Gianfranco Gonzales, Zain Smith, Oreva Omodior, Tyler Hayik, Cooper Young, Garrett Fairbanks

Senior Capstone - SLUG

Introduction

Garrett Fairbanks

Gianfranco Gonzales

Tyler Hayik

Introduction

Oreva Omodior

Zain Smith

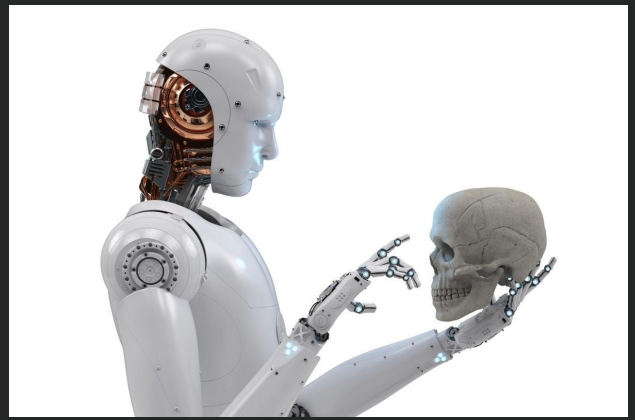
Cooper Young

SECTION 1

Motivation

Motivation

- General interest in robotics, computer engineering, and mobility



Section 2

Overview

Project Overview

Autonomous Environment Mapping Robot

Autonomous: Able to move without human control

Environment Mapping: Using LiDAR, the robot will create a point cloud of its surroundings.

Robot Design/Chassis: Our robot design features a treaded tank chassis with custom made enclosure.

Section 3

Objectives and Constraints

Objectives

- Demonstrate ability to climb stairs of a multitude of heights
- Complete designated path within the span of 10 minutes
- From beginning to end, prototype should be able to complete course/path within desired time
- Complete navigation around unforeseen obstacles within designated path
- Includes people, objects, walls, etc.

Objectives…

- Produces minimal amount of noise coming from parts such as the motors, chassis, etc.
- Prototype cost is less than the desired $200 region
- Slight flexibility with +- $25 ($1000 in our case)

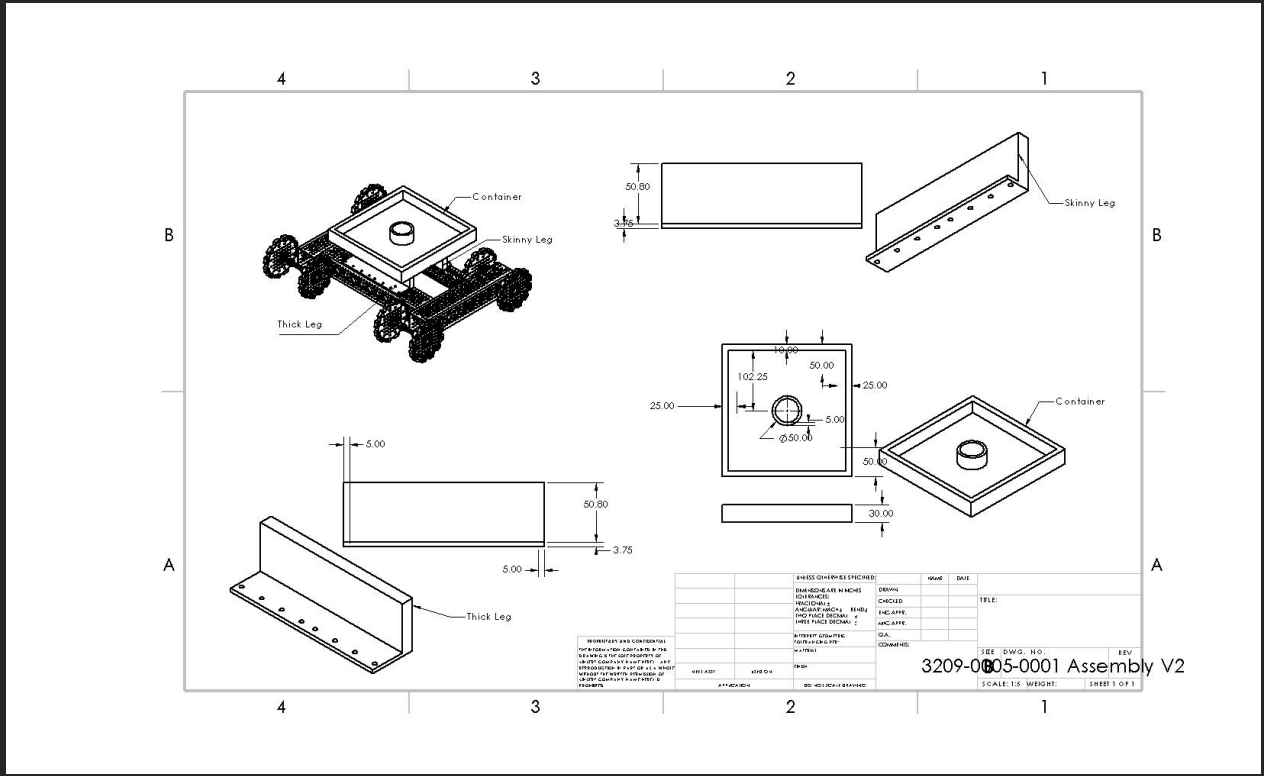
Constraints

- Self Powered
  - Contains its own battery
- Falls under the proper routes of safety
  - Doesn’t place others at risk of injury (such as falling) when navigating stairs
- Durability
  - Able to withstand slippage, rollbacks, falling, etc

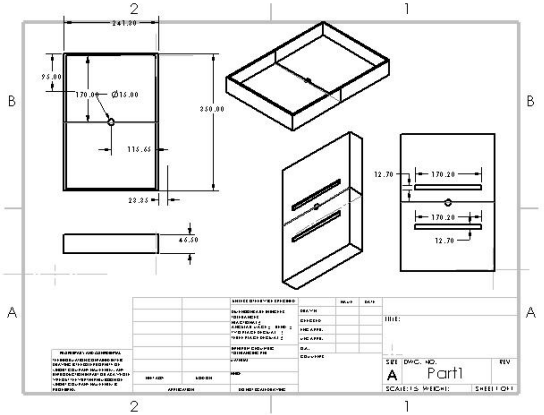
SECTION 3

Design/Development

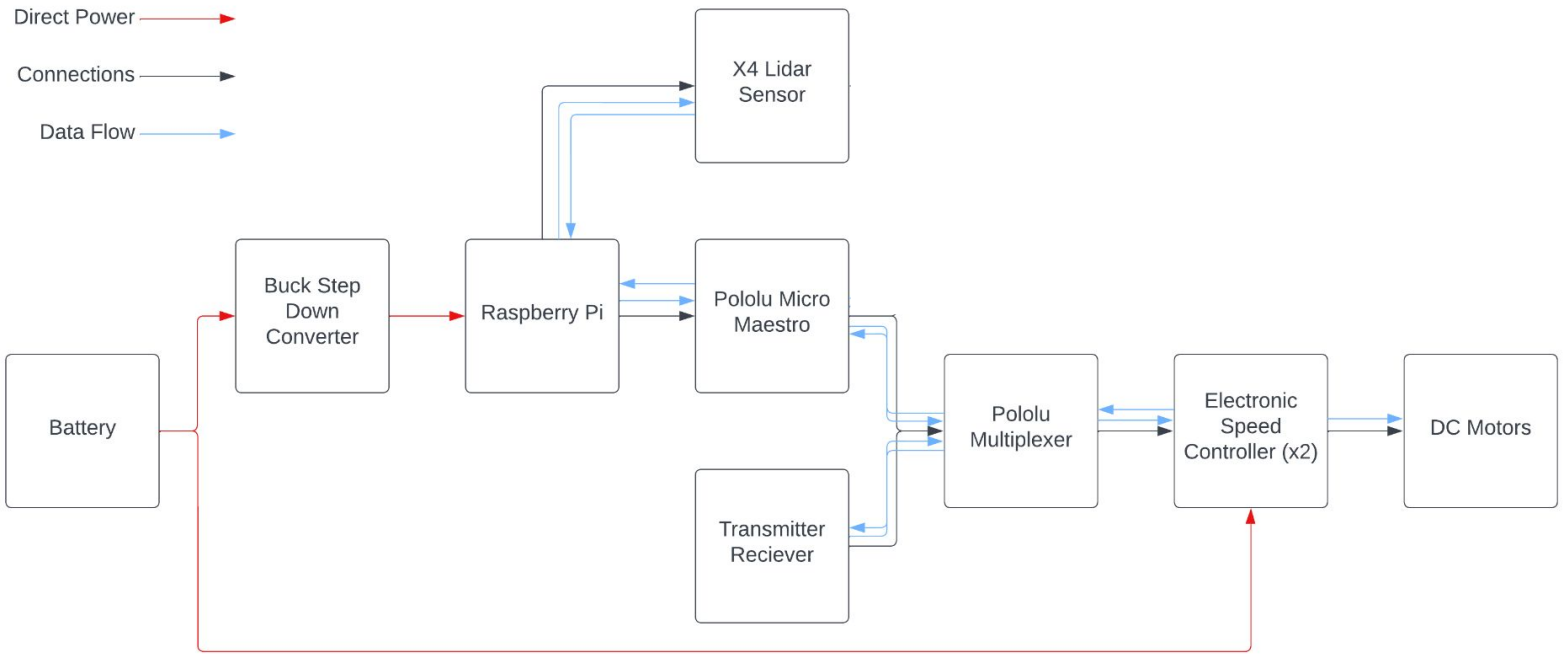
Chassis/Enclosure



Chassis/Enclosure



Overall Components



Chassis Movement

- Raspberry Pi inconsistent PWM output
- Custom Maestro python class for proper communication between Pi and Maestro
- Maestro supplies constant PWM signal from Raspberry Pi communication

Raspberry Pi

Pololu Micro Maestro

Chassis Movement

Custom Maestro Python Class

```python
class Controller:

    def __init__(self,ttyStr='/dev/ttyACM0',device=0x0c):
        # Open the command port
        self.usb = serial.Serial(ttyStr)
        # Command lead-in and device number are sent for each Pololu serial command.
        self.PololuCmd = chr(0xaa) + chr(device)
        # Track target position for each servo. The function isMoving() will
        # use the Target vs Current servo position to determine if movement is
        # occuring.  Upto 24 servos on a Maestro, (0-23). Targets start at 0.
        self.Targets = [0] * 24
        # Servo minimum and maximum targets can be restricted to protect components.
        self.Mins = [0] * 24
        self.Maxs = [0] * 24
```

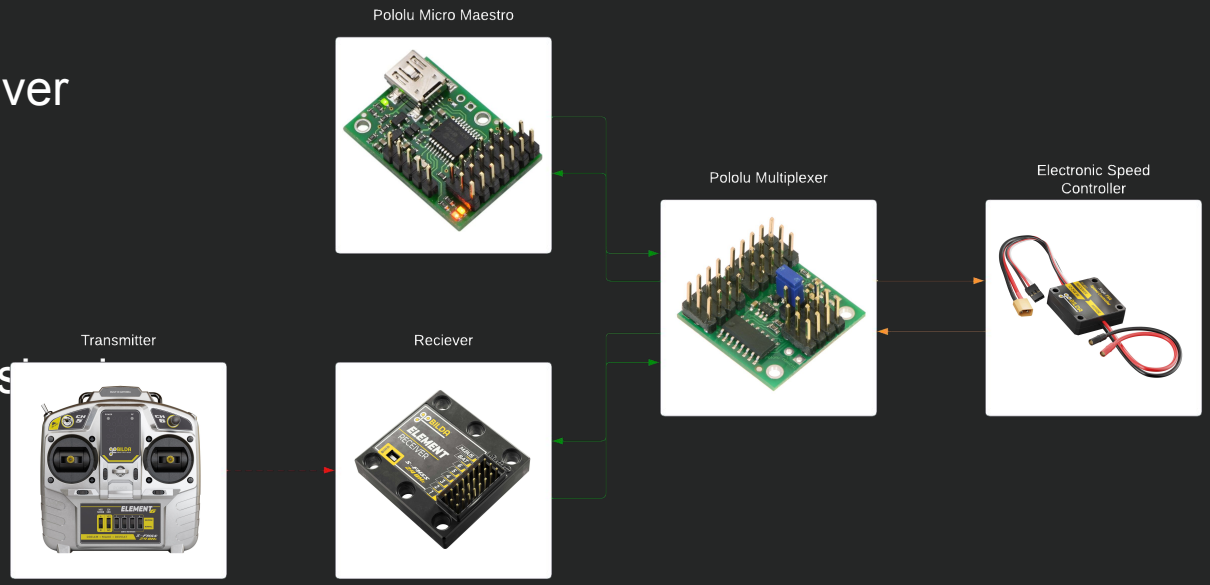
```python
def setTarget(self, chan, target):
    # if Min is defined and Target is below, force to Min
    if self.Mins[chan] > 0 and target < self.Mins[chan]:
        target = self.Mins[chan]
    # if Max is defined and Target is above, force to Max
    if self.Maxs[chan] > 0 and target > self.Maxs[chan]:
        target = self.Maxs[chan]
    #
    lsb = target & 0x7f #7 bits for least significant byte
    msb = (target >> 7) & 0x7f #shift 7 and take next 7 bits for msb
    cmd = chr(0x04) + chr(chan) + chr(lsb) + chr(msb)
    self.sendCmd(cmd)
    # Record Target value
    self.Targets[chan] = target
```

```python
def setSpeed(self, chan, speed):
    lsb = speed & 0x7f #7 bits for least significant byte
    msb = (speed >> 7) & 0x7f #shift 7 and take next 7 bits for msb
    cmd = chr(0x07) + chr(chan) + chr(lsb) + chr(msb)
    self.sendCmd(cmd)
```

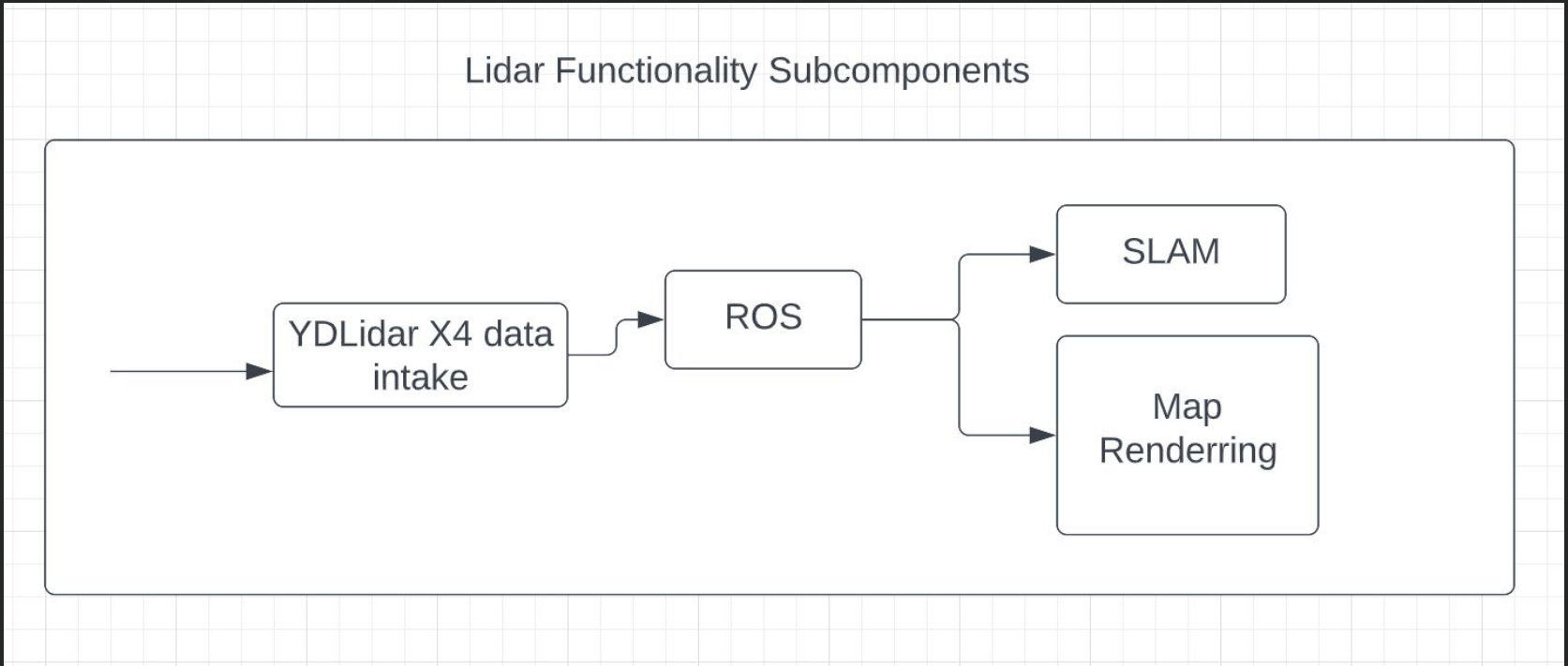
```python
def setAccel(self, chan, accel):
    lsb = accel & 0x7f #7 bits for least significant byte
    msb = (accel >> 7) & 0x7f #shift 7 and take next 7 bits for msb
    cmd = chr(0x09) + chr(chan) + chr(lsb) + chr(msb)
    self.sendCmd(cmd)
```

Chassis Movement

- Need to split control input between Maestro & Receiver
- Pololu Multiplexer
  - 2 Inputs
  - 1 Output
  - 1 Switch
- Deadswitch capability for s switching



Navigation


Lidar Functionality Subcomponents
YDLidar X4 data intake
ROS
SLAM
Map Renderring

LiDAR

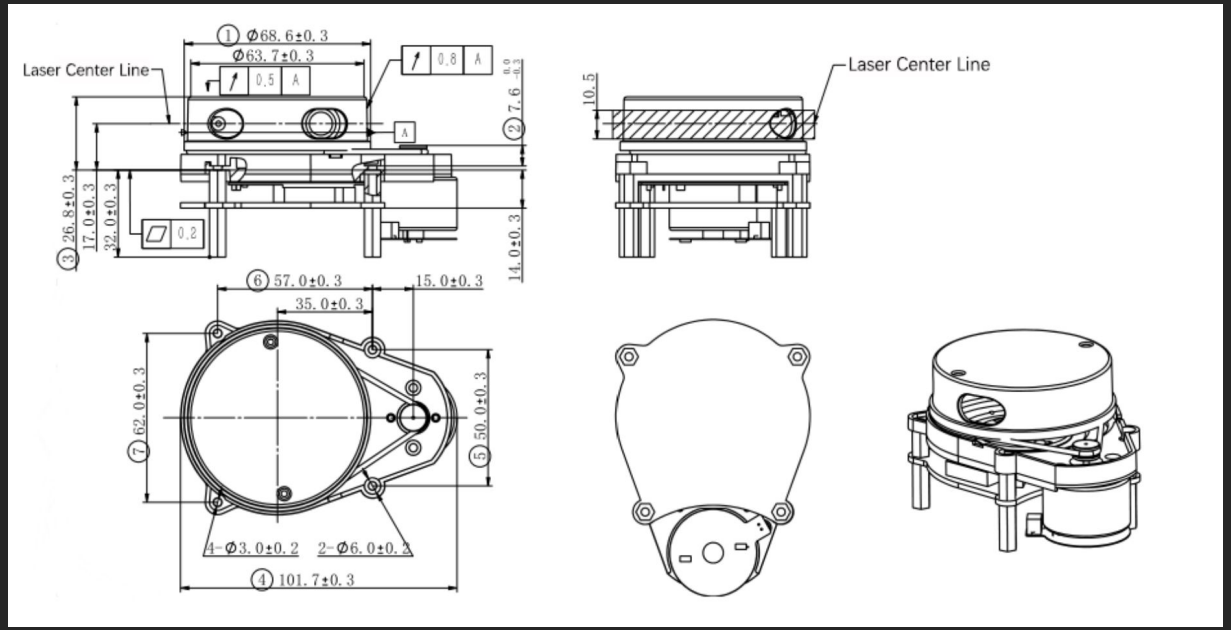
1. Light Detection and Ranging

   Measure ranges via light (pulsing laser)

2. YDLidar X4

   Inexpensive, 360° with 10m detection radius

YDLIDAR X4 CAD Drawing

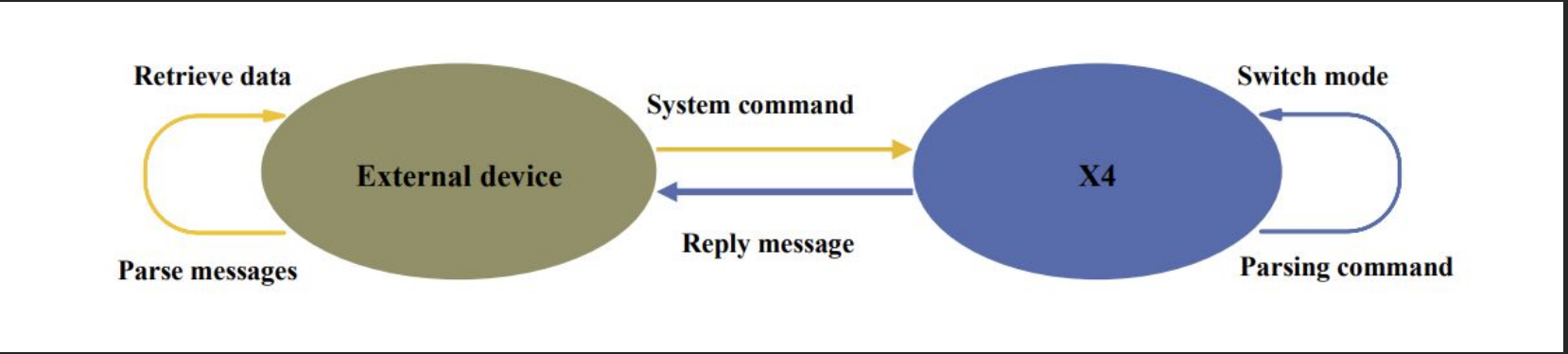


LiDAR…

Component	Light Source	Scanning freq	Measuring distance	Working Voltage	Weight	Operating Temp	Size
YDLidar X4	Infrared (785nm)	~ 7 Hz	0.12m - 10m	5V	193 g	Room Temp	35 x 35 x 60 mm

LiDAR Specifics

- LiDAR range
- LiDAR signal example



LiDAR…

Initial code using PyLidar3

Gather angle and data -> print

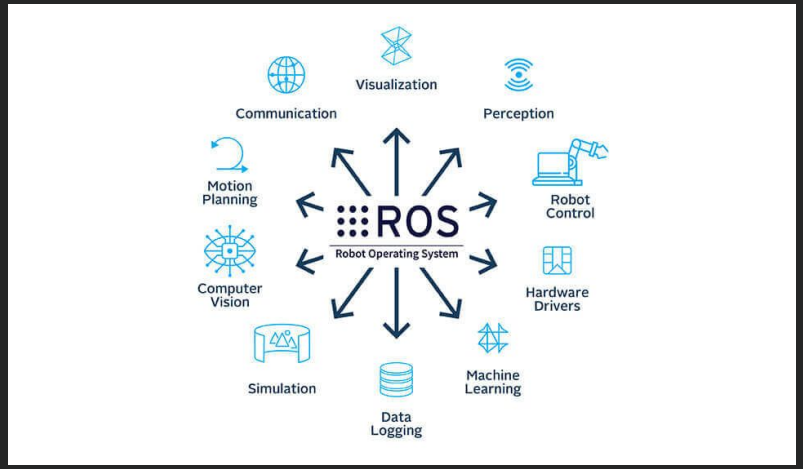
```
port = "/dev/ttyUSB0" #linux
Obj = PyLidar3.YdLidarX4(port) #PyLidar3.your_version_of_lidar(port,chunk_size)


data = []
if(Obj.Connect()): #continuous loop
    print("hello")
    print(Obj.GetDeviceInfo())
    gen = Obj.StartScanning()
    t = time.time() # start time

    #Calculate distance and insert conditions into for loop
       # print(next(gen))

    while(time.time() - t) < 5:
        print(next(gen))
        time.sleep(0.5)
    for angle in range(0,360):
        #data = next(gen)
        if(data[angle]>1000):
            x[angle] = data[angle] * math.cos(math.radians(angle)) #convert to radians
            y[angle] = data[angle] * math.sin(math.radians(angle)) #convert to radians
            print("The x angle is : " + x[angle])
            print("The y angle is : " + y[angle])
            #time.sleep(0.5)
    Obj.StopScanning()
    Obj.Disconnect()
else:
    print("Error connecting to device")
```

SLAM

- Simultaneous localization and mapping
  - allows map to be built + find robot’s location
- Hector SLAM
  - ROS nodes
- Use information such as
  - Number of wheel revelations
  - Data from sensors (camera, lidar, imaging sensors)
- Many types of SLAM
  - vSLAM, ToF, LiDAR
  - Lasers are more precise

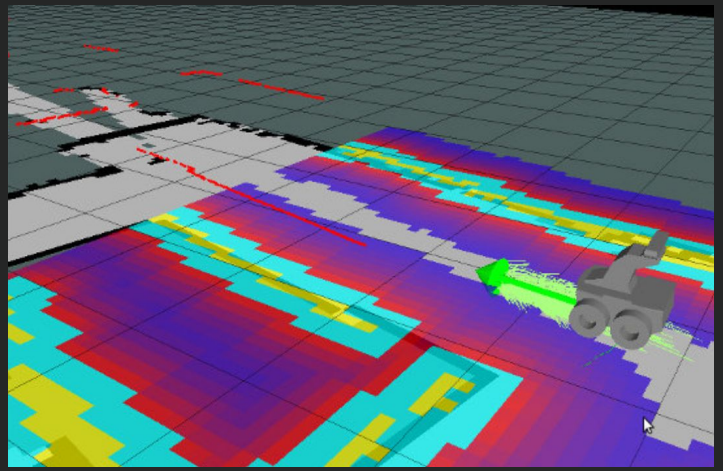
ROS

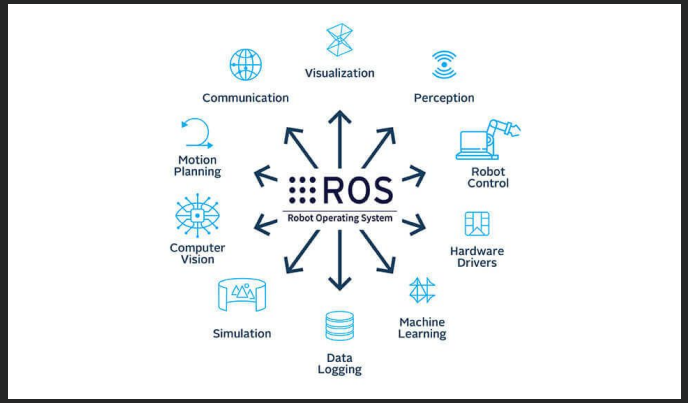
- Robot Operating System
  - open-source framework specific to drones, vehicles, mobile robots
- ROS is not specific to any language
- Contains a multitude of packages (useful for SLAM, map rendering, handling lidar data)
- What other projects and instances has it been used
- How will we be using it



ROS: Rviz

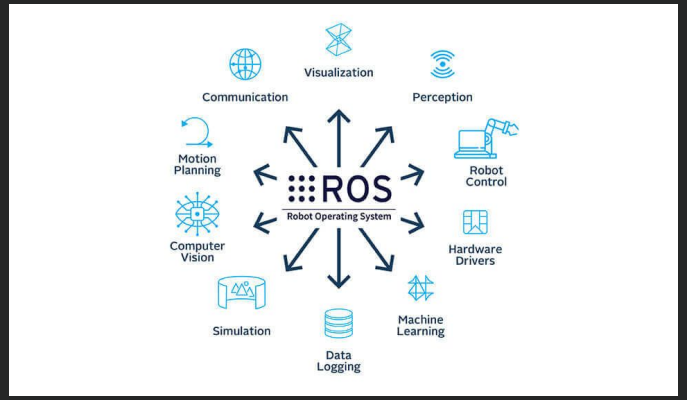
- Visualization tool for LIDAR data and environment mapping

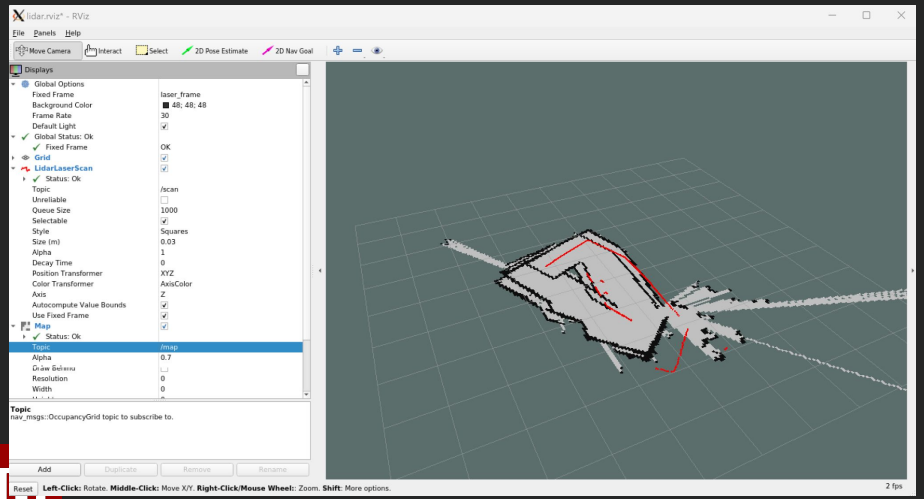




ROS: Hector SLAM Mapping

- SLAM approach without odometry
- Leverages high update rate of LIDAR





Mathematical Modeling

Convert Cartesian to Polar

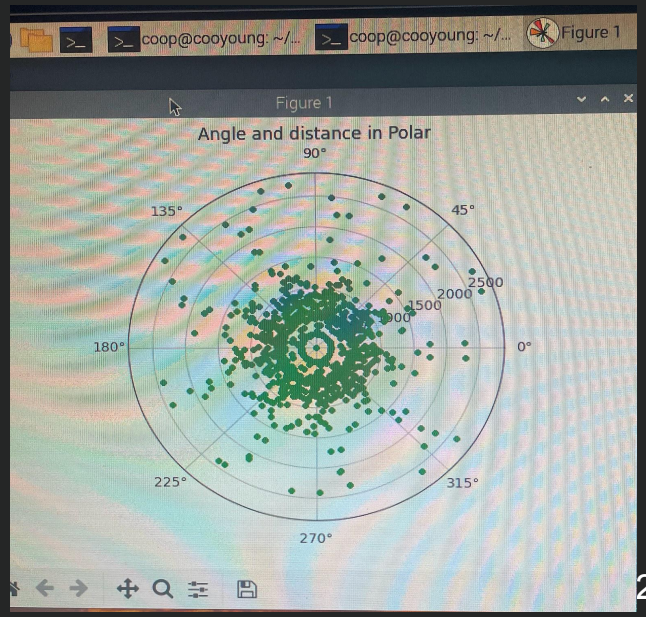
$$r = \sqrt{x^2 + y^2}$$

$$\theta = \tan^{-1}\frac{y}{x}$$

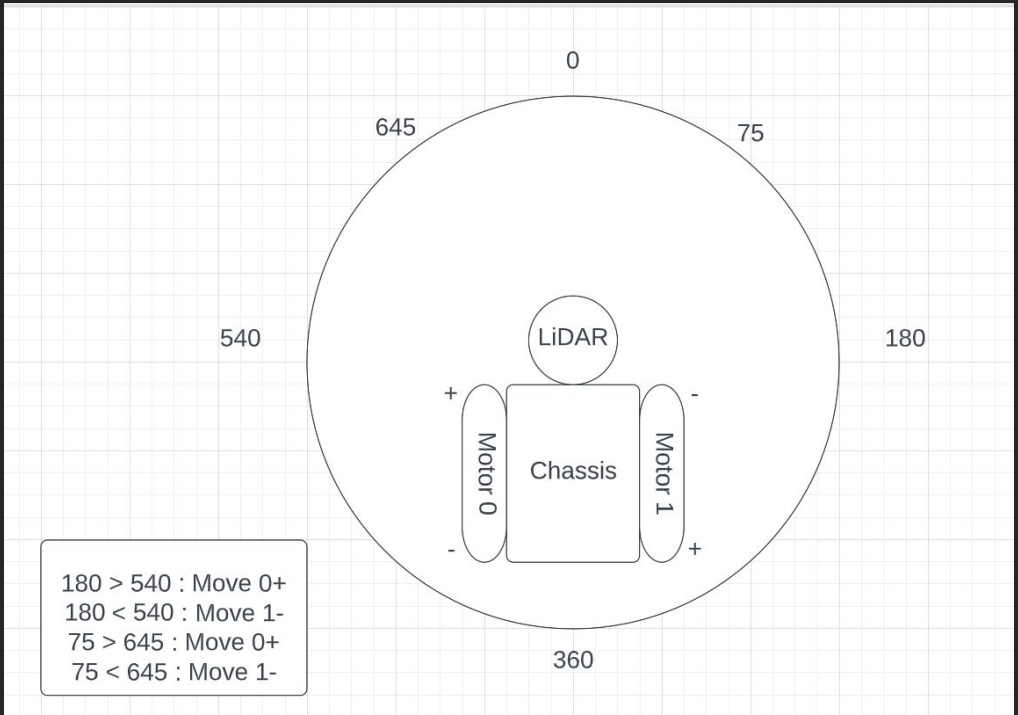
note: may need to add 180°

r in Lidar is distance from detected point

Initial Lidar data is in polar form, converted to Cartesian in Rviz



Directional Computations



Economic Aspects

Economic Aspects

- Consider manufacturing our own embedded systems and chassis system to reduce supply chain cost
- Integrate more components to cut down on overall cost
- Expensive chassis system
  - Consider different design

Societal Aspects

Societal Aspects

- There are many positive applications for designing a self-mapping robot
  - High risk area for humans
  - Search and rescue
  - Autonomous vacuum/lawn mower
- Optimization of cost to reach more customers
  - Lower motor and system cost
  - Experiment with alternative Lidar sensors

Societal Aspects…

- Ease of use for anyone regardless of age, gender, or culture
  - Simple usability
- Consideration for design change based off environment
  - Varying stair heights
  - Indoor/Outdoor environment

Ethical, Political, and Legal Issues

Ethical issues

- Hazards commonly associated with mobile robots
  - Failure of circuitry
  - Causing harm to bystanders
- Issues associated with autonomous vehicles

Legal issues

- Intellectual Property constraints
  - Raspberry pi, YDLIDAR X4, chassis system
  - Consider building a personalized embedded system

Manufacturability and Sustainability

Manufacturability

- The chassis is built by goBilda and has multiple guides online.
- The encasing has special places for each component inside.
- ROS has multiple guides online.
- Software is commented to allow for easy understanding and manipulation.

Sustainability

- To make our software sustainable, we ensured it was error free and able to run on mac and windows machines.
  - The software is supposed to stop Slug before hitting a surface to prevent breaking the LiDAR sensor
- By creating our own encasing using 3D printing, we minimized chip-on-chip collisions to reduce broken pieces.
- Slug is designed to avoid humans, but in case of an emergency, there is an instant stop switch on the controller, to ensure no one is harmed.
- To ensure a longer life-cycle, batteries should be properly charged, and the components turned off when not in use.

SECTION 5

Implementation and Testing

Maestro Testing (Time Based Movement)

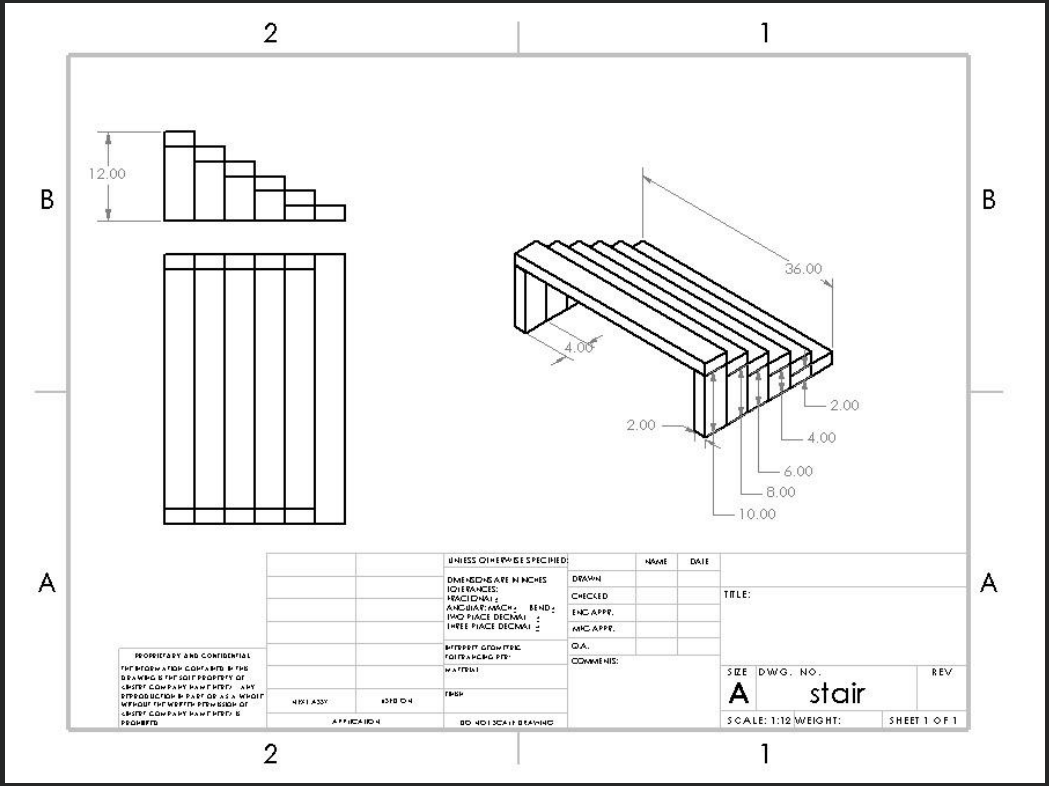
Proves movement without controller

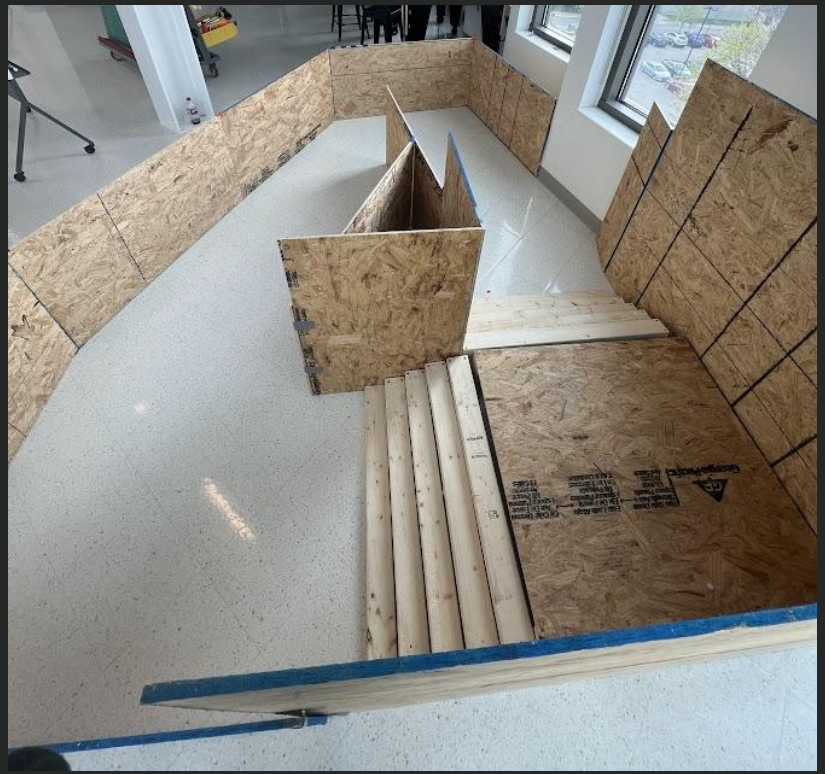
Controlled Movement

Proves basic navigation in maze

Arena Creation

We designed our stairs in Solidworks before creating it in real life.





Final Code Program

First terminal

Run all_nodes.launch

- Opens rviz and creates map

lidar.launch

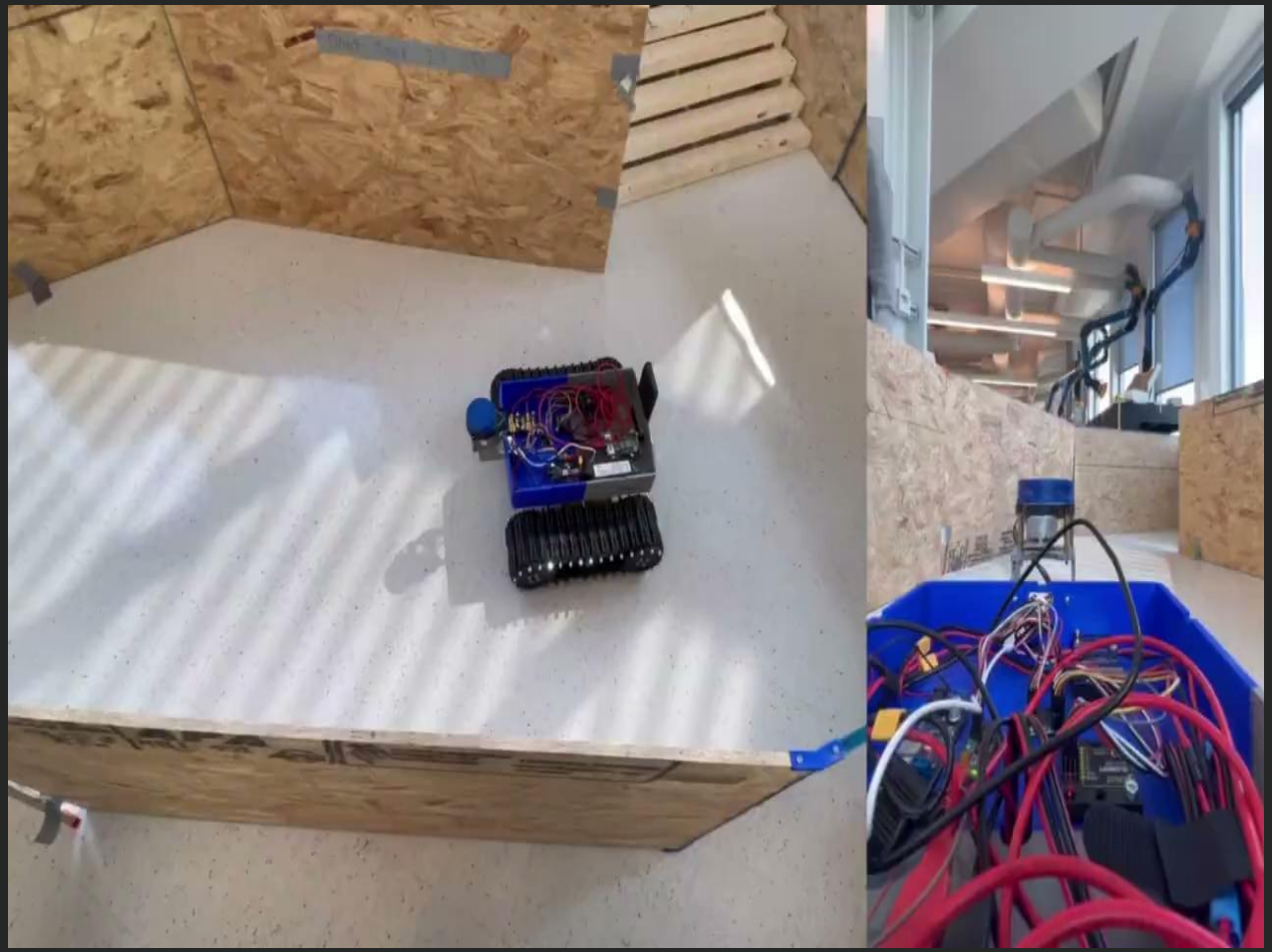
- Spins YDLidar
- Sends distances in ROS format

Second terminal

Runs test.py

- PID loop
- Initializes a ROS topic (/scan)
- Constantly receives distances from YDLidar
- Updates errors accordingly

Results, Analysis and Video Demonstration

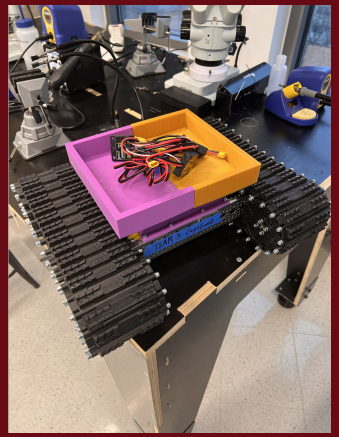
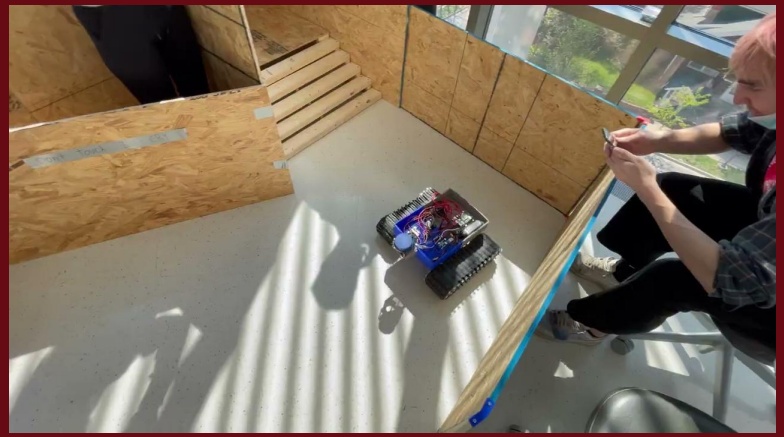
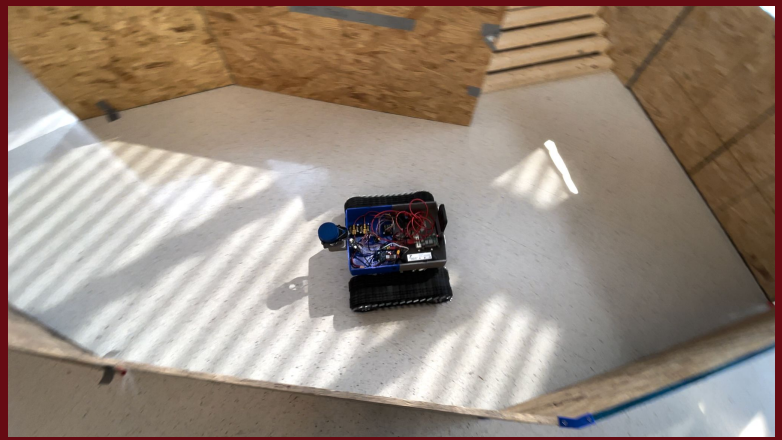
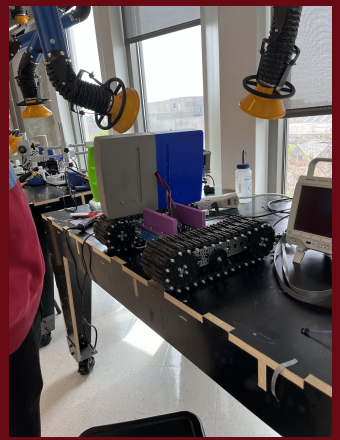


Acknowledgement & References

Acknowledgements

- Luddy ISE Department
- Professor Himebaugh
- Professor Lasassmeh
- Peers and Classmates

References

- automaticaddison. (2021, May 22). How to build an indoor map using ROS and lidar-based Slam. Automatic Addison. Retrieved December 9, 2022, from https://automaticaddison.com/how-to-build-an-indoor-map-using-ros-and-lidar-based-slam/
- Goods transport stair climber robot. Nevon Projects. (2021, December 24). Retrieved December 9, 2022, from https://nevonprojects.com/goods-transport-stair-climber-robot/
- Google. (n.d.). US7969558B2 - high definition lidar system. Google Patents. Retrieved December 9, 2022, from https://patents.google.com/patent/US7969558B2/en
- Google. (n.d.). US8644991B2 - maneuvering robotic vehicles. Google Patents. Retrieved December 9, 2022, from https://patents.google.com/patent/US8644991B2/en
- Google. (n.d.). US9527213B2 - automatic stair-climbing robot platform. Google Patents. Retrieved December 9, 2022, from https://patents.google.com/patent/US9527213B2/en
- Home. Karl H. Bartels GmbH. (n.d.). Retrieved December 9, 2022, from https://www.bartels-germany.com/electrically-operated-stair-climber/stair-climber-domino/
- Implement simultaneous localization and mapping (SLAM) with LIDAR SCANS. Implement Simultaneous Localization And Mapping (SLAM) with Lidar Scans - MATLAB & Simulink. (n.d.). Retrieved December 9, 2022, from https://www.mathworks.com/help/nav/ug/implement-simultaneous-localization-and-mapping-with-lidar-scans.html
- Lakkundi, N. (2014, May 4). How to build a stair climbing robot? How to Build an Stair Climbing Robot? Retrieved December 9, 2022, from https://www.skyfilabs.com/project-ideas/stair-climbing-robot
- Moore, E. (n.d.). Stair climbing tracked robots -left to right the MURV [26] , Vanguard ... Retrieved December 9, 2022, from https://www.researchgate.net/figure/16-Stair-climbing-tracked-robots-left-to-right-the-MURV-26-Vanguard-25-Wheeled_fig7_274084271
- What is slam? simultaneous localization and mapping. GeoSLAM. (2022, August 9). Retrieved December 9, 2022, from https://geoslam.com/us/what-is-slam/#:~:text=What%20is%20LiDAR%20SLAM%3F,an%20active%20laser%20%E2%80%9Cpulse%E2%80%9D.
- https://www.ydlidar.com/Public/upload/files/2022-06-28/YDLIDAR%20X4%20Data%20sheet%20V1.1(211230).pdf YDLIDARX4 datasheet

Thank You!