

SCE Training Curriculum

Siemens Automation Cooperates with Education | 05/2017

TIA Portal Module 032-500 Analog Values for SIMATIC S7-1500



For unrestricted use in educational and R&D institutions. © Siemens AG 2017. All rights reserved.

Matching SCE trainer packages for these training curriculums

SIMATIC Controllers

- SIMATIC ET 200SP Open Controller CPU 1515SP PC F and HMI RT SW Order no.: 6ES7677-2FA41-4AB1
- SIMATIC ET 200SP Distributed Controller CPU 1512SP F-1 PN Safety Order no.: 6ES7512-1SK00-4AB2
- SIMATIC CPU 1516F PN/DP Safety Order no.: 6ES7516-3FN00-4AB2
- SIMATIC S7 CPU 1516-3 PN/DP Order no.: 6ES7516-3AN00-4AB3
- SIMATIC CPU 1512C PN with Software and PM 1507 Order no.: 6ES7512-1CK00-4AB1
- SIMATIC CPU 1512C PN with Software, PM 1507 and CP 1542-5 (PROFIBUS) Order no.: 6ES7512-1CK00-4AB2
- SIMATIC CPU 1512C PN with Software Order no.: 6ES7512-1CK00-4AB6
- SIMATIC CPU 1512C PN with Software and CP 1542-5 (PROFIBUS) Order no.: 6ES7512-1CK00-4AB7

SIMATIC STEP 7 Software for Training

- SIMATIC STEP 7 Professional V14 SP1 Single license Order no.: 6ES7822-1AA04-4YA5
- SIMATIC STEP 7 Professional V14 SP1- Classroom license (up to 6 users) Order no.: 6ES7822-1BA04-4YA5
- SIMATIC STEP 7 Professional V14 SP1 Upgrade license (up to 6 users) Order no.: 6ES7822-1AA04-4YE5
- SIMATIC STEP 7 Professional V14 SP1 Student license (up to 20 users) Order no.: 6ES7822-1AC04-4YA5

Please note that these trainer packages are replaced with successor packages when necessary. An overview of the currently available SCE packages is provided at: <u>siemens.com/sce/tp</u>

Continued training

For regional Siemens SCE continued training, please contact your regional SCE contact siemens.com/sce/contact

Additional information regarding SCE

siemens.com/sce

Information regarding use

The SCE training curriculum for the integrated automation solution Totally Integrated Automation (TIA) was prepared for the program "Siemens Automation Cooperates with Education (SCE)" specifically for training purposes for public educational and R&D institutions. Siemens AG does not guarantee the contents.

This document is to be used only for initial training on Siemens products/systems. This means it can be copied in whole or part and given to those being trained for use within the scope of their training. Circulation or copying this training curriculum and sharing its content is permitted within public training and advanced training facilities for training purposes.

Exceptions require written consent from the Siemens AG contact: Roland Scheuerer roland.scheuerer@siemens.com.

Offenders will be held liable. All rights including translation are reserved, particularly if a patent is granted or a utility model or design is registered.

Use for industrial customer courses is expressly prohibited. We do not consent to commercial use of the training curriculums.

We wish to thank the TU Dresden, especially Prof. Dr.-Ing. Leon Urbas, the Michael Dziallas Engineering Corporation and all other involved persons for their support during the preparation of this training curriculum.

Table of contents

1	Goa	l	. 5
2	Prer	equisite	. 5
3	Req	uired hardware and software	. 6
4	The	ory	. 7
	4.1	Analog signals	. 7
	4.2	Measuring transducers	. 8
	4.3	Analog modules – A/D converter	. 8
	4.4	Data types of the SIMATIC S7-1500	. 9
	4.5	Reading/outputting analog values	10
	4.6	Normalizing analog values	11
5	Tasl	<	12
6	Plan	ning	12
	6.1	Analog control of the conveyor speed	12
	6.2	Technology diagram	13
	6.3	Reference list	14
7	Stru	ctured step-by-step instructions	15
	7.1	Retrieve an existing project	15
	7.2	Create the "MOTOR_SPEEDCONTROL" function	17
	7.3	Configuration of the analog output channel	24
	7.4	Expand the tag table to include analog signals	25
	7.5	Call the block in the organization block	26
	7.6	Save and compile the program	29
	7.7	Download the program	30
	7.8	Monitor program blocks	31
	7.9	Archive the project	33
8	Che	cklist	34
9	Exe	rcise	35
	9.1	Task – Exercise	35
	9.2	Technology diagram	36
	9.3	Reference list	37
	9.4	Planning	37
	9.5	Checklist – Exercise	38
1() Add	itional information	39

ANALOG VALUES FOR SIMATIC S7-1500

1 Goal

In this chapter, you will become acquainted with the analog value processing of the SIMATIC S7-1500 with the TIA Portal programming tool.

The module explains the acquisition and processing of analog signals and gives a step-bystep description of read and write access to analog values in the SIMATIC S7-1500.

The SIMATIC S7 controllers listed in Chapter 3 can be used.

2 Prerequisite

This chapter builds on the chapter IEC Timers and Counters with the SIMATIC S7 CPU1516F-3 PN/DP. You can use the following project for this chapter, for example: 032-300 IEC Timers and Counters.zap13

3 Required hardware and software

- 1 Engineering station: requirements include hardware and operating system (for additional information, see Readme on the TIA Portal Installation DVDs)
- 2 SIMATIC STEP 7 Professional software in TIA Portal as of V13
- SIMATIC S7-1500/S7-1200/S7-300 controller, e.g. CPU 1516F-3 PN/DP –
 Firmware as of V1.6 with memory card and 16DI/16DO and 2AI/1AO
 Note: The digital inputs and analog inputs and outputs should be fed out to a control panel.
- 4 Ethernet connection between engineering station and controller



4 Theory

4.1 Analog signals

In contrast to a binary signal, which can assume only two signal states ("Voltage present +24 V" and "Voltage not present 0 V"), analog signals can assume any value within a defined range. A typical example of an analog sensor is a potentiometer. Depending on the position of the knob, any resistance can be set, up to the maximum value.

Examples of analog quantities in control engineering:

- Temperature -50 to +150 °C
- Flow rate 0 to 200 l/min
- Speed -500 to +50 rpm
- etc.

4.2 Measuring transducers

These quantities are converted to electrical voltages, currents or resistances with the help of a measuring transducer. If, for example, a speed is to be measured, the speed range of 500 to 1500 rpm can be converted to a voltage range of 0 to +10 V using a measuring transducer. At a measured speed of 865 rpm, the measuring transducer would output a voltage value of +3.65 V.



4.3 Analog modules – A/D converter

These electrical voltages, currents or resistances are then connected to an analog module that digitizes this signal for further processing in the PLC.

If analog quantities will be processed with a PLC, the read-in voltage, current or resistance value must be converted to digital information. The analog value is converted to a bit pattern. This conversion is referred to as analog-to-digital conversion (A/D conversion). This means, for example, that the voltage value of 3.65 V is stored as information in a series of binary digits.

The result of this conversion is always a 16-bit word for SIMATIC products. The integrated ADC (analog-to-digital converter) of the analog input module digitizes the analog signal being acquired and approximates its value in the form of a stepped curve. The most important parameters of an ADC are its resolution and conversion rate.



1: Analog value

2. Digital value

The more binary digits the digital representation uses, the finer the resolution is. For example, if only 1 bit was available for the voltage range of 0 to +10 V, you would only know whether the measured voltage is between 0 and +5 V or between +5 V and +10 V. With 2 bits, the range can be divided into 4 individual ranges, i.e., 0 to 2.5 / 2.5 to 5 / 5 to 7.5 / 7.5 to 10 V. Conventional A/D converters in control engineering use 8 or 11 bits for converting.

With 8 bits you have 256 individual ranges, while 11 bits provide a resolution of 2048 individual ranges.

0A/0V		20mA/1	10V
	11-bit		10 V: 2048 = 0.0048828 \rightarrow Voltage differences of <5 mV can be detected
0		2048	

4.4 Data types of the SIMATIC S7-1500

The SIMATIC S7-1500 has many different data types for representing different numerical formats. A list of some of the elementary data types is given below.

Data type	Size (bits)	Range	Example of constant entry
Bool	1	0 to 1	TRUE, FALSE, O, 1
Byte	8	16#00 to 16#FF	16#12, 16#AB
Word	16	16#0000 to 16#FFFF	16#ABCD, 16#0001
DWord	32	16#00000000 to 16#FFFFFFF	16#02468ACE
Char	8	16#00 to 16#FF	'A', 'r', '@'
Sint	8	-128 to 127	123,-123
Int	16	-32,768 to 32,767	123, -123
Dint	32	-2,147,483,648 to 2,147,483,647	123, -123
USInt	8	0 to 255	123
UInt	16	0 to 65,535	123
UDInt	32	0 to 4,294,967,295	123
Real	32	+/-1.18 x 10 ⁻³⁸ to +/-3.40 x 10 ³⁸	123.456, -3.4, -1.2E+12, 3.4E- 3
LReal	64	+/-2.23 x 10 ⁻³⁰⁸ to +/-1.79 x 10 ³⁰⁸	12345.123456789 -1.2E+40
Time	32	T#-24d_20h_31 m_23s_648ms to T#24d_20h_31 m_23s_647ms Saved as: -2,147,483,648 ms to +2,147,483,647 ms	T#5m_30s 5#-2d T#1d_2h_15m_30x_45ms
String	Variable	0 to 254 characters in byte size	'ABC'

Note: The **'INT'** and **'REAL'** data types play a large role in analog value processing. This is because read-in analog values exist as 16-bit integers in the **'INT'** format, and in order to ensure exact further processing only **'REAL'** floating-point numbers should be used due to rounding errors in the case of **'INT'**.

4.5 Reading/outputting analog values

Analog values are read into the PLC or output from the PLC as word information. These words are accessed, for example, with the following operands:

%IW 64	Analog input word 64
%QW 64	Analog output word 64

Each analog value ("channel") occupies one input or output word. The format is '**Int'**, an integer.

The addressing of input and output words conforms to the addressing in the device overview. For example:



Here, the address of the first analog input would be %IW 64, that of the second analog input %IW 66, that of the third analog input %IW68, that of the fourth analog input %IW70, that of the fifth analog input %IW72, that of the sixth analog input %IW74, that of the seventh analog input %IW 76 and that of the eighth analog input %IW78.

The address of the first analog output would be %QW64, that of the second analog output %QW66, that of the third analog output %QW 68 and that of the fourth analog output %QW70.

The analog value transformation for further processing in the PLC is the same for analog inputs and analog outputs.

The digitized value ranges are as follows:



Often, these digitized values still have to be normalized by further processing them in the PLC in an appropriate manner.

4.6 Normalizing analog values

If an analog input value exists as a digitized value in the range +/- 27648, it must usually still be normalized so that the numerical values correspond to the physical quantities in the process.

Likewise, the analog output usually results from setting of a normalized value that then still has to be scaled to the output value +/- 27648.

In the TIA Portal, ready-made blocks or arithmetic operations are used for normalizing and scaling.

For this to be carried out as exactly as possible, the values for the normalizing must be converted to the REAL data type to minimize rounding errors.

5 Task

In this chapter, a function for analog control of the conveyor speed will be added to the program from chapter "SCE_EN_032-300 IEC Timers and Counters".

6 Planning

The analog control of the conveyor speed will be programmed in the "MOTOR_SPEEDCONTROL" [FC10] function as an expansion of the "SCE_EN_032-300 IEC Timers and Counters" project. This project must be retrieved from the archive in order to add this function. The "MOTOR_SPEEDCONTROL" [FC10] function will be called in the "Main" [OB1]" organization block and wired. The control of the conveyor motor must be changed to - Q3 (conveyor motor -M1 variable speed).

6.1 Analog control of the conveyor speed

The speed will be set at an input of the "MOTOR_SPEEDCONTROL" [FC10] function in revolutions per minute (range: +/- 50 rpm). The data type is 32-bit floating-point number (Real).

First, the function will be checked for correct entry of the speed setpoint in the range +/- 50 rpm.

If the speed setpoint is outside the range +/- 50 rpm, the value 0 with data type 16-bit integer (Int) will be output at the output. The return value of the function (Ret_Val) will then be assigned the value TRUE (1).

If the speed setting is within the range +/- 50 rpm, this value will first be normalized to the range 0...1 and then scaled to +/- 27648 with data type 16-bit integer (Int) for output as the speed manipulated value at the analog output.

The output will then be connected with signal U1 (manipulated value speed of the motor in 2 directions +/- 10V corresponds to +/- 50 rpm).

6.2 Technology diagram

Here you see the technology diagram for the task.



Figure 1: Technology diagram

Schalter der Sortieranlage Switches of sorting station -P1 ein/on	Automatikk Automatic	etrieb mode -P5 gestartet/started	Handb -S3 Tip Ma	etrieb / Manual mode pbetrieb -M1 vorwärts/ nual -M1 forwards	
-Q0 Hauptschalter/Main switch	-S1 Start/start		-S4 Tip Ma	pbetrieb -M1 rückwärts/ nual -M1 backwards	
-P4 aktiviert/active -P4 aktiviert/active -P4 aktiviert/active -P2 Handimanual -P3 Auto/auto	-S2 Stopp/stop		-S6 Zyl	inder -M4 ausfahren/ nder -M4 extend	-P7 ausgefahren/extended
-S0 Betriebsart/operating mode			-S5 Zyl	inder -M4 einfahren/ inder -M4 retract	•

Figure 2: Control panel

6.3 Reference list

DI	Туре	Identifier	Function	NC/NO
I 0.0	BOOL	-A1	Return signal emergency stop OK	NC
I 0.1	BOOL	-K0	Main switch "ON"	NO
I 0.2	BOOL	-S0	Mode selector manual (0)/ automatic (1)	Manual = 0 Auto = 1
I 0.3	BOOL	-S1	Pushbutton automatic start	NO
I 0.4	BOOL	-S2	Pushbutton automatic stop	NC
I 0.5	BOOL	-B1	Sensor cylinder -M4 retracted	NO
I 1.0	BOOL	-B4	Sensor part at slide	NO
l 1.3	BOOL	-B7	Sensor part at end of conveyor	NO

The following signals are required as global operands for this task.

DO	Туре	Identifier	Function	
Q 0.2	BOOL	-Q3	Conveyor motor -M1 variable speed	
QW 64	BOOL	-U1	Manipulated value speed of the motor in 2 directions +/- 10V corresponds to +/- 50 rpm	

Legend for reference list

AI

DI	Digital Input	DO	Digital Output

Analog Input AO Analog Output

Q

Output

- I Input
- NC Normally Closed
- NO Normally Open

7 Structured step-by-step instructions

You can find instructions on how to carry out planning below. If you already have a good understanding of everything, it will be sufficient to focus on the numbered steps. Otherwise, simply follow the detailed steps in the instructions.

7.1 Retrieve an existing project

→ Before we can expand the "SCE_EN_032-300_IEC_Timers_Counters.zap13 project from chapter "SCE_EN_032-300_IEC_Timers_Counters", we must retrieve this project from the archive. To retrieve an existing project that has been archived, you must select the relevant archive with → Project → Retrieve in the project view. Confirm your selection with Open.

V13	Sieme	ens			
Pro	oject	Edit	View	Insert	0
÷	New				
	Open.			Ctrl+O	
	Migrat	te proj	ect		
	Close			Ctrl+W	
	Save			Ctrl+S	
	Save a	as	Ctr	rl+Shift+S	
		-		Challer F	
	Delete	e proje	ct	Ctri+E	
	Archiv	e	ct	Ctri+E	 Y
	Archiv Retriev	e proje e ve		Ctri+E	ÿ
-	Archiv Retriev Card R	e proje e ve keader	USB me	emory	ÿ
-	Archiv Retriev Card R Memo	e proje e ve teader iry care	/USB me d file	emory	y
-	Archiv Retriev Card R Memo	e proje e ve keader ory care de	/USB me	emory I	y
-	Archiv Retriev Card R Memo Upgra Exit	e proje e ve Reader ory card	/USB me	emory I	y •

 $(\rightarrow \text{Project} \rightarrow \text{Retrieve} \rightarrow \text{Select a .zap archive} \rightarrow \text{Open})$

→ The next step is to select the target directory where the retrieved project will be stored. Confirm your selection with "OK".

 $(\rightarrow \text{Target directory} \rightarrow \text{OK})$

 \rightarrow Save the opened project under the name 032-500_Analog_Values.

游 Siemens - G:\Automation\032_300_IEC_Timers_Counters\032_300_IEC_Timers_Counters . • X Totally Integrated Automation PORTAL Project Edit View Insert Online Options
Project New...
Options
Options
Ctrl+O
Migrate project...
Close
Ctrl+W : Tools Window Help) 호 (예 호 웹 대 대 및 및 정 Go online & Go offline 🏭 대 대 🕺 🖃 🛄 Save Save as.. Tasks Delete project. Archive... Retrieve... Ctrl+E T Card Reader/USB memory Card Reader/USB memory G:lAutomat....1032_300_IEC_Timers_Counters G:lAutomation1...1032_200_FB-Programming G:lAutomation1...1032_200_FB-Programming Exit 🖻 Properties 🚺 Info 👔 🖞 Diagnostics 👘 🗖 🗖 🤜 General No 'properties' available. No 'properties' can be shown at the moment. There is either no object selected or the selected object does not have any displayable properties > Details view Portal view

 $(\rightarrow \text{Project} \rightarrow \text{Save as} \dots \rightarrow 032\text{-}500\text{-}\text{Analog}\text{-}\text{Values} \rightarrow \text{Save})$

7.2 Create the "MOTOR_SPEEDCONTROL" function

→ Select the 'Program blocks' folder of your CPU 1516F-3 PN/DP and then click "Add new block" to create a new function there.



 $(\rightarrow \text{CPU}_{1516F} \text{ [CPU 1516F-3 PN/DP]} \rightarrow \text{Add new block})$

 \rightarrow Select **The in the next dialog and rename your new block to:**

"MOTOR_SPEEDCONTROL". Set the language to FBD and manually assign the number "10". Select the "Add new and open" check box. Click "OK".

(→ $\stackrel{\blacksquare}{\longrightarrow}$ → Name: MOTOR_SPEEDCONTROL → Language: FBD → Number: 10 Manual → $\stackrel{\blacksquare}{\blacksquare}$ Add new and open → OK)

Add new block					×
Name:					
MOTOR_SPEEDCONTR	ROL				
Organization	Language: Number:	FBD 10 Manual	v		
Function block	Description: Functions are o	code blocks or subrout	tines without dedi	cated memory.	
Function					
Data block	More				
> Additional inform	mation		_		
Add new and open				ОК	Cancel

→ Create the local tags with their comments as shown here and change the data type of the 'Return' tag from 'Void' to 'Bool'.

 $(\rightarrow \text{Bool})$

	Val	ues	→ CF	VU151	16F [(CPU 1	516F-3 PN	/DP] 🕨 Progra	m blocks	MOTOR_	SPEEDCONTROL [FC10]	_∎≣×
юã	a a # # = = = = = = = = = = = = = = = =											
	MO	то	R_SPE	EDCO	DNTR	OL						
		Nar	me					Data type		Default value	Comment	
1	-00	•	Input									
2	-00	•	Set	point_	speed			Real				
3	-00	•	Output	t								
4	-	•	Ma	nipula	ted_va	riable.	_speed_AO	Int				
5		•	InOut									
6		•	<ad< td=""><td>ld new</td><td>/></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></ad<>	ld new	/>							
7		•	Temp									
8		•	Set	point_	speed	_ок		Bool				
9		•	Ma	nipula	ted_va	riable.	_speed_Nor	m Real				
10		•	Consta	nt								
11		•	<ac< td=""><td>ld new</td><td>/></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></ac<>	ld new	/>							
12		•	Return									
13	-00	•	MO	TOR_S	PEEDC	ONTRO)L	Bool				
	<											>
8	3	> = 1	??	-	-01	↦	-[=]					
-	Blo	ck t	title:	Speed	contro	ol via a	nalog outp	ut				^
C	om	mer	nt									
												=
•	1	let	work 1	:								
	C	om	ment									
												~
										100	% -	

Note: Be sure to use the correct data types.

→ Insert an Assignment ' -[-] in the first network and an 'And' in front of it. Then use drag-and-drop to move the 'Comparator operation' 'Less or equal' from the 'Basic instructions' onto the first input of the AND logic operation.

(→	-{=] →	8	\rightarrow Basic	instructions \rightarrow	Comparator	$operations \rightarrow$	CMP<=)
----	--------	---	---------------------	----------------------------	------------	--------------------------	--------

					Options	
i 🗚 学 👻 🍬 🖿 🚍 🚍 💬 📲 🕿	• 🖃 🎲 🥐 😡 d	🗄 🐏 🤒 I ₌ N	= 🗳 🌹 🔢		Mi Mi 🗌	
MOTOR_SPEEDCONTROL					> Favorites	
Name	Data type	Default value	Comment		✓ Basic instructions	
🕣 🔻 Input				~	Name Ver	i
Setpoint_speed	Real			≡	🕨 🦳 General	-
💷 🔻 Output					Bit logic operations	
Manipulated_variable_speed_AO	Int				Timer operations	
💷 🔻 InOut				~	Figure 1 Counter operations	
<				>	 Comparator operations 	
					CMP ==	
					E CMP ↔	
Block title: Speed control via analog output				^	E CMP >=	
omment					E CMP <=	
				- 11	CMP >	
Network 1: check setpoint speed for correct	t input range +/- 50 r/n:	nın			CMP <	
Comment					IN_Range	
					OUT_Range	
&	?.?			=	🗉 - ОК -	
?.? -	=				🗉 – NOT_OK -	
?.? -12***					🕨 🛅 Variant	

→ Next use drag-and-drop to move the 'Comparator operation' 'Greater or equal' onto the second input of the AND logic operation.

 $(\rightarrow \text{Basic instructions} \rightarrow \text{Comparator operations} \rightarrow \text{CMP} >=)$

'U1516F [CPU 1516F-3 PN/DP] > Program I	olocks • MOTOR_	SPEEDCONTROI	L [FC10]	_ 7 =	×	Instructions	
						Options	
ത്ത് 🖻 👻 🎭 🖿 🚍 🚍 🗩 😫 🖢 😤	😑 🗊 🥙 💊 🖉	= 🐖 🤣 I = 🦄			4	tiù jiù	
MOTOR_SPEEDCONTROL						> Favorites	
Name	Data type	Default value	Comment		-	✓ Basic instructions	
1 📶 🔻 Input					^	Name	Versi
2 💷 = Setpoint_speed	Real				Ξ	General	•
3 🕣 🔻 Output						Bit logic operations	_
4 📹 🔹 Manipulated_variable_speed_AO	Int					Timer operations	
5 📶 🔻 InOut					¥	+1 Counter operations	
<				>		 Comparator operations 	
						CMP ==	=
 Block title: Speed control via analog output 					^	E CMP >=	
Comment						E CMP <=	
						CMP >	
Network 1: check setpoint speed for correct	t input range +/- 50 r/m	iin				CMP <	
Comment						IN_Range	
						OUT_Range	
<=					≡	🗉 - ОК -	
???						-INOT_OK -	
?? — IN1	&					Variant	
?? IN2	<	1.7>				Hath functions	
		=				Move operations	
?.? -•• *		—				Conversion operations	
5. H+1						Program control operati	~
						<	>

→ Connect the contacts in Network 1 with the constants and local tags as shown here. The data types in the comparator operations are automatically adapted to 'Real'.



→ Use drag-and-drop to move the 'Conversion operation' 'NORM_X' into Network 2 in order to normalize the speed setpoint of +/- 50 rpm to +/- 1.

 $(\rightarrow \text{Basic instructions} \rightarrow \text{Conversion operations} \rightarrow \text{NORM}_X)$



→ Connect the contacts in Network 2 with the constants and local tags as shown here. The data types in 'NORM_X' are automatically adapted to 'Real'.



→ Use drag-and-drop to move the 'Conversion operation' 'SCALE_X' into Network 3 in order to scale the speed setpoint from the normalized +/- 1 onto the range for the analog output +/- 27468.

U1516F [CPU 1516F-3 PN/DP] > Program	blocks • MOTOR	SPEEDCONTRO	L [FC10]	_ ∎ ≡ ×	(Ir	nstructions	. II	
					0	ptions		
ාස් ශ් 🖻 👻 🎭 🖿 🚍 🚍 💬 🗃 ± 🖴	± 🖃 🗊 🥲 💊 🖉	🗄 🐖 🤣 📭 🦄	- 8 🐨 B	2 📑		îrij iri		
MOTOR_SPEEDCONTROL					5	Favorites		_
Name	Data type	Default value	Comment			Rasic instructions	_	_
1 📶 🕶 Input				1		me	Verci	
2 💷 = Setpoint_speed	Real					General	versi	
3 📶 🔻 Output					Ľ	Bit logic operations		F
4 📹 🔹 Manipulated_variable_speed_AO	Int				Ľ	Timer operations		
5 🕣 🔻 InOut					Б	+1 Counter operations		
6 Add new>						Comparator operations		
<	11	+ Math functions		=				
					1,	Move operations		
& >=1 ??? ⊣ −ol ↦ -[=]					-	Conversion operations		
	_					CONVERT		
NORM_X Peal to Peal				*	ì			
Real to Real					ι.			
#Setpoint_					ι.	FLOOR		
speed_OK EN	#Manipulate	d_			ι.	TRUNC		
#Setpoint speedVALUS	variable_spe	ed_			E	SCALE_X		
#Setpoint_speed — VALUE						NORM_X		
50.0 - MAX	ENO					Legacy		
						Program control operati		
Network 3: Scale normalised setpoint speed for	or analog output to +/-	27648		-	•	Word logic operations		
Comment						😝 Shift and rotate		
connent					Þ	ETC Legacy		~
-				_	<		>	•
100 mg 100 mg					>	Extended instructions		

 $(\rightarrow \text{Basic instructions} \rightarrow \text{Conversion operations} \rightarrow \text{SCALE}_X)$

→ Connect the contacts with the constants and local tags in Network 3 as well, as shown here. The data types in 'SCALE_X' are automatically changed to 'Real' or 'Int'.



→ Insert an Assignment ' f=1 ' in the fourth network. Use drag-and-drop to move the 'Move' command from the 'Move operations' folder under 'Basic instructions' in front of the Assignment.

'U1516F [CPU 1516F-3 PN/DP] • Program	blocks MOTOR_	SPEEDCONTRO	L [FC10]		×	Instructions		
					- [Options		
ുൽ 🖉 👻 🎭 🖿 🚍 🗁 🚝 ± 🚇	• 🖃 🗊 🥙 😡 🖉	l 🕫 🤣 🖣 🦹	- e 🚏			nit ni		
MOTOR_SPEEDCONTROL						> Favorites		
Name	Data type	Default value	Comment			 Basic instructions 		
1 🕣 🔻 Input					^	Vame	Versi	
2 < Setpoint_speed	Real				Ξ	General		~
3 🕣 🔻 Output						Bit logic operations		F
4 💶 🔹 Manipulated_variable_speed_AO	Int 🔳]				Timer operations		
5 📶 🔻 InOut					*	Lounter operations		
<				>		Comparator operations		
					-	Math functions		=
& >=1 [??] → -0 → -[=]						🕶 🔁 Move operations		
#Manipulated	#Manipul	ated				MOVE		
variable_speed_	variable_	speed_				🗉 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>	
- Notarial A. Consideration and former of	FO desire a Maniaulatar	a contrata a contra	10 0/0-0-			UMOVE_BLK		
 Network 4: Speed setpoint out of range +i- 	50 rimin -> Manipulate	d_variable_speed	$_AO = 07$ ketur	n = IKUE		FILL_BLK		
Comment						UFILL_BLK		
						SWAP		
						🕨 🛅 Array DB		
					\equiv	🕨 🛅 Variant		

 $(\rightarrow || f^{-1} || \rightarrow \text{Basic instructions} \rightarrow \text{Move operations} \rightarrow \text{MOVE})$

→ The contacts in Network 4 will now be connected with constants and local tags as shown here. If the speed setpoint is not within the range +/- 50 rpm, the value '0' is output at the analog output and the value TRUE is assigned to the return value (Return) of the "MOTOR_SPEEDCONTROL" function.



→ Do not forget to click Save project. The finished function "MOTOR_SPEEDCONTROL" [FC10] in FBD is shown below.



For unrestricted use in educational and R&D institutions. © Siemens AG 2017. All rights reserved. SCE_EN_032-500 Analog Values_S7-1500_R1703.docx

7.3 Configuration of the analog output channel

 \rightarrow Double-click the 'Device configuration' to open it.



 \rightarrow Check the address setting and the configuration of the analog output channel 0.

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



7.4 Expand the tag table to include analog signals

 \rightarrow Double-click the 'Tag table_sorting station' to open it.



→ Add the global tags for the analog value processing to the "Tag table_sorting station". An analog input B8 and an analog output U1 must be added.

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

032	-500_	Analog_Values → CPU1516	F [CPU 1516F-	3 PN/DP] 🕨 PLC	tags 🕨	Tag tab	le_sortiı	ng station [30] 📃 🗖 🖬 🗙
								🕢 Tags 🔳 User constants
	÷ -	xii 🕾 🕂						
Т	ag tal	ble_sorting station						
	N	ame	Data type	Address	Retain	Visibl	Acces	Comment
15	-00	-55	Bool	%I1.6				pushbutton manual mode cylinder -M4 retract (no) 🔺
16	-	-56	Bool	%11.7				pushbutton manual mode cylinder -M4 extend (no)
17	-	-Q1	Bool	%Q0.0				conveyor motor -M1 forwards fixed speed
18	-	-Q2	Bool	%Q0.1				conveyor motor -M1 backwards fixed speed
19		-Q3	Bool	%Q0.2		Image: A start and a start		conveyor motor -M1 variable speed
20		-M2	Bool	%Q0.3				cylinder -M4 retract
21		-M3	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 activated"
26	-	-P5	Bool	%Q1.1				display "automatic mode started"
27		-P6	Bool	%Q1.2		Image: A start and a start		display cylinder -M4 "retracted" 📃
28		-P7	Bool	%Q1.3				display cylinder -M4 "extended"
29		-U1	Int	%QW64				manipulated value speed in 2 directions +/- 10V
30		-B8	Int	%IW64				sensor actual value speed 010V
31		<add new=""></add>				V	V	

7.5 Call the block in the organization block

 \rightarrow Open the "Main [OB1]" organization block with a double-click.



→ Add the temporary tag 'Motor_speed_monitoring_Ret_Val' to the local tags of OB1. These will be needed in order to interconnect the return value of the "MOTOR SPEEDCONTROL" function.

 $(\rightarrow \text{Temp} \rightarrow \text{Motor_speed_monitoring_Ret_Val} \rightarrow \text{Bool})$

03	2-5(00_	_Analog_Values	6F-3 PN/DP] 🕨 Pro	gram blocks 🔸	Main [OB1] 📃 🖬 🖬	×
ю́	В	ŝ) 🔮 🐁 🖿 🚍 🚍 💬 📲 ± 🖴 🖿	🎲 🥙 💊 🖑 🗺	🥵 I 🔤 🖁		-
	Ma	in					
		Na	me	Data type	Default value	Comment	
1		•	Input				
2		•	Initial_Call	Bool		Initial call of this OB	
З		•	Remanence	Bool		=True, if remanent data are available	
4		٠	Temp				
5	-00	•	Motor_speed_monitoring_Ret_Val	Bool 🔳			
6		•	<add new=""></add>				
7		٠	Constant				
8		•	<add new=""></add>				

→ Select the block title of OB1 and then click '¹¹/¹/¹ to insert a new Network 1 in front of the other networks

(→ 🕅)

03	2-5	00	_Analog_Values → CPU1516F [CPU 151	6F-3 PN/DP] > Pro	gram blocks 🔸	Main (OB1) 📃 🖬	∎×
H	ю	K I	🖗 🔮 🍬 🖿 🚍 🚍 💬 📲 ± 🖀 ± 🚍	😥 🥙 💊 🖉	l \$\$ I _≡ 3 _≡ d	e 😨 🔢	
V	⁵ Ma	in					
U	nser	t ne	twork	Data type	Default value	Comment	
1	-00	•	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			
6		•	<add new=""></add>				
7		•	Constant				
8		•	<add new=""></add>				
	<						>
•	Blo	ck	title: "Main Program Sweep (Cycle)"				^
	lom	me	nt				
•	1	Net	twork 1: Control conveyor motor forwards in a	utomatic mode			=
	C	om	iment				
							_

→ Use drag-and-drop to move your "MOTOR_SPEEDCONTROL [FC10]" function onto the green line in Network 1.

Vi	Siemens - G:\Automation\032-500_Analo	g_Valu	es\0)32-	500_Analog_Valu	les				_ 0	×
P	roject Edit View Insert Online Optior 🛉 🎦 🔒 Save project 블 💥 🗐 🗎 🗙	ns To	ols (~ 1	Wii Ł Ę	ndow Help 🗟 🔃 🖸 🖳 🛤	💋 Go online 📓 Go offlir	ne 🚮 🖪 🖪 🗡	=	Totally Integrated Au	itomation PORTAL	
	Project tree		0	82-5	00_Analog_Val	ues 🕨 CPU1516F [CPU 1	1516F-3 PN/DP] >	Program blocks	Main [OB1]	_ = = ×	•
	Devices		Γ								
	1×00	1 과	J.Å	8.5	x 🗈 🛋 🖦 =	= 🚍 🚍 🖂 🥞 🕮 + 🙉 +	三翁 @ 6. 8	QH 405 1_ 1_	8 99 III		S.
-											Ē
÷	▼ □ 032-500 Analog Values	•		IVIC	Name		Data type	Default value	Comment		8
Ē	Add new device		1	-111			boto type	Derbart volue	comment		2
16	Bevices & networks		2	-	Initial Call		Bool		Initial call of this OB		
l E	▼ CPU1516F [CPU 1516F-3 PN/DP]		3	-	 Remanence 	e	Bool		=True, if remanent data	a are available	V.
Ľ	Device configuration		4		 Temp 						1
-	😧 Online & diagnostics		5		Motor_spectrum	ed_monitoring_Ret_Val	Bool				sti
	▼ 🕞 Program blocks		6		Add new>						<u>Bu</u>
	Add new block	≡	7		 Constant 						
	Hain [OB1]		8		Add new>						Ð
	MOTOR_SPEEDCONTROL [FC10]			<						>	Ta
	MOTOR_AUTO [FB1]		E	DL		control (Control)	▲ I ▼ I				sks
	MOTOR_AUTO_DB [DB1]		Ľ	BIC	OCK TITLE: Main Pr	ogram Sweep (Cycle)				<u> </u>	
	Technology objects			Com	iment						
	External source files		-		Network 1:						E
	🕨 🕨 🔁 PLC tags		E	7	Commont					=	ra
	PLC data types				commente						ies
	Watch and force tables			-	- MC	TOR_SPEEDCONTROL [FC10]					
	Online backups					2					
	🕨 🔁 Traces		I								

→ Connect the contacts with the constants and global and local tags here as shown.



→ Change the connection of output tag "Conveyor_motor_automatic_mode" in Network 2 to '-Q3' (Conveyor motor -M1 variable speed) so that the conveyor motor is controlled taking the analog speed setting into consideration.





7.6 Save and compile the program

→ To save your project, select the **Save project** button in the menu. To compile all blocks,

click the "Program blocks" folder and select the 🛅 icon for compiling in the menu.

 $(\rightarrow \square$ Save project \rightarrow Program blocks $\rightarrow \square$)

٧ß	Siemens - G:\Automation\032-500_Analog_Valu	Jes\032-500_An	alog_Values _ C	×
Pr	oject Edit View Insert Online Options To 🛉 📑 🗔 Save project 🚇 🐰 🏥 🗊 🗶 🏹 🖆	ools Window (~ ± 🔄 🔃 !	Help Totally Integrated Automation PORTAL PORTAL	
	Project tree 🔲 🖣	032-500 An	alog_Values → CPU1516F [CPU 1516F-3 PN/DP] → Program blocks → Main [OB1] _ ■ ■ ■ ×	
	Devices	Compile		
		ы 🖄 🚔 🛢	🖗 🏍 🖿 🖻 🗩 📲 ± 🗏 🐲 🥙 🐜 🖑 😘 🖑 🖫 🚱 🙄 🔢 👘 📮	nst
2			Block interface	5
ogrammir	O32-500_Analog_Values Add new device Devices & networks	Block title	: "Main Program Sweep (Cycle)"	tions
L T	CPU1516F [CPU 1516F-3 PN/DP]	 Network 	k 1: Speed control analog oputput conveyor motor	Ų,
brc ¹	Construction (Construction) Device configuration Online & diagnostics Program blocks Add new block Main (D81) MOTOR_AUTO (F61) MOTOR_AUTO (F61) MOTOR_AUTO (F61) MOTOR_AUTO (F61) External source files Device External	Commer	t t t t *#C10 *#Motor_speed_ monitoring_Ret_ Ret_Val - Val Manipulated_ valabe_ *Val Speed_AO - 'U1* 15.0 - Setpoint_speed ENO -	🛛 Testing 🛛 🔝 Tasks 🛛 🔁 Librario
	Watch and force tables	▼ Networ	k 2: Control conveyor motor forwards in automatic mode	Si
	Online backups	Commer	r	
	< m >	<	Ⅲ > 100% ▼	
	> Details view		🔟 Properties 🚺 Info 🚯 🗓 Diagnostics 💿 🗖 🗛	
	🖣 Portal view 🔠 Overview 🔹	Main	✓ The project 032-500_Analog_Values w	

 $\rightarrow~$ The "Info", "Compile" area shows which blocks were successfully compiled.

	Roperties	🗓 Info	<u>i</u> 🖁 🕻)iagnost	ics	
General (i) Cross-references	Compile Syntax					
😧 🛕 🚺 Show all messages	•					
Compiling completed (errors: 0; warning	igs: 0)					
! Path	Description	Go to	?	Errors	Warnings	Time
✓ ▼ CPU1516F		- X		0	0	12:2
Program blocks		- X		0	0	12:2
MOTOR_SPEEDCONTRO	Block was successfully compiled.	×				12:2
S Main (OB1)	Block was successfully compiled.	×				12:2
S	Compiling completed (errors: 0; warnings: 0)					12:2
<	111					>

7.7 Download the program

→ After successful compilation, the complete controller with the created program including the hardware configuration can, as described in the previous modules, be downloaded.



概	Siemens - G:\Automation\032-500_Analog_\	/alue	es\03	2-500_Analog_Values	_ ¤ ×
Pro	ject Edit View Insert Online Options	Тос	ls I	Window Help Totally Integrated Automa	ation
2	🖥 📑 Save project 📑 🐰 🛅 🗂 🗙 🍤	± (24 ±	🖥 🖳 🗓 🖳 🧖 Go online 🖉 Go offline 🛔 🖪 🖪 👫 🗶 🖃 🛄 🕺 🦻	ORTAL
	Project tree		032	2-500 Analog Values → CPU1516F [CPU 1516F-3 PN/DP] → Program blocks → Main [OB1]	∎≡× ◀
	Devices			Download to device	- 8
	B O O E	1	кőі	K # # = = = = = = = = = = = = = = = = =	
2				Block interface	E C
1 B	032-500_Analog_Values	^			
E I	Add new device		8	>=1 [??] ⊣ −•Ⅰ ↦ -[=]	12
- B	h Devices & networks		-	Nock title: "Main Program Sween (Orcle)"	
E.	CPU1516F [CPU 1516F-3 PN/DP]		•		<u>v</u>
121	Device configuration	≡		annene (m. 1997).	e
-	Q Online & diagnostics		-	Network 1: Speed control analog oputput conveyor motor	= ti
	🔻 🔜 Program blocks			Comment	<u>D</u>
	📑 Add new block		-		
	Hain [OB1]			%FC10	- D
	MOTOR_SPEEDCONTROL [FC10]			"MOTOR_SPEEDCONTROL"	a la
	MOTOR_AUTO [FB1]				sks
	MOTOR_AUTO_DB [DB1]			#Motor_speed_ monitoring_Pet	
	Technology objects			Bet Val Val	
	External source files			Manipulated_	5
	PLC tags			variable_ %QW64	ora
	PLC data types			Speed_AO - OI	Tie
	Watch and force tables			13.0 Setpoint_speed ENO	
	🕨 🙀 Online backups				~
r i	< <u> </u>		<	III > 100%	
	> Details view			🔍 Properties 🔰 🗓 Info 👔 🗓 Diagnostics	
	Portal view Overview		//ain	📲 MOTOR_SPE 💙 The project 032-500_Analog_Values w	

7.8 Monitor program blocks

 \rightarrow The desired block must be open for monitoring the downloaded program. The monitoring

can now be activated/deactivated by clicking the



→ The "MOTOR_SPEEDCONTROL" [FC10] function called in the "Main [OB1]" organization block can be selected directly for "Open and monitor" after right-clicking and the program code in the function can thus be monitored.

🔥 🐼 🖈 👘 🖿 🚍 🚍 🖓 😫 😫 😫 😫 😫 😫 😫 😫 😫 😫 😫 😭 🔛 ?? -ol 🕁 -[=] -8 >=1 ^ Network 1: Speed control analog oputput conveyor motor Comment FC10 Modify "MOTOR_SPEEDCONTRC . Open Open and Define tag... Ctrl+Shift+I Ctrl+Shift+T Rename tag.. Rewire tag... Ctrl+Shift+P Manip Ctrl+X X Cut EN spe Ctrl+C Copy 15.0 -Setpoint_speed Ŵ Paste Ctrl+V X Delete Del _ 🖬 🖬 🗙 💑 🐼 学 🐏 🖿 🚍 🖃 🗶 😫 🖿 🔛 😫 🛏 🔛 😢 😘 🥙 🐜 🖉 🖳 🔛 ۱. Call path: Main [OB1] >=1 ?? -oI → -[=] 2 - H





7.9 Archive the project

→ As the final step, we want to archive the complete project. Select the → 'Archive ...' command in the → 'Project' menu. Select a folder where you want to archive your project and save it with the file type "TIA Portal project archive".

 $(\rightarrow$ Project \rightarrow Archive \rightarrow TIA Portal project archive \rightarrow 032-500_Analog_Values.... \rightarrow Save)

Ϋ́́́	Siemens - G:\Automation	n\032-500_Analog	_Valu	es\032-500_Ar	ialog_Values						_ L	X
Pr	oject Edit View Insert New	Online Options	; Tor)±(ols Window	Help 🚹 🖳 📮 💋 Go (online 🔊 Go of	fline 🗛 🚺	To	tally Integrate	d Automatio POF	on RTAL	
	Open Migrate project	Ctrl+O		nalog_Va	ues 🕨 CPU1516F	[CPU 1516F-3	PN/DP] ▶	Program block	s ▶ Main [OB	1] 💷 🖬	∎×	•
	Close	Ctrl+W										
	Save Save as	Ctrl+S Ctrl+Shift+S] 🖻	iể Ki 🖗	# 🖦 🖿 🗖 🗖	🗩 岩 ± 🔏 :	🗄 📰 🔝 🔮	° ,₀ (≣ ⊊ ≣:	₿ I = <i>x</i> = 6	× 💦 🖪		Instruc
	Delete project Archive Retrieve	Ctrl+E	^	& >=1	??}⊣-∘≀ ↦	-[=]						tions
-	Card Reader/USB memory	•	-	 Netwo 	rk 1: Speed control	analog oputput o	conveyor moto	or			^	Q .
	Memory card file	•	=	Comme	nt							Fe
	Upgrade Print	Ctrl+P			'MO	%FC10 TOR_SPEEDCONT	TROL"				=	sting
8	Print preview							#Motor speed				D
	Export module labeling strip)S					Ret Val	monitoring_Ret_ /al				Tas
	G:\Automation\03\032-500	_Analog_Values				Man	ipulated_					ks
	G:\Automation\\032_300	_Analog_Values			EN.	s	peed AO	%QW64 '-U1"				
	G:\Automation\\032_200_F	B-Programming			15.0 Setpoint	speed	ENO -					
	G:\Automation\\032_200_F	B-Programming										bra
	Exit			▼ Notwo	rk 2: Control conve	or motor forward	ls in automati	r mode			- 11	ries
	Vigi watch and lord	e tables		- Netwo	KZ. Control convey	or motor lorward	is in automati	c mode				
	 Image: Online backups Image: Online backups 	5	~	Comme				100%			~	
	< III		>	<			D	100%	(m)			
	> Details view		_			-	Properties	L Info	🛛 🔁 Diagnosti	ics		
	Portal view	🔛 Overview	-	Main				Connection	to CPU1516F ter	minated.		

8 Checklist

No.	Description	Completed
1	Compiling successful and without error message	
2	Download successful and without error message	
3	Switch on station (-K0 = 1) Cylinder retracted / Feedback activated (-B1 = 1) EMERGENCY OFF (-A1 = 1) not activated AUTOMATIC mode (-S0 = 1) Pushbutton automatic stop not actuated (-S2 = 1) Briefly press the automatic start pushbutton (-S1 = 1) Sensor part at slide activated (-B4 = 1) then Conveyor motor M1 variable speed (-Q3 = 1) switches on and stays on. The speed corresponds to the speed setpoint in the range +/- 50 rpm	
4	Sensor part at end of conveyor activated (-B7 = 1) \rightarrow -Q3 = 0 (after 2 seconds)	
5	Briefly press the automatic stop pushbutton (-S2 = 0) \rightarrow -Q3 = 0	
6	Activate EMERGENCY OFF (-A1 = 0) \rightarrow -Q3 = 0	
7	Manual mode (-S0 = 0) \rightarrow -Q3 = 0	
8	Switch off station (-K0 = 0) \rightarrow -Q3 = 0	
9	Cylinder not retracted (-B1 = 0) \rightarrow -Q3 = 0	
10	Project successfully archived	

9 Exercise

9.1 Task – Exercise

In this exercise a "MOTOR_SPEEDMONITORING" [FC11] function will be created additionally.

The actual value will be made available to -B8 (sensor actual value speed of the motor +/-10V corresponds to +/- 50 rpm) as an analog value and queried at an input of the "MOTOR_SPEEDMONITORING" [FC11] function. The data type is 16-bit integer (Int.).

This actual speed value will first be normalized to the range +/- 1 as 32-bit floating-point number (Real) in the function.

The normalized actual speed value will then be scaled to revolutions per minute (range: +/- 50 rpm) as 32-bit floating-point number (Real) and made available at an output.

The following 4 limit values can be specified as 32-bit floating-point numbers (Real) at the block inputs in order to monitor them in the function:

Speed > Motor_speed_monitoring_error_max

Speed > Motor_speed_monitoring_warning_max

Speed < Motor_speed_monitoring_warning_min

Speed < Motor_speed_monitoring_error_min

If a limit value is exceeded or fallen below, the value TRUE (1) is assigned to the corresponding output bit.

If a fault is present, the protective tripping of the "MOTOR_AUTO" [FB1] function block will be tripped.

9.2 Technology diagram

Here you see the technology diagram for the task.



Figure 3: Technology diagram

Schalter der Sortieranlage		Automatikbetrieb	Handbetrieb / Manual mode
Switches of sorting station		Automatic mode	-S3 Tippbetrieb -M1 vorwärts/ Manual -M1 forwards
-P1 ein/on		-P5 gestartet/started	-S4 Tipphatriah -M1 rückwärts/
			Manual -M1 backwards
-P4 autoenzeuve		-S2 Stopp/stop	-P7 ausgefahren/extended
-P2 Hand/manual -P3 Auto/auto			-S6 Zylinder -M4 ausfahren/ cylinder -M4 extend
-S0 Betriebsart/operating mode			-S5 Zylinder -M4 einfahren/
			- cylinder -M4 retract

Figure 4: Control panel

9.3 Reference list

DI	Туре	Identifier	Function	NC/NO
I 0.0	BOOL	-A1	Return signal emergency stop OK	NC
I 0.1	BOOL	-K0	Main switch "ON"	NO
I 0.2	BOOL	-S0	Mode selector manual (0)/ automatic (1)	Manual = 0 Auto = 1
I 0.3	BOOL	-S1	Pushbutton automatic start	NO
I 0.4	BOOL	-S2	Pushbutton automatic stop	NC
I 0.5	BOOL	-B1	Sensor cylinder -M4 retracted	NO
I 1.0	BOOL	-B4	Sensor part at slide	NO
I 1.3	BOOL	-B7	Sensor part at end of conveyor	NO
IW64	BOOL	-B8	Sensor actual value speed of the motor +/-10V corresponds to +/- 50 rpm	

The following signals are required as global operands for this task.

DO	Туре	Identifier	Function	
Q 0.2	BOOL	-Q3	Conveyor motor -M1 variable speed	
QW 64	BOOL	-U1	Manipulated value speed of the motor in 2 directions +/- 10V corresponds to +/- 50 rpm	

Legend for reference list

- DI Digital Input DO Digital Output
- AI Analog Input AO Analog Output
- I Input Q Output
- NC Normally Closed
- NO Normally Open

9.4 Planning

Plan the implementation of the task on your own.

9.5 Checklist – Exercise

No.	Description	Completed
1	Compiling successful and without error message	
2	Download successful and without error message	
3	Switch on station (-K0 = 1) Cylinder retracted / Feedback activated (-B1 = 1) EMERGENCY OFF (-A1 = 1) not activated AUTOMATIC mode (-S0 = 1) Pushbutton automatic stop not actuated (-S2 = 1) Briefly press the automatic start pushbutton (-S1 = 1) Sensor part at slide activated (-B4 = 1) then Conveyor motor -M1 variable speed (-Q3 = 1) switches on and stays on. The speed corresponds to the speed setpoint in the range +/- 50 rpm	
4	Sensor part at end of conveyor activated (-B7 = 1) \rightarrow -Q3 = 0 (after 2 seconds)	
5	Briefly press the automatic stop pushbutton (-S2 = 0) \rightarrow -Q3 = 0	
6	Activate EMERGENCY OFF (-A1 = 0) \rightarrow -Q3 = 0	
7	Manual mode $(-S0 = 0) \rightarrow -Q3 = 0$	
8	Switch off station (-K0 = 0) \rightarrow -Q3 = 0	
9	Cylinder not retracted (-B1 = 0) \rightarrow -Q3 = 0	
10	Speed > Motor_speed_monitoring_error_max \rightarrow -Q3 = 0	
11	Speed < Motor_speed_monitoring_error_min \rightarrow -Q3 = 0	
12	Project successfully archived	

10 Additional information

You can find additional information as an orientation aid for initial and advanced training, for example: Getting Started, videos, tutorials, apps, manuals, programming guidelines and trial software/firmware, at the following link:

www.siemens.com/sce/s7-1500