

CAN BUS PROTOCOL

Ammar Imran khan, Kashif Raza
Hochschule Hamm-Lippstadt
Bachelor of Engineering - Electronic Engineering
Lippstadt, Germany

Abstract—The Controller Area Network bus protocol has become a strong and widely used communication standard in the domains of automotive and industry. This paper presents an overall survey on architecture, standards, and implementation of the CAN bus system. First proposed by BOSCH, the CAN protocol supports efficient data exchange in between electronic control units in real time with reliability and low complexity in terms of wiring. It also includes ISO 11898 standards that govern the operations of CAN, the network topology, node structure, mechanisms for error detection, and fault confinement. In addition, the details of CAN framing will be considered, moving from data frames to remote frames and error frames with the different bit-stuffing techniques used for synchronization. The paper will treat in detail the arbitration process, which is important for the management of message priority and access to a bus. The paper also deals with strategies for protecting CAN data, which is a very important requirement of security in modern applications. Though the CAN bus has its advantages, some major limitations are that the maximum data rate reaches 1 Mbps and the bus is subject to physical faults. These aspects are critically analyzed in a manner that gives a balanced perspective of the protocol's capabilities and limitations. The paper concludes with the following statements: a CAN bus remains relevant in modern real-time systems, and its further evolution is necessary for its application in many areas.

Index Terms—Controller Area Network (CAN), CAN bus, automotive communication, industrial communication, error detection, fault confinement, data framing, message arbitration, CAN data protection, network topology, bit-stuffing techniques, real-time systems, high-speed applications, low-speed applications, differential signaling, electromagnetic interference (EMI).

I. INTRODUCTION

A Controller Area Network (CAN) is like the nervous system of a lot of today's vehicles and machinery. Here, the sensors function like receptors, while ECUs function like sensory neurons. In this respect, CAN bus systems acts as channels of communication that hold the key to the interchange or interpretation of data between these electronic control units. It is because of the inefficiency of the inter-ECU communication process that has motivated the development of CAN bus technology with its selectivity support scheme and strong error-detection capability. The distributed system provides efficient transmission, reduces wiring complexity, and enhances the integrity of the whole system. Further, ongoing research is using the CAN bus for precision agriculture, vehicle autonomy, and wireless applications. [1] BOSCH developed the CAN

bus with a multi master message broadcast system in mind, having an operating signaling rate of 1 megabit per second. Unlike conventional networks, which require end-to-end connections, CAN employs the broadcast principle in which short messages are transmitted throughout to achieve consistency in the data. Proper understanding of the basics of CAN stays instrumental in studying its implementation, typical waveform, and transceiver features. [2]

II. STANDARDS

The CAN bus system operates within a defined framework of standards established by the International Organization for Standardization (ISO) under ISO 11898. This will ensure that devices from different vendors seamlessly communicate, transitioning to a seamless plug-and-play environment. At the core of the standards is ISO 11898-1, defining all operations relating to the data link layer or the operation of Layer 2. It defines exactly how data packets are structured, which, among other concepts, includes message arbitration uses for prioritization, error detection schemes to enable recognizing corrupted data, and acknowledgment procedures to confirm the correct reception of data. This comprehensive treatment make sure that communication is reliable and robust over a CAN bus. [3] It can be further seen that ISO 11898-2 and ISO 11898-3 only specify the requirements for the physical layer, generally describing how data is transferred on the physical bus. ISO 11898-2 supports high-speed applications—as in today's vehicles—by specifying a maximum bit rate of 1 Mbps and allowing a bus length of up to 40 meters. It requires that differential signaling is balanced to enhance noise immunity in these high-speed environments; two wires would carry equal but opposite voltages, effectively canceling out external electrical noise. Noise immunity can be improved by using twisted-pair cabling that considers actually twisting the wires together. [4] In contrast, ISO 11898-3 is targeted toward applications that require even slower data rates and perhaps other configurations of the network. This standard allows bit rates as low as 125 kbit/s and hence is targeted toward less time-critical applications. This enables further flexibility in network topology—linear structures of the bus, star configurations, or a combination. This flexibility gives ISO 11898-3 countless of other applications that would not require very high speeds, but are otherwise flexible and cheap. [5]

III. CAN NETWORK

A Controller Area Network (CAN) consists of multiple CAN nodes interconnected via a physical transmission medium called the CAN bus. The CAN network will normally have the line topology, that is, there is a linear bus to which multi-numerous ECUs are attached with a CAN interface. Another topology that can be used is often the most common passive star topology, which offers several advantages depending on the design requirements of the network. The maximum data rate supported by the CAN network is 1 Mbit/s, and it only permits a maximum length of about 40 meters to ensure optimum performance. There are bus termination resistors at the two ends of the CAN network for damping transient phenomena like signal reflections. According to the ISO 11898 standard, the maximum number of nodes in a CAN network can be up to 32. Beyond its basic configuration, the CAN protocol incorporates error detection, fault confinement, and automatic re-transmission to enable robust communication even in a noisy environment. These characteristics make the CAN Network very suitable for real-time applications in automotive systems, industrial automation, medical devices, and other embedded systems where reliability and speed are important. This flexibility and scalability of the CAN network in itself allow for easy extension and integration of nodes in case of further system needs. [7]

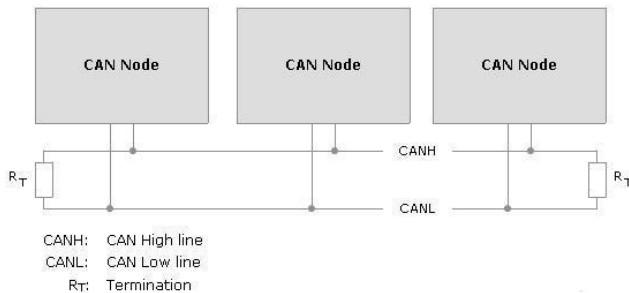


Fig. 1. CAN Network [6]

CAN node is an individual device or micro-controller within a Controller Area Network (CAN) that is capable of transmitting and receiving data over the CAN bus. Every node on a CAN includes a CAN controller and a CAN transceiver. The CAN controller handles the communication protocol; it assembles and disassembles messages and detects and signals errors. It converts the digital signals from a CAN controller into electrical signals to be transmitted on a CAN bus and performs the reverse process. The CAN nodes system has an important role in the running of a CAN network; this allows for real-time communication between the diverse electronic units that make up a network and are attached to the ECUs. Those nodes can be a wide variety of devices, including sensors, actuators, and micro-controllers controlling various aspects of a specific system. For example, in automotive applications, CAN nodes could control engine management, transmissions, airbags, anti-lock braking systems, and infotainment.[7]

An electronic control unit (ECU) intending to engage in CAN communication needs a CAN interface; this interface consists of a CAN controller and a CAN transceiver. **CAN controller** is responsible for carrying out communication functions according to the CAN protocol. This has a high load reduction on the host micro-controller. It encapsulates frames, filters messages, and provides error detection to guarantee reliable transmission and reception of data. **CAN transceiver** converts the digital signal supplied by the CAN controller into differential signals, which is suitable for transmission over the CAN bus. This would require signal level shifting, filtering, and impedance matching so the integrity of the signal remains and reliability is guaranteed over the network. This capability is essential for ensuring strong communication, particularly in environments with potential electrical noise or interference. Modern transceivers also host protection mechanisms against electrical spikes and noise to guarantee stable operation in very different industrial and automotive applications.[7]

A CAN Network uses differential signal transmission for communication, a way that turns down electromagnetic trouble created by devices such as motors, ignition systems, and switches. This method utilizes two dedicated lines: **CAN high (CANH)** and **CAN low (CANL)**, forming the CAN bus. Normally, these lines are twisted together to further reduce electromagnetic interference, but this will ensure that magnetic field emissions are significantly reduced. This is a standard practice for any CAN network. Such twisted pair configurations ensure strong and reliable data communication between two nodes, with many applications in modern technologies in industry and in-vehicle networks. A differential voltage approach avoids noise-related errors, thus providing high signal integrity for real-time communication and system reliability. In CAN, there are also some differences between high-speed and low-speed systems, mainly because of the different fields of application and current characteristics.[7]

High-Speed systems operate with a maximum data rate of 500 kbit/s, utilizing a differential two-wire configuration: CAN high and CAN low. If one gets damaged, then it will stop working correctly. High-speed systems require a high degree of balance in transmitting between the two lines, CANH and CANL, to ensure reliable communication among them. Therefore, in cases of damage to one of the wires or its disconnection, receiving differential signals will not be possible, resulting in communication errors that may bring the system down. It allows for a maximum of 10 nodes per network, making it ideal for applications that require real-time communications, such as automotive engine control and transmission control.[8]

On the other hand, **Low-speed systems** use the same wires, CANH and CANL, with a maximum data rate of 100 kbit/s. It allows up to 24 nodes per network and is therefore less expensive for less data-intensive tasks, such as vehicle body modules and peripheral systems. Besides this, the low-speed systems can also be configured using a single wire in case one of the wires gets damaged or even disconnected.[8]

A CAN network uses the multi-master architecture in combination with line topology; every node in a CAN network is empowered to send a message onto the bus. In CAN, transmission of a message does not take place by any pre-determined time sequence of transmission but is event-driven. Any node may start communicating at will. CAN utilizes receiver-selective addressing, which gives great flexibility in configuration and reduces dependencies between nodes. All the messages are broadcast in CAN and are therefore available to all nodes. Each message has a unique ID and is recognized by the nodes with specific filtering. This adds some overhead but makes it easier to add nodes without modifying the configuration.[7]

IV. CAN FRAMING

CAN (Controller Area Network) framing refers to the structure and organization of data packets, known as frames, that are used for communication within a CAN network. Here's an overview of CAN framing:

Data frame: The data frame in CAN (Controller Area Network) serves as the fundamental unit for transmitting user data between nodes within the network. According to standard ISO 11898-1, the maximum size of the payload transferred can be up to eight bytes. At the center of the data frame lies the data field, which holds information to be transmitted. This payload is then surrounded by all other fields—essential parts mandated by the CAN communication protocol. These include the ID, which defines message priority and destination; DLC, which denotes the number of bytes available in the data field and can range from 0 to 8 bytes; and lastly, CRC, used in error detection to ensure the integrity of data. Moreover, the frame contains an acknowledgment slot that indicates successful reception of the frame by the destination node and an end-of-frame marker that indicates that transmission has been completed.[7]

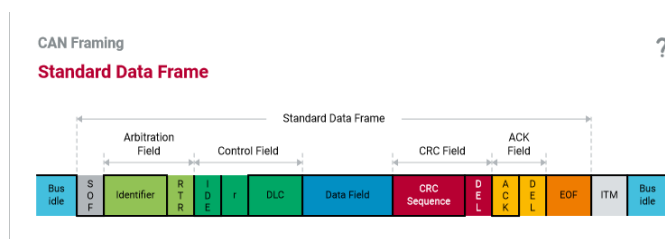


Fig. 2. Standard Data Frame [6]

Remote frame: In addition to the data frame, the remote frame is another important frame type in the CAN protocol. If data frames are mainly initiated by a generator ECU to transfer the user data on the network, remote frames are employed to request specific data frames from other CAN nodes. A remote frame, much like a data frame in structure, includes the following basic elements: the ID field, DLC, CRC, ACK, and EOF. It differs from a data frame in that it doesn't include any data fields. Instead, through the ID field, a remote frame specifies the data frame that is to be

sent back from another node. The remote frame functionality allows an effective and flexible way of communication across a CAN network, in which nodes request data from other nodes without having to keep watching the bus for incoming messages, reducing traffic on the bus and increasing the overall efficiency of the network. This becomes extremely important when applied to nodes that require data on a sporadic basis.[7]

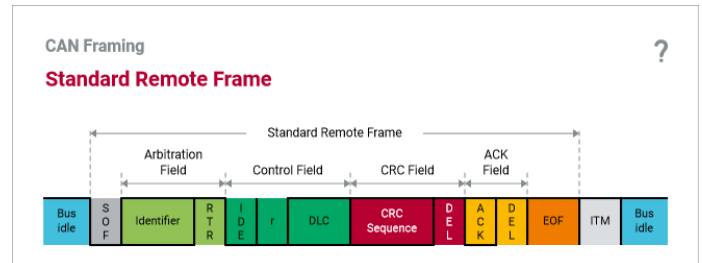


Fig. 3. Standard Remote Frame [6]

Error frame: Error frames in the CAN protocol provide the basic mechanism for error detection and error handling while communication between nodes is active. On detecting any error—whether due to transmission failures, electrical noise, or any other fluctuation—the current data transfer is immediately aborted, and an error frame would be sent out instantaneously. Unlike data frames or remote frames, which carry user data or request information, the structure of error frames is simplified for fast detection and resolution. Here, an error frame simply contains two types of elements: the error flag and the error delimiter. The error flag marks the very beginning of the frame to notify the receiving nodes that something has gone wrong during transmission. This flag, therefore, helps nodes quickly identify erroneous data on the bus for an appropriate response. It marks the end of the error frame and clearly defines its conclusion, thus distinguishing it from regular data frames that follow the error flag. This small structure then enables any node to handle and counteract errors efficiently, guaranteeing integrity and reliability in data transmission over a CAN network.[7]

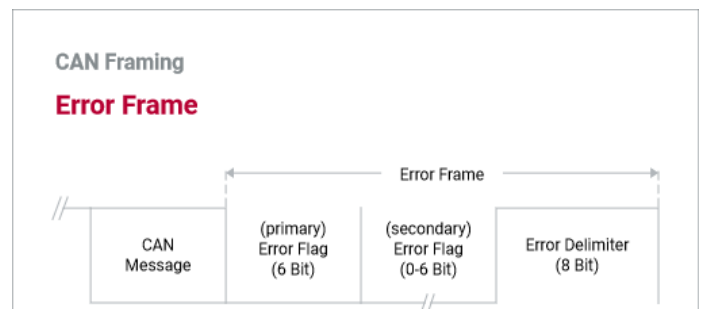


Fig. 4. Error Frame [6]

Bit stuffing: Bit stuffing is a methodology that ensures the avoidance of periods of long sequences of identical bits (control signals or errors in transmission) at the receiving

end of any communication by retaining correct timing and alignment between the transmitting and receiving nodes. During transmission, whenever five continuous bits of the same logic level are found in the data stream (all '0's or all '1's), a complementary bit is automatically introduced by the transmitter to break up the sequence of the five identical bits. This process of stuffing a '1' bit is called bit stuffing; it balances the number of transmitted '1's and '0's in the data and makes them recognizable to the receiver in case of noise or any interference on the CAN bus. Once the stuffed bits are received, the receiver recognizes them during the course of decoding and removes them to reconstruct the original data. The mechanism also provides integrity to the data and increased reliability in CAN networks since the risks related to synchronization errors are reduced and communication between nodes is homogeneous.[6]

Arbitration: The arbitration corresponds to the basic rule of the CAN protocol, defining how the nodes on the bus prioritize and manage access to transmit their messages. CAN communication allows multiple nodes to send messages simultaneously during the communication process. Arbitration will ensure that at any time, only one message—identified by its ID—goes out for transmission, thereby preventing data collision and ensuring orderly communication. The arbitration process in a CAN bus is decentralized, with no central arbiter controlling access. Instead, nodes can dynamically determine their priority based on the unique binary representation of messages encapsulated in their IDs. This means that lower numerical IDs have higher priorities, and nodes with critical or time-critical information get an opportunity to go first. In the event that two nodes start sending a message simultaneously with the same IDs, if there is an inbuilt arbitration logic within CAN, it guarantees that the node that is sending the dominant bit on the bus—a logical '0'—will win the arbitration and is allowed to transmit; the other node will then back off and try again.[6]

check (CRC) is a method used to detect errors during the transmission of CAN communication. Every CAN frame normally includes a CRC sequence at the end of a data field, which may be either a data frame or a remote frame. The computation of CRC will therefore be based on every element of a frame, which includes ID, DLC, and data fields, among other bits that bear a bearing on the computation. Now, when the node receives a frame, it will recalculate the CRC executed on the basis of the data received, which will be matched with the value the CRC sender has transmitted. If the calculated value of CRC matches the received CRC, then it is considered to be error-free; otherwise, if there is a mismatch, which indicates a transmission error, then the frame will be discarded by the receiving node, and the receiving node may request a retry. [6]

Acknowledgment (ACK):): ACK is a mechanism through which nodes affirm the successful reception of a transmitted frame. When sending a frame, a transmitter shall watch the bus for an ACK bit. If it detects that all the nodes connected to the bus have received the frame without error, it shall expect to receive an ACK bit. The absence of an ACK bit simply means that at least one node has not received the frame correctly; hence, it triggers the sender for error handling procedures, especially re-transmission. [6]

V. CAN BUS ACCESS

The CAN bus access operates on the principle of decentralized arbitration, ensuring that the nodes transmit messages in an orderly manner. In the case when multiple nodes wish to have access to the bus simultaneously, the method used by the CAN is CSMA/CA, which involves carrier sense multiple access with bitwise arbitration. It follows message prioritization through their respective IDs, with the lower numeric IDs having higher priorities. It implies that, at any time, the node on the CAN with the most important message, specified by its ID, shall have access to the bus and be able to send a message without competition from other nodes sending lower-priority messages. Because this is a full, decentralized arbitration mechanism for access to the bus, it can accomplish efficient usage of the bus bandwidth in real-time systems, especially where time-critical data needs to be sent. While CAN itself provides several mechanisms for protecting higher-priority messages against lower-priority ones, bad system design or configuration will still create scenarios where lower-priority CAN messages are delayed, even risking never being sent. The above illustrates the necessity of careful ID assignment and network planning in the quest for the best bus access that ensures message prioritization within the CAN network.[7]

Bitwise arbitration determines which node gains priority to transmit messages on the bus when multiple nodes attempt to communicate simultaneously. During the arbitration process, nodes compare their message IDs bit by bit, starting with the most significant bit. On the CAN bus, nodes transmit their message bits one after another. Nodes monitor the bus as each bit is transmitted, looking for both 'dominant' (a logic '0') and

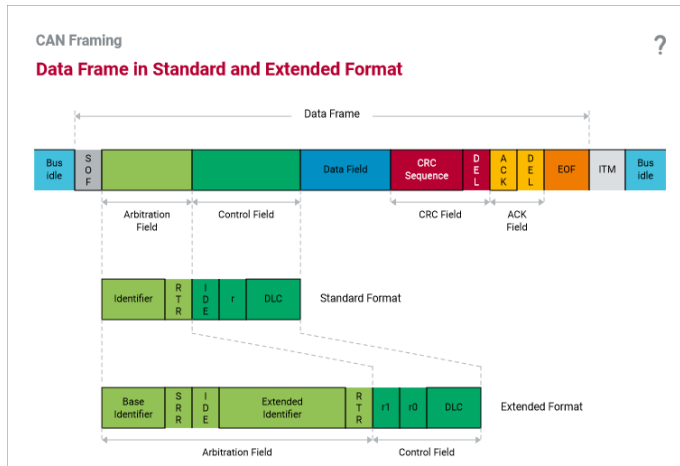


Fig. 5. Data Frame Extended Format [6]

Cyclic Redundancy Check (CRC): Cyclic redundancy

'recessive' bits. If, in a particular position in the transmission, a node sends a dominant bit and another node sends a recessive bit, the node that sent the dominant bit will continue to assert its dominance. The process is repeated repetitively, with nodes arguably continually transmitting their bits and watching the dominant bits until only one node has its bit dominate all of the time in the arbitration sequence; such a node then proceeds to send its message first, thus establishing its message as having priority over others on the bus as shown in Fig.6.[7]

The need for prioritization in the Controller Area Network arises from the fact that, in a system with numerous nodes competing to transmit at the same time, only one can occupy the bus. Prioritization itself is grouped into four major categories: high, medium, low, and very low. The priority of CAN messages will thus be mostly predetermined by their ID, which it transmits bit by bit from most significant to least significant. CAN has implemented a Wired-AND bus logic along with arbitration logic to handle very efficiently the priorities of the messages. The lower the numerical value of the identifier, the higher the message's priority will be. For instance, an ID with a value closer to zero implies higher priority and thus avoids delay in sending vital messages as shown in Fig.7.[7]

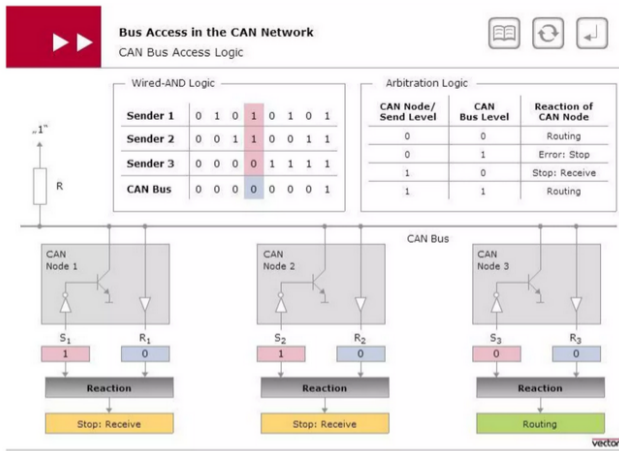


Fig. 6. Access Logic [7]

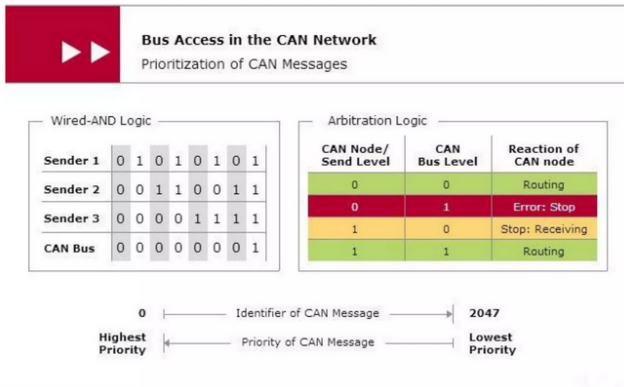


Fig. 7. Prioritization of CAN Messages [7]

VI. CAN DATA PROTECTION

CAN data protection consists of strategies and protocols developed to protect the integrity, confidentiality, and availability of data in transmission over the controller area network. Since CAN is a broadcast-based protocol, several nodes use the same bus. Therefore, it becomes vital that security regarding the data be ensured to avoid unauthorized access and tampering. The implementation includes strong message filtering and access control mechanisms to protect the data in the CAN. Configuring specific settings to make them accept only those messages relevant to their function could reduce the risks of unauthorized nodes injecting malicious data into the network. Encryption techniques can be applied from outside to protect sensitive information, and authentication protocols can be used to verify that messages are originating from trusted sources. Physical security is not less important in the protection of CAN data. This includes shielding against EMI, access security to the CAN nodes, and the isolation of critical parts physically that decreases associated risks due to physical tampering or unauthorized access. To ensure integrity, CAN error detection uses CRC checks and similar error-detection mechanisms to detect errors that may have occurred during transmission. Proper error-handling protocols ensure that corrupted messages will either be corrected or simply removed from the system, so wrong data does not influence the operation. [7]

VII. DISADVANTAGES OF CAN BUS

The Controller Area Network (CAN) bus, while widely adopted for its efficiency and reliability, has several inherent limitations. Its maximum data rate is 1 Mbps, and that may seem pretty low for applications that involve high-speed data transmission. A delay may happen to be witnessed in lower-priority messages on busy networks because of this priority-based arbitration inherent in CAN. Furthermore, there are no built-in security features on the CAN bus, such as encryption, which can help protect it from the rising threat of cybersecurity. Physical bus faults may knock out the whole network; furthermore, there is a limitation in distance per segment, which harms scalability, while fault diagnosis difficulties raise maintenance costs. CAN continues to be preferred for automotive and industry applications due to deterministic communication and cost in the specific operational envelope. [9]

VIII. CONCLUSION

The CAN bus protocol has been able to hold its position as one of the most important communication standards in the automotive and industrial worlds due to increased reliability, efficiency, reduced wiring complexity, and a few other reasons. This paper aims to provide an overview of the architecture, standards, and implementation details of the CAN bus, underlining its merits for real-time data interchange, error detection, and fault confinement. Although it has a maximum data rate of 1 Mbps and it is subject to physical failures, the robustness in the design of the CAN protocol—framing, arbitration, and protection of data—ensures dependable operation in noisy

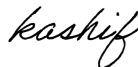
environments. The fact that CAN is still evolving and being applied to current systems underlines its validity and hence shows the need for further improvements to suit the future requirements of different sectors.

REFERENCES

- [1] S. Corrigan, "Introduction to the Controller Area Network (CAN)," SLOA101B, Industrial Interface, August 2002; revised May 2016.
- [2] H. M. Boland, M. I. Burgett, A. J. Etienne, and R. M. Stwalley III, "An Overview of CAN-BUS Development, Utilization, and Future Potential in Serial Network Messaging for Off-Road Mobile Equipment," July 2021. DOI: 10.5772/intechopen.98444.
- [3] ISO, "Road vehicles — Controller area network (CAN) Part 1: Data link layer and physical signalling," ISO 11898-1:2015, Edition 2, 2015. [Online]. Available: <https://www.iso.org/standard/72429.html>.
- [4] ISO, "Road vehicles — Controller area network (CAN) Part 2: High-speed CAN (HS CAN)," ISO 11898-2:2016, 2016. [Online]. Available: <https://www.iso.org/obp/ui/#iso:std:iso:11898:-2:en>.
- [5] ISO, "Road vehicles — Controller area network (CAN) Part 3: Low-speed CAN (LS CAN)," ISO 11898-3:2007, 2007. [Online]. Available: <https://www.iso.org/obp/ui/es/#iso:std:iso:11898:-3:ed-1:en>.
- [6] "Virtual Vector Academy," [Online]. Available: <https://elearning.vector.com/mod/page/view.php?id=333>. Accessed: Jul. 5, 2024.
- [7] A. Tiwari, "A Seminar Report on CAN BUS PROTOCOL," [Online]. Available: <https://www.slideshare.net/abhinawambitious/a-seminar-report-on-can-bus-protocol#19>. Accessed: Jul. 5, 2024.
- [8] "CAN Communication Protocol," [Online]. Available: <https://microcontrollerslab.com/can-communication-protocol/>. Accessed: Jul. 5, 2024.
- [9] D. Jiang and Y. Chen, "Schedulability Analysis for CAN Bus Messages of Different Transmission Types," 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9779187>. Accessed: Jul. 5, 2024.

IX. DECLARATION OF ORIGINALITY

I hereby declare that I myself have written this paper, and that I have not used any external sources other than those mentioned. Anything borrowed from a different source, whether phraseology or idea, is duly acknowledged. I further declare that this paper has not previously been submitted for any course or examination, in this form or in a similar version.



Kashif Raza
Lippstadt, 01.07.2024

I hereby declare that I myself have written this paper, and that I have not used any external sources other than those mentioned. Anything borrowed from a different source, whether phraseology or idea, is duly acknowledged. I further declare that this paper has not previously been submitted for any course or examination, in this form or in a similar version.



Ammar Imran Khan
Lippstadt, 01.07.2024