

Study of Real-time Data Collection and Release with In-vehicle System

Wang Shuai

School of Information and Electrical, Engineer
China University of Mining and Technology
Xuzhou, China

Li Jian ,Jiang Shuming, Wei Zhiqiang , Zhang
Jianfeng , Xu Shijie
Information Institute
Shandong Academy of Sciences
Jinan 250014, China

Abstract—Linux-based Remote monitoring system can be used in vehicles for remote data collection and management for the purpose of vehicle monitoring and control. This paper analyzes the requirement of in-vehicle data collection; management with embedded system, and explains the principles of users interaction with the remote embedded control terminal by embedded dynamic websites, and realized the transplanting of embedded Web Server BOA and embedded database SQLite under S3C2440. The paper also discusses the application of SQLite in real-time data collection with a serial communication program.

Keywords—Remote Monitoring, SQLite, BOA, Vehicle data acquisition

I. INTRODUCTION

The vehicle remote monitoring is that through the network system for remote monitoring and control of vehicles, including remote data acquisition of vehicle, remote monitoring, remote maintenance and dispatch management and so on. Through the vehicle remote monitoring, officers do not visit the scene, and they can work on vehicles to monitor, analyze and troubleshoot[1-3]. These applications are related to the vehicle associated with a large number of real-time data acquisition and processing.

Data processing can generally be taken two ways: one is based on files, the other is based on the database. For files, users can manipulate files directly, but I/O is expensive, and the application opens the file in an independent manner, therefore, the sharing of data is poor, and it affects the over performance of the system. For the database, as for the independence of data and programs, so, through the transaction and concurrency control to the schedule, computer can effectively have the shared operations such as query, storage, delete and so on to the data. Meanwhile, because the database have the security of the test, integrity checks and other security measures, so we can ensure that the system has better overall performance.

The paper with the application of remote monitoring system in vehicle management introduces a data collection, management and publication system of a serial data acquisition, SQLite as database, BOA as a server.

II. VEHICLE REMOTE CONTROL SYSTEM REQUIREMENTS

ANALYSIS

Through analysis, there are many vehicle relative parameters that need to monitor, the important parameters of them are operating parameters of a vehicle (position and speed, etc.), engine operating parameters, operation parameters and tire parameters and so on. According to the vehicle parameters that attained under the monitoring system, vehicle module can use pattern recognition technique to identify the vehicle breakdowns, dealing with fault classification, and real-time monitoring, diagnosing the fault, and treating the emergency of vehicle status[4]. For non-emergency fault and exception that not to determine online, it needs the information exchange between the vehicle module and remote monitoring Service center, so that can collaborate with remote knowledge base to complete the remote monitoring and fault diagnosis of vehicle condition.

Therefore, the basic functions of vehicle remote monitoring system can be summarized as following: real-time collection and transmission of the field data; processing and analysis of the collected data; analysis, judgments, and alarm of the processing results. Remote monitoring system architecture diagram shown in Figure 1.

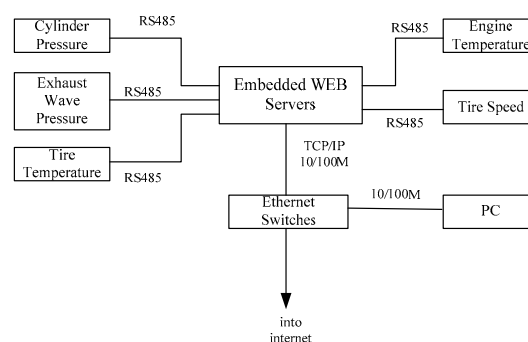


Figure 1. Remote monitoring system architecture diagram

The remote monitoring system architecture consists of three main parts: field data acquisition and transmission equipment, data server as the information interface, and the monitoring host.

From the above we can see: the embedded database to the way regular inspection of the sensor and the

microprocessor via the bus real-time signal acquisition and processing. If you use the vehicle database, the data is dynamic, so in addition to the state of synchronization with the associated equipment, but also the needs and data sources such as remote database to synchronize. On the other hand, as the information interface, the host Web server allows remote monitoring site via the browser in real time to monitor changes in the signal, so the administrator can promptly grasp the status of vehicle operation, and abnormal phenomena related to treatment. In practical applications, the embedded Web server is loaded inside the vehicle, monitoring the host only through the embedded Web server to access the wireless network.

Remote real-time vehicle monitoring system has many characteristics which users need, such as high message throughput, multi-user data sharing, and strong real-time characteristics, therefore, it is better to use database mode to do consolidated data operation and strengthen the repeatability of the application program.

III. HARDWARE STRUCTURE OF EMBEDDED REMOTE MONITORING SYSTEM

In order to realize the dynamic management of the embedded Web and analyze datum resources, we use the ARM9 processor + embedded Linux operating system as an embedded control system platform, and use wired network for data transmission now. And the development board is Samsung S3C244. The system structure shown in Figure 2.

We use Samsung S3C2440A as Microprocessor, basic frequency is 400MHz, it can realize networking protocol Conversion, control of Data Switching and Data Processing. The Ethernet interface chip is DM9000 which in line with IEEE802.3 as a interface chip, do communication in local area network on the physical layer, we use RJ-45 interface, which can be connected to the network directly. Memory modules are Samsung NOR Flash (4M) and the Samsung NAND Flash (64M), we store boot code, Linux kernel, user applications and system data in them. UART interface is using for microcontroller UART communication with the monitoring equipment. The board also has CCD external interfaces, LCD, USB, RS232, etc. We can expand memory by using SD card interface, or IDE interface.

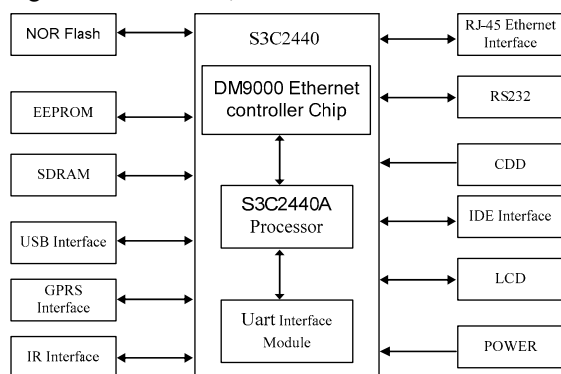


Figure 2. The system structure

IV. PORTING SQLITE EMBEDDED DATABASE

SQLite is a lightweight relational database, it is open source embedded database engine which written in C language [5]. It read and write on the hardware directly which is different from other SQL Database, it is complete independence and does not rely on external. It can use the finite resource of embedded systems efficiently, increases data access speed, and enhance system security.

SQLite provides support for SQL92, including support for multilist and indexes, transactions, views, triggers, a series of user interface and driver. SQLite is not running in an independent process space, but occupies a process space with the application program which it serves. It was embedded in the application program, provides a zero-configuration operation mode, and occupies very limited resource. SQLite is good at real-time characteristic, it has a small system expense, strong low-level control, and other satisfactory features.

SQLite can supports database which Data quantity as high as 2Tb. It divides each data file into equal number of pages, then organize these pages in the B + tree way. SQLite can supports null, real, text, integer, and blob data type but static data types, it uses column ties. This means, SQLite data types have the properties of the data itself, not the listed properties. Inserting a data to the database, SQLite will check whether it matches the related type of the column, and it will translate the data which does not match. If SQLite can not do the translation, the data will be stored as its own type.

Embedede systems is usually a resource-constrained systems. Directly write software in the hardware platform for embedded systems is more difficult. At present, usually the first compile or write programmes on a computer, and then, through cross-compiled to produce binary code that runs on the target platform, and finally is downloaded to the target platforms. Therefore, the method of transplantation SQLite database is that compiles relevant documents in a virtual machine firstly, then downloads the resulting file to the development board. This paper adopts open cross-development environment for embedded Linux, the package of Linux cross compiler tool chain is cross-3.4.1.tar.gz. PC machine's environment is RedHat9.0. The following steps are performed in the virtual machine:

1) Creat cross-compiler environment. Copy the cross-3.4.1.tar.gz to a directory, then enter the directory, do the extracting the command `tar -zxvf cross-3.4.1.tar.gz`. Set the cross compiler path `export PATH=/usr/local/arm/3.4.1/bin:$PATH`.

2) Download the latest version of `sqlite-3.6.18.tar.gz`, and copy it to the directory, then decompress.

3) Run the command: `./configure--disable-tcl--host=arm-Linux--prefix=/root/sqlite-3.6.16/build/target`, and then generate relevant documents.

4) Compile and install `make && make install`. Under the target directory there will be generated three objective folders: `bin`, `include`, `lib`. Copy the three folderes under the target directory to development board under the directory `usr` corresponding `bin`, `include`, `lib`.

V. EMBEDDED DATABASE BOA TRANSPLANT

BOA is a single task of the http server, open source, high-performance. With other Web servers is different, BOA is a single process, and it does not create a separate process for each visit, only for the CGI program to create new process. It also has the functions such as the automatic generation of directory, automatically extracting file and so on. Therefore, BOA has a high processing speed and efficiency of the HTTP request, and has high value in embedded systems[6].

BOA can complete the receive of client requests, the request of analyze, the request of responding, the results of the return to the client and other tasks. Its work involves:

- 1) Complete the initialization of Web server, including the creation of an environment variable, create a TCP socket, bind to port, listen, and wait to receive the client browser requests a connection.

- 2) After receive the connected of the client request, Web server saves the relevant request information, and analysis the request information. Parse out: request method, URL target, optional information such as inquiries and forms. At the same time, upon request, make the appropriate treatment.

- 3) After completing the appropriate Web server, BOA sends a response message to the client browser, closes the TCP connection with the client[2].

BOA database migration method is that compile the related files in a virtual, and then download the resulting file to the development board. The steps are as follows:

- 1) Download BOA source, run `./configure` under Linux environment, generate the makefile files.

- 2) Modify the Makefile file and find the `CC=gcc`, then make it to `CC=arm-linux-gcc`, find `CPP=gcc-E`, make it to `CPP=arm-linux-gcc-E`.

- 3) Run make command to receive an boa executable program, run `arm-linux-strip boa` command to remove debugging information in order to reduce the space occupied.

- 4) Modify `boa.conf` profile, according to their needs set up the document root directory of the server in the `boa.conf` and home page `index.html`, CGI files and `boa.conf` storage directory so the file itself, but must ensure that other supporting files and settings and `boa.conf` which match the configuration.

- 5) Mime.types stored in the server supports MIME file formats that can be modified according to actual needs.

After configuration, the documents related to BOA will be downloaded to the board under development, and complete IP configuration. The BOA web server program and the required documents such as CGI programs added to the development board of the corresponding directory. Start BOA Web Server, users can through a web browser (IE, etc.) to access the server.

VI. REALIZATION OF SYSTEM FUNCTIONS

The software used in the system are: embedded Linux operating system, SQLite-3.6.18 database, BOA-0.94-13

server, CGI written in C and serial data acquisition program, the Web with Html language editor. Software block diagram shown in Figure 3.

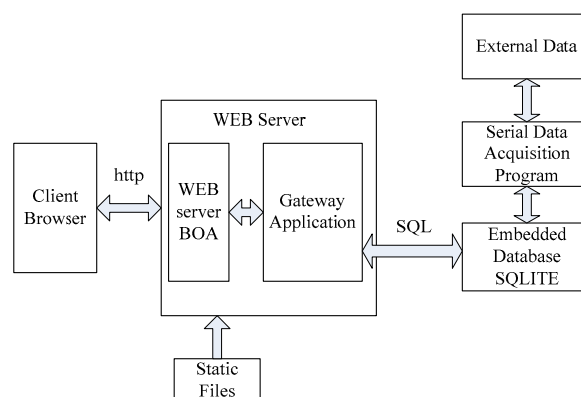


Figure 3. Block diagram of system software

In Linux, to implement dynamic web pages, it usually chooses CGI (common gateway interface). CGI provides a Web server to run external program access, this technology enables the server between the browser and server interaction[7,8]. Embedded WEB server BOA upon different request methods to make a different response.

Real-time data acquisition board is the basis for real-time system monitoring. Real-time data acquisition board follows the procedure of modules: data acquisition module, data processing module, data storage module. Data acquisition module uses sensors and microprocessors to obtain information on the vehicle to run an external signal components. The data collected will be sent to the data processing module. Data processing module filters the data collected, and the data processing module put the filtered data into the public data buffer[9]. Car-based remote monitoring of embedded database system architecture shown in Figure 3. Data storage module is responsible for saving the public buffer data to SQLite database. The key component of Data acquisition program is the baud rate settings and data stored in the database operation.

VII. SUMMARIES

In the embedded system, we using embedded database storage data which collected through external devices, combining with the embedded Web-Server to dynamically accessing and management of embedded systems application development is an important direction. This method simplifies the wholly application development processing, improving the data access and storage, achieving a multi-user data sharing, which is complete meet the embedded data acquisition and management of remote monitoring system. With the development of wireless networks and wireless communications equipment cost reduction, in the future, it will be a major aspect of embedded database which exchange data between inner and external system with a wireless network.

REFERENCES

- [1] ZhenXing Wang; LinXiang Shi; ZhongYuan Liu; ChuanQun Jiang; Embedded Web server and Database Based On Wap, Computer and Automation Engineering, 2009. ICCAE '09. International Conference on 8-10 March 2009 Page(s):200 - 204
- [2] Ahamed, S.I.; Vallecha, S.; Component-based embedded database for mobile embedded systems, Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on Volume 1, 2004 Page(s):534 - 538 Vol.1
- [3] Limin Xia. Equipment and vehicles remote fault diagnosis system based on B_S structure [J]. Applied Science, 2009, 2(12):124-125.
- [4] Gang Chen. The design and implementation of vehicle remote monitoring and fault diagnosis system based on ARM [J].Information Technology, 2008, 2(15):28-29.
- [5] SQLite Official Homepage [z]. <http://www.sqlite.org/>.
- [6] Wang ZhenXing, Ren XianYi. A Study on Cgi Of Embedded Webserver, Computer Science and Computational Technology, 2008. ISCSCT '08. International Symposium on Volume 1, 20-22 Dec. 2008 Page(s):480 – 483.
- [7] Danyi Liu; Al-Anbuky, A.; Ching Lin; Mobile embedded database for remote process management system, 2-5 Dec.2007 Page(s):515 – 520.
- [8] Yishi Xie, Bing Xu, The design of embedded web server and CGI implementation [J]. Computer Engineering and Design. 2007. 4. 1598- 1600.
- [9] Jianmin Ban, Baochuan Fu, Yin Zhu, The method of real-time data collection based on embedded database [J]. Microcomputer information. 2005, Vol21, No.7 .1-3.

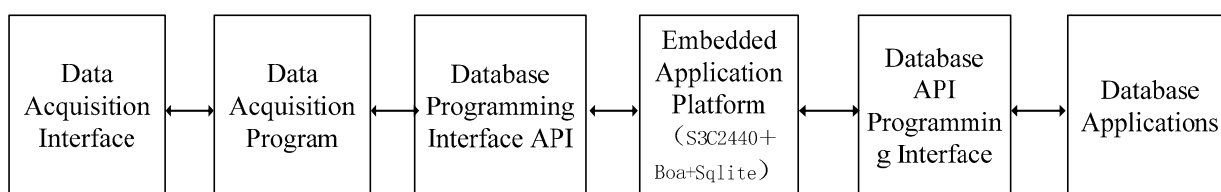


Figure 4. Car remote monitoring system structure