
SMART AUTOMOBILE SELF-DIAGNOSIS DATA EVALUATION SYSTEM

Samuel Damilola, Akhibi

Max Planck Ring 6c, Ilmenau. Germany

Samuel Damilola, Akhibi (2022) Smart Automobile Self-Diagnosis Data Evaluation System, *European Journal of Computer Science and Information Technology*, Vol.10, No.1, pp.38-45

ABSTRACT: *The optimal performance of an automobile requires that underlining problems are checked at intervals by qualified technicians. Many a time, the checks are not carried out routinely until a fault in the vehicle raises its ugly head, this can cause disruptions of all kinds and disrupt the schedule of the owner. Over time, with the advent of technological advancement, there has been an increase in the usage of fault monitoring sensors in keeping track of the performance and maintenance requirements of automobiles. As the trend in technological advancement continues, diagnosis and evaluation techniques are now focused on a more active and user-centric system. In this paper, we would be analyzing current diagnosis and evaluation systems and looking into the research for self-diagnosing and evaluation for automotive systems*

KEYWORDS: automobiles, diagnosis, scanning, Data, System

INTRODUCTION

In this paper, we would be taking a concise look at the history, implementation and workings of certain electrical components of the automobile and how it has metamorphosed into the application of self-diagnosing vehicles. Taking a keen analysis of the technology involved and how it has taken shape over the years to what is applicable in today's automotive industry. Overall, this paper would be taking a look at the usage of this technology and how it can create a better user experience by employing self-diagnostic technology.

Overview of the Technology

Over the years the automotive industry has experienced a continued increase in the deployment and implementation of electrical systems in the motor vehicle. These systems which are overseen by central control, facilitate communication between vehicle systems and subsystems, advanced controls, increased vehicle functionality etc. These features are made possible by the interconnectivity of sensors, switches, actuators and Electronic Control Units (ECUs). In contemporary vehicles communication between the array of vehicular systems is made possible via a digital network known as the in-vehicle network (IVN) which could be of varying types; "controller area network" (CAN), "flex ray," "local interconnect," (LIN) and "media-oriented systems transport" (MOST). [2]

These networks make use of communication media such as wires, co-axials and optical fibres depending on the transmission needs and budget of the vehicle.

Controller Area Network (CAN)

The Controller Area Network CAN as it is most commonly referred to was born out of the need to effectively communicate between the increasing number of ECU. To achieve the desired effective flow of information between the ECU's which was required to optimize fuel consumption, emissions and reliability Bosch in conjunction with Intel developed the controller area network (CAN) [8]

CAN employs a message-based protocol popularly referred to as 'content-addressed implies that each message has the destination within. A priority level and the data to be transferred is embedded into the message and broadcast to every node on the network [8].

Each individual node acknowledges the received message and the condition of the receipt and is tasked with processing or discarding the information

In a typical controller area network (CAN), the specifications and protocol are governed by the Society for Automotive Engineers (SAE). The controller area network (CAN) comprises a series of mutually connected subsystems ECU (Electronic control unit) connected in a bus topology. These subsystems (ECU's) are responsible for different vehicular capabilities such as steering, braking and acceleration and constantly transfer massive amounts of messages back and forth. The CAN serves as a network controller coordinating the messages sent over the network by the numerous ECU's and also comes into play when information from an ECU is required by another ECU on the IVN. The bus topology employed for the transmission of information over the IVN is cost-effective as it reduces the requisite wiring and allows for the transmission of more information as compared with the conventional system of one wire per bit of information.

CAN bus systems are classified into two types depending on the length of the ID in the data frame:

- (i) 2.0A (11-bit ID) are composed of 11 bits for the identifier.
- (ii) Extended CAN 2.0B (29-bit ID) are composed of 29 bits for the identifier.

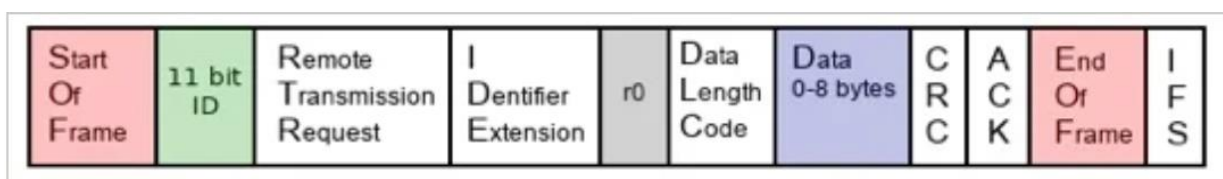


Figure 1: Standard CAN

Source: Adapted from [11]

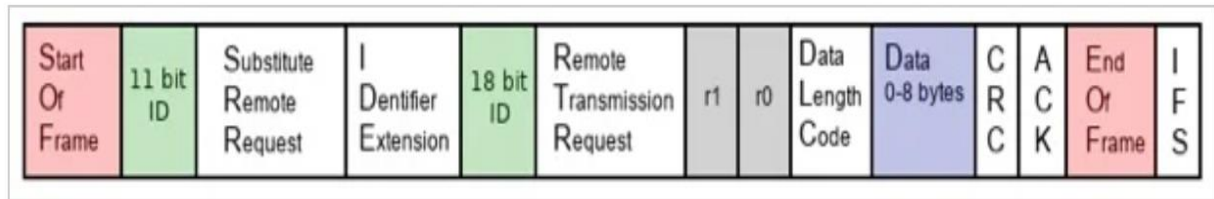


Figure 2: Extended CAN
Source: Adapted from [11]

To determine if the format is standard or extended, the extended frame format has an additional IDE bit. Its frames begin with a SOF, Start of Frame, and bit to denote the start of the transmission.

Table 1 CAN MESSAGES/FRAME TYPES

Data Frame	It is always used when information transmits
Remote Frame	Has the same properties as a Data Frame with an additional bit to signify it is a Remote Transmit Request (RTR bit)
Error Frame	It is generated by any node that detects an error. Errors can be of the five error conditions defined by CAN.
Overload Frame	Generated to give the nodes processing the message they already received more time.

Source: Adopted from [8]

CAN functions as an interconnection of nodes. The CAN is the principal means of communication within a node. These nodes function with a host processor, a CAN controller, and a transceiver.

The architecture of a node requires the host processor to derive received messages and calculate what

To send, once the host processor has stored the desired message to be transmitted into the CAN controller, the transition will commence serially in the form of bits onto the bus.

To receive, the CAN controller serially stores bits and is available if the message pertains to the host processors request.

A CAN Controller has a synchronous clock. The final part of the node structure is a transceiver often built into the CAN controller. The Transceiver translates transmitted and received messages to and from the bus and the controller respectively. While receiving, the transceiver will adjust the signal levels from the bus to those the CAN controller can process. Additional

functions of the transceiver's communication's assignment is that it also provides the CAN controller with a buffer protective circuitry layer.

(On-Board Diagnostics II) OBD-II

OBD II is a communication protocol on top of the CAN and is used to provide diagnostics information like emissions, vehicle speed, etc to vehicle owners, mechanics and dealers. OBD (On-Board Diagnostics) was made mandatory after The 1990 Clean Air Act. It requires that vehicles built after 1994 include onboard computer systems to monitor vehicle emissions and to provide repair shops and car-owners with the same tools that dealerships use to test and repair vehicles. [5]

Over the years the Clean Air Act has morphed into providing more than just emission data through onboard diagnostic tools but also making it possible to read CAN bus messages through the OBD port. This has given rise to OBD II scanners which are capable of reading diagnostic information, clearing engine errors via a wired connection to the 16-pin OBD port found in the footwell of the driver's side. These scanners work by sending a request to the ECU, this message contains a hexadecimal code with predefined parameters by the SAE and would be invariably interpreted according to one of five OBD II signaling protocols. [7]

OBD-II draws from the positive applications of OBD I and a result is a cumulation of years of research and development by the Society of Automotive Engineers (SAE), the CARB and the EPA.

With the legal necessity of ensuring automobiles abide by emission laws, the engineering bodies involved developed a system that would later be standardized across the continental United States and thereafter the rest of the world.

For it to become a global standard this requirement which would go on to become OBD II required a standardized diagnostic connector and location consistent format as well as a consistent signal transfer of information.

The standard to be enforced mandated the specifics of what was to be monitored, the extent of its monitoring and also required that in an event where the vehicle was functioning below the required threshold, a corresponding alert would be displayed on the vehicle's dashboard, most times in an ambiguous check engine indicator with the ease of performing these checks, the standardization of OBD II was quickly followed by other vehicle manufacturers outside the United States.

The usage of OBD II diagnostics that require a wired connection to a handheld device (OBD II scanner) can be simplified into a wireless device capable of OBD diagnostics paired with the owner's phone or any other wireless-enabled device to provide added functionality while allowing for simplistic usage. Thus, this project seeks to explore the implementation of OBD diagnosis through a portable device capable of functioning wirelessly.

Types of OBD-II

The present standard of Onboard diagnostics has a history spanning back to the late '80s. This technology eventually gave rise to three different types of OBD-II.

The first, and oldest standard, is ISO 9141-2. This standard was defined by the International Organization for Standardization in 1989 in response to the California Air Resources Board's call for cleaner emissions. [8]

The following version was standardized by the Society of Automotive Engineers. This standard is known as SAE J1850 Communications Standard. The SAE J1850 can be implemented in either one of two ways. The implementation occurs through a process referred to as Variable Pulse Width (VPW), or Pulse-Width Modulation (PWM).

METHODOLOGY

Implementation of Wireless OBD II

To achieve this novel feature, a new architecture to translate the ensuing information is required. Firstly, a portable mobile wireless unit will be required to interface with the vehicle's OBD II port. Its function would be to read all messages from the vehicle's OBD-II port and

thereafter translate these messages to RS232 signals. This OBD II to RS 232 translation eliminates any ambiguity involved in understanding the vehicle's protocols. Thus, eliminating the ambiguity of understanding which protocol is being used on the vehicle.

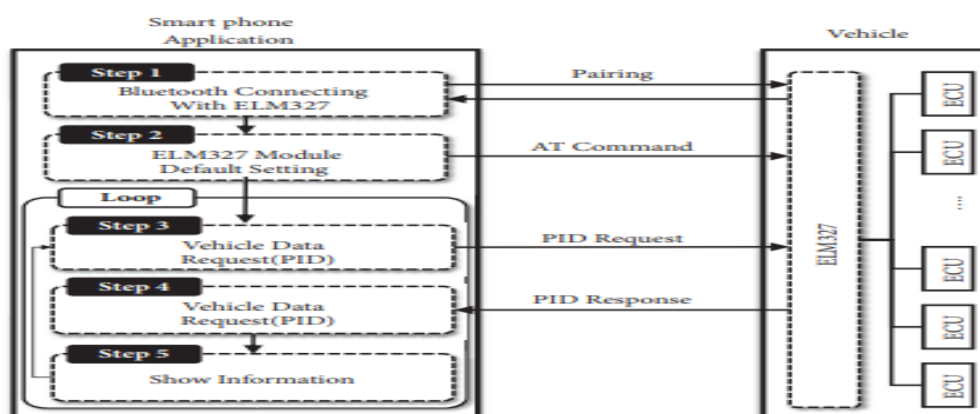


Figure 3: Process of checking vehicle status using smartphone and ELM327.

Source: Adopted from [10]

The proposed architecture for the implementation of this project is synonymous with that of a network layer structure as it comprises two layers. A physical layer and an application layer. The physical layer would handle all direct communications with the outside interfaces. I.e the ELM device with the OBD II port, alongside the accompanying device that would be responsible for receiving the diagnosis. Once communication has already been established via the physical layer. Data from the vehicle via the OBD II port is translated into RS232 codes using a circuit defined by the project specification. The resultant RS 232 codes would then be

converted into bits and then passed on to the transport layer. The transport layer would then broadcast the containing information through a wireless point to point medium which would eventually be received and read by the accompanying application. The materials required for the on-board wireless scanner are as follows

- ELM 327 wireless Dongle
- Accompanying GUI application to be run on PC or mobile phone.

ELM 327: The ELM 327 is a proprietary programmed microcontroller used in translating the onboard diagnostics (OBD) information. The ELM 327 device which was originally manufactured by ELM electronics interfaces with the IVN through the 16-pin OBD port found in most modern cars. ELM 327 supports both SAE J1850 PWM (41.6 kbit/s)

SAE J1850 VPW (10.4 kbit/s) protocols [8]

The ELM327 command protocol is one of the most popular PC-to-OBD interface standards capable of wired and wireless connections to both PC's and mobile phones and has now been incorporated by other diagnostic equipment vendors. By incorporating Bluetooth connectivity with ELM 327 devices it allows car owners and repair technicians easier usability and more functionality by eliminating the need for bulky wired OBD scanners

Accompanying Application: An accompanying application is required to interface with the ELM 327 device to receive the information output. Such application would be required to run on a Bluetooth enabled mobile phone or PC. Once the ELM 327 device has been connected to the OBD II port, the mobile phone/PC would be paired to the ELM 327 via Bluetooth. The application's GUI would be used to receive the RS232 codes transmitted by the ELM 327 device and subsequent diagnostics information can be read.

Product Specifications

1. The device passes all data sent from OBD-II port wirelessly to the scan tool on the accompanying device.
2. The system must be able to function flawlessly under normal working conditions. Normal working are inclusive of exposure to dust, dirt, and a frequent usage of up to 5 to 30 times per day.
3. The scan tool app on the accompanying device must be capable of informing the user on the state of the connection to the device. As a result, the user must be able to tell if the device is able to communicate with the OBD II port and also if the app is able to connect with the device.
4. The device should have a functionality range of up to 100 feet.
5. The system should be simple enough to require minimal user interaction and input to ensure optimal function.

6. The device is mandated to adhere to protocols J1850 VPW, J1850 PWM, ISO 9141-2, and Automotive CAN.

Additional Measures

Because CAN was designed only for closed network environments, it does not provide basic information protection features (e.g., confidentiality, authentication, and access control) [9]. There is the need to put into place added layers of security to avoid malicious usage of wireless ELM 327 devices. The absence of added security to the functionality of wireless OBD II readers has led to reported cases of malicious access to key vehicle functionality and could lead to possible cases of theft of physical belongings as well as user information. [3],[4],[9].

These added layers of security include but are not limited to deployment of firewall on the OBD II port, application authentication on wireless dongles, priority management on multiple connected devices.

CONCLUSION

On completion of the wireless OBD II reader, vehicle diagnostics, as well as live vehicle performance monitoring, can be seamlessly carried out by non-technical users, thereby cutting down the cost of vehicle maintenance while also making available to car owners valuable vehicular information.

References

- [1]A. S. Morris and R. Langari, MEASUREMENT AND INSTRUMENTATION : theory and application. S.L.: Elsevier Academic Press, 2020.
- [2]W. Ribbens, Understanding Automotive Electronics An Engineering Perspective. Saint Louis Elsevier Science, 2017.
- [3]H. Wen, Q. Chen, and Z. Lin, “Plug-N-Pwned: Comprehensive Vulnerability Analysis of OBD-II Dongles as A New Over-the-Air Attack Surface in Automotive IoT.” Accessed: Nov. 23, 2021. [Online]. Available: https://www.usenix.org/system/files/sec20summer_wen_prepub.pdf.
- [4]D. Klinedinst and C. King, “On Board Diagnostics: Risks and Vulnerabilities of the Connected Vehicle,” 2016. [Online]. Available: https://resources.sei.cmu.edu/asset_files/WhitePaper/2016_019_001_453877.pdf.
- [5]D. S. Eisinger and P. Wathern, “Policy evolution and clean air: The case of US motor vehicle inspection and maintenance,” Transportation Research Part D: Transport and Environment, vol. 13, no. 6, pp. 359–368, Aug. 2008, doi: 10.1016/j.trd.2008.05.001.
- [6]U. Shafi, A. Safi, A. R. Shahid, S. Ziauddin, and M. Q. Saleem, “Vehicle Remote Health Monitoring and Prognostic Maintenance System,” Journal of Advanced Transportation, vol. 2018, pp. 1–10, 2018, doi: 10.1155/2018/8061514.

[7]R. Malekian, N. R. Moloisane, L. Nair, B. T. Maharaj, and U. A. K. Chude-Okonkwo, "Design and Implementation of a Wireless OBD II Fleet Management System," IEEE Sensors Journal, vol. 17, no. 4, pp. 1154–1164, Feb. 2017, doi: 10.1109/jsen.2016.2631542.

[8]J. Iii and S. Bitar, "Creating A Wireless OBDII Scanner A Major Qualifying Project Report." Accessed: Nov. 24, 2021. [Online]. Available: <https://digital.wpi.edu/downloads/hd76s192s>.

[9]Y. Lee, S. Woo, J. Lee, Y. Song, H. Moon, and D. H. Lee, "Enhanced Android App-Repackaging Attack on In-Vehicle Network," Wireless Communications and Mobile Computing, vol. 2019, pp. 1–13, Feb. 2019, doi: 10.1155/2019/5650245.

[10]Y. Lee, S. Woo, J. Lee, Y. Song, H. Moon, and D. H. Lee, "Enhanced Android AppRepackaging Attack on InVehicle Network," Wireless Communications and Mobile Computing, vol. 2019, p. 5650245, 2019, doi: 10.1155/2019/5650245.

[11]All About Circuits, "Introduction to CAN (Controller Area Network)," Allaboutcircuits.com, Feb. 19, 2019. <https://www.allaboutcircuits.com/technical-articles/introduction-to-can-controller-area-network/>.