LookSee

small business surveillance system

Group 6

Stavros Avdella	Computer Engineering (BSCpE)
Austin Pena	Computer Engineering (BSCpE)
Tyler Wallace	Computer Engineering (BSCpE)
Jade Zsiros	Computer Engineering (BSCpE)

Project Overview

For our project, we built a robot that autonomously navigates an empty indoor building and calls a human emergency contact if an intruder is detected.

The robot is made from an RC car and uses an Nvidia Jetson as its central processor. It uses ROS, the Robot Operating System, to process sensor data, navigate, and trigger events.

Our robot will be named LookSee.

JZ

Motivations

01 We wanted to select and design a project that is challenging.

02 We needed to design a project that has many different components, so that we do not have to rely on in-person collaboration.

03 We wanted to develop skills related to our chosen fields.

04 We wanted a project we could all be excited about.

JZ

Primary Requirements

01 By the end of the semester, LookSee should be able to autonomously navigate a space similar to the average office building for at least 10 minutes without getting stuck.

02 By the end of the semester, LookSee should be able to identify a human moving around in front of its path with at least 70% accuracy.

03 When LookSee identifies a human intruder, it must call the emergency contact, and allow the contact to take over navigation of the robot.

JZ

Specifications

By the end of this project

- LookSee will navigate autonomously with Find-the-Gap navigation
- LookSee will be able to identify a human in the room through a combination of computer vision and thermal sensing
- LookSee will have a user interface that can be accessed through a web page
- LookSee's user interface will allow a human operator to drive the robot
- LookSee will be able to video call an emergency contact using Skype if a human is detected.
- LookSee will use ROS as its software backbone.

JZ

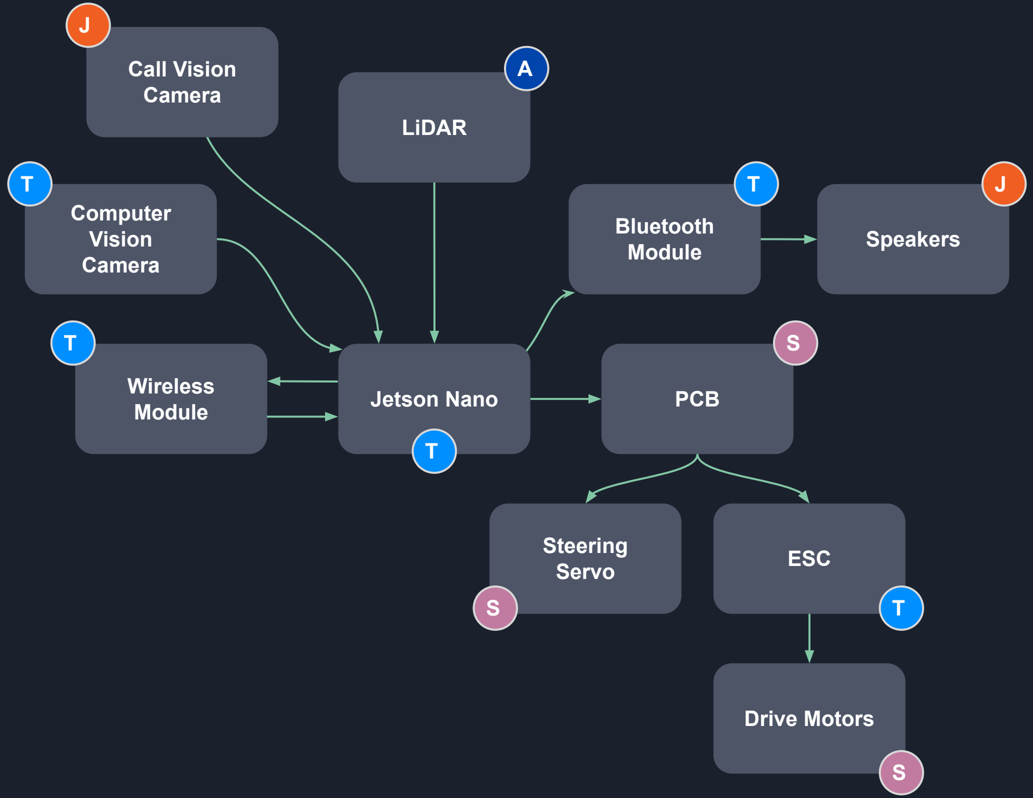
Target Audience

The target market for this robot is owners of small to medium-sized businesses. LookSee is designed to be much cheaper than installing a CCTV system or hiring a night guard to patrol or any other surveillance equipment, so it will appeal to businesses looking to save money on insurance and wages, and avoid risks related to human error or casualty related to health risks undertaken by a night guard.

As this is a small, low-budget platform, its use case does not include government or defense contractor facilities, or anything else of that nature.

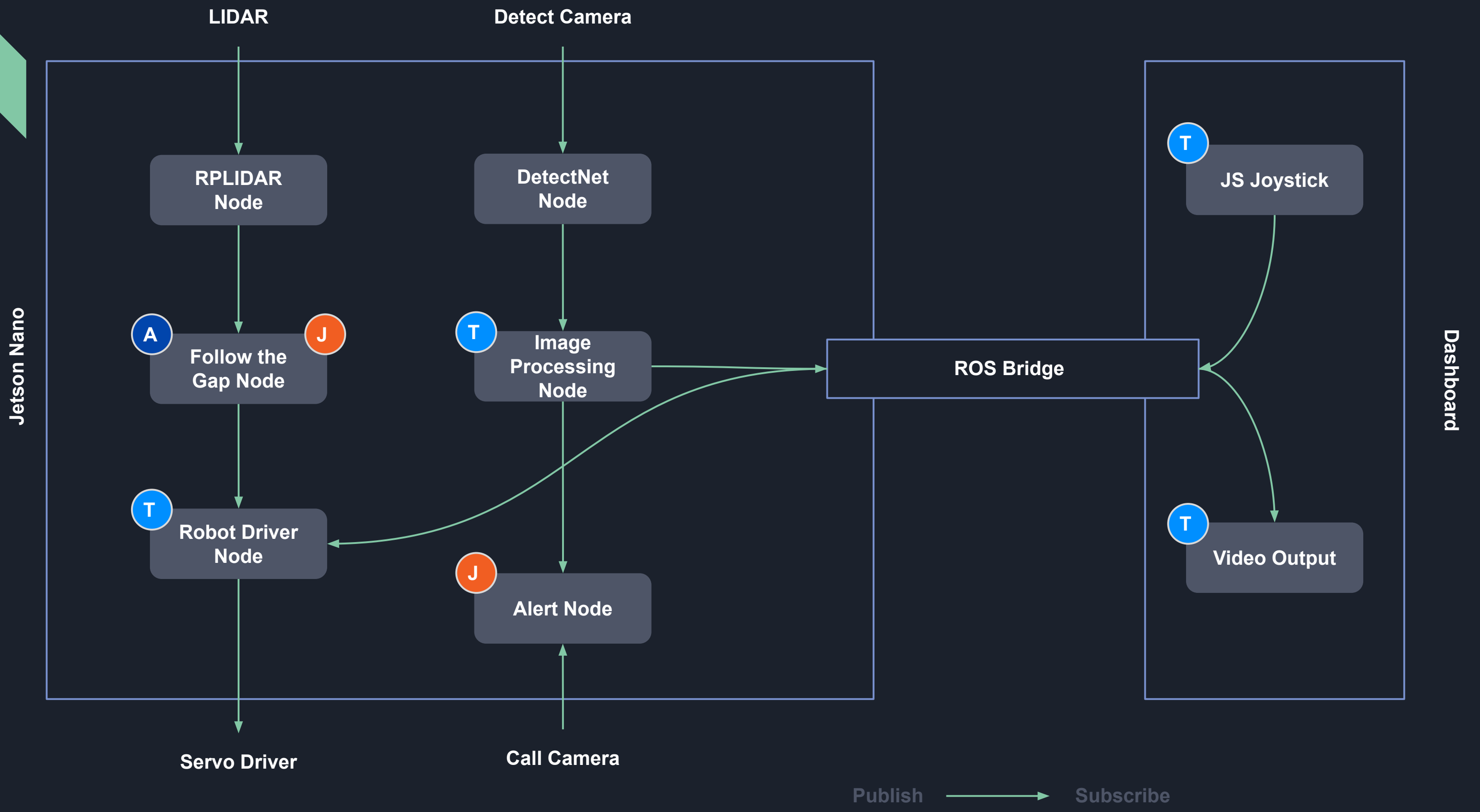
JZ

Hardware Diagram



TW

Software Diagram

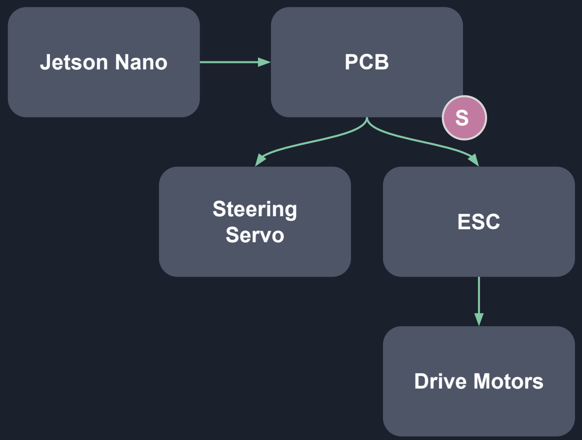


TW

Hardware Topics

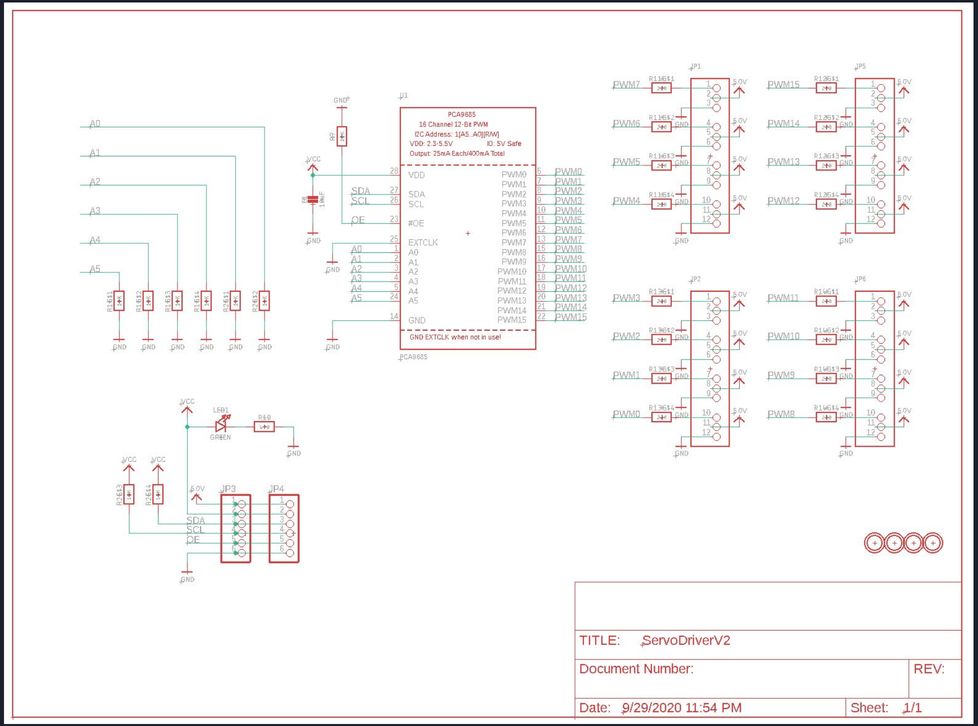
PCB

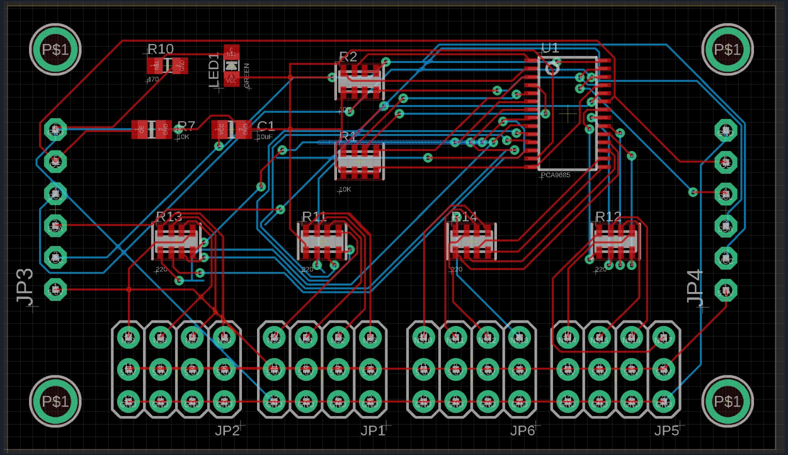
Our PCB acts as our servo driver that will fully control our motor and servo to maneuver our RC car as desired. It uses a PCA9685 chip that is an I2C-bus controlled 16-channel LED controller.

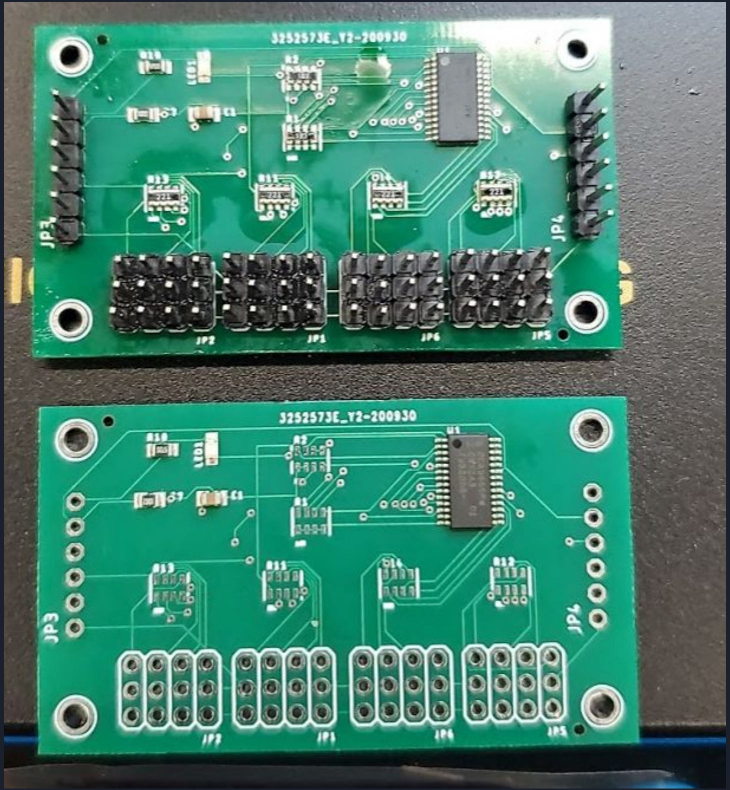


SA

PCB Design


PCA9685
16 Channel 12-Bit PWM
I2C Address: 1[A5..A0][R/W]
VDD: 2.3-5.5V
IO: 5V Safe
Output: 25mA Each/400mA Total
GND EXTCLK when not in use!
TITLE: ServoDriverV2
Document Number:
REV:
Date: 9/29/2020 11:54 PM
Sheet: 1/1


P$1
R10
LED1
R2
U1
R7
C1
R1
PCA9685
R13
R11
R14
R12
JP3
JP4
JP2
JP1
JP6
JP5



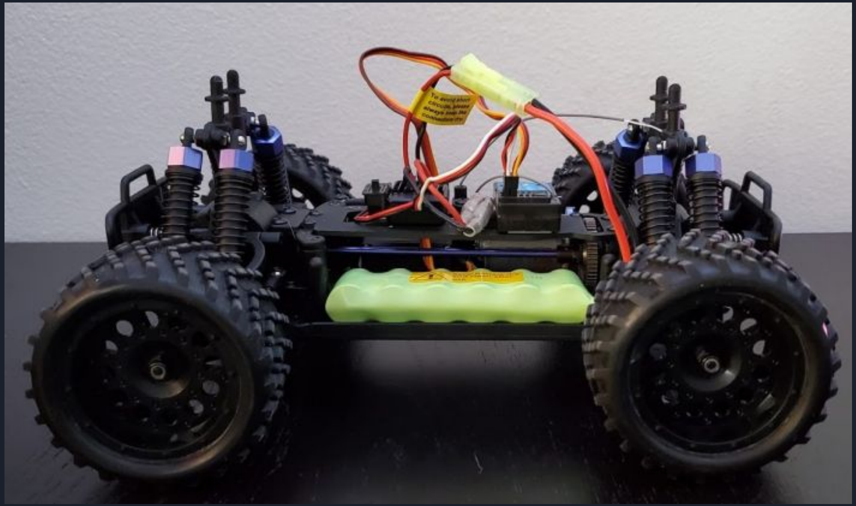
Components:

- Chip Resistor Array consisting of 4 resistors. 220Ω and 10kΩ
- PCA9685 Chip
- Pin headers for our wires

Purchased from JLCPCB

Mechanical System - RC Car

- Exceed RC Magnet EP Electric RTP Off Road Truck
- Four Wheel Drive
- Double wishbone suspension
- Double Coil spring shocks
- Rubber non slip tires
- All terrain
- $120



SA

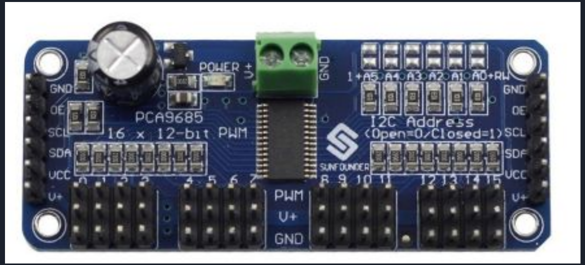
Mechanical System - Motors

- Exceed-RC RC380 Brushed Motor
- Exceed-RC Sp6003 High Torque servo
- Turn Steering method
- Electronic speed controller used
- Geared to send power to all four wheels

SA

Electrical System - Batteries and Power Distribution

- Servo Driver
- High Capacity (11000mAh) 7.2V Ni-MH battery
- Anker Portable Battery 20000mAh
- RC Battery Powers RC car
- Portable Battery Power Rasp Pi
- Rasp Pi Powers Sensors



SA

Sensors - General

The individual component(s) have been carefully selected for integration within our ROS Melodic ecosystem for ease of integration. Price of the component was also weighed in decision making.

Each sensor will have extensive testing for their respective marks of accuracy to ensure no product was damaged in transit.

In order to meet our objective goals for autonomous reactive traversal, and intruder detection. Through sensor fusion various sensors will be required to work in conjunction with each other in order completely cover all aspects of LookSee's responsibilities.

AP

Sensors - Camera

Logitech C615 Webcam

- 5 MegaPixels
- 1080p at 30 FPS
- Infrared sensors for night vision
- Inexpensive
- Uses USB connection
- Has built-in microphone

SA

Sensors - IR Array

The MLX90640 Thermal Camera Breakout (110°) from Pimoroni was in part due to the value of the Melexis MLX90640 with regard to the (768 pixel - 32x34-$25) resolution when in comparison between the GRIDEYE (64 pixel - 8x8-$60) and consumer available FLIR (4,800 pixel-80x60-$200) the MLX90640 offers a significant boost in performance for a very reasonable price.

When used in conjunction with the camera, the infrared image will be composed of an assigned color (blue for cold, red for hot) corresponding to a calculated temperature value (calculated from the reflected temperature and emissivity of an object) using the drivers written by Melexis. Once each of the 768 pixels has an assigned color a frame can be generated and then the process of chaining together frames can be accomplished to create relatively low cost thermal imaging. (Available at https://github.com/melexis/mlx90640-library) High temperature resolution may come at the cost of overall image resolution or a lower frame per second count.

For LookSee temperature and thermal imaging may be useful in confirming the identity of an object suspected to be human (an intruder) if our confidence with image recognition is not above our desired threshold of 70%.

AP

Sensors - IR Array (Abolished)

Unfortunately despite the numerous upsides to thermal imaging within the context of our project.

The thermal camera we selected was initially picked for its compatibility with the Raspberry Pi, rather than its ability to integrate in with ROS. It was thought that an achievable integration would be possible into our ROS ecosystem given it’s ease of use with the Pi.

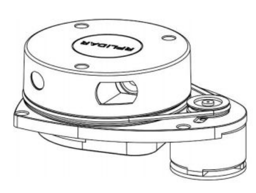
Though upon shifting to the Jetson Nano for superior traditional image recognition. Issues arose both with our selected library, and melexis originating library.

What occurs is follows via slavysis (a collaborator for the melexis library) ”Since both the MLX90640 and the Nvidia are expecting the other device to take control of the SDA, no device does that and the SDA stays high all the time = the MLX90640 writes FF to the EEPROM address instead of reading the data from it.”

https://github.com/pimoroni/mlx90640-library/issues/38
https://github.com/melexis/mlx90640-library/issues/61

AP

Sensors - Lidar



The RPLidar A1M8 offers an introduction the realm of Lidar scanners, providing a 2D - 360° scan capable of detecting up to 6 meters. It has a configurable variable scan frequency and is all offered for an incredibly affordable $100. In comparison 3D scanners quickly scale into the thousands of dollars which is impossible given our student funded budget.

Besides the phenomenal price for the Lidar and compatibility with our selected microcontroller; Slamtec has made a software development kit offering various functions to interact with the RPLidar directly as well as a separate open-source ROS node for integration with our other subsystems through ROS via publishing and subscribing.

Autonomous traversal requires information about the surroundings to make decisions, and Lidar accomplishes this task by providing a laser scan of the environment giving actionable data about the distances relative to the scanner at the given moment in time of after a scan has been completed.

Single Board Computer - Raspberry Pi 4b (Abolished)

For the relatively low entry price of 35$ the Raspberry Pi 4b comes with a Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz and a variable choice of ram (2gb, 4gb, 8gb) with standard gigabit ethernet, and an onboard wireless networking card that is bluetooth compatible.

With this low cost comes the added benefit of popularity, with a wider user base at various skill levels comes the benefit of a strong network for support from fellow users who have troubleshooted issues in the past.

Though a plethora of tools were designed with the Raspberry Pi in mind, and components are readily available . Due to constraints with image recognition and low framerate we decided to go with the NVIDIA Jetson Nano. More detail is provided on the subsequent slide.

AP

Single Board Computer - NVIDIA Jetson Nano

The Jetson Nano was more costly but provided a huge boost in performance when running any image processing software. The Jetson Nano combines a CPU and GPU into a single embedded module with shared memory. The goal of this product was to provide deep learning on the ‘edge’ while using relatively low power.

The Jetson Nano sports a 128 CUDA core Maxwell GPU, a Quad-core ARM CPU @ 1.43 GHz, 4 GB of LPDDR4 memory, and advanced camera support. The key part is the GPU which will greatly increase the performance of our image detection software.

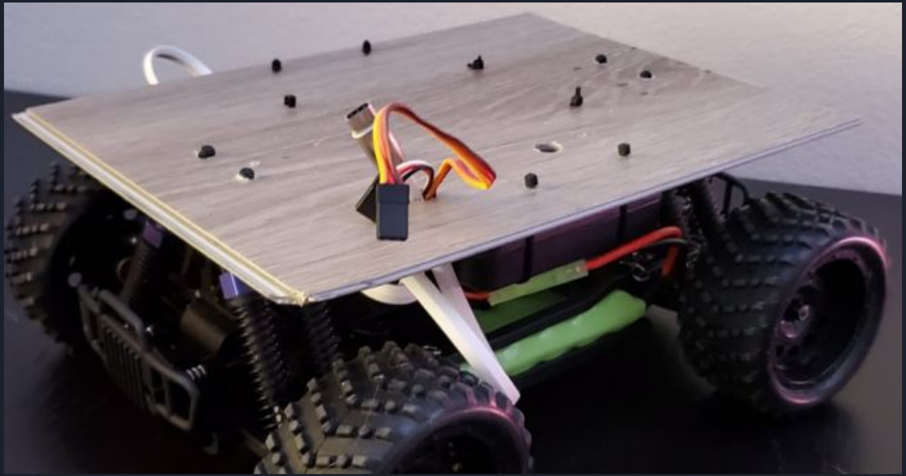
Also NVIDIA has already packaged together optimized solutions for the Jetson Nano when it comes to object detection and recognition.

TW

Prototyping

RC Car Prototype

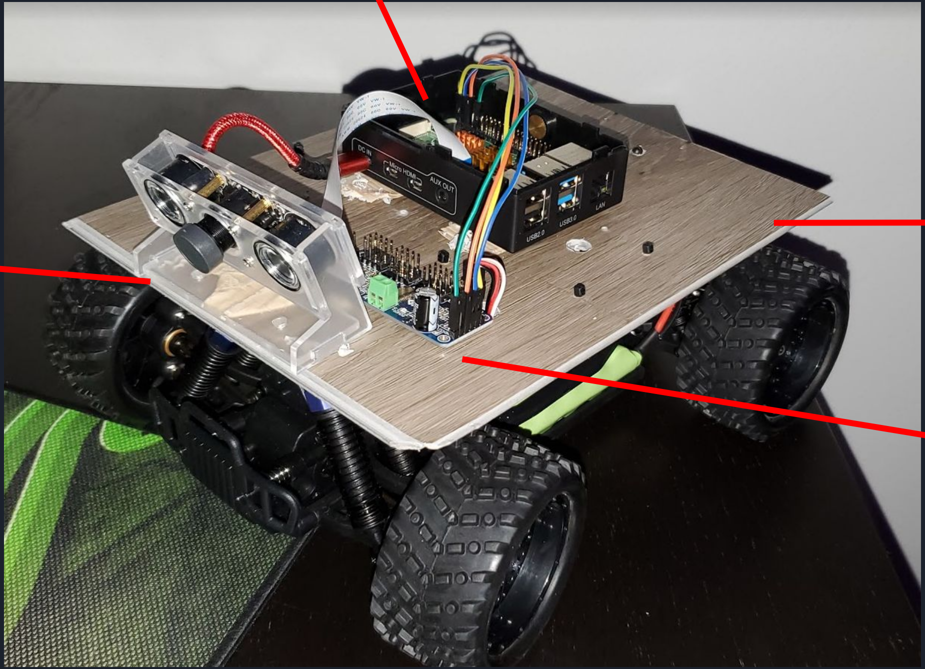
Base plate to mount electronics



Portable battery mounted under base plate

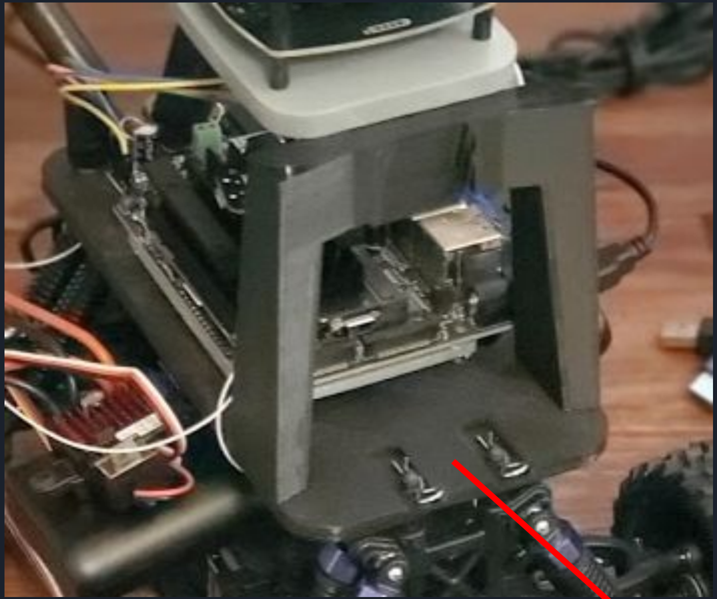
SA

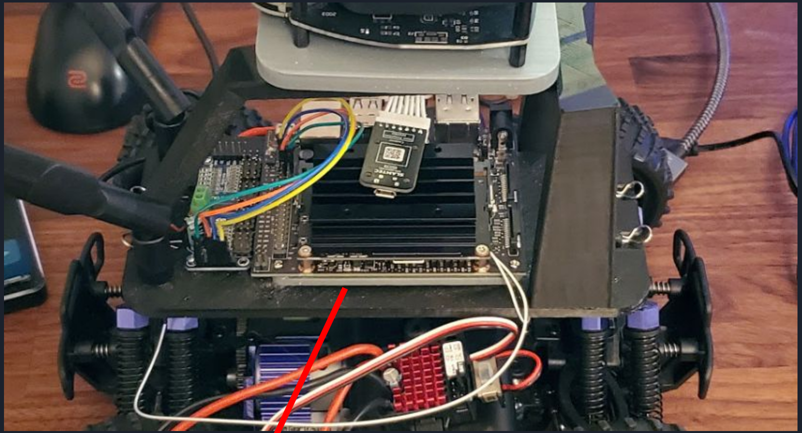
RC Car Prototype

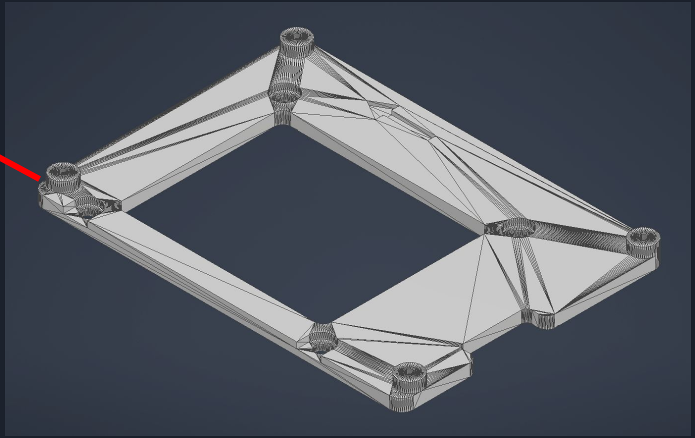


SA

Build Process







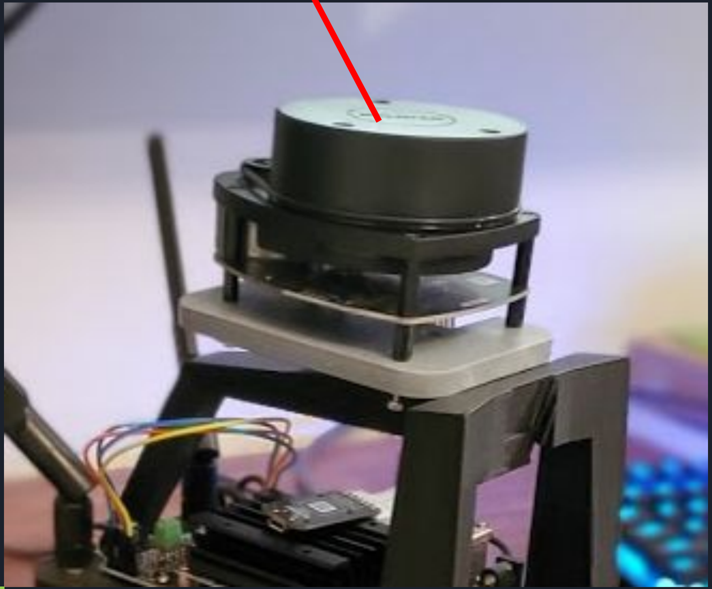
3D-Printed Adapter

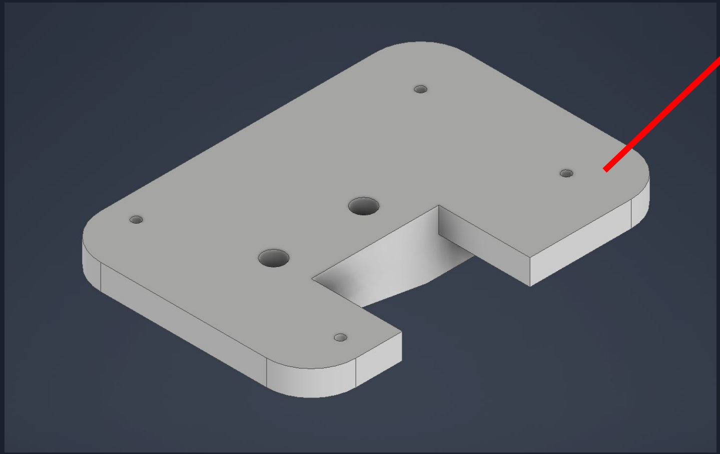
3D-Printed Frame

SA

Build Process

LiDAR





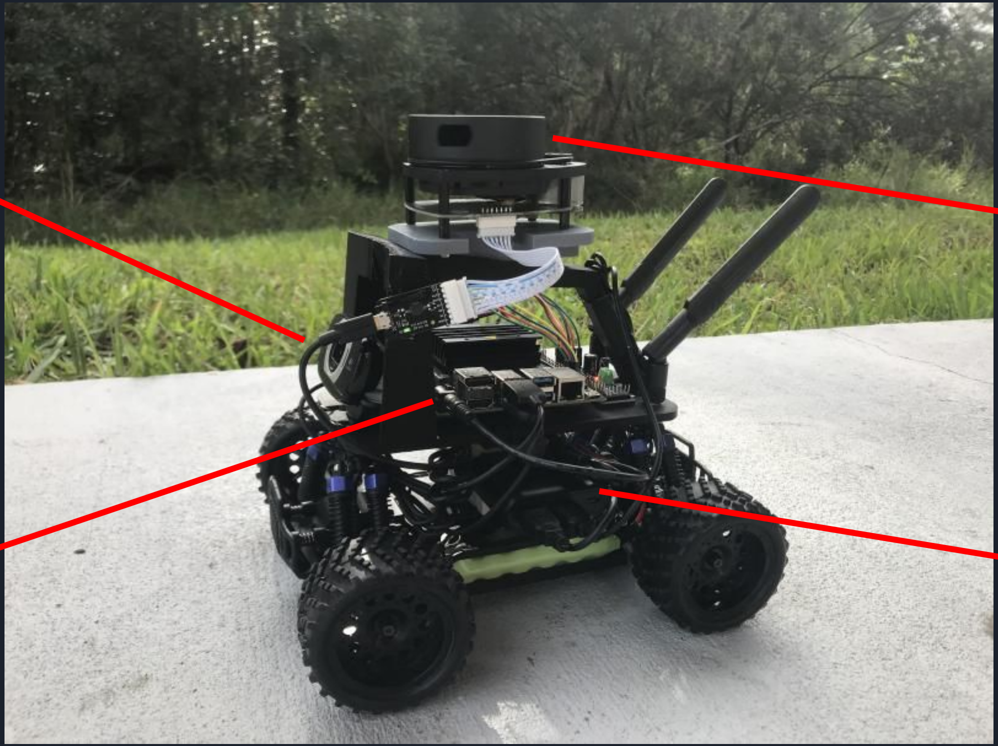
3D Printed
LiDAR
Mount

SA

Final Design

Web Cam

LiDAR



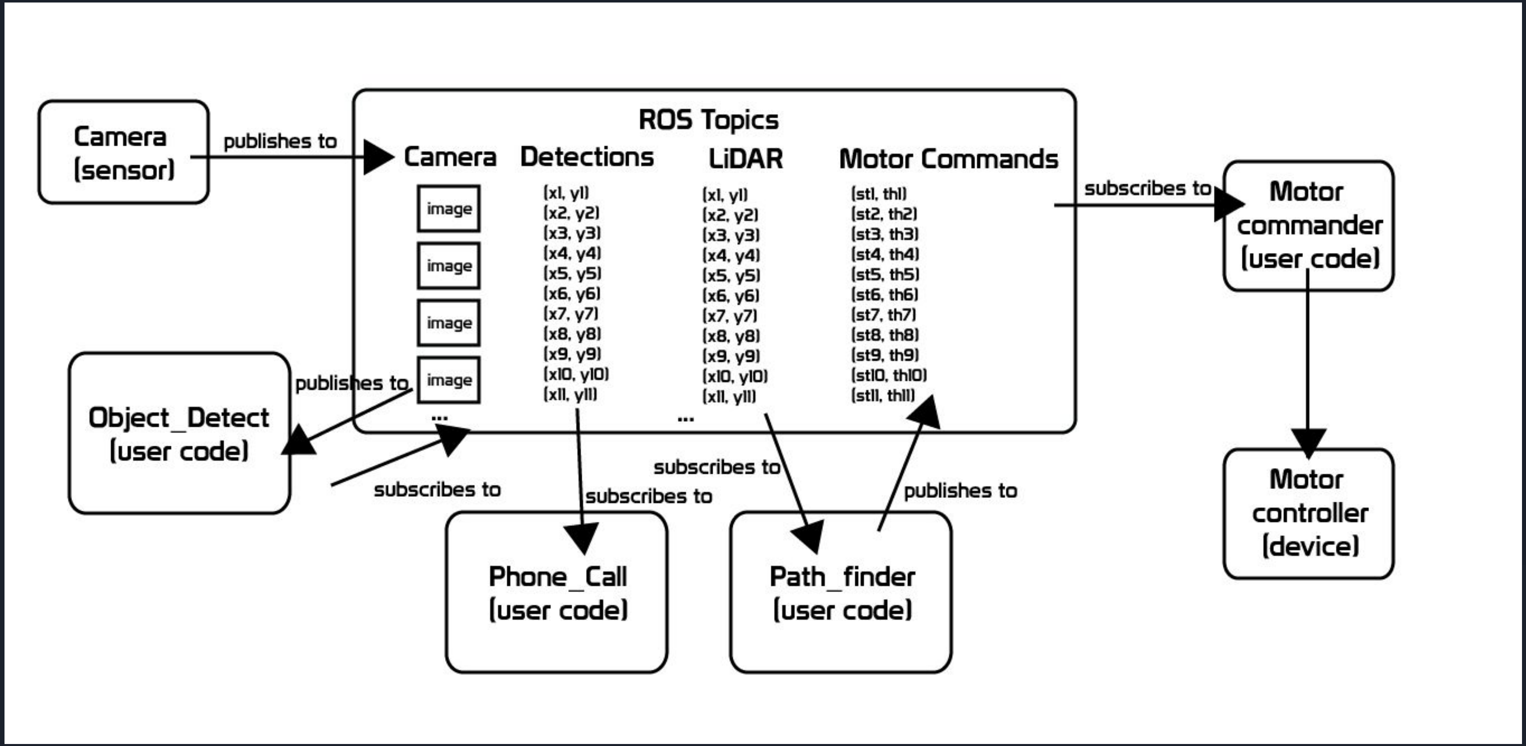
Jetson Nano

Batteries

SA

Software

Software Framework - ROS



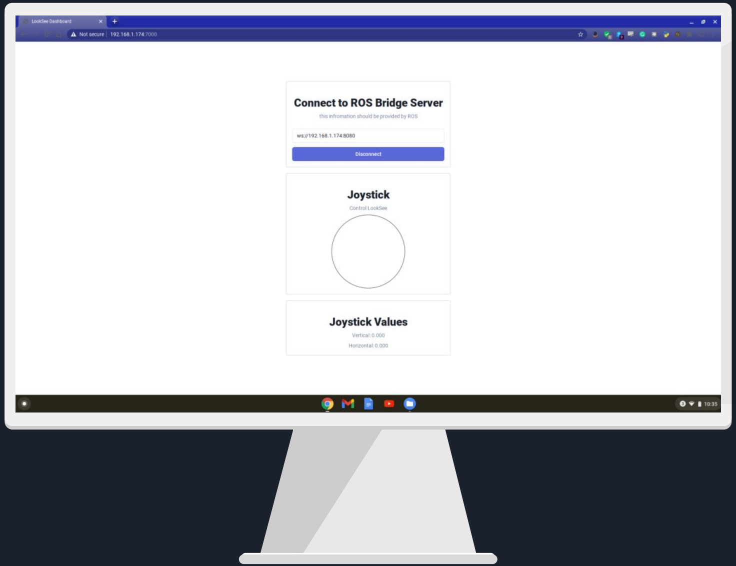
JZ

Software - User Interface

ROSBridge is a ROS component that creates a JSON API for a ROS system so non-ROS programs can interact with it.

We used roslibjs, a javascript library which uses the JSON API exposed by ROSBridge, to create a frontend that allows the user to drive the robot remotely. The on-screen joystick publishes ROS Twist messages to the robot_driver node.

The dashboard's UI was created using VueJS and TailwindCSS.



Software - Human Detection

Detecting a person using the camera is one of the main goals. We aim for a 70% detection rate.

The main things to consider is that we may only see someones legs or torso but we would want to count as a detection.

The goal is to use an integration of TensorFlow for object / person detection and ROS.

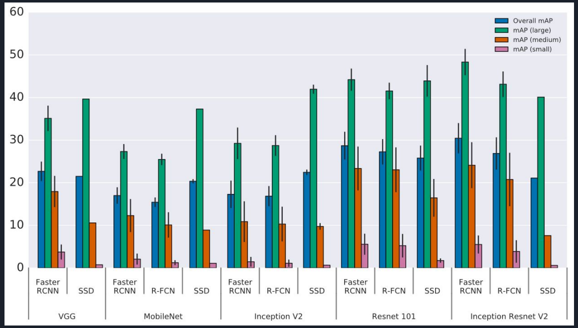
Luckily NVIDIA already has included many of the standard pre-trained models like SSD-Mobilenet-v1, SSD-Mobilenet-v2, DetectNet-COCO-Dog, and others.

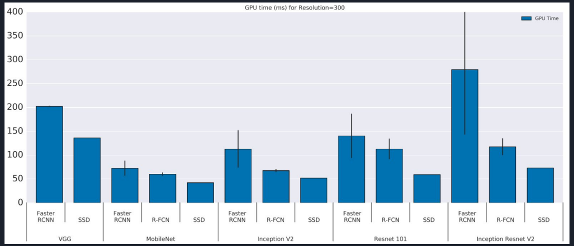
TW

Software - Human Detection

We used the ‘Speed/Accuracy Trade-offs for Modern Convolutional Object Detectors’ Google Research paper to compare and contrast the different types of neural network architectures.

The Single Shot Detector using the MobileNet for feature extraction performed the best when it came to the accuracy metric mean average precision (mAP) and GPU time in milliseconds for each model.





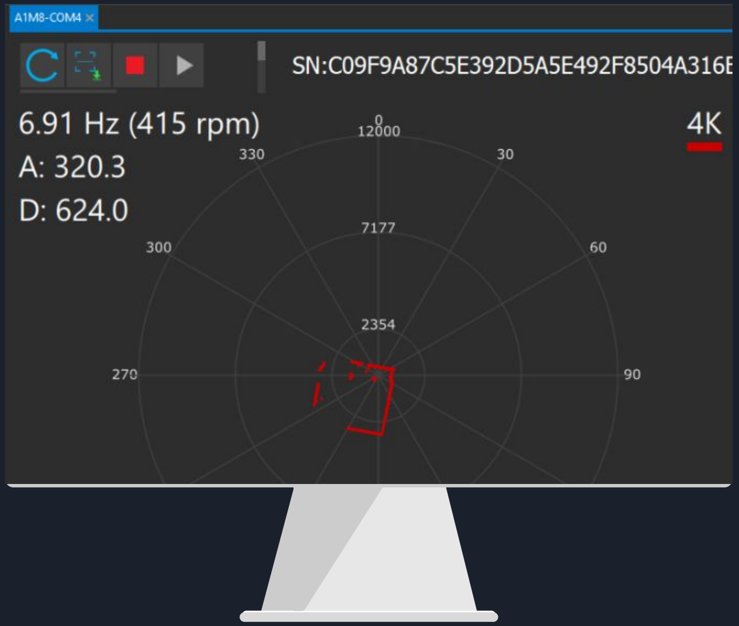
https://arxiv.org/pdf/1611.10012.pdf

TW

Software - Lidar

With the capabilities of the RPLidar demonstrated thoroughly with various examples in the software development kit, in addition to the ROS node created by Slamtec. Autonomous navigation is achievable through algorithms like Finding the Gap.

Our implementation of Find the Gap was based around extending the walls of closer distance (by the radius of the robot) over walls of further distance identifiable by disparities in distance readings. This allowed us to effectively convert everything into the into the point object space of the robot by enhancing the radius of every object by the size of the robot. Effectively this means that if the robot is capable of clearing a gap in the lidar readouts, it is capable of clearing the gap in the real world.



AP

Budget

Item	Quanity	Vendor	Donated	Cost with Qty.
RC Car	1	Exceed-RC		$119.95
RC Battery	1	Exceed-RC		$0.00
RC Battery Charger	1	Exceed-RC		$0.00
Power Bank	1	Anker		$49.99
Nvidia Jetson Nano	1	Nvidia		$99.99
USB to Barrel	1	SIOCEN		$7.99
Servo Driver	1	SunFounder		$9.99
Logitech Web Cam	1	Logitech	Yes	$49.99
Screws	1	Kuman		$8.00
USB Sound Card	1	Roccat		$19.99
LiDAR	1	RPLIDAR		$99.00
Twilio Phone Number	1	Twilio		$21.00
PCB	1	JLCPCB		$47.95
Micro SD Card	1	SanDisk		$9.99
RC Battery	1	Venom		$49.99
Wiring/Sleeving	1			$19.99
		Total		$613.81
		Donated Parts		$49.99
		Net Total		$563.82

Our aim is for the project to cost no more than 200 dollars to each member of the team. As our objective is to make this a low cost robot. We are able and willing to spend any extra if needed due to the possibility of something breaking or not working.

SA

Plan for Completion

We have a meticulous plan for each of us to complete certain tasks during specific weeks. We broke up our responsibilities into modules that require minimal in-person interaction until later in the semester.

The four modules are:

- Computer Vision - Tyler
- Autonomous Navigation and Thermal Imaging - Austin
- Teleop Navigation and User Interface - Jade
- Hardware/Power/PCB - Stavros

Week	Computer Vision	Navigation/Thermal	User Interface	Hardware
1	All: Install Linux and ROS on personal computer, verify functionality. Verify all parts have been received and are functional.			
2-3	Write a program that can identify a human in an image and output a confidence value.	Write a program that can check the temperature at a location in an "image" received from the thermal array.	Create a user-facing page for a web server with nine buttons, space for a camera feed, and appropriate graphics.	Design PCB and place order. Pay for express shipping.
4	Modify above program so that the location of the suspected person is published to a ROS topic.	Modify the above program so that it checks the location published by the computer vision program, and publishes a flag to a new ROS topic if 60% confidence is exceeded.	Modify code for buttons so that all buttons publish to their respective ROS topics.	Create additional layers on chassis to securely hold sensors and other electronics.
5-6	Work with Hardware to integrate all code the team has completed thus far into ROS on the Raspberry Pi that will be used on LookSee.	Begin implementing find-the-gap navigation and testing in Gazebo if hardware has not yet been received.	Write Skype call software.	Work with Computer Vision to integrate all code the team has completed thus far into ROS on the Raspberry Pi that will be used on LookSee.
7	Work with User Interface to begin testing intruder detection and alert features.	Work with Hardware to test and fine-tune autonomous navigation.	Work with Computer Vision to begin testing intruder detection and alert features. Implement UI video stream.	Assemble the robot. Work with Navigation to test and fine-tune autonomous navigation, test hardware.
8	All: Meet to perform all tests again, identify problem areas, strategize for the remainder of the semester, and decide which stretch goals to pursue.			
9-End	All: Fix issues that may have been identified, pursue advanced and stretch goals, perform tests.			

Timeline

WEEK 1

Installations and Checks

All members install all software and test any hardware they are responsible for.

WEEKS 2-3

Preliminary Coding

Programmers write a version of their code that runs independently of ROS. The PCB is designed and ordered.

WEEK 4

ROS Integration

Programmers modify code to work with ROS. Chassis assembly finalized.

WEEKS 5-6

Code Integration

Everyone works together to put finishing touches on the code and make it work on the robot instead of personal computers.

WEEK 7

Preliminary Testing

All tests are performed, notes are taken to determine what needs to be focused on.

WEEKS 8+

Adjustments, Improvements, Final Testing

Adjustments are made based on test results, and tests are redone. Optional additional features may be added.

JZ

Primary Requirements - Fulfillment

01 By the end of the semester, LookSee should be able to autonomously navigate a space similar to the average office building for at least 10 minutes without getting stuck.

RESULT: Looksee can navigate autonomously in a reasonably uncluttered environment until its batteries run out.

02 By the end of the semester, LookSee should be able to identify a human moving around in front of its path with at least 70% accuracy.

RESULT: The computer vision has near-100% accuracy.

03 When LookSee identifies a human intruder, it must call the emergency contact, and allow the contact to take over navigation of the robot.

RESULT: LookSee does this with 100% accuracy.

JZ

Specifications

By the end of this project

- LookSee navigates autonomously with Find-the-Gap navigation
- LookSee's computer vision is accurate enough that we did not use thermal sensing to augment it.
- LookSee will have a user interface that can be accessed through a web page.
- LookSee's user interface allows a human operator to drive the robot
- LookSee uses Twilio and Jitsi Meet to video call the human user.
- LookSee uses ROS as its software backbone.

JZ

LookSee

small business surveillance system

Thank you!