

다기능 다목적성을 갖는 산업용 제어기 설계

(A Design of Industrial Controller with Multi-function and Multi-purpose)

정 보 환* 남 진 문**
(Bo-Hwan Jung) (Jin-Moon Nam)

요 약

본 논문에서는 소량 다품종 환경에 적극 대처 할 수 있는 다기능, 다목적성을 갖는 산업용 제어기를 설계 제안하였다. 설계된 제어기는 모든 정보가 집중되는 MU(Main Unit)와 표시장치를 나타내는 DU(Display Unit)로 구성된다. 소프트웨어 관점에서 MU는 상위 태스크와 하위 드라이버로 구성되며, 각 태스크는 멀티태스킹을 지원 운영체제를 이용하여 병렬 처리된다.

설계된 제어기는 제어 알고리즘의 수정, 제어 대상 변경, 제어대상의 특성 변경, Man-Machine-Interface(MMI)의 변경 등에 능동적이면서 신속한 해결책을 제시하기 위하여, 제어기의 구조를 3개의 계층으로 구조화하였다. 1st layer인 하드웨어와 2nd layer인 펌웨어를 규격화하여 변경하지 않고 생산할 수 있으며, 각각의 제어 대상에 따라 달라지는 고유의 제어 알고리즘에 해당하는 3rd layer의 개발은 하위 2개의 계층에 관련 없이 독립적으로 신속히 개발될 수 있도록 하였다.

ABSTRACT

In this paper, we propose the industrial controller with multi-function/multi-purpose in order to cope with a small-amount and large-items environments. The controller designed consists of Main Unit including all of information and Display Unit. The software in the Main Unit is composed of tasks and device drivers and each task is being processed in parallel by operating system supporting multitasking.

The controller is structured in three levels to promptly address the control algorithm's modification, MMI's change, and so on. We can produce a controller without changing the first layer(hardware) and the second layer(firmware). We only modify the third layer(control algorithm) depending on control targets.

1. 서론

급격한 기술 발전, 시장 조건 그리고 빠르게 변화하는 네트워크 환경 등을 신속히 대응하기 위해 범용 구조를 가지는 디지털 제어기의 구현이 절실히 요구된다. 또한, 산업 특성상 대량 생산 보다는 점차 소량 다품종의 생산 체제로 변화하고 있는 실정이다. 이러한 산업 환경 변화에 적합한 디지털 제어기의 구현이 본 논문의 목적이다.

범용 목적을 갖는 디지털 제어기를 구현하기 위해 여기에서는 제어기의 구조를 3개의 계층으로 구조화하였다. 즉, 제1계층 하드웨어, 제2계층 펌웨어, 그리고 제3계층은 각각의 제어 대상에 따라 달라지는 고유의 제어 알고리즘 등으로 구성된다. 따라서 개발, 생산하고자하는 어떤 제어기 대상에 대해 1st layer인 하드웨어와 2nd layer인 펌웨어를 규격화하여 변경하

* 정회원 : 용인송담대학 정보통신과 교수
** 정회원 : 용인송담대학 정보통신과 겸임교수

논문접수 : 2001. 4. 20.
심사완료 : 2001. 4. 30.

지 않고 재생산할 수 있으며, 각각의 제어 대상에 따라 달라지는 고유의 제어 알고리즘에 해당하는 3rd layer의 개발은 하위 2개의 계층에 관련 없이 독립적으로 신속히 개발될 수 있도록 하였다.

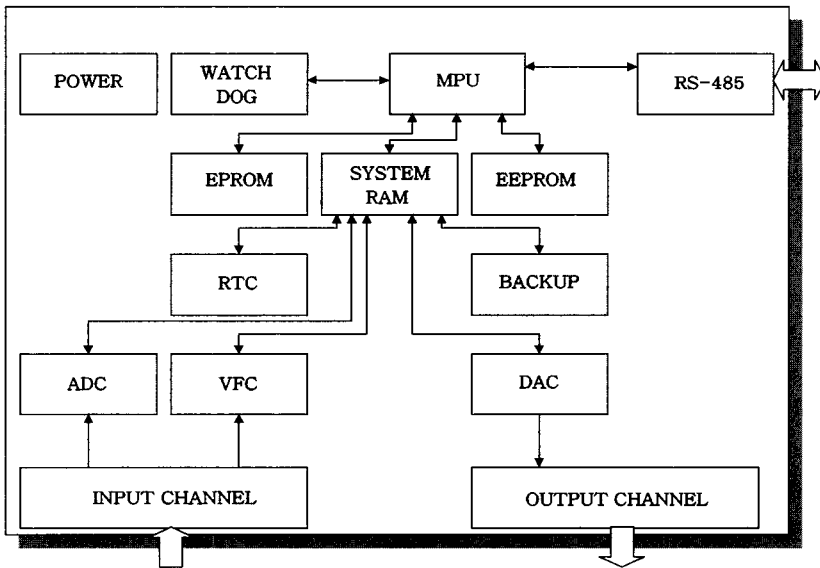
여기에서 제시하는 산업용 제어기의 가장 큰 특징은 범용성과 네트워크 환경에 적합한 구성을 들 수 있다. 즉, 어떠한 제어 사양을 요구하더라도 최소한의 시간과 비용으로 능동적으로 대처할 수 있을 뿐 만 아니라 네트워크를 용이하도록 설계하였다. 이는 모든 종류의 입력 채널과 출력 채널을 공용화 함과 동시에 데이터베이스 동적 구성을 자동으로 생성 가능하도록 설계하였다는 것을 의미한다. 즉, 사용자가 원하는 어떠한 입력이나 출력도 별도의 하드웨어적 변경 없이 동일한 채널로 처리가 됨으로서, 설계된 제어기는 원하는 제어 기기 시스템에 공통적으로 사용될 수가 있다. 또한 제품에 따라 변하게 되는 제어 사양에 따라 제어기 내부적으로 자동으로 데이터베이스를 생성하게 된다. 표시장치의 경우도 이 동적 데이터베이스에 따라 자동으로 화면을 구성함으로써 어떠한 제어 사양에도 자동으로 대처 할 수 있도록 설계하였다. 이러한 범용 및 공통성은 소량 다품종 체계라는 한계를 극복할 수 있는 수단이 된다. 이러

한 특성을 갖는 제어기는 하나의 제어기로 거의 모든 제어시스템에 적용 될 수 있다는 것을 의미한다. 본 논문의 2장에서는 하드웨어 기능 및 구성, 3장에서는 소프트웨어 기능 및 구성 그리고 4장에서는 결론을 맺는다.

2. 하드웨어 기능 및 구성

본 제어기는 고속 고성능 16-bit 마이크로프로세서인 Am188을 채택하여 정밀하고 빠르게 산업용 기기를 제어할 수 있도록 설계하였다. 또한, 제어 대상인 입력 채널 형태 및 출력 채널 형태를 제어 응용에 따라 프로그램이 가능하도록 설계하였다.

설계된 제어기는 2개의 Unit들(MU, DU)로 구성된다. 첫째, MU(Main Unit)는 모든 정보가 집중되어 있는 Unit, 그리고 DU(Display Unit)는 표시장치 Unit를 각각 나타낸다. 그리고 ON(Operator Network)은 MU와 DU사이의 통신기능을 담당한다. 다음은 MU 및 DU에 대한 주요 기능, 블록 다이어그램, 구현된 제어기를 <표 1>, <표 2>, [그림 1], [그림 2], [그림 3], [그림 4]에 각각 나타내었다.



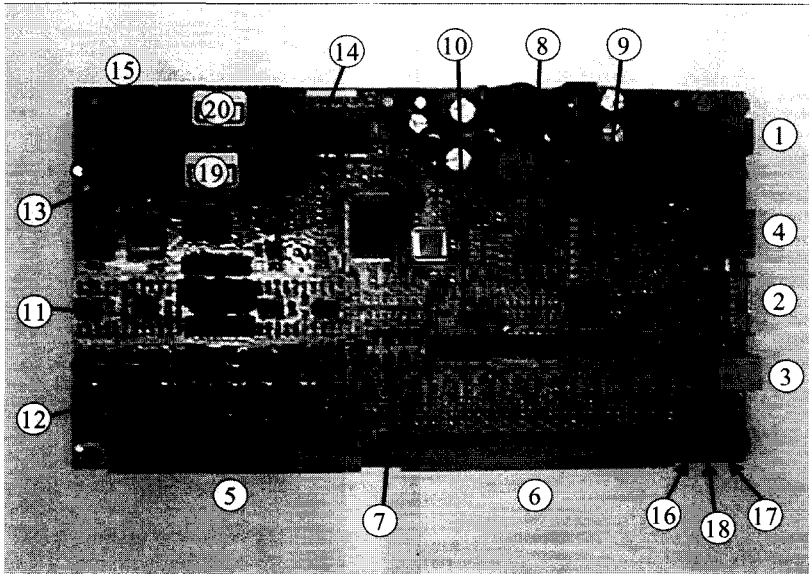
[그림 1] MU의 내부 블록 다이어그램

[Fig. 1] The Block Diagram of MU

<표 1> MU 하드웨어의 기능

<Table 1> The Function of MU hardware

항 목	기 능
제어 procedure	<ul style="list-style-type: none"> · 입력 채널 및 출력 채널 설정 · 소프트웨어 채널 설정 · DU을 위한 구조화된 테이블 생성 · 운전 일정 (월/일/시/분/요일) 설정
8 입력 채널	· 형태 구성
1 기준 전압	· 24VDC
실시간 클럭	· 시간 구성
Backup	· /년/월/일/시/분/요일, DU을 통한 설정
시스템 watchdog	<ul style="list-style-type: none"> · 시스템 에러 체크 · 시스템 재시작 · watchdog 타이머
Operator Network	<ul style="list-style-type: none"> · RS-485 I/F, 19200bps/ No Parity / 8bits / 1stop · 주소 설정 가능 · MAC(Media Access Control)
LED & Buzzer	<ul style="list-style-type: none"> · Buzzer 사용가능 · Power Status LED · ON 통신 LED



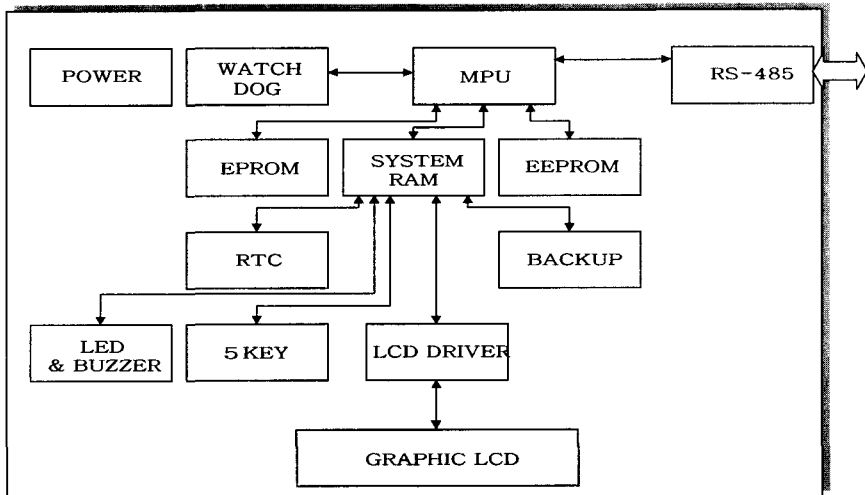
[그림 2] 구현된 Main Unit

[Fig. 2] The Implementation of MU

<표 2> DU 하드웨어의 기능

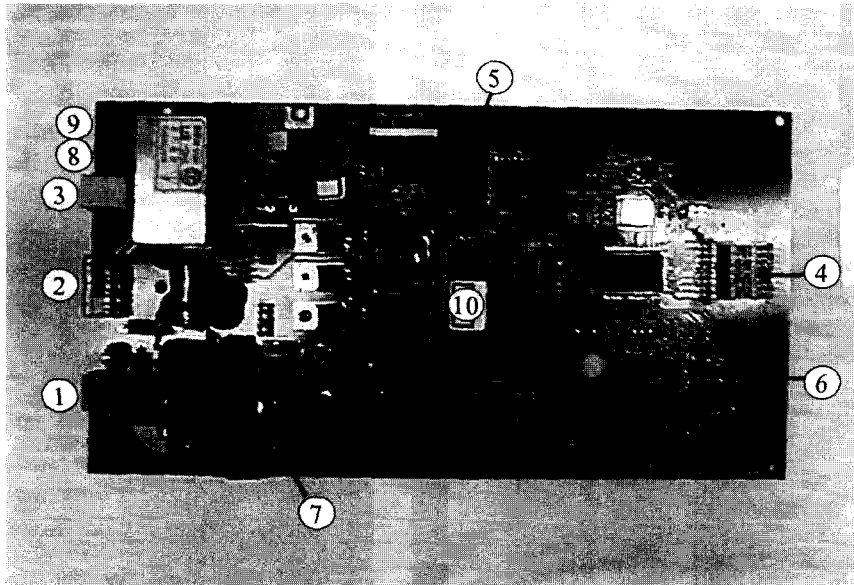
<Table 2> The Function of DU hardware

항 목	기 능
제어 procedure	· 입력 채널 및 출력 채널 모니터링 및 시험 · 운전 일정 (월/일/시/분/요일) 설정 및 수정
Real Time Clock	· 년/월/일/시/분/요일, 설정 및 Monitoring · Backup
System Control	· System Error Check · System Restart · System Monitoring
Operator Network	· RS-485 I/F / 19200BPS / No Parity / 8Bits / 1Stop · Address Switch 설정 기능 · MAC(Media Access Control)
LED & Buzzer	· Buzzer 사용기능 · Power Status LED · ON-통신 LED



[그림 3] DU의 내부 블록 다이어그램

[Fig. 3] The Block Diagram of DU



□림 4] 구현된 Display Unit

[Fig. 4] The Implementation of Display Unit

3. 소프트웨어의 기능 및 구성

설계된 제어기의 소프트웨어는 기본적으로 상위 태스크와 하위 드라이버로 구성된다. 각 태스크는 멀티태스킹을 지원하는 운영체제를 이용하여 병렬 처리된다. 병렬처리로 인해 부하가 큰 하나의 태스크로 인한 병목 현상 등의 현상을 예방할 수 있으며, 태스크의 추가 등에 유연한 환경을 제공한다.

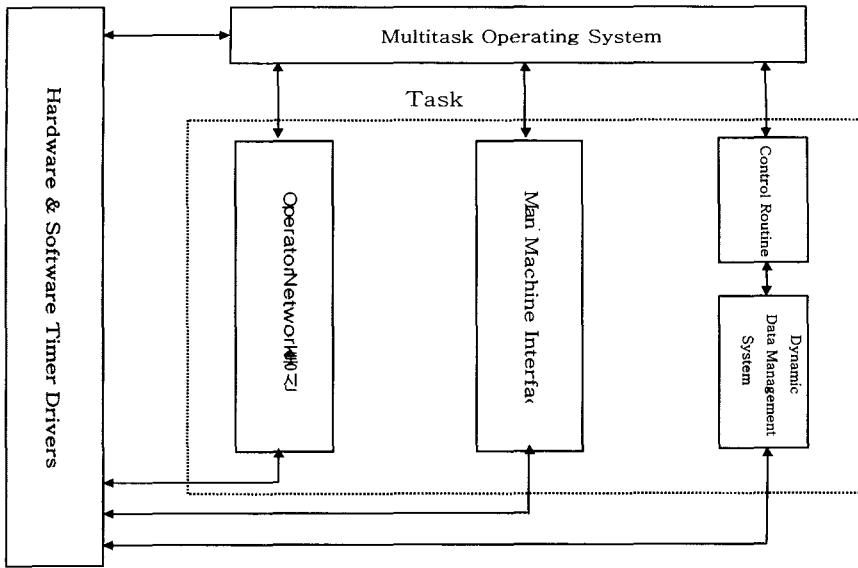
드라이버는 각각의 주변장치들과 소프트웨어 타이머의 동작에 대한 지원함수의 형태로 제공되며 태스크와 운영체제 서비스 제공을 위해 사용된다.

각각의 태스크들과 운영체제 그리고 드라이버와의 관계는 [그림 5]와 같다. 또한, 각 태스크가 담당하는 주요한 기능은 <표 3>과 같으며, 이는 보드별로 차이가 있으며 운영체제부와 드라이버부는 모든 보드에 기본적으로 탑재된다.

<표 3> 각 태스크의 기능

<Table 3> The Function of Each Task

Task	관련된 직업	탑재 Board
Man-Machine Interface(MMI)	<ul style="list-style-type: none"> · 제어 기기의 운전 상태 감시 · 제어 기기의 운전 및 제어 설정 · 운전일정 입력 및 수정 · 년/월/일 입력 및 수정 	DU
제어 루틴	<ul style="list-style-type: none"> · Control Routine 실행 	MU
동적 데이터 관리 시스템	<ul style="list-style-type: none"> · Database의 동적인 관리 	MU
Operator Network	<ul style="list-style-type: none"> · 통신 기능 제공 	MU / DU



[그림 5] 소프트웨어의 구조
[Fig. 5] The Structure of Software

3.1 운영체제부

통신부(ON)는 제어 루틴의 동작에 상관없이 항상 실시간으로 데이터를 전송이 하여야 하며 이를 구현하기 위하여 CPU 시간을 효율적으로 관리하여야 한다. Background에서 이들 태스크를 구조적으로 병렬 처리 실행하는 운영체제부가 필요하게 되는데 통상 사용되는 고가의 상품화된 것을 이용하지 않고 응용되는 제어 기기 특성에 맞게 자체 개발하여 탑재하였다.

본 운영체제부는 시간 슬롯 방식을 채택하여 규칙적인 데이터 통신에 효율적으로 대처하였으며, ON서 실시간 데이터 통신을 수행한다. 소프트웨어 타이머 및 Watchdog 타이머 기능이 있으며 타이머 ID는 하나의 태스크에 255개까지 등록이 가능하다.

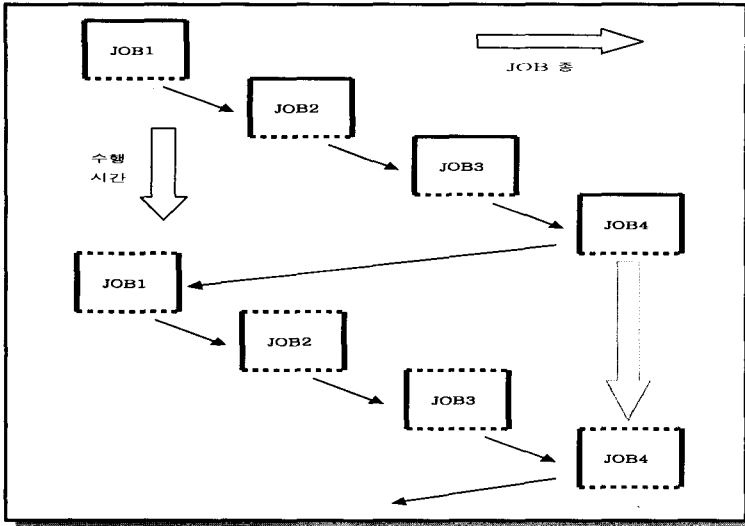
3.1.1 Job Schedule

서로 독립적인 여러 가지의 작업을 순차적으로 각각의 시간 슬롯을 실행시켜 주는 최상위 Background 프로그램인 JOB 스케줄러가 존재하는데 이는 운영체제의 근간이 된다. [그림 4]와 같이 JOB 스케줄러는

타이머 인터럽트 형태로 존재하며 타이머 관리 및 Watchdog 관리도 각각의 작업의 수행에 대하여 독립성을 보장한다.

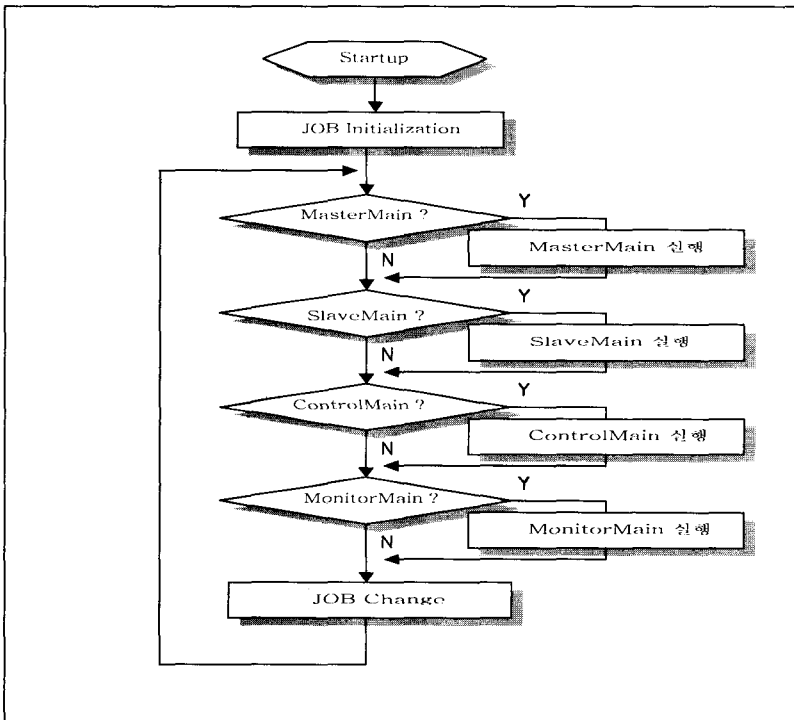
[그림 5]와 같이 JOB은 MasterMain, SlaveMain, ControlMain, MonitorMain 등으로 구성되는데, 각각의 JOB 프로그래머는 다른 JOB에 영향을 받지 않고 독립적인 구성을 할 수 있다. JOB 스케줄러는 각각 구성된 JOB을 Foreground에서 아래와 같이 할당된 시간 슬롯에서 나누어 실행시킨다. JOB이 실행되는 도중에 요구되는 부분 JOB이 발생할 경우는 하드웨어 인터럽트를 설정하여 JOB 스케줄러보다 우선적으로 처리 될 수 있도록 프로그램 한다.

만약 하드웨어 인터럽트에서 수행 중인 JOB이 많을 경우는 하드웨어 우선 순위에 따라서 수행 및 관리가 이루어진다. 따라서 하드웨어 인터럽트 수행 시간이 길어지면 다음 순서에서 대기 중인 JOB의 수행이 늦어져 문제가 발생할 수 있으므로, 가능한 짧은 서비스 루틴이 구성 되도록 한다.



[그림 6] JOB 스케줄러

[Fig. 6] JOB Scheduler



[그림 7] JOB 스케줄러 순서도

[Fig. 7] The Flow Chart of JOB Schedule

3.1.2 Device Driver

운영체제부 하위에 있는 드라이버는 소프트웨어 타이머, Watchdog, 스케줄러 등이 있으며 이들이 담당하는 주요한 기능은 <표 4>과 같다.

<표 4> 디바이스 드라이버 기능
<Table 4> The Function of Device Drive

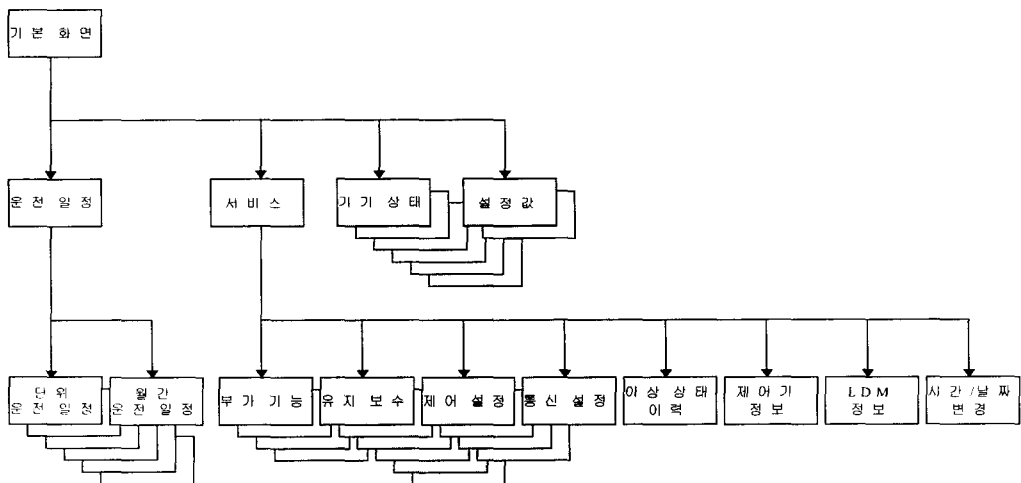
드라이버	기능
소프트웨어 타이머	<ul style="list-style-type: none"> · 타이머 등록 · 타임 아웃 체실시간이 크 · 타이머 취소 · 타이머 ID 관리
Watchdog 타이머	<ul style="list-style-type: none"> · 태스크 운용 모니터 · 시스템 다운 모니터 · 시스템 리셋
스케줄러	<ul style="list-style-type: none"> · 멀티태스크 운영 · 태스크 등록 · 태스크 제거
딜레이	· 시간 지연 기능

3.2 MMI(Man-Machine-Interface)부

본 제어기 MMI의 가장 큰 특징은 사용자가 DU Display 내용의 변경을 위해서 별도로 코딩 작업을 하지 않아도 되는데 있다. 사용자가 해주어야 할 유일한 일은 응용부분에서 원하는 종류의 테이블 선언과 선언된 테이블에 소속될 변수들을 그룹화하는 것이다. 여기서는 여러 가지 부가적인 기능 중 대표적인 것 만 기술한다.

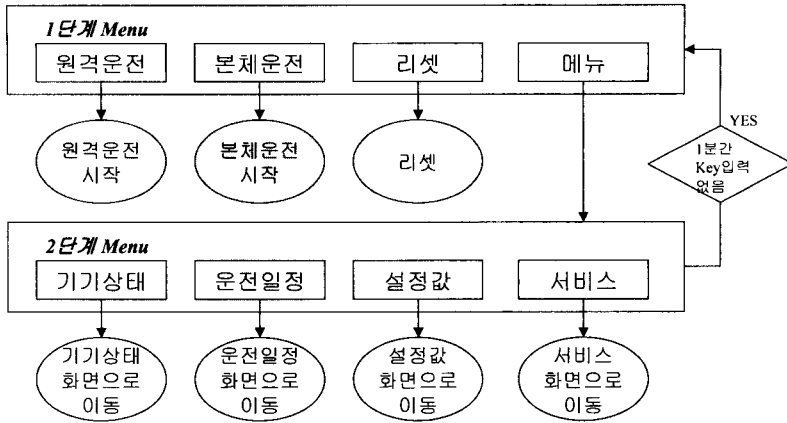
3.2.1 DU의 일반적인 기능

DU은 ON을 통해 네트워크상의 모든 장비와 통신을 할 수 있는 물리적인 조건을 만족하여야 한다. 아울러 MU 상태의 관측, 보수, 설정 기능을 제공하기 위해서는 MU에 논리적으로 부착되어야 한다. 즉, DU을 통신상의 마스터로 하고 MU을 통신상의 슬레이브로 하는 관계의 통신 설정이 가능해야 한다. 또한, 데이터베이스가 제공하는 모든 종류의 테이블 형태에 그룹화된 변수들에 대해 관측 기능을 제공하고 변수의 구성 정보를 참조하여 보수, 설정할 수 있는 기능을 제공한다.



[그림 8] DU의 메뉴 트리

[Fig. 8] The Menu Tree of DU



[그림 9] 기본화면의 메뉴 구성
 [Fig. 9] The Menu Structure of Basic Frame

3.2.2 기본 화면

기본 화면은 프로그램의 적재가 완료된 상태에서 갱신이 진행되면서 나타나는 초기 화면이다. 사용자의 키 조작이 없을 경우에는 항상 이 화면을 유지하게 되고 [그림 8]의 메뉴 트리에서 확인 할 수 있듯이 트리의 최상부에 존재하기 때문에 이 기본 화면을 통해서 다른 모든 화면으로의 접근이 가능하게 된다. 기본 화면의 구성은 [그림 9]와 같으며 1분 이상 키 입력이 없을 경우에는 다시 1단계 Menu로 복귀한다.

4. 결론

오늘날 제품의 다양화에 따라 제어 기술도 점점 정밀, 복잡해지면서 다품종 소량 생산화 되는 추세에 있다. 따라서 이전과 같은 전용 제어기는 다양한 제어 기능에 대해 신속, 유연하게 대처하기에는 많은 어려움이 뒤따른다. 즉, 제품 개발 기간 및 신뢰성 확보를 위한 시험기간이 많이 소요된다. 또한, 이로 인한 제품의 개발비용 및 생산비용의 증가에 따른 제품단가 상승 문제가 있어 합리적인 해결책을 제시할 수가 없었다.

따라서 본 논문에서는 제어 알고리즘의 수정, 제어 대상 변경, 제어 대상의 특성 변경, Man-Machine-Interface의 변경 등에 능동적이면서 신속한 해결책을 제시하기 위하여, 제어기의 구조를 3개의 계층으로 구조화하였다.

1st layer인 하드웨어와 2nd layer인 펌웨어를 규격화하여 변경하지 않고 생산할 수 있으며, 각각의 제어 대상에 따라 달라지는 고유의 제어 알고리즘에 해당하는 3rd layer의 개발은 하위 2개의 계층에 관련 없이 독립적으로 신속히 개발될 수 있도록 하였다.

※ 참고문헌

- [1] Jean Labrosse, "Micro-C/OS", R&D publication, 1992
- [2] Katsuhiko Ogata, "Designing Linear Control Systems with MATLAB", Prentice hall, 1994
- [3] Am188 User's Manual, AMD, 1997
- [4] Graphic LCD User's Manual, Powertip, 1997
- [5] 이강수 외, "운영체제 개념과 원리", 생능출판사, 1998
- [6] 엄영익 외, "컴퓨터 운영체제론", 생능출판사, 1999
- [7] 임인철 외, "C 프로그래밍", 도서출판 동광, 2000
- [8] 김태운, "데이터 통신과 컴퓨터 네트워크", 기한재, 2000
- [9] 진달복, "디지털 공학", 양서각, 2000
- [10] 김종식, "선형제어시스템 공학", 청문각, 1994
- [11] 송재철 외, "C/C++ 프로그래밍", 양서각, 1999
- [12] 김영규 외, "자료구조", 기한재, 1997

정보환



1987년 2월 경북대학교
전자공학과 졸업
1992년 2월 경북대학교 대학원
전자공학과 졸업
1998년 2월 경북대학교 대학원
전자공학과 박사과정 수료
1992년 1월~1998년 2월
대우중공업(주)
1995년 9월~1997년 2월
TRW(美) 다목적 실용
인공위성 자세제어계 개발
1998년 3월~현재 용인송담대학
정보통신과 교수

남진문



1987년 2월 경북대학교
전자공학과 졸업
1999년 2월 연세대학교
산업대학원 전자계산학과
1987년 2월~1988년 1월
(주)동양정밀
1988년 1월~1995년 1월
(주)금성사 중앙연구소
1999년 3월~현재 용인송담대학
정보통신과 겸임교수