MICREX-SX 교육자료

BASIC COURSE

2008. 1.

Fuji Electric FA Component & Systems Co.,Ltd
KIA MOTOR Co., Ltd

Ⅰ MICREX-SX 기초편

Ⅱ MICREX-SX 하드웨어편

Ⅲ MICREX-SX 프로그래밍 언어와 명령

Ⅳ MICREX-SX 조작 매뉴얼

Ⅰ. MICREX-SX

기초편

MICREX-SX 기초편 목차

1. IEC 규격에 대하여

(1) IEC란

IEC(International Electrotechnical Commission : 국제전기표준회의)는, 전자와 전기분야의 국제표준규격을 정하는 것을 목적으로 하며, 1908년에 설립된 국제기관으로, 전기기술 이외의 공업기술에 관하여 국제규격을 정하는 ISO(International Organization for Standadization : 국제표준화기구)와의 자매조직입니다.

(2) IEC61131 규격

IEC61131은 IEC가 작성하는 PLC의 국제규격으로, 몇 개의 파트로 구성됩니다.
아래는 JIS화되어 있는 주요한 파트를 표합니다.

IEC규격번호	JIS규격번호	타이틀	내용
IEC61131-1	JIS B 3501	일반정보	기본용어와 개념의 정의
IEC61131-2	JIS B 3502	장치의 요구사항과 시험	하드웨어에 관한 규정 · PLC의 하드웨어에 관한 요구사항 · PLC메이커의 제공사항 · 시험방법과 수순
IEC61131-3	JIS B 3503	프로그램 언어	프로그램의 공통요소, 프로그램 언어 프로그램 실행의 정의

(3) 주요한 내용

① IEC61131-1

프로그래머블 컨트롤러 기능특성(JIS B 3501)

PC 및 주변장치의 선택, 적용에 맞춰 필요되는 기능특성, 사양 등을 제공해야하는 정보에 대한 규정.

· PC시스템의 기본 기능 등에 관한 특성

처리, 외부 인터페이스, MMI, 프로그래밍, 디버그, 시험, 전원, 어베일러빌리티, 신뢰성, 인간공학적 특성

· 장치의 가동조건, 수송, 보관에 관한 요구사항

물리적 환경(온도, 습도, 오염도, 내 부식성, 고도), 전기적 가동조건(전원, 노이즈, 변동), 기계적 조건(진동, 충격, 자연낙하), 특수가동조건, 운송보관

· 전기적 사양

전원(정격, 동작범위, 순간정지, 전압낙하)

DI, DO, AI, AO의 사양(인터페이스형식, 전압전류동작영역, 임피던스 한계치, 제조업자의 제공정보)

통신 인터페이스, MPU, 메모리, 리모트 IO, 프로그래밍 툴, 노이즈이뮤니티 절연특성, 자기진단

· 기계적 사양

감전보호, 공간거리, 연면거리, 내화성, 인클로저. 단자, 설치, 전선, 착탈, 전지, 표시식별

② IEC61131-2
프로그래머블 콘트롤러 시험 및 검증방법(JIS B 3502)
PLC 및 주변장치가 IEC1131-1에 적합한가를 판정하기 위해서 필요한 시험 및 검증방법

· 내후성 시험
고온, 저온, 온도변화, 온습도 싸이클 내성

· 기계적 시험
진동, 충격, 자연낙하, 안전성

· 전기적 시험
절연내력, 보호접지, 노이즈이뮤니티(정전기방전, 방사전자계, 퍼스트 트랜지트 버스트, 감쇠진동파)

· 전원특성 검증
전원전압변동, 주파수변동, 정전, 순간단전, 전원 오접속, 메모리 백업

· 입출력 특성 검사(DI, DO, AI, AO)
동작범위, 신호역극성내성, 보호부 출력, 과부하특성

· MPU의 특성검증

· 리모트 IO의 검증
응답시간, 교신차단시험

(4)IEC61131-3의 목적

프로그램 언어의 규격인 IEC61131-3의 목적은 다음과 같습니다.

· PLC 프로그램 언어의 통일된 구문 및 의미의 규정
· PLC로의 프로그램의 실제 장치를 지원하는 구성요소의 규정
구성요소 : 구성, 리소스, 태스크, 글로벌 변수 등,
· PLC와 자동화 시스템의 다른 컴포넌트와의 통신사양의 규정
(규격JIS B 3503 1.4목적으로)

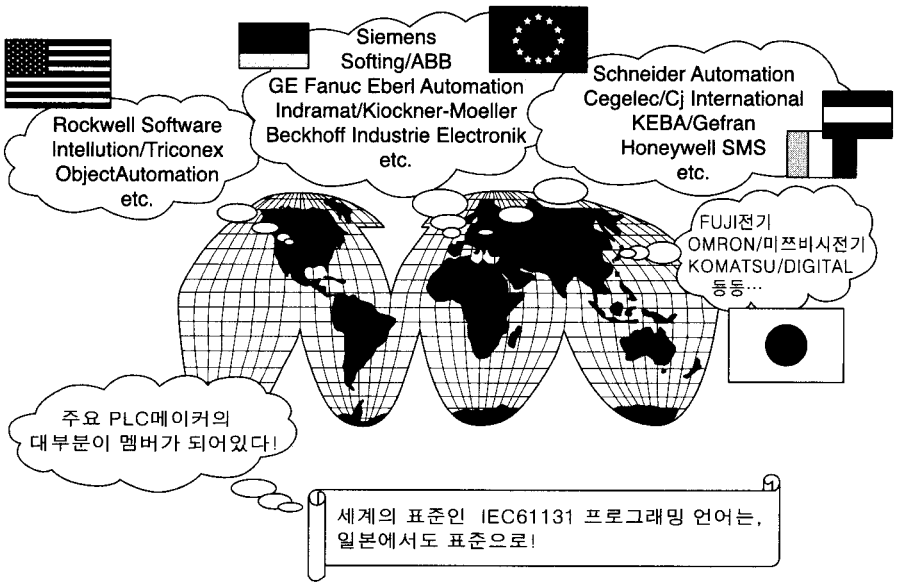
2. IEC 규격 대응 PLC 동향

(1) IEC규격이 지향하고 있는 것

· 국제적 레벨로 표준화된 프로그램의 실현
· PLC의 기종에 의존하지 않는 프로그램의 실현
· 구조화 프로그래밍에 의해 알기 쉽고, 관리하기 쉬운 프로그램의 실현
· 프로그램 부품의 재 이용에 의해 프로그래밍 효율의 향상, 및 프로그램 부품의 메이커를 초월한 유통의 실현
· 데이터형의 엄밀한 선언 등, 엄밀한 문법 체크에 의한 잘못이 적은 프로그램의 실현.⇒ 상류 공정의 중시. 실기시험 등 후 공정의 효율향상.

소프트웨어 공학이 뒷받침이 된 프로그래밍 수법

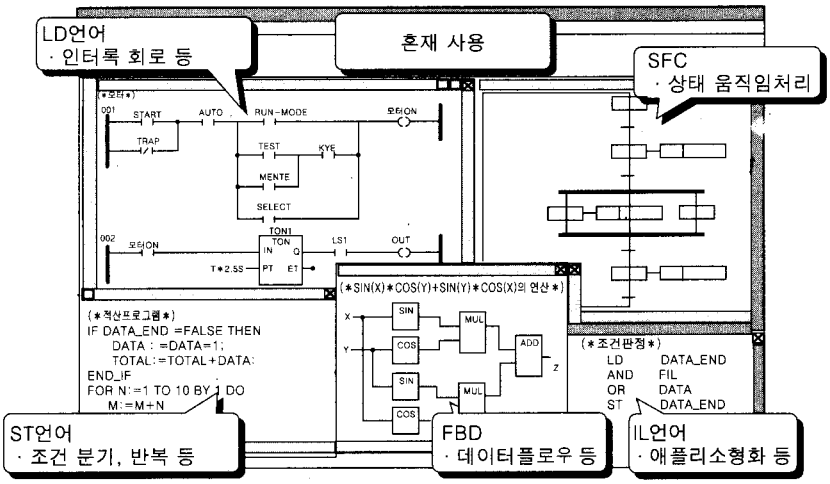
(2) 세계 PLC 메이커의 PLCopen으로의 가입



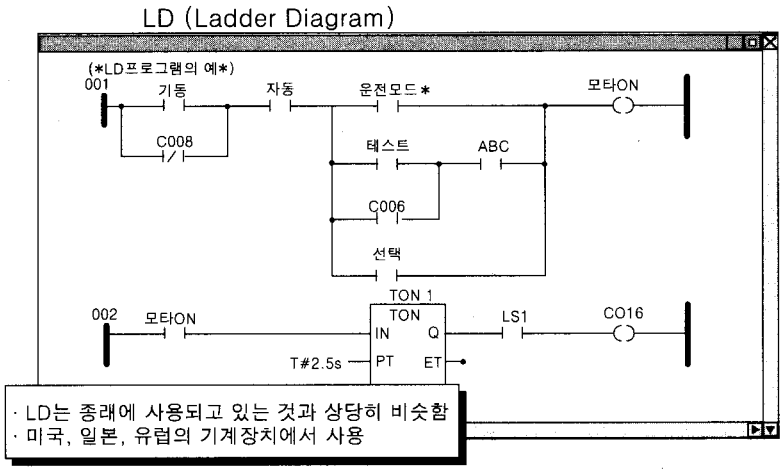
3. IEC 61131-3(프로그램)의 특징

3.1 IEC5 언어를 지원(용도별의 언어 세트)

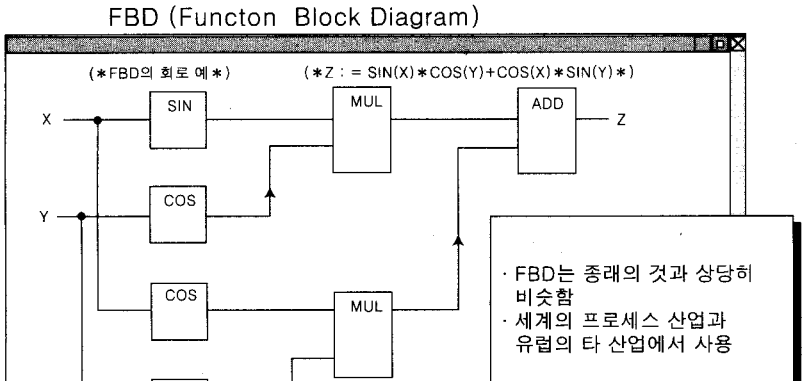
D300win은, IEC61131-3이 규정하는 PLC용 프로그래밍 언어인 5 언어(IL, ST, LD, FBD, SFC)모두를 지원하고 있습니다. 제어 내용에 대응하여, 또는 프로그래머의 익숙함에 맞추어 최적의 언어를 사용할 수 있습니다.



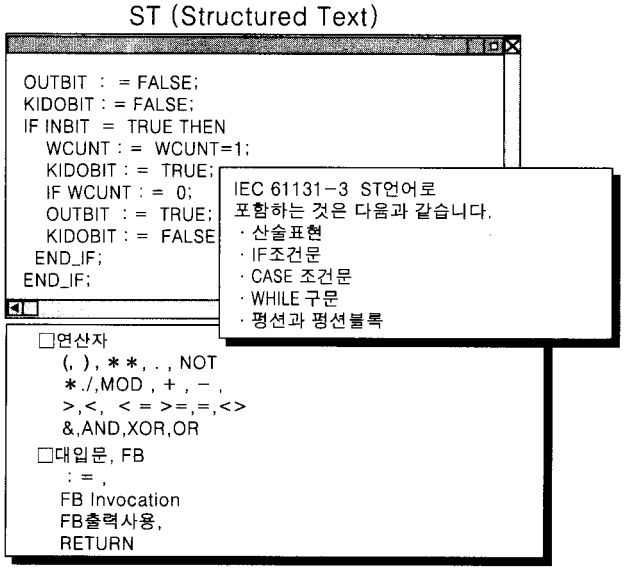
(1) LD(Ladder Diagram)



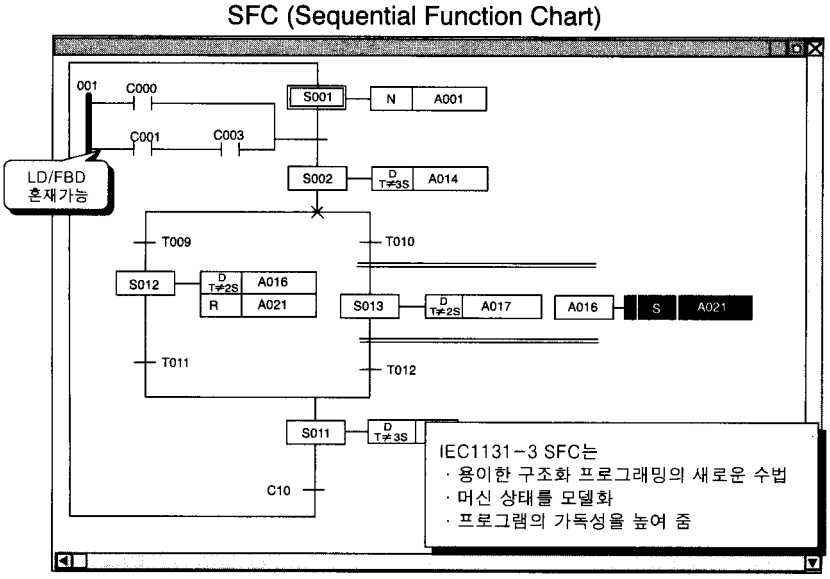
(2) FBD(Functon Block Diagram)



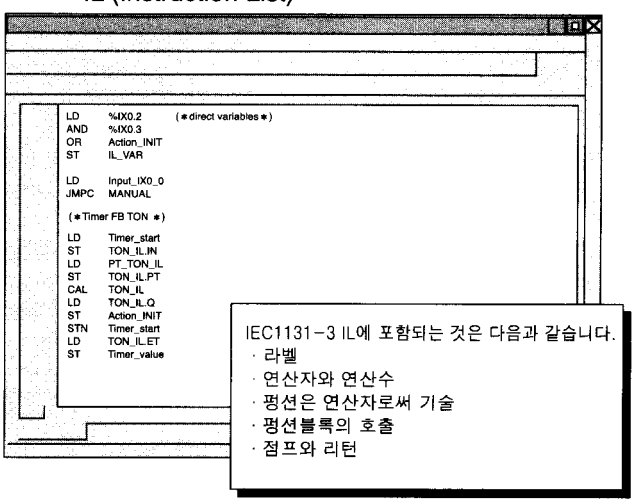
(3) ST(Structured Text)



(4) SFC(Sequential Function Chart)



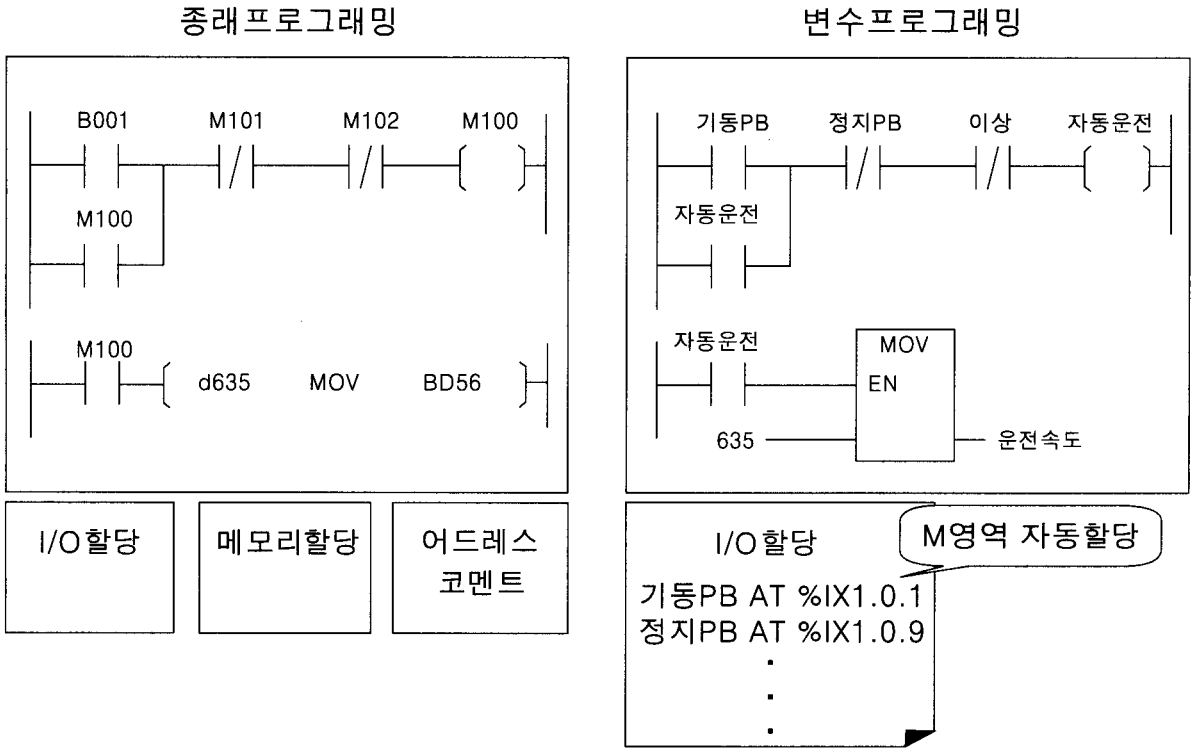
(5) IL(Instruction List)



3.2 변수에 의한 프로그래밍

(1) 메모리 자동할당에 의한 프로그램 향상

메모리 자동 할당에 의해 프로그래밍 효율이 향상됩니다.



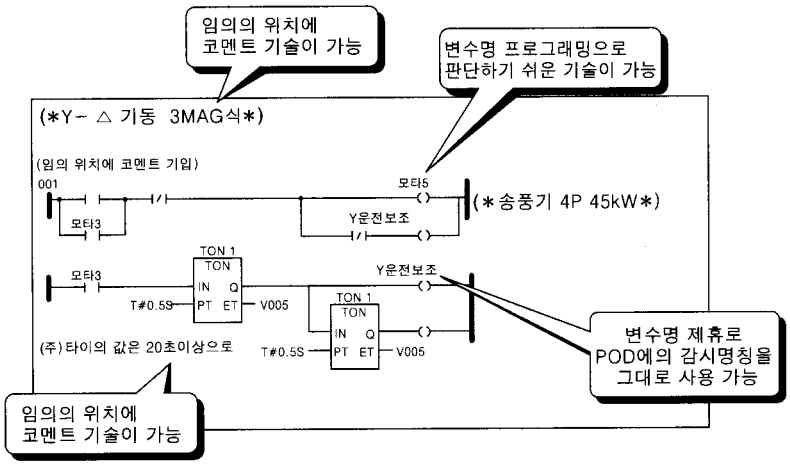
· 메모리 어드레스를 의식하지 않아 프로그래밍이 가능합니다.
· 여러 사람이 작성할 때, 메모리의 할당이 쉽습니다.
· M영역 : 어드레스 자동 할당
· I/O 할당 : 프로그램과 독립

종래 프로그래밍 수법과의 큰 차이점으로, 변수에 의한 프로그래밍이 있습니다.

<종래는 번지 코멘트>
사용하는 PLC 개별의 실 I/O어드레스를 사용합니다. 이 어드레스는 단순히 I/O의 배치된 물리적 어드레스로, 실제의 신호와 데이터와의 관련이 없기 때문에 알기 어려우며, 또한 항상 메모리 할당표를 참조하지 않으면 안됩니다.
시스템 구성의 다른 PLC로의 재 이용에는, 작성한 프로그램 그것을 수정하지 않으면 안됩니다.
<변수 프로그래밍은>
실제의 I/O의 실 어드레스와 관계가 없고, 자유롭게 이름을 정의할 수 있습니다.
따라서, 취급하는 신호와 데이터를 연상하는 이름을 지을 수 있어, 프로그램이 알기 쉽도록 됩니다.
또한 개개의 PLC 시스템에 의존하지 않는 프로그램의 표준화가 가능합니다.
실제의 I/O 할당이 결정되지 않아도 프로그램 작성이 가능합니다.
개개의 PLC의 실 I/O의 할당 작업은 EXCEL등으로 작성하면, 효율이 더욱 향상됩니다.

(2) 프로그램 가독성의 향상

코멘트의 자유 위치 기술과 변수명이 프로그래밍 가능하기 때문에, 회로와 회로블록에 적절한 코멘트 기재가 가능하며, 설계자의 의도를 시험, 결국 보수 담당자에게 명확하게 전달하는 것이 가능합니다.



프로그램 및 코멘트는 시트상의 자유로운 위치에 기술하는 것이 가능합니다.
①변수명에는 신호와 데이터를 연상시키는 알기 쉬운 이름을 붙여, ②프로그램을 기능블록마다에 모아 두고, ③또한 적절한 설명문을 덧붙이면, 다른 설계자, 보수담당자에게도 대단히 알기 쉬운 설명서(프로그램)가 됩니다.

(3) 변수 이름의 제약사항 및 주의 사항

- 변수 이름에 사용가능한 문자는 영숫자만 입니다.
- 변수 이름에 사용 가능한 기호는 '_' 뿐입니다. 단, '_' 의 연속 사용 '__' 은 안됩니다.
- 변수 이름의 선두에 반각숫자를 사용할 수는 없습니다.
- 변수 이름의 길이는 반각으로 30문자, 전각으로 15문자입니다.
- 시스템에 예약해 있는 예약어는 사용할 수 없습니다.
  예약에에 관해서는 User' s manual FEH200 Appendix6을 참조해 주십시오.
- 변수 이름은, 영문자의 대문자와 소문자는 구별되지 않습니다. 따라서 'ABCD' 'abcd' 는 동일 변수로 봅니다.
- 변수 이름에 공백(스패이스)을 사용할 수는 없습니다.

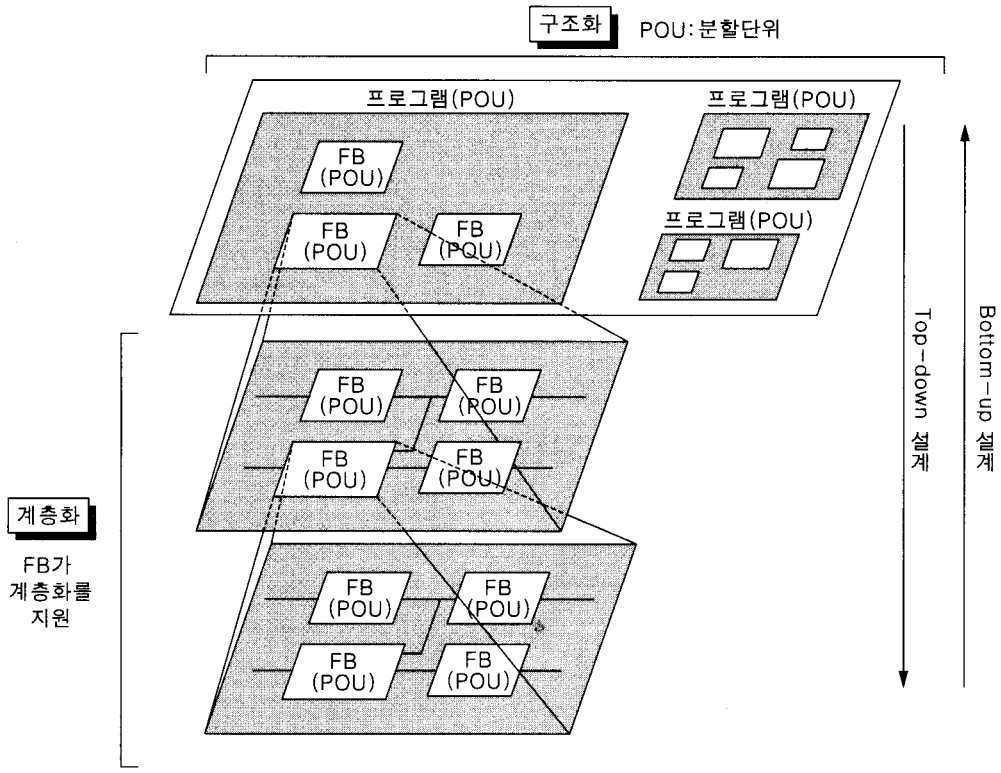
3.3 프로그램의 구조화 · 계층화 설계

(1)프로그램의 구조화 · 계층화 설계란

규격 IEC61131-3의 가장 큰 장점인 구조화 · 계층화 설계의 기법은, 1970년대 전반에 네덜란드의 E.W 다이쿠스트러에 의해서 제창된 기법으로, 결코 새로운 기법은 아니지만 오늘의 컴퓨터와 퍼스널 컴퓨터에 있어 프로그래밍 기술의 베이스가 되고 있는 기법입니다.

다이쿠스트러의 구조화 기법은, 1개의 소프트웨어를 큰 기능에서부터 1단씩 작은 기능으로의 톱다운으로 전체의 구조를 계층적으로 구조화하는 것입니다. 그리고, 그 과정에 있어서 1개의 기능 부분에만 집중하여 해결방법을 생각하는 것으로 고품질의 소프트웨어를 구축하고자 하는 것입니다. 그 때문에, 브레이크다운의 과정에서, 반드시 한 단계씩 그 소프트웨어의 올바름을 검증하는 것을 장점으로 하고 있습니다.

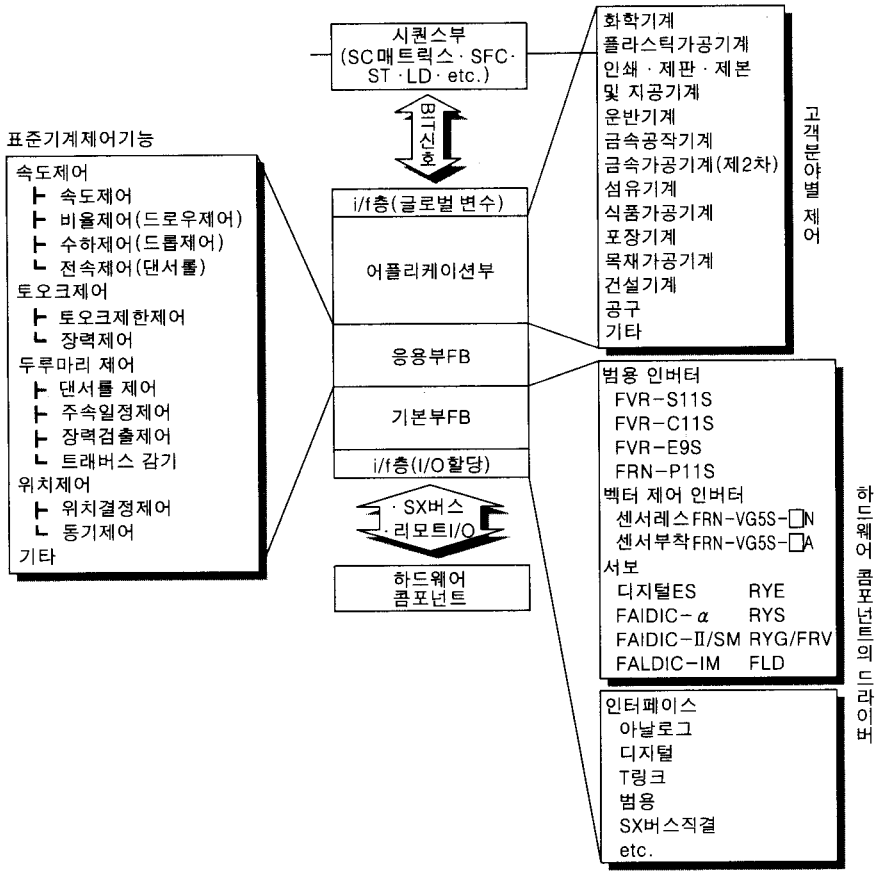
결국, 구조화라는 것은, 1개의 프로그램을 몇 개의 작은 기능부분으로 분할해서, 어떠한 구조(짜임새)를 부여하는 것입니다. 그리고, 계층화라는 것은, 구조화의 과정에서 1개의 프로그램의 기능에 초점을 맞추고, 보다 작은 기능부분으로 순차적으로 분해해 나가는 것입니다. 그 이미지는 아래의 예와 같습니다.



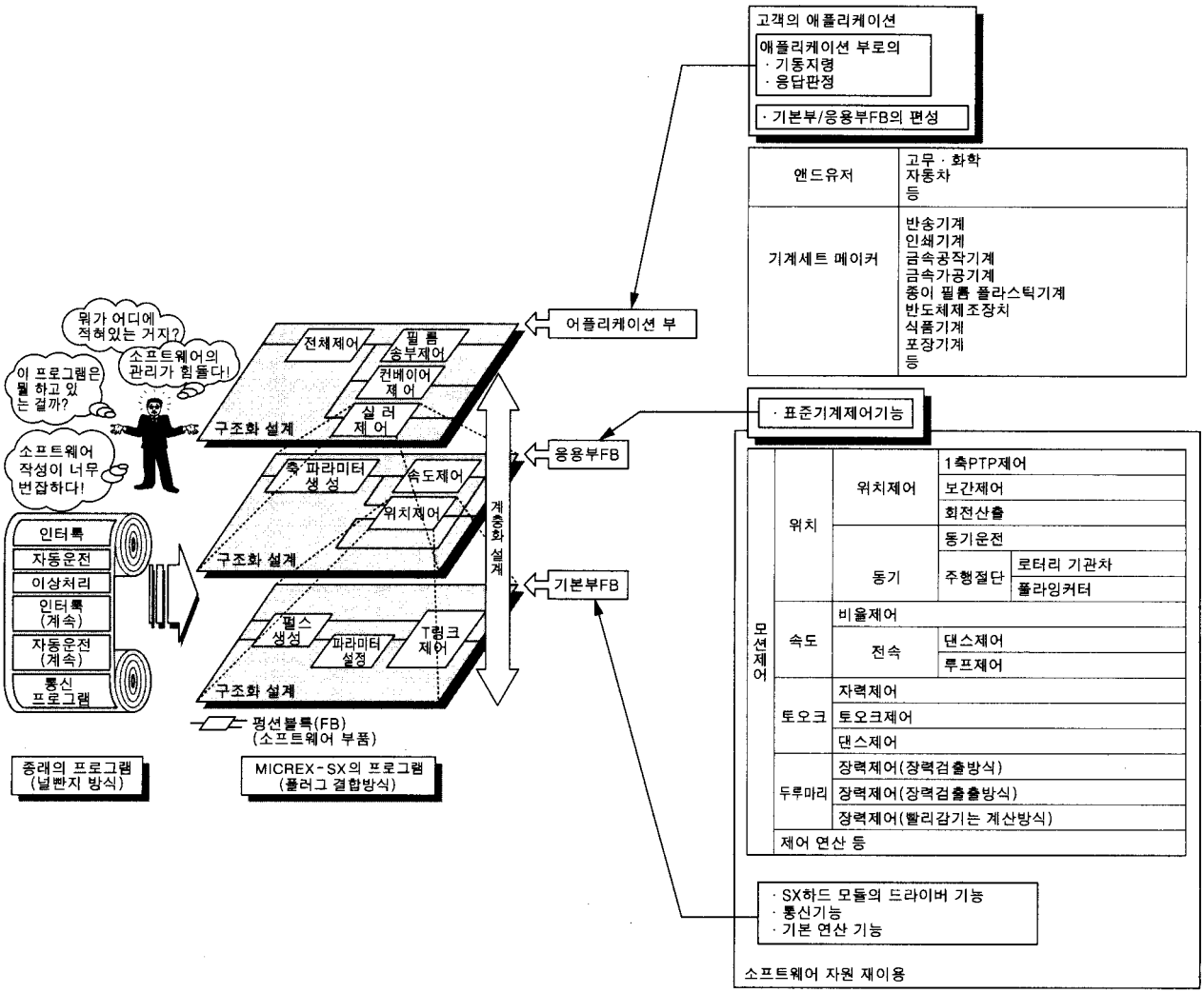
(2) 대규모의 프로그램을 구조화 · 계층화하는 메리트는 다음과 같습니다.

- 한 개의 기능을 한 개의 단위로 분할함으로써, 프로그램의 복잡함을 경감하고, 그 설계와 테스트가 용이하게 되며, 품질의 향상을 가져온다.
- 한 개 한 개의 프로그램의 독립성을 고양함으로 해서, 수정 · 변경의 영향을 최소한으로 할 수 있음.
- 복수의 프로그램 설계자가 병행해서 프로그램의 설계를 할 수가 있으며, 개발시기가 단축된다.
- 작은 기능 단위의 프로그램을 작성함으로 해서, 표준화, 재 이용하기 쉬운 프로그램이 된다.

(3) 기계제어의 대응



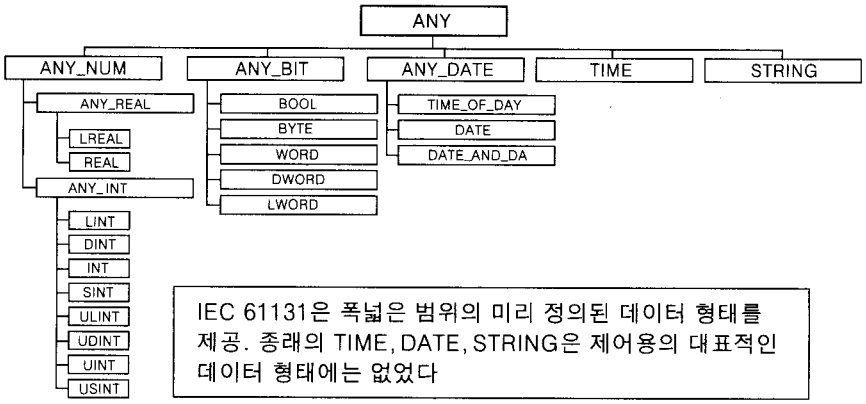
(4) 구체적 구성의 예


고객의 애플리케이션
애플리케이션 부로의
· 기동지령
· 응답판정
· 기본부/응용부FB의 편성
앤드유저
고무 · 화학
자동차
등
기계세트 메이커
반송기계
인쇄기계
금속공작기계
금속가공기계
종이 필름 플라스틱기계
반도체제조장치
식품기계
포장기계
등
어플리케이션 부
· 표준기계제어기능
응용부FB
기본부FB
계층화 설계
구조화 설계
전체제어
필름 송부제어
컨베이어 제어
실 러 제 어
축 파라미터 생 성
속도제어
위치제어
펄스 생성
파라미터 설정
T링크 제어
뭐가 어디에 적혀있는 거지?
소프트웨어의 관리가 힘들다!
이 프로그램은 뭘 하고 있는 걸까?
소프트웨어 작성이 너무 번잡하다!
인터록
자동운전
이상처리
인터록 (계속)
자동운전 (계속)
통신 프로그램
평션블록(FB) (소프트웨어 부품)
종래의 프로그램 (널빤지 방식)
MICREX-SX의 프로그램 (풀러그 결합방식)
모션제어
위치
위치제어
1축PTP제어
보간제어
회전산출
동기
동기운전
주행절단
로터리 기관차
플라잉커터
속도
비율제어
전속
댄스제어
루프제어
토오크
자력제어
토오크제어
댄스제어
두루마리
장력제어(장력검출방식)
장력제어(장력검출방식)
장력제어(빨리감기는 계산방식)
제어 연산 등
· SX하드 모듈의 드라이버 기능
· 통신기능
· 기본 연산 기능
소프트웨어 자원 재이용

3.4 데이터 형

IEC61131-3에는 다음과 같은 데이터 형태가 준비되어 있습니다.

기본 데이터 형



IEC 61131은 폭넓은 범위의 미리 정의된 데이터 형태를 제공. 종래의 TIME, DATE, STRING은 제어용의 대표적인 데이터 형태에는 없었다

MICREX-SX로 지원하는 데이터 형태

NO	키워드	데이터 형	비트 수	데이터 범위	MICREX-SX
1	BLLO	불 형	1	0 or 1	○
2	SINT	단정도정수 형	8	$-2^{7}\sim2^{7}-1$	–
3	INT	정수 형	16	$-2^{15}\sim2^{15}-1$	○
4	DINT	배정도정수 형	32	$-2^{31}\sim2^{31}-1$	○
5	LINT	4배정도정수 형	64	$-2^{63}\sim2^{63}-1$	–
6	USINT	부호없이 단정도정수 형	8	$0\sim2^{8}-1$	–
7	UINT	부호없이 정수형	16	$0\sim2^{16}-1$	○
8	UDINT	부호없이 배정도정수 형	32	$0\sim2^{32}-1$	○
9	ULINT	부호없이 4배정도정수 형	64	$0\sim2^{64}-1$	–
10	REAL	실수형	32	$\pm2^{-127}\sim\pm2^{129}$	○
11	LREAL	배정도실수 형	64	$\pm2^{-1023}\sim\pm2^{1025}$	–
12	TIME	계속시간 형	처리계의존	0일~50일	○
13	DATE	날짜 형	처리계의존	1970년~2106년	○
14	TOD	시각 형	처리계의존	0:00~23:59	○
15	DT	날짜시각 형	처리계의존	1970년~2106년	○
16	STRING	가변장문자열 형	처리계의존	–	○
17	BYTE	길이8비트열 형	8	–	–
18	WORD	길이16비트열 형	16	–	○
19	DWORD	길이32비트열 형	32	–	○
20	LWORD	길이64비트열 형	64	–	–

파생형 데이터 : 배열(1차원, 2차원)　　TIME : ms단위, 부호 없이 배정도정수
구조체(구조체의 배열, 배열의 구조체)　　DATE : s단위, 1970/1/1기준, 부호 없이 배정도정수
DT : s단위, 1970/1/1 0:00기준, 부호 없이 배정도정수

유저정의 데이터 형

Table 12-Data type declaration features

No.	Feature/textual example
1	Direct derivation from elementary types, e.g.: TYPE R:REAL:END_TYPE
2	Enumerated data types, e.g.: TYPE ANALOG_SIGNAL_TYPE:(SINGLE_ENDED, DIFFERENTIAL):END_TYPE
3	Subrange data types, e.g.: TYPE ANALOG_DATA:INT(-4095..4095):END_TYPE
4	Array data types, e.g.: TYPE ANALOG_16_INPUT_DATA:ARRAY[1..16]OF ANALOG_DATA:END_TYPE
5	Structured data types,e.g.: TYPE ANALOG_CHANNEL_CONFIGURATION: STRUCT RANGE:ANALOG_SIGNAL_RANGE: MIN_SCALE:ANALOG_DATA: MAX_SCALE:ANALOG_DATA: END_STRUCT: ANALOG_16_INPUT_CONFIGURATION: STRUCT SIGNAL_TYPE:ANALOG_SIGNAL_TYPE: FILTER_PARAMETER:INT(0..99): CHANNEL:ARRAY[1..16]OF ANALOG_CHANNEL_CONFIGURATION: END_STRUCT: END_TYPE
	NOTE-For examples of the use of these types in variable declarations, see 2.3.3.3.2.4.1.2, and table 17.

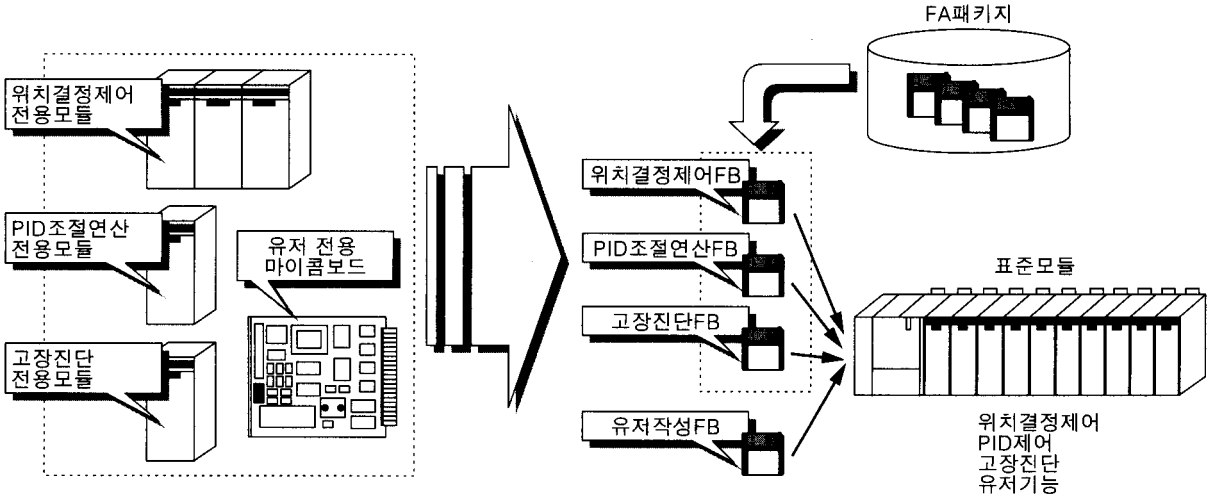
IEC 61131에서 유저는 자기 자신의 데이터 형태를 선언할 수 있다.

배열 데이터형, 구조화 데이터형, 배열과 구조체를 사용하면 프로그램은 일반적으로 짧아질 가능성이 높다

3.5 Function Block Program

(1) 제어용 프로그램 부품(FB : 펑션 블록)에 의한 프로그래밍

제어 용도에 맞춘 프로그램 부품(FB)을 작성 회로에 첨가함으로 프로그램의 작성이 가능합니다. 이것으로써, 프로그램 개발을 처음부터 해야할 필요가 없어졌으며, 개발 시간의 단축되었습니다. FUJI 전기의 옵션 FB 패키지(일부 발매 예정)로써 위치결정, 전자캠, 고장진단, PID, FA기기 범용통신 등이 있습니다.

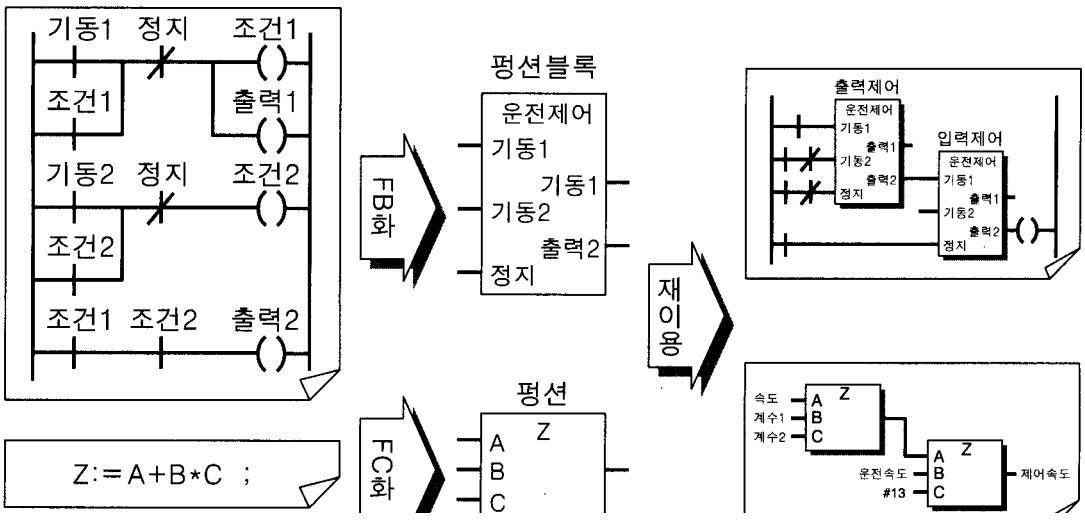


(2) 제어용 프로그램 부품은 유저도 작성가능

유저로서 작성한 회로(래더, 니모닉 등)를 프로그램 부품으로써 패키지화(FB)하는 것도 가능하며, 프로그램의 유용화가 가능하게 됩니다.

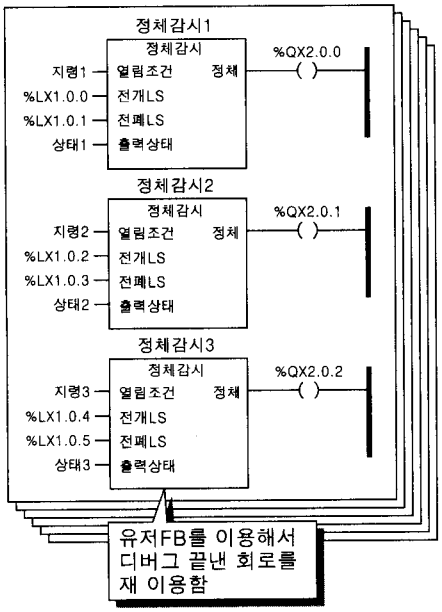
- 각 제어단위에 프로그램을 패키지화하고 유용화하는 것으로, 개발 시간의 대폭 단축과 프로그램의 구조화가 가능합니다.
- 패키지화한 프로그램의 내용은 마스터(소스)프로그램으로밖에 변경할 수 없기 때문에, 중요한 프로그램(제어)의 보호에 사용할 수 있습니다.

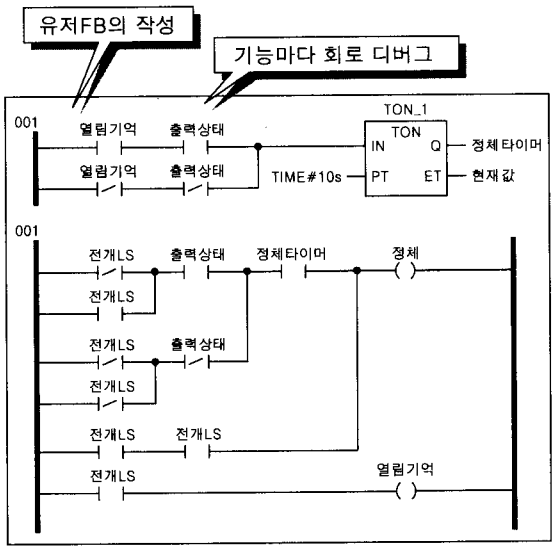
회로의 재 이용이 용이하게 된다⇒펑션블록(FB), 펑션(FB)에 의한 부품화

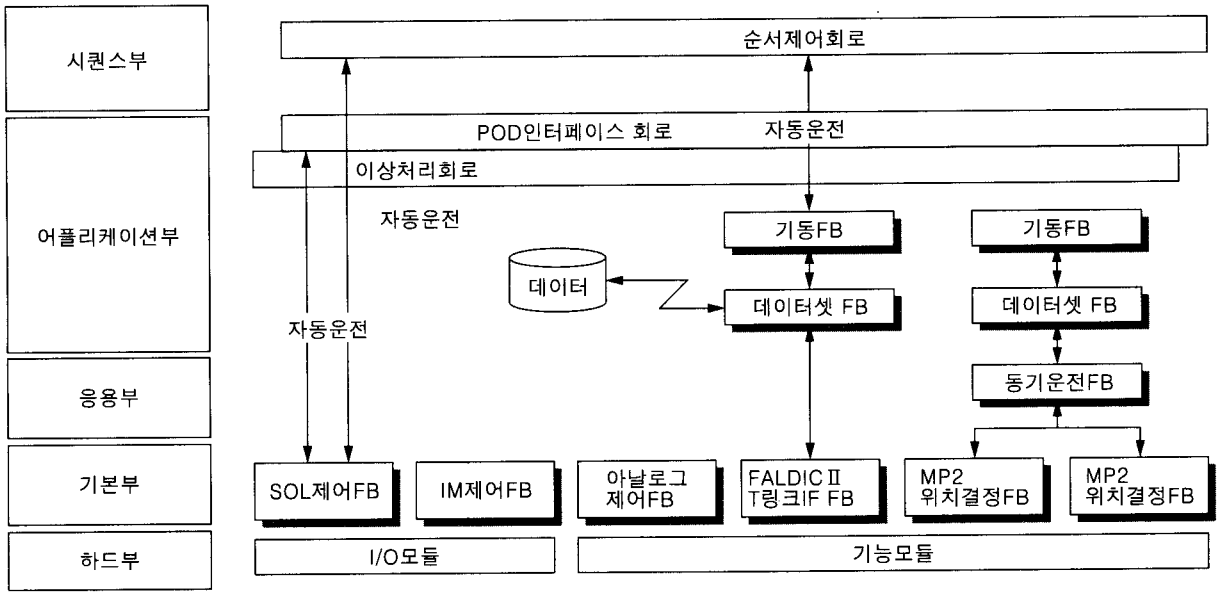


(3) FB활용에 의한 프로그래밍의 효율의 향상

• 유저 FB에 의한 프로그래밍의 효율의 향상 예







펑션 및 펑션블록에는, IEC61131-1로 정의되어 있는 표준 FCT, FB와 PLC메이커가 독자적으로 제안하고 공급하는 것, 그리고, 유저 자신이 작성, 등록하는 것이 있습니다.

FCT, FB는 블록화 된 소프트웨어 요소로써, 애플리케이션 프로그램 안에 여러 가지의 부분으로 용이하게 재 이용이 가능한 것입니다.

유저 FB화에는, 각각의 기능을 개별적으로 작성, 디버그해서 표준화 할 수 있으며, 개발 효율이 향상됩니다. 또한 그 유저 FB를 불러내는 메인 프로그램 측에서, 내부의 세부적인 논리를 의식하지 않고, 입출력 신호에게만 주목해서 프로그램을 작성할 수 있고, 프로그램을 단순하고 알기 쉽게 할 수 있습니다. 메인터넌스에 있어서도 유효합니다.

• 프로그램의 유저 FB화의 메리트

표준 회로를 디버그하지 않고 재 이용

⇒개발 효율의 향상

⇒기능마다 작성된 보기 편한 프로그램

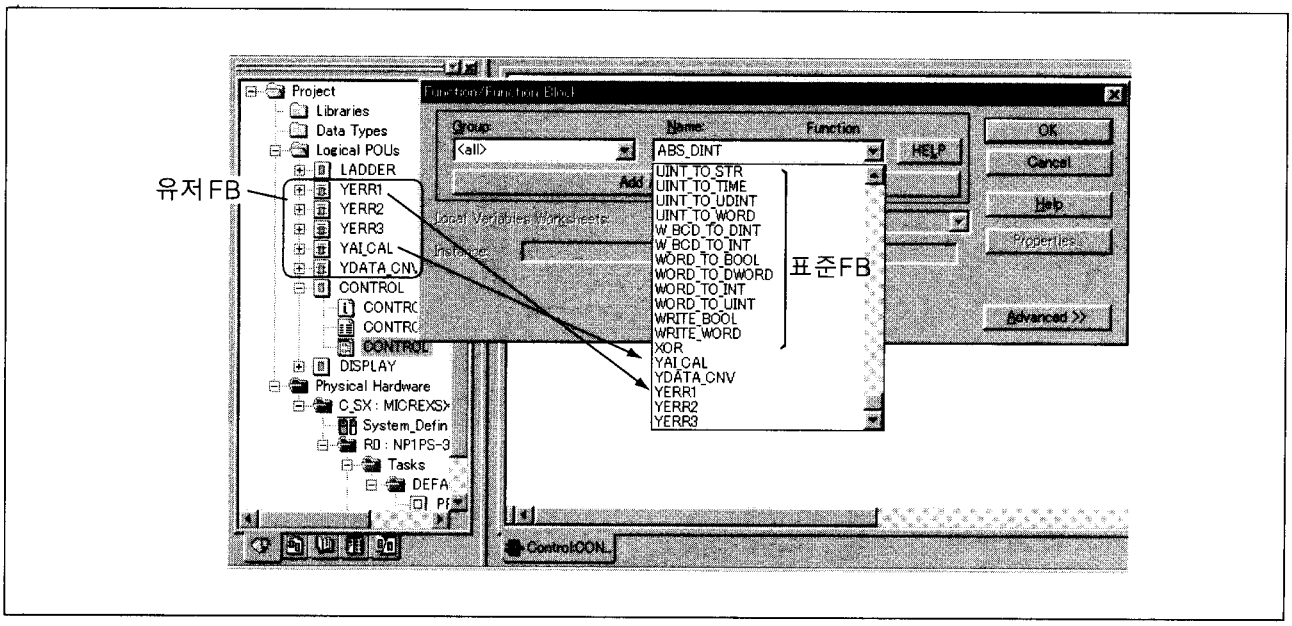
· 기계와의 IF회로를 독립해서 작성 · 디버그 가능

⇒상세 제어사양이 확정되기 이전에, 개발을 시작할 수 있다.

· 복수의 프로그래머에 의한 평행 작업이 가능

⇒납기의 압축

• D300win상의 FB의 호출 예



위의 그림처럼 D300win상에서는, FB의 등록 호출을 간단히 할 수 있고, 사용하기 쉬운 형태로 프로그램에 짜 넣을 수 있습니다.

• 유저 FB에 의한 프로그래밍 효율의 향상

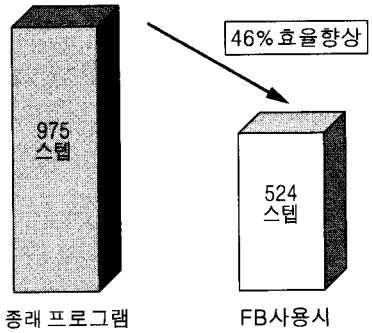
신규로 개발해야하는 스텝 수의 소멸

재 이용할 수 있었던 스텝 수/모든 스텝 수=451스텝/975스텝 프로그램 개발과, 46%의 효율향상

여기서는, 스텝 수의 소멸율을 개발(설계 · 작성 · 디버그)효율의 향상으로 간주하고 있습니다.

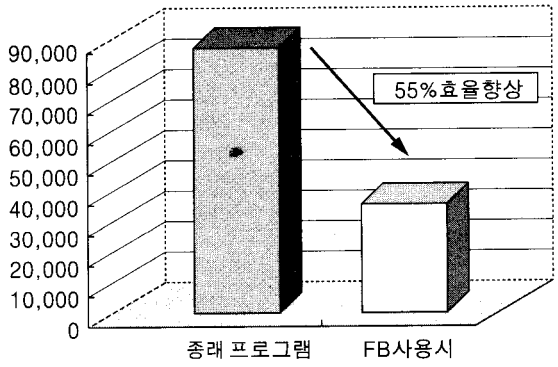
이 자료의 효율향상의 가치는, 기존 기종으로 개발한 실 어플리케이션의 프로그램을, D300win으로 MICREX-SX시리즈 SPH300용으로 프로그램 개발을 실행해, 얻어진 데이터를 근거로 산출하고 있습니다.

— 프로그래밍 개발 효율의 향상 예 1(발전기 제어)



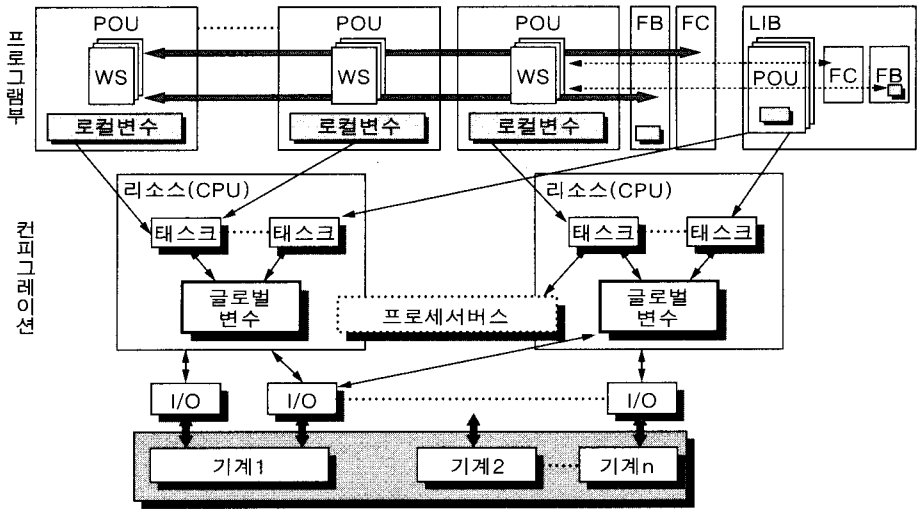
여기서는, 주로 FB이용에 의한 스텝 수의 소멸율을 개발(설계 · 작성 · 디버그)효율의 향상으로 간주하고 있습니다.

— 프로그래밍 개발 효율의 향상 예 2(타이어 제조기)



여기서는, 주로 FB이용에 의한 스텝 수의 소멸율을 개발(설계 · 작성 · 디버그)효율의 향상으로 간주하고 있습니다.

3.6 프로그래밍 구조



프로그램은 펑션, 펑션블록을 포함한 각 프로그램 구성 단위(POU)마다에 작성할 수 있습니다. 이 구조화에 의해서, ①기능마다에 개별의 프로그램을 작성, 표준화할 수 있습니다. (여러 사람들의 프로그램 작성, 디버그가 가능하고, 개발 효율이 향상됨) ②프로그램의 재 이용이 용이, 등의 효과가 있습니다.

또한, 프로그램에 사용하는 신호와 데이터에는 PLC내의 실 I/O어드레스를 의식하지 않고, 자유로운 이름은 붙일 수도 있습니다. (신호와 데이터를 연상할 수 있는 영숫자에 의한 이름, 예를들면 「START_PB_1」, 「EMERGENCY_STOP」등) 또한, 이 이름(변수)은, 그 POU만으로 사용하는 것(로컬 변수)과, 공통으로 사용하는 것을 구별하여 정의할 수 있습니다.

이런 것에 의해서, PLC 고유의 실 I/O어드레스에 의존하지 않고, 표준적인 프로그램의 작성, 디버그가 가능하게 되었습니다.

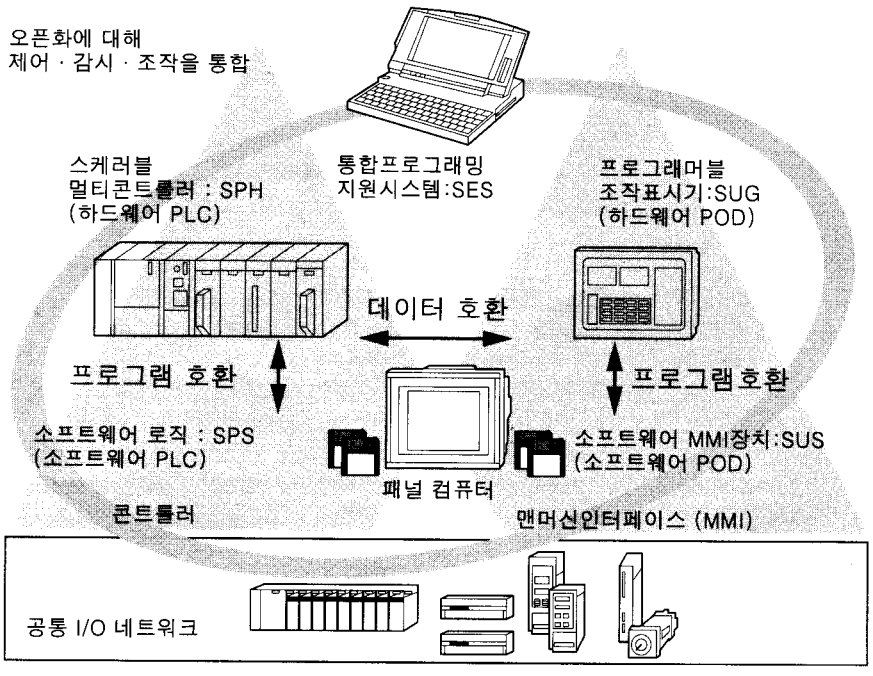
4. MICREX-SX시리즈란

4.1 MICREX-SX의 개요와 장점

MICREX-SX는 국제규격 IEC61131에 준거한 고속 · 고성능의 PLC로, 다음과 같은 장점을 가지고 있습니다.

① 고성능 CPU에 의한 고속의 프로그램의 실행처리
② SX버스에 의한 고속의 I/O 리프레시
③ 멀티 CPU에 의한 병렬처리와 기능분산처리
④ 국제표준언어 IEC61131-3의 채용에 의한 소프트생산성의 향상
 · 구조화 · 계층화설계에 의한 프로그램 설계효율의 향상과 품질의 향상
 · 펑션블록에 의한 프로그램의 표준화, 재 이용
 · 5언어의 채용에 의한 제어사양과 설계자의 기술에 맞춘 최적의 언어 선택
 · 변수에 의한 가독성 높은 프로그램 기술
 · 배열, 구조체 등 데이터표현의 충실
⑤ 용장화 시스템(CPU, 전원의 이중화)에 의한 고 신뢰성 시스템의 구축
⑥ 오픈 네트워크로의 대응
오픈 네트워크대응으로서 Ethernet, FL-net, OPCN-1, DeviceNet, LonWorks, AS-i의 마스터 모듈을 실제 장착이 가능(Ethernet는 PC카드 I/F로 대응)

4.1.1 통합 콘트롤러



4.1.2 고속처리

(a) MICREX-SX(SPH-300)의 사양

항목		사양	비고
명령	언어	IEC언어준거(IEC61131-3)	
	속도	접점명령:0.02μs 코일명령:0.04μs 고정소수점가감산:0.04μs 곱 셈:0.08μs 나눗셈:1.08μs 부동소수점가감산:0.08μs 곱 셈:0.08μs 나눗셈:1.48μs 타이머명령:0.44μs 카운터명령:0.22μs	DPRAM액서스:60ns (25nsSRAM사용시) 내부데이터메모리 액서스:20ns
메모리용량	프로그램 메모리	타입1:32K스텝	
		타입2:74K스텝	
	데이터 메모리	타입1:32K스텝	
		타입2:128K스텝	
	ZIP파일	128Kbyte	
입출력점수	디지털 아날로그	디지털SX버스접속+리모트링크접속:8192점Max. 아날로그SX버스접속+리모트링크접속:16비트 512점Max.합계점수제약:(생략)	
	타이머 카운터 미분릴레이 (FF,P,N등)	타이머(T):2048점 Max.(분해능력:0.001초) 카운터(CT):4096점 Max. 미분릴레이(PN):8192점 Max. 합계점수제약:T×8+CT×4+PN×2≦Max.16384	

전용 LSI에 의한 등급 최고처리속도 20ns

· 래더25k스텝1ms상당(부동소수점연산 80ns)

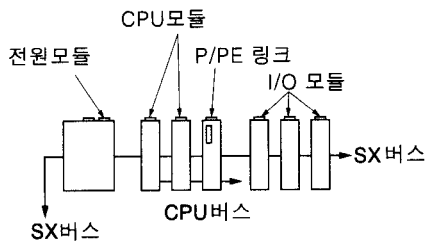
· OS프로세서와의 병렬동작

(b) SX버스에 의한 신호 리프레시 1ms

탁트 제어에 의한 I/O 리프레시 주기의 정시성을 확보하고 있습니다.

I/O 리프레시 싸이클이 최소 0.5ms, 0.5ms단위로 설정가능하기 때문에, 탁트 타임이 엄격하게 요구되는 처리에 최적입니다.

25Mbps의 I/O버스를 채용하여, I/O 리프레시를 최적화하고 있습니다.



처리속도의 향상은, PLC본래의 기능에 있는 시퀀스명령의 고속화뿐만 아니라, 수치연산의 고속화도 가져다 줍니다.
특히, 부동소수점 연산의 고속화는, PLC의 응용범위를 크게 확대, 모션컨트롤 분야, 퍼스널컴퓨터에서는 수치연산사용분야로의 적용 등, 지금까지 PLC가 할 수 없었던 분야로의 적용이 가능해졌습니다.

(c) 계산성능

No.	닥트시간	CPU대수	입출력 점수(비트)		프로그램 스텝(kstep)	
			직결 I/O	원격 I/O	20ns환산	35.2ns환산
1	1.0	1	512	0	29.0	16.0
2		2	512	0	29.0×2	16.0×2
3		1	1,024	0	20.0	11.0
4	2.0	1	2,048	2,048	36.0	20.0
5	3.0	1	4,096	4,096	72.0	41.0
6		2	4,096	4,096	72.0×2	41.0×2

(주) · CPU, I/O등의 하드웨어 구성에 의해 필요 탁트 시간은 변화합니다
· 제어내용(사용명령 외)에 의해, 탁트 시간내의 실행이 가능한 프로그램 스텝 수는 변화합니다

4.1.3 국제규격대응 오픈화 지향의 멀티 컨트롤러

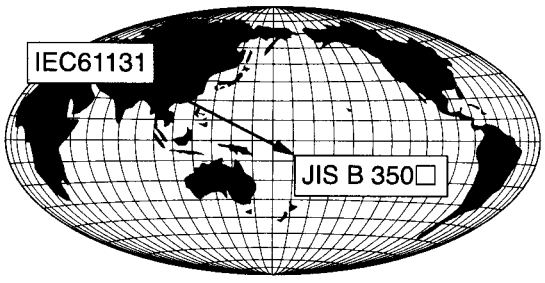
(a) 국제규격대응

① 국제표준

SX시리즈는 컴퓨터의 세계에서도 키워드가 되어 있는 「국제표준(프로그래밍 언어의 표준화와 오픈 네트워크로의 대응)」을 실현했습니다.

국제규격 IEC61131 준거

· 하드웨어, 소프트웨어와 함께 프로머블 컨트롤러의 국제규격인 IEC61131에 준거하고 있습니다.
· IEC61131은 이미 JIS규격화(JIS B3500~3)되어 있고, 국내에서도 표준규격이 되어 있습니다. 프로그래밍 언어와 국제규격 IEC61131-3에 완벽하게 준거. 세계 어느 곳 누구라도 알 수 있는 애플리케이션 프로그램의 작성이 가능합니다. 더욱이, 인더스트리얼 · 컨트롤 · 프로그래밍의 표준화(IEC61131-3의 보급)를 목표로 하는 국제조직「PLCopen」의 호환인증 BASELEVEL(기본 레벨)의 인정을 얻을 예정입니다.



② 적합 규격

아래의 각종 국제규격에 적합합니다.

적용규격

CE 마킹	
저전압지령	(IEC61131-2, UL508<절연거리>)
EMC지령	(EN50081-2<EMI>
	EN50082-<EMS>)

UL/CSA

NK, 로이드 규격

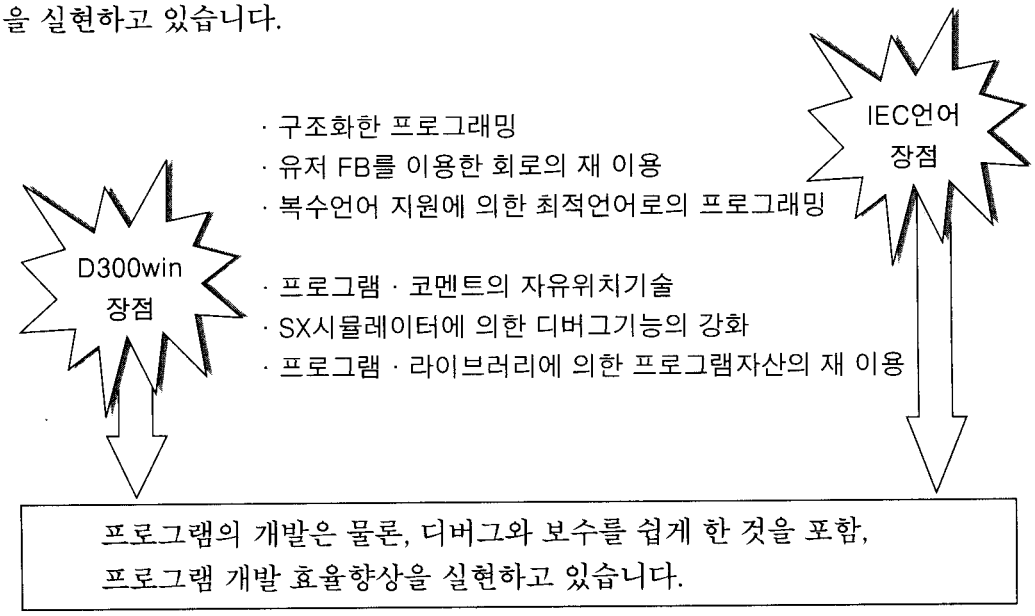
준거규격

IEC61131

JIS B3501~3	프로그래머블 컨트롤러
JEM-TR1200	범용PLC의 고주파억제대책 실시 요령
JEM-TR188	범용PLC의 카탈로그 기재사항
JEM-TR191	PLC의 선정과 적용지침

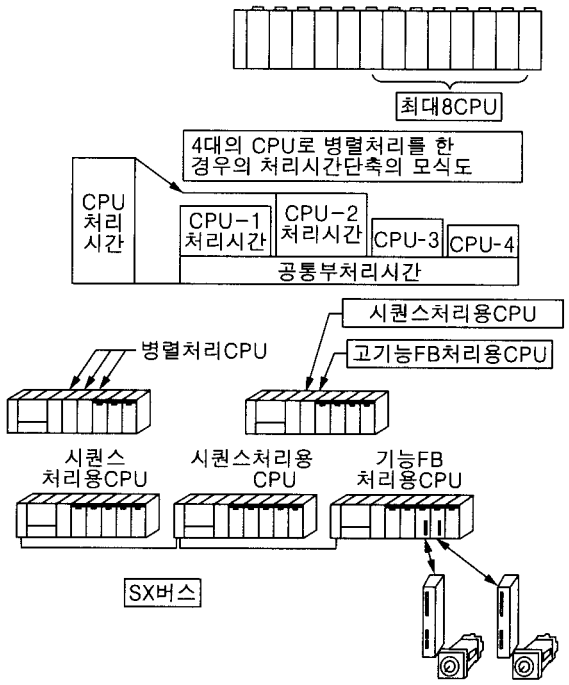
(b) 국제표준언어 IEC61131-3의 채용에 의한 소프트생산성의 향상

MICREX-SX용 프로그래밍 툴 D300win은 IEC61131-3의 모든 기능을 지원하고, 또한 최근의 퍼스널컴퓨터의 소프트(Windows GUI)의 편리성을 채택해 위화감이 없는 조작성, 다른 소프트와의 데이터 연계, 소프트 PLC에 의한 시뮬레이션 기능 등에 의해, 보다 효율적인 PLC소프트의 작성, 메인터넌스 성을 실현하고 있습니다.



4.1.4 멀티 CPU시스템이 가능

· 최대 8CPU에 의한 병렬처리가 가능합니다.
· 제어 내용에 따라서 프로그램을 개별의 CPU에 격납할 수 있습니다.
· 기능 모듈용, 처리프로그램전용 CPU를 분리하여 사용할 수 있습니다.
· 복수 CPU 시, I/O 리프레시 관리는 자동적으로 실행합니다.
· 분산 멀티 CPU구성에 의해 분산제어에 대응합니다.
· 기능 모듈전용 CPU로서, 기능 분산시스템의 구축이 가능합니다.



멀티 CPU의 목적

① 처리의 고속화

1개의 CPU로 실행시키는 것보다, 고속 처리가 가능

② 대용량의 프로그램에 스케일러블하게 대응

1개의 CPU로 실행시키는 것보다, 대용량의 프로그램 처리가 가능. 또한 프로그램을 추가하는 경우, 만약 용량이 1 CPU의 범위를 넘었어도, 한 등급 위의 시리즈로 하지 않고. 멀티CPU로 간단하게 대응할 수 있음

③ 프로그램의 구조화

프로그램 제어를 CPU마다에 기능분담하고, 알기 쉬운 메인터넌스가 가능

④ 분산제어

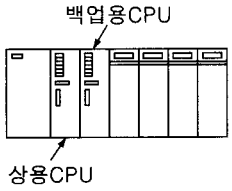
블록마다에 기능을 분담하고, 기능분산이 가능

4.1.5 이중화 시스템(CPU전원의 이중화)에 의한 높은 신뢰성 시스템의 구축

멀티 CPU에 의한 CPU의 용장화가 가능해서, 시스템의 안전 · 고 신뢰성의 확보에 공헌합니다.

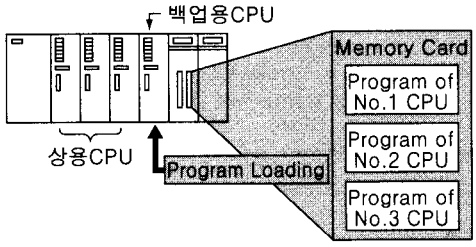
1 : 1 웜 스탠바이

CPU가 다운되어도 시스템을 정지시키지 않고 운전을 계속할 수가 있습니다.
본 구성은 최대 4조까지의 용장화 구성이 가능합니다.



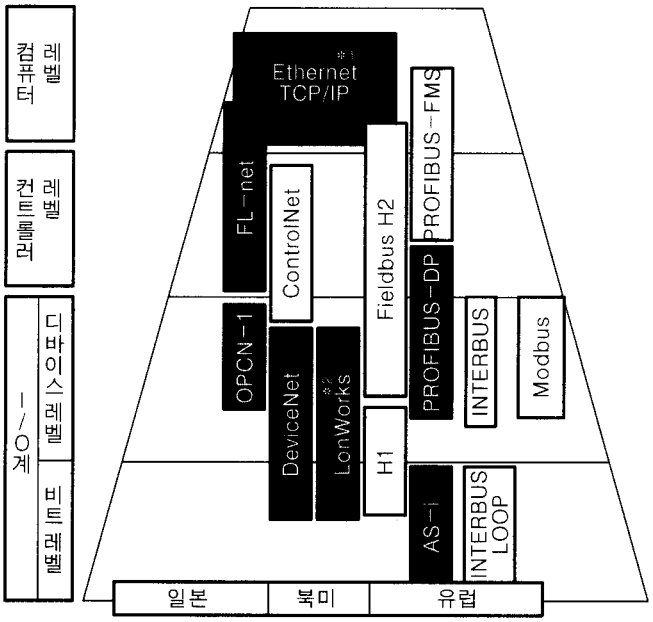
N : 1 콜드 스탠바이

본 시스템은, 복수의 상용계 CPU와 백업계 CPU 1대 및 메모리카드 모듈 1대로 구성합니다. 지금까지의 범용PLC계의 상식이던 1 : 1 시스템과 비교해서 굉장히 경제적인 용장화 시스템입니다.



4.1.6 MICREX-SX에 대응하는 오픈 네트워크

MICREX-SX는 종래의 프라이비트 네트워크인 P/PE링크, T링크를 계속 지원 하는가 동시에, 각종 오픈 네트워크에도 대응하고 있습니다.



■:MICREX-SX에 대응하는 오픈 네트워크

*1 Ethernet:미국Xerox Corp.의 등록상표
*2 LONWORKS:미국Echelon Corp.의 등록상표

오픈 네트워크 대응기기

항목 / 네트워크 카테고리	주요 대응 네트워크	마스터 모듈	오픈 네트워크 대응기기					비고
			집합형 인터페이스 모듈	단자대형 리모트 I/O	POD	인버터	서보	
컴퓨터 레벨	Ethernet	○	–	–	○	–	–	
컨트롤러 레벨	FL-net	○	–	–	△	–	–	
디바이스 레벨	OPCN-1	○	○	○	○	–	△	JIS 완료 B3511, B3512
	DeviceNet	○	–	○	–	○	△	
	PROFIBUS-DP	△	△	△	△	○	△	
	LONWORKS	○	–	△	–	–	–	
비트(센서)레벨	AS-i	○	–	○	–	–	–	

○대응 끝남, △개발 · 계획 중

ⓐ 상위 네트워크

다음은 PLC간 주요 네트워크를 표시합니다.

PLC-PLC간 주요한 네트워크 일람

항목	FL-net(=OPCN-2)	PROFIBUS-FMS	ControlNet	FIELDBUS(H2)	MELSECNET/10	P/PE링크
접속구성	버스	버스	버스		링	버스
거리	전기: 185m/500m/1.5km (1세그먼트, 리피터로 연장가) 광:미규정	전기 200m(1.5Mbps)	전기:250m~1km 리피터 사용 6km 광:30km		전기:500m/세그먼트 리피터 사용 2.5km 광:1km/세그먼트 리피터 사용 30km	전기:250m 광:1km
접속대수 (노드 수)	논리 :254 대 하드 : 1 세그먼트 100 대 리피터 사용 254 대	127대	논리:99대 하드: 1세그먼트48대 리피터사용99대		논리:32대 리피터사용64대 광:64대	16/64대
전송속도	10Mbps	500K/1.5M/12Mbps	5Mbps		10Mbps	5Mbps
전송매체	동축 광	트위스트 페어 광	동축 광		동축 광	동축 광
리프레시 시간	17k바이트 (2kbit+2kw)×32대= 50ms:토큰주기 32국 유저메모리 전송:100ms	16점 I/O×32대: 10ms(1.5M)	17k바이트 (2kbit+2kw)×32대= 10ms:토큰주기 수루풋21m		17k바이트 (2kbit+2kw)×32대= 33.7ms:토큰주기 32국 유저메모리 전송:77~444ms	4k바이트/10ms
특징	· 오픈 네트워크 후보 · FA용 네트워크 · Ethernet 베이스의 오픈 네트워크 UDP/IP 가 기본 · FA용으로 싸이클릭 전송 코몬메모리 사상을 도입 · 브로드캐스트 각국간 통신	· 오픈 네트워크 후보 · FA용 네트워크 · EN50170 · DINT9245	· 오픈 네트워크 후보 · FA용 네트워크		· 전용네트워크 · FA용네트워크 · 브로드캐스트 통신	· 전용네트워크 · FA용네트워크 · 브로드캐스트 통신
동향	· 자동차 공업회, 통산성의 주선 으로 스타트한 일본 처음의 PLC 간 표준네트워크 · JEMA의 OPCN- 2로써도 채용방향 · 개발 프로젝트 FUJI (리더), 히타치 (서브리더), 옴론, 미쓰비시 샤프, 요꼬가와 파나크, 가와중, 오오크마 등 주요 메이커 18개사	· 회원 약 600사는 일본에서 FUJI, 히타치 등 11개사	· 미국의 PLC 쉐어 35% 의 AB 사가 공개한 네트워크 · 인증체제도 검토중	· 8/3Fast Ether 베이스에 결정 2000년 가을 목표로 사양결정 · 회원수 약 110사 일본에서는 FUJI, 요꼬가와 등 20개사	· NEC, 디지털과 제휴	
추진단체	MSTC	프로피버스 유저 협회	AB	필드버스 협회	미쓰비시 전기	FUJI 전기

ⓑ 하위 네트워크

아래에 PLC하위용 주요 네트워크를 표시합니다.

PLC 하위용 주요 네트워크 일람

항목	OPCN-1	PROFIBUS-DP	DeviceNet	INTERBUS	LonWorks	CAM open	Modbus Plus	Genius Bus	CC-Link	T링크
접속구성	버스	버스	버스	링	버스	버스	버스	버스	버스	버스
거리	전기: 480m(500kbps) 240m(1Mbps)	전기: 200m(1.5Mbps) 100m(12Mbps)	전기: 100m(500kbps) 200m(250kbps) 250m(125kbps)	전기: 국간:400m 총길이:13km	전기: 500m(1.25Mbps) 무선: 100m(4.9kbps)	전기: 5km(10kbps) 1km(50kbps) 100m(500kbps) 50m(800kbps) 25m(1Mbps)	전기: 450m(1Mbps)	전기: 2.3km(38.4kbps) 1.4km(76.8kbps) 1.1km(153.6kbps)	전기: 1.2km(156kbps) 600m(625kbps) 200m(2.5Mbps) 150m(5Mbps) 100m(10Mbps)	전기:1km 광:1km
접속대수 (노드수)	마스터:1대 슬레이브:31대	127대	64대	512대	32,358대(127노드 ×255서브네트)	127대	32대(리미터 부착64대)	32대	64대	35대
전송속도	125k/250k/500k /1Mbps	1.5M/12Mbps	125k/250k/ 500kbps	500kbps	600bps~ 1.25Mbps	10k~1Mbps	1Mbps	38.4k~ 153.6kbps	156k~10Mbps	500kbps
전송매체	트위스트페어 (1P+S)	트위스트페어 (1P+S), 광	전용트위스트페어 (2P+S)	트위스트페어 (1P+S), 광	트위스트페어(1P+S) 무선, 전력선, 광	전용트위스트페어 (2P+S), 광	전용트위스트페어	트위스트페어(1P+S) 또는 Twinax, 광	트위스트페어 (1P+S)	트위스트페어 (1P+S), 광
리프레시 시간	32점I/O×16대 18ms(1M)	32점I/O×30대 2ms(12M)	1점I/O×64대 7ms(500k)	16점I/O×30대 5ms이하	1바이트/7ms (1.25Mbps)			32점I/O×30대 32ms	16점I/O×32대 10ms	16점I/O×32대 10ms
특징	· 오픈네트워크 · PLC하위용 네트워크 · JEM-F3008	· 오픈네트워크 · FA/PA용 네트워크 · FA분야로의 12Mbps추가 · EN50170 · DIN19245	· 오픈네트워크 · FA용네트워크 4선식 케이블에 의한 리모트 급전 · CAN베이스	· 오픈네트워크 · FA용네트워크 · DIN19258 · EN50245	· 오픈네트워크 · 빌딩관리, HA에 강함, FA에도 진출 · ISO의 OSI참조 모델에 준거	· 오픈네트워크 · FA용네트워크 · FA용네트워크 4선식 케이블에 의한 리모트 급전 · CAN베이스	· 오픈네트워크 · FA용네트워크 · 멀티태스크 및 1:1통신	· 전용네트워크 · FA용네트워크 · PLC간 글로벌 데이터통신	· 전용네트워크 · FA용네트워크 · 로컬국간 N:N통신	· 전용네트워크 · FA용네트워크
동향	· 97년 가을에 LSI, 프로토콜 에널라이저 시판,98년 봄 부터 인증제도, 기관발족 · FUJI, 미쓰비시, 옴론, 히타치, 파나크, 가와중, 디지탈, 야마 타케, SMC, CKD 등 20여개 사가 제품화	· 유럽 셰어 NO1. · 회원 약600개사. 일본에서는 FUJI 히타치등 11개 사	· 미국중심, 일본 에는 옴론이 중심이되어 보급활동. · SEMI가 표준 지정. · OMAC(보디 라인)에서 채용 · ODVA회원 약 150개 사 · 일본에는 FUJI, 옴론, 히타치, 토시바, 미쓰 비시, 요코가와, 야스카와 등 약 11개 사.	· 유럽 셰어 NO2. · 자동차 메이커 주체로 납입 실적 큼. · 일본에선 거의 사용실적 없음. · OMAC(엔진 라인)에서 채용 · INTERBUS CLUB11개국, 세계에서 400개 사 이상 · 약700개 사에서 제품가 공급	· 미국중심. · LonWorks약 170개 사, 일본 에서는 히타치, 토시바, 미쓰비 시, 요코가와의 4개 사 · 토시바의 빌관 NetBEE가 LonWorks	· 유럽중심 · 일본에서는 거의 사용실적 없음. · CiA회원 약320개 사	· 유럽중심 · AEG Modicon 주도로 많은 벤더가 지원. · 일본에서는 거의 사용실적 없음	· 미국중심	· 아시아시장에 서의 디펙트화 를 노림 · 파트너 기업 약 60개 사 · 마스터비 공개	· 발매 이래 45만 국의 실적. · 디지털, 엔에스, 디엠, 가켄, SMC, CKD, 코가네이 SS등 약 30개 사가 제품화
추진단체	JEMA	PROFIBUS 유저협회	ODVA	INTERBUS CLUB	LonMark 협회	Can in Automation	Schneider Electric (Modicon)	GE Fanuc	미쓰비시전기	FUJI전기

ⓒ성선화 Network (Bit Network)

아래에 성선화용 주요 네트워크를 표합니다.

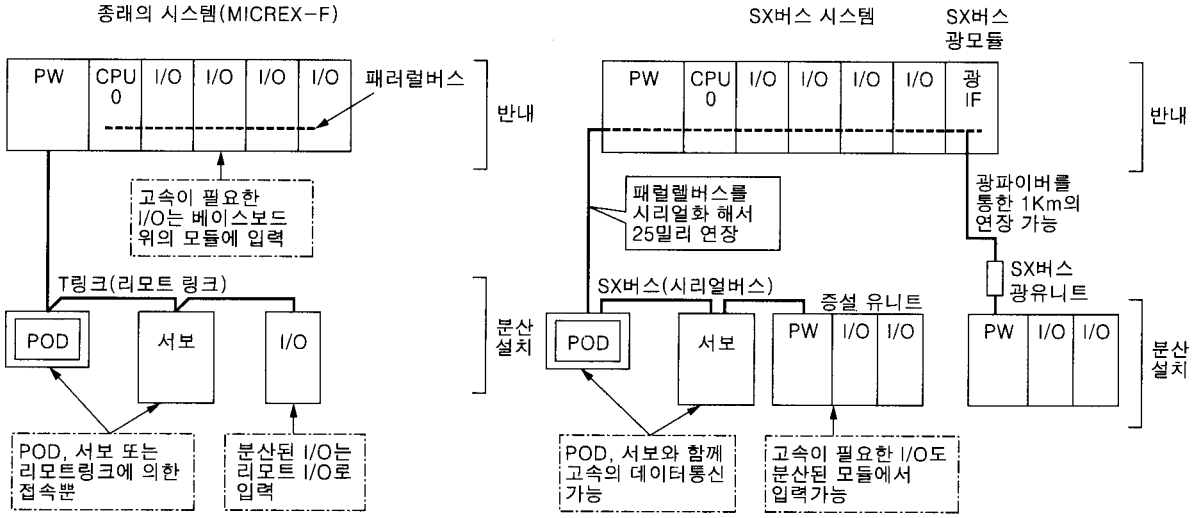
성선화용 주요 네트워크 일람

항목	AS-i	INTERBUS(LOOP)	S-LINK	유니와이어
접속구성	버스	링	버스	버스
거리	전기 : 100m 리피터사용 : 300m	전기 : 국간 10m, 총길이 100m	전기 : 200m	전기 : 200m
접속대수 (노드수)	31 대	64 대	20 대	20 대
전송속도	167kbps	500kbps	28.5kbps	28.5kbps
선송매체	병행전용 케이블 DC24V 전원선 겸용	트위스트페어 (2P+S) DC24V 전원선 겸용	전용 케이블 DC24V 전원선 겸용	전용 케이블 DC24V 전원선 겸용
리프레시 시간	4 점 I/O × 31 대 5ms	1, 000 점 I/O 3.7ms	최대 128 점 4.8ms	최대 128 점 4.8ms
특징	· 오픈 네트워크 · 온오프 기기용 네트 (센서. 액추에이터) · DC24V, 2A 신호선 이용 급전 · 케이블 접속 : 압접 · pr. EN50295	· 오픈 네트워크 · FA 용네트워크 · INTERBUS-S 간선에서 분기하는 지선 루푸로, 단순한 센서 / 액추에이터 레벨의 네트워크 · DIN19258 · EN50444(심사 중)	· 전용네트워크 · FA 용네트워크 · S-Link 와 유니와이어는 접속방식, 전송방식, 지연시간 등 통일. NKE 에서 OEM 공급을 받고 있던 산쿠스가 94 년에 자사품으로 전환, S-Link 로 했다	· 전용네트워크 · FA 용네트워크 · 쿠로다 정공, 요시다 전기, 야스가와 콘트롤에 OEM 공급
동향	· 유럽중심 · AS-i 협회 회원수 약 70개 사, 일본에는 FUJI, 옴론, 미쓰비시 · 이미 상품을 풍부히 갖춤 · 유럽에서는 AS-i 와 INTERBUS(LOOP) 두개가 어깨를 나란히 하다가 지금은 AS-i 가 석권		· 반송, 반도체 제조장치에 강함	· 반도체 제조장치, 자동차 (혼다) 에 강함
추진단체	AS-i 협회	INTERBUS CLUB	산쿠스	NKE 그룹

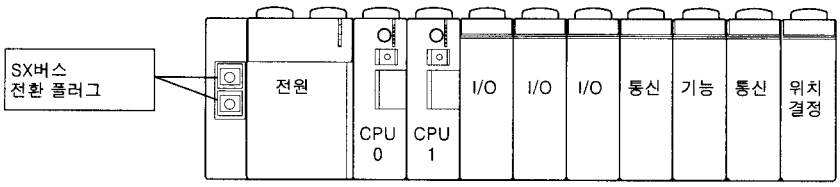
4.2 SX버스란

4.2.1 SX버스의 개요

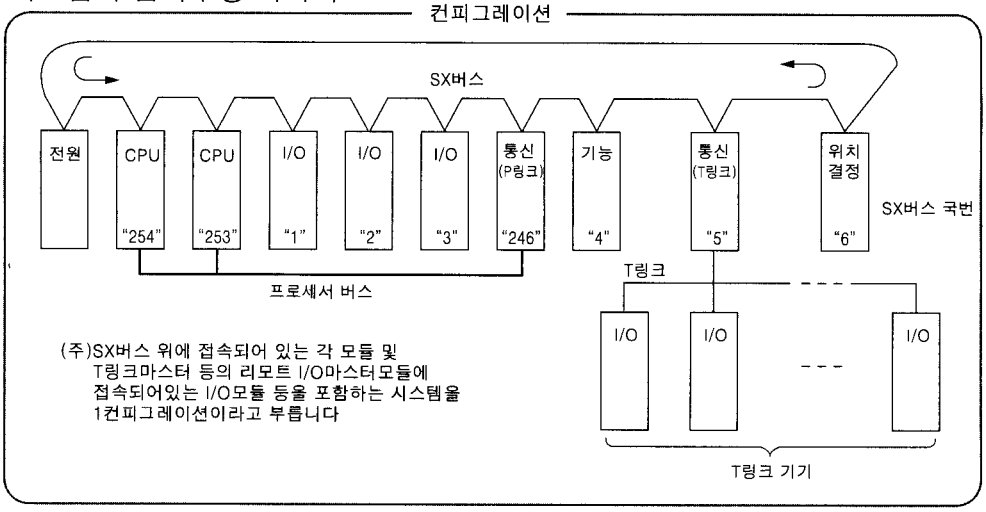
SX버스란, 종전의 패러럴버스를 고속 시리얼 통신하는 것으로 해서, 고속의 I/O리플레시를 유지해 나가면서 각종 모듈의 분산설치를 실현시키는 것입니다. 종전의 시스템과 SX버스 시스템의 비교를 아래에 제시합니다.



베이스보드 위의 SX버스시스템의 접속이미지를 아래에 표시합니다.



시스템의 접속구성 이미지



베이스보드 내부에는, SX버스라고 불리는 신호선이 있어서, 베이스보드 위의 모든 모듈은 SX버스에 접속되어 있습니다. SX버스에 접속되어 있는 각 모듈(전원모듈을 제외한)은 그 각각의 모듈들에게 SX버스 국번을 할당하고 있습니다. 더욱이 프로세서버스라는 신호선도 내장되어 있어, CPU간 그리고 CPU-P/PE 링크간의 고속통신에 사용합니다.

SX버스란

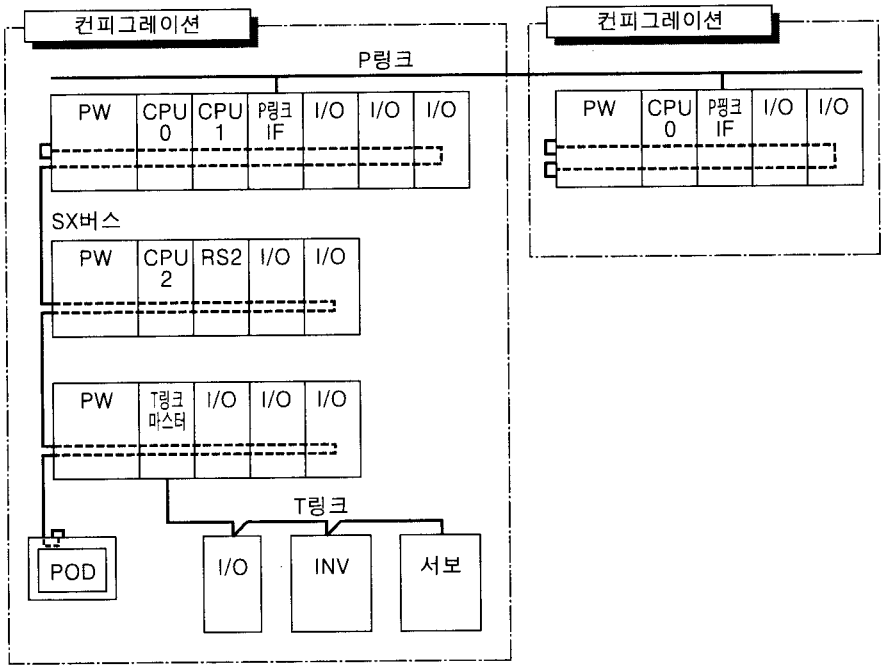
전송속도 25Mbps, 총 연장거리25m, 최대접속국수 254국의 MICREX-SX시리즈 전용 고속 데이터 버스입니다. 앞 페이지의 SX버스 저속 이미지 그림처럼 루푸 구조를 하고 있습니다. 따라서 SX버스의 양끝(베이스보드)에 SX버스 전환 플러그를 장착할 필요가 있습니다.

프로세스버스란

전송속도 25Mbps(버스8본), 한 베이스보드 위의 CPU모듈 및 P/PE링크모듈에 접속되어있는 고속 데이터버스입니다. 같은 컨피그레이션에 속해 있어도 다른 베이스보드 위의 CPU, P/PE링크 모듈에는 접속되지 않습니다. CPU-CPU간, CPU-P/PE 링크간의 데이터 통신에 사용합니다.

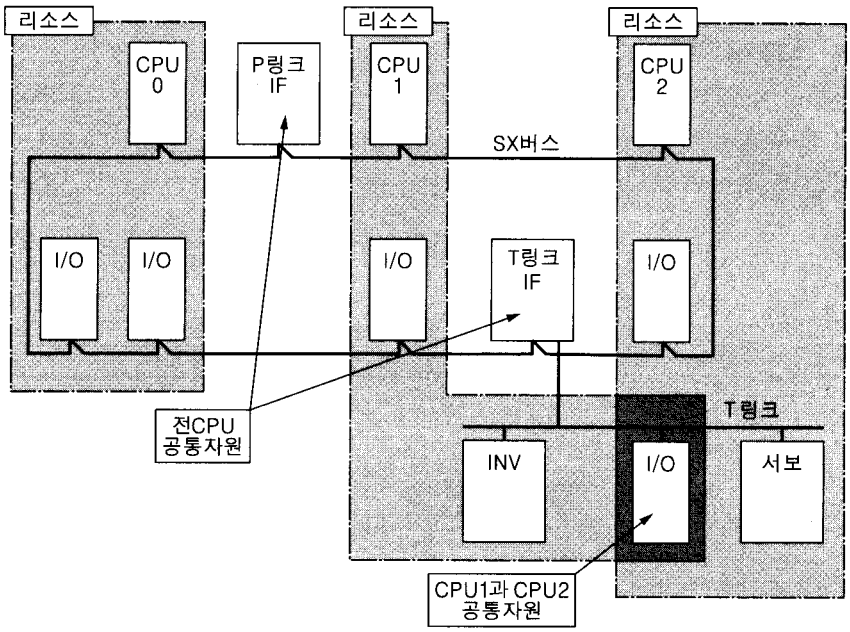
컨피그레이션이란

SX버스에 직접 접속되어 있는 각종 모듈 및 리모트 I/O마스터(T링크마스터, JPCN-1마스터 등)에 접속되는 슬레브와 I/O모듈의 집합체입니다. 1개의 SX버스에 접속되는 PLC시스템이 1개의 컨피그레이션에 해당합니다.



리소스란

1대의 CPU와 거기에 할당된 자원(각종 모듈, 프로그램 등)의 집합체입니다. 리소스는 프로그램을 실행하는 처리기능에 해당하고, 협의의 의미는 리소스=CPU라고 생각할 수 있습니다.



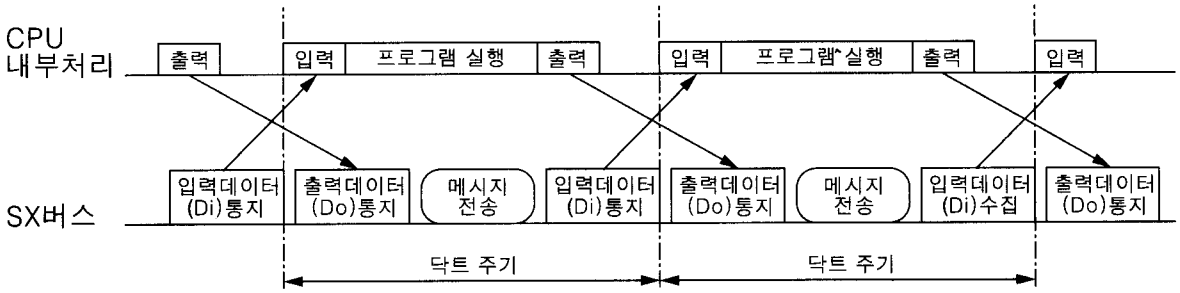
4.2.2 SX버스의 사양

항목	사양	비고
전송속도	25Mbps	
총 길이	25m	
접속형태	버스형(물리적으론 링모양)	T형 분기 유니트에 의해 트리 구조도 가능
송신권 제어방식	토큰패스 및 독자방식	
전송데이터	· 프로세스 데이터 입력 : 토탈프레임방식 출력 : 멀티캐스터방식 · 프로세서간 데이터 멀티캐스터/메시지방식	
최대접속 노드수	254대	마스터국 포함
최대접속 마스터 국	16대	실제로 접속 가능한 CPU대수는 8대

4.2.3 SX버스의 데이터 전송방식

SX버스의 데이터 전송은 아래의 표시처럼, 입력데이터(Di)의 수집과 출력데이터(Do)의 통지 및 메시지 전송으로 구성됩니다. SX버스 위에는, 이 I/O리프레시 처리가 주기적으로 되풀이됩니다. 이 주기를 탁트 주기라고 합니다. CPU의 내부처리에 있는 데이터입력→프로그램실행→데이터출력은 이 SX버스의 탁트 주기의 같은 시기에 처리됩니다.

입력데이터의 수집과 출력데이터의 통지는 1탁트 주기 내에서 모든 처리가 실행됩니다만, 메시지전송은 입력데이터의 수집과 출력데이터의 통지가 종료한, 남은 시간에 처리됩니다. 메시지전송은 1 : 1통신입니다. 메시지전송의 데이터로써는, 대 프로그래밍 툴, 대 POD데이터 등이 있습니다.



4.2.4 SX버스의 탁트 주기

SX버스의 전송주기(I/O리프레시)를 탁트 주기라고 부르고, 그 시간은 시스템구성에 의존합니다. 탁트 주기에 관련되는 시스템구성요소는 아래와 같습니다.

(1) I/O점수
(2) CPU대수(SPH300에 한함)
(3) 프로세서링크 대수(P/PE링크, FL-net)
(4) 리모트I/O마스터 대수
(5) 통신 모듈 대수

SPH200의 경우, 로더커맨드 동시 발행국 대수가 증가하면 프로그램을 실행하는 시간이 저하됩니다. (참고치는 매뉴얼 FEH200 INSTRUCTION Appendix3을 참조)로더커맨드 동시 발행국에는, 프로그래밍 툴, POD, 메시지 관련명령에 대응하고 있는 모듈(범용통신 모듈 등)이 있습니다.

탁트 주기는, 정주기의 기동 타이밍(탁트 주기의 정수배)과 이벤트태스크의 이벤트 검출에도 관계(이벤트 탁트 각 주기에 체크된다)하기 때문에, 정주기와 이벤트(사이에 들어감)를 사용하는 경우에는, 시스템의 제어주기를 고려한 후에, 설정할 필요가 있습니다. 탁트 주기는, 0.5(SPH300에 한함), 1, 1.5, 2, 2.5, 3, …, 18, 18.5, 19, 19.5, 20ms(0.5ms단위)에서 선택 설정합니다.

탁트 주기의 개략치를 참고적으로 아래에 표시합니다.
탁트 주기시간의 상세 기간에 대해서는, User's manual FEH200 Appendix3의 산출식으로 산출합니다.

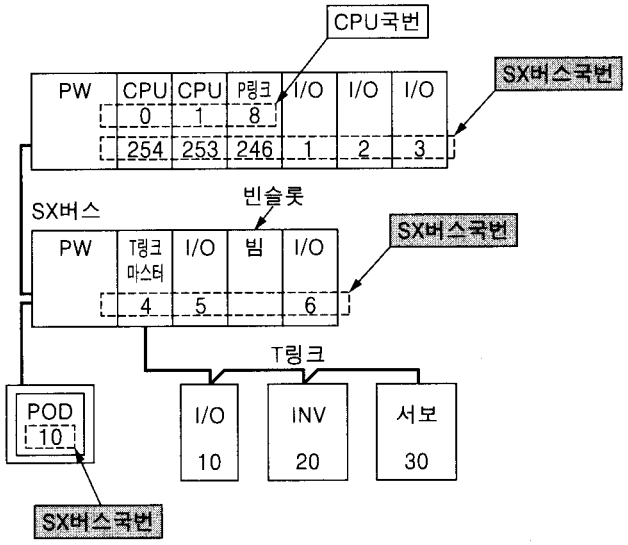
단위ms

접속 I/O 점수(점)	0	32	128	256	512	1024	2048	3072	4096	6144	8192
CPU1대	0.418	0.504	0.507	0.510	0.56	0.70	1.04	1.39	1.52	1.72	1.92
		1	1	1	1	1	1.5	1.5	2	2	2
CPU2대	0.84	0.93	0.93	0.93	0.98	1.12	1.47	1.98	2.11	2.31	2.51
	1	1	1	1	1	1.5	1.5	2	2.5	2.5	3
CPU3대	1.05	1.14	1.14	1.14	1.19	1.33	1.68	2.18	2.31	2.51	2.71
	1.5	1.5	1.5	1.5	1.5	1.5	2	2.5	2.5	3	3
CPU1대 리모트마스터 1대	1.10	1.19	1.19	1.19	1.24	1.38	1.73	2.40	2.53	2.73	
	1.5	1.5	1.5	1.5	1.5	1.5	2	2.5	3	3	
CPU2대 리모트마스터 1대	1.70	1.79	1.79	1.79	1.84	1.98	2.33	2.80	2.93	3.13	
	2	2	2	2	2	2	2.5	3	3	3.5	
CPU3대 리모트마스터 1대	2.04	2.13	2.13	2.13	2.18	2.32	2.67	3.21	3.34	3.53	
	2.5	2.5	2.5	2.5	2.5	2.5	3	3.5	3.5	4	
CPU1대 리모트마스터 1대 통신모듈 1대	1.23	1.32	1.32	1.32	1.37	1.51	1.85	2.53	2.66	2.85	
	1.5	1.5	1.5	1.5	1.5	2	2	3	3	3	
CPU2대 리모트마스터 1대 통신모듈 1대	1.83	1.92	1.92	1.92	1.97	2.11	2.45	2.93	3.06	3.25	
	2	2	2	2	2	2.5	2.5	3	3.5	3.5	
CPU3대 리모트마스터 1대 통신모듈 1대	2.17	2.26	2.26	2.26	2.31	2.45	2.79	3.34	3.47	3.66	
	2.5	2.5	2.5	2.5	2.5	2.5	3	3.5	3.5	4	
CPU8대 리모트마스터 3대 통신모듈 16대	6.19	6.28	6.28	6.28	6.33	6.47	6.81				
	6.5	6.5	6.5	6.5	6.5	6.5	7				

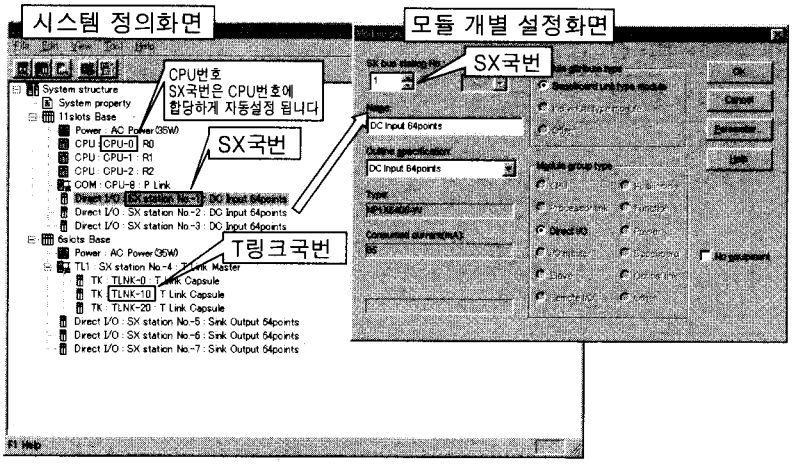
(주) 1. 표 가운데 하단의 수치는 탁트 주기의 설정치를 나타냅니다.
2. 직결I/O점수로는, SX버스에 직접 접속되는 I/O점수입니다. 리모트마스터에 접속되는 I/O점수는 2048점(128W)/1ch(회선)으로써 계산하고 있습니다. 따라서 리모트마스터에 접속되는 I/O점수는 가산할 필요가 없습니다.
3. AS-i마스터 모듈만은 특수하게, 위의 표의 리모트마스터 대수로써 가산할 필요는 없습니다. 따라서 AS-i마스터에게 접속되는 I/O는 직결 I/O로써 계산합니다.
4. I/O점수는 입력과 출력의 점수비가 1:1로 계산하고 있습니다.
출력점수가 증가하면 탁트 주기의 시간은 증가하고, 출력점수가 작아지면 감소합니다. 그 변동 범위는 약 20%입니다. 따라서 탁트 주기 0.5ms의 가능성은 CPU 1대, I/O점수 256점 이하, 리모트마스터 없이, 통신 모듈 없이, I/O점수와 사용 I/O모듈의 종류에 의존합니다. 상세 계산식은 별지를 참조해 주십시오.
5. CPU 대수, 리모트마스터 대수, 통신 모듈 대수의 증가에 대해서는 하기의 개략치를 가산합니다.
CPU 대수 : 0.41ms/1대
리모트마스터 대수 : 0.28ms/1대
통신 모듈 대수 : 0.13ms/1대

4.2.5 SX버스 국번

SX버스에 접속되는 CPU 및 모듈에는, SX버스 국번을 설정할 필요가 있습니다.
전원 모듈과 리모트 I/O마스터(T링크마스터 등)에 접속되는 리모트의 I/O 모듈군은 SX버스 국번을 설정할 필요가 없습니다. 단, 리모트 I/O마스터에 접속되는 I/O 모듈군에는, 종전과 마찬가지로, 리모트 국번(예를 들어 T링크 국번)을 설정할 필요가 있습니다.
SX국번은, 하드에 국번 설정 SW를 가지고 있는 것(예를 들어 POD, 서보 등)과 가지고 있는 않는 것(그 외의 모듈 등)이 있습니다. 어쨌든 프로그래밍 툴의 시스템 정의로써 국번의 설정을 할 필요가 있습니다. SX국번은 각 모듈에 대해서 자유롭게 설정할 수가 있습니다만, 입출력 어드레스의 표현과 관련되기 때문에, 기본적으로는 CPU의 실제 설치위치를 기점으로 해서, 순번대로 설정해야만 합니다. (2.2항「입출력 어드레스의 표현」을 참조) 또한, 베이스보드 위에는 빈 슬롯을 만들 수가 있습니다.
아래는 SX버스의 국번 설정의 예를 표시한 것입니다.



아래는 D300win에 대한 시스템 정의의 설정 예입니다.



국번 설정의 기본은 다음과 같습니다.

(1) 국번은 CPU를 기점으로 해서 순번대로 설정합니다.

(2) T링크처럼 모듈의 점유 어드레스를 고려할 필요 없이, 순번 설정합니다.

(3) 번호는 생략해서 설정하는 것도 가능하지만, T링크처럼 하드의 국번설정 SW가 없기 때문에, 실제의 시스템에서 어드레스를 찾는 경우를 고려해서 순번 설정을 추천합니다.
설정 범위는 1~238번

(4) CPU와 P/PE링크, FL-net 모듈은 전면의 로터리 SW로 설정하는 CPU번호에 맞추어 아래처럼 SX국번이 이미 할당되어 있습니다. 따라서, 로터리 SW의 설정에 의해서 SX국번은 자동적으로 할당됩니다.

CPU번호	SX국번	대상모듈
0	254	CPU
1	253	CPU
2	252	CPU
3	251	CPU
4	250	CPU
5	249	CPU
6	248	CPU
7	247	CPU
8	246	P/PE링크, FL-net
9	245	P/PE링크, FL-net
A	244	미사용
B	243	미사용
C	242	미사용
D	241	미사용
E	240	미사용
F	239	미사용

0~7번 CPU용, 8, 9P/PE링크, FL-net용

4.2.6 접속 대수의 제약

(1) 1 컨피그레이션에 접속할 수 있는 모듈 수

1 컨피그레이션에 접속할 수 있는 모듈의 수는 다음과 같습니다.

	모듈	최대접속대수		
전원	AC100/240V전원 (NP1S-22) DC24V전원 (NP1S-42)	무제한		
CPU	SPH200 (NP1PH-08/16)	1대		
	SPH300 (NP1PS-32/74)	8대 (SPH200와의 혼재 불가)		
통신모듈	P/PE링크모듈 (NP1L-PL1/PE1) FL-net모듈 (NP1L-FL1)	2대	16대	
	범용통신모듈 (NP1L-RS1/2/4) 메모리카드 I/F모듈 (NP1F-MM1)	16대		238대
	PC카드 I/F모듈 (NP1F-PC2)	4대		
리모트마스터 모듈	T링크 마스터모듈 (NP1L-TL1) OPCN-1마스터모듈 (NP1L-JP1)	8대		
	AS-i 마스터모듈 (NP1L-AS1)	19대	(512W)	
I/O관련모듈	I/O모듈 I/O터미널 아날로그 모듈 위치결정 모듈 고속카운터 모듈 T링크접속기기 JPCN-1접속기기 AS-i 접속기기	238대		
베이스보드	베이스보드 SX버스 T분기 유니트	25대		

(2) 모듈의 소비전류에 의한 실제 장치 대수의 제약

베이스보드 위에 모듈을 실장하는 경우에는, 전원 모듈의 출력전류에 대한 모듈의 소비전류를 검토할 필요가 없습니다.

산출 예는 아래와 같습니다. 단, 각 모듈의 소비전류는 메뉴얼의 하드편을 참조하십시오.

PW	CPU 0	CPU 1	P링크 IF	Di 64	Di 64	Do 64

전원: NP1S-22(출력전류 1.46A)
CPU: NP1PS-32(소비전류 200mA)
베이스보드: NP1BS-08(소비전류 50mA)
P링크: NP1L-PL1(소비전류 160mA)
Di모듈: NP1X6406-W(소비전류 85mA)
Do모듈: NP1Y64T09P1(소비전류 90mA)

200×2+160+85×2+90+50=870mA<1.46A · · · 문제없음

4.3 SX시리즈의 모듈

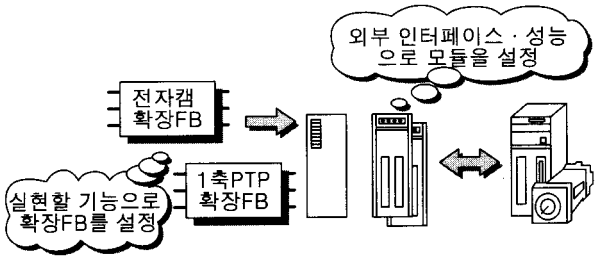
4.3.1 모듈의 장점

MICREX-SX SPH 시리즈는, 하드웨어의 멀티 기능화를 실현했습니다.

· 아날로그 모듈은 멀티레인지 대응
· 펄스캐치 기능은, 고속 입력 모듈에 내장
· 24V 입력 모듈은, 싱크 · 소스 쌍방향 입력에 대응

기능의 소프트화에 의해, 기능 모듈의 종류는 외부 인터페이스의 종류가 되었습니다.

· 위치 결정 모듈은, 펄스열 출력과 아날로그 출력의 2ch/1ch로 집약
· 범용 통신의 기능은, 확장 FB로 집약
· PID 제어와 외부 고장감시기능은, 확장FB로써 소프트화



4.3.2 모듈 일람

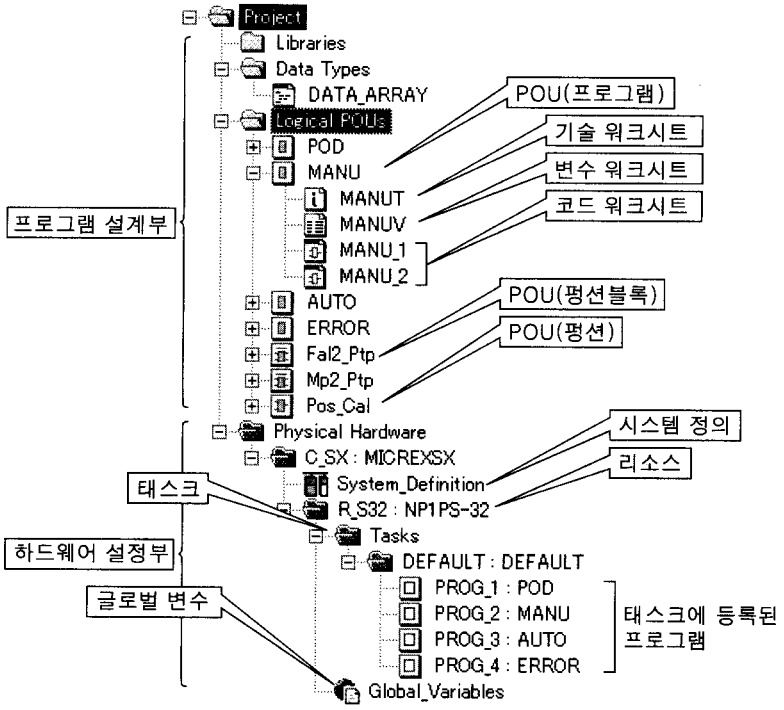
1. 직결I/O 그룹	입력모듈	DC24V입력16점, DC24입력32점, DC24V입력64점, 고속DC입력32점 AC100V입력8점, AC100V입력16점, AC200V입력8점, TTL입력32점
	출력모듈	싱크출력8점, 싱크출력16점, 싱크출력32점, 싱크출력64점, 소스출력8점, 소스출력16점, 소스출력32점, 소스출력64점 SSR출력6범, SSR출력8점, RY출력8점, RY출력16점, 펄스싱크출력32점
	입출력 혼합 모듈	싱크혼합16점, 싱크혼합32점, 소스혼합16점, 소스혼합32점
	아날로그 모듈	고속아날로그입력4cH, 표준아날로그입력4cH, 표준아날로그입력8cH, 고속아날로그출력2cH, 표준아날로그출력2cH
2. I/O마스터 그룹		T링크마스터, OPCN-1마스터, AS-i마스터, DeviceNet마스터, T링크 슬레이브
3. 위치결정모듈 그룹		고속카운터2cH, 고속카운터8cH, 펄스출력, 아날로그 복합 위치결정2cH, 펄스복합위치결정2cH
4. 기능모듈 그룹		PC카드 통신모듈, 메모리카드 통신모듈, 범용통신모듈1(RS232C/485), 범용통신모듈2(RS232C) 범용통신모듈3(RS-485)
5. 프로세서 링크 그룹		P링크모듈, PE링크모듈, FL-net모듈

4.4 프로그램의 개발

4.4.1 프로그램의 관리

(1) 프로젝트

MICREX-SX의 프로그램은 프로젝트라는 단위로 관리됩니다. 구조화 설계를 지원하는 프로젝트는 D300win상에서는, 다음과 같은 나무 구조로 표현됩니다.



프로젝트 트리

프로젝트 트리는 프로그램 설계부와 하드웨어 설정부의 두 개의 부분으로 구성되어 있습니다.
프로그램 설정부는, 다음의 부분으로 구성되어 있습니다.

① 라이브러리 : 기존의 프로젝트를 재 이용하기 위해서 등록하는 폴더
② 데이터 형 : 유저가 새롭게 작성한 데이터 형(파생 데이터 형)을 등록하는 폴더
③ 프로그램 구성 : 프로그램, FB, FCT를 설계 · 등록하는 폴더

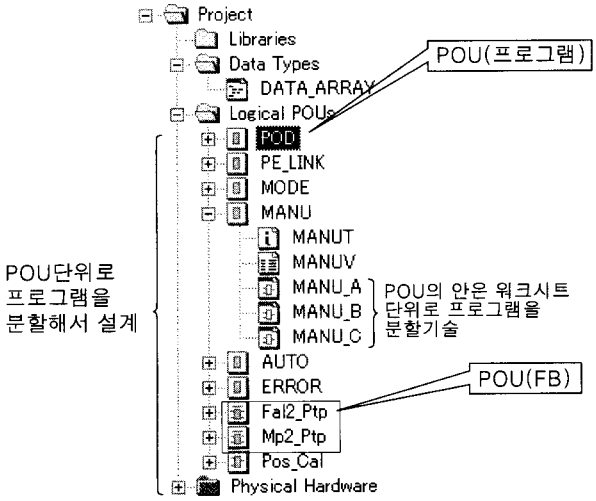
프로그램 구성에는, 프로그램, FB, FCT가 POU라는 단위로 등록되어 있습니다. POU는, POU에 관한 기술을 행하는 기술 워크시트, 변수선언을 행하는 변수 워크시트, 프로그램을 기술하는 코드 워크시트로 구성되어 있습니다. 각 워크시트는, 여러 장을 작성할 수가 있으며, 제어 내용에 맞춘 분할 설계(종래의 페이지관리에 상당)가 가능하도록 되어있습니다.

하드웨어 설정부는, 다음의 부분으로 구성되어 있습니다.

① System_Definition(시스템 정의) : SX버스 탁트 주기의 설정, 혈장화의 설정, 각 CPU의 I/O그룹 설정, CPU와 모듈의 구성등록 등을 합니다.
② 리소스 : CPU의 설정(동작 정의, 메모리 사이즈 정의 등)폴더
③ Tasks(태스크) : 프로그램을 태스크에 등록하는 폴더
④ Global_Variables(글로벌 변수 워크시트) : 글로벌 변수의 선언을 행합니다. 여러 장의 워크시트

(2) 프로그램 구성단위(POU)

프로그램을 구조화 해나갈 때의 프로그램의 분할 단위를 POU(Progam Oraganization Unit)라고 합니다. 펑션블록, 펑션도 POU단위로 설계됩니다. 결국, POU란 프로그램, FB, FCT인 것입니다.
POU는 로컬 변수가 유효한 범위에 있고, 프로그램의 재 이용 단위에도 있습니다. 1개의 POU는 게다가, 몇 개의 워크시트로 분할해서 프로그램을 기술할 수가 있도록 되어 있습니다. 프로그래밍 툴에 있어서 POU의 분할 예입니다.



(3) 프로젝트 트리를 구성하는 요소 명칭의 제어

프로젝트 구조(트리)를 편집(요소를 추가, 삭제 등)하는 경우의 각 요소의 명칭의 제약 사항에 대해서 설명합니다.

① 문자수의 제약

8문자(반각환산)이내에서 지정하는 요소

반각 : 8문자이내, 혹은 전각 : 4문자 이내(반각, 전각의 혼재도 가능)에서 지정하는 요소는 다음과 같습니다.

- 프로젝트 이름
- POU 이름
- 컨피겨레이션 이름
- 리소스 이름
- 프로그램 인스턴스 이름

24문자(반각환산)이내에서 지정하는 요소

반각 : 24문자이내, 혹은 전각:12문자 이내(반각, 전각의 혼재도 가능)에서 지정하는 요소는 다음과 같습니다.

- 데이터 형 워크시트 이름
- 기술 워크시트 이름
- 변수 워크시트 이름
- 글로벌 변수 워크시트 이름(반각문자만 지정이 됩니다. 전각 문자 및 반각 가타카나는 사용할 수 없습니다).

7문자(반각환산)이내에서 지정하는 요소

반각 영수자 : 7문자 이내(전각문자 및 반각 가타카나는 사용불가)에서 지정하는 요소는 다음과 같습니다.

· 태스크 이름

② 사용할 수 없는 문자

프로젝트 트리의 모든 요소에 있어, 다음에 제시하는 문자는 사용할 수 없으므로 주의해 주시기 바랍니다.

¥ / : . , * ? < > | 및 시스템 예약어(예 R, C, POE, COM, PROGRAM 등)

■ 시스템 예약어에 대해서는 User's manual FEH200 Appendix6을 참조바랍니다.

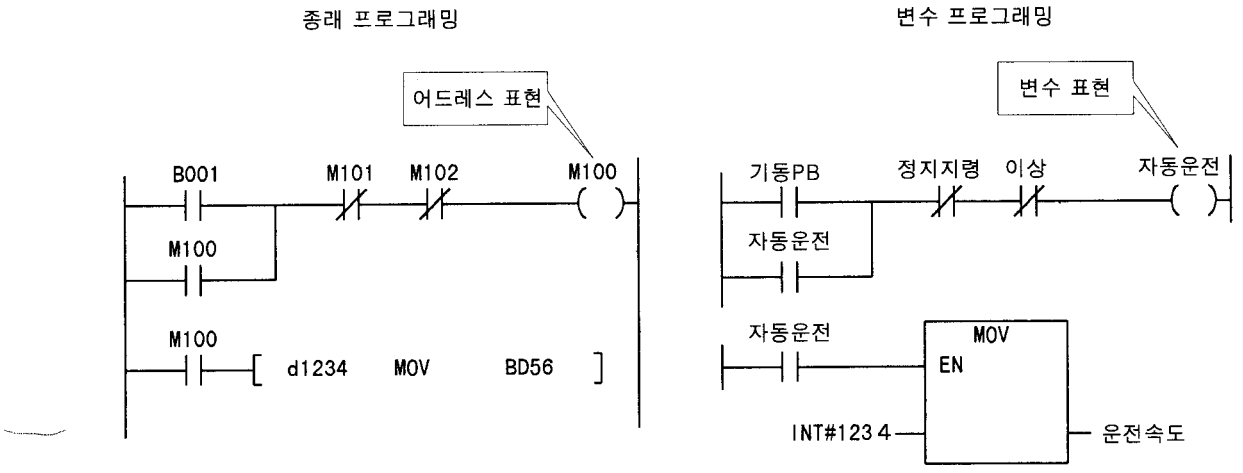
③ 변수 이름의 제약사항 및 주의 사항

· 변수이름에 사용가능한 문자는 영숫자만 입니다.
· 변수 이름에 사용 가능한 기호는 "_"뿐입니다. 단, "_"의 연속 사용"_ _"은 안됩니다.
· 변수 이름의 선두에 반각숫자를 사용할 수는 없습니다.
· 변수 이름의 길이는 반각으로 30문자, 전각으로 15문자입니다.
· 시스템에 예약해 있는 예약어는 사용할 수 없습니다.
  예약어에 관해서는 User' s manual FEH200 Appendix6을 참조해 주십시오
· 변수 이름은, 영문자의 대문자와 소문자는 구별되지 않습니다. 따라서, "abcd", "ABCD", "aBCd"는 동일 변수로 봅니다.
· 변수 이름에 공백(스페이스)을 사용할 수는 없습니다.

4.4.2 변수

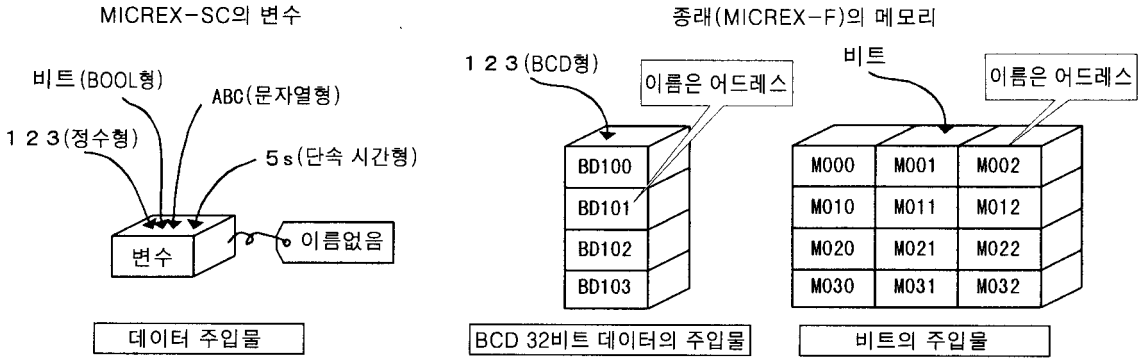
(1) 변수 프로그래밍의 개요

종래의 PLC프로그래밍은, 어드레스(IEC에서 말하는 직접표현변수)에 의한 프로그래밍입니다.
어드레스의 번호와 실제의 제어의 사이에는 어떠한 인과관계도 없습니다. (예를 들어 M100은 프로그램 설계자가 자동운전으로 드물게 배분한 것일 뿐입니다) 하지만, 프로그램 설계자에게는 어드레스가 중복되지 않도록 항상 의식하고 설계하는 것이 요구됩니다. 더욱이 이 어드레스 번호는 이 애플리케이션 고유의 번호이며, 다른 애플리케이션에 이용할 때에는, 중복을 피하도록 어드레스 번호의 재 배분을 해야할 필요가 있습니다. 그렇기 때문에, 어드레스에 의한 프로그래밍은 오랫동안, PLC프로그램의 가독성과 재 이용을 방해하는 커다란 요인이었습니다. 변수 프로그래밍이란 것은, 어드레스의 대신으로 변수를 이용해서 프로그래밍하는 방법으로, 변수 이름은 프로그래밍 설계자가 자신에게 최적인 이름으로 붙일 수 있습니다.
아래에는, 종래의 프로그래밍과 변수프로그래밍의 예를 제시한 것입니다.



(2) 변수란

변수란, 프로그램 속에서 사용하는 데이터의 주입물입니다. 종래의 어드레스 프로그래밍의 경우, 예를 들어, M100은 비트, BD56은 32비트의 BCD라고 한 것처럼 어드레스는 PLC로 결정되어지는 형태를 가지고 있습니다. 따라서 어드레스를 사용하는 경우에 그 형태를 지정(선언)할 필요가 없습니다. 하지만, 변수는 설계자가 자유롭게 명명해서 프로그램 속에서 사용하기 위해서, 그 변수에 격납할 데이터 형태를 선언할 필요가 있습니다.



변수의 메모리 상의 어드레스는 프로그래밍 툴이 자동 배분하기 때문에, 프로그램 설계자가 특별히 어드레스를 의식할 필요가 없습니다. 그러나 외부기기(예를 들어, 조작 표시기와 퍼스널컴퓨터 등)와 어드레스를 끼워 통신을 할 경우, 또한 컴파일 시에 어드레스가 변해서는 곤란할 경우에는, 변수를 의식적으로 고정한 어드레스에 배분할 수가 있습니다.

변수에 의한 프로그래밍의 메리트는 다음과 같습니다.
① 프로그램 설계자는 PLC 고유, 또는 애플리케이션 고유의 물리적인 메모리 관리에서 개방됩니다.
② 변수에는 최적의 이름이 붙을 수 있기 때문에, 프로그램의 가독성이 좋아집니다.
③ PLC와 애플리케이션 고유의 메모리 관리가 필요 없기 때문에, 프로그램의 재 이용성이 향상됩니다.

변수에 의한 프로그래밍의 메리트로써, 하나 하나의 변수 이름을 붙이는 것의 번거로움을 없앨 수 있습니다. 하지만 종래에도 사용한 어드레스에 코멘트를 붙이는 것, 또한 변수 이름을 붙이는 규칙을 어느 정도 결정해 두고, 기계적으로 명명할 수 있도록 해 두면, 변수를 사용하는 메리트는 더욱 커집니다.

(3) 변수의 선언

변수는 프로그램 설계자가 변수의 이름과 그 변수에 넣는 데이터의 형을 결정하여 사용합니다. 따라서, 변수를 사용하는 경우에는, 변수 이름과 데이터형을 선언할 필요가 있습니다. 변수는 특히 어드레스를 지정하지 않아도, 컴파일 시 자동적으로 메모리에 배분됩니다. 하지만 프로그램 설계자가 의식적으로, 특정의 메모리에 변수를 배분하는 것도 가능합니다. 또한 변수는 PLC의 전원 투입 시와 리세트 시에, 데이터형의 디폴트값에 세트됩니다만, 특정의 초기치를 부여할 수도 있습니다. 변수선언의 기본형은, 다음과 같습니다.

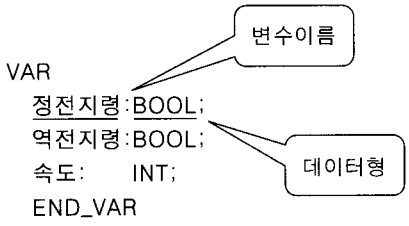
```
[변수이름](AT[메모리 어드레스]) : [데이터 형](:=[초기치]) ;
```

( )은 생략가능하며, 생략한 경우, 메모리 어드레스는 자동배분, 초기치는 데이터형의 디폴트값이 됩니다.

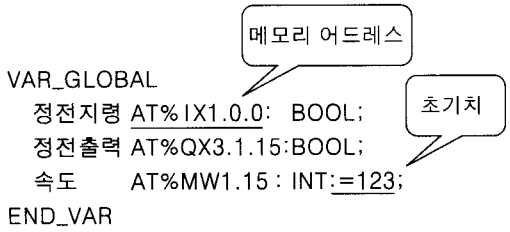
변수는 변수 이름과 데이터형을, 변수 선언의 개시와 종류를 표시하는 키워드인 VAR과 END_VAR로 묶어서 선언합니다. 변수의 종류는 변수 선언의 개시를 표시하는 키워드 VAR(로컬 변수), VAR_GLOBAL(글로벌 변수), VAR_EXTERNAL(외부 변수)등으로 구별됩니다. 변수의 종료를 표시하는 키워드 모두 END_VAR입니다.

아래에 변수의 선언 예를 표시합니다.

로컬 변수의 선언 예

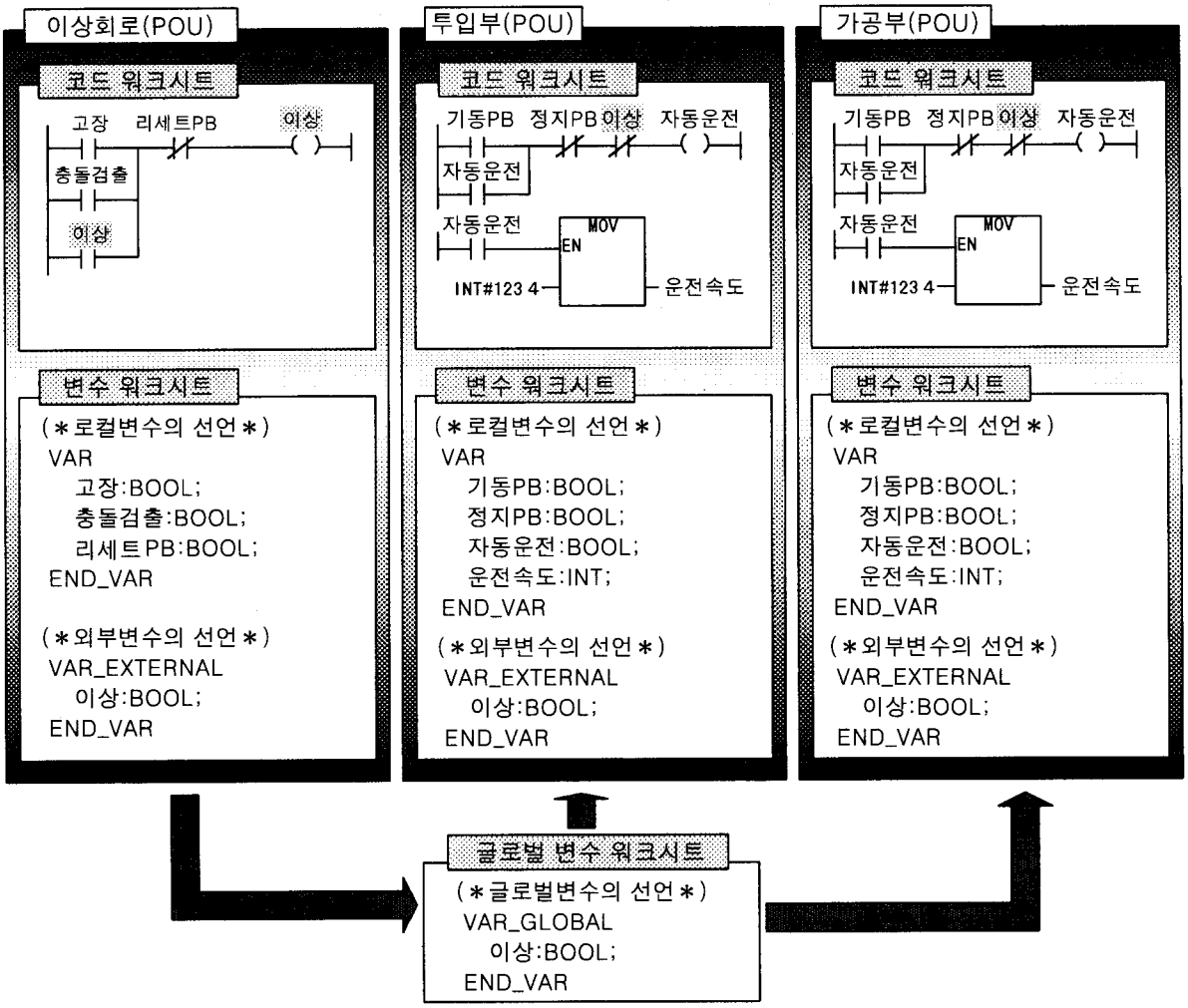


글로벌 변수의 선언 예



(4) 로컬 변수와 글로벌 변수

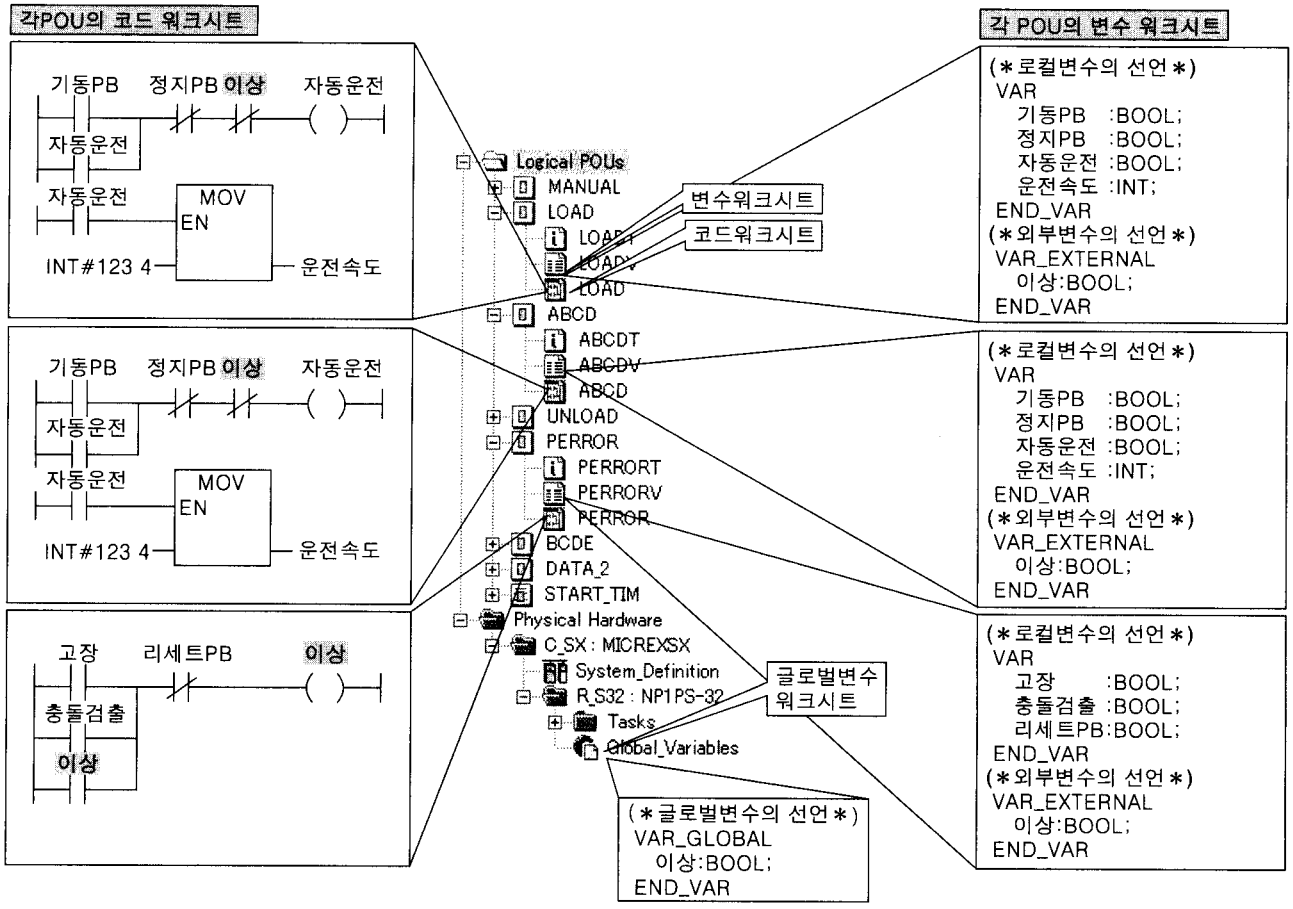
프로그램으로 사용 가능한 변수는 로컬 변수와 글로벌 변수입니다. 로컬 변수는 POU내에서만 유효한 변수이고, 글로벌 변수는 리소스 내의 모든 POU에서 액세스 가능한 변수입니다. 글로벌 변수를 POU 내에서 사용할 때에는, 그 변수를 POU의 변수 워크시트로 외부변수로써 선언할 필요가 있습니다.
로컬 변수와 글로벌 변수의 적용범위를 아래의 표에 표시했습니다.
「이상」은 모든 POU에서 사용 가능한 글로벌 변수로 투입부 프로그램과 가공부 프로그램에서는 글로벌 변수 「이상」을 POU내에서 사용할 수 있도록 외부변수로 해서 선언을 하고 있습니다. 따라서 이상 회로의 「이상」에 의해, 투입부와 가공부의 자동운전은 중지합니다. 하지만, 「기동PB」,「정지PB」,「자동운전」,「운전속도」은 각 POU내에서만 유효한 로컬 변수입니다. 따라서, 투입부의 자동운전은, 가공부의 자동운전에 아무런 영향도 주지 않습니다. 반대도 역시 마찬가지입니다.



로컬 변수와 글로벌 변수의 적용범위

(5) 로컬 변수와 글로벌 변수의 선언방법

변수를 사용하는 경우에는, 변수의 이름과 데이터형을 지정하기 위해서, 변수선언을 해야할 필요가 있습니다. 그 예를 전 항에서 설명한 내용을 기본으로 실제의 프로젝트의 가운데 어떻게 선언할 것인가를 설명합니다. 각 POU 내에서만 유효한 로컬 변수는 각 POU의 변수 워크시트에 있어서, VAR과 END-VAR의 구문을 이용하여 선언됩니다. 글로벌 변수는 각 리소스(각 CPU)마다의 글로벌 변수 워크시트에 VAR-GLOBAL과 END-VAR의 구문을 이용하여 선언됩니다. 선언된 글로벌 변수의 안에서, POU로 사용하는 것은, 각 POU의 변수 워크시트에서 VAR-EXTERNAL과 END-VAR의 구문을 이용해 외부변수로써 선언합니다. 당연히, 글로벌 변수와 외부변수는 같은 데이터형입니다. 이것에 의해, 글로벌 변수가 POU안에서 사용 가능(액세스 가능)하게 됩니다.



로컬 변수와 글로벌 변수의 선언

(6) 로컬 변수와 글로벌 변수의 이용구분

로컬 변수의 적용범위는 각 POU의 내부뿐이며, POU단위로 구조화를 도모하고, 독립성을 높일 수가 있습니다. 따라서, 변수는 될 수 있는 한 로컬 변수로써, 프로그램의 독립성을 높이고, 수정 · 변경에 의한 영향을 최소한으로 해야 합니다. 로컬 변수는 POU 내부에서만 변수 이름에 중복이 없도록 주의해 주십시오. 이 로컬 변수에 의해, 여러 사람의 프로그램 병렬설계와 프로그램의 표준화, 재 이용을 추진할 수가 있습니다.
단, 프로그램으로써의 POU는 외부변수에 의해 로컬 변수를 받아들이기 때문에, 어떻게 해도 리소스와의 관련이 남습니다. 진정한 의미로의 독립성과 그것에 의한 표준화, 재 이용은 후장에서 설명하는 펑션블록(FB)과 펑션(FCT)에 의해 실현할 수가 있습니다.

글로벌 변수는 각 POU 사이에서 통신을 행하기 위한 공통변수입니다. 따라서, 다음과 같은 변수가 글로벌 변수라고 생각되어질 수 있습니다.
① 각 POU 사이에서 공통으로 사용되는 변수(상시 ON변수, 램프 점멸용 플리커 변수 등)
② 모드 관련의 변수(자동 모드 변수, 수동 모드 변수 등)
③ 이상 관련의 변수(INV 1이상 변수, 중고장 변수, 리세트 변수 등)
④ 입출력에 관한 변수
⑤ 기타, POU 사이에서 교환하기 위한 변수

특히, 입출력에 관한 변수는 리소스 고유의 변수로, 어드레스의 배분도 포함하며, 글로벌 변수 워크시트에 모아서 선언해야만 합니다.
다음은 글로벌 변수의 선언 예입니다.

```
VAR_GLOBAL
(*******입력*******)
  준비완료CR0 AT%IX3.0.0:BOOL;
  자동COS1          AT%IX3.0.1:BOOL;
  프로그램 연속COS2  AT%IX3.0.2:BOOL;
  스타트PB          AT%IX3.0.3:BOOL;
(*******출력*******)
  스타트PL          AT%QX4.0.0:BOOL;
  이상PL            AT%QX4.0.1:BOOL;
  냉각 팬           AT%QX4.0.2:BOOL;
  비상정지          AT%QX4.0.3:BOOL;
(*******공통 데이터*******)
  등록 DATA         AT%MW3.0:INT101_11;
  수동속도          AT%MW3.1200 :INT;
  기어설정          AT%MW3.1202 :INT;
END_VAR
```

4.4.3 기본 데이터 형

변수는, 데이터형을 지정(선언)하고 사용합니다만, IEC규격에는, 많은 기본적인 데이터형을 규정하고 있습니다. 이것을 기본 데이터형이라고 합니다.
MICREX-SX에서 지원하고 있는 기본 데이터형은 다음과 같습니다.

No.	키워드	데이터 형	비트 수	데이터 범위
1	BOOL	불형	1	0또는 1
2	INT	정수형	16	-32, 768~32, 767
3	DINT	배정도정수형	32	-2, 147, 483, 648~2, 147, 483, 647
4	UINT	부호없이 정수형	16	0~65,535
5	UDINT	부호없이 배정도정수형	32	0~4,294, 967, 295
6	REAL	실수형	32	$-2^{128}<N\leqq-2^{-126}$, 0, $2^{-126}\leqq N<2^{128}$
7	TIME	계속시간형	32	0ms~4, 294, 967, 295ms(49일17:02:47s295ms)
8	DATE	날짜형	32	1970년1월1일~2106년2월7일
9	TOD	시각형	32	0:00:00~23;59:59
10	DT	날짜시각형	32	1970년1월1일0:00:00~2106년2월7일6:28:15
11	STRING	가변장 문자열형	-	-
12	WORD	길이 16비트열형	16	16#0000~16#FFFF
13	DWORD	길이 32비트열형	32	16#00000000~16#FFFFFFFF

(주) 1. 16#은 16진수를 표시합니다.
2. 실수 형 REAL의 정도보증 범위는 기본적으로 6항입니다. 명령에 의해서 보증 범위의 항 수가 작아 질 경우가 있습니다. 이하의 (b)②를 참조하십시오

(1) 정수형에 대해서

ⓐINT, DINT 형의 변수의 데이터 범위는, $-2^{(\text{비트수}-1)}\sim(2^{(\text{비트수}-1)}-1)$입니다.
결국, 16비트의 INT형의 경우 ~이 됩니다.
$-2^{(16-1)}\sim(2^{(16-1)}-1)\rightarrow\sim-2^{15}\sim(2^{15}-1)\rightarrow-32{,}768\sim32{,}767$
UINT, UDINT형의 변수의 데이터 범위는 $0\sim(2^{(\text{비트수})}-1)$입니다.

ⓑ정수형은 INT, DINT, UINT형과 많은 형이 정의되어 있습니다. 이것은 변수로의 부적절한 수치대입의 방지와 최적의 메모리사이즈에 의해 데이터 메모리의 절약을 위해서입니다. 하지만, 특별히 데이터 메모리 용량을 고려할 필요가 없다면, 그렇게 신중히 생각하지 않아도, 예를 들어, INT형을 기본으로 해서 수치범위가 -32,768~32,767을 넘을 것 같으며 DINT형을 사용하는 등, 기본적인 방침을 결정해 두는 편이 프로그램의 설계와 관리를 쉽게 할 것입니다.

(2) 실수형에 대해서

ⓐ 실수형 REAL은, PLC로 실수를 부호화해서 근사치로 표현한 단정도 부동소수점입니다.
부동소수점은 부호부(sign)과 가수부(mantissa)와 지수부(exponent)라는 3개의 수치의 조합의 위해서 표현됩니다.
REAL형은 32비트데이터로, 데이터의 내역은 다음과 같습니다.

31	30	23 22	0
부호부s(1)	지수부e(8)	가수부m(23)	

이 32비트의 데이터는 2진수(부호부 1비트, 지수부 8비트, 가수부 23비트)에 의해

$(-1)^S \times (1+m) 2^{(e-127)}$ ……(1식)

라는 수치를 표현하고 있습니다. 여기서, s, m, e의 의미는 다음과 같습니다.

s : 부호부 수치가 0이상일 때 "0", 부수로 "1"이 됩니다.

m : 가수부 왼쪽의 비트부터 2^{-1}의 항, 2^{-2}항…으로 해석합니다.

e : 지수부 $1 \leqq e \leqq 254$입니다. 0과 255는 특별한 경우를 표현합니다.

0 : 0을 표현, 255 : ∞을 표현

예를 들어

1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

의 데이터 값의 경우에는

$s=1 \quad e=129 \quad m=2^{-1}+2^{-2}=0.75$

에 의해, (1식)보다

$-1 \times (1+0.75) \times 2^{(129-127)} = -7.0$

이 됩니다.

부동소수점은 정규화(가수부의 최상위가 항상 1이 되도록 지수부를 조정하는 것)하고 있기 때문에, 가수부의 최상위의 1은 생략되어 있습니다. (1식)의 1+m의 1은 생략한 1을 되돌리고 있습니다. 따라서, 23비트의 가수부는 24비트 분의 정도를 가지고 있다고 생각할 수 있습니다. 이것은 10진수로 고치면 유효숫자 7항에 해당합니다.

ⓑ 실수형 REAL의 정도보증범위는, 기본적으로는 유효숫자 6항입니다만, 수치연산 평션에 있어서 5항 이하가 되는 경우가 있습니다.

유효항 수	명령		비고
5항	SIN	정현	입력이 $-2\pi \sim 2\pi$의 경우. 왼쪽이외는 5항 이하
	COS	여현	입력이 $-2\pi \sim 2\pi$의 경우. 왼쪽이외는 5항 이하
	TAN	정접	입력이 $-2\pi \sim 2\pi$의 경우. 왼쪽이외 또는 $\pi/2$의 정수배의 근방으로 5항 이하
	ATAN	역정접	
	SQRT	평방근	
4항	EXP	지수	입출력이 $-64 \sim 64$의 경우. 왼쪽이외는 4항 이하
	ASIN	역정현	| 입력치 | =1.0, 0.998999이하의 경우. 왼쪽이외는 4항 이하
	ACOS	역여현	| 입력치 | =1.0, 0.998999이하의 경우. 왼쪽이외는 4항 이하
	EXPT	누승	
4항 이하	LN	자연대수	
	LOG	상용대수	

(주) SIN, COS, TAN명령의 유효 항수는 소수점 이하 제 4위까지입니다.
예를 들면, 0.0175425…라는 연산결과가 있다면, 0.0175가 유효항수가 됩니다.

ⓒ 실수 형 REAL의 범위는 다음과 같습니다.

$-2^{128} < N \leqq -2^{-126} \rightarrow -3.40282e+38 < N \leqq -1.17549e-38$

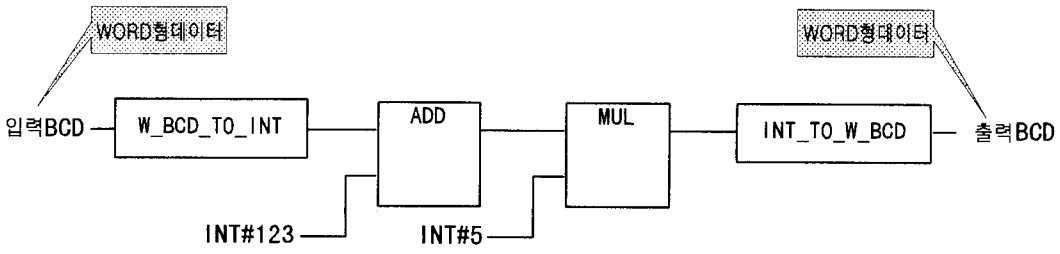
$2^{-126} \leqq N < 2^{128} \rightarrow 1.17549e-38 \leqq N < 3.40282e+38$

에 의해, $-1.17549e-38 < N < 1.17549e-38$의 범위는 0이 됩니다.

BCD에 대해서

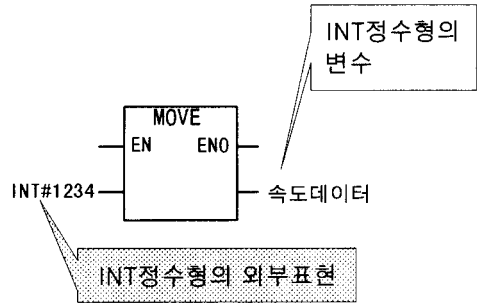
IEC에, 규격 상의 BCD의 데이터형은 존재하지 않습니다. 따라서, BCD는 비트열 형 데이터인 WORD형 또는 DWORD형 데이터를 BCD코드로 인정하고 취급합니다. 구체적으로는, BCD입력은 WORD형 또는 DWORD형의 변수에 넣은 W_BCD_TO_INT와 D_BCD_TO_INT의 형변환 명령에 의해 정수형으로 변환시킨 후 처리를 합니다. 반대로, BCD출력은, 정수형의 데이터를 INT_TO_W_BCD와 INT_TO_D_BCD의 형변환 명령으로 BCD로 변환시켜 출력하게 됩니다. 또, MICREX-F의 BCD는, 최상위 비트(MSB)가 정부를 판단하는 신호비트입니다만, SX의 BCD는 부호 없는 BCD입니다.

데이타형	데이타범위
WORD형	0~9999
DWORD형	0~99999999

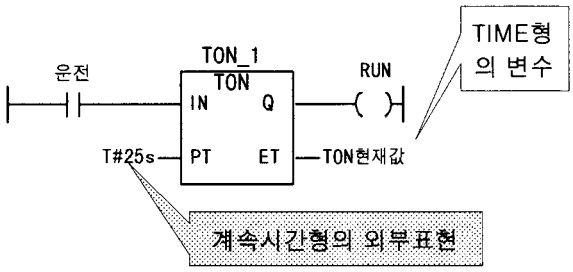


(4) 기본 데이터형의 외부표현(리터럴)

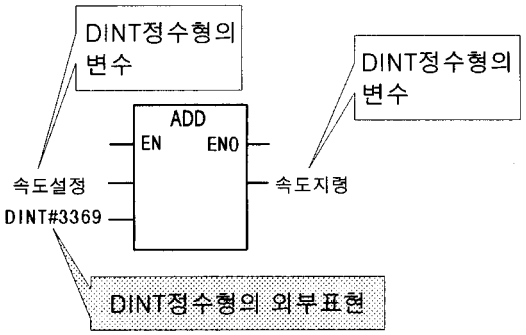
ⓐ 프로그램 안에 수치를 변수로 대입하거나, 타이머에 설정치를 설정하거나 하는 경우에는 각 데이터형에 맞춘 수치표현을 할 필요가 있습니다. 이 정수를 데이터형의 외부표현(리터럴)이라고 합니다. 외부표현(리터럴)은 어떤 데이터형의 변수에 부여하는 수치(정수)를 어떻게 표현하는가를 정의하고 있습니다.



전송처의 데이터형과 맞춤



타이머의 설정치는 계속시간형



산술명령 등은 모두 데이터형과 맞춤

ⓑ 외부표현의 기본은 [데이터 형] # [10진수(2진수 또는 16진수)] 입니다.

데이터형은 생략할 수 있는 것도 있습니다만, 형을 명확하게 하기 위해서 생략하지 않을 것을 적극 추천합니다. 아래에 외부표현의 예를 들었습니다. 아래 표에 나타난 바와 같이 정수형(INT, DINT, UINT, UDINT)는 16진수 표기도 가능하며, 비트열 형(WORD, DWORD)는 2진표기 및 16진 표기가 가능합니다.

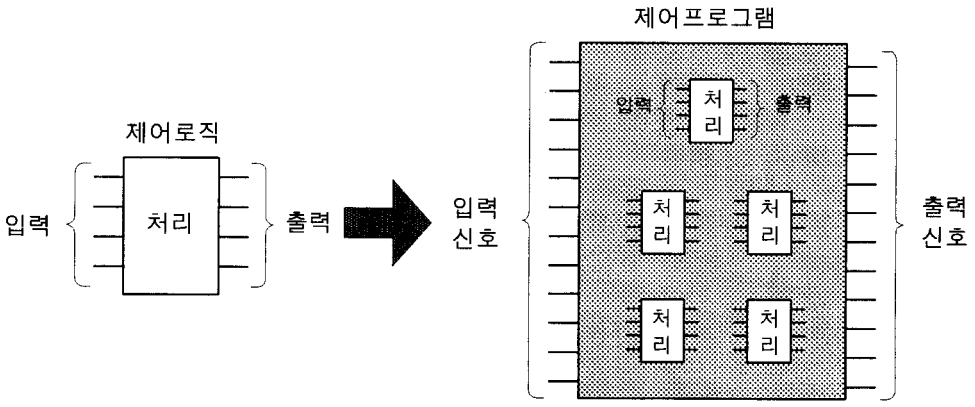
No.	데이터 형	외부표현의 예			비고
1	BOOL	TRUE, FALSE, BOOL#0, BOOL#1			보통은 TRUE, FALSE를 사용
2	INT	INT#1234, INT#16#FF0F, (-1000, 0, 12345)			데이터 형을 구별하기 위해, 데이터 형을 붙이는 것을 추천함
3	DINT	DINT#12345678, DINT#16#F000F00F			
4	UINT	UINT#1000, UINT#16#FFFF			
5	UDINT	UDINT#1000000, UDINT#16#1111FFFF			
6	REAL	-10.354, 0.0, 0.234, REAL#10.0, -1.34E-12, -1.34e-12			소수점 붙이는 것으로 실수형
7	TIME	아래선 기호 없음	기본형	TIME#14ms, time#25s	시간단위는 d(일) h(시) m(분) s(초) ms(밀리 초)이고, 대문자 또는 소문가로 표시함(구별 없음)
			생략형	T#14ms, T#14s, T#14m, T#14h, T#14d T#25h15m, t#5d14h12m18s3ms	
		아래선 기호 있음	기본형	TIME#25h_15m (최대유효항의 오버플로는 허용됨) time#5d_14h_12m_18s_3ms	
			생략형	T#5d_14h_12m_18s_3ms t#25h_15m	
8	DATE	기본형	DATE#1984-06-25 date#1984-06-25		
		생략형	D#1984-06-25 d#1984-06-25		
9	TOD	기본형	TIME_OF_DAY#15:36:15 time_of_day#15:36:15		
		생략형	TOD#15:36:15 tod#15:36:15		
10	DT	기본형	DATE_AND_TIME#1984-06-05-15:36:17 date_and_time#1984-06-05-15:36:17		
		생략형	DT#1984-06-05-15:36:17 dt#1984-06-05-15:36:17		
11	STRING	"(길이 제로의 문자열), ''(공백문자 한 개를 포함한 문자열) 'A', 'ABC'			
12	WORD	2진표현	WORD#2#1010111110101111 WORD#2#1010_1111_1010_1111		"_" 기호는 표현상인 것으로 의미는 없음
		16진표현	WORD#16#AFAF		
13	DWORD	2진표현	DWORD#2#11110000111100001010111110001111		
		16진표현	DWORD#16#F0F0AF8F		

4.4.4 펑션블록(FB)과 펑션(FCT)

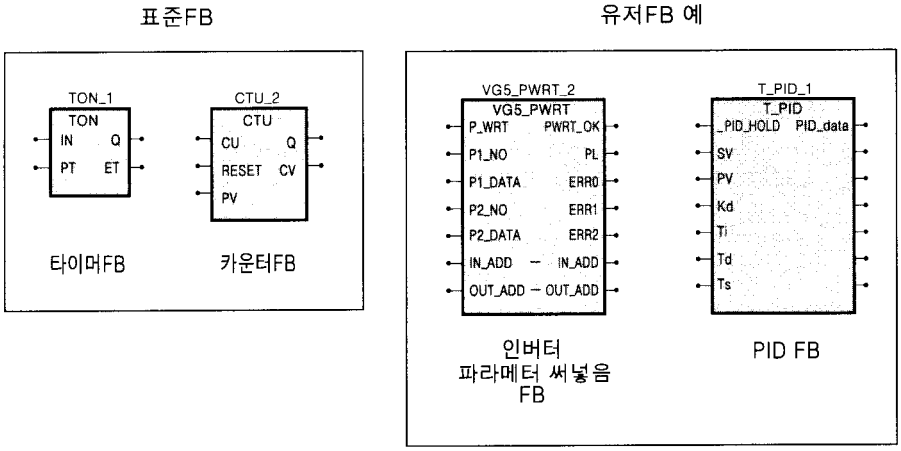
(1) 펑션블록이란

펑션블록(FB)이란, 표준화, 재 이용을 목적으로 하는 패키지화된 소프트웨어 요소(프로그램)입니다. FB에는, IEC규격으로 규정하고 있는 표준FB, MICREX-SX용으로써 표준으로 준비하고 있는 오리지널 FB, 기능 모듈의 제어를 확장하기 위해서 FUJI전기가 제공하는 확장 FB, 그리고 유저가 자유로이 설계할 수 있는 유저 FB가 있습니다.

프로그램은, 어느 기능을 실현하는 것을 목적으로 해서 설계된 제어로직으로, 아래의 그림처럼, 입력→처리→출력으로 표현할 수가 있습니다. 이것은, 입력을 기동 버튼과 센서 등의 입력실호, 출력을 마그네틱콘택터의 입력단락과 인버터의 정전지령 등의 출력신호로 생각한다면 하나의 프로그램 그것입니다. 프로그램을 순차적으로, 작은 기능단위로 분해하더라도, 프로그램을 어떠한 기능을 실현하는 제어 로직이라고 생각한다면, 반드시 이「입력→처리→출력」의 형태로 표현할 수가 있습니다.



작게 분해된 기능단위의 처리 중에는 완전히 같은 양상의 처리를 행하는 것도 있다면, 다른 어플리케이션에 같은 양상의 처리를 갖는 것도 있습니다. 종래의 PLC에서는, 소프트의 기능 단위의 분할(구조화)과 재 이용의 조직이 존재하지 않았으며, 있더라도 한정된 것이었습니다. IEC언어는 구조화와 재 이용을 커다란 장점으로 가지며, FB는 구조와 재 이용을 강력하게 지원하는 소프트웨어 요소입니다. 다음의 표시처럼, 그래픽언어(래더 그림 언어, FBD언어)상에서는 「입력→처리→출력」의 형식과 완전히 같은 양상의 표현형식을 갖습니다.



FB의 장점을 말하자면 아래와 같습니다.

① FB의 외부와는 입력 변수와 출력 변수만을 인터페이스해서, 완전히 캡슐화(외부로부터, FB의 내부 코드와 데이터를 조작할 수 없는 것)되어서, 재 이용을 목적으로 한 소프트웨어 패키지입니다.

② FB의 외부 파라미터 사양과 FB의 기능을 이해한다면, 내장 구조를 자세히 알지 못하더라도, FB를 조직화하는 것보다, 어플리케이션을 높은 효율로 구축할 수가 있습니다.

③ 충분히 시험된 FB를 사용함으로써 고품질의 어플리케이션을 설계할 수 있습니다.

④ D300win의 지원기능에 의해, 프로그램 언어의 표준명령과 완전한 동일 취급으로, 유저 FB를 이용한 프로그래밍이 가능합니다. 또한, 라이브러리 기능 등, 표준화, 재 이용을 위한 기능도 충실히 합니다.

⑤ FB는, 로더 D300win으로 내부회로의 개별적인 모니터를 할 수 있기 때문에, 디버그가 굉장히 용이합니다.

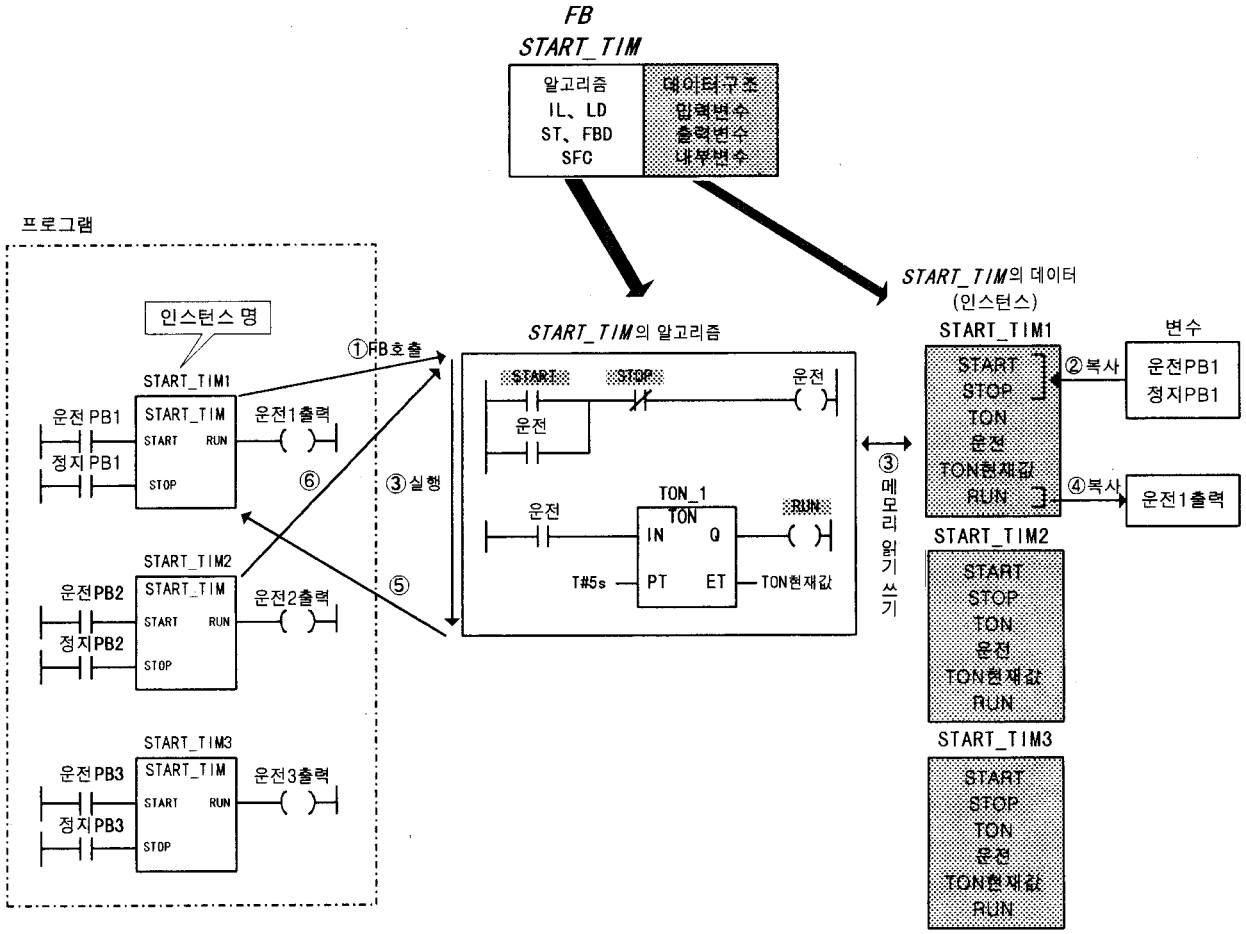
(2) 펑션블록의 동작

FB는 입력 변수, 출력 변수, 내부 변수의 데이터 구조와 IL, LD, FBD, ST, SFC로 기술되는 프로그램(알고리즘)으로 구성됩니다. 입력 변수, 출력 변수는 외부와의 인터페이스 변수이고, 내부 변수는 FB 내에서 사용되는 로컬 변수입니다. 변수는 FB가 다음 실행되기까지의 사이, 데이터를 보존해 둘 수가 있습니다. 그렇기 때문에, 입력 변수, 출력 변수, 내부 변수를 보존해 두는 메모리 영역을 FB의 호출수(사용수)만큼 가질 필요가 있습니다. 이 영역에 세트되는 데이터 값을 인스턴스라고 부릅니다. 인스턴스는 명명된 인스턴스 이름으로 구별됩니다. FB의 사용 시에 인스턴스 이름을 붙이는 것은, 이 데이터 값의 구별을 위해서입니다. 이 조직화로 인해서, FB는 사용하는 수가 늘어도, 프로그램 스텝 수는 증가하지 않습니다. 단지, 호출하는 주변회로의 스텝 수가 증가할 뿐입니다. 하지만, 데이터 용량은 FB의 호출수(사용수)분 만큼 증가하게 됩니다.

다른 프로그램 내에서 FB가 호출되면, 인스턴스 내의 입력 변수에 접속되어있는 변수의 데이터 값이 복사되고, 그 FB의 알고리즘인 프로그램이 인스턴스 내의 데이터 값을 참조하면서 실행됩니다. 프로그램 실행된 결과인 내부변수와 출력 변수는 인스턴스에 다시 쓰여집니다.

FB 실행 처리 후에 인스턴스 내의 출력 변수는 출력 변수에 접속되어 있는 변수에 데이터 값을 전하고, FB의 실행 처리가 완료됩니다.

아래에서는 FB가 실행된 개요를 나타냅니다.



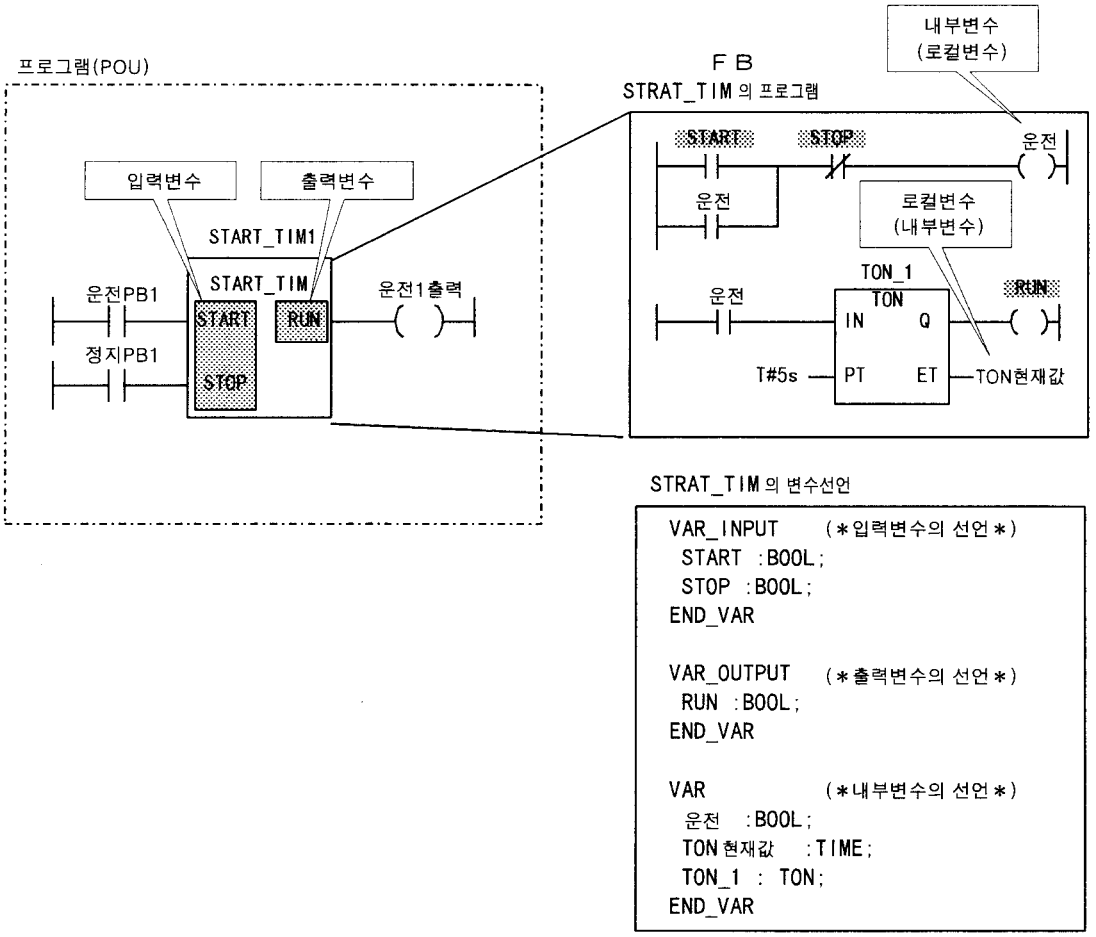
FB의 프로그램은 IEC가 규정해 놓은 5언어의 1개를 사용해서 기술되기 때문에, 대부분의 프로그램은 FB로써 패키지 화 할 수가 있습니다. FB는 어셈블리의 서브루틴과 MICREX-F의 FM처럼 다른 어플리케이션에서 자유롭게 호출해 사용됩니다.

(3)펑션블록의 변수 선언

(a)입력 변수와 출력 변수

펑션블록(FB)에서 사용 가능한 변수는, 입력변수, 출력변수, 입출력 변수, 내부변수(로컬변수)입니다. 입력변수는 외부로부터 FB가 데이터를 받아들이는 입력 파라미터의 역할을 합니다. 또한 출력변수는, FB로부터 외부로 데이터를 넘겨주는 출력 파라미터의 역할을 합니다. 입력변수는 변수 워크시트 중에 개시와 종료를 선언하는 키워드 VAR_INPUT와 END_VAR로 묶어서 선언됩니다. 마찬가지로, 출력 변수는 키워드 VAR_OUTPUT와 END_VAR로 묶어서 선언되며, 내부 변수는 키워드 VAR와 END_VAR로 묶어서 선언됩니다.

그래픽 언어(래더 그림 언어. FBD언어)에 있어 FB의 형태를 보면, 입력 변수와 출력변수의 역할은 명확합니다. FB에의 데이터의 주고받음은, 이 입력 변수, 출력 변수에 접속되는 변수를 개입시켜 행하여집니다. 물론, 내부변수(로컬 변수)는 FB내에서만 유효한 변수이며, 외부로부터 액세스할 수는 없습니다. 이것에 대해, 프로그램(POU)은 외부 변수에 의해, 다른 POU와 통신(데이터의 주고받음)을 행합니다. 결국, 글로벌 변수를 외부 변수로 해서 POU의 변수 워크시트에 선언하기 때문에, 리소스 고유의 글로벌 변수와의 관련이 남습니다. FB는 입력 변수와 출력 변수를 인터페이스로 하고 있기 때문에, 외부와의 관련성이 없습니다. 그 때문에, 완전한 독립성을 유지할 수 있고, 프로그램의 표준화, 재이용을 추진할 수가 있습니다.



STRAT_TIM 의 변수선언

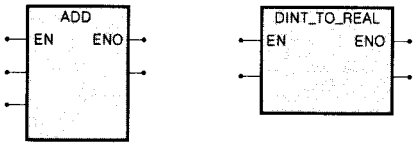
```
VAR_INPUT     (*입력변수의 선언*)
  START :BOOL;
  STOP :BOOL;
END_VAR

VAR_OUTPUT    (*출력변수의 선언*)
  RUN :BOOL;
END_VAR

VAR           (*내부변수의 선언*)
  운전 :BOOL;
  TON현재값 :TIME;
  TON_1 : TON;
END_VAR
```

(4) 펑션(FCT)

(a) 펑션이란

펑션(FCT)도 펑션블록(FB)과 마찬가지로 재 이용 가능한 소프트패키지입니다. FCT와 FB의 커다란 차이는, FCT는 내장 변수와 출력 변수의 데이터 값을 보존하지 않는 것입니다. 따라서, FCT는 FB처럼 데이터 값을 보존할 인스턴스를 가질 필요 없이, 개개의 인스턴스 이름을 붙일 필요도 없습니다. 이 때문에 FCT는, 매회, 입력값 만을 기본으로 연산 처리합니다. 그 결과, 어느 특정의 입력값에 대해서, 항상 같은 값을 회답하도록 되었습니다. FCT는 인스턴스를 가지고 있지 않기 때문에, 사용수의 증가에 의한 데이터 용량 증가는 없습니다. 또한, 복수의 출력 변수를 가질 수 있는 FB에 대해서, FCT는 어느 특정의 입력값에 대해서 생성하는 데이터 요소가 하나 있습니다. (단, 데이터 요소에는 배열 같은 다요소 변수도 포함)

FCT도 IEC에서 규정하고 있는 표준 FCT와 MICREX-SX로 준비해 있는 오리지널 FCT, 유저로서 자유롭게 설계 가능한 유저 FCT가 있습니다. FCT의 예로써는 산술연산의 가산명령(ADD)과 형변환 명령(DINT_TO_REAL)을 다음과 같이 표시합니다.



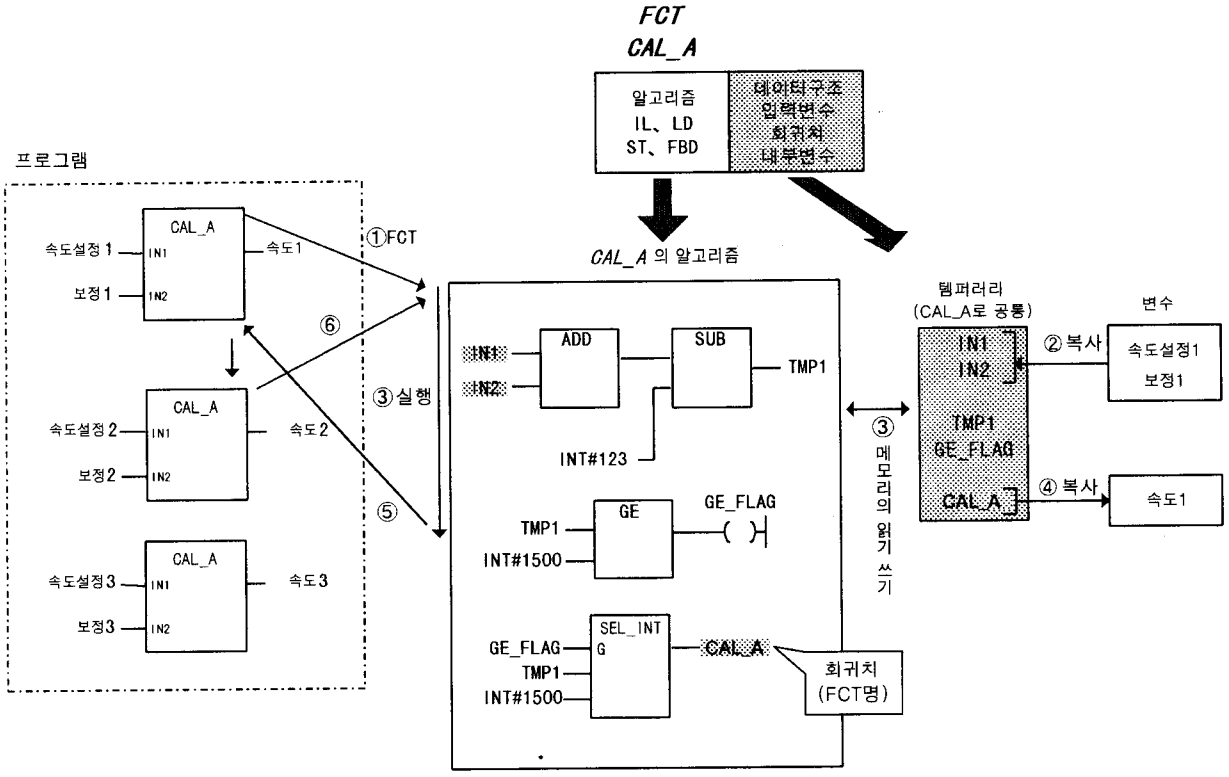
가산FCT　　형변환FCT

FCT의 예

(b) 펑션의 동작

펑션(FCT)은, FB와 달리 매회, 템포러리 영역을 이용해서 연산을 시행하고, 개별의 인스턴스 영역을 가지지 않습니다. FCT는, INPUT에 접속되어 있는 변수의 값을 템포러리 영역에 복사해서, FCT의 알고리즘(프로그램)을 템포러리 영역을 참조하면서 실행합니다. 프로그램의 종류로 OUTPUT에 접속되어 있는 변수로의 결과를 복사해서, FCT의 실행 처리가 종료됩니다. FCT의 실행 처리가 종료하면, 이 템퍼러리 영역은 다음의 다른 POU의 처리를 위해서 개방됩니다.

아래는 FCT가 실행되는 개요입니다.



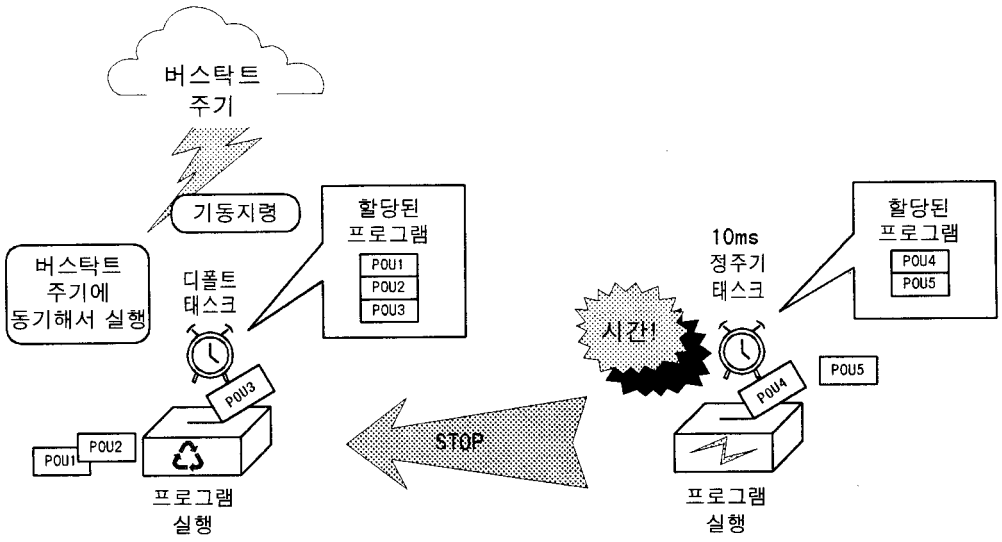
4.5 프로그램의 실행방법

4.5.1 태스크

태스크란, POU(프로그램)의 실행 스케쥴을 제어하는 실행 제어기능입니다. 종래의 PLC프로그램은 특별한 설정을 하지 않는 한, 설계한 프로그램은 전부 싸이클릭 스캔의 주기로 실행 됩니다. 특정의 주기, 예를 들어, 정주기로 실행하고 싶은 프로그램의 경우만은, 정주기 프로그램을 지시하는 명령어 가운데에 프로그램을 기술합니다.

종래, 프로그램 처리는 1본의 싸이클릭 스캔 가운데서 실행됩니다. 그 때문에, 고속처리로의 대응에는, 점프 명령에 의해 프로그램의 스킵과 마스터 컨트롤 릴레이에 의한 명령의 불실행에 의해, 스캔타임의 단축을 도모해 왔습니다. 하지만, 명령어의 실행, 불실행의 처리시간의 차이와 프로그램의 추가, 변경에 의해, 스캔타임이 변동하기 때문에, 그 확인, 조정에 많은 노력을 요했습니다. 프로그램 설계 전에 프로그램의 정확한 스캔타임을 계산하는 것은 , 실제로는 곤란한 것입니다. 또한 고속 제어부분의 정주기 실행과 끼워 넣음 실행도 고속화로의 대응책으로써 채용되어 왔습니다만, 그 실행 본수와 사용법에 제한이 있었습니다.

IEC규격에는, 프로그램을 보다 유연하게 실행 제어하기 위해서 태스크라고 하는 개념을 도입했습니다. 구조화 설계된 프로그램은, 각각 개별적으로 최적의 실행 타이밍을 부여할 수가 있습니다. 결국, 프로그램 설계와 그 실행 제어를 따로따로 생각할 수가 있습니다. 이것은, 종래의 PLC제어에는 없는 커다란 장점입니다. 따라서, SX에서는 설계된 프로그램은 태스크에 할당되어 처음 실행됩니다. 태스크는 설정된 시간이 오면, 할당된 POU를 할당 순번에 따라서 순차적으로 실행합니다.

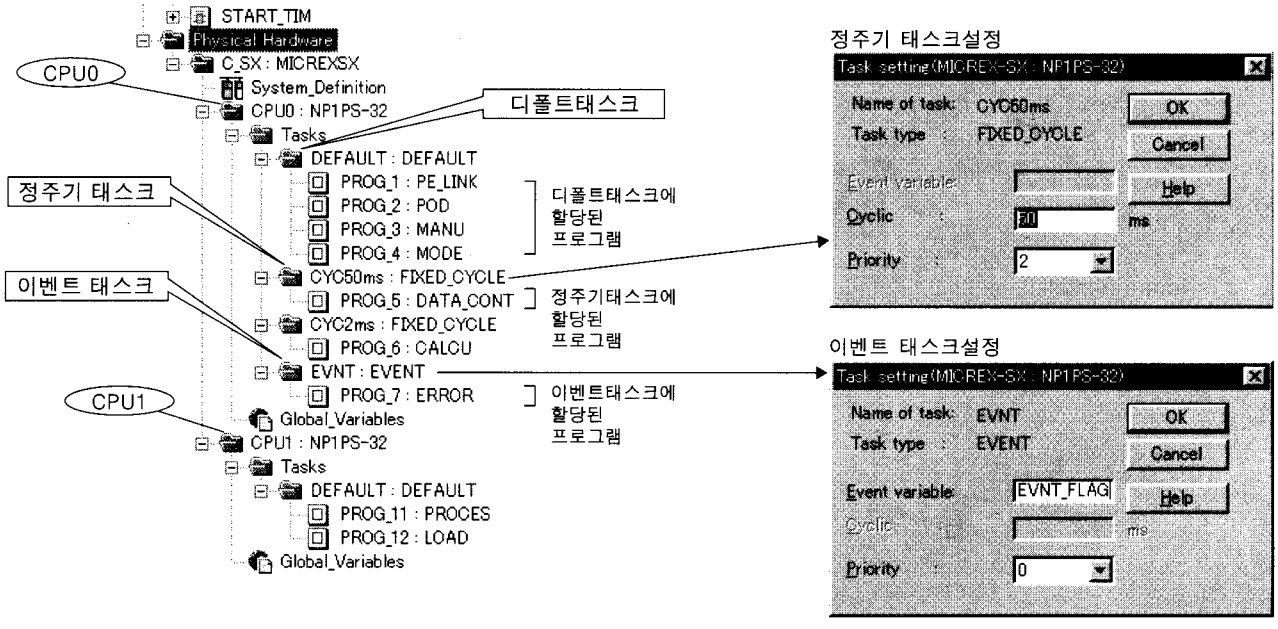


태스크에는 디폴트 태스크, 정주기 태스크, 이벤트 태스크의 3종류가 준비되어 있습니다. SX의 경우, 태스크 등록을 할 수 있는 것은, 프로그램만으로, FB. FCT는 단독으로 태스크 등록이 불가능합니다. 따라서, FB를 정주기 제어하는 데에는, FB를 프로그램 안으로 불러들여서, 그 프로그램을 정주기로 실행하게 됩니다.
각 태스크의 사양은 아래와 같습니다.

항목	사양	
태스크의 종류	디폴트태스크 (싸이클릭 스캔)	· 항상 되풀이 실행됩니다. · 종래의 싸이클릭 스캔의 프로그램에 상당합니다.
	정주기 태스크	· 지정한 주기로 1회 실행됩니다. · 종래의 정주기 프로그램에 상당합니다. · 실행주기는 탁트 주기의 정수배설정, 단, SX버스탁트주기가 0.5ms의 경우, 정주기의 설정0.5ms는 가능하지만, 정주기 설정 1.5ms와 2.5ms의 설정은 할 수 없음. 1ms이상은 정수값 설정이 됨. · 태스크 우선도를 0~3레벨로 설정
	이벤트태스크	· 지정한 BOOL형의 변수가 실제로 변화한 때에 실행됨. · 종래의 끼어 넣기 프로그램에 상당함.
태스크의 본수	1본(디폴트) + 4본(정주기와 이벤트의 합계)	
태스크의 우선도	디폴트 이외의 정주기 및 이벤트태스크에는, 동시 기동시의 우선도를 4단계 (최우선이 0)로 붙이는 것이 가능함. 0〉 1〉 2〉 3〉 디폴트	

4.5.2 태스크로의 프로그램의 할당 방법

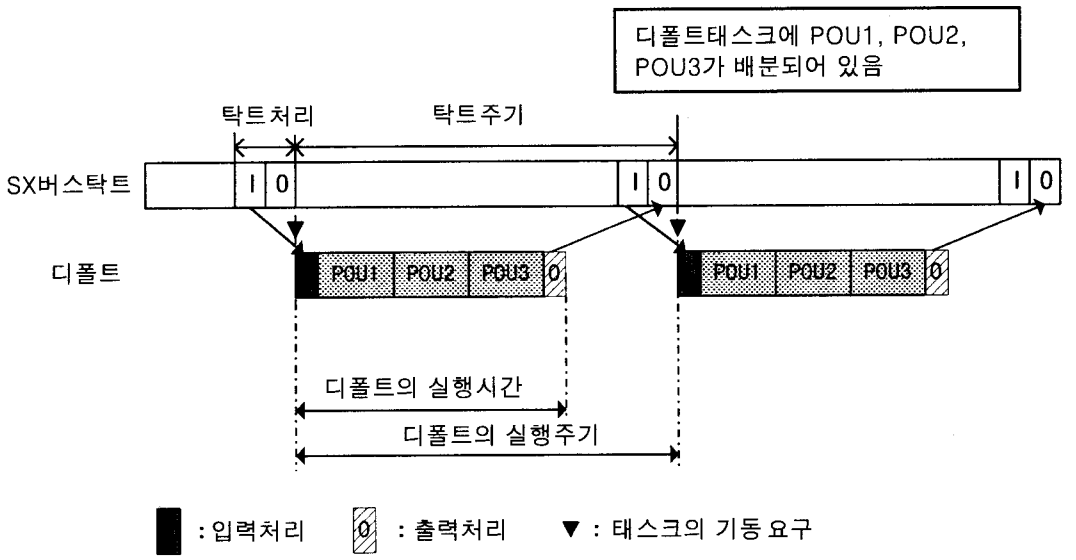
태스크로의 배분은 D300win의 프로젝트 트리에 있어 하드웨어 설정부에서 행합니다. 다음은 프로그램의 태스크로의 할당의 예입니다.



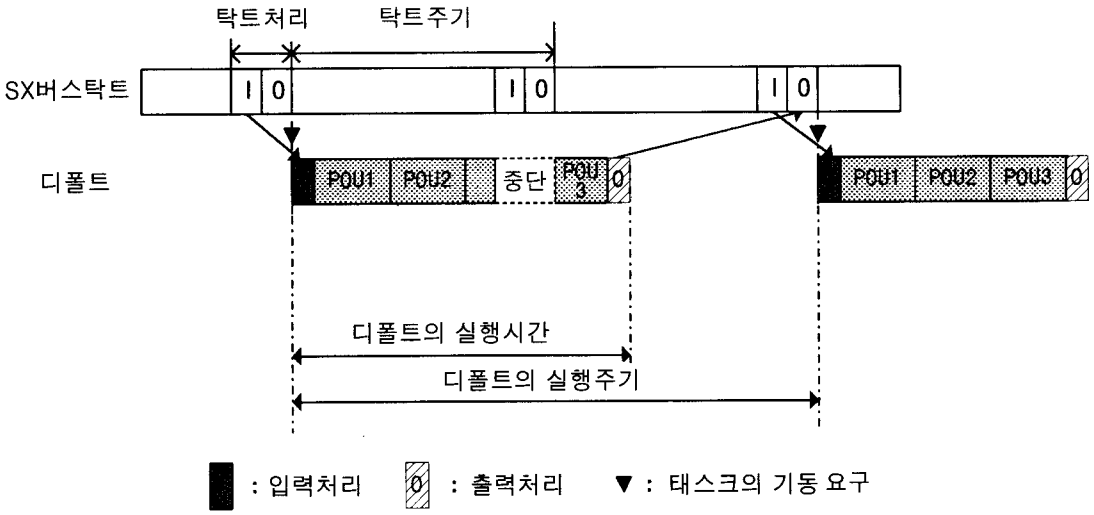
4.5.3 디폴트태스크

싸이클릭 스캔으로 POU(프로그램)을 실행하는 태스크입니다. 종래의 프로그램 실행과 동등한 처리이며, 통상은 이 태스크에 모든 POU가 배분되어 있습니다. 우선도는 가장 낮고, 정주기 태스크와 이벤트태스크의 실행 시에는 처리를 중단합니다. 기동은, SX버스의 버스 탁트주기와 같은 주기로 기동됩니다.

아래는 디폴트태스크의 동작 예입니다.



탁트 주기보다 디폴트태스크의 POU실행 시간이 짧은 경우



탁트 주기보다 디폴트태스크의 POU실행 시간이 길 경우

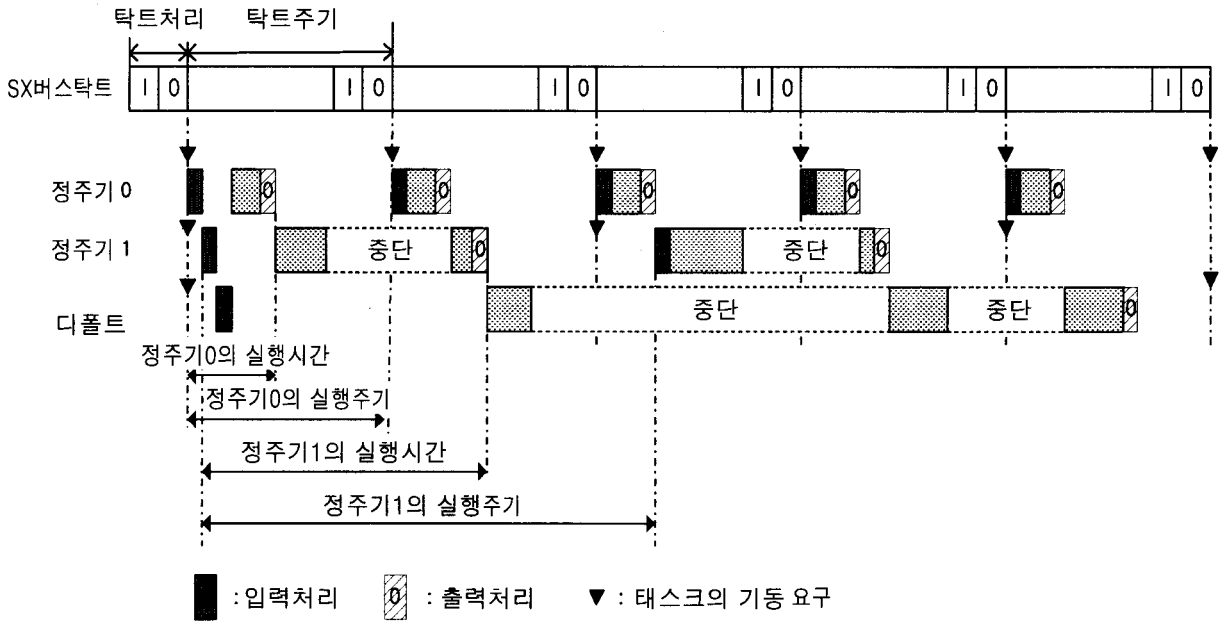
4.5.4 정주기태스크

정기적인 주기로 실행하고 싶은 POU(프로그램)을 이 태스크에 할당합니다. 정주기 태스크에는 주기 시간과 동시에 기동이 걸렸을 때(예를 들어 10ms 정주기 태스크와 20ms 정주기 태스크)를 위해서 우선도를 지정해 놓습니다. 주기시간은 SX의 버스탁트 주기의 정수배로 설정됩니다. SX 버스 탁트 주기의 설정은 0.5ms, 1ms, 2ms, … 19ms, 20ms(1ms단위)입니다. 단, SX버스탁트 주기가 0.5ms의 경우, 정주기 0.5ms의 설정은 가능합니다만, 정주기 1.5ms와 2.5ms의 설정은 할 수 없습니다. 1ms 이상은 정수설정이 됩니다. 정주기를 사용하는 조건으로써, 아래와 같은 경우라고 생각되어 집니다.

① 필터와 적분명령에 의해서, 일정주기로 실행할 필요가 있는 경우
② 버스 탁트 주기는 짧지만(예를 들어 1ms), 디폴트태스크의 실행시간은 긴(예를 들어 5ms)고속의 제어를 할 수 없는 경우

②의 경우, 고속응답을 필요로 하는 POU를 정주기 태스크에 할당, 그 주기를 1ms로 설정하는 것으로, 1ms 스캔 프로그램에 상당하는 고 응답제어가 가능합니다. 단, 정주기 프로그램이 커지는 경우는 디폴트태스크를 실행할 시간이 없어져서, 전체의 제어에 지장을 줌으로 밸런스를 고려할 필요가 있습니다.
아래는 정주기 태스크의 동작 예입니다.

동작조건
버스 탁트주기: 1ms
정주기0: 정주기1ms(1탁트주기)
정주기1: 정주기2ms(2탁트주기)
태스크우선도: 정주기0>정주기1>디폴트



정주기태스크의 동작 예

4.5.5 이벤트태스크

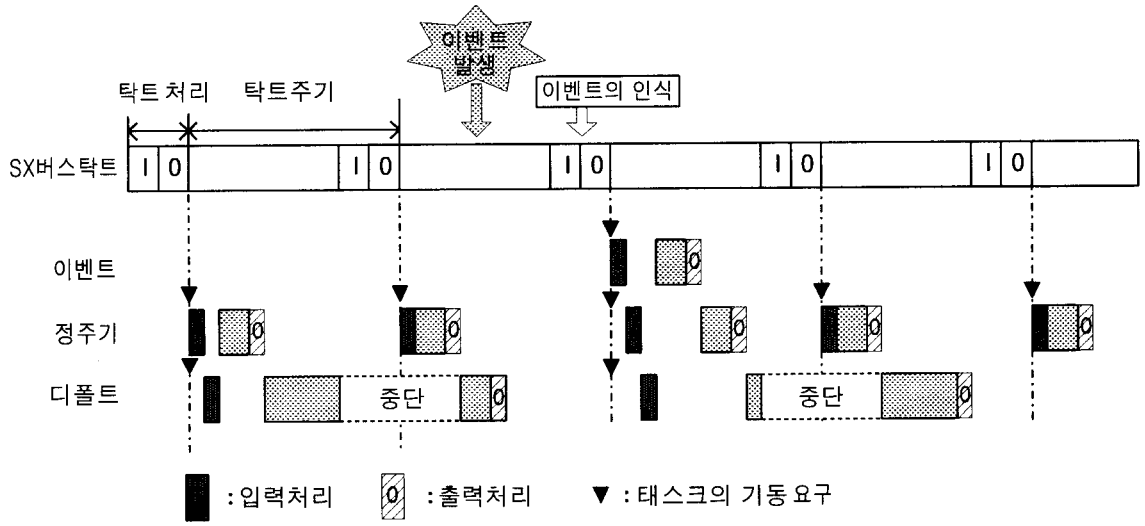
이벤트태스크에 할당된 POU(프로그램)은, 종래의 끼어 넣기 프로그램과 동등한 동작을 합니다. 이벤트태스크는 이벤트가 발생한 타이밍으로 즉시 기동되지 않고, 다음의 탁트 주기의 시작에서 이벤트가 인식되어, 태스크가 기동합니다. 이벤트에는 BOOL형의 변수가 지정됩니다. 이벤트에 지정한 변수는, SX버스 탁트 주기의 시작에 인식되기 때문에, 특히 입력 신호를 이벤트로써 사용하는 경우에는, 입력 신호는 버스탁트 주기 이상의 ON시간이 필요합니다. 이벤트태스크도 동시 기동시의 우선도를 설정할 필요가 있습니다.

동작조건

버스닥터주기: 1ms

정주기0: 정주기1ms(1닥터주기)

태스크우선도: 이벤트>정주기>디폴트



이벤트태스크의 동작 예

5. 고도의 사용방법

5.1 라이브러리의 활용

(1) 라이브러리 기능과 사용

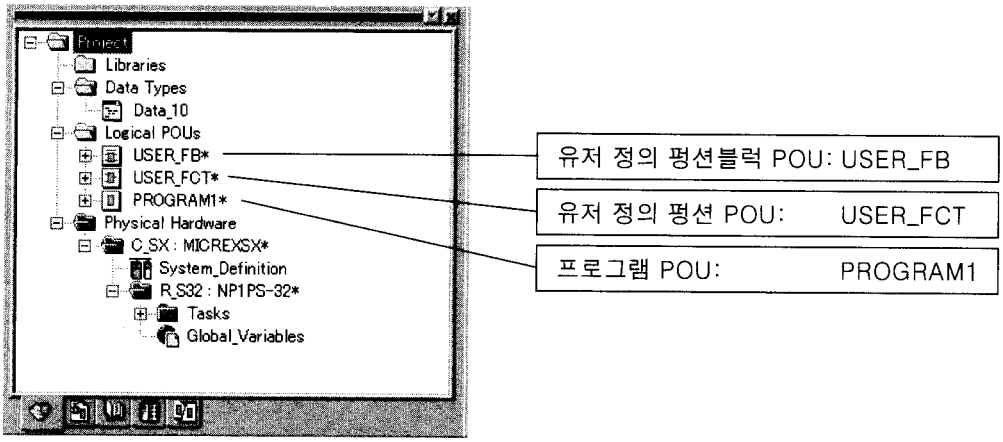
D300win은, 프로젝트에 기존의 프로젝트를 라이브러리로써 등록할 수 있습니다.
라이브러리로써 등록한 프로젝트 내에 사용되고 있는 프로그램(POU), 유저 정의 펑션블록, 유저 정의 펑션 및 데이터형 정의를 재 이용할 수 있습니다([Physical Hardware]부분은 이용할 수 없습니다). 또한, 라이브러리 에딕터에 의해 내부정보를 참조할 수 있습니다. (단, 내용을 변경하거나 인쇄하는 것을 안됩니다.

☞ 라이브러리로써 등록하는 프로젝트 내의 POU와 메인 프로젝트에 같은 이름의 POU가 존재할 경우, 라이브러리 내의 POU는 쓸모 없게 됩니다. (사용할 수 없습니다)

(2) 라이브러리의 등록

새로운 프로젝트의 라이브러리(폴더)에 프로젝트(파일)를 등록하는 방법에 대해서 설명합니다.

① 등록하는 프로젝트와 그 내용
다음에 표하는 프로젝트를 라이브러리에 등록합니다.

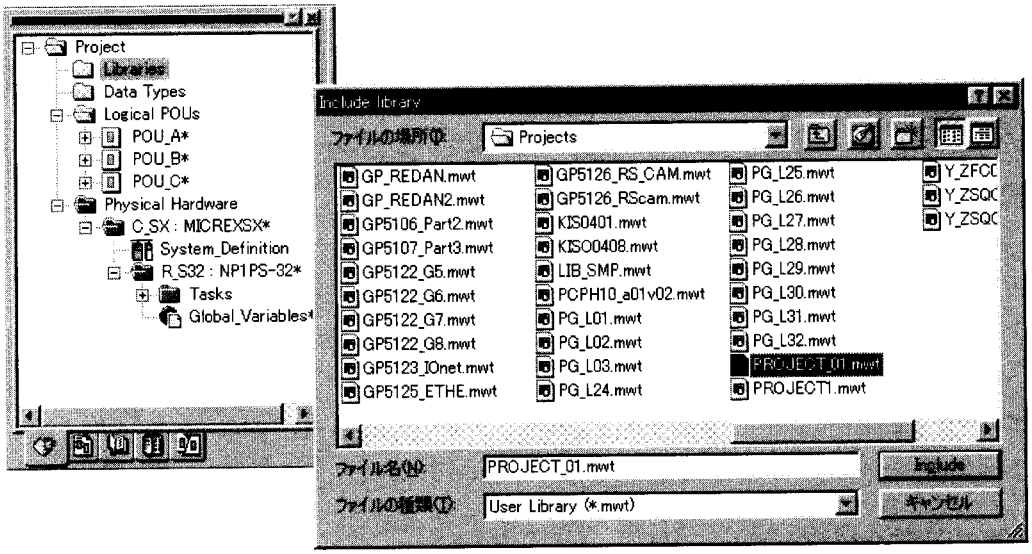


☞ 라이브러리에 등록하는 프로젝트는, 등록 전에 컴파일을 해 주십시오, 또한 컴파일 에러가 검출된 경우에는 에러를 제거해 주십시오.

② 라이브러리의 등록방법

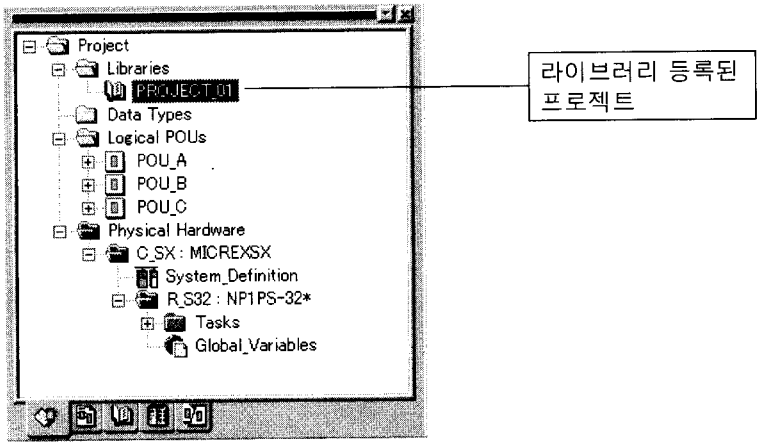
● 신규 또는 기존 프로젝트를 열고

● 프로젝트 트리의 서브 트리[Libraries]를 왼쪽 클릭하고, [Insert]탭을 왼쪽 클릭하면, [Include Library]다이얼로그가 표시됩니다. [Libraries]를 왼쪽 클릭한 후, 숏컷 메뉴의「Insert…」커맨드를 선택해도 같습니다.



● 라이브러리로써 등록하는 프로젝트를 선택합니다.
여기서는「PROJECT_01.mwt」선택합니다.

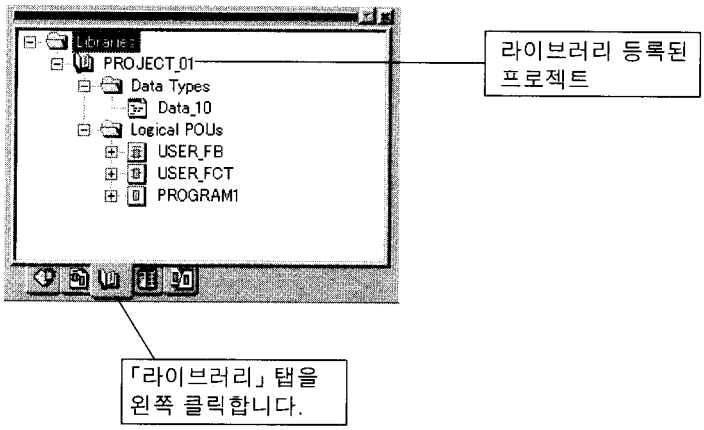
● [Insert]탭을 왼쪽 클릭합니다.
다음의 표에 표시된 것처럼 프로젝트 트리의 서브 트리에[PROJECT_01]이라는 프로젝트가 삽입됩니다.



③ 등록한 라이브러리의 표시

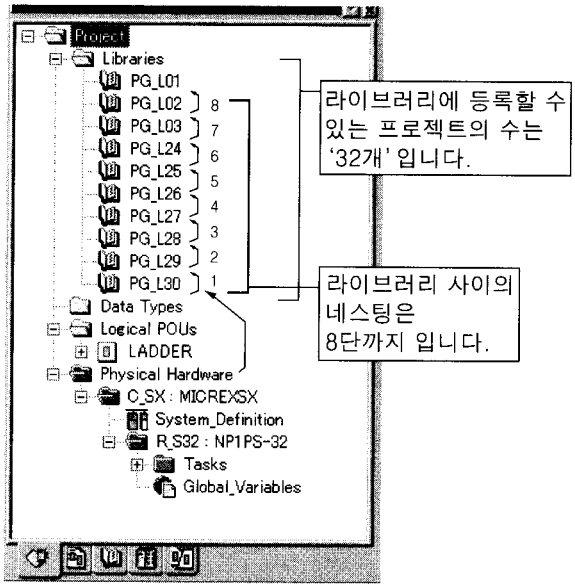
라이브러리에 등록한 프로젝트는, 표시시켜 확인할 수가 있습니다. 단, 편집할 수는 없습니다.

● 프로젝트 트리 윈도우의 하단에 있는 [Libraries]탭을 누릅니다. 라이브러리에 등록한 프로젝트 「PROJECT_01」이 표시됩니다.



④ 라이브러리 등록시의 주의사항

라이브러리에 프로젝트를 등록해 사용하는 경우, 다음 표에 표시 범위에서 행하여 주십시오.



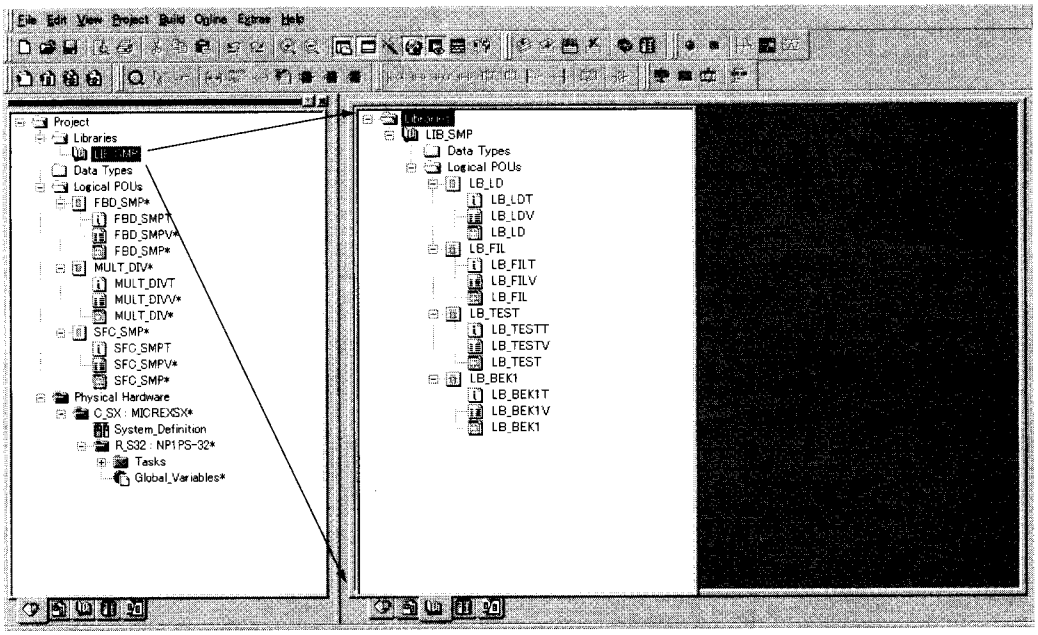
(3) 라이브러리의 사용방법

펑션/펑션블록의 사용방법

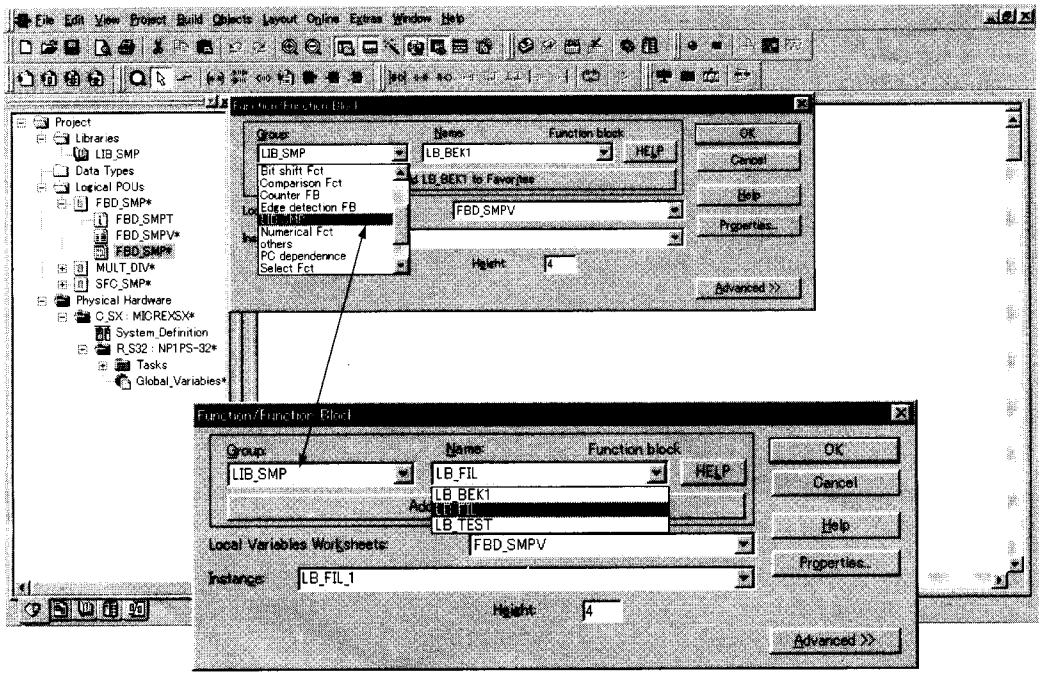
라이브러리로써 등록한 프로젝트 내의 유저 정의 펑션 및 유저 정의 펑션블록은, 코드워크시트 편집 시의 「Function/Function Block」다이얼로그 내의 [Name](명령리스트)에 포함됩니다.

또한, 에디터 위저드 사용 시에도 명령 리스트에 포함됩니다.

① 등록된 라이브러리



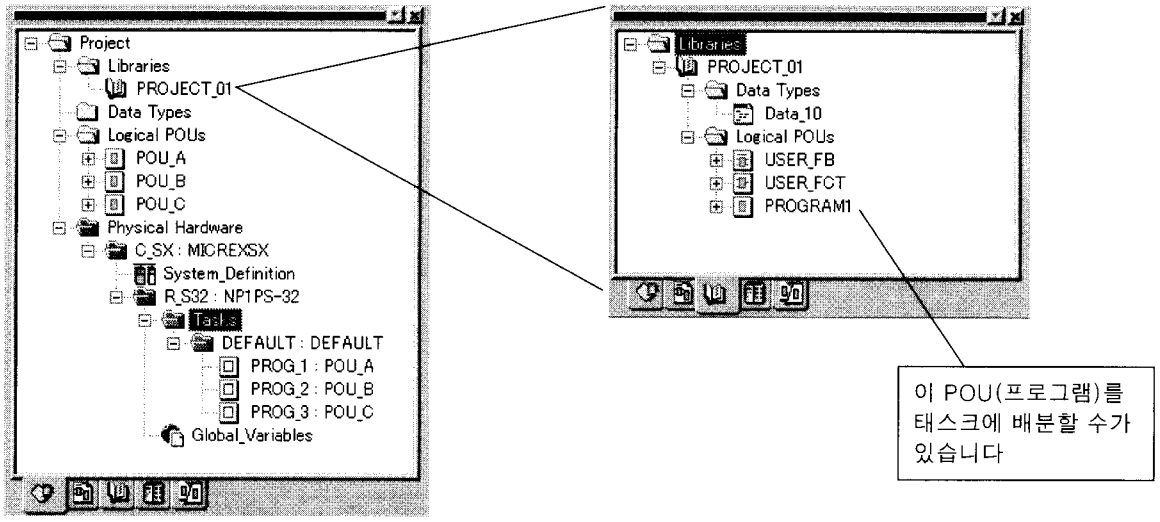
② 라이브러리 내의 FB의 인용

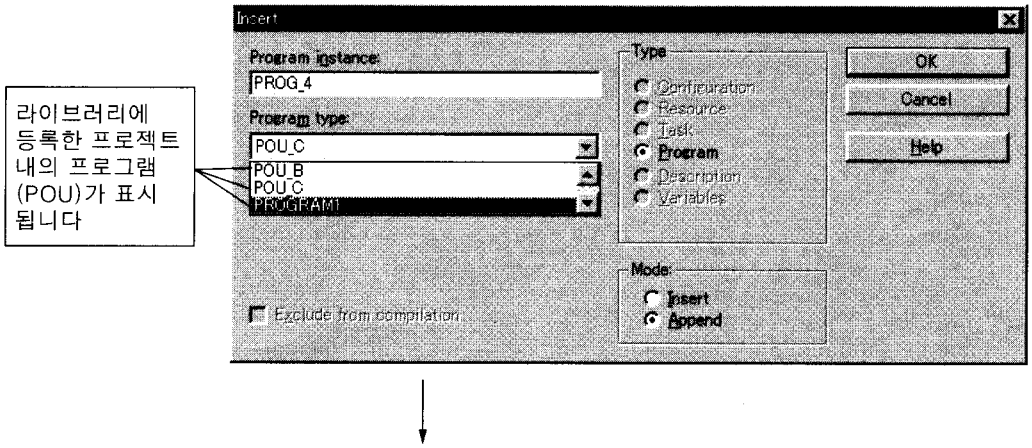


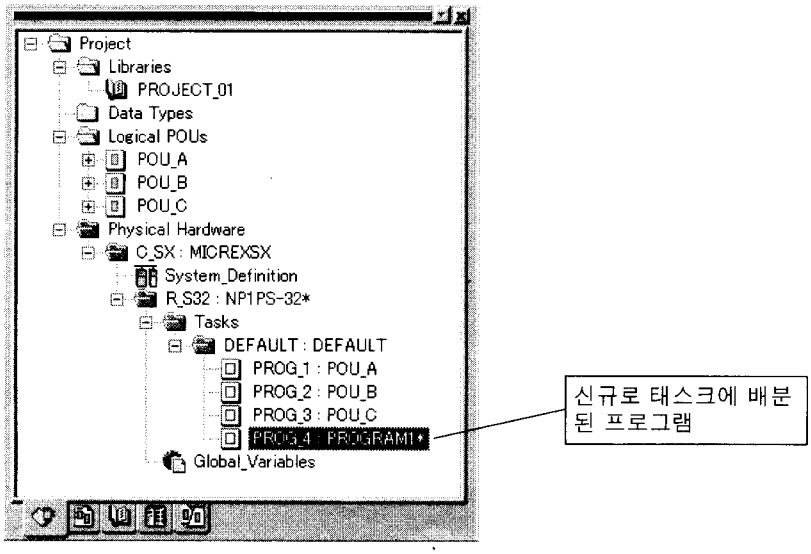
③ 프로그램의 사용방법

라이브러리에 등록한 프로젝트 내의「프로그램(POU)」을, 태스크에 배분하는 것으로 프로그램 동작을 할 수가 있습니다.

다음에 표시하는 프로젝트를 예로 설명합니다.







5.2.ST언어

(1) ST언어의 개요

ST언어로 기술하는 코드는, 식과 문장으로부터 구성됩니다. 또한, 아스테리스크(*)와 괄호를 사용해, 코멘트를 삽입할 수가 있습니다.

① 식이란

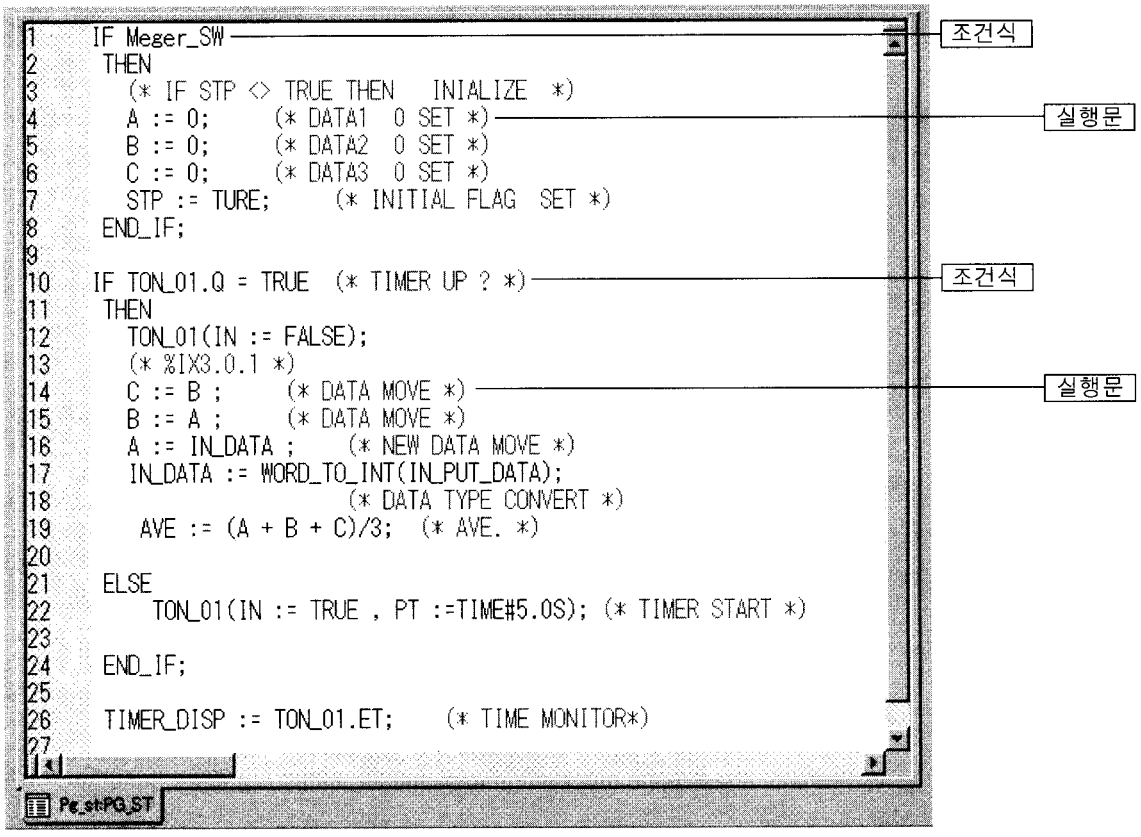
식이란, 문장을 구성하는 요소로, 1개의 값을 돌리는 구문입니다.

② 문장이란

문장은 연산자와 오퍼랜드로부터 구성됩니다. 오퍼랜드는, 정수, 변수, 펑션 또는 다른 표현식의 호출이 있습니다.

연산자와 그 개개의 우선순위는 IEC61131-3에 규정되어 있습니다. 연산자는, 우선순위를 충분히 고려하고, 오퍼랜드를 사용할 필요가 있습니다.

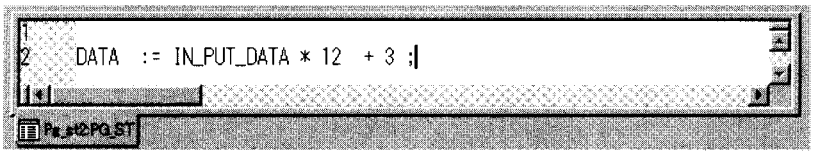
다음은 ST언어에 의한 프로그램의 예입니다.



(2) ST언어의 명령과 대입문

다음 그림에 표시하는 대입문「: =」은, 우측 식의 연산결과를 좌측의 변수로 격납하는 문장입니다.

ST에 있어서 대입문의 구성 예



대입문에 있어서는, 좌측의 변수와 우측의 식이 같은 데이터형인 것이 중요합니다. 데이터형이 다른 경우에는 형 변환을 사용해서, 동일의 데이터형으로 할 필요가 있습니다.
상기의 식은 오퍼랜드와 명령으로부터 구성됩니다. 오퍼랜드는 정수, 변수, 펑션 호출, 또는 다른 식도 사용할 수 있습니다. IEC61131-3에는 오퍼랜드와 접속해서 사용할 수 있는 명령의 리스트가 준비되어 있습니다.
식의 평가는, 연산자에 의해서 우선순위가 결정되며, 그 순위에 따라 연산자를 실행합니다. 연산자의 우선순위는,「(3)ST언어에 있어서 연산자 리스트」에 기재됩니다.

(3) ST언어에 있어서 연산자 리스트

No.	연산	연산자	데이터형	예 또는 설명	예의 값	우선순위
1	호	(식)		(2+3)*(4+5)	45	최상위
2	펑션	펑션명 (파라메터)		LN(A) MAX_INT(X, Y)		
3	누승	**	REAL(저, 지수공)	3.0**4.0	8.1E+1	
4	부호반전	-	INT, DINT, REAL	-V001(V001의 값이 10인 경우)	-10	
5	논리부정	NOT	BOOL, WORD, DWORD	비트 단위의 논리부정 NOT TRUE	false	
6	곱셈	*	INT, DINT, UINT, UDINT, REAL	10*3	30	
7	나눗셈	/	INT, DINT, UINT, UDINT, REAL	6/2	3	
8	승제셈	MOD	INT, DINT, UINT, UDINT	17 MOD 10	7	
9	덧셈	+	INT, DINT, UINT, UDINT, REAL	2+3	5	
10	뺄셈	-	INT, DINT, UINT, UDINT, REAL	4-2	2	
11	비교	<, >, <=, >=	ANY(ANY_BIT를 제거)	4>12	false	
12	등식	=	ANY	T#26h=T#1d2h	true	
13	부등식	<>	ANY	8<>16	true	
14	논리식	&, AND	BOOL, WORD, DWORD	TRUE & FALSE	false	
15	베타적논리화	XOR	BOOL, WORD, DWORD	TRUE XOR FALSE	true	
16	논리화	OR	BOOL, WORD, DWORD	TRUE OR FALSE	true	최하위

(4) ST언어에 있어서 명령문

IEC61131-3에는 다음 형식의 문이 규정되어 있습니다. 문의 키워드, 문의 사용 예 및 그 의미를 아래의 표에 표시합니다.

ST언어에 있어서 명령문

문형	사용예	설명
: =	v:=(a+b+c)/3;	대입문 대입문 우측의 식, 변수, 수치를 대입문 좌측의 변수에 대입합니다. 그 예는 a, b, c의 평균치를 v에 대입합니다.
IF	IF a<b THEN c:=1; ELSIF a=b THEN c:=2; ELSE c:=3; END_IF;	선택문 조건식의 평가결과가 사실일 때 실행문이 실행됩니다. 가짜의 경우는, 문이 실행되지 않지만, ELSE의 다음의 문이 실행됩니다.
CASE	CASE f OF 1: a:=3; 2: a:=4; ELSE a:=0; END_CASE;	선택문 조건식의 값에 의해 실행하는 실행문이 선택됩니다. 변수 또는 조건식「f」의 데이터형은, INT형이어야만 합니다.

FOR	FOR a:=1 TO 10 BY 2 DO
	f[a]:=b;
	END_FOR;
		반복문
		초기치, 최종치, 증감치의 조건에 의해 실행문이 반복 처리됩니다.
		미리 반복 실행 회수를 알고 있는 경우에 사용합니다
		이 예는, 변수「a」를 2씩 증가시켜,「10」에 달하면 종료합니다.

WHILE	a:=1;
	WHILE b>1 DO
	b:=b/2;
	f[a]:=b;
	a:=a+1;
	END_WHILE;
		반복문
		반복 실행 조건이 사실 동안, 실행문이 반복 실행됩니다.
		미리 반복 실행 회수를 알고 있지 않을 경우 등에 사용합니다.
		이 예는, b> 1의 평가결과가 사실일 때 b의 값을 2로 나누어서, 결과를 b에 대입합니다.
		이 문의 조건은, 문의 실행에 앞서 테스트됩니다만, 값이 다른 경우에는, WHILE…DO내의 문이 실행되지 않습니다.

REPEAT	a:=1;
	total:=0;
	REPEAT
	total:=total+a;
	UNTIL a>10;
	END_REPEAT;
		반복문
		반복 실행 조건이 사실로 될 때까지 실행문이 반복 실행됩니다.
		미리 반복 실행 회수를 알고 있지 않을 경우 등에 사용됩니다.
		이 예는, 1에서10의 수치를 가산합니다.
		이 문의 조건은, 문의 실행 후에 테스트됩니다.
		REREAT…UNTIL내의 문은 적어도 1회 실행됩니다.

RETURN	RETURN;	리턴문 불러 낸 펑션, 펑션블록에서 불러내 원 POU로 돌립니다.
EXIT	FOR n:=1 TO 10 BY 1 DO m:=m+n IF m>20 THEN EXIT; END_IF; END_FOR;	반복 종료문 반복 처리를 중지할 때 사용합니다. 이 예는, m의 초기치에 응하여, m+m> 20이 된 때 반복 처리에서 빠져나갑니다.

*위 표의 예에 표시합니다.「소문자」는 변수를 표시합니다.

(5) ST코드의 기술

ST로 코드를 작성하는 경우의 서식에 대한 설명입니다.
프로그램을 기술할 경우, 문법 상, 정해져 있는 서식(기호와 식을 기술하는 순서)과 자유롭게 기술할 수 있는 유저에 맡겨진 스타일(단어와 기호의 배치)이 있습니다.

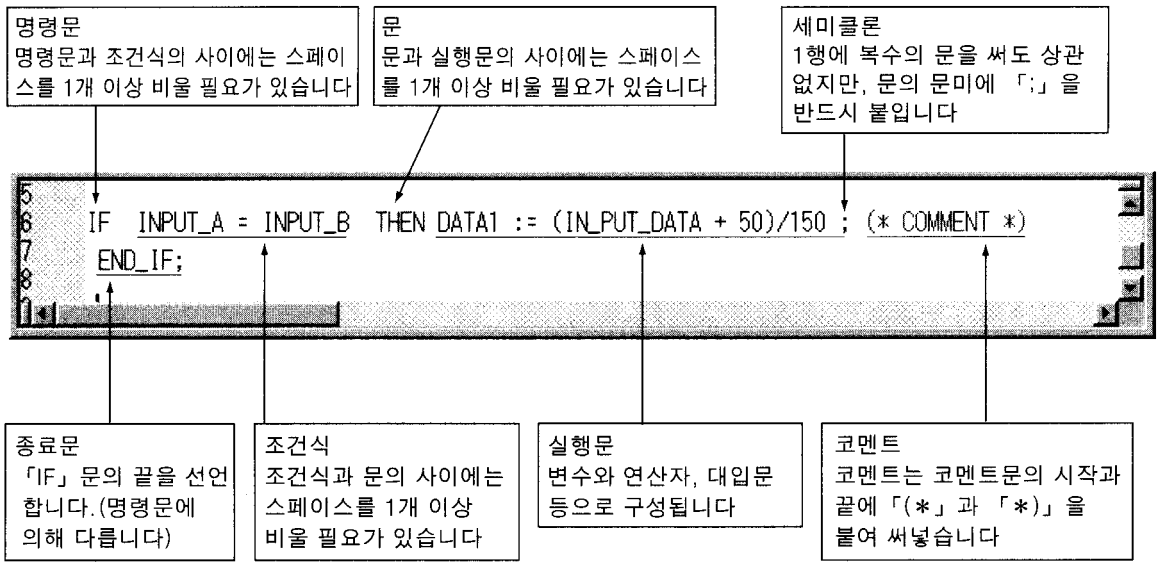
① 코드의 서식

아래에서 표시하는 프로그램은, IF, THEN문을 사용한 프로그램의 일례입니다.

IF, THEN구조

조건식이 사실일 때, 실행문이 실행되며, 조건식이 거짓일 때는, 아무것도 실행하지 않습니다.

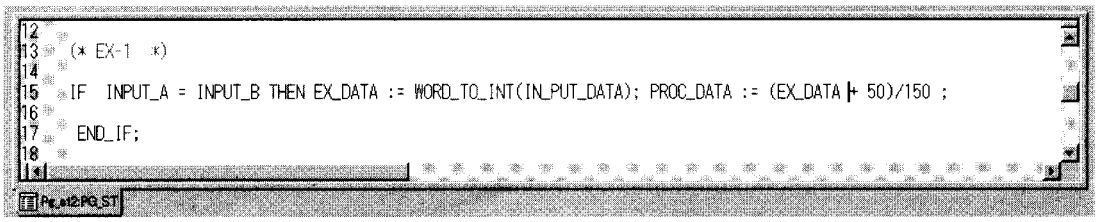
```
IF 조건식    THEN 실행문;
END_IF;
```



② 코드의 기술 스타일

다음 그림에서 표시하는 2개의 프로그램 예는, 기술 스타일은 다르더라도 프로그램 처리는 같게 실행됩니다. 처리 내용마다에 고쳐 행하고, 보기 쉽게 기술할 수가 있습니다.

프로그램 예1



프로그램 예2