

## 유무선 통신을 이용한 DCS 터미널 원격 감시 시스템 구현

백영태\*, 이세훈\*\*

### Implementation of a DCS Terminal Remote Monitoring System using Wire and Wireless Communication

Yeong-Tae Baek\*, Lee Se Hoon\*\*

#### 요 약

이 논문에서는 터미널 부분의 중요 신호를 유무선 방식을 통하여 통합적으로 감시, 관리하기 위한 DCS 터미널 원격 감시 시스템을 설계 및 구현한다. 시스템은 터미널 부분의 릴레이들의 동작 상태 및 퓨즈의 이상 유무를 실시간으로 감시하기 위한 보드와 각 릴레이들의 동작 상태를 실시간으로 확인이 가능하며 예러 발생 시 예러 유형을 즉시 제공하며 릴레이의 규격 및 사용기간을 고려하여 교체시기를 알려주어 DCS 터미널 원격 감시 미들웨어로 구성된다. 이 시스템은 릴레이의 고장을 쉽게 진단할 뿐만 아니라 사전에 예방함으로써 산업분야의 경제적인 효과가 기대된다.

#### Abstract

In this paper, we designed and implemented DCS terminal remote monitoring middleware for real-time auditor and management terminal status signal with relay and fuse status at power management room. The middleware consist of four component as event analyzer, event manager, service manager and user interface. We expected economic effects extend relay life-time that pre-detected errors and prevented relay fault.

▶ Keyword : 원격감시 및 관리시스템(Remote Monitoring and Management System), DCS 터미널 감시 보드(DCS terminal monitoring board), 제어 및 모니터링 소프트웨어(Control and monitoring software), 릴레이(Relay)

---

• 제1저자 : 백영태  
• 접수일 : 2007.4.16, 심사일 : 2007.4.28, 심사완료일 : 2007. 5.2.  
\* 김포대학 IT학부 부교수 \*\* 인하공업전문대학 컴퓨터시스템과 교수

## I. 서론

사회 각 분야에 있어서 원격 제어 및 원격 감시 기술은 이제 일반화 되었으며 공장자동화뿐만 아니라 홈 네트워크에서도 필수적인 기술이 되었다. 최근에 웹과 네트워크를 기반으로 하여 여러 분야에 적용되는 원격 감시, 제어 시스템 개발에 관한 연구가 활발히 진행되고 있으며 이러한 자동화는 방대한 양의 물품을 생산해 내거나 위험한 작업이 이루어지는 발전소, 화학공장, 선박, 자동차, 반도체, 철강 및 수처리 산업 등에서 계측 제어 장비 및 센서 등을 자동 제어 시스템(DCS & PLC)으로 제어 하는 형태로써 사용된다. 그러나 현장계기와 자동제어시스템의 신호 연결부분인 터미널 부분은 인위적인 점검 및 조치에 의존하고 있어 많은 인력과 비용이 소요되고 있는 실정이다[1,2,3].

자동제어시스템의 신호전달 체계의 중간 단계인 전원인 가 신호의 터미네이션 부분은 일대일 유선 연결 방식으로 다량의 연결선, 릴레이 및 터미널 단자가 필요하며 이 부분의 통합기술은 일체형 보드로 이미 신형발전소에 적용되어 있으며 자동제어시스템의 용량 및 프로세스의 영향으로 대용량 자동 제어 시스템에만 적용가능하며, 유선 방식으로 이용하고 있다[1,5,6].

그러나 기존의 자동제어 시스템은 신호 터미널 부분을 위해 다량의 자재 및 설치공간의 대형화를 필요로 하며, 터미널 부분의 이상 여부를 육안 검사 등 인위적으로 진단해야 하는 불편함과 이상 발생 시 원인 분석 및 즉각 대처가 불가능하여 환경 감시, 전력 설비, 무인 공장, 원자력 제어 등과 같은 사람이 현장에서 직접 시스템을 운영하기 어려운 분야에 특성상 컴퓨터의 이용은 필수적이다[3,4,5,7].

따라서 이 논문에서는 터미널 부분의 중요 신호를 유무선 방식을 통하여 통합적으로 감시, 관리하기 위한 DCS 터미널 원격 감시 시스템을 설계하고 구현한다. 기존 개발된 일체형의 통합 터미널 보드의 유선방식을 배제하고 무선방식으로 전환하여 자동제어 시스템에서의 중요 신호에 대한 부분적인 관리가 아닌 전체적 통합관리 시스템 구축을 위한 DCS 터미널 원격 감시 보드로 터미널 부분의 릴레이들의 동작 상태 및 퓨즈의 이상 유무를 실시간으로 감시하기 위한 보드를 설계 제작하며, 원격감시 모니터링 미들웨어를 구현한다. 또 전체적 통합관리 시스템 구축은 상태의 변화가 이루어지거나 일정 시간마다의 데이터를 DB에 기록해두기 때문에 릴레이의 수명을 고려하여 릴레이의 고장을 예방 조치 할 수 있는 효과를 기대할 수 있다.

## II. 관련 기술 고찰

이 장에서는 산업현장의 DCS 터미널의 원격 감시 시스템 구축과 관련된 기술에 대해서 기술한다.

### 2.1 분산 제어 시스템

컴퓨터를 이용한 제어 개념은 60년대 중반부터 프로세스 제어분야에 도입되기 시작하였으며 초기의 컴퓨터를 이용한 제어개념은 70년대 중반까지 한 대의 컴퓨터에 의한 DDC(Direct Digital Control) 시스템과 현장 아날로그 연결에 의한 제어방식이 주종을 이루고 있었다. DDC 시스템이란 한 대의 컴퓨터에 프로세스 데이터의 입력, 출력 및 플랜트의 감시, 조작, 제어 등을 모두 집중화시켜 관리하는 시스템이다. DDC의 경우 모든 제어 기능이 한 대의 컴퓨터에 집중되어 있으므로 컴퓨터에 이상이 발생하면 공정 전체가 제어 불능 상태가 되어 신뢰성 저하가 문제가 되었다. 분산 제어 시스템(DCS: Distributed Control System)는 DDC의 단점을 보완하기 위하여 하나의 중앙처리 장치를 여러 개의 작은 중앙처리 장치로 나누어 기능별로 분리하고 작은 용량의 중앙처리 장치를 갖는 각각의 컴퓨터를 통신 네트워크로 연결시켜 전체 시스템을 구성하도록 되어 있다[8].

DCS의 기본개념은 공정제어에 적용되는 시스템을 각 플랜트에 알맞은 단위 하위 시스템으로 분리하고 각 소 단위 시스템에서는 각각의 주어진 역할을 수행하며, 상호간에 통신이 가능하도록 한 것으로서 DCS는 소형 DDC 시스템 여러 개를 유기적으로 연결하여 전체 시스템을 구성한 것이라 볼 수 있다.

DCS의 기본 특징은 프로세스 제어 기능을 여러 대의 컴퓨터에 분산시켜서 신뢰성은 향상시키고 이상 발생 시 그 파급 효과를 최소화시키며 프로세스 정보처리 및 운전 조작 그리고 분산 설치된 컴퓨터들의 관리기능 등은 중앙의 주 컴퓨터에 집중화시킴으로서 자료처리 및 운영관리를 원활하게 하는데 있다. 즉 DCS는 기능의 분산과 정보의 집중이라는 2가지 특징 사이에서 균형을 유지하면서 개발되었다[3,4].

DCS의 구성요소는 시스템 인터페이스, 프로세스 인터페이스, 운영자 인터페이스로 분리할 수 있다. 시스템 인터페이스는 CPU, Data Highway가 있다. 프로세스 인터페이스는 제어 대상과 직결해서 제어정보의 처리를 행하기 위한 컴퓨터의 입출력을 말하며 아날로그 입력 및 출력, 디지털 입력 및 출력 등이 있다. 운영자 인터페이스는 운영자가 필

요에 따라서 제어대상을 제어하기도 하고 공정 상태를 감시하기 위해서 준비된 조작스위치 및 상태표시 장치를 총칭하여 말한다. 최근에는 MMI(Man Machine Interface)기능을 강화하여 운영자의 부담을 경감하고, 고도의 상황판단 능력을 가지는 방향으로 발전되고 있다[8,9].

### 2.2 RS485 통신

RS485는 RS232나 RS422처럼 Full Duplex가 아닌 Half Duplex 전송방식만 지원하기 때문에 RS422의 Multi-Drop 모드의 슬레이브처럼 RS485의 모든 마스터는 TXD신호를 멀티포인트 버스에 접속 또는 단락시켜야하고 RXD신호 역시 모드에 따라서는 접속, 단락의 제어를 하여야 한다[10]. 멀티포인트는 RS485의 모든 마스터가 공유하는 신호라인을 뜻한다.

### 2.3 AVR 162 마이크로프로세서

Atmel사의 AVR 마이크로컨트롤러로써 Single 싸이클 명령 실행구조를 갖는 RISC형 프로세서이다[11,12,13].

효율적인 I/O포트 구조와 각 모델에 따라서 내부 발전 회로, 타이머, 시리얼 통신, SPI, AD변환기, 풀업저항, 펄스폭 변조제어, 아날로그 비교기 그리고 와치독 타이머 등이 내장 되어 있다. AVR 마이크로프로세서의 명령어들은 C언어나 기계어로 개발할 때 프로그램의 크기를 최적화하도록 되어있으며 AVR 시리즈에는 플래시 메모리가 내장되어 있어 짧은 시간 내에 최소의 비용으로 개발하려는 사용자들에게 적합한 마이크로 컨트롤러이다.

## III. DCS 터미널 원격 감시 보드 설계 및 구현

실제 산업현장에서는 전기 장비들에게 전원을 공급하는 수 많은 릴레이들이 캐비닛(cabinet)이라는 특정 장소에 위치하며 릴레이에서 에러가 발생했을 때 많은 릴레이 중에서 에러가 발생한 릴레이 위치와 에러 타입을 검출하여 메시지 형태로 원격감시서버에 전송하기 위한 보드를 설계한다.

### 3.1 시스템 구성

캐비닛 안에는 수많은 릴레이들이 위치하며 이 릴레이들의 상태를 실시간으로 체크하여 일정시간 단위 내지는 고장이 일어났을 경우 원격감시 서버에 이 사실을 알려준다.

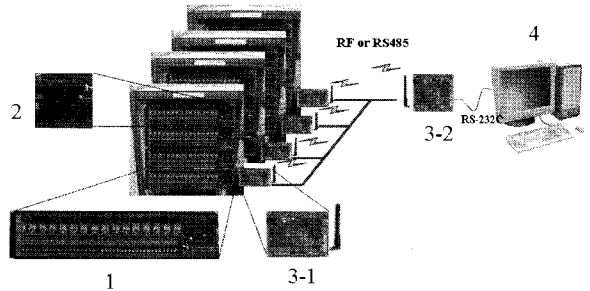


그림 1. 원격 감시 보드 및 시스템 구성  
Fig. 1 Remote Monitoring Board & System Configuration

원격감시 시스템은 릴레이의 상태와 퓨즈의 상태를 모니터링보드에서 체크하여 485통신을 이용하여 마스터보드에 전송한다. 마스터에서는 캐비닛 안의 릴레이의 상태를 모두 수신하게 되면 유무선 통신을 이용하여 중계 장치에 수집한 정보를 전송한다. 그림 2는 모니터링 보드와 마스터 보드의 통신 구성을 보여주고 있다.

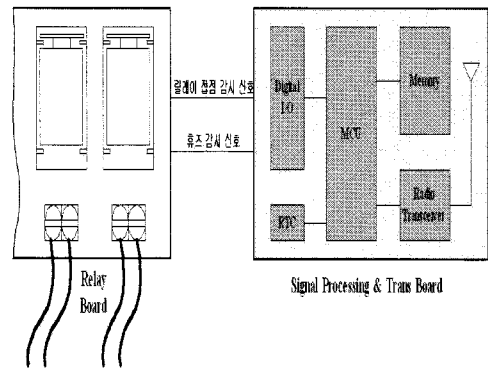


그림 2. 모니터링보드와 마스터보드의 통신 구성  
Fig. 2 Communication Configuration of Monitoring and Master Board

중계 장치에서는 각 캐비닛에서 전송된 상태 정보를 가지고 릴레이의 정상작동 및 고장여부를 판단하여 원격감시 서버에 정보를 전송한다.

### 3.2 릴레이보드 설계

릴레이는 전자적으로 제어가 가능한 스위치로서 구동 코일과 접점으로 나눌 수 있다. 구동 코일에 정격 전압이 인가되면 접점이 서로 붙게 된다. 여기서는 2점접 회로로 표현을 하였으나 실제 보드에서는 4점접 신호를 사용하고 있

다. 각 릴레이의 구동 코일과 연결된 두 단자 중 한 단자에 공통으로 24V의 전압이 인가되고 다른 단자는 DCS나 PLC의 출력이 연결되어 있다.

이 릴레이보드는 많은 전류를 사용하는 모터나, 밸브를 제어하기 위한 보드로 접점에는 L상과 N상의 전압을 인가할 수 있으며, 릴레이가 ON되면 접점에는 L상의 전압이, B접점에는 N상의 전압이 나타난다.

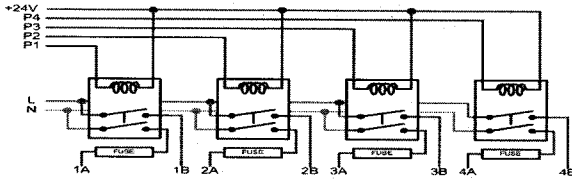


그림 3. 릴레이 보드 구성도  
Fig. 3 Relay Board Configuration

### 3.3 모니터링 보드

모니터링 보드의 MCU는 AT89C4051, Xtal 22.0592MHz이며, 릴레이 보드의 각각 16개의 퓨즈 및 접점 상태 검출 데이터는 쉬프트 레지스터를 거쳐 AT89C4051의 한 핀에 직렬로 데이터가 들어오고 마찬가지로 16개의 코일 검출 데이터도 Photo Coupler로 상태를 읽어와 쉬프트레지스터를 거쳐 AT89C4051에 직렬로 데이터가 들어오도록 구성하였다. MAX-485의 탑재로 RS-485통신을 하도록 구성하며, 통신상의 보드ID값은 I2C 메모리에 저장한다.

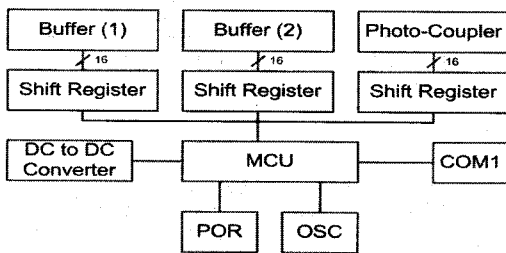


그림 4. 모니터링 보드의 구성  
Fig. 4 Monitoring Board Configuration

모니터링 보드의 동작 펌웨어는 우선 변수 ScanTime을 이용 일정 간격으로 Relay의 상태를 읽고, 남은 ScanTime의 시간에 수신된 패킷이 있는지 확인하며, 수신된 패킷이 있다면 패킷의 Func 부분을 확인하고 해당 명령을 수행한다. 펌웨어의 흐름도는 그림 5와 같다.

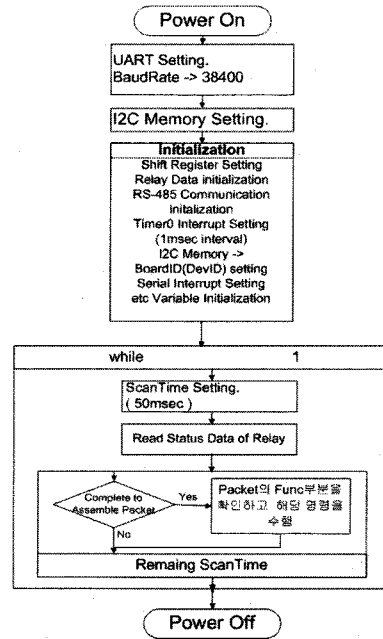


그림 5. 모니터링 펌웨어 흐름도  
Fig. 5 Monitoring Firmware Flowchart

패킷(packet) 구조는 다음과 같다.

ID	FUNC	Size	Data	Checksum
1byte	1byte	1byte	nbyte	1byte

ID는 Slave의 주소 01h ~ 7Fh만 사용 가능 최상위 1비트는 에러 표시이다. Func : 00h ~ 7Fh는 질의 80h에서 FFh까지는 응답이다. SIZE : data열의 바이트 단위의 크기이다. DATA는 N 바이트 열이다. CHECK SUM은 본 항을 제외한 패킷 모든 항을 더하고, 그 결과를 2의 보수로 취한 값이다. 질의응답 패킷의 구조도 동일하다.

### 3.4 마스터보드

마스터 보드의 MCU는 ATmega162, Xtal16.0000MHz이다. RS-485통신을 위한 MAX-485가 COM1, COM2에 연결되도록 두 개가 탑재되어 있고, RF모듈을 위한 슬롯 하나가 COM2에 연결되도록 있다. 이를 접퍼를 두어 COM1, COM2를 필요에 따라 MAX-485혹은 RF모듈과 연결가능하다. 모니터링 보드와 다르게 통신에 쓰이는 ID값은 내부 EEPROM에서 저장 관리 된다. 모니터링 보드와 485통신으로 제어하기 위해 COM1은 MAX485에 캐비넷

의 같은 485통신상에 있는 모니터링 보드의 릴레이 데이터들을 수집하여 서버에 전송하기 위한 무선 RF모듈이 COM2에 연결된다.

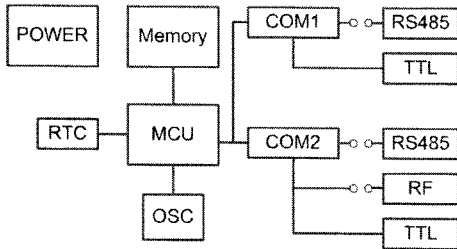


그림 6. 마스터 보드의 구성  
Fig. 6 Master Board configuration

마스터 보드 펌웨어는 우선 변수 Scan Time을 이용 일정 간격으로 캐비닛의 모니터링 보드에 패킷을 보내고, Time Out전까지 응답을 기다린다. 남은 시간에 수신된 패킷이 있는지 확인하고, 있다면 패킷의 Func 부분을 확인하고 해당 명령을 수행한다. 만약 응답이 없으면 해당 보드의 ID의 최상의 비트에 1을 Bit OR하여 통신 이상이 있음을 기록한다. 펌웨어의 흐름도는 그림 7과 같다.

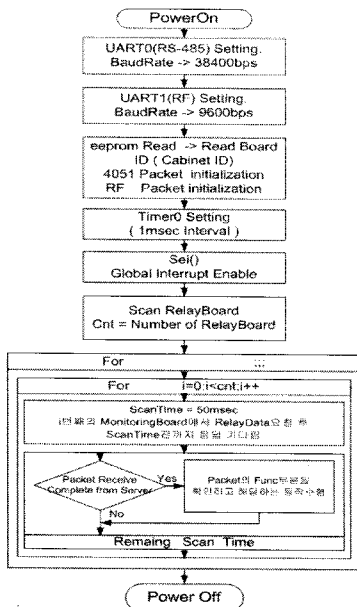


그림 7. 마스터 보드 펌웨어 흐름도  
Fig. 7 Master Board Firmware Flowchart

## IV. DCS 터미널 원격 감시 미들웨어 설계 및 구현

이 장에서는 산업현장에 설치되어 있는 수 많은 릴레이들을 원격에서 감시 및 관리를 하기 위해 미들웨어를 설계 및 구현한다. 설계하는 미들웨어는 캐비닛에 설치된 데이터 수집 및 전송 보드를 통해 릴레이들의 상태 변화에 따른 데이터를 받아 데이터를 분석, 처리하며 이 데이터를 효율적으로 관리하고 관리자에게 보다 편하게 데이터를 확인할 수 있도록 모니터링 정보를 표현해주는 역할을 담당한다.

### 4.1 시스템 개요

원격 감시 미들웨어는 릴레이보드에서 발생한 상태 정보를 원격감시보드를 통해 전송받아 데이터를 분석하여 고장이 발생한 릴레이 위치를 확인하고 이벤트에 맞는 서비스를 실행시킨다.

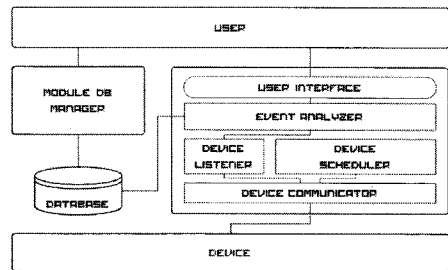


그림 8. 미들웨어 구성도  
Fig. 8 The Middleware Configuration

설계한 미들웨어의 전체 구성은 그림 8과 같으며 이벤트분석기, 사용자 인터페이스, DB관리자, 장치관리자로 구성된다.

### 4.2 이벤트 분석기

이벤트 분석기는 데이터 수집 및 전송 보드를 통해 전송된 데이터를 분석하는 과정을 수행하고 이를 이벤트 관리자에게 전송하는 역할을 한다. 이 데이터를 분석하는 과정에서 이 데이터가 어떤 형태의 이벤트인지를 분석하게 되는데 이 때 결정되는 이벤트의 유형은 표 1과 같다.

표 1 이벤트 유형  
Table 1 Event Type

이벤트 종류	발생 시기
휴즈 이상	릴레이 에러 발생시
코일 이상	
접점 이상	
릴레이 동작 횟수	릴레이 100회단위 동작시
릴레이 교체	릴레이 교체

이렇게 발생한 이벤트를 데이터화하여 이벤트관리자로 전송하며 아래 리스트는 이벤트관리자의 동작을 나타내는 코드이다.

```
EventAnalyzer(){
    dataStructure=DataAnalyzer(RecvData_232C())
    event=EventType(dataStructure.state)
    EventManager(dataStructure, event)
}
```

### 4.3 DB 관리자

DB 관리자는 이벤트 분석기에서 보내온 데이터를 관리하고 이를 데이터베이스에 저장하는 하는 역할을 하게 된다. 또한 이렇게 처리된 이벤트를 서비스 관리자에게 알려주어 이벤트에 해당하는 서비스를 지원하도록 하며 사용자인터페이스를 통해 입력된 사용자의 입력을 서비스 관리자를 통해 전달받아 처리하고 메시지를 전달하는 역할을 담당한다. 리스트 1은 DB관리자의 동작을 나타내는 코드이다.

```
리스트 1 DB 관리자의 동작 코드
List 1 Op code of DB Manger
DBManager(dataStructure, event){
    switch(event)
    {
        case FuseAbnormal, CoilAbnormal,
            ContactAbnormal :
            DB_Error_InsertOrUpdate(dataStructure)
            ServiceManager(event,dataStructure)
        case RelayCounter :
            DB_Count_InsertOrUpdate(dataStructure)
            ServiceManager(event,dataStructure)
        case ReplacementCounter :
            DB_Maintenance_InsertOrUpdate(dataStructure)
            ServiceManager(event,dataStructure)
        case Initialize_RelayCounter
            DB_Count_Update(dataStructure)
            WriteData_232C(dataStructure)
    }
}
```

데이터베이스는 캐비닛, 모듈, 릴레이, 알람 히스토리 등의 엔티티를 갖고며 관계도는 그림 9와 같다.

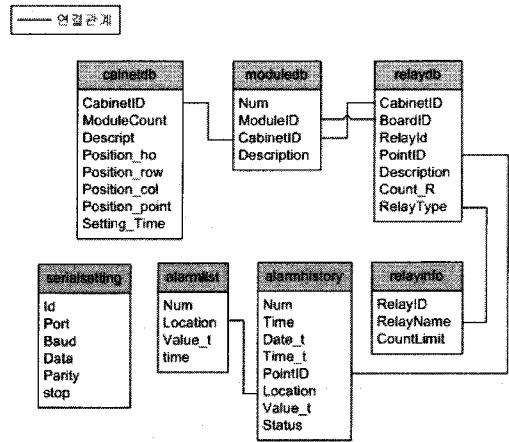


그림 9. 데이터베이스 관계도  
Fig. 9 Database Relationship Diagram

### 4.4 장치 관리자

장치 관리자는 DB 관리자로부터 받은 이벤트 데이터를 가지고 서비스를 수행하는 역할을 한다. 장치 관리자에서 지원하는 서비스는 세 가지가 있는데 이는 세 가지의 이벤트 유형과 매치된다.

#### ○ 휴즈 이상, 코일 이상, 접점 이상

휴즈 이상과 코일 이상, 접점 이상의 경우 에러로 판단하여 관리자에게 에러 발생 시기, 에러가 발생한 캐비닛, 릴레이보드, 릴레이 위치를 표시해주며 어떤 에러가 발생했는지를 저장하여 관리자의 GUI 인터페이스를 및 알람 등과 같은 다양한 방법으로 관리자에게 알려주는 역할을 한다.

#### ○ 릴레이 동작 횟수 검사

릴레이 동작 횟수는 100회 단위로 데이터 수집 및 전송 보드로부터 원격 감시 모듈웨어로 전송되며 서비스 관리자에서는 이 데이터와 데이터베이스의 해당 릴레이의 정격 동작 횟수를 비교하여 정격 동작횟수를 초과할 경우 관리자 인터페이스를 통해 해당 릴레이의 교체가 필요하다는 메시지를 알려준다. 또한 관리자가 지정한 사용기간이 경과된 릴레이에 대해서도 같은 방법으로 교체시기를 미리 알려준다.

#### ○ 릴레이 교체

릴레이 교체는 릴레이에서 에러가 발생하여 교체가 필요한 현장에서 직접 릴레이를 교체하게 되면 발생하는 것으로

데이터 수집 및 전송 보드를 통해 전송된 데이터를 장치 관리자에서는 관리자 인터페이스를 통해 디스플레이 해주는 역할을 하게 된다. 또한 이때 변경된 릴레이 정보와 교체 완료를 확인하는 정보를 데이터베이스에 반영하여 수정한다. 리스트 2는 장치관리자의 동작을 나타내는 코드이다.

```

리스트 2 장치 관리자의 동작 코드
List 2 Op code of Device Manger
DeviceManager(dataStructure, event){
    switch(event)
    {
        case FuseAbnormal, CoilAbnormal,
            ContactAbnormal :
            DisplayError(dataStructure)
        case RelayCounter :
            DisplayCount(dataStructure)
        case ReplacementCounter :
            DB_Maintenance_Update(dataStructure);
            DisplayMaintenance(dataStructure)
    }
}
    
```

### 4.5 사용자 인터페이스

사용자 인터페이스는 서비스 관리자에서 각 서비스에 필요한 정보를 사용자 인터페이스를 통해 관리자가 확인할 수 있는 기능과 관리자가 UI를 통해서 릴레이의 품명, 규격, 설치 일자 등을 데이터베이스에 저장할 때 사용자 인터페이스를 제공한다.

그림 10은 감시 프로그램 모니터링 사용자 인터페이스의 구성이다.

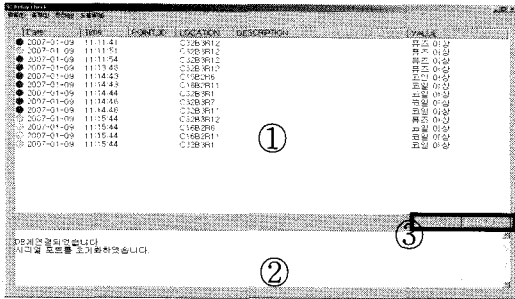


그림 10. 감시 프로그램 모니터링 사용자 인터페이스  
Fig. 10 Monitoring User Interface

사용자 인터페이스 화면의 구성은 다음과 같다. ① 이상 상태를 디스플레이 하는 리스트 뷰 상태, 발생날짜, 시간, 에러발생 Relay ID, 위치, 설명 에러 타입 등을 리스트형태로 화면에 표시한다. ② 프로그램 정보 창 프로그램 구동 시 필요한 정보를 Display해주는 창이다. ③ 사용자에게 의

한 이상상태 Status변경 버튼 사용자가 이상상태를 인지하였나? 에러 복구하였나를 판단하여 입력한다.

모니터링 시스템은 현재 데이터베이스에 등록되어있는 캐비닛과 모듈들의 데이터를 기준으로 동작을 한다. 데이터베이스에 등록되어 있는 데이터라 함은 현재 설치되어 있는 모듈들의 정보를 의미한다. 그러므로 이 시스템을 도입할 때에 몇 개의 캐비닛을 설치하는지, 그리고 캐비닛의 ID, 캐비닛 안에 들어가는 모니터링 보드의 개수와 ID를 등록해 주어야 한다. 이때 등록하는 데이터는 모듈들의 고유ID를 명확히 기재해야 한다. 명확하지 않은 데이터를 입력할 경우 엉뚱한 ID에 계속 데이터를 보내기 때문에 통신에러를 발생하게 된다. 장치등록 프로그램의 구성은 그림 11과 같다. 왼쪽에는 트리구조로 현재 등록되어 있는 장치들을 보여준다. 제어실 안에 2개의 캐비닛이 설치되어 있고 현재 각 캐비닛에는 1개씩의 모니터링 보드가 설치되어 있다.

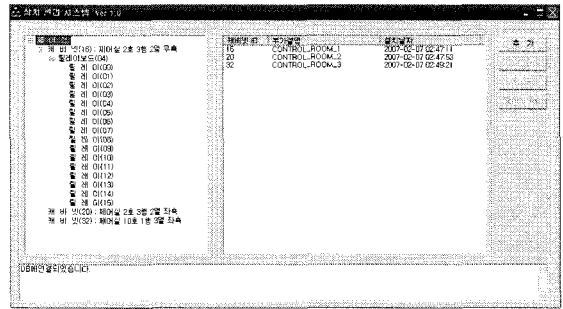


그림 11. 장치 등록 프로그램 화면  
Fig. 11 Device Registering Program

## V. 실험 및 평가

이 장에서는 이 논문에서 설계한 DCS 터미널 원격 감시 시스템 미들웨어의 기능과 성능을 실험 및 평가하기 위해서 원격 감시 미들웨어를 구현하고 구현된 미들웨어가 데이터 수집 및 전송 보드로부터 데이터를 정확하게 수신하여 4.1에서 기술된 기능을 수행하는지 평가하기 위하여 실험을 진행하였다.

프로젝트의 검증 및 결과분석을 위해 전체 시스템 중 하드웨어 부분에선 릴레이보드, 모니터링 보드, 마스터 보드를 각각 2세트 구성하여 상황들을 테스트 하였다. 우선 그림 12는 모든 보드가 통신상으로 연결되며 전원이 인가되어 서로 통신을 시작한 모습이다. 각각 보드들의 아이디는 상단에 위치한 모니터링 보드는 02, 하단은 03이다.

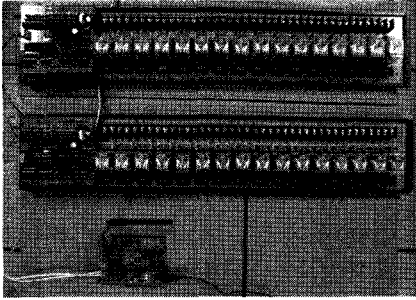


그림 12. 실험 장치 그림  
Fig. 12 Experiment Real Picture

릴레이 에러의 정확한 감지를 확인하기 위해, 코일 에러, 접점 에러, 퓨즈 에러, 동일 위치의 Coil & Fuse Error, 동일 위치의 접점 & Fuse Error, 동시에 발생하는 복합적인 에러 등을 확인하였다.

동시에 발생하는 복합적인 에러의 경우, 상항으로 02번 모니터링 보드 7번, 13번 릴레이, 5번 퓨즈 & 03번 모니터링 보드 1번, 7번, 11번 릴레이 등에 문제가 발생한 상황에서의 실험이다. 이 경우 패킷 형태는 그림 13과 같으며, 감시프로그램은 그림 14와 같다. 캐비닛 내 여러 보드에서 동시에 발생하는 여러 가지 에러도 마스터보드에서 감시프로그램으로 정확히 전달되는 것을 볼 수 있다. 감시프로그램의 디스플레이에서도 각각의 릴레이보드에서 생긴 문제점들을 분석하여 정확히 출력해주는 것을 볼 수 있다. 감시프로그램의 Time부분을 보았을 때 현재 출력된 문제가 동시에 발생했다는 걸 알 수 있다.



그림 13. 복합적인 에러의 패킷  
Fig. 13 Data Packet of Mixed Error Status

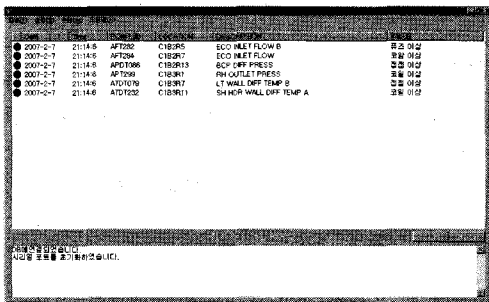


그림 14. 복합적인 에러의 경우 감시 프로그램  
Fig. 14 Monitoring Program of Mixed Error Status

테스트 결과 전송시간 수신시간, 디스플레이 시간은 표 2와 같다. 수신 요청 쿼리를 전송수 1초내로 수신 받고 이를 디스플레이 하는데 까지 수신 요청 전송 후 최대 2초, 평균 1초 이내, 수신 후 최대 1초, 평균 1초 이내의 시간으로 응답이 오는 것을 확인할 수 있다. 최초 제시한 에러 발생후 최대 5초 이내로 확인이라는 제한 조건에 충분히 들어올 수 있음을 확인할 수 있다.

표 2 에러 체크 수행시간표  
Table 2 Error Check Running Time Table

전송		수신		디스플레이	
2007. 2. 6.					
Cabinet:1	20:55:20	Cabinet:1	20:55:20	Cabinet:1, BoardId:2	20:55:20
				Cabinet:1, BoardId:3	20:55:21
Cabinet:1	20:55:24	Cabinet:1	20:55:25	Cabinet:1, BoardId:3	20:55:26
				Cabinet:1, BoardId:2	20:55:33
Cabinet:1	20:55:33	Cabinet:1	20:55:33	Cabinet:1, BoardId:3	20:55:33
				Cabinet:1, BoardId:3	20:55:36
Cabinet:1	20:55:36	Cabinet:1	20:55:36	Cabinet:1, BoardId:3	20:55:36
				Cabinet:1, BoardId:2	20:55:57
				Cabinet:1, BoardId:3	20:55:57
Cabinet:1	21:0:19	Cabinet:1	21:0:19	Cabinet:1, BoardId:2	21:0:19
				Cabinet:1, BoardId:3	21:0:19
				Cabinet:1, BoardId:2	21:1:29
Cabinet:1	21:1:28	Cabinet:1	21:1:28	Cabinet:1, BoardId:2	21:1:29
				Cabinet:1, BoardId:3	21:1:29
Cabinet:1	21:5:36	Cabinet:1	21:5:36	Cabinet:1, BoardId:2	21:5:36
				Cabinet:1, BoardId:3	21:5:36
Cabinet:1	21:5:38	Cabinet:1	21:5:38	Cabinet:1, BoardId:3	21:5:38
				Cabinet:1, BoardId:2	21:9:46
Cabinet:1	21:9:46	Cabinet:1	21:9:46	Cabinet:1, BoardId:3	21:9:46
				Cabinet:1, BoardId:3	21:11:16
Cabinet:1	21:11:15	Cabinet:1	21:11:15	Cabinet:1, BoardId:3	21:11:16
				Cabinet:1, BoardId:3	21:11:29
Cabinet:1	21:11:28	Cabinet:1	21:11:28	Cabinet:1, BoardId:3	21:11:29
2007. 2. 7.					
				Cabinet:1, BoardId:2	16:37:49
Cabinet:1	16:37:49	Cabinet:1	16:37:49	Cabinet:1, BoardId:3	16:37:49
				Cabinet:1, BoardId:2	16:43:50
Cabinet:1	16:43:49	Cabinet:1	16:43:49	Cabinet:1, BoardId:2	16:43:50
				Cabinet:1, BoardId:3	16:43:51
Cabinet:1	16:43:51	Cabinet:1	16:43:51	Cabinet:1, BoardId:3	16:43:51
				Cabinet:1, BoardId:2	16:51:8
Cabinet:1	16:51:8	Cabinet:1	16:51:8	Cabinet:1, BoardId:2	16:51:8
				Cabinet:1, BoardId:3	16:51:9
				Cabinet:1, BoardId:2	16:51:13
Cabinet:1	16:51:12	Cabinet:1	16:51:13	Cabinet:1, BoardId:3	16:51:14
				Cabinet:1, BoardId:2	16:51:20
Cabinet:1	16:51:20	Cabinet:1	16:51:20	Cabinet:1, BoardId:3	16:51:21
				Cabinet:1, BoardId:2	16:51:35
Cabinet:1	16:51:34	Cabinet:1	16:51:35	Cabinet:1, BoardId:3	16:51:35
				Cabinet:1, BoardId:3	16:51:40
Cabinet:1	16:51:40	Cabinet:1	16:51:40	Cabinet:1, BoardId:3	16:51:41
				Cabinet:1, BoardId:2	17:03:45
Cabinet:1	17:03:45	Cabinet:1	17:03:45	Cabinet:1, BoardId:3	17:03:45
				Cabinet:1, BoardId:3	17:03:49
Cabinet:1	17:03:47	Cabinet:1	17:03:48	Cabinet:1, BoardId:2	17:31:22
				Cabinet:1, BoardId:3	17:31:22
Cabinet:1	17:31:22	Cabinet:1	17:31:22	Cabinet:1, BoardId:3	17:31:22

이 외의 실험은 다음과 같은 것을 진행하였다. 감시프로그램의 디스플레이로 사용자는 문제가 생긴 릴레이의 위치를 정확히 알 것이다. 따라서 사용자는 이 문제를 찾아가 교체, 수리, 정정 등의 문제를 해결을 할 것이다. 이 문제 해결을 한 후 감시프로그램의 원할 한 갱신이 되는지의 실험도 하였다. 릴레이의 에러 감지 이 외에도 각 릴레이마다 접점 횟수를 데이터로 하여 접점횟수검색 및 접점 한도 초과 알람 등의 기능을 가지고 있다. 이러한 카운트 기능의 검증을 하였다. 마스터 보드와 모니터링 보드 사이의 통신 장애에 발생하였을 경우에 감시 프로그램에 통신 장애라는 알람이 뜨는 것을 확인하였다.



## VI. 결론

현재 많은 산업현장에서 릴레이의 고장이 일어나 기계가 동작되지 않는 시간에 대한 경제적 피해를 많이 갖고 있다. 그리고 고장난 릴레이의 위치를 파악하는 것 또한 시간이 오래 걸리는 일이기 때문에 피해는 더욱 크다.

따라서 이 논문에서는 터미널 부분의 중요 신호를 유무선 방식을 통하여 통합적으로 감시, 관리하기 위한 DCS 터미널 원격 감시 미들웨어를 설계하고 구현하였으며 미들웨어의 성능 및 기능을 평가하기 위해 터미널 부분의 릴레이들의 동작 상태 및 퓨즈의 이상 유무를 실시간으로 감시하기 위한 보드를 설계 제작하였다.

모니터링 보드는 릴레이의 상태 데이터를 수집하며, RS-485 통신 기능과 I2C메모리를 이용한 id 설정을 한다. 모니터링 보드의 릴레이 데이터를 수집하는 마스터 보드를 개발하였으며, RS-485 통신 기능, RF통신 기능을 갖고 내부 EEPROM을 활용한 id를 설정한다. 원격 감시 프로그램은 MFC 기반으로 개발되었으며, 사용자에게 즉각적인 예러 발생 알람을 주고, 릴레이 장비의 상태를 정확히 출력해주는 장비 관리 프로그램 구축과 예러 및 접점 횟수의 통괄적인 관리를 위한 데이터베이스 구축을 하였다.

제안된 DCS 터미널 원격 감시 미들웨어는 각 릴레이들의 동작상태 및 릴레이와 퓨즈의 이상 유무를 실시간으로 확인이 가능하며 릴레이의 수명을 고려하여 릴레이의 고장을 사전에 예방함으로써 산업분야의 경제적인 효과가 기대된다.

## 참고문헌

- [1] W. Bolton, Programmable Logic Controllers, Fourth Edition, Newnes, 2006
- [2] Karthik Soundararajan1 and Robert W. Brennan1, Design Patterns for Distributed Control System Benchmarking, IFIP International Federation for Information Processing, Springer Boston, 2006
- [3] Frost & Sullican, World Distributed Control Systems Markets, 2006.
- [4] Kent Wert, DCS vs. PLC.(distributed control systems compared to programmable logic controllers), Putman Media, Inc, 1999
- [5] 한국계측제어협회, 분산 제어 시스템, 계측기술, 제10권 제1호, 한국공업출판사, 2002.

- [6] 권옥현 외, 산업용 필드버스 통신망, 성안당, 2004
- [7] 한국자동화표준시스템연구조합, <http://www.kasas.or.kr>
- [8] 김영수, 하수처리에서 DCS 적용 및 자동운전 사례, <http://www.ableduck.com/>
- [9] 이명진, 대규모 분산 실시간 감시 제어 시스템의 설정 자동화 시스템 설계 및 구현, 석사학위논문, 건국대학교 대학원, 2004.
- [10] 김규철 역, 고급 마이크로프로세서 시스템 설계, 시그마프레스
- [11] Atmel corp., 8051 Microcontrollers Hardware & Instruction Set, <http://www.atmel.com/>
- [12] Atmel corp., ATmega162 (V) revision I, <http://www.atmel.com/>
- [13] Atmel corp., AVR Instruction Set (User Guide), <http://www.atmel.com/>

## 저자 소개



### 백 영 태

- 1989년 2월 : 인하대학교 전자계산학과 (이학사)
- 1993년 2월 : 인하대학교 전자계산공학과 (공학석사)
- 2002년 8월 : 인하대학교 전자계산공학과 (공학박사)
- 1993년 5월 ~ 1998년 2월 : 대상정보기술(주) 정보통신연구소 선임연구원
- 2001년 3월 : 멀티미디어기술사
- 1998년 3월 ~ 현재 : 김포대학 IT학부 부교수 (관심분야) 유비쿼터스 컴퓨팅, e-Learning, 멀티미디어콘텐츠개발



### 이 세 훈

- 1985년 2월 : 인하대학교 전자계산학과 (이학사)
- 1987년 2월 : 인하대학교 전자계산학과 (이학석사)
- 1996년 2월 : 인하대학교 전자계산공학과 (공학박사)
- 1987년 ~ 1990년 : 해병대 장교
- 1991년 ~ 1993년 : 비트컴퓨터 기술연구소 선임연구원
- 1999년 6월 : 멀티미디어기술사
- 2001년 ~ 2002년 : New Jersey Institute of Technology(NJIT), USA, Visiting professor
- 1993년 ~ 현재 : 인하공업전문대학 컴퓨터시스템과 교수 (관심분야) 유비쿼터스 컴퓨팅, 임베디드시스템 S/W, 모바일컴퓨팅, 웹서비스