# Net Zero Off-Grid Energy Management System

Jeff Redmond, *Member, IEEE*, Julia Redmond, *Member, IEEE*, Walter Emery, *Member, IEEE, and* Josh Moyers, *Member, IEEE*

***Abstract*—Contained within this paper is an approach to creating an energy management system for Photovoltaics with battery storage. The importance on this is pressing not only because of environmental concerns but because these energy sources are becoming more widespread and with forethought they can be deployed more economically and effectively. In section II the paper discusses how we will go about controlling our system. Section III details how we can simulate a power system. Section IV give a brief overview of component sizing and what part were used. Section V gives a detailed explanation of the user operates the system.**

## I. Introduction

Even with the best battery systems, energy from renewable generation sources is not always available. The sun does not always shine, and the wind does not always blow. Energy generation is also most abundant during off-peak power times, and energy storage becomes absolutely critical. Consumers often are not aware of just how much energy they are using. Most devices draw power even when turned off, a phenomenon known as "Vampire Draw". Smart management can mitigate the vampire draw of devices.

An Energy Management System (EMS) solves a few of the net-zero off-grid challenges. By monitoring loads and switching off devices during their vampire draw state, an energy management system can stop this wasteful state of energy usage. Less energy usage translates to a higher return on investment for renewable energy purchases, meaning less time to "break even." This is the time at which the initial renewable energy investment is equal to the power which the renewables have generated. An efficient home allows easier access to net-zero off-grid living and reduces dependence on foreign oil and the utility company. For a home tied into the grid, an EMS will still dramatically reduce a customer's energy usage.

However, the EMS must still be efficient. Monitoring loads also draws power, and a good EMS should save energy. The energy drawn by monitoring and controlling loads must be less than the energy saved by controlling the loads. A load management application is essential for user control. Without user input, the EMS will balance loads automatically and try to be as efficient as possible. The user will still have full control of the system and can specifically tell the EMS not to manage certain loads. A good time for this would be during peak load time when the washer is running, the TV is on, and dinner's in the oven.

The Graphical User Interface (GUI) has both autonomous and user-controlled inputs. It monitors all energy usage and can perform load-shedding to maximize efficiency and battery storage. The user also has the opportunity to control loads and efficiency through the GUI. The more renewable sources the user relies upon, the more useful the project, as the GUI has more components to combine for maximum efficiency. The GUI is written to be expandable, with the maximum amount of loads being much higher than what is designed.

## II. Fundamental Design and Control Flow

### A. Combined Data and Control Context Diagram

The EMS requires many inputs and outputs. In order to control loads, a low-voltage relay capable of passing high voltage signals is required to allow the switching of DC and AC power loads. The EMS is also monitoring each load's current through sensors, whose input data is displayed on a touchscreen interface. Voltage is not monitored as the voltage is controlled elsewhere and is assumed constant from the EMS perspective. This is accurate enough to allow good estimation of the power being used.

The system can control and monitor up to 4 loads. These loads can be large or small. Each load can draw a maximum of 15 Amps at either AC or DC. This is an expandable system and another relay module plus 4 current sensors is required in order to increase the monitoring and control capability to 8 loads, each at the same current level.

This project was funded through the Electric Power Research Institute's GridEd Undergraduate Design Grant.

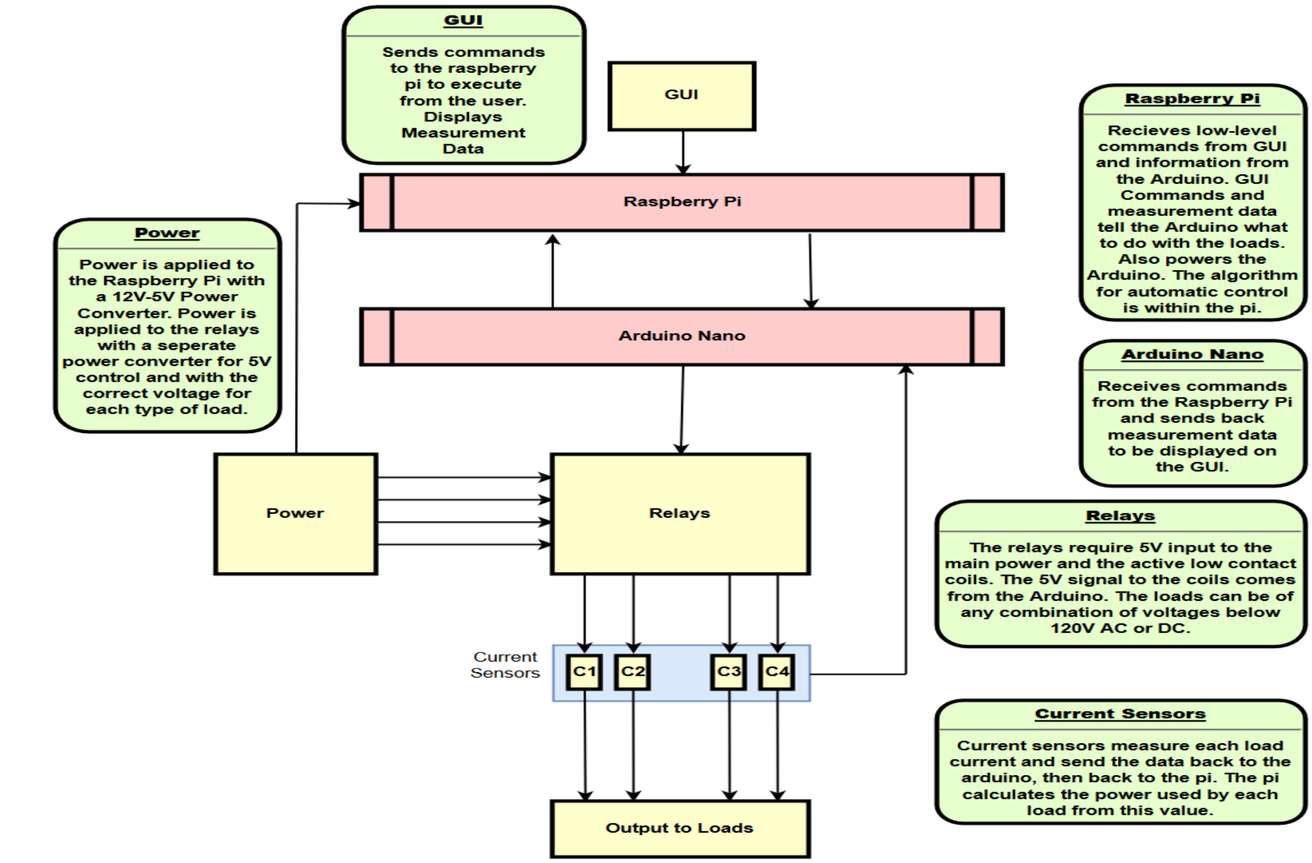


*Figure 1 - Data and Control Context Diagram*

The GUI is powered by a raspberry pi, where all monitoring and control is passed to an Arduino through a serial connection. This is done to centralize control and monitoring to replaceable devices. This is a two-way connection, where the Arduino receives commands from the pi and sends back monitoring data. Each Arduino can control up to 8 loads, with the pi capable of communicating with 4 Arduinos for a maximum of 32 loads. Internal algorithms dictate if and when the relays need to remove loads automatically. A diagram describing the inputs to the Arduino and the outputs to the relay is shown in Figure 1.

## *B. Combined Data Flow and Control Flow Diagrams*

Data flows to and from every controllable device except for the relay, which can only take inputs and not give any outputs. The GUI talks to the raspberry pi to tell it when to turn on/off loads and to determine if a load is designated as “essential”, meaning it cannot be shed. The raspberry pi is the central brain of the operation. Data flows to and from the raspberry pi from every device. The raspberry pi sends power data back to the GUI, makes final decisions on turning on/off loads, takes a current measurement from the Arduino for all four loads, determines if the load is drawing too much current, and automatically removes loads when they are not designated as essential. The Arduino takes four analog inputs from current sensors and sends this data back to the raspberry pi. The Arduino is the actual controller for the relays, but all relay control processing is done at the raspberry pi. The Arduino is part of the centralized control scheme, where the Raspberry Pi does not directly control anything, but the pi does all of the processing to tell the Arduino what to control. Figure 3 shows the data flow and control flow diagram.

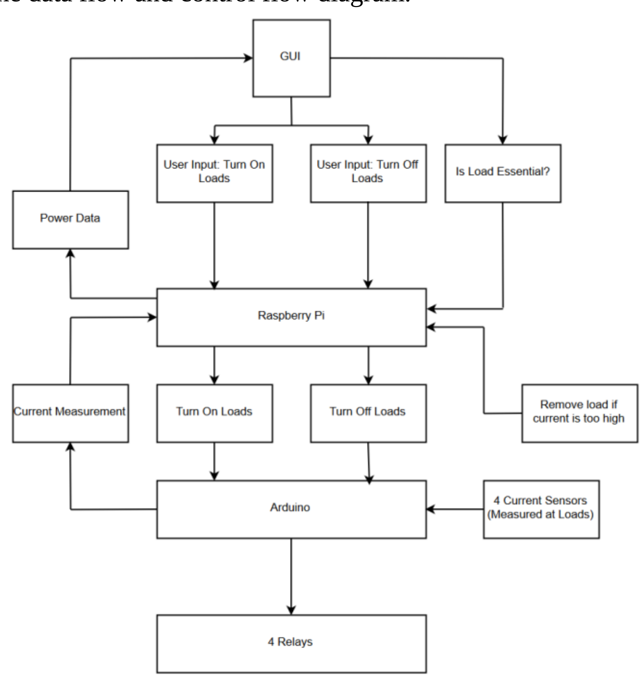


*Figure 2- Combined Data Flow and Control Flow Diagram*

### C. *Architectural Context Diagram*

The control system is interfaced with power generation and power storage components on one side, and on the other side of the relays is the loads. There is 12Vdc exterior and interior lights and a 12Vdc main bus capable of powering any 12V components. There is also an AC inverter capable of 2000W connected to the relay board, which allows the operation of AC loads. This diagram shows all of the connections except for the connections from the control flow diagram from the Raspberry Pi and Arduino. The arrows indicate the direction of power flow. The charge controller is in charge of dispatching loads from the battery as well as charging the battery. The inverter is directly connected to the battery due to the high current required by the inverter. There is uninterrupted power to the Raspberry Pi unless the whole system is shut down.

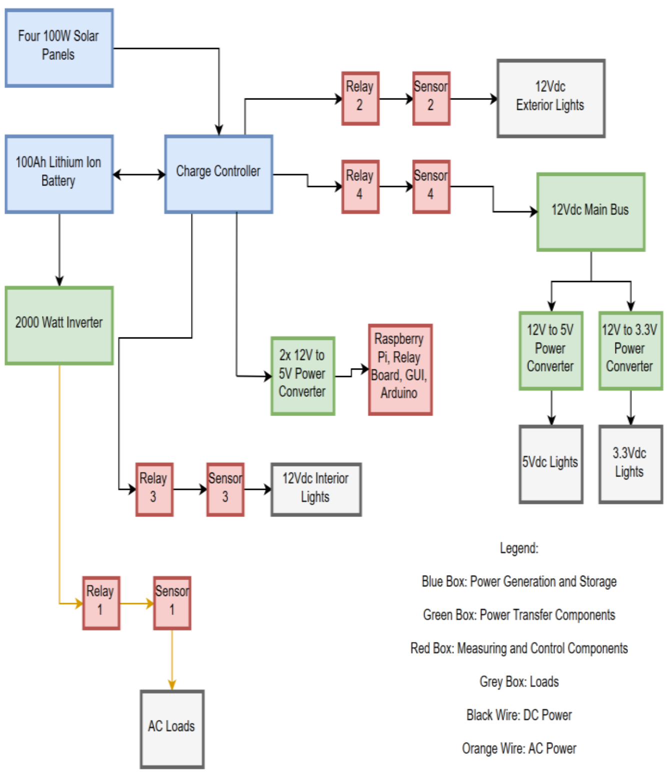


*Figure 3 - Architectural Context Diagram*

### D. *Combined Architectural Interconnect and Flow Diagram*

The combined architectural interconnect and flow diagram is shown in Figure 5. This figure combines Figure 4 and Figure 3 for a total overview of the system. All arrows point in the direction of power flow or information flow, as outlined in the legend. The one thing not shown is that all four relays are on the same board and have their own 5V power. The 2x 12V to 5V power converter block would have the second converter powering the whole board of relays to offload the current demand on the Arduino, allowing the Arduino to control more relays than it's current output would normally allow.

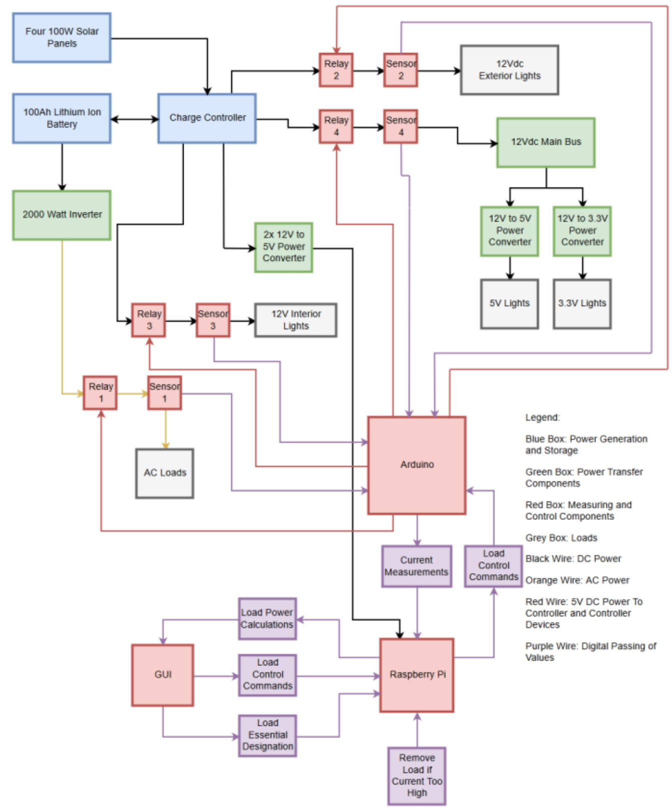


*Figure 4 - Combined Architectural Interconnect and Flow Diagram*

## III. SIMULATION

The high-level model of our circuit is shown in figure 5, it contains 5 stages. Stage one is the PV model this model contains a current source that is dependent on irradiance and the input is represented by the equation for $I_{Irr}$. There is a second current source that is opposite in direction that is dependent on temperature and is represented by the equation for $I_{dio}$ this stage also is directly dependent on the output voltage. The PV model could be simplified into only a current source but that would not capture all the dynamics. At the edge of the model a diode will block power flowing from the battery into the PV, which is an inherited quality of PV cells.

Stage two of the model is that of a buck boost converter, this stage will regulate the voltage from the PV model so it's voltage output will output 12V to the battery. Power will only flow one direction, from the PV and into the battery. The battery would need an additional charge controller in the real world; however, that was not represented here we assumed that holding the voltage at 12V would be enough. The equations and values are below in buck/boost section. The reason a buck/boost converter was chosen here is because our panels maximum output was 18V but during lower irradiance the voltage output would be lower thus our converter need both bucking and boosting capabilities.

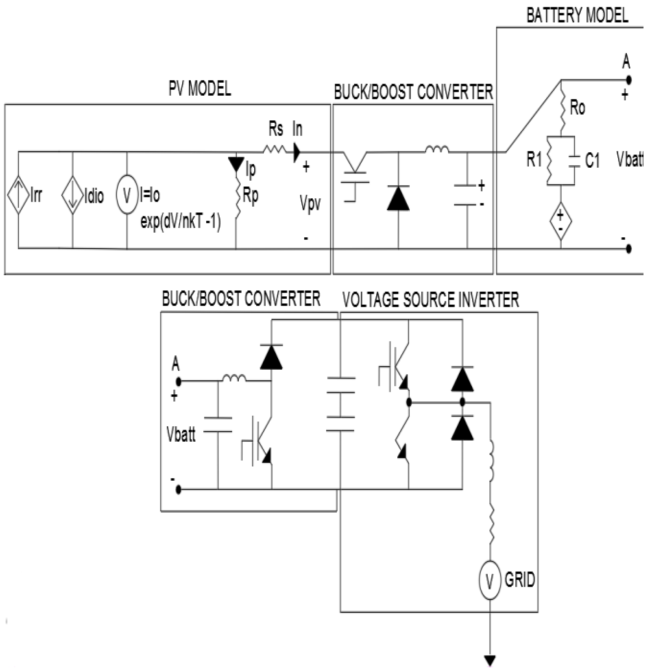


*Figure 5 - High level model of the system model*

The third stage will model a batteries behavior. This is based on the state of charge of the battery, the size of the battery, the open circuit voltage, and how much power is currently being drawn. This model does not account for the age or the temperature of the battery. The power converters on either side of the battery assume that voltage at the battery terminals will be within a certain tolerance of optimal. Equations for this model can be seen below. At this stage we will be tapping our DC loads, power will flow from the PV first if available or from the battery if not.

The fourth stage is a boost converter as seen by the battery of a buck converter as seen by the grid. The voltages at both ends of this converter are held nearly constant, on side will be nearly always 12V and the far end will be held to 40V. This happens because the battery in will act as a voltage source and the voltage source inverter on the other side will output 40V.

The final stage, the voltage source inverter, is what will transform our DC sources into an AC source. Depending if the system is connected to a grid or is creating one will affect what type of controller is used here. Our project will not be connecting to a grid, so our voltage source inverter will need to create its own frequency.

**Boost Converter**

Steady State Values:

Battery Voltage: $V_{batt} = 12\ V$

Voltage output of the boost stage: $V_{boost\ out} = 40\ V$

Duty Cycle: $d = \frac{V_{batt}}{V_{boost\ out}} = 0.3$

| | Diode | Transistor |
|---|---|---|
| $V_{peak}$ | $40\ V$ | $40\ V$ |
| $d$ | 70% | 30% |
| $I_{peak}$ | $33.3\ A$ | $33.3\ A$ |
| $< I >$ | $23.31\ A$ | $9.99\ A$ |
| $I_{RMS}$ | $27.9\ A$ | $18.2\ A$ |

**Boost Converter State Equations**:

$$\frac{d}{dt} i_{Lp0} = \frac{1}{L_{p0}} \left( V_{cp1} - dVct0 \right)$$

$$\frac{d}{dt} V_{ct0} = \frac{1}{C_{t0}} \left( di_{Lp0} - \frac{V_{ct0}}{R_{eq}} + \widetilde{i_{ac}} \right)$$

$$\frac{d}{dt} V_{cp1} = \frac{1}{C_{p1}} \left( \frac{V_{batt} + V_{cp1} + \widetilde{v_{ac}}}{R_b} - i_{Lp0} \right)$$

$$V_{boost\ out} = V_{ct0}$$

**Reactive Element Sizing**:

$$C_{t0} \geq \frac{I_{peak} d T_{sw}}{\Delta V_{Ct0}} = 832.50\ \mu F$$

$$L_{p0} \geq \frac{V_{peak} d T_{sw}}{\Delta i_{Lp0}} = 1.2\ mH$$

Let $C_{t0} = C_{p1} = 1000 \mu F$ and $L_{p0} = 2\ mH$

**Battery Model equations** [2]:

$$v(t) = V_{oc} z(t) - R_1 i_{r1}(t) - R_0 i(t)$$

$$z(t) = z(t_0) - \frac{1}{Q} \int_{t_0}^{t} i(\tau) d\tau$$

$$\dot{z}(t) = -\frac{i(t)}{Q}$$

$V_{oc}$: open circuit voltage of the battery
$z(t)$: state of charge of the battery
$Q$: Total

## IV. Calculations and Modeling Sizing

Load sizing:
LED strips: 3 LEDs in series/segment. For single color LEDs, 20mA/LED @ 3.3V or 20mA/segment @ 12V. For 3 color RGB LEDS 60mA/segment @ 12V. 10 segments/meter for 30 LEDs/meter = 600mA for RGB. 20 segments/meter for 60 LEDs/meter = 1200mA for RGB. For 40 feet: 1200 mA/meter*12 meters = 14.4A; P=IV: 14.4A * 12V = 172W/hr

1250W AC travel hair dryer, 125V, 60Hz: 15 min use = 313w/hr

Total watts for system: (172W/hr * 3.5 hours) + (1250W/hr * 1/4hr) + misc components = 945W/day

Solar:
100W solar panel @ 4 hours of sun = 400W of charging per day. 4 panels at reduced efficiency for 1000W

Battery:
100Ah battery * 12V = 1200W @ 80% depth of discharge = 960W

Inverter:
2000W inverter with 4000 W peak power

Requirements:
One: 100Ah LIFEPO4 battery giving 960W efficient storage; Four: 100 watt flexible solar panels giving 1000W of charging power; One: 2000W inverter with 4000W peak power.

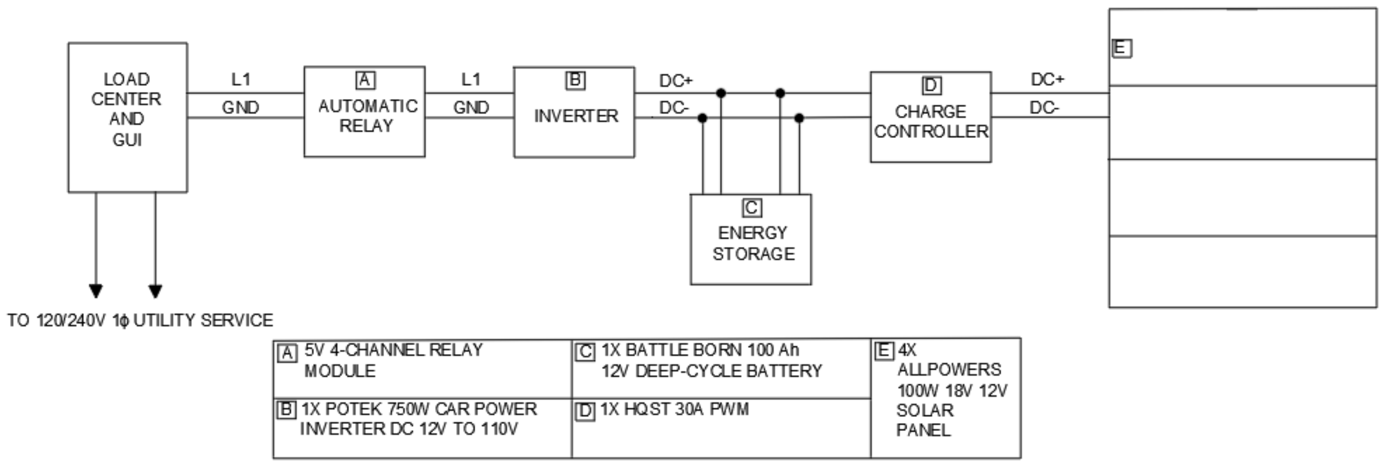


*Figure 6*

## V. Operating Manual

Operation of the Energy Management System is designed to be simple and intuitive. The program can be loaded by double tapping the icon on the desktop of the touchscreen. Once running, the program greets the user with four buttons marked “AC Loads”, “Ext. Lights”, “Int. Lights”, and “Main Bus”. Under these labels is where the power measurement shows up. Upon initialization, this should display “No Power Flow”.

Directly under the power monitor is the option to turn on each load. Pressing on for any load will toggle the respective relay and a load will come online. The power monitor will change to displaying the load power draw. Pressing off will revert to initial state. Under this is the “essential” button. This is used to designate if the load will time out or not. If the essential box is checked, the load will not be automatically switched out. Otherwise, the load will be switched out after a user-defined time. This timeout must be pre-defined before running the EMS program. If there is no user operation on the EMS for 5 minutes, the display will time out, turning it off. To bring it back online you only need to tap the screen and you will have full control of the system once again.

When the AC Load is turned on, the power strip connected to the inverter becomes live. The inverter must be on for this to work. Very large loads must be connected directly to the inverter due to limitations of the relays which control the loads. If your load is too large for the relay, the relay will trip out the load. This is an indication to plug your load into the inverter instead. The limit for the relays is 1200 Watts, which is below the rating of the relay just to be safe

When the exterior lights are turned on, the lights around the exterior immediately turn on. If the raspberry pi is online, the exterior lights can be brought online. Otherwise they cannot. There is no alternative method like there is with large AC loads. The exterior lights do not have enough power draw to cause an overcurrent protective trip on the relay.

The interior lights cannot be brought online unless the main bus is online. The main bus powers any 12V systems inside the house, including the interior lights. If the interior lights are switched online without the main bus, the power meter will read zero Watts, until the main bus is brought online.

The main bus controls everything inside of the house that is not part of the AC Loads. Power is monitored for all of the interior lights as well as the power converters to 5V and 3.3V. There is two light switches that control a few more things connected to the main bus. When these are brought online, the power monitor reflects the additional load.

The relays will stay online indefinitely, or until the battery gets too low and the raspberry pi shuts off, unless the “essential” button is not pressed. A good candidate for the “essential” designation is the main bus, as everything non-AC inside the house depends on the main bus to function. Leaving the essential designation off for the interior lights is recommended, as this ensures there is no possibility to leave lights on overnight or after you leave.

If this were in an actual house with additional loads, raspberry pi management systems could be placed on multiple floors of the house, or even in multiple rooms. Turning lights on in the room with a timeout would mitigate young children leaving lights on and wasting energy or leaving a TV on after you fall asleep watching it. This feature is designed to mitigate phantom draw as well, so the TV would be removed from the circuit completely instead of draw power while turned off. This has the effect of reducing carbon emissions if you’re on the grid or extending battery life if you’re off grid.

Pressing Exit will turn off all loads until the next time the GUI is initialized, and loads are switched online. If the raspberry pi is turned off or loses power, all loads will be turned off as well. The system can run indefinitely as long as there is enough power stored in the battery to keep the raspberry pi running.

[Type here]

## VI. References


[1] [1] M. Carrasco and F. Mancilla-David, "Maximum power point tracking algorithms for single-stage photovoltaic power plants under time-varying reactive power injection," Solar Energy, vol. 132, pp. 321–331, 2016.

[2] [2] F. Zhang, M. M. U. Rehman, H. Wang, Y. Levron, G. Plett, R. Zane, and D. Maksimovic, "State-of-charge estimation based on microcontroller-implemented sigma-point Kalman filter in a modular cell balancing system for Lithium-Ion battery packs," 2015 IEEE 16th Workshop on Control and Modeling for Power Electronics (COMPEL), 2015.


## VII. Biographies

**Julia Redmond** was born in Stanford Hospital in California on March 16, 1986. She studied law at the University of Denver, graduating in 2011, and Electrical Engineering at the University of Colorado Denver, graduating in 2018.

Her employment experience included Xcel Energy, where she worked on Distribution System Performance, and NEI Electric Power Engineering working on Substation Design. Her special areas of interest include renewable energy systems and system protection.

**Jeff Redmond** was born in Neenah Wisconsin, on June 24, 1984. He graduated from Redstone College with an AOS in Wind Energy Technology and from the University of Colorado, Denver with a BS in Electrical Engineering.

He is a U.S. Army Combat Engineer Veteran. His employment experience includes bringing industrial scale wind turbines back online with Sky Climber Wind Solutions and Vestas, solar installation as a sub-contractor with Solar City, and Substation Field Engineering intern work with Xcel Energy.

Jeff Redmond is the owner and founder of Drop Zone Drone, LLC specializing in aerial photography and video. He is also the owner and founder of Red Bear Rentals, LLC which brings peer to peer rentals, sales, and services to one convenient online location for comparisons and purchases.

**Walter L. Emery** is a resident of Denver Colorado. A recent graduate from the University of Colorado Denver. His interests include Power Electronics, Power Systems, and Renewable Energy resources.

He worked as an intern with Xcel Energy in the Distribution Planning and Strategy Department. Currently He is working at Salient Power Engineering as an Engineer I working on substation relays and protection.

**Joshua Moyers** was born in Honolulu, Hawaii in the United States of America on June 20, 1996. He studied Electrical Engineering at the University of Colorado, Denver.

At this time, Joshua is employed with the United States Department of Energy working as a Protection Engineer dealing with high voltage transmission lines. His special topic of interest is power system automation and control.