Tinjac Is Not Just A Cart

Group 13

Senior Design II Summer 2018 – Fall 2018

John McFarland	Computer Engineer
Shekh Arefen	Computer Engineer
James Johnson	Electrical Engineer
Edgar Velez	Electrical Engineer

Table of Contents

Figure Index

1 Executive Summary

With technology advancing faster than people can adapt, the future of retail and grocery shopping will soon be inundated with digital monitors, scan-and-go mobile apps, and cashierless checkout services. Despite some of the negative stigmas this may bring forward, it ultimately will provide shoppers with the swiftest and most efficient shopping experience.

Not only will these technological advancements increase overall effectiveness, it will provide assistance to those customers that are physically unable to go shopping for themselves or require some sort of assistance due to a handicap, injury, or any other impediments that they may face on a daily basis. So now we ask: How can we make shopping fast and easy for those that need help while still holding on to some of the more traditional shopping characteristics?

Our project idea is a self-driving shopping cart that would utilize motion and proximity sensors so that no physical touching of the cart is involved in order to navigate to the desired location.

The design would work where the customer chooses which grocery items they want on an LCD screen, and based on this, the cart would navigate to where these items are by using the shortest distance possible.

This would utilize proximity and motion sensors to navigate. Proximity would be used to ensure the cart doesn't run into objects, or other customers, and the motion sensors would be used to ensure that the cart driver is following behind, if not, the cart would stop until the driver resumes walking. The cart would also automatically adjust to anyone's walking speed.

The power source for the cart is to be based on solar cells since they're environmentally friendly and would be a quiet power source in comparison to a gas motor. The solar cells will convert the energy from the sun and store them into rechargeable batteries which will power on the cart.

Solar energy would work especially well for a shopping cart since when they're not in use, they're most likely outside, sitting in the sun, and gaining energy. The cart is perfect for any shopper while also benefiting the environment, and a great alternative to the classic cart design.

Tinjac could be considered nothing short of an improvement into the shoppers experience, once fully functional and implemented the above components into the system as a whole. Tinjac would create a friendly environment where the user could simply walk hands-free to their destination, accident free.

Not only beneficial for the consumers of the grocery store, Tinjac would also provide numerous benefits for the stores with Tinjac carts in them as well. Advertisement could be animated and placed on specific sections of Tinjac, using small amounts of the cart's power source.

Tinjac also would create an environment that consumers would enjoy more and thus theoretically increase the floor traffic to stores with Tinjac in them. Filled with numerous positive impacts, Tinjac's design is nothing short of influential.

2 Project Description

This chapter gives detail to what inspired this project and made it a possibility, as well as information that was taken into consideration when thinking of the project and what the project is supposed to do. Tinjac is a project composed of several motivations which founded the ability to put in the extra effort to reach all the goals we set out to accomplish.

In order to achieve every core goal laid out in our cart's list of objectives and goals, Tinjac must be carefully planned and implemented to minimize the amount of mistakes that could be encountered during the combination of all the subsystems.

Each of the following subsection in this chapter touches on both the "why" and the "how" when it comes to Tinjac, both being equal in magnitude in regards to importance.

2.1 Motivation and Goals

The motivation for this project stems from inspiration in the wave of "smart" technology (smart phones, televisions, etc.), and thinking of a current invention that could possibly benefit from the "smart" treatment. When the future is envisioned, it will be a world run by technology, and technology that can help people with every day simple tasks and make them even simpler, and that's what Tinjac is, a cart of the future. The main motivation is to make the future close to today.

Other motivation for this project comes from the University of Central Florida and years of education here and wanting to create a senior design invention that demonstrates most of the skills that have been learned here and puts it all together. It's necessary to have an invention impressive enough to not only look good for the University, but also on future job resumes to impress employers.

The main goal of this project is to create a device that's accessible for any type of person. The device should be able to navigate through a store at a pace comfortable enough for the customer and reach each destination without running into anything through the use of motion and proximity sensors that will be placed all around the cart in a non-distracting manner so that there is no room for error.

Goals of this design include being able to save the customer time as well as money. With a simplistic LCD design, customers can easily choose the items they need, and then the cart will take the shortest path to these items. This is especially helpful for someone who is unfamiliar with the store. This design makes a 20-minute trip to the store of endlessly looking for every item, to just knowing exactly where each and every item is, which would shorten shopping times by at least seventy five percent.

Another goal of this project is to benefit and aid the customers that may have some sort of physical impairment due to an injury. They may not be able to physically push a cart for whatever reason and therefore would greatly profit from this product. Without needing to push, this impediment becomes resolved thanks to Tinjac.

Another hopefully goal of this project is to make the unit as cheap as possible. This is so we can attract and market the product towards large supermarket chains such as Walmart and Publix. A lower cost would increase our chances of being integrated within a few stores.

2.2 Objectives

The overall objective of this project is to essentially introduce a prototype that can potentially be adopted by stores in order to help make shopping easy for their customers who are handicapped, debilitated, or injured, as well as increasing the general feasibility for every customer.

The desired end product would have the power supply, and main logic board in an enclosed container so there aren't any exposed wire or circuits. The solar panel will be placed on the bottom of the cart on top of the grating, with the battery under the grating.

The motion and proximity sensors will be placed on the front of the cart, and one on the back so that the cart has a full range of protection from unwanted interaction with other objects. The only object that should be exposed is the LCD display and the corresponding buttons for it.

The cart requires at least 12V to be able to supply enough voltage to the two DC motors. The voltage source would be able to be placed in parallel with the motors so they both receive 12V. The launchpad that is being used, MSP430F5529LP, requires 3.3-5V. The launchpad can output up to 5V, and the sensor modules require 5V, so the output of the launchpad could be connected directly to the sensors to power them on and control their signal.

For the software design, the controller will be able to give the signal controls to the corresponding IR and sonar modules (will be used for the motion and proximity sensing) which direct the speed and turning of the cart.

The most challenging objective is having the cart be able to map out its location and based on that, the relative location of the grocery items. The same way a car is able to track how many miles it travels; the cart will have to do the same except for both the x and y axis. As in, if the cart is starting from the base of the store and the cart goes 3 meters upward, and then 5 meters to the left, then the coordinate will be (3, -5), and continuously updating its coordinate until the cart as it travels through the store. This is difficult since the cart won't always be turning at a perfect ninety-degree angle, and a slight error in the coordinate could throw off the cart's distance.

Below is a diagram which details how the main objective will be executed along with a diagram on how the cart's location is being calculated.

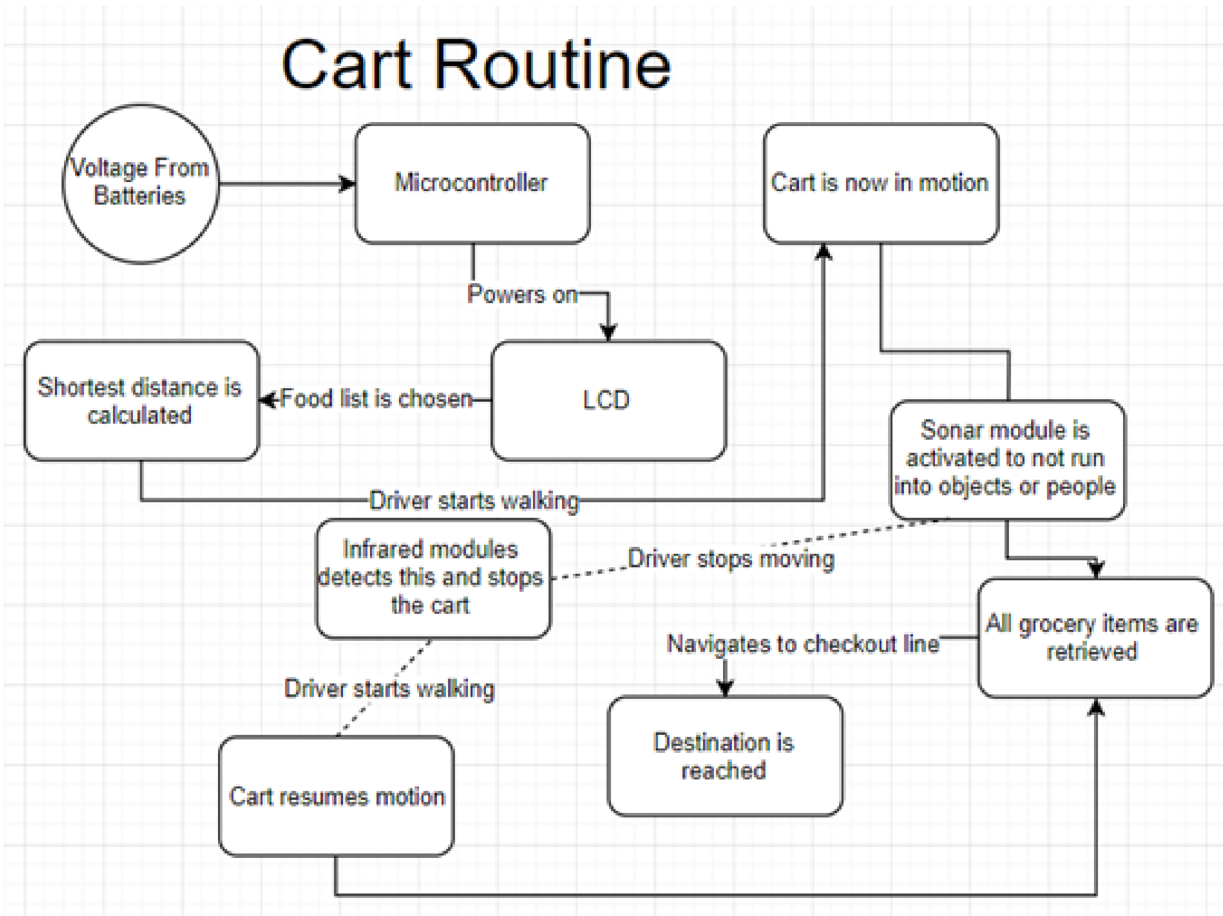

Figure 2-1 ~ Basic Layout of Cart Routine

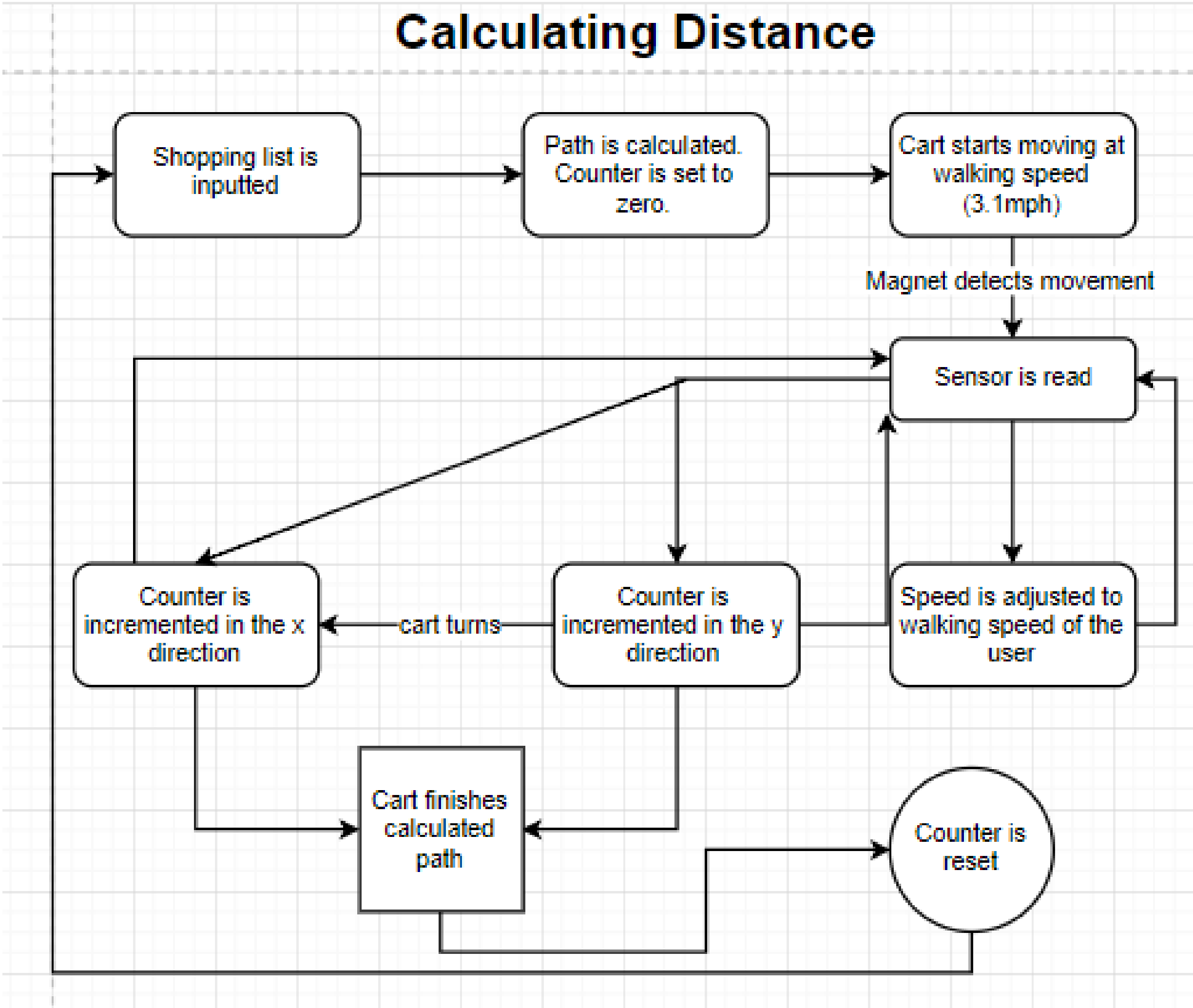

Figure 2-2 ~ Distance Calculation Diagram

This diagram (Figure 2-2) displays how the project construction will be split up. The software part of the project will be divided evenly amongst the two computer engineering majors that are part of our group while the hardware and electrical set up will be evenly divided amongst the remaining two electrical engineering majors. Additional details on the work division can be found in section 2.3.

2.3 Work Division

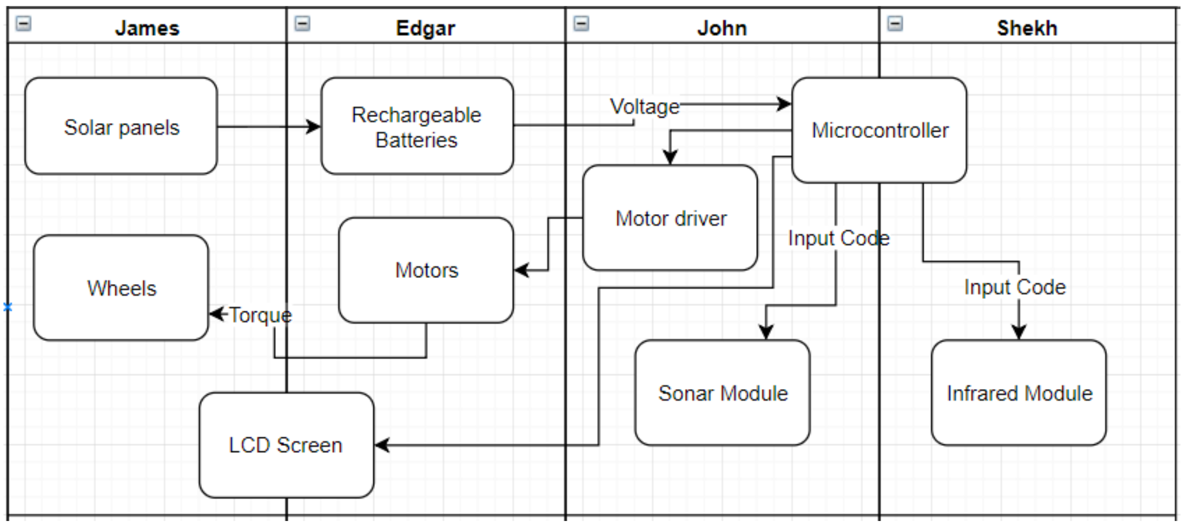

Figure 2-3 ~ Work Division Diagram

Tinjac is the masterpiece crafted from the masterminds known as John, Shekh, Edgar and Jimmy. The design and implementation of this project worked out perfectly as Tinjac has nearly a fifty-fifty division of both software and hardware subsystems in terms of work hours. With John and Shekh being computer engineers, and Edgar and Jimmy being electrical engineers, the work division was split into two major divisions: hardware versus software.

2.3.1 Software Work Division

Between the two computer engineers, dividing the software side of the project was much simpler to implement than initially worried about. Once the divide-and-conquer approach is broken down each individual objective, the work will simply be split in half and given to each computer engineer. However, in order for them to accomplish their objectives as efficiently as possible the work will all be done at the same time.

Working together on separate problems allows the two computer engineers to be able to communicate and collaborate on their own hurdles. While they do not share the same problems, they will be able to solve each other's issues as they would work through their problems aloud. This creates an environment that accomplishes two things: a work environment that stimulates communication and team-building, as well as creating our own take on "rubber duck decoding". Using the rubber duck decoding method, we will encourage one another to solve all of our problems in half the time.

2.3.2 Hardware Work Division

With two electrical engineers, creating the physical circuits and subsystems of Tinjac will be simpler versus only have one member working on those specific subsystems.

Once the circuit and physical systems is divided from the entire system, both electrical engineers will be able to take the subsystems and decide which of them will be the most work. When this step is done, they both will decide which subsystems they desire to work on to split up the work evenly. If one finishes their subsystem they will be able to jump over to one of the remaining subsystems and assist with the design and implementation of that system. This will create a schedule for both of the engineers to work fairly, while also supporting each other and design Tinjac as physically efficient as possible.

2.3.3 Work Division Diagram

The previous figure (Figure 2-3) also illustrates how the work is expected to be divided, where everyone should be an expert of the objects in their columns through research and implementation. Work was divided this way according to everyone's specialized skills as well as with what their major is. Since James and Edgar are electrical engineers, their tasks are more power focused, and since John and Shekh are computer engineers, their tasks are more coding focused.

This in absolutely no way means that there can't be crossover or help from other teammates on an object that's assigned specifically to that person. It just means that they should be the most knowledgeable at that specific part. During any critical hurdles during the implementation of the cart, or any other part that requires more hands than those 'assigned', would permit all members of the team to come together and find a solution that would best fit Tinjac's needs while also satisfying the teams wants.

2.4 Requirement Specifications

The requirement specification section covers the intended features of the Tinjac. It goes over the core features that the group decided could be implemented into the Tinjac design and that it should be able to accomplish when it is finished. It also covers some stretch goal features that if enough time is left to properly implement they are considerations to add to further improve the Tinjac.

Each following subsections touch on several of the core points listed while our team agreed to when narrowing down the feature list to the base requirements. Once these steps have been completed, we expanded our possibilities and voted on what specific features would be a good idea to list as potential stretch goals for our project.

2.4.1 Core Feature Overview

- An LCD display that allows a user to select from a given list of grocery items. The cart then takes the customer to each item. Items are able to be added/removed from the grocery list even after the initial list is inputted.
- The shopping cart will have a selected store's display programmed to know where items on the list are located.
- The shopping cart should be able to avoid obstacles using IR and sonar module sensors to calculate the shortest path to gather all the items on the list.
- The cart should be able to know when the user steps away from it and will stop accordingly.
- The battery on the shopping cart is rechargeable and will use a solar panel to supplement this as well.
- Solar panel to supply 12V to a rechargeable battery.
- The weight should be kept under 15lbs, with the weight of the cart being 10 lbs. or less.
- Cart should be able to hold at least 50 lbs. and still function without slowdown.
- Dimensions are 21 by 24.5 inches. A regular sized cart won't be used due to budgeting.

2.4.2 Stretch Goal Overview

- Implementing an app that has the same functions of the LCD display.
- Using a button (or the same app) to call a shopping cart to you.
- An alternative mode would be to (likely hit a button or app to enable this mode instead to) allow the cart to follow the shopper, instead of the shopper following the cart. Most likely will be done by placing a device like an infrared light, on the shopper for an infrared sensor to track.
- Have a sensor on the cart that scans each grocery items to calculate the total price and lets the user pay right there on the spot
- Cart should be able to hold at least 100 lbs. without slowing down.
- Cart should be able to follow the shopper to their car, making their shopping journey slightly more easier. This would be much more difficult to implement versus having the cart just traverse through the store.

The stretch goals listed above are explained in great detail and length in section 8.4 Stretch Goals.

2.5 Quality of House Analysis

Matrix Key

↑ = Positive correlation

↑↑ = Strong positive correlation

↓ = Negative correlation

↓↓ = Strong negative correlation

+ = Increasing the Requirement

- = Decreasing the Requirement

	Marketing	**High Power**	**High Performance**	**Safety**	**Cost**	**Engineering Targets**
Engineering		+	+	+	-	
Power Efficiency	+	↑↑	↑	↑	↓	≥70%
Pathing Accuracy	+	↑	↑↑	↑↑	↓	≥85%
Obstacle Avoidance	+	↑	↑↑	↑↑	↓	≥85%
Speed	-	↑	↑	↑↑	↓	≥ 1 m/s
Weight	-	↑↑	↑	↑↑	↑	≥ 80 lbs.

Cost	-	↓ ↓	↓ ↓	↓	↑ ↑	≥ $300

Figure 2-4 ~ Quality of House Analysis Table

The final cart design needs to have a balance of all the best qualities while also attempting to consume as little cost as possible. Since a smaller cart is being used, this gives the ability to use cheaper motors since not as much torque would then be needed, and then a motor with less torque would require a less expensive power source. The remainder of the cost would be able to be focused on accurate motion sensors and efficient signal controlling to ensure for a great pathing accuracy and object avoidance.

In Figure 2-4, the parameters listed on the left-hand side of the table are the engineering requirements that set that boundaries for Tinjac. Along the top of the table are the marketing requirements, which would restrict Tinjac in a more commercial sense. These restrictions are often related to one another and together have an effect on the system as a whole.

Also showing within the same figure (Figure 2-4) is the goals we state for each parameter. Found on the right side, these objectives were unanimously agreed upon on the first designing Tinjac. Setting these limits for ourselves continuously created frameworks for us to build around when designing each of the subsystems for Tinjac.

Having these restrictions in the back of our minds while building Tinjac also creates a positive environment for all the team members. It would allow us to reflect upon and make the right steps towards a more efficient and stable version of Tinjac possible.

3 Research Related to Project Definition

In order to come up with a design for our shopping cart, we had to break down each single component that would be necessary for the unit itself. After figuring out the components, extensive research was conducted on each part and the several options that we came across. These components consist of the shopping cart, microcontroller, IR and sonar module sensors, motor driver chips, LCD modules, batteries, and solar panels. This section covers existing projects and relevant technologies that are related to the Tinjac.

It also extensively covers the components that are being considered to build the project. It will discuss the pros and cons of the different parts that are being considered and the products the group decided as a whole to use.

Lastly, this section will discuss some of the influences that drove the group's decision to to use certain parts over other parts even if the other part is more suited to the project.

3.1 Existing Similar Projects and Products

There are a number of smart inventions that help to improve the shopping experience for customers which have been in the process of being made by large companies. This inspiration for them was all started in response to Amazon's new store, Amazon Go, where the most impressive feature of the store is that it's completely cashier-less. The customers can simply just walk out of the store with their items, as the items are already scanned and then automatically billed to the customer through the Amazon app. By using a series of cameras and sensors, the inventory management is able to determine which item is picked up, based on weight, location, image analysis, and a shopper's checkout history [1].

Figure 3-1 ~ Amazon Go Functionality

Amazon took about four years to make the technology in their store a reality. Four years can be a long time in regards to technology. Given Moore's Law, where the number of transistors in a device can double every two years, this means that retail stores will have to work fast to keep up, especially now that Amazon has introduced these new capabilities into the playing field. Large companies such as 7/11, Wawa, Walgreens, CVS[2], Walmart, among others, are all striving to meet these consumer needs for a more efficient shopping experience.

Walmart has brought the most interesting advancements to the table. In 2016, Walmart unveiled their plans in development that includes a driverless shopping cart which guides customers to the items they desire, and robots that scan the shelves to check the stock, as well as the ability for customers to be tracked through the usage of wearables[3].

Not many specifics are known about the technology which will be used, but Walmart's plans include their app to have an ability for the cart to bring the customers the items from their grocery list to the parking lot so customers would have the option to not even step foot into the store. If the customers do decide to go into the store, the carts would be able to lead them directly to the items they want.

Walmart has also put into place some measures to ensure that every item is in stock by having tower like robots[4] roam the store and take pictures of the shelves and analyze them to check if a product is missing, if it is, the stock will be checked so that it'd be able to be replaced. The stock is checked through the use of drones which scan all the items.

Another invention in the works that is probably identical and definitely much more advanced than our proposed project is the autonomous shopping cart known as "Eli" unveiled by a South Korean chain known as E-Mart. This cart has sensors that is able to differentiate between human voices in order to follow their particular consumer. It has sensors to detect obstacles in its path and go about avoiding them.

Not only that, payments can be made as well using the shopping cart itself. This completely eliminates the need for cashiers. Using this project as a reference, our shopping cart will have some similar features that won't be as advanced as the Eli but still aimed towards the same goal: making the shopping experience easier for all users.

3.2 Relevant Technologies

The most obvious relevant technology would be the autonomous cars that have been coming out on the market. These cars detect other vehicle's locations nearby through the use of radar sensors positioned around the car. The cars also have video camera to detect lights, signs, obstacles, and pedestrians and tracking vehicles. Lidar sensors are used to help identify lane markings. Ultrasonic sensors are placed in the wheels to help detect curbs and vehicles, then with all the data from the sensors, the car's computer takes all the data into account to control steering, speed, and braking. [6]

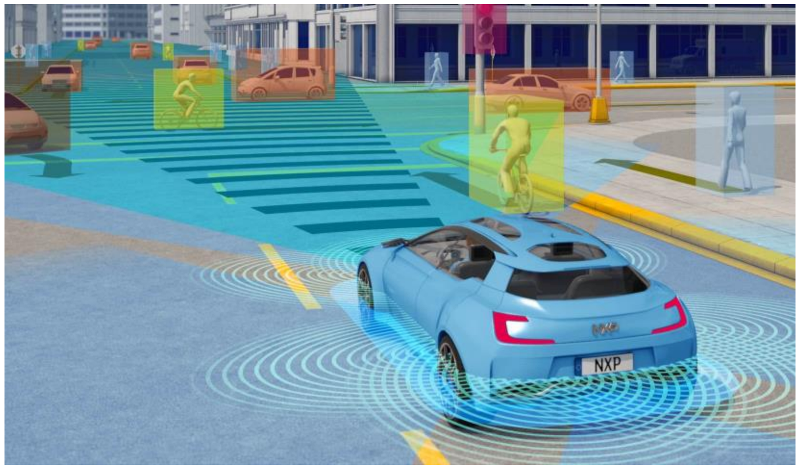

Figure 3-2 ~ Smart Car Sensory Field of Effect

This is all relevant to how the cart will operate but on a more scaled down level. Instead of individually identifying people, vehicles, objects, it will just group them all together and avoid everything, except for the sensor module behind the cart which identifies the speed to drive the motors.

Sensor technology is used everywhere especially with the new technology advancements. Virtual reality is making more and more use of sensor technology, bathrooms are almost hands free due to infrared sensors, and television remotes have been wireless since 1955 with the help of infrared receiver and transmitter sensors.

The technology hasn't changed much since it was first used, but what has changed is the ability to use these sensors to help with more complex computing algorithms which take into account the data from sensors.

3.3 Strategic Components and Part Selections

In order for our shopping cart to be as fully autonomous as possible, we must be able to strategically pick out each single component and determine the compatibility it would have on the other parts of the system. This may be determined by numerous factors such as: cost, power usage, force (or rotational force), physical size, and more.

To determine what we needed as a group we chose several options that could fit what we would need to make Tinjac work. Then we communicated in a group setting to decide together which of the parts we picked out would be the best add-on towards Tinjac.

3.3.1 Microcontroller Options

Listed below are several options of microcontrollers that we have considered during the planning phase of our project. All of these microcontrollers have unique specifications in terms of memory capacity, power characteristics, cost, and the number of general purpose input and output pins. By analyzing each different microcontroller and its specifications against each other, we were able to come up with the most optimal microcontroller to use. We also had help selecting a microcontroller from Dr. Samuel Richie.

The final deciding factor was the result of a combination of a few traits, but most specifically our Senior Design restrictions as well as our computational need. However we chose the best fit for our design as a group among the several option we found as potential candidates. This selection, along with all the other electronic parts of our project, will be stated in Section 3.5 Parts Selection Summary.

Texas Instruments MSP430F5529LP Microcontroller

The Texas Instruments MSP430F5529LP was considered by the team because of its competitive parametric and features. These features include having a 16-bit RISC Architecture, Non-volatile memory size of 32KB, a RAM of 4KB, and 44 GPIO Pins. in terms of cost, this is one of the cheaper MCU (microcontroller unit) options. For the purposes of our project, this microcontroller will serve well in terms of powering our LCD module and the motor drivers. Below is the functional diagram of this microcontroller:

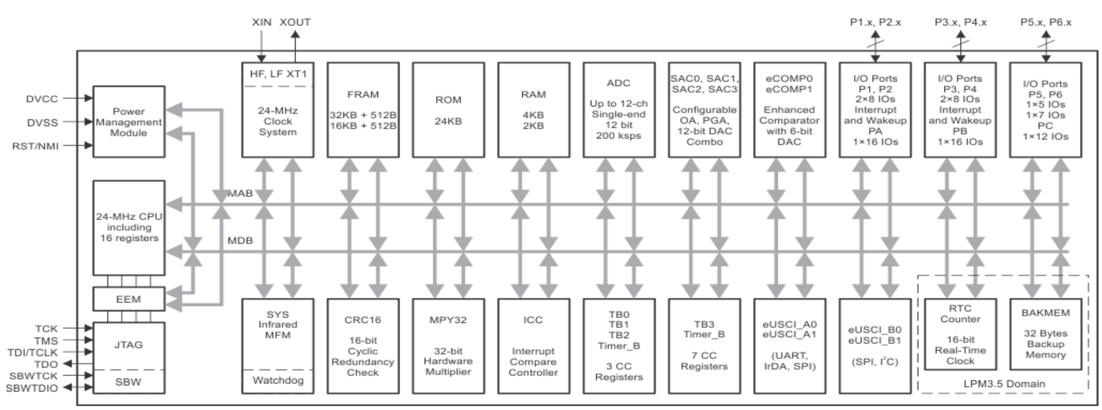

Figure 3-3 ~ MSP430 Function Diagram

Texas Instruments MSP432P401R Microcontroller

Along with the MSP430 microcontroller, the MSP432P401R was also highly considered by our team. It was also recommended by Dr. Samuel Richie so this has the highest chances of being selected to integrate into our project. This was an MCU we highly reflected upon and contemplated over because of its ARM 32-Bit Cortex CPI. It has up to 256KB of flash main memory, up to 64 KB of SRAM and up to 48 I/O pins with Capacitive-Touch Capabilities. Below is the functional diagram of the microcontroller.

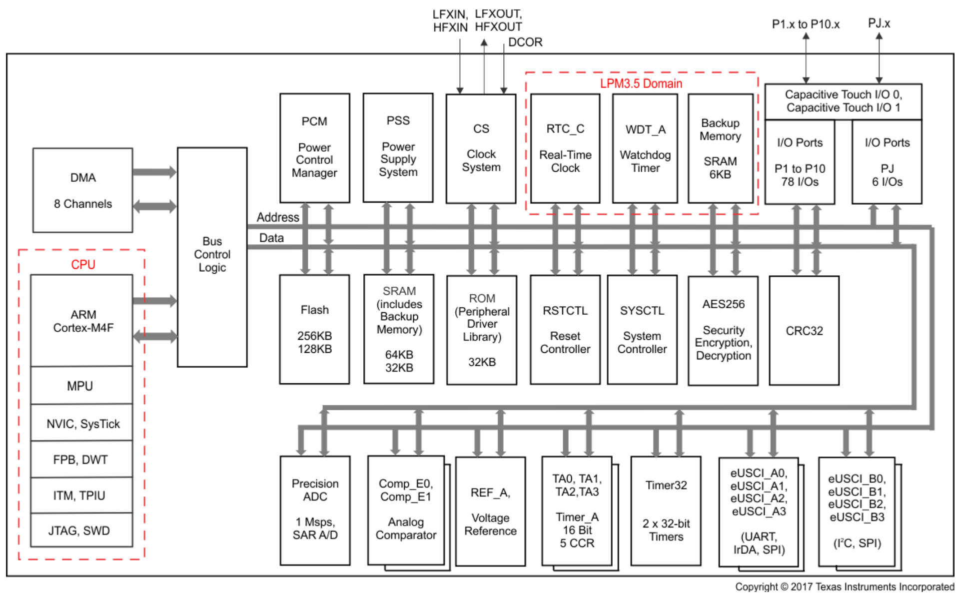

Figure 3-4~ MSP432 Function Diagram

Raspberry Pi 3 Model B+

The Raspberry Pi Microcontroller was actually the first MCU we considered. This was mainly due to the fact that the Pi has its own operating system. Being as this is the latest model that was recently released, its specs include a quad core 64-bit processor, Bluetooth 4.2, wireless Lan, and high speed ethernet capabilities up to 300Mbps. Similar to the MSP432 MCU, it also has an ARM Cortex CPU. Although this MCU is very dominating in terms of its technological capabilities, it is ultimately unnecessary. During the planning phase, we considered using robot vision to have our shopping cart move through the aisles of a store. That would have required a camera so therefore, the Pi would have been the final choice. But that requirement was discarded in order to have a simpler and more cheaper approach. Displayed below are some of the schematics of the GPIO pins.

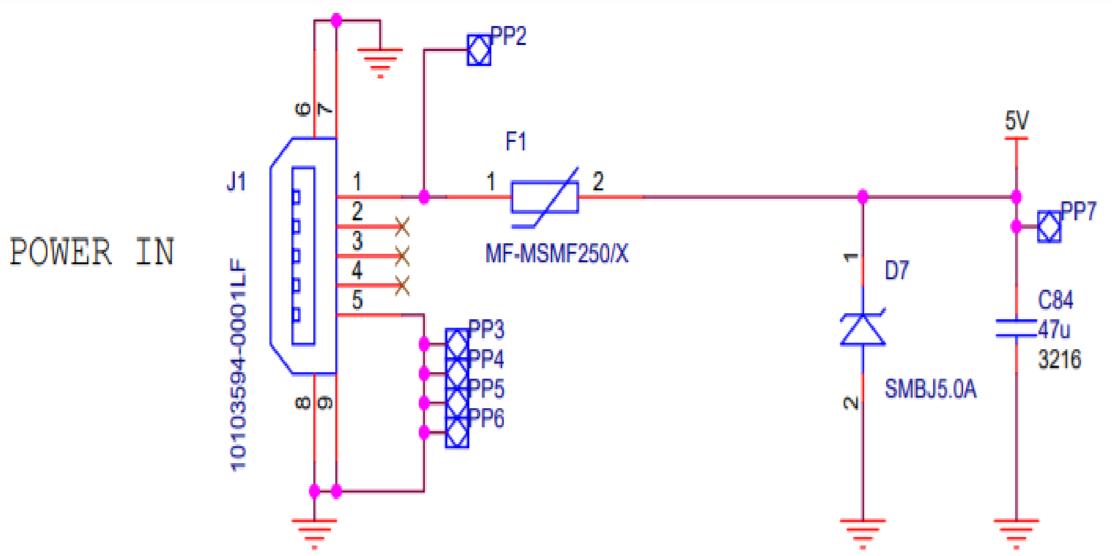

Figure 3-5 ~ Pi 3 B+ Power Schematic

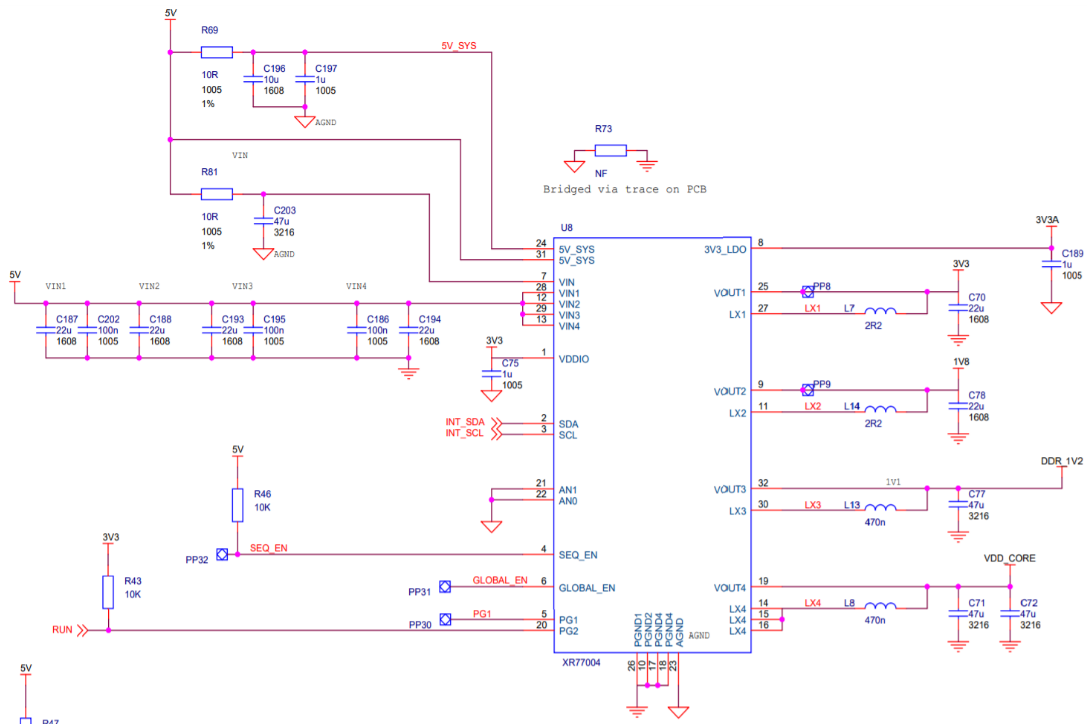

Figure 3-6 ~ Pi 3 B+ Full Schematic

Arduino Diecimila ATmega328 Microcontroller

The Arduino Diecimila MCU is the safest option to go with in terms of usability and cost. Arduinos are easy to use and contain several manuals on its operability but it would have been difficult to implement with our project. The Diecimila comes with 32KB of flash memory, 2.048KB of SRAM, 14 GPIO pins, and provides UART serial communication. This microcontroller was considered prior to having any sort of professional feedback. Ultimately, this will probably be discarded as an option. Displayed below are the block diagram and the AVR core architecture that the MCU provides.

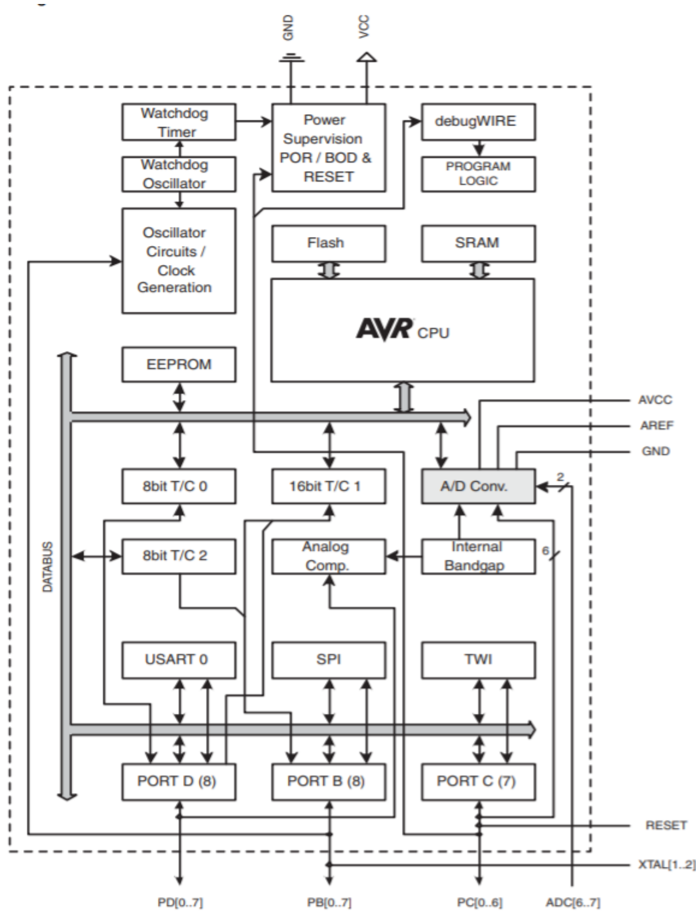

Figure 3-7 ~ ATMega328 Diagram

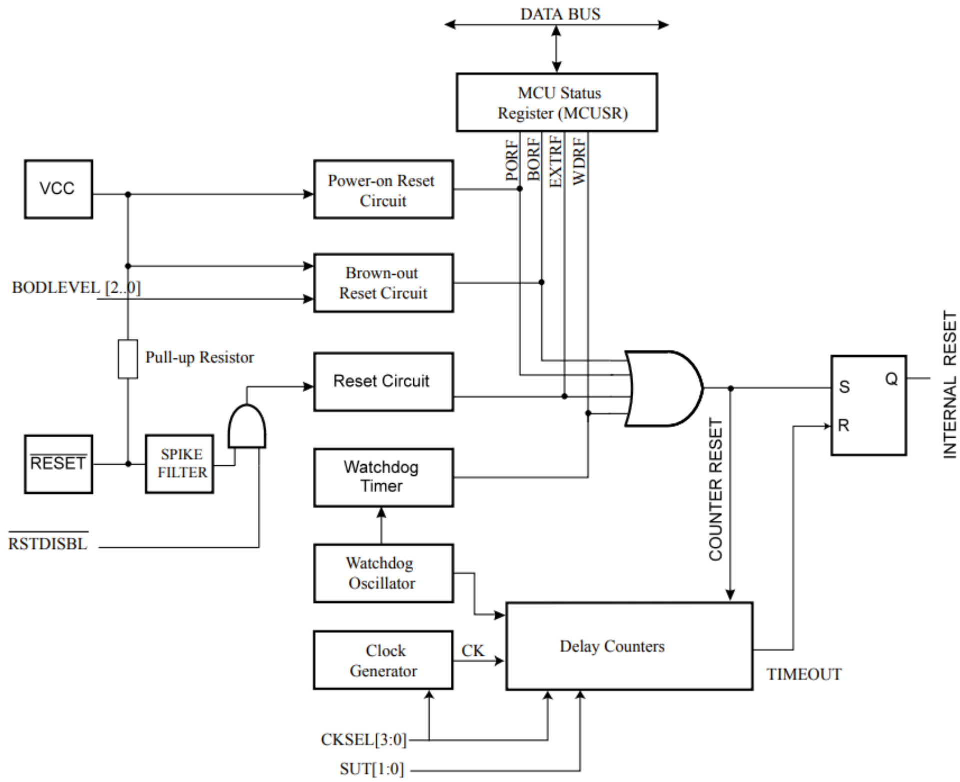

Figure 3-8 ~ ATMega328 Function Diagram

STM32F4Discovery Microcontroller

This microcontroller comes very close to the MSP432 in terms of power, memory capacity, and durability. It comes with a 32-bit ARM Cortex, 1Mb of flash memory, and 192KB of Ram. It also has eight LED's and 100 GPIO pins. In terms of specs, this clearly has the MSP432 beat. But the decisive factor that will ultimately aid in our decision is cost. This is obviously costlier than the MSP432 and cost is something we have to highly consider when making our choice. Displayed is the functional block diagram of this MCU.

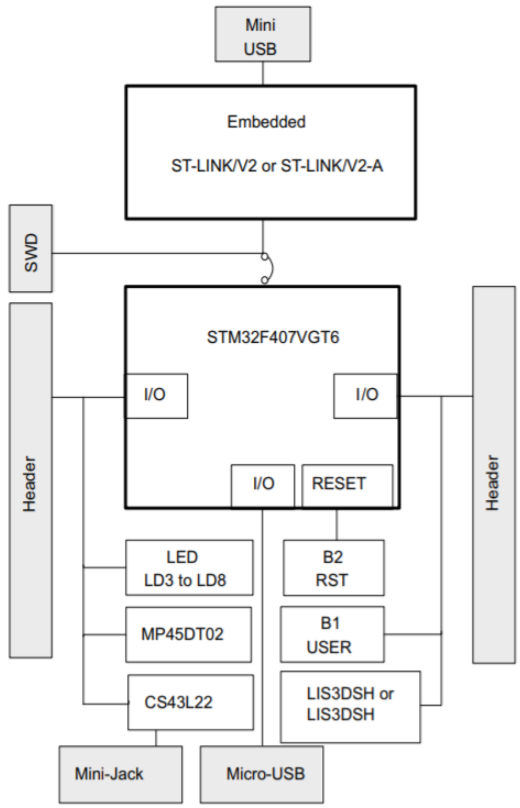

Figure 3-9 ~ STM32F4Discovery Diagram

3.3.2 Microcontroller Comparisons

Out of the five microcontrollers listed above, there are four different specifications that will be essential in determining which microcontroller we will choose to implement with our shopping cart.

The first specification is memory. With a microcontroller unit, the most common types of memory are flash memory and RAM. Flash memory is the ability to retain data after a complete power cycle. This allows for more complex source code to be written to the microcontroller. Therefore, the higher the flash memory, the more data you can store. The second type of memory is RAM memory. Unlike flash memory, RAM memory is volatile. This means that once the power has been turned off, whatever data that was stored has now been complete erased. Therefore, this is used to store data that is temporary, and not pertinent to the MCU in the long run. Similar to flash memory, the more RAM that is available for an MCU, the more data it can store and the more it can do. This will fundamentally let us add more features and enhancements to our shopping cart without sacrificing other attributes.

The second specification is power usage. Our overall goal is to have a microcontroller that utilizes the least amount of power in a given amount of time. Most of our considered MCU options are ultra-low powered devices but that will not be the biggest factor in our decision. Ideally, we want enough power so that the microcontroller can power the LCD, motor drivers, and IR sensors for a certain amount of time but not to utilize so much power that it runs out before completing its task. For example, we would expect our selected microcontroller to power the shopping cart for at least a few hours before it needs to be charged again.

The third specification, and probably one of the most important, is cost. Cost will play a huge role in our selection, not only because of budgeting purposes, but also because we want our product to be as inexpensive and affordable as possible. Economically, we want our product to be used by anyone and everyone that needs help shopping.

Having a microcontroller that is costly will definitely come with specifications that are much more advantageous than one that is cheaper but it will also be unnecessary financially and therefore would not be the best cost-effective resource.

The last specification that went into play is the clock speed of each microcontroller. In this case, the faster the clock speed, the better. This is because we want our microcontroller's CPU to be executing tasks as fast as possible. In terms of time management, we want the user of our product to be able to input their grocery item selection and start their shopping as soon as possible. It would be inefficient and frustrating to have to wait for the microcontroller to perform the task given in a long period of time.

Microcontroller	**Memory(Flash/ RAM)**	**Power Usage**	**Cost (for one unit)**	**Clock Speed**
TI MSP430F5529LP	16KB/0.512KB	0.438 mW	$4.63	16 MHz
TI MSP432P401R	256 KB/64KB	0.13 mW	$7.72	48 MHz
Raspberry Pi 3 Model B+	NA/1GB	1.9 W	$35.00	1.4 GHz
Arduino Diecimila ATmega328	32KB/2KB	0.360 mW	$3.50	20 MHz
STM32F4Discovery	1MB/192KB	0.2928 mW	$19.50	180 MHz

Figure 3-10 ~ Chart Comparison of Microcontrollers

As you can see, all of these different microcontrollers each have their advantages and disadvantages. Out of our options, the STM32F4 Discovery MCU provides the most flash memory and RAM. Its clock frequency is also the second fastest but it is also the second most expensive out of all. For our purposes, we are going to go with a microcontroller that's a bit more inexpensive yet still has a good amount of memory to it. Sacrificing the amount of memory in order to spend less money is something that our project can afford. The Raspberry Pi would have been a great option had we needed to utilize a camera for our shopping cart. It has the most superior clock speed and an incredible amount of RAM but it is the most expensive unit there is. Therefore, this option will no longer be considered. The Arduino Diecimila ATmega328 and the MSP430 MCUs are close competitors to each other in terms of cost, clock speed, and power consumption but the Atmega328's specs end up being more preferable by just a small amount. Lastly, it seems that the MSP432 is in between the MSP430/Atmega328 and the STM32F4 Discovery/Raspberry Pi. It doesn't have an excessive amount of memory like the Discovery and the Pi, has a reasonable price, and also has a higher clock rate than the MSP430/Atmega328. All in all, its power consumption is the lowest out of all 5 microcontrollers. We are leaning heavily towards choosing the MSP432 microcontroller for our shopping cart.

3.3.3 Infrared Sensor Options

Listed below are the three different choices we had for an infrared obstacle detection sensor. There is the Gowoops, Vishay, and the Osoyoo IR sensors that we have decided to pursue. Some of the specifications we looked at were detection distance, cost, weight, and the effective pointing angle of the sensor. There were also other factors we had to consider such as the level of difficulty to implement the IR sensors and how much power it would use from the microcontroller. Analyzing all these specs will later help us determine which IR sensor to go with. Below is a description of the IR sensors and their capabilities.

Gowoops E18-D80NK IR Sensor

The Gowoops IR sensor has been a huge contemplation with our group because of its high detection distance range, which is a maximum of 80 centimeters. This is impressive because most low power IR sensors have a standard 40-centimeter maximum distance. Having twice the detection distance range is a huge advantage and would give our shopping cart an extra edge in detecting obstacles. As with every advantage, there are a few disadvantages. The most crucial of them being cost and the effective pointing angle. The cost is about $7.00 per unit and the effective pointing angle is 15°. This is a downside in comparison to a the Osoyoo IR sensor that we have considered. Displayed below is the diagram of how the transmitter and receiver works.

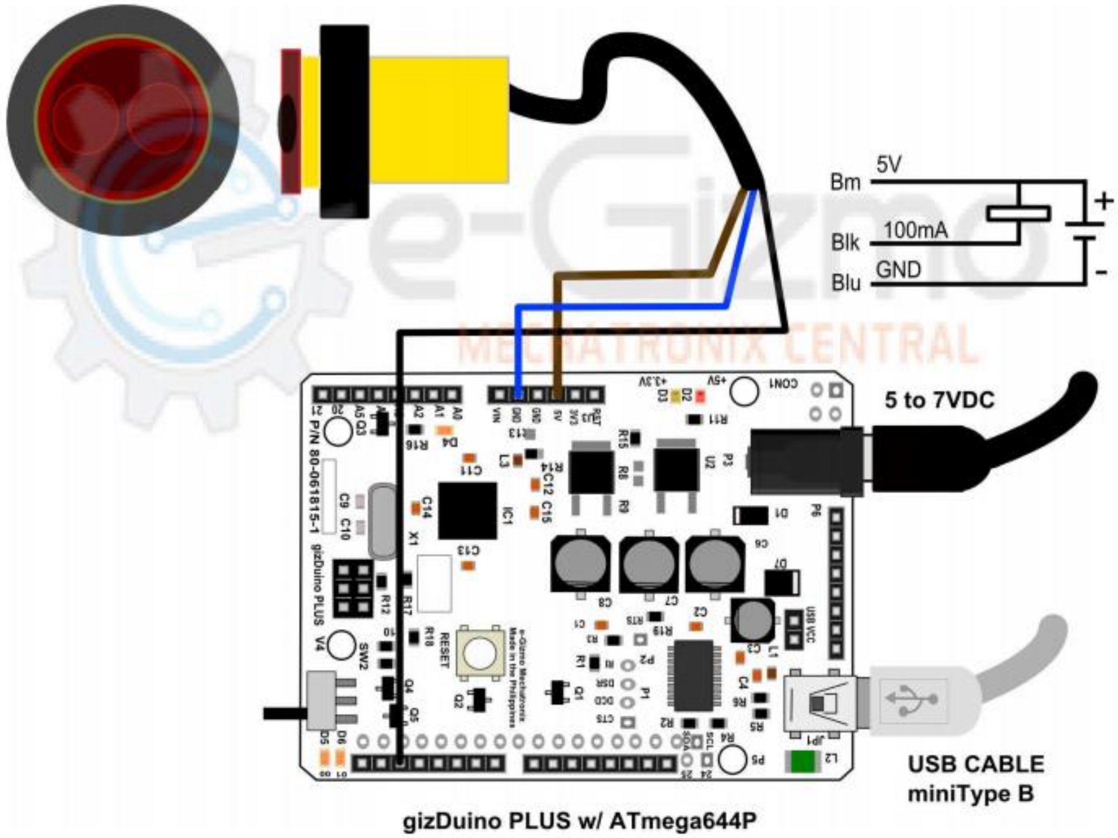

Figure 3-11 ~ Gowoops IR Detector Configuration

Vishay TSSP4056 IR Sensor

The Vishay TSSP4056 caught our eye for a few reasons, but ultimately was discarded. The reason why it was considered by the team was because of its superior detection range of 2 meters and its overall cheap cost. For 10 units, each unit would cost about $0.963. It has an effective pointing angle of 45° and is functional using very low power. The reason why it was ultimately discarded from our list of options is because the detection distance is not modifiable. This would conflict with the overall goal of our project, which is to be able to detect obstacles in front of the shopping cart. Having a non-modifiable distance of 2 meters, no matter how impressive that is, would mean that the cart would be detecting things in its path that aren't technically obstacles, such as the aisles on the sides. Displayed below is a block diagram of how the IR sensor senses an object.

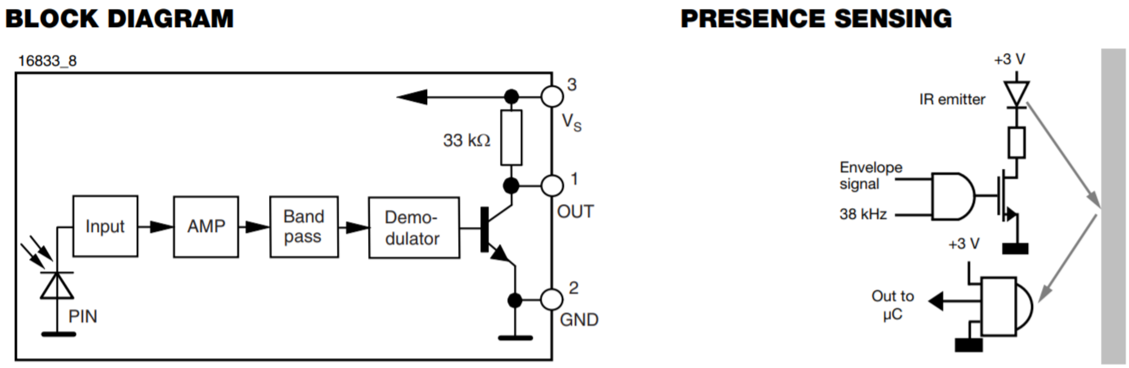

Figure 3-12 ~ IR Sensor Diagram

Osoyoo IR Sensor

The Osoyoo IR Sensor is an option we heavily considered due to its price. It is very low cost to the point where we can get 10 units for $10.00. As for the specifications, the detection distance goes as far as 40 centimeters (1.3 feet/15.75 inches), it weighs only 5 grams, and has an effective pointing angle of 35°. This is sufficient enough for our project because being able to detect a little over a foot is all we need to have the shopping cart stop and take the necessary course of action from there (stop, correct its position, and continue on its path). The detection distance is also modifiable so we don't have to worry about the cart detecting things it shouldn't be. Displayed below is a diagram of how light is reflected off the photodiode onto the IR LED.

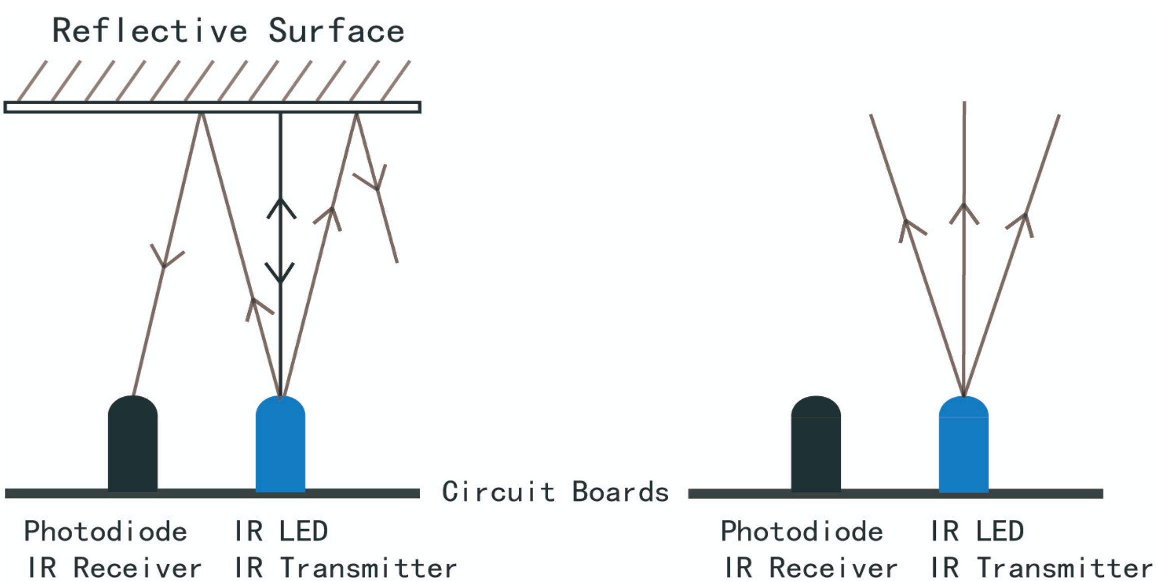

Figure 3-13 ~ Osoyoo Sensor Diagram

3.3.4 Infrared Sensor Comparisons

IR Sensor	Detection Distance	Pointing Angle	Cost (for one unit)
Gowoops E18-D80NK	80 cm	15°	~$7.00
Vishay TSSP4056	200 cm	45°	$0.963
Osoyoo	40 cm	35°	$0.99

Figure 3-14 ~ IR Comparison Table

These three different IR options have similarities within one or more specification. The Vishay and the Osoyoo have essentially the same price. For 10 units together, they're approximately $1.00. The major difference between them is the detection range. The Vishay's detection range goes way past what we would like to implement for our project. It would be detecting the nearby aisles and would cause the shopping cart to stop. Therefore, it will no longer be considered. The Osoyoo IR sensor has the smallest detection range, which may benefit us for the purposes of this project. We want to be able to detect an obstacle right in front of the shopping cart. Along with that, it would be a very cost-effective module to get in bulk. That leaves us with the Gowoops. It is the most expensive option in comparison to the other two and has a decent distance detection range. Since we want to be fiscally economical yet have superior functionality in detecting obstacles, we may or may not select this item.

3.3.5 Sonar Module Options

Our project, in addition to having IR sensors, will also implement sonar modules. This way, our shopping cart can detect nearby obstacles by sound alone. The form of sonar we want to implement is passive sonar. The module will be listening for nearby sounds in order to transmit to the microcontroller which action to take. We have considered the Devantec SRF10, Maxbotix EZ1, and the HC-SR04 sonar modules for our shopping cart. Some of the specifications we will be looking at is the cost, ranging distance, power requirements, and the approximate beam pattern. Listed below are their specifications and what sort of features each would contribute to the project.

Devantec SRF10 Ultrasonic Range Finder

This sonar module has a ranging distance of 3 cm to 6 m with a 40 kHz ping. It is known for being the world's smallest ultrasonic range finder. It only requires 5 V and its beam pattern is the widest out of all the Devantec sonar modules. This serves as an advantage because we want our low power microcontroller to be able to power the sonar module without utilizing a majority of the charge. In terms of cost, one unit of this sonar module costs $33.68. Despite these impressive specifications, cost is a major factor we have to consider and this would sway our decision to use this sonar module. Displayed below is an image of the beam pattern.

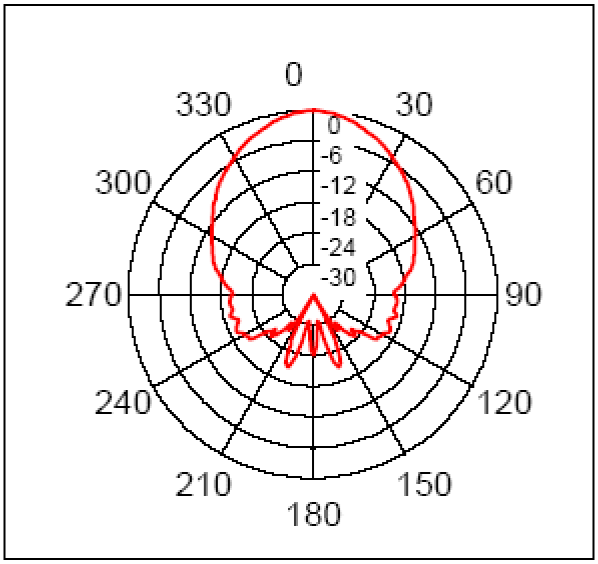

Figure 3-15 ~ Alien Enemies of Samus

Maxbotix MaxSonar-EZ1 Sonar Module

The EZ1 high performance sonar module has the best range out of all the selections. It can detect objects all the way up to 6.45 meters with a 1-inch resolution. This particular module is also compatible with various low voltage sources ranging from 2.5V to 5.5V, a huge benefit for our microcontroller's power requirement. Object detection with this sensor includes zero range objects. The cost of one unit is $24.95. Seeing as how there exists many similarities between this sonar module and the Devantec SRF10 sonar module, the cost is the one differentiator in this case. Displayed below is the approximate beam pattern.

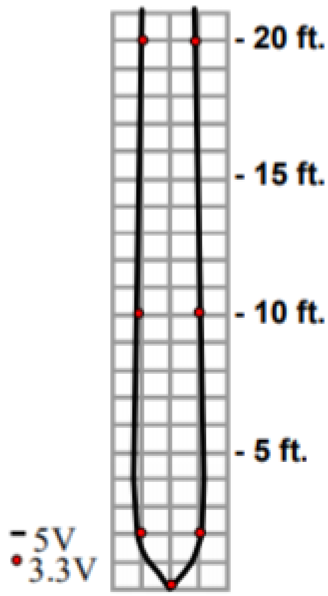

Figure 3-16 ~ How Samus' Hyper Beam Works

HC-SR04 Ultrasonic Sonar Module

This particular sonar module is a personal favorite amongst the team members. It provides the same ranging accuracy and longer ranging distance than some of its more expensive competitors, like the Sharp IR ranging module. The ranging distance goes from 2cm to 500 cm. The cost for one of these is only a mere $2.50. Although the ranging distance isn't as close as the other two sonar modules we were considering, the cost is a huge takeaway. We are most likely going to be choosing this component to add to our shopping cart. Shown below is the beam pattern of this module.

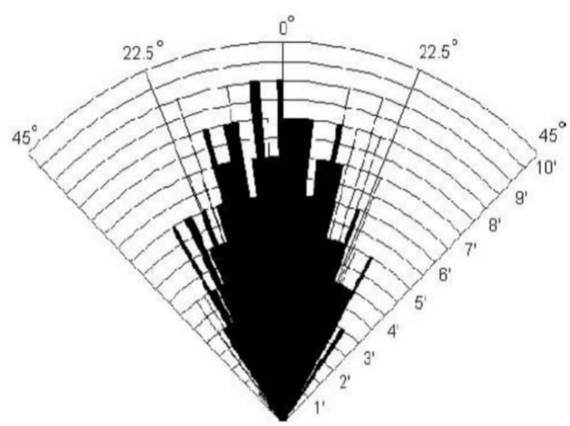

Figure 3-17 ~ Area of Effect for Samus' Hyper Beam

3.3.6 Sonar Module Comparisons

Sonar Module	**Dimension (mm)**	**Power Requirements (V/C)**	**Ranging Distance (cm)**	**Cost (for one unit)**
Devantec SRF10	32x15x10	5V/3mA	3 cm - 600 cm	$33.68
Maxbotix EZ1	19.9x22.1x16.4	5V/15mA	15.24 cm - 645 cm	$24.95
HC-SR04	45x20x15	5V/2mA	2 cm-400 cm	$2.50

Figure 3-18 ~ Sonar Comparison Table

As you can see, the HC-SR04 utilizes the least amount of energy but its maximum ranging distance is the lowest out of all the sonar modules. That isn't a problem for our project because we don't need a high maximum distance range. It is also the cheapest by a huge percentage in comparison to the Devantec and Maxbotix. Although the other two have its advantages, we are leaning heavily towards choosing the HC-SR04.

3.3.7 DC Motor Options

In order for our shopping cart to move autonomously, it must have direct mechanical energy diverted to it somehow in order to propulse itself forward (or backwards). This is where our DC motors come in. A DC motor is a rotary electrical component that converts direct current electrical energy into mechanical energy. In our case, the DC motors will be mounted to the back wheels of our shopping cart, which will make the wheels move forward. In order to make a strategic selection on which DC motor to use, we had to consider a number of specifications. Some of these specifications were the nominal voltage produced by the motor, the no load RPM (revolutions per minute), and the stall current. Other factors we had to consider involved simplicity of use and obviously cost. Listed below are three of the different DC gear motors we highly considered.

Lynxmotion 12V 90RPM DC Gear Motor

This high power brushed DC gear motor has a nominal voltage of 12V, stall current of 10A, and a no load RPM of 3000. Since this specific motor includes an internal planetary gearbox, it will produce an ideal amount of torque needed to move the wheels with less power. This also means that the cart will be able to move with more weight than it could with a simple motor powering the wheels.

The Lynxmotion motor was one of our initial choices in designing Tinjac, only using 1.58A to power on the motor at peak efficiency it will keep our power life several hours longer than some of the other options.

DFRobot 12V 146RPM DC Gear Motor

Also built with high torque and low power usage, the DFRobot motor can also meet our power constraints while also being able to provide the amount of torque a typical load would require from the cart.

However, what sets this motor apart from the other potential options for motors is the gear ratio. Having one of the highest gear ratios of the motors we searched, this motor will be able to get to the speed we require as needed and within an instant. This indicates a significant difference in RPM's within the motor, allowing the motor to be able to rotate more within less time and still a similar amount of power in comparison.

Pololu 6V 1010RPM DC Gear Motor

This motor was an appealing choice because of its smaller size. Included in this difference in size is the amount of power it consumes, which was remarkably less than every other choice we looked into. This would allow for more space to fit the other components on the whole of a system.

With less power comes less reliability. While the benefits of having the motor be more compact and less power-hungry help make the system more efficient, the trade off of having a 1:1 gear ratio causes a system with less torque and less RPMs will severely limit our options of how Tinjac can move effectively.

We kept this motor on as a potential option since it was a cheap motor that also allowed us to create a version of Tinjac that could be more compact and still functional. However, we will likely choose a motor will more power and suffer the costs to do so.

3.3.8 DC Motor Comparisons

DC Motor	Power Requirements (V/A)	Gear Ratio	RPM (Revolution per minute)	Cost (per unit)
Lynxmotion	12v / 0.25	26.9 : 1	3000	$70.40
DFRobot	12v / 0.23A	51 : 1	8000	$45.99
Pololu	6v / 0.275A	1 : 1	1010	$36.95

Figure 3-19 ~ DC Motor Comparison Table

The Pololu motor, as we stated, is a cost-friendly and size-friendly alternative should we decide we needed to conserve space and power. Between the two motors that have a high gear ratio, we were stuck deciding which would be the better fit for our design.

The Lynxmotion motor seemed like a great middle ground for the gearbox ratio (not too high, not too low) and was going to be the motor of choice initially. However, costing $25 per unit more than the DFRobot, which had nearly three times the RPM's, we decided against it and going with the DFRobot for our initial prototype of Tinjac.

3.3.9 Motor Driver Options

Because our project's main functionality is traversal, having an efficient and effective motor driver is the key to successfully having our shopping cart properly move forward, backward, turn, etc. The selected motor driver will be powered by the microcontroller, so the main specifications we will be looking for in our motor drivers is power usage (including average and peak current), size, and cost. The power usage has to be perfect enough to where the motor driver doesn't consume too much power but also has the proper amount of power to drive the wheels forward. Size and cost, just like all the other specifications of our other parts, will also play a huge part in our consideration. Typically, the size of motor drivers is small enough to where it shouldn't affect the weight of the shopping cart, but the cost needs to be cheap enough to where we'll be able to focus our budget on the bigger components of our project.

EasyDriver v4.5 A3976 Motor Driver

This motor driver is a bipolar stepper motor driver that can either work with a 3.3 V or a 5V power system. It requires a 7V to 20V supply to power the motor It has an adjustable current control ranging from 150mA to 750mA. This allows for adaptability in usage and would work well with a microcontroller. The cost for one unit of this motor driver is $5.99, so it wouldn't be contributing too much towards our spending. Below is a functional diagram of the A3976.

FUNCTIONAL BLOCK DIAGRAM

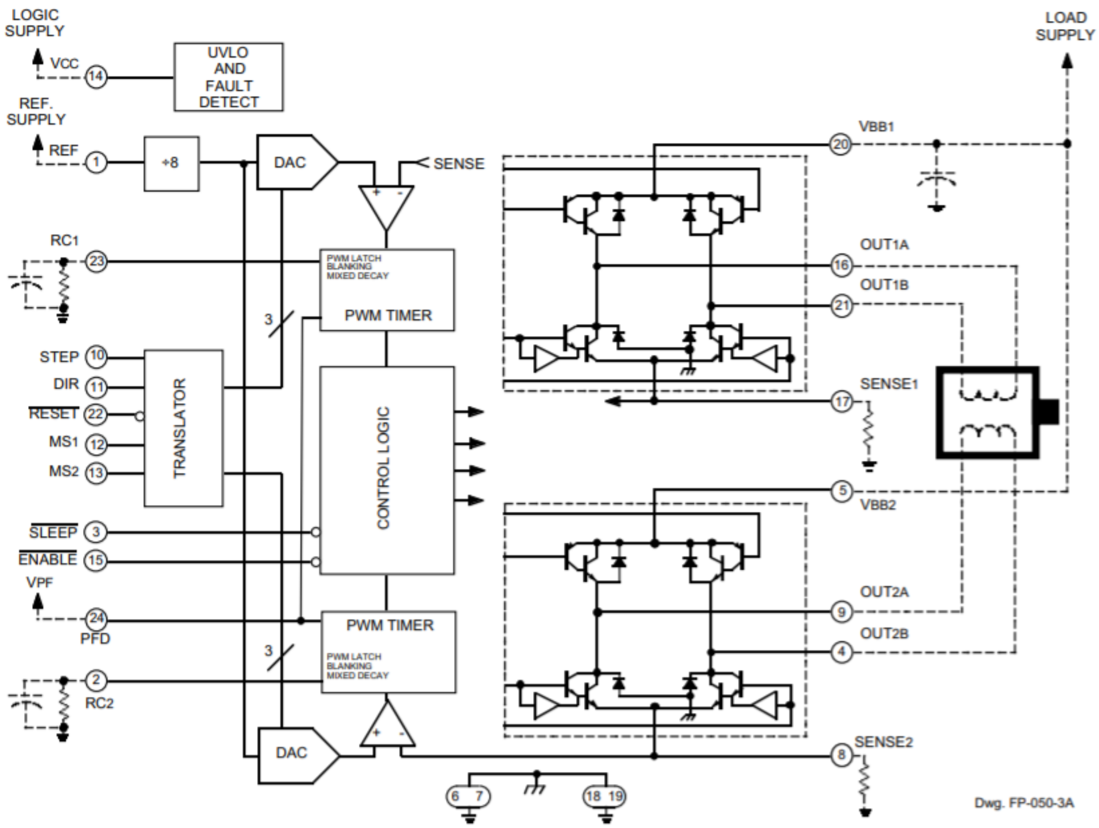

Figure 3-20 ~ Copyright: Allegro Microsystems [10-A]

Texas Instruments L293D Motor Driver

The Texas Instruments L293D motor driver is a quadruple high-current driver. It's voltage range is 4.5V to 36V. This allows for a wide range of power levels that we can consider when connecting it with our microcontroller. It provides bidirectional current to up to 600 mA. In terms of compatibility, the L293D would be most compatible with the MSP432 microcontroller. Given the combination of these components, we are highly considering this motor driver. It would provide enough power to move the wheels of our shopping cart fast enough to keep up with the average speed of how fast a person walks. Another specification of this product that would greatly benefit our budgeting purposes is the cost. For one unit, it is a mere $3.34. Shown below is the functional block diagram of this motor driver.

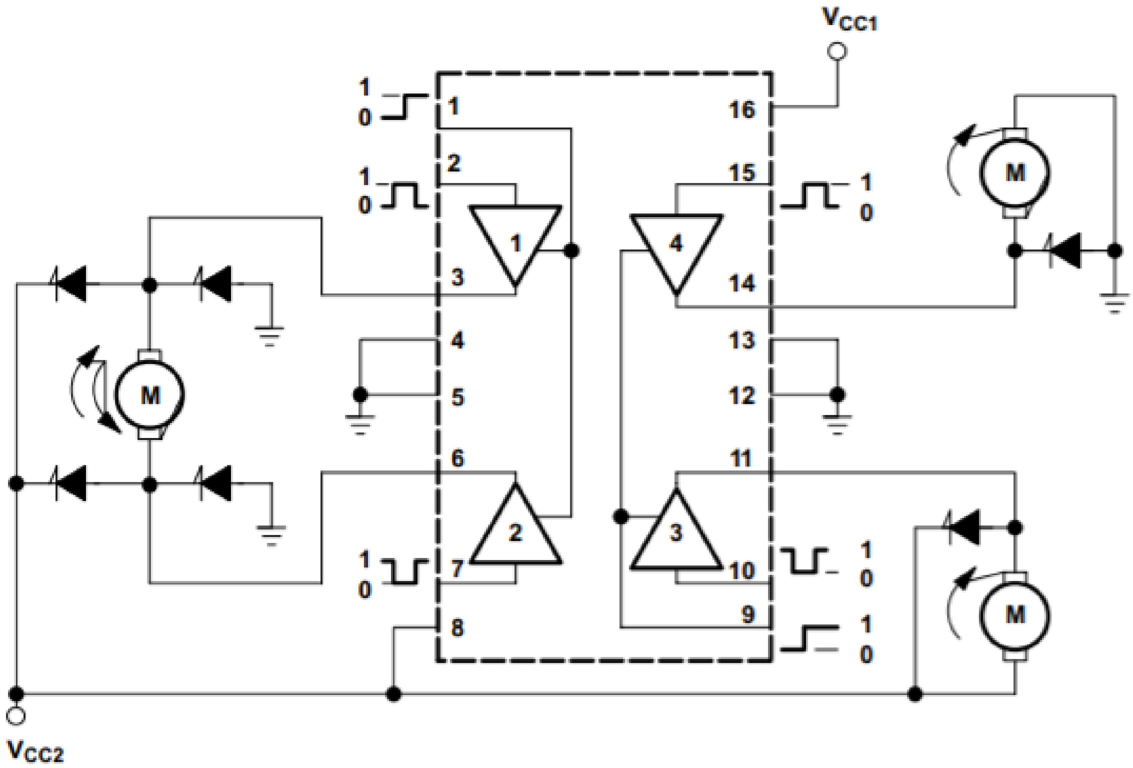

Figure 3-21 ~ TI Diagram [10A]

Toshiba TB6612FNG Motor Driver

This motor driver has a voltage range of 2.7V to 5.5V. It has a constant current of 1.2A and a peak current of 3.2A. It can control up to two DC motors simultaneously by allowing the speed of each motor to be controlled via a pulse-width modulation input signal. This feature would be most useful for applying different speeds of one individual motor for the purpose of turning our wheels on the shopping cart. For example, in order for the cart to make a successful 45° turn, we are planning on having only one of the wheels turn while the other wheel stays stationary (via some sort of braking system). Other than that, the cost of this product is $4.95, which is decent. Below is the block diagram of how this motor driver functions.

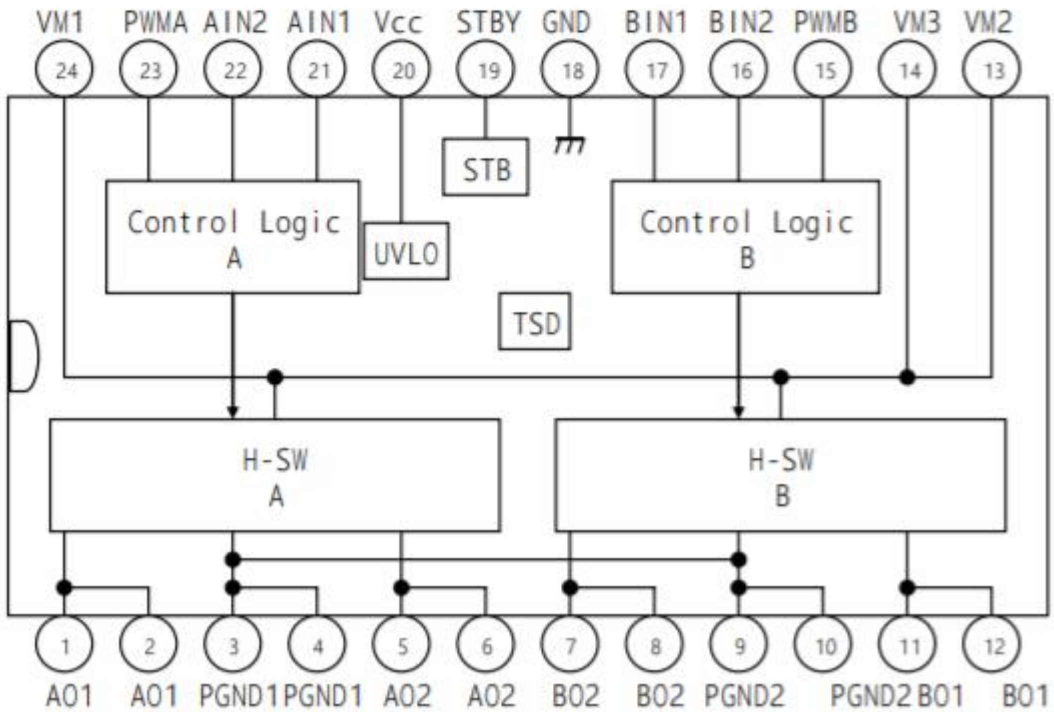

Figure 3-22 ~ Copyright: Toshiba [10A]

3.3.10 Motor Driver Comparisons

Motor Driver	**Power Requirements (V/C)**	**Cost (for one unit)**
EasyDriver v4.5 A3976	7V-20V/150mA-750mA	$5.99
TI L293D	4.5V-36V/600mA-1.2A	$3.34
Toshiba TB6612FNG	2.7V-5.5V/1.2A-3.2A	$4.95

Figure 3-23

As you can see, the EasyDriver v4.5 motor driver is easily the least useful motor driver in this situation. Not only is it costlier than the other two, its power requirements are unsatisfactory. We need a good amount of power to move an entire shopping cart forward. It is highly unlikely that the motor driver would be able to provide that much power for our purposes. That leaves us with the TI L293D and the Toshiba motor driver. The TI L293D requires less power overall to function and it is also the most cost effective with cost. It would also be more convenient to hook it up to a TI MSP microcontroller unit (if that is what we decide to select). In combination with those factors, the Toshiba motor driver would require too much power and that would take away from all the other responsibilities of the microcontroller unit (powering the IR sensors, sonar modules, LCD modules, etc.).

3.3.11 LCD Module Options

This component of our project assembles everything into place. It is the connection between the user and the functionalities provided to that user. The LCD display will be used to display the list of grocery items that the user can choose from. They can add it like normal to a queue before the cart starts moving or even add an item while it's moving. Because of budget restrictions and the need to be economical and affordable, the LCD module we want should be cheap, have a big enough display so that the user can see the words properly, even with vision problems, and have low power requirements. Below are some of the options we have considered.

Parallax Serial #27979 LCD Module

The Parallax serial LCD module is the first LCD module we considered because of the all the group members' familiarity with Parallax BASIC Stamp microcontroller. The size of this LCD screen is 20 (character) x 4 (lines/rows) with 40-pixel characters. This is slightly larger than most common LCD screen sizes, which is 16x2. This allows us to display the grocery items more clearly for those that may vision problems. For the power requirements, it needs 5V to have both the backlight on and off. For one unit of this LCD screen, it costs $29.99. This may be too much for our budget and may be the primary reason we resort to our alternative options, despite the competitive specifications of this LCD.

Winstar WH1602A LCD Module

The Winstar LCD module caught our eye because of its user experience simplicity. The character display size is 16 (characters) x 2 (lines/rows). Normally, the Winstar LCD comes built in with the ST7066 microcontroller, but that is an option we decided not to further pursue. This LCD comes with different interfaces backlight colors. For power requirements, the lowest the input voltage can be being 3V. This requires less power than the parallax LCD module we considered up above and benefits us because our microcontroller will be providing most of the power to the entire shopping cart unit as a whole. In terms of cost, it is $28.38. Once again, this takes a big hit on our budget and would prevent us from contributing to components that are more vital to our shopping cart. Displayed below is the schematic of the LCD screen.

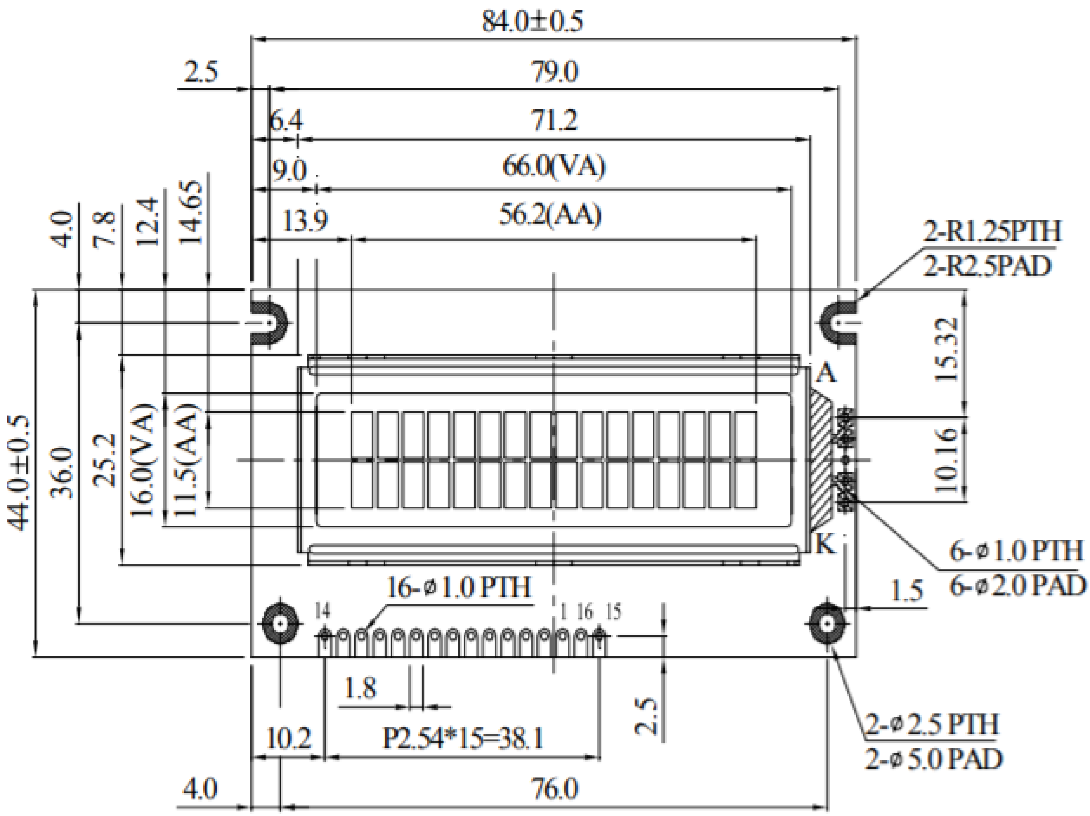

Figure 3-24 ~ LCD Module Diagram

Shenzhen LCD1602 LCD Module

The Shenzhen LCD1602 module is an option we considered because it is absolutely the cheapest out of all the other LCDs we looked at. At a mere $1.25 per unit, it comes with a standard 16x2 display with a minimum power supply requirement of 5V. The pin connection integration on this unit is very minimal and simple if you utilize the module that it comes with. It manages the I2C (inter-integrated circuit) communication and therefore reduces the outputs to only 4 pins. This is most likely the LCD display that we will end up choosing. Displayed below are the block diagrams for the LCD panel and the power supply.

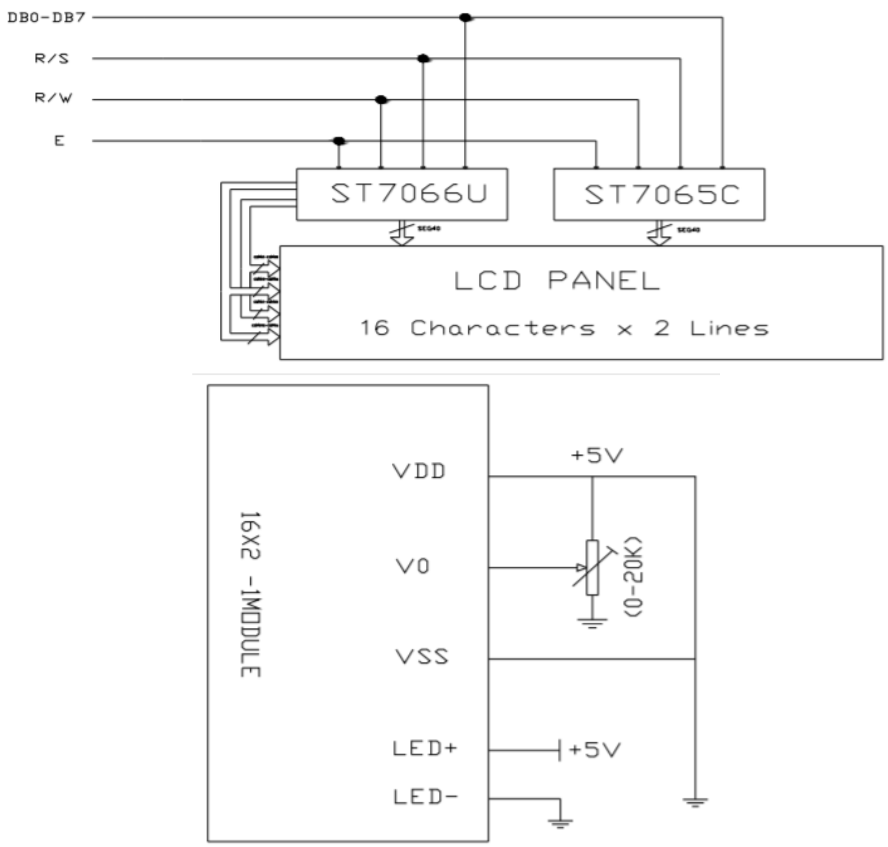

Figure 3-25 ~ LCD Module Schematic [10A]

RioRand LCD Module

The RioRand LCD module is the final one we considered because of its compatibility with the Arduino microcontroller. The size of this LCD screen is also 20 (character) x 4 (lines/rows). This is slightly larger than most common LCD screen sizes, which is 16x2. This allows us to display the grocery items more clearly for those that may vision problems. For the power requirements, it needs 5V to have both the backlight on and off. For one unit of this LCD screen, it costs $7.99. As you can see, despite having various similarities with the Parallax LCD module discussed above, the RioRand is much more cheaper. This is perfect for our budget and may be the LCD module we end up selecting in the end.

3.3.12 LCD Module Comparisons

As you can see, the Shenzhen LCD1602 Module has all the other alternatives beat when it comes to price. Although the Parallax and RioRand LCD have the superior display dimensions, the RioRand is far cheaper than the Parallax. Therefore, the Parallax LCD will no longer be considered. The Winstar LCD is the same size as the Shenzhen with a lower power requirement but it is substantially more expensive. We are now presented with two viable options: the RioRand and the Shenzhen. This is a typical trade off situation where we can either give up quality or cost. In section 3.5 is where we'll declare our official selection of our parts.

Sonar Module	**LCD Dimensions (char x line)**	**Power Requirements (V)**	**Cost (for one unit)**
Parallax #27979	20 x 4	5V	$29.99
Winstar WH1602A	16 x 2	3V	$28.38
Shenzhen LCD1602	16 x 2	5V	$1.25
RioRand	20 x 4	5V	$7.99

Figure 3-26 ~ LCD Module Comparison Table

3.3.13 Solar Panel Module Options

Listed below are three of the best solar panel modules we have considered. Seeing as how solar panels will most likely be the most expensive part of our shopping cart unit, we will be considering cost as the number one factor in our decision as well as other specifications. Some of these other specs will include cell type, number of cells, power parameters, and cost.

For cell type, there will be two different types of cells that we will be looking for: monocrystalline and polycrystalline. The main differences between the two is efficiency and cost. Monocrystalline solar panels are more expensive but have the highest efficiency rates because they're constructed from high-grade silicon. Polycrystalline solar panels are the total opposite, being much more cheaper and having approximately only 14-16% efficiency due to the low amount of silicon purity.

Due to the scope of our project, these specifications will play a decisive role in our selection since our overall goal is affordability. Cost will end up being the number one decider in what we want for our shopping cart. Listed below are the three options that we considered.

Sunnytech Mini Solar Panel Module

This solar panel is smaller compared to typical solar panels as its being used to a smaller extent. It is made up of polycrystalline silicon cells with a max voltage of 5V, a max current of 250mA, and a max power output of 1.25W. The number of cells it contains is 10. It comes with a blocking diode already installed to prevent the panel from overcharging and current backflow. This particular solar panel was designed for smaller scale solar projects so its cost is a mere $9.99. For the purposes of our project, this solar panel is ideal enough to be able to power some of the other components that our shopping cart will have.

Qlhshop Solar Panel Module

This solar panel is an easy to install, maintenance free solar panel that is mostly used in cars as a backup battery charger. It can charge 12V batteries and produce 5W of power. Just like the Sunnytech solar panel described above, it also comes with a built in blocking diode in order to prevent reverse discharge. It's max power voltage is 17.9V and its max current produced is 300mA. As similar as these specifications are compared to the Sunnytech described above, the biggest difference would be their cost. The Qlhshop solar panel costs $23.99. This factor in itself is probably enough for us to veer more towards the Sunnytech. Another thing that may sway our decision away from this selection is the fact that this solar panel is used mostly for cars. Therefore, it probably won't be able to power all of the different components of our shopping cart.

Dasol DS-A18-10 Solar Panel Module

This option is one our most expensive options due to the quality of this particular brand of solar panels. The Dasol off-grid solar modules offer utilize a heavy duty anodized aluminum frame and utilizes polycrystalline solar cells for the base. It can produce a maximum power of 10W, twice the amount of the standard max power given by the smaller solar panels. It's maximum power voltage is 18V and it's maximum power current is 560 mA. As you can see, this solar panel module is far more superior to the other two options we considered above.

Some of the drawbacks though include the high cost of this particular unit, its weight, and its size. It costs $39.99 with a possibility of being charged with 30% tax credit (due to the recent tariff imposed which is discussed in detail below in the constraints section). This would take a huge hit on our budget and would limit us from buying other important components. In terms of weight and dimension, the solar panel is a whole four pounds and its dimensions are 14.6 x 9.8 x 0.7 inches. This is the heaviest and biggest solar panel we considered and will probably not be chosen due to these three factors. It's size is too big to fit efficiently on our shopping cart (which will have to be pretty small in itself due to budget constraints). This directly correlates for the need to have the solar panel also be light enough (in terms of weight) in order to not limit the shopping cart's movement. Given different circumstances, this would have been a perfect option in order to power our shopping car's microcontroller, motors, LCD module, IR and Sonar sensors, etc.).

3.3.14 Solar Panel Module Comparisons

As is displayed, the Sunnytech solar panel module is the cheapest out of all the options. Although it's power specifications aren't as competitive as the Qlhshop or the Dasol, it might be just enough to power our shopping cart along with each of its components. The Dasol is the most expensive and the most powerful in terms of the amount of power it can produce, but its size and weight will ultimately dissuade us in selecting it. The Qlhshop is a nice option to have in the middle. It produces four times the amount of power as the Sunnytech and also produces a much greater voltage, matching up with the Dasol's voltage output while still having a cost difference of $16.00.

Solar Panel Module	**Cell Material**	**Power Requirements (W/V/C)**	**Cost (for one unit)**
Sunnytech	Polycrystalline	1.25W/5V/0.25mA	$9.99
Qlhshop	Polycrystalline	5W/17.9V/0.3A	$23.99
Dasol	Monocrystalline	10W/18V/0.56A	$39.99

Figure 3-27 & 3-28 ~ Solar Panel Comparison Table and Cost Measurements

3.3.15 Solar Charger Controller Options

Solar charger controllers regulate the amount of power that is received from solar panels to the batteries that it charges. This keeps the batteries from overcharging and causing damage to themselves. A charge controller isn't always needed but is highly urged to consider for solar panels putting out more than 2 watts. Because our project consists of powering up several small components simultaneously, we will need to produce more than 2 watts of power in general and would therefore need a charger controller to ensure the safety and functionality of our components and our shopping cart as a whole.

There are two different types of charge controllers: MPPT (Maximum Power Point Tracking) and PWM (Pulse Width Modulation). The only difference between these two types are their respective charging efficiency rate and their unit cost. MPPT controllers are more efficient than the PWM controllers. They have an average efficiency rate of 90% while a decent PWM controller has a maximum efficiency rate of 80%. But with such advantages, there will always be drawbacks. The cost of getting an MPPT controller would obviously be its higher price, making PWM controllers the more inexpensive option when constrained by a budget.

Some of the specifications that we'll be considering, other than cost and power requirements, will be its max charge/discharge current, operation temperature, size, and weight.

Anself Solar Charge Controller

This is a very simple charge controller in that it doesn't have any special features. This device contains six ports to connect the battery, solar panel, and the output voltage, as well as three small lights to indicate the status of the battery and 2 lights for the status of the controller. This particular product isn't very expensive, costing a mere $9.99 which comes with an option for 10A charging. To upgrade to 20A charging, we would only have to pay three dollars more. For the sake of our projects requirements and design specifications, a 20A feature would be unnecessary. Overall, it would be very well suited for our backup solar energy system within our shopping cart.

Figure 3-29 ~ Solar Charge Controller

ALLPOWERS Solar Charge Controller

This controller has more features than the previous controller mentioned above such as it has two ports for USB charging, and an LCD display. This LCD display shows the status of the battery and charger as well as showing what voltage the battery is at. A handy feature of this product is that it has timer settings for if the user only wants their battery to be charging at a specific time during the day or the user can set a timer to charge for only a few hours. In terms of its specific power features, its output voltage can either be 12V or 24V, depending on what you pay for. It's rated charge current and rated loaded current is 20A. Displayed below is a simple design on how this controller would be implemented with solar panels. For our case, the solar panels would be connected to our battery that will be powering our shopping cart.

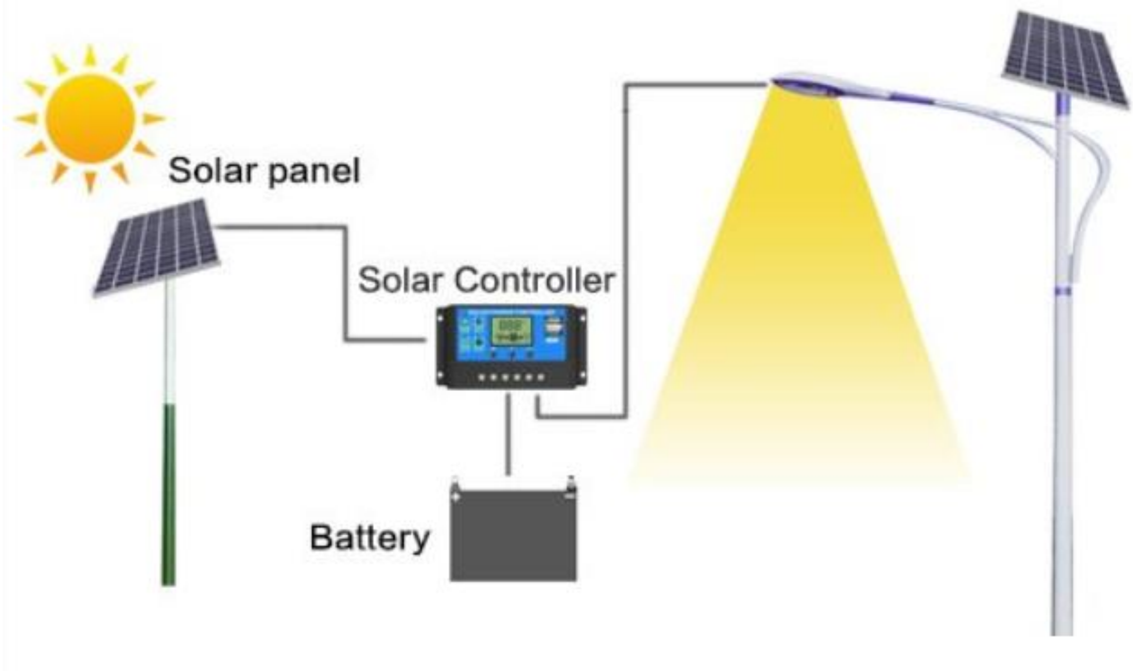

Figure 3-30 ~ Solar Charge Controller Functionality

HQST Solar Charge Controller

This controller also has an LCD display but only has one output USB port instead of two. The controller automatically compensates for temperature and the way it charges can be adjusted by being able to take into account different types of loads as well as being able to set a max voltage for charging. The device is also able to show amps using the amps metric. Its rated working voltage is 12V/24V along with a rated working current of 10A.

Figure 3-31 ~ HQST Solar Charge Controller

Y-Solar Solar Charge Controller

This controller takes a different design than the other products as it takes a more rectangular casing. It has a charge/discharge current of 40A, which would provide plenty of current for our shopping cart. What sets this controller from the other ones is that it's waterproof which would actually be very useful since it's being used for solar charging. With how often and randomly it rains in Florida, someone might not be ready when disaster is about to hit, especially if it's being left outside to charge without any protection. Purchasing this product also includes a 12 month warranty.

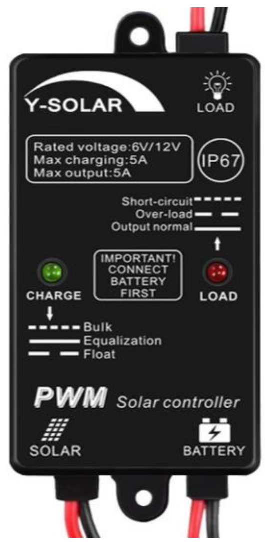

Figure 3-32 ~ Y-Solar Solar Charge Controller

3.3.16 Solar Charger Controller Comparisons

Out of the four solar charger controller options we considered, there were numerous specifications that we recorded and had to observe in order to make an educated selection. Our prime factor was whether or not our solar charger would be compatible with whatever solar panel we decided to select. But listed below are several specifications of each of the products, such as weight, price, size, power, etc.

The first specification is weight. With our shopping cart unit, we need to consider the weight of all our components in order to ensure that movement does not get hindered in any way, shape, or form. This is why weight must be kept to a minimum, given that other specifications don't end up lacking or falling behind due to this strict designation. The second specification is cost. Obviously, we want the cost of our entire shopping cart to be as minimal as possible, while still being able to retain all the desired functionalities that we need. This will only be achievable if we keep our costs to a minimum. The third specification that we will need to keep in mind is the size of the solar charge controller itself. Just like all the previously stated specifications, it should be as small as possible. Our shopping cart unit will not be able to fit, yet alone carry, all the components necessary for it to be a fully autonomous shopping cart. Therefore, all the space that we can spare will contribute to the placement of other, more important, components. Our last important specification is the power requirement. We want the solar charge controller to efficiently regulate the amount of power received from the solar panels in order to charge our battery. Displayed below is a table of the four controllers and its many specifications.

Charge Controller	*Anself*	*ALLPOWERS*	*HQST*	*Y-Solar*
Weight (grams)	117 grams	204.117 grams	99.2 grams	33 grams
Cost ($)	$9.99	$20.99	$15.99	$12.99
Size (in)	4 x 3.74 x 1.5	5.6 x 3 x 1.7	4.1 x 2.8 x 1.42	3.15 x 1.46 x 0.87
Voltage (V)	12 Stop supply: 10.8 Resume supply: 12.6	12V	12V	12V
Current (A)	10A	20A	10A	5A
Pro	Price	Good UI	Accurate	Waterproof
Con	Simple design	Ports used to insert wires aren't secure	Small screen	Only voltage reading on screen, no amps
Temperature (Celcius)	-20 to 60	-35 to 60	-30 to 60	-10 to 40

Figure 3-33 ~ Solar Charge Controller Comparison Table

As you can see, there are many similarities and differences amongst these capable solar charge controllers. The Anself is the cheapest option but it is not the lightest or the smallest. The Y-Solar charger controller costs only $3.00 more and it weighs the lightest and also has the smallest volume. The ALLPOWERS solar charge controller is better in terms of discharge/charge current but is unnecessarily large. This means that no matter what we choose, we will have to sacrifice either cost, weight, or size. Our option for this product will be revealed in the components selection section.

3.3.17 Rechargeable Battery Options

Listed below are three different options of rechargeable batteries that we considered during the planning phase of our project. All of these rechargeable batteries have similar specifications, with a few minor differences. The most basic specifications are obviously the power requirements, which tend to be similar throughout all the different models we've looked at. We will be breaking the minor differences down, picking out which specification will work best for our project, and go about strategically selecting a complete rechargeable battery. Some of the minor differences in specifications that we will be analyzing are the battery's weight, dimensions, and cost. Due to the scope of our project, the dimensions and weight of the battery will greatly affect the way our shopping cart moves. Overall, the battery we will be selecting will meet the requirement need to power all the components of our shopping cart.

ExpertPower EXP1270 Rechargeable Lead Acid Battery

This seal lead acid battery is low self-discharge and maintenance-free. It produces 12V and 7A. You can use it for various different applications such as powering an alarm system, medical equipment, communication equipment, and security systems. This is perfect for us to use in order to power our microcontroller unit along with the other components of our shopping cart. Because it is also rechargeable, we won't have to worry about replacing it. In terms of weight and dimensions, this battery weighs 4.3 lbs and is 5.9 x 2.5 x 3.7 inches. As you can see, this battery is proportionately heavy with it's volume. For our shopping cart, the weight of the battery might affect the speed of our cart, seeing as we need it to travel the average walking speed of the human being (1.4 meters per second (m/s), or about 3.1 miles per hour (mph)).

PowerSonic PS-1280-F1 Rechargeable Battery

This seal lead acid battery is valve regulated, has absorbent glass mat, and has spill proof construction. It produces 12V and 8A. As you can see, the power requirements of this battery are very identical to the EXP1270 battery described above. In terms of dimensions, the PS-1280 is 6 x 2.6 x 3.7 inches, essentially a duplicate of the EXP1270. The biggest notable difference though is the weight, which is 5.6 lbs. As you can see, this battery is an entire 1.3 lbs heavier than the EXP1270. This is a specification that will greatly affect our decision in selecting a battery.

CSB HR1234WF2 Rechargeable Lead Acid Battery

This rechargeable battery produces 12V and 9A, very similar to the rest of the other batteries. There are essentially very minor differences, seeing as how most of these rechargeable batteries have the same dimensions and weigh about the same, give or take one pound. The CSB battery specifically weighs 5.5 lbs and has dimensions of 6 x 2.6 x 3.7 inches. The pricing for this battery is also around the same, with an average of about $20.00.

3.3.18 Rechargeable Battery Comparisons

Rechargeable Battery	**Power Requirements (V/C)**	**Weight (lbs)**	**Dimensions (inches)**	**Cost (for one unit)**
ExpertPower EXP1270	12V/7A	4.3 lbs	5.9 x 2.5 x 3.7	$19.99
PowerSonic PS-1280-F1	12V/8A	5.6 lbs	5.9 x 2.5 x 3.7	$18.00
CSB HR1234WF2	12V/9A	5.5 lbs	6.0 x 2.5 x 3.7	$22.99

Figure 3-34 ~ Rechargeable Battery Comparison Table

As previously stated, all of these batteries have very minor differences. They all have the same dimensions, so that won't be an issue because all of them are small enough to fit into our shopping cart. The PowerSonic and the CSB batteries both weigh the same at about 5.5 lbs and the ExpertPower weighs an entire pound less than both those two batteries. Therefore, in terms of weight, choosing the ExpertPower battery would be the best decision. But if we were to base our decision on cost, we would select the PowerSonic and would have to accomodate having that extra pound burdened on our shopping cart. Overall, the two best batteries for us is the ExpertPower and PowerSonic. Our decision will be stated in section 3.5 "Parts Selection Summary".

3.3.19 Shopping Cart Options

Last but not least, the primary component of our project is the shopping cart itself. Due to the lack of sponsorship, we were unable to get a retail grocery store to provide us with a cart (such as Publix or Walmart). This isn't that big of an issue because we wanted a smaller than usual cart in order to build our prototype and go about improving it once we figured out all the functionalities. Therefore, we resorted to considering small size shopping carts from online retailers. All of the shopping carts are very different in terms of size, amount of material it can hold, and the design of the cart itself. Due to the nature of this component, the specifications that we will be analyzing are very physical in nature. We'll be looking at the overall design, loading capacity, weight of the cart, and determining how well the shopping cart can traverse. Described below are some of the options that we considered from various different online retailers.

Advance EXpress 6000 Two-Tier Shopping Cart

Figure 3-35 ~ Xpress Shopping Cart

As displayed, The EXpress6000 two-tier shopping cart has two different baskets for customers to be able to put more of their grocery items on and even separate them however they feel. In a sense, it is a mixture of a hand basket and a larger traditional shopping cart. This shopping cart weighs 60 lbs, has a loading capacity of 150 lbs, and also has a short wheelbase which makes maneuvering around narrow aisles of a store much more easier. The wheels also have a non-marking feature so that the floors of the store do not get scratched. Another helpful addition to the shopping cart is the smaller basket in between the two large ones that protrude outwards. This shelf could possibly be used to store all of our major electrical components such as the microcontroller, LCD screen, and perhaps even the rechargeable battery. Despite all these features, cost will once again be the ultimate decider. This particular shopping cart costs approximately $210.00 (depending on the particular model you get). This price would negatively affect our budget to a great extent, seeing as how we’d have to succumb to having a lower quality in our other, more important, components.

Sandusky FSC4021 Folding Shopping Cart

Figure 3-36 ~ Sandusky Shopping Cart

This utility shopping cart, unlike the Xpress6000 mentioned above, can be folded for storage. It is made out of steel and is resistant to wear and corrosion. It has a maximum load capacity of 110 lbs while the actual cart itself weighs 11 lbs. The wheels are made out of rubber so they cannot go flat. The dimensions of the cart are as follows: 40 x 21 x 25 inches. With a mere cost of only $25.96 per unit, this option seems much more viable for our project. For the sake of building our prototype, this shopping cart is light and portable enough to carry a great amount of items. The only difficulty in this is making all of our components fit onto the cart due to its small size. A majority of our time would be spent trying to figure out how to place the electrical components onto the cart itself. Also, If we design it correctly, we could incorporate all our electrical hardware components to not be affected by the folding of the cart.

Goplus Double Basket Folding Shopping Cart

Figure 3-37 ~ GoPlus Shopping Cart

The Goplus folding shopping cart is a lightweight and sturdy folding utility cart that is unique from the other two options described above because it comes with swivel wheels. It weighs approximately 13 lbs and the dimensions are: 24.4 x 24 x 40 inches (L x W x H). It is also made out of premium, heavy duty, durable, and stable metal. The wheels are made out of rubber and the front swivel wheels allow the cart to turn much smoother and with a tighter turning radius than the standard shopping carts that are available in retail grocery stores. It also comes with two baskets so that the customer may pick out additional items. It also makes our job easier because we can utilize that additional basket to install our electrical components on it. The cost of this particular cart is $49.99. This shopping cart is a perfect combination of the first and second carts.

3.3.20 Shopping Cart Comparisons

As you can see below, there are a few tradeoffs with each of the shopping carts. The Advance Xpress shopping cart is obviously much more expensive than the rest. This is primarily because of the weight of the actual cart. It is made out of sturdy metal and comes with two additional baskets, with one extra basket protruding outwards in the back. It’s loading capacity is also the highest. But due to the price itself, this option may be out of consideration. The Sandusky is the cheapest option (and therefore might already be the best option) and is small enough to function well as our prototype model. It is also the lightest with a loading capacity of 110 lbs. The Goplus is an excellent competitor to the Sandusky. They are very similar when it comes to their features, but the Goplus has swivel wheels, which will make turning this shopping cart combined and integrated with our motors an excellent candidate.

Shopping Cart	Loading Capacity (lbs)	Weight (lbs)	Dimensions (LxWxH inches)	Cost (for one unit)
Advance Xpress	150 lbs	60 lbs	29 x 21.5 x 41.25	$210.00
Sandusky	110 lbs	11 lbs	21 x 25 x 40	$25.96
Goplus	120 lbs	13 lbs	24.4 x 24 x 40	$49.99

Figure 3-38 ~ Shopping Cart Comparison Table

3.3.21 Magnetic Pickup Sensor Options

This component is more of a minor part of the design. The purpose of having a magnetic pickup sensor is to be able to know whenever a wheel makes a full revolution. this would be used to track the distance being traveled even more accurately. How a magnetic pickup works is that it when a metal object is passed through it’s magnetic field, it will emit a small pulse as a result of modulating the flux density. This part is made by wrapping a coil around a permanently magnetized pole, which would then be connected to the MCU which could see whenever a pulse is given off. For this to fully work, a small piece of metal would need to be added to one of the wheels on the cart to be detected by the pickup. This simple technology is used in cars to track the mileage on them. This device will be connected directly to the MCU, with the ground wire going to the ground pin, and the voltage wire going to one of the empty input or output pins on the MCU.

Some of the specifications that we looked at when conducting our research included the size of the MPU (magnetic pickup), weight, output at cranking range, and measuring range. Described below are two different magnetic pickup sensors that we considered for our project.

Cummins 3034572 Magnetic Pickup Sensor

This particular magnetic pickup sensor was mostly considered due to its wide availability within online stores. These components in general are somewhat pricey, selling at an average price of $30.00. The cost of this particular unit is $29.00. It weighs 3.2 ounces so it won’t be too much of a burden on our shopping cart. With its price, it’ll definitely affect our budget, causing us to either expand our maximum allocation of financial resources, or spend less on some of the other components. The measuring range for this magnet ranges from 50-5000 Hz. Its proximity to gear teeth is 0.028 - 0.042 inches, which should cooperate well with our shopping cart wheels, allowing us to measure the distance it travels quite well. The voltage output during the cranking speed is 1.5 volts.

MSP 676 Magnetic Pickup Sensor

As previously stated, the reason this MPU speed sensor was considered was simply because of its availability in the online stores that we looked at. The price of this MPU is $34.80 and the component weighs about 3.5 ounces. Its proximity to gear teeth is 0.025 - 0.035 inches, having a narrower range than the MPU mentioned above. The output voltage during the cranking speed is 1 volt. Some of the advantages of using the MSP 676 is that it is tougher than most plastic sensors and can surmount semiconductor devices at high temperatures. You may also have it customized to whatever configuration fits your demands.

3.3.22 Magnetic Pickup Sensor Comparisons

Shown below is a table summarizing the different specifications of these products and how we decided to weigh the advantages and disadvantages and finally come to decide on which component to select.

Shopping Cart	Proximity to Gear Teeth Range (inches)	Weight (oz)	Cost (for one unit)
Cummins MPU	0.028-0.042 inches	3.2 ounces	$29.00
MSP 676 MPU	0.025-0.035 inches	3.5 ounces	$34.80

Figure 3-39

As displayed, both of these magnetic pickup sensors are very similar to each other. The biggest difference is the cost of each product, a minor $5.00 difference. Selecting these products will ultimately come down to each of our preferences.

There is no real difference in these specifications and because this component is only a minor part of the project, selecting a magnetic pickup sensor won't negatively or positively affect us to a great extent. As for the selection, because the Cummins MPU is cheaper and weighs slightly less than the MSP 676 MPU, it will most likely be the one chosen.

3.4 Architectures and Diagrams

Displayed below is the ideal design of Tinjac and how we essentially imagine it to look like once everything is put into place.

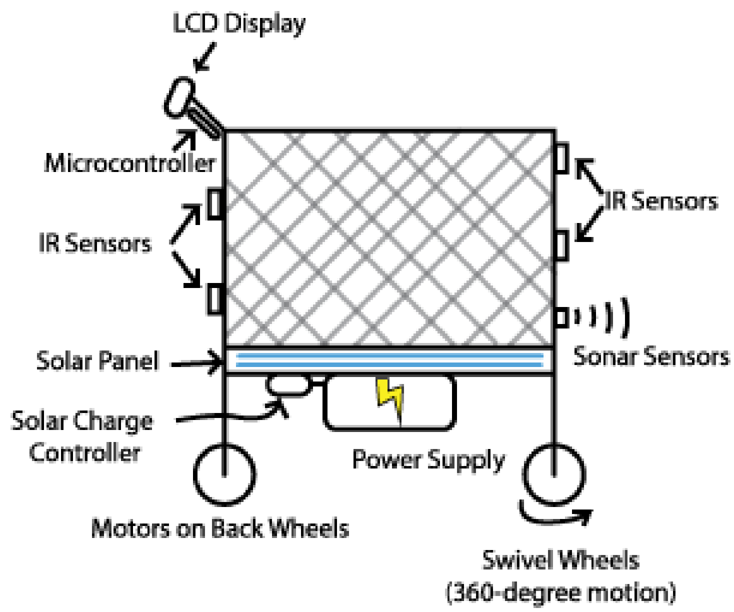

Figure 3-40 ~ Tinjac Design

Tinjac will be utilizing a microcontroller to run the code which is a large part of this project. Since we are working on the assumption that the floor plan and were items are located is known and we have accurate measurements and dimensions of the store the code will need to accurately measure distance travel so that it can reach the destinations provided by the users. It should also be able to calculate the shortest path to gather all the items. The major way the user will interact with the Tinjac is to use the LCD display which will allow users to add and remove items to their shopping list. The possible design that the team is considering to build the Tinjac focus on using two different types of sensors for obstacle detection which IR sensors and sonar sensors. Once an obstacle is noticed by Tinjac it will proceed to travel around the object in the way and continue on its path to the next item on the list. Another thing the sensors will be used for it detect when the user steps away from Tinjac so that it knows to stop and wait for the user to return. The power supply that Tinjac will use is a 12V battery that we intend to have solar panels to charge it. In order to prevent overcharging the battery and potentially damage it a solar panel charger control will be used which should stop charging the battery once it is completely charged. This battery will be used to power everything on the Tinjac including the motors.

3.5 Parts Selection Description

Tinjac is composed of numerous parts, each critical in the system's overall success and efficiency. The major components involved in each of our subsystems we debated over other, similar models, until we reached an agreement in using the listed parts in the previous section.

Some of the factors that contributed to our decisions were memory capacity, power characteristics, cost, detection distance (for the IR and sonar modules), and overall weight and size. With these specifications in mind, we were able to strategically select which components would best fit the needs of our autonomous shopping cart.

3.5.1 Texas Instruments MSP430F5529LP Microcontroller

In order for Tinjac to represent more than just a cart, an optimal brain must be resolving all the necessary instruction handling as efficiently as possible. This specific microcontroller balances out our need for certain features while still maintaining an acceptable level of power consumption. When the device is active, it uses 80µA/MHz which will allow our power source to allow continuous supply to our system. The microcontroller also has 24 independent input channels, which effectively allows us to code each separate module as its own extension of the system, rather than group "similar" detectors as a single input. In combination with its clock frequency at 48 MHz, we will be able to run all necessary movement and detection functions in a timely manner.

3.5.2 Gowoops E18-D80NK Infrared Sensor

In regards to safety, the eyes of the system are just as important as the brain itself. Using IR sensors, Tinjac will be able to detect obstacles at certain distances and send a signal to the microcontroller to alert the system of nearby boundaries. The Gowoops sensors function as most IR sensors would. The distance that it can detect an object can be set and once it has detected an object within its perimeter, a signal voltage will be sent to the microcontroller. Using multiple IR sensors, the accuracy will greatly improve. Some of the other factors that were considered included the effective pointing angle and cost. The effective angle is important to make sure that the IR sensor does not pick up anything that wouldn't be considered an obstacle (such as the aisles on the sides). This would be the case if the effective angle was too large, such as with the Vishay TSSP4056 IR Sensor.

3.5.2 HC-SR04 Ultrasonic Sonar Module

The human body has multiple senses to alert ourselves of nearby danger, Tinjac also uses different types of sensing modules to be able to detect potential collisions. When using the HC-SR04 sonar modules, once the distance has been set to our needs using the control pin, the device can detect nearby obstacles based on sound alone. In combination with the installed IR detectors, the system as a whole will be able to detect neighboring obstacles with a high rate of success. We believe the HC-SR04 sonar module will perform these feats successfully due to its adjustable ranging distance that can go up to 400 cm. Along with its low power consumption, the sonar module will be implemented in various parts of the shopping cart. This is due to the sonar module’s low cost of only $2.50 which allows us to buy more of these units and make our shopping cart that much more precise in detecting obstacles.

3.5.3 12V DC Motor 146RPM w/Encoder

This DC motor was chosen through the website, robotshop, with their help of an online tool that helps choose out the correct motor called, Drive Motor Sizing Tool. This tool lets the user know the required torque required for the motor which is what motors are generally chosen by, this and the rotations per minute (RPM) that the motor outputs. Since the cart was made with the intention of going walking speed, we used the average walking speed for the RPM calculation.
3.1(mph) * 5280 (feet/miles)*12(inch/ft)/60*pi*9 (diameter of wheel)=115.78 RPM. The torque required was 556 oz-in which is exactly what this motor uses. The RPM isn’t exact, but it’s close enough that it could be adjusted through the motor driver, and it’d also be able to account for people who walk faster than the average walking speed. The fact that this motor comes with an encoder is an even greater bonus as this would make the speed of the motor easily adjustable. With two of these motors on the cart, with their high torque, they would definitely be able to handle heavy amounts of weight in the shopping cart.

3.5.4 Texas Instruments L293D Motor Driver Chip

The TI L293D Motor Driver Chip will be used to trigger the motors to move and turn. This will allow the motors to use a completely separate power source from the one used to power the microcontroller, which has a variety of benefits including longevity and circuit overload prevention. Our design for Tinjac includes only 2 motors, allowing a single motor driver chip to be able to control both of the motors. Depending on the signal sent from the microcontroller, the motor driver will trigger the motors to either move (forward or backwards) or rotate (left or right). With its wide range of power requirements that were considered, the shopping cart will also be able to maintain the average walking speed of 1.4 meters per second (3.1 MPH). This was also chosen due to its low price in comparison with the two other motor driver options we had in mind, along with having the ease of compatibility of connecting with the TI MSP430 microcontroller.

3.5.5 Riorand LCD Module

Tinjac will use the RioRand LCD module in order to directly communicate with the user, displaying lists of available shopping stops as well as any prompts to inform the shopper. This can also be used to indicate the status of other subsystems of Tinjac, such as the powering system (and if there is a need for a recharge relatively soon). This particular LCD module was chosen due to its relatively affordable price, costing $9.99 per unit. The size of the LCD display is 20 characters x 4 lines, which is typically larger than the standard and shouldn’t pose a problem for the user in terms of being able to read all the characters on the display with no difficulties. Another thing we had to consider was how simple the implementation would be to integrate it with our TI MSP430 microcontroller, despite us originally considering to use this LCD because it had the best compatibility with the Arduino Uno Microcontroller. Most LCD modules come with a complicated pin connection integration process but the RioRand module manages the inter-integrated circuit communication and therefore reduces the outputs to only 4 pins.

3.5.6 Sunnytech Mini Solar Panel Module

The Sunnytech solar panel module is made up of polycrystalline silicon cells with a max voltage of 5V, a max current of 250mA, and a max power output of 1.25W. The number of cells it contains is 10. It comes with a blocking diode already installed to prevent the panel from overcharging and current backflow. The reason this was chosen was because of its cost, which was a mere $9.99. Although the other options would have provided a higher maximum power output, they would have cost a lot more. Also, some of the power specifications of the other options would have been too much for the purposes of this project. This is why we decided to go with a much smaller solar panel.

3.5.7 Anself Solar Charge Controller

We decided to go with this specific charge controller because of its simplicity in usage, design, cheap cost, and minimum weight. This controller device contains six ports to connect the battery, solar panel, and the output voltage, as well as three small lights to indicate the status of the battery and 2 lights for the status of the controller. At a mere $9.99, we are able to regulate the amount of power that is received from solar panels to the batteries that it charges. The anself charge controller itself weighs 117 grams. Despite being the second heaviest in all our options, the weight of the controller was still light enough to not make a huge impact on the overall weight of our cart. Although this device isn't necessarily needed, we didn't want to risk our components malfunctioning because of any overcharge damage. The solar panel we plan on using will probably be putting out more than 2 watts so we decided to spend the extra money and stay on the cautionary side.

3.5.8 ExpertPower EXP1270 Rechargeable Lead Acid Battery

With so many hardware components involved in our shopping cart, a rechargeable battery was necessary in order to meet our initial requirements. It needed to be rechargeable so that we wouldn't have to worry about replacing the battery every time it died, thus saving us time and money in the long run. Our group decided to select the EXP1270 battery because of its overall mass, dimension, and price. Out of all the options, the battery weighed only 4.3 lbs. This was over one pound lighter compared to the other two batteries we considered. The weight matters to a substantial degree because our shopping cart can only hold a limited amount of weight and having the electrical components taking up most of that loading capacity would take away from the shopper's ability to put an extensive amount of grocery items on the cart. This would then defeat the main purpose of our project. The dimensions of the battery didn't matter because they were all the same size, but having a small sized battery played an essential part in keeping our space on the cart as available as possible. To conclude, the EXP1270 battery can produce 12V with a discharge current of 7A and the material surrounding the battery has a strong resistance to shock, vibration, and heat.

3.5.9 Goplus Double Basket Folding Shopping Cart

This particular shopping cart was selected because it was essentially a mixture of the other two alternatives. The Goplus has an additional basket and is also able to fold. The additional basket allows us more flexibility when installing all our hardware components. We won't have to worry about placing the components onto the cart itself when we can simply utilize the extra basket as a placeholder for the parts. Along with these benefits, the weight of this cart is also significantly light, weighing at about 13 lbs while maintaining a loading capacity of 120 lbs, nearly ten times its own weight. Another benefit for us are the swivel wheels. Because we want our shopping cart to turn as efficiently as possible when traversing an aisle, having swivel wheels allows us not to worry about the front wheels when we implement the turning of the back wheels. Therefore, this is the best option serving as the base of our autonomous shopping cart.

3.5.10 Cummins 3034572 Magnetic Pickup Sensor

One of Tinjacs main features is to be able to traverse autonomously. To be able to do that, we needed a way to measure the distance that the cart needs to go. The best (and cheapest) way to do that was to implement a magnetic pickup sensor within our shopping cart. This particular magnetic pickup sensor was chosen due to its availability, cost, and weight. When searching for the most optimal MPU to use, the only option we kept coming across was the Cummins 3034572. Compared to the alternative, it was also the cheapest and weighed slightly less. This will allow us to optimize our weight distribution on our shopping cart and won't affect our budget to such a great extent.

3.6 Parts Selection Summary

In order for us to deliver a complete and functioning system, we must construct proper functioning subsystems. These subsystems are composed of individual parts that harmonize to accomplish the objective of each of their respective subsystems.

While deciding which specific parts we would use in each mentioned subsystem, we had to take into account several factors before every choosing which part would work best for the goals we have for the subsystems. After our long debated conversations, we settled on specific models that we all agreed would be best for what we would want our project to accomplish.

Figure 3-41 and Figure 3-42 show the specific parts that we finalized on our choices. In Figure 3-41 it is apparent which of the main components we chose for the system as a whole, but neglects some other (less impactful) parts such as the cart frame itself. Figure 3-42 is the specific details of the model and functionality of the specific core components we decided to go with. This is listed to help provide insight to why we chose these specific models over some of the competing choices. Once a fully-functioning prototype is designed we may alter these decisions and adapt them to meet realistic restraints that we did not foresee.

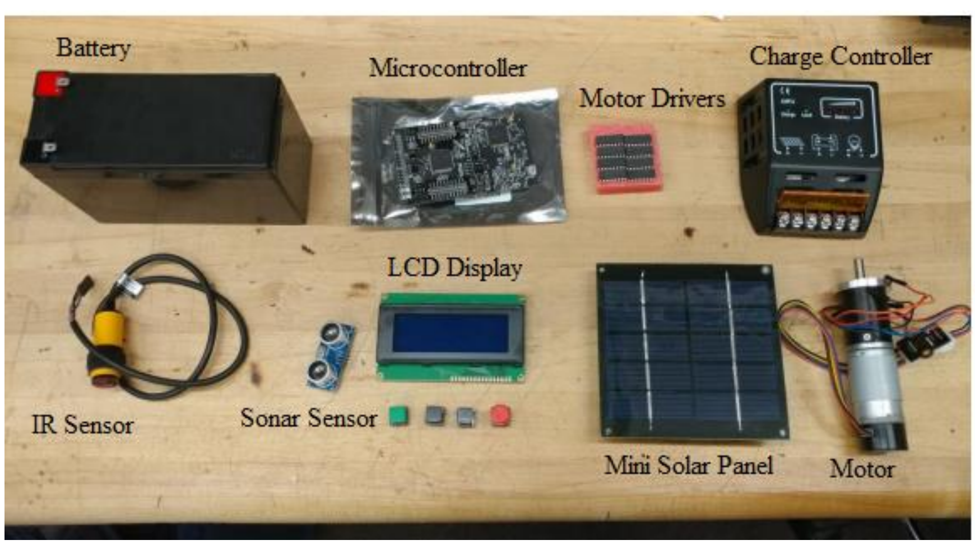

Figure 3-41

Part	Description
ExpertPower EXP1270 Battery	A lead acid battery that is capable of being recharged.
MSP430F5529LP	The microcontroller being used to control the whole system.
L293D Motor Driver	A 16 pin motor driver that will be used to help regulate the two motors.
Anself Charge Controller	The charge controller that was chosen to be able to regulate the power coming from the solar panel to the battery without overcharging.
Gowoops E18-D80NK	The infrared sensor module to be used to detect objects. There will be 7 of these placed on the cart.
HC-SR04 Sonar	The ultrasonic sonar module to be used to help detect motion and the lack of motion. There will be 2 of these placed on the cart.
Riorand LCD	This display will be used to act as a user interface when paired with some buttons.
Sunnytech Solar Panel	A small solar panel capable of charging the battery chosen.
12V 146RPM DC Motor w/Encoder	This motor was chosen based on the high torque it has and the low rotations per minute to meet walking speed.
LMZ12002 Voltage Regulator	This voltage regulator steps the 12V output voltage to 3.3V
TPS563210ddf Voltage Regulator	This voltage regulator steps the 12V output voltage to 5.0V

Figure 3-42

3.7 Realistic Impacts in Design Choices

An idea cannot simply be declared “good” when viewed only through one perspective, which is why Tinjac is forged from the unanimous agreement that every subsystem and subsequent feature would be a benefit in several different ways. Tinjac is designed to appease multiple sources of criticism, doing so allows us to worry less about theoretical long-term issues.

We had to also consider the practical implementation every time we even thought about adding on a new feature to the cart, asking questions like: “Would shoppers actually want this?”, “Would this actually be cost effective in the long term?”, “Is this something just we would want, or would most consumers enjoy this feature?”, and more.

3.7.1 Economic Influence

Perhaps one of the largest hurdles for a project to see its birth into the market, a proper ratio of cost versus effectiveness will always play a role in the free market. We made sure that every single module was the cheapest option, while also allowing us to have the relevant subsystem running at an acceptable accuracy.

We set out to make Tinjac work efficiently, while also being composed of the cheapest components we could find that would still allow the cart to function properly. This stopped us from using other modules with more exquisite features, such as a more complex microcontroller as well as hyper-accurate sonar modules.

While deciding on which IR sensor to use, we also considered a relatively cheaper module. However, while discussing this move as a group we chose to investigate the major differences between the two modules and discovered that the cheaper model lacked the distance and control that we needed to create a safe and effective cart. Even though they are the same module, simply because a module is cheaper does not mean that all the functionality would remain the same.

3.7.2 Consumer Want

A product could be design as cost-effective and as intricate as possible, however, if the customers of the free market see no interest in the design then it would be an inevitable failure. From the beginning, Tinjac was designed to set out to please the consumer. Every decision we made was contested upon with the ideas of what the consumers may want in a smart-cart.

We had numerous ideas that we wished to implement into Tinjac, but upon further discussion we decided that a shopper would find it more monotonous rather than convenient. One of these ideas, of many others, was to create a UI system that would recommend the shopper what to purchase. This could take into account a diet of some kind, or financial status, or even randomness. We decided that since Tinjac sought out to make shopping an easier experience, adding extra steps to be able to shop would be more often a hassle than not.

3.7.3 Safety-Conscience Decisions

Tinjac could run perfectly 99.99% of the time, however in that 0.01% chance of error we must ensure that a situation could not cause serious damage to the users or others.

The largest impact that safety has on Tinjac lies within its software, as we have added numerous lines of code just in the microscopic chance that everything else fails. The physical components of Tinjac are used to create a safe path for Tinjac to traverse, and in the event of a collision being detected Tinjac can avoid it or create a new path altogether.

This would satisfy nearly every possibility of Tinjac running into something or someone, however we also designed Tinjac with extra safety-related function in case its standard functions should fail. In the end, if every other safety net should fail, Tinjac can initiate a total shut-down and simply call it quits. While this is not an ending we would wish to happen with our product, without it another option could be colliding with an unsuspecting person.

4 Related Standards & Design Constraints

Engineering standards are documents that stipulate characteristics and technical specifics that must be met by the invention. The persistence for developing and obeying standards is to safeguard the least possible efficiency while meeting safety prerequisites and ensuring that the invention is consistent. Different kinds of engineers must adhere to certain standards depending on their field and certain protocols must be followed when designing such products in order to avoid liabilities that could occur if standards are not met. For example, an electrical engineer with a focus on wireless networks should be fully aware of the IEEE 802 standards which specify on how to properly deal with local area networks and other types of area networks. This section covers materials related to standards utilized in the design and development of Tinjac. It also covers realistic design constraints that affect the design of Tinjac.

4.1 Related Standards

Information about related standards regarding new fangled technologies and applications dealing with electronics, power systems, and anything electricity related can be found with the IEEE (Institute of Electrical and Electronics Engineers) Standards Association. This is a subdivision within IEEE that develop standards in the power and energy, biomedical, telecommunication, transportation, and nanotechnology industry. Because our project is primarily hardware and electrical related, we will be discussing the related standards and constraints that we must adhere to in relation to our autonomous shopping cart. Some of the standards used in the design of Tinjac is ISO/IEC 9899, IEC 61215, and IEC 62368-1.

4.1.1 ISO/IEC 9899

ISO/IEC 9899:2018 also known as C11 is the most updated standard for the C language. It specifies the syntax and constraints of the C language, the representation of input and output data processed and produced by C programs, and the limitations imposed by the implementation of the C language. This standard is what most current and up to date compilers use to compile C code. This is related to our design of Tinjac because C is the coding language that is being used to write the code for Tinjac. This coding language is being used because all of the group members have the most familiarity with it, having taken courses where C was utilized such as Intro to C, Digital Systems, Computer Organization, and Embedded Systems. In the case that the electrical engineers also need to get involved in the coding process, they will be able to contribute to a certain extent and not hinder the computer engineers in the group.

4.1.2 IEEE 830

This standard is known as the IEEE Recommended Practice for Software Requirements Specifications (SRS). these standards lay out a template on how stakeholders, business entities, developers, etc. can describe, understand, and specify exactly what they want for their particular software/hardware product. This standard essentially establishes a solid foundation for agreement between the clienteles, providers, and how their product will function. Section four of IEEE 830 lays out how to produce a good SRS, where there are six different phases consisting of the nature, environment, characteristics, evolution, prototyping, and design of your project. The nature of the SRS is to describe in detail the goals, functionality, performance, qualities, and design constraints of our project. The environment of the SRS is to describe where exactly it fits in the hierarchy of the project. Characteristics of a good SRS should lay out exactly how complete, concise, and unambiguous the requirements are. The evolution phase should describe how your product will need to undergo any changes throughout the development and designing phase of the overall project. This is why the SRS should be modifiable to an extent while also not being too broad. Prototyping allows for the development team to determine the product's progress as well as addressing any technical issues that may have been caught. The prototype of the product should allow the team to determine if the requirements of the product 's functionality has been met or not and what they can do to change this. Last but not least, the final design layout allows the team to focus on the external behavior of the product.

With our autonomous shopping cart, Tinjac, in mind, we will be utilizing these concepts to guide us in building our first prototype. Realistically, we will come across technical issues that may force us to change some of the design concepts of Tinjac but appropriate steps will be taken to ensure that our primary motivation and goals for this project do not get compromised.

4.1.3 IEC 61215

This standard is essentially a qualification test that tests how durable solar panels are. It tests many different outside environment, so a solar panel that follows these standards are more likely to be to stand up to outdoors use, less likely to have design flaws and they will suffer little to no degradation in power output in the test environments. This relates to the Tinjac design because it utilizes solar panels charge the power supply. Ideally when the user is done using the Tinjac it will be outside so it can charge while not in use. This ensures that the solar panel will not break due to the weather.

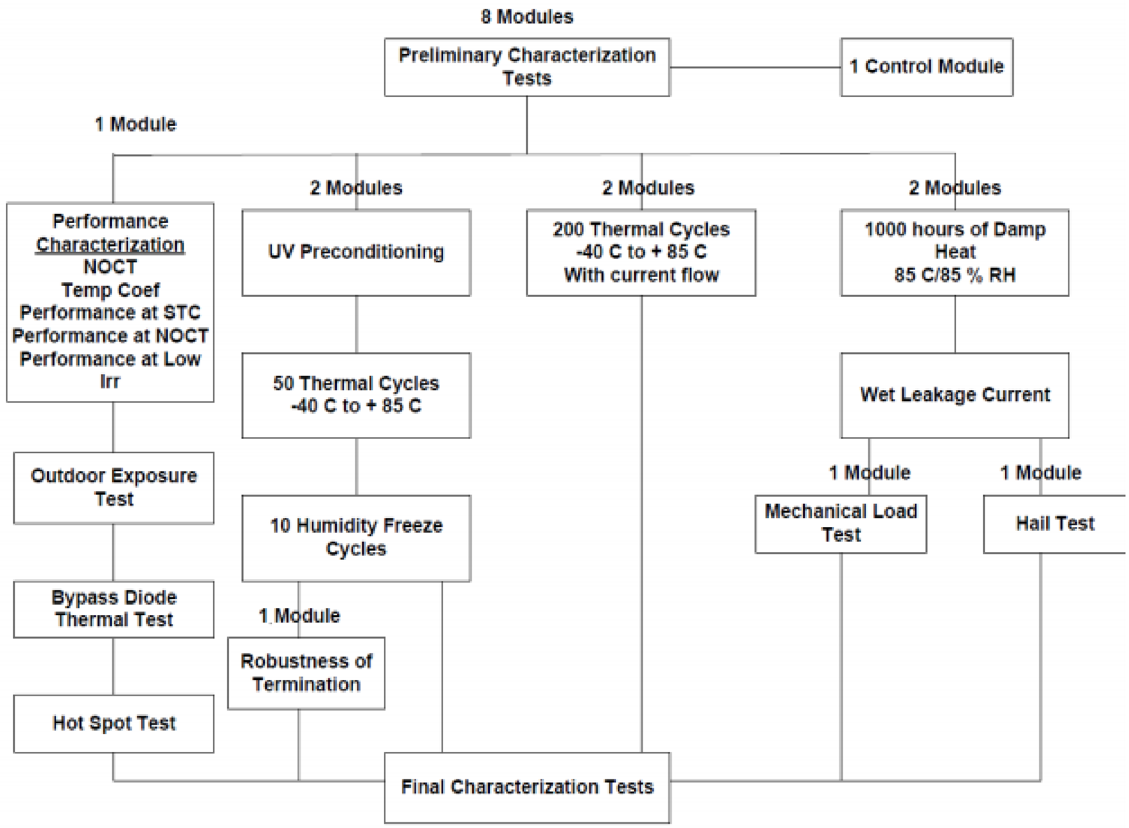

Figure 4-1 ~ [10C - 7]

4.1.4 IEC 62368-1

This is a safety standard for power supplies that most electronic equipment follows. The goal of this standard is "identify potential hazards as energy sources capable of causing pain or injury to others, while finding ways to prevent such energy transfer" [7]. This applies to Tinjac's design and is being used in the development of Tinjac as a safety measure for the user to prevent from any potential hazards including electrical shock and flammability.

4.1.5 Bluetooth Standard

With one of the stretch goals of the project being to design an app to run the cart which would be able to accomplish various task and help with accessibility to the end users of Tinjac, the app must be able to communicate with the microcontroller on the Tinjac. This can be accomplished through Bluetooth technology which is a standard that is constantly being updated by Bluetooth Special Interest Group (SIG) and is being used more and more as time continues to pass and the Internet of Things idea and technology continues to grow. The Internet of Things is basically the idea of a bunch of local devices communicating with each other. The most recent version of the Bluetooth standard that was found is called Bluetooth 5, which was used in devices such as the IPhone 8 and IPhone X, which includes all other previous Bluetooth standards and some more improvements. This communication of devices using this standard is accomplished by using short-wavelength radio waves in the frequency range of 2.4 GHz to 2.485 GHz. Following this standard will also allow for higher data transmission speeds and a low power mode so that when the app is not communicating with the Bluetooth module and will not consume as much power. This will in turn increase the battery life of Tinjac because the the Bluetooth module will only being using the minimum amount of power it needs to detect when the Bluetooth module needs to communicate with the microcontroller on the Tinjac.

4.1.6 PCB Design Standards

There are many standards related to the design and purchasing of a PCB. One of the first standards to take a look at is the IPC-60111 which is a standard to determine the class of the PCB.

The class determines what the PCB will be used for and how sophisticated the PCB needs to be. The classes are range from one to three and the higher the class the more sophisticated and tolerance for imperfections become stricter. For the purposes of the Tinjac class 1 should be considered since it is a consumer product we are designing and we can allow some imperfections in the PCB board.

Next standards to take a look at would be IPC-4101, IPC-9691, IPC-2141, IPC-2251, and IPC-2152. These are used to determine the appropriate material for the PCB to made of, the material of the CAF (Conductive Anodic Filament) and will help take into consideration the impedances, frequency and amperes that are involved with the PCB board when designing it. As for the rest of related PCB standards related to design they are shown in the figure 4-2 which is below with a title that describes what that standard relates to.

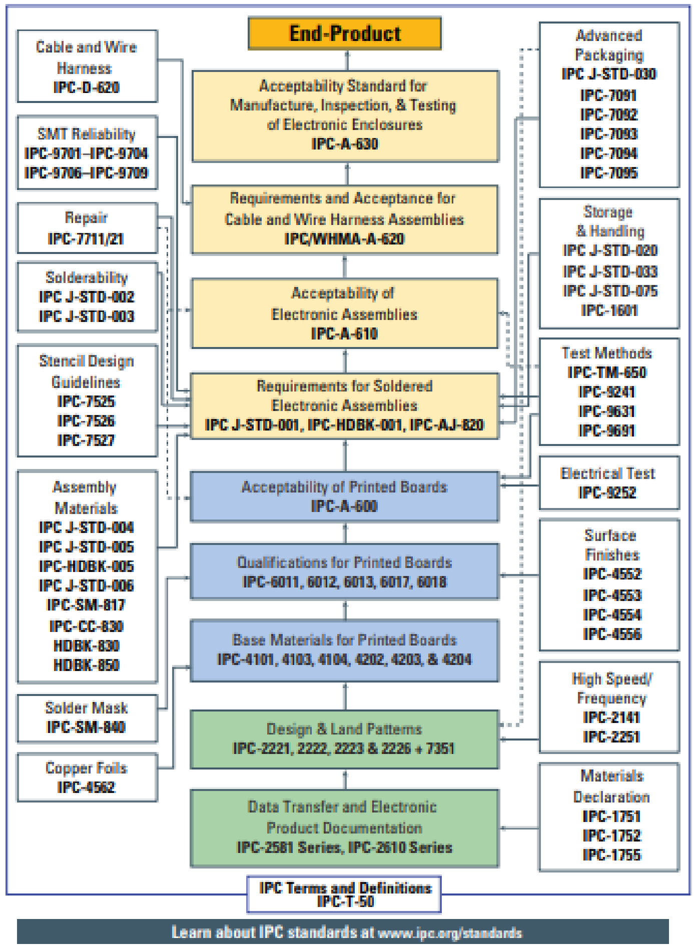

Figure 4-2

4.2 Realistic Design Constraints

The most challenging portion of managing a wide scale project is maintaining the design constraints that come with that project. Just like projects, there exists several different constraints of any product that have ever existed and they are all dependent on one another. In order to produce the most optimal version of your product, you must consider the risks, resources, quality, time, cost, and scope of the product.

Most of the time, these constraints are reciprocally connected. Therefore, a hinderance upon one of the constraints will definitely affect one or more of the other constraints. An example of this can be seen in a typical quality assurance scenario. If the QA team determines that the quality of a software product has not met deliverable standards, more resources will be needed, which will proportionately affect cost and therefore also increase the time it takes to roll out the product. In relation to our autonomous shopping cart, we will be discussing the constraints that have an impact on the design of our project. This includes constraints that are economic, time, environmental, social, political, ethical, health and safety, and manufacturability and sustainability related.

4.2.1 Economic and Time Constraints

Economic constraints limit what parts or components are used in the design of the project. Since this project is not sponsored by a company, the group has decided the budget to be around $400 to $500 to develop and build one Tinjac unit with all the sensors, batteries, motors, and etc. included in the design. With this limited budget, some components that may be ideal will have to be passed up on for more affordable, less advance, pieces of technology. One clear compromise made because of the limited budget is that the cart used is not a full-sized shopping cart that would normally be used in say a Publix or Walmart. Something that also came up when looking at the cost for certain components was that because of a recent policy solar panels imported received a higher tariff which increased the prices of solar panels that would have met the specifications we were looking for.

Along with having a lack of sponsorship as a primary economic cost constraint, the goal of this project is to keep costs as low as possible so that affordability is a feature of our product. Realistically, we would want stores to adopt this technology without having to worry about cost as an issue. In return, this would benefit customers greatly.

The time constraints that are applicable to this project are due to the structure of the class. This affects how research and development is handled for the project. The total time allotted for the project is the summer semester and fall semester which is from May to December which is about 8 months' time to research and develop one working Tinjac unit model. This means it should meet all the required specifications stated in the above sections and if possible accomplish any of the stretch goals.

4.2.2 Environmental, Social, and Political Constraints

Environmental constraints are put in place to help sustain the current environment. Most of Tinjac's design does not impact the environment greatly but even so some steps have been taken to reduce the effects it will have on the environment. The battery in the system is has to be rechargeable so we opted to use solar panels to charge the battery instead of plugging it into some power station.

As for social constraints, the objective of the Tinjac is to make an easy to use self-driving cart that can be easily operated by the user. For this purpose, we decided to use LCD display and a couple of buttons to add/remove items from a list and scroll through a list to make it easy to understand.

With stretch goals in mind we also decided to include an app feature if we can which will also be easy to use and clear about what each thing does. Since it is a self-driving it could accommodate all types of people whether they be healthy individuals, elderly folk, or handicapped individuals. For this purpose, we set intended to set the speed to the average walking speed of a person and also have the cart detect when the user is a certain distance away so it should never race off ahead and leave the user behind.

The one of the major political constraint that could be considered is the potential to affect people’s jobs. We do not believe this to be an issue though since it is only a small task this cart will be fulfilling by being able to drive itself back to a designated location instead of the employee working at the store manually collecting all the carts and bringing them to a designated location.

Something that also came up while searching for solar panels to use and the prices was that the price for the same solar panels were increasing and was discovered that due to some new policy, there was a tariff placed on all solar panels imported into the country which in turn could affect which solar panels we are willing to buy if the prices keep increasing. This could also affect other parts as well in the design and was something that was not taken into consideration before and therefore could start to limit what we can actually buy and use to complete the design of Tinjac.

4.2.3 Ethical, Health and Safety Constraints

For the ethical constraints, they kind of go with social constraints discussed previously. We are trying to make this technology easily accessible and easy to use by anyone. Some other implications it could have is potentially affecting people's jobs since it is removing the need for to employees to collect the carts after they used. As time goes on things start to become more automated which in turn also removes the need to employ people for jobs that can be easily automated.

The health and safety constraint mainly involve the sensors on the Tinjac. The major safety issue involved with any self-driving thing is the possibility of it potentially hitting someone or something. To avoid this, multiple sensors around the front and sides would be used to help it avoid any possible collisions. Other safety issue occurs with any electronics using rechargeable batteries their proclivity to catch fire. Since this is using a solar panel to recharge the battery a solar panel charger controller will be used to stop recharging the battery when it is done.

4.3.4 Manufacturability and Sustainability Constraints

Manufacturability is the ability for the design of Tinjac to be easy to mass produce. This means that items and components that are hard to acquire reduce manufacturability and therefore become a constraint. In our design of the Tinjac unit, we try to use technology that is not overly complicated and more likely to exist in stock. This means that it is easier to produce and that makes the cart easier to mass produce in return. This also makes getting replacement parts easier because individual parts are so easily available.

When selecting the components for the project, it noticed that certain things were out of stock from the manufacturer's website. Despite having specifications that could have greatly benefited our Tinjac unit, it would be useless if the manufacturer themselves did not have the part in stock.

Sustainability is the idea that this device should be able to last as long as possible without needing replacement parts or completely breaking down. To help maximize the sustainability of Tinjac, all electrical components should be covered to reduce possible damage that may occur.

5 Project Hardware and Software Design

This section details the main schematics for the project. Also included is an overall schematic that will be used to create a PCB with, as well as schematics that give more detail on the individual subsystems.

The second half of this section involves the software design of how the functions perform. It lists all the objectives of each individual function, and includes a layout of the basic flow of function for each method.

These sections are the cornerstone for Tinjac's functionality. Composed of almost half hardware and half software, Tinjac requires a tightly tuned and optimized setup for its subsystems. Each decision when designing the hardware subsystems and the code to implement those same subsystems were made with care and deepful insight. In order to minimize mistakes we had to make sure to attempt to do it correctly the initial attempt.

5.1 Initial Design and Related Diagrams

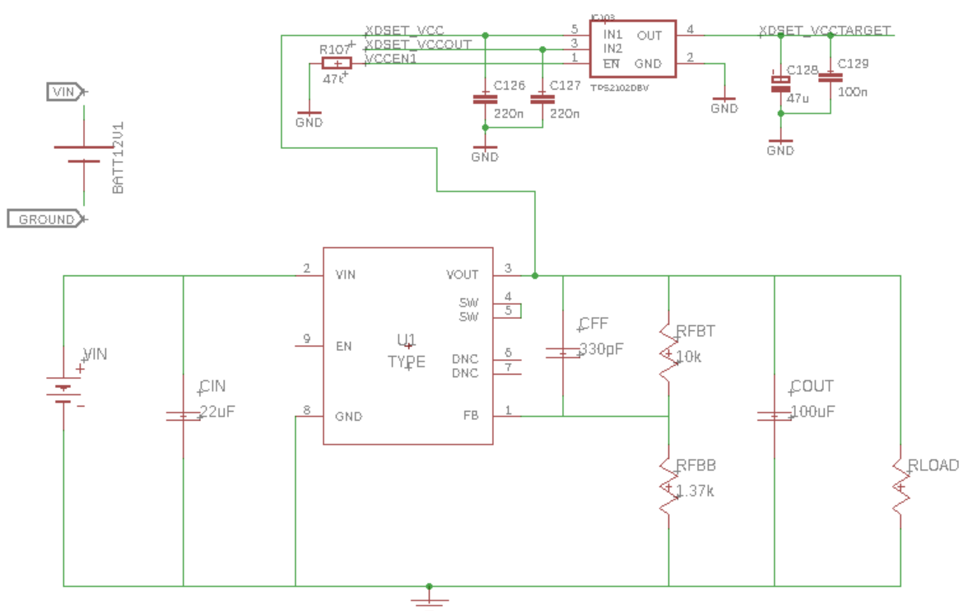

Figure 5-1

This is the layout for the power source for the launchpad. Here in this diagram, the 12V battery is being connected to the input of the power management IC, and then the output of the IC is connected to the input of the launchpad. The tpsm84209 is used as the power management IC since during normal operation, the launchpad uses a current of 5.5V and 250mA.

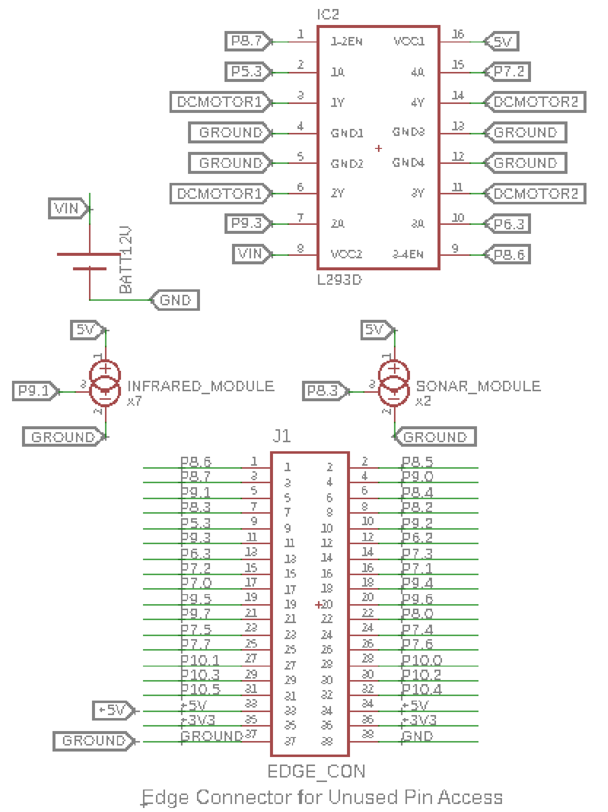

Figure 5-2

This diagram details how the DC motors, as well as the sensor modules will be connected to the MCU. The DC motors require all 12 volts from the battery. The dc motors are connected to a motor driver which is controlled with the launchpad through its unused pins and 5V output pin. The sensor modules require 5V for operation. The data from the modules is sent to the unused pins from the launchpad, where the IR requires 7 (for 7 devices), and the sonar will require 2 (for 2 devices). Pin number 1 and 9 on the motor drive are enable pins which control the speed for the motors.

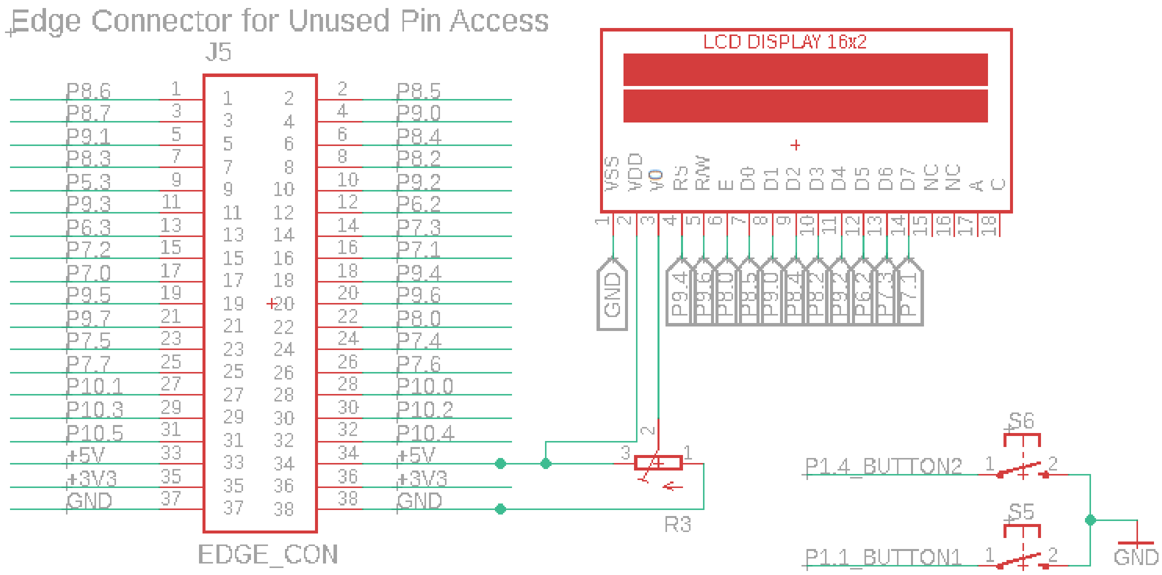

Figure 5-3

An LCD display would be able to be powered through the launchpad, or through the battery, but since the launchpad is already capable of outputting exactly 5V, this would be easiest.

On the LCD, pin RS is for register selection, RW, is read/write, E is enable, and pins D0-D7 are data pins with D7 being the pin that says whether or not the LCD is busy and will or won't accept information at that moment.

Pin V0 is connected to a potentiometer that varies from 0k-20k. The LCD will also make use of the buttons available on the launchpad to help communication with the desired user input. As shown in Figure 5-3, the pin connections are laid out and most will be directed to both the LCD panels and the buttons as well.

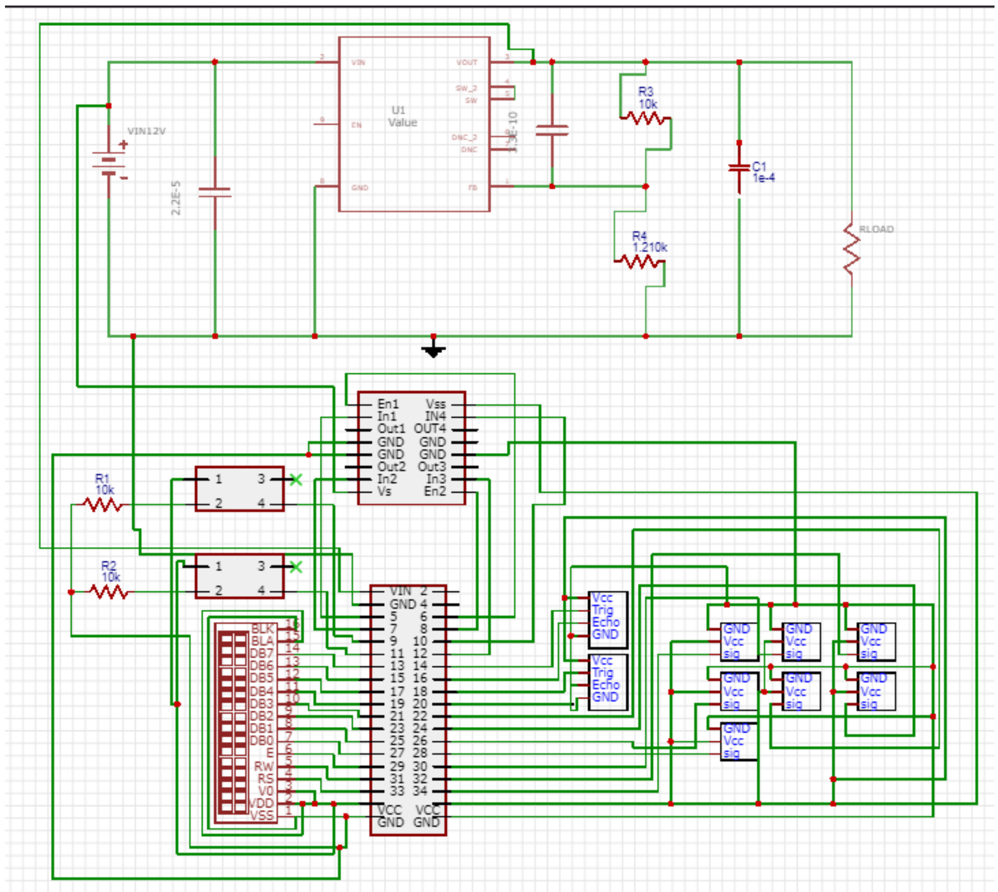

Figure 5-4

Since we decided to use EasyEDA for the PCB, the schematic also had to be completed in here instead of Eagle like the previous schematics. Figure 5-4 shows the complete schematic for the design. In the middle, there is the MCU with all the unused pins that will be used for the project to connect to all the input and output pins. There is the power management IC at the top which is connected to the MCU, and at the input of the IC, would be the battery. On the bottom left the LCD screen is connected to the MCU right below the buttons which will be used to help control the UI. On the right side is the seven infrared modules and the two sonar modules.

5.2 Printed Circuit Board

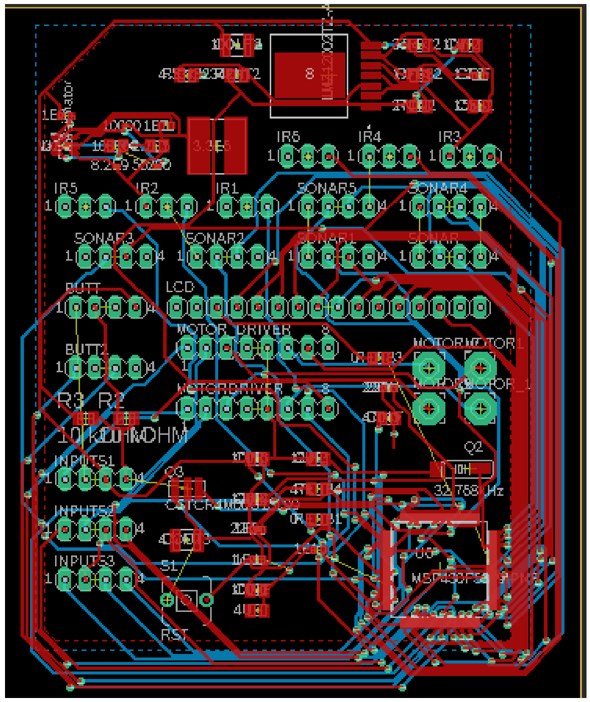

Figure 5-5

Figure 5-5 shows what the printed circuit board looks like based on the schematic, currently. The footprints in the printed circuit board makes the pin able for easy access so all the devices will be connected to the board with no problems. A ground pour was added onto the printed circuit board so that all the ground connections could be easily connected, and it would cause the board to be a lot more accurate, with less noise. For viewing, the ground pour was left unfilled so that the PCB can be seen easier above. The ground pour uses thermal connections so that soldering onto the board will be easier.

For this project Tinjac will exclusively use PCB’s in its final stages of design, replacing more complex microcontrollers that were manufactured by large companies and instead using the schematics of our own design.

5.3 First Subsystem - Detection

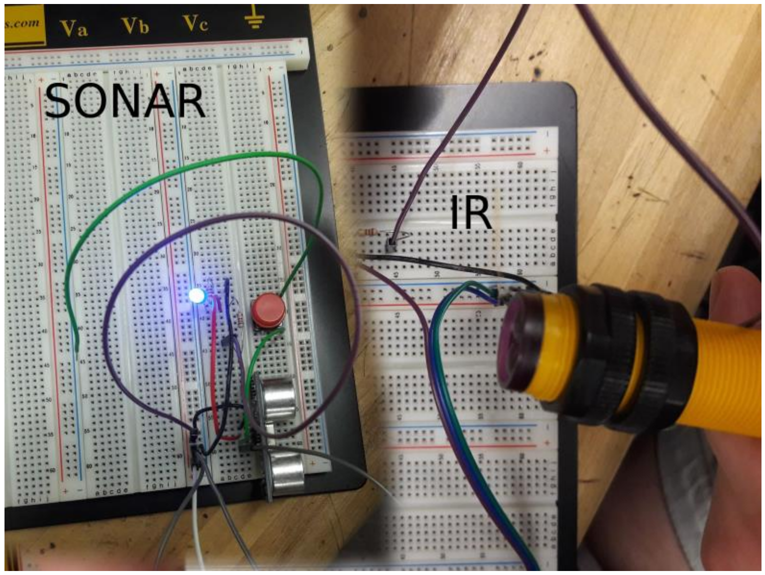

Figure 5-6 ~ Sensor Modules

The first subsystem of Tinjac is the network of sensory modules on the cart. The sonar and the IR sensors communicate with each other, then back to the microcontroller. The code within the Microcontroller handles the input taken in from the first subsystem and uses it to produce any necessary outputs to other subsystems depending on the configuration. The main thing this subsystem needs to accomplish is collision detection to prevent the cart from running into obstacles in front of it. Once an obstacle is detected it should then begin its routine to bypass the obstacle and get around obstacle. Its secondary function is that the sensors on the back of the cart should be able to notice when the user of the cart steps away from the cart and should produce the appropriate response which is to stop and wait for the user to return. In order to accomplish these task there will several of these sensors place along the cart mostly located on the front and back of the cart. Since the cart does not move side to side there is no reason to have sensors along the side to detect for any obstacle collision.

In Figure 5-6, we connected the two types of sensory tools Tinjac will be optimizing (they are set up in separate circuits, but displayed here as a single picture). We wired each pin the sensory tool uses to properly test the accuracy and functionality of each part.

5.4 Second Subsystem - Motion

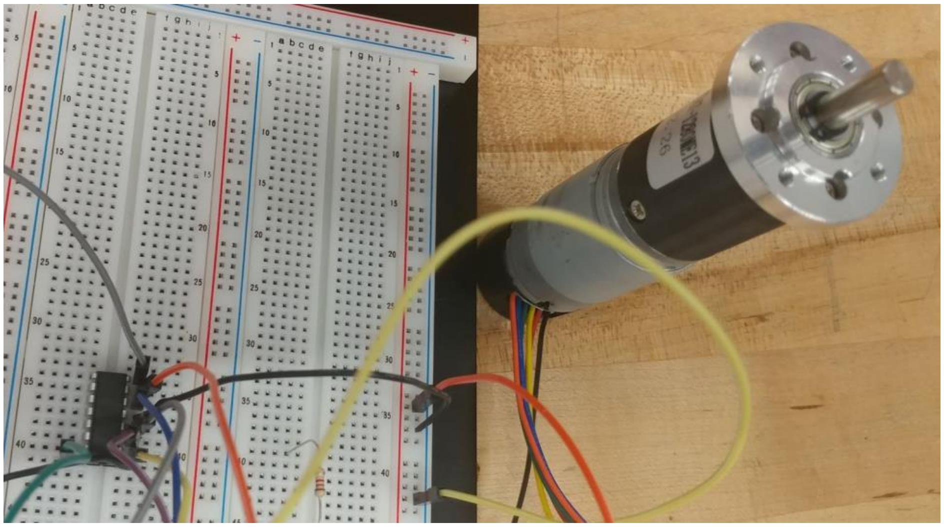

Figure 5-7 ~ Motor and Motor Driver

Our second subsystem is composed of our system of motors and triggers to enable/disable them. Part of this subsystem also includes the code functions that control the direction of the motor as well as enabling/disabling them through the microcontroller. The motor driver chip acts as the middle-man between the motors and the microcontroller, only turning on the motors when they receive the proper signal from the microcontroller.

In Figure 5-7 we have documented how we are testing one of the two motors (of the same model) we intend on using for Tinjac.

While the motor has an internal encoder to manipulate the speed and direction of the motor, we are actually using a motor driver chip instead to be able to control the motor (as it was actually much more simple to integrate than using the encoder itself.

5.5 Third Subsystem - User Interface

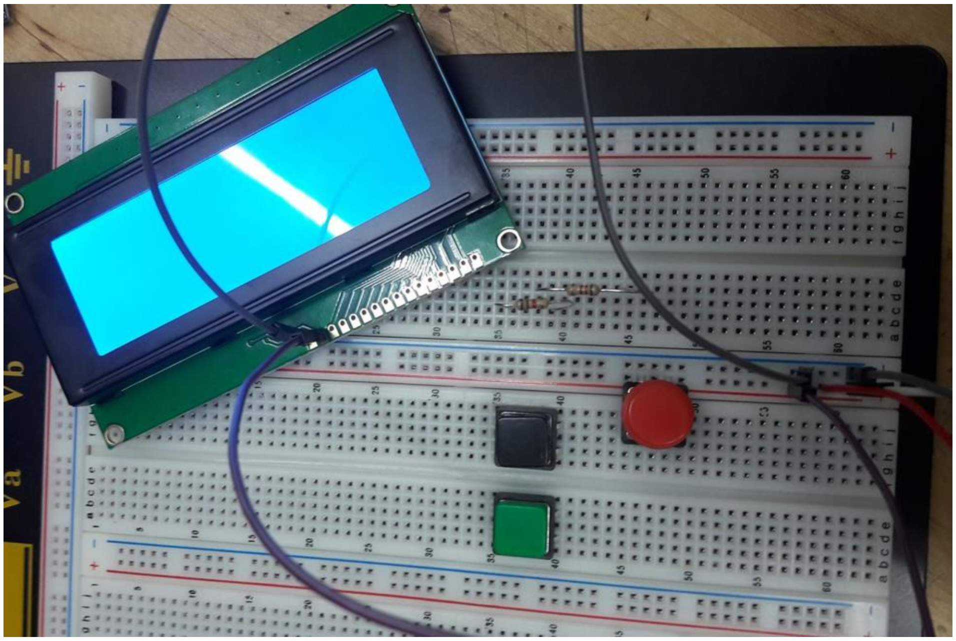

Figure 5-8 ~ User Interface

The third subsystem of our project is the User Interface. The design and implementation of the user interface allows the user to input the items they would like to shop for, as well as the user interface can also communicate any issues with the user doing the cart's progress.

The LCD panel itself will connect directly to the microcontroller and be updated through the pins that it connects to on the microcontroller. It will also get any error or warning notifications pushed to it from the microcontroller as well. The module will be powered through one of the microcontroller pins dedicated to voltage supply.

The buttons will connect to pins that are part of the microcontroller and send signals to notify the microcontroller whenever one is pushed, indicating that the user interface needs to be updated.

In the displayed picture (Figure 5-8), we made a simple circuit to connect the LCD display to a few pins to be able to detect that it can receive and input signal. We also put into view the buttons we intend on using for our early prototype designs of Tinjac.

5.6 Fourth Subsystem - Power

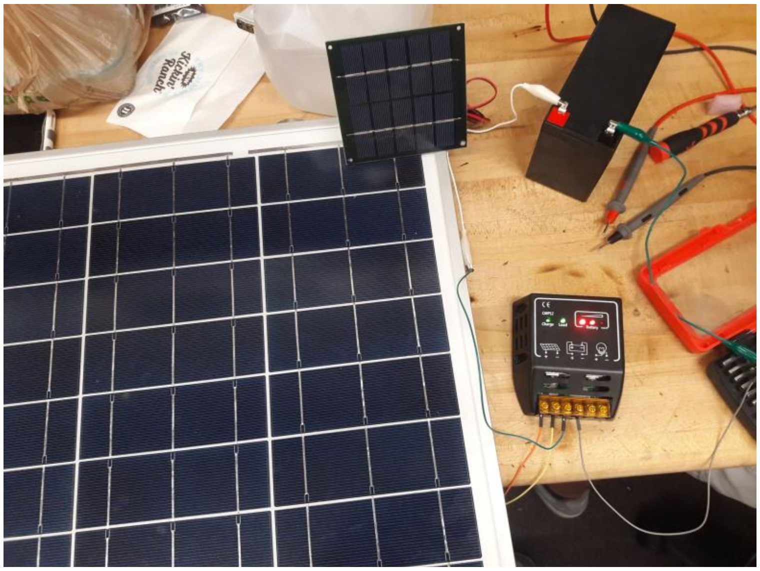

Figure 5-9 ~ Solar Panel x 2, Solar Charge Module, Battery

The fourth subsystem is our powering subsystem. Tinjac uses a large battery to allow maximum amount of usage before needing to recharge. The connected solar panels allow the battery to be recharged when the cart is not being used.

The solar charge module is what prevents the battery from being overcharged from the solar panels. It also provides other useful information like the battery life remaining, when its charging and also when load can be turned on. The battery will supply power to all the other subsystems in Tinjac, including the motors (which require the most amount of power).

To show both the variation in a large and a small solar panel for Tinjac, Figure 5-9 shows the two types of solar panels connected to our charge modules which is then connected to our battery source.

5.7 Fifth Subsystem - Skeleton

Figure 5-10 ~ Sexy Cart on Gross Carpet

The last division of our major system is the actual skeleton of the system. Composing of the cart and modified wheels, the physical framing of Tinjac is what holds all the other subsystems together and allows them to connect directly.

The back wheels of the system will have to be custom-made and designed to fit our motors so they can move and turn with pinpoint precision. The front wheels of the cart will have to be modified to allow smooth turning and avoiding alterations to our straight path design.

The bottom of the cart will be modified so that it can house the solar panel at the base, but still allow items to be placed on top. This will allow the cart to have an ideal center of gravity for movement, while also allowing the cart to recharge properly when the journey is complete (and the cart has been emptied).

Figure 5-10 shows the base foundation for what we intend to use as Tinjac's skeletal system. It will be the support to every other component that we attach on to it and rely on its flexibility to be able to maneuver as we would like.

5.8 Software Design

Using the MSP430 microcontroller, our software structure will be solely constructed using the C language and using the standard 'Divide and Conquer' approach that we have decided to utilize in order to break down each objective into smaller necessary functions.

After each core objective was broken down into smaller functions, a simply flowchart can map out how the function will work at a basic level.

5.8.1 Software Function Flowchart

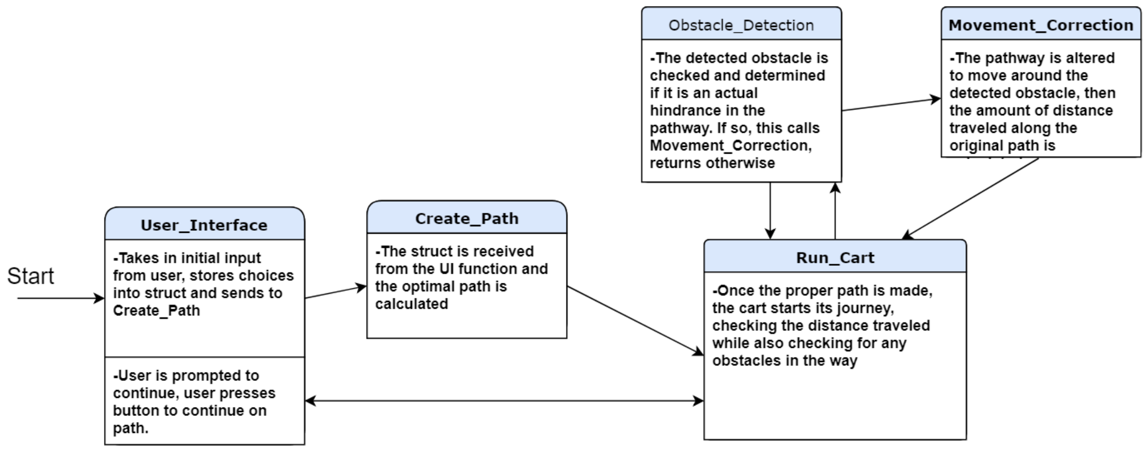

Figure 5-11 ~ Basic Functionality of Software Implementation

The above flowchart (Figure 5-11) is an illustration on exactly how Tinjac will work in an optimal setting using all of the core functions. Once the cart reaches its destination, the Run Cart function will simply exit out and prompt the user in the same way it would to initialize the path.

5.8.2 Core Function List

The following functions would be the main foundation of the software design for Tinjac, as these functions are divisions of all the critical tasks from the combined system to function properly.

User Interface

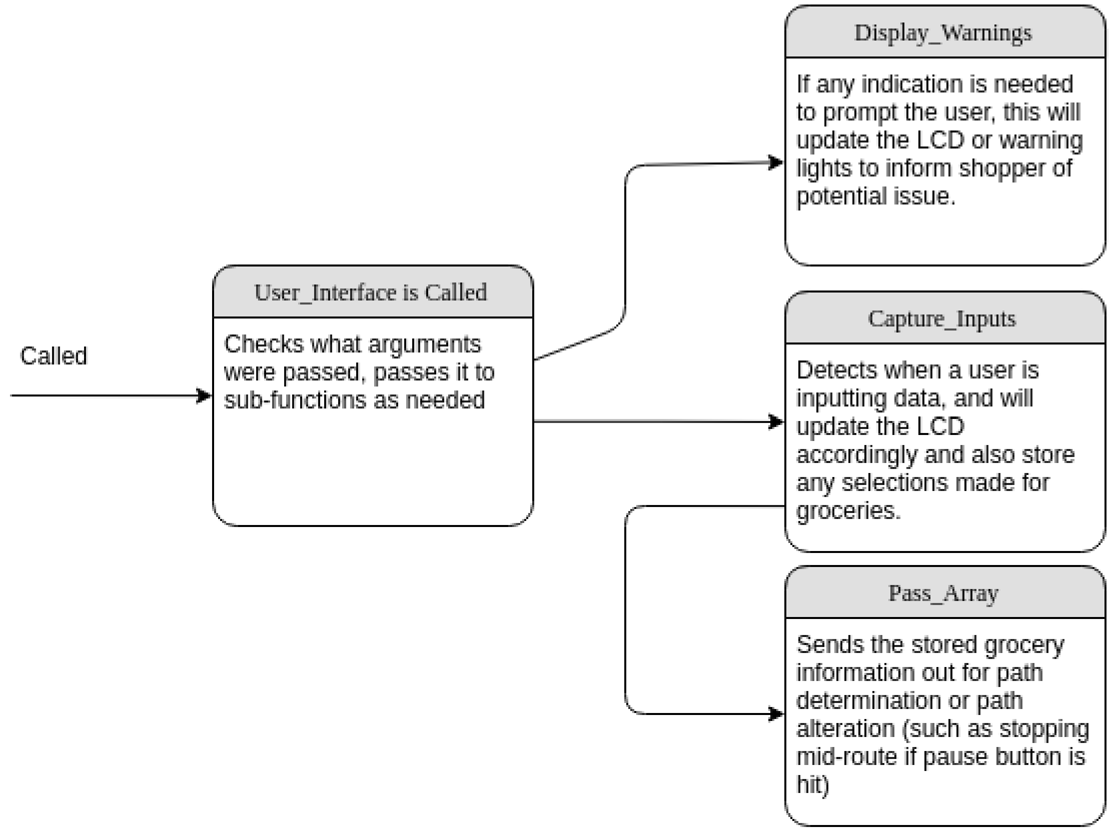

Figure 5-12 ~ User Interface Basic Functionality

Our User Interface function will be designed to handle several user-related function calls, as well as system management related functions. This will be divided into sub-functions for easier management and error avoidance.

This function will be tasked with taking in the user input from the physical buttons on the physical panel of Tinjac and updated the LCD in accordance with which button was hit. In order to properly display every possibility of what type of groceries to purchase, we will have an array of grocery type objects (such as “Eggs”, “Milk”, “Cereal”, etc.) and sort it alphabetically. This sorting is done beforehand (similarly to how we will already have a predetermined store layout), and we will print it to the LCD screen by using a pointer that transverses along the array in a fashion similar to that of a linked list. Once an item is selected, it will be removed from the array and placed in a separate array of “Chosen Items”. This new array of items that the shopper selects will be filled and sent to the Create Path function.

The User Interface will also be tasked with communicating warnings as well as status updates. If there is an issue with the integrity of the cart system, such as a disconnect between sensory modules or the system is low on power, the UI will have to display this issue. If an obstruction is detected, during the Object Detection function, the UI will be updated to reflect that the cart is currently attempting to check and correct the path due to the sensed obstacles.

Create Path

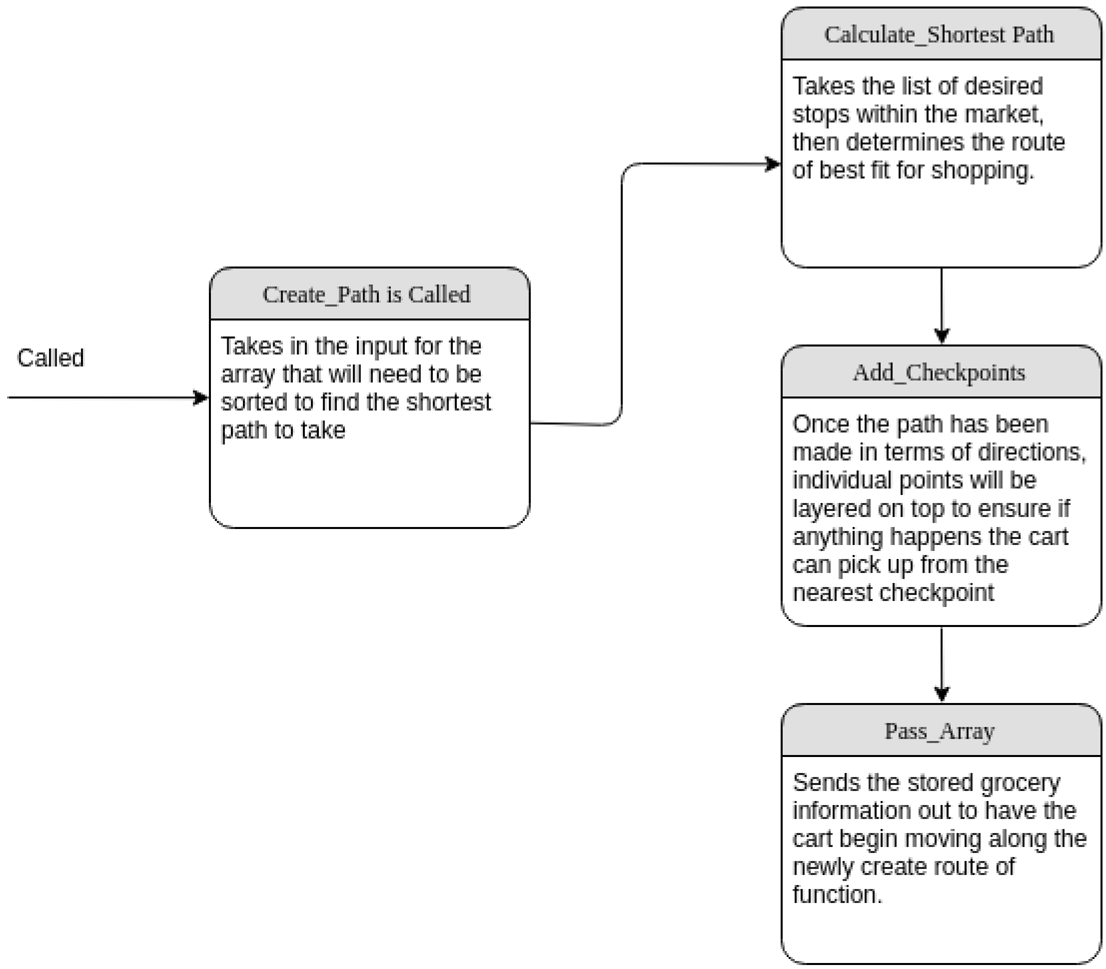

Figure 5-13 ~ Create_Path Basic Functionality

This function takes in a struct of all the grocery items that the shopper wishes to add to the cart, then calculates the absolute shortest path. It will reference a predetermined layout of the store, with entrance and checkout locations.

Since this would consist of very few possible combinations of possibilities, the type of algorithm we use to find the shortest path will have little real-life impact on performance and calculation. However, for a compromise between efficiency and simplicity, we will implement a basic Dijkstra algorithm to find the optimal path.

Once an accurate path is made, it is passed to the Run Cart function, so that continuous progress can be monitored.

Run Cart

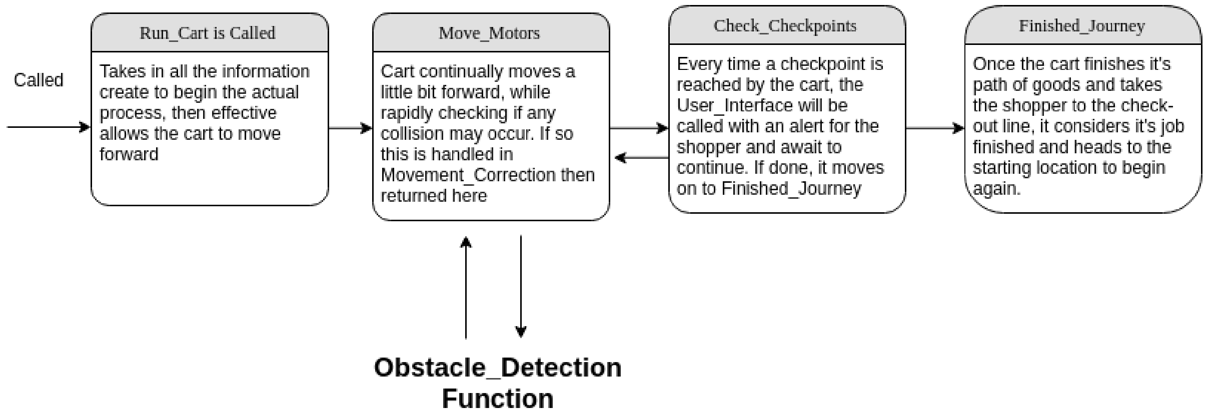

Figure 5-14 ~ Run_Cart Basic Functionality

As the cart physically moves forward, we will use distance detection along with collision checks to monitor when the cart arrives at each checkpoint. A "checkpoint" is simply a designated node within reach of one of the items the shopper entered. Once the next node in the path is reached, the cart will stall until the user prompts that they are ready for the next stop in the path. If any obstacle occurs during this process, the Obstacle Detection function is called mid-run until the specific collision is resolved. Once resolved, the Run Cart receives the amount of distance that the Obstacle Detection function moved the cart further along the path.

These set of instructions will loop until a checkpoint is reached. Once it arrives, the Run Cart function will notify the system and stop the motors. Then the User Interface will be called and inform the shopper it has arrived and will await an input from the shopper to continue along the path to the next node.

Obstacle Detection

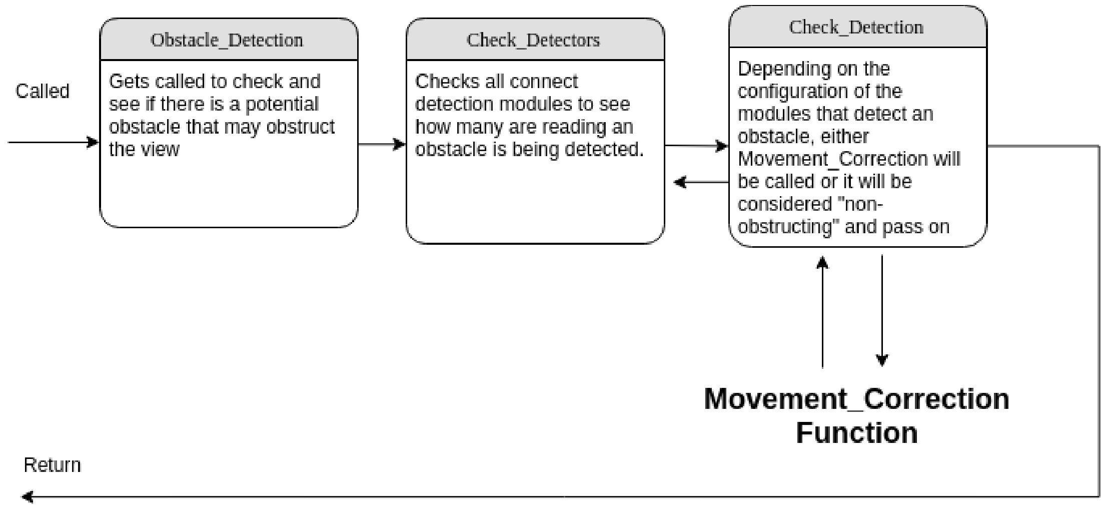

Figure 5-15 ~ Obstacle_Detection Basic Functionality

The object detection function will have a single, yet invaluable, task. We will structure the Object Detection function to work in combination with every form of detection on the cart. Since we are using both sonar and IR detecting modules, we will break this function up into sub-methods so we can accurately check to see that the thing being detected is definitively an obstacle to the path.

When each module senses an object in its field of detection, it will send a signal back to the microcontroller. The microcontroller will then be able to call any appropriate sub-function and will use the combination of detection modules to pinpoint where the obstacle is in relation to the cart's position. After this information is obtained the function then returns the info back out of the function so the microcontroller can send it to the Movement Correction function. If the object is determined to not be an actual hindrance to the path, then the Obstacle Detection simply sends that information back out so the cart can continue its journey.

Movement Correction

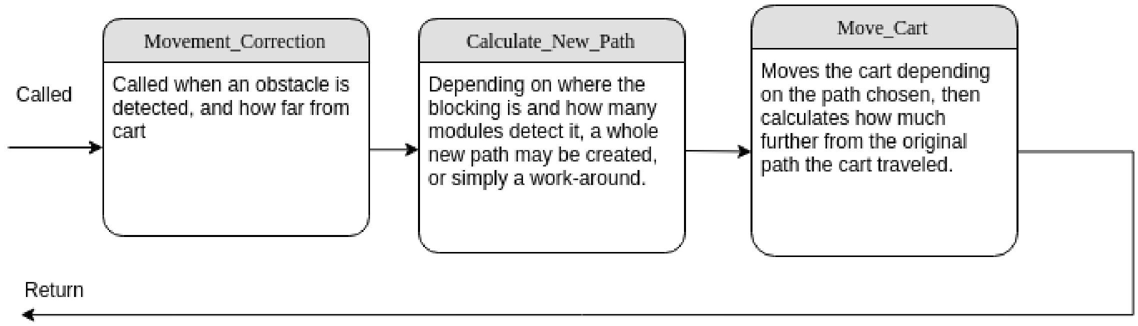

Figure 5-16 ~ Movement_Correction Basic Functionality

Using the detection modules, the Object Detection function checks if there is an obstacle and where it is in relation to the cart. The Movement Correction function will take this relative position and trigger the motors in the specified way needed to move the cart out of the way of the obstacle, but still on the path towards the objective.

With the installed motor driver, this function will be able to trigger the motor in one of three possible ways each: forward, backward, stop. Using both motors and different commands this will allow the function to turn the cart or move it forward/backward.

Once this correction has been made, the Movement Correction function will use the distance traveled to calculate how much further along the path the cart has travelled. Then it will exit out and return all the way back to the Run Cart function with the updated distance so the checkpoint can continue to be accurately monitored.

5.8.3 Secondary Function List

The critical functions in the core function list set up the framework that allows Tinjac to operate effectively in ideal scenarios, however we must also plan and implement solutions that will resolve any errors we could theoretically run into throughout Tinjac's journey.

LED Lights

Some warnings can simply be displayed as an LED indication light, rather than displaying the entire error out on the LCD panel. We will setup LED lights attached to the LCD panel to act as a visual notice to any potential problems. This will be called from within the User Interface function, simply to take a warning and turn on/off an LED depending on the warning passed to it.

This will be primarily used to show the shopper a few basic warning indications, such as low-battery and checkpoint-reached.

Shopper Check

The Shopper Check function is simplistic in its design as the sole objective is to detect when the shopper is too far away from the cart. If the cart wishes to continue along the calculated path, it must do so only if the shopper is close enough to the cart.

There will be a routine check while the Run Cart function is running, to call the Shopper Check function (which will have a miniscule impact on the performance of the cart). Once called, the function will simply check if there is a shopper in the small area directly behind the cart before exiting back out to the original Run Cart function. If there is a situation where the shopper has left the cart, then the cart will stop all motors and continue to check again in intervals until it detects that the shopper has returned back to the cart. Despite this action "pausing" the cart during its path, this action is not actually calling the Pause Cart function, but instead continuously checking to see the exact moment the shopper returns

Pause Cart

Throughout the shopper's journey, there may be a few situations where they shopper may want to stop. They may wish to add something to their list, or simply grab something that is nearby (and not near an actual checkpoint). The shopper may hit one of the buttons on the user interface and the cart will pause wherever it is in its path and wait for the customer to hit the button to continue once more.

This is also important in regards to safety. If all other failsafes fail within the software design, the user can press the button to shut down the cart's movement directly.

Hard Reset

If continuous errors consistently stack on top of one after the other, this function may be called if no other solution can be found. The Hard Reset function is setup to essentially shut down everything that Tinjac is doing and use the list of groceries (the ones that have not yet been reached along the previous path) to create an entire path based on current location.

Calling this function will guarantee that the new path created will use the current location as a starting point to create an entirely new path to see if all the issues can be resolved simply by starting all over.

Emergency Stop

Similar to the Hard Reset function, this function will serve as an emergency fail-safe. In order to avoid any unwanted collisions that could potentially be a threat to shoppers or the store's property, this function can be called if there is a physical issue that can no longer be resolved with software.

Once this function is called, all power to external modules are cut off, with exception to the User Interface display. Then the UI is called and given the information needed to display that a critical warning has been encountered and the system has decided to shut down due to safety precautions.

5.9 Summary of Design

For the software design of TINJAC, the front end will consist of a simple user interface displayed by our LCD screen. The user interface allows the user to select from a list of grocery items and add it to the queue. Once all the desired items have been chosen, the cart will then proceed with its traversal. This is where the back-end part of the project will come in.

There will be a function that takes into account the layout of the store and creates the most optimal traversal path possible to ensure speedy efficiency. Once an accurate path is made, it will trigger the Run Cart function so that the TINJAC can initiate its journey. Part of the Run Cart function consists of two other subfunctions known for detecting obstacles and then correcting the movement to get out of the way of the obstacle and running its course.

For the hardware design of TINJAC, two DC motors with enough torque will help the cart and customer reach their destination. A 12V solar panel will be connected to a 12V battery which is responsible of supplying all the power to the cart. The solar panel will also be required to connect to a controller that stops the battery from taking anymore charge when it's at full capacity.

The controller is necessary from prevention overcharging as overcharging can cause the battery to burn and become damaged which drastically affects performance. With the battery supplying the power, it will be able to power on the MC. Through the MC, the sensor modules are powered on, along with the motor driver. The MC communicates by analyzing the signals from all the components to tell the DC motors how fast to go, when to turn, and when to stop.

6 Project Prototype Construction and Coding

Since this this project is being built from scratch, constructing the project requires some planning and consideration. Schematics can help with this planning and construction process. For the components that are being used in the design looking at the schematics can also help understand how the components work and some things that would need to be taken into consideration when making the final schematic of the project.

Once the final schematic is designed and the group approves of it the next step is to look at the PCB and where to get one made at a realistic and affordable price. Lastly since this project involves calculations to determine distance and were it is located coding is an important part to plan to determine what are some of the functions it should include.

In this section we cover the schematics used in the Tinjac design. It also goes over the some of the vendors being considered to order the PCB used in the Tinjac. Lastly it will go over the coding plan for the Tinjac which states what the code should be able to accomplish.

6.1 Integrated Schematics

This subsection primarily focusing on looking at the schematics of the components that are being used can help with the understanding of certain pieces of Tinjac. It helps with designing the final schematic that will be used to implement in the final design of the Tinjac.

This section we show some of the schematics used to implement the Tinjac design. It goes over the schematics for the individual components of the carts implementation as well as a brief description of what the component does or how it works. This technical section of the paper is a critical part for the proper understanding of how we would need Tinjac to function, and the limits of its capabilities.

Please note that the majority of the following schematics are directly taken from the original source of the specific component to avoid any inaccuracies in the display of the part functionality. In order to properly cite each of the unoriginal designs, the original authors have been reached out to and any communication is notated within the Appendix section of this paper.

6.1.1 Generic IR Sensor for Object Detection

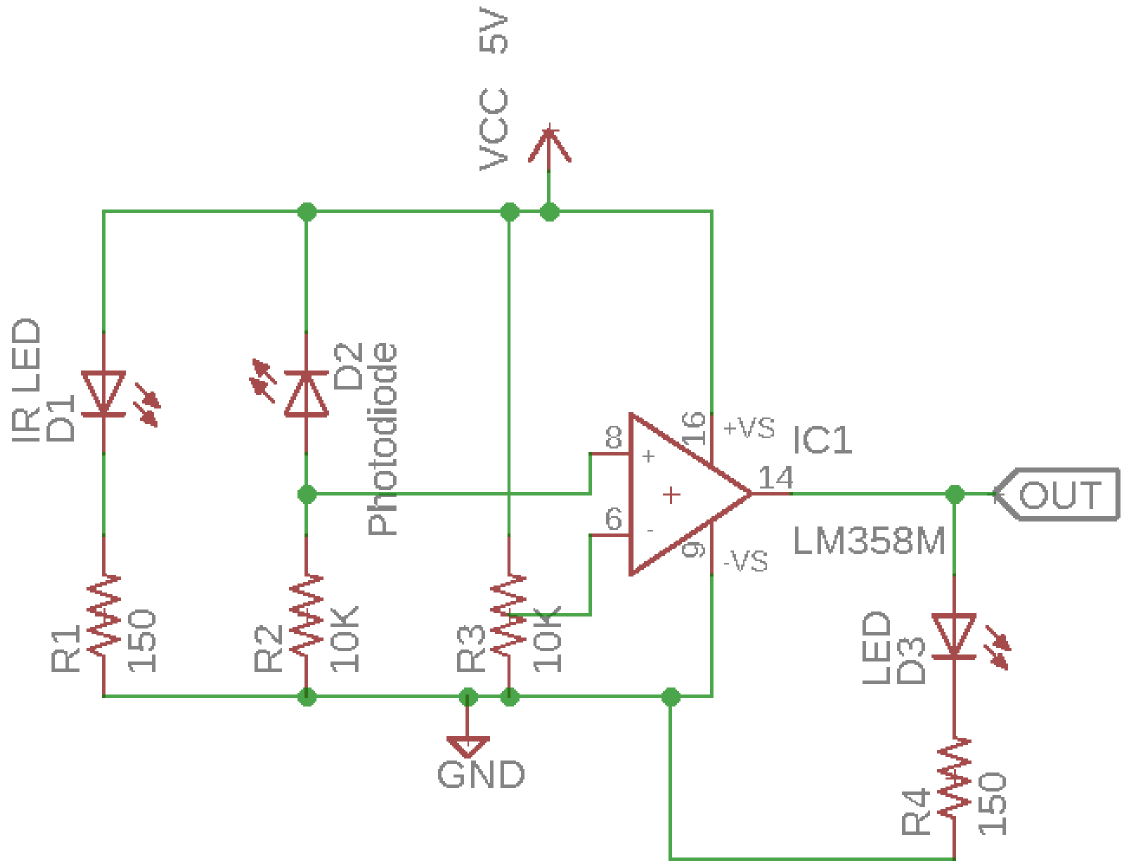

Figure 6-1

The IR LED in the circuit sends out long wavelengths of lights. Then the photodiode converts light into a current and this happens when a photon is absorbed. The circuit measures heat and detects motion by measuring infrared radiation.

Figure 6-1 is a generic implementation of how an IR module could potentially function. This shows us the basic schematic we can referencing when designing that specific subsystem, as well as our PCB so we can simply implement it directly.

6.1.2 Sonar Module Schematic

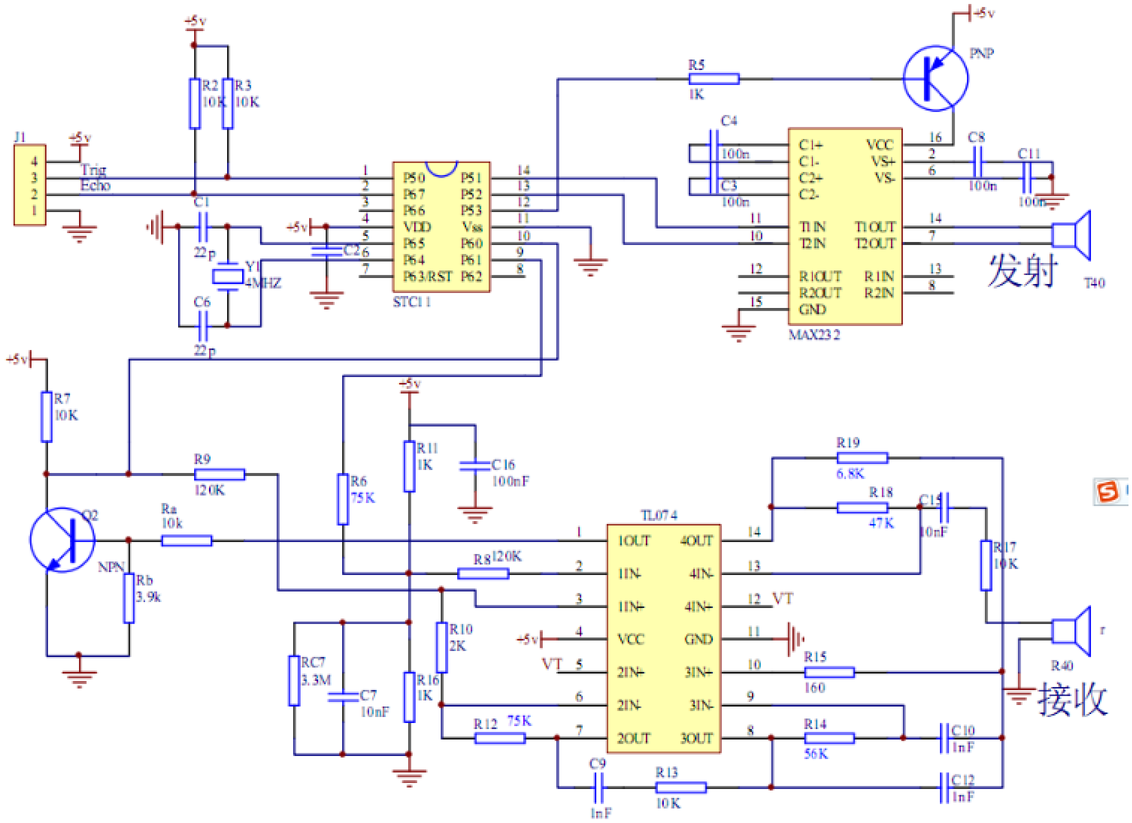

Figure 6-2

The circuit sends eight 40 kHz signals and then detects if the signal returns and analyzes the time it took to return. This signal is sent out through two speakers as shown in the circuit. 40 kHz is much too low of a frequency to be heard by the human ear, hence the term, sonar. If the 40kHz signal returns, to the circuit the circuit then sends out a small voltage pulse which helps let the MCU determine if an object has been detected.

6.1.3 MSP430 Schematic

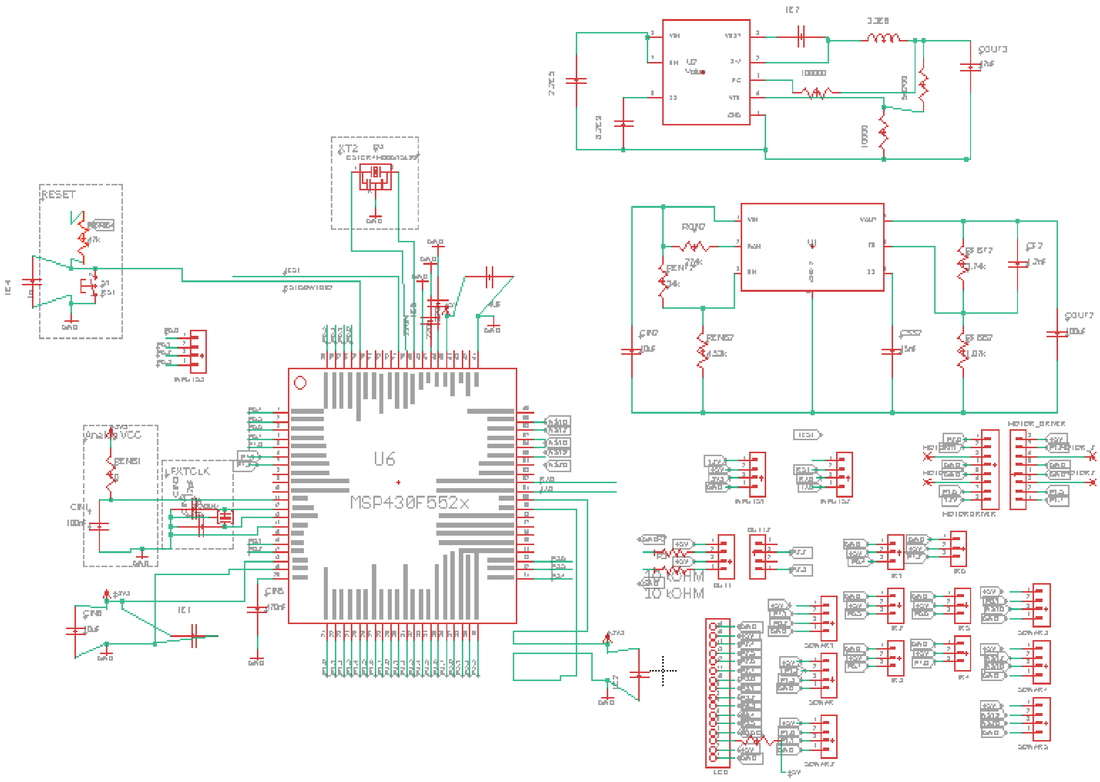

Figure 6-3

This schematic is showing some of the pins from the MCU. Specifically, it's showing the analog and digital supply voltages and grounds and the internal components. The analog ground and voltage are especially important to be able to convert the analog voltage signals into digital so that the MCU is able to process it best.

6.1.4 TI L293D Motor Driver Schematic

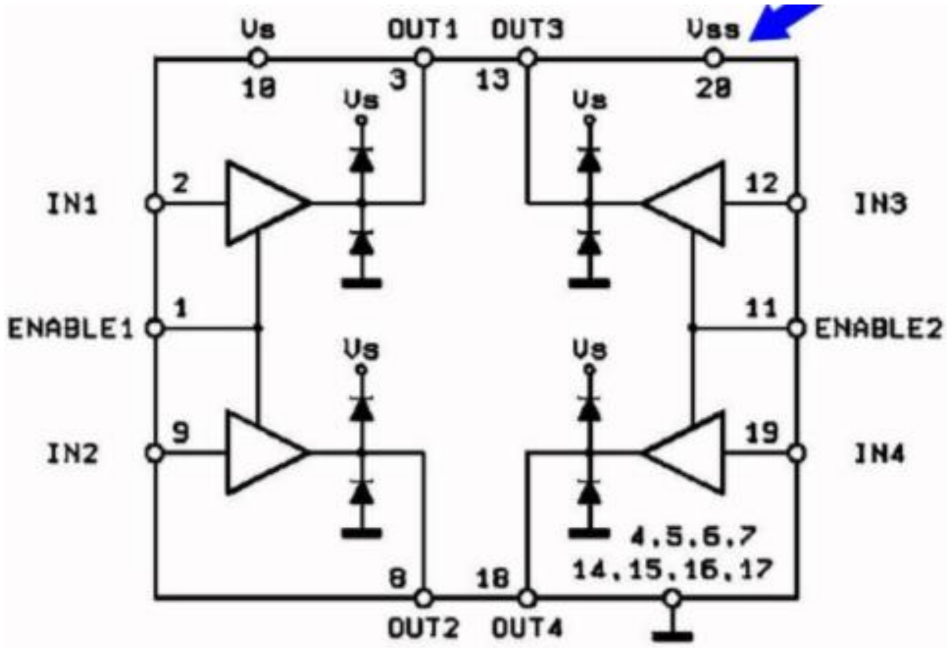

Figure 6-4

Here in the schematic for the motor driver, we are able to see how the user is able to control each motor through the motor driver. An enable pin is connected to the op amps which tells the op amps how much voltage is able to be passed through the output. There are only two enable bits though which means that two of the motors would working together.

Otherwise, the motor driver could just be used for a total of two motors instead of four if the user wanted to control each independently. The four outputs in the schematic are each connected to the corresponding DC motor.

6.1.5 Solar Panel Charging Controller Schematic

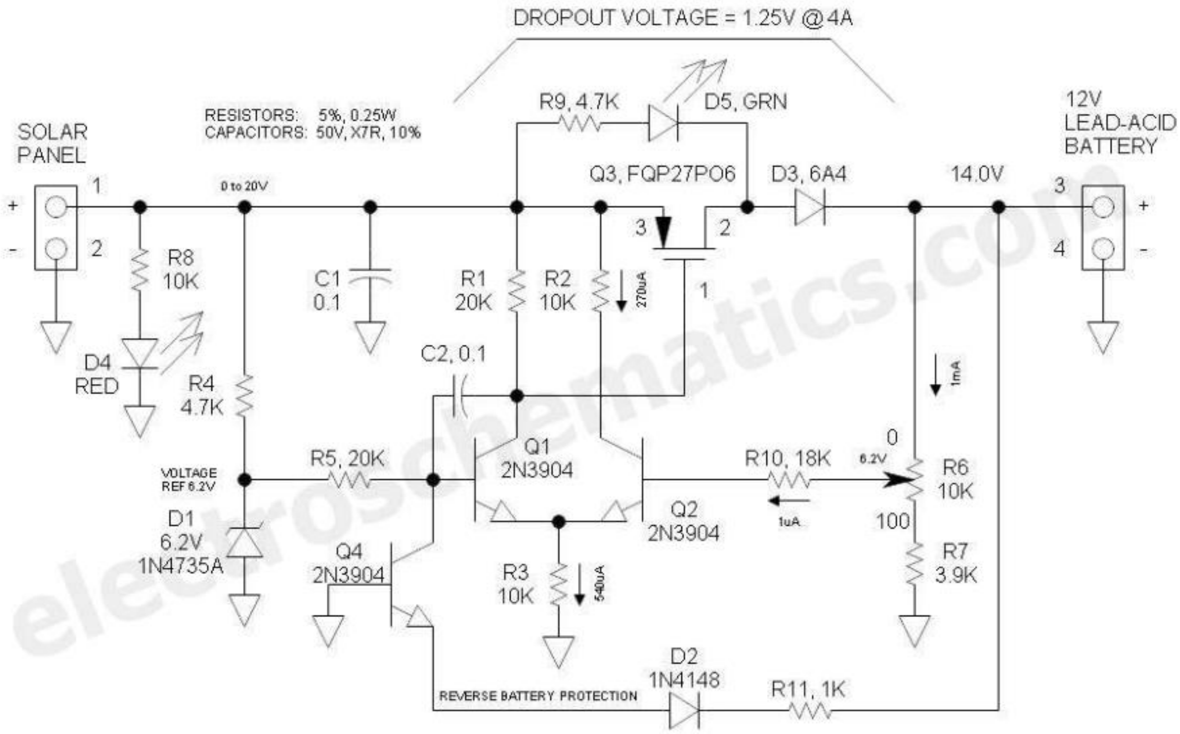

Figure 6-5

For Figure 6-5 it is shown what is inside of the charging controller which will be hooked up to both the solar panel and the rechargeable battery. This device is necessary to prevent overcharging on the battery. When the battery is at maximum capacity, the solar panel stops storing energy.

6.2 PCB Vendor and Assembly

Once a final schematic is decided on then the next step is to get a PCB made. There are many options available of companies that build PCBs with the schematics that the designers give. They offer many different types of PCBs materials and other things that will need to be decided by the designers. This needs to be taken into consideration by the designer which will affect pricing and delivery times.

The following sections also discuss some of the PCB vendor options that are available for use. It also describes the pricing for the various companies and also some of the time frames for when the PCB boards will be delivered. The companies track records are also taken into consideration when deciding which vendor could potentially be used.

6.2.1 Advanced Circuits

This vendor offers qualify PCB designs with efficient manufacturing, assembling capabilities, and shipping. In fact, this company claims to have the best shipping in the industry. The company also doesn’t have any order requirements, if the person ordering is a student, meaning that someone could just order one really simple PCB and they will still accept it. The pricing is $33 for each 2-layer board, and $66 for a 4 layer.

6.2.2 Gold Phoenix PCB

This vendor is known for having relatively cheap, and good deals for PCB designs, with shipping only taking 5 days for a PCB of 2 layers. Testing the finished PCB is necessary when ordering from this company since they are known to have errors, but very rarely.

6.2.3 Sunstone PCB Express

Sunstone is known to have decent customer service. This company designs their PCBs with cheap material which thus allows them to make the boards more affordable. This company also has the option to have the lead time be anywhere from 1 day to 3 weeks, and the advantage to choosing 3 weeks is that the final price is cheaper, and in some cases, it can be a lot cheaper.

6.2.4 JLCPCB

This company is the biggest PCB prototype in China, and they offer affordable pricing and fully tested boards. For a two layer board, this company charges two dollars for every ten pieces on the board if the size is less than 100x100mm which is a very decent deal. Otherwise, the price for a two layer board is 58 dollars for every square meter. Another great option about this company is that they can export directly from the PCB program, EasyEDA, which is free, online, and very easy to use.

Out of these four options, Advanced Circuits would be the best since they have the best deals for a PCB design and shipping doesn’t take very long either. Although Gold Phoenix is known for sometimes having errors, this would probably be most likely to occur on very complex PCB designs which our PCB won’t be.

If we have our PCB design ready and wouldn’t be needing it for use for at least another month, then Sunstone would be the better option since they could deliver it relatively cheaper than the other companies that way.

Given all these great traits about these companies, the best option for our project would be JLCPCB as they have a really good rate per parts used. Our schematic would just have to be very conservative with it’s parts usage. Another selling point of this company is how easily PCB designs can be sent to them from Easy EDA, which is a really good program to use.

6.3 Final Coding Plan

Tinjac’s software design will be setup using the C language and implemented directly onto the microcontroller. In order to have a fully functioning program assisting the Tinjac system, the code’s main focus will be to accurately handle all the instructions that are given to and from every pin located on the microcontroller. Key challenges with this design would be handling interrupts in the middle of running the main function so the cart moves forward.

In order to accomplish real-time error handling, the code is written so that the cart will be instructed to turn on the motors and move forward. Once this step is done, the code enters a “listening” state, checking for any potential interrupts that would stop the cart for any reason such as user input or obstacle detection.

While in this listening state, the code will also check for less-critical errors to prompt the user, like when the battery is detecting a low charge or if the user may be straying too far from the cart and a stall could occur.

Once the cart has finished stopping at all scheduled points, the code will send a prompt to the User Interface and ask the user if they wish to proceed to the checkout. If so, the cart will head towards the predetermined location for the checkout aisle. After that it will wait for the user to press the continue button on the UI panel to inform Tinjac that the shopper has finished their transaction.

After Tinjac has completed its journey, the cart will be set into a “charge mode” which will turn off all excess modules to conserve power and enable the solar panel system to begin charging.

7 Project Prototype Testing Plan

The primary purpose of this section to outline how we are going to go about testing the project and all the various components and code individually and when they are all combined for the final demonstration of the design working. The reason it is important to test components individually is because if something doesn't work properly or as expected and when it's all built the designer now has to figure out exactly what is wrong and troubleshoot the whole thing, were as if it was tested individually there is only one thing that could be wrong.

This section also covers the testing from a hardware and software standpoint. This is done so the group can familiarize themselves with how connecting the devices works and how the programing of the devices and subsystems should be done. It also covers some some of the design methods for coding that could be used when building the code.

7.1 Hardware Testing

This subsection specifically focuses on testing the hardware devices individually to make sure they are working properly. It also helps with understanding how the devices need to be connected and will provide some things that will need to be taken into consideration when building the final project. These tests will try to observe the components from a purely hardware perspective so the microcontroller will try to be used as little as possible so a coding error won't be introduced.

In order to focus on the objective of the subsystem, only the specific core components of the related subsystem are thoroughly tested for each set of trials. In order to decide how we would strain and monitor each individual part we were forced to take a step back and plan out a set of accomplishments that each part would have to reach in order to consider it "passable" for the test. This could be composed of numerous different small checkpoints (like meeting specific voltage levels), or fewer large-scale goals such as displaying a specific message to show accuracy.

7.1.1 IR Sensor Test

Most of the IR sensor modules that were found had some things in common that being that there are three pins which are for ground, voltage, and signal. The ground and voltage pins are just to turn it on and have it running and the signal pin sends a signal once it detects something within its range of detection. Sometimes there is an extra enable pin which allows for it to turn off or on. A simple test to check if the IR sensor is working from a purely hardware perspective is to set it up on the breadboard and have an LED hooked up to the signal pin. When the IR sensor is triggered it will send a voltage that will turn on the LED which indicates that the IR sensor is working as intended otherwise something is wrong with module.

IR Test Results

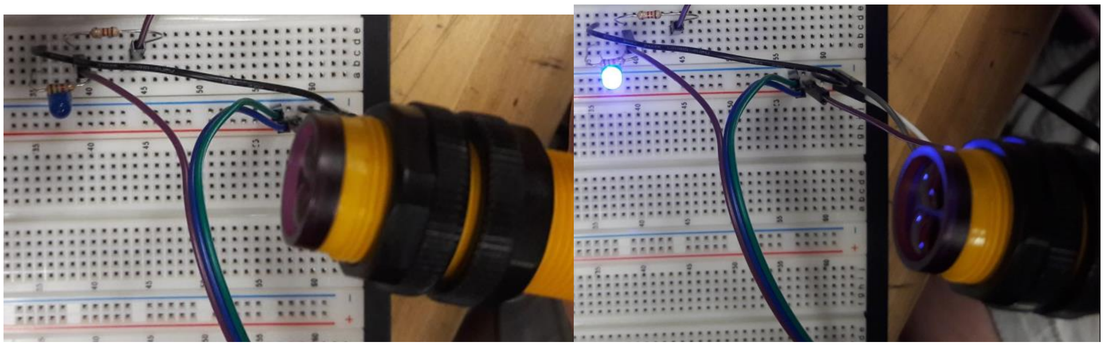

Figure 7-1 ~ IR Test

The hardware test tried to replicate conditions that would occur in the final state of the Tinjac design and worked as expected. The test shows that when nothing is in front of the IR sensor then it sends out a voltage. Then when something enters its range it stops sending out a voltage. The range that was measure for the IR sensor was about 10 inches.

The signal that the IR sensor sends out is about 5V which is useful information if we need the signal to turn anything on and we can use a voltage divider to adjust the voltage accordingly like the LED we used in the test because the first one we used burned out because we did not realize this.

7.1.2 Sonar Sensor Test

The sonar test will be similar to the IR test since it also serves the same purpose of detecting something in front of it. The Sonar module has 4 pins, those being the voltage, ground, trig, and echo pin. The Voltage and ground are just to have the module running which the trig and echo are the pins for actually sensing things. The trig pin controls the frequency of the wave that is sent out and once that wave hits something and comes back the echo pin will turn on letting the device know that something is in front of it. So similar to the IR sensor hook it up to the breadboard have an LED set up to the echo pin. Once a voltage goes to the trig pin the frequency will be sent out and if it bounces off anything and the echo pin receives something it will turn on the LED.

Sonar Sensor Test Results

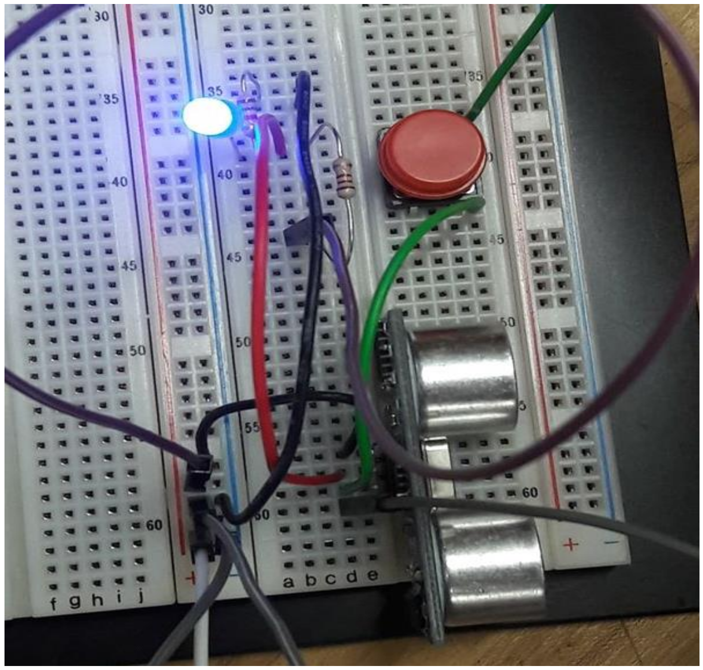

Figure 7-2

The sonar sensor test worked as intended to a certain extent. When nothing is in front of the sonar the LED is always on but when motion is detected in front of the module then the light flickers but does not turn off. This means it is detecting some kind of motion or obstacle but we will need to experiment more with it to see how and what it is really doing.

We believe this can be solved when we actually start writing out programs for the module since in the experiment the sonar module is constantly sending out a frequency whereas when we program it will send out the frequency in short bursts, then stop after each burst and wait and see if the frequency bounces back. The range for the sonar module was farther than the IR sensor but also wider so this is something else that will have to be taken into consideration when programming it.

7.1.3 Motors and Motor Driver Chip

The motor chip that is being used for this project can control two motors max. The important pins that control the motors are the two input pins that will determine whether the motor goes forward or in reverse.

In the final build of the project this will be controlled by the microcontroller but to test this it can be replicated from a purely hardware standpoint on a breadboard. This can be accomplished by hooking a motor to the two output terminals on the chip and then to test which way its spinning hook one input to the voltage and the other input to ground. This will cause it spin one way and then if we flip the voltage and ground it should spin the other way. If both inputs are off or connected to ground it should not spin at all. Assuming everything is working as intended then the chips and motor work fine otherwise something is not working properly on the chip or motor and the motor would be easy to check by just touching the voltage and ground on the terminals of the motor.

Motors and Motor Driver Chip Test Results

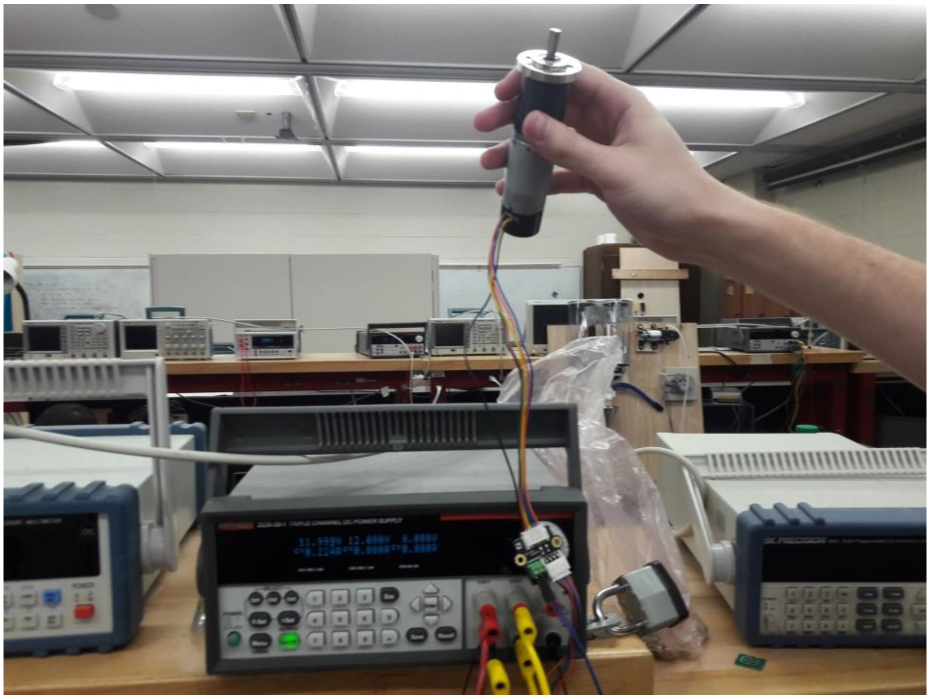

Figure 7-3

This specific testing setup was probably one of the more complicated tests due to lack of experience with the parts being used here. Pictured in Figure 7-3, our motor switching test is shown. To simply turn the motor on is simple enough since all it requires is 12V to be supplied to as shown in the figure above. The problem was with the motor driver chip and not realizing the different voltages that have to go in it and how it works exactly. These are 12V to power the motors, 3.3 voltages to turn on the chip, and 5V to determine which way we want to motor to turn.

Once we figured out the pin layout of the motor driver chip and and the proper voltages to turn everything on things worked as expected. By changing were certain voltages are applied we are able to stop the motor and change its rotation from clockwise to counterclockwise thats to the motor driver chip.The figure shown in a previous part shows how we hooked up the motor on the breadboard.

7.1.4 Power Supply and Solar Panels

To test the power supply, it is just a simple test with a voltmeter see if it is outputting the desired results. To test the solar panel, we would hook it up to the power supply and this device called a solar panel charger control. This device allows for the solar panels to stop charging the power supply once it is fully charged. The solar panel charge control has some LEDs on it to show various things like the battery life remaining, whether the solar panel is charging the battery, and whether the load can be turned on or not.

Power Supply and Solar Panels Test Results

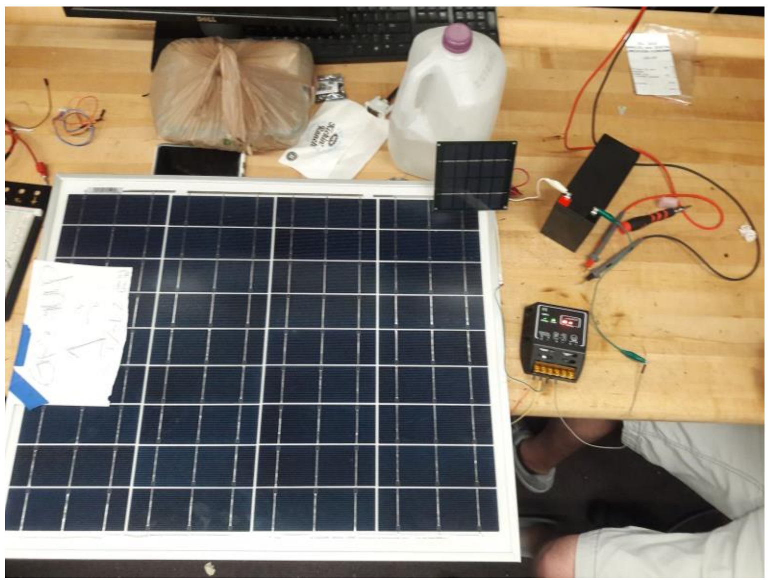

Figure 7-4

The first part of this test was just using a measuring the voltage across the battery and make sure it is 12V which it is a little over that so we know the battery is working properly.

Next was setting up the solar panel and the charge controller. The original solar panel we intended to test this with was not working properly so we borrowed a 50W one from the lab which is able to produce 10V indoors from the lights in the room. This is more than we expected from it and we will probably end up getting a smaller one since this panel is to big to fit on the cart but it allows us to test how the solar charge controller works for now.

As for the charge controller it showed that the solar panel is charging the battery with the Charge LED lighting up and it shows that the battery life of the battery. As for the Load LED on the charge controller it lights up indicating that a load can be applied and it will be able to turn on.

7.1.5 LCD Display

From a purely hardware perspective, in order to test if this is working properly, we just need to hook it up to a power source and make sure all the segments light up. This would go well in conjunction with the software testing, seeing as how the LCD would be programmed in C and all the testing of the functionality of the LCD can be easily done using the microcontroller and the source code that is written.

LCD Display Test Results

In order to properly test the LCD we applied 5V to certain pins we are able to turn on the LCD display. In order to test it further we are going to need to write a couple programs to make sure each segment is working properly.

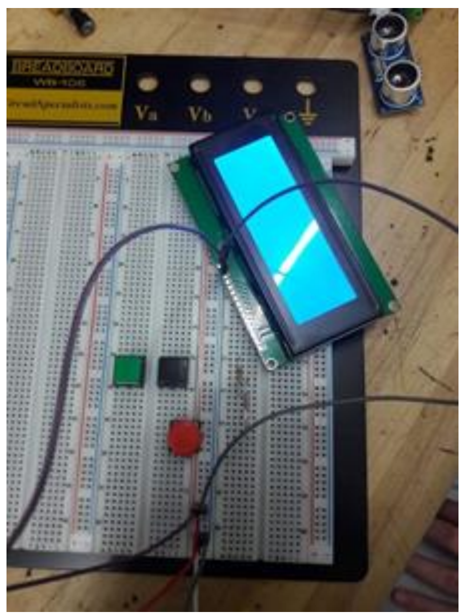

Figure 7-5

Shown in Figure 7-5, our basic configuration for testing was composed of the LCD panels, some basic pins for connecting to the LCD and a power source to make sure the pins connected to the proper channels on the LCD board. This test was satisfactory to our current standards, however we will have to wait until an early prototype is constructed for us to be able to properly test it by printing to the screen itself.

7.2 Software Testing

This section specifically focuses on testing the software to make sure they are working properly. It will also discuss the different types of software process models that we will be using, the development tools that will be used, and the primary programming languages used to develop and implement the code.

Using different models when making progress through the software design has different impacts on the speed and efficiency of the code we implement. This is the reason we listed out several options to choose from for the models we could potentially use for our project while writing the code for the subsystems.

Testing the software will come in individual parts as well, using specific code to run a simple set of tests for subsystems that would require some amount of software to fully test the functionality of (for example, the User Interface subsystem since it includes the LCD panel).

7.2.1 Software Process Models

In order to impose consistency and structure on a set of activities, you must have a solid process laid out. This will essentially guide you to understand, control, examine, and improve the activities of your project. Because our project involves integrating software with hardware, we will need to find inconsistencies, redundancies, and omissions so that we can evaluate the appropriate activities for reaching our project goals. The way to do this is to select an appropriate software process model that will work with our requirements and allow us to complete each major phase of our project.

The first approach we considered was using just the Waterfall Model. The reason why is because it is simple and easy to explain to our end users and presents a linear sequence of process activities. Displayed below is a diagram of how the waterfall model works:

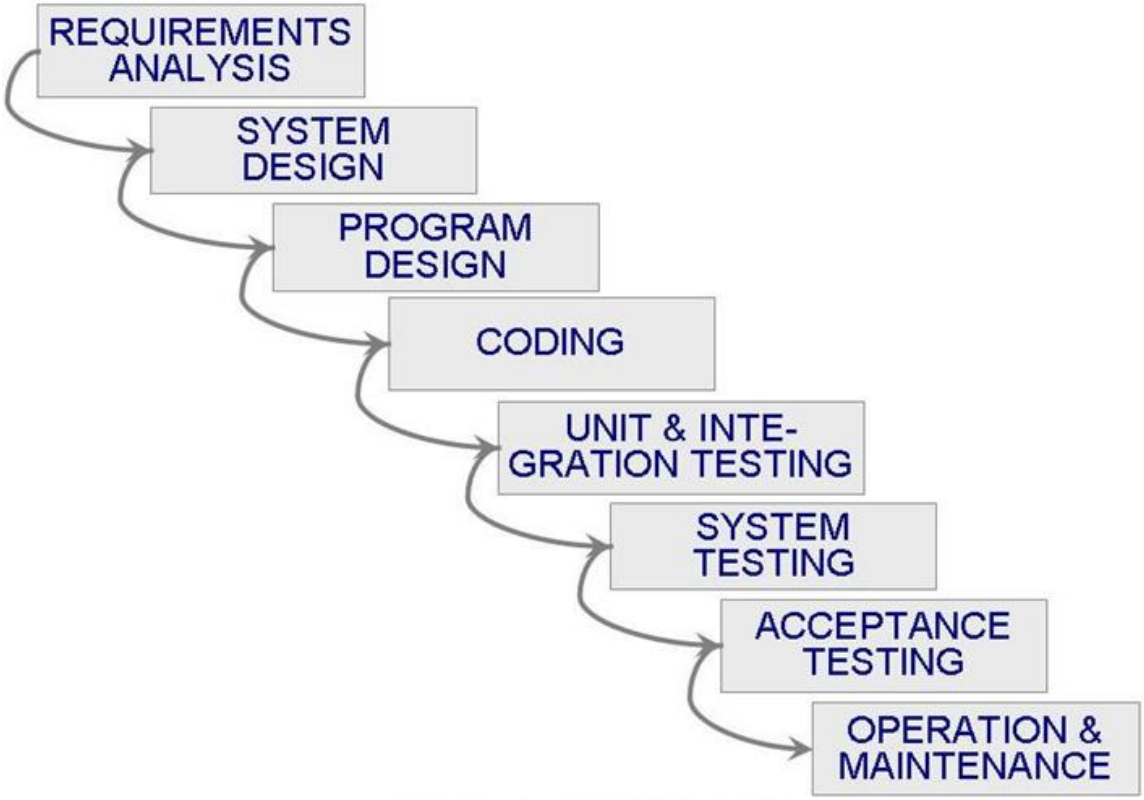

Figure 7-6 ~ Waterfall [10A]

Shown in Figure 7-6, there is no iteration in the waterfall model. Most software developments apply a great amount of iterations. This will not be the case with us since our project involves the integration of both electrical and software components. Some of the other drawbacks of utilizing a strict waterfall model during the software process is that it provides no guidance on how to handle changes to products and activities during development. The process is more similar to a standard manufacturing process rather than being a creative process. This in turn can cause a long wait before a final product is delivered.

The second approach we considered was using the Prototyping model. This model allows us to identify just the basic requirements of our project, developing the initial prototype, reviewing and examining and getting feedback on potential changes, and then finally revising and enhancing the prototype in order to improve the final product. Advantages of prototyping include having a shorter development process, reduced risk of failure, and it serves as a better fit to customer requirements. A huge disadvantage is return is that in the case of something failing, there will be less preparation set in place, which will in turn affect the entire software development life cycle.

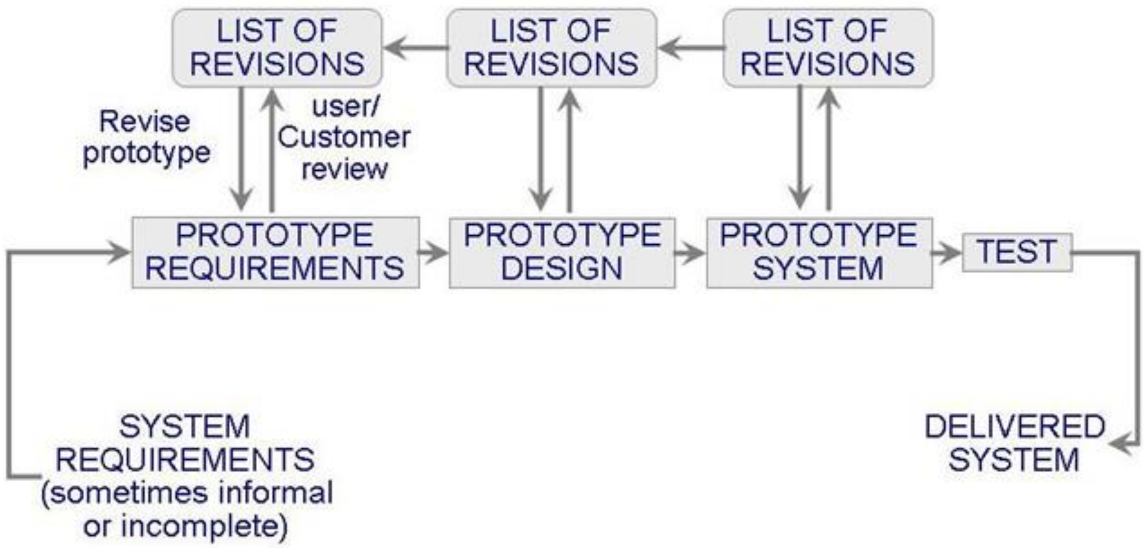

Figure 7-7 ~ Prototyping Model [10A]

This model above (Figure 7-7) allows for repeated investigation of the requirements and design of the project. Showing that any step of the way would require verification before progressing towards a final system. Once the loops of each revisions is completed, the testing checkpoint is reached which is the final step before a complete system can be delivered.

Since both models listed above provide their own benefits for us in terms of constructing our autonomous shopping cart, we will be following the Waterfall Model combined with prototyping. The reason we chose this, and not something more versatile, like the Agile/Scrum methodology, was because this model works well for problems with minimal or no changes in the requirements.

With a partially developed product, we will be able to assess alternative design strategies and understand what our system will be like in order to properly verify and validate our project requirements.

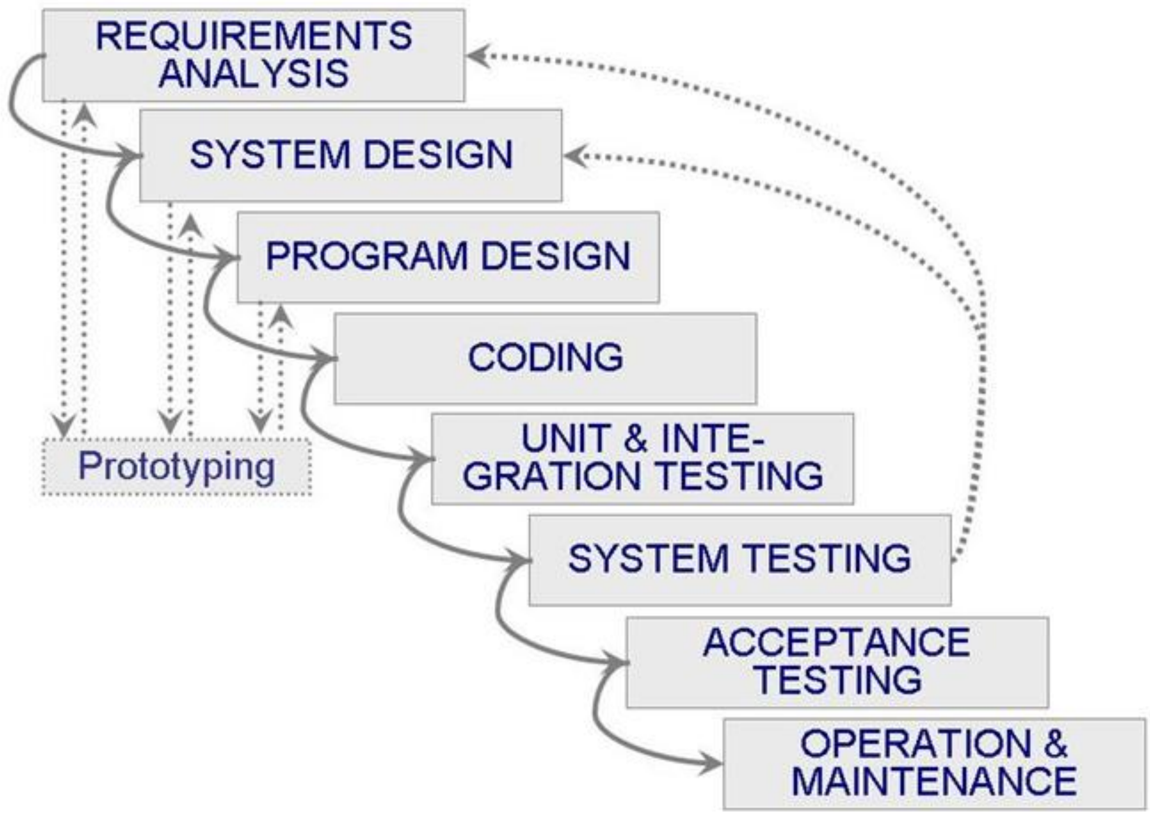

Figure 7-8 ~ Waterfall with Prototyping [10A]

The figure above (Figure 7-8) shows what an ideal division of the design model, using a waterfall technique combined with prototyping. This progression model allows us to optimize on any mistakes we make and prevents us from having to redo the entire project or large sections of it that could be affected by the mistake.

7.2.2 Development Tools

During the planning phase of our project, we were faced with the decision to either choose an Android application to control our autonomous shopping cart or to have a basic LCD UI system connected to the cart itself which would then transmit user selections to the microcontroller. For an android application, we'd have to use a Java based IDE such as Eclipse, Netbeans, or the Android Studio IDE to write the source code for our application.

The benefits of using Java to create a GUI is that it provides a better user experience in terms of look, feel, and usability. In addition to having an optimal GUI that looks great, some other benefits of using the specific Android Studio IDE is that it comes with an Android OS Emulator. This is helpful if none of the developers have an Android device. It also allows you to simulate certain functionality by simply using your laptop. The disadvantage of this is that you cannot test features which require communication amongst physical components. In the end, this is a much more difficult approach then programming an LCD and having a simpler UI do the same job an android application would be able to do.

For a C based project, specifically, with an MSP430 microcontroller, the development tools to use would be Texas Instruments' Energia IDE and their Code Composer Studio IDE. Both of these IDEs provide an efficient and simple way to write code and debug as you go. Here are some of the major differences of each IDE which will ultimately aid us in selecting an IDE that we'll use to test our major source code.

The TI Code Composer Studio includes a C/C++ compiler, a code debugger, assembly compilation, and provides a single user interface which aids in developing the application that you want. The main thing about CCS that we want to be able to utilize to its highest extent is its code debugger.

The debugger allows the programmer to see what variable holds what value during whichever iteration that the code is running. We believe this will help us find defects in our code that we would not be able to recognize otherwise. Another thing that would benefit the development of our project is the ability to program in the microcontroller assembly language. This lets the programmers work with registers.

The TI Energia IDE has one huge benefit over Code Composer Studio: it doesn't require the programmer to have to physically code flags, bit manipulations, and enable/disable watchdog interrupters. Other than that, everything else that the Energia IDE can do can also be carried out by the TI Code Composer Studio IDE.

7.2.3 Programming Languages

With a majority of the Texas Instruments microcontrollers having the ability to support programming in both their respective assembly language and the general-purpose C programming language, we will have more flexibility when programming our LCD and developing the functions of the shopping cart. Given both options, there are advantages and disadvantages to writing code in assembly and C.

When it comes to assembly, the programmer is able to control every CPU instruction to the utmost of detail. This allows the programmer to optimize any part of a program to whichever degree they want. With this advantage comes a few drawbacks. The biggest drawback is that assembly is machine dependent. Whatever algorithm that is developed for a particular microcontroller will need to be retranslated or even implemented completely differently depending on what embedded system you're using. This is where utilizing C comes in handy.

C essentially abstracts programming constructs and eliminates the issue of having specialized code. The programmer does not need to understand complicated architectures of how their microcontroller or microprocessor works. That is what the compiler is for. Not only that, but performance is greatly enhanced and sped up thanks to strategic algorithms that have been developed over the decades. Therefore, for simplicity's sake, we will be writing all of our source code in C.

7.2.4 Software Specific Testing

In the software and hardware industry, software testing involves the use of various automation tools which is typically done by a team of QA testers. This helps to upsurge efforts essential for distributing test case results and abbreviates test implementation cycles.

Testers will typically first define their scope of testing, then select an automation tool (such as Selenium, Appium, Test Studio, etc.), and finally begin automated testing development. Due to the small scale of our project, we will not be using any automation tools or a group of testers to test the functionality of our project. Instead, everything will be tested by the group members manually and repeated until everything has been perfected.

Because we don't have to worry about multiple developers working on the source code (only two programmers will be writing and modifying the code), everything will be tested on our prototype device. This includes working on two separate IDEs and manually testing each function and code block that we have.

The user interface system which will be used to control the maneuverability of Tinjac will be programmed in the general-purpose, procedural programming language known as C. We will be utilizing both the Energia Integrated Development Environment (IDE) and the Code Composer Studio (CSS) IDE to write, build, and run our source code. The source code will be integrated with the MSP430 microcontroller which will be used to control the UI system.

The UI system will further be utilized with the separate parts of Tinjac in order to calculate the shortest distance the cart can take to the closest grocery item, traverse the path, and stop/go when signaled.

In order to ensure our UI is fully compatible with Tinjac, we must perform testing in order to identify the max number of defects and bugs that we can find. This will be done by first performing a simple functionality test.

For the functionality test, we will essentially be making sure that every button and command correlate with one another on the UI. This serves to check that the functionality of each button (up, down, select, add, etc.) performs the correct action that it is supposed to. For example, we want to make sure that pressing the UP button on our UI makes the selection cursor move up the list of grocery items that we have.

Similarly, we want to make sure that pressing the SELECT button will cause the grocery item to be added to the queue so that TINJAC knows where it eventually needs to go.

Not only will we be checking the functionality of the buttons, we will be checking for the test cases that make the UI crash. This is to further explore bugs/defects that we may not have discovered. This can be done by spamming the buttons and seeing how much the software can handle before it crashes. For example, if pressing the SELECT buttons repeatedly over and over on a specific item causes the UI to crash, then we will know to go back to our source code, identify the particular defect causing this issue, and debug it.

Once the simple functionality test of the buttons has been dealt with, we will move on to testing the actual parts of the code separately. This will be mainly done to ensure each function of our source code properly does what it is supposed to do. For example, two of the functions that our cart will be able to do is 1. Detect objects/obstacles in the way of the shopping cart's traversal path and 2. Being able to readjust its position, moving out of the way of the obstacle, and going on about its route.

The way this functionality will be tested is simple. On our prototype model, we will present an obstacle in front of the shopping cart while it is en route to its destination. We shall then observe how the IR and sonar modules react and if they're able to detect the foreign object in its path. If they are able to successfully pick up the signal, then we will know that our object detection function works.

The same testing methodology will be used to see if our movement correction function works properly. Because we are only testing the software connections of our project in this specific section, the only way to test if the movement correction function works is to see if it can successfully transmit a signal to our microcontroller. This will be explained more in detail in the following paragraph.

After determining the performance functionality of the object detection function, we will test if a successful signal can be sent to the microcontroller. This can be easily tested by running some sort of LED light up code against one of our functions. For example, if the object detection function is called, we can have the red LED on our microcontroller light up. If the movement correction function is properly called after the microcontroller has received the signal sent via the object detection function, we can have the green LED light up.

The next couple sections will cover some basic codes to see if the modules we are using work as they are intended and to familiarize ourselves with writing code for these modules. They will not be implementing the end goal functions of the Tin jac they are simple tests to get the things working in the most minimal way possible usually being to turn an LED on or see if something happens. These test will include the IR sensor, Sonar Sensor, LCD module, and Motor. A solar panel test is excluded because that is all mainly handled by the solar panel charger controller and won't be taking in any input from the microcontroller.

7.2.5 Software Test: IR Sensor

This is a simple test to turn on the IR sensor and test some code to light up one of the LEDs on the microcontroller development board we are using for the cart. The IR sensors power and ground pins will be connected to the 5V pin and ground pin respectively on the microcontroller because it needs 3V - 5V to turn on. The signal pin will be connected to one of the open input pins on the dev board. The simple code that will be used to test this device will turn on an LED on the dev board when the IR sensor detects something. This should work because when the IR sensor detects something in front of it the signal pin will turn on sending a voltage to the microcontroller telling it to turn the LED on and when that voltage is gone the microcontroller will tell it to turn off.

7.2.6 Software Test: Sonar Sensor

This test is a little more complex but should accomplish the same thing as the IR sensor test. Same as the IR sensor the power and ground pins will be connected on the 5V and ground pins respectively. The trig pin will be connected to an open output pin and the echo pin will be connected to an open input pin. The code will turn on the trig pin for a couple milliseconds then turn this off for a longer period of time but still in a millisecond range.

This test allows the frequency of the device to be sent out. Then if the wave bounce back and is picked up by the device the echo pin will turn on sending a voltage to the microcontroller telling it to turn toggle the LED. This code will be continually cycling the trig pin on and off because that is the ideal way to use this sonar module that was used.

7.2.7 Software Test: LCD Module

This test is pretty simple compared to the other software test the only thing is it needs to be connected properly. The goal of this test is to use the microcontroller to turn on all the individual segments and make sure they are properly lit. this can also be used to check if any of the segments are not working properly which would require a new LCD module to be purchased. It has a power and ground on the LCD display which will be connected to the 5V and ground pins on the microcontroller and the remaining pins on the display will be connected to open output pins.

7.2.8 Software Test: Motors

The goal of this test is to use the microcontroller to turn on the motors and have them spin clockwise and counterclockwise for for a couple of seconds. In the test we will be utilizing the motor driver chip to make connecting the motor to the microcontroller easier. Once the chip is hooked up to the motor chip properly we can have the microcontroller send a voltage for a couple seconds to one of the input pins on the motor chip which will turn it one way. After that time limit is reached the voltage will stop being sent turning off the motor then we will have it send a voltage to the other input pin on the motor chip which will then turn it the opposite direction.

7.3 Demonstration Test

Once each subsystem, as well as the code itself, is completed its individual testing, it is then time to put together the pieces of the project and setup the full demonstration. For the demonstration, it was decided to use a classroom to replicate a similar real-world environment. Desks would be used to represent the isles and certain locations would be marked with were designated items will be.

Once a list of the items we want to obtain is inputted the cart will begin to calculate the shortest path and proceed to the locations. The user will be behind the cart and step away at random times to see if the cart stops which will determine if the sensors to locate when someone steps away from the cart is working. An obstacle will be placed in one of the isles and the cart will try to navigate around the obstacle. The accuracy of the codes measurement and its ability to know where it is will be tested if it can reach the destinations.

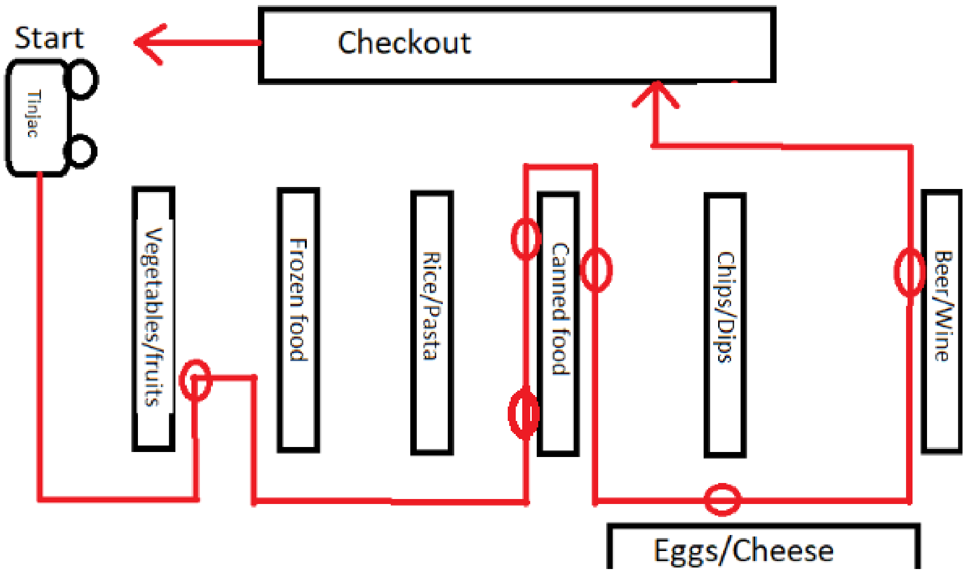

Figure 7-9

Figure 7-9 is a rough sketch of optimal pathing we envision Tinjac to be able to determine on its own. Each circle representing a “stop” needed by the shopper in order to grab all of the groceries that are needed to complete their trip to the store.

8 Administrative Content

This section of the paper will discuss how the group decided to approach meeting deadlines, managing the overall budget, splitting up the work division, and implementing potential stretch goals. Some of the dates, budget pricing, and implementation is due to change as we progress with building the project and these changes will be properly and correctly reflected.

While this section does not contain any sublayers involving the design of Tinjac itself, without following the listed steps below we would definitely fail to accomplish anything.

8.1 Milestones

#	Task	Start	End	Member Assigned
Senior Design I				
1	Agree on Ideas	5/14/18	5/30/18	Group 13
2	Divide and Conquer Report	5/30/18	6/8/18	Group 13
Research and Design				
4	Initial Design	6/8/18	6/15/18	Group 13
5	Schematics	6/15/18	7/1/18	James / Edgar
7	PCB Layout	6/15/18	7/25/18	James / Edgar
8	UI Interface (LCD)	6/15/18	7/25/18	John / Shekh
9	MC Circuit Analysis	6/15/18	7/25/18	James / Edgar
10	Signal Trigger Events	6/15/18	7/25/18	John / Shekh
Finalization				

12	Final Documentation	6/15/18	7/30/18	Group 13
13	Order/Test Parts	6/30/18	7/10/18	Group 13
Senior Design II				
14	Build Prototype	8/10/18	8/31/18	Group 13
15	Test / Adjust	8/12/18	TBD	Group 13
16	Finalize Prototype	TBD	TBD	Group 13
17	Peer Presentation	TBD	TBD	Group 13
18	Final Presentation	TBD	TBD	Group 13

Figure 8-1

Milestones for this project were established during the first week of senior design. Because this is during the summer semester, the group's main focus was designing the project and working on the documentations first, since the semester is four weeks shorter than a typical Fall or Spring semester. Therefore, most of the end dates seen below regarding the separate components of the project have yet to be determined. The construction of the project will begin in Senior Design II and the end dates will be further determined and updated once that semester begins.

In the above figure (Figure 8-1), our schedule was unanimously agreed upon by everyone within our group. Each starting date and finish date that has passed prior to the final completion of this document has been verified as we use this schedule to continue map out when we would need to progress on specific divisions of Tinjac. This has been split into two sections - Senior Design I, and Senior Design II. This helps us break down when we should be meeting the milestones we need to in the correct time frame.

8.2 Budget and Finance Discussion

The budget of this project isn't very large since there isn't any company sponsoring the project. Costs will be evenly split amongst the team members to ensure spending equality. Our overall goal is to construct and implement every portion of this project in the cheapest yet most efficient way possible to demonstrate affordability to our main audience. It would not be beneficial to real life customers and stores if our product was too expensive for them to utilize. In order to meet this financial goal, certain functionalities, parts, and specifications must be sacrificed.

Item Name	Quantity	Price/Part	Cost/Order
High-Torque Motor	3	≈ $15	$45
MSP 430	1	≈ $20	$20
Compact Size Shopping Cart	1	≈ $50	$50
Solar Panel	4	≈ $10	$40
High-Capacity Rechargeable Battery	1	≈ $20	$20
Sonar Module	1	≈ $2	$2
Infrared Module	1	≈ $15	$15
L293D Motor Driver Chip	1	≈ $1	$1
Miscellaneous Wires/Gears	N/A	≤ $10	$10
Small PCB	1	≈ $30	$30
LCD Module	1	≈ $10	$10
Total:	15		$258

Figure 8-2

Figure 8-2 shows the financial breakdown of the components required for the initial prototype design of Tinnjac. We constantly refer to this table to keep track of our financial division on subsystems of Tinjac, and what we can afford to allocate more money into should we need additional or more efficient parts to add into that specific subsystem.

8.3 Stretch Goals

Tinjac is composed of several subsystems that will require a perfect configuration in order to accurately weave through a generated pathway. However, if we are able to accomplish each one of the core goals and create Tinjac to be as accurate and as reliable as we wish, then we will set out to achieve even further goals to make Tinjac even more amazing.

Each one of these stretch goals were made to be accomplished if we were able to achieve everything else, however we still made these goals while keeping the concept of “attainable” in mind. Each of the stretch goals that we added we only did so if we believed that we could accomplish them with any spare time we had while designing Tinjac.

The following sections break down each specific stretch goal, its objective it is attempting to accomplish, and how we attend on meeting the goal. We also included *why* we thought each additional goal may be a great addon to Tinjac.

8.3.1 Mobile Application GUI

An alternative to the physical UI that we designed for Tinjac, was to create an app that could be downloaded on a shopper’s smartphone so they can add whatever they wished directly on their own phone, which would make the cart be able to be almost completely touch free. We would utilize the object-oriented language of Java to create the GUI along with the application. Having a Java/Android environment would allow for a better user experience and would expand the functionalities of our project by a much larger scope. But along with knowing how to create a well-functioning GUI system, we would also need to use C to additionally program our microcontroller. This puts unnecessary strain on the developers of the project, who could simply go with an all C based focus for the entire project (UI system, LCD functionality, sensor controls, motor controls, etc.).

While this idea was deemed optimal at first, we continued to discuss the implementations of how to connect it with the specific cart. We foresaw issues with using an app and having a specific cart (if there were several Tinjac’s created) as well as forcing the shopper to download an app that they may not wish to have on their mobile device may also prove to be more negative than positive.

Along with that, having an app with the list of grocery items would mean implementing a database of grocery items and that would require the computer engineering students to work on connecting the application to the database server, which would take time away from constructing the actual shopping cart. Even with all those conflicting points, we agreed that if we had extra time we would design the app and try to implement it as simply and as effectively as possible.

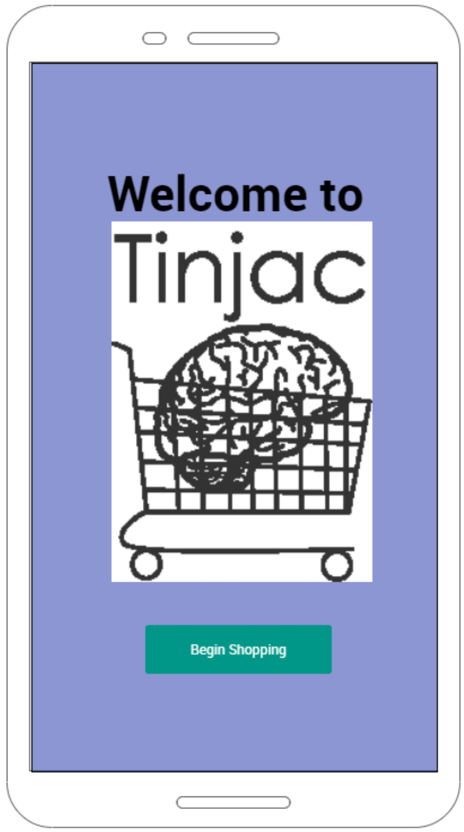

Figure 8-4

Figure 8-4 shows how a basic UI would display on any mobile device. Should we be able to implement this stretch goal, the app will be able to display numerous features all from the touch of your own phone or other smart device.

8.3.2 Call Button / Cart Summon App

Tinjac will be designed to start at a specified point and calculate the desired path from the entryway, however a more convenient setup for the shopper will be to setup a pedestal that the user can hit a button on. Then the cart will drive itself all the way to the podium (from the cart return section) and wait for the user to begin their shopping experience.

We also discussed about integrating this idea into the app design mentioned in the previous section, allowing the shopper to summon a cart from the cart return section simply by pulling up the app and clicking on a button within the app itself.

We initially came up with this concept on a way for us to begin the pathing, however we quickly came to an agreement that this method would be much more work than was required to create an exquisite shopping experience through Tinjac. We decided that if we could complete our main objectives in time we could come back to this idea and try again to make it work.

8.3.3 Alternative Following System

Our cart is setup to stop moving when the shopper walks away from the cart. We also decided that implementing a second way the cart can traverse the path would be to *follow* the shopper, rather than move on its own. If we decided to set the cart up in this way, it would not create a path, but instead freely move about and follow the shopper specifically. This would be done by having the customer wear some sort of device, possibly a belt, or an infrared light so that the sensors on the cart could act as a signal receiver somewhat, and then follow where the projected signal is coming from. This means that the customer is always in front of the cart, with the cart following closely behind which allows the user to walk around wherever and whenever they would wish.

An idea for a second alternative mode would be that instead of the customer walking in front of the cart, they walk behind the cart instead and use hand motions to let the cart know where they want to go. For example, if the customer wants to make a hard-right turn, the customer would put their hand very close to the sensor on the right side of the cart, but if they want to make a slow right turn, the customer would put their hand more further away from the sensor.

We agreed these implementations were more than ideal. However, with all the other subsystems needing reconfiguration to even allow it, we decided that we would attempt to set the cart up to do so only if we were able to achieve total efficiency over the primary mode of travel.

8.3.4 Total Price Estimator and On-Board Check Out

While the shopper is following the cart and grabbing the items they wish to put into their cart, we came up with a great add-on to Tinjac that we would love to implement if we achieve all previous tasks. We would implement a system that will automatically scan every item placed into the cart and add it to the total cost of the groceries, with barcode scanner technology. Since the cost total would be able to be updated consistently, once the total amount is calculated, the cashier would be able to check the customer out much quicker than having to scan all the items at once.

An additional feature that could be added on the app that’d work alongside the item scanner is if a customer suffers from a dairy allergy (or any other allergy), then when an item is scanned that contains dairy, it will automatically send them a warning, informing them about it. A warning could also be displayed for items with a short expiration date.

To go even a further step beyond, we could implement a feature where the shopper can just pay with their credit/debit card on the shopping cart itself. This would be done using the typical self-checkout technologies that are prevalent in today's society. This consists of installing a credit/debit card reader on the cart itself but that would introduce more complex systems where we'd need to design a way to transmit a customer's credit card information and transact data to their respective banks in a secure manner.

The agreement about these ideas were unanimous when deciding if it would be an ideal addition to Tinjac, however, we also agreed we would only attempt to implement this once we have reached every other primary goal that we aim for.

Figure 8-5 ~ EA Exclusive Gaming Peripherals

8.3.5 High Weight Limit

The more groceries that get added into Tinjac, the harder it will be to move the cart forward. We wish to aim for a nearly-limitless cart in regards to this aspect, and design Tinjac to function properly no matter how much you put into it (within logical reason), so the cart can support those shoppers who purchase a higher amount of groceries than the average person.

While this stretch goal is not as flashy as the other add-ons we wish to improve Tinjac with, it is the most applicable one in regards to usage. While we will be able to get Tinjac to function properly without "accomplishing" this specific goal, being able to function without being restricted on what you can place within the cart is an ideal situation for both the shopper and the store.

8.3.6 Enclosure

If this product were to advance past senior design and become a value invention, it would most likely need a very presentable appearance without wires hanging everywhere. This will also protect the devices from outside harm that them being exposed could potentially cause.

A good idea for this since the design is a cart with sensors all around it, and hence it would have wires coming from all sides, is that instead of the regular metal beams that a shopping cart has, it could have a hollow inside instead so that the wires from the sensors could travel through them.

At the bottom of the cart there would be a plastic enclosure to include the battery, MCU, printed circuit board, wires from the solar panel (solar panel would be positioned right above the enclosure), and then the wires from the sensors that were are passed through the hollow beams would all come through the bottom and meet inside the enclosure.

A singular hole will be placed on the top of the enclosure so all the wires could go through the same place. With the enclosure, no wires would be visible to anyone which minimizes the risk of someone breaking the technology.

Figure 8-6 ~ High-Tech Polymer-Based Containment Capsule

In the above figure (Figure 8-6) is a specific example of how we can enclose a microcontroller or any PCB's used in our system. This will help centralize all the wiring for the major components while also helping safeguard it from external hazards.

8.3.7 Advertising

One of the ways that companies could largely benefit from this invention is for them to utilize the user interface's screen on the cart. A way that this could happen is that once a cart enters an aisle, the user interface will display the best sales and deals occuring in that specific aisle.

Once the sales and deals are displayed on the screen, the user will be able to choose a deal that they find enticing and then the cart will add the item to the already calculated path. This would be a great way for stores to promote specific products they want to sell and also for the store to sell more products through great product deals. This wouldn't be too difficult to implement, all it would require is an addition of a wi-fi module to be able to connect to the store's internet once a day to download the update containing the new products deals.

Figure 8-7

Figure 8-7 shows one specific example of how we could potentially use tactifully placed ads to both generate revenue for the store that opts in for Tinjac, as well as the ads could also be used to house more vulnerable components such as the microcontroller or connecting wires.

9 Conclusion

Discussed below will include the experience that all the team members have gained, both collectively and individually. We will at long last provide the closing statement of our Senior Design project. Designing this project has, to say the least, left an impact on our engineering careers. Throughout the Tinjac's design and implementation we have encounter just as many mistakes as we have successes. It is through this mistakes that we are able to finalize onto a conclusion, as each jump over our hurdles granted us a new perspective.

Tinjac, whether considered by others as ingenious or simply average, has been nothing short of a fantastic learning experience. The subsystems that compose the product as a whole each stood as their own puzzle, rewarding us each with specific knowledge from completing the tasks required to solve said puzzle.

9.1 Experience Gained

One of the most important skills that have been learned while participating in the making of this design is collaboration. There have been lots of group projects before senior design, but definitely nothing of this magnitude previously. This project requires the utmost trust of every person in the group with each other since everyone has a certain responsibility they hold by being in the group. Collaborating in this project means being able to equally divide and section off the work to each member and then ensuring that everyone gets their share done. Why this demands the importance of collaboration so much is because every section requires information from other sections, and so each group member needs to be able to come together in a civil manner in order to make sure that every section is able to be completed to make a whole, and uniform product. This is such an important skill to have because of future jobs that will require heavy collaboration in the workplace with possibly even larger projects, and to be able to gel well with random collaborative team members is certainly a skill that every project needs to become a success.

Besides lab reports, engineers have never really had to write much in their classes which is unfortunate since many jobs require a very professional style of writing, and it's something that is common for engineers to lack in. This paper has a minimum writing requirement of 120 pages, and divided by 4 people that's at least 30 pages each. That page requirement is definitely a scare to any engineer, and so to be able to overcome this, has been a great feat. Being able to complete this design paper has shown everyone in this group that long page requirements aren't nearly as scary as they sounded a few months ago, especially when the work is divided into little swallowable sections. This is very vital and important that this fear of long page requirements has been overcame because no matter what job, or field someone is working in, there will most likely be some form of writing required as writing is a skill that can be used throughout someone's whole life.

Lots of technical experience has been gained from this design as well. To create the schematics, the programs, Eagle, as well as EasyEDA had to be learned. Eagle is a CAD software and this program is able to produce universally accepted schematics, and printed circuit board layouts of the schematics, but that won't be learned until the second half of senior design. EasyEDA also was learned so that later, printed circuit board layout from the schematic can be exported directly to a specific company that's compatible with this program. Not to mention, EasyEDA is also a little easier to use and understand more than Eagle. Hands on testing was also experienced by ensuring that all the parts work exactly how we all envisioned them, but of course this hands on experience will be expanded greatly on, later in the second half of senior design. There has been lots of experience in the circuit class's laboratory sections but generally, the circuits that were required to build for the class were just to show how specific components would work in a perfect imaginary setting. But the main purpose of these labs were to set up the students to implement real circuits and designs, and that's exactly what our group was able to experience first hand. While testing parts, we had to use different power sources, add resistors to make sure our equipment didn't become fried, as well as measuring the output voltages and currents with a multimeter. It's a very different ball game when actually having to put the knowledge we have learned into a real life setting and experiment.

9.2 Closing Statement

In this ever-changing society where technology affects the way we live, learn, and work, there will always come advancements that aim to facilitate the many processes of human behavior. Our self-driving shopping cart aims to help those that require assistance by removing the physical burden of having to push around a shopping cart and by allowing the customer to simply input the desired location of where it wants the cart to go.

The cart will then traverse through the store, going to each of the specified locations and will have the ability to stop depending on if the user is behind the cart or not. The overall objective of this project is to essentially introduce a prototype that can potentially be adopted by stores in order to help make shopping easy for their customers who are handicapped, debilitated, or injured.

With our even split in engineering disciplines (computer and electrical engineering), we were able to implement software and hardware features for our project. We started off by researching similar projects that were either already on the market or being prototyped by companies. This helped us set up a base prototype design for our shopping cart where we decided what features to implement or not depending on if it had been done already in the past. Once we figured out the general design, we had to do more research on all the technologies that would be implemented to make our shopping cart autonomous. Once we had all that research done, selecting the parts consisted of balancing out how well we want our features to function versus how much we want to spend (since we have a limitation on our budget).

Whether it involved running into roadblocks, deciding on a new part to implement, arguing over which method was the best to code, or even simply trying to decide which goal to add or remove from the project; we evolved our way of thinking while designing Tinjac.

Tinjac is not just a cart, it is a way of life. We created this idea to help improve the lives of others while trying to maintain a hold on the evolving technology around us. While there are those who would falsely claim that Tinjac is just some unnecessary invention, it is those same people who are blind to the possibilities of advancement and expansions. Through Tinjac so much more can be accomplished in a person’s day-to-day shopping experience. Capitalism is the heart and soul of this great nation, and one of the reasons that our country was able to thrive throughout harsh decades.

Tinjac is not just a cart, it is so much more than a device to push around the food you wish to purchase. It is a statement of importance. It shows the rest of the world that just because you can do something, doesn’t mean you should. Just because you are able to push the cart to the exact spot you want, doesn’t mean it is you who should be doing the pushing. With Tinjac helping people they can spend less time using their physical strength and more time thinking. They can use this free time to think about how they can improve their own lives, in the exact same way that Tinjac has improved their shopping experience.

Tinjac is not just a cart,
All hail Tinjac.

10 Appendices

Appendices A - Schematic Copyright Permissions

Figure 3-5 & Figure 3-6 (Awaiting Response):

Blog Downloads Community Help Forums Education

CONTACT US

Our help section and FAQs answer most questions about Raspberry Pi and about getting started with our hardware and software. Check these to find answers quickly.

We do not provide support with using your Raspberry Pi by email. Please make sure that your question isn't already answered in our help section, our FAQs, or our discussion forums, and if you're still stuck, ask your question in a forum post. Our engineers and thousands of Raspberry Pi experts are waiting there to help beginners and advanced users alike.

Thanks for contacting us! We'll get in touch with you shortly.

Figure 3-20 (Pre-Approved): "Permission is hereby given to copy any information contained herein, provided it is clear on the copy that the information came from Allegro."

Figure 3-21 (Pre-Approved for Non-Commercial):

Copyrights

Texas Instruments is pleased to provide the information on these pages of the World Wide Web. We encourage you to read and use this information in developing new products.

TI grants permission to download, print copies, store downloaded files on a computer and reference this information in your documents only for your personal and non-commercial use. But remember, TI retains its copyright in all of this information. This means that you may not further display, reproduce, or distribute this information without permission from Texas Instruments. This also means you may not, without our permission, "mirror" this information on your own server, or modify or re-use this information on another system.

Figure 3-22 (Approved):

Copyright © 1995-2018 TOSHIBA CORPORATION, All Rights Reserved.

This TOSHIBA WWW Server provides a variety of information concerning products, information service, technical service, news releases and the like ("Documents"). TOSHIBA and/or its affiliated companies own copyrights in these Documents, unless otherwise expressly stated.

These Documents are protected by copyright law and other applicable laws. Unauthorized use is strictly prohibited and you may use the TOSHIBA WWW Server only under the following conditions:

1. You may use, copy or download for only internal and non-commercial purposes.
2. When you copy or download, you must include this copyright notice and conditions.
3. You are prohibited to exploit, such as modify, distribute, publicly transmit (including making transmittable) or display and so on, any copies of Documents.

Figure 3-25 (No Copyright Claim):

Document citation:
https://www.openhacks.com/uploadsproductos/eone-1602a1.pdf

Figure 5-1, Figure 5-2 & Figure 5-3 (Pre-Approved for Non-Commercial):

Copyrights

Texas Instruments is pleased to provide the information on these pages of the World Wide Web. We encourage you to read and use this information in developing new products.

TI grants permission to download, print copies, store downloaded files on a computer and reference this information in your documents only for your personal and non-commercial use. But remember, TI retains its copyright in all of this information. This means that you may not further display, reproduce, or distribute this information without permission from Texas Instruments. This also means you may not, without our permission, "mirror" this information on your own server, or modify or re-use this information on another system.

Figure 6-2 & 6-5 (Pre-Approved for Non-Commercial):

Copyright

Unless otherwise specified, the copyrights in all copyrightable subject matter on the Site are either owned by us or used lawfully by us. To the extent that we have the right to do so without compensation to third parties, and except for materials or information specifically provided under other terms, we grant you permission to copy or otherwise download from the Site, information and materials (including related graphics), provided:

- The materials are for your internal use only, and
- Any copies of materials or portions thereof must include the copyright notice specified on the Site.

You may not copy or display for redistribution to third parties any portion of the Site without our prior written permission. Documents we post on the Site may contain other proprietary notices or describe products, services, processes or technologies owned by us or third parties. Nothing contained herein shall be construed by implication, estoppel or otherwise as granting to the user a license under any copyright, trademark, patent or other intellectual property right of us or any third party.

Figure 6-3 & 6-4 (Pre-Approved for Non-Commercial):

Copyrights

Texas Instruments is pleased to provide the information on these pages of the World Wide Web. We encourage you to read and use this information in developing new products.

TI grants permission to download, print copies, store downloaded files on a computer and reference this information in your documents only for your personal and non-commercial use. But remember, TI retains its copyright in all of this information. This means that you may not further display, reproduce, or distribute this information without permission from Texas Instruments. This also means you may not, without our permission, "mirror" this information on your own server, or modify or re-use this information on another system.

Figure 7-6, Figure 7-7 & Figure 7-8:

D damla.turgut@gmail.com on behalf of Damla Turgut <turgut@eecs.ucf.edu>
Fri 7/27, 2:47 PM

Sure, you are most welcome to reference them as long as you provide the reference at the end of the presentation just like you would do for any other references.

Sincerely,

Damla Turgut

...

SA Shekh Arefen
Fri 7/27, 11:27 AM
turgut@cs.ucf.edu

Reply all

PA-chapter02.ppt
2 MB

Download   Save to OneDrive - Knights - University of Central Florida

Good Morning Dr. Turgut,

I am a previous student that was enrolled in your COP 4331 course in the Fall of 2017 semester. I was wondering if it is okay if I reference some of the information on your slides (chapter 2) in my group's Senior Design Paper?

Appendices B - Figure Citation:

Figure 3-1:

"What Impact Will the Amazon Go Store Have On SMB Retailers?" *Software Advice*™, 14 Feb. 2017, www.softwareadvice.com/resources/amazon-grocery-store/.

Figure 3-2:

Wilson, Richard. "CES: Autonomous Cars and the Sensors to Make Them Safe." *Electronics Weekly*, 3 Jan. 2017, www.electronicsweekly.com/market-sectors/automotive-electronics/ces-autonomous-cars-sensors-make-safe-2017-01/.

Figure 3-3:

"MSP430FR2355 (ACTIVE)." *PC & Notebooks IC Solutions| Overview| TI.com*, www.ti.com/product/msp430fr2355#parametrics.

Figure 3-4:

"MSP432P401M (ACTIVE)." *PC & Notebooks IC Solutions| Overview| TI.com*, www.ti.com/product/MSP432P401M.

Figure 3-7:

"Microchip Technology ATmega328 8-Bit AVR® Microcontrollers." *Mouser Electronics - Electronic Components Distributor*, www.mouser.com/new/microchip/atmelatmega328/.

Figure 3-8:

"Microchip Technology ATmega328 8-Bit AVR® Microcontrollers." *Mouser Electronics - Electronic Components Distributor*, www.mouser.com/new/microchip/atmelatmega328/.

Figure 3-9:

"UM1472 User Manual." *Discovery Kit with STM32F407VG MCU*, www.st.com/content/ccc/resource/technical/document/user_manual/70/fe/4a/3f/e7/e1/4f/7d/DM00039084.pdf/files/DM00039084.pdf/jcr:content/translations/en.DM00039084.pdf.

Figure 3-11:

"E18-D80NK Long Range Adjustable IR Sensor." *e-Gizmo Mechatronix Central*, www.e-gizmo.net/oc/index.php?route=product/product&path=78&product_id=1255.

Figure 3-12:

IR Sensor Module for Remote Control Systems. Vishay, www.vishay.com/docs/82486/tsmp58138.pdf.

Figure 3-13:

"Arduino Lesson." *IR Obstacle Avoidance Sensor*, Osoyoo, osoyoo.com/2017/07/24/arduino-lesson-obstacle-avoidance-sensor/ .

Figure 3-15:

"SENSOR TOKO ROBOT." *SRF10 - The Worlds Smallest Dual Transducer Ultrasonic Ranger*, sensortokorobot.blogspot.com/2011/06/srf10-worlds-smallest-dual-transducer.html.

Figure 3-16:

Adafruit Industries. “Maxbotix Ultrasonic Rangefinder - LV-EZ3.” *Adafruit Industries Blog RSS*, www.adafruit.com/product/981.

Figure 3-17:

“Arduino Based Distance Measure Box.” *Ardumotive Arduino Greek Playground*, www.ardumotive.com/distance-measure-box.html.

Figure 3-24:

Winstar Display Co., embedded.ifmo.ru/sdk/sdk61/components/lcd/WH1602B-YGK-CP.pdf.

Figure 3-29:

“Anself.com.” *Anself*, www.anself.com/p-h17919-1.html.

Figure 3-30:

ALLPOWERS, iallpowers.com/index.php?c=product&id=371.

Figure 3-31:

“10Amp PWM Solar Charge Controller with LCD Display | HQST Solar.” *Www.hqsolarpower.com*, www.hqsolarpower.com/10-Amp-PWM-Charge-Controller-with-LCD-Display-p/hqst-ctrl-pwm10-lcd.htm.

Figure 3-32:

“Solar Charge Controller Waterproof, Y-SOLAR 5A Solar Panel Controller 6V/12V Auto 24Hours Load on IP67 Waterproof Solar Controller for Lead-Acid Battery Charging and Discharging Control : Garden & Outdoor.” *Amazon*, Amazon, www.amazon.com/Controller-Waterproof-Y-SOLAR-Lead-acid-Discharging/dp/B00XTQ76WW/ref=as_li_ss_tl?tag=shopperz_origin3-20&ascsubtag=102433879-2-&SubscriptionId=AKIAJO7E5OLQ67NVPFZA.

Figure 3-35:

“EXpress6000 Two-Tier Wire Convenience Shopping Cart.” *Versacart*, versacart.com/shop/shopping-carts/convenience-carts/express6000/.

Figure 3-36:

“Sandusky FSC3012 Folding Shopping Cart, 66 Lbs Capacity: Service Carts: Amazon.com: Industrial & Scientific.” *Amazon*, Amazon, www.amazon.com/Sandusky-FSC3012-Folding-Shopping-Cart/dp/B01BLEMDTE

Figure 3-37:

“Goplus Folding Shopping Cart Jumbo Double Basket Perfect for Grocery Laundry Book Luggage Travel W/Swivel Wheels : Office Products.” *Amazon*, Amazon, www.amazon.com/Goplus-Folding-Shopping-Perfect-Grocery/dp/B072M4HDM8.

Figure 4-1:

“IEC 61215: What It Is.” *IEC*, https://www.nrel.gov/docs/fy12osti/54714.pdf.

Figure 4-2:

“IPC Standards Tree.” *IPC*, IPC, www.ipc.org/4.0_Knowledge/4.1_Standards/SpecTree.pdf.

Figure 8-5(a):

“Barcodes,Inc.” *BarcodesInc*, www.barcodesinc.com/motorola/ls2208.htm.

Figure 8-5(b):

“TranXat Excellence in Every Transaction... Trust in Every Way.” *TranXat Merchant Services*, www.tranxat.com

Figure 8-6:

“SimcoBox Disclosures - Plastic Enclosures Blog.” *SIMCO Disclosures*, SIMCO Disclosures, www.simcobox.com/blog/page/2/.

Figure 8-7:

“McDonald's: Burgers, Fries & More. Quality Ingredients.” *McDonald's*, McDonald's, www.mcdonalds.com/us/en-us.html

Appendices C - Works Cited:

[1] Bishop, Todd. “How 'Amazon Go' Works: The Technology behind the Online Retailer's Groundbreaking New Grocery Store.” *GeekWire*, GeekWire, 14 Jan.

2017, www.geekwire.com/2016/amazon-go-works-technology-behind-online-retailers-groundbreaking-new-grocery-store/.

[2] Dignan, Larry. “Amazon Go: The Impact on Human Jobs, Retail Innovation, Amazon's Bottom Line.” *ZDNet*, ZDNet, 23 Jan. 2018, www.zdnet.com/article/amazon-go-heres-a-look-at-the-impact-on-human-jobs-retail-innovation-amazons-bottom-line/.

[3] Menegus, Bryan. “Walmart Considering Smart Carts, Drone Assistants.” *Gizmodo*, Gizmodo, 22 Mar. 2018, https://gizmodo.com/walmart-considering-smart-carts-drone-assistants-1823991966

[4] “Why Walmart Is Testing Robots In Stores-and Here's What It Learned.” *Fortune*, Fortune, fortune.com/2018/03/26/walmart-robot-bossa-nova/.

[5] *MSP432P401R SimpleLink™ Microcontroller LaunchPad™ Development Kit (MSP- EXP432P401R)*. Texas Instruments, www.ti.com/lit/ug/slau597f/slau597f.pdf.

[6] Armstrong, James. “How Do Driverless Cars Work?” *The Telegraph*, Telegraph Media Group, 1 July 2016, www.telegraph.co.uk/cars/features/how-do-driverless-cars-work/.

[7] *Power Supply Safety Standards, Agencies, and Marks*. CUI Inc, www.cui.com/catalog/resource/power-supply-safety-standards-agencies-and-marks.pdf.

[8] “MSP430FR2355 (ACTIVE).” *PC & Notebooks IC Solutions| Overview| TI.com*, www.ti.com/product/msp430fr2355#parametrics.