IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

CAN DATABASE DEVELOPMENT AND GEAR SHIFT AUTOMATION

Dr.K.KAMALA DEVI*1, N.ALEKYA *2, Dr.B.VIJAYAKRISHNA *3, M.SRINIVAS *4, K.JAGADISH NAIK *5

*1,*3Assistant Professor, Electrical and Electronics Engineering, Bapatla Engineering College, Andhra Pradesh, India

*2,,4,5, Students, Electrical and Electronics Engineering Department, Bapatla Engineering College, Bapatla, Andhra Pradesh, India.

Abstract - This paper proposes leveraging Controller Area Network (CAN) databases for efficient gear shift automation in modern automotive systems. By integrating CAN databases with advanced control algorithms, real-time monitoring and adaptive gear shift operations are enabled based on vehicle parameters and environmental conditions. Key components include CAN database modeling, data acquisition, preprocessing, and adaptive control strategies. The system enhances reliability, scalability, and interoperability, facilitating seamless integration with existing vehicle architectures.

Keywords – Control Area Network, CAN database, Gear shift Automation, Automotive design, CAN bus, Electronic Control Units (ECUs).

I. Introduction

Gear shift automation using Controller Area Network (CAN) database involves leveraging the communication protocol of CAN to automate the shifting process in vehicles. CAN is a robust and widely used communication protocol in modern vehicles, allowing various electronic control units (ECUs) to communicate with each other efficiently.

Controller Area Network (CAN): CAN is a standard communication protocol used in vehicles to enable microcontrollers and devices to communicate with each other in real-time without a host computer. It allows for high-speed transmission of data among various ECUs, including those responsible for engine control, transmission, braking, and more.

Automation of Gear Shift: Traditional manual transmissions require the driver to manually engage gears by operating the clutch and shifting gears using a gear stick. With gear shift automation, this process is automated, either partially or fully, by electronic control systems.

CAN Database (CANdb): A CAN database contains information about the messages, signals, and parameters exchanged on a CAN bus. It defines the structure and content of the messages transmitted between ECUs. This database is crucial for developing applications that interact with the vehicle's CAN network.

Integration of CAN with Gear Shifting Systems: By integrating CAN with the gear shifting system, relevant information such as vehicle speed, engine RPM, throttle position, and other parameters can be accessed in real-time. This data is essential for determining the optimal timing and strategy for gear shifts.

Transmission Control Module (TCM): In modern vehicles, the Transmission Control Module is responsible for controlling the operation of automatic transmissions or automated manual transmissions. It receives input from various sensors via the CAN bus and uses this data to make decisions regarding gear selection, shift timing, and torque management.

Advantages of Gear Shift Automation using CAN: Automated gear shifting can optimize engine performance and reduce fuel consumption. Smooth and precise gear shifts improve comfort and drivability. Automated shifting can minimize mechanical stress on transmission components, leading to longer component lifespan. CAN-based systems can adapt gear shift patterns based on driving conditions, such as road gradient, vehicle load, and driver behavior.

II. SOFTWARES *USED*

The softwares used in developing CAN database and Gear shift automation are :

***1.Vector CANdb++* :** Vector CANdb++ is a tool primarily used for creating, editing, and managing CAN databases (CANdb) in automotive and embedded systems development. Here are some brief points about it:

CAN Database Management: It provides a platform for creating and managing Controller Area Network (CAN) databases, which are used for communication between electronic control units (ECUs) in vehicles.

User-Friendly Interface: It typically offers a user-friendly interface that allows engineers to define message structures, signals, and other parameters related to the CAN network.

Signal Definition: Engineers can define the signals within messages, specifying parameters such as signal name, start bit, length, scaling, unit, and value range.

Message Definition: It enables the definition of message properties like message ID, DLC (Data Length Code), transmission mode, and cyclic transmission.

Support for Various Formats: CANdb++ often supports various file formats for import and export, facilitating compatibility with other tools and systems used in the development process.

Integration with Development Tools: It may integrate with other development tools commonly used in automotive embedded systems development, such as CANoe or CANalyzer, for simulation and testing purposes.

Version Control: Some versions of CANdb++ offer version control features, allowing multiple users to collaborate on the same database while maintaining a history of changes.

Validation and Verification: It may include features for validating and verifying the CAN database to ensure its correctness and consistency.

Documentation: It often provides options for generating documentation from the CAN database, which can be useful for reference and compliance purposes.

Customization: Depending on the version and configuration, it might offer customization options to tailor the tool to specific project requirements or industry standards.

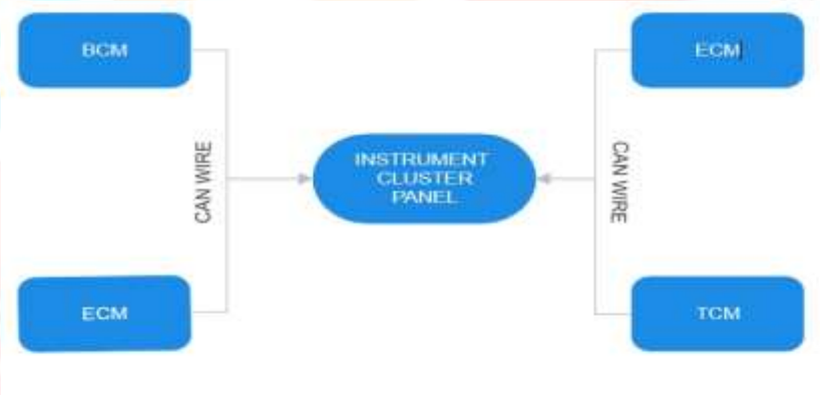
2. TS master : It is a versatile virtual instrument software designed to facilitate seamless connectivity, configuration, and management of TOSUN's hardware tools. Its functionalities span automotive bus monitoring, simulation, diagnostics, calibration, BootLoader, I/O control, measurement testing, and EOL. Additionally, TSMaster is equipped to handle multiple protocols including LIN, CAN, CAN FD, FlexRay, and Ethernet, ensuring robust support for various applications.

- Searching online for the official website or documentation of the software.
- Contacting the manufacturer or developer of the software for information.
- Checking with colleagues or peers who may have experience using the software.

III. CAN DATABASE

Store historical data regarding seat belt usage in a database. This data can include timestamps, seat occupancy status, seat belt buckle status, and any relevant vehicle information. By logging this information, you can track trends over time, identify patterns of non-compliance, and generate reports for analysis. Allow users to customize seat belt reminder settings, such as the frequency or intensity of reminders, through a user interface. Store these preferences in a database to ensure they persist across vehicle sessions. This allows occupants to tailor the seat belt monitoring system to their preferences for a more personalized experience. Store diagnostic information related to the seat belt monitoring system in a database. This can include error codes, system status, and troubleshooting logs. By centralizing diagnostic data, you can facilitate maintenance and troubleshooting activities, enabling technicians to quickly identify and address issues with the system. Integrate the database with other vehicle systems, such as the onboard diagnostics (OBD) system or the vehicle's telematics unit. This allows for seamless exchange of data between different vehicle components and enables advanced features such as remote diagnostics or predictive maintenance based on seat belt usage patterns. Implement security measures to protect sensitive data stored in the database, such as encryption, access controls, and audit logs. Ensure that only authorized users or systems have access to the data to maintain privacy and compliance with data Utilize the database for real-time data processing and analysis to support dynamic decisionmaking within the seat belt monitoring system. For example, you can use stored data to adjust reminder settings based on individual driving behavior or to trigger alerts for immediate action in case of non-compliance. Design the database architecture with scalability and performance in mind to accommodate the potential growth of data volume and system usage over time. Consider factors such as data indexing, partitioning, and caching to optimize database performance.

IV. PROCESS FLOW



COMMUNICATION BETWEEN ECU'S AND CLUSTER

The following are the ECUs used in database :

1. **Anti-lock Braking System (ABS)**: This is a safety system in vehicles designed to prevent wheels from locking up during braking, thereby maintaining tractive contact with the road surface. ABS helps the driver maintain steering control during hard braking, especially in slippery conditions.
2. **Engine Control Module(ECM) :** This is a type of electronic control unit (ECU) specifically responsible for managing the engine's operation. It monitors various engine sensors and adjusts fuel delivery, ignition timing, and other parameters to optimize engine performance, fuel efficiency, and emissions.
3. **Body Control Module (BCM)**: This is a control unit in a vehicle responsible for managing various electrical systems and functions related to the body of the vehicle. The BCM acts as a centralized hub, receiving input from switches, sensors, and other control modules throughout the vehicle's body, and then controlling functions such as lighting, power windows, door locks, climate control, and more. Essentially, the BCM helps to streamline and coordinate the operation of various electrical components within the vehicle's body.
4. **Transmission Control Module (TCM)**: This is an electronic control unit (ECU) specifically dedicated to managing the automatic transmission system in a vehicle. The TCM receives input from various sensors, including vehicle speed sensors, throttle position sensors, and engine speed sensors. Based on this input, the TCM controls the operation of the transmission, including gear selection, shifting points, and torque converter lockup. It ensures smooth gear changes, optimal performance, and fuel efficiency of the transmission system.

V. CAPL SCRIPTING

CAPL (Communication Access Programming Language) is a scripting language used in the development and testing of embedded systems, particularly in the automotive industry for Controller Area Network (CAN) bus systems. CAPL scripts are typically written and executed within tools like Vector CANoe or CANalyzer, which are widely used for CAN network analysis, simulation, and testing.

Here's an overview of CAPL scripting:

1. Purpose: CAPL scripting is used for simulating and testing the behavior of ECUs (Electronic Control Units) andother devices communicating over the CAN bus. It allows developers to create custom test scenarios, manipulate messages on the CAN bus, and simulate various conditions for testing purposes.

2. Features:

- Message Manipulation : CAPL allows you to create, send, receive, and modify messages on the CAN bus. This includes changing message data, identifiers, cycle times, and signal values.

- Event-Driven Programming : CAPL scripts are event-driven, meaning they respond to specific events or triggers such as receiving a message, timer expiration, or user input.

- Variables and Functions : CAPL supports variables, arrays, structures, and functions, allowing for complex script logic and data manipulation.

- Access to System Functions : CAPL provides access to built-in functions for tasks such as reading system time, generating random numbers, and performing mathematical operations.

- Simulation Control : CAPL scripts can control the simulation environment, including starting and stopping simulations, configuring simulation parameters, and logging data.

3. Common Use Cases :

- Simulation of ECU Behavior : CAPL scripts can simulate the behavior of ECUs by sending and receiving CAN messages according to predefined scenarios.

- Diagnostic Testing : CAPL scripts can be used to simulate diagnostic messages and test the behavior of diagnostic functions in ECUs.

- Fault Injection : CAPL scripts can inject simulated faults or error conditions into the CAN bus to test the response of ECUs and other devices.

- Automation of Test Procedures : CAPL scripts can automate test procedures by defining sequences of actions and responses on the CAN bus.

4. Syntax and Structure :

- CAPL scripts are typically organized into functions, each responding to specific events or triggers.

- Scripts can contain variable declarations, function definitions, event handlers, and control flow statements.

- CAPL syntax is C-like, making it familiar to programmers with experience in C or similar languages.

5. Integration with Tools : CAPL scripts are commonly used with tools like Vector CANoe or CANalyzer, which provide a graphical user interface for creating and running simulations, as well as monitoring CAN bus activity.

Overall, CAPL scripting is a powerful tool for developing and testing CAN-based systems, offering flexibility and control over simulation scenarios and test conditions. It's an essential skill for engineers working in automotive embedded systems development and validation

VI. GEAR SHIFT AUTOMATION

Monitoring the status of gear shift automation involves observing various parameters and signals to ensure that the system is functioning correctly and providing the expected performance. Here are some key aspects of status monitoring in gear shift automation:

1.Gear Position : Monitoring the current gear position is fundamental. Sensors or algorithms can determine which gear the transmission is currently in. Anomalies such as incorrect gear engagement or failure to shift into the desired gear can indicate issues with the automation system.

2. Vehicle Speed : Gear shifting strategies often depend on vehicle speed. Monitoring vehicle speed ensures that gear shifts occur at appropriate speeds to optimize performance, fuel efficiency, and driver comfort. Abnormalities in speed readings or discrepancies between speed and gear selection could indicate problems.

3.Engine RPM : Engine speed is critical for determining the optimal timing of gear shifts. Monitoring engine RPM helps ensure smooth and timely shifts, preventing over-revving or stalling. Deviations from expected RPM ranges during shifts could indicate transmission or engine issues.

4.Throttle Position : Throttle position is another factor considered in gear shifting strategies. It affects engine load and torque output, influencing the timing and aggressiveness of gear shifts. Monitoring throttle position helps ensure that gear shifts are appropriately matched to driver inputs and driving conditions.

5. Clutch Status (if applicable) : In vehicles with automated manual transmissions or dual-clutch transmissions, monitoring clutch engagement and disengagement is crucial. Proper clutch operation is essential for smooth gear shifts and preventing premature wear on transmission components.

6.CAN Bus Communication: Monitoring communication on the CAN bus is vital for detecting errors or malfunctions in the gear shift automation system. Lost or corrupted messages, bus-off conditions, or abnormal bus load can indicate issues that require troubleshooting and corrective action.

8.System Diagnostics : Implementing built-in diagnostic routines can help identify potential issues with sensors, actuators, or control algorithms. Diagnostic trouble codes (DTCs) or fault flags can alert drivers or service technicians to specific problems requiring attention.

SI.NO	ECU'S	MESSAGES	SIGNAL
1	BCM	BCM	IGNITION
2	ECM	ECM	THROTTLE POSITION
3	TCM	TCM	VEHICLE SPEED
4	ECM	ECM	ENGINE RPM

Fig 2 : ECUs used in the database

VII. GEAR SHIFT AUTOMATION

Fig 3 shows the design of the gear shift automation with integration of all signals, each assigned specific addresses obtained from the database. The below panel consists of gauges for speed, RPM, and throttle position, alongside ignition switch. Additionally, LEDs with different colors such as red, green, and yellow indicate various signals like economy, power and over speed. However, during the off condition of the simulation, the design does not illustrate movement in gauges or blinking in LEDs.

Fig 3 : Before simulation

Fig 4 shows the panel after simulation starts. Whenever the ignition switch and throttle position are ON the speed and RPM values in the panel changes. Also the economy, power and over speed indications will also glow according to the range of speed given in the database.

Fig 4 : after simulation

VIII. CONCLUSION

In brief, the convergence of database development with gear shift status automation promises transformative benefits for automotive industries and beyond. By integrating databases with automated gear shift status monitoring, businesses can enhance efficiency, streamline operations, and advance towards smarter, more connected transportation systems. This synergy not only optimizes vehicle performance but also lays the under ground for innovative solutions in safety, maintenance and autonomous driving technologies, ushering in a new era of intelligent mobility.

IX. REFERENCES

[1]. Surekha P. Gaikwad and Atmeshkumar S. Patel, "Automotive control system using CAN protocol", in International Journal of Advanced Research, Ideas and Innovations in Technology, 2019. ISSN: 2454-132X.

[2]. A. A. Salunkhe, Pravin P Kamble and Rohit Jadhav, "Design And Implementation of CAN bus Protocol for Monitoring Vehicle Parameters", in IEEE International Conference On Recent Trends In Electronics Information Communication Technology, May 20-21, 2016, India, ISBN: 978-1-5090-0774-5/16.

[3].S. N Chikhale,"Automobile Design and Implementation of CAN bus Protocol- A Review", in IJRDO - Journal of Electrical and Electronics Engineering, Volume- 4/Issue 1/January 2018, ISSN: 2456-6055.

[4]. Guilherme Marcon Zago and Edison Pignaton deFreitas, "A Quantitative Performance Study on CAN and CAN FD Vehicular Networks " in IEEE Transactions On Industrial Electronics,Vol-65, No.-5, May 2018,ISSN: 0278-0046.

[5]. Y. Tao, Y. Yue and P. Craig, " A hybrid approach to detection and recognition of dashboard information in real-time," 2017 4th International Conference on Systems and Informatics (ICSAI), Hangzhou, 2017, pp. 1141-1145

[6]. Y. Kwon, J Choi, J. Jeon, K. Kim and B. Jang, " Design of Automotive Digital Instrument Cluster Adjustable to Driver's Cognitive Characteristics , " 2019 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea (South), 2019, pp.461-463.