CS437: Internet of Things
Spring 2023 Capstone Project

Name: Michael Edukonis
NETID: meduk2
Late Days Used: 0
Video Link: https://youtu.be/vVUvGYvW_wY
Gitlab repository: https://gitlab.engr.illinois.edu/meduk2/capstone
Team Members: N/A

Final Project Proposal: IoT Electricity Use Monitor

Motivation:

The goal of this project is to build an IoT device that will track electricity usage. The motivation for this build is our growing need for energy and our lack of immediate visibility of our daily energy use. This project will focus on electricity. We have the ability to utilize off the shelf parts and some ingenuity to monitor and temper our electricity use. Most households take note of their electricity use when it's time to pay the bill. The device that may offer a glimpse into your real time electricity use, the electric meter, is inconveniently placed on the outside of the home in most situations. Smart meters are being rolled out however their abilities are lacking, and the energy companies are not motivated to spend time and resources to provide more functionality to the end user. The meter's primary purpose is to report usage to the company for billing.

There are some products on the market that provide readings for instantaneous energy use however they do not store any usage data or provide any meaningful insights. Since a household's main energy use is interior environment (highest) and hot water heating (2nd highest), the original thought was to track a heat pump and/or hot water heater usage with this device. It would involve a direct connection with parts that can handle the higher amperage.

Instead, this project will be a proof of concept that will focus on ease of use for the end user while utilizing standard off the shelf parts to handle 15-20 amps at 120v. The design of this device, in theory, can be expanded to anything in the household using electricity. One of the goals of this build is ease of use so the device will incorporate pass through ability, meaning the device will have male plug on one end to be plugged in to the socket and female on the other end of the device for the load to plug in to.

This device will give granular, instantaneous, and average use (through reporting) details to provide the user a very clear picture of their energy use. This information will be used to gain clarity and make informed decisions on energy use. If this device were deployed to multiple areas throughout a home or business, the data provided could be used to model costs, make the user aware of high energy use devices, and allow the user to adjust their usage of energy.

Technical approach and details:

Power monitor utilizes a WCS1800 Hall Effect Current sensor with analog output to an ESP-32 board with Wi-Fi and Bluetooth capability. The WCS1800 is pre-mounted to a board that includes the necessary circuitry for safe operation as outlined on the data sheet. The amperage being used by the circuit will be measured, stored, and broadcast on the network that it is connected to utilizing the Message Queuing Telemetry and Transport protocol (MQTT).

A hall effect current sensor provides a proportional signal in response to the strength of the magnetic field coming off the circuit being measured. The magnetic field is proportional to the amperage being pulled in the circuit. The measurement is taken from the hot part of the circuit that passes through the center of the toroid. The output of the sensor is a much smaller dc voltage that ranges in proportion to the amperage being utilized in the circuit. Because of the toroid shape, it may be challenging to simply insert the device into the circuit. Design of this device will prevent having to open the circuit to get a measurement.

The power monitor will have the ability to power itself from main power utilzing an 120v – 5v inverter and provide visual output via an OLED screen. A library is provided by the WCS1800 sensor manufacturer to conduct the calculations necessary. Output will be calibrated using known loads and another meter.

On first power up, the device will go into wireless access point mode. The user can connect to the device with their computer or phone and select their wireless access point, password, and mqtt broker details. This will be saved by the device, reboot itself and then connect to the directed access point. Device will have the ability to beacon periodic information in JSON or CSV format. Separate software (app, client, etc) can be utilized to receive this beacon and store/display the info. The board should also act as a server and provide information on demand as a client connects to it.

The device will have a safety measure built in to protect the user from excess current draw. The sensor is capable of handling 35 amps however the wiring is rated for a standard 15 amp household circuit. A 15 amp circuit breaker is connected at the point electricity enters the device.

Software:

ESP 32 board is programmable using the Arduino environment.

- Microsoft Visual Studio Code was also utilized for coding the project.
- Arduino provided MQTT client for publishing was utilized
- Wifi Manager provided by https://github.com/tzapu
    - Do not have to hardcode wifi AP or MQTT credentials
- Adafruit provided library for OLED screen on the ESP32 board.
- MQTT payload is in CSV format consisting of:
    - Date, Time (UTC), mqtt topic, Amps reading.

- Setup includes time sync with ntp.org. Once setup is completed, the reading takes place in the loop function at 1 second intervals.

Mosquitto MQTT Broker was installed on a linux desktop

- Custom MQTT client coded in python utilizing paho mqtt library

WCS_1800 sensor
A software library is provided by the company that makes the WCS_1800 sensor (Winson Semiconductor). The library is not compatible as is with ESP32 but some minor adjustments make it usable. These adjustments are explained under results and challenges.

Python scripts utilizing mapplotlib and pandas provide visual analysis of historic usage metrics for daily and hourly basis. Ideally, these will be run on a cron schedule and visuals will be saved to png or jpeg file and copied to the Apache2 webserver directory on the linux desktop.

Project Timeline for deliverables:

Milestone	Due Date	Status
Submit Proposal	20-Feb-23	Completed
Obtain parts	25-Feb-23	Completed
Test individual parts/order replacements	4-Mar-23	Completed
Wire prototype	11-Mar-23	Completed
Measure/order enclosure	18-Mar-23	Completed
Write software	19-Mar-23	Completed
Test/Debug/refactor	25-Mar-23	Completed
Modify/set/mount to enclosure	1-Apr-23	Completed
Expand notes/findings into report	28-Apr-23*	Completed
Produce Video	29-Apr-23*	Completed
Submit Report and Video	29-Apr-23*	Completed

*adjusted

Results and Challenges
The project is a success. Most of the original project proposal ideas were implemented without change however there were some challenges that needed to be resolved along the way:

- ESP32 OLED screen cable was too short to mount board inside with screen outside box. Next version will utilize a separate oled screen to give more mounting options. This issue was overcome with a design change to the enclosure.
- ESP32 OLED screen was too small to show all information at once
    - Software adjustment changes the screen output at 10 second intervals.

- The enclosure was printed multiple times in an attempt to shrink it as much as possible. Success was achieved with 100mm x 50mm box however assembly was very tight.
    - Different/smaller circuit breaker or resettable fuse would be helpful.
    - Layout parts and assemble on a custom PCB would aid assembly.

WCS_1800 sensor requires sensitivity adjustment in the software to dial in results. The procedure involved an inline and clamp ammeters on the live circuit to compare results and adjust accordingly (photo below).

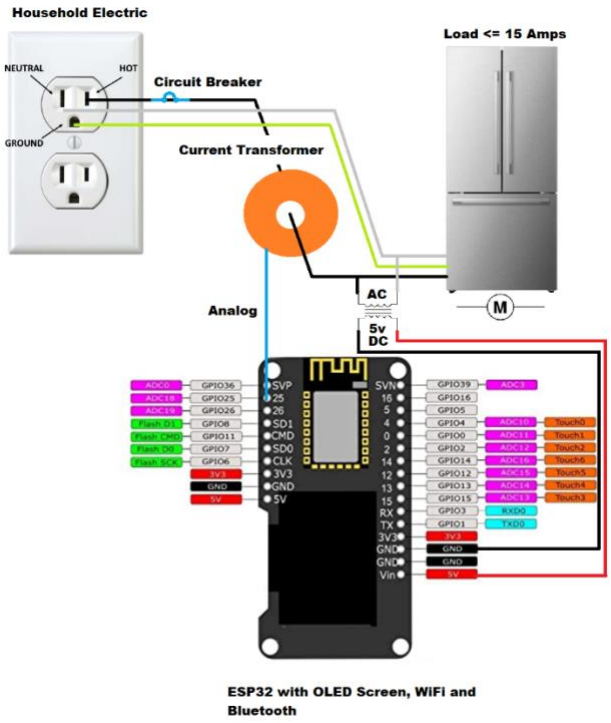
Winson provides an Arduino library to aid in the development and use of their products however it is one library to be used with multiple devices that they manufacture. The library is written specifically for atmel AVR chip used on most Arduino boards. The library was not compatible with the Espressif ESP32 utilized in this project. Specifically, the SoftwareSerial implementation required by another Winson sensor was preventing the compiling of the library with this project. To overcome this problem, the code in wcs.h and wcs.cpp library files was refactored to eliminate everything except the class required for WCS_1800 sensor.

Board continues working but stops transmitting over the network after approximately 8 hours. This requires a reset of the board which simply involves a button push. Network and MQTT credentials are stored so it reconnects automatically upon reboot however the reason for the bug is not yet understood. Further debugging is required and possibly implement a watchdog timer to reboot the device.

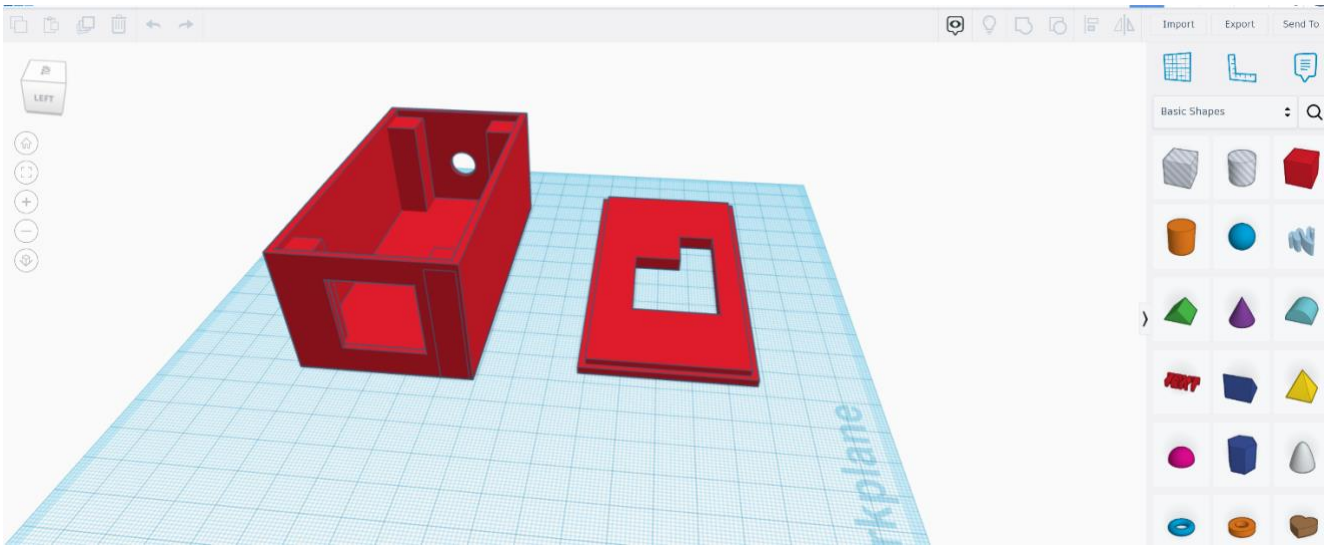
The power cord requires some kind of strain relief.

Physical Characteristics and circuit design

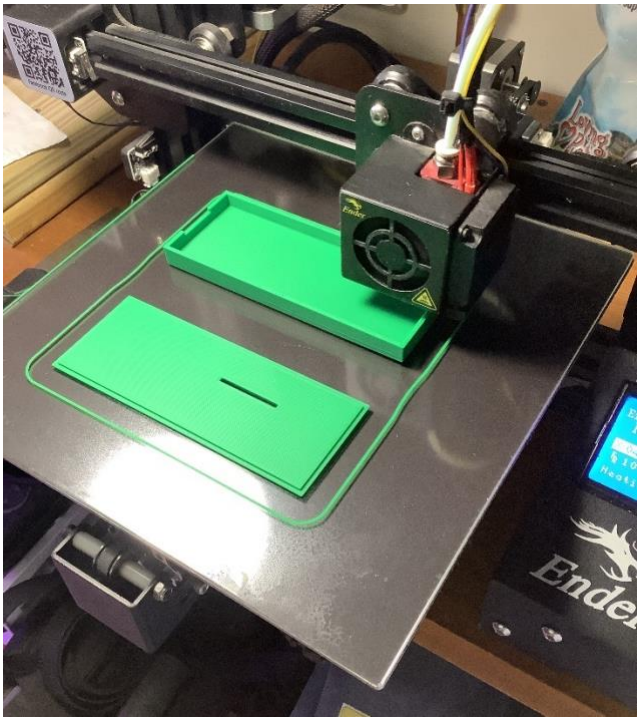
*Hall effect sensor used in place of a current transformer.



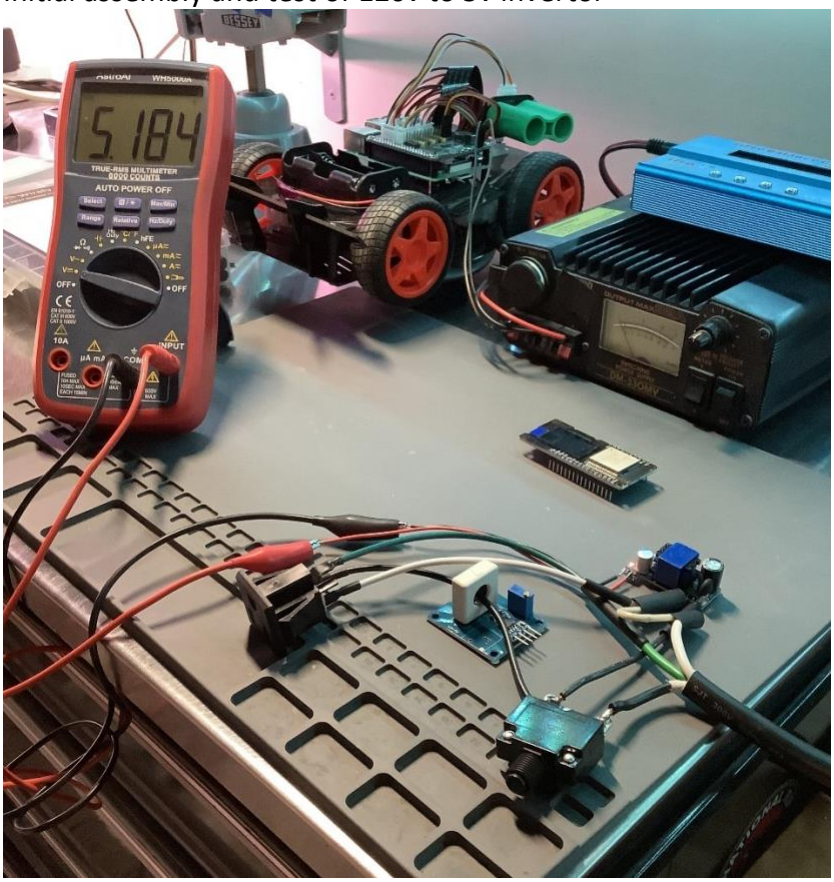
Box construction utilizing TinkerCad 100mm x 50mm. Corner blocks are for screws with holes to be drilled afterward. Circuit breaker hole was drilled afterward.



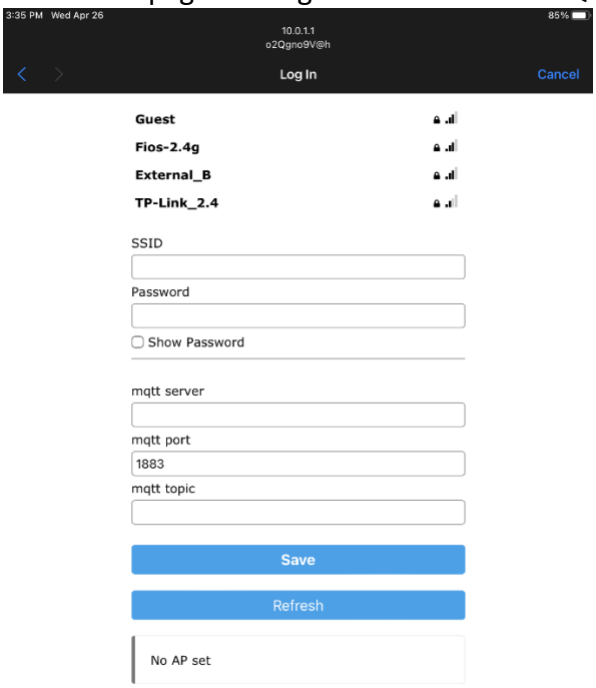
3d Printing of box. This lid version did not work and the final version in the tinkercad photo above was used. The original plan was to have the ESP32 board inside the box with the OLED screen outside however the OLED screen cable was not long enough.



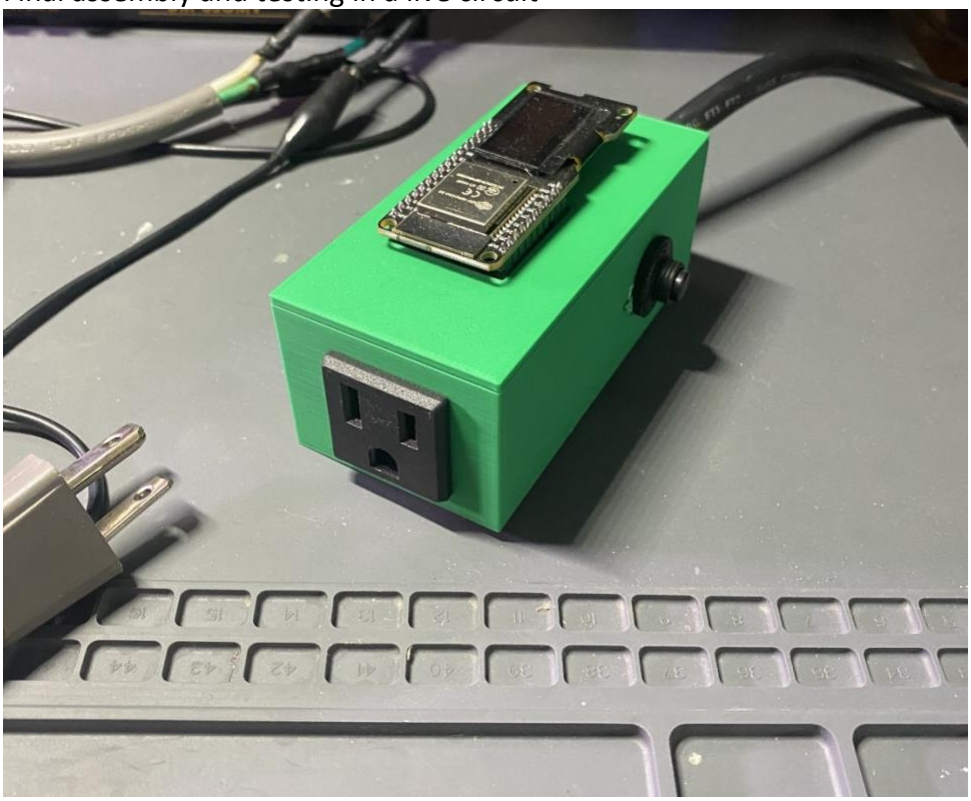
Initial assembly and test of 120v to 5v invertor



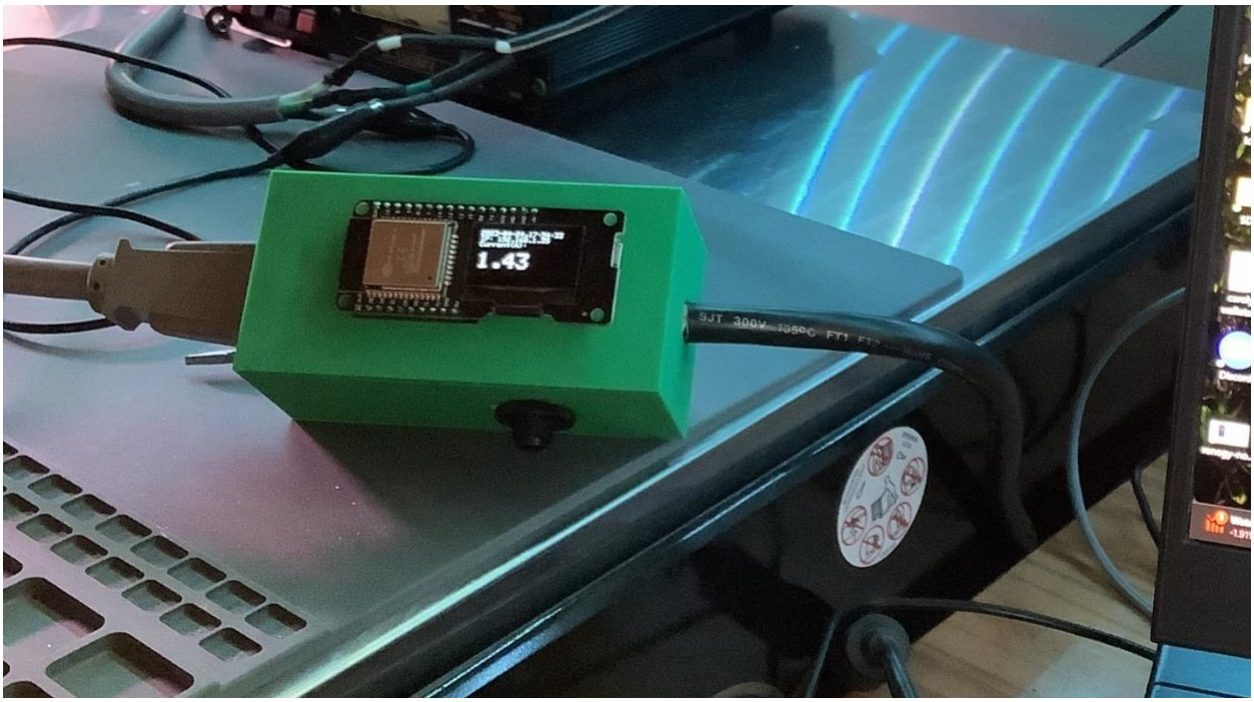
Wifi/MQTT setup manager. When the device is first powered up, it starts in AP mode and serves this page looking for network and MQTT instructions.



Final assembly and testing in a live circuit



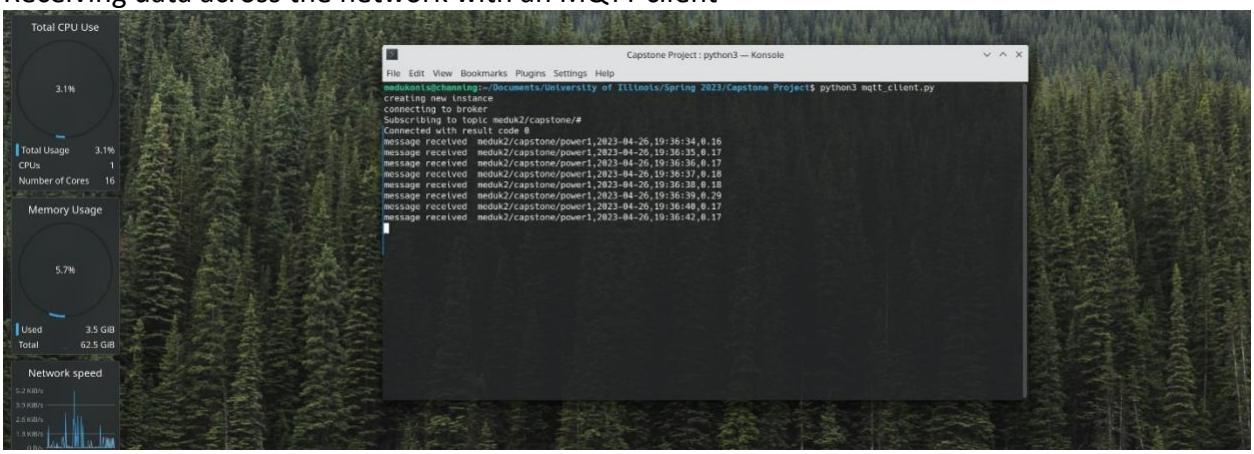

15A/250VAC
15A 32VDC
12S/250VAC


1.43

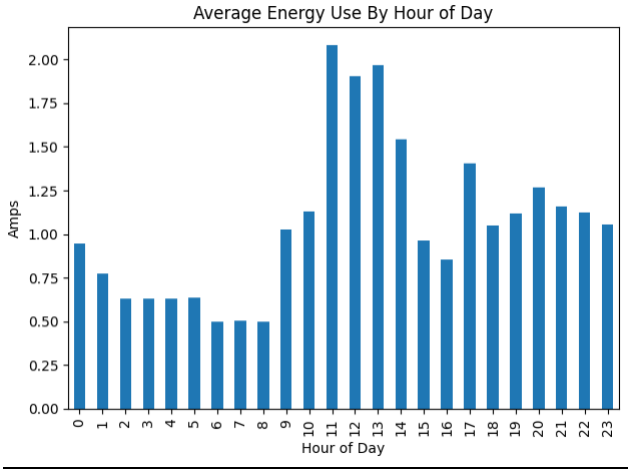
Testing and adjustment with inline and clamp ammeters.

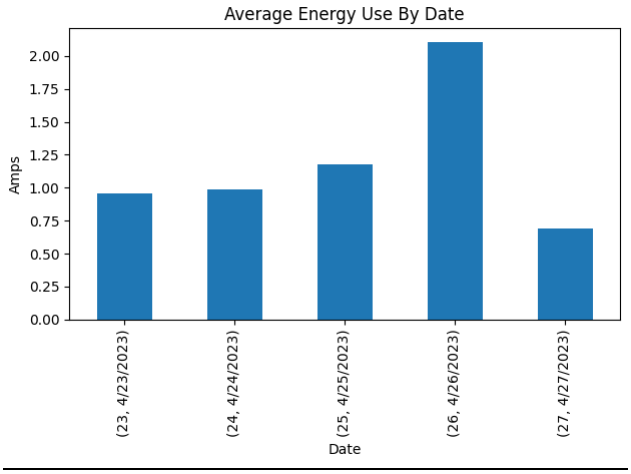

96308
20A~
HOLD
200A~
400A~
300V~
300V⎓
2000Ω
200kΩ
OFF
CEN-TECH
3129923
137
Digital Clamp Meter
COM
V/Ω/⏭
MAX 300V
CAT II
1.41
300V

Receiving data across the network with an MQTT client



Data is appended every second to a CSV file. Graphs were generated with matplotlib and pandas libraries in python. Data can just as easily be manipulated using MS Excel or imported to MySQL. Scripts that generate this analysis can be run automatically (cron job) with resulting png files moved to webserver directory.





Build Parts:

Board - https://www.amazon.com/HiLetgo-ESP-WROOM-32-Bluetooth-Development-Display/dp/B072HBW53G/ref=sr_1_3?keywords=esp32+oled&qid=1675898874&sr=8-3

Female plug end bulkhead
https://www.amazon.com/gp/product/B092VQNKN7/ref=ppx_yo_dt_b_asin_title_o03_s00?ie=UTF8&th=1

WCS1800 Hall Effect Current Sensor
https://www.amazon.com/Comimark-WCS1800-Current-Detection-Overcurrent/dp/B07XY7NT6K/ref=sr_1_2?crid=3LT7NSIO12IF9&keywords=wcs1800&qid=1675898986&sprefix=wcs1800%2Caps%2C70&sr=8-2

Inverter – 120v AC to 5v DC 2 amp
https://www.amazon.com/Converter-Universal-Isolated-Switching-Version/dp/B07SGQ6XXR/ref=sr_1_3?crid=1U69WM7VOOL7Y&keywords=120v+to+5v&qid=1675899104&sprefix=120v+to+%2Caps%2C77&sr=8-3

15A Circuit Breaker
https://www.amazon.com/gp/product/B088R8DKLP/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&th=1

Works Cited

Storr, Wayne. "*Hall Effect Sensor and How Magnets Make It Works*." *Basic Electronics Tutorials*, 6 Aug. 2022, https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html.

Hall Effect Base Linear Current Sensor - Winson. 29 Sep. 2020https://www.winson.com.tw/uploads/images/WCS1800.pdf.

WinsonLib Application Notes. Winson Semiconductor Corp, 21 July 2022, https://www.winson.com.tw/uploads/images/WinsonLib_ApplicationNote.pdf.

Tzapu. "TZAPU/WiFiManager: Esp8266 WIFI Connection Manager with Web Captive Portal." *GitHub*, Dec. 2022, https://github.com/tzapu/WiFiManager.

Craggs, Ian. "Eclipse Paho: The Eclipse Foundation." *The Community for Open Innovation and Collaboration*, (Python MQTT library) https://www.eclipse.org/paho/.