Development of Telematics Control Unit for Electric Vehicles

S J Ruthvik[1]

[1]S J Ruthvik, Department of Electronics and Instrumentation Engineering, R V College of Engineering, Bengaluru, Karnataka, India

---***---

Abstract - *The primary objective is to develop a system that enables control, tracking and visualization of the vehicle location and other parameters from a remote location. In order to realize such a system, a technology called Internet of Things has been adopted which acts as a framework for development and implementation of the product providing an end-to-end solution. Various state of the art protocols such as MQTT, CAN, USART, etc., and platforms such as AWS have been used in order to create connectivity between the hardware devices, cloud platforms and end user-interface applications. Various parameters from the vehicle are obtained and transmitted onto the cloud for easy analysis and tracking of the vehicle and its health, thus, ensuring real time monitoring and safety of the vehicle in on-road or off-road conditions.*

Key Words*: *Internet of Things, Electric Vehicles, Green tech, Vehicular Network, CAN, MQTT, Amazon Web Services, REST API, EV, Electric Car, Buck Converter, GSM

1. INTRODUCTION

Irresponsible use of natural resources by the human race has caused significant and irreversible repercussions on the environment, even with the concentrated energy conservation and increased efficiency of present systems, basic energy sources such as fossil fuels are rapidly decreasing. As crude oil prices have hit the skies and global awareness of environmental issues has grown, the development of newer technologies that are less dependent on fossil fuels and which are more environment friendly is the need of the hour. The move to a greener future especially in automobiles has been aided by the invention of Hybrid Electric Vehicles that run on rechargeable batteries. These batteries are environmentally friendly and comparatively very less dependent on fossil fuels. The invention of electric vehicles has given rise to a greener and more environmentally friendly transportation technology which is the need of the hour in order to ensure the planet is safe for the future inhabitants. In today's fast paced world, technology is rapidly changing, with time efficiency being one of the major motives for development of newer technologies. Transportation being one of the major sectors today has a great demand for fleet management systems in order to track, control and visualize the vehicle and their parameters in real time. Hence, development of a Telematics Control Unit that connects the vehicle and the user wirelessly is absolutely necessary.

1.1 What is an Electric Vehicle?

An Electric Vehicle (EV) in distinction to the normal vehicles comprises an electric motor that is responsible for the operation of the vehicle instead of an Internal Combustion Engine (IC) that generates power by burning a mixture of fuel and gases inside the vehicle. Electric Vehicles can be classified majorly into two types namely:

- **Battery Electric Vehicles (BEVs)**: BEVs consist of huge battery packs that supply power to the EVs. The packs have to be charged from time to time using either a wall outlet or charging equipment, which are also termed as Electric Vehicle Supply Equipment (EVSP)
- **Fuel Cell Electric Vehicles (FEVs)**: FEVs are vehicles that consist of fuel cells along with a combination of a small battery or super capacitor to provide power to the electric motor. Most common ones are Hydrogen fuel cells and Aluminum fuel cells.

Electric Vehicles run on electricity from the batteries or fuel cells, which eliminates the emission of exhaustion from tailpipes which reduces pollution by a considerable amount.

1.2 Telematics Control Unit (TCU)

Telematics Control Unit or TCU in short is an on-board embedded system that connects the vehicle to a cloud platform and is responsible for the control and monitoring of wireless tracking, diagnostics and communication of the EV via a cellular network. TCU is responsible for the collection and transmission of telemetry data from the vehicle, such as position, speed, connectivity quality etc., and other various on-board parameters such as State of Charge (SoC), Charging details, Battery Management System (BMS) Parameters : Cell voltage, current, temperature etc., are achieved by interfacing various subsystems of the vehicle over control and data-buses. Major components of a TCU are as discussed below.

Sub Systems of TCU:

1. **Telematics System Power -** Major function of this system is to ensure that all the subsystems of the TCU receive the required minimum power supply for optimal operation. The system comprises multiple DC/DC converters including buck, boost, Linear regulators etc., Load switches are also added

as a part of the system which provides control over supply and distribution of power to each of the other subsystems present in the TCU.

2. **Off - Battery Power and Safety -** This sub-system's responsibility is to ensure safety of the TCU in case of any short circuits or system breakdowns. The system is also responsible for immediate supply of power to the TCU in case of unforeseen circumstances or low power input from the vehicle battery system. As a safety feature the unit includes Reverse battery protection as well.

3. **Internal Health Monitoring and Statistics -** Various measurement devices such as Temperature sensors, Voltage and Current sensors are clustered together to form this sub-system which is responsible for monitoring the health status of each sub-system and provide the required statistics for timely analysis and threat identification.

4. **Wireless Connectivity -** It consists of Transceivers, satellite navigation and connectivity modules such as Global navigation satellite system (GNSS), Global System for Mobile Communications (GSM) etc. These modules together form the communication layer through which the TCU interacts with the cloud server. The GNSS module helps in tracking of the vehicle in real time by providing the latitude and the longitudinal coordinates

5. **RF Power and Antenna -** The system requires a robust antenna with good range to ensure a great connectivity strength that provides an interface for the wireless connection of the TCU unit. In order to obtain high efficiency of the antenna performance it is very vital to ensure that the power supply for the antenna has a very low noise, hence, RF systems are used in order to achieve the same.

6. **Micro Controller Unit (MCU) -** This serves as the brain of the TCU. It controls the operations and monitoring of each sub-system and facilitates the internal and external communication.

7. **Communication Interface -** The system is responsible for all the internal and external communication that occurs in the TCU. It consists of On-Board Diagnostics (OBD) port, Controller Area Network (CAN) transceivers which are used for interfacing the MCU of the TCU with other sub-systems and boot loading the device in case of software or firmware updates.

8. **Real Time Clock (RTC) -** It provides functional support to the MCU in order to maintain an uninterrupted flow of operations and time-keeping. General components of this unit are oscillators, RTCs etc.

9. **Memory -** External memory unit to provide additional storage space for the MCU which stores the faulty data during breakdowns, error messages, warnings and data obtained from Internal Health Monitoring and Statistics sub-system. This memory block is also used for conducting Over the Air (OTA) updates of the TCU or other nodes that are connected to the CAN bus.

10. **Cloud -** All the data transmitted from the TCU has to be stored on to a database on the cloud. The data stored on the cloud is sorted and tabulated as per the requirement, this is then retrieved by the front-end application such as web page or mobile application for real time monitoring and graphical visualization of the data.

11. **Front End -** The front-end is either a web page for mobile application that visualizes the data from the server graphically. It also includes many other options for data monitoring, control, and location of the vehicle on the google maps in real time.

1.3 Assumptions made / Constraints

Considering the product developed will be deployed in real time the assumptions and constraints during the design and analysis of the product are very limited.

The assumptions made for the execution of the project is as follows:

- The product is not affected by any variations in the environmental conditions such extremely high or low temperature and humidity.
- The connection between the TCU and the TCU server is very strong and there are rare chances of dysconnectivity.

Some of the major constraints of the product developed are:

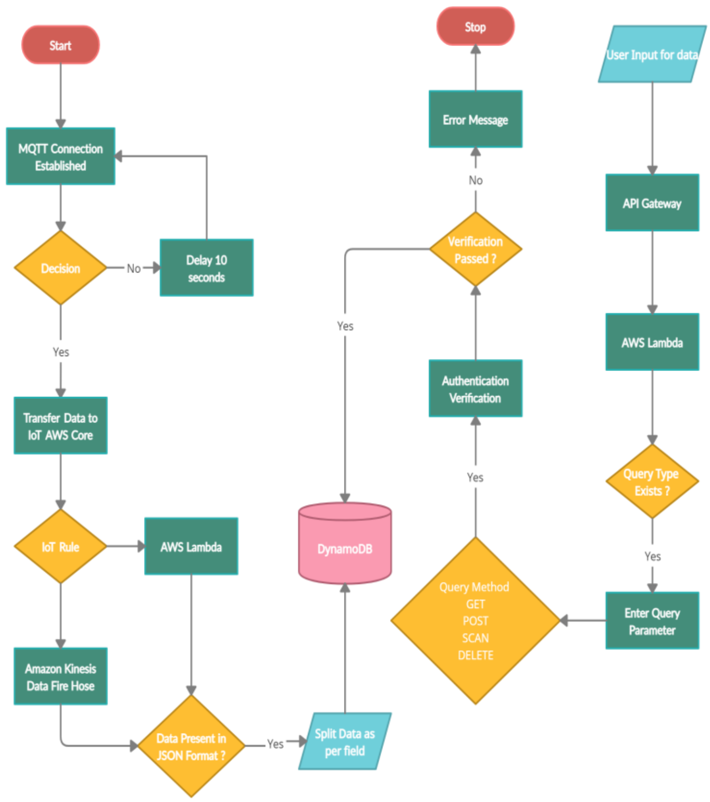
- Placement and positioning of TCU antennae is such that it produces maximum radiation in order to achieve strong connectivity.
- The product designed has to be portable.

2. METHODOLOGY / ALGORITHM

A summarized algorithm explaining the product working mechanism is explained below. The algorithm provides an outline of the flow that is followed for the technical implementation and integration of various technologies that are used in the product.

- Step 1: Integrate MCU with MC60 evaluation board, using USART communication protocols.
- Step 2: Activate the GSM module using AT Commands.
- Step 3: Establish connection with AWS IoT-Core.
- Step 4: Request data from the MCU.
- Step 5: Transmit data parsed in json format to the AWS IoT-Core.
- Step 6: Data received in JSON format is passed onto Lambda function for sorting as per requirement.
- Step 7: Latitude and longitude values are sent separately as a second message.
- Step 8: Latitude and longitude values are passed onto AWS Firehose.
- Step 9: Data from Firehose and lambda function is parsed into a dynamic cloud database called DynamoDB.
- Step 10: The database stores all the values at regular intervals of time in an orderly fashion.
- Step 11: REST API is created to access the data from the database in a secured fashion.
- Step 12: Database is integrated with the front-end dashboard development tool.
- Step 13: Using SQL Query and REST APIs the data is obtained on the dashboard.
- Step 14: The obtained data is mapped as charts and graphs for easy visualization and monitoring of the values in real time.

3. FLOWCHART



4. IMPLEMENTATION

The product being developed consists of both hardware as well as software components. Various softwares and platforms have been used for configuring and integrating all the subsystems of the TCU. As per the IoT framework the product implementation has been divided into three major aspects namely:

1. Hardware Subsystems.
2. Back-end / Server-end.
3. Front-end / User-end.

4.1 Hardware Subsystems

The various subsystems of the Telematics Control Unit as discussed earlier have to be integrated with each other. The following section explains how each subsystem is integrated with each other and how central control operations are managed by the MCU.

MCU is responsible for the control of the transmission and reception of all the data between the Electric Vehicle and AWS. The MC 60 module is integrated with the MCU using USART communication protocols. MC 60 module is responsible for establishing the connection between the hardware systems and the cloud via GSM module, the communication is carried over AT Commands. The following algorithm explains the flow of the interface.

- Step 1: MCU sends a CAN Command to request data from the BMS.
- Step 2: BMS transfers the data via CAN BUS to the CAN transceiver.
- Step 3: CAN transceiver encodes the data and transfers it to the MCU.
- Step 4: MCU obtains date and time stamp using RTC unit.
- Step 5: MCU binds all the required data in JSON format.
- Step 6: MCU initiates USART communication with the GSM module.
- Step 7: GSM is initiated using AT Commands.
- Step 8: GSM establishes active connection.
- Step 9: MQTT protocol initiated.
- Step 10: All required certificates are uploaded.
- Step 11: Secure Sockets Layer (SSL) encryption layer is enabled.
- Step 12: Direct access to the destined Uniform Resource Locator (URL).
- Step 13: Subscribe or Publish data to the specific topic
- Step 14: Perform data transmission and reception.

4.2 Back-end / Server-end

The back-end implementation is carried out on the AWS cloud platform. The different tools and platforms required for the implementation have been discussed above. The following section discusses the implementation methodology and flow that has been followed for the back-end development.

- Step 1: Establish MQTT connection, retry after 10 seconds in case of failure.
- Step 2: Transfer data to AWS IoT core.
- Step 3: Check rules in IoT rule and split data between AWS lambda and Firehose for formatting and parsing.
- Step 4: Check for data format.
- Step 5: If data is present in JSON format, split and sort the data as per requirement.
- Step 6: Store sorted data in DynamoDB.
- Step 7: Check for data requests from the user end.
- Step 8: Invoke REST API for data fetching.
- Step 9: Invoke the associated lambda function based on the API request.
- Step 10: Select API query type as per the user requirement.
- Step 11: Enter a query parameter which acts as the address to the data source.
- Step 12: Verify API authentication, if verification is passed connect to database, else invoke error code.
- Step 13: Perform action on the database as per the query method.
- Step 14: In case of authentication failure, stop the API gateway in order to protect the database.
- Step 15: Transfer data onto the web page and complete the user request

4.3 Front-end / User-end

In this project the front-end is a visual IoT dashboard which has been implemented using an open source platform where the data is obtained by invoking the REST API that has been created for the database. The obtained data is then represented in graphical forms for easy monitoring and visualization. The platform has an integrated google maps feature that allows the users to connect the latitude and longitude location using google maps API and the REST API. Based upon the location data obtained from the server.

5. RESULTS AND DISCUSSION

Alpha testing is the first end-to-end testing of a product to ensure it meets the business requirements and functions correctly. It is typically performed by internal employees and conducted in a lab / stage environment. In layman terms "an alpha test ensures that the product really works and does everything it's supposed to do". An alpha test is an initial opportunity to evaluate the performance and comprehensive functionality of a given product release. Some of the results of the test are as discussed in brief.

5.1 Data Transmission and Reception

The first test to be conducted is the connection between the hardware components and the cloud platform. Since we are using the MQTT protocol for transmission and reception of the data the MCU has to establish an active connection with the AWS IoT Core. The following graphs show a visual representation of various parameters and analysis such as successful connections, failed connections, data published to the cloud, rules executed, etc.

5.2 Database and API integration

All the data that is obtained during the process is stored on the cloud database; in this case the chosen database is AWS DynamoDB. Proper functioning of the database is very important to ensure that there is no data loss. The mapping of data from the database happens with the help of REST APIs which are designed to parse data onto the web page in a specific format so that it can be visualized and monitored in real time.

5.2.1 Database Test

The graphs below indicate the various analyses conducted on the performance of the database during the testing process.

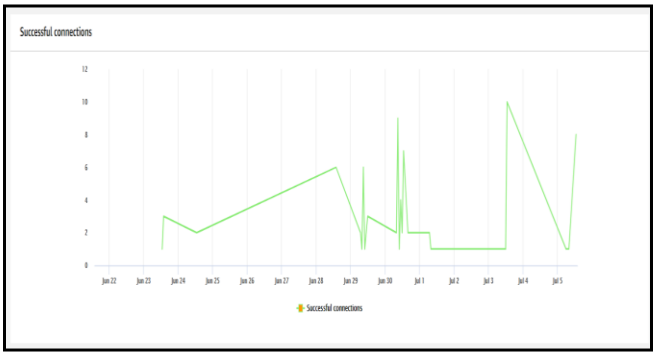


Figure 2. Number of successful connections established with AWS IoT Core

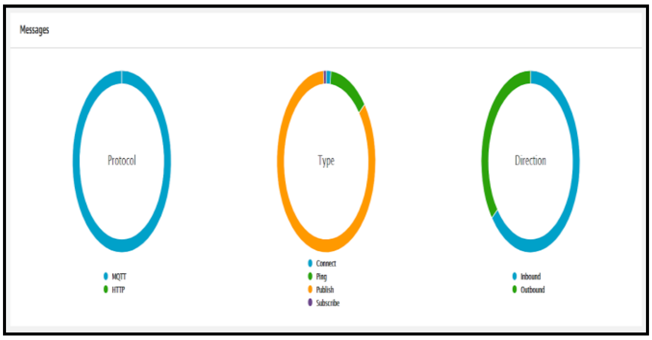


Figure 3. Types of connections established with AWS IoT Core

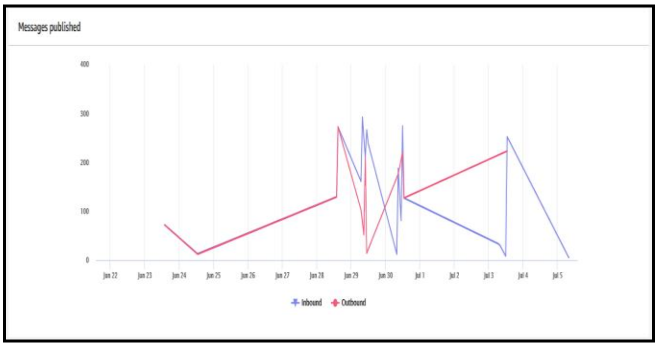


Figure 4. Number of messages published to IoT Core

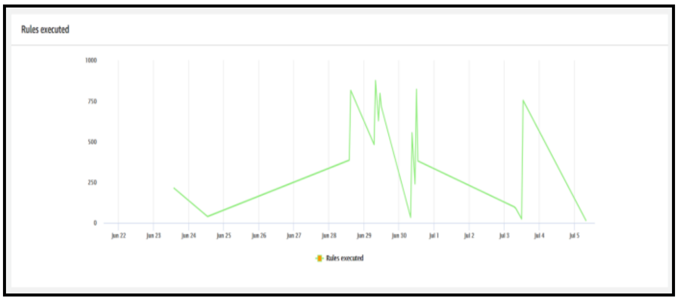


Figure 5. Number of rules executed by AWS IoT Rule

5.2 Database and API integration

All the data that is obtained during the process is stored on the cloud database; in this case the chosen database is AWS DynamoDB. Proper functioning of the database is very important to ensure that there is no data loss. The mapping of data from the database happens with the help of REST APIs which are designed to parse data onto the web page in a specific format so that it can be visualized and monitored in real time.

5.2.1 Database Test

The graphs below indicate the various analyses conducted on the performance of the database during the testing process.

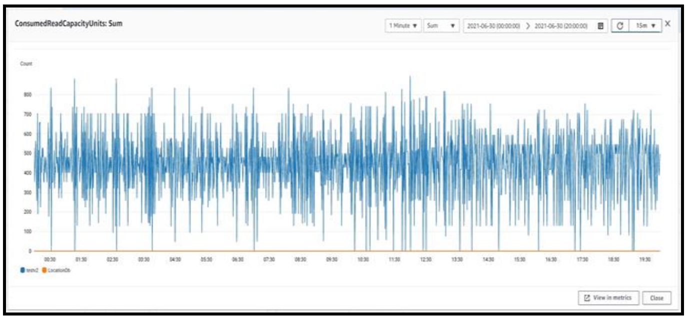


Figure 6. Sum of No. of read capacity units consumed every minute

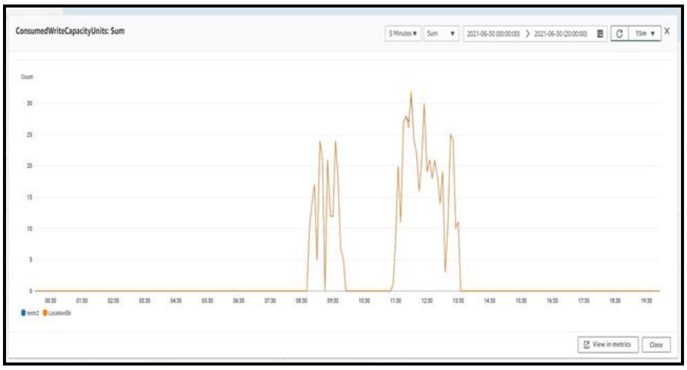


Figure 7. Sum of No. of write capacity units consumed every minute

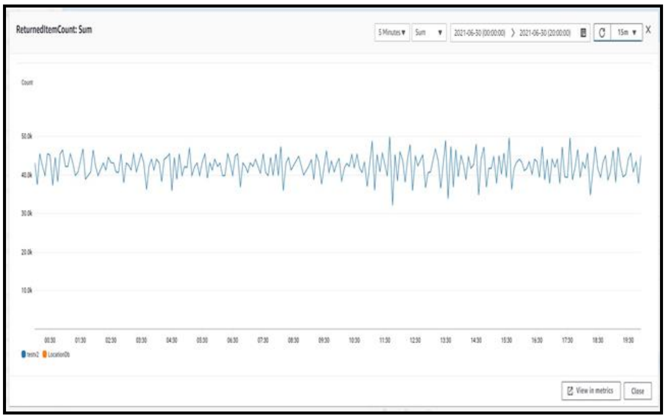


Figure 8. Sum of returned item count per minute

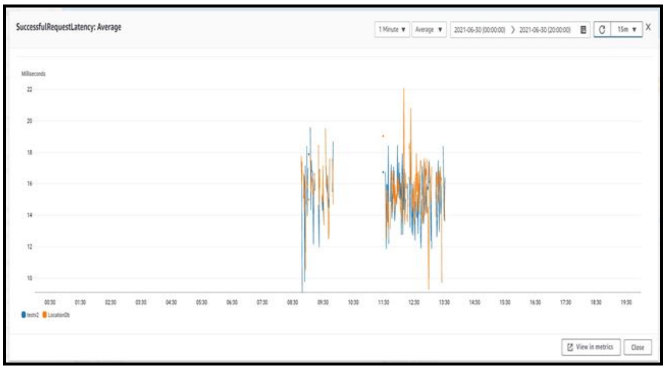


Figure 9. Average of successful requested latencies occurred every minute

5.2.2 API Testing

For testing the API and the API Gateway an open source tool called Postman is used. The figure below shows a test result obtained when the API created was run under a test case, the respective API returned all the values from the database as per the query request set for the test case.

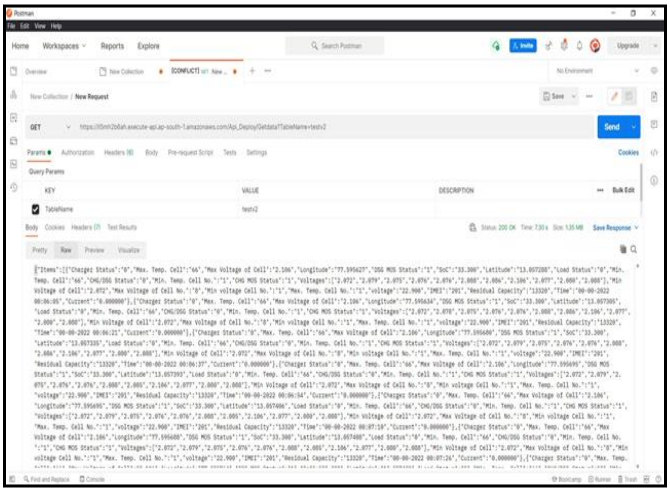


Figure 10. Test result of REST API

5.3 Front-end

The front-end monitoring visualization system is built using an open source platform and integrating it with REST APIs that have been built for parsing data from the database to the front-end platform. The following images are screenshots of the front-end system that has been developed.

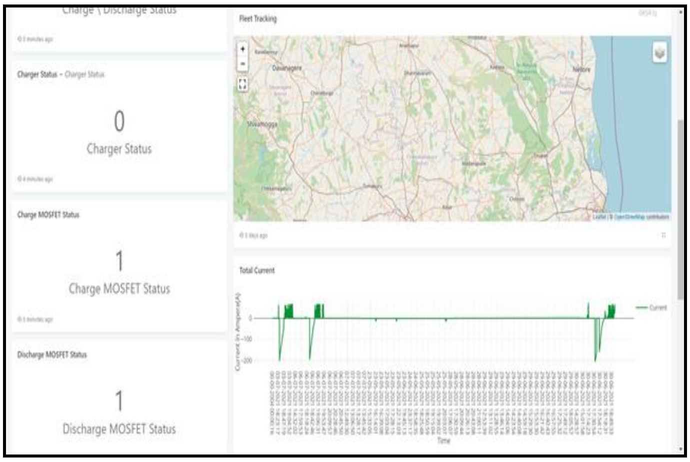


Figure 11. Screenshots of the front-end website

6. CONCLUSION AND FUTURE SCOPE

Primary objective of the project is to develop an In-House Telematics Control Unit (TCU) for real time visualization, analysis and tracking of various parameters of an Electric Vehicle (EV) such as location, SoC, voltage, residual charge, range, etc., which play a vital role in determining the battery health and condition of the Electric Vehicle, enabling the users to monitor and take actions as per the notifications.

Internet of Things (IoT) which is one of the most widely used state of the art technologies is the basis of the product that is developed. Based on the literature review it can be learnt that there are various technologies that can be used in various ways in order to develop the product and meet the objectives. Upon careful analysis and testing it was decided that GSM based communication via MQTT protocol is most optimum for the product development. On the server-side AWS platform was chosen based on the features offered and other distinct characteristics. The objectives of the product can be viewed in three parts namely: hardware implementation, server implementation and front-end user application. These classifications follow standard IoT stack.

The methodologies mentioned above were designed keeping in mind the outcome and scalability of the project. Understanding how the hardware components interact with each other, how the different software components interact with each other and interaction of the hardware components with the software components form an integral part of the development. Therefore, if chosen components don't link well, scalability and performance of the project drops drastically. However, in this project all the implementation has been carried out as per the standard IoT Stack, thus, ensuring that each component supports the other component and follows a specific error handler in case of any failures. Therefore, this product is scalable, real-time, easy to use and responsive.

Based on the results as discussed in the previous section we can conclude that the product developed is better than the existing products in the market and has successfully met the objective of real time tracking, visualization and analysis of various parameters of an Electric Vehicle.

6.1 Future Scope

The implementation of this project involves data received and processed by various hardware components and server-based database systems that are analyzed and displayed in the front-end in real-time. However, with this much influx of data over time, it would be very beneficial to perform data science and Machine Learning on the data. It is very likely that there are trends in the data we don't notice firsthand, but some analysis and oversight can prove to be beneficial for the application. Similarly, Machine Learning can be used for anomaly detection in the data.

This project is currently used only for in-house testing and analysis that is conducted by a testing engineers' team. Currently, an open source based front-end website has been built for visualization but a full stack website or a mobile application needs to be developed in order to complete the end-to-end development of the product before it gets commercialized. Technology is forever growing and we must adapt to the changes as they come. For this, the product must be scalable. But it also needs to be maintained whenever newer technologies are rolled out so that the application remains future proof.

REFERENCES

[1] A. H. Alquhali, M. Roslee, M. Y. Alias, and K. S. Mohamed, "Iot based real-time vehicle tracking system," in 2019 IEEE Conference on Sustainable Utilization and Development in Engineering and Technologies (CSUDET), 2019, pp. 265–270. doi: 10.1109/CSUDET47057.2019.9214633.

[2] J. Lee, H.-J. Kim, G.-L. Park, H.-Y. Kwak, Y.-c. Kim, and J. Song, "Electric vehicle telematics framework for smart transportation," in Advanced Communication and Networking, T.-h. Kim, H. Adeli, R. J. Robles, and M. Balitanas, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 176–182, isbn:978-3-642-23312-8.

[3] S. Amir Alavi, A. Rahimian, K. Mehran, and J. Mehr Ardestani, "An iot-based data collection platform for situational awareness-centric microgrids," in 2018 IEEE Canadian Conference on Electrical Computer Engineering (CCECE), 2018, pp. 1– 4. doi: 10.1109/CCECE.2018.8447718.

[4] A. Dubey, M. Lakhani, S. Dave, and J. J. Patoliya, "Internet of things based adaptive traffic management system as a part of intelligent transportation system (its)," in 2017 International Conference on Soft Computing and its Engineering Applications (icSoftComp), 2017, pp. 1–6. doi: 10.1109/ICSOFTCOMP.2017.8280081.

[5] G. Tricomi, Z. Benomar, F. Aragona, G. Merlino, F. Longo, and A. Puliafito, "A nodered-based dashboard to deploy pipelines on top of iot infrastructure," in 2020 IEEE International Conference on Smart Computing (SMARTCOMP), 2020, pp. 122–129. doi: 10.1109/SMARTCOMP50058.2020.00036.

[6] M. Mahmoodpour, A. Lobov, M. Lanz, P. Ma¨kela¨, and N. Rundas, "Role-based visualization of industrial iot-based systems," English, in 2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, MESA 2018, IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications ; Conference date: 02-07-2018 Through 04-07-2018, IEEE, Aug. 2018, isbn: 9781538646434. doi: 10.1109/MESA.2018.8449183.

[7] S. Ahdan, E. R. Susanto, and N. Rachmana Syambas, "Proposed design and modeling of smart energy dashboard system by implementing iot (internet of things) based on mobile devices," in 2019 IEEE 13th International Conference on Telecommunication Systems, Services, and Applications (TSSA), 2019, pp. 194–199. Doi: 10.1109/TSSA48701.2019.8985492.

[8] S. Liu, L. Guo, H. Webb, X. Ya, and X. Chang, "Internet of things monitoring system of modern eco-agriculture based on cloud computing," IEEE Access, vol. 7, pp. 37050–37058, 2019. doi: 10.1109/ACCESS.2019.2903720.

[9] K. K. Lwin, Y. Sekimoto, W. Takeuchi, and K. Zettsu, "City geospatial dashboard: Iot and big data analytics for geospatial solutions provider in disaster management," in 2019 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM), 2019, pp. 1–4. doi: 10.1109/ICT-DM47966. 2019.9032921.

[10] M. Patel and N. Chauhan, "Smart dashboard: A novel approach for sustainable development of smart cities using fog computing," in 2019 3rd International conference on Electronics, Communication and

Aerospace Technology (ICECA), 2019, pp. 632–636. doi: 10.1109/ICECA.2019.8821813.

[11] S. Sanaei, B. Majidi, and E. Akhtarkavan, “Deep multisensor dashboard for composition layer of web of things in the smart city,” in 2018 9th International Symposium on Telecommunications (IST), 2018, pp. 211–215. doi: 10.1109/ISTEL. 2018.8661092.

[12] Z. Yunzheng, Q. Shaofei, A. C. Huang, and X. Jianhong, “Smart vehicle control unit an integrated bms and vcu system,” IFAC-PapersOnLine, vol. 51, no. 31, pp. 676–679, 2018, 5th IFAC Conference on Engine and Powertrain Control, Simulation and Modeling E-COSM 2018, issn: 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2018.10.156.

[13] T. Andromeda, I. Haryanto, J. Setiawan, Hermawan, B. Nugroho, M. I. Romadhon, I. Setiawan, M. Facta, and A. R. M. Sidek, “Design of dc fast charging buck converter for lfp battery on electric car,” in 2019 6th International Conference on Electric Vehicular Technology (ICEVT), 2019, pp. 258–262. doi: 10.1109/ICEVT48285.2019.8993974.

[14] M. Poongothai, P. M. Subramanian, and A. Rajeswari, “Design and implementation of iot based smart laboratory,” in 2018 5th International Conference on Industrial Engineering and Applications (ICIEA), 2018, pp. 169–173. doi: 10.1109/IEA. 2018.8387090.

[15] A. Bhatnagar, V. Sharma, and G. Raj, “Iot based car pollution detection using aws,” in 2018 International Conference on Advances in Computing and Communication Engineering (ICACCE), 2018, pp. 306–311. doi: 10.1109/ICACCE.2018. 8441730.

[16] M. K. Ishak, O. Ali, E. A. Sirajuduin, and L. S. Qi, “Vehicle sensors programming based on controller area network (can) bus using canoe,” in 2019 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2019, pp. 1–4. doi: 10.1109/ECTICON47248.2019.8955161.

[17] W. Yin Mok, “A feasible schema design strategy for amazon dynamodb: A nested normal form approach,” in 2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2020, pp. 903–907. doi: 10.1109/IEEM45057.2020.9309967.

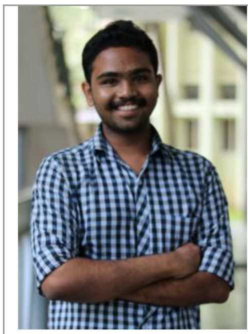
[18] M. Wagner, S. Schildt, and M. Poehnl, “Service-oriented communication for controller area networks,” in 2016 IEEE 84th Vehicular Technology Conference (VTCFall), 2016, pp. 1–5. doi: 10.1109/VTCFall.2016.7881222.

[19] H. Wenhui, H. Yu, L. Xueyang, and X. Chen, “Study on rest api test model supporting web service integration,” in 2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids), 2017, pp. 133–138. doi: 10.1109/BigDataSecurity. 2017.35.

[20] M. Ismail, A. Tarek, P. M. Plaza, D. Martin Gomez, J. M. Armingol, and M. Abdelaziz, “Advanced mapping and localization for autonomous vehicles using osm,” in 2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES), 2019, pp. 1–6. doi: 10.1109/ICVES.2019.8906306.

[21] K. Guntupally, R. Devarakonda, and K. Kehoe, “Spring boot based rest api to improve data quality report generation for big scientific data: Arm data center example,” in 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 5328–5329. doi: 10.1109/BigData.2018.8621924.

BIOGRAPHIES



S J Ruthvik is currently pursuing B.E degree in Electronics and Instrumentation from RV College of Engineering, Karnataka. His primary research interests include Internet of Things, Embedded Systems, Electric Vehicles and Battery Management Systems.