International Journal of Control and Automation
Vol. 9, No. 6 (2016), pp.201-208
http//dx.doi.org/10.14257/ijca.2016.9.6.19

Implementation of Protocol Conversion Control Board for Industrial Communication

Sang-hee Eum

Department of Ship Building Marine, Dongju College, Busan, Korea.
nyx2k@naver.com

Abstract

Recently, many industrial instruments faced the problem of protocol compatibility with the external monitoring and control system. This is due to different environments, field bus devices and protocols. Therefore, it needs to deal with the problem on how to communicate with each other between different protocols. For this purpose, this study is implemented in the main control board and sub-communication board to support the industrial communication protocol conversion. The sub-communication board is designed to ensure the communication connection of CAN bus and Ethernet. The ATmega 2560 microcontroller is used as the processor of the main control board and 4 RS485 serial slots for sub-board are placed too. One of those is used to communicate CAN bus and Ethernet. The sub-communication board is consisted of the ATmega 128 microcontroller, CAN Transceiver and Ethernet connecter. The performance test results have successfully showed the transmission and conversion.

Keywords*: Protocol Conversion, CAN, Ethernet, Industrial communication*

1. Introduction

The communication methods and protocol in the industrial communications depend on the environment and hardware configuration specification of the industrial site. There are many difficulties in the development of industrial monitoring and control systems because of various communication methods such as field bus devices and protocols. It is also necessary to repeat the system hardware analysis and communication protocol analysis operations every time for this purpose. Therefore, it needs to deal with the problem on how to communicate with each other between different protocols [1].

Currently, due to various kinds of field bus, it is not of great value to develop only one bus conversion device. It has become a trend for development of multi-protocol switching platforms. Between the application layer and link layer protocol, it increases a uniform layer, which provides a software interface for each bus link layer, shields up the specific implementation details and ensures the independence of the layers. The modular design will help reduce time and difficulty of development. In this structure, in order to achieve uniformity and scalability of protocol conversion, there is link layer added on the basis of existing field bus physical layer and protocol layer. It provides a unified communication interface for applications and information so as to play a role on masking differences of the bus. For the top users, the unified protocol provides an interface to access to the bus protocol and shields differences between the underlying buses. Protocol conversion is a process that a variety of bus data is going up layer by layer in the gateway and then drills down to another bus [2].

If the protocol conversion device was applied in the equipment and systems, the cost would be very low and the industry system would have an advantage in easy customization and extension. This paper describes the protocol conversion main

ISSN: 2005-4297 IJCA

control board and sub-communication board. The main control board can be divided into two parts, the main control part for protocol conversion and the slots for sub-boards. The main control board processor is the ATmega 2560 and the serial slot is 4ea RS485 method for sub-communication board. One of those is used to communicate CAN bus and Ethernet. The sub-communication board was designed to ensure the communication connection of CAN bus and Ethernet. The sub-communication board is consisted of the ATmega 128 processor, CAN transceiver and Ethernet connecter. These are connected to the commercial CAN module and Ethernet modules and enables communication.

2. System Description

Many industrial sites have made various communications between the sensors and devices for monitoring and control. Industrial monitoring includes power plants, factories, monitoring and control systems and data logging of manufacture. The processing speed of the device had been fast with a competitive productions increase. A variety of communication methods has appeared because of performance, quality control, as well as monitoring and automation efficiency. These processes are used in different protocols; therefore there is a need for a protocol acquisition and analysis [3, 4]. Figure 1 shows the configuration of the protocol conversion control board that was developed in this study. The board is composed of a main control board and the sub-board. The main control board was designed to expand the communication using sub-board. The RS485 communication was selected for this purpose. If the communication methods are different from each instrument it can also be converted the protocol and then transmission.

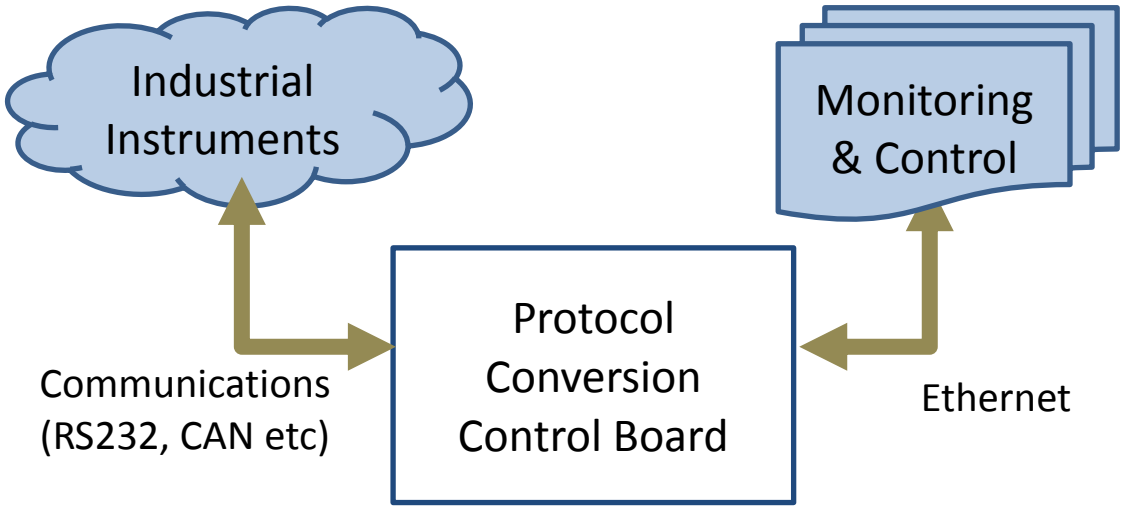

Figure 1. An industrial Network Diagram

2.1. Introduction to the Controller Area Network (CAN)

The controller area network specification defines the Data Link Layer; ISO 11898 defines the Physical Layer. A number of different data rates are defined, with 1Mbps being the top end, and 10Kbps the minimum rate. All modules must support 20Kbps. Cable length depends on the data rates used. The maximum line length is 1 Km, 40 meters at 1Mbps. Termination resistors are used at each end of the cable. The worst-case transmission time of an 8 byte frame with an 11 bit identifier is 134 bit times. The CAN Bus interface uses an asynchronous transmission scheme controlled by start and stop bits at the beginning and end of each character. This

interface is used, employing serial binary interchange. The data frame is composed of an Arbitration field, Control field, Data field, and CRC feedback field. The frame begins with a 'Start of frame(SOF)', and ends with an 'End of frame(EOF)'. The receiving node synchronizes itself with every start bit of the message. CAN uses a technique called Bit stuffing to have the proper communication of the messages. After every consecutive bit levels, the transmitter will automatically stuff a bit of opposite polarity into the bit stream. The receiver of the message will automatically delete this stuff bit. So if any receiving node detects six consecutive bits of the same level, then a stuff error is flagged. Theoretically, for every 5 bits a stuff bit is sent by the sender. On an average, every message has 3 to 4 stuff bits. The CAN controller of every receiving node checks the message format and also the checksum. It sends an acknowledgement or Error frame within the Acknowledge and EOF fields in the frame. In case of an error frame, the received message is ignored by all the nodes in the network. The sender will automatically try to send the message later on the bus. The nominal bit timing is the time needed to transmit a bit across the network. All the nodes in the network should have the same bit rate. That is why the synchronization is being done in several ways. The maximum transmission time for a message with 8 data bytes and of higher priority is 225 micro seconds and with 29 bit identifier is 260 micro seconds [5, 6].

2.2. Introduction to Ethernet

The Ethernet was developed by Xerox Corporation's Palo Alto Research Center(PARC) in the 1970s. Ethernet was the technological basis for the IEEE 802.3 specification, which was initially released in 1980. When it was developed, Ethernet was designed to fill the middle ground between long-distance, low-speed networks and specialized, computer-room networks carrying data at high speeds for very limited distances. Ethernet is well suited to applications where a local communication medium must carry sporadic, occasionally heavy traffic, at high peak data rates.

Ethernet frame begins with an alternating pattern of ones and zeros called a preamble. The preamble tells receiving stations that a frame is coming. The byte before the destination address in an Ethernet frame is a start-of-frame (SOF) delimiter. This byte ends with two consecutive one bits which serve to synchronize the frame reception portions of all stations on the LAN. Immediately following the preamble in Ethernet is the destination and source address fields. Ethernet address is 6 bytes long. Addresses are contained in hardware on the Ethernet interface cards. The first 3 bytes of the addresses are specified by the IEEE on a vendor-dependent basis, while the last 3 bytes are specified by the Ethernet vendor. The source address is always a unicast (single node) address, while the destination address may be unicast, multicast (group), or broadcast (all nodes). The 2-byte field following the source address is a type field. This field specifies the upper-layer protocol to receive the data after Ethernet processing is complete. After the physical-layer and link-layer processing is complete, this data will eventually be sent to an upper-layer protocol. The upper-layer protocol is identified in the type field. If data in the frame is insufficient to fill the frame to its minimum 64-byte size, padding bytes are inserted to ensure at least a 64-byte frame. After the data field is a 4-byte FCS field containing a cyclic redundancy check (CRC) value. The CRC is created by the sending device and recalculated by the receiving device to check for damage that might have occurred to the frame in transit [7].

2.3. Overview of RS485

RS485 is used in situations with a severe ground level shift of several volts, where at the same time high bit rates are possible because the transition between logical 0 and logical 1 is only a few hundred millivolts. RS485 receivers with an input resistance of 12 kΩ can connect 32 devices to the network. Currently, the available high-resistance RS485 inputs allow this number to be expanded to 256. RS485 repeaters are also available which make it possible to increase the number of nodes to several thousands, spanning multiple kilometers. For higher speeds and longer lines, the termination resistances are necessary on both ends of the line to eliminate reflections and we can use 100 Ω resistors on both ends. How does RS485 function in practice? Default, all the senders on the RS485 bus are in tri-state with high impedance. In higher level protocols, one of the nodes is defined as a master who sends queries or commands over the RS485 bus. All other nodes receive these data. Depending on the information in the sent data, zero or more nodes on the line respond to the master. In this situation, bandwidth can be used to almost 100%. There are other implementations of RS485 networks where every node can start a data session on its own. Because there is a chance of data collision with this implementation, theory tells us that in this case only 37% of the bandwidth will be effectively used. With such an implementation of a RS485 network it is necessary that there is error detection implemented in the higher level protocol to detect the data corruption and resend the information at a later time [8].

2.4. The Software Structure of Protocol Conversion

The different hardware systems and devices used in industrial networks are using a number of communication methods and protocols. The structure of the protocol conversion is largely divided into the communication method analysis and data protocol conversion. In this study, the authors used a protocol conversion process as shown in Figure 2. It is composed to be performed by firmware. First, the input communication data through the CAN communication and the Ethernet are performed by a protocol analysis process. The data determines the type of protocol by the protocol filtering and it is divided into address, data and error checking code, *etc*. as frame format. The data parsing procedure is processed by communication syntax. Next, the parsing data are extracted and converted to the CAN protocol or the Ethernet protocol. This conversion data is available to use as monitoring and control by user.

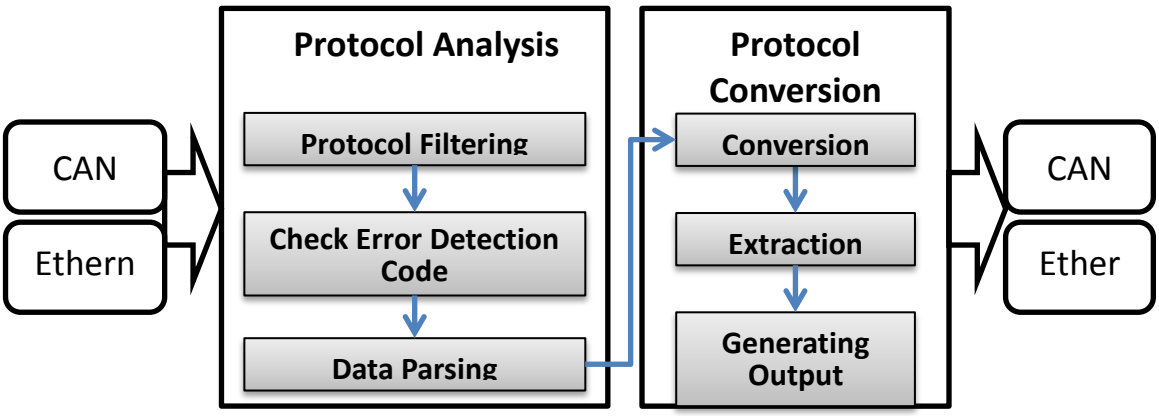

Figure 2. The Flowchart of Protocol Conversion Program

3. Hardware Design

3.1. ATmega 2560 microcontroller introduction

The ATmega 2560 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, this microcontroller achieves throughputs approaching 1 MIPS per MHz allowing the designer of the system to optimize power consumption versus processing speed. The Atmel® AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

This microcontroller provides the following features: 64K/128K/256K bytes of In-System Programmable Flash with Read-While-Write capabilities, 4Kbytes EEPROM, 8Kbytes SRAM, 54/86 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), six flexible Timer/Counters with compare modes and PWM, four USARTs, a byte oriented 2-wire Serial Interface, a 16-channel, 10-bit ADC with optional differential input stage with programmable gain, programmable Watchdog Timer with Internal Oscillator, and an SPI serial port, IEEE® std. 1149.1 compliant JTAG test interface is also used for accessing the On-chip Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interruption or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run [9].

3.2. The Hardware Architecture

The hardware architecture of the protocol conversion control board is shown in Figure 3. This mainly includes two parts, the main control board and the sub-communication board. The main control board module exists to support the industrial communication protocol conversion. The microcontroller in the main control board is used as ATmega 2560 using the Dip switch to select a sub-slot that supports various communication methods. The RS485 is applied to communicate between the main control board and the sub-board. For the expansion of I/O, 20 pins are placed to the I/O connectors. The serial communication protocol between host and control board for interpretation and application upload is also supported by RS485. The sub-communication board was designed to ensure the protocol conversion of CAN bus and Ethernet. This board is consisted of the ATmega128 microcontroller, CAN transceiver and Ethernet connecter. These are connected to the commercial CAN module and Ethernet modules and enables communication.

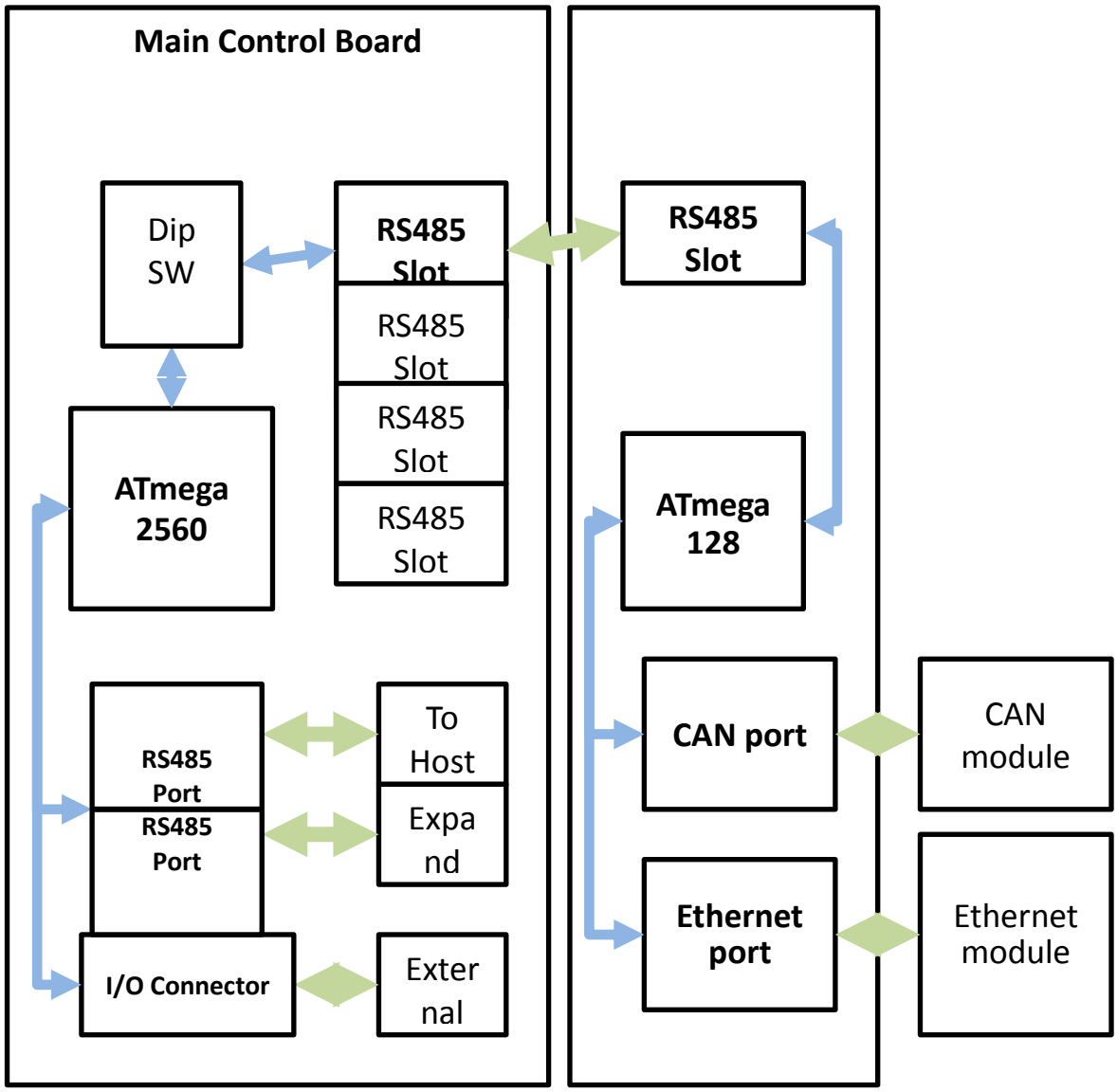

Figure 3. The Architecture of Protocol Conversion Control Board

4. Implementation

Figure 4 shows the developed main control board for protocol conversion and sub-communication board. The microcontroller of the main board is placed on the back side. The center of the top area has four RS485 slots for sub-board, power outputs, the ISP(in system programming) port and two RS485 ports : one is to communicate with Host and another is to expand RS485 slot, are placed in the bottom edge. The I/O ports are on both sides of the board for the check and the communication expansion. The right side image of Figure 4 shows the sub-communication board. The CAN connector is on the top left, and Ethernet connecter is at the bottom of the center in this sub-board.

This board can operate in two ways: one of it can receive the CAN bus and convert to the Ethernet frame and transfer to the computer; the other one can receive Ethernet and convert to the CAN bus format and transfer to another device. The specifications of the development boards as follows:

- The main control board for protocol conversion

1. Main Power : DC24V 3A

2. DC/DC Converter Output : 5V 3A, 15V 0.35A
3. MCU : ATmega 2560 Microcontroller
4. Input Port: 40pin for external data input
5. I/O Card Slot: slot for data input and output modules RS485 4CH
6. ISP Download: 1CH MCU ports for MCU
7. Dip switch : I/O Card slot selection switch
8. LED : State display

- The sub-communication board

1. MCU : ATmega 128 Microcontroller
2. Communication : Ethernet and CAN connecter
3. ISP Download: 1CH MCU ports for MCU

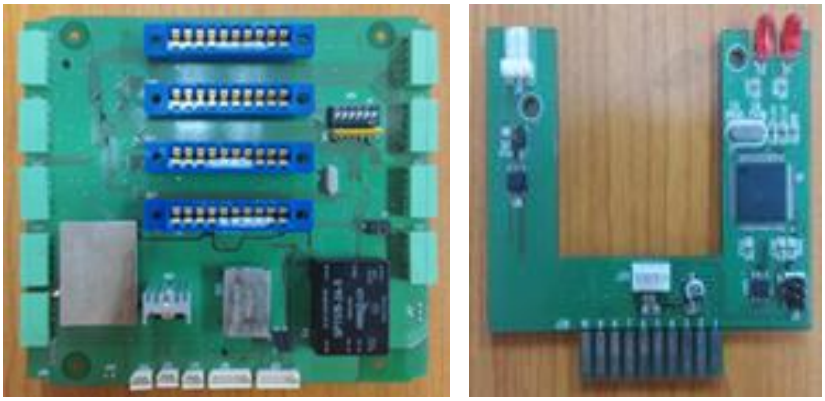

Figure 4. The Developed Main Control Board And Sub-Board

The transmission performance tester was made to check the reliability of the developed protocol conversion control board. This program can calculate the error rate from the state of the send and receive data. The 3% error rate is shown in Figure 5 when receiving the 1000 data sent from the CAN to the Ethernet.

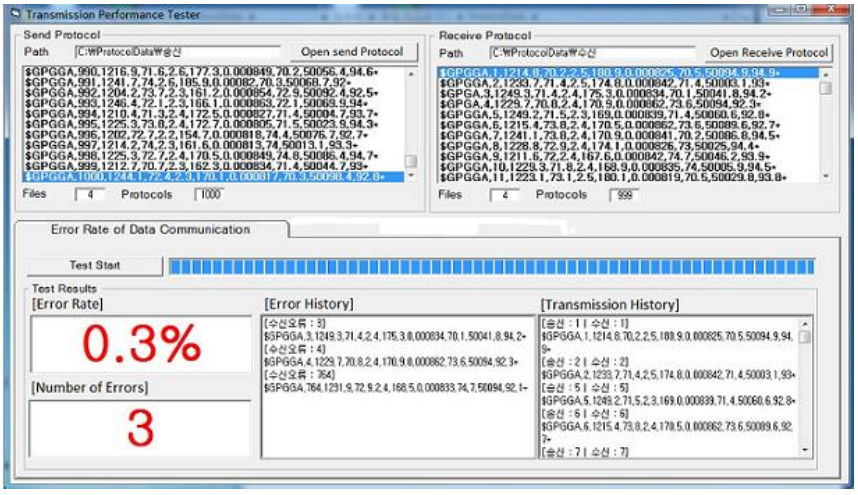

Figure 5. The Transmission Performance Test Program

5. Conclusions

This paper was implemented for the protocol conversion control board using an ATmega 2560 and ATmega128 microcontroller for industrial communication based on the CAN bus and Ethernet. This board has been designed in order to obtain the low cost and easy customization and extension. The program can make not only the realization of the protocol conversion but also generation of a user protocol. The testing results show that the control board can realize the conversion and the transmission in CAN bus and Ethernet.

Acknowledgments

This research was supported by Dongju College (The research year of professor in 2012).

References

[1] F. Zhang, Y. Zhu, C. Yan, J. Bi, H. Xiong and S. Yuan, "A Realization Method of Protocol Conversion Between Modbus and IEC61850", Open Journal of Applied Sciences, **(2013)**, pp.18-23.
[2] H. Zhang, G. Xu, Y. Lu and G. Lou, "Research on the Field Bus Protocol Conversion Gateway", Journal of Convergence Information Technology(JCIT), vol. 7, no. 15, **(2012)**, pp. 160-168.
[3] S. H. Eum and S. K. Hong, "Development of User Protocol Converter about Modbus and NMEA0183", Journal of the Korea Institute of Information and Communication Engineering(KIICE), vol. 19, no. 11, **(2015)**, pp. 2584-2589.
[4] Y. Zhang, X. Feng and Y. Guo, "Design of Ethernet-CAN Protocol Conversion Module Based on STM32", International Journal of Future Generation Communication and Networking Advanced Science and Technology Letters, vol.7, no.1, **(2014)**, pp. 89-96.
[5] G. S. Kumar, "Designing and Development of a CAN Bus Analyzer for Industrial Applications Using ARM and PIC", International Journal of Computer Science and Information Technologies, vol. 3 (2), **(2012)**, pp. 3749-3753
[6] Texas Instruments, "Introduction to the Controller Area Network (CAN)", SLOA101A–August 2002–Revised July 2008.
[7] http://srohit.tripod.com/Ethernet.pdf.
[8] V. Sangeetha, V. Vinothini, R. Suganya, R. Rajaprabha and S. C. Bhagavathi, "Microcontroller based Modbus Protocol Converter using PIC17C756", International Journal of New Trends in Electronics and Communication, vol. 2, issue. 1, **(2014)**, pp.15-20.
[9] http://www.atmel.com/.

Authors

Sang-hee Eum, is currently an associate professor in Dongju College. He received a Bachelor and Masters of Engineering degrees in Electrical Engineering from the Dong-A University in Busan(Korea) and a Ph. D. in Electronics Engineering at Busan National University in Busan(Korea). His research interests are Signal Processing, Artificial Intelligence, Biomedical Engineering, IT Convergence.

