Thirty Gallon Robot, Movement and Obstacle Avoidance

April 22, 2022

Presented by:

Fernando Diaz, UlugBek Abdullayev, Jonathan Gomez, Brandon Jester

Team Mentor:

Han Peng

Robotics

Robotics is a great field, continually expanding. Many manufacturers currently use a form of robotics to facilitate manufacturing of products given the high demand.

Autonomous driving is also a field of robotics being currently tested, and many big companies such as Tesla, Ford, and others currently search for a solution.

The Client: Dr. Leverington

- Lecturer of Computer Science at NAU
- Wants a “wow” factor to bring in new engineering students
    - Autonomous touring robot will inspire more people to join the school

The Thirty Gallon Robot

Big Picture:

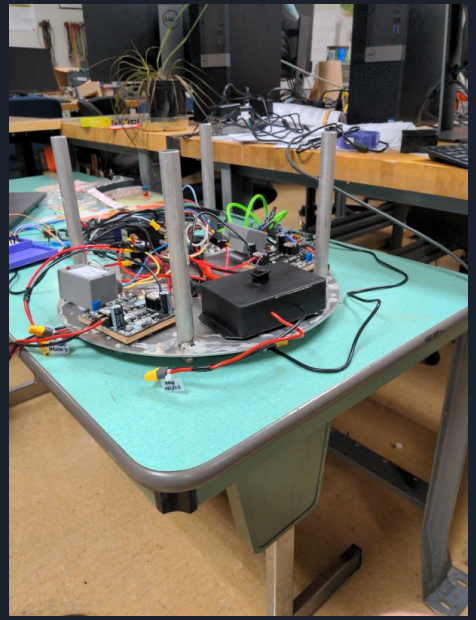
- Full touring implementation requires multiple modules operating together
- Obstacle avoidance and autonomous movement are crucial implementations for further development
- Need an affordable robot that can give tours and easy to build

Project Objectives:

- Autonomous navigation module can be difficult to implement, given the vast amount of variables that may be present, such as obstacles.
- Obstacle avoidance must also be recognized and handled appropriately.

Problem Overview

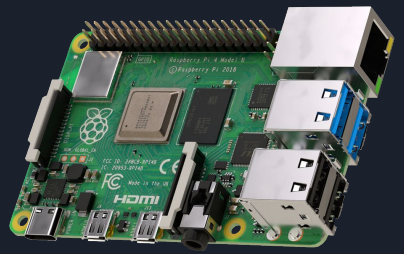
- The hardware doesn’t move on its own.
- No control unit for autonomous driving and obstacle avoidance
- Currently no software architecture that employs the use of the components.

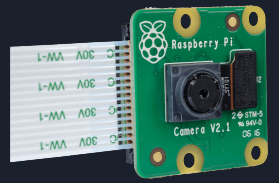


Solution Statement

Our plan is to implement three major components:

- Autonomous Movement module
- Obstacle Avoidance module
- Affordable Control Unit





Implemented with:

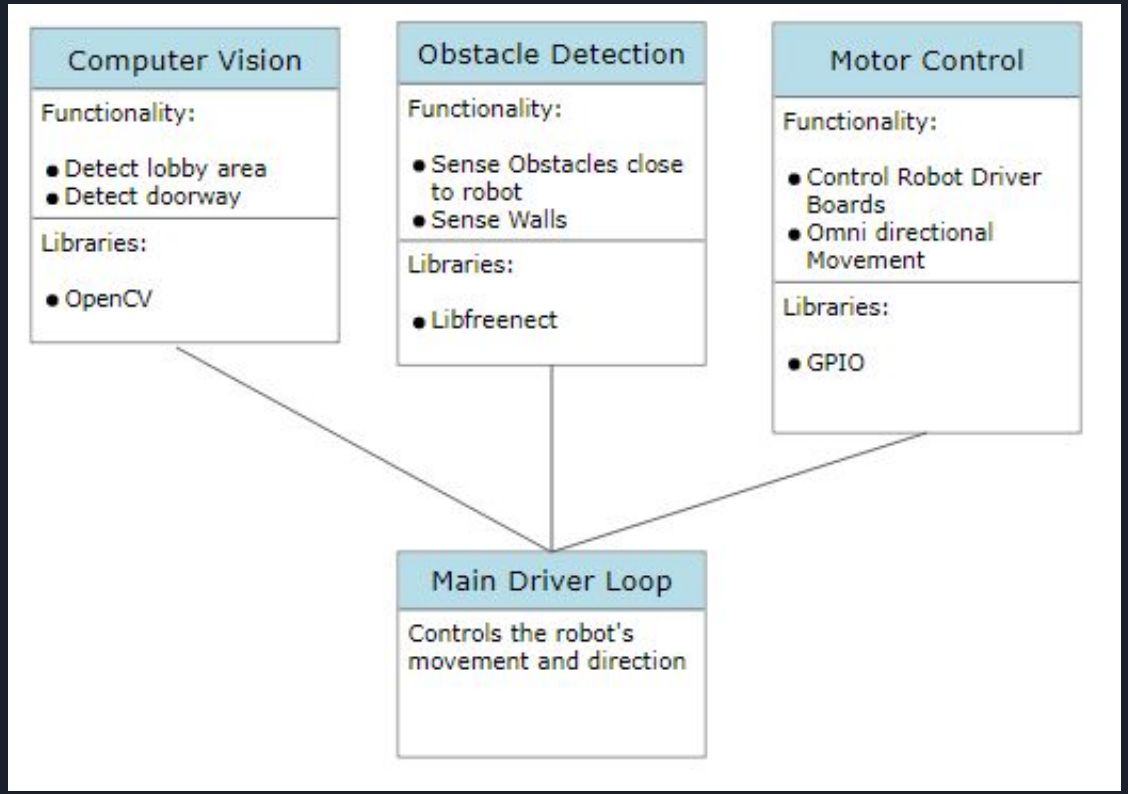
- Robot uses a raspberry pi as the central computer and controller.
  - Controls motors
- Kinect sensor for detecting obstacles
  - Tells the pi when it should avoid an object
- Small camera module for end of hallway detection
  - Image classification/recognition via deep learning

Requirements

- Autonomously move down the long hallway of the second floor engineering building
  - Move at the average human walking speed
- Avoid any obstacles in the way
  - These obstacles may be random
  - 1 to 3 meters away
  - At least ½ meter tall
- Detect when the robot has reached the end of the hallway
- Be able to turn around and come back
- Detect when it has returned the starting point and stop moving

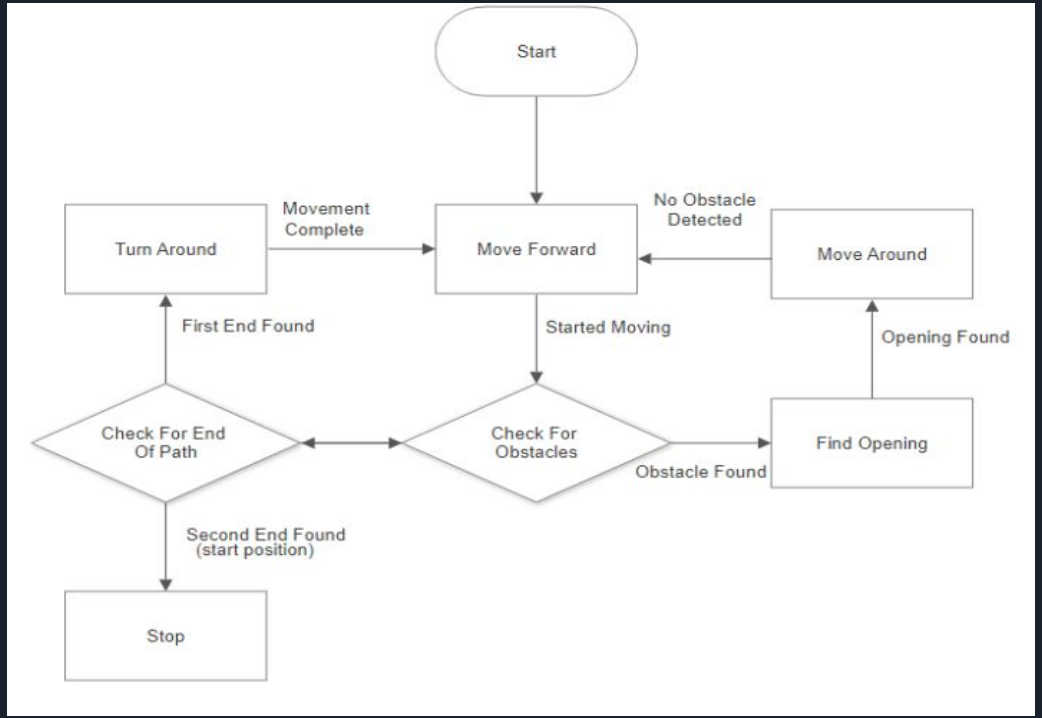
Architecture Overview

- Computer Vision:
  - OpenCV
  - Allow camera access
  - Deep CNN
- Obstacle Detection:
  - Libfreenect
  - Allow Kinect Sensor access
  - Object detection
- Motor Control
  - GPIO
  - Allows access to motors
  - Robot movement



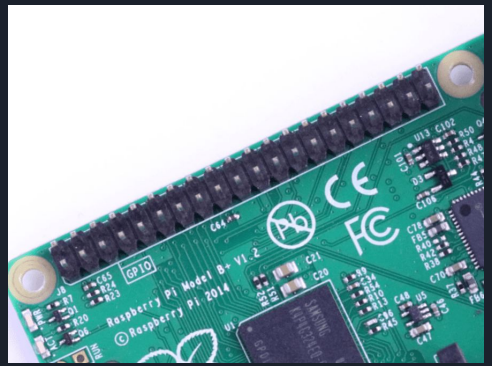
Implementation Overview

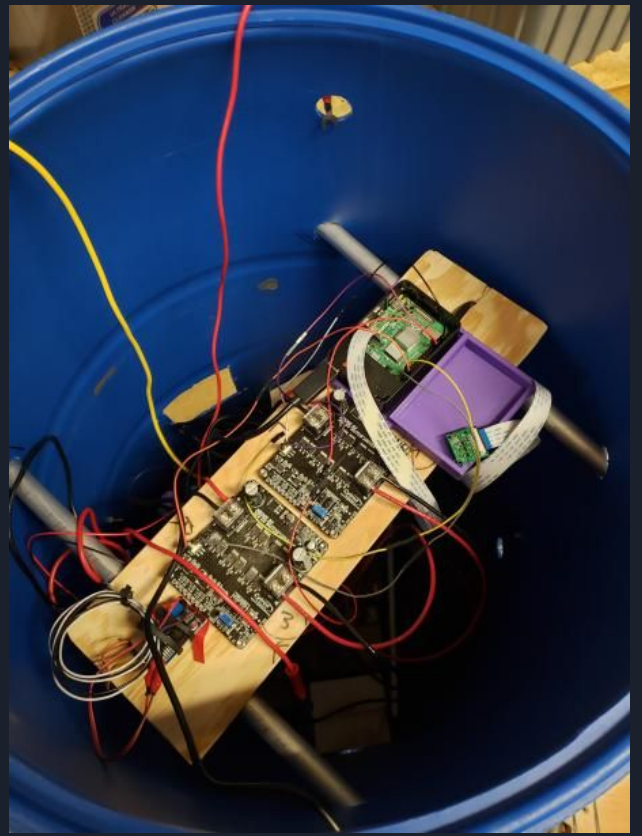
- One main loop
  - Does constant checking for obstacles and end of hallway conditions
- Checks for two end of hallway conditions
- Checks for obstacles in path



Prototype Review - Control Unit

- Hardware connection between the Raspberry Pi's GPIO pins and the motor driver pins
- Provides interface that may be called upon in the program:
  - move_forward()
  - move_backward()
  - turn_left()
  - turn_right()
  - stop()





Prototype Review - Obstacle Detection

- The kinect sensor is opened and the depth images are taken and processed
- The side in which an obstacle lies on will be determined and rotate the robot in the opposite way
- Finding an opening resulting in the robot continuing forward down its path

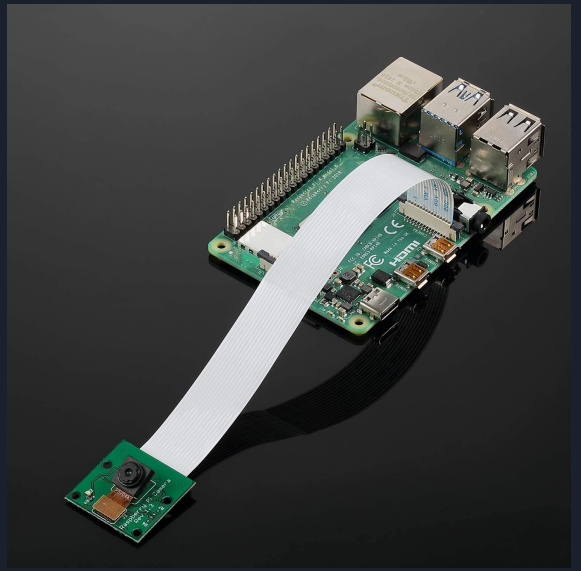
Prototype Review - Hallway Detection

- The images from the Raspberry Pi Camera are received and are used to predict which end of the hall the robot is at using the deep learning model.
- In order to account for false positives we check that the camera detects the same end of the hall multiple times consecutively.
- The first end detection results in the robot turning around, and the second ends the program.

```
rpi@rpi:~/ThirtyGallon$ python3 main.py
======================== PROGRAM STARTING ========================
Trying To Open Kinect...
Kinect Opened
Moving Forward
Checking for Obstacle
Obstacle Found, Waiting 3 seconds --------------------------------

Obstacle On Left Side|XXXXXXXXXXXXXXXXXX
>>>>> Turning Right >>>>>

Opening Found, Continuing ----------------------------------------
Moving Forward
Checking for Obstacle
Result = LOBBY     Lobby +++: 1  |  Door +++: 0
Result = LOBBY     Lobby +++: 2  |  Door +++: 0
Result = LOBBY     Lobby +++: 3  |  Door +++: 0

END OF HALLWAY - LOBBY
TURNING AROUND

Obstacle Found, Waiting 3 seconds --------------------------------

XXXXXXXXXXXXXXXXXX|Obstacle On Right Side
<<<<< Turning Left <<<<<

Opening Found, Continuing ----------------------------------------
Moving Forward
Checking for Obstacle
Result = DOOR      Lobby +++: 0  |  Door +++: 1
Result = DOOR      Lobby +++: 0  |  Door +++: 2
Result = DOOR      Lobby +++: 0  |  Door +++: 3

END OF HALLWAY - DOOR
STOPPING PROGRAM
======================== PROGRAM ENDED ========================
```

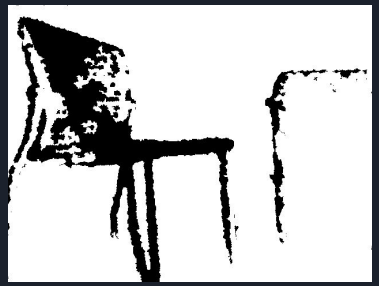
Challenges and Resolutions

- Detecting the end of the robot's path
  - Implementing a small camera
  - Using AI to model and recognize the end of the path
- Electrical issues with robot
  - Getting the robot re-built with more sound architecture
  - Reengineer power module and drivetrain for the robot



Testing Plan

- Unit and Integration testing is used to ensure full functionality of the thirty-gallon robot.
- Computer Vision is integrated with Obstacle Avoidance.
- Hardware Stress Test for potential overload
- Obstacle Avoidance, and will scan for appropriate obstacles and possible end of the hallway.
- Qualitative testing has been conducted to test for end-user usability.



Implementation Timeline

Thirty-Gallon Robot

Task	Jan 2 - 20	Jan 22 - 29	Feb 1 -10	Feb 13-20	Feb 26 - Mar 1	Mar 3 - 14	Mar 20 - 30	Apr 1 - 7	Apr 10- 18	Apr 18 - 25	Apr 26 - May 5
1 *Xbox Kinect: Inital Software Design*											
2 *Hardware Stress Test*											
3 *Relay Replacement*											
4 *Code Modularization*											
4.1 *Azure DevOps*											
5 *Kinect Sensor Installation*											
6 *Obstacle Avoidance Software v. 2*											
7 *End of the Hallway Implementation*											
7.1 *Machine Learning Implementation*											
8 *Obstacle Avoidance Stress Test 1*											
9 *End of the Hallway Test 1*											
10 *Kinect software update*											
10.1 *Sunlight adjustment for kinect*											
10.2 *Platofrm Installation for Kinect*											
11 *Full Hallway Rundown*											
12 *Final Software Release*											

Legend: Jon, Ulugbek, Brandon, Fernando, Team

Future Work

- Kinect infrared sensor is impacted by sunlight
  - Manage infrared sensor to avoid sunlight exposure

- Robot localization needs to be implemented
  - Future teams need to have a module for robot to localize itself in indoor environment

Conclusion

- Cheaper implementation of robotics can open learning opportunities
- Dr. Leverington is looking for an initial software architecture as proof of concept for touring robot
- Team Poseidon Wayfinding has implemented obstacle avoidance and autonomous movement
- Raspberry Pi module and integration of machine learning can enhance performance
    - Future teams and stakeholders can further develop the software architecture

THANK YOU