

그래픽 제어 언어를 사용한 공정제어용 다중루프 제어기의 개발

이 회 규 분 봉 채 김 병 국 변 중 남
한국과학기술원 전기 및 전자공학과

Development of Multiloop Programmable Controller Using Graphic Control Language

Heegyoo Lee Bong Chae Moon Byung Kook Kim Zeungnam Bien
Dept. of Electrical Engineering
Korea Advanced Institute of Science and Technology

ABSTRACT

A multiloop programmable controller using graphic control language and a configuration graphic editor for designing control algorithm in graphic windows are developed. Using the graphic control language, large and complex control algorithm can be designed easily and the modification of control algorithm is simple. The proposed controller using graphic language can be effectively used for complex plants like power plants.

1. 서론

1950년대 중반 공정 제어를 위하여 디지털 컴퓨터가 사용되기 시작한 이래, 컴퓨터 기술의 발달은 산업공정 제어에 매우 큰 영향을 미치게 되었다. 이에 따라 1980년대에 들어 와서는 발전소 등과 같은 대규모 공정제어 분야에도 디지털 제장제어 시스템이 도입되고 있다. 디지털 제장제어 시스템은 제어 프로그램의 변경으로 제어 알고리즘을 쉽게 바꿀수 있으나 대규모 공정을 제어하기 위한 제어 알고리즘은 매우 복잡해 지므로 제어기의 구성을 쉽게 파악하고 변경하기 곤란하다.

본 연구에서는 대규모 산업공정인 발전소를 대상으로 하여 확장성 있는 프로그래밍 다중루프 제어기 (Multi Loop Programmable Controller, MLPC)와 그래픽 에디터를 이용하여 제어 알고리즘을 쉽게 구현하도록 하는 제어기 구성기능(Configuration)을 구현하였다.

제어기 구성기능은 사용자와 디지털 제어기를 연결하여 주는 기능을 담당하여 그래픽 에디터를 통하여 사용자가 원하는 디지털 제어기를 구성할 수 있도록 하는 기능으로

사용자가 전문적인 지식 없이도 쉽게 제어기를 구성할 수 있도록 한다. 제어기 구성 기능은 대화 형식으로 그래픽 화면상에서 제어기를 구성하는 그래픽 에디터와 기능 코드의 각종 파라미터를 구축하는 코드 라이브러리 그리고 그래픽 에디터의 데이터로부터 기능 코드를 만들어 내는 기능 코드 발생기로 구성된다.

제어기의 제어 알고리즘은 기능 코드의 나열로 구성되며 기능 코드는 제어기에 필요한 기본 기능을 제공한다. 기능코드 발생기는 임의로 만들어진 그래픽 데이터를 제어기가 제어기능을 수행할 수 있도록 신호의 입출력 관계에 의하여 정리하는 기능을 갖고 있다.

그래픽에디터를 통하여 만들어진 제어 알고리즘을 수행하는 디지털 다중루프 제어기(MLPC)는 MC68000 CPU를 갖는 CPU module(MLC), SRAM module, 신호 입출력을 담당하는 I/O module(SIB/STB) 그리고 local panel 기능을 담당하는 DCS와 manual operation을 위한 AMS등으로 구성되며, 고 신뢰도를 위하여 CPU module과 I/O module은 이중화 구조로 구성하였다. 제어 알고리즘은 통신회선을 통하여 SRAM에 저장되고 MLC는 이를 해석하여 제어 기능을 수행하게 된다.

다중루프제어기의 소프트웨어는 시스템 초기화 부분, 기능코드 변환기 그리고 기능코드 해석기로 구성되어 Configuration에서 만들어진 기능 코드를 해석하여 제어기능을 수행하며 수행 도중에도 제어 내용을 쉽게 변경할 수 있도록 하기 위하여 인터프리터(Interpreter) 구조로 설계되었다.

Configuration에서 만들어진 제어 알고리즘은 MLC에 다운로드되고 나면 기능 코드 변환기에 의하여 MLC가 효과적으로 제어기능을 수행할 수 있도록 기능코드를 변환한다. 이 동작은 오프라인으로 수행되며 시작명령에 의해 기능코드해석기에서 각 기능코드를 해석 수행함으로써 제어기능을 수행한다.

2. 제어기 구성 기능

디지털 제어기는 제어 알고리즘이 소프트웨어로 구현되므로 기존의 아날로그(Analog) 방식에 비해 제어 알고리즘의 확장 및 변경이 용이한 특징이 있다. 또한, 마이크로 프로세서 기술의 진보에 따라 하나의 모듈에서 처리할 수 있는 용량이 증가되므로 전체 디지털 제어기의 크기는 작아지면서도 처리용량은 증가된다. 따라서 제어 알고리즘을 효과적으로 설계할 수 있는 방법이 필요하다.

제어기 구성기능(Configuration)은 그래픽 에디터를 통하여 복잡한 제어 알고리즘을 전문적인 지식없이도 쉽게 구성할 수 있는 방법을 제공한다. 즉, 사용자(Operator)는 미리 준비된 기능 블록(Function Code Block)을 선택하고 연결하면 자동적으로 디지털 제어기가 인식할 수 있는 형태로 제어 알고리즘을 구현할 수 있다. 그래픽 에디터에서 구성된 디지털 제어 프로그램은 통신 시스템을 통하여 다중 루프 제어기(MLC)에 다운로드(Download)되어지며 다중 루프 제어기는 시작 명령에 의해 제어 알고리즘을 수행한다.

그래픽 에디터에서 각 기능코드는 이에 대응하는 코드 심볼로 표현되어 사용자가 쉽게 이해할 수 있으며 구성하고자 하는 제어 다이어그램이 커지더라도 제어 알고리즘의 흐름을 쉽게 파악할 수 있으므로 오류의 발견이나 수정이 매우 용이한 장점이 있다. 또한, 신호의 흐름에 따라 자동적으로 다중 루프 제어기가 인식하고 수행할 수 있는 제어 프로그램을 생성하므로 사용자는 쉽게 제어 알고리즘을 설계하고 수정할 수 있는 환경을 제공한다.

제어기 구성 기능은 사용자로부터 입력을 받아 그래픽 화면상에서 제어기를 구성하도록 하는 그래픽 에디터를 중심으로 기능코드 발생기와 구성된 제어기 다이어그램을 레이저 프린터로 프린트 하는 기능등으로 구성된다. 제어기 구성 기능의 전체 구성은 (그림 1)과 같다.

제어기 구성기능은 그래픽 에디터로 대표되는 그래픽 환경을 제공한다. 그래픽 에디터는 SUN 워크스테이션의 GKS (Graphics Kernel System) 윈도우 환경에서 운용되며 주 입력 장치로는 마우스와 키보드가 이용된다. 디스플레이 장치로는 1152x900 픽셀의 해상도를 갖는 컬러 디스플레이 모니터가 사용되고 화면은 Canvas 윈도우, Text 윈도우, Message 윈도우, Command Menu 윈도우 그리고 Code Menu 윈도우 등의 다섯개의 겹쳐지지 않는 윈도우로 구성되어 있다.

그래픽 에디터의 동작은 처음에 초기화 루틴을 수행한 후 사용자가 명령어 메뉴에서 원하는 명령어를 선택하여 Canvas 윈도우에 제어 다이어그램을 작성하게 된다. 그래픽 에디터는 기능적으로 초기화부, 명령 메뉴 입력부 그리고 명령어 처리부 등으로 구성된다. 그래픽 에디터의 초기화면은 (그림 2)와 같다.

명령어 메뉴는 주메뉴와 5개의 부메뉴로 구성되어 있으며 동시에 8개까지의 명령어 메뉴가 제공된다. 사용자가 마우스로 명령어를 선택하면 명령어 처리루틴은 실행 가능한 명령어이면 해당 명령어 처리루틴을 수행하고 다른 메뉴로의 이동 명령이면 변경된 메뉴를 윈도우에 새로 표시하고 사용자의 선택을 기다린다.

그래픽에디터에서의 데이터는 링크드 리스트(Linked list)의 형태로 관리된다. 제어 다이어그램의 크기가 큰 경우에는 여러 페이지로 나누어 다이어그램을 작성할 수 있으며 두 기능코드의 신호를 연결하면 그래픽 화면상에서 자동적으로 선이 연결되며 화면이 변화되면 자동적으로 재구성되어 화면에 표시된다.

3. 기능 코드 발생기능

그래픽 에디터에서 만들어진 데이터는 각 기능 코드 블록의 구성이 사용자가 작성한 순서대로 되어 있으므로 다중 루프 제어기(MLC)가 바로 제어 기능을 수행할 수 없다. 그러므로 그래픽 에디터의 데이터에서 필요한 기능 코드를 만들어내는 과정이 필요하며 이 과정에는 기능 코드 데이터를 다중 루프 제어기(MLC)의 수행 순서에 맞도록 순서를 재배치 하는 부분(Sorting)과 필요한 데이터만을 골라내어 기능 코드로 만들어내는 부분(Filtering)으로 구성된다.

기능코드의 발생 방법은 링 버퍼(Ring Buffer)와 스택(Stack)을 사용하여 각 기능 코드의 입출력에 연결되는 기능 코드와의 선후 관계를 이용하여 전체 기능 코드의 순서를 재배치한다.

먼저 기능 코드 데이터를 검색하여 각 루우프의 최종 코드 블록을 찾아내어 링 버퍼(Ring Buffer)로 구성된 버퍼에 코드를 입력시킨다. 다음에는 링 버퍼를 순차적으로 검색하여 코드를 스택에 푸쉬(Push)하고 입력에 연결되는 코드를 다시 링 버퍼에 입력한다. 제어 다이어그램에서 한 코드블럭의 출력신호가 여러개의 코드블럭에 연결된 경우에는 스택을 검사하여 같은 블럭이 존재하면 스택상의 블럭을 제거하여 리던던시를 없앤다. 이 과정을 버퍼에 데이터를 없어질 때까지 수행한 후 스택에서 코드를 하나씩 꺼내어 기능 코드의 포맷(Format)에 맞도록 필요한 데이터를 골라내어 기능 코드를 작성한다.

이 과정을 마치면 그래픽 에디터에서 구성된 제어 알고리즘의 제어 프로그램이 만들어 지고 다중루프 제어기는 사용자가 구성한대로 제어기능을 수행하게 된다.

4. 기능코드의 구성

그래픽 에디터에서 사용되는 기능코드 블럭은 각각 다중루프 제어기에서의 기능 코드에 대응된다. 각 기능

코드는 제어기에 필요한 기본 기능(Primitive)을 수행하는 서브루틴으로 볼 수 있으며 MLC는 이러한 서브루틴들을 호출함으로써 제어 알고리즘을 수행하고 제어 입력을 계산해내는 것으로 생각할 수 있다. 기능코드는 아스키코드의 형태로 MLC에 전달되는데 그 구성은 출력 블럭 번호, 아스키 기능 코드 이름, 입력 블럭 번호, 파라미터들의 순서로 나열되며 각 데이터는 공백 문자로 구별된다. 블럭 번호는 그 기능의 결과를 저장하는 변수의 번호로 할당 되고 입력 블럭번호와 파라미터들은 기능코드에 필요하지 않는 경우에는 생략된다.

하나의 기능코드는 하나의 라인으로 구성된다. 두개의 입력을 받아들여 더한 결과를 출력하는 기능코드의 예를 보이면 다음과 같다.

```
1 AIN 14 4
2 AIN 14 2
3 ADD 1 2
4 AOUT 3 14 1
5 RET
```

기능코드를 통한 디지털 제어기의 제어프로그램은 인터프리터방식의 언어를 사용함으로써 하드웨어 또는 제어기 프로그램의 변경없이 단지 기능코드만을 변경함으로써 제어기의 구성을 변경할 수 있다. 또한, 만들어진 제어 프로그램은 아스키 형태로 되어 있으므로 그래픽 에디터를 이용하지 않고 직접 작성할 수도 있다. 기능코드는 현재 약 30여개가 정의되어 있으며 기능코드의 추가 또는 변경이 용이하게 되어있다.

5. 다중 루프 제어기

디지털 공정 제어기는 CPU 모듈인 MLC(Multi Loop Controller), I/O 모듈인 SIB(Signal I/O Board), 신호 조절 기능을 하는 모듈인 STB(Signal Termination Board) 및 사람과의 인터페이스(Operator Interface)를 용이하게 하기 위해 DCS(Digital Control Station)와 AMS(Auto/Manual Station) 모듈들을 포함하여 전체적인 공정 제어 서비스 시스템을 이루게 된다. 이들 상호간의 관계를 나타내는 블럭 다이어그램을 (그림 3)에 나타내었다.

공정제어를 위해 사용되는 제어 알고리즘을 내장하고 있는 MLC는 MC68000 CPU를 갖는 단일 보드 컴퓨터(Signal Board Computer)로서 내장된 제어 알고리즘으로 제어 입력을 계산하여 그 결과를 공정 접속 모듈로 보내준다. 다중 루우프 제어기의 소프트웨어는 시스템을 초기화하는 부분, 기능코드를 변환하는 부분, 그리고 기능 코드를 해석하는 부분으로 나눌 수 있다. 디지털 제어기의 전체 소프트웨어의 구조는 (그림 4)와 같다.

COM으로부터 MLC로 다운로드되는 기능 코드는 아스키 문자로 이루어진 코드로서 사용자가 직접 눈으로 확인하기에는 편리하나 온라인 중에 이를 해석하고 수행하기에는 효과적이지 못하다. 그래픽 에디터를 이용하여 만들어진 제어 프로그램이 SRM에 다운로드(Download)되면 기능코드 변환기는 해석기가 효과적으로 명령을 수행할 수 있도록 기능코드를 토큰의 형태로 변환하여 구조화된 데이터를 SRM상에 재구성한다. 즉 MLC는 다운로드된 기능코드 각각에 구조화된 데이터 메모리 맵(Memory Map)을 만들어 전체 제어 알고리즘에 대한 데이터 베이스를 형성한다.

기능 코드 변환이 끝나면 시작 명령에 의해 타이머 인터럽트를 인에이블하며 MLC는 자체진단을 하게 된다. 기능코드 해석기는 타이머 인터럽트 서비스 루틴에서 호출되어 기능 코드 명령을 해석하고 수행하며 크게 세부분으로 나누어진다.

첫번째 부분은 데이터 입력 부분으로 공정으로부터의 정보가 SIB를 거쳐 MLC로 들어오는데 MLC는 이 정보를 SRM의 입력 신호 테이블의 지정된 위치에 저장하는 일을 수행한다.

두번째 부분은 제어 입력을 계산하는 부분 즉 기능 코드 해석기이다. 기능 코드 변환기가 만들어 놓은 데이터 베이스의 첫번째 구조화된 데이터로부터 마지막 RET에 해당하는 구조화된 데이터까지를 차례로 증가하면서 데이터 입력부분에서 해놓은 정보를 이용하여 구조화된 데이터 배열순서에 따라 각각의 출력을 계산해 내고 그 출력을 연결된 다른 기능코드가 입력으로 사용하여 출력을 만들어 내는 방식으로 'AOUT'의 기능코드에서 최종의 제어 입력을 만들어 내게 된다. 이때 만들어진 제어입력은 출력 신호 테이블의 정해진 위치에 저장된다. 마지막 기능 코드까지 수행되면 기능 코드 해석기의 루틴을 빠져 나온다.

마지막 부분은 제어 입력의 출력 부분으로 출력 신호 테이블상의 데이터를 SIB를 거쳐 공정으로 내주는 역할을 한다.

6. 결론

본 연구에서는 그래픽 에디터를 통한 제어알고리즘 구성기능과 기능코드를 이용하는 프로그램형 제어언어를 제안하고 이를 발전소를 대상으로 한 디지털 계장제어 시스템에 구현하여 타당성을 확인하였다. 제어 프로그램의 변경으로 제어 알고리즘을 쉽게 변경 할 수 있고 그래픽 에디터의 이용으로 제어 내용을 쉽게 확인하고 수정할 수 있는 유연한 구조를 갖음으로서 발전소등과 같은 대규모 공정을 제어하는데 효과적으로 적용될 수 있음을 보였다.

7. 참고문헌

- [1] Herbert Schildt, *Advanced C*, McGraw-Hill, 1986.
- [2] 정현규, "제어언어를 이용한 다중 루프 프로그램형 제어기에 관한 연구", 한국과학기술원, 석사논문, 1988.
- [3] Sun Microsystems, *Sun System Overview*, 1986.
- [4] Sun Microsystems, *Sun GKS Reference Manual*, 1986.
- [5] Bailey Controls, *Bailey Network 90 - Function Code Reference Manual*.
- [6] 한전 기술연구원, 발전소 제어용 디지털 계장제어 시스템 개발 (최종 연구 보고서), 1990.
- [7] 이창구, 김성중, "컴퓨터 공정제어", 전기학회지, 제35권, 제12호, 1986.
- [8] 변중남, 김병국, 조영조, "전력플랜트에 적용되는 다중루프 제어시스템의 구성", 전기학회지, 제38권, 제3호, 1989.

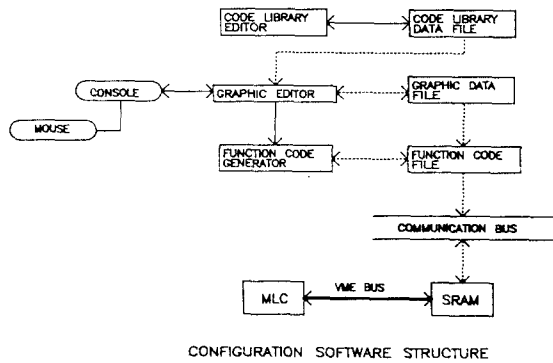


그림 1. 전체 구성
Fig. 1. The Overall Structure

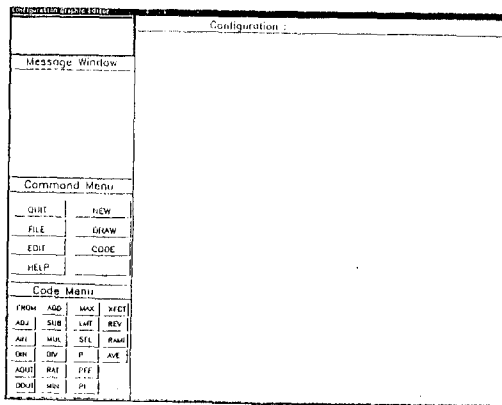


그림 2. 그래픽 에디터의 초기화면
Fig. 2. Graphic Editor Screen at Start

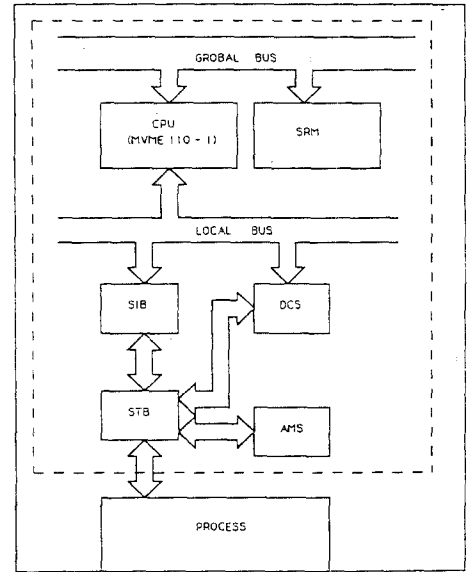


그림 3. 다중루프 제어기의 하드웨어 구성

Fig. 3. The Hardware Structure of the Multiloop Controller

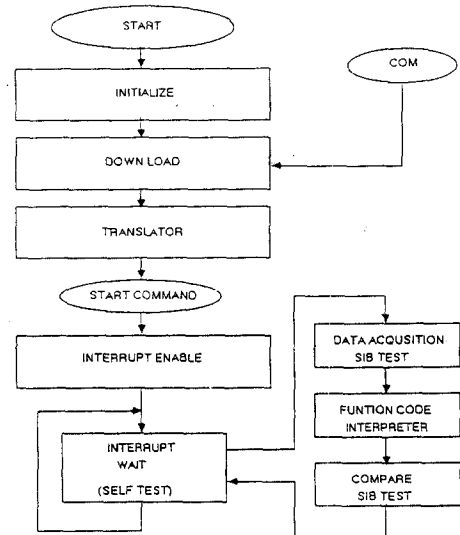


그림 4. 다중루프 제어기의 소프트웨어 구성

Fig. 4. The Software Structure of the Multiloop Controller