

데이터 형: **STRING**□ **STRING** (문자열) 형식의 변수:

- ○ 데이터 형 **STRING**은 254자까지의 문자열을 표시합니다.
 - 응용: 메시지 텍스트의 처리
 - 선언:
 - 문자열 이름: **STRING**[최대 숫자]: '초기 텍스트'
(최대 숫자 문자까지에 대한 문자열 변수 S, 최대 숫자: 0... 254)
 - 문자열 이름: **STRING**: '초기 텍스트'
(254자까지에 대한 문자열 변수)

□ 보기:

- 변수 선언:
 - 오류 신호: **STRING** 'Motor failure 4'
(변수 오류 신호는 위의 텍스트로 초기화됩니다.)
 - 경고: **STRING**[50] ''
(“비어 있음” 변수 경고는 문자 50자까지 받아들일 수 있습니다.)
- 처리:
 - 기본 액세스:
 - L Fault signal[5] (오류 신호에서 5번째 문자를 로드합니다.)
 - IEC-Library FC에 의해 처리

SIMATIC S7

Siemens AG 1998. All rights reserved.

Date: 2002-02-23
File: PRO2_05e.20Information and Training Center
Knowledge for Automation

개요

String 데이터 형은 문자열 (예: 메시지 텍스트)을 저장하기 위해 사용됩니다. 이런 방식으로 단순한 “(메시지) 워드 프로세싱 시스템”이 **S7-CPU**에서 실행될 수 있습니다. **STRING** 데이터 형은 254자까지의 문자열을 나타냅니다.

선언에서 꺾쇠괄호로 지정된 숫자(1..254)는 **STRING** 변수에 저장될 수 있는 최대 문자 수를 나타냅니다. 만약 이 정보가 제공되지 않으면 **STL/LAD/FBD** 편집기는 254자의 길이를 가정합니다.

STRING 변수에 대한 액세스

STRING 변수의 개별 문자는 다음과 같은 기본 **STL** 인스트럭션을 이용해서 액세스될 수 있습니다.

- L StringName[5](변수에 저장된 5번째 문자를 로드합니다.)

STRING 변수 (메시지 텍스트)의 실제 처리는 **IEC-Library FC**를 통해 가능합니다.

초기화

선언에서 **STRING** 데이터 형의 변수에 시작 텍스트를 미리 할당할 수 있습니다 (**FC**로의 블록 파라미터나 **FB**의 입/출력 파라미터, 또는 임시 변수로서는 안됨).

초기화는 작은 따옴표로 묶은 **ASCII** 코드 문자로 이루어집니다. 만약 용어를 통제하기 위한 특수 문자가 포함된다면 달러 문자 (\$)를 앞에 놓아야 합니다.

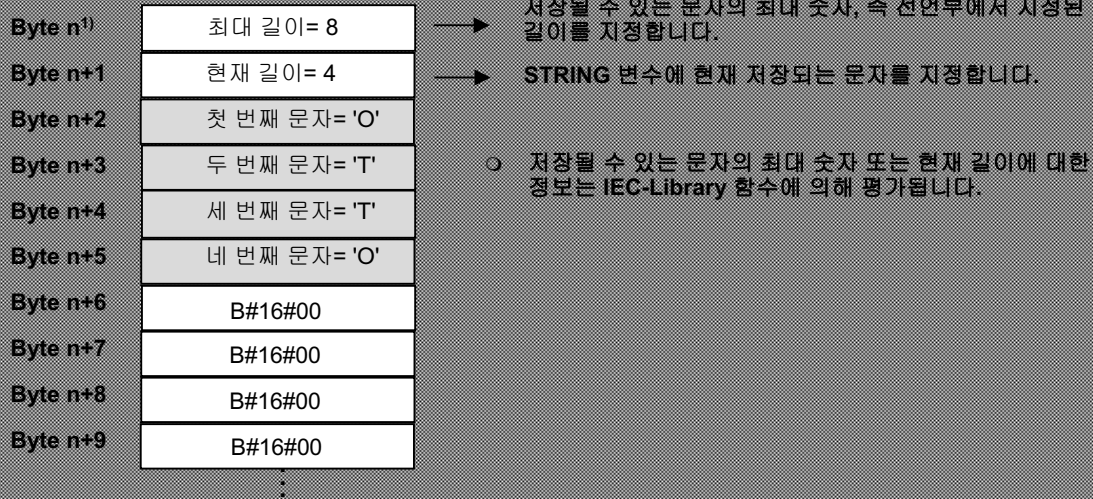
사용할 수 없는 특수 문자는 다음과 같습니다.

- \$\$ 단순한 달러 문자
- \$L, \$l 줄 바꿈 (LF)
- \$P, \$p 페이지 바꿈
- \$R, \$r 캐리지 리턴
- \$T, \$t 탭 키

메모리에 **STRING** 변수 저장

□ 보기:

- 초기화로 선언
 - 주어진 이름: `STRING[8]: 'OTTO'`
- **STRING** 변수 "주어진 이름" 저장



¹⁾ n = 짝수

SIMATIC S7

Siemens AG 1998. All rights reserved.

Date: 2002-02-23
File: PRO2_05e.21



Information and Training Center
Knowledge for Automation

개요

STRING 데이터 형의 변수는 최대 길이가 256 바이트이며 따라서 254자의 "순수 데이터" 즉 문자가 수용될 수 있습니다.

변수 저장

STRING 변수는 항상 워드 한계, 즉 짝수 어드레스를 가지는 바이트에서 시작합니다.

변수를 설정할 때 최대 길이가 변수 선언에 따라 변수의 첫 번째 바이트에 입력됩니다. 마찬가지로 사전 할당이나 처리에서 현재 사용되는 길이, 즉 저장된 문자열의 길이가 **IEC-Library** 함수를 이용해 두 번째 바이트에 입력됩니다. 이러한 정보는 모두 **STRING** 변수를 처리할 때 **IEC-Library**에 의해 요구됩니다.

그 후에 **ASCII**-포맷의 문자가 따라 옵니다. **STRING** 변수에 포함되지 않은 문자는 초기화할 때 **B#16#00**로 채워집니다.

파라미터 전달

STRING 데이터 형의 변수는 **ARRAY**나 **STRUCT** 변수와 마찬가지로 데이터 형이 같은, 즉 **STRING** 길이가 같은 블록 파라미터에 전달될 수 있습니다.

POINTER 또는 **ANY** 데이터 형의 **FC** 또는 **FB** 파라미터로의 전달도 가능합니다.

STRING 변수를 처리하기 위한 함수

□ StdLib30에서의 IEC-Library

- ○ **FC2 (CONCAT):** 함수 FC2는 2개의 STRING 변수를 한 개의 문자열로 결합합니다.
- **FC4 (DELETE):** 함수 FC 4는 문자열에서 P. 문자로서의 L 문자를 삭제합니다.
- **FC11 (FINF):** 함수 FC 11은 첫 번째 문자열에서 두 번째 문자열의 위치를 전달합니다.
- **FC17 (INSERT):** 함수 FC 17은 P. 문자 이후에 파라미터 IN2의 문자열을 파라미터 IN1의 문자열에 삽입합니다.
- **FC20 (LEFT):** 함수 FC 20은 문자열의 첫 번째 L 문자를 전달합니다.
- **FC21 (LEN):** 함수 FC 21은 문자열의 현재 길이 (유효한 문자 수)를 출력합니다.
- **FC26 (MID):** 함수 FC 26은 문자열의 중간 부분을 전달합니다.
- **FC31 (REPLACE):** 함수 FC 31은 P. 문자로서 (포함) 첫 번째 문자열 (IN1)의 L 문자를 두 번째 문자열 (IN2)로 대체합니다.
- **FC32 (RIGHT):** 함수 FC 32는 문자열의 마지막 L 문자를 전달합니다.
- **STRING 변수에 대한 비교 함수:** FC10 (EQ_STRING), FC13 (GE_STRING), FC15 (GT_STRING), FC19 (LE_STRING), FC24 (LT_STRING), FC29 (NE_STRING)

SIMATIC S7

Siemens AG 1998. All rights reserved.

Date: 2002-02-23
File: PRO2_05e.22



Information and Training Center
Knowledge for Automation

개요

STEP7을 설치하면 하위 라이브러리 IEC를 가지는 라이브러리 StdLib30도 설치되며 이 라이브러리는 IEC 데이터 형의 처리를 위한 함수를 포함합니다. 위의 슬라이드는 STRING 형의 변수를 처리하기 위한 함수의 목록을 보여줍니다.

참고

일반적으로 함수는 사용된 최대 길이 또는 실제 길이에 대한 정보를 이용해 오류 평가를 실행합니다. 함수가 오류를 인식하면 일반적으로 BR 비트는 "0"으로 설정됩니다.

개별 함수에 대한 자세한 설명은 IEC-Library에 대한 온라인 도움말에 나옵니다.

비교 함수

비교 함수는 문자열의 사전 편집상의 비교를 실행합니다. 왼쪽에서 시작해서 문자는 ASCII 코딩과 비교됩니다 (예: 'a'는 'A'보다 크며 'A'는 'B'보다 더 작습니다).

첫 번째 다른 문자가 비교 결과를 결정합니다. 만약 보다 긴 문자열의 왼쪽 부분이 보다 짧은 문자열과 같다면 보다 긴 문자열이 더 긴 것으로 간주됩니다.

함수는 오류를 표시하지 않습니다. 각 비교 함수는 반환 값 RET_VAL을 이용해 비교가 완료되었는지 (RET_VAL=TRUE) 아닌지 (RET_VAL=FALSE)를 나타냅니다.

사용자 정의 데이터 형: UDT

□ 사용자 정의 데이터 형에 대한 응용:

- ○ 속성이 확장되었거나 변경된 새 데이터 형 만들기 (예: 초기값)
- 자동화 작업에 대한 보다 효율적인 해결 방안
- 프로그램 컨테이너의 모든 블록에 대해 보편적으로 유효
- 변수 정의에서 원하는 만큼 자주 사용되거나 전역 (공유) DB를 만들기 위해 사용될 수 있음

□ 보기:

○ 새 데이터 형의 정의 (구조체):

```
UDT5          STRUCT
ON            : BOOL;
OFF           : BOOL;
SetpointSpeed: INT;
ActualSpeed  : INT;
END_STRUCT;
...
```

○ 변수 선언:

```
Motor_1:      UDT5;
Motor_2:      UDT5;
```

○ 변수에 대한 액세스:

```
L Motor_1.ActualSpeed
```

SIMATIC S7

Siemens AG 1998. All rights reserved.

Date: 2002-02-23
File: PRO2_05e.23



Information and Training Center
Knowledge for Automation

개요

데이터 구조가 사용자 프로그램에서 빈번하게 반복되거나 데이터 구조에 이름이 주어지면 **STEP7**은 자체 사용자 정의 데이터 형 (**UDT= 사용자 정의 데이터 형**)이 정의될 수 있게 허용합니다 (예: 고급 언어 “C”에서의 *typedef*).

응용 프로그램과 관련된 데이터 형의 사용을 통해 해결해야 할 작업이 보다 효율적으로 프로그램될 수 있습니다. 제조업체와 같은 사용자는 자신의 문제에 맞게 조정된 데이터 형을 작성해서 설치할 수 있습니다.

UDT 만들기

UDT는 점증식 편집기나 텍스트 편집기로 만들어져 블록 컨테이너에 “독립적” 블록 (**UDT1 ... UDT65535**)으로 저장됩니다.

기호 이름이 이 **UDT**에 할당되거나 전역 기호 표의 이름을 통해 관련 데이터 구조에 할당될 수 있습니다.

전역적으로 유효한 “템플릿”이 **UDT**를 통해 만들어지며, 이 템플릿은 새 변수의 선언에서 또는 공유 **DB**를 만들기 위해 원하는 만큼 자주 사용될 수 있습니다.