

# 빈번한 변경이 요구되는 데이터의 효율적인 실시간 처리 기법

김진덕\* · 진교홍\*\* · 이성진\*\*\* · 정해원\*\*\*

\*동의대학교 컴퓨터공학과 · \*\*창원대학교 전자공학과 · \*\*\*두산중공업(주) 기술연구원

## An Efficient Real Time Processing Method for Frequently Updated Data

Jin-Deog Kim Kyo-Hong Jin Sung-Jin Lee Hae-Won Jung

Donggeui University, Changwon University, Doosan Heavy Industries

E-mail : jdk@deu.ac.kr

### 요 약

최근 다수의 외부 장치를 제어하는 운전원 모듈에서는 빈번하게 변하는 신호를 자동적으로 관리하는 시스템을 이용하고 있다. 이러한 모듈들은 높은 신뢰성과 실시간 처리를 필요로 한다. 데이터들은 매우 용량이 크며, 짧은 보고 주기, 비동기적인 보고 시간을 가진다. 시스템에 가장 일반적으로 사용되는 질의는 최신의 값을 검색하는 현재질의, 과거 특정시점의 값을 검색하는 스냅샷 질의, 과거부터 현재까지의 값들을 검색하는 이력 질의 등이 있다.

이 논문에서는 QNX 운영체제에서 파일 구조화된 데이터베이스를 이용하여 위와 같은 신호들을 효율적으로 관리하는 기법을 제안하고자한다. 장치들 간의 데이터 통신은 Profibus-FMS 프로토콜을 이용하며, 모니터 주기를 자유롭게 설정하거나 데이터를 저장하기 위해 파일 데이터베이스를 이용한다. 파일 데이터베이스는 QNX COM의 적은 저장장치, 낮은 계산 능력을 감안하여 델타버전과 주기적 백업 등의 방법을 도입한다.

### ABSTRACT

Recently, the operator modules to control external devices are concerned about automatic management system to process continuously changed signals. They need a efficient data management with high reliability and real time processing. The characteristics of these data are a large volume, a short report interval and asynchronous report time. The typical queries of these systems consist of the current query to search the latest signal value, the snapshot query to search the signal value of a past time, the historical query to search the signal value of a past time to current.

In this paper, we propose the efficient method to manage the above signals by using a file structured database in QNX operating systems. The data communications among the devices are done by Profibus-FMS protocol and the file databases are used for adjusting monitoring frequency and storing signals. The file database adopts a delta version and a periodical back up in due consideration of the resource limit of a small storage and a low computing power in QNX COM(Cabinet Operator Module).

### 키워드

파일 버전, COM, Profibus-FMS 통신, 델타버전

### 1. 서 론

산업 생산 현장의 자동화가 활발히 진행되는 가운데 최근 각종 외부 기기와의 통신을 통해 상태를 실시간으로 감시하고, 제어하는 시스템을 많이 이용하고 있다. 원자로 보호 계통 운전원 모듈 또한 그 대표적인 예이다. 국내 유일의 발전설비 전문업체인 두산중공업(주)의 기술연구원은 디지털 원자로 안전계통 제작 및 시험평가 중에 있다. 기 개발된 원자로 보호계통은 핵증기

공급계통의 상태를 감시하여 정해진 안전계통 설정치에 도달하면 원자로를 정지시키는 기능을 제공하며, 계측채널의 신호를 수집하고 설정치를 비교하는 각종 주변 기기 및 제어스위치로 구성된 캐비닛운전원모듈(COM), 개시회로 등으로 구성되어 있다.

그러나 지금까지의 원자로 보호계통 운전원 모듈은 각 외부 장치로 부터의 현재 신호를 점검하여 동작상태를 검증하는 방식이었다. 반면, 복잡해지고, 다양해지는 원자로 안전장치에 대한

과거 특정 시점의 신호 상태나 과거부터 현재까지의 이력 정보를 검색하기 위한 기능을 구현되지 못하고 있다.

따라서 본 논문에서는 원자로 보호시스템의 구성 모듈로부터 입력되는 신호 정보들을 실시간으로 데이터베이스화하고, 현재와 과거 신호 및 이력 신호 정보를 검색할 수 있는 DB 모듈을 설계하고자 한다. 메시지 송수신 수단으로 Profibus-FMS 프로토콜을 사용하며, 빈번한 변경이 요구되는 신호의 실시간 처리를 위해 QNX 기반 파일 구조의 DB 시스템을 제안하며, 질의 검색을 위한 API 및 성능 향상 기법을 제안한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 현재 사용 중인 원자로 안전 계통 모듈의 전체 시스템 구조 및 특징을 살펴보고, 제 3장에서는 이 논문에서 설계하고자 하는 DB 시스템 구성도 및 데이터 송수신 모듈, 질의 API 등에 대해 자세히 설명하며, 제 4장에서 결론을 내린다.

## II. 원자로 보호계통 구성도

기존의 원자로 보호계통 시스템은 핵증기공급 계통의 상태를 감시하여 정해진 안전계통 설정치에 도달하면 원자로를 정지시키는 기능을 제공하며, 계측채널의 신호를 수집하고 설정치를 비교하는 비교논리프로세서(BP), 다중 채널의 트립신호를 Voting하는 동시논리프로세서(CP), 시험 및 유지보수를 관장하는 자동시험 및 연계프로세서(ATIP), 운전원화면 및 제어스위치로 구성된 캐비닛운전원모듈(COM), 개시회로 등으로 구성되어 있다. 또한 이들 각 구성장치들은 Profibus-FMS 프로토콜[2,3]을 이용하여 정보를 송수신하도록 연결되어 있다.

운전원 모듈의 운영체제로는 QNX[1]를 사용하고 있다. QNX는 실시간운영체제의 한 가지 종류로 최근에 들어 uITRON, LINUX 등과 함께 임베디드시스템을 위한 운영체제로 각광받고 있으며, 특히 국내의 자동차 회사에서는 QNX Neutrino를 자사의 텔레매틱스를 위한 운영체제로 채택하고 있다[3]. QNX는 1980년 IBM-PC용의 상용 실시간운영체제로 시작하여 20년 동안 PC용 상용제어시스템의 분야에서 발전해오고 있다.

Profibus는 산업용 표준으로 널리 사용되고 있는 필드버스의 한 종류이다. 주로 PLC, CNC, RC 등의 산업용 제어장치와 센서, 액추에이터 등의 장치 등을 연결하기 위해 사용된다. Profibus는 실시간 처리 요구조건을 만족한다.

본 논문에서는 PhAB를 이용하여 운전원 모듈의 GUI 응용프로그램을 지원하며, Profibus-FMS 카드를 이용하여 운전원 모듈에서 필요로 하는 정보를 수집하고 전달한다. 이에 따라 Profibus-FMS 카드를 이용한 메시지 처리 루틴을 개발하여야 하는데 현재 고려중인 Profibus-FMS 카드인 Hilscher사의 CIF-50PB, 80PB를 위한 API를 이용한다[5].

## III. 신호의 실시간 처리 기법

### 3.1 COM DB 시스템 구성도

COM 모듈은 BP, CP, ATIP 등으로부터 시험신호를 받아들여 각 장치의 상태를 Operator에게 보고하는 역할을 수행한다. 그러나 지금까지 시스템은 누적되는 신호에 대한 관리하여 과거 정보 및 통계 정보를 관찰하는 것이 불가능하다.

이에 본 논문에서는 그림 1과 같이 주 운전원 모듈인 COM과 외부 장치 간에 데이터베이스를 두어 실시간으로 변경되는 신호들을 체계적으로 관리하며, 질의에 응답할 수 있는 환경을 제공한다.

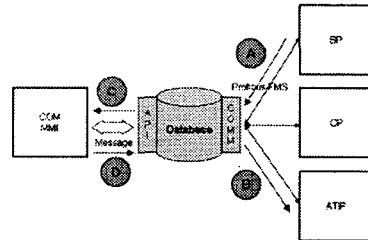


그림 1. COM DB 시스템 구성도

현재 각 장치로부터 전달되는 신호는 약 16,000개에 다르며, 이들 신호들의 대부분은 Historical Data로 유지되어 장치의 이전 상태도 언제든지 확인할 수 있다.

또한 COM 장비에서 이들 데이터를 수집할 때 얼마나 자주 수집하여야 하는지, 즉 각 신호별 또는 장치별 주기가 설정하며, Event Report Window에 출력될 신호는 별도로 처리한다.

위와 같은 신호별 주기, Historical Data 유지 등을 수행하기 위해 본 과제에서는 데이터베이스를 구축하여 COM의 MMI 프로그램을 보다 효율적으로 관리하고자 한다.

데이터베이스는 COM MMI 장비에 탑재되어 동작되며, COM 장비의 리소스 한계(적은 메모리, 낮은 연산능력)와 사용되는 질의에 적합한 파일 구조 데이터베이스를 설계하고자 한다.

데이터베이스 모듈이 제공하는 메시지 서비스는 크게 3가지로 구분된다.

Demand-driven Aperiodic Service(DAS)는 그림 2와 같이 COM MMI에서 사용자의 요구에 따라 특정 장비에게 데이터를 요구하고, 전달되는 데이터를 데이터베이스에 저장도 하고 그 결과 값을 COM MMI에게까지 전달하여 주는 서비스이다.

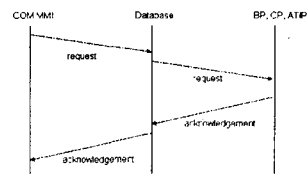


그림 2. DAS 서비스

Periodic Service(PS)는 그림 3과 같이 COM MMI에서 사용자의 요구가 없어도 정해진 주기에 따라 특정 장비에게 데이터를 요구하고, 전달되는 데이터를 데이터베이스에 저장하여 두는 서비스이다.

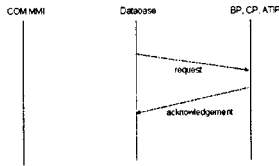


그림 3. PS 서비스

Alarm Service(AS)는 그림 4와 같이 BP, CP, ATIP에서 발생하는 event를 해당 장비가 COM MMI에게 전달하는 경우 이를 Database에서 수신하여 저장하고 COM MMI에게도 전달하여 주는 서비스이다.

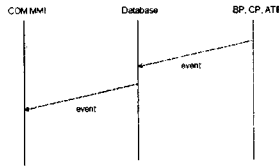


그림 4. AS 서비스

### 3.2 데이터 송수신 모듈

Profibus 카드를 이용한 통신을 수행은 그림 5와 같이 COM 모듈에 장착된 Profibus 카드와 각 외부 장치 Profibus 카드의 통신 버퍼가 대응되는 구조이다[4].

본 논문에서는 COM에서 DevPutMessage 함수와 DevGetMessage 함수를 이용하여 입력되는 신호를 획득하여 DB에 저장하는 방식을 택한다. 신호의 획득은 Profibus 카드내의 OD값을 읽거나 쓰기 작업을 수행함으로써 가능하며, 구조체 자료형인 MSG\_STRUC를 이용하여 보다 효율적인 신호의 송수신이 가능하다.

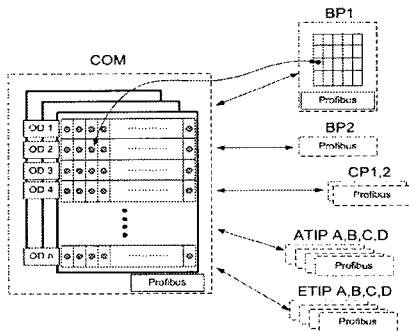


그림 5. Profibus를 이용한 통신

그림 5에서 각 Profibus에는 여러 개의 OD (Object Dictionary)가 있으며, 각 OD는 여러 개의 Address와 bit로 세분된다.

그림 6은 OD의 구조를 나타낸 것이다. 각 OD는 230바이트의 데이터영역이 있으며, Configuration 파일에 따라 각 Slot이 2, 4, 8 바이트로 구성된다.

그림 6의 case 1은 각 Slot이 2바이트이며 이 경우에는 각 Slot마다 정수값(int) 또는 부울(Boolean)값이 비트 단위로 저장된다. 그림 6의 case 2는 각 Slot이 4바이트이며 이 경우에는 각 Slot마다 실수값(real) 또는 부울(Boolean)값이 비트 단위로 저장된다. 이 외에도 1 또는 4 바이트 정수값을 저장하는 Slot을 지정할 수도 있다. 이 논문에서는 각 신호의 유형별로 OD를 지정하였으며, 해당 그룹 별로 색인의 주기를 달리하는 기법을 통해 효율적으로 데이터베이스화하는 방법을 제안한다.

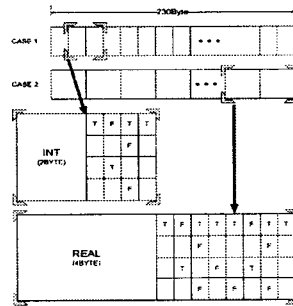


그림 6. OD 구조도

그림 7은 본 시스템에서 사용되는 신호의 예를 나타낸 것으로서 OD 칼럼은 OD의 번호를, Address는 OD 내에서의 Slot의 번호를, Bit는 각 Slot내에서의 비트 번호를 각각 의미한다. 그림 7에서 OD 20은 각 Slot이 2바이트 정수형 자료로 구성되며, Bool 신호는 성격에 따라 하나의 Slot을 차지할 수도 있고, 27번 Slot처럼 각 비트를 최대한 활용하여 16개의 Bool 신호를 하나의 Slot에 저장할 수도 있다.

그림 7. 원자로 보호계통 신호의 예

다음은 전술한 구조체 MSG\_STRUC의 구성요소 및 특징을 나타낸 것이다.

```

typedef struct tagMSG_STRUC {
    unsigned char rx; //Receiver
    unsigned char tx; //Transmitter
    unsigned char ln; //Length
    unsigned char nr; //Number
}
    
```

```

unsigned char a; //Answer
unsigned char f; //Fault
unsigned char b ; //Command
unsigned char e ; //Extension
unsigned char data[255]; //Data
unsigned char dummy[25]; // Compatible
} MSG_STRUC;
    
```

여기서 구성요소 In은 바이트단위로 tMessage.data[0]이후로의 개수를 의미하며, nr은 메시지의 ID를 의미한다. data[2]는 OD 번호이며, data[5]는 data[8] 이후의 데이터 개수이며, 데이터 송신 시에는 실제 데이터를 tMessage.data[8]이후에 저장하면 된다. data[6]은 각 Slot의 데이터 타입으로서 1:integer16, 2:integer8, 4:integer32, 5:unsigned int8, 6:unsigned int16, 7:unsigned int32 등등을 지정할 수 있으며, data[8~255]는 전송할 데이터가 저장된다. 이것은 data[6]에서 지정한 타입에 따라 배열의 한 element의 타입이 지정된다.

3.3 파일 DB 구축 및 API 모듈

그림 1에서 살펴본 바와 같이 원자로 안전계통 신호의 흐름은 크게 4가지로 분류할 수 있다.

그림 1의 A는 외부 장치들로부터 DB로 입력되는 신호들로서 정상 신호들(전체 16000여개의 Digital, Analog신호)에 대해서는 델타버전 기법을 통해 압축하고, Alarm 신호에 대해서는 신호 흐름도의 C와 연계하여 오퍼레이터에게 알려주고 DB에 기록하고 Alarm 로그를 남긴다.

그림 1의 B는 신호 흐름도의 D와 연계하여 외부 장치에 제어 신호를 전달한다.

그림 1의 C는 COM의 요청에 의해 파일 DB에서 COM으로 신호 값을 전달하는 과정으로 3가지의 질의와 하나의 alarm 이벤트로 구성된다. 3가지의 질의는 전송할 현재질의, Snapshot 질의, History 질의(그림 9)로 구성된다.

그림 1의 D는 COM에서 파일 DB로 데이터를 전송하는 것으로서 신호 흐름도 B와 연계된다. 주로 COM에서 외부 장치로 제어 신호를 보내는 것이며, 외부 장치의 ACK 신호의 처리결과를 DB에 로그는 남기는 역할을 수행한다.

3.2절에서 설명한 OD를 이용 방법을 통해 획득한 각 외부 장치의 신호를 파일 구조 기반 DB에 저장하고 관리한다. 원자로 보호계통 신호는 개수가 매우 많고, 빈번히 변하는 특성을 가지므로 이 논문에서는 델타버전 기법을 이용한다. 그림 8에서 주기적으로 전체 신호가 포함된 Master File를 구축하고, 일정 주기별로 이전 데이터와의 차이값만을 수용하는 Delta 파일을 만들어 QNX의 작은 저장구조에 적합하도록 하였다. 또한 검색 질의의 효율성을 위해 델타파일을 생성하는 주기 및 개수를 신호의 특성에 따라 달리하고자 한다.

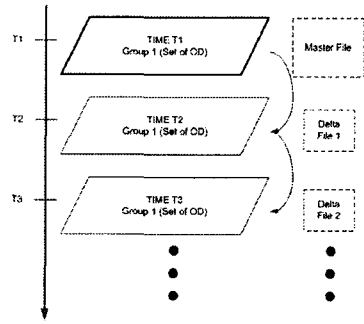


그림 8. Master 파일과 델타 파일

신호의 특성에 따라 OD를 여러 개의 그룹으로 분류하였으며, 분류기준은 다음과 같다.

- Input or Output
- 값의 변화 주기에 따라 보고주기의 장,중,단
- 데이터의 중요도에 따라 Alarm, Normal

위에서 Alarm 신호는 미리 정해진 특수 신호와 경계치 값을 벗어난 신호일 경우에 즉시 오퍼레이터에게 알려주는 역할을 하는 신호이다.

또한 본 논문에서는 OD의 그룹별로 색인을 달리 구성하여 질의 검색 속도의 향상을 도모하고자 한다.

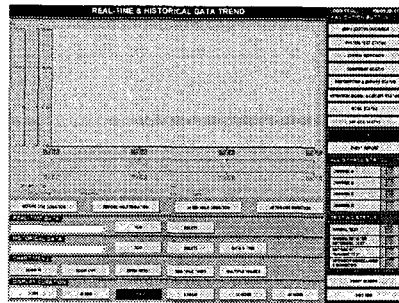


그림 9. PhAB에서의 History 질의

IV. 결론

이 논문에서는 QNX 4.25 운영체제에서 원자로 보호 계통 운전원 시스템의 각 외부 장치로부터 입력되는 빈번한 변경 연산이 요구되는 신호 정보들을 실시간으로 데이터베이스화하고, 현재와 과거 신호 및 이력 신호 정보를 효과적으로 검색할 수 있는 DB 모듈을 설계하였다.

특히, 운전원 모듈 COM의 리소스 한계를 극복하기 위해 Profibus의 각 OD를 신호의 특성에 따라 그룹화하였으며, 작은 저장 용량에 적합하게 델타버전 기법을 이용하여 DB를 구축하는 기법을 제안하였다.

이와 같은 연구 결과는 QNX를 널리 사용하는 국내의 텔레메틱스/스마트홈 산업에서도 기술의 활용도가 높을 것으로 판단된다. 향후 이 논문에서 설계한 기법을 구현하여 두산 중공업

에서 운용중인 원자로 보호 계통 시스템에 적용하고자 한다.

### 참고문헌

- [1]QNX Software Systems, <http://www.qnx.com/>
- [2]PROFIBUS International Open Solutions for the World of Automation, <http://www.profibus.com/>
- [3]"프로피넷 기술" 생산현장 속으로, 디지털타임스, Dec., 2004
- [4]"PROFIBUS Specification," Normative Parts of PROFIBUS - FMS, -DP, -PA according to the European Standard EN 50 170 Volume 2, Edition 1.0, March, 1998
- [5]hilscher, <http://www.hilscher.com/>