Knight Brawlers

Senior Design I
Summer 2013

Group 1
Allen Davila
William Allen
Carlos Davila
Joshua Thames

Table of Contents

1 Executive Summary

The following documentation describes the details and specific information going in to the motivation, goals, desired specifications, investigations, detailed design, and planning in order to complete the Knight Brawlers project for Senior Design. The project is to be achieved with the cooperative efforts of Joshua Thames (Computer Engineering undergraduate student), William Allen (Electrical Engineering undergraduate student), Allen Davila (Electrical Engineering undergraduate student), and Carlos Davila (Electrical Engineering undergraduate student) with the supervision and guidance of the University of Central Florida department of Electrical Engineering and Computer Science. The coursework is ultimately aimed at satisfying the requirements established by the Accreditation Board for Engineering and Technology (ABET) for all graduating electrical and computer engineers from an ABET accredited university. Although the project details will be expounded upon in this document, a brief overview will be provided.

The project function of Knight Brawlers itself is to provide a fun gaming experience that utilizes remote control cars that engage in combat. The cars will be controlled by the user through their smartphone accelerometer while giving a live feed of the action as seen on the cars themselves. The objective of the gamers will be to reduce the health of the other cars by bumping into them and diminishing their LED status. An android app will be designed in order to provide an appeasing user interface that enhances the gaming experience and the ease of use.

The whole project will employ several subsystems that will work in tandem with each other in order to provide an entire, fully functional game. While each subsystem serves an individual function and will not preform every task, the project cannot be achieved without each subsystem doing their job in an effective manner.

From the conception of the Knight Brawlers idea, the objective was to provide a manner of combining the interests of the individual group members into a project that would be approved as two semesters worth of work for senior design. A few of these areas of interest include power design, embedded systems, PCB design, mobile device incorporation, and sensor integration. It was also kept in mind that the project would provide a fun and enjoyable experience in the design and building aspects. The ideal situation will be to have four fully functional remote control cars so that each member of the group may partake in the action.

2 Project Description

2.0 Project Requirements and Specifications

The initial requirement of a Senior Design course for electrical and computer engineering students was to satisfy the requirements for ABET. The Accreditation Board for Engineering and Technology wanted to have students exposed to real world experience in building and designing.

While designing and building a project, there are a number of requirements and specifications that the Knight Brawlers team must abide by. Some of those project requirements and specifications are set forth by the University of Central Florida electrical engineering and computer science department and others are set forth by the team members.

A couple of things required by the EECS department's Senior Design course had to be met in order to pass. Before beginning research and coming up with project details, the Knight Brawlers team had to have the project idea approved by the Senior Design instructor. In order to meet approval the project had to have a certain level of complexity. This was to insure that there would be two semesters worth of work for a team of four members. The project idea also had to be relevant enough to the electrical engineering and computer engineering coursework taken by all four members in order to demonstrate the ability to apply the knowledge gained through the UCF EECS curriculum to an actual project. The next requirement is to have an extensive amount of documentation of the process of designing and building the Knight Brawlers project. This includes initial documentation, presentations, conference paper, and final documentation for Senior Design I and II. The final documentation for Senior Design I must be thorough and number a minimum of one hundred and twenty pages. As for the project itself, Knight Brawlers must be able to function and meet the core specifications. Full functionality must be demonstrated to the department and meet approval. A couple of more technical details are also required of the Knight Brawlers project. The team is allowed to use a development board in order to build the project, but must use its own printed circuit board design in the final iteration. However, it is allowed to use development boards in the other remote control cars since there will be more than one car. The team must also use approved materials to build the project with.

A couple of the specifications were also set forth by the Knight Brawlers team members. These specifications were created by the team and were made in order to meet project approval. The specifications and requirements are as follows:

- ❖ The RC cars are to be controlled with a mobile device's accelerometer.

- The RC cars must have a live video feed that provides a first person view to their mobile device.
- The RC cars must be able to sense an impact from another car.
- There needs to be an application on the mobile device to provide a user interface.
- The RC cars must have a power system that will adequately power all sub-systems.
- There must be an established scoring system.

In addition to these required project specifications, there are also a couple that are desired but not crucial or necessary to the completion of Knight Brawlers. These include a remote control car built for each team member, extra LEDs for the remote control cars, and an extra level of intricacy and sophistication to the mobile device application.

2.1 Significance

The significance of designing and building the Knight Brawlers project depends on perspective. To the team members themselves, Knight Brawlers will be significant because it will be the first real project built by each member that will actually have real ramifications. These ramifications include potentially impressing companies seeking electrical and computer engineers and building something that will meet the approval of an organization, in this case, the EECS department. It is also significant in terms of using software, components, and techniques that are used in industry. Exposing the team members to these types of tools will help in building experience and insight into a career in electrical or computer engineering. Furthermore there is significance in working as a team. In the real world and in industry, working cooperatively as a unit is an important characteristic of successful employers and employees. Knight Brawlers will have to be designed and built by a cohesive unit in order to achieve the requirements and specifications discussed earlier. Lastly the project may provide some level of significance, depending on how successful it will be, to future electrical and computer engineering students. This is because it could provide relevant research and techniques to future projects, much like previous projects have been significant in influencing and helping create the Knight Brawlers project.

2.2 Division of labor

The division of labor (figure 2.2) for the Knight Brawlers project was decided upon with having fairness and individual motivations in mind. It is important to have different members in charge of different systems in order to assure that everyone is held accountable and that there are no areas of the project that are neglected. It was also agreed upon that although each member will be responsible for research in separate areas, work load will be shared and team members will assist each other with their subsystems and work together as a team.

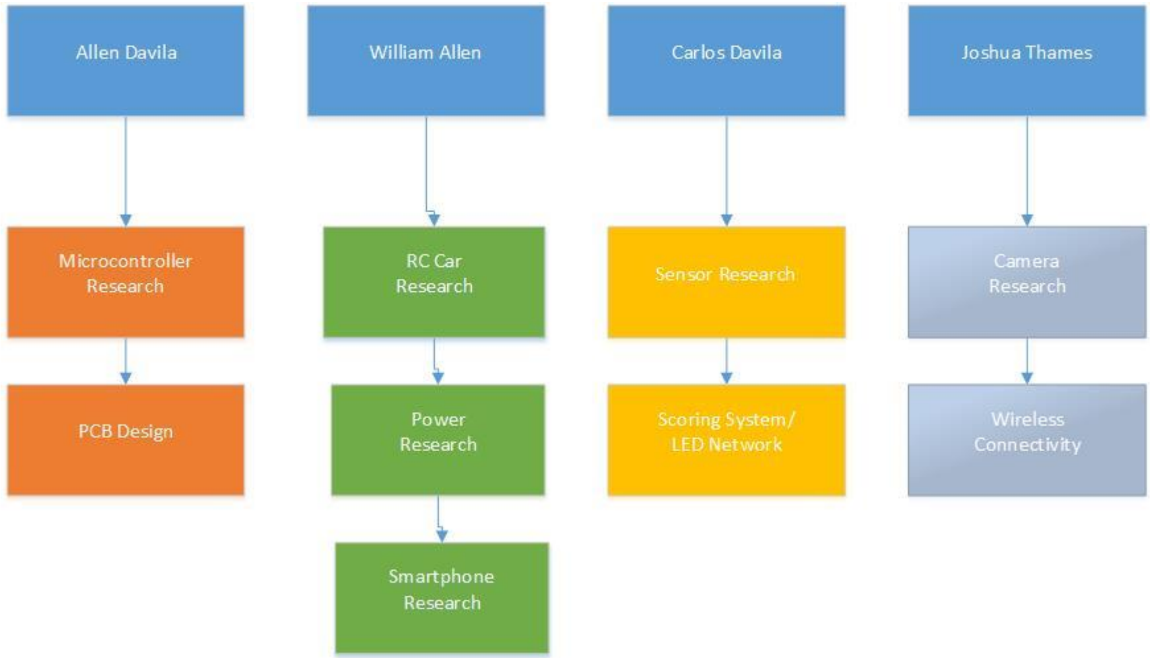


Figure 2.2 Division of labor between team members

2.3 Goals and Motivations

Various goals and motivations are going in to the Knight Brawlers project. These aspirations have been the driving force for the desire for a successful Senior Design project. Some of the goals and motivations are different for each individual team member but there is one central and ultimate objective and that is to successfully build a project that will allow us to finish our engineering coursework and graduate with our Bachelors in either computer or electrical engineering.

The individual goals and motivations are different to each team member and are essential for a fun and working project. Carlos Davila is an electrical engineering student looking to get some experience in building and troubleshooting circuits while also integrating different systems to work together. In order to meet this goal, he will be designing the sensor and LED network which is basically two systems that need to work well together to create a scoring system for the project. Both systems involve circuitry and working together to keep score. Joshua Thames is a computer engineering student looking to get some knowledge in image processing and wireless communications. To meet this goal, the project includes a live video subsystem as well as the need for wireless communication between the car and a mobile device. William Allen is an electrical engineering student that is looking to get some experience in powering subsystems and developing software. To achieve this, he will be handling the power research and designing the mobile application for the mobile device. Allen Davila is an electrical engineering student looking to get experience in embedded devices. Because of this he will be handling the microcontroller research and will handle most of the printed circuit board design.

In addition to these individual goals, the Knight Brawlers team members also share some common goals, these include the experience of working as a team, learning to put together a comprehensive document that details the entire project, and experience in building and designing from the ground up.

3 Research Related to Project Definition

3.0 Existing Similar Projects and Products

Several projects and products have existed for years that have had similarities to the Knight Brawlers project. Gaming products that utilize remote control have had many different functions and offer a new level of experiencing a game while delighting hobbyists, youth, and casual fans alike. Coupling that with the wide range of uses that have been added to RC cars over the years, the Knight Brawlers project was able to draw knowledge from various sources.

In the process of deciding upon the premises of the Knight Brawlers project, existing products and projects were very helpful in coming upon inspiration for added features to the project specifications. Also it added the benefit of observing various techniques that may be useful in designing different aspects of the project. It should be mentioned though that when looking upon older projects and products, the fast paced evolution of technology must be taken into consideration. Something that may have been impossible, sluggish, or difficult to manage may be approached in a different way with the added technologies that have progressed over time. Looking at past product and project documentation, it is also possible to find problems or delays that past groups have experienced before. This helps allot extra time for areas that may pose a problem or avoid making the same mistakes that have been done in the past. In looking at those past errors, the Knight Brawlers project will hopefully experience a smoother building process and could have the benefit of working more efficiently. When looking at professional products that may have similarities, there is also the added bonus of looking at the work of professionals. A few of the projects and products observed during the research process will be discussed further.

3.0.0 Existing Products

Bump ‘n Chuck made by Kid Galaxy is a similar gaming idea to Knight Brawlers in that its primary objective is fighting vehicles. With Bump ‘n Chuck there are two remote controlled cars that are driven around by a pair of users. When one of the cars has been crashed into or sideswiped, the other car’s toy driver is ejected from the car. The interesting feature in this product is that once the other car has been hit, a noise is activated that adds to the experience of the game. This is interesting because, one of the boards that has been strongly considered for this project, the beagle bone black, is able to play audio and perhaps can be programmed to add noise when one of the Knight Brawlers is hit. That would be an added feature that could enhance the project if there is extra time for additional project expansion.

There are also a number of different boxing robot types that have been put out by various manufacturers. The most relevant kind are the ones that utilize a scoring system in which a number of hits decided whether the user is unable to continue. This served as the inspiration to the fighting system that is intended to be implemented in the Knight Brawlers project. Much like the robots, three hits in one gaming session means you are no longer able to compete. LED's that indicate health status also closely resemble features of a lot of fighting robot products.

WiRC is also a product that has been very useful in researching ideas for a project. Stumbling across this product, the idea of being able to have a first person view of the action from the remote control cars to your smartphone was added to Knight Brawlers. WiRC uses wifi to connect to an app on either an apple or android smartphone that displays the cars movements streamed live from an onboard camera. WiRC utilizes a pleasant user interface that makes controlling the car enjoyable. Although an app and movement commands from your smartphone were already considered in the Knight Brawlers project, the WiRC product proved that it could be done in a reasonable and effective manner.

3.0.1 Existing Projects

A trove of projects have been designed and developed for Senior Design. Sifting through the previous coursework is an important and useful method of research because it gives a perspective from fellow electrical and computer engineering students.

Knightro Kart was a UCF Senior Design project that spanned the fall 2012 and spring 2013 semesters. Their project involved many similarities to the Knight Brawlers project. These include: controlling remote control cars from your mobile device, using an android app, and utilizing Bluetooth as a means of communication. Knightro Kart actually provided much inspiration in conceiving a project idea. Looking at their documentation, a large amount of reference information was made available for the Knight Brawlers project. One of the most important things found was the trouble they had with certain aspects. They had an overheating issue with their h-bridge, which helps control the remote control car's movement. In order to fix this problem, the Knightro Kart team used heat sinks to absorb unwanted heat. This could prove useful when searching for an appropriate h-bridge to use on the remote control cars. It would be wise to keep in mind to use h-bridges that withstand more heat. Another problem they encountered was that their printed circuit board was a bit too big for their vehicles which lowered the quality of use. To combat this, a possible solution may be to choose larger remote control cars for the Knight Brawlers project. That could improve the quality of play and the overall aesthetics. Rather than having a lumbering printed circuit board making the remote control cars looking overwhelmed, it could look more appeasing in terms of proportions. This is important because the wow factor of gaming products such as Knight Brawlers play an important part in desirability. Furthermore, another problem encountered by the Knightro Kart team was syncing issues between vehicles. An improvement upon this problem will make play more enjoyable for the users, rather than wasting time on merely trying to start a gaming session. The last relevant issue that posed a problem for the previous team was that they did not combine the headers on the printed circuit board which cause difficulty in wiring. When a limited amount of space is an issue on the remote control cars, this is an important thing to avoid. That includes the possibility of saving time in doing so.

Another project used as a reference for designing the Knight Brawlers project was the fall 2011 - spring 2012 RC Ghost Rider project, another UCF student endeavor. Again this project utilized a remote control car as one of its main features. They designed a cockpit in which a person would sit down in and experience a range of sensations that were to emulate what was going on with the remote control car. The most interesting part though was the video streaming they had going from the remote control car to the cockpit. This is very similar to the video streaming that is desired in Knight Brawlers, although in the case of RC Ghost Rider, it was streamed to a monitor rather than to a phone. They chose to go with a camera and receiver set that did not involve them having to encode and decode video. So although that option is not the intended method of achieving wireless streaming in Knight Brawlers, it was considered and approved as a failsafe in the event of an inability to solve a suitable system for live video streaming. Another good source of information gathered from the RC Ghost Rider project was that they encountered trouble with their accelerometer jittering and synchronization between the two main systems. This provides a good look at what to watch out for in designing Knight Brawlers.

The last project that was a useful reference was Autonomous Optical Guidance System made by University of Central Florida students during fall 2012 – spring 2013. The most interesting part of their project is the development board they chose, the STM32F4 discovery board. The STM32F4 has been eyed by the Knight Brawlers team as the most suitable development board for the project. Knowing that it has been successfully used in the past by another senior design team is a big factor in boosting the team's confidence in using the board. This is because it lets us know that there will be plenty of documentation available for the board including gerber files and schematics. Also it lets the team know that it is possible to sample and acquire the microprocessor that the board uses and other vital parts. The Autonomous Optical Guidance System team also did not document any trouble they had with the STM32F4 which is a major plus.

In addition to various project specifics, some general problems that have arisen from most if not all previous UCF senior design projects have affected planning for Knight Brawlers. This would include difficulty of parts acquisition, poor time management, insufficient testing, and lack of cooperation between team members. These are all significant issues that the Knight Brawler team hopes to avoid by learning from the past mistakes of others.

3.1 Possible Architectures and Related Diagrams

Designing part locations is an important piece of the project's development. It is important to arrange everything with the foremost concern being functionality and aesthetic value as a secondary consideration. The first decision to be made was not really much of an option. The motor location would have to be near the wheels because it provides the most stable and practical configuration. The next location consideration to be made would be that of the printed circuit board. The board location all depended on what type of vehicle would be used. The Knight Brawlers team considered buying a remote control truck to possibly place the board in the bed of the truck. That however did not seem like a good idea because no suitable trucks were found. The second idea was to mount the board on the roof of the car, but the team agreed that it would make the cars look less appeasing visually. The decision was then made to buy a larger scale remote control car in order to try and fit the board within the vehicle itself. The next design question was where to place the camera. The Knight Brawlers team wanted the video feed to best match that of being in an actual car. Therefore, the decision was made to mount the camera to the hood of the remote control car so that it would give the most realistic first person view as possible on their mobile device. The next consideration was that of the sensors. The team wanted to have various hit points so the decision was made to place one on each side and one on the rear of the remote control car. The last consideration was the easy decision to put the LEDs on the roof of the car to provide the best visibility. Putting together all those ideas together (figure 3.1a) made the remote control cars easy to imagine and document.

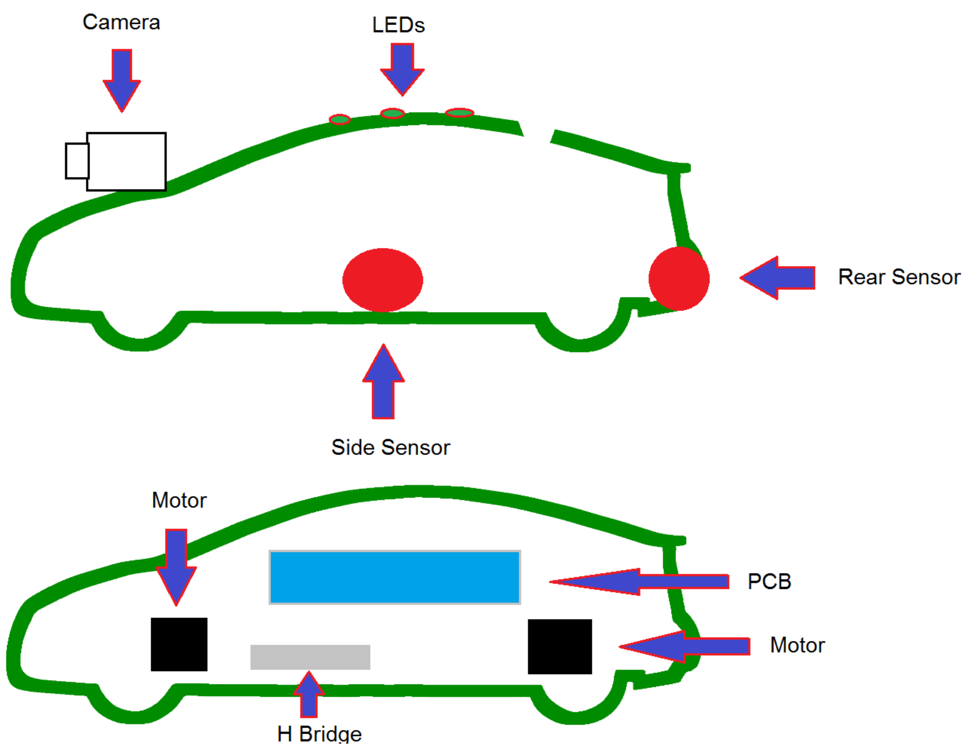


Figure 3.1a External view of car (top) and internal view of car (bottom)

Once a general picture was created, the various subsystems that would be needed were discussed. It was important to cover every area of the project in order to effectively research each subsystem. It is also helpful to visualize how the subsystems will communicate and depend on each other (figure 3.1b).

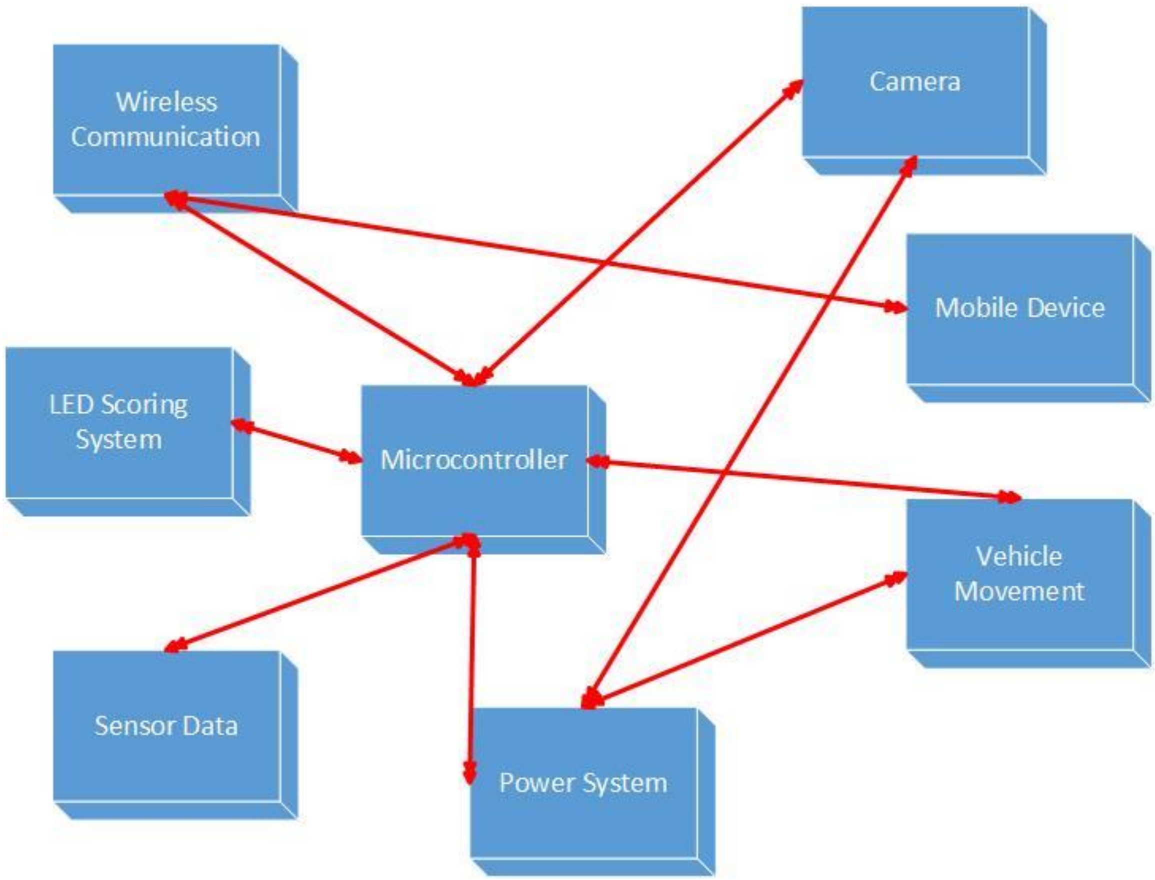


Figure 3.1b Subsystem block diagram

3.2 Project Ideas Considered

There were several project ideas considered during the brainstorming process. Some of these ideas are still realistic possibilities in case the plans for Knight Brawlers have to be changed or augmented. The Knight Brawlers team considered making the project in to a real product, one that perhaps could be marketed and produced. This seemed unrealistic though because the market for similar devices is already flooded with similar products. Another idea considered was receiving a sponsor to assist with the resources and project financing. The Knight Brawlers team looked into perhaps using a solar cell to power or at least add some extra battery life to the remote control cars, in hopes of getting an alternative energy sponsor. In the past, UCF Senior Design projects have received funding from alternative energy companies for merely adding a solar cell for a boost of battery life. The Knight Brawlers team however decided not to take this approach however because the remote cars would already have several components on them. Adding more weight and taking up more space could potentially compromise the project.

4 Project Hardware and Software Design Research

4.0 RC Car Research

4.1.0 Choosing a Remote Controlled Car

While choosing a Remote Controlled (RC) Car it while be difficult to decide what size car and what type of car to use that will fit our project specifications exactly. With the Knight Brawler, we will need as much space as possible to fit all the additional hardware we will like to add to the car. With that said price will play a big part in our decision making process. Prices will vary with what brand, and the additional features that are put in to the remote controlled car. The size of the car is one of the major factors in how much the RC car will cost, so that were we will put our effort in to get the best size RC car that will have all the size needed to put all the materials and wiring inside the car so it can look as neat as possible.

4.1.1 Remote Controlled Car Dimension

We decided to be able to make the Knight Brawler the best we could make is to some requirement needed to be set. Those requirements were:

- Wheels Covered by Frame
- At least ¼ of a inch of free space around undercarriage and frame
- Less than 4 lbs
- The undercarriage has a lot of extra space
- Non-metal Frame
- Cheap as possible

With those parameter, we plan to find the perfect remote controlled car for the Knight Brawler and be able to make all four of the car systems that we need to make the Knight Brawler a full function game that can we be played anywhere on any device.

Remote Controlled Cars are measured on the actually size of the car it is modeling if any. Then, the cars divided a number to get a usable scale base on the full size car. So, Remote controlled cars with the size scale of RC cars as shown in figure 1 are can be heavily detailed with the seat and adjustable part to can make the RC car look better and more accurate to the full size car.

1/10 Size: The 1/10 model RC car size is the largest dimension vehicle that we are looking into for the Knight Brawler. The upside for using the 1/10 size is that it has the most space to work with. This will give us a clean, attractive look to the RC Car because there aren't any spare wires or part that will hang over or on top of the car. Also, with this size car the heat from the processor and other component will be less of a problem because of the air the will be able the room through the car and the probability to add fans and heat sinks with the extra space. The downsides to this RC car is that it heavy and may cause a problem when we add the components, but along with that it is the fastest car so it may be able to support itself or with the size of the RC car we may always be able to add more powerful motor that can support the weight of all the components.

1/12 Size: The 1/12 model RC car size is the second largest dimension vehicle that we are looking into for the Knight Brawler. The upside for using the 1/12 size is that it will be usually detail with more attractive accessories for collectors. Even through this will give us more space to work with depend on the what manufacture give us a good amount of room to put component inside of it, but with these size RC cars the 1/12 RC car may come from it may be very difficult to extract the detail accessories out of the car or even damage the car. With these size RC Cars, the frame of the RC car will be usually about the same as the 1/12 on the inside frame. This is to not allow any spare wires or part that will hang over or on top of the car. Also, with this size we maybe still be able to add a fan or a cooling system to keep the cars from overheating or something worst. The downsides are once again that the detail accessories to the RC car may cause problem with damage to the car and unnecessary risk that we may not be willing to take. On the same hand, the RC Car will is smaller than the 1/10 size RC car because it is not make for performance. But the size of the RC car we may always be able to add more powerful motor that can support the weight of all the components and everything else.

1/14 Size: The 1/14 model RC car size is the third largest dimension vehicle that we are looking into for the Knight Brawler. The 1/14 RC car model size is very similar to the 1/12 RC car but it is smaller than the 1/12 RC car. This size has about the same upsides and downsides with the car except for a few exceptions. For example, The RC car will probably not be able to mount a fan or a cooling system inside the car, which will cause problems for us and will look sloppy and not very well organized. Also, on the other hand the RC car is light for its size, but will not be able to carry to many components inside it. With the weight of the car of the components needed for the knight brawler this car will struggle to car all everything and be able to control the car.

1/16 Size: The 1/16 model RC car size is the smallest dimension vehicle that we are looking into for the Knight Brawler. The upside for using the 1/16 size is that in is the lightest RC car out of the one we had to pick from. With the small size of the car, it had more size to space to work around with the 1/12 scale and the 1/14 scale as long you pull out the accessories. It will be about the same as them. T Some of the downsides will be will this size car it will not give us a clean, attractive look to the RC Car because the spare wires or part will hang over or on top of the car. Also, with this size car the heat from the processor and other component will cause more problems because of the air the will not be able to go through the car or will not have the probability to add fans and heat sinks with the extra space. The upsides to this RC car is that it light and may cause a problem when we add the components, but along with that it is the second fastest car so it may be able to support itself but with the size of the body it will be hard to add different components with the size limitation with this size RC car. See table 4.1.1 for size comparison.

1/16	1/14	1/12	1/10
11 .75 inches	13.4	15.66	18.8
3.6 inches	4.11	4.8	5.76
3 inches	3.4	4	4.8
Usually not detailed	Usually detailed	Usually detail	Usually not detailed

Table 4.1.1 - RC Car Size Chart Based on a 12’ Ford Mustang GT

4.1.2 RC Car Undercarriage and Frame

The RC cars frame and undercarriage is a great deal to us as a group, because we need to be allowed to put as many components and parts in our RC car without losing the quality that we need to make that Knight Brawler. So, we decided that the best way to do this is to put a manufacture as a constant so we know we are getting a good quality car with a standard that we need to have for each one of the Knight Brawler that we make. This was a hard task to find data for because there isn’t any data sheet or website that tells the consumer how larger is the undercarriage or how deep the frame is. The RC Car manufactures that we decided to use for this trial run were:

- Jada Toys
- New Bright

- Wergs Racing.
- X Factory

From the previous point, we decide to purchase a number of different sizes and manufactures, and then unscrew parts from the body off the RC car to see which would be the best fit for this project. So, we picked from three of major RC car manufactures in the world. We find that all the cars are built very different from each other with a few standards in place.

Jada Toys has the whole undercarriage cover in most of their designs which even though it is a good thing for mounting component to the car, but can cause difficulty in the long road because it will make thing tight in some areas and have to push it into area or force us to have to make modulation to the RC car which may end up hurting. For the Frame for the Jada Toys, it doesn’t have much depth to it with will allow us to add a camera and other components that be need to build the Knight Brawler. With the Jada Toys RC car the cars were very nicely detailed on the outside which give us an attractive look that we are looking for in the Knight Brawler.

New Bright doesn’t have much of the undercarriage cover in most of their designs which even though it is a good thing space for mounting component to the RC car, but can cause difficulty in the long road because it will make things too heavy for the frame to support some of the loads that we are trying to add to the RC car and big space in some areas may have too much air with in some area or force us to have to make modulation to the RC car which may end up hurting. For the Frame for the New Bright, it does have a lot depth to it which will allow us to add a camera and other components that will be need to build the Knight Brawler. With the New Bright RC car the cars were also a very nicely detailed on the outside which give us an attractive look that we are looking for in the Knight Brawler.

Wergs Racing has a couple of gaps in the undercarriage cover in most of their designs which is a terrible thing to deal with for the Knight Brawler even though it probably be made in to a good thing for mounting component to the car, but will be more problem then it is worth because we will have to do a lot to modify the RC car to how we want to use it for because it will make thing awkward in some areas and have to push it into area or force us to have to make modulation to the RC car which may end up hurting. For the Frame for the Wergs Racing, it have much more depth to it with will allow us to add a camera and other components that be need to build the Knight Brawler. Also, with the Wergs Racing RC car the cars were not very detail as a whole on the outside which didn’t make it very attractive for us to us for the Knight Brawler.

X Factory has the whole undercarriage cover in most of their designs which even though it is a good thing for mounting component to the car, but can cause difficulty in the long road because it will make thing tight in some areas and have to push it into area or force us to have to make modulation to the RC car which may end up hurting. For the Frame for the X Factory, it doesn't really a have a good frame for us to use because it is very brittle and that will not be very good for the knight Brawler because it will crashing into other cars and we don't want to have parts falling off due to the collision that the RC car may be taking. Also, this RC car was made more for performance and professional use so it is built for speed not so much for durable which is the basis for the Knight Brawler itself. But with that the frame has some depth to it with will allow us to add a camera and other components that be need to build the Knight Brawler. With the X Factory RC car the cars were also not very nicely detailed on the outside which is not very appealing to look and which is not what we are looking for in the Knight Brawler.

4.1.3 Remote Controlled Car Price

The price for the Knight Brawler is very important to our group because with no sponsor or any financial support from any organization to help us get the supplies we need to buy everything we need for the Knight Brawler. So, we decided to split the money for the Knight Brawler and which will allow everyone of us to keep an RC car and whatever extra parts that we are working on to give us some experience for the project and in our life. With that said, we need to make the Knight Brawler as cheap as we could make it without losing attractiveness and not have a cheap look to it. As is shown on in Table 2, is the price of the RC car so, we decided to spend a maximum of 500 totally on the entire project after testing and initiation evaluation.

As we researched this issue of price, we find have to decided how much we were willing to spend on just the four RC cars for the Knight Brawler Because of the probably expense cost of the other equipment that will be need to have our project turn out to be a success. We have to spend the less amount of money on the RC car again without losing how the look of the car will look and the official game to play with and be a great looking vehicle that will impressive the competition. The most we are will to spend is about 40 dollars each and 160 dollars for all for four of them. This will allow us to have 340 to spend on the other accessories and parts that we need for the car.

	Jada Toys	New Bright	Wergs Racing	X Factory
1/10	$35-60	$25-40	$35-60	$35-55
1/12	$25-45	25-40	25-30	$23-40
1/16	$15-30	$20-35	$30-50	$15-23
1/18	$10-15	$13-20	$10-15	$11-18

*Table 4.1.2 - RC Car Manufacturer Prices *Prices may vary*

4.2 Microcontroller Introduction and Purpose

In order to filter through the litany of available processors, it is important to be able to try and choose one that fits the needs specifically for Knight Brawlers. The processor chosen will be critical to Knight Brawlers' functionality from power consumption, speed, performance, I/O's, and several other features necessary for functionality. The requirements for the processor are as follows:

1. Wireless capabilities: This is perhaps the most important area of focus for the Knight Brawlers project. Since the gameplay functionality for Knight Brawlers is via Android, it is detrimental to the project to be able to have a controller with great wireless capabilities to handle the transmitted data. The controller must not only be able to receive data but also send the data as well.

2. Speed: The Knight Brawlers' controller will be handling real time data from an Android device for vehicle navigation and from its bumper sensors. The steering and speed for navigation have to be handled quickly and accurately by the controller for optimal vehicle navigating precision. Data from its bumper sensors must also be handled quickly to update the LED network for health status and scores in the Android app.

3. I/O's: It is necessary for the controller to have a generous amount of general purpose inputs and outputs for its features. The GPIO's are needed for Knight Brawler's awesome LED network, wireless module, bumping sensors, camera, vehicle motors and steering.

4. Support: The controller for Knight Brawlers must have great technical support and documentation to help in its functionality integration. It must be a first time user friendly.

5. Cost: Since Knight Brawlers will be self-funded, a low cost for the controller will be very important. Not only should the controller be of a low cost, but it would be a great plus if the shields, boosters, or modules integrated with the controller are of low cost as well.

6. Performance: Finally, performance is a very important choice for a controller. The reliability and response of controller is critical for long term use of Knight Brawlers. For all the bumping fun to go on forever, the controller must be well known for its reliability and performance in any environment. Heat caused by the motors will be an issue that the controller must be able to operate in, its temperature operability will be crucial.

After reviewing Knight Brawlers' processor requirements, five units come up as possible selections. They are the ATmega328 by Atmel, the MSP430G2553 by Texas Instruments, the ARM1176JZF-S by ARM, the STM32F407VGT6 by STMicroelectronics, and the Sitara AM3359 ARM Cortex A-8 by Texas Instruments. These selections meet the requirements for our processor but only one will be utilized for Knight Brawlers. The processors and the development boards that feature them will now be compared so that the optimal processor is chosen for Knight Brawlers.

4.2.1 Microcontroller Research

ATmega328: The ATmega328 is a 16 MHz, 8 bit, 2 KB memory processor by Atmel utilized by the Arduino Uno development board. The Arduino Uno is a popular development board due to its user friendly community support, power, and array of open source hardware available. There are a various additional hardware options for wireless connectivity for the Arduino Uno. Including Wi-Fi and Bluetooth options. It has 20 I/O available, 14 digital and 6 analog. ATmega328's I/O count makes it a very strong candidate to fill the project requirements for Knight Brawlers. It's I/O's on the Arduino are configured so that they can easily be jumped to a breadboard while also being easily configured to support male pin hardware.

These female header pin connections are a fantastic feature to add wireless hardware to the Arduino to test the ATmega328's connectivity. Six of its I/O are available to provide PWM output which will benefit the motor operation of Knight Brawlers. Its pins are capable to produce a max current @40mA and they operate at 5 V. The Arduino uses the Integrated Development Environment software for programming via USB and can be linked via UART. It uses a subset of C for its programming language.

The Arduino's Integrated Development Environment software (IDE) comes with a lot of example programs and has a user friendly configuration to make programming the Arduino and its ATmega328 processor simple. The 32kB of Flash will easily hold all the code needed for Knight Brawlers. These features of the Arduino will allow for the project to completely test the processor requirements for the ATmega328 processor. The max operating temperature for the ATmega328 is 185 degrees Fahrenheit, which is high enough to be able to handle heat from the environment, from the motors and from the batteries. Due to its development board, its I/O count, wireless options, and ease of programming the ATmega328 is a strong processor candidate for Knight Brawlers.

MSP430G2553: The next option is a MSP430G2533 microcontroller by Texas Instruments. It is a 16 MHz, 16 bit, 512 Byte processor. It is utilized by the MSP430 Launchpad available by TI for $9.99. The Launchpad includes a USB cable that connects to the board for programming in C or assembly for quick prototyping of Knight Brawlers. The MSP430 has 16KB of flash memory available for programming. The MSP430 uses Code Composer Studio for its programming functions. CCS has a great debugging function, allowing users to quickly decipher issues with their code. It is free charge to program the MSP430 and has good documentation available. Energia Integrated Development Environment is also available for the MSP430, which has open source options and community support. The MSP430 comes with TI's community support, which offers trouble shooting forums and solutions.

The MSP430 Launchpad has booster packs for Wi-Fi and Bluetooth connectivity. It can easily connect to these booster packs through its 20 pin socket. The MSP430 has 8 analog and 8 digital pins (2 pins can be configured as PWM output through the MSP430's TimerA), and up to 24 GPIO depending on how the device is configured. The pins have a max diode current of plus/minus 2mA. The MSP430 Launchpad comes with female and male headers for convenience. The ability to have both header configurations will allow more options and easy access to the processor for Knight Brawlers' prototyping.

The MSP430's five low power modes could be a great benefit to conservation of power consumption for Knight Brawlers. It has active mode of 230 micro amps and standby mode of .5 micro amps. It takes the MSP430 less than a micro second to return to full power. It has the same max temperature operability as the ATmega328 which is 185 degrees Fahrenheit. The MSP430's low price, low power modes, development board and good community support makes it a candidate for Knight Brawlers.

ARM1176JZF-S: The next candidate for Knight Brawlers is the ARM1176JZF-S by ARM. The ARM1176JZF-S is a 700 MHz, 8 to 32 bit (depending on which instruction set you use), 512 MB memory processor. It is a very fast, high memory possible option for Knight Brawlers. The ARM11 is featured on the Raspberry Pi development board by the Raspberry Pi foundation. The micro USB powered Raspberry Pi has 26 GPIO available in a male header configuration, which include UART and voltage sources. This configuration allows this project to have quick access to prototyping and testing. It can output 50 mA max at 3.3 V. The Raspberry Pi has only one available PWM output, which will make it difficult to handle the motors for Knight Brawlers. However, there is a shield available which the Raspberry Pi can use to have more PWM outputs. A great feature of the Raspberry Pi is its onboard Ethernet connection allowing it to connect to the internet. This Ethernet connection will be very handy for our project to add wireless connectivity. It also has Wi-Fi and Bluetooth shields available for wireless connectivity to test the ARM11 processor which is a must for Knight Brawlers.

A neat feature about the Raspberry Pi is that it can program its ARM11 processor in many languages due to its ARMV6 chip. Python, HTML5, JavaScript, C, C++, and many other languages are supported by the ARMV6 chip. The Raspberry Pi uses Arch Linux as its operating system platform for the ARM11. It is a distribution of Linux for Arm computers. Arch Linux aims for simple and complete control for the user. Its support community can be found online with forums, examples, and open source files. The Raspberry Pi also has its own support community offering the same features as the support community for Arch Linux. The support for the Raspberry Pi and built in JTAG connection will simplify Knight Brawlers' debugging issues for the ARM11 processor.

The Raspberry Pi has graphics processing unit that is capable of Blu-ray quality play back and a HDMI output. A camera can be easily connected to the Raspberry Pi through its CSI camera connector. This a great feature to the test video capabilities of the processor for this project. It can interface an external camera and handle wireless video processing easy by its ARM processor and video gpu. This is important for Knight Brawlers' so that it can process video for its camera to the Android device. The max temperature for the ARM11 processor is 185 degrees Fahrenheit which is high enough to handle the temperature operating range for this project. The Raspberry Pi is priced at 35 USD which makes it a cheap, powerful option for Knight Brawlers. Due to its powerful development board, wireless options, video capabilities and ease of programming the ARM1176JZF-S is a strong processor candidate for Knight Brawlers.

<u>STM32F407VGT6</u>: A strong microcontroller candidate for Knight Brawlers is the STM32F407VGT6 by STMicroelectronics. The STM32F4 is a 168 MHz, 32 bit, 192 KB memory ARM Cortex M4F CPU. Being that it is an ARM processor, it gives it a reliability and a performance edge with low power consumption. The microcontroller is featured on STMicroelectronics STM32F4Discovery Board. The STM32F4Discovery has a 100 pin male header configuration. This would allow Knight Brawlers to prototype the STM32F4 by connecting jumper wires from the male header configuration to a breadboard. The STM32F4 has up to 82 GPIO's, which allow plenty of room to play with for Knight Brawlers.

Two outputs can be configured as PWM's for motor control for this project. The Discovery board has USB programmer and debugger for quick access troubleshooting and development. The STM32F4 also has built in Ethernet capabilities and is compatible with Bluetooth modules. This capability will allow for great communication between the Android and RC vehicle. Its max current output pin is .49 mA. Other output pins are slightly less than this value. The STM32F4 has 1 MB of flash memory available for programming in C or assembly. There are several compilers available by ST for programming the Discovery. One of the compiler options for this board is Atollic TrueStudio Compiler. It's an Eclipse based IDE that will allow a friendly user interface. It's limited to 32KB of flash which could be a hazard depending on the coding for this project.

The discovery has plenty of documentation and support available online with great community support. The STM32F4 also has JTAG capabilities for debugging Knight Brawlers. The camera interface available for this processor could be a great benefit for this project's video functions. The Discovery STM32F4 microcontroller has 2 UART's and USB 2.0 OTG for communication interfaces. The STM32F4'sw max operating temperature is at 221 degrees Fahrenheit and has sleep modes for power management. The STM32F4 could provide an excellent microcontroller platform for this project and will be further considered.

<u>AM3359:</u> Finally, the last microcontroller candidate for Knight Brawlers is the Sitara AM3359 ARM Cortex A-8 by Texas Instruments. The AM3559 is 1 GHz, 32 Bit RISC, 512 MB memory processor. The AM3359 processor is featured on Circuitco's BeagleBone Black development board The BeagleBone Black can be

powered via external battery and with a mini USB during coding. The BeagleBone Black has 65 general purpose I/O available in female header configuration. This configuration will easily allow this project to connect a wire and quickly start prototyping. Its female header configuration allows the BeagleBone Black to have the availability of capes (shields, boosters) to be stacked on to the board.

Since the BeagleBone Black is fairly new, the availability of capes is quite limited. Therefore, in order to test different functions of the AM3359 on the BeagleBone Black it will take more prototyping by the user. The female header connections include voltage sources, grounds, GPIO's, UARTs and other connections necessary for prototyping this project. Its pin current varies by pin from 4 to 6 mA. The BeagleBone Black has 8 PWM outputs, which can be used to test the AM3359's ability to control the motor control operations for this project. The BeagleBone black has 65 digital I/O, 7 analog pins. It has 6 UARTs and 4 serial port for communication. It has JTAG for debugging right on the board. This will be of good use to quickly handle debugging issues for this project.

The BeagleBone Black offers plenty of operating system options for onboard development. It is shipped with Angstrom Distribution from Linux that allows you to program in C++, Phython, and other languages. It also includes its own version of Javascript called Bonescript which enables the Beagle Bone to be programmed using node.js. Node.js allows for a device to handle several connections with minimal overhead on a single process. Therefore, makes the computing run faster for our project. Since the Beagle Bone boots Linux, all the programming can be done right on the board. The Beagle Bone is a mini Linux computer. The Beagle Bone Black has 2GB of onboard flash memory to store code. It uses Cloud9 IDE to edit programs live on the board. Cloud9 IDE is open source web based development environment that can be accessed via web browser on any computer. Cloud9 has a community support that provides plenty examples and allows users to collaborate on a project together. The ability to work together on Cloud9 will benefit Knight Brawlers to integrate different aspects of its functionality. Cloud9 supports syntax highlighting for C#, C, C++, Javascript, Bonescript, Python, Xml and many more languages.

A vary neat ability of the Beaglebone Black that would be a great benefit to Knight Brawlers is the ability to run the Android Operating System. Texas Instruments offers an Android Development Kit that is a complete software that can get users to evaluate and optimize the AM3359 on the Android Operating System. The Android development kit is fully optimized to work with the Sitara Arm Cortex A8 processors and aimed to serve fundamental software platforms to build Android products through hardware by Texas Instruments. Versions of Android available to run on the BeagleBone Black are Gingerbread 2.3.4, Ice Cream Sandwich 4.0.3, and Jelly Bean 4.1.2-4.2.2. The ADK includes host tools, debugging options, documentation and support for WLAN, Component Video, Camera, Ethernet, USB, Bluetooth and other features. The ADK also includes plugins for Eclipse and Code Composer Studio to be used as an Android IDE. This allows for the development and debugging of Android applications using the

Sitara AM3359 processor. This project could use this feature to build the app while at the same time integrating the hardware to control the vehicle.

The BeagleBone Black has an on board Ethernet connection that allows for internet connection. There are also tutorials available that can set up the BeagleBone Black for wireless USB internet connection. This Ethernet connection will be of great use to this project to add a wireless interface between the phone and the car. The Beagle Bone Black also has the option for Bluetooth connectivity, but it must be configured by the user. The lack of Capes available for the Beagle Bone development board are illustrated in testing the wireless capabilities of the AM3359 processor for this project. The AM3359 has the GPU capabilities to output high resolution, HDMI video for Knight Brawlers via microHDMI. This microHDMI connection has the ability to add very high quality video to our project if configured through an HDMI dongle.

The max temperature for the AM3359 is 194-221 degrees Fahrenheit depending IO voltage supply. This temperature range is high enough to handle the heat from the motors and environment planned for Knight Brawlers. The Beagle Bone Black is priced at 45 dollars and it comes with tini USB for connectivity. It features a very powerful AM3359 ARM Cortex A-8 by Texas Instruments that is fast, friendly to use, customizable and full of peripherals available to use with Knight Brawlers. With this in mind, the AM3359 is an excellent candidate to fill the requirements for Knight Brawlers' functionality and success.

4.2.2 Microcontroller Decision

After reviewing the options it is helpful to see the information in a tabulated form (see Table 4.2.2). Based on the table it can be noted that there are a variety of options to choose from. There's speed, cost, programming languages, and communication preferences to be determined. The development board costs won't be too much of an issue for this project as long as the actual processor costs are not overwhelming. The main purpose of the development board for the Knight Brawlers project will be to prototype and test project designs. Then the development board designs will be taken and matched to the needs of the circuit board for this project. With this point considered, the cost of the development boards can be crossed out as a non-determining factor of the final processor for this project.

That makes the Arduino Uno, Raspberry Pi, and BeagleBone Black development boards look as more attractive options. When choosing a processor from a development board it is important to check the availability of that processor on the market. Is the processor available for purchase? To sample? Are the processors schematics and Gerber files able to be obtained? When considering these questions and after some investigating it was determined that the ARM1176JZF-S processor will not be able to be obtained quite so easily. ARM requires licensing for this processor which will not be cost productive nor obtainable for this project. Furthermore, the Gerber files and schematics for this processor are not readily available to be viewed by the public. This information leads to the elimination of the ARM1176JZF-S processor.

	Atmega 328	MSP430 G2533	STM32F407V GT6	ARM1176 JZF-S	AM3359
Speed	16 MHz	16 MHz	168 MHz	700 MHz	1 GHz
Memory	2 KB	512 B	192 KB	512 MB	512 MB
Flash memory	32 KB	16 KB	1 MB	Memory Card Expansion	2 GB
GPIO	20	24	82	26	65
Development board cost	$35.00 Arduino Uno	$9.99 Launchpad	$14.25 Discovery Kit	$35 Raspberry Pi	$45 BeagleBone Black
Programming Language	C with Arduino functions Assembly	C, Assembly	C, Assembly	Python, HTML5, JavaScript, Java, C, C++, and many other languages	C#, C, C++, Javascript, Bonescript, Python, Xml and many more languages
PWM Outputs	6	2	2	1	8
Wireless Connectivity	Bluetooth and Wi-Fi available	Bluetooth and Wi-Fi available	Bluetooth and Wi-Fi available	Bluetooth available, Ethernet on board	Onboard Ethernet, Bluetooth, Wi-Fi
Max Operating Temperature	185 F	185 F	221 F	185 F	194-221
UART	1	1	2	1	6

Table 4.2.2 Processor Comparison

A properly working video feed is vital to the success of Knight Brawlers. For this reason, it must be determined how this video will be transmitted. A processor with a high processing rate would be ideal for a situation in which the video feed needs to be done by the processor. If the data is sent via an IP camera then the other project functions can be handled by a slower processor. Taking a look at table 4.2.2, the Atmega328 and the MSP430 are the two choices for low processing speed (as compared to the other options.) They're both pretty similar in their processing characteristics. For this project video transmission will be done without an IP camera so the Atmega328 and MSP430 will be eliminated and the faster options will be considered.

The options available for a fast processing speed are the STM32F407VGT6 and the AM3359. The STM32F4 is at 168 MHz while the AM3559 is at 1 GHz. Again considering the requirements to transmit video from the processor, the STM32F4 has a much lower processing speed as compared to the AM3359. The AM3559 would be a better choice for transmitting video. AM3359's processor being that it is an ARM Cortex-A8 has better video capabilities then M4F processor of the STM32F4. It also has higher memory available 512 MB versus 192 KB for the STM32F4.The STM32F4 has more GPIO but less UART and PWM's available than the AM3559. However, the STM32F4 has more than enough of these features for this project. Both the AM3559 and the STM32F4 are ARM processors, which would add some value of working with a different type of processor for this experiment.

The STM32F4 is a memory ARM Cortex M4F while the AM3559 is an ARM Cortex A-8. There's no question when it comes to which one is more powerful, the AM3559 stands above the STM32F4. However, as of this date the AM3359 processor is around 30 dollars while the STM32F4 can be sampled for free. Also, the AM3359 is out of stock in North America and would require internationally shipping to acquire this processor. The development board for both processors also serve as a deciding factor. The AM3359's BeagleBone Black is a 6 layer PCB while STM32F4Discovery kit is 2 layers. The AM3359 therefore could drive up our PCB cost for our prototype requiring a deeper layer board. The STM32F4 is fully capable of this project and for those reasons will be this project's processor.

4.2.3 Microcontroller Conclusion

The STM32F407VGT6 from STMicroelectronics will be the processor for this project. The pin package connected to the development board will be the LQFP100 package shown in figure 4.2.3. It has 100 pins on the chip but not necessarily all of them will be used. On the development board that we are prototyping on there are no JTAG or Ethernet connections. These connections will have to be implemented on our final PCB design. They will be connected using the Ethernet and JTAG connections available in the block diagram for this processor as can be seen in figure 4.2.4.

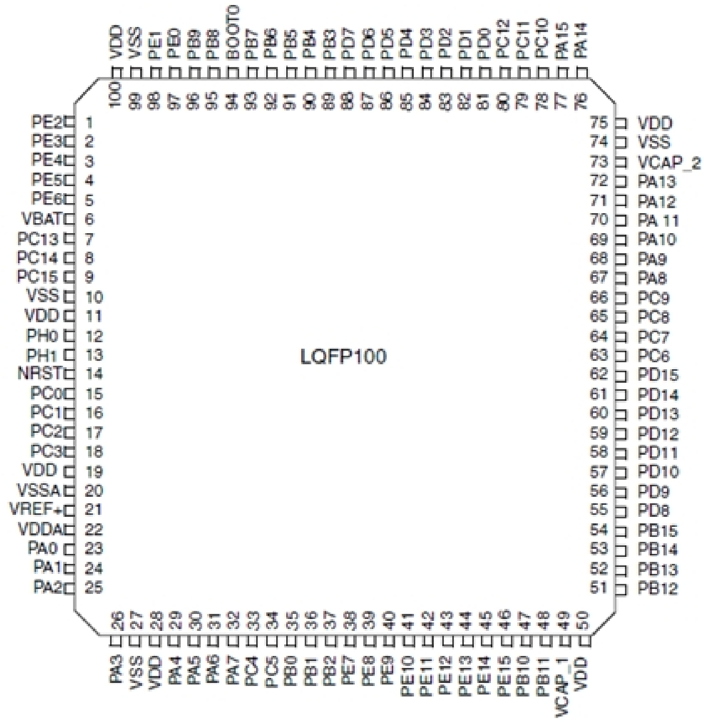


Figure 4.2.3 Pin Package for STM32F407VGT6
(Reprinted with permission from STMicroelectronics.)

With the plethora of pins available in this processors pin package, these new features or any other feature can be easily implemented without any detriment to any other functions for this project. The appropriate connections for each pin are labeled in the data sheet for this processor. This will let us know which pins from the package are necessary to our features and which ones are not.

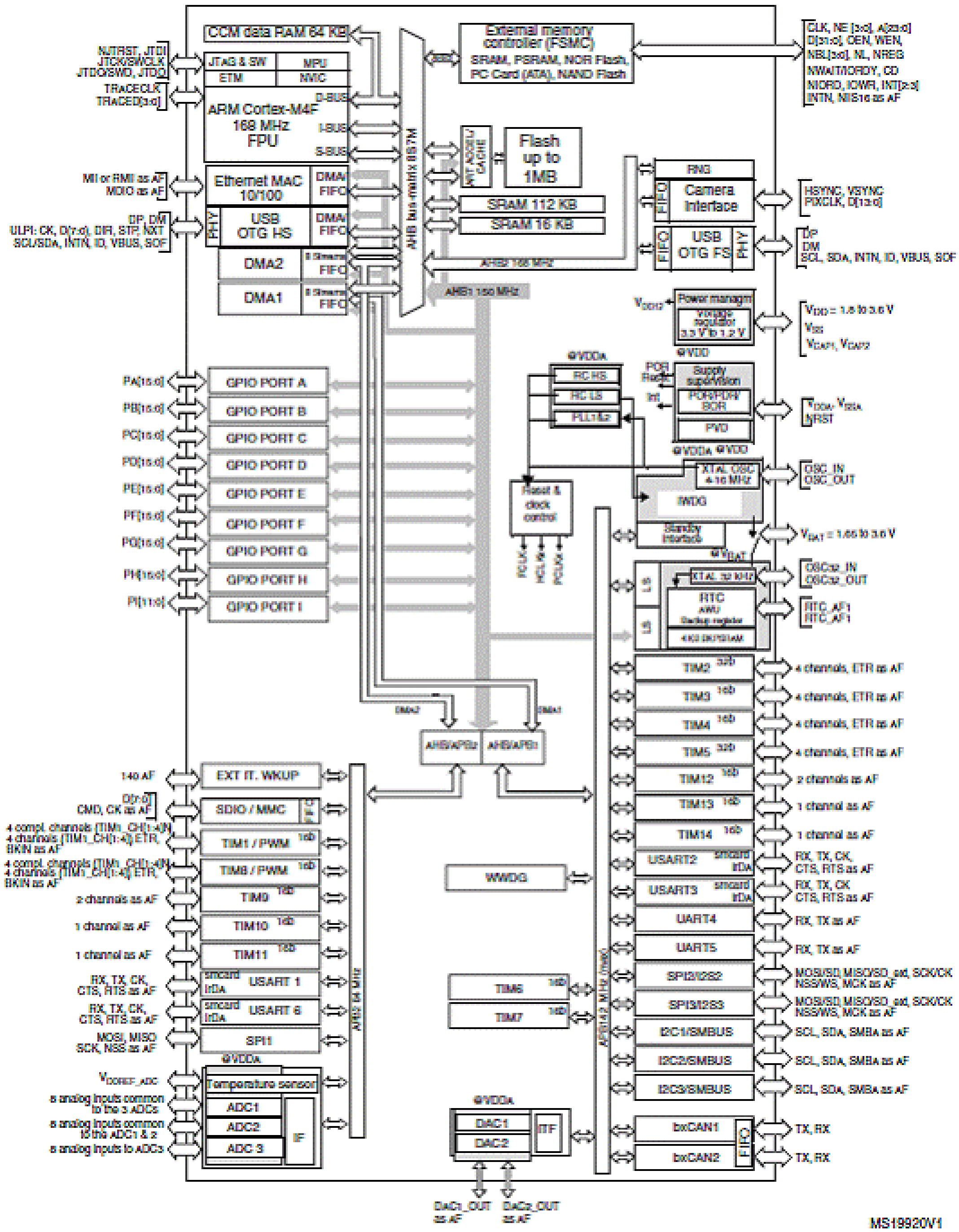


Figure 4.2.4 Block Diagram of STM32F407VGT6
(Reprinted with permission from STMicroelectronics.)

The possible pins used for this project are shown in table 4.3.3. This table will help provide a guide as to where we can make connections based on what function is needed.

Function	Pin number on LQFP100 package
PWM (TIM1/TIM8)	PA6-12, PB0-PB1, PB12-15, PE7-15 PA0, PA5-7, PB0-1, PB14-15, PC6-9
UART	PA0-1, PC10-12, PD2
GPIO (Bumper sensor, LED network)	PA[15:0] PB[15:0] PC[15:0], PD[15:0], PE[15:0], PH0-1
Ethernet	PA0-3, PA7, PB0-PB1, PB5, PB8, PB10-13, PC1-5, PE2
Camera Interface	PA4, PA6, PA9-10, PB5-PB9, PC6-PC12, PD2, PE0-PE1, PE4-6,
JTAG	PA13-15, PB3-4,

Table 4.2.3 Pin Functions Related to Project

The pin connections in table 4.3.3 show the pins for each function. Each function may need more than one pin. For example, JTAG needs two pins TMS (PA13) and TCK (PA14) for debugging. The other pins can be configured for GPIO. The camera interface can handle up to 54 MB/s. It needs a total of 3 pins for function (DCMI_HSYNC, DCMI_VSYNC, DCMI_PIXCK) and 14 for data (DCMI_D0-13). Figure 4.2.5 shows the timing diagram for the camera data pins for this project.

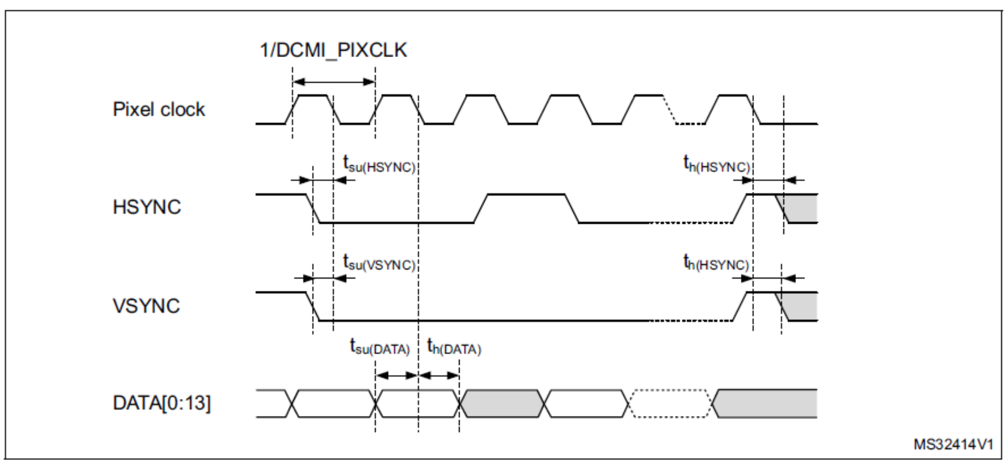


Figure 4.2.5 Camera Interface Timing Diagram
(Reprinted with permission from STMicroelectronics.)

Almost every single pin on this package can be configured as GPIO but it can also serve another purpose. This processor will have plenty of GPIO available to service our LED network and bumper sensors. See table 4.2.3 for GPIO pin locations on the STM32F4 processor. The UART pins we could use for data transmission and receiving need two per UART. The STM32F4 has two UART's available. One of the pins will be the transmission line called Tx (PA0, PC10, PC12) and the other pin will be the receiving line called Rx (PA1, PC11, PD2). One of the UART's can be configured on one of two pin sets, that's why there are six pins instead of just four. For motor control, we will be using the PWM pins.

The PWM pins will use two advanced control timers TIM1 and TIM8. They are three phase PWM generators that are multiplexed on 6 channels. They also have complementary PWM outputs. The pins for these two, six channel timers can be seen in table 4.2.3. The timers have a max frequency of 168 MHz with modulation capability of 0-100 percent. Therefore, we have really good frequency ranges to control our motors. The Ethernet available for possible Wi-Fi connectivity comes with a Media Independent Interface (MII) at 50 MHz and Reduced Media Independent Interface (RMII) at 25MHz.

Both options are available in case of pin limitations but for this project there will be an abundance of leftover pins. Therefore, we could use MII at with no loss of pins to our projects functions. Eight pins are needed for RMII while sixteen pins are needed for MII. However, using MII might make our project designs a little more complicated than using RMII because of the extra signals needed to operate MII. The pins available for both configurations are available in table 4.2.3. Once the pins are assigned and tested for each of the features of this project they will be drawn up in a schematic and then translated to a PCB design.

4.3.0 Remote Controlled Car Power Supply

For the Knight Brawler power supply we have decided to use rechargeable batteries because for a RC car it will be perfect to use. It will easily managed to fit our RC car because there is any replace of any batteries just recharging the once we have. There is a couple of reason we decided to go rechargeable. The major reasons are that:

Save money – Rechargeable batteries will save us money in the long run because even though they are more expensive now, we will not have to keep replacing then. Which will allow us to have to buy multiple packs of double A, Triple A batteries, and even the 9v batteries. Rechargeable batteries can usually last for years after you buy then where as non-rechargeable batteries will only last a few weeks of continues use.

Better for the environment – This is a big deal for our group because we all believe in a sense of responsible to keeping the earth and it environment clean and free of extra stuff. With rechargeable is doesn't have all the extra toxic materials and heavy metals that will end up being thrown away every time a non rechargeable battery is thrown away. So, rechargeable batteries will greatly reduce the number of batteries that will be requires for the Knight Brawler and disposed of.

Prevents waste – Rechargeable batteries will be allow to be able to get used more than one time which extremely lowers the waste of disposable batteries. Far less batteries needed to be used than when using single use disposable batteries. This is another important feature in the Knight Brawler because we need so there is not much waste.

There are many different types of rechargeable batteries that were considered for the Knight Brawlers. Our group decided to use rechargeable batteries rather than using disposable batteries because the Knight Brawler will be used for an extended period of time which will be very useful. Disposable batteries may have a longer shelf life but they don't last as long as rechargeable batteries. Disposable batteries are like bottle energy so it can sit on a shelf with dissolving. The electrical components integrated on our RC vehicle demands for more power another reason why we decided the option of using rechargeable batteries. For the Knight Brawler vehicle, a battery with high current measured in milliamps per hour will be chosen. Because, of the different accessories that will be need to be powered to the component. Furthermore, disposable batteries would to be expensive due to the considerable amount we would have to use in order to keep the Knight Brawler running for a specified amount of time. Rechargeable batteries produce current through an electrochemical reaction that involves the anode, cathode, and the electrolyte where the reaction is reversible.

There are many different types of rechargeable batteries that will be suitable for the Knight Brawler. Both, with all the different types of rechargeable batteries with the capabilities have we will need for the Knight Brawler, so we decide to pick between a couple different rechargeable batteries. The four types of rechargeable batteries that we found most compatible and common for the Knight Brawler vehicle are:

Nickel Cadmium (NiCd)

Nickel Metal Hydride (NiMH)

Lithium Ion (Li-Ion)

Lithium Ion Polymer (LiPo)

The rechargeable batteries for the Knight Brawler has to includes size of the battery, durability, the cost, charge capacity and safe use of the battery. With these parameters it will help us find rechargeable batteries that will be perfect for the Knight Brawler and our group. Also a list of the advantages and disadvantages will be listed for each rechargeable battery. We need to find the best rechargeable battery that is best suitable for the Knight Brawler vehicle because the power being distributed to the whole system is very important and needs to be carefully taking into consideration.

4.3.1 Nickel Cadmium (NiCd)

The Nickel Cadmium batteries are very well known rechargeable battery that is composed of nickel oxide hydroxide and metallic cadmium. The process behind the recharging capabilities of the battery starts with oxidation which at the negative electrode equals the oxidation reduction at the positive electrode. This occurrence generates power within the battery that is able to power device and can be reactivated multiple time to prevent waste. The Nickel Cadmium batteries are relatively inexpensive for low power devices like the Knight Brawler. The Nickel Cadmium batteries were one of the first widely used rechargeable batteries. Nickel Cadmium batteries are available in a wide range of sizes. The Nickel Cadmium battery has a voltage during discharge of 1.2 Volts. Nickel Cadmium Batteries are characterized with wide temperature range operating in temperatures from -40 to 60 C, good electrical performance, and good resistance to overcharge.

Some of the benefits of the Nickel Cadmium battery are listed

- Can provide large currents, suitable for high-drain application
- Works well in cold weather
- Has a long shelf life
- Can withstand 400-1000 cycles with minimum capacity loss
- Can be stored discharged

Some of the disadvantages of the Nickel Cadmium batteries are

- They have a high self- discharge rate (20% or higher per month)
- Has low power to weight ratio compared to NiMH or Li-Ion
- Very toxic inside and has to be recycled
- Can develop internal shorts over time
- Has the memory effect

The concept of the “memory effect” is the idea that the battery loses its charge faster. If the Nickel Cadmium battery was not fully discharged it would lose its maximum capacity. What causes this defect is the formation of Cadmium crystals inside the battery. The metal Cadmium is a highly toxic element.

4.3.2 Nickel Metal Hydride (NiMH)

The nickel metal Hydride batteries are similar rechargeable battery to the Nickel Cadmium batteries. Even electrically, Nickel Metal Hydride batteries are similar to Nickel Cadmium batteries, but a major difference between the two rechargeable batteries are that the nickel negative electrodes use a hydrogen absorbing alloy. The electrolyte is alkaline potassium hydroxide which is able to transfer ion between the cathode and the anode. When a nickel metal Hydride batteries are charged electrons leave the positive electrode. This causes the nickel in the electrode to oxidize. During this process hydrogen atoms leave the positive electrode and react with the electrolyte. During the charge at the same time, electrons go into the negative electrode which causes a reduction reaction. This causes the electrode to absorb hydrogen from the electrolyte. This allows for the Nickel Metal Hydride battery to be used as many times as it is needed.

The Nickel Metal Hydride battery has a high electrolyte conductivity rate which allows for high power applications. The system of the battery system can be contained which reduces the maintenance and leakage issues. The nickel metal Hydride batteries are great for high drain devices. High drain devices are those that need energy quickly like cameras. They are also used in high voltage automotive applications. They also have a much larger capacity than Nickel Cadmium batteries.

Some of the advantages of the Nickel metal hydride batteries are:

- They have a high energy compared to NiCd, up to 3600 maH per cell compared to 2400 in NiCd
- They are cheaper than Li-Ion
- They have a high shelf life
- Can withstand 500 cycles with minimum capacity loss

Some of the disadvantages are:

- They have a high self-discharge rate(30% or higher per month)
- They cannot provide as much current as NiCd (Nickel Cadmium)
- They cannot be charged fast without shortening cell life
- Must be stored charged

4.3.3 Lithium Ion Battery (Li-Ion)

The Lithium Ion battery is a rechargeable battery where the electrodes are made up of lithium and carbon. The cathode is made up of lithium cobalt oxide and the anode is made up of carbon. When the battery charges up, the lithium based positive electrode releases some of its lithium ions which travels through the negative electrode and remains there. Energy built up and stored during this process. When no more ions flow the battery is fully charged. When the battery discharges, lithium ions move back to the positive electrode giving the power the battery needs. When all the ions have move back the battery is fully discharged and needs to be charged up again. The lithium ion battery is much lighter than other types because of its size. Lithium is a highly reactive element meaning that a lot of energy can be stored. Lithium batteries are very popular and are found in your everyday electronics such as Laptops, I-pods, cell phones and etc.

Some of the many advantages of the Lithium Ion battery include:

- Charge. A lithium-ion battery pack can lose only about five percent of its charge per month, compared to a 20 percent loss per month for NiMH batteries.
- Lithium-ion batteries do not a memory effect, you do not have to completely discharge them before recharging, as with some other battery chemistries
- Lithium-ion batteries can handle hundreds of charge/discharge cycles.

The Lithium Ion Battery is not without its disadvantages:

- They start degrading as soon as they leave the factory. They will only last two or three years from the date of manufacture whether you use them or not.
- They are extremely sensitive to high temperatures. Heat causes lithium ion battery packs to degrade much faster than they normally would.
- If you completely discharge a lithium-ion battery, it is ruined.
- A lithium-ion battery pack must have an on-board computer to manage the battery. This makes them even more expensive than they already are.

4.3.4 Lithium Ion Polymer (LiPo)

Lithium polymers batteries are another form of rechargeable batteries (LiPo).They are composed of several identical cells in parallel addition which increases discharge current. A difference between Lithium polymer and lithium ion is that lithium polymer battery does not use a liquid electrolyte and instead uses a dry electrolyte polymer that resembles a thin plastic film. The film is smashed in between the anode and the cathode. Lithium polymer is considered to be an upgraded version of the lithium ion battery.

Some of the advantages of the Lithium Ion Polymer battery are

- Slim and able to be assembled into a credit card
- They can economically made a suitable size
- Light weight: battery used polymer electrolyte need not metal case as protection package
- Improved Safety; more stable overcharge and low rate of electrolyte Leakage
- More environmentally friendly

Some of the disadvantages include

- As opposed to lithium-ion battery, energy density and cycle times decrease
- There is a high manufacture cost for the Lithium Polymer battery
- Prices are more expensive than that of the lithium-ion battery
- No standard shape, mostly made for high capacity consumer market
- Not universally available
- Non-convertible/modifiable
- Chargers are not available in electronics stores

4.3.5 Batteries Conclusion

The battery we will use in the Knight Brawler is an important part of our project because it how the device is powered is really important in how it will work. See table 4.3.5 for the battery comparison. So with all the batteries that are above are all great options for the Knight Brawler. But as going through research the Nickel Cadmium batteries seem to be very reliable the Nickel Metal Hydride has a distinct advantage over the battery which makes the Nickel Cadmium batteries out of consideration for the Knight Brawler. Now, though the lithium Ion polymer battery is smoother and thinner, lithium Ion batteries have a higher energy density and are less expensive to manufacture. Lithium ion batteries have an overheating issue and require an active protection circuit which prevents overheating of the battery and bursting into flame whereas Lithium polymer batteries do not have to that protection circuit which makes its size smaller and easy to manufacture. Lithium ion batteries are susceptible to degrade at the point they are manufactured and become dead in two to three years if they are not used.

	Nickel Cadmium	Nickel Metal Hydride	Lithium Ion	Lithium Polymer
Type	Rechargeable	Rechargeable	Rechargeable	Rechargeable
Disposal	Can be recycled	Can be recycled	Can be recycled	Can be recycled
Storage Temperature	Room Temperature	Room Temperature	-4 to 140 degrees F	Depends on electrolyte
Storage Life	Loses less than 50 % per month	Loses less than 20 % per month	Loses less than 5 % per month	Loses less than .1 % per month
Chemical Reaction	Depends on electrolyte	Depends on electrolyte	Depends on electrolyte	Depends on electrolyte
Capacity	60% of Nickel Metal Hydride	Surpasses the Nickel Cadmium	Two times the capacity	Surpasses the lithium ion
Discharge Rate	flat	flat	flat	flat
Recharge Life	400-1000 cycles	500 cycles	300-400 cycles	300-400 cycles
Temperature	-40 to 60 C	-40 to 60 C	4 to 140 degrees F	Better performance at high and low temperatures
Recommended For	Low Powered Devices	Cameras	Cell phones	Cell phones
Initial Voltage	1.2 and 6.0 Volts	1.2 and 6.0 Volts	3.6 and 7.2 Volts	3.6 and 7.2 Volts

Table 4.3.5 Battery Comparison

Lithium ion batteries have leverage in size, weight, energy density, ability to work in wider temperature range and that they can be recharged before getting totally discharged which any effect on memory. Lithium ion batteries have a lot more energy capacity than that of Lithium polymer batteries so Lithium ion batteries are used in devices which have higher current requirement. Lithium polymer batteries are favorable because of the size which crucial considering the size of our autonomous vehicle, but with the cost is considered, the Nickel Metal Hydride battery is the choice as they are cheaper to manufacture than Lithium polymer battery.

4.4 RC Car Motors and Motor Controls

4.4.0 RC Car Motors Introduction

For the Knight Brawler, we want it to be the most power RC car that we can fit in the small frame we expect to put in it. It is key that we get an electric motor that is high power and low on energy burning levels. With that said, what type of motors should we use? An electric motor is a machine that converts electricity into a mechanical motion and there are so many different type it's hard to exactly which ones will better fit the Knight Brawler the best. The Knight Brawler isn't a unique project so there is a lot of different types of motor that car fit what we need to do in our project so we decided to pick from a hand full of motors and see which one will work the best for what we need it for. The motor are listed below

- AC motors
- DC motors
- Stepper motors
- Servos motors

We select these motors because they could be a great fit for the Knight Brawler RC car. These motors are small enough to fit in any RC car shell that we are looking to purchase and that they are durable and easy to work with so it wouldn't be a problem to change any of the components or add the wheels or other components with concern. For the most part we need to make sure the motors are made for hobbies and project development uses to make it easier on us to work around the different concerns that may come up in the development of the Knight Brawler which may cause problem to use in the long run.

4.4.1 Servo Motors

Servo motors are used to get more accurate angle for directional purposes. These motors are very in RC cars because its ability to get precise angling in exactly where you may want by angling the controller by getting exactly where it may can go. A servo motor is a rotary actuator that allows for precise control of angular position. It consists of a motor coupled to a sensor for position feedback, through a reduction gearbox is a gear system consisting of one or more outer gears, revolving about a central. A servomotor is a closed loop servomechanism that uses position feedback to control its motion and final position. The input to its control is either analog or digital signal, representing the position commanded for the output shaft. The servo motors can require a relatively complicated controller that is often a dedicated module designed which is very helpful in the Knight Brawler

The motor is paired with some type of encodes to provide position and speed feedback. In the simplest case, only the position is measured. The measured position of the output is compared to the command position, the external input to the controller. If the output position differs from that required, an errors signals is generated which then causes the motor to rotate in either direction, as needed to bring the output shaft to the appropriate position. As the positions approach, the error signal reduces to zero and the motor stops.

With more advance servo motors it can give you some much more option that just run at full speed or even at constant speed. Servomotors measure both the position and also the speed of the output shaft. They may also control the speed of their motor, rather than always running at full speed. This is brought to from the servo motor by some many different and compliance formulas that allow the servomotor to be brought to its commanded position more quickly and more precisely, with less overshooting. Servo motors can could in many different sizes and size from the very simplest and cheapest level with uses use resistive potentiometers as their position encoder with are very similar and closely competition with the stepper motors to the more modern day servomotor motors that allows you to add a lot more capable which can get really expensive and more than we would really need for the Knight Brawlers.

Servo motors that are used for higher purpose can use optical encoders which can either be used in absolute way or an incremental way. Optical encodes are very useful servo motors because they work with a better performance module which is grand and can be implemented well for the Knight Brawler. Absolute encoders can determine their position at power-on which is a good thing because it allows the motor to be allows running without any delay time or starting up pulses. But are more complicated and expensive which is some that we don't need to the Knight Brawler, it needs to be as simply and cheap to make as probably. Incremental encoders are simpler, cheaper and work at faster speeds. Incremental systems often combine their inherent ability to measure intervals of rotation with a simple zero-position sensor to set their position at start-up. Servo motors can have a special component added to it like a drive module which will have the motor act as a mosfec H-bridge and is a standard and accessible industrial component. This may be a very useful part of using the Knight Brawler because it will allow us to simplify our circuit design and board layout to help use save money on the board.

Servo motors for the Knight Brawler will have to be more of a general nature with ether just high or low speeds with a constant acceleration. Servo motors can do that but it is a very expensive way to do it when the cheapest servo motors are in the range of 25 to about 40 dollars apiece. The servo motor may be a good motor to get for our steering capabilities, because it may allow to angle our turn more affectively so the camera that we will be use to give live feed back to the phone that will be using. Also, for the purposes for add in the H-bridge servo motors could be a very helpful process in the Knight Brawler but for those modules will be outside of our price range to make four Knight Brawler cars that RC cars.

4.4.2 Direct Current (D.C.) Motors

DC motors are reliable motor that can be adjusted in speed by how much current is supplied through the motor from the circuit. DC motors are an electric motor that runs on direct current electricity from source like batteries and other DC components. DC motor is a mechanically commutated electric motor that is powered from direct current to the motor. The connection in the different fields and winding can change the speed and torque regulation characteristics which are a make the use as a standard model or for only one purpose. The speed of a DC motor can be controlled by how much current is given to the motor so it is an easily access state.

DC motors can operate directly from rechargeable batteries, providing the motive power which lower the cost and gain some interest for dc motor because it allows us to make to the Knight Brawler without have to worry about replacing many parts after it is put together. There are two types of DC motor brushed DC motor and brushless DC motors. Brushed DC motors are internally commutated electric motor designed to be run from a direct current power source and Brushless DC motors are a synchronous electric motor which are powered by direct current electricity and has an electronically controlled commutation system, instead of a mechanical commutation system based on brushes. Both of these type of motors can be used in the Knight Brawler because of its small size and easily accessible from a store or online.

The brushed DC motor generates torque directly from DC power supplied to the motor by using internal commutation, stationary which is the portion of the motor that is not in motion either the magnets and rotating the portion of the motor that is in motion electrical magnets. Advantages of a brushed DC motor include low initial cost, high reliability, and simple control of motor speed and the disadvantages are high maintenance and low life-span for high intensity uses. Maintenance involves regularly replacing the brushes and springs which carry the electric current, as well as cleaning or replacing the commuter. These components are necessary for transferring electrical power from outside the motor to the spinning wire windings of the rotor inside the motor. This motor will be a very good motor for what we are looking for in a RC car for the Knight Brawler and with the price for these motor is will fit in our price range very easily. The brushed motors downed sides aren't much of a problem for the Knight Brawler because it will not be operated in full power.

Brushless DC motors use a rotating permanent magnet in the rotor, and stationary electrical current/coil magnets on the motor housing for the rotor, but the symmetrical opposite is also possible. A motor controller converts DC to AC. This design is simpler than that of brushed motors because it eliminates the complication of transferring power from outside the motor to the spinning rotor. Advantages of brushless motor are its long life span, little or no maintenance, and high efficiency and disadvantages include high initial cost, and more complicated motor speed controllers. For the Knight Brawler, this may not be the best option because of the cost of the motors which is a big factor to how the project will turn out and the with the a low amount of stress that the Knight Brawler will exert on the motors it will not need to have motors that will last longer. This motor is good if the Knight Brawler was a more based on performance and over all skill which is not what we are trying to get on of this project.

D.C motors are very useful motors for a lot of general purpose ideas in the world today. The Knights Brawler is a relatively small RC car and is not a directly implemented for performance based goals, so a DC motor could be a good fit for the Knight Brawler. Because, DC motors are generally use for high performance application with it low range of operations that it is able to function with. It is a can be very cheap and durable which is also an upside for using it in the Knight Brawler. Out of the two type of DC motors we are thinking of using in the Knight Brawler the brushed DC motor are more particular for this project. The brushed motor are very low in cost and will last for a long time as long as it is not use in high stress situations which is not how we plan to use the Knight Brawler. It can also do all the functions that we or looking for in a motor.

4.4.3 Alternating Current (A.C.) Motors

AC motor is a motor that is driven by alternating current. It is a very affect motor if it is attached to high current source because it can last for a long period of time with a high operation level and high current output. It commonly consists of two basic parts, an outside stationary stator having coils supplied with alternating current to produce a rotating magnetic field, and an inside rotor attached to the output shaft that is given a torque by the rotating field. There are many different types of AC motors that have many different functions that will be very useful advantages in the Knight Brawlers design, but some of it is very costly and may be too large to fit in the Knight Brawler frame.

With that knowledge, we have decided to narrow our search for AC motors on simple synchronous AC motor and simple induction AC motors. A synchronous AC motor is an alternating current motor distinguished by a rotor spinning with coils passing magnets at the same rate as the alternating current and resulting magnetic field which drives it. Also, the induction AC motor is a type of asynchronous alternating current motor where power is supplied to the rotating device by means creating of an electromagnetic induction. These types of motors are a more complicated than the other motors we are longing in to because there is a more factor that are in play which allow these type of motor to carry out some of the objective that the DC and servo motors can’t really involve operate at all or for long periods of times which are high current situation.

Induction motor which are also called asynchronous motor that this relies on a small difference in speed between the rotating magnetic field and the rotor to induce rotor current. These are very use because of this trait to allow the motor to act as a lever to change speeds and add a level of flexible to the motor Which the size of this motor it will not be very sizable to the Knight Brawler because we want the Knight Brawler to be as light as probably and that is hard to do with the motor is as heavier than the RC car frame is. This AC motor is also having a high magnetic field that may disturb some of our other component in the Knight Brawler layout design and some of the other device that may be in the area of the RC car game while it is getting played.

In a synchronous AC motor which is a big difference from a induction ac motor because it doesn't rely on induction as the induction motors do it actually can rotate exactly at the supply frequency which allow the motor to be able to motor to be able to act as a standard or permanent structure in a motor. These types of AC motors are the most widely used because of its application to plug direct into device that are powered by a AC current like most device that are being it use today. Because AC current is the type of current to is supplied to device for home and most company's environment. The magnetic field on the rotor is either generated by current delivered through slip rings or by a permanent magnet.

With that said, they is a great application for these synchronous AC motor as small timer and devices that need to be precisely measure and used in operating at a precise speed. This is good for the Knight Brawler because it will allow the RC car to have a constant level and speed for a lot period of time. A problem with these small scale synchronous AC motor is that it doesn't have a high output speed which is needed for the Knight Brawler. These small synchronous AC motor as more use for timing circuit and those type of designs. With these motor they are available in self exciting sizes that will work excellent with the RC car we are use and the cost for these type of vehicle aren't in the low range of 10 to 20 with a probability to be able to get it in an even low price from other sources.

4.4.4 Stepper Motors

Stepper motors are very interesting motors to say at least. It uses a capabilities to rotate it output shaft into equally spaced fraction of a full rotation to let the motor be allow to step through a cycle in phases. This allows the motor to have cycle over what exactly is it powering and or moving through whatever object it is operating for. These motor don't usually have a high output on these motors because of the phases that it will go through doesn't allow the motor to act at a high output before blowing out or anything else. The stepper motor is a mixed of a DC motor and a servo motor to boot. The motor operates with brushless DC motors which are divided into equal step sizes or rotation numbers. The motor's position can then be commanded to move and hold at one of these steps without any feedback sensor, as long as the motor is carefully sized to the application.

This is a help feature with stepper motors because a long as it is made for the specialize case it will be perfect. With could be both a good thing or a bad thing in the Knight Brawler, because of it will either have to be made special for our RC car or if we find out that works similarly to our car it may not work properly enough to get the Knight Brawler in top shape without causing problems and a lot of testing would be needed. For the Knight Brawler we would be able to use switched reluctance stepper motors which are very large stepping motors that reduced pole count, and generally are closed-loop commutated. But, the size of those motor will be fighting against what we are trying to do for this project as a whole. Once again we don't want to have to deal with the motor being heavier than the car itself and the cost for these motors are not in the price range that we are looking to spend for the Knight Brawler scale.

In a stepper motor, they are also much different type of ways that work for our function of the Knight Brawlers. But, the four main ways are listed below:

1. Permanent magnet stepper motor
2. Hybrid synchronous stepper motor
3. Variable reluctance stepper motor
4. Lavet type stepper motor

All these types of stepper motor will be allow to fit in to the Knight Brawler design so we will discuss the possible of all these motors fitting in the design of the Knight Brawler RC car frame. Permanent magnet motors use a permanent magnet in the rotor and operate on the attraction or repulsion between the rotor permanent magnet and the stator electromagnets which is an interesting method because of it using two kinds of magnetic to create a larger magnetic field. Variable reluctance motors have a plain iron rotor and operate based on the principle that minimum reluctance occurs with minimum gap, hence the rotor points are attracted toward the stator magnet poles. This method could be a possible for the Knight Brawler because of its cost and power output. Hybrid stepper motors use a combination of permanent magnet and Variable reluctance techniques to achieve maximum power in a small package size. This is again possible for us because of its small frame and possible power output levels for the motors.

4.4.5 Motors Conclusion

In conclusion, there are a lot of motors that we could use in the Knight Brawler RC vehicle. Between the choices that I have presented in the above section it is unclear exactly which motor should be use for the Knight Brawler RC vehicle. The choices have been limited down to a more suitable selection. The AC motor will not be used for the RC vehicle because of it size and cost it will not be suitable. Also, with using batteries pack system in our RC vehicles we would need an AC/DC convertor to change to DC power from the batteries to AC power that will be used to power the motors.

So, that would be a very useless way to go about doing this because of the waste of resource to do this when there are cheaper and simpler ways to get the same result. So the underlining factor to us was the cost of the motors and the best performance that we could get out of the motors. So, we decide to go with a DC motor because cost of the DC motors are cheaper than the servo motor and stepper motors and It will be more suitable for what we are using it for inside a RC Car. The next step we have to decide whether we want a brushless DC motor or a brushed DC motor. Between these two motors there is many of a factor for our application of the Knight Brawler, brushed motor we will use just because there are little easy to access.

4.4.6 Motor Control

The H-Bridge is the circuit that is directly connected to the motor(s) of the car. When the RC car has two motors controlled by two different H bridges, the H-bridge connected to the rear motor controls the Forward and reverse movement of the car. While the front motor connected to the second H-Bridge controls the car's ability to steer to the left or right direction. An H-Bridge is usually made up of a circuit containing 4 transistors following the general schematic layout below.

This will be very useful for us because it will control how our motors will work and how it can operate in reserve, forward, and turning left or right. The H – bridge will be program to follow sequences of command that will come in from the instruction set of the controller and from the users input from the remote control devices as shown in Table 4.4.6a. This will tell the H – bridge to close and open different circuits to run the motors in the appropriate way that it is design to go. (The H-Bridge follows the sequences, where 1 = closed and 0 = open)

S1	S2	S3	S4	Result
1	0	0	1	Motor Moves Right
0	1	1	0	Motor Moves Left
0	0	0	0	Motor Free Runs
0	1	0	1	Motor Breaks
1	0	1	0	Motor Breaks
1	1	0	0	Shoot-Through
0	0	1	1	Shoot-Through
1	1	1	1	Shoot-Through

Table 4.4.6a H-Bridge Control.

Depending on the type of H – bridge is used for we can control 1 motor with One H- Bridge or for our application 4 motors with one H – bridge. This is all depending on how many we want to spend and how it will be implemented in our design of the Knight Brawler. For a simple one input motor H-bridge which is a standard in the design of all H-bridge which is listed below in Table 4.4.6b just that some addition pins or multiple outputs or inputs to for the multiple motors.

Pin Name	Name	Description
GND	Device ground	Ground device
VCC	Device Supply	Control power to H - bridge
VM	Motor Supply	Control power to motor
IN1	Input 1	Input power
IN2	Input 2	Input power
SLEEP	Sleep Mode Input	Save Power
OUT1	Output 1	Connect to motor winding to control direction
OUT2	Output 2	Connect to motor winding to control direction

Table 4.4.6b H-Bridge Design Functions

4.4.7 H-Bridge Selection

<u>Texas Instruments SN754410:</u> Texas Instruments offers an integrated circuit chip called the SN754410 which implements the H-bridge circuit just as well as if the H-bridge circuit had been implemented using only discrete components, giving us the option of saving space on the car and turning out to be overall more cost efficient seeing as the TI SN754410 is available for $1.85 from the TI website. The SN754410 has the ability to run two motors through it allowing it to be the ideal choice, so the H-bridge chip will be able to command the car to move forward, in reverse, left and right.

The SN754410 needs at least 4.5 Volts in order to work, so set VCC1 to a regulated +5V in order for the chip to work, if needed a capacitor will be added at the VCC1 input to clear up the feedback and keep a constant voltage. Once the chip is working the VCC2 will need to be activated by directly connecting a battery to it with a max voltage of 36V (which is way more than is needed for this specific project). The factor that will enable each pin is whether the pin is set to High (at least 5V) or Low (0V). In order to enable Motor 1 or Motor 2 their pins must be set on high, and setting the forward, reverse, left, or right Pin to high will enable that command forcing the corresponding motor to act as instructed.

<u>Texas Instruments DRV8833:</u> The DRV8833 features a 4A peak output when configured in the parallel mode. The current limiting ability can be controlled with pulse-width modulation from the microcontroller. The motor power supply voltage range is 2.7 – 10.8 V which is ideal for RC-type motor conditions.

Texas Instruments offers an integrated circuit chip called the DRV8833 which implements the H-bridge circuit just as well as if the H-bridge circuit had been implemented using only discrete components, giving us the option of saving space on the car and turning out to be overall more cost efficient seeing as the TI DRV8833 is available for $0.67 from the TI website, but can be get free samples on the TI website as well. The DRV8837 has the ability to run one motors through it allowing it to be the ideal choice, so the H-bridge chip will be able to command a vehicle to move forward, in reverse, left and right.

The DRV8833 needs at least 2.7 Volts in order to work, so set VCC1 to a regulated +10.8V in order for the chip to work, if needed a capacitor will be added at the VCC1 input to clear up the feedback and keep a constant voltage. The factor that will enable each pin is whether the pin is set to High or Low. The key features which make this H-bridge desirable to many others include are that the pulse-width modulation winding current regulating/limiting, thermally enhanced surface mount package, abundance of application support and literature.

Texas Instruments DRV8837: Texas Instruments offers an integrated circuit chip called the DRV8837 which implements the H-bridge circuit just as well as if the H-bridge circuit had been implemented using only discrete components, giving us the option of saving space on the car and turning out to be overall more cost efficient seeing as the TI DRV8837 is available for $0.53 from the TI website, but can be get free samples on the TI website as well. The DRV8837 has the ability to run one motors through it allowing it to be the ideal choice, so the H-bridge chip will be able to command a vehicle to move forward, in reverse, left and right. The DRV8837 needs at least 1.8 Volts in order to work, so set VCC1 to a regulated +11V in order for the chip to work, if needed a capacitor will be added at the VCC1 input to clear up the feedback and keep a constant voltage. The factor that will enable each pin is whether the pin is set to High or Low. This will be the H-Bridge selected for Knight Brawlers.

4.5 Impact Sensors

4.5.0 Introduction and Purpose

In order to know when a hit has been taken from another competitor, the remote control cars must have an onboard sensor that will serve that function. The sensor must also be able to relay that information to the rest of the gaming system to indicate the score change. That includes changes reported to LED status, the user app, and the processor. The impact sensor should also react to a specific level of impact so that soft bumps or accidental contact or contact in an area that is not considered an impact zone will not trigger an incorrect change in score.

When selecting an impact sensor, more than just one factor is to be taken in to consideration. In the real world, products are deemed successful or worth producing based on key components that make them feasible. This requires a comprehensive discussion that covers all aspects that may affect the project as a whole

Functionality - The primary area of concern would be how the impact sensor functions. How does it function in accordance with what we want to achieve? If the sensor is able to detect the impact but unable to relay any type of information to the rest of the subsystems, then it will be rendered useless to our goals and objectives. We want something that will give data that will be able to be accepted by the processor so that it may communicate with the other aspects of the project.

Size - The cars we are building will have a limited amount of space to work with. Considering that several components going into the project will take a considerable amount of space on the remote control cars, the size of the impact sensors is a big factor going into which one is chosen. It is perhaps the second most important consideration taken into account when choosing an impact sensor after the functionality. The sensors will have to be mounted on the sides of the remote control car and the rear. That means we are confined by the dimension of the cars. The sensor will have to go between the wheels of the remote control cars when mounting them on the sides and preferably centered on the rear. This also limits the size of the impact sensors. An impact sensor that is too big may erroneously detect impacts to areas of the cars that are not considered valid scoring zones. It may also add weight to the remote control cars that will already be bogged down with other subsystems. Impact sensors that are too small may also pose a problem. They may not be able to consistently detect a hit if they can't cover the area of impact that is desired. They also may be at risk of easily being dislodged.

Durability - As these cars will be taking hits, the impact sensors will be the center of that attention. Due to this they will incur plenty of damage over the course of several games. Ergo, the impact sensors will have to have an amount of durability so as to not hinder the ability to play several games without any sort of problems. This does not mean that the sensors will have to be iron clad components that will be indestructible. What is needed is a reasonable amount of sustainability for a period of time that will not cause an inconvenience.

Part Reputation - When finally narrowing down a type of impact sensor, the reputation of the brand of vendor will be a key aspect to consider. The type of sensor may be a good part to select for the project specifications but if the part is poorly made then there may be a high chance of incorrect response or low durability. To avoid this we must choose reputable parts that have been tested and are not made by obscure or suspect companies or developers.

Cost - Cost will be the final and a very important factor to consider when choosing the impact sensors. Considering that the remote control cars will have sensors on the sides and rear mean that each car will have to at least have three sensors each. Then considering the fact that it is desired to have 4 fully functional Knight Brawlers, that number jumps to 12 sensors. It also has to be taken into consideration that backups should be purchased in order to cover any type of problems that may arise such as faulty parts or testing for prototypes. Due to these reasons the price must be taken strongly in to consideration. An individual sensor that amounts to a charge of thirty dollars will easily escalate to 360 or 600 dollars due to the need of multiples. In order to achieve a low cost solution, buying in bulk may be a reasonable option.

4.5.1 Possible Options

Meticulous research is required in exploring every possible type of sensor that could be used in the project. When exploring the available options, the factors previously discussed are considered. Each option will be scrutinized based on functionality, size, durability, and cost. Part reputation will be considered after a sensor type is chosen.

4.5.1.0 Piezoelectric Disk

The first option to be discussed for an impact sensor is the piezoelectric disk. The piezoelectric disk works on the basis of piezoelectricity. Piezoelectricity is an electrical charge that results from mechanical stress. It measures force, stress pressure and acceleration from the charge that results from the mechanical stress. Disks specifically produce the electrical charge when they are deformed. Piezoelectric disks are used in various applications and therefore cover many needs and have an added advantage of versatility. They can be used in measuring vibrations in an instrument, touchpads on a phone, or combustion engines. Piezoelectric disks are considered mature technology for their wide array of uses and longevity. In order to fully analyze the piezoelectric disk for feasibility in the desired use as an impact sensor for a remote control car, the previous factors discussed must be taken into consideration.

Functionality - In terms of functionality, the piezoelectric disk is very viable option. It is able to detect various levels of physical impact by giving an electric charge. The charge can be measured depending on how hard the stress is incurred on the disk. This means that a threshold can be set on what is an acceptable level of impact that will trigger a change in score. Also very importantly, the electric disk can be hooked up to the microprocessor in order to relay the change in stress which relates to an impact. This means it can successfully communicate with the rest of the subsystems.

Size - Piezoelectric disks can come in various sizes which is a very good quality. The piezoelectric disk size can be changed in accordance with need and size restrictions dictated by the size of the remote control cars. They are also quite thin which means they can easily be mounted between the body of the remote control cars and the internal parts.

Durability - The piezoelectric disks have a very good durability. Although it must be taken into consideration that high sensitivity does decrease in time and is due to high temperatures in some piezoelectric disks depending on the material they are made from. Piezoelectric disks with the least amount of high sensitivity degradation are made with single crystal materials. High sensitivity will not be needed for this project so this problem may not pose any hindrance.

Cost - Piezoelectric disks can vary in price depending on materials they are made from and size. For this project, high sensitivity will not be required which means the piezoelectric disks can be purchased without having to be fettered by a specific material choice. For that reason, they can be acquired cheaply and in great quantity which is excellent considering that several sensors are fitted to the four remote control cars and additional sensors are needed as backups in case of an emergency.

4.5.1.1 Micro switch

There are a variety of micro switches that can serve different purposes. They are another form of sensor to strongly consider. They can be found in several different applications such as microwaves, jet ways, and the space shuttle. Most micro switches work on the basis of an open and closed contact. The switch is triggered in order to cause a closed circuit that is normally open. This is how the switch works to send of signal of some sort of action. In our case this will be the impact of another remote control car. Out of the several varieties of switches, the one that works best and will be discussed for the Knight Brawlers project would be a lever switch. A good lever switch to use would be one that can be pressed without much force and that will not be activated with very little force. Finding one suitable in that manner will require testing different flexibilities of the levers on the switch.

Functionality - When looking at functionality, micro switches work well as an impact sensor. By using a lever micro switch, the best feature would be that it is simple. It is either pressed down to create a signal or it is not. Once the switch is hit, the lever props right back up so that it would not need to be reset manually. The ability to communicate with the processor and the rest of the subsystems also is a big factor in using a lever micro switch.

Size - Micro switches can come in different sizes, including the lever switch. They can be small and mounted onto the remote control cars without taking much space. The lever on the switch itself is also small but can be modified and elongated in order to give a larger contact surface so that the remote control cars won’t have to target an area that may be too small.

Durability - Micro switches can be very durable. They can withstand several uses and depending on the quality and brand, they can also last a very long time and are very viable for the Knight Brawlers project.

Cost - Micro switches vary in price depending on the size and type of switch. Considering the remote control cars will need small lever switches, the price for four cars will not be very high. They can be acquired cheaply and locally which saves on shipping and handling.

4.5.1.2 Other Possibilities

Load cells are sometimes used as impact sensors and work on the basis of converting force to an electrical signal. There are different types of load cells that work with varying levels of effectiveness. Although despite the versatility, Load sensors will be difficult to mount and to utilize the signal.

Tubular sensors are widely used in industry to detect impacts on products during the handling process. They can offer different levels of sensitivity and can be acquired cheaply. Tubular sensors work on the basis of deformation due to shock. Although some of the qualities of tubular sensors are desirable, the fact that they work based on visual inspection, rules them out for our desired use.

Other ideas researched and considered were a bladder system that sensed movement (at the suggestion of the UCF EECS' faculty member Dr. Richie), various other micro switches, micro phones that would hear impact noise, flex sensors, accelerometers, and conductive foam. All these presented challenges or incompatibility.

4.5.2 Sensor Conclusion

By comparing all the options researched (table 4.5.2), two of the sensor types stand out, the piezoelectric disks and the micro switches. The load cell and tubular sensors are good forms of sensors and indicators and are useful in other applications. In the case of Knight Brawlers, they would not be very practical. As mentioned before, the load sensors add an unnecessary level of complexity and the tubular sensors are too simple and not complex enough.

Micro switches and piezoelectric disks both offer viable options in terms of the criteria discussed earlier. They offer functionality, they can be small in size, they are durable, and both are relatively cheap. In order to separate the two, the faults of each must be further scrutinized.

Looking first at the piezoelectric disks, they can produce varying amounts of electric pulses depending on how much force is applied. Although this may be very desirable in other applications, for the knight brawlers project, it may pose an obstacle. Varying levels of response mean that high levels would have to be compensated by adding more components to the circuit in order to assure there isn't any electrical damage done by a surge in electricity. Low levels may also create a signal that is too weak and may go undetected. There is also the problem of accidently triggering the sensors within your own car if you hit someone else. Time would need to be spent calibrating how much is too little or too much.

For the micro switches, size and cost are suitable for the Knight Brawlers project, but are not ideal either. The switches may be harder to mount than the piezoelectric disks but not by much. The cost of each switch may also be higher than the disks but not enough to make a large dent in the budget.

Sensor Type	Functionality	Size	Durability	Cost
Piezoelectric Disk	It can send a signal to the rest of the subsystems. Sensitivity varies.	Can come in small sizes.	High durability depending on material	Cheap depending on material
Load Cell	Can send a signal but may be harder to interpret, sensitivity varies	Can come in different sizes, although not small enough	High durability	Expensive than other options
Self-Contained Tubular Sensor	Does not relay signal	Small	Low durability	Cheap
Micro Switch (lever switch)	Can send an easy to interpret signal	Can be small	High Durability	Cheap

Table 4.5.2 Red indicates negative feature, blue indicates positive feature

Upon further inspection of the piezoelectric disks and the micro switches, it seems that the micro switches (figure 4.5.2a) may be a more suitable option for the Knight Brawlers project. Quite simply the simplicity of the functionality is too good to ignore. It may be slightly harder to mount and a bit more costly, but the signal it produces would require far less troubleshooting than the piezoelectric disks.

It is either pressed or not pressed as opposed to the piezoelectric disks which create varying levels of signal. Although the piezoelectric disks were not chosen for this particular aspect of the project, they may be considered for an added feature if time permits. Working with the disks adds a level of interest and would be suitable for a simple game of tag which would eliminate the need to detect a specific level of contact or needing to solve the problem of mutual hits.

Figure 4.5.2a Lever micro switch, which was chosen as a suitable impact sensor

The particular lever micro switch that was chosen was the single pole double throw (SPDT) Zippy lever switch. Single pole means that there is only a single circuit that the switch controls. This is all that is needed for the purposes of indicating a hit. Double throw means that there are two possible positions that the switch can activate. The first position would be the low state (figure 4.5.2b) in which the contacts within the switch are not active, thus not completing the circuit. The second state is when the switch is pressed and the circuit is completed (figure 4.5.2b) which allows the signal to be transmitted. The switch contains three terminals which are labeled normally closed (NC), normally open (NO), and closed (C) (figure 4.5.2b). More often than not, only three of the terminals will be used. Only in specific cases will all three be used. Using the normally open and common terminals is the most widely used configuration. Using those two would provide a normal operation of the switch which would be no signal when the switch isn't pressed and a signal and completed circuit when the switch is pressed. The normally closed terminal is sometimes used with the common terminal to create an opposite operation of the switch. When the switch isn't pressed, the circuit is completed and the signal is relayed. When the switch is pressed, the circuit is open and there is no signal transmitted to the rest of the subsystems. It is rare that there is ever a connection between the normally open and the normally closed terminals and that will not be necessary in this particular application.

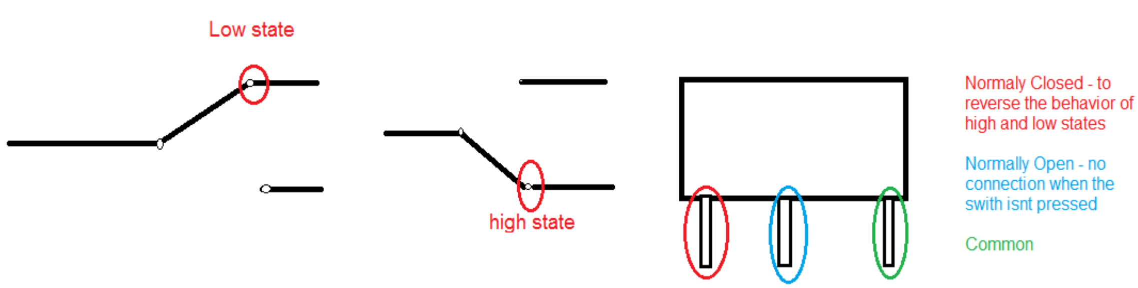


Figure 4.5.2b From left to right: the low state, high state, and terminal layout

The issue of having to use a debounce circuit for the switch must also be taken into consideration. Since bounce is a common problem with mechanical switches, this could cause an issue with proper operation of the switch. A circuit could be designed to low pass filter multiple pulses created by the elasticity and momentum of the metal contacts. Upon further inspection though, the use of a debounce circuit will be unnecessary with the Knight Brawlers micro switch sensor. Usually the concern over debouncing involves applications in which the first time you press the switch something is done and the second time you press it, something different is performed. That does not fit the application of merely sensing three hits.

Connecting the spdt micro switch is one of the final design aspects to consider for the sensor subsystem. There are several ways in connecting the switch and each can have its own advantage. It should be mentioned that since the switch is basically just connecting two wires together, it will not matter the order of connection, so long as it is properly done. One way could be to use the normally open and common terminals which would require a connection on one leg of the switch to ground, a resistor, and one of the I/O pins on the printed circuit board. The other leg would be connected to a voltage power supply. This type a connection would create a low state when the switch is not pressed and a high state when the switch is pressed.

The second possible connection would be the same as the previous but connecting the normally closed terminal instead of the normally open terminal. This would cause a low state when the switch is pressed and a high state when it is not. Using this method, we would need to act on a low state rather than a high state. The last connection possibility would exclude the need for a resistor. One leg of the switch would be connected ground while the other would be connected to one of the digital I/O pins. This type of connection though utilizes internal pull-ups which means the sense of test would have to be inverted. Low would be high and vice versa.

So if this method is combined with using the normally closed terminal which also inverts the two high low states, normal operation of the switch could be used. This means low for when the switch is not pressed and high when it is pressed. This may be the best method to use because if the intention is to try and design the project much like what would be done in the real world, using fewer components would be the best option. The third option (figure 4.5.2c) eliminates the need of a resistor and also does not invert the high and low states. In the case of this project, the cost of one resistor will not be much of an issue but if were to be produced on a mass scale with multiple vehicles per gaming set, then the matter could be a wise thing to inspect.

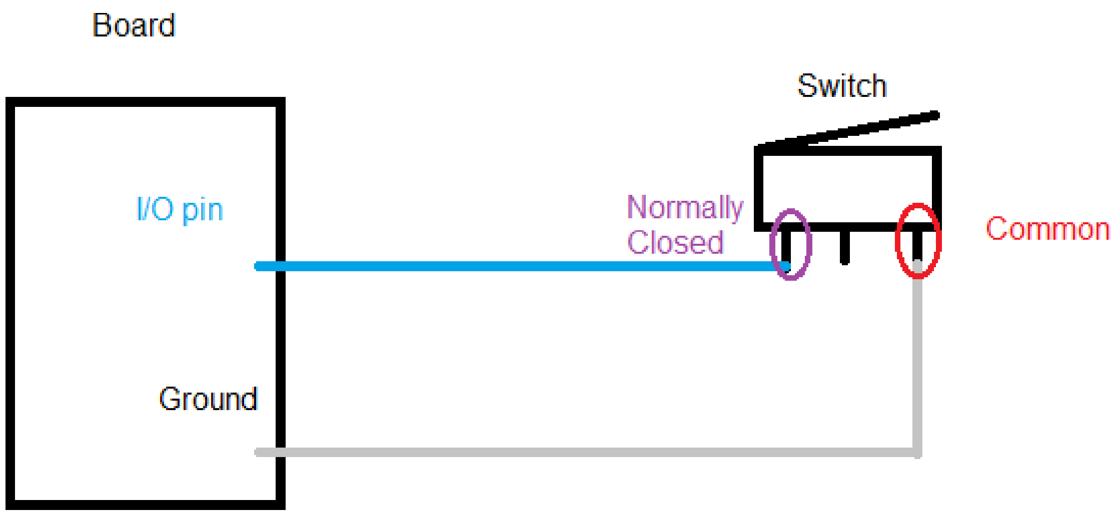


Figure 4.5.2c Final sensor connection

4.6 LED Network / Scoring System

4.6.0 Introduction and Purpose

In most gaming systems such as the Knight Brawlers project, visual aids are utilized in order to convey score, status, or other aspects of the game. For the Knight Brawlers project, light emitting diodes (LEDs) will be used in this manner. In order to keep another form of status other than on the android phone, the LEDs will be fixed on each of the remote control cars. Ideally, each remote control car will have three LEDS mounted on the roof or hood of the car. It will have 3 LEDs because the game will consist of three hits. At the beginning of each game, each player will have all three LEDs lit in green to visually convey full health status (figure 4.6.0).

As the game progresses and the player take his or her first hit, the health status diminishes. The LEDs in turn respond to the hit. After the first hit the LEDs will be displayed as yellow and only two will be lit (figure 4.6.0).

If the player receives a second hit, the health status of the remote control car will diminish even further. The LEDs on the top of the car will not be red and only one will be lit (figure 4.6.0).

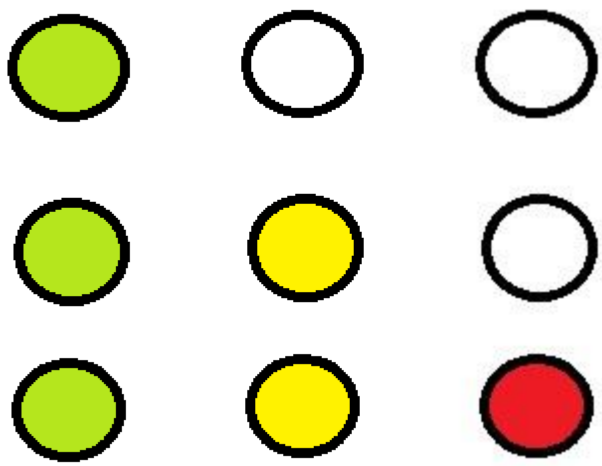


Figure 4.6.0 From left to right: Full health, one hit, two hits

The third and final hit will cause all the LEDs to flash red once and then become unlit for the remainder of the current gaming session. This will indicate to the player that they have been knocked out and unable to participate anymore until the next game.

Additional LEDs may be added to the cars if time permits. They may be added to the bottom of the cars to light them up and emulate a real life modification on a lot of street cars. These LEDs will be purely for aesthetic purposes and will not be necessary to the project.

4.6.1 Choosing an LED type

The type of LED to be used does not need much consideration. Red green blue LEDs are the most suitable type of LED because of their ability to be versatile. Red green blue LEDs are able to display a variety of colors by running independent pulse width modulation to each of the single red, green, and blue chips inside a single LED. This means that for the project, it is possible to get an array of colors form a single LED as opposed to a single color from other LED types. This is ideal because it will require less LEDs, less wiring, and will save space when mounting them on the remote control cars. Each red green blue LED has one ground which means only three wires for ground will be needed on the remote control cars.

Sizes of the LEDs will have to be small, but not too small so that it can give an adequate visualization for the players. Also certain LEDs give a wider angle of view than others. For this project, the most visible will work the best.

4.6.2 LED Driver

LED drivers are needed in most applications where LEDs are utilized. The drivers will assure good performance for the LEDs and prevent failures by controlling the power supplied to the LEDs. That way there will not be an excess (possible failure) or insufficient amount (low light).

When looking at different drivers, there are a couple of things to consider. One of the main things is how many outputs does the driver have. The minimum amount of individual LEDs used on each car will be three, but each red green blue chip will need an output. Therefore, a minimum nine outputs on the driver will be the ideal case for the LED network on each remote control car. The other factor would be complexity. The job we need the LEDs to do in this project is not very complex and therefore a simple driver will suffice.

Based on that information, the LED driver chosen is the TLC5940 (figure 4.6.2a) from Texas Instrument. It is a commonly used driver and has a reputation for easy use compared to some other more complex drivers. The TLC5940 is a constant current sink driver which eliminates the need for any resistors. This is a plus because it reduces the amount of components and cuts down on cost. Although the cost of a couple of resistors is not a big deal, if you think in terms of large scale production, it does make a difference. It also has 16 outputs which will more than suffice for what is needed.

Figure 4.6.2a TLC5940 Driver that will be used

Looking at the data sheet for the TLC5940, we can anticipate how the connection for the red green blue LEDs will go (figure 4.6.2b). Using the TLC5940 means we are going to have to use common anode LED’s.

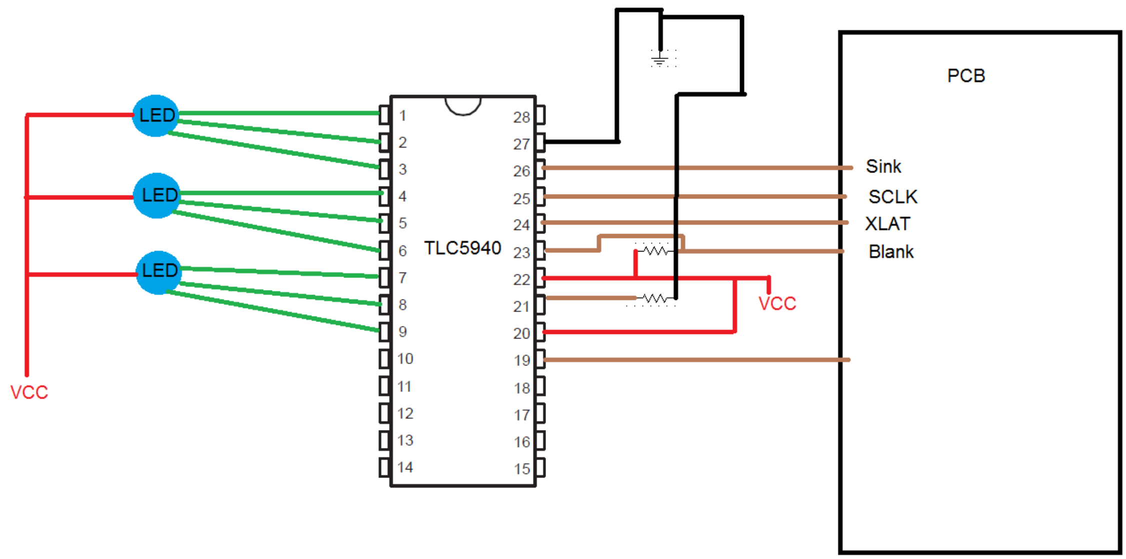


Figure 4.6.2b LED and Driver connection

4.6.3 Programming LED network

There are several pins on the TLC5940 LED driver, such as sink, sclk, xlat, and blank so it's important to fully understand which ones will be needed and how they will be used. A simple and fail safe approach will be enacted if it becomes to difficult or ineffective to fully use all of the driver's capabilities. That would mean connecting the TLC5940 to the output pins on the printed circuit board (or possibly the development board since only one of the cars will probably have a printed circuit board) and simply turning them off and on to use different colors. This however will mean using three drivers instead of one so that they can each connect to a different color chip on the LED's. This will be a simple and easy method to implement. The second possible option is to program all the different capabilities of the TLC5940. The LED network will respond to the sensor input so that means the program written for it will have the two systems working together as illustrated in figure 4.6.3.

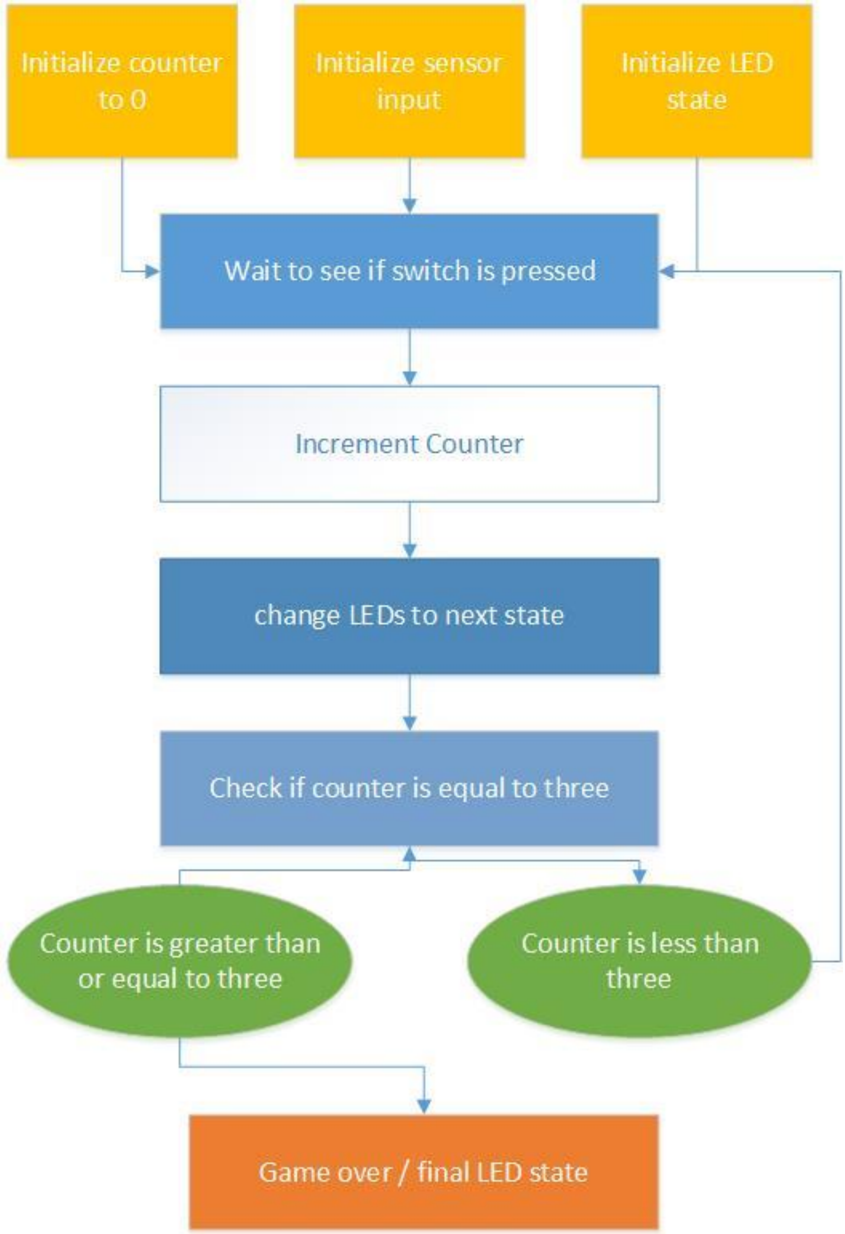


Figure 4.6.3 Programming Flow between Sensors and States

4.7 Digital Video Cameras

4.7.0 Introduction and Purpose

To be covered in this section are: the purposes of the digital video camera on the car, desired outcome from implementing the digital video camera technology on the remote controlled car, the various types and specifications of digital video cameras available for us to choose from, our desired specifications, the likely required specifications due to various limitations, any complications that might arise from including this technology, and any constraints that including a digital video camera on the remote controlled car might put on other design components. How a digital video camera works will not be discussed in this document but the different technologies that are available to the market for consumers and developers will be.

The purpose of the digital video camera for our project will be to transmit a first person view (fpv) of the player's car to his or her Android cell phone. The user will be able to be away from the playing arena with others who are also away from the playing arena, and compete by only being able to see through the fpv camera on the car. This will add a unique game play variation, making the game more challenging and interactive. The team would also like to include the digital video camera to obtain practical experience with embedded hardware design such as: input configuration, data manipulation, encoding and decoding a video stream real-time, and transmission of a video stream wirelessly.

4.7.1 The Internet Protocol Camera

The internet protocol (IP) camera, in its simplest form, is a digital camera that sends and receives data digitally via the wide area network (WAN) we know as the internet, or a local area network (LAN), which might be an individual's home network. The typical application for an IP camera is for security surveillance (both in the home and small scale commercial properties) and baby monitors. The vast majority of IP cameras available are not intended for a hobbyist or development platforms so one can find many IP cameras that are completely encapsulated bundles of technology wrapped up in an attractive shell. It is common for an IP camera to contain all of: digital video camera with infrared lens, infrared LEDs for night vision coupled with the infrared lens, microphone, Ethernet port to transmit data, Wi-Fi module for wireless connectivity, MPEG, MJPEG, or h.264 encoding, and bundled software to interface all the options; likely with both Android and iOS options. Some IP cameras are so sophisticated that they include all of the options previously mentioned in the above paragraph and can even stream a video feed to a specified internet protocol (IP) address without the assistance from any external hardware like a computer or directly to a smartphone.

4.7.1.0 Wireless Transmission Technologies

IP cameras transmit their data exclusively through a WAN or LAN so either the internet or a local router must be involved in viewing or recording the video. The video stream from an IP camera can also be transmitted wirelessly through an onboard Wi-Fi radio transmitter -assuming one is included. Because most of the IP cameras are designed for a consumer with very low technical skills in mind, the manufacturers like to keep the details of their devices very general. However, IP camera manufacturers with Wi-Fi connectivity commonly advertise the 802.11b/g and sometimes "n" specifications.

Model	WVC80N-NP	CWD-TB-11489
Brand	Linksys	CWD
Connection Type	RJ45	
Wireless Standard	IEEE 802.3u, 802.11g, 802.11b, draft 802.11n	IEEE 802.11 b/g
Networking Protocol	TCP/IP SMTP HTTP DHCP FTP	Unknown
MAX Resolution	VGA 640 x 480	VGA 640 x 480
Encoding	MPEG-4, MJPEG, H.264	MJPEG
Frame Rate	Up to 30 fps	Up to 30 fps
Security	WEP, WPA, Wi-Fi Protected Access 2 (WPA2) Security Key Bits: Up to 128-bit encryption	WEP, WPA, WPA2
Audio	built-in microphone	NO Audio
Operating Systems Supported	Apple MacOS X 10.4 or later, Microsoft Windows Vista / XP	Windows 7
Power over Ethernet	Yes	No
Power	5V, 1A	3V, 320mA
Approximate Dimensions	3.5” x 1.5” x 4.7”	30mm x 27mm x 27mm
Approximate Weight	4.6 oz	100g

Table 4.7.1 IP Camera Specs

Above in table 4.7.1 are the specifications for two IP cameras that are under consideration: the Linksys WVC80N-NP and the CWD-TB-11489. These two cameras are good representatives of the feature sets that one can find on any given IP camera, depending on its size and cost. The Linksys WVC80N camera has the ability to operate as a stand-alone unit; in other-words, it can send its video stream via RTSP protocol so that mobile devices like a smart phone can access the feed. Also, the software to configure the device (which does require a PC to setup) is robust enough to allow manual configuration of all: IP Address, subnet mask, gateway, primary DNS, secondary DNS, RTSP port, RTP data port, RTP data packet and more through a graphical user interface.

4.7.1.1 Encryption Technologies

Many IP cameras offer various encryption technologies. For the unit above, WEP, WPA, and WPA2 with 128 bit encryption are available; these options appear to representative for the majority of the IP cameras in the market.

4.7.1.2 Form Factor of the IP Cameras

Form factor for every element of the car's design is very important. Initially, the design has been limited to: width of 5.5", height of 6.0", length 10.0". These dimensions were chosen because the designers plan on building onto an existing platform with a body and frame which cannot be changed.

The options available for IP camera form factors are quite vast. They range from tiny modules about the size of a medicine bottle like the one pictured below, to cameras with housing the size of a woman's shoe box. Below are two pictures demonstrating a size comparison to a person's hand.

The first picture is of a Linksys WVC080NA IP camera which has already been discussed. It does not even fit in the man's hand in this picture so it would look ridiculous strapped to a remote controlled car with a width of 5.5". The next picture is the CWD-TB-11489, a tiny IP camera that features MJPEG encoding via Wi-Fi. This is a much more modest size to incorporate into the Knight Brawler's project.

The CWD-TB-11489's dimensions are: 30mm in diameter, 35mm in length and 100grams. This model does not have as diverse of a feature set like the Linksys model does. For instance, it only supports MJPEG encoding which does not have a significant degree of compression like what will likely be required for smooth reproduction of fast moving objects. This compromise needs to be tested before deciding and developing on a particular platform.

Other factors that seem to be a consequence of form factor are power consumption. The Linksys model is not designed for portable use, as such it consumes 5V, 1A typically while the CWD-TB-11489 uses 3V and 320mA typically. The power consumption of the Linksys combined with its large form factor disqualifies it from the design for the Knight Brawlers project.

Can we use an IP camera? If, we were to purchase an IP camera no work would have to be done except setting the configuration tools to match our requirements. This is not going to be allowed on our project because it is one of the key components in the complexity of the Knight Brawlers design. Most IP cameras are designed for the low-tech consumer in mind which simply does not suit our needs.

4.7.1.3 IP Camera as a Reference Design

Many manufacturers offer reference designs with accompanying software development kits (SDK) for the developer. These designs are intended to shorten the time to market by other companies who are using them for things like image analytics. A typical reference design for an IP camera might include: lens, enclosure, JTAG debug board, power adapter, ribbon cable, RS-232 cable, audio cable, tripod, source code, SDK, and Gerber files are often available as well. This is very attractive to the Knight Brawlers designers. These reference designs often show case their ability to do h.264 encoding at 30fps as the first item in a list of specifications. The quote below from Texas Instruments summarizes the intention of most reference design manufacturers.

> "Texas Instruments offers multiple highly optimized reference designs based on the DM38x, DM812x, DM3xx and DMVAx DaVinci™ video processors for the IP camera market to enable developers to speed through the design process as well as reducing overall bill of materials costs. These reference designs: reduce development time by 90%, deliver higher quality video, up to 10 megapixel at reduced frame rate, optimize electronic bill of materials, and empower customers to design sub $100 HD IP cameras."
> (http://www.ti.com/apps/docs/mrktgenpage.tsp?contentId=41246&appId=79)

The IP camera as a reference design is one of the most encapsulated solutions and total reference designs for the Knight Brawlers project. All that would be left to design is the H-bridge for power control and integrate a Wi-Fi module. The other components could be controlled by the onboard ARM9 processor that runs at 432 MHZ.

One reference design under consideration is the Stretch S6106 which has very robust image analytics and encoding capabilities that the Knight Brawlers wish to integrate into their own design. The main board of the S6106 is 32mm x 72mm with most of the space being taken up in the picture below by the large lens designed to capture high quality high resolution images for surveillance applications.

IP Camera as a Reference Design Summary: After careful consideration the designers have decided that this subset of IP camera solutions will not be included in any of the design of the Knight Brawlers project. The processors on board are too complex for the desired application and budget of the project. The reference designs often include some image processing tools in their toolset which is also not desired in the Knight Brawlers project.

4.7.2 Camera Module on PCB Introduction and Purpose

This section of documentation will introduce the digital camera module on PCB as it will be referred to. The purpose of including this component in its available forms are for several reasons: (1) small foot print allows for easy integration into existing remote controlled car frame that will be used, (2) simple nature of its design allows for flexibility of integration, (3) several options from several manufacturers to choose from.

The simplest of these cameras consist of a CMOS light sensor and lens on a PCB board. Some of the cameras reviewed have circuitry that converts the captured frames to JPEG images, MJPEG video, or NTSC/PAL video.

4.7.2.1 OmniVision OV9665

The OV9655 color image sensor is a ¼" CMOS 1.3 mega pixel device that provides up to SXGA resolution output and on-chip image quality enhancements like fixed pattern noise, smearing reduction, exposure control, gamma control, white balance, color saturation, hue control, and white pixel canceling all through the provided software controls. This device can deliver up to 30 fps in VGA (640x480) for smooth representation of fast moving objects. The chip can scale the image down from the 1280x1024 to 40x30 to fit whichever resolution best matches the Knight Brawlers application testing.

Pros: The OV6955 image sensor has a long list of great features that the Knight Brawlers project can utilize. Most notably is the breadth of image processing features that the chip can do natively before the main microcontroller on the car ever has to touch the image. This chip reportedly has adequate hardware for low light operation as well but testing will have to confirm that claim. This chip's advantage is in color depth and resolution. The intended applications for this product include: smartphones, PC multimedia devices like web cams, toys, and digital still and video cameras.

The most positive aspect of this camera is that Waveshare Electronics sells it for $9.99 on a development board with pages and pages of datasheets, schematics, and even source code for the STM32F4 microcontroller which the group has chosen as the primary controller! The available output formats are Raw RGB, RGB (GRB 4:2:2, RGB565/555), YUV (4:2:2) and YCbCr (4:2:2). These options are desirable but not completely optimal in terms of data size. The most optimal is of course a compressed M-JPEG or H.264 output.

Cons: While the footprint of the image sensor its self is quite small, the board from Waveshare Electronics is large and cumbersome to fit into the tight quarters of the remote controlled car frame. In addition, there is no onboard video encoding. The lack of encoding leaves the developers to have to design a solution to handle the large uncompressed video signals which the designers are making great effort to avoid.

4.7.2.2 OmniVision OV7725 with FIFO

The OmniVision OV7725 with FIFO includes 3Mbits of DRAM with an internal DRAM controller. The DRAM on this chip can do a read/write in 20 nanoseconds which is very quick. The OV7725 is very similar to the OV9655 described above but is better suited to low light conditions and an incredibly fast frame rate of 120fps at QVGA (320x240). The pin layout and most of the datasheet is identical to that of the OV9655 as well.

Pros: The 3Mb FIFO operating at 20ns is the greatest advantage with using this chip. No other camera module that has been researched included this large amount of onboard memory. This enables smooth video feed, ease of synchronization, and even enables video processing on the main MCU because of its large DRAM size which can act as cache or buffer. The camera offers the same output formats as the OV9655 as raw RGB, YUV, or YCbCr to suit the controller’s needs. The lens on top of the image sensor is a high quality glass construction with a strong and light weight magnesium alloy shell. Strength and high quality build materials will be a nice safety feature seeing as how the remote controlled cars are expected to crash into one another at 10+ mph.

This module has a strong collection of firmware available for STM32 MCUs as does the OV9655. Both cameras from OmniVision were used in the SRV-1 blackfin controllers for the Surveyor SRV-1 Blackfin Robot which has many application notes, schematics, and etc. Support is equally above average for the two cameras from OmniVision.

Cons: This interfaces via parallel so there are no fewer than 20 pins that must be connected. The STM32F4 which is the digital camera will be matched to does support parallel connection types with its many pins but RS232 or USB would have been a great alternative. In addition, the OV7725 does not feature encoding on its DSP even with all of the image processing on board. As with the OV9655, this leaves the developers to come up with another solution to encode the video stream with.

The team thought it necessary to conduct some simple testing on the similar models offered by OmniVision. The results are shown below in OV77725 VGA low light (Figure 4.2.7a) and OV9655 VGA Low Light (Figure 4.2.7b), which is a comparison of image quality in low light situations. The designers of the Knight Brawlers project do not expect the users to use the device at night time outdoors, they would still like the image to retain most of its quality through a breadth of conditions. Also it was important to test the different cameras at the same resolution so VGA 640x480 was chosen.

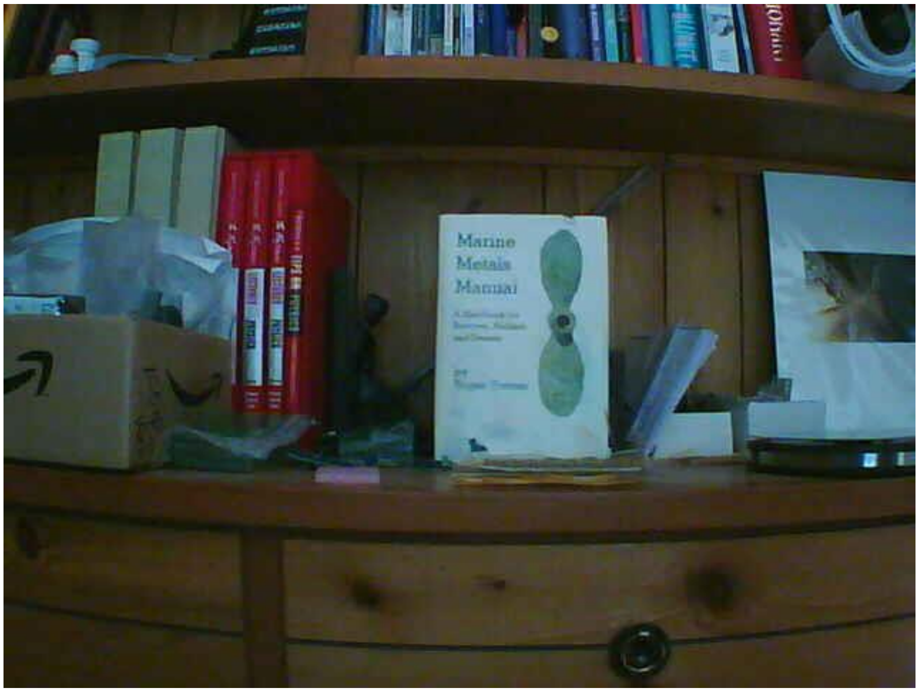


Figure 4.7.2a - OV7725 VGA Low Light

Figure 4.7.2b - OV9655 VGA Low Light

Each picture was taken in VGA 640x480 resolution stacked one on top of the other for equal comparison. One can easily see that the OV7725 far exceeds the OV9655 in low light conditions though the OV9655 appears to have deeper color representation.

4.7.2.3 Vimicro VC0706 TTL Serial Camera

The miniature TTL serial JPEG camera with NTSC video which uses a Vimicro VC0706 DSP has gotten attention from the designers for several reasons. This board is incredibly small at 20mm x 28mm. This is a huge plus compared to OV9655 model from Waveshare Electronics. The camera interfaces with a controller via 3 wire RS232 serial.

Pros: After reviewing the specifications of this camera, the small size is the most beneficial feature. Also, white balance, exposure, and gain control are all automatically controlled onboard. This camera does feature a quick 30fps at 640x480 resolution with a 60 degree field of view and progressive scan for greater image clarity. Another great feature is that the distributer AdaFruit Industries provides tutorials, custom documentation, source code and projects.

Cons: The first negative with using the device is its NTSC analog video output. Though this device is small it likely require an analog to digital video conversion chip so that the android device will be able to play the video. The solution will require another IC with approximately 22 more pins to wire – of course all pins must be soldered well and wired perfectly to work correctly. The second con to this camera is the lack of on board encoding. The IC on the camera's PCB only encodes still frames in JPEG but the Knight Brawlers project requires motion-JPEG for video and not NTSC.

4.7.2.4 Si Cube SB102D

The camera is under consideration for many reasons, (1) High resolution max output at 2048x1536, (2) supports compressed M-JPEG output, (3) image size scaling, (4) acceptable frame rate of 30fps in VGA, and (5) low power consumption. This module is reported by the manufacturer as "not designed as consumer items but are ideal for equipment manufacturers, experimenters and hobbyists".

Pros: The manufacturer claims, superior low light performance, high quality lens, active pixel technology for sharp images and accurate color reproduction. The greatest asset that this camera can provide is its claim to output in M-JPEG. If this is true, the camera can win almost hands down.

Cons: Poor customer support. The designers have already had to contact the manufacturer for documentation because none was provided on their website. What was sent and labeled as a datasheet in once place and a user's manual in another was really just a very vague product brief. This chips is manufactured in and ships from China which introduces several challenges –namely good communication in English. After studying the minimal product brief that was provided, there is a suspicion that the camera can only output in YUV format which is *used by* M-JPEG to compress its image frames.

4.7.2.5 3JTech M-JPEG VGA Camera Module

This is technically not designed for embedded applications as it is a UVC camera but it is sold without a case and does output M-JPEG for light weight video streaming as the application for the Knight Brawlers project requires.

Pros: The device uses the PAS6371 image sensor which supports VGA, QVGA, and QQVGA resolutions with a max transfer rate at 30fps. The USB port can simply transferring the video signal to the Wi-Fi module as it only needs 4 pins. The onboard MCU is programmable and can store new code on its onboard memory. The biggest asset of this camera is its built in M-JPEG encoder which the group really wants to implement.

Cons: This is another board designed, manufactured, and maintained in China. The datasheets are primarily in English but do contain some to dig through and there is not user community support like there is with the OmniVision cameras. Using this camera is a big risk for the team if they cannot get the interface working correctly.

4.7.2.6 Summary of Camera Modules on PCB

Below, in table 4.7.2.6 is a summary of the most significant features and factors for consideration. All the cameras that were reviewed above have been included in the table below. Five cameras in this section have been discussed. Two from OmniVision, OV7725 and OV9655, Vimicro VC0706, Si Cube SB102D, and the 3JTech M-JPEG VGA Camera Module. The designers will likely have to implement one of these that have been mentioned already as opposed to an IP camera or web cam to satisfy the unique engineering design that is required from senior design.

Elimination: First up on the chopping block is the Vimicro VC0706, though its features are very similar to the other cameras and it does compress still images it does not do M-JPEG compression natively. In addition, this chip has an adequate frame rate and great documentation it only outputs video in analog NTSC format which is incompatible with the current design and will be passed over.

	OV7725	OV9655	VC0706	SB102D	3JTech
Res.	VGA, QVGA to 40x30	SXGA, VGA, CIF, to 40x30	VGA, QVGA, QQVGA	QXGA, SXGA, VGA, CIF, to 160x120	VGA, QVGA, QQVGA
FPS @ VGA	60	30	30	30	15
Size	3cm x 3xm	N/A	20mm x 28mm	19mm x 19mm	42mm x 36mm
Focus	Auto	Auto	Manual	Auto	Auto
FOV	60deg	60deg	60deg	N/A	N/A
In Low Light	Excellent	Poor	Fair	Good	Fair
Image Enhancements	EE, hue & sat. contrl, gamma, AGC, AWB	EEE, hue & sat. contrl, gamma, AGC, AWB	AGC, AWB, AEC	Active pixel technology	N/A
Compression	None	None	None	M-JPEG	M-JPEG
Output formats	Raw RGB, RGB (GRB 4:2:2, RGB565/555), YUV (4:2:2) and YCbCr (4:2:2)	Raw RGB, RGB (GRB 4:2:2, RGB565/555), YUV (4:2:2) and YCbCr (4:2:2)	NTSC	YUV/M-JPEG ,BMP/JPG AVI/WMV	YUV M-JPEG
Supply voltage	3.3V 120mW	3.3V 120mW	5V	5V	5V
Connection type	20 pin parallel total/8 data	20 pin parallel total/8 data	3.3V TTL 3 wire TX, RX, GND	USB 4 pins	USB 4 pins
Quality of Doc	Very Good	Excellent	Excellent	Poor	Poor
Onboard mem	384 Kb	None	None	N/A	40KB Rom, 24KB SRAM

Table 4.7.2.6 Camera Features Comparison

The next unit to go is the SB102D from Si Cube. Its poor documentation does not outweigh its reported M-JPEG compression. So what if it does what the manufacturer claims? If it cannot be integrated because of impossible to figure out firmware configuration it will have to be discarded in favor of another camera. The minimum order quantity was also 4 units at $45 a piece which is a big risk for a low budget project.

The USB UVC camera from 3JTech was the next to be eliminated. Its large size and poor documentation are not worth the possibility of late integration completion or failure to integrate completely.

The two cameras from OmniVision, OV7725 and OV9655, are much more difficult to choose from. The OV9655 has excellent documentation and only costs $10 while the OV7725 has onboard memory, highest frame rates available and good documentation. What separates the documentation quality from the two is that the OV9655 is supported by STMicroelectronics, the manufacturer of the main MCU the group is using, which has 20-30 C++ files for quick implementation. The OV7725 has source code but only for the STM3210x. Pre integration testing, no one is sure of the difficulties in porting the code. The designers have decided to purchase both models, test them thoroughly, and make the final decisions then.

4.7.3 Camera Hardware Design

This section will discuss the design and layout of the two selected models, the OV7725 and the OV9655. For the two cameras from OmniVision, the pin layouts are conveniently identical. The only exception is that the OV9655 includes to more optional data lines for raw RGB data format. In table 4.7.3 the pin layout that maps to the pins on the STM32F4 are given. This is a quick reference to get the designers rolling with hardware integration.

Pin#	OV7670 Camera	STM32F4	Description
1	VCC	VCC (3.3V)	Input voltage
2	3.3V	GND	Ground
3	SIO_C	PA5	Serial clock
4	SIO_D	PA7	Serial data
5	VSYNC	PB7	Vertical sync output
6	HREF	PA4	HREF output
7	PCLK	PA6	Pixel clock output
8	XCLK	PA8	System clock input

9	D7	PB9	Data out
10	D6	PB8	Data out
11	D5	PB6	Data out
12	D4	PC11	Data out
13	D3	PE1	Data out
14	D2	PE0	Data out
15	D1	PA10	Data out
16	D0	PA9	Data out
17	Reset	PA11	Reset
18	PWDN	PA12	PWDN

Table 4.7.3 Pin Layout

4.7.4 Camera Software Design

The cameras only require minimal setup to configure the ports, clocks, memory, and the QCIF format. There are two major components that need to be interface from the MCU to the camera. One is the OmniVision serial camera control bus (SCCB) which allows the user to configure the camera from the MCU, and the other is the STM32F4 digital camera interface (DCMI) which allows parallel data transfer from the camera to the MCU. The SCCB is very similar to the UART Rx/Tx configuration but of course it needs a synchronization (clock) and power/ground totaling 5 wires. The reference design that was found totals no less than 359 lines of code just to initialize the SCCB on the STM32F4 to the OV9655 camera. Luckily for us, the code can be ported directly to the Knight Brawlers project because it is available under the GNU Lesser General Public License.

In figure 4.7.4 gives the overview for the initialization (not including SCCB) of the camera. All that is left is to begin gathering frames that are sent right to onboard MCU memory controlled by the DMA.

The DCMI is another 354 lines of code but that can be ported to the Knight Brawlers directly as well. From there, the MCU will wait for a ready ‘S’ char to begin capturing frames one at a time. After stills are being capture and stored to memory video will be captured and sent to the Wi-Fi for output.

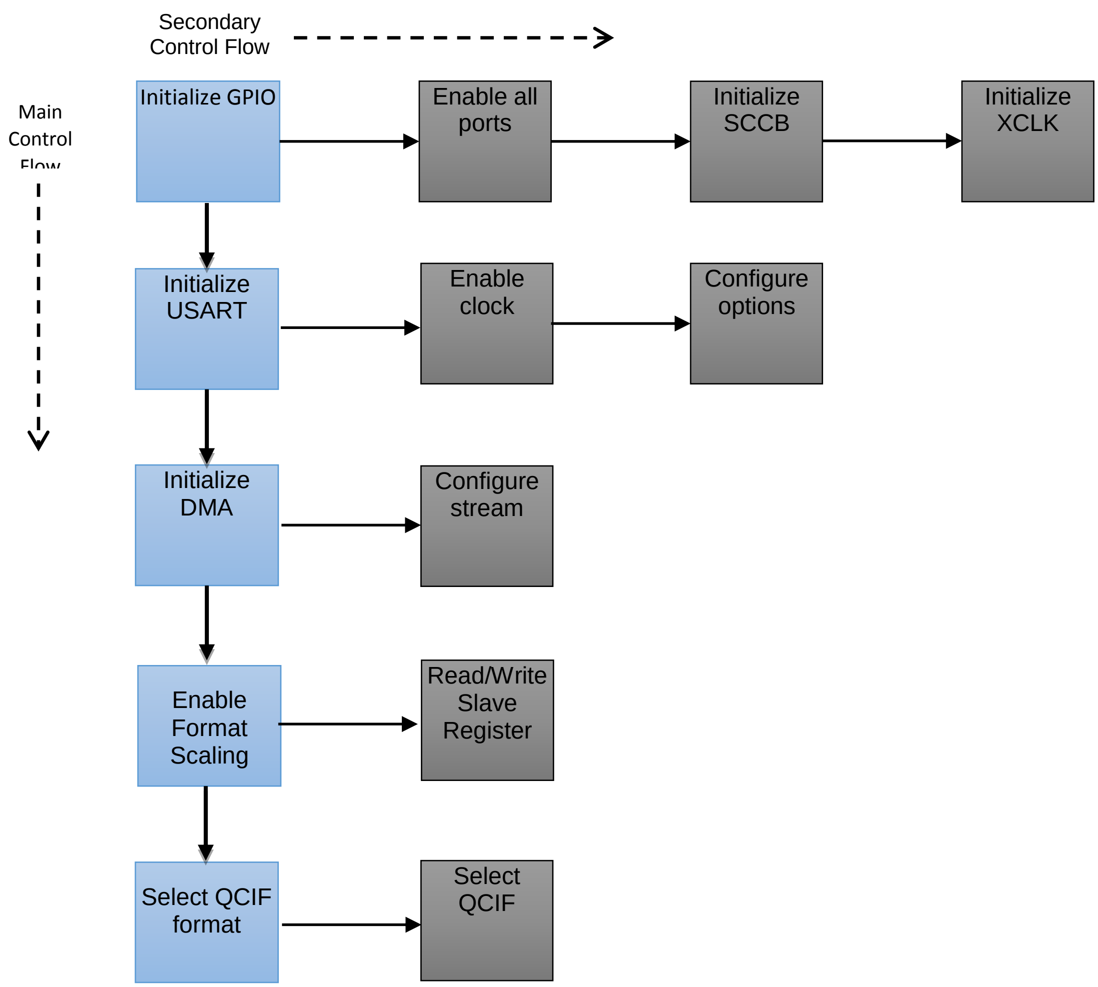


Figure 4.7.4 Camera Initialization

4.8 Smartphone Operating System

We have to deciding on a platform to use as operating system as the basis for the remote control for the Knight Brawler. The Knight Brawler is a system that must be controlled by some kind of remote. In order to present an easily use and richer user experience, the Knight Brawler will be controlled by a custom Android application. The remote control application will be designed to work on specific Android devices and to interact with the RC car. For the operating system, there are three major operating system platforms that we are looking to using which are:

- Apple's mobile platform, iOS
- Window's Mobile Platform
- Google Mobile Platform, Android

We knew it would be more users friendly to use a mobile device as the platform for the remote control. However, deciding between iOS, Windows, and Android was more difficult because of all the different benefits that each operating systems has. But, it came down to a couple of factors to decide which operating system we wanted to use. Table 4.8 shows some important operating systems and information. These are some of the major factors which include:

- Cost
- Availability of knowledge
- Access to the systems

	Programming Language	**Cost to Develop**	**Devices Readily Available?**	**Familiarity**
Android	Java/XML	Free	Yes	Medium
iOS	Objective-C	$99/year	Yes	Medium
Windows Phone	.NET framework/ Visual C++/XNA	Free	No	Low

Table 4.8 Smartphone Operating Systems

First, we took into account the programming language used on each operating system because it is important for us to be allow to pick up this programming language relatively quick and master the different function if we decide to use a language that we don't know how to use. The remote control application requires high level programming, the construction of graphical user interfaces (GUIs), as well as interfacing with built in hardware. iOS uses Apple's own programming language called Objective-C language. This language can easily be figured out by programmers who are familiar with C and C++ programming languages. Unfortunately, we have never used Objective C, but do we have some experience with C++. On the other hand Android uses Java and XML to program mobile applications. This was a plus for our development team, as some of our team already has some familiar with Android programming and with traditional Java programming. Even though, we have about to same amount of experience with each of the programming language so between the programming language they wasn't much of a deciding factor to choose which programming language to use.

Another factor that was considered when choosing a mobile operating system was the availability of reference material and the openness of the developer environment. Also, with the usability of the mobile developer environments is important to be allowing to struggle with locating different parts of the debugging process and code development. Android is well known as an open source development platform. So Android's parent company, Google, has provided hundreds of pages of reference materials on various Android features. The Android developers' website has seemingly endless reference documents on the separate classes, types, and structures built into the Android ecosystem. Similarly, there are many resources for developers on subjects like setting up the development environment, as well as tips for designing the best user interface. Android developers are also free to use any development software they choose.

On the other hand, Apple's development and window phone development processes is a lot more restrictive than the android programming. We must use the development tools that are given to us from Apple or Microsoft unlike Android where they are multiple different platforms of development environment that can be use but there is a main system that is used is eclipse. However, there is a wealth of technical resources and reference material on Apple's developer website similar to those on the Android developer website. But unfortunately Apple requires developers to enroll in the iOS Developer Program to create fully functioning applications. The enrollment fee is currently $99 per year. The enrollment fee was a significant deterrent for us when it came to choosing an operating system.

Android does not charge there developers to develop, but they do charge a registration fee for developers who want to put their applications on the Market. This is an important level for us as a group because once again we do not have any support in pricing materials and developing software we will like to get most of the equipment that we can free. This means we have to make cost an extremely primary factor for us to decide which level development platform we want to use. However, Android allows developers to download their application directly from the development environment to their personal devices. Therefore, for the purposes of the system there will be no need to publish the application to the Android Market, and development of the remote control will essentially be free on the Android operating system.

The final deciding factor that influenced our decision to choose an operating system for remote control for the Knight Brawler was the readily use of the hardware and ability to get multiple platforms for testing and uses. All of us have immediate access to Apple's mobile devices, which is a good thing because often expensive to come by. Alternatively, we don't have immediate access to many Android mobile device due to the fact that only one of us have an android phone, but unlike apple phone they are a bit easier to get for a low price or even for free.

Given the previous knowledge that we have, the cost of development, and the availability of hardware, it was decided that Android's mobile operating system was the best choice for the remote control in the system. Choosing an operating system was only the first step in researching it. Next, we had to determine the various features and intricacies of the Android operating system. Android has gone through several iterations over the years, adding new features every time. However, not all users receive the updates. Even amongst our team, no two devices were running the same iteration of the Android operating system. As a result it had to be determined which versions of the operating system would be developed for.

We decided to develop the remote control for the platforms that the Android devices were most likely to contain the percentage of users who use devices at each operating system iteration. Which make since to us because we could use the platform that the most user are actually already are using today, So it was seen by targeting devices at Android operating system 2.2 or higher, it is possible to develop for 96.5% of the Android ecosystem. By developing for operating system 2.2 and higher, we have also eliminated device compatibility concerns within the completed remote control application.

Choosing the target operating system iteration was also not enough. We also had to ensure that the operating system supported the various features required for the remote control. These features include Bluetooth communication and sensor support for the accelerometer. Fortunately Bluetooth support was introduced as a new platform technology in Android 2.2. Android 2.2 also introduced improved multi-touch gesture recognition. Sensor support already existed in previous iterations of the operating system, so the accelerometer is still supported. With that effect most of most of feature that we would to put in the Knight Brawler will be supported by the android operating system.

4.8.1 Android Software

For the remote control system for the Knight Brawler we will be using an android Phone, so next is deciding the software application and how it will be implemented for the Knight Brawler. There are many different software development kit (SDK) and graphical user interface design that will help us increase the Knight Brawler usability and accessibility. These devices will be any running Android operating system 2.2 (Froyo) or higher which will be implemented at the software development kit before the project is started in the setup section. As a result, programming will be done using minimum software development kit 8, but designed to target SDK 16. SDK 16 will be the most up to date package at the time of programming and represents the most modern and user friendly interface available that is available for any android devices. However, since the minimum package is SDK 8, the application will not be able to use any classes or features present only in later SDKs. But, even without having those classes and features to help us there will be workarounds that will allow us to implement some of these features if it is necessary. This is a useful affect because it will allow us to be allowed to use the android application on a wide range of android operating systems and phones.

A graphical user interface (GUI) will be implemented to the Knight Brawler remote control system to create a more usability to the system and to allow the system to be more interactive. The GUI is a forward facing interface that allows users to control the application through a series of images and shapes. As a result of implementing the GUI, the user will not need much, if any, training on how to use the remote control. Buttons will be used to make decisions. Examples of such decisions are enabling Bluetooth, selecting another device to pair with, and exiting the remote control application. The GUI will be implemented using a combination of XML and Java programming languages. Shapes must be defined in the application's XML code. However, defining the XML objects does not instantly make them perform any action when they are selected by the user. In fact, if the GUI objects are only defined in the XML and not in the Java, then the object will be displayed to the user, but it will perform no action when selected by them. By implementing the GUI objects in the Java portion of the code, actions for each object can be defined. This system will be implemented in on an android development kit 6.1 which is the android development environment that we will be develop the GUI in which is supported and back up from eclipse. Eclipse is a java integrated development environment (IDE) that is used for developing software application and projects.

4.9 Wireless Communication Tech

4.9.0 Introduction and Purpose

To be covered in this section are the available wireless communication technologies that the market has available for the developers of Knight Brawlers. It is imperative that the correct choice be made for the execution of this feature because of breadth of power, cost, and bandwidth variables that are present with wireless communication. The primary and majorly limiting factors are what the current smart phones communication systems have available. Obviously a smart phone has the cell radio on board to communicate audio from one person to the next, and often feature Bluetooth and Wi-Fi. These common features on modern smart phones will be reviewed as to which best fits the Knight Brawlers feature set.

The wireless communication technology chosen will also affect the codec used to compress the video stream, total project costs, difficulty to implement, response time from user input to action on the car, quality of video feed able to be transmitted, power consumption, and likely other factors yet to be discovered until the design phase. All these affects will need to be taken into consideration during the research phase.

It is important to understand the basics of these technologies because for our project since we will likely have to implement custom firmware to control our wireless devices. For our project, it is desired to have as great of a data-rate transmission as possible to facilitate a fluid video stream to our smartphone devices. Selecting the highest quality WiFi protocol, coupled with an efficient video codec, will enable an enjoyable experience for the users.

4.9.1 Wi-Fi

To be covered in this section is the 802.11 family protocol. Under each subheading range, bandwidth, costs, availability, and difficulty to develop will also be covered. At the end of the Wi-Fi discussion will be a summary of the Wi-Fi technology and an estimate of likely use in the Knight Brawlers project.

4.9.1.0 802.11 a

First up is the 802.11 a protocol. This protocol operates on the 5.75 GHz frequency which has a few advantages and disadvantages. The 2.4 GHz frequency is so commonly used that interference with other tech devices is common –like microwaves and cordless phones. Operation at the 5.75GHz range alleviates this problem and 802.11a further contributes by enabling 23 channels in the United States, twice that of the 802.11 b/g. Having such a high number of channels and a relatively clear frequency range is desirable because it helps to ensure reliable data transfer from point to point. 802.11 a can transmit data at a rate of 54 Mbps ($54*10^6$ bits per second). During initial precursory research on transmitting a video signal, the bandwidth necessary will likely be around 25 Mbps. If that is true then this protocol can handle the video transmission with room to spare.

The available 802.11 a modules are not any more or less expensive. It appears many of the modules are capable of a, b, g, and n protocols all on the same chip. This is great for the designers as it will not set drastic limitations on the design if during testing, our chosen protocol fails significant benchmarks.

The availability of a Wi-Fi module with the “a” designation is not a problem. Many manufacturers offer a range of specifications. However, on many smartphones with Wi-Fi, the “a” designation is not available. One member has a Samsung i727 which only supports b, g, and n protocols. This will be a significantly impactful con on the choice of this protocol.

Range is limited at the high 5.75 GHz range because the higher frequencies get dissipated through solid objects like walls. Operating the remote car in another room could pose problems is the users decided to do that.

4.9.1.1 802.11 b/g

Wi-Fi connectivity commonly advertises the 802.11b/g specification. The 802.11 b/g transmission specifications are subsets of the 802.11 physical layer specifications but exclusively operate on the 2.4GHz frequency. 802.11 b can transmit and receive data at a max of 11 Mbps while 802.11 g can transmit and receive at a max of 54Mbps. More is better! These two protocols are so common and widely used in today’s Wi-Fi enabled consumer devices that one can almost always assume that b/g is automatically compatible.

The 802.11 b/g protocol is well developed with many low power options available. One is from TI, the CC3000 BoosterPack and evaluation board, which can operate the “g” designation at 190/207 mA (typical/max) and 3.6 volts. This component has been designed for the use on battery powered devices so it fits our low power requirement well.

4.9.1.2 802.11 n

One protocol that is also commonly used is the 802.11n. From the same family as the previously mentioned protocols, the “n” designation means that a device featuring this technology can operate on either 2.4 or 5GHz. The “n” amendment enables this dual frequency output by implementing multiple-input multiple-output antennas (MIMO). The “n” protocol also allows for up to 700 Mbps of raw data, according to IEEE but support for that bandwidth is unknown.

As far as availability is concerned, there are many manufactures that support this protocol. Many smartphones also support this protocol so availability should not be a concern.

Because the 802.11 n does support much greater bandwidths than the a/b/g specs, the pricing range can be two-four times as much. The MIMO technology is likely the major contributor to the increased costs.

One 802.11 b/g/n evaluation board the designers came across is the WizFi630 which can operate at a max of 150 Mbps but requires 3.3 volts and 600mA typical (1A max) to operate. This is a huge detriment when choosing the n protocol. To get a greater bandwidth it is apparently necessary to sacrifice the low power requirements.

4.9.1.3 Summary of Wi-Fi 802.11 Specifications

The common members of the 802.11 family have been reviewed including the a, b, g, and n protocols. A summary of available configuration are shown in table 4.9.1.3. As you can from the table below, b/g protocols offer a nice midway point between price, availability, and bandwidth. While none have been clearly shown to be the obvious winner there are two key players – the 802.11 g and n protocols. The g protocol offers low power options (even more so than the b) and a sufficiently low cost/low power option. The Knight Brawlers project will likely go with the 802.11 b/g Wi-Fi module from TI because of the above mentioned pros - unless bandwidth takes a much larger priority than initially calculated.

802.11 Protocol	Freq (GHz)	Data Rate (Mbits/s)	Apx Indoor range (m)	Apx Outdoor range	Apx Cost	Availability
a	3.7, 5	6, 9, 12, 18, 24, 36, 48, 54	35	120	\$15-\$115 Average is ~\$35	Fair to good
b	2.4	1, 2, 5.5, 11	35	140	\$15-\$115	Great – most Wi-Fi dev boards use this protocol
g	2.4	6, 9, 12, 18, 24	38	140	\$15-\$115	Great " "
n	2.4, 5	7.2 – 700	70	250	\$45-\$150	Fair – the technology is more complex which affects price and availability

Table 4.91.3 Summary of Wi-Fi

4.9.2 Bluetooth

The wireless specification called "Bluetooth" has a variety of connection speeds and bandwidths that are of great concern and consideration for the Knight Brawlers project. The first thing that will need to be considered is how a network is established and maintained. This information will be relevant to the project because possibly many different user cars will be in use at the same time so multiple players all synched to the same simple network is desired. Next the connection speeds will be reviewed in order to determine if any of them are a good fit for transmitting wireless video.

4.9.2.0 Bluetooth Networking

Bluetooth specification implements a standard called the personal area network (PAN). The PAN is the means that Bluetooth uses to connect two or more Bluetooth enabled devices to a common network that is inaccessible to outsiders, oftentimes only unless a connection access code or pin has been granted and entered. This methodology creates an environment that is sterile from outside attackers who wish to take advantage of the weaknesses in Wi-Fi technology to gain access to someone's personal data. This is beneficial to the Knight Brawlers project as the designers do not wish to compromise their user's personal data that might be accessed via a WLAN.

One of Bluetooth's primary features is its low power consumption designed for lightweight battery powered embedded devices. This combined with its low cost to include a wireless module on the Knight Brawler's main board makes Bluetooth an incredibly attractive. With that said, there is not getting around the need to transmit a 640 by 480 resolution image at 600kbps consistently.

4.9.2.1 Bluetooth Bandwidth

Bluetooth 2.0 is the most common specification for Bluetooth as of the writing of this paper. The 2.0+EDR spec claims to operate at a max of 2.1 Mbps but has a typical transmission rate of 200 Kbps to approximately 1Mbps. These numbers are practical claims from developers on small scale projects similar to the scope of the Knight Brawler's. This means that the max reliable bandwidth of a Bluetooth 2.0+EDR device is not sufficient.

The Bluetooth 3.0 HS and beyond support 24 Mbps but only through an established 802.11 connection meaning that a Bluetooth 3.0 HS module only uses the Bluetooth communication spec to set up and establish the connection to another device then transmits data via Wi-Fi protocol. These 3.0 HS modules require a Wi-Fi radio because they actual use Wi-Fi transmission. Learning leads the designers of the Knight Brawlers project to forgo the use of Bluetooth on the project.

4.9.2.2 Summary of Bluetooth

From the table 4.9.2.2 it shows the limitations of Bluetooth's data transmission speed – at its max, it is half that of 802.11g. Bluetooth offers a range of options in terms of power consumption, data transmission ranges, data transmission bandwidths, and available hardware (development board). Even with all of the available options in mind, it appears that including Bluetooth as the primary means of transmitting a video signal is not practical for the Knight Brawlers. No team member specifically desires Bluetooth experience so Bluetooth is simply just an option. Range is significantly limited to 30 meters and that only under optimal line of sight conditions. The added complexity of the Bluetooth 3.0 HS hardware leaves the designers asking the question, "Why bother?" especially since Wi-Fi offers higher bandwidth options at a similar level of complexity and power consumption.

Version	Data Rate (Mb/s)
1.0	.721
1.2	.721
2.0 + EDR	2.1
2.1 + EDR	3
3.0 + HS	24
4.0	24

Table 4.9.2.2 Bluetooth Data Transmission Speed

4.9.3 Other Wireless Communication Technologies

There are other solutions to transmit a video signal from a microcontroller; however, the designers of the Knight Brawlers project are severely limited by the compatibility of the common smart phone communications systems. XBee, a popular manufacturer for small wireless communication modules, has no less than 16 different model to choose from with all sorts of various radio frequencies and wireless protocol specifications. Smart phones consistently provide Bluetooth, 802.11 b/g/n protocols and the cell radio. Therefore, no other options can be considered.

4.9.4 Choosing the Wi-Fi Module

As previously mentioned, the TI CC3000 will likely be chosen as the group's Wi-Fi evaluation board. This section will therefore be a summary of the Wi-Fi modules under consideration.

4.9.4.0 WizFi630

The WizFi630 is a gateway module that converts the UART RS-232C protocol to Wi-Fi 802.11 b/g/n with robust development options. A development shield is available to quickly connect the board's mini PCI express card slot to 2 serial UART connectors and 3 Ethernet ports. Included in the evaluation package is, a 2dBi Wi-Fi antenna, serial cable, LAN cable, and DC 5V/2A adapter. This board's primary reasons for consideration are, (1) quality documentation, (2) available SDK, (3) many operational modes, (4) fastest bandwidth at a max of 150 Mbps and a 2 UARTs at 921,600 bps, (5) GUI configuration utilities

Pros: The features listed above are all pros. The Knight Brawlers project is going to design a wireless communication standard that can stream a raw 640x480 eight bit color frame at 25 fps which comes out to approximately 61.44 Mbps. There are forums online that report that this is possible but testing will have to verify. This is a great Wi-Fi module for development purposes and seems to support all the necessary requirements, namely, the high data bandwidth and good documentation.

Cons: There are several cons to weigh when considering this module. The first is the higher costs. The evaluation board is $132 from Mouser Electronics and the standalone receiver/transmitter is $42. The next is the breadth of pins on the modules' PCI express interface including no less than 52 pins to map. In addition, there is a lack of example projects using this particular board for any solutions. However, there is no firmware necessary as the board interfaces to a MCU via the UART specification and is configured on a Windows machine with an included GUI script utility.

4.9.4.1 Texas Instruments CC3000

The TI CC3000 is intended to simplify implementation of wireless internet connectivity by minimizing software requirements of the MCU. This is an excellent start, simplicity is greatly desired where it can be attained without sacrificing quality and application practicality.

Pros: The primary benefit from implementing this module would be Texas Instrument's thorough documentation, software support, and customer service. This is invaluable when difficulties arise that are out of a novice's abilities and bugs need to be fixed quickly. Another consideration to go with TI is the low power requirement. The CC3000 operates the faster 802.11 g protocol (54 Mbps) at 190/207 mA (typical/max) and 3.6 volts which is a third of the current draw that that from the faster WizFi630.

Part of the beauty of using products from a large corporation which emphasizes its customer service and support is that the result from their efforts is often very user friendly. Easier to use products means less work on their part. For example, the CC3000 implements SmartConfig™ technology which claim to only require 6 API calls to get the unit configured to transmit and receive data.

Cons: The CC3000's slower data transfer rate and limited clock to 16MHz is the only negative aspect. This could be a potential deathblow to the Knight Brawlers project if video compression is not used or if all conditions are not optimal to use all of the available bandwidth.

4.9.4.2 Realtek RTL8192CU Wi-Fi Dongle

The Realtek RTL8192CU is actually a Wi-Fi dongle which is a transmitter, antenna, and control unit all in a package that is only a few millimeters larger than the male end of a USB port. Those the Wi-Fi dongles are not friendly for embedded designers, they are hack-able and contain all the necessary ingredients for a successful high-speed data transmission at rates that would otherwise be impossible to achieve. The reason that this particular unit is included in the run-down is because a similar project was found online forum using the same MCU, STM32F4, and the RTL8192CU without any apparent challenges.

Pros: The unit's incredibly small size and high bandwidth are its best features. The RTL8192CU can achieve 300 Mbps using the IEEE 802.11 n specification. There are likely going to be limitations to reaching that high of a number for any length of time but the possibilities are there. This module is also very inexpensive and can be purchases for $10.48 at auto-gadget.com

Cons: Obtaining or creating a driver in C for the RTL8192CU will be very difficult. The user in the forum that was mentioned above made no comments on the difficulty in getting the drivers to configure properly. The lack of manufacturer software support puts using a Wi-Fi dongle on the bottom on the list of available options. Also, there is not a single datasheet available for this device as the market is for consumers not designers.

4.9.4.3 Wi-Fi Module Conclusion

Of the three models that were chosen for consideration, each was hand picked to be the best representative models in its class. The RTL8192CU from Realtek is a mini Wi-Fi dongle used for desktop and laptop computers to easily interface with an operating system and is not intended for development purposes. This particular model has been shown to be hack-able and was used in a very similar application in conjunction with the same MCU, the STM32F4. However, because of the difficulty in porting the Linux drivers available to C++ and the designers' complete inexperience with the task, this Wi-Fi module will no longer be included in the design considerations.

The next to be removed from design is the WizFi630. The WizFi630 has a very high bandwidth operation at a max data rate of 150Mps which is the fastest for a Wi-Fi development board platform that was found. This board was promising because of its software support, thorough documentation, and high data bandwidth but will be the runner up to TI's CC3000. The two primary reasons that this unit get second place is because of the lack of reference designs and complex physical interface (the PCI express).

Below table 4.9.4.3 contains a concise review of all of the more important features that are under consideration for the project. The WizFi630 unashamedly admitted their effective bandwidth at 90 Mbps which is a reduction of 40% from the claimed max at 150 Mbps. This brings the designers to logically question the effective bandwidth of the slowest module, the CC3000. The support, documentation, and ultimately the reference designs (source code) lead the designers to go with the module from Texas Instruments. This decision is not without its limitations as a raw 640x480 8 bit color depth and 25 fps image will require 61.44 Mbps of bandwidth. This number will have to be reduced by decreasing the frames per second and resolution. If compression is able be achieve that may allow to the designers to keep the original resolution, frame rate, and color depth. Real time testing will have to determine what is possible.

	WizFi630	RTL8192CU	CC3000
IEEE protocols	802.11 b/g/n	802.11 b/g/n	802.11 b/g
Max Bandwidth	150 Mbps	300 Mbps	54 Mbps
Effective Bandw.	90 Mbps	N/A	N/A
Operation Modes	Gateway, AP, AP-Client, Client, AD-HOC	Infrastructure, AD-HOC	N/A
Volts/Current	3.3V, 600mA, 1A (typical/max)	3.3V	3.6V, 190/207 mA (typical/max)
Size	33 x 43 x 4.5mm	N/A	16.3 × 13.5 × 2mm
Interface	PCI express (52 pins)	4 pin USB 2.0 and 2.0 HS	20 pin header, SPI
Software Support	Good, Windows Utility, Web Server, Serial Command	Linux/Windows/MAC drivers available	Sample projects, Radio Tool Package
Doc. Quality	Good	Poor	Very Good
Technical Support	Good	Poor	Very Good
Ref. Designs	None	Poor	Very Good

Table 4.9.4.3 Wi-Fi Options Review

4.9.4.4 Wi-Fi Hardware Design

In this section, an overview of the Knight Brawlers' Wi-Fi module implementation will be discussed. The first step is to determine what pins need to go where. The CC3000 will be initially purchased as an evaluation board that needs to be configured to the STM32F4's SPI connection pins. During development and testing the physical link will be male (CC3000 side) to female (STM32F4) jumper wires. In table 4.9.4.4a the evaluation board pin connections are shown, they will eventually be integrated into the main PCB that will be designed for the final product. Below in figure 4.9.4.4 is the most fundamental communication configuration that the SimpleLink CC3000 implements. The protocol is called serial peripheral interface (SPI) and only needs 5 pins to fully function. Table 4.9.4.4b describes the pin names and functions are in table 4.9.4.4c

Pin Name	Description
SPI_CLK	Synchronous clock line, 16MHz max
SPI_CS	Signal to indicate incoming signal (active low)
SPI_DIN	Data in from master
SPI_IRQ	Interrupt from slave
SPI_DOUT	Data from slave to host

Table 4.9.4.4a SPI Pins

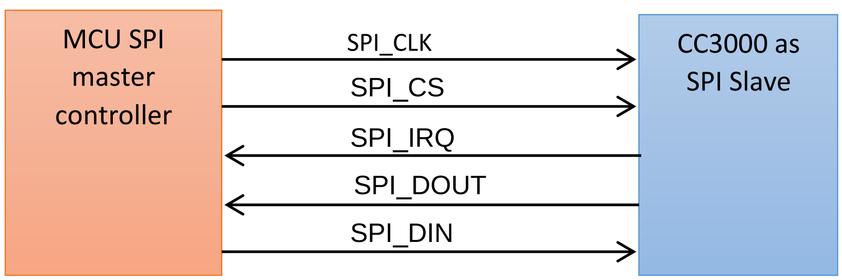


Figure 4.9.4.4 CC3000 Main Connections

J6 Pin	Pin Name	Module Pin Type	Description
1	GND	-	GND
5	Reserved	-	Reserved
10	VBAT_SW_EN	Input	Act hi, Enable sig from mstr
12	WL_SPI_IRQ	Output	Mstr interface SPI interupt req
14	WL_SPI_CS	Input	Mstr interface SPI CS
16	WL_SPI_CLK	Input	Mstr interface SPI clk input
18	WL_SPI_DIN	Input	Mstr interface SPI data input
19	GND	-	GND
20	WL_SPI_DOUT	Output	Mstr interface SPI data output

Table 4.9.4b CC3000 J6 Header Pins

J7 Pin	Pin Name	Module Pin Type	Description
2	GND	-	GND
7	VBAT_IN	Power In	Input voltage
9	VBAT_IN	Power In	Input voltage
15	Reserved	-	Reserved

Table 4.9.4c CC3000 J7 Header Pins

Although the SPI interface only needs 5 pins to transfer data, there are an additional 8 pins that must be connected from the J6 and J7 headers. Some are ground, input voltage to power the device, or the enable signal from the MCU. Table 4.9.4c and 4.9.4d describe the pins and what they connect to.

4.9.4.5 Wi-Fi Software Design

The TI SimpleLink CC3000 is designed for simplicity, even in the software. Below in is the first time initialization of the software configuration. Other events are required to stream data but they will be omitted at the present. This setup below in table 4.9.4e is configured to the CC3000's non-volatile memory for later access.

Description of code in MCU
External event triggers connections to a give AP, which SSID and security keys are known
MCU waits for First Time Config done event
First Time Config done. Auto connect to AP in future sessions
Config CC3000 to connect automatically to receive AP
Reset CC3000 to apply policy
Wait for event of connection to AP

Table 4.9.4e CC3000 First Time Setup

Below in Figure 4.9.4b – Wi-Fi Software *Flowchart* is a flowchart of the initializion, configuraiton, connection, and data reception for the CC3000. This summarizes the control flow for the CC3000 device.

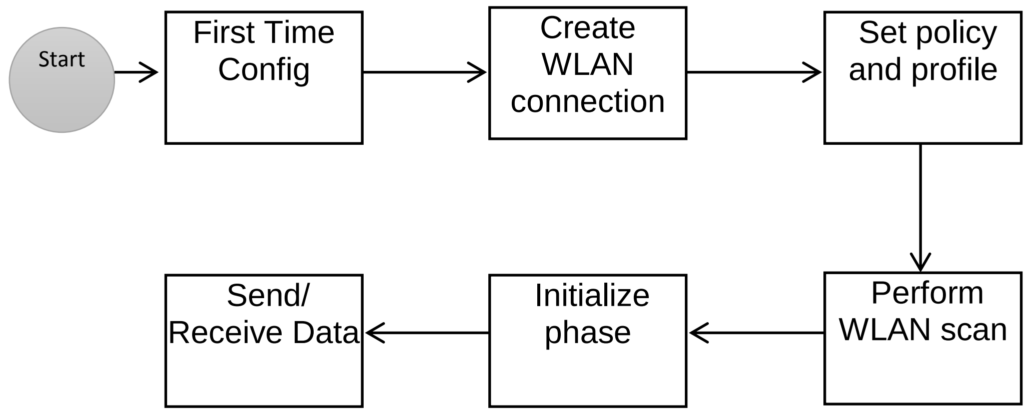


Figure 4.9.4b – Wi-Fi Software Flowchart

4.10 CODECS

After the discussion on “wireless communication technologies”, showing a typical bandwidth of 54 Mbps (megabits per second or 6.75 megabytes per second) for the 802.11 g protocol, it is important to have a way to transport our video data efficiently. For example, one second of uncompressed 1080p video with 36 bits/color and at 25 frames per second results 699.84 MB of raw video data. The cost to transmit that type of raw data wirelessly is far beyond our budget – in steps video codes.

A codec can be used to compressing raw video (through a process called encoding) data into a format that is more easily transported and stored. Conversely, a codec can also be used to decode an encoded file or file stream to a format that is able to be manipulated by the processor. Encoding/decoding, transcoding, and video conversion are all interchangeable terms which can often confuse those new to this technology. Encoding is usually done by the device that is acquiring the multimedia data to store, as with a video recorder; in our project however, the hardware will be transmitting the data wirelessly with no recording option. Many digital video cameras offer various encoding options right on the unit itself.

4.10.0 M-JPEG

Motion-joint photographic experts group (M-JPEG) is known for its ability to indiscriminately compress a video to a fixed size independent of motion from one frame to the next. M-JPEG does this through a process called intraframe compression which is a compression concept that disregards the previous and next frames’ content to decide how to compress the current frame. M-JPEG gets its name from because it uses the JPEG algorithm to do the compression. MJPEG is often wrapped in AVI format but this project will likely by-pass that step if possible.

4.10.0.0 Pros of MJPEG

One benefit of the MJPEG codec is that it is one of the lightest weight, computationally speaking, in use today. This codec can consistently compress video data at approximately 1:20 ratio. MJPEG also does not need more processing power due to high amounts of movement from one frame to the next –this will increase the total frames per second delivered to the network card to be transmitted wirelessly to the user's smartphone. This codec uses a familiar algorithm called Huffman encoding. One of the members already has experience with the Huffman encoding process and has seen the effects the algorithm on raw text data first hand. This lends itself to its simplistic nature – great for a senior design project component. The simple base-line algorithm, as it's called, will also allow the developers to write their own codec like this one a microcontroller without any supported options. For that reason, this codec would be the choice for new processors without any or little supporting libraries and for microcontroller that operate at a slower clock frequency, say on the range of 300 MHz.

4.10.0.1 Cons of MJPEG

Because of MJPEG's still frame encoding format, this codec does not have the higher compression ratios on the order of 1:50 verses 1:20. For the designers on the Knight Brawlers project this means that there is likely a significantly less bandwidth in terms of frames per second that can be transmitted wirelessly, especially if we are forced to go with the 802.11 b WiFi protocol. The designers are aiming for 25 fps for a 640x480 frame size; this will require a fast connection and the video feed will likely eat up most of the transmission data on the WiFi connection.

4.10.0.2 Summary of M-JPEG

If the developers of Knight Brawlers are to choose a codec, it will greatly depend on the chipset chosen. Will our microcontroller board also have a DSP or not? Will our microcontroller board have a clock speed of around 1 GHz? These are all factors that must be answered. Also the difficulty in using any codec on a particular microcontroller can present major roadblocks as well if the microcontroller does not have any supported libraries for the codec. If the libraries are not available and the designers still choose to go with the unsupported codec than they would have to write the code themselves. This should not be too difficult of a problem, time permitting, for a choice of MJPEG.

4.10.1 MPEG-2

Introduction and purpose: Like most codecs, MPEG-2 offers many specifications that offer a range of compression ratio commonly in the range of 6:1 to 13:1. This codec is very similar to the JPEG codec for stills and subsequently the M-JPEG for video. MJPEG's core technology is the utilization of JPEG's lossy Discrete Cosine Transform (DCT), quantization, and entropy processes. MPEG's algorithm also takes advantage of redundancies in images like repeated colors and textures. [Ruiu, Dragos. An Overview of MPEG-2]

Inter and P-frame utilization: This is an excellent feature of the MPEG-2 codec which attempts to take transmit some frames as only the most important differences, those pixels that have moved, so as to further compress the image. MPEG-2 attempts to remove error by forcing the transmission of a compressed version of an original detail frame every 12 frames or so.

Algorithm Overview: The primary steps are to (1) acquire RGB raw image data, (2) convert to YUV 4:2:0 lossy sampling, (3) encode a intraframe which includes DCT, quantization, and entropy, (4) attempt to encode a P-frame containing the difference + motion vectors, (5) JPEG huffmann encoding to further compress data, (6) MJPEG-2 bit-stream. Figure 4.10.1 - MPEG-2 Flowchart shows the progression.

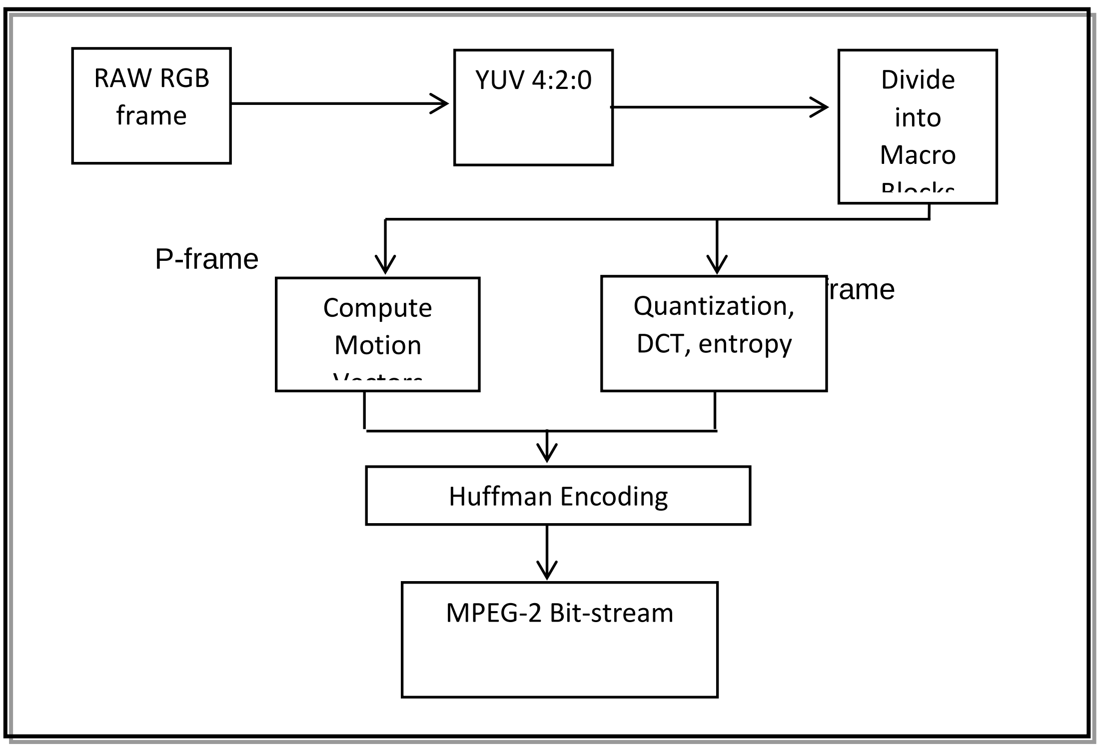


Figure 4.10.1 - MPEG-2 Flowchart

MPEG-2 Summary: This codec specification has many favorable qualities that the designers of the Knight Brawlers mark as a positive asset. Because MPEG-2 has been out since 1995, the codec is now available for free download, both in executable and C/C++ source code format. Combining this information with its rather simplistic approach to encoding video, MPEG-2 can stand for possible revision by adding to taking away from its original form to fit the needs of Knight Brawlers more closely. MPEG-2 is a reliable choice that has been used in many light weight processor applications so the designers of Knight Brawlers are excited to see how this codec performs during testing. With that said, real-time application testing will have to be the final verdict.

4.10.2 MPEG-4 part 2

This specification defined by IEEE has many different aliases: MPEG-4 part 2, h.263, Visual Coding, or just MPEG-4. This standard will be referred to as MPEG-4 during the rest of this document. MPEG-4 was designed for applications ranging from low resolution/low bandwidth requirements all the way to the higher resolution/high bandwidth requirements for 1080p quality encoding. The primary application that developers have attached themselves to for this codec is for those video encoding systems that do not require high definition (720p or 1080p) image processing. MPEG-4 part 10 (discussed below and commonly referred to as h.264) is better suited for high resolution (on the order of 1920x1080 progressive scan).

In terms of quality, MPEG-4 offers progressive scan and interlaced video but a low chroma sampling of 4:2:0 for the Simple Profile (SP) which highlights its “light weight” encoding compression ratios.

4.10.2.1 Profiles

This codec can operate under several different profiles. The Simple Profile has been established to facilitate users who need a low bit rate/low bandwidth resolution, often for transmitting and streaming video over the internet. It is implied that the input video stream is also a low resolution. Examples for this profile would also include electronic surveillance systems, low end cell phones, and video conferencing systems. See table 4.10.2.1. To review a comparison of profiles SP to ASP and their corresponding resolutions. The Knight Brawlers developers desire to stick to their minimum resolution of 640x480. Profile level 5 for SP and ASP are the closest to their desired goal. Also, SP supports the more straight-forward intraframe and interframe compression technique. The former encodes each frame independent from the previous and next. This is great for a fixed and predictable output bandwidth and lower CPU usage but compromises the total compression ratio possible.

Profile	Level	Typical Resolution
Simple Profile	0	176 x 144
Simple Profile	1	176 x 144
Simple Profile	2	352 x 288
Simple Profile	3	352 x 288
Simple Profile	4a	640 x 480
Simple Profile	5	720 x 576
Advanced Simple	0	176 x 144
Advanced Simple	1	176 x 144
Advanced Simple	2	352 x 288
Advanced Simple	3	352 x 288
Advanced Simple	3b	352 x 288
Advanced Simple	4	352 x 756
Advanced Simple	5	720 x 576

Figure 4.10.2.1 Profiles

MPEG4's use of the Advanced Simple Profile (ASP) encode with the same sampling as H.263. In addition to the features of the SP, ASP also allows for the use of interlaced video, B-frames, quarter-pel motion compensation, more quantization tables, and global motion compensation. ASP is not known for its quality motion compression (which is desired) so if MPEG4 is chosen, not all features will be implemented.

4.10.2.2 Overview of Algorithm

The MPEG-4 part 2 and part 4 both use interframe compression in the Advanced Simple profile. Interframe compression starts by compressing one or more reference frames, then it analyzes the next frame and compresses only the differences. MPEG-4 compresses and transmits a combination of I-frames, P-frames, and B-frames. Below is an overview of their definition and interrelatedness.

- I-frames: a.k.a. intraframes, key frames, access frames; These frames are the reference frames that the algorithm uses to refer back to do in order to encode the P-frames and B-frames. These frames contain all data in a given frame. As the video changes from scene to scene, new I-frames must be created.

- P-frames: a.k.a. predicted frames; These are motion compensated frames which only comprise information that has changed from the previous I-frame so that data in not recompressed and retransmitted. The differences that a P-frame utilizes are those differences based on motion as opposed to a change in background. The computation of a P-frame can be complex and the implementation of this type of frame adds to the computation power needed.
- B-frames: a.k.a. bi-directional frames; These may contain image data and motion displacements which help to predict the next frame. This frame gets its name by allowing two motion compensation vectors per macroblock of image data.

4.10.2.3 Summary of MPEG-4

MPEG-4 is a robust codec with many features that encapsulate our needs for a reliably, low bit rate, low bandwidth, and medium to high quality over wireless transmission. This codec works well on low end processors, such as the ARM cortex A8, which the Knight Brawlers team is most likely going to implement. In comparison to MPEG-2 (MPEG-4's predecessor), the MPEG-4 codec is not necessarily a better codec but it does offer a more wide range of encoding and decoding specifications that allow for higher quality video streams per the same bit rate than that of MPEG-2. With these factors in mind, it is not without its faults. *Quality* motion compensation is left wanting but may be acceptable come testing time. Real time application testing will decide which codec is best.

4.10.3 H.264 MPEG-4 part 10 AVC

Taking one codec known for its high compression ratio and quality retention, h.264, and the example of raw data sizes from "Introduction and Purpose", we can see a practical example of how a codec may perform. Applying the h.264 codec from the raw data at 699.84 MBps (megabytes per second) results in a stream size of approximately 10.5 MBps after compression. This is a dramatic reduction of file size and necessary bandwidth; the new encoded format can now be transmitted wirelessly with some latency visible on the user's side. The bandwidth requirements can be further reduced by simply lowering the resolution of the video stream. For example, 480p resolution and 25 fps can result in 2.5 MBps or less.

Features: The way H.264 achieves its high compression ratio is primarily through the intelligent use of interframe encoding. Interframe encoding is the method of using previous frame encoding to help decide how to and how much to encode on the current frame. H.264 allows up to 16 reference frames to handle this task. The developers of the Knight Brawlers team wish to test the efficiency of this technique by playing the bumper car game in a fixed size area with a white wall. The repetition of the white background should enable the interframe processing to do its job well and lighten the bandwidth load. Real time application testing will verify if the theoretical results translate to useful technology.

H.264 – A Reliable Choice: H.264 is most commonly used in high definition encoding because of its high compression ration but is also useful in lower resolution video as well. The reader might find it interesting to know that it is the codec of choice for all Blu-ray disc players. This codec is also gaining popularity with streaming internet sources like YouTube, and Vimeo. The adoption of this technology by such websites is encouraging to the designers on the Knight Brawlers project because it has shown itself proven for streaming vast amounts of multimedia data via the internet. To use technology that is proven, well documented, and accepted by the majority speaks volumes about its overall quality and practical usefulness.

Availability: The H.264 codec (software solutions) is available for 5 out 7 digital signal processor chips from Texas Instruments. This codec is incredibly popular and every manufacture that offers HD image solutions is sure to include H.264 in some form or another. However, because H.264 is such a complex encoding algorithm requiring significantly more processing power and memory on board compared to JPEG, many H.264 camera modules (hardware solutions) also offer image processing for image analytics which increases the cost. Many camera modules are available from China but without much documentation, software support, or reliable shipping standards. A few reliable sources for complete camera modules supporting direct encoding to H.264 have been found. Application specific hardware support for H.264 is fair.

4.10.4 Codec Summary

The most important factors concerning the codecs in the Knight Brawlers project are listed below table 4.10.4. After review the information and the careful consideration of the hardware requirements and the impacts of those factors, the designers of Knight Brawlers are choosing to go with the pure intraframe code solution M-JPEG. The hardware complexity to implement the more efficient algorithms like H.264 are beyond the budget of the Knight Brawlers designers and the scope of the complexity involved in implemented those technologies is also beyond that of the Knight Brawlers project.

Codec	Expected Compression Ratio	Pros	Cons	Conclusion
M-JPEG	10:1 to 15:1	Requires least CPU power Predictable bandwidth consumption Loss of one frame will not affect others	least compression ration Requires greater average bandwidth	This is the baseline encoding standard that Knight Brawlers will likely implement primarily due to low CPU demands
MPEG-2	31:1	Proven, reliable, available source code	This interframe prediction codec requires more CPU power than intraframe compression Outdated	This codec will be tested on our processor of choice and is the likely candidate for predictive compression
MPEG-4	45:1	Profiles for all levels of bitrates and quality Good overall compression ratio	Outdated Significant memory and CPU consumption	Heavy processor demands will likely cause designers to forgo this option
H.264	60:1	Highest compression ratios possible Maintains quality	High-end hardware requirements	Heavy processor demands will likely cause designers to forgo this option

Table 4.10.4 Video Summary

4.11 Encryption Technologies

Wired Encryption Privacy (WEP) is an outdated security standard (by 10 years) that has been replaced by WPA and now WPA 2. The new version of WPA uses dynamic packet key encryption, creating a new authentication key on each network card for each packet of data sent and many other safety features that had been previously exploited in WEP and WPA. With that said, it is now 2013 and many weaknesses and exploits have been found in even the most robust encryption standards. For our project, however, security is not a significant concern. Our project does not contain any information that is confidential and must be hidden from unknown persons. One concern we do have is that forcing the users of Knight Brawlers to subscribe their home network to a protocol with known security threats like WEP would cause their information to become vulnerable. Though the security vulnerability would involve Knight Brawlers only indirectly, we do not wish our proposed users to submit to these constraints against their will. Therefore, we desire to keep compatibility with as many security protocols available to the end users as possible.

Summary of Encryption Technologies: The designers of the Knight Brawlers project do not wish to compromise the integrity of the user's internet home security so they will likely go with the safest WPA 2 technology unless hardware limitations become a significant factor.

5 MCU Design Summary of Hardware & Software

5.1 Design Integration

Once the parts and features for the project have been determined they will need to be integrated into a design that can be programmed on to the custom PCB. Building a schematic design for this step will help us route all of our connections properly. Figure 5.1 is a diagram of what will need to be connected to our microcontroller in our schematic design.

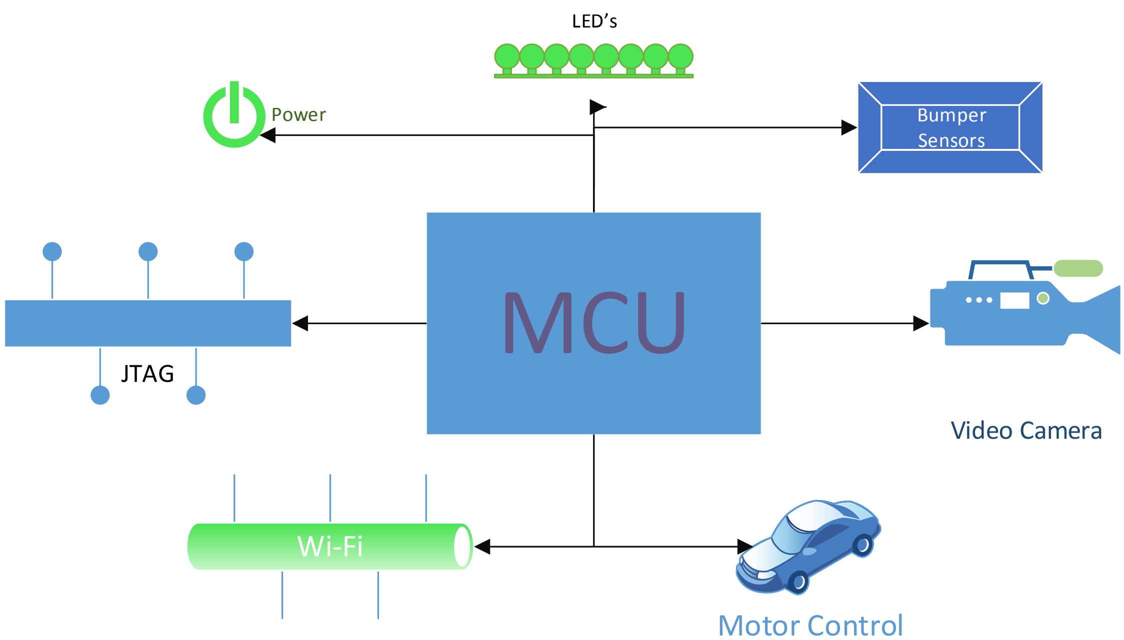


Figure 5.1 Diagram of Components for Schematic Design

5.2 Power Design Integration

The MCU will be powered by a 3.3V power supply. It will be necessary to connect a voltage regulator to step down the battery supply (figure 5.2.1). Also, the power supply must decoupled with a filter ceramic capacitor for every input pin. They must be as close to the processor as possible. Figure 5.2.2 shows the power schematic for the STM32407VGT6 processor from the discovery kit user manual. It's stated in the user manual that even if you are not using the analog part of the chip it must still be connected to the power supply. If it were to be left unconnected then the chip could behave erratically. All other voltage signals must be grounded or connected to the power supply as well.

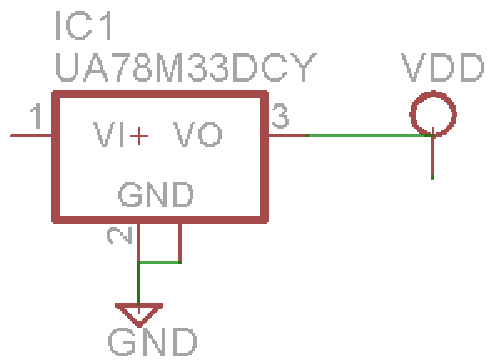


Figure 5.2.1Eagle Schematic of Voltage Regulator (Battery to VDD of MCU)

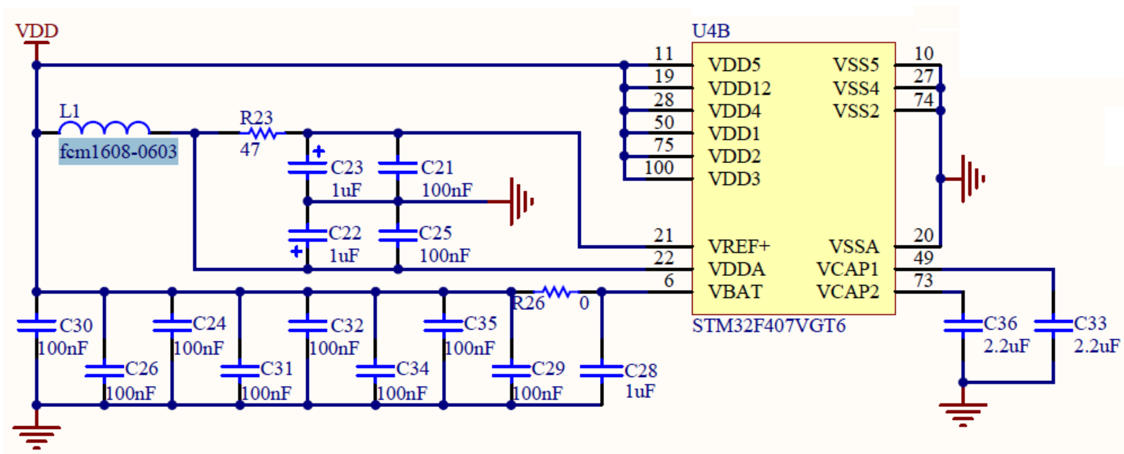


Figure 5.2.2 Power Supply for Discovery STM32F4
(Reprinted with permission from STMicroelectronics.)

5.3 LED Network Integration

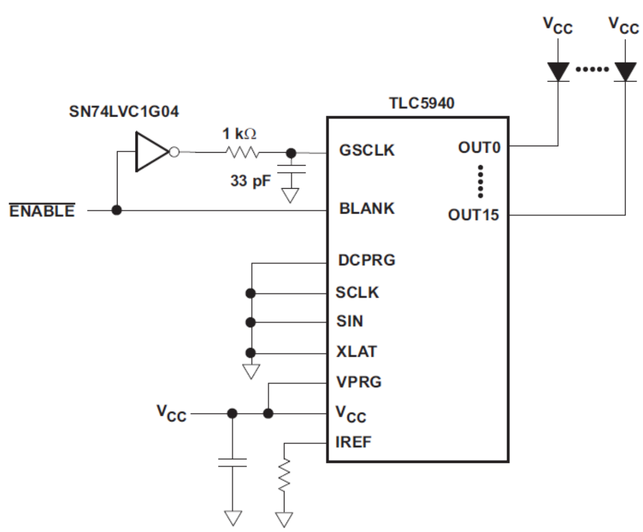
The LED network will be responsible for displaying health status of the car during the game. Different colors will illuminate to display the health status of the vehicle. The MCU will be responsible for determining which LED color to turn on through software. The Schematic in figure 5.3 shows the layout of the LED network for this project. The LED's will be connected via an LED driver to eliminate the need for extra resistors and extra wires in our schematic. The LED driver will help deliver a constant current to each LED connected to its channels.

The LED driver for this project will be the TLC5940 by Texas Instruments. It's a 16 channel, constant current sink LED driver. All 16 outputs of the Driver can be controlled via the TLC5940. Budget permitting and space on the RC car will determine the number of LED's for this project. The LED chosen will be a RGB LED. For each anode lead of the RGB LED there will be an LED driver controlling the on and off state of the LED. A total of three drivers for each color LED. Out of each driver there will be a max of 16 LED's connected.

The MCU will control the status of the LED drivers. When the MCU outputs high to the Enable line of the driver all the OUTn pins of the driver will be forced off regardless of the logic programmed to the device. When the Enable line is to set to low OUTn channels will be set to on. The GSCLK signal must be delayed which is why the RC circuit is implemented. The current for each LED will be determined by the resistor connected to Iref.

The LED driver has EEPROM for dot correction, it will be left as default allowing for 100% brightness. The Cathode end of the LED will be connected to ground. To determine connectivity of the car and the ANDROID phone there will be another LED to display this status. If it's off then that means it is not connected. If it is blinking that means that it is trying to connect. Finally, if it is on that means it is connected. The MCU will control the status of this LED from one of its GPIO (PE15).

The Power LED will display when unit is on (pin60/PD13). Below is the schematic design for the LED network (figure 5.3). The figure on the left shows the schematic for the TLC5940 LED driver and the figure on the right shows the Eagle schematic for the Status LED and Power LED. It is possible to turn on different LEDs using the driver. This will require testing on the breadboard to get the correct outputs. For our first schematic, the one wire approach will be drawn out, but the other implementation will be tested and the schematic could then change.



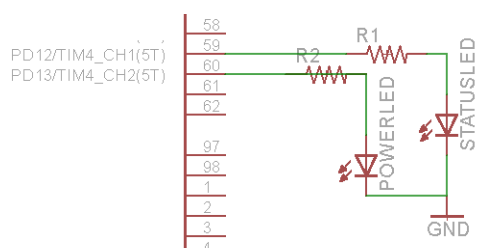


Figure 5.3 LED network
(Figure at Left Reprinted with permission from Texas Instruments.)

5.4 Serial Wire JTAG Debug Port Integration

The serial wire JTAG debug port is a feature that is available on our development board, the STM32F4Discovery. The development board also includes an ST-Link/V2 debugger interface available by STMicroelectronics. The processor for Knight Brawlers will need these debugging solutions implemented on its PCB.

This will allow us to effectively debug any issues we may encounter with our project. On the ARM32F4VGT6 the serial wire debug (SWD) and JTAG are combined on the same on the same port. This will enable the port to either debug through JTAG or SWD.

Debugging is performed only using two pins of the STM32F4. JTAG TMS is shared with SWD pin SWDIO (processor pin PA13) and JTAG TCK pin is shared with SWCLK (processor pin PA14). In order to switch between the two debuggers a specific sequence must be applied by the link on the TMS pin. This will allow for the debugger to be switched from JTAG-DP and SWD-DP. This is a neat feature that will allow two debug port options for our processor. A connector will be necessary to implement for this function.

Depending on the final PCB design for this project we may want to implement an extra connector on our PCB for an ST-LINK/V2 debugging connector. The ST-LINK will provide an in circuit debugger to our design. This will be easy to implement since it shares the same lines for our JTAG/SWD port (SWCLK and SWDIO). It uses these lines to debug the on board circuitry. Schematics from the DiscoverySTM32F4 for these debuggers are shown below in figure 5.4. The additional pins needed for configuration are NRST (internal pull up resistor) and PB3.

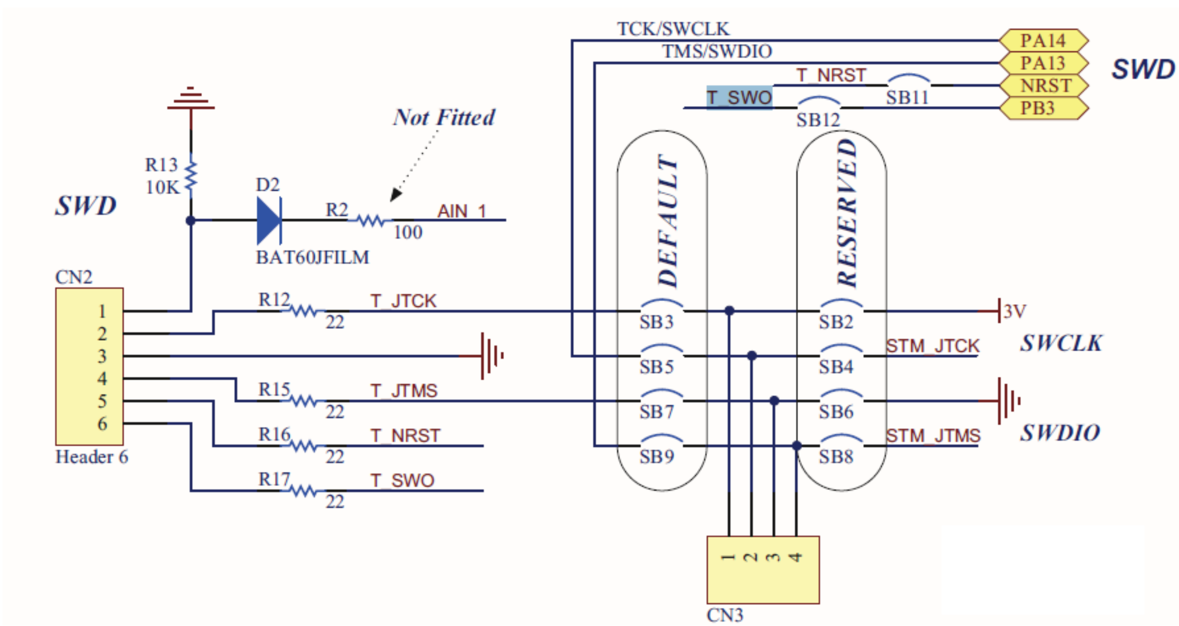


Figure 5.4 Discovery Schematic for JTAG/SWD PORT & ST/Link
(Reprinted with permission from STMicroelectronics.)

5.5 Bumper Sensors Integration

The bumper sensor connected to our design will tell the MCU when the RC car has been hit. There will be three bumper sensors connected to the vehicle (left side, right side, back bumper). These sensors will be implemented using a SPDT

Switch. The schematic design of the switch is relatively simple. One terminal of the switch will be connected to ground and the other will be connected to a GPIO line of the MCU. Figure 5.5.2 shows the schematic for the bumper switch. The MCU will use an interrupt function to tell the processor when the switch has been pressed. Almost every GPIO for the STM32 has interrupt capability.

The code will require to enable the External Interrupt/Event Controller (EXTI). The EXTI can handle interrupts based on priority. The EXTI can detect a short pulse connected to one of its lines. Therefore, it is not required for a huge voltage reading from the switch. PA0 will be configured to handle this interrupt and Pin 99 (Vss_4) is our ground connection. The MCU will be required to handle this interrupt to keep a counter of how many times the RC car has been hit and to turn on the respective LED. Figure 5.2.1 shows a state diagram to show how the interrupt function will be handled.

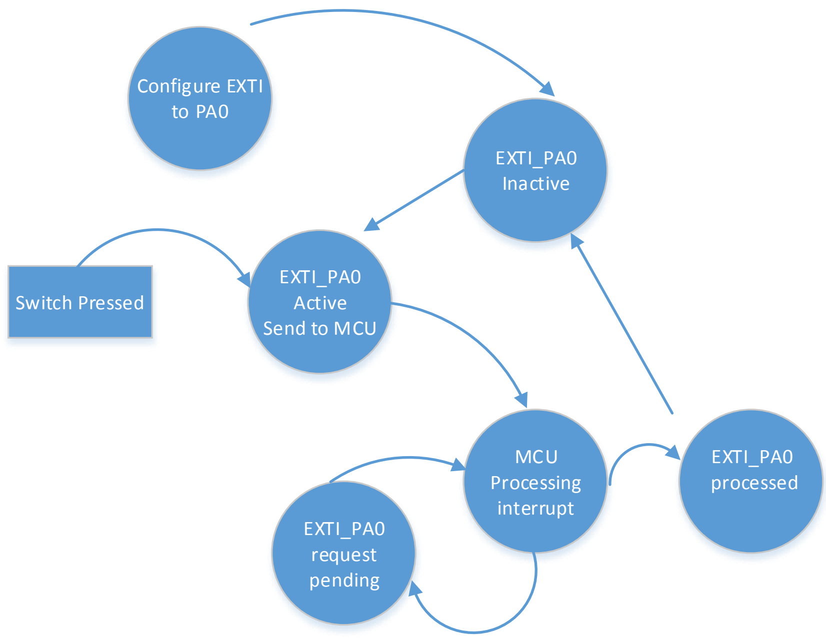


Figure 5.2.1 EXTI feature for Knight Brawlers

There are two ways the switch can be set up. For this project the switch will be set to create a voltage pulse when it has been pressed. Therefore, the NC and C terminals of the switch are used. All of the bumper switches can be connected to one EXTI line. They will be connected in parallel. Since they are connected in parallel, at any time one of the switches can generate a pulse to EXTI line which will then have the MCU service the interrupt. The switch will not be on the PCB its self it will be connected via wire.

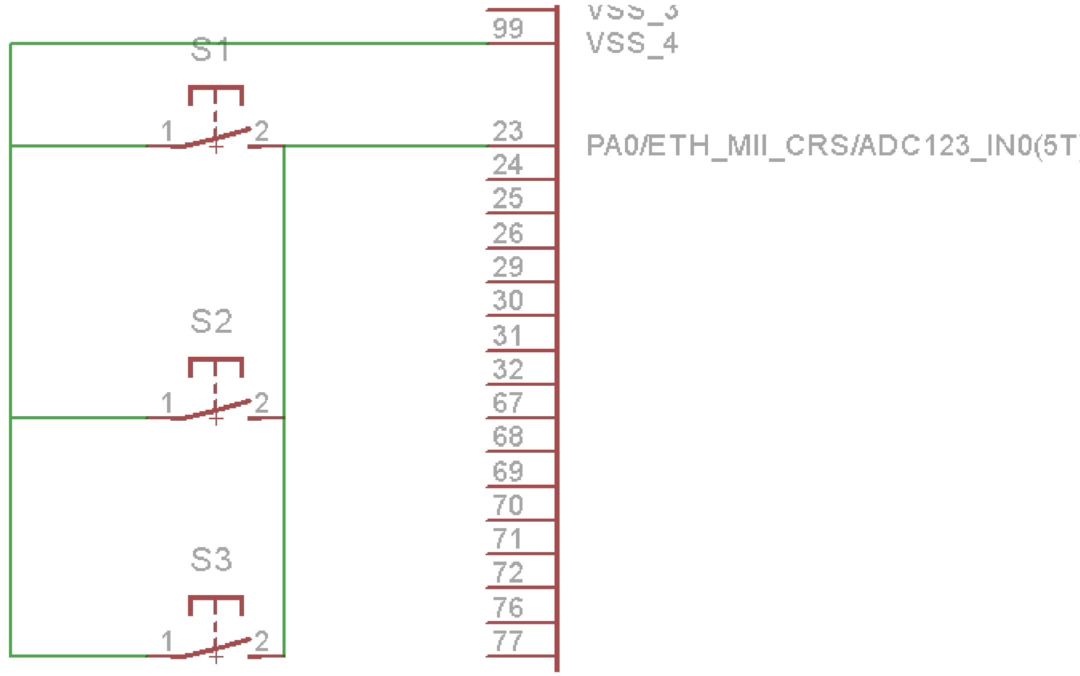


Figure 5.5.2 Bumper Sensor Switches

5.6 Motor Control Integration

One of the main reasons that the STM32F4VGT6 processor was chosen was its PWM outputs. We will use these PWM outputs to control the motors for Knight Brawlers. PWM's will allow the MCU to control the motors for this project by adjusting the duty cycle of a PWM to control the amount of energy the motor receives. This will us to control the steering and acceleration of the RC car. The PWM output of the MCU will be fed to an H bridge driver that will correlate the PWM input into motor controls for the RC car.

The STM32F4VGT6 uses the advanced-control timers TIM1, TIM2, TIM3, TIM4, TIM5, and TIM8 for PWM output. TIM1 and TIM8 timers have full modulation capability so we will be able to use 0-100% duty cycle for our motor control. The PWM signals have multiple channels that can send the same signal. For this project there will be three motors that need to receive PWM signals. One of the motors will receive the PWM signal from TIM1's and TIM8's PWM channels. This motor will be the driving motor so it will need the full modulation capability of TIM1 and TIM8 timers.

The next two motors will receive a PWM signal from TIM2 and TIM3. TIM2 and TIM3 have less than full modulation, but this will be okay because the steering does not need that much of a turning radius. Since it is only one motor per direction (left and right), it will only require one channel of the PWM on TIM2 and one channel of the PWM on TIM3. The PWM signals will be sent to our DBR8837 H-Bridge motor driver from Texas Instruments. Our H-bridge motor can control 1 DC motor at a time so one will be needed to control our driving motor for this project. One H-Bridge driver will be necessary for the forward, reverse, stop, and coast functions of the motor. See figure 5.6.1 and table 5.6.1 for their operations.

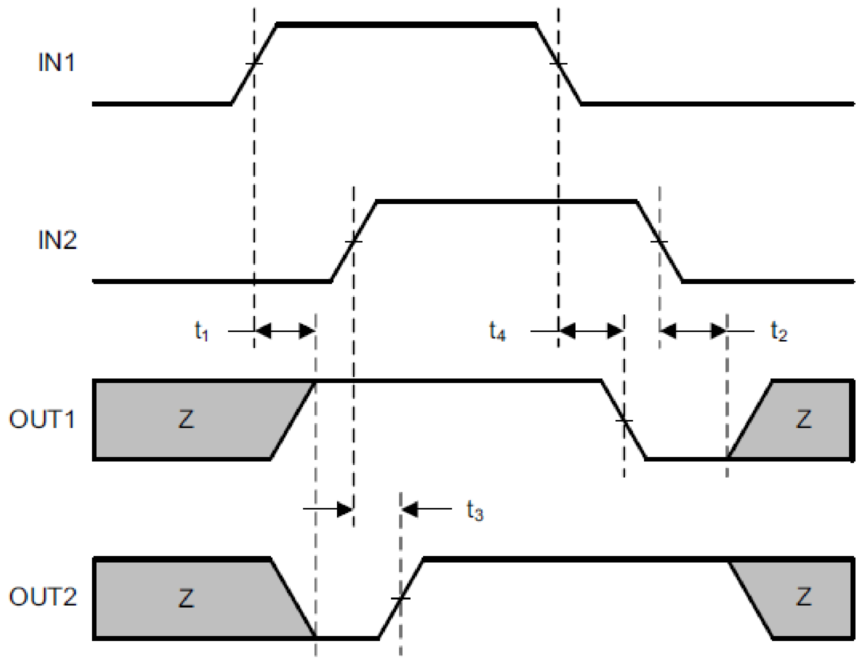


Figure 5.6.1 Timing Diagram of In/Out Motor Function
(Courtesy of Texas Instruments)

IN1	IN2	OUT1	OUT2	Function (DC Motor)
0	0	Z	Z	Coast
0	1	L	H	Reverse
1	0	H	L	Forward
1	1	L	L	Brake

Table 5.6.1 DVR8837 Logic (Courtesy of Texas Instruments)

We will take the two PWM outputs from our STM32F4 processor and use them as IN1 and IN2 for the DVR8837 H-Bridge Driver. The function of the motor is then determined on the high and low states of the PWM input. OUT1 and OUT2 are connected the DC motor. Because of the full modulation capabilities of TIM1 and TIM8 we will be able to control the reverse, forward, and brake functions of the motor very accurately.

TIM1 will provide the PWM input from TIM1_CH1 (PE9 pin) for IN1 of the H-Bridge motor1. TIM8 will provide the PWM input from TIM8_CH1 (PC6) for IN2 of the H-Bridge motor1. The four other pins of the driver are for motor voltage VM, VCC, ground and sleep. The sleep and VCC pins of the driver will be connected together to make sure that the driver does not enter a low power sleep mode. Figure 5.6.2 shows the schematic design for our MCU and H-bridge driver.

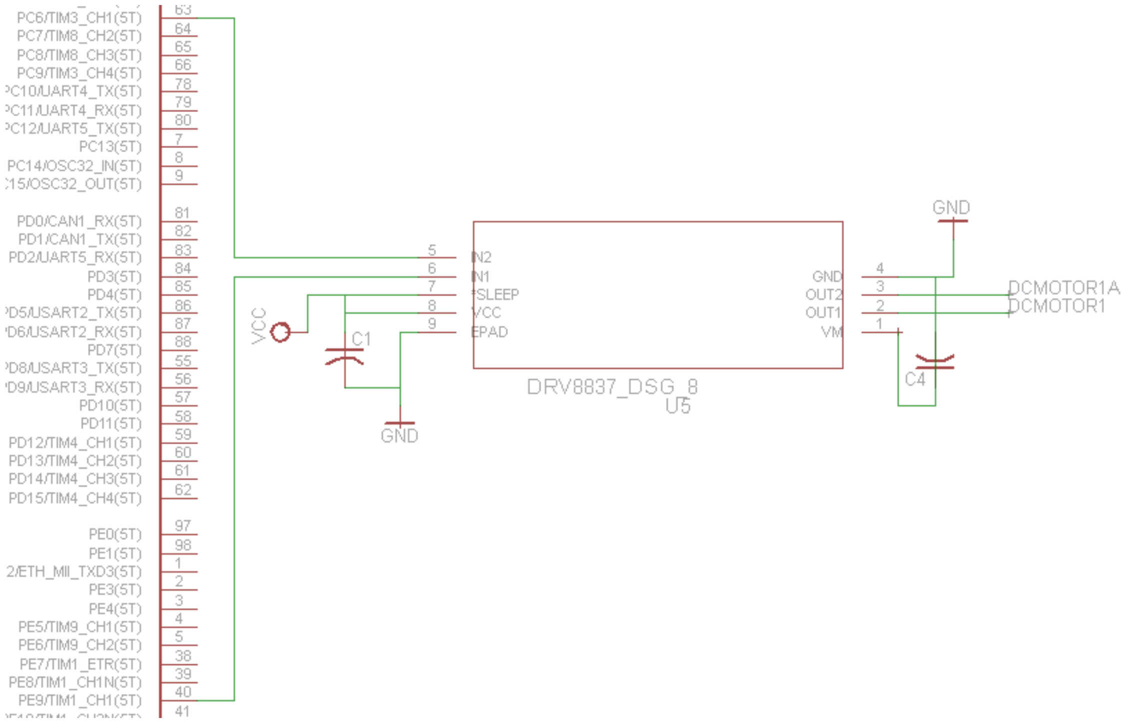


Figure 5.6.2 Motor1 and Motor2 H-Bridge Schematic

The next two motors that need to be integrated in our design will be the DC motors that will drive the steering function for this project. One of the DC motors will be reserved for left steer and the second DC motor will be reserved for right steer. During the prototype stage it will be necessary to see if we can use one motor for both steering directions, but as of now our design will have a DC motor for each function. DC motor2 and DC motor3 will use TIM2 and TIM3 of the STM32F4 processor. They will use the PWM function available from TIM2 and TIM3 to determine how much to steer in which direction.

Depending on the duty cycle of TIM2 will determine how sharp to turn right and depending on the duty cycle of TIM3 will determine how sharp to turn left. If the motors are not receiving any kind of PWM signal from the MCU then they will be on coast mode. Meaning the vehicle will steer straight. See table 5.6.2 for steering and operations. TIM2_ch2 and TIM3_ch1 are available on pins PB3 and PB4 respectively. They will be connected directly to Motor2 and Motor3 respectively.

5.7 Camera Integration

The Camera component will be an important feature to the android application of this project. The STM32F407VGT6 can embed a camera interface that can connect through 8 bit to 14 pit bit parallel interface to receive video data. Therefore, connecting our camera will be simply configuring the ports already available on the MCU to the pins of our camera. Under consideration for camera options are the OV7725 and the OV9655. They both have similar pin configurations to the other so their schematics will be somewhat identical. The schematic that will be drawn out is the OV9655 camera. The STM32F4 has 14 data bits for the DCMI interface.

The camera we are using has an 8 bit data (D2-D9) output which will be configured to our DCMI data lines DCMI_D1-D4 and DCMI_D6-D9 (DCMI_D0 pin is already being used for TIM_ch1).Data bits for the camera that will be used are D2-D9 and they will be connected to pins PC7-12 and PE5-6 respectively. The other data lines of the OV9655 will not be used, they for RGB data. The other remaining connections to the OV9655 are power, sync, reset, and clock to the MCU. Table 5.7 explains the other OV9655 connections to the STM32F4 pins.

OV9655 Pin	STM32F4VGT6	Connection type
C5	PA4	HREF output of camera connects to DCMI_HSYNC of MCU
B5	PB7	Vertical sync output
F3	PA6	Pixel Clock connection
A1	PD6	Power enable 0=normal operation,1= power down
B1	PD12	Reset,1=normal operation,0= reset
B4	PB9	SCCB serial interface data I/O.
A4	PB8	SCCB serial interface clock input.

Table 5.7 Camera to MCU Data Connections.

The remaining connections of the OV9655 pins are for powering the camera, grounds, and a system clock for the camera. The system clock for the camera runs at 24 MHz, it will input to the pin E2. The power supply for the core, DVDD (pin F4), is at 1.8 V, I/O and analog power supply, SIO_D and AVDD (pins B4 and B3) are at 3 V. Luckily, these connections are already made on the camera’s chip minus the system clock and power supply. In order to connect our camera to PCB will need jumper wires so that camera can be placed anywhere on the car. So, the PCB will need a Female header to implement this design rule. The header will contain the 18 pins necessary to connect the camera to the MCU on the PCB. Figure 5.7 illustrates the camera schematic design for this project.

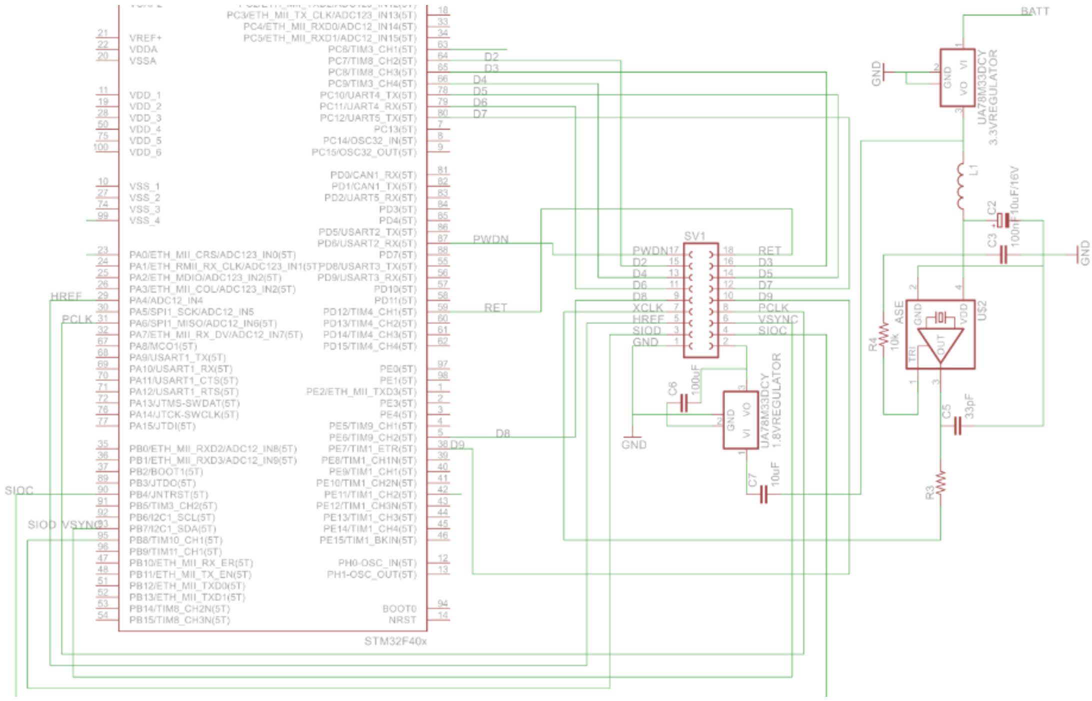


Figure 5.7 Camera Schematic

5.8 Wi-Fi Integration

To establish connectivity between the stm32F407VGT6 and the Android phone a Wi-Fi link will be established to exchange information. The data exchanged will include phone accelerometer data, number of hits taken, and video feed. This group will use the TI SimpleLink™ CC3000 Module – Wi-Fi 802.11b/g Network Processor from Texas Instruments. This network processor is intended to create a self-contained wireless network that is a simple solution for embedded applications.

It connects to the MCU through a 5 wire serial peripheral interface. The MCU will act as the master and the CC3000 will act as the slave. The five SPI lines are SPI_CLK, SPI_CS, SPI_DIN, SPI_IRQ, and SPI_DOUT. SPI_CLK is the clock line from the STM32F4 to the CC3000. SPI_NSS/SPI_CS is the chip select signal from host to slave. This signal indicates that the master wants to communicate with the slave. SPI_DIN/SPI_MISO sends the data from the host to slave. SPI_DOUT/SPI_MOSI sends data from the slave to host. Finally, SPI_IRQ is an interrupt line. When SPI_IRQ is low and SPI_CS is deasserted the CC3000 module is ready to receive data. When only SPI_IRQ is low the CC3000 sends data to the MCU.

The rest of the 46 pin package for the CC3000 include several ground, RF antenna, power supply, and different mode pins. Some of the mode connections are not necessary so they will be grounded. The STM32F4 features up to three SPIs. SPI1 can communicate up to 42 Mbits/s. SPI2 and SPI3 can communicate up to 21 Mbits/s. We will be using SPI1 to ensure a faster data rate transfer. The table below lists the pin connections from the MCU to the CC3000 and the SPI type (Table 5.8).

STM32F407VGT6	CC3000	Pin Connection type
PA5	17	SPI_CLK
PA7	13	SPI_MOSI
PA15	12	SPI_CS
PB4	15	SPI_MISO

Table 5.8

The final two pins that must be attached to the MCU is the SPI_IRQ interrupt line of the CC3000 and the VBAT_SW_EN module enable line that will be connected to GPIO line PA9. This line needs to be connected to a GPIO line of the STM32F4. The GPIO line needs to be configured as a digital input with an internal pull up resistor. The internal pull up resistor will allow SPI_IRQ to go low when the PWR_EN connection of the CC3000 goes high. Essentially, PWR_EN acts a switch that shorts the SPI_IRQ connection when it is enabled. Any of the GPIO pins available to the STM32F4 can be enabled as digital input with an internal pull-up resistor. We will use the pin PB12 of the STM32F407VGT6 to connect to the SPI_IRQ line of the CC3000. Figure 5.8.1 shows our schematic design. VBAT_SW_EN is a module enable line connected to GPIO line PA9. Figure 5.8.2 shows the schematic design for the Knight Brawlers Project.

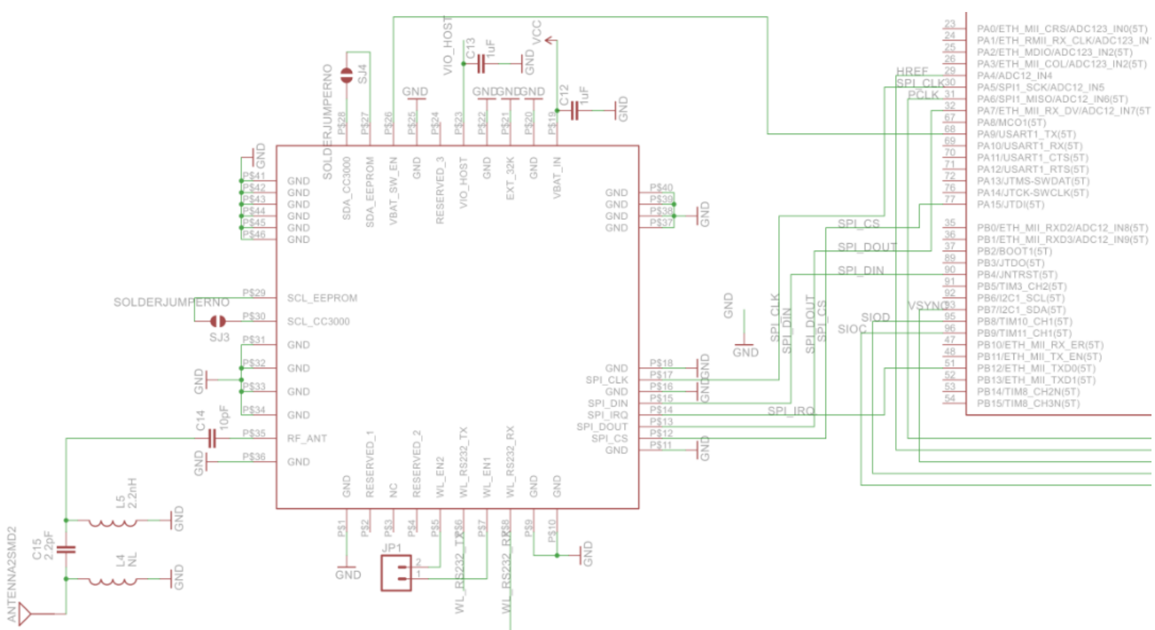


Figure 5.8.1 CC3000 Wi-Fi schematic

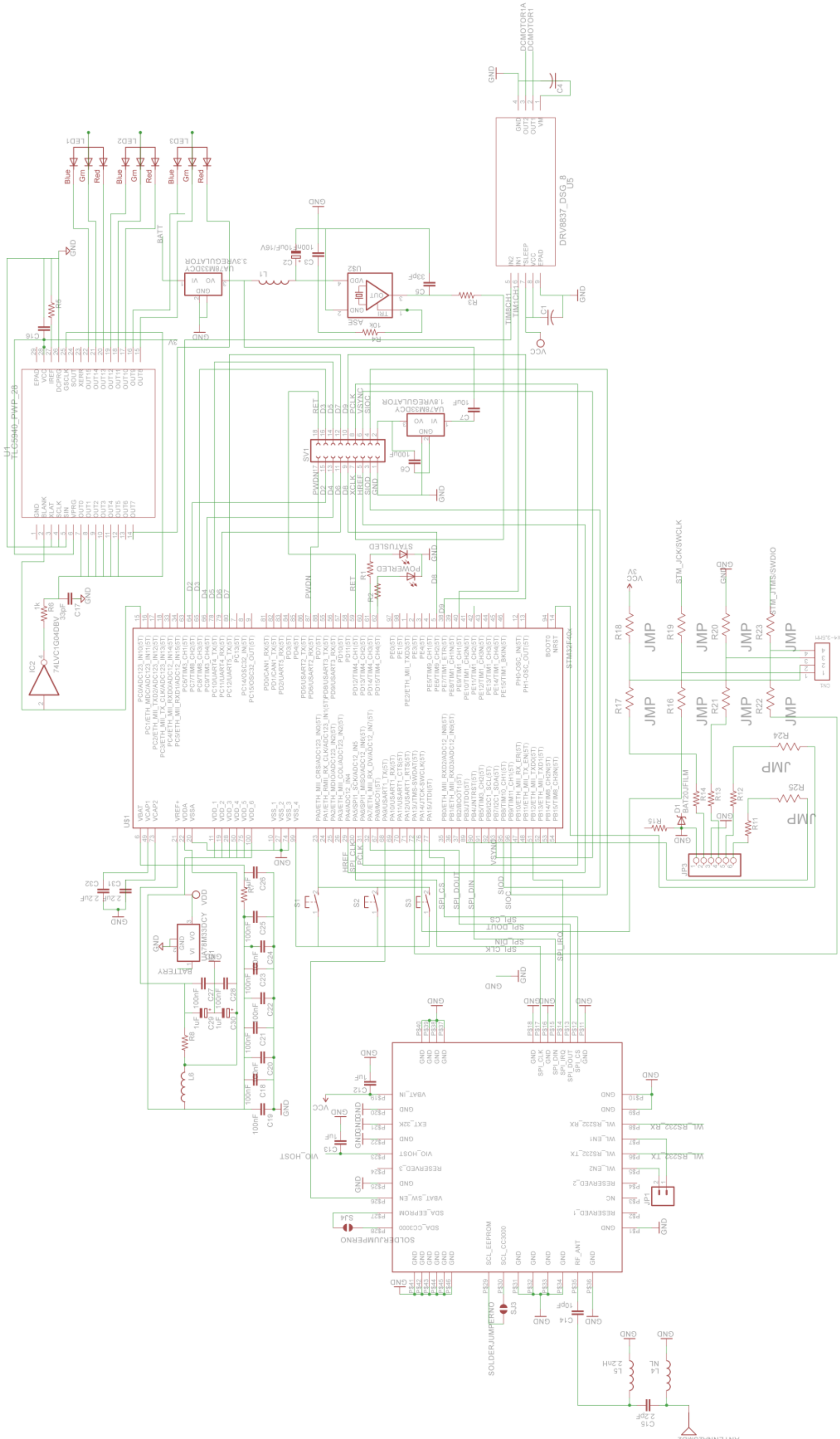


Figure 5.8.2 Knight Brawlers Schematic Design

6 Project Prototyping Construction & Coding

6.1 Parts Acquisition and BOM

The parts required for this project will have to be ordered as they are selected to ensure that they will arrive within due time for this project to be completed successfully. This section lists the part name, vendor, description, and cost. They are placed in table 6.1 below. There are four cars, budget permitting, to be constructed. The final cost must be multiplied by four to get an estimate of the actual cost. Since this project does not have any sponsors it will be important to cut costs as much as possible. How many cars are built depends on the sum of the table below. Resistors and capacitors are not included in the BOM due to their really low price and ease of availability.

Name/Part#	**Manufacturer/Vendor**	**Unit Cost**
Voltage Regulator/UA78M33	*Texas Instruments*	*Free*
Description: The UA78M33 is a three terminal positive voltage regulator. Its output is 3.3 V. Max input is 25 V, and min input is 5.3 V. Will regulate voltage for this project.		
RGB 5mm LED	*EBay*	*100@13.85*
Description: 5mm common cathode RGB LEDs. Will provided aesthetics for Knight Brawlers' project. Display blue, true green and red.		
SPDT Switch	*Zippy/RadioShack*	*3@$10.47*
Description: The SPDT switch will determine when the RC car is impacted by another car.		
STM32F4Discovery	*STMicroelectronics*	*$14.25*
Description: The STM32F4 is the development Board that will be used to prototype this project. It is necessary to test the wireless connection, LED network, bumper sensors, and camera feature of this project. It contains the STM32F407VGT6 ARM Cortex M4-F processor.		
STM32F407VGT6	*STMicroelectronics*	*Free*
Description: The STM32F407VGT6 is the processor for this project. It is a 168 MHz, 32 bit, 192 KB memory ARM Cortex M4F CPU. It will be our final design's processor on our custom PCB board. It will be in charge of interfacing our mobile device with our RC car. It will be in charge of controlling our RC car based on accelerometer signals sent via Android, sending video to the Android device, and		

operating the LED network. This processor is free to sample on STMicroelectronics' website.		
Knight Brawlers PCB	*4PCB.com*	*$40.00*
Description: The Knight Brawlers PCB will be the custom PCB designed for the Knight Brawlers project. It will be a full spec two layer board from 4PCB.com. It will be a blank board that will have its components mounted to it. It will be designed in Eagle and then have its Gerber files sent to 4PCB.com for manufacturing. Its cost is reduced by a student discounted specialty available by 4PCB.		
LED Driver/ TLC5940	*Texas Instruments*	*Free*
Description: The TLC5940 LED driver will provide stable power to LED network for this project. It will be used to configure the array of LED's needed for aesthetics purposes of this project. It has 16 channels, providing current sync for up to 16 LED's with only one pin connected to the MCU. Free samples are able to be obtained.		
ST-LINK/V2	*STMicroelectronics*	*$20.82*
Description: The ST-Link/V2 is an in circuit debugger that will be necessary for our PCB design. It has JTAG and Serial Wire Debugging interfaces. It communicates with the compiler to debug the circuitry on chip. It will connect to the board via connectors on the board.		
USB to TTL Serial Cable	*Adafruit*	*$9.95*
Description: The USB to TTl Serial cable will allow for the board to be powered while it connecting to the login/debug console of the processor. It will allow this program to be debugged via Serial Wire Debug.		
DRV8837	*Texas Instruments*	*(3) Free*
Description: This is the H-Bridge Driver for this project's motors. It can control 1 DC motor at a time. It inputs two PWM signals to generate motor functions. Motor functions include forward, reverse, brake and coast operations. It has a low power mode and comes in an 8 pin package.		
RC Car	*New Bright*	*$25.99*
Description: This is the RC car for Knight Brawlers. It is 1:16 scale. It has three DC motors. Two for drive and the other for steering. The motors will be detached from current MCU and will be replaced with a custom PCB that will operate the vehicle.		
Camera/ OV9665	*Ominivison*	*$10.00*
Description: The OV9965 is a 1.3 Megapixel camera digital camera that will enable this project to have stream video to its Android interface. Its 28 pin		

package conveniently connects to the DCMI pins available on the STM32F407VGT6. It needs only 8 data bits for video. It has a ¼'' frame that can perfectly fit on the RC car for this project. Its max transfer rate is 30 frames per second.		
1.5V/9V Batteries	*Duracell*	*$30.00*
Description: Rechargeable 1.5V and 9V NiMH batteries. This will be the power source for this project.		
TI SimpleLink™ CC3000 Module	*Texas Instruments*	*Free*
Description: A self-contained wireless network processor that will add Wi-Fi connectivity to our project. It's low cost, simple solution, wireless communication processor solution for embedded devices. It runs IEEE 802.11 b/g. It will be responsible for transmitting data from the MCU unit to the Android phone. It will transmit video, hits taken and receive accelerometer values from the Android phone		

Table 6.1

6.2.0 PCB Software

Once the processor is chosen for Knight Brawlers and the necessary parts for the project are determined, the next step will be to take our project from the prototype/development stage to actually building the project itself. There are several of choices when it comes to printed circuit board vendors and the software needed to develop the PCB. Before sending the project to the PCB vendor, it will be developed using software tools. There are two possible software choices for constructing the PCB for this project. PCB artist and Cadsoft's Eagle software are the two software candidate tools. Once the necessary files are developed on either one of the previous mentioned software choices, it will then be send to a PCB vendor. While there are plenty of PCB vendors available, this project will use services of 4PCB.com.

6.2.1 PCB Artist

PCB artist is available through the PCB vendor supply 4PCB.com. It's available free of charge and its downloaded straight from the 4PCB website. It has great documentation including videos and step by step guides to help the user get started on their PCB designs. PCB artist has all the tools that allows the user to create a schematic then convert their schematic design to a PCB. PCB artist integrates the extensive library of components to schematic and PCB design. Once the user has all the correct components properly drawn out in the schematic, PCB artist under its tools menu takes care of the rest. It automatically forwards the schematic design in to your PCB layout. There are a few times were the user may have to manually route connections and add copper wires to the PCB design, but the software takes care of locating the components in the PCB layout and places them on the board for you. This saves a good deal of time for repeat edits to the schematic design.

While in the PCB menu, the software allows you to choose the number of layers for your board. That way you can set up multiple wires on top of each other through different layers. Thus, the user can save a lot of space and keep the design tight and compact. The user can also edit the size of the board to meet their project specifications. After the PCB design has been finished the user can the use the design rule check function of PCB artist to check for errors. The design rule check helps eliminate errors due to spacing issues, manufacturing issues, and nets on the board. After the software has run its design rule check it highlights and displays the errors on the PCB design. The user can then check how to solve those errors or ignore them and continue with the design. Once the final design is completed the software allows the user to place their order within the software. PCB artist would be a great PCB software choice for Knight Brawlers due to its extensive component library, user friendly layout, customizable options, and its schematic to PCB feature.

6.2.2 Eagle

Eagle is an electronic design editor with schematic capture and PCB design tools. Much like PCB Artist, it has an extensive library of components that will serve this project well. Eagle will allow this project to insert its own components and libraries if they are not already available in the programs library. Eagle will also allow this project to transfer its schematic diagrams into PCB layouts and has built in functions to auto route connections in the PCB layout like the ones in the schematic design constructed. Eagle allows the user to keep track of a build of materials while constructing a project. This will be handy to keep track of the budget for Knight Brawlers. Eagle also like PCB artist allows the user to use multi-layer PCB layouts for increased area to work with for connections. Eagle also has built in functions to debug possible errors for this projects PCB design. The CAM processor from eagle will generate the necessary Gerber files for our PCB design.

6.2.3 PCB software conclusion

PCB Artist and Eagle both can provide this project with the necessary software tools to develop our circuit board efficiently. PCB Artist is a lot simpler to manage than Eagle as far as the layout of the software, locating components, debugging errors, and building custom boards. Not to mention PCB Artist is also free. However, Eagle is a more professional software. It would be a great benefit to the members of this project to gain knowledge and experience in a program such as Eagle. It would be great resume builder. Eagle is available at the labs where this project will be built and also has a trial version that will be able to fit the needs for this project. Therefore, Cadsoft’s Eagle Software will be the PCB and schematic design software of choice for Knight Brawlers.

6.2.4 PCB Assembly

Now that this project has chosen a software to develop its schematic and PCB designs, the next step will be to draw out the schematic layout for the processor. In order to keep the design for the schematic and PCB simple, some of the features from our development board that feature our processor will need to be eliminated. After those features are determined worthless to this project the schematic design for the processor is developed in Eagle. Once the schematic design is in Eagle, we will add our own feature to the project. This step will be completed once prototyping on our development board is complete.

6.3. PCB Vendor

The PCB vendor of choice for this project is 4PCB.com. 4PCB.com will allow this project to use its discounted student program to keep the PCB cheap and under budget. They allow students to use their PCB services and software to generate multilayer PCBs. Although this project will use Eagle as its PCB software design, it will use the services of 4PCB.com to generate the PCB. 4PCB.com has their own software available. Their student discount starts at 33 dollars for a two layer board and 66 dollars for a four layer board. This will be of a great benefit for Knight Brawlers to keep the board design nice and compact. The developers of Knight Brawlers will send their Gerber files of this project to 4PCB where they will construct the PCB design for this project. Our PCB will then be mounted with the necessary components for this project.

6.4 Final Coding Plan

In order to make sure that Knight Brawlers will work flawlessly, the code written for this project will need to be tested time and time again to allow for debugging. There are two main levels of code for this project. The top level coding will be handled by the android phone’s processor. This is where the code for the user interface/app will be. The user’s input will be taken from here and translated through the phone and sent to the blank processor on the RC car. The code written for the android phone must make the app look clean and interactive for the user. The code will be responsible for sharing and communicating information from the Knight Brawlers’ processor.

The low level coding will be done on the Knight Brawlers’ blank processor. The code on this processor will be responsible for receiving accelerometer values from the android phone. It will also need to be able to communicate and share hit values for the bumper function of Knight Brawlers to the android interface. The code written will need to control the motor function, LED network, sensor network and video functions for this project. It is imperative to have this code written and working before the PCB is built and connected to our device. Mostly the code for both levels will need to be done in the prototyping stage on the development board. This will ensure that the code will work on the final design.

An important step to achieve a good coding plan will be to make sure that our code is appropriately commented so that every function, call, subroutine's function be documented. This will help in the debugging stage to isolate problems between the two levels of coding. Functions, subroutines and libraries will keep the functions for the project in a manageable form. The code for Knight Brawlers will be split up into different sections (see figure 6.3). These sections will be responsible for a certain function as noted in figure 6.3. They will be tackled one by one to ensure that they are working appropriately.

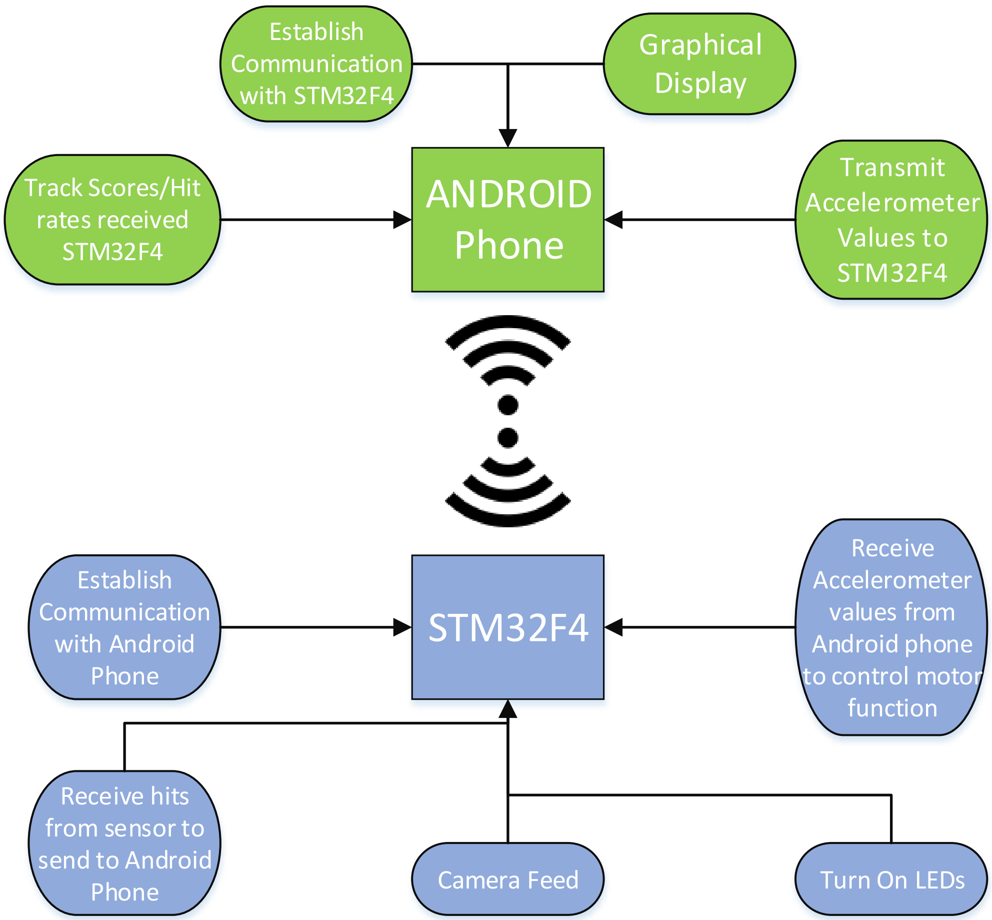


Figure 6.3 Code Sections

The first step to create a successful coding plan will be to tackle each one of the I/O devices for the STM32F4. That way, the bugs for basic operation of those devices can be worked out before they are synced with the Android phone. The STM32F4 will need to be able to receive and save information from its bumper sensors. It will need to be able to turn on the connected LED's based on that information. The controller must also be able to control the motors. Once it is confirmed that these functions are operable, we will proceed to establishing communication with the STM32F4 and the Android device.

Once the link is established, the link will be tested with the bumper sensor information. The Android device will need to be able to read the bumper sensor values from the STM32F4 and keep a score. Then the link will be tested to receive and transmit accelerometer values from the Android phone to the STM32F4. The STM32F4 will then use these values to control the motor functions. Finally, the link will be tested for video transmission.

The STM32F4 will need to transmit video feed from the camera configured to it. The graphical interface will be coded during testing of the each section. Breaking up the code into sections and testing for operability in a piecemeal fashion will ensure that the final code will be working through each step of the procedure. This will ensure project stability. Now that the project has functional code it can go to the schematic/PCB design phase.

7 Project Prototype Testing

Prototype testing will be done in several environments and will include vehicle tests, hardware tests, and software tests. The successful testing of our prototype will allow us to move from different phases in our project. From breadboard and development board phase to custom PCB testing phase and then to actual environment tests. The environment tests will include taking the project outside the lab to test around campus. This will ensure that our project will be able to function anywhere. The prototype tests are discussed in this section.

7.1 Vehicle Test Environment

The vehicle test environment will be a very important part in how this Knight Brawler is operated so we are looking for an ideal situation as possible for us to test the Knight Brawler in. The RC Car is relatively large and will need to have a large enough area that will fit up to for RC Cars to play the different games implemented into the systems. The area will be in a closed off 12 feet by 12 feet section that will be have a 6 inch wall which allow to Knight Brawler vehicle not to leave the area designed for the RC vehicle to operate in. The section which we will be called Knight City (Fig. 7.1) which will have several different objects and obstacle in the way of the RC Vehicle that allows the user to play in solo or multiple player mode which will increase the enjoyable of user.

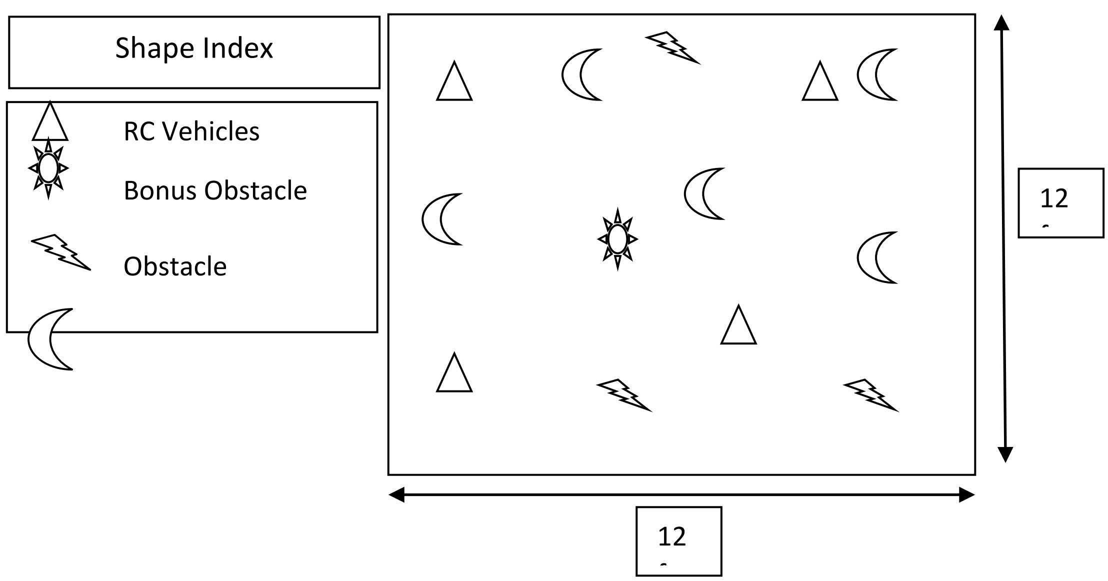


Figure 1 – Knight City Test Environment

7.2 Hardware Test Environment

Hardware testing will include testing the development board with the parts we deem necessary for our project and testing our PCB once it has been mounted with the parts necessary. Our development board will be jumped to a breadboard where different components of the project are tested. The environment for this testing will be the senior design lab or at home. Most members of this group have breadboards and testing equipment available at home. Testing of the main hardware components such as the LED drivers, switches, H-Bridge drivers, Camera, and CC3000 wireless network will be done on the breadboard. They will be connected to the development board via jumper wires. We must be able to correctly configure these components. Testing of the LED drivers will ensure that we can light up our LEDs.

Once we have the LEDs tested, we will test the switches of this project to ensure that we can light up our LED’s based on the number of times the switches are pressed. Testing of our H-Bridge drivers will ensure that we can control the motors for this project. Once correctly configured to the development board we will need to test how we can make the drivers speed up and slow down the motors and make them turn left and turn right. The camera will also be tested on the breadboard to assure that we can get video feed streaming. The wireless network will then be tested on the breadboard. It will need to be able to turn on to send and receive data wirelessly.

Once we have confirmed that all of our components can work with our development board, we will need to configure the wireless connection between the phone and the board. The testing of the Android interface can be done in several ways. Most likely, the first test to see if there is a connection will be to read accelerometer data from the phone to the board. Once this data is confirmed we will know that the connection exists between the phone and the board. Once we complete the steps above, we will have the necessary lists of components and connections to the development board that we need to make our PCB. After mounting our PCB, we will test the components mentioned above to assure that every aspect of our PCB is working.

After the PCB with its mounted parts has been confirmed to be working, it will need to be fitted to the vehicle for Knight Brawlers. Once on the vehicle, it will be tested once again. The LEDs will need to be placed strategically on the car to create the best visual display. The Camera will need to be placed on the car where it will not be harmed and so that the view from the camera will create a first person driving experience for the player’s point of view on the Android phone. The motors will then be tested to see if the MCU and Android are controlling the cars appropriately. Temperature will also need to be measured to see if heat sinks are appropriate. The motors and H-Bridge are known to cause undesired heat effects which will be taken care of in each step of the testing process to make sure that this effect will be limited. The switches will also be tested and place so they are able to make easy contact with the bumpers of the other vehicles. Finally, the vehicles will be tested within the members of this project to make sure the hardware is functioning.

7.3 Software Test Environment

7.3.1 MCU software

The STM32F4 microcontroller unit is a complex ARM M4 microprocessor. Everything has to be set up perfectly to interface with external hardware which is where most of the testing and integration complexity is. The STM32 family has available the STM-Studio which is a run-time variables monitoring and visualizing tool which will be used to verify the functionality various integral components that comprise the Knight Brawlers project. The testing procedure will be kept as simple as possible making no major changes until the previous step has passed. It is poor practice to code an entire program without testing each functional component as progress goes forth.

The list of events below describe the testing procedures for the host MCU and its interface with the collision sensors, LED network, camera, and Wi-Fi module.

1. Initialize MCU with blinking light program
2. Write functions that handle sensor inputs
3. The Sensors will be tested in real time with all possible combinations of inputs at different timing intervals. Simulation testing consists of physically pressing the sensors to trigger a valid input.
4. Test each sensor individually ensuring that the registers read correct values when a collision is simulated
5. Make changes as necessary
6. Test sensors in combinations ensuring there is no cross wiring and that the MCU can process multiple data input conditions without going into race state or crashing.
7. Ensure that upon the correct number of sensor inputs as “hits” from other cars will trigger the car to shut down indication the player has lost that round
8. Make changes as necessary
9. Write functions to implement LED network
10. Trigger LEDs with sensor input
11. Verify all combinations of LED outputs from sensor inputs
12. Ensure that LEDs match up during car shut down and not start over
13. Write functions to implement camera
14. Scan camera registers and print out their values to verify connectivity and correct configuration parameters
15. Send ready signal to camera
16. Wait for reply
17. Send frame capture signal to camera
18. DMA will load the frame into predetermined memory address
19. Read the memory address locations and print out to screen in hex to verify functionality

20. Write functions to initialize Wi-Fi module
21. Fist time Wi-Fi config has event done event that indicates first time config has been completed by sending signal back to the MCU
22. Establish event connection and wait for reply from the CC3000
23. Establish connection
24. Wait for asynchronous connection even reply
25. Initialize phase for data transfer
26. MCU reads buffer size
27. CC3000 return its number of buffers and is verified correct
28. Send packet of data to module
29. MCU must decrement the number of buffers if > 0
30. CC3000 returns number of completed packets unsolicited event which must match the number of packets sent

7.3.2 Android Software Testing

In order to ensure the remote control is functioning correctly, it must be tested. Due to the lack of support for accelerometers and Bluetooth communication in the Android development emulator which is the Android computer software that will model a Smartphone so we wouldn't have to implement the software on an actual device, we will have to do some actual testing on an android device. Also, these emulators are made for not very complex applications that don't use very many hardware components. The remote control application must be tested on a physical Android device that has accelerometers and Bluetooth communication. To begin the test and after the software application is created, an Android device that is to be used for the Knight Brawler will receive the remote control application via USB connection from the development computer. Once the application has been downloaded, testing can begin. (Fig. 7.3.2)

First, there will be a tab or section that will allow to begin testing the application, the accelerometers, and the phone in general before being allowed to move on to several other options in the application. This prompt will allow you to determine whether the accelerometers are reading the correct values and if the necessary features are implemented correctly and is working correctly. The user must keep in mind that the values displayed on the remote control are not completely exact. However, the values displayed must be within 1.0 of the recommended testing values. This first prompt will tell the user to determine if the accelerometers are functioning. There will be several different values to set the axis and collaborate the phone to know to it is implemented in a usable way. This step should be mostly automated besides a few requests that will be asked from the user. The user will be allowed to skip this section at their our risk, but skipping will allow the user to come back to this tab or section and redo the test if they aren't happy how the controller is navigating the Knight Brawler vehicle. Next the remote control will be tested for correct communication. The test will begin when opening the application, and observing whether the tester was asked to turn Bluetooth on via pop up notification.

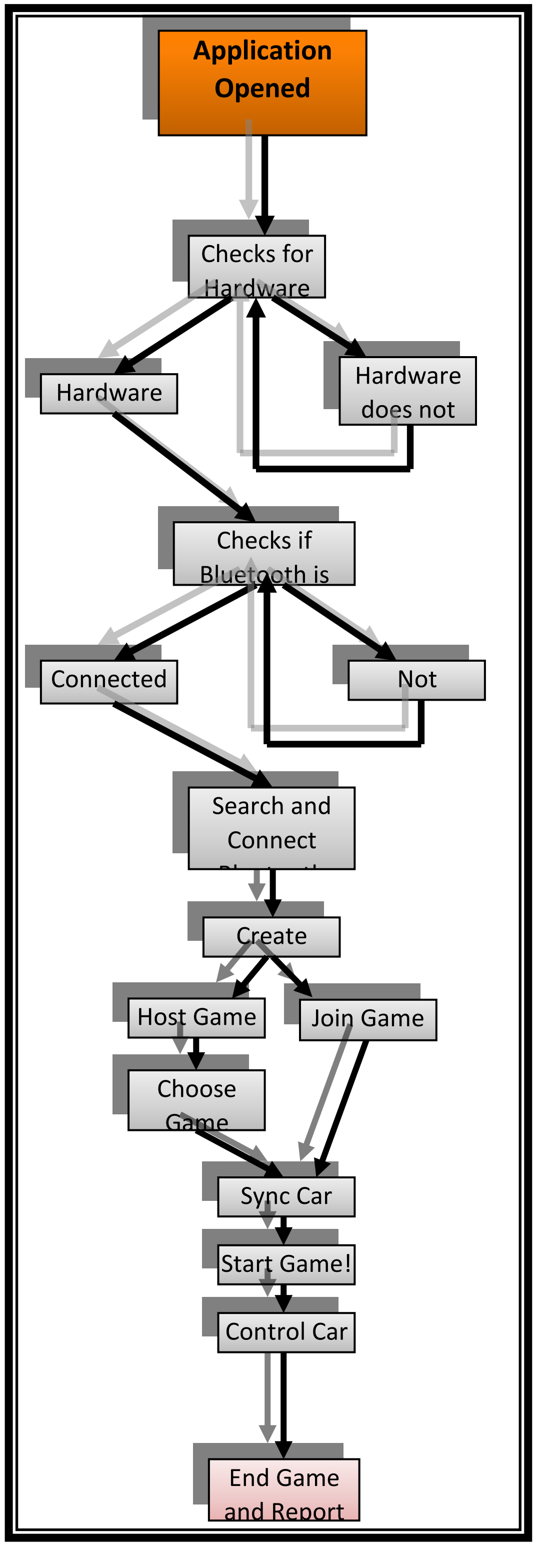


Fig. 7.3.2 Block Diagram Smartphone App

If they were not asked to enable Bluetooth, the user should see a label displayed at the bottom of the screen letting the user know that Bluetooth is on. If this is not displayed there is an error in the software that needs to be fixed. If there is a pop up notification, Bluetooth has been enabled when the user selected yes. If not, the program has worked properly so far. Next, the user will continue to the next section of the remote control application and wait for the Bluetooth enable request to come up again. Once in the application the tester will again select yes when notified to enable Bluetooth, but this time they will stay in the application and wait for Bluetooth discovery and pairing to occur.

At this point the user will be shown a list of available Bluetooth devices nearby. The user will select a Bluetooth device that corresponds to the RC vehicle. Once a device is connected, the user must attempt to communicate with the vehicle by tilting the remote control. While the user tests the communication between the remote control and the RC vehicle, they must determine whether the remote control movements are matching the direction and degree of movement seen on the RC vehicle. The vehicle must slow down, stop, speed up, reverse, turn left, turn right, or go forward only when directed by the remote control. Last, the users have to test the video feed from the video camera that is on each of the RC vehicles. This will be tested by just allowing the video to be displayed on to the phone screen. If not the application will be allowed to link with the phone so it will be disabled for a setting tab or section.

7.4 Final Test Plan

The final test plan will require us to test the subsystems together at once. Final testing will occur once all the individual subsystems have been tested together and subsequently combined. The Knight Brawlers team will have to assure that all the systems work together in unison to negate any sort of malfunctions. In order to achieve this we must first look actions versus reactions. When the gaming session is initialized, the app must indicate to the users that the cars are ready for battle. In turn, the RC cars themselves must be at their initial state. The team must test whether the first hit on the user’s vehicle changes the LED status and the car’s status on the app. The states on the cars and on the app must always be the same. The next thing to test is whether all the users are uniformly synchronized. The team must try every situation possible in the game in order to confirm that everything is in working order. This means trying out when all cars have full health, all cars have one hit remaining, simultaneous hits, etc. Testing every possible situation may expose any sort of errors in the coding. Several full games will have to be played one after another as well. That way the team knows that everything is reinitialized after each game and will not provide any sort of problem. Final testing will combine all the testing mentioned in the previous sections.

8 Administrative Content

8.0 Milestone Discussion

Keeping track of the Knight Brawlers team's milestones is an important way to document our triumphs and setbacks. At the moment, we can only account accurately for our Senior Design I course but it is also important to anticipate what may lay ahead in Senior Design II.

8.0.0 Senior Design I

During Senior Design I, there were several milestones that the Knight Brawlers team had to achieve in order to reach its final objective, which was to write a comprehensive report that documented the process, design, and specifications of the project. These milestones are illustrated in figure 8.0.0.

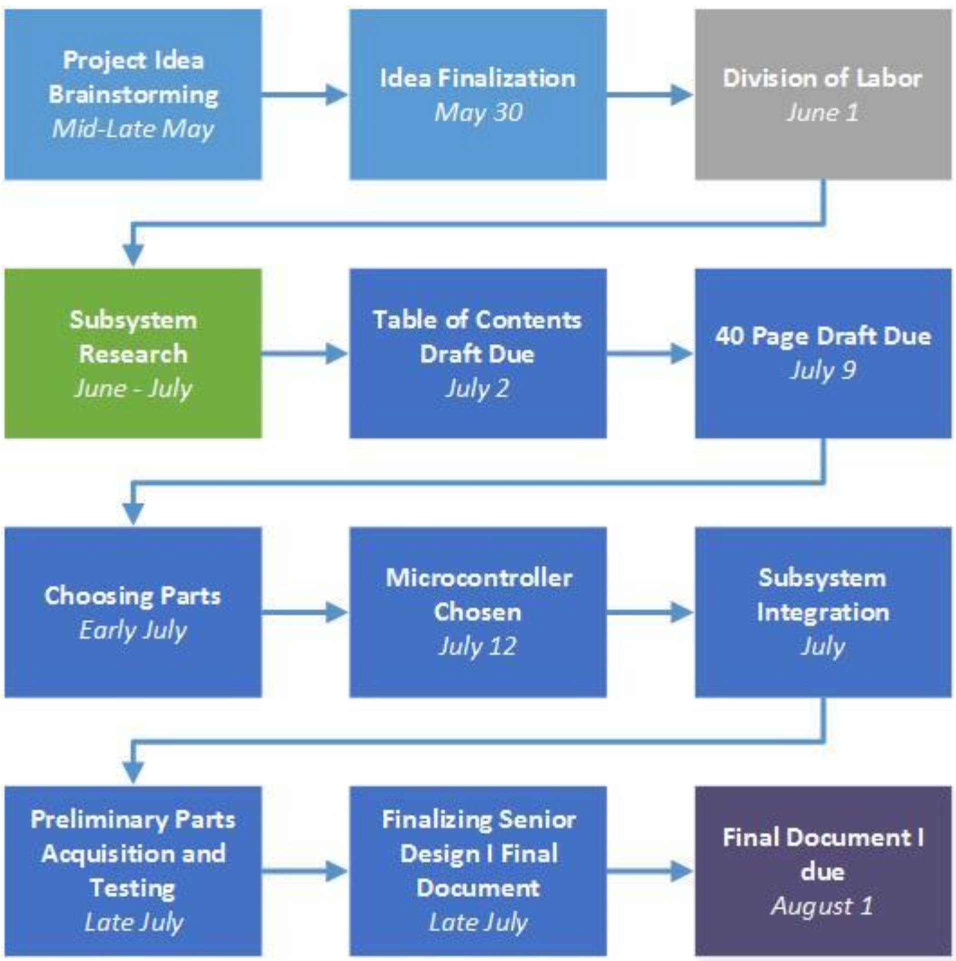


Figure 8.0.0 Senior Design I milestones

8.0.1 Senior Design II

If all goes well in Senior Design I, the Knight Brawlers team will have to look ahead to Senior Design II. Although no deadlines have yet been given by the instructor, nor any specific assignments, it is important to set a general timeline (figure 8.0.1). This is merely an estimate and could change drastically.

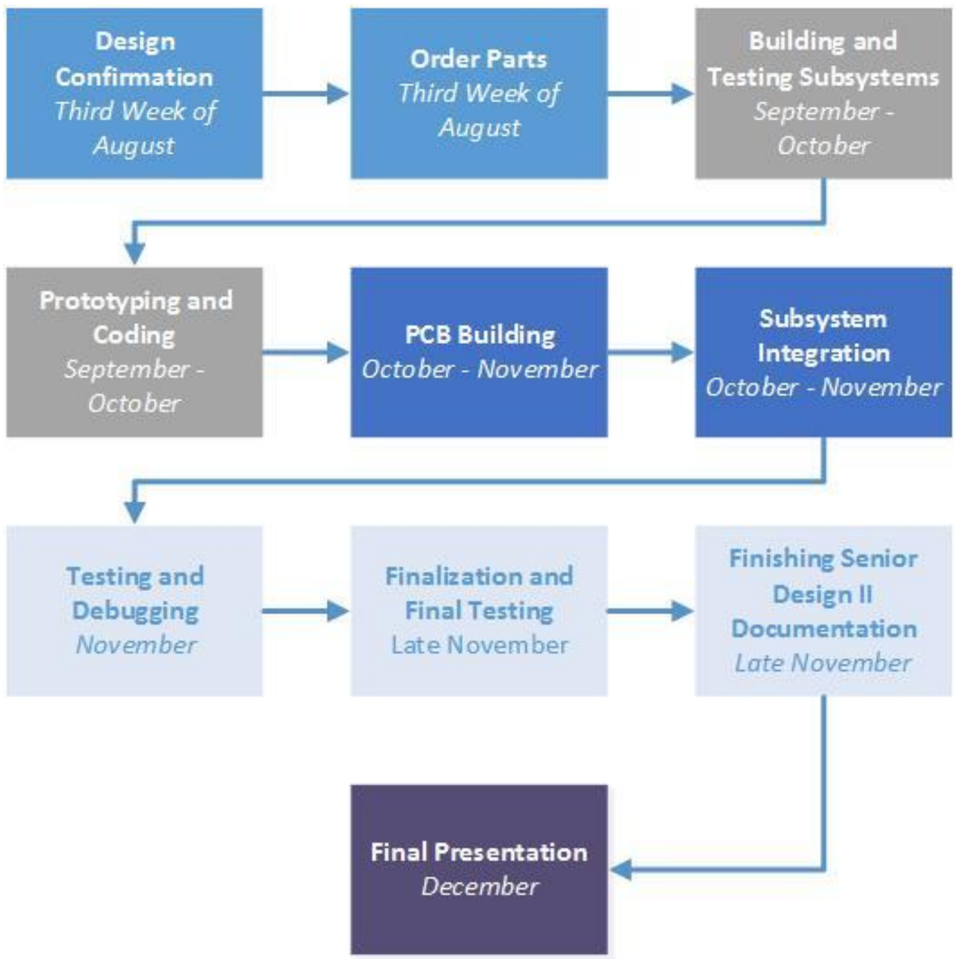


Figure 8.0.1 Anticipated Senior Design II timeline

It is paramount to try to abide by some sort of organizational schedule even if it may not end up being exactly like figure 8.0.1. As the Knight Brawlers team experienced in Senior Design I, it is helpful to be able to see a general time frame at which something should be done. Not only is this helpful with being organized and thorough overall, it also increases accountability for all the team members.

8.1 Budget and Finances

Although most of the money spent on the project will be spent during Senior Design II, the Knight Brawlers team looked into rough budget estimates before going ahead with the project. This was to assure that financing the project would not get overboard and perhaps be a hindrance to the process. Any costs to date and possible future expenses are illustrated in table 8.1.

	Cost Per Unit ($)	Units	Cost ($)
	Already Purchased		
Lever Micro Switch Sensors	3.50	3	10.50
RC Cars	40.00	1	40.00
	Anticipated Purchases		
Sensors for Other Cars	3.50	9	31.50
Separate RC Cars	40.00	3	120.00
Cameras Modules	50.00	4	200.00
PCB	35.00	1	35.00
RGB LEDs	3.50	12	42.00
Bluetooth Modules	50.00	4	200.00
Power Systems	25.00	4	100.00
Jumper Wire Pack	7.00	1	7.00
Miscellaneous	NA	NA	200.00
Total			986.00

Table 8.1 A list of possible finances

Another potential costs is that of the smartphones that will control the remote control cars. Since only one of the team members had an android mobile device, which is the desired platform for the application, the others will have to acquire one of their own. This matter however was left to the discretion of each team member so that they may get any model to their liking. Because of this, the price of the mobile devices will not be distributed among the team and is not displayed in the above table. Before the team went ahead with the project it was also agreed that if any unexpected expenditures had to be made, everyone had no problem having the budget reach 1500 dollars.

Due to the fact that the Knight Brawlers team was unable to get anyone or any organization to sponsor the project, all of the expenses occurred will have to be paid by the team members themselves. Due to that fact, the team will try to be responsible while working on the project in order to avoid any additional or unnecessary costs. This may mean having to go the extra mile in researching a part or components so that each team member knows that it will work well with the project and will be useful.

9 Process Considerations

9.0 Equipment Maintenance

Properly maintaining the equipment that we will use for the Knight Brawlers project will be an important consideration. The Knight Brawlers team wants to assure that there will be no equipment or part failures. To assure this, regular testing and proper storage of anything of use to the project must be performed. Storing the equipment properly will assure nothing gets damaged and is easily accessed. Much like a mechanic, the team will store everything neatly to produce an environment and habit conducive to efficient and enjoyable building. Regular testing will help make sure that the parts are stable and durable and will be in working order when it really counts. Regular calibration is a big part of regular testing as well.

It is also a priority to properly measure any components that the Knight Brawlers team will use for the project. As we have experienced in engineering labs, sometimes parts and components may not be the exact value as intended. A resistor for example may be labeled as a 1k ohm resistor but may in actuality measure as 990 ohms. Accounting for any slight deviations in components in equipment can help in explaining certain unexpected results.

9.1 Replacement Parts

To avoid unnecessary and careless setbacks, the Knight Brawlers team is planning on ordering plenty of replacement parts and components. Having extra parts on hand will allow the team to quickly replace anything that may malfunction and potentially compromise two semesters worth of diligent work. Although the team may not be able to apply this to the more expensive parts such as cameras or cars, it is beneficial to attain as many replacements as reasonably possible.

10 Final Thoughts

In writing this paper for Senior Design I, there are a couple of things the team can take from the entire process. Thorough research is important and vital to putting together an effective and complete semblance of what you want to accomplish. This was the first time that any of the team members have attempted a project with so much importance and work required. The team learned that it is better to work cooperatively rather than individually. Also, it is better to set a timeline in which to accomplish an objective so that it wouldn't be necessary to scramble and potentially jeopardize everything at the last minute. The team also learned that when you try and create something, it is essential to seek as many valuable resources as possible and to gauge whether going a certain route is feasible or not. By taking all these lessons learned from Senior Design I, the Knight Brawlers team can look ahead to Senior Design II.

In Senior Design II, the team can expect for the level of difficulty to rise as we try to take what we learned from Senior Design I and actually try to put it together. There are a number of things that the team should take extra time in assuring that they will work properly. The most important and challenging thing will be to have all the subsystem work together and efficiently. Also the team will have to assure that all the systems will work consistently and can work for long periods without any need of having to make frequent adjustments. It will require a lot of work, but the Knight Brawlers team hopes to come out successful.

Appendix A – Image Permission

Texas Instruments

Thank you for contacting Texas Instruments Technical Support. Your email has been received and a Service Request# 1-1074443912 has been assigned to your inquiry.

Hello Carlos,

I have included a link to TI's policy on this issue below

http://www.ti.com/corp/docs/legal/copyright.shtml

The policy specifically states "TI further grants permission to non-profit, educational institutions (specifically K-12, universities and community colleges) to download, reproduce, display and distribute the information on these pages solely for use in the classroom." If you have any further questions there is an email included in the page, copyrightcounsel@list.ti.com. If you have any further questions please direct your inquiry to this email. Thank you for contacting TI support.

Regards,

Michael Stepenaskie
TI Customer Support
Americas Customer Support Center
512-434-1560

[Prefix: Mr.]
[First Name: Carlos]
[Last Name: Davila]
[Job Title: Student]
[Company: University of Central Florida]
[Email: cdavila91@knights.ucf.edu]
[Phone: +407 (924) 5012]
[FAX:]
[Country: USA]
[Address1: 5109 Vista Lago Dr]
[Address2:]
[City: Orlando]
[State: FL]
[Postal Code: 32811]
[Part# or Description: TLC5940]
[Category: Access and Licensing]
[Application: Other]
[Design Stage: Other]
[Estimated Annual Production: 0 units]
[Production Date: 12/01/13]

[Problem:
I would like permission to use some images used on the TI website in my report for a school project. Thanks for your time.]

STMicroelectronics

Data sheet address for STM32F407VGT6:

http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/DM00037051.pdf

STM copy right policy:

TERMS OF USE

STMicroelectronics Terms of Use

PLEASE READ THESE TERMS OF USE CAREFULLY BEFORE USING THIS SITE. BY ACCESS OF THIS WEB SITE AND/OR BY USE OF THIS WEB SITE AND/OR THE DOWNLOADING OF MATERIALS FROM THIS SITE, YOU AGREE TO BE BOUND BY THESE TERMS AND/OR THE LAWS AND REGULATIONS AND YOUR USE CONSTITUTES AGREEMENT TO THE TERMS OF USE. IF YOU DO NOT AGREE TO BE BOUND BY THESE TERMS AND/OR THE LAWS AND REGULATIONS, YOU ARE NOT AUTHORIZED TO ACCESS THE SITE AND/OR USE THE SITE OR MATERIALS OR SERVICES, AND YOU ARE REQUIRED TO EXIT THIS SITE. STMicroelectronics ("ST") provides this web site to you, subject to these terms of use for the Web site and other web sites that are operated and/or controlled by ST or that are affiliated or linked with this site (collectively "the Site") and all the information, communications, drawings, photos, text, and other materials and services found on the Site (collectively "Materials"), is intended for the lawful use of ST's customers, and those authorized employees of ST's customers. All rights and licenses granted to you hereunder are personal to you and shall not include any right to grant sublicenses to any third party except as maybe expressly set forth herein.

ST reserves the right in its sole discretion to refuse or terminate access to the Site by you at any time. Permission to use the Site terminates automatically if you breach any of these Terms of Use. Further, the Terms of Use may be updated by ST from time to time without notice to you

RESTRICTIONS ON USE OF CONTENT:

The Materials contained in the Site are protected by copyright laws, international copyright law treaties, and other intellectual property laws and treaties ("IP Rights"). Except as stated herein, the Materials may not be reproduced, uploaded, linked to another URL, copied, transmitted, modified, displayed or distributed in any form without ST's prior written permission and the unauthorized copying, displaying or other use of any content (by linking, framing, or any other method) from this Site is a violation of the law. You acknowledge having been advised by ST that the Materials are protected by a variety of laws. You are hereby granted permission to access the Materials from the Site in whole or in part, solely for your personal, non-commercial use of viewing and browsing through the Site or ordering products from the Site. If pursuant to an authorization from ST, you download software from the Site, the software, including all code, files, images, contained in or generated by the software, and accompanying data, are deemed to be licensed to you by ST. Neither title nor intellectual property rights are transferred to you, but remain with ST, who owns full and complete title to the IP Rights. ST does not grant you any permission to use the Materials other than the permission expressly stated in these Terms of Use.

GENERAL INFORMATION:

The Terms of Use constitute the entire agreement between you and ST and govern your use of the Site, superceding any prior agreements between you and ST relating to your use of this site. If any provision of these Terms of Use is held to be invalid by any law, rule, order or regulation of any government or by the final determination of any state or federal court, such invalidity will not affect the enforceability of any other provision of the Terms of Use. The failure of ST to exercise or enforce any right or provision of the Terms of Use will not constitute a waiver of such right or provision.

Email Request:

Reply Reply All Forward

Fri 7/26/2013 8:38 PM
AllenDavila
Copyright Permission

To 'curt.paterson@st.com'

Hello,

I was wondering how one would attain copyright permission from STM for images on some product data sheets. I'm working on a school project and I was wondering if I could use some of the images I found in the data sheets. Please let me know.

Thank You,
Allen Davila