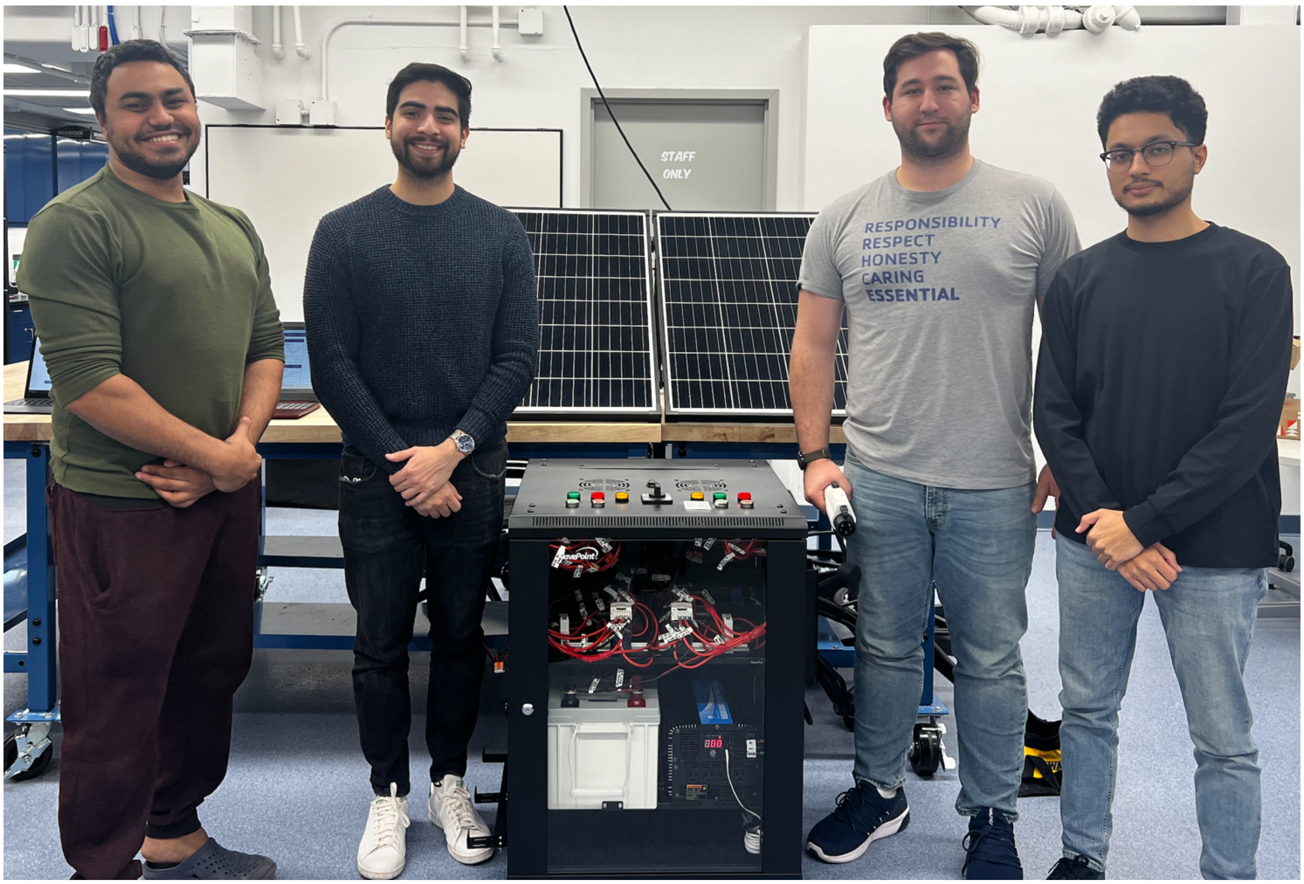
Final Project Report



From Left to Right: Abanoub Said, Carlos Figueroa, Eric Iliyev, Mostafa Younes

Senior Design Project – EV Charger, Fall 2022

Mentor: Ahmed Mohammad

Objective

The increasing market share of electric cars and renewable energy resources saw a significant investment made by the federal government for utility companies. This project aims to assist consumers and utilities through the development and implementation of a hybrid Level 1 EV Charger. The system will be powered using either a 120V outlet or a solar-battery circuit composed of high-voltage switches with current and voltage sensors connected to a microcontroller. The final product will be mobile, aesthetically appealing, and practical.

Table of Contents


Final System Demonstration

System Operation

The system can be operated using two modes:

1. Outlet
2. Solar-Battery-Inverter

Control System

The control system is responsible for switching the source of 120V in the system. Operations of the controls are mechanically controlled electrical components. There are three modes on the knob: 0 – Off, 1 – Outlet, and 2 – Inverter. Three videos demonstrating the system: Normal Operation, Software Demonstration, and Sensor location.

Main Control Panel

The left side of the controls are related to the Outlet. The right side are related to the Inverter. The Orange LED light indicates 120V Power at the source bus bar, the Red LED indicates the EV Charger is Off, the Green LED indicates the EV Charger is on.

The switch between the two fans is an On/Off rocker to control the case fans.

To turn the EV Charger On using the Outlet:

1) Plug the cord into an outlet
2) Ensure the Orange and Red LED illuminate
3) Turn the switch left to position 1
4) Push the green button
5) A clicking sound will occur and the Red LED will turn off
6) The Green LED will illuminate
7) The EV Charger should indicate "Ready to Charge"

To turn the EV Charger Off using the Outlet:

1) Push the Red button OR turn the switch right to position 0
2) A clicking sound will occur and the Green LED will turn off
3) The Red LED will illuminate

To turn the EV Charger On using the Inverter:

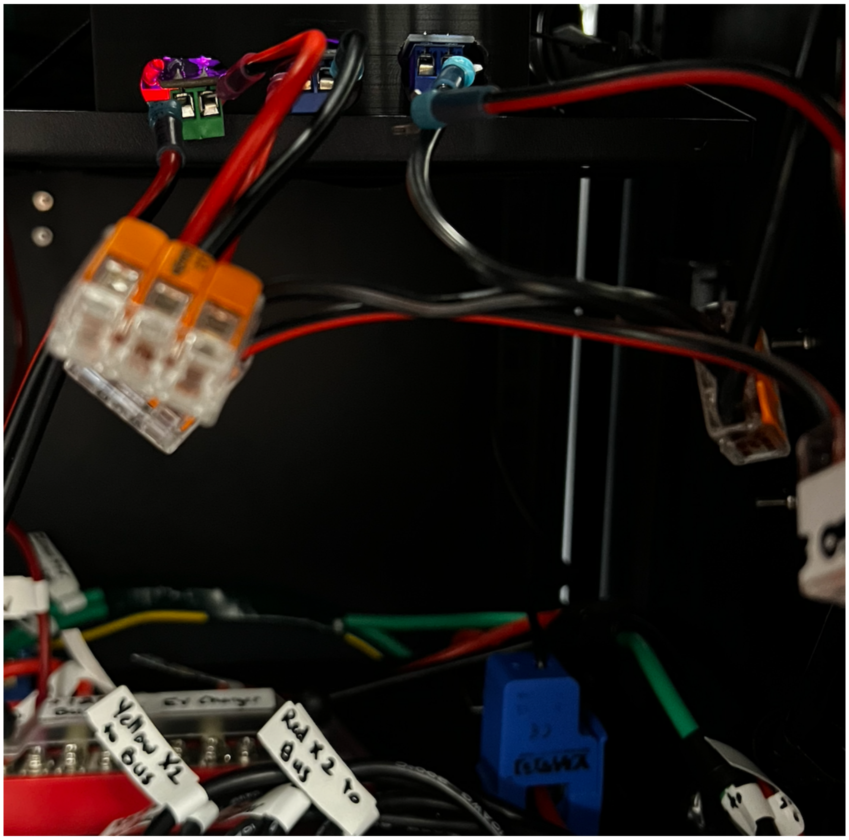
1) Press the Power button underneath the Orange LED to turn on the inverter
2) An audible beep should be heard and a small green "Power" light will illuminate
3) Ensure the Orange and Red LED illuminate
4) Turn the switch left to position 2
5) Push the green button
6) A clicking sound will occur and the Red LED will turn off
7) The Green LED will illuminate
8) The EV Charger should indicate "Ready to Charge"

To turn the EV Charger Off using the Inverter:

1) Push the Red button OR turn the switch right to position 0
2) A clicking sound will occur and the Green LED will turn off
3) The Red LED will illuminate

System Design and Images

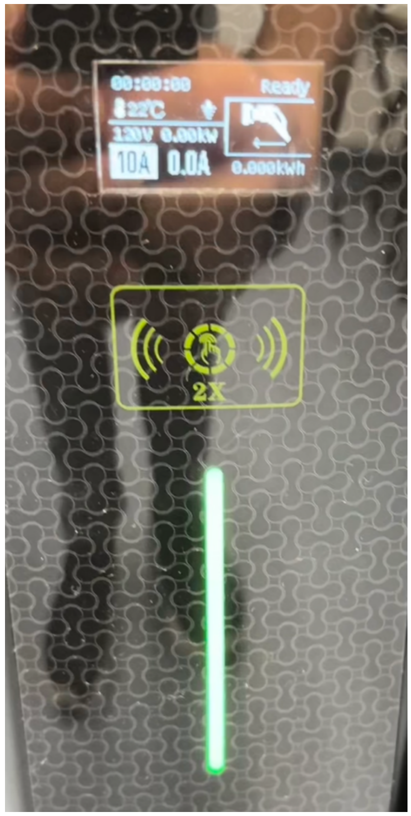
All wires, switches, and components are labeled to indicate their purpose and understand the system.



Voltage and Current Sensors

Illuminated Control System

Front Panel Wiring View



Powered Up EV Charger

Solar Panel PWM Controller

WARNINGS

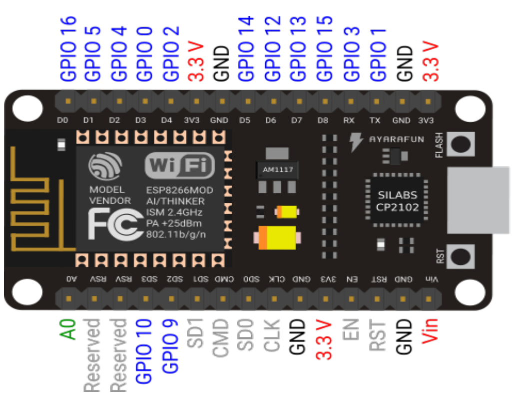
- The case fans will not turn on unless there is power to the EV Charger.
- The Energy Management system will not work unless the inverter is powered on.
- Solar panels will only charge the battery once the connectors on the left side of the case are made to the solar panel's PWM Controller wires.
- It is not possible to power the EV Charger using both the inverter and outlet at the same time.
- For proper grounding the 3-prong cable must be connected to an outlet.
- Case is not weatherproof. Do not expose to elements. Store in dry place at 60-80°F.

Energy Management

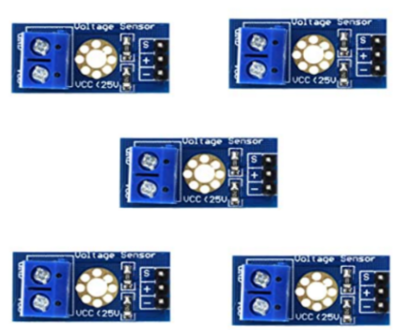
Purpose

The purpose of energy management system is to monitor how much energy is generated from solar panels which can help optimize savings and alleviate strain on the grid. For our system, we can measure how much power our solar panels output in real time and analyze our data using cloud data services. In addition, our system can measure the capacity of the battery as well as power at the inverter. Using this data, we can optimize our energy usage by relying on solar panels when needed to energize the EV charging. All necessary hardware components were fully integrated in our energy management system according to the specifications required by the high voltage system.

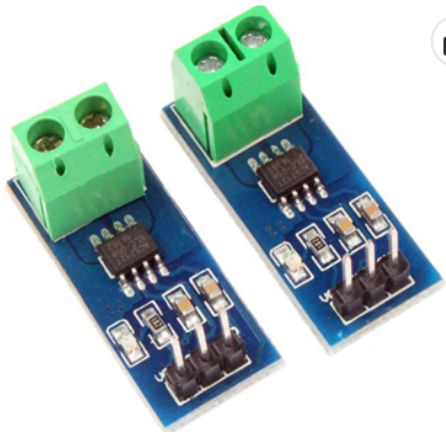
List of Hardware:



Node MCU ESP32

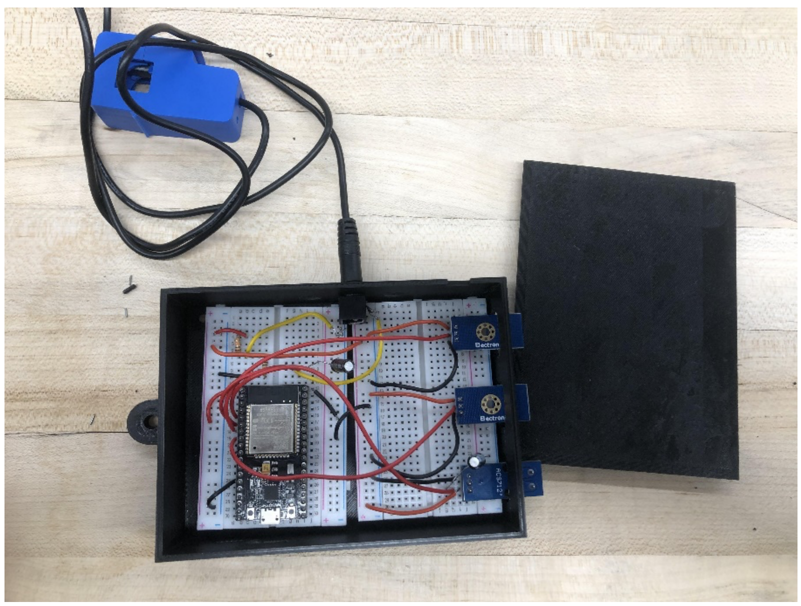


DC Voltage Sensors (25V range)



DC Current Sensor (30A Range)

SCT 100A AC Current Sensor

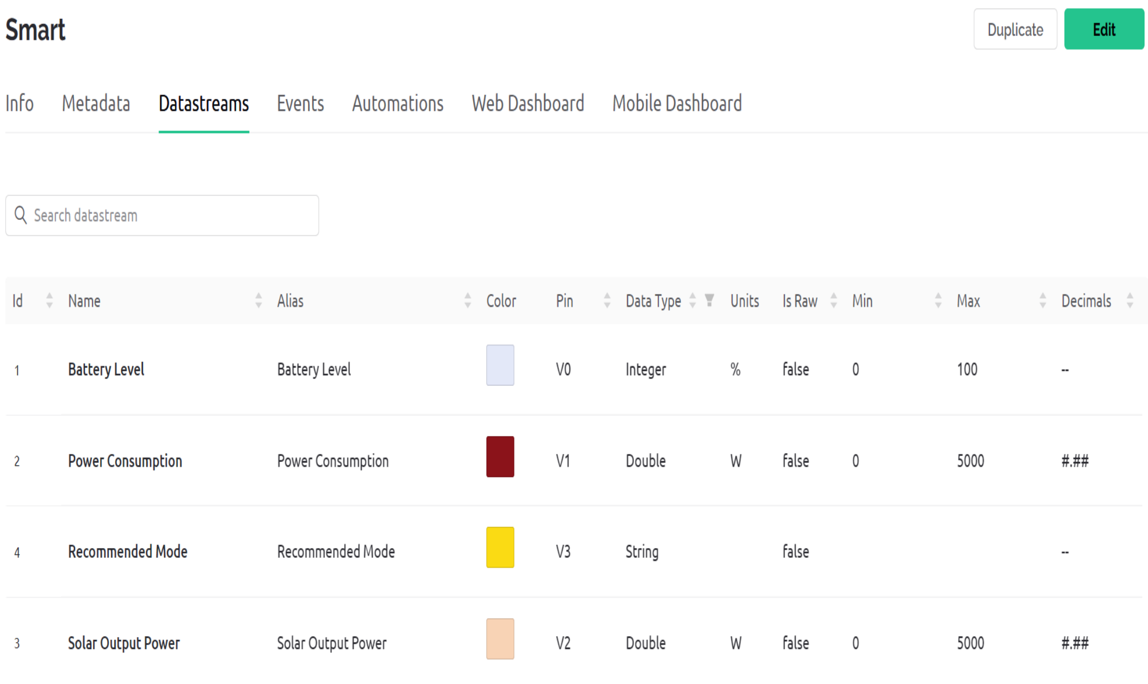


Circuit Built

For our system we have ESP32 WROOM, DC current sensor (30A), DC Voltage sensors (25V), and SCT 100A AC Current Sensor. The Node MCU ESP32 is a development board that has both Wi-Fi and Bluetooth capabilities. The board has 25 GPIO pins, 12 ADC Pins, two DAC pins, 2 I2C pins, and 2 SPI pins. For the DC Voltage sensors, we are using 0~25V Voltage Detection Module Sensor which uses a potential divider to allow you to read high voltage using a microcontroller. Using this module, we will be able to monitor the voltage at the battery level and at the PWM of the solar. For the DC Current Sensor, we are using ACS712 fully integrated, hall effect-based linear current sensor that can measure both DC Current and AC Current. This specific module is the ACS712ELCTR-30A-T which can read up to 30A of current and has a sensitivity of 66 mV/A. We calibrate this current sensor using the Arduino IDE and add a capacitor of 10 microfarad to reduce noise. This sensor is used to measure DC current between the battery and the solar. The 100A SCT Current sensor is a current transformer that is used to measure AC Current. This sensor is placed after the inverter in the system and is used to measure the current. For this sensor, a voltage divider circuit is built of two 10k Ohm resistors with a capacitor of 10 microfarad. After calibrating all the sensors using Arduino IDE, the data is then transferred to the Blynk Application as well as IOT ThingSpeak platform for monitoring and analysis.

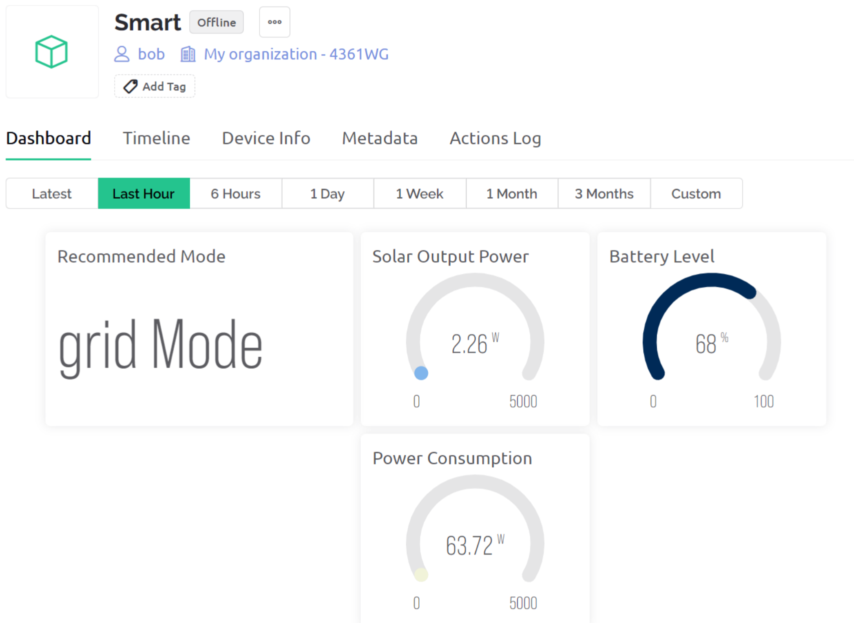
Cloud:

We have created two ways of monitoring the system over the cloud from anywhere in the world. An operation mode for the regular everyday operation and a limited one for instantaneous values. The former method utilizes Blynk IOT Platform where we can program our microcontroller to interact with a web dashboard to display useful information regarding the system and which mode is preferred for usage. This information can be displayed both on the web and on an application for the user. The latter method uses IOT Thingsspeak which is a way for the engineer's point of view, and it includes all the values measured and recorded by the system and allows for long term storage and analysis for further improvement and optimization of the system. The two methods are implemented using preexisting cloud infrastructure that are provided by math works and Blynk io. The system uses internet connection through Wi-Fi to interface with two APIS and send necessary requests to regularly update the data with sensor measurements. The system is designed to retrieve regular forecast and timing updates from a public API and timing servers and use it to predict and update the operating mode of the device.

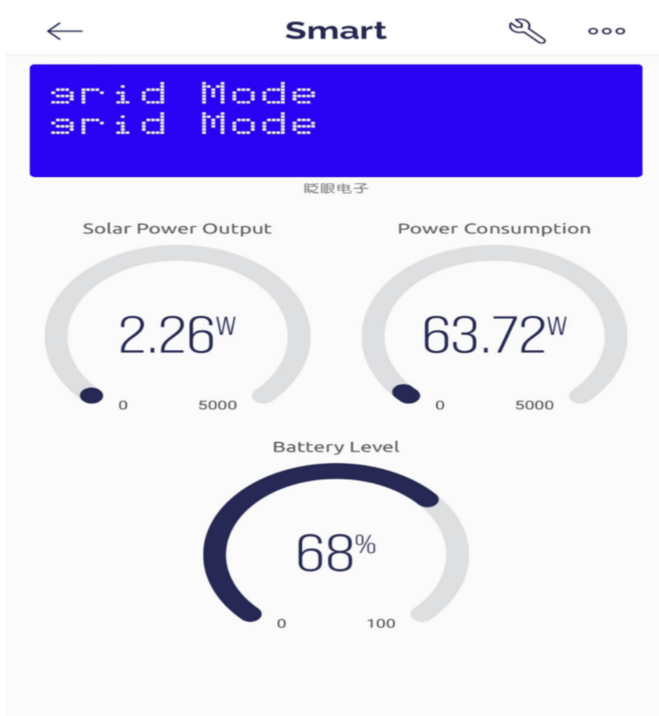


Data Streams

Using Blynk application, we designed our data streams using virtual pins V0, V1, and V2. These data streams hold information about the battery level, power consumption and the solar output power. Additionally pin V3 determines the desired mode of operation for the system. Virtual pins are a way of exchanging data between the hardware and the Blynk application. They are channels that are used to interface with Blynk built in libraries and implement custom logic. In our program, data collected from the sensors such as voltage and current are stored and used to determine the parameters that we want to display such as battery capacity, solar power, and power consumptions. The recommended mode is then determined based on these parameters including a weather forecast that is retrieved by the microcontroller from a public API which helps the user determine the mode of operation.

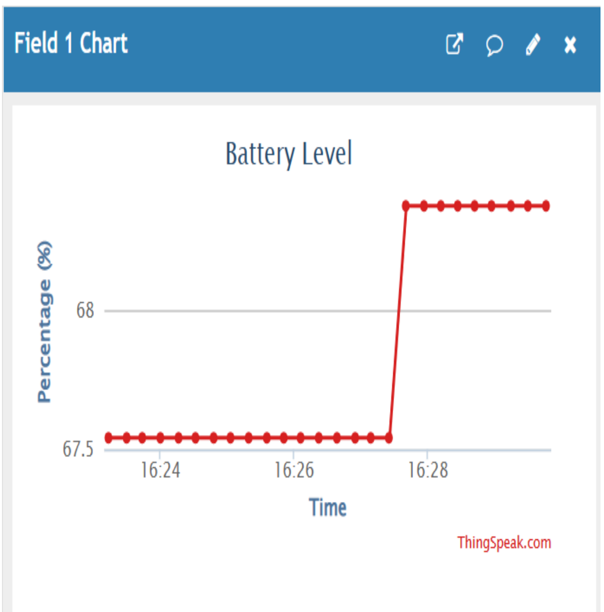


Web Dashboard

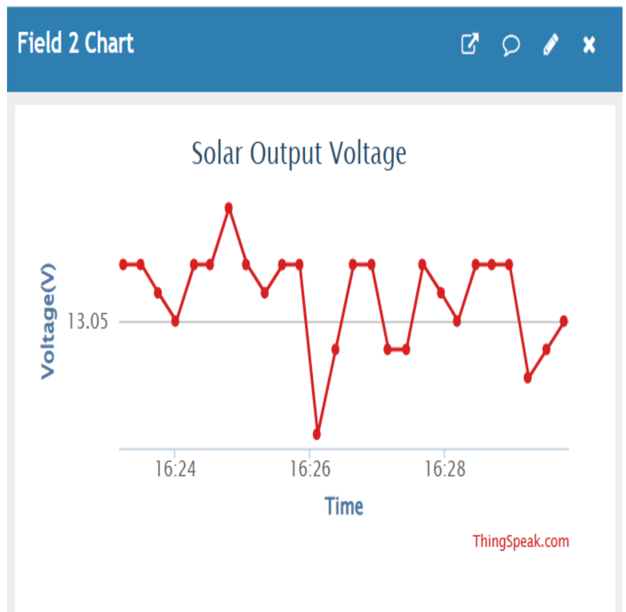


Mobile App

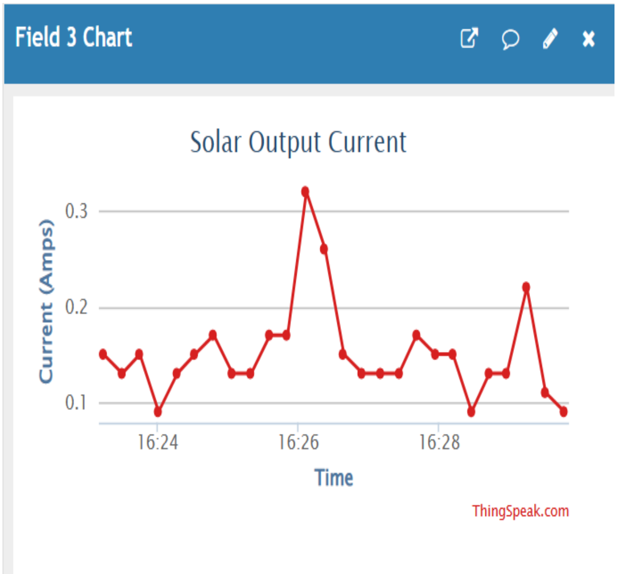
Using the data streams, we can build a web dashboard that displays the information needed such as the solar output power, the battery level, and the power consumption. To display useful information, we select a Gauge icon for each measurement and a string for the desired operation mode. Each gauge displays a different virtual pin as specified by the data stream. Battery level receives data from virtual pin V0, Power Consumption receives data from virtual pin V1, and Solar Power receives data from virtual pin V2. Lastly, the mode of operation receives data from virtual pin V3. Using this layout for our dashboard we can view the desired operation mode as well as monitor the data using both the web version and the mobile application version. During our demonstration, the following readings determined the desired operation mode as grid mode which is displayed on the mobile app.



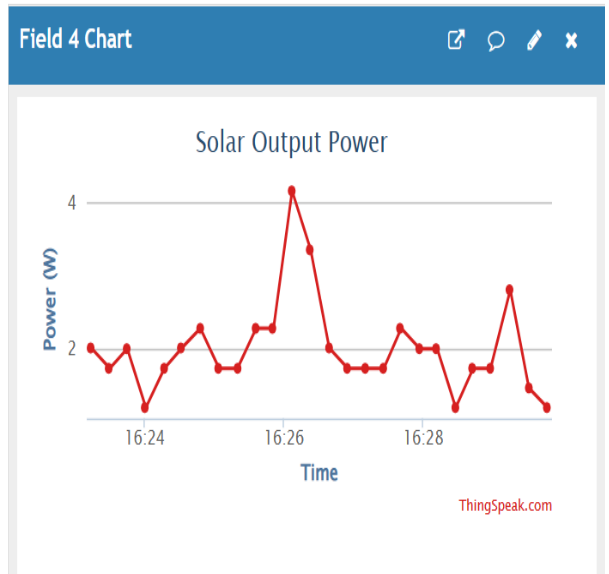
Battery Level Analysis



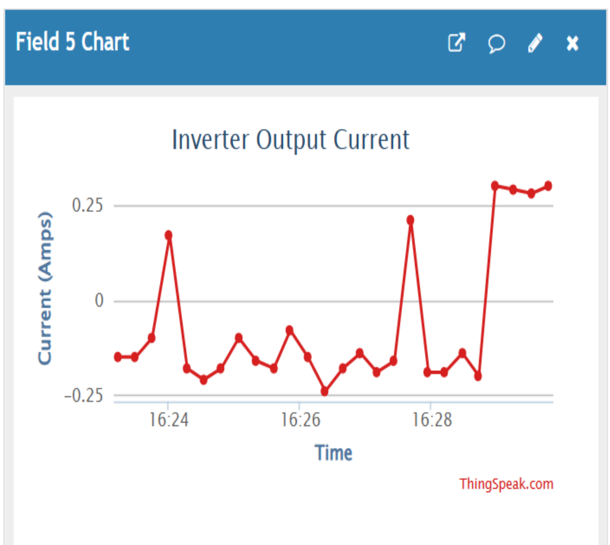
Solar Output Voltage Analysis



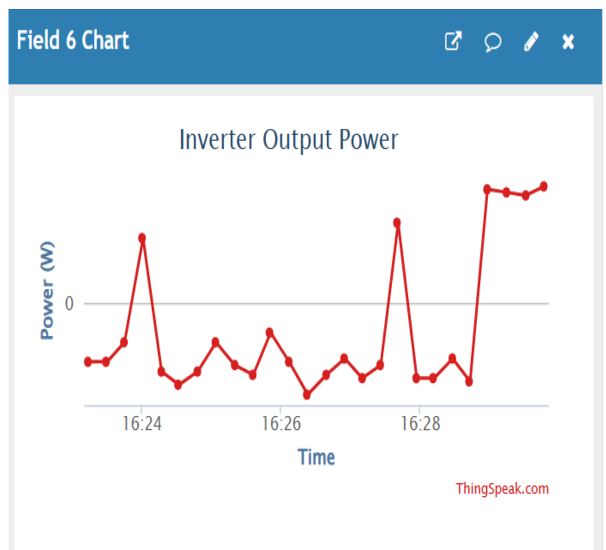
Solar Output Current Analysis



Solar Output Power Analysis



Inverter Output Current Analysis



Inverter Output Power Analysis

ThingSpeak uses channels to store data from a device. For our energy analysis, we created one channel which can hold up to 8 fields. We created 6 different fields that record the battery level, solar output voltage, output current, output power, inverter output current and inverter output power as shown above. In order to send real time data, we first had to use an API key which is selected from the platform and included in our program as well as the network used. Then, we store all the data retrieved from our sensors and the calculated values and display them in different fields as shown above. In field 1, we have the battery level which was stable at 67.5% during demonstation before the addition of load. In field 2, 3 and 4 show readings related to the solar panel. The solar output voltage shows slight fluctuations over time as well as solar output current and power and stabilizes over time. Field 5 and 6 show the inverter output current and power which changes slightly over time. These graphs are are useful in determining the system's behavior over time and the desired operation mode for the overall system.

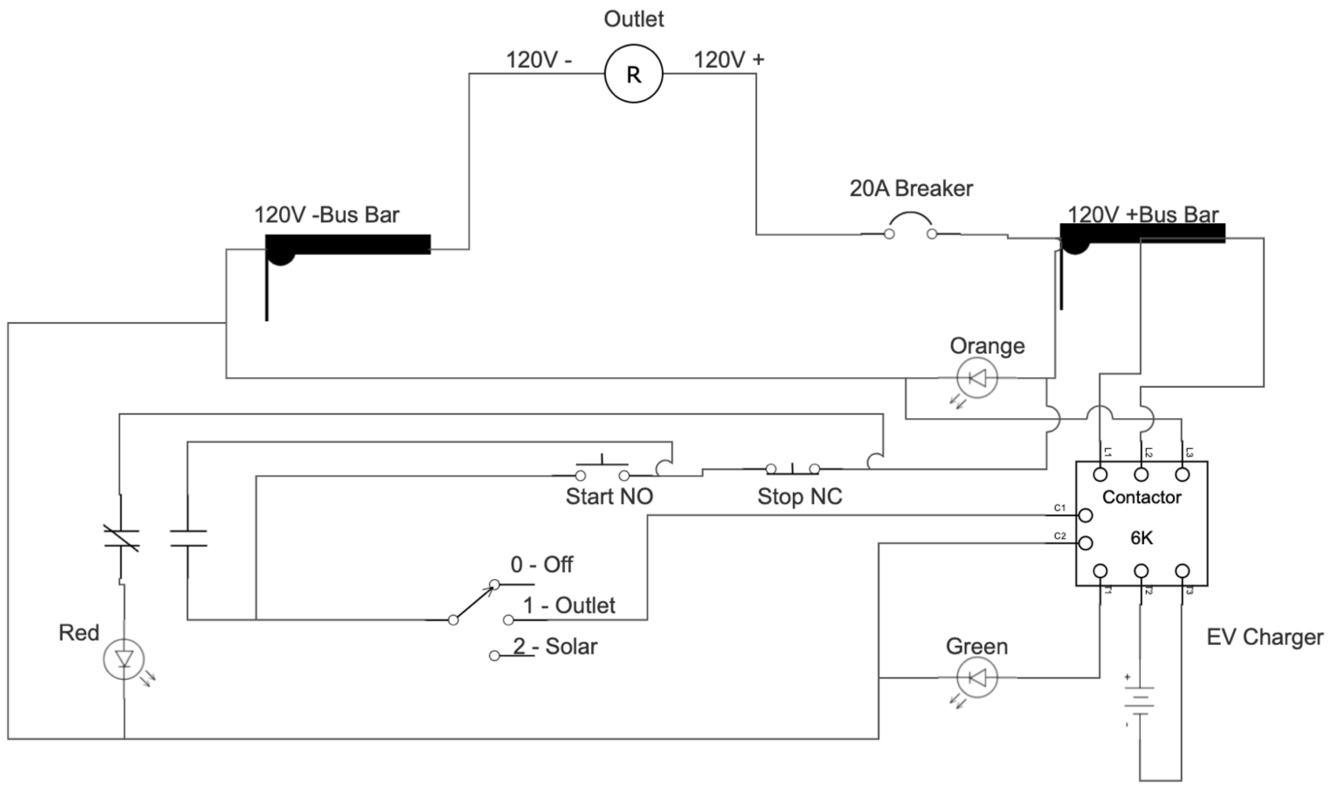
Limiatation:

For future improvement, we would recommend a more robust noise filter for our sensors. During the testing phase of the sensors, we noticed a lot of fluctuations in the sensor readings which prompted the use of filter capacitors. This resulted in more accurate readings however a more robust nosie filtering can be beneifical. The mode of switching capability of the system is another area that we can improve upon. In our system, the mode of switching is done manually due to the scope of the project, however this can improved by integrating the microcontroller to to automatically control this process. The software and the circuit protection would need to be modified and the system can then be controlled seamlessly from anywhere in the world and become self sustaining with any human interaction.

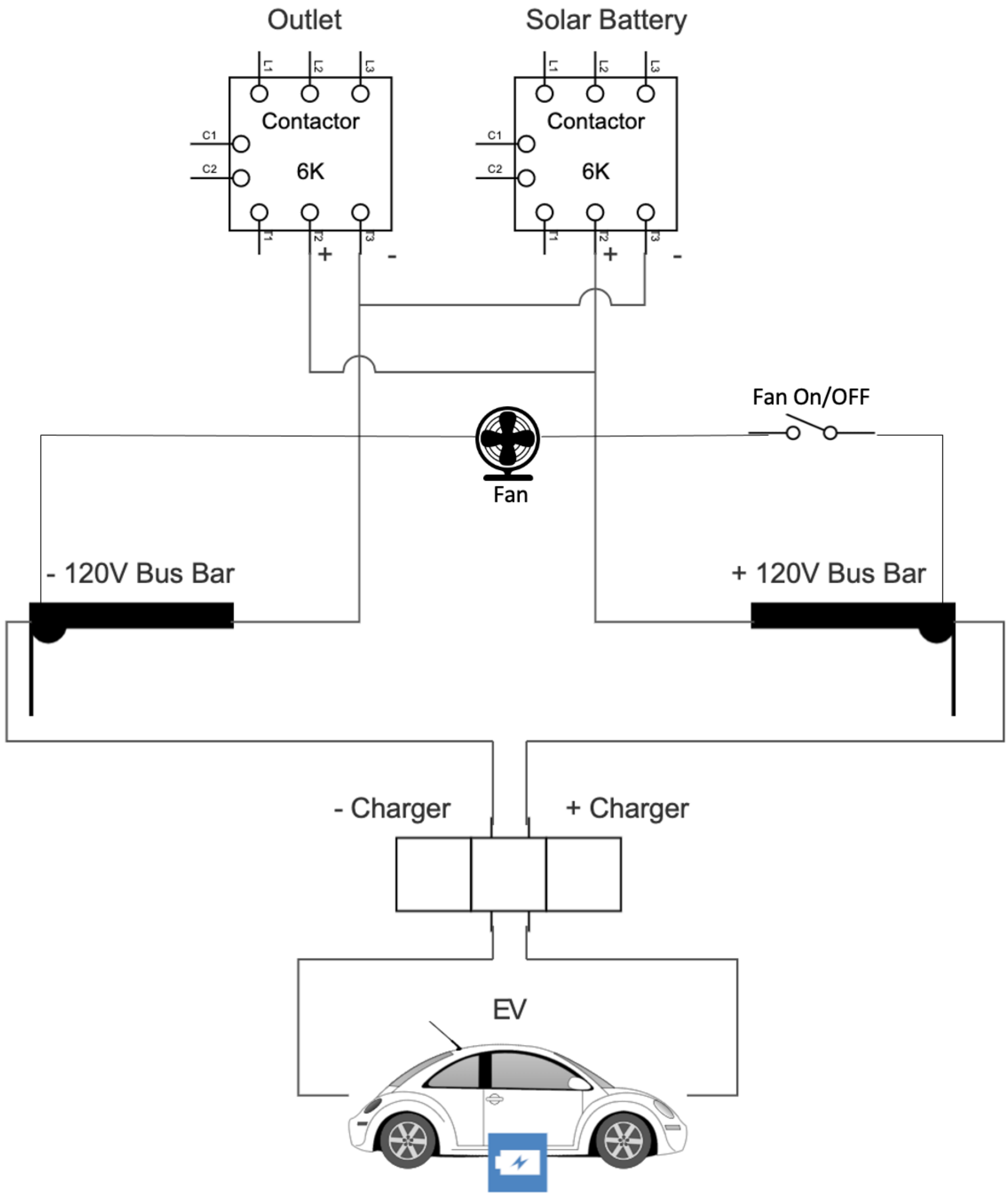
Sources:

- esp32-wroom-32_datasheet_en.pdf (espressif.com)
- ACS712GenMkt.indd (sparkfun.com)
- Voltage Sensor Module Pinout, Features, Specifications & Arduino Circuit (components101.com)
- SCT-013 Series Datasheet by Seeed Technology Co., Ltd | Digi-Key Electronics (digikey.be)
- Introduction - Blynk Documentation
- Virtual Pins - Blynk Documentation
- ThingSpeak Documentation (mathworks.com)
- API Reference - MATLAB & Simulink (mathworks.com)

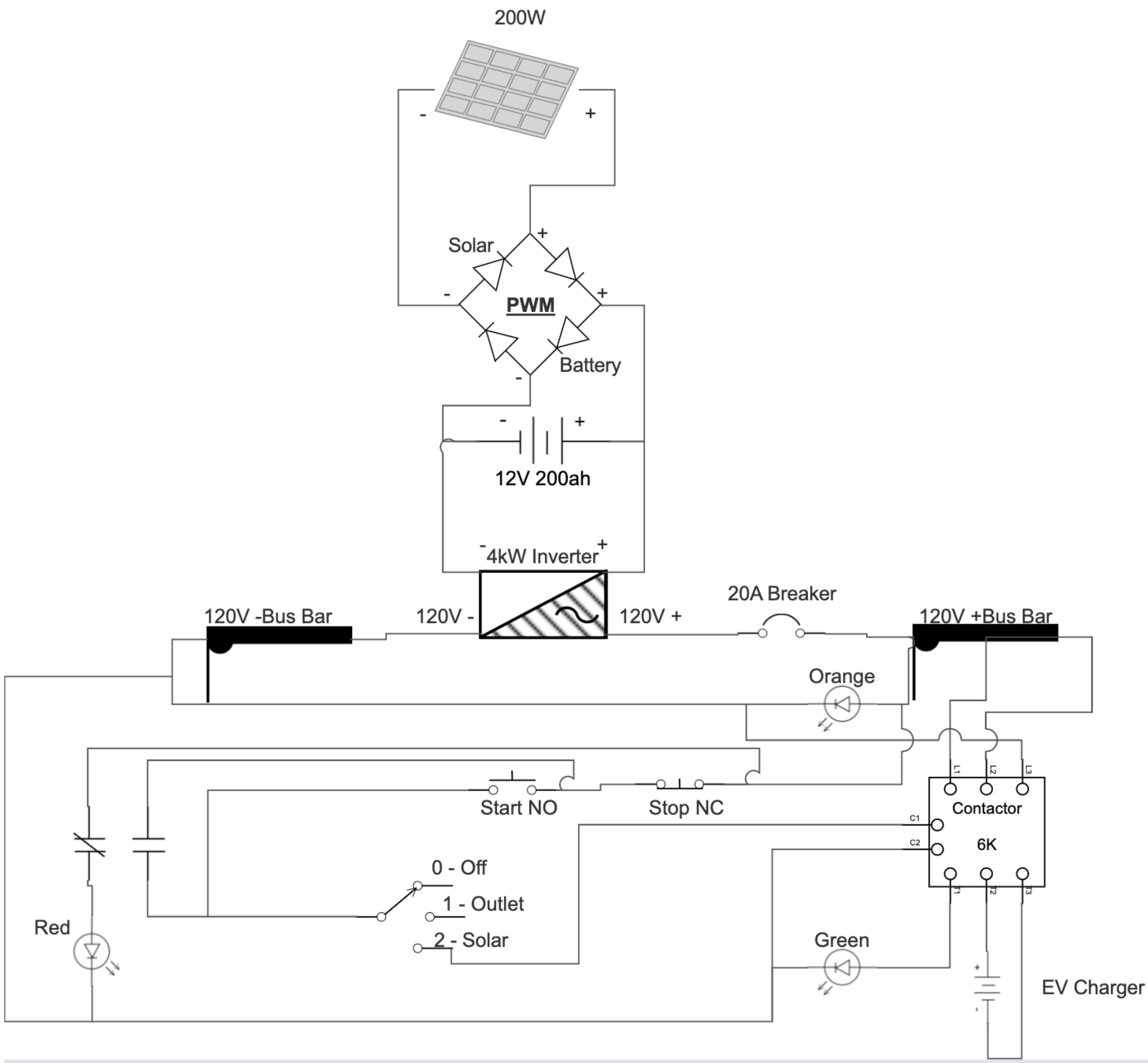
Schematics



Outlet Schematic



EV Bus Bars Schematic



Solar-Battery-Inverter Schematic