

INTERBUSs를 이용한 모델 플랜트의 구동 소프트웨어의 개발*

조영호¹ · 정병호²

¹전북대학교 산업공학과 인턴연구원/ ²전북대학교 산업공학과, 의견교환저자

Development of an Operating Software for a Model Plant using INTERBUSs

Young- Ho Jo¹ · Byung- Ho Jeong²

This study develops an operating software for a model plant. The model plant consists of an AS/RS system, two machining centers, an assembly line, and supplementary material handling equipments. Devices of each component are connected with the IBS RT24 DIO 16/16-T I/O module. Each I/O module communicates digital signals with the INTERBUS controller board via SUB-D 9 Connector cable. This study is a previous stage for developing an educational CIM software. Petri Nets is used for modelling the storage/retrieval of the AS/RS system, the flow of workpieces and the assembly line for parts. The operating software is coded with Microsoft Visual C++ 5.0 and Interbus Library which is a software driver for the controller board. The operating software can be run on a MS Windows 95. Microsoft Access is used for the implementation of databases for BOM, AS/RS, and parts.

1. 서론

최근 기업의 생산 환경은 소비자의 개성 증시 현상과 제품 수명(Life cycle)의 감소에 따라 다품종 소량 생산 체제로 급속히 변모해 가고 있으며, 이로 인해서 생산의 효율성과 유연성을 갖는 생산 시스템 구축에 대한 필요성이 증대되고 있다(Artman, 1988; Templemeier and Kuhn, 1993). 따라서 흐름 생산의 효율성과 주문 생산의 유연성을 동시에 추구하는 유연 생산 시스템(FMS)과 컴퓨터 통합 생산(CIM) 시스템에 관심이 집중되고 있다. 유연 생산 시스템은 다수의 전용/범용의 장비, 자동 창고, 자동 물류 장치들로 구성되어 있고, 각 공정들을 전용 컴퓨터로 제어하여 생산의 효율성과 유연성을 얻을 수 있는 생산 시스템이다(Groover, 1988; Templemeier and Kuhn, 1993).

이러한 시스템에서 생산의 효율성과 유연성을 얻기 위해서는 시스템 내에 있는 다수의 장비, 자동 물류 장치, 자동 창고들

제어하기 위한 운용 소프트웨어와 이들의 통합을 필요로 한다. 1980년대 초 생산 자동화 환경에 적합한 표준화된 통신망으로 생산자동화 프로토콜(MAP)이 개발되어지면서, 그 후 실시간 통신과 경제적인 통신망의 필요성이 제기 되면서 3계층 구조의 필드버스가 등장하였다. 필드버스에 관한 표준화가 계속해서 이루어졌으며, 생산현장의 수요에 따라 필드버스의 활용이 급속하게 늘어나고 있다.

Jones & McLean(Jones and McLean, 1986)은 생산시스템의 구조 및 프레임 워크에 대한 연구로, 5단계로 이루어진 자동 생산 시스템의 일반적인 계층적 구조를 제시하였는데, 제시한 계층 구조를 살펴보면 공장 레벨(Facility level), 작업장 레벨(Shop level), 셀 레벨, 워크스테이션 레벨, 장비 레벨(Equipment level)등으로 분류하였다. 공장 레벨은 생산능력 관리, 비용 산출, 재고 회계, 주문 관리, 전체 노동시간 추적 등을 수행하기 위한 정보 관리 모듈과 장기 생산계획, 생산 원료 수요량, 초과 생산치를 결정하기 위한 생산관리 모듈로 구성되었다. 작업장 레벨은 직무관

* 본 연구는 전북대학교 공업기술연구소의 지원에 의해 이루어졌음.

리자(Task manager), 자원관리자(Resource manager)로 구분되는 두 개의 주요 컴포넌트로 구성되어 있다. 셀 레벨에서는 각 부품의 작업 순서(Sequencing)와 자재, 물품 운반 관리(Material handling) 등을 수행한다. 워크스테이션 레벨은 장비들의 그룹으로 구성되어 있으며, 이들의 명령을 주고받으며, 조정 및 통제하는 레벨로 장비의 운영에 관련된 작업 순서(Sequencing)를 결정해 준다. 장비 레벨은 로봇이나 NC 머신, 계측기, 운반 장치, 저장 및 출고 장치 등의 컨트롤러로서 이들 장비에게 명령을 내려 작업을 수행하는 레벨이다. 따라서, 장비 레벨에서는 상위 레벨에서 받은 작업 명령 순서를 이행하고, 각 하드웨어의 정보를 모니터링해서 상위 레벨로 넘겨주는 기능을 갖추어야 한다. Smith & Joshi(Smith and Joshi, 1995)은 자동 생산 환경을 위한 Shop floor controller를 개발하기 위한 객체 지향 접근방법 및 클래스 등을 제시하였다. 제어 구조는 일반화된 3단계의 기본적인 레벨로 분류하였다. 작업장 레벨에서는 시스템의 인터페이스와 중앙 데이터 베이스의 제어와 관리를 담당하며, 워크스테이션 레벨은 프로세싱, 저장고, 물류 운반 서비스 등의 클래스로 구성되어 있으며, 장비 레벨은 각 기계의 컨트롤러를 정의 및 제어하도록 클래스가 구성되어 있다.

박창권(박창권 등, 1993)등은 서울대학교 부설 자동화 시스템 공동 연구소내의 FMS/CLM center를 위한 응용 소프트웨어를 구축하였다. 개발된 소프트웨어는 주 생산 계획, 자재 소요 계획, 재고 관리, 구매 관리, 재무관리, 영업 수주 관리 등의 기능을 수행하고, 세 가지 모듈로 나누어진다. 생산 관리 모듈에서는 단위 주기 생산 계획과 작업 일정 전개 등이 있으며, 공장 제어 분야 모듈은 플랜트 동작 제어와 가공, 조립, 검사셀 운영 제어, 무인 반송차 및 자동장고 제어용, 공장 관제 및 작업 실적 보고용 소프트웨어가 있다. 제조 데이터 베이스 모듈은 BOM, 표준 공정 루틴, 부품 정보, 치공구 정보 등 제품 관련 데이터베이스

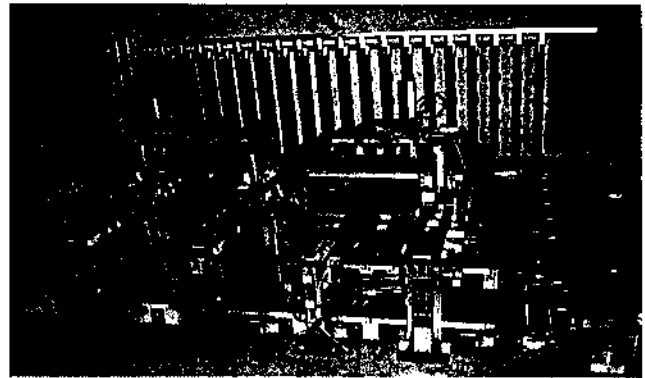


그림 1. 모델 플랜트.

등이 구축되어 있다. 이명재 & 정병수(이명재, 1992; 정병수, 1992)는 자동화 공정 내에 있는 각종 산업용 장비를 소프트웨어적으로 통합, 제어하기 위한 통신 소프트웨어와 작업 일정 계획을 수립하는 스케줄러 프로그램과 수립된 일정 계획에 따라 작업들을 자동적으로 실행, 감시하는 작업 자동 실행 프로그램(Automatic cell operation program)을 개발하였다. 이 연구에서는 종래의 셀의 기능에다 작업 일정 계획, 자동 처리, 장비의 준비 등의 기능을 추가하여 보다 능동적인 특성을 갖는 셀을 제시하였다. 관련된 태스크는 몇 가지 주요 부분으로 나뉘어진다. 우선, 컨트롤 프로그램으로 구성 요소들은 사용자 인터페이스, 컨피규레이션 태스크(Configuration task), 스케줄 태스크, 작업 자동 실행 태스크(ACO task)등으로 이루어져 있으며, 셀의 운영과 제어, 작업 영역내의 각종 기기들의 환경설정, 작업 스케줄링, 작업의 일정 처리 및 감시를 담당한다. 통신 프로그램 컨트롤 태스크 부분으로 계획된 작업 일정에 따라 처리하게 될 태스크간의 통신과 작업 자동 실행 태스크간의 정보의 흐름을 제어한다. 마지막으로 각종 기기들의 컨트롤을 담당하

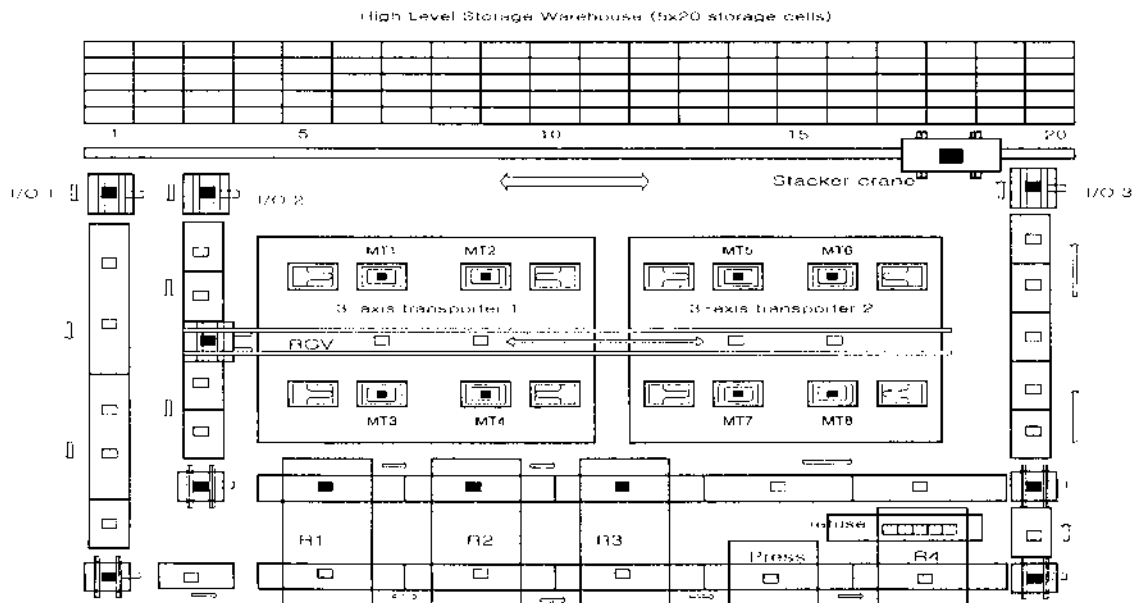


그림 2. 모델 플랜트의 배치도.

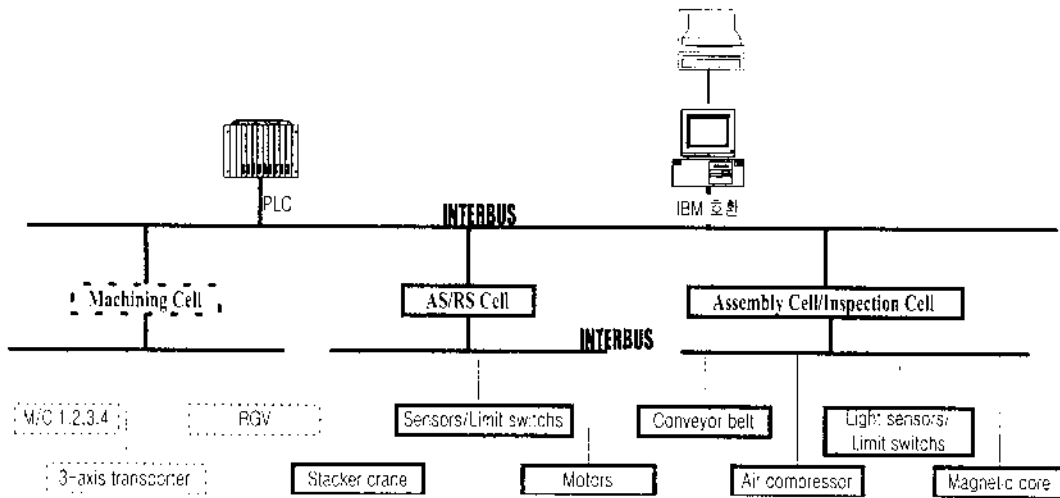


그림 3. 모형공장의 네트워크 구성도.

는 부분으로 프로토콜 변환기 태스크와 디스패처 태스크등이 있고, 상위 레벨과의 인터페이스 태스크가 있어 상부구조인 통합 셀과의 정보의 흐름을 제어한다.

본 논문은 독일의 Fischertechnik사로부터 도입한 모델 플랜트 <그림 1>을 이용한 교육용 CIM 시스템 구축의 전 단계로서 작업장내의 물류이동을 위한 운용 소프트웨어를 개발하기 위한 것이다. 이를 위해서 우선 자동창고와 조립 라인을 중심으로 모델 플랜트를 구동시키는 소프트웨어를 개발하였다. 이 시스템은 상위 레벨의 제어시스템을 개발하는 기초가 될 것이다. 시스템은 INTERBUS controller board의 소프트웨어 드라이버인 INTERBUS Library와 MS Visual C++를 이용하여 구현하였다.

2. 모델 플랜트와 INTERBUS

2.1 모델 플랜트의 개요

본 연구에 사용되어진 시스템은 컴퓨터 통합 생산에 대한 교육 및 연구용으로 도입한 시스템으로 독일의 Fischertechnik사에

서 제작하였으며, 모델 플랜트는 자동 창고와 가공 및 조립 라인으로 구성되어 있으며, 각 모듈은 단품들의 결합에 의해 이루어져 있다. <그림 1>, <그림 2>에서 모델 플랜트의 전체 사진과 배치도를 보여주고 있다. 시스템의 주요 모듈로는 AS/RS, Conveyor belt, RGV(Rail guided vehicle), Robot을 포함한 Assembly/Inspection cell, Machining cell, 3-axis transporter 등이며, 자동 창고는 100개의 Workpiece 보관 셀과 입출고를 담당하는 1대의 Stacker crane을 가지고 있다.

이 시스템은 Box, Ball, Disk와 Cover등 4 가지 종류의 Workpiece를 사용하도록 설계되었으며, Box에 Ball과 Disk를 삽입하고 마지막으로 Cover를 씌움으로써 완제품 모형을 생산하게 된다. 이러한 4 종류의 기본 Workpiece들의 조합에 의해 다양한 종류의 완제품 모형을 정의할 수 있다. Workpiece의 흐름을 살펴보면, 자동창고에서 Stacker crane을 통해 출고가 되고, RGV, 3-axis portal transporter, Conveyor belt 등을 통해 이동이 이루어지고, 각 부품의 경로에 따라 조립 라인이나 가공 라인을 이동하면서 하나의 완제품이 된다. 완제품은 다시 Conveyor belt를 통해 자동 창고로 이송되어 진다. 본 시스템의 Machining cell에서는 실제 가공은 이루어지지 않고 loading된 상태에서 지정된 가

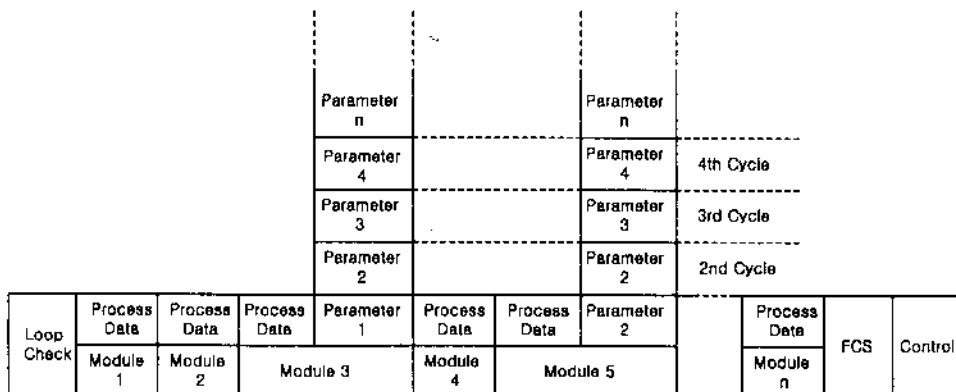


그림 4. INTERBUS transfer protocol.

공 특성에 따라 해당 시간만큼 시간을 지연시킨 후 방출하는 것으로 가공을 대신한다

본 시스템의 네트워크 구조는 <그림 3>과 같으며, 그림에서 실선으로 본 논문의 개발 영역을 표현하고 있다. 각 Cell내의 광 센서, 근접 센서, 리미트 스위치, 에어 컴프레서 개폐 스위치 등은 I/O 모듈에 연결이 되어 있고, 다시 I/O 모듈들을 하나의 시리얼 케이블로 연결하여 그 케이블을 통해 PC나 PLC로부터 오는 모든 입출력 신호를 주고 받도록 구성되어 있다. 모델 플랜트는 모두 11개의 I/O 모듈을 가지고 있으며, I/O 모듈들은 각 16개의 입·출력점을 가지고 있다.

2.2 INTERBUS 컨트롤러와 통신

본 시스템은 INTERBUS를 이용하여 센서, 리미트 스위치, 모터, 에어 컴프레서 개폐 스위치등이 시리얼 네트워크로 연결되어 있으며, 각 디바이스에서 발생되는 입력 신호와 PC에서 보내는 출력 신호들은 INTERBUS-PC Interface board를 통하여 디바이스와 PC간의 디지털 통신이 이루어진다. INTERBUS-PC Interface board에는 Assembly language나 High level language를 이용하여 제어할 수 있도록 Firmware가 포함되어 있다. 본 소프트웨어와 FieldBus와의 통신에 사용되어진 데이터 전송 프로토콜 구조는 <그림 4>와 같이 정의되어 있다(INTERBUS, 1997). 버스 시스템에 장착된 센서(Sensor), 액츄에이터(Actuator), 계측기등의 제어는 프로세스 데이터와 파라미터를 통해 제어가 이루어지고 있으며, 또한 프로세스 데이터를 통해 모니터 할 수 있다. 프로세스 데이터는 각 밸브나 컨택터(Contactor)의 운전 시그널(Signal)이나 스위치 상태 등을 담고 있으며, 드라이버 컨트롤 시스템의 구동 비트를 나타내고 있다. 프로세스 데이터는 네트워크 망을 통해 연속적으로 갱신되어지는 시스템의 정보 즉 I/O Data를 주기적으로 받는다. 따라서, 프로세스 데이터는 터미널 디바이스나 I/O 장비의 주소를 명시하고 있다. 반면 파라미터 정보는 특정 명령 또는 메시지가 있을 경우에 보내는 정보이므로 파라미터 블록 내에 특정 장비의 주소를 갖지 않는다. 하드웨어와 소프트웨어 간의 인터페이스는 Multi-Port memory를 통해 정보의 교환이 이루어지고, 디바이스 드라이버 Function을 통해서만 액세스 할 수 있다. 디바이스 드라이버 인터페이스에

표 1. DDI functions

| 함 수 | 기 능 |
|------------------------|-------------------|
| DDI_DevOpenNode | 노드에 데이터 채널을 할당함 |
| DDI_DevCloseNode | 노드에 데이터 채널을 제거함 |
| DDI_MXI_SndMessage | MPM에 메시지를 보냄 |
| DDI_MXI_RcvMessage | MPM에서 메시지를 받아들음 |
| DDI_DTI_ReadData | MPM에서 데이터를 읽어옴 |
| DDI_DTI_WriteData | MPM에 데이터를 보냄 |
| DDI_SetMsgNotification | 메시지 통보 모드를 활성화함 |
| DDI_ClrMsgNotification | 메시지 통보 모드를 비활성화시킴 |

표 2. Hardware control functions

| 함 수 | 기 능 |
|--------------------|---------------------------------------|
| GetSysFailRegister | 시스템 에러 레지스터에 들어있는 내용을 받아들음 |
| EnableWatchDog | 감시모드를 활성화시킴 |
| TriggerWatchDog | 감시모드를 기동시킴 |
| ClearWatchDog | 감시모드를 제거함 |
| GetWatchDogState | 감시모드의 정보를 받아들음 |
| GetIBSDiagnostic | INTERBUS Master board의 상태 평가 정보를 받아들음 |

서는 데이터 채널 관리를 위한 기능, 메일박스 인터페이스(MXI), 데이터 인터페이스(DTI), 검사 기능(Diagnostic function) 등을 제공한다. 데이터 채널 관리 기능에서는 컨트롤러 보드와 MPM 접근기(MPM accessor), 인터페이스, 디바이스명 파라미터(DevName)등을 이용하여 노드 핸들(Node handle)을 열어 주어 데이터 전송 준비를 한다. 메일박스 인터페이스는 MPM accessor간의 메시지를 전송하기 위한 프로토콜-기반 인터페이스이며, 데이터 인터페이스는 열려진 노드를 통해 핸드셰이킹(Handshaking)없이 데이터를 주고 받는 인터페이스이다.

MS Windows 95를 위한 드라이버는 가상 디바이스 드라이버(VxD)를 가지는 Dynamic link library(DLL)를 제공하고 있다. <표 1>과 <표 2>에서 VIBSSCD.VXD에 포함되어 있는 DDI function과 Control function을 일부 보여주고 있다. DDI function을 이용하여 MPM accessor에 접근할 수 있으며, 이를 통하여 장비들을 컨트롤 할 수 있다.

3. 소프트웨어 설계 및 모델링

3.1 모델 플랜트의 모델링

자동화된 생산시스템의 모델링과 설계시에는 각 작업공정

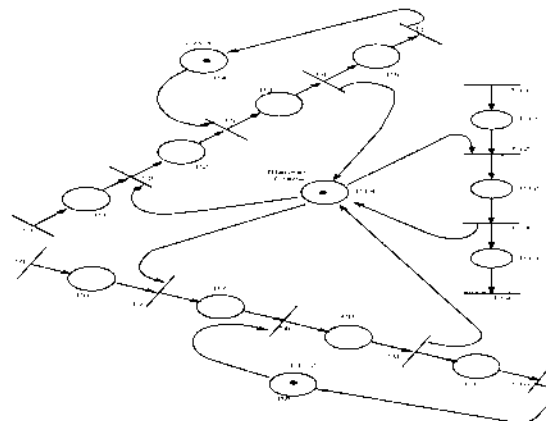


그림 5. 자동 창고 시스템에 대한 페트리 넷 모델.

표 3. 자동 창고 모델의 트랜지션과 플레이스에 대한 설명

| 트랜지션 | 설 명 | 플레이스 | 설 명 |
|------|--------------------|------|---------------------------|
| T1 | T1 출고 | P1 | 부품 I 출고 명령 |
| T2 | 출고 시작 | P2 | stacker crane으로 부품 I 을 운반 |
| T3 | I/O 1에 부품 I 내려 놓기 | P3 | 부품 I 의 I/O 1 진입 완료 |
| T4 | 팔렛 놓기 | P4 | I/O 1 이용 가능 |
| T5 | conveyor 셀로 T1 이동 | P5 | 운반 완료 |
| T6 | T2 출고 | P6 | 부품 II 출고 명령 |
| T7 | 출고 시작 | P7 | stacker crane으로 부품 II을 운반 |
| T8 | I/O 2에 부품 II 내려 놓기 | P8 | I/O 2 이용 가능 |
| T9 | 팔렛 놓기 | P9 | 부품 II의 I/O 2 진입 완료 |
| T10 | conveyor 셀로 T2 이동 | P10 | 운반 완료 |
| T11 | 입고 | P11 | 입고 명령 |
| T12 | 입고 시작 | P12 | 완성품을 창고로 이동 |
| T13 | 창고로 이동 시작 | P13 | 입고 완료 |
| T14 | 창고에 완성품 저장 | P14 | stacker crane 이용 가능 |

의 선행관계와 농정간의 내부 관계, 사용할 수 있는 자원의 작업 능력, 자원의 할당 등이 고려되어야 한다. 이러한 문제에 대해 모델링 도구로써 페트리 넷트는 생산시스템의 동시성과 비동기성 등을 시각적으로 쉽게 표현해 주고, 충돌회피와 공유자원의 할당, 교착상태 방지 등의 제어논리를 제공해 준다(James, 1981; Son, et al., 1991; Thomas, et al., 1989).

본 절에서는 모델 플랜트내의 조립 라인과 자동 창고 셀 내에서 이루어지는 각 Workpiece의 흐름과 로봇의 가용 여부, 자원 할당 문제 등에 대해 페트리 넷트를 이용하여 모델을 제시한다. 자동 창고와 조립 라인의 페트리 넷트 모델에서 사용되어진 트랜지션들은 조립 작업이나 해당 명령에 대한 출력 작업, 입력 플레이스의 상태 변화에 따른 동작을 표현하였고, 플레이스들은 각 트랜지션들의 제어와 리소스의 가용 여부, 용량 등을 표현하였다.

3.1.1 자동 창고의 기본구조 및 모델링

모델 플랜트내의 자동 창고 시스템은 가로 방향으로 20개의 위치 지점이 있고 각 위치에 5단의 셀이 있다. 즉, 자동 창고 시스템 내에는 Workpiece 100개를 보관할 수 있는 보관 셀이 존재한다. 자동창고 시스템 내에서의 모든 운반은 하나의 Stacker crane에 의해 이루어지며, 자동 창고에서 출고된 Workpiece들은 해당 부품의 특성에 따라 두 개의 출고 지점으로 이동하게 된다. 조립 라인 셀을 거쳐 1차 가공이 끝난 가공품이나 완성품은 한 곳의 입고 지점으로 오게 된다. 초기의 자동 창고내의 Workpiece는 각 부품별로 20개씩 저장되어 있고, 각 부품들은 일차 가공이 되어 바로 조립이 가능하다고 가정한다. 출고 명령을 받게 되면 출고 작업 트랜지션 T2의 작업이 시작되는데, 출고 작업 트랜지션 T2 점화는 Stacker crane의 플레이스의 토큰과 출고 작업 지시의 조건이 만족되어야만 점화가 이루어진다. 트랜지션 T3의 작업은 I/O 1 station인 플레이스 P4에 작업 공간이 비어 있어야 점화될 수 있다. 트랜지션 T4가 점화됨으로써 점유하였던 Stacker crane을 놓아주고, I/O 1 station의 토큰은 트

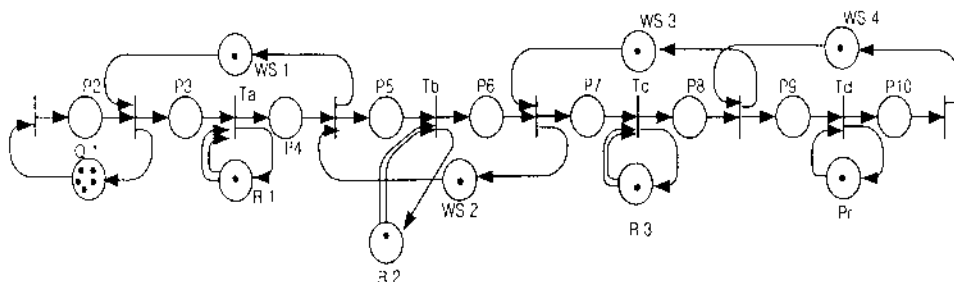


그림 6. 부품 B 형의 작업 흐름에 대한 페트리 넷트 모델.

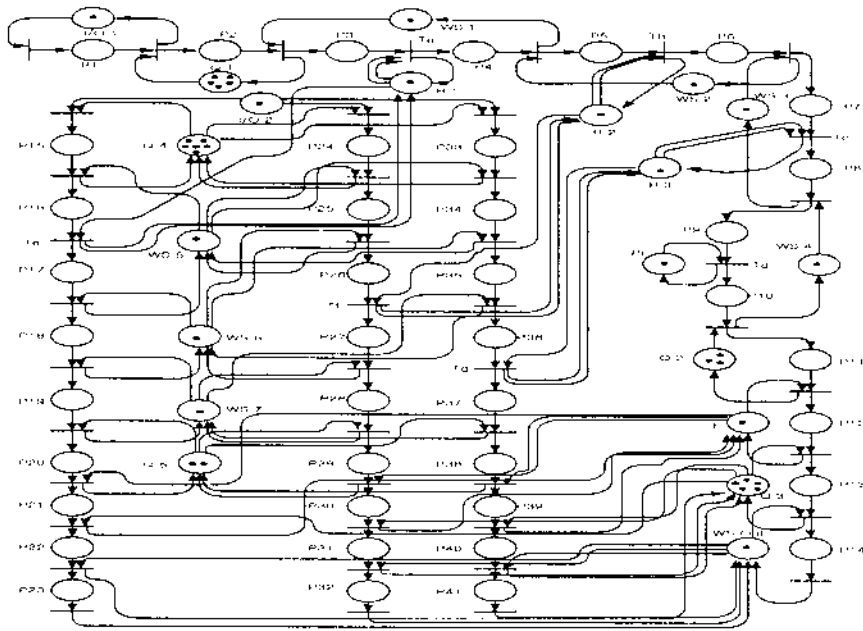


그림 7. 조립 공정의 페트리 넷 모델.

랜지션 5의 점화로 돌려 받게 된다. 자동 창고 내에서는 각 부품 타입에 대한 두가지 출고 작업과 완성품 입고 및 팔렛 등에 대한 입고 작업 등 세가지 작업이 동시에 진행된다. 따라서 자동 창고 모델에서는 이 세가지 작업들을 표현한 모델들로 구성되어야 한다. <그림 5>에서는 각 부품 타입의 출고와 작업이 종료된 완성품과 빈 팔렛의 입고 등 세 가지 작업을 반영한 자동 창고 시스템 페트리 넷 모델을 보여주고 있다. <표 3>은 자동 창고 페트리 넷에서 사용된 플레이스들과 트랜지션들에 대한 설명을 보여주고 있다.

3.1.2 조립 라인셀의 기본 구조 및 모델링

조립 라인 셀은 공압으로 작동하는 3개의 이동 로봇, 1개의 프레스 로봇, 그리고 컨베이어 벨트와 2개의 턴 테이블로 구성되어 있다. 부품의 조립 작업은 자동 창고의 Stacker crane으로부터 부품을 공급받음과 동시에 시작된다. 즉, 3-3절에서 소개하게 될 BOM 정보에서 부품 타입 B형은 I/O 스테이션 1 지점으로 공급되어지며, 부품 타입 A 형은 I/O 스테이션 2 지점으로 공급되어 각각 해당되는 Workstation으로 이동된 후 로봇의 작업을 기다리게 된다. 부품 타입 B형의 조립 작업과 이동 작업에 대한 페트리 넷 모델은<그림 6>과 같다.

부품 타입 B형의 이동 작업은 컨베이어 벨트를 통해 행해지며, 조립 작업은 세 개의 이동 로봇으로부터 부품을 제공받으며 마지막 프레스 작업으로 조립 작업은 종료한다.

I/O 스테이션 2로 놓여진 부품 타입 B형은 컨베이어 벨트를 통해 WS1 작업영역으로 이동하여 조립 작업 1의 수행을 기다리게 된다. 조립 작업 1은 로봇 1이 유희하고 제공받아야할 부품이 있어야 수행하게 되는데, 이를 제어하기 위한 토큰 생성은 다른 부품 라인에서 타입 A형의 부품을 로봇 1이 집고 WS1

지역으로 이동하는 작업을 마쳐야 이루어지게 된다. 즉, <그림 6>에서 조립 작업 1인 트랜지션 Ta는 R1 플레이스에 2개의 토큰이 있어야 점화가 되는데, 이는 다른 부품 라인의 부품 제공 작업인 트랜지션 Te에서 먼저 점화가 이루어져야 2개의 토큰을 돌려 받기 때문에 작업의 선행관계를 제어할 수 있다. 조립 작업 1은 R1 플레이스에 로봇 가용 토큰 1개를 돌려주어 부품 라인에서 로봇 1을 사용할 수 있게 해 줌으로써 작업을 마치게 된다. 그 이후 다른 조립 작업인 트랜지션 Tb와 Tc는 조립 작업 1과 동일하게 각 해당 로봇으로부터 토큰 2개를 제공받아야 작업이 이루어지며, 프레스 작업은 프레스 가용 여부인 Pr 플레이스의 토큰에 따라 작업의 진행 여부가 결정되어 진다. 조립 작업과 프레스 작업이 완료되면 컨베이어 벨트를 통해 WS Our 지역으로 이동하여 자동 창고로부터 Stacker crane의 입고 작업을 기다리게 된다. 조립 라인에서는 부품 타입 A와 B형이 각 해당 라인을 통해 이동하며 부품에 따라 다른 작업들이 이루어진다. 따라서 각 부품의 흐름에 대해 서로 다른 페트리 넷 모델이 설계되어지며 이들 모델들을 하나의 페트리 넷 모델로 조합을 하였다.

<그림 7>에서 각 부품에 대한 모델들을 통합하여 하나의 페트리 넷 모델로 구성한 조립 라인 전체의 페트리 넷 모델을 볼 수 있으며, 페트리 넷에 사용된 각각의 플레이스에 대한 설명은 <표 4>에 나타나 있다.

3.2 기능 모듈의 구성

워크스테이션 레벨이나 장비 레벨은 기존의 CAD/CAM, FMS에 대한 연구 결과로 비교적 실용화가 많이 이루어져 있으므로 본 연구에서는 제외됐다. 본 연구는 작업장내의 불차 흐름을

표 4. 조립 공정의 페트리 넷 모델의 플레이스에 대한 설명

| 플레이스 | 설 명 | 플레이스 | 설 명 |
|--------|--------------------------|------|-------------|
| I/O i | I/O i 이용 가능 (i=1,2) | P 18 | WS 6에 부품 진입 |
| WS i | WS i 이용 가능 (i=1,2,...,7) | P 19 | WS 7에 부품 진입 |
| Q i | Q i 이용 가능 (i=1,...,5) | P 20 | Q5에 부품 진입 |
| R i | 로봇 i 이용 가능 (i=1,2,3) | P 21 | 방향 전환 완료 |
| Pr | 프레스 이용 가능 | P 22 | Q3에 부품 진입 |
| T | 턴 테이블 이용 가능 | P 23 | 입고 작업 준비 완료 |
| WS Out | WS Out 이용 가능 | P 24 | Q4에 부품 진입 |
| P 1 | Part B 준비 완료 | P 25 | WS 4에 부품 진입 |
| P 2 | Q1에 부품 진입 | P 26 | WS 5에 부품 진입 |
| P 3 | 작업 1 준비 완료 | P 27 | 디스크 작업 완료 |
| P 4 | 작업 1 완료 | P 28 | WS 7에 부품 진입 |
| P 5 | 작업 2 준비 완료 | P 29 | Q5에 부품 진입 |
| P 6 | 작업 2 완료 | P 30 | 방향 전환 완료 |
| P 7 | 작업 3 준비 완료 | P 31 | Q3에 부품 진입 |
| P 8 | 작업 3 완료 | P 32 | 입고 작업 준비 완료 |
| P 9 | 프레스 작업 준비 완료 | P 33 | Q4에 부품 진입 |
| P 10 | 프레스 작업 완료 | P 34 | WS 5에 부품 진입 |
| P 11 | Q2에 부품 진입 | P 35 | WS 6에 부품 진입 |
| P 12 | 방향 전환 완료 | P 36 | WS 7에 부품 진입 |
| P 13 | Q3에 부품 진입 | P 37 | 커버 작업 완료 |
| P 14 | 입고 작업 준비 완료 | P 38 | Q5에 부품 진입 |
| P 15 | Q4에 부품 진입 | P 39 | 방향 전환 완료 |
| P 16 | WS 5에 부품 진입 | P 40 | Q3에 부품 진입 |
| P 17 | 볼 작업 완료 | P 41 | 입고 작업 준비 완료 |

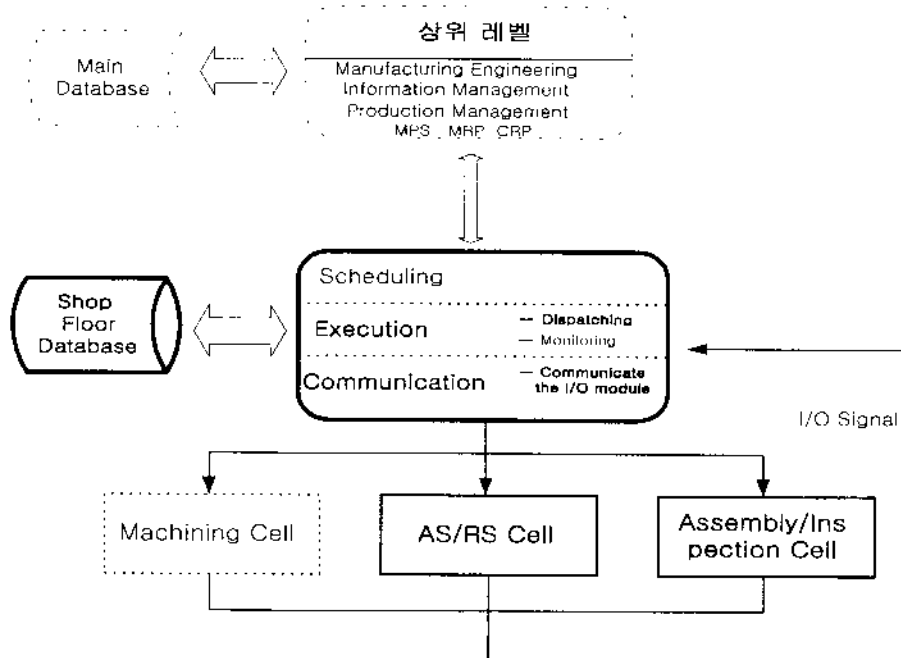


그림 8. 시스템의 기능 모듈 구성도.

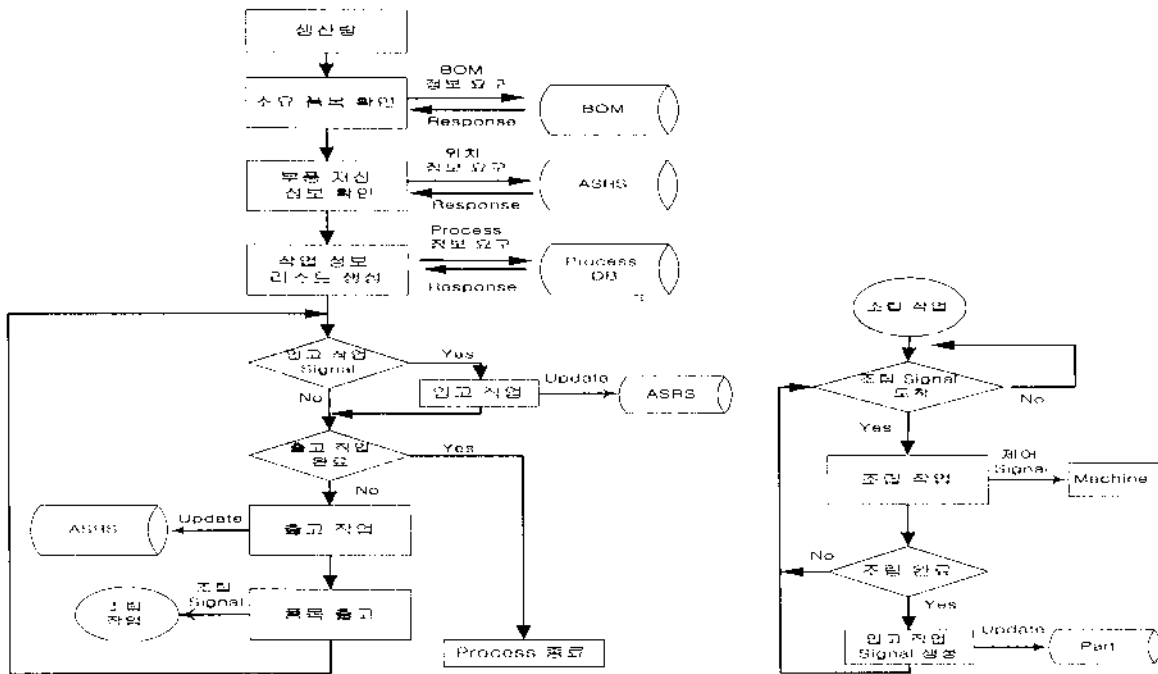


그림 9. 데이터 흐름도.

제어하기 위한 구동 소프트웨어 개발에 주안점을 두며 이는 공장 레벨을 포함한 운용 소프트웨어 개발의 기초가 된다. 작업장내의 물자 흐름을 제어하기 위해서는 Scheduling, Execution, Communication 등 3가지의 기능들이 구현되어야 한다. 본 연구에서 개발된 모듈의 기능을 <그림 8>에서 요약하여 보여주고 있다. 이번 논문에서는 작업장 레벨에서의 Execution, Communication 기능을 구현하는데 중점을 둔다.

Execution Module : 부품의 투입 순서에 따라 각 part를 AS/RS로부터 인출하는 작업으로 part flow에 대한 제어를 담당한다. 또한, 각 작업장에 장착된 센서를 이용해 shop내의 part의 흐름을 감시하며, 센서로부터 출력된 I/O 신호를 입력받는다. 즉, 작동 스위치에 신호를 보내주어 각 장치들이 작업을 수행하도록 한다.

Communication Module : Execution Module과 모델 플랜트내의 디바이스와의 통신을 담당하는 모듈로 INTERBUS-PC Interface board를 이용하여 제어한다. 제어 신호는 인터버스 프로토콜의 형식에 맞추어 프로세스 데이터와 파라미터 정보를 디지털 신호 형태로 컨트롤러 보드에 전송해 준다.

Shop Floor DB: 프로세스 정보, 작업장의 상황 정보, 작업의 현재 상태, 작업물에 대한 정보, 재고 정보를 포함하는 자동창고의 Cell occupancy 정보 등을 담고 있으며, 이들 정보 중 일부는 빈번하게 참조 및 갱신이 이루어져야 되고 정보 단위 및 수명도 짧은 형태를 취하고 있다. 따라서 이들은 실시간형 데이터베이스로 수사로 데이터 갱신이 이루어지도록 구축한다.

본 시스템에서 개발된 부분은 <그림 8>의 기능 모듈 구성도에서 실선과 굵은체로 표현된 부분이며, 조립라인과 자동창고의 구동을 위해 생산품목과 생산량을 입력받아 소요 부품의 중

류, 수량, 자동창고내의 저장위치, 조립장비들의 정보, 현 시스템의 상황 등을 참조하여 각 작업을 제어하고 조정한다. 스케줄링은 차후에 개발될 모듈에 포함될 것이며 본 논문에서는 입력받는 것으로 가정하였다. 작업상 데이터베이스에는 부품 정보와 프로세스 정보, 자동 창고에 관한 정보를 담고 있다. 실행 모듈에서는 전기적 신호를 통신 모듈에 주어 부품을 가공 공정이나 조립 라인으로 이동시키고, 완성품을 자동 창고로 이동시켜주는 역할을 담당하고 있다. <그림 9>에서 각 모듈에서 이루어지고 있는 데이터 흐름에 대한 내용을 보여주고 있으며 크게 3개의 프로세스로 구분된다. 첫 번째 프로세스는 작업 리스트를 생성하는 부분으로 생산량을 입력받으면 해당 작업에 필요한 정보를 수집하여 작업 정보 리스트를 만들어 준다. 두 번째 프로세스는 작업 정보가 구성이 되면, 해당 작업물에 대해 출고 및 입고를 수행하는 부분이다. 세 번째 프로세스는 조립라인 제어 부분으로 부품 출고가 이루어지면 해당 부품을 이동하여 조립 작업을 하도록 지시한다. 입고출고 프로세스와 조립라인 제어 프로세스는 서로 독립적으로 작업이 이루어지도록 쓰레드 프로그램을 통하여 구성하였으며, 각 각의 작업 지시여부는 시그널을 주고 받으며 메시지를 전달한다.

3.3 데이터 베이스 구성

작업장 구동 소프트웨어에서는 다음과 같은 기능을 수행할 수 있도록 데이터 베이스를 구성하였다. 우선 상위 레벨로부터 스케줄 정보를 받아 해당되는 작업물에 대한 정보를 구성할 수 있는 테이블이 있어야 한다. 현 시스템에서 스케줄 정보는 시

표 5. 자동 창고 정보

| Field | Type | Length | Descriptions |
|---------|------|--------|--------------|
| Cell_X | N | 2 | 자동창고의 X 좌표 |
| Cell_Y | N | 2 | 자동창고의 Y 좌표 |
| Part_ID | C | 8 | 부품 번호 |

표 6. Part DB (부품 관련 정보)

| Field | Type | Length | Descriptions |
|-----------|------|--------|--------------|
| Part_ID | C | 8 | 부품 번호 |
| INV | N | 2 | 현 재고량 |
| Part_Type | C | 12 | 메모 필드 |

표 7. BOM 정보

| Field | Type | Length | Descriptions |
|---------|------|--------|--------------|
| Assy_ID | C | 8 | 완성품번 |
| Part_ID | C | 8 | 부품 번호 |
| N1 | N | 4 | 수 량 |
| Part ID | C | 8 | 부품 번호 |
| N2 | N | 4 | 수 량 |
| ... | C | 8 | 부품 번호 |
| ... | N | 4 | 수 량 |
| Part ID | C | 8 | 부품 번호 |
| Nn | N | 4 | 수 량 |

스텝 관리 및 스케줄링을 할 수 있는 상위의 운영 소프트웨어가 개발 중이므로 스케줄 관련 정보는 주어진다 가정한다. 작업물 리스트 테이블에 저장될 정보는 작업장 Operation

controller가 작업을 원활히 수행할 수 있도록 부품의 위치 정보 및 프로세스 정보 등을 가지고 있다. Job status report부분에서는 장비 레벨에서 들어오는 부품의 가공 정보 등의 피드백 정보를 사용자가 모니터하기 위해 필요한 정보를 제공할 수 있도록 구성한다. Inventory report부분은 부품에 대한 재고 정보와 원재료 정보를 제공하는 기능을 담당한다. 작업장 데이터 베이스에는 부품과 각 기계, RGV, Conveyor belt 등의 장비 정보, 작업 진행 정보, Resource에 대한 정보를 가지고 있어 작업장 Controller에 해당 정보를 제공한다. <표 5>와 <표 6>에 자동 창고 데이터 정보와 부품 관련 정보에 필요한 내용을 보여주고 있으며, <표 7>과 <표 8>에서 BOM 정보와 부품의 프로세스 내용을 보여 준다.

4. 운용 소프트웨어의 개발

4.1 개발환경

본 소프트웨어에서는 자동 창고셀, 자동 조립셀, 자동 가공셀 내의 작업이 서로 독립적으로 수행되어야 한다. 따라서 다중 작업(multi-tasking)이 가능한 오피레이팅 시스템인 OSR2 (Windows 95)환경에서 개발하였다. 또한 병렬 처리를 위해 각 작업을 Multi process로 분리하였고, 각 Process에서도 독립적으로 작업을 하도록 하기 위해 Win 32 SDK에서 제공하는 쓰레드 클래스를 이용하여 프로그램 모듈을 구성하였다.

사용자 인터페이스 모듈은 OSR2상에서 이식성이 높고 Win 32 SDK를 이용할 수 있는 Microsoft Visual C++을 이용하여 구축하였다. Fieldbus와의 인터페이스는 Phoenix Contact사에서 제공하는 INTERBUS 인터페이스 Library와 Firmware를 사용하였다.

표 8. 완성품 가공 정보

| 완성품번 | 작업 소요 시간 | | | | | | | |
|-------|----------|---|-------|---|----------|----|------------|----|
| | 조립 (sec) | | | | 프레스(sec) | | 총소요시간(sec) | |
| P1111 | A4003 | 5 | A4001 | 5 | A1001 | 5 | 3 | 18 |
| P1122 | A4003 | 5 | A4002 | 7 | A2001 | 10 | 3 | 25 |
| P2213 | A4004 | 7 | A4001 | 5 | A3001 | 15 | 3 | 30 |
| P2114 | A4003 | 5 | A4001 | 5 | A2001 | 10 | 3 | 23 |
| P3123 | A4004 | 7 | A4002 | 7 | A3001 | 15 | 3 | 32 |

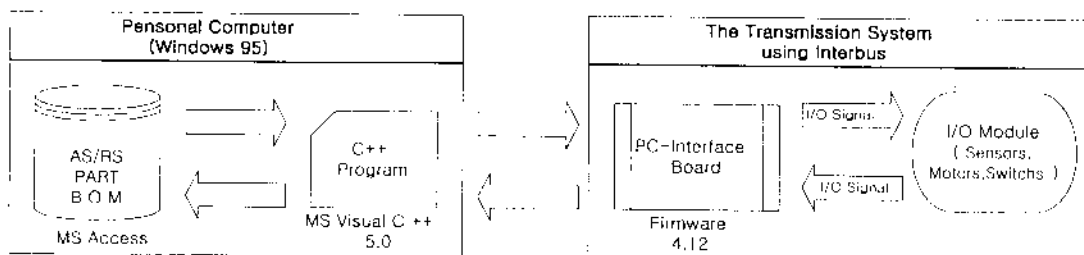


그림 10. 개발 환경.

표 9. PROCESSDATA STRUCTURE

```

Structure PROCESSDATA
{
    WORD deviceNumber;
    WORD addressMPM;
    BYTE bit PositionMPM;
    BYTE dataDirection;
    BYTE Datatype;
    WORD bytePositionInterbus;
    BYTE bitPosition
    BYTE length;
};
    
```

Shop floor controller에서 사용될 각 정보를 저장할 데이터베이스는 Microsoft Access에서 제공되어지는 데이터베이스를 사용하였다. 본 연구에서 개발된 시스템의 환경을 <그림 10>에서 볼 수 있다.

4.2 소프트웨어 소개 및 클래스

본 논문에서 사용된 클래스는 크게 4종류로 분류할 수 있다. DB 관련 클래스, INTERBUS 제어 클래스, PC-Interface board에 내장된 Firmware 4.12를 제어하기 위한 Communication 클래스 그리고 마지막으로 전체 프레임을 표현하기 위한 클래스 등으로 나누어진다. 데이터베이스를 관리하기 위한 AS/RS 관련 DB에는 ODBC를 실제로 컨트롤하는 CDatabase 클래스와 CRecordset 클래스가 포함되어 있으며, AS/RS 관련 DB 클래스로는 ASRS_DB 클래스, Part_DB 클래스, BOM 클래스가 있고, DB 제어 클래스로 CimSet 클래스가 있다. INTERBUS 제어 관련 클래스에서는 INTERBUS 관련 정보나 실제 작업을 지시하는 클래스들로 데이터베이스로부터 얻은 조립 작업 정보를 가지고 각 센서와 액츄에이터에 어떤 신호를 보내줄 것인가 결정해주는 역할을 맡고 있다. 또한 처음 INTERBUS와 신호를 교환하기 위한 노드 핸들(Node handle)을 얻는 작업과 I/O 모듈의 초기화 작업도 담당한다. Communication 관련 클래스에서는 데이터 처리 작업과 데이터 전송 작업 등으로 실제 디바이스에 보내지게 될 데이터 구조 정의와 Firmware 정의의 클래스이다. PROCESSDATA STRUCTURE는 IBS RT 24 DIO 16/16-T 모듈에서 인식할 수 있는 정보에 대한 구조체를 표현하고 있다. 따라서 PROCESSDATA 구조체에는 디바이스 번호, 데이터 방향, 데이터 형식, 송수신할 데이터 Length등을 포함하고 있다. <표 9>에 PROCESSDATA에 대한 구조체를 볼 수 있다.

소프트웨어 구조를 분류해 보면 크게 기본 정보 입력과 작업 지시 및 수행 역할의 두 가지 모듈로 구성되어 있다. 기본 정보 모듈은 자동 참고 데이터 베이스 참조/수정 모듈, 부품 정보 모

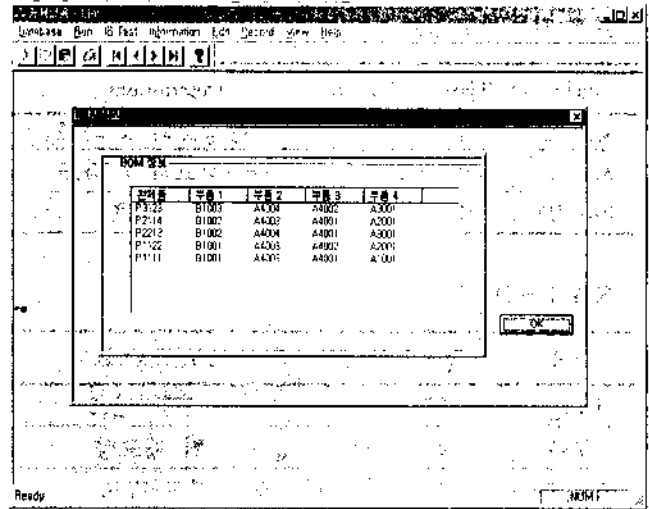


그림 11. BOM 정보 조회 화면.

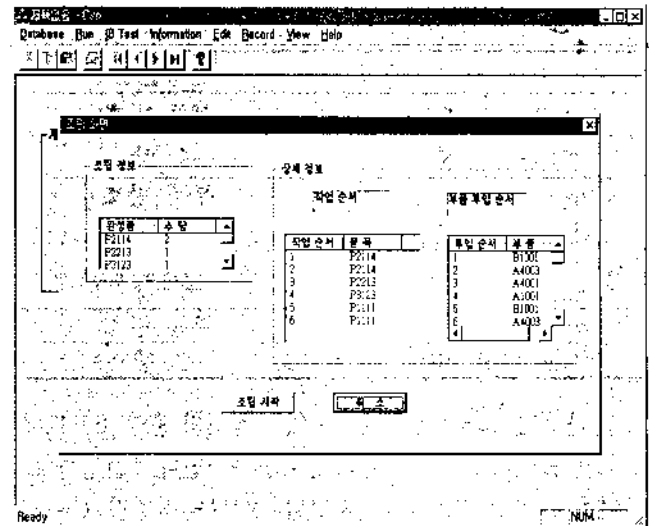


그림 12. 조립 화면.

듈의 입력/수정 모듈 그리고 BOM 정보 참조 모듈로 나뉘어진다. 기본 정보의 자동 참고 수정 모듈은 자동 참고 내의 각 셀에 대한 정보를 참조하거나 수정 할 수 있는 입력 화면을 제공한다. 부품 정보 모듈에서는 각 부품 변경 정보와 부품 정보에 대해 수정/참조할 수 있도록 구성하였다. 각 완성품 별로 소요되는 부품의 정보는 기본 정보 중 BOM 정보 모듈에서 조회할 수 있도록 하였다. <그림 11>에서 BOM 정보에 대한 조회 화면을 보여 주고 있다. 수행 모듈에서는 모뎀 플랜트를 구동하기 위한 초기화 모듈과 조립 작업을 진행할 프로세싱 모듈로 구성되어 있다. 초기화 모듈에서는 INTERBUS 컨트롤러를 제어할 수 있는 노드 핸들을 얻어서 데이터 전송을 준비해주는 모듈이다.

프로세싱 모듈은 실제 조립 작업을 맡고 있는 모듈로 데이터 베이스로부터 각 부품의 정보와 저장되어 있는 자동 참고 셀 위치를 파악하여 해당 품목을 출고해서 조립 라인으로 보내는 자동 참고 로직과 출고된 부품을 해당 워크스테이션으로 이동

후 조립작업을 수행하는 조립 모듈 로직으로 되어 있다. <그림 12>에 프로세싱 모듈에 대한 처리 화면을 보여 주고 있다. 프로세싱 처리 모듈에서는 작업할 완성품 리스트와 각 완성품 당 필요한 품목들의 투입 순서를 표시하여 실제 자동창고에서 빠져나갈 품목을 보여준다. 조립시작 버튼을 클릭함과 동시에 상세 정보에 나와 있는 각 부품의 투입 순서대로 모델 플랜트에 대한 제어가 이루어진다.

5. 결 론

본 논문에서 모델 플랜트에 대한 모델링은 페트리 네트를 이용하여 표현하였으며, 이를 통해 시스템 전반에 대한 정보를 제시하였다. 본문에서 제시된 페트리 네트는 시스템내의 작업에 대한 흐름, 자원의 할당, 용량 문제, Workpiece의 흐름 등을 시각적으로 보여 주고 있으므로 추후에 PLC를 통한 제어 코드가 가공 공정 셀을 운용하기 위한 소프트웨어 개발에도 활용할 수 있다. 추후에 가공 공정 셀의 추가와 스케줄러가 포함된 상위 레벨의 운용 소프트웨어로서의 확장을 위해 시스템의 각 기능들에 대해 모듈화하였다. 본 시스템에서는 모델 플랜트내의 자동 창고 셀과 조립 라인 셀에 대한 구동 소프트웨어 개발에 중점을 둔 것으로 가공 셀에 대한 로직은 지원되지 않고 있다. 본 논문에서 개발된 시스템은 INTERBUS를 통해 각 디바이스들이 연결된 모델 플랜트의 구동 소프트웨어로 차후에 개발될 가공라인 운영 소프트웨어와 함께 상위 단계의 Shop floor 운용 소프트웨어 개발의 기반이 될 것이다. 즉, 모델 플랜트를 대상

으로 한 교육용 CIM S/W의 상위 레벨 소프트웨어 개발의 기초를 제공하고 있다.

참고문헌

박찬권, 박진우, 강석호(1993), ASRI-FMS/CIM을 위한 운용 소프트웨어의 구축, *IE Interfaces*, 6(2).

이명재(1992), Software development for cell controller operation and scheduling in a CIM system, *한양대학교 대학원 석사 학위 논문*.

정병수(1992), 자동화시스템내 셀 컨트롤러의 통신 소프트웨어 개발 및 실험, *한양대학교 대학원 석사 학위 논문*.

Artman, H. W.(1988), Computer integrated manufacturing, *Computer Integrated Manufacturing Current status and Challenges*, 3-36.

Groover, M. P.(1988), Automation, Production systems, and Computer Integrated Manufacturing, *Prentice-Hall INC*.

INTERBUS(1997), User manual Generation 4 Controller Boards as of Firmware 4.12, *Phoenix Contact*.

Jones, A. and McLean, C. R.(1986), A proposed hierarchical control model for automated manufacturing systems, *Journal of Manufacturing Systems*, 5, 15-25.

James L. P.(1981), Petri net theory and the modeling of systems, .

Smith, J. S. and Joshi, S. B.(1995), A shop floor controller class for computer-integrated manufacturing, *International Journal of Computer integrated manufacturing*, 8(5), 327-339.

Son, S. K., Choi, B. K., Park, T. U. and Lee, K. H.(1991), Resolution of multiple conflicts in the Petri-net model of FAS, *International Journal of Computer Integrated Manufacturing*, 4(4), 253-262.

Thomas O., Bourcher, Mohsen A., Jafari, and Glenn, A. M.(1989), Petri net control of an automated manufacturing Cell, *Computers Ind. Engng.*, 17(1-4), 459-463.

Tempelmeier, H. and Kubo, H.(1993), Flexible Manufacturing Systems, *John Wiley & Sons INC*.



조영호
 1993년 중앙대 통계학 석사
 1997년 충남대 통계학과 박사과정
 현재: 한국전자통신연구원 시스템분석팀
 선임 연구원 재직중
 관심 분야: 경제성공학, 시스템분석, 신경망
 이론, 데이터마이닝



정병호
 1993년 중앙대 통계학 석사
 1997년 충남대 통계학과 박사과정
 현재: 한국전자통신연구원 시스템분석팀
 선임 연구원 재직중
 관심 분야: 경제성공학, 시스템분석, 신경망
 이론, 데이터마이닝