TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.5, May 2014, pp. 3578 ~ 3585
DOI: http://dx.doi.org/10.11591/telkomnika.v12i5.4916

A Networking and Data Transmission Mode of Mobile Seismic Station

Junlei Song*, **Ying Chen**, **Dianhong Wang**
Faculty of Mechanical & Electronic Information, China University of Geosciences, Wuhan, China
*Corresponding author, e-mail: junleisong@gmail.com

Abstract

The networking and data transmission mode of the field mobile seismic station involves such matters as richness, real-time and effectiveness of seismic data. In this paper, we propose a networking and data transmission mode with CAN bus and embedded technology. In light of the effectiveness, distance, cost and construction relating to data transmission in the field environment, CAN bus was utilized to connect the scattered seismic collection nodes, and the embedded microprocessor ARM was used to manage collection nodes within a certain scope. In addition, both the multiple task operating system Linux and application framework Qt were employed to construct a seismic network local management system, which significantly simplified the management task of mobile station. Such networking and data transmission mode shows such features as lower cost, easy construction, and higher reliability and instantaneity, also enables the possibility of constructing a mobile collection station with larger scale.

Keywords*: seismic data transmission, CAN bus, field communication, ARM, Linux*

1. Introduction

At present, the earthquake prediction information is mainly based on the seismic data acquired by widely scattered seismic observation stations. With the development and expansion of seismic observation station, demand for real-time performance and data quantity becomes higher and higher. There are many mobile observation stations in the field, in addition to the fixed ones. Mobile observation station is mainly designed for intensive data observation within a certain period of time and a certain region for the purpose of enhancing monitoring ability and anomaly tracing ability, as well as for collecting various kinds of physical quantities and feeding them back to the network center [1]. In such a case, a mobile station would always be deployed with a large number of dispersed collection nodes. While, on the other hand, how to transmit the data from such scattered collection nodes to the seismic network is one of the crucial factors to the field mobile observation technology.

In the complex field environment, it is unable to access each collection node to the network separately and the general approach is to gather the sampling node data to the mobile station, and then access to the network in a centralized way. Various transmission modes are available for local data, so the choice depends on such factors as the distance, convenience, reliability and cost. In this paper, a control area network (CAN) was employed as the transmission mode for seismic data from mobile observation station, which could meet the requirement of networking communication for scattered collection nodes in a relatively wide scope.

2. System Framework

Since there are large numbers of field collection nodes and the installation locations are quite scattered, directly constructing local network connection between such collection nodes and mobile station, no matter in what networking mode, will mean more investment of labor force and material resources as well as heavier management burden on the mobile station. Therefore, based on the theory of relay station, data of collection nodes in a certain area are gathered in the main controller through the bus mode, and then transmitted to the mobile station, with system structure as shown in Figure 1. By using the distributed way, the cost of constructing network for collection nodes is reduced. In addition, the main controller can

Received October 24, 2013; Revised December 16, 2013; Accepted January 5, 2014

supplant partial functions of the mobile station, manage and monitor collection nodes in the LAN, which facilitates the construction of field mobile station of larger scale.

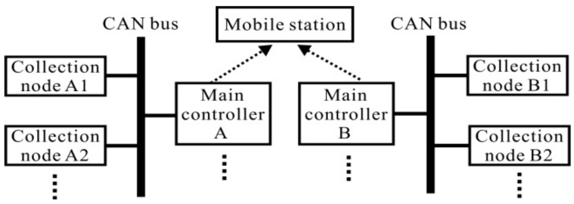


Figure 1. System Structure Diagram

The Controller Area Network (CAN) employed in the system is an ISO serial communication protocol, can effectively support the distributed type real-time control of higher safety level. Relative to other buses, CAN bus shows such advantages as longer distribution distance, faster speed, automatic retransmission, and non-destructive bus arbitration mechanism. The protocol is of short frame structure with high anti-interference ability and reliability, which can automatically disconnect wrong nodes from bus without affecting other nodes on the bus [2, 3], applicable to the no-manned field environment. In addition, CAN bus just requires simple transmission media so that the bus can be conveniently wired with lower cost in the field.

3. CAN Communication for Collection Nodes

C54 series DSP from TI is selected as the main control chip for collection nodes. CAN communication nodes should be designed on DSP platform since the chip has no CAN control port.

3.1. Design of DSP-based Transmission Node

The independent controller MCP2515 with SPI port is selected as the bus control end. The bus transceiver ADM 3054 is utilized as the transceiver. The chip directly integrates both the three-channel isolator and the transceiver, therefore the coupling isolation between the controller and transceiver is spared and the voltage level matching problem between them is under control [4], which in turn simplifies the peripheral circuit design, with its design diagram as shown in Figure 2. A crystal oscillator with stable performance must be provided to ensure the reliability of bus communication.

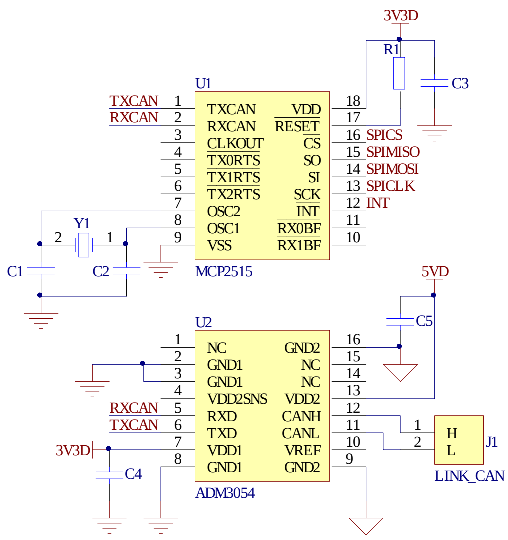


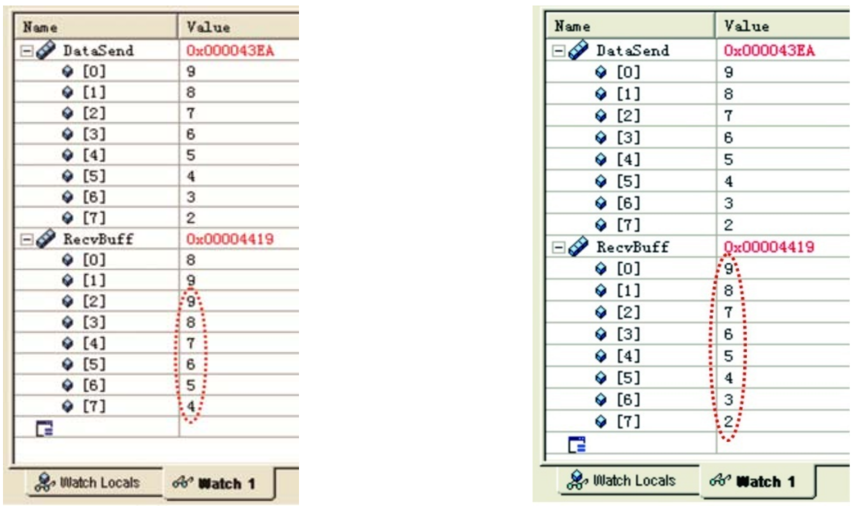
Figure 2. Schematic Diagram of CAN Bus Node

DSP multi-channel buffered serial port (McBSP) is configured into SPI compatible mode, and each frame has fixed length under the mode. While, since the MCP2515 SPI control command length is not uniform, a compromise solution is selected to configure serial ports with the longest command digits, namely 32 bits each frame.

Operation time matching is very important during communication in SPI mode. Since DSP has a work clock frequency far above SPI's transmission clock frequency, sufficient delay is required to be added during read operation. Both the picture (a) and (b) in Figure 3 shows the testing by using the loopback mode of controller: the transmitted 8-byte data is stored in DataSend digit group, while the value acquired by SPI read receive data register is stored in RecvBuff digit group. Figure 3(a) shows the read operation without adding sufficient delay, represented in the dotted box, in which, the read operation is subject to time warping, practically the data sent back by above command is read in current read operation. While in Figure 3(b), time correspondence is achieved and the correct data is read after adding sufficient delay.

3.2. Communication Strategy for Nodes

CAN protocol completes address coding for messages, facilitates the addition and deletion of notes on the bus. As each node can only communicate with the main controller, therefore CAN message frame ID code is utilized in this paper to unify the identification code of transmit frame, the node and of receiving filter at the node, for instance, the message sent by No.01 node has the message identification code 01, receiving the message with identification code 01 only. In addition, no message filtering is provided at the main controller. It is able to determine which node the message comes from by reading the ID of the received message, with the communication strategy used as shown in Figure 4. Such a design has in fact converted the bus topological structure into the star-like topological structure, facilitating the management and control of all nodes.



(a) (b)

Figure 3. Result of Continuous Read Operation (a) without Delay (b) with Sufficient Delay

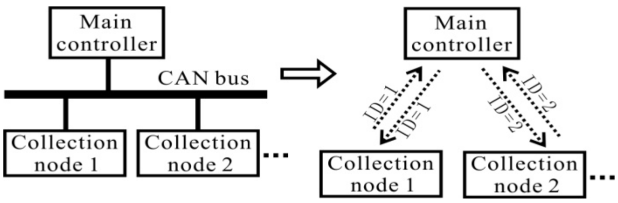


Figure 4. Communication Strategy

Unlike many MCUs, C54 series DSP has the minimum length of storage cell of 16-bit [5]. When the 8-bit CAN protocol is used for data transmission, the necessary data cutting will be required. Taking the floating-point number as an example, the floating-point number is stored by using 32-bit data, accounting for 2 memory cells in DSP, and 4 CAN controller data cells are required for storage, as shown in Figure 5. Therefore, the higher 8 bits and lower 8 bits data should be selected respectively, meanwhile recombining the data at the receiving terminal into 32-bit floating-point number.

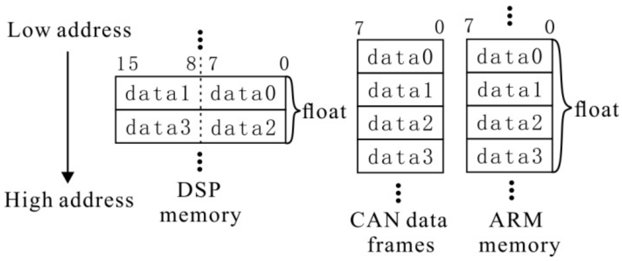


Figure 5. Floating-point Transmission Mode

4. Design of Main Controller Node

Relative to collection nodes with simple tasks like acquisition and processing, the main controller has more functions, including monitoring node, receiving and dispatching data, storage, human-computer interaction and communicating with the host computer, which is quite complex if single-task MCU is used, while becomes relatively simple for the multiple task operating system. Therefore, embedded microprocessor ARM11 is selected as the hardware platform of main controller. ARM with rich peripheral interfaces and ideal real-time control performance is superior than the ordinary single chip machine in the addressing space and the processing performance, ideally served as the small control center.

4.1. CAN Network Device Driver under Linux

The kernel of Linux 3.0.1 version is transplanted to the system as the system platform for software operation, which will provide a uniform interface for driver. CAN equipments are not simply defined as the character device but the network equipment in the kernel, more similar to the fact that CAN is a LAN. Therefore, Socket may be used to operate CAN equipment, also the high-class networking protocol and queue provided by the kernel can be used.

In order to implement and realize the hardware platform, the drive file MCP2515 provided with kernel should be modified. All of the equipments are under management by defining the mcp251x_priv structure, as shown in Figure 6.

```
struct mcp251x_priv {
        struct can_priv     can;
        struct net_device *net;
        struct spi_device *spi;
    ……
};
```

Figure 6. The mcp251x_priv Structure

The driver module of network equipment includes four layers: network protocol interface layer, network device interface layer, device driver function layer and the network device and medium layer [6]. See Figure 7 for the structural relationship of each module in the kernel. The hardware layer operation is achieved by using SPI subsystem. The bridge between the network device and SPI device is the driver function layer, by which the underlying hardware operation-interface function of network device is provided to the upper layer.

When loading the driver module, the initialization function calls up the registration function to perform registration and initialization operation for 2515's driver. Initialization operation will not allow for interrupt application, since it would waste the limited interrupt resources, unfavorable for the realization of interrupt sharing mechanism [7].

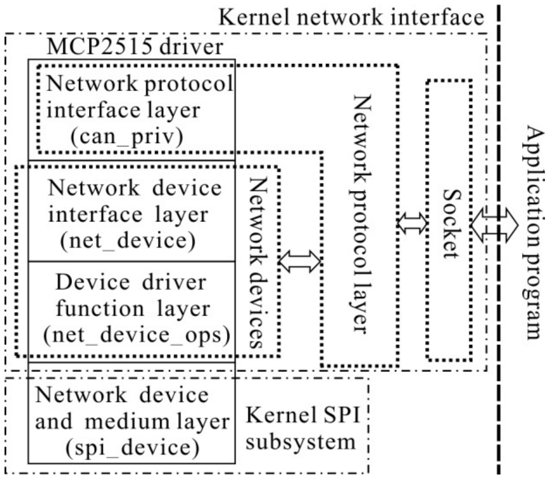


Figure 7. Driver Structure Relation Diagram

Both the "top half" and "bottom half" mechanism is used to handle with the interrupt and solve the time-consuming problem encountered while read-writing SPI in the interrupt. Such mechanisms allow the interrupt program taking a long time to operate in a safer period so as to let the system timely response to other interrupt request [7]. Therefore, the work queue of processing function in the interrupt function is only awakened, while allow the true processing program operate in the work queue to ensure the real-time performance of system.

For operating the driver, the necessary hardware information should be provided for the kernel, therefore such hardware platform information as the oscillator frequency, SPI bus number, hardware interrupt number should be registered in the kernel, among which the device name must be identical to that defined in the driver, or the kernel would be unable to transmit such information to the driver.

4.2. Development of Control Application Program Based on Qt

Application program in the main controller is critical for realizing all control functions, also the human-computer interaction platform. A good application framework can provide desirable human-computer interaction performance, therefore, the cross-platform C++ application development framework Qt is utilized in this paper to develop graphic user interface program because of its ideal adaptability and portability.

Qt library of 4.8.0 version is transplanted in the system, as shown in Figure 8. The major application framework is constructed by using window widget class QWidget. In addition, to make the interface more intuitive, the program is designed to support the label form class QTabWidget. Meanwhile, Socket CAN communication is realized by multi-thread since the multi-thread mechanism will not affect the response of user interface while Qt program is operating several blocking or repetitive tasks. The child thread class RecvThrd is mainly responsible for monitoring CAN port, transmitting data to the main thread via signal- slot mechanism, and the new thread task may be loaded just by rewriting virtual member function. Similar to other network protocol operations, Socket CAN supports such major operations as socket establishment, binding port and monitoring, while the difference between them is that CAN communication requires no connection with the partner, with major implementation model as shown in Figure 9.

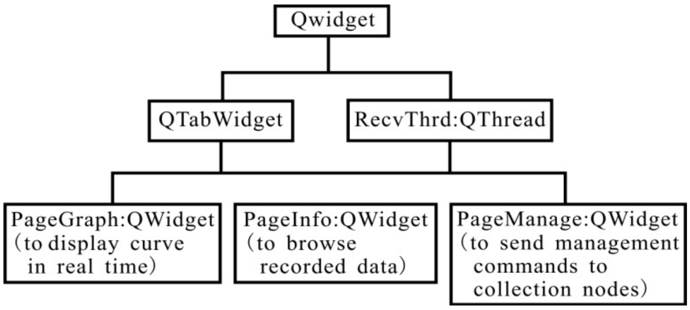


Figure 8. Application Framework Structure

```
void RecvThrd :: run( )
{
            rv_sk=socket(PF_CAN,SOCK_RAW,
                            CAN_RAW);
            recv_addr.can_family = AF_CAN;
            recv_addr.can_ifindex = 0;
            bind(rv_sk, (struct sockaddr *)&recv_addr, len);
            recvfrom(rv_sk, (void *)&recv_frame,
              sizeof(struct can_frame), 0,
              (struct sockaddr *)&recv_addr, &len);
       ……
}
```

Figure 9. Major Implementation Model of Socket CAN

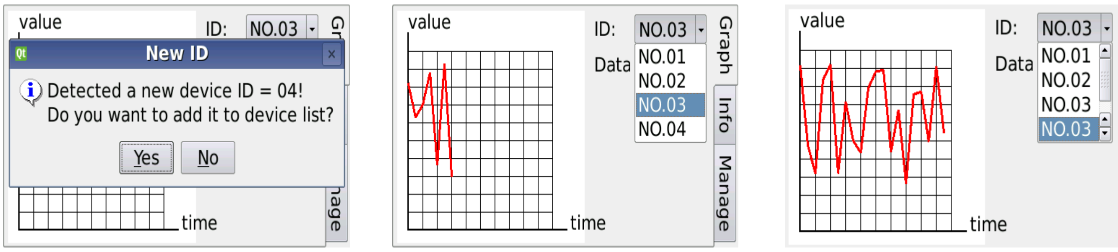
As shown in Figure 10(b), to display the real-time dynamic data curve and coordinate in the child window Graph, the drawing class QPainter should be used. To draw the coordinate curve, change the position of the origin of coordinates first by using the window conversion function of QPainter, since the default position of the origin of drawing coordinates is located at top left corner of the screen, not in line with most users' habit. The dynamic curve, which is drawn with broken lines, will make a left-ward shift when its width beyond a certain value. This paper is not intended for some specific data, therefore, the scale and unit of coordinate is not defined in the figure, just for reference.

In the network, the collection node number is subject to the management of window widget QComboBox. The nodes can be dynamically added to the list by using the member function addItem. Since the specifically defined data type is used in Qt, the conventional data type used in C must be converted before used and displayed by Qt.

Qt mainly uses QFile for file management and stream operation QTextStream for read-write. The reason for not read-write files by directly using system call is that using system calls frequently would increase the program running time, while the stream operation can both effectively reduce the number of system calls and improve running speed. In addition, reading files by using stream operation can prevent occupation of substantial memory by the program.

5. Testing and Result

After cross compiling, Qt application can be downloaded to ARM platform to generate the tab window as shown in Figure 10(b). A message box as shown in Figure 10(a) will pop out when the main thread detects out new collection nodes. According to the user instruction, the program will select and add the new node "NO.04" to the device list of each child window, as shown in Figure 10(b), or being abandoned. Failure to make command processing in time would lead to the condition as shown in Figure 10(c). The program detected out new node in several times and then added the same ID number. The introduction of mutex mechanism in Qt will effectively prevent such case.



(a) (b) (c)

Figure 10. New Node Processing

After clicking Info interface, it is allowed to to browse recorded data, select the corresponding sampling node and check box, also view the data stored at different nodes in SD card, as shown in Figure 11.

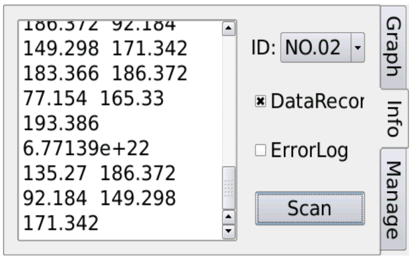
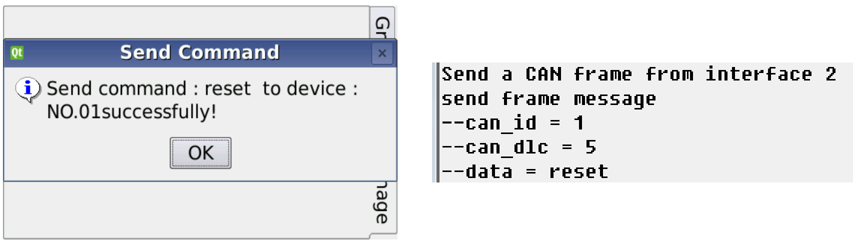


Figure 11. Data Browse

The management interface Manage can be used to send the management command to the corresponding seismic collection node by constructing a new Socket CAN. Figure 12(a) displays the feedback information after successful sending of commands, and Figure 12(b) shows the actually sent CAN data frame information outputted via serial port, both are consistent with each other.



(a) (b)

Figure 12. Send the Command

The test shows that, the main controller based on ARM platform is capable of performing several tasks such as data receiving and dispatching, real-time display, nodes monitoring and anomaly record. It can effectively perform human-computer interaction and real-time control without affecting its communication with seismic collection nodes, serves as the relay station in the whole of network system, thus facilitating the control and management by the mobile station of scattered seismic collection nodes as well as the data acquisition and summarization.

6. Conclusion

In this paper, a data transmission mode between seismic collection nodes and mobile station is designed by utilizing CAN bus communication technology. By taking such characteristics of CAN bus communication as long transmission distance, high anti-interference ability and easy wiring construction as well as of the excellent real-time control capability and good communication capability of microprocessor ARM, main controller nodes are constructed between seismic collection nodes and mobile station, and the collection nodes scattered in a certain scope are under centralized control in the form of star topology. Meanwhile, Linux multi-task system platform on ARM hardware platform is constructed. Human-computer interaction interface is established by using embedded application framework Qt. The ideal real-time control and management function are performed. The seismic data transmission mode based on CAN bus and embedded technology facilitates the construction and expansion of network in field environment, also helps reduce the management task of mobile station.

Acknowledgements

This work was financially supported by the Natural Science Foundation of China (61271274), the Natural Science Foundation of Hubei Province (2012FFB6412), the Fundamental Research Funds for the National University, China University of Geosciences (Wuhan) for the Excellent Youth Scholars (CUGL100222) and the Characteristic Discipline Team Project (CUG120119), which I herewith acknowledge with best thanks.

References

[1] Wand Chen, Teng Yun-tian, Wang Xiao-mei, Zhang Lian, and Fang Xiao-yong. The wireless networking system of earthquake precursor mobile field observation. *Progress in Geophysics*. 2012; 27(6): 0945-0949.
[2] Guo Cheng-zhen, He Di. Design and implementation of PLC communication network based on CAN bus. *Communications Technology.* 2012; 45(7): 51-53, 74.
[3] Liang Jin-zhi. Application of CAN bus protocol in embedded system. *Computer Knowledge and Technology*. 2011; 7(7): 5003-5004.
[4] ADM3054 Data Sheet. Analog Devices, Inc. 2011.
[5] TMS320C54x Optimizing C/C++ Compiler User's Guide. Texas Instruments. 2002.
[6] Song Bao-hua. *Linux device driver development.* Beijing: Posts and Telecom Press. 2010.
[7] Corbet J, Rubini A. *Linux device drivers.* Beijing: China Electric Power Press. 2005.