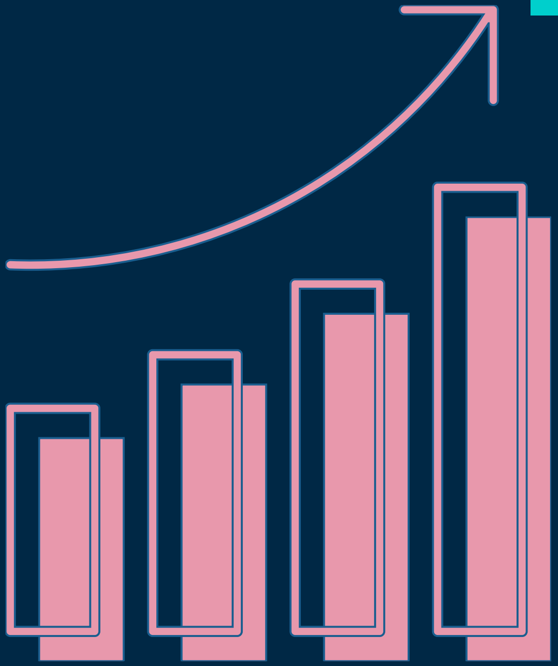
TIMELINE



Category
2023
Q2
Q3
Mechanical Design
Rough sketching
Mar 1 - Mar 10
Designing the robot
Mar 10 - Apr 25
Choosing materials
Apr 25 - May 1
Stress Analysis simulation
May 1 - May 20
Laser cutting
May 20 - Jun 1
Robot body assembly
Jun 1 - Jun 10
Modifications
Jun 10 - Jun 20
Finallizing the robot
Jun 20 - Jun 25
Electrical Design
Components Plan
Mar 1 - Mar 31
Schematic design
Apr 1 - Apr 15
Assembling Circuit
Apr 16 - Apr 30
Circuit Testing and Modifications
May 1 - Jun 1
Wiring and Circuit Assembly in the robot
Jun 20 - Jul 7
Software
Robot Tasks Design
Mar 1 - Mar 15
Robot Motion Simulation
Mar 16 - Mar 31
Robot SLAM Simulation
Apr 1 - May 15
Robot Path Planning Simualtion
May 16 - May 31
Hardware Drivers Testing
Apr 1 - May 15
ROS and Arduino Communication
May 16 - May 31
Real Robot Motion with ROS
Jun 1 - Jun 15
SLAM Algorithm on the Real Robot
Jun 16 - Jun 23
Path Planning Algorithms on the Real Robot
Jun 24 - Jul 3
Evaluating behaviour and development
Jul 4 - Jul 7

BUSINESS OPORTUNITIES

How we make Money ?



Do you have any questions?

THANK You