

제 7 절

데이터 블록과 함수 블록

제 7 절에서는

- 제어 대상(process) 데이터를 데이터 블록에 저장하는 방법을 설명한다.
- 입/출력과 데이터 블록을 호출하는 방법을 설명한다.
- 함수 블록의 특성을 설명하고 이 블록들이 PLC에서 어떻게 처리되는가를 설명한다.
- 실제 예를 통해 데이터 블록의 입력과 함수 블록 호출을 보여준다.

이 교육 과정을 모두 마친 교육생은

- 데이터 블록의 구조와 기능을 알게 된다.
- 데이터 블록을 입력하고, 출력하고, 호출하고 시험할 수 있게 된다.
- 함수 블록의 가능한 용도를 학습한다.
- 프로그램에서 함수 블록을 호출 할 수 있게 된다.

사용자 프로그램의 구조

사용자 프로그램은 특정한 기술적 과제에 따라 명백히 이해 가능한 부분으로 분리되어야 한다. 각각의 프로그램 부분(소프트웨어 블록)을 특정한 기술적 기능에 따라 분할하는 것은 프로그램을 분명하게 정의하고 쉽게 읽을 수 있도록 할 뿐만 아니라, 사용자 프로그램의 다른 부분과 쉽게 연결될 수 있도록 해준다.

더우기, 이 구조화 기법은 프로그램을 부분별로 시험할 수 있도록 하여 프로그램의 오류의 처리를 쉽게 해 준다.

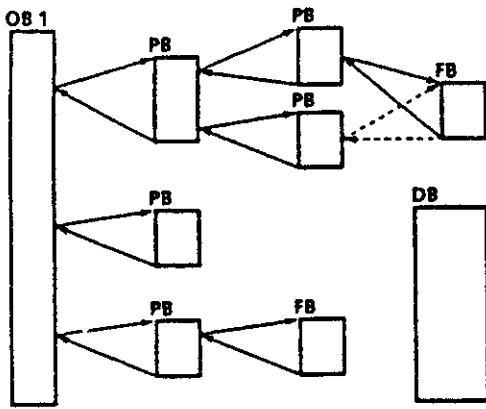
소프트웨어 블록의 형태

서로 다른 형태의 소프트웨어 블록은 STEP 5 프로그램 언어의 구조적 요소를 이루며 프로그램을 구조화 하는 데 사용된다. 다음과 같은 형태의 소프트웨어 블록이 프로그램 내에서 다양한 기능을 제공 해준다 :

- 조직 블록(OB)은 사용자 프로그램을 조직한다.
- 프로그램 블록(PB)은 사용자 프로그램을 특정한 기술적 기능과 관련된 부분으로 구조화 시킨다.
- 데이터 블록(DB)은 사용자 프로그램에서 필요로 하는 데이터를 공급해 주고 임시 저장소로 사용될 수 있는 내부의 데이터 영역이다.
- 함수 블록(FB)은 반복적이거나 복잡한 기능에 사용된다.
- 순차 블록(SB)은 간단한 순차 프로그래밍에 사용된다.

**블록의 괄호치기 깊이
(Nesting depth)**

S5-115U는 최대 16개의 블록(OB, PB, FB, SB)을 임의의 조합으로 괄호를 쳐서 내포할 수(nest) 있다. CPU 943 사용기종 또는 동등 이상에서는 32 블록을 내포할 수 있다. 데이터 블록은 내포 깊이에 영향을 주지 않는다.



Structures of the STEP 5 program:

- OB Organization block
- PB Program block
- FB Function block
- DB Data block

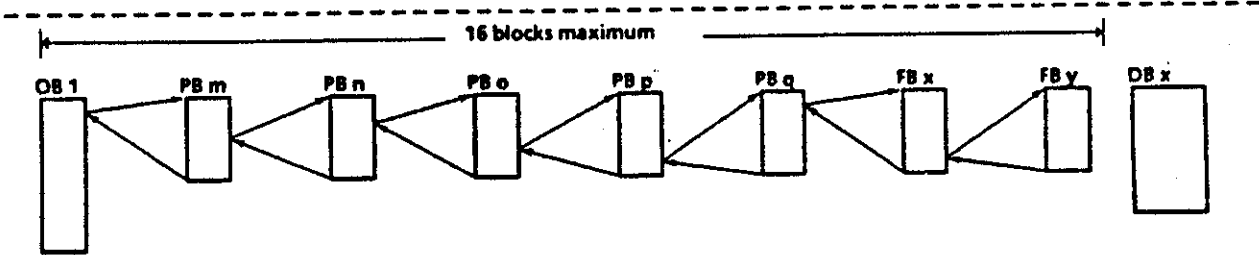


그림 7.1

Notes :

진정한 의미에서 제어 프로그램을 구성하는 PB, FB 그리고 OB 이외에도, 사용자 프로그램에서 데이터 블록이 필요한 경우가 자주 있다. 데이터 블록은 제어 대상의 데이터를 저장하는 데 사용된다.

제어 대상의 데이터 (process data)

- 설정치, 실제 값, 수치 연산의 결과와 같은 수치값
- 입력, 출력이나 플래그 등의 신호 상태
- 시간
- 계수값
- 메시지 텍스트에서 사용되는 문자, 구둑점 그리고 숫자 등등

데이터 워드 (data word)

데이터 블록에서 각각의 저장 공간은 16비트 길이를 가지며 이를 데이터 워드라 부른다. 한 데이터 워드는 예를 들어 16 비트의 고정 소수점 정보를 저장한다. 각각의 데이터 워드는 DW 0...DW 255와 같은 숫자(= 주소)에 의해 인식된다.

데이터 블록의 갯수

최대 255 데이터 블록을 PLC의 기억 장치에 적재할 수 있다. 데이터 블록은 고정된 데이터나 프로그램 검색 도중에 변경되는 데이터를 저장할 수 있다. 고정된 데이터를 포함하는 데이터 블록은 사용자 기억 장치의 EPROM 부분에 저장될 수 있다. 변수를 포함하는 데이터 블록은 사용자 기억 장치의 RAM 부분에 저장되어야 한다.

데이터 형식

데이터 블록에 저장된 정보는 그 의미와 형식이 대단히 다양하다. 이러한 이유에서, "0" 또는 "1" 상태를 갖는 데이터 워드의 16 이진 숫자는 그 의미 즉, "데이터 형식"을 알아야만이 정확하게 해석할 수 있다.

프로그램 작성기를 사용하여 데이터 블록을 입력할 경우, 교육생은 데이터 표현에 대하여 유효한 형식을 사용해야만 한다. 예를 들어,

Arrangement of the software blocks in PC memory

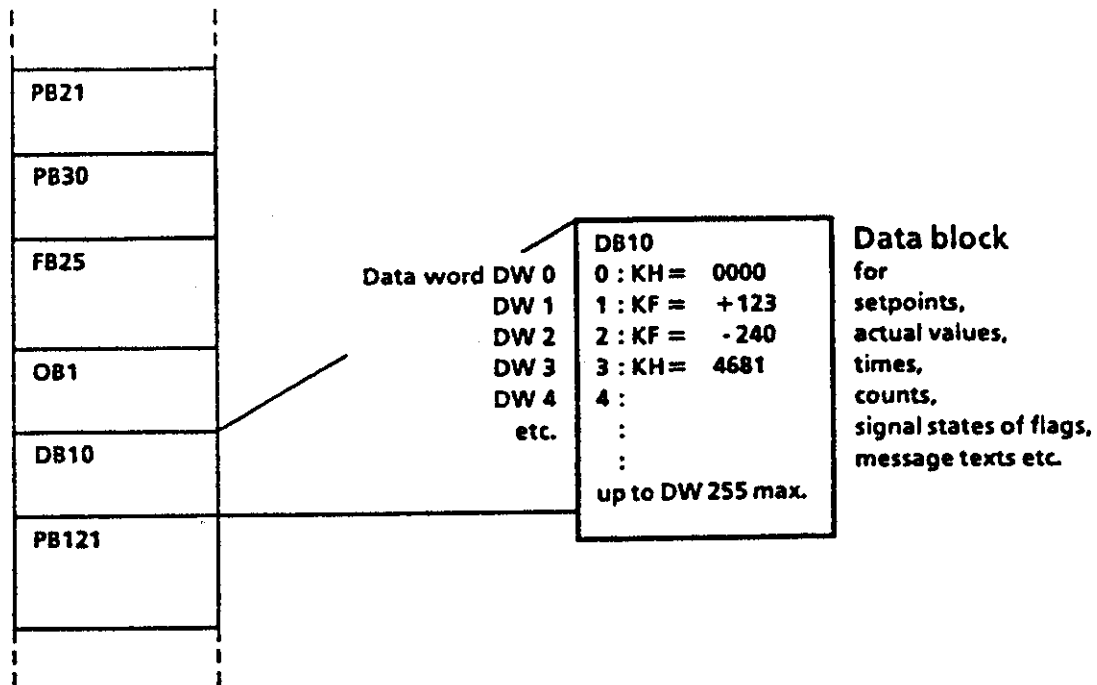


그림 7.2

- KM = 비트 제한(16 비트의 상태)
- KH = 16진 숫자
- KY = 바이트; 0에서 255까지의 2 절대 2진값
- KF = 고정 소수점 데이터
- KG = 부동소수점 데이터(더블워드)
- KS = 문자(영문자, 숫자, 구두점, 부호)
- KT = 시간
- KC = 카운트값(계수값)

예 제 :

0010011000111111
 263F
 38, 63
 + 9791
 + 1356123 + 12
 ABCDEF;123456;-+&?
 055.2
 234

-참 고

부동 소수점 데이터는 유리수(fractional number)이다(양수 또는 음수). SIMATIC-S5 제어기에서 이 데이터는 32 비트 즉, 한 개의 더블워드를 사용한다. 이러한 이유로, 부동 소수점 데이터는 더 강력한 상위 기종의 SIMATIC-S5 PLC에서만 표현/처리될 수 있고 S5-115U에서는 사용할 수 없다.

Notes :

Input:

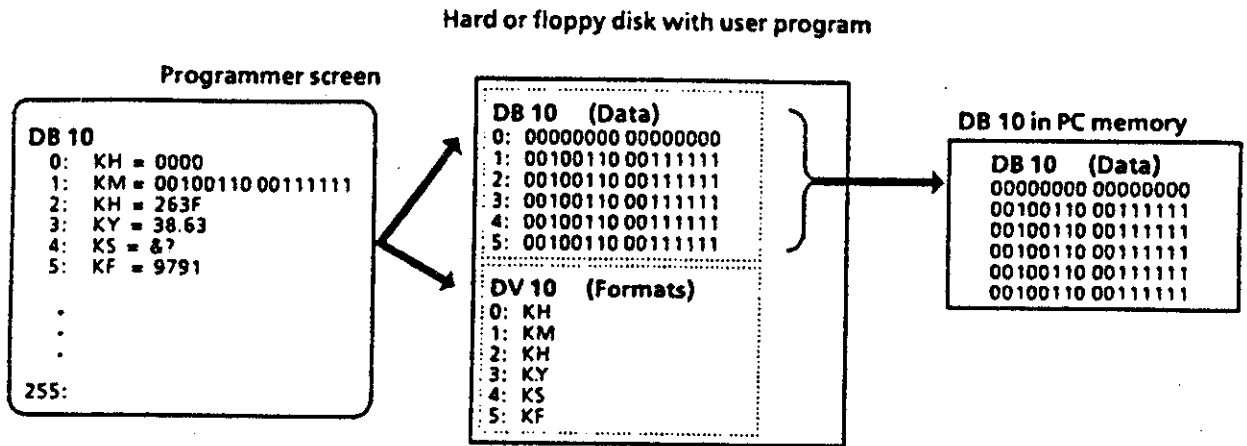


그림 7.3a

입력한 내용 (entry)

사용자 프로그램의 다른 블록과 마찬가지로, 데이터 블록은 프로그램 작성기의 하드 디스크에 저장된 후 PLC로 전송된다. 데이터 블록을 입력할 때, 교육생은 필요한 저장 공간 (데이터 워드)의 갯수를 정해 주어야 한다.

데이터 헤더 DV

데이터 블록 입력 도중에 각각의 데이터 워드에 지정된 데이터 형식은 하드 디스크나 플로피 디스크상의 "데이터 헤더"(data header)로 저장된다. 프로그램 작성기는 입력된 데이터를 주어진 형식에 의해 16 비트 패턴으로 변환한다. 이 비트 패턴이 최종 단계에서 PLC의 사용자 기억 장치에 기억되는 비트 패턴이 된다.

Output (display):

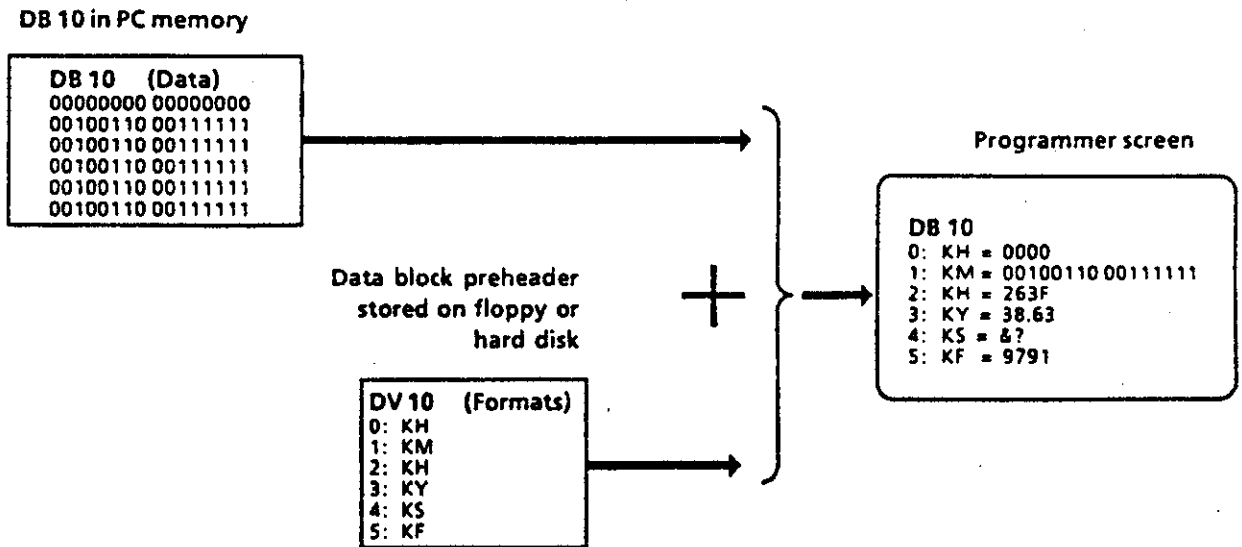


그림 7.3b

화면 표시

PLC 기억 장치에서 데이터 블록을 화면 표시할 때, 프로그램 작성기는 각각의 데이터 워드의 비트 패턴을 하드 디스크나 플로피 디스크 상의 데이터 헤더에 저장된 형식에 맞추어 사용자가 직접 읽을 수 있는 "텍스트" 형태로 변환한다.

하드 디스크나 플로피 디스크의 헤더와 일치하지 않으면, 프로그램 작성기는 생략시(default) 형식의 데이터를 화면에 표시한다.

**입력한 내용의 화면 표시
(entry display)**

데이터 블록을 입력할 때, 사용자가 친숙하고 쉽게 이해할 수 있는 데이터 형식을 선택하도록 유의하여야 한다. 데이터 블록은 데이터만 저장하고 제어 명령은 저장하지 않으므로 BE(블록 끝) 명령문으로 끝나지 않는다.

실습 문제

- * PLC를 STOP으로 설정한다.
- * 그림에 나와있는 DB 34 를 하드 디스크에 저장한다 :
INPUT DEVICE : *FD* BLOCK : *DB34*

각각 데이터 워드 입력을 CR 키를 눌러 종료하고, DB 입력은 Enter 키를 눌러 종료한다.

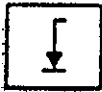
- * 데이터 블록 DB 34를 PLC로 전송하고 PLC 기억 장치에서 DB 34를 읽어온다.

데이터 형식을 저장하고 있는 데이터 헤더가 하드 디스크에만 저장되어 있으므로, 데이터 블록이 PLC 기억 장치에서 읽혀질 때 하드 디스크도 동작한다(적색 LED가 켜진다).

수 정

교육생이 입력한 데이터 워드를 수정하려면 그 위에 다시 겹쳐쓰면 된다.

**수직으로 확장 키
(expand vertical)**



데이터 블록을 높이려면 블록의 끝 부분에 데이터 워드를 추가하거나 데이터 블록 중간에 데이터 워드를 삽입한다. 이때 "수직 확장 키"를 사용한다.

**문자 삭제 키
(delete character)**



"문자 삭제 키"를 누르면 커서가 가리키는 데이터 워드가 지워진다.

**프로그램에 주석 달기
(documentation)**

프로그램에 주석 달기의 옵션은 여러가지 블록 타입에 따라 다양하며, 식별자만이 붙여진다. 예를 들어, 각각의 DB에 블록 헤더와 각각의 한 줄(a line comment)을 지정하고, 즉 각각의 데이터 워드에는 주석문을 지정하는데, 앞의 경우는 세그먼트 헤더와 유사하며, 두번째 경우는 STL에서의 명령문의 주석과 유사하다.


```

DB 34                B:S15A@@ST.SSD                LEN=14

                    DB for multiplication

0:   KH = 0000
1:   KM = 00000000 00000000
2:   KF = 2                Factor 1
3:   KF = 0
4:   KF = 0
5:   KF = 0
6:   KF = +15            Factor 2
7:   KF = 0
8:   KF = 0
9:
    
```

그림 7.4

주석
(Comment)



COM 키를 눌러 블록 헤더(블록 제목이라고도 한다)를 입력하고, 헤더를 입력하고, 다시 CR 키를 눌러 OUTPUT 모드로 돌아간다.

그후에 데이터 워드 DW 2와 DW 6에 대한 주석문의 줄(line comment)을 입력한다. "오른쪽 화살표"(right arrow) 키를 눌러 커서를 주석문 필드로 옮긴 후, 주석문을 입력하고, CR 키를 눌러 OUTPUT 모드로 돌아간다.

주석문 입력을 마치려면 Enter를 누른다.

참고

프로그램 작성기는 DB 34와 DC 34에 대한 블록 헤더와 주석문을 하드 디스크에 저장한다.

다양한 데이터를 가진
데이터 블록

정수 데이터 외에도, 설정치, 실제값과 같은 변수 데이터 등도 프로그램의 제어에 의하여 데이터 블록에 전송되거나 데이터 블록에서 읽어 올 수 있다.

프로그램에서, 데이터 워드는 적재(Load) 및 전송(Transfer) 동작을 사용하여 접근할 수 있다.

L DW 1 데이터 워드 1을 읽는다
T DW 1 데이터 워드 1을 쓴다

데이터 블록의 호출

어떤 데이터 블록을 처리하려면 먼저 그 데이터 블록을 호출해야 한다. 데이터 블록은 아래의 명령문으로 호출한다 :

C DB x 데이터 블록 호출

PLC의 기능
STATUS VAR

STATUS VAR 기능은 프로그램이 스캐닝을 하고 있는 도중에 임의의 연산수(예를 들어, 입력, 출력, 플래그, 타이머, 카운터, 출력 워드, 데이터 워드 등)을 선택하고 이를 화면에 표시할 수 있게 해준다. 대부분의 경우에, 출력 형식은 사용자가 정해 주어야 한다.

STATUS VAR는 PLC의 기능이며, F4 = PC FCT와 F4 = STAT VAR를 사용해 실행할 수 있다.

참 고

데이터 워드를, STATUS VAR를 사용하여 나열하기 전에, 해당 데이터 블록 번호를 지정해 주어야 한다.

주 의 !

데이터 블록은 주기적으로 처리되지 않기 때문에 STATUS 시험 기능으로는 화면 표시할 수 없다.

실습 문제 :

- * PLC를 STOP으로 설정한다.
- * OB 1의 PB 32를 호출하고 PLC 기능 STATUS VAR(F4/F4)를 사용하여 프로그램 검색을 확인해 본다.

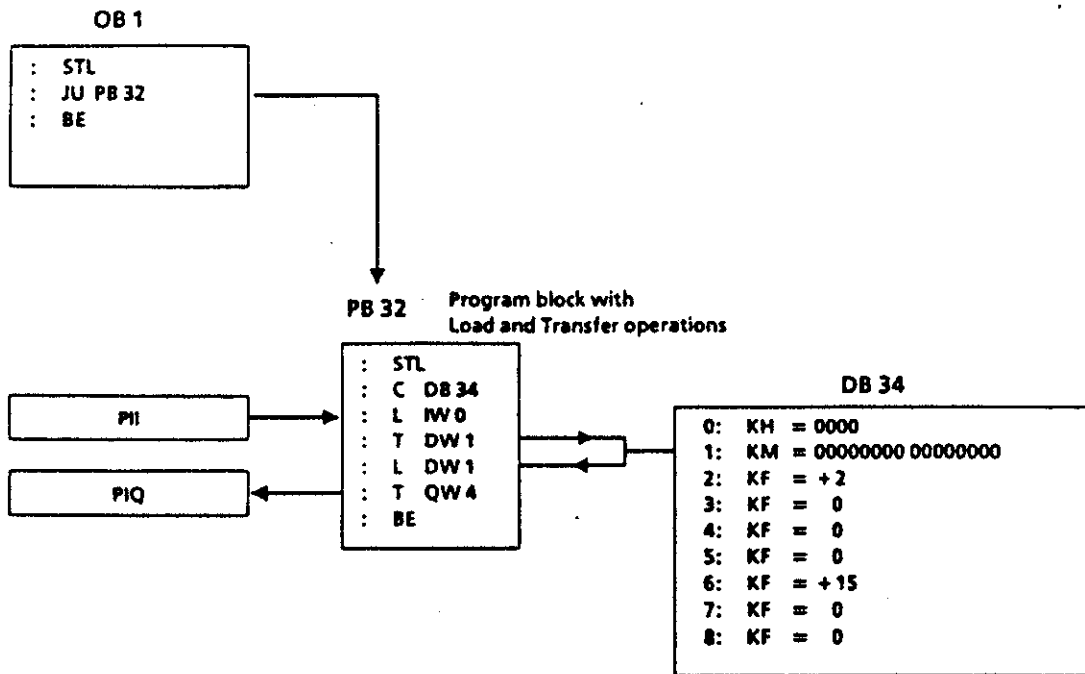


그림 7.5

- 아래에 나열한 데이터 블록 번호, 연산수와 형식을 입력한다. 사용자는 특정한 형식을 필요하다면 변경할 수 있다.

연산수 :	형 식 :
DB 34	
DW 0	KH
DW 1	KM
DW 2	KF
DW 3	KF
DW 4	KF
DW 5	KF
DW 6	KF
DW 7	KF
DW 8	KF

- Enter 키를 누르면 원하는 데이터를 프로그램 작성의 화면에 표시된다.

참 고

STATUS VAR 기능이 다시 호출되면 최종적으로 입력되었던 연산수의 목록이 화면 표시된다. Break를 누르면 수정이나 삭제, 삽입 등과 같은 변동을 일으킬 수 있다.

자주 되풀이하여 사용되는 함수와 복잡한 함수는 함수 블록으로 프로그램 되어 있다.

표준 함수 블록

지멘스(Siemens)에서는 일반적으로 사용되는 다수의 표준 함수를 위해 표준 함수 블록 팩키지를 디스켓에 담아 제공한다. 이들 표준 함수 블록은, 보고(reporting), 페루프 제어, 게루프 제어, 리스팅 등의 디지털 함수를 위한 사용자 프로그램을 빠르고, 신뢰성 있고, 경제적으로 개발 할 수 있도록 지원해 준다. 자세한 정보는 카탈로그 ST57을 참조하시오.

표준 함수 블록 FB 240에서 FB 243까지는 곱셈, 나눗셈 및 수의 진법 형식의 변환과 같은 디지털 기능을 가지며 S5-115U PLC의 CPU에 이미 설치되어 있다.

사용자 함수 블록

다음과 같은 경우에 교육생은 자신의 함수 블록을 STL 형식으로 작성할 수 있다.

- a) STEP 5 프로그램 언어의 모든 연산(시스템과 보조 연산 포함)을 활용하고자 할 때.
- b) 프로그램에 자주 되풀이 사용되는 함수(자유롭게 매개 변수를 지정할 수 있는 함수 블록)를 위한 블록을 만들고자 할 때.

사용자 함수 블록 개발은 S 16 과 S 20 과정에서 다루고 있다.

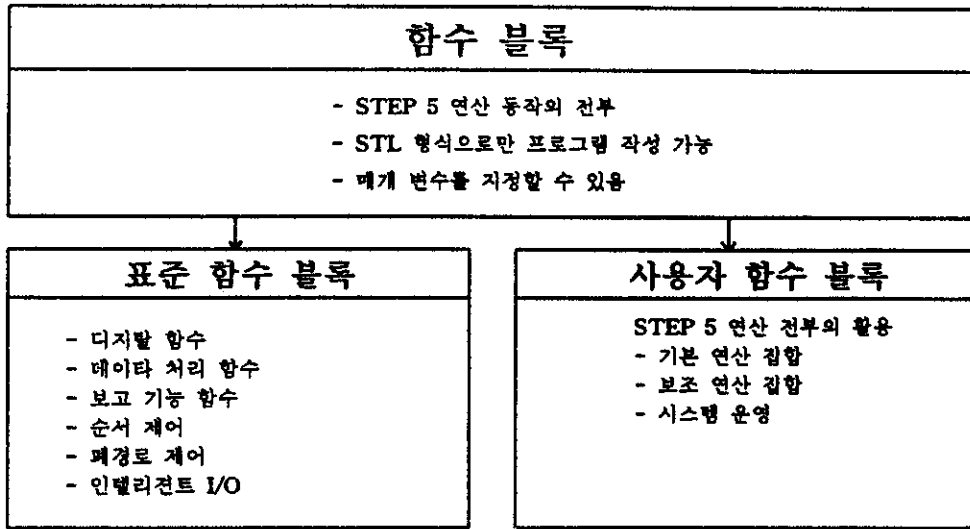


그림 7.6

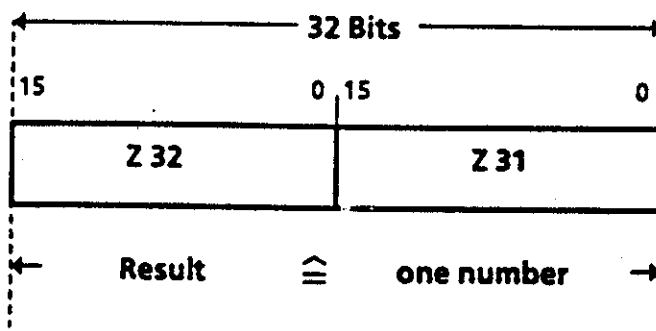
Notes :

표준 함수 블록의
"초기화"

이미 존재하는 표준 함수 블록을 호출할 때, 이 함수 블록의 모든 입력과 출력(정형화된 연산수)은 프로그램에서 그 함수 블록(실제 연산수)이 실행하고자 하는 그 연산수에 지정되어야 한다.

표준 함수 블록
FB 242의 기능
(MUL: 16)

표준 함수 블록 FB 242는 S5-115U에 통합 내장되어 있는데, 두개의 고정소수점 데이터 Z1과 Z2를 곱하여 그 결과를 두개의 16 비트 워드 Z32와 Z31에 하나의 32 비트 이진 수로 저장한다(아래 그림 참조) :



곱셈 결과가 0인 경우, 이 결과는 FB 242의 출력 Z3에 플래그 된다.

표준 함수 블록

표준 함수 블록은 광범위하게 응용 가능한 프로그램을 포함하고 있으므로, I 1.0, Q 4.7, F 2.0, DW 5 등의 특별한 연산수를 사용하지 않고, Z1, Z2, X1, X2와 같은 일반적인 연산수를 사용한다. 이러한 연산수는 표준 함수 블록에 대한 입/출력을 의미한다.

Standard function block for multiplication
(integrated in the S5-115U)

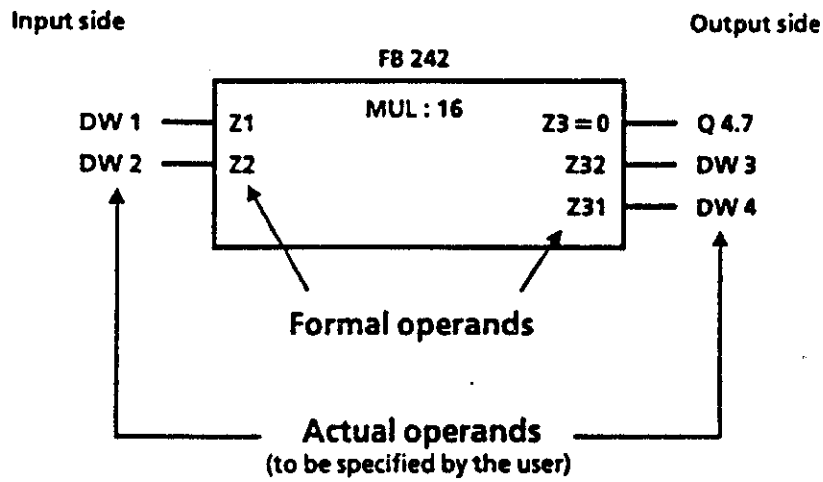


그림 7.7

Notes :

함수 블록의 호출

만약 함수 블록이 사용자 프로그램의 다른 블록에서 호출된다면, 각각의 기억 장치 즉, PLC 내부나 디스크 상에서 사용 가능해야 한다. 그렇지 못할 경우, 프로그램 작성기는 다음 메시지를 화면에 표시 한다 :

CALLED FB DOES NOT EXIST.(호출된 함수 블록 FB는 없음)

표준 함수 블록이 보편적인 프로그램을 포함하고 있으므로, 사용자 프로그램 내부에서 여러번 사용될 수 있다. 이 예제에서, 곱셈 블록 FB 242는 두개의 다른 곱셈을 하는 데 사용된다.

표준 함수 블록이 호출될 때 새로운 매개 변수를 지정하는 것은 사용자 책임이다. 즉, 교육생은 모든 연산수에 값을 지정하여야 한다 (그림 참조).

다른 소프트웨어 블록과는 달리, 함수 블록은 CSF, LAD, STL과 같은 모든 표현 방식을 이용해서 호출할 수 있다.

STL에서는 JU FB ... 또는 JC FB ... 명령을 사용하여 호출할 수 있다.

특정한 함수 블록의 호출



CSF와 LAD형식에서는 FB 키와 F1(=JU FB...)나 F2(=JC FB...)를 이용하여 호출할 수 있다.

CSF나 LAD에서 조건부/무조건 FB 호출은 다음과 같이 할 수 있다:



참 고

절대 블록 호출과는 달리, 조건부 호출은 호출을 하기 전(예. A I 1.0)에 조건이 만족하면(예를 들어 RLO = "1") 실행된다.

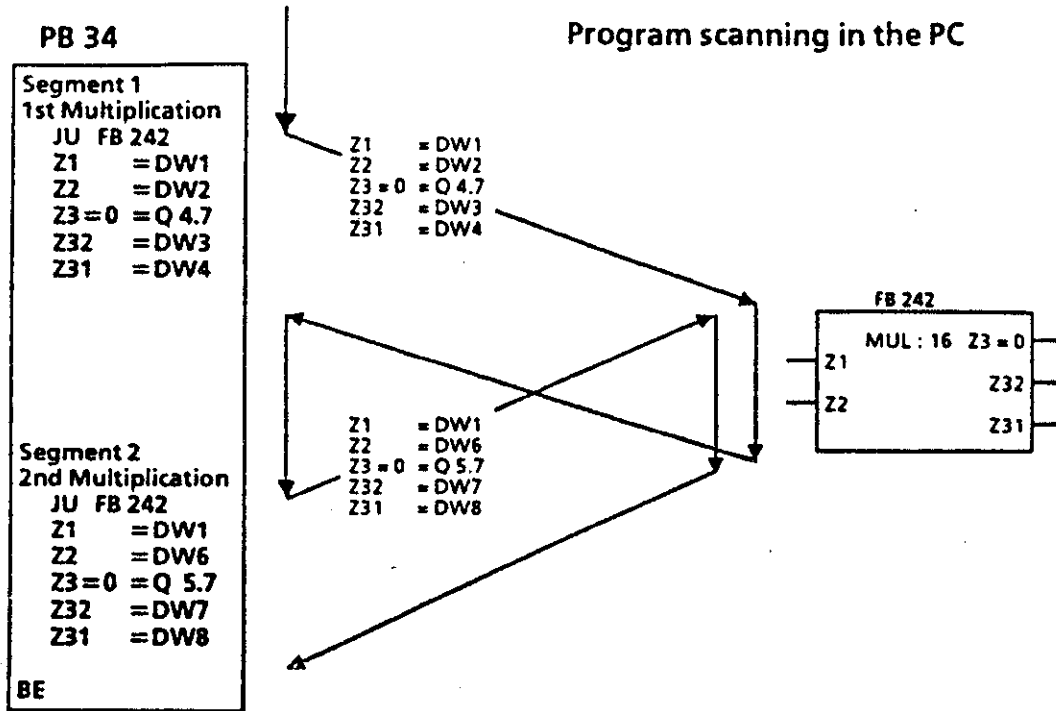


그림 7.8

프로그램 실행

PB 34의 세그먼트 1에서 첫 FB 242 호출되어 처리될 때, 프로세서는 프로그램 블록 PB 34를 떠나서 FB 242로 분기한다. 이러한 동작 중에, 프로세서는 데이터 워드 DW 1 과 DW 2에 대해 곱셈을 수행하도록 실제 연산수 및 형식 연산수의 목록을 참고한다.

FB 242를 처음 검색이 끝나면 PB 34의 세그먼트 2로 다시 분기해 간다. 두번째 곱셈을 하기 위해 FB 242가 다시 한번 호출 되므로, 프로세서는 PB 34를 다시 한번 떠나서 FB 242를 두번째로 실행한다. 그러나 이번에는 두번째 곱셈에 대한 새로운 실제 연산수가 사용된다. 그러므로, 이번에는 데이터 워드 DW 1과 DW 6에 대해 곱셈을 수행한다.

FB 242를 두번째 수행한 후, 프로세서는 PB 34로 다시 돌아온다.

문제 정의

입력 IB 0에 설정된 임의의 수를 한번은 2로, 한번은 15로 곱한다. 이 목적을 위해, 2와 15가 이미 DB 34의 데이터 워드 DW 2와 DW 6에 저장되어 있다. IB 0에 설정된 값은 PB 34에 의해 세그먼트 1의 DW 1에 전송된다. 두 곱셈의 결과는 데이터 워드 DW 3과 DW 4 또는 DW 7 또는 DW 8 각각에 저장된다.

실습 문제

- * PLC 34를 PLC의 기억 장치에 CSF 형식으로 저장한다. FB 242는 세그먼트 2에서 CSF 형식으로 호출하고 세그먼트 3에서는 STL 형태로 호출하고자 한다.

- * 세그먼트 1의 입력(STL)

DB 34가 PB 34의 실행 초기에 호출된다는 사실을 주의하시오. 데이터 블록 DB 34는 지난번 7-8쪽의 연습문제를 다룰 때 이미 PLC의 기억 장치에 사용 가능하다.

함수 블록 호출



- * 세그먼트 2의 입력(CSF)

FB 키를 누르고 그다음에 메뉴에서 무조건 호출을 선택하고 기능 키 F1 = FB-JU를 누른다. 기호 위쪽에 호출되는 함수 블록 즉, FB 242의 연산수 식별자와 매개 변수를 입력하고, 그림에 나온 바와 같이 PB의 입력과 출력을 실제 연산수로 초기화한다.

- * 세그먼트 3의 입력(STL)

: STL 명령문으로 STL로 바꾼 뒤 JU FB 242 명령을 사용해, FB 242를 호출한다. 화면에는 FB 242의 이름 즉, MUL: 16이 출력된다. CR 키를 누른 뒤 다음 그림에 나온것 처럼 각각의 연산수에 실제 연산수를 지정한다.

- * 구성 블록 OB 1의 PB 34를 호출한다.

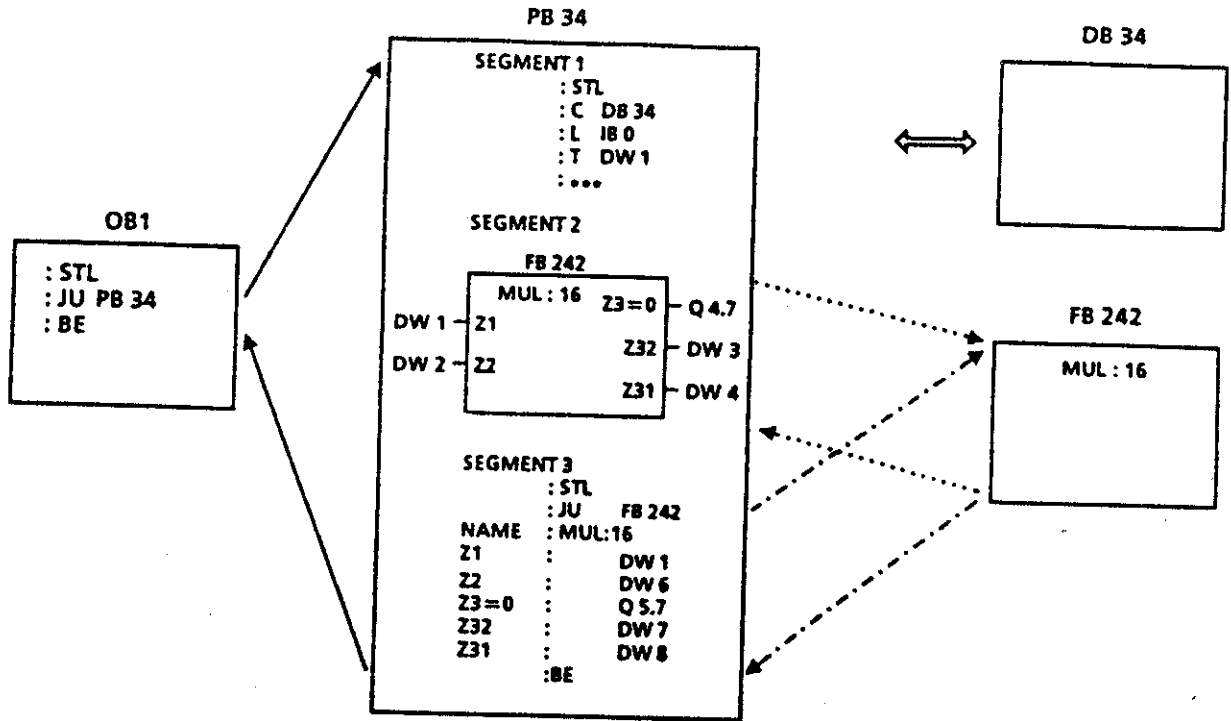


그림 7.9

Notes :

PLC 기능
CONTROL VAR

STATUS VAR와 마찬가지로, 교육생은 임의의 연산수를 선택하고 그들의 신호 상태를 출력하기 위해 CONTROL VAR를 사용 할 수 있다. STATUS VAR의 범위를 지나서, CONTROL VAR를 사용하면 사용자가 선택한 연산수를 바이트나 워드 기준으로 수정할 수 있다. 즉, 제어 대상(프로세스)의 상태를 모의(simulate)할 수 있다. CONTROL VAR는 PLC 함수이며, F4 = PC FCT, F6 = CTRL VAR를 이용하여 호출할 수 있다.

실습 문제(계속)

- * 프로그램을 CONTROL VAR를 이용하여 시험하고, 다음 연산수를 관찰한다 :

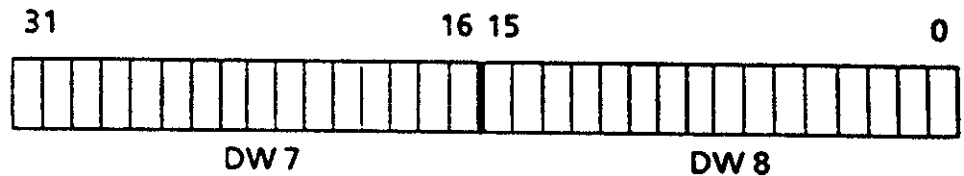
연산수 :	형 식 :
DB 34	
DW 1	KF
DW 2	KF
DW 4	KF
DW 1	KF
DW 6	KF
DW 8	KF

IB 0은 0에서 255까지의 값만 지정할 수 있으므로, 그 결과는 0에서 3825까지의 범위에 들게 된다. 즉, DW 4나 DW 8과 같은 하나의 워드로 표현된다.

- * DW6의 팩터 15를 1024로 겹쳐쓴다. CONTROL VAR가 동작 하는 도중 BREAK를 누르면 된다. 다음과 같은 메시지가 화면 표시된다 :

OPERANDS CONTROL PROCESS IMAGE
(연산수 제어 대상의 디지털 입력)

- * 커서를 DW 6으로 움직이고, 1024를 입력하고, Enter를 누른다.
- * IB 0의 값을 64로 설정하고 곱셈을 시험해 본다. DW 7과 DW 8에 있는 결과를 KM 형식으로 값을 계산해 본다.



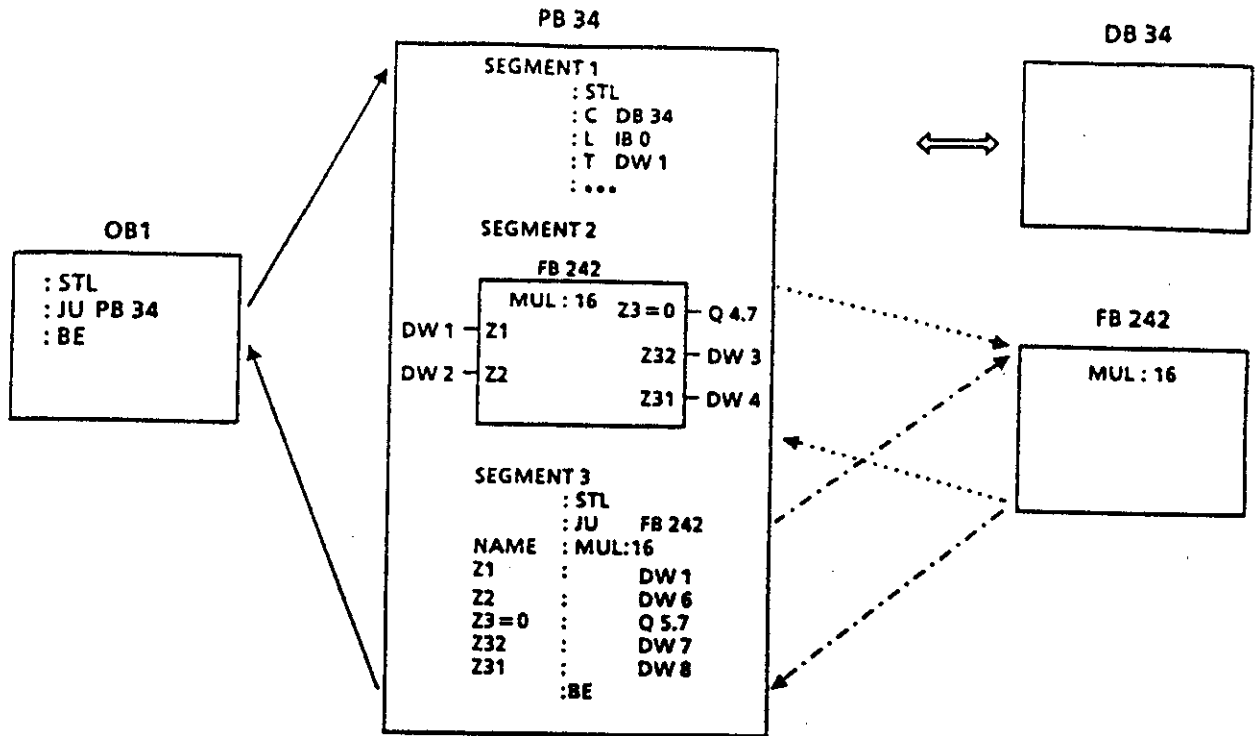


그림 7.10

Notes :