Journal of Knowledge Information Technology and Systems

ISSN 1975-7700

http://www.kkits.or.kr

Making and Application of Educational mini-Board under Visual C++ Development Environment

Su-Chul Hwang*

Department of Computer Systems and Engineering, Inha Technical College

ABSTRACT

A control system using computer is used in variety field such as image process and device control etc. Therefore the University's curriculum includes some subjects for teaching a basic knowledge of H/W control using PC. But there is a difficult in training of interface application programming in Visual C++ development environment due to absence of appropriate education tools of low price using Visual C++ that is a powerful develop language. In order to solve this problem we suggest new tool with compose of a mini-board, libraries and application programs based on Visual C++ MFC environment. First, we made the board of 120mm*100mm size and a firmware to support communication protocol based on a Modbus protocol which is used in industrial fields widely. And we implemented a few library functions to control devices mounted on the board. Last we provided of some sample programs to enable students to self-study of H/W control in its environment. If this result is used in programming education course of university, it is expected that students may have interesting about it and naturally learn a basic knowledge of device control field.

ⓒ 2016 KKITS All rights reserved

KEYWORDS: Interface application programming, Firmware, Communication protocol, Visual C++, H/W control

ARTICLE INFO: Received 10 November 2016, Revised 12 December 2016, Accepted 12 December 2016.

*Corresponding author is with the Department of *Computer Systems and Engineering, Inha Technical College,* 100 Inha-ro Nam-gu Incheon, 22212, KOREA.

E-mail address: schwang@inhatc.ac.kr

www.kci.go.kr

1. 서 론

컴퓨터를 이용한 하드웨어 제어 즉, 하드웨어 인터페이스 응용 분야의 교육 도구들 중에 개발 언어 환경이 Visual C++인 저가의 실습용 키트의 경우는 부재한 실정이다. 최근에 이러한 점을 감안하여 MFC 기반의 교육용 타겟 보드에 관한 연구[1]가 있었으며, 저가형 보드를 사용한 교육에의 활용 가능성을 보였다. 하지만 통신 프로토콜[2][3] 관련 펌웨어[4]가 표준을 따르지 못하고 타겟 보드 설계의 사소한 오류 등으로 인해 불안정성을 보였다.

그래서 본 연구에서는 새로이 미니 보드를 제작하고 산업계에서 널리 사용되는 모드버스(ModBus)[5][6] 기반의 통신 프로토콜을 지원하는 펌웨어를 구현하였으며 미니보드에 탑재된 장치들을 제어 할 수 있도록 라이브러리 함수도 구현하였다. 그리고 교육생으로 하여금 MS사의 Visual C++ MFC[7][8] 기반의 개발 환경에서 제어 분야대한 자가 학습이 가능하도록 샘플 응용 프로그램을 제공하였다.

이후 본 논문의 구성은 다음과 같다. 제2장에서는 미니 보드의 회로 및 구성을 포함하는 교육용 미니 보드 및 설계에 대해 기술한 후에 모드버스 기반의 MyBus 통신 프로토콜 구현에 대해 언급하였다. 그리고 미니 보드내의 장치들을 제어하기 위해 구현된 라이브러리 파일인 MyBus DLL을 소개했으며, 이를 활용하기 위한 응용 프로그램 구현과 실행예를 제시하였다. 그리고 제3장에서는 결론을 기술하였다.

2. 교육용 미니 보드 및 응용 개발

2.1 미니보드 설계 및 제작

본 연구에서 보드 설계의 핵심은 작은 보드에 최소의 I/O장치를 탑재하여 간단한 프로그램을 통해 하드웨어를 제어하도록 하는 것이다. 이를 위해 <그림 1>과 같은 회로를 PCB 아트워크[9]를 통해 120mm*100mm 미니 보드로 제작하여 10개의 I/O 부품을 탑재하였다.

탑재되는 부품들은 출력장치에 해당되는 4자리 FND, 피에조, LED, RGB LED, 모터와 입력장치에 해당되는 전위차계, 조도센서, 온도센서, 초음파 센서, Tact 스위치이다. CPU는 8-bit 마이크로 컨트롤러인 AVR ATmegat128가 장착하였으며, 외부와의 통신을 위해서 FT230x Basic UART[10]를 지원하도록 mini 5p USB 포트를 장착하였다.

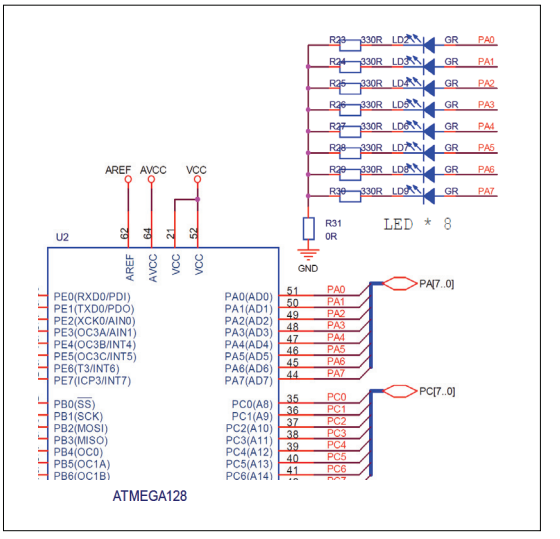


그림 1. 미니 보드의 회로도 일부
Figure 1. A Part Circuit of Mini-Board

<표 1>은 이러한 입출력 장치에 대한 특성을 간략히 제시한 것이며, <그림 2>와<그림3>은 미니 보드의 기본 구성과 최종적으로 제작된 미니 보드를 보인 것이다.

www.kci.go.kr

표 1. 보드설계 및 장치 특징
Table 1. Configuration of Board Design and Unit

구분	장치명	설계 및 장치 특징
출력	FND	0~9999 숫자 표시,다이나믹 방식 구동
	피어조	타이머/카운터 0의 CTC모드를 이용 출력 주파수를 조절하여 원하는 음계를 만들 수 있도록 설계
	LED	8개의 LED가 PORTA에 연결됨. 논리 1에 On, 논리 0에서 Off 되도록 설계
	RGB LED	타이머/카운터 0의 PWM 출력과 연결되어 LED 밝기를 조절할 수 있음
	Fan모터	출력포트에 트랜지스터를 연결하여 모터 구동에 필요한 전류를 공급하도록 설계
입력	전위차계	임의의 전압을 만들어 A/D 컨버터를 테스트
	조도센서	밝은 곳에서는 높은 전압, 어두운 곳에서는 낮은 전압 유지
	온도센서	1℃당 10.0mV의 비율로 측정 가능 온도 -40℃~125℃ 까지 측정 가능
	초음파센서	HS-SR04 모듈을 이용해 거리 측정
	Tact스위치	스위치 ↓:논리 0, 스위치 ↑:논리 1

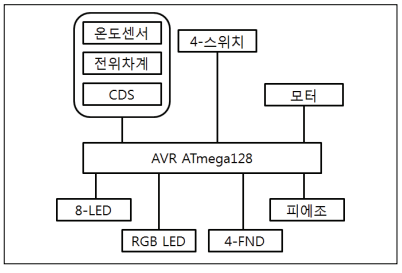


그림 2. 미니 보드 기본 구성
Figure 2. Basic Structure of Mini-Board

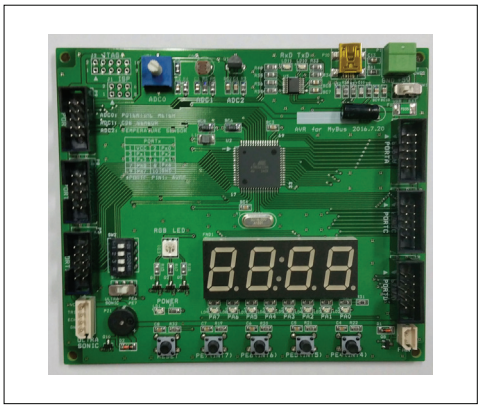


그림 3. 제작된 미니 보드
Figure 3. Manufactured Mini Board

2.2 MyBus 프로토콜 구현

(1) MyBus 통신 프로토콜

시리얼 통신을 이용해서 미니보드와 연결된 주변 장치를 제어하기 위해서는 메모리 내의 데이터 영역인 레지스터나 포트의 값을 읽거나 변경할 수 있어야 한다. 단순 통신이 아닌 제어가 목적인 통신 프로토콜은 레지스터나 포트와 같은 저장소에 읽기/쓰기 명령을 제공해 주어야 한다. MyBus는 이러한 기능을 수행하도록 설계된 ModBus 기반의 통신 프로토콜이다.

ModBus 프로토콜[11]은 공개 프로토콜로 설치와 보수가 용이하고 비트단위나 워드 단위로 정보조작이 용이하며 240여개의 장비들을 서로 연결할 수 있어서 산업용 어플리케이션에서 많이 사용되고 있다. 본 연구에서는 이 프로토콜을 기반으로 본 연구 목적에 맞게 변형하여 MyBus 프로토콜을 구현하였으며, 입출력 기능을 수행하기 위해서 미니보드에 펌웨어로 내장된다. 작동 방식은 기본적으로 마스터/슬레이브 형태의 질의-응답 방식이며, 마스터에서 보낸 프레임(명령)에 따라 미니보드에 탑재된 장치가 제어된다.

(2) MyBus 프로토콜 프레임 구조와 매핑

MyBus 프로토콜 프레임의 크기는 4바이트이며, 미니보드의 레지스터-포트에 대한 read-write 명령이 저장된다. <표 2>는 MyBus 프로토콜 프레임 구조를 보인 것으로 명령어 코드, 주소, 상위데이터 및 하위데이터 필드로 구성된다. 각 필드의 기능은 다음과 같다.

www.kci.go.kr

① 명령어 코드(Instruction Code)

<표 3>에 나열된 6 종류의 마스터(M)/슬레이브(S) 입출력 명령 코드가 저장되며, 마스터에서는 질의로 사용되고, 슬레이브에서는 응답용으로 사용된다.

② 주소(Address)

명령어 코드에 따라서 레지스터의 주소(번호)나 포트의 주소(번호)가 저장된다. 명령어 코드가 레지스터 입출력과 관련된 것이면 레지스터 주소로 사용되고, 포트 입출력과 관련된 명령 코드라면 포트 주소로 사용된다.

③ 상위데이터(Hi Data)

명령어 코드에 따라 상위데이터나 비트(bit)번호로 사용된다. 입출력 레지스터 관련 명령 코드이면 상위 8-비트 데이터로 사용되고, 입출력 포트와 관련된 명령어 코드라면 제어할 포트의 비트 번호로 사용된다.

④ 하위데이터(Low Data)

명령어 코드에 따라 하위데이터나 논리 상태값으로 사용된다. 입출력 레지스터와 관련된 명령어 코드라면 하위 8-비트 데이터로 사용되고, 입출력 포트와 관련된 명령 코드라면 논리값 0/1(0x01/oxff)을 나타내는데 사용된다.

표 2. MyBus 프로토콜 프레임 구조
Table 2. Structure of MyBus Protocol Frame

1th Byte (Instruction Code)	2th Byte (Address)	3th Byte (Hi Data)	4th Byte (Low Data)
마스터(M):질의 슬레이브(S):응답 에러(E)	I/O 레지스터 번호 또는 I/O 포트 번호 (0~255)	상위데이터 또는 비트번호 (0~255)	하위데이터 (0~255)

표 3. 6-종류의 명령어 코드
Table 3. Instruction Code of 6-Type

명령어코드(MS/E)	기능
ROP(0x01/0x81)	출력포트영역에서 지정한 비트의 논리 상태를 읽어옴
RIP(0x02/0x82)	입력포트영역에서 지정한 비트의 논리 상태를 읽어옴
ROR(0x03/0x83)	출력레지스터영역에서 지정한 레지스터의 값을 읽어옴
RIR(0x04/0x84)	입력레지스터영역에서 지정한 레지스터의 값을 읽어옴
WOP(0x05/0x85)	출력포트영역에서 지정한 비트의 값을 변경시킴
WOR(0x06/0x86)	출력레지스터영역에서 지정한 레지스터의 값을 변경시킴

<그림 4>는 MyBus에서 사용 가능한 미니보드 메모리내의 데이터 영역을 보인 것으로 명령 코드에 따라서 사용되는 영역이 달라진다. 입출력을 위한 포트 영역과 레지스터 영역으로 구분되며 각각 250개이다. 4개 영역 중 입력 포트와 레지스터는 읽기만 가능하고, 출력 포트와 레지스터는 읽기와 쓰기가 가능하다. 여기서 포트는 비트 단위로 제어가 되는 영역이고, 레지스터는 최대 16비트 크기의 값을 읽거나 쓸 수 있는 영역이다.

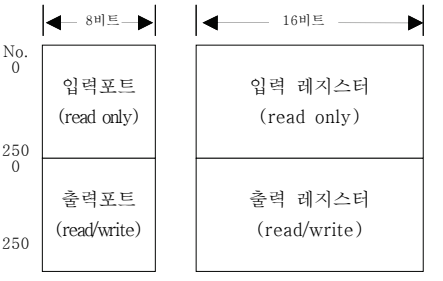


그림 4. MyBus 데이터 영역
Figure 4. Data Area for MyBus

메모리 내의 입출력 포트 및 입출력 레지스터용 데이터 영역은 미니보드에 탑재된 장치들과 <표 4>와 같은 형식으로 매핑 된다. MyBus는 슬레이브로 동작하며 마스터에 해당되는 응용 프로그램에서 보낸 명령(프레임)에 따라 해당 장치가 제어된다.

www.kci.go.kr

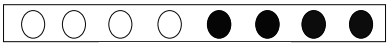
예를 들어, 레지스터(또는 포트) 0번이 PORTA와 매핑(연결)되고 그것이 LED라면 레지스터 0번에 임의의 값을 쓰는 명령으로 <그림 5>처럼 LED의 상태를 변경시킬 수 있다.

표 4. 출력 레지스터 영역 매핑
Table 4. Area Mapping of Output Port

번호	이름	설명(기능)
0x00	VC_PORTA	ATMega128A PORTA (출력 포트 레지스터 A)
0x01	VC_PORTB	ATMega128A PORTB (출력 포트 레지스터 B)
0x02	VC_PORTC	ATMega128A PORTC (출력 포트 레지스터 C)
0x03	VC_PORTD	ATMega128A PORTD (출력 포트 레지스터 D)
0x04	VC_PORTE	ATMega128A PORTE (출력 포트 레지스터 E)
0x05	VC_PORTF	ATMega128A PORTF (출력 포트 레지스터 F)
...	...	...

명령어 코드	레지스터 번호	상위 데이터	하위 데이터
0x06	0x00	0x00	0x0f

(a) LED 제어에 대한 MyBus 질의 명령



(b) 제어된 LED 상태

그림 5. 매핑 주소 사용 예
Figure 5. Example for Mapping Address

2.3 MyBus DLL 구현

MyBus DLL은 시리얼 통신을 제어하는 루틴과 MyBus를 지원하는 루틴을 DLL(Dynamic Link Library)에 담은 프로그램 모듈로 응용 프로그램에 항상 포함시켜 사용해야 한다. 이 라이브러리 파일은 <표 5>처럼 시리얼 통신 포트 설정, 시리얼 데이터 송신, MyBus 프레임 송신 등 크게 3 종류의 라이브러리 함수를 제공한다. 시리얼 데이터 수신과 MyBus 프레임 수신은 윈도우 메시지 형식으로 응용 프로그램 전달된다.

표 5. MyBus DLL과 윈도우 메시지
Table 5. MyBus DLL and Window Message

함수종류 메시지	함수 및 메시지명과 기능	
	함수 및 메시지	기능
시리얼 통신포트 설정	CallCommSetDlg()	시리얼 통신포트 설정 대화상자 호출
	SetCommCfgData()	시이얼 통신 포트 설정 데이터를 MyBus DLL로 전송
	GetCommDfgData()	포트 설정 데이터를 MyBus DLL에서 가져옴
시리얼 데이터 송신	CommOpen()	시리얼 통신 포트 연결
	CommClose()	시리얼 통신 포트 연결 해제
	CommPutChar()	한 문자를 시리얼 통신 포트를 통해 전송
	CommPutData()	일정량의 데이터를 전송
프레임 송신	MyBusSendFrame()	명령 프레임을 전송
	MyBusLED_OnOff()	미니보드내 8-LED를 제어
수신관련 윈도우 메시지	WM_MYNUS_COMM_CLOSE	장치 에러시 발생 메시지
	WM_MYBUS_SEND_FRAME	MyBusSendFrame() 호출시 발생되는 메시지
	WM_MYBUS_RECV_FRAME	미니보드에서 응답 프레임을 수신하면 발생되는 메시지

시리얼 통신 포트와 관련된 함수에는 <그림 6>과 같은 자료 구조를 갖는 변수가 사용되며, 명령 프레임 전송 시에는 <그림 7>과 같은 자료 구조를 갖는 변수가 사용된다.

```
struct      CommConfigData{
  HWND      hOwner; // DLL함수를 호출하는 응용 프로그램의 핸들
  BYTE      PortNum; // 시리얼 통신 포트 번호 (COM1, COM2,..)
  DWORD     BaudRate;   // 통신 속도 2400bps ~ 115200bps
  BYTE      ByteSize;   // Byte 크기 5,6,7,8 (기본 8bit)
  BYTE      StopBits;   // Stop Bit수 1, 1.5 2 (기본 1bit)
  BYTE      Parity;     // 홀수,짝수,없음
};
```

그림 6. 통신 포트 관련 함수에서 사용하는 자료구조
Figure 6. Data Structure using in Communication Port Function

www.kci.go.kr

```
typedef struct{
   UINT8    ui8Func;    // 명령코드 (1~6)
   UINT8    ui8Addr;    // 주소 (0~255)
   UINT8    ui8DataH;   // 상위 데이터(0~255)
   UINT8    ui8DataL;   // 하위 데이터(0~255)
}_MyBusFrame;
```

그림 7. 명령 프레임 관련 함수에서 사용하는 자료구조
Figure 7. Data Structure using in Instruction Frame Function

<그림 8>은 MyBus 통신 프로토콜 프레임 기반의 질의(명령)-응답과 관련된 미니보드와 응용 프로그램간의 전체적인 처리 과정을 보인 것이다. 예들 들어, 응용 프로그램에서 조도센서를 제어하는 명령이 보내지면 통신 모듈(MyBus DLL)이 이를 받아서 명령프레임을 만들어 펌웨어에 전달하며, 펌웨어는 이를 받아 분석한 후에 해당 명령어를 통해 포트에 연결된 장치들을 동작시킨다. 명령 프레임을 송수신 할 때에는 관련 윈도우 메시지가 발생한다. <그림 9>는 응용 프로그램이 어떻게 MyBus DLL과 미니보드와 연결되어 처리되는지 전체적인 실행과정을 보인 것이다.

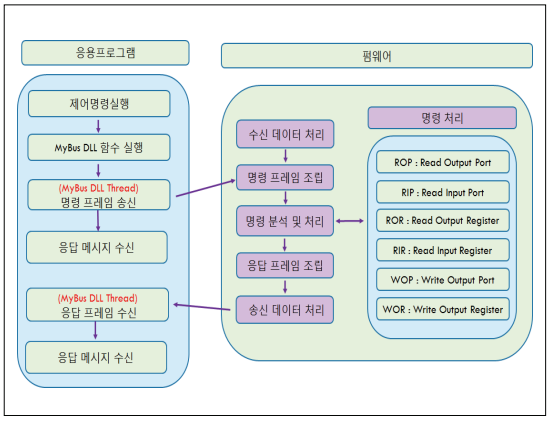


그림 8. 응용과 펌웨어간 명령 처리 과정
Figure 8. Instruction Process between Application and Firmware

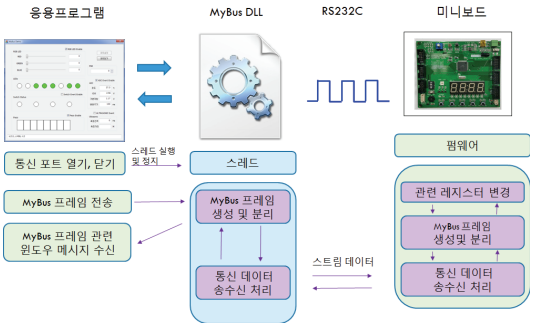


그림 9. 응용 프로그램의 전체적인 실행과정
Figure 9. Overall Execution Process of Application Program

2.4 응용 프로그램 구현 및 실행

<표 6>에서 제시한 것처럼 10개의 샘플 응용 프로그램과 이를 하나의 대화상자 창에 통합한 응용 프로그램을 구현하였다. 그리고 응용 프로그램을 제작할 때마다 미니보드와 연동하기 위해서 매번 코딩해야하는 불편함을 없애기 위해서 통신 포트 설정 및 연결 기능을 담은 템플릿을 구현하여 활용할 수 있게 하였다.

표 6. 샘플 응용 프로그램 리스트
Table 6. List of Sample Application Program

순번	응용 프로그램
1	LED 제어 프로그램
2	RGB LED 제어 프로그램
3	FND 제어 프로그램
4	피에조 제어 프로그램
5	초음파 센서 제어 프로그램
6	A/D 변환 프로그램
7	조도 센서 입력 프로그램
8	온도 센선 입력 프로그램
9	Tact 스위치 입력 프로그램
10	팬모터 제어 프로그램

MS사의 Visual Studio내의 Visual C++로 새로운 프로젝트를 생성할 때 미리 제작해 놓은 템플릿(통신 포트 연결 프로그램)을 선택한 후에 앞서 기술

www.kci.go.kr

한 <표 5>의 라이브러리 함수들을 이용하여 원하는 기능을 프로그래밍 함으로써 10종류의 장치 제어가 가능하다. <그림 10>과 <그림 11>는 이들 중에 대표적으로 LED와 팬모터 제어를 위한 실행 화면과 미니 보드의 상태를 보인 것이다.

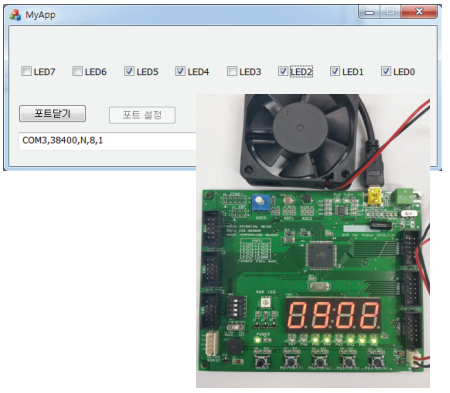


그림 10. LED 제어 실행 예
Figure 10. Execution Example of LED Control

<그림 10>은 포트 설정 및 연결 후에 실행 화면에서 led0,1,2와 led4,5를 체크하면 미니 보드의 해당 Led 가 On되는 것을 보여 준 것이다. 이러한 명령을 포트 출력으로 할 때에는 포트의 비트 값을 설정해 주어야 하고, 레지스터로 출력으로 할 때에는 16비트 값을 설정하는 응용 프로그램을 작성해야 한다.

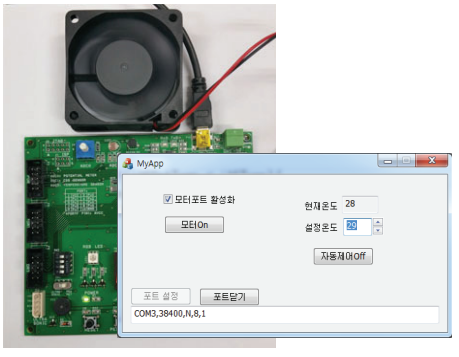


그림 11. 팬 제어 실행 예
Figure 11. Execution Example of Fan Control

<그림 11>은 포트를 설정 및 연결한 다음에 모터의 포터를 먼저 활성화하고 모터 On/Off를 통해서 모터를 제어 할 수 있거나, 입력되는 온도 값(29℃)이 미니 보드에 탑재된 온도 센서의 현재 온도 값(28℃) 보다 크면 팬을 작동시킬 수 있는 예를 보인 것이다.

3. 결 론

오늘날 프로그래밍을 통한 전자 부품의 작동 교육을 초등학생부터 실시하기 위한 연구[12]도 진행된 바 있으며, 컴퓨터를 이용한 제어 분야의 범위가 다양해짐에 따라 대학 교육에서 제어 분야의 기초 지식을 함양시키기 위해서 다양한 수업이 진행되고 있다. 이 수업에는 저가의 아두이노(arduino) 보드를 이용한 인터페이스 응용[13][14]에서부터 고가의 교육용 키트[15][16]가 사용되고 있다. 이때 사용하는 개발 도구(언어)는 주로 C또는 JAVA 계열의 언어로 Visual C++ 개발 환경에서의 실습 장비는 부재한 상태이다. 과거 프린터 포트를 이용한 VC++ 인터페이스 응용이란 연구와 교재가 있었지만 현재는 단종된 상태이다[17][18].

그래서 본 연구에서는 Visual C++ MFC 개발 환경에서 제어 분야에 대한 기초적인 지식을 함양할 수 있는 교육용 미니 보드를 PCB 아트워크를 통해 구성 및 제작하고 탑재된 장치들을 응용 프로그램에서 제어할 수 있도록 MyBus 통신 프로토콜을 구현하였으며, 이와 관련된 MyBus DLL을 제공하였다. 그리고 미니 보드에 탑재되어 있는 장치들을 간단히 제어할 수 있는 샘플 응용 프로그램도 제공하여 대학에서 교육용으로 활용할 수 있게 하였다.

본 결과물은 학생들로 하여금 흥미를 갖고서 자연스럽게 장비 제어 분야의 기초 지식을 습득할 수 있게 해줄 것이라 기대하며, 추후 심화된 내용으로 라이브러리 파일의 작성 과정과 펌웨어의 제

www.kci.go.kr

작 과정을 본 결과물에 추가 한다면 시스템 프로그래머 수준의 자가 학습도 가능하리라 기대한다.

References

[1] S-C. Hwang, *A study of educational target board based on MFC*, Proceedings of the 19th KKITS Spring Conference, Korea Knowledge Information Technology Society, Vol. 10, No. 1. pp. 8-11, 2016.

[2] J. Axelson, *Serial port complete:COM ports, USB virtual COM ports, and ports for embedded systems*, Lakeview Research, 2007.

[3] T. Kientzle, *The working programmer's guide to serial protocols: Learn everything you need to know to create fast file transfer protocols*, Coriolis Group, 1995.

[4] H-T. Kim, H-K. Oh, and S-B. *Du, Firmware development - ARM R CORTEX TM-MO NUC140*, NaeHa Press, 2012.

[5] J. S. Rinaldi, *Modbus: The everyman's guide to modbus*, CreateSpace Independent Publishing Platform, 2015.

[6] Woodhead Software & Electronics, *Protocol Manaul:JBus/Modbus,Woodhead*, 2003.

[7] S. Njornander, *Microsoft visual C++: Window Application by Example*, Packt Pub., 2008.

[8] P-S. Choi, *Visual C++ MFC window programming*, Daall Media, 2011.

[9] H-S. Min, J-S. Song, *PCB ARTWORK*, Seong andang, 2011.

[10] Future Tech Devices International Ltd., http://www.ftdichip.com/Products/ICs/FT230X.htm, Sep. 2016.

[11] IT Tip,http://ymkimit.blogspot.kr/2013/11/modbus_25.html#!/2013/11/modbus_25.html, Sep. 2016.

[12] P-W. Park, *A model for programming education with digital kits*, Preceedings of Spring Conference, The Korea Contents Assoication, pp. 357-358, 2011.

[13] J. Oxer, and H. Blemings, *Practical arduino:Cool projects open source hardware*, Apress, 2009.

[14] P. D. Justo, and E. Gertz, *Atmospheric monitoring with arduuino*, O'Reilly Media Inc., 2013.

[15] HANBACK Electronics, http://www.hanback.co.kr, Sep. 2016.

[16] HUINS, http://www.huins.com, Sep. 2016.

[17] G-J. Ryu, J-B. Ahn, H-J. Seo, Y-W. Kim, and T-W. Cho, *A design of parallel port application kit using GUI method in VC++*, Preceedings of the Summer Conference, The Institute of Electronicsand Information Engineers,Vol. 31, No. 1, pp. 1193-1194, 2008.

[18] C-G. Jeon, and T-H Cha, I*nterface application by visual C++ to control with printer port*, Bogdoo Press, 2008.

Visual C++ 개발환경하의 교육용 미니 보드 제작 및 응용

황수철

인하공업전문대학 컴퓨터시스템과

요 약

컴퓨터를 이용하는 제어 시스템은 영상처리, 장비 제어 등 다양한 분야에서 사용되고 있다. 이에 따라서 대학의 교육과정은 pc를 사용하여 하드웨어를 제어하는 기본 지식을 가리키는 교과목을 포함한다. 하지만 강력한 프로그램 개발 언어인 Visual C++을 사용하는 저가형의 적당한 교육 도구 부재로 인해 Visual C++ 개발 환경에서 하드웨어를 제어하는 인터페이스

www.kci.go.kr

응용 프로그래밍을 실습하는데 어려움이 있다. 이러한 문제점을 해결하기 위해서 우리는 Visual C++ MFC 환경 기반의 미니보드, 라이브러리 그리고 응용 프로그램으로 구성된 새로운 툴을 제시하였다. 먼저 120 mm*120mm 크기의 보드와 산업계에서 널리 사용되는 모드버스 기반의 통신 프로토콜을 지원하는 펌웨어를 제작하였다. 그리고 미니보드에 탑재된 장치들을 제어할 수 있도록 라이브러리 함수도 구현하였다. 마지막으로 교육생으로 하여금 MS사의 Visual C++ MFC 기반의 개발 환경에서 제어 분야대한 자가 학습이 가능하도록 샘플 응용 프로그램을 제공하였다. 본 결과물을 대학의 프로그래밍 교육 과정에 활용한다면 학생들로 하여금 흥미를 갖고서 자연스럽게 장비 제어 분야의 기초 지식을 습득할 수 있게 해줄 것이라 기대한다.

감사의 글

이 논문은 2016년 인하공업전문대학의 교내연구비 지원에 의하여 연구되었음.

Su Chul Hwang received the bachelor's, the M.S. and Ph.D. degree in the Department of Computer Science from the Chung-ang University in 1986, 1988 and 1993, respectively. He has been a professor in the Department of Computer Systems and Engineering at Inha Technical College since 1991. His main research fields include artificial intelligence, intelligent systems, internet application, IT fusion system etc. Recently he is interesting in an interface application programming based on Visual C++ MFC development environment. He is a life member of the KKITS.

E-mail address: schwang@inhatc.ac.kr

www.kci.go.kr