

論文94-31A-10-3

分散 制御 시스템에서 遠隔 制御局의 實時間 通信에 關한 研究

(A Study on Real-Time communication of Remote Station in the Distributed Control System)

金來鎮*, 金珍泰*, 朴仁甲*

(Nae Jin Kim, Jin Tae Kim and In Kap Park)

要約

본 논문에서는 분산 제어 시스템의 설계 제작 및 실시간 통신에 관하여 기술하였다. 시스템 신뢰성을 위하여 본 논문에서 설계한 분산 제어 시스템은 CPU 고장시 recovery 장점이 있는 네트워크 파일 시스템으로 설계하였고, 원격 제어국의 파일 시스템 접근이 용이한 DB 구조 및 공유 메모리 기법을 채택하였으며, 실시간 처리용 VME147 board를 채택하여 네트워크의 지연시간을 단축하였다. 구현된 분산 제어 시스템에서 원격 제어국의 실시간처리에 대한 성능 분석은 아날로그신호를 인터페이스 카드로 받아 CPU에서 -2,500 ~ 10,000까지의 컴퓨터 값으로 데이터를 바꾸는 LOOP 처리 기능과 해당 DO point의 bit를 setting하는 LOGIC 처리 기능을 수행한 후 처리된 데이터 결과를 중앙 제어국으로 실시간 전송하는데 까지 걸리는 총 수행시간을 분석하였다.

Abstract

We discussed the Distributed Control System's design on preface and analyzed time of the real-time communication by using designed system. The DCS proposed in this thesis was implemented to network file system having recovery advantage and shared memory method to access file system of a Remote Station with ease. Also, this system minimized the network delay-time by using the real-time VME147 board.

In implemented DCS, the performance analysis of real-time process of a Remote Station was done to get the total time for real-time communication from a Remote Station to the Central Station after terminating of process.

For the analysis of system performance, we experimented by three steps. Firstly, we measured the processing time of LOOP function that real-time CPU convertes to -2,500 ~ 10,000 values from the input data of the Analog Interface Card. Secondly, we measured the processing time of the LOGIC function and the LOOP function. Lastly, we measured total processing time for communication from a Remote Station to the Control Station.

1. 서론

*正會員, 建國大學校 電子工學科
(Dept. of Elec. Eng., KonKuk Univ.)
接受日字: 1994年 3月 26日

지난 '70년 공업용 분야산업의 플랜트제어용 컴퓨터의 사용은 DDC(direct digital control) 시스템으

로 적용하기 시작했으나 플랜트의 감시 조작은 물론이고, 프로세스 제어까지 집중화함으로써 발생하는 신뢰성 저하 및 운영의 어려움때문에 분산 제어 시스템인 DCS(distributed control system)가 75년 이후 산업용 제어시스템에서 일반화 되었다. DCS는 DDC시스템이라고도 하는데 DDC는 컴퓨터의 중앙 처리장치가 process data의 입력, 연산처리, 출력 등을 관장하는 시스템으로 초기에는 처리속도가 비교적 빠르고 연산처리가 효율적이었으나 점차 부하량이 늘어남으로써 처리속도가 늦어지고 집중화됨으로써 생기는 여러가지 문제가 대두되었다. 그러나, 현재의 제어 시스템은 빠른 속도가 요구되고 입출력의 일괄 처리가 요구되기 때문에 90년대 근거리 통신망(LAN)의 보급, 응용과 자료의 투명성을 제공하면서 FA 분야의 자동화산업으로 중요하게 자리잡을 수 있는 시스템은 분산 제어 시스템이 적합하게 되었고, 90년대의 분산된 컴퓨터 네트워크를 이용하여 통합된 EIC(electric, instrument, control) 분산 제어 시스템의 시기가 도래하였다. ^[11-12]

따라서 시스템이 발달하면 발달할수록 운영자에게는 가용성과 신뢰성을 보장하고 있으며, 운영자에게 있어 네트워크를 이용한 DB의 수정을 쉽게할 수 있는 방향으로 시스템 발달은 하고 있다. 그러나 네트워크 이용시 고속의 LAN으로 통신한다해도 work time이 이루어지는 시점에서는 network bottleneck의 문제를 해결하는데 장애요인이 되고 있고, UNIX CPU로 실시간을 보장하기 어렵기 때문에 실시간을 요구하는 운영자에게는 이러한 대책이 강구되어야한다.

본 논문에서는 사용자에게 응용은 덜 요구되지만 매우 융통성이 있고, 편리하며, 시스템 성능면에서는 타 시스템과 비교해 지연시간이 짧은 시스템 설계 및 구현과 이러한 환경하에서의 분산 제어 시스템의 실시간 처리에 따른 성능 분석을 기술하고자 한다. 이를 위해서 DCS의 구성은 네트워크 파일 시스템을 지원하는 DB 구조를 채택하였으며 이러한 DB 구조의 단점이 네트워크 지연 시간임을 고려하여 실시간 처리를 지원하는 VME147 board를 이용하여 지연 시간을 단축시켰으며, 이러한 시스템 성능 분석을 위해 LOOP 및 LOGIC 처리에 따른 통신시간을 종합하여 측정하였다. ^{[25] [26] [27]}

본 논문의 구성은 Ⅱ장에서는 일반적인 분산 제어 시스템의 구성과 본 논문에서 구현한 시스템 사양을 검토하였고, Ⅲ장에서는 본 논문에서 필수적으로 요구되는 원격 제어국(R.S)에 대한 설명과 중앙 제어국(C.S) 및 원격 제어국의 s/w구성, 성능 분석에 필

요한 알고리즘을 기술하였다. Ⅳ장에서는 분산 제어 시스템의 구현에 따른 실시간 통신 성능 분석 및 검토를 하였다.

Ⅱ. 분산 제어 시스템

분산 제어 시스템의 중요한 구성요소는 다음과 같다.

(1) Process 접속장치 - 현장 계기와 제어 시스템의 연결고리로서 측정 신호의 변화, 제어연산, 제어값의 출력 역할을 담당하며 Input/Output Module, Controller Module 등이 여기에 속한다.

(2) Man-Machine Interface - 시스템내의 공정 제어 정보를 통합하여 운전자와 시스템과의 대화 창구가 되며 이 장치를 통하여 공장의 감시 및 운전을 하게되며 CRT, 프린터, 키보드 등이 여기에 속한다.

(3) Data 전송 장치 - 각각의 process 접속 장치와 Man-Machine Interface간에 자료를 전송하고 각 장비를 하나로 연결시키는 기능을 갖고 있으며 LAN 등이 여기에 속한다.

분산 제어 시스템의 장점은 기존의 아날로그 판넬 제어 시스템과 비교하여 보면 제공하는 잇점은 여러 측면에서 다음과 같은 장점이 있다.

(1) 제어연산이 자유롭다.

(2) 알고리즘 또는 복잡한 제어 루프의 연결이 용이하므로 고급 제어의 응용이 가능해진다.

(3) Man Machine Interface 기능이 강화됨에 따라 소수의 운전자가 많은 정보의 처리 및 제어를 용이하게 할 수 있다.

(4) 상위 컴퓨터(supervisory computer)와 접속이 가능해져 정보 처리 및 관리 기능이 강화된다.

(5) 시스템의 자체 고장 진단 및 프로세스의 이상진단 기능을 통해 플랜트의 안전 조업이 향상되게 된다.

(6) 원하는 목표치에 접근하여 조절 가능하므로써 생산성 향상과 품질의 균일화를 향상 시킬 수 있다.

(7) 정보 관리가 용이하다.

(8) 인력의 절감을 가져온다.

본 논문에서 구현한 분산 제어 시스템은 감시와 시스템조작을 위한 중앙 제어국(central station, CS)과 계측 제어를 위한 마이크로프로세서로서 입출력 콘트롤러를 포함하는 원격 제어국(remote station, R.S) 사이에 이중의 고속 Ethernet에 서로 연결된 하나의 총괄 시스템(total system)이며, 중앙 제어국은 Man Machine Interface기능을 가진 MMI 중심체로서 프로세스의 감시와 제어 및 유지관리를 할 수 있는 32비트 CPU로 구성되어 있으며, 중앙 제어국과 원격 제어국 두 시스템은 공통의 데이터 베이스를 갖고 다른 원

격 제어국으로 데이터의 전송이 가능하다. 각 원격 제어국안에 있는 각각의 마이크로프로세서는 LOOP 제어와 Sequence 제어를 처리 할 수 있으며 부분적인 고장이 전체 시스템에 영향을 미치지 않는 결합 내성을 갖추고 있고, 확장성도 고려되어 있다.

분산 제어 시스템의 기본 구성 방법은 실시간 제어 즉, 외부에서 발생한 요구에 대하여 제한된 시간 내에 처리할 수 있도록 제어를 하기 위한 각각의 제어 방식으로써 객체의 기본 단위를 블럭으로 구성하고, 이 블럭들은 일반적인 기능을 수행하는 알고리즘으로 구성하게 되며 통신망이 매우 중요한 분산 제어 시스템상에서 지연이나 부분 정지 등의 상황에서도 전체 시스템에 영향을 주지 않고 각각의 시스템들과 아니면 시스템 내부와의 통신이 이루어지도록 설계가 되었다. 분산 제어 시스템의 특징은 데이터가 어떤 메세지 프로토콜을 통하여 시스템을 공유되도록 요구되어지며, 분산 제어 시스템에 있어서는, 메세지 확인, 전송, 보안, 이식성에 대한 기술이 가장 중요하다. 본 논문의 시스템 구성은 다음 그림 1과 같다.

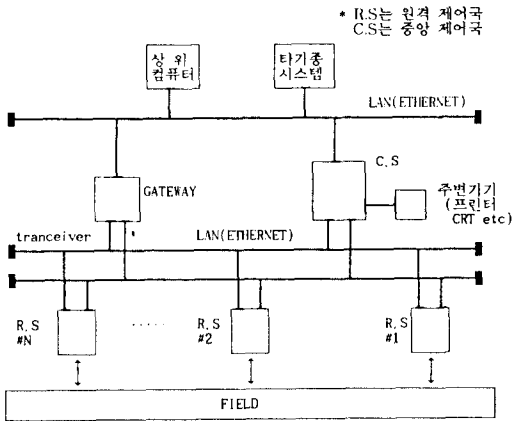


그림 1. 시스템의 전체 구성도
Fig. 1. Full configuration's drawing of a system.

그림에서의 상위 컴퓨터와 타기종 시스템은 OPEN SYSTEM을 구성할 때 연결시킬 수 있을 것을 고려한 것이고 시스템의 안정성을 향상시키기 위하여 통신로의 이중화를 본 논문에서 구현하였다. [15] [16]

대량의 데이터 처리를 위한 응용분야에 적합한 수행을 위하여 감시조작을 위한 인터페이스 기능으로 중앙 제어국의 구성 환경은 32bit CPU (MC 68030+MC 68882)를 채용하여 속도와 조작성을 향상시키기 위하여 범용 LAN (IEEE 802.3, Eth-ernet)을 사용하여, 10Mbps로 이기종의 시스템과 고속의 데이터 전

송이 가능하도록 했으며 각 시스템 및 제어기(controller)내의 시스템 버스는 VME bus (IEEE 1014) 방식을 사용하였다. [15] [18] [19]

이러한 VME bus는 멀티 프로세서에 적당하고 주종 CPU 보드의 비동기성 지원 및 비다중 데이터 전송구조와 7 종류의 우선권을 갖는 인터럽트를 지원하며 본 논문에서 채용한 Motorola CPU 보드를 지원하는 데 가장 적합한 데이터 버스로써 공장과 필드와 같이 충격과 진동, 부식성 등의 우려가 있는 곳에서 강한 내구성을 가지고 있다. 그리고 통신 프로토콜은 Open System Architecture를 지원하기 위해 TCP/IP 프로토콜을 사용하였고, [14] 범용 OS는 다중 프로세스 다중 사용자 (multi-process multi-user) O.S인 UNIX를 사용하고, 데이터 관리용 O.S는 소프트웨어가 간단한 Real Time Multi Task O.S인 VMEexec을 사용하였다.

1. 중앙 제어국의 기능

그림 1의 시스템 구성도에서 중앙 제어국은 시스템 내의 모든 상태 및 조건을 감시하여 중앙 제어국에 각각의 데이터 변화를 표시하며 원격 제어국에서의 제어에 필요한 항목을 수정하여 제어한다. 중앙 제어국은 원격 제어국에 측정자료 및 제어신호를 보낼 수 있고, 각 원격 제어국으로부터 상황과 측정자료를 받을 수 있다. 중앙 제어국의 감시는 모든 플랜트로부터 필요한 데이터를 원격 제어국으로부터 통신 콘트롤러 모듈을 통해 수집하여 기기 상태를 표시하며, 측정 자료를 표시하는 기능을 가진다.

On-line 처리의 빠른 응답 만으로 실시간 처리의 요구사항을 만족시킬 수 없으므로 주기적인 경우와 비주기적인 경우의 사건을 위해 sensor-based 처리, clock-based 처리, interactive 처리의 3부분으로 나누어 원격 제어국으로부터 데이터 수집, 플랜트 감시, 조작 및 제어를 하며 CRT 화면 처리, 키보드로부터 데이터 입력 처리 등의 명령을 위해서 제한된 시간내에 필요한 서비스를 수행하도록 기능을 강화하였으며 중앙 제어국에는 TCP/IP 사양에 따른 근거리 통신망 콘트롤러 및 serial 4 port, parallel 1 port에 인터페이스기능을 갖는 범용 연산 CPU 카드와 원격 제어국과의 교신을 위한 제반 프로토콜 처리를 할 수 있는 CPU 카드로 구성되고 이 각각의 카드들은 시스템의 표준 버스인 VME BUS (IEEE 1014)에 접속된다. 이러한 중앙 제어국의 주요 기능으로는 감시 및 제어 기능, 프로세스 관리를 위한 프로세스 데이터 베이스 구축 기능 및 RAS [신뢰도 (reliability), 유용도 (availability), 실용도

(serviceability)] 기능이 있다.

2. 고속 데이터 전송의 ETHERNET

전체 시스템의 데이터 통신 기능을 강화하기 위하여 Ethernet를 채용하여 중앙 제어국과 원격 제어국간 또는 Ethernet상의 컴퓨터 및 타 시스템과의 접속을 위하여 범용 근거리 통신망(LAN)을 사용하였다.

데이터 전송은 전체 시스템을 구성하고 있는 각 국(station)과 상위 컴퓨터 및 타 시스템과의 데이터 교환을 하기위하여 Ethernet(IEEE802.3)을 채용하여 고속으로 데이터를 통신할 수 있으며 전송로를 이중화하여 이상시에 대처가 용이하도록 하였다.^{[2] [4] [15] [16]}

3. 관계형 데이터 베이스 시스템(relational database system)

관계형 데이터 베이스 시스템이란 데이터 또는 정보의 집합체로서 데이터를 보관하고 갱신, 검색할 수 있는 기능을 가진 데이터 베이스 관리 시스템(database management system, DMS)으로 소프트웨어와 데이터를 보관하는 하드웨어로 구성되며 이러한 데이터 베이스의 특징으로는 중복(redundancy) 및 데이터의 불일치(inconsistency)를 피할 수 있고, 데이터의 표준화(standard) 및 보안(security)에 유리하다. 또한 데이터의 유지관리가 쉽고 데이터를 많은 시스템에서 공동으로 사용하게 할 수 있다.

이러한 데이터 베이스를 기초로 본 시스템은 계층형 구조나 망 구조 데이터 베이스보다 간단하고 우수한 관계형 데이터 베이스(RDB)를 채택하여 시스템을 구성하였으며, RDB의 기본적인 구조 체계는 그림 2와 같다.

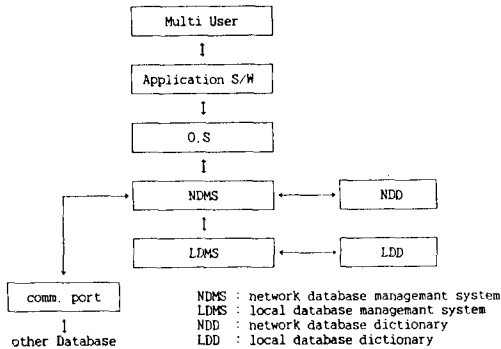


그림 2. RDB의 기본 구성 체계
Fig. 2. Based configuration of RDB.

RDB 구성 체계를 이용하면 분산 제어 시스템의

여러 TASK에서 실행중인 데이터의 경로가 처음에는 정해지지 않고 있다가 수정이 필요한 데이터 베이스가 있다면 실제 조회시 TASK의 경로가 그때 만들어 지므로 TASK의 수정이 용이하고 또한 한곳에서만 수정하면 전체 데이터 베이스의 수정을 가능하게 할 수 있는 장점이 있다. 따라서 논문에서는 UNIX O.S의 장점을 최대한 살릴 수 있는 데이터 구조와 기능에 모듈화 개념을 적용함으로써 중앙제어국의 LDMS(local database management system)를 CDMS(central database management system), 원격제어국의 LDMS를 RDMS(remote database management system)으로 표시하며 NDMS(network database management system)는 중앙제어국과 원격제어국의 통신을 관리하는 데이터베이스로 규정한다. 또한 각 분산된 TASK의 테이블 접근은 해쉬테이블을 이용하여 접근하며 접근된 TASK의 테이블에서 키(key)값을 받아 그 TASK의 정보를 얻어내도록 시스템을 설계하였으며 이러한 관계형 데이터 베이스의 개념을 이용하여 중앙 제어국의 기능별 TASK와 원격제어국의 기능별 TASK의 구성 및 C.S와 R.S 사이의 통신로와 모듈화의 데이터를 III장에서 제시하였다.^{[14] [16]}

III. 실시간 통신의 원격 제어국

1. 원격 제어국의 구성

실시간 시스템이라고 하면 시스템이 외부의 사건에 대하여 빠른 반응을 나타낼 수 있는 능력을 말하므로 이 경우 대부분 실시간 작업에 높은 우선순위를 주어서 서비스를 제공한다. 이러한 우선순위에 의한 실시간 서비스는 서비스의 수행시 각 원격 제어국에서 수행되고 정해진 시간 이내에 서비스를 제공하여야 한다. 따라서 본 논문의 원격 제어국은 LOOP 제어, LOGIC 제어등 실제적인 필드계기의 제어를 할 수 있는 제어국이며 규모나 제어 연산 스피드의 요구에 따른 연산 기능과 고 신뢰도의 운전을 위한 통신부로서 제어 대상에 최적의 시스템 구성을 실현할 수 있도록 설계되어 있다.

원격 제어국의 구성은 제어 처리 및 중앙 제어국과의 근거리 통신을 위한 32비트 CPU의 메인 콘트롤러 카드와 메모리 모듈 및 각종 I/O 카드들의 정보를 담당하는 ICM(I/O control module), ICM의 주기적인 데이터를 풀링하여 얻는 ILM(I/O link module)로 구성되어 지능적으로 자기 진단 기능 및 분담된 각각의 기능을 독자적으로 수행하도록 기능 분산형 multi-CPU 시스템(process control unit-

PCU)으로 구성되었다. 또한 각종 입,출력 모듈을 실장하여 입,출력기능을 담당하는 I/O 카드, 프로세서와 I/O 카드(I/O unit)사이에서 전기적 입,출력 신호의 변환 기능을 수행하는 인터페이스 카드(I/F unit), 각 유니트에 전원을 공급하는 전원장치(power distributed unit-PDU)들로 구성되어 있다. 원격 제어국의 시스템 구성도는 그림 3과 같고 PCU의 구성도는 그림 4와 같다. 그림 4의 전송모듈(transition module - TM)은 제어 처리 및 중앙 제어국과의 LAN통신을 위한 32 bit CPU와의 인터페이스를 지원하고 VME 버스를 통하여 고속으로 데이터를 전송하는 서비스를 지원한다.

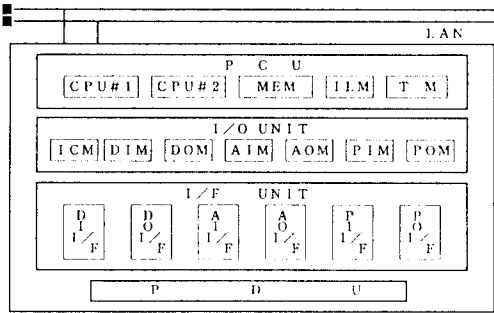


그림 3. 원격 제어국의 시스템 구성도
Fig. 3. System configuration's drawing of a remote station.

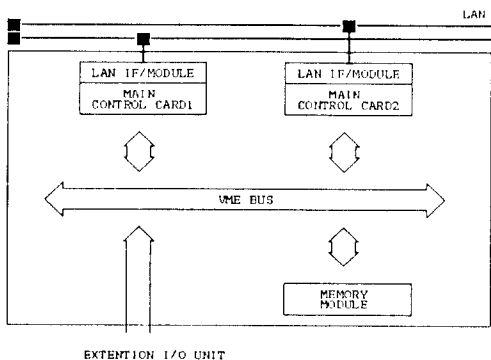


그림 4. Process Control Unit 구성
Fig. 4. Structure of a Process Control Unit.

PCU는 공정 이행, 이상 인터록, 프로세스 이상 등의 로직기능, PID 제어, 비율 제어, 프로그램 설정, 지시경보 등의 루프제어, 스위치, Annunciator, 논리 연산, 디지털 입 출력, 아날로그 입 출력 등의 공통 기능, 방송망(broadcast) 통신, Event 통신, 데이터 전송 등의 통신 기능을 수행한다.

PCU의 메인 컨트롤러 카드는 루프제어, 로직제어, 데이터 연산 처리, 데이터 통신을 하며, CPU는 32bit 마이크로프로세서인 MC68030을 사용하였고 memory module(MEM)은 battery backup 기능을 구비하여, 전원 이상이나, 시스템 이상시 데이터를 장시간 안전하게 보관하여 시스템의 신뢰도를 향상시킨다.

각 I/O UNIT중의 아날로그 입력 모듈 (AIM)은 DC 1 ~ 5V 신호를 디지털 데이터로 변환하는 기능을 가지며 아날로그 입력 인터페이스 모듈과 조합하여 사용하고, 아날로그 출력 모듈 (AOM)은 디지털 데이터를 아날로그로 변환하여 DC 1 ~ 5V 신호로 출력시키는 기능을 가지며 아날로그 출력 인터페이스 카드와 조합하여 사용한다.

또한 디지털 입력 모듈 (DIM)은 디지털 접점 입력을 받는 기능을 가지고 디지털 출력 모듈 (DOM)은 디지털 신호를 출력시키는 기능을 가지며 이들 각각의 카드들은 디지털 입,출력 인터페이스 카드와 조합하여 사용한다. 펄스 입,출력 모듈 (PIM, POM)은 전압형 펄스신호를 받아들여 적산하고 출력시키는 기능을 가지며 각각의 펄스 입,출력 인터페이스 카드와 조합하여 사용할 수 있도록 설계되어 있다.

디지털 인터페이스 카드는 디지털 입력, 출력 모듈과 접속되어 접점 신호 입력, 접점 신호 출력 및 내부, 외부와의 절연 기능을 행하고, 아날로그, 펄스 인터페이스 카드는 아날로그 입력 출력 모듈 및 펄스 입력 모듈과 field 신호 인터페이스 기능을 수행한다. 그리고, ILM과 ICM 사이는 bit bus를 이용하였으며, bit bus는 분산 제어 시스템을 위한 intel field bus로서 RS 485 인터페이스규격에 맞추어 250개의 부국(sub-station)을 트윈스트 페어로 연결한다. 정보 전송은 NRZI encoding과 16bit CRC 에러 처리 기술을 이용하여 IBM SDLC 프로토콜을 기본으로 한다. 이것의 데이터 전송률은 64 kbit/s에서 2.4 Mbit/s이다.

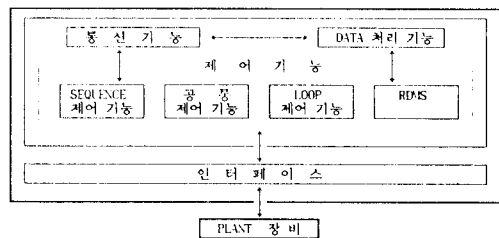


그림 5. 원격 제어국의 제어기능
Fig. 5. Drawing of control function in a remote station.

원격 제어국의 중요기능은 루프제어와 로직제어라 할 수 있으며 그림 5에 이러한 원격제어국의 제어기능을 나타내었다.

2. 원격 제어국의 TASK 구성

원격 제어국의 TASK 구성은 그림 6과 같으며 기능은 다음과 같다.

1) RSexec : bit_bus, module 처리

여러 TASK(task)에 의해 공유되는 global IO data area를 설정하고 초기화 시에 이 영역의 시작번지를 msg Q를 통하여 각 task에 보내며, bit_bus에 연결된 각 노드로부터 구성 정보를 읽어서 구성 정보 테이블에 저장하고 정주기 200ms로 각 노드에서 I/O 데이터를 읽어 들인다. 모듈(module)처리 루틴은 결선 주소 테이블이 지정하는 데이터를 읽어서 여러가지 처리를 실행하게 한다.

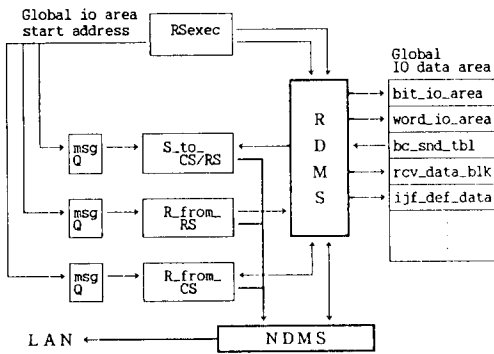


그림 6. 원격 제어국의 TASK 구성
Fig. 6. The configuration of Task in Remote Station.

2) S_to_CS/RS

RDMS의 bc_snd_tbl를 참조하여 다중 경로 (multicast) TASK인 NDMS로 데이터의 불럭을 알리면 NDMS에서는 근거리 통신망(LAN)을 통하여 중앙 제어국과 다른 원격 제어국의 RDMS 또는 CDMS로 데이터를 전송한다. 이때 원격 제어국이 데이터를 전송하고자 할 때는 방송망(broadcasting) 방식을 이용한다.

3) R_from_RS

다른 원격 제어국으로부터 온 정주기 데이터를 NDMS로 받아서 RDMS의 rcv_data_block에 저장하며, 이때 global IO data area 저장된 데이터의 내용은 RDMS table에 등록하여 추후 RSexec TASK의 호출이 있을 경우 사용할 수 있게 한다.

4) R_from_CS

중앙 제어국에 있는 MMI TASK의 비주기 명령을 받아서 RDMS의 테이블을 참조하여 RSexec으로의 실행을 보내며 실행 결과를 MMI TASK로 되돌려 준다.

5) RDMS

모든 원격 제어국의 TASK는 데이터의 입출력을 결정하기 위해 RDMS의 영역을 참조한다. 따라서 이 RDMS는 테이블에서 초기화시 각 TASK의 시작번지와 global IO data area의 모든 내용물들을 위한 포인터가 지정되고, 데이터의 입출력을 빠르게 하기 위하여 해쉬(hash) 테이블을 이용해 데이터들을 접근한다.

이상의 각각의 알고리즘을 기반으로 메시지 전달과정을 중앙 제어국과 원격 제어국으로 연결하여 직접 통신이 가능한 상태로 계층적 모델링을 구성하면 그림 7과 같다. 이들간의 연결은 10Mbps의 속도로 처리하는 Ethernet으로 연결되어 있고, 각각의 기능들을 잘 보여주고 있다.

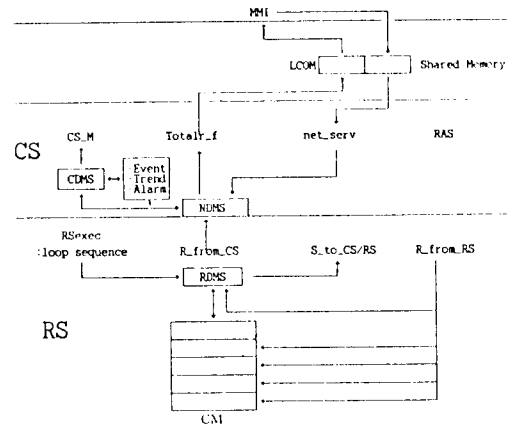


그림 7. 시스템의 계층적 구조
Fig. 7. Hierarchical structure of the system.

먼저 원격 제어국 층의 RSexec는 플랜트로부터 입력값을 받아 제어를 하는 과정이며 이때 아날로그의 처리는 200msec로 처리되어 메모리에 저장하고 RDMS에는 데이터의 정보를 담은 내용이 테이블로 보관된다. 메모리에 저장된 데이터는 불럭 단위로 저장되어 S_to_CS/RS에서 NDMS로 방송망 전송을 하게 되며, 이때 중앙 제어국으로 보내는 데이터는 CDMS 테이블의 시작영역과 끝나는 영역을 포인터로 지시만하면 Total_r_f가 CDMS 공유 메모리(shared memory)의 내용을 LCOM에 보내어

MMI가 이 LCOM영역을 참조하게 한다. 결국은 중앙 제어국에서는 각 원격 제어국이 보내오는 데이터를 저장하기 위해서 원격 제어국의 메모리의 총 합보다 많아야 된다. 그리고 각 원격 제어국들은 원격 제어국이나 중앙 제어국으로부터 오는 데이터를 저장하기 위하여 각각의 메모리를 할당하게 된다.

그림 7의 R_from_RS는 다른 원격 제어국으로부터 들어오는 데이터를 받아서 저장하고 이러한 데이터의 저장은 RDMS의 테이블에 등록한다. 아날로그로 들어온 데이터의 전송시에는 메모리에 비트값으로 저장하여 데이터는 블럭단위로 1000 msec안에 전송이 끝나야 한다. 여기서 한 블럭(block)의 전송량은 98 kbyte 이다.

MMI에서 LCOM의 공유메모리에 메시지 Q로 데이터를 보내면 net_serv는 LCOM의 채널을 통하여 비정주기로 데이터를 받아 NDMS로 전송하고, NDMS에서는 CDMS의 데이터 영역을 참조하여 다른 RDMS가 받을 준비가 되어 있다면 CDMS 공유메모리의 영역을 R_from_CS로 보내고 RDMS_table에 등록한다. 여기에 대한 응답은 Totalr_f로 보내어 다시 LCOM의 공유메모리로 보내서 MMI로 처리 데이터를 전송한다.

중앙 제어국의 Csexec는 CDMS의 multicast data area를 설정하고 이 영역을 UNIX상의 MMI 타스크와 공유할 수 있도록 준비하며, event, trend, alarm 타스크도 이 multicast 영역을 공유하므로 시작시 영역의 시작 번지를 메시지 Q를 통하여 각 타스크로 보낸다. 또한, 근거리 통신망을 통해 원격 제어국으로부터 들어온 multicast 데이터는 CDMS_table에 등록하여 각 원격 제어국별로 할당된 영역에 저장하며, 이 저장영역을 Csexec은 참조하여 실행한다.

각각의 event, trend, alarm 타스크는 모두 정주기 1초로 프로세스 정보, 시스템 정보 및 trend 데이터 수집, 또는 데이터의 수집의 합계, 평균을 내고 하드디스크에 저장하는 타스크이다. 그림 8은 LOOP 처리의 전처리 과정을 거친 입력 데이터를 컴퓨터의 값인 -2,500 ~ 10,000 값으로 환산하여 LOGIC처리를 한후 결과를 중앙으로 보내는 과정을 C-언어로 작성한 알고리즘을 보였다. LOOP나 LOGIC 처리의 수행이 되지 않았다면 총 수행시간에서 제외를 시켰으며, LOOP나 LOGIC 처리가 수행되었다 하더라도 네트워크를 통하여 중앙에 보고하는 통신과정의 실패는 총시간에 더해졌다. 이유는 통신로상의 문제는 해당 원격제어국의 제어와는 아무런 관련이 없기때문에 LOOP 및 LOGIC 처리는 성공적이라고 가정하는 것

이다. 네트워크의 성능분석은 TCP/IP 프로토콜을 따르며 데이터 패킷의 크기는 100K, 1,000K, 10,000K로 보내 측정시간을 구했다.

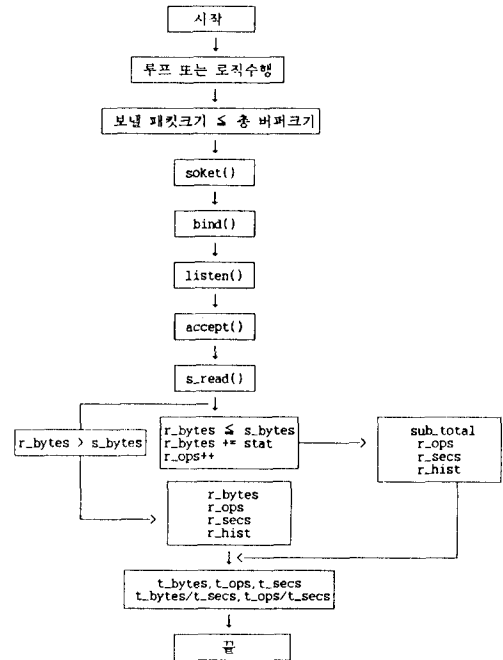


그림 8. 처리 알고리즘

Fig. 8. Algorithm of test processing.

IV. 실시간 처리시스템의 성능분석

원격 제어국의 실시간 처리기능은 크게 LOOP 와 LOGIC 2가지 처리이며, III장에서 제시한 알고리즘 수행을 위해 LOOP 처리기능은 입력처리1 결선번지로부터 입력 데이터를 받아 LOOP처리를 위한 내부계기처리를 위한 내부계기 처리결선번지로서의 analog 데이터 값들의 연산이나 적산 수행후 출력 데이터를 적당한 형태 즉 디지털 출력, 아날로그 출력의 형태로 보내는 내부계기 출력 데이터를 거쳐 LOOP처리를 200msec이내로 처리시간을 수행하도록 설계하였으며, LOGIC처리를 위해 나머지 시간을 할당한다. LOGIC 처리기능은 입력신호 변환기를 거쳐 입력카드로 들어온 데이터를 sequence table에 있는 LOGIC 처리기능을 활용하도록 도와주는 입력처리를 거쳐 출력하는 형태로 데이터 처리시간을 계산한다. 따라서 우선순위는 LOOP처리이고, LOGIC 처리는 그 다음으로 수행된다. 이렇게 수행된 데이터는 원격 제어국에서 중앙 제어국으로 감

시를 위해 정주기로 방송망 전송을 하게 된다. 원격 제어국에서 LOOP 나 LOGIC 처리를 하고, 방송망 전송을 위해 DB구조를 고려하여 원격제어국은 정주기로 중앙제어국의 메모리를 갱신시키는 데이터 크기는 98K (RAS 4K, table 갱신 4K, LOGIC 40K, LOOP 50K)가 요구된다.

표 1에서의 실행 바이트는 통신로상의 데이터가 방송망 전송을 하는 과정이고, ops는 버퍼에 패킷을 몇 번 보내었는가를 측정하는 데이터이다. COUNT NO. 1,2,3은 통신로상에서 LOOP, LOGIC 처리가 끝나 데이터를 보내는 것을 의미하며, COUNT NO. 4는 LOGIC처리만을 수행하며, 5 ~ 10 번까지는 테이블 갱신 및 RAS 데이터 정보만 보내는 데이터 처리 바이트를 나타낸다. 이것은 본 논문에서 원격 제어국의 DB를 고려하여 메모리 크기를 고려한 98K에 해당하는 실험 데이터이다. LOGIC 이나 LOOP 처리는 kernel구성에서 TASK 실행시간을 각각 100msec, 200msec로 만들어 처리하였고, 표 1에서 얻어진 평균 전송 시간인 0.5sec의 시간을 더 하고, 만약 3번의 실패 경우(한번 실패의 경우 0.2sec)를 고려한다해도 약 1sec 정도로 원격 제어국에서 중앙 제어국으로 데이터를 처리함을 보인다.

표 1. RS간 측정 데이터(I)

Table 1. Measured data between RS.

COUNT NO.	실행 바이트		ops		전송 바이트 : 100 Kbytes		바이트/처리시간		ops / 처리시간	
	write	read	write	read	write	read	write	read	write	read
1	102500	/ 102499	1025	/ 91	102500	/ 102499	1025	/ 91	1025	/ 45
2	102500	/ 102499	1025	/ 91	51250	/ 51249	512	/ 45		
3	102500	/ 102499	1025	/ 91	102500	/ 102499	1025	/ 91		
4	43000	/ 42999	430	/ 38	43000	/ 42999	430	/ 38		
5	6600	/ 6599	66	/ 7	0	/ 0	0	/ 0		
6	2000	/ 1999	20	/ 3	0	/ 0	0	/ 0		
7	1800	/ 1799	18	/ 3	0	/ 0	0	/ 0		
8	1700	/ 1699	17	/ 3	0	/ 0	0	/ 0		
9	1600	/ 1599	16	/ 3	0	/ 0	0	/ 0		
10	1500	/ 1499	15	/ 3	0	/ 0	0	/ 0		

Total Bytes: 366800 ops: 3668 sec: 5 bytes/sec: 73360 ops/sec: 733
 Total Bytes: 366700 ops: 333 sec: 5 bytes/sec: 73368 ops/sec: 66

표 2의 내용은 LOGIC 이나 LOOP의 수행을 함에 있어서 10 LOOP 처리에 대한 과정을 제외하고는 표 1의 실험과정과 동일하게 반복한 결과이다. 표 3의 내용은 100 LOOP수행을 전체 처리한 과정으로서 평균 1231sec를 얻은 결과이다. 그러나 공정상 100 LOOP 수행은 사용하지 않기 때문에 실험 데이터로의 결과만을 보였다. 결과적으로 표 1,4,5는 하나의 RS로 들어온 데이터의 신호를 받아 LOOP나 LOGIC 처리를 하여 다른 RS 나 MMI로 보내는 데이터 처리시간을 측정할 기록으로 버퍼에서 패킷을 읽고 쓰는 시간은 정수형으로 선언을 하였고 LOGIC 이나 LOOP 수행후

패킷이 수신되거나 실패된 것을 총 합하여 얻어진 시간은 100K 패킷의 경우 5sec, 1000K 패킷의 경우 39sec, 10000K 패킷은 1231sec로 논문에서 구현한 메모리 크기100K 패킷의 경우 평균 0.5sec로 실시간 처리에 적당함을 보여준다.

표 2. 실시간 처리시간 측정 데이터(II)

Table 2. Measured data of real time processing time.

COUNT NO.	실행 바이트		ops		전송 바이트 : 1000 Kbytes		바이트/처리시간		ops / 처리시간	
	write	read	write	read	write	read	write	read	write	read
1	1024100	/ 1024099	10241	/ 896	85341	/ 85341	853	/ 74		
2	1024100	/ 1024099	10241	/ 897	85341	/ 85341	853	/ 74		
3	1024100	/ 1024099	10241	/ 893	78776	/ 78776	787	/ 68		
4	22500	/ 22499	225	/ 20	0	/ 0	0	/ 0		
5	2600	/ 2599	26	/ 3	0	/ 0	0	/ 0		
6	2000	/ 1999	20	/ 4	0	/ 0	0	/ 0		
7	1500	/ 1499	15	/ 1	0	/ 0	0	/ 0		
8	1300	/ 1299	13	/ 1	0	/ 0	0	/ 0		
9	1100	/ 1099	11	/ 1	0	/ 0	0	/ 0		
10	1000	/ 999	10	/ 1	0	/ 0	0	/ 0		

Total Bytes: 3100000 ops: 31000 sec: 38 bytes/sec: 8880 ops/sec: 818
 Total Bytes: 3100800 ops: 2723 sec: 41 bytes/sec: 7560 ops/sec: 66

표 3. 실시간 처리시간 측정 데이터(III)

Table 3. Measured data of real time processing time.

COUNT NO.	실행 바이트		ops		전송 바이트 : 10000 Kbytes		바이트/처리시간		ops / 처리시간	
	write	read	write	read	write	read	write	read	write	read
1	10240100	/ 10240099	102401	/ 8942	83252	/ 83252	832	/ 72		
2	10240100	/ 10240099	102401	/ 8952	83252	/ 83252	832	/ 72		
3	10240100	/ 10240099	102401	/ 8943	83252	/ 83252	832	/ 72		
4	10240100	/ 10240099	102401	/ 8945	83252	/ 83252	832	/ 72		
5	10240100	/ 10240099	102401	/ 8952	83252	/ 83252	832	/ 72		
6	10240100	/ 10240099	102401	/ 8937	83252	/ 83252	832	/ 72		
7	10240100	/ 10240099	102401	/ 8934	83252	/ 83252	832	/ 72		
8	10240100	/ 10240099	102401	/ 8929	83252	/ 83252	832	/ 72		
9	10240100	/ 10240099	102401	/ 8929	83581	/ 83581	825	/ 72		
10	10240100	/ 10240099	102401	/ 8929	83581	/ 93581	825	/ 72		

Total Bytes: 102401000 ops: 1024010 sec: 1231 bytes/sec: 83185 ops/sec: 831
 Total Bytes: 102400000 ops: 83304 sec: 1231 bytes/sec: 83185 ops/sec: 72

데이터 분석은 처리시간을 최소화 할 수 있는 처리 과정을 실행 시켜 보았으며, 이 결과자료는 분산 제어 시스템의 평가자료뿐만 아니라 처리시간의 비교자료로 사용할 수 있을 것이다.

V. 결론

기존의 네트워크 download 분산 시스템들은 데이터를 수정하고자 할때 TASK들의 데이터를 로컬 메모리로 공유하는 것이 아니라 복사하므로 해당하는 TASK의 수정이 불가피하여, 운영자에게 불편과 통신상의 문제점이 야기되고 간편하게 하나의 TASK만 수정하면 되는 시스템을 설계할 때 테이블 참조 지연이라는 성능 저하를 초래하는 문제점이 발생된다.

본 논문에서는 기존의 데이터 베이스 시스템에서 I/O가 많으면 관리에 어려움이 많았고 변경시 많은 수정을 해야했던 문제점들을 극복하기 위해 관계형 데이터 베이스를 이용하여 분산 시스템을 설계하였으며, 테이블 참조 지연에 관한 문제는 해쉬 테이블 도입 및 실시간 O.S 채택으로 해결하였다. 또한 On-line상에서도 데이터 저장과 실행이 다른 타스크를 돕으로써 변경을 할 수 있는 장점이 있다. 성능 분석은 아나로그신호 4 ~ 20 mA를 field에서 받아 CPU에서 -2,500 ~ 10,000까지의 컴퓨터 값으로 데이터를 바꾸는 LOOP 처리 기능과 해당 DO point의 bit를 setting하는 LOGIC처리 기능을 수행하는 과정을 분석하였다. 이러한 모든 DB구조를 고려하여 시스템을 구현하면 DB의 데이터를 정주기로 갱신시키는 메모리 크기가 98K(RAS 4K, table 갱신 4K, LOGIC 40K, LOOP 50K)가 요구되고, 시스템 성능은 IV장의 데이터 메모리 크기를 갖는 시스템에서 실시간 처리에 만족함을 보였다.

앞으로의 연구과제로서는 관계형 데이터 베이스의 테이블 접근시 많은 지연 시간으로 인하여 이에 대한 해결 및 CPU의 이중화를 이루어 안정성을 고려한 빠른 시스템 처리, 프로세스의 효율적인 관리 및 확장성을 고려한 시스템에 관하여 더 많은 연구가 수행되어야 할 것이다.

參考文獻

[1] Andrzej Goscinski, "Distributed Operating Systems : The Logic Design", Addison-Wesley Pub. Company, Inc., Australia, 1991.

[2] Morris Sloman & Jeff Kramer, "Distributed System and Computer Network", Prentice-Hall, Inc., 1987.

[3] Popovic & Vijay P. Bhatkar, "Distributed Computer Control for Industrial Automation", Marcel Dekker, Inc., 1990.

[4] Sape Mullender, "Distributed Systems", ACM Press, New York, 1989.

[5] George F. Coulouris & J. Dollimore "Distributed Systems : concepts and Design", Addison-Wesley Pub. Company, Inc., London, 1988.

[6] George A. Champine, "Distributed Computer System : Impact On

Management, Design, And Analysis", North-Holland Pub. Company, 1980.

- [7] James Martin, "Computer Networks and Distributed Processing : Software, Techniques, and Architecture", Englewood Cliff, NJ : Prentice-Hall, Inc., 1981.
- [8] Andrew S. Tanenbaum & Robbert Van Renesse, "Distributed Operating Systems", Computing Survey, Vol.17, No.4, Dec. 1985.
- [9] David R. Cheriton, "The V Distributed Systems", Commun. of the ACM, Vol. 31, No.3, March 1988.
- [10] H. Zhou, K. Schwan and Ian F. Akyildiz, "Performance Effects of Information Sharing in a Distributed Multiprocessor Real-Time Scheduler", Proc. REAL-TIME SYSTEM SYMPOSIUM, Arizona, Dec. 1992.
- [11] 진 병현, "석유 화학 공업에서의 제어 시스템의 도입 사례", 제어 계측, pp 21 - 23, 11월 1992
- [12] 전익수, "차세대를 대비한 DCS의 방향", 제어 계측, pp5 - 10, 3월 1992
- [13] J. J. M. Hooman and W. P. de Roever, "Design and Verification in Real-Time Distributed Computing : An Introduction to Compositional Methods", Protocol Specification, Testing, and Verification, 37-56, 1990.
- [14] Douglas E. Comer, "Internetworking with TCP/IP(I) : Principles, Protocols, and Architecture", Englewood Cliff, NJ : Prentice-Hall Inc., 1991.
- [15] Metcalfe, R. M. and Boggs, D. R. , "ETHERNET : distributed packet switchig for local computer networks", Commun. ACM, Vol.17, 1976.
- [16] W. Richard Stevens, "UNIX NETWORKING Programmiang", Prentice-Hall Inc., 1991.
- [17] Phillip Laplante, "Real-Time Systems Design and Analysis : An Engineer's Handbook", IEEE Computer Society Press, 1992.

[18] 원 철호, "VME 버스", 월간 전자과학, 4월, 1987
 [19] "VME 버스", μ -PROCESS 개론, 삼성 종합 기술원
 [20] M.F.Kasahoe, A.S.Tanenbaum, S.F. Hummel, "An Efficient Reliable Broadcast Protocol", Distributed Journal, pp5-19, 1992.
 [21] "pROBE+ Debbuger User's Manual",

VMEExray Vol. IV ,Motorola.
 [22] "VMEExray Users's Reference Manual", VMEExray Vol. IV ,Motorola.
 [23] "VMEExgen Users's Reference Manual", VMEExray Vol. IV ,Motorola.

(※ 본 연구는 1993년도 건국대학교 자체연구 과제로 수행되었음)

— 著 者 紹 介 —



金來鎮(正會員)

1966年 3月 21日生. 1989年 2月 건국대학교 전자공학과 졸업(공학사). 1991年 8月 건국대학교 대학원 전자공학과 졸업(공학석사). 1991年 8月 ~ 현재 건국대학교 대학원 박사과정 재학중. 1991年 ~ 7月 현재 (주)광명제어 연구소 근무. 주관심 분야는 DCS, SCADA, UNIX networking 등임.



金珍泰(正會員)

1967年生. 1991年 2月 건국대학교 전자공학과 졸업(공학사). 1991年 2月 ~ 1992年 2月 삼성 전자 근무. 1993年 8月 ~ 현재 건국대학교 대학원 전자공학과 졸업(공학석사). 1994年 3月 ~ 현재 건국대학교 대학원 박사과정 재학중. 1993年 8月 ~ 현재 (주)광명제어 연구소 근무. 주관심 분야는 DCS, SCADA, Computer network 등임.



朴仁甲(正會員)

1950年 2月 1日生. 1973年 2月 고려대학교 전자공학과 졸업(공학사). 1986年 8月 고려대학교 대학원 전자공학과 졸업(공학박사). 1989年 7月 ~ 1990年 6月 Oregon state Univ. Post doc.. 1980年 9月 ~ 현재 건국대학교 전자공학과 교수. 주관심 분야는 computer network, parallel processing, distributed control system 등임.