The Design and Implementation of a Network-based Stand-alone Motion System

Myoung-Chol Cho*, and Jae Wook Jeon**

* School of Information and Communication Engineering, SungKyunKwan University, Suwon, Korea
(Tel : +82-31-290-7237; E-mail: nowwake@ece.skku.ac.kr)

** School of Information and Communication Engineering, SungKyunKwan University, Suwon, Korea
(Tel : +82-31-290-7129; E-mail: jwjeon@yurim.skku.ac.kr)

Abstract: A motion controller has been used variously in industry such as semiconductor manufacture equipment, industrial robot, assembly/conveyor line applications and CNC equipment. There are several types of controller in motion control. One of these is a PC-based motion controller such as PCI or ISA, and another is stand-alone motion controller.

The PC bus-based motion controller is popular because of improving bus architectures and GUI (Graphic User Interface) that offer convenience of use to user. There are some problems in this. The PC bus-based solution allows for only one of the form factors, so it has a poor flexibility. The overall system package size is bigger than other motion control system. And also, additional axes of control require additional slot, however the number of slots is limited. Furthermore, unwieldy and many wirings come to connect plants or I/O. The stand-alone motion controller has also this limit of axes of control and wiring problems. To resolve these problems, controller must have capability of operating as stand-alone devices that resides outside the computer and it needs network capability to communicate to each motion device.

In this paper, a network-based stand-alone motion system is proposed. This system integrates PC and motion controller into one stand-alone motion system, and uses CAN (Controller Area Network) as network protocol. Single board computer that is type of 3.5" FDD form factor is used to reduce the system size and cost. It works with Windows XP Embedded as operating system. This motion system operates by itself or serves as master motion controller that communicates to slave motion controller. The Slave motion controllers can easily connect to master motion system through CAN-network.

Keywords: Stand-alone, Network-based, Motion Controller, CAN

1. INTRODUCTION

A motion controller has been used variously in industry such as semiconductor manufacture equipment, industrial robot, assembly/conveyor line applications and CNC equipment. There are several types of controller in motion control. One of these is a PC-based motion controller such as PCI or ISA, and another is stand-alone motion controller.

The PC bus-based motion controller is currently popular because of improving bus architectures that offer faster speeds and greater throughput. Furthermore, the PC is operated in Windows operating system, and the user can easily control motion controller through GUI (Graphic User Interface) that offer convenience of use. There are some problems in this type of motion controller. The PC bus-based solution allows for only one of the form factors such as PCI, so it has a poor flexibility. The overall system package size is bigger than other motion control system because PC bus-based motion controller always needs PC. Additional axes of control require additional slot, however the number of slots is limited. And in addition unwieldy and many wiring must be snaked out through yet another open expansion slot to connect plants and I/O. The stand-alone motion controller has also this limit of axes of control and wiring problem.

The package size and flexibility problem is important in motion control. To resolve this problem, controller must have capability of operating as stand-alone devices that resides outside the computer. In this paper, we integrate PC and motion controller into one stand-alone motion system. This stand-alone motion system operates by itself or serves as master that communicates to slave devices such as motion control system or industrial I/O. This motion controller can be used as slave device also. A single board computer that is type of 3.5" FDD form factor is used as replacing PC to reduce the system size and cost. This single board computer is developed for small sized embedded application. This system uses Windows XP Embedded as operating system. Windows XP Embedded is the componentized version of the leading desktop operating system based on the same binaries as Windows XP Professional. So, it is useful in reducing development time because it can use win32 application and Windows device driver without modification.

The PC/104-plus is used to connect motion controller to single board computer. The PC/104-plus specification establishes a standard for the use of a high speed PCI bus in embedded application. The PCI bridge chipset that controls PCI bus is implemented in FPGA using PCI core of type of IP (Intellectual Property). In addition, the FPGA integrates encoder counter and pulse generator for motion control into one chip. It is efficient in reducing board size and system cost.

The other important problems in motion control are number of axes of control and many wiring. Network can resolve these problems. We can add additional axes of control by connecting slave devices to this motion system by network. In this paper, we use a CAN (Controller Area Network) that is verified successfully in poor environment of vehicles for past years. In the CAN-network, the connection is only simple 3 lines. The wiring simplicity can resolve unwieldy and many wiring problem. The CAN guarantees the transmission speed up to 1Mbit/sec. It is fast enough for most motion control application.

In this paper, a network-based stand-alone motion system is proposed and implemented. It communicates to two slave motion controllers and each controller control one step motor with encoder in experiment. In section 2, the configuration of proposed motion system is explained. In here, the single board computer, Windows XP Embedded, and CAN are explained. In section 3, the H/W configuration of motion controller is explained. In section 4, we experiment performance with a simple GUI. In section 5, the characteristic of proposed motion system is discussed.

2. CONFIGURATION OF NETWORK-BASED STAND-ALONE MOTION SYSTEM

The proposed network-based stand-alone motion system integrates PC and motion controller into one stand-alone system. The single board computer is used as replacing PC and is connected with motion controller by PC/104-plus. The Windows XP Embedded is used as operating system. In this paper, the proposed motion system serves as master that communicates to slave motion controllers. The CAN-network is used to connect master motion system and slave motion controller. Fig.1 shows the configuration of network-based stand-alone motion system.

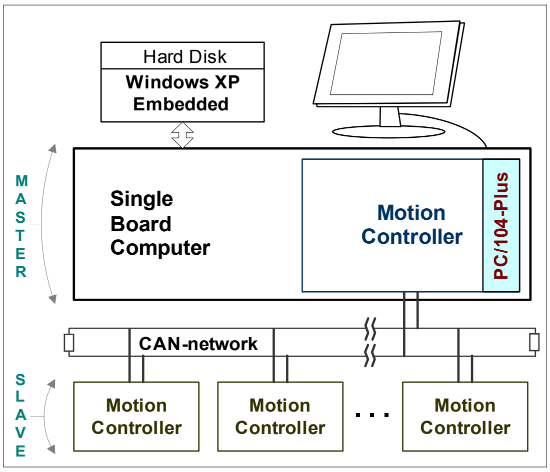

Fig. 1 The configuration of a network-based stand-alone motion system

The PC/104-plus interface that is used to connect motion controller to single board computer establishes a standard for the use of a high speed PCI bus in embedded application. It incorporates the PCI bus within the industry proven PC/104 form-factor, and it brings many advantages to user including fast data transfer over a PCI bus, low cost due to PC/104's unique self-stacking bus, and high reliability due to PC/104's inherent ruggedness.

2.1 Single Board Computer

In this paper, the proposed motion system uses a single board computer instead of PC that is big size and high cost. Fig. 2 is the picture of single board computer that is developed in Evalue Technology Inc.

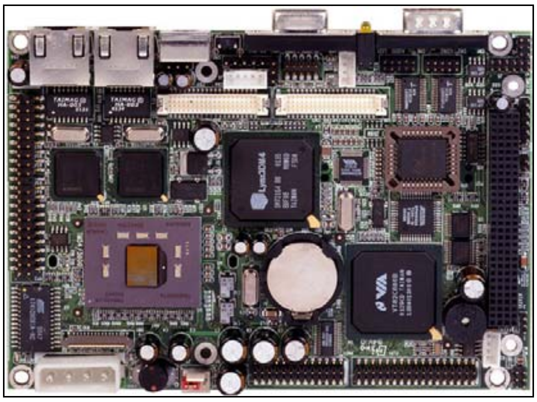

Fig. 2 The picture of single board computer

This single board computer is a small size of type of 3.5" form factor and is cost-effective by integrating CPU, memory, VGA, and peripherals into one micro module. It features onboard Transmeta Crusoe TM5400 500MHz Low-power CPU, onboard one 144-pin SODIMM socket supports up to 256MB SDRAM, onboard one 32-bit 120-pin PC/104-plus interface, etc [1]. And also, it simultaneously supports CRT and flat panel (TFT & DSTN Color) displays. Using the small flat panel display device, the user can reduce overall system size. The PC/104-plus interface using stack-through connector is allowable to stack several modules in order. The single board computer enables integrating motion controller and other system module, in example, such as vision system, with PC/104-plus.

2.2 Windows XP Embedded

The single board computer works with Windows XP Embedded as operating system. Windows XP Embedded is the componentized version of the leading desktop operating system, enabling rapid development of reliable and full-featured connected devices. Based on the same binaries as Windows XP Professional, Windows XP Embedded enables embedded developers to individually select only the rich features they need for customized, reduced-footprint embedded devices [2]. And also, the user can use device driver and software application that was used in desktop Windows operating system. It is very helpful to reduce development time.

Windows XP Embedded contains the Windows Embedded Studio toolset, which provides access to the componentized Windows technologies and enables developers to rapidly configure, build, and deploy smart designs. The tools include target designer, target analyzer, component designer, and component database manager. Fig. 3 shows a process of building and deploying the Windows XP Embedded with using Windows Embedded Studio.

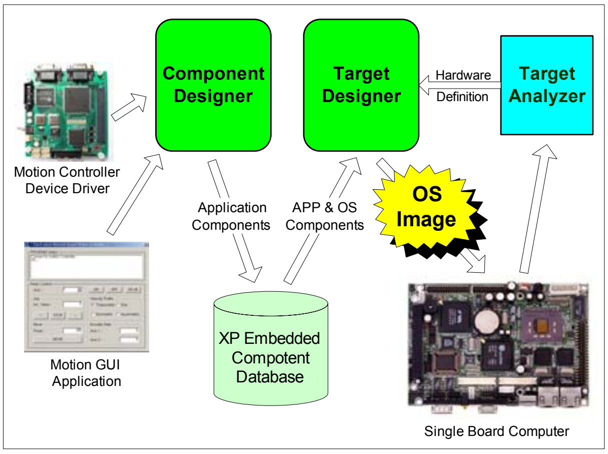

Fig. 3 The process of building and deploying the Windows XP Embedded

The build process is as follows. Creating a configuration using Target Analyzer is first step of building. Target Analyzer detects the hardware components of the target device and Target Designer creates base configuration. Component Designer converts unique driver and applications, in this paper, motion controller driver and motion GUI, into components. The components are added to the XP Embedded database and then are added to base configuration in Target Designer. At last, Target Designer builds Windows XP Embedded operating system image.

2.3 CAN (Controller Area Network)

In this paper, the CAN is used to connect master motion system and slave motion controllers. The CAN protocol, developed by ROBERT BOSCH GmbH in 1983, offers a comprehensive solution to managing communication between multiple CPUs. Communications may occur at a maximum recommended rate of 1Mbit/sec. The protocol has found wide acceptance in automotive in-vehicle applications as well as many non-automotive due to its low cost and high performance [3].

The CAN protocol is based on the so-called broadcast communication mechanism. This broadcast communication is achieved by using a message oriented transmission protocol. Thus not defining devices and device addresses, it only defines message. These messages are identified by using a message identifier, which has to be unique within the whole network and it defines not only the content but also the priority of the message. This will be important when several stations compete for bus access [6]. Fig. 5 shows the principles of data exchange.

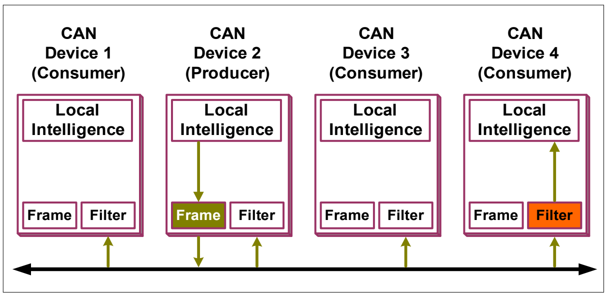

Fig. 4 The principles of data exchange

A high degree of system and configuration flexibility is achieved as a result of the content-oriented addressing scheme. It is very easy to add devices to an existing CAN network without making any hardware or software modifications to the existing devices as long as the new devices are purely CAN receivers.

The CAN protocol supports two message frame formats, the only essential difference being in the length of the identifier. The CAN version 2.0A can have 11bit's identifier and create 2032 different messages all, and 2.0B can have 29bit's identifier and create 536,870,912 different messages [4], [5]. Fig. 4 shows CAN 2.0B message frame format.

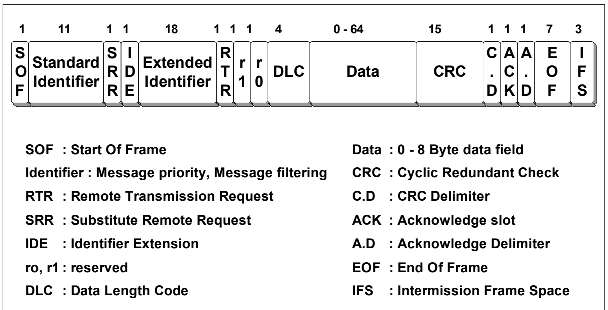

Fig. 4 CAN 2.0B message frame format

The CAN automatically transmits message again when message is collided. Error detection mechanisms, such as a 15-bit Cyclical Redundancy Check (CRC), frame check and ACK errors, provide a high level of data integrity.

3. THE H/W CONFIGURATION OF MOTION CONTROLLER

In this section, the hardware configuration of proposed motion controller is discussed. Fig. 5 shows the configuration of the proposed motion controller.

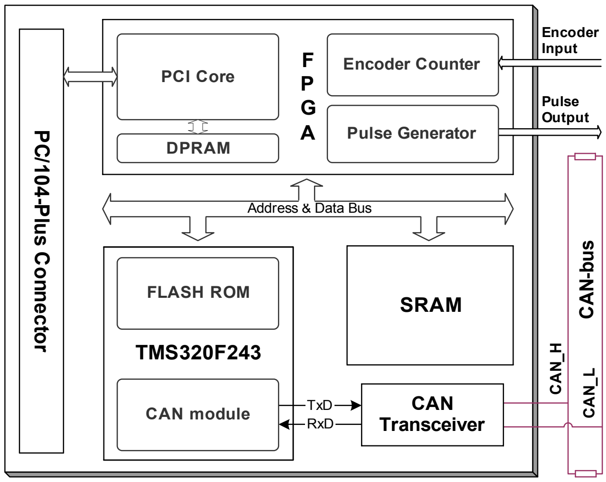

Fig. 5 The configuration of motion controller

The motion controller is consists of Processor, SRAM, FPGA and CAN transceiver. It features as follows.

- TMS320F243 that is designed for use in control application
- 64K words program memory and 32K words data memory
- FPGA that is integrated some peripherals into one chip
- CAN transceiver for high speed application of up to 1Mb/s
- PC/104-plus connectivity

In the next, each feature is explained in detail. Fig. 6 is the picture of proposed motion controller.

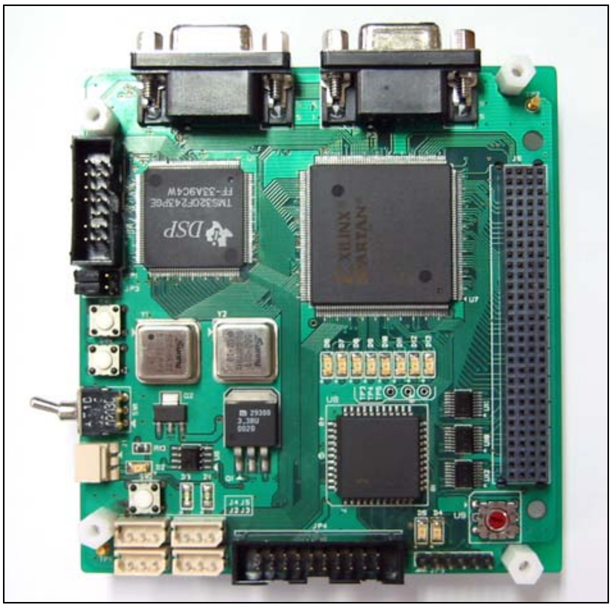

Fig. 6 The picture of motion controller

3.1 Processor and Memory

The processor of this motion controller is a TMS320F243 DSP of Texas Instruments Inc. which is designed for use in digital control application. It has 20 MIPS operation capacity with using system Clock of 20MHz. This DSP controller has several advanced peripherals optimized for motor/motion control applications. These peripherals include the event manager module, which provides general-purpose timers and PWM registers to generate PWM outputs. And also, it offers a unique combination of on-chip peripherals such as 8K words flash memory, ultra-fast A/D converters, and robust CAN modules. With these features, developers can advance innovation and time to market as well as reduce their system costs.

To support applications requiring large memory sizes, the motion controller uses external memory. The external memory is composed of two fast SRAMs of 128 K x 8bits. 64 K words are used as program memory, and 32 K words are used as data memory.

3.2 FPGA (Field Programmable Gate Array)

In this paper, the motion controller needs several peripheral such as encoder counter, pulse generator, PCI bridge chipset, and DPRAM (Dual Port RAM). Using each of these separately, the board size becomes large and system cost becomes high. So, FPGA is used to integrate these peripheral into on chip. It reduces the chip count and reduces the integral system cost. In this motion controller, The XC2S200 device that is Spartan series FPGA in Xilinx, Inc. is used. This device has 200K system gates.

The PC/104-plus is used to connect motion controller to single board computer. It establishes a standard for the use of a PCI bus. The PCI bridge chipset that controls PCI bus is needed, so it is implemented in FPGA using PCI core of type of IP (Intellectual Property). And also, DPRAM is needed to save the motion information transmitted from single board computer for motion control and is needed to save the data transmitted from motion controller to single board computer. That is, DPRAM is used to communicate between single board computer and motion controller. It is implemented in FPGA, also.

In order to obtain absolute position information of motor and to do closed-loop control, we must find a direction of motor and count encoder signal that is entered from encoder of motor. The 24bit encoder counter is designed to count pulses. The encoder output often has some glitches that cause fatal error of counter. So we use digital delay filter to eliminate glitch. It uses 2-of-3 voting method with using four D-flip-flops. It is positioned encoder signal phase A, phase B and Phase Z. Fig. 7 shows the principle of digital delay filter.

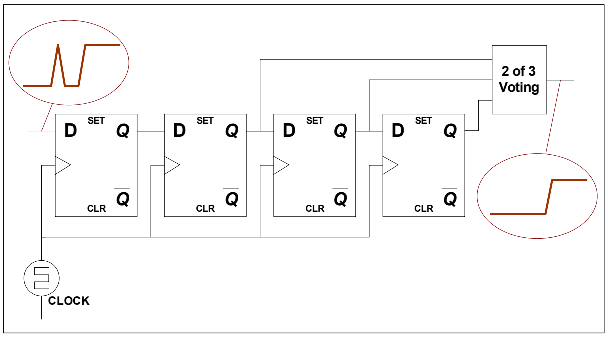

Fig. 7 The principle of digital delay filter

The proposed motion controller controls step motor. So, the pulse generator is needed to control step motor. The processor calculates pulse count per sampling time, and transfer to pulse generator. The pulse generator consists of two functions, which are DDA (Digital Differential Analyzer) and Phase Generator. The DDA is generates reference clock for phase generator. The phase generator generates four signal (phase A, phase /A, phase B, phase /B) of step motor with using this reference clock. These four phase signal connected with motor driver. The encoder counter and pulse generator are implemented in FPGA.

3.3 CAN transceiver

In this proposed motion controller, a CAN transceiver is needed to convert CAN_H and CAN_L that is CAN-bus signal to TTL logic level. Fig. 8 shows the function of CAN transceiver.

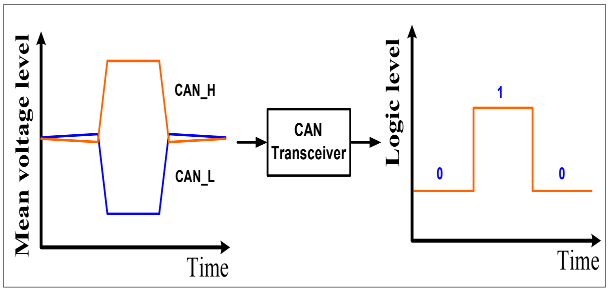

Fig. 8 The function of CAN transceiver

The SN65HVD251 of Texas Instruments Inc. is used in this motion controller. As a CAN transceiver, the SN65HVD251 provides differential transmit capability to the bus and differential receive capability to a CAN controller at speeds up to 1Mbit/sec.

4. EXPERIMENT

In the experiment, two slave motion controllers connected to network-based stand-alone motion system. Each motion controller controls one step motor with encoder. The encoder generates 1,000 pulses per one rotation. Each slave motion controller has external motor driver for driving motor. The whole system works with simple motion GUI. Fig. 9 shows the configuration of overall motion system that is used in experiment.

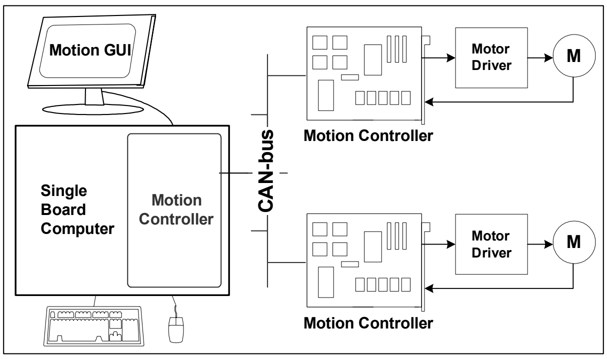

Fig. 9 The configuration of overall system used in experiment

The operation of network-based stand-alone motion system is as follows. The user orders motion command in motion GUI. The single board computer transmits the 8 bytes motion command and data to the master motion controller through PC/104-plus. Fig. 10 shows the 8 bytes motion command and data.

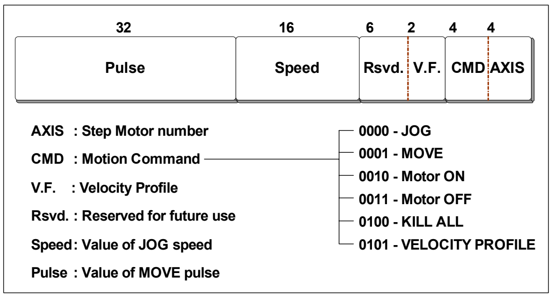

Fig. 10 The 8 bytes motion command and data

This motion command and data is saved in the internal DPRAM of FPGA, and interrupt is generated and requested to the DSP. The DSP of master motion controller fetches this command from DPRAM and analysis command. Then, the master motion controller transmits motion command and data to slave motion controllers through CAN-bus. The CAN message Frame has 29bit's identifier, and each motion controller is identified by unique identifier. The slave motion controller executes this motion command and data.

The motion GUI is made with using Microsoft Visual C++ 6.0. It is operated on Windows XP Embedded. Fig. 11 is the motion GUI that is used in experiment.

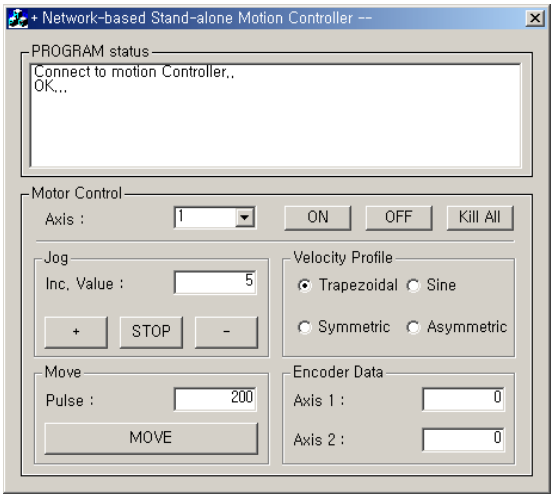

Fig. 11 The motion GUI that is used in experiment

The motion GUI is consist of two part – PROGRAM status and Motor Control part. The PROGRAM status displays the operated results of motion command and error message that is generated during operation of motor. In the motor control part, the user can order several motion commands. There are axis on-off, jog, move and velocity profile command. Using 'JOG' command, motor speed can be increased or decreased. Using 'MOVE' command, the motor moves as specified pulse. The current encoder count is displayed in "Encoder Data" box during operation of 'MOVE' command. The "Velocity Profile" command selects one of the various acceleration and deceleration velocity profiles such as trapezoidal, sine, symmetric and asymmetric curve. The expected various acceleration and deceleration velocity profiles enable to reduce trajectory errors [7], [8]. It is efficient method that improves the performance of position control in motion controller.

The slave motion controller receives CAN message including motion command and data through CAN-bus, and compares CAN message's identifier with own identifier. If it is same, the motion controller accepts that message. Then, the motion controller executes received command. Fig. 12 shows this operating process.

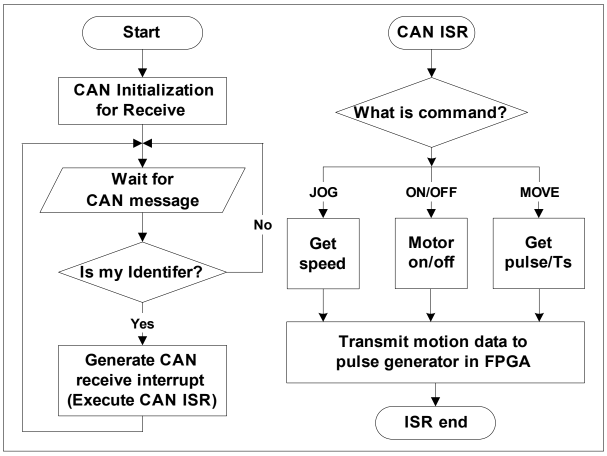

Fig. 12 The operation of slave motion controller

If 'MOVE' command is selected in motion GUI, the specified pulse is transmitted to master motion controller and the master motion controller calculates pulse count per sampling time, 8ms in this experiment, with using specified velocity profile. The pulse count is transmitted to slave motion controller at every 8msec. The slave motion controller uses P-control algorithm to control motor correctly, which uses difference between received pulse count and current motor position that is known by encoder counter. Fig. 13 shows the result of 'MOVE' command when the specified pulse is 20,000 and velocity profile of each axis are trapezoidal and S-curve, respectively.

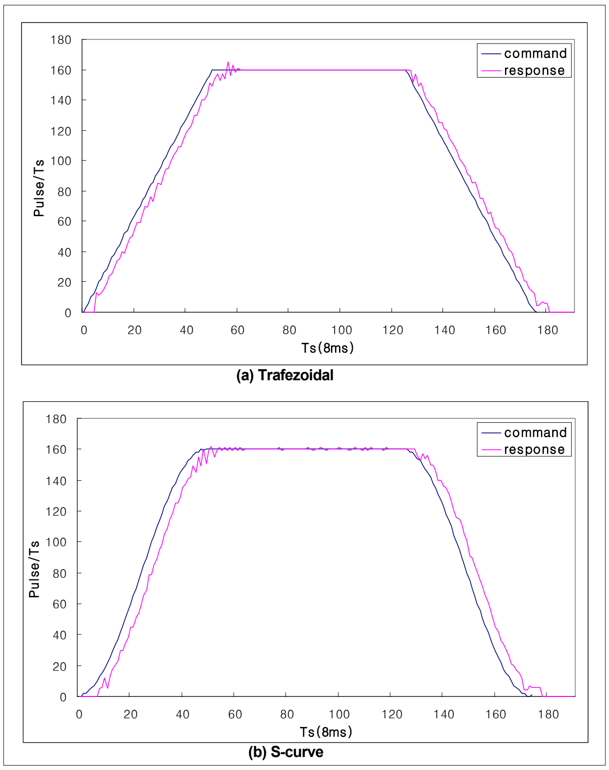

Fig. 13 The result of 'MOVE' command in experiment

5. CONCLUSION

In this paper, the network-based stand-alone motion system is proposed and implemented. The former motion controller based on PC has several problems such as system size, many wiring and limitation of axes of control. Using the single board computer instead of PC, it reduces the overall system size and reduces the integral system cost. Also, the Windows XP Embedded operating system allows the developer to reduce the development time. Using the operating system, it can supports motion GUI that allows the user to easily use motion controller. The network functionality of motion controller resolves the wiring problem and the limitation of axes of control. The CAN-network used in this motion controller has only 3 signal line, and makes wiring simply between devices. Also, to add additional axes of control, the user has only to connect the new motion controllers to an existing CAN-network without making any hardware or software modifications to the existing system as long as the new ones are purely CAN receivers.

This network-based stand-alone motion system can be used in small automation devices by itself and various factory environments that need many axes of control with using additional slave motion controllers. In experiment, the function of slave motion controller is very simple. So, it can be integrated into one FPGA including CAN controller and motion functions. Also, the bandwidth of CAN-protocol may be insufficient according as axis of control increases. The bandwidth problem will be resolved with using high speed network protocol such as Ethernet and IEEE1394 and it will improve the system performance.

REFERENCES

[1] Single board Computer, http://www.evalue-tech.com.

[2] Windows XP Embedded, http://www.microsoft.com/windows/embedded/xp.

[3] Introduction to the Controller Area Network (CAN) Protocol, Intel Corporation, 1995.

[4] Robert Bosch GmbH, "CAN Specification" Version 2.0, 1992, ISO/TC22/SC3/WGI document.

[5] Lars-Berno Fredriksson, "Controller Area Networks and The Protocol CAN for Machine Control Systems".

[6] CAN protocol, http://www.can-cia.de/can/protocol.

[7] J. W. Jeon, and Y. G. Kim, "FPGA Based Acceleration and Deceleration Circuit for Industrial Robots and CNC Machine Tools," *Mechatronics*, Vol. 12, Issue 4, pp. 635-642, May. 2002.

[8] J. W. Jeon, "An Efficient Acceleration for Fast Motion of Industrial Robots," *Proceedings of 1995 IEEE 21st IECON,* Orlando, FL, pp. 1336-1341, Nov. 6-10, 1995.