

# SCE 교육 커리큘럼

Siemens Automation Cooperates with Education | 05/2017

## TIA Portal Module 032-500 아날로그 값



교육 시설 및 R&D 기관에서의 사용에는 제한이 없습니다. ⓒ Siemens AG 2017. All rights reserved.

교육 커리큘럼에 따른 적합한 SCE 트레이너 패키지

#### SIMATIC 컨트롤러

- SIMATIC ET 200SP Open Controller CPU 1515SP PC F 및 HMI RT SW 주문 번호: 6ES7677-2FA41-4AB1
- SIMATIC ET 200SP Distributed Controller CPU 1512SP F-1 PN Safety 주문 번호: 6ES7512-1SK00-4AB2
- SIMATIC CPU 1516F PN/DP Safety 주문 번호: 6ES7516-3FN00-4AB2
- SIMATIC S7 CPU 1516-3 PN/DP 주문 번호: 6ES7516-3AN00-4AB3
- SIMATIC CPU 1512C PN(소프트웨어 장착) 및 PM 1507 주문 번호: 6ES7512-1CK00-4AB1
- SIMATIC CPU 1512C PN(소프트웨어 장착), PM 1507 및 CP 1542-5 (PROFIBUS) 주문 번호: 6ES7512-1CK00-4AB2
- SIMATIC CPU 1512C PN(소프트웨어 장착) 주문 번호: 6ES7512-1CK00-4AB6
- SIMATIC CPU 1512C PN(소프트웨어 장착) 및 CP 1542-5 (PROFIBUS) 주문 번호: 6ES7512-1CK00-4AB7

#### 교육용 SIMATIC STEP 7 소프트웨어

- SIMATIC STEP 7 Professional V14 SP1 단일 라이센스 주문 번호: 6ES7822-1AA04-4YA5
- SIMATIC STEP 7 Professional V14 SP1 강의실 라이센스 (최대 인원 6명) 주문 번호: 6ES7822-1BA04-4YA5
- SIMATIC STEP 7 Professional V14 SP1 업그레이드 라이센스 (최대 인원 6명) 주문 번호: 6ES7822-1AA04-4YE5
- SIMATIC STEP 7 Professional V14 SP1 학생 라이센스 (최대 인원 20명) 주문 번호: 6ES7822-1AC04-4YA5

위 트레이너 패키지는 필요 시 후속 모델 패키지로 대체가 된다는 점에 유의하십시오. 현재 출시된 SCE 패키지에 대한 개요는 <u>siemens.com/sce/tp</u>에서 제공됩니다.

#### 보충 교육

지멘스의 지역별 SCE 보충 교육에 대한 내용은 해당 지역의 SCE 고객 센터로 문의하시기 바랍니다. siemens.com/sce/contact

#### SCE 관련 추가 정보

siemens.com/sce

#### 사용 관련 정보

통합 자동화 솔루션인 TIA(Totally Integrated Automation)를 위한 SCE 교육 커리큘럼은 공교육 시설 및 R&D 기관 교육 목적의 "SCE(Siemens Automation Cooperates with Education) 프로그램을 위해 마련된 것입니다. Siemens AG는 프로그램의 내용을 보증하지 않습니다.

본 문서는 지멘스 제품/시스템을 초기 교육하는 용도로만 사용되어야 합니다. 따라서 교육 범위 내에서의 사용 목적으로 전체 또는 일부를 복사하여 교육생들에게 제공할 수 있습니다. 본 문서는 공공 교육 및 고등 교육 시설 내에서의 교육을 위한 목적으로의 배포, 복사 및 내용의 공유가 가능합니다.

예외적인 경우에는 Siemens AG 담당자의 서면 동의가 필요합니다. Roland Scheuerer roland.scheuerer@siemens.com.

해당 규정의 위반 시에는 그에 대한 책임이 부과될 수 있습니다. 특히 특허가 부여되었거나 실용신안 또는 의장등록이 된 경우, 번역을 포함한 제반 권리는 지멘스의 소유입니다.

산업체 고객을 위한 교육 과정의 사용은 명시적으로 금지됩니다. 지멘스는 교육 커리큘럼의 상업적 이용을 거부합니다.

드레스덴공대(TU Dresden), 특히 공학 박사 Leon Urbas 교수와 Michael Dziallas Engineering Corporation, 그리고 본 교육 커리큘럼을 준비하는 과정에서 도움을 주신 모든 관계자들께 감사의 말씀을 전합니다.

### 목차

TOC

### SIMATIC S7-1500을 위한 아날로그 값

### 1 목표

이 챕터에서는 TIA Portal 프로그래밍 툴을 통해 SIMATIC S7-1500의 아날로그 값을 처리하는 방법에 대해 배워보겠습니다.

여기에는 아날로그 신호를 수집해서 처리하는 방법이 나와 있으며, SIMATIC S7-1500의 아날로그 값에 대한 읽기/쓰기 액세스가 단계별로 설명되어 있습니다.

제3장에 기술된 SIMATIC S7 제어 장치를 사용할 수 있습니다.

### 2 전제 조건

SIMATIC S7 CPU1516F-3 PN/DP에서 챕터 "IEC 타이머 및 카운터"에서 배운 내용을 토대로 합니다. 이 챕터에서는 예를 들어 032-300 IEC Timers and Counters.zap13 같은 프로젝트를 사용할 수 있습니다.

### 3 필요한 하드웨어 및 소프트웨어

- 엔지니어링 스테이션: 하드웨어 및 운영 시스템이 필요합니다(자세한 정보는 TIA 포털의 설 치 DVD Readme/Liesmich를 참조하세요).
- 2 TIA 포털의 소프트웨어 SIMATIC STEP 7 Professional V13부터
- 3 SIMATIC S7-1500/S7-1200/S7-300 제어 장치, 예: CPU 1516F-3 PN/DP 펌웨어 버전 V1.6 이상, 메모리 카드와 16DI/16DO 및 2AI/1AO 포함 참고: 디지털 입력과 아날로그 입력 및 출력은 컨트롤 패널에서 실행해야 합니다.
- 4 엔지니어링 스테이션과 제어 장치 간 이더넷 연결



### 4 이론

### 4.1 아날로그 신호.

2개의 신호 상태 ("전압 공급 +24 V" 및 "전압 미공급 0 V")만 가질 수 있는 바이너리 신호와 달리, 아날로그 신호는 정의된 범위 내에서 어떤 값이든 가질 수 있습니다. 대표적인 아날로그 센서로 포텐셔메타가 있습니다. 노브의 위치에 따라 저항을 최대값까지 자유롭게 설정할 수 있습니다.

제어 엔지니어링에서 아날로그 범위의 예:

- 온도: -50 ~ +150 °C
- 유속: 0 ~ 200 l/min
- 속도: -500 ~ +50 rpm
- 기타

### 4.2 측정 변환기(Measuring trasducer)

측정 변환기를 통해 이러한 아날로그 양을 전기 전압, 전류 또는 저항으로 변환할 수 있습니다. 예를 들어 속도를 측정할 경우, 측정 변환기를 이용해 500 ~ 1500 rpm의 속도 범위를 0 ~ +10 V의 전압 범위로 변환할 수 있습니다. 측정 속도가 865 rpm일 때 측정 변환기는 +3.65 V의 전압 값을 출력하게 됩니다.

500	86	5	1500	rpm
	365			10 V: 1000 rpm = 0.01 V/rpm
		1000 rpm		365 rpm x 0.01 V/rpm = 3.65 V
		10V		
0 V			+1	0 V

4.3 아날로그 모듈 - A/D 컨버터

이러한 전기적 전압, 전류 또는 저항은 아날로그 모듈에 연결이 되고, 아날로그 모듈은 PLC에서 추가 처리를 위해 이 신호를 디지털화합니다.

PLC에서 아날로그 값이 처리되는 경우에는 읽어들인 전압, 전류 또는 저항 값을 디지털 정보로 변환해야 합니다. 아날로그 값은 비트 패턴으로 변환이 됩니다. 이러한 변환을 아날로그-디지털 변환(A/D 변환)이라고 합니다. 예를 들어 전압 값 3.65 V은 연속적인 이진 숫자로 저장이 됩니다.

SIMATIC 제품에서는 이러한 변환의 결과가 항상 16비트 워드로 표시됩니다. 아날로그 입력 모듈의 통합 ADC(아날로그-디지털 컨버터)는 획득한 아날로그 신호를 디지털화해서 그 값을 스텝 커브의 형태로 근사치화 시킵니다. ADC에서 가장 중요한 파라미터는 해상도와 변환율입니다.



1: 아날로그 값

2. 디지털 값

디지털 표현에 사용하는 이진 숫자가 많을수록 해상도가 높아집니다. 예를 들어 0 ~ +10 V의 전압 범위에서 1비트만 사용할 수 있는 경우에는 측정된 전압이 0 ~ +5 V 또는 +5 V ~ +10 V에 있는지 여부만 알 수 있습니다. 그러나 2비트에서는 범위가 4개(0 ~ 2.5 / 2.5 ~ 5 / 5 ~ 7.5 / 7.5 ~ 10 V)로 나뉠 수 있습니다. 제어 엔지니어링의 기존 A/D 컨버터는 변환에 8비트나 11비트를 사용합니다.

8비트의 경우 256개의 범위를, 11비트의 경우 2048개의 범위를 제공합니다.

0A/0V

20mA/10V

		10 V: 2048 = 0,0048828
	11비트	→ 5mV 미만의 선압 차를 감지할 수
0	2048	

0

2048

### 4.4 SIMATIC S7-1500의 데이터 타입

SIMATIC S7-1500은 다양한 숫자 형식을 표현할 수 있도록 다양한 유형의 데이터를 지원합니다. 몇 가지 기본적인 데이터 타입의 목록이 아래 나와 있습니다.

데이터	크기 (비트)	범위	상수 입력의 예
타입			
Bool	1	0~1	TRUE, FALSE, O, 1
Byte	8	16#00 ~ 16#FF	16#12, 16#AB
Word	16	16#0000 ~ 16#FFFF	16#ABCD, 16#0001
DWord	32	16#00000000 ~ 16#FFFFFFFF	16#02468ACE
Char	8	16#00 ~ 16#FF	'A', 'r', '@'
Sint	8	-128~127	123,-123
Int	16	-32,768 ~ 32,767	123, -123
Dint	32	-2,147,483,648 ~ 2,147,483,647	123, -123
USInt	8	0 ~ 255	123
UInt	16	0 ~ 65,535	123
UDInt	32	0 ~ 4,294,967,295	123
Real	32	+/-1.18 x 10 -38 ~ +/-3.40 x 10 38	123.456, -3.4, -1.2E+12, 3.4E-3
LReal	64	+/-2.23 x 10 $^{-308}$ ~ +/-1.79 x 10 $^{308}$	12345.123456789-1.2E+40
Time	32	T#-24d_20h_31 m_23s_648ms ~ T#24d_20h_31 m_23s_647ms 다음과 같이 저장됨: -2,147,483,648 ms ~ +2,147,483,647 ms	T#5m_30s 5#-2d T#1d_2h_15m_30x_45ms
String	가변적	0 ~ 254개 문자 (바이트 크기)	'ABC'

참고: 'INT'와 'REAL' 데이터 타입은 아날로그 값 처리에서 중요한 역할을 합니다. 이는 읽어들인 아날로그 값이 'INT' 형식의 16비트 정수로서 존재하기 때문이며, 'INT'의 경우 반올림 오차가 발생할 수 있기 때문에 정확한 추가 처리를 위해서는 'REAL' 형식의 부동 소수점 수만 사용해야 합니다.

### 4.5 아날로그 값 읽기/출력

PLC는 워드 형식으로 아날로그 값을 읽어오고 또한 출력합니다. 이러한 워드는 예를 들어 다음과 같은 오퍼랜드를 통해 액세스됩니다.

%IW 64 아날로그 입력 워드 64

%QW 64 아날로그 출력 워드 64

각 아날로그 값 ("Channel")은 1개의 입력/출력 워드를 차지합니다. 형식은 정수인 'INT'입니다. 입력 및 출력 워드의 어드레스는 Device overview에서 설정합니다. 예를 들면 아래와 같습니다.



첫 번째 아날로그 입력의 주소는 %IW 64, 두 번째 아날로그 입력의 주소는 %IW 66, 세 번째 아날로그 입력의 주소는 %IW68, 네 번째 아날로그 입력의 주소는 %IW70, 다섯 번째 아날로그 입력의 주소는 %IW72, 여섯 번째 아날로그 입력의 주소는 %IW74, 일곱 번째 아날로그 입력의 주소는 %IW 76, 여덟 번째 아날로그 입력의 주소는 %IW78이 됩니다.

또한, 첫 번째 아날로그 출력의 주소는 %QW64, 두 번째 아날로그 출력의 주소는 %QW66, 세 번째 아날로그 출력의 주소는 %QW 68, 네 번째 아날로그 출력의 주소는 %QW70이 됩니다. PLC에서 제어를 위한 아날로그 값 변환 방식은 아날로그 입력과 아날로그 출력 모두 동일한 방식을 사용합니다.

디지털화된 값 범위는 다음과 같습니다.



이렇게 디지털화된 값들은 PLC에서 적절한 방식으로 추가 처리를 해서 정규화시켜야 하는 경우가 종종 있습니다.

#### 4.6 아날로그 값 정규화

아날로그 입력 값이 +/- 27648 범위에서 디지털화된 값으로서 존재할 경우에는 숫자 값이 프로세스의 물리량에 해당되도록 정규화를 시켜야 하는 것이 일반적입니다.

마찬가지로, 아날로그 출력은 정규화된 값을 설정한 결과이기 때문에 출력 값 +/- 27648으로 스케일링을 해야 합니다.

TIA Portal에서는 정규화 및 스케일링을 위해 이미 생성된 블록 또는 산술 명령어가 사용됩니다.

이를 가능한 정확하게 수행하기 위해서는 정규화를 위한 값들을 REAL 데이터 타입으로 변환해서 반올림 오차를 최소화해야 합니다.

### 5 과제

이 챕터에서는 컨베이어 속도의 아날로그 제어 평션을 챕터 "SCE\_EN\_032-300 IEC 타이머 및 카운터"에서 생성된 프로그램에 추가해 보겠습니다.

### 6 계획 수립

컨베이어 속도의 아날로그 제어는 "SCE\_EN\_032-300 IEC 타이머 및 카운터" 프로젝트의 연장선으로 "MOTOR\_SPEEDCONTROL" [FC10] 평션에서 프로그래밍됩니다. 이 평션을 추가하기 위해서는 이 프로젝트의 압축을 풀어야 합니다. "MOTOR\_ SPEEDCONTROL" [FC10] 평션은 "Main" [OB1]" 오거나이제이션 블록에서 호출됩니다. 컨베이어 모터의 제어를 - Q3 (컨베이어 모터 -M1 가변 속도)로 변경을 해야 합니다.

#### 6.1 컨베이어 속도의 아날로그 제어

분당 회전수 (주황색: +/- 50 rpm)로 "MOTOR\_SPEEDCONTROL" [FC10] 펑션의 입력에서 속도가 설정됩니다. 데이터 타입은 32비트 부동 소수점 수 (REAL)입니다.

먼저, +/- 50 rpm의 범위에서 속도 설정값(setpoint)이 올바르게 입력되었는지 평션에 대한 확인이 이루어집니다.

만약 속도 설정값이 +/- 50 rpm 범위 밖에 있는 경우에는 데이터 타입이 16비트 정수(INT)인 값 0이 출력됩니다. 평션의 반환 값 (Ret\_Val)에 TRUE (1) 값이 할당됩니다.

속도 설정값이 +/- 50 rpm 범위 내에 있는 경우에는 이 값이 먼저 범위 0...1로 정규화가 된 다음, 아날로그 출력의 조작 값(manipulated value)으로서 출력이 되도록 데이터 타입이 16비트 정수(INT)인 +/- 27648으로 스케일링이 됩니다.

그리고 출력이 신호 U1 (2방향 모터의 조작 속도 값 +/- 10V는 +/- 50 rpm에 해당)에 연결됩니다.

### 6.2 기술 다이어그램

여기에는 과제를 위한 기술 다이어그램이 나와 있습니다.



#### 그림 1: 기술 다이어그램

Schalter der Sortieranlage	Automatikbetrieb	Handbetrieb / Manual mode
Switches of sorting station	Automatic mode	-S3 Tippbetrieb -M1 vorwärts/
-P1 ein/on	-P5 gestartet/started	<ul> <li>Manual -M1 forwards</li> </ul>
-Q0 Hauptschalter/Main switch	-S1 Start/start	-S4 Tippbetrieb -M1 rückwärts/
-P4 aktiviert/active		P7 ausoefahren/extended
-A1 NOTHALT/Emergency stop	-S2 Stopp/stop	-S6 Zylinder -M4 ausfahren/
-P2 Hand/manual -P3 Auto/auto		cylinder -M4 extend -P6 eingefahren/retracted
-S0 Betriebsart/operating mode		-S5 Zylinder -M4 einfahren/
		cylinder -m4 retract

그림 2: 제어 패널

### 6.3 참조 목록

이 과제를 위한 글로벌 오퍼랜드로서 아래와 같은 신호들이 필요합니다.

DI	유형	식별자	기능	NC/NO
1 0.0	BOOL	-A1	반환 신호 비상 정지 OK	NC
I 0.1	BOOL	-K0	메인 스위치 "온"	NO
1 0.2	BOOL	-S0	모드 선택 수동 (0) / 자동 (1)	수동 = 0
				자동 = 1
I 0.3	BOOL	-S1	푸시버튼 자동 시작	NO
I 0.4	BOOL	-S2	푸시버튼 자동 정지	NC
I 0.5	BOOL	-B1	센서 실린더 -M4 복귀	NO
I 1.0	BOOL	-B4	슬라이드의 센서	NO
1.3	BOOL	-B7	컨베이어 끝 부분 센서	NO

DO	유형	식별자	기능	
Q 0.2	BOOL	-Q3	컨베이어 모터 -M1 가변 속도	
QW 64	BOOL	-U1	2방향 모터의 속도 값 +/- 10V는 +/- 50 rpm에 해당	

참조 목록 범례

AI

L

- DI Digital Input DO Digital Output
  - AO Analog Output
    - Q Output
- NC Normally Closed

Input

Analog Input

NO Normally Open

### 7 단계별 따라 해보기

아래에는 계획 수립 방법에 대한 지침이 나와 있습니다. 모든 내용을 이미 충분히 숙지했다면 숫자가 표시된 단계로 넘어가도 좋습니다. 그렇지 않다면, 아래에 나와 있는 지침의 단계를 따라가면 됩니다.

### 7.1 기존 프로젝트 압축풀기

→ 챕터 "SCE\_EN\_032-300\_IEC\_Timers\_Counters"에서 생성된 "SCE\_EN\_032-300\_IEC\_Timers\_Counters.zap13 프로젝트를 확장할 수 있으려면 먼저 아카이브에서 이 프로젝트의 압축을 풀어야 합니다. 아카이브된 기존 프로젝트의 압축을 풀려면 "Project"의 "Retrieve"로 가서 해당되는 아카이브를 선택해야 합니다. "Open"을 클릭해 선택합니다.



(→ Project → Retrieve → .zap 아카이브 선택 → Open)

→ 그 다음으로 이 프로젝트가 저장될 대상 디렉토리를 선택합니다. "OK"를 눌러 선택합니다.

 $(\rightarrow$  Target directory  $\rightarrow$  OK)

→ 열린 프로젝트를 032-500\_Analog\_Values라는 이름으로 저장을 합니다.

 $(\rightarrow$  Project  $\rightarrow$  Save as...  $\rightarrow$  032-500\_Analog\_Values  $\rightarrow$  Save)



### 7.2 "MOTOR\_SPEEDCONTROL" 펑션 생성

- → CPU 1516F-3 PN/DP의 'Program blocks' 폴더를 선택한 다음, "Add new block"을 클릭해 새로운 블록을 생성합니다.
  - $(\rightarrow CPU_1516F [CPU 1516F-3 PN/DP] \rightarrow Add new block)$



→ 다음 대화상자에서 📲 를 선택하고 새 블록의 이름을 "MOTOR\_SPEEDCONTROL"으로 변경합니다. 언어를 FBD로 설정하고 번호 "10"을 수동으로 지정합니다. "Add new and open" 체크박스를 선택하고 'OK'를 클릭합니다.

 $(\rightarrow$   $\Rightarrow$   $\rightarrow$ 

Name: MOTOR_SPEEDCON	ROL				
-	Language:	FBD			
OB	Number:	10	٢		
Organization block		<ul> <li>Manual</li> <li>Automatic</li> </ul>			
Function block	Description: Functions are c	ode blocks or subrou	tines[without de	dicated memory.	
<b>:</b>					
Function					
БВ					
Data block	More				
Additional info	rmation				

→ 여기 나와 있듯이 코멘트가 있는 로컬 태그를 생성하고 'Return' 태그의 데이터 타입을 'Void'에서 'Bool'로 변경합니다.

(→ Bool)

]	Val	ues	▶ CPU1516F [CPU 1516F-3 PN/DF	P] 🕨 Program block	s ▶ MOTOR_S	SPEEDCONTROL [FC10]	_∎≡×
ő	ы	K 3	) 🔮 🍬 🖹 🖻 🖻 🗩 🕾 😂	± 🖃 😰 🎨 🌜	i 🕫 🤣 🕍	¥≡ 0° 📽 🔢	
	M	ото	R_SPEEDCONTROL				
		Na	ne	Data type	Default value	Comment	
	1	-	Input		1		
	-		Setpoint_speed	Real	1		
5		-	Output	1			
	1	-	Manipulated_variable_speed_AO	Int			
2			and a sup				
D 7	-	2	<add new=""></add>				
			Setuciet speed OK	Real			
2			Setpoint_speed_OK	Bool			
1.			Manipulated_variable_speed_Norm	Real			
10			Add asus				
17	-	-	<aud new=""></aud>				
12	1			Beel			
2	-	-	MOTOR_SPEEDCONTROL	6001			
_	5						
8		>=1	1777 → → → → → → →				
•	Blo	ck	title: Speed control via analog output				^
C	om	me	nt				-
		1.4					
	-	Net	WOIK 1				
	C	lom	ment				
							~
					100%		-3

참고: 올바른 데이터 타입을 사용해야 합니다.

→ 첫 번째 네트워크에 "Assignment" - [=]을 삽입하고 그 앞에 "AND" 🔹 를 삽입합니다.

그런 다음, 끌어다 놓기 기능을 통해 "Basic instructions"에서 비교 연산 "<="을 🧾 AND 논리 연산의 첫 번째 입력으로 이동시킵니다.

 $(\rightarrow$  **[=]**  $\rightarrow$  **Basic instructions**  $\rightarrow$  Comparator operations  $\rightarrow$  CMP<=)

'U1516F [CPU 1516F-3 PN/DP] > Program	n blocks ▶ MO	TOR_SPEEDCONTRO	DL [FC10] 🗕	∎≡×	Instructions 🗐 🗉 🕨
					Options
a 🖉 👻 💺 🔚 🚍 💬 🕮 🛎 🖉	± 🖃 🕼 🥐	⊊ 🖑 🕅 🍄 I≘	¥ & 📽 🔢		mi twi 🥅 🛄
MOTOR_SPEEDCONTROL					> Favorites
Name	Data type	Default value	Comment		✓ Basic instructions
1 🕣 💌 Input				^	Name Versi
2 🔄 🗉 Setpoint_speed	Real			=	🕨 🦳 General
3 🕣 🔻 Output					Bit logic operations
4 📶 🏽 Manipulated_variable_speed_AO	Int				Timer operations
5 🤕 🔻 InOut				~	▶ 🛐 Counter operations
•				2	Comparator operations
a >=1 1??? → -ol → -f=1					E CMP ==
					E CMP ⇔
<ul> <li>Block title: Speed control via analog output</li> </ul>				^	E CMP >=
Comment					E CMP <=
					E CMP >
Network 1: check setpoint speed for correction	ect input range +i-	50 r/min			E CMP <
Comment					IN_Range
					OUT_Range
8	?.?			-	
<7.2>	=			1000	-INOT_OK -
<17.2> - 12					🕨 🛅 Variant
					Math functions

- → 그런 다음, 끌어다 놓기 기능을 통해 비교 연산 ">="을 MAND 논리 연산의 두 번째 입력으로 이동시킵니다.
  - $(\rightarrow$  Basic instructions  $\rightarrow$  Comparator operations  $\rightarrow$  CMP>=)

'U1516F [CPU 1516F-3 PN/DP] > Program	ıblocks ▶ MOT	OR_SPEEDCONTRO	DL [FC10]	_ 7	×	Instructions	
						Options	
ත් ත් 🥩 👻 🌭 🔚 🚍 🗩 🖏 හි	± 🖃 🗊 🥙 🤅	o 🖑 🕬 🕹 📭	1 of the second	B		init .	int 🗌 🛄
MOTOR_SPEEDCONTROL						> Favorites	
Name	Data type	Default value	Comment			✓ Basic instructions	
1 📶 🔻 Input					^	Name	Versi
2 🔕 = Setpoint_speed	Real				=	Ceneral	versi
3 📶 🔻 Output						Bit logic operations	
4 📶 = Manipulated_variable_speed_AO	Int						
5 📶 🔻 InOut					~		
<					>	Comparator operations	nr.
						EL CMP	=
a >=1 [??] ⊣ −ol ↦ -[=]							
Block title: Speed control via analog output							-
Comment							
comment							
▼ 🕄 Network 1: check setpoint speed for corre	ct input range +/- 5	0 r/min					
Comment							
comment.					_		
1					=		
777					-196		
27775 1615	0						
2222 1112	a	~77.7~				Variant	
NIN INZ		Same				Math functions	
-77 7		-				Move operations	
						Conversion operation	15
\$~ <b>+</b>						Program control oper	ati

→ 다음 그림과 같이 네트워크 1의 접점을 상수 및 로컬 태그와 연결합니다. 비교 연산의 데이터 타입이 "REAL"로 자동 변경됩니다.



- → +/- 50 rpm이라는 속도 설정값을 +/- 1으로 정규화하기 위해 끌어다 놓기 기능을 통해 변환 연산 "NORM\_X"을 네트워크 2로 이동시킵니다.
  - $(\rightarrow$  Basic instruction  $\rightarrow$  Conversion operations  $\rightarrow$  NORM\_X)

.'U1516F [CPU 1516F-3 PN/DP] → Program	n blocks ▶ MOTO	R_SPEEDCONTRO	DL [FC10]		Ins	structions	1	
					Op	tions		
s 🗚 🥩 👻 🌭 🖿 🗖 🗖 🖓 🏖 🖉	1 = = 🗊 🥙 😡	😹 🍓 🤣 🖣	1 of 12 IV			ent eq		
MOTOR_SPEEDCONTROL					>	Favorites		
Name	Data type	Default value	Comment		~	Basic instructions	_	
🤕 🔻 Input				^	Nan	ne	Versi	
🕣 = Setpoint_speed	Real			=		General		
🤕 🔻 Output						Bit logic operations		F
🛛 💶 🔹 Manipulated_variable_speed_AO	Int				l i	Timer operations		
an 🕶 🕶 📶				*	<b>.</b>	+1 Counter operations		
<			2	>		Comparator operations		
						± Math functions		1
						Move operations		
>=	#Setpo	pint_		^	-	Conversion operations		
Real	speed	_ок				CONVERT		
#Setpoint_speed — IN1	=				-	E ROUND		
-50.0 <u></u> IN2 #		<u> </u>		14		CEIL		
						FLOOR		
N				=		TRUNC		
Network 2: Normalise setpoint speed from +	- 50 r/min to +/-1					SCALE_X		
Comment						NORM_X		
						Legacy		-
N					•	Program control operati.		
						拱 Word logic operations		
						Shift and rotate		

→ 아래 그림과 같이 네트워크 2의 접점을 상수 및 로컬 태그와 연결합니다. 'NORM\_X'의 데이터 타입이 "REAL"로 자동 변경됩니다.

!	U15	516	F [CPU 1516F-3 PN/DP] • Program	blocks ▶ MOTO	R_SPEEDCONTRO	DL [FC10] 📃		Instructions	• 🗉 🕨
								Options	
Ŕ	i R	S16F [CPU 1516F-3 PN/DP] > Program blocks > MOTOR_SPEEDCONTROL [FC10]       Instructions         Options         X ※ ※ % % ※ ※ % % ※ ※ ※ ※ % % % % % % %							
	M	ото	DR_SPEEDCONTROL	· · · · · · · · · · · · · · · · · · ·				> Favorites	_
		Na	me	Data type	Default value	Comment		✓ Basic instructions	_
	-	•	Input				^	Name Ve	rsi
2	-		Setpoint_speed	Real				▶ General	
ř.		•	Output				=	Bit logic operations	
8	-00		Manipulated_variable_speed_AO	Int				Timer operations	
5	-	•	InOut					+ Counter operations	
5			<add new=""></add>					Comparator operations	
7	-	•	Temp					Math functions	
3			Setpoint_speed_OK	Bool				Move operations	
9			Manipulated_variable_speed_Norm	Real			*	- Conversion operations	
	<						>		
								E ROUND	_
8		> = 1						E CEIL	
3	Net	wo	rk 2: Normalise setpoint speed from +/- 5	50 r/min to +/-1			^	FLOOR	
1	om	me	at .						
	-0111	in c					_	SCALE_X	
			NOPM	_				NORM_X	
			Real to Real				10	🕨 🎦 Legacy	
			Contraction of the second s					▶ 🖬 Program control operati	
		# S						Word logic operations	
			0.0 - MIN	#Manipula	ted_			🕨 🛱 Shift and rotate	
-	#Set	poi	nt speed - VALUE	OUT - Norm	reed_			FTC Legacy	
	0.077-074-0		50.0 - MAX	ENO -				•	>
								> Extended instructions	

→ 속도 설정값을 정규화된 +/- 1 rpm로부터 아날로그 출력 범위인 +/- 27468로 스케일링하기 위해 끌어다 놓기 기능을 통해 변환 연산 "SCALE\_X"를 네트워크 3으로 이동시킵니다.

							anni ana	
ä 🔏 🖈 🐔 🍬 🖿 🖻 🖻 🗩 🗐 📲 🖉	i±⊟≣ (° 6	e 🕼 🧐 🦆 🧤	4 8 ° 14			ptions init init		
MOTOR_SPEEDCONTROL					>	Favorites	_	
Name	Data type	Default value	Comment		~	Rasic instructions		
an 💌 Input				^	Na	me	Verci	
🕘 = Setpoint_speed	Real			=		General	VC131	
al 🕶 Output					1.	Bit logic operations		F
🛛 💶 🔹 Manipulated_variable_speed_AO	Int				Ľ	Timer operations		
🤕 🔻 InOut					Ľ	Counter operations		
Add new>				~		Comparator operations		
<			1	>	1.	Math functions		=
				1		Move operations		
& >=1 [??] → -0  → -[=]					-	Conversion operations		
NORM_X Peal to Peal				^		E ROUND		
Real to Real								
#Setpoint_						FLOOR		
speed_OK — EN	#Manipul	ated_				TRUNC		
#Sataciat canad	variable_	speed_				SCALE_X		
#Setpoint_speed VALUE						INORM_X		
50.0 - MAX	ENO					Legacy		
					•	Program control operati		
Network 3: Scale normalised setpoint speed	for analog output to	+/- 27648		=	•	Word logic operations		
Comment					•	🛱 Shift and rotate		
comment					•	ETC Legacy		v
					1	and the second s	1	

(→ 기본 명령 → 변환 연산 → SCALE\_X)

→ 여기 그림에서와 같이 네트워크 3의 접점을 상수 및 로컬 태그와 연결합니다. "SCALE\_X"의 데이터 타입이 "REAL" 또는 "INT"로 자동 변경됩니다.



→ 네 번째 네트워크에 Assignment 【=】를 삽입합니다. 끌어다 놓기 기능을 통해 Basic instruction 아래 Move operations 폴더에서 "Move" 명령을 "Assignment" 앞으로 이동시킵니다.

'U1516F [CPU 1516F-3 PN/DP] 🕨 Program	nblocks ▶ MOTOR_	SPEEDCONTRO	L [FC10] 🗕		In	structions		1)
					0	ptions		
u 🕹 🔮 🔮 💺 🔚 🚍 💬 🕾 🕹	± 🖃 🗊 🥙 😡 🖉	🖩 😪 🕹 📭	5 8 😤 🗷		E	ini ini		
MOTOR_SPEEDCONTROL					>	Favorites		
Name	Data type	Default value	Comment		~	Basic instructions		
1 📶 🔻 Input				^	Na	me	Versi	
2 📲 Setpoint_speed	Real			=	•	General		
🗧 🐨 Output						Bit logic operations		Ì
Manipulated_variable_speed_AO	Int	]			•	Imer operations		
anout 🗸 🐨				~	•	+1 Counter operations		
<				>	•	Comparator operations		
					•	± Math functions		
& >=1 <sup>[</sup> ??] ⊢ −ol ↦ –[=]					-	🔁 Move operations		
#Manipulated	#Manipu	lated		~		MOVE		
variable_speed_	variable_	speed_		1000		🗉 Deserialize	<u>V1.1</u>	
Norm — VALUE	ON - TUO					Serialize	<u>V1.1</u>	
27648 — MAX	ENO -					MOVE_BLK		
						MOVE_BLK_VARIANT	<u>V1.2</u>	
				_	1.	UMOVE_BLK		
Network 4: Speed setpoint out of range +	/- 50 r/min -> Manipulate	d_variable_speed	_AO = 0 / Return =	TRUE		FILL_BLK		
Comment								
						SWAP		
A CONTRACTOR OF						Array DB		
<sup>5</sup> "(+)				=		🕨 🛅 Variant		

→ 다음 그림과 같이 네트워크 4의 접점을 상수 및 로컬 태그와 연결합니다. 속도 설정값이 +/ 50 rpm 범위를 벗어날 경우에는 값 '0'이 아날로그 출력에 출력되고

"MOTOR\_SPEEDCONTROL" 펑션의 반환 값(Return)에 TRUE 값이 할당됩니다.

!	115	516	F [CPU	151	6F-3	PN/DF	?] ▶ F	Program	blocks 🕨 N	MOTOR_	SPEEDCONTR	DL [FC10]	_ 7 =	×	In	structions		• •
															Op	otions		
Ŕ	В	1	éž ≝ž	8	E		01	🛛 ± 🖉 :	• 🖃 🗊 🕐	° 🕻 🖉	😔 😍 🖣	¥ 🔗 🚏 [				ini ini		
	MC	ото	R_SPE	EDC	ONTR	OL									>	Favorites		_
		Na	me						Data type		Default value	Comment		-	~	Basic instructions		_
7	-	•	Temp											^	Nar	me	Versi.	
8	-		Set	point	speed	d_ok			Bool						+	General		~
9			Man	nipula	ted_v	ariable	_speed	d_Norm	Real					=		- Bit logic operations		
10	-	•	Consta	nt										260		Timer operations		
11			<ad< td=""><td>d nev</td><td>N&gt;</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>~</td><td>+</td><td>+1 Counter operations</td><td></td><td></td></ad<>	d nev	N>									~	+	+1 Counter operations		
	<								III				>			Comparator operations		
-			- 1 I				1		· · · · ·					-		± Math functions		-
8		> = 1	??	-	-01	$\rightarrow$	-[=]								-	Move operations		
																MOVE		
•	1	Net	work 4	: Sp	eed se	etpoint	t out of	range +/-	50 r/min -> Ma	anipulated	_variable_spee	d_AO = 0 / Return	n = TRUE	^		Deserialize	V1.1	-
	C	lom	ment													Serialize	V1.1	
																MOVE BLK		
					1	MOV	E									MOVE BLK VARIANT	V1.2	
								#Manin	ulated	#MO	TOR_					UMOVE_BLK		
			#Setp	oint_				variabl	e_speed_	SPEEDCO	ONTROL					FILL_BLK		
			speed	J_OK	-0 EN	1 - <del>*</del> * 1	OUTI -	_ AO		1	()					UFILL_BLK		
				0	- IN	L	ENO			-						SWAP		
																Array DB		

→ 💽 Save project 를 클릭하여 프로젝트를 저장합니다. 완료된 펑션 "MOTOR\_SPEEDCONTROL" [FC10]이 FBD 언어로 아래와 같이 나타납니다.



### 7.3 아날로그 출력 채널 구성

→ "Device configuation"을 더블클릭하여 엽니다.



→ 아날로그 출력 채널 0의 주소 설정 및 구성을 확인합니다.

 $(\rightarrow$  Q Address: 64...71 → Properties → General → Output 0 - 3 → Output → Channel 0 → Output type: Voltage → Output range: +/- 10 V → Reaction to CPU STOP: Shutdown)

032-500_Analog_Values	CPU1516F [CPU 1516F-3 PN/DP]					_ 7	
		📑 Topology view	Netwo	rk view	🚺 De	vice view	N
CPU1516F		Device overview					
30	1 5 5	A Module	Rack	Slot	I address	Q address	5
201	acht use one at	PM 190W 120/230VAC	0	0			
2014 5164	24VU 2224 WIRT UNST	CPU1516F	0	1			
ANTS PUTS	322 - Q 31 35t 29 11	PROFINET interface_1	0	1 X1			
~ U	0.0 4 4	PROFINET interface_2	0	1 X2			
		DP interface_1	0	1 X3			L
0 1	2 2 4 5 6	DI 32x24VDC HF_1	0	2	03		
	2 3 4 5 0	DQ 32x24VDC/0.5A ST_1	0	З		03	
Rail_0		AI 8xU/I/RTD/TC ST_1	0	4	6479		
		AQ 4xU/I ST_1	0	5		6471	
			0	6			
			0	7			
			0	8			
<u> <u> </u></u>			0	9			
			0	10			
			0	11			
			0	17		1	ľ
AQ 4×U/I ST_1 [AQ 4×U/I S	T]	Sector Properties	Info 追	🖁 Diag	inostics		
General IO tags	System constants Texts						
General							
Module parameters	Channel 0						-
• Output 0 - 3							
General	Parameter settings	Manual				-	
✓ Outputs	Turbine cer se cungs						
Channel 0	Diagnostics						
Channel 1							
Channel 2		No supply voltage L+					
Channel 3		Wire break					
I/O addresses		Short circuit to ground					
Hardware identifier							
	•	Underflow					
	Output						
	Output type:	Voltage				-	
	Output range:	+6-10				V	
	Reaction to CPU STOP	Shutdown					
	Cubating and a	- Shadowi)					
	Substitute value:						

### 7.4 아날로그 신호를 포함하도록 태그 테이블 확장

→ "Tag table\_sorting station"을 더블클릭하여 엽니다.



→ 아날로그 값 처리를 위한 글로벌 태그를 "Tag table\_sorting station"에 추가합니다. 아날로그 입력 B8 및 아날로그 출력 U1을 추가해야 합니다.

 $(\rightarrow U1 \rightarrow \%QW64 \rightarrow B8 \rightarrow \%IW64)$ 

032	-500_	Analog_Values 🕨 C	PU1516F [CPU 1516]	F-3 PN/DP] → P	LC tags 🔸	Tag tab	le_sorti	ng station [30]	_ = = ×
								🕣 Tags	User constants
,	jý [	) 🕾 🕅							
1	ag ta	ble_sorting station							
	N	lame	Data type	Address	Retain	Visibl	Acces	Comment	
15	-	-55	Bool	%I1.6				pushbutton manual mode c	ylinder -M4 retract (no) 🔺
16	-	-56	Bool	%11.7				pushbutton manual mode c	dinder -M4 extend (no)
17	-	-Q1	Bool	%Q0.0				conveyor motor -M1 forwards	fixed speed
18	-	-Q2	Bool	%Q0.1				conveyor motor -M1 backwar	rds fixed speed
19	-	-Q3	Bool	%Q0.2				conveyor motor -M1 variable	speed
20		-M2	Bool	%Q0.3				cylinder -M4 retract	
21	-	-MB	Bool	%Q0.4				cylinder -M4 extend	
22	-	-P1	Bool	%Q0.5				display "main switch on"	
23		-P2	Bool	%Q0.6				display "manual mode"	
24	-	-P3	Bool	%Q0.7				display "automatic mode"	
25		-P4	Bool	%Q1.0				display "emergency stop act	ivated"
26	-	-P5	Bool	%Q1.1				display "automatic mode sta	rted"
27	-	-P6	Bool	%Q1.2				display cylinder -M4 "retracte	d″ 🔤
28	-	-P7	Bool	%Q1.3				display cylinder -M4 "extende	:d*
29	-	-U1	Int	%QW64				manipulated value speed in	2 directions +/- 10V
30	-	-88	Int	%IW64				sensor actual value speed 0	10V
31		<add new=""></add>							

### 7.5 오거나이제이션 블록에서 블록 호출

→ "Main [OB1]" 오거나이제이션 블록을 더블클릭해서 엽니다.



→ 임시 태그 'Motor\_speed\_monitoring\_Ret\_Val'을 OB1의 로컬 태그에 추가합니다. 이는 "MOTOR\_SPEEDCONTROL" 평션의 반환 값을 상호 연결(interconnect)하기 위해 필요합니다.

 $(\rightarrow \text{Temp} \rightarrow \text{Motor\_speed\_monitoring\_Ret\_Val} \rightarrow \text{Bool})$ 

03	32-5	00	_Analog_Values  ▶ CPU1516F [CPU 1]	516F-3 PN/DP] 🕨	Program blocks	Main [OB1]	_ <b>=</b> = ×
ю	ю	K I	🖗 🕐 🌭 🖿 🗖 🖬 💬 📲 📲 🗄	= 😥 🧐 😡 🖉	≣ 🖓 🥵 📲	8 😤 🔢	
	Ma	in					
		Na	me	Data type	Default value	Comment	
1	-	-	Input				
2			Initial_Call	Bool		Initial call of this OB	
3			Remanence	Bool		=True, if remanent da	ata are available
4	-	•	Temp				
5			Motor_speed_monitoring_Ret_Val	Bool			
6			<add new=""></add>				
7	-	•	Constant				
8			<add new=""></add>				

→ OB1의 블록 타이틀을 선택한 다음, 📷를 클릭해 기존의 네트워크 앞에 새로운 네트워크 1을 삽입합니다.

03	2-5	00	_Analog_Values → CPU1516F [CPU 151	6F-3 PN/DP] → Pi	ogram blocks 🕨	Main [OB1] 🛛 🗕 🖬	×∎
н	н	X	0 2 🐁 🖿 🗖 🗖 🖓 🗷 2 1 🖻	😥 😋 😜 🕅		°. ∞. ∏	
In	Ma	t ne	twork	Data type	Default value	Comment	
1	-	-	Input	ooto type	Delibare rollac	connent	
2	-		Initial_Call	Bool		Initial call of this OB	
3			Remanence	Bool		=True, if remanent data are available	e
F. 1	-	•	Temp				
	-		Motor_speed_monitoring_Ret_Val	Bool	]		
			<add new=""></add>				
12. z	-	-	Constant				
			<add new=""></add>				
	<			III			:
•	<b>Blo</b>	me	title: *Main Program Sweep (Cycle)* nt				1
•	1	Net	twork 1: Control conveyor motor forwards in a ment	utomatic mode			1

→ 끌어다 놓기 기능을 이용해 "MOTOR\_SPEEDCONTROL [FC10]" 펑션을 네트워크 1의 녹색 라인으로 이동시킵니다.

M Siemens - G:\Automation\032-500_An	alog_V	alue	s\03	2-50	0_Analog_Values				_ L	⊐ ×
Project Edit View Insert Online Op Project 🛃 🌄 Save project 📇 💥 🗐 🔒	ptions X	Tool t (?	s • ±	Wind	ow Help 🔃 🏠 🖳 📮 💋 Go online 🖉 (	So offline 🔥 🖪 📑	* 🗆 🛛	Totally	Integrated Automation PORTAI	L
Project tree		•				[CPU 1516F-3 PN/DP]			[OB1] 💶 🖬 🖬 🗙	
Devices										-
1 1 0 0		2	1	<u>ل</u> م		<b>黒±</b> 目診 他 6-	A G I I -	1_ 0 00		Ins
5		-	1.5.4	Main						true
▼ 32-500_Analog_Values		~		N	lame	Data type	Default v	alue Comn	nent	tio
Add new device			1	-00 -	Input					SU
Devices & networks			2	-	Initial_Call	Bool		Initial	call of this OB	L
CPU1516F [CPU 1516F-3 PN/DP]			3	-	Remanence	Bool		=True	, if remanent data are available	8
Device configuration			4	-	Temp					e
🖳 🗓 Online & diagnostics			5	-	Motor_speed_monitoring_Ret_Va	al Bool				SUI
🔻 🕁 Program blocks			6		<add new=""></add>					6L
Add new block		=	7	-	Constant					
Hain [OB1]			8		<add new=""></add>					1
MOTOR_SPEEDCONTROL [FI	C10]			<					>	a
MOTOR_AUTO [FB1]			_				역			SKS
MOTOR_AUTO_DB [DB1]			•	RIOCH	title: Main Program Sweep (Cycle)				<u>^</u>	
🕨 🉀 Technology objects			0	omm	ent					
🕨 🔙 External source files			•	Ne	etwork 1:					Ξ
PLC tags									=	ora
PLC data types				COL	nment					rie
Watch and force tables					MOTOR_SPEEDCONTROL	[FC10]			L	Ĩ
🕨 📴 Online backups										
🕨 🔛 Traces										

→ 아래 그림과 같이 접점을 상수, 그리고 글로벌 및 로컬 태그와 연결합니다.

03	2-5	00	_Analog_Values  ▶ CPU1516F [CPU 151	6F-3 PN/DP] → Pro	ogram blocks 🔸	Main [OB1] _ 🗕 🖬 🖬 🗙
ю	Ю	in	# # % E A E 9 8 ± 8 ± 8	😥 🧐 💊 🖓 🕅	1 <b>1</b> ≩ 1 <sub>≣</sub> 1 <sub>≝</sub> (	
		Na	me	Data type	Default value	Comment
1	-	-	Input			
2	-		Initial_Call	Bool		Initial call of this OB
3	-		Remanence	Bool		=True, if remanent data are available
4	-	•	Temp			
5	-		Motor_speed_monitoring_Ret_Val	Bool	1	
6			<add new=""></add>			
7	-00	•	Constant			
8			<add new=""></add>			
	<					2
•	Com	me <b>Vet</b>	nt <b>twork 1:</b> Speed control analog oputput convey	yor motor		
			%FC10 "MOTOR_SPEEDCONTROL"	#Motor_speed monitoring_Re	ī	

→ 아날로그 설정값으로 컨베이어 모터를 제어할 수 있도록 네트워크 2의 출력 태그 "Conveyor\_motor\_automatic\_mode"의 연결을 '-Q3' (컨베이어 모터 -M1 가변 속도)로 변경합니다.

(→ -Q3)



교육 시설 및 R&D 기관에서의 사용에는 제한이 없습니다. ⓒ Siemens AG 2017. All rights reserved. SCE\_KO\_032-500 \_S7-1500\_ 아날로그 값\_R1705.docx

### 7.6 프로그램 저장 및 컴파일

- → 프로젝트를 저장하려면 메뉴에서 🔚 Save project 버튼을 선택합니다. 모든 블록을 컴파일하려면 "Program blocks" 폴더를 클릭하고 메뉴에서 컴파일을 위한 뒢 아이콘을 선택합니다.
  - $(\rightarrow \square$  Save project  $\rightarrow$  Program blocks  $\rightarrow \square$ )



→ "Info" 아래의 "Compile" 영역에 블록이 성공적으로 컴파일이 되었는지 나타납니다.

2			S.F	roperties	i, Info	i	B Diagnos	tics	78 -
G	ieneral (1) Cross-references	Compile S	yntax						
3	🔒 🕕 Show all messages								
Co	mpiling completed (errors: 0; warnin	gs: 0)							
1	Path	Description			Go to	?	Errors	Warnings	Time
0	▼ CPU1516F				7		0	0	12:2
0	<ul> <li>Program blocks</li> </ul>				~		0	0	12:2
0	MOTOR_SPEEDCONTRO	Block was successfully	y compiled.		~				12:2
0	Main (OB1)	Block was successfully	y compiled.		~				12:2
0		Compiling completed	(errors: 0; w	arnings: 0)					12:2
<			III						>

#### 7.7 프로그램 다운로드



### 7.8 프로그램 블록 모니터링

→ 다운로드된 프로그램을 모니터링하려면 해당 블록을 열어야 합니다. 한 아이콘을 클릭해서
 모니터링을 활성화/비활성화할 수 있습니다.

(́→ N	lain [C	DB1]	→														
na	log_Va	lues	• C	PU 15	16F (	CPU 15	16F-3	B PN/DP	] ▶	Progra	m blo	cks 🕨	Mair	ı [OB	1]	_ •	∎×
ю <mark>я</mark> і	¥ ∌	2	R.	EE	3 📼	92	± -2	± 📰	🎓 🕻 iterfac	0 <b>6</b> 0	e = 6	\$	1 = 3	<u>ا</u>	°€ 2	012	
å	>=1	177	н	-01	→	-[-]			1						Mor	nitorin	g on/off
▼ B	lock titl	e: "	Main	Progra	m Swe	ep (Cycle	e)*										^
Co	mment						- /										
•	Netwo	ork 1:	Spe	eed co	ntrol a	nalog op	output	conveyo	or mot	or							
	Comme	nt															
					мот	OR_SPEE	Ma	Ret_V Ret_V variable	/al — d_ e_	#Motor monito Val %QW64	_speed ring_Re	1_ :t_					1
na	log Va	lues	15.0	EN Set	point_	speed	16F-3	speed_A EN 3 PN/DF	40 — 40 —	"-U1" Progra	m blo	cks )	Mair	n IOB	11	_ 6	
na	log_Va ĸX ≇	lues	15.0 ·	PU15	:point_ 16F [	speed CPU 15	16F-3	Speed_A EN PN/DF	40	Progra	im blo	cks ) ∎ �	Main I <sub>≣</sub> 1	n [OB	1]		
na ЮЯ́	log_Va toX ≇ >=1	lues ঊ	15.0 · • (	PU 15	:point_ 16F ( ] :=	speed CPU 15 (PD 15	16F-	Speed_F EN PN/DF	40	Progra	e e	cks ) ⊪ �	Main I = 1	n [OB	1]		
na ⊦õi i ª	log_Va kX ≇ >=1	lues ≝ి [??] le: *	15.0 ·	PU15	n Swe	speed CPU 15 (PU 25 (PU 15 (PU 15 (PU 15) (PU	16F-3 ± .⊒ e)*	speed_f EN B PN/DF	40 10 ?] ▶ ≋⊉ ( iterfac	Progra	im blo	cks )	Main I <sub>=</sub> 1	n [OB	1] •		
na ⊢औ ∎ ≈ В	log_Va KX ≇ >=1 lock titl mment	lues ≝ి (??) le: *I	15.0	PU 15	noint_ 16F [ ] := 	speed CPU 15 (PU 15 (PU 15 (PU 15 (PU 15) (PU	16F- ★ 2 e)*	speed_f EN B PN/DP	40 10 2] ▶ 10 11 11 11 11 11 11 11 11 11 11 11 11	Progra	en blo	cks	Main I <sub>=</sub> 1	n [OB '≘   d	1] •		
na ⊮Öi I & ▼ B Co	log_Va KX ﷺ >=1 lock titl mment Netwo	lues 같 [깜] le: *)	• C • C • C • C • C • C • C	PU15	ntrol a	speed CPU 15 (PU 15 () () () () () () () () () () () () ()	16F-3 ★	Speed_4 EN B PN/DF Block in Block in	AO	Progra	e e	cks )	Main I = 1	n [OB	1] P		
una KÖİl a ▼ B Co	log_Va KX ﷺ >=1 lock titl mment Networ Comme	lues	• • • • • • • • • • • • • • • • • • •	EN Set PU15	point_ 16F [ 	speed CPU 15 (PU 15 (CPU 15 (CPU 15 (CPU 15 (CPU 15) (CPU	16F. ± ₽ e)*	Speed_4 EN PN/DF Block in Conveyo	AO	Progra	e S	cks 🕨	Main I = 1	n [OB	11] P	05	
na ĕ ▼ B Co	log_Va ⇒=1 lock titl mment Networ Comme	lues 같 [깜] le: *I ork 1: ent	• C • C • C • C • C • C • C • C	EN Set	ipoint_ 16F [ 	speed CPU 15 (PU 15 (PU 15 (PU 15) (PU	16F- 2 @ e)* putput EDCOM	Speed_4 EN PN/DP E Block in Block in Block in Ret_V Ret_V NTROL	AO	FALSE #Moto monito Val 8294	r_speed	cks →	Main I <sub>=</sub> 1		1] • •		

- → 마우스 오른쪽 버튼을 클릭하여 "Open and monitor"에서 "Main [OB1]" 오거나이제이션 블록에서 호출된 "MOTOR\_SPEEDCONTROL" [FC10] 펑션을 직접 선택하여 펑션 블록의 프로그램 코드를 모니터링할 수 있습니다.
  - ( $\rightarrow$  "MOTOR\_SPEEDCONTROL" [FC10]  $\rightarrow$  Open and monitor)

na	log_V	alues	• 0	PU 15	16F [	CPU 15	16F-3 PN	I/DP]	Progr	am ble	ocks 🕨	Main	[OB1]	-	i E X
<u>.</u>	<b>X</b> =	: _\$	B <sub>2</sub>	= 2	1 :20	<b>.</b>	+ 22+	- 18%	<u>/0</u> (,	e 19	(m. 40b	1_ 1_	<u>_</u>		
Э	- ко	-					Blo	ck inter	face	( <u> </u>	- <u>-</u>			- LC.4	
8	>=1	177	4	-01	4	-[=]	1	A	- 004						
-						••									
	Comm	vork 1	: Sp	eed co	ntrol a	analog op	output con	iveyor n	notor						
F	conn	i cinc													_
					2401	%FC		Mo	dify				•		ļ
					MOI	UN_SPEC	DeoNine	00	en						
								Op	en and m	nonîtor					
								De	fine tag		3	Ctrl+Sh	ift+I		
							F	Re	name tag			Ctrl+Sh	ift+T		
							Manipu vai	V cu	wire tag			Cui+sni	IL+F		
				- EN			spe	Co	ру			Ct	rl+C		
			15.0	Set	point	speed		Pa	ste			Ct	rl+V		
Ċ				101				X De	lete				Del		
<del>я</del> ́ і	-X ₹	. ≣,	8			93	± 🚜 ± Blo	ck inter	tace	¢# (	₩ ¥	1 <sup>=</sup> <i>x</i> <sup>=</sup>	<b>%</b>		
all p	oath: M	ain [O	B1]												1
5	>=1	??	H	-01	↦	-[=]									
	Netw	ork 3	: Sci	ale nor	malis	ed setpoi	nt speed f	foranal	og output	t to +/- :	27648				
	Comm	nent													
F				1			40.000	14							_
						Real to	L_X Int								- 1
		1	TRUE												- 1
		#Setp	oint_												
		spee	0_0	EN MIN											
									0004						
	#Ma	anipuli	ated_						#Mani	pulate	d_				- 1
	varia	ble_sp	Norm	VAL	UE			OUT	variab AO	le_spe	ed_				
		2	7648	MA	X			- ENO	12 17 17 19 19 19 19 19 19 19 19 19 19 19 19 19						
-					1					09/		r	-		1
				11	1			4	1 10	0%			- II	Y	

#### 7.9 프로젝트 아카이브

→ 마지막 단계로 완전한 프로젝트를 아카이브하기 위해 "Project" 메뉴에서 "Archive..." 항목을 선택합니다. 프로젝트를 아카이브하고자 하는 폴더를 선택하고 "TIA Portal 프로젝트 아카이브" 파일 유형으로 이를 저장합니다.

(→ Project → Archive → TIA Portal Project archive → 032-500\_Analog\_Values.... → Save)



### 8 체크리스트

번호	설명	완료
1	오류 메시지 없이 성공적으로 컴파일	
2	오류 메시지 없이 성공적으로 다운로드	
3	스테이션 전원 켜기 (-K0 = 1) 실린더 복귀 / 피드백 활성화 (-B1 = 1) 비상 정지 오프 (-A1 = 1)가 활성화되지 않음 자동 모드 (-S0 = 1) 푸시버튼 자동 정지가 구동되지 않음 (-S2 =1) 자동 시작 푸시버튼을 짧게 누르기 (-S2 = 1) 슬라이드의 센서 부분이 활성화되고 (-B4 = 1) 컨베이어 모터 M1 가변 속도 (-Q3 = 1) 스위치를 켠 상태로 유지 속도는 +/- 50 rpm 범위의 속도 설정값에 해당	
4	컨베이어 끝의 센서가 활성화 (-B7 = 1) → -Q3 = 0 (2초 후)	
5	자동 정지 푸시버튼을 짧게 누르기 (-S2 = 0) → -Q3 = 0	
6	비상 정지 오프를 활성화 (-A1 = 0) → -Q3 = 0	
7	수동 모드 (-S0 = 0) → -Q3 = 0	
8	스테이션 전원 끄기 (-K0 = 0) → -Q3 = 0	
9	실린더가 복귀되지 않음 (-B1 = 0) → -Q3 = 0	
10	프로젝트 아카이브하기	

### 9 연습

#### 9.1 과제 - 연습

이 연습에서는 "MOTOR\_SPEEDMONITORING" [FC11] 펑션을 추가적으로 생성해 보겠습니다.

-B8 (모터의 센서 실제 속도 값 +/-10V은 +/- 50 rpm에 해당) 아날로그 값을 실제 속도값으로 사용하여, "MOTOR\_SPEEDMONITORING" [FC11] 펑션의 입력으로 쿼리가 수행됩니다. 데이터 타입은 16비트 정수 (INT)입니다.

이러한 실제 속도 값은 먼저 평션에서 32비트 부동 소수점 수 (REAL)로서 범위 +/- 1로 정규화됩니다.

정규화된 실제 속도 값은 32비트 부동 소수점 수 (REAL)로서 분당 회전수(범위: +/- 50 rpm)로 스케일링이 되어 출력으로 사용할 수 있는 값이 됩니다.

평션에서 모니터링을 할 수 있도록 아래와 같이 4개의 제한 값을 평션의 입력에 32비트 부동 소수점 수 (REAL)로서 사용할 수 있습니다.

- 속도 > Motor\_speed\_monitoring\_error\_max
- 속도 > Motor\_speed\_monitoring\_warning\_max
- 속도 < Motor\_speed\_monitoring\_warning\_min
- 속도 < Motor\_speed\_monitoring\_error\_min

제한 값을 초과하거나 미달하면 해당 출력 비트에 TRUE (1) 값이 지정됩니다.

고장이 발생하면 "MOTOR\_AUTO" [FB1] 펑션 블록에 대한 보호 트리핑이 작동됩니다.

### 9.2 기술 다이어그램

아래에 과제를 위한 기술 다이어그램이 나와 있습니다.



#### 그림 3: 기술 다이어그램

Schalter der Sortieranlage		Automatikbetrieb		Handbetrieb / Manual mode
Switches of sorting station		Automatic mode -S3 Tippbetrieb -M1 vorwärts/		-S3 Tippbetrieb -M1 vorwärts/
-P1 ein/on		-P5 gestartet/started		Manual -M1 forwards
-Q0 Hauptschalter/Main switch		-S1 Start/start		-S4 Tippbetrieb -M1 rückwärts/
-P4 aktiviert/active				-P7 ausgefahren/extended
-A1 NOTHALT/Emergency stop		-S2 Stopp/stop		-S6 Zvlinder -M4 ausfahren/
-P2 Hand/manual -P3 Auto/auto				cylinder -M4 extend -P6 eingefahren/retracted
-S0 Betriebsart/operating mode				-S5 Zylinder -M4 einfahren/
				cylinder -ma redact

그림 4: 제어 패널

### 9.3 참조 목록

이 과제를 위한 글로벌 오퍼랜드로서 아래와 같은 신호들이 필요합니다.

DI	유형	식별자	기능	NC/NO
I 0.0	BOOL	-A1	비상 정지 OK 반환 신호	NC
I 0.1	BOOL	-K0	메인 스위치 "온"	NO
I 0.2	BOOL	-S0	모드 선택 수동 (0) / 자동 (1)	수동 = 0
				자동 = 1
I 0.3	BOOL	-S1	푸시버튼 자동 시작	NO
I 0.4	BOOL	-S2	푸시버튼 자동 정지	NC
I 0.5	BOOL	-B1	센서 실린더 -M4 복귀	NO
I 1.0	BOOL	-B4	슬라이드의 센서	NO
I 1.3	BOOL	-B7	컨베이어 끝 센서	NO
IW64	BOOL	-B8	모터의 센서 실제 속도 값 +/- 10V는 +/- 50 rpm에 해당	

DO	유형	식별자	기능	
Q 0.2	BOOL	-Q3	컨베이어 모터 -M1 가변 속도	
QW 64	BOOL	-U1	2방향 모터의 조작 속도 값 +/- 10V는 +/- 50 rpm에 해당	

#### 참조 목록 범례

DI	Digital Input	DO	Digital Output
AI	Analog Input	AO	Analog Output
I	Input	Q	Output
NC	Normally Closed		

NO Normally Open

### 9.4 계획 수립

과제 수행에 대한 계획을 자체적으로 수립합니다.

### 9.5 체크리스트 - 연습

번호	설명	완료
1	오류 메시지 없이 성공적으로 컴파일	
2	오류 메시지 없이 성공적으로 다운로드	
3	스테이션 전원 켜기 (-K0 = 1) 실린더 복귀 / 피드백 활성화 (-B1 = 1) 비상 정지 오프 (-A1 = 1)가 활성화되지 않음 자동 모드 (-S0 = 1) 푸시버튼 자동 정지가 구동되지 않음 (-S2 =1) 자동 시작 푸시버튼을 짧게 누르기 (-S2 = 1) 슬라이드의 센서 부분이 활성화되고 (-B4 = 1) 컨베이어 모터 -M1 가변 속도 (-Q3 = 1) 스위치를 켠 상태로 유지 속도는 +/- 50 rpm 범위의 속도 설정값에 해당	
4	컨베이어 끝의 센서 부분이 활성화 (-B7 = 1) → -Q3 = 0 (2초 후)	
5	자동 정지 푸시버튼을 짧게 누르기 (-S2 = 0) → -Q3 = 0	
6	비상 정지 오프를 활성화 (-A1 = 0) → -Q3 = 0	
7	수동 모드 (-S0 = 0) → -Q3 = 0	
8	스테이션 전원 끄기 (-K0 = 0) → -Q3 = 0	
9	실린더가 복귀되지 않음 (-B1 = 0) → -Q3 = 0	
10	속도 > Motor_speed_monitoring_error_max → -Q3 = 0	
11	속도 < Motor_speed_monitoring_error_min → -Q3 = 0	
12	프로젝트가 성공적으로 아카이브 됨	

### 10 추가 정보

초기 및 심화 교육에 방향을 제시하는 도구의 차원에서 TIA Portal 모듈에 대한 추가 정보를 활용할 수 있습니다. 시작하기, 동영상, 교재, 앱, 매뉴얼, 프로그래밍 지침, 체험용 소프트웨어/펌웨어 등을 아래 링크에서 찾아보실 수 있습니다.

www.siemens.com/sce/s7-1500