

제 12 절

프로그램의 문서화

제 12 절에서는

- 프로그램 작성기 PG 685 또는 PG 750의 프로그램의 문서화 방법을 설명하고
- 연산수의 기호 주소 지정 및 절대 주소 지정하는 법을 설명한다.

교육생은

- 프로그램 작성기의 다양한 문서화의 수단을 알게 된다.
- 주석문을 입력할 수 있다.
- 연산수에 기호 주소와 절대 주소를 지정할 수 있다.
- 완벽한 프로그램 문서화의 필요성을 이해할 것이다.

PG 685 또는 PG 750 프로그램 작성기는 광범위한 표준 문서 능력을 갖는다.

- 기호 번지 지정 방식
(SYMBOLS EDITOR(기호 편집기) 패키지)
- 주석문의 다른 형태
(LAD, CSF, STL 패키지)
- 표제 블록(각주 기능, footer)
(S5 유틸리티 프로그램)
- I/Q/F 참조 목록, 교차 참조 목록, 프로그램 개요 화면 표시
(XRF,COMP, REW 패키지)

프로그램의 문서화

좋은 문서화는 프로그램 또는 시스템의 이해를 쉽게 하는데 있어서 필수 불가결하며, 또한 시스템 시운전과 고장 수리에 아주 유용하다.

기호 연산수

절대 명칭(예 Q 5.4) 또는 기호 명칭(예 MOTOR)은 연산수를 읽거나 쓰는데 사용될 수 있다. 기호 명칭은 기호 편집기 패키지를 이용하여 지정 목록에 기입된다.

연산수의 주석

지정 목록에서 각각의 지정은 하나의 연산수 주석문을 최대 40개 문자를 사용하여 보충할 수 있다.

LAD, CSF, 그리고 STL 패키지는 대략적으로 다음의 세 가지로 나뉘어 질 수 있다 :

- 소프트웨어 블록 주석과 STEP 5 프로그램 주석
- 데이터 블록 주석
- 블록과 독립적인 주석

세그먼트 헤더

최대 32문자의 헤더는 LAD, CSF, STL의 각 세그먼트에 지정될 수 있다.

명령문 주석

STL에서 모든 명령문은 최대 32 문자까지의 명령문의 주석을 지정할 수 있다.

세그먼트 헤더와 명령문 주석은 주석 블록에 저장된다.

세그먼트 주석

세그먼트 주석은, 예를 들면, 세그먼트를 나타내기 위해 사용된 대략적으로 DIN A4 규격으로 2쪽(page) 분량으로 작성할 수 있는 주석문이다. 이 세그먼트 주석은 문서 블록에 저장되어질 수 있다.

데이터 블록 주석

데이터 블록 주석은 CSF, LAD와 STL에서의 세그먼트 주석에 지정된다. 단지 수식어(descriptor)만이 다르다 ; 블록 헤더 대신 세그먼트 헤더로, 라인 주석 대신 명령 주석으로 그리고 블록 주석 대신 세그먼트 주석으로.

Type of text	Characters	Block or file	Remarks
Symbolic operands	8 - 24	Symbols file	Input in assignment list with symbols editor e.g. Q 5.4 - K 1
Operand commentary	1 - 40	Symbols file	Input with symbolic operand e.g. Q 5.4 - K 1 CONVEYOR MOTOR
Segment header Block header	32	Comment block OC, PC, SC, FC, DC	Input directly in CSF, LAD, STL or in comment block e.g. PC 52, SEG 1 = FINAL ASSEMBLY
Statement commentary Line commentary	32	Comment block OC, PC, SC, FC, DC	Input directly in STL or in comment block e.g. PC 52, LINE 4 = FLAG FOR BAY 1
Segment commentary Block commentary	2 Kbytes	Documentation file #OBDO.nnn, #PBDO.nnn, #SBDO.nnn, #FBDO.nnn, #DBDO.nnn	Input directly in CSF, LAD, STL or in documentation file e.g. #PBDO.052 for PB 52
Plant commentary	2 Kbytes	Documentation file #name (8 char. rmax.)	Input/output on floppy disk or hard disk e.g. #CONVBELT

그림 12.1

제어 대상의 주석

제어 대상의 주석은 순수한 문장 파일(문서화 파일)이며, 따라서 블록에 종속된다. DIN A4 규격으로 2쪽 분량까지 작성할 수 있고, 예를 들면, 전체 시스템을 설명하는데 이용할 수 있다.

표제 블록 (각주)

각주(footer)는 프린트 출력을 정의하는데 사용되고, 인쇄 지면의 각 쪽(page) 하단에 인쇄된다.

KOMDOK

KOMDOK 소프트웨어 패키지는 개량되고 사용자에게 친숙한 문서화 소프트웨어이며 옵션으로 이용 가능하다. KOMDOK에서는 아래의 사항들을 간결하고, 명쾌한 형식으로 프린터 또는 파일로 출력된다.

- STEP 5 프로그램을 LAD, CSF 및 STL 양식으로,
- 주석문
- 지정 목록
- 교차 참조 목록
- 참조 사항 목록
- I/Q/F(입력/출력/플래그) 참조 목록
- 제어 목록
- 프로그램 구조

그림은 포괄적인 문서화의 보기를 보여준다 :
 보기에 포함된 것은 다음과 같다.

- ① 기호 주소 지정 방식
- ② 연산수 주석
- ③ 세그먼트 헤더
- ④ 명령문의 주석
- ⑤ 세그먼트 주석

주 의 ! 주석은 사용자가 쉽게 이해하기 위한 것이며, 따라서 PLC 또는 EPROM 보조 모듈로 전송되지 않는다.

주석 또는 기호 번지 지정 방식이 "미리 설정"(생략시) 형식으로 지정되었다면, 모든 주석과 지정 목록은 인쇄 출력된다.

프린터는 아래와 같은 PTR : (프린터) 명령문 라인을 완료하면 동작한다 :

주 의 ! PTR :

- * 표준 프린터 출력
- 1 표준 폰트(Font) 출력
- 2 왼쪽 마진을 가지고 좁은 폰트로 출력
- 3 극히 좁은 폰트(DIN A4만이)로 출력

STL형식의 기호 번지 지정 방식의 경우에
 ! * 2와 3만을 사용하라. 명령문 주석은 1의 경우에 생략된다.

1 PB53 -PREASBL
 SEGMENT 1 0000
 3 - PRE-ASSEMBLY BAY 1 -
LEN=38
SYM

5SEGMENT COMMENT.....
 = THIS IS AN EXAMPLE OF VERY ELABORATE DOCUMENTATION, SYMBOLIC
 = ADDRESSING (1), OPERAND COMMENTS (2) SEGMENT HEADER (3), OR STATEMENT
 = COMMENTS (4) AND SEGMENT COMMENTS (5) ARE COMBINED IN STL REPRESENTATION.

0000	:A	Q	5.3	-FIN. ASBL	1	4	SET FLAG FOR BAY 1
0002	:AN	Q	5.4	-MOTORON			IF FINAL ASSEMBLY IND. LAMP ON,
0004	:A	I	1.4	-CONFREE			MOTOR IS RUNNING AND CONVEYOR IS
0006	:S	F	1.1	-ASBAY 1			ENABLED.
0008	:A	Q	5.0	-INDBAY 1			SET FLAG FOR BAY 1. IF
000A	:A	I	1.1	-BAY 1 OK			LAMP AND READY SIGNAL FOR
000C	:R	F	1.1	-ASBAY 1			BAY 1 ARE ON. RESET FLAG
000E	:A	F	1.1	-ASBAY 1			FOR BAY 1.
0010	:A	Q	4.1	-ONSTATE			IF FLAG FOR BAY 1 IS SET AND
0012	:*	Q	5.0	-INDBAY 1			PLANT IS ON. SET IND. LAMP FOR BAY 1.
0014	:***						

Q	5.3	= FIN. ASBL	2	IND. FOR FINAL ASSEMBLY
Q	5.4	= MOTORON		CONTACTOR F. MOTOR DRIVE
I	1.4	= CONFREE		"CONVEYOR FREE" (PUSHBUTTON)
F	1.1	= ASBAY 1		FLAG F. PRE-ASS. BAY 1
Q	5.0	= INDBAY 1		IND. FOR BAY 1
I	1.1	= BAY 1 OK		BAY 1 READY (PUSHBUTTON)
Q	4.1	= ONSTATE		ON STATE INDICATOR

그림 12.2

Notes :

연산수에는 절대 번지(예 1.0.1) 또는 기호 번지(예 -S1E)가 주어진다. 교육생은 기호 번지를 가지고 연산수를 화면에 표시하기 전에 지정 목록에 기호로 절대 번지를 지정해야만 한다. PLC는 이 지정 없이 기호 번지에 관련된 STEP 5 동작을 수행할 수 없다.

기호 파일

프로그램 작성기는 지정 목록을 그 자체의 "기호 파일"(ZULI 파일)에 기록한다.

주 의 !

PG 675는 디스켓의 데이터 블록에 기호 연산수로 구성된 지정 목록을 저장해야만 한다.

순차 파일의 편집

지정 목록은 직접적으로 기호 파일로 작성되지 않고 "순차 파일"로 저장되어진다(그림 참조). 이 파일에서 지정은 순서적으로 수정되고 확장될 수 있다. 이런 과정을 "편집"이라고 한다.

컴파일

편집한 후에, 사용자는 순차 파일(...ZO.INI)로부터 ZULI 파일(...ZO.INI)로 지정 목록을 전송하는 컴파일 과정을 동작시키고, 동시에 절대 연산수로 지정 목록을 분류 정렬(sort)한다.

보조 파일

컴파일 동안에 2개의 보조 파일(...Z1.INI와 ...Z2.INI)이 만들어진다:
 프로그램 작성기로 지정 목록에 접근하면, 파일 ...Z2.INI에서 지정 목록 내부의 위치의 정보를 이용할 수 있으므로 특정한 기호 연산수에 대한 절대 연산수를 찾을 수 있다.

파일 이름

모든 4개의 파일은 위치 7과 8을 제외하면 모두 같은 이름을 가진다. 프로그래머에 의해 각각의 이름에 붙여진 파일 형태의 정의(.XXX)는, 파일이 프로그램 파일인지 아니면 순차 파일인지 또는 기호 파일 중의 하나인지 등을 지정한다.

- S15A@@ST.S5D = 프로그램 파일
- S15A@@Z0.SEQ = 순차 파일
- S15A@@Z0.INI = 기호 파일(ZULI)
- S15A@@Z1.INI = 보조 기호 파일
- S15A@@Z2.INI = 보조 기호 파일

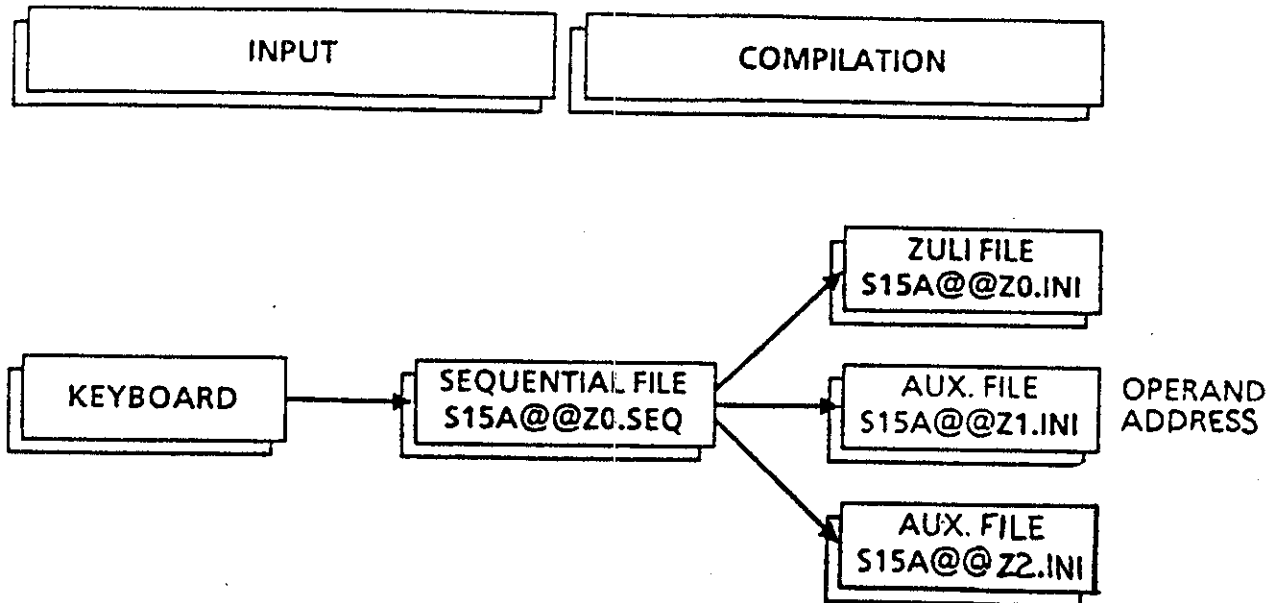


그림 12.3

일반적으로, 사용자 프로그램의 프로그램 파일과 기호 지정 파일은 동일한 이름, 예를 들면, S15A@@ST.S5D와 S15A@@Z0.INI가 붙는다. 그러나, 필요하다면 다른 이름을 붙여도 좋다.

순서 파일의 삭제

기호 파일이 일단 작성되면, 사용자는 순서 파일(sequential file)을 삭제하여도 된다. 만일 나중의 단계에서, 기호 파일을 수정 또는 확장할 필요가 생기면, 프로그램 작성기를 사용하여 기호 파일로부터 순서 파일을 새로 만든다.

주 의 ! 기호 파일의 생략시 양식(PRESETTING 형식)이 확인되었으면, 기호의 길이와 주석문의 길이에 대한 지정된 값은 변경할 수 없다.

세그먼트 헤더

CSF, LAD 또는 STL에서 쓰여진 각각의 세그먼트에 헤더를 붙일 수 있다. 이들 헤더는 하나의 블록(예. PB 51)에 지정된 주석문 블록(예. PC 51)에 저장되며, 하드 디스크 또는 플로피 디스크에만 저장된다.

세그먼트 헤더를 입력하려면,

- 사용자 프로그램을 포함하는 파일이 선택되어야만 한다.
- 미리 설정(생략시) 형식은 주석문 작성을 허용하여야 한다(YES);
- 각각의 세그먼트 헤더는 최대 32 문자 길이까지 작성 가능하다.

명령문 주석

STL 형식으로 쓰여진 각각의 명령은 최대 32 문자의 명령문 주석이 가능하다. 세그먼트 헤더와 같이, 세그먼트 주석은 주석문 블록에 저장된다.

입력/수정

명령문 주석 또는 세그먼트 헤더를 입력하고 수정하는 데에는 두가지의 방법이 있다 :

1. 블록(OBn, PBn, FBn)을 통하여 직접적으로 입력/수정하라.
상용하는 주석 블록(OCn, PCn, FCn)은 자동적으로 발생되며, 해당 블록은 저장될 때 덮어 쓰여질 수 있다.
2. 주석 블록에서 직접적으로 입력/수정하라.

이 중에서 첫번째 방법을 사용하길 권한다.

참 고

세그먼트 헤더를 수정할 때마다, 주석문 블록에 걸쳐 작성하여야 한다.

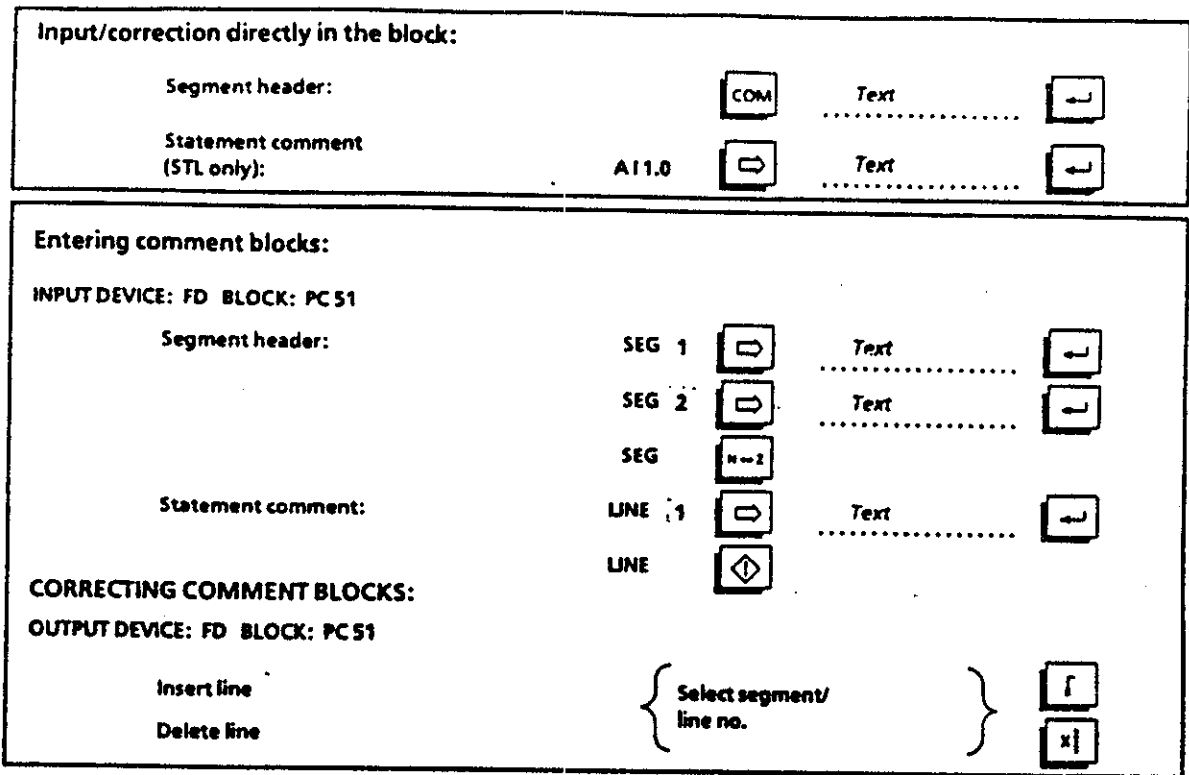


그림 12.4

Notes :

세그먼트 주석

각각의 세그먼트는 대략적으로 DIN A4 규격으로 4쪽(페이지) 분량의 세그먼트 주석으로 표현할 수 있으며, 문장은 문서화 블록에 저장된다. 이 문서 블록은 항상 상응하는 STEP 5 블록으로 지정된다.

STEP 5 block	Documentation block
OBn	#OBDO.n
PBn	#PBDO.n
SBn	#SBDO.n
FBn	#FBDO.n

**세그먼트 주석을
입력하고 수정하기**

여기서는 두가지의 방법이 사용된다 :

1. 입력/수정을 직접 해당 STEP 5 블록을 통하여 수행한다. 해당 문서화 블록은 STEP 5 블록이 저장될 때에 만들어 지거나 또는 겹쳐쓸 수 있다.
2. 적절한 문서 블록을 수정 또는 입력하라.

이 중에서 첫번째 방법이 권장된다.

필수 조건 :

- 사용자 프로그램을 포함하는 파일을 선택할 것.
- 주석은 미리 설정(생략시) 형식에서 허락(YES)되어야만 한다.

참 고

세그먼트 코드(예. \$! @ ↑)에 대한 제어 문자는 덮어 쓰여지지 말아야 하며, 그렇지 않으면 주석은 세그먼트에 지정될 수 없다. 본문의 입력은 문자 I로 시작된다.

PB52 -FIN. ASBL
 SEGMENT 1 0000 FINAL ASSEMBLY + DISPLAY LEN=17 SYM

.....
 * THIS SEGMENT CONTAINS THE FUNCTIONS OF THE FINAL ASSEMBLY. THE INDICATOR
 * LAMP "FIN. ASBL" IS TO FLASH WHEN THE CONVEYOR IS RUNNING AND SHOW
 * STEADY LIGHT WHEN WORK IS BEING CARRIED OUT AT THE FINAL ASSEMBLY POINT.

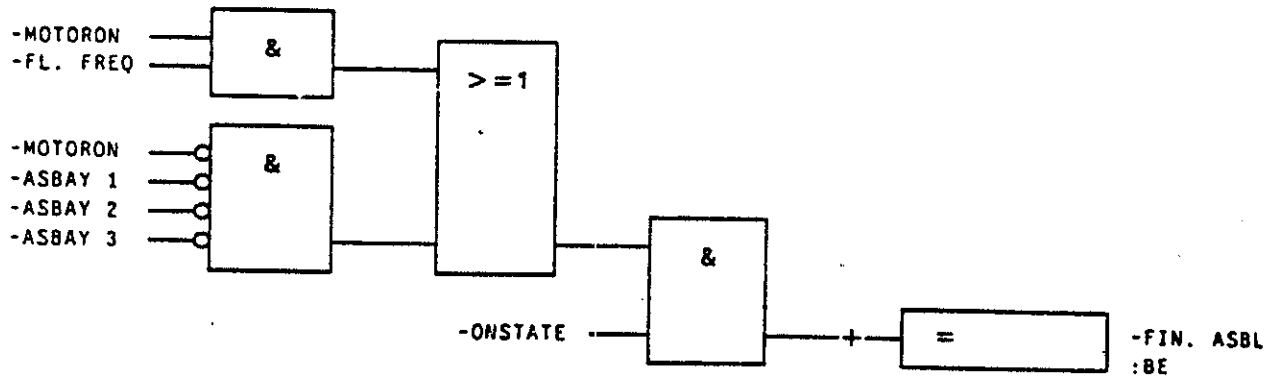


그림 12.5

Notes :

제어 대상 주석

제어 대상 주석문은 DIN A4 규격으로 대략 4쪽 분량까지 하드 디스크의 문서화 블록에 저장할 수 있다.

문서 파일과 소프트웨어 블록을 구별하기 위해서 문서 파일의 이름은 문자 #를 앞에 붙여야 한다. 예를 들면 :

```
INPUT DEVICE :   FD      BLOCK : # ASBLINE(조립 라인)
OUTPUT DEVICE :  FD      BLOCK : # ASBLINE(조립 라인)
```

문서 파일의 이름은 최대 9개의 문자(#을 포함하여)로 구성할 수 있다.

참 고

프로그램 설명서는 제어 대상 주석 또는 문서 파일의 입력, 출력(화면 표시), 그리고 수정에 대한 더 많은 정보를 제공한다.

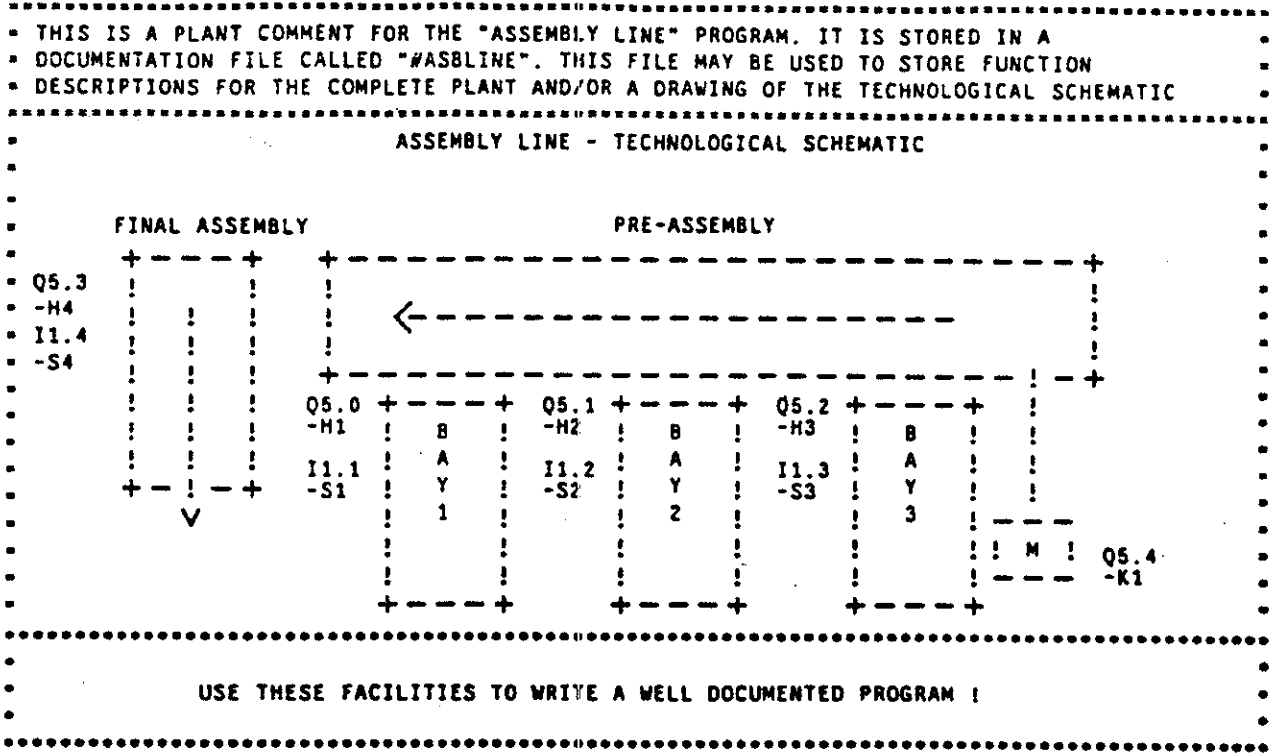


그림 12.6

Notes :

교차 참조 목록 기능
(Cross-reference list)


XRF, COMPARE, REW 패키지 교차 목록 기능은 여분의 유용한 문서 자원을 제공한다. 가장 중요한 요소는 프로그램 개요, 교차 참조 목록 그리고 I/Q/F 참조 목록이다.

호출

다양한 문서 기능은 XRF, COMPARE, REW 패키지에서 기능 키 F1 = XRF를 누르고 명령 라인을 입력하는 것에 의해 호출된다.

HELP 키



OUTPUT XRF DEVICE : PC BLOCK : A TERM 

참고

교차 참조 목록 기능의 요소는 특히, 프린터에 대한 출력에 적합하다.

DB에 대한
프로그램 개요

프로그램에 주석 달기는 사용자 프로그램에서 모든 블록의 목록과 더불어 시작한다. 블록의 길이는 각각의 블록을 보여주는데, 각각의 블록 형태의 총 길이와 전체 사용자 프로그램의 길이를 나타낸다. 실제 프로그램 개요는 블록의 중첩된 정도를 나타내며, OB 1으로 시작된다 ; 모든 블록은 OB 1에서 호출하는 순서로 가장 왼쪽 수직 선을 따라 목록이 나열된다. 두번째 수직선은 두번째 레벨등(예. FB 236은 PB 36에서 호출된다)에서 호출된 모든 블록을 나타낸다. 몇개의 블록(예. FB 232)은 특별한 식별 기호(identifier)로 표시된다. 예를 들어, 다양한 식별 기호의 의미는 그림에 표시되어 있다.

1st PROGRAM OVERVIEW DISPLAY WITH DB

```

PB 36:      LENGTH : 128
PB 41:      LENGTH : 65
PB 55:      LENGTH : 37
PB 57:      LENGTH : 44
SB 05:      LENGTH : 493
SB 55:      LENGTH : 137
FB 70S:     LENGTH : 1052
FB 203:     LENGTH : 14
etc.
LENGTH :      PB      274
LENGTH :      SB      630
LENGTH :      FB      1077
LENGTH :      FX      0
LENGTH :      OB      58
LENGTH :      DB      335
LENGTH :      DX      0
LENGTH :                               2374
    
```

```

--OB 1--PB 36--FB 236-
      I
      --PB 55--DB 100
      I
      --PB 57--FB70

--OB 13--FB 232-????????

--OB 21-

--OB 22-
    
```

Identifiers:

```

- JU (unconditional jump), C DB
= JC (conditional jump)
???? No such source block
    
```

그림 12.7

Notes :

X REFERENCE LIST: OUTPUTS

Q	4.0 -	SB 55 S OB 13	1* 1*
Q	4.3 -	PB 36	7, 8, 13*
Q	4.4 -	PB 36 PB 41 PB 57	6, 8, 9, 11, 12* 7* 4*
Q	5.4 -	PB 36 PB 41 PB 57	9, 11, 14* 6? 2?
Q	5.5 -	PB 36 PB 41 PB 57	9* 5, 6? 2?
QB	7 -	PB 57	3*
OW	6 -	PB 41	6?

Identifiers:

None Operand used in scan

• Operand used in assignment

? Operand used as parameter in an FB call

그림 12.8

교차 참조 목록

교차 참조 목록은 모든 연산수에 대하여 출력할 수 있다. 앞서 보여진 표본 목록은 프로그램에서 사용된 모든 출력을 나열한 것이다. 각각의 출력 다음에는 출력이 사용된 블록과 세그먼트가 있다. 교차 참조 목록은 연산수를 재지정하고 연산수를 수정하는데 도움을 준다 (예를 들면, 1에 대한 검색을 0에 대한 검색으로 변경하는 것).

I / Q / F P L A N

INPUTS IN PROGRAM

	17	6	5	4	3	2	1	0	I	B	W	D
BYTE 0	1	X	X	X	X	?	X	?	X	1	-	-
BYTE 2	1								1	-	X	-
BYTE 4	1	-	-	-	-	-	-	-	1	-	-	-
BYTE 6	1	-	-	-	-	-	-	-	1	-	-	-
BYTE 8	1	X	X	-	X	-	X	X	1	-	-	-
BYTE 10	1	-	-	-	-	-	-	-	1	-	-	-
BYTE 12	1	-	-	-	-	-	-	-	1	-	-	-
BYTE 14	1	-	-	-	-	-	-	-	1	-	-	-

	17	6	5	4	3	2	1	0	I	B	W	D
BYTE 1	1	7	-	-	-	-	-	-	X	1	-	-
BYTE 3	1								1	X	-	-
BYTE 5	1	-	-	-	-	-	-	-	1	-	-	-
BYTE 7	1	-	-	-	-	-	-	-	1	-	-	-
BYTE 9	1	-	-	-	X	X	X	1	-	-	-	-
BYTE 11	1	-	-	-	-	-	-	-	1	-	-	-
BYTE 13	1	-	-	-	-	-	-	-	1	-	-	-
BYTE 15	1	-	-	-	-	-	-	-	1	-	-	-

FLAGS IN PROGRAM

BYTE 40	1	-	-	X	X	X	X	X	1	-	-	-
BYTE 42	1								1	-	?	-
BYTE 44	1								1	-	X	-
BYTE 46	1								1	-	X	-
BYTE 48	1	-	-	-	-	-	-	-	1	-	-	-
BYTE 50	1	-	-	-	-	-	-	-	1	-	-	-
...												
BYTE 230	1	-	-	-	-	-	-	-	1	-	-	-
BYTE 232	1	-	-	-	-	-	-	-	1	-	-	-
BYTE 234	1	X	X	X	X	X	X	X	1	-	-	-
BYTE 236	1	-	-	-	-	-	-	-	1	-	-	-
BYTE 238	1	-	-	-	-	-	-	-	1	-	-	-
BYTE 240	1	-	-	-	-	-	-	-	1	-	-	-

BYTE 41	1	-	X	-	-	X	-	-	1	-	-	-
BYTE 43	1				X	X	1	-	-	-	-	-
BYTE 45	1							1	-	-	-	-
BYTE 47	1							1	-	-	-	-
BYTE 49	1	-	-	-	-	-	-	-	1	-	-	-
BYTE 51	1	-	-	-	-	-	-	-	1	-	-	-
...												
BYTE 231	1	-	-	-	-	-	-	-	1	-	-	-
BYTE 233	1	-	-	-	-	-	-	-	1	-	-	-
BYTE 235	1	-	-	-	-	X	1	-	-	-	-	-
BYTE 237	1	-	-	-	-	-	-	-	1	-	-	-
BYTE 239	1	-	-	-	-	-	-	-	1	-	-	-
BYTE 241	1	-	-	-	-	-	-	-	1	-	-	-

Identifiers:

- Operand not used
- X Operand used as bit
- ? Operand used as parameter in an FB call
- None Operand used as byte or word (not as bit)
- S Operand used in a standard function block
- I Operand used in a standard function block and in a user block

그림 12.8a

I/Q/F 편성표

I/Q/F 편성표(입력/출력/플래그 편성표)는 표의 형식을 가지고 있으며, 프로그램에 나타나는 그리고 아직 사용되지 않은(예, 프로그램을 확장할 때 사용함) 입력, 출력 및 플래그를 보여준다. I/Q/F 편성표에서는 또한 연산수가 바이트(B), 워드(W), 또는 이중 워드(D) 연산수 등으로 사용됨을 보여준다.

필요하다면, 이튼바 표제 블록(각주, footer)은 각 페이지의 프린트 출력 하단에 포함될 수 있다.

표제 블록 표제 블록은 80 또는 132개의 문자로 이루어질 수도 있다. 표제 블록 화면 형식에서 일부 영역은 생략시 설정 값을 포함하지만 겹쳐 쓸 수도 있다.

프린터 파일 프린터 파일은 프린터에 대한 매개 변수를 포함하며, 표준 프린터 (PT 88/89/90)가 선택될 때 자동적으로 발생된다.

호출 표제 블록 편집기는 F2 = UTILITY와 F2 = T.BLOCK에 의하여 "PACKAGE SELECTION" 메뉴에서 호출된다. PRESETTING 형식이 완료될 때, 4개의 필드로 이루어진 T.BLOCK은 모니터 화면에 표시된다. F1 = EDIT를 누르고, 커서 제어키로 필드를 선택하고, F1 = TEXT IN을 누름에 의해 표제 블록의 입력을 시작하라. F2 = TEXT END로 입력을 확인하라.

참고 새로운 필드에 대한 입력 내용은 F1 = TEXT IN으로 시작해서, F2 = TEXT END로 확인해야만 한다. 필드는 커서 제어 키를 통하여 선택된다.

Defaults

Title block 80 characters

Title block file: B:S15A@@F1.INI
 Printer file: B:S15A@@DR.INI
 Printer type: PT88

SIEMENS AG		SAMPLE DOCUMENTATION 80-CHARACTER TITLE BLOCK	
JOB IDENTIFIER:	S 15	DATE:	1.1.1990
SYSTEM:	EXERCISE	CHANGE:	*** BLNR
SYSTEM POS. NO.:	7.23	USER:	YOUR NAME

그림 12.9

Notes :