

Solar Powered Golf Cart

Jake Betis, Jacob Krueger, Matt Roland, and
Matthew Tourtelot

College of Electrical Engineering and
Computer Science, University of Central
Florida, Orlando, Florida, 32816-2450

Abstract – This project incorporates a more advanced and efficient golf cart. The golf cart includes three different drive modes: standard, full performance and energy efficient. The full performance mode allows for maximum velocity and full control of instrumentation. Energy efficient mode limits power drawn from motor and instruments to increase vehicle run time. The golf cart utilizes two roof-mounted solar panel arrays as well as an optional wall charger for power input. Included in the golf cart is a touch screen display used to select drive mode, utilize GPS map, and monitor the vehicle's diagnostics.

Index Terms – Solar Power, Charge Controller, Maximum Power Point Tracking, and Motor Controller

I. INTRODUCTION

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 solar powered golf cart project. This project allows our group to focus on different areas such as solar panel battery charge controllers, golf cart motor controllers, and a visual display 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. The touch screen display will give the user many options such as efficiency mode, navigational assistance, and status of the golf cart.

The first component of this project that users will see is the touch screen display. 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 touch screen display 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 touch screen display 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.

II. TOUCHSCREEN DISPLAY

The idea for a touchscreen display arose while contemplating the original high-level design of the cart's systems at the beginning of the project process. The idea was proposed as a means of providing an interface between the cart systems and the cart's users. By creating a touch screen display system the cart would be capable of sending user input to the cart's systems and providing feedback of the cart's systems to the user during use.

The cart's Touchscreen 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 resulted in a testable Touchscreen Display prototype. The following sections will elaborate upon each step of the Touchscreen Display system's development process.

The software for the Touchscreen Display system is loaded on the chosen microcomputer in the form of an Android Application running on the Android 4.4 KitKat mobile environment. This application consists of four classes to accomplish the requirements 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 1.

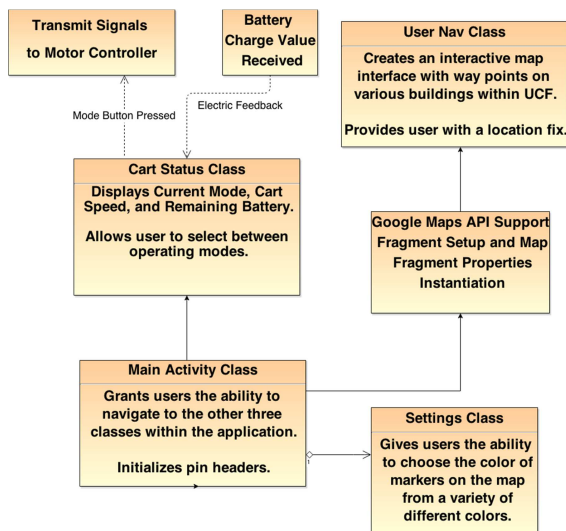


Figure 1: Android Application Class Diagram with Included Features

The navigation page was implemented via its own class called UserNav.java and makes 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’s 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.

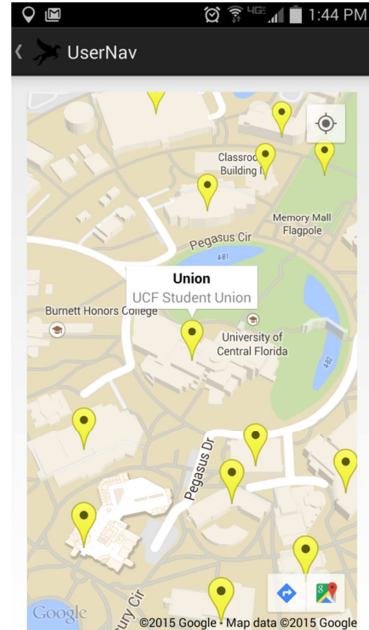


Figure 2: UserNav Activity Screen

The Cart Status Screen includes 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 mode selection are detailed in the state diagram of Figure 3.

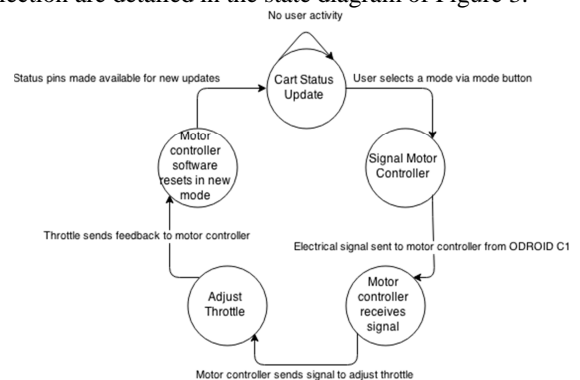


Figure 3: CartStatus.java Communication and Status Update State Diagram

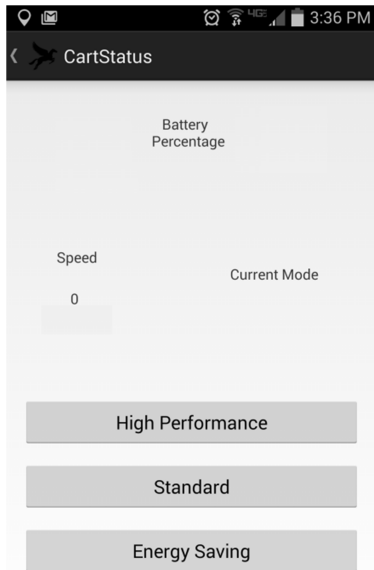


Figure 4: Cart Status Activity Screen

From the Cart Status Activity Screen the user is able to view relevant data pertaining to the cart's operation. The Battery Percentage field will display a percentage of the available battery charge remaining to the cart. The Speed field will display the cart's current velocity in m/s^2 . The Current Mode field will display the current operating mode of the cart's systems.

An additional function of the Cart Status Activity Screen is the mode selection buttons. When the user selects one of the three modes by pressing the respective interface button the Android Application will change the values of the three output GPIO pin headers on the ODROID C1. Once the pin values are updated the Motor Controller will read the input values and switch the cart to the appropriate operating mode requested by the user. This process is detailed in Figure 3 of the previous page.

In order to meet the specified requirements of running an Android Application and displaying pertinent information to the cart's users, the Touchscreen Display needed the proper hardware. For the Touchscreen Display system to run a navigational Android Application it requires 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 the requirement of having a GPS measurement instrument, a specialized PCB was designed and built to communicate with the Android OS running on the selected microcomputer.

The display used for video output needed to have an HDMI input as well as a straightforward means of providing touch screen capabilities. Thus, the HDMI 4 Pi: 5" Display was the display system chosen for the display system design. The HDMI 4 Pi 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 is connected to the ODROID C1 via the HDMI connection for video input and connected to the cart's onboard power system through the provided power cable to supply the display with the necessary 5V, .5A required to power the display and the touchscreen driver. The touchscreen adapter PCB has a USB miniB output and a 40pin to 50pin adapter which is connected to the ODROID C1 through the USB miniB output to the ODROID C1's USB A Standard input in order to transmit the user's touchscreen input to the Android Application on the ODROID C1. The wired PCB with the five buttons to control a menu system for the display is not used and was removed from the PCB in the final cart design since all display setting changes will be done through the Android OS.

The microcomputer selected for the Touch Screen Display system is the lynch pin of the entire Touch Screen Display system. It needed to be able to host an OS with OpenGL ES 2.0 graphics enabled in order to generate the Google Maps API fragment as well as have the necessary GPIO, UART, and ADC pin headers in order to effectively communicate with the other modules within the designed cart system. Keeping these requirements in mind it was decided that Hardkernel's ODROID C1 microcomputer development board would be used within the Touch Screen Display system design. The ODROID C1 shipped with a detachable Android 4.4 KitKat eMMC module with pre-loaded WiringPi libraries in order to easily interface with the 40 I/O pin headers located on the board throughout development. It also had the OpenGL ES 2.0 resources pre-installed so that the navigational map interface could be properly loaded within the Android Application. The ODROID C1 is powered through a voltage regulator connected to the cart's battery bank that supplies the ODROID C1 with the necessary 5V, 2A for normal operation. The ODROID C1 will be connected to three other devices through the 40 pin I/O headers, these devices are the GPS module PCB, the Motor Controller, and the battery capacity measurement circuit. The pin headers used for the connection with the GPS module PCB are pin #1 which is the 3.3V power pin header and pins #8 and #10 which are TX and RX respectively for the UART connection needed to send GPS coordinate information. The pin headers used in communications

with the Motor Controller are pins #31, #33, and #35 which are GPIO #100, #108, and #97 respectively, one of these pins will be high and the other two low depending on which operating mode is currently set. The pin used in measuring the cart's remaining battery charge is pin #37 which is one of the board's AIN pins that can measure a voltage between 0-1.8V in order to calculate the percentage of battery charge left.

In order to provide the Android Application with GPS data used in generating user location in the UserNav Activity screen along with the cart's speed in the Cart Status Activity screen a GPS Module would need to be integrated into the Touch Screen Display design. Therefore the Ultimate GPS Module from Adafruit Industries based on the MTK3339 Chipset was chosen to be used to gather relevant GPS data for a location fix and cart velocity. The original design plan was to create a custom shield for the BeagleBone Black that would house the MTK3339 GPS module along with providing the necessary power and UART connections for power supply and GPS data transmission. Upon discovering that the BeagleBone Black was not able to load the OpenGL ES 2.0 libraries the microcomputer development platform changed to the ODROID C1. Therefore all connections between the GPS module PCB and the ODROID C1 are wired since the GPS module PCB does not fit correctly on the ODROID C1's pin headers.



Figure 5: MTK3339 GPS Module PCB with headers

The GPS module PCB was designed using EAGLE Layout Editor Software which provided an

environment to create the connection schematic, select appropriate parts to be used in the board design, and generate a board design to send out for fabrication. The PCB design includes a connection to the 3.3V power pin to supply power to the MTK3339 GPS Module and a connection to the pin headers for the TX/RX pins on the MTK3339 GPS Module to communicate GPS data to the ODROID C1. There is another connection to a uFL coaxial connector on the PCB in order to connect the external antenna to the GPS module PCB so that it can receive satellite data while housed in an enclosure within the cart's chassis. The mentioned EAGLE schematic can be viewed in Figure 6.

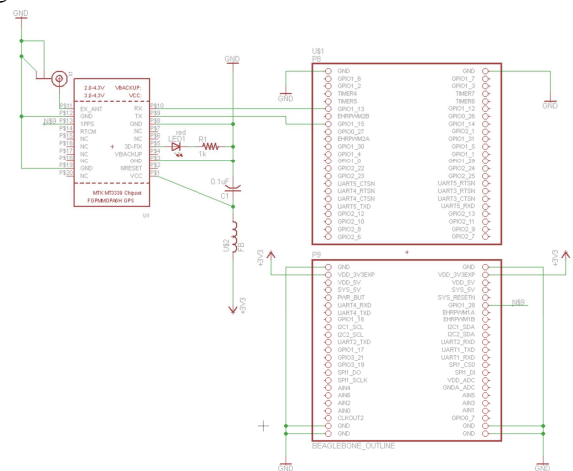


Figure 6: MTK3339 GPS Module PCB Connection Schematic in EAGLE design software

In the development process of the Touch Screen Display System three prototypes were created to test the features of the Android Application. The Touch Screen Display System prototype was a Samsung S4 running Android 4.4 KitKat. Only the UserNav Activity and Settings Menu could be tested on this prototype since there were no I/O Expansions available on the S4. The second prototype consisted of the HDMI 4 Pi touchscreen display, the ODROID C1, and the MTK3339 GPS Module PCB connected together and powered through wall outlets. This prototype can be viewed in Figure 7. The third prototype consisted of all the previous modules enclosed within designed and purchased enclosures, still being powered through wall outlets. The enclosure for the HDMI 4 Pi touchscreen display and the GPS Module PCB were designed in the SketchUp Make 3d modeling software which generated the necessary files to 3d print the enclosures in UCF's TI Innovation Lab. The final design of the Touch Screen Display system is achieved by powering the final prototype through the cart's power supply along with

connecting it to the Motor Controller and battery charge level circuit.

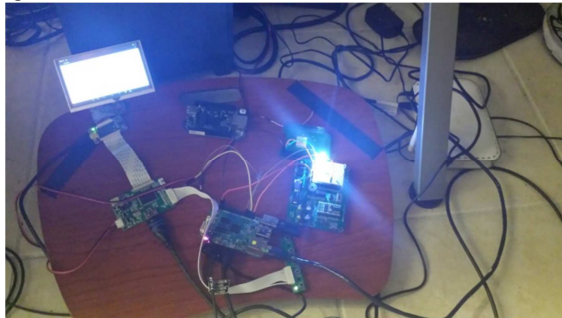


Figure 7: Second Phase of Touch Screen Display prototyping with HDMI 4 Pi monitor, ODR01D C1, GPS Module PCB, and a breadboard for testing the GPIO output used in cart operating mode selection.

III. MOTOR CONTROLLER

The power board of a motor controller contains 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 opens the path from B- to M- and this allows current to flow through the golf cart's direction selector. Figure 8 shows the overall system that the motor controller operates in. The motor controller interfaces with the solenoid for pre-charging, the direction selector, and a pedal telling it how fast to accelerate.

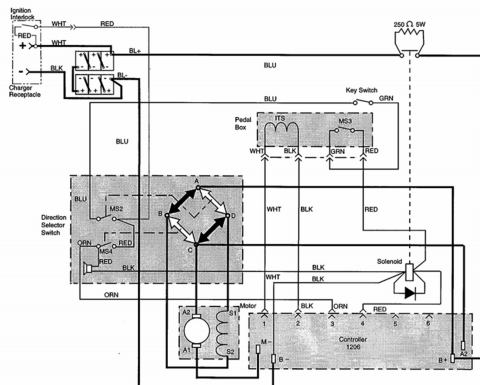


Figure 8: Motor Controller System

The number of MOSFETs was calculated to divide the amount of current evenly without going above a dangerous temperature. The diodes chosen did not need to handle as much current as the MOSFETs because the regenerative braking current is much lower than the initial amount required to get

the motor moving. The power board in this design was prefabricated from a vendor for electric vehicles that already had the spaces needed for each component. This allowed for simple pick and place of each component based on the needs of our project.

A typical golf cart motor runs at 36 or 48 volts so the MOSFETs were chosen to allow for a maximum voltage of 75V in the case that the user wanted to switch to 72V. The MOSFETs chosen had an on-resistance of .00145 ohms which will produce minimal heat during conduction.

The capacitors were left at the same values as in the reference design to provide a robust stabilization of the voltage during regenerative braking. 16 1000uF capacitors were used because this would be more than enough to handle the current fed back from a DC motor of any size.

A large aluminum heat sink was used to dissipate heat through the MOSFETs and diodes. This will allow for passive removal of heat and lessen the strain on each component. In addition to an aluminum heat sink, copper bus bars are used to make good contact with each pad of the power board. The reference schematic [1] of the power board is shown in Figure 9.

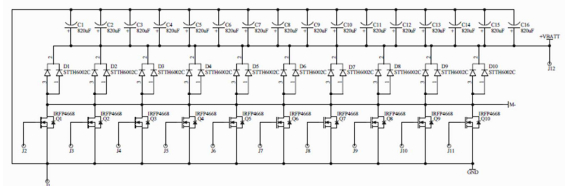


Figure 9: Motor Controller Reference Design

A custom MSP430 board was used to send the PWM signal controlling the power board and interfacing with the touchscreen display microcontroller. The design was done in Eagle for fabrication and part selection. There is a voltage regulator for MCU power, PWM gate driver, GPIO pins for each mode, and a voltage regulator to interface with the pedal. The schematic for the logic board is shown in Figure 10.

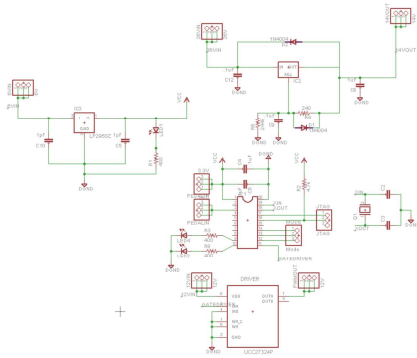


Figure 10: Motor Controller Logic Board

The suggested voltage for the MOSFET gates is 10V so a gate driver was chosen that would be able to handle a minimum of 10V. Each battery in the golf cart is 6V so a driver was chosen that could handle 12V and had enough current to drive 10 gates. The fixed 3.3V regulator drops one of the 6V batteries down in order to power the microcontroller and provide power to the potentiometer pedal. The output from the pedal will vary from 0V to 3.3V and be read into the ADC pin labeled “pedalin.”

Each mode of operation is a ramping software. In each mode, the ramp speed is increasingly quick until it gets to high performance where the controller is an exact match to the pedal input. This software was written in the Energia environment and programmed to the msp430 using a FET from a Launchpad for programming. In this software, calculations are performed at a rate of 10 kHz which allows the PWM plenty of time to not overdrive the capacitive load that it is driving. The test setup for programming the MSP430 board is shown in Figure 11.

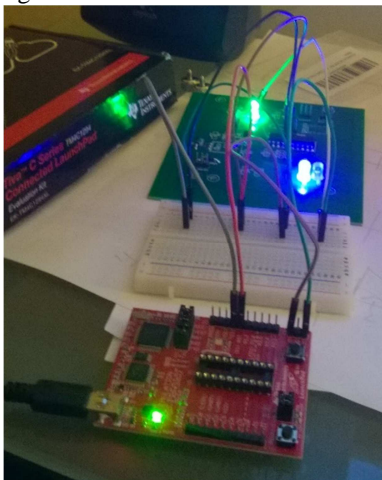


Figure 11: MSP430 Debugging Setup

Prototyping was done using a smaller DC series wound motor to emulate the golf cart’s motor. A potentiometer was used as the pedal and the logic board was emulated using a TivaC Launchpad. The setup is shown in Figure 12. Each mode was tested for a noticeable difference in ramp speed. In this test setup, the direction selector and solenoid were not taken into account.

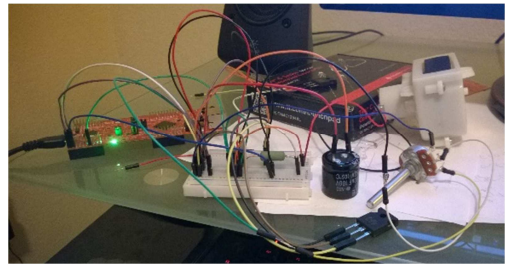


Figure 12: Motor Controller Test Setup

IV. CHARGE CONTROLLER

A solar charge controller is used to safely and effectively charge the golf carts battery bank using the two roof mounted solar panels. This is a vital part to the carts alternative energy design, as it allows for efficient charging using the solar panels.

The cart is equipped with two 250 Watt Grape Solar© high efficiency, mono-crystalline PV modules. Each panel is made up of 60 cells and has an open circuit voltage and current of 37.7 Volts and 8.72 Amps, respectively. Connected in series the panels will provide a nominal open circuit voltage of 75.4 Volts. The battery bank is made up of six, 6 volt flooded lead acid batteries from US Batteries. Connecting these in series provides the 36 Volt battery bank needed to power the golf cart motor and all onboard electronics.

In order to efficiently charge the batteries, the solar charge controller will implement a Maximum Power Point Tracking (MPPT) algorithm. This will allow the charge controller to track the maximum power point for the solar panels depending on sunlight intensity and cell temperature. The MPPT algorithm was programmed using a Perturb and Observe approach. This method adjusts the input voltage from the solar panels in small increments and measures the power; if the power increases, further adjustments are made until the power no longer increases. This method was chosen because of its ease of implementation and efficiency.

The design for the solar charge controller was modeled using the TIDA 00120 Solar MPPT Charge Controller reference design from Texas Instruments [2]. The reference design was modified in order to handle the high voltage and current inputs

and outputs. There are three stages to this design. The power stage, the logic control stage, and the bias supply stage. Figure 13 shows the circuit schematic design using Altium Designer.

brains for the charge controller. It will be performing the majority of the algorithms and precision functions. The MSP430F5132 was chosen because of its low power consumption and wide application range. The

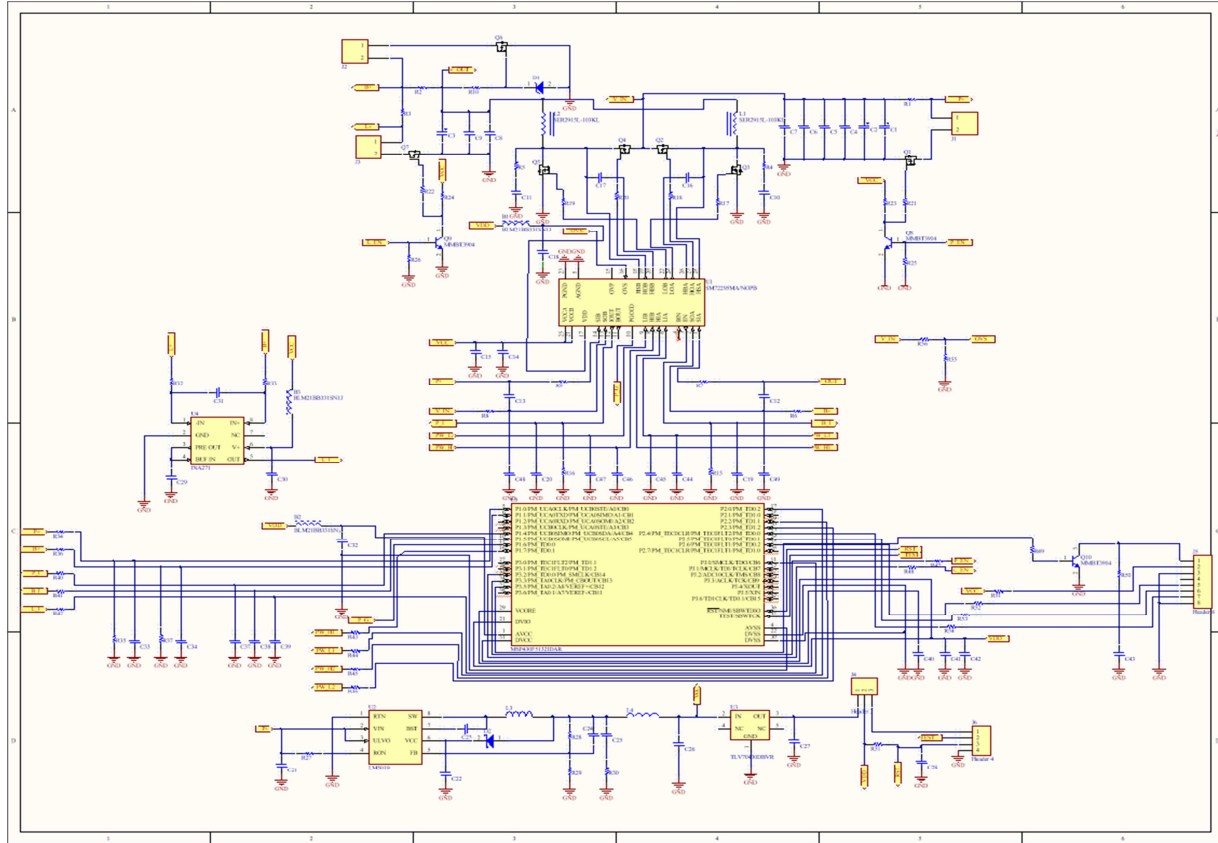


Figure 13: Solar Charge Controller Schematic

The power stage of the design uses a SM72295 photovoltaic full bridge driver which is responsible for driving DC/DC conversion. It is designed to drive 4 discrete N type MOSFET's in a full bridge configuration. The drivers provide 3 A of peak current for fast efficient switching. Current sensing is provided by 2 trans-conductance 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 that provide a low impedance interface to an A/D converter. Under voltage lockout is enabled using a PGOOD indicator which prevents the drivers from operating if VCC is too low. The switching MOSFETs will be changed to 100V rated parts to allow for a higher input voltage. In order to increase the max current rating to 40A the MOSFETs will be switched from a SOP package used in the reference design to a TO-220 package version.

The logic portion is designed using a MSP430F5132 microcontroller. This is an ultra-low power mixed signal microcontroller and acts as the

device features a powerful 16-bit reduced instruction set computing (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 μ s. The MSP430F5132 has two 16-bit high-resolution timers, two universal serial communication interfaces, a 32-bit hardware multiplier, a high-performance 10-bit analog-to-digital converter (ADC) and an on-chip comparator. The MPPT algorithm was programmed using code composer studio (CCS) and a MSP430 launch pad that is used to make a connection with the microcontroller. The reference design provided a GUI interface that allows for easy application use and to set certain parameters.

The next stage is the bias supply stage. This stage consists of a 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. Combined in an output format, these three devices provide the bias supply which can be connected to a load.

V. CONCLUSION

This project provided our group with a wide variety of technical experience in the fields of solar power, electronics, and software.

REFERENCE DESIGNS

Motor Controller Power Board

[1] Open Revolt Wiki Projects: 1000 Amp DC Motor Controller

http://ecomodder.com/wiki/index.php/Open_ReVolt

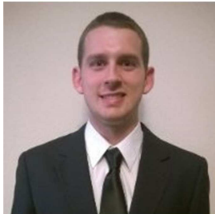
Solar Charge Controller

[2] MPPT Solar Charge Controller: TIDA00120

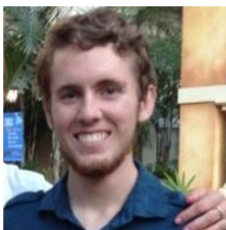
<http://www.ti.com/tool/tida-00120>

ACKNOWLEDGEMENT

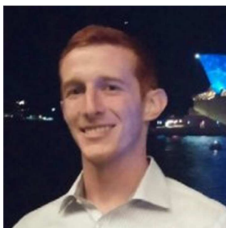
BIOGRAPHIES



Jake Bettis is currently a senior studying Electrical Engineering at the University of Central Florida. He has already accepted a position for after graduation in May 2015.



Jacob Krueger is currently a senior at University of Central Florida studying Computer Engineering. He graduates in May 2015 and plans to pursue a career in the computer science profession.



Matthew Tourtelot is currently a senior at University of Central Florida studying Electrical Engineering. Matthew plans to attend graduate school at

UCF to earn his Master's in Electrical Engineering.



Matthew Roland is currently a senior at University of Central Florida studying Electrical Engineering. He is graduating in May 2015 and plans to pursue a career in the electrical or electronics engineering profession.