Transactions of the Korean Nuclear Society Autumn Meeting
Yeosu, Korea, October 25-26, 2018

Data I/O Module Design Methodology for Integrated Radiation Monitoring System Software

Sung-Ho Lee [*], Jong-Su Kim, Oh-Young Kwon, Seung-Eun Jung
Korean Atomic Energy Research Institute (KAERI), 989-11 Daedeok-daero, Yuseong-gu, Daejeon, Korea
[*]*Corresponding author: leesungho@kaeri.re.kr*

1. Introduction

Radiation monitoring system (RMS) devices provide the measured radioactivity data in the workspace or facility. For productivity improvement with the data, the software (SW) operation is essential; this contributes better data management, easier user-interface and lower manpower losses.

For accessing data in RMS devices and entering it to a database such as MySQL, MS-SQL, Oracle, a data I/O module is required. The data I/O module is responsible for reading and writing data in a RMS device according to each data communication protocol.

Data I/O modules in the conventional RMS SW provide an interface between a server and devices with only same protocol. However, this makes the server to add other devices with different protocol be impossible.

In this paper, an efficient data I/O module design methodology is presented; this is designed considering the function and purpose of RMS SW. From the design procedure, the data I/O module that can deal with various RMS devices with different protocol can be developed.

2. Analysis of Data Communication Protocol

As shown in Fig. 1, the communication between a server (master) and a RMS device (slave) is started by the request from the master and is terminated by the response from the slave. There is time delay between the master's request and slave's response; it is necessary for data processing according to the request. This chapter presents analysis of data packet for communication between the master and slave. This chapter analyzes the request and response message's data packets for each device; it contributes to figure out how to interface with devices, what kinds of data are transferred from devices, and how to make a data modeling with data provided from individual device with different data communication protocol.

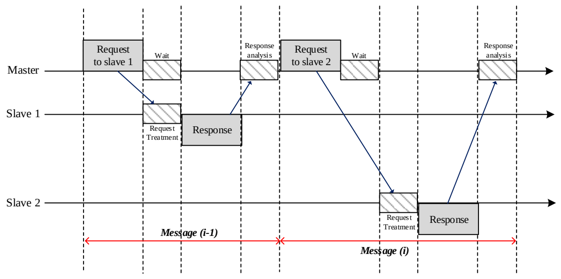

Fig.1. Data communication between a master and slaves.

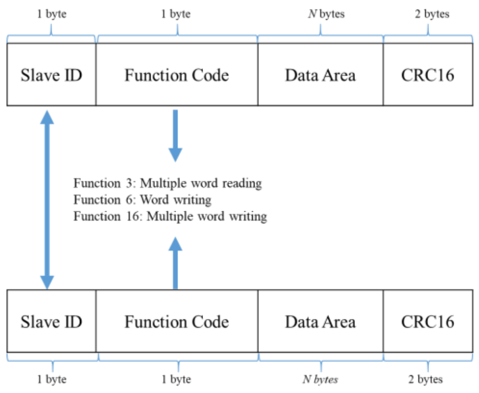

Fig 2. MODBUS RTU data packet form.

2.1. MIRION (MGPI)

MIRION's RMS devices employ MODBUS RTU protocol [1]; MODBUS has widely used in various industrial field. Fig. 2 shows the request and response messages share the same data packet form in MODBUS RTU protocol.

As the "Slave ID" is a kind of identifiers, it is a slave responding to the request by the master. "Function code" is determined according to which function is used. In a request message, data area may include information associated with memory address, the quantity, and detail value of the requested data. On the other hands, in the data area of the response message, it may have reading data value, the quantity of responded data. CRC 16-MODBUS is used to check if an error is in communication process.

2.2 G64 and ICAM by CANBERRA

A gamma monitor, G64 and a continuous air monitor, ICAM have the same data packet as shown in Fig. 3 for data communication. The codes "0x01" and "0x19" means that start and end of a message, respectively. As "Source" and "DEST'N" are as identifiers; they mean a sender and a receiver, respectively. Moreover, "Mno" is a tag of the message. Thus, "Source", "DEST'N", and "Mno" perform as the identifiers in this protocol. G64 and ICAM employ CRC16-ITTT for the error check. In the responded message, it is noted that there is "Transaction code". Although the master requests a certain data to the slave according to the data packet in Fig. 3, the slave transmits an error code if specific parameter such as a memory address of request data in the data area is wrong; the error code is "transaction code".

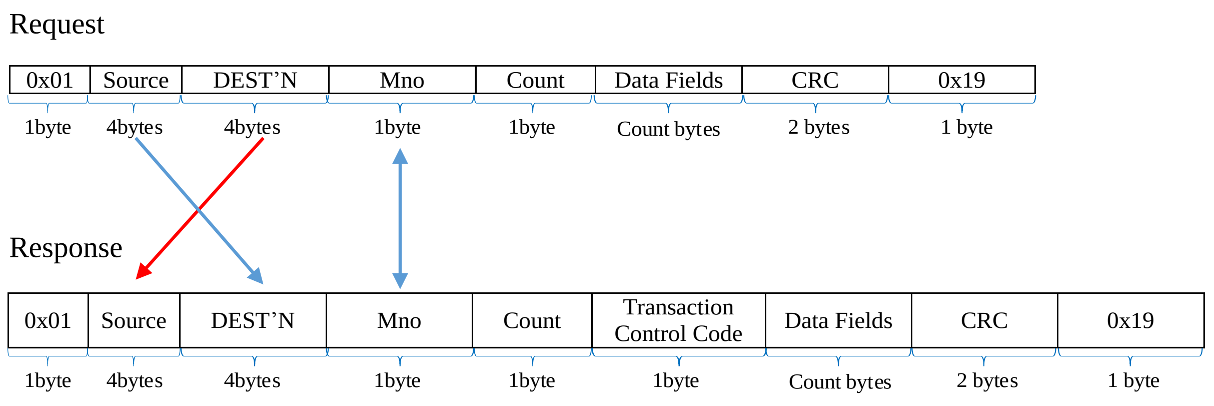

Fig. 3. Data packet used in G64 and ICams.

Transaction ID	Protocol ID	Length	Unit ID	Function Code	Data Area
2 bytes	*2 bytes*	*2 bytes*	*1 byte*	*1 byte*	*n bytes*
MBAP Header					

Fig. 4. Data packet used in DTIonix.

2.3. DTIonix by Premium Analysis

A tritium monitor, DTIonix employs MODBUS TCP; its data packet is similar to that of MODBUS RTU which is used in MIRION's device. "MBAP HEADER" replaces "the slave ID" of MODBUS RTU; It performs as an identifier. MODBUS TCP does not use a CRC algorithm for checking error.

3. Data Modeling and Data Communication Implementation

3.1. RMS' SW Function Definition and Data modeling

Using aforementioned protocols, users can access RMS device's memory and load (or enter) data such as the measurement value, device's status, alarm level, calibration factor, device control command, efficiency. However, for a suitable operation, easy maintenance, high security, the SW function definition is essential. Considering that the purpose on the operation of the RMS SW, the information to load and record to a database in real time are the measurement data and the status of the device. Moreover, for synchronization between the alarm level saved in device and that of a database, the alarm level should be load. However, it is very ineffective to often request the alarm level. So, the request interval for the alarm level should be designed to be long relatively. Table I shows the data model in the RMS SW.

3.2. Data Communication Implementation

Fig. 5 shows the overall diagram for KAREI's integrated RMS SW; this can deal with various devices such as MIRIONs, G64, ICAM, DTIonix, and ADM606.

Table I. Essential data model in RMS SW

	Data type	**Request interval**
Measurement value	Float [2]	Short
Status	Byte or Word	Short
Alarm level	Float	Long

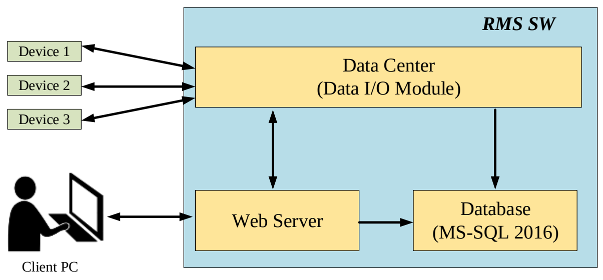

Fig. 5. Configuration for KAERI's integrated RMS SW.

As a database, MS-SQL 2016 is employed. By interfacing with devices, the data I/O module provides the data in table I and enters it to the database.

For high reliability in data communication, the verification of the exchanged data is necessary. As shown in Chapter 2, because all RMS devices have identifier, we can be used as the primary verification tool. MIRION devices CRC16-MODBUS is utilized as the second verification tool. For G64 and ICAM, the use of the transaction code as the secondary verification tool is more effective than the use of CRC16-ITTT. For DTIonix, the secondary verification such as checksum, CRC is not necessary because TCP protocol is used unlike other devices. Fig. 6 shows the flow chart for a data communication process from a request to entering data to a database.

4. Conclusions

In this paper, the data I/O module's design methodology for developing integrated RMS SW is presented. The design aims at a reliable operation, efficient maintenance and includes communication protocol analysis for each device, data modeling, and data verification method. From each device protocol

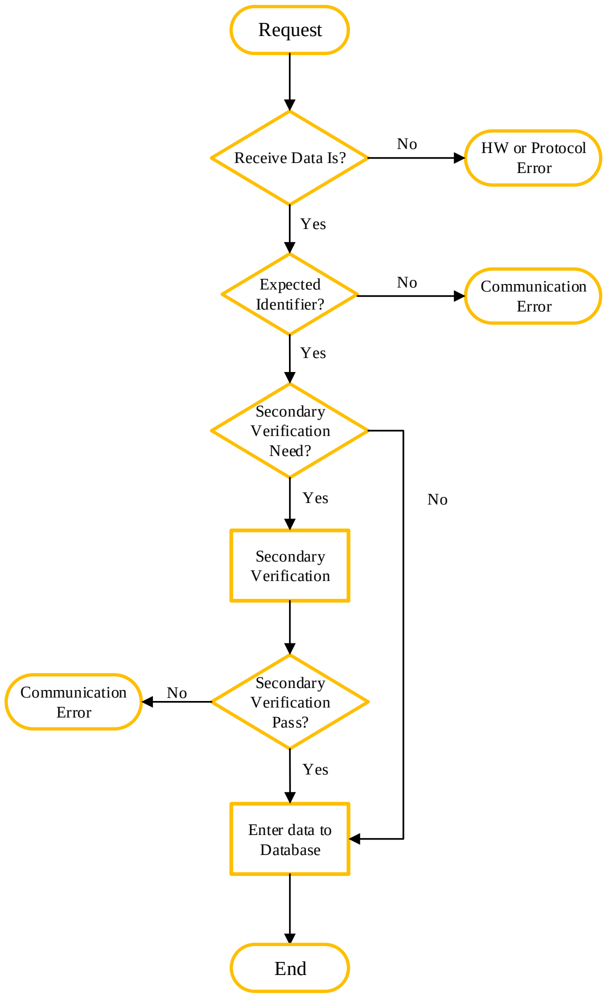

Fig. 6. Flow chart for a data communication process from a request to entering data to a database.

analysis, the available information and error check function can be figured out. Moreover, the data modeling considering the SW function makes the RMS SW operate with high efficiency and low cost. Finally, from the design methodology, the data I/O module for integrated RMS SW can be developed effectively.

REFERENCES

[1] http://www.modbus.org.
[2] *IEEE Standard for Floating-Point Arithmetic*, IEEE Standard 754-2008, Aug. 2008, pp. 1-70.