

# 자동화 시스템내 셀 제어기의 통신 소프트웨어 개발 및 실험

도 성 희,\* 정 별 수,\*\* 박 경 진\*\*\*

## Communication Software Development and Experiments for a Cell Controller in a CIM System

Sung-Hee Do,\* Byeong-Soo Jung,\*\* Gyung-Jin Park\*\*\*

### ABSTRACT

The demand for automatic manufacturing systems is increasing. Flexible Manufacturing System(FMS) is usually considered as a solution for the shop floor automation. One of the difficulties in FMS is the communications problem. Since various machineries with different communications protocols are included, applying a unified scheme is almost impossible. Therefore, a systematic approach is a key point to solve the communication problem. A cell is defined as an automation unit where closely related for a job reside together. A cell is a messenger between upper level computers and lower level machine equipment. In this research, the functions of the cell are defined to have more capabilities than conventional cells since a cell can be often a total manufacturing system in a small to medium sized factory. The cell conducts communications with different machines through the communications schemes established here. A set of software system has been developed according to the defined communications. The software has been tested for a simulation and real experiments for proof.

**Key Words** : FMS, FMC, Cell Controller, DNC, Communication, Protocol

### 1. 서 론

최근 들어 제품을 생산하기 위한 생산기술은 여러가지 사회적, 기술적 문제를 안고 있다. 숙련기술자의 부족, 젊은 인재의 제조업 이탈, 제조설비의 고가화, 제

조업체의 다국적화, 소비자 욕구의 다양화 등으로 제조업 분야는 국가간의 경쟁이 치열해지고 있으며, 제조업 관계자는 그 적응을 위한 노력에 필사적이다. 이러한 요구를 충족시키기 위해서 생산시스템에서의 각 기능사이의 정보통합 체계에 대해 여러 모델이 제시되고 있

\* 한양대학교 기계설계학과 대학원

\*\* 두산기계주식회사

\*\*\* 한양대학교 공학대학 기계공학과

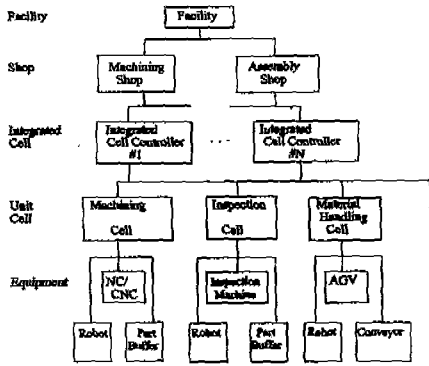


Fig. 1 CIM hierarchy

다.<sup>(1)</sup> 이들 모델들은 대개 계층구조로 이루어지며, 그 형태는 Fig. 1과 같이 나타낼 수 있다.

여러종류의 모델중 생산시스템의 대표적인 예로는 유연생산시스템(flexible manufacturing system, FMS)을 들 수 있다. 그 중에서 생산활동이 직접적으로 이루어지는 셀 레벨과 장비 레벨 즉 하부구조에 대한 연구는 현재 많은 연구가 진행 중에 있으며 이 부분을 셀 제어기(cell controller)라 부르기도 한다. 그러나 가변적인 생산환경에 적합한 셀 제어기를 구축하는데에는 시스템의 구성 및 운용을 담당하는 전략수립과 이종 기기와 컴퓨터간의 연결(interface) 체제 구축에 상당한 어려움이 따르게 된다. 따라서 유연생산시스템을 추구하는데 있어서 이 셀 제어기는 매우 중요한 역할을 담당하고 있으며 그 접근방법도 다양하다. 유럽지역에서는 필요한 기능들을 PC에 분산시킨 뒤 이들을 네트워크로 연결한 형태의 셀 제어기가 있으며,<sup>(2)</sup> 미국에는 계층구조를 따르기보다는 분산처리 능력을 향상시킨 셀 제어기라든가,<sup>(3)</sup> 각각의 작업간의 협조체제를 강조한 셀 제어기가 연구되고 있기도 하다.<sup>(4)</sup>

셀 제어기에 있어서 통신기능은 가장 기본적으로 제공되어야 하는 기술 중에 하나이다. 그러나, 셀 제어기와 연결되어 있는 각종 장비들의 컨트롤러(controller) 사양중에는 해당기기의 통신기기 및 통신방법에 대한 기술이 공개되지 않는 경우가 대부분이며 공개되어 있는 기술 또한 판독하기가 매우 어려운 실정이다. 또한, 셀 제어기를 이용한 이종 기기 간의 통합기술은 아직까지 상품화된 것이 없는 실정으로서 동일회사에서 제작된 공작 기기 들을 통합한 형태의 제품이 나와 있을 정도이다.<sup>(5)</sup>

본 연구에서는 셀 제어기 스스로가 유연성을 가지고

생산 활동을 직접 관장할 수 있는 능동적인 셀을 정의하고 정의된 '셀의 통신기능을 실현하는 프로그램을 개발하는데 목적이 있다. 셀의 통신기능은 두 가지로 요약될 수 있으며 그 첫 번째 기능은 셀 제어기와 다수의 CNC 공작기계 그리고 공장자동화 요소 기기를 RS-232C 등 통신회선으로 결합하여 가공프로그램을 관리, 전송하고, 기계의 상태 감시등을 수행한다. 두 번째 기능은 셀 내부의 각종 태스크와의 메시지 전달을 위한 통신기능으로서 셀 제어기의 자동운전에 필요한 기능이다. 개발된 통신 소프트웨어는 모듈화를 통하여 유연하고 효율성 있게 설계하였으며 새로운 이종 기기와의 연결시 최소한의 조정만으로 연결을 가능하도록 하였다. 또한 모의 실험 및 실제 기기들과의 실험을 통하여 정의된 셀 제어기의 타당성과 실효성을 검증하려고 한다.

## 2. 통신 및 셀 제어기에 대한 정의

### 2-1 자동화 시스템에서의 통신

시스템 상호간에 매개체를 통한 정보 교환을 통신이라고 하며, 정보 교환을 위한 규칙 및 정보의 신뢰도를 높이기 위한 모든 노력을 프로토콜(protocol)이라고 한다. 대량생산의 자동화 시스템에서 다품종 소량생산체제인 유연생산시스템으로 변화하는 시점에 있어서 정보 교환에 대한 중요성은 시스템 전체의 성능에 상당한 영향을 미치고 있는 것이 사실이다. 따라서 이러한 정보의 흐름의 효율적인 제어와 그에 따른 통신 시스템의 구축에 대한 연구는 컴퓨터 통합생산(computer integrated manufacturing, CIM)이라는 주제로 많이 수행되어지고 있는 실정이다.

Fig. 1에서 보여지듯이 컴퓨터 통합생산은 상부구조와 하부구조로 분류할 수 있으며, 상부구조에서는 생산에 필요한 모든 계획 및 정보의 관리 등을 담당하고, 하부구조에서는 수립된 생산계획에 따라서 작업의 처리 및 감시를 통한 생산을 담당한다. 상부구조에서는 생산에 필요한 모든 정보를 다루고 있기 때문에 많은 양의 데이터를 다루게 되며 주로 근거리 통신망(local area network, LAN) 기술을 이용한 네트워크(network)를 구성하여 통신 시스템을 구축한다. 상부구조에서 사용되는 기기들은 대부분 컴퓨터로 구성되는데 컴퓨터 자체가 통신에 맞게 제작되어 있으며 상품화된 제품을 이용하면 되므로 통신에 관한 한 큰 어려움은 존재하지 않는다.

컴퓨터 통합 생산의 하부구조에서는 셀 제어기 기능을 수행하는 컴퓨터와 공장내의 각종 장비들과의 통신이 이루어지며, 개별적인 생산 정보가 교환된다. 이들 기기와의 통신방법은 크게 세 가지로 나뉘어 지는데, 근거리 통신망을 이용하여 통신대상이 되는 기기의 콘트롤러에 직접 LAN 카드를 장착하여 통신을 시도하는 방법과 근거리 통신망을 이용하되 MAP(manufacturing automation protocol) 카드를 통신대상이 되는 기기에 장착하여 통신을 시도하는 방법 그리고, 기기가 지원하는 직렬 포트(serial port, RS232C/RS422/RS485)를 통하여 통신을 실현하는 방법이 그것이다. 근거리 통신망을 이용하는 경우에는 통신 속도나 통신 안정성 등은 높으나 콘트롤러가 다른 이 기종간의 연결은 불가능하고 같은 제조회사에서 제작한 콘트롤러들만 연결 가능하다는 단점이 있다. 이 경우 LAN 카드의 설계방식이나 통신방법에 대한 기술을 공개하지 않아 기술 종속화 현상을 유발하기도 한다.<sup>(6,7)</sup> MAP 카드를 이용하는 방법은 LAN 카드를 이용하는 방법과는 달리 콘트롤러가 다른 이 기종간의 연결이 가능하고 기기와의 통신을 위한 통신규약이 MMS(manufacturing message specification) 형태로 표준화 되어있기 때문에 LAN 카드를 이용한 방식보다 광범위하게 사용할 수 있다.<sup>(8)</sup> 그러나 위의 두 가지 방법들을 구현하기에는 많은 비용이 들기 때문에 중소기업체등과 같이 규모가 작은 업체에서는 채택하기 어려운 방식이다.

직렬 포트를 이용한 통신방법은 근거리 통신망을 이용한 방식에 비해 통신 속도나 통신 안정성은 그리 높지는 않으나 모든 기기마다 직렬 포트를 제공하기 때문에 이 기종간의 연결이 용이하고, 제조회사 마다 직렬 포트를 이용한 통신방법에 대한 기술을 설명서(manual)를 통해 공개하기 때문에 통신에 관련된 설명서만 있다면 통신을 위한 소프트웨어는 통신방법의 판독이 어렵더라도 제작이 가능하다. 접속해야 할 기기의 수가 많은 경우, RS232C를 사용하는 경우에는 multi-port 등을 이용하여 통신회선수를 증설할 수 있으며 RS422 혹은 RS485등을 사용하는 경우에는 multi-drop 방식을 사용하므로 통신 프로그램 상에서 통신 회선수를 증설할 수 있다. 이러한 통신방법은 프로토콜이 복잡한 경우와 단순한 경우, 두 가지로 구분할 수 있으며 로봇(robot), PLC(programmable logic controller), CNC등과 같이 프로토콜이 복잡한 경우에는 이들 기기와의 통신을 위해서 소프트웨어적으로 복잡한 과정을

거쳐야 하는데, 이를 프로토콜 변환기(protocol converter)라 지칭한다. 한편 NC등과 같이 공장내의 일부 장비들은 프로토콜이 단순하여 컴퓨터와의 통신이 복잡한 과정이 없이 간단히 이루어지도록 설계되어 있으며, 이러한 경우의 통신 소프트웨어를 디스패처(dispatcher)라고 명명한다.

컴퓨터 통합생산의 상부구조와 하부구조간의 정보교환은 컴퓨터간의 통신으로서 근거리 통신망을 사용하는 것이 보편적인 추세이다. 그 결과 Fig. 1에서 통합셀 레벨 이하를 유연생산 시스템으로 보는 경우가 지배적이며 이러한 유연생산 시스템의 한 유형으로 DNC(distributed numerical control) 시스템이 개발되고 있기도 하다.<sup>(9)</sup> 이러한 시스템들은 모두 컴퓨터와 다수의 CNC 공작기계 그리고 공장자동화 요소기기를 LAN이나 직렬포트등의 통신회선으로 결합하여 제어용 컴퓨터에서 가공 프로그램을 관리, 전송, 편집하고, 기계의 상태감시, 공구 데이터 관리, 생산 일정 및 관리 등 제반업무를 집중처리하는 기능들로 구성되어 있다.

본 연구에서는 하부구조의 통신에 초점을 맞추어 상부구조에서의 통신에 관한 부분은 다루지 않으며, 각종 기기와의 통신에 있어서 보다 유연성을 추구하고 중소기업 등에서도 쉽게 이용할 수 있는 저렴한 방식인 직렬 포트를 이용한 통신방식을 시도한다. 아울러 이 기종간의 통합을 통한 일반적인 셀 제어기를 개발하려 한다.

## 2-2 셀 제어기에 대한 정의

셀 제어기는 앞에서 언급한 바와 같이 장비 레벨에 마련되어 있는 몇가지의 기기들과 물리적으로 연결되어 이 기기들의 작동을 소프트웨어적으로 관리하는 역할을 담당한다. 즉, 하나의 셀은 실제 생산을 위한 단위기계군으로 이루어진 하나의 작업공간을 담당하는데, 작업활동의 특성에 따라 가공 셀(machining cell), 검사 셀(inspection cell), 물류 관리 셀(material handling cell), 또는 조립 셀(assembly cell)등으로 분류된다. 이렇게 몇 가지의 셀이 모여서 유연 생산 시스템이 구축되며 Fig. 1에서 보여지듯이 컴퓨터 통합생산의 상부구조로 부터의 생산 계획에 따라 작업 일정 처리, 작업의 진행 감시, 통신을 통한 기기들의 직접 혹은 간접적인 제어 등의 기능을 갖는다.

본 연구에서는 가공 셀의 역할을 담당하는 셀 제어기를 정의하고자 한다. 아울러 정의된 셀 제어기는 셀의 종류에 따라 쉽게 전이될 수 있도록 일반적인 구성을

갖도록 한다. 정의된 셀 제어기는 Fig. 2와 같으며 종래의 셀 제어기에 비해 현장의 변화에 쉽게 대처할 수 있는 유연성을 부여하였다.<sup>(9-11)</sup> Fig. 2의 셀 영역(cell domain)은 다중작업 오퍼레이팅 시스템(multitasking operating system) 예를 들어 UNIX, VMS, OS/2 등)을 갖는 컴퓨터이며 각 블록은 하나의 독립된 프로그램으로서 오퍼레이팅 시스템내의 태스크로서 존재한다. 셀 영역의 모든 기능은 몇가지 컴퓨터 프로그램들로 구성되는 하나의 소프트웨어 시스템으로 구현되며, 그 프로그램들은 셀의 제반 기능을 관리하는 제어 프로그램(control program) 군과 장비 레벨의 기기들과 직접 통신기능을 수행하는 통신 프로그램(communication program) 군으로 구분할 수 있다.

모든 태스크는 각자의 고유한 기능을 가지며, 여러 태스크가 서로 상호 작용을 하여 전체적인 셀의 기능을 수행한다. Fig. 2에 정의된 각 태스크의 기능을 간략히 설명하면 다음과 같다.

(1) 컨트롤 프로그램 군(control program group)은 셀 제어기의 운영 및 제어를 담당하며, 컨피규레이션 태스크, 스케줄러 태스크, 작업 자동실행 태스크 등으로 구성되어 있다.

(2) 통신 프로그램 군(communication program group)은 작업 기기들과의 통신을 수행하는 기능을 가지며, 작업개시 명령 등을 발송한다든지 기기의 현재 상태 등을 검출하는 역할을 담당한다. Fig. 2에는 표시되지 않았으며 디스패처 태스크와 프로토콜 변환기 태스크로 구성되어 있다.

(3) 사용자 인터페이스 태스크(user interface

task)는 단말기를 통해 각 기기와 일정한 상태를 오퍼레이팅 시스템에 증속적인 그래픽 도구를 통해 시각적으로 보여줌으로써 사용자가 원하는 태스크의 실행이나 작업의 상태확인 등을 가능하도록 한다.

(4) 컨피규레이션 태스크(configuration task)는 셀 제어기 내부의 모든 태스크들의 동작상황을 검사하는 기능을 담당한다. 아울러 각각의 태스크들의 생성 및 진단기능등을 수행한다. 본 태스크는 현재 개발되어 있지는 않다.

(5) 스케줄러 태스크(scheduler task)는 셀 영역의 작업계획(operation scheduling)을 담당한다. 셀 내로 할당된 작업들을 실행하기 위한 작업일정을 수립하는 중요한 역할을 맡는다.

(6) 작업 자동실행 태스크(automatic cell operation task, ACO)는 수립된 작업 일정계획에 따라 작업을 실행, 감시하는 기능을 갖는다. 이 태스크는 셀 제어기의 운용에 가장 중요한 태스크로서 셀 내의 생산 활동을 총괄하여 관리한다. 스케줄러 태스크, 디스패처 태스크 및 프로토콜 변환기 태스크와 작업정보를 교환하여 작업의 실행 및 감시기능을 수행한다.

(7) 통신 프로그램 제어기 태스크(communication program controller task, CPC)는 셀 제어기 내부의 독립적인 태스크들과 정보전달을 수행하는 기능을 담당한다. 즉, 셀 제어기가 처리해야 할 일정을 수행하는 과정에 있어서 내부통신 기법을 사용하여 태스크간 정보처리를 수행하는 것으로서 이는 근거리 통신망을 이용한 정보처리기능과 유사한 효과를 얻을 수 있다.

(8) 디스패처 태스크(dispatcher task)는 단순한 프로토콜을 갖는 기기와의 통신을 담당한다.

(9) 프로토콜 변환기 태스크(protocol converter task)는 다소 복잡한 프로토콜을 갖는 기기와의 통신을 담당하는 역할을 수행한다.

(10) 상부구조 인터페이스 태스크(higher level interface task)는 통합셀과의 연결을 담당한다. 이 태스크를 통해 상부구조에서 내려오는 정보를 전달받는다. 본 태스크는 현재 개발되어 있지는 않으며 여러 셀 제어기를 통합하는 통합셀 제어기를 구축하려 할 때 개발 예정에 있다.

(11) 각각의 태스크들은 데이터 베이스의 정보를 토대로 운용된다.

(12) 독립적인 여러 태스크와의 상호 정보교환을 위해서 각각의 태스크와의 통신은 오퍼레이팅 시스템이

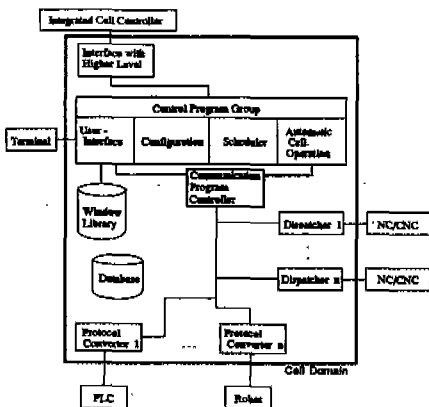


Fig. 2 Cell controller configuration

Table 1 Organization of the database

Directory Name	Description	Tasks Concerned
/Configuration_Data	Data for the configuration of a cell	Configuration, User-Interface
/Schedule_Data	Data for scheduling(Data on jobs & operation)	Scheduler, Automatic Cell Operation
/Operation_Status	Information of current status of operations	
/Machine_Data	Information of machines	Dispatcher, Protocol Converter, Automatic Cell Operation
/Communication_PGM	Communication program with machines	
/NC_Data	NC Part programs	
/Robot_Data	Robot programs	
/PLC_Data	Ladder Logic programs	
/ICON_Data	Information for the display	User-Interface

제공하는 내부통신(interprocess communication, IPC) 방법을 사용한다. 이는 태스크간의 정보교환이 특별한 장비(예를 들어 RS232C)를 필요로 하지 않으며 오퍼레이팅 시스템이 동작하는 동안에는 언제든지 그 서비스를 제공받을 수 있기 때문이다.

셀 제어기에 적합한 데이터 베이스를 구축하는 방법은 상용 소프트웨어를 사용하는 등 여러 가지 방법이 있을 수 있으나 본 연구에서는 데이터의 특성을 감안하여 오퍼레이팅 시스템이 제공하는 디렉토리(directory) 구조를 이용하여 조직적이고 체계적인 데이터 베이스를 정의하였다. 이렇게 함으로서 추후에 상용 데이터 베이스를 사용할 경우 쉽게 전이가 가능하도록 한 것이다. 본 연구에서 정의된 셀 제어기에 필요한 데이터 베이스의 구조는 Table 1에 표시하였는데 그 내용을 살펴보면, 제어하고자 하는 정보의 성격에 따라 디렉토리 별로 구분하였으며 그 안에 필요한 정보를 저장하였다. 예를 들어 /Machine\_Data라는 디렉토리에는 각종 기기들과의 통신을 시도할 때 기기들과의 통신방법에 필요한 인자(parameter)들을 저장하였고 관련된 태스크로는 작업 자동실행 태스크와 프로토콜 변환기 태스크 및 디스페처 태스크가 있다.

### 3. 통신 소프트웨어 설계

Fig. 2와 같이 정의한 셀 제어기의 작동을 살펴보면, 먼저 스케줄러 태스크로부터 작업 일정에 대한 계획을 수립하고 수립된 정보를 바탕으로 작업 자동 실행 태스크에서는 정해진 시점이 되면 실행되어야 할 작업

에 대한 정보를 통신프로그램 제어기에 전달한다. 통신 프로그램 제어기는 작업 정보를 해석하여 해당 통신 프로그램을 활성화시켜 지시된 작업을 실행시킨다. 일부의 통신 프로그램들은 실행된 작업의 진행 상황을 모니터링 하여 통신 프로그램 제어기에 전달하며 통신 프로그램 제어기는 모니터링 정보를 작업 자동 실행 태스크에 전달한다.

본 장에서는 Fig. 2의 셀 제어기 중 통신 프로그램군과 통신 프로그램 제어기 태스크를 제작하는 방법을 설명한다. 셀 제어기내로 할당된 작업들에 대하여 장비 레벨에 등록된 기기들의 여건에 맞는 작업 일정계획을 수립하는 스케줄러 태스크(scheduler task)와 수립된 일정 계획에 따라 작업들을 자동적으로 실행, 감시하는 작업 자동 실행 태스크(automatic cell operation task)를 포함하는 컨트롤 프로그램군에 대한 내용은 본 논문의 범위를 넘는 것으로서 이미 개발된 제어 프로그램을 이용한다.<sup>(12)</sup>

#### 3-1 기계 관련 데이터 베이스의 구축

Fig. 2에서 정의된 데이터 베이스는 디렉토리 구조를 이용하였으므로 각각의 정보들은 ASCII 코드로 구성된 일반 문자화일(text file)형태 및 실행 화일(executable file) 형태로 저장될 수 있다. 이 특성을 이용하여 데이터 베이스 내부에 직접 실행화일 까지 저장한다면 아주 작은 단위까지도 프로그램을 모듈화 할 수 있다는 장점을 얻을 수 있다.

Table 2는 셀 제어기 중 본 연구의 개발 대상이 되는 디스페처 태스크와 프로토콜 변환기 태스크의 입력

정보와 출력정보를 보관하는 기기관련 데이터 베이스 항목을 표현한 것이다. 이들 항목을 살펴보면 기기명, 기기의 상태, 통신을 위한 직렬 포트명, 통신사양 등의 정보가 입력되어 있다. 7번 주 통신 프로그램명 항목이 하에 저장되어 있는 내용은 해당 기기가 보유하는 컴퓨터와의 통신기능(예를 들어, down load, up load, 기기상태 검사 등)들로서 기기마다 제공하는 내용과 기능 및 종류가 다른 구성을 이루고 있다. 프로토콜 변환기 태스크와 같은 경우 기기가 제공하는 컴퓨터와의 통신기능으로는 보통 가공시작(cycle start), 기계상태 검사(status check), 프로그램 전송(down load, up load) 등과 같이 여러 형태가 있을 수 있으며 그 종류 또한 기기마다 다른 특성을 보유하는 것이 보통이다. 따라서 이들 항목들을 적절히 모듈화 하여 데이터 베이스

에 저장한다면 이종 기기라도 쉽게 셀 제어가 관장하는 통신 프로그램군의 일원으로 등록할 수 있다.

각종 기기 구입시 제공되는 설명서 중 통신기능에 대한 설명을 참조하면 해당 기기가 제공하는 통신기능을 구분할 수 있으며, 이들을 기능별로 실행화일을 제작하는 작업은 전체 프로그램을 개발하는 것보다는 비교가 안될 정도로 간단하다. 이렇게 제작된 기능 프로그램들을 Table 2에 정의된 구조대로 저장하여 이를 사용자 인터페이스 태스크에서 이용한다면 기기가 달라지는 경우나 새로운 기능이 추가되는 경우라도 사용자 인터페이스 프로그램 등을 변경하지 않고도 쉽게 메뉴체계를 구성하여 통신 프로그램 군을 관리할 수 있다. Table 3은 각종 기기와 컴퓨터와의 통신기능들을 정리한 것으로서 현장에 있는 각종기기들은 이들 기능들 중 몇 가지 혹은 전부를 지원한다.

Table 2 Definition for the machine database

No	Attribute	Description
1	MCID	machine ID
2	MACHDES	machine description (description for the machine & controller unit, Machine grade, machine size, load limit)
3	MACHST	machine status code (machine status, repair time expected at the break-down time)
4	ICON	ICON image drawing file name
5	CPORT	serial port name
6	PROTOCOL	communication protocol (RTS/CTS hardware handshaking, XON/XOFF flow control, response time, baud rate, stop bits, data bits, parity check)
7	COMM_PGM	communication main program name
8	FUNC_ACO	function program name to execute when Automatic Cell Operation is activated
9	NO_FUNC	group number of available function programs
10	NO_F1SUB NO_F2SUB ...	function program number of group 1 function program number of group 2 ...
11	FUNC1 FUNC2 ...	function program names of group 1 function program names of group 2 ...

### 3-2 통신 프로그램

통신 프로그램은 공장내의 각종 장비들의 프로토콜에 따라 기기 들과의 통신을 통해 해당 장비들이 지원하는 기능을 소프트웨어적으로 구현하기 위한 프로그램이다. 셀 제어기에 장착될 통신 프로그램의 설계시에 주요한

Table 3 Function definition for communication

Function	Description
Cycle Start.	Machining process start automatically
Cycle Stop	Machining process stop automatically
Down Load	Machining program transfer from computer to machine
Up Load	Machining program transfer from machine to computer
Program Search	Search the machining program in the machine's memory
Program Delete	Delete the machining program in the machine's memory
Status Check (Data Inquiry)	Machine status check(read)
Status Setting (Data Setting)	Machine status initialize
Offset Up Load	NC tool offset data transfer from machine to computer
Offset Down Load	NC tool offset data transfer from computer to machine
Emergency Stop	Machining process stop immediately
Register Read	PMC register information read

관점은 다음의 두 가지로 집약될 수 있다. 하나는 공장 내 설비의 변경에 대비하고 전체 셀 제어기와 간단히 통합될 수 있어야 한다. 다른 하나는 통신 프로그램을 통한 기기에 대한 접근은 사용자가 사용자 인터페이스 태스크를 통하여 통신 프로그램을 대화식으로 실행하여 접근하는 방법과 작업 자동 실행시에 통신 프로그램 제어기에 의해 통신 프로그램을 실행하여 작업을 자동으로 처리할 수 있어야 한다.

본 연구에서는 이러한 관점을 토대로 Fig. 3과 같이 통신 프로그램을 구성하고자 한다. Fig. 3에서 보여지듯이 먼저 실행 모드의 이원화를 기본으로 주 프로그램(main program), 기기의 각 기능을 구현하는 기능 프로그램(function program), 작업자동 실행과 연계되었을 때 실행되는 기능 프로그램(function program with ACO)등으로 기능과 실행에 따른 모듈화를 통해 프로그램의 유연성(flexibility) 및 확장성을 극대화하였으며 그 구성은 Table 2에 설명한 바와 같이 /Machine\_data 디렉토리내의 데이터 베이스에 저장되어 있다. 이를 바탕으로, 통신 프로그램이 사용자 인터페이스에 의해 독자적으로 실행될 때는 사용자가 대화식으로 통신 프로그램의 원하는 기능을 순차적으로 실행할 수 있도록 하고자 한다. 작업 자동 실행과 연계되어 실행될 때는 통신 프로그램 제어기에 의해 작업 자동실행과 연계되었을 때 실행되는 기능프로그램이 활성화되어 작업 자동실행 태스크와 통신 프로그램 제어기 그리고 통신 프로그램 군 등이 내부적으로 정보를 전달하며 셀 제어기가 운용된다. 또한 기기의 작업 상태의 감시가 가능하다면 모니터링 정보를 통신 프로그램 제어기에 전달하게 된다. 모니터링에 대한 내용은 뒤에서 다루기로 하겠다.

### 3-3 통신 프로그램 제어기(communication program controller, CPC)

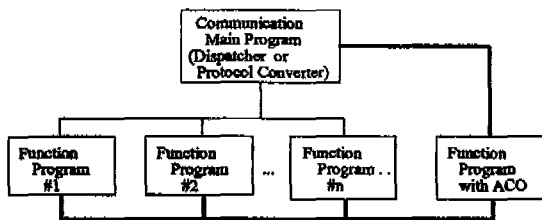


Fig. 3 Construction of the communication program

각각의 태스크간의 정보교환을 원활히 지원하기 위해서는 오퍼레이팅 시스템이 제공하는 내부통신방법(interprocess communication method)을 사용함으로써 그 문제를 해결할 수 있다. 태스크간의 내부 통신을 위해서는 프로그램 개발자가 정의하는 통신규약이 필요하게 되는데 본 연구에서 정의한 내부 통신 규약은 Table 4에 표시하였다. 각각의 항목을 설명하면 head는 통신 패킷의 시작부를 의미하고 section code는 통신 패킷을 발송하는 태스크를 data field는 보내고자 하는 정보를 의미한다. 정보의 종류로는 대상 기기명(machine id), 부품 번호(part id), 가공 프로그램 번호(program id) 및 각종 정보(information)들이 있을 수 있다. 마지막으로 tail은 통신 패킷의 끝부분을 의미한다.

통신 프로그램 제어기는 Fig. 2에서 보여지듯이 셀의 작업 자동 실행시에 통신프로그램 군들과 작업 자동

Table 4 Definition for the interprocess communication packet

Field	Description
Head	STX(0×02)
Section Code	Name of the Task which sends Message
Data Field	Machine_ID, Part_ID, Program ID, Lot Size, Information, etc.
Tail	ETX(0×03)

실행 태스크간의 연결을 담당하고 있는데 Fig. 4에서와 같이 통신 프로그램 군들과 작업 자동 실행 프로그램간의 정보의 흐름을 제어한다. Fig. 4에서 보여지듯이 통신 프로그램 제어기는 작업 자동 실행 태스크로부터의 기기명 및 각종 정보 등의 작업 실행에 관한 정보를 Table 4에 정의된 바와 같은 형태로 받아서 이를 바탕으로 개별적인 통신 프로그램을 실행시키고, 이 통신

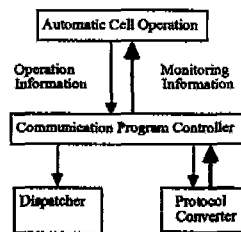


Fig. 4 Message flow between tasks

프로그램에 의해서 직접 혹은 간접적으로 필요한 기기가 작동 된다. 또한 작업의 시작 및 종료, 작업의 진행 상태 등의 모니터링 정보를 모니터링이 가능한 프로토콜 변환기로부터 받아서 이를 해석한 후 필요에 따라서 작업 자동 실행 태스크에 보내 준다.

### 3-4. 모니터링 (monitoring)

자동화 시스템에서의 모니터링은 생산 작업의 자동화에 필수적인 요건으로서 자동화 시스템의 성패의 한 요인이 되고 있다. 모니터링을 위한 방법은 여러 가지가 있을 수 있는데, 기기의 콘트롤러에 직접 LAN 카드와 같은 장비를 부착하여 모니터링 정보를 수행하는 방법<sup>6)</sup>, 기기의 콘트롤러 내부의 특정 부분과 PLC를 연결하여 PLC를 통한 기기의 콘트롤러 내부정보를 추출해 내는 방법<sup>13)</sup> 그리고 기기에 장착되어 있는 직렬 포트를 이용하여 컴퓨터와 기기와의 통신을 통해 모니터링 정보를 추출하는 방법 등이 있을 수 있다.

본 연구에서는 이중 기기 간의 통합이라는 관점을 충족시키고자 기기에 장착되어 있는 직렬 포트를 이용하여 모니터링 정보를 추출하는 방법을 시도하고자 한다. 셀 제어기에서의 기기의 작업 실행 정보 및 작업 상태의 감시 정보 등의 모니터링은 프로토콜이 단순한 기기와 복잡한 기기 별로 다음과 같은 전략 하에 수행하고자 한다.

프로토콜 변환기의 대상이 되는 프로토콜이 복잡한 기기 들은 일반적으로 기기와의 통신을 통해서 작업을 실행시키고 실행된 작업의 진행을 감시할 수 있으며 실행되고 있는 작업의 중지도 가능하다. 즉 컴퓨터의 통신기능 만으로도 해당 기기의 상태라든지 현재 작업상황들을 감지할 수 있다. 따라서 프로토콜 변환기의 경우 모니터링 정보는 프로토콜 변환기의 자체적인 기능에 의해서 수집할 수 있다.

디스페처의 대상이 되는 프로토콜이 단순한 기기 들은 해당 기기의 작업에 대한 모니터링이 불가능하다. 이는 해당 기기와 컴퓨터와의 통신 기능 자체가 제한을 가지고 있기 때문이며, 본 연구에서는 작업자 앞에 터미널 (terminal, PC)을 두어 필요한 사항을 작업자가 직접 모니터링 정보를 입력해 주는 방식을 구상하였다. 여기에서 터미널은 공장 내에 있는 기기로 생각할 수 있으며 이들 터미널과 연결되는 셀제어기내의 통신 태스크는 프로토콜 변환기로 사용할 수 있다. 디스페처의 대상이 되는 기기들의 모니터링 정보를 추출하는 방법

으로 또한 가지 생각할 수 있는 것은 기기를 제어하는 콘트롤러와 PLC의 접점을 연결하여 각 기기의 작업의 시작과 작업의 진행 상황을 PLC를 통하여 모니터링 하는 방법도 있을 수 있다. 이 경우에는 PLC를 담당하는 프로토콜 변환기가 따로 필요하게 되며, 이 프로토콜 변환기에 의해서 모니터링 정보가 통신 프로그램 제어기로 전달된다.

## 4. 소프트웨어의 개발 및 실험

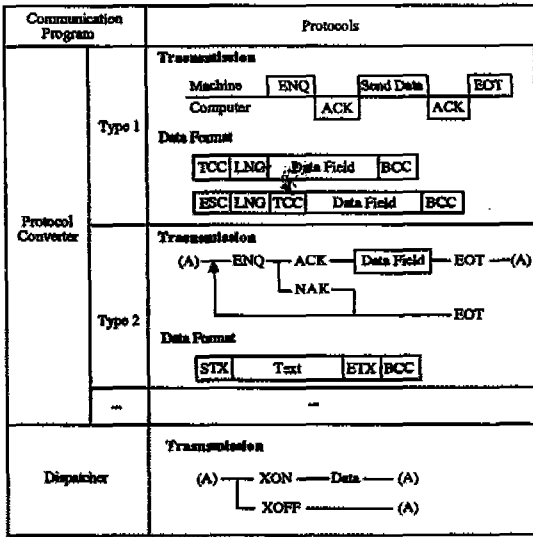
본 소프트웨어에서는 오퍼레이팅 시스템 내에 여러 개의 태스크가 동시에 독립적으로 수행되어야 한다. 그러므로 다중작업 (multitasking)이 가능한 오퍼레이팅 시스템 중 가장 널리 보급되어 있는 유닉스 오퍼레이팅 시스템 (UNIX operating system)을 개발 환경으로 하였다. 또한, 통신 프로그램 제어기에서 반드시 필요한 태스크간의 통신 즉, 프로세스 (process)간의 통신 수단으로는 UNIX가 제공하는 명명된 파이프 시스템 (named pipe system)을 사용하였다.<sup>(14,15)</sup>

사용자 인터페이스 태스크는 UNIX상에서 사용 기종에 관계없이 모두 통용되는 이식성이 아주 높은 그래픽 도구인 X 윈도우 (X-window system)를 이용하여 구축하였다. 프로그래밍 언어로는 C 언어를 사용하였다. 한편, 통신의 매개체가 되는 직렬 포트의 제어 기법은 UNIX 시스템에서 지원하는 문자 장치 구동기 (character device driver)를 이용하였다.<sup>(16)</sup>

### 4-1 통신방식의 분류

셀 제어기에 장착되는 통신 소프트웨어는 통신 방식에 따라 프로토콜 변환기와 디스페처로 분류된다는 것은 이미 설명한 바가 있다. 대개의 경우 통신은 컴퓨터와 기기 양측이 서로 특정한 약속을 하여 통신을 수행해 나가는 순서를 정해 놓고 이 순서에 근거하여 순차적으로 통신을 시도해 나가며 통신 중간에 순서에 어긋나는 정보라든가. 간섭에 의해 잘못된 정보가 제공되는 경우에는 오류 코드를 출력하여 정상상태로 되돌아갈 수 있는 체계로 되어 있다. 한번에 전송되는 정보들도 여러 가지의 정보를 혼합하여 암호화된 형태로 만든 뒤 이 정보를 전송하는 방법이 주로 사용된다. 이렇게 암호화시키는 이유는 통신과정시에 발생할 수 있는 오류를 쉽게 발견할 수 있도록 하기 위하여 하는 것으로 데이터를 전송 받으면 반드시 암호화된 데이터들을 검사





TCC : Transmission Control Character  
 LNG : Number of Characters  
 Data Field : Information Characters  
 BHC : Block Head Character  
 BCC : Block Check Character  
 ESC : Expanded ESCAPE

ACK : Acknowledge  
 ENQ : Inquiry  
 BOT : End of Transmission  
 NAK : Negative Acknowledge  
 STX : Start of Text  
 ETX : End of Text

Fig. 5 Example of communication protocol

하여 오류여부를 검증한다. 이와 같이 통신을 수행해 나가는 순서에 대한 정의라든가 전송되는 정보를 구축하는 방법 등이 프로토콜이라 불리는 것으로 기기마다 다른 형태로 제공된다.

Fig. 5는 이들 통신 방법 중 몇 가지 예를 보여주는 것으로 전송체계(transmission system)는 통신을 수행해 나가는 순서를 표시하고, 데이터 형태(data format)는 한번에 전송되는 정보의 구축방법 등을 보여 준다.

#### 4-2 통신 프로그램

현재 개발된 통신 소프트웨어로는 프로토콜 변환기의 경우 FANUC Controller를 탑재한 수직형 머시닝 센터(MCH-10)<sup>(17)</sup>와 역시 FANUC Controller를 탑재한 Arc-mate Robot(R-G2),<sup>(18)</sup> 그리고 디스페처 대상 기기로 Cincinnati Milacron에서 제작한 5축 머시닝 센터(A950)<sup>(19)</sup>가 개발되어 있다.

프로토콜 변환기의 대상이 되는 기기들의 프로토콜 형태는 Fig. 5와 같이 여러가지가 있으며 몇 가지의 단계를 거쳐 통신이 이루어지게 된다. Fig. 5에 표시한 이들 방법중 형태 1(type 1)은 전송하려는 데이터의 앞뒤에 데이터의 처음과 끝을 나타내는 특수 문자를 첨

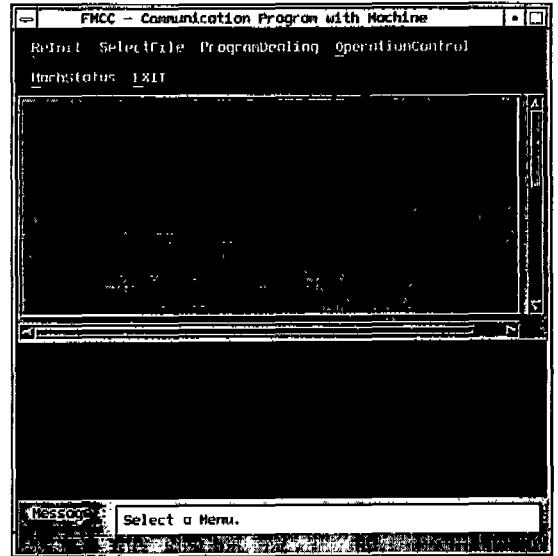


Fig. 6 Initial window of the communication main program

가하여 이들을 전송하는 방법이다. 현재 개발된 MCH-10 머시닝 센터가 바로 이와 같은 방식이다. 형태 2(type 2)는 데이터를 보낼 때 보내는 데이터의 크기를 데이터의 앞부분에 명시하여 데이터를 보내는 방법이며 R-G2 로봇트가 이에 해당한다.

디스페처의 대상이 되는 장비들은 자동화 레벨이 매우 낮으며, 통신을 하기 위한 프로토콜 또한 매우 간단하다. 이러한 프로토콜의 대표적인 예로는 간단한 XON/XOFF 문자 흐름 제어 규약이 있으며 보통 컴퓨터와 해당 기기 간의 작업 프로그램(예를 들어 NC part program)의 송수신만이 통신을 통해서 가능하다. 보통 NC/CNC가 이에 해당하였으나 자동화 시스템의 요구에 따라서 이러한 디스페처의 대상이 되는 장비들은 점차 줄어들고 있는 추세이다.

Fig. 6은 작업자가 사용자 인터페이스 태스크를 통하여 직접 기기와의 통신을 수행할 때 나타나는 주 화면으로서 메뉴를 통해 특정기능을 사용할 수 있다. Fig. 7은 작업자가 직접 가공 프로그램을 선택하여 전송하기 위해 가공 프로그램을 선택하는 화면을 나타낸다. Fig. 8은 통신을 시도하고자 하는 기기들의 통신 방식 설정을 위한 화면으로서 기기의 상태, 통신을 위한 직렬 포트명, 통신사양 등의 정보를 변경할 수 있으며 이들 변경된 정보는 Table 2에 표시한 대로 데이터 베이스에 저장되어 운영된다. Fig. 9는 기기에 가공 프

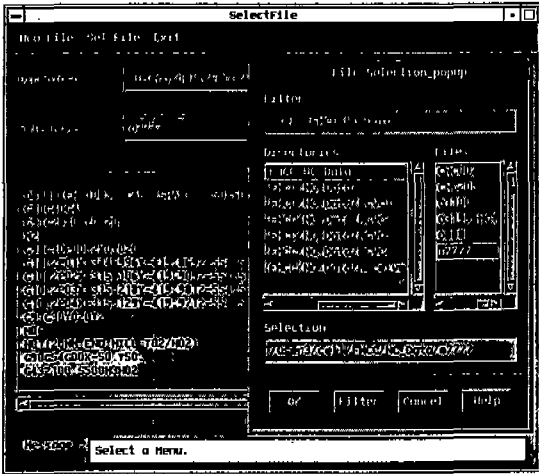


Fig. 7 SelectFile menu display

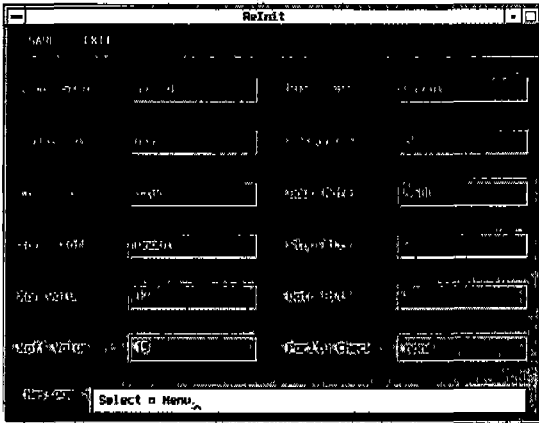


Fig. 8 Communication parameter setting window

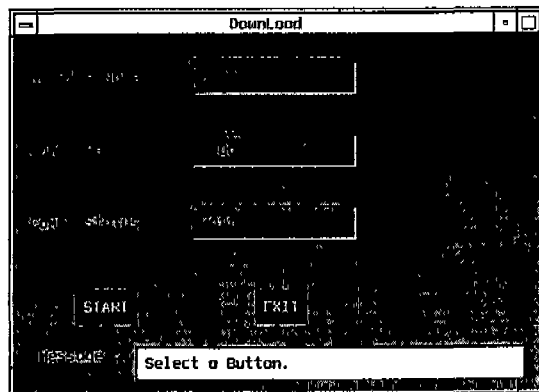


Fig. 9 DownLoad menu display

로그그램을 전송하는 down load 기능 프로그램을 실행한 경우 나타나는 예제 화면이다.

### 4-3 셀 제어기의 통신 기능 실험

셀 제어기와 셀 제어기의 일부분을 차지하는 통신 소프트웨어의 실효성을 입증하기 위해서 본 연구에서는 두 가지의 실험 과정을 설정하였다. 첫 번째 방법으로는 실제의 기기 들과 직접 연결되어 있지 않은 상태에서 PC(personal computer)를 가상의 기기로 가정하여 실험하는 방법이며, 두 번째 방법으로는 첫 번째 방법이 성공한 경우 직접 공장내의 기기 들과 연결하여 실험하는 것이다. 이들 두 가지의 실험에는 각종 기기 들과의 실험 및 내부통신의 실험 모두가 적용되었다.

첫 번째 방법을 위해서 Fig. 10과 같은 실험환경을 구축하였다. 즉, PC에 실험대상이 되는 기기와 같은 동작을 하는 통신 프로그램(emulator program)을 작성하여 설치하고 셀 제어기 프로그램이 설치된 컴퓨터와 RS-232C를 이용하여 연결하였다. PC용 통신 프로그램은 직접 기기와 연결하여 동작성능을 인정받은 것으로 셀제어기와 연결하여 통신을 수행하는데 있어서 기기로 가정하여도 무방한 기능을 제공한다. 이 모의 실험의 장점은 고가의 기계대신 PC를 이용하므로 모의 기기의 수를 많게 하여 개발된 프로그램의 기능 및 동작을 검증하는 한편 셀 제어기의 시뮬레이션에도 이용할 수 있다는 것이다. Fig. 6-9의 화면을 이용하여 실험을 하여본 결과 통신기능은 전송선로간의 간섭 등과 같은 상황이 발생되지 않는 한 적절하게 동작이 수행되는 것을 확인할 수 있었다. 아울러 Fig. 2에 설명한 제어 프로그램 군과의 연계를 통한 자동운전시험도 그 성능을 확인할 수 있었다.

모의실험을 거친 후에는 실제 작업장에서의 실험이 요구되므로 모의 실험의 대상이 되는 기기들과의 직접적인 연결을 시도하였다. 모든 기기가 한 장소에 없는 관계로 우선 MCH-10과 R-G2에 대한 실험은 협동관계(cooperation)가 필요한 현장 혹은 개별작업 수행이 필요한 현장 등을 가정하여 실험하였다. 여기에서 협동관계란 robot로부터 가공소재를 받아서(loading) CNC기계가 가공을 마치면 다시 robot가 가공물을 제거하는(unloading)등과 같은 일련의 협동작업의 의미한다. 가공 프로그램을 전송(down load)하고 작업개시 명령(cycle start)등을 발송하고 기기의 상태를 점검(status check)하는 등의 개별적인 통신을 통한 통

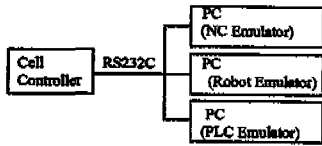


Fig. 10 System construction for simulation

신기능 시험 및 제어 프로그램 군내의 스케줄링 과정을 거친 작업 자동실행기능과의 연계실험등을 하였다. 한편 A950에 대한 실험은 위에서 수행한 과정의 일부분으로 개별적인 통신기능만을 실험하였으나 모의실험의 예에서 별다른 어려운 점을 찾지 못하였으므로 그대로 적용하여도 무방하다고 하겠다. 이러한 일련의 실험을 통해 셀 제어기의 소프트웨어적인 구현가능성을 타진할 수 있었다.

### 5. 결 론

본 연구에서는 자동 생산시스템인 유연생산 시스템을 구축하기 위해 셀 제어기와 작업장에 있는 기기와의 통신에 대해 연구하였다. 공장 자동화 시스템의 기본이 되는 셀에 종래에 갖고 있던 기능에 작업 일정을 계획할 수 있는 기능 등을 추가하여 보다 능동적인 특성을 지니는 셀에 대한 일반적인 정의를 내리고 그 운용을 위한 데이터 베이스를 정의함으로써 컴퓨터 통합 생산 시스템에서 가장 기본이 되는 셀에 대한 전략을 수립하였다. 이러한 전략 하에서 셀 제어기 내에서 공장내의 각종 기기와의 통신을 담당하는 몇 가지의 통신 소프트웨어를 개발하였다. 개발된 통신 소프트웨어는 셀에서 대두되는 통신의 어려움을 극복하기 위해 우수하고 유연성 있는 설계를 바탕으로 하고 있다. 또한 통신 소프트웨어와 셀 제어기를 유효 적절한 방법을 통해서 통합하였고, 간단한 모니터링 전략을 수립하여 하나의 셀 제어기를 이루어 내었다. 이러한 소프트웨어를 바탕으로 셀의 자동 실행을 위한 실험을 행함으로써 자동화 시스템에서의 자동화 구현의 한 방법을 제시하였다.

현재 개발되어진 셀 제어기는 직접 생산 현장에 적용하기에는 아직 많은 문제점을 안고 있다. 그 이유는 통신시 직렬 포트를 사용하기 때문에 작업장 현장에서 발생하는 각종 소음 등으로 통신에 장애를 일으키는 경우가 종종 발생하게되는데 이들을 극복하기 위한 통신 오류 방지 및 오류 발생 후 정상상태로의 환원기법등에 대해 많은 실험을 행해야 할 것이다. 아울러 모니터링

정보의 추출을 위한 보다 다각적인 방법의 모색 또한 필요하다.

### 참고문헌

1. Rembold, U. and B. Nnaji, "The Role of Manufacturing Models for the Information Technology of the Factory of 1990s", *Journal of Design and Manufacturing*, Vol.1, 1991.
2. Arne Bliberg and Leo Alting, "A Flexible and Integrated Control Concept", *Annals of the CIRP*, Vol.39/1/1990, pp.463-466, 1990.
3. Neil A. Duffie, "Synthesis of the Heterarchical Manufacturing Systems", *Computers in Industry*, Vol.14, pp.167-174, 1990.
4. Shimon Y. Nof, "Collaborative Coordination Control (CCC) of Distributed Multi-machine Manufacturing", *Annals of the CIRP*, Vol.41/1/1992, pp.441-445, 1992.
5. FANUC Company, "FD-Mate에 관하여", 12ed, 1989.
6. McNamara J.E., "Local Area Networks an Introduction to the Technology", Digital Press, Burlington, Massachusetts, 1985.
7. Stacy A. H., *The MAP Book: An Introduction to Industrial Networking*, Industrial Networking Incorporated, Santa Clara, California, 1987.
8. 이춘식 외 12, "DNC 시스템 개발", 상공부 1차년도 중간보고서, 개발사업주관기관 한국기계연구원, 1992.
9. 박경진, 도성희, 이재명, 정병수, 강무진, "소프트웨어 개발 관점에서 본 셀 컨트롤러의 설계", 한국정밀공학회 추계학술대회, 1992.
10. 허신, 임준홍, 박경진, "공장자동화를 위한 Cell Controller의 운영기법 및 S/W 개발에 관한 연구", 상공부 중간보고서, 개발사업주관기관 한양대학교, 1992.
11. G.J. Park, S.H. Do, S. Her, Y.S. Moon and J. Lim "Design of a Cell Controller

- from the Software Development Viewpoint", Proceeding of the International Federation of Automatic Control, Australia, 1993.
12. 이재명, 도성희, 박경진, 강무진, "자동화 공정내의 셀 콘트롤러 작동 소프트웨어 개발", 한국정밀공학회지, Vol.12, No.5, 1994,
  13. 김선호, 이춘식, "PLC를 이용한 경제성 있는 실시간 가공 Cell 감시/제어 시스템", 한국정밀공학회 1993 추계학술대회, 1993. 11
  14. Maurice J. Bach, "The Design of The UNIX Operating System", Prentice Hall Inc., 1986.
  15. 홍릉 과학 출판사 편집부역, "UNIX™ 시스템 프로그래밍", 홍릉 과학 출판사, 1991.
  16. Janet I. Egan and Thomas J. Teixeira, "Writing a UNIX™ Device Driver", John Wiley & Sons, Inc., 1988.
  17. FANUC Company, "FANUC PMC-MODEL N 프로그래밍 설명서", 1991.
  18. FANUC Company, "R-G2 Controller", 1991.
  19. Cincinnati Milacron, "Acramatic 950 Serial Data Interface", 1989.