

Controller Area Network Protocol for Vehicle System

1st Sheikh Muhammad Adib Bin Sh Abu Bakar

Hochshule Hamm-Lippstadt
B.Eng. Electronic Engineering
Lippstadt, Germany

sheikh-muhammad-adib.bin-sh-abu-bakar@stud.hshl.de

2nd Zafirul Izzat Bin Mohamad Zaidi

Hochshule Hamm-Lippstadt
B.Eng. Electronic Engineering
Lippstadt, Germany

zafirul-izzat-bin.mohamad-zaidi@stud.hshl.de

3rd Muhammad Farid Izwan Bin Mohamad Shabri

Hochshule Hamm-Lippstadt
B.Eng. Electronic Engineering
Lippstadt, Germany

muhammad-farid-izwan.bin-mohamad-shabri@stud.hshl.de

Abstract—The modern world is dependent on the availability of vast communication networks that connect computers and other electronic devices all over the world. Clearly, there are several issues that must be addressed. These issues might be numerous and complex. In the framework of network communication challenges, we shall take a theoretical approach to several specific concerns. This paper will serve about communication network. This article will put a light on Controller Area Network (CAN) communication network and the way it solves others flaws. This post would also show the background of CAN and also some basic knowledge to assist the reader get a basic and a rough understanding. In this paper, the implementation of CAN model is also added to illustrate the usage of CAN protocol, but first, the reader must know the problem arise using other communication network to understand the need of CAN protocol in the model.

Last but not least, the reader may read our goals and what we have accomplished during the preparation of this paper on the outlook and conclusion.

Index Terms—Controller Area Network, Brake System

I. INTRODUCTION

Over the last several decades, the increasing power and cost-effectiveness of electronic equipment has had an impact on many fields of human endeavour. This is also true for industrial control systems, which are constantly improving. Although digital controllers eventually superseded analogue control, connections to the field were still made via analogue signals. The move to digital systems prompted the creation of new field and controller communication protocols. The most common name for these communication protocols is field bus protocols. Recently, digital control systems have begun to encompass networking at all levels of industrial control, as well as the connectivity of commercial and industrial equipment through the use of Ethernet standards.

An industrial communication network serves as the core of any automation system design since it provides a reliable way of data interchange, data controllability, and the ability

to connect many devices. The implementation of proprietary digital communication networks, such as Controller Area Network Protocol, specifically in automotive industries over the last decade, has resulted in better end-to-end digital signal accuracy and integrity. Industrial communication networks enable smart manufacturing scenarios by enabling real-time communication between machines and control centres. They ensure that supply chain anomalies are identified and prevented by monitoring processes and activities. Predictive maintenance is a great example of how the Industrial Internet of Things may increase efficiency and cost optimization by connecting and analysing data.

This paper serves as an introduction to industrial communication networks. Industrial networking concerns itself with the implementation of communications protocols between field equipment, digital controllers, various software suites and also to external systems. To assist the reader, the communication network that is widely used specially in industrial field will be explained generally so that the reader can have humongous imagination and deep understanding regarding this topic in the next section. The chosen communication network which is CAN will be thoroughly discussed and contrast with those other networks. Many aspects of the operation and philosophy of industrial networks has evolved over a significant period of time, and as such a history of the field is provided. The operation of modern control networks is examined, and some popular protocols are described. Any machine with sensors, for example, may collect critical information about its status and interactions with other equipment and operators.

II. TYPE OF COMMUNICATION

Communication is highly important in our lives, especially in any system. The meaning behind communication network is a pattern or shape that is used in an organization to efficiently transmit information. The communication network is the established system in which messages can travel in

one or more ways inside the company depending on the requirements. There are two kinds of communication. Wireless communication is the first, while wired communication is the second. For further information, multiple communication networks may be built based on their efficacy depending on the system's needs.

A. wireless network

Wireless means that there is no physical touch between two sites. Thus, wireless networks are computer networks that are not connected by any form of cable. The deployment of a wireless network allows businesses to avoid the costly procedure of installing wires in buildings or connecting multiple equipment locations. Radio waves serve as the foundation of wireless systems, acting as a mode of transport for data transmission and reception. The examples of wireless networks are shown below.

1) *Bluetooth*: Bluetooth was developed as an alternative to using cables and wires to build a network. It was designed to unite telecommunications protocols under a single worldwide standard. It is used for wireless communications and broadcasts over short distances. Bluetooth-enabled devices construct ad hoc networks (networks with a single function) with one another to exchange or broadcast data. While Bluetooth was designed to be a cable replacement, it is now extensively utilized because it generates a dynamic Personal Area Network (PAN) [2].

2) *Wi-Fi*: Wi-Fi is a wireless networking protocol. Wi-Fi is simply a very modern digital radio that operates at frequencies in the electromagnetic spectrum ranging from 2 gigahertz to 5 gigahertz, which is roughly the same as microwave ovens. Each standard is an approved modification throughout time. The standards operate at variable frequencies, provide varying bandwidth, and support varying numbers of channels [3].

B. Wired network

A wired network is one of the most popular types of wired arrangement. Ethernet cables are commonly used in wired networks to transport data between linked PCs. A single router may connect all PCs in a small wired network. Larger networks frequently have many routers or switches that communicate with one another. One of these devices is often linked to a cable modem, T1 line, or other sort of Internet connection, which offers Internet access to all network devices. Wired can also refer to peripheral devices. Below are the example of wired network.

1) *UART*: In UART communication, two UARTs communicate directly with each other. The transmitting UART turns parallel data from a controlling device, such as a CPU, into serial data and transmits it to the receiving UART, which converts the serial data back into parallel data for the receiving device. Only two wires are necessary to transfer data between two UARTs. Data is sent from the Tx pin of the transmitting UART to the Rx pin of the receiving UART. The diagram below depicts the UART connection [4].

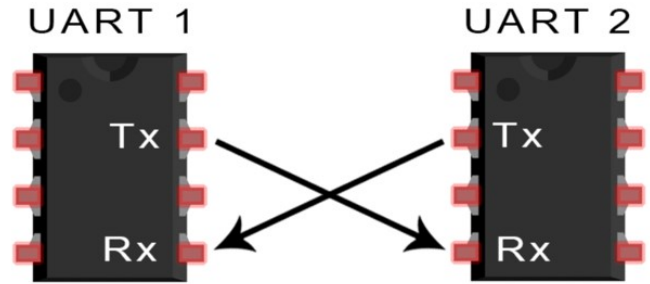


Fig. 1. Connection of UART

2) *I2C*: I2C combines the greatest qualities of SPI and UARTs. I2C allows you to connect several slaves to a single master (similar to SPI) and control single or multiple slaves. This is especially useful when many microcontrollers are recording data to a single memory card or displaying text on a single LCD. I2C communication, like UART communication, uses just two wires to send and receive data between devices: Because I2C is synchronous, a clock signal shared by the master and slave syncs the bit output to the bit sampling. The clock signal is always in the hands of the master. SDA (Serial Data) - The data transmission and reception line between the master and slave. SCL (Serial Clock Line) — The line that carries the clock signal [4].

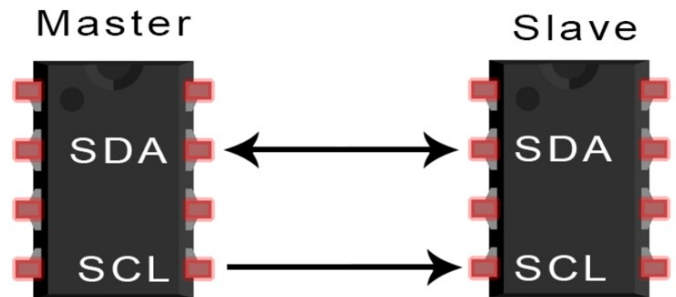


Fig. 2. Connection of I2C

III. PROBLEM

Many electronic components make up today's automobile, and they all communicate with one another. For example, a system that receives human input, such as a pedal brake, will connect with the brake lamp system. At least two wires may be required for communication. In addition, other systems, such as the safety belt system and the door lock system, also need the user's input. Means that more and more cable will be added to this communication environment. This might get a lot messier and more difficult to maintain.

The core principle of the approach is to use a bus system. The bus system should meet our primary criteria of using less cable across the communication system. I2C is a common bus system protocol that requires only two wires for the whole communication system. After experimenting with I2C, we

discovered that it is not a viable solution since communication becomes unreliable when the cable length exceeds one meter. If we choose a very low frequency, which indicates a low data rate transmission, the communication becomes stable over a long distance.

When a driver has to apply an emergency brake, communicating at a low frequency can cause the brake lamp system to respond slowly, causing the car behind to be unable to respond quickly. As a result, you will have a disastrous outcome. That is, we need to enhance our communication protocol requirements by adding additional parameters, such as support for high frequency or high data rate transfer across long distances wire.

To meet all of these requirements, we will need to conduct more study on a protocol that can fix the problem. As a result, the optimum solution for this problem or demand is the Controller Area Network. In the next part, we will go through the protocol in detail.

IV. CAN COMMUNICATION PROTOCOL

Controller Area Network (CAN) is a highly integrated system that connects intelligent devices for real-time control applications by utilizing serial bus and communications protocol. Control Area Network use a vehicle bus standard that allows communication such as, transferring messages, data or informations between micro-controllers and other devices. CAN is able to operate at data speeds of up to 1 megabits per second (1Mbps) over a possible line length of up to several kilometres, and has outstanding error detection and confinement capabilities. Control Area Network which is also well-known as CAN Communication was introduced by Robert Bosch GmbH in 1983. CAN is purposely created to make automobile industry system more reliable and efficient but it is also widely used in industrial automation and other fields [8].

A. The CAN Standard

The International Organization for Standardization (ISO) designed the CAN serial communications bus for the automotive industry to replace the complex wiring harness with a two-wire bus. According to the standard, high immunity to electrical interference is necessary, as well as the capacity to self-diagnose and correct data errors. CAN is widely utilized in a variety of industries, including building automation, medical, and manufacturing, because to these qualities [8].

The CAN communications protocol, ISO-11898: 2003, specifies how data is transmitted between devices on a network and follows the Open Systems Interconnection (OSI) paradigm, which is divided into layers. The physical layer of the model defines the actual communication between devices connected by the physical medium. as shown in “Fig. 3”

B. Standard CAN

The CAN communication protocol is a carrier-sense, multiple-access protocol (CSMA/CD+AMP) that includes collision detection and message priority arbitration. Each node

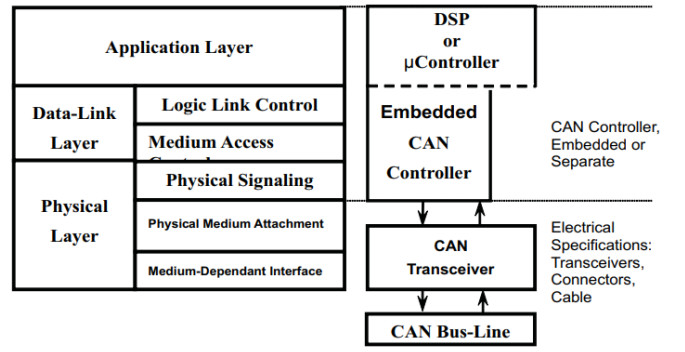


Fig. 3. The Layered ISO 11898 Standard Architecture [8]

on a bus must wait for a certain duration of idleness before attempting to send a message, according to CSMA. CD+AMP denotes that collisions are resolved using bit-wise arbitration, with each message’s priority preprogrammed in the identification field. Bus access is always granted to the identification with the highest priority. In that example, because the last logic high in the identification has the highest priority, it continues to transmit. An arbitrating node knows if it placed the logic-high bit on the bus because every node on the bus participates in writing every bit “as it is written.” [8].

C. A CAN Message

In this section, CAN Message is explained more details in term of its arbitration. CAN also has several types of messages such as Data Frame, Remote Frame etc [8].

1) *Arbitration:* The opposing logic state between the bus and the driver input and receiver output is a basic CAN property, as shown in “Fig. 4”. A logic high is normally linked with a one, and a logic low with a zero, however this is not the case on a CAN bus. This is why the driver input and receiver output pins of TI CAN transceivers are internally passively pushed high, such that the device defaults to a recessive bus state on all input and output pins in the absence of any input [8].

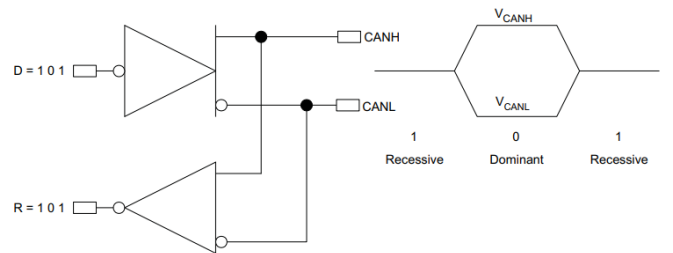


Fig. 4. The Inverted Logic of a CAN Bus [8]

Bus access is based on events and occurs at random. If two nodes try to occupy the bus at the same time, non-destructive bit-wise arbitration is used to provide access. Non-destructive means that the node that wins arbitration keeps sending the message without it being deleted or corrupted by another node [8].

A feature of CAN that makes it particularly appealing for application in a real-time control environment is the allocation of priority to messages in the identification. The higher the priority, the smaller the binary message identifying number. Because it keeps the bus dominant the longest, a message with an identification made entirely of zeros is the highest priority message on a network. As a result, if two nodes start transmitting at the same time, the node that sends a zero (dominant) as the last identifying bit while the others send a one (recessive) holds control of the CAN bus and continues to complete its message. On a CAN bus, a dominant bit always overwrites a recessive bit [8].

2) *Message Types*: There are four different message types (or frames) that can be transmitted on a CAN system as below [8]:

- The Data Frame
- The Remote Frame
- The Error Frame
- The Overload Frame

3) *A Valid Frame*: When the last bit of a message's terminating EOF field is received in the error-free recessive state, the message is considered error-free. The transmitter repeats a broadcast if a dominating bit in the EOF field is present [8].

4) *Error Checking and Fault Confinement*: The abundance of error-checking mechanisms in CAN contribute to its robustness. The CAN protocol has five error-checking methods. Three of the method at the message level and two of them at the bit level. If a message fails to pass any of these error detection mechanisms, it is not accepted, and the receiving node generates an error frame. This compels the transmitting node to relay the message several times until it is appropriately received. If a malfunctioning node hangs up a bus by repeatedly repeating an error, its controller disables its transmit capacity if an error limit is reached. Error checking is done in two level which are message level and bit level as explained below [8].

- **Message Level** : A form check is performed at the message level. This check examines the message for fields that must always be recessive bits. An error is generated if a dominating bit is discovered. The SOF, EOF, ACK delimiter, and CRC delimiter bits are all examined.
- **Bit Level** : Each bit transferred is monitored at the bit level by the message's transmitter. An error is caused if a data bit (not an arbitration bit) is written onto the bus and its opposite is read. The message identity field, which is used for arbitration, and the acknowledge slot, which requires a recessive bit to be replaced by a dominant bit, are the only exceptions.

D. The CAN Bus

The CAN standard establishes a communication network that connects all bus nodes and allows them to communicate with one another. A central control node may or may not exist, and nodes may be added at any moment, even while

the network is operational (hot-plugging) . As shown in “Fig. 5”, CAN Bus is illustrated. It consists of 3 main components which are CPU, CAN Controller and Transceiver. Electronic Control Unit (ECU) also known as nodes. All nodes are connected to two wire busg that have signal CAN HIGH and CAN LOW [8].

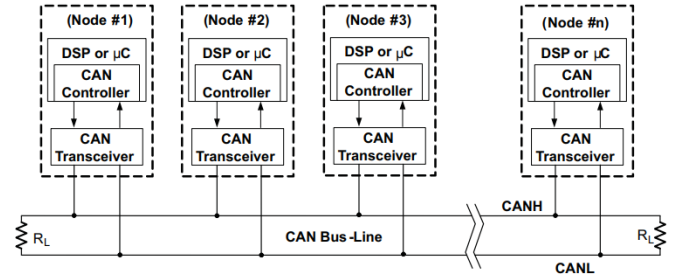


Fig. 5. Details of a CAN Bus [8]

1) *CAN Transceiver Features*: A few transceivers in CAN protocol features are listed below [8].

- 3.3V Supply Voltage
- ESD Protection
- Common-Mode Voltage Operating Range
- Common-Mode Noise Rejection
- Controlled Driver Output Transition Times
- Low-Current Bus Monitor, Standby and Sleep Modes
- Low-Current Bus Monitor, Standby and Sleep Modes
- Thermal Shutdown Protection
- Bus Input Impedance
- Glitch-Free Power Up and Power Down
- Unpowered Node Protection
- Reference Voltage
- V-Split
- Loopback
- Autobaud Loopback

V. APPLICATION OF CAN PROTOCOL

In this section, the project in developing a prototype subsystem in a car that use CAN protocol as its communication standard will be discussed. For that, we use the problem as state in problem section as our main reference for our prototype.

After we analyse the problem, we refine our requirement in requirement diagram as shown in “Fig. 6”. As mentioned before, the protocol should work for long distance at high frequency.

As explained in the problem section, Controller area network is chosen because it supports high data rate transfer for long distance and minimize number of wire used. Next, how the model of the solution will be explained based on the requirement.

A. Application Model

Our aim at this stage, which is modelling our solution, is to have a concrete system architecture including communication

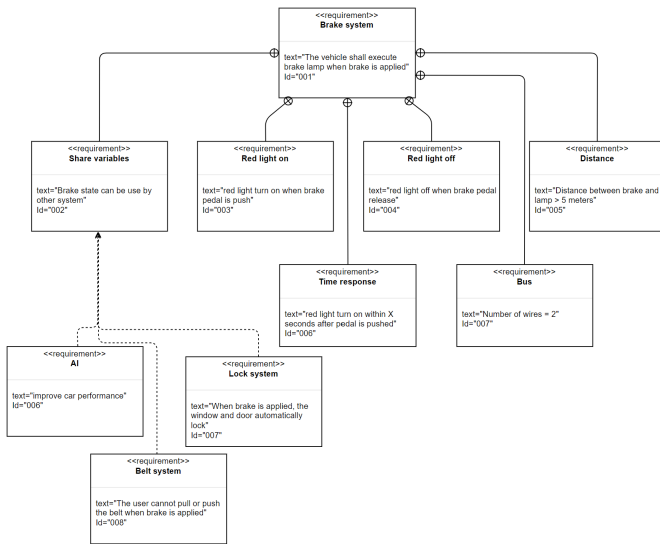


Fig. 6. Requirement Diagram

system where we will implement the communication protocol, so that it will be clear for us to do the implementation. The steps in this stage will be discussed in this section.

After analysing the scenario, requirements and protocol that we want to use, we create a use case diagram to have a clear boundary of the system as shown in “Fig. 7”.

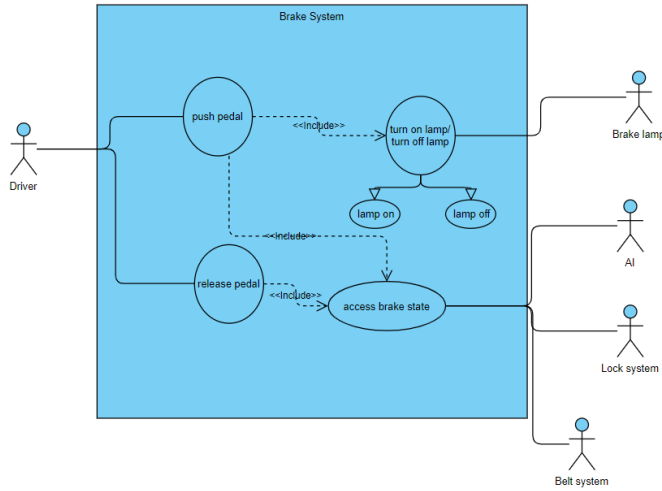


Fig. 7. Use Case Diagram

Next, we extend each use case using activity diagrams to understand their behaviour as pictured in “Fig. 8”, and “Fig. 9”. later we use all this diagram to make a sequence diagram to illustrate how object in the subsystem interact with each other as shown in “Fig 10”.

Eventually, by analysing all the pervious diagram, we are able to refine our system architecture as shown in “Fig. 11”. This indicates that we are ready for implementation stage, which will be the focus of the following subsection.

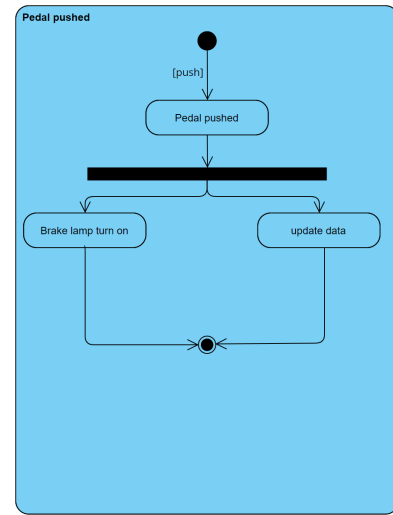


Fig. 8. Activity diagram when the pedal pushed

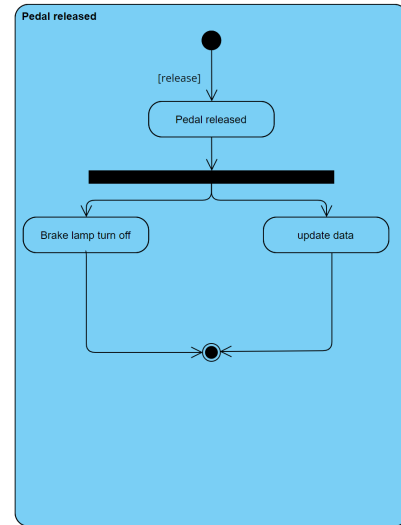


Fig. 9. Activity diagram when the pedal is released

B. Implementation

For the implementation we divide it into two parts which is hardware and software part. We will first explain the hardware part followed by software part.

1) *Hardware*: For the hardware setup we will use Arduino UNO with CAN module as shown in “Fig. 12” to represent each subsystem as drawn in the system architecture that we present before in “Fig. 11” which are pedal system, database system and brake lamp system as shown in “Fig. 13”, “Fig. 14”, “Fig. 15”. The CAN module that we are used is MCP2515. For the connection with between the Arduino and the CAN module, we use SPI communication protocol. Considering that the protocol should work for long distance, we also use ca. 1.5 meter for the connection between subsystem. After we have done the setup as illustrate in Figure

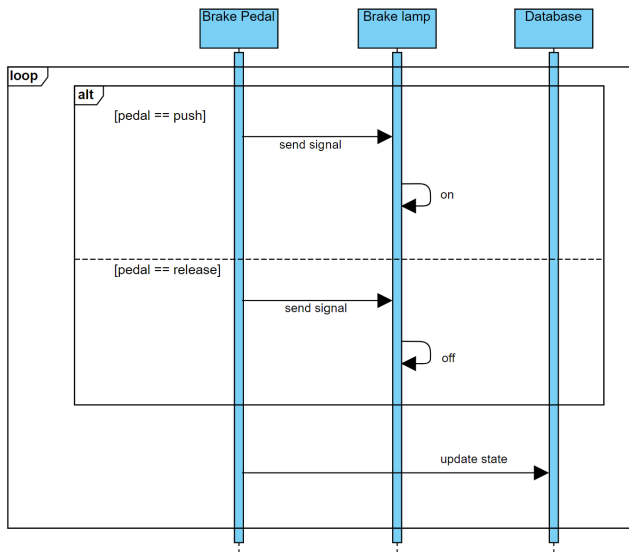


Fig. 10. Sequence Diagram

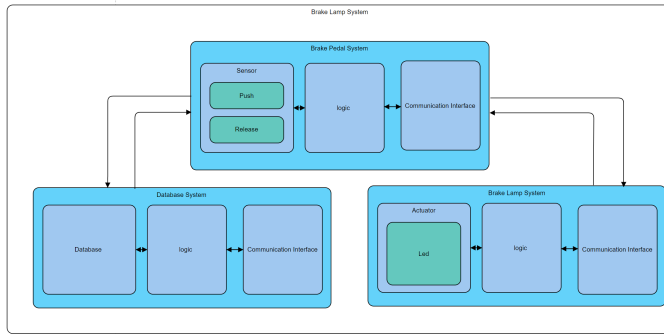


Fig. 11. System Architecture

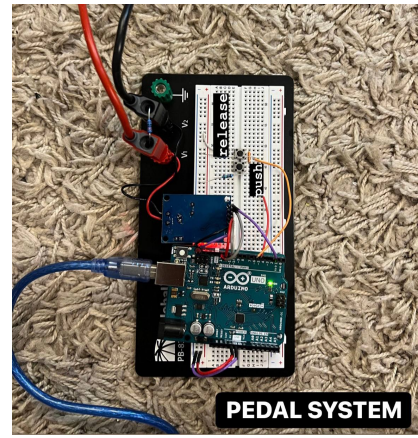


Fig. 13. Pedal subsystem

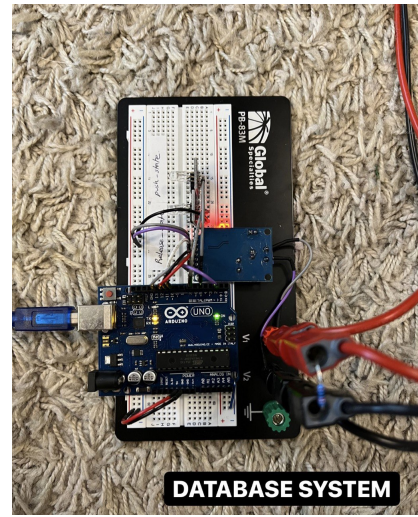


Fig. 14. Database subsystem

“Fig. 16” we proceed to the next step which is the software implementation.



Fig. 12. MCP2521

2) *Software*: Since we use Arduino as our subsystem platform, we use Arduino IDE to implement our software. To make the communication works as we had planned, we use CAN library provided by Arduino. In proving that It could work at high frequency, we set the data rate transfer to 1Mbps. The code for each subsystem can be found in <https://github.com/Adib6637/Industrial-Communication>.

VI. STATE OF THE ART

The CAN bus has played an essential role in the industry since its inception in 1987. Many companies, notably those in the automotive industry, are concentrating on CAN development. BMW, for example, first employed CAN bus technology in 1995. They used a star topology network with five electronic control units in their 7 Series automobiles. They then expanded the concept by integrating a second network for body electronics. This allowed two separate CAN-BUS networks to be linked via gateway connections. CAN systems gain trust in the automotive sector through protocol standardization and security testing to ensure system compatibility [5]. The CAN bus has established a name for itself not just in the automobile industry, but also in the medical arena. Philips Medical Systems, for example, was one of the first medical equipment manufacturers to employ the CAN bus for internal networking of their X-ray machines [6]. It's probable that vending machines will be created in the near future. As observed, a Chinese start-up company developed a prototype and deployed a CAN network. The prototype has the ability to

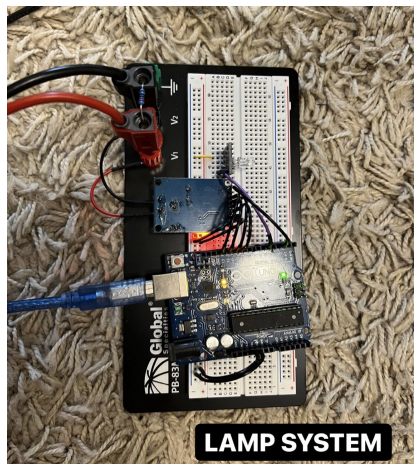


Fig. 15. Brake lamp subsystem

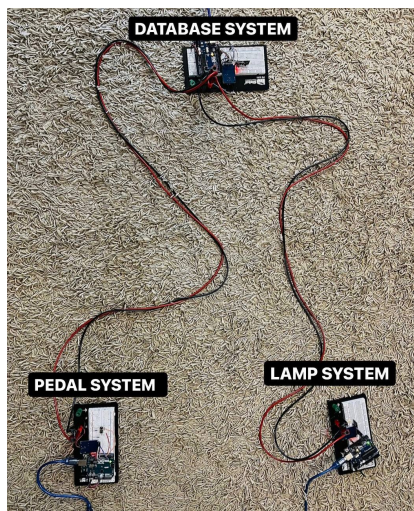


Fig. 16. Hardware setup

make coffee. In other cases, it might be supplied with vending machines that sell vegetables and other meal supplies. Shuttle services using self-driving vehicles and autonomous freight will also be possible. A CAN-compliant vehicle control unit is installed in the vehicle. It controls the lights, horn, windscreen wiper, handles, accelerator, and other functions. The electric steering unit also supports CAN connectivity [7].

VII. DISCUSSION

Our project is eventually completed successfully. As previously stated, the data transfer rate that is chosen is 1Mbps, and the length of the cable used is around 5 meters. Those requirements have been met after several iterations of the development and testing processes.

VIII. CONCLUSION

Based on our project, we believe that the Controller Area Network protocol is well suited for long-distance communication that necessitates the use of a bus system and high data transmission rates. This type of bus system is extremely useful,

especially for systems that require long-distance communication without jeopardizing the environment's safety, such as cars and automation robots.

ACKNOWLEDGMENT

We would like to express our very great appreciation to Prof. Dr.-Ing. João Paulo Javidi da Costa for his valuable and constructive suggestions during the planning and development of this project. His willingness to give his time so generously has been very much appreciated.

REFERENCES

- [1] B. Galloway and G. P. Hancke, "Introduction to Industrial Control Networks," in *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 860-880, Second Quarter 2013, doi: 10.1109/SURV.2012.071812.00124.
- [2] J. Wu, "ECE Senior Capstone Project," *How Bluetooth Works*, 2017.
- [3] M. Sparkes, "What does Wi-Fi Stand For?," *New Scientist*, 2011. [Online]. Available: <https://www.newscientist.com/question/what-does-wi-fi-stand-for/>. [Accessed: 24-May-2022].
- [4] S. Campbell, "Basics of Uart Communication," *Circuit Basics*, 14-Nov-2021. [Online]. Available: <https://www.circuitbasics.com/basics-uart-communication/>. [Accessed: 24-May-2022].
- [5] F. Ahmad, M. Sultan (Eds.) *Technology in Agriculture*. London, United Kingdom, IntechOpen, 2021 [Online]. Available from: <https://www.intechopen.com/books/10454> doi: 10.5772/intechopen.92469
- [6] L. B. Fredriksson, "Medical: Kvaser: Advanced can solutions," Kvaser, 15-Jun-2020. [Online]. Available: <https://www.kvaser.com/about-us/applications/medical/>. [Accessed: 26-May-2022].
- [7] C. S. amp; Webinar, "Vending vehicle with robot making coffee," *Applications*, 2022. [Online]. Available: <https://can-newsletter.org/engineering/applications/180813-vending-vehicle-with-robot-making-coffee-pix>. [Accessed: 26-May-2022].
- [8] S. Corrigan, *Academia.edu*, 2002. [Online]. Available: <https://www.academia.edu/23928983/IntroductiontotheControllerAreaNetworkCAN>. [Accessed: 31- May- 2022].