Design of Remote Monitoring System for New Energy Vehicles Based on Data Acquisition and Transmission

Lijie Gao1,*, Dapeng Yang1

1Zhuhai College of Science and Technology, Zhuhai, Guangdong, China, 519040

*Corresponding author e-mail: gaolijie354@jluzh.edu.cn

Abstract. This paper designs a new remote monitoring system for new energy vehicles based on data acquisition and transmission, hoping to make up for the shortcomings of the existing system. The following article will start from the shortcomings of the existing system, then describe the performance of the new system, and finally describe the design scheme of the vehicle terminal of the system in detail. The performance of the new system will be even better, and it will have positive significance for the improvement of other remote monitoring devices.

Keywords: Data Acquisition, Data Transmission, New Energy Vehicles, Remote Monitoring System Design

1. Introduction
As the number of new energy vehicles continues to grow, the number of old vehicles continues to increase, the frequency of accidents is significantly higher, the safety of new energy vehicles is highly valued by the state. Therefore, it is urgent to design a remote monitoring system for new energy vehicles to realize real-time data collection and transmission. The new energy vehicle remote monitoring system can not only meet the national regulatory requirements, but also protect public life safety and enhance the effect of skill application.

2. Current vehicle remote monitoring system
Most of the current vehicle remote monitoring systems are based on 20/25G wireless networks such as GSM/GPRS, GPRS in actual use, communication time delay is large, RTT (Round Trip Time) is usually in 1000ms-3000ms, and may even swell to 10s; data throughput rate is affected by actual communication conditions, such as terminal movement speed, cell switching, network congestion, etc., and the range of variation is very large [1]. The data throughput rate is affected by the actual communication conditions, such as terminal movement speed, cell switching, network congestion, etc., and the range of variation is very large, and the packet loss rate will increase, which greatly affects the system real-time and communication reliability; due to the principle factors, its TCP performance stability and network security are also inferior to CDMA1xEVDO network. For these reasons, GPRS will not be able to meet the needs of applications with large data traffic and high interactivity, such as the Telematics service currently being developed.
To address the above deficiencies of GPRS network, the system adopts CDMA1xEVDO network as the wireless communication solution to prepare for the demand of communication bandwidth for the expansion of multimedia services such as image and sound, and fully considers the scalability and software portability in the system hardware and software architecture design; in order to improve the system real-time and resource management capability, the system engages RTOS for the real-time task design of the vehicle terminal. In order to improve the system's real-time performance and resource management capability, RTOS is used for the real-time task design of the vehicle-mounted terminal; at the same time, the communication data is verified and the communication procedure is reasonably designed to minimize the data loss rate and improve the communication reliability; the data collection and monitoring center carries out authentication and authority management for the accessing users to ensure the security; in addition, the system supports the functions of real-time query of data locally and remotely, historical statistical information query, vehicle-mounted terminal operation self-test, and remote online software upgrade [2].

3. CDMA1 x EVDOTCP performance and system communication process
CDMA1 x EVDO is a technology optimized specifically for packet data services, which is designed to reach peak rates of 3.1 Mbps downlink/18Mbps uplink (RevA) and 2.4Mbps downlink/153Kbps uplink (Rev.0), which is compatible with the next version. According to the actual test, the data throughput rate can reach 304-572Kbps downlink/85-135Kbps uplink and can be stable above 300Kbps downlink/80Kbps uplink; the packet error rate is below 0.2% downlink/1% uplink in 90% probability; the communication time delay (RTT) is usually 150-1400ms and below 220ms in 90% probability. In addition, CDMA1xEVDO increases the interface between different Access Networks (ANs) and improves the switching performance between ANs, which shows that the TCP performance of CDMA1xEVDO (hereinafter referred to as EVDO) is much better than GPRS network. The composition of the remote monitoring system based on EVDO network is shown in Figure 1.

![Communication chain of CDMA1x EVDO-based remote monitoring system](image)

**Figure 1.** Communication chain of CDMA1x EVDO-based remote monitoring system

The system communication process can be divided into 3 processes: uplink communication, downlink communication and remote access of the client [3].

Uplink communication: The vehicle terminal controller establishes HRPD (High RatePacketData) and PPP (Point-to-Point Protocol) sessions by controlling the EVDO module, and PDSN (Pocket DataServiceNode) assigns IP addresses to the module to connect the vehicle terminal to the Internet, and then establish TCP connection with the remote monitoring center server (if TCP protocol is used), so that the terminal can upload the data collected from the vehicle bus and GPS receiver to the data collection monitoring center database, and close the TCP connection after the transmission is finished [4].
Downlink communication: Assuming that the EVDO module of the vehicle-mounted terminal is in the listening state of the server, the remote communication server initiates a TCP connection request to the vehicle-mounted terminal at this time, and after the vehicle-mounted terminal answers, a downlink TCP connection can be established, and then the TCP connection is closed after the downlink transmission of data. Authorized users can log in to query the real-time, historical information or fault information of their vehicles through client browsers such as PC smartphones [5].

4. Vehicle terminal design

4.1. Overall design
The vehicle-mounted terminal will complete the following functions: data collection and transmission at regular intervals, local remote real-time data query, fault alarm and prompt, and remote online software upgrade. In addition, the terminal also supports running self-test, local data storage, uplink data and downlink data verification and other functions.

(1) External interfaces and protocols: The vehicle-mounted terminal has four types of external interfaces: the interface for receiving GPS satellite broadcast signals, the CAN bus interface for communicating with new energy vehicles, the wireless communication interface with remote communication servers and the human-machine interface for interacting with users. CAN bus interface; EVDO module is the wireless access interface to complete the remote transmission of data; human-machine interface is composed of buttons, buzzer, LCD or microphone speaker and other devices, because the CAN communication network application layer protocol of new energy vehicles such as electric vehicles has not formed a standard in the industry, mostly based on SAE J1939 protocol design, such as CN2004A (B), the application needs to Most of them are designed based on SAE J1939 protocol, such as CN2004A(B), and the parameters need to be modified to adapt to the specific model; while the remote communication is based on TCP/IP protocol but the application protocol needs to be designed [6].

(2) M0: Timing mode. The general working mode of the vehicle-mounted terminal is to complete the vehicle status data timing collection, sampling GPS data (A) and vehicle-mounted data (B) at regular intervals and storing them locally, with a sampling period of no more than 10s; timing transmission, uploading the formatted GPS data (A) and vehicle-mounted data (B) to the remote communication server with an upload period of no more than 602. The operation status of the vehicle-mounted terminal is monitored to ensure its orderly and reliable operation [7].

(3) M1: Interrupt mode. The vehicle-mounted terminal enters interrupt mode in two cases: key interrupt and remote interrupt. The key interruption will collect the vehicle status data (A, B) in real time, and the data (without formatting; still AB) will be sent to the vehicle information terminal for display through the CAN bus (the vehicle information terminal should have the corresponding display driver); after the remote interruption is resolved by signaling, there will be three cases if it is fault information, the terminal will make the corresponding fault processing call; if it is a query command, the vehicle status data (A, B) will be displayed to the vehicle information terminal. Vehicle status data (A,B) will be collected in real time, and the data (without formatting at this time, still A,B) will be uploaded to the remote communication server; if it is an upgrade command, the terminal will make an upgrade processing call, thus entering the upgrade mode [8].

M2: Upgrade mode. When the software of vehicle terminal needs to be upgraded, the remote communication server sends the upgrade command to the terminal, and the terminal enters the upgrade mode from the interrupt mode to complete the remote download of image file (C) and Flash programming to realize online upgrade. After the upgrade, the terminal will restart the timing mode.

(4) Vehicle status data
The vehicle terminal collects at least the vehicle position, power battery voltage and current, temperature, SOC charging and discharging status, drive motor temperature, speed and torque, etc. In addition, other data can be collected according to the application needs.
4.2. Hardware design
According to the above function and behavior requirements, the vehicle terminal should include the main controller CAN controller and transceiver, wireless communication EVDO module, GPS receiver module, auxiliary storage unit (such as SD card, U disk), human-machine interface (such as LCD, buzzer, button), power supply unit clock unit, reset unit (including operation monitoring, such as Watchdog), programming and debugging interface (such as BDM (Background Debug Model), JTAG) and various interface adapters to ensure the normal operation of the terminal. An example hardware structure of the design is shown in Figure 2.

Figure 2. Software structure of the vehicle-mounted terminal
The connection relationship of each module is explained as follows: Since the main sampling is the vehicle powertrain status data, the vehicle CAN network uses high-speed CAN protocol communication rate is generally 250Kbps or 500Kbps; the GPS receiver and MCU use UART connection, and the output uses NMEA0183 protocol; the SD card module and MCU use SPI bus connection, which can realize high-speed The buzzer is driven by PWM and can output different frequencies according to the fault level; the EVDO module (the core device is Huawei MC703EVDO module) is connected to the MCU by standard RS-232-C, and the PCB can be made separately for easy disassembly or replacement; the TFTLCD and Button are connected to the MCU by GPIO (Generic Purpose Input Out put) and MCU;
for other modules, MCU has the corresponding special interface to connect with them. In addition, the terminal can also add modules as needed to achieve functional expansion: such as adding voice modules or CCD image sensors, etc. Which achieve multimedia applications [9].

4.3. Software design
Since there are many tasks in the vehicle-mounted terminal, the uC/OSII real-time kernel is used to design the software of the vehicle-mounted terminal in order to ensure the real-time performance of the system and the full utilization of resources. VDXOS standard.

(1) Software architecture. According to the concept of abstract design interface and implementation separation, the software of the vehicle-mounted terminal is abstracted layer by layer, and a common scheme of terminal application design can be established, and its software architecture is shown in Figure 3.

![Software architecture of vehicle-mounted terminal](image)

**Figure 3.** Software architecture of vehicle-mounted terminal

The software design of in-vehicle terminal can be abstracted into three layers: basic software layer, system interface layer, and application layer, which are described as follows:

First, the basic software layer. It includes uBoot (a section of assembly program to complete the C runtime environment configuration of MCU registers, interrupt vectors and storage space before the kernel is started, and to guide the kernel), IAP (In ApplicationProgramming) program (the core program for remote online upgrade of software), MCU and peripheral device drivers (of which FatFs is a kind of open-source file for small embedded applications), and the software interface layer. open source file system for small embedded applications) C/OS-II real-time kernel (which needs to be ported and slightly modified), and the base software provides the interface to access the underlying hardware of the vehicle terminal.

Second, the system interface layer. The various APIs (Application ProgrammingInterface) provided by the base software are further encapsulated into a unified system API to provide a run-time environment (RTE (Run-time Environment)) for the application layer real-time tasks.

Third, the application layer. The core procedure to realize the functions and behaviors of the vehicle terminal is the task division and its priority determination, and the task design should meet the system real-time and minimize the cost of inter-task communication and synchronization. DataTransfer), priority 9, runtime period 40s; Task4 RunningSupervisor, priority 11, runtime period 60s to achieve the monitoring of the terminal hardware runtime state. Among them, Task1 and Task2 need to be synchronized, and the message queue mechanism provided by uC/OSII can be used to implement the interrupt mode designed for two interrupt service tasks; Button ISR interrupt priority is designed for 15, to complete the local real-time query of vehicle status data; RemoteComISR, interrupt priority is 13 according to the signaling type to complete the vehicle fault Alarm, remote real-time query or switch to software online upgrade program [10].
(2) Communication protocol. Since the MCU has internal CAN controller and SPI controller, the standard CAN20 and SPI communication protocols are used, so these two protocols are not discussed; the UART-based NMEA0183V30 and UBX protocols are used for communication between the MCU and GPSReceiver, where UBX is used to set the GPSReceiver output data rate positioning. The NMEA0183V30 defines the output message format, sets the baud rate to 9600bps, the refresh frequency to 02Hz, the positioning mode to vehicle mode, and the MCU to interrupt the receiving mode.

The remote communication protocol is divided into uplink communication protocol and downlink communication protocol, the downlink communication protocol includes the control protocol for the vehicle terminal and the file transfer protocol for remote online software upgrade, whose communication model is shown in Figure 7. The reliability of remote communication can be ensured by ensuring the transmission reliability of UART. If the MC703 is already connected to the Inter net, when uploading data, the MCU first sends an AT command to control the MC703 to establish a TCP connection with the remote communication server and turn on the transparent transmission mode, then transmit the data according to the above packet format, after the transmission is finished, the remote communication server sends an answer signal, and the MCU decides to end the transmission according to the answer signal (sends an AT command to control the MC703 to disconnect the TCP connection). The MCU decides to end the transmission according to the answer signal (send an AT command to control the MC703 to disconnect the TCP connection), or retransmit. If no answer signal is received for more than 15s, the communication is considered to have failed and the transmission is restarted.

In the downlink control protocol, the MC703 works in the server listening state. The remote communication server sends data to the MCU according to the above-mentioned packet format, and when the data arrives, UART1 of the MCU generates a receive interrupt, and the MCU enters the RemoteComISR interrupt service program (first sends a receive answer signal, and then responds according to the ID type). The remote communication server decides to end the transmission (disconnect TCP connection) or retransmit if no signal is received for more than 10 seconds based on the answer signal.

(3) Program image file transfer protocol. The software remote online upgrade program needs to download the program image file (binary file) compiled and generated by the remote host to the vehicle terminal, and the corresponding file transfer protocol must be defined. Fortunately, such file transfer protocols are already available, such as Xmodem, Ymodem, KERMIT, etc. If the commonly used Ymodem transfer protocol is used, the download data is subject to CRC checksum and error retransmission mechanisms to ensure transmission reliability.

5. Data acquisition and monitoring center server group model

The data acquisition and monitoring center needs to complete remote communication with vehicle terminals, manage vehicle status data, data analysis and fault diagnosis, provide vehicle users with remote query of their vehicle information, program file management and software development functions, and is a C/S, B/S hybrid architecture model, to achieve a certain structure and functional redundancy, to improve system reliability.

Communication Server realizes remote communication with each vehicle terminal (C/S mode), and the concurrency rate cannot be less than 10% of the total number of vehicles it monitors, which can be realized by using port multiplexing and multi-threading technologies; WEB/WAP server is designed in MVC mode (B/S mode, only real-time queries are communicated with vehicle terminals, and the rest of real-time are obtained from the central database). The rest of the real-time data are obtained from the central database, need to take into account the PC, mobile client of a variety of browsers and access to user rights management, authorized users can access the vehicle real-time, historical statistics and fault status information.

6. Conclusion

This paper designs a scalable and hierarchical hardware and software architecture, as well as a remote communication protocol based on UART and TCP/IP protocols to ensure communication reliability,
and proposes a C/S, B/S hybrid architecture of data acquisition and monitoring center server group model, which has certain guiding significance for the design of remote real-time monitoring system of new energy vehicles, and can also be used as a reference for the design of remote monitoring system of other devices. It is also a reference for the design of remote monitoring system for other devices.

References
[1] Zhang XF, Shen Y, Song SH. Design of remote monitoring system for new energy vehicles for large-scale demonstration operation [J] Automotive Engineering, 2012.34(5)
[2] Peng Jianye Feng Xin Jize, Functional design and application research of remote monitoring and diagnosis system for automobiles [J]. Shanghai Automotive Design Research 2011(3)
[3] Shen Yang, Research on the design of electric vehicle remote monitoring system based on GPRS network [J] Shanghai Jiaotong University Master's Thesis, 2013 Aerospace New Long March Electric Vehicle Technology Co.
[4] Tong Y, Chen Shundong. Design of remote monitoring terminal for pure electric buses[J]. Anhui Science and Technology, 2011(6):44-46.
[5] Xu Minrui. Research and application of remote monitoring system for charging and switching stations and electric vehicles [J]. Jiangsu Electrical Engineering, 2011, 30(5):50-52.
[6] He Xiangling,Zheng Gang. NMEA protocol for GPS communication and extraction of positioning data [J]. Computer Applications and Software, 2000, 21 (12) : 121 - 122.
[7] Chen W, Yu ZW, Wei CJ. Modeling analysis of GPS monitoring system data transmission based on GPRS [J]. Journal of Shandong University of Science and Technology (Natural Science Edition ), 2006,Vol.20 (6) : 51 - 54.
[8] Liu Jie. Vehicle remote monitoring system based on GPS and GPRS [D]. Xi'an University of Electronic Science and Technology, 2005.
[9] Bao JY. Analysis and application of target information transmission mode in GPS vehicle real-time monitoring system [J]. Electronic Science and Technology, 2004, 4: 15 - 17.
[10] Li Lingli, Yin Jun. Design and implementation of vehicle-based remote wireless monitoring system [J], 2008, 28 (3), 250 - 253, 257.