IoT Enabled Power Meter

Mini Project Submitted in Partial Fulfillment of the requirements for the Degree of
Bachelor of Technology
in Electronics & Communication Engineering
Under Maulana Abul Kalam Azad University of Technology
(Paper Code - EC681)

by
Rishav Dutta
10200319043
015371 OF 2019-20

Anindya Kanti Mitra
10200319045
043488 OF 2019-20

Anubhav Dutta
10200319037
040790 OF 2019-20

Arnab Das
10200319017
016794 OF 2019-20

Under the supervision of
Mrs. Bandana Barman
(Asst. Professor, Electronics & Communications Engg.)
Department of Electronics & Communications Engineering
Kalyani Government Engineering College
Kalyani, Nadia, West Bengal - 741235

KALYANI GOVERNMENT ENGINEERING COLLEGE

Kalyani, Nadia – 741235

CERTIFICATE OF RECOMMENDATION & APPROVAL

This is to recommend that the work done in this mini project titled **"IoT Enabled Power Meter"** has been carried out by Anindya Kanti Mitra, Arnab Das, Anubhav Dutta and Rishav Dutta (bearing Roll no.s 10200319045, 10200319017, 10200319037, 10200319043 respectively) under my supervision and can be accepted in partial fulfillment of the requirement for the degree of Bachelor of Technology from Kalyani Government Engineering College, under Maulana Abul Kalam Azad University of Technology for the sixth semester as of 2022. It is hereby approved and certified as a credible study in the topic presented and is carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the above mentioned degree.

It is understood that by this approval the undersigned does not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approves the project for the purpose for which it is submitted.

_____

Dr. Sukla Basu
Professor and Head
Department of Electronics and Communication Engineering,
Kalyani Govt. Engineering College,
Kalyani, Nadia

_____

Mrs. Bandana Barman
Assistant Professor
Department of Electronics and Communication Engineering,
Kalyani Govt. Engineering College,
Kalyani, Nadia

ABSTRACT

The Internet of Things (IoT) has facilitated integration of wireless communication capabilities into several devices that we use on a daily basis. IoT enables remote monitoring of many electrical, physical and environmental parameters. Electrical power consumption is one of such important parameters to be monitored. In literature, many power monitoring devices have been proposed for this purpose, but most of them are expensive, complex and lack adequate security measures. In our project **“IoT Enabled Power Meter”**, we have implemented a simple, cost effective, convenient, and secure IoT enabled energy monitoring system. All measurements are uploaded to a secure authenticated database over WiFi and the data is securely fetched and presented to the user owning the electric connection in an easy to understand way. The centralized database also allows power utility companies to monitor usage conveniently from a single interface and send bills accordingly, instead of having to monitor each meter separately. Using this project one can remotely monitor the power usage and can also detect power leakage from anywhere in the world.

Table Of Contents

INTRODUCTION

Internet of Things (IoT) is a recent revolutionary technology which consists of integration of sensing as well as communication capabilities to common things, in order to gather useful data and obtain insight to further optimize the activity being facilitated by IoT. Such IoT enabled devices can be used to monitor various important physical, electrical or environmental parameters. This information is then used to analyze, identify and solve different problems related to everyday life. Electrical power management for efficient use of electricity is one such important problem. IoT enabled power monitoring devices can help solve this problem by providing granular information about electricity consumption [1].

In the present Indian scenario, conventional electric meters supplied by electricity suppliers measure power consumption of the whole building. Consumers have no means to monitor power remotely or without going to the meter. These meters also lack networking features as well as any option to analyze data. Due to absence of any communication facility in the meter, power consumption has to be noted down manually at each meter location for billing purposes. This process itself is prone to human error or manipulation.

Using smart power meters, the power consumption by electrical appliances can be automatically measured, logged and analyzed. This power consumption data can be securely transmitted to the electricity supplier, avoiding manual billing as well as any possibility of human error. It can also be sent to the owner i.e., the user of the particular electrical connection where the meter is connected, allowing the owner to monitor his usage in real-time.

Along with data like Voltage, Current, Power, Frequency, Energy, the system can also detect power cuts and can log the data to the database.

LITERATURE REVIEW

Several research works have been done on designing an IoT based power monitor. We have gone through some of the papers and understood their designs. This has helped us design our own power monitor that is more efficient and economical.

The paper in [1] presents the development of a GSM and ZigBee based smart meter using Arduino microcontroller. This meter can measure the energy and send the information to the service provider, who can store this information and notify the consumer through SMS messages or through the internet. The consumer also can access their energy data via a dedicated mobile application or website.

However, the system is bulky and inefficient. ATMega2560, GSM Module, XBee chip, a commercial meter are used in the power monitor and those components will increase the cost of manufacturing of the whole system.

Paper [2] proposed a simple Wi-Fi based energy monitor. It has been successfully implemented and tested and the accuracy of power measurement is shown to be suitable for reliable use as a power monitoring sensor. Considering the small size, low-cost and simplicity, multiple sensors can be used to monitor power consumption of multiple electrical appliances simultaneously. The measurement data is also stored in a database which can be easily used for statistical analysis.

In paper [3], an Arduino UNO along with ESP8266 MCU based energy monitoring system is implemented. The system logs the real time energy readings on a MQTT server. It has the functionality to detect energy theft whenever someone tries to tamper with the device. The electricity cost and other metrics for a particular load can be displayed on a built-in display.

But the available third party MQTT servers are not very reliable when data security and stability is taken into consideration.

Paper [4] also proposed a commercial meter based energy monitoring system using Arduino and ESP8266 microcontrollers. It uses an ACS712 sensor module for current measurement. The system logs the measured data to a third party IoT platform's database that also provides statistical features and graphical representation of the usage details.

PROPOSED WORK

Our proposed project is called “power-iot-meter”.

By studying all the literature about the existing work in this domain, we understood the drawbacks which need to be addressed, such as, product cost, convenience, ease of deployment, security and overall user-base i.e., number of users using such projects.

We have used technologies such as Firebase Database, Firebase Authentication, Angular Framework, Espressif ESP32 microcontroller-networking using WiFi stack, Arduino Framework, PlatformIO and PZEM-004T Power Meter Module to build our project.

Our project features all the basic functionalities required for it to be a power monitoring device. It can measure voltage, power, current, energy and frequency. But, we have gone further and also extended our database to store the power-outage logs, i.e., the start time and the end time of a power-cut event.

All these measurements are shown on a very user-friendly UI on Angular Framework for a smooth user experience which is convenient for anyone who can use a smartphone or a computer or both.

The database access rules set on it to make the measurement data extremely secure and immutable. Users can only see the data sent by “their” meter after authentication and cannot modify the data by any means. Other users can’t access measurements from devices not owned by them as well.

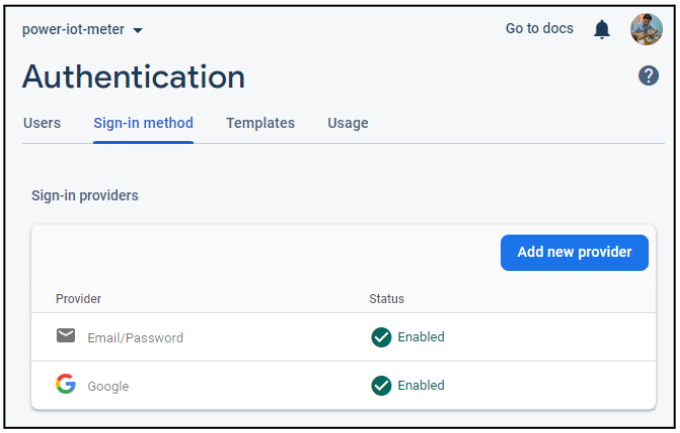
Now we shall delve deeper into each and every section involved in making this project under the Implementation section.

IMPLEMENTATION

Software Aspect:

Firstly, we are describing the database that we are using which is Firebase Realtime Database. It is an infrastructure-less database which can store a large nested json document and it is possible to update the data i.e., the key-value pairs of the json document in real time by using Firebase's RTDB SDK. The Realtime Database can also have some rules set which can control the read/write access to its documents at different levels. It is an authenticated database which uses Firebase Authentication for authenticating (signing up/ signing in) users to the application.

Firebase Authentication allows users to sign up or sign in through multiple ways (providers). The native providers are, Anonymous, Email/Password, Phone and the additional providers are, Google Sign In, Facebook Sign In, GitHub Sign In, Apple Sign In, Microsoft Sign In and many more.



In our project, we are using Google and Email/Password as the providers for authentication. Users can use both of the providers whereas devices explicitly use the Email/Password authentication only. The unique Chip ID of the microcontroller (derived from its MAC address) is used as the username of the device. Once a user or device is registered/signed up, a UID or Unique ID is generated against each user or device.

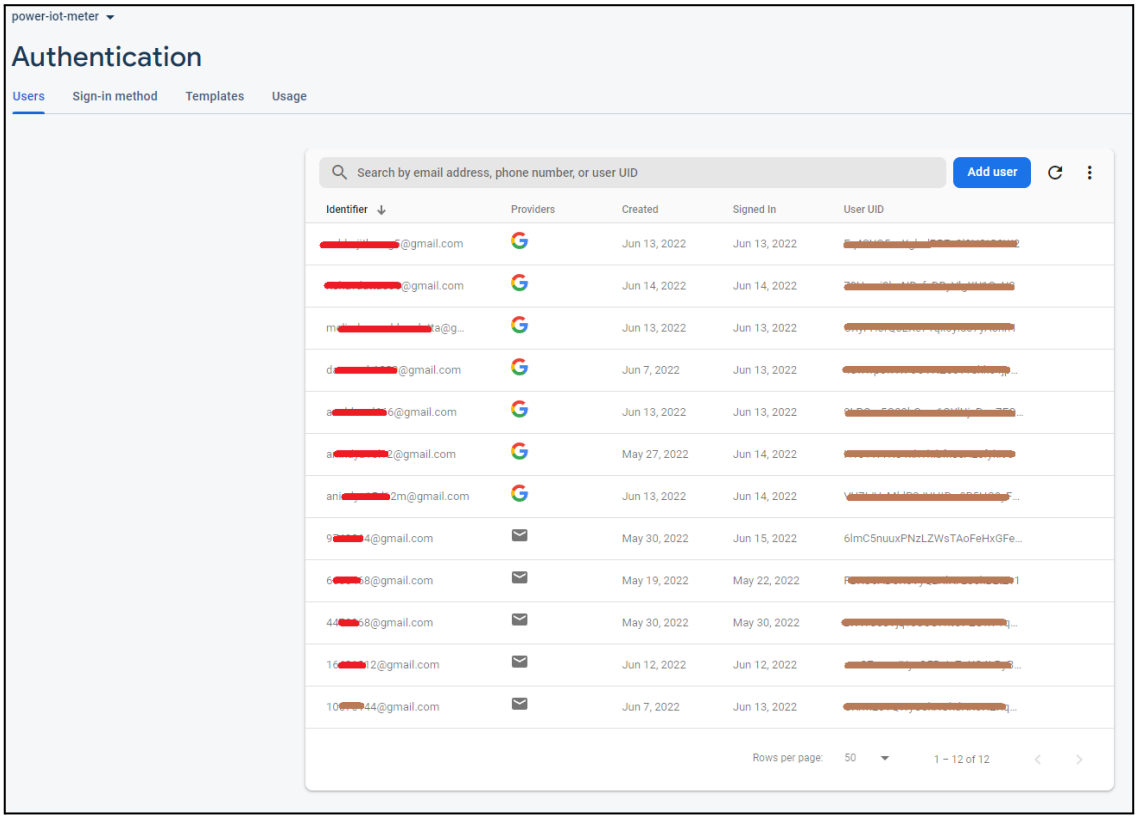
Now, in the database, rules are set like this:

Starting with "users", every authenticated user can read the list of users present in the database but only the admin can add/update/write in the "users" list.

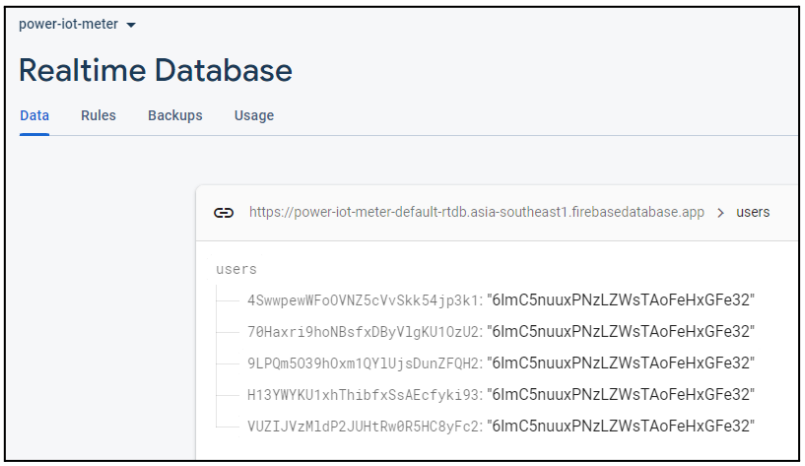
The measurements pushed from the

devices are stored under their respective UID in the document. Only the respective device can store under its own UID as described by rule: auth.uid == $uid. However, the user listed in the “users” can only view the measurements under the UID of the device he/she owns as described by the rule: data.parent().child('users').hasChild(auth.uid). A user can only own one device at a time.

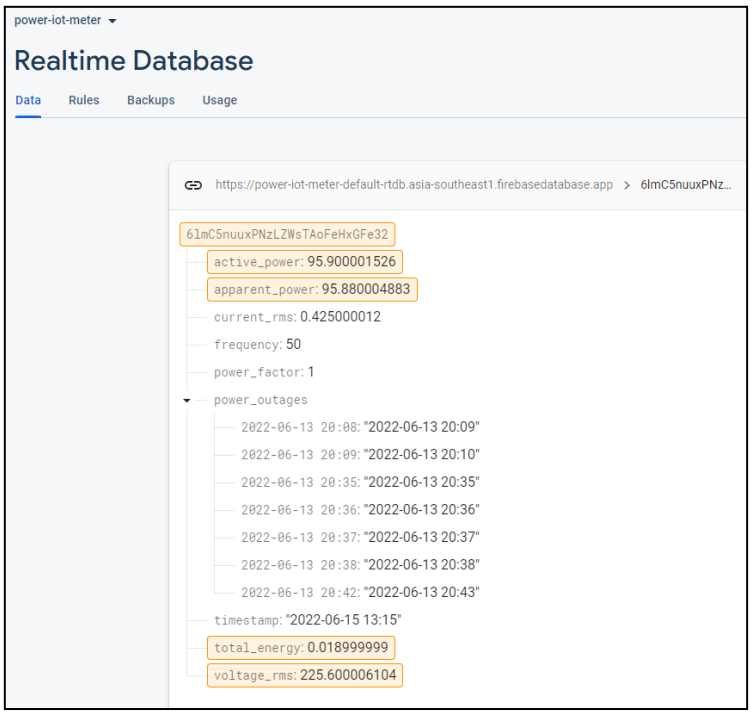
Here is a snap of the authenticated users in our project.



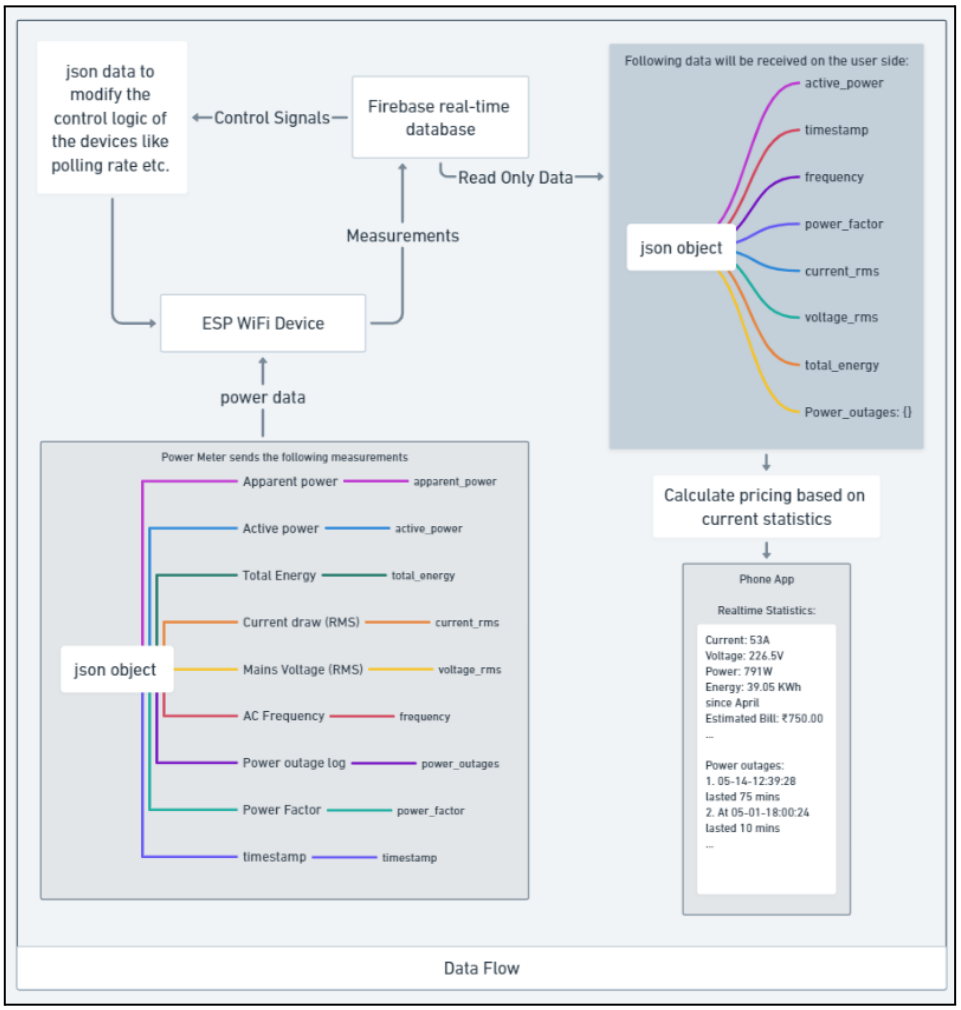
Here is a snap of the “users” list in the database:



Here is a snap of the measurements stored under the UID of a working device:

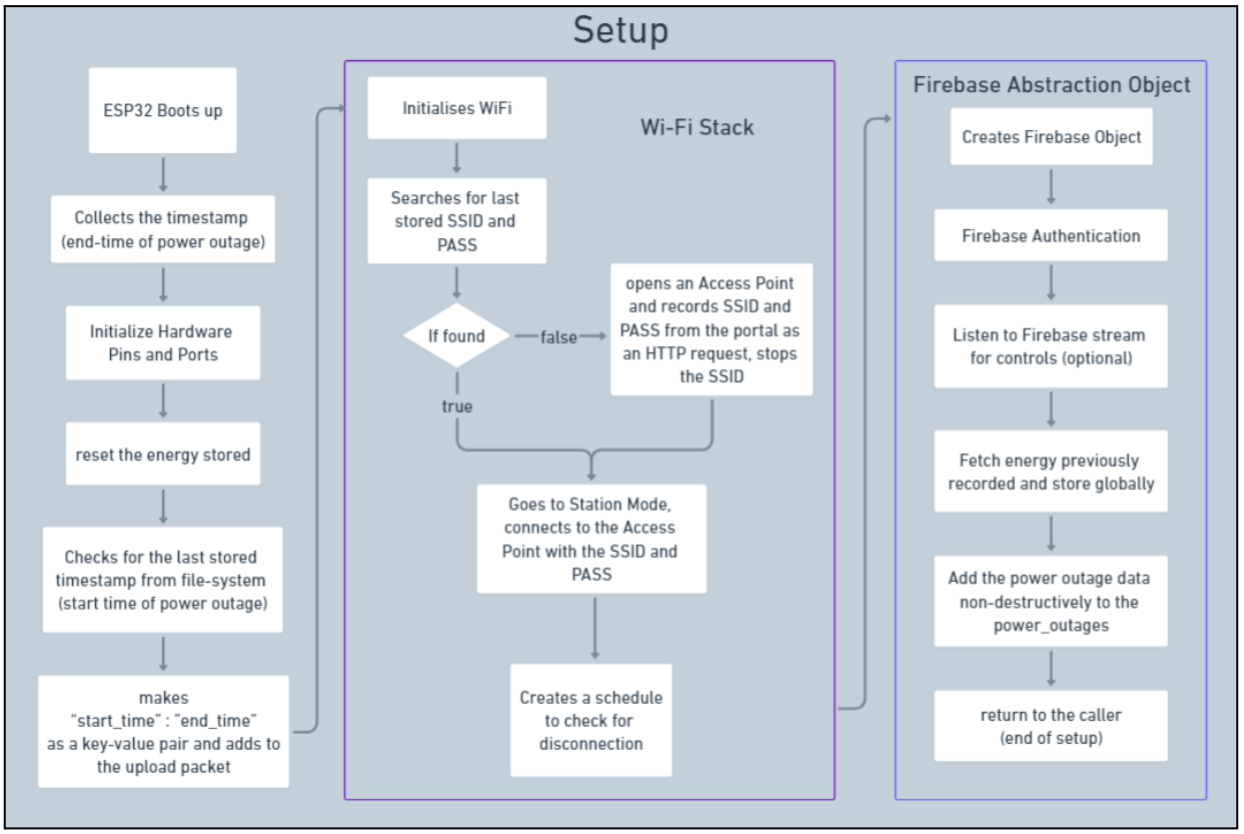


The data flowing from device to the user:



The firmware logic can be explained by the flow diagram below. It has two sections, one that runs once during bootup and the other one that runs in a loop:

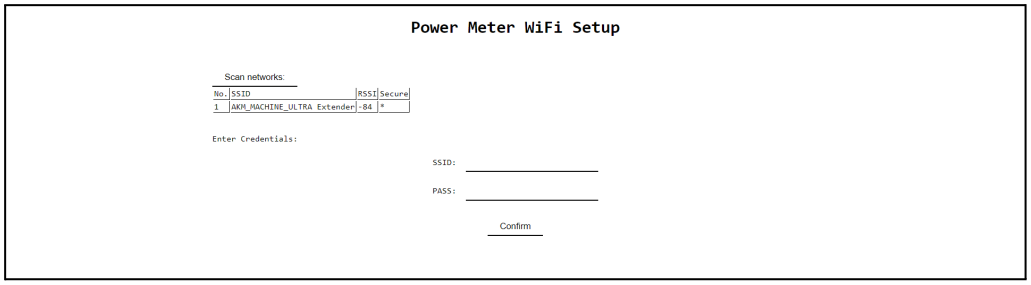
In the setup, as ESP32 boots up, it initializes Hardware Pins to connect with other devices. Then, it initializes the WiFi connection to automatically configure it to station mode and connect to the saved access point or AP.



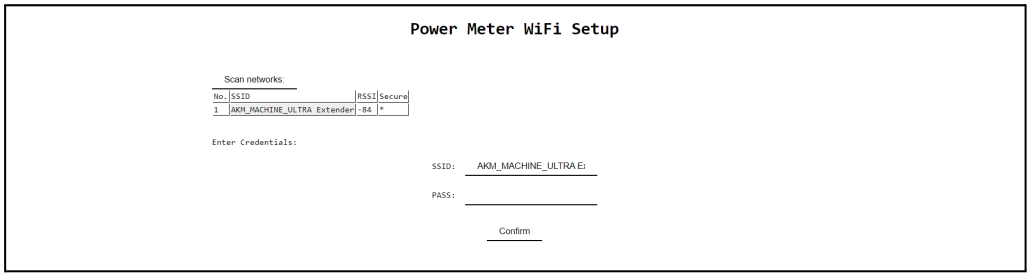
If the ESP32 doesn't have saved AP credentials, it automatically configures itself to AP mode where it creates an access point or WiFi hotspot to allow users to connect to. As a user connects to tht AP, they can type the IP address 192.168.4.1 to fetch the configuration webpage from the flash memory to scan WiFi stations, configure and save the WiFi credentials.

Here is how the page looks like:

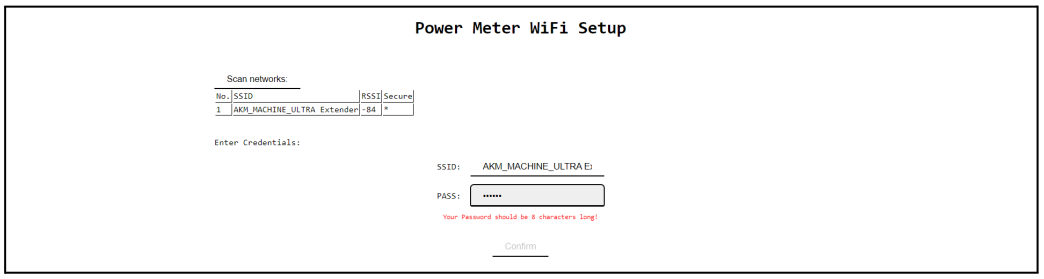
After scanning stations:



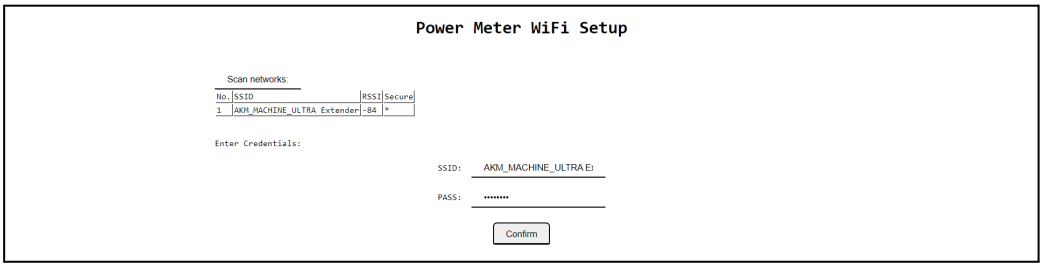
As a user clicks on a scanned station and SSID field is automatically filled:



Webpage features Password validation:



As the user presses the confirm button, a request is instantly sent to the ESP32's local HTTP server and the configuration is saved inside it.



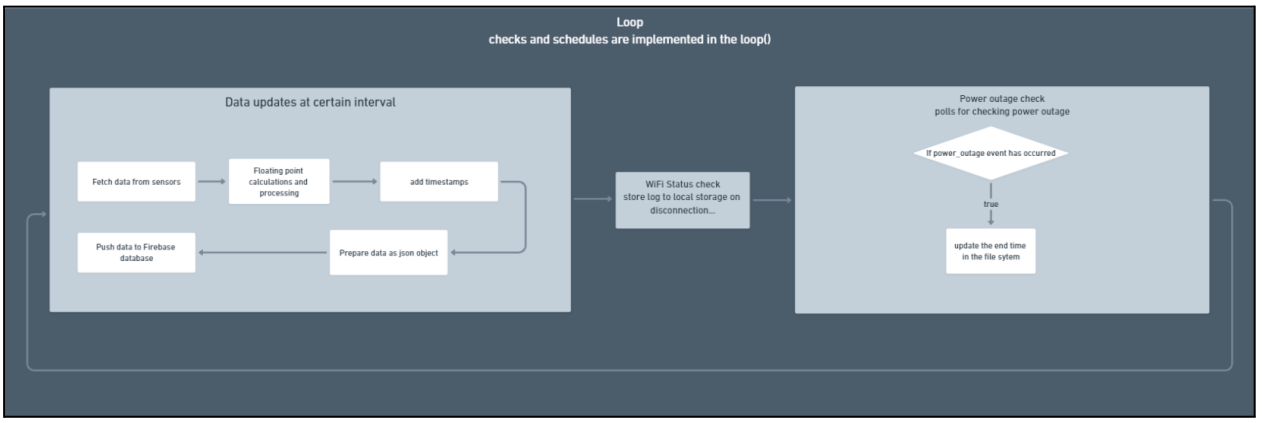
After the configuration is saved, the ESP32 attempts to connect to the configured SSID and Password. On a successful connection, it connects to the Firebase database to sign-up, authenticate and finally connect to the database. After connection to the database, the scope of setup ends and the loop begins to be executed.

In the loop, a schedule is run, which does the following things:

1. Fetches the Sensor Data.
2. Calculates various parameters and stores it in a C struct.
3. Prepares a Json document and adds the data in the struct.
4. Adds timestamp in "YYYY-MM-DD HH:MM" string format.
5. Updates the document in Firebase Database.

The loop also checks for WiFi connection status to perform certain actions like re-connection, debugging etc.

Power outage or Load shedding is also checked in the loop, whenever a power outage occurs, the ESP32 can obtain the approx. start time of the outage and store it in the flash memory. When the power is again restored, the ESP32 obtains the approx. end time of the outage, and pushes the data to Firebase when it reconnects.



Hardware Aspect:

The main challenge for the hardware was to make it simple and economic. In our prototype we have used the following components:

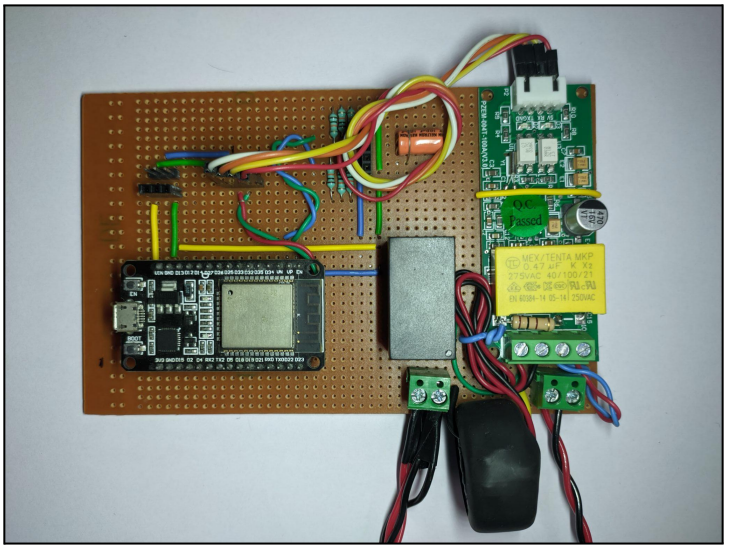
Sl.No.	Item Name	Specification	Quantity	Approx. Cost (Rs.)
1	ESP32	32bit Dual Core, 4MB Flash, 320 KiB SRAM, 3.3V DC	1	350.00
2	Hi Link Power Supply Module	HLK-2M05 5V DC 2W	1	250.00
3	Power Meter Module	PZEM004T, UART, 5V	1	1125.00
4	Perfboard	IC perfboard, dot-style	1	35.00
5	Screw Terminals	1x2 20A	1	30.00

6	Headers	Male and Female Headers 2mm	2	80.00
7	Wires	23 SWG Solid Core Rubber Coated Coloured	2	20.00
8	Capacitor	100uF 25V Electrolytic	1	4.00
9	Resistors	10k 1/4W Carbon film	6	6.00

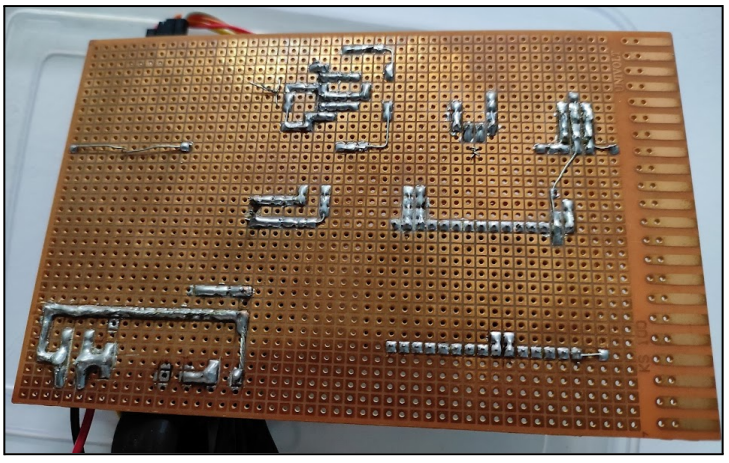
The hardware was built on the perfboard and the components were soldered with appropriate sockets, wires and connections.

Here is how the actual prototype finally looked like:

Top View:



Bottom View:

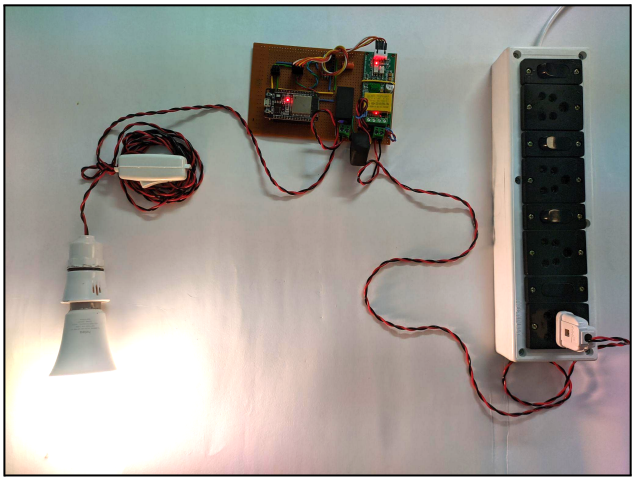


Thorough testing and troubleshooting were conducted before connecting the circuit to the AC mains supply. Electrical safety was kept in mind throughout the process.

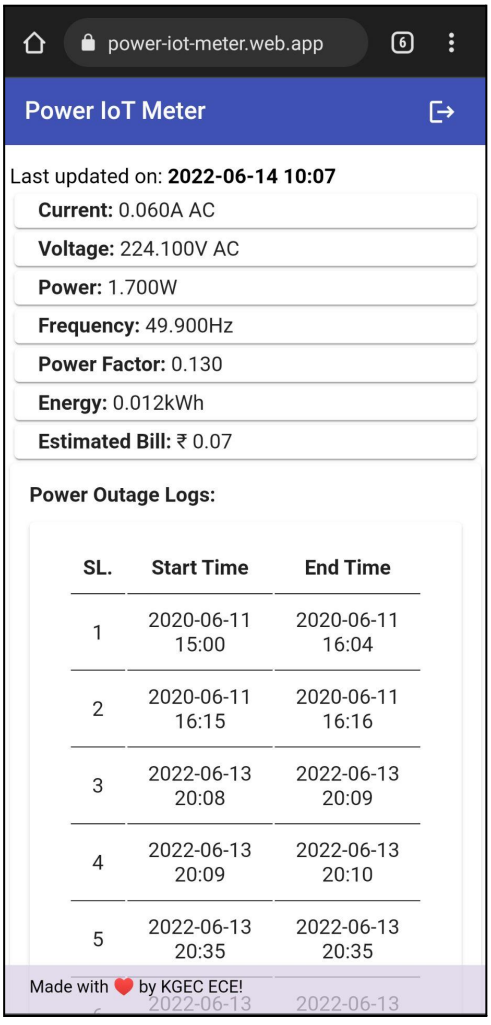
RESULTS & DISCUSSION

The proposed power monitoring system is tested with resistive and reactive loads and the observations are discussed below.

With a **reactive load** we have used a 3W LED lamp connected to the mains supply through the PZEM004T module. The ESP-32 then fetches the energy data from the module via UART and uploads it to the Firebase RTDB along with the timestamp of the readings. This data is then displayed to the user dashboard.



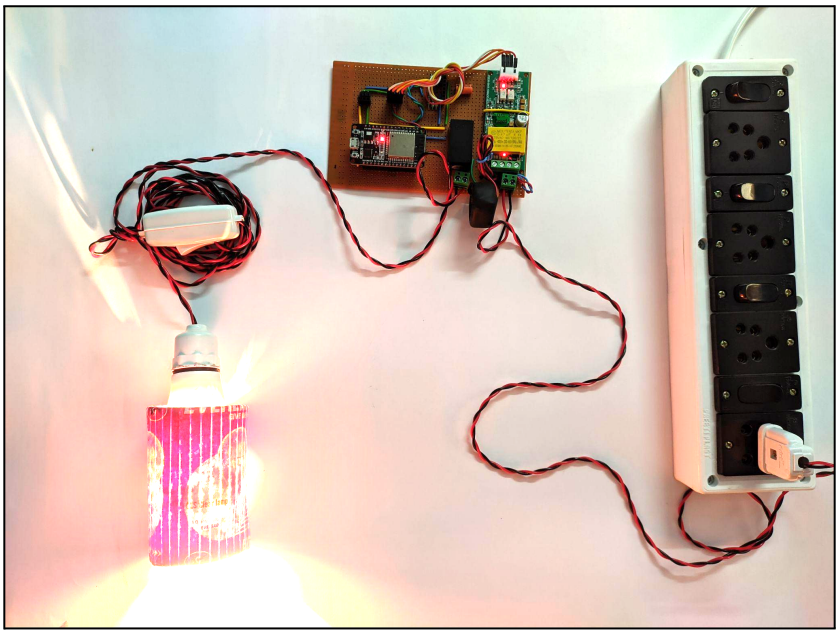
The user dashboard for the reactive load is shown below.



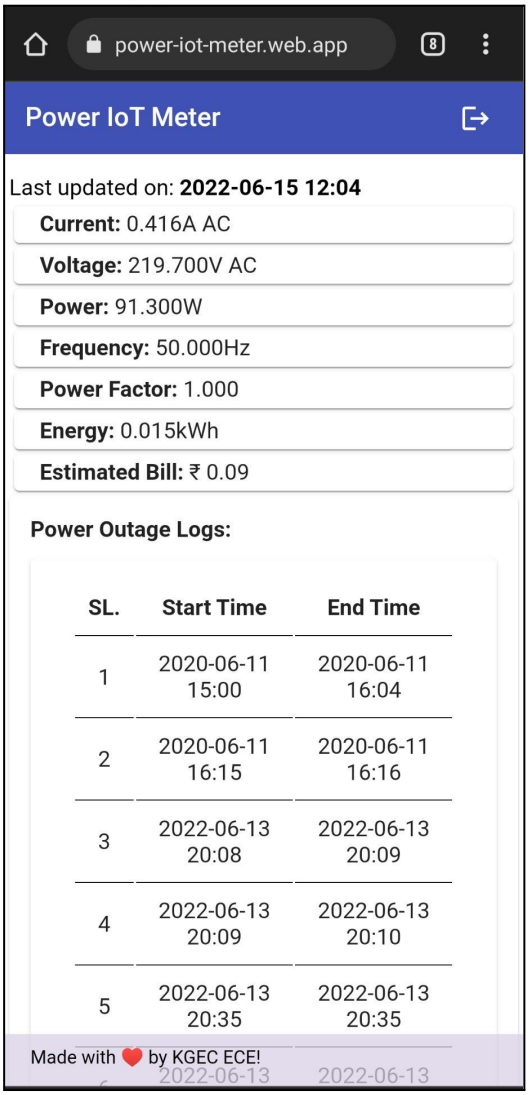
This is the user dashboard that is being viewed from a mobile phone browser. The first thing the dashboard shows is the last update timestamp. Then all the metrics are displayed and finally the power outage logs are displayed in a tabular format with power outage start-time and end-time.

In this case, the reactive load is drawing 0.060A AC current from the mains. The AC voltage across the load is 224.1V(depends upon locality). The frequency of the main supply is 49.9Hz. The rated 3W LED bulb is drawing about 1.7W of power with a power factor of 0.13. The energy consumed by the bulb is about 0.012kWh and the estimated bill for that amount of energy is ₹0.07(depends on power utility companies).

For **resistive load** we have replaced the previous lamp with a 100W incandescent lamp in the same lamp holder. The circuit was powered on and reading was taken for a few minutes.



The user dashboard for the resistive load is shown below.



All the metrics are displayed and finally the power outage logs are displayed in a very user-friendly tabular format.

As it can be observed, the mains voltage is 219.7V AC (which is a bit lower than the rated voltage of the lamp), so the 100W rated lamp draws 91.3W.

Since, an incandescent lamp is a resistive load, the power factor is exactly 1.0 . The current and energy consumed can also be viewed from the UI as 0.416A and 0.015kWh respectively.

The UI also estimates the electricity bill for the energy consumed. The consumption bill is shown in rupees.

CONCLUSION & FUTURE WORK

No device is perfect and neither is ours. Some of the potential drawbacks are:

1. Dependent on WiFi, in rural locations where WiFi connection isn't as popular, alternative technologies like LTE or LoRaWAN may be needed.

2.Our device doesn't have a display attached to allow for immediate meter readouts.

Instead of using the PZEM-004T module, analog voltage and current sensors can be used for measurement for a more exhaustive approach to accurate data collection. A custom PCB can be made that will make the device footprint smaller and compatible with existing power monitoring infrastructure. We can also integrate features like power theft detection, energy usage profiling to generate power draw heatmap for systemic scheduling or even add remote cut off functionality.

LIST OF REFERENCES

1. Anmar Arif, Muhannad Al-Hussain, Nawaf Al-Mutairi, Essam Al-Ammar, Yasin Khan and Nazar Malik "Experimental study and design of smart energy meter for the smart grid" in Conference: Renewable and Sustainable Energy Conference (IRSEC), 2013 International.
2. Akshay Ramesh Jadhav, P. Rajalakshmi in "IoT Enabled Smart and Secure Power Monitor" in Indian Institute of Technology Hyderabad, India.
3. Sagar Dadhe, Rohit Maske, Rohit Kalukhe and Meghananavare "Iot Based Smart Energy Meter" in International Journal Of Innovations in Engineering Research And Technology [IJIERT], ISSN: 2394-3696.
4. Rishabh Jain, Sharvi Gupta, Chirag Mahajan and Ashish Chauhan "IOT based Smart Energy Meter Monitoring and Controlling System" in International Journal Of Research In Electronics And Computer Engineering.
5. Vishal Kumar, Tanishq Sharma and Abu Farhan "IoT Based Smart Energy Meter" in International Journal of Engineering Research in Electronics and Communication Engineering, ISSN (Online) 2394-6849.
6. S.G. Priyadharshini, C. Subramani and J. Preetha Roselyn "An IOT based smart metering development for energy management system" in International Journal of Electrical and Computer Engineering (IJECE)