IT-478 Internet of Things

Project Report

Project Title: Hot Water Management in Hostel

Guide : Prof. Sanjay Srivastava

Group 5:

Kirtan Patel	201601081
Dipesh Dhameliya	201601118
Yash Jobanputra	201601232

ACKNOWLEDGEMENT

The satisfaction on the successful completion of any task is incomplete without mentioning the people who made it possible whose constant guidance and encouragement crowned out effort with success.

We would like to express our heartfelt gratitude to our Guide, Prof. Sanjay Srivastava for his technical guidance and valuable suggestions. We are grateful to the DAIICT, for giving us the opportunity to execute this project, which is an integral part of the curriculum in B.Tech programme Internet of Things (IoT) course at the DAIICT, Gandhinagar,Gujarat.

Many thanks to Mr. Abhishek Jani and Ms. Nikita Joshi who helped us in our project work for their generous contribution towards enriching the quality.

ABSTRACT

We identified that Hot Water Facility Management is a important problem in the Hostels with shared bathrooms. Timings for Hot Water Availability are fixed which leads to wastage of energy when not in use and inconvenience for residents. All the Geysers have limited carrying capacity at a time, which means that Availability of Hot Water is not continuous. Keeping in mind these and many other problems we propose an IOT Based Solution which is simple, scalable, cost efficient and energy efficient. This solution increases convenience for the users as well as makes system more eco-friendly and makes it easy to monitor.

1 Problem Description

Most of the Hostels have sharing washrooms between students. Hostels provide Hot Water Facilities especially during Winter Season. For Hot Water Management, they simply use a Timer Circuit which Turns On and Off as per the Hard Coded Time everyday. But this system has a lot of problems and inconvenience for users. These problems are listed below :

- Hot Water Timings are fixed, but every floor has different set of students involved in different set of activities which makes this fixed Timings concept tiresome.
- On Weekend or Holidays, Student's schedule is very different compared to other days. But Hot Water Timings remain same.
- Because of limiting carrying capacity of geysers at a time, it's often possible that right before a student fills his bucket someone else has collected hot water.
- It's not easy for college authorities to check on working of geyser everyday. Hence, once a geyser stops functioning, it takes days or months to get it repaired.
- There are multiple geyser in hostels, but it's not possible to find out which one is nearest and has hot water.
- Also in certain colleges, students have limited Quota for Hot Water, but it's difficult to monitor the consumption.
- If a geyser is on even when no one [Especially in Vacations] is going to take a bath, there is wastage of energy and hence it's not eco-friendly.

Hence, a Modern Solution is required to address all of these problems.

2 Design Methodology

2.1 NodeMCU (ESP8266)

Features :

Open-source, Interactive, Programmable, Low cost, Simple, Smart, WI-FI enabled

Arduino-like hardware IO:

Advanced API for hardware IO, which can dramatically reduce the redundant work for configuring and manipulating hardware. Code like arduino, but interactively in Lua script.

Node.js style working:

Event-driven API for network applications, which facilitates developers writing code running on a MCU in Node Js style. Greatly speed up your IOT application development process.

2.2 Firebase realtime database

The Firebase Realtime Database lets you build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, real time events continue to fire, giving the end user a responsive experience. When the device regains connection, the Realtime Database synchronizes the local data changes with the remote updates that occurred while the client was offline, merging any conflicts automatically.The Realtime Database provides a flexible, expression-based rules language, called Firebase Realtime Database Security Rules, to define how your data should be structured and when data can be read from or written to. When integrated with Firebase Authentication, developers can define who has access to what data, and how they can access it. The Realtime Database is a NoSQL database and as such has different optimizations and functionality compared to a relational database. The Realtime Database API is designed to only allow operations that can be executed quickly. This enables you to build a great real time experience that can serve millions of users without compromising on responsiveness.

Also firebase allow you write your own triggers, functions or another event driven function to perform some calculation and save it to real time database.

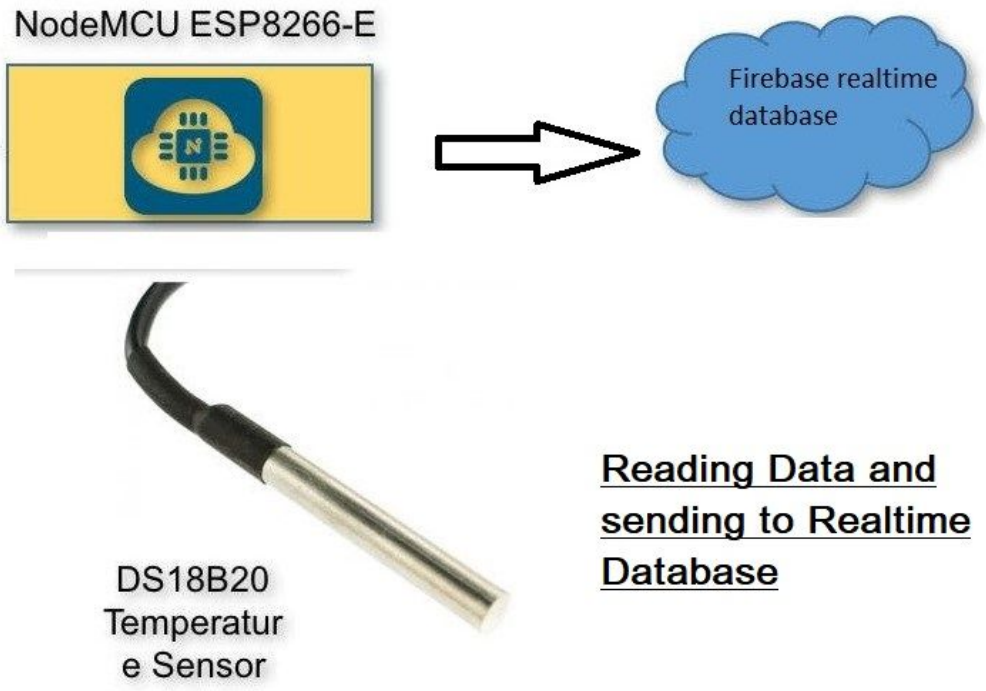
2.3 Node.js functions:

First we configure our database with a nodejs script using API key and some other details. Then we use auth service by firebase to authenticate that this user is our admin or not using request/response. Then we have some functions which are using request/response model to send data to our front end, which we can show to our admin. Then we have a function which uses node-schedule library and run exactly at our prefered time (for practical 10pm), but for showing project we'll use time of our project presentation. This function goes floor by floor and read preferences of all users and decide a time slot so that maximum user will satisfy and update this time in firebase so that NodeMCU can use this time.

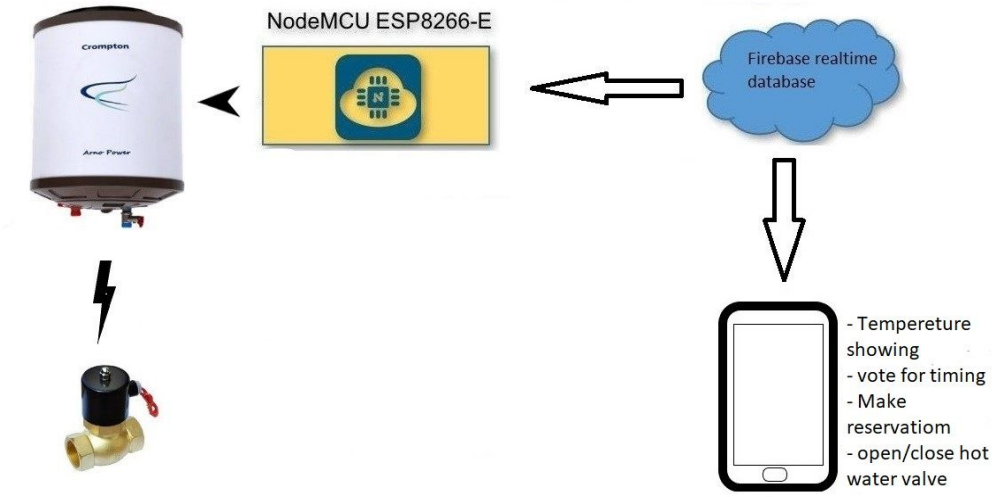
2.4 DS18B20 (Waterproof temperature sensor)

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply. Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area.

2.5 Circuit Diagram for sending data and controlling hot water valve:



1. **Reading Temperature data and send to Realtime database**



2. Mobile app and Node mcu communicate to database for reservation, open/close valve, voting

3 Solution

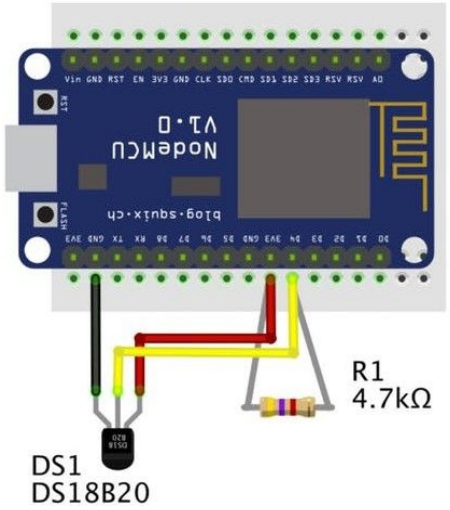
Implementation of above mentioned solution can be divided into 4 different modules as follows:

1. Setting Up Node Mcu and Sending Data to Realtime database[Firebase]
2. Creating a Mobile Application for users
3. Creating necessary Nodejs and firebase functions
4. Web Application for Admin

3.1 NodeMCU communicating to realtime database

There are total three functionalities that NodeMCU is performing,

1. **Reading Data and sending to Realtime Database**
   For this we use DS18B20 which is waterproof temperature sensor to read hot water temperature and send it to Realtime database. Basic introduction of DS18B20 is given above. Configuration part is given below.
   All nodeMCU code be found here. For code to run there are some libraries that need to be installed on computer before running code.



2. **Power on/off geyser**
   Our aim is to start geyser or say setting geyser timing according to students interest. For this we provide functionalities in mobile app where every student can vote for tomorrow's preference of geyser timing, according to votes functions on firebase side is called every night and set timing of geyser which then read by NodeMCU and next day according to that Geyser will be automatically started.

3. **Opening/Closing valve**

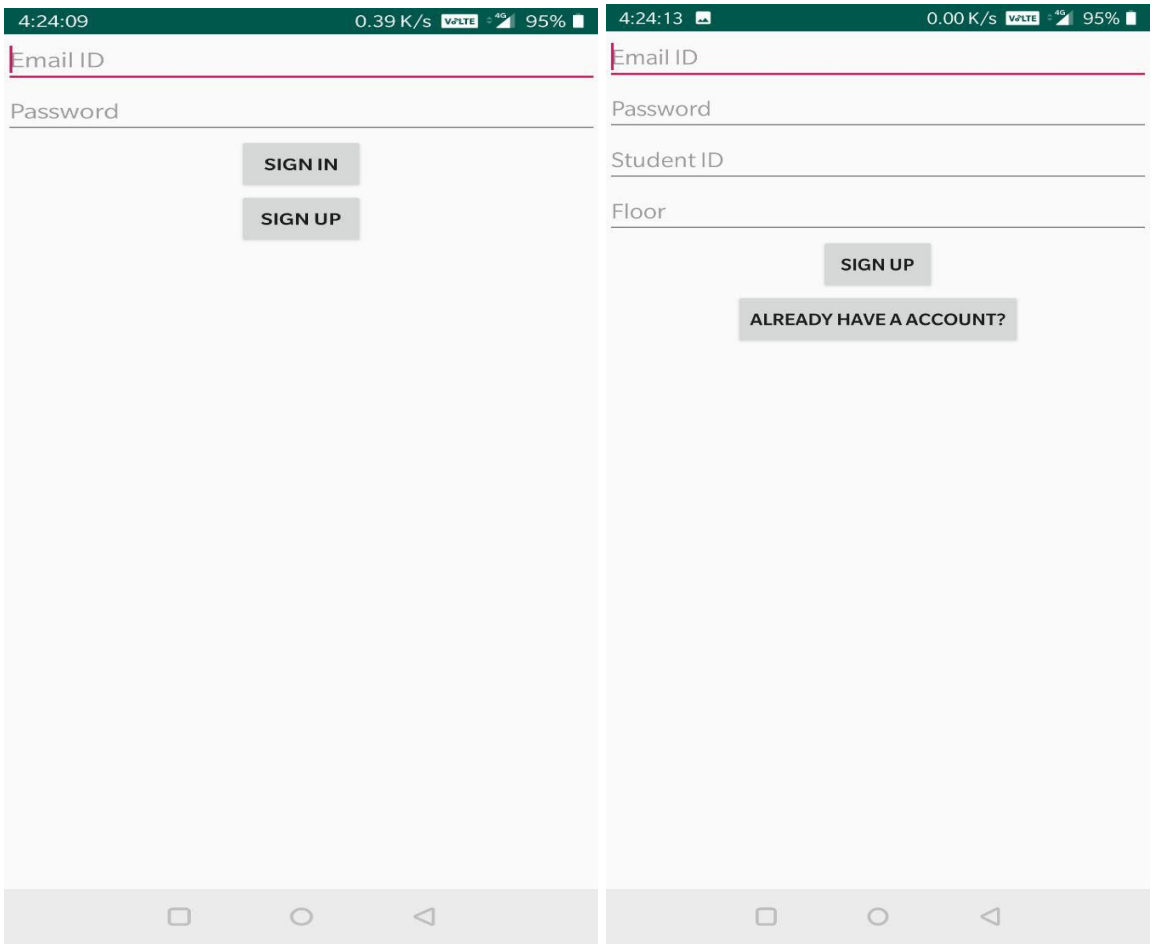
Once geyser timing was set and notified to all students, student then can make reservation of five minutes and he/she can withdrawn water using on/off button in application itself during this time all other student can not make request for reservation.
To electronically open/close valve we are making use of 12V motor but as future advancement we can use Solenoid valve which is just like any other water tap but can open using current.

3.2 Mobile Application

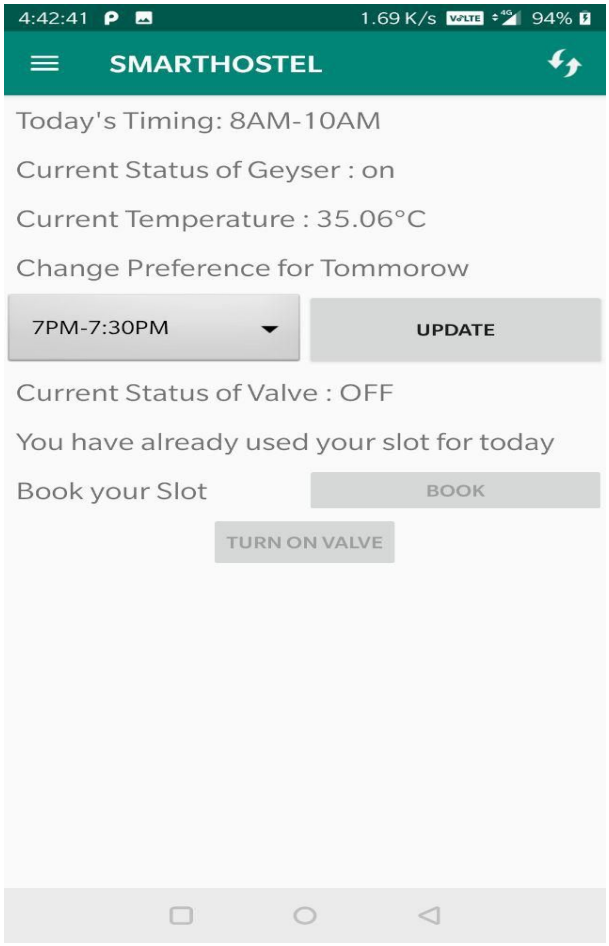
Mobile Application for the Users have the following functionalities:

1. **LOGIN/SIGN UP:**

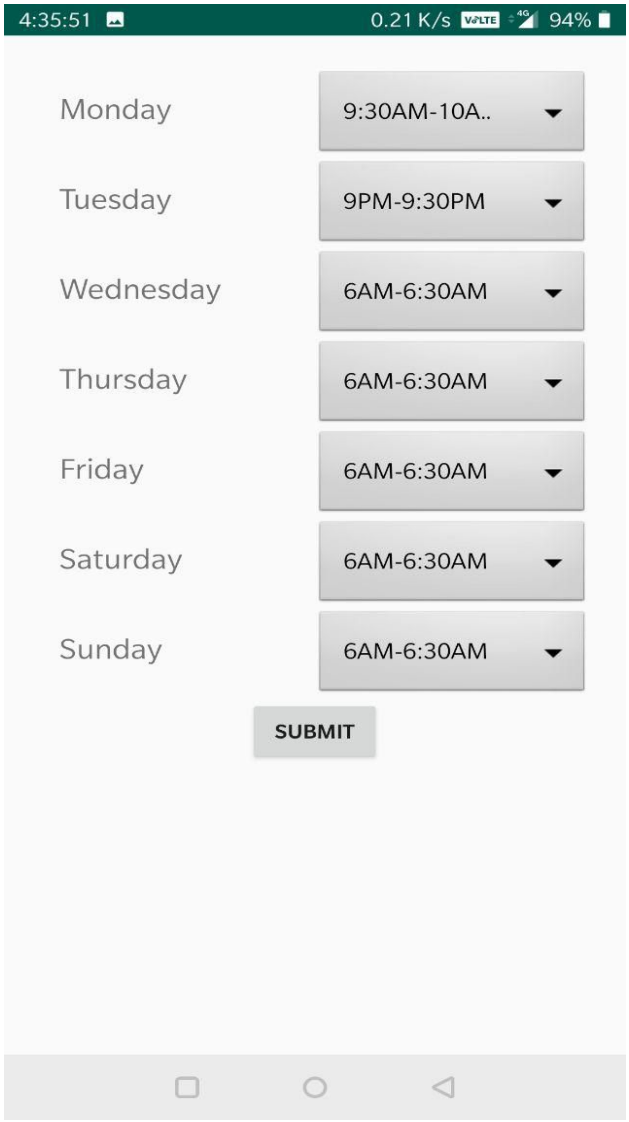
   Every user can sign up and sign in with their student ID, Floor and Email ID. We need Login Mechanism in order to keep track of the users interactions with the system like booking a slot and turning on the valve after booking a slot. We have used firebase Authentication API by Firebase Console to use Email as a Login and password for authentication. Firebase Stores user password as a hash in order to maintain user privacy. Following is the screenshot of LOGIN PAGE, SIGN UP PAGE.



2. **Main Functionalities:**
   User can see next day's geyser timings as soon as it is decided. And before that timings for the same day. Hence, as soon as the timing for next day is decided using Node.js script, this timing will be change. User can monitor live temperature of their own floor in order to see if the water is hot enough for them. For once a day, user can block other users from taking Hot Water for 5 minutes so that if he/she observes that water is hot, no one can else can take it out before him/her. As soon as his/her has booked a slot, turn on valve button is enabled which allows them to Turn On the Valve when they reach washroom.



3. **Selecting default preference:**
   If user is going to follow a specific routine for the next few weeks, he/she can set default preference from the activity as shown below. In case user doesn't select the timings for the next day manually, this preferences are used in taking decision for the next day.



3.3 Web Application

There is a sign in page which will only allow admin to login. We are showing real time temperature of all floors and there are buttons for every floor geyser to power on/off. Also there is complaints box where all complaints will be shown logged by students using mobile application.

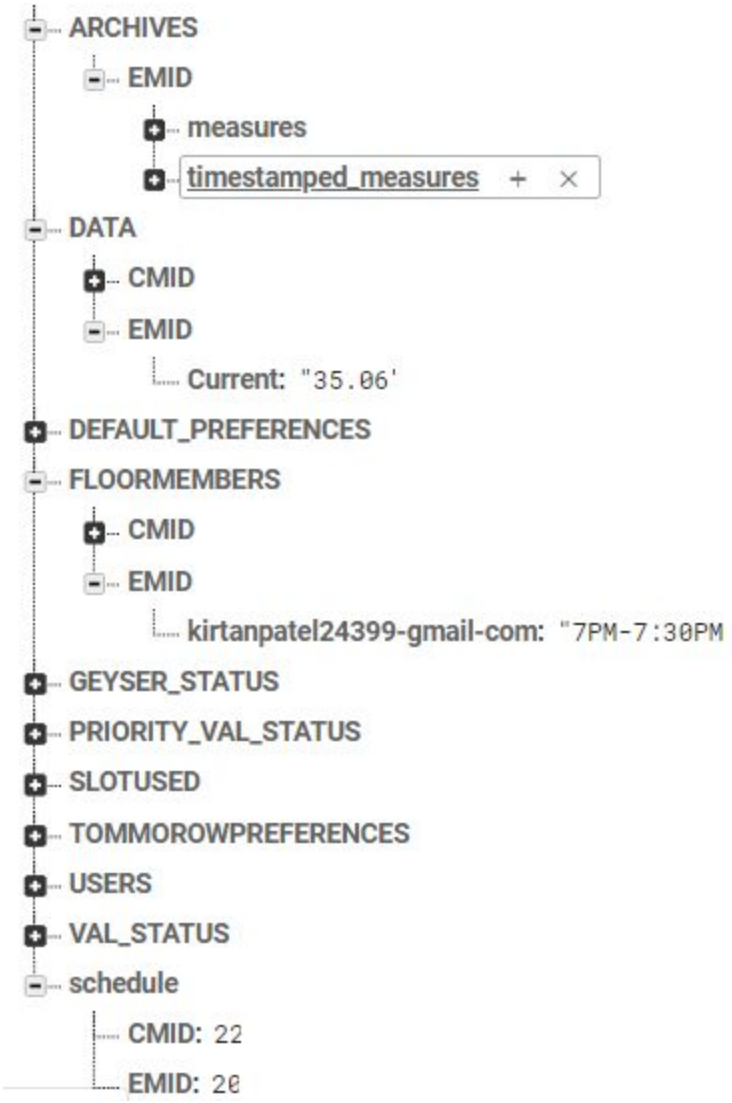
3.4 Firebase Realtime Database

So this part is heart of solution, Firebase provides very rich functionalities, services which can be used by Client side and server side to make things easy (e.g, scheduling, Triggering events on time, etc).

First of, Data read by NodeMCU send to database, where every push of entry trigger one function that attach timestamp as key to data and store it. This is mainly used due to timing mismatch clock of NodeMCU and Server (because NodeMCU has its own local clock).

Second all users are organized by floorwise so that they can monitor live temperature on mobile application.Also users geyser timing votes also are calculated based on their floor wise.

Database schema is shown below:



4 Future Scope

We have created a very basic prototype due to our limited resources and knowledge. There is a lot of scope in developing this model further and use it in real life:

1. There might be a few concurrency errors while booking the slot which we haven't faced because of the limited number of users during stimulations.

2. We have faced a lot of problems because of the time-based-triggers. Instead of creating them using node-js , we can use trigger service providers who can do this and Integrate them.
3. If Admin wants to put a cap on the amount of water each user can use everyday, then because of varying pressure of water, it is difficult to analyze the amount of time to keep the valve on. Hence, some sort of solution like combing pressure sensor or using flow monitor sensors can be integrated in this solution.
4. Time Selection algorithm can be made smarter by using machine learning or other algorithms.
5. Functionality to change the amount of time geyser remains on everyday according to number of floor members can be integrated if we can ask each user to vote everyday.

5 Conclusion

We have addressed problems related to the convenience of the user and efficiency of the energy. Our solution schedules the geyser at the time when usage will be maximized and energy is saved. Our solution is more convenient for user as they can be assured to get hot water whenever they reach washroom if they have pre booked the slot. Also this system makes it easier for admin to figure out faults by using website and they can be repaired by electrician as soon as possible. This reduces response time which is more convenient for the users. Hence, our IOT based solution is modern solution to a really specific problem. This kind of solution can also be used to solve other similar problems where there is sharing of resources between uncoordinated users. It also proves potential of IOT in addressing our day to day problems and increasing efficiency of our day to day operations.

6 Reference

1. https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf
2. https://www.instructables.com/id/Get-Started-With-NodeMCU/
3. https://medium.com/@o.lourme/our-iot-journey-through-esp8266-firebase-angular-and-plotly-js-part-1-a07db495ac5f