'96 대한전기학회 하계학술대회 논문집 (1996. 7. 22~24)

데이터베이스를 이용한 통합제어시스템 用 공정 데이터 관리 시스템

신경봉 , *허우정, *김문철, *김용석, **김왕길, ***황진식
* 삼성종합기술원 산업전자연구실
** 삼성중공업 기전연구소, *** 삼성중공업 플랜트기술팀

Data management system for integrated control system

Kyeong-Bong Shin , *Woo-Jung Huh, *Moon-Cheol Kim, *Eung-Seok Kim
**Wang-Kil Kim, ***Jin-Sik Hwang
* System Control Laboratory, Samsung Advanced Institute of Technology.
** Research Institute of Machinary & ElectroTechnology, Samsung Heavy Industries Co.
*** Plant Team, Samsung Heavy Industries Co.

Abstracts

There has been much interested on the issues of designing and implementing the data acquisition and display system. This paper describes how to acquire and manage the data to be generated from sensor sources in a manufacturing process. The functionality of this data management system is composed of data acquisition, database management, processing and display of the available data. Also, this system has a adaptability to be carried throughout configuration management.

1. 서론

통합 제어 시스템은 조작자의 운영 및 조작 인터페이스를 위한 MMI(Man Machine Interface) 시스템과 각종 제어 알고리듬의 수행 및 입출력 데이터를 관리하기 위한 중앙제어기, 그리고 원거리 입출력 장치(Remote I/O), 직접 입출력 장치(Direct I/O) 등의 각종 입출력 장치로 구성된다. 통합 제어 시스템에는 기능상 크게 제어 프로그래밍 툴에서 담당하는 제어 알고리듬의 작성 및 수행 기능과 공정 데이터 관리 기능으로 나눌 수 있는데 공정 데이터 관리 시스템의 기능은 통합 제어 시스템 구성에 있어서 중앙제어기 상에서 각종 입출력 장치에서 발생하고 있는 공정 데이터를 획득하고 효율적으로 관리하는 기능과 중앙제어기 상에 존재하는 공정 데이터를 MMI 시스템에 전송하는 기능, 그리고 MMI 시스템 상에서 표준 데이터베이스에 저장하고 제어 프로그래밍 툴(SoftLogic Designer), 감시툴(Monitoring Tool), 보고서 작성기 (Report Writor) 등과 연계하는 기능으로 크게 고려 할 수 있다.

본 논문은 통합 제어 시스템(Integrated Control System) 중에서 공정 데이터를 획득하고 데이터베이스를 이용하여 공정 데이터를 유지 관리하는 시스템에 관하여 서술하고 있으며, 공정 데이터 관리 시스템의 특징은 MMI 시스템에서의 공정 데이터 관리시 표준 데이터베이스 시스템을 활용하므로써 융통성(Flexibility)이 높을 뿐만 아니라 그래픽 사용자 인터페이스인 마이크로소프트 사의 Windows '95를 이용하기 때문에 사용자에게 편리성을 주고 있다.

2. 시스템 구성

통합 제어 시스템의 서브 시스템(Subsystem)인 공정 데이터 관리 시스템의 시스템 구성은 그림 1에서 보여 주고 있다. MMI 시스템은 PC(Personal Computer)와 완전히 호환되는 산업용 컴퓨터를 기반으로 하고 있으며 중앙제어기는 실시간 운영체제인 (주)마이크로텍 사의 VRTX 0.S가 탑재되고 VME 버스 기반 시스템으로 구축되었다. 중앙제어기에서는 MMI 시스템과 이더넷으로 통신하고 있는데 MMI 시스템에서 수행되고 있는 제어 프로그래밍 툴을 통해 입력된 제어 알고리듬이 이더넷 통신으로 중앙제어기에 다운로드(Download)되고 이를 수행하고 있으며 직접 입출력 장치나 원거리 입출력 장치의 입출력 데이터, 즉 공정 데이터를 유지 관리하고 이를 MMI 시스템으로 전송하는 기능을 담당한다. PLC 및 원거리 입출력 장치들은 표준 통신 프로토콜인 필드버스(FieldBus)로 연결되어 있으며 이들 장치에서 발생하고 있는 공정 데이터 들을 중앙제어기로 전송하여 주 메모리의 일정 영역에 로드하는 기능이 존재한다.

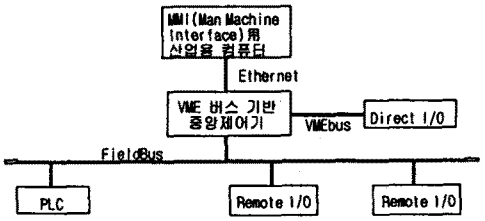

그림 1. 전체 하드웨어 시스템 구성

그림 2에서는 공정 데이터 관리 시스템의 시스템 소프트웨어 구성을 보여 주고 있다. 제어 프로그래밍 툴은 제어 대상에 적합한 제어 프로그램을 작성하고 제어 알고리듬 수행시 입출력 데이터의 동작 상황을 간단하게 감시하는 기능을 가지고 있으며 감시툴은 현 시점에서 발생하는 공정 데이터를 감시하거나 이전에 발생했던 이력 데이터를 이용하여 동작했던 공정 상황을 재현하는데 사용되며 보고서 편집기는 공정 데이터의 통계 처리 결과 등에 대한 보고서를 작성하는데 사용되는데 보고서 작성에 필요한 데이터를 데이터베이스 관리부를 통해 추출한다.

구성툴(Configuration Tool)은 시스템 전체 구성을 나타내는 구성 데이터와 입출력 접점의 특성을 나타내는 특성 데이터를 설치자가 입력하기 위한 툴이다. 데이터베이스 관리부는 통신서버를 통해서 전송되는 공정 데이터를 주기적으로 데이터베이스에 저장하고 (주) 마이크로소프트 사의 Windows'95의 OLE(Object Linking and Embedding)를 이용하여 공정 데이터를 감시툴, 보고서 편집기 등에 넘겨 주는 역할을 담당한다. 데이터베이스는 데이터베이스 관리 시스템을 뜻하는 것으로 본 개발에서는 (주) 삼성전자에서 개발한 EASYBASE 데이터베이스를 이용하여 구현할 수 있다. 데이터베이스에의 저장 및 추출은 ODBC (Open DataBase Connectivity)라는 표준 인터페이스 함수를 이용한다.

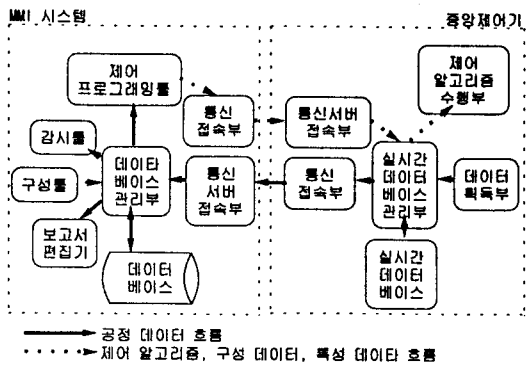

그림 2. 전체 소프트웨어 구성

MMI 시스템에서 통신 서버 접속부는 중앙제어기의 통신 접속부와 연계하여 공정 데이터를 주기적으로 전송 받고 데이터베이스 관리부를 통해 데이터베이스에 공정 데이터를 저장한다. MMI 시스템에서 통신 접속부는 중앙제어기에 제어 알고리듬, 구성 데이터, 특성 데이터 등을 중앙제어기의 통신 서버 접속부를 통해 다운로드하고 다운로드된 제어 알고리듬은 제어 알고리듬 수행부에서 수행된다.

구성 데이터(configuration data)는 시스템 구성에 관하여 설치자가 입력한 데이터로써 직접 입출력 장치(Direct I/O)에 대한 장치 갯수, 타입(디지탈/아날로그, 입력/출력), 입출력 접점수, 메모리 매핑 주소 등과 원거리 입출력장치(Remote I/O)에 대한 노드 수, 각 노드에 설치된 보드(board) 수 및 각 보드의 타입(디지탈/아날로그, 입력/출력)과 입출력 접점수 등이 해당되며 특성 데이터(attributedata)는 입출력 접점의 특성을 나타내는데 입출력 접점의 이름, 접점 변수의 타입(디지탈/아날로그, 직접/원거리, BOOL/INT/REAL), 초기값, 간단한 설명, 논리적인 주소, 데이터 변환에 필요한 최대값, 최소값, 공학적인 단위 등을 유지하는 데이터이다. 논리적인 주소는 직접 입출력 장치인 경우 메모리 매핑 주소로 표현되며, 원거리 입출력 장치인 경우는 노드 번호, 보드 번호, 채널 번호로 표현 된다. 중앙제어기에서의 실시간 데이터베이스 관리부는 데이터 획득부에서 획득한 공정 데이터와 제어 알고리듬, 구성 데이터, 특성 데이터 등을 유지하고 있는 실시간 데이터베이스를 관리하며 주기적으로 공정 데이터를 MMI 시스템에 전송하는 역할을 담당한다. 실시간 데이터베이스는 할당받은 주 메모리 영역을 뜻하는데 필드버스에 연결된 입출력 장치 들과 중앙제어기 상의 주 메모리 간에 맵핑 영역(mapping area)과 제어 알고리듬 영역, 특성 및 구성 데이터 영역등이 이에 속한다. 실시간 데이터베이스는 주 메모리(DRAM) 상에 구현되어 있어 중앙제어기가 오 동작으로 인해 재부팅(booting)시 실시간데이터베이스에 관리되고 있는 공정 데이터는 없어지게 된다. 이를 해결하기 위하여 VMEbus에 공유메모리(SRAM) 보드를 채택하여 실시간 데이터베이스의 영역을 주기적으로 공유메모리에 백업한다.

3. 공정 데이터의 획득

공정 데이터의 발생원은 필드버스(FieldBus) 통신 프로토콜로 연결된 각종 입출력 장치 들이다. 필드버스는 공장 자동화에 많이 사용되는 산업 표준 통신 프로토콜로써 ISO/OSI 7 계층 모델 중에 물리층 (1계층)과 데이터링크층(2계층)과 어플리케이션층 (7계층)을 지원하고 있으며 멀티 마스터 멀티 슬레이브 프로토콜(multi-master multi-slave protocol)이다. 또한 MAP 통신 프로토콜과 호환성을 갖고 있으며 노드 간의 거리는 1200미터까지 가능하다.

공정 데이터를 획득하기 이전에 구성 데이터를 이용하여 각 필드버스에 연결된 입출력 장치 들과 중앙제어기 상의 주메모리 간에 맵핑 영역(mapping area)을 할당 받는다. 공정 데이터의 획득은 일련의 필드버스 프로그래밍 인터페이스를 이용하여 획득하게 되는데 획득된 공정 데이터는 특성 데이터를 이용하여 잡음(Noise)을 제거하는 평활화(Smoothing) 및 단위변환(conversion) 단계를 거치게 되고 맵핑 정보에 따라 주메모리의 맵핑 영역에 저장된다.

4. MMI 시스템과 중앙제어기 간의 통신

산업용 PC를 기반으로 하는 MMI 시스템과 VME 버스 시스템을 기반으로 하는 중앙제어기 간의 통신은 현재 산업 표준으로 널리 사용되고 있는 이더넷(Ethernet)을 이용한다. 통신 프로토콜은 TCP/IP를 사용하는데 이는 ISO/OSI 7 계층 네트워크 구조 중에서 물리층, 데이터 링크층, 네트워크 층, 트랜스포트(Transport)층을 지원하는 프로토콜로써 프로그래밍을 위한 대표적인 인터페이스로 UNIX System V 계열의 TLI와 BSD UNIX 계열의 소켓(Socket)이 존재한다. 중앙제어기 상에는 실시간 운영체제로 마이크로텍 사의 VRTXsa O.S를 채택하여 탑재하였으며 VRTXsa O.S 상위에서 제공되는 TCP/IP 통신 프로토콜 패키지로 SNX (STREAMS-based Networking)를 지원해 주고 있어 이를 사용하였다. 그림 2에서 보여주는 바와 같이 MMI 시스템과 중앙제어기 간에 두개의 소켓 포트(Port)를 이용하여 양방향 통신을 하고 있다. 즉, MMI 시스템에 통신 서버가 존재하여 항시 중앙제어기로부터 전송되는 데이터를 받고 또한 중앙제어기에 통신 서버가 존재하여 MMI 시스템으로부터 전송되는 데이터를 받는다.

MMI 시스템에서 중앙제어기로 전송되는 데이터의 종류로는 전체적인 시스템 구성에 관한 구성 데이터와 각종 입출력 데이터의 특성에 관한 특성 데이터와 제어 프로그래밍 툴에서 중앙제어기로 전송되는 알고리듬과 명령(download, go, stop, pause) 등이며 중앙제어기에서 MMI 시스템으로 전송되는 데이터는 시스템의 감시(Monitoring)를 위해 각종 입출력 접점에서 발생되는 데이터와 경보 데이터 등이 있다.

MMI 시스템과 중앙제어기 간의 통신 기능 구현은 VRTXsa O.S의 SNX에서 제공해 주고 있는 BSD 소켓 함수(Socket Library)를 이용하여 구현하면 되는데 보다 편리하게 통신 시스템을 구현하기 위하여 소켓 함수 상위에 새로운 API (Application Programming Interface)를 정의하였다. 또한 공정 데이터 관리 시스템에서는 MMI 시스템과 중앙제어기 상호간에 전송되는 데이터의 종류와 크기가 다양하기 때문에 실제 데이터를 전송하기 전에 미리 헤더정보를 전송한다. 그림 3은 헤더 구조체를 보여 주고 있는데 헤더에는 유효성 검사를 하기 위한 유효 키(Validation Key)와 데이터의 종류와 통신 유형을 정의하는 타입 항목과 다음에 전송될 실제 데이터의 크기를 나타내는 항목으로 구성되어 있다.

```
// 6 byte_fixed length
typedef struct com_data_head {
    char  vaild_key;        /* 유효키 */
    char  type;             /* 타입   */
    unsigned int data_len;  /* 데이터 크기 */
} COMM_DATA_HEAD, *pCOMM_DATA_HEAD;
```

그림 3. 통신 데이터 헤더 구조체

여기서, 위 구조체의 타입 항목은 전송 데이터의 4가지 통신 유형을 나타내고 있다. 첫째 유형은 통신 검사와 같은 시스템 내부에서 사용하기 위해서 예약해 놓은 유형이며 둘째 유형은 실제 데이터를 필요치 않아 통신 헤더 정보만을 전송하는 경우에 해당하며 셋째 유형은 헤더와 실제 데이터를 전송하는 경우이고 넷째는 MMI 시스템에서 중앙제어기에 필요한

공정 데이터 및 특정 데이터를 전송해 달라는 요청을 하는 경우이다. 중앙제어기에 데이터를 전송하거나 중앙제어기에서 필요한 데이터를 얻으려고 하는 응용프로그램은 적당한 통신 유형을 타입에 채워 헤더 정보를 보내야 한다. 즉, 구성 데이터, 특성 데이터, 제어 알고리듬은 셋째 유형에 해당되며 명령(download, go, stop, pause)과 같은 데이터는 위 구조체의 타입 항목에 정보를 첨가하여 전송하는 둘째 유형에 속한다. 또한 감시툴에서 현재 발생되고 있는 공정 데이터를 실시간으로 얻으려고 할 때는 넷째 유형을 사용한다.

```
/* communication server running in target */
STATUS target_comm_server()
{
   COMM_DATA_HEAD       data_info;

   /* 통신 연결 설정한다. */
   if(HT_com_init_in_target() == ERROR) HT_com_init_fail();
      while(TRUE){
            /* 헤더 정보를 받는다 */
            if(HT_receive_head(&data_info) == ERROR) .....
            /* 유효성 검사한다. */
            if(data_info.valid_key  !=  VALID_KEY)  .....
            /* 통신 유형을 체크, 그에 상응하는 처리를 한다. */
            switch(data_info.type){
            case 유형1: .....
            case 유형2: .....
            case 유형3:
                  /* 메모리를 할당받고 데이터를 얻는다 */
                  if(HT_receive(&data_info, data) == ERROR) ..
                        /* 전송받은 데이터 처리 ...*/
            case 유형4:
                  /* 1. 전송할 데이터의 정보를 얻는다. */
                  /* 2. 전송된 정보에 따라 전송할 데이터를
                        만든다. */
                  /* 3. 만들어진 데이터를 전송한다. */
      } /* while(TRUE) */
} /* STATUS target_comm_server */
```

그림 4. 통신 서버의 구조

그림 4에서 보여 주고 있는 중앙제어기에서의 통신 서버(Communication Server)는 MMI 시스템으로부터 전송되는 데이터를 받아 해석하고 그에 상응하는 처리를 담당하는 태스크로써 먼저 헤더 데이터를 받은 후에 데이터의 유효성을 검사하고 데이터의 4가지 통신 유형에 따라 처리를 하게 되는데 유형3은 미리 전송 받는 데이터의 데이터 영역을 할당받으며 통신 유형4는 MMI 시스템에서 필요로 하는 데이터의 목록을 먼저 전송 받고 이 목록에 따라 데이터를 실시간 데이터 관리부에서 얻은 다음 MMI 시스템으로 전송한다.유형2는 MMI 시스템에서 조작자의 명령을 처리하는 것으로 명령 정보는 그림 3의 구조체에서 type 항목을 체크하여 알 수 있다. 명령 정보에 따라 임의의 태스크를 생성하고 삭제한다.

5. 이력 데이터 관리

본 공정 데이터 관리 시스템에서 EasyBase 데이터베이스에 저장하고 관리하는 데이터의 종류로는 앞에서 언급된 구성 데이터(Configuration data), 특성 데이터(attribute data), 경보데이터, 각종 입출력 접점에서 발생한 공정 데이터 등을 들 수 있으며 응용프로그램에서 자기 고유의 데이터를 저장 할 수 도 있다. 이러한 데이터는 데이터베이스의 논리적인 테이블 형태로 저장되는데 본 공정 데이터 관리 시스템은 시스템 전체 구성을 나타내는 구성(configuration) 데이터 유지를 위한 하나의 테이블과 입출력 접점의 특성을 나타내는 특성(attribute) 데이터를 유지하기 위한 하나의 테이블과 경보 데이터를 유지하기 위한 하나의 테이블과 공정 데이터를 유지하기 위한 입출력 접점 수 만큼의 테이블을 유지하고 있다.

ODBC(Open DataBase Connectivity)는 (주) 마이크로소프트 사에서 제안한 데이터베이스 접근을 위한 표준 프로그래밍 인터페이스로써 함수, 데이터 타입, SQL 문법, 표준 에러코드를 정의하고 있으며, 그림 5는 ODBC 구성도를 보여주고 있다. 드라이버 관리자(Driver Manager)는 프로그램이 요청한 데이터 소스를 연결하는 역할을 하며 드라이버(Driver)는 DLL(Dynamic Linking Library)로써 ODBC 함수를 뜻한다. 데이터소스(Data Source)는 DBMS와 DBMS를 운영하는 운영체제, 그리고 원거리 노드에 연결하기 위한 네트웍크의 집합체를 뜻한다. ODBC 함수를 이용하여 데이터베이스를 접근하는 응용프로그램은 원거리 노드의 데이터베이스에 접근할 수 있으며 소스코드의 수정 없이 다른 데이터 소스에 접근 할 수 있다. 본 공정 데이터 관리 시스템도 ODBC 함수를 이용하여 EasyBase 데이터베이스에 접근하고 있으며 구현하는데 있어서는 ODBC 함수 상위에 편리한 함수를 재 정의하여 사용하였다.

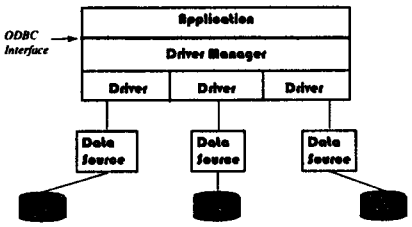

그림 5. ODBC 구성도

공정 데이터 관리 시스템은 입출력 접점에서 발생한 공정데이터를 중앙제어기의 실시간 데이터베이스 관리부에서 주기적으로 MMI 시스템으로 전송하고 MMI 시스템은 통신 서버에서 이를 받아 OLE를 이용하여 감시툴로 데이터를 전송하거나 ODBC 함수를 이용하여 공정 데이터를 EasyBase 데이터베이스에 저장한다. 데이터베이스에서 관리되고 있는 이력 데이터는 보고서편집기에서 보고서로 편집되거나, 자료처리기에서 관리되며 감시툴에서 공정 상황이 재현되기도 한다.

6. 결론

통합 제어 시스템(Integrated Control System)의 서브 시스템인 공정 데이터 관리 시스템은 필드버스에 연결된 원거리 입출력 장치와 VME 버스에 직접 연결된 직접 입출력 장치에서 발생하는 공정 데이터를 획득하고 이를 표준 데이터베이스를 이용하여 관리하며 감시툴, 보고서편집기, 자료처리기, 제어 프로그래밍 툴 등의 응용 프로그램과 연계하여 통합 제어 시스템에서 공정 데이터를 효율적으로 관리하기 위한 시스템으로 본 논문은 이러한 공정 데이터 관리 시스템의 구성과 구현 방법에 대하여 서술 하였다. 제시된 공정 데이터 관리 시스템은 공정 데이터 관리시 표준 데이터베이스 관리 시스템을 활용하므로써 개방성과 융통성이 매우 증가하였고 그래픽 사용자 인터페이스인 Windows '95를 이용하기 때문에 사용자에게 편리성을 주고 있다. 현재의 자동화 분야의 입출력 장치는 매우 다양하며 고유의 통신 사양을 가지고 있어 앞으로 범용성을 위해 다양한 입출력 장치의 드라이버(Driver) 개발 및 구성툴을 보다 편리하게 구현해야 하는 필요성을 느낀다.

참고문헌

[1] 김응석외, *Crane Control System 중간보고서*, 삼성종합기술원, 1996

[2] *VRTX/OS VRTXsa User's Guide*, Microtec Research, 1995

[3] *VRTX/OS SNX User's Guide*, Microtec Research, 1995

[4] *ODBC 2.0 Programmer's Reference and SDK Guide*, Microsoft, 1992

[5] W.Richard Stevens, *UNIX Network Programming*, Prentice hall, 1990