ABSTRACT

This paper presents a prototype of an indoor mapping robot which uses a low cost laser RPLidar,Arduino mega etc. This work will be implemented on a prototype of a mobile robot for indoor industrial applications. Initially, a mobile indoor robot incorporated with the lidar was developed .A research laboratory with spacious rooms with many objects is used for testing purposes. The collected data were analyzed using three methods of pre-processing. This includes a raw filter, a moving average of smooth filters, and finally the combination of the raw filter and the moving average. Besides, the scanning accuracy of the developed system shows success with more than 85% of correctness. As a conclusion, the laser rangefinder and the pre-processing used is capable of mapping the room space with a clear image and hence can be used for a variety of applications.

CONTENTS

CHAPTER 1

INTRODUCTION

The indoor environment mapping is one of the important aspects in robotics studies when dealing with autonomous navigation, localization, positioning, along with industrial applications. Its success depends on the accuracy of its implementation and may depend on sensors which are used to acquire the indoor environment data. As technologies become more prominent in robotics, the relationship between humans and trained robots become more vivid. Autonomous robots have the ability to replace manpower and helps in labor reduction of industries. Autonomous robots even have the ability to understand the response of humans. These trends towards robotic involvement in industry processes made companies to invest and improve productivity and customer experience, and gained a competitive advantage in every industries.Autonomous robots can be used to improve the speed and accuracy of routine operations, particularly in textile warehouse and manufacturing industries; work competency with humans for more efficiency; and reduce the risk of employee injury in dangerous environments.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE REVIEW OF SURVEY

1.ROS: "An adaptable structure for composing robot programming. It is an assortment of apparatuses, libraries, and shows that mean to improve on the errand of making complicated and vigorous robot conduct across a wide assortment of mechanical platforms." This working framework is the premise of the independent driving vehicle. It runs on the Laptop as a full work area form and on the raspberry pi as an implanted adaptation.

2.urg_node: The urg_node is the ROS driver library for the Hokuyo LiDAR sensor utilized on the vehicle to check the climate. This empowers involving the LiDAR as an ROS hub along with different instruments required for the undertaking.

3.hector_slam This is a bundle including the hector_mapping hub to make a 2D guide in view of the "great update pace of current LIDAR frameworks" produced for ROS. It is utilized to establish the 2D guide of the climate where the vehicle ought to drive and which is utilized to design the most brief course from the assigned highlight the objective of the independent vehicle.

4.hector_slam_example: It is a lot of send off documents and design records to utilize the LiDAR sensor to make a SLAM map. Utilizing these records makes it more simpler
What's more, it is quicker to make the map.

5.navigation_stack : The route stack is a library that incorporates immeasurably significant capacities to explore a ROS running robot through a 2D guide with the assistance of sensor information. "It should be designed for the shape and elements of a robot to perform at a significant level". The library contains the global_planner to view as the most brief course from a beginning stage to an objective, the move_base bundle is assigned to provide specific speed orders for the robot's engine the board in view of the course and amcl limits the vehicle on the 2D guide utilizing the LiDAR sensor data.

6. teb_local_planner "The teb_local_planner bundle executes a module to the base_local_planner of the 2D route stack."It empowers tracking down a nearby course in light of the LiDAR sensor information to keep away from impediments and make speed orders upgraded for a vehicle like robot with Ackermann steering.The teb_local_planner is liable for rethinking the first course of the vehicle, should the LiDAR sensor identify a few unknown hindrances while the vehicle is moving.

2.2. PROBLEM STATEMENT

During corona time, The small scale textile industries in kerala was facing issues of spreading covid 19 among its labors.So the mode of operations were hard to compete with the further industries. The demand was huge but the intake of laborers was less according to covid protocol. This leads to the development of an omnidirectional robot using rplidar which is capable of :

1. Moving in each and every nook and corners of the room and detecting objects using rplidar.
2. The robot was capable of carrying the clothes on its top and moving to the next destination where the next stitching of clothes is done.
3. The robot replaced human labor in so many steps and hence the labors were sufficient according to covid protocol.

CHAPTER 3
BACKGROUND

3.1 ABOUT SLAM

The term SLAM is as expressed an abbreviation for Simultaneous Localization And Planning. It was fundamentally evolved by Hugh Durrant-Whyte and John J. Leonard in view of prior work by Smith, Self and Cheeseman. DurrantWhyte and Leonard at first named it SMALL however it was subsequently different to give a better effect. Hammer is worried about the issue of building a guide of an obscure region by a versatile robot while simultaneously exploring through the climate utilizing the guide. So the robot begins anyplace in the climate what's more, has to know whenever where it at present is comparative with the perceived hindrance.

Hammer can be carried out in loads of ways, with an immense measure of equipment that can be utilized. Hammer is something beyond one calculation, it is more similar to an entirety idea to fix a couple of issue. It comprises different parts like "Milestone extraction, information affiliation, state assessment, state update and milestone update" Every of these little pieces of the idea can be settled by a ton's of ways. It relies upon the utilization, similar to what robot will be utilized. In which region will the robot be driving? Is it a corridor with just static articles or is it a metropolitan swarmed road with loads of development during the structure of the guide? Do the robot need a 2D or a 3D vision? Questions like this should be replied previously beginning to carry out the SLAM interaction. They all impact what equipment gadgets and calculations used to execute a well working SLAM idea. In the instance of this proposition, the independent driving vehicle, just a 2D movement is thought of. The region where the vehicle ought to drive is only an indoor climate with static articles.

3.2 REQUIRED HARDWARE

1. 203MM MECANUM WHEEL SET-BEARING ROLLERS ●

The 203mm Aluminum Mecanum wheels Basic (Bush sort rollers). This is a rock solid wheel. The 203mm Mecanum wheel set comprises of 1x left, 1x right. It very well may be utilized as all inclusive center points to interface the wheels with the engine drivers, utilizing rollers on the 45° rotational pivot of the fundamental wheel. As the coordination of wheels utilizing Arduino and engine driver pivot in a state of harmony, the vehicle moves advances or in reverse , yet by turning sets of wheels in inverse headings, the rollers empower sideways just as 45 degree corner developments.

Particulars:

- 203mm Aluminum mecanum wheels set i.e 1x left &1x right
- Included a progression of rollers set on 45° rotational pivot
- Width: 80mm
- Length of Roller: 31.2mm
- Body material: Aluminum
- Number of rollers : 12/per Wheel
- Number of plates: 2
- Spacer material: Aluminum
- Net weight: 1.8kg
- Load limit: 100~125Kg/Set
-

2. RHINO INDUSTRIAL BRUSHLESS DC MOTOR DRIVER (BLDC) 250W WITH RS485 MODBUS COMPATIBLE ● RMCS-3001 RS485 modbus is Rhino Motion Controls initial Brushless low voltage (15-48 V) DC drive intended for smooth force and control at higher rates. It has impeded engine yields, overvoltage and under-voltage insurance and will endure coincidental engine disengages while fueled up.

● The ENABLE, BRAKE and DIRECTION inputs are optically separated. These sources of info work with 2.5V to 5V rational drive signals. Engine will run on utilization of +ve input supply to ENABLE pins. To work with the bearing difference in the Motor shaft, a committed DIR pin is given. Subsequent to applying the +ve input supply to the DIR pin, power reset is required. BRAKE highlights accommodate slowing down the engine pivot.

- This drive has speed criticism (Hz) and current input (A). This drive has the choice to control the length of movement, when the length of that movement is surpassed the engine will consequently enter brake condition. There is a capacity in the drive for setting the Modbus Slave Address from 1 to 247 utilizing MODBUS Tool Device (Software Setting).
- Details:
- Input Supply Voltage: 15 to 48V
- Evaluated current : 5A
- Top Current: 20A
- Smooth and calm activity at all rates and amazingly low

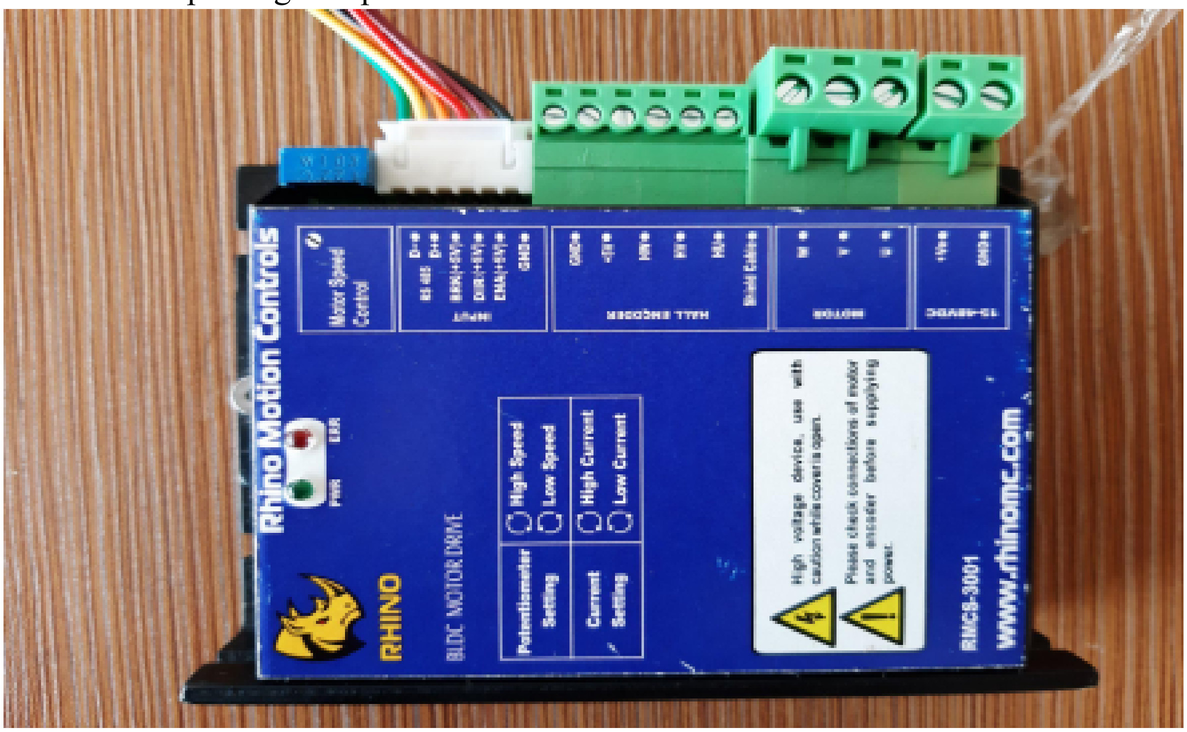
engine warming ● Completely configurable position and speed circle

● Conveys up to 5A of persistent current

● 2.5V, 3.3V and 5V viable BRAKE, DIRECTION and ENABLE contributions with 2-wire opto-disengaged interface

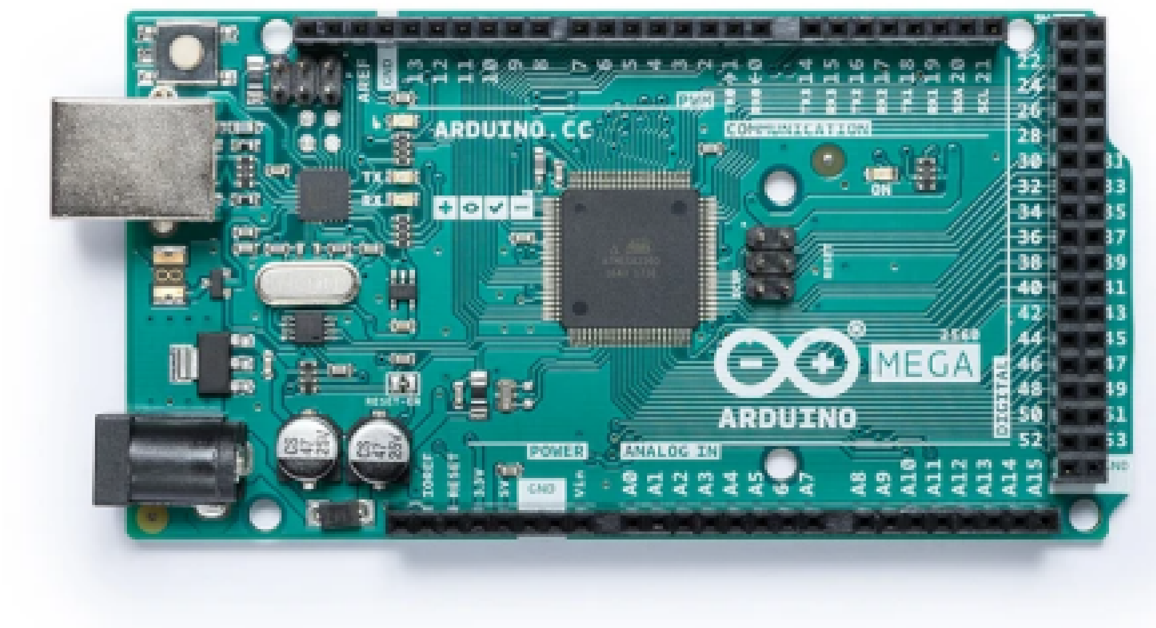
● Cut off for the engine yields, overvoltage and under-voltage assurance ● Driven sign for power and mistake states

● Encompassing Temperature : 0-70°C



3. ARDUINO MEGA2560

The Arduino MEGA 2560 is intended for projects that require more I/O lines, more sketch memory and more RAM. With 54 advanced I/O pins, 16 simple information sources and a bigger space for your sketch it is the suggested load up for 3D printers and mechanical technology projects.The Arduino Mega 2560 is customized utilizing the Arduino Software (IDE), our Integrated Development Environment normal to every one of our loads up and running both on the web and offline.If you need to program your Arduino Mega 2560 while disconnected you want to introduce the Arduino Desktop IDE.Connect your Mega2560 load up with A B USB link; at times this link is known as a USB printer link.



4. NEMA23 PLANETARY GEARED BLDC MOTOR 35KGCM 230 RPM 100 W

The Rhino Industrial Planetary Geared (IPG) 100W BLDC engine is a conservative workhorse for the modern transport lines, food handling hardware, versatile modern robots and other consistent use applications in the business. The Rhino IPG 100W BLDC engine RMCS-2070 is accessible in the Nema23 outline size alongside an all metal incorporated planetary gearbox with a 1:13 stuff proportion. This gives a usable appraised force of 35Kgcm at 230RPM. This force at 230 RPM in the conservative Nema23 outline size is an industry first special contribution from Rhino Motion Controls. This has been made conceivable in view of the High quality engines joined with the High Performance Brushless DC engine driver RMCS - 3001 which offers a superior presentation shut circle modern answer for these engines working at 24V.

The Rhino IPG 100W Brushless DC engine and drive are evaluated to work at 24 volts which is the appraised voltage for some modern applications. Additionally the dependability and long haul inconvenience free execution is ensured by the high form quality and tough guidelines for QC assessment set up at Rhino Motion Controls.

The Rhino IPG 100W brushless DC engines are shut circle engines gave low commotion corridor sensor criticism. This sensor input empowers an exact speed control of the engines alongside keeping up with the high force prerequisites.

Engine Specifications with Planetary Gearbox:

Evaluated Torque : 35 Kgcm

Evaluated speed : 230 Rpm

Gear proportion : 1 : 13

Engine length without Shaft and gearbox : 87mm

Gear box length(L) : 49.5mm

Shaft Length: 28mm

Shaft Dia: 12mm

Proficiency: 0.81%

Input Voltage : 24VDC

Input Current : up till 10 A(power supply of 240 W, 24V, 10 An ought to be utilized for ideal outcomes)

Yield Power : 100 W

Weight: 2Kg

Base Motor Specifications:

Number of posts : 4

Stage to Phase opposition : 0.44 ohms

No heap speed : 3800 Rpm

No heap current : 1 Amps Max.

Appraised force : 0.33 N-m

Appraised speed : 3000 Rpm

Protection class : B

Engine Wiring Connections:

The engine has 8 wires which should be associated with the drive according to beneath : Engine Phase U (Yellow shading 20 AWG wire)

Engine Phase V (Green shading 20 AWG wire)

Engine Phase W (Blue shading 20 AWG wire)

Encoder Phase U (Yellow shading 26 AWG wire)

Encoder Phase V (Green shading 26 AWG wire)

Encoder Phase W (Blue shading 26 AWG wire)

Encoder VCC (+5 V DC Red shading wire)

Encoder Ground (Ground Black shading wire)

5. **TETRIX WIRELESS JOYSTICK GAMEPAD SYSTEM MODEL40377**

- Its a reasonable 2.4 GHz, four-direct regulator. It is made in the gamepad style with double joysticks. It offers very good quality advantages, including obstruction free execution, servo turning around, engine trim change, simple connecting, and a R2004GF collector.

- Details
- Weight: 378 g for each pack
- Note: This gamepad is exclusively for the 42084 TETRIX R/C Receiver.

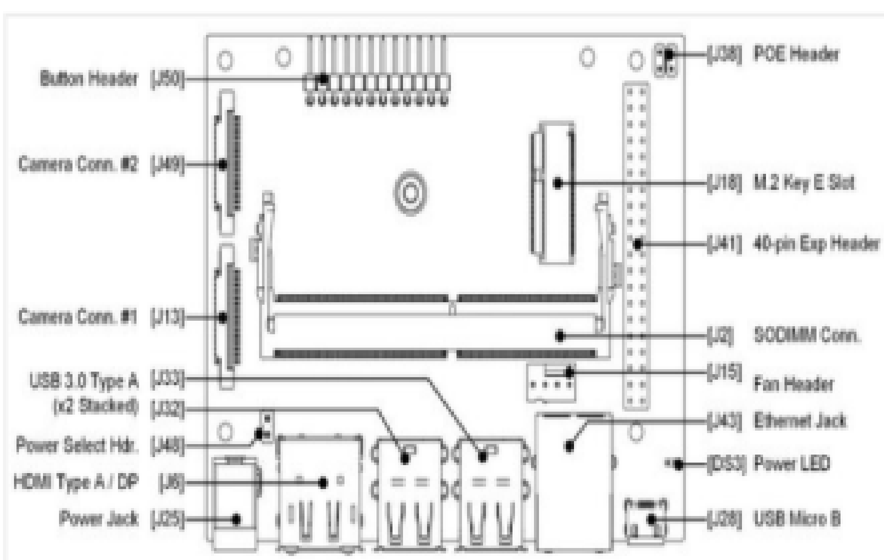
6. R2004GF (FHSS) RECEIVER

For use with the Futaba 3PL and 4YF 2.4GHz FHSS transmitters. Adaptable coaxial radio wire makes it simple to position inside a model for ideal gathering. Forestalls impedance and advances constancy with recurrence jumping and a wide spread-range innovation.

7. JETSON NANO

NVIDIA Jetson Nano allows you to carry amazing new capacities to a great many little, power-effective AI frameworks. It opens new universes of inserted IoT applications, including passage level Network Video Recorders (NVRs), home robots, and shrewd doors with full investigation capacities.

Jetson Nano is additionally the ideal device to begin finding out with regards to AI and advanced mechanics in true settings, with prepared to-attempt projects and the help of a functioning and enthusiastic engineer local area.



8. RPLIDAR

RPLIDAR A1 is a minimal expense 360 degree 2D laser scanner (LIDAR) arrangement created by SLAMTEC. The framework can perform 360 degree examine inside 6 meter range. The created 2D point cloud

information can be utilized in planning, confinement and item/climate displaying. RPLIDAR A1's filtering recurrence arrived at 5.5 hz when inspecting 360 focuses each round. Furthermore it very well may be arranged up to 10 hz
most extreme. RPLIDAR A1 is fundamentally a laser triangulation estimation framework. It can work fantastic in a wide range of indoor climate and outside climate without daylight. RPLIDAR A1 contains a reach scanner framework and an engine framework. After power on each sub-framework, RPLIDAR A1 begin turning and checking clockwise. Client can get range check information through the correspondence interface (Serial port/USB).RPLIDAR A1 accompanies a speed discovery and versatile framework. The framework will change recurrence of laser scanner naturally as per engine speed. Furthermore have framework can get RPLIDAR A1's genuine speed through correspondence interface. The straightforward power supply pattern saves LIDAR framework's BOM cost and makes RPLIDAR A1 a lot simpler to utilize. Point by point particular with regards to power and correspondence point of interaction can be found in the accompanying areas.

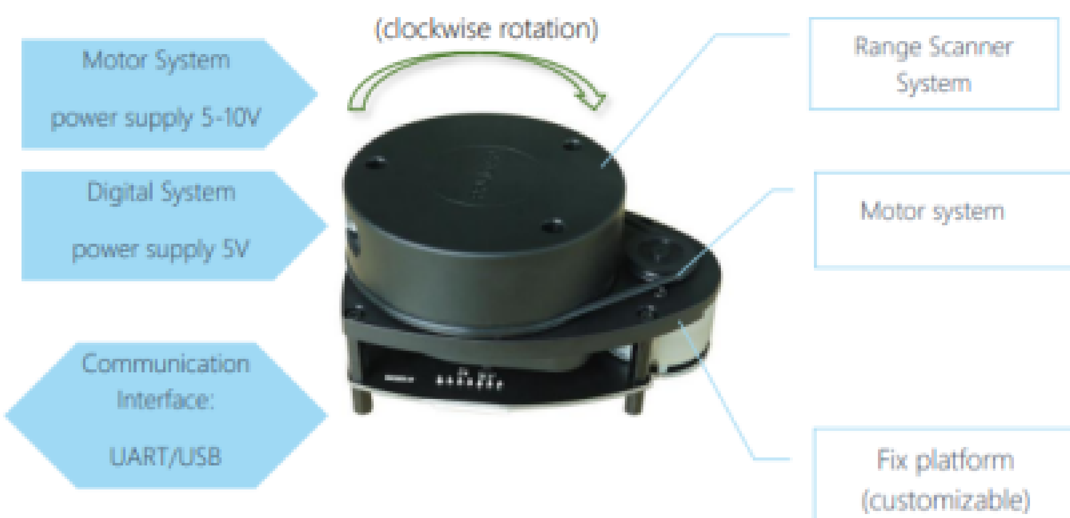


FIG:RPLIDAR A1 SYSTEM COMPOSITION

Mechanism :
RPLIDAR depends on laser triangulation running rule and uses high velocity vision obtaining and handling equipment created by SLAMTEC. The framework estimates distance information in excess of 2000 times' each second and with high

goal distance yield (<1% of the distance). It regulates infrared laser signal and the laser signal is then to be reflected by the item to be recognized back. The returning sign is inspected by vision procurement framework in RPLIDAR A1 and the DSP implanted in RPLIDAR A1 begin handling the example information and result distance worth and point esteem among item and RPLIDAR A1 through correspondence interface.The fast going scanner framework is mounted on a turning rotator with a form in rakish encoding framework. During turning, a 360 degree output of the current climate will be performed.

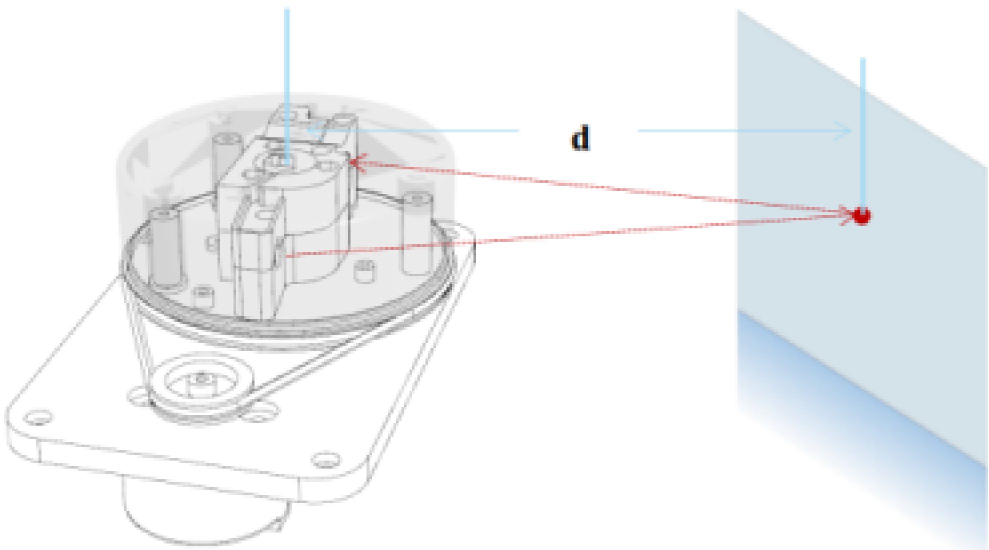


FIG: WORKING SCHEMATIC.

Security and Scope:

RPLIDAR A1 framework utilize a very low power (<5mW) infrared laser as its main light source. It drives utilizing tweaked beat. The laser emanates in an extremely brief period of time which can ensure its wellbeing to human and pet and arrive at Class I laser security standard. The tweaked laser can adequately forestall encompassing light and daylight during going examining process. This make RPLIDAR A1 work fantastic in a wide range of indoor climate and outside climate without daylight.

Information Output :

At the point when RPLIDAR A1 is working, inspecting information will result to correspondence interface. Each example point contains beneath data. RPLIDAR A1 yields examining information persistently. Have frameworks can design yield organization and stop RPLIDAR A1 by sending stop order. On the off chance that you really want itemized information arrangement and correspondence convention, if it's not too much trouble, contact with SLAMTEC.

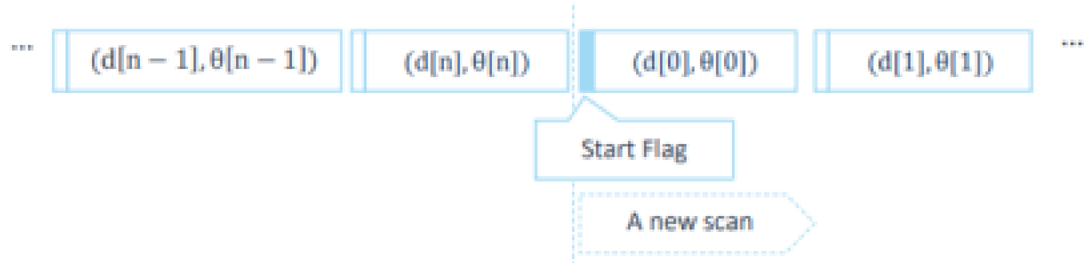


FIG: SAMPLE PONTS DATA FRAMES

CHAPTER 4

HARDWARE/SOFTWARE DESIGN AND ITS DEVELOPMENT

1. ARDUINO IDE:

The Arduino Integrated Development Environment is a cross-stage application that is written in capacities from C and C++. It is utilized to compose and transfer projects to Arduino viable sheets, yet additionally, with the assistance of outsider centers, other seller improvement sheets.

2. JETSON NANO DEVELOPER KIT:

NVIDIA Jetson designer packs are utilized by experts to create and test programming for items in view of Jetson modules, and by understudies and devotees for undertakings and learning. Every engineer unit incorporates a non-creation determination Jetson module connected to a reference transporter board with standard equipment interfaces for adaptable turn of events and fast prototyping.

3. ROS PACKAGE (MELODIC):

Bundles are constructed and facilitated on framework kept up with and paid for by the Open Source Robotics Foundation.

Step1: configure UBUNTU in jetson Nano.
Step2:Setup your source list using code :- sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
Step3:Set up your keys using below:-
sudo apt install curl

curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -

Step4:Installation and Update
sudo apt update

Ros package Installation : ROS, rqt, rviz, robot-generic libraries, 2D/3D simulators and 2D/3D perception.
sudo apt install ros-melodic-desktop-full

To find available packages from whole package , use: apt search ros-melodic Step 5: Environment setup is done

accordingly.

```
echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

Step 6: Dependencies for building packages done according to needs.

```
sudo apt install python-rosdep python-rosinstall
python-rosinstall-generator python-wstool build-essential
```

```
sudo apt install python-rosdep
```

Step 7: Final setup.

```
sudo rosdep init
rosdep update
```

4. Rviz

Rviz is a 3d perception device for ROS applications. It gives a perspective on your robot model, catch sensor data from robot sensors, and replay caught information. It can show information from camera, lasers, from 3D and 2D gadgets including pictures and point mists.

4.1 THEORY

There are multiple ways of empowering robots to continue on a strong surface, like utilizing haggles. Since wheels give speed and simplicity of control, they are normally the best option for robots in the average size, little size, and junior robot soccer leagues.The omni-directional wheel is a sort of wheel that can openly move in more than one bearing. It's famous for versatile robots since it doesn't have to turn first for moving starting with one point then onto the next in a straight way. In addition, moving along a way can be joined with turn, so the robot can show up to its objective at the right point.

Omni-wheels have more modest rollers on the edges that move totally opposite to the actual wheel. With this sort of wheel, they should be mounted opposite to the focal point of the robot. A well known kind of omni-wheel is the Mecanum, which is exceptional in that the little rollers are at a 45 degree point. This permits them to be mounted like typical wheels yet give a similar style of development as omni-wheels.In use, a complete power vector in any ideal bearing is created and can move the stage toward any path without changing the wheel's direction.Omni-directional wheels can be considered as would be expected wheels with the capacity to roll or slip sideways. Consider them controlled casters. The overall rule of an omni-directional wheel is that while the fundamental wheel gives foothold toward the path ordinary to the engine pivot, it can slide frictionless in the engine hub course in light of the little rollers.

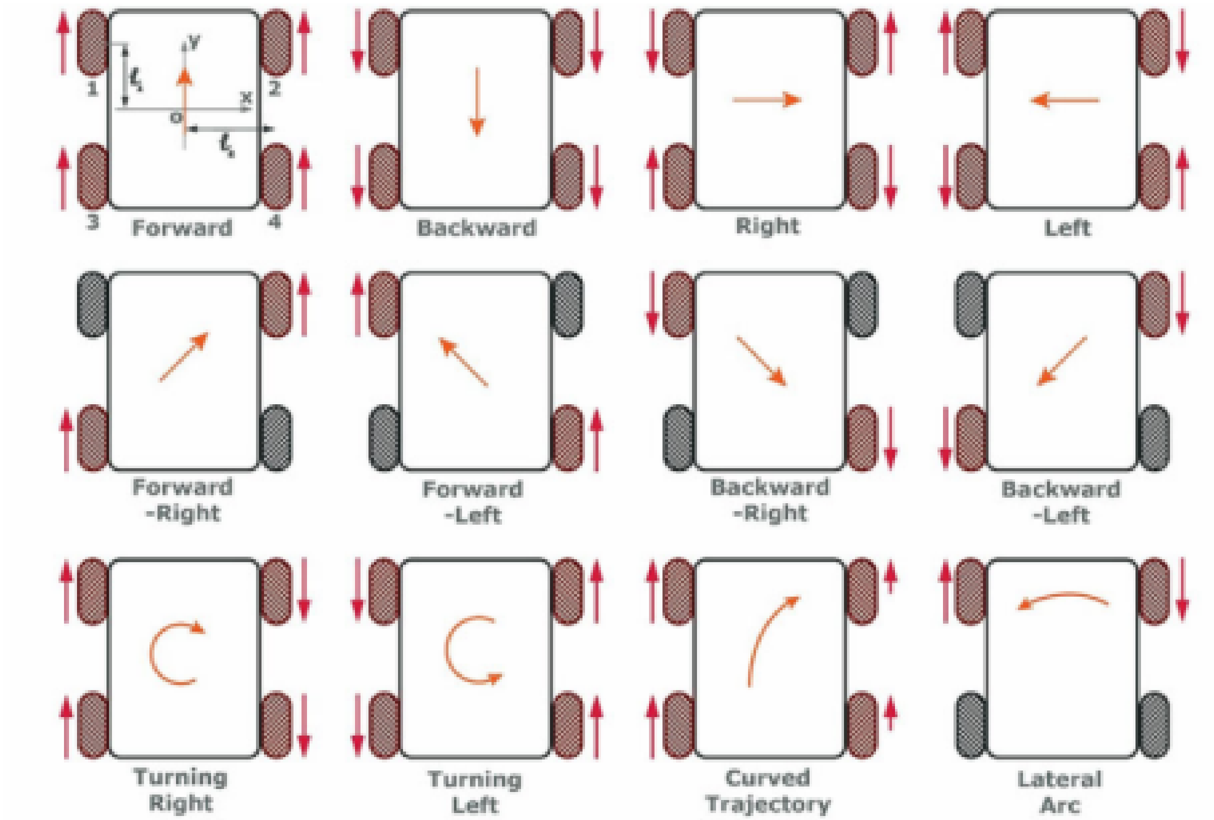


FIG: Omnidirectional wheel movement

The project deals with industrial purposes mainly in textile industries,or any small scale industries So the movement of the robot should be to every nook and corners.In Short omnidirectional wheel is perfect.The movement of wheels is as follows:

- For forward movement , the left side wheel of indoor robot should be anticlockwise and the right side should move clockwise.
- For backward movement the vice versa.
- To turn 180 degrees left, the left side of the Indoor robot should move outward and the right side of the robot should move inward.
- To turn 180 degrees right, the indoor robot has to move vice versa.
- To move 45 degrees forward right, the forward right wheel and backward left wheel has to move forward.
- To move 45 degrees backward left , the forward right wheel and backward

left wheel has to move backward.

- To move 45 degrees forward left, the forward left wheel and backward right wheel of indoor robot has to move forward.
- To move 45 degrees backward right, the forward left wheel and backward right wheel has to move backward

CHAPTER 5.

EXPERIMENTAL SETUP, RESULTS AND DISCUSSIONS

The whole module consists of Basic robot structure such as the Arduino board,battery,receiver,motor. Finally after coding part the whole module is tested using the remote.The omnidirectional movement was tested initially.Then the integrated part is tested for omnidirectional movement. The final output is working according to the

industrial purpose.Its verified and checked at different times for the final run.

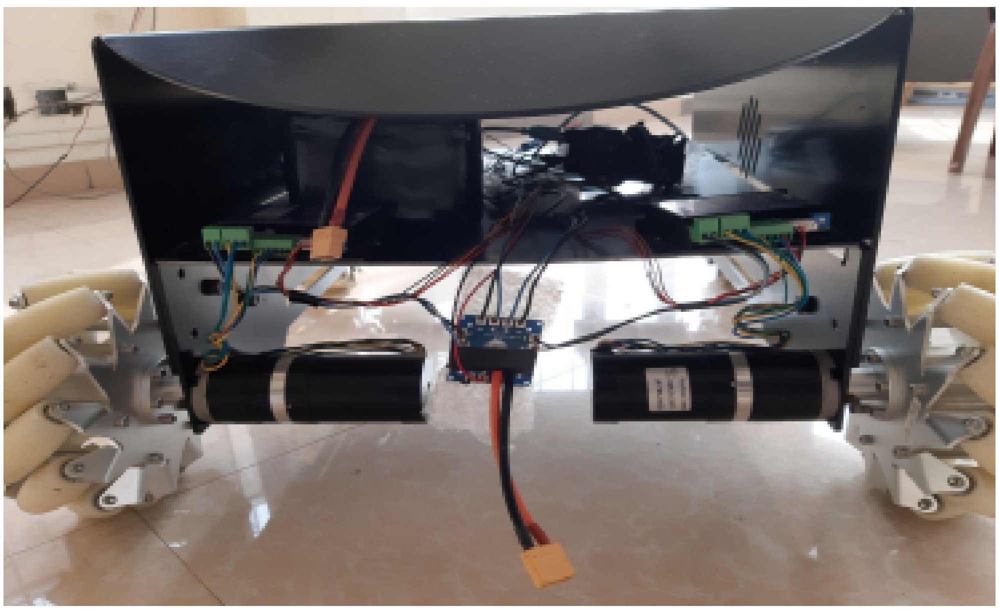


Fig: module

TESTING LIDAR OUTPUTS

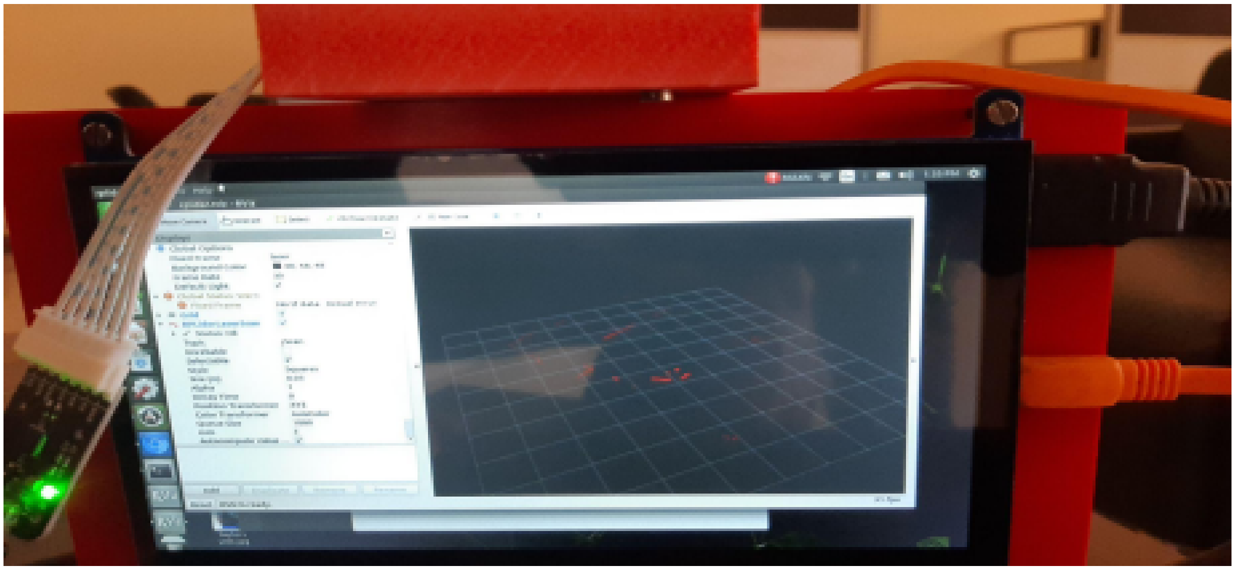
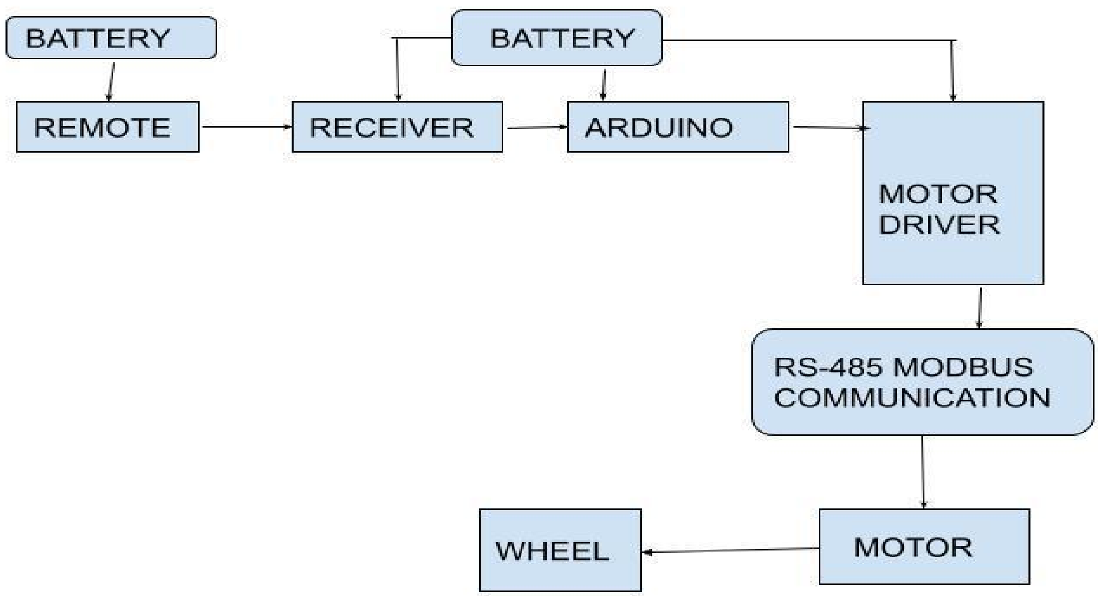


Fig: Rviz output

BLOCK DIAGRAM



CHAPTER 6.

ADVANTAGES AND APPLICATIONS

1. Indoor delivery robots.
2. Mapping of underground mines.
3. Robot vacuum cleaners.
4. Industries.

CHAPTER 7.

CONCLUSION AND SCOPE FOR FUTURE WORK

The work is based on an autonomous mobile robot for indoor environment applications.The environment mapping in one of the necessary aspects in mobile robotics studies when dealing with localization, positioning, autonomous navigation, as well as search and rescue. Its success depends on the accuracy and reliability of its implementation and may depend on sensors which are used to acquire the environment data. we tested Simultaneous localization and mapping about mobile robots in indoor environment, where all experiments were conducted based on the Robot Operating System (ROS).

REFERENCES

1. https://www.youtube.com/
2. www.google.com
3. https://www.hackster.io/361126/autonomous-uv-robot-with-slam-2 15203
4. http://wiki.ros.org/ROS/Tutorials
5. https://www.nvidia.com/en-us/training/

APPENDIX-1

Arduino code

```
int BLE = 2;

int BLD = 3;

int BLB = 4;

int FLE = 5;

int FLD = 6;
int FLB = 7;



int FRB = 8;

int FRD = 9;

int FRE = 10;

int BRB = 11;

int BRD = 13;

int BRE = 12;



int lr = 17;

int fb = 18;
```

```
int rot = 20;

int dig = 19;

long FB, LR, ROT, DIG;



void setup() {

   pinMode(FRB, OUTPUT);
  pinMode(FRD,

  OUTPUT);

  pinMode(FRE,

  OUTPUT);

  pinMode(BRB,

  OUTPUT);

  pinMode(BRD,

  OUTPUT);

  pinMode(BRE,

  OUTPUT);

  pinMode(FLB,

  OUTPUT);
```

```
pinMode(FLD,

OUTPUT);

pinMode(FLE,

OUTPUT);

pinMode(BLB,

OUTPUT);

pinMode(BLD,

OUTPUT);

pinMode(BLE,

OUTPUT);

pinMode(fb,

INPUT);

pinMode(lr,

INPUT);

pinMode(rot,

INPUT);

pinMode(dig,
```

```
  INPUT);
  digitalWrite(FRB,
  LOW);
  digitalWrite(FLB,
  LOW);
  digitalWrite(BRB,
  LOW);
  digitalWrite(BLB,
  LOW);
  Serial.begin(9600
  );

}

void loop() {
```

```
FB = pulseIn(fb,

HIGH); LR =

pulseIn(lr, HIGH);

ROT = pulseIn(rot,

HIGH); DIG =

pulseIn(dig,

HIGH);



Serial.print(" FB=");

Serial.print(FB);

Serial.print(" LR=");
Serial.print(LR);

Serial.print(" ROT=");

Serial.print(ROT);

Serial.print(" DIG=");

Serial.print(DIG);

if (FB < 850 && LR < 850)

{
```

```
  delay(500);

  digitalWrite(FRE, LOW);

  digitalWrite(FLE, LOW);

  digitalWrite(BRE, LOW);

  digitalWrite(BLE, LOW);

  Serial.println(" STOP");

}

else if (FB < 1350 && LR > 1350 && LR

< 1650) {

  delay(500);
  digitalWrite(FRD, LOW);

  digitalWrite(FLD, HIGH);

  digitalWrite(BRD, LOW);

  digitalWrite(BLD, HIGH);



  digitalWrite(FRE, HIGH);

  digitalWrite(FLE, HIGH);

  digitalWrite(BRE, HIGH);
```

```
  digitalWrite(BLE, HIGH);

  Serial.println(" FORWARD");

}

else if (FB > 1650 && LR > 1350 && LR

< 1650) {

  delay(500);

  digitalWrite(FRD, HIGH);

  digitalWrite(FLD, LOW);

  digitalWrite(BRD, HIGH);
  digitalWrite(BLD, LOW);



  digitalWrite(FRE, HIGH);

  digitalWrite(FLE, HIGH);

  digitalWrite(BRE, HIGH);

  digitalWrite(BLE, HIGH);

  Serial.println(" BACKWARD");

}

else if (LR < 1350 && FB > 1350 && FB
```

```
< 1650) {

  delay(500);

  digitalWrite(FRD, LOW);

  digitalWrite(FLD, LOW);

  digitalWrite(BRD, HIGH);

  digitalWrite(BLD, HIGH);



  digitalWrite(FRE, HIGH);
  digitalWrite(FLE, HIGH);

  digitalWrite(BRE, HIGH);

  digitalWrite(BLE, HIGH);

  Serial.println(" LEFT");

}

else if (LR > 1650 && FB > 1350 && FB

< 1650) {

  delay(500);

  digitalWrite(FRD, HIGH);
```

```
  digitalWrite(FLD, HIGH);

  digitalWrite(BRD, LOW);

  digitalWrite(BLD, LOW);



  digitalWrite(FRE, HIGH);

  digitalWrite(FLE, HIGH);

  digitalWrite(BRE, HIGH);

  digitalWrite(BLE, HIGH);
  Serial.println(" RIGHT");

}

else if (FB < 1350 && LR

> 1650) {

  delay(500);

  digitalWrite(FRD,

  HIGH);

  digitalWrite(FLD,

  HIGH);

  digitalWrite(BRD,
```

```
  LOW);

  digitalWrite(BLD,

  LOW);



  digitalWrite(FRE, LOW);

  digitalWrite(FLE,

  HIGH);

  digitalWrite(BRE,

  HIGH);

  digitalWrite(BLE,

  LOW);

  Serial.println(" 45 F

RIGHT"); }

else if (FB > 1650 && LR < 1350)
{

  delay(500);

  digitalWrite(FRD, LOW);

  digitalWrite(FLD, LOW);
```

```
  digitalWrite(BRD,
  HIGH);
  digitalWrite(BLD,
  HIGH);


  digitalWrite(FRE, LOW);
  digitalWrite(FLE,
  HIGH);
  digitalWrite(BRE,
  HIGH);
  digitalWrite(BLE,
  LOW);
  Serial.println(" 45 B
LEFT"); }
else if (FB > 1650 && LR
> 1650) {
  delay(500);
```

```
  digitalWrite(FRD, HIGH);
  digitalWrite(FLD,

  HIGH);

  digitalWrite(BRD,

  LOW);

  digitalWrite(BLD,

  LOW);



  digitalWrite(FRE,

  HIGH);

  digitalWrite(FLE,

  LOW);

  digitalWrite(BRE, LOW);

  digitalWrite(BLE,

HIGH);

Serial.println(" 45 F

LEFT"); }

else if (FB < 1350 && LR
```

```
< 1350) {

 delay(500);

 digitalWrite(FRD, LOW);

 digitalWrite(FLD, LOW);

 digitalWrite(BRD,

 HIGH);

 digitalWrite(BLD,

 HIGH);

 digitalWrite(FRE,

 HIGH);

 digitalWrite(FLE,

 LOW);

 digitalWrite(BRE,

 LOW);

 digitalWrite(BLE,

 HIGH);

 Serial.println(" 45 B
```

```
RIGHT"); }

else if (ROT < 1350)

{

  delay(500);

  digitalWrite(FRD,

  HIGH);

  digitalWrite(FLD,

  HIGH);

  digitalWrite(BRD,

  HIGH);

  digitalWrite(BLD,

  HIGH);



  digitalWrite(FRE,

  HIGH);

  digitalWrite(FLE,

  HIGH);
```

```
 digitalWrite(BRE,
HIGH);
digitalWrite(BLE,
HIGH);
Serial.println(" L
TURN"); }
else if (ROT > 1650)
{
 delay(500);
 digitalWrite(FRD,
 LOW);
 digitalWrite(FLD,
 LOW);
 digitalWrite(BRD,
 LOW);
 digitalWrite(BLD,
 LOW);
```

```
  digitalWrite(FRE,
  HIGH);
  digitalWrite(FLE,
  HIGH);
  digitalWrite(BRE,
  HIGH);
  digitalWrite(BLE,
  HIGH);
  Serial.println("    R
  TURN");
}
else
{
  delay(500);
  digitalWrite(FRE, LOW);
  digitalWrite(FLE, LOW);
  digitalWrite(BRE, LOW);
```

```
    digitalWrite(BLE, LOW);

    Serial.println(" STOP");

  }



}
```