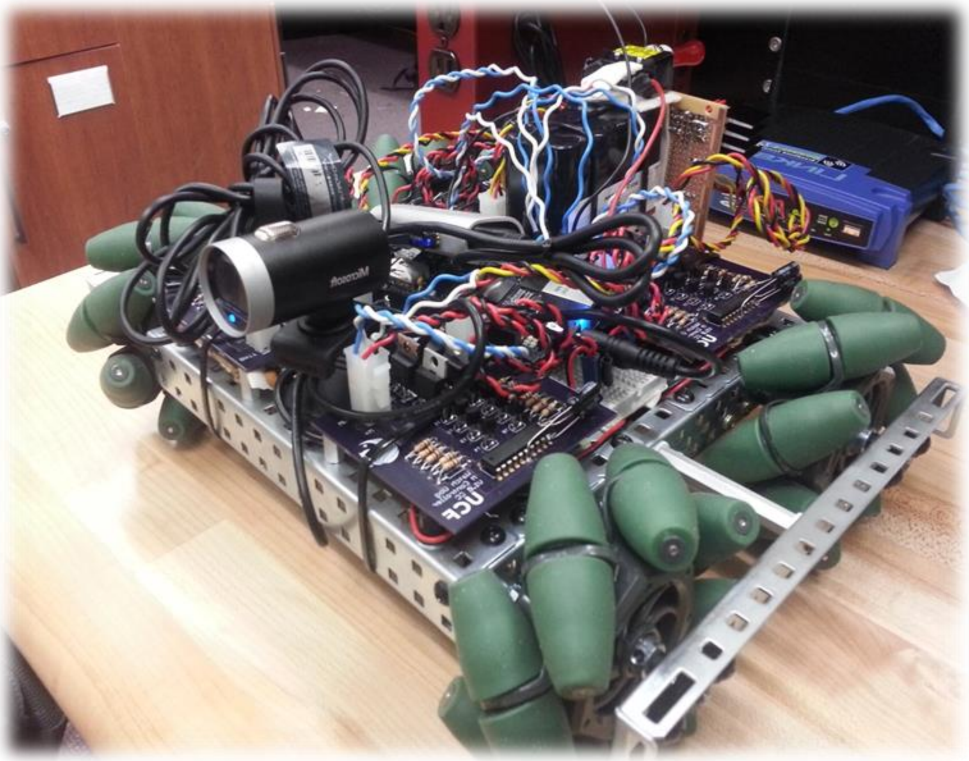
T-100 Watchdog



Department of Electrical Engineering & Computer Science
University of Central Florida
Summer 2014-Senior Design II
Group #4

Ismael Rivera (Computer)	ismaeljrivera@knights.ucf.edu
Journey Sumlar (Electrical)	sumljf@knights.ucf.edu
Chris Carmichael (Electrical)	Carmichael.Chris@knights.ucf.edu
Warayut Techarutchatano (Computer)	Warayut@knights.ucf.edu

1.0 Executive Summary

The purpose of this project is to gain experience through creating a project and being part of the research and development from its initiation until its end. This will allow us to gain real world experience while still in a classroom environment. The popularity of computer vision projects has grown recently, thanks to the availability and advancements in the technology of embedded systems and computer vision based algorithms. For our senior design project, we wanted to implement computer vision as well as automation. One application that will allow us to implement both of these features would be to design a robotic vehicle with a mission of tracking and pursuit. In addition, we wanted to make use of thermal imaging. Thermal cameras have usually been restricted for military and law enforcement use, but recently it has become easier for regular consumers to obtain. Although there are some restrictions as to what models a regular consumer can purchase, this has still allowed a large group of hobbyist to add thermal imaging to their projects.

The robotic vehicle will be equipped with multiple subsystems including cameras, control and mobility, and wireless communications. For the camera subsystem we will be utilizing a regular webcam as well as a thermal camera. The thermal camera will be primarily used in for tracking our targets, while the regular webcam will be primarily for relaying images to the user. For the control and mobility subsystem we would like for the robot to be autonomous. We will utilize OpenCV algorithms which will allow the robot to detect our targets. Once movement has been detected, OpenCV algorithms will be utilized to track a specific target and maneuver the vehicle across the room while avoiding any obstacles in the way. In addition to the vehicle being autonomous, we would like to allow the user to control the robot via a mobile application. The wireless communication subsystem will make it possible to stream a live video feed of what the robot is seeing via the mobile application. In addition, a text message and a snapshot will be sent as a notification for when any movement is detected.

All the mentioned subsystems will be combined to make a robotic “Watchdog” that can be used for home security purposes. We expect this experience will be both challenging and rewarding for each member of the group. This project will broaden our knowledge and skills by allowing us to work as a team consisting of members from two different disciplines of engineering to design, build, and test a unique piece of technology.

2.0 Project Description

The following sections will detail the motivation behind the project, as well as the goals and specification that we would like the project to attain.

2.1 Motivation

As a team we decided that we each wanted to put our individual skills to practice, while simultaneously learning new skills to put into practice as well. Since our team consists of half Computer Engineers and half Electrical Engineers, we wanted to use our senior design project as a chance to combine all of our experience and knowledge in these two related fields while pursuing new topics of interest for each team member. From the Computer Engineering side, algorithms are always a subject of interest and one where there is never enough practice. In addition, mobile applications are currently quite popular along with web services. From the Electrical Engineering side, implementation of hardware is a very important part of this project. The ability to design, implement, and integrate hardware are needed to become a successful Electrical Engineer.

In order to accommodate all the skills mentioned, we decided to use a robotic vehicle as the base of our project. By using a robotic vehicle as the base of our project, we are able to simply add separate subsystems to enhance its functionality, which will help us apply and gain the new skills desired. By combining all of our ideas, desired interests as a team, cost, and difficulty, we decided to make an autonomous vehicle that will track and follow a person based on their heat signature. In order for the software side of the team to gain experience with algorithms, we decided to use the OpenCV library. The OpenCV library is packed with many features, which makes it able to give any project endless possibilities. Another advantage is that it is free for both educational and commercial usage. In order to gain experience with mobile platforms, along with network communications, we decided to implement a mobile application for Android devices where the user will have the ability to control the vehicle and stream a video feed through the internet. As for the Electrical Engineering side, the hardware requirements of the vehicle (motor controllers, power systems, and sensors) will allow the Electrical Engineers a chance to apply skills that they have learned in theory, and gain general experience in fabrication.

Each year UCF students come up with many great ideas, most of which turn into great projects. As a team we wanted to do something different, something that will make our project stand out from all the other ones. Because of this, we decided to use a commercial grade thermal camera for our tracking sub system. These types of cameras are not new technology but are a technology that has not been used often in previous projects.

2.2 Objective and Goals

The goal for this project is to create an autonomous robotic vehicle that is able to alert a user when someone has gained access into their home. In addition, we would like for the user to be able to see what is actually occurring when an alert is received, as well as have the ability to control the vehicle. The project consists of the following objectives: Object Detection, Object Tracking, Object Avoidance, Autonomous Control, Remote Control, and Wireless Communication.

For the Object Detection, we would like for the vehicle to be stationary while scanning a specific area or room, with possible entrance points, for any movement or changes in the view. Once a movement is detected, the robot is to be taken out of its idle state and be ready to perform. Once the robot has been set to a "ready" state, the robot needs to follow the object that triggered the change of status. While the tracking is occurring, it must also avoid any objects that are on the way by utilizing proximity sensors that are equipped on the perimeter of the vehicle. If an object is blocking the robot's path, the robot has to be able to find a different path to continue.

Another objective is for the robot to be autonomous, meaning it will not need any user interaction to be able to perform the tasks that it has been assigned to complete. Additionally, we would like to add an option where the user can take control of the vehicle by using a mobile app. If the user decides to use the mobile application, any commands from the user are to take priority and the robot is to follow the user's commands.

The last objective is to have integrated wireless communications. By having wireless communications, we would be able to stream a live video feed of what the vehicle is seeing to the user via the mobile application. In addition, we would like for the user to be able to save snapshots from the vehicles camera, allowing the user to keep a copy on their device.

2.3 Project Requirements and Specification

After researching different components for the vehicle we decided what basic capabilities we wanted the vehicle to have. We then made a list of the specifications we would like the vehicle to meet. These are the project goals that we will use while making design choices for the vehicle.

- To provide a fully autonomous robot.
- Automatically detect and track a person that enters its range of view.
- Detection and tracking will be achieved with a thermal camera.
- Connect to Wi-Fi network.
- Transmit video feed to a mobile device.
- The user will be able to tell the robot which human to detect from the mobile app GUI (If more than one person is in sight).
- If someone is running towards the robot, it will run away to safety.
- Navigate throughout the house using object avoidance.

Dimensions (l x w)	11” x 6”
Operating Range	12 ft.
Power Supply Voltage	15 V
Maximum Weight	20 lbs.
Minimum Speed	1 m/s
Max Speed	10 m/s
Battery Life	3 hours
Recharge Time	< 8 hours

Table 2.3-1 Vehicle Specifications

2.3.1 Sensors and Microcontrollers

As we were researching different microcontrollers we will use a checklist of several requirements needed for the vehicle to run efficiently and function properly. In order for the vehicle to move on the right path while tracking, the vehicle will use the given sensors to determine if obstacles are present and to calculate a path to avoid them. This microcontroller will be the brain of the vehicle. It not only has to calculate the signals required for the motor controllers to appropriately steer the vehicle, the microcontroller will also have to make all of the calculations involved in tracking with the thermal camera, and handle the wireless communications to connect with the mobile app. These requirements are all detailed in table 2.3.1-1 below.

Requirement ID	Requirement Description
1	The microcontroller must be able to control the motors speed
2	The microcontroller must be able to control the motors direction
3	The microcontroller must be able to use obstacle avoidance algorithms to calculate the path
4	The microcontroller must be able to control the wireless transmitter allowing for the user to receive video stream
5	The microcontroller must be able to receive and execute commands from the mobile application
6	The microcontroller must be able to monitor/control the state the robot is currently in. (Idle or active)
7	The microcontroller must be able to use thermal and conventional web camera for detection and tracking

Table 2.3.1-1 Microcontroller Requirements

2.3.2 Mobile Application/Web Server/Wireless Communication

The vehicle will be able to connect to the internet via wireless communications. It will then connect to a web server that is attached to the mobile application, allowing the user to take control of the vehicle, receive a video stream from the web cam, and have the ability to save snapshots of a potential intruder. All of the specifications for this sub system are listed in the table below.

Requirement ID	Requirement Description
1	The robot shall the able to receive commands from the user via the mobile application.
2	The robot shall be able to stream a smooth live video feed of at least 15 frames per second.
3	The robot shall be able to stay connected to the Internet via a wireless connection.
4	The robot shall always listen and accept any incoming connection from a mobile device running the mobile application.
5	The robot shall be able to switch between being autonomous and being controlled by the user.
6	The robot's mobile application shall only allow access to an existing account with correct credentials.
7	The robot's mobile application shall be able to load and show the user the video feed.
8	The robot's mobile application shall be able to dial 9-1-1 from inside the application's interface.
9	The robot's mobile application shall allow the user to save a snapshot of a specific frame from the video feed to the internal storage of the mobile device.
10	The robot's mobile application and web server shall communicate between each other without any noticeable lag. Each instruction should take 1 seconds or less.
11	The robot's mobile application shall be able to control the movement of the robot in any direction.

Table 2.3.2-1 Mobile/Network Requirements

2.3.3 Cameras/Image Processing

The vehicle will be equipped with two separate cameras; a thermal camera for tracking, and a regular web cam for transmitting a video stream to the user. All of the camera subsystem specifications are listed in the table below.

Requirement ID	Requirement Description
1	The robot shall be able to use a generic webcam to capture video and pictures.
2	The robot shall be able to use a Thermal camera to capture video and pictures.
3	The robot shall be able to toggle between the webcam and Thermal camera depending on the time or light conditions.
4	The robot shall detect any moving person within 10 feet.
5	The robot shall track the detected person and continue tracking as long as possible.
6	The robot shall stay within two feet of the person detected and being tracked.
7	The robot shall be able to utilize the cameras to find a new path if any collision is detected.

Table 2.3.3-1 Camera/ Image Processing Requirements

3.0 Project Research

In this section we will document all of the research we performed throughout the design process.

3.1 Existing/Similar Projects

Autonomous Tank- A fully autonomous vehicle that could potentially serve a purpose on the battle field in the future. This was a project (pictured below) from a group of students at the Georgia Institute of Technology. The group hopes to save lives on the front line of the battle field with a more efficient maneuver than that of human beings. Minimizing casualties by eliminating the need for humans to be in the danger zone of battle fields is the ultimate goal of this scaled proof of concept for the autonomous battlefield tank.

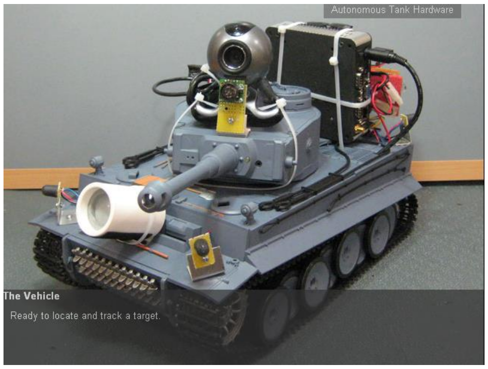


Figure 3.1-1: Autonomous Tank

The vehicle is equipped with infrared sensors as an environmental detection system. On top of that, the color detection algorithm and digital imaging device are added to assist with this concept. With all of these in play, the robot can make a decision to fire onto the acquired target. Utilizing a projectile motion concept and the real time detection, the aiming system is able to acquire the target accurately.

The software was developed in Visual Studio 2005 under a Microsoft .NET environment on top of the Windows Embedded CE 6.0 with a computer vision library OpenCV to handle image processing. For this project, the control system is built on top of an operating system which allows more functionality to be added on for later versions. The team also published the code for the tank as an open source for anyone to try to develop the same or similar project.

All the hardware used on the tank is very much conventional parts that could be acquired online or at a hobby shop. Although there are a few parts that might require more knowledge than plug and play such as the CV-HB 401 Dual H-Bridge for the motor control system. It accepts the voltage range from 5V up to 28V for the operation. The part is produced by Critical Velocity and well-documented specifications are readily available.

Autonomous Ball Collector- This project has a nickname A.B.C and is from an Engineering team of students here at University of Central Florida. The concept of design comes from a very simple idea of assisting a game of tennis. Everyone who plays tennis would know how annoying and dreadful it is to stop the game or the practice just to pick up the ball from the ground, let alone walk over 20 feet to the left side of the court and another 20 feet to the right to gather all the balls. A.B.C (pictured below) is designed and developed entirely just to serve that purpose. The tennis game is now more pleasant.

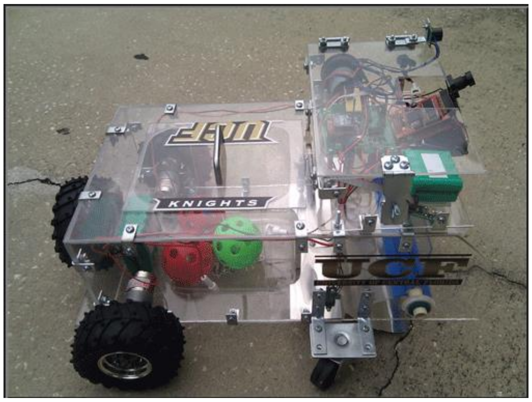


Figure 3.1-2: Autonomous Ball Collector

The robot seems to be the conventional robot that can be seen at science fair but the functionality is what the team is going for. The ease of use is simple enough. All is required is turning on the power and just let it collect all the balls. There’s a plastic ball cage in the back of the frame that holds all the collected tennis balls and it is as straight forward as opening the container and reaching for the ball. For any tennis enthusiast, that’s more than they could ask for. A tournament could even save money by using A.B.C instead of hiring a ball boy.

The software interface of A.B.C is an AVR programmer. It is produced by Atmel at a low cost and ready to be use with plug and play functionality. The chip utilizes a flash memory and will execute whatever program that is written inside. The chip runs at the speed about 10MHz with built-in 1KB of RAM and 10KB of storage. It seems very inefficient compare to a typical modern computer but the idea is to consume the least amount of energy as possible throughout the entire session of tennis.

Enriching the user interface for A.B.C, the robot even comes with a manual to help the user navigate through all the features of the robot. As mentioned above, there's probably nothing more to operating this robot than let it do its thing. A.B.C comes with a 12v battery that is rechargeable and have a switch on and off for the autonomous mode. The very cool thing is that A.B.C has a station that it will return to when the balls are fully collected.

KnightCop- Another project from local talent here at UCF, KnightCop (pictured below), is an autonomous robot that's aimed to assist the public services of the law-enforcement and even fire fighter. Many might have a negative connotation of surveillance robotic systems but the purpose of this robot is nothing remotely close to being sinister. The assistance that the KnightCop is supposed to provide is for the safety for our local men and women in law enforcement. In scenarios with potentially unstable terrain or environments, the aid of this robot would be invaluable in preserving human lives.

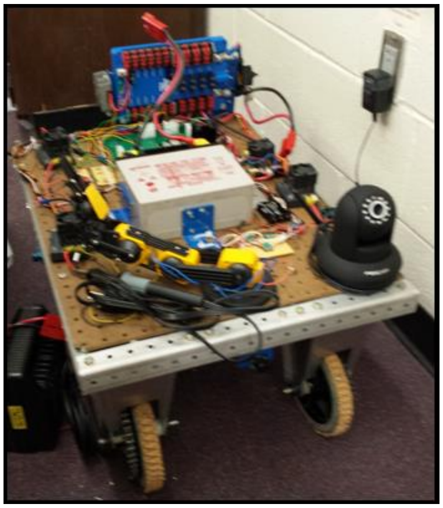


Figure 3.1-3: KnightCop

KnightCop is controlled and operated with an MSP430-G2553 from Texas Instrument. The vehicle is equipped with multiple sensors and the ability to navigate itself through different types of terrain. The wireless controls allow for human interference with certain decision making processes. It is, otherwise, mostly able to operate on its own. Along with other useful features, KnightCop offers two-way audio communication capabilities, allowing immobilized victim to communicate back to the rescuers.

On the high-level design point of view, the third perspective control is derived from an Android application. The user interface is simple enough to eliminate learning curve and capable of communicating to a personal mobile device such as a cellular phone. KnightCop also streams out a real-time image through IP network, allowing the operator to see what the robot is seeing.

Components such as temperature sensors and ambient light sensors are on board to give the robot more awareness of its environment. These kind of information can help the rescuers decide how hazardous the situation is at the particular moment. Knowing as much as possible could have significant impact on preparation in an emergency situation.

Wall-E Robot- It is exactly what the name refers to. The one and only beloved Wall-E from Pixar Animation Studios in toy form, modified with an overhaul in the internal components. Apparently, the original toy was purchased from Toys R Us with a price of $34.99. The project was intended to serve for amusement purposes entirely. In other words, this toy was bought and turned into a super toy. However, the fact that it is just a toy and serves no practical functions, does not take away from its sophistication and creativeness.

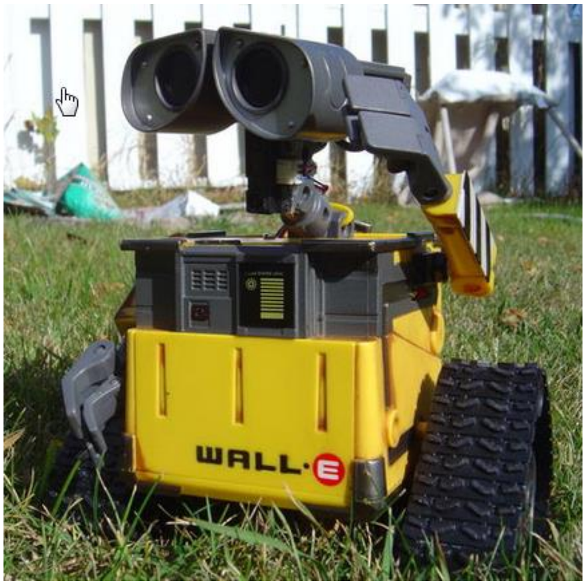


Figure 3.1-4: Wall-E Robot

The first feature that Wall-E demonstrates is the ability to receive voice commands and follow through with the execution. His creator voiced out the command “Wall-E, follow my face”, he then stare right at him and started following his face. Also, Wall-E does the same with a red ball. The impressive part is that Wall-E does not just execute finite sets of instructions base on the voice command. He continues to follow the ball even when it is moved away from him. Another functionality that he’s able to perform is dancing. These features alone make Wall-E more advanced than any of the other toy robots in the store.

Wall-E’s mobility comes from the eight servo motors it has on board. All of the parts are taken out from EZ-Robot original parts. The components are very standardized for conventional hobbyist. The EZ-Robot platform comes fully equipped with its own proprietary software for the control system. There are available options for both programmers and non-programmers. EZ-Robot also has mobile software to control the robot from an Android or iOS mobile device. Wall-E also demonstrated this functionality. This is considered to be a hobby project more than an engineering design one, however it showcases possibilities of what we could do.

3.2 Autonomous Vehicles

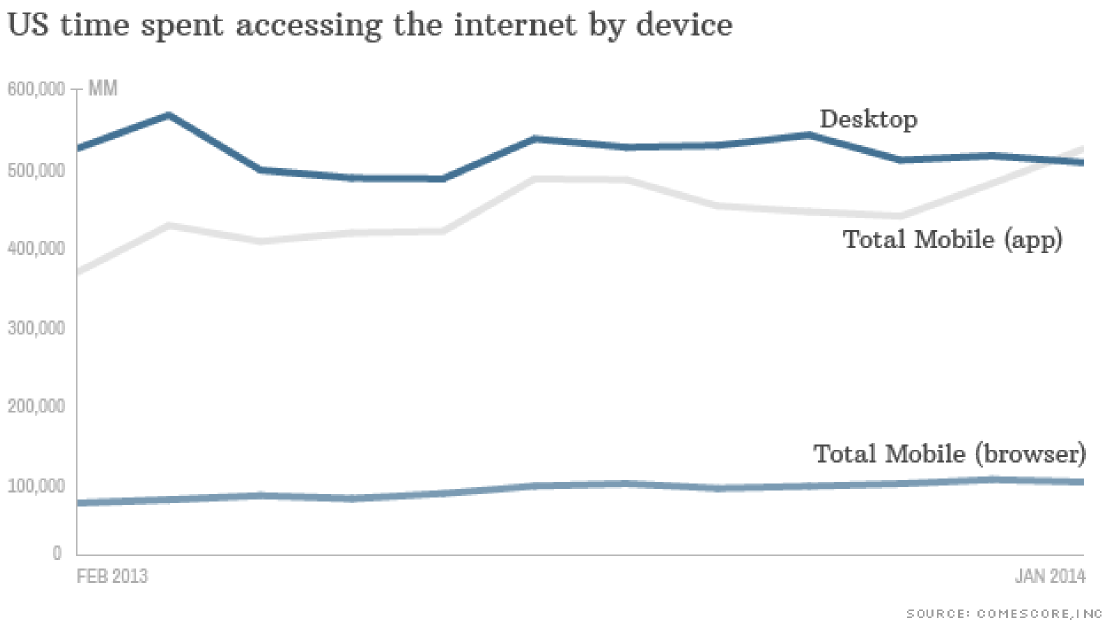
Autonomous robots are intelligent robots that are capable of performing tasks with some degree of self-sufficiency. The unique feature of autonomous machines is their ability to function without continuous human guidance. These specific robots are capable of interacting with the environment, sometimes even gaining knowledge and adapting to their surroundings. While these features are impressive, there is another feature that should be noted of the fully autonomous robots, which is self-maintenance.

Through modern advancements, autonomous robots are able to efficiently perform on land, in water, air, and space. By integrating various types of sensors and peripherals, these robots are capable of maneuvering around and interacting with a multitude of environments. For example, infrared and ultrasound sensors are used to allow the robot to detect obstacles. Advanced robots are capable of analyzing their environment and adapting to it accordingly.

It seems that most experts in field believe that the key to superior autonomy lies within machine learning. The common notion of self-reliance entity requires very minimum external influence with their operations. The conventional programming method such as feeding explicit instructions tends to limit the range of adaptability for the robot. With machine learning, all we need to do is direct the machine instead of write out every single instruction. A bipedal walking robot can use just a single line of code to learn to balance itself, instead of having to use hundreds of if statements. This may sound conveniently simple to achieve such milestone but that one particular line of code could take scientists and researchers months to derive it. However, in the end, results are the driving factor for any achievement.

3.3 Control Methods

In order to notify the user if any intruder has entered his/her home a mobile application needs to be created. By having a mobile application a message can be sent to the user’s phone or tablet where the user will be alerted of the situation and he can make a decision. According to a CNN Money report made on February 28th, 2014 mobile devices accounted for 55% of Internet usage in the United States in January 2014 as can be seen in the chart below:



SOURCE: COMESCORE,INC

Figure 3.3-1 Internet use of Mobile Devices vs Desktops

This is the first time that mobile devices have surpassed the PC Internet usage. As a result, a mobile application will be created in order to take advantage of the market share.

3.3.1 Mobile Application

Mobile phones have increased in popularity in the past few years. The top 4 mobile operating systems are:

- Android
- Blackberry
- iOS
- Windows

According to Kantar’s 4th Quarter 2013 report on mobile market share, the Android operating system accounted for 50.6% of market share in the United States alone. In a Google earnings call in 2013, there were 1.5 million Android devices being activated per day. This of course is including smartphones, tablets, as well as other new devices running the Android operating system.

The Blackberry operating system has been declining for the past few years, although it once was a very popular operating system especially for business people. The market share for the Blackberry operating system was 0.4% in the

United States. Apple's mobile operating system, iOS is the second most popular to Google's Android with a market share of 43.9%. Microsoft's mobile operating system, Windows Mobile, had a market share of 4.3%.

Since Android is the most popular mobile operating system, we decided to create the mobile application for Android devices. Below are some advantages and disadvantages of Android.

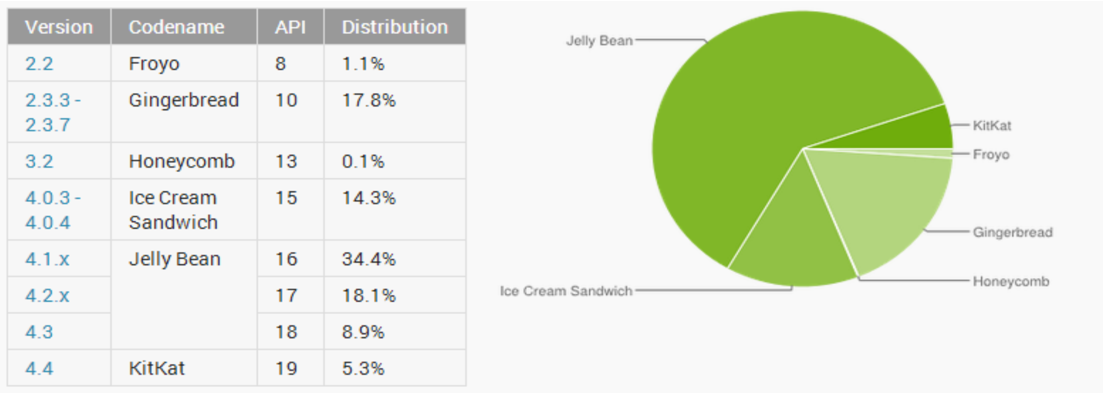
Advantages

- No learning curve, the programmers in the group are familiar with JAVA and Android mobile application development.
- No overhead costs, team members already possess an Android device.
- Open Source, there are many resources online with help for different types of implementations and applications.

Disadvantages

- App Compatibility, many devices have different size of screen, different types of inner hardware, and different type of processors which can cause issues.
- Fragmentation, many Android users are running old versions of the operating system. This makes it hard to use the latest features and user interface improvements.

Android has many different versions and each year the operating system is updated. Unfortunately, one of the main issues of Android is the fragmentation between all the versions of Android. As of April 2014, Android 4.4 KitKat is the latest version. In order to decide what version of Android to focus on, we referred to Google's April 2014 platform version report as shown below:



Version	Codename	API	Distribution
2.2	Froyo	8	1.1%
2.3.3 - 2.3.7	Gingerbread	10	17.8%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	14.3%
4.1.x	Jelly Bean	16	34.4%
4.2.x		17	18.1%
4.3		18	8.9%
4.4	KitKat	19	5.3%

Figure 3.3.1-1 Android Version Distribution

By looking at the report, it can be concluded that almost all Android devices are running Android 2.3.3 (Gingerbread) or higher. As a result, development will be focused for an application to run on Android 2.3.3 Gingerbread to Android 4.4 KitKat.

Within the application the user will be able to do the following:

- View a live video feed, color and thermal.
- Ability to call 9-1-1 by pressing a button.
- Save a screenshot to their mobile device.
- Control the robot with a Joy Stick using the device's touch screen.

3.3.2 Web Application

In order to support more devices a Web Application can be created instead which can be accessible by any regular computer or mobile device no matter what operating system the device is running. Unfortunately the members of the group are not very familiar and/or comfortable with HTML5. As a result, this option will be visited again if time allows for the implementation and testing of a Web Application.

3.4 Power

Powering the Surveillance Bot can be implemented by using batteries as the power source. Many different options that we came across while doing extensive research on batteries. According to our specifications and requirements, we came up with two different cases in implementing the power supply.

One option was to have a single power source control every subcomponent of the robot. The other scenario is where we have a power source to power each of the subsystems to achieve optimal performance. In order to figure out which of these two scenarios was best for our project we had to determine which type of battery would be most cost efficient and achieve the performance wanted for the vehicle. Performance criterion was based on the requirements for voltage and current we would need for the subsystems. We wanted to be very cautious of trying to meet the requirements for every subsystem and to make sure that the power source achieved maximum efficiency. Some of the key components we are looking for in an ideal battery are as follows:

3.4.1 Discharge Rate

Discharge rate is the rate that the current can flow from the battery. This is a very important aspect for our project because we need to make sure the battery selected is able to deliver high current burst for short periods of time. In our projects we are dealing with motion and may need to go from slow to fast or vice versa. Our battery needs to be able to support every scenario at any given time.

3.4.2 Memory Effect

Memory effect applies to the rechargeable batteries. When memory effect happens the batteries tend to lose their maximum energy capacity, if they are repeatedly recharged after being partially discharged. The battery will remember the smaller capacity. A common process is voltage depression. This is where you repeatedly keep overcharging the battery and this causes the battery discharge quickly. This is important to our project because of the fact we will want to use a rechargeable battery. As a result, we would want to know if memory effect is a factor of the battery when we have to recharge.

3.4.3 Nominal Voltage

Nominal voltage is set by the manufacturer as a recommendation. This voltage is the recommended operating voltage when using the battery and serves as a reference. The reason why this reference is important is because in each of our subsystems we will have a specific required voltage range. We have a maximum voltage and minimum voltage. So if in a particular component a voltage of 5V is required to operate, any voltage under that will cause conflict and cause the subsystem to not function properly. We need to make sure the voltage of our power supply can maintain the constant voltage needed over a timespan. This will ensure that all of the vehicles sub components are able to run at maximum efficiency and properly.

3.4.4 Capacity

Capacity is the amount of charge that can be used from the battery. Capacity is calculated by multiplying the discharge current by the time required to reach max voltage. So a battery’s capacity depends on the discharge rate. The lower the discharge rate the higher the capacity. This is important in making our selection in our battery because this is telling us the amount of power a battery can deliver over a time span. For an example a battery with a rating of 6C, means the battery delivers 6 amp of current for one hour.

3.4.5 Battery Types

Alkaline Batteries - Alkaline batteries are dependent upon the reaction of zinc and manganese. Alkaline batteries have higher energy density and a long shell life. There are two types of alkaline batteries; rechargeable and non-rechargeable. Alkaline batteries are usually disposed after all the stored power is used but in some cases they are rechargeable alkaline batteries and can be reused. One main pro of alkaline battery is the cost of the batteries. With that being said, alkaline batteries tend to be a lot cheaper at the time of purchase but tend to actually be a lot more expensive over time because you have to keep replacing them.

Even though rechargeable alkaline batteries can be a potential solution to the problem of constantly having to replace batteries, the disadvantage is that after each recharge cycle the voltage and efficiency of the battery declines. Also most of the rechargeable alkaline batteries require a special charger to recharge the batteries. These special chargers tend to be a little more expensive and will add to our cost.

The voltages of most alkaline batteries are 1.5 V for a fresh battery out of the pack. The voltage tends to decline as time increases. Voltage can be maximized by using a series of batteries. The main uses of alkaline batteries are for our everyday items such as digital cameras, remote controllers, and radios. The average life cycle for these devices using alkaline batteries tends to be 7 hours.

Nickel Cadmium - Nickel Cadmium is also a rechargeable battery. These were one of the most popular batteries before lithium ion came out and replaced them. The Nominal voltage for a cell of Nickel Cadmium is 1.2V just as NiMH. These batteries have a wide range of sizes and capacities. They tend to offer a very high discharge rate. These batteries suffer from memory effect meaning that they will at some point suffer a sudden drop in voltage. Some pros of nickel Cadmium is that they are very difficult to damage meaning they tend to discharge for longer time spans. The cost for these batteries is moderate. They are not the cheapest batteries but depending on the size and weight they are most ideal for our robot. The capacity of the Nickel Cadmium battery is hardly affected by the discharge of current of the battery.

Nickel Metal Hydride (NiMH) - These batteries have high capacities and are a type of rechargeable battery. Compared to the alkaline battery, NiMH voltage is lower being recorded at 1.2V. NiMH tends to have a high discharge rate and also do not store well over long periods of time as seen in Figure 3.4.5-1 below. NiMH batteries tend to hold the 1.2V charge longer, while discharging, than an alkaline battery. Just like any other battery, they will lose charge at some point but the performance over time won't decrease as fast.

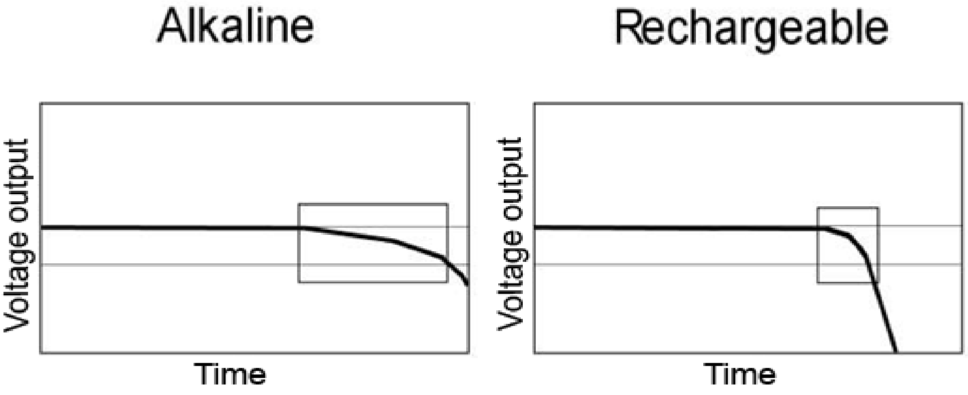


Figure 3.4.5-1 Alkaline vs. NiMH Batteries (Reprinted with permission from Turtle Beach)

The memory loss for these batteries is fairly good. They maintain full power all the way until the battery's life drops off. In the figure above it shows the decline in voltage over time. Nickel metal hydride batteries can keep a constant voltage until that drop occurs. They are very small and cost more than the alkaline batteries. On the downside, these batteries take a very long time to recharge. They can sometimes take up to 10 hours just to recharge.

Lithium-Ion: This is one of the batteries that are rechargeable. These batteries are one of the most popular batteries on the markets today for electronics. They are known for one of the highest energy densities for its weight and slow loss of charge and memory effect. They tend to be one of the most efficient in the amount of current they produce and have been very effective in today's market. Lithium-Ion batteries are one of the lightest of the batteries on the market. This is a major advantage for us because we wouldn't have to worry about the battery getting in the way and causing problems with speed or too much weight. Since these are one of the most popular batteries on the market due to its high demand and advantages, they are more expensive than most batteries depending on the kind of battery you get. Another disadvantage of the lithium battery is that it's very fragile and needs a protection circuit to maintain safe operation. They don't possess any memory effect. The nominal voltage of lithium-ion is 3.7 V which is extremely higher than the alkaline 1.5V or NiMH of 1.2V. Figure 3.4.5-2 below describes the Capacity for Nickel Cadmium, Nickel Metal Hydride, and Lithium- Ion in reference to temperature. As you can see Lithium-ion (yellow) has a constant capacity as temperature increases. This relates well for the vehicle because lithium ion battery cuts down a lot of error with a constant capacity.

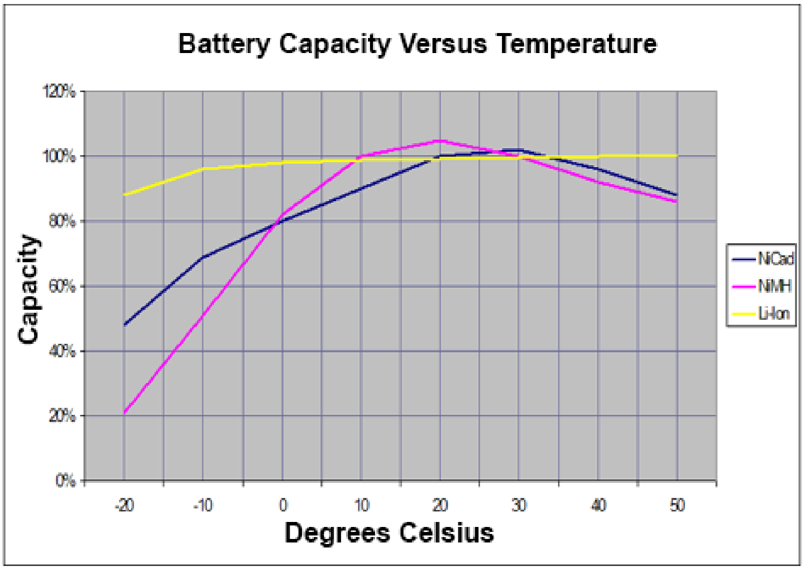


Figure 3.4.5-2 Capacity vs. Temperature (Reprinted with permission from Battery Data)

Lithium Polymer: Lithium Polymer (LiPo) is another rechargeable type of battery. The nominal voltage is 3.7V (just as the lithium-ion battery). The Lithium polymer was evolved from the lithium-ion battery and shared a lot of the same characteristics as the lithium ion battery they are very lightweight just like the lithium batteries. LiPo's rate of discharge are lower meaning they have a higher capacity. In terms of life cycle, LiPo do not have a longer life cycle than lithium-ion. They also can be quite dangerous if you over charge. The price for lithium polymer is more than the lithium battery so they are fairly expensive.

3.5 Voltage Regulator

The vehicle has various subcomponents that will require a certain level of voltage to run efficiently. In doing so, we need to accurately send the right amount of voltage to each subsystem. We do not want to send below the amount needed because the component can malfunction and we also don't want to send too much because that may cause overheating in the subcomponent and cause the system to fail. In order to achieve the desired voltage we want we will need a voltage regulator. There are several different voltage regulators that we can choose from: Linear Regulator, Switching Voltage Regulator, and Zener Diode Regulator.

3.5.1 Linear Regulator

A linear regulator is one of the ways in which to control a voltage that you want to use. These regulators are very easy and very inexpensive to execute. They are usually used for low voltage and low power systems. Linear Regulators tend to work by adjusting the series resistance by a feedback loop. A lot of the regulators on the market are very basic and simple due to the basic circuitry needed when executing them. The figure below is a basic linear regulator, which uses a Voltage-Controlled Current Source to force a fixed voltage amount at the end of the load. As the output voltage continuously goes up and down, the sense/control circuitry watches the output voltage and adjusts the current controlled source to maintain the desired voltage wanted through a mere feedback loop.

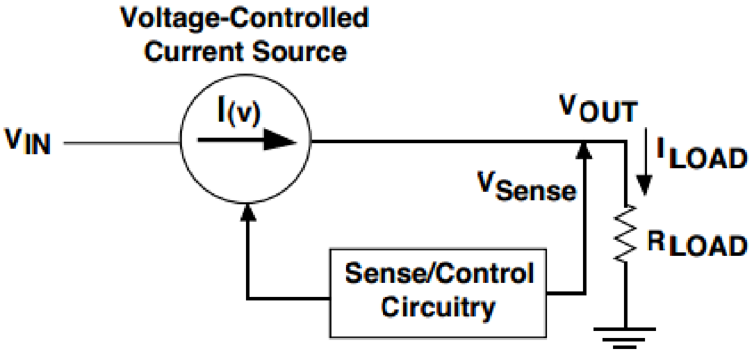


Figure 3.5.1-1: Linear Regulator Circuit (Reprinted with permission from Texas Instrument)

Most linear voltage regulators are able to work for various systems to produce the desired output you want. But a lot of the voltage regulators are not very efficient when it comes to trying to execute a very large amount of voltage. A lot of the linear regulators suffer from dropout voltage. Dropout voltage is the minimum voltage drop required from the regulator to maintain the desired output voltage wanted. Linear regulators are designed to be reliable and maintain a safe minimum voltage. An example of this would be if a linear regulator had a large voltage drop of 3V and you need your output voltage to be 8V. You would want to be producing 11 volts for the input voltage to account for the dropout voltage in the linear regulator

3.5.2 Switching Regulators

Switching Regulators are the most popular of the type of voltage regulators used today. This type of regulator is known as one of the more efficient type of voltage regulators and tends to be a little more expensive. The switching regulator doesn't take the voltage all at once. It takes the input voltage source in pieces and sends the voltage to the output. Figure 3.5.1-2 is an example of a schematic of a switcher circuit. An electrical switch is in place, which alters the voltage being sent out. This is better known as the switches duty cycle that sets the amount of energy transferred. The switching regulators are also known to be able to produce higher output voltages than the voltage sent into the regulator.

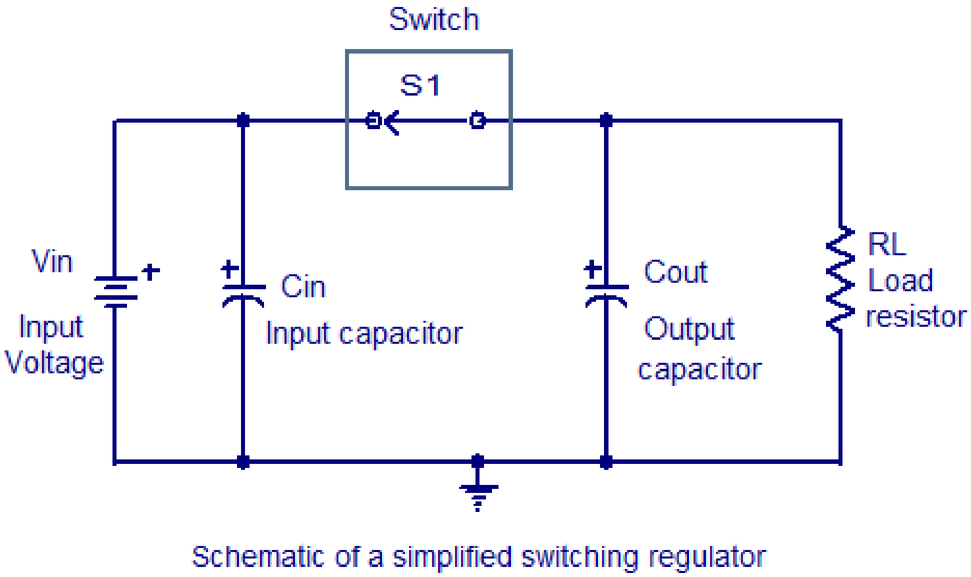


Figure 3.5.1-2 Simplified switching circuit (Reprinted with permission form Circuits Today)

As you can see from the figure above, S1 (switch) when closed the output capacitor is charged. When the switch is open the output capacitor discharges. The output voltage relies on the input voltage and the duty cycle that the switch controls. In reference with our project the switching regulator would be very beneficial due to the fact it's very dependable. We don't necessarily have very high voltage to deal with but in comparison with the linear regulator the switcher regulator is able to have a fixed voltage set.

3.5.3 Zener Diode

The Zener Diode can also be used as a type of voltage regulator. These type of regulators run by the use of a diode. Diodes are a very useful way to control voltages. In the figure below, we see that this is an example of a Zener Diode circuit. There is a resistor in series with the diode. This series connection controls the current flow to the diode. The voltage of the Zener diode is taken across the diode as seen below.

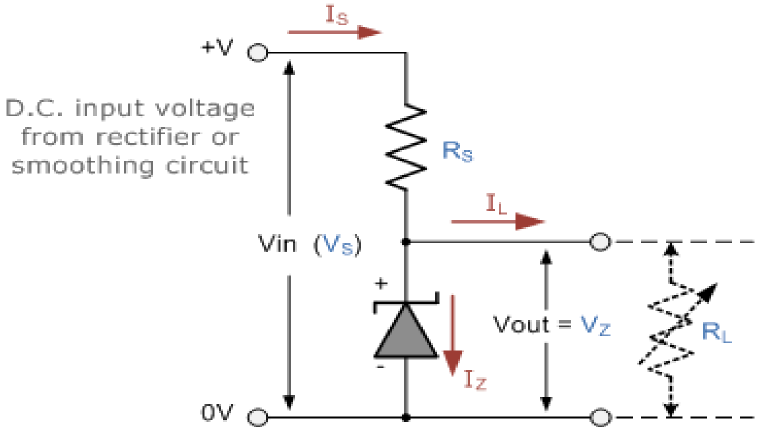


Figure 3.5.3-1 Zener Diode Circuit (Reprinted with permission from Wayne-Electronics)

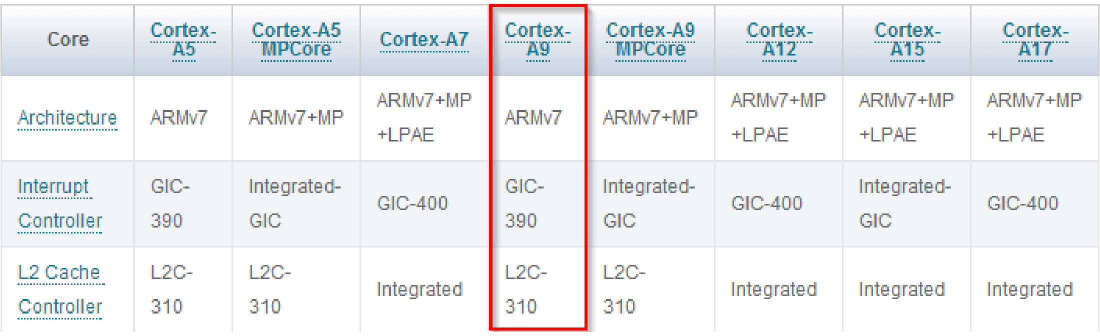
The advantage of using this type of regulator is that you can control the voltage to make sure it doesn't exceed a certain voltage output. As stated above this type of regulator controls how much current is being delivered to the load. Another advantage is that this type of regulator is the cheapest out of all the different types of regulators do to the Zener diodes being relatively cheap. The problem with using this type of regulator is that you have to be very careful with the amount of voltage coming in because it's very hard in trying to control dc current. When trying to regulate a constant voltage there tends to be a lot of electrical noise.

3.6 Microcontroller

ARM Holdings is a semiconductor manufacturer that issues and license ARM core architecture. It is currently an industry leader for a unit that houses a complex operating system. This allows more fluid process of development for engineers to design and implement a smart device. The flexibility of the architectural design allows a wide range of applications from a simple dedicated embedded system to complex computation analysis capability.

The vast spectrum of availability of ARM processors requires more extensive specification and requirement processes. The processors are organized by various categories such as voltage requirement, memory implementation, processing power, etc. Picking the right processor with undershooting the needs and avoiding overshooting the necessary requirements can be very tedious. The chips are named under the series of “Cortex” and the Cortex of interest is the A series. Note that the top factors included in the consideration for picking the right part for this project are cost and computing power.

Cortex A- The majority of Cortex A series is an extension of ARMv7A architecture. They are categorized based on computing capability and power efficiency. Also Cortex-A8 and Cortex-A9 share the same architecture. The breakdown is shown in the figure below.



Core	Cortex-A5	Cortex-A5 MPCore	Cortex-A7	Cortex-A9	Cortex-A9 MPCore	Cortex-A12	Cortex-A15	Cortex-A17
Architecture	ARMv7	ARMv7+MP	ARMv7+MP +LPAE	ARMv7	ARMv7+MP	ARMv7+MP +LPAE	ARMv7+MP +LPAE	ARMv7+MP +LPAE
Interrupt Controller	GIC-390	Integrated-GIC	GIC-400	GIC-390	Integrated-GIC	GIC-400	Integrated-GIC	GIC-400
L2 Cache Controller	L2C-310	L2C-310	Integrated	L2C-310	L2C-310	Integrated	Integrated	Integrated

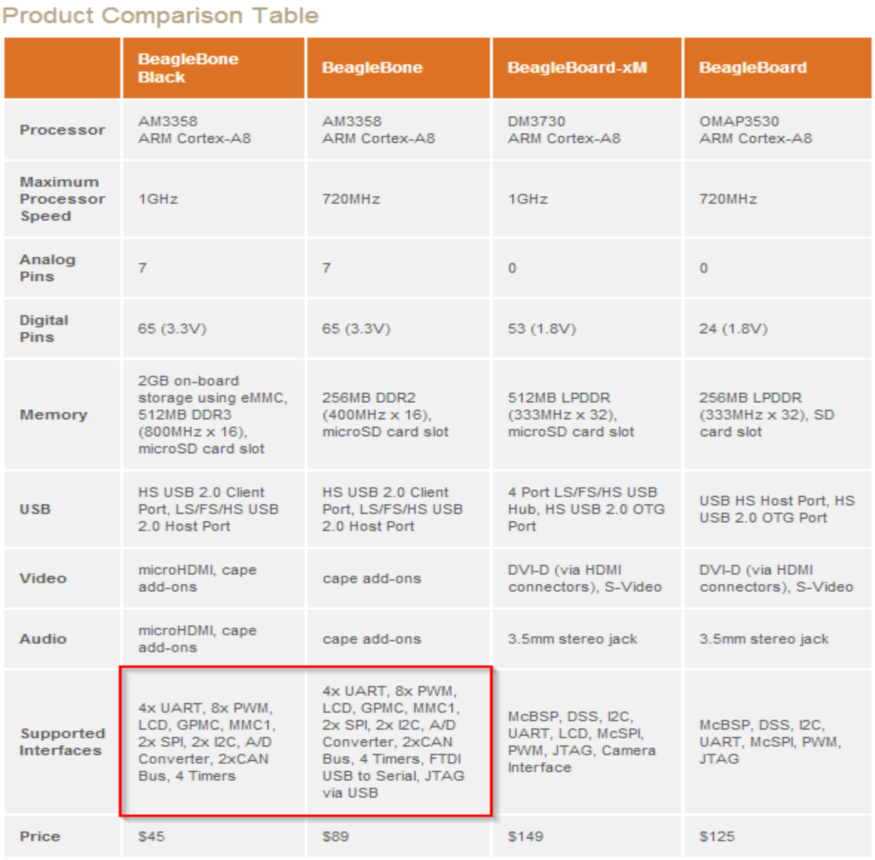
Figure 3.6-1 Cortex A Specifications

First thing that we look at is availability of the chip. ARM Holdings has multiple semiconductor partners throughout the variety of their products. Each individual manufacturer also has their own unique implementations on their own design. The vast market for ARM processor has become so large that each of these unique designs actually serves its own benefit in their own niche within the market. Texas Instruments seems to be the most convenient for this particular project due the existing developing tools and hardware knowledge of developers on the team. The Cortex-A8, the chip of interest, is licensed to the companies such as Texas Instruments, Samsung, Freescale, and Broadcom. However, the TI seems to cater the most to the developers with all the development tools.

Texas Instrument SITARA- Sitara AM335X is designed for products that implement graphic functionality for user friendly purposes. With this particular board, the interface design will also be more fluid and easier to customize. One of the most important features on this chip is the computing power. The AM335X is range all the way from 300MHz – 1GHz which gives us more freedom on the operation that the device will be performing. The AM3358 particularly are available in 600MHz, 800MHz, and 1GHz. The Cortex-A is considered one of the more operating system friendly chips in the market. AM335X supports OS such as Linux, Android, and Windows Embedded CE; allowing the richer resource environment for tasks execution.

SITARA Design Kits & Evaluation Modules- There are a few viable options for development environments that are available as a starter kit. The design tools are available from affordable range to a professional standard environment. From a beginners stand point, Sitara from Texas Instruments is the best product to learn how to design an embedded machine. All the options that are selected for discussed in this document are purely considered based on cost.

BeagleBone Black - BeagleBone.org Foundation is a non-profit organization founded by enthusiast engineers and former Texas Instruments employees. The purpose of this group is dedicated to the learning of the science of embedded hardware. The results of their effort are development boards such as BeagleBoard, BeagleBone, BeagleBone Black, and BeagleBone-xM. The BeagleBoard and the xM seem to have a design agenda. The two main boards recommended by TI are BeagleBone (regular) and BeagleBone Black. The two boards are very much the same board except the difference in availability of interfaces (and also prices). The main components that are available in BeagleBone are USB and the JTAG, which seem very convenient if available to utilize. However, BeagleBone is $89 but the BeagleBone Black is $45 from the website. The BeagleBone Foundation is definitely one very good option for a development environment considering all boards use ARM Cortex-A8 architect. Here is the boards' side by side comparison:



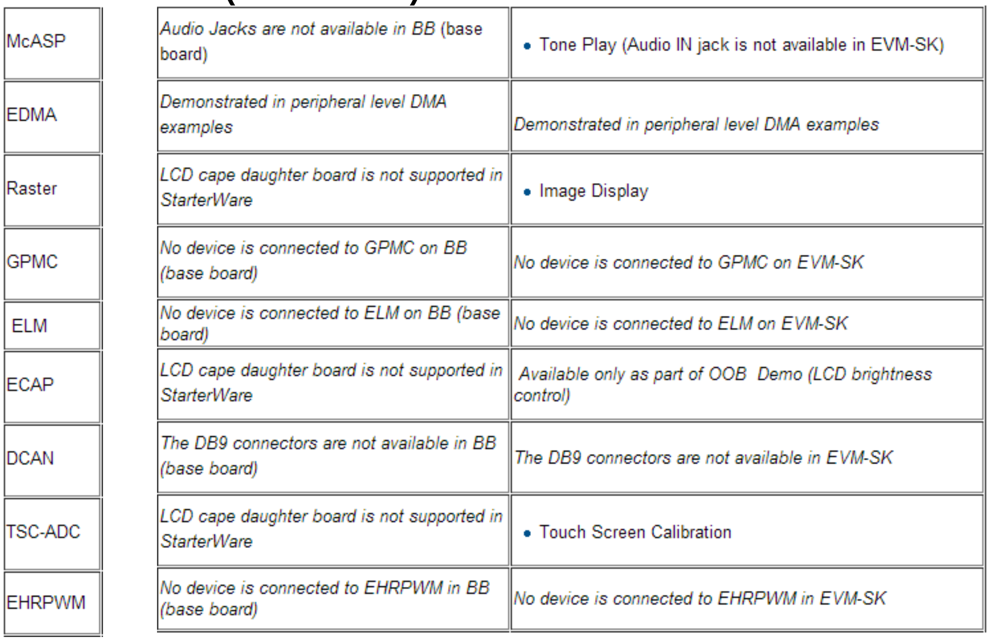
Product Comparison Table

	BeagleBone Black	BeagleBone	BeagleBoard-xM	BeagleBoard
Processor	AM3358 ARM Cortex-A8	AM3358 ARM Cortex-A8	DM3730 ARM Cortex-A8	OMAP3530 ARM Cortex-A8
Maximum Processor Speed	1GHz	720MHz	1GHz	720MHz
Analog Pins	7	7	0	0
Digital Pins	65 (3.3V)	65 (3.3V)	53 (1.8V)	24 (1.8V)
Memory	2GB on-board storage using eMMC, 512MB DDR3 (800MHz x 16), microSD card slot	256MB DDR2 (400MHz x 16), microSD card slot	512MB LPDDR (333MHz x 32), microSD card slot	256MB LPDDR (333MHz x 32), SD card slot
USB	HS USB 2.0 Client Port, LS/FS/HS USB 2.0 Host Port	HS USB 2.0 Client Port, LS/FS/HS USB 2.0 Host Port	4 Port LS/FS/HS USB Hub, HS USB 2.0 OTG Port	USB HS Host Port, HS USB 2.0 OTG Port
Video	microHDMI, cape add-ons	cape add-ons	DVI-D (via HDMI connectors), S-Video	DVI-D (via HDMI connectors), S-Video
Audio	microHDMI, cape add-ons	cape add-ons	3.5mm stereo jack	3.5mm stereo jack
Supported Interfaces	4x UART, 8x PWM, LCD, GPMC, MMC1, 2x SPI, 2x I2C, A/D Converter, 2xCAN Bus, 4 Timers	4x UART, 8x PWM, LCD, GPMC, MMC1, 2x SPI, 2x I2C, A/D Converter, 2xCAN Bus, 4 Timers, FTDI USB to Serial, JTAG via USB	McBSP, DSS, I2C, UART, LCD, McSPI, PWM, JTAG, Camera Interface	McBSP, DSS, I2C, UART, McSPI, PWM, JTAG
Price	$45	$89	$149	$125

Figure 3.6-2 BeagleBone board comparison

TI AM335x Starter Kit- the EVM-SK (another name) comes fully loaded and even more equipped with multiple interfaces than the BeagleBoard. The starter kit is practically a prototype of a mobile device that is ready for an operating system. One of the components that the kit offers is the onboard WLAN device which will be needed for the design. The kit is even equipped with an LCD screen to access the start GUI of the chosen OS. This is most convenient feature of the starter kit. If implementing with TI StarterWare; the development process will be much more fluid. The StarterWare allows the chips to operate without an operating system. Ultimately this is an ideal environment for the design if all is goes well for the project. Although the Starter Kit is priced at $199 from the TI web store, it could be a good investment.

Essentially both options are as effective as each other, the only factors that are left to considered is the bonus and fancy features that are available in the EVM-SK. The kit seems to have a few more convenient features for the developers. This allows more freedom for us to when considering the possibility of the projects versus having to implement each individual extra feature that are not available at hands from the beginning. This time, saving reason is a big factor of choosing the right developing environment for our project and should we go the wrong route it might be less headache to spend a little bit more in the very beginning than having to stop the flow of the developing process in definite incoming unforeseen obstacles. The following figure compares those minor features between the two.



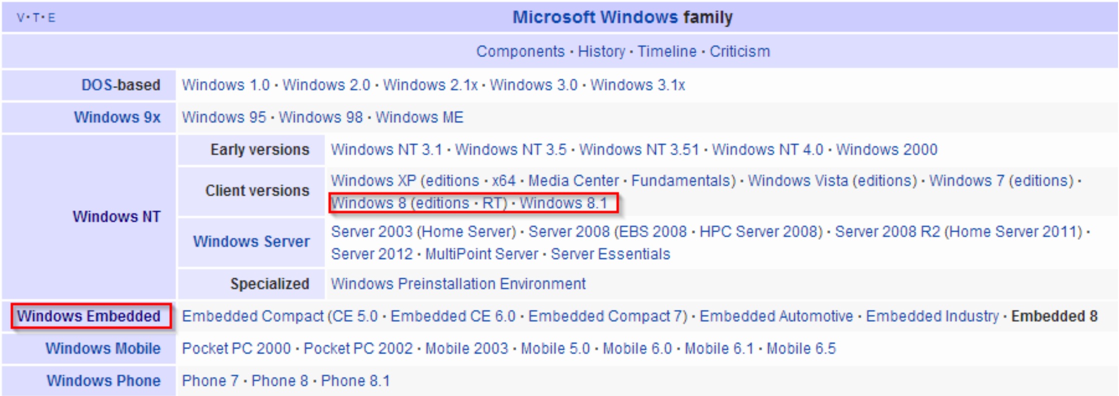
Features	BeagleBone (Base board)	EVM-SK
McASP	*Audio Jacks are not available in BB (base board)*	• Tone Play (Audio IN jack is not available in EVM-SK)
EDMA	*Demonstrated in peripheral level DMA examples*	*Demonstrated in peripheral level DMA examples*
Raster	*LCD cape daughter board is not supported in StarterWare*	• Image Display
GPMC	*No device is connected to GPMC on BB (base board)*	*No device is connected to GPMC on EVM-SK*
ELM	*No device is connected to ELM on BB (base board)*	*No device is connected to ELM on EVM-SK*
ECAP	*LCD cape daughter board is not supported in StarterWare*	*Available only as part of OOB Demo (LCD brightness control)*
DCAN	*The DB9 connectors are not available in BB (base board)*	*The DB9 connectors are not available in EVM-SK*
TSC-ADC	*LCD cape daughter board is not supported in StarterWare*	• Touch Screen Calibration
EHRPWM	*No device is connected to EHRPWM in BB (base board)*	*No device is connected to EHRPWM in EVM-SK*

Figure 3.6-3 Comparison between BeagleBone and EVM-SK

3.6.2 Operating Systems

Windows Embedded- Microsoft is currently the biggest distributor of operating system in the world. All of us know and have used Windows throughout our college career. Not only Windows is available in the regular mass consumer format but it is available in the dedicated machine format as well. The Windows Embedded is designed to be used in smaller machine and less generalized computers. Typically we would see Windows Embedded in action at bars and restaurants with their Point of sale (POS) machines.

The Windows operating systems are separated in multiple groups within the family such as 9x, NT, Embedded, etc. The following figure shows the layout of the Windows product line and the classification of each one of them.



V·T·E		Microsoft Windows family
		Components · History · Timeline · Criticism
DOS-based		Windows 1.0 · Windows 2.0 · Windows 2.1x · Windows 3.0 · Windows 3.1x
Windows 9x		Windows 95 · Windows 98 · Windows ME
Windows NT	Early versions	Windows NT 3.1 · Windows NT 3.5 · Windows NT 3.51 · Windows NT 4.0 · Windows 2000
	Client versions	Windows XP (editions · x64 · Media Center · Fundamentals) · Windows Vista (editions) · Windows 7 (editions) · Windows 8 (editions · RT) · Windows 8.1
	Windows Server	Server 2003 (Home Server) · Server 2008 (EBS 2008 · HPC Server 2008) · Server 2008 R2 (Home Server 2011) · Server 2012 · MultiPoint Server · Server Essentials
	Specialized	Windows Preinstallation Environment
Windows Embedded		Embedded Compact (CE 5.0 · Embedded CE 6.0 · Embedded Compact 7) · Embedded Automotive · Embedded Industry · **Embedded 8**
Windows Mobile		Pocket PC 2000 · Pocket PC 2002 · Mobile 2003 · Mobile 5.0 · Mobile 6.0 · Mobile 6.1 · Mobile 6.5
Windows Phone		Phone 7 · Phone 8 · Phone 8.1

Figure 3.6-4 Family of Microsoft Operating Systems

We can observe that the version that it falls into is the Windows NT group, and that “Embedded” is a separate category of Windows entirely. Also within the embedded category, we can see that there are quite a few series of product available for developers. Embedded 8 is the most current version and the most updated with latest technology. The Compact series and Embedded 8 are compatible with ARM architecture, although it is most seen on the x86 architect. Series of ARM based devices have been showcased using Windows Embedded and successfully demonstrated the richness of the performance.

A company called Adeneo Embedded integrates popular operating systems and embedded machines to product end consumer’s products. One of the projects that the company has available to the public is the integration of Windows Embedded Compact 7 and the BeagleBone Board. AE has the binary-image of the operating system ready for used and it is available for free on its website. The device will be turned on and ready to be operated just like a regular computer except embedded version has less features than a regular version.

Arch Linux | ARM- ALA is an extended branch of Arch Linux operating system. It is aimed toward ARM processor such as ARMv5, ARMv6 and ARMv7. Also ALA is another operating system that is built to fit for small machines and requires less computing resources from the device. The original Arch Linux was built for the x86 and x64 chips but later the ARM version of the OS was developed and released as open source. In general Arch Linux is a light weight, and UNIX like environment system. The idea is not to cater to a general purpose usage but a minimal environment machine, therefore the GUI is typically stripped from the system. The idea of Arch Linux is to be operated on a dedicated machine where it executes chosen finite tasks better than a regular computer.

The interface is basically just like any other Linux interface. For someone who is very familiar with the Linux already, Arch Linux|ARM is only few minor ticks different than the rest. Most of the learning curve will lie within the getting acquainted with the list of fancy features that might be less available due to its simplicity idea.

Technically, any other flavor of Linux is adaptable with the intended functionality for the particular purpose in the document. Although, Arch Linux|ARM may or may not be the most popular choice in the market, there are some pros and cons in comparison against other distributions. Note that the systems discussed were the ones that are closely related and relevant to Arch Linux.

- Debian GNU/LINUX
    - Debian does not offer ports system.
    - Arch is a rolling release, but the Debian is stable.
    - Both can be stripped of undesired packages; Arch is minimal base, but Debian is a pre-selected packages. Both systems have a reliable package management system.
- Fedora
    - Fedora does not feature ports system.
    - Both are considered as bleeding edge release; Fedora is a test bed but Arch is not.
    - Arch development main focus is minimalistic (highly suitable for the project) but Fedora focus on free software.
- Ubuntu
    - Ubuntu does not offer ports system.
    - Ubuntu focus heavily on user friendly, while Arch focus on minimalist design (Although Ubuntu is seen on a lot of embedded project).
    - Ubuntu community is much larger which could mean a better chance of resolving an issue on the system.

TI-StarterWare- On top of all other options for stripped down version of conventional operating systems, Texas Instrument offers even more bare back software for the ARM processor. The StarterWare is embedded software dedicated to the TI chips available in various versions of the ARM chips including the processor of our interest, Sitara. It offers a direct C based API to the chip and also includes stacks to enable other peripherals for the ease of functionality. With all that being said and done, StarterWare seems to be a better fit for an absolute dedicated task oriented device, as far as development time, when it does not involve too much in high level complex data. In many different ways, StarterWare offers the leanest and the most direct control over the supported devices. While a large C library is available for managing components such as Ethernet, USB, Graphical and etc., it is considered as a “no-OS” developing environment.

3.7 Sensor

There are many obstacles to consider when maneuvering in household environments. In order to successfully track a person through this environment we will need a sensor array around the perimeter of the vehicle to detect obstacles and allow the vehicle to make course corrections to avoid them so it can continue to its tracking mission.

Tactile sensors- These are sensors that would require physical contact with objects. Pressure from contact with the object would then be translated by some kind of sensor which would relay the information of an objects presence.

Mechanical Switches- Using metal arms attached to a spring loaded mechanical switches (shown in figure 3.7.1-1). This method is the simplest and lowest tech option. However it is also the slowest and most imprecise method of obstacle detection. The vehicle would need to be far too close to make hard enough contact with the objet to throw the switch, at which point it would need to reverse away from the object before trying a new course correction.

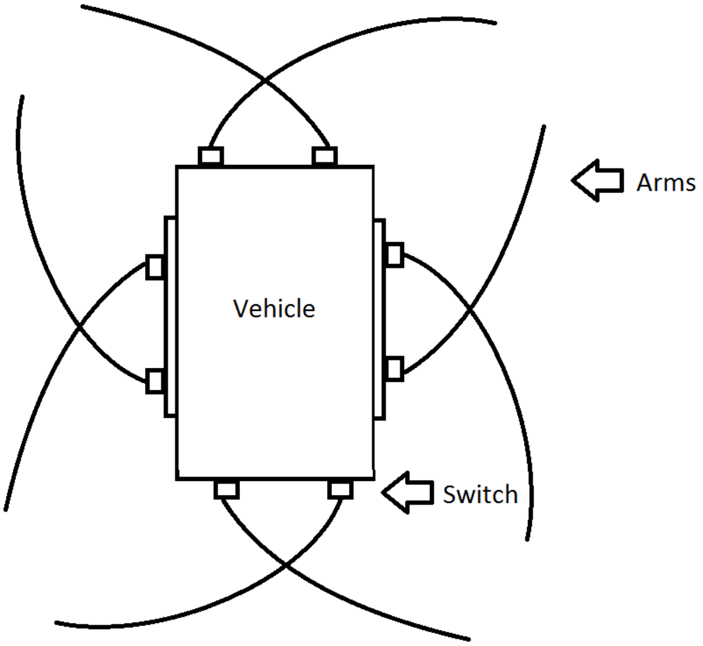


Figure 3.7.1-1 Mechanical switch sensor vehicle diagram

Piezoelectric pressure sensors- Similar to the Mechanical switch sensor system we can use a piezoelectric material in place of the mechanical switch that causes a voltage spike whenever pressure is applied to the material. With this system we could detect objects using less pressure than would be required for the mechanical switch to detect object, however physical contact is still required which again would slow down the procedure of obstacle avoidance as the vehicle would have to reverse out of contact with the object before trying applying a course correction.

Non-tactile- These are sensors that do not require physical contact to detect objects. They typically use a source emitter and a detector for reflections of the source on the environment (Shown in figure 3.7.1-2). Because no physical contact is required to detect objects these sensors could apply course corrections without needing to reverse away from an object, which would cut down on time spent navigating around objects and allow the vehicle to proceed with its mission of tracking and following heat signatures with less delay.

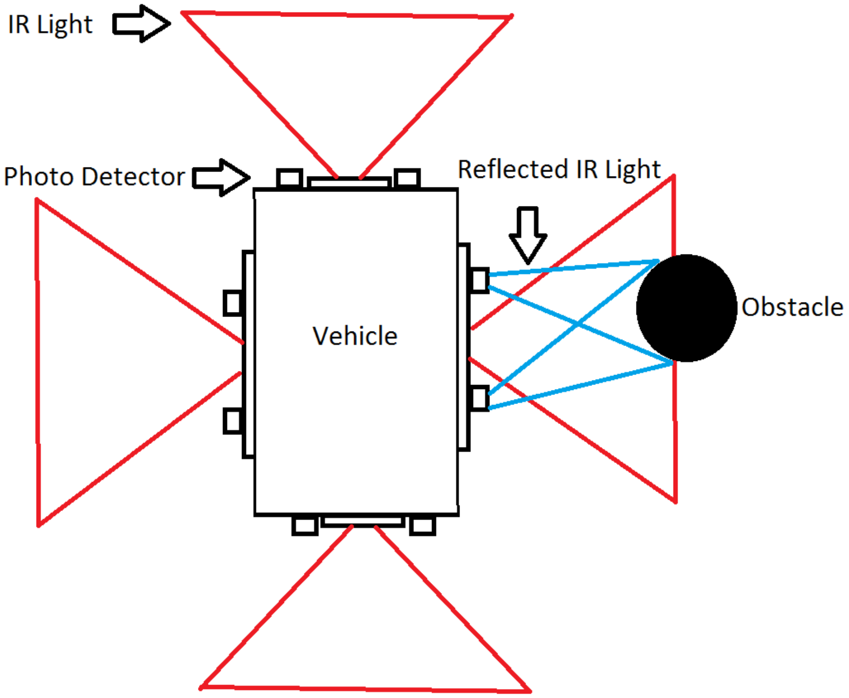


Figure 3.7.1-2 Proximity IR vehicle diagram

IR Sensor-This method would require an IR light array, which would broadcast IR light in the sensing areas. And an IR light sensor, which would detect reflections of the broadcasted light. The intensity of the reflected light is then used to determine proximity with an object as shown in Figure 3.7.1-3 for the VCNL3020 Proximity Sensor w/ Infrared Emitter.

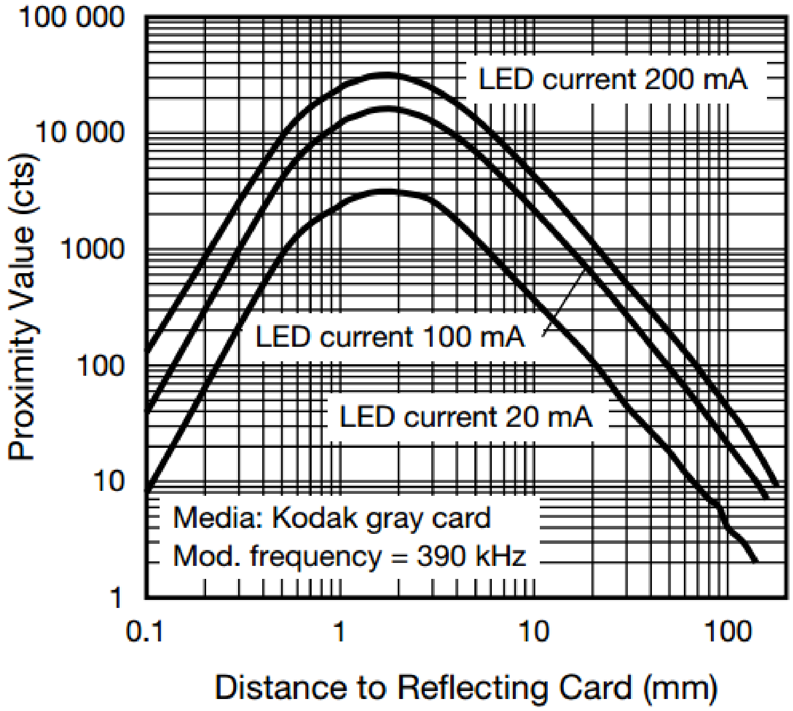


Figure 3.7.1-3 Proximity Value vs. Distance **(‘**Courtesy of Vishay Intertechnology’)

Sonic- This method uses ultrasonic sound waves to detect objects. Sound is emitted from a source and the reflections of the sound waves off of objects are picked up by receivers. Depending on the location of and number of sensors we can calculate the distance from the object as well as the shape of it (to some degree of accuracy depending on the number of sensors).

Ultrasonic vs. IR- The two devices we have chosen to compare are the HC-SR04 Ultrasonic Ranging Module, and the VCNL3020 Proximity Sensor w/
Infrared Emitter. Table 3.7.1-1 was made from each item’s data sheet.

	HC-SR04 (Ultrasonic)	VCNL3020 (IR)
Operating Range (mm)	20-4000	1-200
Measuring Angle	≈ 30°	≈ 40°
Measurements per sec	1 - 16	2 - 250
Operating Voltage range (V)	4.5 - 5.5	2.5 – 3.6
Operating current (mA)	10 - 20	10 – 200
Price per unit (USD)	≈ 1.50	≈ 3.20

Table 3.7.1-1 Comparison of HC-SR04 and VCNL3020

We can see clearly that the Ultrasonic sensor has a much larger operating range compared to the IR sensor, this is a fairly large advantage as it allows us a greater distance to perform course correcting procedures to avoid the obstacle. The measuring angle of the IR sensor is slightly larger; however the difference is not large enough to have much impact on our final decision. The IR has a clear advantage in measurements per sec due to the speed of light being so much greater than the speed of sound, however the extra measurements would just be added processes for the CPU to handle and ultimately unnecessary for our purpose. We also see that the IR system consumes more power, and has a higher cost. However, the biggest downfall of using the IR sensor is that the light it emits encompasses the same spectrum of light that we will be using for the detection of thermal bodies in the tracking system for the vehicle. Using these sensors could cause difficulties with the tracking functionality of the thermal camera by washing out the surrounding area with unwanted IR light.

3.8 Wireless Device

In general, wireless modem is best and easiest to work with if handled by some sort of operating system (assuming the driver supports that particular OS). The operating system of the interest for this project is Arch Linux|ARM which is simply just another flavor of open source Linux that runs with very limited graphical user interface. It is designed to be used for embedded devices. Linux is notorious for being picky about hardware compatibility. Arch Linux seems to have a very well documented list of hardware that it supports. The choice, however, is very limited according to the Wiki site.

EDIMAX EW-7811Un- A very inexpensive USB wireless adapter from EDIMAX is just about 0.28" x 0.59" x 0.73" in dimensions and very convenient to install and connected serially to the device. The driver also supports our version of Linux. There are many trails and errors documented on the open source documentation. It’s very fortunate that others that went through the hardship of setting it up also care enough to share the knowledge. Below are some relevant specifications:

Standards	IEEE 802.11b/g/n
Data Rates	Up to 150Mbps
Frequency Band	2.4 GHz – 2.4835 GHz
LEDs*	Link/Activity
System Requirements	Windows 7/Vista/XP, Mac OS, Linux
Warranty(Parts/Labor)	2 Years Limited

Table 3.8-1 EDIMAX Specifications

The IEEE universal standard helps take care of the headache for us. The data rate is the one we care most about since it the robots will be communicating in real-time. We have test UDP application of this data rate and it seems very smooth. The speed of the device should be plenty. The LEDs feature will offer a piece of mind to the developing process. It would be nice to know just from looking if the device is turned on or not. As far as system requirements, Linux is supported however it is suggested to get an updated version of the driver instead of the standard from the CD. It might be silly to care about warranty but this project is under a tight budget. A 2 year period warranty will definitely cover our timeline.

LOGIC SUPPLY UWN100/UWN200- Another choice for wireless adaptor that is compatible with open source Linux comes in two trims, UWN100 and UWN200. They both use MediaTek MT7601 for a controller unit and also both considered as generic Wi-Fi adaptor. LOGIC SUPPLY manufactured these two with the only difference that the UWN200 comes with an antenna to extend the reach. The unit is very small and is about the size of the USB head connector with the dimension of 0.5" x .25" x 0.7" without the antenna. For the UWN200 the head is exactly the same size with the standard SMA antenna extended on the top with could be an issue if compactness is a big factor. Note that both are identical except for the antenna, so here’s the specification of both.

Standards	IEEE 802.11b/g/n
Data Rates	Up to 450 Mbps
Frequency Band	2.4 GHz – 2.4835 GHz
LEDs*	n/a
System Requirements	Windows, Mac, Linux
Warranty(Parts/Labor)	2 Years Limited

Table 3.8-2 UWN100 Specifications

Another selling point for one of these is the large number of recommendations from users in the BeagleBone community. There are already multiple different tutorials written on the forum available for any beginner to grab. Also a very big factor for going with this Wi-Fi device is the data rates. The specs from the manufacturer promises up to 450 Mbps which is more than most devices we’ve encounter in the same price range.

The device does not have any LEDs on board which is a bit of a drawback when working on the embedded level. It might be minor thing to complain about, but as mentioned earlier, the blinking light offer the developers a piece of mind so they can quickly confirm if the device is somewhat working. However, it is not a deciding factor if LEDs were not there.

3.9 Cameras

Our robot will be equipped with two cameras; one will be a regular webcam while the other one is going to be a thermal camera. Both of the cameras will be used for different purposes, the thermal camera will be used for tracking while the webcam will be used for object detection, video streaming, and snapshot notification. For the webcam category there are many options widely available and they are very affordable. As for the thermal cameras, it’s a very expensive technology that is geared towards commercial applications. Lately, with the increased popularity of drones the prices of thermal cameras have dropped and there’s a greater amount of options available to regular consumers. Although the options are still costly compare to regular cameras, it’s feasible to acquire one at a reasonable price today. Some of the popular areas where thermal cameras are being utilized are:

- Predictive Maintenance at factories
- Test for stress and fault in Mechanical and Electrical systems
- Firefighting by locating hot spots
- Law Enforcement to protect officers in the field
- Night Vision alternative

3.9.1 Thermal Cameras

Many people confuse Thermal Cameras with Night Vision cameras. Night Vision cameras apply a small amount of physical light so that objects can been seen at night. One of the disadvantages of night vision is that there must be a light source present, such as moonlight in order to work. Different from Night Vision cameras, Thermal Cameras detect the heat (infrared light) given off an object or person without the need of light. In addition, a Thermal Camera may be used as an alternative to a Night Vision camera. The Infrared's wavelengths are outside the visible spectrum to the human naked eye.

The amount of energy in a light wave is related to its wavelength. Longer wavelengths have lower energy while shorter wavelengths have higher energy. Any object with a temperature greater than absolute zero will emit radiation. Thermal Cameras utilize special lenses which allows for the detection of heat and radiation. Glass doesn't transmit radiation well, as a result the lenses of Thermal Cameras are made of Germanium. Germanium is an expensive metal and good transmitter of infrared radiation that can be readily cut and polished into lenses and windows.

Lately, the popularity of drones with thermal imaging for remote control and DIY (Do It Yourself) fanatics have increased. As a result, it has become easier and possible for regular consumers to acquire one. Below are some of the options available to consumers that can be implemented for different types of applications as well as options that we have investigated for the project.

Flir Quark- The Flir Quark is available in two different resolutions: 336 x 256 and 640 x 512. Both of these resolutions have a pixel pitch of 17 micron pixels. It features low power consumption at around 1.0 W and is shock and vibration resistant. Although the robot will be for indoor use, once the robot is in movement it will cause the thermal camera to shake. Using the camera's built-in software is possible to auto-focus the image in order to get a good quality image. The Flir Quark also features a digital zoom of 2x and 4x of the video image. The camera's temperature range is from -40° F to 176° F which is perfect for indoor use since the heat will fall under this range. The video output is analog and is available as NTSC or PAL. The Flir Quark's price ranges from $3,645 to $8,099.00 depending on the resolution. This price doesn't include any development board or accessories required for the camera to board connections.

Flir Tau2- Another option from Flir is the Tau 2. The Flir Tau 2 is available in three different resolutions: 640 x 512, 336 x 256, and 324 x 256. The 640 x 512 and 336 x 256 come with a pixel pitch of 17 micron pixels while the 324 x 256 comes with a pixel pitch of 25 micron pixels. The Flir Tau 2 uses less than 1.0 W of power, lower than the Flir Quark. It also features a rugged design suitable for UAV's (Unmanned Aerial Vehicle) and handheld devices. Digital Zoom is also available in 2x and 4x of the video image by using the built-in software. Similar to the Flir Quark, auto-focus can be accomplished by using the built-in software. The camera's temperature range is from -40° F to 176° F which is perfect for indoor use and the video output is analog available as NTSC or PAL. The Flir Tau2's price ranges from $1,840.00 to $9,100.00 depending on the resolution. This price doesn't include any development board or accessories required for the camera to board connections.

Mikrosens MSE070D- The Mikrosens MSE070D has a resolution of 160 x 240 and comes with a 7.5 to 13.5 pixel pitch. It has an ultra-compact form factor which makes it possible to be used for multiple applications. It utilizes very low power, according to the specifications sheet is at 0.6 mW. There is no mention of digital zoom or image stabilization supported by the MikroSens MSE070D. In addition the temperature range is unknown as it is not specified in the specifications sheets. The video output is digital. There is not a lot of information available at the manufacture's website, because of this we are waiting for more details via email.

ICI 320X IR Core- The ICI 320X IR Core is made by Infrared Cameras Inc. and has a resolution of 320 x 240 with a 7-14 micron pixels pixel pitch. It's very lightweight and consumes less than 5.0 W of power. The ICI 320X IR Core features digital zoom of 2x and 4x of the video image as well as automatic focus. The output is in digital format. One of the drawbacks for our application is that the temperature range since it ranges from 302° F to 1004° F. With these temperature range, this specific model is or suitable for outdoor use or fire. The price for the ICI 320X IR Core starts at $1,600.00.

ICI 7640 P-Series- The ICI 7640 P-Series is also made by Infrared Cameras Inc. but has a higher resolution of 640 x 480 with a pixel pitch of 7-14 micron pixels. It has very low power consumption of less than 1.0 W and can be powered by either a USB port or DC. In contrast to the ICI 320X IR Core, the ICI 7640 P-Series features shock and vibration resistant which will help with the quality of the image. It doesn't support digital zoom but it features automatic focus of the video image. The video output is in Digital format. The temperature range for this model is from -4° F to 122° F which is perfect for indoor applications such as our project. The price for the ICI 7640 P-Series starts at $1,995.00.

Thermoteknix MicroCam2- The Thermoteknix MicroCam2 is available in two different types of resolution: 384 x 288 and 640 x 480. The 384 x 288 resolution is available in two models, one with 25 micron pixels and the other with 17 micron pixels for the pixel pitch. The 640 x 480 comes with a pixel pitch of 17 micron pixels.

It has a low power consumption of less than 0.6 W. In addition, it’s very light weight and features high shock resistance. The MicroCam2’s built-in software allows it to have automatic focus and digital zoom of 2x and 4x of the video image. The video output is analog available as PAL or NTSC. The temperature range is between -20° F and 140° F which works perfectly for indoor application. The price is quite high in the $5,000.00 and $6,000.00 range depending on the resolution chosen.

Sofradir-EC Core Atom 80- The Sofradir-EC Core Atom 80 has a smaller resolution than all the other options compared for this project. It has a resolution of 80 x 80 and a pixel pitch of 8-14 micron pixels. The Sofradir-EC Core Atom 80 has a low power consumption of less than 0.25 W and it can be powered by a USB 2.0 port. It utilizes a very light and compact design making it possible to be used for different applications. The camera must be setup with its proprietary software in a Windows computer first. In the software, the focus and digital zoom settings can be enabled and modified depending on the application. The video output is digital and the temperature range is between -4° F and 140° F. Compared to the other options, this is a low-end thermal core and is also reflected in the price. The Sofradir-EC Core Atom 80 is priced at $1,500.00 including the software and a development kit.

DRS Technologies Tamarisk 320- The Tamarisk 320 is made by DRS Technology. It has a resolution of 320 x 240 and a pixel pitch of 17 micron pixels. It’s very small in size and low weight making it possible for different types of applications. The Tamarisk 320 has very low power consumption at less than 0.75 watts. Using the camera’s software the automatic focus feature can be configured. Another feature that it has is a digital zoom of 1x to 4x of the video image. The video output can be either digital or analog in NTSC or PAL. The price for the Tamarisk 320 is between $3,000.00 and $5,000.00 depending if a developer kit or accessories are purchased.

DRS Technologies hosts a yearly competition where students from universities in the United States can participate. By participating, DRS Technologies sponsors your project by providing the team with a Tamarisk 320 at no cost. Even though some of the features of the Tamarisk 320 fall short compared to the other options, camera being provided at no cost has a huge impact in the project’s budget. As a result, the Tamarisk 320 was chosen as our option.

3.9.2 Webcam

Our project will utilize a regular webcam whose main function will be to detect any movement. In addition to movement detection, the webcam will also provide a video feed as well as snapshots which will be streamed to the user's mobile phone or tablet. For the movement detection as well as object tracking, the OpenCV library will be utilized. The OpenCV library has many algorithms which will make the implementation easier but a compatible camera must be chosen. In addition, the algorithms are resource extensive and since it will be running on a microcontroller, the image quality should not be very high since this will just require more resources.

In order to use fewer resources and still be able to perform the OpenCV image processing algorithms the image from the webcam will be downscaled as well as turned into black-and-white. By doing this, less processing power will be utilized allowing us to provide a better user experience. Once the image processing algorithms are done we can scale back to a color image from the Webcam. As for the resolution, further testing needs to be completed, but it might be necessary to scale down the image resolution in order to use less bandwidth. As a result, a smooth live video feed can be achievable instead of sending/receiving frames every other second. Some of the Webcams that can fit the project's description are:

- Logitech HD C270
- Logitech C300
- V7 Vantage Webcam 300
- Microsoft LifeCam HD3000 L2
- Atmega32 OV7670 Camera Module

Logitech HD C270- The Logitech HD C270 is an inexpensive Webcam that is priced at less than $30.00 brand new from Amazon or for less than $20.00 if a refurbished one is chosen from Newegg. Although Logitech's website states that Linux is not supported, users from the Ubuntu Forum have had success using an open source generic driver that allows the Webcam to work with Linux. The Webcam can take pictures of up to 1280 x 720. Although this resolution is in the HD range, since the image will be scaled-down to a lower resolution for the purpose of using less resources, starting with a high quality image will help once the quality is lower since not many details will be lost. The Logitech HD C270 has a microphone which can make it possible to implement an audio feed to the robot if it seems feasible within the project's deadline. Another feature is that the Webcam has automatic built-in lighting software which improves the image if the image's light condition is poor.

Logitech C300- The Logitech C300 is one of the cheapest options available. It's not widely available since it's an older model but a brand new unit may be purchased from Ebay for less than $16.00. Similar to the Logitech HD C270, it also has a microphone which can be an option to implement an audio feed to the robot. According to Logitech's website, Linux is not supported. Although is not compatible with Linux out of the box, a generic driver can be utilized in order to make it work. The Webcam can take pictures of up to 1280 x 1024. A drawback for this model is that it has no built-in solution to improve the image's lighting.

V7 Vantage Webcam 300- The V7 Vantage Webcam 300 is the cheapest Webcam available compared to the others. Although it doesn't possess many features that the Logitech Webcam have it meets the project requirements. The

Webcam can be purchased for less than $10 from multiple sources such as Amazon, Newegg, or Wal-Mart. The resolution of the Webcam is 640 X 480. Although the resolution is lower compared to other solutions, the image processing algorithms will utilize fewer resources if this camera is used as the input for the video. It can record videos at 30 frames per second which is more than sufficient if not more than what is required for the project. In addition the camera features a microphone.

Microsoft LifeCam HD3000 L2- The Microsoft LifeCam HD3000 L2 is one the most expensive options compared to the previous Webcams. The webcam is compatible with Ubuntu and other Linux distributions by utilizing open source generic drivers. It can be purchase for $40 from Amazon. The Webcam can take pictures of up to 1280 x 720. Similar to the Logitech HD C270, it possesses built-in software which improves the quality of the image such as lighting conditions. It also features a Microphone with noise cancellation which can be used for an audio feed implementation to the robot. The image quality is of high definition which allows us to down-scale the resolution without losing details in the image.

Atmega32 OV7670 Camera Module- The Atmega32 OV7670 Camera Module is a very simple camera module for the Atmega32 microcontroller. It can be purchased for less than $14.00 from Amazon. This camera module can take pictures of up to 640 x 480. Compared to the other options available, this is the simplest camera that still meets the project's description. It might not be necessary to scale-down the image quality for the image processing algorithms since doing so will cause the image to lose too much detail. Some of the drawbacks of this camera is that it doesn't possess a built-in solution for improving the picture quality or lighting conditions and there is no microphone.

As can be seen from the table below, all of the cameras are very affordable and can be easily compatible with a Linux distribution. The Logitech C300, with its built-in microphone and ability to reduce the picture resolution, makes it a very probable solution. Since it's an affordable webcam, there are many guides and tutorials online that will help with the implementation.

Name	**Resolution**	**Microphone**	**Price**
Logitech HD C270	1280 x 720	Yes	$20-$30
Logitech C300	1280 x 1024	Yes	$16
V7 Vantage Webcam 300	640 x 480	Yes	$10
Atmega32 OV7670 Camera Module	640 x 480	No	$13

Table 3.9.2-1 Camera Comparison

3.9.3 Video Streaming

One of the features of our project is the ability to stream a live video feed to the user. In addition, we would like to add the option to be able to store a specific snapshot of the video feed to the mobile device. An available option that allows us to accomplish both of these features is a FOSS (Free Open Source Software) called MJPEG-Streamer. According to the software's website, MJPEG-Streamer is a command line application to stream JPEG files over an IP-based network from the webcam to a device running a web browser, either mobile device or computer. The mobile application for the project will be developed for the Android platform. Android has an API which allows us to embed a WebView. An Android WebView behaves like a browser which allows us to embed the MJPEG-Streamer.

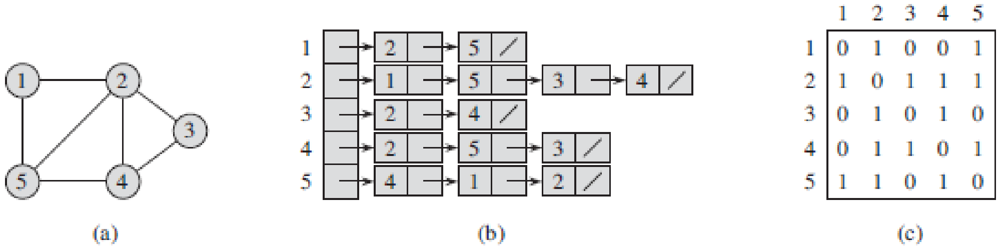
MJPEG-Streamer is a very light application and it has been written for embedded solutions. According to MJPEG-Streamer's website, streaming a video with frames at a resolution of 960 x 720 consumes less than 10% CPU on a 200 MHz router. MJPEG-Streamer is able to accomplish this by utilizing hardware compression of specific webcams to reduce the amount of CPU time spent compressing the video frame. The software works by using a series of input and output plugins. There are many types of plugins available in the Internet and each one has a different functionality. There's an input plugin called input_uvc.so which once setup with the camera, it grabs the images from the camera and stores them in a memory location. From here, the input plugin notifies the output plugin. The output plugin that will be used for our solution is called output_http.so. This output plugin allows the images from input to be made available in a video form via the Internet by using the webserver in the robot. In addition, the output plugin has a function that allows to only send one frame by executing a specific command. This will help accomplish the requirement for the mobile application.

3.10 Routing Algorithm

In a very basic concept, in order for one to travel from point A to point B efficiently a direction must be known. The challenge of autonomy navigation lies within the ability to self-generate a direction from one point to another. To be precisely correct with the concept of autonomy that is applied here, it refers to a state of being independent from any peripheral operator. The goal isn't to build a system that has a pre-determined set of directions based on permutated sets of nodes being ready for an automated look-up task. The self-routing system must be able to process a particular problem or obstacles in real time.

Graph theory seems to be the most popular, also most explored, approach to this sort of problems. Graph theory is a study of set of entities that are linked together in any manner. It uses the abstract idea of relations between entities and can be applied to real world problems with the appropriate modeling method. This document includes mainly all the graph algorithms that build up to what might be needed to achieve self-navigation shall be discussed in this section. Before any discussion of any path algorithm, let's look a quick tour of graph theory so we clearly align notations in this document consistently (due to several conventions existed out there).

Mathematical Representation of graphs- As far as this document it is not so relevant to identify all of the properties of the graph fundamentals. This is simple to set forth the notation convention and regularity in this document. Basic standard presentation of graphs: $G = (V, E)$ where V is a set of vertices, and E is a set of edges. Also two main methods to apply this notion are *adjacency list* and *adjacency matrix*. The following figure showcases the simple graph model.



	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

(a) A simple undirected graph **G** with five nodes: 1, 2, 3, 4, 5
(b) An adjacency-list representation of **G**
(c) A adjacency-matrix representation of **G**
As for this particular case, V = {1, 2, 3, 4, 5} and E = {(1,2), (1,5), (2,4),…(*u,v*)}
Figure 3.10-1 Graph Representation

Minimum Spanning Trees- Suppose we have circuit board with a set of *n* electronic components in *n* fixed locations embedded into the board. If we need to connect all of them together, one way to do it might be using the total *n*-1 wires to connect them but if less number of wires could be achieved while connecting all of them together that should be a more appealing approach solve this problem. This could be applied to self-routing problems discussed earlier. Instead of having electronic components, we think of it as destinations instead. The distance between these two destinations can also be evaluated as scalar quantity. The greater distance might have a higher number. To keep things in the abstract manner, we will continue to refer these entities as vertices or nodes and distance is referred as a weighted edge. Also minimum spanning tree is just MST.

The minimum spanning trees guarantee less weight of connections for including every single node in the graph. It could be thought of as a sub-graph within a graph and the only difference is that the sub graph has better path between all the nodes. Here's a generic MST within a graph:

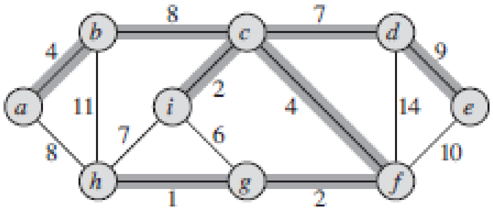


Figure 3.10-2 Generic MST graph

There are two groups of graphs that are shown in this figure. All the vertices are connected together in a certain manner but there's a sub network of vertices exits within the whole system. Notice that in the sub graph that a path from one node to another has the least weight if we follow the sub routing. This is a property of a minimum spanning tree. The generic process of growing a minimum spanning tree follows a greedy strategy. The algorithm will grow the tree one edge at a time. In an abstract level, a generic algorithm is as follow:

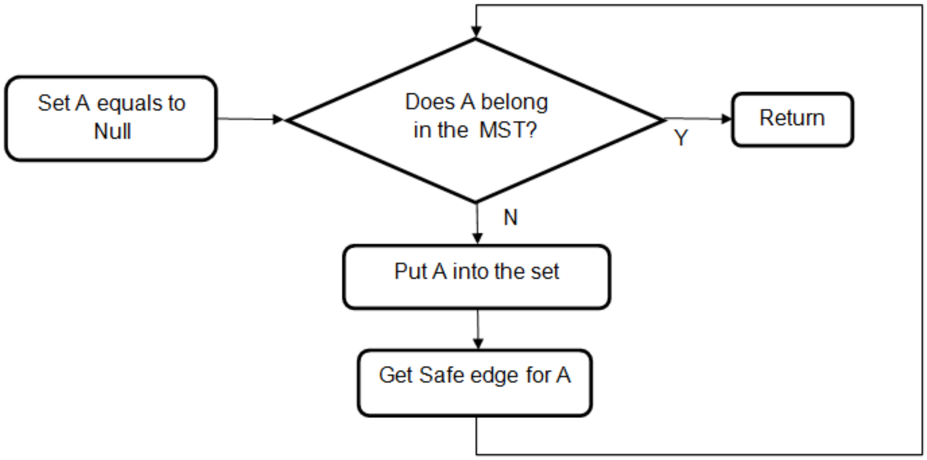


Figure 3.10-3 MST Algorithm Logic Diagram

Also notice that the definition of each line of this generic frame work of an algorithm isn't meant to be a precise definition. MST grows out of these simple directions. With the idea of minimum spanning tree in mind, we can see that the generic version of the algorithm is full of rooms to be filled with something that human should understand. A machine will need a more elaborated descriptive instructions than this. The fact is there are numbers of MST algorithms out there that came from this particular generic concept of growing it. From all the research and testing, there's one that seems to be very standard and easy for other to adjusted to their own model of application. Since many of commercial products in our findings seem to have strong links to Kruskal's algorithm, we dedicate the effort in our research to involve this algorithm than other MST.

Kruskal's algorithm is based off of greedy algorithm. It executes and evaluates all the elements in an orderly manner. The process utilizes disjoint-set data structure to control the remaining disjoint sets in the system. The edges are added to the tree one by one based on the weight, in this case lowest to greatest. It is like seeing a bucket containing the edges being poured into a small one but has enough room for just the smallest pieces. A logic diagram of the algorithm is shown below:

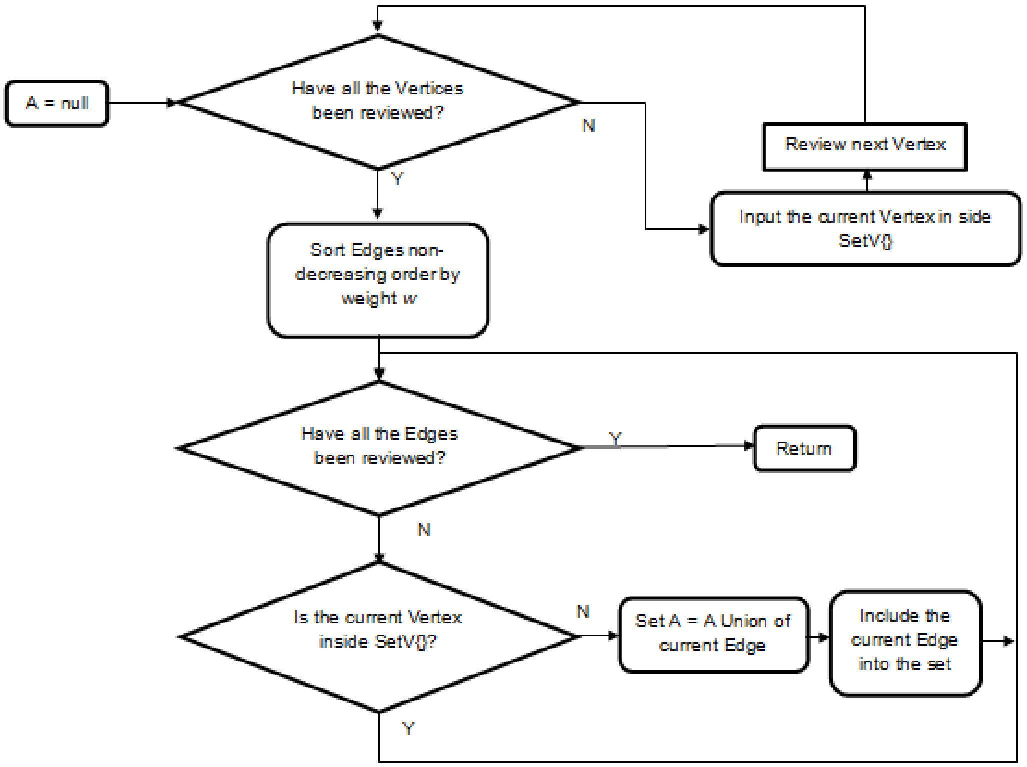

A = null
Have all the Vertices been reviewed?
N
Y
Review next Vertex
Input the current Vertex in side SetV{}
Sort Edges non-decreasing order by weight *w*
Have all the Edges been reviewed?
Y
Return
N
Is the current Vertex inside SetV{}?
N
Set A = A Union of current Edge
Include the current Edge into the set
Y

Figure 3.10-4 MST-Kruskal Algorithm Logic Diagram

This particular step by step diagram below takes place in the second loop after all the nodes have been rearranged. But before this process happens, two main things were already setup for the iteration to process. One is the make set, where all the vertices are added into a list (or a set). Two is sorting; all the edges are sorted in an ascending order. Many high-level Object-oriented programming (OOP) languages prefer to use priority queue for this process. The second for loop then would just iterate through the set of edges (one edge is consisted of two nodes *u* and *v*) from lowest to highest and for each edge it would then check whether the set that *u* belongs in are the same as the set that *v* belongs in. It would make sure that these two sets will merge by the end of the turn. The for loop iterates each edge once from low to high, it then automatically joins in the lowest weight into a new set and as a result, we end up with a sub network which is a minimum spanning tree.

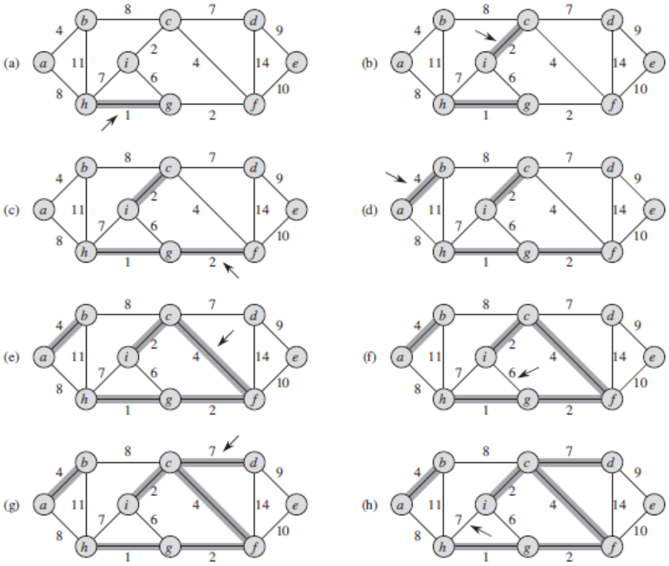


Figure 3.10-5 Steps by Steps of Kruskal's Algorithm Implementation

This particular step by step diagram below takes place in the second loop after all the nodes have been rearranged. But before this process happens, two main things were already setup for the iteration to process. One is the make set, where all the vertices are added into a list (or a set). Two is sorting; all the edges are sorted in an ascending order. Many high-level Object-oriented programming (OOP) languages prefer to use priority queue for this process. The second for loop then would just iterate through the set of edges (one edge is consisted of two nodes *u* and *v*) from lowest to highest and for each edge it would then check whether the set that *u* belongs in are the same as the set that *v* belongs in. It would make sure that these two sets will merge by the end of the turn. The for loop iterates each edge once from low to high, it then automatically joins in the lowest weight into a new set and as a result, we end up with a sub network which is a minimum spanning tree.

Single-Source Shortest Paths- Another concept that should be considered in this particular problem is determining the best path from a particular spot to any location in the map. This sounds simple enough and seems easier than generating the minimum spanning tree but it requires a bit different approach. For the design that has a simplified weight for each edge, Dijkstra's Algorithm could easily handle the path finding process in a very efficient manner. Here's a quick look at a standard version of the algorithm:

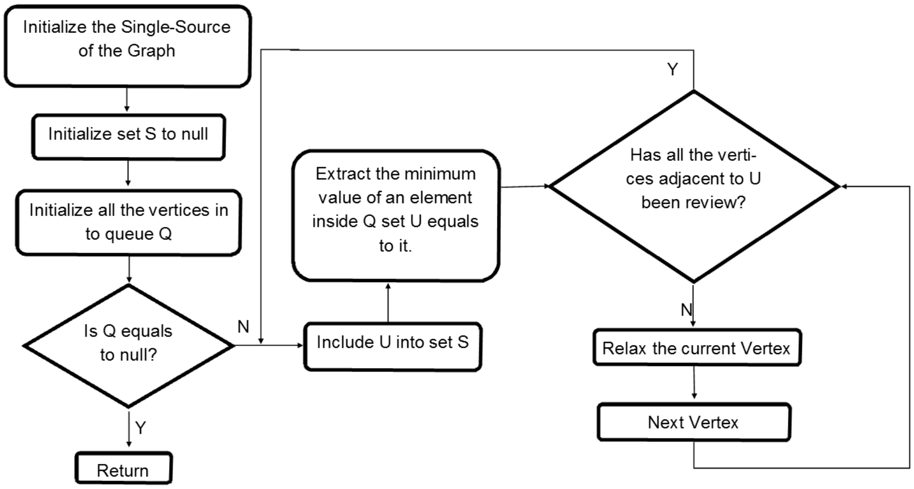

Initialize the Single-Source of the Graph
Initialize set S to null
Initialize all the vertices in to queue Q
Is Q equals to null?
Y
Return
N
Include U into set S
Extract the minimum value of an element inside Q set U equals to it.
Has all the vertices adjacent to U been review?
Y
N
Relax the current Vertex
Next Vertex

Figure 3.10-6 Dijkstra's Algorithm Logic Diagram

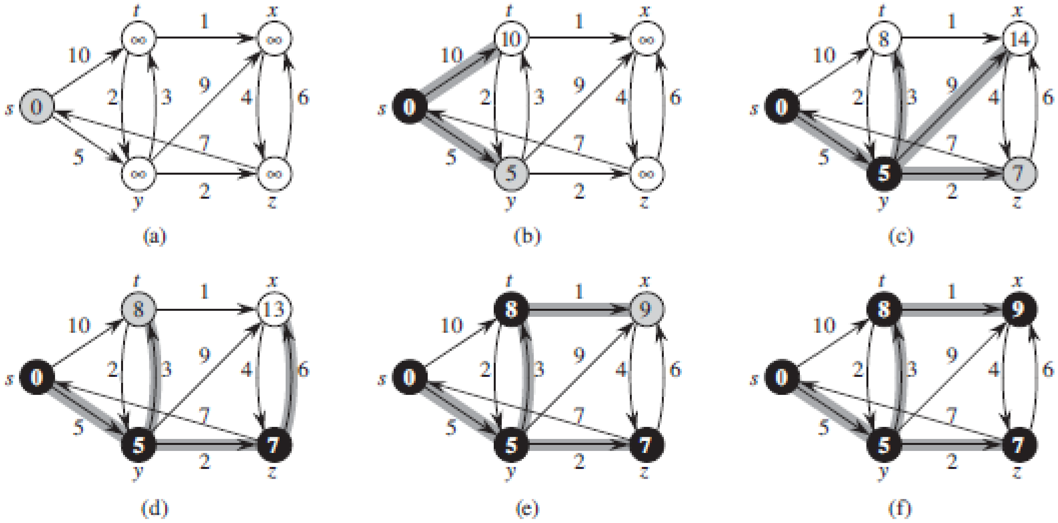


Figure 3.10-7 Step by Step of Dijkstra's Algorithm Implementation

The steps shown in the figure occurs inside of the while loop. The initial condition would be all the vertices are valued at ∞ and the origin vertex is zero. The iteration just keep evaluating each vertex and its edges then put it in set S and so forth. Then, we keep extracting the vertex with the minimum value from the Q and relax every edge that has the vertex in it.

The graph model can be very useful if we apply this in a control environment where there are static landmark in a given area. The runtime of both algorithms is O (n log n), so it can easily be applied in a dynamic landmark. To think more abstract, we can model constantly changing targets as dynamic nodes, which will only require an update of a graph for every cycle. Both algorithms can easily be modified in our advantage in traversing around multiple objects.

3.11 Video Image Processing

An important part of the project is the Image Processing subsystem since it will allow us to make the robot autonomous. In order to facilitate this subsystem, we decided to break it down into three sub-systems: Movement Detection, Acquire Target, and Target Tracking. The option of creating our own algorithms was not considered due to the complexity and time allotted. While researching for an existing option or library to facilitate the implementation we came across the Robot Operating System (ROS) and the Open Source Computer Vision Library (OpenCV). ROS was a new tool for the team since we had not heard of it before. Similar to OpenCV, ROS has a collection of tools and libraries aim to simplify the implementation of a robotic platform. Upon further research and investigation of their documentation, we decided to use OpenCV due to its popularity, documentation, and vast amount of tutorials. OpenCV provides the necessary algorithms in order to achieve the project's requirements. In addition, it's compatible with many different platforms allowing us to have a greater option of hardware.

Movement Detection– Movement Detection is one of the most important of the three sub-systems since it's the one that activates the remaining two sub-systems. While the robot is in idle, it will be taking as input video frames from either the webcam or thermal camera. OpenCV has a "Differential Images" function which allows us to determine when any movement occurs. The way the function works is that it compare the latest frame with the previous frame. These two frames are then subtracted and if any difference in the pixels is found, some type of movement occurred.

Another method that can be utilized is by using the Collins et al. algorithm. This algorithm is similar to the "Differential Images" except that it utilizes three frames (previous, current, and next) instead of two. The previous and next frames are subtracted first, afterwards, the current and next frames are subtracted. The result of these two computations are then compared with an AND gate, and a threshold is found. In order to prevent false positives, it is recommended to only consider a threshold higher than 5. If the threshold found is higher than 5, this means that a movement occurred.

According to different implementations found online and a GitHub repository the latter method seems to be the preferred method. Not only is it more accurate, but it also utilizes less resources, although it requires one extra frame. Both of the methods will be tested for accuracy in our implementation and the best one will be picked depending on the results.

Acquire Target– In order to determine the object or target that triggered the movement detection algorithm, a window base search algorithm will be utilized. The result from the movement detection sub-system will be used as the target. From here, a window based search will be performed on the current frame in order to determine the current location of the target. Another option that we have is to train the algorithm with a library of images. This is done by having a group of images of people in different positions such as standing or walking. A disadvantage of this is that the algorithm depends on finding the target in a position similar to one found in the library. In order have a better chance of finding the target, a large library will be required. Even though this will increase the accuracy of the algorithm, more resources will be required since the algorithm will have to traverse through a larger list of possibilities. As a result, we will focus on implementing the window based search algorithm without the library.

Target Tracking– In order to track the acquired target, the output from the acquire target sub-system will be utilized as input. OpenCV offers the CAMshift (Continuously Adaptive Meanshift) algorithm which selects the top left corner of the target, in this case, the output from the acquire target sub-system and keeps on tracking that point while in motion. The tracking is done by continuously performing a window based search on the video frames.

Since one of our project requirements is for the vehicle to be autonomous, we need to keep the target being tracked in the center of the camera view at all times. In order to accomplish this, the output from the target tracking sub-system will be utilized to determine in what direction to move the vehicle.

3.12 Web Server

One of the features of our project is the ability for the user to interact with the robot via the Internet using the mobile application. Some of the options that the user will have available are:

- View a live video feed of what the robot is currently seeing through the Webcam.
- Save a screenshot to the user's phone/tablet.

In order to accomplish these tasks a web server is required to handle the commands from the user to the robot. There are a few options as for the type of server such as a local web server or a cloud-based solution. In the cloud-based solution the options that we have available are the Amazon Elastic Compute Cloud (Amazon EC2) and the Google App Engine. Both of these two options would allow the software to make use of faster hardware making it possible to run resource extensive tasks. The Amazon EC2 offers a free trial that lasts 1 year but afterwards the user has to pay in order to continue using their service. Since we would like a free permanent solution, the fact that Amazon EC2 stops being a free service stops being one of our options. The Google App Engine offers a free plan and although it has some usage limitations, the limitations do not affect our project. Since the Google App Engine seemed like a possible solution for our project, we decided to come up with a scenario that the robot might encounter. A scenario that our robot will encounter is shown below:

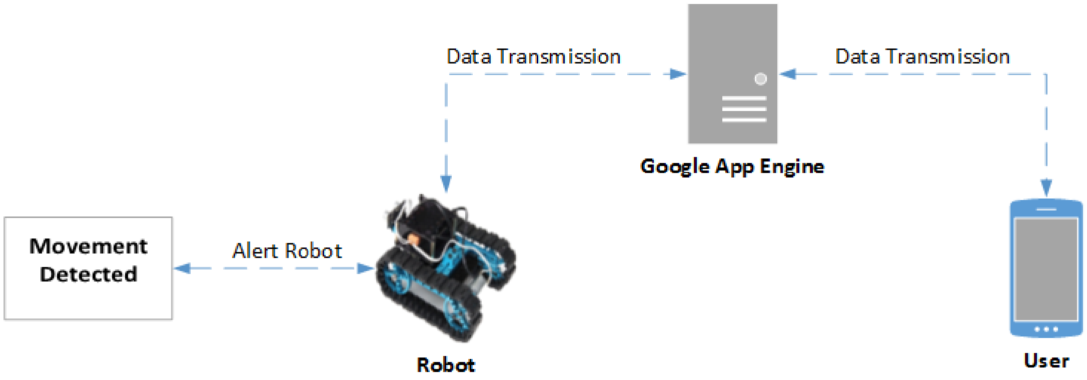


Figure 3.12-1 Google App Engine Scenario

In the Scenario shown above, the robot detects a movement and it alerts the user. In order to alert the user, the robot has to first communicate with Google App Engine. Once the alert has been sent to Google Alert Engine, the alert is then redirected to the user. If the user decides to take action and send a command, the route will be the same but in reverse order. This scenario allowed us to see a conflict that might affect our project. Since we are trying to create a real-time solution, having the commands and data transmission go through multiple channels before reaching the destination can create delays. As a result, we decided to not rely on a cloud-based solution and go for a local-based solution instead. By using a local-based solution, we will have more control with the data transmission and bandwidth being utilized. In addition, a communication channel is removed decreasing the number of delays.

A disadvantage of running a local-based web server is that the IP (Internet Protocol) address could change at any given time. The majority of ISP (Internet Service Provider) by default provide their customers with Dynamic IP addressing. Having a Dynamic IP address can cause a problem because the IP can change at any time. If the IP address changes, the user would have to find out what is the new IP address and then has to change the server IP in the mobile application. There are two different ways where this can be solved. The first way is to request Static IP addressing from the ISP. Usually, Static IP addressing is only offered to Business locations. If the user is able to acquire Static IP addressing there is usually an extra fee that has to be paid. The other solution is to setup and host the DNS (Domain Name System) via a free DNS provider. The way a DNS provider works is that once an account is created, a sub-domain is assigned to the account along with a pair of name servers. The web server has to be setup with the provided name servers, this way whenever the IP changes, it will still be accessible via the sub-domain. By using this solution, the user will not run into problems with the communication setup of the mobile application and web server. There are various free DNS providers such as No-IP, Duck DNS, and Free DNS. All of the mentioned providers are a possible solution and are completely free.

Since we will be using a port of Linux for our project, we have a couple of open source options. Some of the options that were considered are shown below:

- Apache
- OpenLiteSpeed
- Nginx
- Node.js

Apache- Apache is known as the most popular web server. According to W^{3}Techs website, as of February 2014, 60.7% of active web sites are currently using Apache as a web server. Apache is a well-known web server and very matured due to its age. It's packed with many features, many of which are not needed for most projects. As a result of having so many features, it makes it a resource hog on hardware of all scales by using too much Random-Access Memory (RAM) and Central Processing Unit (CPU). In order to mitigate this, we will try to only turn on the modules that are needed for our project. An advantage to using a popular web server such as Apache is that there are many different types of implementations available in the Internet.

OpenLiteSpeed- Another option that we are considering is OpenLiteSpeed. OpenLiteSpeed is the free version of the commercial web server LiteSpeed. According to W^{3}Techs, as of February 2014, 2.0% of active websites are currently using LiteSpeed as a web server. The LiteSpeed web server is well known for its performance under very high loads and compatibility among applications. One of the reasons why LiteSpeed doesn't have a larger market share is because their open source web server, OpenLiteSpeed, was only released last year. Since it's now an open source option, it will be considered for our project as a possible solution.

Nginx- Nginx is known as the high-performance HTTP server. According to W^{3}Techs, as of February 2014, 20.2% of active websites are currently using Nginx as a web server. Some of the popular companies running Nginx are Netflix, GitHub, and WordPress. It's a web server that has been proven by today's standard of heavy loads and praised for its simplicity and ease of setup. One of the advantages of Nginx is the way it handles requests. Instead of using threads as Apache and OpenLiteSpeed, Nginx is event-driven and asynchronous. Nginx is compatible with multiple Linux distributions and many different implementations can be found online.

Node.js- Node.js is a fairly new framework written on java-script that acts as a web server. Similar to Nginx, it's also event driven and asynchronous. An area where Node.js differs is that it's not a complete solution out of the box. Node.js allows the developer to create a fully customized web server for a specific task. According to W^{3}Techs, as of February 2014, 0.1% of active websites are currently using Node.js as a web server. Since our robot will always be completing the same task, Node.js becomes a very favorable solution to our project. Further testing will have to be done as well as creating a working implementation. Since Node.js is fairly new, there are not many implementations available. Lately, Node.js has been becoming more popular and new applications are being created based off of Node.js. By the time our project is in the testing phase, we hope that Node.js has matured a little more and the implementation becomes easier

Server Benchmarks- In order to decide which web server to utilize, each one of the studied web server solutions will be tested with a prototype mobile application by performing a benchmark. The benchmark consists on running a simple TCP/IP web server which will have a Server side and a Client side. The TCP/IP web server will be implemented using the JAVA language. The Server side will reside on a virtual private server and it will always be listening for any incoming connections from the client. The Client side will reside on the mobile device which will be running a prototype application that sends commands to move the robot to the Server. The virtual private server specifications are the following:

- CPU: Intel Xeon L5520 at 2.27 GHz, limited to 1 core and down-clocked to 1 GHz.
- Random Access Memory (RAM): 512 Megabytes
- Operating System: Centos OS 6.5 64-bit
- Java Development Kit (JDK) Installed

Each web server will be installed and implemented on a fresh install. In addition, the web server settings will be as is from the developer, no optimization will be performed other than the required dependencies. It will be assumed that only one user will be interacting with the web server at any given time. The following is the information that will be captured during the benchmark:

- Memory usage before installing/implementing the webserver
- Memory usage with webserver installed and running
- Memory usage with TCP/IP server being idle
- CPU usage with TCP/IP server being idle
- Memory usage with TCP/IP server running and receiving data
- CPU usage with TCP/IP server running and receiving data

Apache Benchmark Results- As it can be seen in the figure below, once Apache is installed the memory usage increases by 38 megabytes. The TCP/IP Server doesn’t consume any memory while is listening for incoming connections which is excellent since it will be listening for incoming connections for most of the time. Once the TCP/IP Server is receiving and parsing data, the memory usage increases by 4.4 megabytes. With both Apache and TCP/IP server running, the total memory usage is 42.4 megabytes.

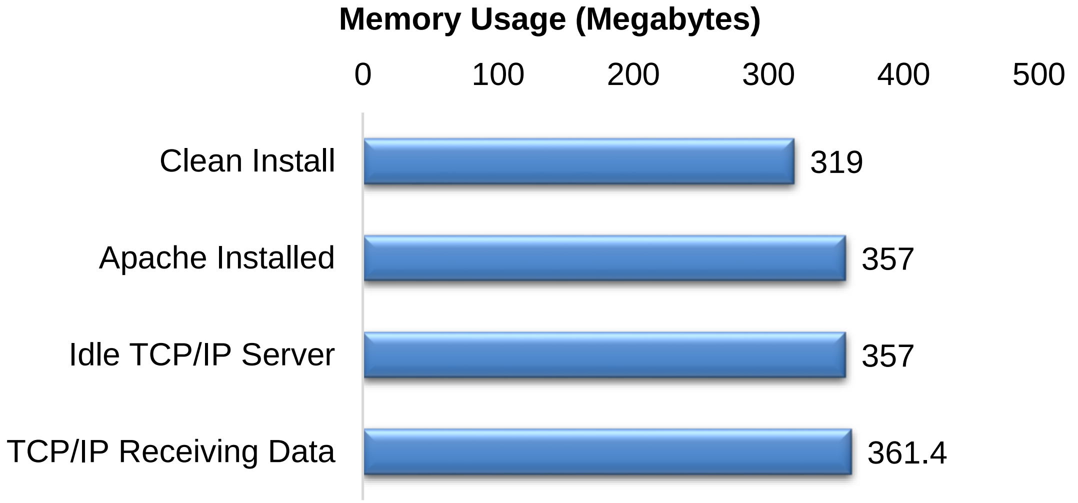


Figure 3.12-2 Apache Memory Usage

CPU is one of the most important factors since image processing is very CPU intensive. By utilizing a very simple TCP/IP Server the CPU usage only increases by 1%, as shown below, when it is receiving data from the mobile application. This allows us to reserve the rest of the CPU for the image processing algorithms.

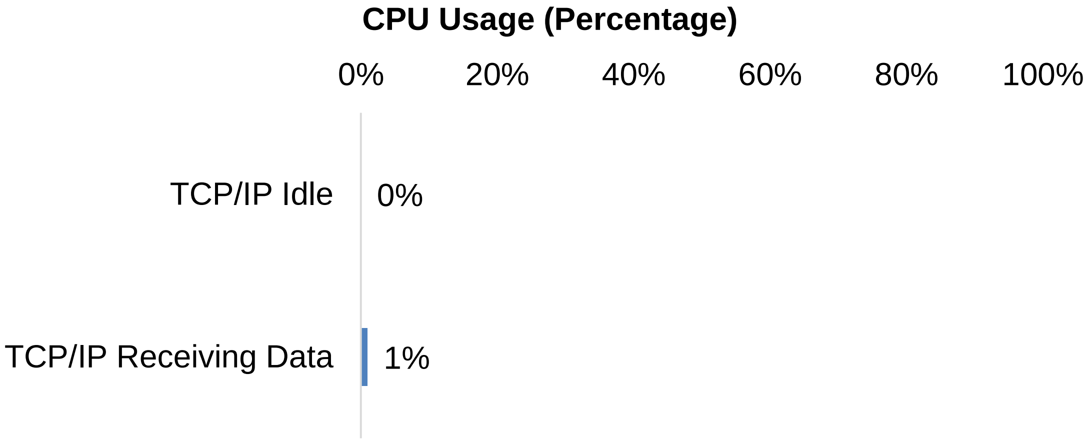


Figure 3.12-3 Apache CPU Usage

Nginx Benchmark Results- As it can be seen in the figure below, once Nginx is installed the memory usage increases by 22 megabytes. The TCP/IP Server doesn’t consume any memory while is listening for incoming connections which is excellent since it will be listening for incoming connections for most of the time. Once the TCP/IP Server is receiving and parsing data, the memory usage increases by 3.3 megabytes. With both Nginx and TCP/IP server running, the total memory usage is 25.3 megabytes.

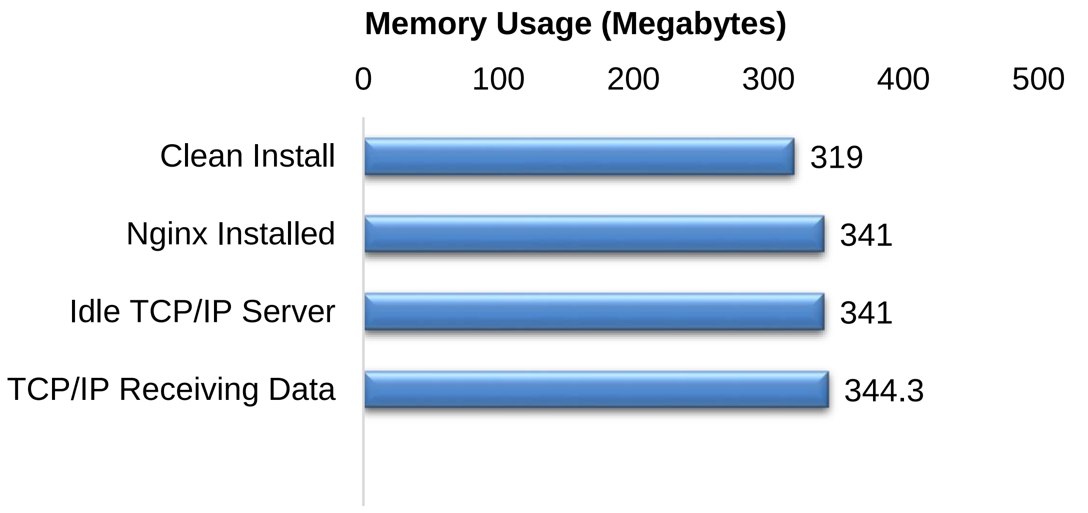


Figure 3.12-4 Nginx Memory Usage

As for the Nginx CPU usage, the result is the same as with Apache with only 1% of the CPU being utilized as processing power.

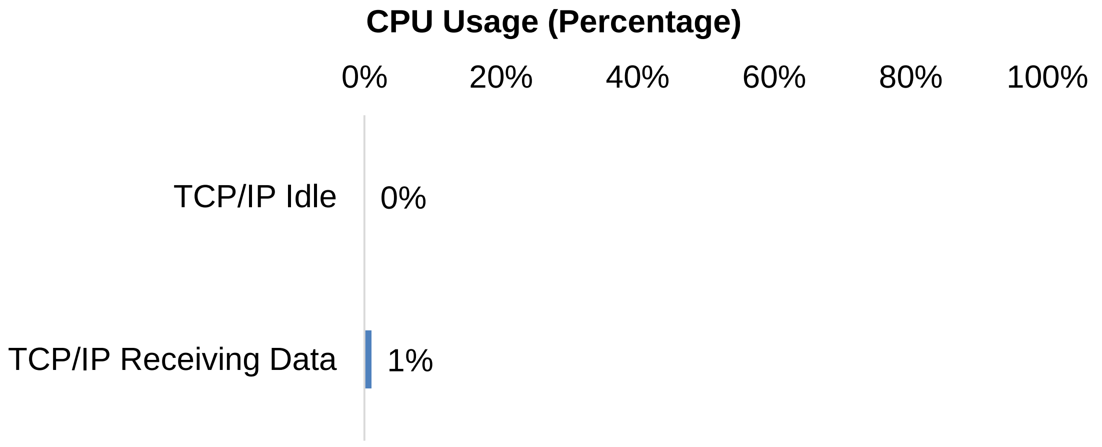


Figure 3.12-5 Nginx CPU Usage

OpenLiteSpeed Benchmark Results- As it can be seen in the figure below, once OpenLiteSpeed is installed the memory usage increases by 43 megabytes. The TCP/IP Server doesn’t consume any memory while is listening for incoming connections which is excellent since it will be listening for incoming connections for most of the time. Once the TCP/IP Server is receiving and parsing data, the memory usage increases by 4.1 megabytes. With both OpenLiteSpeed and TCP/IP server running, the total memory usage is 47.1 megabytes.

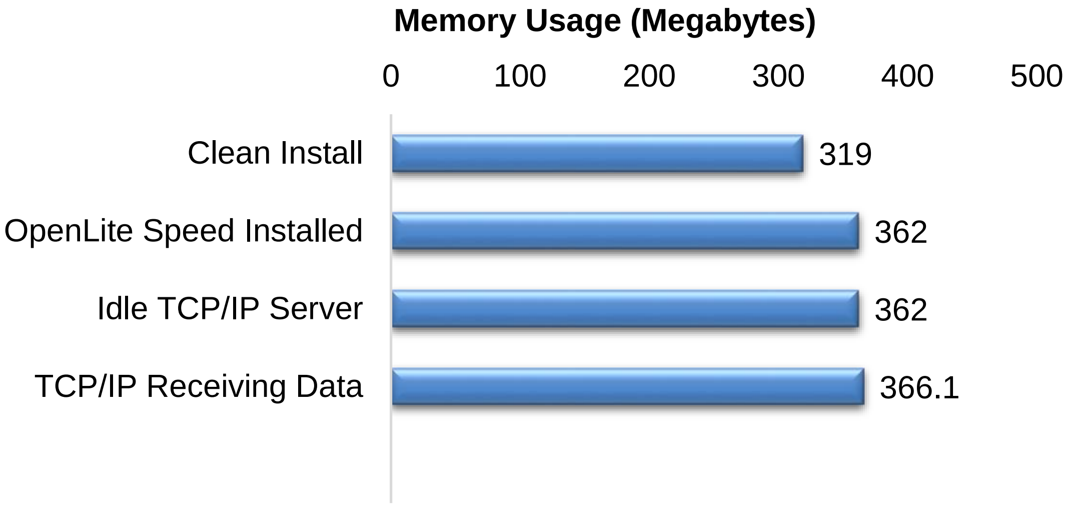


Figure 3.12-6 OpenLiteSpeed Memory Usage

As for the OpenLiteSpeed CPU usage, the CPU usage is a little lower compared to Apache, Nginx, and Node.js coming at 0.7%.

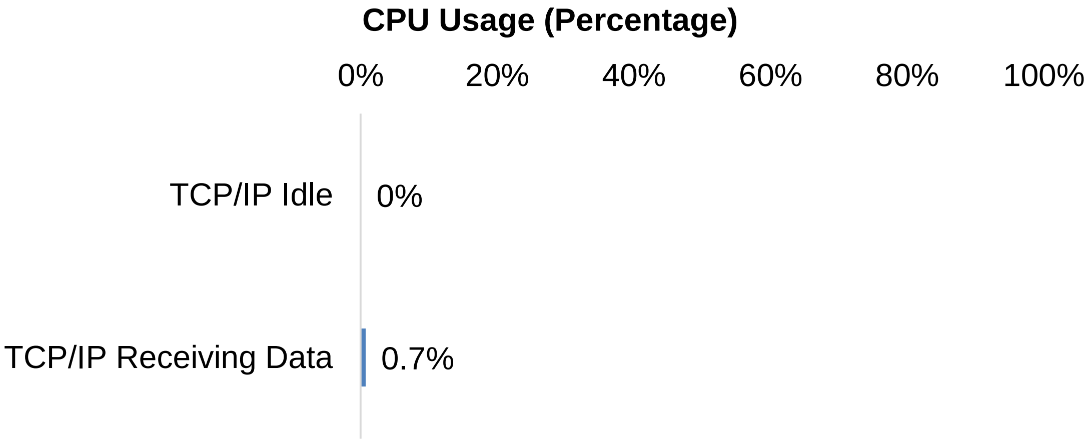


Figure 3.12-7 OpenLiteSpeed CPU Usage

Node.js Benchmark Results- As it can be seen in the figure below, once Node.js is installed the memory usage increases by 8 megabytes. The TCP/IP Server doesn’t consume any memory while is listening for incoming connections which is excellent since it will be listening for incoming connections for most of the time. Once the TCP/IP Server is receiving and parsing data, the memory usage increases by 4.1 megabytes. With both OpenLite Speed and TCP/IP server running, the total memory usage is 12.1 megabytes.

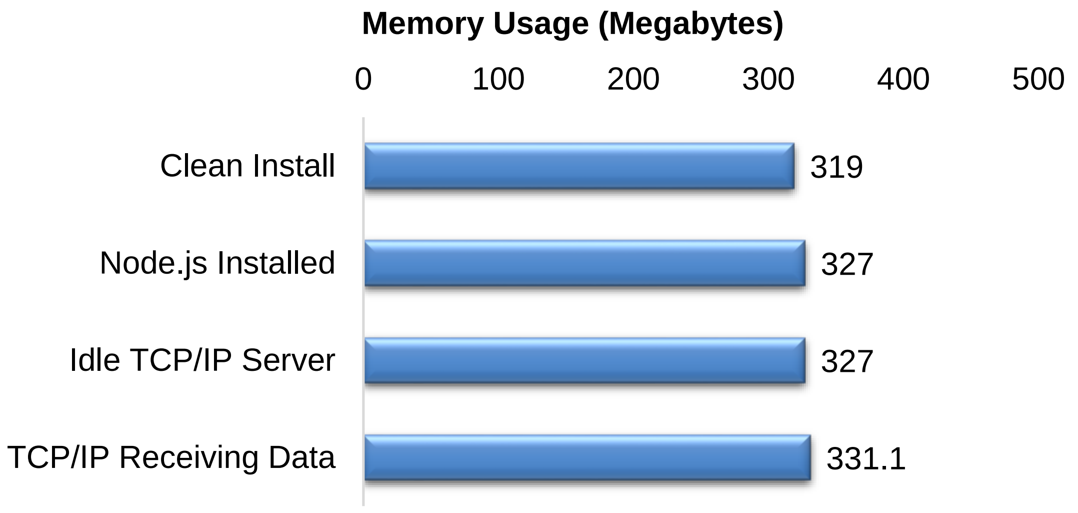


Figure 3.12-8 Node.js Memory Usage

As for the Node.js CPU usage, once again the result is the same as with Apache and Nginx with only 1% of the CPU being utilized as processing power.

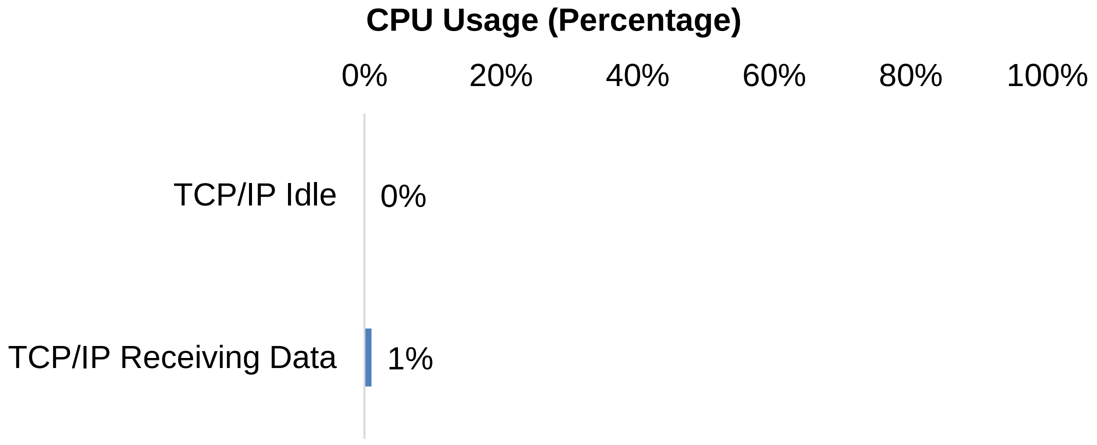


Figure 3.12-9 Node.js CPU Usage

Benchmark Conclusion- From the conducted benchmark it can be concluded that Node.js and Nginx can both be possible solutions for the project. Apache, although simpler to implement and to work with, is just too big and consumes a lot of memory. OpenLiteSpeed, although is faster than Apache and consumes less memory under high loads as well as less CPU, it is not a practical solution for just one user interacting with the server. When there’s only one user interacting, OpenLiteSpeed consumes more memory than Apache. In addition, OpenLite Speed requires too many dependencies that might not be available for a microcontroller or FPGA. As for the CPU usage, all the webservers tested used 1% or less of the CPU. This was a great find since image processing algorithms will need as much processing power as possible in order to improve accuracy and response.

3.13 Wheel Type/ Movement

This is the section where we will provide a very detailed explanation on the type of wheels we are considering. We are looking for the best wheels possible that will provide multi-directional movement and ease of implementation. In this section we will be talking about three types of wheels. Mecanum, Tank style, and the traditional RC wheels.

3.13.1 Mecanum Wheels

Introduction- Mecanum wheels are designed to move in any direction wanted. They can move forward, backward, and side-to-side unlike the traditional wheels. One movement that they can also do is rotational. This is the best thing about these wheels because the movement of these wheels makes vehicles move a lot smoother with endless possibilities.

In the figure below you can see the architecture and design of the wheels. The angle between the roller’s axis and central wheel are usually about 45 degrees but can actually be any angle. Each roller is precisely manufactured to receive maximum performance when in motion. The way these wheels move in any directions is a big advantage for our vehicle. As a result, the object avoidance and tracking algorithms as well as the motor controller can take advantage.

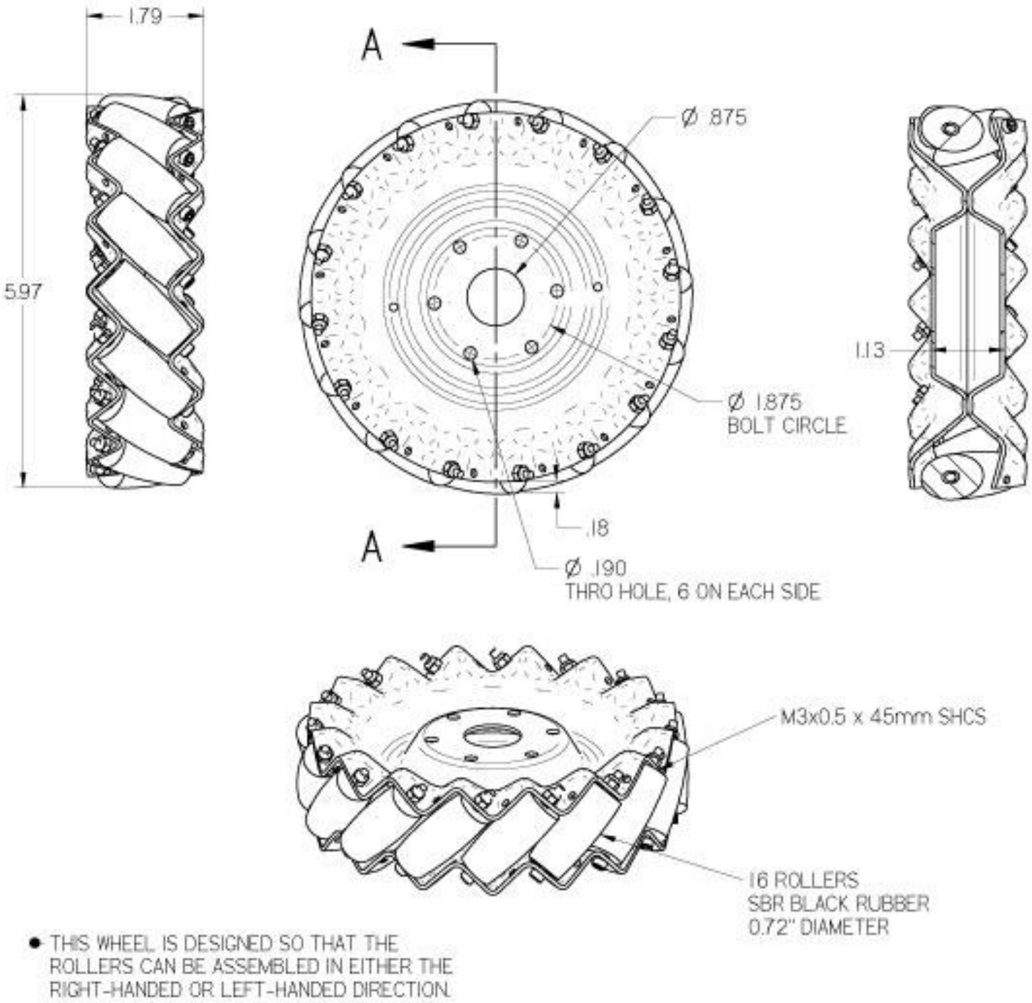

1.79
5.97
A
A
Ø .875
Ø 1.875
BOLT CIRCLE
.18
Ø .190
THRO HOLE, 6 ON EACH SIDE
1.13
M3x0.5 x 45mm SHCS
16 ROLLERS
SBR BLACK RUBBER
0.72" DIAMETER
• THIS WHEEL IS DESIGNED SO THAT THE ROLLERS CAN BE ASSEMBLED IN EITHER THE RIGHT-HANDED OR LEFT-HANDED DIRECTION.

Figure 3.13.1-1 Mecanum Wheel Dimension's (Reprinted with permission from Richar Kornyak)

Movement- As you can see in figure 3.13.1-2 below comparing Mecanum wheels to conventional wheels you can notice some similarities and differences in the diagram. For forward direction, both of the different types of wheels are moving in the same direction. Notice how all wheels are moving (movement is the same for backward motion). As you take a look at the left diagram, you can see there is a big difference in how the wheels are facing. The Mecanum wheels do not have to rotate like the conventional tires do in order to go left or right. This saves time and makes the robot move more efficiently while tracking. To go left or right the top left and the bottom right wheels move in the same direction and the top right and bottom left move in same direction. The best movement is using the diagonal. As you can see when going diagonal the platform only uses two tires instead of having to use all four of the tires.

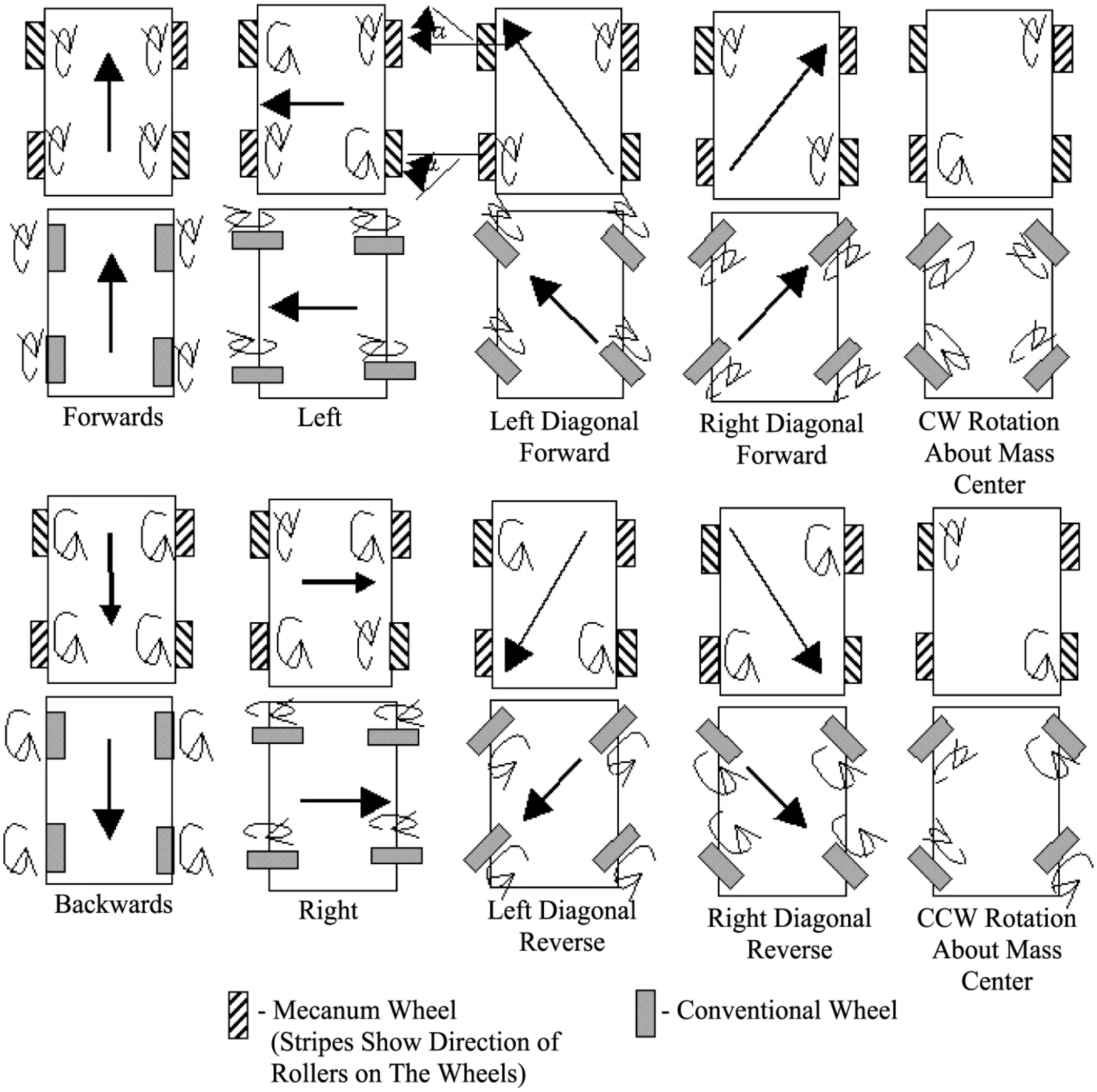


Figure 3.13.1-2 Mecanum Wheel's Movement (Reprinted with permissions from Syed Mohsin)

Advantages and Disadvantages- The movement section explained the motion of these wheels and how they will be beneficial to use for motion. Also other benefits of having these wheels are their compact size and high load capacity. For our project we will be focusing in on the 100mm to 203mm wheels if we decide to use this type of wheel. As you can see from the table below, the wheels can withstand very high load capacities.

Diameter	Width	Net Weight	Load Capacity	Price (set of 4)
100mm	50mm	400g	8kg	$217.99
152.4mm	55.52mm	600g	15kg	$317.99
203mm	78mm	2.4 kg	>150kg	$497.99

Table 3.13.1-1 Mecanum Wheel Comparisons by Size

A disadvantage of these wheels is slipping. The reason being is because the wheel is made of only one roller. The roller inside of the wheel means that there is only one single point of contact with ground any time the wheel is being used. Another disadvantage of supporting these wheels is the price. These wheels may be some of the best wheels on the market but price reflects why. They tend to be very expensive but are highly efficient. As a result of the motions that can be achieved, Mecanum wheels are a very viable option for our project.

3.13.2 Continuous Tracks (Tank Style)

Introduction- Continuous tracks are the wheels made on the traditional tanks. The tracks are very large and can make the robot stable by evenly distributing the weight of the robot. The use of tracks can be a very useful tool if you plan on moving over many surface areas. They can be used not only on smooth surfaces, but on rural surfaces such as gravel, sand and be very beneficial on inclines. With the continuous tracks you can have two different types; metal or rubber. The metal tracks are very durable and are hard to damage. This could be an option for the robot to gain maximum efficient in numerous different surfaces. The robot could be efficient in any type of atmosphere and even in the house.

Movement- The movement of these tracks all depend on the two drive sprockets attached to the inside of the continuous tracks. From the figure below, you can see the two sprockets are evenly distributed from the center. This is what allows even distribution of weight of the robot on both sides. The tracks are not very thick but have ridges to provide traction as needed. Motors are what control the sprockets and the tracks move around the two sprockets. A very common comparison of continuous tracks is conveyor belts. They move ideally the same way and are controlled by the motors that make the tracks move.

Lay Out of Tracked Wheel (dimensions are in mm)

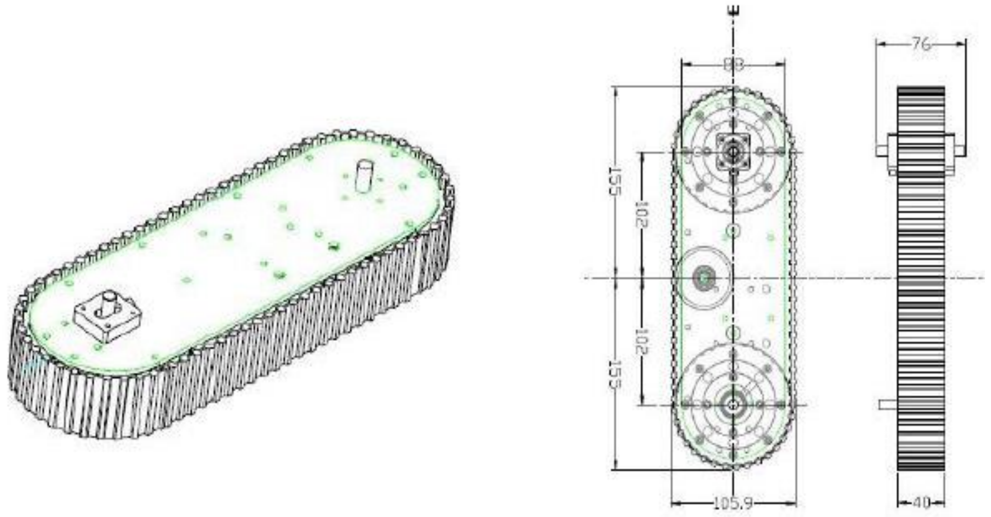


Figure 3.13.2-1 Continuous Tracks Dimensions (Reprinted with permissions from vex robotics)

Advantages and Disadvantages- Some advantages of using continuous tracks are that they are lightweight. This only applies if you are using the rubber tracks. By using the rubber tracks you also will provide great traction and also not damage the surface you are on. This can be very useful for the robot if we plan on using it inside on a wooden floor, since we don't want to damage the floors. Also, another advantage of using track is the mobility. They have good mobility in difficult places to drive i.e. if you are trying to go up and down stairs. Not too many wheels can successfully maneuver up and down stairs and support the vehicle without crashing. Another advantage of continuous tracks is that they are very cheap. On average for a set of continuous tracks they are about $15-$20 dollars.

A disadvantage of the continuous tracks is that they tend to be slower. They have a more complex system in movement than most traditional wheels. The fact that they can maneuver over rural surfaces successfully tends to slow them down. They have a lot more power than traditional wheels. They also have a very low life span. The rubber tracks tend to break by the conditions they withstand. By doing so, they are hard to fix and have to be replaced. In doing so, maintenance costs will increase making it not a viable solution for the project.

Soucy Tracks- Recently the continuous tracks have been modified to be more efficient. The Soucy Tracks were made to provide an alternative to the traditional Continuous tracks made above. The Soucy tracks have many advantages by using the continuous tracks. One of the advantages is shown on figure 3.13.2-2 below. These are two tanks driving with the two different types of tracks. As you can see the Soucy's Track system has a smoother curve meaning that the use of these wheels are more maneuverable. When trying to turn, you do not have to slow down either one of the tracks to make a turn. The vehicle will have full power at any given moment of operation.

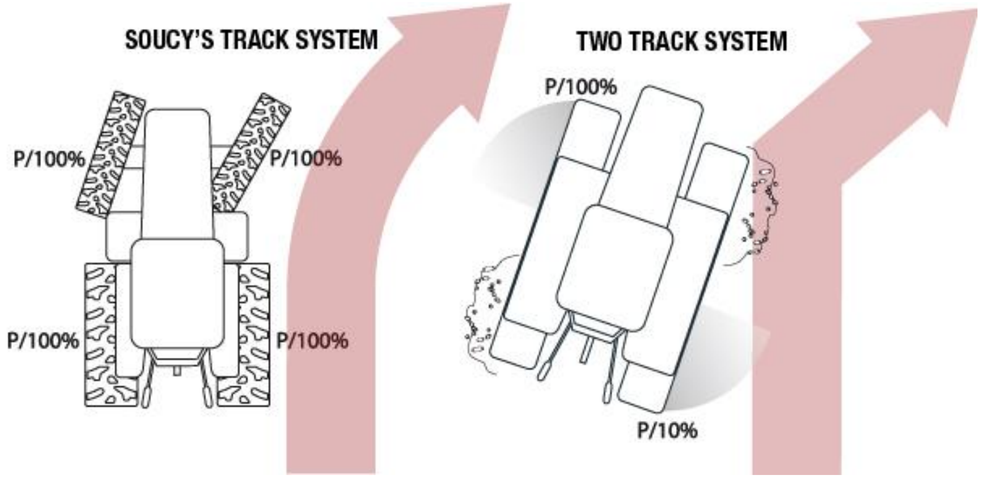


Figure 3.13.2-2 Soucy's Track System Movement (Reprinted with permissions from Yannick)

Another advantage of using Soucy's tracks is that because of the new structure of the tanks they provide a smoother ride vs. the continuous tracks. From the figure below you can see the Soucy's modified shape of the continuous track vs. the two-track system. The Soucy's track system provides a smoother ride over surface areas. By doing so, this means that the tracks would last longer and the rubber on the tracks would not be worn down as fast.

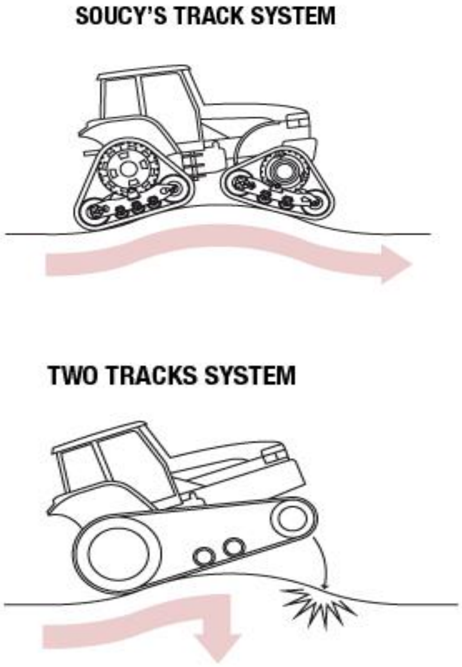


Figure 3.13.2-3 Soucy's Track System over Incline (Reprinted with permissions from Yannick)

3.13.4 Traditional RC Wheels

Introduction- The usage of RC wheels is the most common and safest choice when selecting the wheels. The reason is because they are very sturdy and dependable. In retail, a lot of the remote control cars use this design of wheels. As you can see from the figure below, these are a pair of Pololu wheels. They are not very big in width and do not sit very low to the ground like the continuous tracks. These wheels would work for somebody who is focusing on steering and maneuverability. These wheels are also very lightweight weighing about .55 oz. This would be very nice for the robot if we wanted to improve the speed on our robot. The robot would ideally be able to move faster and be more agile.

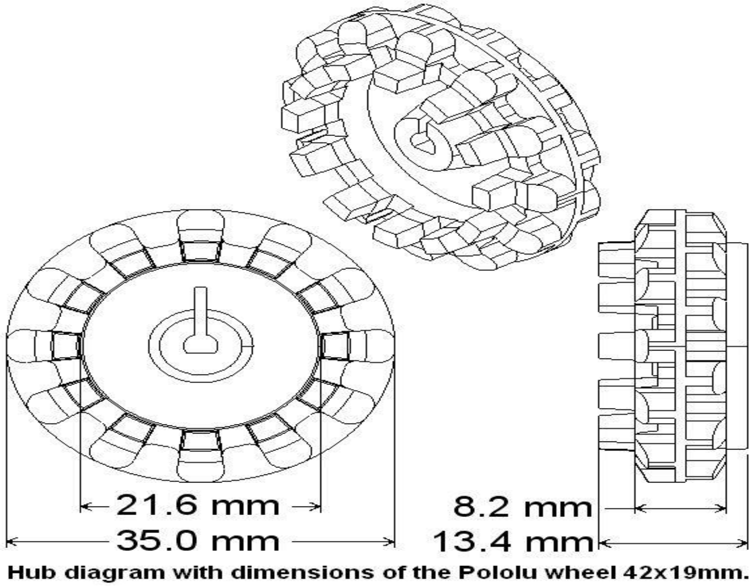


Figure 3.13.4-1 Pololu wheel Dimensions (Reprinted with permission from Vex robotics)

Advantages and disadvantages- One advantage of the traditional RC wheels is suspension. Depending on the type of wheels you are using your robot can be lifted off the ground more and have springs acting as suspension. This can be very useful in our design because if we were to run into an obstacle such as a bumpy surface, the suspension system will be able to handle it. Another advantage is the affordability of the wheels. The price for these type of wheels range between 10-20 dollars. This can come in handy just in case we are low on funding when purchasing other components. A disadvantage is not being able to move over inclines efficiently.

4.0 Hardware and Software Design

4.1 Chassis

The main factors we focused on when picking out a chassis were weight of material, size of vehicle, and cost. Because our vehicle is not going to be designed to carry any loads greater than its components, many typical chassis factors will not concern us. The combined weight of the components will be minimal compared to the batteries and the chassis itself. So minimizing the weight is not a priority. The second factor is cost. Because we are not expecting the vehicle to ever need to be able to support any noteworthy sized loads, the materials that we are able to use for the chassis is quite broad. We can use this to our advantage by finding the cheapest means of assembling a chassis. The size of the vehicle is another crucial factor, not only will a smaller vehicle increase the mobility, it also helps to keep the weight and the costs down.

VEX chassis kit- The first route we explored was to purchase a chassis building kit, filled with various aluminum parts which can be fitted together to form a frame. The VEX Metal and Hardware kit, comes with the parts to build a simple frame (shown below) and extra parts for attaching platforms and other components. The advantages of this would be that because all of the pieces are aluminum and checked with holes the weight of these parts would be kept to a minimum, while still maintaining a strong frame. The medium sized frame below weighs about 1.3 lbs. and is a square with a length of about 12.5 inches. This kit costs about $79 before shipping.

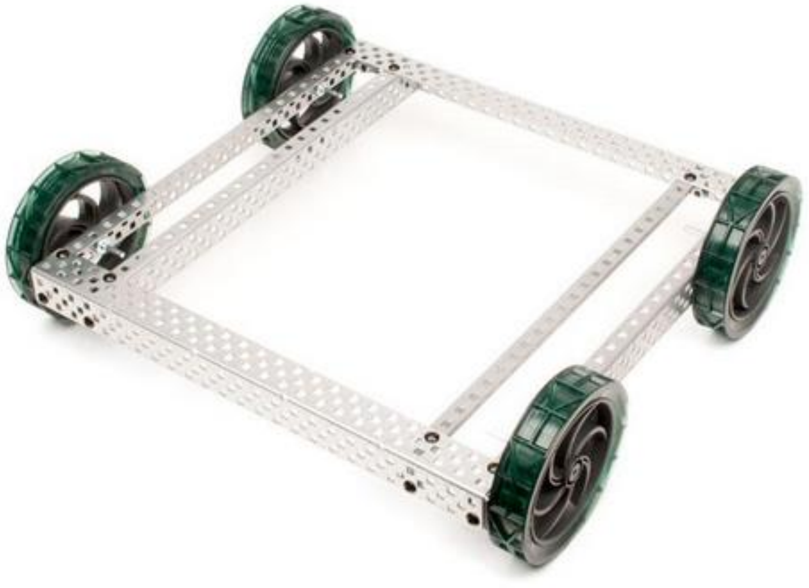


Figure 4.1-1 VEX medium sized aluminum frame

3-D Printing- The other option we came up with was to have the whole chassis 3-D printed. This way we could control every aspect of the dimensions, and could make custom fittings for our other components. Using the plastic filament material would be much lighter than using a metal, especially since the printing process allows us to select how solid the structure is printed, meaning we can choose to only fill in 20% of the solid areas thus cutting out 80% of the total volume while retaining integrity from the honeycomb pattern used when filling in a structure. A one kilogram spool of filament costs around $49 before shipping, and should provide more than enough material for at least one full print. Some basic designs were made in SolidWorks to incorporate some of the components we had already chosen.

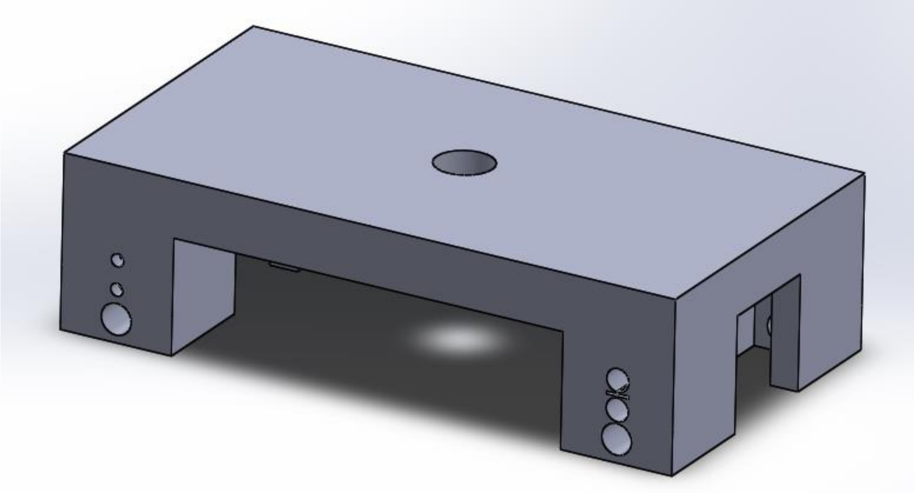


Figure 4.1-2 Basic square frame with mounting holes for motor assembly

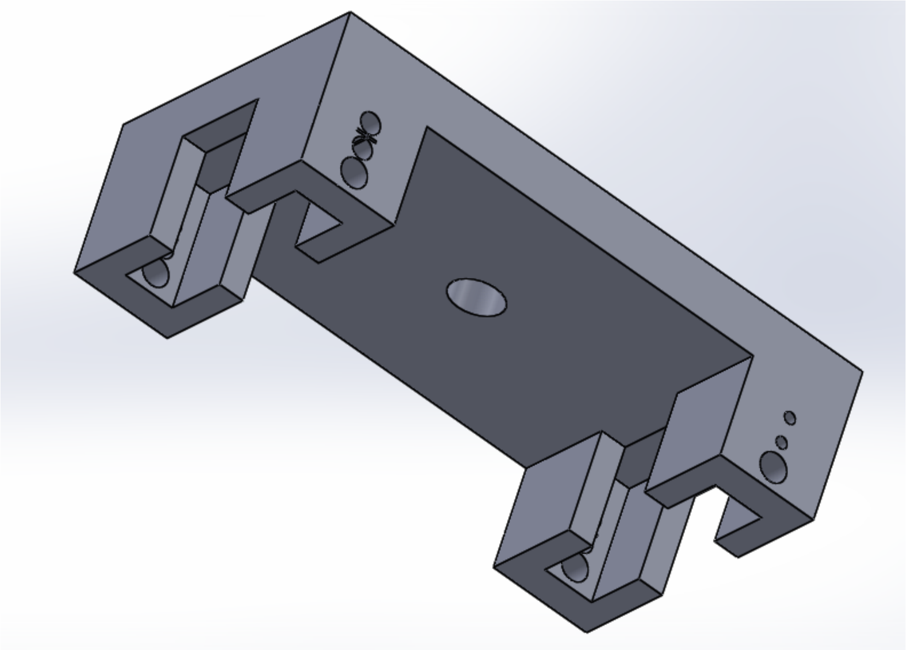


Figure 4.1-3 Underside showing the fit for the motor assembly

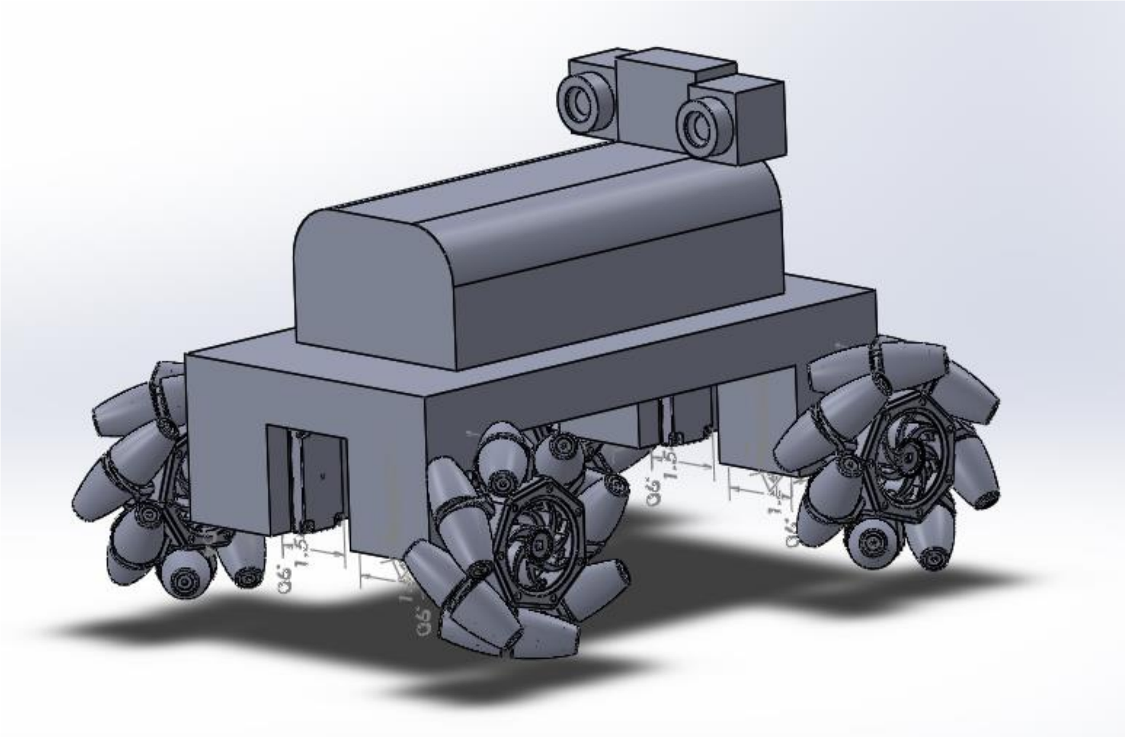


Figure 4.1-4 Mock up including wheels, motors, cameras, and a circuit board shield.

4.1.1 Integration Changes

We did buy the Vex chassis kit for the frame of the vehicle but did not get a chance to implement the 3-D printing. The reason was because at the lab we were planning to use, they have very strict size requirements on how much filament can be used for the design. Our design for the cover was out of the requirement range in required size, in order to save the time of trying to find another lab to print and having to buy the filament material needed, we decided to deviate from the 3-D printing.

4.2 Wheels

In selecting wheels we are looking for the wheels that are the lightest, cheapest, and most effective. For programming and maneuverability we have eliminated our choices down to two types of wheels, the tank style tracks and the Mecanum wheels. For movement these two tracks are better than the traditional wheels. The both tend to have their ups and downs so we have to make a decision if we are going to go with the lower cost or are we going to go with the more effective wheels. Between the two different types of wheel, they both provide beneficial aspects towards our project. For the purpose of our project we decided to use the Mecanum wheels. Some key benefits of Mecanum wheels are:

- Lightweight
- Maneuverability
- Custom look
- Can move in any direction
- Can hold a high load Capacity

As stated in the research, the movement of these wheels would be the most efficient wheels that we can use. We have to figure out exactly what size we will actually use. There are plenty of different sizes and materials that we can use. From the wheel research we were focused on the smaller design wheels. The reason being was that our first design was supposed to be a very small robot. Once we started adding more components and started altering the design of the robot, we came to the conclusion to use the bigger size wheels. The dilemma that we are facing is exactly what size.

The wheels shown below in figure 4.2-1 are the 4 pack set found at sparkfun.com. These ideally would be a very nice pick due to the fact they provide the EPDM rubber rollers which means that the roller's material ensures durability. This is a very important aspect due to the fact that we will be doing a lot of test trials using these wheels. We want to make sure that they are able to withstand the different tests that we will provide for the robot. These wheels are top of the line and the price reflects that at $74.95.

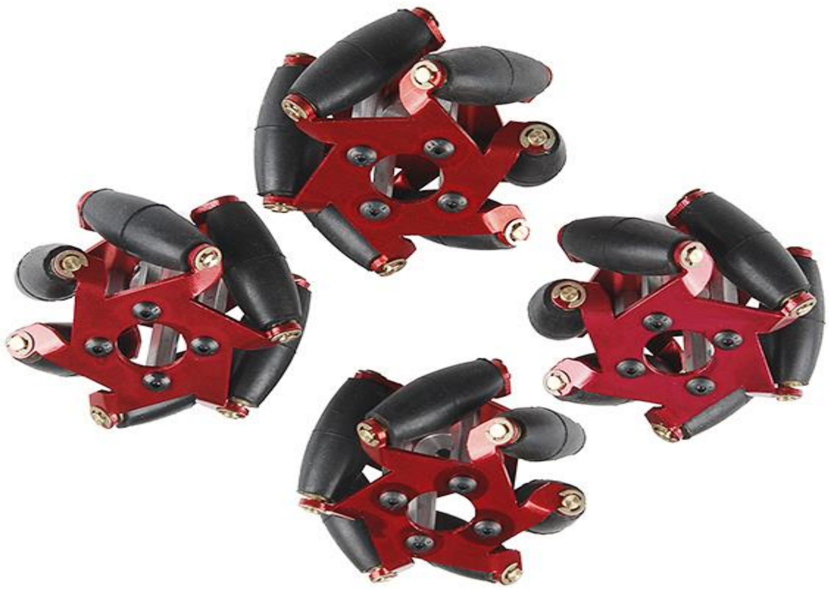


Figure 4.2-1 4 Inch Mecanum Wheels (Reprinted with permission from Sparkfun)

Below are some of the wheel's specifications:

- All metal construction
- Outside Diameter 54mm
- Wheel Width 34mm
- EPDM Rubber Rollers
- Weight: 60 g
- Price 74.95 for set

The wheels shown below in figure 4.2-2 are also in the 4 inch range. These wheels tend to have a lot of key differences from the other wheels. The wheels are heavy and they don't have the EDPM rubber rollers. The main advantage about these wheels is the price. The price is a lot cheaper for these wheels but they also don't utilize the metal constructions used in the other set of wheels.

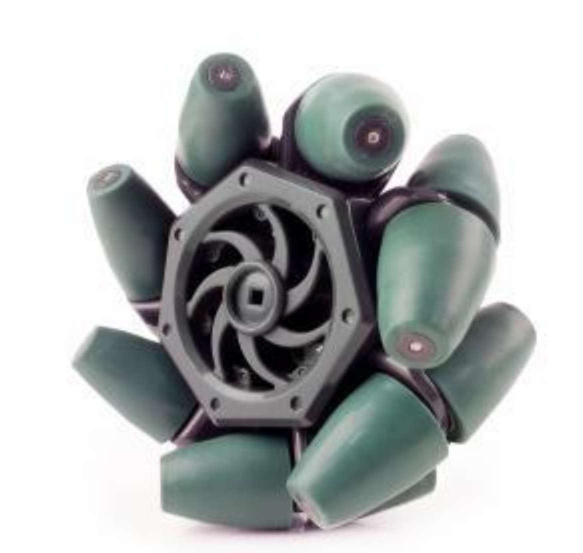


Figure 4.2-2 4 Inch Mecanum Wheels (Reprinted with permission from Vex Robotics)

Below are some of the wheel's specifications:

- Not all metal construction
- Weight 186 g
- Price 59.99
- Diameter and width not accessible

The wheels shown in figure 4.2-3 are from Amazon.com. These wheels are the best that money can buy you. They are made of steel and that means that they are the most dependable choice. They are just too big and ideally using this design of the wheels would not be very beneficial for us. This is because our robot is small as it was stated previously. These wheels are too heavy and are usually made for the bigger vehicles. Ultimately, if our vehicle would have been larger, then we would probably go with this size of the wheels. With that being said, we did not look at any more sizes greater than 6 inches. Also, the price ranges for these are out of our budget.

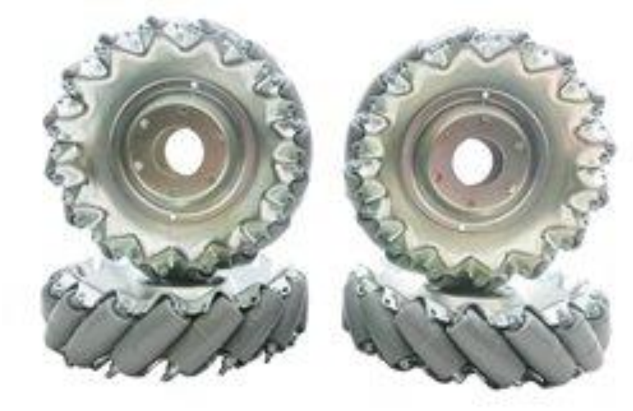


Figure 4.2-3 6 Inch Mecanum Wheels (Reprinted with permission from Sparkfun)

Below are some of the wheel's specifications:

- Outside Diameter: 151 mm
- Width of Wheel 45.66mm
- Made with steel
- Weight 512 grams
- Price 253.00

The wheels that we ended up picking for our project are the 4 inch Mecanum wheels from sparkfun.com. They are actually an upgraded version of the Fingertech's first version. They made adjustments to the rollers which reduces friction between the rollers and the hubs. We just have to make sure that we have the rollers well-greased before movement. By doing so we should be able to achieve maximum production out of the wheels. As far as balance goes, just like any other type of wheels, we need to make sure that there is not too much weight on either side of the platform. The reason being is that we need the weight to be evenly distributed on all sides.

4.2.1 High-Level Mobile Control

High-Level Mobile Control- In a general idea, simple modes of standard automobiles are forward and backward. The change in directions is implemented with turning of the wheels. Sometimes rear wheels are involved in turning to help increase maneuverability. The speed, of course, has different magnitudes that are controlled by a motor. The mobility can increase greatly when each wheel operates independently from each other. If the left wheels of a car are in the opposite direction of the right wheels, we can expect a different kind of movement from the vehicle. There are many combinations of wheels turning that can form much different kind of movements. The Mecanum wheels enable exactly that freedom of mobility.

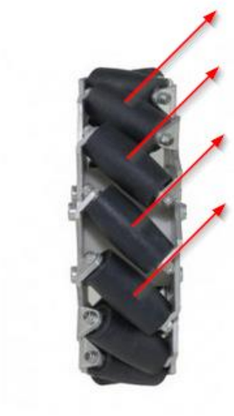


Figure 4.2.1-1 Mecanum Wheel Force Direction

A Mecanum wheel has a track of rollers laid diagonally on the outer edge imposing force in a diagonal direction from the line of direction of the wheel. When a wheel turns forward, the force actually pushes in an angle of the turning direction, so a combination of force vectors can be design to program a robot's mobility.

Wheel Control- The wheels will ultimately be controlled by the motor controller board which receives an analog signal from the microcontroller. On the high-level design, the model is created through the data structure that will be carried out by sets of algorithms to execute mobility. The numbers represented in the model will eventually be translated into a more appropriate format base on the data structure of the program.

In the most basic control format, each wheel is controlled independently; therefore each one can be operated at the same time while maintaining different magnitude. Each wheel should have a magnitude ranging from -n to n where 0 being at rest and negative value representing backward rotation. An interface for this control should look something like *moveWheel(wheel, speed)*, where "wheel" is the target wheel. As far as the hardware, after executing the instruction, the state of the wheel should stay at the input value until further instruction is given. Here is a simple logic diagram for the wheel control.

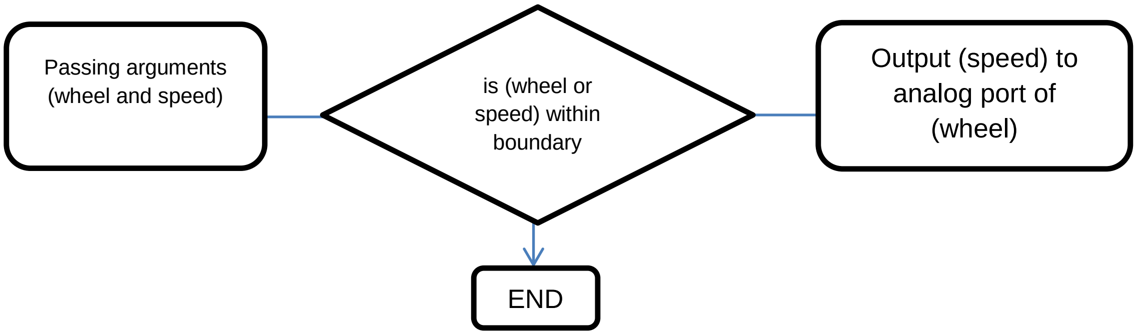

Passing arguments (wheel and speed)
is (wheel or speed) within boundary
Output (speed) to analog port of (wheel)
END

Figure 4.2.1-2 Function moveWheel Diagram

Movement Control- As discussed earlier, a movement from Mecanum wheels is formed from contribution of force from each wheel. The following will be the naming convention for this section of the design specification.

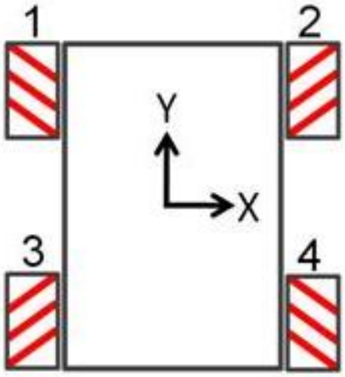
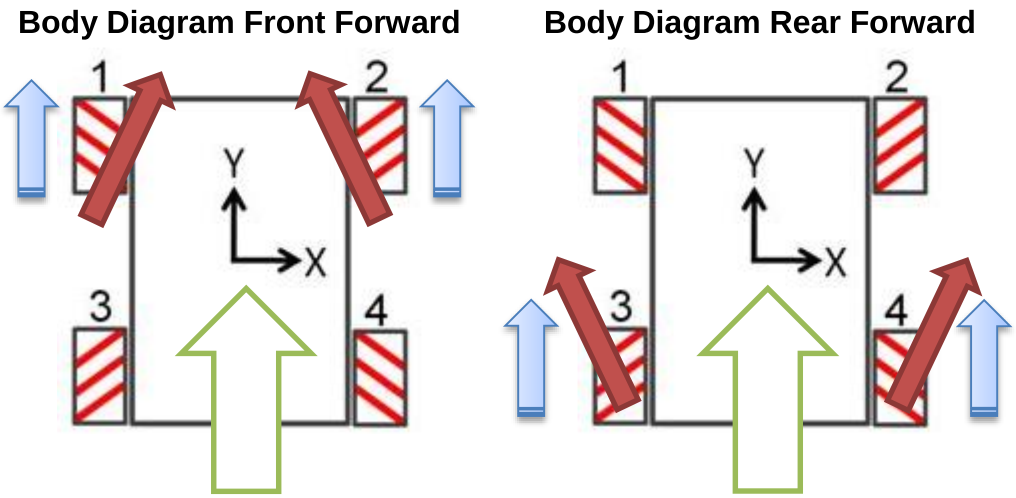


Figure 4.2.1-3 Full Wheel-Set Body Diagram

Every single wheel has to be working together flawlessly in order to perform a precise movement. Keep in mind that the microcontroller executes instructions very fast but it still does it one instruction at a time. All four motors need to be orchestrated seamlessly. Let's look at the first movement.

Forward Translation- By the body diagram convention, the forward direction refers to positive direction of Y axis. In the simplest dynamic function, going forward with Mecanum wheels involves any two wheels that form a resulting force in X direction that will equal to zero.


Body Diagram Front Forward

Body Diagram Rear Forward

Figure 4.2.1-4 Forward Motion Diagram

From the diagram, if wheel1 and wheel2 generate the same magnitude of force, forward motion will be achieved. The resulting force should be solely in Y direction. Also same direction can be generated from the rear wheels with the same exact principle of combining two force vectors. We'll later discuss the different effects on the body by using different sets.

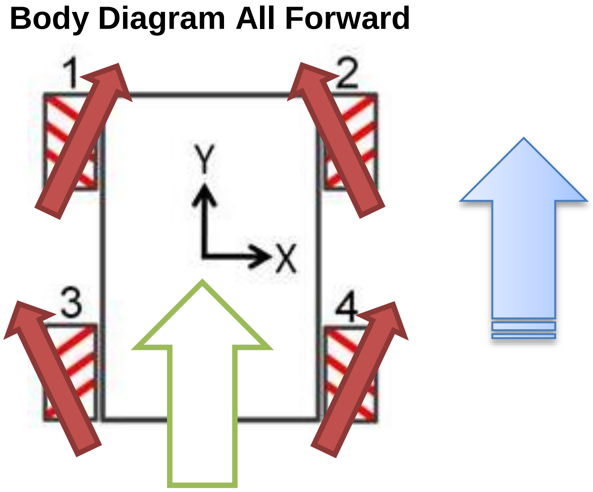

Body Diagram All Forward

Figure 4.2.1-4 Forward Motion Diagram

The forward motion that involves all four wheels requires the most amount of energy. The all-wheel drive mode can offer more torque to the movements that might be necessary to overcome obstacles. Also, many surfaces have uneven plane which could cause certain wheels to lose contact with the ground. Turning on four wheels could be the best answers when all other methods have failed. Lastly, the total cooperation from all four wheels reduce forces impose on the body.

Backward Translation- As simple as it is going forward, backward is the exact same operation in reverse. All is needed is feeding the signal in the negative direction of the same magnitude. Since the forces from the pair are symmetrical, the opposition direction of the forces still produces resulting force on the X axis of zero.

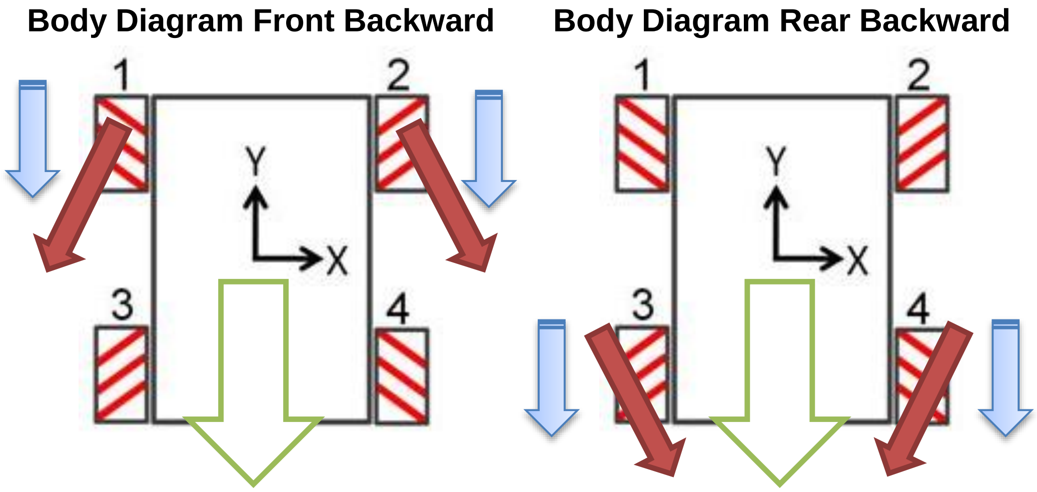
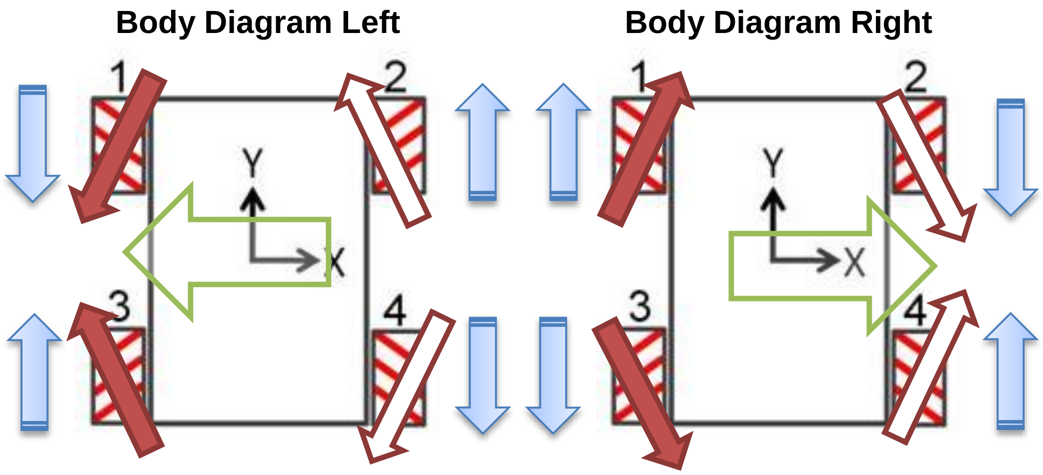

Body Diagram Front Backward **Body Diagram Rear Backward**

Figure 4.2.1-5 Backward Motion Diagram

As far as motion, we're modeling this framework as a solid body and ignoring all the forces in between. Although going backwards still produces the same effect on the X axis as going forward, both modes of translation create different effect on the body of the vehicle. Rotating front wheels forward causes the pressure enforcing into the front section of the body while rotating backward causes the tension on the section. Also the same effects happen on the rear section with the rear wheels. The effect might be very insignificant on the body but this is something that could have noticeable outcome on the suspension mechanics.

Side-to-Side Translation- There's really nothing more to the horizontal translation other than just using the different combination of wheels to cancel out the resultant force in the Y-direction as shown in the diagram below. We can also create force based on the selected pair of wheels.



Body Diagram Left

Body Diagram Right

Figure 4.2.1-6 Side-to-Side Motion Diagram

From the diagram, the pairs are colored coded by solid and hollow. Notice that the pair is now different from the Y-axis translation. As long as the correct pair is picked and the magnitude of the rotation is maintain, the movement should be translating in a constant rate. Typically, Mecanum wheels induce more friction on the ground than conventional wheels. The speed achieved by Mecanum wheels, normally, is not as efficient.

Control Methods- Now that we have discussed the basic concept of the Mecanum wheels, implementing the translation functionality is just the matter of applying the properties of the wheels to the control logic. Ideal, we would like to move in full range of motions, instead of just vertical or horizontal directions. Forward movement with a little bit of tip to the side could be very helpful in some situations. At the same magnitude, each direction has different composition of force. From the Figure 4.2.1-7, direction1 and direction2 are both translating to the right but direction1 is going backwards and direction2 is going forward.

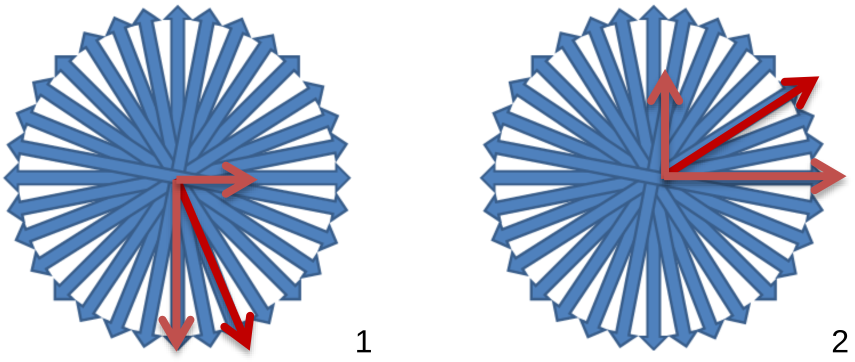
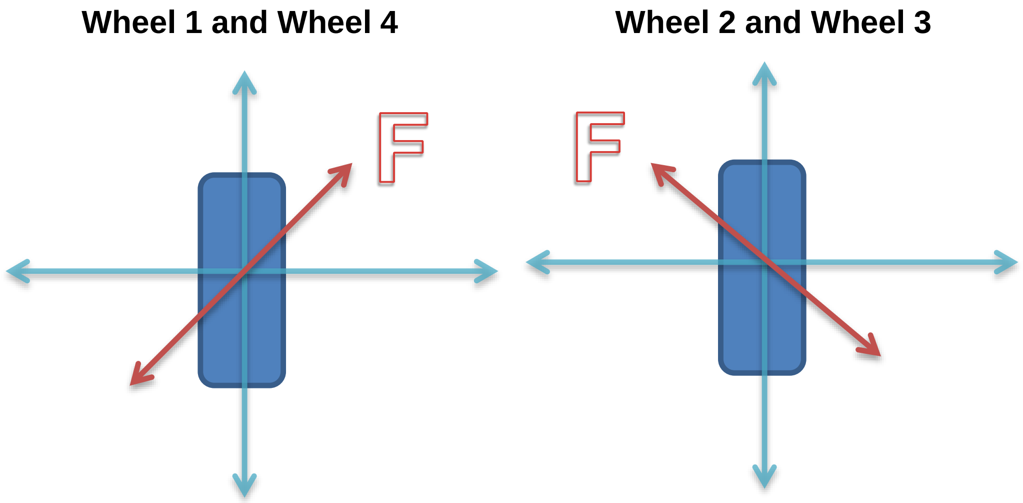


Figure 4.2.1-7 Full Range of Directions

The X-component of the direction1 is less than direction2. Combinations of these components can be emulated by operating the wheels precisely. If inspected closely, we can see that wheel1 and wheel4 create the same direction of force and vice versa with wheel2 and wheel3 as shown below.



Wheel 1 and Wheel 4

Wheel 2 and Wheel 3

Figure 4.2.1-8 Wheel Types

Also, we notice that the wheels create a force 45 degrees off of a standard Cartesian coordinate. Wheel1 and wheel4 only create force in quadrant 1 and 3, but quadrant 2 and 4 allows for low friction motion of the wheels. First off, the value of the angle off of the X-axis has to be calculated from the magnitude of X and Y using inverse of inverse of tangent, then adjust the magnitude of the wheel accordingly using trigonometry functions offset by 45 degrees. The calculations are as follow:

Wheel1 and Wheel4: *Rotation_Magnitude = Sin(Θ – 45)*
Wheel2 and Wheel3: *Rotation_Magnitude = Cos(45 – Θ),*
Where Θ = tan(magY / magX)

The function name will be based on wheel type (TLBR for Top Left Bottom Right and TRBL Top Right Bottom Left). The abstract is as follow:

TLBR(magX, magY)
TRBL(magX, magY).

Both return the magnitude of the rotation of each wheel. The values represent the percentage of the desired speed of each wheel to achieve the direction. They can then be fed into *moveWheel* function for precise wheel control. The calibration of precision should be adjusted accordingly based on the response time of the operating system.

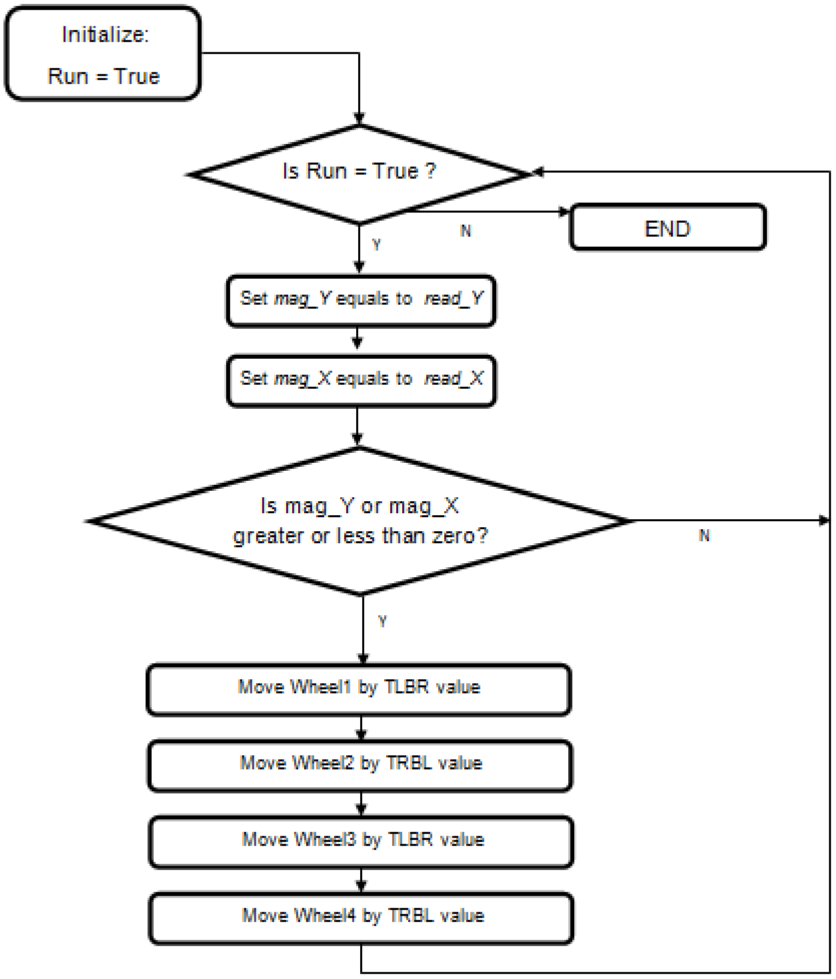

Initialize:
Run = True
Is Run = True ?
N
END
Y
Set mag_Y equals to read_Y
Set mag_X equals to read_X
Is mag_Y or mag_X greater or less than zero?
N
Y
Move Wheel1 by TLBR value
Move Wheel2 by TRBL value
Move Wheel3 by TLBR value
Move Wheel4 by TRBL value

Figure 4.2.1-9 Free Range Translation Control Logic Diagram

Ideally, the control function will be running the entire time of operation. The function will initially set the variable *Run* to true and feed it into our while loop which will run as long as the machine is powered. The variables *mag_X* and *mag_Y* store the magnitude values of the X and Y directions which is retrieved from *read_X()* and *read_Y()* functions. Another logic statement is then implemented to check whether or not the wheels need to be control. If there's not value for either *mag_X* or *mag_Y* we shouldn't have to waste any computing power on the *moveWheel()* function. In best or worst case, the function loops itself back to the *Run* loop to start the process all over again.

4.3 Motor

Choosing the motors for our vehicle was an extremely important task. The motors we choose would need to be small and light yet powerful enough to accelerate our vehicle to speeds capable of keeping up with an average person's walking speed. Price is also a large factor due to the Mecanum wheels each needing to be controlled independently. This means that we will need four motors and four motor controllers. To achieve a speed equal to average human walking speed (about 1.4 m/s) with our four-inch wheels, then we will need a motor to output more than 130 revs/min. Assuming an upper limit for the total weight of our vehicle to be 20 lbs. and a required acceleration of at least 1 ft./s^2, we will then require the motors to have at least 5.2 Nm of torque, or splitting that between the four motors 1.3 Nm each. Because this vehicle is meant to be mobile it will need the motors to run off of the DC batteries, and be power efficient enough to run off of them for extended periods. The motor we have decided to use is the VEX 2-Wire motor 393 (Figure 4.3-1). This motor comes with an adjustable gear ratio to provide both a low speed and high speed mode. In the high speed mode it can reach 160 RPM which is above our requirements; however the torque drops to just under the 1.3Nm we calculated as a requirement. This is an acceptable compromise as that calculation was assuming a higher end range for the vehicles weight. Assuming that the vehicle will come in under the 20lb. estimate, we won't have a problem. If however it does come in at 20lbs. or over, we will have the option to switch the motors back to the low speed gear set, which will give us the torque we would need while only sacrificing some of our top speed. This motor matches all of our criteria within acceptable error as shown in table 4.3-1.

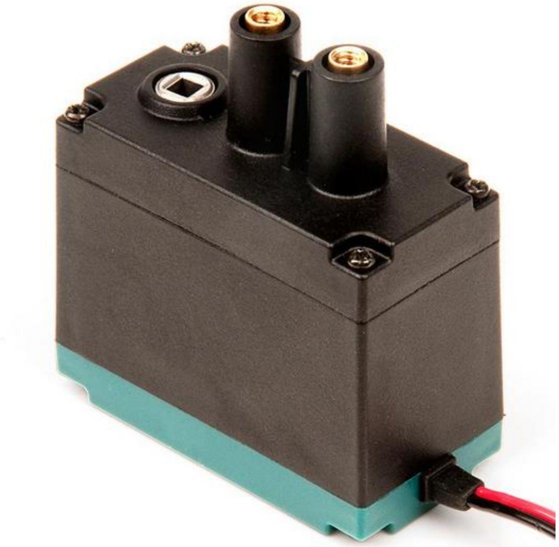


Figure 4.3-1 VEX 2-Wire Motor 393, 1.5 in x 1 in x 2 in (Reprinted with permission from VEX robotics)

VEX 2-Wire Motor 393	Requirements	Specs-Low Speed	Specs- High speed
Price	< $20	$14.99	$14.99
RPM's	130	100	160
Torque	1.3 Nm	1.67 Nm	1.04 Nm
Weight	< .5 lb.	.2 lb.	.2 lb.
Stall current		4.8 A	4.8A
Max Power	All measured at 7.2 V	34.56 W	34.56 W

Table 4.3-1 Vex Motor Specifications

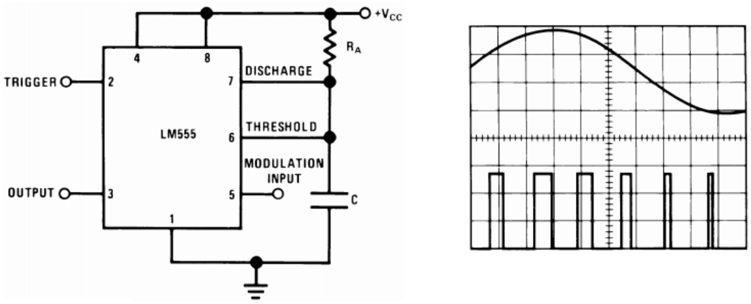
4.3.1 Motor Controller

The motor controllers are essential components in making the vehicle mobile. Since the motors we will be using are going to be brushed DC motors the controllers will be how we can run the motors in both forward and reverse, as well as adjust the speed of rotation. They will also give the control to turn the motors off completely, to the processor. Since we are using Mecanum wheels to give our vehicle added mobility options, we will need the four motors to be controlled independently, and this means either a four port motor controller or four separate motor controllers will be needed to allow all of the variable configurations which will let our vehicle to move in any direction without the need to turn.

Biasing the motors is the quintessential function of any DC motor controller. By forward or reverse biasing the motor we are able to control whether the motor will turn clockwise or anti-clockwise. The best method of performing this function electronically is by means of an H-Bridge relay. While this simple circuit will allow us to forward and reverse bias the motors very simply, we are still stuck with simple control of the motor (forward, reverse) with no way of controlling the speed of the vehicle. This can be solved by adding an oscillating circuit to switch the H-Bridge on and off, and by varying the duty cycle of a square waveform function we can alter how long the motor is on during each cycle allowing us to control the speed. This method is known as Pulse Width Modulation, or PWM.

The defining function of the H-Bridge relay is the use of transistors as electrical switches. This allows us to control the current by controlling which path in the circuit is open for current to travel freely. Paths are opened and closed by biasing certain transistor pairs in such a way that the free path for the current to travel through includes the motor, with the other pair allowing the same but with the current flowing in the reverse direction through the motor.

One of the most popular methods of creating a PWM signal uses the LM555 timer. This is a highly stable device that can generate very accurate time delays or oscillations. By triggering this device with a continuous pulse train, we can modulate the pulse width with the signal applied to the control voltage pin on the device. This can give us a modulated output, as shown in Figure 4.3.1-1.



Pulse Width Modulator

Input signal over output modulated signal

Figure 4.3.1-1 (Reprinted with permission from Texas Instrument)

Many micro controllers have the ability to create a PWM signal, however not many can produce 4 separate signals, this leaves up with three options. We can create four separate PWM circuits which will be controlled from the microcontroller with a simple analog signal. We can use logic circuits and digital pins to control the H-Bridge Relays and share a PWM signal between the motors for controlling speed. Or if the microcontroller has the ability to create four PWM signals, then we can simply use those. The first method would give us the most control over the motors, and allow us to use many store bought motor controllers that are controlled with only a PWM signal. With the second method, we could use a single PWM signal from the microcontroller to control all four wheels, limiting our ability to control each wheels speed independently from the other wheels.

Vex motor controller 29- This is the motor controller that Vex Robotics has made to control the 2-wire motor 393 with their proprietary control units. It has a max voltage of 8.5V and a max current of 4 amps, and is controlled with a PWM input (1-2ms from full revers to full forward, with 1.5ms being neutral). This would be a very good candidate for a motor controller as its specifications are made to the normal operating specs of the motors that we are looking at using. The only downside is that they require a PWM signal to operate, so unless our microcontroller can generate 4 PWM signals then we will still have to add a way to generate them in the design. These controllers cost around $10 each.

Figure 4.3.1-2 VEX motor controller 29 (Reprinted with permission from VEX robotics)

Jaguar- This is a much more robust controller, capable of an operating voltage up to 24 V and 40 amps continuous. It supports three different input types. The first is the standard RC style interface, using pulse width modulation (PWM). The second is a controller area network interface (CAN), which is a standard protocol that allows a microcontroller to communicate with multiple devices through a CAN bus. Finally the jaguar supports standard serial (RS232C) interface. Along with these extra input methods come a slew of other features, such as control of motor current, position, and voltage measurements. These added features are not necessary for our needs and would only serve to further complicate the coding of the vehicle

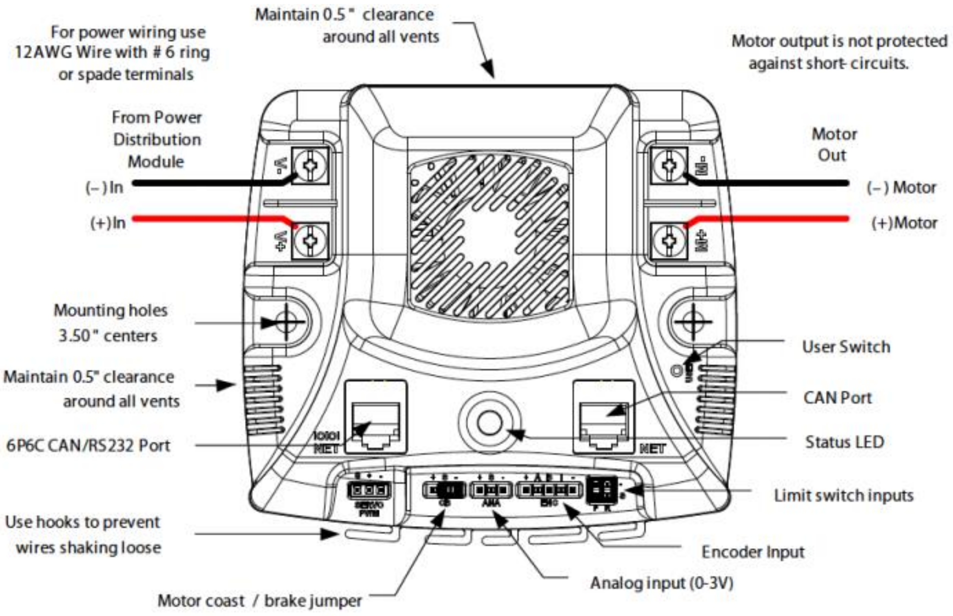


Figure 4.3.1-3 Jaguar Motor controller (Reprinted with permission from VEX robotics)

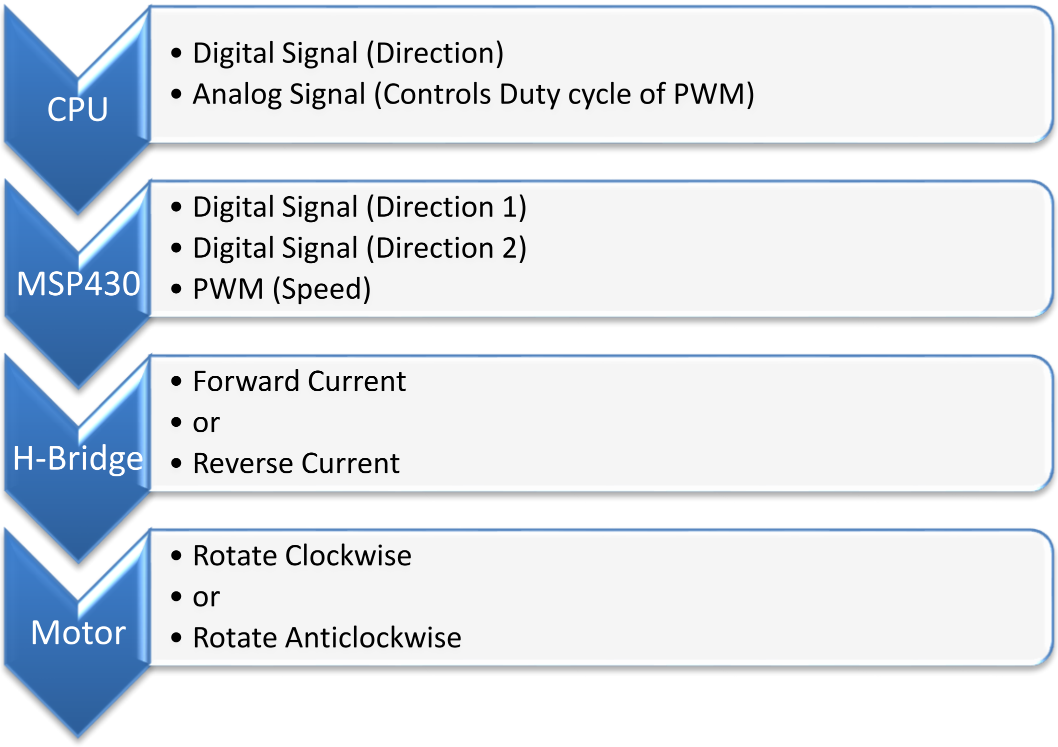
Designing a Motor controller- A third option would be to design our motor controllers and build them from component parts. As shown previously the circuitry is fairly simple and there are vast amounts of resources to help with designing and building them available. One of the biggest choices to make when designing a controller like this comes from which type of transistor to use for the switches.

Bi-polar Junction Transistors- these are a common choice due to availability, size, and price. It is also a simpler circuit to deal with. The disadvantages are that compared to a FET they have slower switching times, and consume more power. Another major drawback is that each transistor will cause a voltage drop of around .7V (assuming silicon) this means that in the basic H-Bridge configuration there will be 1.4V being dropped over the transistors, rather than the motor. If we were to use a 7.2 V Battery than the motor would only be able to use 5.8V due to the voltage drop across the two transistors.

Field Effect Transistors- these are the other choice when choosing transistors for building an H-Bridge. These transistors have faster switching times and consume less power. However the circuit design for them is a bit more complicated, they are larger in size and more expensive.

H-Bridge Chips- These are integrated circuit chips that have the components of an H-Bridge built together in one convenient package. In a lot of cases it’s unnecessary to build an H-Bridge from scratch due to these, they can save a lot of hassle in choosing and designing from scratch and having to deal with offset voltages. Many of these chips also come with more than one H-Bridge built into the package which saves on space and costs

Design process- To save processor power on the main CPU we have decided to create our own motor controllers, which are capable of generating their own PWM signal for speed control, so the main CPU has only to send a simple analog signal rather than creating separate PWM signals for each of the motors. This will also provide us with a more cost efficient way to control the motors, since we can build the controllers to the specific specs of the motors, rather than using the costly motor controllers that cover a broader range of applications and are loaded with features that we don’t intend to use. The controllers will use an MSP430 to generate the PWM signal to control the speed of our motors. It will also control the H-Bridge circuitry to alter the direction the motor is rotating. The MSP430 will receive two signals from the processor, an analog signal which will be used to alter the duty cycle of the PWM being generated, and a digital signal to convey the desired rotation direction. The MSP430 will then alter the PWM that it is generating and send it along with two control signals to the H-Bridge. The two control signals will determine which of the transistors in the H-Bridge will allow current to flow through them, given certain combination of signals the current will be allowed to flow through the motor in one direction, and given a different combination the current will flow in the opposite direction. The PWM will be applied to a separate transistor which will act as an enable line for the circuit. With the Duty cycle of the PWM determining the duration that the current is allowed to flow during each cycle, this is how we can control the speed of the motor.



CPU

- Digital Signal (Direction)
- Analog Signal (Controls Duty cycle of PWM)

MSP430

- Digital Signal (Direction 1)
- Digital Signal (Direction 2)
- PWM (Speed)

H-Bridge

- Forward Current
- or
- Reverse Current

Motor

- Rotate Clockwise
- or
- Rotate Anticlockwise

Figure 4.3.1-7 Flow Diagram for the motor controller

The basic schematic for the H-Bridge circuitry consists of the four power-transistors arranged in the circuit's namesake H formation around the motor (shown as a multi-meter in the schematic below). There are also some logic gates used for controlling the power transistors and integrating the PWM into the direction signal to allow for speed control. In this schematic the NAND gate coming from the direction input signal acts as an inverter, this allows the circuit to be on in only the forward and reverse configurations. While the AND gates are used to integrate the PWM into the direction signal modulating the time that current is allowed to flow per cycle. This is a much over simplified schematic for the H-Bridge, however it allows us to lay out the circuit and helps visualize how we want the H-Bridge to perform.

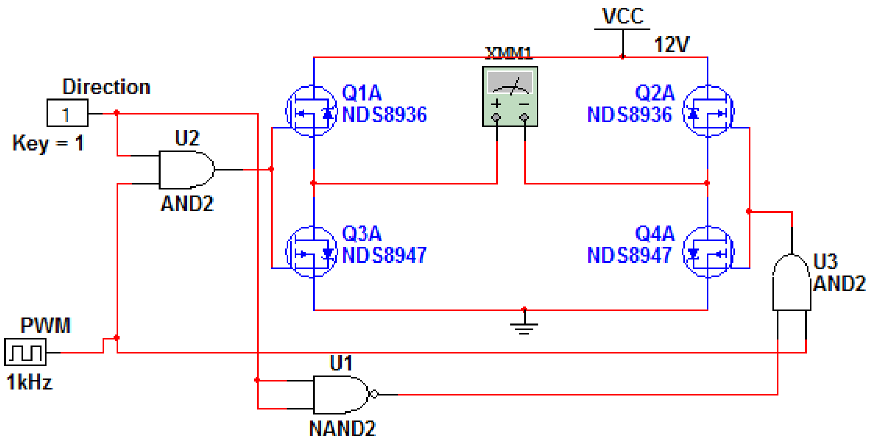


Figure 4.03.1-8 Basic H-Bridge Schematic

The full design for the H-Bridge will contain “pull up” resistors on the gates of the field-effect transistors to give us the gate voltages required to turn them on completely and allow current to flow. We will use bi-polar transistors in place of the logic gates to activate the field effect transistors; this gives us a nice buffer between the full power section of the circuit and the control lines. Each field effect transistor will have a bi polar transistor attached to its gate which will control the current allowed to flow through the “pull up” resistor which we will be using to activate and deactivate the field affect transistors. Normally we would want to add “Flyback” diodes in parallel with our field affect transistors to prevent damage to the circuit and power supply from the motor generating and sending unwanted energy into the circuit, however the field effect transistors I have chosen should have powerful enough “body diodes” built in that we should be able to avoid this problem. If however we do see issues with this we can always add the extra diodes later. To prevent the two of the field effect transistors on either side from both being active, thus shorting the battery, I have connected the emitter of Q5 and Q7 to the opposite direction signal from their base, this way if both signals go high then the base emitter voltage will drop to zero and the P-Channel field effect transistors will both switch off. This will essentially ground both terminals of the motor giving us breaking functionality. The final bi-polar transistors in the circuit form the AND gates that we will use to add the enable line to our circuit for the PWM to control. Finally I have added a bypass capacitor in parallel with the motor terminals.

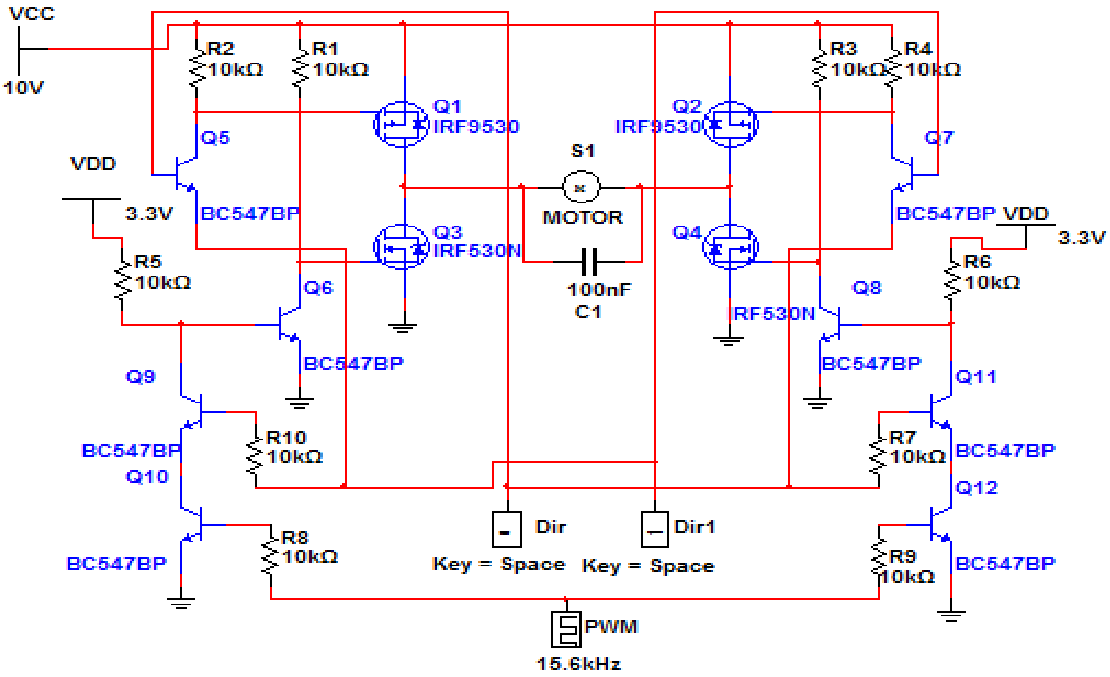


Figure 4.3.1-9 Full H-Bridge Schematic

For the Field effect transistors we chose the IRF 9530 P-Channel, and the IRF 530 N-Channel. These Devices are rated to handle a maximum Drain-Source Voltage of 100 V. They can handle a continuous drain current of up to 12A and 14 A respectively. The BC547 NPN bi-polar transistor was selected for the rest of our transistors. It has a breakdown voltage of 45V and a collector current max of 100 mA, with a gain of 200. The capacitor will be chosen to have a voltage rating of at least 30V (double the typical load) to ensure that it isn't overloaded during operational changes. The price table shows that we will be able to build all four motor controllers for less than $50, which is cheaper than any of the other motor controller choices we came across.

Component	How many	Price per unit	Total
MSP430	4	2.79	$11.16
IRF9530	8	1.29	$10.32
IRF530	8	1.03	$8.24
BC547	64	0.20	$12.80
Capacitor(100nF)	4	.05	$0.20
Resistor (10k)	40	.04	$1.60
		Total (without PCB)	**$44.32**

Table 4.3.1-1 Table of motor controller components and their prices

4.4 Sensor

Due to the information we found, as noted in the research section, we decided to use the HC-SR04 (Ultrasonic) sensors. We will have one sensor attached to each side of the vehicle, for a total of four sensors. They will each be attached directly to the processor, and provide an analog signal to relay information about nearby obstacles to the processor.

4.4.1 Integration Changes

We decided not to implement the sensors in the final design of our vehicle. The reason was because we were having a lot of problems with the autonomous feature on the robot. The cause of this was the slow frames per second that the camera was producing. Since the autonomous feature was partially working, we decided not to use object avoidance and just let the robot detect the motion of the person and rotate in the direction where the person is.

4.5 Power

Power Design-The first aspect that will go into consideration is how the power supply will work. There are two considerations for the design of the power supply. The first scenario is that we are going to have a separate power supply power the different subcomponents. The second scenario is having a main power source power all of the components. There are several pros and cons to both of the different scenarios. Having multiple power supplies can be a good approach because this is a sure way to know if each sub component is receiving the right amount of voltage when testing. The down side to this approach is that we don’t need a lot of voltage to power each sub system individually, so it can be done by having multiple power sources but is not really needed.

The positive aspects of having one power source are that it will be a neater setup and fewer wires when wiring the project. This may not be a big concern but while building we want to eliminate all errors and by wires touching that can cause conflict in our circuitry. In our project we have decided to build one power system to power the every component of the robot. The main decision in building the power system is we need to find battery to power the vehicle. This is very important aspect in order for the vehicle to achieve the project requirements. So we have to make sure that the correct battery is selected. If we do not provide the right amount of power for each of the different sub components then ideally, our vehicle will not be efficient. The battery requirements are:

- Lightweight
- High Discharge
- No/Low Memory Effect
- High Capacity

- Price Effective
- Moderate Nominal Voltage
- Provide 16 A of current to support motors
- Needs to provide at least 12V
- Battery Life needs to be 4 hours

Based off these requirements we came to the conclusion that we can either buy a high voltage battery and use that at the power source or have the smaller ones in parallel to produce the voltage requirements. The voltage that I am focusing on getting is the 7.4V for batteries. We decided to go with smaller batteries because the larger voltage ones are heavy and our frame can only support up to 20Ibs. Below in the figure is the comparison of different types of batteries with prices.

Model	31604	11427	18650
Brand	Tenergy	Tenergy	Supower
Discharge rate (mAh)	1600	2000	2200
Voltage (V)	7.4	9.6	7.4
Weight (lb)	0.2	0.52	0.2
Capacity	20	10	2.5
Price	24.5	19.99	17.99
Chemistry	Lipo	NiMH	Li-on

Figure 4.5-1 Battery Comparison

From research we learned that NiMH, Lipo, and Li-on batteries all provide a very sufficient battery life. We are leaning towards using the Lithium Ion (Li-on) or the Lithium Ion Polymer (Lipo). This is because (as you can see from the comparison) the capacity is not as high and the battery weighs 0.52Ibs for the NiMH batteries. The Lipo batteries as you can see meet all the qualifications wanted. The capacity plays a major role in selection because this rating of energy that can be put out. There is a very big margin between these batteries in this category. From our requirements it was said that we needed to have at least 16A to support the motors. With that being said, we think that it will be more efficient and safer to have two different batteries in parallel in conjunction to having one battery powering everything. This is because in order to figure the peak current that a battery can discharge you have to use this equation: $\text{Capacity (C) x Discharge Rate (mAh)} = \text{Max Current Output}$. The discharge rate is only the amount current for one particular hour but it serves as a warning not to exceed that rating when in performance. Because we are using two batteries and not one, we need to have at least 8A for each battery. Referring back to Figure 4.5-1; looking at the Lipo and Li-ion batteries. The Supower battery is low in cost and has a higher discharge rate but the capacity is way less than the Tenergy battery. The Supower battery will only produce 5.5A per battery, which is not acceptable. On the other hand, the Tenergy battery produces about 32A. For about $7 more dollars we can get way more than enough current needed. This is great because we have way more than enough current for the battery.

Battery Selection- The battery shown in figure 4.5-2 is the battery that we selected to use for our power system. This is the Tenergy 7.4V 1600mAh rechargeable battery pack. From the comparison figure you can see that the battery provides a lot more amps and capacity than all the other batteries. The battery does not suffer from memory effect and can provide more power than a 12V NiMH battery with it only being half the size. This is great because it's lightweight so we don't have to worry about the weight making our vehicle move.

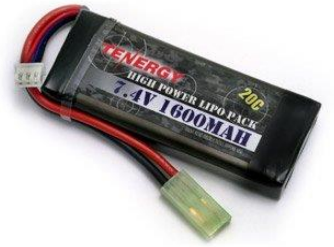


Figure 4.5-2 Tenergy 7.4V 1600maH Battery, 87mm x 35mm x 14mm (Reprinted with permission from Tenergy)

This battery was found on Amazon.com and is priced at $24.50. This is the lowest price we found for now but we have reached out to several local battery shops and are waiting on further communication.

Charger Selection- Figure 4.5-3 shows the Tenergy 1-4 Cells Lipo balance Airsoft Battery Charger. The reason for picking this charger instead of designing one is because this is the more efficient and safer choice. Many different sources who actually tried to develop chargers for these types of batteries tend to fail. They end up overheating and damaging the battery.

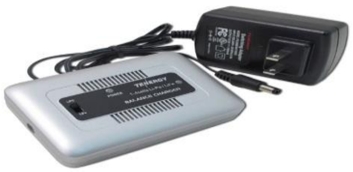


Figure 4.5-3 Tenergy Airsoft Battery Charger, 88mm x 57.3mm x 16.8mm (Reprinted with permission from Tenergy)

Since we have a lot of special components on the vehicle we need to take the safer route and use this charger. Some of the features of this charger are:

- Built in balancer
- Applicable with Li-on and Lipo
- Safer and faster charging process

- Automatic Charging process
- Input: Dc 10—16V, 1.5A

The price of this battery charger is $19.99 on Amazon.com. In addition, we have contacted a few local retailers and are waiting on pricing information.

Voltage Regulator – The voltage requirements from Table 4.5-1 were utilized as reference to create the circuits for the voltage regulators.

Items	Voltage Input needed	Current input need
BeagleBone Black	3.3V	0.5A
*Motor Controllers	7.2v	16A
*Thermal Camera	3-5.5V	<600ma
*Conventional Webcam	3v	<500ma
*Proximity Sensors	5v	15ma

*These components will be connected to BeagleBone Black

Table 4.5-1 Voltage requirements

Motor Controller- The type of Voltage Regulator that the Motor controller will use is a TPS4531 step down converter with Eco-mode. This voltage regulator was found by using the Ti Power Architect for multiple loads. A schematic is shown below:

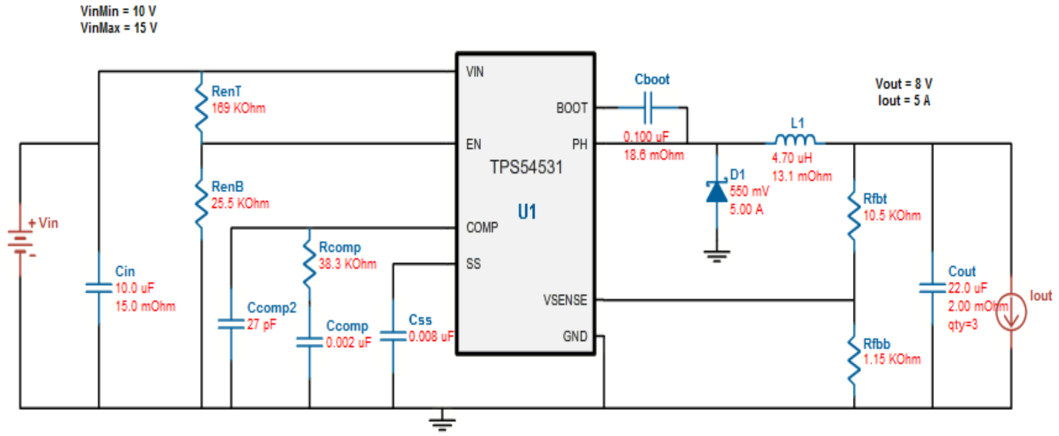


Figure 4.5-5 Voltage Regulator for Motors

Microcontroller- The type of voltage regulator that the microcontroller will use from the circuit below is a TPS54229 2A Synchronous Buck Converter. This voltage regulator was found by using the Ti Power Architect for multiple loads.

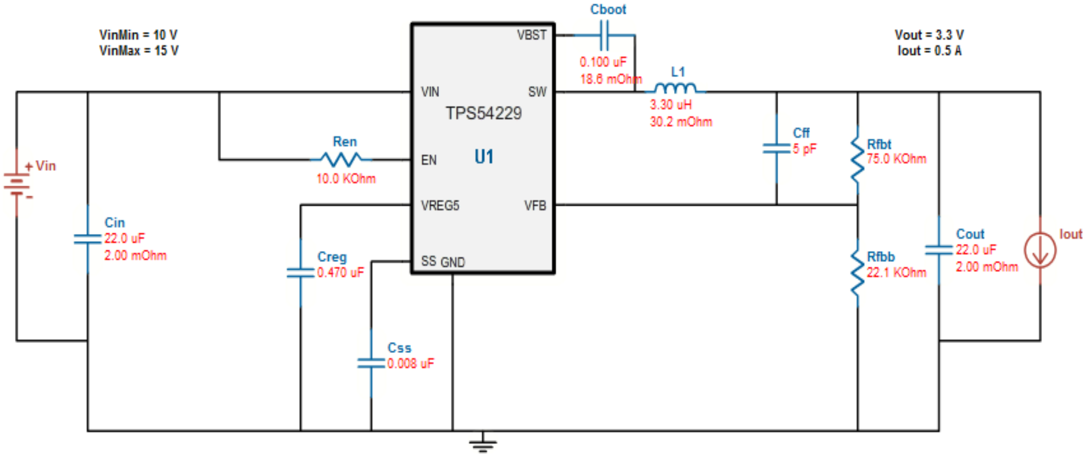


Figure 4.5-6 Voltage Regulator for BeagleBone Board

As you can see from both schematics, most of the voltage needed is for the motors to make the wheels run. A block diagram of how power will be distributed throughout the system is shown below:

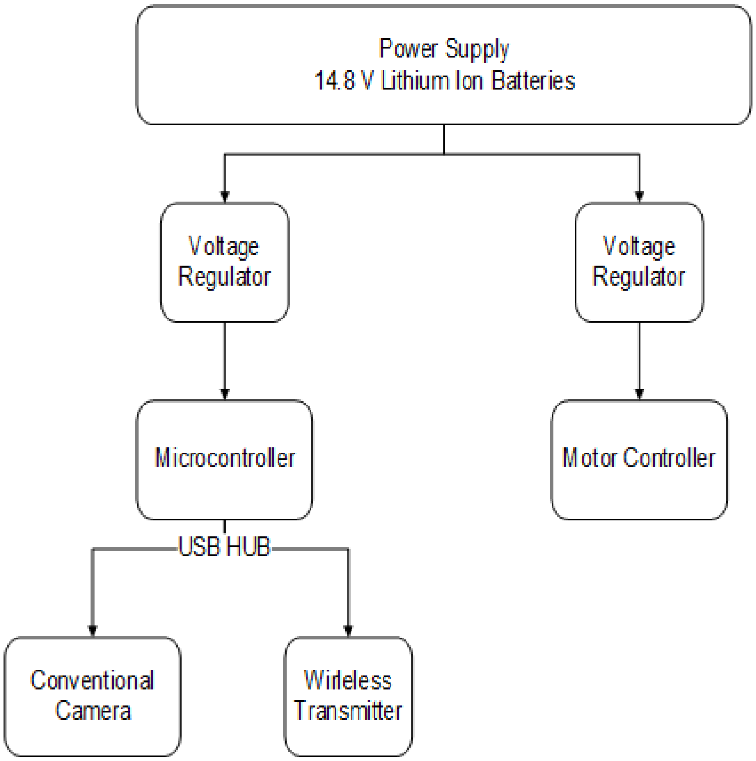


Figure 4.5-7 Power Supply Block Diagram

From the figure above you can see the breakdown of each of the different components. We have two 7.4V Lithium Ion Polymer battery which serves as the power source for the vehicle. We will have two switcher voltage regulators that will regulate voltages to the motor controller and microcontrollers. As you can see above, we are planning to connect the thermal camera, conventional camera, wireless transmitter and proximity sensors to the microcontroller. The microcontroller we are planning to use requires 3.3 V for operation. For the motor controller the requirements are;

- Each of the 4 motors need 4A (16A total)
- Needs 7.2 V to operate
- Max weight has to be under 20 lbs.

4.5.1 Integration Changes

There were several changes in the power system design due to the changes we made on our sub components. We decided just to buy a 14.8V 4400mAh lithium ion battery (motor controllers) and a 7.4V 6600mah Lithium ion portable battery supply for the BeagleBone Black. We decided to use this instead of two 7.4V in series because of the limited space that we had on our chassis and because we wouldn’t have to worry about charging a lot of batteries. We took away some sub components and changed the ratings of some loads. The motor controller’s voltages were stepped down due to the fact that the motors where running extremely too fast. With that being said, the voltage from the battery needed to be stepped down to 10V (not the 12V or 14.4V as specified). For the voltage regulators we didn’t use the buck converters due to the amount funding we had left. We also made 4 printed circuit boards, as a result we decided to use some LM three-pin voltage regulators for the sake of money and time purposes.

4.6 Computer Vision Controller

4.6.1 Video Image Processing

The video image processing will be divided in three subsystems to facilitate the implementation. The three subsystems are Movement Detection, Acquire Target, and Target Tracking. The input will be coming from the webcam or thermal camera depending on the time or light conditions. A series of frames will be stored in memory which will be utilized by the OpenCV algorithms for the necessary computations. A subsystem that will always be running will be the Movement Detection subsystem to determine if any movement has occurred in the current room where the vehicle is stationed at. Only when a movement is detected is when the other two subsystems will be activated. Once activated, the Acquire Target subsystem will perform a window based search in order to find the target that triggered the subsystem. Once the target has been found, it is sent to the Target Tracking subsystem which will utilize it as input. From here, another window based search will be performed and the top left corner of the target will be stored as an object. Once the target's position is stored in an object, its motion can be tracked in any direction. Each time the position changes, the motor controllers will be notified in order to move the vehicle and keep the target in the vehicles view. Below is a diagram of how all the mentioned subsystems function:

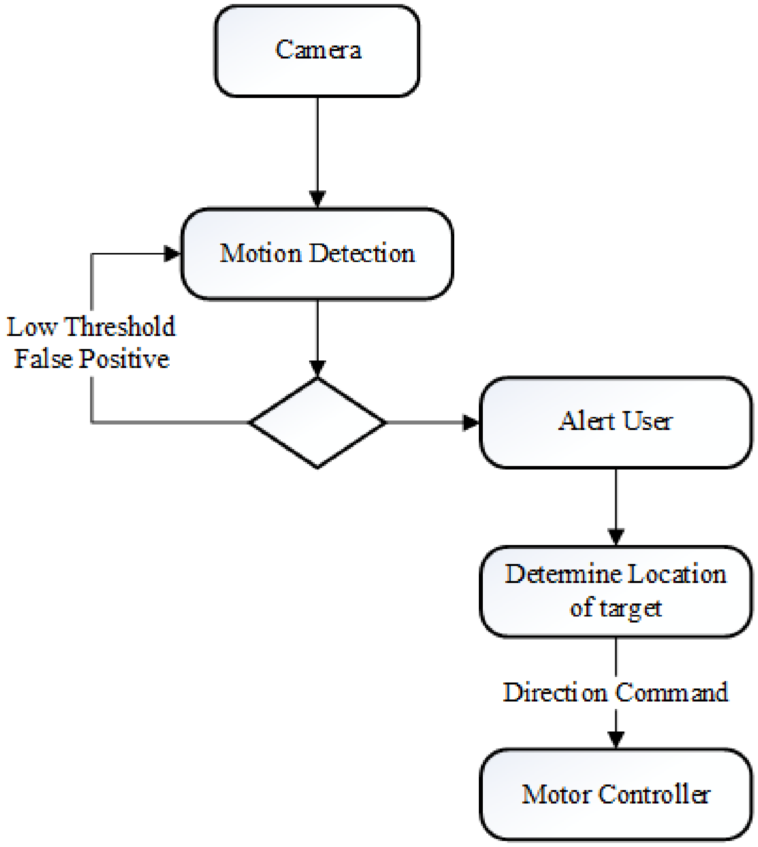


Figure 4.6.1-1 Video Image Processing Diagram

4.6.2 Cameras

The project will make use of the Logitech C300 as a Webcam and the Tamarisk 320 as the Thermal camera. Depending on the time, day or night, each camera will be toggled. During the day, the Webcam will be utilized since it will provide a better view. During the night, the Tamarisk 320 will be activated in order to show a detailed heat signature. For the position of the cameras, various positions will be tried and tested to see which one provides the best result during the testing phase.

4.6.2.2 Integration Changes

We only used the Microsoft LifeCam Camera. We were unable to convert the analog signal of the thermal camera to a format that our development board could understand. We went through a number of different alternatives to deploy the image from the thermal camera to display but each method failed. As for the

Computer Vision algorithm, we had to do a combination of background subtraction to detect motion and also to track the movement of the target. We had to resort to the background subtraction method due to compatibility issues and running out of time for the testing phase. Even with background subtraction, we were only able to get 1-2 frames per second. After careful debugging we found out it was a problem with the driver being utilized. The driver would toggle on and off the camera while taking pictures resulting in the1-2 frames per second. Although the vehicle moved while taking pictures, the background subtraction method worked pretty decent.

4.6.3 Video Streaming

The input of the video streaming will be the Logitech C300 Webcam. Since OpenCV will always be running, the camera will always be on but the streaming subsystem is only activated once a user opens the mobile application and a request is made. This will allow us to conserve resources such as processing power and battery power. Once a request is made, the streaming subsystem activates the input plugin which starts saving image frames from the camera to the internal storage of the Beagle Bone board. Once there's a specific amount of frames stored, the output plugin is activated. The output plugin will then either stream the images in a video form or output a single frame depending on the user's command. A state diagram of the video streaming subsystem is shown below depicting all the possibilities.

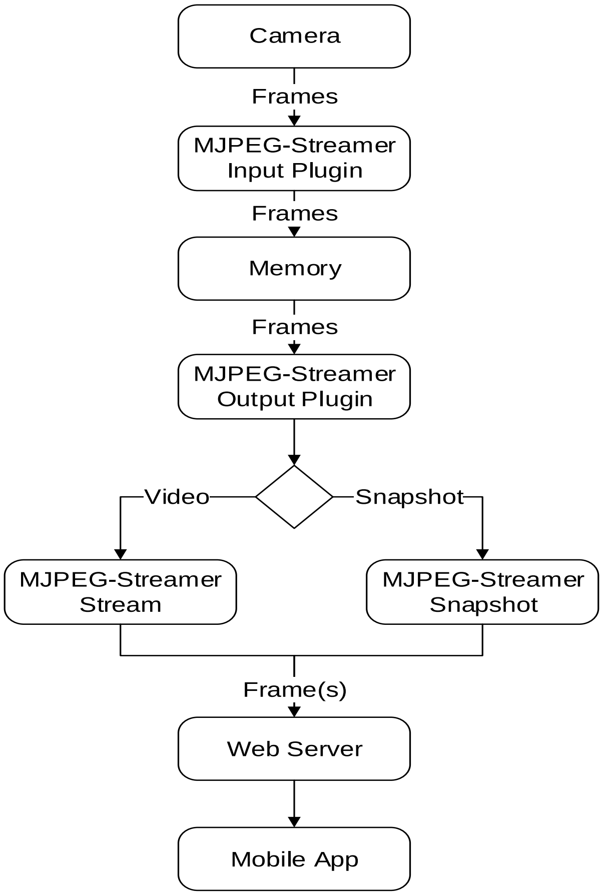


Figure 4.6.3 Video Streaming Diagram

4.7 Mobile Application

Description- The mobile application plays a crucial part in the project. It will allow the user to view a live video feed of what the robot is seeing as well as have the option to control the robot. Since the application is going to be able to stream a video feed and is done via the internet, it's necessary to apply security measures. The video feed will be available via the internet by opening a port in a wireless router/modem. Since the mobile application will be compressed as an application package file (APK), it is safe to define the port that will be used in the application's program code. In order to protect the privacy of the user, once the application is launched the user will have the option to create an account along with a username and password. Once the account is created, the credentials will be stored in the internal storage of the device using Cryptography as a form of encryption. The encryption will be done by using Android's built-in functions that can generate a key for the encryption as well as the decryption. Each time the application is launched, the user will be asked to login. As a result, if anyone else takes unauthorized possession of the phone, it will not be possible to open the application. In addition, once the application becomes a background process, meaning that is not the currently active application, it will be terminated immediately. As a result, the user will have to login again.

Design- The application will consist of three different sections, the registration/login section, the registration section, and the control section. The registration/login section will be where the user will have the ability to either register for an account or login with an existing account. If the user already has an existing account, the user will input their username and password. From here, the username and password will be tested to see if they match with the username and password the user used for registration. Below is a diagram of the graphical user interface (GUI) of how the registration/login section will look like:

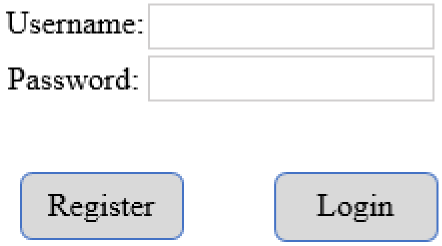
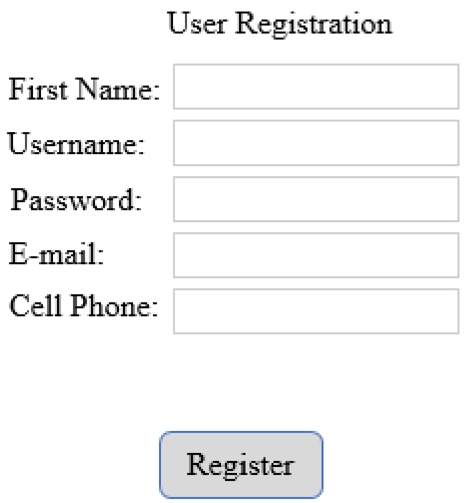


Figure 4.7-1 Login Form

If the user doesn't have an existing account, the user can choose to register an account. Once the user picks the option to register for an account, the user will be directed to the registration section. From the registration section, the user will be asked the following: First Name, Username, Password, E-mail, and Cell Phone number. The first name will be utilized to create a customized welcome message in the login/registration section as well as the control section. The username and password will be the credentials that user has to use in order to gain access to the application's functionality. Lastly, the e-mail and cell phone number will be utilized in order to alert the user whenever the robot has detected any movement. Below is a diagram of the graphical user interface (GUI) on how the registration section will look like:



User Registration

First Name:

Username:

Password:

E-mail:

Cell Phone:

Figure 4.7-2 Registration Form

If the user logged in successfully, the user will be presented with the control section. In the control section the user has the ability to view the live video feed transmitted via the internet as well having the ability to save a snapshot of the video feed. In addition, the user will have the ability to control the robot by using the virtual joystick in the application. In case there’s an emergency and the user requires calling 9-1-1, a button will be included in the control section where once activated the application will make the call to 9-1-1 for the user. Below is a diagram of the graphical user interface on how the control section will look like:

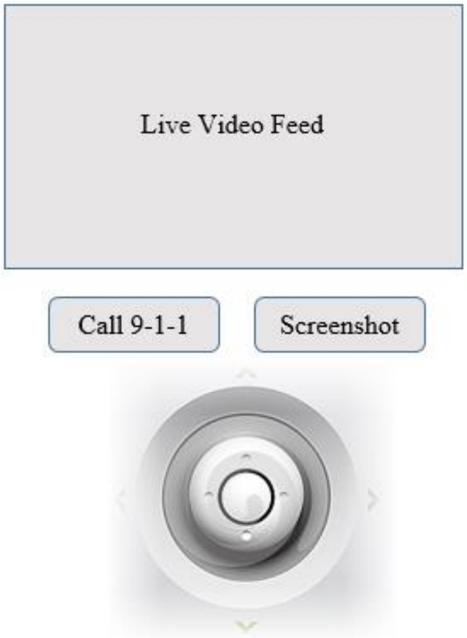


Figure 4.7-3 Control GUI

State Diagrams- The mobile application has various section and each serve a different purpose. Below are detailed state diagrams for each section depicting all the possibilities that the user may encounter.

Registration/Login Section-

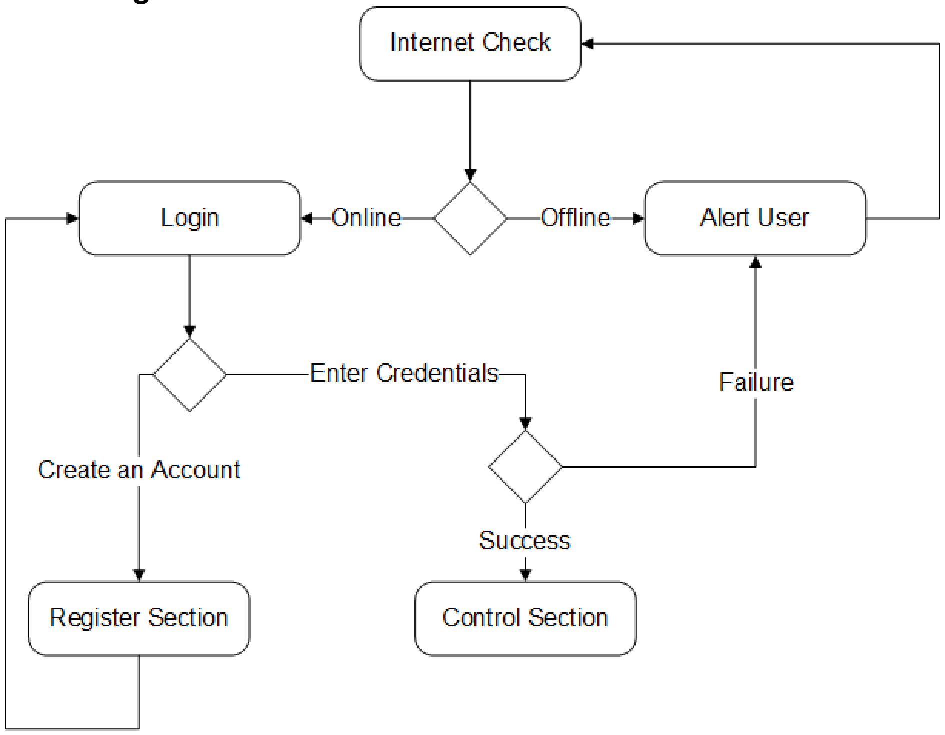


Figure 4.7-4 Mobile Application Registration/Login State Diagram

Register Section-

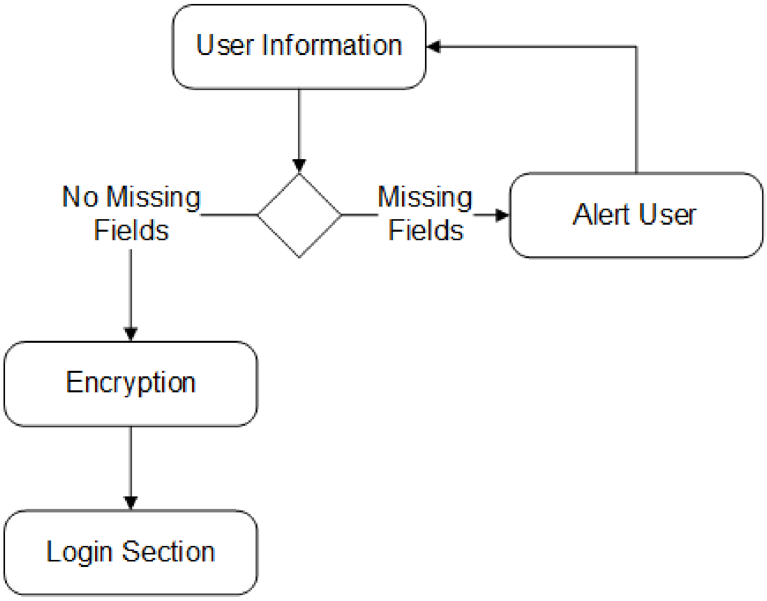


Figure 4.7-5 Mobile Application Registration State Diagram

Control Section-

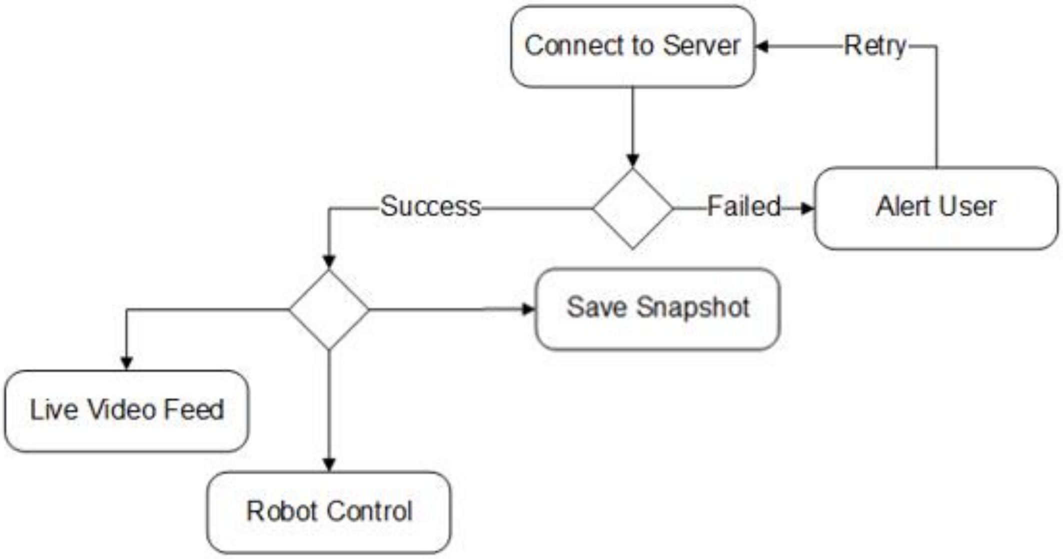


Figure 4.7-6 Mobile Application Control State Diagram

Class Diagram- In Android, each section or screen is referred to an Activity and each Activity is a JAVA class. In order to go from Activity A to Activity B an intent is required. An intent is an “intention” to perform an action, in this case, to switch between activities. The mobile application will consist of three activities, the first activity will be where the user has the option to register or login with their existing account. The second activity will be the registration activity which will allow the user to create an account. The third activity will be the control activity where the user has the ability to control and send commands to the robot. Below are the class diagrams along with the necessary methods and functions:

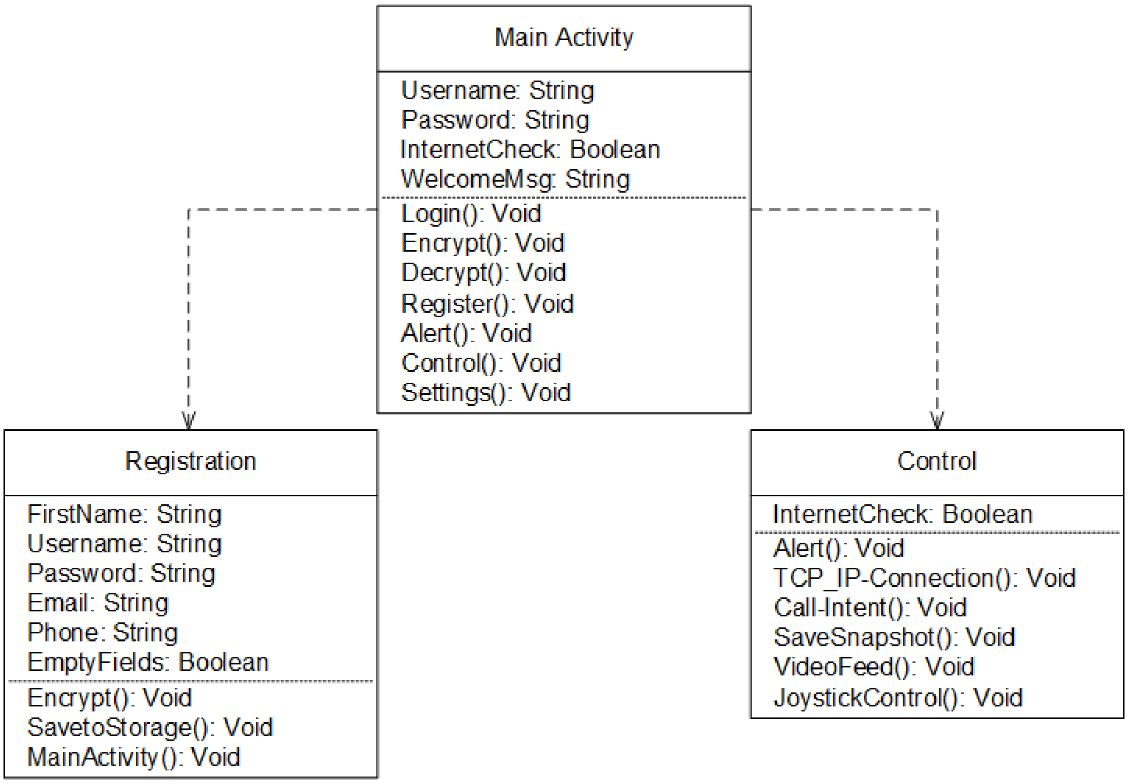


Figure 4.7-7 Mobile Application Class Diagram

The Main Activity will present the user with the graphical user interface (GUI). From the Main Activity depending the user's choice, intent will be made to either show the Registration Activity or the Control Activity.

4.9 Web Server

The backend Web Server software that will be used will either be Nginx or Node.js. Both of these backend servers performed very well during the benchmark phase by using very little CPU and RAM. The sole purpose of the backend Web Server software will be to enable an open port, making it visible to the Internet. The Web Server that will be utilized will consist of a simple TCP/IP (Transmission Control Protocol/Internet Protocol) which will allow the mobile application (client) to connect to the robot (server) by always listening for incoming connections. Since Android is being used for the mobile application, the TCP/IP will be implemented in JAVA. The main function of the TCP/IP is to make the web server available over the Internet as well as send/receive commands between the server (robot) and client (mobile application). Once the robot is connected to the Internet it will be assigned an IP address which can be used by the mobile application to connect to it. From the mobile application the user has the option to control the robot by using the virtual joystick. Depending on the position of the joystick, a string containing a pattern of values is generated and sent to the robot (server). The string format is as shown in the table below:

X	Y	Angle	Direction

Table 4.9-1 String Format

The X and Y values will control the direction of the robot in the X and Y direction while the Angle value will determine what angle the robot needs to translate at. The Direction variable is only turned on for debugging purposes which allows us to test the control algorithm. Once the string message is received by the robot (server) the message will be parsed and sent to the motor controllers. Depending on the values received, each motor will be activated and a specific amount of voltage will be applied which will be handled by the motor controller.

5.0 Design Summary

5.1 Hardware

Due to the complexity of the PCB required for our main CPU board to be able to load an operating system onto, we have elected to use the BeagleBone Black development board and will incorporate the PCB design requirement into our motor controllers. The main hub for all of our hardware components will be the beagle bone development board. All of our hardware components, with the exception of the motors, will be directly controlled by the BeagleBone. The motors will be controlled by MSP430 micro controllers which we will use in the motor controllers to create a PWM and to send control signals to the H-Bridge circuit. A diagram can be seen below:

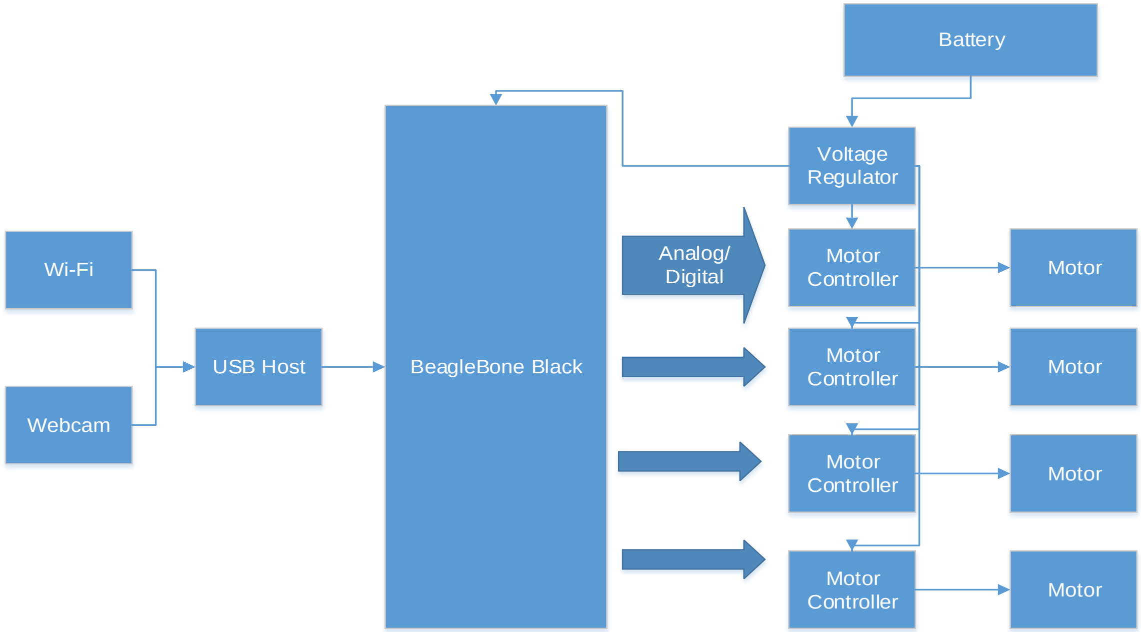


Figure 5.1-1 Hardware Flowchart

The motor controller has two main sections; the MSP430 controller, and the H-Bridge circuit. It will receive two signals from the CPU; an analog voltage to control the duty cycle of the PWM being generated, and a Digital signal to control the DIR lines. This allows the CPU to control the direction and speed of the DC motors used for the wheels. The MSP 430 will require 3.3 V from the voltage regulator, and the H-Bridge will use 3.3 V to control some of the transistors being used, and 10 V which will be the voltage that is passed on to the motors as shown below:

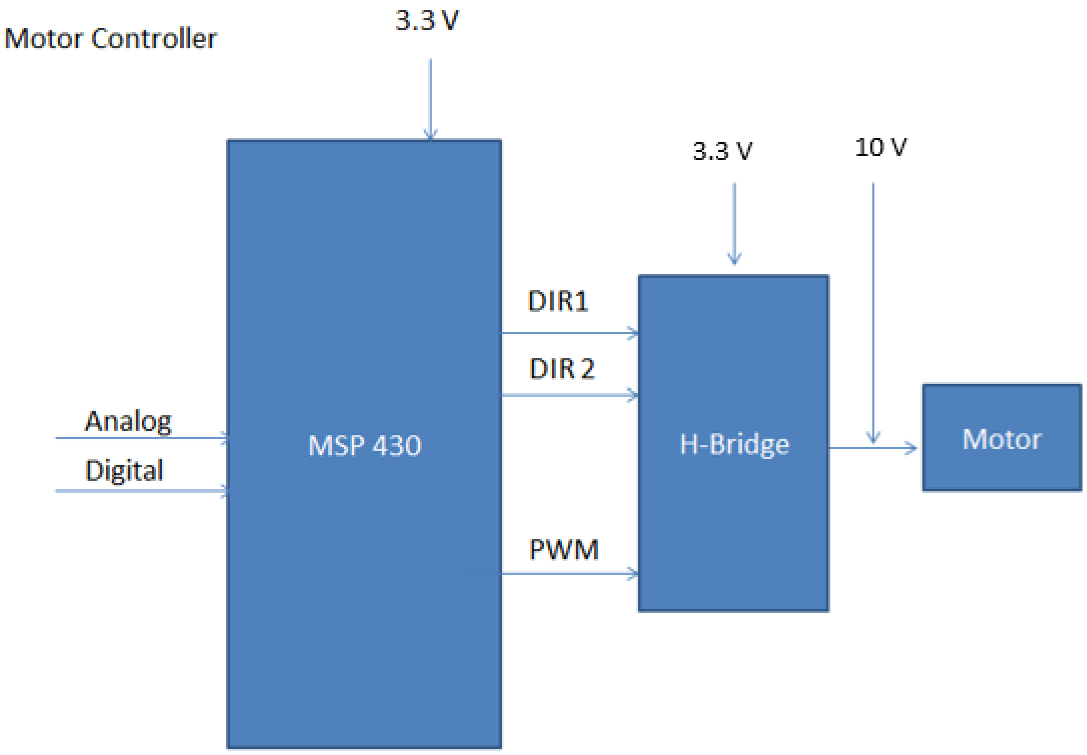


Figure 5.1-2 Motor Controller Flowchart

In order to power the motor controller and all of the devise on the robot we need several different loads powered. As explained previously there are several loads that need to be powered. The ratings are MSP 430 (3.3V) and the H Bridge (10V). Also, You have the BeagleBone Black at 5V 2A that needs to be powered. Below in Figure 5.1-3 is an overal schematic of the power system.

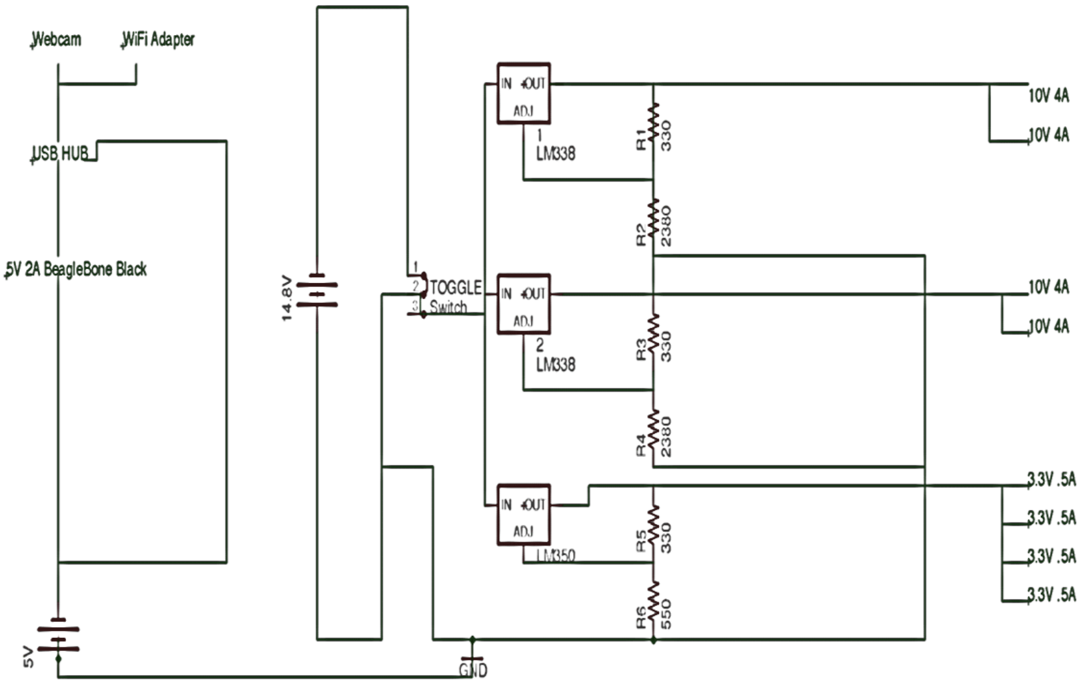


Figure 5.1-3 Power System Schematic

5.2 Software

The software architecture will be divided into subsystems in order to facilitate the implementation with the hardware subsystems. The software architecture on the Beagle Bone Board contains the webserver, TCP/IP server, video streaming, and image processing. There will be a lot of communication between the different modules; as a result, a lot of time will be spent during the integration and testing phase to make sure all are working as expected. In addition to the Beagle Bone Board software architecture, there's also the mobile application which will be able to stream a live video feed as well as optionally control the vehicle. Below is a high-level diagram depicting the software architecture.

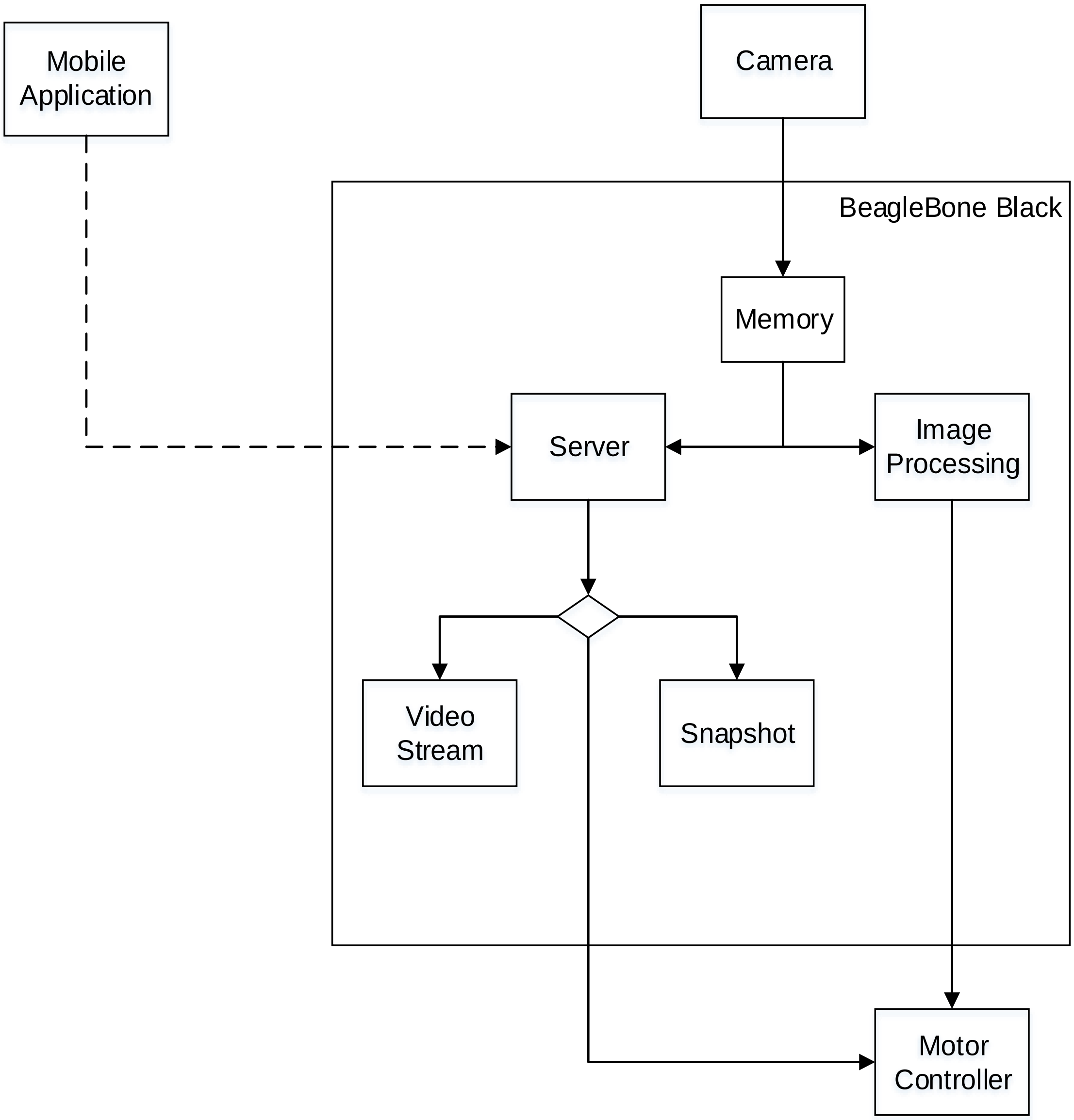


Figure 5.2-1 High-Level Software Architecture Diagram

6.0 Project Prototype Construction

6.1 Parts Acquisition/Financial Budget

Our project is currently being sponsored by DRS Technologies and Boeing. DRS Technologies is providing us with a fully functional commercial grade Thermal Camera while Boeing is providing us with a donated lump sum of money which will cover the cost of the remaining parts. We will try our best to stay under budget. In the event that our budget surpasses the amount being donated, the cost will be split equally among team members. Some of the necessary parts have already been acquired; any remaining parts will be acquired before Senior Design 2 starts from multiple local and online retailers such as Amazon, Element14, Walmart, Hobby Shops, and SparkFun.

Part Name	Quantity	Price (each)	Total
Wi-Fi USB Adapter	1	$5.99	$5.99
MicroSD Card	1	$7.99	$7.99
USB AC Powered Hub	1	$7.99	$7.99
Motor	5	$14.99	$74.95
Mecanum Wheels	1	$59.99	$59.99
Robot Frame kit	1	$79.99	$79.99
Motor controller PCBs	2	$43.95	$87.90
Motor Controller Parts	1	$46.00	$46.00
22 gauge wire set	1	$20.00	$20.00
Wire connectors	1	$40.00	$40.00
Barrel jack connectors	2	$2.95	$5.90
USB Battery	1	$29.95	$29.95
Battery + Charger	1	$88.55	$88.55
BeagleBone Black	1	$62.71	$62.71
Miscellaneous			$50
		Total	$667.91
		Over budget	$42.54

Table 6.1-1 Parts/Financial Budget

6.2 PCB Vendor and Assembly

One of the requirements of the Senior Design class is to have a printed circuit board (PCB) in the project. By having a PCB our project will look professional and of high quality. There are a couple of software available for free that allow us to design the PCB, one of the well-known one is Eagle Cad. Eagle Cad offers a free version which meets our requirements and the file format is widely accepted by multiple manufacturers. As for getting the PCB design printed, there are many options available online. Almost all the PCB manufacturers accept multiple file formats, allowing us to try different options no matter what software for design we choose. This also allow us to compare prices, reviews, and delivery time. Another option is to bypass the manufacturer and create it ourselves helping us save some money. Unfortunately, no one in the team has previous experience working with PCB, as a result we decided to compare a few online manufacturers since it would take too much time to make one ourselves.

PCB-Pool- PCB-Pool is one of the online manufactures that we are considering. They are based in Canada and have good reviews in do-it-yourself and hobby forum communities. Their price ranges between $42 and $105 depending if a 2-layer or 4-layer PCB is needed. Their average delivery time is 8 days plus shipping time.

OSH-Park- OSH-Park is a community PCB order, where the PCB board is printed along with other PCB board from different customers. This is done to save cost for the manufacturer and the consumer. The only drawback is waiting time is usually longer. OSH Park charges a flat $5 shipping fee and all orders include three PCB's. Their price ranges from $5 per square inch for a 2-layer PCB and $10 per square inch for a 4-layer PCB.

Express PCB- Express PCB charges a flat fee for a 2-layer and 4-layer PCB. Similar to OSH-Park, the orders include three PCB's and their price for a 2-layer PCB is $51 flat while a 4-layer PCB will cost $98 flat. In addition to the PCB board fee, there's a shipping fee of $10 which guarantees a 2-day delivery as long as the order is placed during business time.

4PCB- Another option that we are considering is 4PCB. 4PCB is located in USA and they offer a student program for college students. As a result, we as students get a discount for printing our PCB with them. A 2-layer full spec PCB costs $33 each and there is no minimum on how many we can order. A 4-layer full spec PCB costs $66 and like the 2-layer, there is no limit in the amount of orders. In addition, 4PCB offers their own PCB layout software for free as well as a PCB file checker to check for compatibility.

6.3 Prototype Construction and Configuration

6.3.1 Software Build Plan

Two of the group members will be in charge of developing the software that will be executed on the BeagleBone Black Board. The software consists of the video/image processing, web server, video streaming, and wireless communication. One group member will focus entirely on the video/image processing since it's a challenging and time consuming task. The testing and development will be performed on a computer running the Linux Operating System in order to prevent compatibility issues during the integration phase. The OpenCV SDK will be modified so that only the necessary dependencies are compiled for the BeagleBone Board. As a result, we can lower the resources needed such as RAM and CPU. The other group member will be in charge of the web server, video streaming, video streaming, and wireless communication. A Linux platform, in this case a VPS (Virtual Private Server), will also be utilized to prevent any compatibility issues during the integration phase. An advantage of using a VPS is that it already has Static IP addressing and multiple web servers can be tested in a matter of minutes since installation is just a matter of a few clicks.

6.3.2 Hardware Build Plan

The two team members in charge of hardware design will divide the sub system components and perform early integration tests after successful unit testing of the separate components. One team member will be tasked with creating and organizing the power systems, while the other will be building the motor controllers. Then both will work together with integrating all of the hardware components to finish building the complete chassis

6.4 Milestones

To be able to complete the project on time, a milestone was created in order to give us an idea, both as a group and individually, of our progress on the project in order to stay on track. Although the set milestones have to be followed, they might need to be adjusted later on in case we find an area more difficult than anticipated or run into problems. For Senior Design I, research and design were given priority throughout the semester as can be shown below:

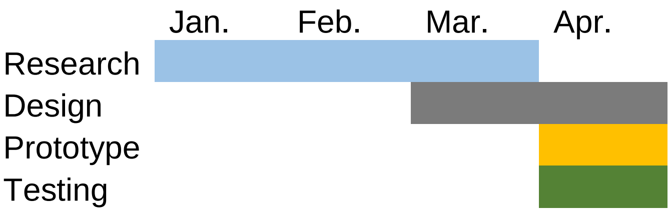


Figure 6.4-1 Senior Design I Milestones

The webserver as well as a prototype of the mobile application were tested in order determine how much resource power was being consumed. For Senior Design II, our goal is for an aggressive schedule to start implementing the hardware as soon as possible. In order to achieve this, some parts have already been ordered prior to the end of Senior Design I. During the break between Senior Design I and Senior Design II, the remaining parts will be ordered. If the parts arrive on time, this would allow us to start implementation before the summer semester begins. Below is our milestone for Senior Design II:

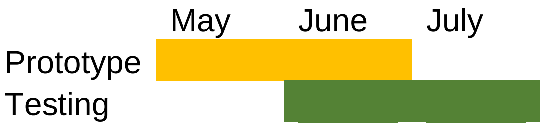


Figure 6.4-2 Senior Design II Milestones

Task Name	Duration	Start	Finish
Research	**46 days**	**2/1/2014**	**3/19/2014**
Hardware	**46 days**	**2/1/2014**	**3/19/2014**
Microcontrollers			3/19/2014
Motor Controllers			3/19/2014
Sensors			3/19/2014
Voltage Regulators			3/19/2014
Software	**46 days**	**2/1/2014**	**3/19/2014**
Mobile Application			3/19/2014
Webserver			3/19/2014
OpenCV SDK			3/19/2014
Design	**35 days**	**3/20/2014**	**4/24/2014**
Hardware	**35 days**	**3/20/2014**	**4/24/2014**
Motor Controllers			4/24/2014
Voltage Regulators			4/24/2014
Software	**35 days**	**3/20/2014**	**4/24/2014**
Webserver			4/24/2014
Mobile Application			4/24/2014
Image Processing			4/24/2014
Prototype	**49 days**	**6/1/2014**	**6/30/214**
Hardware	**49 days**	**6/1/2014**	**6/30/214**
Motor Controllers			6/30/214
Voltage Regulators			6/30/214
Software	**49 days**	**6/1/2014**	**6/30/214**
Webserver			6/30/214
Mobile Application			6/30/214
Image Processing			6/30/214
Testing	**20 days**	**7/1/2014**	**7/20/2014**

Table 6.4-1 Milestone Details

7.0 Project Prototype Test Plan

In this section the testing methods for the project will be discussed. Each test case will contain a description explaining how it will be performed, materials required, purpose, and expected result. There will be two phases for the project’s testing plan. The first phase is Unit Testing, where each individual subsystem will be tested by the corresponding team members. The second phase will be Integration testing, where the subsystems will be combined together one by one and test their functionality to make sure all the subsystems are behaving correctly. From here, the system will be tested to make sure each and every subsystem is behaving correctly without being affected by the combination or modification of a different subsystem.

7.1 Test Environment

The test environment that will be utilized consists of the Senior Design lab and each member's home. The Senior Design lab will mostly be used to test the electronic side of the hardware thanks to the availability of the tools and parts required. Each group member will test their assigned subsystems at their preferred location for Unit Testing phase as long as accurate and correct results can be obtained. For the Integration Phase a meeting will be required at one of the group member's house in order to combine the different subsystems and perform the tests.

7.2 Unit Testing

Team members will be performing unit tests separately and simultaneously to quickly have sub components ready for integration testing.

7.2.1 Web Server

Test Name: Accessible Web Server via Internet

Objective: The objective of this test is to verify that the web server can be accessed via a regular web browser from a computer or mobile device and the HTTP connection is successful.

Supplies:

- Computer
- Mobile Device (Cell Phone/Tablet)
- Modem
- Router
- Single-Board Computer

Preparation: The Nginx or Node.js HTTP server will be installed onto the single-board computer running a Linux environment and set the HHTP port to 80. The board will be connected to the Internet either wirelessly or via an Ethernet cord. The Computer or mobile device will also be connected to the Internet but in a different network. Once the board is connected to the Internet, its IP address needs to be noted.

Procedure: Open a web browser in either the computer or mobile device and point to <Board-IP>:80 address to initiate a HTTP connection to the board.

Expected Result:
Each Web Server will contain a customized welcome page with a welcome message. Depending on the Web Server used, the corresponding welcome page should be loaded. If the welcome page is loaded, then everything is working

correctly. If a 404 error is shown, then an error occurred and the setup was not successful. The port could have been set wrong or the IP address noted is not the correct one.

Test Name: Accessible Web Server via Local Network

Objective: The objective of this test is to verify that the web server can be accessed via a regular web browser from a computer or mobile device and the HTTP connection is successful in a local network.

Supplies:

- Computer
- Mobile Device (Cell Phone/Tablet)
- Modem
- Router
- Single-Board Computer

Preparation: The Nginx or Node.js HTTP server will be installed onto the single-board computer running a Linux environment and set the HHTP port to 80. The board will be connected to the Internet either wirelessly or via an Ethernet cord. The Computer or mobile device will also be connected to the Internet in the same server. Once the board is connected to the Internet, its IP address needs to be noted.

Procedure: Open a web browser in either the computer or mobile device and point to <Board-IP>:80 address to initiate a HTTP connection to the board.

Expected Result:
Each Web Server contains a customized welcome page with a welcome message. Depending on the Web Server used, the corresponding welcome page should be loaded. If the welcome page is loaded, then everything is working correctly. If a 404 error is shown, then an error occurred and the setup was not successful. The port could have been set wrong or the IP address noted is not the correct one.

Test Name: Receive Commands from mobile device

Objective: The objective of this test is to verify that the server is able to listen and accept a connection, and be able to receive commands from the mobile device.

Supplies:

- Single-Board with web server installed and running
- Mobile Device with application installed
- Modem
- Router
- HDML TV/Monitor

Preparation: Connect the single-board and mobile device to the Internet. Open the mobile application in the mobile device.

Procedure: Once the mobile application is active, a series of commands will be sent to the web server containing values for direction.

Expected Result:
The values displayed on the mobile device are to be received by the web server and printed to the screen as a confirmation that the commands were received.

7.2.2 Cameras

Test Name: Video stream plugins

Objective: The objective of this test is to verify that the input and output of the video stream are functioning correctly.

Supplies:

- Logitech C300 Webcam
- Tamarisk 320 thermal camera
- BeagleBone Board
- HDMI TV/Monitor
- Micro USB charger
- Mouse/Keyboard

Preparation: Plug in the Logitech C300 webcam into the BeagleBone Board along with a keyboard to the USB ports. Apply power to the BeagleBone Board with the Micro USB charger.

Procedure:

1. Start up the BeagleBone Board
2. Use a webcam viewer to see the output of the camera
3. Setup the input plugin to capture frames from the camera to the BeagleBone Board. Modify the output plugin so the frames are stored in the /root/frames/ directory.
4. Turn off the camera and check that there are frames in the /root/frames/ directory.
5. Turn off the BeagleBone Board.
6. Disconnect the webcam and replace it with the thermal camera.
7. Repeat steps 1-4.

Expected Result:
The frames outputted from the camera and then saved to the BeagleBone Board memory are to be found in the /root/frames/ directory as it was setup during the procedure. As a result, this proofs that the input/output plugins are working correctly.

Test Name: Video Image Processing

Objective: The objective for this test is to verify that OpenCV can get frames from the webcam and thermal cameras, detect a target, and track the movement.

Supplies:

- Logitech C300 Webcam
- Tamarisk 320 Thermal Camera
- Beagle Bone Board
- Micro USB charger
- HDMI TV/Monitor
- OpenCV SDK

Preparation: Connect the Logitech C300 Webcam to the Beagle Bone Board. Apply power to the Beagle Bone Board.

Procedure:

1. Boot up the BeagleBone Board.
2. Have one team member walk within the cameras view angle, 2-3 feet away.
3. Run the target detection algorithm, this algorithm will detect any target that are in the camera's view and produces an array with the target's values.
4. Run the target tracking algorithm, this algorithm will utilize the output from the target detection algorithm to track the target detected.
5. Ask the team member to continue on moving and test that the target tracking algorithm follows the member.
6. Repeat steps 1-5 but this time with the Tamarisk 320 thermal camera.

Expected Result:

The outcome of this test is to test the accuracy of the OpenCV algorithms while using frames as input from the Logitech C300 webcam and Tamarisk 320 thermal camera.

7.2.3 Mobile Application

Test Name: View live video feed

Objective: The objective for this test is to verify that the user can view a live video feed within the mobile application.

Supplies:

- Mobile Device (Phone/Tablet) with application installed
- Beagle Bone Board
- Micro USB charger
- Modem
- Router
- Logitech C300 Webcam

Preparation: Plug in the Logitech C300 webcam to the Beagle Bone Board along with a keyboard. Power up the Beagle Bone Board with the Micro USB charger.

Procedure:

1. Boot up the Beagle Bone Board.
2. Connect Mobile Device and Beagle Bone Board to the Internet or same Local Network.
3. Open mobile application.
4. Login with existing account.
5. Wait for video stream to load and view stream.

Expected Result:

The outcome of this test is to verify that the mobile application can open and display the video stream from the Beagle Bone Board.

Test Name: Installation of Mobile Application

Objective: The objective of this test is to make sure the application is successfully installed in a mobile device (Phone/Tablet) that it was developed for.

Supplies:

- Android Mobile Device (Phone/Tablet)
- Eclipse with Android SDK
- Computer
- Micro-USB cable

Preparation: The mobile application needs to be compiled by using Eclipse with the Android SDK installed. Once the mobile application is compiled, it will be transferred to the mobile device of choice to start the installation process.

Procedure: Using the mobile device, navigate to the folder where the compiled mobile application was stored and open it.

Expected Result:

Once the application is opened, the installation process will start automatically. If the installation was successful, the mobile device is compatible. If the installation fails, the mobile device can be running an outdated version of Android and the application is not compatible.

Test Name: Offline Mobile Device (Phone/Tablet)

Objective: The objective of this test is to verify that the mobile application can detect if a mobile device (Phone/Tablet) is offline.

Supplies:

- Mobile Device (Phone/Tablet) with application installed
- No Internet Connection

Preparation: The mobile device is to be disconnected from the Internet. In order to be 100% sure, Airplane mode will be enabled. By enabling Airplane mode, any and all communication signals are disabled in the mobile device.

Procedure: Open the mobile application in the mobile device.

Expected Result:
As soon as the application starts, it will test if the mobile device is connected to the Internet. Since Airplane mode has been enabled, the application will alert the user that the mobile device is not connected to the Internet. If the test fails and the user is not alerted, the function in the mobile application needs to be checked and is not working correctly.

Test Name: Online Mobile Device (Phone/Tablet)

Objective: The objective of this test is to verify that the mobile application can detect if a mobile device (Phone/Tablet) is online.

Supplies:

- Mobile Device (Phone/Tablet) with application installed
- Modem
- Router

Preparation: The mobile device is to be connected to the Internet. The connection can be to a wireless access point or enabling mobile data on the mobile device.

Procedure: Open the mobile application in the mobile device.

Expected Result:
As soon as the application starts, it will test if the mobile device is connected to the Internet. If the mobile device is found to be connected to the Internet, no alert is required. If the user is alerted that the mobile device is offline, then this is a false positive and the function in the mobile application needs to be checked.

Test Name: Account Creation

Objective: The Objective of this test is to verify that the user is able to register an account using the mobile application.

Supplies:

- Mobile Device (Phone/Tablet)
- Modem
- Router

Preparation: Install the mobile application to the mobile device. Once installation is completed, open the application.

Procedure: Once the application is opened, the user will be presented with a login screen. Since no account exists yet, select the Register button. Once the Register button is selected, the user will be presented with various fields that need to be filled including username and password.

Expected Result:
The user will be able to create an account using the mobile application. Once all the information has been filled and the register button is selected, a confirmation will be shown to the user.

Test Name: Login with existing account

Objective: The objective of this test is to verify that the account registration is working correctly. This will test if the account exists.

Supplies:

- Mobile Device (Phone/Tablet) with application installed
- Modem
- Router

Preparation: Open the application in the mobile device to be presented with the login screen.

Procedure: Once in the login screen, input the username and password combination from the existing account. Once the username and password field have been filled, select the login button.

Expected Result:
Once logged in, the user will be presented with the Control activity where the user can send command, and view the video feed.

Test Name: Login with wrong credentials

Objective: The objective of this test is to verify that no unauthorized user can get access to the Control section of the mobile application.

Supplies:

- Mobile Device (Phone/Tablet) with application installed
- Modem
- Router

Preparation: Open the application in the mobile device to be presented with the login screen.

Procedure: Once in the login screen, input a wrong username and password combination. Once the username and password field have been filled, select the login button.

Expected Result:
The application should alert the user if the provided login credentials are incorrect, or that they don't exist. The user will have the option to try again or quit the application.

Test Name: Saving Video Snapshot to Mobile Device

Objective: The objective of this test is to verify the functionality of storing a snapshot, of the video stream, to the internal storage of the mobile device.

Supplies:

- Mobile Device (Phone/Tablet)
- Modem
- Router
- Single-board Computer
- HTTP Web Server

Preparation: Setup the web server on the Single-board computer and make sure that is connected to the Internet. Install the application in the mobile device and create a new account.

Procedure: Open the application and login with the correct credentials in order to see the Control section.

Expected Result:
Once access has been given to access the Control section, the user will be able to see a live video stream. Select the "Save Snapshot" button in order to save a specific frame to the internal storage of the mobile device. Open the mobile's device gallery app to confirm the snapshot was saved.

7.2.4 Microcontroller

Test Name: Arch Linux|ARM on eMMC

Objective: To determine the full functionality of an operating system on an embedded media

Components:

- BeagleBone Black
- eMMC
- Micro SD Card
- Arch Linux
- A computer
- Ethernet Cable

Preparation: The standard BeagleBone comes equipped with a Linux flavor Ångström installed on the eMMC on board. Before this test could be performed, the standard operating system has to be wiped out and replaced with an Arch Linux|ARM. There is a separate procedure for to install the new OS.

Procedure: This test is best using the SSH protocol to connect to the BeagleBone board. All is needed is another computer with any standard operating system. In this case, we will use Linux for the procedure

1. Connect the BeagleBone Black to the computer with a Ethernet Cable
2. Target the IP address 192.168.7.2 using the SSH protocol to gain control over the BeagleBone Black (SSH root@192.168.7.2)
3. Once accessed the board input *$ lsb_release –irc*, then the version information should be display onto the screen

Expected Result:
We are expecting to see the display of the Linux distributor information plus release version and the codename along with the display. The line that we interested in is *Distributor ID: Arch Linux.* Once we confirm the version, it is safe to assume that Arch LINUX|ARM is working properly.

Test Name: MicroSD Functionality

Objective: To determine the accessibility over Micro SD reader through an implemented operating system, Arch Linux.

Components:

- BeagleBone Black
- Computer
- Ethernet Cable
- Micro SD card w/ some data on it (not blank)

Procedure: Many Linux versions offer the automatic mount functionality but our version of Linux may or may not have that feature. To ensure that we do have access, we must to it manually.

1. Insert the MicroSD card
2. In the command line, type*: # fdisk –l* and the system should display all of the storage formats on the device. We are interested in something similar to this: */dev/sdb1*
3. Type: *$ pmount /dev/sdb1 sdcard* to mount the device.

Expected Result:
The system should take care of everything at that point, so what we can expect to see at this point is being able to access the directory of that device my just using *ls* command and search for the files existed in the device. We can move on with our development process after knowing that our SD card is accessible.

7.2.4 Power Testing

Test: Battery Testing: Charging

Objective: This test is to check the amount of time it takes for the battery to reach back to the nominal voltage of 7.2V.

Components:

- Lithium ion Batteries
- Multi-meter
- Battery Charger

Preparation: Before we start the test of the battery we want to do a full expectation of the battery. We want to make sure that the container of the battery is in operational conditions. There are to be no cracks or corrosion on the battery. The connection terminal should be as clean as possible with no traces of dirt. You have to make sure that there is no short in terminals because this could cause potential overheating which can lead to a fire. Also you want to make sure the testing conditions for the battery are in moderate temperature.

Procedure: Before starting this test make sure you follow all of the preparation. The testing of the battery shall be done in this chronological order:

1. Check for voltage of the battery with Multi-meter
2. Check current of the battery with Multi-meter
3. Connect battery to charger
4. Monitor the time it takes for battery to reach initial state.

Expected Results: The results that should be found is that it should take less than 8 hours for the battery to recharge. We also want to confirm that there’s no Memory effect.

Test: Battery Testing: Discharge rate to microcontroller

Objective: This test is to make sure that the power supply is discharging the requirements need to for the microcontroller to run properly

Components:

- Lithium ion Batteries
- Multi-meter
- Voltage Regulator with circuit
- Beagle Bone Black

Preparation: Before we start the test of the battery we want to do a full expectation of the battery. We want to make sure that the container of the battery is in operational conditions. There are to be no cracks or corrosion on the battery. The connection terminal should be as clean as possible with no traces of dirt. You have to make sure that there is no short in terminals because this could cause potential overheating which can lead to a fire. Also you want to make sure the testing conditions for the battery are in moderate temperature.

Procedure:

1. Set up the voltage regulator with circuit for the beagle bone black
2. Have the batteries set up as power source for the circuit
3. Check for voltage of the battery with Multi-meter
4. Check current of the battery with Multi-meter

Expected Results:

That the output to the microcontroller circuit should be about 5v and the current should be about 2A of current.

Test: Battery Testing: Discharge rate to motors

Objective: This test is to make sure that the power supply is discharging the requirements need to for the motors to run properly

Components:

- Lithium ion Batteries
- Multi-meter
- Voltage Regulator with circuit
- Motors

Preparation: Before we start the test of the battery we want to do a full expectation of the battery. We want to make sure that the container of the battery is in operational conditions. There are to be no cracks or corrosion on the battery. The connection terminal should be as clean as possible with no traces of dirt. You have to make sure that there is no short in terminals because this could cause potential overheating which can lead to a fire. Also you want to make sure the testing conditions for the battery are in moderate temperature.

Procedure:

1. Set up the voltage regulator with circuit for the beagle bone black
2. Have the batteries set up as power source for the circuit
3. Check for voltage of the battery with Multi-meter
4. Check current of the battery with Multi-meter

Expected Results: The output to the microcontroller circuit should be about 7.2 v and the current should be about 16A of current between the four motors.

7.2.5 Motor Controller

Test name: H-Bridge functionality testing

Objective: These tests are to ensure that the H-Bridge we have designed will properly control the direction of the flow of the current through the output terminals, and that the transistors we have chosen can handle the amount of current that the motors will pull. We will also check the switching times for the circuit.

Supplies:

- Breadboard
- DC Power supply
- Function Generator
- Oscilloscope
- Multi-meter
- IRF9530 x 2
- IRF530 x 2
- BC547 x 8
- 10kΩ Resistor x 10
- Varying sized load resistors

Preparation

Assemble the circuit on the breadboard according to fig 4.03.1-10.

- Make it a little spread out across the board so parts can be easily identified and probed during analysis and troubleshooting

Procedure:

1. Verify all components are connected correctly on the breadboard.
2. Short the enable line to VDD so the circuit is always enabled
3. Set VCC to 7V so we can check the functionality before the capacity
4. Probe each transistor to ensure expected voltages and current are appearing
5. If no problems occur turn up VCC to 10 V and repeat step 4
6. Repeat step 5 with VCC= 14.4V

7. Ensure that the Mos-Fets are not overheating, heat sinks may need to be attached.
8. Using the function generator create a step function for the Enable line.
9. Set the frequency for 100 hz
10. Vary the Duty cycle from 0 to 100 % in steps of 10 % repeating step 4 at each interval.

Expected Results:
We expect that the output will have a drop that is about 1.4 volts less than VCC due to voltage drops across the BJT's in the enable section. The transistors are rated for higher currents than we will be pulling through them, so we shouldn't have any problems with overheating. After applying the step function to the enable line we should see the voltage and current at the load terminals increase from zero to the previous value that we found when the enable line was shorted.

Test Name: Motor as load for H-Bridge

Objective: To verify that the motors rotational speeds and direction can be controlled and altered using the H-Bridge circuit. And to find the motors optimal frequency that we should set our PWM to.

Supplies:

- H-Bridge circuit
- Motor
- DC power supply
- Function generator
- Oscilloscope
- Multi-Meter

Preparation: Assemble the circuit on the breadboard according to fig 4.03.1-10. Make it a little spread out across the board so parts can be easily identified and probed during analysis and troubleshooting

Procedure:

1. Verify all components are connected correctly on the breadboard.
2. Short the enable line to VDD so the circuit is always enabled
3. Set VCC to 7V so we can check the functionality before the capacity
4. Probe each transistor to ensure expected voltages and current are appearing
5. Measure the current pulled when the motor is stalling
6. If no problems occur turn up VCC to 10 V and repeat step 4
7. Repeat step 5 with VCC= 14.4V
8. Ensure that the Mos-Fets are not overheating, heat sinks may need to be attached.
9. Using the function generator create a step function for the Enable line.
10. Set the frequency for 100 hz

11. Vary the Duty cycle from 0 to 100 % in steps of 10 % repeating steps 4 and 5 at each interval.
12. Set the duty cycle for 100% and alter the frequency until the motor pulls the rated stall current with maximum torque.

Expected Results:
The motor's stall current should be around 4.7 amps. And we are expecting the pwm frequency required for maximum torque to be in the range of 100-700 Hz.

Test name: MSP 430- PWM and H-Bridge control signals.

Objective: To check and tune the frequency of the PWM being generated for the H-Bridge's enable lines. We will also ensure that the DIR control lines are being activated properly giving enough time for the FET's to shut off before opening the opposing set. To perform these tests we will also need to ensure that the controller's inputs are working properly.

Supplies:

- Bread board
- MSP 430
- DC power supply
- Oscilloscope
- Multi-Meter

Preparation: Write the code for the MSP 430.

- Generating PWM signal
- Control of duty cycle from analog signal input
- DIR out with delay from Digital input

Procedure:

1. Verify all components are connected correctly on the breadboard and power is being applied to the controller.
2. Slowly raise the voltage from 0 to 3 V on the PWM control line while measuring the PWM on the oscilloscope. This should change the duty cycle of the PWM from 0 to 100 %
3. Using the oscilloscope measure the frequency of the PWM and make adjustments to the code as needed to reach optimal timing.
4. Measuring both DIR out lines from the controller change the input to the digital input that controls these, the two lines should never be active simultaneously.

Expected Results:
We expect that the range for the analog input will be 0 to 3 V to give us a range of 0 to 100% duty cycles on the PWM.

Test Name: Completed motor controller testing

Objective: To ensure that all of the controller's components are still operating properly after assembling the components onto the custom PCB.

Supplies:

- DC Power supply
- Oscilloscope
- Multi-meter
- IRF9530 x 8
- IRF530 x 8
- BC547 x 64
- 10kΩ Resistor x 40
- Custom PCB x 4
- Motor

Preparation: Designing and fabricating the PCB, and soldering all of the components into place on them.

Procedure:

1. Verify all components are connected correctly on the PCB and correct voltages are being applied
2. Slowly raise the voltage from 0 to 3 V on the PWM control line while measuring the PWM on the oscilloscope. This should change the duty cycle of the PWM from 0 to 100 %
3. Measuring both DIR out lines from the controller change the input to the digital input that controls these, the two lines should never be active simultaneously.
4. Attach motor and check the Stall current at 100% duty cycle.

Expected Results:

We expect that the range for the analog input will be 0 to 3 V to give us a range of 0 to 100% duty cycles on the PWM. And that the delay in switching the DIR lines will be enough to prevent the battery from shorting through two fully open Mos-Fets. We expect the Stall current to be around 4 A.

7.2.6 Sensors

Test Name: Obstacle Detection

Objective: The objective of this test is to verify that each sensor is able to detect any objects blocking the way.

Supplies:

- Four HC-SR04 Sensors
- Multimeter

- Obstacle (Flat Object)
- Ruler
- Breadboard
- DC Power Supply

Preparation: Connect one sensor at a time and power the supplies; check specifications. Make sure the voltage is the range of 3V – 5V.

Procedure:

1. Verify that the HC-SR04 sensor is connected correctly to the breadboard.
2. Stand 5 feet away from the sensor with obstacle on hand.
3. Measure the output voltage with the multimeter while moving towards the sensor.
4. Repeat Step 1-3 with the remaining sensors.

Expected Result:
The voltage output should change dramatically as the obstacle in hand gets close to the sensor. This will allow us to get accurate reading to detect any obstacle.

7.3 Integration Testing

Test Name: Cameras/OpenCV/Beagle Bone Board/Motor Controller Integration Test

Objective: The objective of this test is to verify that the integration between the mentioned subsystems are behaving correctly.

Supplies:

- Logitech C300 Webcam
- Tamarisk 320 Thermal Camera
- OpenCV SDK
- Beagle Bone Board

Preparation: Power the Beagle Bone Board using a 14.8V source. Connect the Tamarisk 320 Thermal Camera and the Logitech C300 webcam to the Beagle Bone Board.

Procedure:

1. Boot up the Beagle Bone Board
2. Receive input from the Logitech C300 Webcam
3. Have one of the team members stand inside the camera’s range.
4. Verify that the team member is detected as a target.
5. Verify that the team member is being tracked while moving.
6. Verify that vehicle moves with the target.
7. Repeat steps 1-6 using the Tamarisk 320 Thermal Camera.

Expected Result:
The outcome of this is to verify that all the mentioned subsystem are communicating with each other after they have been integrated.

Test Name: Obstacle Course and Object Avoidance

Objective: The objective of this test is to test the object avoidance. The vehicle will travel from the beginning of an obstacle course until the end while avoiding the obstacles and finding a different path.

Supplies:

- Logitech C300 Webcam
- Tamarisk 320 Thermal Camera
- OpenCV SDK
- Beagle Bone Board
- Four HC-SR04 Sensors
- Motor Controller
- Chassis

Preparation: Power the Beagle Bone Board using a 14.8V source. Connect the Tamarisk 320 Thermal Camera and the Logitech C300 webcam to the Beagle Bone Board.

Procedure: Setup an obstacle course challenging enough to make the vehicle search for a new path at least three times. Place the vehicle at the beginning of the obstacle course.

Expected Result:
The outcome of this test is to verify the accuracy of the object avoidance algorithms as well as the accuracy of the HC-SR04 Sensors. Once an obstacle has been detected, the vehicle should automatically find a new path.

Test Name: Mobile Application/Beagle Bone Board/Video Stream

Objective: The objective of this test is to verify that the mobile application is able to communicate with the vehicle by showing the live video feed as well as controlling the vehicle.

Supplies:

- Logitech C300 Webcam
- Tamarisk 320 Thermal Camera
- OpenCV SDK
- Beagle Bone Board
- Motor Controller
- Chassis
- Mobile Device (Phone/Tablet) with mobile application installed.

Preparation:
Power the Beagle Bone Board using a 14.8V source. Connect the Tamarisk 320 Thermal Camera and the Logitech C300 webcam to the Beagle Bone Board.

Procedure:

1. Boot up the Beagle Bone Board.
2. Connect the mobile device and Beagle Bone Board to the Internet.
3. Open the application on the mobile device and login.
4. Verify that the vehicle is communicating with the mobile application by viewing the live video feed.
5. Control the vehicle using the virtual joystick on the mobile application.

Expected Result:
The outcome of this test is to verify the wireless communication between the vehicle and the mobile device. During this test, the video feed functionality along with the control functionality of the mobile application will be tested.

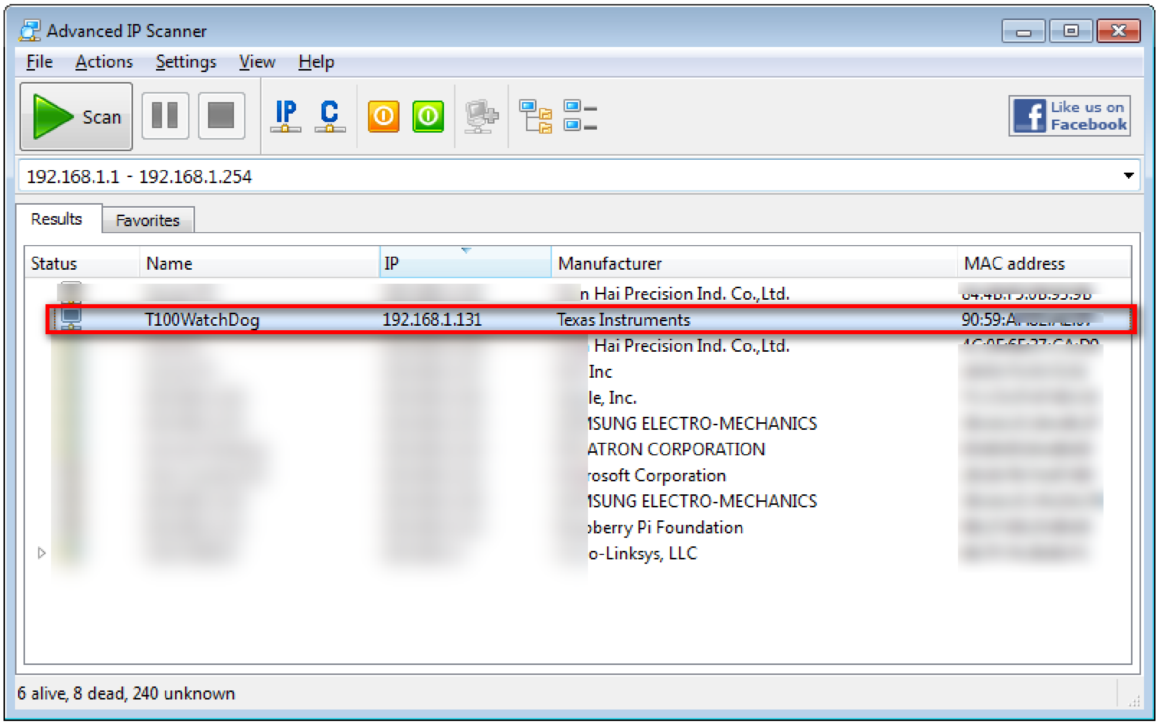
8.0 System Operations

Although the T-100 Watchdog can transition from autonomous to remote control mode automatically without interaction from the user, an initial setup is required. This includes setting up the Wi-Fi connection, Dynamic IP client, and optionally changing settings that will change the behavior of the vehicle. Below are some of the tools required in order to perform the initial setup.

- Computer
- SSH Client (Ex: Putty)
- Ethernet Cord
- Modem/Router
  - Wi-Fi with WPA Security protocol
- Familiar with Linux commands
- Advance Ip Scanner (http://www.advanced-ip-scanner.com/)
- No-IP Account

8.1 Wi-Fi Setup

In order for the BeagleBone Black to connect to your Wi-Fi connection, the name of the network and security key has to be set in the BeagleBone Black. Connect the BeagleBone Black to the Internet via a wired connection using the Ethernet cable. Boot up the BeagleBone Black and a wait for a minute while the board boots up and connects to the Internet. Download Advance Ip Scanner (http://www.advanced-ip-scanner.com/) and install it in your computer. Once installed, run the program and click the Scan button. The program will now scan all the devices connected to your network and show their local ip. The results will look like in Figure 8.1-1 below.



192.168.1.1 - 192.168.1.254

Status	Name	IP	Manufacturer	MAC address
	[redacted]	[redacted]	n Hai Precision Ind. Co.,Ltd.	[redacted]
	T100WatchDog	192.168.1.131	Texas Instruments	90:59:A[redacted]
	[redacted]	[redacted]	Hai Precision Ind. Co.,Ltd.	[redacted]
	[redacted]	[redacted]	Inc	[redacted]
	[redacted]	[redacted]	le, Inc.	[redacted]
	[redacted]	[redacted]	ISUNG ELECTRO-MECHANICS	[redacted]
	[redacted]	[redacted]	ATRON CORPORATION	[redacted]
	[redacted]	[redacted]	rosoft Corporation	[redacted]
	[redacted]	[redacted]	ISUNG ELECTRO-MECHANICS	[redacted]
	[redacted]	[redacted]	berry Pi Foundation	[redacted]
	[redacted]	[redacted]	o-Linksys, LLC	[redacted]

6 alive, 8 dead, 240 unknown

Figure 8.1-1 Advance Ip Scanner Results

The entry with the name of “T100WatchDog” is the BeagleBone Black. Open your SSH client of choice (Ex: Putty) and use the IP address found by Advance Ip Scanner to connect to the BeagleBone Black. When asked for a username, type “root”, there is no password. You can set a password to add security to the BeagleBone Black later on.

Once you have access to the BeagleBone Black, we need to add the details of the Wi-Fi connection to the interfaces file. Run the following command in order to open the interface file.

```
nano /etc/network/interfaces
```

Once the file opens, scroll to the end where you will find a line that says “wpa-ssid” and “wpa-psk”, edit this fields with the name of your Wi-Fi network and password as shown below:

```
wpa-ssid networkname
wpa-psk networkpassword
```

The last step is to turn on the Wi-Fi dongle and allow it to scan for your connection. The commands to achieve this are below:

```
ifconfig wlan0 up
iwlist wlan0 scanning
```

Now that the Wi-Fi adapter has been activated and the Wi-Fi connection has been established, the next step is to shutdown the board and disconnect the Ethernet cord to test the Wi-Fi connection. To shutdown the board, use the following command:

```
shutdown –tH now
```

In order to test the Wi-Fi connection, boot up the BeagleBone Black without the Ethernet cord and after about a minute, connect to it using your SSH client with the same IP, the connection should be successful. If the connection is not successful, the credentials of the connection need to be checked to make sure they are correct.

8.2 Dynamic IP Client Setup

As discussed in Section 3.12, we are utilizing the No-IP client in order to keep track of IP changes and not have to change the Android application each time the IP changes. The client has already been installed and set up. The update interval has been set to every five hours, in case a faster update interval is needed please run the following command to change the update interval.

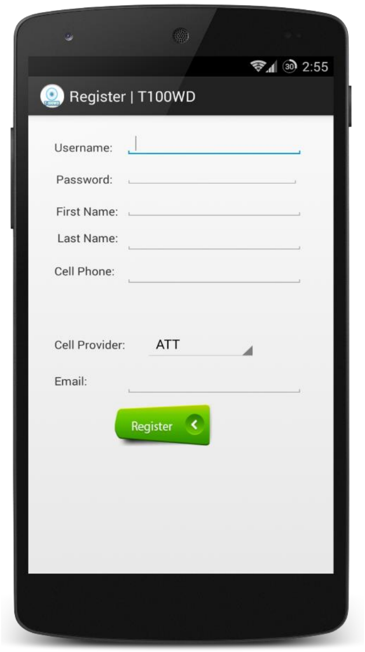
```
noip2 –U x
```

The “x” in the command above is the number of minutes that you would like to use, for example if you would like the update interval to be every 20 minutes the command is as follows:

```
noip2 –U 20
```

8.3 Contact Information

In order to receive an alert once motion has been detected and the vehicle is done transitioning to remote control mode, you need to register for an account using the mobile application. Once the application is installed in your mobile device, select the Register button and the Register Activity will come up as shown below in Figure 8.3-1.



Register | T100WD

Username:

Password:

First Name:

Last Name:

Cell Phone:

Cell Provider: ATT

Email:

Figure 8.3-1 Registration Activity

Fill out the fields with the correct information. Once completed, tap the Register button. The email and cell phone are sent to an external server which the BeagleBone Black will connect to and parse the required information in order to alert the user.

8.4 Start the T-100 Watchdog

Once Sections 8.1 to 8.3 have been completed, we can go ahead and start the T-100 Watchdog by turning the switch on the back on. Connect to the T-100 Watchdog as explained in Section 8.1 and run the following commands to start the computer vision subsystem:

```
cd /home/debian/Scripts/JavaDetector/fswebcam
./run
```

Once the commands have been sent to the T-100 Watchdog, the vehicle will detect any motion. If motion is detected, an email and text message will be sent alerting the user that motion has been detected. In the meantime, the T-100 Watchdog will rotate in the direction from where the motion was detected. During this time, the computer vision subsystem will continue running for 20-30 seconds. Afterwards, the T100-Watchdog will transition to remote control mode and alert the user once the transition has been completed. Once completed, the user has the ability to control the vehicle remotely and view the live video stream.

9.0 Conclusion

The senior design project is a final measurement of our knowledge that we've gained during our time in the college of engineering. The students are expected to demonstrate everything that we've learned both inside and outside of our classroom. Some of us may have side projects that we've worked on in our own time. Engineering students are curious people by nature. Many times in the last several years of being here, we've all had the urge to apply our knowledge to the real world problems. Sometimes we do wait to finish certain courses first before accepting a challenge due to lack of initial knowledge of the subject. Now that the process of becoming an engineer is practically approaching the end of the road, we are now more equip with enough knowledge to tackle real challenges. We are now ready to demonstrate our capable hands to the world. This project offers exactly that to us with a very professional manner.

T-100 Watch Dog is a robotic project that involves electronic circuits and high-level logic control design. It has provided us with lots of information on related subjects such as electrical power, mechanical control, board logic systems, etc. There is definitely an unexpected amount of knowledge and experience that we've gained through the research and design process of the project. More than most people would value is the experience that we've collected beyond the classroom setting. Although, we've built vast strength of basic fundamental structure of knowledge in series of lectures, there are just as much and more outside of that box. The hardware aspect of the project opened our eyes to the process of researching components based on manufacturers and even criteria based on the industry standard including factors such as government regulations. There are many technologies out there that are beyond what we've learned in school. However, our basic understanding of electrical and computer engineering helped us better guide ourselves to compare and contrast many of the same products from brand to brand. The same can also be said about the software. Many of the products that we've found, open and close source, are something that we've never seen before. However, we are able to understand concepts and methodology that many of the documents discussed. After seeing programs after programs, we start to see industry standards of certain components and what the end users would expect from the developers. It has become a process of transforming ourselves to into a real engineer. We now know better to write a certain program that maintains a certain level of run-time or even building a control board that can perform the same or more amount of tasks than an average product out there in the market.

Appendix A: Sources/Permission

Sources:

Thermal Cameras:

- http://www.flir.com/cvs/cores/view/?id=54717
- http://www.mikrosens.com.tr/?q=solutions
- http://www.infraredcamerasinc.com/ICI-320x-Thermal-IR-Camera-Core.html
- http://www.infraredcamerasinc.com/infrared-camera-7640.html
- http://www.thermoteknix.com/products/oem-thermal-imaging/microcam-low-power-thermal-imaging-cores/
- http://www.sofradir-ec.com/
- http://www.drs.com/products/rsta/tamarisk.aspx

Webcam Cameras:

- http://www.logitech.com/en-us/product/hd-webcam-c270
- http://www.logitech.com/en-us/support/5863?osid=14&bit=64
- http://www.v7-world.com/v7_us/products/connect/webcam/vantage-webcam-300.html
- http://www.microsoft.com/hardware/en-us/p/lifecam-hd-3000
- http://www.aliexpress.com/store/product/2-8-TFT-LCD-Module-AVR-Atmega32-Resource-OV7670-CMOS-Camera/900905_307308862.html

Web Server:

- http://w3techs.com/technologies/overview/web_server/all
- http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/nightvision2.htm/printable
- http://whatismyipaddress.com/dynamic-static
- http://httpd.apache.org/
- http://open.litespeedtech.com/mediawiki/
- http://wiki.nginx.org/Main
- http://nodejs.org/
- https://code.google.com/p/mjpg-streamer/
- http://sourceforge.net/projects/mjpg-streamer/

Android/Mobile:

- http://money.cnn.com/2014/02/28/technology/mobile/mobile-apps-internet/
- http://www.pocket-lint.com/news/122459-1-5m-android-devices-activated-daily-1-billion-total-devices-on-horizon

Motor controller Circuitry:

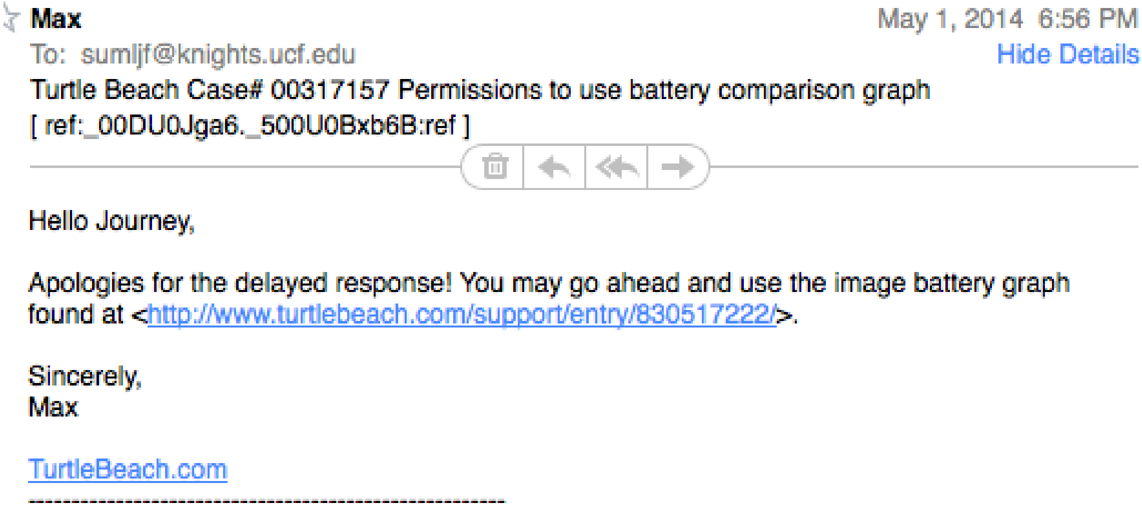
- **http://www.ti.com/lsds/ti/analog/webench/overview.page?DCMP=hpa_design_center&HQS=analogdesigncenter**

Routing Algorithm:

- Introduction to Algorithms 3rd Edition, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 2009
    - Figure 22.1 page 590
    - Figure 23.1 page 625
    - Figure 23.4 page 632
    - Figure 24.6 page 659

Permissions:

Alkaline vs NiMH Comparisons Figure
http://www.turtlebeach.com/support/entry/830517222/



Max May 1, 2014 6:56 PM
To: sumljf@knights.ucf.edu

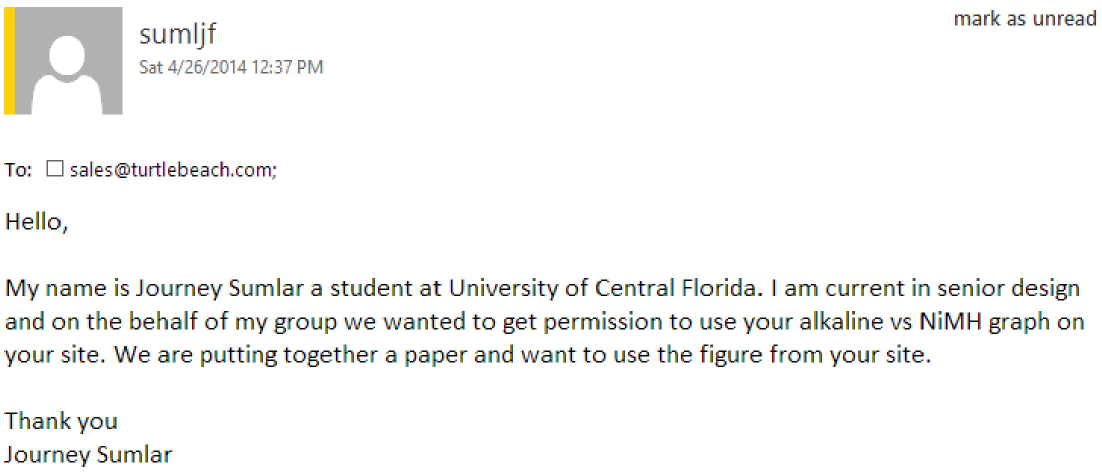
Turtle Beach Case# 00317157 Permissions to use battery comparison graph
[ref:_00DU0Jga6._500U0Bxb6B:ref]

Hello Journey,

Apologies for the delayed response! You may go ahead and use the image battery graph found at <http://www.turtlebeach.com/support/entry/830517222/>.

Sincerely,
Max

TurtleBeach.com



sumljf
Sat 4/26/2014 12:37 PM

To: □ sales@turtlebeach.com;

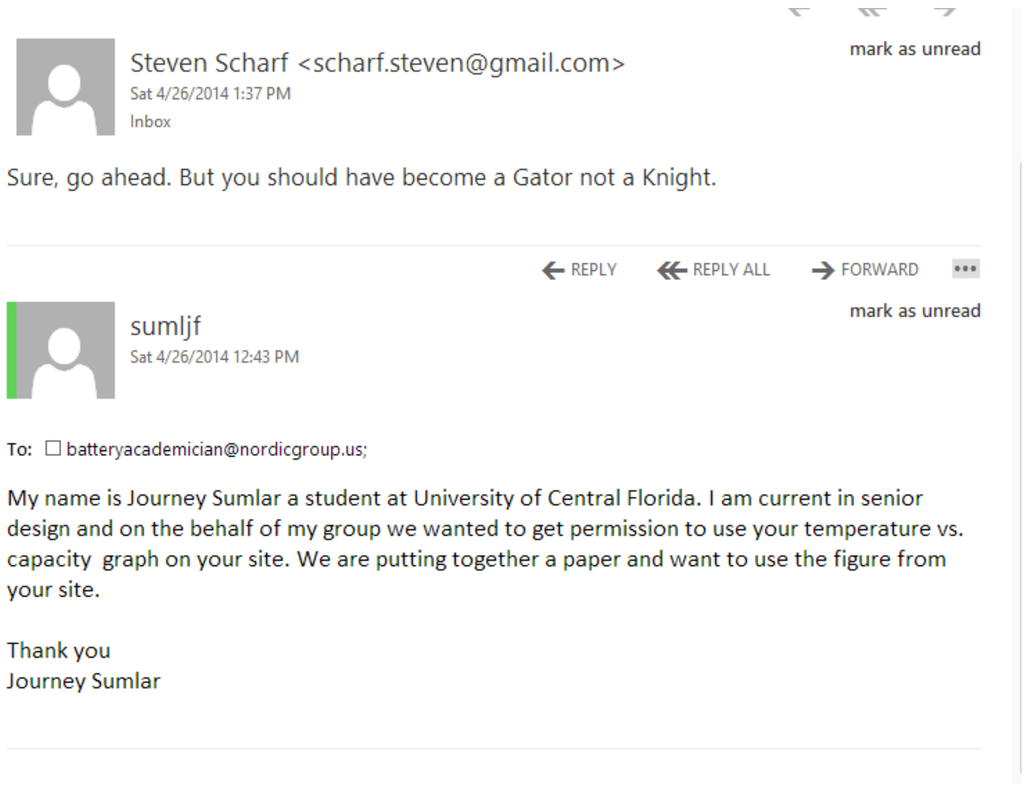
Hello,

My name is Journey Sumlar a student at University of Central Florida. I am current in senior design and on the behalf of my group we wanted to get permission to use your alkaline vs NiMH graph on your site. We are putting together a paper and want to use the figure from your site.

Thank you
Journey Sumlar

Permission Status:Approved

Capacity vs. Temperature Figure
http://batterydata.com/



Steven Scharf <scharf.steven@gmail.com>
Sat 4/26/2014 1:37 PM
Inbox

Sure, go ahead. But you should have become a Gator not a Knight.

sumljf
Sat 4/26/2014 12:43 PM

To: □ batteryacademician@nordicgroup.us;

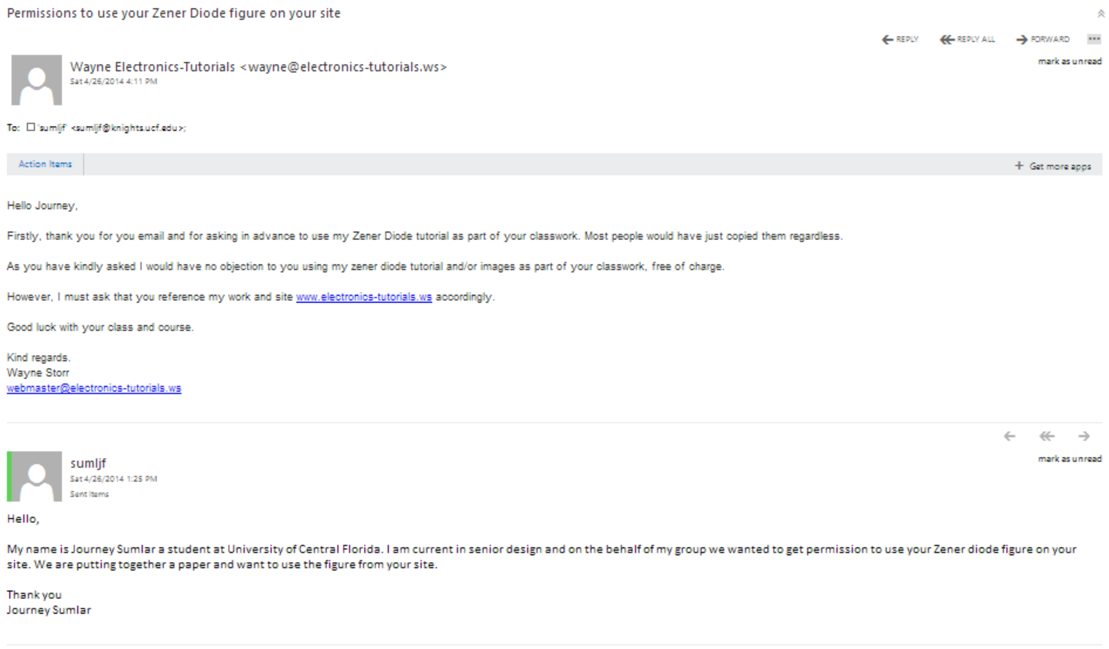
My name is Journey Sumlar a student at University of Central Florida. I am current in senior design and on the behalf of my group we wanted to get permission to use your temperature vs. capacity graph on your site. We are putting together a paper and want to use the figure from your site.

Thank you
Journey Sumlar

Permission Status: Approved

Zener Diode

http://www.electronics-tutorials.ws/diode/diode_7.html



Permissions to use your Zener Diode figure on your site

Wayne Electronics-Tutorials <wayne@electronics-tutorials.ws>
Sat 4/26/2014 4:11 PM

To: □ 'sumljf' <sumljf@knights.ucf.edu>;

Hello Journey,

Firstly, thank you for you email and for asking in advance to use my Zener Diode tutorial as part of your classwork. Most people would have just copied them regardless.

As you have kindly asked I would have no objection to you using my zener diode tutorial and/or images as part of your classwork, free of charge.

However, I must ask that you reference my work and site www.electronics-tutorials.ws accordingly.

Good luck with your class and course.

Kind regards.
Wayne Storr
webmaster@electronics-tutorials.ws

sumljf
Sat 4/26/2014 1:25 PM
Sent Items

Hello,

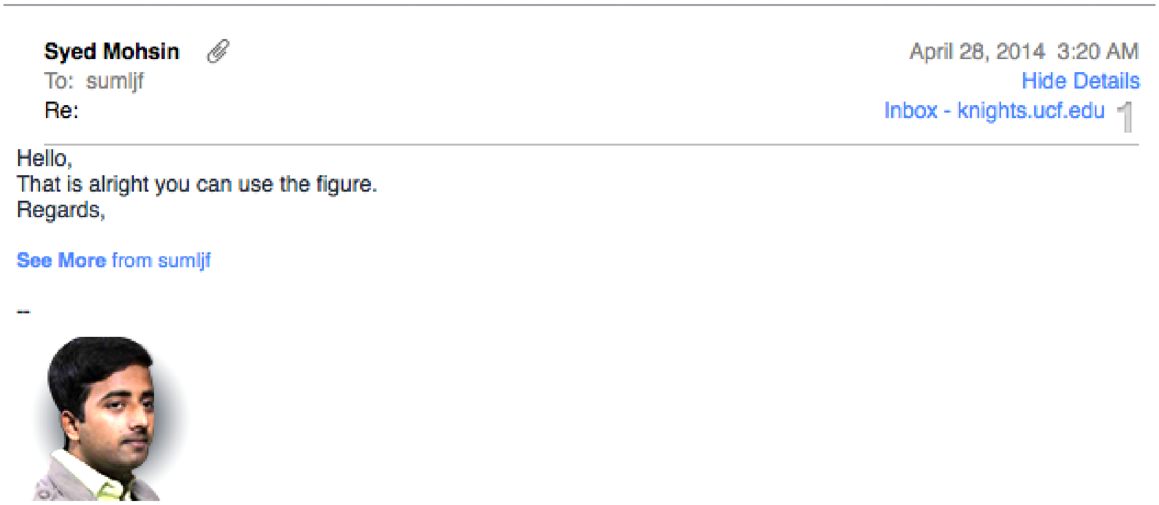
My name is Journey Sumlar a student at University of Central Florida. I am current in senior design and on the behalf of my group we wanted to get permission to use your Zener diode figure on your site. We are putting together a paper and want to use the figure from your site.

Thank you
Journey Sumlar

Permission Status: Accepted

Switching Regulator

http://www.circuitstoday.com/few-switching-regulator-circuits



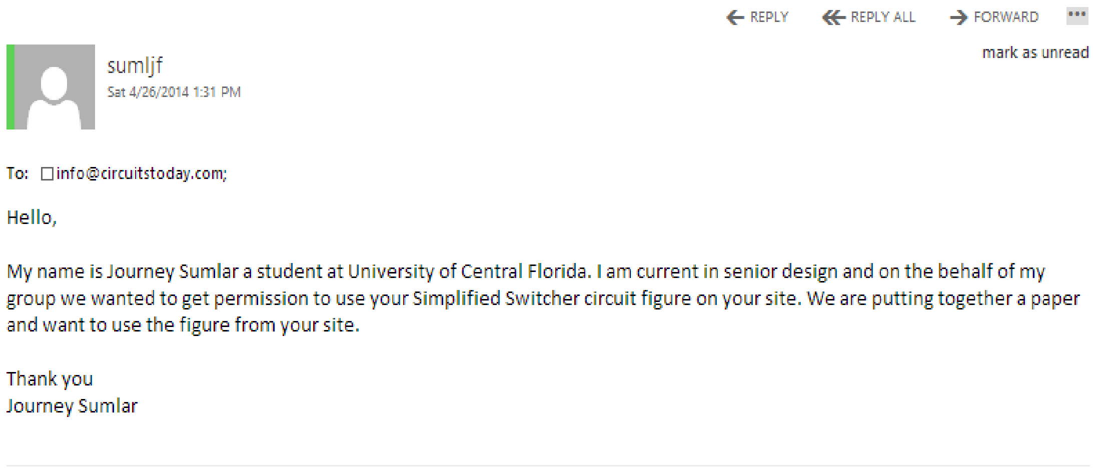
Syed Mohsin April 28, 2014 3:20 AM

To: sumljf

Re:

Hello,
That is alright you can use the figure.
Regards,

--



sumljf
Sat 4/26/2014 1:31 PM

To: □info@circuitstoday.com;

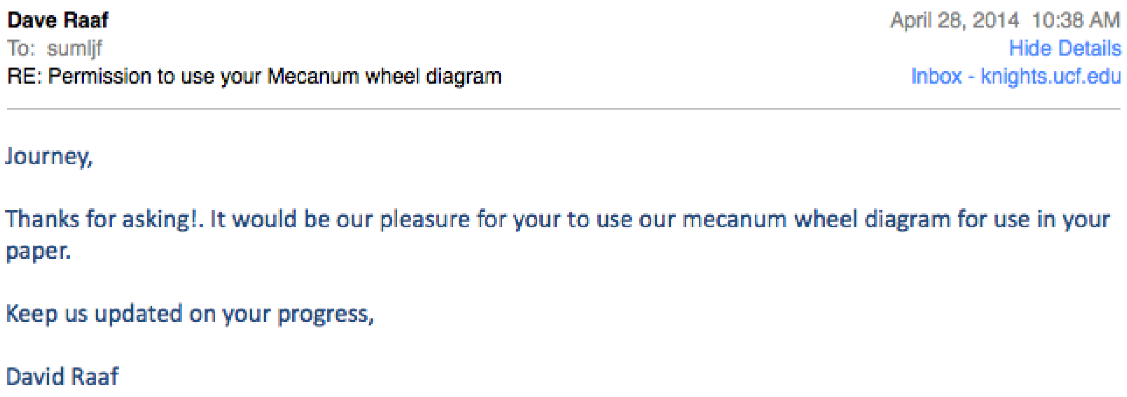
Hello,

My name is Journey Sumlar a student at University of Central Florida. I am current in senior design and on the behalf of my group we wanted to get permission to use your Simplified Switcher circuit figure on your site. We are putting together a paper and want to use the figure from your site.

Thank you
Journey Sumlar

Permission Status: Accepted

Mecanum Wheel Diagram
http://www.icrobotics.org/contact



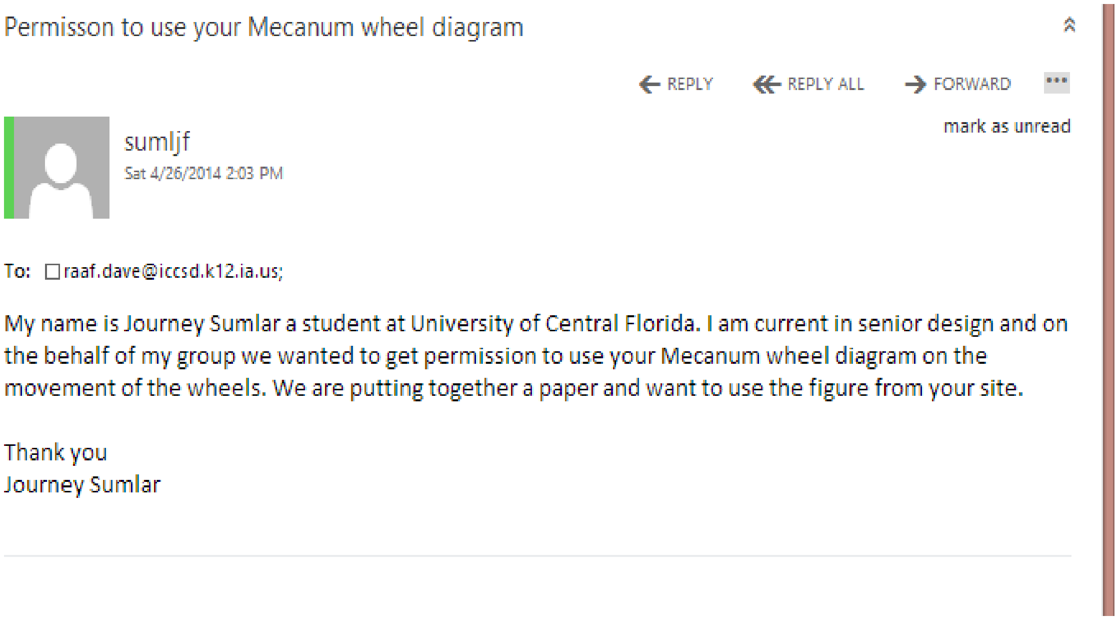
Dave Raaf April 28, 2014 10:38 AM
To: sumljf
RE: Permission to use your Mecanum wheel diagram

Journey,

Thanks for asking!. It would be our pleasure for your to use our mecanum wheel diagram for use in your paper.

Keep us updated on your progress,

David Raaf



Permisson to use your Mecanum wheel diagram

sumljf
Sat 4/26/2014 2:03 PM

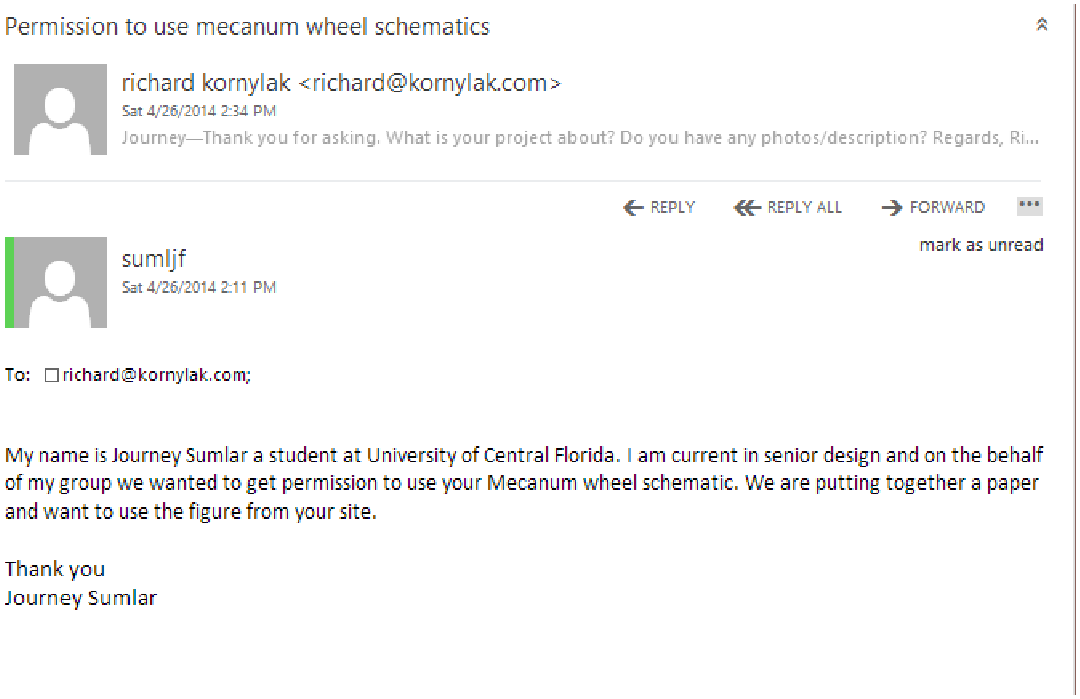
To: □ raaf.dave@iccsd.k12.ia.us;

My name is Journey Sumlar a student at University of Central Florida. I am current in senior design and on the behalf of my group we wanted to get permission to use your Mecanum wheel diagram on the movement of the wheels. We are putting together a paper and want to use the figure from your site.

Thank you
Journey Sumlar

Permission Status: Approved

Mecanum Wheel Dimension Figure 3.13.1-1
http://www.kornylak.com/images/wheel/6-mecanum-diag.jpg



Permission to use mecanum wheel schematics

richard kornylak <richard@kornylak.com>
Sat 4/26/2014 2:34 PM
Journey—Thank you for asking. What is your project about? Do you have any photos/description? Regards, Ri...

sumljf
Sat 4/26/2014 2:11 PM

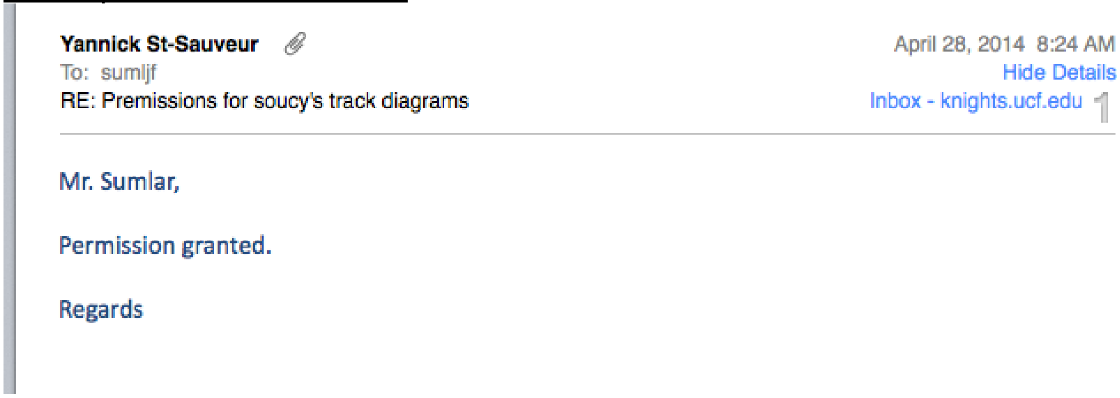
To: □richard@kornylak.com;

My name is Journey Sumlar a student at University of Central Florida. I am current in senior design and on the behalf of my group we wanted to get permission to use your Mecanum wheel schematic. We are putting together a paper and want to use the figure from your site.

Thank you
Journey Sumlar

Permission Status: Accepted

Soucy Tracks
http://www.soucy-track.com/en-CA/about/avantages/agricultural-sector/performance-vehicule

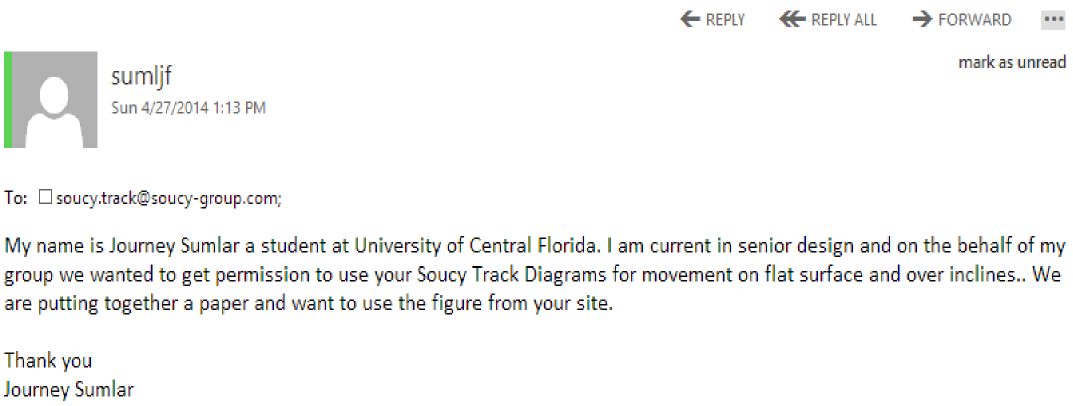


Yannick St-Sauveur
To: sumljf
RE: Premissions for soucy's track diagrams
April 28, 2014 8:24 AM

Mr. Sumlar,

Permission granted.

Regards



sumljf
Sun 4/27/2014 1:13 PM

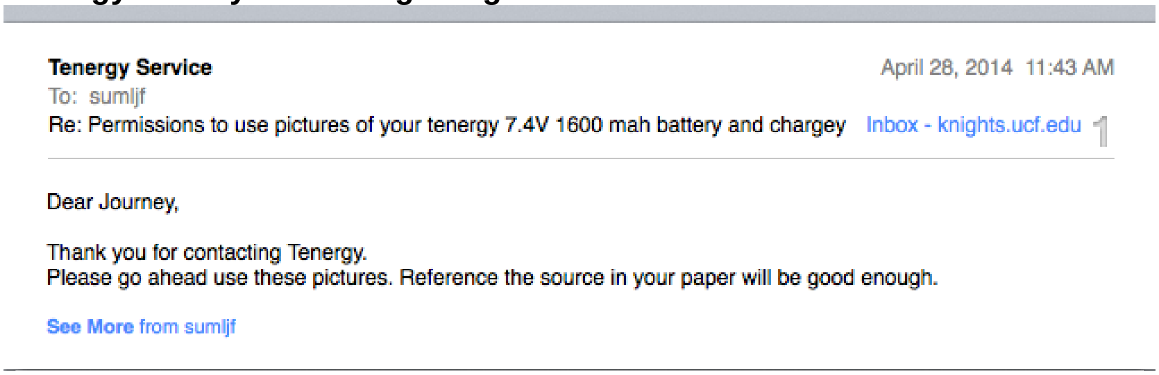
To: □soucy.track@soucy-group.com;

My name is Journey Sumlar a student at University of Central Florida. I am current in senior design and on the behalf of my group we wanted to get permission to use your Soucy Track Diagrams for movement on flat surface and over inclines.. We are putting together a paper and want to use the figure from your site.

Thank you
Journey Sumlar

Permission Status: Approved

Tenergy Battery and Charger Figures 4.5-2 and 4.5-3

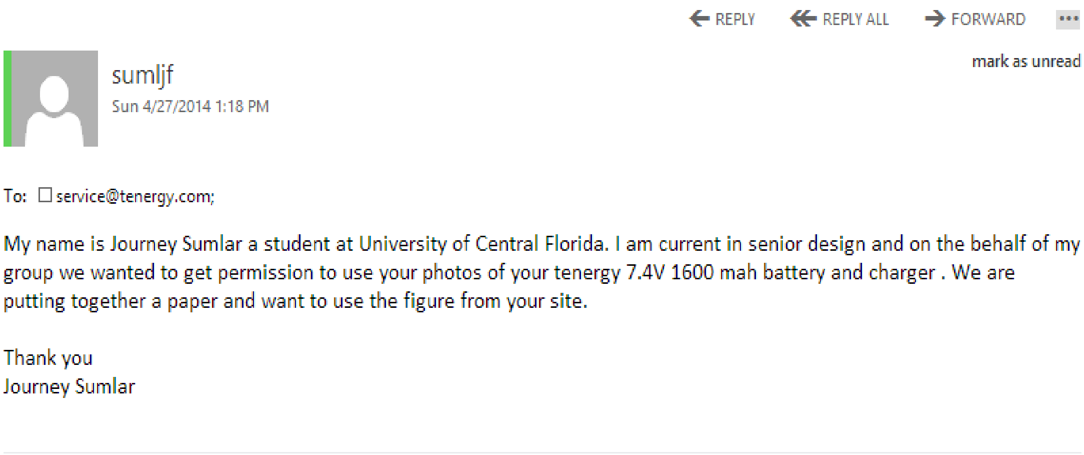


Tenergy Service April 28, 2014 11:43 AM
To: sumljf
Re: Permissions to use pictures of your tenergy 7.4V 1600 mah battery and chargey

Dear Journey,

Thank you for contacting Tenergy.
Please go ahead use these pictures. Reference the source in your paper will be good enough.

See More from sumljf



sumljf
Sun 4/27/2014 1:18 PM

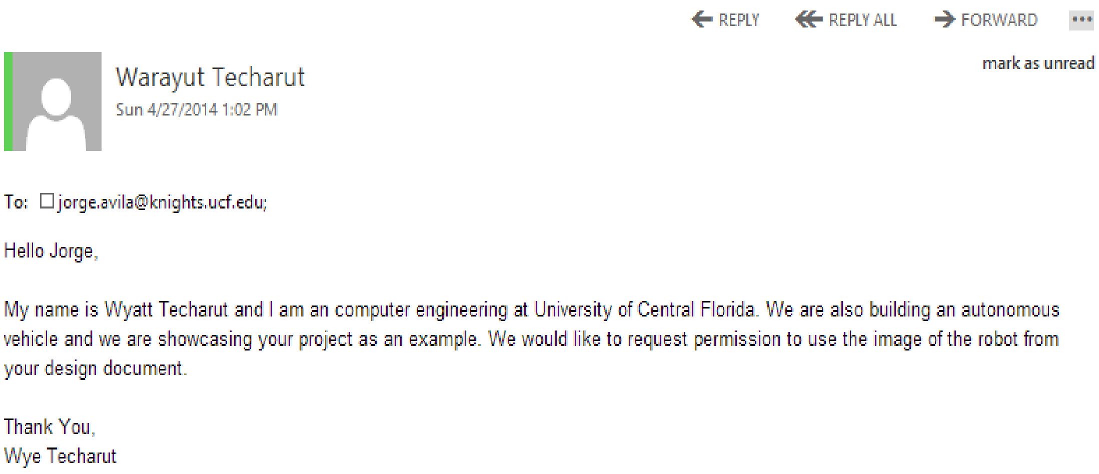
To: □service@tenergy.com;

My name is Journey Sumlar a student at University of Central Florida. I am current in senior design and on the behalf of my group we wanted to get permission to use your photos of your tenergy 7.4V 1600 mah battery and charger . We are putting together a paper and want to use the figure from your site.

Thank you
Journey Sumlar

Permission Status: Accepted

Autonomous Ball Collector Figure 3.1-2



Warayut Techarut
Sun 4/27/2014 1:02 PM

To: □jorge.avila@knights.ucf.edu;

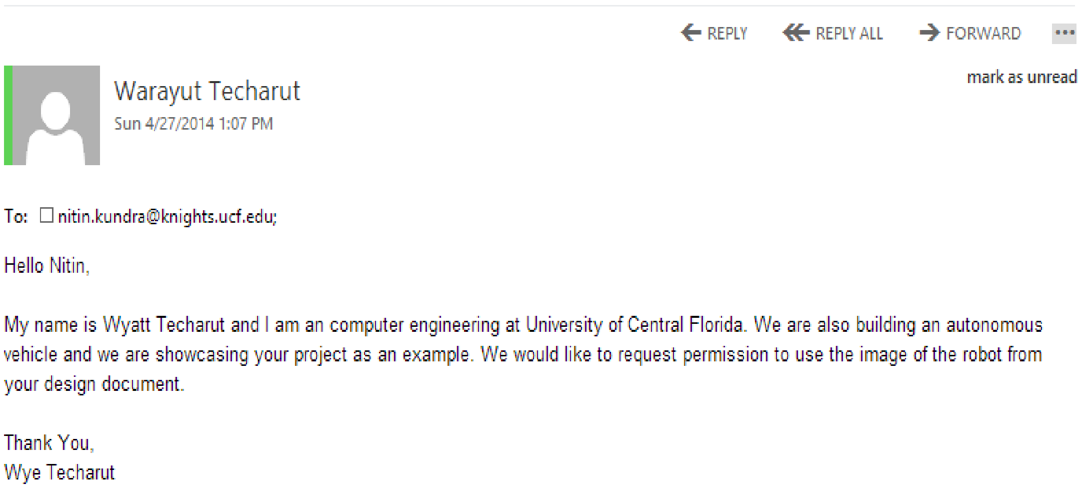
Hello Jorge,

My name is Wyatt Techarut and I am an computer engineering at University of Central Florida. We are also building an autonomous vehicle and we are showcasing your project as an example. We would like to request permission to use the image of the robot from your design document.

Thank You,
Wye Techarut

Permission Status: Pending

KnightCop Project Figure 3.1-3



Warayut Techarut
Sun 4/27/2014 1:07 PM

To: □nitin.kundra@knights.ucf.edu;

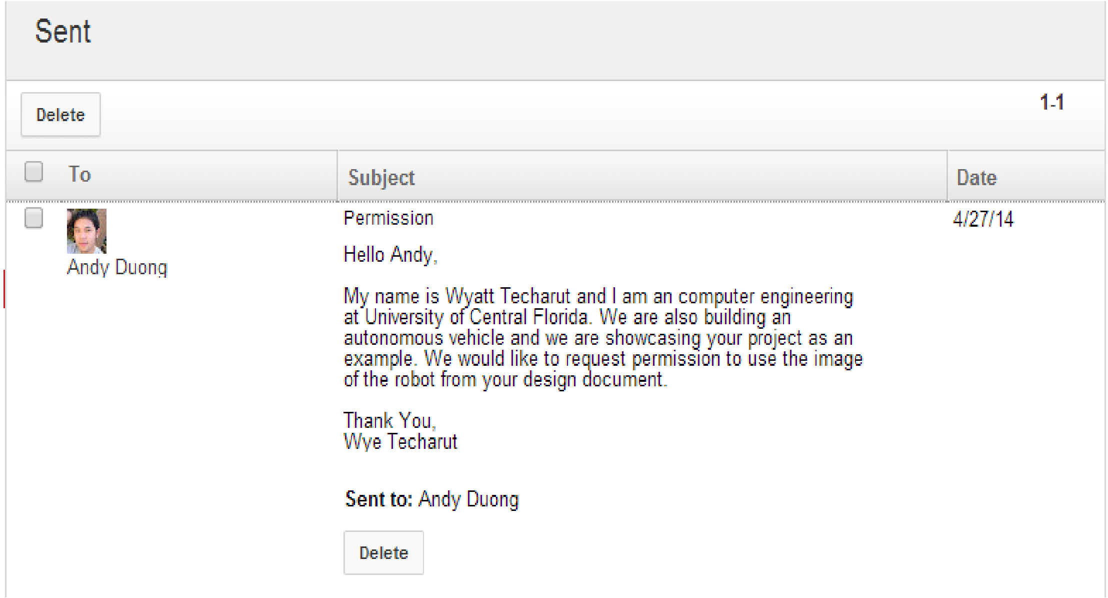
Hello Nitin,

My name is Wyatt Techarut and I am an computer engineering at University of Central Florida. We are also building an autonomous vehicle and we are showcasing your project as an example. We would like to request permission to use the image of the robot from your design document.

Thank You,
Wye Techarut

Permission Status: Pending

Autonomous Tank Project Figure 3.1-1



Andy Duong

Permission

4/27/14

Hello Andy,

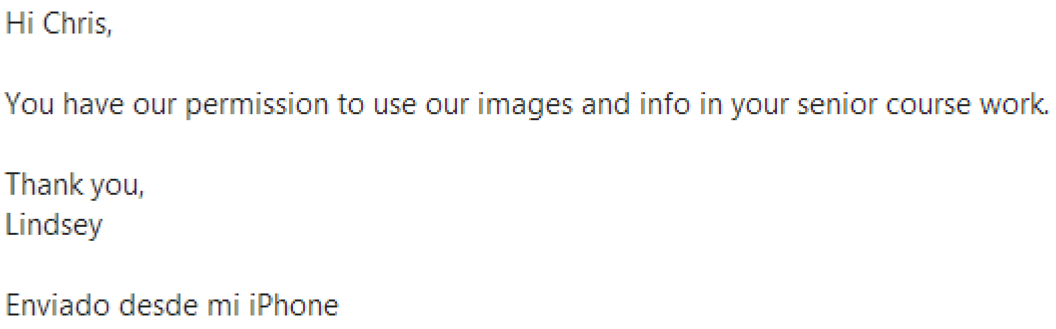
My name is Wyatt Techarut and I am an computer engineering at University of Central Florida. We are also building an autonomous vehicle and we are showcasing your project as an example. We would like to request permission to use the image of the robot from your design document.

Thank You,
Wye Techarut

Sent to: Andy Duong

Permission status: Pending

VexRobotics

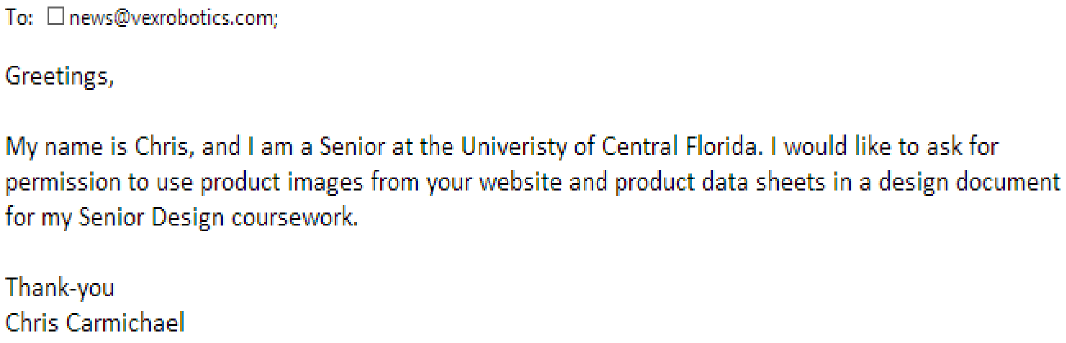


Hi Chris,

You have our permission to use our images and info in your senior course work.

Thank you,
Lindsey

Enviado desde mi iPhone



To: □ news@vexrobotics.com;

Greetings,

My name is Chris, and I am a Senior at the Univeristy of Central Florida. I would like to ask for permission to use product images from your website and product data sheets in a design document for my Senior Design coursework.

Thank-you
Chris Carmichael

Permission Granted

Vishay

http://www.vishay.com/company/privacy-legal/

> *”Vishay Intertechnology, Inc. permits browsing of the Vishay Intertechnology, Inc. web site and the necessary caching of the Web Content onto your computer. Vishay Intertechnology, Inc. permits the download, reproduction, display and internal distribution of the Web Content posted on this site solely for informational and non-commercial (the use of the web Content in purchasing and design applications are not considered commercial uses) or personal use, provided that the Web Content is not modified and provided further that all copyright and proprietary notices remain as they appear within such Web Content. Vishay Intertechnology, Inc. further grants to educational institutions (specifically K-12, universities and community colleges) permission to download, reproduce, display and distribute the Web Content posted on this site solely for use in the classroom, provided that such institutions identify Vishay Intertechnology, Inc. as the source of the Web Content and include the following credit line: "Courtesy of Vishay Intertechnology".”*

Texas Instruments
http://www.ti.com/corp/docs/legal/termsofuse.shtml

> *“TI grants permission to download, reproduce, display and distribute the Materials posted on this site solely for informational and non-commercial or personal use, provided that you do not modify such Materials and provided further that you retain all copyright and proprietary notices as they appear in such Materials. TI further grants to educational institutions (specifically K-12, universities and community colleges) permission to download, reproduce, display and distribute the Materials posted on this site solely for use in the classroom, provided that such institutions identify TI as the source of the Materials and include the following credit line: "Courtesy of Texas Instruments". Unauthorized use of any of these Materials is expressly prohibited by law, and may result in civil and criminal penalties. This permission terminates if you breach any of these terms and conditions. Upon termination you agree to destroy any Materials downloaded from this site.”*

Sparkfun

> **“Product Photos:** SparkFun product photos may be used without permission for educational purposes (research papers, school projects, etc.). Permission must be granted for commercial use and proper credit to SparkFun must be given. For inquiries about the use of our product photos or permission to use them, please contact marketing@sparkfun.com.”