SAFER (Safety Autonomous Following Escort Robot) Knights Project

University of Central Florida
College of Engineering
Department of Electrical and Computer Engineering

Senior Design 1 Project Documentation

Group 16

Megan White	Electrical Engineering
Jordan Harty	Electrical Engineering
Jonathan Dillard	Computer Engineering
Michael Burton	Computer Engineering

Table of Contents

1 Executive Summary

Whether it be late night studying or visiting a friend, students often travel across campus after dark. For those who feel unsafe walking alone on campus at night, resources like SEPS (Safe Escort Patrol Service) and Knight Ride are there to help make sure you make your way home successfully. Sometimes, these resources are not available or desired. The purpose of the SAFER Knights Project is to provide an alternative to such pre-established precautions on the UCF campus. An autonomous vehicle equipped with various sensors and lights follows students home to provide a feeling of security when walking alone day or night. It will follow the student to wherever they need to go on campus. Features such as: an emergency protocol, lights, and speakers are available in order to provide peace of mind to students. Emergency mode involves the user pressing a button on their phone to enable it, and the robot will start flashing lights and start a siren to alert any nearby passerby. Since the robot should be user-friendly, so a mobile app will be developed to easily connect with the robot and provide further information and access to Emergency mode. The robot will be mainly using a Bluetooth connection and PixyCam to follow the user.

From a technical perspective, the robot has to be of a decent size to hold necessary equipment and move fast enough to keep up with the user. The vehicle base is therefore a Power Wheels, usually a toy for young children, which has now been retrofitted to reach our design objectives. To follow the user, a PixyCam plugged into a Raspberry Pi will use computer vision to communicate with the custom PCB. It will know where the user is via a color code, and through coding will know how far away to stay from the user to avoid collision. A speaker and lights will be mounted on the robot as well to light the surroundings and sound an alarm when necessary. The robot will also have space on the back to allow the user to place their backpack or something small, so they do not have to carry it during travel. All of this is meant to keep the user safe in the best way possible while still being a passive observer.

2 Project Description

The project description section serves as a way to introduce the entirety of reasons and ways the project came about and how it will be enacted. It includes the motivation behind the project, the goals and objectives that hoped to be achieved, the requirement specifications outlining the mandatory achievable outcomes as well as advanced and stretch features, and the House of Quality. Advanced and stretch requirements are not mandatory but would greatly improve the overall presentation and effectiveness of the project. Advanced features are achievable but would take extra time and possibly money to do, so only a few will be targeted. Time permitting, attempts to meet and attempt the stretch features will be made, otherwise, they will not be necessary for the success of the project. The House of Quality is also analyzed to show a comparison between the desires of the project and the actual capabilities of the product.

2.1 Project Motivation

The main motivation of this project is to give students another viable option for them to get home safely day or night. The idea came from previous experiences of trying to contact SEPS only to be told that they were busy and there was a wait until the next escort home. While UCF does attempt to make the campus as safe as possible, it is not always 100% safe, so giving students the ability to feel safer is important. Even in the event of an emergency, users could be able to have a video recording of the event in case evidence is needed. The robot will also be able to act as a deterrent to possible assailants from approaching students that may be walking alone.

2.2 Goals & Objectives

Every knight deserves safer nights. The overall goal of this project is to create an autonomous vehicle that will avoid obstacles and follow students to their desired location. Some stretch goals could include voice commands/warnings, live video streaming, or an app for a checkout/return system.

Objectives must be established in order to accomplish the above goals. The milestones table, on page 8 chronologically lists a series of achievable objectives relating to the group's progress to efficiently finish the SAFER Knights project.

The possible main objectives of the SAFER Knights Project are:

- Provide a sense of safety to the user
- Give students another option to protect themselves when walking alone
- Build a robot that uses Bluetooth capabilities via phone connection
- Use applicable sensors to avoid obstacles in the robot's path
- Provide light to see around the robot

- Record the surroundings as a form of surveillance and possible evidence
- Have an Emergency mode meant to deter possible assailants
- Use a speaker to act as a siren as necessary
- Create a mobile app to improve the user experience
- Design a custom PCB

2.3 Requirement Specifications

Requirement specifications are meant to be a strict guide on how to build a project, how the project should perform, and what the limitations will be. While building the project requirements such as the size of the product, materials used, power consumed, and other things should be taken into account. The project's performance requirements are a more specific view on how the product should behave to achieve the project's goals and objectives, such as how far the product will travel and way of following the user. The various limitations are also factored in here, such as the requirement to have a custom PCB, use cost effective materials, and how the vehicle has to react to objects and avoid them. Below is the list of requirement specifications for this project, broken up into core features, which should all be attained by the time of the project's deadline, advanced features, which the product should have some of to be a more effective project, and stretch features, which will only be done if time allows.

Core Features:
- The vehicle shall be no larger than 4x4x4 ft
- The vehicle shall be capable of following a person via phone connection or infrared beacon within 15 feet
    - Maintain a steady following distance of 5-7 feet
    - Stop within 2 seconds of user stopping
- The vehicle shall go no faster than 5 mph
- The vehicle shall have a total following distance of at least 1 mile
- The vehicle shall be able to store 30 minutes of video
- Phone app with panic button and Bluetooth capabilities

Advanced Features:
- The vehicle shall be weather resistant
- The vehicle shall have a place to hold a bag
- The vehicle shall have a remote-control mode

Stretch Features:
- The vehicle shall go into Emergency Mode per user's request
    - Siren will sound until command terminated by user
    - Flashing lights will flash until command terminated by user
- The vehicle shall have a live video stream whenever in use
- The vehicle shall avoid obstacles bigger than a 4x4x4 inch objects
- The vehicle shall respond to voice commands
- The vehicle shall have a return station within 100 feet of its current location

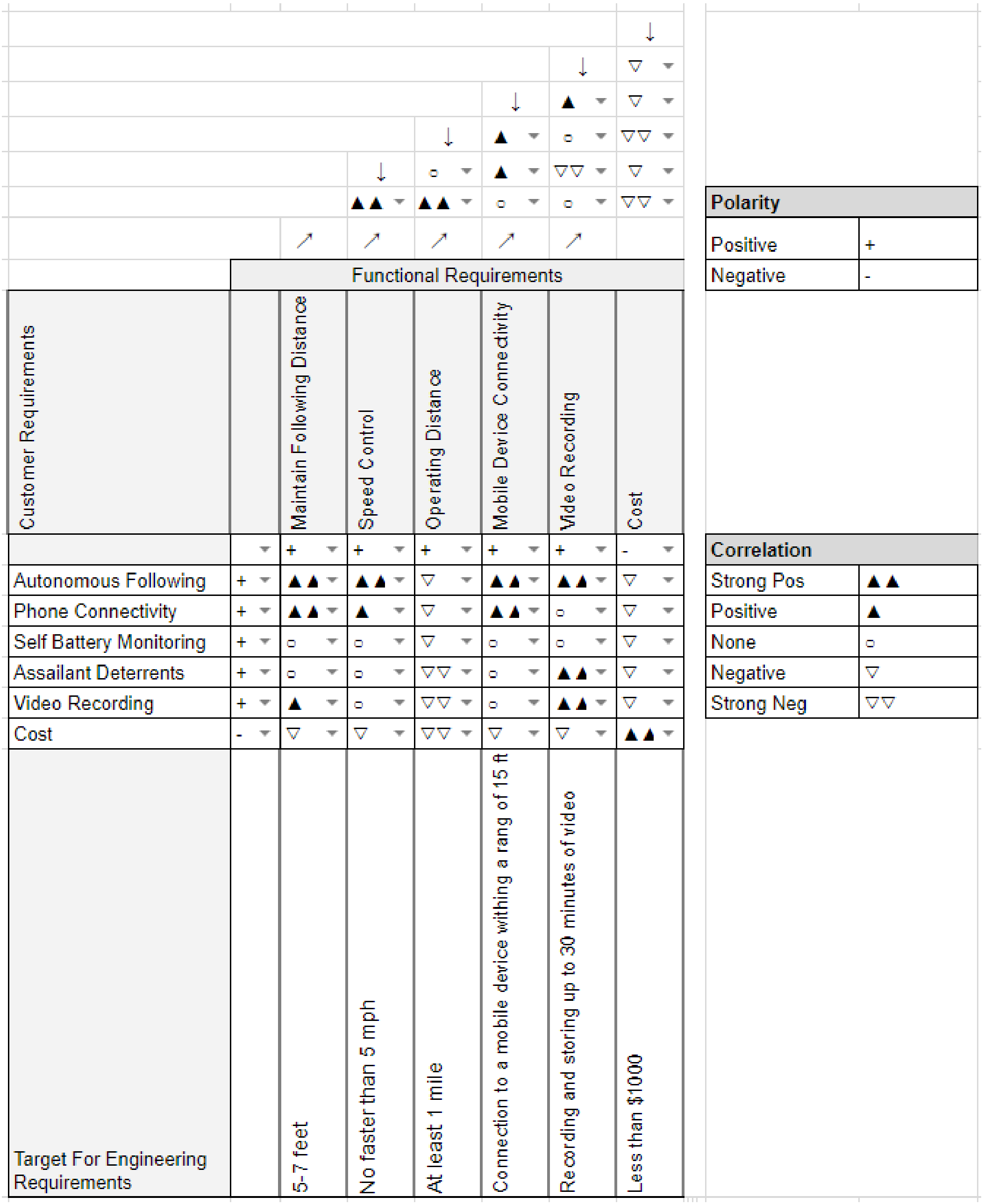
The core features chosen help define the main functionality and requirements set for this project. The size requirement was set so that the vehicle would not be too large and take up too much room on sidewalks while also still being big enough to be easily spotted by everyone in the surrounding area. The following requirements are made to ensure that the vehicle does not get too close to the user and risk possibly running into them or trail too far behind to avoid losing Bluetooth connection to the user. Having a steady following distance a few feet back allows the robot to be close enough to still be connected to the user’s phone via Bluetooth while also being far enough back that it can keep the user’s body in the camera’s field of view for recording purposes. The vehicle must also follow the user at a casual walking pace, so it has no need to go faster than 5 mph. To be able to avoid obstacles, the size of the obstacle must also be defined. The vehicle will be big enough that running over very small objects won’t pose much of a concern, so by comparing a Power Wheels to different sized objects that may impede the Power Wheel’s journey were measured until a conclusion was made that anything bigger than a 4x4x4 inch object should trigger the robot’s object detection algorithm. Part of the point of the robot is surveillance, so recording the surrounding areas for the comfort of the user is important. Therefore, at least 30 minutes of video will be stored on a memory storage device, although this amount can be increased if it is found that users take more than 30 minutes to walk to their destination.

The advanced features were chosen as additional features that are achievable and would be beneficial to the project, while also not being crucial to the core of the design of the project. While having a vehicle that is weather resistant would be ideal, for the scope of the project it was found to not necessarily be a critical feature since testing will be done when the weather is not inclement. Having a return location to get the vehicle out of the way of possible cars and traffic would also be ideal, but not absolutely necessary for achieving the goal of this project, which is to provide the user with a sense of safety and security as they go to their destination at night.

The stretch features are meant as optimistic goals that could be achieved if all other features and abilities have been met with time left over before the project’s due date. For this, instead of having the robot move out of the way by 10 feet, the idea was that the robot could go to a nearby return station to fully make sure it’s not in the way of anyone. Emergency mode is a special feature that enables the robot to potentially help the user by deterring any possible assailants through the use of sirens and rapidly blinking lights. Emergency mode can be enacted through a mobile app, which will be used both as a way to help the user pair their phone via Bluetooth and also provide a button that can activate Emergency Mode. Being able to respond to voice commands would be incredibly useful if the user were in trouble and was unable to use their phone to ask for help from the robot. However, voice commands pose a lot of trouble and work, so that is why it is considered a stretch feature. The live stream feature would be useful to people like the campus police or other security people for monitoring the user’s journey back home live. This way others could make sure the user got home safely. For the scope of this project, a simple storage device like an external hard drive will suffice unless there is time leftover to implement this feature into the final design.

2.4 House of Quality

The figure below shows the House of Quality. The House of Quality (HOQ) shows the correlation between the customer / marketing requirements and the functional / engineering requirements. These correlations are the relation of each to the improvement of each other. The polarity of each requirement is also displayed next to or under it. This polarity is used to keep a positive or negative correlation consistently good across all cells in the matrix. This means that a positive correlation will always be good or bad for all cases in a given House of Quality. These correlations also can be weak or strong correlations.



							↓
						↓	▽
					↓	▲	▽
				↓	▲	○	▽▽
			↓	○	▲	▽▽	▽
			▲▲	▲▲	○	○	▽▽
		↗	↗	↗	↗	↗	

Customer Requirements		Maintain Following Distance	Speed Control	Operating Distance	Mobile Device Connectivity	Video Recording	Cost
		+	+	+	+	+	-
Autonomous Following	+	▲▲	▲▲	▽	▲▲	▲▲	▽
Phone Connectivity	+	▲▲	▲	▽	▲▲	○	▽
Self Battery Monitoring	+	○	○	▽	○	○	▽
Assailant Deterrents	+	○	○	▽▽	○	▲▲	▽
Video Recording	+	▲	○	▽▽	○	▲▲	▽
Cost	-	▽	▽	▽▽	▽	▽	▲▲
Target For Engineering Requirements		5-7 feet	No faster than 5 mph	At least 1 mile	Connection to a mobile device withing a rang of 15 ft	Recording and storing up to 30 minutes of video	Less than $1000

Polarity	
Positive	+
Negative	-

Correlation	
Strong Pos	▲▲
Positive	▲
None	○
Negative	▽
Strong Neg	▽▽

Figure: House of Quality

Each of the engineering requirements will also come with a target for the engineering requirement. These are the overall goals or specifics for that requirement. While the marketing goals are more overarching and broader, the engineering requirements must be stricter by nature. In the marketing requirements, we can be more overarching because it is not an exact science. The engineering requirements cannot be as vague because in engineering, number will influence the solution to the problem. These requirements should have a range. For example, “SAFER should follow the user maintaining a distance of 5 to 7 feet.” This range will give the engineering team more room to create a system that may be better or cheaper as a slightly less than optimal operational point. These numbers can also be a ceiling. For example, no more than five miles per hour. This settlement can also apply to flooring values too. These values are subject the change as the product is developed. Some lessons are to be learned about the nature of the problem that may change the requirements. In the beginning these are mostly just predictions.

As just stated, a positive polarity will indicate that more is better, and a negative polarity will indicate less is better. As an example, if the quality of the build material goes up (engineering requirement), then the cost (marketing requirement) will also go up. Even though they both are going up, this trend has a negative correlation because the cost has a negative polarity. So, while the total cost is raising, the metric of cost is decreasing because the marketing requirement wants the cost to be as low as possible. It is possible for an engineering requirement and a marketing requirement to have no correlation between each other. For example, the quality of one hardware component used for a special feature with its own marketing requirement can have no correlation with the marketing requirement of another feature. This does bring up a point about the two types of requirements.

The engineering requirements can have a marketing requirement that is very similar with a large correlation. These are common because some hardware and software components are specifically designed to satisfy these requirements, but it works putting both in the House of Quality because these components can influence other marketing requirements such as cost.

Another important note is that these requirements are technology independent. That means that we do not include the ways in which we want to achieve these goals. For example, we would not explicitly stage that we will be using computer vision in one of the requirements because we may find a better solution to the core problem.

Above (in the “roof”) are the functional requirements correlated to each other. The roof correlations branch diagonally to the right from the arrow above them in a straight line. The mark that reflects each pair lies along the intersection of each diagonal mark directly above the other. A more solid way to see this is by looking at the “Maintain Following Distance” requirement at the top all the way to the left. The correlation with each of the other requirements will be the top most mark above the other requirement. This is because of its diagonal line indicated by the arrow above it. These correlations work in the same way as the ones in the center of the House of Quality. Meaning they follow the same trends as described in the above section.

Looking at our specific House of Quality, we have six marketing requirements and six engineering requirements. This gives us the square grid you see. Here is a brief description of the marketing requirements:

- “Autonomous Following”. This is referring the SAFERS capability to follow the user without any controlling inputs from the user. This can be achieved through a multitude of methods described in other sections. As stated above the requirements should be agnostic to the technology being used to achieve them. This allows a protective layer for the engineers to try different technology that may solve the core problem in a better way. The polarity of this is positive because the better the autonomous following gets, the better the product will satisfy the requirement
- “Phone Connectivity”. This means that SAFER will be able to connect to a phone for two-way communication between the two. This will allow the user to send commands to SAFER like triggering a distress mode. This will also allow SAFER to get information about the user's locations and send it signals back to possibly indicate things like battery levels. This is a key piece of the puzzle to make SAFER a successful product. The polarity of this is positive because the better the phone connectivity gets, the better the product will satisfy the requirement and the customer experience.
- “Self-Battery Monitoring” refers the SAFER’s ability to know how much power is left in its battery supply to make judgments on how it can operate. It can also use this information to inform the user so that they can also make informed decisions. Decisions like whether to embark on a long journey or when to charge the device. This goal is more in the stretch category for our project and will depend heavily on extra time. The polarity of this is positive because the better the self-battery monitoring gets, the better the product will satisfy the requirement and the customer experience.
- “Assailant Deterrents” refers to SAFER’s ability to go into an emergency mode and take precautions to protect the user. One of these will be a way thought sound or light or other means to deter the possible assailant form harming the user and achieving the goal of SAFER. Again, this requirement was intently left vague because we do not want to tie down the specific method that may be used to archive this requirement. The polarity of this is positive because the better the assailant deterrents gets, the better the product will satisfy the requirement and the overall goal of the product.
- “Video Recording” refers to SAFER’s possible ability to record its environment for the purpose of deterring incidents or recording them. A camera can be a powerful deterrent for wrongdoers who do not want to be caught in the act. The camera will also have the ability to record any incident that may happen for examination for legal or criminal reasons. The polarity of this is positive because the better the video recording gets, the better the product will satisfy the requirement and the customer experience.

- "Cost" refers the overall cost of the product. This is the only requirement is the only one that has a negative polarity. This is because as the cost gets larger the requirement is less and less satisfied. This is also the only requirement that will almost always have a correlation with each other requirement in the House of Quality. The lower the cost the better for the customer, but the lower the quality of the product. This will be a hard requirement to balance with all the others.

Now looking at the engineering / functional requirements. There are six of these engineering requirements also. As stated in a previous section, there may be a lot of overlap between these requirements. This is due to the fact that the marketing requirements are developed first and then from them the engineering requirements precipitate. This means some of them will seem to directly correlate to another marketing requirement, but this is not always the case. In this section we will also discuss the correlations between the marketing requirements as well as the other engineering requirements. Here is a brief description of the engineering requirements:

- "Maintain Following Distance" is defined to have a value of five to seven feet. This means as SAFER follows the user around on a given trip SAFER will be able to keep up with the user without the potential to run into them. This distance should keep SAFER within a safe stopping distance, while also allowing the robot the keep a large part of its environment in its field of view (not blocked by the presence of the user) regardless of the method used by the robot to actually follow the user.
    - Marketing vs Engineering:
        - Autonomous Following: The correlation between maintain following distance and autonomous following is a strong positive because the following distance is a derivative of the act of autonomous following. In the act of autonomies, it is necessary to define a following distance.
        - Phone Connectivity: The correlation between maintaining following distance and phone connectivity is strong because in half of the autonomous following methods we have researched there is a phone connection requirement.
        - Self-Battery Monitoring: The correlation between maintaining following distance and self-battery monitoring is nonexistent because the state of the battery will not affect the following distance of the robot or how it will achieve the requirement. The better the follow distance accuracy gets, the worse the battery life may get due to better computers and other equipment using more power for better results, but this requirement only measures monitoring.
        - Assailant Deterrents: The correlation between maintaining following distance and assailant deterrents is nonexistent because while in emergency mode, the robot will cease to follow the user. This will make the following distance a moot requirement until it resumes regular operation.

        - Video Recording: The correlation between maintaining following distance and video recording is a weak positive because in our research some of the devices used for the following mechanic will involve cameras for things like computer vision. This means as we use better cameras the recording for SAFER can also improve as well. It is not a strong positive because only some of these methods will correlate, and all of them do not address things like storage for the video assets.
        - Cost: The correlation between maintaining following distance and cost is a weak negative because making the robot better at maintaining a defined following distance will mostly like increase the overall cost of SAFER due to the add quality or number of components to do so.
    - Engineering vs Engineering:
        - Maintaining Following Distance: The correlation is a strong positive because requirements correlated to themselves will always have a strong positive correlation just due the nature of the way the terms are defined.
        - Speed Control: The correlation between maintaining following distance and speed control is a strong positive because to properly follow the user, SAFER will need to control its speed. As the speed control of SAFER improves so will the maintenance of the following distance. An example of this is if there was very little speed control to the point where SAFER was only able to go one speed and be stopped. If the user was not traveling at that one speed that SAFER can move, then the robot will constantly have to stop and start to keep the user in the defined following distance. Therefore, these two requirements have such a strong correlation.
        - Operating Distance: The correlation between maintaining following distance and operating distance is nonexistence because the operation distance is related to the amount of power the robot has. This means that the source for both to improve is different. The only way these can have any correlation is though the speed controller but that will be discussed in that section.
        - Mobile Device Connectivity: The correlation between maintaining following distance and mobile device connectivity is a weak positive because there are some methods for measuring following distance that can involve a phone connection and two-way data transfer.
        - Video Recording: The correlation between maintaining following distance and video recording is a weak positive because as stated before, there are some methods for the following mechanic that can

use a camera for computer vison. This means as the camera gets better for computer vision so will the Carma for recording.

    - Cost: While this cost requirement is in the engineering category the same factors apply to it is as the cost requirement in the marketing requirement above. For that reason, please see the above section for analysis.

- "Speed Control" is defined as no faster than five miles per hour. This requirement means that SAFER will be able to vary its speed to comfortably follow the user at an array of speed. This means if the robot only has one speed it can travel at and the user is traveling at a slower speed than that one speed of the SAFER. The robot will constantly have to stop and start again to match the speed of the user this is also true if the user is traveling at a speed faster than the one speed of the robot. The user would have to wait for the robot to catch up every so often because it is traveling too slow. This is also not a good user experience. This means that SAFER will need to be able to travel at varying speeds while it follows the user. With some experience we have in an adjacent subject we know that people usually have a walking pace of around three miles per hour and up to around five miles per hour. This knowledge comes from expertise as a hiking and backpacking hobbies. This means that SAFER would be able to follow most any person as they walk to their destination. In the current definition of this requirement we do not allow the robot to follow the user at a running pace to their destination. We do not allow the robot to follow the user at the faster pace for two reasons. First of which is due to the added complexity it would add to the system. This would require SAFER to be able to process signals and make decisions with less time between samples. This added speed can do things to shake the robot more while moving making accurate reading of things like the Infrared sensors and possible computer vision cameras harder to obtain. With these two points it would then make the overall quality of the experience less then desirable for factors like autonomous following. The second reason for keeping the speed to a walking pace will keep SAFER from becoming unsafe itself. With this added speed and less control over the robot the robot may be able to cause harm to person or property. This means as the robot is traveling faster and having a harder time sensing its environment it may collide with the user. This would be the exact opposite outcome we would desire from the user's experience. SAFER is meant to make the user feel SAFER and more comfortable in situations that can cause anxiety, very similar to Robot Cop. This added speed can also become a hazard to property as well. If the robot were to collide with an object it is much more likely to cause damage at higher speeds. The also includes the robot, if SAFER were to collide with an object, it is possible that is could damage its sensors or driving mechanisms. This would not allow the user to complete their journey with the robot. After making these two points we would also like to say that we think that most people would not want SAFER to be a running buddy. We think its applications lie much closer to an escort to your dorm from a late-night study at the library.

  - Marketing vs Engineering:

    - Autonomous Following: The correlation between SAFER’s engineering requirement related to speed control and autonomous following is a strong positive. This relation is because the autonomous system will have a direct control over the speed of the robot. This means that these requirements can almost be thought of in one in the same.
    - Phone Connectivity: The correlation between SAFER’s engineering requirement related to speed control and phone connectivity is week positive. This is because some of the following mechanics we have researched have suggested the use of signal strength as an indicator of distance and therefor informing the speed of the robot to maintain a proper distance.
    - Self-Battery Monitoring: The correlation between SAFER’s engineering requirement related to speed control and self-battery monitoring is nonexistent. This is because the battery energy levels have no effect on the speed controllability of the robot as it follows the user.
    - Assailant Deterrents: The correlation between SAFER’s engineering requirement related to speed control and assailant deterrents is nonexistence. This is because the robot will remain stationary while in assailant deterrent mode. This will obviously not require any speed control to do so.
    - Video Recording: The correlation between SAFER’s engineering requirement related to speed control and video recording is nonexistence. This is because SAFER will be continually recording regardless of the speed at which it is traveling.
    - Cost: The correlation between SAFER’s engineering requirement related to speed control and cost is a weak negative. This is do the cheapness of the parts required to do speed control on SAFER.
  - Engineering vs Engineering:
    - Maintaining Following Distance: The correlation between the robot's speed control and autonomous following is a strong positive. For a more in-depth analysis of the cause of this correlation please the above section. It will describe the come of the pros and cons of the relationship and why the final verdict was reached. The entirety of the description that is alluded to is not included in this section for the sake of brevity.
    - Speed Control: The correlation is a strong positive because requirement correlated to themselves will always have a strong positive correlation just do the nature of the way the terms is defined.

        - Operating Distance: The correlation between SAFER's engineering requirement related to speed control and operating distance is a strong positive. This is because, like autonomous following, SAFER will need to utilize the speed controller to maintain a proper operating distance.
        - Mobile Device Connectivity: The correlation between SAFER's engineering requirement related to speed control and mobile device connectivity is nonexistence. This is due to the same reasons listed in the "Phone Connectivity" subsection directly above this.
        - Video Recording: The correlation between SAFER's engineering requirement related to speed control and video recording are nonexistence. This is due to the same reasons listed in the "Phone Connectivity" subsection directly above this.
        - Cost: While this cost requirement is in the engineering category the same factors apply to is as the cost requirement in the marketing requirement above. For that reason, please see the above section for analysis.

- "Operating Distance" is defined as at least one mile. This metric was chosen as a middle ground. The robot needs to be able to travel far enough for a user to get use out of it. We estimated that the majority of pairs of point a user would travel would be under a mile on the UCF campus. Distance farther than that would be to a place that SAFER might not get utilized due to the lack of possible users.
    - Marketing vs Engineering:
        - Autonomous Following: The correlation between the operating distance and autonomous following is a weak negative. This is due to the battle for the amount of power the battery can hold. As the operating distance increases, the amount of power that can be used for autonomous following decreases.
        - Phone Connectivity: The correlation between the operating distance and phone connectivity is a weak negative. This is due to the same reason above. The battle for energy for the battery.
        - Self-Battery Monitoring: The correlation between the operating distance and self-battery monitoring is a weak negative.
        - Assailant Deterrents: The correlation between the operating distance and assailant deterrents is a strong negative. This is due to all the added weight of the components needed. This will make it harder for SAFER to travel farther.
        - Video Recording: The correlation between the operating distance and video recording is a strong negative. This is because the energy for hard drives to store the media can add up during continues operations.

        - Cost: The correlation between the operating distance and cost is a strong negative because the cost of batteries is large. To make SAFER go farther would take more or larger batteries.
    - Engineering vs Engineering:
        - Maintaining Following Distance: The correlation between the robot's speed control and autonomous following is nonexistent. For a more in-depth analysis of the cause of this correlation please the above section. It will describe the come of the pros and cons of the relationship and why the final verdict was reached. The entirety of the description that is alluded to is not included in this section for the sake of brevity.
        - Speed Control: The correlation between the robot's speed control and autonomous following is a strong positive. For a more in-depth analysis of the cause of this correlation please the above section. It will describe the come of the pros and cons of the relationship and why the final verdict was reached. The entirety of the description that is alluded to is not included in this section for the sake of brevity.
        - Operating Distance: The correlation is a strong positive because requirement correlated to themselves will always have a strong positive correlation just do the nature of the way the terms is defined.
        - Mobile Device Connectivity: The correlation between the operating distance and mobile device connectivity is nonexistent because the power need for device connectivity is very small comparted to the poser needed to move SAFER.
        - Video Recording: The correlation between the operating distance and video recording is a strong negative because of the power needs for hard drive to store the video assets.
        - Cost: While this cost requirement is in the engineering category the same factors apply to is as the cost requirement in the marketing requirement above. For that reason, please see the above section for analysis.
- “Mobile Device Connectivity” is defined as connecting to a mobile device at a distance of at least 15 feet. This would be the maximum range that SAFER would ever want between the user and itself.
    - Marketing vs Engineering:
        - Autonomous Following: The correlation between mobile device connectivity and autonomous following is a strong positive because the device is one of the inputs for knowing where the user is located.

    - Phone Connectivity: The correlation between mobile device connectivity and phone connectivity is a strong positive because these are the same requirement in different categories.
    - Self-Battery Monitoring: The correlation between mobile device connectivity and nonexistent because neither impact the other one.
    - Assailant Deterrents: The correlation between mobile device connectivity and nonexistent because neither impact the other one. There is a possibility of the user being able to trigger emergency mode via phone. This is still being figured out.
    - Video Recording: The correlation between mobile device connectivity and nonexistent because neither impact the other one.
    - Cost: The correlation between mobile device connectivity and is a weak negative because the parts needed do not seem to be too expensive.
  - Engineering vs Engineering:
    - Maintaining Following Distance: The correlation between the robot's speed control and autonomous following is a weak positive. For a more in-depth analysis of the cause of this correlation please the above section. It will describe the come of the pros and cons of the relationship and why the final verdict was reached. The entirety of the description that is alluded to is not included in this section for the sake of brevity.
    - Speed Control: The correlation between the robot's speed control and autonomous following is nonexistent. For a more in-depth analysis of the cause of this correlation please the above section. It will describe the come of the pros and cons of the relationship and why the final verdict was reached. The entirety of the description that is alluded to is not included in this section for the sake of brevity.
    - Operating Distance: The correlation between the robot's speed control and operating distance is nonexistent. For a more in-depth analysis of the cause of this correlation please the above section. It will describe the come of the pros and cons of the relationship and why the final verdict was reached. The entirety of the description that is alluded to is not included in this section for the sake of brevity.
    - Mobile Device Connectivity: The correlation between speed control and operating distance is a strong positive because requirement correlated to themselves will always have a strong positive correlation just do the nature of the way the terms is defined.
    - Video Recording: The correlation between speed control and video recording is nonexistent. The better or worse that either ones get it will not affect the others performance.

        - Cost: While this cost requirement is in the engineering category the same factors apply to is as the cost requirement in the marketing requirement above. For that reason, please see the above section for analysis.

- “Video Recording” is defined as reading and storing thirty minutes of video. During regular operations SAFER will continually record, overwriting the oldest media. When the emergency mode it triggered SAFER will record as long as possible without overwriting any of the old media. This will allow for a video recount of any incidence that SAFER may witness.
    - Marketing vs Engineering:
        - Autonomous Following: The correlation between video recording and autonomous following is a strong positive because of the overlap of some of the components needed. If we were to use computer vison then as the camera gets better so will the recording.
        - Phone Connectivity: The correlation between video recording and phone connectivity is nonexistent. This is because the phone or its connection will not improve the recording capability of SAFER.
        - Self-Battery Monitoring: The correlation between video recording and self-battery monitoring is nonexistence.
        - Assailant Deterrents: The correlation between video recording and is a strong positive because the recording will be a strong part of the assailant deterring system.
        - Video Recording: The correlation between video recording and video record is a strong positive because they are the same requirement under different categories
        - Cost: The correlation between video recording and cost is a weak negative because the cost of parts is not to large.
    - Engineering vs Engineering:
        - Maintaining Following Distance: The correlation between the robot's speed control and autonomous following nonexistent. For a more in-depth analysis of the cause of this correlation please the above section. It will describe the come of the pros and cons of the relationship and why the final verdict was reached. The entirety of the description that is alluded to is not included in this section for the sake of brevity.
        - Speed Control: The correlation between the robot's speed control and autonomous following is nonexistent. For a more in-depth analysis of the cause of this correlation please the above section. It will describe the come of the pros and cons of the relationship and

why the final verdict was reached. The entirety of the description that is alluded to is not included in this section for the sake of brevity.

        - Operating Distance: The correlation between the robot's speed control and autonomous following is nonexistent. For a more in-depth analysis of the cause of this correlation please the above section. It will describe the come of the pros and cons of the relationship and why the final verdict was reached. The entirety of the description that is alluded to is not included in this section for the sake of brevity.

        - Mobile Device Connectivity: The correlation between the robot's speed control and autonomous following is nonexistent. For a more in-depth analysis of the cause of this correlation please the above section. It will describe the come of the pros and cons of the relationship and why the final verdict was reached. The entirety of the description that is alluded to is not included in this section for the sake of brevity.

        - Video Recording: The correlation is a strong positive because requirement correlated to themselves will always have a strong positive correlation just do the nature of the way the terms is defined.

        - Cost: While this cost requirement is in the engineering category the same factors apply to is as the cost requirement in the marketing requirement above. For that reason, please see the above section for analysis.

- “Cost” is defined as less than $1000 dollars. It is important to remember that cost has a negative polarity.

    - Marketing vs Engineering:

        - Autonomous Following: The correlating between cost and autonomous following is a weak negative. The component needed for our design will not be of great cost.

        - Phone Connectivity: Following: The correlating between cost and phone connectivity is a weak negative. The component needed for our design will not be of great cost.

        - Self-Battery Monitoring: Following: The correlating between cost and self-battery monitoring is a weak negative. The component needed for our design will not be of great cost.

        - Assailant Deterrents: Following: The correlating between cost and assailant deterrents is a weak negative. The component needed for our design will not be of great cost.

        - Video Recording: Following: The correlating between cost and video recording is a weak negative. The component needed for our design will not be of great cost.

        - Cost: Following: The correlating between cost and cost is a strong positive because they are one and the same.

    - Engineering vs Engineering:

        - Maintaining Following Distance: The correlation between the robot's speed control and autonomous following is a weak negative. For a more in-depth analysis of the cause of this correlation please the above section. It will describe the come of the pros and cons of the relationship and why the final verdict was reached. The entirety of the description that is alluded to is not included in this section for the sake of brevity.

        - Speed Control: The correlation between the robot's speed control and autonomous following is a weak negative. For a more in-depth analysis of the cause of this correlation please the above section. It will describe the come of the pros and cons of the relationship and why the final verdict was reached. The entirety of the description that is alluded to is not included in this section for the sake of brevity.

        - Operating Distance: The correlation between the robot's speed control and autonomous following is a strong negative. For a more in-depth analysis of the cause of this correlation please the above section. It will describe the come of the pros and cons of the relationship and why the final verdict was reached. The entirety of the description that is alluded to is not included in this section for the sake of brevity.

        - Mobile Device Connectivity: The correlation between the robot's speed control and autonomous following is a weak negative. For a more in-depth analysis of the cause of this correlation please the above section. It will describe the come of the pros and cons of the relationship and why the final verdict was reached. The entirety of the description that is alluded to is not included in this section for the sake of brevity.

        - Video Recording: The correlation between the robot's speed control and autonomous following is a strong negative. For a more in-depth analysis of the cause of this correlation please the above section. It will describe the come of the pros and cons of the relationship and why the final verdict was reached. The entirety of the description that is alluded to is not included in this section for the sake of brevity.

        - Cost: While this cost requirement is in the engineering category the same factors apply to is as the cost requirement in the marketing requirement above. For that reason, please see the above section for analysis.

3 Research

Research is critical to understanding both the basics of various components as well as learning how to integrate systems together. Having background information helps in deciding which technologies will be best to use, what others have used in the past, and what mistakes to avoid that others have already gone through. Below is the breakdown into the groups research on past relevant projects, general research done, relevant technologies found and looked into, and a comparison of different parts as well as the selection of parts that best fit the project's needs. Research had to be done mainly in the areas of object detection, various sensors, speakers, lights, microcontrollers, and vehicle bases. After research has been done, the next step will be to purchase relevant components that can then be used to build and test the project's design.

3.1 Similar Past Projects

Engineering is best done by reviewing past projects and relevant information and applying it towards a new goal. Therefore, to create the SAFER Knights robot, research was done to look into both past Senior Design projects as well as other robots that achieved similar objectives to help figure out the best way to implement our project.

3.1.1 ABEC (Autonomous Brilliantly Engineered Cooler)

Figure: ABEC Vehicle
[Permission Pending]

This Senior Design project from fall 2012 held many similarities with our project, so much research was done to see what we could learn from them, things that were a good idea, and what to improve upon. The point of their project was to have an autonomous cooler for game days to relieve fans from hauling around their own coolers filled with drinks and food, as well as being solar powered. Most notably, they also used a Power Wheels vehicle as the base of their robot, as well as it being an autonomous vehicle meant to follow a user. The robot used Bluetooth via phone connection to connect to users through a mobile app. For object detection, it had ultrasonic sensors attached to it. Overall, this project holds many of the same components and methods that can be used for our own projects. However, the ultrasonic sensors have been nixed in favor of a Microsoft Kinect, so there will be some major differences between the two projects.

3.1.2 Tailgate Buddy

Another Senior Design project from fall 2008, it is not found up on the Senior Design website so the only information that could be gleaned about it came from ABEC's documentation about previous projects. Similar to ABEC's project, it is also a Power Wheels with a mounted cooler that is autonomous but not solar powered. Since ABEC based much of their design on this previous project, it still proves a beneficial resource for us as well.

3.1.3 "Follow Me" Cooler from Hacker Shack

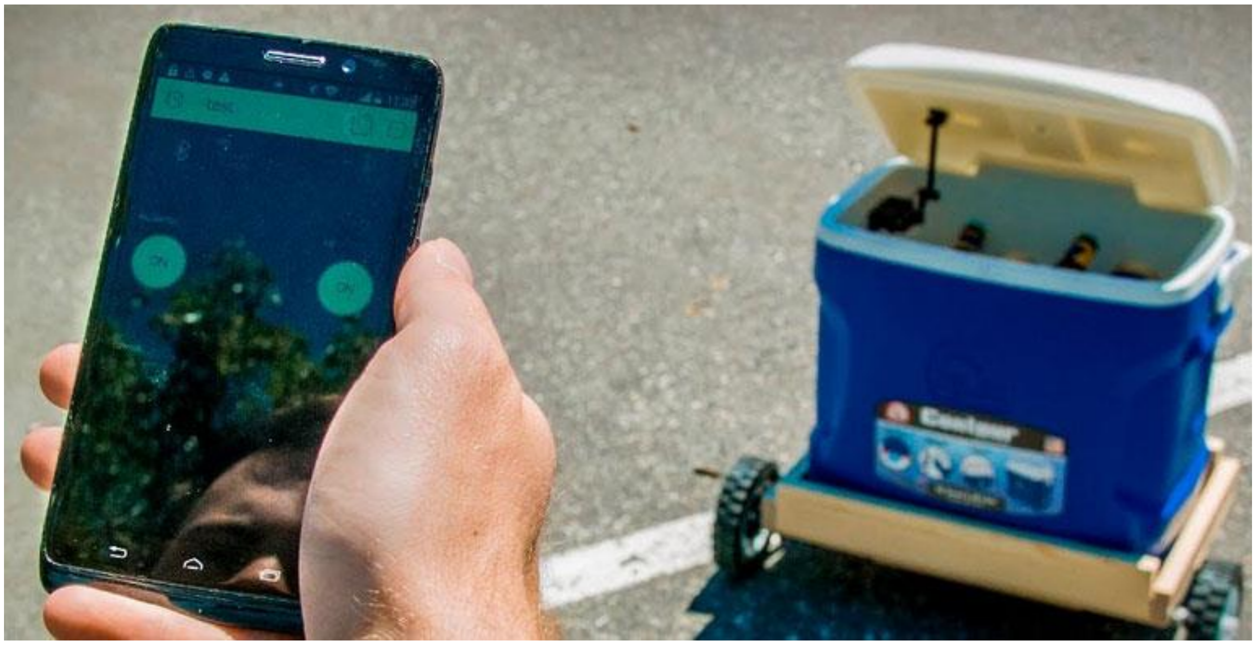


Figure: "Follow Me" Cooler from Hack Shack
[Permission Pending]

Searching online, an autonomous user following robot was found online by Hacker Shack. They included a complete list of parts, instructions, and demonstrations on how to build a robot that follows a user's phone via Bluetooth and uses GPS to navigate. As this is the method we have chosen to get our robot to follow a user, it proved to be a very useful guide. It did not, however, include any sort of sensors to prevent the robot from running into obstacles in its way, while our vehicle will need to operate more successfully.

3.1.4 FollowBot

The FollowBot is a Senior Design Project from summer 2017 and uses Bluetooth to have a robot follow behind a user in an airport carrying their luggage. The user connects via app, and ultrasonic sensors are used to avoid obstacles especially in a highly trafficked area such as the airport. This project shows many similarities to ours while attaining different goals, so it was looked into for ideas and lessons learned.

3.2 General Research and Relevant Technologies

This section introduces the basics and background of various research and technologies looked into to achieve the goals of this project. It includes a discussion on the mechanics of the robot following the user, ways in which the robot can avoid possible interfering objects in its environment, and a discussion on some of the relevant technologies that pertain to the project's mission.

3.2.1 Following Mechanics

The SAFER Knights project required some way to track the person it was supposed to be following. The following research is based on the average specification for the following sensor options: Bluetooth, infrared, lidar, and ultrasonic. Factors such as the frequency range, signal distance and current draw were used to determine which option would be optimal. The table below shows a clear comparison between the tracking options mentioned:

Table: Tracking options comparison

	Bluetooth	Infrared	Ultrasonic	Lidar
Frequency range	2402-2480 Or 2400-2483.5M Hz	430 THz - 300 GHz	40k Hz	50-200.0 k Hz
Distance (m)	100.0	1-5.0	20.0	50.0
Transfer speed (Mbps)	25.0	1000.0		
Current draw (mA)	30 .0	10-20.0	15	100

The following mechanics and object detection of SAFER could both be solved by similar technologies, but with different techniques. They are distinct enough to both warrant their own section. SAFER will need to be able to follow the user to their destination. The goal of these components will be to have the robot follow the user within the requirements such as maintaining a following distance of 5-7 feet. The sensor options above and more were researched below:

3.2.1.1 Infrared

To use infrared as a tracking mechanism we will need to have three kinds of components. The first is an infrared beacon where the user will emit an infrared signal of a certain frequency. The second kind of component needed will be infrared sensors, these will be placed on the SAFER to be able to receive the signal emitted from the infrared beacon worn by the user. If this solution is pursued, there would need to be multiple infrared sensors placed on SAFER. This requirement is needed to be a second dimension to the robot's "vision". If there were to be only one infrared sensor then the robot would not be able to tell the direction of the user, only if they are emitting infrared in such a way it is not blocked from the robot. This would only allow the robot to move in a binary fashion, moving or not moving depending on the user beacon's visibility. If we are to add multiple infrared sensors to SAFER, placed on the front end facing its direction of motion, it will allow SAFER to know if the user is to the left or right of itself. For example, if there were to be two infrared sensors they would each have their own vision cone like the figure below. Each infrared sensor will be able to register if the beacon is in its cone. If the infrared beacon is in both cones, that means that the user is in front of SAFER. The use of more infrared sensors will add to this overlapping effect and create better precision as to the locations of the beacon in relation to SAFER.

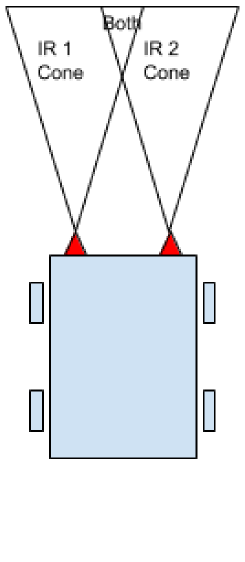


Figure: Robot with Infrared Sensors

This point brings us to the last component, a compute module. SAFER would require that these infrared sensors would send their output to some sort of computer hardware to integrate the meaning of the signals. This module would take in the signals from the infrared sensors and determine the placement of the beacon with relation to the front of the robot. It would be the device determining if the beacon is in zero, one, or more of the infrared sensors' cones of vision. Therefore, this computer will need the ability to have pin connections for infrared sensors to send their signal. This compute module can be in any form from a small single chip device to a more traditional architecture. Ideally the compute module would be a small single chip computer like a Raspberry Pi or a microcontroller like the Arduino. This smaller design would save power consumption and space on the robot. If this alone required a computer, the choice of a smaller computer would be easy, but there are two possibilities of more components to be added that may push the amount of processing power over the capabilities of some of the smaller single chip computers. The move to a larger design even to traditional laptop scale would still work with this system.

Thinking about the beacon again now that we have talked about the rest of the system, this device will have to be worn in a location where the infrared signal being transmitted is visible to the SAFER robot. This presents a difficult problem for the user of the beacon. The user would have to hold or wear the beacon in such a way that is behind his or her walking direction. This will also mean if the user turns a corner or is out of direct line of sight, SAFER will not be able to see his or her location. This idea abstracts out to any object or motion that can block line of sight, that is if the sensor was drawing a straight line from it to the beacon, no object would collide with that line, from SAFER would cause it to lose signal. The reason for signal loss could be hard for the robot to determine with the use of its infrared sensors alone. From now on in this section, we will call the event of the infrared beacon not being seen by one or more of the infrared sensors mounted on SAFER a "loss of signal." Some of the loss of signals can lead to some logical steps that SAFER could take to reacquire signal. Inferring two infrared sensors, placed in a way similar to the figure above, here is how SAFER would follow the user using its computer:

- If the beacon is placed in such a way that both of the infrared sensors can detect its presence, then the robot can just move forward because the beacon is directly in front of the SAFER. This is the most desired state.
- If the beacon is placed in such a way that only one of the sensors can detect its presence the robot can move in such a way that it tries to have both sensors try to detect the beacon. This would require steering. If only the left sensor can detect the beacon, then the robot would want to steer left to move the sensor closer to its center, once again meaning that the user is directly in front of the robot. Similar steps can be taken if only the right infrared sensor can detect the beacon.
- In the case of the signal loss, SAFER can make some educated guesses as to how to reacquire the signal emitting from the beacon. If only the left-most infrared sensor can detect the presence of the beacon, and then none of the infrared sensors can detect the presence of the beacon, SAFER can infer that the user has moved more left than the infrared sensor's cone of vision can detect. The robot

can infer that it should still steer more left. The same method can be taken to the right infrared sensor.

These ideas could be abstracted to more than two sensors to create a more granular idea of the user's position. This being said, there are some situations where SAFER could have trouble. If, during the course of operation, SAFER were to lose signal due to an object between the user and it, SAFER might think that the user has turned a corner or has taken a sharp turn left or right. This also brings us to back to the point about the infrared sensors only having line of sight detection capabilities. Below is a table with some high-level pros and cons to using a system like the one described.

Table: Infrared Pros and Cons

Infrared Following Pros	Infrared Following Cons
Low amount of computations to utilize inputs for following logic	Low location accuracy with low numbers of sensors.
Low power usage and compute power needed	Sensitive to light conditions, daylight in particular
Simple component with well-established performance histories	Requires user to wield a beacon for the robot to detect
Low cost components	Following distance would require extra hardware
Operational at night	

3.2.1.2 Computer Vision

Computer Vision is another option for following the user. If we use this option, it would solve some of the problems presented by the infrared option but add some of its own obstacles. The use of computer vision would require two modules. The first module would be some sort of camera used to see the environment. This camera’s image quality can vary depending on application, but for our application it can be less than HD quality. This camera would send its image signal to some sort of computer. The processing power of the computer is more relevant than in the infrared sensor option for following. The processing power needed for computer vision can be significant, especially when the quality of the image is increased to HD (1280 x 720 or more). This is because the computer will have to iterate over each of the pixels for each frame at least one time and sometimes more, possibility also complex, meta calculations that will be used to make following decision. This means that the more pixels in the image, the more computational power required. Luckily, to get decent reliability, the camera sensor does not have to be HD quality. The video quality will only need to be within the range of 240P to 480P to get the amount of information we would need from it. This will mean the onboard computer power would be above an Arduino but would be doable by a Raspberry Pi or greater. We would like to have the smallest computer possible to save on power consumption and size requirements.

There are computer vision solutions that combine the camera and computation needed into one chip. One example of these types of solutions are called PixyCam. A PixyCam will have a programmable API that will be able to send signals to a main computer system that can have the compute power of an Arduino or Pi. This can be done by building custom hardware on to their chips. This would most likely be the best option for our applications due to the robustness of the camera's API and ability to connect to a lower power central computer board.

Some of the advantages of the computer vision are that it offers the ability to track the user to a more granular level while they are in the field of vision. Instead of the robot only knowing if the user is to the left, in the middle, or to the right, the following logic will be similar to the one described in the infrared section while also being able to know to what magnitude the user is to what side and adjust the steering in a non-binary fashion. Another advantage is the ease that object detection and avoidance will integrate into the SAFER without any extra hardware, more on that in a separate section.

Some of the disadvantages of the computer vision option is the difficulty of identifying who to follow. The robot will have difficulty classifying what the user looks like and following him or her exclusively. Using computer vision, the robot will not have a simple color or pattern it needs to follow and track, it will have a three-dimensional human it will need to track. This sort of tracking is difficult in computer vision because it is not one pattern trying to be matched because human looks different as it is walking from frame to frame and from different angles. Even if the robot has that capability it will be hard to distinguish user from other people. The robot could only look for the color of the user's shirt and follow that color, but that would present other problems for false positives on that color on other people or objects. This is one of the pitfalls of computer vision. It is extremely good at pattern recognition, but hard to abstract out to other non-static applications. There are other concerns mentioned above also about power consumption and computational power requirements.

3.2.1.3 Wireless Signals Strength

One more novel way to think about following is to track the signal strength of a wireless signal. The two wireless signals in question could be WIFI or Bluetooth. Bluetooth being the most probable for our use. This application will require 2 or 3 modules depending on your granularity. The first is a Bluetooth enabled device such as a cell phone. The smartphone will be the type of device we would use for our application. The smartphone will connect to component 2/3 which is a Bluetooth sensor, magnetic compass, and possibly a GPS connected sort of central computer. Once these two devices are connected using the Bluetooth protocol, they will share information about the connection. One data point shared between the two devices is the received signal strength indicator or RSSI. The received signal strength indicator will tell each of the devices how powerful the signal of the other device is at its given location. So, in our case, if SAFER were to stay still and the user's connected smartphone were to move farther away, the received signal strength indicator of SAFER would go down. As we discuss this option more when we mention the received signal strength indicator, we will be talking about the received signal strength indicator of SAFER. The received signal strength indicator of the smartphone is irrelevant to our application.

One of the challenges of using the received signal strength indicator is that it is not a vector. The received signal strength indicator is only the scalar part of what we need to know about the direction and distance of the user. To find this distance we will need to sample the received signal strength indicator at two different locations. To do this we will need to move in the most educated guess of a direction of the user based on possible previous data. This will give us the two points with two corresponding received signal strength indicators. To know the difference in locations we can use the compass and infer distance traveled or use the GPS but that will introduce sampling problems in the next discussions. We can take these two locations and derive a gradient vector. This gradient vector will aim to go up the steepest path of the wireless signal. Imagine a stereotype signal strength of a wireless signal without any interference. The gradience vector will point to the device emitting the signal. This is the same concept with SAFER's received signal strength indicator algorithm. It will collect the two points to find that gradient vector to follow the user. The face that user is possibly also moving can present a problem for long sampling durations. As we sample the received signal strength indicator more often the problem will be reduced.

Some of the advantages of this technique is in the simplicity of the required hardware. The only needs are two devices that most people already have and are easy to buy, but that is just about where this technique stops sounding good.

The disadvantages of the technique are mostly due to noise and margins of error. This method is not the intended use of the received signal strength indicator. The received signal strength indicator is intended to communicate if the connection is strong enough to reliably transmit data. This application is just a hacked way of using the received signal strength indicator. The received signal strength indicator alone will vary from device to device on both ends. The smartphone and the SAFER's emitter will produce inconsistent absolute values of the received signal strength indicator. There is also a very real effect of noise on the received signal strength indicator. This makes the data often less reliable.

3.2.1.4 LIDAR

Lidar can go by many names. There is LIDAR, LiDAR and LADAR. For our discussion we will use LIDAR. LIDAR stands for light detection and ranging. Lidar is becoming more and more popular currently because autonomous vehicles are becoming more and more popular. They are particularly popular for driverless cars. LIDAR has many uses, most popular of which is object avoidance. We will discuss our possible utilization of LIDAR for that use in a different section. In this section will focus on possible applications of LIDAR for following the user. LIDAR sensors are very precise at measuring the distance between the sensor and the closest object in front of it. The sensor can be on a fixed position of SAFER offering a single point of measurement. The LIDAR sensor can also be rotating to offer several points of measurement. This rotating LIDAR is the more desired of the two configurations. The rotating LIDAR would require some mechanics for rotating the sensor in a measured way so that a computer can take the rotation information along with the LIDAR data and know what way the sensor is facing while it took that measurement.

The information coming from the LIDAR, as said before, can be very good for object avoidance, but may be hard to use for following the user. The information coming back is only range data and it will be hard or next to impossible to distinguish the user that SAFER is trying to follow and any other object it is sensing. It would be possible for SAFER to use this data to follow the user.

Below is a visualization of what the LIDAR sensor may see. The dots represent distance measurement from the center where SAFER is. Thinking of a square room without any obstacles, SAFER would be able to detect mostly a square. The two dots that are out of place in the square are points in which the robot has detected that an object is closer than the other dots in its plane. This would have to be the user that SAFER it trying to follow. This effect would be the same as the user moves around the room. The group of protruding dots would always be the user in an empty room. This would make it possible for SAFER to track the user.

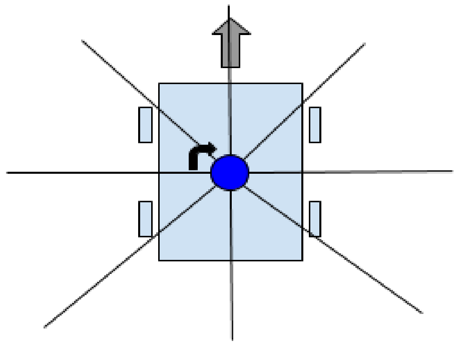


Figure: Robot with Lidar Sensor

Now think of a room that is not uniform or has obstacles in it such that the shape is irregular. While the user is stationary the robot would not be able to distinguish them from the other irregularly of the environment, but if the user started to move around, SAFER would be able to detect that movement and identify it as the user. The robot is now able to follow the user again. As soon as the robot starts to move the environment starts to change and it is no longer able to detect movement.

Finally, think back to the original figure where the room was empty, and the user was a group of measurements closest to the robot. If we take out the walls and introduce an irregular environment we are back to situation 2. If the robot only focused on that group of dots that represented the user and always tries to keep them within the proper following distance it can now ignore the surrounding environment. This would create a foreground/background effect. Where the robot can only focus on keeping the foreground separate form the background and thus the user closer than the other objects round it. This method would work with two strike condition. The user would have to be known to the robot initially and the user can never become indistinguishable from the environment. These two conditions would be hard to always satisfy. The breaking of these conditions can also create following system failure.

Another disadvantage of this system would be similar to the problem in computer vision where the user would be hard to distinguish from other people in the its line of sight. This problem would only be made worse in the case of LIDAR because it does not have color information or any other way to distinguish the other people form the user. These points would make it hard to use LIDAR as the primary sensor for the following mechanic.

3.2.1.5 GPS

As most technical people know, GPS stands for Global Positioning System. GPS is commonly built into a lot of devices in the current era. GPS has also improved its accuracy ability to about 3 to 7 meters depending on the sensor. In one possible use of GPS one SAFER is to use the GPS information from a smartphone and send it via Bluetooth to the SAFER robot. As mentioned the first component of this system would be a smartphone with GPS and Bluetooth capability. The cell phone talked to cell tower of a known location and based on signal travel time, signal strength and other factors calculated its location. The more cell coverage in an area the better the accuracy of the system is. The phone now knowing its location use a wireless communication standard such as Bluetooth to communicate to SAFER its location. The smartphone would continuously send the stream of GPS coordinates to SAFER as time progresses. Obviously, the other component of this system would be a computer module with Bluetooth and GPS capability. This module would take in the stream of GPS data form the smartphone and use it to find the path to follow. The SAFER GPS would work a bit differently than the smartphones would. The GPS from SAFER would communicate to satellites in known orbits around Earth and calculate its location based on travel time of signals back and forth. The accuracy of both methods is comparable under the best conditions for both.

SAFER knowing the GPS coordinates of the user and at the time of these measurements the robot would be able to mimic the path of the user. This would let SAFER know the path of the user and its approximate distance from the user. The robot can then determine its steering based one the location of the next GPS coordinate. The robot would be able to know the speed of the user and its distance from then to be able to calculate the speed at which it should be traveling to maintain the proper following distance from the user. This method would have to deal with margins accuracy in the GPS data. We predicted these margins would not be large enough to drastically affect the viability of this method.

This method does not include any sort of object avoidance. Some of the other methods for following the user can also be used or object detection, but this system is not capable of that functionally. This means that another system would have to be used alongside this system. More on that in another section.

3.2.1.6 Sensor Fusion

In relation to the SAFER need for object avoidance, some of the methods for following would of also tied into the ability of SAFER to avoid objects would any extra needed hardware. Also, to that point some of these methods for following are better suited to object avoidance. For these reasons there would of most likely been a mixture of these methods integrated into SAFER. Here is a table of the advantages and disadvantages of the methods discussed above.

Table: Sensor Comparison

Method	Infrared	Wireless Signal Strength	Computer Vision	LIDAR	GPS
Pros	- Low noise - Cheap - Established	- Build in to standards -Low Compute Needs	- Flexible	- Accuracy	- Simplicity
Cons	- Extra hand-held device - Medium Location Accuracy	- Extremely noisy - Low accuracy	-Large compute need -Lots of information	- Lack of flexibility	- Precision

3.2.1.7 Conclusion

Infrared provides a very clean and simple solution to following but adds a needed beacon and has more outdated design. Wireless signal strength is not a good enough method to be the primary method, but it can be an added secondary option. Computer Vision is a very viable solution but requires large compute power and lots of ambiguity. LIDAR would present a lot of challenges if it were to be used as the main method of following. GPS presents a clean simple solution to following similar to infrared as long as coordinate accuracy is good enough, which unfortunately it wasn’t. Our plan was to use GPS but we ended up using a Pixy cam and computer vison.

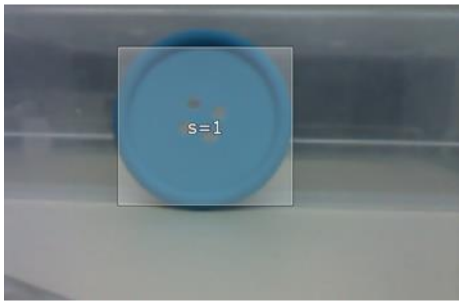


Figure. Color Tracking using the PixyCam

In the end, we chose to use computer vision to follow the user. The GPS and compass method ended up not being accurate enough to follow the user. Using a PixyCam, a Raspberry Pi was used to handle the video processing. The PixyCam is a useful fast vision sensor that is made for computer vision. It comes ready out of the box able to sense objects by their colors or bar codes. For simplicity's sake, color was chosen. Initial testing of the camera showed that learning an object by color can lead to false positives if objects of the same color are within the cameras field of vision. Luckily, the PixyCam has the ability to create color codes, which involves using two or more color tags close together. That way the camera will only register the desired object to be followed when the correct color combination is seen, such as orange and green.

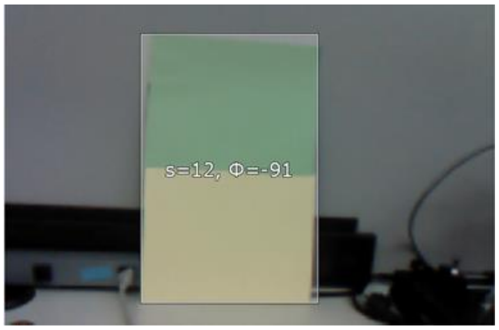


Fig. 3. Yellow and Green Color Code using PixyCam

To follow the user, the camera has to be able to know when the user is left, right, and center. To do this, the user will wear a color code board, which will come with the robot, on their back. The PixyCam is also able to determine the dimensions of an object that it is tracking and its X and Y coordinates in pixels. Since the board stays a constant size, its current dimensions can be measured and used to determine distance. The farther away the person is, the smaller the board will seem, and the robot will know that the user is far away. The code will also know how far left or right the color code is from center, so the robot knows when to turn and when to stay centered.

3.2.2 Object Avoidance

During the course of operation SAFER did need to avoid object. These objects would include stationary things like walls and trash cans and other objects that are moving such as other people or tumbleweeds. Below are some viable methods for object evidence that are variable methods to implement on SAFER.

3.2.2.1 Infrared

Similar to many other methods that can be used for SAFER's following mechanic so can Infrared sensors. To use infrared for object avoidance would work similarly to how the following infrared sensors would work. This may cause some issues if both are used, but that will be discussed in a little while. In the following Infrared sensor method there are modules. One module was the infrared beacon emitting infrared signals from the user's location, and the infrared sensor(s) detecting that infrared light. In this application both of those modules will be mounted onto SAFER. There will need to be a minimum of two infrared sensors, one for the left front and one for the right front. There will also need to be at least one central infrared emitter or more likely one infrared emitter per infrared sensor. These infrared emitters will send out infrared waves in the direction they are pointing. If there is an object in these infrared waves path they will bounce off this object back in the direction of SAFER. Then the infrared sensors will detect the infrared light coming towards the robot. This would tell the computer module that there is an object possibly in the path of SAFER. The closer an object comes to this infrared array, the more of the infrared will be reflected back to the infrared sensor. This will tell SAFER an estimate of the distance from the object. Like mentioned before, this method is most commonly used with an infrared emitter per infrared sensor. If this method is to be used, we would like to explore the possibility of having a single infrared emitter in the center of the robot and letting that broadcast out to all infrared sensors. This might be possible if there is an infrared light emission cone wide enough but not too wide as to automatically hit the infrared sensors. Thus, this might be in folly but I would like to see the results of the experiment. Another advantage of this system is the ability to operate at night or in dark conditions.

Below is a diagram of how the infrared emitter and infrared sensor array might detect an object. The red represents the infrared emitter and the infrared light being emitted. The blue represents the infrared being reflected by the object in the path of SAFER. The blue triangle is also the infrared sensor. As you can see, the closer an object is to SAFER, the more infrared will be detected by the infrared sensor, and if no object is in the path of SAFER no or very little infrared will be reflected back.

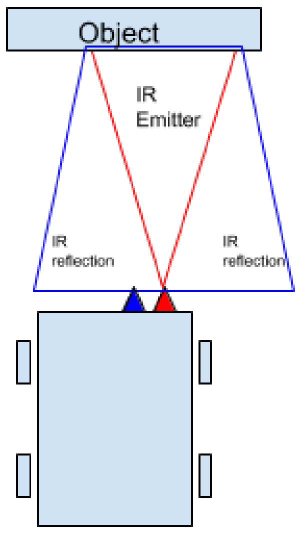


Figure: Robot with IR emitter

As alluded to in the start of this section, there are some drawbacks to using infrared for object avoidance. The first of which is the possible cross talk of using infrared to both follow the user and infrared to avoid objects. The frequency of light used for both systems would have to be different enough such that SAFER cannot mistake an object in its path as a user, or a user as an object in its path it has to avoid. Without experimentation it is unknown by us if this is possible. Another downfall of this method has to do with distance measuring. The distances able to be measured are often in the centimeter scale. In our application we will need to measure on the feet scale due to the size and turning radius of SAFER. There also can be calibration needs for different environments for the infrared object detection array. Branching of this point operation during the day can be severely hindered by the infrared light being emitted by the Sun. This may have a particularly bad impact on our application. Finally, the distance prediction of this infrared array is not very precise.

3.2.2.2 Computer Vision

Computer Vision is widely used in object avoidance. In our research we learned of an ongoing competition of sorts between autonomous car manufacturers between the need of computer vision only or computer vision paired with LIDAR for robust object detection and object avoidance. The advantages the LIDAR may add are discussed in the section labeled LIDAR in the object avoidance section. In this current section, only computer vision will be discussed.

Computer vision can be a very robust method for computers to sense the environment around them. It has many applications in object detection and classification. The application of computer vision to object avoidance is very common practice also. As mentioned in the section about computer vision in SAFER following mechanic, computer vision can be very computationally expensive. This may be solved by using things like the PixyCam that has purposely built hardware attached to its camera to offload some of the computation. Another possible obstacle that we may encounter with computer vision is the difficulty to actually measuring distance to an object without knowing its size. This can be counteracted by some algorithmic methods but is not a trivial problem to solve.

3.2.2.3 LIDAR

As discussed in following section, LIDAR is light detection and ranging. There are many ways LIDAR can work internally, but those details do not affect its performance in our applications. There are also many was that LIDAR can be used to model one-dimensional, two-dimensional, and three-dimensional environments. To get these three kinds of models there are corresponding LIDAR systems that can be purchased. The more dimensions added, the more the cost goes up in almost a multiplicative manner.

In our application we could use a LIDAR to find objects in SAFERS path that SAFER needs to avoid in order not to collide. Using a one-dimensional LIDAR system would not give us enough information about our surrounding environment. The only information we would know is the distance on a single object in front of SAFER. Two-dimensional and

Two-dimensional LIDAR are the most desirable systems. Three-dimensional systems are a bit overkill for our applications, so we would strive for a Two-dimensional system. Two-dimensional LIDAR are possibility over the budget for this project. Is it possible to purchase a simple LIDAR sensor and build a small device to create a two-dimensional LIDAR system?

If we were to buy a simple LIDAR sensor and convert it into a two-dimension LIDAR by attaching it to a servo or a stepper motor via gears or a band. This will give the ability to detect the distance of any objects on the same plane as the sensor. Now having the ability to see any object in the same plane as the LIDAR sensor we can develop a system to find the exact location in relation to SAFER. If we were to know the angle from an origin point that each LIDAR measurement was taken, that is the distance to the object in front of it, we can then calculate the distance to objects in front of SAFER instead of a single point in front of it. Below is a pictorial representation of how SAFER might see its environment by rotating the LIDAR sensor. The lines represent the reading line that the LIDAR will take its reading on. The arrow represents the direction of the sensors rotation.

To give the LIDAR sensor this 360-degree motion we would have to design a system that will enable this. The three components needed would be:

1. A way to rotate the sensor in some way. The three best options would include a servo, DC brushless motor or a stepper motor. The main constraint on this on the selection of right component would come from the need to know the angle of the sensor while it takes the LIDAR measurement. To accurately know the location of the object being measured SAFER will need to know direction the sensor is face while it took the measurement so that the robot will know if that object is in front of it. The means the DC brushless motor cannot be used. It is not capable of knowing its position while rotating. These leave us with the servo or the stepper motor. While both of these options would work the servo is slower and noisier. The LIDAR sensor is able to take reading in a very rapid manner. This means the sensor can rotate quickly and still get all of the readings needed. For these reasons the stepper motor would be the best option
2. The second component is as way to allow the sensor to continually rotate while not tangling the wires going out of the sensor and into its power source or compute module. A slip ring should allow for this.
3. The last component needed is the actual hardware to mount these pieces to get the rotation effect we need to sense more than a single point. This could be done many ways but thinking about is simply the best way that comes to mind is a simple design similar to a bicycle chain. The driving motor, most likely a stepper motor, will be attached to a gear. Kind of like where the bicycle pedals are attached. This gear will be meshed with another gear to hold the sensor. This will mean that when the motor rotates so will the sensor. There can be any number of gear ratios for this system depending on the needs for each component, but a 1 to 1 ratio would make calculations a bit more simple. Instead of the large gear there could also be smaller gears similar again to have a bicycle works with a chain connection them.

More cogs that gears. The last modification would be to drop the teach form each cog and have a belt connect these two components.

Now that the sensor can rotate in a 360-degree manner we can detect object all around SAFER, although most of the time only objects in the front of SAFER are relevant to its object avoidance. The latency or time relevance of the reading will be very good considering the expected revolutions per minute that the entire sensor ridge should be able to support. This is also compounded by the fact that SAFER will not be moving very fast, only around 5 miles per hour. This means what we can loosely call the image resolution will be very high. Another advantage of this system is that it will require very little computation power from the central processing unit. This is especially and advantage over other option such as computer vision. The image fidelity is also very precise. In other applications it is hard to tell the distance or size of an object but with this method it is very accurate and easy to tell precisely where the object is in relation to the robot.

There are some restrictions to this method still. The sensor while rotating around will be able to see with great precision on a single plane. That is the only objects that will be sensed will be on the plane in which the sensor is sending out its range detection light. This will mean for some objects that have to potential to collide with SAFER will be invisible by this LIDAR system. There are three edge cases of this sort. One would be an object like short poll, stump for dead body in SAFER’s path. If the position of the sensor would be above the object and level to the ground as it rotates it would go undetected. The easy fix to avoid this would be to move the sensor lower thinking that all object we want to avoid will be placed or originating from the ground in some way. This method could work but would making detecting objects like humans where their base is a lot smaller compared to their main mass. Humans would be a major category of object SAFER would want to avoid. Another reason against a lower placement of the LIDAR sensor would be the second edge case. As with the first edge case SAFER is be unable to see object outside of the plane on which the LIDAR sensor rotates around. This means that object that may hand in the path of SAFER with the potential of collision will not be detected. Placing the sensor at the highest point would only exacerbate the first possible more common edge case. This would mean, for the three reasons mentioned above, that the sensor would be better placed high up at somewhere between 3 and 5 feet. The last edge case is not exactly related to object avoidance but is noteworthy. In an over exaggerated example: If SAFER were to approach a cliff or some large drop off, it would be unable to detect the steep drop off, because the LIDAR sensor is not monitoring that plane. This would mean the robot could unknowingly meet its demise while simultaneously failing the user. Most of these problems could be solved with a 3-dimensional LIDAR system, but as said before those are most definitely out of our budget range.

In conclusion a two-dimensional LIDAR system could be a very good method for object avoidance for SAFER. It can come at a high cost financial and would probably not be multipurpose in respect to the following mechanic. Still this may be a useful method for object avoidance.

3.2.2.4 Ultrasonic

This method of object avoidance would work in a similar fashion to infrared. One of the advantages of using infrared as the light medium is the last of its abundance in the natural environment. There is definitely infrared present in the natural environment, but we can produce and hewn in on one particular frequency. This can create a situation of minimal light noise compared to other wavelengths. The same idea exists with ultrasonic object avoidance. The vast majority of sound energy in the natural environment will not be in the ultrasonic range of above 20,000 Hz or cycles per second. This means if SAFER produces a known frequency of ultrasonic sounds in a directed manner it can listen for those sound waves being bounced back. To help image this concept please refer to the diagram below. The red indicates the ultrasonic emitter. These are sound waves above 20,000 Hz. The blue indicates the ultrasonic reflection of the object in the path of SAFER. The intensity of these ultrasonic waves being reflected back onto SAFER will indicate the distance from the object. The more intense and abundant the ultrasonic waves being reflected back will indicate that the object is closer to SAFER. If there is no object, the blue parts or the ultrasonic reflection will be very small.

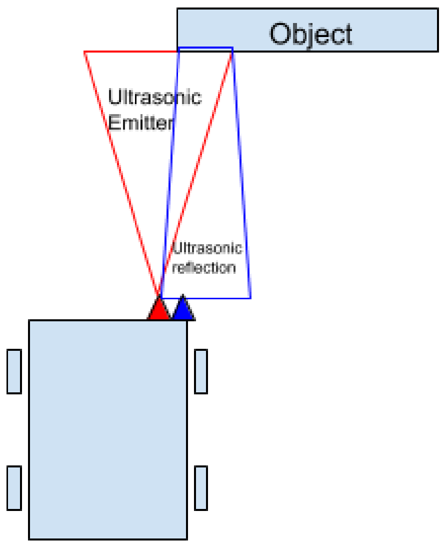


Figure: Robot with Ultrasonic Emitter

The advantage to ultrasonic object avoidance is substantial. The cost is very low for this method of object avoidance. The parts needed can be found for under 20 dollars and some as low as 5 dollars. The emitters and sensor are usually manufactured as one piece. This will cut down on the need to make our own rig similar to the case made for LIDAR. There is also a low computational need and power consumption. This would let us keep a smaller CPU and maintain battery life of SAFER.

Some of the disadvantages to this method cut the advantages slightly. The biggest disadvantage is the granularity of object detection. With this method it is hard for SAFER to know where exactly an object is located and at what distance. This can be improved by adding more of these sensors, but that did increase cost affect for other factors. This means the robot did know an object is in front of it but wcouldill now know exactly where that object is. The distance detection can also be less than accurate depending of the environment. This can be helped by calibration, but the robot will be used in a variety of different environments, reintroducing the problem. Ultrasonic can also be hard to use in loud and noisy environments.

3.2.2.5 Conclusion

For SAFER this requirement was not needed after the redesign of the robot

Table: Sensor Method Comparison

Method	Infrared	Computer Vision	LIDAR	Ultrasonic
Pros	- Low noise - Cheap - Established	- Flexible - Locality	- Extremely accurate - Locality	- Low noise - Cheap - Established
Cons	- Hard to Use for Both Follow and object avoidance - Medium Accuracy - No Locality	- Large compute need - Lots of information to parse	- Expensive - Extra build involved	- Medium Accuracy - No Locality - Medium Accuracy

3.2.3 Relevant Technologies

Below is a discussion of relevant technologies found that could be used for the project. After looking into them, they are pitted against each other to compare and contrast the pros and cons of each technology in comparison to the others. Afterwards, the ideal one is chosen for its superior performance, cheaper cost, ease of use, or a combination of factors.

3.2.3.1 Sensors

Multiple sensors were researched in regard to obstruction detection and avoidance. During the research certain factors were taken into consideration when determining the best option for our situation, some of these factors were cost, availability, range, limitations, and accuracy.

3.2.3.2 LiDAR

LiDAR technology typically utilizes an infrared laser spread to determine different aspects of the environment its used in. The LiDAR sensors fires lasers in succession and determines the amount of time the lasers take to return to the sensor. Using this data, the LiDAR sensor can determine the distance between the sensor and the object. LiDAR also creates a point cloud or a three-dimensional map of its environment, allowing not only distance but object detection. This sensor could allow for the determination of the size of an object as well a distance from the object. This had a great potential for us since we needed to be able to detect objects as well as their distance to determine obstacle avoidance. This allowed for a single sensor to be used for object detection, and distance measuring. Many models of LiDARs can be found online with a price range of less than $100.00 to more than $6,000.00, though it was noted that the cheaper the LiDAR the more it was reported to be of lesser quality and the more difficult it was to work with.

3.2.3.3 Ultrasonic

Ultrasonic technology uses high frequency sound waves to determine details about the area it is used in. Using these sound waves ultrasonic can calculate distance by determining how long it takes for the sound waves to return to the sensor. Basic ultrasonic sensors are not able to detect an objects dimensions like other sensors can. These aspects are important to the success of the project as both distance and object size is desired for accurate object avoidance. Ultrasonic is also not affected by ambient light, since our project is designed to be used at night having no reliance on other light sources is also important. Ultrasonic can be affected by softer objects that are better at absorbing sound which is an important detail. Ultrasonic sensors also have a useful amount of information on implementation that can be used as reference. Ultrasonic sensors can be found online for a relatively lower price range from $2.50 to $12.57.

3.2.3.4 Infrared Distance Sensor

Infrared distance sensors emit an infrared beam of light to determine distance. By measuring the amount of time it takes for the infrared beam to return to the receiver on an infrared sensor the distance between the sensor and object can then be calculated. Infrared distance sensors do come with some limitations. Infrared sensors are affected by the lack of light and color of an object. Since the project is being designed to work at night the limitation of light on an infrared sensor is an inhibiting factor. Further an infrared sensor does not work as well on darker objects. Since the color of an obstructing object is not controllable this would be another hinderance. The most prominent problem with the infrared distance sensor is its short range. The infrared distance sensors have a range between 10 cm to 500 cm, a very small and limited range which would not allow a significant amount of time to redirect the robot’s path. The price range of infrared distance sensors are relatively cheap ranging from $12.99 to $24.95.

3.2.3.5 Kinect

The Kinect sensor made by the Microsoft corporation, was originally designed for the Xbox 360 game system. Due to the technology that makes up the Kinect it has been used for many other avenues including scientific research, education, and robotic systems. The Kinect is made up of four primary sensors: an RGB camera, an IR depth camera, a microphone array, and an accelerometer as well as a motorized tilting axis. Each of these sensors has their individual usefulness and to the overall operation of the SAFER Knights system.

3.2.3.6 RGB Camera

The Kinect sensor features an RGB camera which has a resolution of 640x480 and a frame rate of 30 frames per second. This camera visualizes the red, green, and blue color spectrums. The SAFER Knights robot's utilization of this sensor involves the recording functionality. The SAFER Knights could need to utilize an on-board camera to record video of the usage of the device. Since the Kinect comes with an RGB camera this can be utilized to record the necessary video. The Kinect also has the potential to record in higher resolutions, although when recording in higher resolutions the Kinect must sacrifice the frame rate to do so. Another use for the RGB camera is to utilize the video input to make path-based decisions. The SAFER Knights system can utilize the video stream with computer vision to see and analyze different objects including people and obstructions. From there the system can utilize the information to aid in its following techniques and further aid in object avoidance.

3.2.3.7 Infrared Depth Camera

The next sensor featured by the Kinect is the sensor's infrared depth camera. This sensor is the most significant in the functionality of the connect. The infrared depth camera is used primarily for depth perception. This allows the Kinect to determine the distance or depth of everything in its field of vision. The infrared depth camera is made up of two different lenses the first being an infrared projector and a monochrome complimentary metal-oxide semiconductor (CMOS) sensor. The infrared projector shines multiple infrared beams within its field of vision. The monochrome CMOS sensor is then used to detect the reflection of infrared beams of all surfaces within the Kinects field of vision. The Kinect calculates the amount of time that the infrared beam takes from when it leaves the infrared projector to when it is received by the monochrome CMOS sensor, from there it can determine the depth of each beam and draw a three-dimensional map based on all the depths of the infrared beams. The infrared sensor also has a relatively large range of depth perception, between three to twelve feet. This method of depth retrieval is not affected by lighting conditions, this works in favor of the SAFER Knight system as it is designed to be used at night without any stable lighting conditions. This sensor of the Kinect is comparable to the LiDAR sensors, following the similar way of using beams of light to measure distance and create a three-dimensional model of the space. Though the Kinect is able to measure these values in such a way as to be able to create high definition outputs while still fitting in an efficient size and price range compared to LiDAR which for higher quality sensors require larger sensors which are more expensive. The infrared depth functionality of the Kinect can be used for the integral part of the SAFER Knights system. To avoid objects the SAFER Knights system needs to be able to detect objects as well as how far the object is from the robot. Using this information, the robot can make the appropriate decision to avoid the object based on size and distance.

3.2.3.8 Microphone Array

The Kinect also hosts a microphone array. This microphone array consists of four different microphones that are utilized in tandem. With the four microphones used together it can isolate voices from background audio, allowing voice isolation. The impact this has on the SAFER Knights system is the potential for voice isolation which can be used for voice commands. A potential for the SAFER Knights system is voice commands to initialize emergency mode. The voice isolation that comes standard with Kinect gives potential to the possibility to incorporate voice commands while eliminating the noise produced by the system.

The Kinect incorporates its multiple sensors to utilize even further functionality. By obtaining the data from the RGB camera, which is used for facial and body recognition, and combining it with the data from the infrared depth camera, the Kinect is able to do advance functions such as track and map multiple human bodies at once. Using this advance function, the SAFER Knights system can increase the precision of its tracking and pathfinding. With the SAFER Knights utilizing a follow system for the pathfinding there is a margin of error that can occur depending on the following technique. To minimize this margin the SAFER Knights system can utilize the body tracking to better pinpoint the exact body that is to be followed.

3.2.3.10 Drive Train

The power wheels used in the SAFER Knights project comes with rear wheel drive. This leaves us with a choice to change them to front wheel or four-wheel drive. In order to make the best choice, each were researched and compared.

The first option was to change the vehicle to front wheel drive. Some advantages of this would be: more room inside the vehicle, fewer components, less weight, better power consumption and improved traction. Disadvantages may include: limited acceleration, farther forward center of gravity and steering difficulties.

Another option was four-wheel drive. Things such as: having the best traction of all options and being great for off-road driving are all advantages. Disadvantages would include: the weight and power consumption.

The final, and most practical option, was to leave the vehicle as rear wheel drive. Rear wheel drive has advantages and disadvantages as well. Advantages would include: better handling for dry conditions, easy maintenance, even weight distribution and great turning radius. Some disadvantages would be:

Below is a table to visually represent the comparison between front, rear and four-wheel drive.

Table: Comparison Between Wheel Drives

	Order (Best to Worst)
Power consumption	Front Wheel Drive Rear Wheel Drive Four Wheel Drive
Traction	Four Wheel Drive Front Wheel Drive Rear Wheel Drive
Steering	. Rear Wheel Drive . Four Wheel Drive . Front Wheel Drive
Weight	. Front Wheel Drive . Rear Wheel Drive . Four Wheel Drive
Cost	. Rear Wheel Drive . Front Wheel Drive . Four Wheel Drive

After researching the above choices, it was in the best interest of the project to leave the vehicle as rear wheel drive. Although it finished last in traction, for dry conditions rear wheel is perfectly acceptable. The deciding aspect was the cost. Usually rear wheel drive is more expensive, but in terms of this project it was pre-made and therefore provided no extra cost at all. In conjunction with the cost, no one in the group is in the mechanical engineering discipline. This would create many more problems than advantages.

3.2.3.11 Hand Held Tracker

SAFER did require a companion device to allow the user to communicate with it. This device as discussed before can be in many forms depending on some of the design choices made for the main systems of SAFER. The two systems that can impact the device are the following sub system and the object avoids sub system, with the following sub system having a greater impact than the object avoidance sub system. The following sub system can use an array of technologies from an Infrared sensor array or ultra-sonic sensor array to Computer Vision. Some of these options can require a beacon to emit the signals needed for following. These include infrared, wireless signal strength and over wave-based formats. This beacon could emit the signal that could be picked up by the robot. Depending on the beacon type, the robot could use different processes to determine the location of the beacon and take actions to move towards it.

There is a more detailed description of how this might work in other sections, therefore we could not get into the details of the mechanics in this section. The other device that may be used for SAFER is a cell phone using Bluetooth and global positioning system.

This device could be carried by the user of the robot while they walk along their path to their location. While the user is walking along their path to their location, the Bluetooth enabled device could stream the global positioning system location of the device. This could allow the SAFER to know the location of the user along with the time they are located at that position. This information could be streamed into the robot at a rate of at least or less than one position per second. With this information, the robot can simply try to mimic the location of the user using its own global positioning system. This could allow the robot to seem to follow the user when, in fact, it is just following the user location. The robot can take actions to be at the location of the of the global positioning system coordinates as a set number of seconds after the user was already there. This did help the robot keep its set following distance.

Through our research, we found it is most likely that we could use the method of streaming the global positioning system coordinates from a blue tooth enabled device. This method could come with a few problems. The robot did have no idea of the object in front of it without some kind of object avoidance. This is not inherently baked into the robot following system if we use the method of streaming the global positioning system coordinates from a Bluetooth enabled device. This did mean we had to have to use some other form of hardware and software that the robot was use to sense any object in its path. The most likely candidate for this is the use of an Xbox Kinect. With the right software this device was be able to output a depth map and red green blue video feed of the environment around it.

One improvement that may be made is addition of some average. As is the way of most technology, global positioning systems have a margin of error for accuracy. This margin is usually around one to three feet depending on the model of global positioning system we buy. We expect that the global positioning system we buy did have a margin of error around one and a half feet. To help compensate for the margin of error in accuracy of the global positioning system, we can take the point and average them together. This did form a smoother flowing path for the robot. This smoothing effect is applying a curve of best fist to a dot plot. The figure below shows how the this could produce a better outcome for the robot's path while following the user. The left part of the figure shows how the robot would travel if it were to go to the location of each of the points streamed to it by the user’s global positioning system enabled Bluetooth device. This path is very indirect and would not be very useful to the user. It would involve a lot of speeding up and slowing down while swerving, wasting a lot of battery while looking like it has a little too much acid in its batteries. The right part of the figure shows how the path could be drastically improved by adding in averaging. In this method the robot would look at some number n of the global positioning system coordinates in front and behind each of the points and find an average. This did result in a line more accurate to the actual path of the user and showing the user that it in fact has the right amount of acid in its batteries. This method is very similar to the idea of fitting a curve to a dot plot in some statically analysis software.

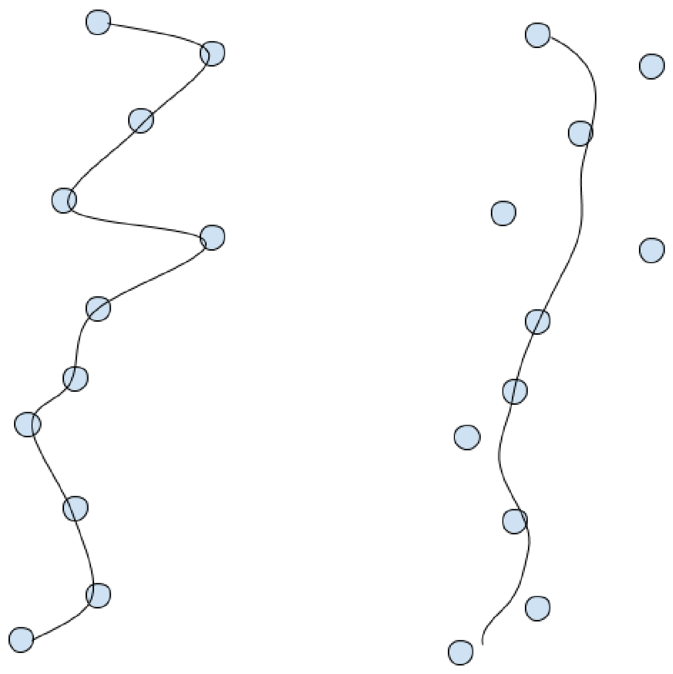


Figure: Non-Averaged Vs Average Paths

Thinking about the global positioning system coordinates from a Bluetooth enabled device once again. This global positioning system enabled Bluetooth device did have to have interface for the user to interact with. If we were to develop our own global positioning system enabled Bluetooth device, it would involve a lot of development. We would have to build a piece of hardware with a custom-built operating system and interfacing software as well as custom chip set for the device. This would be a large undertaking in of itself do the complexity of modern-day electronics. The thought of even undertaking such a task would ensure much sleepless nights for the SAFER group. To avoid such nights, we did look for a global positioning system enabled Bluetooth device that already has the components we need built in to it. The components and characteristics we are looking for in a prebuilt global positioning system enabled Bluetooth device are:

- A built in Bluetooth chip set
  - Preferably with Blue tooth low energy
- GPS capable hardware
  - Preferably with good accuracy and precision
- A Compass
  - For determining direction
- The ability to be hand held
  - Preferably with one hand
- A built-in power source such as a battery
  - Preferably with more than 2 hours of capacity
- An operating system that can support:
  - A Graphical user interfaces

    - Device communication for GPS and Bluetooth
    - Other application to be run
- A Screen
    - Preferably large enough for the user to be able to see the necessary information
    - Preferably also illuminated so that it can be seen at night
- A way for the user to operate the device
    - Preferably via an interface that can be operated with one hand, but while walking to their destination
    - A touch screen interface comes to mind with thinking about this requirement.
- Possibly Speakers
    - For playing sounds from an application
- Possibly Microphone(s)
    - For hearing the user

Looking at all of the components and characteristics we are looking for in a prebuilt global positioning system enabled Bluetooth device the best way to develop this would probability be using a smart phone with global positioning systems and Bluetooth. Most modern and not so modern smartphones have these capabilities built into their chip set along with other components and characteristics mentioned above such as:

- A compass,
- Being hand held
- Built in power supply such as a battery
- An operating system that can support other applications and device communications
- A screen with a touch interface
- Speakers and microphones.

Below is an image of a spare smart phone that was use during the development of SAFERs mobile application during the semester. It is a Samsung galaxy S4 running Android.

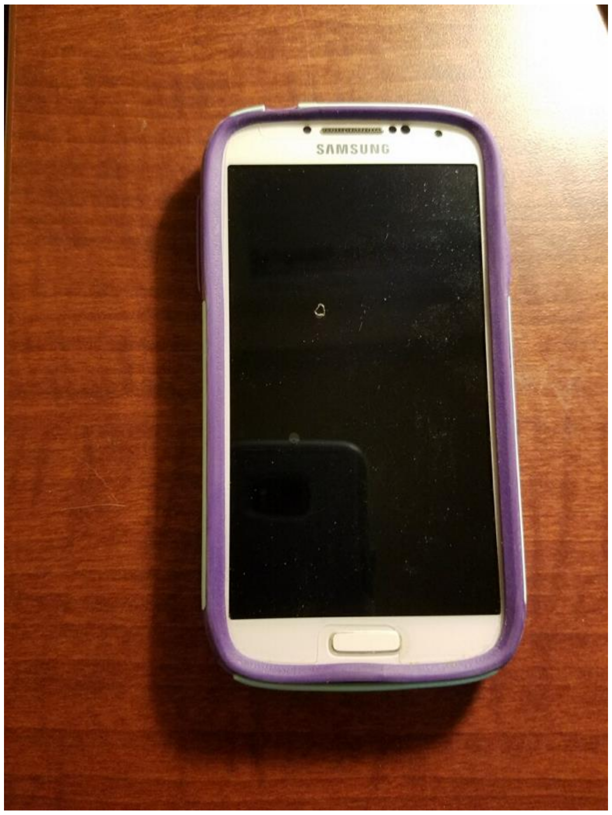


Figure: Smart Phone for SAFER Development

As discussed several times before, SAFER did require a companion application for it to follow the user. As also said many times before, this is not the only way that the robot can follow the user around. With the Kinect we are using we may be able to use computer vision to trace the user as they move around the environment, but also remembering that this method has some challenges. Some of these challenges include knowing what body to track or seeing around corners. These are many more challenges and some advantages, but to read more about these please see the other section in which they are discussed, the details are omitted from this section because it is off topic and would not be a brief discussion.

When using a smart phone, there are many different kinds of makes and models to choose from. The major makers of smartphones currently are Apple and Samsung. These two providers both use different operation systems. Apple uses an operating system they call iOS and it can only be loaded onto the phones that they manufacture. Samsung makes phones and then loads an operating system onto them call Android. Android is an open source project that was started and is largely maintained by Google. This operating system is free to put on any device with the proper hardware to support it. This is a stark difference between Apple's iOS and Google's Android. Due to the open source nature and large undertaking that is required to roll your own operating system the Android operating system is the widest spread operating system among smart phones. This is largely due to the fact that lot of lower end smartphones use android to reduce cost. These manufactures also do not keep up with security updates and leave these lower end devices vulnerable to breaches, but once again that is off topic and was not be discussed in this paper for the sake of brevity.

These two operating systems leave our group with a decision to make. We have to choose between using the Android mobile operating system or iPhone iOS mobile operating system for our application development.

iOS	Android
• Programing Language: Swift – a new language similar to Objective C, it can only be developed in XCode using a Macintosh computer • Cost: Requires a developer to pay a developer fee to be a part of the developer's network. The fee is around $100 • Compatibility: iPhones are generally much more compatible with the development environment. • Ease of Development: Medium	• Programing Language: Java – Google has also created an android development IDE with a great deal of tools • Cost: Free • Compatibility: Difficult, there are many makers of Android phones, and many versions of the software • Ease of Development: Medium

There are some platforms that can support development of an application for both operating systems semitonally, but these platforms can present some issues such as:

- Corona SDK
- Unity
- CoCos2D
- PhoneGap
- QT
- Xamarin
- Marmalade
- Appceloraor

These options seem like they might have been a good idea, but they do not pass certain needs for the SAFER. First, they do not get around the fact that we would have to pay for a developer license to publish that application to the iPhone. They are also written in a language that is interpolated into the native language. This can cause issues with different versions of the software and can cause issues that are hard for the developer to diagnose. The other platforms also make it hard for the developer to access lower level hardware like the Bluetooth sensors and global position systems. This can cause issues because these are the main systems we need to utilize to perform the main objectives of the application. Another issue with these platforms is the way they are meant to be used. These platforms are usually meant to make mobile games, the most popular kind of mobile application.

3.2.3.12 Levels of Autonomy

While researching for the development of SAFER we needed to learn more about the autonomous robotics industry. We have learned that there are three levels of automation for robotics, these relate mostly to autonomy cars, but have strong implication for our project of an autonomous following robot:

- Level 0: The machine has no automation at all, it can only be operated by a human to perform its task. This is hard to even call a robot to begin with due to its lack of intelligence, it is basically just a machine.
- Level 1: Some simple automation, in a car like cruise control to maintain a constant speed while operating.
- Level 2: The robot can take over in safety situations, for example breaking. It can also take over for things like accelerations and maintain a safe driving distance. This is the first sign of intelligence and decision-making.
- Level 3: The robot can mostly drive on predefine paths without the user's interaction
- Level 4: The robot has full autonomy in most predefined situations
- Level 5: The robot has full autonomy in every situation no matter the circumstances.

3.3 Part Comparison and Selection

Although many different technologies and components have been researched, not all are ideal to meet the demands of the project. This is where a comparison between parts is needed to narrow down the components that best fit the job. After comparing the components are compared, an analysis is done to select the ideal component which was then be purchased and tested for use in the final project.

3.3.1 Power Wheels

In order to move across campus, a vehicle base must be either built or bought to achieve our goals. Since building a base would be timely and possibly costly it is in our best interest to purchase a vehicle that is cheap, functioning, and small enough to get around campus without being an annoyance. Therefore, the Power Wheels was chosen, a toy meant for small children to be able to emulate driving a car. Fairly common, these vehicles can only be used for a short time before the child outgrows them, so plenty of used ones can be found online for relatively cheap. Browsing Facebook Marketplace, a fully-functioning Green Lil’ Kawasaki 6 Volt Battery-Powered Ride-On Power Wheels was found for only $25. The seller also offered up a non-functioning Silver Ford F-150 12 Volt Battery-Powered Ride-On Power Wheels for free. As it is a bigger vehicle with a better battery, the hope is to get this Power Wheels up and running and use it. If unable to fix it, the smaller Lil’ Kawasaki was be used instead.

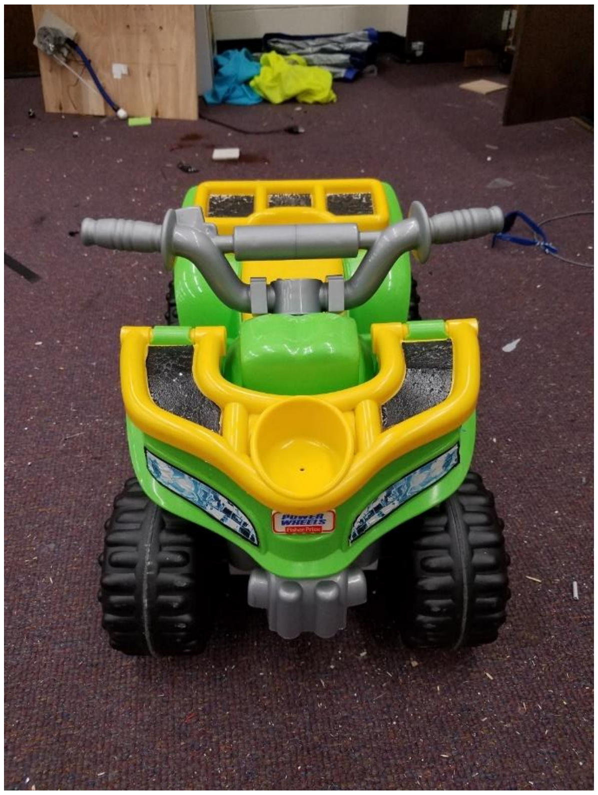
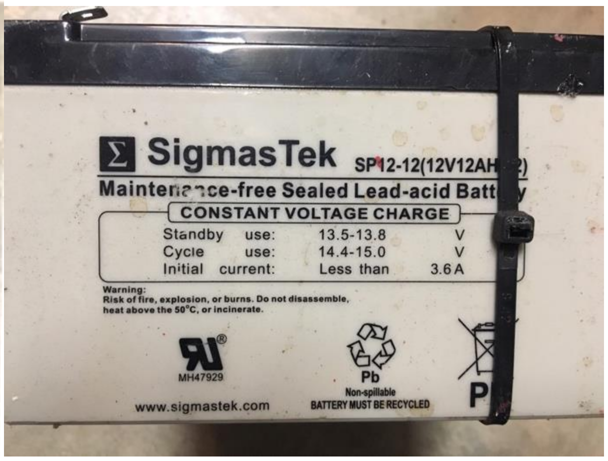


Figure: Power Wheels Green Lil' Kawasaki 6V Battery-Powered Ride-On Quad

Figure: Power Wheels Silver Ford F-150 12 Volt Battery-Powered Ride-On

3.3.1.1 Possible Other Parts

In our possession we have two battery operated electrical scooters that we plan on using for some of the components if need. This was a most cost-effective way of building SAFER. Below is a picture of the batteries that can be used for SAFER power supply. From our experience the batteries in on scooter can travel between one and two miles with a fully-grown adult on them. Being that there are two scooters for a total of four batteries we feel confident that these was be plenty for powering SAFER. There is also a battery monitor system that is attached to the two scooters. Also, from the two scooters we have a possible drive train alternative with variable speed control. This can be used in the event that the current power wheels are not fast enough.



Figures: Battery Powered Scooter Battery

Figures: Battery Powered Scooter Battery Indicator (left) and Drive Train (right)

3.3.2 Electronic Speed Controller

Since the robot does following behind the user at a constant distance rather than a constant speed, the robot does have to be able to adjust its speed according to the user's walking pace. To do this, an electronic speed controller is being looked into, as its function is to control and regulate the speed of an electronic motor like the one used in a Power Wheels. It does vary the amount of power going to the servo-motor controlling the wheels of the Power Wheels. The speed does be changed according to the master controller, which did send signals to the electronic speed controller to make adjustments as necessary. ESCs are also good for dynamic braking, so it looks like a good fit for the project's needs.

3.3.3 Servo Motors

A servo motor is needed to manually change the direction the wheels of the Power Wheels are facing. The servo motor utilizes a potentiometer to provide an analog signal to show position. The Power Wheels, being a children's toy, typically has the child rotate the steering wheel of the vehicle to turn left or right, but since the robot was be autonomous it must be able to steer on its own. A servo motor as be used in place of a driver to turn the steering wheel. The servo motor was be connected to the master controller via a PWM connection, from there the master controller was decide when, how far, and in what direction the servo motor needs to turn by taking into account both the user's coordinates, the robot's own coordinates via Bluetooth, and object detection via the sensor system for course correction.

Most servos only turn about 170-180 degrees, but luckily continuous servos were created that are able to rotate the whole 360 degrees, which was more useful for the project as the wheel may need to turn more than 170 degrees at times. By nature of their design, continuous servos come with built-in H bridges, negating the need for building ones of our own. This way, the continuous servo motor did only need power and a pulse signal to operate the servo. The Raspberry Pi has a pin that can produce the necessary pulse to work a servo motor. One thing to take into account was the power, as the servo did most likely draw too much current and crash the Raspberry Pi if it were to be used as the power source for the servo, so instead the servo's power did have to come from a separate power source.

Another important factor to take into account is torque. Depending on the size and weight of the Power Wheels, a more expensive servo motor with more torque may need to be purchased in order to get the wheel to turn. For now, the smaller Lil' Kawasaki was the assumed vehicle of choice since it is currently in complete working order. Since the Lil' Kawasaki is small and light, a cheaper servo motor with less torque can be purchased while still succeeding in doing its job. Below is a brief analysis of different servo motors looked into to fit our purposes.

3.3.3.1 5V DC TowerPro SG92R

This 5 V DC servo motor does not have much torque but is rather cheap. As the manufacturer is Adafruit, the makers of the Raspberry Pi, this servo motor is great for hobbyists and people using the Raspberry Pi to control it. However, it uses a positional rotation, which is not ideal for turning a wheel 360 degrees. It is rather cheap at only $5.95.

3.3.3.2 5V DC FS90R

This little servo motor, also from Adafruit, has a similar torque to that of the TowerPro SG92R and the same voltage, but uses a continuous rotation, which allowed it to turn a full 360 degrees. It’s a little more expensive than the first servo motor at $7.50.

3.3.3.3 Gearmotor 6 V DC Servo

From Parallax Inc., this servo motor has more torque than the previous servos looked at while still being having a continuous rotational motor. Parallax is also well-known and has resources to help with questions and extra information. However, it is more expensive than the first two servo motors at $14.99.

3.3.3.4 LewanSoul LX-16A Serial Bus Servo

Different than the others, this servo is meant for situations where lots of torque is needed, supplying a good 208 oz/in of torque, which is more than any of the other servo motors looked into. This servo motor is ideal if our secondary Power Wheels, the Ford F-150 Power Wheels, is able to be fixed up enough to become our primary choice of vehicle for the project. Because the Ford F-150 is much larger and heavier than the Lil’ Kawasaki, the servo motor was much more powerful to control the wheel. The LewanSoul LX-16A should be enough to do the job. This servo also gives real-time feedback on position, voltage, and temperature and works at 6.6 V.

3.3.3.5 Comparison and Conclusion

Looking at the table below, there are various factors to be looked at when choosing a servo. The major concerns when looking for a servo motor was torque and voltage, both important facets for the functionality of the servo and the project.

Table: Servo Motor Comparison

Product	Torque (oz-in)	Voltage	Motor Type	Seller	Price
5V DC TowerPro SG92R	22.22	5 V	Positional Rotation	Digi-Key	$5.95

5V DC FS90R	20.86	5 V	Continuous Rotation	Digi-Key	$7.50
Gearmotor 6 V DC Servo	38	6 V	Continuous Rotation	Parallax	$14.99
LewanSoul LX-16A Serial Bus Servo	208	6.6 V	Positional Rotation	Amazon	$14.99

However, it was decided in the end to use the Gearmotor 6 V DC Servo, not only because it provided more torque than the first two servo motors looked at, but also because members of the group already have them, so instead of spending $14.99 there was be no cost incurred instead. If more torque is needed, then the LewanSoul LX-16A was be purchased.

Figure: Acquired Gearmotor 6 V DC Servo

3.3.4 Master Controller

Various development boards were researched to see what the best fit would be to achieve the objectives for the project. Some house microcontrollers while others have the power of being single-board computers. Kept in mind were the capabilities of each, ease of use, resources available, and price. Below are some of the contenders that were looked into.

3.3.4.1 Arduino Uno

A very popular microcontroller development board in the maker world, this was looked into first, as there are plentiful resources online of how to use it and projects to look at. It's very user friendly and made for beginners while still being very robust and having various capabilities. Being open-source, many people upload their code as well with their projects and there are many forums for discussion. It uses an Atmega328P microcontroller on the board. It has a USB cable, power jack, a set of digital and analog input/output pins, 16 MHz crystal, an 8-bit Microchip AVR CPU, and various purchasable peripherals for specific uses. If purchased directly from the online site's store, the cost is $22. Luckily enough, one of the members of the team already have an Arduino Uno in their possession, so the cost is not a concern.

3.3.4.2 MSP432

The MSP432 is a mixed-signal microcontroller development board made by Texas Instruments. A mixed signal IC such as this has both analog and digital circuits on it. They process both analog and digital signals together, and they are often cost-effective. The MSP432 has a 32-bit ARM Cortex-M4F CPU and is meant for low power requirements. This microcontroller was looked into because UCF students are familiar with this Texas Instrument line of microcontrollers, as they were used in previous class' labs to write code, so a main advantage of using the MSP432 is familiarity with the product. Programming would also be in C, and the MSP432 has several built-in peripheral devices. Various versions of the MSP432 have slightly different specs and pricing, mostly around $12-$20.

3.3.4.3 BeagleBone Black

Not every option that was looked into was a microcontroller development board. The BeagleBone Black by Board is one of those cases, as instead it a low-power single-board computer. It was made by Texas Instruments collaborating with Digi-Key and Newark element14. As it uses open-source software and was intended to be an educational board, it also has a good amount of resources online. And since it is a single-board computer, not a microcontroller, there is more processing power available. It uses an AM3358 ARM Cortex-A8 processor, has a USB port, a micro HDMI port, a microSD card slot, 2 GB on-board storage, and can run Linux. The price runs around $49.

3.3.4.4 Raspberry Pi 3

Another single-board computer, this is also a popular choice in the maker world. It uses a 1.2 GHz Broadcom BCM 64-bit CPU, with 1 GB RAM, 4 USB ports, Ethernet port, HDMI port, and a Micro SD card slot. Like the Arduino, because of its popularity, there are plenty of resources and projects online to work off of. Code is written in Python, a fairly simple language that is well known. Relatively cheap, it provides a lot of bang for your buck at around $35.

3.3.4.5 LattePanda

The LattePanda is different than the others in the fact that it is a development board that runs the full version of Windows. It does this with the help of an Intel Quad Core processor and runs the full Windows 10 Operating System. This is definitely the most powerful and versatile of the bunch, and has three USB ports, integrated WIFI, and Bluetooth. An Arduino co-processor is utilized to control peripheral devices. It has 2 GB Ram and 32 GB of onboard flash memory. All this usability comes at a cost, and the most basic unit starts at $89.

Table: Microcontroller Comparison Chart

Possible Solution	**Type**	**Online Resources**	**Level of Difficulty**	**Bluetooth**	**Price**
Arduino Uno	Microcontroller	Great	Easy	Add-On	$22
MSP432	Microcontroller	Okay	Moderate	Add-On	$12-20
BeagleBone Black	Single-board computer	Good	Easy	Yes	$49
Raspberry Pi 3 Model B	Single-board computer	Great	Easy	Yes	$35
LattePanda	Single-board computer	Okay	Moderate	Yes	$89

3.3.4.6 Conclusion

Comparing all of the options that were researched, it was determined that the Raspberry Pi 3 was best and was used as the master controller for the robot.

3.3.5 Speakers

For this project, it was necessary to use speakers if the robot goes into Emergency Mode. When it does, siren-like sounds began playing that were stored as audio files from the Raspberry Pi. An important thing to take into account was the power being used by these speakers, as well as how they connect to the Raspberry Pi. To avoid drawing away any power from other subsystems, the speakers was self-sustained battery-powered speakers that can be recharged separately. That way there is less of a concern about power going to the speakers while the robot is in use and may need power directed somewhere else. The Raspberry Pi has 4 USB ports and a combined 3.5 mm audio jack and composite video, so there are options on how to connect a speaker to it.

Table: Speakers Comparison Chart

Speaker	Supplier	Power Supply?	Connection	Sound Quality	Price
Speaker for Raspberry Pi	Dexter Industries	Yes	Aux jack	Okay	$7.99
Z50 Grey	Logitech	No	Aux jack	Good	$16.99
Docooler MiniHamburg Speaker	Docooler	Yes	Aux jack	Poor	$6.36
Mini Portable Speaker Plug & Play	Leadsound	Yes	Aux jack	Good	$16.99
Mini Hamburger Speaker	TTSAM	Yes	Aux jack	Good	$8.99

Looking at the speakers, there were many options at many different prices and sound qualities. The Mini Hamburger Speaker from TTSAM was chosen because it was cheap without sacrificing too much sound quality and was be able to connect to the Raspberry Pi via audio jack. It also could be powered via USB before use and would last a couple hours before needing to be recharged. This way the speaker was not using any external power source while the robot is being used.

The two main types of speakers are powered and unpowered. An unpowered speaker requires an external amplifier and source to function, but the powered speaker has an internal amplifier and source. In regard to the project, a passive speaker The reason the unpowered type was chosen is because a custom PCB needed to be made and this would be one function of that board. In addition, it was good experience to design the amplifier circuit.

The next decision was what size speaker would be acceptable. The dimensions should be between four and eight inches. Factors such as impedance, peak power, frequency response, weight and price were key in deciding which speaker was needed. In the table below is a comparison table of the mentioned factors for various sized speakers:

Table: Speaker Comparison

Speaker	Specifications				
	Impedance (ohms)	Peak Power (W)	Frequency Response (Hz)	Weight (lbs.)	Price ($)
QTX QT6 6.5"	8.0	100.0	80-16k	15.9 (2 speakers)	26.65 (2 speakers)
SubZero SZS-P8 8"	8.0	100.0	90-18k	8.4	42.70
Skar Audio FSX65-8 6.5"	8.0	300.0	100-8k	3.85	19.99
Visaton BG13P 5"	8.0	40.0	150-20k	1.05	14.20

The QTX 6.5'' was the best option in regard to price. The biggest problem with this speaker was the weight. The SAFER Knights project must be as lightweight as possible to allow for optimal results. It would not be ideal to have speakers that weigh the vehicle down. In terms of power, the QTX speakers were relatively average. T

The Subzeros provided a comparison for the upper end of the preferred sized speaker. They provided approximately the same performance as the QTX speakers. Both had similar peak power, weight and frequency responses. The price and size were where they differed. Since the speakers were extremely comparable, the price difference alone was enough to conclude the QTX were superior.

The next comparison was between the Skar Audio and QTX. The Skar provided another 6.5'' speaker with much less weight. The major difference comes in the frequency range and peak power. The Skar Audio was able to produce 3 times the power, but half of the frequency range. Although the Skar can produce up to 8k Hz, the human ear can best hear sounds from the frequencies 1k to 5k Hz. This is at the upper end of the specification and therefore would result in not using this speaker in the project.

The final comparison was the QTX and Visatron speakers. The price was the only similar aspect of these options. The Visatron had a better frequency range and was far superior in terms of weight. In addition, it also had a much lower peak power. This 5'' speaker fit all the necessary criteria: low power, lightweight and affordable. Because of these key factors and previous experience operating a similar speaker to that of the Visatron, it was an obvious choice to use in the SAFER Knights project.

3.3.6 Lights

The SAFER Knights mission heavily entailed a need for various lights. The lights provide a sense of safety to students by making things easily visible, bringing attention to the area and are also used in the emergency protocol. In order to accomplish this goal the most efficient lights, in both cost and performance, must be used. We had many options including: incandescent, LEDs, metal halide and fluorescent bulbs. After comparing the options with the categories previously mentioned, it is obvious LEDs are the best choice.

Incandescent bulbs were the worst in terms of performance, but the price was something worth noting. When comparing these bulbs to LEDs, the lifetime and lumens per watt were not even close. In addition, the price range between them was basically the same. Further research showed the power consumption comparison was also in favor of the LEDs.

The next comparison was compact fluorescent versus LEDs. Again LEDs were superior in every performance statistic. The key factor was the price ranges because one would expect the better performance product to be higher priced, but LEDs were slightly cheaper as well. This bulb was the closest to competing with LEDs, but in terms of lifetime the compacts were clearly worse.

Due to the high lumens per watt of the metal halide bulbs, they became the final option. LEDs once again won is all the other performance specifications. Aside from the lifetime, the price difference was the key factor in deciding LEDs were the best choice for the SAFER Knights project.

Below is a table that allows for a quick and easy comparison between the various bulb options taken into consideration:

Table: Bulb Option Comparison

	Lumens per Watt	Lifetime (hours)	Price
LED	65 - 85.0	50,000	2 - $6
Incandescent	16.0	1,000 - 2,000	1 - $5
Compact Fluorescent	50-70.0	6,000 - 15,000	2 - $10
Metal Halide	75 - 100.0	6,000 - 15,000	12 - $25

3.3.6.1 LED Flood Lights

The initial idea of the SAFER Knights vehicle incorporates a bigger flood light on the front to illuminate the path for any student. The table below compares possible options to complete the task.

Table: Flood Light Comparison

Model	Lumens	Voltage (V)	Power (W)	Price ($)
CSFL-30-5K	2700	100-277	30	36.95
CSFL-30A-5K	2700	100-277	30	38.95
F9922-31	650	120	9	29.95
LED-SLC12WH	1000	120	12	33.95
OVFL LED 1RH P1 40K 120 DDB HP17 M6	948	120	30	33.95
OVFL LED 2RH 40K 120 PE DDB HP17 M4	1770	120	20	49.95

The first two CSFL models are very similar in specifications. As seen above, they had been the brightest lights of all the options. This advantage came paired with a high power and voltage consumption. The voltage and power could require a different battery than the one currently powering the vehicle. As far as the price, relative to the other options above, it was about average.

The research then was directed toward finding a cheaper option. This was when the F9922-31 was found. Although it had the lowest lumens of all the options, this caused the power consumption to be the lowest as well. The practicality of this option because of the low power and price was what made this one appealing to the overall goal of the project.

Next came the LED-SLC12WH. The power consumption of this model was slightly higher than the F9922-31. The wattage is realistic for the SAFER Knights project and is a serious option. The biggest difficulty of this option would be the mounting. It is possible to elevate it on a pole-like object and keep it down with screws. The biggest advantage of this model was the high lumens it produced.

The 1RH P1 40K model was the closest to a headlight of all the options. This would make the mounting on the light to the dash of the vehicle the easiest choice. As far as the specifications of the light, the lumens were about average, but the power consumption was tied for the highest. The power could be a problem because it could require using a different power source than the one currently operating the vehicle. The price of this option was very reasonable. Overall this option was a strong choice.

Finally, the 2RH 40K was more of a security light that could double as a head light. This model had the second highest lumens of all the above options. Another advantage of this model is the separate lights can be pointed in whichever direction that would be the most beneficial. The power consumption was a little above average. 20 Watts is still an obtainable goal. Although the price was relatively high, it was still a realistic option.

3.3.6.2 LED Controllers

Table: LED Controller Comparison

Model	Voltages (V)	Power (W)	Max Output Current (A)
PWM Dimmer Knob LC-OL-2DIM	12 24	96 192	8
Mini LED PWM Dimmer LC-LF-16DIM	5 12 24	60 144 288	12
LED PWM Dimmer LC-OL-8DIM	12 24	24 48	2
Mini LED Touch Dimmer LC-LF-26DIM	12 24	48 96	4
PWM LED Knob Dimmer LC-LF-2DIM	12 24	360 720	30
LED Mini Dimmer LC-LF-19DIM	5 12 24	25 60 120	5

There are many different kinds of LED controllers. The problem with choosing a specific one is it needs to meet the specific needs of whatever light is chosen. The table above was made to represent a variety of voltages and currents to cover whichever light ends up being necessary for the SAFER Knights project.

The first search was for low current controllers, from 1 to 4 amps. LC-OL-8DIM was the first one with a maximum current of 2 amps. This model comes with barrel jacks specifically for easy connections between lights and the power supply. The controller also has a knob used as a dimmer and requires an input voltage of either 12 or 24 volts. The connections are 5.5 by 2.1 millimeter jacks. The other low voltage controller was the LC-LF-26DIM. This controller is used for single color 12-24 volt LED lights. It also used a 5.5 by 2.1 millimeter barrel connector between the lights and power supply. If the input is a 12 volt supply then it can deliver up to 48 watts, but if it is a 24 volt supply then it can be up to 96 watts.

The next set of controllers ranged from 5 to 10 amps. The lowest model was LC-LF-19DIM with a maximum of 5 amps. This controller even comes with a RF remote and has an input range of 5, 12 or 24 volts. The maximum power is 25, 60 and 120 respectively. There was one more controller that fell within the range and this was the LC-OL-2DIM with a maximum current of 8 amps. The knob on the dimmer operates with a 12 or 24 volt input and can deliver 96 or 192 watts respectively. The wires are connected with a screw down mechanism between the power and lights. This controller is specifically used for one color LEDs.

Finally, higher current controllers were sought out. The LC-LF-16DIM operated with less than 12 amps and an input voltage of 5, 12 or 24 volts. This controller can deliver up to 60, 144 or 288 watts respectively. It operates a single color and has a built in strobe option. The strobe option specifically relates to the project due to the flashing lights feature in the emergency protocol. The last controller operated in less than 30 amps making it able to handle the most current by far of all the others. The LC-LF-2DIM uses a knob and an input voltage of either 12 or 24 volts to deliver up to 360 or 720 watts of power. Screw down connections are made between the power supply and the lights. This controller is best used for strips, modules, light ribbons or almost any other LED lights.

3.3.6.3 LED Strips

Table: LED Strips Comparison

Model	Lumens per Foot	Power (W)	Length (in)	Price ($)
LED-T2430L-1-WT	135	2	12	19.00
LED-T24W-1-WT	200	3	12	27.00
Kichler 6HS30K12AL	215	4	12	16.13
WAC Lighting LED-T24C-2IN-WT	200	0.5	2	7.50
Kichler 6HS30K06AL	215	2.7	6	11.25
Progress Lighting P7040-30	120	3	12	24.39
WAC Lighting LED-TX2430-6IN-WT	275	2	6	21.00

The initial concept for the SAFER Knights project has the vehicle wrapped with LED strips on the sides. The purpose of these strips is to illuminate the surrounding area not covered by the flood light on the front hood. Strips are low power consumption and lower in price than traditional lights.

The first three options in the table above were 12 inches long. This length was first chosen because it did cover a whole side of the vehicle. As seen above, the Kichler had the highest lumens per foot. The most surprising aspect of the Kichler was the price. Although it produced the most brightness it was actually the lowest cost of the three. The LED-T2430L-1-WT was the lowest power consumption of the three options. This is the most likely cause of the low lumens per foot as well. It is a safe assumption to conclude the Kichler was the best fit of the options discussed.

Next the thought was to find smaller pieces that can be put together and possibly be more cost effective. This was where the LED-T24C-2IN-WT, LED-TX2430-6IN-WT and 6HS30K06AL were discovered. The LED-T24C model was the smallest of them with a length of 2 inches. Due to the shear size of the strip, the power consumption was extremely small. Despite the low power, the lumens per foot was very comparable to the rest of the options. Because this specific model's power consumption was only 0.5 watts, it makes this option easy to turn on and off. The other two options were 6 inches in length. LED-TX2430-6IN-WT had the highest lumens per foot of any strip on the table above. The amazing part about this model was the average to low power consumption the strip has relative to the rest. These things make the TX2430 the most well-rounded option.

3.3.7 Memory Storage Devices

Since an option was to record surroundings, onboard storage was a possibility. An hour's worth of video takes up about 5 GB of storage, so getting enough memory so that the robot can store several hours' worth of video is within reason and price. The Raspberry Pi has both USB ports and a micro SD slot, so there were plenty of options to choose from. The main three would be a micro SD card, a flash drive, and an external hard drive. External hard drives hold a lot of memory but are more expensive and bulkier. Flash drives are common and relatively cheap, and since the Raspberry Pi has 4 USB slots it won't take up too many resources from the Raspberry Pi. However, because of the shape of a USB, it's possible that the movement of the vehicle might cause the flash drive to snap in half since it sticks out of the USB outlet. Micro SD cards are cheap, have a good memory capacity, and have less possibility of breaking than flash drives. However, there is only one micro SD slot on the Raspberry Pi. Below is a table to quickly compare the cost of each possible storage device.

Table: Memory Storage Device Comparison

Product	Memory Capacity	Price Range
Micro SD card	128 GB	$8-26
Flash Drive	128 GB	$20-35
External Hard Drive	128 GB	$10-50

Luckily, one of the team members has a spare external hard drive that connects via USB, and so that is what was used for the project. The externals hard drive used is the G-Drive mobile USB, which has 1 TB of storage. This was more than enough memory to meet our storage requirements.

3.3.8 BJT Versus MOSFET

When it comes to convenience and price, BJTs are the “go-to” transistor. Although, efficiency is one thing they lack because of the current used when turning on.
The other big factor was BJTs can easily burn. In theory when they are placed in parallel the current will be split equally between both transistors, but this is not a realistic assumption. The only way to correct this flaw is to put relatively small resistors at the emitter of each BJT. This dissipates and controls the voltage across the base-emitter allowing the BJT to continue working properly.

Considering the circuit gave 5 watts to an 8-ohm speaker; it is considered a low power system. Since most common MOSFETs require about 10 volts to even turn on, the project would require a logic level MOSFET. Logic level transistors are much harder to find and more expensive than simply using a BJT.

Given the above research it was apparent BJTs were the most realistic choice of the two transistors for the SAFER Knights project. Below is a table to visually represent a variety of BJT models:

Table: BJT Models

	2N2222A	*KSD1691 G*	2N3904	2N5401	TIP2955	2N3251A
Type of BJT	NPN	NPN	NPN	PNP	PNP	PNP
Max current (Ic) (A)	0.8	5.0	0.2	600.0	15.0	0.2
Emitter-Base Breakdown Voltage (V)	6.0	7.0	6.0	5.0	7.0	5.0
Collector-Base Breakdown Voltage (V)	75.0	60.0	60.0	60.0	100.0	60.0
Base-Emitter Saturation Voltage (V)	0.6	0.9-1.2	0.65-0.95	1.0	1.8	0.9-1.2
Collector-Emitter Saturation	0.3-1	0.1-0.3	0.2	0.2-0.5	1-3	0.25-0.5

Voltage (V)						
DC Current Gain	35-300.0	200-400.0	60-300	50-240.0	70.0	300.0
Price per 10 ($)	2.00	4.90	1.00	2.50	23.00	0.89

3.3.9 Microcontroller Comparison

This section aims to compare various microcontrollers to be used specifically for testing and building the custom PCB, as opposed to the pre-made boards that house microcontrollers and single-board computers discussed earlier.

3.3.9.1 Atmega328P

The Atmega328P is a popular and well-known option as it is the microcontroller used in the Arduino Uno board. Widely available and used by many hobbyists and engineers alike, this was a good option since there are plentiful resources online. It has a 32 KB program memory size and 32 available pins. It supports UART, SPI, and I2C connections. A major advantage is that testing for this microcontroller can be done using the Arduino Uno board, which is easy to program. The Arduino was already in possession of one of the team members, so testing already started with the board. The cost of a single microcontroller is $2.14 from Digi-Key.

3.3.9.4 ATmega168V

This microcontroller can be found on the Lilypad Arduino Main Board, another board sold by Arduino. Similar to the Atmega328P, it has 32 pins available and supports UART, SPI, and I2C connections. It only has 16 KB of program memory space. This microcontroller serves as another possible choice for the custom PCB and can be programmed and tested using the Lilypad Arduino Main Board. However, the price of the microcontroller is slightly higher than the Atmega328P at $3.01 at Digi-Key.

3.3.9.3 M430G2452

This microcontroller happens to be an extra part with the MSP-EXP430G2 Launchpad development kit, which makes the actual price of the microcontroller $0 since multiple team members already owned the MSP430 Launchpad from previous classes and was already in the team’s possession. It is a low-power 16-bit MSP430 microcontroller with universal serial interface and 8 KB flash memory. While not the same microcontroller that is on the MSP430 Launchpad, the MSP430 Launchpad has a microcontroller with similar specs and capabilities, so the microcontrollers are similar enough to not cause concern when testing using the MSP430 Launchpad. To purchase the microcontroller by itself, it is only $1.91 on Texas Instrument's website.

3.3.9.4 MSP430FR5994

The MSP430FR5994 is another low-power solution made by Texas Instruments and is also featured in the MSP430FR5994 Launchpad Development Kit. It has a Low-Energy Accelerator (LEA) for digital signal processing, 256 KB of embedded FRAM, and 8 KB of SRAM. The Launchpad Kit has 40 pins and can be used with multiple development tools such as TI Eclipse-based Code Composer Studio and IAR Embedded Workbench to help with power management and debugging. Texas Instruments provides a good amount of software examples and design files for the Launchpad Kit, and is an easy-to-use development tool for both beginners and professionals alike. The microcontroller itself is $3.55 and the Launchpad Kit is $16.99 from the Texas Instruments website.

3.3.9.5 Conclusion

Below is a comparison between the aforementiond microcontrollers that the team is currently looking into. Since multiple team members have the MSP-EXP430G2 Launchpad development kit, it was decided that this would be the best choice. Not only is the M430G2452 the cheapest compared with the other microcontrollers, but the MSP-EXP430G2 Launchpad development kit also conveniently comes with an extra microcontroller.

Table: Microcontroller Comparison

Microcontroller	Resources	Seller	Price
Atmega328P	Plentiful	Digi-Key	$2.14
ATmega168V	Okay	Digi-Key	$3.01
M430G2452	Good	Texas Instruments	$1.91
MSP430FR5994	Good	Texas Instruments	$3.55

3.3.10 Bluetooth Module

We will be using a Bluetooth connection to remotely control the robot. We have a HC-05 Bluetooth module. This module utilizes a serial connection to connect to the ATmega328P chip. The Bluetooth typically communicates over a hardware implemented serial bus, but we utilized a specific library called SoftwareSerial that allowed for communication over standard input/output pins and allowed software to emulate a serial bus. When in the main loop of the program, the Bluetooth connection is polled to see if we are receiving any new data. If so, we take the appropriate action. The module is powered by a 5-volt connection on the PCB. This communicates at a default baud rate of 9600 and can be configured to be a master or slave. We use it in the master mode. We have encountered some issue with the connection stability, but we will talk more about that in the mobile application section.

We will be using a Bluetooth connection to remotely control the robot. We have a HC-05 Bluetooth module. This module utilizes a serial connection to connect to the ATmega328P chip. The Bluetooth typically communicates over a hardware implemented serial bus, but we utilized a specific library called SoftwareSerial that allowed for communication over standard input/output pins and allowed software to emulate a serial bus. When in the main loop of the program, the Bluetooth connection is polled to see if we are receiving any new data. If so, we take the appropriate action. The module is powered by a 5-volt connection on the PCB. This communicates at a default baud rate of 9600 and can be configured to be a master or slave. We use it in the master mode. We have encountered some issue with the connection stability, but we will talk more about that in the mobile application section.

4 Standards and Realistic Design Constraints

Standards and constraints are used to help guide the design and build process that engineers go through. This section will explore the various standards and constraints that are applicable to the technologies and processes used for this specific process.

4.1 Standards

Standards are used in the engineering world to try and make every engineer’s process of designing and building uniform across the world. This ensures that safety requirements and performance is consistent, repeatable and, ensures compatibility with equipment. Several organizations take charge in creating national, regional, and international standards such as the American National Standards Institute (ANSI), IEEE, and ASME. Below is a summary of relevant standards for this project.

4.1.1 PCB Standards

PCB standards are made by the international industry association formerly known as the Institute for Printed Circuits (IPC), now known as the Association Connecting Electronics Industries. It is an association with more than 4000-member companies that help make standards about PCB design, manufacturing, and electronic assembly. The American National Standards Institute (ANSI) accredits IPC to be able to establish international PCB standards. While there are many standards, they are mainly categorized into the following groups: general documents, design specifications, material specifications, performance and inspection documents, and flex assembly and materials standards. Since there are so many standards, they will not all be covered, but will be taken into account while designing and ordering custom PCBs for the project.

4.1.2 Python Standards

The python programming language, which is regarded as a popular high-level programming language, was a significant part of the overall SAFER Knights system. Since the main microcontroller unit was planned as a python-based system, it was more efficient to use the python language. Python typically follows certain standards, regarding different aspects of the language. In regard to style and formatting the python language typically follows the Python Enhancement Proposal (PEP) 8 style guide. The PEP 8 standard will be implemented in the writing of the python codebase of the SAFER Knights robotic system.

4.1.2.1 Code Format

The code formatting for the Python language is important for multiple reasons. Following the code format standard increases the readability of the code. This will allow for easier editing of written code as well as greater understanding of code written by others. Another importance of the code formatting pertains to the nature of the Python language. The Python language uses formatting not just for code readability but for code indexing. In other similar high-level languages, code is separated by symbols such as semicolons (;)

and curly brackets ({}), but in Python, much of code separation and indexing is handled by whitespaces, thus making code formatting an integral part of code execution versus other high-level languages.

Many times, during software development lines of code are wrapped to avoid long, hard to read lines of code. To do so, Python has standards to properly handle line indentation. The recommended ways of properly indenting lines are to either use vertical line indentation or hanging indentation. When using vertical line indentation, the arguments or conditional statements of each line were aligned with the arguments or conditional statements of the first line. Aligning all lines of arguments or conditional statements allows for the grouping of multiple of lines for proper indexing. Vertical indexing also allows for arguments or conditional statements to be passed on the first line of declaration. Hanging indentation allows for a single line to be partitioned onto multiple lines by equally indenting each line to a specific level. In contrast to vertical indent, hanging indent arguments or conditional statements cannot be declared on the first line and each line will be on an equivalent level of indentation. With hanging indentation, enough indentation is needed to make the hanging indentation a distinctive level. If the hanging indentation is not equal, or the indentation does not create a distinctive level the code will not parse and result in a compilation error. Regardless of whether vertical alignment or hanging indent is used, the indentation can be accomplished in different ways. The two accepted ways of indentation are either spaces or tabs. The recommended indentation by Python standards is using spaces. Another way for indentation is using tabs. The use of tabs is not recommended by Python but are accepted. Tabs are used predominantly in documents that already use tabs. Python 2 allows for the mixture of tabs and spaces for indentation although it is recommended that a mixed document is converted to space indentations only, while Python 3 no longer supports the use of both spaces and tabs for the use of indentation, which requires a document to be converted to only one type of indentation, spaces being the recommended option.

4.1.3 C Standards

The C language, a high-level language with the capability of low-level hardware manipulation. The C language will be used predominately with a Texas Instrument microcontroller unit in the SAFER Knights system. The C language has released standards on the language and have continuously updated such standards regularly. The current C language standard is considered C18 and previous versions include ANSI C, ISO C, C99, and C11. These standards include data type declarations, keywords, and storage declarations. Unlike the Python standards which include formatting standards, code written in the C language uses operators such as semicolons (;) and curly brackets ({}) to properly contain and index code. This makes whitespace and formatting less of importance.

The C languages utilize multiple keywords and data structures that it reserves for system use. For linkage the C language utilizes static and extern. The keyword static is reserved for an internal linkage of declared objects or functions. The keyword extern is reserved for objects and functions that are declared to be external. The extern keyword is affected by the previous declarations of the objects or functions. The C language also contains standard data types. These standard data types are preset in the language and used to store information of the declared type. These data types are some of the basis of the C language and are you extensively in all implementations of the C language. An item that is created with a _Bool data type can only store a TRUE (1) or a FALSE (0) value, a data type typically used in tandem with conditional statements and decision making. The Char data type can hold an object typically from the ASCII character set. The data held by a Char is going to be a nonnegative value as long as the value of the object is within the ASCII character set, if the value is not part of the ASCII data set it will still need to be within the bounds of the Char data type.

The Char data type can also be declared as a signed Char and unsigned Char, the specified signed Char utilized the same amount of space as a plain or unspecified Char object, all three Chars are identified as the character types. The C language features the int or integer data type. There are five different integer data types: signed char, short int, int, long int, and long int, where a plain int type can contain a value between the INT_MIN and INT_MAX values, these values are set in the limit.h file, which is set and dictated by the system the C code is being executed on. The signed integer data types will utilize either the state of the object's sign bit, one's compliment, or two's compliment to store the appropriate sign of the value. The integer types can also be preceded by the extended keyword, this will dictate if the object will be a standard integer type or an extended integer type. The C language also supports three floating point data types. The three floating point data types are float, double, and long double. The values that can be stored into a float is a subset of the values that can be stored into a double which is in itself a subset of the values that can be stored into a long double.

4.2 Realistic Design Constraints

Design constraints are limitations that an engineering project can face when it comes to the design and build of the project. Constraints can include factors such as ethical, environmental, political, manufacturing, economic, time, and safety. All these constraints must be kept in mind, as not adhering to them can cause the entire project to become pointless. Below are some of the constraints that must be acknowledged and followed for this specific engineering project.

4.2.1 Outdoor and Environmental Constraints

As this was a robot that will be operating outside, there were certain moving and environmental constraints attached. The robot had to be able to follow the user as they walk on sidewalks and streets to get to their location. The robot was unable to follow the user if they decide to go onto grassy areas or take stairs instead of a handicap ramp. Being located in central Florida, heat and rain also have to be taken into account when using this vehicle. Fortunately, since the robot will only be used during evening hours, the effects of heat will be reduced and less of a concern compared to daytime use. Rain is a bigger concern. Although the robot will be made to be as weather resistant as possible, it won’t be used it heavy rain. As long as the rain is light and there isn’t flooding, the robot was functional.

4.2.2 Economic and Time Constraints

As with most projects, money is a chief constraint on the build of the robot. A budget was made with a list of all parts necessary as well as their quantity and cost. This budget was agreed upon by all members of the team. Another chief concern was time, which was limited for this project. Research, design, building, and testing all happened in a matter of months. Time and money go hand in hand for this project, as certain parts can ease or lengthen the time spent on designing the robot. However, parts that reduce time might also increase money, so a delicate balance was struck between time and money to keep the project on budget but also still completed within the designated time frame.

4.2.3 Social and Political Constraints

To have the robot actively run on campus, consent would need to be gotten from the university and the campus police notified. Not everyone would be comfortable with using a robot to help them home, and if too many students didn’t like the idea of an autonomous robot on campus the robot might be barred from working on campus. Another issue was the unknown possible legal ramifications of having an autonomous robot capable of video recording.

4.2.4 Ethical, Health, and Safety Constraints

The goal of this project was to give students a feeling of safety and security at night, therefore it was important to not do anything that might take away from this goal. This robot was meant to provide surveillance and avoid harming others in any way. It is a following robot, so it always stayed a few feet behind the user and used both GPS via Bluetooth and object detection to avoid running into people or things at any time. When a student chooses to use the robot, they will have to agree to terms and conditions via mobile app to learn and accept possible ramifications of using the robot. This way users will know and accept that a robot will be following them and filming their surroundings at all times, and that the robot is not able to follow the user everywhere they go if there is an obstacle to the robot.

4.2.5 Other Constraints

The battery life of the robot was a major factor for how long the robot operated and how far it traveled. The robot needed an additional battery to extend its lifespan. The Power Wheels was also used and therefore, wear and tear was another concern on the lifespan of the robots. Failing parts needed to be fixed or replaced.

5 Project Hardware and Software Design

The SAFER Knights robot will have multiple systems working in tandem a system. The following block diagrams show the different aspects of the overall system including hardware, power, and the software of the robotic system. From there each diagram can be broken down into the individual subsystems and how they interact with each other.

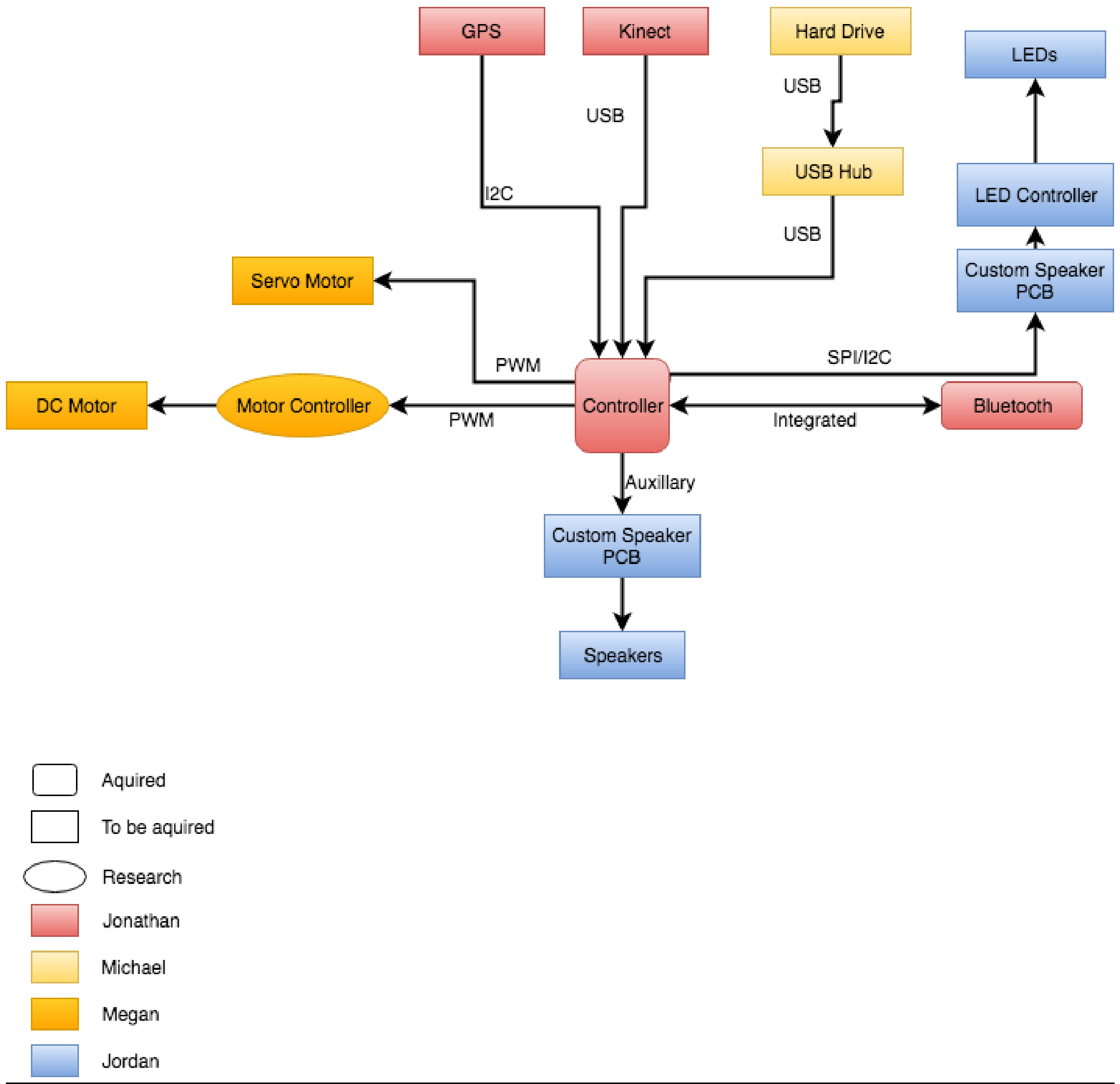


Figure: Initial Hardware Block Diagram

The overall hardware block diagram describes the hardware configuration of the entire SAFER Knights robotic system. It can also be broken into multiple subsystems with different roles in controlling and maintaining the system.

A master controller was needed to control all other subsystems. It controlled all movement and communication with the user and the robot’s surroundings. Instead of a microcontroller, the Raspberry Pi has been chosen to be the master controller of all of the robot’s subsystems. The Raspberry Pi is a small and affordable computer, which gives it more capabilities and robustness than the simpler microcontroller. This way, the robot can tackle all its various tasks such as communicating to a phone via Bluetooth and object detection via a Microsoft Kinect. Once the master controller has information on its location as well as the location of the user, it then tells the motor controller how fast to go as well as which way to steer. The master controller also tells the lights when to turn on and off, as well as plays any necessary audio. Since the Microsoft Kinect records video, the master microcontroller monitors the transfer of video to a memory storage device for playback. When the user triggers Emergency mode, signals will go to the appropriate subsystems. The LED lights flash, and audio warnings play. When the user turns off Emergency mode, the master microcontroller turns off appropriate subsystems to end Emergency model.

The master controller had plenty of room since a power wheels is being used as the base vehicle. It was attached to the vehicle in a way that prevents it from possible damage via outside conditions. As the vehicle was made for evening use, overheating from outside lighting was not a major concern.

5.1 Sensor/Microcontroller Unit Subsystem

The Sensor/Microcontroller Unit subsystem is one of the most essential subsystems of the overall SAFER Knights system. The microcontroller unit is not only the main component that incorporates all the other subsystems together but also handles all sensor data to appropriately make decisions that affect the other subsystems. The Sensor/Microcontroller Unit subsystem consists of the main microcontroller unit, a GPS module, a Kinect sensor, an external hard drive, a USB hub, and a Bluetooth module. The microcontroller handles all calculations and decision making of the SAFER Knights system. The Bluetooth module, which will come integrated in the microcontroller unit, will allow the microcontroller unit to communicate data with an external mobile device. The microcontroller unit will utilize the Bluetooth connection to constantly receive GPS coordinate data. The microcontroller unit processes this GPS coordinate information and then utilizes the GPS module. The GPS module gives the microcontroller unit the GPS coordinate data of the robot itself. From there the microcontroller uses the two GPS data sets to determine the signals it needs to send to the motor controller and servo motor.

The microcontroller also utilizes the Kinect sensor. The Kinect sensor can be broken down into three different sensors, a camera, a microphone array, and an IR depth camera. Using the Kinect sensor, the microcontroller determines if there are obstructions in the path and adjust the signals sent to the motor subsystem to follow the path dictated by the GPS information while avoiding the obstruction. Further the Sensor/Microcontroller unit utilizes the Kinect sensor to record video which may be saved to the external hard drive. The external hard drive was connected to the microcontroller unit via a USB hub.

The purpose of the USB hub was to provide power to the external hard drive from another source other than the microcontroller unit itself as the external hard drive will require more power than the microcontroller unit can provide. The microcontroller unit also utilized the

Bluetooth module to control the activation and deactivation of the emergency mode. Since the external mobile device stands as the user interface of the SAFER Knights system the microcontroller unit uses the Bluetooth module to receive the emergency mode flag from the user. When the emergency mode flag is received the microcontroller unit will then send a signal to the LED subsystem to start/stop emergency mode and start/stop the emergency mode audio sent to the speaker subsystem.

The Kinect sensor can be broken down into three different sensors, a camera, a microphone array, and an IR depth camera. Using the Kinect sensor, the microcontroller can determine if there are obstructions in the path and adjust the signals sent to the motor subsystem to follow the path dictated by the GPS information while avoiding the obstruction. Further the Sensor/Microcontroller unit will utilize the Kinect sensor to record video which will be saved to the external hard drive. The external hard drive will be connected to the microcontroller unit via a USB hub. The purpose of the USB hub will be to provide power to the external hard drive from another source other than the microcontroller unit itself as the external hard drive will require more power than the microcontroller unit can provide. The microcontroller unit will also utilize the Bluetooth module to control the activation and deactivation of the emergency mode. Since the external mobile device will stand as the user interface of the SAFER Knights system the microcontroller unit will use the Bluetooth module to receive the emergency mode flag from the user. When the emergency mode flag is received the microcontroller unit then sends a signal to the LED subsystem to start/stop emergency mode and start/stop the emergency mode audio sent to the speaker subsystem.

5.2 Motor Subsystem

An important part of the SAFER Knights overall system is the motor subsystem. The motor subsystem was responsible for controlling all motor functions on the robot including the overall movement and steering of the system. The motor subsystem consists of two motors and one motor controller. The motors consist of a DC motor that was responsible for the movement of the robotic system and a servo motor that was used for the directional control of the robotic system. The DC motor requires a motor driver to be able to control the speed of the motor. The motor controller regulated the current that was passed to the motor to control whether or not the robot was moving, the speed of the motors, and the direction the motor spins based on the input provided by the microcontroller unit. The servo motor requires a separate motor controller since servo motors typically contain an integrated motor controller. The servo motor had a PWM connection that was plugged directly into the microcontroller unit, from there the microcontroller unit was able to control the servo motor’s rotation and speed, which allowed the robot to make appropriate turns.

The rear wheels of the Power Wheels are the driving force for the car. Originally, the Power Wheels had one speed mode that was enabled when the pedal was pushed down. Two h-bridges were added between the batteries and the DC motors to control the power delivered to the back motors. The h-bridges receive signals from the custom PCB in the front of the vehicle and will alter the speed of the motors when necessary.

5.2.1 Rear Wheel Drive

The rear wheels of the Power Wheels are the driving force for the car. Originally, the Power Wheels had one speed mode that was enabled when the pedal was pushed down. Two h-bridges were added between the batteries and the DC motors to control the power delivered to the back motors. The h-bridges receive signals from the custom PCB in the front of the vehicle and will alter the speed of the motors when necessary.

5.2.2 Steering

A major design challenge came in converting the manual driving to an autonomous steering system to allow the robot to turn on its own when needed. The steering wheel was rid of and holes were drilled into the axle to allow for the new autonomous set up. A 12 Volt Gear Head DC Motor is attached to the bottom of the vehicle and can rotate its axle clockwise or counter-clockwise depending on the signal received from an h-bridge connected to the custom PCB. As the DC motor axle rotates, a gear on the axle moves in a linear manner to turn the car’s axle, which is connect to the moving gear, either left or right as needed. In addition, a 10 kΩ sliding linear potentiometer is mounted to the bottom of the vehicle to track where the moving gear is. The potentiometer sends a signal back to the microcontroller on the custom PCB between the values of 0 to 1023, and the code knows which values mean that the wheels are turned right, left, and center. With this, an RC Mode can run where the user can simply tap to command the robot to go left, right, and forward.

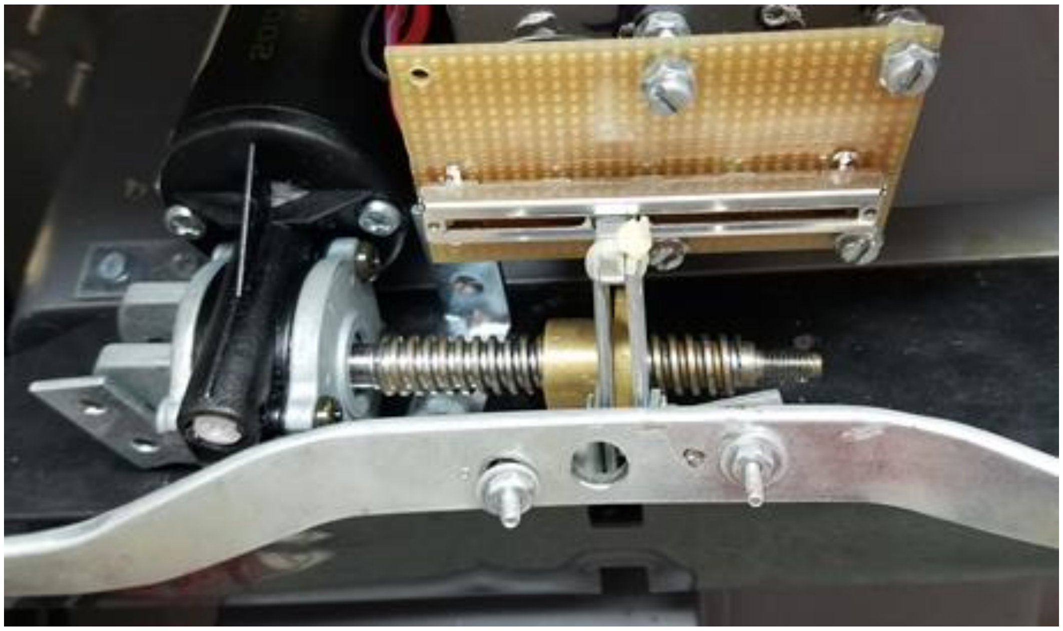


Figure: Steering Mechanism with mounted 12 V DC Motor and Sliding Potentiometer

5.3 LED Subsystem

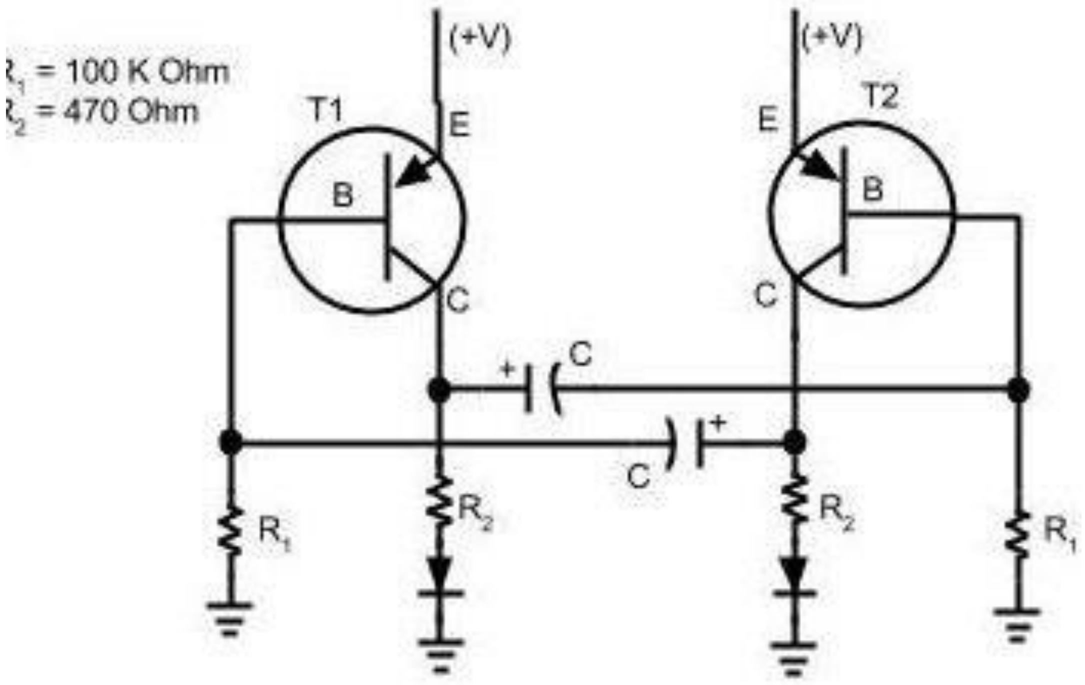
The LED subsystem is responsible for controlling all necessary actions related to the LEDs on the SAFER Knights system including initial powering of the LEDs to appropriate actions related to the SAFER Knights emergency mode. The LED subsystem consists of multiple LEDs, LED controllers, and a custom PCB that was designed and manufactured to control the LED controllers. The LEDs consist of standard LED strips/bulbs, these LEDs were connected to the LED controllers. The LED controllers then controlled the LEDs, initially the controllers power on the LEDs, when necessary it is the responsibility of the controllers to properly control the LEDs for the emergency mode function based on the commands received be the custom LED PCB. The custom LED PCB board hosts a microcontroller unit that directly controls the LED controllers. The custom LED PCB board, on start, makes the LED controllers power on all the LEDs. The custom PCB board maintains the constant state of the LEDs until either the mode changes or the SAFER Knights system returns to standby mode. When the microcontroller unit sends the emergency mode signal to the custom LED PCB board the board then starts the emergency mode actions for the LEDs. The custom LED PCB board uses the LED controllers to alternate the state of the LEDs to cause them to flash. The custom LED PCB board continues to flash the LEDs on the SAFER Knights system until the microcontroller unit sends another signal indicating the end of the emergency mode or until the robot enters standby mode, in which the LEDs turn off and return to the normal mode upon waking up from standby mode.

The following two pictures illustrate the different LEDs that were ordered and will be used in the SAFER Knights project:

Figure: Hit Lights LED Strip Lights

Figure: Searchlight Portable Handheld LED Flashlight

5.3.1 Reference LED Design



(Instructables. "Simple Blinking LED Circuit." *Instructables.com*, Instructables, 10 Oct. 2017, www.instructables.com/id/Simple-Blinking-LED-Circuit/.)

Figure: Reference LED Designs

The above components are as follows:

- R1 = 100k Ohms
- R2 = 470k Ohms
- C = 10 microFarads
- 2 PNP Transistors

The basis of the design is dependent on the “charging” and “discharging” of the capacitors. Essentially, the capacitors will turn the bipolar transistors on and off. Since the LED lights (modelled as diodes above) are controlled by the current flowing from the transistors, they turn on and off as well. If you would prefer the lights to blink faster or more slowly than one could simply change the value of the capacitors.

5.3.2 Initial LED Design

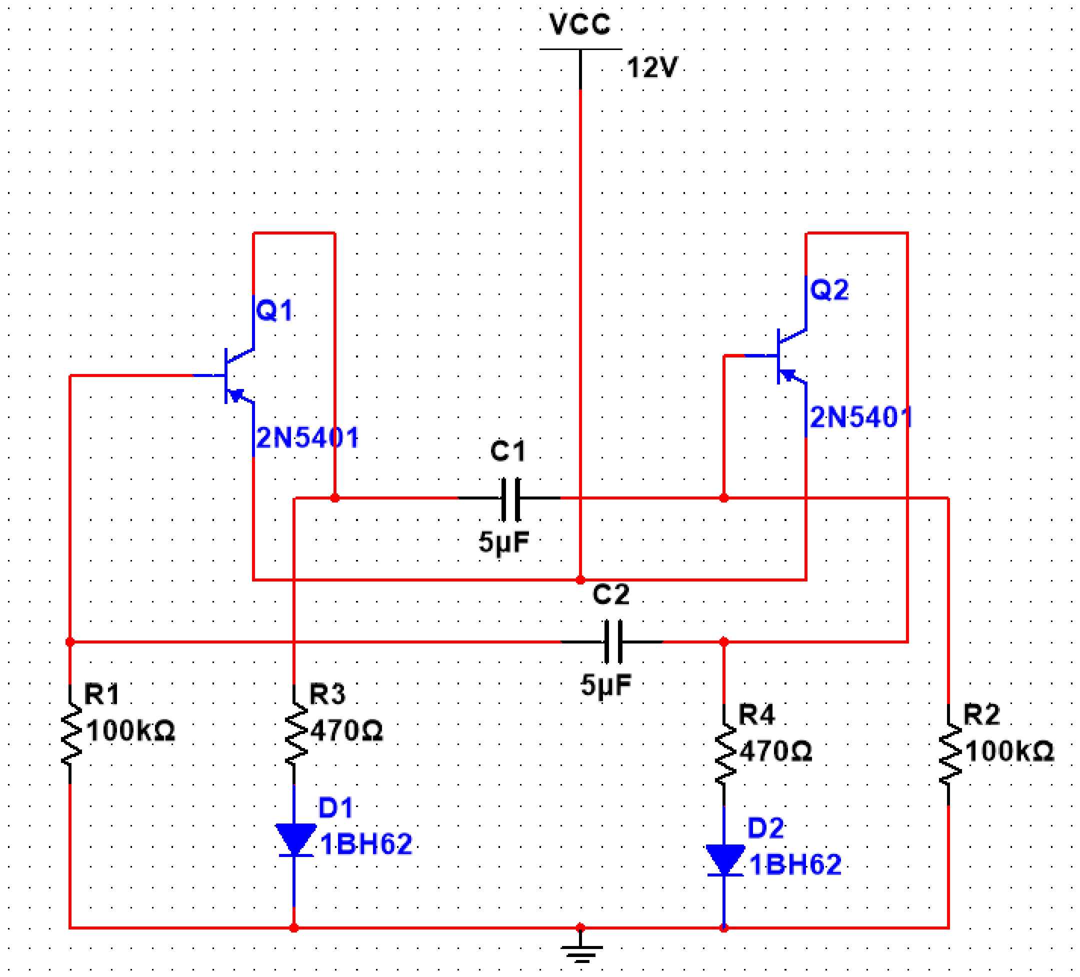


Figure: LED Design

The above components are as follows:

- R1 = R2 = 100k Ohms
- R3 = R4 = 470 Ohms
- C = 5 microFarads
- 2 PNP Transistors

Figure 5.3.2.1, above, is an option for the initial design of a circuit to make the LED lights flash. The main takeaway from the design is the capacitor values. The specifics of how these values affect the desired output is described in the reference design section. Based on some basic principles, it is theorized that the lower capacitor values will make the lights blink faster.

Some things that may need to be changed in future designs could include:
- The supply voltage
- Capacitor values
- Resistor values
- Bipolar transistor models
- Diode models

The supply voltage was 12 volts, but it is possible a separate power source would be needed or even a voltage regulator used to step down the supply. The voltage or current required by LED lights varies with respect to the specific lights being used. Specification sheets corresponding to the lights used give all necessary information and the components were be adjusted from there.

Component values are also things easily controlled and adjusted. Resistors in this circuit are used to regulate the voltage being supplied to the lights. Without these resistors, the lights would burn or even not turn on. The other components in this system are capacitors. The values would only be changed to adjust the speed of the flashing lights. Either they were increased or decreased depending on the desired outcome.

The Bipolar transistors are also easily changed. Transistors control the current being distributed throughout the above system. Because different models have varying maximum collector currents and beta values, it is important to take this into account for future designs. Due to the mentioned variations, each model creates a different amount of current. The desired information was found on corresponding data sheets to the transistor being used.

Finally, the model of the diode may be changed. In reality this component is used to model the LED. Therefore, it is important to choose a diode that can accurately depict the voltage and current being received by the lights.

5.3.3 Final LED Light Designs

Spotlights are the first set of lights used in the project. The spotlights are mounted, one on each side of the vehicle, in order to provide light in the forward direction. A separate battery pack, charging circuit, and relay complete the spotlight's subsystem as shown below. When the relay is switched on, it connects the positive wire from the battery to the positive wire of the spotlights. When the relay is off, the switch in the relay is creates an open circuit.

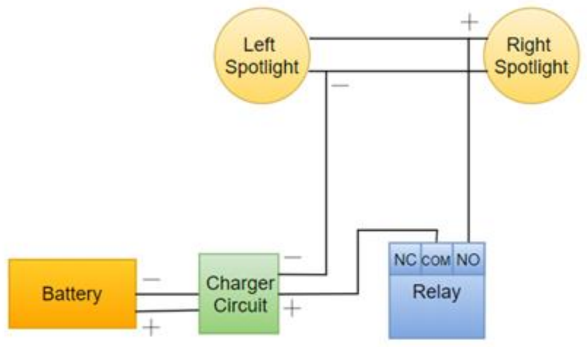


Figure 4: Spotlight Circuit with Relay

The relay is controlled through a PWM signal to the MCU for the purpose of allowing the spotlights an option to flash when emergency mode is activated. The other lights used in the project are 12 Volt LED strips. Due to the ability to cut the strips to any length necessary and the practicality of mounting, they were clearly the best option in providing light to the sides and rear of the vehicle.

5.4 Speaker Subsystem

The SAFER Knights system utilized a speaker to appropriately play necessary audio. The speaker subsystem consisted of speaker and a custom speaker PCB board. The speaker was connected the custom speaker PCB board and played the signal that it receives. The custom speaker PCB board acted as an amplifier. The custom speaker PCB board received an audio input from the microcontroller unit via an auxiliary connection. When received, the custom speaker PCB board then modulates and processes the signal and translate it to the speakers to be played.

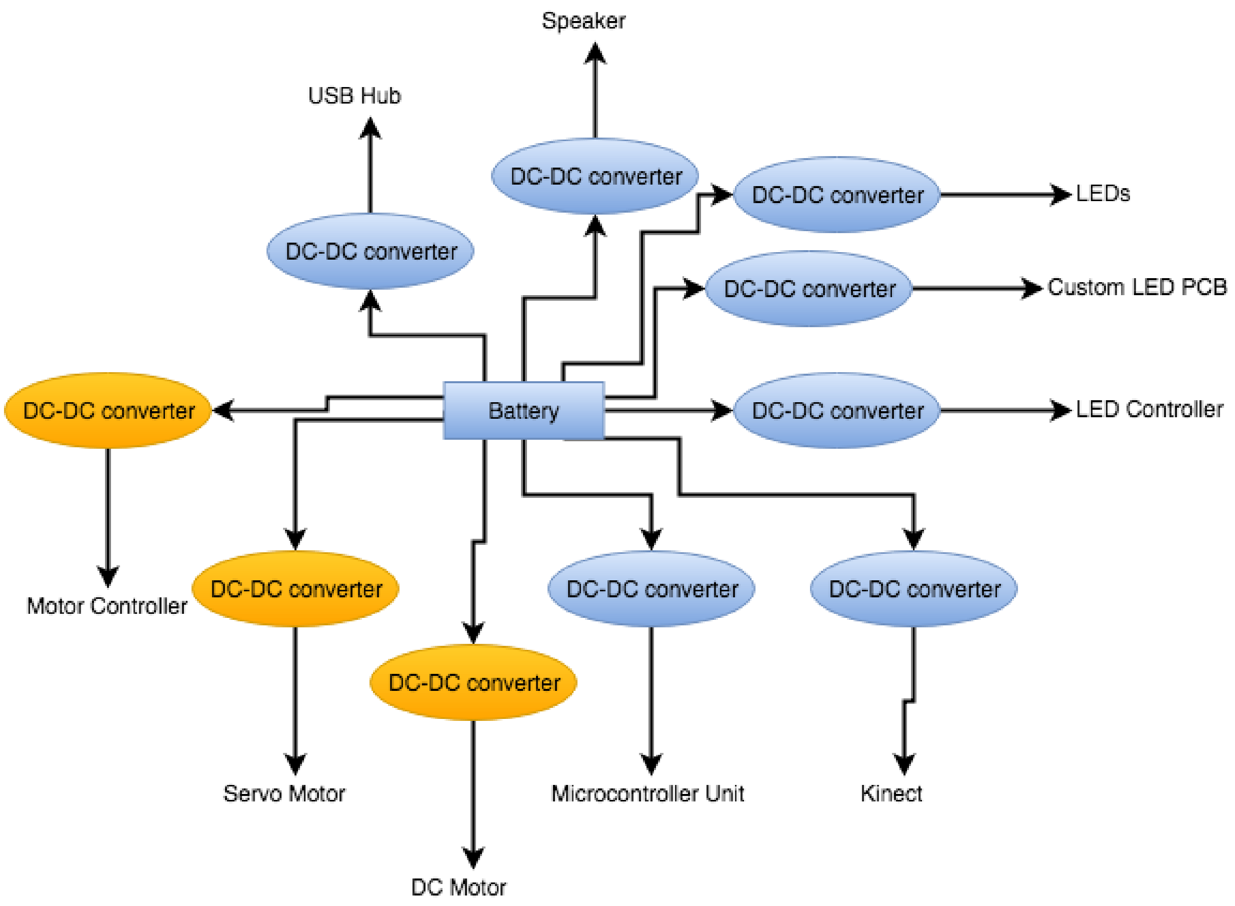


Figure: Power Block Diagram

Figure: 8 Ohm Speaker

The speaker was also ordered. The picture directly above depicts the speaker that will be used in the SAFER Knights project.

5.4.1 Speaker Reference Design

In the table below is the design researched and developed to form the initial speaker design used in the SAFER Knights project.

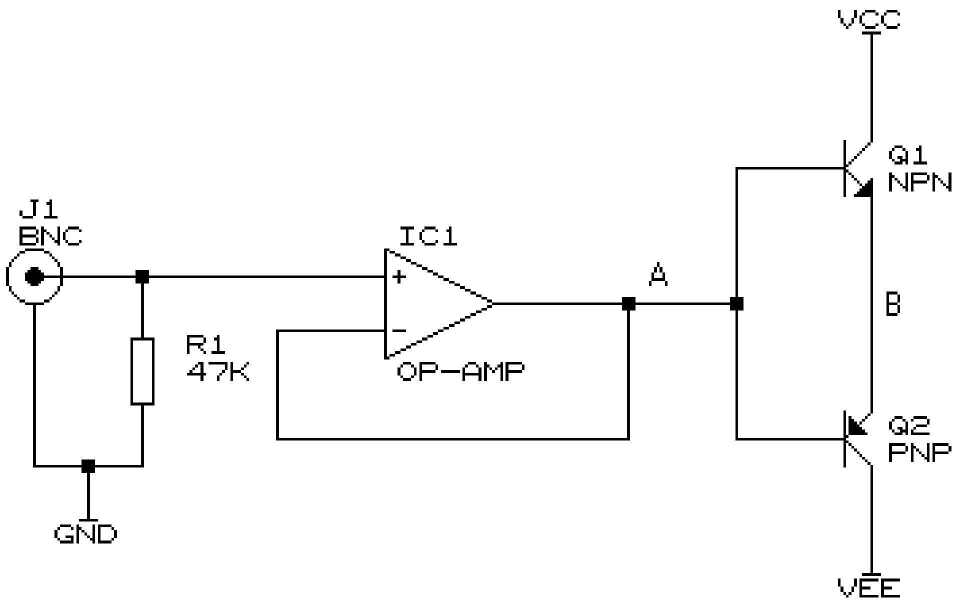


Figure: Initial Speaker Design

In the figure you can see a single unity gain op amp with the output connected to a push pull output stage. This design a very basic one that could be applicable to the SAFER Knights project if some important changes are made.

5.4.2 Speaker Design

One of the first things needed to play audio through a speaker is the audio file itself. This is one of the things done by the raspberry pi board. The file is stored on the pi and an extension was added in order to allow the board to play the audio file. The pi then becomes the input of the custom PCB.

The Visatron speaker is a passive speaker, therefore it needs the custom PCB to work. An amplifier and output stage provide the necessary power for the speaker. Since the load impedance is 8 ohms and the rated power is 20 watts, the initial design supplies roughly 6 Watts to run the speaker. A combination of ohm’s law and the power equation prove the voltage supplied should be approximately 7.5-8 volts. The approximate current should be 0.9-1 Amps.

The voltage output of an audio jack is about 1 volt; therefore, a voltage amplifier is needed. Ideally, op amp would be used to apply a gain of about 6-8 to achieve the desired voltage. The problem then becomes the current from the output of an op amp is very small. In order to overcome this, an output stage composed of MOSFETs or BJTs must be used.

One set of MOSFETs or BJTs would not create a sufficient output current. In order to correct this, there must be another set put in parallel. The choice then becomes whether it would be best to you BJTs or MOSFETs.

The speaker is mounted underneath the hood of the vehicle with holes drilled into the opposite side for improved audio quality. An audio amplifier was necessary to power the speaker and therefore needed to be implemented on our custom PCB. Since we can store audio files on the Pi, it is possible to play multiple files whenever we chose to do so. A 9 mm audio cable connects the Pi to the custom PCB and another one goes from the PCB to the speaker as seen below.

5.4.3 First Speaker Design

Previously, it was stated the speaker would need from 7-8 volts and .9-1 amps to run effectively. Below, in figure 5.5.x.y, is a visual representation of the initial design used to power the Visatron speaker.

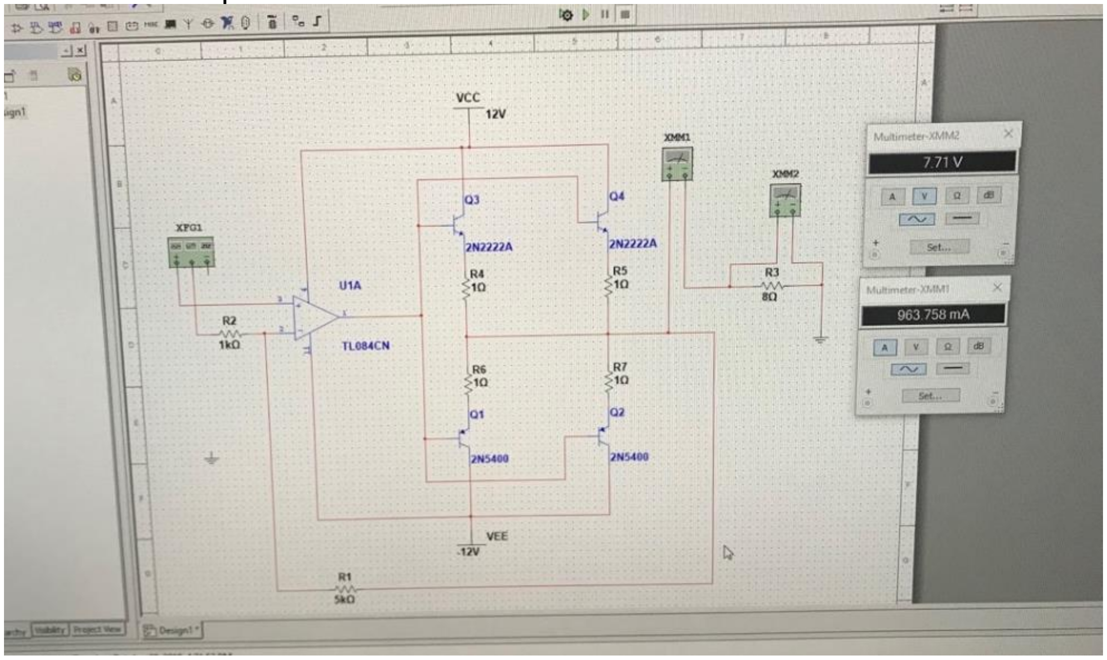


Figure: Speaker Design

5.4.4 Reference Design and First Design Comparison

The reference design, seen above in Figure 5.5.1.1, is a single op amp with only one pair of NPN and PNP bipolar junction transistors. This exact design would not provide the necessary power to run the speaker.

The first change necessary was addressing the unity gain op amp. Since the input of the op amp does not supply the appropriate voltage, there was no way this gain would suffice. There must be enough voltage gain to support the load. This gain was said to be somewhere between 6 and 8.

The next change made for the initial design was regarding the bipolar junction transistors in the output stage. One pair would not be able to provide the necessary current. This was a problem that could easily be fixed by placing another set of NPN and PNP BJTs in parallel with the first set. To place the BJTs in parallel, the initial design connected the emitter, base and collector of each NPN and PNP transistor. If more current is needed in future designs, another set can be placed in parallel or a higher gain can be made in the op amp stage to increase the voltage.

Another apparent change from the reference to the initial design was the resistors placed in the emitter of each transistor. These resistors serve to stabilize the voltage across each base and emitter. In theory the emitter resistors would not be needed, but without them the transistors will quickly burn. If they were to burn, the circuit would not work correctly. One other thing you may notice about the resistors is they are very small. If they were bigger, it could force the transistors to be unable to reach their “turn-on voltage”. This would also render the circuit useless. Therefore, they were designed to be have just enough resistance to balance the base-emitter voltage and keep the transistors on.

5.5 Power Subsystem

The power block diagram and the hardware block diagram go hand in hand but for simplicity sake the two diagrams are displayed separately. The power diagram dictates the overall power distribution of the SAFER Knights system. Each component in the SAFER Knights system that requires an external power source will also need a DC to DC power converter. The DC to DC power converters will convert the voltage and current received from the battery into the appropriate voltage and current needed by the component. Further the DC to DC power converters will also regulate the current, this will allow a constant current to be delivered to the attached components without the risk of a spike in the current damaging any of the components in the SAFER Knights system. The microcontroller unit will be powered of a DC to DC converter connected to the battery. Even though the microcontroller unit can deliver some power to connected components, it is limited on how much power it can produce, due to this fact multiple components that are connected to the microcontroller unit is also connected to their own power source.

The Kinect sensor and external hard drive requires more power than the microcontroller unit can produce to run efficiently thus they need to be connected to an external power source. The Kinect will be connected directly to a dedicated DC to DC converter to receive the appropriate power. The external hard drive on the other hand cannot be attached directly to an external power source. Due to this fact the SAFER Knights system will include a USB hub which will be powered with an external power source. Since the USB hub is powered from an external power source it can handle the power requirements of the external hard drive but still allow for the exchange of information over the USB bus without drawing power from the microcontroller unit's USB port. The motor controller, DC motor, and servo motor need to be powered by an external power source as well. Even though the servo motor is directly controlled by the microcontroller unit, the power draw over the PWM connection would be too great, because of this the servo motor's DC connection will be connected to a DC to DC power converter. The DC motor also will require an external power source but instead of it directly connected to the DC to DC converter the connection will go through the DC motor controller. The DC motor controller will accept the constant current from the DC to DC converter and adjust the current that is being delivered to the DC motor based on the signals received from the microcontroller unit to control the speed and direction of the motor. The DC motor converter on the other hand will have another direct connection to a DC to DC converter to power itself, while a connection to the microcontroller unit controls it.

The custom LED PCB board is in itself a separate smaller secondary microcontroller unit, this unit will need to be powered by a power source thus it is connected to another DC to DC converter. Similar to the DC motors the LEDs will also be powered by a DC to DC converter, but the connection will be wired through the LED drive in order to control the power that is being delivered to the LEDs which will allow for controlling of the LED state. The LED controller will also need a power source to operate, receiving its signals from the custom LED PCB board. The final component to be powered is the speaker. The speaker requires power to play the signal received from the custom speaker PCB board. The custom speaker PCB board does not need to be powered as the board only requires the auxiliary input from the microcontroller unit, which it will process and deliver to the speaker.

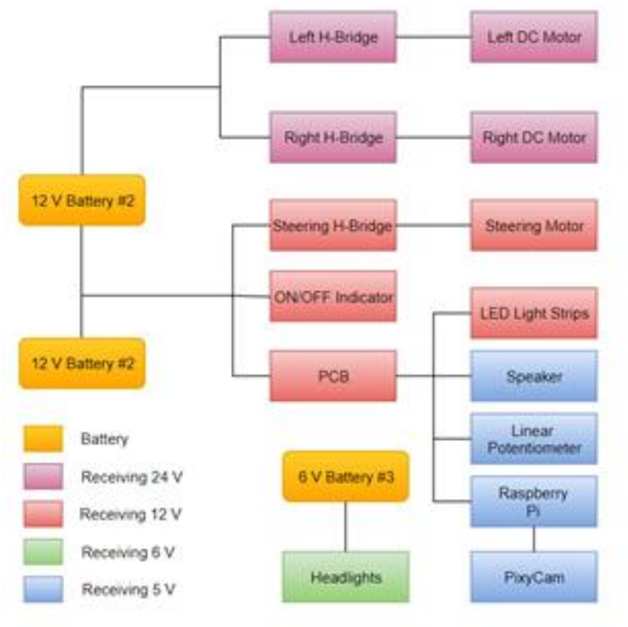


Figure. Power Diagram

The two main sources of power are a pair of 12 Volt batteries. These batteries are in series and provide 24 volts to the rear motors and 12 volts to the custom PCB, LED strips, and steering motor's H-bridge. 5 Volt regulators, on the PCB, step down the 12 volts from the battery. This 5 Volt supply then powers the Raspberry Pi, Bluetooth, MCU, and potentiometer; as seen by the PCB layout. Additionally, power is then drawn from the Pi to the PixyCam. Lastly, the headlights subsystem consists of a battery pack, relay, and charging circuit.

5.6 Software Design

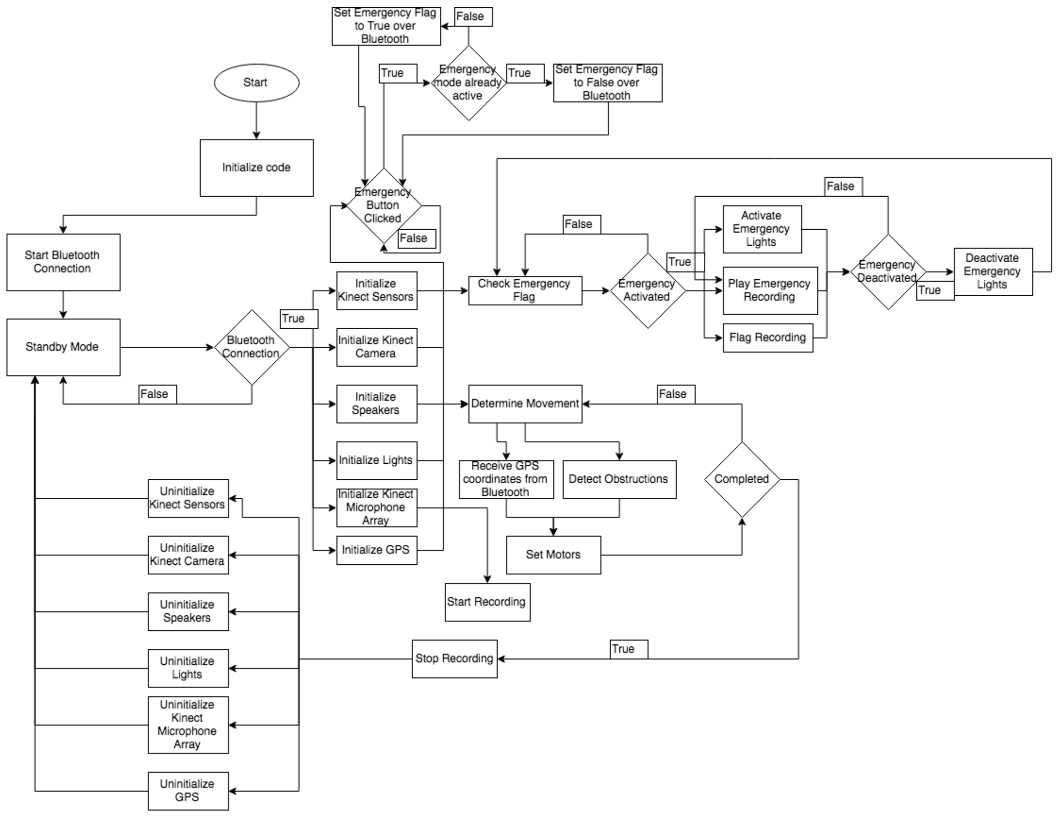


Figure: Software Block Diagram

The software block diagram illustrates the functions and processes that the SAFER Knights system will take to from start up to task completion. Each part of the diagram displays a different point in the overall process of the SAFER Knights system and how all the individual processes interconnect to make up the overall system. The diagram is largely made up of loops and conditional statements to determine and hold states until the necessary conditions are met for the furthering of the process. Due to the repetitive nature of the software, after initialization the overall system itself can also be considered a loop, which is constantly waiting for the starting condition.

There are a lot of different types of software running the robot. The piece that the user will interact with is the Mobile Application, but there is also custom software that is running on the ATmega328P and Raspberry Pi. These three components communicate in various different ways and have different roles.

5.6.1 Start

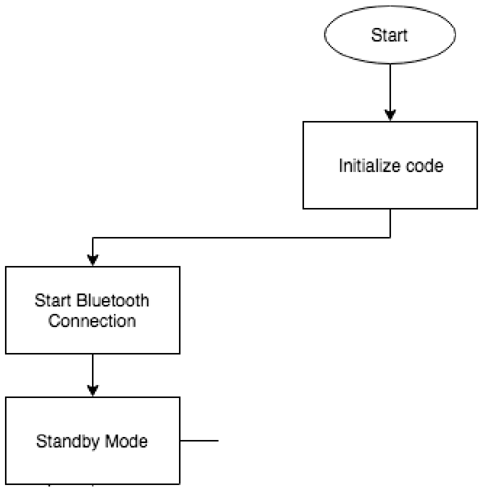


Figure: Start

When the SAFER Knights system is initially powered on the system automatically loads the software to be ran. When the software is loaded its first task is to initialize the Bluetooth module, this will allow an external mobile device to connect to the SAFER Knights system, which will later be used for further initialization. Once the Bluetooth module has been set up and configured the program will enter standby mode, in this mode no further component initialization occurs nor does any other processes other than Bluetooth connections occur. The reason for this is that during standby mode the SAFER Knights robot should conserve power as there are no necessary processes that need to occur. This will lead to longer active time and better battery management.

5.6.2 Initialization

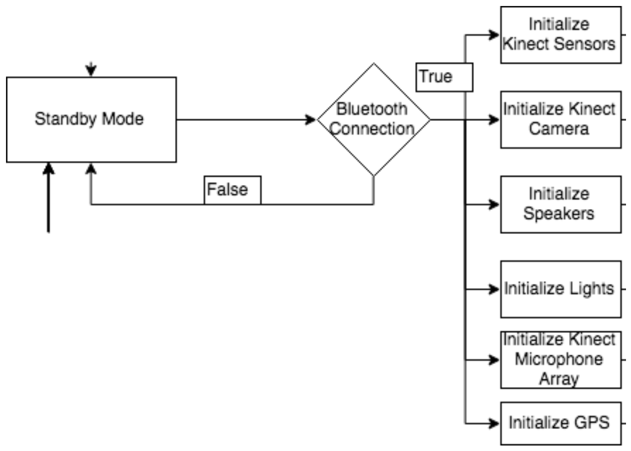


Figure: Initialization

Once the SAFER Knights software enters standby mode it waited for a Bluetooth connection, if no connection is yet established it stays in standby mode. When a Bluetooth connection is established then the program starts with the initialization process. Now that a Bluetooth connection is established the SAFER Knights system needs to prepare for use. The software leaves standby mode and initializes all the sensors and components which include the Kinect (IR Sensors, Camera, and microphone array), speakers, LEDs, and GPS module. When initializing the LEDs, the microcontroller unit will send an initialization signal to the custom LED PCB board that will cause the LED's normal operation mode to start. With the components initialized the program will now start the three main processes: path determination and movement, emergency mode flagging, and emergency mode initialization.

5.6.3 Path Determination and Movement

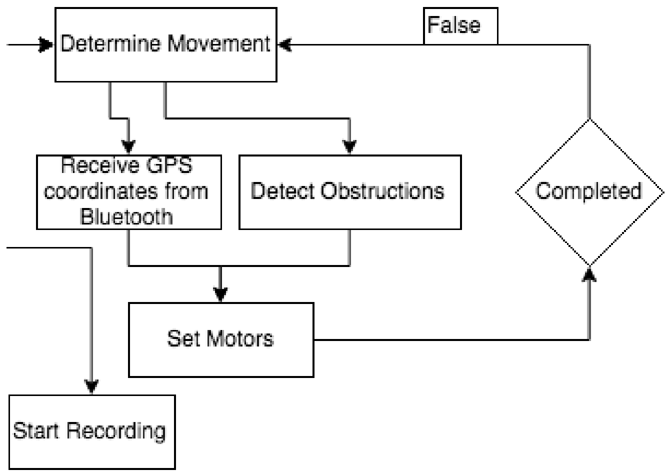
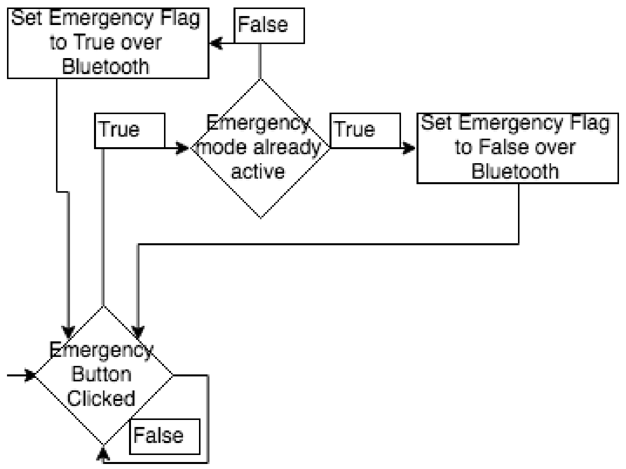


Figure: Path Determination and Movement

With all the components of the SAFER Knights system initialized the main process of path determination and robot movement is started as well as the video recording, which is stored on the external hard drive. The system will call the function to determine the movement path that needs to be taken, to determine said path the function must receive information from two different sources, the GPS information and the Obstruction detection. The GPS information of the target will be received from the external mobile device over the Bluetooth connection while the GPS information of the robot is determined by the GPS module. With the GPS information received the process can now determine a path from the current location to the target location. Now that a path is created it needs to be determined if there is an obstruction in that path. Using the Kinect sensor, the robot will determine whether or not there is an obstruction, how large it is, and how far it is. With this information the path received by the GPS can now be altered slightly to reflect the required object avoidance allowing for an unimpeded path. Once the path has been determined the movement needs to take place. The function will then send the appropriate signals to the motor controller and servo motor, moving the robot and following the path. Once the movement takes place the system will determine if the completed flag has been set, this occurs if the user selects complete on then mobile device UI, if not the process repeats until the flag has been set, if so then the process will continue to deinitialization.

5.6.4 Emergency Mode Flagging


Set Emergency Flag to True over Bluetooth
False
True
Emergency mode already active
True
Set Emergency Flag to False over Bluetooth
Emergency Button Clicked
False
Figure: Emergency Mode Flagging

While the SAFER Knights runs the path determination and movement it is also running the emergency mode flagging subroutine. This process will be used to determine when it is necessary to switch the emergency flag status of the system. The program checks and determines if the emergency button clicked signal is received from the Bluetooth connection, if not the subroutine will continue to wait until the signal is received. Once received the software will determine the current status of the emergency flag. If the emergency flag is set to False then the process will set the flag to true, the vice versa also applies, if the process determines that the emergency flag is set to false then the flag will be set to true. From there the process will return to the state in which it is waiting for a signal from the Bluetooth module to change the state of the flag. The emergency flag will then be used to initialize the emergency mode.

5.6.5 Emergency Mode Initialization/Deinitialization

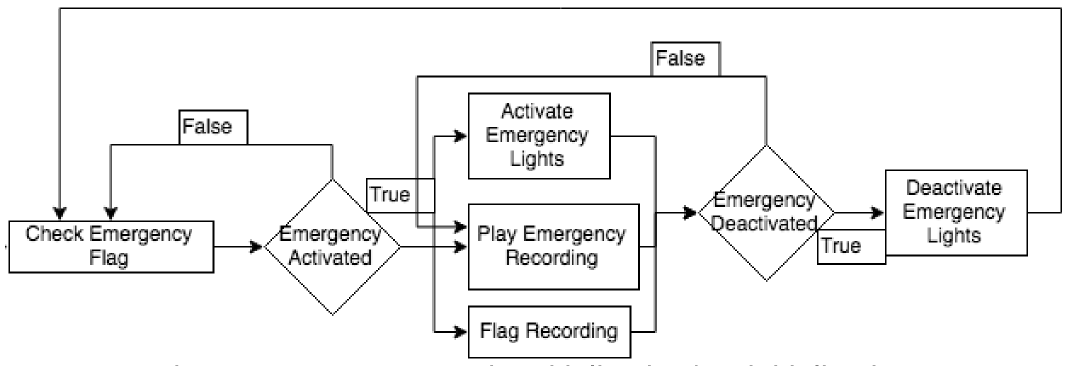


Figure: Emergency Mode Initialization/ Deinitialization

The emergency mode flagging process utilized the Bluetooth connection with the mobile device to determine whether or not to change the state of the emergency mode flag. The emergency mode initialization/deinitialization subroutine is the process that executed based of the emergency mode flag. The process starts by checking the emergency flag, if the emergency flag is not raised then the program will continue to check until the flag status changes. When the emergency flag is raised then the software will initialize the emergency mode. The microcontroller unit will send a signal to the custom LED PCB board to start the emergency LED mode. The custom LED PCB board will alternate the LEDs between the on and off state by alternating the signals it is sending to the LED controllers. The emergency mode also will flag the video recording by storing the video file data as a log file on the hard drive, this will allow anyone who goes back to review the recording to see which recordings are flagged and properly investigate. The emergency mode will also play a specified audio file as a deterrent. Once the initialization is complete the process will check to see if the emergency flag is no longer set, if it is still set the process will just keep repeating the audio file until the flag is cleared. Once the flag is cleared a signal is sent to the custom LED PCB board informing it to deactivate emergency LED mode and return to normal LED operations. Once the LEDs have been deactivated the process will return to the check flag state, monitoring whether or not the flag is set again. It is important to note that if the system returns to standby mode while in emergency mode, the emergency mode is deactivated, and the emergency mode flag is cleared.

5.6.6 Deinitialization

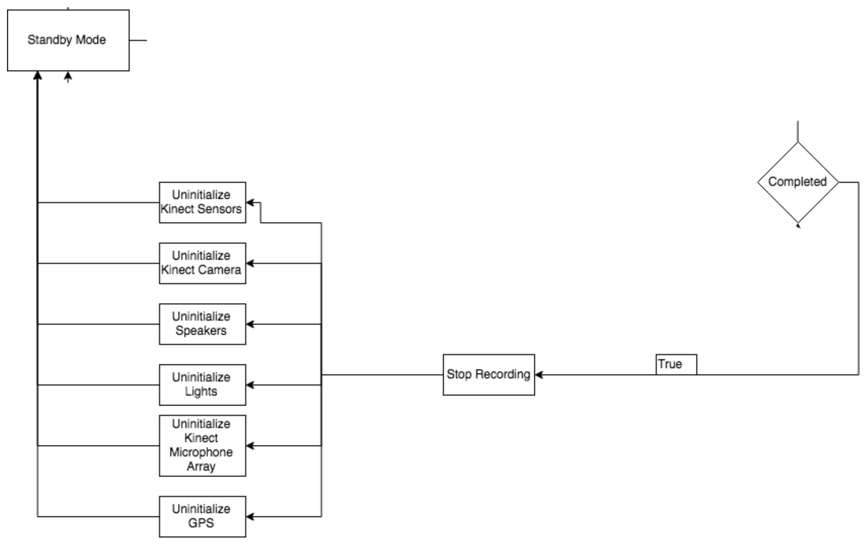


Figure: Deinitialization

Once the completed flag has been set by the user through the external mobile device UI, then the software will start deinitialization. Once the completed flag is raised the program will stop the video recording and close the created video file, this file will be available for later review as well as any log files created by the emergency mode. The program will then deinitialize the SAFER Knights' sensors and components including the Kinect (IR sensor, camera, microphone array), speakers, LEDs, and GPS module. To deinitialize the LEDs the microcontroller unit sends the deinitilize signal to the custom LED PCB board, from there the custom LED PCB board then sets all values for the LEDs to off, turning off all the LEDs until they are next initialized. Deinitializing the sensors and components will now clear the related values and flags associated those components for the next time the system is activated. With the components cleared the system will return to standby mode, where the process will wait for a new Bluetooth connection to start the process over again.

This was the initial software layout of the SAFER Knights system. Due to system changes and reconfigurations the system is handled differently. The system will no longer implement devices such as the compass, GPS, or Kinect. The system also no longer implements object avoidance. Instead the software streamed the data from the PixyCam and determine direction appropriately. The system also got flags from the Bluetooth when it was time to initialize different components rather than initialize on connection. The software for the ATmega328P and the Raspberry Pi follows these software diagrams respectively.

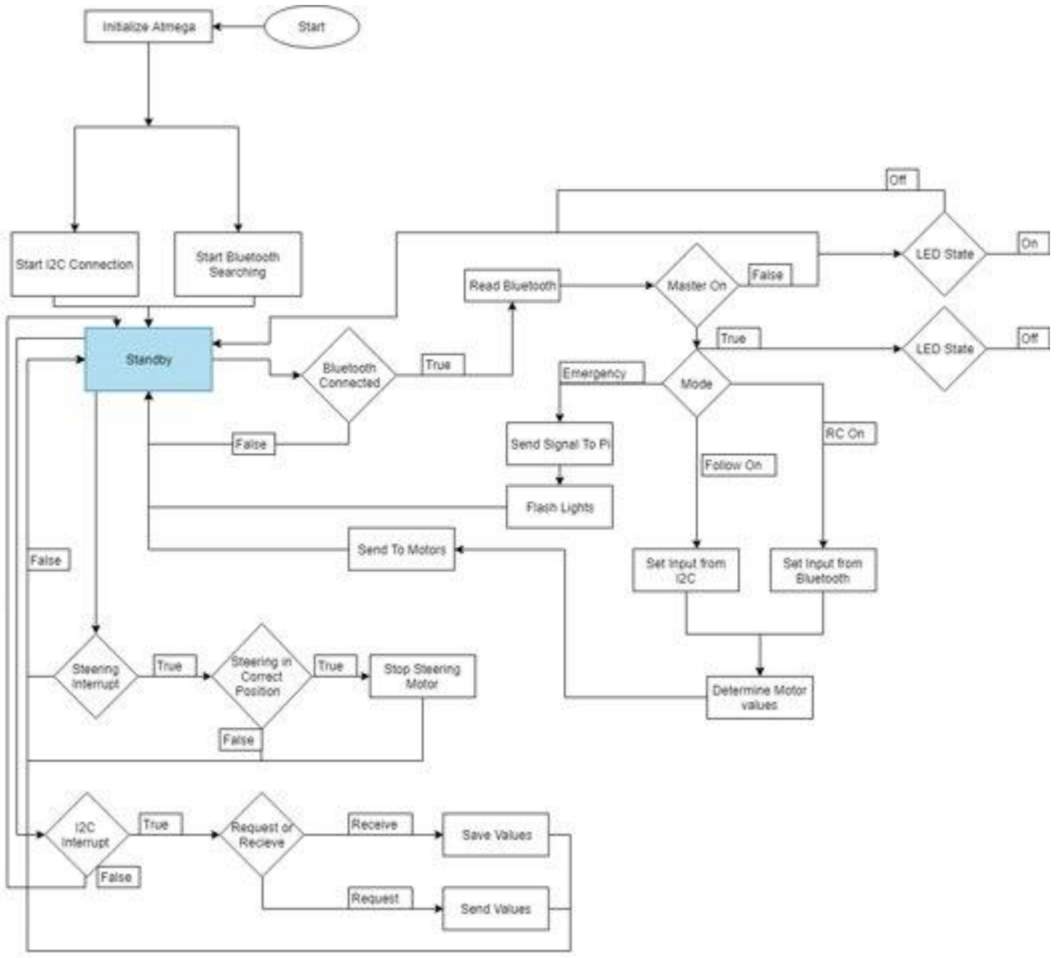


Figure: ATmega32P Software Diagram

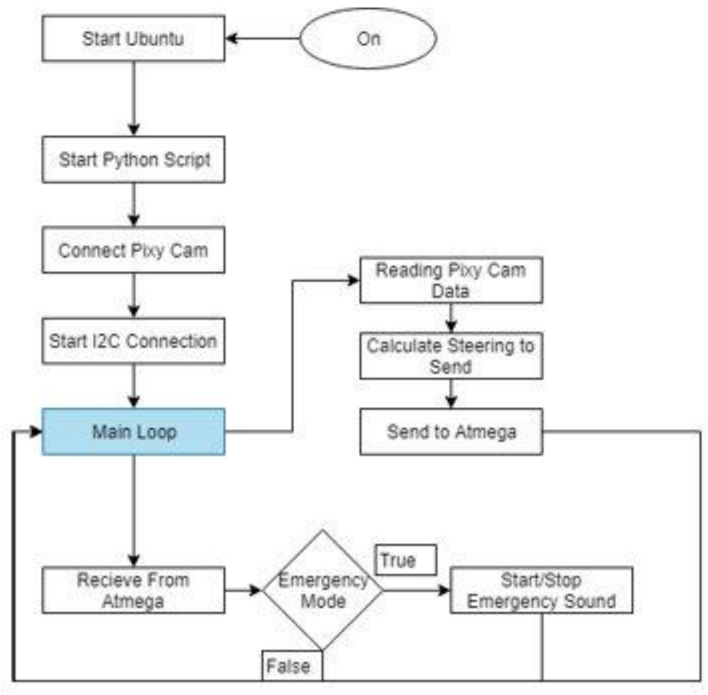


Figure: Raspberry Pi Software Diagram

5.6.7 ATmega328P (Robotics Controller)

The ATmega328P microcontroller is running custom code we wrote. This software is responsible for *controlling* the hardware like steering and the rear wheels. It also connects to the mobile application via Bluetooth and the Raspberry Pi via an Inter-Integrated Circuit (I2C) connection.

The code starts by initializing the connection to the phone and Raspberry and setting up all the variables used in the main loop. It also will get the proper pins for each of the connected devices and set if they are input or outputs. The robot has two rear motor controllers and one steering controller. All three of these are H Bridges controlled via pulse wide modulation (PWM). There is a Bluetooth module also connected to the ATmega328P.

Once the main loop starts it will first start by poling the Bluetooth connection. If the user has sent a command to the robot it will take appropriate action. For example, the user pressed the follow button, so it will be read in and the following function will initialize and start running. The connection with the Raspberry Pi is an I2C connection. Their communication is interrupt based. This means when the Raspberry Pi sends data to the ATmega328P chip it will stop what it’s doing to read that data. This is good for our operations because it means we will always be working off the most recent computer vision data. This connection will work both ways, so the ATmega328P code will send data to the Raspberry Pi when things like the emergency mode is initialized.

For the actual mechanical operation of the robot there are two modes. One where the user can remove control and one used for following. In the remote-control mode, the user will press button to issue commands that correspond to function in the code that steer or adjust the motor speeds. In the following mode the Raspberry Pi will send directional and speed data many times a second. The ATmega328P will read in this data and adjust its internal variables so that when the follow function is called it will change speed and direction appropriately.

To turn the robot, we have a linear potentiometer mounted to the bottom of the vehicle along with a motor that produces linear motion traditionally used to move car seats. As setting arm moves left and right it will also slide the potentiometer. As we are steering the left and right, we are also reading in the values of the potentiometer. This allows us to steer to a very precise degree. If we were not to have this linear potentiometer, we would not be able to know what the position the wheels are in and how to adjust them. This is one of the disadvantages of motors over servos. To engage the rear motors, we can just send their values between 0 and 255 and allow the PWM to do the rest. One issue that we have had to address is blocking while steering. That is, when the robot was issued a steering command it would be stuck in that loop unable to execute any other code until it was done.

This issue is compounded because the steering motor is not particularly fast, so by the time it achieved its goal seconds could have passed. This results in the robot drastically steering off path. To fix this issue we utilized timer interrupts. Utilizing the 16 MHz clock on the PCB, we can set up a timer interrupt to occur at a predetermined time offset. On a set amount of millisecond an interrupt is triggered, and it will check the current position of the steering against a global variable set by the follow or RC modes. If that position is met it will turn off the motor and if it is not met it will send a signal to move in the correct direction. This means instead of being stuck in a loop to check the steering position we are free to run other code. This makes the system much more real time. It has made a significant improvement on the operation of the robot.

5.6.8 Raspberry Pi (Computer Vison Controller)

The Raspberry Pi is the brains of the operation. The Raspberry Pi is running Ubuntu operating system with various software packages. The main responsibility of the Raspberry Pi is the computer vision. The Raspberry Pi is connected to a Pixy Cam, a powerful camera with robust opensource libraries used for computer vision. These two modules allow us to write some Python scripts to capture the camera information, process it and then determine how far the user is and if they are in the center, left, or right of the frame.

The user is tracked by an object that they hold or is pinned to them as they walk. This object is a colorful pattern of defined size. It is colorful, so it is easy to distinguish its pattern from other objects in the frame and it has a defined size so that we can calculate the distance to the robot based on its size in the frame. As the object becomes smaller, we know it is getting farther away.

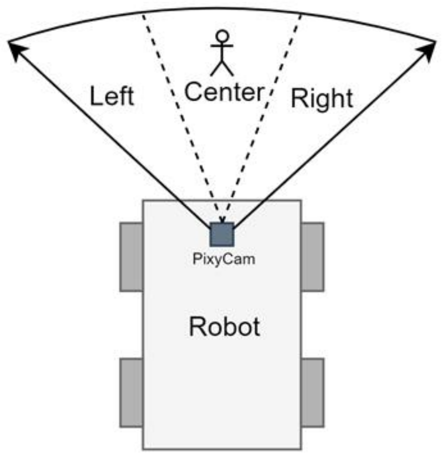


Figure: Robot Following Field of Vision

Each frame of video is analyzed to find the following object. Once it is found its size and placement is calculated. If the size of the object is within a defined rage the Raspberry Pi will tell the ATmega328P to speed up or slow down. The placement of the bottom right corner of the following object is marked and if it is within a defined rage it will tell the robot to turn. For example, if the following object is small and to the left of the frame the Raspberry Pi will tell the ATmega328P to speed up and turn left. The ideas situation is to have following object the save size and placement in the frame. This would be perfect following.

As stated, before the Raspberry Pi will send its data over an I2C connection. The Raspberry Pi will also get data over this connection. When the user engages emergency mode, it will raise a flag that will be sent to the Raspberry Pi where the Pi will play certain audio files on a separate thread.

5.7 Mobile Application

The SAFER Knights system utilizes data from different sources to achieve its overall goal, these systems include GPS, video, infrared, and Bluetooth. For the system to properly utilize the Bluetooth connection it needs a device to connect to that will feed it the appropriate data. To achieve this a mobile application will be required, one that will work on multiple stock devices. This mobile application will provide the additional data that the SAFER Knights system requires from the mobile device that it cannot receive by other means.

The mobile application will be required to interface with the SAFER Knights software and be able to provide and receive the appropriate data. Initially the mobile application will need to be able to verify whether or not it is connected to the SAFER Knights system. By determining the connection status, it will be able to provide the appropriate functionality to the user, and it will be able to verify that the any additional commands sent to the connected device will be appropriately received. Once connected the app will need to send a connection flag to the SAFER Knights system. This connection flag notifies the system that it needs to start the main functionality and initiate the appropriate subroutines.

Now that the SAFER Knights has been initiated the mobile app will need to send the GPS data from the hosted mobile device to system at a constant rate. Since the SAFER Knights pathfinding algorithm is based on the GPS location of the mobile device and the GPS location of the robot itself, the mobile application will need to send the GPS data at a faster rate to avoid a poorly drawn path. If the GPS data is sent at a too slow of a rate the robot's movement would be staggered and the path that is drawn between two points might not be the exact path taken by the user and could cause the robot to attempt to go through unfavorable terrain (i.e. puddles, mud, drops, etc.).

The other functionality the mobile application will support is that the mobile application will provide the user with a UI access point to control the activation/deactivation of the SAFER Knights' emergency mode. The mobile application will have an icon on the main screen when clicked will send the emergency flag to the robot. The emergency flag will be sent over the Bluetooth connection to be process by the microcontroller unit on the robot. This intern will then enable the SAFER Knights' emergency mode's processes and actions. The same icon on the UI will also be responsible for deactivating the SAFER Knights' emergency mode. While the emergency mode is activated, if the icon is clicked again the mobile device will send another emergency flag to the system. At this point the system will process this emergency flag as the deactivation flag, which will cause the SAFER Knights system to kill the emergency mode processes and return the functionality to normal processes.

The mobile application also needs to control when the device disconnects. The user will need to be given a way to manually disconnect the SAFER Knights system from their mobile device. To do this the mobile application will need to host an icon that when clicked will disconnect the host mobile device from the SAFER Knights system. From there the system will be able to detect that it is no longer connected to the mobile device and it will uninitialize its components before entering the standby mode. This will then allow the SAFER Knights system to be paired with another device to be used again. The app should now no longer offer the emergency mode or disconnect options since there is no appropriate SAFER Knights system to receive it.

Using the mobile app also marks advantages. The first advantage is the ease of use for the users. They will be able to easily issue commands like turning the robot on and off, switching between the remote-control mode and following mode, as well as read information the robot is sending to the user. The second advantage is most everyone has a smart phone so our application will be able to be deployed easily and updated when needed.

We originally used an application development platform called Blynk. It is meant for internet of things (IOT) maker projects, but it had all the functionality we needed for out project. Using this platform saved us from rolling out our own app and user interface, but it ended up working too poorly due to a Bluetooth connectivity issue. The phone and robot were unable to maintain a connection longer than 10 seconds. They would become out of sync and we would get a “packet too big” error. This seemed to be an issue we could not work around because it was baked into the Blynk platform.

In the end we had to create our own phone app. We contemplated which platforms to develop the app and decided to choose Android Studio to make the app. Due to the restrictions including development environment, language limitations, and deployment cost we decided not to implement the application for iOS devices.

Our app will prompt the user to connect to the robot’s Bluetooth module. Once connected the user will be able to see messages from the robot about its status. The user then will be able to enable the robot with a master enable switch. Once the robot is enabled the user can choose to activate Remote Control (RC) mode or Follow mode. In Remote Control mode the user can use a slider to select the speed of the robot and set the direction of the robot using a few buttons labeled center, full left and soon on.

Most of the time the robot will be in follow mode. In this mode the robot will be completely autonomous. The user will still be able to turn the master switch on and off, but the movement will be controlled by the following algorithms. At any time, the user can click the emergency mode button to activate the emergency mode. Once the user is done using the robot, they can then disconnect form the Bluetooth connection.

The Mobile Application follows the following software diagram.

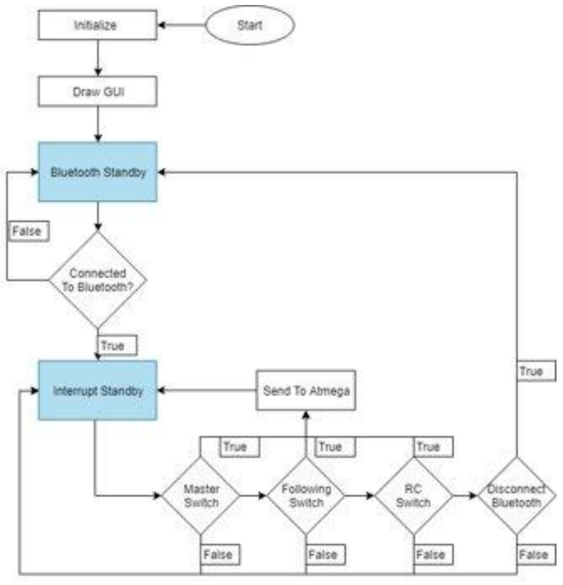
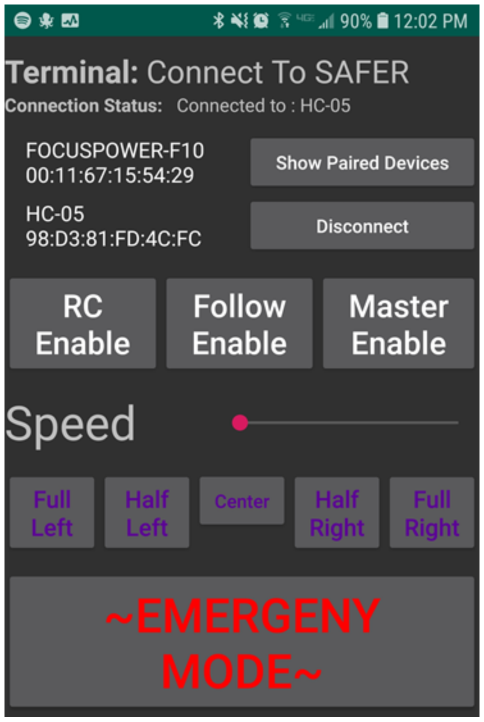


Figure: SAFER Mobile Application Flowchart



Terminal: Connect To SAFER
Connection Status: Connected to : HC-05

FOCUSPOWER-F10
00:11:67:15:54:29

HC-05
98:D3:81:FD:4C:FC

Figure: SAFER Mobile Application

6 Project Prototype Construction and Coding

The first part of this section will outline everything relating to the custom PCB that is required for the project. The purpose of the custom PCB of this project is to control the lights and sound on the robot. The initial design was found via reference designs from online that would best fit our purposes. Different PCB design software are compared and contrasted, and the best chosen. The same type of comparison is done to choose a PCB vendor in order to assemble the custom PCB quickly and cheaply. Afterwards, the final coding plan will be discussed. This coding plan is an outline showing how who will be doing which parts of the coding, what the code will do, and how the final code will work. It serves as a succinct summary of earlier research and planning.

6.1 PCB Design and Schematics

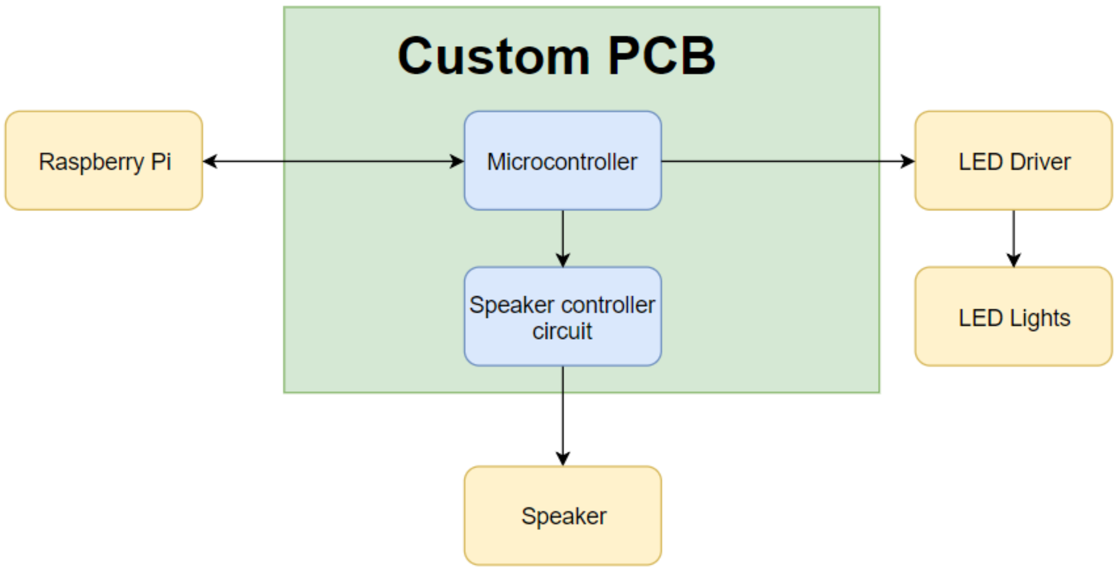


Figure: Initial PCB Design Plan

As a requirement, a printed circuit board, or PCB, will need to be custom designed and built for the project. For this project, the aim of the PCB was to control the lights and audio for the robot. Audio files were held on the Raspberry Pi and executed when necessary. As the project moved on, the PCB was designed and improved upon keeping all light and audio requirements in mind. The initial design, shown above, is a simple model to show the connections coming into and out of the custom PCB. The Raspberry Pi sent signals into the PCB when needed. The microcontroller, which was a ATmega328P to allow for testing on a Arduino Uno, controlled the speakers and the LED lights. For the LED lights, the microcontroller sent signals to the LED Circuit, which will regulate the power going to the LED lights. These were kept constantly on while the robot was being controlled by the user and blinked rapidly when the vehicle was set to Emergency Mode. For the speakers, a circuit was designed to send the appropriate signals and power to the speaker. The speaker was sent signals when Emergency Mode was activated to emit a siren or other warnings to help the user if they are in a dangerous situation.

For the final design seen below, the custom PCB is the brains of the robot. For this project, the aim of the PCB is to communicate with the Pi, control the steering and rear wheel driving, as well as controlling the lights and audio for the robot. Audio files will be held on the Raspberry Pi and executed when necessary. The basic block design is a simple model to show the connections coming into and out of the custom PCB.

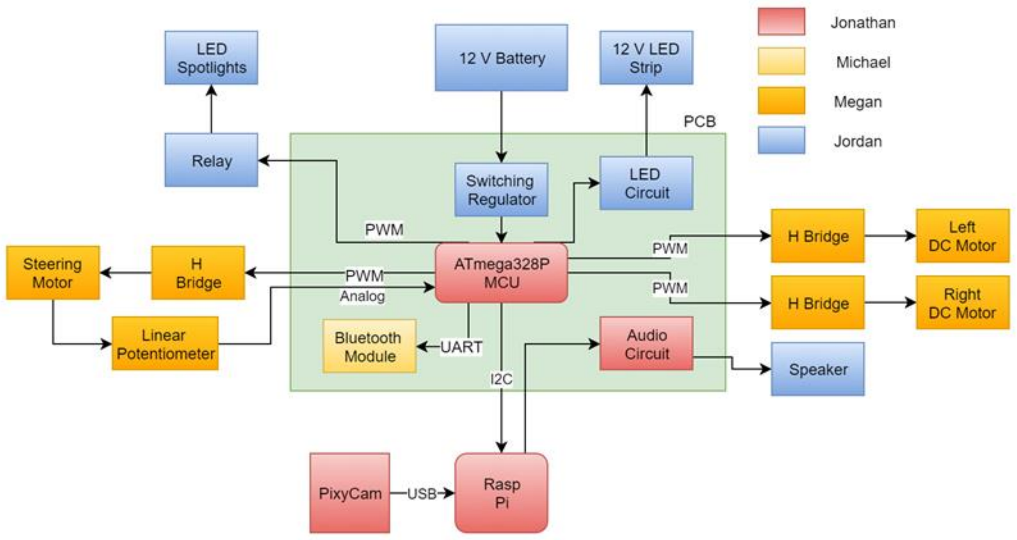


Figure 5. PCB Connection Diagram

6.1.1 Microcontroller Unit

The Atmega328P is a popular and well-known option as it is the microcontroller used in the Arduino Uno board. Widely available and used by many hobbyists and engineers alike, this was a good option since there are plentiful resources online. It has a 32 KB program memory size, 14 general purpose input/output pins, and 6 analog input pins. It supports UART, SPI, and I2C connections. A major advantage was that testing for this microcontroller can be done using the Arduino Uno board, which is easy to program.

6.1.2 LED Light Strip Circuit

To control the LED light strips that line the sides of the vehicle, the LED light strips need 12 volts of power as well as a signal connection to the microcontroller unit so they can blink when necessary. The light circuit shown below allows for controlling the LED light strips by sending a signal to the base of the transistor, which will allow current to flow when the signal is put forth to turn the lights on. This way, when Emergency Mode is triggered, the lights can be toggled on and off to create flashing.

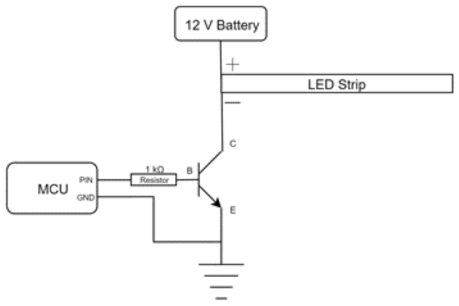


Figure 6. LED Light Strip Circuit

The headlights themselves will be toggled on and off by sending a signal through a relay that is mounted, and these lights can be recharged when necessary.

6.1.3 Audio Amplification Circuit

For the audio amplification circuit, the LM384 5 W Audio Amplifier chip is being used from Texas Instruments. On the data sheet was a reference design to use for typical consumer application of a 5 W amplifier using an 8 Ohm Speaker. This circuit connects from the Raspberry Pi, which holds the audio files to be played, amplifies the sound, and then plays it through the speaker mounted to the robot.

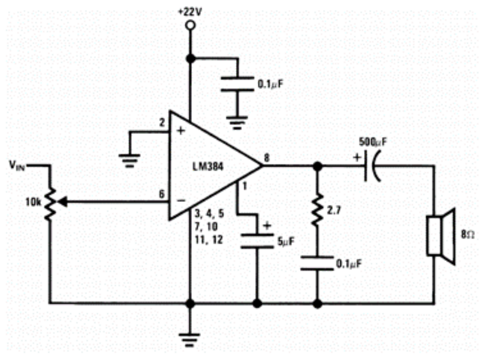


Fig 7. Audio Amplification Circuit

6.1.4 Raspberry Pi

The Raspberry Pi has two main jobs: computer vision and audio implementation. For computer vision, the Raspberry Pi receives information from the PixyCam via USB connection. An I2C bus is implemented between the Raspberry Pi and the ATmega328P. The I2C connection implements the Raspberry Pi as the master while the ATmega328P is a slave. The Raspberry Pi will then relay the processed vision information through the I2C connection to the ATmega328P on the custom PCB while receiving certain flags from the ATmega328P over the connection upon request.

Depending on the received flags the Raspberry Pi will implement multithreading to play audio without impacting the performance of the other aspects of the program.

6.1.5 Schematic and Board Layout

Below is the schematic for the custom PCB, made with EagleCAD. On it is the circuit components necessary for the ATmega328P to work, the LED light strip circuit talked about previously, and the audio amplification circuit. There are also various pin outs to connect to steering, driving, power, etc. There are also pin outs for a possible GPS module, compass module, Microsoft Kinect, and a USB hub, which were not used for the scope of this project. There are three on-board voltage switching regulators, two 2 Amp regulators and one 1 Amp regulator. The 1 Amp regulator takes the 12 Volt input and regulates to 5 Volts with a 1 Ampere max current, and this is used for the ATmega328P microcontroller unit. The 2 Amp regulators also regulate the 12 Volts down to 5 Volts but with a 2 Amp max current, and this supplies power to the Microsoft Kinect and USB hub.

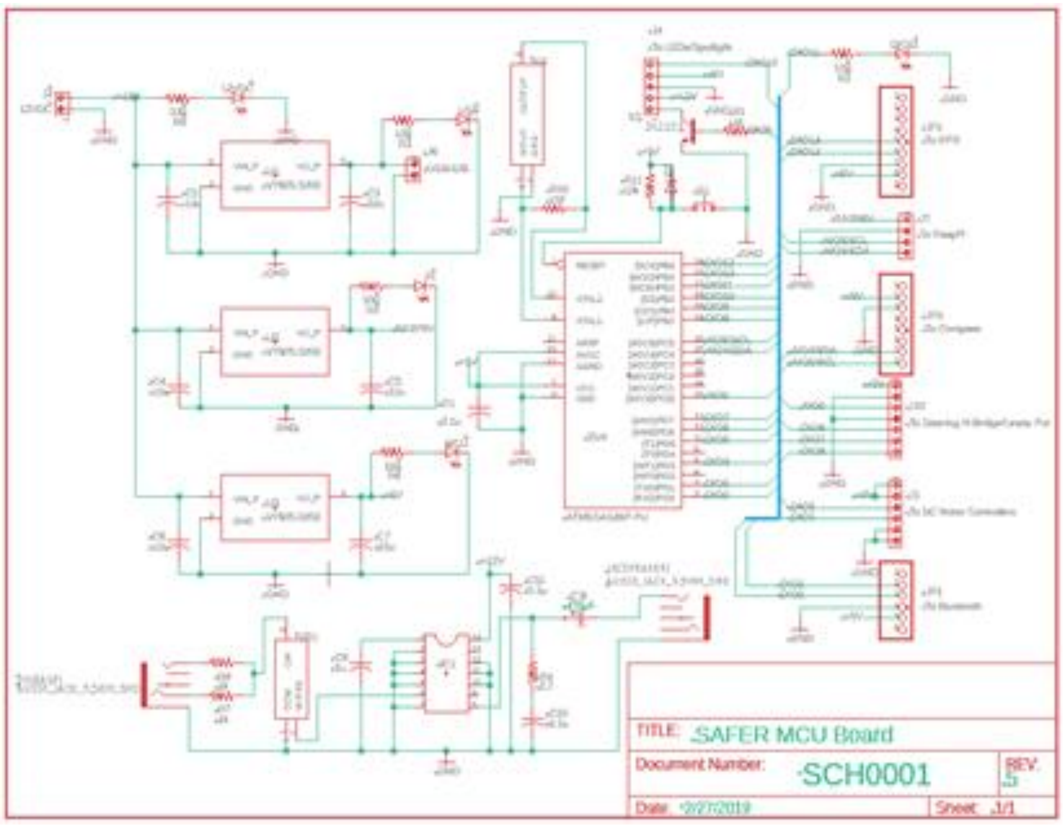


Figure: Overall Board Schematic

The Printed Circuit Board (PCB) was then designed based on the schematic. Due to limitations of the version of EagleCAD that was being used and the cost limitations of the PCB manufacturer we decided to go with a board that is 80 mm x 100 mm. The PCB houses the many main components of the SAFER system including the ATmega328P, audio amplifier, voltage regulators, and component connectors. Most of the passive components including resistors, capacitors, and diodes utilized a 1206 footprint to allow for easier hand soldering. The PCB also features DIP sockets for the ATmega328P and the LM384 for easy replacement of any damaged units. The PCB also contains the amplifier circuit that utilizes the LM384 amplifier chip that allows for the Raspberry Pi that connects to the PCB to play audio through the connected speaker.

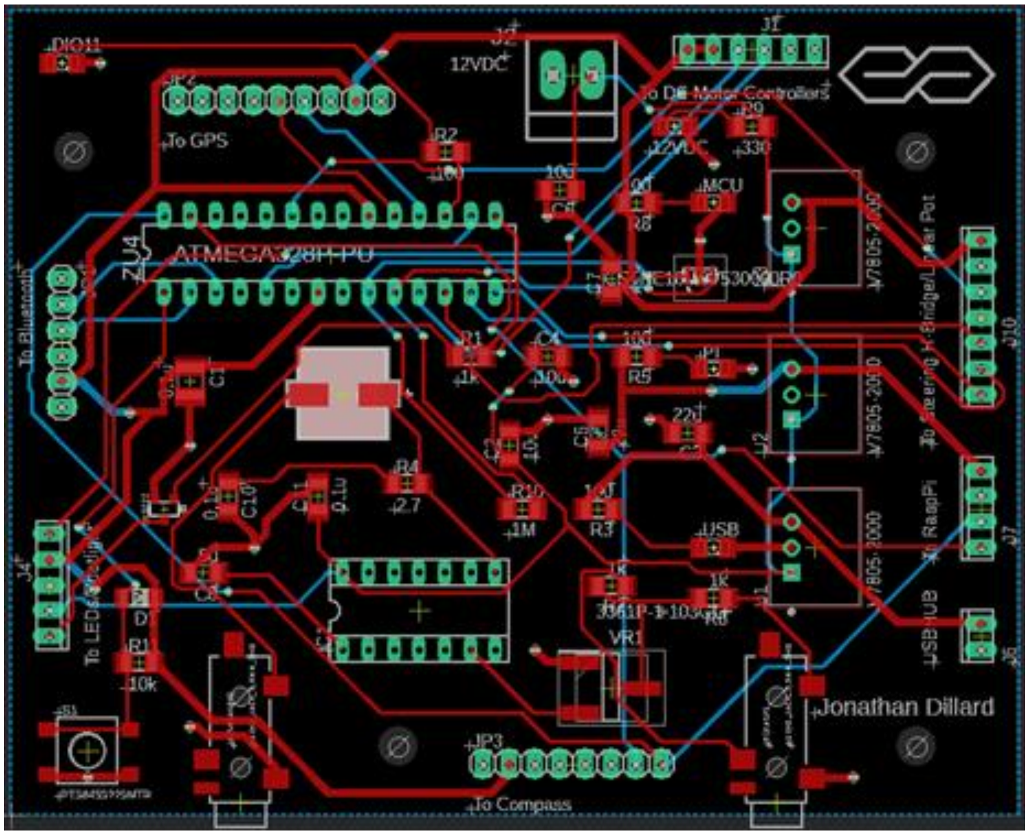


Figure 9. PCB Board Layout

6.2 PCB Design Software

There are plenty of software choices out there to help aid design, called Computer Aided Design, or CADs. They allow for various components to be designed as a schematic and organized onto a virtual PCB with the components connected in an ideal format to fit onto the frame of the PCB. Components can be moved around to produce a better layout and so that the PCB doesn’t end up with all the components cramped together. These files can then be uploaded to a PCB Vendor’s site to get the PCB custom ordered. Below is a comparison between some well-known PCB design software options that can be found online.

Table: PCB Design Software

Product	**User Base**	**Learning Curve**	**Resources**	**Price**
KiCAD	Hobbyists	Okay	Okay	Free

EagleCAD	Anyone	Okay	Plentiful	Free
DipTrace	Hobbyist	Easy	Okay	Free
Altium	Professionals	Difficult	Okay	Pricey
EasyEDA	Hobbyists	Easy	Good	Free

There are plenty of available PCB design programs available, and it can be overwhelming trying to figure out which is best to use. To decide, a variety of factors determined our choice in software. EagleCAD is a fairly popular and is suggested by people who use it. It's used both by industry professionals and hobbyist, so it's a good program to know for the future. Its design files are also accepted by many PCB manufacturers, including the one chosen by our group for this project. People in our group have also used it before and are familiar with its functions. While EagleCAD can be purchased, they offer a free version, so the cost of the program is not an issue. There are also plenty of tutorials, designs, and resources online to help use the program.

6.3 PCB Vendor and Assembly

Since a custom PCB is being designed, a PCB manufacturer must be chosen to order them. They must be able to accept EagleCAD design files, be cheap, and relatively quick, with low shipping costs if necessary. Therefore, out of many available, JLC PCB was chosen as our PCB vendor of choice. This is mainly because they accept EagleCAD and have been used by group members in the past with positive reviews. They offer 10 PCB for only $2, so they are definitely cheap enough. They've proved to be quick and on time in the past, so as long as the PCB is designed well in advance, the time will not be a concern.

6.4 Final Coding Plan

Since programming plays such a big role in the project to achieve autonomous following as well as object detection and avoidance, it's important to have a plan when it comes to the code. Below is the final coding plan, emphasizing the main features and functions of the code that will allow the robot to work successfully.

The following was what the original coding plan contained. The current code implementation however differs where more control of the system was passed to the Microcontroller unit on the PCB which is now an ATmega328P.

6.4.1 SAFER Knights Architecture

The SAFER Knights system utilized a Linux based operating system to support the operations of multiple functions of the unit. The utilization of the Linux operating system offers multiple additional tools that other solutions do not offer. Since the Linux operating system is a fully developed operating system it offers multiple tools and programs that will better aid in the development and implementation of the SAFER Knights system. Many microcontroller units do not support the use of an operating directly on the hardware but rather expect you to write the systems firmware directly to microcontroller units flash storage through an external device, IDE, and compiler. This then would require additional functions and programs to be created in order to run additional functionality. This also can cause problems with device recognition and utilization of proprietary devices such as the Kinect. File storage would also become a large problem as well. File storage will require some form of file system, additional non-volatile storage, and a way to transfer the files between the SAFER Knight systems and an external device. Some of the problems can be lessened by utilizing external libraries but still leaves restrictions such as complexity of file creation, manipulation, and alteration. These restrictions are also affected by hardware restrictions of microcontroller units that cannot support an operating system and typically utilizes code flashing such as the Arduino units. By utilizing the Linux operating system we negate all software problems that is presented with direct firmware programming. The SAFER Knights system will utilize different operating system utilities to properly function and increase proper system utilization. The SAFER Knights system will rely on the operating system to handle low level operations such as memory management, pin layout, device initialization, etc. The SAFER Knights will also rely on the Linux operating system to handle higher level processes to better aid in the execution of the overall system. The SAFER Knights system is to be initialized at startup of the system. To achieve this a bash script will be written that will launch the SAFER Knights start up script. This bash script will then be stored in the /etc/init.d directory of the operating system. This specific directory is used to store bash scripts that are run when the operating system boots. By putting the bash script that launches the SAFER Knights start up script into the /etc/init.d directory, it guarantees that the SAFER Knights system starts at system launch.

The Linux operating system also offers a solution to file handling with its file system. The Linux operating system will be used to handle file creation and manipulation. This allows for the simplification in both the video recording as well as the emergency event logging. Using the Linux file system, the SAFER Knights system can easily store both text and video files at predefined locations in the system storage and utilize the Linux operating system prepackaged software to handle all low-level file handling. The Linux file system simplifies the use of all file related resources, without the Linux file system, a rudimentary file system would have to be created as long as the hard drive supports it by having a large enough non-volatile memory space. Creating a new file system to support the video recording and log creating would add a significant amount of complexity to the system. The system will then also need to be able to take the information from the Kinect and be able to parse the data and create a video file. This additional complexity can be negated by utilizing the provided functionality given in the Linux operating system. Since the Linux operating system is a full operating system it can install and utilize device drivers. Since the SAFER Knights system will implement object avoidance and external storage, the system needs to connect to the appropriate hardware devices to do these tasks. Since the certain pieces of propriety hardware that was chosen such as the Kinect sensor or the external hard disk drive requires drivers to appropriately operate the Linux operating system can support the driver software, allowing the system to utilize these devices where as other software architectures might not support the utilization of hardware devices that require driver software to operate.

6.4.2 Startup Software

The SAFER Knights software was initialized on system start up. This allows for the SAFER Knights software to start without any external interaction will the software initially. The SAFER Knights start up script will be responsible for starting the SAFER Knights software. The SAFER Knights will be self-contained in a infinite loop. In this infinite loop the SAFER Knights will stand by until the appropriate signal is received by the ATmega328P device.

Once the ATmega328 returns the appropriate signal then the software will trigger the function connected to the appropriate signal.

6.4.3 Initialization Software

Once the startup software calls the initialization script the initialization script will be used to startup the additional hardware devices. The initialization script will start by initializing the appropriate pins and inputs for the different hardware devices including the compass, GPS, Kinect, motor controllers, etc. This now allows for these devices to be used by the microcontroller unit. The program will now assign appropriate variables and any other preparation steps required to utilize these devices. With the devices initialized the initialization program will then start the communication operations required to communicate with certain hardware devices. The I2C connection will be initialized and set to a variable which in turn will be passed to the movement and object avoidance function to allow for the reading and utilization of data. The I2C interface will be utilized to read the data coming from the compass. This data will come through the connection which will be utilized by later functions and scripts. The function will then initialize the SPI interface. In a similar way the initialization function handled the I2C connection, the SPI connection will be initialized and set to a variable that will handle the interface connection.

Next the initialization script will start a session that will interface with the GPS module. This GPS module will rely on the gps and gpsd external libraries to appropriately set the UART connections and receive the appropriate data. With the external libraries imported, the function will set a session variable to a GPS object with the appropriate host and port information passed. This session variable along with the I2C variable and the SPI variable mentioned above and all other declared variables for the other hardware devices will then be passed to the appropriate function calls. As the last step the initialization function will then call three additional functions. The initialization function will call the emergency mode flagging function while passing the Bluetooth connection variable as an argument. This will allow the SAFER Knights system to monitor the Bluetooth connection for the emergency mode flag.

The next function call will be the emergency mode initialization function with the variable containing the SPI connection being passed as an argument. By passing the SPI variable, this will allow the emergency mode to control the signals being sent the LED microcontroller unit. The final function to be called will be the movement/object avoidance function with the Bluetooth variable, I2C variable, GPS session variable and a Kinect sensor variable as arguments. By passing these arguments the movement/object avoidance functions will be able to access the sensors that it will need for path creation and the sensors needed for object detection. Since these variables are initialized, they are now active and consuming power. By initializing the variables after a positive Bluetooth connection is established allows for better energy performance due to the lack of down time that the initiated hardware devices have during standby mode.

The current system no longer implements the GPS or compass module. The PCB was used to implement the Bluetooth module to communicate with the phone app. The PCB and Raspberry Pi then both implement the I2C Bus. The system then waits for proper flags to be implemented

6.4.4 GPS and Movement Software

One of the SAFER Knights system objectives is to follow the user to their destination. To do this the user needs to connect their mobile device to the SAFER Knights robot and keep it on their person. This will allow the mobile device to stream its GPS coordinates to the SAFER Knights main microcontroller unit. The microcontroller unit itself will have its own GPS module. This GPS module will work in a similar way to the GPS module that is installed on the user’s mobile device. The GPS module will constantly report the GPS coordinates of the SAFER Knights system. From there the system can use the two sets of coordinates in tandem with the compass to determine a path between the two points.

6.4.4.1 Initialize Function

The SAFER Knights movement subsystem will contain a script that will be used to receive GPS coordinates of both the user mobile device and the SAFER Knights system, determine the distance between the two points, determine the appropriate direction to travel to reduce the distance, and then use this path to move the system. The script will start by initializing the appropriate values. A current latitude and longitude variable for both the mobile device and the SAFER Knights system, a previous latitude and longitude variable for the mobile device and initialize the serialization for receiving the data from the sensors and the connected Bluetooth device.

6.4.4.2 Main Function

The software will now move onto the main function that will handle the calculations and movement. This function will consist of a loop that will be used to constantly update necessary data that is used for calculation (i.e. mobile device GPS data, SAFER Knights GPS data, etc.) and will not exit until the user declares that they have reached their destination and completes the use of the system. Inside the loop the system will call the GPS coordinates from the mobile device via the serialized Bluetooth connection that has been established. The data received from the Bluetooth connection to the mobile device will then be passed to another function. This function will accept the raw data received from the mobile device and parse the data. Once the data is parsed the function will return the now parsed and usable data back to the main function. From here the main function will now store the current values of the current latitude and longitude variables for the mobile device into the previous latitude and longitude variables and stores the returned parsed data into the current latitude and longitude variables for the mobile device. The code will then call the GPS module to receive the GPS coordinates for the SAFER Knights system.

The data received from the GPS module will then be sent to a function that will be able to parse the raw data received from the module. Once the function returns the parsed data, it is stored in the current latitude and longitude variables for the SAFER Knights robot. The main function will then call a distance function, this function will accept the current latitude and longitude variables for the mobile device and SAFER Knights robot and return the distance between the two points. With the distance the main function will then determine whether or not the distance is within a certain distance threshold. The distance threshold is used to determine if the SAFER Knights robot is close enough to the mobile device to stop moving. This will then guarantee that the value of the current latitude and longitude variable of the mobile device and the value of the current latitude and longitude variable of the SAFER Knights robot will never equal and thus never overlap. This distance threshold is a safety precaution taken to avoid the robot from getting too close to the user and causing damage or harm to said user. If the main function determines the distance to be within the threshold, the function will set the motors to a speed of 0 stopping the robot from getting any closer to the user and mobile device. If the distance is not within the threshold then the main function checks if there are any obstructions in the way of the SAFER Knights system if so, the function will pass control of the SAFER Knights robot to the object avoidance software.

Once the object has been avoided the main function will skip to the next iteration of the loop to retrieve the new coordinates and redraw the path. If there are no obstructions in the path, then the function determines if the current values of the latitude and longitude variables for the mobile device equals the previous values of the latitude and longitude variables for the mobile device. If the two variables are equal it means the user and mobile device have not moved since the last iteration of the loop and there is no need to adjust the motors as the motors should still be on the path as previously calculated, so the function continues to the next iteration of the loop. Its only when the user has moved that the path needs to be changed. If the current values of the latitude and longitude variable does not equal the previous values of the latitude and longitude variable, then the path needs to be adjusted to compensate the new target location. The main function will then pull the directional information from the compass. The compass will provide the raw data on the direction that the SAFER Knights robot is facing. The function will then start calculating the appropriate values of the angles of movement. Using this data, the SAFER Knights system will then calculate the angle between the currently facing direction and the direction of the target location. From there the function can determine the angle the steering wheel needs to turn in order to reach the desired location. Using the angle of direction, the function will calculate the angle of steering. With the angle of steering calculated the function will set the steering servo motor with the appropriate angle of steering in order to appropriately change the angle the vehicle is facing. This change in steering direction will align the SAFER Knights robot with the user destination. With the steering servo set, the main function will adjust the DC drive motor, whether the SAFER Knights system needs to increase or decrease speed. The main function will then continue onto the next iteration of the loop where the process will be redone until the loop is broken by the user via the mobile application on their mobile device and received through the Bluetooth connection.

The system will now utilize the PixyCam to detect the X location and largest dimension of the color code to determine speed and direction. By utilizing these two components the system can then do the appropriate math and send the data to the motor controls.

6.4.5 Emergency Mode Software

The SAFER Knights system's purpose is to safely escort the user to their desired destination. To do this the SAFER Knights system implements an emergency mode setting that, in the situation something does occur, will attempt to dissuade any hostile entities as well as flag recordings of the events to be analyzed at a later time. The emergency mode is initiated via the mobile application interface that the user uses to connect their mobile device to the SAFER Knights system. This gives the user to activate the emergency mode at will when they feel it is necessary to do so. A further reaching goal of the emergency mode system is to implement a voice command interface to allow the activation of emergency mode without the need to interact with the mobile application. The emergency mode, which consists of both flagging via Bluetooth and initialization, will utilize multiple subsystems in order to interface with different hardware components in attempt to dissuade any mishaps from occurring.

The SAFER Knights now implements the emergency mode as follows. To trigger this mode the user will press the emergency button on the companion app. The robot will then go into emergency mode. The robot will then flash its headlights and LED light strip. The robot will also utilize its speaker to start playing a siren sound. These effects will continue until the user chooses to disengage emergency mode using the same button. This feature is meant to scare away any hazards and to draw attention to the user for any potential help. To achieve this mode, we are utilizing all three of our software environments. The mobile app will send the signal to the PCB to engage in emergency mode. The PCB will then start strobing the lights and send a command to the Raspberry Pi. The Raspberry Pi will take care of playing the audio until the user turns off emergency mode.

6.4.5.1 Emergency Mode Flag

The emergency mode of the SAFER Knights system will be initiated by the overarching architecture of the system and will consist of two main components. The flagging component is responsible for obtaining data from the mobile device via a Bluetooth connection and flagging the system when the emergency mode signal is received. The emergency mode flagging script will begin with an initialization function. Since the emergency mode is reset each time the SAFER Knights system goes through its initialization process, the emergency mode flagging script will begin with the initialization of a boolean variable, which will represent the emergency flag itself, to false to ensure that the emergency mode is deactivated at start up. The initialization function will then initialize a serial connection with Bluetooth in order to receive the appropriate data. Once the initialization function completes the emergency mode flagging script will start its main function. The main function will consist of an infinite loop that will constantly be scanning for the appropriate signal. This loop will constantly be running until the SAFER Knights system goes through deinitialization in which case the script will be stopped by the system itself. Within this loop the emergency mode flagging script will wait for the signal from the user via the mobile application that the emergency mode UI element has been selected. Once the value returned by the Bluetooth connection is true the main function will then contain an if…else conditional block.

The if…else conditional block will determine the current status of the emergency mode flag boolean variable. If the value of the boolean variable is determined to be false, the emergency mode flagging script will then set the emergency mode flag to true. Otherwise the emergency mode flag is already set to true so thus the emergency mode flagging script will set the status of the emergency mode flag to false. From this point the emergency mode flagging script will continue to the next iteration of the infinite loop where the same process will be followed again until, as stated above, the SAFER Knights system goes through deinitialization and exits the emergency mode flagging script. The emergency mode flag that has been set in this script will now affect the actions of the second script in the emergency mode subsystem, emergency mode initialization/deinitialization.

6.4.5.2 Emergency Mode Initialization/Deinitialization Software

The second part of the emergency mode subsystem is the emergency mode initialization/deinitialization script. The emergency mode initialization/deinitialization script is responsible for the majority of the emergency mode subsystem. The emergency mode consists of an initialization function and a main function, which contains an infinite loop. The emergency mode initialization/deinitialization script is called at the SAFER Knights system initialization. The initialization function's task is to initialize the I2C/SPI connection that will be used to communicate with the LED's microcontroller unit. Once the connection to the LED's microcontroller unit is initialized the emergency modeinitialization/deinitialization script then starts the main function which contains the infinite loop. This infinite loop is used to constantly scan the status of the emergency mode flag and is only ended when the SAFER Knights system deinitializes. The main function of the emergency mode initialization/deinitialization script starts the infinite loop. Within the loop is an if…else conditional block. This block's condition is controlled directly by the value of the emergency mode flag set by the emergency mode flagging script. The emergency mode initialization/deinitialization conditional block will determine if the emergency mode flag is set to true. If the flag is set to true, the emergency mode initialization/deinitialization script will then send the initialization signal over the I2C/SPI connection to the LED’s microcontroller unit so that it can execute its preprogrammed instructions to handle the LEDs.

The next step is to flag the video recording for later review. The emergency mode initialization/deinitialization script will then check and see if a log file exists in the file system at a predetermined location. If such a log file does not exist, the emergency mode initialization/deinitialization script will create a new log file at that file location. If the file does exist, then the emergency mode initialization/deinitialization script will append on to the existing file. Once the log file is located the emergency mode initialization/deinitialization script will then load the file for editing. The emergency mode initialization/deinitialization script will then write to the file, the playback time of the activation of the emergency mode in reference to the recorded video by taking the start time of the video which is declared in another part of the SAFER Knights system minus the time of activation of the emergency mode and the name of the video file currently being recorded which is declared in another part of the SAFER Knights system as well. The emergency mode initialization/deinitialization script will then enter another loop. This loop is a continuous loop with a halting condition of when the emergency flag is false. In this loop the emergency mode initialization/deinitialization script will continuously call the speaker script which will play the prerecorded audio file. This will allow for continuous playing of the audio feature of the emergency mode. Once the emergency mode flag is set to false the emergency mode initialization/deinitialization script will break out of the audio loop and then send another signal to the LED’s microcontroller unit. This second signal will tell the LED’s microcontroller unit to stop running its emergency mode instructions and return to its normal state. The emergency mode initialization/deinitialization script will now reenter the infinite loop where move onto the next iteration of the loop which will keep scanning for the emergency mode flag to be set to true again.

6.4.6 LED Microcontroller Unit

The SAFER Knights robot will feature a custom PCB board that will control the system’s LEDs. This custom PCB board will host a separate microcontroller unit from the main one controlling the overall system. This microcontroller unit will be utilized to control the states of the individual LEDs. When the emergency state is activated the main microcontroller unit will send a signal to the custom LED PCB board and the LED microcontroller unit will control the LEDs based on the preprogrammed instructions for the emergency LED mode. The LED microcontroller unit can be programmed using either the TI Code Composer Studio or Energia. The two pieces of software offer a way to interface and flash the microcontroller unit with code. Due to its simplicity and its preconfigured pin layout, Energia will be used to program the LED microcontroller unit so long as the microcontroller unit is supported by the Energia software.

6.4.6.1 Initialize Function

On startup, the LED’s microcontroller unit will run the initialize function. The initialize function’s main task will be to initialize inputs, outputs, and any variables that will be used in the overall code of the LED subsystem. The initialize function will first initialize the connection with the main microcontroller unit via an I2C or SPI configuration to receive the appropriate data for the emergency mode. From there the function will initialize the individual pins that each LED controller is set to. The function will initialize the necessary pins to outputs pins and set their values to high, which will turn on the associated LEDs. The initialize function will then declare and initialize a boolean variable to false, which will be used later on in the loop of the program. The initialize function will then setup the interrupts that will be used by the microcontroller unit to properly control the system. The function will initialize the digital pin interrupt. This interrupt will allow the system to interrupt based on the connection input. Next the initialize function will setup the timer interrupt, which will be used to toggle the LED states, and disable this interrupt as it will not be used immediately.

6.4.6.2 Program Loop

The next function to be called by the microcontroller unit is the loop function. The loop function is a function that contains an infinite loop that will constantly run and update the LEDs as appropriate. Within the loop function, there will be no executable code. The point of the loop function is to keep the program continuously running as the majority of the program's functionality is ran through interrupts. When main microcontroller unit sends the signal to the LED microcontroller unit the pin interrupt that was initialized in the initialize function will occur and will interrupt the loop function. Once the LED microcontroller unit is interrupted it will run the pin interrupt service routine. The pin interrupt service routine will contain a conditional if…else block that will determine the status of the Boolean variable initialized in the initialize function. If the variable is set to false, this means the system was not previously in emergency mode. The conditional block will then set the Boolean variable to true, enable the timer interrupt, and exit the interrupt service routine. The system will then resume the loop function with the timer interrupts enabled.

Once the specified amount of time has passed, as declared in the initialize function, the microcontroller unit will interrupt and call the timer interrupt service routine. The timer interrupt handler will then go through each pin that was initialized in the initialize function and change the current state of the pin to the opposite output, i.e. if the pin is set to high then the interrupt service routine will set the pin to low and vice versa. This interrupt service handler is what is responsible for alternating or flashing the LEDs in the SAFER Knights' emergency mode. Once all pins have been set to the opposite state the interrupt service routine returns to the loop function. The interrupt will continue to activate at the appropriate time intervals to change the state of the LEDs while in emergency mode. Once the main microcontroller unit sends the signal to the LED microcontroller unit to disable emergency mode, the LED microcontroller unit's pin interrupt will activate which will call the pin interrupt service routine again. When the pin interrupt service routine's if…else conditional block checks the state of the boolean variable and verifies that its set to true, the interrupt service handler will disable the timer interrupt and set the variable to false. Now that the emergency mode has been disabled the lights need no longer flash, but rather keep its normal operation state. From there the interrupt service handler will go through the pins initialized in the initialize function and set their states to high to turn them all on. The interrupt service routine will then return to the loop function where the program will stay until either the system is shutdown, or the pin interrupt is called again.

6.4.7 Speaker Software

The SAFER Knights robotic system’s emergency mode will feature a speaker subsystem that will be utilized to play the appropriate audio when the emergency mode is activated. In contrast to the LED subsystem, the speaker subsystem will not have its own dedicated microcontroller unit but rather will use the main microcontroller unit to control the playing of audio. The main microcontroller unit will have a python script that will be used to play the appropriate audio file saved on the system and send the audio to the auxiliary output.

For the emergency mode the audio that will be played will be prerecorded and saved into a static location of the file directory of the main microcontroller unit. This audio file will be used by the script to play the desired audio. When the SAFER Knights’ microcontroller unit activates the emergency mode, the overarching system will call the speaker python script. The python script will then load the audio file from the predefined location and play it by invoking the systems integrated audio player and passing the audio file location. The audio will then be sent to microcontroller unit’s default audio output, which will be the microcontroller unit’s auxiliary port. The auxiliary port will then send the audio the speaker’s PCB board to be played. Once the speaker’s python script has been ran, the system will return to the program that called the speaker python script, that program will wait the appropriate time to allow the audio to play in its entirety before calling the speaker script again to replay the audio. This allows the emergency mode program to control the playing of the speaker script and allows it to more easily stop the loop used to constantly play the audio rather than have the speaker script control the audio loop itself.

6.4.8 Deinitialization Software

Once the user has completed their use of the SAFER Knights system, they use their mobile device to send the complete signal to the SAFER Knights system. At this point the system needs to call the appropriate functions to set the SAFER Knights system back into standby mode. To do this all hardware devices that were initialized prior in the initialization function (i.e. Kinect, GPS, compass, etc) needs to be deinitialized. This will affect the systems overall efficiency as stated previously. When the movement/object avoidance program receives the Bluetooth signal that the user has reached their destination, it passes the system execution to the deinitialization script. This script will deinitialize the hardware devices except for the Bluetooth device. When the script launches it is passed all the variables that relate to the devices including the I2C variable, SPI variable, GPS variable, Kinect variable, etc.

From there the function will set the pass argument variables to null. By setting these variables to null the systems trash collector will determine that the objects that were connected to the variables are determined to be unreferenced and will delete them. With no objects tied to the specific hardware devices, these devices will be uninitialized. With the hardware devices uninitialized the devices will no longer be consuming as much power, saving on the efficiency of the power used for each of these devices. From this point now that the deinitialization function has cleared all the hardware devices the function will return the execution back to the standby mode script. This will put the SAFER Knights system back into standby mode awaiting the next Bluetooth connection and the next user.

The final implementation of SAFER moves most of the control to the PCB microcontroller unit. The Raspberry Pi controls the audio threads and computer vision and communicates over the I2C connection, where the ATmega328P controls all hardware including motors, sensors, and lights.

7 Project Prototype Testing Plan

A testing plan must be created in order to quickly and efficiently conduct testing in the future with separate parts as well as the entire system. Once all separate subsystems have been tested successfully on their own, all the subsystems will be integrated together to form the final product. By doing intermediate testing piece by piece, it will be easier to isolate potential problems and tackle them sooner rather than later. Below are the testing plans for both hardware and software.

7.1 Hardware Testing

This section explains the various plans and ways that testing will be done for making sure the hardware works as needed. Major components will need to be tested such as the vehicle base, the Power Wheels needs to be tested to make sure it works, that the wheels turns responsively once the servo motor is integrated into the design, and to make sure the braking works. Below are the test plans for the vehicle base, voltage regulator, speaker, and PCB.

7.1.1 Vehicle Testing

SAFER's objective is to ensure that the user is able to arrive at their destination safely and with piece of mind. The technology we are developing focuses on the safety aspect more on the mechanical means of operation. This means that we don't need to build all aspects of this robot from scratch, especially the robotic platform. As mentioned previously we plan on using a common kids toy called a Power Wheel. We will systematically test the power wheel step by step to make sure everything is working and working as intended before we add more components.

With this power wheel we will get some robust premade robotic components. The first of which is the drive train. Power wheels have large tires meant for all terrain operation. The "gearing" of the power wheels is also low. This means it has low torque and large all terrain tires. This is the result we desire for our robotic platform. We will need to take the power wheels in conditions similar to its expected operation environment like around the UCF campus and verity it is operation within its given parameters. We also will need to test the power wheels in conditions it may be operating in less often like open grass areas for more wooded areas to verity its drive train is able to perform as needed.

When the drive train of the vehicle is tested and verified we will need to start thinking about adding speed control to the motor. By default, the power wheels usually have only one or two speeds. Although this is more of a stretch goal we hope to add a more granular speed controller to the power wheels. We will need to test the speed variation and its consistency of speed.

Once the power wheels has passed the drive train tests we will have to add the ability to steer the vehicle. To autonomy steer we will attach servos to the front tires to the same affect turning the steering wheel. This will have to be tested to verify that is has the needed strength turn the tires while there is a load in the vehicle. This condition will also have to be meet while the vehicle is immobile, the hardest time to move the wheels in a steering motion.

Once the vehicle has verified drive ability we will need to test its battery. Once knowing the specifications for voltage, amperage, and total wattage we will test to see if the vehicle’s battery can still meet those specifications. We also then have to opportunity to see if the battery is able to provide enough power for other operations also. Our team does have two larger backups twelve-volt batteries in case the original is defective.

7.1.2 Voltage Regulator Testing

There are many things that could go wrong when many things are connected together. The first measure taken to combat this is a voltage regulator in order to ensure the bias voltage of the PCB is constant. Assuming the regulator works properly, this would account for any spikes and prevents a system from delivering less power than expected. The regulator will be tested with a multimeter.

Before even connecting the multimeter to the regulator you must first read the data sheet to be sure which pin is the input, output and ground. Each test will have a different expected outcome and therefore this layout must be addressed. The expected values will come from the data sheet as well. Finally, ensure the multimeter is set to the voltage setting.

The first test will be to connect the multimeter to the input and ground of the regulator. The expected output should be 1-2 volts higher than the intended voltage. If the device reads nothing, then the regulator must not be properly receiving current from the power supply.

Once the multimeter is reading the expected value of the input and ground, the next test would be between output and ground. The output should be the same as the one listed in the data sheet of that particular regulator. It is possible the device reads something else. In that case, it is known the regulator is faulty and must be replaced.

7.1.3 Speaker Testing

There are only a limited number of components in the speaker design, therefore it will be easy to test the system. Things such as the op amp, power supply, NPN BJTs or the PNP BJTs could be faulty and give some output other than the expected one.

A multimeter can easily test the op amp. The first measurement will be the voltage at both the positive input and negative input. If the op amp is functioning correctly, both of the outputs will be the same. The other key aspect of a functioning op amp is there should be no current entering either the positive or negative terminal. Both of these tests are assuming there is a negative feedback and effectively infinite gain/input resistance.

Finally, the op amp will be connected with a unity gain and the output will be measured with a known input voltage. The output must be equal to the input of the op amp since it was given a gain of 1.

When the whole design of the audio amplifier is made, it is possible to burn either a NPN or PNP transistor. These transistors are designed into a push pull output stage in the speaker design. This means one pair will control the positive output and the other controls the negative. The output can be measured with an oscilloscope with respect to the input.

One possible output could have a distorted positive region. The power supply will then be observed and immediately be shut off. If the current reading on the power supply spikes, it can be assumed one of your transistors burned. With both of these results it is safe to say one of the NPN transistors is no longer working and must be replaced. After the power is off, carefully touch the transistors to see which of them is hot. This transistor is then replaced, and the design and connections of the system must be corrected.

Another output could show a distorted negative region. This would be caused by the PNP transistors. The testing would then be the exact same as the NPN above.

The final possible fault could be the connection between the power supply and audio amplifier system. In this case the components will not be properly biased. This would cause the transistors to be operating in the incorrect mode or could cause clipping in the output.

It is possible the system could be given less power than the supply is reading. If this is happening, you may see the output being clipped. The clipping occurs when the voltage range given to the system is less than the expected output will generate. A simple multimeter reading of the power supply inputs will take care of that.

The other possibility would be the transistors could be operating in an unwanted mode. This would cause the current being supplied to the speaker to be essentially nothing and therefore resulting in no sound. The voltage across the base and emitter of each transistor can then be measured with a multimeter. These values will then be compared to the saturation values on the datasheet for the corresponding transistor. If it is less, then the BJT has not turned on. If all the transistors appear to be on, then the connections and design of the system must be addressed.

7.1.4 PCB Testing

To make sure the custom PCB will work as intended with the rest of the project, testing will need to be done to check the various aspects of it. The microcontroller will be tested initially using a development board. This has been narrowed down to two main choices. The first choice is the MSP430 Launchpad with the M40G2553 microcontroller. Since multiple team members have MSP 430 Launchpads from previous classes, it seemed to be the ideal way to start testing. Not only is the Launchpad already in possession, but it has also been used by multiple members of the group and will be easy to program.

The second choice is the Arduino Uno, a simple microcontroller development kit that is very popular amongst hobbyists and engineers. Although this choice was passed over in favor of the Raspberry Pi to be the master controller for the project, it can still be a good choice for testing the microcontroller for the custom PCB. The Arduino Uno has an ATmega328P microcontroller, and this particular microcontroller has numerous examples and resources online specifically tailored to it. This gives it a great advantage over other microcontrollers. For now, the TI MSP430 Launchpad has been chosen because multiple team members not only own the development kit, but also be they have experience using it.
The purpose of the custom PCB is to control the LED lights on the robot as well as turning on the speaker when the vehicle goes into Emergency Mode. To test the speaker circuit, the custom PCB will be connected to an external power source as well as the speakers. The Raspberry Pi, which will also be connected to the custom PCB, will provide the signal specifying that Emergency Mode has been enabled. Once the Raspberry Pi has sent out a signal, the audio file should play from the speakers.

To test the LED lights, the custom PCB will be attached to the Raspberry Pi, the LED lights, and an external power source. A simple blinking on and off program will signal the LED lights to blink on and off as an initial test of the system. Once that has been confirmed to works as predicted, a larger test will be done to make sure that the actual code will work. The code will turn the LED lights on initially at a user prompt, and then will start blink rapidly when the user initiates the Emergency Mode function. Once everything has been integrated into the final system, the prompts to turn the LED lights and Emergency Mode on will come from the user’s Bluetooth connected smart phone.

7.1.5 LED Testing

The SAFER Knights project requires lights to illuminate the surrounding area, as well as, flash when prompted. The only was to be sure the lights work properly is through testing. Some possible issues that could rise are, but not limited to:

- Power supply connection
- Faulty light
- Faulty transistor
- Human error (circuit connected incorrectly)

7.1.5.1 Power supply

If a connection between the power supply and the LED circuit is misplaced or faulty it could result in too much or too little power. Too little could result in the transistors not operating in the correct mode or simply not enough power going to the lights. On the other hand, too much could result in burning the transistors or the lights.

If the system is supplying too much power, there are a couple ways to confirm this. You could first measure the the voltage across the base and emitter. The expected voltage should be around 0.7 volts and can be found on the corresponding data sheet of the transistor. Next, the voltage and current going through the lights should be measured. The specifications on the lights should tell what the appropriate voltage and current would be. If either of these things aren't the case, then the input voltage should be measured.

If the system is supplying too little power, then it is possible little to no current is flowing through the system. This can easily be observed by measuring the collector current of the transistors or the voltage/current through the lights. If the collector current is lower than the design expected, then it is safe to assume the power given to the lights is insufficient and therefore is the cause of the lights not turning on.

The best solution to this problem is to have a voltage regulator connected to the input of the system. This would ensure the system either receives the full voltage required or none at all. The regulator would prevent any of the components from being burned.

7.1.5.2 Faulty Light

It is possible the LED itself is faulty. The first thing to check would be the manual corresponding to the lights. It would be able to tell you the specifications necessary to operate effectively operate them.

Once the necessary specifications are known, a multimeter can be used to simply measure the voltage and current being applied to the LEDs. The instrument will either read the expected values or lower than expected. If the expected values are measured, then it is very likely a faulty light.

In order to fix this issue, we would easily order another light and replace it. If the problem persists, then it would require revisiting the initial design.

7.1.5.3 Faulty Transistor

Another possible problem could be a faulty transistor. This could be caused by too much current flowing through the collector or the manufacturing process could have made a mistake. Either way this is a common problem and needs to be addressed.

The first test would be to utilize a multimeter to measure the voltage across the base-emitter of the transistors and then to measure the current flowing through each collector. The datasheet corresponding to each transistor will show the saturation voltage as well as the max collector current.

If these values match the data sheet, then it must be another issue. If they do not match, then we will need to replace whichever transistor is malfunctioning. Not only will we replace it, but we will need to find the cause. The most common cause is the current going through the collecting exceeding the max. If this is the case, then the design will need to be revised.

7.1.5.4 Human Error

The final common error is due to human error. This can constitute things such as incorrect components, connections, or bad design. The only way to identify such an error is to thoroughly measure all expected values and compare them to the actual values of the system. If none of the above problems apply to the situation, then this is where the design and components of the system need to be reevaluated.

If the system is created with components that do not correspond to the initial design, there are a variety of possible outcomes. For example, if a different PNP transistor is used, then there are certain specifications that need to be considered. These specifications can be found in the data sheet associated with the component. Things such as: saturation voltage, max collector current and current gain should be noted. A higher saturation voltage than expected could cause the transistor to operate in a mode that is not wanted. This would result in the lights receiving the incorrect voltage and current. If the max collector current for the transistor is lower than the current produced, then the system will burn the transistor and render it useless. Lastly, the current gain could cause the current and voltage received by the lights to surpass the necessary specifications, thus burning the lights.

Another common mistake is a missing or incorrect connection made between components. The problem with this error is the results shown can vary depending on what connection is incorrect. For example, a capacitor could be bypassed and make one or both of the lights stay on. Another possibility could be a connection from the power supply to the transistors. This could result in clipping or receiving no results at all. The only way to analyze and resolve this type of issue is to carefully go through the whole system to ensure every connection is correct.

7.1.6 Kinect Sensor Testing

The Kinect sensor consists of multiple sensors that make it up and will be utilized in the overall SAFER Knights system. With multiple sensors comes multiple points of failure. The Kinect sensor consists of an RGB camera, an IR depth sensor, and a microphone array. Prior to any utilization with code related to the SAFER Knights system, each sensor that makes up the Kinect needs to be individually tested to verify full functionality. Without testing each sensor can result if a false failure of critical SAFER Knights operating code and can delay production. To test these sensors, the Kinect can be utilized with the Kinect software development kit and the Kinect developer toolkit. The Kinect software development kit is a Windows based distribution of libraries that allow easier interfacing with the Kinect sensor and allows developers to develop windows software utilizing the Kinect. The Kinect developer toolkit is an application that allows for the download of documentation, sample source code, and other Kinect oriented resources. With the combination of the two software packages, we can test the functionality of the Kinect sensor.

Figure: Kinect Sensor

The Kinect software development kit gives us the ability to allow the computer to properly interface with the Kinect sensor. The Kinect software development kit is required to support the software and source code obtained through the Kinect developer toolkit as it provides the appropriate libraries used in the given examples. The Kinect developer toolkit offers sample source code that allows us to utilize the sensors that make up the Kinect including the RGB camera, the IR depth camera, and the microphone array. The first test that is ran is the RGB camera test. To test the RGB camera, the sample source code from the Kinect developer toolkit that utilizes the RGB camera will be implemented. Using the Kinect developer toolkit, the script is located and loaded which opens a display showing the output of the RGB camera. The Figure: Kinect RGB Camera Test shows the output from the display, it details both static and dynamic objects and although it cannot be shown in the picture alone it was verified that the camera did successfully show all changes in motion in near real-time which verifies the functionality of the RGB camera which will be used for the video recording functionality of the SAFETY Knights system.

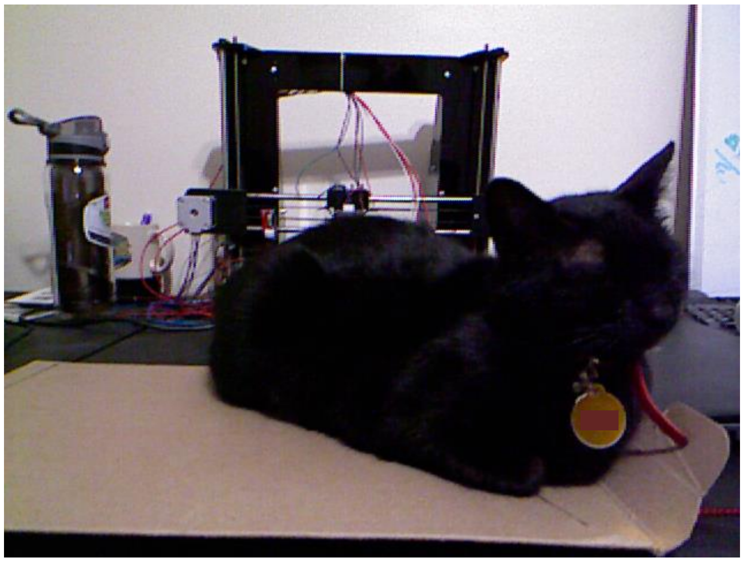


Figure: Kinect RGB Camera Test

The next test to be ran will be used to test the functionality of the IR depth camera. The IR depth camera will be used to measure the depths of objects within the field of vision. To do this the Kinect utilizes a scatter projection to determine the distance of each point by measuring the time it takes for each beam to return to the camera. It is important that the scatter projection works properly. If the scatter projection is not functioning as expected it can have unforeseen effects on further testing and implementation of the Kinect sensor. To test the scatter projection, we will implement a source code example from the Kinect developer toolkit that streams the video from the IR camera to a display on the computer. Figure: Kinect IR Camera Test shows the output from this test. The display presented the video stream in real-time of the IR camera, verifying the scatter projection is functioning and capturing updated movement of dynamic objects.

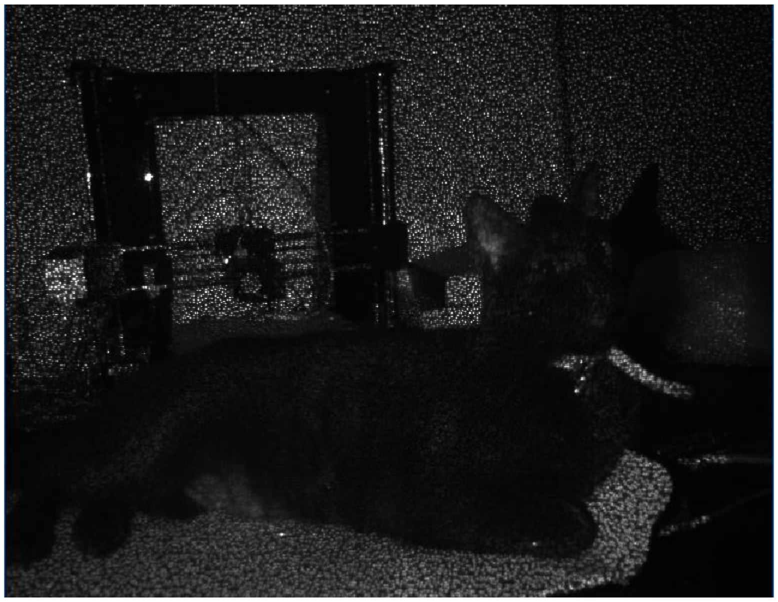


Figure: Kinect IR Camera Test

With the scatter projection verified, the next test will be used to determine if the data received from the IR depth camera is correct. The previous test although tested the scatter projection and the visibility of the IR camera it did not cover the depth measurements received from the IR camera. The functionality of the depth measurements is one of the most crucial aspects of the Kinects usage towards object detection. To verify functionality of the depth measurements received from the IR camera we will utilize another source code example from the Kinect developer kit. This example source code utilizes the IR camera to read the depth values and creates a mono color depth video stream that illustrates the depths of different objects by how light or dark the color is. The Figure: Kinect Depth Test shows the result of the test. The figure shows a single image of a depth mapped display showing which objects and surfaces are closer and which are farther. This test utilized both static and dynamically moving objects to show that the depths are constantly adjusted based on changing positions. The use of dynamically moving objects in this test is extremely important as it shows the Kinect is able to appropriately adjust depth values of an object that is moving, which will be a necessity when dealing with dynamically moving objects that need to be avoided during deployment.

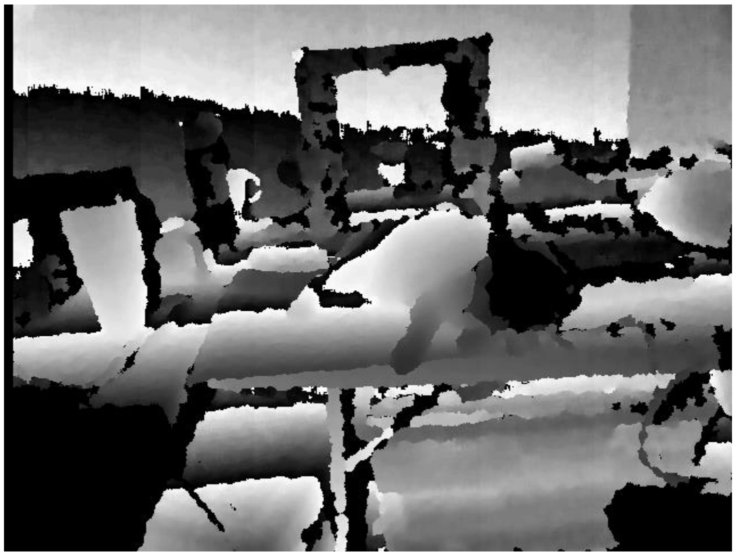


Figure: Kinect Depth Test

7.2 Software Testing

This section is a breakdown of the plans on how to test the software. Testing must be done to make sure the code not only compiles without issue but more importantly, that it does what it is supposed to. Initially, the MSP 430 will be used to test the microcontroller, but the majority of the code will be written and tested using the Raspberry Pi. Python will be the language of choice for ease of use. The code will be tested to ensure that it is capable of detecting and avoiding objects. Testing will also be done to ensure that Emergency Mode works as expected when it is expected. The mobile app will be tested to ensure that it connects to the robot and will trigger Emergency Mode when needed.

7.2.1 Object Detection

During the movement of the SAFER Knights robot, the system must be aware of any objects in its path. Testing the functionality of the object detection is integral part of verifying the overall functionality of the SAFER Knights system. Testing the object detection will require the microcontroller has an open remote desktop connection and an open image detection window. From there the Kinect needs to be pointed at static object. The image detection window should identify the object and output a distance. The next step is to test against dynamic objects. With the image windows still open, an object needs to move through the Kinect's field of vision. As the object moves through the field of vision the Kinect should constantly show the object as detected, while constantly updating the objects distance.

Now with the object detection verified, the object avoidance must be verified. A remote console to the microcontroller unit should be open. While the SAFER Knights is active and following a user, the user must walk over a static object. The image window should detect the object and output the distance as in the previous test. Now the robot must determine if the object is an appropriate size to avoid or not. The robot should now adjust its path to avoid the object by changing its direction using the servo motor to avoid the object. While still following the user, the robot needs to be tested against a dynamically moving object. As the robot moves an object needs to be moved through the robot's field of vision. The object should be identified in the image window, as well as have the distance outputted and updated. The robot should now adjust its speed and direction to avoid the object. The DC motor should be adjusted, and the servo motor should be used to avoid the object. The SAFER Knights should successfully avoid both the static and dynamic objects.

7.2.2 Emergency Mode

The emergency mode of the SAFER Knights system will be tested in different stages of development using different techniques to verify results. Initially the system will be tested by verifying that it enters and exits emergency mode. To do this in early development the emergency mode will be triggered and deactivated by a keyboard input rather than data received over Bluetooth. The reason for this being in early development having a system's functionality dependent on another system in development can cause delays in test and verification and can cause larger avoidable problems later in development. Once verified that the keyboard command causes the system to enter emergency mode, the three actions of the emergency mode needs to be tested. To verify the functionality of the emergency audio, the emergency mode will need to be initiated. If during the emergency mode, the specified audio is played repeatedly, and the audio is stopped once emergency mode is exited then the audio functionality of emergency mode can be verified. To verify functionality of the emergency lights, the output of the emergency flag that would normally be outputted to pins to send to the custom LED PCB will directed to a console output. When the emergency mode is activated or deactivated the flag will be printed to the screen. Both flags need to be verified on the screen for LED emergency mode to be verified. To test the recording flag of the emergency mode, both the playback time and file name will need to be set to static variables. From there it needs to be verified that the appropriate log file is created on the external hard drive with the appropriate data.

For later development the testing of the emergency mode will need to be altered. The SAFER Knights system's emergency mode will need to be verified. To do this a signal will need to be sent to the system over a Bluetooth connection. This will verify that the signal received from Bluetooth does indeed activate/deactivate the system's emergency mode. From there the emergency mode's audio testing will be the same as early development. The emergency mode should play the audio repeatedly while emergency mode is activated, then stop playing when emergency mode is deactivated. To test the emergency mode's LEDs, the microcontroller unit will need to send the signal through a connection to the custom LED PCB board. If the custom LCD PCB successfully receives both the activation and deactivation signals, then it will be successful. Furthermore, to test the functionality of the emergency mode's recording flag system the log file needs to be created. Within the log file the name and the playtime of the recording should be stated.

As long as these two pieces of data is recorded then the emergency mode recording flag is successful. A final test will be run to verify all systems are working properly, during this test the same procedures will be ran to test emergency mode functionality.

7.2.3 LED Microcontroller Unit Testing

The SAFER Knight's LED subsystem utilizes a secondary microcontroller unit to control all the LEDs of the overall system. This microcontroller unit is programmed separately from the main microcontroller unit of the overall system and the software needs to be tested as well to ensure proper functionality. The functionality of the software internally and its interaction with external factors need to be tested and verified to determine whether or not it functions as desired. Failure of these test will show the improper functionality of the LED subsystem and thus the overall SAFER Knights robotic system. To more easily test the software of the LED microcontroller unit, a development board will be used for development and testing to verify the integrity of the code prior to deployment to the microcontroller unit on the LED's custom PCB board. Since there are multiple IDEs that can be used to program microcontroller units, the specific one used will also determine the available debug tools to test with. Some IDEs offer a more extensive variety of debug tools available, but at the cost of a more complex development environment and increasing the complexity of the overall software.

The first test is to determine that the program compiles without any compilation errors. Compilation errors are the first inclination that the system does not operate properly and are usually well explained at compile times depending on the compiler used. With compilation or build errors the software failed to be compiled by the compiler and will not be deployed to the board. Once the system compiles and deploys with no errors then the individual parts of the code need to be tested. With the initialization of the pins the best way to test the pin outputs to verify the pins have been properly initialized is by using a multimeter on the appropriate pins to test the voltages of the initialized pins. The voltages on the initialized pins should read an approximate high value depending on the specific microcontroller unit chosen (~3V for a MSP430). By determining the appropriate values are being measured for the initialized pins, it can be inferred that the pins have been properly initialized. Testing of systems interrupts can be handled in different ways depending on the software environment being used. The TI code composer studio offers full real time debugging capabilities, while Energia does not. To test if the pin interrupt functions as intended the LED subsystem will need to be connected to the main microcontroller unit with the I2C/SPI connection established. In the pin interrupt service routine either a breakpoint or a debug statement needs to be placed in the beginning and end of the routine. This breakpoint/ statement will be used to determine if the interrupt was successfully activated and completed.

With the system loaded the signal should be sent from the main microcontroller unit. The system should then hit then breakpoint/debug message. When it is verified that the first debug point is reached then it can be verified that the connection between the main microcontroller unit and the LED microcontroller unit is successfully established. From this point the program should hit the breakpoint/debug message at the end of the pin interrupt service routine and verify that the interrupt runs successfully. If using the TI code composer studio, it can be verified that the value of the boolean variable is properly updated. The easiest way to test the functionality of the timer interrupt is by viewing the values being produce from the multimeter. Since the values of the pins are being changed at a set static interval the voltage values of the pins should be changing at the specified rate. If there is no change in voltage values on the pins a breakpoint/debug message can be placed in the timer interrupt service routine to determine whether or not the interrupt is occurring. When the voltages on the multimeter are changing at the desired rate the timer interrupt can be verified. The last part of the software to verify is that the emergency mode is properly exited. Since at this point the pin interrupt has already been verified the best way to determine proper operation after emergency mode deactivation is by using the multimeters again. Using the main microcontroller unit, the deactivation signal needs to be sent to the LED microcontroller unit. Once sent, the multimeter should read high values on all initialized pins as it did initially prior to emergency mode activation. If the values are not what is expected, it needs to be verified that all initialized pins are being set to the appropriate values in the pin interrupt service routine since the routine initialization has already been verified.

7.2.4 Speaker Software Testing

Another important part of the SAFER Knights' emergency system is the audio that is played during its activation. The speaker system itself does not have a secondary microcontroller unit like the LED subsystem, but rather is handled directly by the system's main microcontroller unit. The main microcontroller unit stores the appropriate audio and will determine when to initialize and play the audio. From there the system will utilize the microcontroller unit's auxiliary connection to send the audio to the speaker's custom PCB and then the speakers themselves. The code for this, well rather simple, does also need to be tested in order to verify the full code base for the SAFER Knights' emergency subsystem.

The speaker code will be contained within a python script. To determine initial competency the code needs to compile. If the code does not build and compile, then the is not operational. To remedy this situation the reason for the error in compilation needs to be addressed. The best way to determine the initial problem is by reading the errors returned. Usually these errors will tell where the error is occurring and why, from there the problem can be addressed. In situations like this the error would mostly have to do with syntax. The syntax of the script needs to be checked to determine that everything is spelled correctly, and formatting is correct. Since python is heavily reliant on formatting in order to properly parse code, checking the format (including spaces, indentation, and tabs) can be a solution to compilation errors. Fixing any formatting, syntax, and compilation errors should now allow for a successful build and compile. Now that the code is verified to compile, the functionality needs to be verified. Since the speaker python script is called by the SAFER Knights' emergency system, the script itself can be tested separately from the rest of the emergency system. The script is also tested independently than the rest of the speaker subsystem. The speaker subsystem consists of the speakers, the custom speaker PCB and the software handling the audio. Since the audio is transferred over a standard auxiliary connection any auxiliary compatible source can be used to verify the functionality of the software. Using either a direct connection to the main microcontroller unit, an SSH connection, or a VNC connection, the speaker python script can be ran using the command line. By utilizing a direct connection, SSH connection, or a VNC connection the system can be accessed directly, which allows for the code to be ran in the same environment in which it will be permanently ran instead of an intermediate environment where certain factors might differ such as file structure or audio codecs, and possibly provide a false positive, which in the end does not work on the permanent main microcontroller environment. Once the script has been loaded it should be verified that the specified audio file plays properly from whichever hardware audio source used. If the audio does not play and hardware problems are assured not to be a factor, then the command line should be referenced to determine if any runtime error messages are reported.

Using any reported runtime error messages can aid in the solution of the problem. If a runtime error occurs certain factors need to be checked. The path to the audio file in the code should be verified. The code should point to the specific file via a predefined file path determined once the audio file has been placed in the file system. Once placed the audio file should not be moved from the directory that it is stored in. If the audio file is moved out of the directory the file path that is defined in the code will no longer be valid. If the file path is wrong the audio file would not be found. The audio file itself also needs to be verified. The integrity of the audio file also should be checked. If the file is corrupt this could also cause the speaker python script to fail. From this point it should be verified that the speaker subsystem python script runs correctly and plays the audio via the auxiliary connection. This allows further verification of the overall Safety Knights' emergency subsystem.

8 Administrative Content

Below is a summation of both the project milestones and a discussion on the project's budget. The project milestones section enumerates each milestone, their starting date, and ending date. It includes both information about the goals set for the team at the beginning of the project's start date, as well as current progress, and then finally what is predicted to be done at certain dates in the future. The budgets and finances section discusses the proposed budget for the budget and where the cost are predicted to lie, as well as the quantity of parts. However, since some of the parts are already in the team's possession, the hope is that the budget will be greatly be reduced. The last table shows the current purchases by the team as well as the parts already in possession of the team at the start of the project.

8.1 Project Milestones

Below is a table featuring the team's project milestones that were determined at the beginning of the project timetable.

Table: Project Milestones

Assignment:	**Starting Date:**	**Finish Date:**
Divide and Conquer	September 3, 2018	September 14, 2018
Updated Document	September 14, 2018	September 28, 2018
60 Page Draft	September 28, 2018	October 22, 2018
100 Page Draft	October 22, 2018	November 16, 2018
Order Parts	October 22, 2018	November 20, 2018
Final Document	November 16, 2018	November 19, 2018
Prototype 1	January 10, 2019	January 31, 2019
Prototype 2	February 1, 2019	February 28, 2019
Prototype 3	March 1, 2019	March 20, 2019
Hardware Check	March 21, 2019	March 25, 2019
Software Check	March 26, 2019	March 31, 2019
Final Prototype	April 1, 2019	April 10, 2019

8.2 Budget and Finances

Our hope is to keep this project affordable. Below is the table that is a broad list of what was projected we might use during Senior Design 1 to help us achieve our goal. Cost can be reduced by taking advantage of used or already acquired hardware. Sponsorship will be sought out to help with financing.

Part description	**Quantity**	**Price**
Vehicle base	1	$30-40
Microcontroller	2	$0-80
LED lights	1-5	$20-100
Ultrasonic Sensors	2-10	$5-25
Microsoft Kinect	1	$0-80
Lidar	1	$500
GPS module	1	$15-50
Night vision camera	1	$40-100
Speed controller	1	$10-25
Custom PCB	5-10	$50-200
100 GB Hard drive	1	$20-50
Servo motor	1	$5-20
Total Cost		$695-$1270

Figure 2: Budget

The vehicle base, a used Power Wheels, was purchased through Facebook Marketplace and picked up from Tampa for the low cost of $25. The make is a green Lil' Kawasaki 6 Volt Power Wheels, which will be adequate for now. The seller also pitched in a non-functioning silver 12 Volt Ford F-150 Power Wheels, which would be ideal for its bigger size and bigger battery. Efforts will be made to fix this in a reasonable amount of time, and if not the Lil' Kawasaki will be used. Next are the sensors to use for object detection. For this, a Microsoft Kinect will be used, as it also has a camera that can be used for the robot's surveillance needs as well. This is cost efficient and more convenient, especially since a group already has a Microsoft Kinect, making the total cost of both the sensors and camera amount to $0. For now, no more lidar or sensors will be used unless further investigation shows that they would be useful and worth the cost.

The microcontroller chosen is not quite a microcontroller but rather a single-board computer known as the Raspberry Pi 3 Model B. This way it will have more computing power than just a microcontroller while still be cost efficient. Since multiple group member already own the Raspberry Pi, the cost is $0. Custom PCBs will need to be ordered as well and can be purchased relatively cheaply through JLC PCB. EagleCAD will be used to design PCBs as the software comes in a free version. A speaker will be purchased, and there are a variety of options available so that they can be found as cheap as possible while still achieving what we need. There are also a variety of options to find a memory device to store video recordings. Most likely, an SD card, flash drive, or external hard drive will be used depending on price and ability to connect to the Raspberry Pi. LED lights will be purchased to mount on the robot and are relatively low cost.

Table: Current Purchases Made

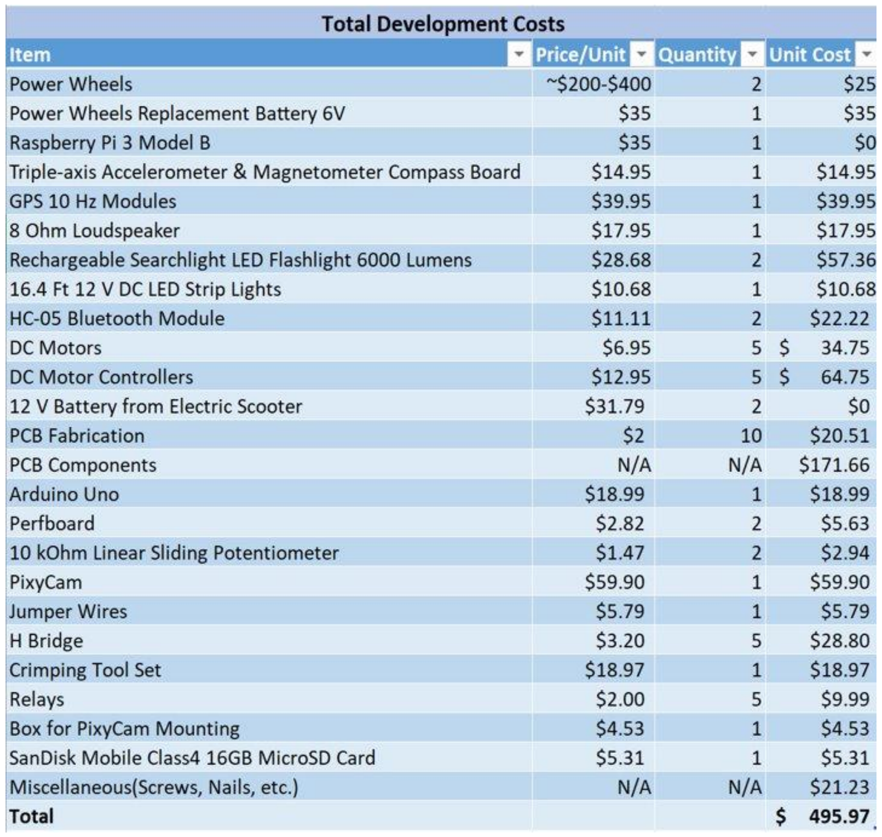
Parts	Quantity	Seller	Price
Power Wheels	2	Facebook Marketplace	$25
Power Wheels Replacement 6 V Battery	1	Amazon	$35
Microsoft Kinect	1	Team member	$0
Raspberry Pi 3 Model B	1	Team Member	$0
MSP430 Launchpad Development Kit	1	Team Member	$0
G-Drive Mobile USB 1 TB Hard Drive	1	Team Member	$0
Rechargeable Searchlight LED Flashlight 6000 Lumens	2	Amazon	$57.36
8 Ohm Loudspeaker	1	Amazon	$17.95
16.4 Ft 12 V DC LED Strip Lights	1	Amazon	$10.68
Gearmotor 6 V DC Servo Motor	2	Team Member	$0
Triple-axis Accelerometer & Magnetometer Compass Board	1	Adafruit	$14.95
GPS 10 Hz Modules	1	Adafruit	$39.95
HC-05 Bluetooth Module		Amazon	$11.11
6-24 V DC Motor	2	Skycraft	$13.90

6-40 V DC Motor Controller	2	Skycraft	$25.90
Total Cost			$251.80

As can be seen from the table above, the team is currently well below budget thanks to all used and previously owned products. Even if more purchases need to be made, such as the custom PCB from the PCB supplier, the team is on track to be on budgets for this project.

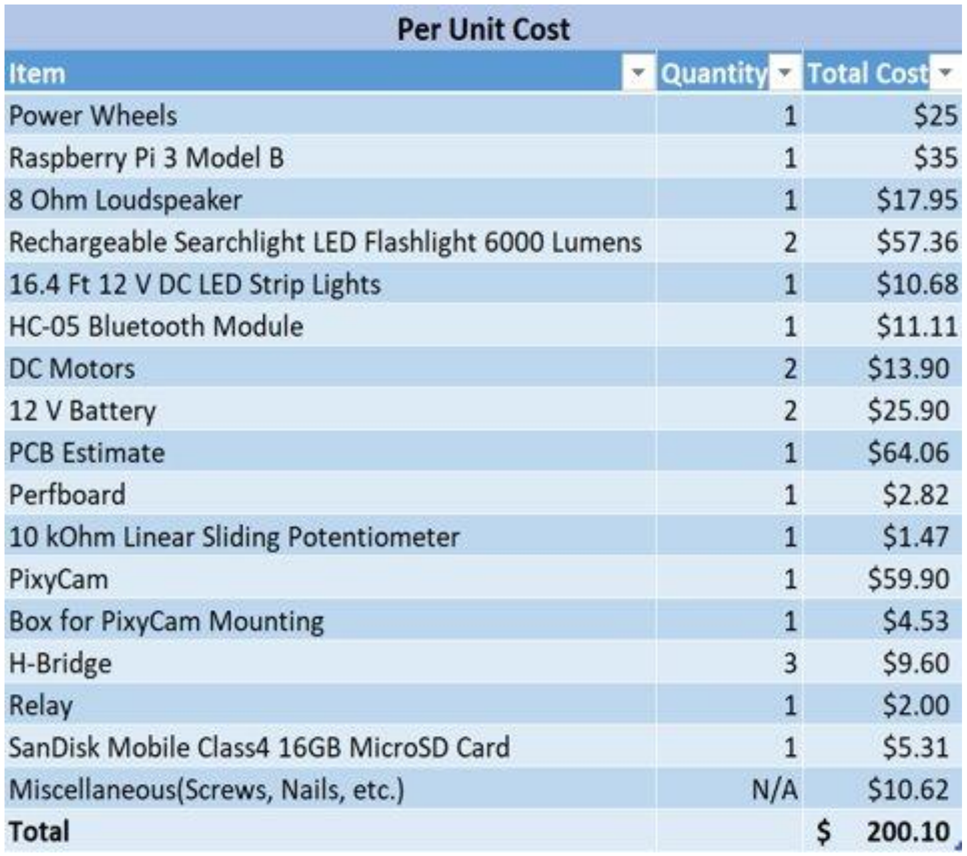
The next table shows the total costs after Senior Design 2 was completed. The first table shows the total developments costs, including everything purchased for the project. The second table shows the cost per unit, so everything that can be seen on the actual robot itself.

The next table shows the total costs after Senior Design 2 was completed. The first table shows the total developments costs, including everything purchased for the project. The second table shows the cost per unit, so everything that can be seen on the actual robot itself.



Total Development Costs			
Item	Price/Unit	Quantity	Unit Cost
Power Wheels	~$200-$400	2	$25
Power Wheels Replacement Battery 6V	$35	1	$35
Raspberry Pi 3 Model B	$35	1	$0
Triple-axis Accelerometer & Magnetometer Compass Board	$14.95	1	$14.95
GPS 10 Hz Modules	$39.95	1	$39.95
8 Ohm Loudspeaker	$17.95	1	$17.95
Rechargeable Searchlight LED Flashlight 6000 Lumens	$28.68	2	$57.36
16.4 Ft 12 V DC LED Strip Lights	$10.68	1	$10.68
HC-05 Bluetooth Module	$11.11	2	$22.22
DC Motors	$6.95	5	$ 34.75
DC Motor Controllers	$12.95	5	$ 64.75
12 V Battery from Electric Scooter	$31.79	2	$0
PCB Fabrication	$2	10	$20.51
PCB Components	N/A	N/A	$171.66
Arduino Uno	$18.99	1	$18.99
Perfboard	$2.82	2	$5.63
10 kOhm Linear Sliding Potentiometer	$1.47	2	$2.94
PixyCam	$59.90	1	$59.90
Jumper Wires	$5.79	1	$5.79
H Bridge	$3.20	5	$28.80
Crimping Tool Set	$18.97	1	$18.97
Relays	$2.00	5	$9.99
Box for PixyCam Mounting	$4.53	1	$4.53
SanDisk Mobile Class4 16GB MicroSD Card	$5.31	1	$5.31
Miscellaneous(Screws, Nails, etc.)	N/A	N/A	$21.23
Total			**$ 495.97**

Figure: Final Money Spent



Per Unit Cost		
Item	Quantity	Total Cost
Power Wheels	1	$25
Raspberry Pi 3 Model B	1	$35
8 Ohm Loudspeaker	1	$17.95
Rechargeable Searchlight LED Flashlight 6000 Lumens	2	$57.36
16.4 Ft 12 V DC LED Strip Lights	1	$10.68
HC-05 Bluetooth Module	1	$11.11
DC Motors	2	$13.90
12 V Battery	2	$25.90
PCB Estimate	1	$64.06
Perfboard	1	$2.82
10 kOhm Linear Sliding Potentiometer	1	$1.47
PixyCam	1	$59.90
Box for PixyCam Mounting	1	$4.53
H-Bridge	3	$9.60
Relay	1	$2.00
SanDisk Mobile Class4 16GB MicroSD Card	1	$5.31
Miscellaneous(Screws, Nails, etc.)	N/A	$10.62
Total		**$ 200.10**

Figure: Per Unit Cost

9 Appendices

The next two sections show where some of the information referenced and images used come from. All other images that don't mention citations were original pictures taken by team members.

9.1 Works Cited

Davis, Nick. "The History and Basics of IPC Standards: The Official Standards for PCBs." *All About Circuits*, 20 Oct. 2017, www.allaboutcircuits.com/news/ipc-standards-the-official-standards-for-pcbs/.

Shack, Hacker. "Make an Autonomous 'Follow Me' Cooler." *Hackster*www.hackster/hackershack/make-an-autonomous-follow-me-cooler-7ca8bc.www.hackster.io/hackershack/make-an-autonomous-follow-me-cooler-7ca8bc., 25 Nov. 2018, www.hackster.io/hackershack/make-an-autonomous-follow-me-cooler-7ca8bc.

"Advantages & Disadvantages of Front Wheel Drive." *Semi-Automatic Transmission ~*, mechanicalmania.blogspot.com/2011/07/advantaes-disadvantages-of-front-wheel.html.

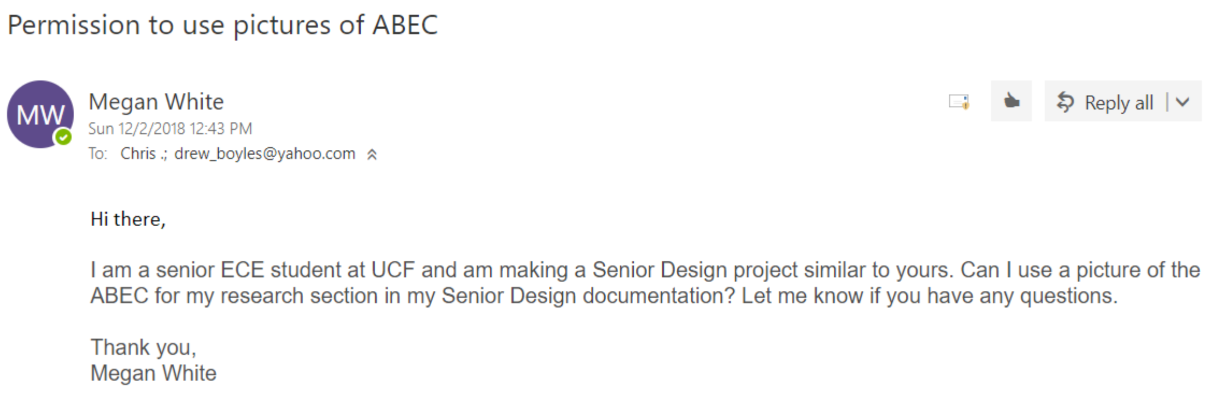
"Welcome to ECE Senior Design." *ECE Senior Design*, www.eecs.ucf.edu/seniordesign/index.php.

"Programming Languages — C." Open Standards, 2017, www.open-std.org/jtc1/sc22/wg14/www/abq/c17_updated_proposed_fdis.pdf.

python-dev. "PEP 8 -- Style Guide for Python Code." Python.org, 2013, www.python.org/dev/peps/pep-0008.

"4 Wheel Drive Vs. 2 Wheel Drive." *WheelZine*, WheelZine, wheelzine.com/4wheel-drive-vs-2wheel-drive.

9.2 Image Permissions



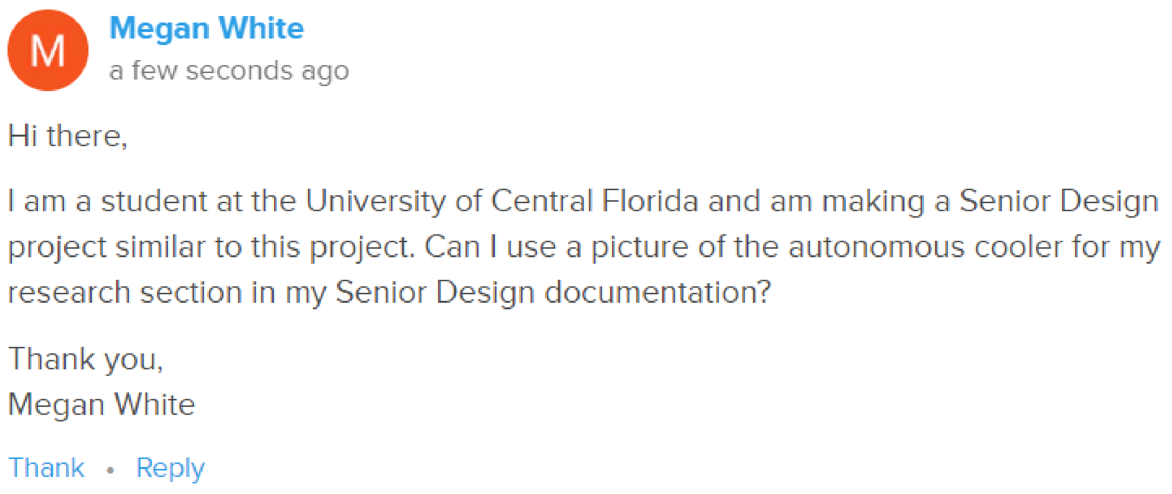
Permission to use pictures of ABEC

Megan White
Sun 12/2/2018 12:43 PM
To: Chris .; drew_boyles@yahoo.com

Hi there,

I am a senior ECE student at UCF and am making a Senior Design project similar to yours. Can I use a picture of the ABEC for my research section in my Senior Design documentation? Let me know if you have any questions.

Thank you,
Megan White



Megan White
a few seconds ago

Hi there,

I am a student at the University of Central Florida and am making a Senior Design project similar to this project. Can I use a picture of the autonomous cooler for my research section in my Senior Design documentation?

Thank you,
Megan White