25. OTOMATİK KONTROL ULUSAL KONFERANSI (TOK2024)

12-14 Eylül 2024

KONYA TEKNİK ÜNİVERSİTESİ GELİŞİM YERLEŞKESİ

BİLDİRİLER KİTABI

Editörler

Ercan Yaldız

Ömer Aydoğdu

Akif Durdu

Serkan Budak

Mahmut Furkan Öztok

tok2024.ktun.edu.tr

ISBN:978-625-7327-07-7

Elektrikli Araçlar İçin Gerçek Zamanlı Veri Toplama Cihazı ve HMI Panel Geliştirme: Musoshi L5 EV Üzerinde Örnek Uygulama

Real-Time Data Acquisition Device and HMI Panel Development for Electric Vehicles: Sample Application on Musoshi L5 EV

Muhammed Özberk[1], Alim Kerem Erdoğmuş[1], Hande Ateş[1], Ali Kafalı[1], Okan Ulu[2], Muhammet Tarık Yılmaz[2], Muhammed Oğuz Taş[2], Muhammed Atav[3]

[1]Arge Departmanı,
ACD Veri Mühendisliği, Eskişehir
muhammedo@acd.com.tr, kereme@acd.com.tr, handea@acd.com.tr, alik@acd.com.tr

[2]Arge Departmanı,
İnovasyon Mühendislik, Eskişehir
okan.ulu@inorobotics.com, tarik.yilmaz@inorobotics.com, oguz@inorobotics.com

[3]Arge Departmanı,
Musoshi, İstanbul
Muhammed.ATAV@musoshi.com

Özetçe

Günümüzde elektrikli araçların yaygın kullanımı ve bu araçların dijital ve son teknoloji yazılımlarla desteklenmesi, araç durumlarını gösteren verilerin anlık ve doğru bir şekilde toplanması gerekliliğini ortaya çıkarmıştır. Bu çalışma, elektrikli araçlardan veri toplamak ve bu verileri bir HMI panelde göstermek için bir uç cihazın nasıl entegre edildiğini ve uygulandığını ele almaktadır. Bu amaç kapsamında, bir Jetson Nano geliştirici kiti, çeşitli sensörler ve yazılım bileşenleri kullanılarak veri toplama cihazı tasarlanmış ve test edilmiştir. Toplanan veriler CAN BUS üzerinden alınmış ve MQTT protokolü ile bir sunucuya iletilmiştir. Bu verilere FIWARE platformu kullanılarak güvenli veri transferi standardı kazandırılmış ve aktarılan bilgiler HMI panelde kullanıcılara gösterilmiştir. Yapılan örnek test sürüşleri ile elde edilen sonuçlar, veri toplama cihazı ve HMI panelinin etkinliğini ortaya koymakta ve gelecek çalışmalar için bir altyapı oluşturmaktadır.

Abstract

Nowadays, the widespread use of electric vehicles and the support of these vehicles with digital and state-of-the-art software have revealed the need to collect data showing vehicle status instantly and accurately. This paper addresses how an edge device is integrated and implemented to collect data from electric vehicles and display this data on an HMI panel. For this purpose, a data acquisition device was designed and tested using a Jetson Nano developer kit, various sensors and software components. The collected data was received via CAN BUS and transmitted to a server through MQTT protocol. These data were given a secure data transfer standard using the FIWARE platform and the transferred information was displayed to the users on the HMI panel. The results obtained from the sample test drives demonstrate the effectiveness of the data collection device and the HMI panel and create an infrastructure for future studies.

1. Introduction

Due to the depletion of fossil fuels, increasing global warming and greenhouse gas emissions, the use of more environmentally friendly alternatives, e.g. electric vehicles (EVs) are becoming increasingly common [1]. In a study conducted by Nayak et al. shown that EVs are more efficient in terms of energy saving, emission reduction and environmental protection compared to fuel-powered vehicles [2]. Due to this situation, the global EV market is expected to grow significantly, reaching 951.9 billion USD by 2030 [3].

With the widespread use of EVs, collecting and analyzing the data obtained from these vehicles has gained significant importance. Svendsen et al. developed a system running on a Linux-based embedded computer that will enable data collection from EVs. This system uses an in-vehicle sensor network and additional sensors to collect data [4]. Using the data collected from EVs, studies are carried out in many areas such as optimizing vehicle performance, battery management, and analyzing driver behavior. Al-Wreikat et al. examined the effects of variables such as ambient temperature, traffic circumstances, and driving behavior on energy consumption by using real-time data [5]. Rauf et al. proposed a machine learning model based on a smart feature selection strategy to predict the capacity loss of Li-ion batteries and stated that the proposed method reduces the prediction accuracy and absolute error when applied with machine learning algorithms [6].

Rzepka et al. presented a method for applying and tuning an extended Kalman filter for SoC estimation of Li-ion batteries. Although the work requires less effort than empirical filter tuning, it can be adapted to different battery models and cell types [7].

In this study, the developments made regarding the edge device to collect data from Musoshi's L5 model EV and the Human-Machine Interface (HMI) panel designed to display the collected data are given. Developed edge device using Jetson Nano, various sensors and software components, has successfully accomplished operations such as data collection via CAN BUS and data transmission through MQTT protocol. The data was standardized using the FIWARE and visualized on the HMI panel, demonstrating the effectiveness of the system. The tests performed demonstrate the ability of the data acquisition device and visualization system to collect and process real-time data.

2. Electric Vehicle

In the study, the L5 model EV from Musoshi's light commercial vehicle segment, shown in Figure 1, was used. This vehicle is based on a fully electric platform that supports a range of 125 km with a 15-kWh battery capacity. A payload capacity of 400 kg and a cargo volume of 1,500 liters make the L5 an ideal solution for last-mile logistics operations. The vehicle, which can reach 90% battery charge in 6 hours with a standard electrical plug, can perform in various road conditions with its ability to climb a 25% slope and reach a maximum speed of 55 km/h, and promises long-term use with 3,000 charging cycles in 5 years [8].

Figure 1: Musoshi L5 model electric vehicle

3. Data Acquisition System

The data acquisition system aims to collect completely and accurately the data from EV and send it to the CAN BUS. This process begins with the vehicle starting and then data is transmitted via the CAN BUS. The collected data is processed using predefined DBC files. The converted data is saved as CSV in the local storage and transferred to the MongoDB database.

3.1. Edge Device Design

The edge device developed to collect data from EV aims to reduce delays by enabling instantaneous processing of data. Equipped with the Korlan USB2CAN Module and 12V-5V converter, the device ensures efficient data collection. The device includes the Jetson Nano board with Ubuntu 18.04 operating system for high processing power and Waveshare Wifi Module components for fast ethernet connection. As can be seen in Figure 2, the box, specially designed and produced with a 3D printer, protects the device against external factors and ensures portability.

Figure 2: Edge device for data collection

3.2. Data Collection

The data collection process ensures that the data sent by the vehicle to the CAN BUS is collected completely and accurately. Data received via CAN BUS is collected in CSV format by the data acquisition system from the moment the device is turned on. The collected data from the EV is decoded and saved using the "*Arbitration ID*" in the DBC files. The raw and decoded data are given in Table 1.

Table 1: Raw and decoded data

<table>
<tr><th>Column</th><th>Raw Data</th><th>Decoded Data</th></tr>
<tr><td>Time_Stamp</td><td colspan="2">2024-07-03 11:45:18.409377</td></tr>
<tr><td>ID</td><td colspan="2">0x1918FF71</td></tr>
<tr><td>Extended</td><td colspan="2">True</td></tr>
<tr><td>Bus</td><td colspan="2">can0</td></tr>
<tr><td>LEN</td><td colspan="2">8</td></tr>
<tr><td>GPS</td><td colspan="2">39.75, 30.48, 864.40</td></tr>
<tr><td>Data</td><td>240, 255, 0, 0, 1, 0, 7, 247</td><td>'Torque_Measured': -1.0, 'Speed_Measured': 0, 'DC_Link_Current': 1, 'SEQ_Torque': 7, 'CS_Torque': 247</td></tr>
</table>

The general structure of the services used in the data collection system is briefly explained below.

Start Service: Starts data connection and listening processes when the vehicle is turned on.

CAN Container Service: Enables the initiation, management, filtering and termination of container services.

CAN Health Service: Monitors the health of the CAN data.

CAN Listener Service: Enables establishing a CAN connection with the EV and listening to data from the CAN.

DBC Service: Analyzes incoming data using message definitions in DBC files and directs it to the MQTT Broker service using the MQTT protocol.

Logging Service: Saves the incoming data in CSV format to the local storage and sends it to the MongoDB service.

MQTT Client Service: Checks whether container systems are working properly and makes the necessary interventions for their healthy operation. If the vehicle's ignition is off, it performs the necessary shutdown operations to turn off the data collection device.

In addition to the data received from the CAN BUS, a GPS card that communicates with Jetson Nano via USB is used to receive location data of the EV. The data obtained is transferred to the edge device and synchronized with the vehicle data by taking it to the data collection network by the *CAN Listener Service*.

3.3. Data Transmission

MQTT, an IoT messaging protocol that offers low energy consumption and uses the subscriber-publisher communication method, is used to transmit the data received from the edge device to the remote server [9]. To use MQTT, a remotely accessible MQTT Broker server is needed. Docker technology was used to install the MQTT Broker server and fast deployment was ensured by creating an isolated environment within the server, independent of other components. FIWARE middleware was used to distribute data among stakeholders in different sectors [10]. FIWARE contains context data that provides information about the state, properties and environment of an object, and the Context Broker structure is available to manage this data. In the study, context data, including updates, and queries, was managed using Orion Context Broker (OCB). An API called FIWARE NGSI has been created for the use of OCB. Thanks to this API, operations such as subscription and filtering can be performed on the data. Apart from these components, the FIWARE IoT Agent component was also used, which allows a group of devices to transfer their data to a Context Broker using their own protocols and manage it through this platform. IoT Agent can also integrate with the security layers (authentication and authorization) of the FIWARE platform.

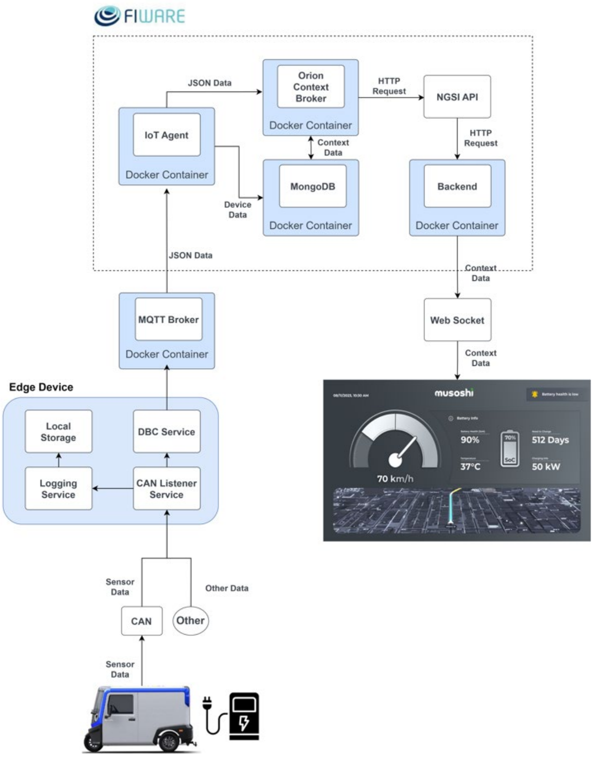

Figure 3: Data transmission architecture

FIWARE can collect data from MQTT automatically. Depending on the movement of the vehicle, an average of 600 lines of data are received per second. Although this data is instantly exported via MQTT, FIWARE cannot process the incoming data at the same speed, so the data accumulates inside and eventually fills the temporary memory, causing the system to crash. To solve this problem, a proxy service was created instead of connecting FIWARE directly to MQTT. This service decodes incoming data and adapts its frequency to FIWARE. It then transmits this back to MQTT channels. Data is now available from FIWARE using the NGSI-v2 API. Figure 3 shows the data transfer architecture.

4. Human-Machine Interface (HMI) Development

Presenting the data collected from EVs in a user-friendly interface is of great importance for real-time monitoring and analysis of vehicle status. The developed HMI panel is responsible for visualizing the data received from the vehicle and presenting it to the user effectively. In the development of the interface, "*@mui/material*", the latest version of "*Material-UI*" from React libraries, and "*Tailwind CSS*", a CSS framework that allows web developers to create customizable designs quickly and effectively, were preferred. Following the main libraries to be used in the development of the application, the necessary packages that enable the integration of some components to be used in the HMI Panel into the platform have been installed. All data displayed to the user on the platform is transferred to the frontend via "*socket.io*" after the backend connection. As shown in Figure 4, the developed panel includes information and map components.

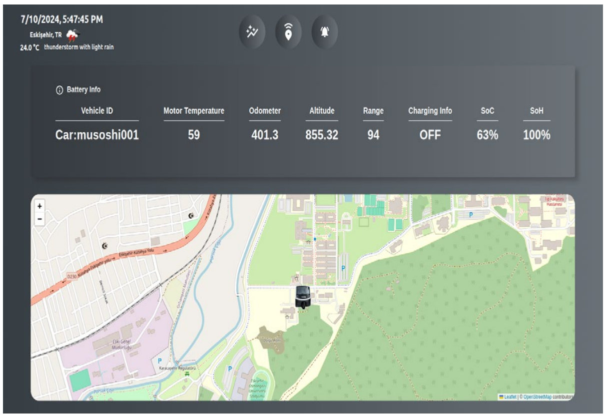

Figure 4: HMI panel

There is an area in the upper left corner of the interface that shows the date, time and weather conditions, and the information in this area changes dynamically. The area where the data is located includes "*Vehicle ID*", "*Motor Temperature*", "*Odometer*", "*Altitude*", "*Range*", "*Charging Information*", "*SoC*" and "*SoH*" information. A map component was created using React-leaflet [11], an open source and free map library, and live location information was displayed on the map using the latitude and longitude information obtained from "*socket.io*". Map data was extracted from "*OpenStreetMap*" with the "*TileLayer*" component, and the user's location was marked on the map with the "*Marker*" component.

5. Application

Musoshi's L5 model EV was used to test the operation of the developed edge device and HMI panel. Real-time tests were carried out at Eskisehir Osmangazi University Meselik campus, and data was collected through the developed edge device during each test process and the collected data was displayed on the HMI panel. Each test process was recorded as shown in Figure 5.

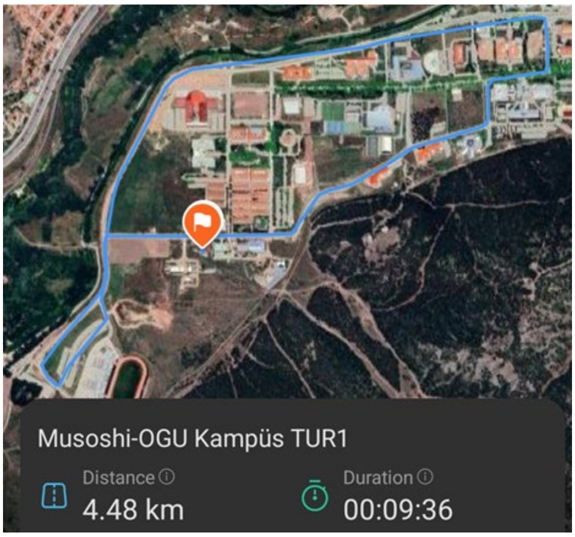

Figure 5: Sample test records

During the test, the data acquisition device collects data from the sensors in the vehicle with the help of the CAN BUS. This data is transmitted to MQTT channels. The proxy service on the remote server accesses the data by listening to these MQTT channels. The accessed data is first analyzed, then its frequency is adapted to FIWARE and transferred back to MQTT channels. FIWARE automatically listens for MQTT channels and creates a subscription for the HMI panel. The necessary data is transferred to Web Socket channels and the HMI panel listens to these channels and provides access to information about the vehicle and displays it to the user via the UI. A video showing the end-to-end functionality of the system is given in Reference [12].

6. Conclusions

Due to global warming and the increase in greenhouse gas emissions, the use of EVs which are less harmful to the environment, has become increasingly common. This has accelerated efforts to collect data from EVs and develop various applications using this data. This study demonstrates how an edge device is developed and implemented to collect data from EVs and display this data on the HMI panel in real time. The designed edge device successfully performed operations such as data collection and data transmission. Visualization of the obtained data on the HMI panel revealed the effectiveness of the system and the convenience it provides to the user. Test drives validated the effectiveness of the data acquisition device and visualization system and demonstrated the system's real-time data collection and processing capability. The collected data has led to studies in areas such as optimizing vehicle performance, ensuring energy efficiency, and analyzing driver behavior.

Acknowledgement

This paper is supported by the OPEVA project that has received funding within the Chips Joint Undertaking (Chips JU) from the European Union’s Horizon Europe Programme and the National Authorities (France, Czechia, Italy, Portugal, Turkey, Switzerland), under grant agreement 101097267. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Chips JU. Neither the European Union nor the granting authority can be held responsible for them.

Also, this study was supported by Scientific and Technological Research Council of Turkey (TUBITAK) under the Grant Number 222N269. The authors thank to TUBITAK for their supports.

References

[1] E. Helmers and P. Marx, "Electric cars: Technical Characteristics and Environmental impacts," *Environmental Sciences Europe*, 2012, 24(1).

[2] S. Nayak, et al., "Modeling and Performance Analysis of an Electric Vehicle," *International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, 23-25 December 2022, Bangolore, India.

[3] “Electric Vehicle Market Size, Share, Growth, Trends & Analysis by 2030.” Accessed: 10.07.2024, Available: https://www.marketsandmarkets.com/Market-Reports/electric-vehicle-market-209371461.html

[4] M. Svendsen, et al., "Electric Vehicle Data Acquisition System," *IEEE International Electric Vehicle Conference (IEVC)*, 17-19 December 2014, Florence, Italy.

[5] Al-Wreikat, Y., Serrano, C., & Sodré, J. R. (2021). Driving behaviour and trip condition effects on the energy consumption of an electric vehicle under real-world driving. *Applied Energy*, 297, 117096.

[6] H. Rauf, M. Khalid, and N. Arshad, "A novel smart feature selection strategy of lithium-ion battery degradation modelling for electric vehicles based on modern machine learning algorithms," *Journal of Energy Storage*, 2023, 68, 107577.

[7] B. Rzepka, S. Bischof, and T. Blank, "Implementing an extended Kalman filter for SOC estimation of a Li-ion battery with hysteresis: A step-by-step guide," *Energies*, 2021, 14(13), 3733.

[8] “Musoshi”, Available: https://www.musoshi.com/tr, Accessed: 10.07.2024.

[9] “MQTT”, Available: https://mqtt.org/, Accessed: 10.07.2024.

[10] “FIWARE”, Available: https://www.fiware.org/, Accessed: 10.07.2024.

[11] “React Leaflet”, Available: https://react-leaflet.js.org/, Accessed: 10.07.2024.

[12] “OPEVA Edge Device & HMI Panel Video” Available: https://drive.google.com/file/d/1GLsBmSA4rRsiNYrrcKheCJDtyedLhqxz/view?usp=sharing, Accessed: 10.07.2024.