

---

# 분산 제어 시스템용 기능 블록 작성기 구현

권만준\*

## Implementation of the Function Block Builder for the Distributed Control System

Mann-Jun Kwon\*

### 요 약

다양한 공정 분야-발전 플랜트, 수처리 플랜트, 소각로 플랜트, 화학 플랜트, 시멘트 플랜트-에서 적용되는 제어 프로그램은 많은 종류가 있다. 이러한 제어 프로그램을 작성하기 위해서는 기존의 텍스트 기반의 제어 언어로는 현장의 엔지니어가 프로그램을 작성, 수정, 테스트 및 디버깅하는데 있어 어려움이 많다. 따라서 본 연구에서는 이러한 텍스트 기반의 단점을 해결하고, 다양한 플랜트에 적용이 가능하며 제어에 대한 일반적인 지식이 있는 엔지니어라면 누구나 쉽게 사용할 수 있는 제어 프로그램의 블록화 도형으로 표현할 수 있는 도구를 구현하였다. 구현된 기능 블록 작성기를 이용하면 제어의 흐름에 대한 시각적 표현과 제어의 중간 출력값 등을 볼 수 있어 보다 정밀한 공정 제어를 효율적으로 수행할 수 있다.

### ABSTRACT

There are so many kind of a control program that is applied in various process fields such as power generation plant, water treatment plant, incinerator plant, chemical plant, cement plant etc.. Because an engineer in field edits and changes and debugs and tests properly control programs using text-based control language, it is very hard for the him to apply to plant. Therefore, this research implemented a graphical tool for control program builder that is applicable to various plants and usable engineers having a little knowledge for control language. I wish to run more efficiently precision process control offering function that can see visual expression about flow of control signal and intermediate output values of control program displayed in screen using this implemented function block builder.

### 키워드

분산제어시스템, 기능 블록, 기능 코드, PID 제어기

## 1. 서 론

공정 산업 분야의 제어기 적용 형태는 1950년대에는 공기식 제어기(Pneumatic Controller)가, 1960년대에는 아날로그 제어기(Analog Controller)가 주류를 형성하였다. 1970년대 마이크로프로세서 출현이후로 단일 루프 제어기(Single Loop Controller)들을 결합한 디지털 분

산제어 시스템의 초기 형태가 적용되었다. 1980년대 들어서 16비트, 32비트 등의 고성능 마이크로프로세서를 이용한 다기능 제어기(Multi-Function Controller)로 구성된 분산 제어 시스템(DCS, Distributed Control System)이 적용되어 기존의 많은 제어 시스템을 교체시켰다. 그리고 보다 고기능과 고신뢰성을 추구하면서 하드웨어 발달과 소프트웨어 분야의 새로운 기술을 도입하여 자동화 단계를 보다 고도화하고

---

\* 대전대학 컴퓨터전자전기학부

비용 절감 및 품질 향상을 피하고자 끊임없는 노력이 기울여지고 있다[1].

분산 제어 시스템이라 통칭하는 대부분의 시스템들은 계층적 구조와 분산 구조를 혼합한 형태를 취하며, 그림 1과 같이 크게 3계층으로 이루어진다[2].

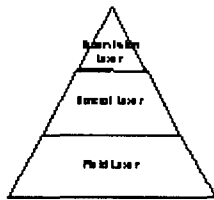


그림 1. DCS의 계층적 구조  
Fig. 1 Hierarchical structure of DCS

최하위의 Field Layer는 센서 또는 I/O 입출력 장치 그리고 Field Bus등으로 구성되며, 두번째 계층인 Control Layer는 하위의 Field Layer의 장비에 대해서 공정 제어 유닛으로 구분하여 제어 기능을 담당하는 곳으로 데이터베이스의 기초가 되며, 최상위의 Supervision Layer는 사용자와의 인터페이스를 담당하며 하위 계층의 데이터베이스의 구성 및 관리, 공정 감시의 알람 처리를 담당한다

이러한 분산 제어 시스템은 마이크로프로세서 기술과 네트워크 기술의 발달을 근간으로 발전되어 왔으며, 1979년의 태동기에는 공정제어 설비의 선도 업체들인 Bailey, Honeywell, Yokogawa, Siemens, BBC, Foxboro를 중심으로 한 몇몇 업체에서 판매를 시작하였으나, 현재는 30여개가 넘는 업체들이 경쟁을 벌이고 있으며, SCADA(Supervisory Control and Data Acquisition) 시스템 등과 같은 유사한 부류의 업체를 포함할 경우 수 백여개에 이른다.

분산 제어 시스템을 적용한 공정에서 작업을 하는 엔지니어 입장에서는 제어 하고자 하는 대상에 대한 특성도 이해해야 하지만 제어 시스템 자체의 기능을 이해하는 일만 해도 상당한 어려움이 있다. 엔지니어링 작업 중에서 데이터 베이스를 구성하고 그래픽 작화 작업을 하며 트렌드 그룹 등 다양한 모니터링 기능을 위한 엔지니어

링 작업을 해야 하지만 무엇보다도 중요한 제어 프로그램 작성에 가장 많은 어려움을 겪고 있다. 대부분의 공정제어 설비를 만드는 개발 회사에 따라서 전용의 제어 언어를 사용하고 있으며 이러한 언어를 숙달하기 위해서는 여러 해에 걸쳐서 경험을 축적해야 한다.

그리고 기존의 제어 언어는 대부분 VISUAL BASIC이나 C언어와 같이 텍스트 기반의 언어로 이루어져 있기 때문에 해당 제어 언어에 익숙하지 않으면 접근이 불가능하게 되어 있었다. 그리고 각 언어 마다 까다로운 문법을 요구하고 있기 때문에 엔지니어 입장에서는 많은 시행착오를 겪기도 한다.

그래서, 본 연구에서는 제어의 기본 개념을 이해하고 있는 엔지니어에게도 충분히 이용이 가능한 기능블록 작성기를 구현하였다. 구현된 기능 블록 작성기는 엔지니어 입장에서 시스템에 관련된 몇 가지 정보-시스템에 관련된 한계 정보-와 기능 블록이라는 개념만 이해 한다면 다양한 제어 프로그램을 작성할 수 있도록 되어 있다. 기능 블록을 그래픽적으로 표시하였기 때문에 개념적으로 이해하기가 쉬우며 기능 블록의 상호 연결에 의해서 제어 프로그램이 작성되기 때문에 엔지니어 입장에서는 별 어려움 없이 쉽게 배울 수 있는 작성기가 된다.

본 연구에서 개발된 작성기를 실제 공정 운전 단계에서도 이용할 수 있기 때문에 제어의 흐름에 대한 시각적 표현과 제어의 중간 출력 값 등을 볼 수 있는 기능을 제공하여 보다 정밀한 공정 제어를 효율적으로 이룩할 수 있다.

이하에서는 이러한 기능 블록 작성기 구현의 상세적인 내용을 기술하고자 한다

## II. 본 론

### II-1. 기능 블록의 개념

본 연구에서 구현한 기능 블록 작성기에서 가장 중요한 개념은 기능 블록(Function Block)의 개념이다. 모든 제어 알고리즘을 기술하는 연산자에 대해서 그림 2와 같이 일반화된 기능 블록으로 표시를 할 수 있다.

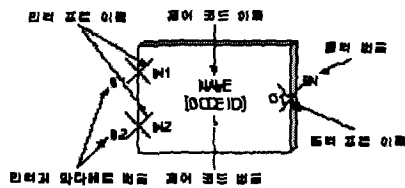


그림 2. 기능 블록의 일반화 그림  
Fig. 2 Generalized picture of the function block

기능 블록은 블록 번호(Block Address 또는 Block Number), 제어 코드(Function Code), 파라미터(Parameters 또는 Specifications)에 의해서 결정된다. 블록 번호는 기능 블록이 위치한 상대적인 번지를 나타내는 것으로 실제 운전 될 때는 블록 번호가 낮은부터 차례대로 실행이 된다. 제어 코드는 각 기능 블록마다 고유한 제어 코드 값을 가지며 이 값은 특정한 연산 기능을 담당하는 ID값과 동일하다. 그리고, 파라미터는 기능 블록의 상세적인 운전 특성을 결정할 수 있는 데이터들이 포함된 공간을 의미한다. 입력과 출력 포트에는 고유한 이름을 부여하여 사용자에게 인식이 쉽게 하도록 하였다. 그리고, 사용자에게 입출력에 대한 명확한 이름을 부여하여 제어 프로그램 작성에 도움을 주었다.

예를들어, 기능 블록을 이용하여 일반적으로 "8 + 9 = 17"라는 수식에서 '8', '9', '+'모두를 제어 코드화 시킬 수가 있다.

일반수식 :  $8 + 9 = 17$

즉, 이와 같은 수식을 기능 블록으로 표시하면 다음과 같다.

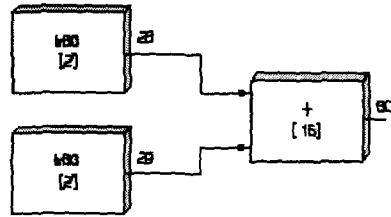


그림 3. 제어 블록 표시 예제(오프라인)  
Fig. 3 Example of the function block(Offline)

그림 3에서 MSC(Manual Set Constant)라는 기능 블록은 사용자에게 의해서 입력한 상수값을 가지는 기능 블록이다. 이 기능 블록에서 [2]는 제어 코드의 ID값이 2임을 나타낸다. 제어 코드 15번은 '2-INPUT SUMMER'(덧셈기)를 나타낸다. 실제 연산에 필요한 '8'과 '9'의 값은 각 MSC의 파라미터 값에 설정이 된다.

이러한 제어 프로그램을 작성하여 시스템에 적용하여 실행하면 그림 4와 같이 출력값이 함께 화면에 표시 된다.

그리고, 각 기능 블록마다 28, 29, 30는 블록 번호를 나타내며 28번째 블록의 MSC가 먼저 실행되고 29번째의 기능 블록이 다음에 실행되며 최종적으로 더하기 기능 블록이 실행된다.

여기서 공정 제어 분야-수처리, 발전, 보일러,

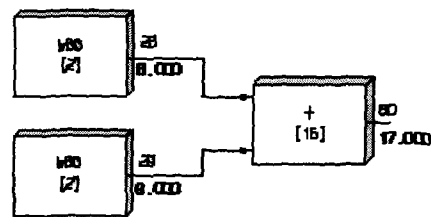


그림 4. 제어 블록 표시 예(온라인)  
Fig. 4 Example of function block(Online)

제철, 석유화학-에서 가장 많이 사용되는 기능 블록인 PID제어기에 대해서 상세적으로 고찰해 보겠다.

비례(Proportional, P), 적분(Integral, I), 미분(Derivative, D) 제어기의 특성에 대해서 논할 때 그림 5와 같은 루프 제어 시스템을 생각 할 수 있다[3].

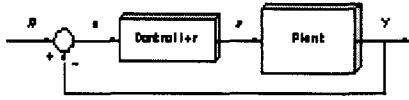


그림 5. 단일 루프 제어기 시스템 구성도  
Fig. 5 System configuration of PID single loop controller

이러한 단일 루프 제어 시스템에서 플랜트에 가해지는 출력값을 다음과 같은 수식으로 나타낼 수 있다.

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (1)$$

$K_p$ : Proportional Gain

$K_i$ : Integral Gain

$K_d$ : Derivative Gain

식 1. 단일 루프 PID제어기의 출력

이러한 PID제어기에 대해서 기능 블록으로 표현하면 그림 6과 같이 된다.

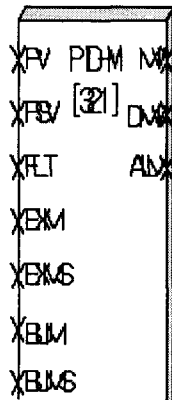


그림 6. PID제어기 기능 블록도  
Fig. 6 PID Function Block

그림 6과 같이 PID모듈의 출력은 MV(Manipulated Value), DMV(Delta Manipulated Value), ALM(Alarm State)의 세 개를 가진다. 입력으로는PV(Process Value), RSV(Remote Process Value), FLT(Fault command), EXM(EXternal MV), EXMS(External MV Selection), BUM(BackUp

MV), BUMS(BackUp MV Selection), PVH-L(PV High Alarm Lock), PVL-L(PV Low Alarm Lock), DVH-L(DV High Alarm Lock), DVL-L(DV Low Alarm Lock), DOMV(Delta Old MV), OMV(Old MV)을 가진다. 그리고 내부적으로 입력 데이터를 포함하여 약 50여개의 파라메타 값으로 구성이 되어있다.

가장 단순한 PID제어를 하고자 하는 경우는 PID모듈의 입력 PV와 RSV를 연결하고 출력의 MV의 값을 플랜트의 입력으로 사용하는 것이 될 수 있다.

위와 같이 제어 시스템에서 사용하는 수식이나 기능에 대해서 기능적으로 구분할 수 있는 최소 단위의 크기(+, -, \*, /)에서부터 여러 개의 기능을 모아서 하나의 큰 기능을 수행할 수 있는 모듈 개념(PID, ALARM, MOTOR 모듈)의 기능 블록을 도입하여 사용자에게 쉽게 인식이 가능한 그래픽으로 표현한 기능 블록을 이용하여 제어 프로그램을 작성할 수 있을 것이다.

## 11-2. 시스템 구성

본 연구에서 구현된 제어 프로그램 작성기는 그림 7와 같은 시스템 구성에서 적용되었다.

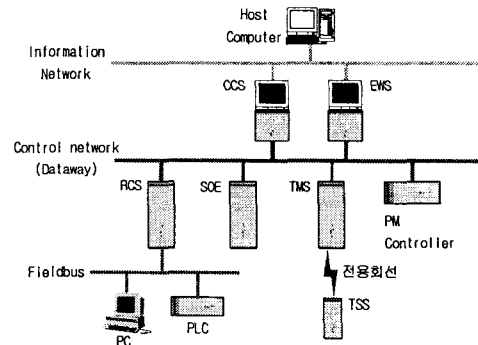


그림 7. 시스템 구성도  
Fig. 7 System configuration

EWS(Engineering WorkStation)라는 컴퓨터 시스템에서 엔지니어 작업을 수행하여 운전시의

상위 시스템인 CCS(Central Control Station)에 작업된 제어 프로그램을 복사를 하여 하위 제어 시스템인 RCS(Remote Control Station)에 다운로드 하여 제어 프로그램을 실행하게 된다.

II-3. 제어 프로그램 작성 및 적용절차

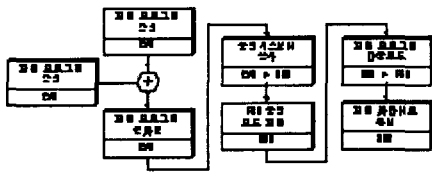
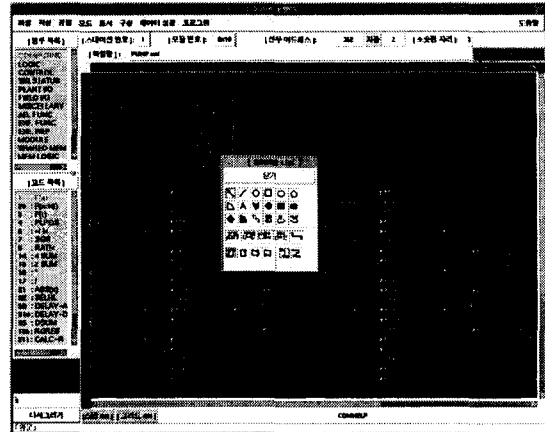


그림 8. 제어 프로그램 적용 절차  
Fig. 8 Procedure for applying control algorithms

그림 8과 같이 제어 프로그램을 작성하여 시스템에 적용하기 위해서는 EWS에서 기능 블록을 이용하여 기능 블록 작성기로 제어 프로그램들을 작성해야 한다. 이것이 완료되면 EWS에서 AI(Analog Input), DI(Digital Input), AO(Analog Output), DO(Digital Output), PIM(Pulse Input Module) 등과 같은 시스템에 실장되어 있는 입/출력 카드 정보를 이용하여 제어 프로그램을 하위 시스템인 RCS가 운영할 수 있는 데이터로 컴파일을 해야 한다. 컴파일이 정상적으로 완료되면 이를 운전 시스템인 CCS에 복사를 해야 한다. 복사가 이루어진 뒤 하위 시스템을 현재의 운전 상태에 오류 동작이 없도록 조절을 한 뒤 컴파일된 데이터를 다운로드한다. 다운로드 완료 후에는 작성된 제어 프로그램이 실행하도록 명령을 조작한다. CCS에서 실제 제어되고 있는 제어 프로그램의 중간값들을 보면서 세밀한 튜닝 작업을 수행하면 된다.

II-4. 기능 블록 작성기 구현

본 연구에서 구현된 기능 블록 작성기의 화면은 그림 9와 같다.



운영체제 : Solaris 2.6  
사용언어 : C, X11 Library, X Toolkit Library, Motif Library

그림 9. 구현된 기능 블록 작성기 화면  
Fig. 9 Screen of the Implemented function block builder

프로그램의 상단에 "CONTROLMASTER"라는 제목이 표시되어 있으며, 그 하단에 메뉴들이 표시되어 있다. 화면의 왼편에 제어 코드의 범주(Category) 목록과 해당 범주에 속한 제어 코드 목록과 현재 선택된 제어 코드의 일반 정보가 화면에 표시된다. 그리고 화면의 오른편에 상단에는 현재 작업하고 있는 파일의 일반 정보가 표시되며, 하단에는 작업에 대한 도움말의 메시지 표시 영역이 있으며, 나머지가 제어 프로그램을 작성할 수 있는 공간인 작성 영역으로 구성된다.

본 연구에서 구현된 제어 프로그램 작성기의 특징을 나열하면 다음과 같다[4].

- ① 다양한 제어 코드 제공(200여개 이상 제공)
- ② Zoom In/Out 기능 제공
- ③ 제어 코드 연결의 수동/자동 모드 설정 기능
- ④ 제어 프로그램이 점유된 메모리 맵 제공
- ⑤ 제어 코드의 방향(좌 우, 우 좌) 선택 기능
- ⑥ 제어 블록 번지의 자동 지정 기능
- ⑦ 제어 블록의 자동 여유 공간 지정 기능
- ⑧ 제어 프로그램의 출력 기능

- ⑨ 실행 순서 모의 기능
- ⑩ 제어 블록 번지 교환 기능
- ⑪ 자주 사용하는 제어 코드 목록 편집 기능
- ⑫ 문자 또는 그림 도형 삽입 기능

이와 같이 다양한 기능을 제공하는 기능 블록 작성기를 구현함으로써 제어의 일반적인 개념과 시스템에 대한 기본 개념을 가지고도 충분히 공정 시스템에 적용이 가능한 제어 프로그램을 작성할 수 있다고 생각된다.

### V. 결 론

본 연구를 통하여 개발된 제어 프로그램 작성기는 제어의 기본 개념을 가지고 있고, 시스템에 대한 간단한 운영 방법만 알고 있다면 어떤 엔지니어도 쉽게 사용할 수 있다. 기존의 까다로운 문법을 지키며 작성된 텍스트 기반의 제어 프로그램 보다는 엔지니어 입장에서는 모든 데이터가 그래픽으로 표시되기 때문에 제어 프로그램의 작성/변경/디버깅/테스트 면에서 상당한 장점이 있다.

본 연구 이후의 개발에 있어 개선해야 할 내용들은 제어 프로그램의 디버깅을 위한 단계별 TRACE기능을 부여할 수 있는 구조에 대한 보다 심층적 연구와, 실제 시스템에 적용하기 전에 시뮬레이션 과정에 의해서 단순한 로직등을 확인할 수 있는 시뮬레이션 기능이 부가된다면 보다 폭 넓은 연구가 이루어질 것으로 기대된다.

### 참고문헌

[1] 한국전력공사 기술연구원, 보일러의 디지털 분산 제어 시스템 개발 및 적용(최종보고서), 1993, 17-21

[2] J.R. LEIGH, APPLIED DIGITAL CONTROL, Prentice/Hall International, 1985, 323-347

[3] BENJAMIN C. KUO, Automatic Control Systems(Fourth Edition), Prentice-Hall,

Inc., 1982, 468-553

[4]. LG산전, 분산형 공정 제어 시스템 MASTER P-3000(CCS엔지니어링 설명서), 1996, 83-86

저자소개

권만준(Mann-Jun Kwon)



1989년:부산대학교 전자공학과 공학사

1991년:한국과학기술원 전기및전자공학과 공학석사

1991-2000년:(주)LG산전 연구소 시스템연구실

2000년-2002년:(주)원클릭테크놀로지스 부설연구소 現, 대천대학 컴퓨터전자전기학부 전임강사

※관심분야 : 컴퓨터통신 및 네트워크, 컴퓨터 시스템