International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 15 (2015) pp 36061-36064

Remote Firmware Update of Networked Data loggers

Janjanam Aditya

Department of ECE, GMRIT, Rajam, Andhra Pradesh, 532127.
adityajanjanam@gmail.com

M.V.Nageswara Rao

Department of ECE, GMRIT, Rajam, Andhra Pradesh, 532127.
nageswararao.mv@gmrit.org

D.Y.V.Prasad

Department of ECE, GMRIT, Rajam, Andhra Pradesh, 532127.
prasadyv.444@gmail.com

Abstract
In the present scenario, Indian railways are using different embedded systems like data loggers, fault analysis system for the analysis and logging of the different parameters and the status of the station assets. Sometimes these systems need to update the firmware to fix the Bugs and enhance the functionality of the system.
It is very difficult to update the individual system manually, when there is need to update a specific system or a group of systems. There will be an effect on time and economy of the service company, when the systems are located far away from the service location. Time and expenditure incurred can be saved during the update, if the systems are updated from remote location. Here a method is proposed to update the firmware of the data loggers from a remote location when they are connected in a network.

Keywords: Data loggers, Firmware, Remote Update.

1. Introduction.

Now a days embedded systems are playing a very crucial role in the monitoring and controlling in different areas. Indian Railways are also using different monitoring equipment like Data Loggers and fault alarm system for the logging of different parameters and to represent the status of the station assets [1, 2]. As the systems are distributed to the different locations for different customers, if any, features are needed to be added or to fix bugs in existing firmware the service engineer needs to go for that particular system to update its firmware. If the systems are distributed very widely and placed far away from the service center, it is difficult for a service person to attend to each site to update.
The data loggers which are placed in the different stations are connected to the central place called Central Monitoring Unit (CMU). The event information is sent to the central place, From where one can monitor the station assets [2]. At present if any data logger is needed to update its firmware to provide enhancements or to fix the bugs in the existing firmware, either the CPU card of the Data logger is sent to the service center or service engineer has to attend to that Data Logger. Sometimes the systems are tightly coupled and unable to access easily to update its firmware. This process will have an impact on the service center and time of service.
To update the firmware of these data logger from the central place, a novel approach is proposed without disturbing the physical layer of the data logger network. For this, the protocol [2] given by the Research Designs and Standards Organization (RDSO) is modified in such a way that the update can be done from the central place.

2. Data Loggers.

Data Logger is the electronic device which is used in the Indian railways to log the change of status of relays and voltages of analog channels connected to it [2]. Data logger system is helpful in monitoring Railway Signal Control and interlocking relays in order to verify their operation, diagnose faults and maintain.
Data logger transmits event information to the central place to generate various exceptions, reports and other information with the help of application software called Network Management for Data Logger (NMDL) [1]. This can be achieved by connecting all the data loggers to the central place as shown in Fig1.

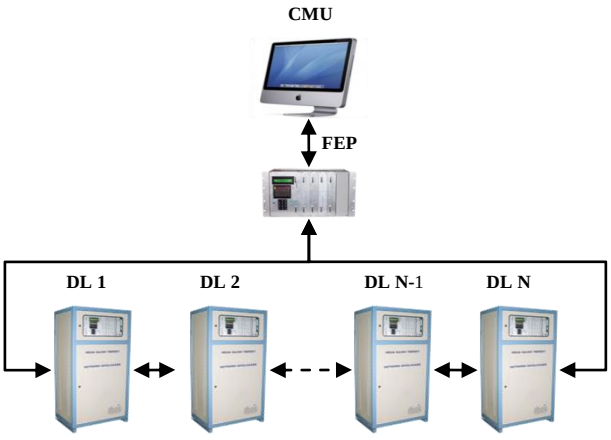

Figure 1. Data Logger Network

Ring Network topology has been adopted in this network. In ring configuration the data loggers are connected serially one to another. If 'N' number of Data loggers are in the network, then the first and Nth data loggers will be connected to the Front End Processor (FEP) to make the network as closed ring. Each Data logger is connected with its neighbor Data logger in either direction with 4 wire leased line modems with baud rate of 57600. FEP is connected to the first and the last Data Logger through the modems, so the event data can be placed into network immediately and simultaneously by all the Data loggers.

Two types of message formats are given by the RDSO to communicate between any two devices. They are event packets and commands.

The event packets will have the information of station assets or the status of the data logger. The event packets created in the Data Logger will be sent immediately in both the direction. The event packets received from one direction is retransmitted to the other direction. Here the sending Data Logger should have to get the acknowledgement from the receiver (adjacent Data Logger) within a specified timeout; otherwise the sender will retransmit the event packet.

Commands always originated from the CMU to facilitate control actions on any one or all the Data Loggers in the network. The commands always work in the Request and Response style.

3. Present Firmware Update Procedure.

At present if any data logger is needed to update its firmware to provide enhancements or to eliminate the bugs in the existing firmware, the CPU card of the Data logger is sent to the service center or any service engineer has to come to that Data Logger. Due to the involvement of the human, travelling time may take from few hours to days from one place to other, service time will take from a few hours to days. This is more than the time of servicing system.

In the following section a novel method is explained to update the firmware of the Data Logger from the remote location called CMU without attending to the each site.

4. Proposed Firmware Update Algorithm.

Here the target system which is needed to be updated with the new version of the firmware is designed with the LPC1778 microcontroller. This microcontroller have the 512KB on-chip Flash.

In this method the Hexadecimal data related to the new version of firmware is transferred to the target system by using the command format given by the RDSO. According to the RDSO specifications one can send the 65540 bytes in a single command. Here for the Supportability of buffers maintained in the DL, a maximum 128 bytes are transferred in a single command.

The processing diagram for Remote firmware update of Data Logger is shown in Fig2.

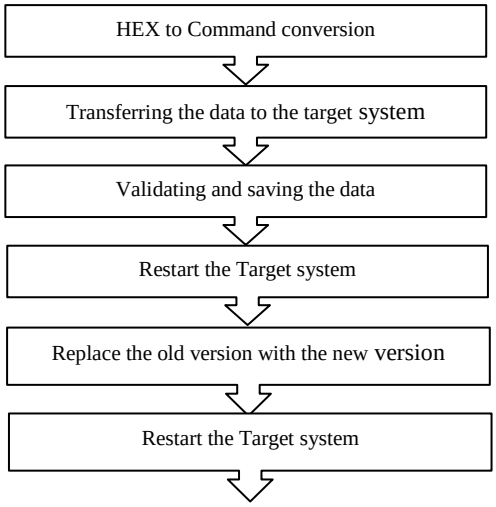

Figure. 2. Processing diagram for firmware update.

In the first step, HEX data related to new version is got from the Integrated Development Environment (IDE) that we used to develop the application. Here Keil 4.74 is used for firmware programming. The HEX data that needs to be programmed in the Data logger is got from this IDE, which is in the format of Intel HEX file format.

The Intel HEX file got from the Keil IDE will transferred to specially designed software called HEX to Command converter. This software can be used to convert the HEX file which got from the Keil IDE will be Converted into RDSO command format.

Segregated images and its checksum will be sent to the target system from the CMU. The Transferring of HEX data to the target system can be achieved by the already existing 4 wired modem network. Any command to the specific data logger can be identified by the Data Logger ID placed in command.

Whenever the target system received an Image and its checksum then it will compare the Checksum of the data with the Received Checksum. The received HEX data will be verified in the three levels. In the first level the individual commands will be verified by the checksum in the every command. In the second level the checksum of one Image is compared with the Received Checksum. In the third level the Code Checksum will be verified with the data. If the verification is failed in the first level then the serial numbers of the failed commands will be sent as the acknowledgment after the second level checking. If the second level verification only failed then the whole image is retransmitted to the target system. Similarly all the images related to the new version will be sent to the target system. Finally a Code Checksum will be sent to the target system to verify all the data.

After verification of the Code Checksum with the stored data the Reset Cause Flag will be stored in the EEPROM and target system is restarted to update with the new version.

Whenever the target system is restarted if the reason for the restart is to update the firmware then the data stored in the new version backup area will be moved to the application area.

Now the restart cause flag will be set to the normal and system is restarted. After restart the system will automatically run with application area code.

The firmware update procedure is divided into the 3 stages:

4.1 Commands Preparation

Intel HEX file is converted into command format by using the specially designed software. This software will segregate the total HEX data into the different small parts with the maximum of 1024 bytes HEX data in the one segregated part called Image. After the 1024 bytes or if the address is changed in the HEX file then this will create a command called Image Checksum. This will be helpful for the data validation for the 1024 bytes. When the total HEX data is converted into the command format, then it will create another command called Code Checksum having the checksum of the total HEX data. This will be helpful for the verification of the total HEX data. All these commands will be stored in the text file.

Each command has different fields like Data Logger ID, Remote Programming Command ID, Address in the flash of the target system where the data need to be store, Image Number of present transmitting data, Packet Number in the Image. From these fields, data logger system can easily be understand the command belongs to which data logger, these will be very helpful to reduce the possible error occurred in the network.

4.2 Commands Transmission

The commands are transmitted to the target system by the already existing 4wire network and these are initiated by software called NMDL (Network Management for Data Logger) which is placed in the CMU.

The prepared commands are transmitted into network till the completion of one Image and its Checksum. After transmitting the one Image NMDL waits for the Acknowledgment. If it received Image Checksum Success Command, then the NMDL starts sending the next Image. If it received Fail Command and Failed Commands serial numbers then the NMDL retransmits the Fail Commands.

This process will continue until the end of the HEX data. After sending all the images, Code Checksum will be sent to the target system and waits for the Acknowledgment. If it receives the Code Checksum Success Command Update Command will be sent to the target system to update with the new version Otherwise the HEX data retransmission will start.

4.3 Updating with New version

The Flash memory (512KB) of the target system is divided into the three parts.

4.3.1 Secondary Boot Loader area (12KB):

The Secondary Boot Loader will organize the different operations when the system is restarted. When the system is restarted, the secondary boot loader first checks the reason for the reset which can be done by checking the "Reset Cause Flag" stored in the EEPROM.

If the reason for the reset is to update the firmware then the data stored in the new version backup area is copied into the application area. Then the Reset Cause Flag will be set to Normal Reset.

If the reason for the reset is Normal Reset then the system is mapped to the application area to run with application.

4.3.2 Application area (244KB):

In the Application area, the user application will be stored. Remote firmware update routines will be added in this user application, to handle the remote firmware update related command while the application is running.

4.3.3 New version Backup area (256KB):

When the target system received the data related to the new version, then it will store the data in this location by considering it as the image of the application area. The data in this region will be copied to the application area while updating firmware.

5. Testing-Results

Here a module designed with the LPC1778 microcontroller is taken as the target system. For the testing of algorithm the target system is connected in a network of Three Data loggers, and is connected as the 3rd data logger. Here the basic functionality of the target system is to monitor the status of the relays connected to it. The monitored information is transferred to Central place by using the event record format given by RDSO. Here the CRC placed in the event record will be generated by the microcontroller.

For the testing of the algorithm the same functionality is maintained in the newer version but the CRC is generated based on the lookup table. So here the HEX data related to the new version is got from the Keil IDE in addition with the CRC table. This HEX file is converted to the command format by using the HEX to Command Converter Software.

Then commands in the command file are transmitted image by image into the network with the help of the NMDL. Here the test was conducted to transmit the data size of the 76 KB. In addition with HEX data command fields are transferred to validate the commands. This will lead to transfer the data of 89KB. The test report of the remote firmware update is given in the table1.

Table 1. Time for the different operations

Operation	Data size (bytes)	Time (m sec)
Image transfer, Acknowledgment	1276[a]	936
Total HEX data transfer[b]	91872	67400
Writing flash while application is running	1024	5
Flash writing[c]	73728	792
Firmware update with new version	73728	69232

a. 1024 bytes of image, 3 bytes of check sum, header and validation fields forms a total of 1276 Bytes.

b. Here the total HEX data means that the data related to the new firmware version.

c. *Copying the data from the new version backup area to the application area.*

6. Conclusion

In the previous firmware update procedure, the time to update the firmware takes a few hours to days which depend on the distance of the target system to the service center. In the proposed method the HEX data related to the new version will be sent to the target system in short time i.e fewer minutes, which intern helps to reduce the update time of the firmware to fewer minutes.

From the test result, the time to update the data logger which is placed at the 3rd place from the CMU is taking nearly 70seconds. If the data logger is near to the CMU the time taken for the update will be reduced because of the time taken at each data logger for processing of the command will be reduced.

By using this method we can reduce the time to update firmware without attending to each site. The latest release of the firmware version can be installed automatically without updating manually for each time whenever the new version is released. Auto update provision can be provided by adding some routine in the secondary boot loader and the main application code.

7. References

[1] www.efftronics.com/images/downloads /1Data%20Logger%20System.pdf

[2] http://122.252.243.98/Departments/snt/Signal% 20Policies/HQ%20NWR/data%20logger%20RDSO. pdf

[3] http://en.wikipedia.org/wiki/Data_logger.

[4] "The Art of Programming Embedded Systems" chapter-File format page257

[5] www.nxp.com/documents/user_manual/ UM10470.pdf