Single Controller, Multi-Robot System

Project Report for Senior Project Design

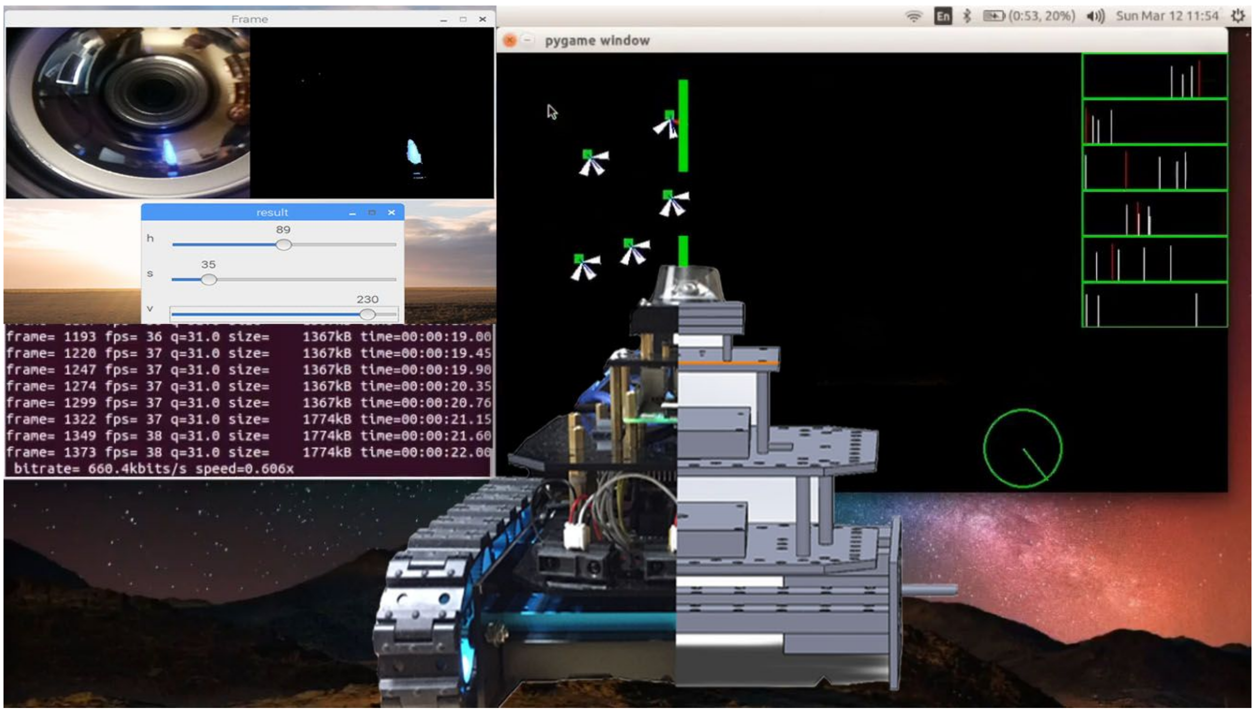


Figure 1 - Overview of the Project

Team name : Team 17

Jeffrey Berhow, Electrical Engineering
Anurag Vaddi Reddy, Electrical Engineering
Carla Contreras, Electrical Engineering
Yuji Dornberg, Computer Engineering

Mentored by:

Dr. Swindlehurst, EECS

Dr. Krichmar, Cognitive Sciences

Dr. Green, EEC

Executive Summary

This project is focused on semi-autonomous operation of mobile ground robots using a single point of control. While remote control of an individual robot is relatively easy, and can be performed with simple human intervention, control of a group of robots is much more difficult and requires some kind of machine intervention. This problem will be approached by first constructing the robots, and then simulating and developing the algorithms to allow them to avoid obstacles. The main goals of the project for senior design is to have the mobile ground robots all move in unison in the direction the controller chooses. We also want each individual robot to avoid obstacles. The obstacle avoidance was accomplished using IR sensors. The controller communicates the direction through wifi. Two stepper motors and three IR sensors are wired to an Arduino Mega. The Arduino Mega is connected to a Raspberry Pi 3 through the USB port. They communicate through serial communication.

Table of Contents

Introduction

The project will involve developing methods for helping a group of robots to navigate through obstacles, such as through narrow passages or around objects in their path, while avoiding collisions and when possible maintaining certain formations. There are a few potential markets for our project. In the commercial market an application of our robots can be as children's toys. We would have to scale our robots down for this application, but there has been a huge demand for robots in the toy industry. In the military market there are few application for our robots. Swarms can be used in military to form an autonomous army. They can be used for reconnaissance and tracking.

It is a project that will add data, solutions, and general experience to a small but growing field with endless applications. There have been other implementations of our proposed project, but they so few and far between that any solution we develop will very likely be entirely unique. As a result, even if our final product is not the most efficient, it will still provide important data for others as they develop designs for related projects.

Background

Prior understanding of engineering topics should be focused on simple data structures and algorithms, electronic circuit design, kinematics, and network analysis. Although the end goal is easy to understand, most of the difficulty comes in terminology and exposure to the concepts in the following disciplines.

- Data Structure and Algorithm knowledge for understanding the ROS graph system as well as the tuning of the obstacle avoidance algorithms.
- Electronic Circuit design is helpful for understanding the components needed for filtering and regulation.
- Kinematics knowledge will help with understanding the experiments needed to test momentum drift, rotation drift, etc.
- Network analysis for the general connecting of multiple ICs and components into one robot.

The majority of the research in the field of swarm robotics deals with autonomy and the absence of a controller or operator. This means that information regarding group robotics isn't as easy to come across, although many of the advancements of swarm robotics can be applied to group robotics. These include: inertial calculations for robots in general, algorithms for object avoidance and communication protocols between robots.

A very important factor that came up, and is still a huge factor for us, is the fact that if certain pieces of the robot go extinct, the robot would have to be redesigned. This includes the Raspberry Pi, the camera and panoramic camera lens, and parts of the chassis. The Pi and the chassis parts would only set us back a bit, if they weren't made anymore. The camera and camera lens, on the other hand, would be very detrimental as there is currently only one supplier for each. The lens actually may go out of existence in the near future as it is a lens that is made specifically for an older phone that is not in production anymore.

Project design

This project produced a swarm of drones, capable of obstacle avoidance, directed movements, and limited swarm positioning capabilities. Each drone is capable of networked communication with the swarm, swarm positioning and ranging using a camera with a 360 degree lens, and obstacle avoidance using IR range sensors. Raspberry pi's conduct image processing on data from the camera and 360 lense while simultaneously handling data from a magnetometer and the controller to assign priority to commands. These prioritized commands are then sent to an Arduino which is connected to infrared sensors for instantaneous obstacle avoidance and the drivers for the stepper motors for control. The Arduino decides based on inputs from the IR sensors whether or not to listen to the commands from the pi or to avoid an immediate collision.

The key features of this system are:

1. **Chassis**
    a. Modular Tracked Base
    b. Primary Chassis
    c. Camera Mount
2. **Multi Robot Communication**
    a. ROS Communication
    b. Serial Communication
3. **Power System**
    a. Batteries
    b. Buck Converter

c. Motor Driver

4. **Sensors**
    a. Camera
    b. 360 Degree Mirror
    c. IR Sensor
    d. Magnetometer
5. **Processing Center**
    a. Raspberry pi 3
        i. Image Processing
        ii. Control Processing
    b. Arduino
        i. Sensor Processing

1. Chassis

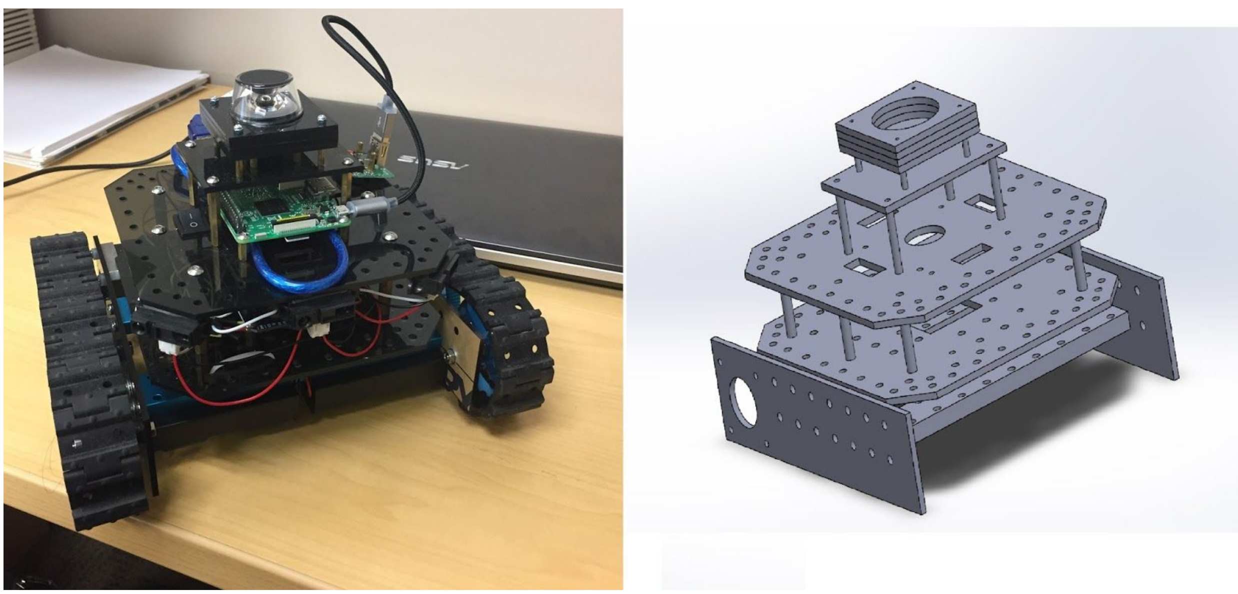


Figure 2 - Robot Chassis

a. Modular Tracked Base

The main purpose of the modular tracked base is to house the primary locomotive hardware, such as the stepper motors, the tracks and track tensioners, and the power source for the drone. The modular tracked base, as its name implies, was designed to be modular in design to afford the maximum amount of flexibility with regards to its own configurations as well as a continually evolving primary chassis. Motors can be mounted on the front or rear of the drone, or offset for better weight distribution.

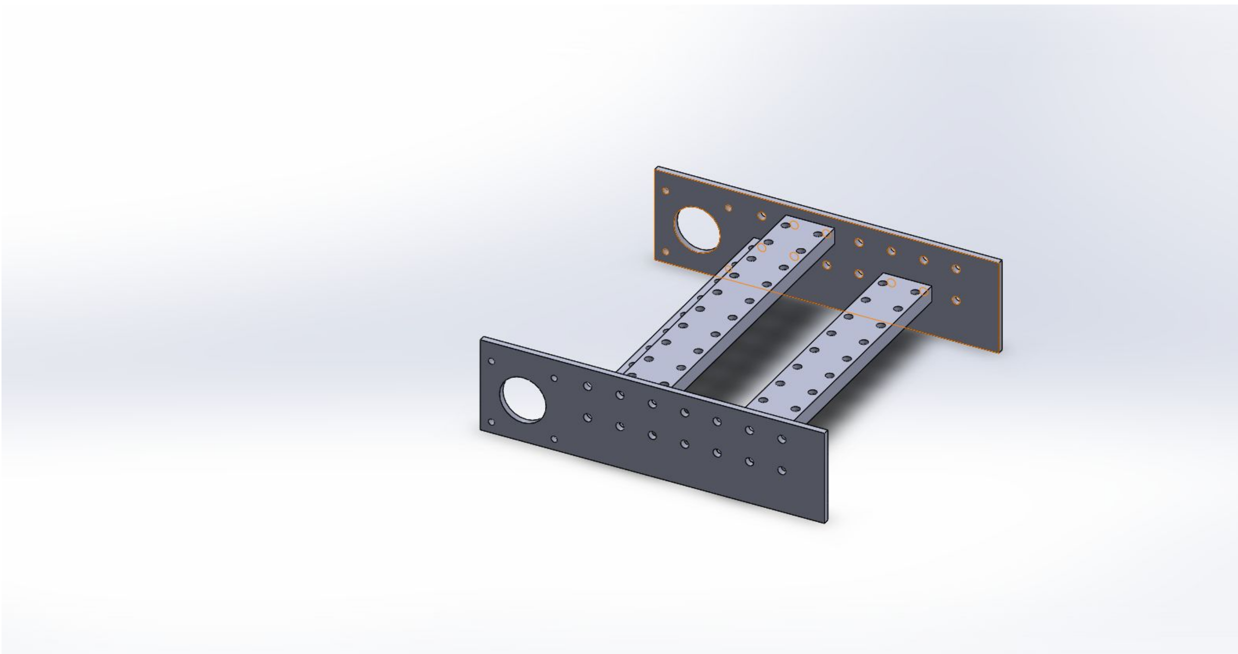


Figure 3 - Tracked Base

The tracked base is composed of two acrylic mounting plates held together by multiple aluminum support bars. On the mounting plates the motors and track tensioners are attached. The primary chassis is mounted to the tracked base via a central aluminum support bar. Acrylic was chosen for the plates because it is cost efficient and easy to customize using solidworks and a laser cutter. The aluminum support bars were purchased from a robotics company for their numerous attachment points.

b. Primary Chassis

The purpose of the primary chassis is to provide a centralized location for all the major components of the drone to mount. The primary chassis provides mounting points for its primary processing units, the Raspberry pi and Arduino, the power regulation system, the camera mount, and the modular tracked base.

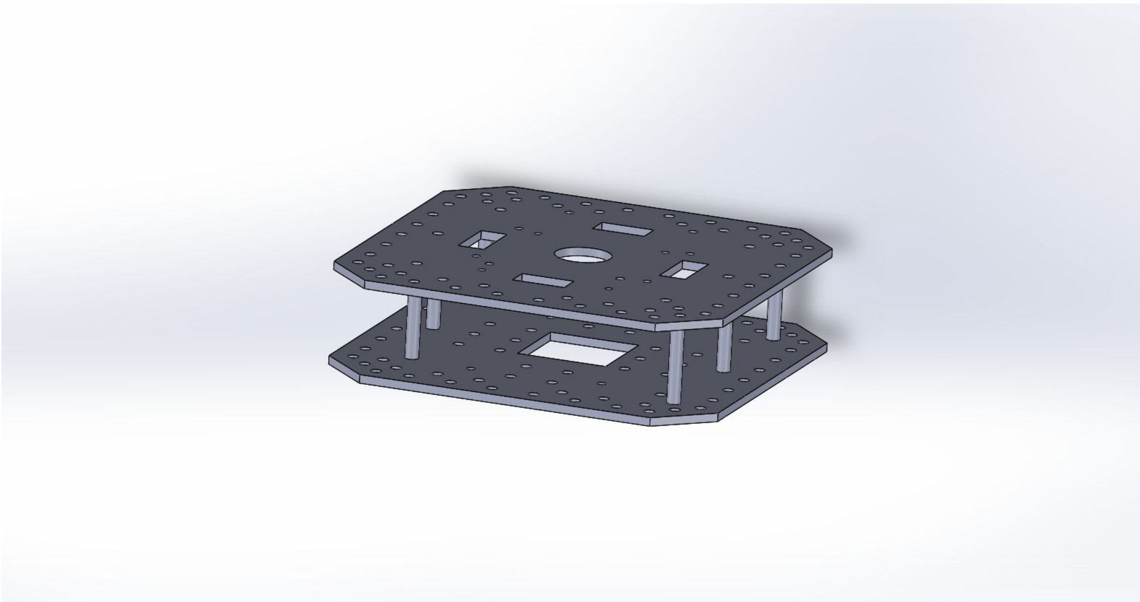


Figure 4 - Modular Base

The chassis is constructed from a top and bottom plate, made from laser cut acrylic, and a series of brass coupling nuts to act as the frame for the plates. As with the modular tracked base, acrylic was chosen for the plates due to its cost effectiveness and ease of customization. The brass coupling nuts were chosen for their ease of use, overall simplicity, and cost effectiveness.

c. Camera Mount

The camera mount was designed to mount the 360 lense to the camera and to mount the camera system to the drone in a secure manner. As the camera is essentially just an exposed circuit board with an unprotected imaging sensor, mounting the 360 lense to the camera so that it would be properly centered required some pretty exact measurements which we were able to generate using laser cut acrylic plates. After mounting the lense to the camera, the camera also had to be mounted to the chassis to protect it from being damaged by the other components on the drone. All of this was done with a single modular system for ease of installation and modification.

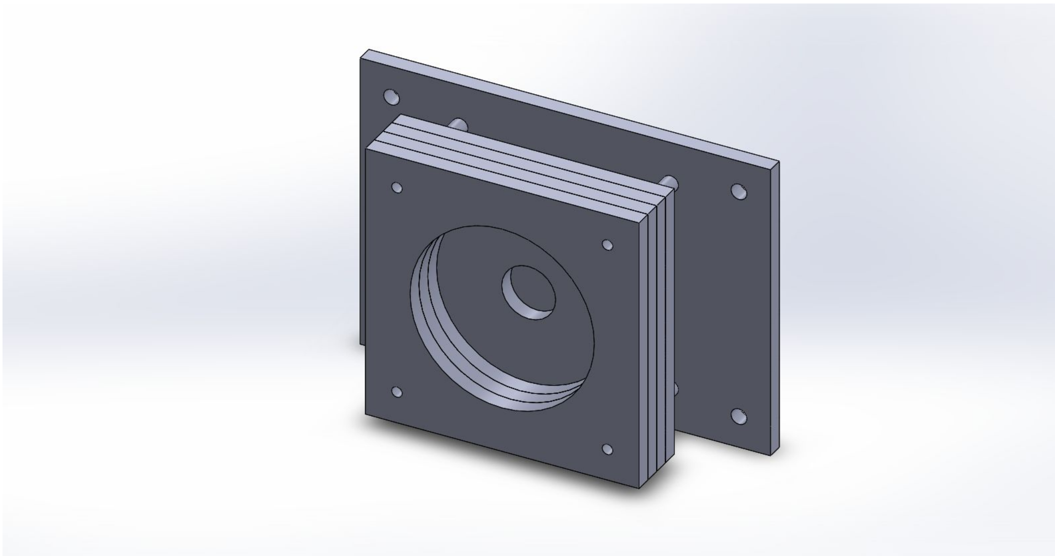


Figure 5 - Camera Holder

In this case acrylic plates were used not just for their ease of customization but the high degree of precision needed to center the lense on the camera and properly secure both components. Brass coupling nuts were once again chosen for their ease of use, overall simplicity, and cost effectiveness.

2. Multi Robot Communication

The communication system should handle all communications, from the individual Raspberry Pis communicating with the Arduinos to the Raspberry Pis communicating with each other as well as the controller and vice versa.

a. ROS Communication

Our communication was based mainly on the Robot Operating System's (ROS) decoupled graph structure and chosen for that reason. Decoupling of components and code allows for easy flexibility and scalability; scalability being a huge driving point for group robotics. In the Figure 6 below, this can be seen as multiple nodes and edges. The nodes denote either topics that will hold messages to be delivered to all robots, or nodes that specifically subscribe or publish to these topics. ROS handles most of the difficulties of communication between robots and controller, including TCP/IP and multithreading. This allows for us to focus more on algorithm implementation than reinventing the wheel that is wireless communication.

The joy_node is responsible for gathering joystick values and publishing them on the topic joy. The controller node, then, subscribes to that topic and changes the joystick message into a usable movement message and publishes that on the cmd_vel topic. cmd_vel is a topic that has Twist message containing a linear movement and an angular movement.

The Raspberry pi listener node subscribes to the cmd_vel topic, processes the linear and angular input and sends it out via Serial to the Arduino to activate the motors. This graph resembles the graph below, but is unrenderable on the Pi, at the moment.

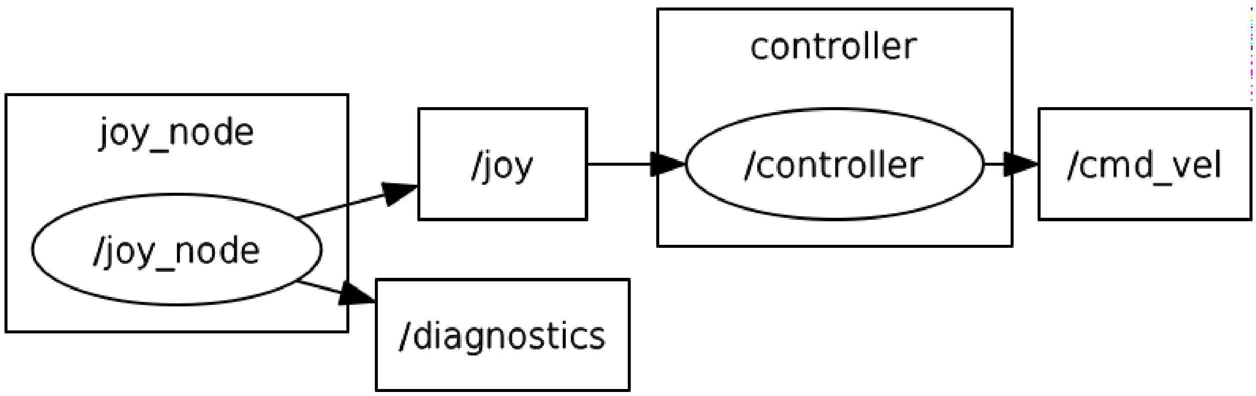


Figure 6 - Graph Structure of the ROS Communication

b. <u>Serial Communication</u>

Information for the movement of motors is sent from the Raspberry pi to the Arduino using the Serial protocol. This is accomplished via a USB cable that connects to the Arduino's UART. The message contains a start byte, end byte and five bytes in between much like in Figure 7. The intermediate bytes contain information about the rotation and forward movement/speed of the robot. Two bytes are allocated for current and required rotation in a positive and negative sense. One byte is used for speed or, in the case of no rotation, forward movement.

Serial communication was the most obvious choice to make, as it allows powering of the Arduino via the Raspberry pi as well as sending data.

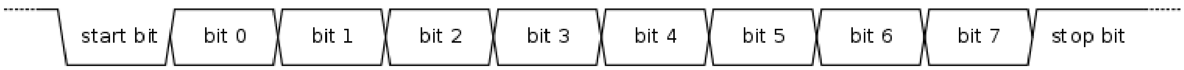


Figure 7 - Example of a Signal sent over UART

3. Power System

The Power System should handle the power distribution to all components as well as to the prime movers of the robot.

a. Batteries

Lithium Ion batteries were a choice made early on from experience with them in our everyday lives. The particular amount chosen was an application driven choice, however. Each lithium ion battery charges to approximately 4V. The stepper motor driver has a minimum voltage of 12V. So we chose to use 4 batteries so that as the voltage of each battery drops the total voltage battery voltage will be greater than 12V. Lithium ion batteries can also draw large currents without overheating. Since we can draw over 1 amp per motor and up to 1.5A from the Pi, lithium ion batteries are a sensible choice. Lastly, lithium ion batteries have a high capacity relative to other batteries. Each of our lithium ion batteries range from 2.5-3.4 Ah of energy. This means we can run the motors and the Pi at max current for an hour.

b. Buck Converter

The Buck converter was needed to take the high voltage of the batteries and lower it for powering of the Raspberry pi. The choice of the buck converter came down to price point and operation over a wide range of input voltages. The input voltage of our buck converter can range from 10V-35V. It can output up to 2A. The Arduino draws power from the Pi, which means the 2A needs to be enough for both the Pi and the Arduino.

c. Motor Driver

The motor driver we used is a Pololu A4988 stepper motor driver. It has a minimum voltage of 8V. There is a higher current stepper motor driver with a minimum voltage of 12V so we decided to design for the higher current driver to futureproof the drone. The A4988 has a max current draw of 2A. The maximum current draw can be changed with an onboard potentiometer. We set the maximum current to approximately 1A to save energy.

4. Sensors

The Camera system is composed of a Raspberry pi camera pointing directly up toward a convex mirror. This should be able to have a complete 360 degree view of the environment, and deliver sufficient detail at the required resolution.

a. Camera

The Raspberry pi has a dedicated camera input on the board that uses none of the GPIO pins, leaving them open for use by other sensors. The Raspberry pi foundation sells a compatible camera for the input and this was the choice that was made. It is easily installed and integrates perfectly with OpenCV libraries that also integrate well with ROS.

b. 360 Degree Mirror

Another device that was almost forced upon us was the 360 degree camera. There just isn't a huge market for this type of modification and the only one we came across was made by Kogeto. This was found from a website that gave a description of how to get a panoramic view of surroundings using a Raspberry pi and was indispensable for setup and use.

c. IR Sensor

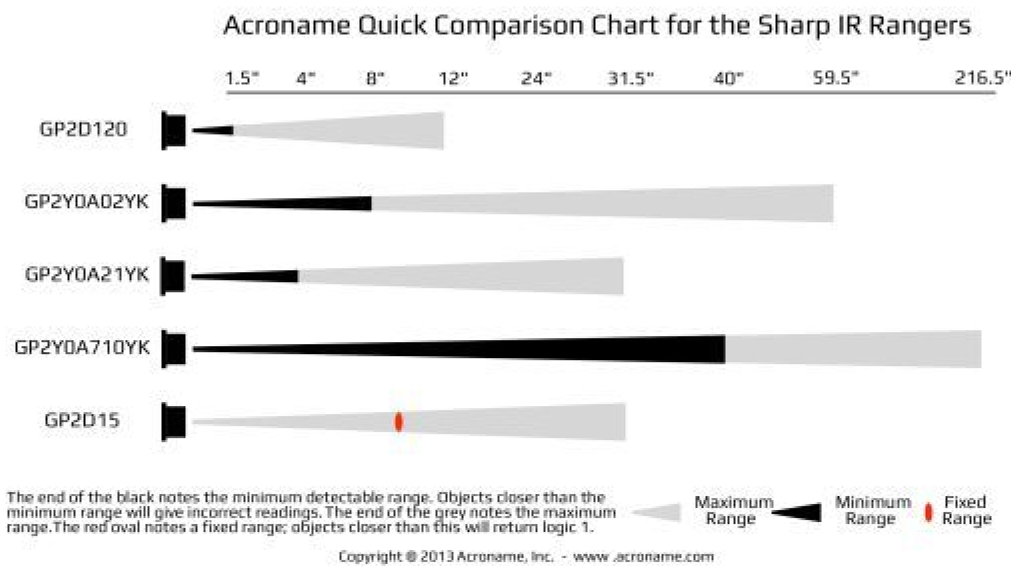


Figure 8 - IR Sensor Comparison

The third sensor in the chart of IR sensors above is the sensor that our robot uses. It runs off the 5V from the Arduino, and is measured using the Arduino's analog input pins. The voltage to distance ratio is nonlinear, but we can get a linear relationship if we take the inverse of the distance as a ratio to the voltage.

d. <u>Magnetometer</u>

The purpose of the magnetometer is to allow the drones to all orient to the correct direction whilst autonomously avoiding obstacles. It was originally added as a temporary fix to test swarm behavior before full implementation of the camera and 360 lens. It will continue to be useful for orienting the swarm to the controller without generating confusion for the operator as well as allowing for the controller to eventually be replaced by beacons that the swarm will autonomously seek out without operator input.

5. Processing Center:

The information processing center is comprised of two primary structures: the conscious, Raspberry pi, and the subconscious, Arduino. The conscious processes high level decisions like determining where the swarm is and if the drone needs to rejoin it and if the drone is traveling in the correct direction. The subconscious controls mobility and avoids instantaneous danger via collisions with objects.

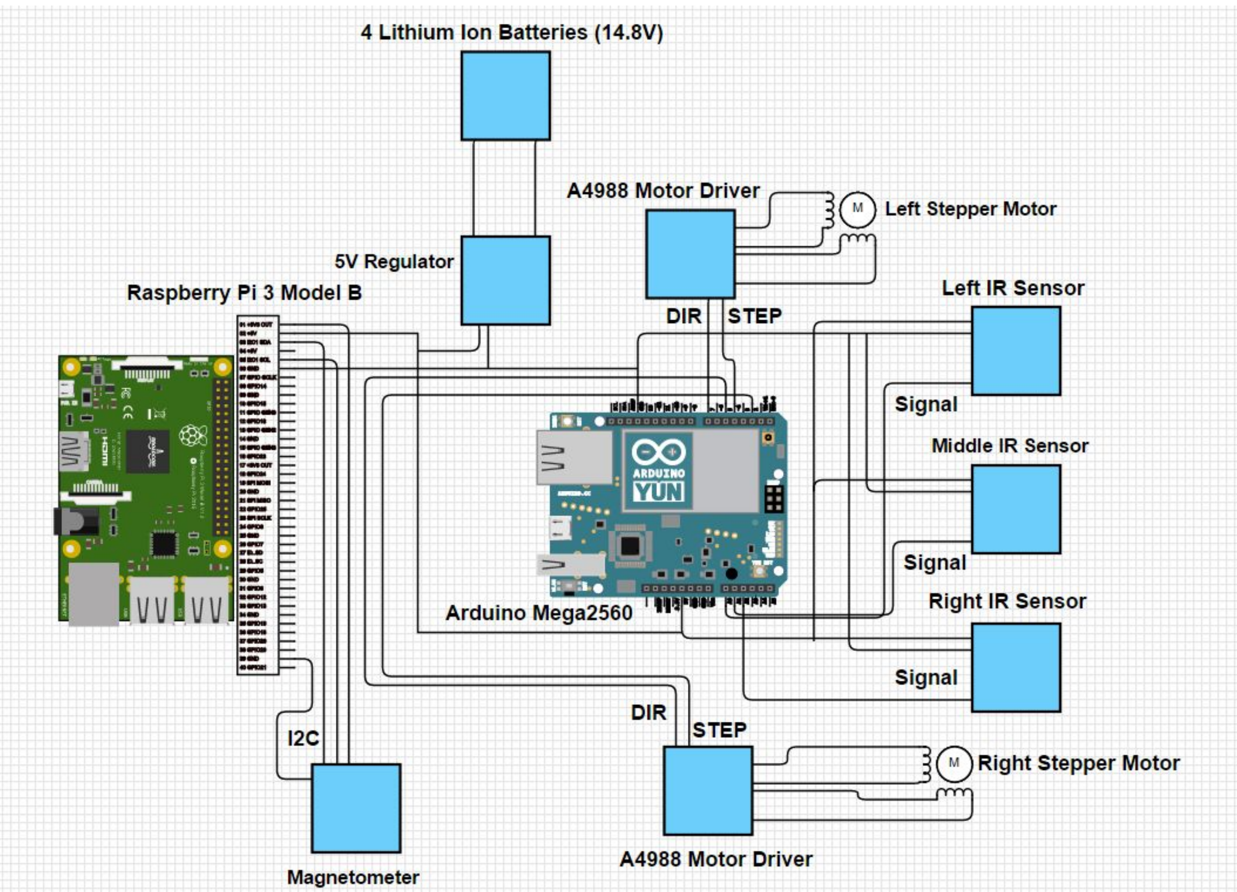


Figure 9 - Wiring Diagram

a. <u>Raspberry Pi 3</u>

The Raspberry Pi subsystem contains the Raspberry Pi 3 and all programs and algorithms associated with the high level decision making. The Pi powers the Arduino through the USB port as well as communicates with it using serial communication.

i. Image Processing

The Raspberry pi runs OpenCV, a vast library used for image processing for computer vision, in order to detect the location of the swarm and how far the drone is from the other drones. This is accomplished using a light beacon, isolating it from its surroundings and measuring its change in area.

ii. Control Processing

While processing this data the pi also takes in controller commands via wifi antenna and decodes the commands into messages published to the ROS topic /cmd_vel. More information can be found in the ROS communication section.

b. Arduino

The Arduino subsystem contains an Arduino Mega 2560 and all algorithms for the primal decision making and motility.

i. Sensor Processing

All sensor testing is handled at the lowest level on the Arduino. This includes changing data from the Pi into movement, as well as using the IR sensor values to override the incoming Pi data. This is akin to a nervous system acting over the commands of the brain.

Characterization

Algorithm Testing

All algorithms were tested using a custom Python simulator created using the Pygame library. The simulator allowed for ideal situations to be tested before the migration of code to the physical robots. Heuristics were mainly used for control adjustments to simulate the reaction of the robots to a controller as the turnaround time of testing was so fast. There was no need to design the algorithms before trying them out, as that would have delayed the prototyping process. Once the robots behaved as expected, and our physical robots were completely built, we moved the code over and continued with the fine tuning of the algorithms on the robots themselves.

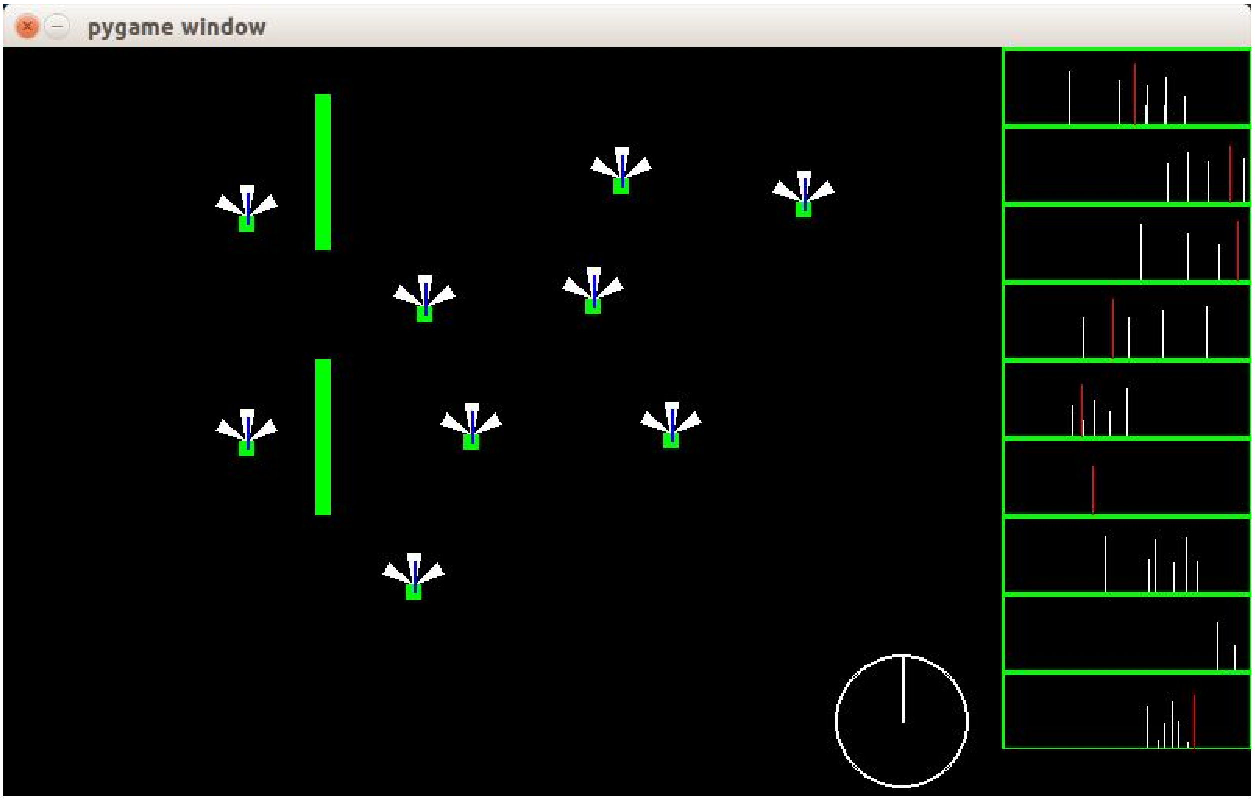


Figure 10 - Simulation Environment for Multi Robots

Camera Testing

Camera color detection tests were handled on a camera by camera basis using an in-house calibration technique. The technique utilizes the splitting of camera data into three channels: Hue, Saturation, and Value (which is the intensity of the image at the respective pixel). After the channels were isolated a slider allowed for honing in on the object to be detected. This is a heuristic process, as all objects will be subject to different lighting and hence different saturation values. The image below shows the real time camera data, the processed camera data dependent on the channel values, and the slider used to change those channel values.

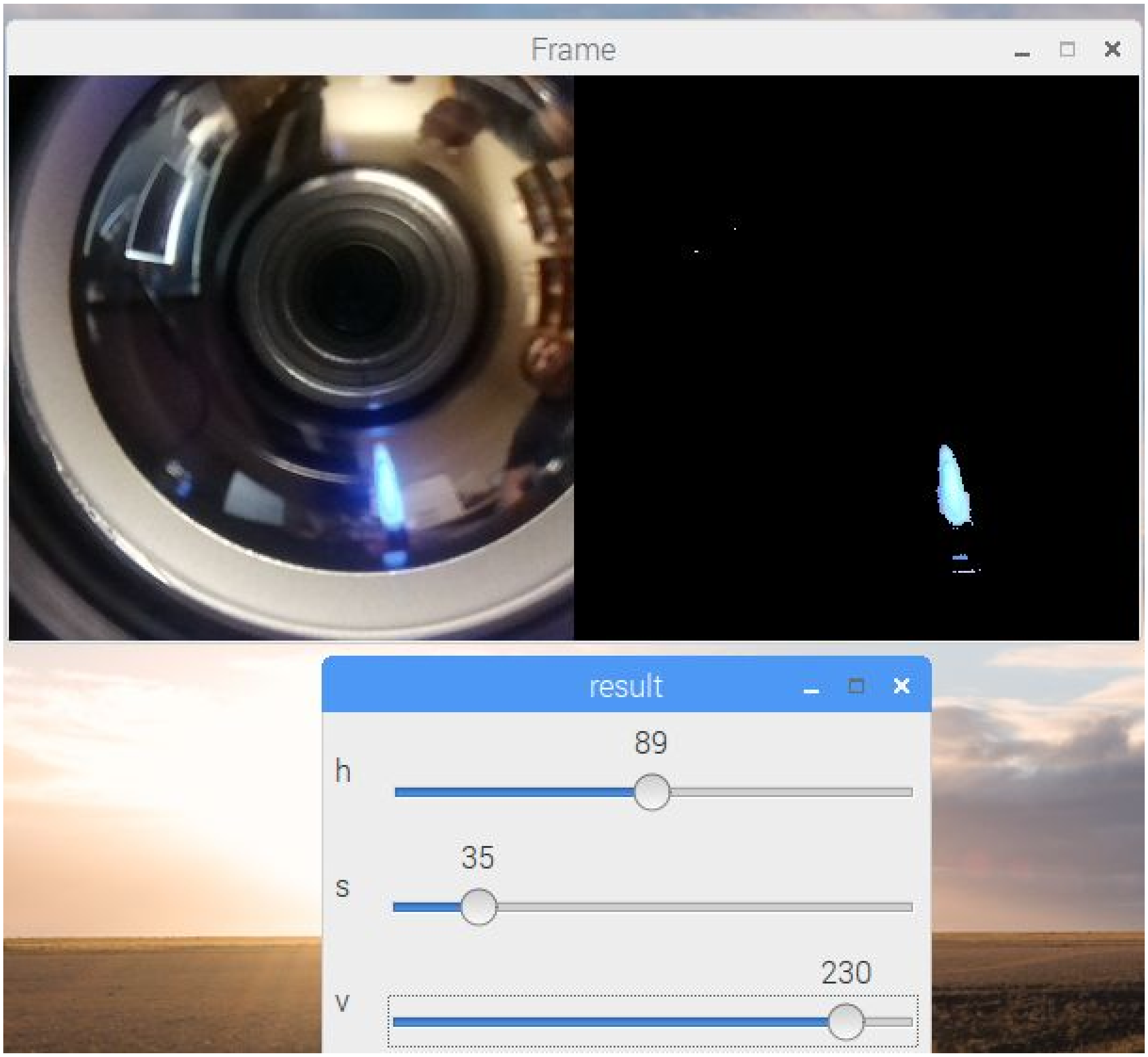


Figure 11 - Camera Calibration Process: Counter-clockwise from upper left a) real time camera data b) processed data c) channel value selection sliders

Sensor Testing

Infrared sensors were tested using a simple program designed to continuously output values from the sensor as it sat in a static position. Objects were then placed at different distances from the sensor to measure sensitivity as well as on either side of the sensors beam to calculate beam-width. Once a general idea of the sensors capabilities had been developed they could be mounted to the drones in such a way as to provide the maximum coverage on the front of the drone. After mounting was complete a program to make the drone continuously move forward, turning only when the sensors detected an obstacle, was run on the drone and it was left to navigate its way through a small room while avoiding obstacles.

Chassis Construction:

The chassis was constructed in solidworks from blueprints laid out on paper and measurements of components taken from schematics found online. Constructing all the parts in solidworks first allowed us to conduct fit tests before ever even purchasing planned components or cutting out parts using the laser cutter. Once the chassis was constructed in solidworks and major components were installed without conflict the parts could be cut out and components ordered for the physical fit test. Construction of the actual drones acted as a final fit test and after some light stress testing, checking for flexing and deformation of materials, the drones were ready to move on to other testing.

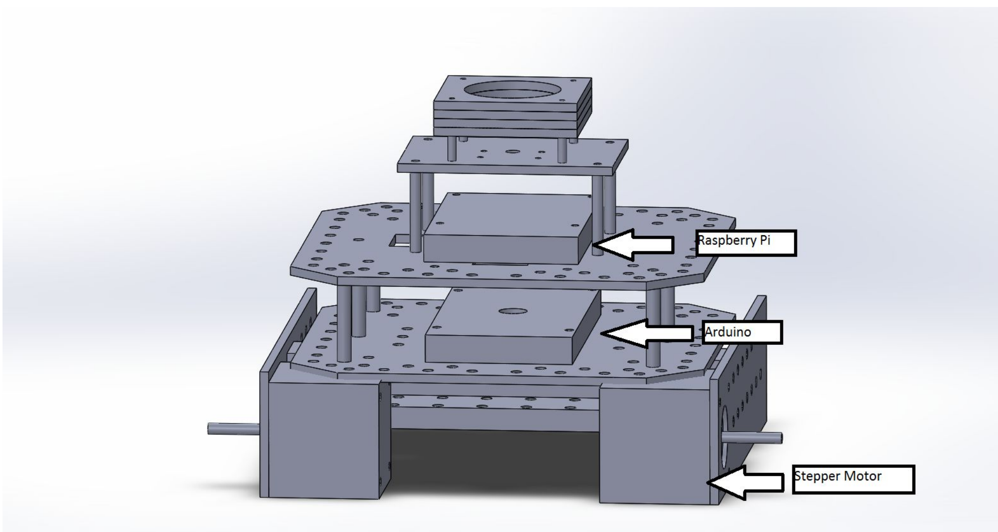


Figure 12 - Chassis Design

Concluding remarks

Final assessments

Most of our design for our robot matches our original goals such as obstacle avoidance, 360 degree awareness, and quick communication between Pis and between the Pi and the Arduino. There are a few issues that need be addressed moving forward. When the simulation for the drones was originally created, IR sensor were envisioned to have wide cone detection. However, in reality the cones are actually very narrow and the edges are not entirely accurate. This results in the drones responding in unexpected ways for obstacles such as buckets. We must update the simulation such that the simulated IR sensors more accurately represent the real IR sensors. Another issue that needs to be resolved is the focal length of the Pi camera. The Pi camera shows a blurry video with the current placement of the 360 lens. We have seen a clearer picture in the past which indicates that the 360 lens is currently out of the focal length of the camera. Finally, the tracks on the drone currently have trouble rotating on rough carpet. When rotating the tracks see an orthogonal force that pulls it off the wheel. We do not expect the drone to operate on carpet so we are not concerned with this as of yet. But we are aware of the issue, which we will correct in the future.

Future work

Our robots have met the goal we expected of them, but we are in a good position to continue on with our work into the Spring quarter. Our ultimate goal, moving forward, is to integrate our camera to help more with robot positioning and separation. This means returning to our simulation to test new algorithms and state machines allowing proper prioritization of movement from each sensor. We would also like more smooth movement by taking advantage of the properties of vectors for speed as opposed to our current orthogonal system that either rotates or translates.

We have also researched the shrinking of our robots in case we would like to move these to market. Raspberry pi now offers a very small device that has wifi on it, and Arduino also has very small devices with just as many pin out options as the Arduino we are currently using. These would help with the prototyping of our robots but eventually we would need to move to more dedicated processors and microcontrollers, such as DSPs and Atmel chips.

Also movement to other chassis materials would have to be researched to make sure design is robust enough to pass strain, stress, temperature, etc. tests. Our motors would have to be scaled down as well as possibly having parts custom made for the smaller design.

Broader impacts

This project, through its expansion and addition of data and solutions to the field of swarm robotics, has the ability to impact three major sectors of society; academia, emergency rescue services, and the military.

Due to our simplistic and cost effective design these drones will be very affordable for universities to either purchase outright or build themselves. This will make swarm robotics, and robotics in general, extremely accessible to the next generation of engineers and computer scientists as they work to develop new swarm algorithms to accomplish various tasks. Because of their modularity and the open source nature of our project research teams will even be able to use them as templates to build customized drones to suit their individualized needs.

Within the emergency response field professionals are at times thrown into very dangerous situations. During these times it is sometimes preferable to send in a robot that can be easily replaced and is not as susceptible to dangerous environments. In situations where collapsed buildings and mountainsides might bury countless lives a swarm of small autonomous drones can mean the difference between life and death for not only those buried but the rescue professionals as well, as they would be provided a much more accurate picture of the situation, potential hazards, and a better idea of the victims potential locations. They can even be useful in large open expanses such as forests where there simply may not be enough humans to search for lost individuals.

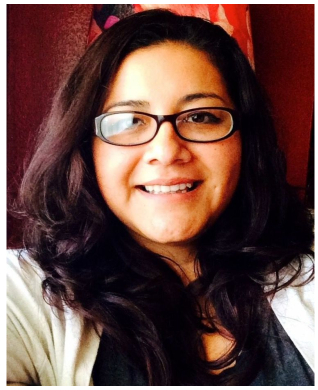
Distributed remote sensing is the main draw of our project for the department of defense. The military is always looking for ways to effectively take men off the front line and out of harm's way. Deploying drones to hostile environments can save lives by replacing teams on reconnaissance missions to detect enemy encampments, conduct area security around friendly positions to prevent enemy ambush, and even clear IED ridden supply routs and mine fields.

Team and acknowledgments

Team Lead: Jeffrey K. Berhow

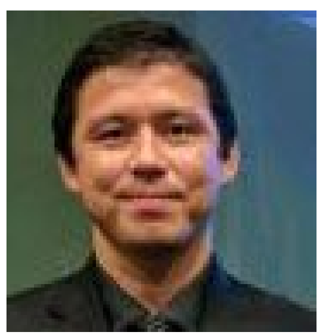
As team lead, Jeff was responsible for project vision and providing final decisions on conflicting paths of development. Jeff drove the simulation environment for the drones to test swarm algorithms, and developed the communication system that allowed for the sharing of information between the robot subsystems and controller-robot interaction.

Carla Contreras



Lead designer, Carla, was responsible for designing and developing the physical drones being operated. In addition Carla ran testing on the IR sensors to determine their exact capabilities and areas of sensing. She also tested alternative sensor options and conducted research on optimal configurations for sensors.

Yuji Dornberg



Yuji assisted with development of the drones as well as some early work on the networks that the swarm would run on. Yuji did the main designs on the chassis. Yuji was responsible for all testing and modifications on the chassis. Additionally, Yuji assisted with logistical and administrative coordination for the project.

Anurag Vaddi Reddy

Computer and electrical engineer, Anurag, was tasked with hardware power specifications. In addition, Anurag worked on the simulation and ported over the algorithms to hardware.

Mentors:

Dr. Lee Swindlehurst, Dept. of Electrical & Computer Engineering
Henry Samueli School of Engineering

Dr. Jeffrey Krichmar , Department of Cognitive Sciences
School of Social Sciences

Dr. Michael Green, Dept. of Electrical & Computer Engineering
Henry Samueli School of Engineering

Acknowledgments:

We would like to thank the following organizations for providing their support and recognition of our research project and the goals set out at the projects start. Without which this project would not be possible.

Orange County Engineering Council

Undergraduate Research Opportunities Program

UROP Multidisciplinary Design Program

A special thanks to:

Dr. Donald R. Blake

References

Various. "Universal asynchronous receiver/transmitter." 3/12/2017. Internet: https://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter

Anonymous. "Camera Module." 3/12/2017. Internet: https://www.raspberrypi.org/products/camera-module/.

Anonymous. "Building a Raspberry Pi 360 Camera." 3/12/2017. Internet: https://www.raspberrypi.org/magpi/building-a-raspberry-pi-360-camera/

Anonymous. "Sharp Infrared Ranger Comparison." 3/12/2017. Internet: https://acroname.com/articles/sharp-infrared-ranger-comparison