DEU FMD 26(78) (2024) 356-363

Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi
Dokuz Eylul University Faculty of Engineering Journal of Science and Engineering
Elektronik/Online ISSN: 2547-958X

RESEARCH ARTICLE / ARAŞTIRMA MAKALESİ

CAN Data Transfer from Vehicle to Cloud via Smartphone

Araçtan Buluta Akıllı Telefon ile CAN Veri Aktarımı

Serhat Oldaç [1*], Yavuz Şenol [2]

[1] Department of Mechatronics Engineering, Graduate School of Natural and Applied Sciences, Dokuz Eylül University, 35390, Izmir, TÜRKİYE
[2] Department of Electrical and Electronics Engineering, Engineering Faculty, Dokuz Eylül University, 35390 Izmir, TÜRKİYE
Corresponding Author / Sorumlu Yazar *: serhat.oldac@ogr.deu.edu.tr

Abstract

Smartphones have evolved into multipurpose handheld devices and have exceeded their limits beyond communication. Apart from communication, these devices have been utilized for mobile payment, cloud-based services, artificial intelligence, Internet of Things (IoT) applications, and more. In this study, an Arduino-based system has been developed to receive commercial heavy-duty vehicle data in accordance with the SAE J1939 standard through the On Board Diagnosis II (OBD-II) connection. In addition, a smartphone application supported by graphical interface has been created to transmit processed data to the cloud. The designed system has been tested and verified by using data from an internal combustion engine passenger bus. The reliable transfer of selected data has been realized in both the smartphone and cloud environments. This study has offered a cost-effective solution for transmitting J1939 standardized vehicle data to a remote operator via a smartphone based on verified successful applications. Moreover, the created system will be a good significant resource for comparison to future applications.

Keywords: *Telematics, Intelligent Transportation Systems, Connectivity, Internet of Things, Mobile Systems, Wireless Monitoring*

Öz

Akıllı telefonlar çok amaçlı elde taşınan cihazlara dönüşmüş ve iletişimin ötesinde sınırlarını aşmıştır. İletişim dışında, bu cihazlar mobil ödeme, bulut tabanlı hizmetler, yapay zeka, Nesnelerin İnterneti (IoT) uygulamaları ve daha fazlası için kullanılmıştır. Bu çalışmada, SAE J1939 standardına uygun ticari ağır vasıta verilerini On Board Diagnosis II (OBD-II) bağlantısı üzerinden almak için Arduino tabanlı bir sistem geliştirilmiştir. Ayrıca, işlenen verilerin buluta iletilmesi için grafik arayüz destekli akıllı telefon uygulaması oluşturulmuştur. Tasarlanan sistem, içten yanmalı motorlu bir yolcu otobüsünden alınan veriler kullanılarak test edilmiş ve doğrulanmıştır. Seçilen verilerin güvenilir aktarımı hem akıllı telefon hem de bulut ortamlarında gerçekleştirilmiştir. Bu çalışma, doğrulanmış başarılı uygulamalara dayalı olarak, J1939 standartlaştırılmış araç verilerinin bir akıllı telefon aracılığıyla uzaktaki bir operatöre iletilmesi için uygun maliyetli bir çözüm sunmuştur. Ayrıca, oluşturulan sistem, gelecekteki uygulamalarla karşılaştırma için önemli bir kaynak olacaktır.

Anahtar Kelimeler: *Telematik, Akıllı Ulaşım Sistemleri, Bağlanabilirlik, Nesnelerin İnterneti, Mobil Sistemler, Kablosuz İzleme*

1. Introduction

The widespread adoption of smartphones in society has expanded their usage beyond voice communication to include a diverse range of applications, such as social media, online gaming, business tools, live traffic mapping with GPS, and more. Consequently, smartphones are now often connected to remote servers or clouds to facilitate the seamless functioning of these applications. Moreover, as an IoT device they are integrated with other smart devices with support of wireless technologies, NFC, GSM, Bluetooth, Wi-Fi, and GPS. For these reasons, some smartphones have even more RAM capacity than computers and CPUs have been improved considerably. Many different studies are targeting to take advantage of today's current mobile technology by applying different methodologies and this study has been inspired by this fact. The telematics industry mainly covers information, communication, automotive electronics, and digital content technology applications. The telematics industry is used in many applications such as GPS satellite positioning, call center, CAN bus vehicle data, general anti-theft security for vehicles and personnel, emergency rescue and road guidance, route planning, driving behavior analysis, etc. [1].

On the other hand, the automotive industry is witnessing a rapid growth in the development of autonomous driving and advanced driver assistance systems, which are interconnected with the same Controller Area Network (CAN) bus through connectivity devices. This has led the automotive society to prioritize solutions that prevent cyber threats and their potential impact on human life. In addition, governments have highlighted the significance of cybersecurity by issuing regulations and guidelines [2]. Although security is not a focus of this study, it remains a crucial aspect that requires attention in the development and implementation of automotive technologies.

Zaldivar, J, et al. developed an accident detection system on the basis of airbag-triggered and deceleration detection [3]. Parametrically, selectable in real-time sensor data is transferred to an android-based smartphone visual interface via the ELM327 Bluetooth device through the OBD-II channel.

DOI: https://doi.org/10.21205/deufmd.2024267801

Geliş Tarihi / Received: 18.06.2023
Kabul Tarihi / Accepted: 04.09.2023

In this way, they aimed to automatically call the necessary emergency numbers and send the location of the accident. Similarly, Sawant, P. R. et al. presented a cost- effective solution for taxi company owners to analyze the driving characteristics of their drivers, where the vehicle data is transferred from the OBD-II device to the server with the help of a smartphone and recorded in the database [4]. The vehicle data is transferred from the OBD-II device to the server with the help of a smartphone and recorded in the database. They don't intend only for the driver, passenger, and vehicle safety but also to analyze driving characteristics and develop efficient driving techniques. Prasad B.V.P. et al. developed a diagnostic device that reads SAE J1939 vehicle data and transmits it to the LCD display and to the phone via Bluetooth [5]. The Renesas RL78 MCU was used as a J1939 monitoring device and the Vector CANoe simulator was used as a simulation device. Some vehicle data such as engine coolant temperature, torque value, intake manifold pressure value, etc. is visually displayed on the LCD screen. The study did not develop smartphone applications and transfer data to the server or cloud but it can be expanded. Rodríguez, A. R. et al. designed an Arduino-based real-time OBD-II scanner device for diagnostic purposes [6]. The filtered and masked CAN bus messages were transferred via USB to a local computer and the values can be displayed visually. Palomino et al. developed a datalogger that scans many parameters from the OBD-II connector in an affordable way [7]. Arduino Nano, MCP2515, microSD slot, LCD Screen, and open source codes and libraries were implemented. But the scope of the study did not include the transfer of vehicle data to the smartphone or cloud.

In this study, an alternative telemetry methodology is proposed and an affordable electronic device is prototyped, realized, and tested. Thus, it can reduce the dependency on expensive GSM modules and monthly subscripted sim cards used in the conventional telemetry device. Unlike most academic research topics that focus on localized diagnostic purposes, this study offers a distinct approach by providing consecutive communication protocols and comprehensive software to ensure reliable data flow between units. Therefore, our intelligent methodology not only serves as a diagnostic tool but also enables the transmission of selectable vehicle data to cloud.

2. System Overview

2.1. Vehicle Side

Modern internal combustion engine HD vehicles are equipped with several Electronic Control Units (ECU) that coordinate and monitor internal components and devices, communicating via one or more vehicle network buses, for example, powertrain CAN bus. Control units can communicate with each other at high speed and work more synchronized. An example of this can be seen in the Figure 1, all ECUs which relays information to the vehicle's instrument display via two wires. This information includes vehicle speed, engine temperature, engine speed, current gear, brake pad status, signal indicators, fault warnings, and trip computer data etc., all of which are displayed to the vehicle user in a matter of microseconds.

OBD-II connector provides access to the vehicle to the CAN bus network as an extended standard SAE J1939. There are a large number of devices designed using J1939 protocols to monitor the vehicle data network through OBD-II by various companies [8]. This standard provides a standard protocol independent of manufacturers. The vehicle controller, Engine, Transmission, ADAS, Brake, etc. units have CAN transceivers connected at pins 6 and 14 of the OBD-II socket. The data primarily begins with the OBD-II port and is transferred to the Google sheet document by various communication protocols utilized in this transfer process.

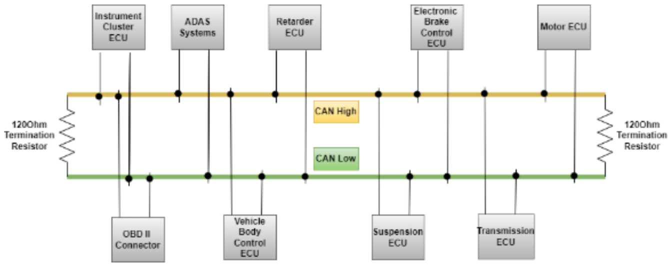

Figure 1. Heavy Duty vehicle powertrain CAN bus diagram.

SAE J1939 is a standardized and high-level protocol that runs on CAN (ISO 11898). J1939 is widely used in industrial diesel engines designed for heavy vehicles, highway vehicles, agricultural vehicles, highway equipment, construction equipment, and other vehicles. SAE released several documents consisting of standards for the automotive industry to provide identic and robust communication between the ECUs in the vehicle with uncomplicated diagnostic capabilities. As can be observed from Figure 2, the J1939 OSI reference model provides a perspective in which each guideline outlines the specific requirements for different layers. It is through adherence to this protocol that ECUs are able to effectively communicate with one another.

Documents of SAE J1939 Standard		Layers of OSI Reference Model
J1939/71 J1939/73	J1939/01 J1939/81	Application
-		Presentation
		Session
J1939/21		Transport
J1939/31		Network
J1939/21		Data Link
J1939/11 J1939/12 J1939/14 J1939/15		Physical

Figure 2. J1939 OSI reference model.

The current heavy-duty vehicles support a CAN 2.0B bitrate of up to 1 Mbit/sec in the physical and data link layer [9], whereas the communication speed utilized in our study is 250 Kbit/sec. Figure 3 illustrates the structure of the data frame that consists of 11bits standard originally, but can also be extended to 29 bits, which is called an extended format. The Start of Frame (SOF) marks the initialization of a frame. The arbitration field encapsulates an CAN identifier and a Remote Transmission Request (RTR) bit that indicates if the message is a remote application. The 4 bits are reserved for Data Length Code (DLC) which presents the length of the data in the control field. To ensure the integrity of the received message, a Cyclic Redundancy Check (CRC) sequence is included in the CRC field. The Acknowledge slot (ACK) field confirms that the transmitted message has been received successfully by at least one of the nodes [10]. This system provided a flexible structure for designers in defining name structure. In this case, new controllers can be implemented in both available formats and utilized commercially in the current automotive market [11].

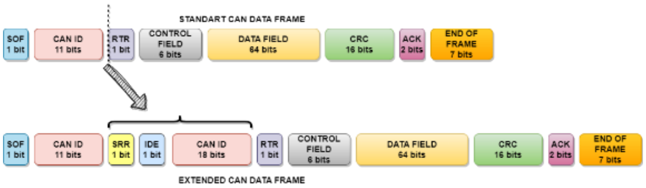

Figure 3. CAN data frame structure.

The CANalyzer software tool, developed by Vector Informatik GmbH, has established itself as a prominent choice for the automotive industry's CAN dataflow analysis needs, particularly in the context of serial line analysis. The software boasts a diverse range of features, including graphical data visualization, computer data recording capabilities, historical data simulation via the replay block, and the ability to send CAN messages to the line with function blocks, or programmable responsiveness via CAPL to facilitate the data streaming between the field and computer, the VN1630A interface device, equipped with both CAN and LIN transceivers, is utilized [12].

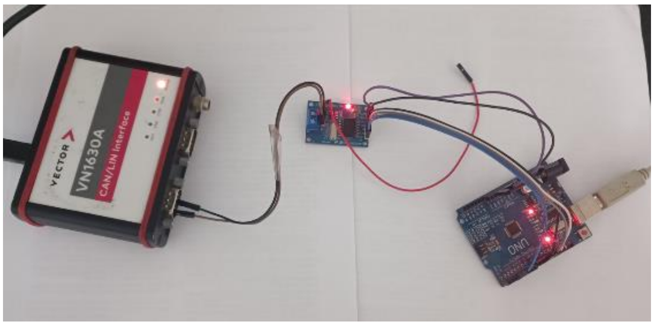

Figure 4. Test environment with VN1630A CANalyzer device.

In our study, we obtained a CAN log text file during a 3-hour city-bus drive in Izmir using the VN1630A device. As shown in Figure 4, the stored data file was subsequently employed to simulate historical data and exercise hardware and software units, thus demonstrating the utility and versatility of the CANalyzer tool in a practical setting.

The real-time recording of messages on the bus line is performed with a precision of one microsecond, and it even provides statistics on the number of frames and busload information interface. When the J1939 database file in .dbc format is imported, the acronyms specified in the SAE J1939/71 Vehicle Application Layer can be seen in the name column in Figure 5. J1939 PGN (Parameter Group Number) is a numerical identifier used in the J1939 protocol to group messages with the same destination and purpose. It consists of 18 bits, with the first 8 bits indicating the priority of the message and the remaining 10 bits indicating the PGN value. J1939 SPN (Suspect Parameter Number) is a unique identifier used to represent individual parameters within a J1939 PGN. For example, TSC1 is referring to Torque Speed Control 1. PGN of this message is zero '0' and possess the highest priority on the line since CAN bus has dominant low side and recessive high side communication structure [13]. ECUs of the driveline transmit TSC1 message to limit the engine torque and speed.

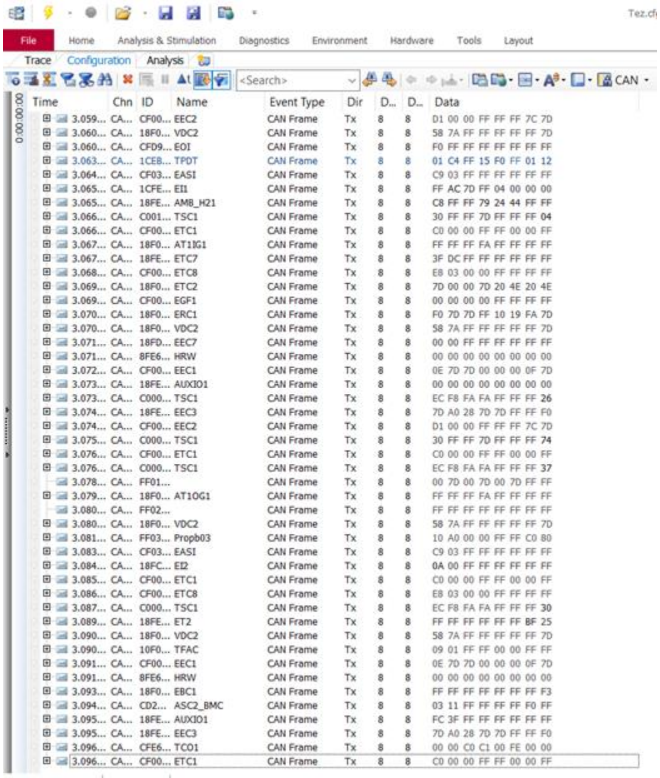

Figure 5. Trace section interface on CANalyzer tool.

2.2. Hardware Configuration and Integration

Arduino has grown in popularity in recent years as it has its own integrated development environment (IDE), integrates many libraries, and is easy to use for professionals or hobbyists. Arduino Uno was selected as a microcontroller, MCP2515 as a CAN bus transceiver, and HC05 as Bluetooth modules were selected to scan vehicle data through the OBD-II port. A smartphone application compatible with both Android and iOS operating systems has been developed with the MIT APP inventor II environment, to receive vehicle data and transmit it to Google sheets as a log text file. Figure 6 indicates a general system overview from a high-level perspective.

The MCP2515 Module serves as a controller with a high-speed CAN transceiver, facilitating communication between the MCP2515 and Arduino via the Serial Peripheral Interface (SPI) protocol [14]. Unlike the Inter-Integrated Circuit (I2C) protocol, the SPI protocol's data lines are unidirectional, and peripheral devices (slaves) do not require addresses. Instead, each peripheral device has a select leg, called Slave Select (SS), whose number is equal to the number of peripheral devices used. Each device has a separate SS pin connected to the master device, and the peripheral device whose SS line is at a low (0 volts) level begins communication with the master device. The MCP2515 acts as a data link layer in the communication system's OSI layers, generating the message frame, performing (arbitration/prioritization), and monitoring the bus state (idle/occupied). It also performs bit timing and synchronization, message validation, acknowledgment, and error detection, and decides which message to forward or reject [15].

The motivation for choosing Bluetooth for local communication in our study is that it has very low cost and high-dimensional data transmission, and most importantly, it allows real-time data transfer since we will perform high-speed communication [16].

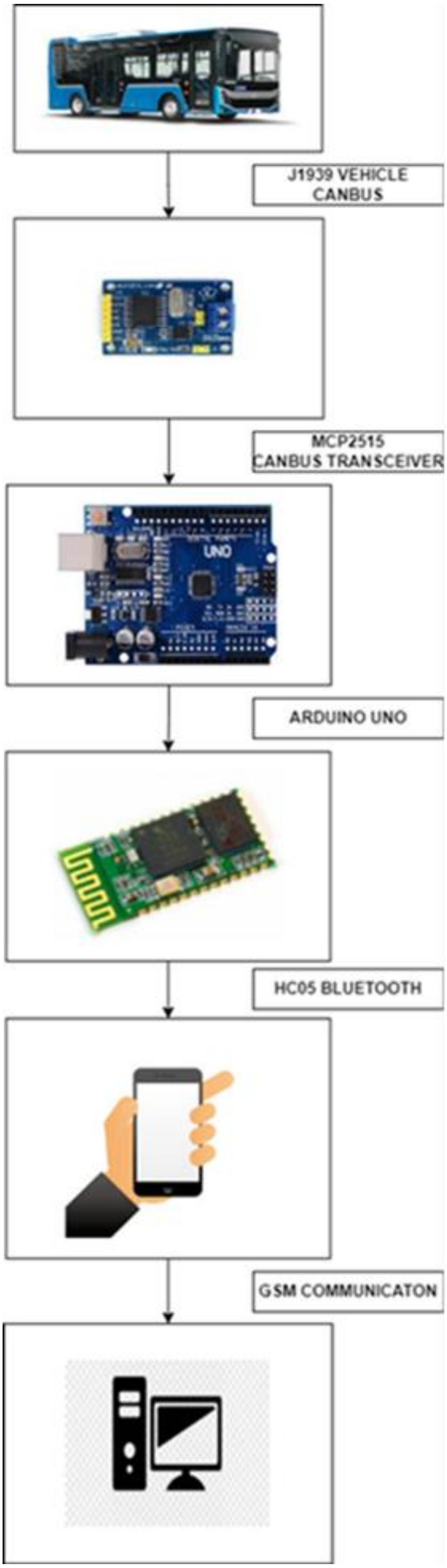

Figure 6. General system overview.

The HC-05 Bluetooth module is a popular Bluetooth serial adapter that provides a simple way to add Bluetooth functionality to microcontrollers. It uses a UART (Universal Asynchronous Receiver/Transmitter) interface to communicate with the microcontroller and supports a wide range of baud rates [17]. To configure the module, the user can enter AT commands through a terminal emulator or a microcontroller such as Arduino. HC05 Bluetooth module connects with Arduino via Tx/Rx protocol. Rx stands for "Receive" and Tx stands for "Transmit". Initially, the module should be configured by AT command mode to adjust the correct baud rate. The HC-05 can communicate in two ways as full-duplex. As depicted in Figure 7, SPI and UART connections were established between the electronics components provide communication data flow.

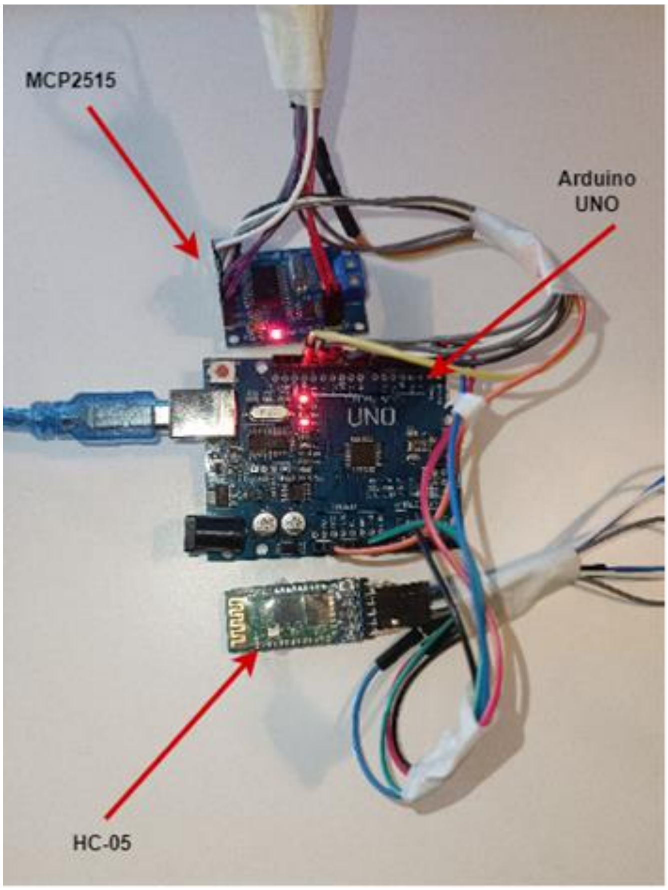

Figure 7. Hardware connections.

2.3. Software Development and Implementation

In the development of the software for data transmission to the cloud, two distinct platforms were utilized to ensure faultless integration and data communication. The Arduino software was implemented at the lowest level using the Arduino IDE environment, while the smartphone application was built using the MIT App Inventor II, a platform that has gained significant attention in academic circles. The Arduino software was designed with two primary methods, namely scanning, filtering, and converting standard or extended raw messages to actual values according to the Society of Automotive Engineers (SAE) instructions. Figure 8 provides a clear representation of the underlying communication infrastructure of the system at detailed. The diagram serves as an essential tool for visualizing the intricate interconnections and information flows between the various system components, thereby facilitating a deeper understanding of the system's functionality and performance. By mapping out the communication pathways and channels, the diagram enables developers and engineers to identify potential bottlenecks and areas for improvement, leading to enhanced system efficiency and reliability. Overall, the communication flow diagram is a valuable asset in the design, implementation, and optimization of complex communication systems.

The Arduino software comprises two primary sections referred to as setup and loop. The setup section comprises initialization codes and functions that are executed once at the first start- up, while the loop section consists of codes that run continuously in real-time cycling. In other sections, various definitions, variables, libraries, and macros can be defined to ensure optimal performance. To interact with the transceiver and retrieve data from the buffer memory the SPI library was imported, and while the MCP2515 Integrated Circuit library set of functions was implemented to enable an Arduino board to communicate with the MCP2515 CAN transceiver chip via the Serial Peripheral Interface (SPI) bus [18]. The development process of the software was focused on achieving efficient data transmission to the cloud and maintaining compatibility between the Arduino and smartphone systems.

The Setup section of the program includes several important functions that are responsible for initializing the system and configuring various hardware components. These functions include "Timer_initialize()", which sets up the system timer and other timing related parameters; "Serial_port_configuration()", which configures the serial communication ports used for transmitting and receiving data; and "CAN_configuration()", which initializes the CAN bus interface and sets up the necessary communication parameters. These functions are critical for ensuring proper operation of the system and must be called before any other functions are executed. In the loop section, the software initially begins by scanning the CAN bus line to check for any message presence. The buffer reader methods are created to receive information regarding the Data Length Code (DLC) and the message content, which can be extended, standard, remote, or error message [19]. These values are merged with the J1939/71 application layer directives using a method that transforms the raw values into actual values. Subsequently, the required data is transmitted according to the information selected by the user from the mobile application.

MIT App Inventor II is an open-source web application originally created by Google and later supported by the Massachusetts Institute of Technology (MIT) [20].

In accordance with Figure 9, MIT App Inventor II provides a user-friendly environment for developing smartphone applications without requiring advanced programming skills. The platform offers a wide range of pre-built blocks and widgets, including buttons, text fields, and sensors, that can be easily dragged and dropped onto the design canvas to create a visual interface [21, 22]. These blocks can be customized using a simple graphical interface, allowing users to modify their behavior and appearance. This approach saves time and effort, as developer can create complex applications without the need for complex coding [23].

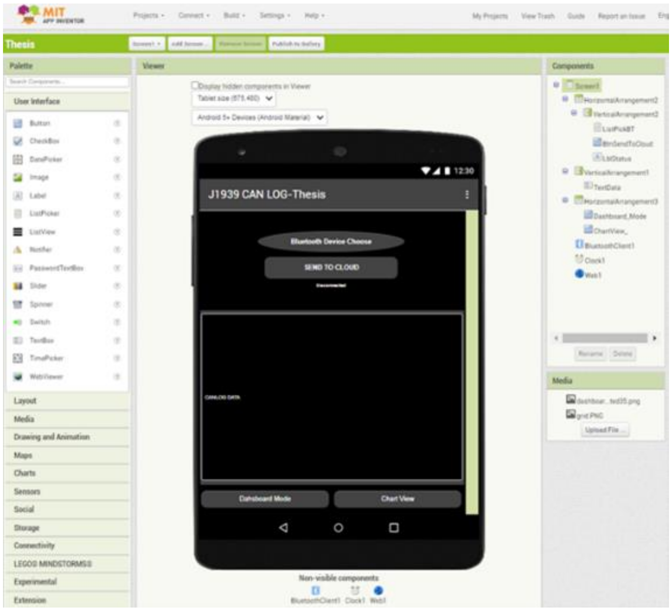

Figure 9. MIT App Inventor II application interface.

Proposed phone application presents the vehicle data to the user in a clear and easy-to- understand format. The application retrieves data from the Arduino through Bluetooth and displays it in a visually appealing format. The user can choose to view the raw data in one text field or select specific data based on J1939 PGN and SPN numbers. Together, PGNs and SPNs provide a standardized way to transmit and interpret data across a wide range of heavy-duty vehicles and equipment, enabling efficient and accurate communication between different systems and components. SPNs provide a standardized way to define and report data related to various vehicle systems and functions, such

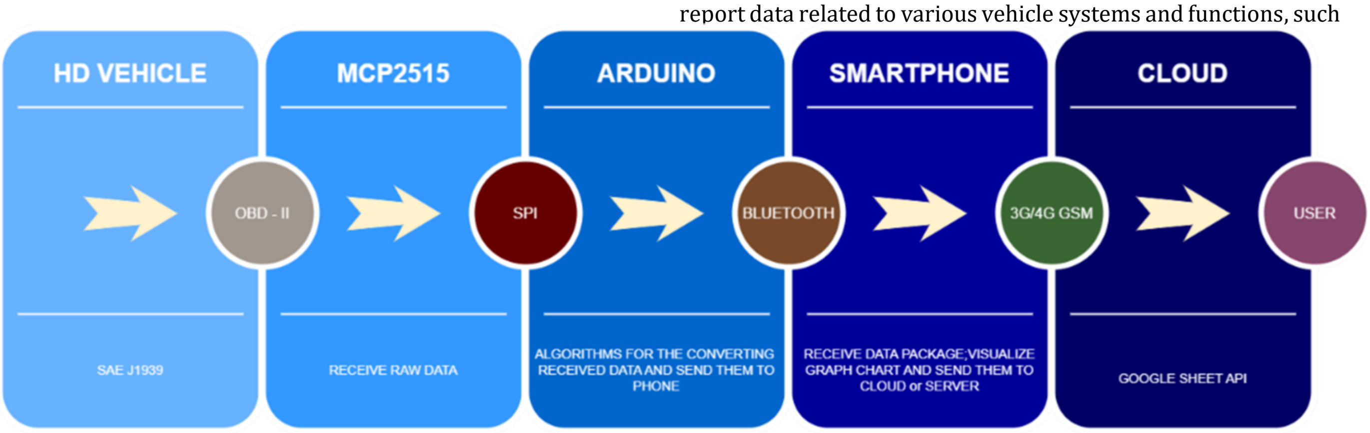

Figure 8. Data flow diagram.

as engine speed, coolant temperature, or fuel level. As evidenced by the screenshot presented in Figure 10, Each SPN is associated with a data length and resolution, allowing for consistent interpretation of the data across different vehicle systems and manufacturers. Additionally, the application can be customized to include a driver instrument panel display, which can provide real-time feedback on critical vehicle parameters such as speed, fuel level, and engine temperature etc. This type of visual display can be very useful for drivers, allowing them to monitor the vehicle's status more easily and make informed decisions about their driving behavior. Notwithstanding regulatory conditions, it is worth noting that the presence of undefined messages cannot be completely ruled out. Owing to the rapid pace of technological advancements, documentation efforts can occasionally fall behind, and in electric vehicles, it is not uncommon to encounter battery management systems that feature messages lacking clear SPN definitions. In such cases, the message can be chosen by entering values such as ID, data content, and cycle time. This approach leads to an increase in functionality.

Figure 10. SPN data selection page in smartphone application.

A significant part of the system is about reading, transmitting, storing, and sharing user data from the moving vehicle on the google sheet to the remote client. This particular application can improve safety on the road as we can identify the vehicle and the driver's ability in relation to the road. In other aspects, the driver is informed that such a malfunction may occur. For example, the coolant temperature may increase rapidly due to the car stopping before engine failure and the driver may be signaled by the phone [24]. As depicted in Figure 11, the captured data can be stored in two different formats: a graphical representation or a message format.

Google Apps Script is a language used in the cloud for creating and enhancing cloud based applications. Developers can use it to make scripts that interact with various Google services, including Google Sheets, to automate tasks such as data entry, extraction, and analysis. When used in mobile application development, Apps Script can act as an intermediary between the mobile application and the target Google Sheets document [25].

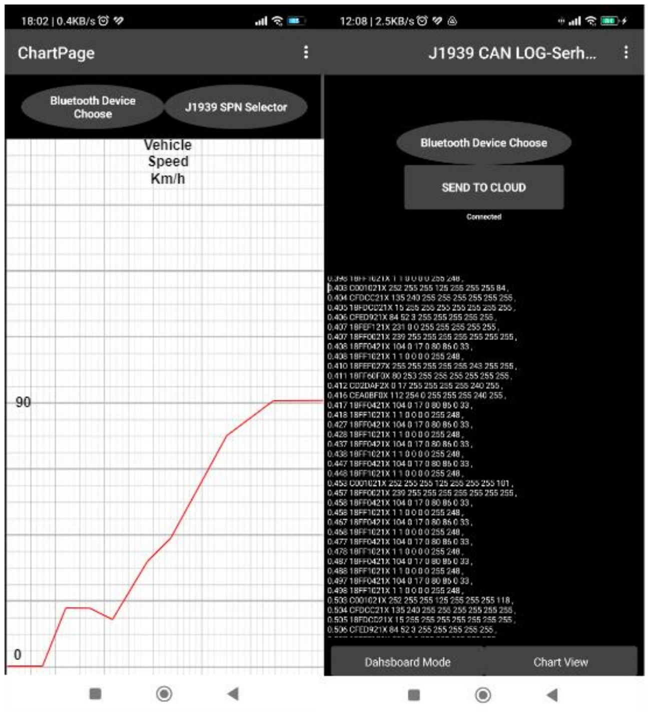

Figure 11. Chart view and CAN-Log record page.

The web app created by the script can receive data from the mobile app and then insert or update the data in the target document using the Google Sheets API. This method ensures secure and efficient data transfer as the data is processed and stored in the cloud and can be accessed from anywhere with an internet connection [26]. However, it is important to ensure that the web app and mobile application are authenticated and authorized to access the target Google Sheets document and have appropriate mechanisms in place for data validation and error handling to prevent data loss or corruption. The proposed software has ability to store data on Google Sheets allows for easy sharing and collaboration with multiple parties. It provides a convenient way for users to access and analyze data from anywhere at any time. The data can also be exported in various formats such as Word, Excel, csv, txt, etc., for further analysis or reporting. Apps Script is a language used in the cloud for creating and enhancing cloud-based applications. Developers can use it to make scripts that interact with various Google services, including Google Sheets, to automate tasks such as data entry, extraction, and analysis. When used in mobile application development, Apps Script can act as an intermediary between the mobile application and the target Google Sheets document. The web app created by the script can receive data from the mobile app and then insert or update the data in the target document using the Google Sheets API. This method ensures secure and efficient data transfer as the data is processed and stored in the cloud and can be accessed from anywhere with an internet connection. However, it is important to ensure that the web app and mobile application are authenticated and authorized to access the target Google Sheets document and have appropriate mechanisms in place for data validation and error handling to prevent data loss or corruption. The proposed software has ability to store data on Google Sheets allows for easy sharing and collaboration with multiple parties. It provides a convenient way for users to access and analyze data from anywhere at any time. The data can also be exported in various formats such as CSV, Word, Txt or Excel for further analysis or reporting.

Figure 12 indicates a basic Google Apps Script function that can be used to add a user's name to a specific sheet in a Google Sheets document. The function has two parts: doGet and doPost, which are triggered when the script is accessed using the respective HTTP method [27]. The script then opens the Google Sheets document, selects a specific sheet within the document, and calls the addUser function to add the user's name to the sheet. The user's name is passed as a parameter called "ad" in the HTTP request. Finally, the addUser function appends a new row to the sheet with the user's name.

3. Discussion

The usage of ELM327 OBD-II Bluetooth device in telemetry applications in the automotive society, along with the integration of smartphones in respect of data collection and analysis, has been extensively studied in various academic works [28-35]. It is noteworthy that the ELM327 OBD tool is not inherently compatible with the SAE J1939 protocol as its intended use is for the OBD-II protocol.

Due to its inflexible structure, instead of using this device, we have integrated our microcontroller and transceiver in a self-built structure where we created embedded software to obtain data in a flexible manner.

```
function doGet(e) {
var ss = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/
1SjrRUUOAAGqVJEZq1Ff26nJ8Fl6mKisXm4d3ZIvaayU/edit#gid=0");
var sheet = ss.getSheetByName("Sayfa1");
  addUser(e,sheet);
}

function doPost(e) {
  var ss = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/
1SjrRUUOAAGqVJEZq1Ff26nJ8Fl6mKisXm4d3ZIvaayU/edit#gid=0");
  var sheet = ss.getSheetByName("Sayfa1");
  addUser(e,sheet);
}

function addUser(e,sheet) {
  var ad = e.parameter.ad ;

  sheet.appendRow([ad]);
}
```

Figure 12. Google Apps script function.

In the present day, the cost and complexity of traditional methods for transmitting J1939 standardized vehicle data to a remote operator have become a significant challenge. To address this issue, we have designed a system that eliminates the need for expensive GSM modules and monthly payments, and instead utilizes a commonly used smartphone, connected to the vehicle's infotainment system via Bluetooth, to provide an alternative method for transmitting the data to a remote operator. Since the data transfer occurs through a smartphone, the current location of the vehicle and the presence of the driver can be easily tracked using simple extensions. Furthermore, in terms of safe driving, in emergency situations or in the event of an accident, the relevant authorities can be notified via the smartphone.

This communication network can also be expanded to include additional applications such as intelligent traffic density monitoring [36] or route calculation algorithms [37]. The technical feasibility and limitations of the system have not considered data security requirements related to cyber security. In the future, this study may serve as an alternative to traditional telemetry devices in fleet tracking systems. It can be utilized in the e-mobility, smart cities, and for smart transportation and route planning. This system exhibits potential for deployment in both heavy-duty vehicles and automobiles, with original equipment manufacturers being able to empower its capabilities for the purpose of remotely diagnosing and addressing vehicle malfunctions.

4. Conclusion

The proposed system offers a cost-effective and efficient solution for transmitting J1939 standardized vehicle data to a remote operator via a smartphone. The use of an Arduino-based software and hardware system, combined with a smartphone application and cloud-based services, provides real-time data visualization and the potential for expanded functionality, such as intelligent traffic density monitoring and route calculation algorithms. While the system does not currently include cyber security measures or support for real-time transmission of all data, future improvements can be made to address these limitations. Moreover, this system's potential extends beyond heavy-duty vehicles to include automobiles, making it a versatile solution for the transportation industry. Original equipment manufacturers can benefit from using this method for detecting and remotely resolving vehicle malfunctions. Overall, the system presented in this study demonstrates the technology's potential to improve safety, efficiency, and cost-effectiveness in the transportation industry. Further research and development in this area will lead to more innovative solutions.

Acknowledgment

The authors would like to thank BMC Otomotiv Sanayi ve Tic. A.Ş. for providing CAN bus data obtained from a prototype truck.

Ethics committee approval and conflict of interest statement

This article does not require ethics committee approval. This article has no conflicts of interest with any individual or

institution.

References

[1] Keeratiwintakorn, P., Thepnorarat, E., Russameesawang, A. 2009. Mar. Ubiquitous communication for V2V and V2I for Thailand intelligent transportation system. InNTC International Conference, Thailand.

[2] Sivanathan, A., Gharakheili, H.H., Sivaraman, V. 2020. Managing IoT cyber-security using programmable telemetry and machine learning. IEEE Transactions on Network and Service Management, Vol.17, s.60-74. DOI:10.1109/TNSM.2020.2971213.

[3] Zaldivar, J., Calafate, C.T., Cano, J.C., Manzoni, P. 2011. Providing accident detection in vehicular networks through OBD-II devices and Android-based smartphones. IEEE 36th Conference on Local Computer Networks, 4-7 October 2011, Germany, 813-819.

[4] Sawant, P.R., Pande, S. 2015. A Mobile Application for Monitoring Inefficient and Unsafe Driving Behavior. Data Mining and Knowledge Engineering. Vol.7, s.105-7.

[5] Prasad, B.V.P., Tang, J.J., Luo, S.J. 2019. Design and implementation of SAE J1939 vehicle diagnostics system. IEEE International Conference on Computation, Communication and Engineering (ICCCE) 8-10 November, China, 71-74.

[6] Rodríguez, A.R., Álvarez, J.R., Rodriguez, R.I. 2018. Implementation of an OBD-II diagnostics tool over CAN-BUS with Arduino. Sistemas y Telemática. 16(45). DOI:https://doi.org/10.18046/syt.v16i45.2747.
[7] Palomino, J., Cuty, E., Huanachin, A. 2021. Development of a CAN Bus datalogger for recording sensor data from an internal combustion ECU. IEEE International Workshop of Electronics, Control, Measurement, Signals and their application to Mechatronics (ECMSM) (pp. 1-4). DOI:10.1109/ECMSM51310.2021.9468837.
[8] Alvear, O., Calafate, C.T., Cano, J.C., Manzoni, P. 2015. Validation of a vehicle emulation platform supporting OBD-II communications. 12th Annual IEEE Consumer Communications and Net-working Conference (CCNC) 9-12 January, 880-885.
[9] Burakova, Y., Hass, B., Millar, L., Weimerskirch, A. 2016. Truck Hacking: An Experimental Analysis of the SAE J1939 Standard. Proceedings of the USENIX Workshop on Offensive Technologies (WOOT). 16:211-20.
[10] Din, M.A.C., Rahman, M.T.A., Munir, H.A., Rahman, A., Hamid, A.F.A. 2019 Development of CAN bus converter for on board diagnostic (OBD-II) system. IOP Conference Series: Materials Science and Engineering. 26-27 August, Malaysia,
[11] Specification. Bosch, C,A., Robert Bosch GmbH, Postfach. 1991 Sep;50.
[12] Pimple, P. 2018. Sniffing the Automotive CAN Bus for Real-Time Data-Logging and Real Time Diagnostics Display [Conference presentation]. International Conference on Smart Electric Drives and Power System (ICSEDPS), India, 167-170.
[13] Popa, L., Groza, B., Jichici, C. & Murvay, P. S. 2022. ECUPrint—Physical Fingerprinting Electronic Control Units on CAN Buses Inside Cars and SAE J1939 Compliant Vehicles. IEEE Transactions on Information Forensics and Security, 17, 1185-1200.
[14] Anil, D., Sivraj, P. 2020. Electric Vehicle Charging Communication Test-bed following CHAdeMO [Conference presentation]. 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), India, 1-7.
[15] Salunkhe, A. A., Kamble, P. P. & Jadhav, R. 2016. Design and implementation of CAN bus protocol for monitoring vehicle parameters [Conference presentation]. IEEE international conference on recent trends in electronics, information & communication technology (RTEICT), 20-21 May, India, 301-304.
[16] Santo Gitakarma, M. & Priyambodo, T. K. 2019. A Real-Time Smart Home System using Android Bluetooth Control Device Module [Conference presentation]. International Symposium on Electronics and Smart Devices (ISESD), 8-9 October. Indonesia, 1-7.
[17] Fenriana, I., Putra, D. S. D., Dermawan, B. & Kurnia, Y. 2022. Smart Home Prototype with HC–05 Bluetooth and RFID Modules, Based on Microcontroller. Bit-Tech, 5(2), 77-84. DOI: https://doi.org/10.32877/bt.v5i2.564.
[18] Desai S, Bhateshvar Y. 2023. Development of unified diagnostic services on CAN using MATLAB and Arduino. Materials Today: Proceedings, 72: 1935-1942. DOI: https://doi.org/10.1016/j.matpr.2022.10.157.
[19] Lakshmi, S. & Kumar, R. H. 2022. Secure Communication between Arduinos using Controller Area Network (CAN) Bus [Conference presentation]. IEEE International Power and Renewable Energy Conference (IPRECON), 19-18 December. India, 1-6.
[20] MIT App Inventor. (n.d.). https://appinventor.mit.edu/
[21] Xie, B. & Abelson, H. 2016. Skill progression in MIT app inventor [Conference presentation]. IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 4-8 September, Cambridge.
[22] Jamaluddin, A., Harjunowibowo, D., Rahardjo, D. T., Adhitama, E. & Hadi, S. 2016. Wireless water flow monitoring based on Android smartphone [Conference presentation]. 2nd International Conference of Industrial, Mechanical, Electrical, and Chemical Engineering (ICIMECE), 6-7 October. Indonesia, 243-247.
[23] Munasinghe, T., Patton, E. W. & Seneviratne, O. 2019.Iot application development using MIT app inventor to collect and analyze sensor data [Conference presentation]. IEEE International Conference on Big Data (Big Data), 9-12 December. USA, 6157-6159.
[24] Ceuca, E., Tulbure, A., Taut, A., Pop, O. & Farkas, I. 2013. Embedded system for remote monitoring of OBD bus [Conference presentation]. Proceedings of the 36th International Spring Seminar on Electronics Technology, 8-12 May. Romania, 305-308.
[25] Petrović, N., Roblek, V., Radenković, M. & Nejković, V. 2020. Approach to Rapid Development of Data-Driven Applications for Smart Cities using AppSheet and Apps Script [Conference presentation]. AIIT International conference on Applied Internet and Information Technologies, 16 October. Serbia, 77-81.
[26] DeBell, T., Goertzen, L., Larson, L., Selbie, W., Selker, J. & Udell, C. 2019. Opens hub: Real-time data logging, connecting field sensors to google sheets. Frontiers in Earth Science, 7, 137. DOI: https://doi.org/10.3389/feart.2019.00137.
[27] Thomas, A. & KP, M. S. 2019. Application of google app scripts in email for providing current awareness services to research scholars, at central university of kerala: an evaluative study. Int J Eng Appl Sci Technol, 4(6), 313-318.
[28] Ameen, H. A., Zaidan, B. B., Zaidan, A. A., Saon, S., Nor, D. M., Malik et al. 2019. A deep review and analysis of data exchange in vehicle-to-vehicle communications systems: coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions. IEEE Access, 7, 158349-158378. DOI:10.1109/ACCESS.2019.2949130.
[29] Engelbrecht, J., Booysen, M. J., van Rooyen, G. J. & Bruwer, F. J. 2015. Survey of smartphone-based sensing in vehicles for intelligent transportation system applications. IET Intelligent Transport Systems, 9(10), 924-935. DOI: https://doi.org/10.1049/iet- its.2014.0248.
[30] Türker, G. F., & Kutlu, A. 2016. Survey of smartphone applications based on OBD-II for intelligent transportation systems. Int. Journal of Engineering Research and Applications, 6(1), 69-73.
[31] Tahat, A., Said, A., Jaouni, F. & Qadamani, W. 2012. Android-based universal vehicle diagnostic and tracking system [Conference presentation]. IEEE 16th International Symposium on Consumer Electronics, 4-6 June. USA, 137-143.
[32] Meseguer, J. E., Calafate, C. T., Cano, J. C. & Manzoni, P. 2013. DrivingStyles: A smartphone application to assess driver behavior [Conference presentation]. IEEE symposium on computers and communications (ISCC), 7-10 July. Croatia, 000535-000540.
[33] Baek, S. H. & Jang, J. W. 2015. Implementation of integrated OBD-II connector with external network. Information Systems. 50:69–75. https://doi.org/10.1016/j.is.2014.06.011.
[34] Yang, Y., Chen, B., Su, L. & Qin, D. 2013. Research and development of hybrid electric vehicles CAN-bus data monitor and diagnostic system through OBD-II and Android-based smartphones. Advances Mechanical Engineering. 5:741240. DOI: https://doi.org/10.1155/2013/7412.
[35] Čabala, M., Gamec, J. 2012. Wireless real-time vehicle monitoring based on android mobile device. Acta Electrotechnica et Informatica. 12(4).
[36] Astarita, V., Festa, D. C. & Giofrè, V. P. 2018. Mobile Systems applied to Traffic Management and Safety: a state of the art. Procedia computer science. 1; 134:407-14. DOI:https://doi.org/10.1016/j.procs.2018.07.19 1
[37] Zhou, L., Chen, Q., Luo, Z., Zhu, H. & Chen, C. 2017. Speed-based location tracking in usage-based automotive insurance [Conference presentation]. IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 5-8 June, USA, 2252-2257.