

Secure Multi-Node Automotive Monitoring and Control System Using CAN FD Communication Protocol

Sandhya Lagade¹

*¹Department of Electronics and Telecommunication Engineering,
TPCT's College of Engineering, Dharashiv, Maharashtra, India*

Abstract - Modern automobiles rely on multiple Electronic Control Units (ECUs) for monitoring, control, safety, and automation functions. Efficient and secure communication among these ECUs is essential for reliable vehicle operation. Conventional Controller Area Network (CAN) systems suffer from limitations such as restricted payload capacity, lack of built-in security, vulnerability to unauthorized node access, and limited scalability. This paper presents a Secure Multi-Node Automotive Monitoring and Control System using CAN with Flexible Data Rate (CANFD) communication protocol. The proposed system employs a distributed ECU architecture with four intelligent nodes: (1) Cabin Temperature & Cooling Fan Control, (2) Oil Monitoring (temperature and level), (3) Rain Detection & Smart Lighting, and (4) Security Monitoring (gas concentration and door status). A Master ECU supervises the entire system and implements Hash-based Message Authentication Code (HMAC) with SHA-1 algorithm for secure node authentication. The system is implemented using Arduino Uno controllers, MCP2518FD CANFD modules, and various sensors including DS18B20, MQ7, rain sensor, LDR, and magnetic door sensors. Experimental results demonstrate reliable CANFD communication, successful node authentication, real-time monitoring, intelligent automation, and enhanced security capabilities. The distributed architecture reduces computational load on the central controller while improving system scalability, reliability, and modularity for modern automotive applications.

Keywords: Automotive Communication, CAN FD, Distributed ECU, HMAC Authentication, Intelligent Monitoring, Secure Communication.

1. INTRODUCTION

Modern vehicles have evolved from conventional transportation systems into intelligent and interconnected platforms integrating multiple Electronic Control Units (ECUs) for various monitoring and control functionalities. These ECUs manage critical vehicle operations including engine management, safety systems, environmental monitoring, lighting control, and passenger security. The increasing complexity of automotive systems demands efficient, reliable, and secure communication protocols to enable real-time data exchange among distributed ECUs [1].

The Controller Area Network (CAN) protocol, introduced by Robert Bosch GmbH, has become the de facto standard for automotive communication due to its robustness, reliability, low

implementation cost, and real-time communication capability [2]. However, with the growing demand for intelligent automotive systems and connected vehicles, classical CAN networks face several significant limitations. These include restricted payload capacity (maximum 8 bytes per frame), limited data transmission speed (up to 1 Mbps), absence of built-in security mechanisms, and vulnerability to unauthorized node insertion and cyber-attacks [3].

To address these challenges, CAN with Flexible Data Rate (CANFD) has emerged as an enhanced version of classical CAN protocol. CANFD provides higher data transmission speeds (up to 8 Mbps in data phase), larger payload capacity (up to 64 bytes per frame), improved bandwidth utilization, and reduced latency, making it suitable for modern automotive applications involving large amounts of sensor data and complex control algorithms [4].

Despite these improvements in communication efficiency, automotive networks remain vulnerable to security threats. The lack of built-in authentication mechanisms in both classical CAN and CANFD protocols allows any unauthorized node connected to the bus to potentially transmit malicious messages, leading to serious safety risks. Several research studies have demonstrated security vulnerabilities in automotive communication systems, highlighting the critical need for secure node authentication and intrusion detection mechanisms [5][6].

This paper presents a comprehensive solution addressing both communication efficiency and security requirements through a Secure Multi-Node Automotive Monitoring and Control System using CANFD communication protocol with HMAC-based node authentication. The proposed system implements a distributed ECU architecture where multiple intelligent nodes perform dedicated monitoring and automation tasks while communicating securely through a CANFD network supervised by a Master ECU.

The main contributions of this work include:

- 1) Implementation of CANFD-based distributed automotive communication architecture
- 2) Development of HMAC-SHA1 based secure node authentication mechanism
- 3) Design and integration of four intelligent monitoring and control nodes
- 4) Real-time environmental sensing and automated control functionalities
- 5) Request-response based communication for reduced network overhead
- 6) Practical prototype demonstration using embedded hardware platform

The remainder of this paper is organized as follows: Section II reviews related work and identifies research gaps. Section III describes the proposed system architecture and methodology. Section IV presents implementation details. Section V discusses experimental results and performance analysis. Section VI concludes the paper and outlines future research directions.

2. RELATED WORK

CAN has been a standard automotive communication technology since its introduction by Robert Bosch GmbH [1]. It was developed to reduce wiring complexity and provide reliable communication among distributed automotive controllers. CAN-based distributed architectures improve modularity and reduce the processing load on a single central controller.

CAN FD enhances the original CAN protocol by supporting a flexible data rate and larger payload capacity [2]. The improved data phase enables faster communication, which is useful when several ECUs transmit sensor values, diagnostic data, or status information over the same bus.

Automotive network security has become increasingly important because vehicle communication buses may be exposed to message-injection, replay, spoofing, and unauthorized-device attacks. Koscher *et al.* demonstrated that modern automotive systems can be vulnerable when attackers obtain access to in-vehicle communication networks [3]. Since the original CAN protocol does not provide source authentication, additional security mechanisms are required.

Lightweight cryptographic approaches, including Hash-based Message Authentication Code (HMAC), have been proposed for embedded and automotive communication systems [4]. HMAC provides message-integrity and source-authentication capability when communicating nodes share a secret key. It can be implemented with lower computational overhead than many full encryption schemes.

Several standalone automotive automation systems have also been reported. Rain-sensing wiper systems use moisture or rain sensors to activate wipers automatically [5]. Similarly, automatic headlight systems use Light Dependent Resistors (LDRs) to identify low-light conditions and control vehicle lighting [6]. However, many such systems operate independently and do not include secure distributed communication between nodes.

The present work combines CAN FD communication, distributed sensor nodes, local automation, Master ECU supervision, and HMAC-based node authentication in a single prototype platform.

Parameter	Conventional Systems	Proposed System
Communication protocol	Classical CAN / standalone control	CAN FD
Data payload	Up to 8 bytes in classical CAN	Up to 64 bytes in CAN FD
Node architecture	Centralized or partially distributed	Fully distributed prototype
Node authentication	Generally absent	HMAC-based node verification
Cabin monitoring	Limited	Cabin temperature and cooling fan control
Oil monitoring	May be standalone	Integrated CAN FD node
Rain and lighting control	Often independent	Combined Node 3
Security monitoring	Limited	CO and door-status monitoring
Scalability	Restricted	Improved modular expansion

Table -1: Comparison of existing and proposed systems

3. PROPOSED SYSTEM ARCHITECTURE

The proposed system consists of one Master ECU and four distributed ECU nodes connected through a common CAN FD communication bus. Each ECU node is responsible for a specific monitoring or control function. The Master ECU supervises node access, requests data, receives status messages, displays important parameters, and performs authentication verification.

Each controller communicates with an MCP2518FD CAN FD module through the Serial Peripheral Interface (SPI).

The architecture uses request-response communication. The Master ECU periodically sends a request to a particular authenticated node. The requested node reads sensors, performs local decision-making, controls actuators if required, and transmits a response frame containing the processed data and status information.

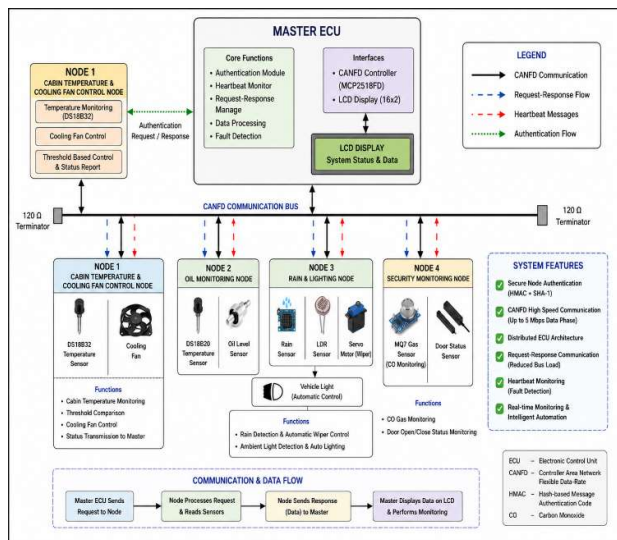


Fig -1: Proposed system architecture

A. Master ECU: CAN FD Supervisor and HMAC Authentication Controller

The Master ECU is responsible for the following functions:

- CAN FD network initialization
- Node registration and authentication
- HMAC verification
- Request-response communication management
- Node-status supervision
- Sensor-data reception and display
- Fault and missing-node indication
- LCD-based monitoring of system status

The Master ECU permits normal communication only after a node is authenticated successfully.

B. Node 1: Cabin Temperature and Cooling Fan Control

Node 1 monitors the cabin temperature using a DS18B20 digital temperature sensor. The measured temperature is compared with a predefined threshold. When the cabin temperature exceeds the threshold, the cooling fan is activated automatically through a driver circuit.

The node transmits the following data to the Master ECU:

- Cabin temperature
- Cooling fan status
- Temperature threshold status
- Node health status

C. Node 2: Oil Monitoring

Node 2 monitors oil temperature and oil level. A DS18B20 sensor is used to measure the oil temperature, while an analog level sensor is used to estimate the oil-level condition. To reduce fluctuations in analog measurements, multiple readings can be averaged before transmission.

The node transmits:

- Oil temperature
- Oil-level percentage or level status
- Warning condition, if applicable
- Node-status information

D. Node 3: Rain Detection and Smart Lighting

Node 3 combines rain sensing, automatic wiper control, ambient-light monitoring, and smart-lighting control. A rain sensor detects water or moisture on the windshield simulation surface. When the rain value exceeds the configured threshold, the node activates a servo motor to simulate automatic wiper movement.

An LDR sensor is used to detect surrounding light intensity. When low-light conditions are detected, the lighting output is activated automatically.

Node 3 transmits:

- Rain-detection status
- Wiper status
- Ambient-light level
- Vehicle-light status
- Node health status

E. Node 4: Security Monitoring

Node 4 monitors vehicle security and cabin safety. An MQ-7 gas sensor detects carbon-monoxide concentration changes, while a magnetic sensor detects whether the vehicle door is open or closed.

The node transmits:

- Carbon-monoxide sensor value
- Door open/close status
- Security warning condition
- Node health status

4. HARDWARE AND SOFTWARE IMPLEMENTATION

The proposed prototype is implemented using Arduino Uno boards as ECU controllers. Each node is connected to an MCP2518FD CAN FD controller module. The CAN FD module communicates with the Arduino Uno using SPI communication.

Table 2: Hardware components used

Component	Specification / Type	Function
Arduino Uno	ATmega328P	ECU controller
MCP2518FD	CAN FD controller	CAN FD communication
DS18B20	Digital temperature sensor	Cabin and oil temperature measurement
Analog oil-level sensor	Analog input sensor	Oil-level measurement
MQ-7 sensor	Carbon-monoxide sensor	Gas monitoring
Rain sensor	Analog/digital sensor	Rain detection
LDR sensor	Light-dependent resistor	Ambient-light sensing
Magnetic door sensor	Reed-switch sensor	Door-status sensing
Cooling fan	DC fan	Cabin cooling simulation
Servo motor	Rotary actuator	Wiper movement simulation

LCD display	16 × 2 display	Master ECU output display
-------------	----------------	---------------------------

A. CAN FD Communication

The MCP2518FD module supports CAN FD communication and provides an interface between the Arduino controller and the CAN FD bus. CAN FD improves communication capacity by allowing a larger payload and faster data transmission during the data phase.

The system uses message identifiers to distinguish between different node functions. The Master ECU sends a request frame, and the addressed node responds with sensor data and actuator status.

Table -3: CAN FD message structure

CAN ID	Message Type	Source	Description
0x00	Request frame	Master ECU	Request sensor/status data
0x01	Cabin monitoring data	Node 1	Cabin temperature and fan status
0x03	Oil monitoring data	Node 2	Oil temperature and level
0x05	Rain and lighting data	Node 3	Rain, wiper, light status
0x07	Security data	Node 4	CO and door status
0x10	Authentication request	ECU node	Node ID and HMAC information
0x11	Authentication response	Master ECU	Authentication acceptance/rejection

B. HMAC-Based Node Authentication

The authentication mechanism is designed to reduce the risk of unauthorized node access. Each valid ECU node and the Master ECU store a pre-shared secret key. When a node attempts to join the network, it sends an authentication request containing its node ID and authentication information.

The HMAC value may be represented as:

$$HMAC(K, m) = H[(K' \oplus opad) \parallel H((K' \oplus ipad) \parallel m)]$$

where:

- K is the secret key,
- m is the authentication message,
- K' is the processed key,
- $ipad$ is the inner padding value,
- $opad$ is the outer padding value, and
- H represents the selected hash function.

In the prototype, the node identity message is generated using the node ID, Master ECU ID, and pre-shared key. The Master ECU independently calculates the expected HMAC and compares it with the received value. If both values match, the node is registered and communication is permitted.

The authentication process consists of the following steps:

1. Node starts and initializes CAN FD communication.
2. Node sends an authentication request.

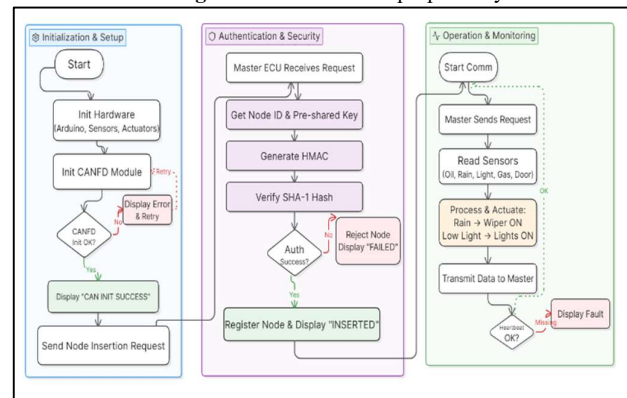
3. Master ECU receives the node ID and HMAC value.
4. Master ECU calculates the expected HMAC.
5. The received and calculated values are compared.
6. If verification is successful, the node is registered.
7. If verification fails, the node is rejected.

5. WORKING METHODOLOGY

The operational flow of the proposed system is as follows:

8. All ECUs initialize their respective sensors, actuators, LCD interface, and CAN FD modules.
9. Each node sends an authentication request to the Master ECU.
10. The Master ECU verifies the HMAC value.
11. Authenticated nodes are added to the communication list.
12. The Master ECU periodically sends request messages to Node 1, Node 2, Node 3, and Node 4.
13. Each requested node reads the associated sensor values.
14. Local control decisions are performed at the node level.
15. Processed data and status values are transmitted to the Master ECU.
16. The Master ECU displays received values and system conditions on the LCD.
17. If a node does not respond within a specified interval, the Master ECU may indicate a communication fault or missing-node condition.

Fig -2: Flowchart of the proposed system



This distributed methodology reduces the processing burden on the Master ECU because sensor acquisition and actuator decisions are performed locally at each node.

6. EXPERIMENTAL RESULTS AND DISCUSSION

The prototype was tested using Arduino Uno controllers, MCP2518FD CAN FD modules, sensors, and actuator modules. The experimental arrangement included the Master ECU and four distributed ECU nodes interconnected through the CAN FD bus.

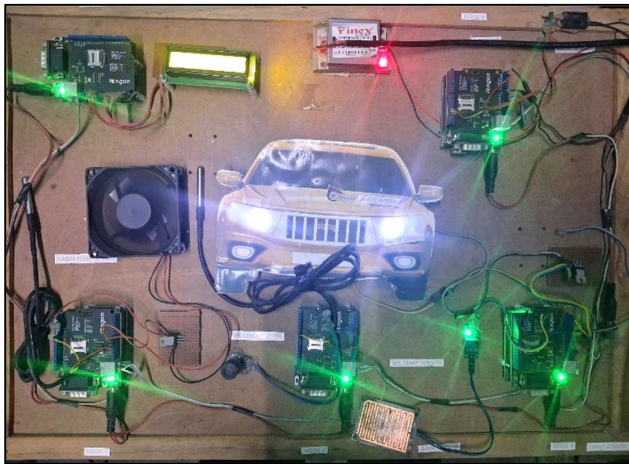


Fig -3: Experimental setup of the proposed system

A. CAN FD Communication Results

The CAN FD communication link between the Master ECU and all sensor nodes was established successfully. The Master ECU transmitted request messages, and authenticated nodes responded with their corresponding sensor and status data. The request-response approach reduced unnecessary periodic transmissions because a node transmitted data mainly when the Master ECU requested it. This method improved bus utilization in the prototype and simplified monitoring at the Master ECU.

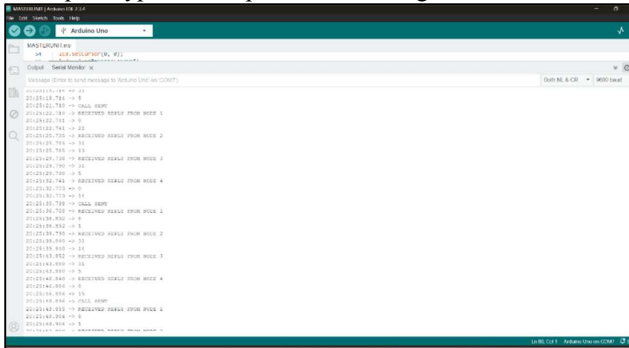


Fig -4: CANFD Communication Results

B. Node Authentication Results

The node-authentication procedure was tested by sending authentication requests from valid and invalid node configurations. Valid nodes that generated the expected HMAC were accepted by the Master ECU. Nodes with invalid authentication values were rejected and were not permitted to participate in normal data communication.

The authentication mechanism demonstrated the following capabilities:

- Verification of node identity

- Rejection of invalid node requests

- Restriction of normal communication to authenticated nodes

- Improved protection against unauthorized-node insertion in the prototype

C. Node 1: Cabin Temperature and Cooling Fan Results

Node 1 successfully measured the cabin temperature using the DS18B20 sensor. When the measured temperature exceeded

the configured threshold, the cooling fan was activated automatically. The fan condition and temperature value were transmitted to the Master ECU.

The test demonstrated that local control at Node 1 allows rapid cooling-fan activation without requiring continuous decision-making by the Master ECU.

D. Node 2: Oil Monitoring Results

Node 2 successfully measured oil temperature and oil-level conditions. The DS18B20 sensor provided digital temperature data, while the analog oil-level sensor provided a level-related signal. The processed values were sent to the Master ECU through CAN FD communication.

The Master ECU displayed oil temperature and oil-level status on the LCD. The experimental result confirms that Node 2 can support real-time condition monitoring for lubrication-related parameters in a prototype vehicle system.

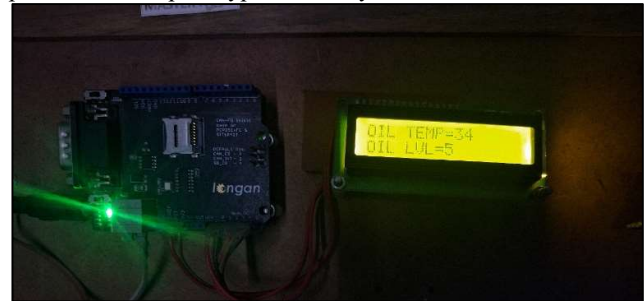


Fig -5: Oil monitoring system results on setup

E. Node 3: Rain Detection and Smart Lighting Results

Node 3 detected rain conditions through the rain sensor. When the rain intensity crossed the configured threshold, the servo motor was activated to simulate automatic windshield-wiper movement.

The LDR sensor successfully detected low-light conditions. During reduced ambient light, the lighting output was activated automatically. The node transmitted rain status, wiper status, light level, and lighting condition to the Master ECU.

The combined rain and smart-lighting node reduced the number of separate controllers required for environmental automation functions.



Fig -6: Rain Detection and Smart Lighting results on setup

F. Node 4: Security Monitoring Results

Node 4 successfully monitored gas-sensor variation and door status. The MQ-7 sensor output was used as a carbon-monoxide

monitoring indicator, while the magnetic door sensor detected open and closed conditions.



Fig -7: Security Monitoring results on setup

The Master ECU received and displayed the security data. The node provides a basic prototype for cabin-safety and intrusion-monitoring applications.

Table 4: Summary of prototype functional results

Function	Result
CAN FD communication	Successfully established between Master ECU and nodes
Node authentication	Valid HMAC nodes accepted; invalid nodes rejected
Cabin temperature monitoring	Successfully measured using DS18B20
Cooling fan control	Activated when threshold condition was met
Oil temperature monitoring	Successfully transmitted to Master ECU
Oil-level monitoring	Analog level status successfully processed
Rain detection	Rain condition detected by Node 3
Wiper control	Servo motor activated based on rain threshold
Smart lighting	Lighting output activated under low-light conditions
Security monitoring	CO sensor variation and door state monitored

7. ADVANTAGES OF THE PROPOSED SYSTEM

The major advantages of the proposed system are as follows:

1. CAN FD provides increased payload support compared with classical CAN.
2. The distributed ECU architecture improves modularity and simplifies future expansion.
3. Local control reduces the processing load on the Master ECU.
4. Node 1 provides automatic cabin cooling based on temperature conditions.
5. Node 2 supports continuous oil-temperature and oil-level monitoring.
6. Node 3 integrates rain sensing, wiper control, and smart-lighting control.
7. Node 4 supports basic gas monitoring and door-security detection.
8. HMAC-based node verification improves security against unauthorized node insertion.
9. Request-response communication reduces unnecessary network traffic.
10. The prototype uses low-cost and readily available embedded hardware.

8. LIMITATIONS AND FUTURE SCOPE

The proposed system is a laboratory-scale prototype and has several limitations. Arduino Uno boards, basic sensors, and non-automotive-grade modules are used for demonstration. The MQ-7 sensor provides an indicative gas response and requires appropriate calibration for accurate concentration estimation. Similarly, rain and LDR thresholds depend on environmental conditions and sensor calibration.

The current authentication design is based on a pre-shared-key HMAC approach. Although suitable for a prototype, a production automotive system should use modern cryptographic methods, secure key storage, anti-replay protection, secure boot, hardware security modules, and stronger hash functions such as SHA-256.

Future work may include:

- Automotive-grade CAN FD ECUs and transceivers
- SHA-256-based HMAC or AES-CMAC authentication
- Secure key exchange and key-rotation mechanisms
- Intrusion Detection System (IDS) for abnormal CAN FD traffic
- Real-time data logging and cloud-based vehicle monitoring
- Mobile application or graphical dashboard
- GPS-based tracking and emergency-alert features
- Predictive maintenance using machine-learning techniques
- Additional ECU nodes for battery monitoring, braking, engine diagnostics, and driver-assistance functions

9. CONCLUSION

This paper presented a secure multi-node automotive monitoring and control system using CAN FD communication

protocol. The developed prototype consists of four functional ECU nodes and one Master ECU supervisor. Node 1 monitors cabin temperature and controls the cooling fan. Node 2 monitors oil temperature and oil level. Node 3 performs rain detection, automatic wiper control, and smart-lighting control. Node 4 provides carbon-monoxide and door-status security monitoring.

The Master ECU supervises CAN FD communication, receives sensor data, displays system status, and verifies nodes through an HMAC-based authentication mechanism. The distributed architecture enables local sensing and actuator decisions, thereby reducing the processing requirement of the Master ECU and improving modularity.

Experimental observations confirmed successful communication between the Master ECU and distributed nodes. The system also demonstrated automatic cooling-fan operation, oil-condition monitoring, rain-triggered wiper control, automatic lighting, and security-status monitoring. The HMAC-based authentication method allowed valid nodes to access the network while rejecting invalid authentication requests.

The proposed design provides a practical foundation for secure automotive embedded communication systems. With automotive-grade hardware and enhanced cryptographic protection, the system can be extended for advanced connected-vehicle, safety, diagnostic, and intelligent transportation applications.

REFERENCES

- [1] R. Bosch GmbH, "CAN Specification Version 2.0," Robert Bosch GmbH, Stuttgart, Germany, 1991.
- [2] F. Hartwich, "CAN with Flexible Data-Rate," in *Proceedings of the International CAN Conference*, Germany, 2012.
- [3] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental security analysis of a modern automobile," in *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 2010, pp. 447–462.
- [4] Q. Wang, S. Sawhney, and M. Ghosh, "Lightweight secure authentication protocol for in-vehicle CAN networks," *IEEE Access*, vol. 7, pp. 148721–148732, 2019.
- [5] S. Patil and P. Kulkarni, "Automatic rain sensing car wiper system using embedded controller," *International Journal of Engineering Research and Applications*, vol. 4, no. 2, pp. 1525–1528, 2014.
- [6] M. K. Hasan, A. Rahman, and T. S. Uddin, "Automatic vehicle headlight control system using LDR sensor," *International Journal of Computer Applications*, vol. 180, no. 30, pp. 22–26, 2018.
- [7] K. Etschberger, *Controller Area Network: Basics, Protocols, Chips and Applications*. Germany: IXXAT Press, 2001.
- [8] H. Kopetz, *Real-Time Systems: Design Principles for Distributed Embedded Applications*. New York, USA: Springer, 2011.
- [9] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 7th ed. Pearson Education, 2017.
- [10] M. Wolf and A. Weimerskirch, "Security in automotive bus systems," in *Workshop on Embedded Security in Cars (ESCAR)*, Germany, 2004.
- [11] Microchip Technology Inc., "MCP2517FD/MCP2518FD CAN FD Controller with SPI Interface Data Sheet," Microchip Technology Inc.
- [12] Arduino Documentation Team, "Arduino Uno Rev3 Technical Specifications," Arduino Official Documentation.
- [13] Analog Devices, "DS18B20 Programmable Resolution 1-Wire Digital Thermometer Data Sheet," Analog Devices, Inc.
- [14] Hanwei Electronics, "MQ-7 Semiconductor Sensor for Carbon Monoxide Data Sheet," Hanwei Electronics.