Indonesian Journal of Electrical Engineering and Computer Science
Vol. 27, No. 3, September 2022, pp. 1661~1669
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v27.i3.pp1661-1669

Internet of things based real-time electric vehicle and charging stations monitoring system

Emad A. Mohammed[1], Mahmood Hameed Qahtan[1], Ahmed J. Ali[2]
[1]Department of Computer Techniques Engineering, Technical Engineering College, Northern Technical University, Mosul, Iraq
[2]Department of Electrical Power Techniques Engineering, Technical Engineering College, Northern Technical University, Mosul, Iraq

Article Info

Article history:

Received Apr 11, 2022
Revised Jun 14, 2022
Accepted Jul 13, 2022

Keywords:

Blynk
Charging station
Electric vehicle
ESP32
Internet of things

ABSTRACT

Due to a shortage of fuel sources and the increment in environmental pollution, efficient techniques should be introduced. The best solution is to move to the use of electric vehicles. The article aims to develop a solution for electric vehicle (EV) charging station locations that utilize the internet of things (IoT) technology. The IoT is a paradigm that uses sensors and transmitting networks to provide current facilities with a real-time global communication perspective of the physical world. This paper proposes a real-time system to provide a real-time update to EV location and charging stations (CSs) location to reduce time lost by users searching CSs, and provides real-time charging station (CS) recommendations for EV users by displaying the nearest CS, provide estimation arrival time to the nearest CS, display distance between nearest CS and EV real-time updated. The work of the proposed system was tested, and the most significant error rate (17 meters) is represented by the difference in the distance obtained from the system and the distance obtained from Google Map. The total accuracy of the design for the tested case is (98.014%).

Corresponding Author:

Mahmood Hameed Qahtan
Department of Computer Techniques Engineering, Technical Engineering College
Northern Technical University
Mosul, Iraq
Email: mahmood.hameed@ntu.edu.iq

1. INTRODUCTION

Countries worldwide have enacted regulatory measures in recent years to address concerns in the transportation industry, such as the rising demand for non-renewable fuels and pollution in urban areas. As a result of worries about the supply of fossil fuels, the environment, and emission regulations, eco-friendly electric vehicles (EVs) have attracted ample space attention [1]-[3]. Previously, there was a small population of electric vehicles; now, electric vehicles will be the means of transportation in the future [4]-[6]. EVs provide significant economic and convenience advantages. Unlike internal combustion engines (ICE), electric cars do not have maintenance expenditures [7]. The battery's capacity determines the driving range of EVs, and the complexity of the engine also plays a factor. Recent research has focused on increasing the range of EVs [8]. The primary goal of internet of things (IoT) is to improve the quality of human existence [9]. The rapid advancement of internet technology has resulted in. The IoT will be at the core of computer engineering research [10].

The number of sensors in our daily lives, such as cellphones, automobiles, and buildings, is rapidly growing [11]. The IoT is a network of devices (things) that interact over the internet. IoT is a revolutionary concept in which all sensor data is saved in the cloud and quickly accessed. The cloud for more than just storing data; we also use it to analyze, gather, and display it. In the internet of things, data is transferred by sensors and saved and analyzed by various IoT platforms such as Blynk, Thinger.io, and Thingspeak [12], [13]. The ESP32 chip has become very popular, and there are already several hardware modifications and software development streams for it [14]. The global positioning system (GPS) will help us find the location and route, allowing us to follow the guide without consulting a map or memorizing the names of specific roads [15]. The Haversine method is used latitude and longitude as variables to calculate the distance between two places on the earth's surface [16]. This method can assist us in guiding the user's position to a specific destination more efficiently and straightforwardly [17].

2. THE PROPOSED METHOD

The proposed system consists of a mobile application shown in Figure 1, designed for EV users based on IoT to allow users to know the location of the charging stations (CSs) and their EV location on the google map in real-time also to determine speed, distance, nearest station, and estimated time to reach the nearest CS all these parameters are to be updated and displayed in a real-time manner. A microcontroller ESP32 with a built-in Wi-Fi module, GPS module, and mobile application (android app) connected to the IoT platform server (Blynk server) are part of the proposed system. The system works by connecting the GPS module to the ESP32, and all information received from the GPS module is collected and processed by ESP32 and transmitted to the Blynk server through the Wi-Fi module built-in ESP 32. The locations of the CSs are written and fixed within the program and displayed on the map with the current location of an EV. the information received from the GPS module, such as (position coordinate with longitude, altitude and speed displayed by the android application and used to calculate distance between current EV location and all CSs location to know the nearest CS also to calculate the estimated time to reach the nearest station. All users can access this information by an android mobile application using Blink application.

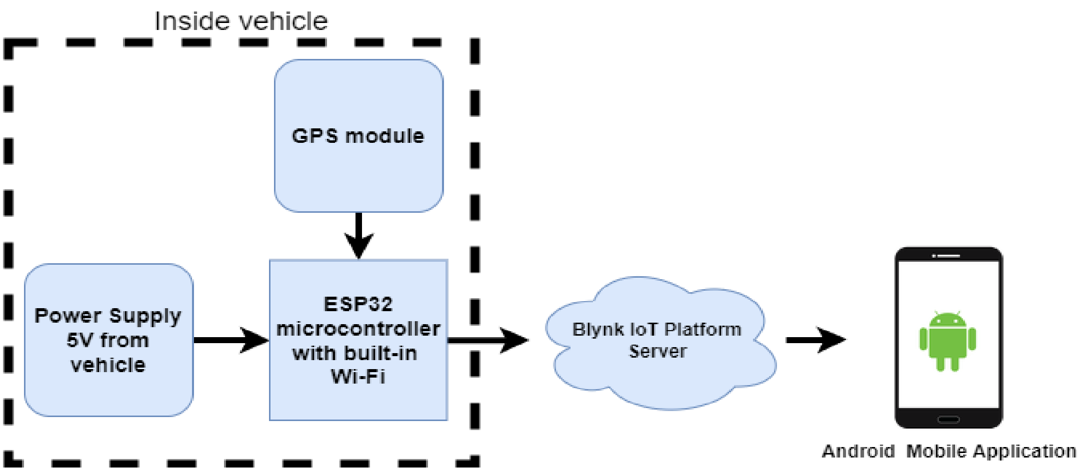

Figure 1. Block diagram of the proposed system

3. RESEARCH METHOD AND MATERIAL

The implemented system consists of an EV and CSs monitoring system based on IoT architecture, shown in Figure 2. It has been implemented by using a microcontroller ESP32 placed inside each EV. Connected GPS module to ESP32 and use universal serial port built in the vehicle using as power supply source with 5v dc needed to power on system. Use 3G service from users mobile as internet source to upload the information from ESP32 to Blynk server IoT platform. Using Wi-Fi module built-in ESP32 to connected to the internet. Information collected from GPS module and processed by the ESP32, as the current location of the EV (longitude and altitude) and speed used to calculated the distance between EV and all CSs available and registered in the system, find the nearest CS (it has the less distance from the current EV location) by using heaversin formula. The estimated time to reach the nearest CS depends on the speed and distance. All received and processed data are uploaded to the Blynk server and displayed using the mobile application. This information is displayed and updated in real-time to allow users to know the location of the electric vehicle charging station.

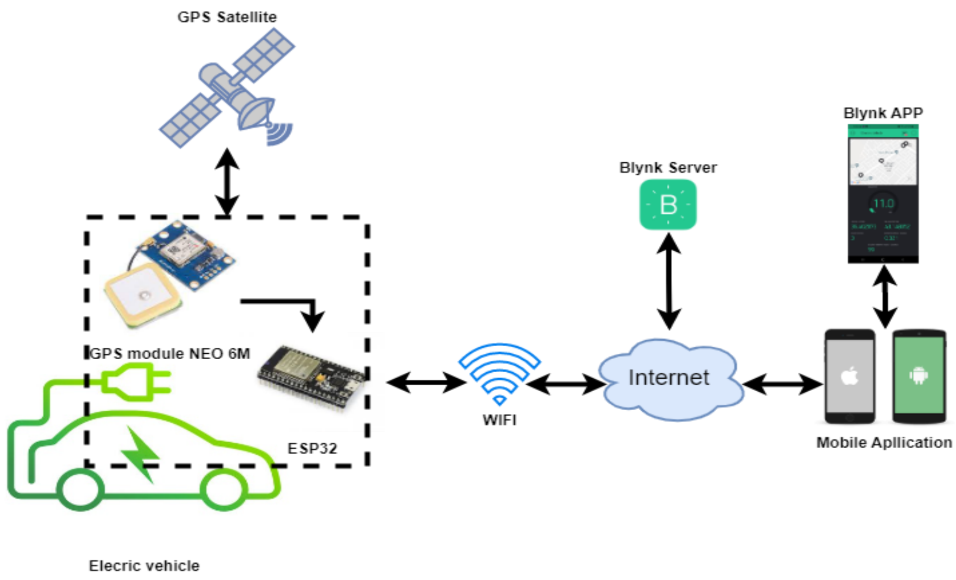

Figure 2. Electric vehicle and charging stations monitoring system architecture

3.1. ESP32 Microcontroller

The ESP32 is a strong system on chip (SoC) microcontroller with built-in Wi-Fi 802.11 b/g/n, dual-mode Bluetooth 4.2, and various peripherals, as seen in Figure 3(a). It is a better successor to the 8266 chip, particularly in including two cores clocked at different rates up to 240 MHz in various variants. Apart from these enhancements, the number of GPIO pins is increased from 17 to 36 and includes 4MB of flash memory.

ESP32 has two main components (Xtensa LX6 processor made with 40 nm technology). Controlling individual CPU cores is possible. On-chip SRAM for data and instructions is provided in 520 KB. Some SoC modules, such as the ESP32-WGrover, have 4 MB of external SPI flash and 8 MB of SPI pseudo static RAM (PSRAM). Depending on the type of board.

There are many options for using SPI, I2S, I2C, CAN, UART, ethernet MAC, and IR. Other built-in sensors are integrated into azure IoT and developer kit and the hall effect sensor, temperature sensor, and touch sensor. AES, SHA-2, RSA, and elliptic curve cryptography (ECC) are all supported by the SoC, as well as a random number generator (RNG) [18], [19].

3.2. GPS Module NEO 6m

Show basic system hardware as shown in Figure 3. Microcontroller ESP32 as illustrated in Figure 3(a) and the GPS U-Blox 6m module, as illustrated in Figure 3(b), GPS Module is a type of stand-alone GPS receiver with outstanding positioning performance. This module's streamlined design, power, and memory make it ideal for devices that use the battery as a resource while keeping costs and space to a minimum. It may reduce time-to-first-fix (TTFF) to less than 1 second because of its 50 channels of positioning engines [20]. For communication, this module has a patch antenna with a sensitivity of 161 dBm [21]. It is used in the proposed system to get the location and speed of EVs because the GPS module operates at 3.3 v; it is powered by ESP32 by using two pins in ESP32 (3.3 V, GND), as shown in Figure 4.

(a) (b)

Figure 3. Show basic system hardware: (a) microcontroller ESP32 and (b) GPS module NEO 6M

3.3. Application

Blynk is the most user-friendly IoT platform, with an app builder that runs on iOS and Android and a collection of libraries that allow you to create great IoT apps in minutes using any chosen hardware platform. It allows to easily create interfaces for controlling and monitoring hardware projects from iOS or Android devices by simply dragging and dropping widgets [22]. A digital dashboard is a graphical user interface formed. Widgets can be dragged and dropped from the widget box. Every widget needs some form of energy to function. A user will receive 2,000 energy (points) when establishing a Blynk account. When a widget is utilized, the energy balance decreases [23]. the android mobile application is used to display all information uploaded by ESP32 to the Blynk server. Blynk application contains a map widget that allows the user to see the current location of EV and all registered CSs on the same map also the current coordinate, speed, nearest CS, estimated time to reach nearest CS and distance between the EV and nearest CS provided by mobile application.

3.4. Distance calculation and arrival time estimation

Data received from the GPS module (speed, current location coordinate of EV) was used to find the distance between the current EV locations. The nearest CS also estimated the time to arrive at the nearest CS by doing some calculations. The Haversine formula, as shown below, is an excellent formula for calculating the straight-line distance between two points. Haversine uses latitude and longitude to conduct computations. Because Haversine can only be used on a globe map, it can't be used to measure distances in units other than latitude and longitude.

The Haversine algorithm is exact. The Haversine technique may compute the distance between two coordinate locations on a globe map. A technique for calculating the distance between two locations on the earth's surface. The Haversine algorithm is the one used. This technique uses latitude and longitude to determine distances. The Haversine algorithm seeks to determine the shortest straight line distance between two points [24], [25]. This algorithm was used in calculating the distance between an EV and all CSs to know the nearest one to provide the user with the nearest CS number and distance between the EV the and nearest CS by displaying this information on the android application dashboard:

$$a = sin2(\Delta\phi/2) + cos\phi1.cos\phi2.sin2(\Delta\lambda/2) \quad (1)$$

$$c = 2.atan2\ (\sqrt{a}.\sqrt{1-a}) \quad (2)$$

$$d = R.c \quad (3)$$

where:
ϕ=latitude; λ=longitude; Δϕ=latitude 2−latitude 1; Δλ=longitude 2−longitude 1; R=earth radius (6371 meters); d=distance between the locationstion.

To find the estimated time needed by electric vehicles to reach the nearest charging station depending on the distance between them calculated from the previous step by using the time formula:

$$T = D/S \quad (4)$$

where:
T: estimated arrival time to the nearest station; D: distance between EV and nearest CSs: average speed of EV. The time obtained from (4) in an hour to convert to mint multiply by 60 because each hour is 60 minutes, as shown in (5).

$$T = d/s * 60 \quad (5)$$

3.5. Circuit design

To connect to the GPS module, the ESP32 only utilized four pinouts, as shown in Figure 4; VCC 3.3V, GND, D16, and D17 are the pins. The GPS module's Tx and Rx were linked to D16 and D17. The signal from the GPS was received by D16 (RX), and it was relayed to the ESP32 via D17 (TX). This GPS module performed admirably with ESP32 to obtain an accurate signal and transmit its data to a cloud server. This circuit combination resulted in a steady signal with exact latitude and longitude.

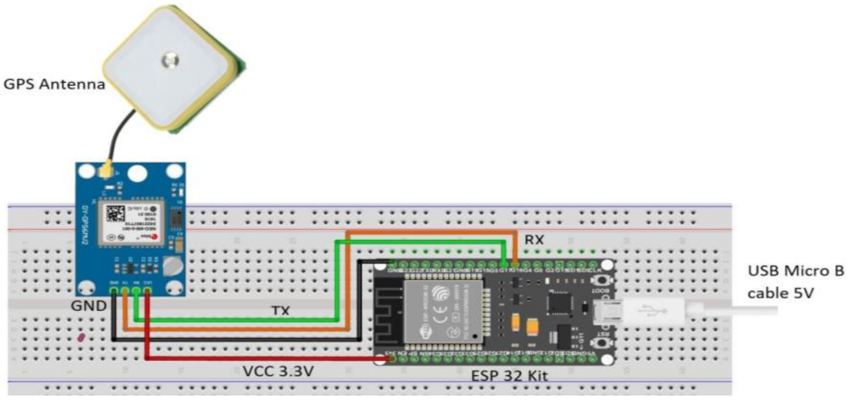

Figure 4. Proposed system circuit connection

3.6. Proposed system performance

The system is performed in steps illustrated in the flowchart shown in Figure 5. first, the location of the registered CSs must place within the main program, then power on the system and stay the GPS module to initialize to receive the current EV location coordinate and speed. If there is any problem with the connection or the GPS does not initialize, try again until a microcontroller gets information. ESP32 has a Built Wi-Fi module used to connect the ESP32 to the internet allowing the Microcontroller to send received and processed data to the Blynk server. To connect to the Blynk server connection must be established by using the Blynk authentication token. After connection successfully, the information sends to the Blynk server. The received data from the GPS module was used to determine the nearest CS and calculate the distance between the EV and nearest CS and the estimated arrival time to the nearest CS. The users display all the information using the Blynk app using some widgets on the application dashboard. The map widget is used to display all CSs, and current EV location to allow the user to see that on google map also allows the user to see calculated information on the application dashboard (EV longitude and altitude, speed, nearest CS number, estimated arrival time, and distance between current EV location and nearest CS). All this information displayed and updated in real-time.

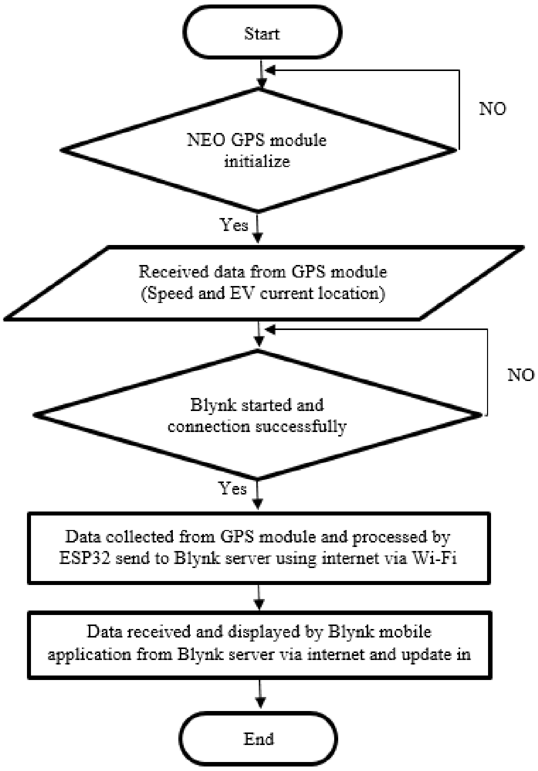

Figure 5. Proposed system flow chart

4. RESULT AND DISCUSSION

To test the proposed system three charging stations have been considered with the location coordinate:station1(latitude:36.396632,longitude:43.139484),station2(latitude:36.399103,longitude:43.143435),station3(latitude: 36.401716, longitude: 43.147552). The vehicle is supplied with the implemented hardware circuit, powered by using a USB port with 5v dc found in the vehicle. The system is equipped with the internet to use internet of things capability. All information collected and processed by ESP32 is transmitted via the internet using the ESP32 built-in Wi-Fi module and displayed by Blynk mobile application. After ensuring that the system properly worked, the moving uploaded the obtained results by the mobile application, as shown in Table 1. As shown in Table 1, the time column represents the time during which readings were taken during the system check. The lines colored in bold black represent the closest distance from the nearest station that was displayed during the car's movement.

Table 1. Data obtained from the proposed system test

Time (min)	latitude	longitude	Speed (Kmph)	Distance (meter)	Nearest CS number	Arrival time(min)	Google map Distance(meter)
0.02	36.396107	43.13863	8.07	95	1	0.706	95
0.04	36.396111	43.138699	6.42	91	1	0.85	92
0.06	36.396118	43.138714	6.44	89	1	0.829	91
0.08	36.396194	43.138809	9.77	82	1	0.598	78
0.10	36.396210	43.138832	10	74	1	0.444	75
0.12	36.396229	43.138855	10.8	74	1	0.444	72
0.14	36.396248	43.138855	11.5	71	1	0.393	71
0.17	36.396366	43.139065	17.4	47	1	0.162	48
0.20	36.396393	43.139107	18.1	42	1	0.139	43
0.22	36.396484	43.139248	19.7	37	1	0.118	27
0.24	36.396572	43.139393	19.7	15	1	0.046	11
0.26	36.396603	43.139442	19.6	4	1	0.012	5
0.28	**36.396694**	**43.139584**	**19.6**	**0**	**1**	**0**	**11**
0.30	36.396751	43.139675	17.4	21	1	0.072	19
0.32	36.396774	43.139717	15.7	26	1	0.099	26
1.04	36.398033	43.141644	30.8	2.17	2	0.409	200
1.06	36.398174	43.141865	29.3	174	2	0.355	170
1.08	36.398216	43.141933	27	166	2	0.368	170
1.1	36.398304	43.142078	24.2	159	2	0.393	150
1.14	36.398354	43.142162	13.2	141	2	0.637	140
1.16	36.398403	43.142197	15.4	137	2	0.532	140
1.20	36.398571	43.142502	23.6	109	2	0.283	100
1.25	36.398727	43.14275	26.2	88	2	0.206	74
1.28	36.398853	43.142948	28	51	2	0.109	51
1.30	36.398895	43.143017	28.3	43	2	0.091	44
1.32	36.399025	43.143227	27.7	20	2	0.074	20
1.34	**36.399147**	**43.143429**	**27.2**	**4**	**2**	**0.009**	3
1.38	36.399223	43.143559	24.5	10	2	0.023	16
1.42	36.399372	43.143845	14	43	2	0.176	47
2.10	36.400536	43.145672	24.9	213	3	0.512	210
2.13	36.400593	43.145763	14.7	207	3	0.635	200
2.16	36.400650	43.145847	9.13	193	3	1.268	190
2.18	36.400665	43.145874	11.3	190	3	1.006	190
2.22	36.400806	43.146099	19.3	169	3	0.568	170
2.24	36.400841	43.146149	20.7	164	3	0.509	160
2.27	36.400875	43.146198	21.2	153	3	0.432	150
2.3	36.401047	43.146469	22.3	122	3	0.327	120
2.34	36.401100	43.146557	13.7	116	3	0.354	110
2.37	36.401146	43.146633	6.88	103	3	0.897	100
2.4	36.401157	43.146652	7.96	101	3	0.761	100
2.44	36.401310	43.146877	20.8	75	3	0.215	75
2.5	36.401512	43.147202	21.3	50	3	0.134	39
2.52	36.401543	43.147255	20.5	32	3	0.093	33
2.57	36.401680	43.147488	9	12	3	0.049	7
2.58	36.401691	43.147511	8.98	4	3	0.028	5
3	36.401703	43.14753	8.68	2	3	0.014	2
3.03	**36.401745**	**43.147602**	**8.07**	**0**	**3**	**0**	6
3.05	36.401772	43.147644	5.25	10	3	0114	10
3.08	36.401779	43.147659	5.16	11	3	0.128	12
Total				4057			3978

After the system is powered on, a mobile application is used to display the current vehicle location coordinate (longitude, altitude), the CSs location and the average speed of the vehicle and the nearest CS between three CSs assumed, estimated time to arrive at the nearest CS as well as the distance between the vehicle and the nearest CS. Clicked on the location icons shown on the map, displaying the location's name, so can know what this location represents. When a vehicle moves in the right direction of the nearest CS, the distance decrease, but when moving far away from the nearest CS, the distance increases, as well, as the number of this station keeps appearing as the closest one, as shown in Table 1 until the next station becomes the closest, The nearest station number is changed in the mobile application dashboard; also the distance between the current vehicle location and the new CS location is displayed. The arrival time calculation depended on the speed and distance. All the information displayed by mobile applications, as shown in Figure 6, is updated in real-time. Compare the obtained result with the google map to check the system and find the different max value (17meter). Using another mobile navigation application to check vehicle speed, most times, the speed in the proposed system application and navigation program is the same but at some times the different (±1Kmph).

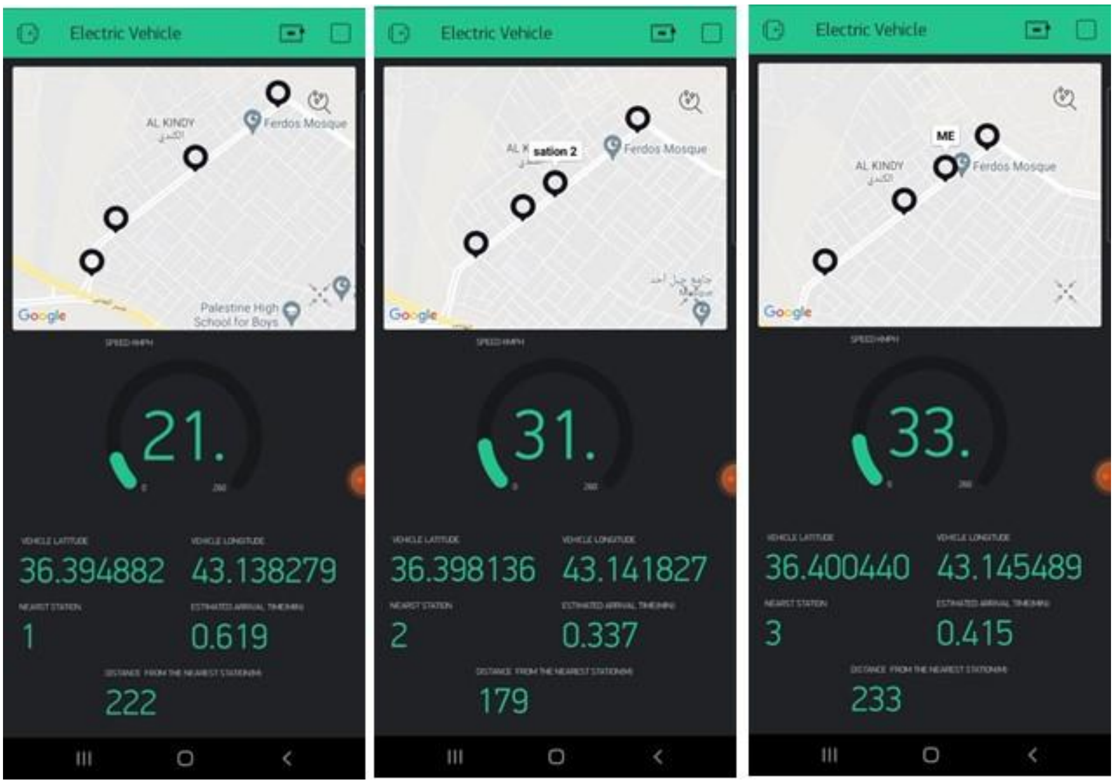

Figure 6. System information displayed by mobile application

4.1. Percentage accuracy of the proposed system

Using the total distance values obtained from google map (as true value=3978meter) and total distance values obtained from the proposed system application (as observed value=4057meter) to find the Percentage Accuracy of the proposed system using the percentage accuracy formula as shown below:

$$\%A = 100 - |(Tv - Ov) / Tv * 100| \quad (6)$$

Where:
%A is the percentage accuracy, Tv is the true value or theoretical value, or is the observed or measured value
%A=100-|(3978-4057)/3978*100|
%A= 98.014
He percentage accuracy of the proposed system is equal to (98.014%).

5. CONCLUSION AND FUTURE WORK

This paper introduces a prototype of an electric vehicle and charging stations monitoring system based on IoT using a Gps module and Microcontroller ESP32. This system uses the data from GPS to get real-time information such as the current location coordinate of the EV, Speed, the nearest CS of EV number and distance between the EV and nearest CS, as well as the estimation time to arrive at the nearest CS. This system provides real-time information to reduce the lost time in searching the CS location. Also, help users select the CS by displaying the nearest one to reduce the power consumption of EV lost by searching for available CS. Estimation time to arrival calculated depending on distance and speed of EV. Future work: developed system by adding some factors to select recommendation CS for the user as the number of EV in the station, available charging capacity also using Edge computing in addition to cloud computing using IoT technology.

REFERENCES

[1] S. Urooj, F. Alrowais, Y. Teekaraman, H. Manoharan, and R. Kuppusamy, “IoT based electric vehicle application using boosting algorithm for smart cities,” *Energies*, vol. 14, no. 4, Feb. 2021, doi: 10.3390/en14041072.

[2] C. E. S. Thomas, "Transportation options in a carbon-constrained world: Hybrids, plug-in hybrids, biofuels, fuel cell electric vehicles, and battery electric vehicles," *Int. J. Hydrogen Energy*, vol. 34, no. 23, pp. 9279–9296, 2009, doi: 10.1016/j.ijhydene.2009.09.058.

[3] L. Dickerman and J. Harrison, "A new car, a new grid: The electric car is back (with help from new batteries, a smarter grid, and uncle sam)," *IEEE Power Energy Mag.*, vol. 8, no. 2, pp. 55–61, 2010, doi: 10.1109/MPE.2009.935553.

[4] P. Muralikrishnan and M. Kalaivani, "IOT based electric vehicle charging station using arduino uno," *Int. J. Adv. Sci. Technol.*, vol. 29, no. 6, pp. 4101–4106, 2020.

[5] S. Nutwong, E. Mujjalinvimut, S. Maiket, T. Khemkhaeng, and N. Wiraphonsawan, "Position detection with online monitoring system for wireless charging of electric vehicles," *Proceeding 2021 9th Int. Electr. Eng. Congr. iEECON 2021*, 2021, pp. 173–176, doi: 10.1109/iEECON51072.2021.9440303.

[6] G. F. Savari, V. Krishnasamy, J. Sathik, Z. M. Ali, and S. H. E. Abdel Aleem, "Internet of things based real-time electric vehicle load forecasting and charging station recommendation," *ISA Trans.*, vol. 97, pp. 431–447, 2020, doi: 10.1016/j.isatra.2019.08.011.

[7] M. Vrazic, D. Vuljaj, A. Pavasovic, and H. Paukovic, "Study of a vehicle conversion from internal combustion engine to electric drive," *ENERGYCON 2014 - IEEE Int. Energy Conf.*, pp. 1544–1548, 2014, doi: 10.1109/ENERGYCON.2014.6850628.

[8] P. Dost, P. Spichartz, and C. Sourkounis, "Charging behaviour of users utilizing battery electric vehicles and extended range electric vehicles within the scope of a field test," *2015 Int. Conf. Renew. Energy Res. Appl. ICRERA 2015*, vol. 54, no. 1, 2015, pp. 1162–1167, doi: 10.1109/ICRERA.2015.7418592.

[9] S. Kiran and S. B. Sriramoju, "A study on the applications of IoT," *Indian J. Public Heal. Res. Dev.*, vol. 9, no. 11, pp. 1173–1175, 2018, doi: 10.5958/0976-5506.2018.01616.9.

[10] Z. M. Yusoff, Z. Muhammad, M. S. I. M. Razi, N. F. Razali, and M. H. C. Hashim, "IOT-Based smart street lighting enhances energy conservation," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 20, no. 1, pp. 528–536, 2020, doi: 10.11591/ijeecs.v20.i1.pp528-536.

[11] H. G. Hamid and Z. T. Alisa, "Survey on IoT application layer protocols," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 21, no. 3, pp. 1663–1672, 2021, doi: 10.11591/ijeecs.v21.i3.pp1663-1672.

[12] B. V. S. Padmaja, V. R. Kolluru, and S. S. Kota, "IoT based implementation of vehicle monitoring and tracking system using node MCU," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 6, pp. 446–450, 2019.

[13] M. Kushwah and R. Rani, "Home automation and security system using internet of things," *2019 Int. Conf. Vis. Towar. Emerg. Trends Commun. Netw.*, 2019, pp. 235–258, doi: 10.4018/978-1-5225-7335-7.ch012.

[14] A. Maier, A. Sharp, and V. Yuriy, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things," *2017 Internet Technol. Appl.*, pp. 143–148, 2014.

[15] N. T. Morallo, "Vehicle tracker system design based on GSM and GPS interface using arduino as platform," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 23, no. 1, pp. 258–264, 2021, doi: 10.11591/ijeecs.v23.i1.pp258-264.

[16] Y. Dian Harja and R. Sarno, "Determine the best option for nearest medical services using Google maps API, Haversine and TOPSIS algorithm," *2018 Int. Conf. Inf. Commun. Technol. ICOIACT 2018*, vol. 2018-Janua, 2018, pp. 814–819, doi: 10.1109/ICOIACT.2018.8350709.

[17] P. Dauni, M. D. Firdaus, R. Asfariani, M. I. N. Saputra, A. A. Hidayat, and W. B. Zulfikar, "Implementation of Haversine formula for school location tracking," *J. Phys. Conf. Ser.*, vol. 1402, no. 7, 2019, doi: 10.1088/1742-6596/1402/7/077028.

[18] M. Babiuch, P. Foltynek, and P. Smutny, "Using the ESP32 microcontroller for data processing," *Proc. 2019 20th Int. Carpathian Control Conf. ICCC 2019*, 2019, pp. 1–6, doi: 10.1109/CarpathianCC.2019.8765944.

[19] A. Kurniawan, *Internet of Things Projects with ESP32: Build exciting and powerful IoT projects using the all-new Espressif ESP32*. Packt Publishing Ltd, 2019.

[20] D. Satria, S. Yana, R. Munadi, and S. Syahreza, "Prototype of google maps-based flood monitoring system using arduino and GSM module," *Int. Res. J. Eng. Technol.*, vol. 4, no. 10, pp. 1044–1047, 2017, [Online]. Available: www.irjet.net.

[21] L. K. Hema, R. Indumathi, Prabhakaran, and D. Kumari, "Handheld tourist guidance system using GPS," *Mater. Today Proc.*, vol. 47, pp. 351–354, 2021, doi: 10.1016/j.matpr.2021.04.561.

[22] P. Seneviratne, *Hands-on internet of things with Blynk build on the power of Blynk to configure smart devices and build exciting iot projects*, Packt Publishing Ltd, 2018.

[23] P. Serikul, N. Nakpong, and N. Nakjuatong, “Smart farm monitoring via the Blynk IoT platform,” *2018 Sixt. Int. Conf. ICT Knowl. Eng.*, 2018, pp. 70–75.

[24] D. A. Prasetya, P. T. Nguyen, R. Faizullin, I. Iswanto, and E. F. Armay, "Resolving the shortest path problem using the Haversine algorithm," *J. Crit. Rev.*, vol. 7, no. 1, pp. 62–64, 2020, doi: 10.22159/jcr.07.01.11.

[25] T. A. Salih and N. K. Younis, "Designing an intelligent real-time public transportation monitoring system based on IoT," *OALib*, vol. 08, no. 10, pp. 1–14, 2021, doi: 10.4236/oalib.1107985.

BIOGRAPHIES OF AUTHORS

Emad A. Mohammed was born in Babylon, Iraq in1972. He received a Bachelor of Science degree in Electrical Engineering from University of Technology, Baghdad Iraq in 1994, MSc. degree in Communication Engineering from University of technology, Baghdad Iraq in 2003 and PhD. Degree in Computer Networks from Mosul University, Mosul Iraq in 2016. He is currently working as a head of the Computer Techniques Engineering Department, Technical Engineering College, Northern Technical University, Iraq. He authored 13 published papers with H-index2. His research's interests include wireless networks, CNN, IoT, and network security. He can be contacted at email: e.a.mohammed@ntu.edu.iq.

Mahmood Hameed Qahtan BSc. of Engineering in computer technology engineering in 2012 from Technical Engineering College/Mosul, now he is a post-graduate student for the master's degree in Computer Engineering in Technical Engineering College/Mosul. He can be contacted at email: mahmood.hameed@ntu.edu.iq.

Ahmed J. Ali is a lecturer at Dep. of power engineering technology/ Engineering Technical College/Mosul. He has get BSc and MSc. From Al-Thechnologia University/ Baghdad in 1995 and 2004 respectively in Electrical Engineering. He has finish his Ph.D. in Electrical Machines from university of Mosul in 2012. He has been Bunsvuer Univesity /Germany in Research Scholarship for 6 months during his Ph.D. studying period. His filed of work concentrate on Adaptive and Intelligent control Systems, Electric machines Design and Drives, wireless power transfer, FEA. Now he is Head Dep. of Power Engineering Technology at Engineering Technical College/Mosul. He can be contacted at email: ahmed.j.ali@ntu.edu.iq.