MARWIN

A journey to Space dream

Vyshak P P
VAS15CS113

Joe S Kolengaden
VAS15CS060

Mohamed Sahin V A
VAS15CS068

Contents

Abstract

The basic aim behind the project is to address the need of open source projects in space industry. MARWIN is intended to be one of the first open source DIY space rover projects which could be developed very easily. The rover is designed to be highly customizable with a modular design approach. MARWIN could open up the future for rapidly deployable rover solutions for exploration of other planets. MARWIN intends to reduce the manufacturing cost and development time of rovers by implementing a quick and standardized design approach which could be customized very easily.

Problem statement

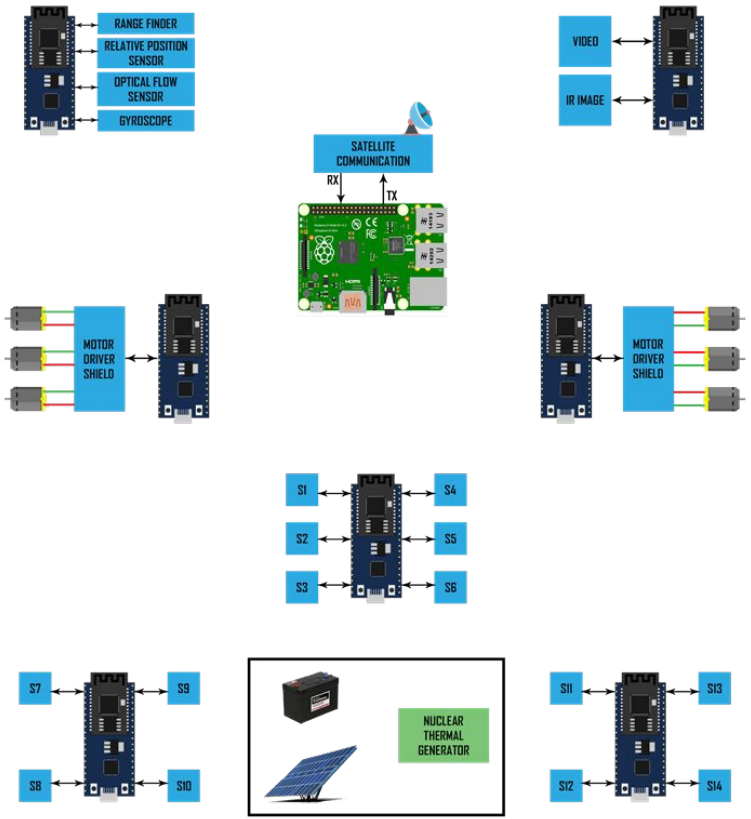
To develop an open source space rover by utilizing all possible open source materials & resources and also make the rover flexible deployable for multiple purposes including research, rescue, military and defence purposes.

Methodologies

- Architecture of MARWIN
- Rocker-Bogie Mechanism
- Sensor cluster
- Obstacle avoidance
  - Obstacle detection
  - Obstacle searching
  - Rerouting
- Object detection
- WebApp

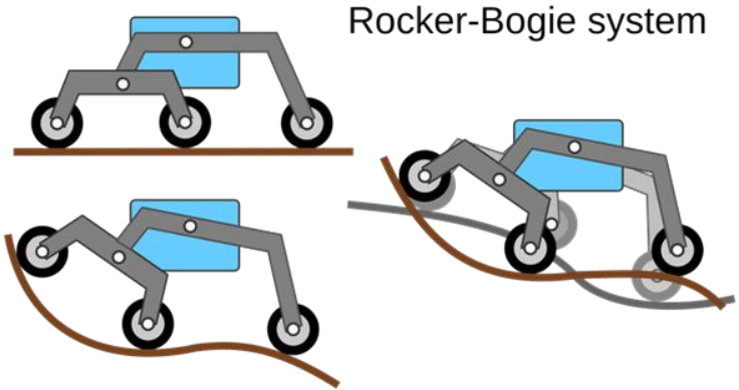
Architecture of MARWIN

Modular designs are used in this architecture.



Rocker-Bogie mechanism

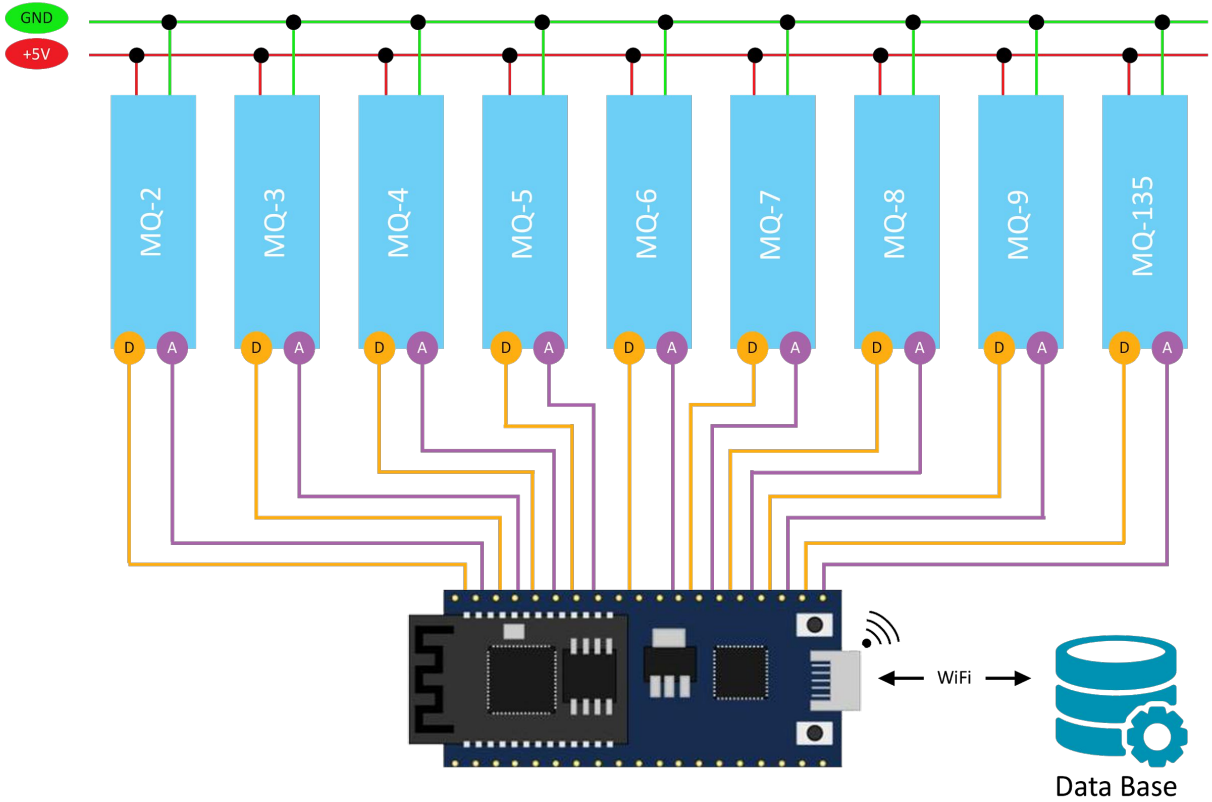
- The ”rocker” part of the term comes from the rocking aspect of the larger, forward leg on each side of the suspension system. These rockers are connected to each other and the vehicle chassis through a differential.
- The ”bogie” part of the term refers to the smaller, rearward leg that pivots to the rocker in the middle and which has a drive wheel at each end.
- The rocker-bogie design has no springs or stub axles for each wheel, allowing the rover to climb over obstacles (such as rocks) that are up to twice the wheel’s diameter in size while keeping all six wheels on the ground.



Gas Sensor Cluster

Model

Model	Target Gas
MQ2	General combustible gas
MQ3	Alcohol
MQ4	Natural gas, Methane
MQ5	LPG, Natural gas, Coal gas
MQ6	LPG, Propane
MQ7	Carbon Monoxide
MQ8	Hydrogen
MQ9	CO and Combustible gas
MQ135	Air Quality Control



Obstacle avoidance

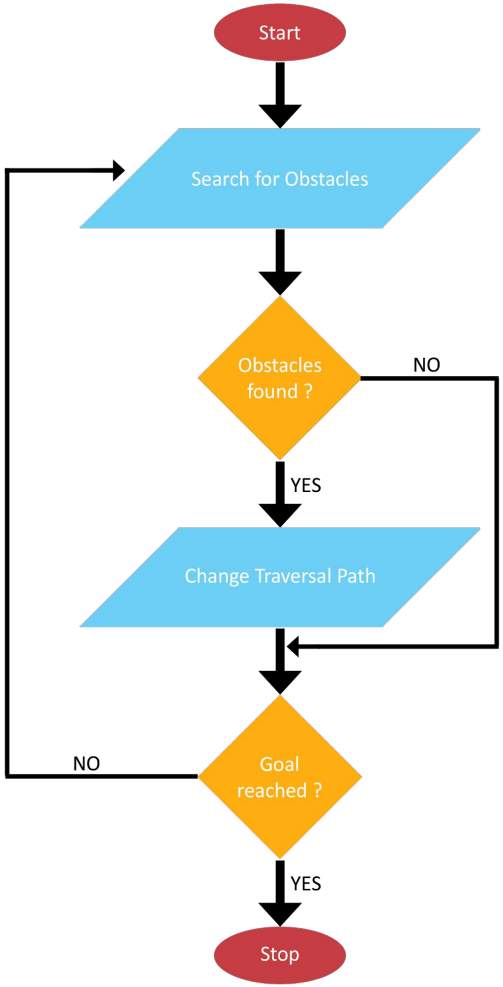
Obstacle avoidance contains basic 3 algorithms namely,

- Obstacle detection
- Obstacle searching
- Rerouting

Obstacle detection

Algorithm & FlowChart

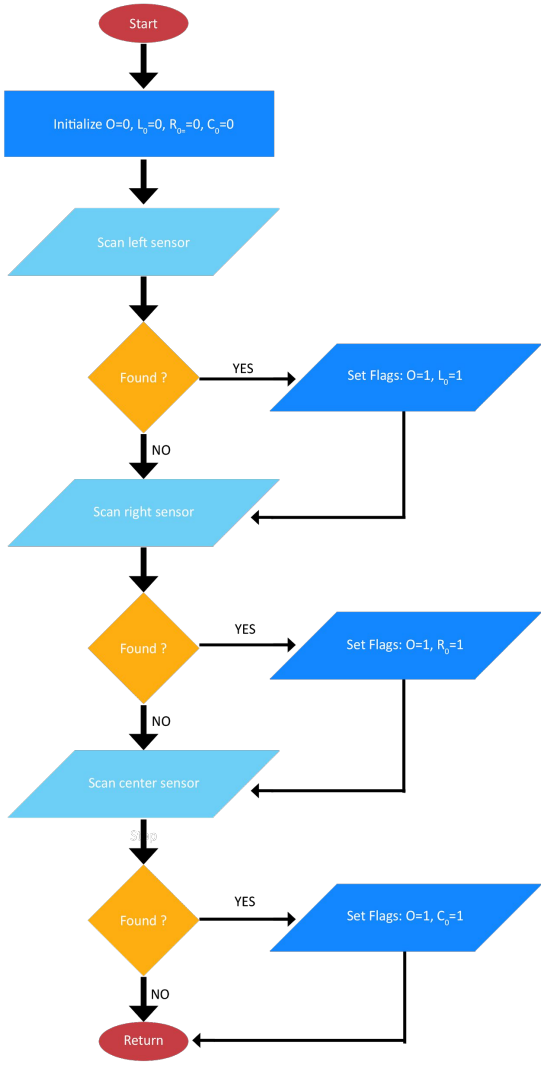
1. Start
2. Search for obstacle using ***Obstacle Searching Algorithm***
3. If found then change traversal path and goto step 5
4. Else goto step 5
5. Check whether goal is reached ?
6. If yes then goto step 8
7. Else goto step 2
8. Stop



Obstacle Searching

Algorithm & FlowChart

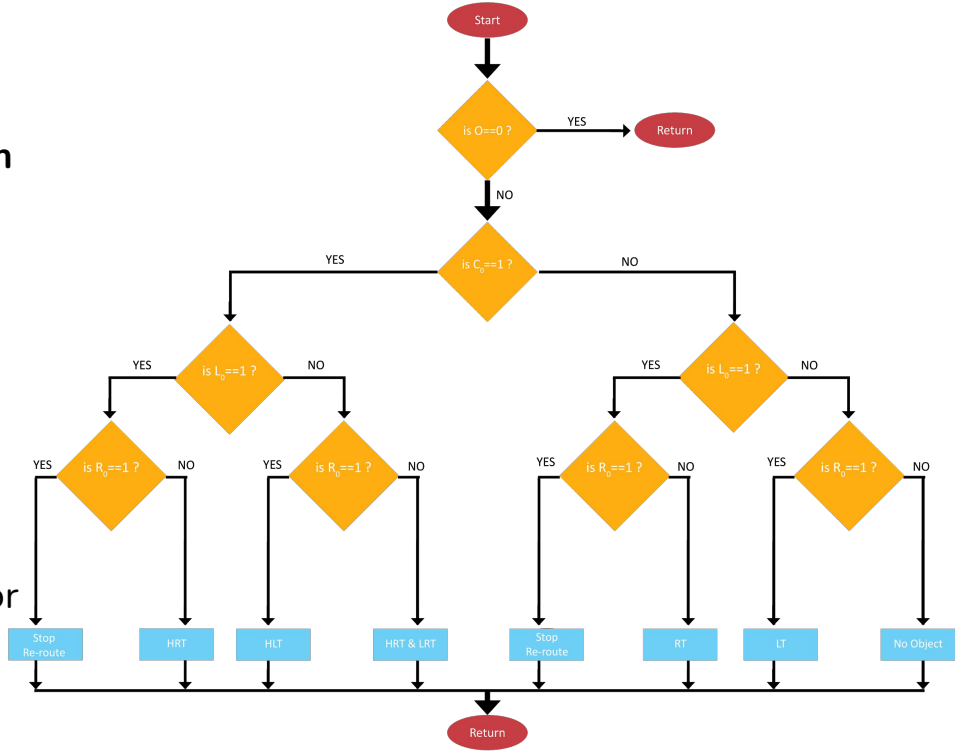
1. Start
2. Initialize flag variables O=0, Lo=0, Ro=0, Co=0
3. Scan Left sensor for obstacles.
4. If found then set flags O=1, Lo=1 and goto step 6
5. Else goto step 6
6. Scan Right sensor for obstacles.
7. If found then set flags O=1, Ro=1 and goto step 9
8. Else goto step 9
9. Scan Center sensor for obstacles.
10. If found then set flags O=1, Co=1 and goto step 12
11. Else goto step 12
12. Return to ***Obstacle Detection Algorithm*** with sensor values.



Rerouting

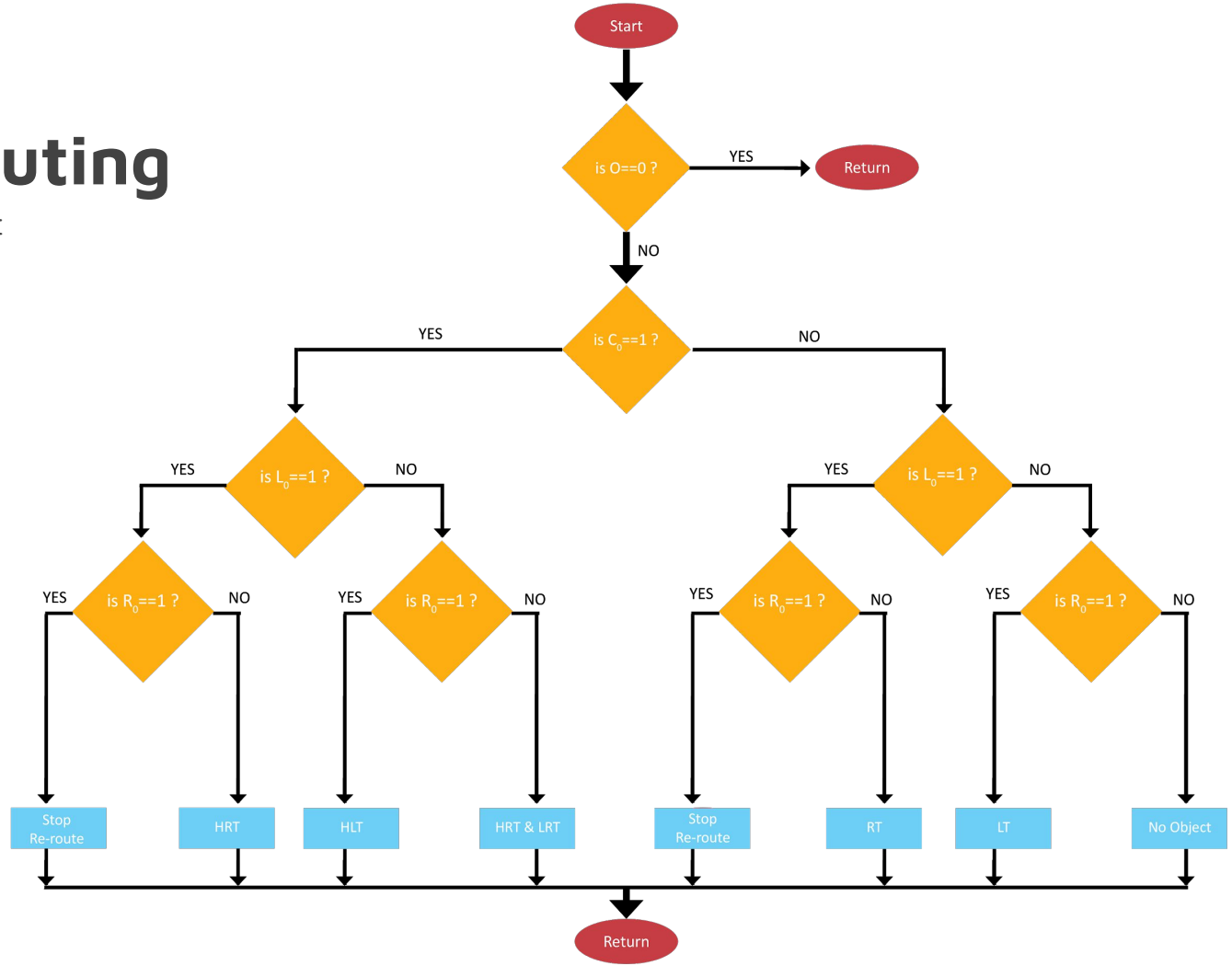
Algorithm & FlowChart

1. Start
2. If O=0 then Return to **Obstacle Detection Algorithm**
3. Else do steps a to h
   a. If Lo=1 & Ro=1 & Co=1 then stop & reroute
   b. If Lo=1 & Ro=0 & Co=1 then HRT
   c. If Lo=1 & Ro=0 & Co=0 then RT
   d. If Lo=0 & Ro=1 & Co=1 then HLT
   e. If Lo=0 & Ro=1 & Co=0 then LT
   f. If Lo=0 & Ro=0 & Co=1 then HRT/HLT
   g. If Lo=1 & Ro=1 & Co=0 then stop & reroute
   h. If Lo=0 & Ro=0 & Co=0 then No Obstacles
4. Return to ***Obstacle Detection Algorithm*** with sensor values.



Rerouting

FlowChart



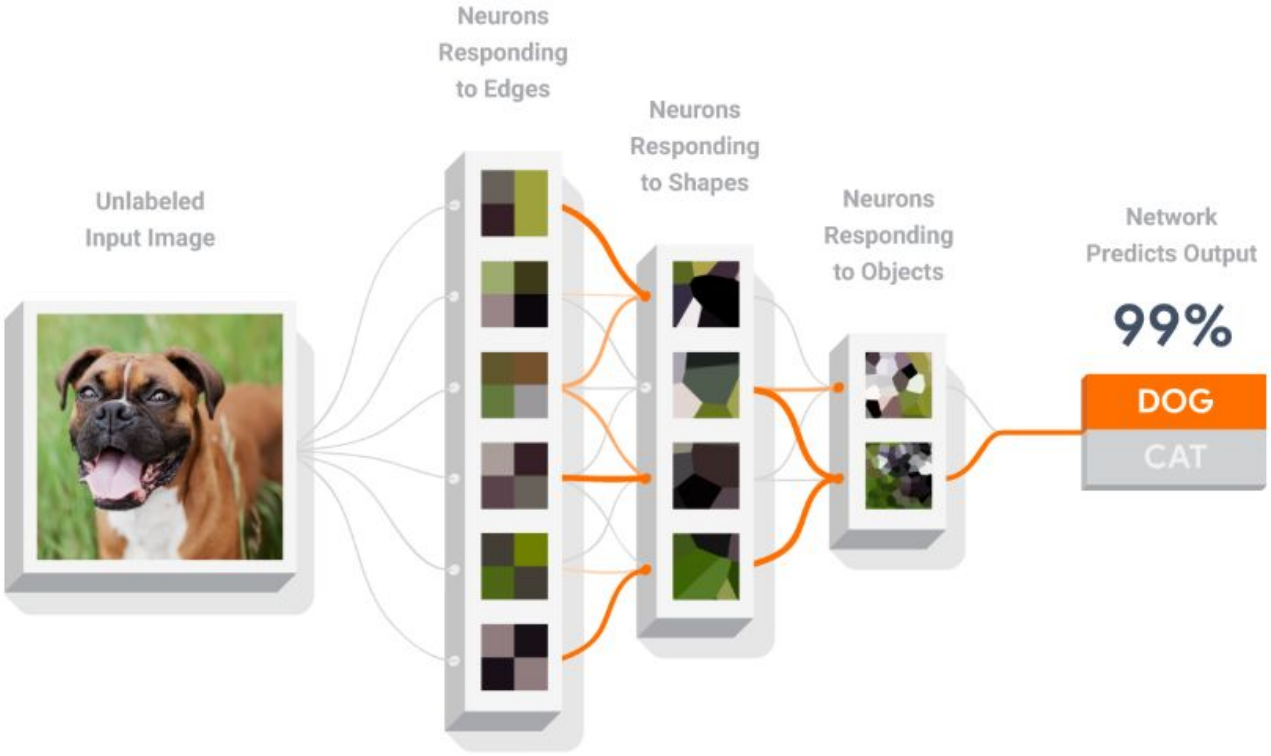
Object detection

Prior detection systems repurpose classifiers or localizers to perform detection. They applythemodeltoanimageatmultiplelocationsandscales. Highscoringregionsofthe image are considered detections.

Here we used TensorFlow for object detection.

Working


GATHER DATA
EXPLORE DATA
PREPARE DATA
BUILD, TRAIN & EVALUATE MODEL
TUNE HYPER-PARAMETERS
DEPLOY MODEL

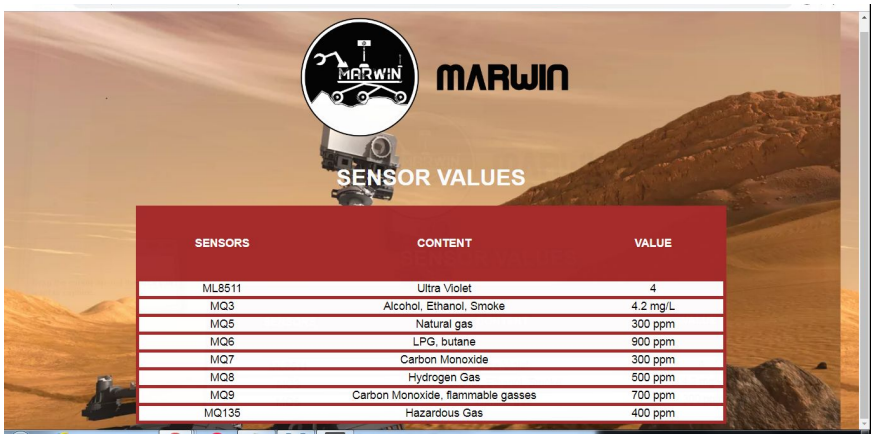

Unlabeled Input Image
Neurons Responding to Edges
Neurons Responding to Shapes
Neurons Responding to Objects
Network Predicts Output
99%
DOG
CAT

WebApp

A web application is a computer program that utilizes web browsers and web technology to perform tasks over the Internet.

Our web app is created using:

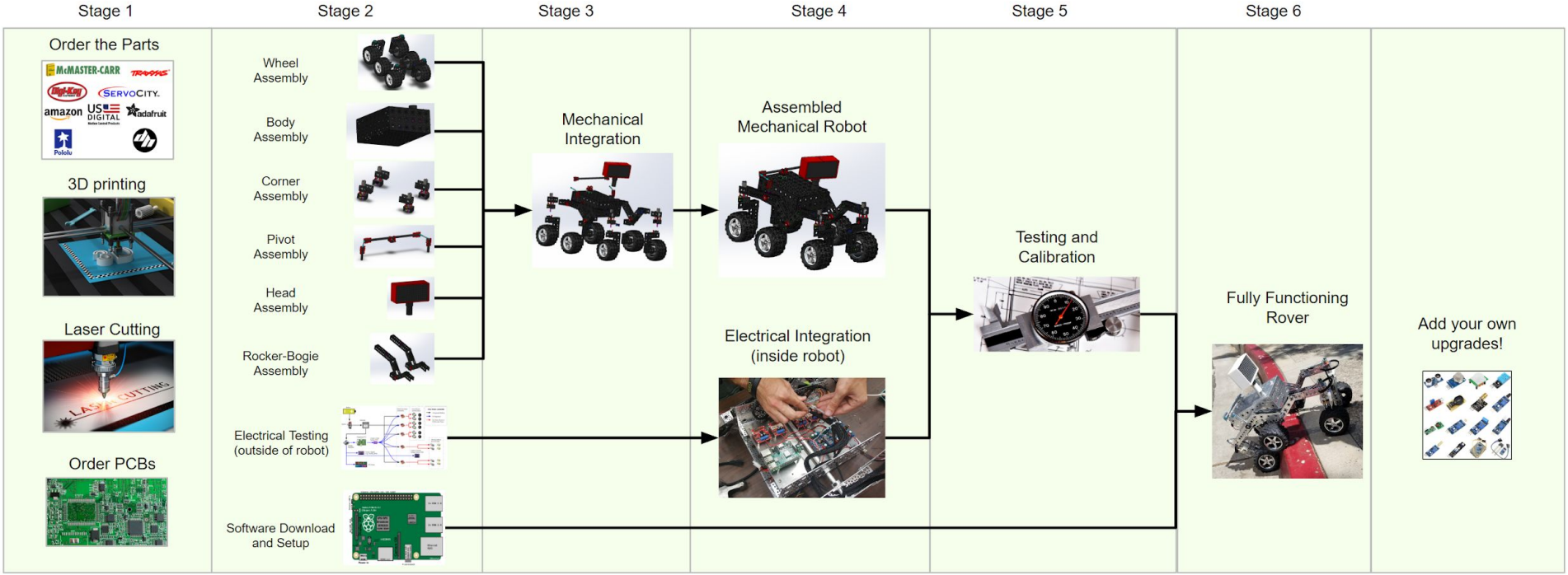
- Flask : Python Based mini-Web Framework
- MongoDB : Database Server
- Pymongo : Database Connector
- HTML : For webpage



Implementation Plan

- Rocker-Bogie system.
- Scientific equipments (sensor clusters).
- Power management system.
- Autonomous navigation.
- Video streaming and analysis.
- Testing and analysis.
- Modification and finalizing the design

RoadMap



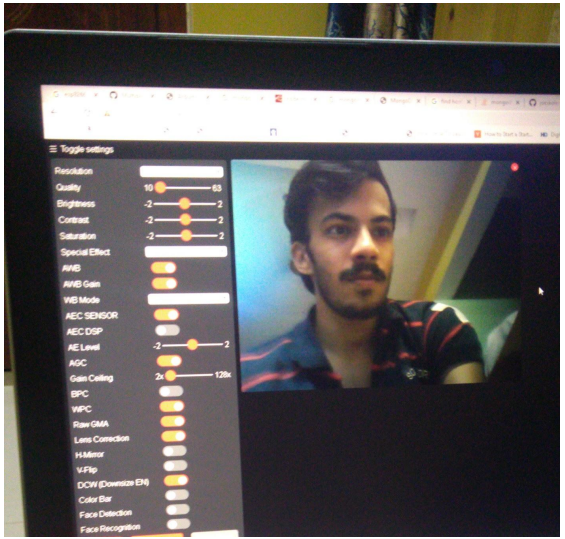
Reference link:- https://github.com/nasa-jpl/open-source-rover/blob/master/images/roadmap.PNG

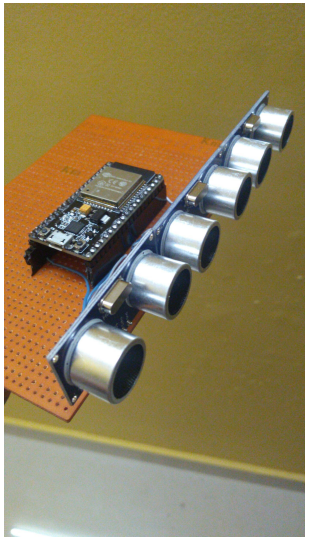
Current Progress

● Rocker-Bogie Mechanism	Designed
● Object Detection	On developing
● Gas Sensor Cluster	Developing cluster
● Wheel Assembly	Implemented
● WebApp	On developing
● Navigation System	On developing

Current Progress

Current Progress





Issues Faced

- Rocker-Bogie Mechanism
- Wheel Assembly
- Sensor Clustering
- Response Time
- LIVE Object Recognition
- Robotic Arm
- Power Source
- Cost

Conclusion and Future scope

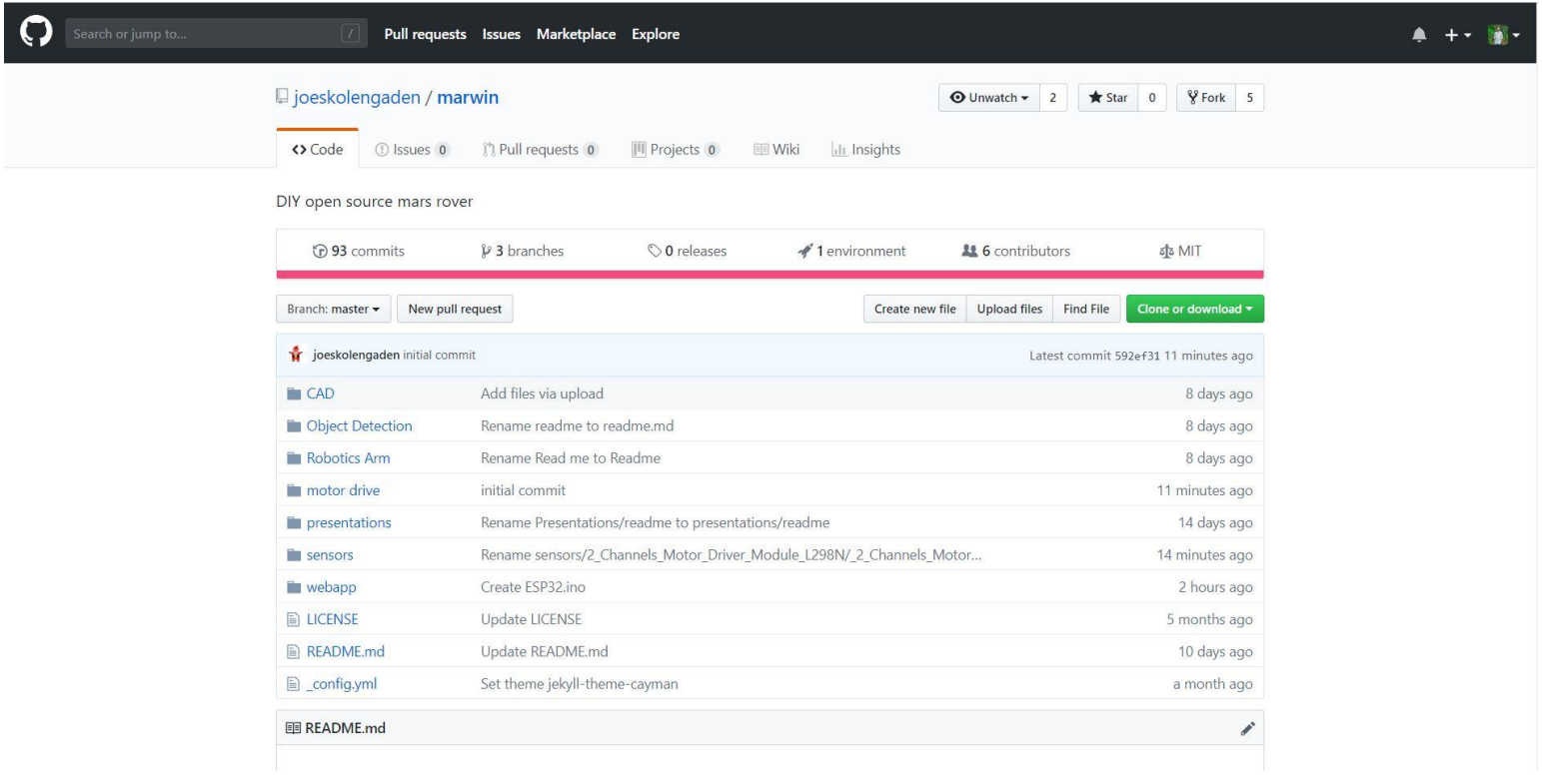
As a team we achieved far more in a short timespan. The journey through the developing and integrating part of this project was a great one. In the present situation there are a high amount disasters happening unexpectedly also different kinds of situations are rising where the human kinds are not reachable. So we are expecting our space rover based marwin could also implement in such situations and can obtain a positive result. In future we will be able to implement the present prototype on the basis of the needs. That can be based on the situations. By improving we can make this rover more user friendly by also including machine learning concept to it. This makes the rover to learn the ongoing datas and can act as a highly AI based robot which will be completely autonomous and the human efforts decreases.

Note

Choose the right materials and good quality electronics for making electro-mechanical projects

GITHub

Open sourced every bit of code via Github.com



Thank You!