SOFT LOGIC을 이용한 전력설비 통합제어 시스템구축에 관한 연구

조남빈
한국수자원공사 합천댐관리단

The study on Intergrated SCADA system for Powerplant using Soft logic

CHO,NAM-BIN
KOWACO HAPCHUN HYDRO POWER STATION

Abstract - In this paper ,the Intergrated SCADA is used to computer systems designed to perform the following functions for power plant.
- to collect data from industrial plant devices or transducers
- to process and perform calculations on the collected data
- to present collected and derived data on displays on MMI
- to accept commands entered by human operators and act on them such as sending control commands to plant devices.
This system is characterised by open architectured that is based on the international -ly recognized industrial standard for industrial automation control language, the IEC 1131-3

1. 서 론

본 연구에서는 IEC표준에 준거한 소프트로직Program을 Embedded Windows-NT에 적용한 실적을 중심으로 종래의 제작사별 제어시스템(PLC,DCS,DDC ....)들이 갖는 특수성에서 오는 제작사 의존도에서 탈피, 초기투자 및 시스템 확장,개선등에 따른 비용을 대폭적으로 절감할 수 있는 다목적댐 수문 및 발전설비를 대상으로 한 통합 자동화시스템 구축사례에 대하여 수행하였다.
본 논문에서는 시스템 설계에 적용한 OPEN-PLC의 특성 및 소프트웨어의 구성 과 적용 운영체계에 대한 HARD REALTIME에 관하여 비교 분석하고 실제의 플랜트에 적용된 시스템을 중심으로 하드웨어의 구성 및 주요기능에 관하여 적용 사례를 제시코자 한다.

2. 본 론

2.1 SOFT LOGIC의 개요

IEC 61131-3표준에 준거한 소프트로직 프로그램은 PC 기반의 소프트웨어로서 PLC(Programmable Logic Controller)와 같은 제어 시스템의 양상을 뒤바꿔 놓을 새로운 개념의 제어용 제품으로 평가 받고 있다. SoftLogic은 표준 제조공정의 구현을 지원하기 위하여 개인용 또는 산업용 컴퓨터에 적용할 수 있는 개방형 구조를 필수로 하고 있다.

2.1.1 OPEN-PLC 개발 배경

종래의 PLC는 1960년대에 General Motors에서 relay를 대체키 위한 새로운 기술 연구 결과 Dick Morley가 이끄는 팀에서 PLC를 개발하게 되었으며, Modicon이 최초의 PLC회사로 등장하게 되었다. 이후 90년대까지 PLC시장은 눈부신 발전을 하여, Allen-Bradley, Siemens/Texas Instrument, Mitsubishi, General Electric, Schneider 그룹 (Modicon, Square D, Telemecanique)등의 제조회사들이 크게 성장하였다. 그러나, 80년대부터 서서히 변화가 시작되어, 1984년 PC 기반용 SCADA 및 MMI용 팩키지가 발표되면서 PLC를 기본으로 한 시스템에서 PC를 기본으로 하는 시스템을 보다 필요로 하게 되었다. 1986년에 최초로 SoftLogic 제품으로 86-Ladders라는, Allen-Bradley PLC-2를 위한 제품을 발표 했지만, 80년대 말에 크게 호응을 받지 못했다. 그러나, 차세대의 SoftLogic은 "PLC open" 위원회에서 제정한 규격이 IEC 1131-3 규격으로 확정되면서, 미국을 비롯한 전 세계의 주요 메이커들이 이 규격과 제품에 관심을 갖기 시작 하였다. 1994년 미국 자동화 메이커의 3대 기업인 General Motors, Ford 및 Chrysler등이 그들의 OMAC(Open Modular Architecture Controller) 자료를 공개함으로써 현재의 SoftLogic의 기반을 마련하게 되었다. 이러한 의미에서 1995년은 SoftLogic의 원년으로 봐야 할 것이다. 과거에는 소규모의 전문회사에서 SoftLogic을 다뤄왔지만, 이제는 대기업과 기존의 주요 PLC제조업체가 Open PLC시장에 적극 참여하는 추세로 급성장의 기미를 보이고 있다.
이와함께 1996년에 마이크로소프트사의 NT 4.0이 발표되면서, 빠른 속도의 확고한 실시간 제어가 가능하게 되었으며, 컴퓨터도 펜티엄과 펜티엄프로가 발표되면서 처리속도가 상당히 빨라졌지만, 이와 반대로 컴퓨터의 가격은 예전에 비해서 상대적으로 저렴하게 되었다. PLC를 대체할 수 있는 PC 기반의 SoftLogic은 이와 같은 환경 변화에 힘 입어 크게 성장할 것으로 전문가들은 보고있다.

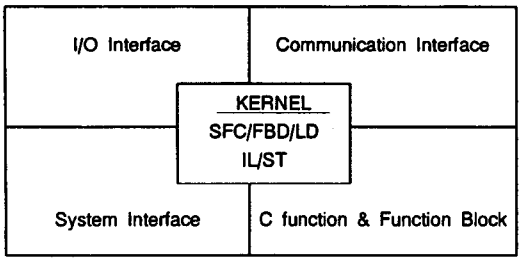
2.2 OPEN-PLC의 구성 및 특성

2.2.1 OPEN-PLC의 개방성 및 적용범위

기존의 PLC가 각각의 독특한 방식으로 프로그램하는 양태와 비교할 때 Open PLC는 표준화된 IEC 1131-3의 규격으로 프로그램을 함으로써 다양한 기종간의 프로그램에 있어서도 쉽게 구성할 수 있으며, PC 기반의 소프트웨어로서의 개방성을 보장하고 있어서, OEM, Integrator 및 사용자들이 고유의 특성에 맞게 사용할 수 있도록 유연성을 갖추고 있다. 적용 범위로는 간단한 제어에서부터 고성능 제어 그리고 높은 신뢰성을 요구하는 작업까지 다양한 분야에 적용할 수 있으며, 수·화력 발전, 정수처리 뿐만 아니라, 고속 열차의 엔진, 브레이크, 액셀레이터 등을 제어하는 분야등 다양한 적용의 유연성을 갖고 있다.

2.2.2 SOFTWARE 구성

SOFTLOGIC은 자동화 제어를 위한 런타임(Runtime) 엔진과 제어 알고리즘 개발 툴로 구성된 산업 소프트웨어 패키지로서, IEC 1131-3(국제 산업표준으로 인정된 산업자동화 제어 언어 규격)을 따르고 있다.



<그림1.> SOFTLOGIC의 Target Platform

특징으로는 DCS와 PLC에서 사용되는 다섯 가지 언어인 SFC(Sequential Function Chart), FBD(Function Block Diagram), LD(Ladder Diagram), ST(Structured Text), IL(Instruction List), Flow Chart를 이용하여 프로그램 로직을 구성할 수 있으며, 그래픽 언어(SFC, FBD, LD, Flow Chart)로 구성된 프로그램을 실행 할 수 있다. 윈도우상에서 개발하여 대상 TM/TM NODE에 DOWNLOAD한다. 사용자가 만든 C function 또는 C Function Block을 SOFTLOGIC 라이브러리에 등록하여 그래픽 환경으로 디버깅 또는 시뮬레이션이 가능한 강력한 성능과 유연함을 가지고 있다.

2.3 운영체계 구성 및 특징

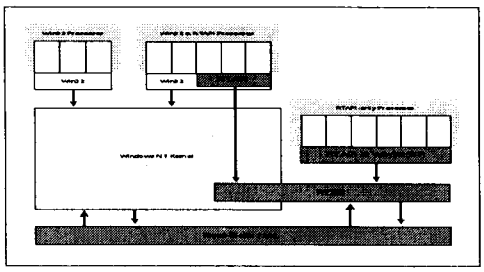
본 시스템 구축에 적용된 Open PLC의 OS PLAT FORM은 다양한 기능이 제공되고 그 신뢰성이 입증된 환경으로 누구나 익혀 사용하고 있는 "Windows-NT"로 EMBEDDED 시스템을 적용 함으로써 기존 IT 인프라를 그대로 사용하면서 기기 들을 제어하는 "Windows NT EMBEDDED OS" 환경을 발전 플랜트의 제어에 적용하였다. NT EMBEDDED 특징으로는 Windows-NT인 경우 VIRTUAL MEMORY를 사용하기 때문에 수행중 PAGING의 SWAPPING이 MEMORY와 HARD DISK사이에 발생하지만 이 SWAPPING을 제거하고 필요한 기능만 선택하여 OS의 PLATFORM을 최소화시키고 MEMORY에서만 수행토록하므로서 PERFORMANCE가 향상되고 설치환경의 먼지와 진동등에 상대적으로 취약한 HARD DISK를 사용하지 않으므로 안정된 시스템을 구축할 수 있다.

2.3.1 HARD REAL-TIME의 기능 및 특성비교

- **REAL-TIME OS**

Windows-NT 특성에 Real-Time 확장을 통한 소프트웨어 개발 툴(Software Development Tool)로, 기존의 Windows NT가 갖고 있는 특성인 호환성/이식성/보안/분산처리/신뢰성/견고성/Localization/확장성에Real-Time을 수정한 기능이다.

Windows NT의 HAL(Hardware Abstraction Layer - 하드웨어 추상 계층)을 RTHAL(Real-Time HAL)로 교체하므로, 고속의 타이머, Win32 프로세스에 우선 처리 및 독립된 인터럽트 처리, Blue Screen Interception을 통해, rtss 프로세스가 지속적으로 수행하여 Windows



NT를 복구(recovery)할 수 있다.

Win32 프로세스는 Soft Real-Time으로 thread를 기준으로 하여 Worst case responsible time이 3 ~ 10 milliseconds으로 처리 시간이 예측 불가능한데 반해, RTSS 프로세스는 Hard Real-Time으로 thread를 기준으로 Worst case responsible time이 40 ~ 100 microseconds으로 Win32 프로세스에 비해 약 100배 빠른 반응 시간을 보인다. 이는 most time critical application 수행을 가능하게 한다.

- **Real-Time HAL**

Real-Time HAL(하드웨어 추상 계층, Hardware Abstraction Layer) 계층은 RTX를 지원하기 위해 만들어 졌다. 이는 RTSS와 Win32 간의 초고속, 결정론적 응답(고속의 클럭과 타이머 서비스, 인터럽트 독립처리)을 위한 리소스를 지원한다. 부가적으로 real-time HAL은 어떠한 Windows NT STOP 이벤트(blue screen)에서도 RTSS 프로세스들이 계속 수행할 수 있도록 Windows NT를 가로채어(intercept), Windows NT가 정상적으로 회복될 때까지 RTSS 프로세스가 수행을 계속한다.

- **Real-TimeSubsystem(RTSS)**

RTSS는 코어 함수와 RTX 리소스 관리를 제공한다. RTSS는 Windows NT 수행부와 통합되어 있으며, 일반적인 API와 Windows NT 객체(Objects) 기술 등의 Windows NT의 기본 특성을 공유한다.

RTSS는 결정론적(deterministic) 수행이 가능하다. 이는 전통적인 Windows NT 환경 서브시스템과 두 가지 기본적인 차이점이 있다. RTSS는 커널 모드에서 Windows NT 디바이스 드라이버가 실행할 때 구현된다. RTSS는 Windows NT 수행부와 구별되는 고정된 우선권 스레드 스케줄러에 의해 수행된다. 이런 특징은 RTSS가 우선적인 low-latency 간섭 서비스에 필요한 Real- Time HAL의 타이머와 간섭 관리 서비스를 사용할 수 있게 한다. 이것은 디바이스 인터럽트에 가장 빠른 응답을 가능하게 하는 RTSS 스레드 스케줄러를 제공하고, RTSS 스레드의 가장 느린 latency 스케줄링을 가능하게 한다.

RTSS는 NT 디바이스 드라이버로서 RTSS는 보호(locked)되고 인접 메모리 할당을 위한 Windows NT 수행부를 통해서 하위 레벨 메모리 관리를 가능하게 할 수 있다.

또 다른 RTSS의 기능은 Win32 서브시스템(이는 본래의 Windows NT 환경 서브시스템이다)과의 일반적인 API, 리얼타임 객체를 공유함으로써, RTXX 서브시스템과 Win32 서브시스템 간의 호환과, 상호사용을 가능하게 한다. 즉, RTSS와 Win32 환경은 events, timers, shared memory, mutexes, semaphores 같은 리얼타임 객체(objects) 공유를 통해 RTSS/Win32 서브시스템 사이에서 통신과 동기화를 허락한다.

- **Real-Time Application Programming Interface (RTAPI) & Win32(RT)**

RTAPI는 디바이스 제어를 위해 Win32 프로그래밍을 확장했다. Win32(RT)와 통합된RTAPI (Win32 함수의 부분)는 RTSS를 위한 API를 제공한다. RTAPI는 RTSS의 필수부분과 Win32 환경에서 호출할 수 있는 DLL로 구현된다. 따라서 어플리케이션은 RTAPI를 사용할 수 있게 개발될 수 있고 어느 한쪽의 환경에서도 수행될 수 있다.

- **Win32(RT)**

Process and Thread Management생성, 우선순위, 제어, Profiling과 RTSS 스레드를 종료하기 위해 Win32 호환 인터페이스를 제공한다.

- Fixed Priority Scheduling

RTSS 스케줄러는 선취 스케줄링과 우선권 반전(inversion)에 대한 관리와 RTSS 객체(objects)의 동기화에 기초하여 우선권을 제공한다. RTSS 스케줄러

는 Win32 우선 순위 확장을 통해 RTSS 프로세스의 실행순서를 보장하는 non- degrading 128개의 고정된 우선권을 제공하기위해 Win32 우선권을 사용할 수 있게 한다.

- Interprocess Communication and Synchronization

RTSS는 RTSS와 Win32 프로세스 간의 공유를 위해 event, semaphore, mutex, shared memory 객체(objects)를 제공한다. 부가적으로 RTSS는 intraprocess 동기를 위해 임계 영역(Critical Section)을 통한 Win 32 호환 인터페이스를 제공한다.

- Memory Management

RTSS는 힙 관리를 위한 Win32 호환 인터페이스을 제공한다.

- **RTAPI**

Memory Management

RTSS 메모리 관리는 하나의 프로세스를 인접된 메모리와 보호된(locked) 메모리에 할당한다. 게다가 RTSS 메모리 관리는 프로세스와 Windows NT 커널 메모리를 보호(lock)하기 위해 Win32 프로세스로서 사용될 수도 있다.기본적으로 RTSS 프로세스는 자동으로 메모리에서 보호된다. 그러나 Windows NT 커널의 일부와 모든 Windows NT 프로세스는 일반적으로 메모리에서 보호되지 않지만 요청에 따라 디스크로부터 페이지된다. 메모리 보호(locking) 기능들은 Win32 프로세스를 위해 페이징 지연(Paging Latency)을 제거할 것이다.

High Speed Clocksand Timers

Real-Time HAL 서비스를 통한 RTSS는 1 Microsecond의 분석(resolution) 속도를 갖는 클럭과 100 Microsecond의 분석 속도를 갖는 타이머를 제공한다. 이러한 객체들은 정밀한 시간 분석 측정, 예측 시간과 인터벌(intervals) 태스크를 스케줄할 수 있다.

Port I/O

RTSS 포트 입/출력 프로그래밍 인터페이스는 리얼타임으로 프로세서의 I/O 공간에서 데이터 이동을 가능하게 한다. 이것은 액세스 되어야 하는 모든 디바이스를 위한 디바이스 드라이버 작성 필요를 없애주고, 디바이스 액세스를 위한 드라이버 서비스 요구와 접목(associate)되는 지연(latencies)시간을 없애준다.

Physical Memory Mapping

RTSS 물리 메모리 사상은 프로세스가 가상 메모리 주소를 통해 물리 메모리를 액세스하는 기능이다.물리 메모리 사상(mapping)은 리얼타임 프로세스가 물리 메모리 주소 영역, 컨트롤러의 메모리와 다른 디바이스 액세스를 허용한다.

Interrupt Management

RTSS 인터럽트 관리 기능은 리얼타임 프로세스가 인터럽트에 인터럽트 핸들링 스레드를 덧붙이게 하고, 인터럽트 처리를 가능하게 하거나 불가능하게 할 수 있다.이는 핸들러를 인터럽트하기 위한 완벽한 스레드 컨텍스트(context)를 제공한다.포트 입/출력과 물리 메모리 사상의 이러한 기능은 리얼타임 프로세스가 디바이스를 전적으로 제어함으로써 디바이스 드라이버를 쓰기(write) 위한 필요를 없애준다.

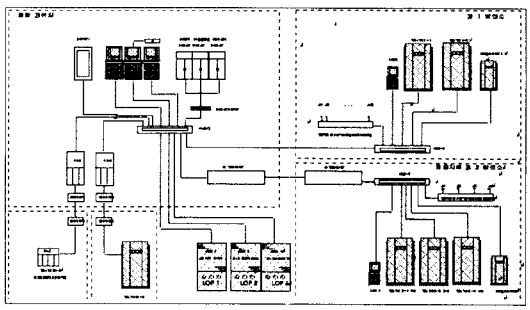
RTSS and Win32 Processes어플리케이션은 RTSS 환경에서 실행하기 위한 타임 크리티컬 기능과 Win32 환경에서 실행하기 위한 non-time critical 기능을 파티션으로 만들 수 있다. 두 환경의 프로세스는 공유된 RTSS 프로세스 간 통신(Interprocess Communication-IPC) 객체를 통해 통신하고 동기화할 수 있다. RTSS환경의 프로세스는 어플리케이션의 빠른 속도로 결정론적인 제어 기능을 수행할 수 있는 반면에, Win32 환경의 프로세스는 그래픽 디스플레이와 네트웍 통신 데이터 저장 같은 Win32 기능의 모든 기능을 사용할 수 있다.리얼타임 프로세스의 특별한 경우는 리얼타임 디바이스 드라이버이다.

Real-Time Device Drivers

RTX는 디바이스 드라이버 작성을 위한 모델을 제공한다. 리얼타임 디바이스 드라이버는 리얼타임 프로세스로 같은 API를 사용하고, 같은 컨텍스트에서 수행한다. 이것은 인터럽트 핸들러의 개발을 간단하게 한다. 인터럽트 핸들러는 모든 스레드 컨텍스트(Context)에서 수행하고 부동 소수점(floating point)을 포함한 모든 API를 사용할 수 있다.인터럽트 핸들러가 스레드 컨텍스트(Context)에서 수행한 후, 스레드 우선순위는 수행될 순위를 결정한다.높은 우선순위의 어플리케이션 스레드는 선점할 수 있고, 낮은 순위의 인터럽트 핸들러 보다 먼저 수행한다. 디바이스 드라이버를 위한 스레드 모델의 사용은 어플리케이션과 디바이스 드라이버 간의 우선순위에 의한 풍부한 융통성을 제공하고, 이들 간의 연결을 위한 메커니즘을 제공한다.

3. 결 론

이상에서 검토한 내용과 같이 다목적댐 수문 및 발전설비에 대한 통합원격감시제어장치의 구축은 그림3.과 같이 설계되어 구축중에 있으며, 본 시스템의 제어대상설비 및 장치의 구성은 다음과 같다.



〈그림3.〉다목적댐 수문.발전설비 원격제어시스템구성도

-제어대상 설비

- Main dam Spillway gate : 5문
- Main dam Intake gate : 1문
- R/R dam Spillway gate : 7문
- R/R dam Intake gate : 2문
- Main dam W/T generator : 2대
- R/R dam W/T generator : 2대

-장치의 구성

- Server computer(cpu dual) : 1set
- TM/TC(Open PLC) : 6set
- LCS(LocalControlStation) : 2set
- GDP&LCP : 각3set
- DIAGNOSIS : 2set
- Digital WHM : 14set
- S/W HUB : 3set
- OPTIC NETWORK : 1식

본 시스템은 개방형설비의 도입과 구성의 단순화로 충분한 신뢰성과 경제성을 확보할 수 있도록 설계하였으며 금번 Project를 통한 성능을 입증하므로써 향후 발.변전설비 및 수처리설비등을 대상으로한 원격감시제어시스템(SCADA)에의 확대 적용이 기대된다.

〔참 고 문 헌〕

〔1〕 한국수자원공사,"현장자료취득을 위한 프로토콜 및 MMI 표준화 연구",1997.12

〔2〕 MICROSOFT사, "WINDOWS-NT EMBEDDED",

〔3〕 WISDOM사, "OPEN PLC",

〔4〕 CJ INTERNATIONAL사," ISAGRAF".

〔5〕 VENTURCOM사, "REAL TIME EXTENTION",