Solar Powered Golf Cart
Group 9
December 4, 2014

Jake Bettis
Jacob Kreuger
Matt Roland
Matt Tourtelot
With support from Leidos
Table of Contents

1 Executive Summary ............................................................................................................ 1

1.1 Initial Proposal .................................................................................................................. 2
  1.1.1 Project Description ....................................................................................................... 2
  1.1.2 Project specifications and requirements ....................................................................... 2
  1.1.3 Project Block Diagrams ............................................................................................... 3

1.2 Project Significance and Motivation .................................................................................. 6

2 Research ............................................................................................................................... 7

2.1 Heads-Up Display (HUD) .................................................................................................. 7
  2.1.1 Navigation .................................................................................................................... 7
  2.1.2 Golf Cart Status .......................................................................................................... 9
  2.1.3 Mode Selection .......................................................................................................... 11
  2.1.4 Touch Screen .............................................................................................................. 12
  2.1.5 Microcontroller .......................................................................................................... 13
  2.1.6 Android Application ................................................................................................... 16

2.2 Motor Controller ................................................................................................................ 20
  2.2.1 Throttle Sensor .......................................................................................................... 20
  2.2.2 Software Control ........................................................................................................ 22
  2.2.3 Circuit Design ........................................................................................................... 25
  2.2.4 Direction Control ........................................................................................................ 32
  2.2.5 Quadrature Encoder Interface (QEI) ........................................................................... 34

2.3 Charge Controller .............................................................................................................. 35
  2.3.1 Maximum Power Point Tracking (MPPT) .................................................................. 36
  2.3.2 Sensors ...................................................................................................................... 40
  2.3.3 Charging Techniques ................................................................................................. 42
  2.3.4 Solar Panels .............................................................................................................. 44

2.4 PCB Design ....................................................................................................................... 47
  2.4.1 PCB Composition ........................................................................................................ 48
  2.4.2 PCB Software ............................................................................................................ 49
  2.4.3 PCB Manufacturer ...................................................................................................... 49

2.5 Batteries ............................................................................................................................ 50
  2.5.1 Options ....................................................................................................................... 50
4.3.2 Phase 2 .................................................................................. 87
4.3.3 Phase 3 .................................................................................. 88
4.3.4 Phase 4 .................................................................................. 88
4.4 Charge Controller .................................................................... 89
  4.4.1 Phase 1 ................................................................................. 89
  4.4.2 Phase 2 ................................................................................. 89
  4.4.3 Phase 3 ................................................................................. 89
5 Testing .......................................................................................... 90
  5.1 HUD ......................................................................................... 90
    5.1.1 Android Application Testing ......................................................... 90
    5.1.2 BeagleBone Black and MTK MT3339 GPS Module PCB Testing .... 95
  5.2 Motor Controller ...................................................................... 98
    5.2.1 Power Board........................................................................... 98
    5.2.2 Motor ..................................................................................... 98
    5.2.3 MCU ...................................................................................... 98
    5.2.4 Test Procedure ..................................................................... 100
  5.3 Autonomous Features .............................................................. 100
    5.3.1 Golf Cart Speed and Braking ....................................................... 100
    5.3.2 Rangefinder Sensors ................................................................. 101
    5.3.3 Threshold Distances and Velocities ............................................ 101
    5.3.4 Cruise Control ..................................................................... 102
    5.3.5 Driver Notification ................................................................ 102
  5.4 Charge Controller .................................................................... 102
    5.4.1 Computer Simulation .............................................................. 102
    5.4.2 Lab Tests .............................................................................. 103
    5.4.3 System Tests ........................................................................ 103
6 Appendix ...................................................................................... 104
  6.1 Reprint Permission Emails ....................................................... 104
    6.1.1 Technicians Manual ............................................................... 104
    6.1.2 Open Revolt ....................................................................... 104
    6.1.3 Google Maps API License .................................................... 105
    6.1.4 Test Motor .......................................................................... 105
1 Executive Summary

Renewable energy has been growing rapidly in the recent years with products such as home solar panels that consumers use to avoid purchasing power from the grid that utilizes nonrenewable sources. Markets such as consumer solar power has caused a huge increase in the demand for engineers that know how to utilize these renewable sources of energy. Our group has a very wide scope of interests which is what sparked our interest in creating a solar powered golf cart. This project allows our group to focus on different areas such as solar panel battery charge controllers, golf cart motor controllers, and a HUD using a touch screen and microcontroller. Each member gets a large portion of design work and learns a lot while creating something that is environmentally friendly. The goal of this project is to create an energy efficient golf cart that is capable of running solely on the solar panels and possibly with some help from an external outlet if time is a concern. A HUD will give the user many options such as efficiency mode, directional help, and status of the golf cart. The first component of this project that users will see if the HUD. This will provide the user with options as to what efficient mode they wish to be in, and will be able to type in a destination and receive directions to it. Google maps will provide the shortest route to the destination. A button in this menu will give the user a view of the status of the batteries and estimated distance remaining. The second component is the custom motor controller. It will provide an energy efficient way to operate the golf cart in a manner that does not accelerate jumpy. This controller will interface with the HUD to determine what mode has been selected by the user and adjust its power output accordingly. Pulse width modulation will be used to allow for accurate control of energy output. A typical golf cart just uses a variable resistor in the pedal to determine how much current is passed through to the motor which is actually very inefficient because variable resistors constantly use energy. The charge controller is the final component of this project which will monitor the power output from the solar panels and charge the batteries without damaging them or creating a fire from overfilling. There will be a communication line to tell the HUD how full the batteries are. It will also charge the golf cart while it is running. This allows for extended range capabilities on sunny days. Each component of this project will have to work together in order to achieve the main objective of creating an efficient vehicle.
1.1 Initial Proposal

The initial proposal lists this project's objectives, goals, specifications, and requirements.

1.1.1 Project Description

The main objective of this project is to design and build a solar-powered, energy efficient electric golf cart inspired by and using resources from a previous Senior Design project. The desired outcome of this development is to have an electric golf cart that will efficiently harvest power from sunlight in order to power the electric motor and onboard electronic systems while providing the user with options for total energy consumption. In order to reach this outcome, the cart will implement the following three different modes of energy operations: standard, max performance, and max efficiency. The standard mode of energy operation will provide the motor with enough energy for 80% of maximum output as well as supply the onboard electronics with enough power for operations and medium brightness levels. When the user switches to the max performance mode of operation, the motor will receive enough power for 100% maximum output and electronics will perform operations while at maximum brightness levels. If the max efficiency mode is selected, then the engine will only receive enough power necessary to run at 60% output and onboard electronics will still perform operations but displays will be set to a more dim brightness. The golf cart power monitoring system will automatically switch to max efficiency mode if it should detect the battery charges dropping below 25% in order to conserve power for the longest amount of time possible. The power that will be required to run the cart’s motor and onboard electronics will be supplied by batteries that will receive the necessary amount of charge from the solar panels or a wall outlet. There will be a monitoring system that will accurately display the remaining power in the batteries as well as check for any defects in the battery power storage. The onboard electronics will feature two touch-screen displays for various user information display. One screen will display power mode options, charge remaining, and current speed. The other display will have a GPS system to provide the user with a map of their current location and also give any necessary navigational directions based on user location.

1.1.2 Project specifications and requirements

The following is a list of specifications and requirements for this project.

1. Must have a top speed of at least 15 mph
2. Must have 3 modes of operation allowing for user control of modes
3. Must run off of a 36V or 48V battery storage bank
4. The batteries must be able to charge from solar panels or a wall outlet
5. Must automatically go into power saving mode at 25% battery capacity
6. Must provide navigational aid to user
7. Must provide charge remaining, power mode options, range left, and current speed
1.1.3  Project Block Diagrams

The following block diagram shows each portion of the project visually and lists a member responsible for each part.

<table>
<thead>
<tr>
<th>Color</th>
<th>Member</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>Jake</td>
</tr>
<tr>
<td>Orange</td>
<td>Jacob</td>
</tr>
<tr>
<td>Green</td>
<td>Matt T.</td>
</tr>
<tr>
<td>Blue</td>
<td>Matt R.</td>
</tr>
</tbody>
</table>

Table 1 Group Member Responsibilities

At this stage in the development, every block is assumed to be designated as “To be acquired”. Our project can be broken down into a few sections. System control, power systems, and software. The MCU makes a decision of how much power to apply to the motor based on the speed setting, position of the pedal, and percent of battery left. The battery monitor will calculate the range left on a charge and report it to the status display. The speed setting is decided by the user through display one touch screen interface.

1.1.3.1 System Control

Figure 1 shows an overview of the system controls for this project.
### 1.1.3.2 Power Systems

The charge controller will accept power inputs from either the solar panels or a wall outlet and determine how much power to allow through to the battery bank. This battery bank then supplies power to multiple DC-DC converters that power various microcontrollers and the displays. This battery bank will also supply voltage to the motor through a relay that is either turned on or off by the key ignition switch. This means that the golf cart will truly be using no power when it is turned off. Each voltage converter used has to be capable of handling at least 36V input and possibly higher depending on what battery configuration is decided upon. The relay also has to be capable of handling a large amount of current. Motors initially require a large surge of current to accelerate in the order of hundreds of amps. If a low quality relay is used, then it will not be efficient and create a lot of heat which will lead to more problems later on. An illustration of this is given in Figure 2.

![Figure 2 Power Systems Overview](image-url)
1.1.3.3 Software

The input from the gas pedal is sent to the motor controller’s microcontroller which outputs a corresponding PWM signal to allow for variable speed. The width of the PWM signal will also depend on what mode is selected by the user from the HUD. Current sensors will be used to avoid an overcurrent situation that could break the motor. In addition to this, the charge controller will be used to test the battery levels and determine the maximum distance the vehicle can reach on the current charge. If a certain charge level is reached, the HUD will automatically put the motor controller into power saving mode. Incorporating GPS into the HUD, we will be able to plot a destination and an algorithm will first determine if the destination is obtainable, then plot the best route using the most energy efficient drive mode. The GPS and vehicle performance will be displayed with a touch screen for the user. Figure 3 gives an illustration of this system.
1.1.3.4 **Project Budget and Finance**

Our current budget is dependent upon having a golf cart reused from a previous semester. These are rough estimates based on internet searches for a general price point. These are subject to change with the needs of our project. We will most likely keep the solar panels and batteries. Table 2 lists the estimated costs of each portion for this project.

<table>
<thead>
<tr>
<th>Part</th>
<th>Estimated Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motor Controller</td>
<td>$600</td>
</tr>
<tr>
<td>Batteries</td>
<td>$600</td>
</tr>
<tr>
<td>Charge Controller</td>
<td>$150</td>
</tr>
<tr>
<td>Microcontroller(s)</td>
<td>$120</td>
</tr>
<tr>
<td>Touch-Screen Displays</td>
<td>$120</td>
</tr>
<tr>
<td>Sensors</td>
<td>$150</td>
</tr>
<tr>
<td>Misc.</td>
<td>$100</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$1,840</strong></td>
</tr>
</tbody>
</table>

Table 2 Project Budget and Finances

1.2 **Project Significance and Motivation**

This project will be a great chance for a group of students to learn about how to make use of solar panels in converting a standard golf cart to run off of the sun entirely. If people see that this can be done with a golf cart and some solar panels they will wonder if the same could be done for their car or home. It could help to increase solar powered awareness at the same time as potentially creating four new engineers that can go into the renewable energy industry.
2 Research

2.1 Heads-Up Display (HUD)

The solar powered golf cart will include a touchscreen display which is to be powered by
the cart’s onboard power system. This display will receive input from a BeagleBone Black
microcontroller that is running the Android Mobile Operating System that has a navigation
application pre-loaded. This Android Application will contain a modified google maps API
to work specifically on the UCF campus as well as features to report cart status to the
user. By using this application via the touchscreen display any user should be able to
simultaneously navigate the UCF campus area and check the cart’s current status with
ease.

2.1.1 Navigation

The first selection option in the display application’s launch menu is the navigate button.
When the user presses this button the application will switch to a new activity screen
where a map will be generated using the Google Maps API. From this screen the user
will be able to track their position, set destination waypoints, navigate to waypoints, and
return to the application’s main menu. Figure 4 shows what the user sees upon clicking
the "Navigate" button, notice the map plugin and associated features.

![Figure 4 Navigation Map](image-url)
The application will supply the user with feedback regarding position throughout standard operations and locomotion of the golf cart. This operation is accomplished through interfacing of the Adafruit Ultimate GPS Breakout microcontroller position data with the Beaglebone Black microcontroller’s Android Application. Upon receiving this data the Android Application will calculate user position by passing the user's current GPS position to the Google Maps API function. All API functions were acquired from google with permission and all possess the required credentials to use them in the Android Application. See appendix A or Figure 5 for detailed license key information.

2.1.1.1 Google Maps
The navigational map is implemented via the Google Maps API which features many interactive user navigation elements while displaying a very accurate map in the area around the current users. The license to use the Google Maps API was acquired through code.google.com's API console and is shown in Figure 5. This license gives permission to use the Google Maps API throughout the entire Android Application project.

```xml
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="A1zaSyBjMgtsy16j8beyRVx0agFdzEkma4-LuM" />
```

Figure 5 Google API License

The interactive map feature is generated as a Fragment in the Navigation Page's Activity File. This is accomplished through declaring it within the .xml documentation in accordance with the proper xml syntax and then followed by making declarations to control the start location and zoom of the map fragment. The map is also modified from within the Navigation Page’s java file in order to implement the Current Location Cursor and other various map features.
2.1.1.2 GPS
The raw GPS data of the user will be supplied to the BeagleBone Black for computation through the Adafruit Ultimate GPS Breakout module. The Adafruit Ultimate GPS Breakout chip is designed and manufactured by Adafruit Industries in order to provide an affordable GPS solution to almost any project. This board is created using the MTK3339 chipset which allows it to track up to 22 satellites on 66 channels with a refresh rate of 10Hz, which will allow for the user’s position to be updated 10 times a second for ensured accuracy to within 3m while the cart is in operation. The module contains a built in ceramic patch antenna built onto it that gives it a tracking sensitivity of -165 dBm and has the necessary functionality to have any 3V GPS antenna attached to by the uFL connector in order to achieve better tracking sensitivities ratings. The board will be connected to the BeagleBone Black via PCB and will output to the BeagleBone Black using the NMEA 0183 electrical communication standard with a default baud rate of 9600. The board will draw 5V and 20mA of current from the cart’s power system and has a built-in data logging system which uses the module’s internal flash memory to provide reliable data recalls up to 16 hours after a power loss occurs.

2.1.2 Golf Cart Status
The second selection at the application’s main menu is the "Cart Status" button. After clicking this button the user is brought to a new screen with several displays that contain the current cart conditions as reported to the BeagleBone Black from the power management microcontrollers as well as buttons that will allow users to switch between the cart’s different modes of use. The four cart status metrics reported to the user are as follows: range left, battery percentage, the current speed, and the current operating mode. Figure 6 details the appearance of the Cart Status screen.

Figure 6 Golf Cart Status Screen
2.1.2.1 Range Left
This first status which is displayed to the user when they arrive at the Cart Status screen is the range left to travel based on the current battery capacity. The range distance quantity will be calculated within a method in the Android Application loaded onto the BeagleBone Black board. This method will receive its input values from the Charge Controller board indicating the current battery charge and the current operating mode for use in determining the distance the cart can travel. Then using these two given values the application computes the remaining distance the cart will travel and displays it accordingly.

2.1.2.2 Battery Percent
The next status that users will observe in the Cart Status screen is the remaining battery percentage in the cart's battery bank. This percentage is generated within the Android Application using inputs supplied by the Motor Controller module to the BeagleBone Black chipset’s GPIO pins that will provide necessary data for the Android Application to generate the remaining battery capacity. Using this supplied data the Android Application will generate a percentage for the user where 0% is no battery capacity and 100% is maximum battery capacity. Once the remaining battery percentage is calculated it is displayed on the Cart Status screen and also used in calculating the Range Left value.

2.1.2.3 Speed Display
The first entry on the second row of status fields is the current cart speed. This value will be displayed in units of miles per hour and will be calculated by the Android Application using data supplied by the GPS coordinate tracking instrument to calculate an estimate of the cart's current speed. The data supplied by the GPS coordinate tracking instrument is stored by the Android Application to compare the current GPS coordinate position to the saved values of the previous GPS coordinate positions to get an approximate velocity of the cart by comparing coordinate distance value differences. The calculations involved in computing the cart velocity will provide a result that can be used by the "Cart Status" screen to provide a displayed value of the current speed in units of miles per hour. This value will most likely need to be updated every two seconds or less in order to create an up-to-date value for users.
2.1.3 Mode Selection

While at the golf cart status screen the user will also be able to control the cart's current operating mode by interfacing with three buttons at the bottom of the screen shown in Figure 7. Through interfacing with these buttons the user will be able to quickly and easily change the current power mode directly from the status screen. Upon pressing any of these buttons the Android Application will send a signal to the motor controller that will then adjust the engine throttle respectively. The three modes of operating power consumption are as follows: High Performance, Standard, and Energy Saving.

![Figure 7 Modes of Operation Buttons]

2.1.3.1 High Performance

Upon user selection of the High Performance mode of operation the Android Application will send a signal to the motor controller to release any previous throttles on the golf cart's engine which will allow for top speeds. The application will also change the brightness setting on the Touch Screen Display to the maximum setting. This mode of operation is not intended for sustainable use as it will drain the cart's battery bank quickly and leave the user unable to perform any other functions until charge has been restored. The user can switch to another operational setting by clicking on either of the other two buttons located on the Cart Status screen.

2.1.3.2 Standard

When the Standard mode of operation is selected at the Golf Cart Status screen the Android Application will inform the motor controller to set the throttle to a maximum speed of approximately 10 mph, which is roughly 67% of maximum engine output. The application will then also change the brightness setting on the Touch Screen Display to 60% of maximum brightness. This mode is best suited to short-medium range travel distances and is best implemented on a route with plenty of direct sunlight. The user may switch out of the Standard mode of operation by clicking on either of the other two buttons located on the Cart Status screen.
2.1.3.3 Energy Saving
The most efficient mode of operation for extended periods of use is the energy saving mode. In this mode the Android Application will send a signal for the motor controller to set the throttle to a maximum speed of about 7 mph, which is 47% of the maximum possible engine output. In this mode the Touch Screen Display’s brightness is set to 40% of the maximum display brightness. This mode should be used when the user wants the highest range possible. This mode will be automatically activated should the battery charge on the cart drop below 25%. If there is more than 25% battery charge the user can change modes from the Energy Saving mode of operation by clicking on either of the other two buttons located on the Cart Status screen.

2.1.4 Touch Screen
The pivotal piece of equipment in the Heads-Up Display system is the touchscreen display, which will output the video signal of the BeagleBone Black for the user’s display and interaction with the Android Application. The display absolutely must be touchscreen in order to facilitate proper interface with the Android Application and should also be very power conservative so that the cart can maintain battery charge for other cart operations. The BeagleBone Black only outputs video signal via HDMI so any touch screen display used must have an HDMI input. As a result of our power and space confinements the ideal display chosen would be within the range of 5" to 10" in size.

2.1.4.1 Options
While deliberating which touchscreen display would be the best fit for the Heads-Up Display system the following requirements were taken into consideration: functionality, resolution, power cost, and monetary cost. In searching the World Wide Web there were three displays which met our requirements. These three displays were the HDMI 4 Pi: 5" Display w/Touch and Mini Driver – 800x480 HDMI, the 664/O/P Lilliput 7” LCD monitor, and the Tekit 619AHT 7” LCD Monitor. All of the listed displays have HDMI video signal input and are equipped with resistive touchscreens for seamless user interfacing with the Android Application. After contemplating the available options for a touchscreen display, it was decided that the HDMI 4 Pi: 5" Display w/Touch and Mini Driver could effectively meet all generated requirements and integrate easily into the Heads-Up Display system. The details for each of the different displays and the comparisons between technical specifications are located in Table 3.
## 2.1.4.2 Configuration

The HDMI 4 Pi: 5” Display w/ Touch and Mini Driver’s display will be mounted on the golf cart’s dash while the driver board and touchscreen adapter board will be concealed within the golf cart’s chassis if possible. The display is powered by a 5V 1A USB power supply on the driver board which may be stripped in order to be connected directly to the cart’s on-board power system. It will receive its display video signal input directly from an HDMI connection that will be connected as an output from the BeagleBone Black. The resistive touchscreen system is managed through a separate pre-built PCB that sends output from the screen through a MiniB USB port to the BeagleBone Black in order to process the user’s input. There is also a separate board included for controlling the brightness, color, and contrast via buttons however, it will not be used as those functions will be implemented within the Android Application on the BeagleBone Black.

## 2.1.5 Microcontroller

The microcontroller chosen to output to the display is the BeagleBone Black open-source microcontroller. It will output video signal to the Heads-Up Display’s touchscreen monitor and also be able to send signals to the other cart control boards with essential operating functions through connections made with the 92 pins on the BeagleBone Black’s two 46-count pin arrays. The Android 4.4 KitKat mobile environment along with the accompanying Android Application will be imaged onto a microSD memory card and inserted into the BeagleBone Black’s microSD memory card slot which will provide the entire module with various resources to send signals to other components in the cart’s system and run an array of transmission and value calculation functions using an Android Application created with specifications to accomplish the planned software functionality. The BeagleBone Black is an open-source development board using Texas Instrument’s OMAP3530 system-on-a-chip design and is manufactured by Circuitco LLC on behalf of BeagleBoard.org.
### 2.1.5.1 Options

When considering which microcontroller to use for the Heads-Up Display system there were several factors to take into account, which included the following: functionality, performance, power cost, and monetary cost. Our preliminary list of possible boards that could possibly meet these requirements included the Raspberry Pi, BeagleBone Black, and Arduino Uno. Each one of these boards are open-source and has USB interfaces as well as various forms of output in order to communicate with the Heads-Up Display along with other boards on the cart. After much deliberation it was deemed that the BeagleBone Black met all of the compiled requirements and would be the easiest choice to integrate into our design. The details for each board and comparisons between the three of them are located in Table 4.

<table>
<thead>
<tr>
<th></th>
<th>Raspberry Pi</th>
<th>BeagleBone Black</th>
<th>Arduino Uno</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Market Price</strong></td>
<td>$35</td>
<td>$45</td>
<td>$25</td>
</tr>
<tr>
<td><strong>SoC</strong></td>
<td>Broadcom BCM2835</td>
<td>AM3359</td>
<td>ATmega328</td>
</tr>
<tr>
<td><strong>CPU</strong></td>
<td>700 MHz ARM1176JZF-S</td>
<td>1000 MHz Cortex-A8 + 2xPRU(200 MHz)</td>
<td>16 MHz ATmega328</td>
</tr>
<tr>
<td><strong>GPU</strong></td>
<td>Broadcom VideoCore IV @ 250 MHz</td>
<td>PowerVR SGX530 (200 MHz)</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>512 MB</td>
<td>512 MB</td>
<td>2 KB</td>
</tr>
<tr>
<td><strong>USB ports</strong></td>
<td>4</td>
<td>1 Standard A host port. 1 mini B device port</td>
<td>1</td>
</tr>
<tr>
<td><strong>Video outputs</strong></td>
<td>HDMI and composite video</td>
<td>Micro-HDMI</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Audio outputs</strong></td>
<td>Analog and HDMI</td>
<td>Micro-HDMI</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Onboard storage</strong></td>
<td>MicroSD</td>
<td>8-bit eMMC with 4GB Debian. MicroSD</td>
<td>32 KB</td>
</tr>
<tr>
<td><strong>Onboard network</strong></td>
<td>10/100 Mbit/s Ethernet USB adapter</td>
<td>Fast Ethernet</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Low-level peripherals</strong></td>
<td>17x GPIO plus the same specific functions and HAT ID bus</td>
<td>4x UART, 8x PWM, LCD, GPMC, MMC1, 2x SPI, 2x I2C, A/D Converter, 2x CAN bus, 4 Timers</td>
<td>6x PWM, 8x Digital I/O, 6x Analog Input</td>
</tr>
<tr>
<td><strong>Power ratings</strong></td>
<td>600 mA (3.0 W)</td>
<td>210-460 mA (2.0 W)</td>
<td>50 mA (0.25 W)</td>
</tr>
<tr>
<td><strong>Power source</strong></td>
<td>5V via MicroUSB or GPIO header</td>
<td>Mini USB, 2.1mm x 5.5 mm 5V jack</td>
<td>2.1mm x 5.5 mm 5V jack</td>
</tr>
<tr>
<td><strong>Size</strong></td>
<td>85.60 mm x 56.6 mm</td>
<td>86.40 mm x 53.3 mm</td>
<td>68.58 mm x 53.3 mm</td>
</tr>
<tr>
<td><strong>Weight</strong></td>
<td>45g</td>
<td>39.68g</td>
<td>28g</td>
</tr>
</tbody>
</table>

Table 4 Proposed HUD MCU Specifications
2.1.5.2 Performance

The biggest factors in choosing an appropriate microcontroller for the Heads-Up Display were the board’s functionality and its data processing performance. The required functionality of the board chosen would need to include a video output to the Heads-Up Display monitor and the ability to run navigational and signal output software. Through consideration of these requirements the group factored out the Arduino Uno from their choices since it would be unable to output video signal to the Heads-Up Display and also does not have the necessary resources to run navigational software with a GUI. The next topic of consideration in our choice of microcontroller is the data processing performance of the board. The microcontroller would need to possess the computing power to run software that would be able to provide an interactive map and navigation options to the user as well as the ability to move to another menu in order to change the cart’s mode of operation and see various metrics concerning cart operations. With these requirements in mind the design team came to the conclusion that an Android Application should be used to easily create an application that will be able to meet the functional requirements and multi-task all application functions as well. While the Raspberry Pi and BeagleBone Black can both run Linux distributions, the BeagleBone Black is the only one of the two that can provide a stable runtime environment for an Android image.

2.1.5.3 Cost

The next factors that were taken into consideration while deciding which one of the microcontrollers to use were the power cost and monetary cost of the chosen board. The Arduino Uno would draw only 0.25 W of power, while the BeagleBone Black and Raspberry Pi would draw 2W and 3W of power, respectively. Since the Arduino Uno did not pass the functionality requirements for the project the next best choice was the BeagleBone Black at 2W of power.

The monetary costs for all of the boards in consideration are as follows: Raspberry Pi - $35, BeagleBone Black - $45, and Arduino Uno - $25. Even though the BeagleBone Black has the highest monetary cost of all the boards, it also surpasses the other boards in performance and consumes less power than the Raspberry Pi. Therefore the choice was made clear to the design team that the BeagleBone Black would be the board most suited to the highest priority requirements of the project.
2.1.6 Android Application

In order for the Heads-Up Display to perform the functions listed in above sections the BeagleBone Black module will boot off the microSD memory inserted that will have an image of the Android OS installed on it. When the user powers on the cart the Android OS environment will boot and the Android Application will be executed immediately so that the user will be brought to the main screen, as seen in Figure 8. This application will be developed entirely by the project team in the Java Object-Oriented language within the Eclipse IDE and compiled for the Android V4.4 API. This application will utilize the Google Maps API support library’s functions with proper permission from Google Inc. in order to facilitate drawing the GPS positioning navigational map and enable all associated icon generation on the map interface, all license and permission documentation for the Android Application project is located in this document's Appendix A.

![Android Application Main Screen](image)

Figure 8 Android Application Main Screen
2.1.6.1 Development

The development of the Android Application was facilitated through the use of Google's ADT bundle which can be found at https://developer.android.com/sdk/index.html?hl=i. This bundle included the Eclipse IDE for Java design platform and the Android SDK manager application executable files along with associated file directories. Together these two applications provided the necessary resources to create Android Application source code, use third-party objects with associated functions, and provided the developers with a means to port the developed app to a prototype device running the Android OS. A helpful set of guidelines and function listings for the Android Application developers is provided for free courtesy of Google Inc. This tutorial for Android Application design can be found at the following url: http://developer.android.com/training/index.html. Another helpful online tutorial that was used in development for reference was the Google Maps JavaScript API tutorial provided for free by Google Inc. which provided the developers with insight into the many key features of the generated navigational map along with how to use each function in the most practical manner possible. This multi-step tutorial can be found online. There were multiple SDK packages used in the application's development to implement various pre-developed functions throughout the android project, these packages are displayed in Figure 9.

Figure 9 List of Installed SDK Packages
2.1.6.2 Integration
The developed Android Application will be integrated into the working golf cart system by being loaded onto the microSD memory card that will be inserted into the BeagleBone Black microcontroller’s respective slot. The BeagleBone Black module will have inputs and outputs connected through the P8 and P9 pin arrays located on the module that will be associated with the inputs and outputs of the other different modules throughout the golf cart's electronic control system. Once the BeagleBone Black module is properly connected to the designed PCB with the MTK MT3339 GPS module and is able to communicate with the GPS module along with the other on-board systems it will then be able to send and receive data concerning data values displayed in the “Cart Status” screen’s fields. This electrical signal data from the other module's in the cart’s design will be sent and received directly from the Android Application via the BeagleBone Black’s connected pins.

2.1.6.3 Runtime Environment
The Android Application will run on the Android 4.4 KitKat mobile environment that is imaged onto microSD memory card located in the BeagleBone Black board’s microSD card slot. The video signal output of the application will be wired to the touchscreen display through an HDMI video signal connection and will provide user's with an interaction similar to any other touchscreen device running the Android OS. The user’s touchscreen input gestures will be sent to the Android OS environment on the BeagleBone Black through a USB connection between the two modules. This transmitted gesture data sent through the USB connection will be directed to Android Application on the Android OS environment for processing of the data, which will then perform the matching action associated with the user-input gesture. Both BeagleBone Black board and the touchscreen display will receive 5V of power in the correct power connections which will be provided to both modules through the cart's on-board power system.
The minimum Android SDK version that the application can be run on is version 8.0 and the target Android SDK version for this application is version 21.0. Since this application uses navigation and location information through user GPS position as well as requiring information to be written to the device’s external storage, there must be permission declarations to allow such functions within the application. The first permission given to the Android Application is android.permissionINTERNET which allows the app to fetch data from the internet using the current design prototype’s internet connection. The next permission declared in the AndroidManifest.xml file is android.permission.ACCESS_NETWORK_STATE which allows the Android Application to check the status of the prototype operating platform’s network connection. The third permission declaration in the AndroidManifest.xml file is android.permission.WRITE_EXTERNAL_STORAGE which gives permission to the Android Application to write to the prototype device’s external storage unit as it sees necessary throughout operation. The last permission that is declared within the Android Application’s AndroidManifest.xml file is android.permission.ACCESS_FINE_LOCATION that will allow the Android Application to request access to the current prototype’s GPS coordinate tracking instrument in order to gather coordinate information on the user’s current GPS position. The Android SDK version and application permissions are located within the AndroidManifest.xml file of the application project library are shown in Figure 10, which is a screenshot of part of the xml file.

```xml
<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="21" />

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Figure 10 Android Application SDK Version and Application Permissions
2.2 Motor Controller
The solar powered golf cart will feature a custom designed motor controller in order to allow for the different modes of operation and cruise control. Having this custom motor controller will allow for future expansion of the golf cart voltage if the owner feels the need to do so. It requires a method of reading in the pedal position, control loops to change the rate of acceleration, and a way to control the average power being delivered to the motor.

2.2.1 Throttle Sensor
The two types of throttles are contact and non-contact. Contact pedals are a variable resistor that relies on the pedal physically moving a potentiometer to output a corresponding value to position. Non-contact pedals are more common now and they include inductive and hall-effect. These are more reliable because it does not wear down over time from use.

2.2.1.1 Inductive Throttle
An inductive throttle uses a loop of current and a piece of metal with varying inductance. As the pedal is pressed down, the inductance changes and a varying level of voltage is output from the current sensor. Because no contact needs to be made from the current loop and the metal being pressed through it, these throttles are very reliable. This is the type of pedal that is currently implemented on the golf cart. From my research so far, 14V is input to the pedal and a varying level of voltage from 0.5V to 1.5V is output depending on the position of the pedal. Figure 11 shows the physical design of the ITS in the EZ-GO golf cart.

Figure 11 EZ-GO Inductive Throttle Diagram (Reprinted with permission from EZ-GO)
2.2.1.2 Potentiometer Throttle

The easiest type of throttle to implement would be a potentiometer. It has three terminals: ground, variable resistance wiper, and supply voltage. As the pedal is pressed into the floor, the wiper slides down the potentiometer and outputs a higher voltage. This variable voltage divider means that the golf cart pedal can send a voltage level to an ADC with respect to the angle of depression. This signal is then read as the desired output which then has to go through software and PID control loops. One type of potentiometer pedal considered was the Ford Five Hundred accelerator pedal APS118. The disadvantage to this type of pedal is that it wears down quickly and outputs a voltage level higher than most microcontrollers can handle. Figure 12 shows an image of this pedal.

![Figure 12 Ford Five Hundred Pedal](Reprinted with Permission from AutoSales inc. Terms of Use)

2.2.1.3 Hall Effect Sensor

A Hall Effect sensor is a transducer that varies its output voltage in response to a magnetic field. These types of throttles are about the same ease of use as a potentiometer with the reliability of an inductive throttle. This makes them the best option in theory, but they are much more expensive to implement than a potentiometer throttle and are difficult to install. Most Hall Effect sensors can range from $300 up to $700 for a nice one. They also tend to not come with pedals which means that more mechanical design will need to be done which is not the point of this project. Figure 13 shows a Curtis PB-6 that was considered, but is impractical because of the cost and requirement to design a mechanical system for it to work with the golf cart.

![Figure 13 Hall Effect Throttle Box](Reprinted with Permission from EVWEST Copyright Conditions of Use)
2.2.2 Software Control
The motor controller software must be able to monitor various statuses at the same time and be able to easily output PWM signals. Both current and voltage sensors will need to be used. The MCU that may be used in this implementation could run on 3.3V, so it may be necessary to use op-amps to multiply and divide both inputs and outputs to not overload the ADC. This will give a wider range of hardware options that are not compatible with 3.3V logic outputs. It might also allow me to keep the same pedal that is currently inside the golf cart.

2.2.2.1 Modes of Operation
The mode of operation will be selected by the touch screen available to the user at the front of the golf cart. This mode is then used by the motor controller software to determine the pulse width of the PWM signal. An illustration of this is given in Figure 14. Note that the illustration does not accurately reflect this specific design's power modes, but is merely an illustration of its effect. The width of each pulse controls the average power delivered to each MOSFET and thus controls the speed of the golf cart.

![PWM Average Power Control](image)

Figure 14 PWM Average Power Control
High power mode will allow the motor to receive 100% of the available current from the motor controller and electronics will have maximum brightness. Standard mode will have moderate brightness and allow the motor to run at a max power of 80%. At a charge level of 25%, the golf cart automatically dips into power saving mode which reduces brightness to a minimum and only allows the golf cart to reach 60% of total power.

### 2.2.2.2 PID Control Loops

A proportional-integral-derivative controller (PID controller) is a control loop feedback tool that is used in controlling the rate at which values reach a desired set point. In this case, speed is the desired value to be controlled. If the golf cart accelerates too fast or too slow it will feel weird. A PID controller calculates an error value as the difference between a measured process variable (PWM width) and a desired set point (pedal position). The controller attempts to minimize the error by adjusting the process through use of a three variables: the proportional, the integral and derivative values, denoted Kp, Ki, and Kd. Kp depends on the present error, Ki on the accumulation of past errors, and Kd is a prediction of future errors based on current rate of change. For the motor controller, the throttle is going to be the desired set point of the algorithm and the current being passed through the motor is going to be the current value. The difference between these two will be the error that calculation is done on. This idea is demonstrated per Figure 15.

![PID Control](image)

**Figure 15 PID Control**

### 2.2.2.3 Communication with HUD

A few methods of communication considered between the motor controller and the HUD includes SPI, I2C, UART, and direct GPIO connection. The design considerations that need to be accounted for is speed of the connection, number of lines, and robustness of the signal. Robustness is the main priority for this project because there will not be much data passing through to the HUD other than what mode to operate in. This means that unless some kind of programmability through the HUD is added later, there is no need for high speed communication lines as long as the connection is consistent and reliable.
2.2.2.3.1 SPI
Serial Peripheral Interface (SPI) is an interface bus commonly used to send data between microcontrollers and small peripherals such as off-chip memory and sensors. It has a clock pin, data out, data in, and chip select wire. This kind of communication would allow for the motor controller and HUD to be talking at the same time. For a high speed communication line this would be great, but because the two MCU’s will most likely only be transmitting what mode it is in, this may not be the best option.

2.2.2.3.2 I2C
Inter-Integrated Circuit (I2C) is a multi-master, multi-slave communication line that is used to communicate at low speeds with two wires. A clock line and Data line can be controlled by any of the devices along the chain, but only one can be talking at a time. This type of communication would allow for fewer wires between the motor controller and HUD, but would limit them from talking back and forth.

2.2.2.3.3 UART
Universal Asynchronous Receiver/Transmitter (UART) uses two wires, transmit and receive, to communicate between two devices. The line is held high until there is data going across in one direction and at that point, the other device would read in each bit in serial. This type of communication line would also require two wires, but would be much easier to implement than the I2C line. Data could also go in both directions at the same time.

2.2.2.3.4 Direct GPIO
Direct GPIO wire to wire connection would be a very simple solution to communicate what power setting the golf cart is in. An easy digital read function would tell the motor controller what mode it should be in.

2.2.2.4 Speed Sensors
Newer EZGO golf carts have a built-in speed sensor that the model we are using does not have. Therefore it is not feasible to have the motor controller calculate speed from GPS. The speed will most likely be determined from the HUD microcontroller and then transmitted to the motor controller.
2.2.3 Circuit Design
Initial research into open source motor controllers showed that the main concerns for circuit design include the logic portion, direction control, reliable shutdown, and heat distribution. The logic contains a microcontroller that senses current and voltage and outputs a corresponding PWM signal, reliable shutdown involves hard overcurrent shutdown that will remove power in the event that too much current is slowing, and heat distribution of the MOSFETs so that they do not just overheat and kill themselves. If chosen carefully, the motor controller should work flawlessly and be easier to maintain than an OEM motor controller.

2.2.3.1 Microcontrollers
MCU decision has to be based considering a few different performance criteria: Number of ADC channels, resolution of the ADC, dedicated PWM channels, enough processing power to control motors in near real-time, power consumption, and number of GPIO. Below are a few different options from TI that were considered. TI was the main choice for this project because of their ease of use with programming and affordability of full board solutions. All three are relatively inexpensive when compared to the entire scope of this project. Table 5 lists a full comparison of each board's specifications.

C2000: This microcontroller specializes in PWM and motor control features. It has built-in hardware to make it easy to control a 3-phase AC motor and high accuracy ADC/PWM pins. It has built-in I2C, SPI, and UART pins for easy communication line setup. Example code is also provided to get users going on projects much quicker than reading a datasheet. The example code and libraries is actually a standard for all of the main choices for microcontroller. This MCU is not optimized to save power however, and is designed purely for accurate motor control.

MSP430: There are many similarities between each MCU in terms of communication capabilities, but the MSP430 excels in power consumption. Code composer automatically optimizes power consumption for this board when compiling the software. The downside to this microcontroller is that the ADC channels are only 10-bit instead of 12 like the C2000 and TivaC. Lower ADC resolution combined with no dedicated PWM channels means that this is the least desirable of all three microcontrollers considered. PWM channels would have to be implemented using counters which would be very difficult and inefficient.
TivaC 129XL: This MCU is a combination of the two previously considered microcontrollers. There is a dedicated hibernation module, high accuracy PWM channel, and high resolution ADC. With this hibernation mode, this MCU will not consume power while the golf cart is off like the C2000 would, but manages to keep the positives of the motor controller features. If I were using an AC motor, I wouldn’t have any choice other than to use the C2000 but since it is a DC motor I have the option of using this versatile MCU.

<table>
<thead>
<tr>
<th>Microcontroller</th>
<th>C2000</th>
<th>MSP430</th>
<th>TivaC 129XL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price (USD)</td>
<td>17</td>
<td>9</td>
<td>20</td>
</tr>
<tr>
<td>Processor Speed (MHz)</td>
<td>60</td>
<td>16</td>
<td>120</td>
</tr>
<tr>
<td>ADC Resolution (bits)</td>
<td>12</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>PWM Channels</td>
<td>8</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>QEI Feedback</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Dedicated SPI/I2C Pins</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Flash Available</td>
<td>64KB</td>
<td>16KB</td>
<td>1MB</td>
</tr>
<tr>
<td>Low Power Mode</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>PWM Hardware Interrupt</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 5 Microcontroller Comparison

The final decision was to go with the TIVAC 129 XL because it offers much more features and controls than the other boards for about the same cost. Despite using more power than the MSP430, it does have a low power state that it can be put into when the golf cart is turned off. The C2000 was a very tempting board because it specializes in motor control, but those features are meant specifically for industrial applications that involve stepper motors and AC motors which will not be used in this project. The specialized communication lines will also make it easier to transmit data back and forth between the HUD MCU and the motor controller. The processing speed of the TivaC is also much quicker than the other two which would make the PID control perform at a more real-time rate. If the rate of control is slow, then the driver might feel some delay when they press down the pedal. It would also cause an odd slowing down scenario where the golf cart speed doesn’t drop immediately. All three microcontrollers had plenty of flash space on them to save the program so that was not really an issue to consider. There are currently no plans to upload multiple large programs or look up tables that would consume large amounts of space.
2.2.3.2 Programming
Software will be a key component in this motor controller for safety precautions and PID control to smooth out the motor acceleration. Without reliable software, the golf cart will accelerate jumpy and may even get stuck in an accelerating mode. The software environment that will be used depends largely on what microcontroller is chosen. For TI microcontrollers, the two software options are Code Composer and Energia. Atmel microcontrollers have the infamous Arduino software environment which is actually very similar to Energia. Many of these different vendors actually run off of the same architecture so some of them are capable of interchanging environments as long as it is setup correctly with the corresponding drivers and include files.

2.2.3.2.1 Code Composer Studio
Code Composer Studio (CCS) is a very powerful programming tool provided by Texas Instruments (TI) that gives direct access to much of the microcontroller, but sacrifices ease of use to do so. Code is written in C or C++ which makes it easy for embedded programmers to pick up with a language that is familiar to them. It has utilities such as memory viewer and API’s for many different Launchpad developer boards. An example screenshot of how difficult it can be to write code in code composer is shown in figure 16. After seeing how much time and lines of code it took just to get an LED blinking, Energia definitely seems like the way to go.

```c
#include <stdio.h>
#include "tcd1254pocpt.h"

int main(void)
{
    volatile uint32_t u32Loop;
    Systick HickPrio = SYSTICK HickPrio = 112;
    u32Loop = SYSTICK HickPrio = 8;
    GPIO_PORTD_DIR = 0x81;
    GPIO_PORTD_0 = 0x81;
    // Loop forever.
    while(1)
    {
        // Turn on the LED.
        GPIO_PORTD_DATA &= ~0x81;
        // Delay for a bit.
        for(u32Loop = 0; u32Loop < 200000; u32Loop++)
        {
        }
        // Turn off the LED.
        GPIO_PORTD_DATA |= 0x81;
        // Delay for a bit.
        for(u32Loop = 0; u32Loop < 200000; u32Loop++)
        {
        }
    }
}
```

Figure 16 Code Composer LED Example


## 2.2.3.2 Energia

Energia is also provided by TI, but is different in many ways. This applications sacrifices in-depth tools for ease of use. They have predefined functions for PWM output, ADC reads, and much easier syntax compared to C or C++. Due to the simple nature of the programming involved with this project, this is the preferred option. Control over ADC, PWM, Sleep Mode, and GPIO are the most important functions to this project and Energia is much easier to use than code composer for each one. Energia programs have two main portions: Setup and Loop. Setup is done in the beginning where you define your pins and their purpose: input/output and analog/digital. Loop is where the actual code is executed endlessly until the user decides to cancel it. This negates the possibility of using a watch dog timer to avoid crashes which is fine because in a worst case scenario for the driver, the key will be able to kill power and the brakes are mechanical so this is not a safety concern.

## 2.2.3.3 IGBT

An alternative to Power MOSFET’s is insulated-gate bipolar transistor (IGBT). This power semiconductor has the turn-on properties of a MOSFET, but the current carrying properties of a bipolar junction transistor (BJT). They are very easy to use devices because they come in modular packages which means they would not have to be cascaded like MOSFETs. They are also much more robust for high voltage and current applications. Despite the ease of use for these devices, the voltage drop across the collector to emitter can be relatively high. Figure 17 shows the typical voltage drop characteristics of an IGBT. As the current goes above 100 amps, over 4V is being dissipated in the circuit. This is over 400 watts which would create an immense amount of heat. IGBT’s are only practical in much higher voltage applications where the duty cycle is extremely low.

![Figure 17 Typical IGBT Performance](Reprinted with Permission from GNU Free Documentation License)
2.2.3.4 Power MOSFET’s

Power MOSFET’s are the best way to control current in a fast switching speed environment. The gate can be pulsed with a PWM signal to control the average current being passed through to the motor. These can be used to control the movement of DC motors or brushless stepper motors directly from a microcontroller or by using PWM type controllers. As a DC motor offers high starting torque and which is also proportional to the armature current, MOSFET switches along with a PWM can be used as a very good speed controller that would provide smooth and quiet motor operation. A very simple MOSFET motor control circuit is shown in Figure 18. The gate of the MOSFET is where the PWM channel would be applied. Note the free-wheeling diode in this circuit which is discussed in section 2.2.3.7.

![MOSFET Motor Control Example](image)

2.2.3.5 Thermal Runaway

IGBT’s do not work in parallel configurations due to their positive thermal coefficient. As they begin to heat up, more current is allowed to pass through which in turn continues to increase the temperature and the cycle ensues. This means that when you try to put two IGBT’s in parallel to increase current capacity, if they are not balanced well thermally then one of the IGBT’s will end up doing all of the work and destroying itself from heat. However, in MOSFETs the exact opposite occurs. As they heat up their current capacity decreases and they automatically balance the current among all parallel MOSFETs. This is another disadvantage to IGBT’s in comparison to MOSFETs.
2.2.3.6 Pre-Charge Circuit

The PWM motor controller that I plan on using in this golf cart needs to have a sizable bank of capacitors on its input to smooth out voltage spikes or dropouts that may occur in unreliable sources such as batteries. If you apply voltage to a bank of capacitors, it is initially seen as a short circuit. If there is very little resistance in the circuit, like a closing contactor with no pre-charge, then the current will be very high. Nearly all of the battery pack voltage will be across the closing contacts. The large voltage difference and sudden high current (known as an inrush current) can cause damage to, and in extreme cases, welding of the relay contacts. This can all be prevented by the use of a pre-charge resistor across the contacts of the main power relay. The pre-charge resistor allows the capacitors in the controller to slowly charge before the contactor closes. This means that there is less voltage across the closing contacts and little or no inrush current. In order to do this, the contactor is held open for a certain amount of time to allow for charging to complete and then finally closing the switch. Fortunately, the golf cart has a solenoid circuit built in that performs this contactor function. The microswitch that activates when the pedal is pressed down is how this contact is made. It is held in low current mode pre-charge state until the micro-switch sends the battery pack voltage to connect the solenoid to the short circuit. A picture for easier conceptualization is shown in Figure 19.

![Pre-Charge Circuit Diagram]

Figure 19 Pre-Charge Circuit
2.2.3.7 Free-Wheeling Diode

A free-wheeling diode is used in motor control circuits to avoid a large backrush current when power is removed the motor. When power is removed from the inductive load, there is a charge stored there that built up while it was spinning. This charge has nowhere to go and can cause a huge spark if it is reconnected to the battery pack voltage. On the other hand, this voltage can be used to recharge the batteries as they discharge. This is where the idea of regenerative braking comes from. When that large charge exceeds the amount needed to turn on the diodes, it will dissipate that energy back into the positive terminal of the batteries to charge them. This is demonstrated in Figure 20. Note that a “flyback diode” is the same as a Free-Wheeling diode. In an IGBT configuration, two IGBT’s are required to create a motor controller and one of them actually performs the function of a free-wheeling diode.

Figure 20 Free-Wheeling Diode Graphic
(Reprinted with permission of GNU Free Documentation License)
2.2.4 Direction Control
The golf cart needs to be capable of moving backwards and forwards thus the need for control over direction.

2.2.4.1 H-Bridge
An H bridge is an electronic circuit that enables a voltage to be applied across a load in either direction. These circuits are often used in DC motors to control the direction of motion. A functional model is shown in Figure 21.

![H-Bridge Functional Model](image)

Figure 21 H-Bridge Functional Model
(reprinted with permission of GNU Free Documentation License)

2.2.4.1.1 Relay Design
One way to build an H bridge is use an array of relays from a relay board. A relay can generally achieve the same electrical functionality as an H bridge. However, a semiconductor-based H bridge would be preferable to the relay where a smaller physical size, high speed switching, or low driving voltage is needed, or where the wearing out of mechanical parts is undesirable. Due to the mechanical nature of a relay, this is not the preferred method because it is hard to control the amount of current passing through it. PWM signals are much too fast and would quickly wear down a relay. This would also require more hardware and a higher cost than retaining the current method of direction control. A model of this mechanical relay is shown in Figure 22.

![Relay H-Bridge](image)

Figure 22 Relay H-Bridge
2.2.4.1.2 Semiconductor Design

A solid-state H bridge is usually constructed using MOSFETs or BJT transistors. Both of these semiconductors exhibit fast switching properties and allow for cascaded topology in situations where high current is necessary. The fast switching nature of MOSFET’s allows for PWM signals to control the amount of current being passed through to the motor. Most microcontrollers now come with multiple PWM channels which makes it easy to implement. An example functional circuit model is shown in Figure 23.

![Figure 23 Semiconductor H-Bridge](image)

2.2.4.2 Field Coil Polarity

The motor in our EZGO golf cart is a series wound DC motor. Series wound DC Motors operate on DC battery sources to provide a constant flow of power as opposed to AC motors which have very different characteristics. In shunt wound motors, the field windings and armature are connected in a parallel combination. It normally uses wire that is around 2-4 gauge in order to support high amounts of current. High current passes through the armature during the motor start up or when the motor is running at lower speed. As the motor’s speed increases, it generates an opposing electric field and reduces the amount of current passing through it. The direction selector switch box changes the way in which current flows through the motor. This is done by changing the polarity of the current flowing through S1 and S2. Figure 24 shows this process along with how the pre-charge circuit is handled in this particular EZ-GO golf cart. In the golf cart provided to us, the MS3 switch is what controls the pre-charge circuit. Once the pedal is pressed to a certain degree the micro-switch is activated that sends the entire battery pack voltage to that solenoid and engages the short instead of that 250 ohm pre-charge resistor. The motor controller can also tell when this is activated and knows to start accelerating.
2.2.5 Quadrature Encoder Interface (QEI)

QEI modules provide three signals which can determine the direction and rate of the motor. There are two signals slightly out of phase to tell which direction it is spinning and the amount of phase difference gives speed. The third signal is an index which gives a frame of reference for the other two. The TivaC MCU has the built-in capability to handle these types of signals and determine speed. An example of the QEI signals and counters is shown in Figure 25.

![Figure 25 QEI Direction Sense Timing Diagram](image-url)
2.3 Charge Controller

One of the most important devices to consider when using solar panels to charge batteries is a charge controller. Simply put, a charge controller is a charge regulator. The main purpose is to prevent batteries from overcharging. It regulates the voltage and current coming from the solar panels going to the battery. If there were nothing to regulate the charge, the batteries would become damaged from overcharging. This is accomplished by the charge controller realizing when the battery is starting to become full and preventing a current backflow. The charge controller will sense that the battery is full and stop feeding it with a current.

There are two main types of charge controllers on the market today. They are Pulse Width Modulation (PWM) charge controllers and Maximum Power Point Tracking (MPPT) charge controllers. PWM charge controllers are much more cost efficient as compared to the MPPT type.

The main disadvantage is that they do not work well with high voltage panels. PWM is often used as a method of float charging. This is done by the controller sending out a series of short charging pulses to the battery instead of a steady current. The device will constantly check the status of the battery and determine how fast and how wide to send the pulses. If the battery is almost completely full the charge controller might just send a “tick” of charge to the battery every so often. On the other hand, if the battery is near full discharge the device may send very long and wide pulses or go into a full on mode and adjust accordingly as it becomes more charged.

The panels we are using will provide us with a nominal ~60 volts. Compare that with respect to the 36 volts the batteries will provide, the PWM charge controller would not work well because the input voltage from our panels is nearly double that of the battery bank. So while the panels should be able to output 60 volts, the batteries are going to receive their 36 volts for full charge but we will be losing the other 20 volts or so. This is why a MPPT charge controller is the better option in our case.
2.3.1 Maximum Power Point Tracking (MPPT)

Before we can begin discussing types of charge controllers or design ideas we must have an understanding of how maximum power point tracking (MPPT) works. A MPPT charge controller is a DC to DC converter that finds the best match between the solar panels and the battery bank. More simply, it converts a higher voltage DC output from the solar panels to a lower voltage needed to charge the batteries. Therefore, the charge controller looks at the output of the panels and compares it to the battery voltage. It then figures out the most efficient power point and then converts this to the best voltage to get maximum amps into the battery. It essentially takes a high DC voltage, converts it to a high frequency AC and then back to match the best DC voltage for the battery. Considering the graph in Figure 26 we can see how MPPT drastically provides a 24 V battery with the maximum charge. Nowadays this is usually done digitally and are controlled by microprocessors. Sometimes these are called Smart Power Trackers. These microprocessors also help to adjust the maximum power point when temperature and light conditions vary throughout a typical day.

![MPPT VS. NON-MPPT](image)

Figure 26 MPPT for a 24 V system
(Reprinted with permission pending from SolarJourneyUSA.com)
For our project the solar panels are specified as producing 30.7 volts at 8.15 amps which is equal to 250 watts. Essentially the point of the MPPT is maintain and utilize all 250 watts. As an example, if the battery bank is half depleted and is currently producing 18 volts, the charge controller would take the 30.7 volts at 8.15 amps and convert it down. So now, instead the batteries would be receiving 18 volts at 13.9 amps. This would allow the panel to charge the battery bank faster while still receiving all 250 watts.

2.3.1.1 MPPT IC Options

There are several options when deciding what kind of MPPT integrated circuit to consider. Many companies offer MPPT circuits that can incorporated into a design for a solar charge controller. There are also many companies which solely produce charge controllers. However, for design experience and cost awareness, this is not the route in which we have chosen. This leaves us with the task of designing our own MPPT solar charge controller. In order to design an effective charge controller we must find a device that supports both the input from our solar panels, and the size of the battery bank. This is crucial in determining if the device will efficiently charge our battery bank or burn up the IC instead.

Most of the MPPT IC options being considered are made by Texas Instruments (TI). The first that we will look at is the TI bq24650. It is a high efficiency synchronous switch-mode charge controller primarily used in small applications. It supports a 5 V-28 V input solar panel and supports a battery up to 26V. This could be a good option for our project, because it is a straightforward all in one device. However, it does not support our battery bank size or the input voltage from the solar panels.

Another option is the TI TIDA 00120. This is a solar MPPT charge controller created for solar panel inputs corresponding to 12V and 24V panels along with 12 and 24V batteries. This device, unlike the bq24650, is designed for scalability. It can easily be adapted to a 48V system by changing the MOSFETs to 100V rated parts. The current can also be increased by using a specific set of MOSFETs. It has been tested to be 97% efficient at full load in a 24V system. Figure 27 shows the efficiency for the test data recorded by TI for a 24V system.
This device uses a method called “perturb and observe” algorithm to acquire the MPP. It provides fast acquisition of the MPP operation. In this algorithm, the controller adjusts the voltage by a small amount until power no longer increases. It relies on the rise of the curve of power against voltage below the maximum power point, and the fall above that point. Figure 28 shows test data from TI using their TIDA device to acquire its maximum power point in a 24V system. This method is the most common method used in MPPT devices because of its ease of implementation.
Another option is to use one of TI's many Solar Magic devices implemented with a MPPT digital controller. This would provide an alternate option to designing our charge controller. Instead of using a straight MPPT device, we would be a microprocessor chip along with a MPPT digital controller. This design requires many separate components to do the job of one device. However, on the plus side this would allow for a little more design scalability on our end; allowing us to pick and choose certain components as needed. The downside to this would be cost. An example design would be the SM3320-BATT-EV charge controller reference design built by TI. It implements a SM72442 MPPT digital controller and SM72295 photovoltaic full bridge driver designed to control high-efficiency DC/DC conversion used in solar applications. The Vin range for this design is 15V to 45V and is up to 98% efficient. It can provide up to 9A charging current and 14.2V max charging voltage. For this reference design, TI built and tested it in the application of charging a 12V car battery. Figure 29 shows the lead-acid charging profile used in this reference design. It can be seen from the data that as the voltage rises past the threshold point the full charge current is applied. Once full charge is detected the system switches to a floating charge and maintains the battery voltage at a certain threshold. This particular device is designed to run in MPPT mode at any time the available power is lower than the power required to achieve voltage or current regulation.

Figure 29 Lead-Acid Charging Profile
(Reprinted with Permission Pending from TI)
2.3.2 Sensors

By incorporating voltage and current sensors into the golf cart, we can monitor and diagnose the power dissipated throughout the system. The sensors will be used to measure the voltage, current and power for the battery bank, solar panels and the electrical instrumentation. By determining the power input and consumption, the total battery life will be determined. Using these calculations, the maximum driving distance in each driving mode can be calculated. Incorporating this with the GPS unit, the quickest and most power efficient route can be used.

Using the voltage and current sensors will provide a reading of the net power. In order to determine the maximum range in each drive modes, tests will be conducted and the net power will be recorded. Running these test in multiple conditions will give a good reading on the maximum distance the golf cart can travel.

In addition, a speed sensor will be used to determine the current speed the golf cart is traveling. In order to calculate the speed, either a physical sensor will be placed on the axle to count revolutions per minute, or the speed will be determined from the GPS module.

2.3.2.1 Voltage Sensors

Voltage sensors will either be placed over each of the six 6 volt batteries or for the combined 36 volt battery bank. The voltage sensors will also be included to measure the voltage input from the solar panels and the voltage drop to the electrical instrumentation. In addition, the voltage readings can either be visually displayed or just used internally for distance calculations. Table 6 compares a few considered voltage sensors.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Voltage</th>
<th>Current Consumption</th>
<th>Error</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phidgets RB-Phi-86</td>
<td>-30 to + 30 V (DC)</td>
<td>3.6 mA</td>
<td>±0.7%</td>
<td>$19.00</td>
</tr>
<tr>
<td>RB-Dfr-120</td>
<td>4.5 to 30 V (DC)</td>
<td></td>
<td>±1%</td>
<td>$6.80</td>
</tr>
<tr>
<td>Phidgets CE-VJ03-32MS2</td>
<td>0 to 250 V (AC)</td>
<td>15mA</td>
<td>±0.5%</td>
<td>$115.00</td>
</tr>
</tbody>
</table>

Table 6 Voltage Sensor Options
2.3.2.2 Current Sensors
Current sensors will also be included in order to determine the power input and dissipation in the golf cart system. Using these sensors will allow for the current to be measured and used with the voltage sensors to measure the power consumption. The current sensors will be implemented in series with each measured component. Table 7 contains a list of options considered.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Current Measuring Range</th>
<th>Sensitivity</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFRobot RB-Dfr-149</td>
<td>-50 to 50 A</td>
<td>40 mV/A</td>
<td>$14.50</td>
</tr>
<tr>
<td>Polou RB-Pol-171</td>
<td>-30 to 30 A</td>
<td>66 mV/A</td>
<td>$8.95</td>
</tr>
<tr>
<td>Polou RB-Pol-127</td>
<td>-30 to 30 A</td>
<td>133 mV/A</td>
<td>$9.95</td>
</tr>
</tbody>
</table>

Table 7 Current Sensor Options

2.3.2.3 Speed Sensors
There are two main ways to determine the speed of the golf cart: physical sensor to determine revolutions per minute (RPM) of the axle and using GPS on the HUD to measure speed. Using the physical axle-mounted speed sensor will provide an accurate and simple speed calculation. On the other hand, depending on the GPS module, the accuracy of the speed calculations can vary when using GPS for speed calculations.

2.3.2.3.1 Physical RPM Sensors
One type of physical speed sensor is a gear tooth sensor, which mounts on the axle to determine RPM. The teeth on the rotation wheel passes a stationary magnet causing a large magnetic flux at the moment when the magnet is closest to a tooth on the gear. By determining the when the magnetic flux is at its largest value, a waveform of the revolutions can be determined. From this waveform, a speed value can be determined using the time and distance between gear teeth with respect to the magnet.

Another way to determine RPM would be to place equally spaced magnets on the axle and use a stationary sensor to report when the distance from sensor to magnet is at its closest. With two equally spaced magnets, the magnetic flux would peak twice in each revolution. The sensor would return a digital value ‘high’ at maximum magnetic flux. This would create a waveform that can be used to determine speed.

2.3.2.3.2 GPS Speed Sensors
The speed of the golf cart can also be determined using the GPS module. From the GPS signal, the velocity can be determined and used accordingly.
2.3.2.4 Distance Calculations

Using the power accumulation and consumption details determined by the voltage and current sensors, the maximum travel distance of the golf cart must be determined. For each drive mode, the maximum travel distance will be dynamically calculated. The sensors will determine the power accumulated from the solar panels and the amount of useable power in the battery bank. In addition, the power dissipated will also be calculated from the power used to run the motor, HUD, and all of the electrical components. Once the power accumulated and the power consumption is determined, a function for the net power can be calculated. These functions will allow for the system to accurately predict the maximum distance the golf cart will be able to travel on the current charge with help from the solar panels.

2.3.3 Charging Techniques

There are several charging technique approaches to consider for our project. Charging lead acid batteries can be a long process. Typically the batteries would receive a full charge overnight from the wall outlet, but for our design part of the purpose is sustainability. Our goal is for the solar panels to charge the batteries in addition to the wall outlet. This poses a problem, as there is only so much daylight, and the panels provide only so much charging power. It is important that we do not partially charge or over charge the batteries as this can cause damage. The charge time from the wall outlet may only take 5 hours to charge the whole battery bank, but from the solar panels it could take up to 5 times that.

In order to avoid overcharging the batteries, it is important to supply each cell with the correct voltage and current. A 6V battery typically has 3 cells. It is important that the voltage per cell is not too low, as this will result in no charge to the battery. The typical charging voltage for a lead acid battery is around 2.2 volts per cell. The charging voltages will vary with temperature and however much charge the battery currently has. The charging voltage for a fully depleted cell could be much higher than that of a full cell. For these reasons it is important to know the best charging technique.
2.3.3.1 Constant Voltage
A constant voltage charging source acts mainly as a DC power supply to each terminal of the battery. Typical lithium ion battery chargers use this method because it is easy to avoid overcharging the battery. The constant voltage means that once the maximum voltage for the battery is reached it automatically stops charging. Figure 30 shows the battery current and voltage after a long duration of time charging with this method. The charging current automatically starts to decrease as the voltage approaches the same voltage as the power source.

![Figure 30 Constant Voltage Battery Characteristics](Reprinted with Permission from Panasonic)

2.3.3.2 Constant Current
Constant current charging is a technique that varies the voltage depending on the charge of the battery. As the name implies, the charging source tries to maintain a constant level of current while changing the voltage. This method allows the separate battery cells to rebalance which can restore the performance back to its original factory distributed state. The circuitry required to design a constant current charger is much more complex than constant voltage because it has to monitor the voltage being output from the battery so that it stops supplying current once it becomes full. Figure 31 shows the difference in characteristics between constant voltage and constant current.

![Figure 31 Constant Current vs. Constant Voltage](Reprinted with Permission from Panasonic)
2.3.3.3 IUI Charging
IU is a combination of constant current and constant voltage charging techniques. It first charges with constant current to balance out the difference in cell levels and then switches to the constant voltage method to charge quicker and stop charging in a safer manner. It is also safer for the initial charge because the current pulled through constant voltage initially is very high. The first part of the charging cycle is called the bulk charge phase in which constant current is used until a preset voltage value is reached. Then constant voltage is used to quickly charge the batteries to the point at which the current drops to a preset value. The last phase is when the source switches back to constant voltage at a slowly diminishing rate.

2.3.3.4 Pulsed Charge
Pulsed Charging is a method of using signals similar to PWM that pulses at different widths to vary the power being applied. There is a small discharge pulse in between each charge pulse which helps to break up crystal growth and passivation. Less harmful chemical reactions occur at the surface of the electrode. This is also great for digital chargers that need to sample the voltage in between each charge pulse to see when the voltage is getting close to its maximum.

2.3.4 Solar Panels
Solar Panels are a cheap and effective way to harness energy from the sun and help power the gold cart along with the charge from a wall outlet. The cart we obtained for this project is already equipped with two Grape Solar© High Efficiency Mono-Crystalline Photovoltaic panels. Unfortunately, these two panels alone will not be strong enough to completely charge the battery bank, but the idea is that they will at the least help prolong the discharge time of the batteries. Realistically, this should considerably help with the power consumption from all of our onboard electronic subsystems as well.

The two roof-mounted solar panels will be used to convert the energy from the sun into usable electricity. These panels will be used to charge the batteries in order to provide power to the golf cart. In order to harness as much energy as possible from these solar panels, the wiring configuration to the motor and subsystems is critical. Wiring these panels in series yields the most voltage to charge the batteries. Another option would be to wire these panels in parallel to produce a higher current, however; this could cause power loss due to one of the panels receiving less sunlight.
2.3.4.1 Watts Needed
Since the project specifies that the cart will not only be charged with the solar panels, but with a wall outlet as well, we do not have to worry about having all of our power coming from the solar panels. In order to at least have an idea of the watts we would like to have from our panels, we must have an understanding of the total wattage the cart and all will require.

As can be expected, the batteries are going to provide power for everything on the cart. Since batteries are rated in amp hours, we can use basic math to give them a watt hours rating. Since \( P(W\,h) = Ah \times V \), with \( Wh \) equal to watt-hours and \( Ah \) equal to amp-hours, we can use the standard amp-hour rating from the batteries to solve for watt-hours. Typically, batteries are rated at a 20 hour rate. We will assume the same for watts. This means each battery provides 1,392 Wh. Again, there are six batteries total all connected in series, so that number will be multiplied by six to give a total of 8,352 Wh. Since that is for a 20 hour rate, that implies that the battery bank will provide ~417.6 watts at a nominal 36 volts for 20 hours. Therefore, hypothetically we would like to at least be able to have enough watts to assist in how long this steady power lasts. Ideally the nominal 500 watts we will be receiving from the solar panels should suffice. As previously stated, the panels will by no means be able to charge the battery bank itself. The plan is that it will be able to provide us with just enough energy to help drastically increase the drive time per discharge.

2.3.4.2 Solar Cells
Solar cells, also call photovoltaic cells, are simply an electrical device that converts the energy of light into usable electricity. The process in which solar cells convert sunlight into electricity is called the photoelectric effect. This is defined as the ability of matter to emit electrons when light is shone on it. Solar panels achieve this effect by using semiconductors, mainly silicon within the cells to create spare electrons and harness these spare electrons into a steady current which can be used as power.

The two panels we are using are made up of sixty cells each. They are high efficiency solar cells with quality silicon material for high module conversion efficiency and long term output stability and reliability. It is important to take into effect the operating temperature of the panels. Operating outside of these temperatures can result in a loss of power production.
The operating temperature for the panels being used is -40°C to +85°C (-40°F to +185°F). Although the chances of going outside of this range is very slim, we do have to consider the effects of low and high temperatures that we will experience. The cart will be operated in Florida and the temperatures here can be pretty intense sometimes. Solar cells tend to decrease in power production once they reach a certain temperature. The drop off in efficiency tends to be around 87 to 91°F, which is not that uncommon in Florida. Solar cells are most efficient in lower temperatures. Preferably in a climate that has cool temperatures and plenty of sunlight. Unfortunately this is not the case here. For that reason we must consider having to cool the panels down if they begin to overheat.

2.3.4.2.1 Mono-Crystalline Silicon Solar Cells

Monocrystalline silicon cells are made from a single silicon crystal, which provides the highest efficiency of around 15-20% among the commercial market. These cells would provide the best power input, but are generally more expensive. The Grape Solar panels we are using are rated at 15.4% efficiency. Monocrystalline solar panels are also space efficient. Since they yield the highest power output compared with other types of solar cells, they require the least amount of space. Monocrystalline solar cells also tend to produce more electricity in low light conditions. This is always an advantage because weather can be very unpredictable. Another advantage with using monocrystalline solar panels is that they work well in warm environments. Typically temperature effects, primarily overheating or not a concern when using these types of cells.

2.3.4.2.2 Polycrystalline Silicon Solar Cells

Polycrystalline silicon is the most common commercial solar cell material used. Unlike monocrystalline silicon, polycrystalline silicon is composed of many smaller crystals. This leads to a reduced efficiency, as well as, reduced cost. Polycrystalline solar cells tend to have less heat tolerance than monocrystalline, which means that they usually perform worse in higher temperatures. However, usually it is negligible enough that it is not very important. The efficiency rating of polycrystalline solar cells is around 13-16%. Because of the lower silicon purity, they are not as efficient as monocrystalline solar cells. It is easy to distinguish polycrystalline cells from monocrystalline by their more of a blue tint. This is due to the other crystal elements that these cells are made up of. This would not be our first choice due to the lower efficiency that polycrystalline cells have.
2.3.4.2.3 Thin Film Solar Cells
Thin film solar cells are basically made by depositing thin layers of photovoltaic material onto a substrate. There are different types of thin film solar cells, each depending on what kind of material is deposited onto the substrate. A few are, Amorphous Silicon, Cadmium telluride, copper indium gallium selenide, and organic photovoltaic cells. Depending on which type is being used, efficiencies can range between 7-13%. They are very cheap to mass produce which makes them much more cost effective than crystalline cells, but with a much lower efficiency. There are some other advantages, such that they can be made flexible and high temperatures and shading have less of an effect on energy production. Since thin film solar cells have such low efficiencies they require a large amount and space of them to compensate for energy production. This alone makes them not feasible for our project.

2.3.4.2.4 Amorphous Silicon Solar Cells
Amorphous Silicon (a-Si) solar cells are a sub category of thin film solar cells as mentioned above. These types of cells are made of single or multiple thin layers stacked on top of each other on a substrate. One of the advantages is that it can be stacked onto many different types of substrate. Whether it be glass, metal, or plastic. They can also be cut into different shapes, unlike crystalline cells. The efficiency of a-Si solar cells is typically around 7%, with a considerable decrease in its first few hours being in the sun. Although it is a very cost effective option, these types of cells would not be efficient in providing us with a substantial amount of power production.

2.4 PCB Design
In order for errors to be reduced in this system, printed circuit boards (PCB) will be used to permanently configure the electric circuits. PCB’s offer a much more secure and dependable connection between circuit elements then breadboards. Breadboard allow for easy manipulations and very cost efficient analysis of a circuit, but do not offer a permanent solution for use with the golf cart. Vibrations and movements encountered in this project can affect the connections between the circuit elements. For this reason, a breadboard will be used first to test and debug the circuits, then a PCB will be designed and purchased. Once the circuit is properly designed and debugged, the circuit will be drawn up using a schematic software. From this schematic, a PCB can be ordered and constructed.
2.4.1 PCB Composition

There are two main ways to implement a circuit design: with the use of breadboards or printed circuit boards (PCB’s). Breadboards allow for easy circuit alterations, but are not a trusted permanent design. Printed circuit boards are extremely consistent and accurate, but do not allow for simple changes to the circuit. PCB’s are circuit boards that use conductive tracks as wires between components. PCB’s allow for the circuit characteristics to be constant. Using a PCB also reduces the chance of wiring errors and loose connections. Once a circuit schematic is designed and debugged, a PCB can be ordered. PCB’s are often machine-made, which allows for a permanent design and reduces errors.

PCB’s are made up of two parts: a substrate and the copper wiring. The substrate is the main part of the board that provides a solid backing for the various circuit elements. Substrates materials are insulators, meaning they do not conduct electric currents. The substrates are split up into three main types: FR-2, FR-4, and RF. FCB substrates can also be either rigid or flexible depending on the application.

FR-2 is generally the cheapest substrate option, composed of a synthetic resin bonded paper to make the board flame resistant. The use of a FR-2 board in a moving condition is generally not recommended, because the constant vibration on the board can cause cracks in the FCB.

FR-4 is a fiberglass-reinforced laminate sheet used as a substrate for PCB’s. FR-4 is a more expensive option then FR-2, but FR-4 substrates offer better technical characteristics. FR-4 is flame retardant and has good water-resistant properties. This allows for the FR-4 to be used in humid conditions without diminishing any mechanical and electrical properties. Because of the flame resistance, minimal water absorption, and ability to be used in a vibrating setting, the FR-4 substrate would be a good option for use on the golf cart in the humid conditions encountered in Florida.

RF substrates are designed to perform favorably in high power radio frequencies. RF substrates offer great electrical performance properties, but lack with mechanical aspects. These substrates are composed of low dielectric plastics which allow for them to be used with high frequencies. PCB’s are mostly designed to be rigid. Rigid PCB’s are cheaper and easier to make, and provide good mechanical properties. In addition to the more common rigid PCB’s, some are specially designed to be flexible. These flex substrates are generally more difficult to manufacture, which in turn increases the price. Flex substrates are designed using thin and flexible films. Flexible PCB’s are intended to be used in tight, specific places to save space.
2.4.2 PCB Software
For the design of the printed circuit boards in this project, Eagle CAD software can be used to design and construct a suitable circuit design. Eagle CAD provides an easy to use interface along with a broad spectrum of resources. Using the schematic editor and layout editor, a circuit design will be designed and sent to a manufacturer to construct. The schematic file will be used as a guide to create a PCB for use in this project. Eagle CAD uses a universal Gerber file for use with multiple manufacturing vendors.

Another easy to use software for PCB design is ExpressPCB. ExpressPCB’s software provides users with simple and efficient interface. ExpressPCB is functional enough for a professional, while still providing a favorably learning curve. ExpressPCB software is very limited, with use only with the ExpressPCB manufacturers. The limited compatibility can be an issue when ordering a PCB.

Also, PCB Layout Software is another free PCB design software. This software is efficient and useful for PCB design. PCB Layout Software is a compatible program with use with 4PCB manufacturing.

2.4.3 PCB Manufacturer
PCB Pool is a manufacturing company that uses Eagle CAD software for developing PCB’s. PCB Pool offers a large variety of board substrate options and broad selection of compatible software. PCB Pool allows for single and double-sided circuit boards. Due to the low production costs, PCB Pool is a good option for designing circuit boards for use with this project.

Another suitable PCB Manufacturing company is OSH Park. Osh Park accepts Eagle CAD circuit design files for the manufacturing of PCB’s. OSH Park offers very competitive pricing with three copies of a two layer board for $5 per square inch. They also boast a quick shipping time at only 12 days after ordering. In addition, OSH Park also offers three copies of four layer boards for $10 with a two week shipping time. OSH Park is a good option due to the cheap production cost and free shipping worldwide.

ExpressPCB is another option with free compatible PCB design software. PCB circuit designs can be implemented quickly and efficiently. The ExpressPCB service sells three copies of two and four layer PCB’s for $51. Shipping is also very quick and cost efficient. ExpressPCB guarantees two day delivery for PCB’s. ExpressPCB’s free software and efficient shipping combination provides a well-rounded option.
In addition, 4PCB is another suitable PCB manufacturing company. 4PCB provides printed boards for a competitive price. Using a student discount, a two layer PCB can be purchased for $33. Also, 4PCB also sells a four layer PCB for $66. 4PCB works with a free “PCB Layout Software” for seamless design and manufacturing. 4PCB also include a free PCB file check to ensure the design is correct and complete.

2.5 Batteries
One of the most important parts of an electric golf cart whether it is solar powered or not, is the batteries. The batteries will be used to power every aspect of the cart, from the electric motor to all of the onboard electronic systems. More importantly, for most solar electric systems there is a battery bank that is used to store the energy captured by the PV panels to be used when the system is off the grid or is not in prime conditions. For our case the batteries will be getting energy from both the PV panels and the wall charger and then being converted and used to power the cart and its subsystems.

Golf carts along with many small solar electric systems use deep cycle batteries. Deep cycle batteries are designed so that they can be discharged down as much as 80% time after time, and not suffer internal damage. One of the reasons deep cycle batteries can be discharged to such a degree is because they have thicker plates. Plate thickness in a typical golf cart battery is around .07 to .11” thick. In small solar electric systems golf cart batteries are widely used for power storage. Because golf cart batteries are typically deep cycle batteries, they have the capacity for continuous service with many charge/discharge cycles for many years. This is a key component in solar power systems because the PV panels will be constantly charging the batteries, while the batteries are also being depleted.

2.5.1 Options
There are three main types of golf cart batteries on the market today. There is the gel cell, AGM, and flooded-lead. The most practical and cost efficient is the flooded-lead. This is also the type that comes standard on the specific golf cart we are using, the E-Z-Go TXT. However, for sake of research and design specifications all three types will be considered for our design. Standard golf cart batteries are typically either 6 V or 8 V deep cycle flooded-lead batteries. Some 12 V options can be found, but are not as practical. The E-Z-Go TXT can be setup using either a 36 V or 48 V battery power systems. This can be obtained by using a series of 6, 8, or 12 V batteries.
There are many aspects to consider when choosing the right battery to use for our cart. We want to maximize our range and efficiency to best fit our specifications, while at the same time staying within a reasonable budget. The batteries chosen will be based on the following criteria:

1. Since we are trying to increase range and efficiency, the batteries to be chosen must have a high energy density.
2. The batteries will also need to have an efficient charge to discharge ratio since it will be receiving charges from both a wall outlet and solar panels.
3. The batteries chosen must be able to be combined in a series pattern for a total of 36 V or 48 V set up.
4. Since most deep cycle batteries are rated in a 20 amp hour rate, the batteries will need to have the highest rate within budget.
5. The batteries will need to have enough voltage output to efficiently power all motor and electronic subsystems etc. on the cart.

Therefore, in order to determine the best battery selection, all of the above mentioned battery types will be researched and analyzed.

2.5.1.1 Gel Cell

Gel cell batteries, or “gelled” batteries, are one of the 3 main types of deep cycle batteries in the market for both golf carts and solar electric systems. Gel cell batteries are usually completely sealed on the top so that the acid cannot spill or leak out. Since there are no vents or caps on the battery, they do not release gas when being charged. Gel Cell batteries contain acid that has been slightly mixed with Silica gel, giving it the name “Gel Cell”, which turns the acid into a solid gooey mass. This results in the cells having much less fluid in them as compared to the typical flooded or wet battery.

There are a few advantages and disadvantages to analyze in determining if Gel Cell batteries are the correct choice for our project. The main advantage is that they cannot spill or leak, nor do they release gas when being charged. This allows for use indoors, but this is not an advantage necessarily applicable to our project. However, there are several disadvantages. They must be charged at a slower rate to prevent excess gas from damaging the cells. When using a charge controller or inverter, gel cell batteries would have to be charged at a lower charging current. Since there will be two power sources for our batteries, this is not efficient. We will need a high charging current to provide a quick charge for our specifications.
2.5.1.2 AGM

AGM, or Absorbed Glass Mat, batteries are another typical type of battery found in solar bank applications. The name comes from their design, which has a woven glass mat between the plates to hold the electrolytes. They are similar to the Gel Cell batteries in that they are spill and leak proof, and that they also do not release any gases while charging. They have all the advantages of the Gel Cell batteries, but can withstand the abuse of higher discharges.

AGM batteries have many advantages to consider in our decision process. Besides being completely sealed and immune to leakage or spilling, the charging voltages are the same for any standard battery. Theses batteries do not require any special adjustments for charging as does the Gel Cell. Another advantage is that AGM batteries have a very low self-discharge. This allows for a much longer period of rest without the need to charge them. This battery would definitely meet our criteria in the aspect of performance. However, there is one main disadvantage that defers us from choosing the AGM battery. That disadvantage is the price. The typical AGM battery is anywhere from one and a half to two times as much as the typical deep cycle battery.

2.5.1.3 Flooded-Lead

The last option to consider is the standard deep cycle flooded-lead battery. These batteries are the most popular with solar bank applications and electric golf carts alike. Flooded-lead batteries consist of cells fill with lead acid. On the top of the battery itself, there are caps that allow for water to be added. Unlike the first two types of batteries discussed, flooded-lead batteries are not completely sealed and can leak. They also release gases when being charges and therefore should not be used indoors. Flooded-lead batteries have a high energy density and an efficient charge to discharge ratio. This means that they are able to withstand deep discharges and still maintain efficiency and longevity.
Compared with an AGM battery of similar amp hour (Ah) rating, the expected life cycles versus depth of discharge (DOD) is almost exactly the same (see figure 36). Figure 34 shows that even at 100% discharge of its discharge it will maintain as many life cycles as the more expensive AGM battery. Another criteria mentioned is the charge to discharge ratio. For our application, the batteries will be receiving a constant charge from the solar panels along with a full charge from a wall outlet. This means that the more important aspect to consider is the time of discharge; especially with respect to current. Since there will be different modes of output for the cart and other subsystems running on the same battery power we will consider a discharge current of around 75 amps per battery for analysis. This will provide us with a little over 2 hours of battery life respectively (see figure 37). From these analysis the flooded-lead battery is the most reasonable battery of choice for our design. It will meet all of the required criteria and stay within a reasonable budget.

Figure 32 Expected Life Cycles vs. DOD
(Reprinted with Permission Pending from US Battery)
Many makes and models were considered for our specifications and project. Based on performance, efficiency, and cost, we decided to go with the US 2200 XC2 battery made by US batteries. It is a flooded-lead acid deep-cycle battery. It has a 20 Ah (amp hour) rating of 232 which is slightly higher than that of its closest competitor, the Trojan t-105. It is also more reasonably priced and is well within budget. Figure 38 shows all of its specifications and ratings.
2.5.2 Configuration

A group of batteries wired together to attain a specific voltage or amperage is called a battery bank. The key to attaining and maximizing this voltage or amperage is directly related to the wiring configuration. For solar applications, battery banks are typically used in order to harvest and store the energy collected from the sun. Typically batteries are wired in either a series or parallel configuration, much like a circuit. The same rules apply for wiring batteries as in a circuit. Batteries connected in series will produce a higher voltage output, while batteries wired in parallel will provide a higher current output. Often, in solar applications, battery banks are configured in both series and parallel configurations together.

2.5.2.1 Series

A series configuration is made by connecting one pair of opposite terminals of different batteries together. For example, this can be done by connecting the positive terminal of one battery to the negative terminal of another battery. This type of configuration allows for an increased output voltage. The voltages of each battery connected in series is simply added together to produce a final output voltage. However, like in a circuit, this configuration will keep the amp hour capacity the same as that of one of the batteries in series. We are more concerned with having a higher total voltage in order to power the cart and all of its subsystems. For this reason, a series configuration is the best option as it will give the most voltage.

2.5.2.2 Parallel

The second type of configuration to consider is parallel configuration. A parallel connection is made by connecting like terminals of two or more batteries together. This can be accomplished by connecting two positive terminals together and two negative terminals together. This type of configuration increases the amp hour capacity while maintaining the voltage the same as one of the batteries connected in parallel. From this diagram it is clear that the Amp hour capacity \( Ah \) = \((number \ of \ batteries) \times (battery \ ah \ rating)\). Because our project is more heavily voltage dependent, this is not the most efficient configuration to use.

Often battery banks, especially in solar applications, are connected in both series and parallel. This is done by connecting two sets of batteries, which are composed of batteries connected in series, in parallel with each other. For example, four 6 volt batteries connected in series from positive terminal to negative terminal are connected in parallel with another four 6 volt batteries connected in series. The advantage to using this configuration is to meet a specific voltage and amperage, depending on its use or application. However, again this is not a good option for our project, because we do not have the space to give us the necessary voltage rating.
2.6 Automated Driving Features

Autonomous vehicles will be everywhere in the future. Incorporating automated driving features to this golf cart will allow for a safer vehicle. Using the power generated by the solar panels, a microcontroller and sensor(s) will be configured to automatically brake and control speed. The range finder sensor(s) will dynamically determine the distance to the nearest object in the path of the golf cart and send a signal to the motor controller to reduce speed or break in order to avoid a collision.

2.6.1 Sensors

For this system, the sensor needed will have to be cost efficient, highly accurate and perform in a desired range. First, the correct distance range will have to be determined. The sensor must detect objects far enough away to give the vehicle enough time to stop before collision. The optimal sensor needed for this project must be used to return the distance from the object, not just if the object is in a certain threshold distance. There are three main types of distance sensors to choose from: infrared, ultrasonic, and laser.

2.6.1.1 Infrared Sensors

There are two main ways for an infrared sensor to determine distance, reflected light and triangulation. For reflected light infrared sensors, the infrared (IR) sensor emits a specific light in the IR wavelength spectrum. When this wavelength becomes in contact with an object, the wavelength bounces off the object and returns to the sensor. The sensor only reads the IR wavelength, and all other spectrums are ignored. The time from the infrared ping to the wavelength returning to the sensor is used to calculate the distance. Converting the time to distance allows for the sensor to determine the distance to the obstacle at that moment. The speed of light in air is used for the calculation of the distance.

In addition, triangulation can be used by the infrared sensors to determine the distance to the sensor. The IR sensor first emits an IR wavelength, then, when the wavelength becomes in contact with an object, the signal is returned to a lens. Using the distance from the IR pulse and the lens, along with the reflected angle, the distance to the object can be determined using trigonometry.
Also, infrared sensors can provide either a binary or analog output. IR sensors with binary outputs will only be able to determine when an object is in a threshold distance. For example, by setting the threshold distance to 5 feet, the binary IR sensor would just return a binary ‘1’ when the object is within this distance. On the other hand, the sensors with analog outputs can return the actual distance to the object from the sensor. For this system, the golf cart must reduce speed when an object is in the current path. There must be multiple threshold distances so that the golf cart can slow down safely and efficiently. For this project, multiple binary sensors could be used with different distances, or an analog sensor could be used to work with multiple threshold distances. With that in mind, the IR sensors with binary output would not be the best choice for this system, despite the lower cost. Table 8 lists a few considered infrared sensors.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Supply Voltage</th>
<th>Operating Supply Current</th>
<th>Output Terminal Voltage</th>
<th>Detection Range</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sharp GP2Y0A710YK IR</td>
<td>4.5V to 5.5V</td>
<td>33mA to 50mA</td>
<td>-0.3V to 5.3V</td>
<td>100cm to 550cm</td>
<td>Analog</td>
</tr>
<tr>
<td>XL-MaxSonar EZ0</td>
<td>3.3V to 5V</td>
<td>NA</td>
<td>NA</td>
<td>10cm to 765cm</td>
<td>Analog</td>
</tr>
<tr>
<td>Sharp GP270AYK0F IR</td>
<td>4.5V to 5.5V</td>
<td>33mA to 50mA</td>
<td>-0.3V to 5.3V</td>
<td>20cm to 150cm</td>
<td>Analog</td>
</tr>
</tbody>
</table>

Table 8 Infrared Sensor Options

The main problem with using infrared sensors is the lack of accuracy when exposed to sunlight. For use on the golf cart in an outside setting, the wavelengths from the sunlight can interfere with the IR sensor. Also, the maximum distance for these IR sensors do not extend far enough for to work properly with this system.

2.6.1.2 Ultrasonic Sensors

Ultrasonic sensors work similarly to infrared sensors, except ultrasonic sensors emit a sound wavelength instead of the light wavelength used from the IR sensors. Ultrasonic sensors use sonar to detect objects. The sensor first emits a chirp of inaudible sound. When this sound wave encounters an obstacle, the wavelength reflects off the object. The sensor detects the returned wavelength and records the time between the chirp and return echo. Using this time and the speed of sound in air, the microcontroller can be used to determine the distance to the object.
Using sound waves instead of infrared waves allows for the sensor to be used outside in direct sunlight. Despite the higher cost for ultrasonic sensors, it is the best choice for this project. Possible problems with ultrasonic sensors include sound interference, absorbent objects, and ghost echoes. Sound interference can come from vehicles, buildings and electric machinery. Sound waves from these outside elements can create a false positive for the sensors. Also, certain materials may absorb the sound waves instead of reflecting them, which would cause the object to be virtually invisible to the sensors. Another possible issue is ghost echoes, which occurs when the sound wave reflects off multiple objects to return a false positive.

Due to a limited accurate range, the use of an ultrasonic sensor as a rangefinder for this project will not be included. Ultrasonic sensors provide a better accurate reading in direct sunlight then IR sensors, but do not have the required range to detect an object in time to slow down the golf cart to rest.

2.6.1.3 Laser Sensors
Laser rangefinder sensors use an extremely concentrated beam of light to determine the distance to objects. Laser signals are often very narrow, but provide great accuracy. Similar to the infrared sensors, laser sensors emit a focused beam of light and measure the time it takes for the light to return to the sensor.

Laser rangefinders emit a beam of light and the reflected light is returned to the sensor. The distance is then determined using the reflected angle using a concept called triangulation. Triangulation provides a very accurate reading; however, the error percent for the distance increases as the distance reaches the maximum range for the sensor.

Another laser rangefinder technique is to use a rotating mirror to disperse the laser across a wider area. This provides an accurate way to determine distance to an object as well as the width of the object. The distance is calculated based on the time it takes for the laser to bounce back to the sensor off of an object.
The use of a rotating mirror laser rangefinder would be ideal for use on this golf cart. Laser rangefinder sensors provide a much more accurate distance reading and the ability to work in direct sunlight make it a better choice than using an IR or ultrasonic sensor. The main problem with including a laser rangefinder with this golf cart is the cost. Laser sensors provide a much better accuracy and range than the other types of sensors, but the cost is extremely high. Some of these sensors considered are listed in Table 9.

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Part Number</th>
<th>Measuring Range</th>
<th>Response Time</th>
<th>Light Source</th>
<th>Output Type</th>
<th>Laser Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>SICK DT35-B15251</td>
<td>1057652</td>
<td>50 mm to 12,000 mm</td>
<td>2.5 ms</td>
<td>Laser, red</td>
<td>1x/2x push-pull: PNP/NPN (100mA), IO-Link</td>
<td>2 (EN 60825-1)</td>
</tr>
<tr>
<td>SICK DL35-B15552</td>
<td>1057657</td>
<td>200 mm to 35,000 mm</td>
<td>2.5 ms</td>
<td>Laser, red</td>
<td>1x/2x push-pull: PNP/NPN (100mA), IO-Link</td>
<td>1 (EN 60825-1)</td>
</tr>
<tr>
<td>SICK DL60-N111B</td>
<td>1026360</td>
<td>300 mm to 24,000 mm</td>
<td>130 ms</td>
<td>Laser, red</td>
<td>1xNPN (100mA)</td>
<td>2 (EN 60825-1)</td>
</tr>
</tbody>
</table>

Table 9 Laser Sensors

2.6.2 Range Finder
Incorporating a range finder sensor will be used to determine the distance to the nearest obstacle in the path of the golf cart. The data from the rangefinder will be sent to the motor controller to reduce speed. The sensor will determine if there is an obstacle in the current path and using a microcontroller, the distance to the object can be determined. Using multiple case statements, the distance read from the sensor will be compared with multiple threshold distances. The golf cart will then reduce speed to a desired value when the input distance is within one of the threshold distances. For the threshold distances to be determined, the maximum velocity of the golf cart must be determined. Once this velocity is determined, the time it takes to slow down to a stop must also be calculated. From these values, the threshold distances can be determined. The goal is for the golf cart to reduce speed as the sensor detects an object and stop completely to avoid a collision.
Without knowing the specific maximum speed and braking distance for this golf cart, it is
difficult to estimate the threshold distances and threshold speeds. Once the maximum
speed and braking distance is determined, the threshold distances can be
calculated. Using several threshold distances will allow for the golf cart to smoothly
reduce speed instead of an abrupt stop. Once the object is within the closest threshold
distance, a negative signal will be sent to the motor controller to stop the golf cart.

The threshold distances will also be dependent on the type of rangefinder sensor
used. The rangefinder sensor must have the following characteristics:
1. Ability to work in direct sunlight
2. An accurate, measurable maximum distance appropriate to the braking distance
3. Analog output
4. Fast response time
5. Cost within budget

2.6.3 Cruise Control
In addition, a cruise control option will be added to allow for the golf cart to maintain a set
speed. Once the desired speed is achieved manually, a button can be pressed to engage
cruise control. Incorporating the range finder sensors, the golf cart will also slow down
when an object is in the current path. A driver notification device will also be active to
alert the driver to upcoming obstacle. In addition, a switch will be added to the brake
pedal to determine when the brake is pressed. When the cruise control is activated,
pushing the break will disengage the cruise control.

2.6.3.1 Cruise Control Switch
A manual button will be used to initiate cruise control. At the moment the switch is
pressed, the current speed will be maintained until the brake has been applied or an
obstacle is observed in the current path. The button will also be used to disengage the
cruise control when needed. A separate button will be placed under the brake pedal in
order to disengage cruise control when the brake is pressed.

2.6.3.2 Driver Notification
In order to create a safe driver experience, an audible driver notification should be
implemented to alert the driver of an obstacle in the current path of the golf cart. This will
be a horn that makes a noise when a collision is imminent. The notification device will
also alert the driver when cruise control is disengaged so that the driver is aware.
2.6.3.3 Microcontroller

A microcontroller would be needed for the autonomous features to be implemented. The microcontroller would connect to the sensors and receive the analog output of the rangefinder. These values would represent the distance to the nearest obstacle ahead of the golf cart, if there was one. The measured distances would be compared to different threshold distance criteria. If the measured distance falls in one of these distance categories, a signal would be sent to the motor controller to reduce the speed accordingly. As the distance to the object decreases, so does the velocity of the golf cart. Figure 46 shows the flow chart of the threshold distance and velocities.

![Figure 35 Threshold distance and velocity flow chart](image)

Ideally, the microcontroller should work with two front-mounted distance sensors to provide a maximum width of detection. Additionally, the microcontroller must interpret each sensor’s signals as an analog distance measurement. Table 10 shows the possible microcontrollers and related specifications.

<table>
<thead>
<tr>
<th>Microcontroller</th>
<th>Processor</th>
<th>Processor Speed</th>
<th>Analog Pins</th>
<th>Memory</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raspberry Pi</td>
<td>ARM1176JZF-S</td>
<td>700 MHz</td>
<td>None</td>
<td>RAM 512 MB</td>
<td>$39.99</td>
</tr>
<tr>
<td>BeagleBone Black</td>
<td>Sitara AM3359A ARM Cortex-A8</td>
<td>1 GHz</td>
<td>7</td>
<td>DRAM 512 MB DDR3L, eMMC 2 GB</td>
<td>$44.99</td>
</tr>
<tr>
<td>Galileo</td>
<td>Intel Quark SoC X1000</td>
<td>400 MHz</td>
<td>6</td>
<td>8MB Flash, 512KB SRAM, 256MB DRAM</td>
<td>$84.99</td>
</tr>
</tbody>
</table>

Table 10 Microcontroller options
2.6.4 Software
The software used for this microcontroller will depend on the choice of microcontroller. A simple case statement can be used to compare the current distance to the threshold distances and change the velocity accordingly. The output will send a signal to the motor controller to reduce the speed if necessary. The microcontroller will receive the inputs from the rangefinder sensors and provide an output to the motor controller.

3 Design
3.1 HUD
The cart’s Heads-Up Display system was designed using multiple software solutions as a result of the system containing both software design and PCB circuit design. The final outcome from collaboration of the two areas of design will result in a testable Heads-Up Display prototype. The following sections will elaborate upon each step of the Heads-Up Display system’s development process.

3.1.1 Android Application Development
The software for the Heads-Up Display system will be loaded on the BeagleBone Black microcomputer in the form of an Android Application running on the Android 4.4 KitKat mobile environment. This application will consist of four classes to accomplish every requirement established in the research phase, these classes are the following: MainActivity.java, CartStatus.java, UserNav.java, and Settings.java. The user will be brought to the MainActivity.java class upon application execution and this class will handle requests from the user to navigate to the various functions throughout the application. Once the user clicks the “Navigate” button on the MainActivity.java class menu the Android Application will start a new activity to invoke the UserNav.java class. The UserNav.java class invokes the SupportMapFragment method to generate an interactive Google Map object using the Google Maps API and will also generate waypoints on the Google Map to help guide users around the UCF campus. The CartStatus.java class is also accessed through the MainActivity.java class in the same manner as the UserNav.java class. This class handles all status signals both to and from the Motor Controller and also from the GPS so that it can properly display all cart status values to the user as well as transmit any operational mode changes to the Motor Controller. The Settings.java class is accessed in the same way as the CartStatus.java and UserNav.java classes from the MainActivity.java class and will provide the user with options to change the display brightness, color, and contrast. All class interactions and contained methods are shown in detail in Figure 47.
3.1.1.1 Navigation Screen Development

The Heads-Up Display navigational page will be developed using Android SDK version 21.0 in the Eclipse IDE. The navigation page will be implemented via its own class called UserNav.java and make use of the Google Maps API in order to generate an interactive map fragment on the page in order to provide users with the necessary interactive functions and position information. The map object will immediately set its displayed view to UCF’s GPS coordinates for the user convenience of being able to quickly have access to navigation around the UCF campus. This starting camera position will be set through modification of variables in the Google Maps API support library.
The modified variables values will be the GPS coordinates of the UCF campus which would be “cameraTargetLat = 28.6016” and “cameraTargetLng = -81.2005”. Through use of the Google Maps API library functions the generated map will contain a “My Location” button that when pressed will use the GPS tracking functionality of the current prototype to pinpoint the user’s current location on the navigational map. This function’s data input, processing, and output will be handled entirely by the Google Maps API library setMyLocation() function using data inputs from the current prototype’s GPS coordinate tracking instrument. A set of waypoint markers will be drawn to the map interface to help users identify specific landmarks and classroom buildings within the UCF campus using the Google Maps API library. These waypoints are generated through use of the Google Maps API library which gives the developer the ability to generate a marker by declaring a new google.maps.Marker generic variable and then passing desired values to the new object for placement and naming purposes. The variables that would be passed values are “position”, “map”, and “title”. The “position” variable is passed the desired GPS coordinates for the markers position on the map, the “map” variable is passed the name of googleMap object that will indicate which map on which to draw the specified marker, and the “title” variable is a string that is passed to indicate which text will be shown when the user hovers on the respective marker on the map interface. This design will guarantee that the UserNav.java class will handle all user gestures, map generation, marker positioning, and raw GPS data passed to the Android Application by the GPS coordinate tracking instrument on the Heads-Up Display PCB. All data flow and method interactions within the UserNav.java class is detailed in Figure 48.

Figure 37 UserNav.java Detailed Data Flow Diagram
3.1.1.2 Cart Status Screen Development

The Cart Status Screen of the Heads-Up Display system is also developed using the Android SDK version 21.0 and Eclipse IDE. The Cart Status Screen will have its own class called CartStatus.java that will contain the necessary methods to generate menu items and interactive buttons. It also contains the proper methods to forward data to the Motor Controller when the user selects a cart mode as well as methods to handle incoming data from the Motor Controller and GPS coordinate tracking instrument to generate the Range Left, Battery Percent, Cart Speed, and Cart Operating Mode. The communications between the CartStatus.java class with the Motor Controller and the GPS coordinate tracking instrument along with status updates are detailed in the state diagram of Figure 49 and the function data flow between the different value generation functions diagramed in Figure 50.

![Figure 38 CartStatus.java Communication and Status Update State Diagram](image)

![Figure 39 CartStatus.java Display Value Function Data Flow Diagram](image)
3.1.1.2.1 Cart Speed and Current Mode

The Current Speed function will be responsible for handling receive data input from the GPS coordinate tracking instrument and for returning a displayable velocity value to the CartStatus.java class. Its function input values will be values regarding the data for the user’s current and previous GPS coordinate positions that were measured by the GPS coordinate tracking instrument. Using this data the function will measure the distance between the cart’s most recent position and previous positions to generate an estimated velocity of the cart. The velocity variable value is then converted to unit of miles per hour instead of meters per second; this operation is done by multiplying the meters per second variable value by a factor of 2.23694. This calculated value is then returned back to CartStatus.java in the form of an integer to be displayed through functions calls into the proper field on the “Cart Status” screen interface. This process will continue to loop without interruption in order to provide users with an up-to-date value at all times. The Current Mode function will not only take input from the user’s touchscreen input data but also from the Motor Controller module’s mode acknowledgement signals. As well as being the middleman between the user and the Motor Controller module, it will also verify that the cart is in the correct user-selected mode and then display the current mode as a string for display in the Current Mode field of the CartStatus.java class. Its function input from the user will be a variable indicating which mode the user selected on the “Cart Status” screen using the mode buttons, if the user has not yet selected a mode since power-on the default operating mode is “Standard”. Once this value has been passed to the function it will make a function call to the BeagleBone Black’s GPIO interface for to send a signal informing the Motor Controller to change the cart’s operating mode to the respective sent value. After transmission the Android Application will wait until the Motor Controller reports back through the BeagleBone Black’s GPIO pin connections and then fetch the transmitted signal values to be used in a conditional operation. The aforementioned conditional operation will check to make sure the acknowledgement signals from the Motor Controller match the new operating mode value, if they do match then the Current Mode function will return the value of the current mode in the form of a string for display on the “Cart Status” screen. If a mismatch occurs then the cart will repeat the requested mode change in a retransmission to the Motor Controller and afterwards go through the same conditional operation again until the values finally match. For testing purposes in the early prototypes the Current Mode function will provide dummy signal values from the Motor Controller module since the Motor Controller won’t be providing signals to the BeagleBone Black PCB system until later prototypes.
### 3.1.1.2.2 Range Left and Battery Percentage

The Range Left and Battery Percentage Functions both require electric signals being received from the Motor Controller Module to the BeagleBone Black’s GPIO pins. Therefore these functions will not be fully implemented and tested until the later prototypes of the Heads-Up Display system.

The Battery Percentage function will be responsible for taking function input from the Motor Controller and then returning a displayable value to CartStatus.java for the Battery Percentage field on the “Cart Status” screen. Its function input values are received from the Motor Controller module giving a quantitative value indicating how much battery charge remains in the cart system. This value is assigned by calling functions to check the input signals received from the Motor Controller on the BeagleBone Black’s GPIO pin connections. The received value is then rounded to nearest integer and returned to the CartStatus.java class for display in the proper Battery Percentage field of the “Cart Status” screen. This function will continuously loop and run uninterrupted for reasons of supplying the user with the most up-to-date values on the power remaining in the cart.

The Range Left function will receive input from the Current Mode function, Speed function, and the Motor Controller and then using these inputs generate a range left value to output back to CartStatus.java for display in the Range Left field on the “Cart Status” screen. The input from the Current Mode function will be passed through to the function through the function’s call in CartStatus.java using the variable value from the user’s input gesture. The data input from the Speed function to the Range Left function will come in the form of an integer calculated in the Speed function that represents the cart’s current velocity. The Motor Controller module’s data input to the Range Left function is a quantitative value indicating how much battery charge remains in the cart’s power reserves. Much like the Battery Percentage function this value is assigned using the proper function calls, variable declarations, and GPIO pin signals except in this function the value is not rounded in order to be used in later calculations within the function. Once the Range Left function has received all the previously mentioned signals from all its inputs it will interpolate the supplied data to generate an approximated value for the range remaining to the user in the current operating mode. This approximated value is rounded for ease of display to the user on the “Cart Status” screen and then returned to CartStatus.java for display purposes. This value will be updated repeatedly through looping the function and sending up-to-date data to be displayed should the cart’s velocity or mode of operation change.
3.1.2 Heads-Up Display Hardware Configuration

In order to meet the specified requirements of running an Android Application and displaying pertinent information to the cart’s users, the Heads-Up Display needed the proper hardware. For the Heads-Up Display system to run a navigational Android Application it will need a microcomputer as a platform for the Android OS in conjunction with a GPS module for user coordinate information. The display of information from the microcomputer running the Android Application will be accomplished by a resistive touchscreen display with appropriate connections for the microcomputer. In order to accomplish all these operational requirements a specialized PCB must be designed and built.

3.1.2.1 PCB Design

When beginning the design process for the PCB that will house the BeagleBone Black board and the GPS module the designer must decide which pins will be connected between the two chipsets and which software program will provide the most suitable design environment. With these requirements in mind it was decided that the schematic editor and PCB generator software to be used is EAGLE 7.1.0 Freeware version which is developed by CadSoft Computer GmbH in Germany. Within the EAGLE software a BEAGLEBONE_SHIELD pinout schematic was added via the Adafruit EAGLE part library as well as a GPS_FGPMMOPA6H pinout schematic for the MTK MT3339 GPS Chipset from the Adafruit EAGLE part library.

The first step to designing the PCB schematic that will provide the appropriate electrical connections between these two module schematics is to add voltage source objects to illustrate where each module will be connected to a power source and indicate the values of the placed sources. Thus two 5V sources and one 3V sources are added to the schematic editing area through navigation and use of the EAGLE support library. The two 5V sources are then linked to the BeagleBone Black module schematic by creating electrical connections between the two P9 VDD_5V pins thus providing an illustration of how the BeagleBone Black module will receive power. The 3V source is to be linked to the MTK MT3339 GPS module schematic by creating an electrical connection between it and the VCC pin on P$1 of the MTK MT3339 GPS module thereby effectively creating the visual representation of how the MTK MT3339 GPS module will be powered during operations.

The next step in the schematic design procedures is to add six GND elements to the schematic editing area through interface with the EAGLE support library. Two of the GND icons are connected to the GND pins on the BeagleBone Black’s P8 pin array thereby giving a visual representation of the GND connections on the P8 pins. Another two of the GND elements are connected to the GND pins on six GND pin connections on the P9 pin array of the BeagleBone Black which generates the graphical representation of the GND connections on the P9 pins. Another one of the GND elements is placed on the left side.
of the MTK MT3339 GPS module and is used to connect the GND pins P$3 and P$8 to a GND signal. This then shows the virtual representation of the P$3 and P$8 GND pins on the MTK MT3339 GPS module being connected to an effective GND. The last GND element added to the design is then placed on right side of the MTK MT3339 GPS module to connect the module’s P$19 and P$12 GND pins to a GND signal. This connection’s generation creates a visual representation of the P$19 and P$12 GND pins connection to a GND signal.

The final step in the schematic design plans is to connect the RX and TX pins on the MTK MT3339 GPS module to two of the BeagleBone Black’s P9 GPIO pins which in effect will allow the BeagleBone Black and the MTK MT3339 GPS module to communicate between each other. The mentioned pin connection is to be included in the PCB schematic design by way of connecting the MTK MT3339 GPS module’s P$9 TX pin and the GPIO_7 pin on the BeagleBone Black’s P9 pin array. This connection visually represents the electrical connection between the P$9 TX pin on the MTK MT3339 GPS module and the GPIO0_7 pin on the BeagleBone Black’s P9 pin array that will transmit GPS tracking information from the MTK MT3339 GPS module to the BeagleBone Black’s Android OS environment. The final pin connection on the PCB design schematic is the electrical connection between the MTK MT3339 GPS module’s P$10 RX pin and the GPIO1_28 pin on the P9 pin housings on the BeagleBone Black module. This connection provides the virtual representation of the electrical connection between the MTK MT3339 GPS module’s P$10 RX pin and the GPIO1_28 pin on the BeagleBone Black’s P9 pin array that will transmit pertinent data from the BeagleBone Black’s Android OS signals to the RX pin on the MTK MT3339 GPS module. With all these connections in place the BeagleBone Black system and the MTK MT3339 GPS module will receive enough power to perform normal device operations such as hosting the Android OS environment for the UCF EzNAV Android Application to execute seamlessly on the BeagleBone Black module or finding GPS coordinate information from onboard GPS coordinate location measurement hardware on the MTK MT3339 GPS module. Another function implemented through the electrical connections in the PCB schematic is the ability for the BeagleBone Black’s Android OS to information to receive data as well as transmit data from the MTK MT3339 GPS module to collect accurate values on the GPS coordinate information of the PCB that will supply the UCF EzNAV Android Application with the required GPS coordinate data to provide a navigation system to the Heads-Up Display’s users. These connections’ and two boards’ schematic are illustrated in Figure 53, which was created in the EAGLE software.
3.1.2.2 Display System

The display used for video output requires far less designing and integration when compared to the other two hardware components in the Heads-Up Display system. The HDMI 4 Pi: 5" Display comes with three assembled PCB’s to support video input, touchscreen capabilities, and a menu system for adjusting the display. The driver board PCB has an HDMI input, a USB power cable, and a 50pin to 40pin + AR1100 Adapter that will be connected to the BeagleBone Black via the HDMI connection for video input and connected to the cart’s onboard power system through the USB power cable to power the display system’s PCBs. The touchscreen adapter PCB has a USB miniB output and a 40pin to 50pin adapter which will be connected to the BeagleBone Black through the USB miniB output to the BeagleBone Black’s USB A Standard input in order to transmit the user’s touchscreen input to the Android Application on the BeagleBone Black. The wired PCB with the five buttons to control a menu system for the display will not be used and will most likely be hidden from the users in the final cart design since all display setting changes will be done through the Android Application’s settings menu.
3.2 Motor Controller

A lot of the design techniques and information used in designing this motor controller was taken from an open source motor controller online. This open source motor controller is part of the “Open Revolt” project that helps people convert their cars to electric power affordably.

3.2.1 Power Board

The power board of a motor controller has to contain a capacitor bank, switching mechanism (MOSFETS in this case), and diodes to prevent feedback of large voltage from when current stops being applied to the motor. There are three terminals on any motor controller: B+ (positive battery pack voltage), B- (negative battery pack voltage), and M- (the terminal that controls the path from B- to M-). The PWM signal will open the path from B- to M- and this allows current to flow through the selected direction in the motor. The direction is controlled externally in this case however. The original schematic of the Open Revolt power board controller is shown in Figure 54. The number of diodes was calculated to divide the amount of current evenly without going above a dangerous temperature. The diode current was assumed to be safe at the amount designed in the open source controller because it was designed around motors much larger than a golf cart motor. Capacitance values were left the same because the voltage levels on golf cart batteries is actually more stable than in the system it was designed for so there was no reason to raise the amount.

![Figure 54 Original Schematic of Open Revolt Power Board](Image)

(Reprinted with Permission from Open Revolt Source)
3.2.1.1 PCB

The PCB used for the motor controller in this project will come straight from a vendor for the “Open Revolt” project. In order to make a PCB of this thickness and current capacity, it would be much too expensive to design one our own. It would take a professional license which costs anywhere from $500 to $5000. To save time and money on that part of the design, it is easier to purchase from a vendor. The PCB is shown in figure 55.

Figure 55 Power Board PCB
(Reprinted with Permission from Open Revolt Source)
3.2.1.2 Diode
The diodes used in the open source power board were chosen because the power board was designed around 144V supply. The golf cart will use significantly lower voltage. Lower voltage diodes tend to come with higher current carrying capability and better recovery properties. STPS80150CW is the part number that was decided on because it can handle higher current, does not have as excessively high voltage requirements, and has better recovery time.

3.2.1.3 MOSFET
The power MOSFET’s chosen in this were also different than the ones used in Open Revolt’s original design. Higher voltage MOSFET’s typically come with higher on resistance and lower current carrying capabilities. This is why IRFP7718PbF was chosen instead of the open source standard. It has a much more appropriate voltage range for the golf cart and much lower on resistance. This means that less heat will be generated at equivalent driving current. The amount of current capable of flowing through these MOSFET’s is much higher than it needs to be, but this allows for a lower duty cycle to create equivalent power. Lower duty cycles mean less heat which is good for the circuit.

3.2.1.4 Capacitor
A replacement capacitor had to be chosen because of the discontinuation of P11622-ND. 381LR821M200J042 is also an equivalent value capacitor that has the same dimensions. The same dimensions had to be retained due to the PCB being milled specifically for this part. A quantity of 16 capacitors totaling 820uF seemed more than ample enough to settle out changes in the battery pack voltage so that was not changed.

3.2.1.5 Power Board BOM
Table 13 contains a list of the cost and parts for this board.

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Description</th>
<th>Quantity</th>
<th>Cost</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRFP7718PbF</td>
<td>Power MOSFET</td>
<td>10</td>
<td>6.38</td>
<td>63.80</td>
</tr>
<tr>
<td>381LR821M200J042</td>
<td>820 uF capacitor</td>
<td>16</td>
<td>3.60</td>
<td>57.60</td>
</tr>
<tr>
<td>STPS80150CW</td>
<td>Diode</td>
<td>10</td>
<td>5.88</td>
<td>94.08</td>
</tr>
<tr>
<td>N/A</td>
<td>Power Board</td>
<td>1</td>
<td>55.00</td>
<td>55.00</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td></td>
<td>270.48</td>
</tr>
</tbody>
</table>

Table 11 Power Board BOM
3.2.1.6 Heat Dissipation

Due to the high amount of current that needs to be carried through this power board, there will naturally be a lot of heat generated. This heat is going to need to be dispersed well in order to save the components from damage. To dissipate heat, aluminum blocks will be used to create a large area for the ground. Aluminum is a great heat spreader and can be easily cut down to the dimensions needed at the UCF milling center. The dimensions needed are 0.75”x1.5”x8”. These were determined by the size of the pre-fabricated PCB that is being used. After using a block of this size to give a good ground to the MOSFET’s and diodes, another large sheet will be connected on top of that to give a large surface area for air flowing to pass over.

3.2.1.7 Copper Bars

The power board will need low resistance contact points for the B+, B-, and M- terminals. If a small contact point is made with only the edge of the board, this may add more resistance and create heat and loss of power. To avoid this, copper bars of dimension .25”x.75”x11.8” will be used. These sizes were pre-determined by the open revolt motor controller group.

3.2.2 Control Board

TIVAC 129 XL is the microcontroller development board being used as the logic control part of this design. It will have battery power supplied from a TI Launchpad battery pack during the testing phase and will eventually switch over to a more permanent dc-dc converter that can be used with a relay from the ignition switch. A general block diagram of the control logic is shown in figure 56.

![Figure 56 Control Logic Diagram](image)
3.2.2.1 Battery Booster Pack

The first phase of integration will be done with testing using a TI Fuel Tank BoosterPack. This will simplify the process of debugging power supplies because it is known to work properly with any Launchpad.

3.2.2.2 DC-DC Logic Power Supply

Final versions of the board will be powered by a DC-DC power supply. It will run off of either 6V from one battery or 12V from two. This supply voltage will be fed through a relay upon switching of the ignition key. Once passed through, this supply voltage will be dropped down to 3.3V using OKI-78SR-3.3. A separate 5V supply will be controlled by OKI-78SR-5 voltage regulator. Both of these should be able to provide stable voltage to the TIVAC’s 3.3V and 5V power layers.

Both of the permanent DC-DC power supplies come with the following features:

- Ultra wide 7 to 36 VDC input range
- Fixed Outputs of 3.3 or 5 VDC up to 1.5 Amps
- “No heat sink” direct replacement for 3-terminal
- High efficiency with no external components
- Short circuit protection
- Outstanding thermal performance

3.2.2.3 PWM

The TM4C129LNCZAD microcontroller found in the TIVAC MCU being used includes the following:

- Eight advanced PWM outputs for motion and energy applications
- Four fault inputs to promote low-latency shutdown
- One Quadrature Encoder Input (QEI)

Despite having eight PWM outputs, all gates of the MOSFET’s are going to be driven by the same channel to avoid phase delays. The fault inputs will be used to avoid overcurrent situations. The motor on this golf cart was originally sold with a 36V 300A setup, but was designed to be able to handle up to 48V and 500A. Using the current sensor chosen, the software will constantly monitor how much current is passing through the motor and use the fault inputs to shut down the MOSFETs in case of an overcurrent situation. Quadrature Encoder Input would be useful in determining the speed of the motor, but will not be used in this project.
3.2.3 Software
Two major software packages were considered during the research portion of this document: Code Composer Studio and Energia. Energia is easiest method of implementation and is therefore the obvious choice.

3.2.3.1 Energia
The energia development environment is full of examples and already links these examples to the TIVAC board. The setup function is where you declare inputs and outputs while the loop function is the code that executes repeatedly while the board has power applied.

3.2.3.1.1 analogRead
analogRead() is the function that energia uses to access the ADC library from this TIVAC board. It will map the 0-3.3V signal to a range of digital values from 0-4096. This number comes from the TIVAC having a 12-bit ADC that has a maximum of 3.3V and is 5V tolerant. This means that the software will read 4096 for anything above 3.3V, but no damage will occur as long as it stays below 5V.

3.2.3.1.2 analogWrite
analogWrite() function uses the pre-existing libraries to create a PWM signal. This PWM signal switches on and off at 3.3V digital logic and gives an average between 3.3V and 0V depending on the duty cycle. This is what will be used to control the gate voltage going into the MOSFETs depending on the pedal’s position.

3.2.3.1.3 digitalRead
This function will provide simple communication between the Beaglebone black and the TIVAC to determine the modes of operation. It simply senses a one or zero on the input of a GPIO pin and returns it to a variable.

3.2.4 Sensors
For this design the TIVAC will need access to the current passing through the motor, the speed of the golf cart, what mode is currently selected, voltage of the battery pack, and voltage coming back from the pedal.

3.2.4.1 Pedal
The pedal needs 14V applied to one end and the other end will output anywhere from 0.4V to 1.5V. This is directly in the range of what the ADC on the TivaC can handle so this will be used directly to an analog input. A microcircuit also outputs a high digital level
once the pedal has been pressed down to a certain point. This provides a digital point for the control software to start. As well as providing the voltage to the ADC, the micro-switch activating will also trigger once the pedal is pressed down far enough to contact the solenoid. This will be read into a GPIO to tell the software that it is ok to output a PWM signal.

3.2.4.2 Current Sensor

For the main current sensor that will read the battery pack supply current, the L01Z500S05 will be used. It can handle up to 500 amps and runs off of 5V. The only downside to this part is that the output range goes up to 5V which the TivaC MCU cannot handle on the ADC pins. It will need to be divided down in a similar fashion to the pedal supply with an op amp so that math can be performed on it. Having this part in the golf cart will make it much safer because there will be a current limit for the motor. The output characteristics are shown in Figure 59.

![Saturation characteristic(L01Z600S05)](image)

**Figure 59 Current Sensor Output Characteristics**
(Reprinted with Permission from TamuraCorp Licensing Rights)
3.2.4.3 Battery Voltage Sensor

The voltage sensors being used for the project will monitor both the battery pack voltage and the pre-charge voltage. The software will not allow current to flow through the motor until the capacitors are charged up to the same voltage as the battery pack. This circuit is already currently implemented with the pedal micro-switch and the solenoid.

3.2.4.4 Non-Volatile Memory

The TivaC non-volatile memory is automatically programmed into the microcontroller when using Energia. This means that power can be removed without losing the motor controller software program. This simplifies programming greatly compared to other methods of control such as an FPGA where loading from non-volatile memory is a much more difficult task.

3.3 Charge Controller

From all of the research the preliminary design for the charge controller will be designed using the TIDA 00120 Solar MPPT Charge Controller. This was determined due to its scalability and operating ranges. The circuit design itself will be modeled using the reference design on the TI website. The MOSFETs will be changed to 100V rated parts to allow for a higher input voltage. In order to increase the current to 40A the MOSFETs will be switched with the TO-220 package versions. Along with the TIDA device are many other analog and digital components to make a highly integrated mixed signal circuit. Figure 60 shows the reference block diagram. Again, this is merely a reference design and our final design may be different.

Figure 60 Reference Design Block Diagram
(Reprinted with Permission Pending from TI)
3.3.1 Power Stage

This particular design is designed around the MSP430F5132 controller IC. This is a mixed signal Microcontroller and acts as the brains for the charge controller. There are two stages to this design. First is the Power stage which consists primarily of analog devices and a SM72295 photovoltaic full bridge driver. This device is used to drive the MOSFETs in a full bridge configuration. The drivers provide 3A of peak current for fast switching and integrated bootstrapping diodes. Figure 61 below shows the reference design schematic layout for the Power Stage of the charge controller.

![Figure 61 Schematic of Power Stage](Reprinted with Permission Pending from TI)
3.3.1.1 MSP430F5132 Microcontroller

The device that will performing the majority of the algorithms and precision functions is the microcontroller. For the TIDA-00120 design we will be using a MSP430F5132 from TI. The Texas Instruments MSP430 family of ultra-low-power microcontrollers consists of several devices featuring different sets of peripherals targeted for various applications. The architecture, combined with five low-power modes, is optimized to achieve extended battery life in portable measurement applications. The device features a powerful 16-bit RISC CPU, 16-bit registers, and constant generators that contribute to maximum code efficiency. The digitally controlled oscillator (DCO) allows the devices to wake up from low-power modes to active mode in less than 5 µs. The MSP430F5132 series is a microcontroller configurations with two 16-bit high-resolution timers, two universal serial communication interfaces (USCI_A0 and USCI_B0), a 32-bit hardware multiplier, a high-performance 10-bit analog-to-digital converter (ADC), an on-chip comparator, a three-channel DMA, 5-V tolerant I/Os, and up to 29 I/O pins. This device helps the control strategy of the design.

3.3.1.2 SM72295 Photovoltaic Full Bridge Driver

The SM72295 will be responsible for driving the full bridge DC/DC conversion. The SM72295 is designed to drive 4 discrete N type MOSFET’s in a full bridge configuration. The drivers provide 3A of peak current for fast efficient switching and integrated high speed bootstrap diodes. Current sensing is provided by 2 transconductance amplifiers with externally programmable gain and filtering to remove ripple current to provide average current information to the control circuit. The current sense amplifiers have buffered outputs available to provide a low impedance interface to an A/D converter if needed. An externally programmable input over voltage comparator is also included to shut down all outputs. Under voltage lockout with a PGOOD indicator prevents the drivers from operating if VCC is too low.

3.3.2 Controller and Bias Supply Stage

The next stage is the controller and bias supply stage. This stage consists of the MSP430F5132IDA which was talked about previously. It also has LM5019MR which is a 100V, 100mA constant on-time synchronous buck regulator. It has integrated high and low side MOSFETs and the constant on-time control scheme requires no loop compensation. This regulator is great because it has a peak current limit circuit which protects against overload conditions. This stage also has a TLV70433DBV which is a 150mA, ultra-low IQ, high Vin low dropout regulator. The last digital device in this stage is the INA271 which acts as a voltage output, current-sense amplifier. This device is able to sense drops across shunt resistors at common mode voltages. Figure 62 shows the circuit schematic for the reference design of the controller and bias supply stage.
3.3.2.1 LM5019MR Buck Regulator

The LM5019MR is a 100V, 100mA synchronous step-down regulator with integrated high side and low side MOSFETs. The constant-on-time (COT) control scheme employed in the LM5019 requires no loop compensation, provides excellent transient response, and enable very low step-down ratios. The on-time varies inversely with the input voltage resulting in nearly constant frequency over the input voltage range. A High voltage startup regulator provides bias power for internal operation of the IC and for integrated gate drivers. A peak current limit circuit protects against overload conditions. The under voltage lockout (UVLO) circuit allows the input under voltage threshold and hysteresis to be independently programmed. Other protection features include thermal shutdown and bias supply under voltage lockout. Figure 63 shows the typical application of this particular Buck regulator.
For use with this project, a PCB will be designed to implement the motor controller. Once the circuit is designed and tested using a breadboard, the design will be finalized using a PCB design software. Using this design schematic, the PCB can be ordered and the manufactured PCB will be implemented into the golf cart system.

**3.4 PCB**

For the design of the PCB’s necessary for this project, Eagle CAD PCB software will be used to design and order a suitable circuit board design. Eagle CAD provides a thorough library of circuit parts, as well as the ability to design and add other components. Eagle CAD has an easy to use interface that makes creating circuit schematics very simple.
4 Prototype

4.1 HUD

After the project’s Heads-Up Display design phase has been completed the construction of the first prototype will begin and more prototypes will follow once more parts have been ordered and integrated into the overall design. Each iteration of the design prototype will undergo thorough testing, with the early prototypes being tested on a few key features whereas the later prototypes will be subjected to thorough testing and evaluation of all system features.

4.1.1 First Prototype

The first design prototype for the Heads-Up Display system is the Samsung SCH-I545’s Android OS environment. This first prototype will provide the Android OS and additional functions that are necessary to execute the UCF EzNav Android Application designed for this project. This prototype will be able to provide the required resources necessary to execute UCF EzNav Android Application, wirelessly connect to the internet, and gather GPS information all from the onboard hardware and software components of the SCH-I545. The SCH-I545 prototype environment will not be able to connect to the Motor Controller interface in any way as a result of not having any GPIO pins included in its design; therefore the only working component of the UCF EzNav Android Application is the “Navigate” screen and its accompanying functions. Figure 65 shows the SCH-I545 prototype’s main menu with the UCF EzNav Android Application launch button.

Figure 65 SCH-I545 Prototype Main Menu with UCF EzNAV Launcher Icon
4.1.2 Second Prototype

The next design prototype for the Heads-Up Display will be a PCB containing the BeagleBone Black microcomputer module and the MTK MT3339 GPS module connected together through the PCB. The BeagleBone Black will receive its necessary power through its two VDD_5V pins connected to the respective 5V rated source and receive GPS data input through pins GPIO1_28 and GPIO0_7, all on the P9 pin array. The MTK MT3339 GPS chipset will receive its power for operation through its VCC pin connected to a 3V rated source and will transmit the required GPS information through its TX and RX pin outputs on pins P$9 and P$10 respectively. The BeagleBone Black module will house the Android OS version 4.4 KitKat by way of an image contained on a microSD memory card inserted into the appropriate bank on the chipset. With both the Android OS environment and the connections between the two modules on the designed PCB this prototype will have the functionality to execute the UCF EzNAV Android Application and gather GPS information through communication with the MTK MT3339 GPS chipset. This version of BeagleBone Black and MTK MT3339 GPS PCB chipset will not be connected to the Motor Controller module although it does have the appropriate pin connections to allow such a attachment. Since this prototype version does not receive signals from the Motor Controller only the Speed and Current Mode fields will be generated on the “Cart Status” screen within the UCF EzNAV Android Application. A design concept of the PCB that will be used for this prototype is shown in Figure 66.

Figure 66 BeagleBone Black and MTK MT3339 GPS PCB Design
4.1.3 Third Prototype

The final design prototype for the Heads-Up Display system will be the BeagleBone Black and MTK MT3339 GPS PCB detailed in the previous section with wire wrap connections to the Motor Controller module. All power connections will remain the same as in the previous iteration of the prototype design however the BeagleBone Black will have input and output connections to the Motor Controller module by way of a number of its thirty GPIO pins located on the P8 pin array, which was unused by the previous prototype version. This design prototype will contain the necessary resources for the BeagleBone Black to forward signals sent from the UCF EzNAV Android application to the Motor Controller and allow the Motor Controller to send signals to the BeagleBone Black for the Android Application to manipulate as needed. In this prototype revision the BeagleBone Black will still be running the same Android OS version through the imaged microSD memory card; however in this revision the UCF EzNAV Android Application will be receiving all signals required for populating the “Cart Status” screen fields and therefore the Android Application can undergo all the prepared prototype tests.

4.2 Motor Controller

4.2.1 Phase 1

The first phase of prototyping the motor controller involves a setup with just the power board, microcontroller, and one of each component for the power board. An external motor and potentiometer will be used to simulate the environment that the motor controller will actually be in with a pedal and motor. It will just be scaled down and implemented with the bare minimum requirements of the system to prove that the concept works. An overview of this setup is shown in Figure 67.
4.2.2 Phase 2
After the concept has been proven to work from phase 1 testing, the next phase is to verify that the different modes of operation work. This phase can be tested by using a power supply to send high digital levels to the pins for different modes that would normally be sent by the HUD. After confirming that the software works correctly, the next step is to test with the actual HUD. This requires the HUD to be in working condition and capable of selecting different modes for test.

4.2.3 Phase 3
The final phase of prototyping will be done with the actual golf cart. All electrical connections will be finalized in this phase. To test this prototype, the motor controller must be completed. In the first two phases it only had one of each component whereas in phase 3, it needs to be fully capable of handling 500 amps at 36V because it is going to be run at full power in the golf cart. At this point in the development stage, the software should be fully capable, but may need some tweaking once it is able to be tested on the full sized motor.

4.3 Autonomous Features
The prototypes will be implemented in multiple steps to ensure the accuracy and efficiency of the system. The first prototype will include a distance display to test the rangefinder sensors. The system will then be connected to the motor controller to control the speed of the golf cart. Also the cruise control prototype will first be created separately from the distance sensors. The driver notification will also be added and the complete semi-autonomous system will be prototyped.

4.3.1 Phase 1
The first phase will be to test the accuracy of the rangefinder sensors. The sensors will be connected as the input to the microcontroller and a visual display will be used as an output. The sensor will then be tested with objects at different distances and measured distance will be compared to the distance displayed. Figure 68 shows the first prototype of the rangefinder sensors.
4.3.2 Phase 2

The next phase will be to incorporate the motor controller in order for the speed of the golf cart to be regulated. The motor controller will be used instead of the output display. A signal will be sent from the microcontroller to the motor controller detailing the speed at which the golf cart should be travelling. Figure 69 shows the second phase prototype with the microcontroller connected to the motor controller.
4.3.3  Phase 3
The next step will be to incorporate the cruise control. This will initially be separate to the rangefinder sensors and braking. A button will be connected to the motor controller and used to initiate and disengage cruise control. Another button will be placed under the brake pedal to deactivate the cruise control when the brake is pressed. Figure 70 shows the initial cruise control system.

Figure 70 Cruise control prototype

4.3.4  Phase 4
The next phase will include the rangefinder sensors and cruise control system in addition to the driver notification. Figure 71 shows the whole autonomous system included in this project.

Figure 71 Autonomous prototype
4.4 Charge Controller

4.4.1 Phase 1
The charge controller will have to go through a prototyping phase once it has been designed. This will be, but may not be limited to certain tests or criteria the initial design will have to undergo in order to determine if it is functioning properly. The first initial phase the design will go through is a computer simulation to make that the circuit is at least wired and grounded properly. If it isn’t we run into the risk of permanently damaging some of the components. Once it has received a thumbs up on the design it will need to be simulated using some sort of circuit analysis and design software. Most likely this will be done in Multisim. Multisim provides us an easy to use software to simulate different aspects of the circuit.

4.4.2 Phase 2
The next phase of prototyping will come after all simulations and once the prototype PCB controller is built. This will allow us to physically take the board into a lab and test it using an oscilloscope and other devices. From these tests we will be able to determine whether or not we need to adjust anything on the board or correct anything before moving onto the next phase. Again, it is important to note that the board used in this phase may not be the actual PCB being used. It could in reality be a handmade circuit built in the lab on a bread board.

4.4.3 Phase 3
The final phase of prototyping for the charge controller requires the final design. This will not be determined until after the second phase is complete and any changes or corrections have been made. This will involve the charge controller being connected to the solar panels and the battery bank and tested for efficiency and completeness. If the charge controller is able to properly and effectively harness the energy from the solar panels and convert it to a steady charge we will know it is on the right track. This phase of prototyping will require some time as we need to allow the batteries to near full charge. By doing this we will be able to make sure that the charge controller is properly keeping the batteries from overcharging and causing permanent damage to the batteries. Figure 72 shows the basic wiring diagram setup of a PV array system for charging a battery. It will look slightly different for our design because ours is attached on the roof of the cart, but this shows the basic concept.
5 Testing
5.1 HUD
Both throughout the design phase and upon creation of the first iteration of the Heads-Up Display prototype multiple tests will be performed on the system to insure proper functioning of all components. Each part of the Heads-Up Display system will be tested individually through development and will be tested upon integration of each part. Most of the preliminary testing will performed in a virtual environment and the later test components will be performed in both virtual and physical mediums.

5.1.1 Android Application Testing
The Android Application testing will be split into three different components which will be tested throughout the various stages of design integration. The different testing components will always be tested within the Android OS environment regardless of the current stage of design integration. The three individual components encompass the following subjects: Navigational Testing, Cart Status Report Testing, and Settings Modification Testing.
5.1.1.1 Navigational Testing

The Navigational Testing portion of the Android Application Testing procedures will be run within the Android Application developed for the Heads-Up Display system. The first Android platform that the Navigational Testing takes place on is the Samsung SCH-I545 running Android version 4.4.2 KitKat. The Android Application is sent to this device through the Eclipse IDE and is automatically installed and ready to execute. The SCH-I545’s Android OS has full access to Fine and Coarse GPS Coordinate information as well as full internet access. The second Android platform that will perform the Navigational Testing is the BeagleBone Black board with adjoining MTK MT3339 GPS module which will be running Android version 4.4.2 KitKat. The Android Application is installed on this environment via the Android OS image on the microSD memory card inserted in the BeagleBone Black microcomputer. While the BeagleBone Black’s Android OS does not have access to the internet it will be receiving Fine and Coarse GPS location data from the MTK MT3339 GPS module in order to appropriately identify the user’s current position.

The actual test portion of the Navigational Testing will consist of three tasks to complete on the “Navigate” application page. The first task to perform on the “Navigate” page will be a visual check that the Google Map Support Fragment correctly loads to the UCF GPS coordinates, evaluating this task will ensure that the map displays the UCF campus to the user upon starting navigational tasks. The second testing task involves clicking on the “My Location” button within the map interface and ensuring that it correctly approximates the user’s current GPS position, by testing this functionality it will be guaranteed that the map interface is correctly processing the received GPS coordinates. For a more detailed, visual explanation of this task please refer to Figure 73. The final task of the Navigational Testing is to click on the generated waypoints on the map interface to certify that they are in the correct positions and display the correct respective information. This final task is a crucial passing point since it will be a main component of providing any user’s accurate navigation around the UCF campus.
5.1.1.2 Cart Status Report Testing

The Cart Status Report Testing component of the Android Application Testing procedures will also be run within the Android Application developed for the Heads-Up Display system. Instead of running on only two testing platforms, the Cart Status Report Testing will require three testing platforms in order to properly gauge the effectiveness of the “Cart Status” screen through testing. The first Android testing platform is again the Samsung SCH-I545 running Android version 4.4.2 KitKat with all the previous data access permissions but this platform will not be able to provide the signals needed to gather the necessary cart status information for testing.
The second Android platform for testing is once more the BeagleBone Black board with the MTK MT3339 GPS module running Android version 4.4.2 KitKat. This Android platform will be able to supply some of the actual signals for the value fields of current speed and Mode of Operation but will be unable to supply the necessary data for the values of battery percent and range left for the purpose of testing. The last testing platform for this series of tests will be the BeagleBone Black and MTK MT3339 modules with the proper connections to the motor controller system. With these connections in place within the test environment all the data values on the “Cart Status” screen will be available to be tested since the Android Application will be receiving and transmitting all required signals to generate and test the value fields.

The interactive testing portion of the Cart Status Report Testing will be comprised of four tasks, some of which can only be completed on the later testing platforms. The first evaluation task is to be completed on all testing platforms, it entails checking that the “Cart Status” screen generated correctly and that users can easily navigate back to previous menus. The next evaluation task for the Cart Status Report Testing can only be done on the second and third test platforms, in this task the evaluator checks the values of the data in the “Speed” and “Current Mode” fields and ensures that they accurately match observed values. This test will confirm that the Android Application is correctly receiving data from the GPS module and generating the values correctly within the Android Application methods for accurate display to the users. The third task can also be completed on both the second and third test platforms, this task involves the evaluator clicking on one of the interface buttons and then checking the “Current Mode” field. This test procedure will denote that the Android Application is correctly setting the Mode of Operation variable. For a more visually detailed representation of this task please refer to Figure 74. The last task for this series of test instructions can only be completed on the final testing platform when the BeagleBone Black has been fully connected to the Motor Controller. This task has the tester check the remaining two data fields, “Range Left” and “Battery Percentage” for accuracy. By completing the task successfully the evaluator will be able to confirm that the BeagleBone Black and the Motor Controller systems are communicating relevant electrical data between them via their wire wrap connection.
Figure 74 Testing Elements of Second Cart Status Report Test
5.1.1.3 Settings Modification Testing

The Settings Modification Testing section of the Android Application Testing will be run using the designed Android Application on the Heads-Up Display system. This group of tests will be administered on all the Android Application testing environments much like the first grouping of Navigational Testing. With the first test platform being the Samsung SCH-I545 and the last two platforms being the BeagleBone Black with the MTK MT3339 GPS module and the BeagleBone Black/GPS PCB connected to the Motor Controller via wire wrap. Among all the platforms, the tasks completed will be entirely from the “Settings” screen.

The hands-on evaluation of the “Settings” screen will have three tasks to assess the overall functionality of the “Settings” screen’s elements. The first task that the evaluator will complete on the “Settings” screen is to touch and drag the selection bar next to “Brightness” to differing degrees of display brightness. This test will examine the “Brightness” selection bar’s functionality and will determine if the Android Application is adjusting the screen’s brightness factor respectively. The second of the three evaluation tasks will have the evaluator adjust the “Contrast” selection bar slider to different positions on the bar area. By performing this task the evaluator can ascertain the functionality of the “Contrast” selection bar and if the Android Application is adjusting the screen’s contrast correctly to valid input. The last task with the Settings Modification Testing component has the tester drag the “Color” selection bar to various points on the bar’s area to generate the different values of displayed color on the monitor. This task’s purpose is to check the “Color” selection bar’s functionality along with verifying that the Android Application is processing the change in the display’s color values correctly with respect to the user’s changes.

5.1.2 BeagleBone Black and MTK MT3339 GPS Module PCB Testing

The BeagleBone Black and MTK MT3339 GPS Module chipset provides the Heads-Up Display system with the necessary computing power, Android OS environment, user interface, and video output to generate the entirety of the Heads-Up Display system. Therefore it’s only natural that this PCB chipset be thoroughly tested for quality assurance and information accuracy. The testing modules will consist of both software testing and electric signal measurements to guarantee that all features of the PCB chipset are working as intended.
5.1.2.1 Android OS Environment Testing
The testing on the Android OS Environment will take place in two of the designed Heads-Up Display prototypes using the HDMI 4 PI: 5” display to perform visual accuracy checks. The first platform that will be used for the Android OS Environment Testing is the BeagleBone Black and MTK MT3339 GPS PCB, which will be able to perform most testing procedures listed in this section. The other platform that will be used in later function testing is the BeagleBone Black and MTK MT3339 GPS PCB with Motor Controller connections.

The first Android OS Environment Testing item will have the test analyst power on the current prototype being tested and observe if the system boots into the Android OS without any errors. This test is appraising the prototype environment’s ability to house a stable Android OS for the navigation and cart status software to run properly. The next task on the Android OS Environment Testing list will require the test arbiter to navigate to the UCF EzNav Android Application using the BeagleBone Black’s output display and then execute it. From within the application the tester will click on the “Navigate” button on the main menu and then interact with the navigational map interface to examine whether it correctly tracks the prototype’s current position. This test will ensure that the Android environment’s UCF EzNav Android Application is able to receive GPS data through the BeagleBone Black from the MTK MT3339 GPS chipset on the PCB. The last test in the Android OS Environment Testing section can only be performed on the last prototype when the BeagleBone Black is able to send and receive information to and from the Motor Controller chipset. This test again directs the user to the UCF EzNav Android Application loaded on the Android OS environment just like in the previous test procedure except this time the user will click on the “Cart Status” button at the application’s main menu. At the “Cart Status” screen the evaluator will observe that all data values are being displayed and are within a specified accuracy. This test’s results will confirm whether or not the Android OS is correctly receiving operating values from the Motor Controller as well as providing error-free hand-off of received data values to the UCF EzNav Android Application.

5.1.2.2 PCB Electrical Connection Testing
The PCB Electrical Connection Testing operations will be performed on two of the Heads-Up Display design prototypes using a voltage measurement instrument to perform voltage checks on system’s electrical connections. The first test platform to be used in the PCB Electrical Connection Testing is the BeagleBone Black and MTK MT3339 GPS PCB, the other being the BeagleBone Black and MTK MT3339 GPS PCB with Motor Controller connections.
The first evaluative task for the PCB Electrical Connection Testing can be performed on either prototype platform and has the test operations inspector place the voltage measurement instrument individually on the two VDD_5V electrical connections on the BeagleBone Black as well as on the VCC connection on the MTK MT3339 GPS module. Once in place the examiner will record the received voltage on each connection when the PCB elements are receiving power from a power source. These recorded results from the test provide the data necessary to determine if the BeagleBone Black and the MTK MT3339 GPS chipset are receiving the required operational power on the PCB. The next task in the PCB Electrical Connection Testing series can also be completed on either prototype platform and again requires the test operations inspector to use a voltage measurement instrument. This time the test will have the inspector individually measure the voltage across the TX connection as well as the RX connection from the MTK MT3339 GPS chipset and then record the values for each connection. The results of this test procedure will verify that the MTK MT3339 GPS chipset is transmitting GPS data signals to the BeagleBone Black module. The final PCB Electrical Connection Testing component is to be performed only on the BeagleBone Black and MTK MT3339 GPS PCB with Motor Controller prototype and once more involves using a voltage measurement instrument. In this task the test’s analyst will measure the voltage across the individual wire wrap connections between the BeagleBone Black module and the Motor Controller chipset, and then record each result for reporting purposes. These results will conclude whether or not the BeagleBone Black and Motor Controller modules are correctly sending electrical signals both to and from each other through the wired connections.

5.1.2.3 Output Display Testing

The Output Display Testing will be completed using two of the Heads-Up Display prototype designs with the addition of the HDMI 4 PI: 5” output display. The first test prototype platform to be used in the Output Display Testing will be the BeagleBone Black and MTK MT3339 GPS PCB connected to the HDMI 4 PI: 5” display. The second prototype platform to be used will be the BeagleBone Black and MTK MT3339 GPS PCB with Motor Controller connections outputting video data to the HDMI 4 PI: 5” display.

The first pieces in the Output Display Testing addendum have the tester power on the HDMI 4 PI display in conjunction with the BeagleBone Black/MTK MT3339 GPS PCB and then observe if the display powers up along with properly displaying the BeagleBone Black’s HDMI video output signal. This test’s result will determine if the HDMI 4 PI: 5” display is receiving adequate power levels as well as if it is correctly receiving and displaying video output signal from the BeagleBone Black’s HDMI connection. The second test item in the Output Display Testing section will direct the testing analyst to press the display screen on various elements of the Android OS and Android Application along with recording the resulting action of each screen element pressed. The recorded behavior of this test will state the functionality and accuracy of the Heads-Up Display system’s touchscreen input signals sent to the BeagleBone Black through USB connection.
5.2 Motor Controller

5.2.1 Power Board

The first test to be done on the power board is a continuity check. Each resistance value below should be verified in order to make sure that the board received was made as specified in the circuit. The continuity checks found in Table 14 should be made with at least one capacitor, one MOSFET, and one diode in place. This will guarantee that nothing is shorted together where it shouldn’t be.

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Expected Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>B+</td>
<td>B-</td>
<td>&gt;3MΩ</td>
</tr>
<tr>
<td>B+</td>
<td>M-</td>
<td>&gt;3MΩ</td>
</tr>
<tr>
<td>B-</td>
<td>M-</td>
<td>&gt;3MΩ</td>
</tr>
</tbody>
</table>

Table 14 Motor Controller Continuity Checks

5.2.2 Motor

Initial testing will be done with a small scale DC motor. This will verify that the logic from the controller is working as intended. A probe will be used to measure how much current is passing through the small scale motor and the values in software will be adjusted according to the scaled up values. Mabuchi #RF-280RA 02142 will be used for this. An image of this motor is shown in Figure 67.

![Test Motor](Reprinted With Permission from All Electronics Corps)

5.2.3 MCU

The test procedure for the motor controller will involve extensive testing of the microcontroller and its logic. The motor from section 5.2.2 will be connected to the power board as if it was the real motor. A potentiometer will then feed a voltage to the microcontroller ADC varying from 0-3.3V. Note that in the real motor, the voltage range does not vary this much, but is purely for testing purposes that it is practical to check the entire range of the ADC. This voltage being supplied to the ADC will act like the pedal...
input and the MCU will send out a varying PWM signal to the gate of the MOSFET that it is attached to and spin the motor.

5.2.3.1 ADC
Serial communications will be used to test the ADC values read in from the potentiometer. Energia supports built in UART communications within the environment that can be used to read in values output through the USB debug UART. As the voltage coming from the potentiometer varies from 0-3.3V the values output to the UART will vary from 0-4096. A Digital Multimeter (DMM) will be used to measure the output from the potentiometer to make sure that the serial values output correspond accurately to the physical analog value.

5.2.3.2 PWM
The PWM output from the microcontroller needs to be verified that the pulse is varying correctly according to the value in software. As the PWM output signal varies from 0 to 255, the signal should change from 0V to 3.3V varying in duty cycle and not amplitude. This test will require an oscilloscope to check the duty cycle and reading in the value corresponding to that PWM signal through the same UART method as described in section 5.2.3.1.

5.2.3.3 Hibernation
Testing the hibernation will require an external power supply and a method to control the hibernation state through the UART. The power dissipated by the microcontroller will be measured before and after being put into sleep mode. Calibrating this will also be able to give the HUD an accurate estimate of the power usage.

5.2.3.4 Potentiometer
The potentiometer must be easily turned by hand and have external prongs that make it easy to read with an oscilloscope so that the output from the potentiometer can easily be compared to the PWM output from the microcontroller. This is why Tokos / Cosmos# RVQ24YN15FB103 was chosen. The ends are easy to probe and wires can be soldered to them for easy connection to the TivaC. A picture of this part shown in Figure 68.

5.2.3.5 Scope
A Tektronix DPO4034B will be used in order to read the outputs from both the potentiometer and the microcontroller. It has four channels which is two more than necessary and a 350MHz bandwidth. A maximum frequency of 120MHz would need to be measured so this is fine. This is the oscilloscope that is available in the senior design lab and other electrical engineering labs on campus.
5.2.4 Test Procedure
The following list of instructions describes in detail how to test the motor controller without actually connecting it to the golf cart motor.

1. Connect one MOSFET, one capacitor, and one diode to the power board PCB.
2. Connect one power terminal of the potentiometer to a 3.3V power supply and the other to ground.
3. Connect the signal pin to the ADC channel being used on the TivaC.
4. Connect the PWM pin of the TivaC to the gate of the MOSFET.
5. Connect one channel of the scope to read the output of the potentiometer and another to read the PWM signal.
6. Vary the potentiometer from 0 to 3.3V and check the output of the PWM signal on the scope.
7. Verify that as the voltage increases the PWM signal increases as well, but not instantly. The increase in PWM should be gradual not instant.
8. Verify that the motor’s speed varies corresponding to the PWM signal.

This test is a combination of verification of the hardware and software. When the motor’s speed gradually changes with the PWM output, this verifies that the hardware is working and nothing is shorted where it shouldn’t be. The oscilloscope verification of the PWM vs. the voltage verifies that the software in the TivaC is working.

5.3 Autonomous Features
5.3.1 Golf Cart Speed and Braking
First, in order to determine the correct rangefinder sensors to use for this project, the golf cart’s maximum speed and braking distance must be determined. By using a speed sensor, we can determine the maximum speed of the golf cart. Implementing a geartooth RPM sensor on the axel will measure the RPM and from there, the velocity can be determined.

Next, the braking distance will be needed in order to determine threshold distances and velocities. From top speed, the golf cart’s brakes will be applied to bring it to a rest. Using the geartooth RPM sensor, the time from full speed to zero velocity can be recorded. With this value and the axle revolutions, the stopping distance can be recorded.
5.3.2 Rangefinder Sensors

The rangefinder distance sensors used will first be tested to make sure the accuracy of the output. Connecting a simple display to the microcontroller as an output will allow for the distances to be displayed. By configuring the correct formula to convert the signal to a measureable distance, the most accurate configuration can be set.

Next, the sensors can be mounted on the vehicle. Tests can be run with the vehicle stationary to ensure the sensors are correctly detecting objects. The next step would be to repeat this step with the vehicle in motion. By varying speeds, the sensors can be tested in multiple conditions. Once this is fine-tuned, the microcontroller can be connected to the motor controller to send signals to control the speed of the golf cart. This can then be tested with a simple code that slows the golf cart when there is an object within an arbitrary distance. The golf cart should reduce speed when an object is within the given distance. Next the threshold distances and velocities can be calculated.

5.3.3 Threshold Distances and Velocities

Using the stopping distance as the maximum threshold distance will allow for the golf cart to come to a stop without colliding with an object. First, this threshold distance will be set and when the object is within this range, the golf cart’s speed will be reduced to a stop. Once this is confirmed, multiple threshold distances can be used in order for the golf cart to reduce speed at a constant rate instead of an abrupt stop. For example, for the furthest threshold distance, the golf cart’s velocity may be reduced to 80% of the current speed. Once the object reaches the next threshold distance, the velocity can be reduced by an additional 80%, and so on. This will allow for a smooth driving experience and avoid collisions.

Multiple tests will be implemented using trial and error. Starting at a low velocity, the golf cart will drive toward an object. Once the object is detected in the threshold distances, the velocity of the golf cart should be reduced accordingly. These tests will be repeated for various speeds and object sizes. Should there be an issue with stopping in time, the threshold distances can be increased to allow for a longer distance for the golf cart to come to a rest.
5.3.4 Cruise Control
Once the cruise control is set up, it can be tested by running it at different speeds. Once the desired speed is achieved, the button will be pressed and cruise control should engage. At the moment the cruise control button is pressed, the current velocity will be maintained. While cruise control is engaged, the button will be pressed to disengage cruise control. Also, the brake pedal will be fitted with a button, so that when the brake is pressed while cruise control is activated, the cruise control will disengage.

5.3.5 Driver Notification
The auditory driver notification will be used with the cruise control and object avoidance to alert the driver if there is an object in the current path and when cruise control is switched off. This can be tested by activating cruise control and then pressing the brake. The brake pedal will shut off the cruise control and simultaneously alert the driver. Also, by driving toward an object, the auditory driver notification device can be tested. Once an object is within the threshold distance, a sound should emit from the device to alert the driver.

5.4 Charge Controller
The testing for the charge control will follow the same phases as the prototyping. The phase will be as follows. Phase one will deal with tests implemented once the preliminary circuit design is done. Phase two will require a test or prototype board to go through certain lab tests. The last phase is testing done once the charge controller is finished and completely connected to the solar panels and battery bank.

5.4.1 Computer Simulation
Once the circuit design in close to completion it can be designed into a computer software like Multisim. In Multisim we are able to run a few different tests. The first test will be to connect a virtual oscilloscope to the circuit and test the input and output waveforms of the circuit. By running this test we are able to determine if our design is running in an efficient manner. This data will be recorded in order to compare with lab tests done in the second phase of testing. Another test we will run is an AC analysis of the circuit. From this simulation we will be able to see the magnitude and phase plots. This is important to analyze in order to make sure the filters are function properly.
5.4.2 Lab Tests
After the computer simulations the test circuit can be built to test in a lab environment. This circuit will be connected to a function generator and power supply to simulate different inputs in regard to what it might experience from the actual array. The output of the circuit will be connected to an oscilloscope to run a couple tests. The first test will be to run the input and output waveforms. These results will be compared with those from the computer simulation and checked for accuracy. From these output waveforms we will be able to analyze how the nodes switch due to the switching mechanism in the charge controller. Figure 70 shows an example of such waveforms from TI’s test on the TIDA device. Next we will run the oscilloscope to output the frequency response and compare with the computer simulation results.

![Switching Node Waveform of 12V system](image)

Figure 78: Switching Node Waveform of 12V system
(Reprinted with Permission Pending from TI)

5.4.3 System Tests
The last sets of tests that need to be completed will be done once all other tests are done and the circuit has been finalized. The final controller design will be connected to the system. The input will come from solar panels and the output will go to the battery bank. The test will consist of making sure that the device is working effectively and efficiently. This will conclude the testing for the device. Any last changes will be made and the controller will be installed and mounted as needed.
6 Appendix
6.1 Reprint Permission Emails
6.1.1 Technicians Manual
Status: Pending
It is from the 1997 TXT electric golf cart technicians' manual. It is a copyrighted picture that I would like to use for my own documentation for the golf cart modification that I am working on. See the picture below.

Thanks,
Jake Bettis

From: Blue, Poppie
Sent: Monday, October 13, 2014 12:42 PM
To: jake.bettis@knights.ucf.edu
Cc: Shopezgo

Mr. Bettis, can you tell me exactly what you are wanting to use? Is it something that’s on our website? Please let me know so I can understand what exactly you are wanting to do.
Thank you!
Poppie Blue
Ecommerce Manager, P&A
(p) 706.771.4617
1451 Marvin Griffin Road
Augusta, GA 30906

6.1.2 Open Revolt
Status: Approved
Hello,

I am planning on using your open source power board for a school project and would like to use a picture of the PCB and schematic from your site in my documentation for that project. Is this alright?

Thanks,
Jake Bettis

6.1.3 Google Maps API License

Status: Approved

6.1.4 Test Motor

Hello,

I am documenting my test procedure for a school project and plan to use a few of your products. We are required to obtain rights to use images from copyrighted websites. May I use the image from your website in the document? One of the parts that I plan to use is:

http://www.allelectronics.com/make-a-store/item/dcm-406/1.5-6-vdc-motor/1.html

Thanks,
Jake Bettis

6.1.5 GNU Free Documentation License

Material licensed under the current version of the license can be used for any purpose, as long as the use meets certain conditions.

- All previous authors of the work must be attributed.
- All changes to the work must be logged.
- All derivative works must be licensed under the same license.
- The full text of the license, unmodified invariant sections as defined by the author if any, and any other added warranty disclaimers (such as a general disclaimer alerting readers that the
document may not be accurate for example) and copyright notices from previous versions must be maintained.

- Technical measures such as DRM may not be used to control or obstruct distribution or editing of the document.

6.1.6 AutoSales inc.
You are hereby authorized to view, copy, print, and distribute these materials subject to the following conditions:
1. The materials may be used for internal informational purposes only;
2. Any copy of these materials or any portion thereof must include the above copyright notice; and
3. Autosales, Incorporated may revoke or modify any of the foregoing rights at any time.

6.1.7 All Electronics Inc.
Hello Jake
Yes you may use images from our web site for you school project.
Please consider this email as your permission.
Best of luck

Woolf Kanter
All Electronics Corp.
www.allelectronics.com
800-826-5432 - fax 818-781-6847

From: jake bettis [mailto:jbettis17@hotmail.com]
Sent: Monday, December 01, 2014 7:04 PM
To: Mail User
Subject: Access to images from website
Hello,

I am documenting my test procedure for a school project and plan to use a few of your products. We are required to obtain rights to use images from copyrighted websites. May I use the image from your website in the document? One of the parts that I plan to use is:

http://www.allelectronics.com/make-a-store/item/dcm-406/1.5-6-vdc-motor/1.html

Thanks,
Jake Bettis

6.1.8 Sensor
From: Mathew A. Dirjish [mailto:mdirjish@questex.com]
Sent: Tuesday, December 2, 2014 4:51 PM
To: Matt Tourtelot
Subject: RE: Permission to use picture
No problem. I would suggest that you note that it came from the article:

Motion/Velocity/Displacement
Advances in Analog Distance Sensing
By: Brian Duval
September 1, 2004

Thanks,

**Mat Dirjish**  
Executive Editor – Sensors Magazine

**sensors**
718-793-5501 | mdirjish@questex.com | http://www.sensorsmag.com

**Subject:** Permission to use picture

Hello,

I was wondering if I could include the image in Figure 6 from the article, http://www.sensorsmag.com/sensors/motion-velocity-displacement/advances-analog-distance-sensing-816 in my Senior Design Project for my University.

Thank you,
Matt Tourtelot

**6.1.9 Laser diagram**

From: Philo [mailto:philo@philohome.com]
Sent: Wednesday, December 3, 2014 2:41 AM
To: Matt Tourtelot
Subject: Re: Permission to use image

Hi Matt,

Sure, you're welcome to do so ;)

Philo

Matt Tourtelot a écrit :
> Hello,
> 
> I was wondering if I could include the laser diagram from the page, http://www.philohome.com/sensors/lasersensor.htm, for use in my Senior Design Project at my University.
Thank you,
Matt Tourtelot

6.1.10 Battery Configurations
Status: Approved
Matthew Roland <rolandmatt727@gmail.com>
8:47 AM (9 hours ago)
Hello,

My name is Matt Roland and I am a senior Electrical Engineering student at the University of Central Florida. I am a part of a project for my senior design class that is building a solar powered golf cart. I used some of the photos from your website to describe battery configurations for our battery bank. I would like to request official permission to use these photos. Thank you for your time and response.

Regards,
Matt Roland

Ray Walters <ray@solarray.com>
10:34 AM (8 hours ago)

You have official permission to use our info, as long as it is for educational purposes only.
Also, we hid this site, but here is info on our solar EVs we've built:
www.electromoto.com

R. Ray Walters
CTO, Solarray, Inc
Nabcep Certified PV Installer,
Licensed Master Electrician
Solar Design Engineer
303 505-8760

6.1.11 Battery Charts and Specs
Status: Pending
Matthew Roland <rolandmatt727@gmail.com>
8:53 AM (9 hours ago)
Hello,

My name is Matt Roland. I am a senior at the University of Central Florida, studying Electrical Engineering. I am doing a senior design project which is to design and build a solar powered golf cart. We have not purchased yet, but plan on using some deep cycle
batteries from your company. The batteries are the US 2200 XC2. I have used some of the graphs and specs as technical photos in our documentation and would like to request official use of these photos. Thank you for your time and response.

Regards,
Matt Roland

6.1.12 Texas Instruments Charts, Tables, and Figures
There is no email address for the general public to use in order to get in contact with TI. All requests must be submitted through their website in a request form. A form was sent on 12/2/2014, but there has been no response. We will assume permission is pending.

6.1.13 Grape Solar PV Diagrams
There is no direct email address for the general public to use in order to get in contact with Grape Solar. All requests must be submitted through their website in a request form. A form was sent on 12/2/2014, but there has been no response. We will assume permission is pending.

6.1.14 Solar Wiring Diagram
Status: Pending
Matthew Roland <rolandmatt727@gmail.com>
6:58 PM (0 minutes ago)
to rob@solarjourneyusa.com
Hello,

I am a student at the University of Central Florida. I would like to request permission to use an image from your site in my senior design project.

Thanks,
Matt Roland

6.1.15 Panasonic
No direct email address is given. A request was sent on 12/2/2013 requesting permission to use their photos for battery charging.

6.2 Software
A. Eclipse IDE Java development platform
B. Android SDK Manager
C. EAGLE 7.1.0 Freeware
D. Energia
6.3 Parts List

6.3.1 HUD
   A. BeagleBone Black microcomputer module
   B. Adafruit MTK MT3339 Ultimate GPS Breakout module
   C. Adafruit HDMI 4 Pi: 5” Display with HDMI input and touchscreen PCB module

6.3.2 Motor Controller
   A. IRFP7718PbF
   B. 381LR821M200J042
   C. STPS80150CW
   D. Power Board
   E. TivaC 129XL