HOME APPLIANCE ENERGY MONITOR: FINAL REPORT

By

Guneet Sachdeva

Om Patel

Ravi Thakkar

Final Report for ECE 445, Senior Design, Fall 2023

TA: Zichang, Ma

04 December 2023

Project No. 25

Abstract

This is the final report for the Home Appliance Energy monitor that was created as part of the University of Illinois at Urbana Champaign ECE 445 Senior Design project. In this project we create a device that can be plugged into the wall and have an appliance be plugged into our device to be able to read the appliances power usage. This power usage will be communicated to the consumer and give the consumer enough information to be able to decide if weather they should turn off the device or not. The consumer will communicate via a GUI using Bluetooth. The device will also use Bluetooth to send information to the GUI. This report details the ideas we implemented, justification for the design decision we made, test results of our product, and reflections on what we achieve and could do better.

Contents

1. Introduction

1.1 Background

As a technologically modern world, we have a lot of home appliances that are consistently reliant on a lot of electricity. However, we tend to overuse these devices, thus leading to dangerously high electricity usage. An average of 34% percent of electricity at the household is wasted. This problem would become more apparent to users if they were able to visualize and track their electricity consumption for each home device because many users are unaware of their electricity consumption patterns and do not realize the extent of wastage that occurs.

1.2 Problem

Due to convenience, entertainment, and forgetfulness, the average modern lifestyle does not spend time being more mindful about their electricity consumption. Even though there are quick actions that can be taken to reduce the consumption of electricity, individuals cannot make informed decisions. For example, homeowners may see high utility consumptions but don’t exactly know what appliances are disproportionately contributing to the total electricity consumption amount.

1.3 Solution

The solution we will be implementing is a Bluetooth-enabled electricity monitor for home appliances. This monitor would track electricity consumption for the connected device over a period. There would be a microcontroller to process the values from the sensors and handle communication. An app would be made to display the results and send notifications to users if a certain device is consuming dangerously high amounts of power. The user will then be able to turn on or off the home appliance via the app based a warning from the app of high energy usage, long term energy usage, or personal preference.

2 Design

Now that we had figured out what project we wanted to do, we had to create a design. We knew our project needed to calculate the average power consumption, therefore we needed to calculate the current and voltage levels of our home appliance. We also knew we wanted the real power, as this is the power that is being consumed by the appliance at the time of taking the measurement. To get the average power, we would need the average current, average voltage, and the power factor, PF, being used as shown by equation 2.1.

$$P_{avg} = I_{RMS} \cdot V_{RMS} \cdot PF \tag{2.1}$$

The reason we use the RMS values is because this is the average of current and voltage coming from the AC power source, the outlet, in DC form. The current and sensor readings would be taken care of by the sensor subsystem. The sensor subsystem would then send the readings to our microcontroller subsystem where it can process and evaluate the data. Our sensor subsystem also contained a temperature sensor that would have read the appliances temperature but was decided that it was no longer needed as the typical consumer would not know how to install the temperature sensor on to their home appliance. These two subsystems and were originally powered by a low power subsystem, which would have used a 5-volt battery. Reason being 5 volts as the sensors need 5 volts to power on, and we can step down these 5 volts to 3.3 volts to power our microcontroller. In our final design, we decided to get rid of the 5-volt battery, and power everything using the high-power subsystem, which is connected to the wall outlet. Our design also will take the high power received to the high-power subsystem, and route it back out to the home appliance. Another small change we made was getting rid of the 3.3 volt to 5 volt boost and replacing it with a BJT transistor which we will discuss more about in the subsystem. Here’s our original and final block diagram showing the design decisions and changes:

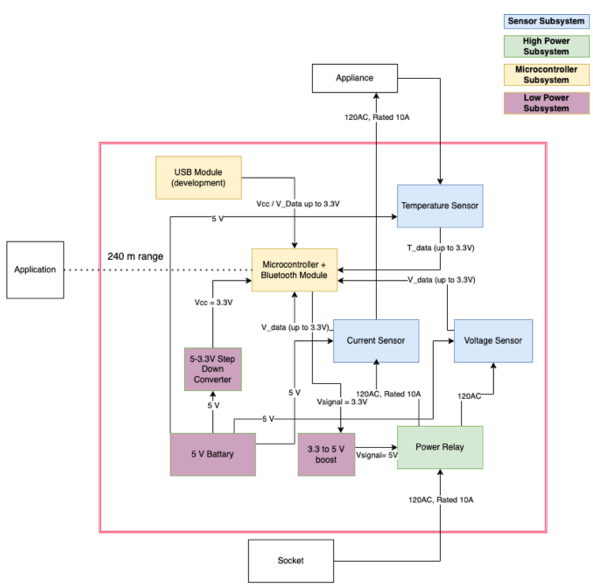

Figure 1 Original block diagram with low power subsystem and temperature sensor

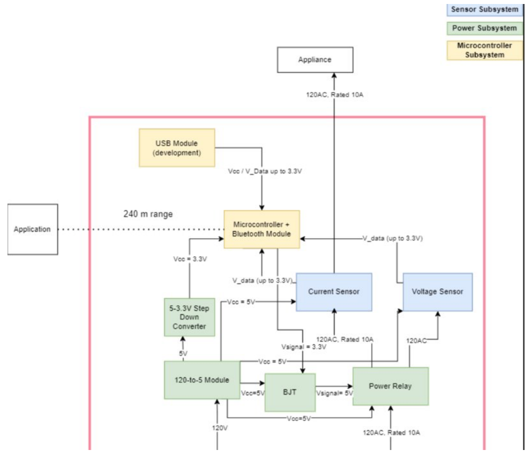

Figure 2 Final block diagram

2.1 Visual Aid

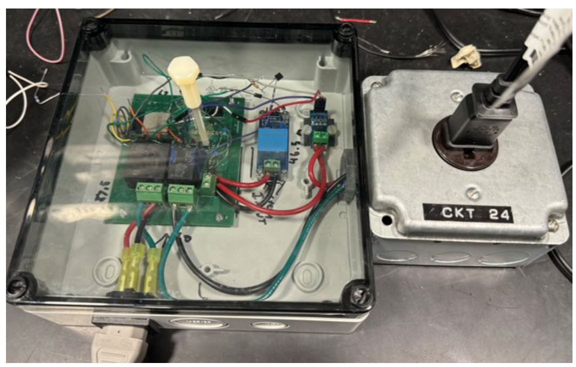

Figure 3 Visual Aid

2.2 Power Subsystem

This subsystem is responsible for powering the components on the board as well as controlling power supply to the appliance. There will be one 120 to 5 V transformer (LRS-35-5) to convert 120 ACV to a 5 DCV that can be used to power the sensor subsystem. Since we are using a microcontroller that is powered by 3.3 volts, there will also be a 5 to 3.3 V (AMS1117-3.3V) voltage regulator to control power going into our microcontroller. We also got rid of the 3.3 to 5 V step up that would turn the 3.3 V signal from the microcontroller to a 5 V signal for the power relay. Our final design did not include this since the 3.3 V signal from the microcontroller was a 2.5 V signal by the time it reached the step up. This was due to internal resistance that caused voltage drop. This 2.5 V input into the step up was not enough to produce the 5 V output needed to control the power relay. We replaced this step up with a transistor and that used the signal from the microcontroller to open or close 5 V access for the power relay. Our final component was the power relay (Omron g5le-14 5vdc) that interacts with our microcontroller to determine if the connected appliance can continue to draw supply voltage or not. This power relay is a single pole power relay which means that when it is not sending power to the appliance, the circuit in the power relay will be an open circuit and not trying to send power through another pin. The power relay also will need a flyback diode to prevent inductive flyback when switching the relay off.

2.3 Sensor Subsystem

This subsystem is responsible for measuring the voltage and current draw of the connected home appliance. A voltage sensor (ZMPT101B) and a current sensor(ACS712ELCTR-20A-T) will be present in this subsystem. The reason we chose this voltage sensor as this was a commonly used module and was able to handle up to 120 Volts since this is the standard current provided by North American outlets. Inputs of the voltage sensor module include V_{cc}, *GND, L* and *N*. V_{cc} is fed in 5 V, *GND* is same ground as rest of circuit board, *L* and *N* is live and neutral which is fed in through the power relay output. The output of the voltage sensor, *Output,* is an analog signal, therefore we connected the output of this sensor to a compatible ADC pin on the microcontroller. The current sensor we chose is a hall-based sensor, which is more simpler and came in as a chip. Shunt based sebsor, we'd have to add more components and it would increase the complexity of using it. We did not want to make that as complex since we were dealing with high voltage and could not afford risks especially since both would have nearly the same accuracy in our use case. Hall base accuracy does not drop since temperatures are not going to high, and there is no magnetic interference. We made sure to pick a 20-amp rated sensor since this is the standard current provided by North American outlets. The input to the current sensor included V_{cc}, *Filter, GND, IP+.* V_{cc} is fed in 5 V, *GND* is same ground as rest of circuit board, *Filter* was used to create a RC circuit to set bandwidth. This bandwidth is set to improve signal-to-noise ratio. The sensor also took in *IP+* which was the live wire going to the application. Output of the sensor was *IP-,* which is live going back out to the appliance, and *Output,* which is a DCV signal that correlates to the current being read. We did have a temperature sensor in our original design but got rid of it since we did not want the user to have to worry about installing the sensor into the home appliance. The voltage and current sensors will connect their output reading to the microcontroller subsystem as the microcontroller will be responsible for processing the values these sensors will be providing. The voltage sensor is connected in parallel to the home appliance load, and the current sensor is in series with this same load.

2.4 Microcontroller Subsystem

The microcontroller subsystem will be responsible for handling the turning on and off the power relay, calculating the power value, and for relaying the power consumption information to our app via Bluetooth. The specific microcontroller we will be using for this subsystem is the ESP32. We chose the ESP32 since it had analog compatible pins (ADC pins) and Bluetooth compatibility. This subsystem was provided through ECE 445 Senior Design course website, which setup the GPIO strapping pins for us. These pins were responsible of putting the microcontroller in various modes such as boot mode and reset. The microcontroller will be powered with 3.3 volts via the power subsystem. The microcontroller also interacts with the power subsystem by sending signals to the transistor which will then give the single for the power relay to turn on and off. The microcontroller will be getting input from the sensor subsystem by reading in voltage and current values periodically to calculate how much energy is being consumed by the connected appliance. The voltage and current sensors output an analog signal, so we connected the output of that voltage sensor to pin 12 and current sensor to pin 14 on our microcontroller which is an ADC supported pin and can read analog signals. Once the microcontroller reads in the values and calculates the energy being used, it will send that data out to our front end via Bluetooth low energy, BLE.

2.4.1 Current Calculation

For the current sensor output, we first convert the analog signal we receive on pin 14 to a digital equivalent and then feed the value into equation of graph shown in figure 4 to get equivalent current value. The equations to do this are equations 2.2 and 2.3.

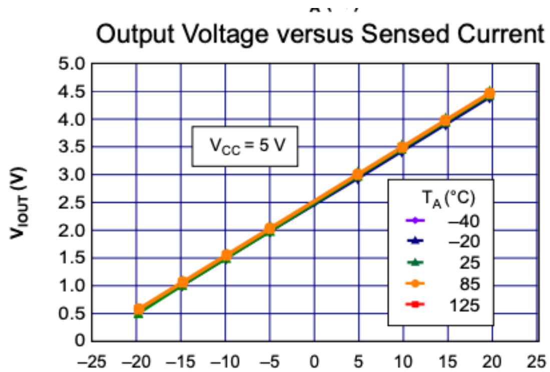

Figure 4 Current Sensor Graph

$$I_{Digital} = (\text{Pin 14 value}) \cdot (5/1023) \tag{2.2}$$

$$I_p = (2.5 - I_{Digital}) / .185 \tag{2.3}$$

2.4.2 Voltage Calculation

For the voltage sensor output, we again read the analog signal from pin 12, and convert to digital equivalent. This is done by equation 2.4 by multiplying by 3.3 as we have to multiply pin 12 value by the maximum voltage signal our microcontroller can take. Then we take that Digital value and use it in equation 2.4.

$$V_{Digital} = (\text{Pin 12 value}) \cdot 3.3 \tag{2.2}$$

$$V_p = V_{Digital} / 4095 \tag{2.4}$$

3. Design Verification

Our design verification used our requirements and verification tables we created in our design document that helped test each subsystem capability. These tests helped test if we are getting the intended behavior or if we can control the behavior of the system. The requirements and verification table we show what requirement we want to fulfill. The verification then gives a brief description of how we can verify if we are getting the behavior and if the test was a success or not.

3.1 Microcontroller Subsystem

Table 1 is our requirement and verification table for the microcontroller subsystem.

Table 1 Labor Costs

Requirement	Verification
We want to be able to read in values every 5 seconds.	This requirement can be confirmed via our front end app. Since our current and voltage sensor were not working, we sent placeholder power values from the firmware. If the ESP32's BLE worked properly, it would send a new power value point to our front end every 5 seconds. Our front end did indeed receive a new point every 5 seconds.
The second requirement for this subsystem is the ability to control the power relay.	When the appliance on button is clicked via our front end, the attached appliance should turn on. When the appliance off button is clicked via our front end, the attached appliance should turn off. We tested this functionality, and the fan responded properly.
The last requirement is the ability for this microcontroller to alert users if power consumption is too high.	We were not able to verify this requirement. Proper functionality of our current and voltage sensor were required to make sure if power consumption was exceeding a threshold. However, those sensors did not work, so we could not utilize the microcontroller to send immediate alerts regarding power consumption being too high.

3.2 Sensor Subsystem

Table 2 is our requirement and verification table for the sensor subsystem. Although our sensor subsystem did not end up working, we have provided a secondary table showcasing our test results of measuring the current and sensor using lab equipment. We used a current clamp and clammed it to the input to our current sensor. We then used voltmeter cables to measure the voltage using the input to our voltmeter. This secondary table is table 3 and it helps show that our circuit with a working sensor subsystem would be able to detect current and voltage changes given power relay state change and changes in the load.

Table 2 Labor Costs

Requirement	Verification
The requirements for the current and voltage sensor are to just send the measured voltage and current values properly to the microcontroller.	We can confirm this by seeing that whatever value is measured by the sensor is received by the microcontroller. We were not able to verify this requirement. The current sensor was burned due to a voltage arc situation. The voltage sensor did not respond properly as well after that. Therefore we were not able to check if these sensors were measuring current and voltage in the right manner.

Table 3 Fan Tests and Results

Fan Level	Power Relay State	Voltage	Current	Power
1	On	120 V	242.5 mA	29.1 W
2	On	120 V	312.5 mA	37.5 W
0	On	120 V	0 mA	0 W
1	Off	0 V	0 mA	0 W

3.3 Power Subsystem

Table 4 shows our power subsystem requirements and verification.

Table 4 Labor Costs

Requirement	Verification
The requirement for the 120 V to 5 V transformer is to properly convert 120 Volts to 5 Volts and then power sensors with the newly converted 5 volts. The requirement for the 5 to 3.3 V transformer is to properly convert 5 volts to 3.3 volts and then power the microcontroller accordingly.	We can confirm all these values by using a voltmeter to test voltage values at different parts of our circuit. We verified this result by testing the voltage of the input into the transformers and then voltage coming out of the transformer. The 120 to 5 transformer did indeed output 5 volts. The 5 to 3.3 transformer did indeed output 3.3 volts. Additionally, the sensors and microcontroller were getting powered properly as well.
The power relay's requirement is to allow the voltage supply value through if the microcontroller allows such to happen and to not allow any voltage supply through if the microcontroller indicates that to happen.	We can test this requirement through again using a voltmeter to test voltage values going through the power relay. When we turned the power relay on via the GUI, the Vout was 120 Volts. When we turned the power relay off via the GUI, the Vout was 0 volts. Therefore, the

3.4 Circuit Diagram and Schematic

In figure 5 and 6, we have provided a circuit diagram of our design and the schematic of our parts and their connections.

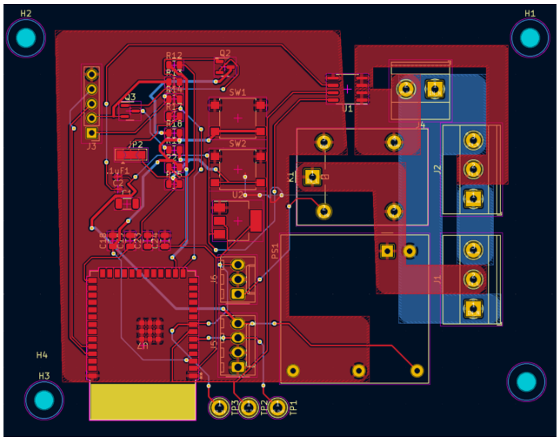

Figure 5 PCB Diagram

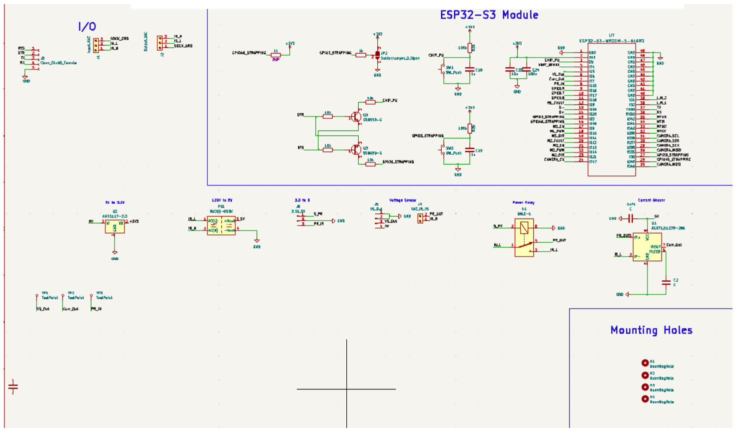

Figure 6 Schematic

4. Costs and Schedule

We have included what the cost would look like in the industry. We factored in cost of parts, labor costs, spare parts, and tax. We believe this would reflect realistic cost of this project.

4.1 Parts

In Table 5, we have included all the parts that we used for this project. The quantity has been increased to make sure we have spare parts in the case one of the parts don't work or get damaged. The unit of these parts are multiplied by the quantity to get total cost for each part. Parts at the bottom of the table that have "Free (ECE)" in the cost column indicate that these supplies were provided by the Electrical and Computer Engineering department at the University of Illinois at Urbana Champaign.

Table 5 Parts Costs

Part	Manufacturer	Part #	Quantity	Total Cost
Microcontroller	Espressif	ESP32-S3-WROOM	2	$6.40
Voltage Sensor	Noyito Technologies	ZMPT101B	2	$13.98
Current Sensor	Allegro Microsystems	ACS712ELCTR-20A-T	3	$11.37
120-5V Transformer	Mean Well USA	RAC10-05SK/277	2	$28.24
Power Relay	Omron Electronics	Omron g5le-14 5vdc	2	$3.04
5-3.3V Regulator	Advanced Monolithic Systems	AMS1117	2	$8.66
Terminal Connector	Pheonix	277-1263-ND	3	$17.16
PCB	PCB Way	N/A	20	$10
Extension Cord	N/A	N/A	1	Free (ECE)
Resistors (all)	N/A	N/A	7	Free (ECE)
Capacitors (all)	N/A	N/A	4	Free (ECE)
Switches	N/A	N/A	2	Free (ECE)
Transistor(all)	N/A	N/A	3	Free (ECE)
Total				**$98.85**

4.2 Labor

Table 6 is provided to showcase labor costs. We chose to give each member of the team a salary of $52,000. We got this number by getting the expected income out of college for our major, $104,000, and dividing it in half to reflect a half year, in our case one semester, long project.

Table 6 Labor Costs

Member	Salary
Guneet Sachdeva	$52,000
Om Patel	$52,000
Ravi Thakkar	$52,000
Total	**$156,000**

4.3 Total Cost

Table 7 puts total cost from section 4.1 and 4.2 into one table to produce the total cost of this project.

Table 7 Labor Costs

Description	Cost
Parts	$98.85
Tax (8%)	$7.90
Total Member Salary	$156,000.00
Total	**$156,106.75**

4.4 Schedule

Table 8 provide us with the schedule we followed based on the courses guidelines. We made this schedule by first taking look at the assignments that we believed would take the longest and set start and goal dates for them. Then went ahead and planned around that schedule for the small tasks. We have also added tasks that the course does not require but more for our team to keep track of when things should be done i.e. Start Firmware. Some start and end dates have "…" to showcase that this should be an ongoing task after the start or before the end. We divided the schedule based on what each team member was going to specialize in for this project. Ravi specialized more in firmware/front end, Guneet specialized on firmware and parts, and Om specialized more on circuit design, assembly, and testing.

Table 8 Schedule

Tasks	Start Date	Goal Date	Individual Contributions
Finish Design Doc	9/24	9/28	Ravi, Guneet, Om
Finalize Parts to be used	9/28	9/30	Ravi, Guneet, Om
PCB first Draft	9/28	10/1	Om
Review PCB/Container design with Machine Shop	10/4	10/8	Ravi, Om
Get first round PCB order in	10/6	10/10	Om
Last machine shop revisions	10/9	10/13	Ravi
Team Evaluation	10/10	10/11	Ravi, Guneet, Om
Update PCB design	10/11	10/15	Om, Guneet
Review PCB design with Machine shop/TA	10/14	10/17	Om, Guneet
Begin Firmware	10/17	10/23	Ravi, Guneet
Solder/Test PCB	10/18	10/22	Om, Guneet, Ravi

Review PCB design with Machine shop/TA	10/19	10/23	Guneet, Om
Individual Progress Reports	10/21	10/25	Ravi, Guneet, Om
FINAL PCB ORDER	10/23	10/24	Om, Guneet
Solder/Testing	10/24	…	Om, Guneet
Firmware	10/24	…	Ravi, Guneet
Create App	10/24	…	Ravi
Prepare draft demo	11/9	11/12	Ravi, Guneet, Om
Mock Demo	11/13	11/17	Ravi, Guneet, Om
Team Contract Fulfillment	11/15	11/17	Ravi, Guneet, Om
Prepare Final Demo	11/15	11/29	Ravi, Guneet, Om
Final Paper	11/17	12/6	Ravi, Guneet, Om
Final Demo	11/27	11/29	Ravi, Guneet, Om
Prepare mock presentation	11/28	11/30	Ravi, Guneet, Om
Mock Presentation	11/30	12/1	Ravi, Guneet, Om
Prepare Final presentation	11/29	12/3	Ravi, Guneet, Om
Final Presentation	12/4	12/6	Ravi, Guneet, Om
Lab Checkout	…	12/7	Ravi, Guneet, Om
Lab Notebook due	…	12.7	Ravi, Guneet, Om

5. Conclusion

5.1 Accomplishments

The first accomplishment we had was safely utilizing the socket voltage and current to power our board. Our power subsystem's transformers took care of this task. The second accomplishment we had was controlling the power supply to the connected appliance. Our power relay functioned properly and responded to commands sent via the microcontroller. Lastly, we were able to harness the BLE capabilities of the ESP32 and send information to the front end application via Bluetooth Low Energy. Therefore, we were able to gain success with our power and microcontroller subsystem.

5.2 Uncertainties

We faced challenges with our voltage sensor, experiencing unexpected behavior despite configuring it correctly with appropriate power levels in the designated location. Surprisingly, the sensor was consistently outputting 2.5 Volts, the default value if nothing is being inputted. We think this sensor could have been burned when the current sensor burned. However, we are unsure if this really is the case.

5.3 Future work

We have multiple ideas on how to refine our project for the future. The first thing we would consider doing is making our own voltage sensor. We had trouble understanding how our voltage sensor worked so making our own voltage sensor would have been a better option in this situation. Additionally, it would have been more wise to use a shunt-based current sensor. The reasoning behind that is that shunt based current sensors offer wider current ranges and direct measurement. Lastly, we need to add a diode after the power relay, to dissipate the excess charge. The excess charge is known to cause voltage spikes which would lead to circuit malfunctions.

5.4 Ethical considerations

Dealing with high voltage required us to take extra precautions during the testing and building of our appliance. We had to ensure that all team members were aware of the dangers of high voltage and had undergone proper training in electrical safety. Work was only done on the circuit when it was not plugged in. Additionally, when dealing with live wires, we implemented a strict grounding protocol for all power sources. There were always multiple of us present during lab work, and whoever was there was using protective equipment (gloves and ESD equipment). Handling different tools in the lab also required some precaution. All of us were trained on the proper usage of each tool and the proper storing of the tool after use. In terms of electrical safety, we carefully read the components' ratings and took notes of the characteristics to ensure compatibility of each component amongst each other. The design having proper insulation, grounding, and compliance with standards was also taken quite seriously. The last piece of safety and ethics was in regards to wireless communication and application safety. All components were checked to see if they were in compliance with wireless communication standards. Lastly, any data being used by the application was secure and protected. All the measurements done by us were in accordance with IEEE 1459-2010.

References

[1] "IEEE Standard Definitions for the Measurement of Electric Power Quantities Under Sinusoidal, Nonsinusoidal, Balanced, or Unbalanced Conditions," in IEEE Std 1459-2010 (Revision of IEEE Std 1459-2000) , vol., no., pp.1-50, 19 March 2010, doi: 10.1109/IEEESTD.2010.5439063.

[2] Estimating appliance and Home Electronic Energy use. Energy.gov. (n.d.). https://www.energy.gov/energysaver/estimating-appliance-and-home-electronic-energy-use

[3] M. Mahith, D. S. B. Kumar, K. C. Prajwal and M. Dakshayini, "Bluetooth Home Automation," 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT), Bangalore, India, 2018, pp. 603-607, doi: 10.1109/ICGCIoT.2018.8753094.