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*—**
***Index Terms*—real time system, resource, protocol**

1

I. Introduction

The rising power and cost effectiveness of electronic devices has affected all aspects of human endeavour during the last several decades. This also not exceptional to industrial control systems which are always improvise progressively. Control of industrial and process facilities was first done mechanically in the early 20th century, either manually with analogue devices or through the use of hydraulic controllers. Mechanical control systems were superseded by electronic control loops that used transducers, relays, and hard-wired control circuits as discrete electronics became more prevalent now a days. These systems were enormous and took up a lot of area, frequently necessitating many kilometres of wire, starting from one point to another point, as example the connection from the field itself to the control circuits. The functionality of many analogue control loops may be reproduced by a single digital controller thanks to the introduction of integrated circuits and microprocessors.

We know that at the beginning phase, a digital computer was for the first time being applied as a digital controller. As time goes by, Digital controllers gradually replaced analogue control, however connection with the field was still done using analogue signals. The transition to digital systems necessitated the development of new communication protocols for the field as well as between controllers. Fieldbus protocols are the most frequent name for these communication protocols. Recently, digital control systems began to include networking at all levels of industrial control, as well as the interconnection of commercial and industrial equipment utilizing Ethernet standards. This has resulted in a networking environment that, on the surface, looks to be comparable to traditional networks but has fundamentally different requirements.

An industrial communication network is the foundation of any automation system design since it provides a strong method of data interchange, data controllability, and the flexibility to link numerous devices. Over the last decade, the deployment of proprietary digital communication networks in industries has resulted in improved end-to-end digital signal correctness and integrity. With the help of real-time communication among machines and control centers, industrial communication networks power smart manufacturing scenarios. They guarantee that supply chain abnormalities may be evaluated and avoided by monitoring procedures and operations. Predictive maintenance is an excellent illustration of how the Industrial Internet of Things may improve efficiency and cost optimization through connection and data analytics. For example, any machine with sensors may collect vital information about its state and interactions with other equipment and operators. The network transmits this data to processing centers, where it is compared to historical and statistical data to find trends, recurrences, and exceptions. They can also build links between certain inputs and any breakdowns or inefficiencies. Intelligent gateways provide smooth interaction between several communication protocols that are structured in separate subnetworks.

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 explain generally so that the reader can have humongous imagination and deep understanding regarding this topic. 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

Identify applicable funding agency here. If none, delete this.

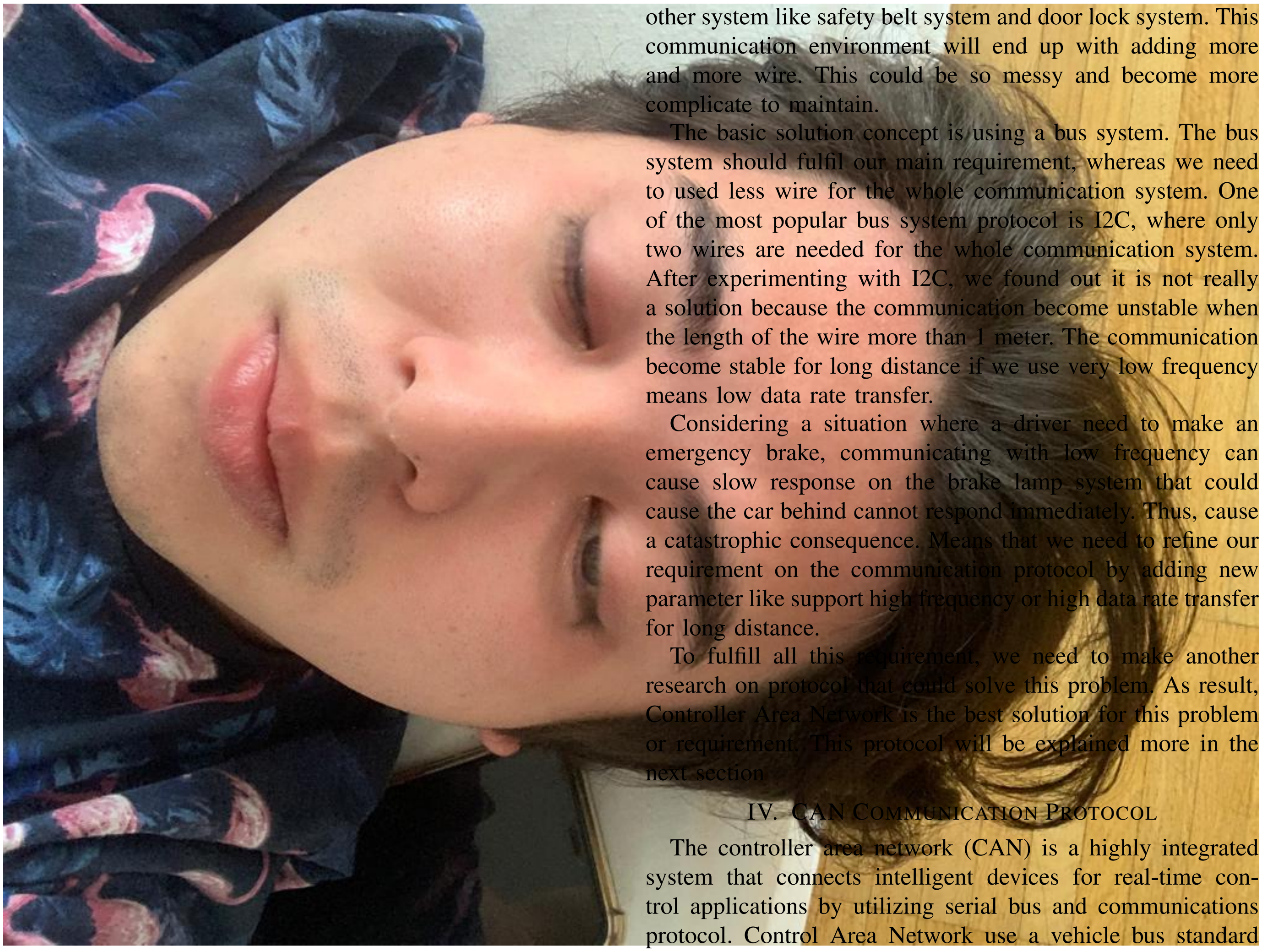

Fig. 1. japi masak lemak cili api. [?]

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

II. Type of communication

A. wireless network

1) *bluetooth:*

2) *wifi:*

B. wired network

1) *I2C:*

2) *UART:*

III. Problem

Today's car consist of many electronic components, and they are communicating with each other. For instance, the system that get input from user like pedal brake will communicate with the brake lamp system. The communication may require at least two wire. The input from the user also may be used by other system like safety belt system and door lock system. This communication environment will end up with adding more and more wire. This could be so messy and become more complicate to maintain.

The basic solution concept is using a bus system. The bus system should fulfil our main requirement, whereas we need to used less wire for the whole communication system. One of the most popular bus system protocol is I2C, where only two wires are needed for the whole communication system. After experimenting with I2C, we found out it is not really a solution because the communication become unstable when the length of the wire more than 1 meter. The communication become stable for long distance if we use very low frequency means low data rate transfer.

Considering a situation where a driver need to make an emergency brake, communicating with low frequency can cause slow response on the brake lamp system that could cause the car behind cannot respond immediately. Thus, cause a catastrophic consequence. Means that we need to refine our requirement on the communication protocol by adding new parameter like support high frequency or high data rate transfer for long distance.

To fulfill all this requirement, we need to make another research on protocol that could solve this problem. As result, Controller Area Network is the best solution for this problem or requirement. This protocol will be explained more in the next section

IV. CAN Communication Protocol

The 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.

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.

The CAN communications protocol, ISO-11898: 2003, specifies how data is transmitted between devices on a network and follows the Open Systems Interconnection (OSI)

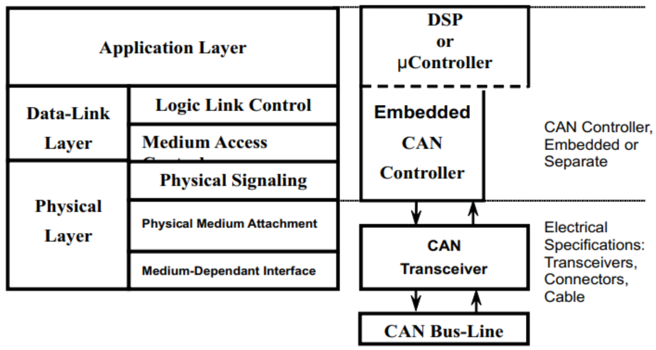

Fig. 2. The Layered ISO 11898 Standard Architecture [?]

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 Figure 2

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 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. 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."

C. A CAN Message

1) Arbitration: The opposing logic state between the bus and the driver input and receiver output is a basic CAN property, as shown in Figure 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.

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

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.

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

- 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.

4) Error Checking and Fault Confinement: The abundance of error-checking mechanisms in CAN contributes to its robustness. The CAN protocol has five error-checking methods: three at the message level and two 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.

1) 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.

2) 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)

1) CAN Transceiver Features: 1) 3.3V Supply Voltage
2) ESD Protection
3)Common- Mode Voltage Operating Range
4) Common-Mode Noise Rejection
5) Controlled Driver Output Transistion Times
6) Low-Current Bus Monitor, Standby and Sleep Modes
7) Low-Current Bus Monitor, Standby and Sleep Modes
8) Thermal Shutdown Protection
9) Bus Input Impedance
10) Glitch-Free Power Up and Power Down

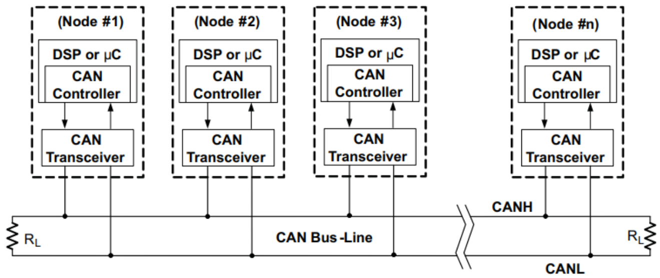

Fig. 3. Details of a CAN Bus [**?**]

11) Unpowered Node Protection
12) Reference Voltage
13) V-Split
14) Loopback
15) Autobaud Loopback

V. Application of CAN Protocol

Now we will go through our project in developing a prototype subsystem in a car that use CAN protocol as its communication standard. 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 Figure 4. As we mention before, the protocol should work for long distance at high frequency.

Fig. 4. Requirement Diagram

As we explain in the problem section, we choose Controller area network because it supports high data rate transfer as stated in the pervious section. Next, we will explain how we model our solution base 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 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 Figure 5.

Fig. 5. Use Case Diagram

Next, we extend each use case using activity diagrams to understand their behaviour as pictured in Figure **??**,**??**,**??**, and **??**. 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 Figure 6.

Eventually, by analysing all the pervious diagram, we are able to refine our system architecture as shown in Figure 7. This indicates that we are ready for implementation stage, which will be the focus of the following subsection.

Fig. 6. Sequence Diagram

Fig. 7. System Architecture

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 Figure **??** to represent each subsystem as drawn in the system architecture that we present before in Figure 7 which are pedal system, database system and brake lamp system as shown in figure **??**. 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 **??** we proceed to the next step which is the software implementation.

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://www.instagram.com/$_{s}heikh_{a}dib/?hl = en$.

VI. State Of The Art

VII. Discussion

VIII. Conclusion

Acknowledgment

We, Sheikh Muhammad Adib and .. and ... thanks..

References

[1] G. C. Buttazzo, "Hard real-time computing systems: Predictable scheduling algorithms and applications". New York: Springer, 2011.