High Capacity Battery Solar Powered System (HCBSPS)

Josiah Best, Mike Howell, Eduard Meighan, Jason Rodriguez

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

***Abstract* — A high capacity battery that is solar charged. This device is the solution to individual solar energy consumption on a small scale. The HCBSPS project employs a LCD touchscreen that displays statistical information for the design and stores them for up to seven days. These processes are implemented using an Adafruit METRO M0 Express, which uses the ATSAMD21G18 microcontroller. This system allows for small appliances like personal fans, phones, and other items to be charged and/or used when other methods of power are unavailable.**

***Index Terms* — Battery Management Systems, Graphical User Interfaces, Inverters, Microcontrollers, Photovoltaic Systems, Pulse Width Modulation, and Solar Panels.**

I. INTRODUCTION

This design was created to be a cost effective and user friendly solution to individual solar power. One of the objectives that we set out to accomplish with our design was semi-portability. By being somewhat portable, this device can be moved to maximize optimal sunlight and decrease the time to charge. This project is designed around the functionality of the Adafruit METRO M0 Express along with the custom battery enclosure, the battery management system, the LCD Display, the custom charge controller, and the lithium iron phosphate battery cells. Lithium iron phosphate was chosen as the chemical make-up of the battery due to its safety to the average consumer compared to other chemical compositions like lead acid, nickel cadmium, and nickel metal hydride. In specific, lithium iron phosphate runs a near zero percent chance of thermal runaway. Thermal runaway is the process of increased energy causing an increase in heat, which will loop until the battery melts down or explodes. While this project has been created before in various forms, what sets this design apart is user friendliness of the solar charge controller with the LCD touchscreen and the semi-portability of the device.

II. SYSTEM COMPONENTS

The High Capacity Battery Solar Powered System can be divided down to the individual components that make up the system. Below you will see a complete diagram of the design, which will include arrows showing data, power, and power out for each component.

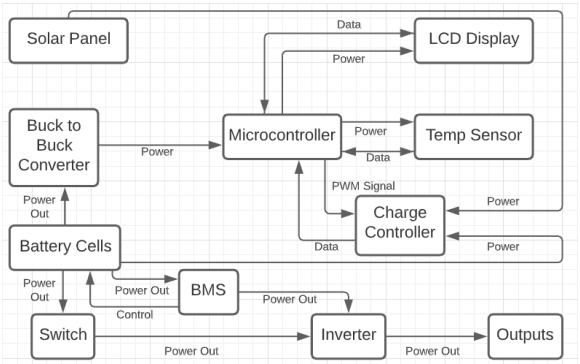

Figure 1. General Hardware Flowchart

A. Adafruit METRO M0 Express (ATSAMD21G18)

The Adafruit METRO M0 Express is used as the main information device for the design. This board allows for power and data to move through multiple components of the device. The microcontroller will receive power from the DC to DC regulator, and will supply power to the LCD Display and temperature sensor. It will receive data from the solar panel, LCD display, temperature sensor, charge controller, and the battery cells. It will also provide data to the charge controller, and the LCD display. One of the most important functions that it has is supplying the PWM signal to the charge controller. On board, there are 3.3V and 5V regulators that can be used to power external peripherals. In addition, there's also a 2 MB SPI flash memory chip that is used for data logging and initializing certain variables and data structures in our software. In addition, the microcontroller also features an internal Real-Time-Clock (RTC), which is important for graphs or functions that rely on time.

B. HEYO Smart BMS F4S12V60A BT

The HEYO Smart BMS F4S12V60A BT is the battery management system in this design. This component is used to manage the health of the battery. While the health of the battery is the main focus of the BMS it has many functions vital to the continued success of the battery. These functions are: monitoring, recording, and calculating the data, balancing the battery cells, and keeping it at the

appropriate temperature [1]. A battery management system is a requirement of any major device that uses a battery composed of multiple cells. For other chemical compositions in batteries, it can help prevent thermal runaway. However, in our system, thermal runaway is essentially a non-existent worry.

C. Renogy 100W 12V Solar Panel

The Renogy 100 watt 12 volt solar panel is a monocrystalline foldable device. These types of solar panels are widely used in solar power applications. This is due to several factors like a uniform appearance and high efficiency. This solar panel was specifically chosen for this design due to being donated to the group. This solar panel has a maximum operating voltage of 18 volts and a maximum operating current of 2.78 amps. The weight of the solar panel is within range of keeping the entire device semi-portable and weighs approximately 27.1 pounds. Additionally, the solar panel has a 25 year efficiency rating which will hold above 80% for that full 25 year period [2].

D. LCD Display

The LCD display is a 2.8" TFT capacitive touchscreen [3]. It comes with a full color 320x240 display and four white LEDs set in parallel for backlighting. The primary viewing direction of this display is at the six o'clock direction, and at this direction it will have the maximum contrast and readability. Its size is comparable to many charge controllers with screens on the market and, in addition, there's a breakout board that also has an SD card reader. The SD card reader can be used for read and write operations. For this project, the user has the ability to store and view data from a card inserted into the reader. There's a pin on the breakout board called "Lite." A PWM signal can drive this pin and adjust the LCD's brightness. In our software, the user can adjust the brightness in 10% increments or decrements.

E. MCP9808 Temperature Sensor

The temperature sensor allows the microcontroller to measure the ambient temperature. It has an operating temperature between -40°C and +125°C with a typical accuracy within ±0.25°C. Its maximum current draw is 5μA and it can convert 0.25°C/bit in 65 ms at 15 samples/sec. The sensor supports higher precision, upto ±0.0625°C, but with slower conversion time. A breakout board is used to interface with the temperature sensor and I2C is used to communicate with the sensor.

F. Lithium Iron Phosphate Battery Cells

The lithium iron phosphate battery cells that were chosen are the VariCore 3.2V 200Ah cells. A total of four cells are placed in series to increase the nominal voltage to 12.8V with a maximum charging voltage of 14.6V. These were chosen due to having the highest max discharge and charging rates. The cells also coincidentally are the lowest weight at thirty four pounds, which continues the semi-portable philosophy.

G. Energizer 2000W 12V Power Inverter

The 2000W 12V power inverter is excessive for the design, but was donated for the use of the project. This inverter has a small screen that allows the user to see the input voltage and output wattage, which allows the user to see that the inverter is not outputting 2000 watts constantly. This inverter will only output the wattage necessary to power the devices that are connected to it, so as to not waste energy. This inverter has protections that will keep the devices connected to it safe. These protections include high and low voltage, temperature, and overload. With these protections the users devices should never be in danger of experiencing a surge in power [4]. On this inverter, the outputs are two USB slots and two outputs for standard United States AC power plugs.

H. Solar Charge Controller (PCB)

The solar charge controller in the design is a custom built component using a printed circuit board. It was designed using KiCad and ordered through JLCPCB. On this PCB are several parts vital to the successful operation of the entire device as a whole. These vital parts are: the TVS diode and the two Schottky diodes. The reason why these parts are so vital is that they protect the entire device from reverse current flow and polarity. Along with these diodes we have other important components like the PMOS, NPN BJT, and the fuse. The fuse is another important protection, just like the diodes, but will kill power across the circuit if the rated amperage is exceeded. This component is accessible and easily replaced in the event that the fuse is tripped.

The PMOS is used to drive the voltage and allow the battery to be charged when the gate of the mosfet has a sufficient voltage provided to it. When the voltage at the gate drops low, it will allow less charge to the battery.

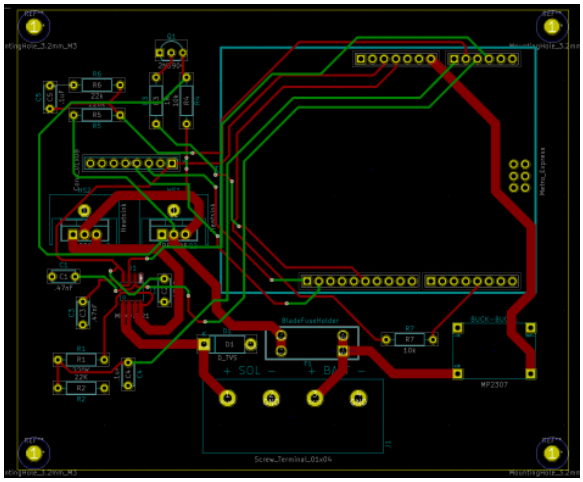

Figure 2: Printed Circuit Board Design

III. SYSTEM CONCEPT

The HCBSPS is a relatively linear system that allows the Adafruit Metro M0 Express to act as the centralized hub to the design. The devices that are connected around it would be considered tangential components that provide power and information to the microcontroller.

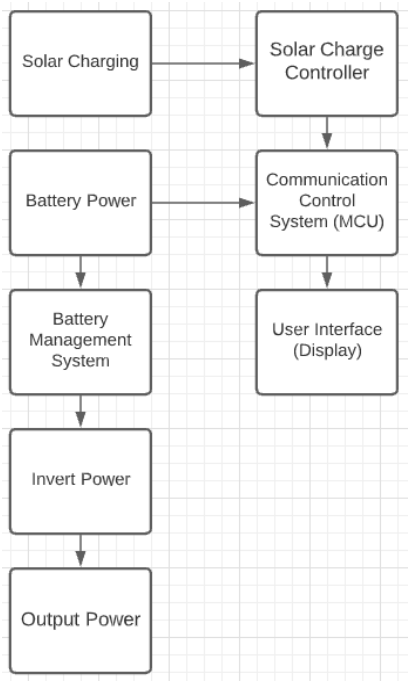

Figure 3: General Flowchart

IV. HARDWARE DETAILS

The hardware contained in this device consists of the items mentioned earlier in the system components. These include the solar panel, the battery cells, the inverter, the regulator, the LCD display, the microcontroller, and the charge controller.

A. Battery Enclosure

Not mentioned prior, is the custom created wood enclosure around the battery cells. This was created through a computer-aided design program, often referred to as CAD. After the cells are connected in series via bus bars, they will be encased in the custom plexiglass enclosure. Due to the extended time period of 3D printing an enclosure, as well as the cost and accessibility to durable materials for it, wood was chosen as the material.

Wood is a non-conductive material, relatively durable, and cost effective. With the wood the enclosure is designed as a shell, with the dimensions seen in figures 4 and 5 below.

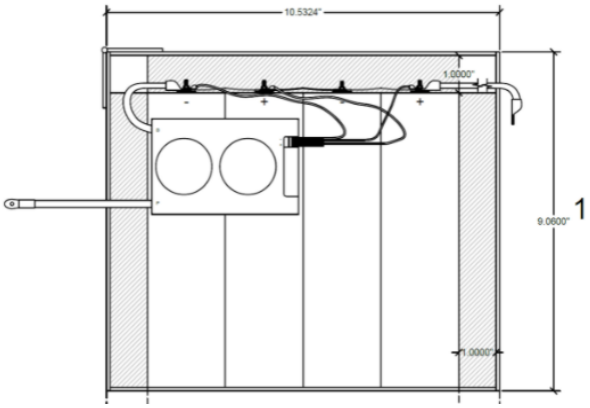

Figure 4: Side view of enclosure with BMS

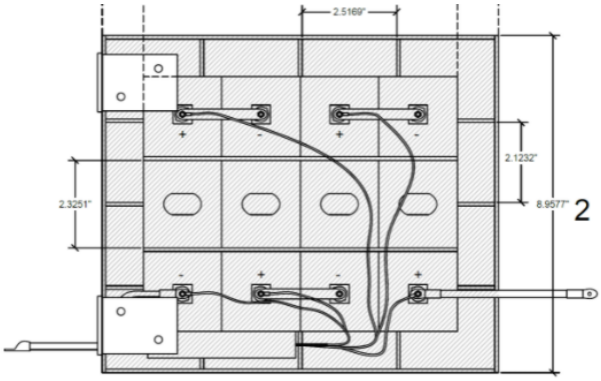

Figure 5: Top down view of custom enclosure

With this design and spacing, the enclosure is both stable and allows for proper heat dissipation. The cells are held tightly in place and the lid is attached with hinges so that it may swing open and close when needed.

B. Solar Charge Controller Enclosure

Along with the custom enclosure for the battery, an enclosure has been designed for the custom solar charge controller as well. Like the previous enclosure, the original material was intended to be 3D printed but adjusted to wood as a cost effective alternative.

V. SOFTWARE DETAILS

The software design is composed of many parts due to the many options and features that are available to the user. Therefore, discussing the software in multiple parts is most suitable. The charging algorithm will be presented before moving on to the graphical user interface (GUI). A minor portion of the software design on this project is the mobile application pre-built into the HEYO battery management system.

A. Charging Algorithm

The charging algorithm determines how the battery should be charged depending on the voltage across from it and its internal resistance, or in other words, its current state of charge (SoC). Different types of batteries may require different charging algorithms. For our battery, the charging algorithm only requires three stages; Bulk stage, Absorption stage, and Float stage. There's also an optional fourth stage, but that's only required for batteries that undergo through standby for more than a year [5]. The flowchart for the charging algorithm is provided in figure 6.

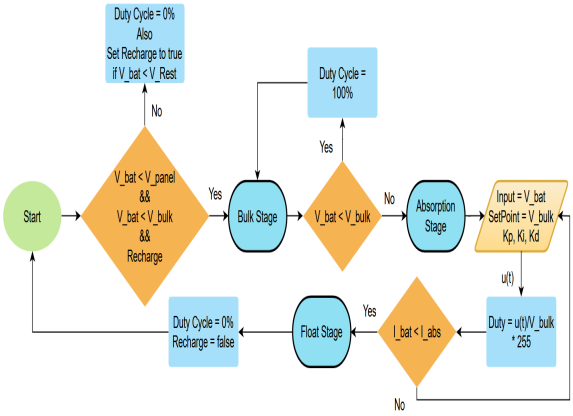

Figure 6: Flowchart for charging algorithm

V_{bat} is the voltage across the battery, V_{panel} is the voltage across the solar panel, V_{bulk} is the voltage of the battery at 100% SoC, I_{bat} is the current going into the battery, and I_{abs} is the absorption current threshold.

When the voltage across the solar panel (V_{panel}) is greater than the voltage across the battery (V_{bat}), charge is able to flow to the battery. The algorithm first checks if the voltage across the solar panel is greater than the voltage across the battery and if the battery is below the 100% SoC. If both of these conditions are true, then the charging algorithm enters the bulk stage. How the SoC is measured is by measuring the voltage and estimating the capacity based on the fact that 0% SoC is when the voltage is 10.8V and 100% when the voltage is 14.6V. The battery is able to handle 200A going through it. In this algorithm, we activate the transistor (thereby activating the Power MOSFET) by sending a 100% duty cycle pwm. The maximum current coming from the solar panel will not reach anywhere near 200A, so driving a 100% pwm is safe.

At a certain point, V_{bat} will reach V_{bulk} and then the algorithm will enter its second stage. At this stage, the internal resistance of the battery rises and the current going into the battery will decrease after some time. We will want to drive enough current such that the voltage across the battery stays around V_{bulk}. We can use a proportional-integral-derivative (PID) controller, which is used to drive a system towards a target value [2]. The PID controller takes in an input, the desired output, and three K constants. The equation is described as

$$u(t) = K_p e(t) + K_i \int_0^t e(t')dt' + K_d \frac{de(t)}{dt} . \quad (1)$$

e(t) is the error function and the K constants determine each of the terms weight. We found that setting K_p to 2, K_i to 5, and K_d to 1 gave satisfactory results. u(t) is used to determine the appropriate duty cycle for our pwm which correlates with the amount of current that is able to go into the battery.

Because of the PID controller, the voltage across the battery will maintain its maximum voltage, but the current will decrease to the point where it is below our threshold value I_{abs}. After this point, the algorithm enters its final stage, the float stage. In other types of batteries such as the lead acid, we would want the float stage to maintain the battery's full capacity while it's on standby. However, because LifePO4 batteries take so long to self-discharge, simply stopping the charge is appropriate for the float stage. There's a boolean value that will be set to false after the battery enters the float stage and prevent the cycle from continuing. The cycle only begins again once the battery's voltage falls below V_{Rest}, our maximum resting voltage.

B. Graphics User Interface

The GUI is the most complex part of the software and is composed of many different parts that are represented as screens as shown in Figures 9, and 11 through 15. These figures are not actual screenshots but are drawn to look like the actual screens for clarity. When the charge controller boots, it'll go through its initialization process. Once the process is done, the LCD will display the devices that were

initialized and the last active date that the device was on. The flow chart shown in figure 8 summarizes what the initialization process looks like.

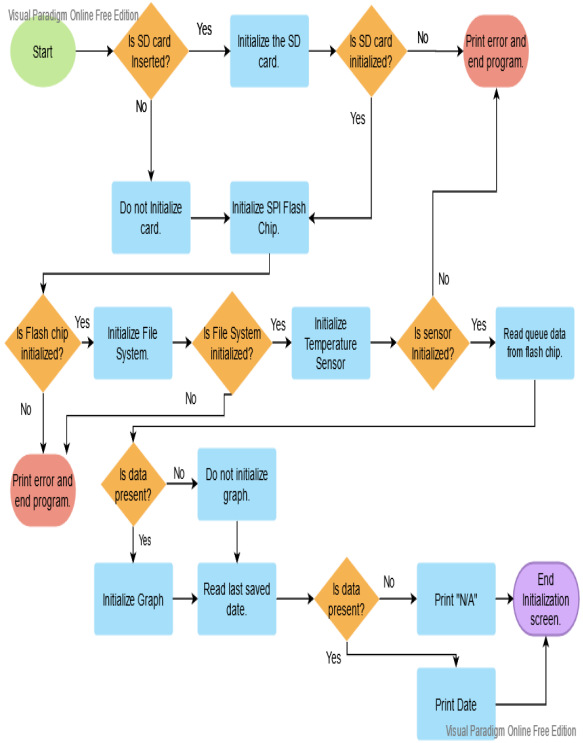

Figure 8: Flowchart for Initialization

The flowchart reveals that if any of the initialization process is unsuccessful, then the system will simply print an error and stop running the rest of the program. This is to prevent any unpredictable behavior from occurring in the main program such as trying to read the temperature when the temperature sensor couldn't initialize or writing files to the flash chip or SD card when they couldn't initialize either. As shown in the flowchart, the microcontroller will check for data. It will only check the flash memory chip since we're guaranteed for the device to be present and initialized. The microcontroller is looking for two files, one of them possessing saved queues data and another that stores a single date. That date, as mentioned before, is the last active date recorded. The file is overwritten whenever the charge controller turns on and after the RTC hits midnight. The file containing the data for the queues also updates whenever midnight occurs. Therefore, whenever the charge controller reboots, the graph restores its previous state.

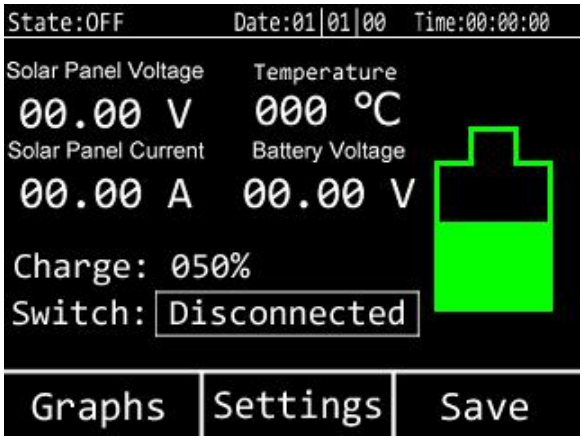

Figure 9: Main Screen

Once the initialization process is done, the user is presented with the screen above. The solar panel's voltage, the battery's voltage, the temperature, and the current coming from the solar panel and going into the battery are shown. In addition, the SoC is shown both qualitatively and quantitatively and the charging state as well as the date and time are shown on top. The date relies on a two digit year, the highest being 2063 or "63" and the time uses a 24 hour format due to simplicity. Finally, the button next to the "switch" label will either disable or enable the pwm. The button is useful for users that want to perform maintenance while the charge controller is connected and do not want the circuit to allow current to flow into the battery. The main screen has three buttons below that will take the user to the graph screen or settings screen. When the user touches "save," the current data on screen would be saved onto the flash chip or SD card if present.

In every screen, with the exception being when the user is viewing data, there are background processes that ensure that data is continuously being updated and that certain necessary checks are performed such as the SD card state. The figure below shows the processes performed while the user interacts with the GUI.

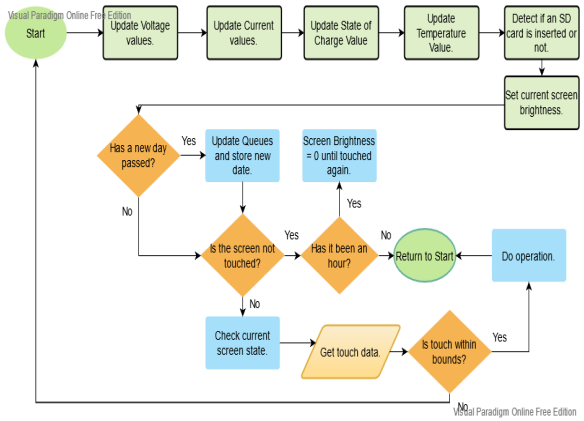

Figure 10: Flowchart for background processes

While the GUI runs, the software does several checks. Aside from checks related to critical charge controller operations such as checking if the temperature is too high or low or the voltage and current comparisons in the charging algorithms, the software also checks if a new day has passed, if a microSD card is inserted or not, and the time since the user used the touch screen. Whenever a new day has passed, the queues that store maximum values of the measured quantities are updated as mentioned before. When an SD card is inserted, the software would briefly initialize the card just like in the initialization screen before going back to normal operations. Finally, after an hour has passed since the user touched the screen, the screen brightness would temporarily be set to 0% until the user touches the screen again. This is to save power and it is the only power saving feature that the GUI uses since the CPU is required for the whole program. The use of interrupts was considered for the SD card insertion function, however, we determined that this method was safer and is quick enough. If interrupts were used, we would have to find and test numerous places within the code to enable and disable interrupts for safe operations. For example we would definitely want to disable the interrupt while files were being written to the flash chip. However, enabling or disabling the interrupt while the files are being read, the graph is being drawn, the PID controller is active, or when the values are being sampled/updated are some examples when it isn't as obvious and would require much more testing.

The green processes with black outlines are actually separate functions simplified for this flowchart. The "Do operation" process is also simplified since the touch screen operations vary from screen to screen. The microcontroller performs many operations at any given time, but the 48 MHz clock speed helps maintain a pleasant user experience.

In conjunction with the data, the GUI allows for the user to have basic options to adjust on the display. These basic options include: screen brightness, font color, temperature unit, precision, and energy unit. This can be seen in figure 11.

Figure 11: Settings Display First Screen

The settings should be self explanatory to the user. In regards to the "precision" setting, the user is able to adjust the number of decimal places. How the user adjusts it would not affect the actual value since it simply truncates it. The setting is there if the user wants to simplify the readings. The screen brightness changes in 10% increments or decrements from 10% to 100%.

The settings screen does contain a second screen. This screen contains simple features like changing the date and time. This allows the user to have an accurate date and time in the event that an incident occurs with the device. This is important for the graph part of the GUI. The RTC keeps track of important events such as leap years and months that end on different days. However, the user can change the date in ways that do not make sense if there weren't conditions that ensure the changes are correct. For example, the user is allowed to change the date to 09/31/21. In our software if the user isn't allowed to make those kinds of changes to the date. In addition, our software is able to check for daylight savings in accordance with US rules. The user is not able to change the seconds within the program. This is for the sake of keeping a consistent timer. As a result, the time might be slightly off. But if the user was given the option to change the time in seconds, the time would already be inaccurate whenever the user changes it since time would have gone by the time the user makes changes. Devices that didn't depend on the internet to keep time would typically have the user only change the hours or minutes like with the Nintendo DS and set the seconds back to 0 whenever the changes were made.

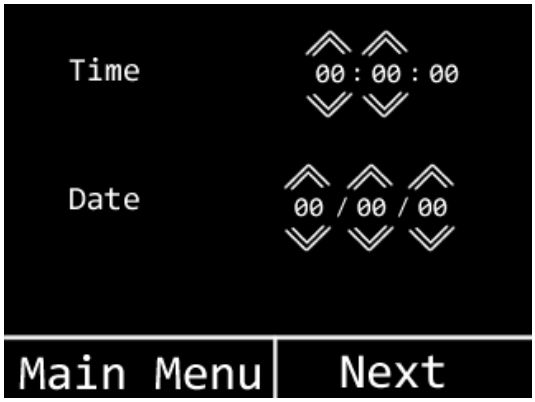

Figure 12: Settings Display Second Screen

The following two screenshots show that the user has the option to view and erase data in the third and last screen setting. While the SD card is not inserted, the default read and write operations occur in the flash chip memory. Deletion, however, only occurs in the flash chip regardless if the SD card is inserted or not. Whenever the user deletes data, not only is the flash chip cleared but also the queues that store graph values. We made deletion exclusive to the flash chip because space is much more limited on the chip compared to a typical SD card and files that store queue values and dates are exclusively saved into the flash chip. Users who use SD cards can view the data from the charge controller or their PC, but even without an SD card, the user can still save and view data from the unit.

Figure 13: Data Viewing Screen

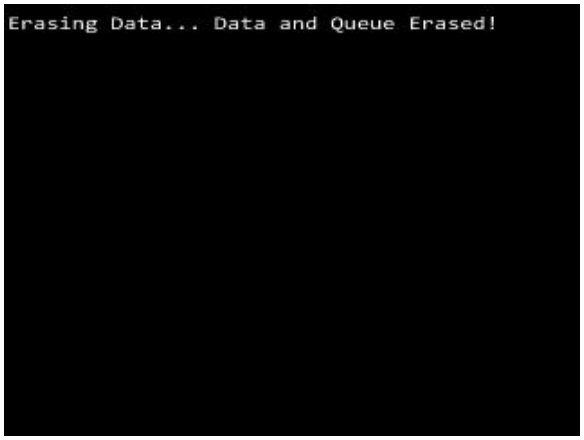

Figure 14: Erasing Data Screen

Another important feature shown through the LCD display is the graphs screen. These graphs show a function related to time, like the battery voltage, and contain seven days worth of information at a time. This timeframe was chosen to conserve the amount of data stored and keep it to a smaller size. An example of the graph screen can be seen below:

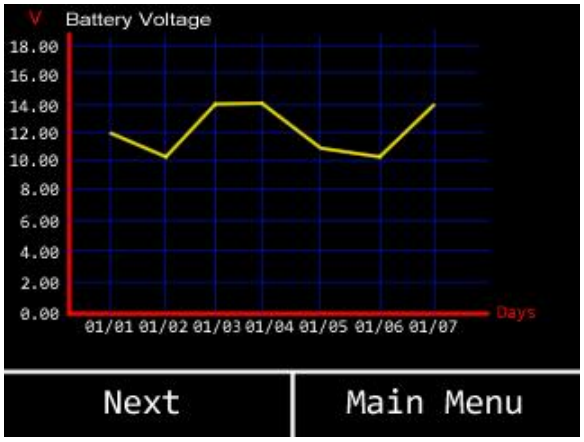

Figure 15: Graph Screen Example

The data plotted are all maximum values for the dates displayed in the horizontal axis. As mentioned before, the data are stored within queues. The queue is the best data structure for this purpose due to its First-In-First-Out (FIFO) structure. When the queue becomes full, the earliest date is removed and the new date is plotted. The user will always get values from the last seven days. The purpose of the graph is to provide information about the system. If anything occurs, such as damage to the solar panels or the battery, that information can be reflected on the graph and users can use the dates to correlate events such as a storm.

A. HEYO Battery Management System Software

The HEYO Battery Management System utilizes a smartphone application called “Smart BMS”. This application allows for a bluetooth connection, through a unique name, for remote monitoring of the battery management system. In addition to remote monitoring of the battery the user can manage the information immediately as well as set and read the protection parameters for the battery.

VI. ADDITIONAL FEATURES

The HCBSPS was designed and assembled during the COVID-19 pandemic, this caused an unnecessary constraint of finding areas that were safe to congregate in long enough to progress on the design. Unfortunately it lead to a few additional features not being implemented. Most notably the additional outputs to power a larger quantity of items. The feature of semi-portability was not altogether cut, but was not implemented as well as was considered in the initial vision.

VII. CONCLUSION

HCBSPS should be a convenient solution to an individual's need for renewable energy on the move. It is an affordable solution with useful applications of items that are used on a daily basis. The most important part of the design is that it can be further scaled to allow for higher storage and consumption of power. When looking to the future, the HCBSPS has growth potential and many options for optimization.

VIII. THE ENGINEERS

Josiah Best is a 22-year old Electrical Engineering student with a focus in radio frequency and microwaves. He enjoys the hardware side of the field, and hopes to continue to gain experience in it through personal and professional projects. He currently works at a structural engineering firm as a lead designer and has come to enjoy the structural side of engineering. Upon graduation he will be pursuing a FE certification and become a civil EIT, in hopes to become a professional structural engineer. Josiah has accepted a full time position with his current company and will remain there to continue his experience in the structural engineering field.

Mike Howell is a 29-year old Electrical Engineering student who is currently looking to utilize his United States military experience and Top Secret clearance along with his education to seek employment in the EE field. His focus is on the hardware side of the development process but he is willing to gain more knowledge and experience to be productive in the software side.

Eduard Meighan is a 24-year old electrical engineering student with a focus in power and renewable energy. He prefers the hardware aspect of the field and plans to continue to gain insight through personal and professional projects. He plans to continue his employment at a small company that designs and assembles heavy lift drones for commercial and government use. Currently he is a facility manager and oversees the construction of new drones with plans to acquire DOD contracts in the future. After graduation he plans to pursue further education in the form of a Master’s degree after a few years in the field. Eduard also plans to acquire a FE certification within a year of graduation to become a professional Electrical Engineer.

Jason Rodriguez is a 23-year old Computer Engineering student and a member of =c. He loves older technology and learning about low level programming and emulation. He hopes to find a job involving Very Large Scale Integration (VLSI) or firmware development. He plans to graduate and obtain work experience in the CPE field.

IX. REFERENCES

[1] "How The BMS Works." *Orion Li-Ion Battery Management System*, www.orionbms.com/general/how-it-works/
[2] "How to Read Solar-Panel Output Specifications?" Solar 4 RVs, www.solar4rvs.com.au/buying/buyer-guides/how-to-read-solar-panel-specifications/
[3] Dube, Ryan. "Capacitive vs. Resistive Touchscreens: What Are the Differences?" *MUO*, 28 Nov. 2018, www.makeuseof.com/tag/differences-capacitive-resistive-touchscreens-si
[4] Rozenblat, Lazar. "YOUR GUIDE TO DC to AC POWER INVERTERS." *Power Inverter Circuits: DC-AC Converter*, www.smps.us/power-inverter.html
[5] Power Sonic "How to charge Lithium Iron Phosphate (LiFePO4) Batteries" https://www.power-sonic.com/blog/how-to-charge-lithium-iron-phosphate-lifepo4-batteries/