

PIC Micro-processor를 이용한 분산제어네트워크 설계

안비오\*, 강기원\*, 김재욱\*, 김민철\*\*  
 \*(주)마이다스코리아 기업부설연구소, \*\*인덕대학 메카트로닉스과

Design of Distributed Control Network system by Using a PIC Micro-processor

Pius Ahn\*, Kiwon Kang\*, Jae-Wook Kim\*, Min-Hyung Kim\*\*  
 \*MIDAS KOREA Re. Inst., \*\*Induk College

**Abstract** - Microchip사의 PIC Family 계열 마이크로프로세서 PIC16C74A를 이용하여 다수의 모터를 on/off 제어하는 컨트롤 보드의 제작과 함께 RS-485 Multi-drop 통신을 이용하여 분산제어 할 수 있는 프로토콜의 선정 및 상위 단 제어 솔루션 개발을 하였다. 프로그램은 하위단 컨트롤보드의 ASM과 함께 Visual BASIC 및 C<sup>++</sup>을 이용하여 상위단 MMI(Man-Machine Interface)를 설계하였다. 본 연구는 여러 개의 장치를 on/off 제어할 수 있는 분산제어네트워크를 경제적으로 설계함으로써 상용화 가능하다는 장점이 있다.

제안된 architecture는 상위단에 control client PC와 중간에 protocol 변환 장치 그리고, 하위단에 one-chip 이 장착된 control board로 구성된다. 하위단 control board에 장착된 one-chip 마이크로프로세서는 개발기간 및 개발비의 절감을 위하여 Microchip사의 PIC16C74A를 이용하였으며, 표1에 사양을 나타내었다.[1-6]

1. DCS 방식의 제어

대규모 시스템의 분산제어 기술은 다양한 통신방법의 개발과 함께 산업체에서 여러가지 방법으로 구현되어왔다. 최근 DCS(Distributed Control System)는 Fieldbus 방식을 이용한 Field point 장비의 이용과 함께 VME, PXI 방식 등이 광범위하게 이용되고 있다. 본 연구에서는 산업체에서 요구되는 저가형 분산제어네트워크 시스템을 그림1과 같이 제안하고자 한다.

2. Protocol의 설계

PIC16C74A는 다음의 그림2와 같은 PIN diagram을 가지며, 그림3과 같은 register file map을 갖는다.

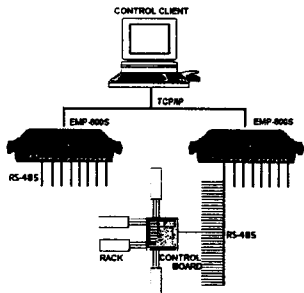


그림 1. 3-level architecture of DCS

표 1. PIC16C74A의 General Description

Maximum Frequency of Operation (MHz)	20
EPROM Program Memory (×14 words)	4K
Data Memory (bytes)	192
Timer Module(s)	TMR0, TMR1, TMR2
Capture/Compare/PWM Module(s)	2
Serial Port(s)	SPI/I <sup>2</sup> C, USART
Parallel slave Port	Yes
A/D Converter (8-bit) Channels	8
Interrupt Sources	12
I/O Pins	33
Voltage Range (Volts)	2.5-6.0
In-Circuit Serial Programming	Yes
Brown-out Reset	Yes
Packages	40-Pin Dip

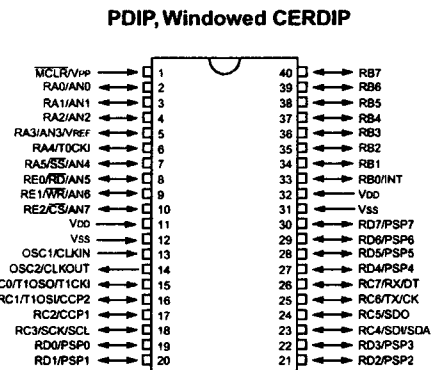


그림 2. PIC16C74A PIN Diagram

File Address	Register Name	File Address	Register Name
00h	INDF <sup>(1)</sup>	80h	INDF <sup>(1)</sup>
01h	TMR0	81h	OPTION
02h	PCL	82h	PCL
03h	STATUS	83h	STATUS
04h	FSR	84h	FSR
05h	PORTA	85h	TRISA
06h	PORTB	86h	TRISB
07h	PORTC	87h	TRISC
08h	PORTD <sup>(1)</sup>	88h	TRISD <sup>(1)</sup>
09h	PORTF <sup>(1)</sup>	89h	TRISF <sup>(1)</sup>
0Ah	PCLATH	8Ah	PCLATH
0Bh	INTCON	8Bh	INTCON
0Ch	PIR1	8Ch	PIE1
0Dh	PIR2	8Dh	PIE2
0Eh	TMR1L	8Eh	PCON
0Fh	TMR1H	8Fh	SPH
10h	T1CON	90h	TMR2
11h	TMR2	91h	PR2
12h	T2CON	92h	SSPBUF
13h	SSPBUF	93h	SSPADD
14h	SSPCON	94h	SSPSTAT
15h	CCP1TL	95h	CCP1H
16h	CCP1H	96h	CCP1CON
17h	CCP2CON	97h	TXSTA
18h	RCSTA	98h	SPBRG
19h	TXREG	99h	RCREG
1Ah	RCREG	9Ah	CCP2L
1Bh	CCP2L	9Bh	CCP2H
1Ch	CCP2H	9Ch	COPR2H
1Dh	COPR2CON	9Dh	COPR2CON
1Eh	ADRES	9Eh	ADCON0
1Fh	ADCON0	9Fh	ADCON1
20h		A0h	

7Fh Bank 0 Bank 1 FFh

그림 3. PIC16C74A 레지스터 화일 MAP

본 연구에서는 다량의 모터를 분산제어 할 수 있도록 3-level architecture를 설계하였으며, 하위단 control board를 통해 on/off 제어되는 대상은 DC24V motor 이고, limit switch를 job end sensor로 이용하였다. 분산제어를 위한 통신 프로토콜은 그림 4, 5, 6, 7과 같이 결정하였으며, 상위단 control client는 TCP/IP로 제어신호를 EMP-800장비로 전달하며, EMP-800장비는 신호를 선택된 RS-485통신 포트를 통해 하위단의 control board로 전달한다. 하위단의 control board는 전달된 신호를 독립적으로 protocol 규정에 따라 각각의 I/O port를 통해 제어대상 모터를 구동하며, job end signal을 받아 구동을 멈추게 한다.

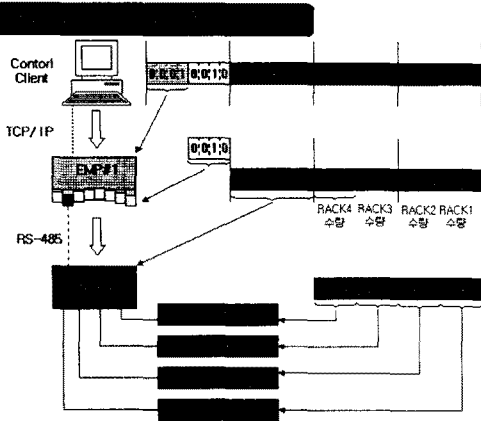


그림 4. Protocol 정의 : Client/EMP/PIC Down

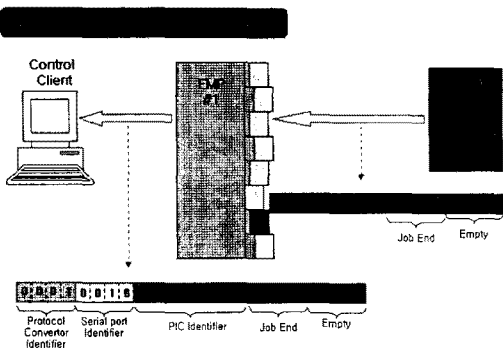


그림 5. Protocol 정의 : Client/EMP/PIC Up

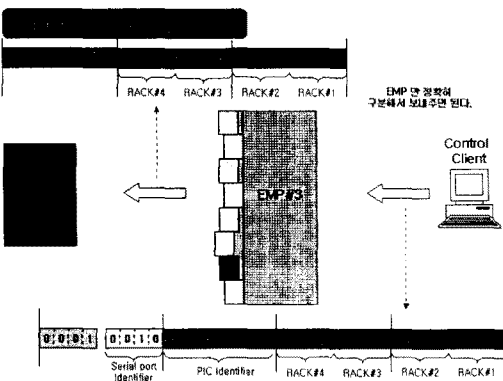


그림 6. Protocol 정의 : EMP/Client Down

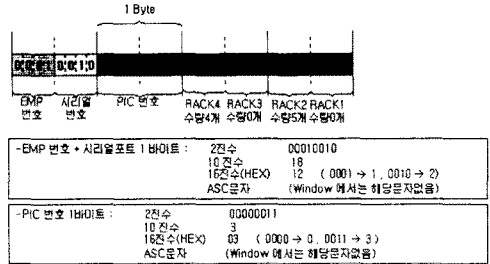


그림 7. Protocol 정의 : Client/EMP Up

### 3. PIC을 이용한 통신과 적용결과

시뮬레이션은 1개의 control client PC와 1개의 EMP-800 장비의 8개의 RS-485 port를 이용하였으며, 각각의 RS-485 port는 PIC이 장착된 32개의 control board를 갖는다. 여기서, 하나의 PIC은 4개의 motor를 control 하도록 설계되었다. PIC16C74A의 port A, B, C, D 중 각각 6, 4, 3, 4개를 이용하였으며, C port는 485 통신 port로 사용하고, A port는 PIC board를 구별하는 dip switch 입력단으로 사용하였다. Port B와 D는 각각 모터 구동신호와 limit switch를 통한 job end 신호의 입출력 port로 사용하였다. 따라서, 전체 제어 point 수는 8(EMP-800 ports)×32(1 port 당 PIC board 수)×4(1 PIC board 당 제어대상 수)=1,024 이다. EMP-800의 수를 n개로 하였을 때 분산제어 할 수 있는 control point는 n×1,024 만큼 늘어나게 된다.

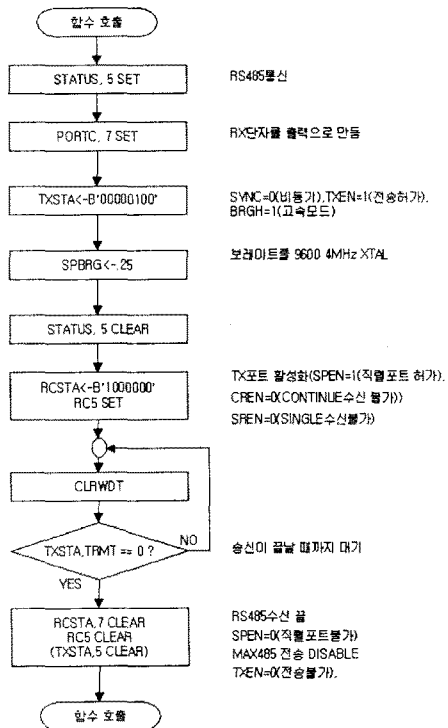


그림 8. RS-485 수신함수

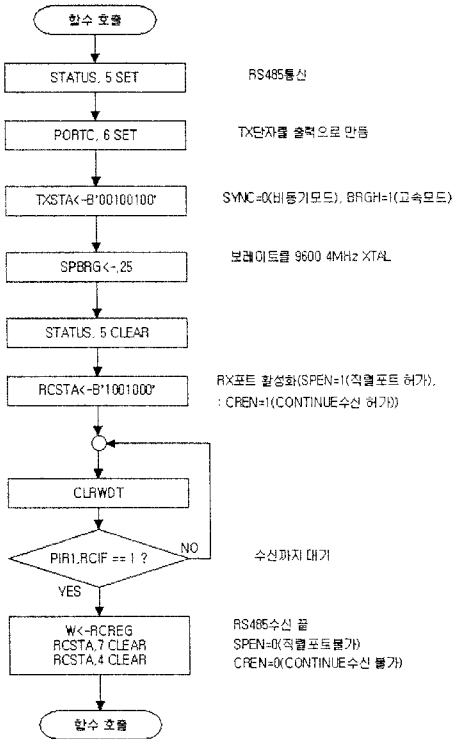


그림 9. RS-485 송수신함수

그림8과 9는 PIC16C74A를 이용한 RS-485 송수신 함수 설정의 flow-chart이며, 그림10은 실험에 사용된 4-Rack control board의 회로도이다.

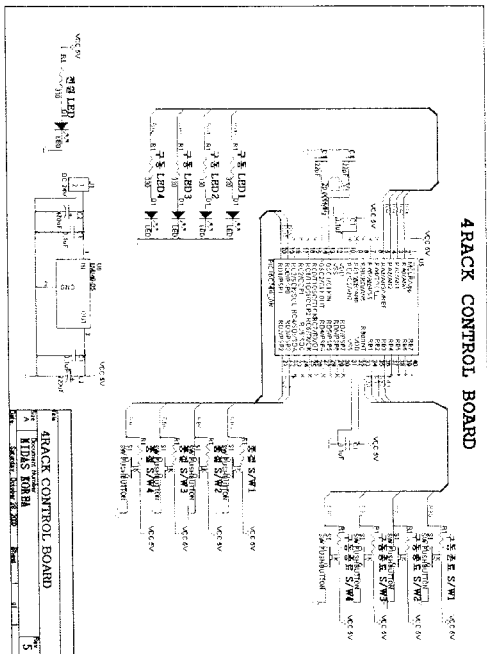


그림 10. 4-Rack Control Board 회로도

실험의 완성도를 높이기 위해 제작된 PCB는 그림11이며, protocol 변환과정과 데이터 송수신과정을 모니터링 할 수 있는 상위단 MMI(Man-Machine Interface) Windows는 그림12와 같다.

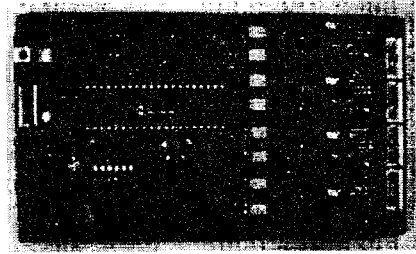


그림 11. 4-Rack Control Board

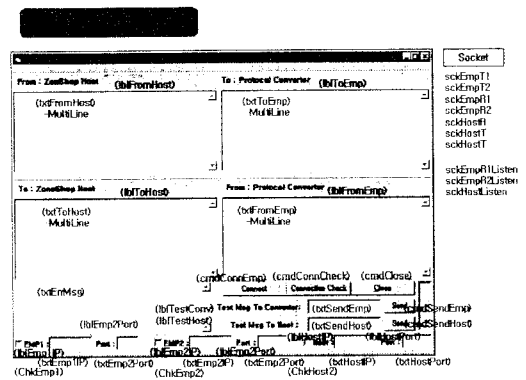


그림 12. 개발된 Man-Machine Interface Windows

#### 4. 연구의 고찰

제작된 DCS 시스템은 EMP-800에 10ms의 polling time을 주었으며, 6개월간의 실험운전결과 만족할 만한 결과를 얻을 수 있었다. 실제, PIC16C74A chip을 이용하여 on/off 제어를 할 경우, 하나의 PIC이 장착된 control board는 최대 16개의 control point를 가질 수 있으며, 이 경우 제안된 3-level architecture는 8×32×16으로 4,096 on/off control point가 가능하다. 또한 개발된 MMI는 원격으로 신호의 전달 및 응답 상태를 모니터링 함으로서 산업체에서 유용하게 활용될 수 있을 것이다.

#### [참 고 문 헌]

- [1] 1997 Microchip Technology Inc. DS30390E PIC16-C7X Data Sheet.
- [2] 진달복, "PIC16C84/71 마이크로 컨트롤러", 민음사, 1997.
- [3] 신철호, "PIC16C7X 테크니컬 핸드북", 컴파일테크놀로지(주), 2000.
- [4] 김민호 외 2인, "Visual Basic Programming Bible ver. 6.x", 영진출판사, 1998.
- [5] 이상엽, "Visual C++ Programming Bible ver. 6.x", 영진출판사, 1998.
- [6] (주)마이다스코리아 기업부설 연구소 2000년 연구개발보고서, MIDAS-KOREA/RR-0001/2000, 2000.