World Electric Vehicle Journal

MDPI

Article

Development of an Improved Communication Control System for ATV Electric Vehicles Using MRS Developers Studio

Natthapon Donjaroennon [1,*], Wattana Nambunlue [2], Suphatchakan Nuchkum [2] and Uthen Leeton [1,*]

[1] School of Electrical Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand
[2] School of Mechatronic Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand; m6601263@g.sut.ac.th (W.N.); suphatchakan@sut.ac.th (S.N.)
* Correspondence: d6500467@g.sut.ac.th (N.D.); uthenleeton@sut.ac.th (U.L.)

Abstract: Transmission, energy management, and distribution systems are critical components of modern electric vehicles, encompassing all sectors of the power system through communication control technology. One widely used communication system in electric vehicles is the Controller Area Network (CAN). This research aims to investigate the development of CAN BUS technology, adapted from large trucks, to control the communication system within an ATV electric vehicle using a communication format similar to bus Communication. The communication control system includes several components: the engine switch, headlight, turn signal, emergency light, horn, forward/reverse gear, and accelerator. The system's communication protocols were developed using MRS Developers Studio version 1.40 software to create the data transmission and reception formats for the vehicle's components. The communication system employs three PLC 1.033.30B.00 type E control boxes, each with limited analog and digital input/output ports. The sequence of communication control begins with the engine start/stop operation, as the system will not function unless the engine is started first. The headlight operation is processed within the CAN BUS1 control box. Simultaneously, the turn signal and emergency light functions are controlled by CAN BUS1 and displayed on both the CAN BUS2 (front of the vehicle) and CAN BUS3 (rear of the vehicle) control boxes. Additionally, the accelerator function is managed within the CAN BUS2 control box and displayed on the CAN BUS3 control box. However, this operation is contingent upon the forward/reverse gear selection, managed by CAN BUS1 and processed by CAN BUS3. All system operations are designed within the software's programming paths. The communication system operates using CAN-High and CAN-Low lines, and communication data fields can be monitored using the PCAN-View software version 4.2.1.533. This study demonstrates the feasibility and effectiveness of adapting CAN BUS technology for ATV electric vehicles, providing insights into the integration and control of various vehicular components within a unified communication framework.

Keywords: controller area network; CAN BUS; communication and control system; MRS developers studio

Citation: Donjaroennon, N.; Nambunlue, W.; Nuchkum, S.; Leeton, U. Development of an Improved Communication Control System for ATV Electric Vehicles Using MRS Developers Studio. *World Electr. Veh. J.* **2024**, *15*, 303. https://doi.org/10.3390/wevj15070303

Academic Editors: Quan Yuan and Xiaoyuan Fu

Received: 12 June 2024
Revised: 24 June 2024
Accepted: 27 June 2024
Published: 9 July 2024

1. Introduction

Communication in electric vehicles encompasses various types, such as Controller Area Network (CAN BUS) [1–4], which employs a highly stable network communication system capable of controlling multiple systems simultaneously. Local Interconnect Network (LIN Bus) [5,6] is used for controlling devices that require stability without high-performance demands. Flex Ray [7,8] is utilized for systems needing fast processing and complex control, such as drive control systems. Media Oriented Systems Transport (MOST) [9–11] manages communication for audiovisual systems, ensuring high-quality sound and image. Ethernet [12,13] is suitable for connecting to the internet or external networks, facilitating rapid data sharing or communication.

The rapid advancement of technology in electric vehicles over the past decade has significantly impacted various types of vehicles, particularly all-terrain vehicles (ATVs) designed for heavy-duty and specialized applications. Enhancing communication efficiency between the internal systems of these vehicles is crucial for improving both performance and safety. One of the key technologies facilitating inter-system communication in vehicles is the CAN BUS, a communication system designed to enable fast and efficient data exchange between electronic devices within a vehicle [14–16]. The CAN BUS is widely recognized in the automotive industry for its reliability and robustness in harsh environmental conditions. This system ensures smooth and secure data transmission between Electronic Control Units (ECUs). Developing software to support the functionality of the CAN BUS is essential for enhancing the operational efficiency of electric vehicles.

J1939 [17] is a standard protocol based on the CAN BUS [18], defined by the Society of Automotive Engineers (SAE) for data communication in large vehicles such as trucks and buses. It specifies the format and structure of messages transmitted over the CAN BUS, including identifiers indicating the type and priority of the data. Each message has a unique identifier, enabling ECUs to accurately interpret and respond to the data. J1939 prioritizes messages, allowing more critical data to be transmitted first, and supports up to 8 bytes of data per message, with multi-packet techniques for longer data transmissions. Additionally, J1939 incorporates error detection and correction mechanisms to ensure data accuracy. This paper explores the application of the J1939 communication format in ATV electric vehicles, utilizing MRS Developers Studio software version 1.40 to design and manage the communication system.

High speed communication and low speed communication are communication standards used for data transmission and reception in electric vehicles, communication devices, and machine learning devices. The primary distinction between these standards lies in their data transmission and reception speeds within the CAN controller. ISO 11898-2 [19] is designated for high-speed communication [18], defined in the range of 10 kbit/s to 1 Mbit/s, whereas ISO 11898-3 [20] is specified for low-speed communication, operating within the range of 10 kbit/s to 125 kbit/s. Moreover, MRS Developers Studio can define data transmission and reception formats based on these communication standards when developing software for electric vehicles.

Developing ATV electric vehicles necessitates a communication system that is both swift and precise to effectively manage various subsystems. MRS Developers Studio is a powerful tool for developing and testing electronic systems in vehicles [21], particularly when used in conjunction with the CAN BUS, a globally accepted communication standard in the automotive industry. Utilizing this software enables detailed analysis and improvement of communication between different control units within the vehicle.

The primary objective of this study is to develop an internal communication system for ATV electric vehicles using CAN BUS technology in conjunction with MRS Developers Studio software. The goal is to enhance the seamless integration and efficiency of various systems within electric vehicles. This development aims to enable ATV electric vehicles to operate more effectively and safely. Additionally, it seeks to reduce maintenance costs and improve the ability to diagnose issues within the vehicle, which is critical for the future advancement of electric vehicles.

Using MRS Developers Studio will facilitate the creation and testing of a highly efficient communication system for ATV electric vehicles by simulating various operational scenarios and analyzing data received from the CAN BUS in detail. This software also provides tools to fine-tune and optimize the communication system's performance.

Developing a communication system for ATVs using CAN BUS technology and MRS Developers Studio is of paramount importance, as it ensures smooth and rapid data exchange between various internal systems of the vehicle. Efficient inter-system communication directly enhances vehicle performance, operational safety, and the capability for long-term maintenance and problem analysis. This research focuses on testing and developing communication techniques using CAN BUS and MRS Developers Studio to

simulate and analyze various operational scenarios of ATVs. The results will be applicable in modernizing and improving internal communication systems within electric vehicles, benefiting the future development of the electric automotive industry [22–24].

This paper presents the development of a communication control system for ATV electric vehicles using CAN BUS technology. The system controls the ECU, including the engine switch, headlights, turn signals, emergency lights, horn, forward/reverse gear, and accelerator, as depicted in the ATV circuit diagram. The system is designed using MRS Developers Studio software, tailored for CAN BUS protocols. Communication control unit uploads are facilitated via Peak CAN (PCAN) cables, connecting the CAN BUS controllers to a computer. Signal connections between control units use twisted pair wires for CAN-High and CAN-Low to reduce noise, with a 120 Ω resistor in parallel to support high-speed communication. The programming follows the J1939 standard, defining communication addresses, starting bits, and data lengths to ensure accurate and efficient data transmission and reception.

2. System Discussion about ATV Electric Vehicles

An ATV is a vehicle characterized by its ability to operate on various terrains such as dirt paths, designed with large wheels to navigate through rugged landscapes. ATVs are commonly used for recreational activities and sports. Additionally, they find application in farming for tasks like soil covering or informal land improvement.

In an era where communication technology and vehicle control systems are rapidly advancing [25], the challenge of developing systems that can support fast and reliable communication has become crucial. This is especially pertinent for ATV electric vehicles, which require robust capabilities for heavy-duty operations and specialized tasks. In this article, we will discuss the development of CAN BUS technology in conjunction with the MRS Developers Studio software, which is a key tool for simulating and testing communication between various vehicle systems.

The CAN BUS is a communication technology designed to facilitate fast and efficient data exchange among electronic devices within vehicles. The CAN BUS utilizes a differential signal and only two signal wires (CAN-High and CAN-Low), which helps reduce the complexity of wiring systems, making installation and maintenance easier. Additionally, the CAN BUS incorporates a message prioritization system, which minimizes communication latency. It also features automatic error detection and correction capabilities, enhancing the reliability of communication.

MRS Developers Studio software is a crucial tool for simulating and testing communication protocols in a virtual environment before actual deployment. The presented image illustrates the use of MRS in testing the communication systems within ATV electric vehicles. MRS enables developers to test and verify the operation of various systems for durability and optimal performance [26–28]. Furthermore, MRS assists in analyzing communication data and system efficiency for efficient management and rapid troubleshooting. MRS Developers Studio is a sophisticated and adaptable software tool developed by MRS Electronic GmbH & Co. KG (Rottweil, Germany). This software is engineered to address the complexities of contemporary embedded system development, with a particular emphasis on the automotive and industrial sectors. Featuring advanced capabilities and comprehensive support, MRS Developers Studio empowers developers to create efficient and reliable applications across a diverse range of embedded systems. This paper uses version 1.40.

In Figure 1, the communication system in the ATV electric vehicle includes various control devices such as the Electronic Control Unit (ECU) and other peripherals connected via the CAN BUS. This setup allows for efficient data exchange between different systems while minimizing signal interference. The use of MRS Developers Studio for simulating and testing system operations before deployment enables developers to quickly identify and resolve issues. This approach helps reduce development time and costs significantly.

Figure 1. MRS Developers Studio software with hardware communication for ATV.

The ATV communication control system consists of the CAN BUS control box model PLC 1.033.30B.00 type E. Due to limitations in the CAN I/O ports (six analog pins, eight digital pins), up to three control boxes are necessary to manage communication devices including the engine switch, headlight, turn signal, emergency light, horn, forward/reverse gear, and accelerator. The CAN-High and CAN-Low connection wires are twisted to reduce signal interference. MRS Developers Studio software utilizes a communication format similar to the J1939 standard used for large buses or trucks in designing communication systems. Refer to Figure 2 for the communication development format of each CAN BUS box.

```
[Control Unit 1]        [Control Unit 2]        [Control Unit 3]
   |                       |                       |
   | CAN_H                 | CAN_H                 | CAN_H
   |-----------------------|-----------------------|
   | CAN_L                 | CAN_L                 | CAN_L
   |-----------------------|-----------------------|
120Ω                    120Ω                    120Ω
(Termination Resistor)  (Termination Resistor)  (Termination Resistor)

Devices:
- Engine Switch     ----> [Control Unit 1]
- Headlight         ----> [Control Unit 1]
- Turn Signal       ----> [Control Unit 1] <--> [Control Unit 2] <--> [Control Unit 3]
- Emergency Light   ----> [Control Unit 1] <--> [Control Unit 2] <--> [Control Unit 3]
- Horn              ----> [Control Unit 1] <--> [Control Unit 2]
- Forward/Reverse   ----> [Control Unit 1] <-------------------------> [Control Unit 3]
- Accelerator       ---->                  [Control Unit 2] <--> [Control Unit 3]

PCAN Interface:
[Computer] <------> [CAN BUS]
```

Figure 2. Communications control system on ATV electric vehicle.

In high-speed CAN BUS communication (ISO 11898-2), 120 Ω termination resistors are crucial for optimal signal integrity and reliable data transmission. These resistors mitigate signal reflection, which occurs when electrical signals travel through the cable and are reflected if the cable is not properly terminated. Signal reflection can lead to transmission errors due to signals bouncing back and forth within the cable. By placing 120 Ω resistors at both ends of the CAN BUS cable, signal reflection is minimized, resulting in stable and efficient communication. The characteristic impedance of the CAN BUS cable is approximately 120 Ω, and using termination resistors ensures that the cable's impedance matches the system's impedance, enhancing signal transmission efficiency and reducing signal loss. Furthermore, these resistors help suppress noise generated by signal reflections, improving the system's immunity to external noise and ensuring clear signal transmission. This impedance matching and noise reduction are essential for maintaining the high-speed performance and reliability of CAN BUS networks.

The characteristic impedance of the CAN BUS cable (Z_0) is determined by the following equation:

$$Z_0 = \sqrt{\frac{L}{C}} \tag{1}$$

where

L is the inductance per unit length of the cable.
C is the capacitance per unit length of the cable.

To prevent signal reflection at the cable ends, the termination resistance (R_T) should be equal to the characteristic impedance of the cable:

$$R_T = Z_0 \tag{2}$$

In a CAN BUS system, the characteristic impedance of the cable is typically designed to be around 120 Ω. Therefore, using a 120 Ω resistor at both ends of the CAN BUS cable ensures proper impedance matching.

Signal Reflection Reduction: When the cable ends are terminated with resistors that match the characteristic impedance of the cable, the reflected voltage (V_r) at the cable ends is greatly reduced. The equation for reflected voltage is as follows:

$$V_r = V_i(\frac{Z_L - Z_0}{Z_L + Z_0}) \tag{3}$$

where

V_i is the input voltage at the cable end.
Z_L is the termination resistance.
Z_0 is the characteristic impedance of the cable.

For the case where $Z_L = Z_0$:

$$V_r = V_i(\frac{Z_L - Z_0}{Z_L + Z_0}) = V_i(\frac{0}{2Z_0}) = 0 \tag{4}$$

This means that the signal reflection will be zero when the termination resistance matches the characteristic impedance of the cable.

The CAN I/O ports of model 1.033.30B.00 type E includes a variety of analog (ANA0–ANA5) and digital ports (I/O0–I/O7) to accommodate different functionalities within the vehicle control system of an ATV. The analog ports (ANA0–ANA5) can connect to devices providing analog signals, such as accelerators within the ATV, with each port capable of reading values converted into digital signals for processing. Additionally, the digital ports (I/O0–I/O7) can serve as inputs or outputs to control various devices, such as switching on/off switches, motor gear controls, or receiving signals from different switches. These digital ports can connect to devices using digital signals, such as switches and relays, to

efficiently manage the operations of the vehicle system. The complete port connections are detailed in Table 1, and the interconnections of these ports are illustrated in Figure 3 of the circuit design communication and control system.

Table 1. Connection CAN I/O PLC model 1.033.30B.00. port.

PIN	NAME	PIN	NAME	PIN	NAME
01	GND.	08	KL15	16	I/O3
02	ANA 5	09	CAN-High	17	Vcc (0–30V)
03	ANA 4	10	CAN-Low	18	I/O4
04	ANA 3	12	Vcc (0–30 V)	19	I/O5
05	ANA 2	13	I/O0	20	I/O6
06	ANA 1	14	I/O1	21	I/O7
07	ANA 0	15	I/O2	22	5 V, Output

Circuit design communication and control system are shown in Figure 3.

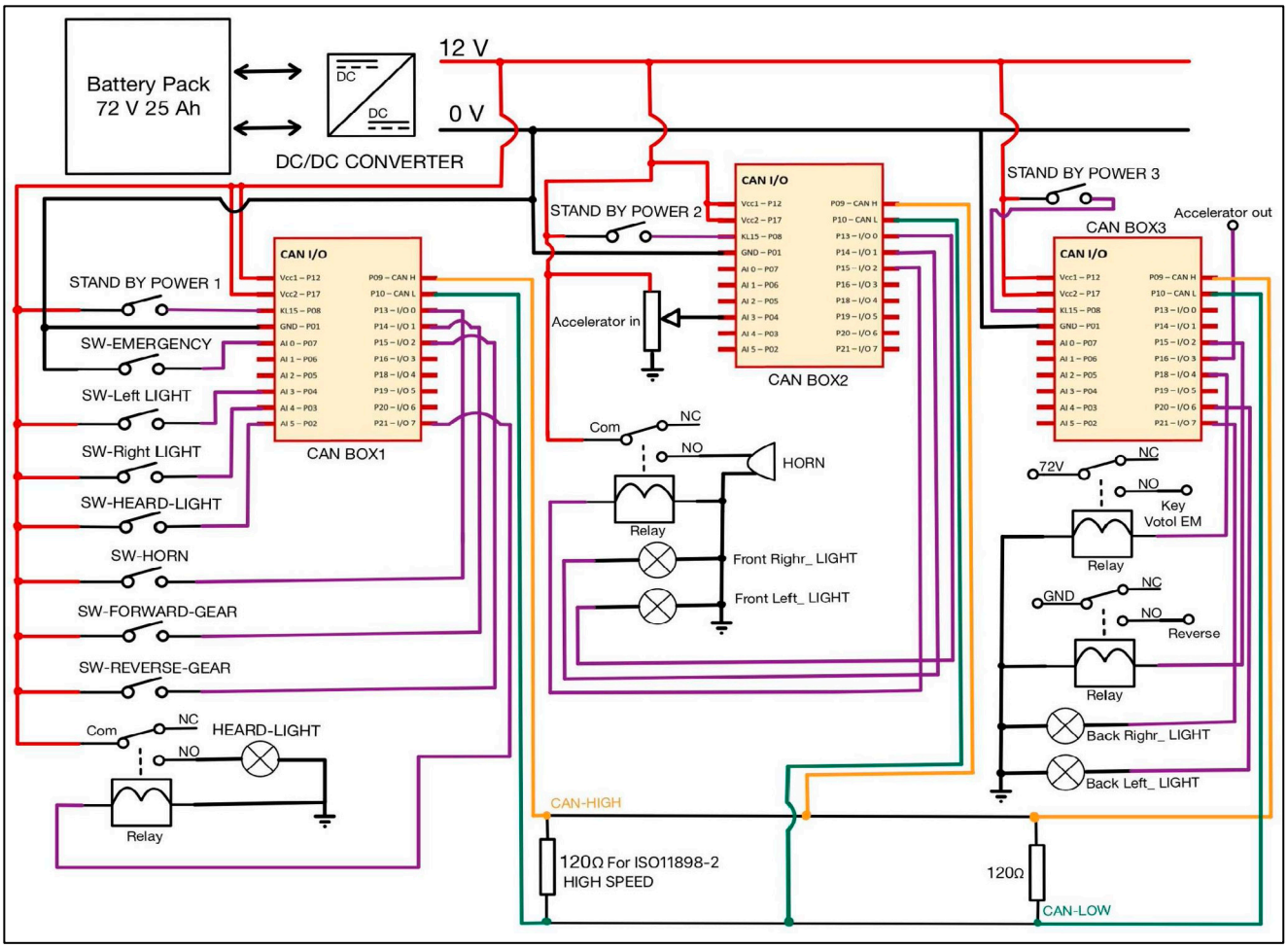


Figure 3. Circuit design communication and control system [use pin from Table 1].

2.1. Battery Packing

Packing together 20 cells of 3.7 V 25 Ah Li-ion NMC batteries [29–31] in series yields a total voltage of 74 V while maintaining a constant capacity of 25 Ah. This configuration is suitable for applications requiring high energy, such as electric vehicles or high-power

devices. The series connection increases the total voltage without altering the battery's capacity. This battery pack can be applied effectively in ATVs to enhance driving range and motor efficiency. Additionally, it allows the ATV to operate in diverse environments requiring high energy demands. Moreover, the battery pack requires a Battery Management System (BMS) and balancing equipment to ensure each cell operates safely and efficiently over time. The battery pack model configuration can be seen in Figure 4.

Figure 4. Battery pack and balance with BMS.

The battery cell packing or arrangement to achieve the desired voltage can be accomplished using the following equation for series connection:

$$V_{total} = V_{cell} \times n \quad (5)$$

where

V_{total} is the total voltage required.
n is the number of battery cells connected in series.
V_{cell} is the voltage of each individual battery cell.

2.2. About MRS Developers Studio Software

MRS Developers Studio facilitates efficient and convenient programming and communication via the CAN BUS. Users can configure relevant parameters, write control programs, simulate operations, and test with actual hardware easily. This ensures smooth and efficient development and maintenance of control systems in industries.

Developing CAN BUS technology in conjunction with MRS Developers Studio software enhances the reliability and performance of communication systems in electric vehicles [32–34], which is crucial for future automotive development. The ability to simulate and test system operations in detail helps developers create high-quality systems that meet user requirements effectively. By integrating these technologies, electric vehicles can have more efficient and reliable communication systems, which are vital for enhancing safety and operational efficiency in future automotive applications [35].

The ATV electric vehicle control system utilizes a 72 V 25 Ah battery pack (Figure 4), with voltage conversion to 12 V via a DC/DC converter for powering various systems. CAN BOX1, CAN BOX2, and CAN BOX3 manage the engine switch, headlight, turn signal [34], emergency

light, horn, forward/reverse gear, and accelerator. These control boxes communicate via CAN-High and CAN-Low signals with a 120 Ω resistor for network management.

2.3. Setting Program

Setting up the communication system for ATV electric vehicles using the CAN I/O control device with a step-by-step guide to using MRS Developers Studio is as follows: (1) Begin by launching the MRS Developers Studio software. (2) Go to the 'File' menu and select 'New Project', name your project as desired, and ensure you input the address correctly according to the CAN I/O box. The CAN I/O box is identified with an address shaped like PLC 1.033.30B.00; select the 'Revision' type as 'E' (Note: the electric vehicle communication will fail if the address and type do not match the CAN I/O box specifications). (3) Follow the order depicted in Figure 5, starting with number one. Use number one to update and configure the control system within the software. Number two is used to save your current project and any changes made. Numbers three and four are critical for setting up the data transmission between the CAN Bus test box and the CAN I/O box (ensure the transmission parameters match the requirements of your CAN Bus system). Number five is used to configure the response speed for data transmission (adjust the response speed according to the needs of your electric vehicle communication system). Number six is designated for writing and configuring the control system specific to the electric vehicle's operations. Number seven shows how to use both analog and digital pins for writing the control system. (Ensure correct pin assignments for proper control functions). Numbers eight and nine are employed to open and manage control during the programming phase. This is essential for testing and validating the communication setup.

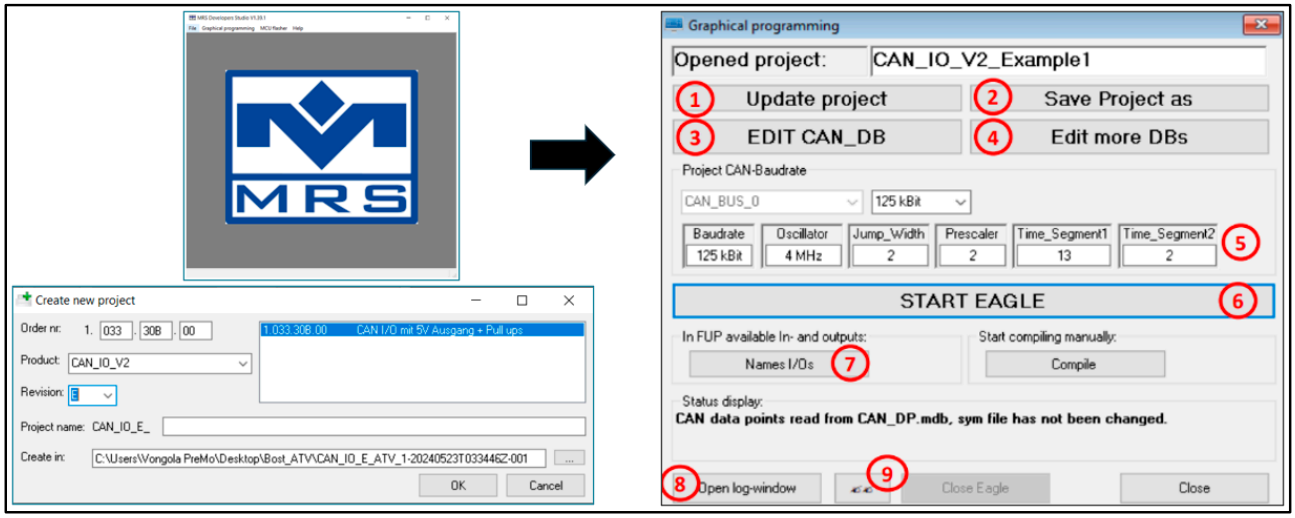


Figure 5. MRS Developers Studio software.

In the graphical programming window, communication uses 125 kbit/s (CAN-High). Click on the button "EDIT CAN_DB" (number three from Figure 5) to display the definitions of CAN blocks and CAN data points, as shown in Figure 6. The definition of CAN blocks is based on an 11 or 29 (extended) identifier, a name, and data content that can be up to 8 bytes. A CAN data point is a variable within a CAN block, defined by several bits within an 8-byte array. Multiple CAN data points can exist within the same CAN block if there are bits available to be assigned.

Each variable in the communication control program must have its bit start and bit length correctly specified. This alignment ensures that the communication control settings match the data addresses accurately, facilitating precise and effective communication control. These configurations are essential for maintaining a robust and efficient CAN BUS communication control system, ensuring that all components communicate seamlessly and reliably.

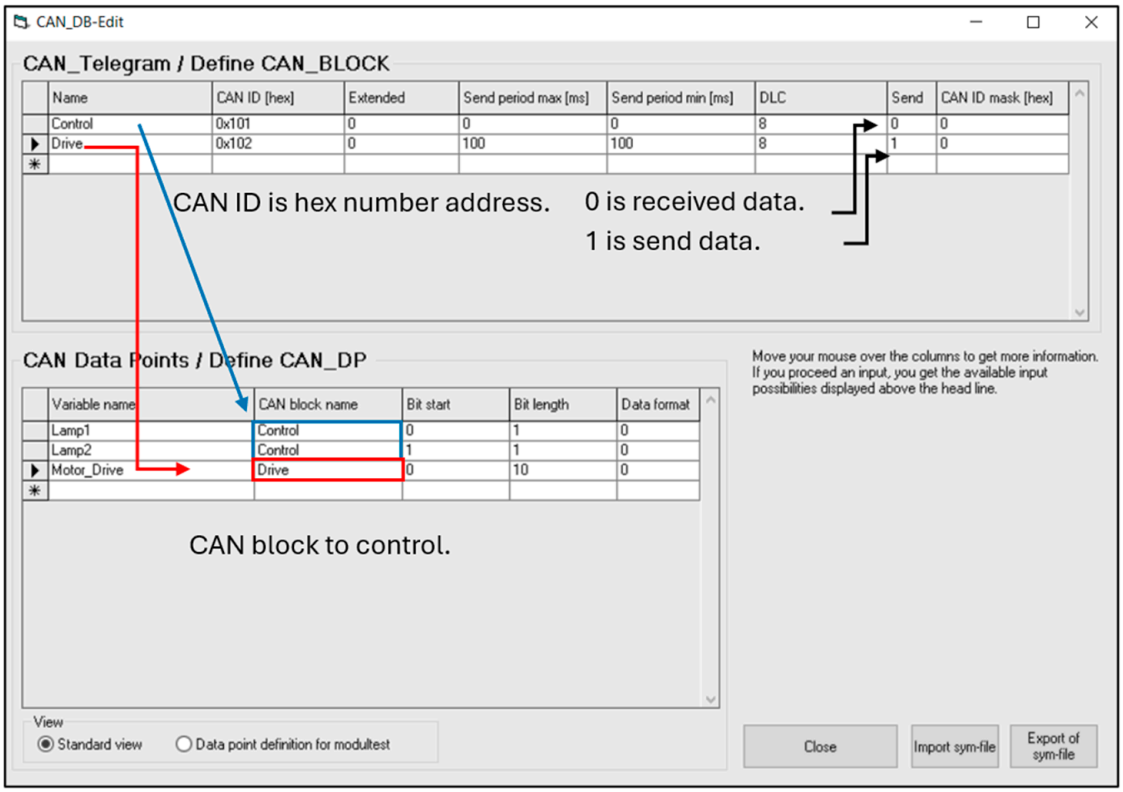


Figure 6. CAN database and CAN data points.

Effective communication control in CAN BUS systems relies on several key settings. Assigning CAN IDs in hexadecimal format ensures each controller has a unique identifier. Selecting the Extended Frame standard (29-bit identifiers) over the Standard Frame (11-bit identifiers) allows for handling more data. Setting the transmission speed in milliseconds is crucial for data flow, with specific values assigned for data reception and transmission. Additionally, configuring the bit start and bit length for each variable ensures precise communication control (This paper uses 29-bit identifiers).

CAN data points refer to the various signals or information transmitted within a vehicle's CAN system, facilitating communication between different devices. These data points are essential for control, analysis, and diagnostics within the vehicle. CAN data points and CAN DB are key components of the communication system using CAN, commonly used in electric vehicles and various automated control systems. CAN data points are defined by several bits within an 8-byte array (up to 64 bits), representing specific data to be transmitted over the CAN BUS. These can include the engine switch, headlight, turn signal, emergency light, horn, forward/reverse gear, and accelerator. Multiple CAN data points can exist within a single CAN block if there are enough bits available for definition.

The CAN database (CAN_DB) is a critical component for managing and storing information related to CAN data points. It typically includes the data structure, metadata, and mapping of signals, enabling an efficient interpretation and utilization of data from CAN data points. This database ensures that the data are accurately understood and effectively used for various applications within the vehicle's communication system. CAN DB, or CAN database, is the database used to manage and define the structure of CAN blocks and CAN data points. The CAN DB is used to create and manage CAN blocks, including the CAN data points within each block, ensuring efficient communication between devices connected to the CAN system. Programs like MRS Developers Studio are used to edit the CAN DB to configure and improve communication between devices in the CAN system. The programming screen can press START EAGLE by PIN and the communication system design format is shown in Figure 7.

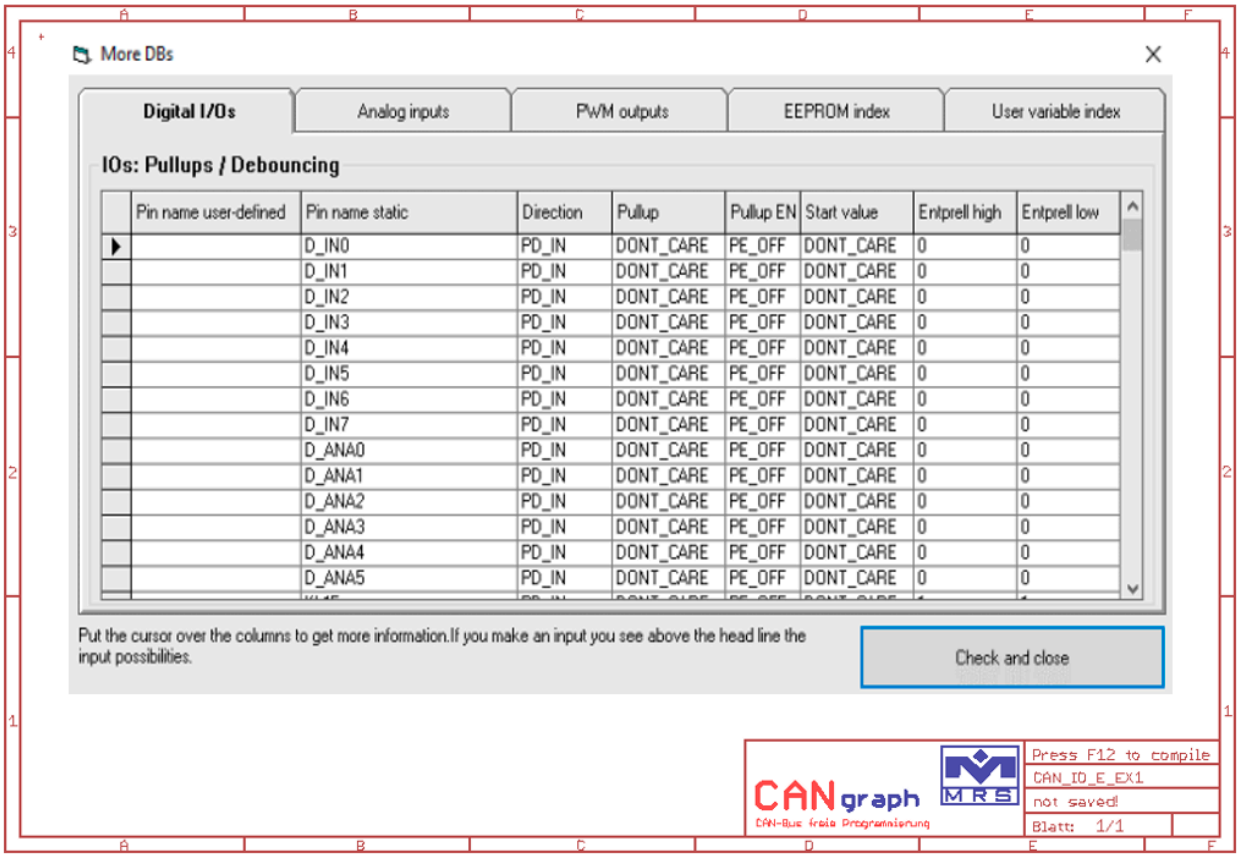


Figure 7. Pin analog and digital for writing programming.

For writing communication control systems using MRS Developers Studio software, it is necessary to use PINs that match the CAN BUS control box model PLC 1.033.30B.00. PIN names cannot be set outside of what the software specifies. PINs are divided into two main types: analog pins and digital pins. Analog pins are used for reading continuous values (temperature measurements, accelerations), while digital pins are used for reading or sending status signals (switch on/off status). In MRS Developers Studio, navigate to the Pins Configuration section, select the desired pins, and specify whether they are analog or digital. Use relevant blocks or modules in the software to write code controlling the pins. For analog pins, code can be written to read sensor values or send signals to other devices. For digital pins, code can be written to check the status of switches or control devices that require on/off status. After completing the code, test the pin's functionality by connecting it to real devices. Check if the pin operates as configured and adjust the code if necessary. Thus, using PINs in MRS Developers Studio facilitates efficient connection and control of various devices in the CAN system. The main steps include configuring pins in the software, writing control code, and testing and adjusting pin functionality as needed.

PCAN Interface is a device that connects a computer to a CAN network to transmit, receive, and analyze data within the CAN network. Produced by PEAK-System Technik, PCAN can connect via USB, PCI, PCIe, and other ports. It works with software such as PCAN-View to analyze and monitor data. It supports both CAN protocols and is used in automotive development and testing, industrial control, and research. PCAN Interface enhances the reliability of communication and data analysis within the CAN network, making data analysis straightforward and efficient.

3. Development of Control System Using CAN BUS Technology with MRS

This section discusses the components involved in communication [36–39] within ATV electric vehicles. This includes using CAN Bus controller equipment, communicating

between multiple controllers, and learning MRS Developers Studio software fundamentals. The procedure begins with communication between the device and the CAN Bus test board software, as shown in Figure 5 before it is installed on the ATV. The test is designed to assess the ability to operate various switches for starting the engine, headlights, turn signals, emergency lights, horn, forward/reverse gear, and the electric accelerator. Besides communication, wiring and the formation of the electric vehicle frame are crucial for achieving the desired results. The system operation is illustrated in Figure 8.

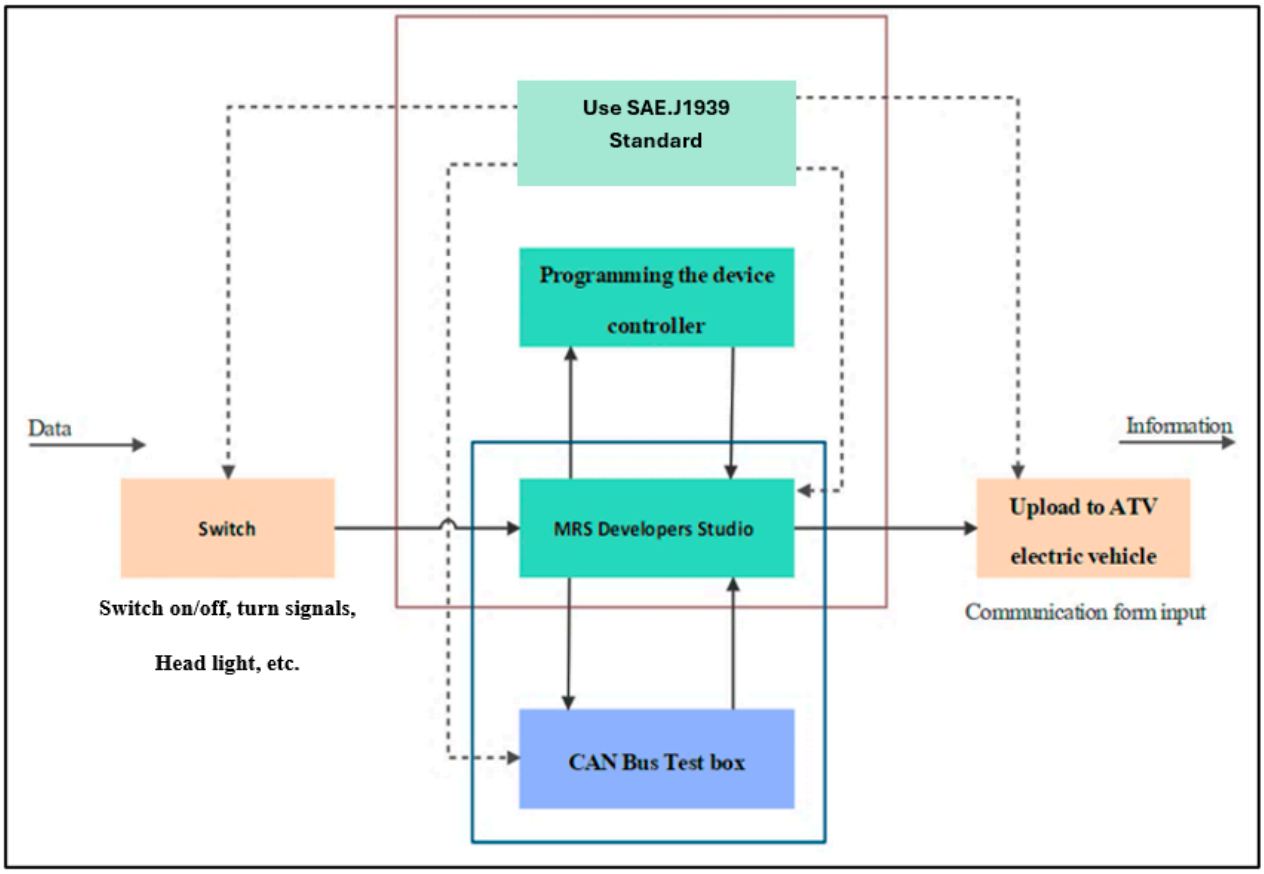


Figure 8. Communication control system with MRS Developers Studio method.

Write Programming

Programming in MRS Developers Studio for vehicle control is essential for efficient communication setup, enhanced security, and comprehensive testing. The software allows for precise configuration of CAN BUS parameters, secure access control through pins, and detailed monitoring of user activities. It also simplifies access rights management and enables extensive simulation and testing of control programs. These features ensure robust, secure, and efficient vehicle control systems, enhancing overall performance and reliability.

The use of pins in MRS Developers Studio, as shown in Figure 7, is essential because pins are used to limit and control access to specific programs or functions that require high security. This ensures that only authorized personnel can access and use these functions. Pins help prevent unauthorized individuals from modifying or accessing critical information. Using pins provides additional authentication that enhances the program's security level. When pins are used to access various functions, the system can track and record each user's activities, making monitoring and analyzing usage easier. Using pins helps define and manage access rights for different users within the system, making access rights management more efficient. Pins are a simple and quick method to control access without the need for complex authentication systems, reducing the complexity of managing access rights in the system.

The programming of the CAN BUS system begins with CAN BUS Box 1, which is responsible for receiving input from the key and necessitates the activation of the engine button for the system to function. The system remains inactive if the switch is not engaged. The turn signal transmission requires pressing a switch on CAN BUS Box 1, which then sends data to be displayed on CAN BUS Boxes 2 and 3. Furthermore, the control circuit driver for the gear system issues operational commands from CAN BUS Box 1, which are processed by CAN BUS Box 3. The program for CAN BUS Box 1 is detailed in Figure 9. This configuration ensures synchronized operation and communication across the CAN BUS network, facilitating efficient control and monitoring of the electric vehicle's systems.

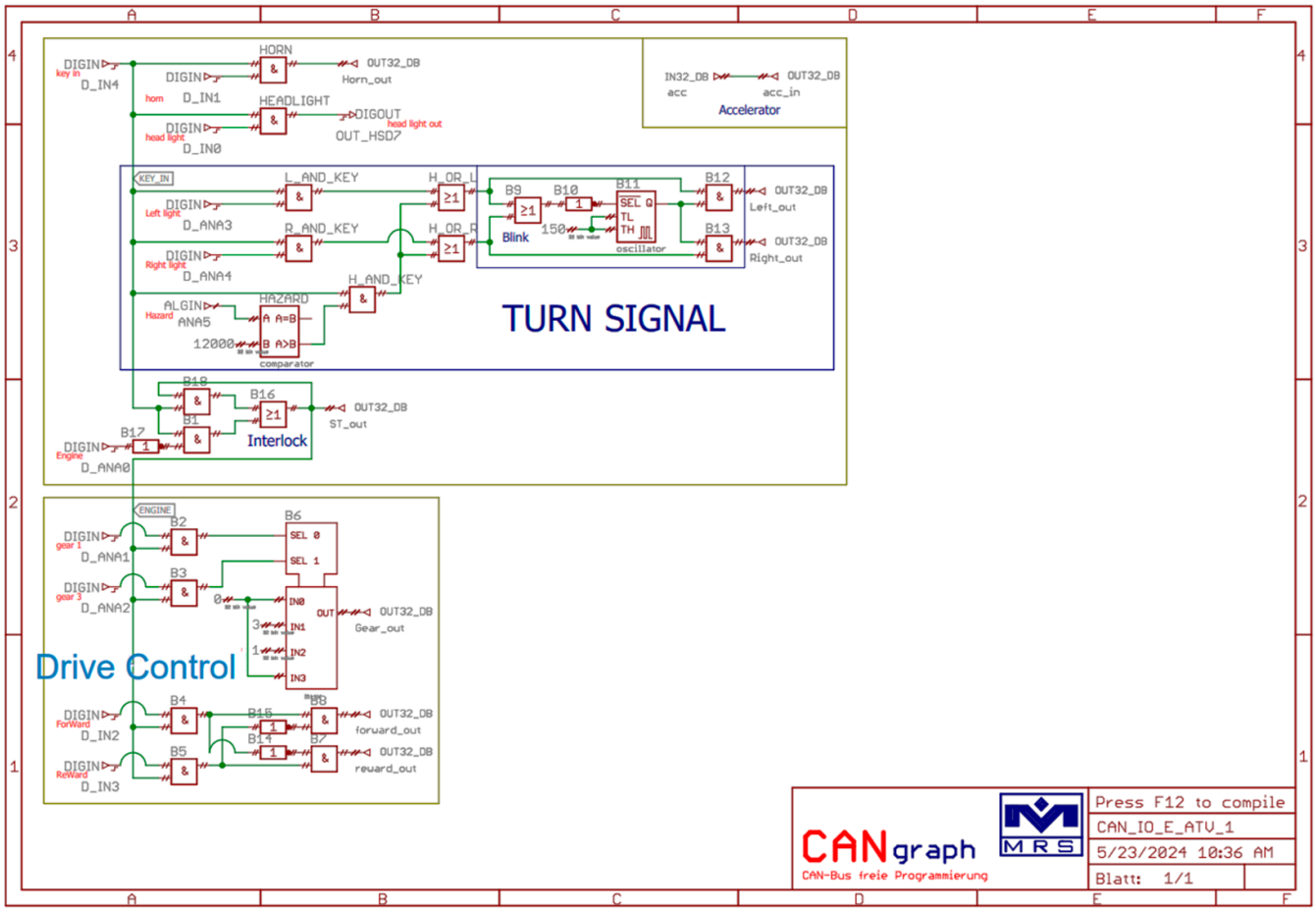


Figure 9. CAN BUS controller box number 1.

The configuration and design of the ATV of the communication system for the CAN BUS controller number 1, as shown in Figure 9, must align with the parameter values specified in Table 2, which shows the key parameters essential for effective system control.

These parameters must be meticulously configured and programmed to ensure the system operates efficiently and reliably. Properly setting these parameters ensures the effective development of the communication system using the CAN BUS, which is crucial for the overall performance and reliability of the vehicle.

The Baud Rate (BR) for a CAN BUS can be calculated using the following equation:

$$BR = \frac{F_{OSC}}{2 \times (BRP + 1) \times (TSEG1 + TSEG2 + 3)} \quad (6)$$

CAN BUS Box 2 is responsible for receiving the turn signal and displaying the signal from CAN BUS Box 1. Additionally, CAN BUS Box 2 sends accelerator signals to be

displayed on CAN BUS Box 3. The programming result for CAN BUS Box 2 is shown in Figure 10.

Table 2. CAN BUS controller number 1: CAN data point.

Name	CAN ID [hex]	Type of CAN	Send Period max [ms]	Send Period min [ms]	DLC	Status	CAN ID Mark [hex]
CAN_A_lighting	0 × 101	1	100	100	8	1	0
CAN_A_Drive	0 × 102	1	100	100	8	1	0
CAN_B_Accelerator	0 × 103	1	0	0	8	0	0
Monitor_C	0 × 104	1	0	0	8	0	0
monitor	0 × 105	1	100	100	8	1	0

CAN Data Points/Define CAN_DB

Variable name	CAN block name	Bit start	Bit length	Data format
Horn_out	CAN_A_lighting	0	1	0
Left_out	CAN_A_lighting	1	1	0
Right_out	CAN_A_lighting	2	1	0
Gear_out	CAN_A_Drive	0	2	0
forward_out	CAN_A_Drive	2	1	0
reward_out	CAN_A_Drive	3	1	0
ST_out	CAN_A_Drive	4	1	0
acc	CAN_B_Accelerator	0	10	0
acc_in	monitor	0	10	0

Note: Baud Rate: determines the communication speed of the CAN bus. CAN ID: provides a unique identifier for each message on the CAN network. Message Priority: sets the priority level for messages to manage traffic on the CAN bus. Data Length Code (DLC): specifies the number of data bytes in a message. Sampling Point: defines the specific time at which the CAN controller samples the bus level. Synchronization Jump Width: allows phase error correction to maintain synchronization. Bit Timing: configures the timing segments of a CAN bit to ensure reliable communication. Error Handling: establishes protocols for detecting and managing errors within the CAN network.

where

FOSC is the oscillator frequency.
BRP is the oscillator frequency.
TSEG1 is Time Segment 1.
TSEG2 is Time Segment 2.

Bit Timing in CAN BUS is determined by dividing the bit time into several segments using the following equation:

$$T_{bit} = T_{SYNC\_SEG} + T_{PROP\_SEG} + T_{PHASE\_SEG1} + T_{PHASE\_SEG2} \quad (7)$$

where

T_{SYNC_SEG} is the synchronization segment (1 Time Quantum).
T_{PROP_SEG} is the propagation segment.
T_{PHASE_SEG1} is Phase Segment 1.
T_{PHASE_SEG2} is Phase Segment 2.

Equations (1)–(7) are used to calculate and set various parameters of a CAN BUS system to ensure effective and stable communication. Developers should use these equations in the design and testing of CAN BUS systems to ensure that the parameters are configured correctly.

CAN BUS Box 2 is responsible for receiving the turn signal and displaying the signal from CAN BUS Box 1. Additionally, CAN BUS Box 2 sends accelerator signals to be displayed on CAN BUS Box 3. The programming result for CAN BUS Box 2 is shown in Figure 10.

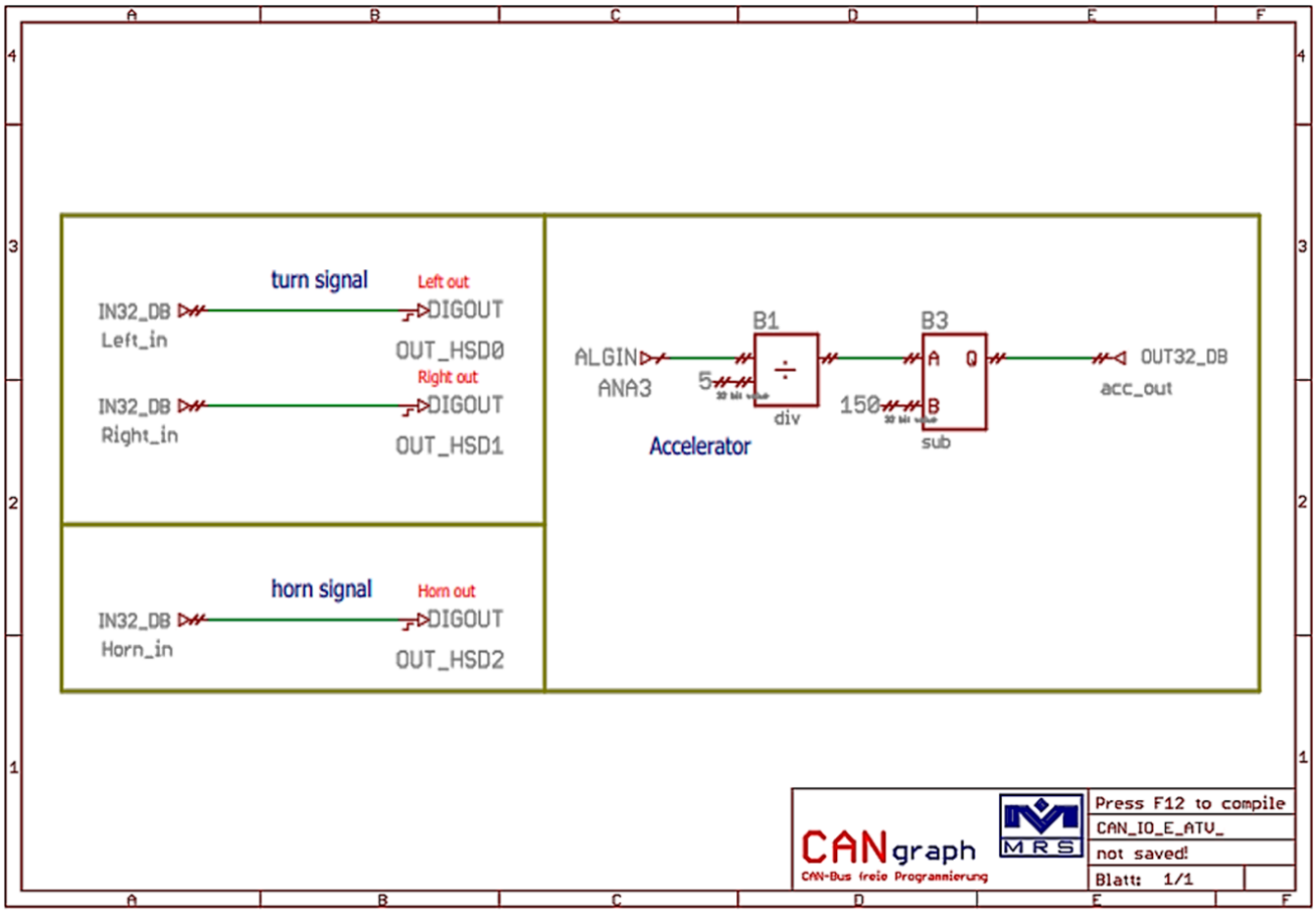


Figure 10. CAN BUS controller box number 2.

For the Configuration and Parameter Setting for Programming of CAN BUS Controller Number 2 (Figure 10, CAN BUS controller box number 2), the settings for the CAN BUS controller number 2 can be configured according to Table 3. The parameters are defined below.

CAN BUS Box 3 is responsible for receiving and displaying signals from CAN BUS Box 1. Additionally, CAN BUS Box 3 receives forward or reverse gear signals from CAN BUS Box 1 and accelerator signals from CAN BUS Box 2 to display the system control results. The programming for this is illustrated in Figure 11.

For the Configuration and Parameter Setting for Programming of CAN BUS Controller Number 3 (Figure 11, CAN BUS controller box number 3), the settings for the CAN BUS controller number 3 can be configured according to Table 4. The parameters are defined below.

Table 3. CAN BUS controller number 2: CAN data point.

Name	CAN ID [hex]	Type of CAN	Send Period max [ms]	Send Period min [ms]	DLC	Status	CAN ID Mark [hex]
CAN_A_lighting	0x101	1	0	0	8	0	0
CAN_B_Accelerator	0x103	1	100	100	8	1	0

CAN data points/Define CAN_DB

Variable name	CAN block name	Bit start	Bit length	Data format
Horn_in	CAN_A_lighting	0	1	0
Left_in	CAN_A_lighting	1	1	0
Right_in	CAN_A_lighting	2	1	0
acc_out	CAN_B_Accelerator	0	10	0
b_test	CAN_B_Accelerator	10	1	0

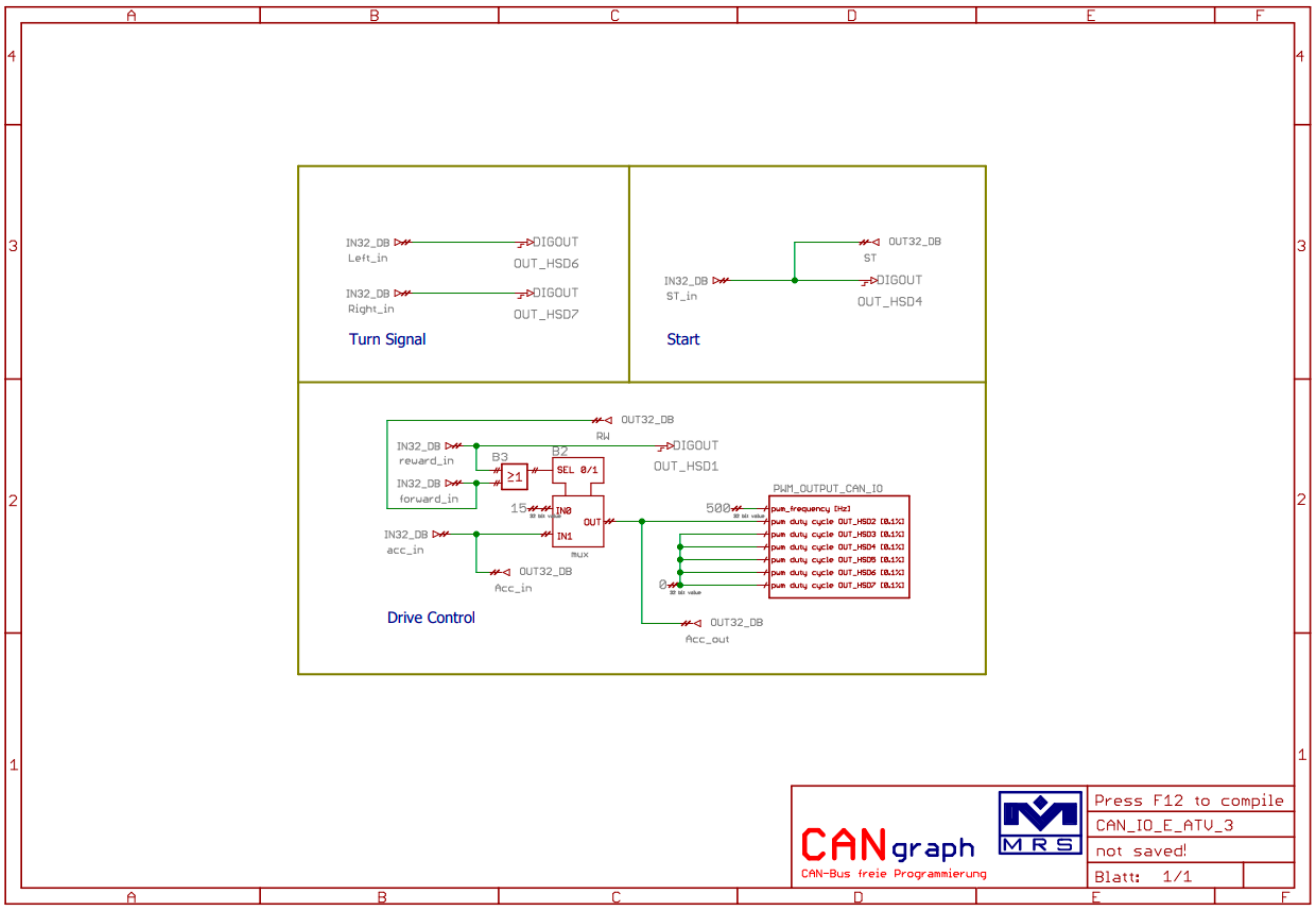


Figure 11. CAN BUS controller box number 3.

Table 4. CAN BUS controller number 3: CAN data point.

Name	CAN ID [hex]	Type of CAN	Send Period max [ms]	Send Period min [ms]	DLC	Status	CAN ID Mark [hex]
CAN_A_lighting	0x101	1	0	0	8	0	0
CAN_A_Drive	0x102	1	0	0	8	0	0
CAN_B_Accelerator	0x103	1	0	0	8	0	0
Monitor_C	0x104	1	100	100	8	1	0
CAN data points/Define CAN_DB							
Variable Name	**CAN Block Name**	**Bit Start**	**Bit Length**	**Data Format**			
Left_in	CAN_A_lighting	1	1	0			
Right_in	CAN_A_lighting	2	1	0			
acc_in	CAN_B_Accelerator	0	10	0			
forward_in	CAN_A_Drive	2	1	0			
reward_in	CAN_A_Drive	3	1	0			
ST_in	CAN_A_Drive	4	1	0			
FW	Monitor_C	0	1	0			
ST	Monitor_C	1	1	0			
RW	Monitor_C	2	1	0			
Acc_in	Monitor_C	3	10	0			
Acc_out	Monitor_C	13	10	0			

4. Discussion

From Figure 8, the operational principles of the CAN BUS communication control system for ATV electric vehicles begin with switches that send operational data, such as engine switch, headlight, turn signal, emergency light, horn, forward/reverse gear, and accelerator to the MRS Developers Studio software. This software is used to simulate and test the system's functionality using a method similar to the SAE J1939 standard [40]. This initial step is highly challenging, requiring precise configuration and data processing to ensure accurate simulation. Following this, MRS Developers Studio uses the gathered data to program the device controllers, ensuring compliance with automotive communication standards. This configuration is then tested using a CAN BUS test box, which represents another significant challenge. The test box ensures that communication between all devices operates correctly and efficiently, demanding rigorous verification and troubleshooting. Once the testing phase is successfully completed, the configured data are uploaded to the ATV electric vehicles. This final step is critical and challenging, as it must ensure that all systems within the vehicle operate seamlessly and reliably under real-world conditions. Presenting this development serves as a prototype for future studies and advancements. The development of the CAN BUS communication control system [41–46] for ATVs is not merely about testing and configuration; it sets the foundation for future educational and developmental pursuits. Tackling this challenge requires expertise and meticulous examination at every stage to ensure that the internal communication system of the vehicle performs with the utmost efficiency and reliability. By navigating these complex challenges, this development not only advances the current state of ATV electric vehicle technology but also lays down the groundwork for future innovations, driving the field forward with an enhanced understanding and improved technical capabilities.

The configuration of the CAN BUS communication control system is paramount to ensuring efficient and reliable communication, particularly in the development of CAN BUS

technology for ATV electric vehicles. This paper presents a method of development and configuration that serves as a prototype for future studies and advancements, leveraging the powerful MRS Developers Studio software to simulate and test the system comprehensively.

4.1. Assigning Controller Addresses

Each CAN controller must be meticulously named and assigned a CAN ID in hexadecimal format (e.g., 0×101, 0×102, 0×103, 0×104, 0×105, ...). This unique identifier is crucial for distinguishing between different controllers within the ATV's network.

4.2. Choosing the Data Transmission Standard

Selecting the appropriate data transmission standard is critical. A value of zero utilizes the Standard Frame or Classic CAN with 11-bit identifiers, while a value of one employs the Extended Frame with 29-bit identifiers. For ATV electric vehicles, the Extended Frame (29-bit identifiers) is preferred due to its ability to handle a greater volume of data, essential for complex and real-time processing requirements.

4.3. Setting Data Transmission Speed

Transmission speed is specified in milliseconds and applies solely to data transmission. For data reception, the speed is set to zero. As shown in Figure 6, a value of zero is assigned for data reception and a value of one for data transmission, ensuring efficient data flow within the system. Appropriate speed settings are vital for the high-speed data exchange necessary in ATV systems.

4.4. Configuring Data Points

Each variable in the communication control program must have its bit start and bit length precisely specified. This precise alignment ensures that the communication control settings accurately match the data addresses, facilitating precise and effective communication control. This precision is critical for the efficient operation of various systems within an ATV electric vehicle, such as motor control, battery management, and safety systems.

From Table 2 to Table 4, the settings of the parameters are as follows: bit start refers to the sequence of communication control, for example, sequence 1 controls engine start, sequence 2 controls headlight operation, and so on. Bit length indicates the length of the control signal. Simply put, for digital signals, there are only two possible values, zero or one, resulting in a bit length of one However, for analog signals, the values are not restricted to just 0 or 1. Instead, they can range from 0 to 255 (2^8), 0 to 1023 (2^{10}), or even higher values. Which is related to the length of the data.

4.5. Utilizing MRS Developers Studio

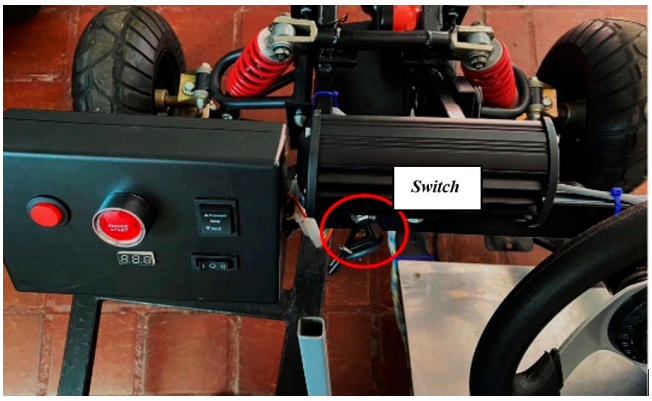
MRS Developers Studio software is an indispensable tool for simulating and testing the CAN BUS system. It allows developers to scrutinize and refine settings meticulously, ensuring optimal performance. This software aids in analyzing data and system efficiency, enabling the rapid identification and resolution of issues.

The detailed structure of the communication control system model for an ATV electric vehicle is presented in Table 5.

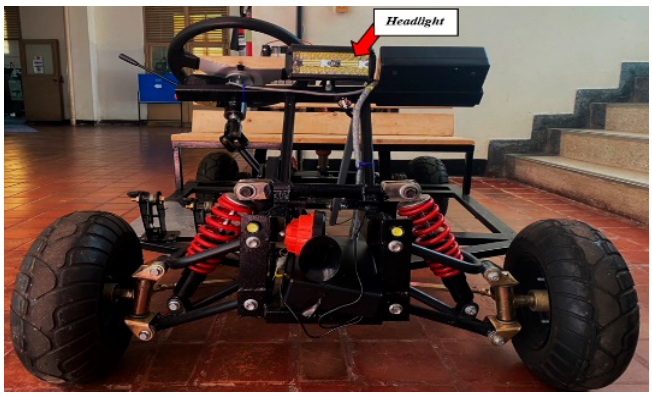
Based on Table 5, if the PCAN cable is connected between the CAN-High and CAN-Low lines to a computer using the PCAN-View software as shown in Figure 12, the principles of communication development using CAN BUS and MRS Developers Studio software with CAN IDs ranging from 0×101 to 0×105 can be described. When the PCAN cable is properly connected and configured, the PCAN-View software will display the CAN messages being transmitted on the network. Each message is identified by its unique CAN ID. In this case, the CAN IDs 0×101 to 0×105 are used to differentiate between different types of messages or devices on the CAN network.

Table 5. The key components of this model.

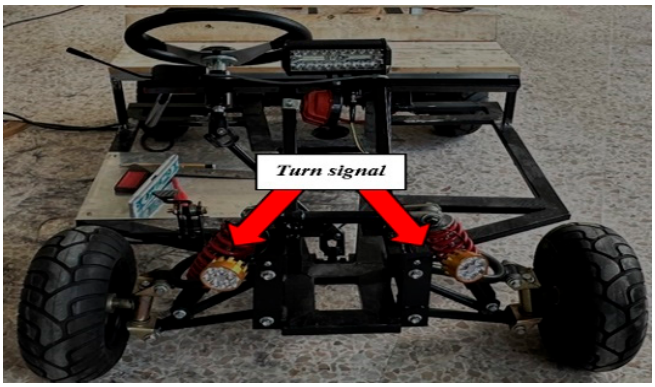
-Simulation diagram of ATV: Switches



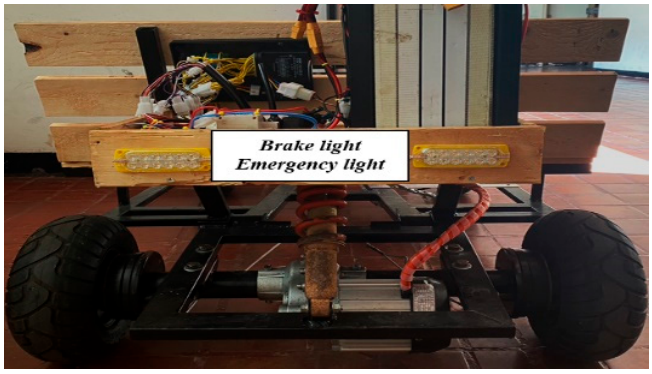
-Simulation diagram of ATV: Headlights



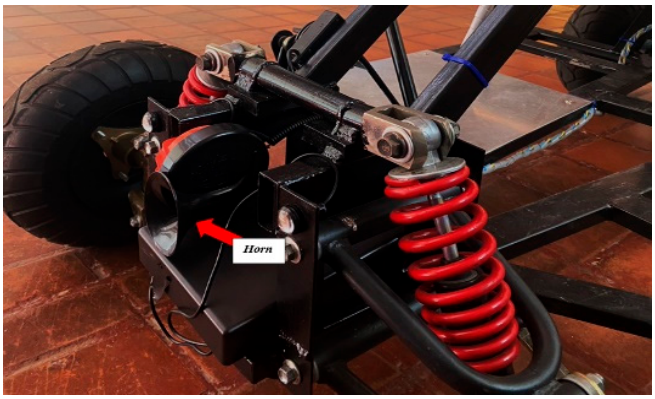
-Simulation diagram of ATV: Turn signal



-Simulation diagram of ATV: Emergency light



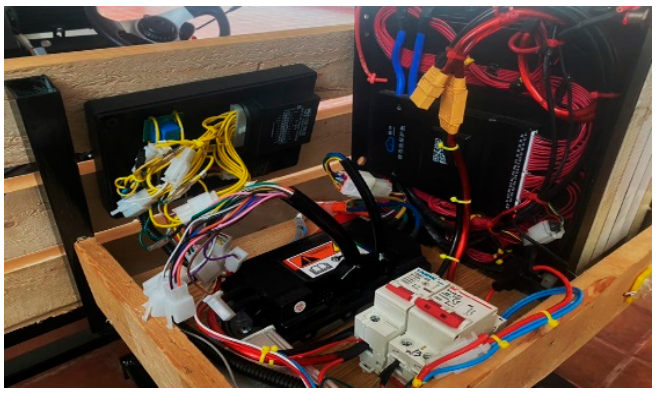
-Simulation diagram of car: Horn



-Simulation diagram of car: Forward/reverse gear

-Simulation diagram of car: Electric accelerator

-Controller box/battery pack/wiring



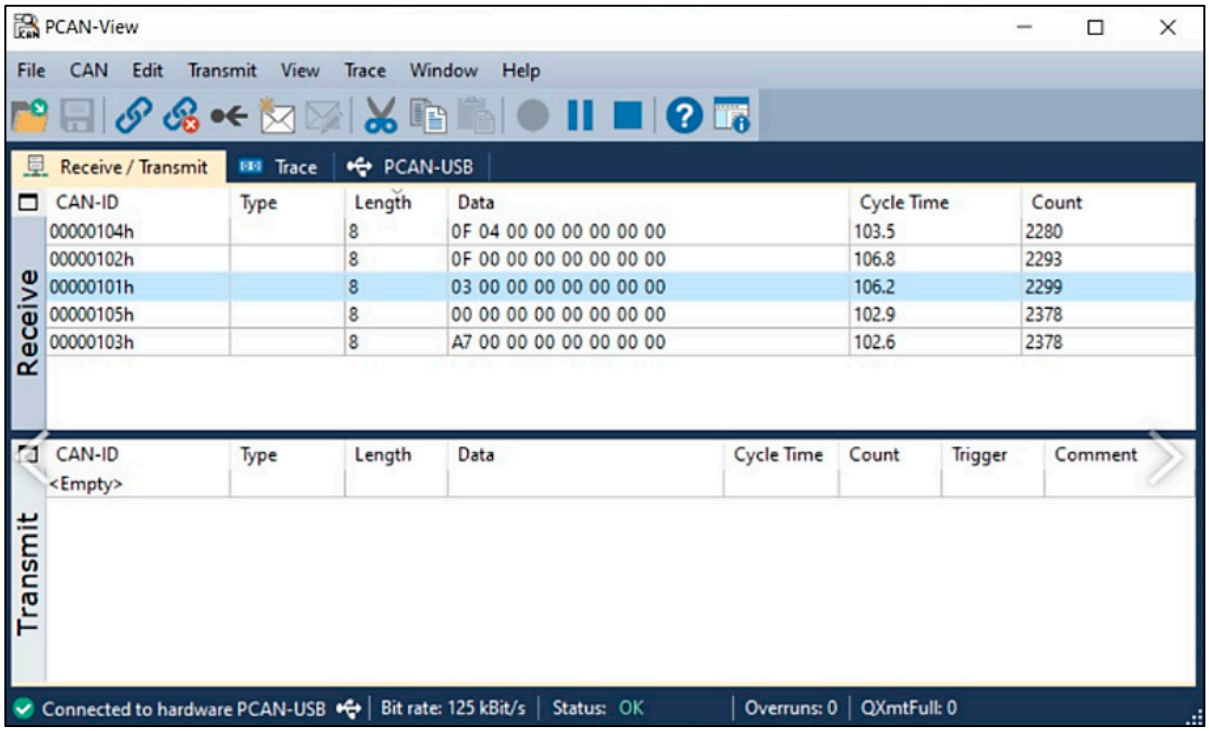


Figure 12. PCAN-View (connected between the CAN-High and CAN-Low lines).

4.6. Upper Section—Receive/Transmit

Receive/Transmit Tab: This tab is selected, indicating the user is viewing both received and transmitted CAN messages.

CAN-USB: The hardware interface being used is a PCAN-USB adapter.

4.7. Columns

CAN-ID: Identifies the CAN message ID. The IDs listed are hexadecimal values (e.g., 00000101h, 00000102h, 00000103h, 00000104h, 00000105h).

Type: The type of CAN message. In Figure 12, all entries are standard CAN messages (no specific type is listed, implying standard messages).

Length: The length of the data in bytes. All messages have a length of 8 bytes.

Data: These data fields contain the necessary information for communication and controlling various devices within the CAN network.

Cycle Time: The time interval between each message in milliseconds. For instance, the message with CAN-ID 00000102h has a cycle time of 106.8 ms.

Count: The number of times the message has been received. For example, the message with CAN-ID 00000105h has been received 2378 times.

4.8. Status Bar

Connection Status: Indicates that the software is connected to the PCAN-USB hardware.

Bit rate: Shows the current bit rate, which is 125 kbit/s.

Status: Shows "OK," indicating that the connection is functioning properly.

The PCAN-View software allows for monitoring and analyzing the CAN BUS communication by providing real-time data on the messages being transmitted. This is crucial for development and troubleshooting within the CAN network, ensuring that messages are being sent and received correctly and that the timing and data integrity are maintained.

5. Conclusions

The development of the CAN BUS communication control system for ATV electric vehicles is a highly challenging endeavor that sets a foundation for future educational

and technological advancements. By navigating these complexities, the project not only enhances the current state of ATV electric vehicle technology but also establishes a robust prototype for future studies. The meticulous approach in assigning controller addresses, selecting the appropriate data transmission standard, configuring transmission speeds, and utilizing MRS Developers Studio for simulation and testing, as well as integrating CAN-High and CAN-Low for differential signaling, all contribute to creating a highly efficient and reliable communication system. This development paves the way for further innovations, driving the field forward with improved understanding and technical capabilities.

Author Contributions: Conceptualization, N.D. and U.L.; methodology, N.D. and S.N.; software, N.D. and W.N.; validation, N.D., S.N. and U.L.; data curation, N.D. and W.N.; writing—original draft preparation, N.D.; writing—review and editing, N.D.; visualization, N.D. and S.N.; supervision, N.D. and U.L.; project administration, N.D. and S.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Suranaree University of Technology of Thailand under the Student Fundamental Research Funds for the Universities. Grant number IRD-thesis-1.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: We would like to thank the Suranaree University of Technology, Faculty of Engineering for their assistance in terms of budget and equipment for the thesis, as well as advice and close monitoring of the paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Adly, S.; Moro, A.; Hammad, S.; Maged, S.A. Prevention of Controller Area Network (CAN) Attacks on Electric Autonomous Vehicles. *Appl. Sci.* **2023**, *13*, 9374. [CrossRef]
2. Yin, L.; Xu, J.; Wang, Z.; Wang, C. A Provable Secure Session Key Distribution Protocol Based on NSSK for In-Vehicle CAN Network. *Mathematics* **2022**, *10*, 2903. [CrossRef]
3. Pham, N.N.; Leuchter, J.; Pham, K.L.; Dong, Q.H. Battery Management System for Unmanned Electric Vehicles with CAN BUS and Internet of Things. *Vehicles* **2022**, *4*, 639–662. [CrossRef]
4. Babangida, A.; Light Odazie, C.M.; Szemes, P.T. Optimal Control Design and Online Controller-Area-Network Bus Data Analysis for a Light Commercial Hybrid Electric Vehicle. *Mathematics* **2023**, *11*, 3436. [CrossRef]
5. Prasad, B.V.; Gao, R.; Jing-Jou, T.; Jhen, H.C. LIN Bus Based Touchpad System for Smart Vehicle Cabin. In Proceedings of the 2022 8th International Conference on Applied System Innovation (ICASI), Nantou, Taiwan, 22–23 April 2022; pp. 54–57. [CrossRef]
6. Ling, B.; Peng, F.; Li, A. The Car Body Control Bus Design Based on CAN/LIN Bus. In Proceedings of the 2011 International Conference on Computational and Information Sciences, Chengdu, China, 21–23 October 2011; pp. 885–888. [CrossRef]
7. Wang, Y.; Liu, H.; Huang, B.; Sun, X. Frame design for vehicular flex ray network based on transmission reliability. In Proceedings of the 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 10–11 December 2017; pp. 850–855. [CrossRef]
8. Khanapurkar, M.; Bajaj, P.; Gharode, D. A design approach for Intelligent Vehicle Black Box System with intra-vehicular communication using LIN/Flex-ray protocols. In Proceedings of the IEEE International Conference on Industrial Technology, Chengdu, China, 21–24 April 2008; pp. 1–6. [CrossRef]
9. Lee, M.Y.; Chung, S.M.; Jin, H.W. Automotive network gateway to control electronic units through MOST network. In Proceedings of the 2010 Digest of Technical Papers International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 9–13 January 2010; pp. 309–310. [CrossRef]
10. Baek, S.H.; Jeong, D.W.; Park, Y.S.; Kim, H.S.; Kim, M.J.; Jang, J.W. Implementation Vehicle Driving State System with OBD-II, MOST network. In Proceedings of the 17th Asia Pacific Conference on Communications, Sabah, Malaysian, 2–5 October 2011; pp. 709–714. [CrossRef]
11. Lu, S.; Fang, Z.; Qu, G.; Gao, S. ETBAC-Based Model in Media Oriented System Transport Network. In Proceedings of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, Liverpool, UK, 26–28 October 2015; pp. 504–511. [CrossRef]
12. Kenjić, D.; Antić, M.; Bjelica, M. Evaluation of Ethernet Subsystem for Domain Controller in Autonomous Vehicles. In Proceedings of the 2021 Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 26–27 May 2021; pp. 59–63. [CrossRef]

13. Hank, P.; Müller, S.; Vermesan, O.; Van Den Keybus, J. Automotive Ethernet: In-vehicle networking and smart mobility. In Proceedings of the 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 18–22 March 2013; pp. 1735–1739. [CrossRef]
14. Zeinali, M.; Erdogan, N.; Bayram, I.S.; Thompson, J.S. Impact of Communication System Characteristics on Electric Vehicle Grid Integration: A Large-Scale Practical Assessment of the UK's Cellular Network for the Internet of Energy. *Electricity* **2023**, *4*, 309–319. [CrossRef]
15. Kavas-Torris, O.; Gelbal, S.Y.; Cantas, M.R.; Aksun Guvenc, B.; Guvenc, L. V2X Communication between Connected and Automated Vehicles (CAVs) and Unmanned Aerial Vehicles (UAVs). *Sensors* **2022**, *22*, 8941. [CrossRef]
16. Boubaker, S.; Alsubaei, F.S.; Said, Y.; Ahmed, H.E. Lightweight Cryptography for Connected Vehicles Communication Security on Edge Devices. *Electronics* **2023**, *12*, 4090. [CrossRef]
17. *J1939DA_202406*; SAE International Supplement, J1939 Digital Annex. SAE Standard: Warrendale, PA, USA, 2024. [CrossRef]
18. Alzahrani, A.; Wangikar, S.M.; Indragandhi, V.; Singh, R.R.; Subramaniyaswamy, V. Design and Implementation of SAE J1939 and Modbus Communication Protocols for Electric Vehicle. *Machines* **2023**, *11*, 201. [CrossRef]
19. Zeltwanger, H. *CAN Standard Review: Changes and Enhancements of the ISO 11898*; SAE Technical Paper 2000-01-0143; SAE Standard: Warrendale, PA, USA, 2000. [CrossRef]
20. *J2889/1_201511*; SAE International Recommended Practice, Measurement of Minimum Noise Emitted by Road Vehicles. SAE Standard: Warrendale, PA, USA, 2011. [CrossRef]
21. Pan, J.; Xie, L.; Jin, X. Realization of CAN FD Shielding in High Speed CAN Transceiver for Partial Networking. In Proceedings of the 2021 9th International Symposium on Next Generation Electronics (ISNE), Changsha, China, 9–11 July 2021; pp. 1–4. [CrossRef]
22. Rattanachan, S.; Nuchkum, S.; Leeton, U. Development of Electric Door Control System using MRS Developers Studio for Low-floor Bus. In Proceedings of the 9th International Electrical Engineering Congress (iEECON), Pattaya, Thailand, 10–12 March 2021; pp. 285–288. [CrossRef]
23. Osman, R.A. Optimizing Autonomous Vehicle Communication through an Adaptive Vehicle-to-Everything (AV2X) Model: A Distributed Deep Learning Approach. *Electronics* **2023**, *12*, 4023. [CrossRef]
24. Ngo, H.; Fang, H.; Wang, H. Cooperative Perception With V2V Communication for Autonomous Vehicles. *IEEE Trans. Veh. Technol.* **2023**, *72*, 11122–11131. [CrossRef]
25. Ahangar, M.N.; Ahmed, Q.Z.; Khan, F.A.; Hafeez, M. A Survey of Autonomous Vehicles: Enabling Communication Technologies and Challenges. *Sensors* **2021**, *21*, 706. [CrossRef] [PubMed]
26. Zhang, L. Cooperative adaptive cruise control in mixed traffic with selective use of vehicle-to-vehicle communication. *IET Intell. Transp. Syst.* **2018**, *12*, 1243–1254. [CrossRef]
27. Abuhdima, E.; Liu, J.; Zhao, C.; Elqaouaq, A.; Comert, G.; Huang, C.T.; Pisu, P.; Nazeri, A.H. Impact of Dust and Sand on 5G Communications for Connected Vehicles Applications. *IEEE J. Radio Freq. Identif.* **2022**, *6*, 229–239. [CrossRef]
28. Mensi, N.; Rawat, D.B. On the Performance of Partial RIS Selection vs. Partial Relay Selection for Vehicular Communications. *IEEE Trans. Veh. Technol.* **2022**, *71*, 9475–9489. [CrossRef]
29. Vinel, A.; Lyamin, N.; Isachenkov, P. Modeling of V2V Communications for C-ITS Safety Applications: A CPS Perspective. *IEEE Commun. Lett.* **2018**, *22*, 1600–1603. [CrossRef]
30. Deng, Y.; Lu, K.; Liu, T.; Wang, X.; Shen, H.; Gong, J. Numerical Simulation of Aerodynamic Characteristics of Electric Vehicles with Battery Packs Mounted on Chassis. *World Electr. Veh. J.* **2023**, *14*, 216. [CrossRef]
31. Pražanová, A.; Knap, V.; Stroe, D.-I. Literature Review, Recycling of Lithium-Ion Batteries from Electric Vehicles, Part II: Environmental and Economic Perspective. *Energies* **2022**, *15*, 7356. [CrossRef]
32. Veza, I.; Abas, M.A.; Djamari, D.W.; Tamaldin, N.; Endrasari, F.; Budiman, B.A.; Idris, M.; Opia, A.C.; Juangsa, F.B.; Aziz, M. Electric Vehicles in Malaysia and Indonesia: Opportunities and Challenges. *Energies* **2022**, *15*, 2564. [CrossRef]
33. Wu, X.; Subramanian, S.; Guha, R.; White, R.G.; Li, J.; Lu, K.W.; Bucceri, A.; Zhang, T. Vehicular Communications Using DSRC: Challenges, Enhancements, and Evolution. *IEEE J. Sel. Areas Commun.* **2013**, *31*, 399–408. [CrossRef]
34. Yang, L.; Li, H. Vehicle-to-vehicle communication based on a peer-to-peer network with graph theory and consensus algorithm. *IET Intell. Transp. Syst.* **2019**, *13*, 280–285. [CrossRef]
35. Yang, Y.; Fei, D.; Dang, S. Inter-vehicle cooperation channel estimation for IEEE 802.11p V2I communications. *J. Commun. Netw.* **2017**, *19*, 227–238. [CrossRef]
36. Cheng, X.; Huang, Z.; Chen, S. Vehicular communication channel measurement, modelling, and application for beyond 5G and 6G. *IET Commun.* **2020**, *14*, 3303–3311. [CrossRef]
37. Osman, R.A.; Zaki, A.I.; Abdelsalam, A.K. Novel Road Traffic Management Strategy for Rapid Clarification of the Emergency Vehicle Route Based on V2V Communications. *Sensors* **2021**, *21*, 5120. [CrossRef] [PubMed]
38. Rathore, R.S.; Hewage, C.; Kaiwartya, O.; Lloret, J. In-Vehicle Communication Cyber Security: Challenges and Solutions. *Sensors* **2022**, *22*, 6679. [CrossRef] [PubMed]
39. Balkus, S.V.; Wang, H.; Cornet, B.D.; Mahabal, C.; Ngo, H.; Fang, H. A Survey of Collaborative Machine Learning Using 5G Vehicular Communications. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 1280–1303. [CrossRef]
40. *J1939/16_201811*; SAE International Recommended Practice, Automatic Baud Rate Detection Process. SAE Standard: Warrendale, PA, USA, 2018. [CrossRef]

41. Osman, R.A.; Peng, X.H.; Omar, M.A. Adaptive cooperative communications for enhancing QoS in vehicular networks. *Phys. Commun.* **2019**, *34*, 285–294. [CrossRef]
42. Osman, R.A.; Saleh, S.N.; Saleh, Y.N.M.; Elagamy, M.N. Enhancing the Reliability of Communication between Vehicle and Everything (V2X) Based on Deep Learning for Providing Efficient Road Traffic Information. *Appl. Sci.* **2021**, *11*, 11382. [CrossRef]
43. Trien, L.T.; Adachi, K.; Yamao, Y. Packet relay-assisted V2V communication with sectorised relay station employing payload combining scheme. *IET Commun.* **2018**, *12*, 458–465. [CrossRef]
44. Feteiha, M.F.; Ahmed, M.H. Multihop Best-Relay Selection for Vehicular Communication over Highways Traffic. *IEEE Trans. Veh. Technol.* **2018**, *67*, 9845–9855. [CrossRef]
45. Wang, S.; Wang, D.; Li, C.; Xu, W. Full Duplex AF and DF Relaying Under Channel Estimation Errors for V2V Communications. *IEEE Access* **2018**, *6*, 65321–65332. [CrossRef]
46. Wei, L.; Hu, R.Q.; Qian, Y.; Wu, G. Energy Efficiency and Spectrum Efficiency of Multihop Device-to-Device Communications Underlaying Cellular Networks. *IEEE Trans. Veh. Technol.* **2016**, *65*, 367–380. [CrossRef]