Design of special vehicle condition monitoring system based on J1939

Xuan Shao*, Xingwu Kang, Xuping Wang, Xiaojing Yuan
Rocket Force University of Engineering Xi’an, China
*1009675774@qq.com

Abstract. The special vehicle is one of the important equipment in China's military and aerospace fields, and its condition is closely related to the success of the mission. In view of the lack of real-time condition monitoring capability of special vehicles, a condition monitoring system based on J1939 protocol is designed in this paper. The system communicate with the Electronic Control Unit (ECU) in Controller Area Network (CAN) through the On Board Diagnostics (OBD) interface to obtain the real-time condition data of the vehicle and improve the active safety of the vehicle. The test results show that the system realizes the data exchange with the Electronic Control Unit (ECU), outputs the data to the personal computer (PC) through the serial port, realizes the protocol data visualization, and meets the expected design purpose.

1. Introduction
With the rapid development of China's military and aerospace fields, special vehicles play an increasingly important role. In order to successfully complete the task, the state of the vehicle should be monitored to reduce safety risks. At present, vehicle condition monitoring is mainly based on regular maintenance, which is lack of real-time dynamic.

At present, most of the special vehicles use electronic diesel engines, and their Electronic Control Unit (ECU) supports J1939 protocol, which can store relevant operation parameters and fault information [1]. Real-time vehicle data be obtained by communicating with the ECU in Controller Area Network (CAN) through the On Board Diagnostics (OBD) interface. But the OBD data utilization rate for special vehicle is still low and intelligent condition monitoring level is not high, in order to solve this problem, this paper developed a condition monitoring system based on J1939 protocol, which can obtain the vehicle state parameters, such as engine speed, speed, engine load and fault code in real time. It lays a foundation for fast and intuitive interfacial display of state information and multi-information fusion in the later period, which has a great significance.

2. J1939 protocol
The J1939 protocol, also known as the SAE J1939 protocol, is an architecture developed by the society of automotive engineers for the standardization of buses. It based on can2.0B protocol, high-speed communication is carried out through CAN bus network with the communication rate up to 250kb/s [2]. It is mainly used for establishing high-speed communication network between ECUs in commercial vehicles such as heavy trucks and buses [3], which is the most widely used protocol among large vehicles at present.
The J1939 protocol is built based on the Open System Interconnection (OSI) model [4], and each layer is described by corresponding documents, including data link layer (J1939-21), vehicle application layer (J1939-71) and diagnostic application layer (J1939-73). The data link layer is described by the can2.0B network protocol, which redefines the 29-bit identifier of the extended frame, including the ECU name and address definition, priority, etc., so that it can describe all features of the message. In addition, the message itself is more precisely defined. The external device transmits data to the ECU through the CAN line in the OBD interface (Figure 1). Different from CAN protocol, which takes frames as data unit, J1939 protocol takes protocol data unit (PDU) as unit during transmission, and each PDU is composed of 29-bit ID and data bytes (0-8) (Table 1). As can be seen from Table 1, identifiers include Priority bits, reserved bit (R), data page bit (DP), PDU format bits (PF), specific PDU bits (PS), and source address bits (SA) [5], and both Substitute Remote Request (SRR) bits and Identifier Extension (IDE) bits are defined in can2.0B, so they are not included in the PDU. PF is used to distinguish between two depressurizes the battery through the vehicle OBD interface, kinds of PDU format (PDU1 and PDU2), when PF between 0-239, PS contains the destination address; When PF between 240-255, PS contains global address.

![Figure 1. OBD interface diagram](image)

**Table 1. J1939 frame format**

| CAN Extended Frame Identifier | Start of Frame (SOF) | 11-bit Identifier | SRR | IDE | 18-bit Identifier |
|-------------------------------|----------------------|-------------------|-----|-----|------------------|
| J1939 Frame Format            | 1                    | Priority R DP PF  | 13  | 14  | PF PS SA         |
|                               |                      | 2-4 5 6 7-12      | 15-16 17-24 25-32 |

**Table 2. Structure of DTC**

| DTC                                                                 | SPN | FMI | CM | OC |
|----------------------------------------------------------------------|-----|-----|----|----|
| Byte 3 8 least significant bits of SPN (bit 8 most significant)    | 8   | 7   | 6  | 5  |
| Byte 4 second byte of SPN (bit 8 most significant)                  | 4   | 3   | 2  | 1  |
| Byte 5 3 most significant bits of SPN and the FMI (bit 8 SPN msb and bit 5 FMI msb) | 8   | 7   | 6  | 5  |
| Byte 6                                                               | 4   | 3   | 2  | 1  |

The application layer of J1939 protocol is divided into vehicle application layer (J1939-71) and diagnostic application layer (J1939-73). The vehicle application layer mainly defines various parameters of vehicle control and communication, such as suspect parameter number (SPN) and parameter group number (PGN) [6]. The diagnostic application layer mainly defines diagnostic message (DM) and diagnostic trouble code (DTC). A diagnostic trouble code (DTC) consists of four
parts: suspect parameter number (SPN), failure mode indicator (FMI), occurrence count (OC) and conversion method (CM) (Tab. II) [7].

![Figure 2. Overall design of system](image)

3. Overall design of system
As shown in figure 2, the system architecture is mainly composed of microcontroller, CAN communication module, serial port module and power module. The power module depressurizes the battery through the vehicle OBD interface, providing power to the microcontroller, CAN communication module and serial port module. CAN communication module is responsible for collecting data information from CAN bus and uploading it to the microcontroller. The microcontroller performs message parsing according to J1939 protocol, and uploads the operation parameters and fault information to the PC for displaying through the serial port module. At the same time, the PC can also send the request command through the serial port module, which will be processed by the microcontroller and uploaded to the CAN bus, and then wait for the ECU response.

4. Hardware design
According to the system design requirements, the system hardware mainly includes microcontroller, CAN communication module and serial port module.
The microcontroller adopts STM32F103C8T6 (Fig. 3), a 32-bit microprocessor based on the ARM cortex-m3 kernel produced by STMicroelectronics (ST). The processor has a mature technology, high cost performance, a processing capacity of 1.25 DMIPS/MHz, which has abundant internal resources, including CAN controller supporting CAN2.0B protocol, and can be externally connected to I2C, ADC, USB, SPI and other interfaces [8]. Combined with the CAN communication module, the bus data can be sent and received in real time and parsed directly through the vehicle OBD interface.

The CAN communication module adopts TJA1050, which is one of the mainstream CAN communication modules at present. It has high speed, low electromagnetic radiation, high reliability and high anti-electromagnetic interference ability. In Fig. 4, TXD terminal and RXD terminal are connected to CAN_TX and CAN_RX terminal in the microcontroller respectively, so as to convert the data information of CAN controller into level signals, which then communicate with the bus through CAN_H and CAN_L to complete data transmission.
The serial port module adopts CH340G chip to realize the transformation between USB and serial port. Its circuit is shown in Fig. 5. The chip is fully compatible with Windows serial applications, with a wide range of power supply and low price to meet the design requirements.

5. Software design

Figure 6. Flow chart of condition monitoring
The condition monitoring system developed in this paper mainly realizes the real-time acquisition of vehicle operation parameters and fault information (Fig. 6). According to the selection of microcontroller, Keil uVision5 MDK was adopted as the software development environment, and the programming language is C language. Library functions were used for development, and the program was connected to the microcontroller SWD port through j-link to realize online simulation and debugging.

The J1939 protocol stipulates that some important parameters shall be broadcast in a certain period [9], each manufacturer can modify them according to its own situation, but the content is similar. In order to realize real-time monitoring of vehicle operation parameters and fault information, this system sends requests to vehicles through CAN bus regularly and actively to obtain real-time parameters and fault information.

After the system sends the request, the ECU receives the message and responds to the request, uploading the relevant data to the CAN bus. The system monitors the CAN bus in real time. When the message ID is detected to be consistent with the request ID, the receiving interrupt is generated and the message is stored in the buffer. In addition, the message is parsed according to J1939 protocol specification to obtain data and fault information. For example, when the system requests the throttle position, the parameter group number of the message sent is FE F2 (PGN=65266), and the message received by ECU is 00, 00, 00, 00, 00, 00, 31, 00. According to the J1939 protocol, the 7th byte represents the throttle position, and the calculation formula = byte data (hex) × resolution + offset, the resolution is 0.4%/ bit, the offset is 0, and the data range is 0%-100%. Converting hexadecimal byte data to decimal number, calculate the throttle position =49×0.4%+0=19.2%. Other parameters can be calculated similarly.

When the message is parsed, the system calls the Usart_SendByte function to upload the extracted data information to the PC through the serial port module, and realize the real-time display of vehicle running parameters and fault information through the serial port debugging assistant.

6. Experimental test

In this paper, J1939 simulator is used for data simulation. The simulator is designed according to J1939 protocol, which can simulate the vehicle ECU for data transmission. CAN analyzer is used for communication interception between the system and the simulator to obtain the original communication data and verify whether the communication is normal. Finally, the serial debugging assistant is used to display the parsed data on PC to check whether the parsed results of each parameter are correct.

In order to complete the system performance test and correctly evaluate the system performance, this paper took the vehicle speed, engine speed, throttle position, coolant temperature (Tab. III), as well as the increased engine coolant load failure and exhaust gas recirculation (EGR) mass flow failure (Tab. IV) as examples to test the system.

| Parameter name         | PGN     | Byte position | Resolution | Offset |
|------------------------|---------|---------------|------------|--------|
| Speed (km/h)           | FEBF    | 1-2           | 1/256      | 0      |
| Engine speed (rpm)     | F004    | 4-5           | 0.125      | 0      |
| Throttle position (%)  | FEF2    | 7             | 0.4        | 0      |
| Coolant temperature (℃)| FEEE    | 1             | 1          | -40    |

| Fault name                   | SPN  | FM | OC |
|------------------------------|------|----|----|
| Increased engine coolant load| 1082 | 15 | 3  |
| Exhaust gas recirculation (EGR) mass flow | 2659 | 16 | 2  |
The CAN bus communication result between the listener system and the simulator is as shown in Fig. 7. It can be seen from the figure that the simulator has sent relevant messages in time to respond to the operating parameters and fault information requests sent by the system.

The parsed message displayed by the PC serial port debugging assistant is shown in Fig. 8. It can be seen from the figure that the parsed message is displayed in the form of hexadecimal numbers. After conversion to decimal, they can correspond to the corresponding parameters of the simulator one by one (respectively the vehicle speed, engine speed, throttle position, coolant temperature, and fault information). Therefore, it can be seen that the message parsing result is basically correct, and the system performance test is normal.

7. Conclusion
According to the requirements of the current special vehicle condition monitoring, this paper designs a special vehicle condition monitoring system based on the J1939 protocol. After testing, the system meets the design requirements. It can collect CAN bus data, realize the real-time monitoring of vehicle operating parameters and fault information, and lay the foundation for subsequent vehicle maintenance management. It is of great significance to improve proactive retentiveness and reduce safety risks.

Acknowledgments
The completion of this paper is inseparable from the help of others. I would like to thank my supervisor Professor Kang Xingwu for his support. His support is my motivation. Thanks to Mr. Wang Xuping for his hard work on the paper. His help, encouragement and guidance allowed me to complete
the paper smoothly. In addition, I also want to thank everyone who silently contributed to the paper, I cannot complete the paper without you, thanks!

References

[1] Y. Zhang, W. H. Zhou, T. F. Yang, X. Q. Guo, “Research on Monitoring Requirements and Management of Electronically Controlled Diesel OBD System,” Journal of Mechanical & Electrical Engineering, vol. 32, pp. 265-269, February 2015.

[2] F. Luo, Z. C. Sun, Principle, Design and Application of Automotive CAN Bus System. Beijing: Electronic Industry Press, 2010.

[3] X. H. Dong, J. B. Hu, H. Huang, Y. M. Hu, H. S. Gu, “Design of Intelligent Vehicle Terminal System Based on SAE J1939,” COMPUTER MEASUREMENT & CONTROL, vol. 27, pp. 209-213, July 2019.

[4] Z. H. Luo, Y. Xu, Q. X. Ma, “Design and Implementation of Commercial Vehicle Online Fault Diagnosis System,” COMPUTER MEASUREMENT & CONTROL, vol. 23, pp. 1098-1101, April 2015.

[5] SAE Standards, Data Link Layer, SAE J1939/21, Society Of Automotive Engineers, 2010.

[6] SAE Standards, Vehicle Application Layer, SAE J1939/71, Society Of Automotive Engineers, 2016.

[7] SAE Standards, Vehicle Application Layer-Diagnostics, SAE J1939/73, Society Of Automotive Engineers, 2016.

[8] Y. X. Chen, W. Y. Li, J. G. Mo, “Vehicle information acquisition and diagnosis module based on SAE J1939 protocol,” ELECTRONIC TECHNOLOGY, vol. 47, pp. 65-66+51, July 2010.

[9] J. H. Zhang, Y. Xu, “Vehicle monitoring and fault diagnosis on-board system based on SAE J1939,” COMPUTER SYSTEMS & APPLICATIONS, vol. 22, pp. 73 - 77, March 2013.