Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8347728

SMART CLOCK BASED ON REAL-TIME TRAFFIC DATA

T. SAI TARUN
Vellore Institute of Technology, Vellore, India. Email: saitarun785@gmail.com

BHARGAV SREEKAR
Vellore Institute of Technology, Vellore, India. Email: bhargavsreekar98765@gmail.com

GERARDINE IMMACULATE MARY
Vellore Institute of Technology, Vellore, India. Email: gerardine@vit.ac.in

Abstract

The internet of things (IoT) and Machine learning (ML) are two rapid growing technologies which connects devices to the internet and enables to be controlled and analyze the data from any internet connected device. Using these, this paper demonstrates an idea to upgrade an ordinary clock to a smarter one which can automatically adjust the alarm ring time based on live road traffic present on user route while user is in sleep state or busy, which makes sure the user reaches on time he marked to his destination and also clock has the ability to controls home appliances using smart mode. To implement this, we have used Distance Matrix API from Google maps to get live road traffic data, Telegram app for user interface and Raspberry Pi 3 microcontroller as prototype design.

Keywords: Distance MatrixAPI, Telegram Bot, Raspberry Pi

I. INTRODUCTION

The clock is one of the most used devices in human's daily life for mostly checking time and setting alarm. Being the device present in almost every household, there is not much advancement introduced on device happened till date. So, we got an idea to solve a humongous problem in daily life peoples i.e., begin late due to sudden traffic jams and add an IoT feature to the clock.

Almost in every major city the biggest problem is road traffic. This traffic cost huge loss to every sector associated to road transport as employees, goods are stuck. It also effects people's health due to stress on reaching on time to office, delivery of goods, to catch a train, flight and added to this, sometimes traffic becomes unpredictable and makes the person to stuck in this unpredicted traffic jam.

To overcome this problem, we have developed a 'smart clock' which can automatically adjust the alarm time based on the traffic present on user route. The alarm will adjust alarm ring time early if traffic is more on the route compared to regular traffic so that the user starts early and reach his destination on time.

There might be a question that the user can get the traffic data through Google maps/Apple Maps in smart phone, yes that is true, but the user cannot track the traffic all day while in his sleep or in a busy work and keep adjusting the alarm time manually each time when there is a change in traffic which is not a feasible solution. So, we designed our 'smart clock' to itself adjust the alarm time automatically once the route and time by which user should reach are set.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8347728

There are the cases that sometimes the traffic on the route peaks suddenly and it gets cleared fast by the time user reaches at that point, which dilutes the use of this alarm. For this case we have used an ML algorithm-based model which keeps track of traffic data on all the days following user travel patterns and also including special condition of weather condition, holidays while training the model.

Added to this the alarm also has a smart IoT features such as controlling home appliances when alarm rings or at a specified time. Use case can be like, when alarm rings in morning to wake the user, the clock can turn off A.C& Fan in the room, open window curtains, and turn on the light, turn on water heater or coffee machine as per user settings in the app.

II. LITERATURE REVIEW

1) For accessing the origin-destination travel pattern of the user, they have used mobile phone as a sensor through which they collect the data such as travel start location as origin and destination twice a day which makes a trip. With the help of mobile phone Call Detail Records (CDR), which contain major detail of users such as location, timestamp which has enables to calculate the trips based on origin, destination and time of day of several anonymous customers over a region and an agglomerative clustering algorithms and other filtering algorithms are applied, which provides vast information about movement traces in time of the day as well as the locations.
2) In this paper they have used an IoT platform to display and monitor the objects and display the temperature other details of that objects located at various places. The IoT platform used doesn't have the feature to display the location where the object is placed in a particular area, so they have integrated Google maps API for displaying the location of object using the coordinates received from Neo-6m GPS module and temperature details from DHT11 sensor. The locations of a particular object in coordinates which are obtained by GPS are sent to API request and the API receives the coordinates value and display the location in Map in IoT platform. For this model NodeMCU is selected as processor for carrying out the operation and REST is used for backend and frontend coding of IoT platform. The delay test parameter between getting the result and making request is also measured.
3) A detailed overview of Google Distance Matrix API and JSON (JavaScript Object Notation)format is given in this paper and finding the nearest health care center from user location has been implemented using Google's Distance Matrix API using SAS method. As API has gained much popularity and python or R for this purpose, in this paper they have implemented using their own SAS code. The shortest distance for hospital, within certain radius has been calculated and displayed in form of matrix. A basic URI (Uniform Resource Identifier) is implemented for placing request by user to calculate time travel between user location and hospital.
4) In this paper tracking of objects through GPS and GSM module is implemented. The object to be tracked such as human, vehicle or any object, the portable is attached, and the module sends coordinate value to tracking center through SMS using GSM module and these coordinate values are sent to Google map API for displaying the

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8347728

location on Google map. Using the Google map API and GPS module communication through mobile SMS is established. This low-cost, real-time implementation has use case in many applications.

5) The paper aims at accomplishing the task of an automatic attendance system for the employees using face recognition and Amazon Web Services like S3 and Quick Sight, and also adjusting the opening-closing process of the doors using the face recognition results. It also aims at triggering the electrical appliances like lights and fans of the office by detecting the movement of the people in the office. The paper also describes the webserver wherein all the appliances of the office as well as the AWS and login-based attendance of the employees can be accessed.

III. OBJECTIVES

The main objectives that need to be covered to make our model work are:

1. Implement Distance matrix API and generate a query for live traffic data from google maps without major delay.
2. Integrate Telegram app for the clock to receive user input such as source-destination, alarm time, controlling home appliances.
3. Machine learning algorithm based trained model to predict traffic on the route under different conditions.
4. Setup Raspberry Pi board to run the programs and work as smart clock included with a LCD display, alarm buzzer, sensors.

IV. CHALLENGES FACED

The major challenge for us was to develop an algorithm for microcontroller to automatically adjust alarm time based on live traffic on users input route.

The distance matrix API is a paid service from Google and has free trail up to certain limit of data query, which made our testing complex as we have to test our algorithm within that free limit and if need to do additional query, need to pay for it.

For the user interface we have used telegram app bot service. The integration of telegram bot to raspberry PI was challenging in which we have to setup communication between app in mobile and raspberry pi microcontroller to receive the user entered data.

Section V and *Section VI* describe the software tools and hardware components we have used in this work.

V. SOFTWARE DESCRIPTION

A) Distance matrix API:

The term API stands for Application Program Interface which is a protocol and a tool to access the web services provided by the owner virtually for different applications who needs integrate a part of those services in their application.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8347728

Similarly, the Distance matrix API is a service provided by Google maps to access its maps data through an HTTP interface. The API returns the data in JSON/XML format which contains 'origin', 'destination', 'date and time', 'estimated travel time on the route', 'distance' and the end user need to extract the content from that.

B) Telegram bot:

Here we have used Telegram's Bot services in smartphone as user interface for controlling alarm and home appliances.

VI. HARDWARE DESCRIPTION

C) Raspberry Pi3-B:

Raspberry Pi is a microcontroller developed by the Raspberry Pi Foundation. It is a low-cost, high performance single-board mini-computers mainly used for projects prototype development.

The Raspberry Pi runs on Linux OS and provides GPIO pins for interfacing external sensors and components.

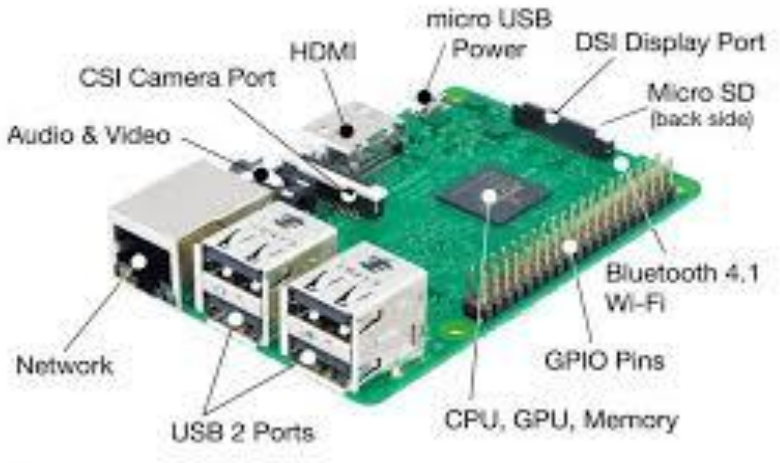

Fig.1: Raspberry Pi 3 Model B

D) 16x2 LCD display:

A16×2 LCD is an electronic display device which has 2 Rows and 16 Column. Due to its low cost, small size, and programmer friendly it is widely used.

In our work we have used this LCD to display current time and any user request data. We plan to integrate a bigger LED display in future work.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8347728

Fig. 2: 16x2 LCD Module Pin out

E) Buzzer:

Buzzer is an electronic device mainly used for generation beep sounds for indicating any action.

In our work we have used this buzzer module as alarm sound. We plan to integrate a mini speaker module which can play any alarm sound tone not being restricted to beep sound.

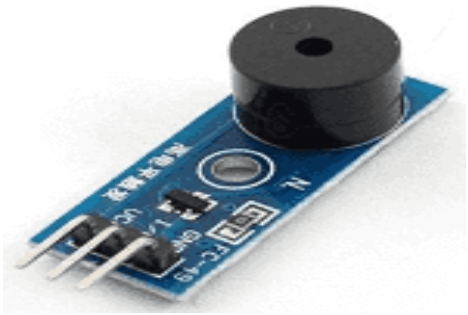

Fig. 3: Buzzer on Chip

VII. METHODOLOGY

This section describes how we have integrated our software and hardware components to achieve our objectives described.

A) Distance MatrixAPI:

The Distance MatrixAPI calculates travel time and distance between destinations-origin and returns information based these user parameters passed to it. (Fig 4)

To use the Distance MatrixAPI, a HTTP request is raised using the URL link with anAPI key included in URL. The API key is unique secure key per user and should not be made public by user.

URL link:

https://maps.GoogleAPI.com/maps/API/distancematrix/outputFormat?parameters

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8347728

The output Format can be

JSON: Indicates output in JavaScript Object Notation

XML: indicates output as XML.

The URL we used as an example is:

https://maps.GoogleAPI.com/maps/API/distancematrix/json?origins=vellore+bustand&destinations=Vit+vellore&key= <your API key>

HTTP request is sent using above URL and data returned by API is in JSON format shown in fig. 10, The JSON file describes the following parameters:

1. Origin address: Vellore bus stand (this is user input)
2. Destination address: VIT, Vellore campus (this is user input)
3. Distance = 5.7 km (data returned by API)
4. Duration = 11mins (data returned by API)

To extract each element from output, it needs to be parsed and we have used using JSON parsing technique suing python language.

The below flowchart describes the process of generating data from API.

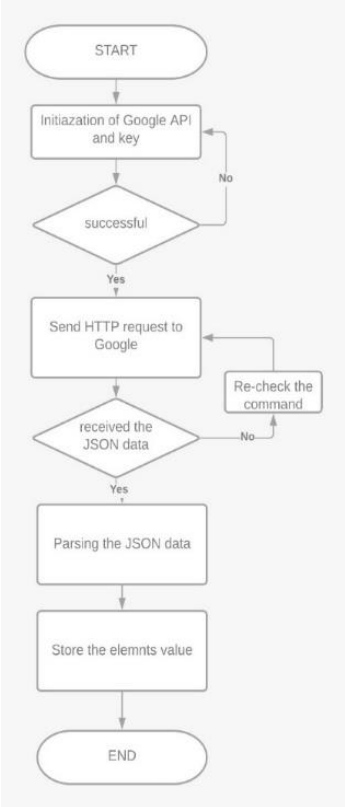

Fig. 4: API HTTP request flow graph

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8347728

B) Telegram BOT:

Fig 5 describes the flow of setting up telegram bot. For the first time, user should register for telegram bot service and then user gets a unique token key.

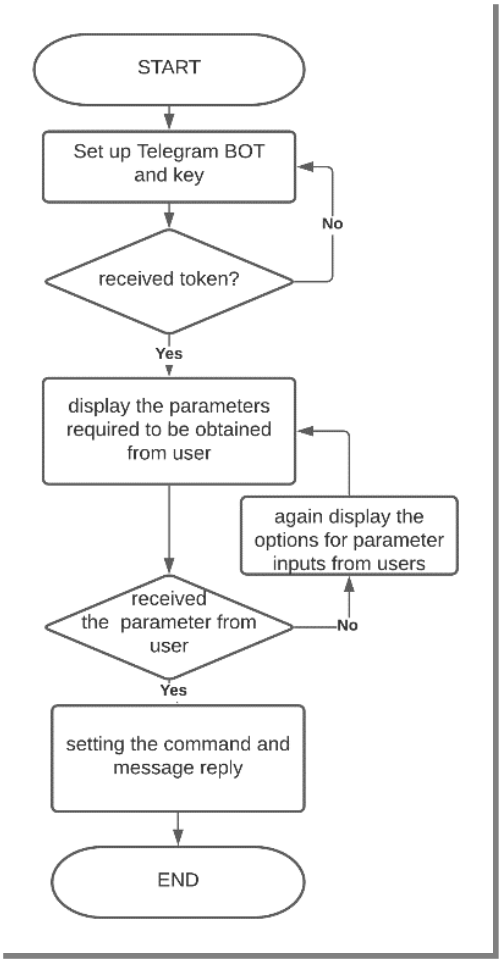

Fig. 5: Telegram Bot flow

The telegram bot is a generic bot capable of just transmitting data from mobile to raspberry Pi and raspberry pi receives the data in a plain text. To make it usable for our project we have made a python program to make raspberry Pi use that data to make the calculation.

First after turning ON the clock, the user needs to start bot service in mobile and the bot will make the connection between it and bot sends the users entered data stored in app to alarm clock. The alarm clock reads those inputs, processes it and sends request to API and gets the data. The user input is stored in clock so when there is no new data entered by user, the clock uses the same previous data.

User inputs are source*, destination, alarm time, time by with user should reach* as shown in Fig 9.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8347728

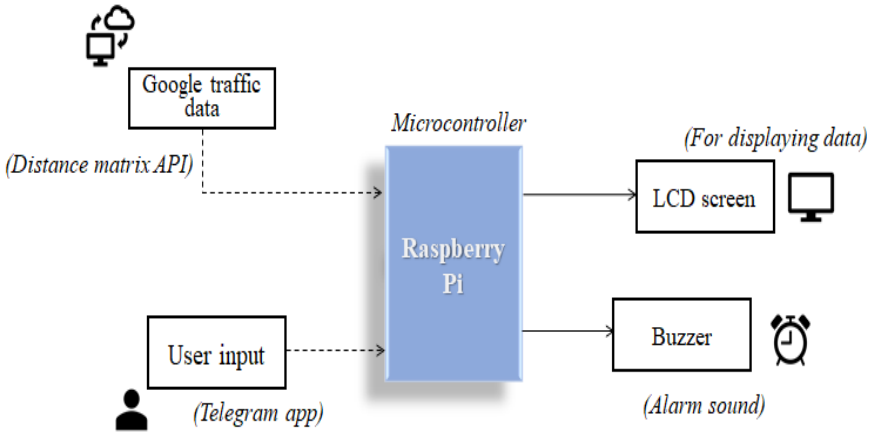

Fig. 6: Smart alarm block diagram (PART A)

VIII. ALGORITHM

This section describes how the algorithm was developed for clock to adjust time automatically based on live road traffic data.

The input given by user is transmitted to clock and based on those values the alarm time is adjusted by using calculations described below.

Once everything is setup, the clock starts tracking the road traffic 2 hrs before alarm time and for every 15mins.interval a request is sent to API till the alarm rings. After getting the data on every request, the clock does the calculation.

Below describes the different variables are used in algorithm which are,

- *time_should_reach -> the time user should reach his destination (input from user)*
- *set_alarm_time -> user set alarm time.*
- *current_time = (alarm_set_time – 2hrs) # at this time the clock starts checking traffic*
- *time_left = (alarm_set_time - current_time) # time left for alarm to ring*
- *travel_time = from Distance Matrix API # each time this gets updated*
- *New_alarm_time = updated alarm time*

```
At current_time:
    If (( estimated_travel_time + user_leaving_time ) > time_should_reach )
    {
      if time_left < 1hr:
            adjust_alarm_time()   # this sets new alarm time
      Else:
            Check for next 15mins also  # if for next consicative check
                                          also its more then change alarm time.
  )
  else
   Repeat for next 15 minutes till time_left = 0 .
```

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8347728

Let us consider an example to understand the working of above algorithm:

Consider the user sets alarm to ring at 8:00 am and gives time to reach his destination is at10:00 am and mentions the time he leaves the house as 9.15 am then,

The clock makes first request at 2hrs before alarm time which is 6 am in this scenario and time left for alarm ring is 2hrs which is (8 am – 6 am).

For first request at 6 am, the estimated travel time received from API is 30 mins and then below 'if condition' is checked.

Estimated travel time + user_leaving_time = 30mins + 9.15am = 9.45 am.

Here at 6 am, the calculated time to reach his destination is 9.45 am which is less than user requirement of 10 am, so the alarm doesn't change the ring time and wait for next 15mins. Before time left for alarm is zero which is 8 am, in any case the calculation says the time to teach is more than 10 am which say there is more traffic on the route, then the clock will change the ring time accordingly to make user reach by 10 am.

The below table shows the estimated travel time calculated by clock at each time interval of 15 mins.

Alarm set time	**time_should_reach**
8:00 AM	10:00 AM

Current Time	6:00	6:15	6:30	6:45	7:00	7:15	7:30
Estimated travel time	30 mins	31 mins	38 mins	40 mins	35 Mins	43 mins	40 mins

7:45	8:00
65 mins	

Here at 7:45 am the estimated travel time is 65 mins and time left is 15mins (8 am – 7.45 am) and the user reaches his destination at 10:20am if he leaves as mentioned9:15 am which is 20 mins late. So, the clock adjusts the rings time to 7:45 am which is 15mins early. Actually, in this case it should be 20mins early i.e., at 7.40am but clock makes a request at 7.30am and 7.45 am which 15mins time interval making 7.40am missed.

Controlling home appliances:

The controlling of home appliances is a smart IoT feature added to the clock apart from adjusting alarm time.

We have designed our smart clock to control appliances such as switching off fans, switching ON rolling motor to open window curtains, water heater at the time of alarm rings. The appliances to switch and at what time, can be customized through the same telegram App.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8347728

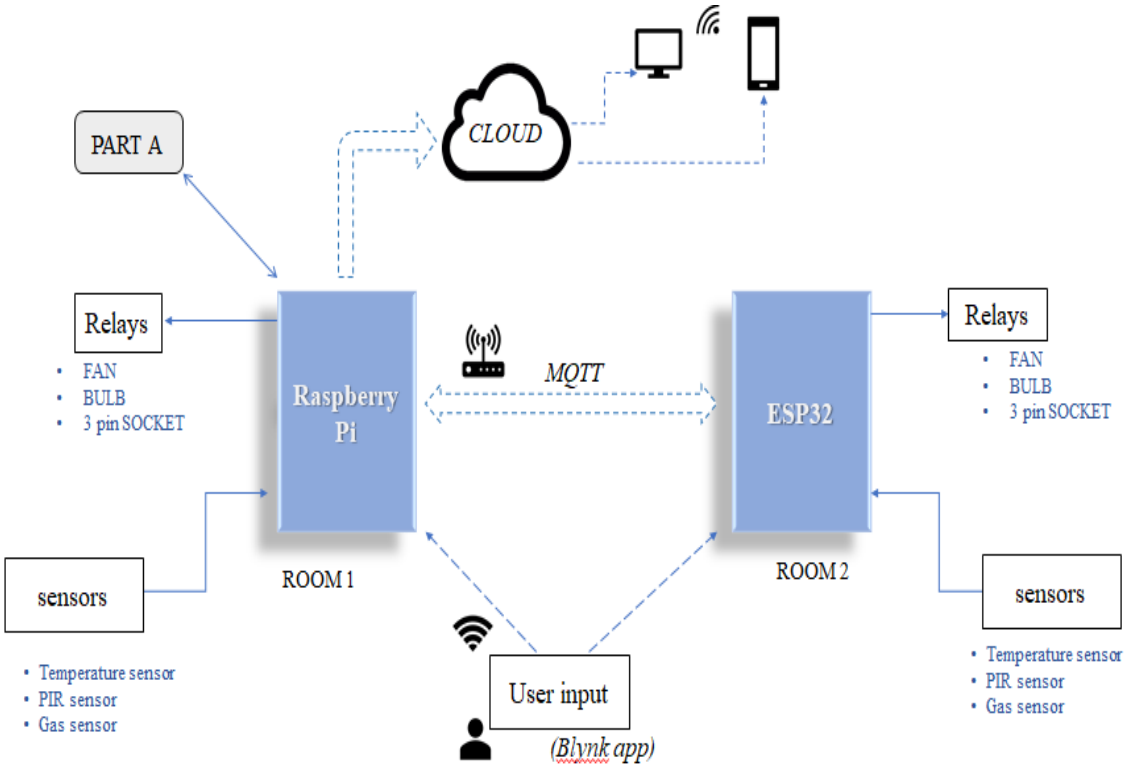

Fig. 7: Block diagram of automating home appliances

Fig 7 describes how the smart alarm (Raspberry Pi) is integrated with an ESP32 microcontroller which is a Wi-Fi module to control appliances in other rooms. The raspberry Pi is the main controller which controls appliances of room 1 and in room 2 we have used ESP32 to control appliances in that room 2. The communication between these two controllers happens through MQTT protocol. The live status of each room appliances will be updated in a cloud dashboard which can be accessed from anywhere using internet.

Smart mode:

In smart mode, the alarm clock switches off selected appliances when not in use by anyone for certain duration. In our model we shave setup alarm clock to switch off fans and lights when there is no one in the room for 15mins. The human detection is done using Passive Infrared (PIR) sensor and Ultrasonic proximity sensors.

In Fig 8, the smart alarm is connected to ESP32 as a second controller placed in other room. Both the microcontrollers measure the surrounding data and share the measured data to Raspberry Pi using MQTT protocol and Raspberry Pi receives the data and transfer it to cloud so the user can check it from anywhere.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8347728

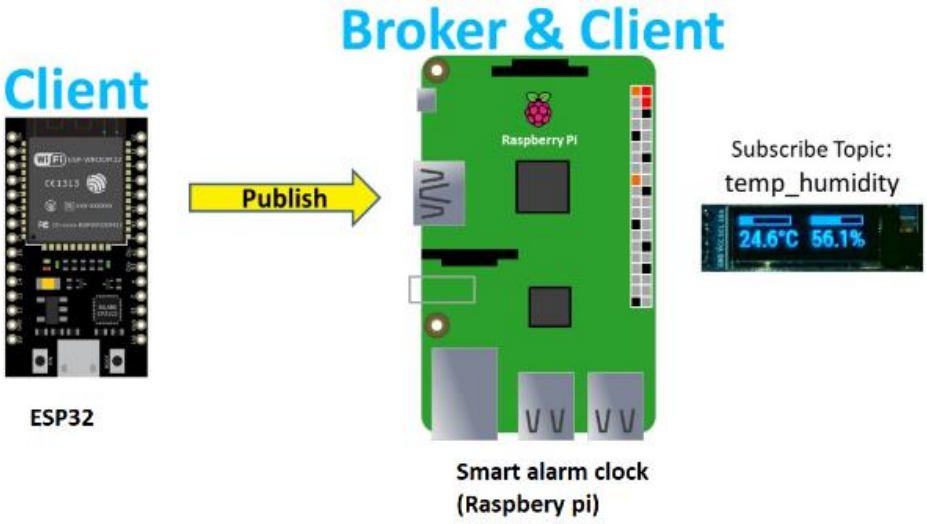

Fig. 8: MQTT setup

- Both the microcontrollers placed in two rooms have PIR sensors and this PIR sensor detects the presence of humans in the room. Four PIR sensors are placed at each corner of the room to increase detecting efficiency.

In this project we have calculated the energy saved by using this smart mode of clock as most of the times a person leaves the home without turning off all the appliances and they remain ON for a long period of time which is wastage of electricity.

For this test we have setup 2 CFL bulbs and 1 fan running in the room for 7 hours and the below calculations showed the electricity charge due to these appliances.

Considering, 2 CFL bulbs and 1 fan was kept running for 7 hours and each CFL bulb has 11W rating and Fan has 40W rating so total watt usage for 7 hours is (11*7 + 40*7) = 298Whr

Cost of one unit = 6.37rs

Total units = 298 / 1000 = 0.298 units

Cost of 0.298 units for 1 day = 0.298 * 6.37

= 1.898 Rs.

If this is repeated for 5 times in a month then it will be 1.898 * 5 = 9.493rs, which is for only Two CFL bulbs and 1 Fan and there may be more appliances left ON unused which increase total cost of energy wastage.

When our smart clock is used in this case, and it was able to turn off the bulbs and fan when user was not present in the room for more than 15mins and saved the energy cost.

We have also considered the electricity running cost of our raspberry Pi which consumes max of 6watts when all the cores are in use, which makes energy consumed per day is 0.144 kwh/day when ran for full 24hrs.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8347728

- **Future traffic prediction using ML model:**

As described in above introduction section on usage of ML model. A linear regression model and functional regression model were used, which will be able to capture trends in the traffic along the route.

The input data to training data was current time, weekday, weather, and travel time between two locations for 2 weeks and the output is travelling time on that particular day and time. The obtained trained model result is compared with Google map data and the output matches at 75% accuracy and we expect more accuracy if larger training data is provided.

IX. RESULTS

During the initial testing of our prototype, it has shown the microcontroller Raspberry Pi 3 is capable enough to handle the smart alarm application including receiving data from GoogleAPI, communication with Telegram bot, displaying the data on LCD and controlling home appliances. As reported from Linux process monitoring service, the smart alarm uses 38-45% CPU and 5% memory during normal operation. So, with so much of memory and processor power available we can further newer tasks and features.

Fig.9 shows the screenshot of telegram bot app working in which the predefined messages such as "*please enter origin*" are written in the code and bot ask these messages when the user first input the text "***start"*** and the user reply to those messages and those reply are treated as input to Google API.

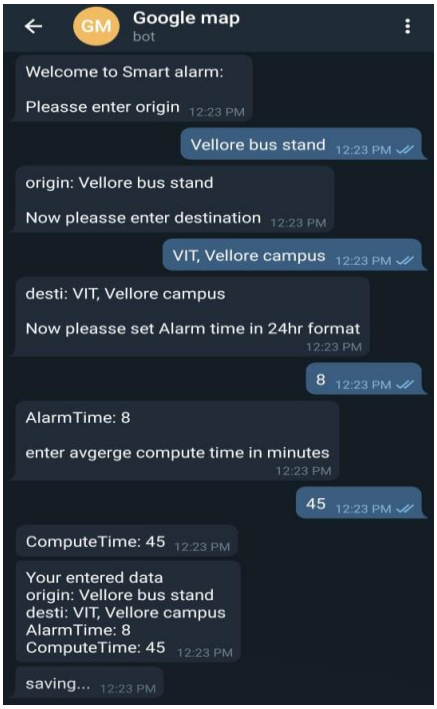

Fig. 9: Working of Telegram bot

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8347728


```
{
   "destination_addresses" : [
      "VIT, Vellore Campus, Tiruvalam Rd, Katpadi, Vellore, Tamil Nadu 632014, India"
   ],
   "origin_addresses" : [
      "Vellore Bus Stand, New Bus Stand Auto Rickshaw Road, Thottapalayam, Vellore, Tamil Nadu 632012, India"
   ],
   "rows" : [
      {
         "elements" : [
            {
               "distance" : {
                  "text" : "5.7 km",
                  "value" : 5738
               },
               "duration" : {
                  "text" : "11 mins",
                  "value" : 638
               },
               "status" : "OK"
            }
         ]
      }
   ],
   "status" : "OK"
}
```


Fig.10: Distance Matrix API response

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8347728

Working prototype of the smart clock:

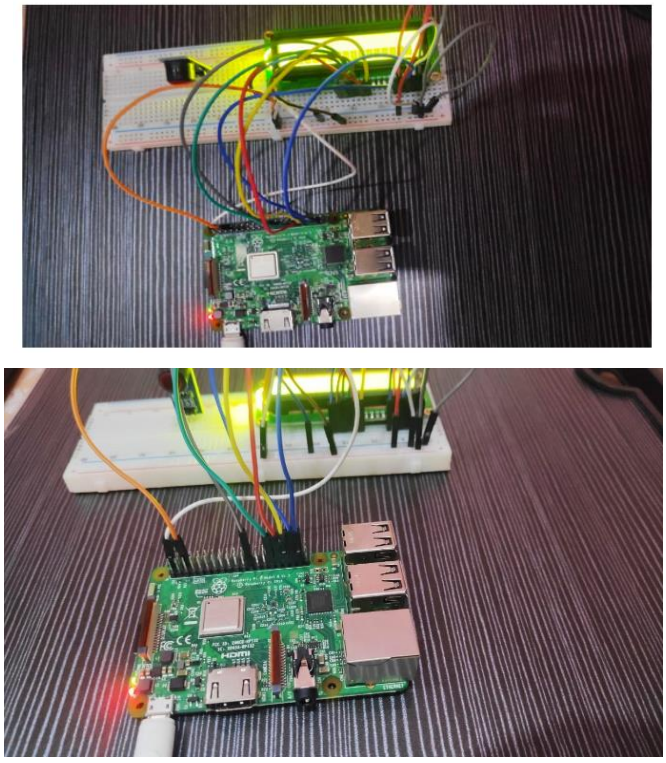

Fig. 11: Project final prototype

Fig. 11 shows the prototype of the Smart clock connecting Raspberry Pi interfaced with 16x2 LCD screen and a buzzer. The LCD displays the data such of current time, origin, destination, travel distance etc. and a buzzer is used for alarm ring sound.

Fig 12 shows that the ESP32 on right side connected to a relay (blue color) and Raspberry Pi on left is also connected to relay and through that really the appliance switch are controlled. [12]

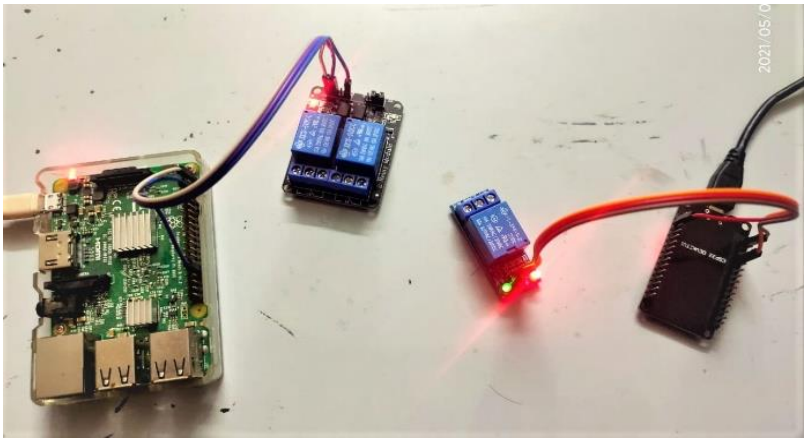

Fig. 12: Controlling home appliances

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8347728

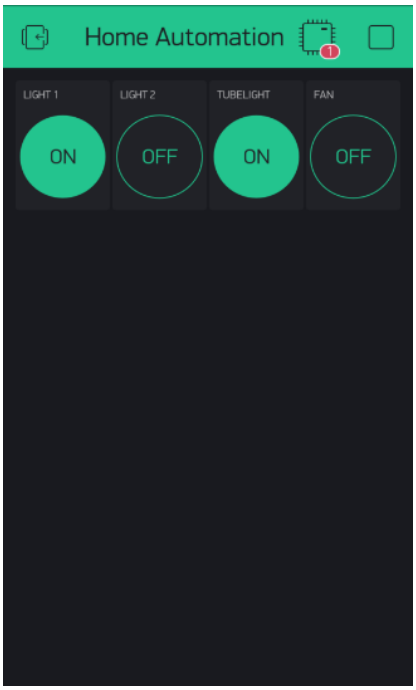

Fig13: Blynk app UI

Customizing the control of appliances at ring time i.e., to select which appliance to turn on/off by the clock at ring, can be done through Telegram bot and also, we have integrated a mobile app called Blynk (Fig 13) for controlling the appliances manually by user.

Fig 13 shows the UI of the APP in which we have setup four appliances, and their ON/OFF state can be triggered from the same.

X. CONCLUSION

As demonstrated in our working prototype which enables some enhanced features and processing capabilities to make a device as smart device using the technology of IoT and automation to reduce electricity wastage which was success due to Raspberry Pi low power consumption and high processing power at that form factor.

The main goal of proposed smart clock was to overcome the biggest problem of human life which is the travel delay caused due to traffic congestion. The smart alarm was able to dynamically set the user alarm according to the traffic present in the user traveling route. The automation of home appliance and controlling the appliances through mobile app is implemented and energy conservation using this model was also demonstrated which can be very useful if adopted at large scale which can save large amounts of electricity.

A simple machine learning model for predicting is implemented using forecasting methods, the model is able to predict the traffic with some good accuracy which can be improved by increasing the data set and more enhanced training model.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8347728

XI. FUTURE SCOPE

Some of the future enhancements that can be done are developing a separate mobile app for user interface which increases user access flexible with more added functions to control clock and appliance in one App and a voice command feature can also be added which enables the user to give some commands over voice to the device. Fig 14 describes the prototype design of our smart clock. A Google calendar can be integrated to the alarm which checks for the schedule and to-do list over the day and informs the user accordingly. The machine learning model can be improved by using more enhanced neural network algorithms and more data set.

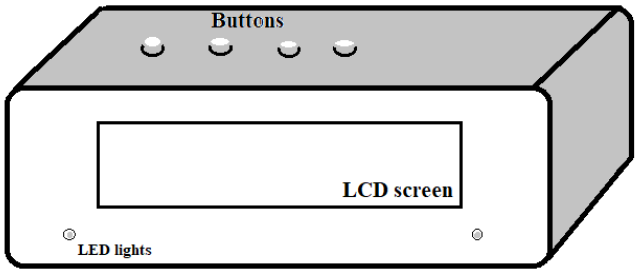

Fig 14: Future design on the clock

References

1) Lauren Alexander, Shan Jiang, Mikel Murga and Marta C. González.**"**Origin–destination trips by purpose and time of day inferred from mobile phone data", Science Direct, Volume 58, Part B, September 2015.

2) A. M. Luthfi, N. Karna and R. Mayasari, "Google Maps API Implementation on IOT Platform for Tracking an Object Using GPS," 2019 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob), BALI, Indonesia, 2019, pp. 126-131, doi: 10.1109/APWiMob48441.2019.8964139.

3) Tsutsui, Anna and Y. Ohno. "Application of Google Distance Matrix API to investigate healthcare access: a study on Hokkaido, Japan." (2020).

4) H. A. Abdallah Dafallah, "Design and implementation of an accurate real time GPS tracking system," The Third International Conference on e-Technologies and Networks for Development (ICeND2014), Beirut, 2014, pp. 183-188, doi: 10.1109/ICeND.2014.6991376.

5) D. Sharma, H. Sharma and D. Panchal, **"**Automatic Office Environment System for Employees Using IoT and Computer Vision," 2020 IEEE 17th India Council International Conference (INDICON), New Delhi, India, 2020, pp. 1-6, doi: 10.1109/INDICON49873.2020.9342455.

6) Aronsson, Linus, and Aron Bengtsson. "Machine learning applied to traffic forecasting." (2019).

7) Zhu Y. Introducing Google Chart Tools and Google Maps API in data visualization courses. IEEE Comput Graph Appl. 2012 Nov-Dec; 32(6):6-9. Doi: 10.1109/MCG.2012.114. PMID: 24807304.

8) http://en.wikipedia.org/wiki/Google_Maps[Online]

9) https://developers.Google.com/maps/documentation[Online]

10) Bots: An introduction for developers (Telegram.org)[Online]

11) https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started. [Online]

12) https://www.codeguru.com/iot/understanding-relays-in-iot-development/#:~:text=Think%20of%20a%20relay%20as,to%20turn%20off%20the%20light.