SCE Training Curriculum

Siemens Automation Cooperates with Education | 05/2017

TIA Portal Module 032-600
Global Data Blocks
for the SIMATIC S7-1500
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: [siemens.com/sce/tp](http://siemens.com/sce/tp)

**Continued training**
For regional Siemens SCE continued training, please contact your regional SCE contact [siemens.com/sce/contact](http://siemens.com/sce/contact)

**Additional information regarding SCE**
[siemens.com/sce](http://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 Urban, the Michael Dziallas Engineering Corporation and all other involved persons for their support during the preparation of this training curriculum.
Table of contents

1 Goal ............................................................................................................................................... 5
2 Prerequisite ................................................................................................................................ 5
3 Required hardware and software ................................................................................................. 6
4 Theory ........................................................................................................................................... 7
  4.1 Data blocks ............................................................................................................................... 7
  4.2 Data types of the SIMATIC S7-1500 ....................................................................................... 8
  4.3 Optimized blocks ...................................................................................................................... 9
  4.4 Download without reinitialization ............................................................................................ 9
5 Task ............................................................................................................................................... 10
6 Planning ....................................................................................................................................... 10
  6.1 Global data block for speed control and speed monitoring of the motor ............................... 10
  6.2 Technology diagram ............................................................................................................... 11
  6.3 Reference list .......................................................................................................................... 12
7 Structured step-by-step instructions ............................................................................................ 13
  7.1 Retrieve an existing project ...................................................................................................... 13
  7.2 Create the global data block “SPEED_MOTOR” ..................................................................... 15
  7.3 Access to data of the data block in the organization block .................................................... 20
  7.4 Save and compile the program ................................................................................................ 24
  7.5 Download the program .......................................................................................................... 25
  7.6 Monitor/modify values in data blocks ..................................................................................... 26
  7.7 Initialize setpoints/reset start values ..................................................................................... 27
  7.8 Snapshots in data blocks ........................................................................................................ 29
  7.9 Expand data block and download it without reinitialization .................................................. 33
  7.10 Archive the project ................................................................................................................ 37
8 Checklist ....................................................................................................................................... 38
9 Exercise ......................................................................................................................................... 39
  9.1 Task – Exercise ....................................................................................................................... 39
  9.2 Technology diagram ............................................................................................................... 39
  9.3 Reference list .......................................................................................................................... 40
  9.4 Planning ................................................................................................................................... 40
  9.5 Checklist – Exercise ............................................................................................................... 41
10 Additional information .................................................................................................................. 42
GLOBAL DATA BLOCKS FOR THE SIMATIC S7-1500

1 Goal

In this chapter, you will become acquainted with the use of global data blocks for the SIMATIC S7-1500 with the TIA Portal programming tool.

The module explains the structure and creation of and access to global data blocks for the SIMATIC S7-1500. It also shows the steps for creating a global data block in the TIA Portal and for accessing this data in the program with read and write access.

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

2 Prerequisite

This chapter builds on the chapter Analog Values with the SIMATIC S7 CPU1516F-3 PN/DP. You can use the following project for this chapter, for example: “SCE_EN_032-500_Analog_Values_R1508.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

3 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 Data blocks

In contrast to logic blocks, data blocks contain no instructions. Rather, they serve as memory for user data.

Data blocks thus contain variable data that is used by the user program. You can define the structure of global data blocks as required.

Global data blocks store data that can be used by all other blocks (see Figure 1). Only the associated function block should access instance data blocks. The maximum size of data blocks varies depending on the utilized CPU.

![Figure 1: Difference between global DB and instance DB.](image)

Access for all blocks

Access only for Function_data_block_12
Application examples for **global data blocks** are:
- Saving of information about a storage system. "Which product is located where?"
- Saving of recipes for particular products.

The data in data blocks is stored retentively in most cases. This data is then retained in the event of a power failure or after a STOP/START of the CPU.

### 4.2 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.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Size (bits)</th>
<th>Range</th>
<th>Example of constant entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bool</td>
<td>1</td>
<td>0 to 1</td>
<td>TRUE, FALSE, O, 1</td>
</tr>
<tr>
<td>Byte</td>
<td>8</td>
<td>16#00 to 16#FF</td>
<td>16#12, 16#AB</td>
</tr>
<tr>
<td>Word</td>
<td>16</td>
<td>16#0000 to 16#FFFF</td>
<td>16#ABCD, 16#0001</td>
</tr>
<tr>
<td>DWord</td>
<td>32</td>
<td>16#00000000 to 16#FFFFFFFF</td>
<td>16#02468ACE</td>
</tr>
<tr>
<td>Char</td>
<td>8</td>
<td>16#00 to 16#FF</td>
<td>'A', 'r', '@'</td>
</tr>
<tr>
<td>Sint</td>
<td>8</td>
<td>-128 to 127</td>
<td>123,-123</td>
</tr>
<tr>
<td>Int</td>
<td>16</td>
<td>-32,768 to 32,767</td>
<td>123,-123</td>
</tr>
<tr>
<td>Dint</td>
<td>32</td>
<td>-2,147,483,648 to 2,147,483,647</td>
<td>123,-123</td>
</tr>
<tr>
<td>USInt</td>
<td>8</td>
<td>0 to 255</td>
<td>123</td>
</tr>
<tr>
<td>Ulnt</td>
<td>16</td>
<td>0 to 65,535</td>
<td>123</td>
</tr>
<tr>
<td>UDInt</td>
<td>32</td>
<td>0 to 4,294,967,295</td>
<td>123</td>
</tr>
<tr>
<td>Real</td>
<td>32</td>
<td>+/-1.18 x 10^-38 to +/-3.40 x 10^38</td>
<td>123.456, -3.4, 1.2E+12 3.4E-3</td>
</tr>
<tr>
<td>LReal</td>
<td>64</td>
<td>+/-2.23 x 10^-308 to +/-1.79 x 10^308</td>
<td>12345.123456789 -1.2E+40</td>
</tr>
<tr>
<td>Time</td>
<td>32</td>
<td>T#-24d_20h_31m_23s_648ms to T#24d_20h_31m_23s_647ms Saved as: -2,147,483,648 ms to +2,147,483,647 ms</td>
<td>T#5m_30s 5#2d T#1d_2h_15m_30x_45ms</td>
</tr>
<tr>
<td>String</td>
<td>Variable</td>
<td>0 to 254 characters in byte size</td>
<td>'ABC'</td>
</tr>
</tbody>
</table>

Array

- With arrays, data of a uniform data type is arranged one after the other and addressed consecutively in the address area. The properties of each array element are identical and are configured in the array tag.

Struct

- The STRUCT data type represents a data structure that consists of a fixed number of components of different data types. Components of STRUCT or ARRAY data type can also be nested in a structure.

... For other data types, refer to the online help.
4.3 Optimized blocks

S7-1500 controllers have optimized data storage. In optimized blocks all tags are automatically sorted based on their data type. The sorting ensures that data gaps between the tags are minimized and the tags are stored in a manner that optimizes their access by the controller.

- The tags are always accessed as fast as possible because the file storage by the system is optimized and is independent of the declaration.
- There is no danger of inconsistencies due to incorrect, absolute accesses because symbolic access is generally used.
- Declaration changes do not result in access errors because accesses by process visualization systems, for example, occur symbolically.
- Individual tags can be selectively defined as retentive.
- No settings are needed or possible in the instance data block. Everything will be set in the assigned FB (e.g., retentivity).
- Memory reserves in the data block enable changes to be made without loss of actual values (download without reinitialization).

4.4 Download without reinitialization

To enable the subsequent editing of user programs that are already running in a CPU, the S7-1500 controllers support the option of expanding the interfaces of optimized function or data blocks during operation. You can download the modified blocks without switching the controller to STOP mode and without affecting the actual values of previously downloaded tags.

The following steps can be performed while the controller is in RUN mode:

1. Activate "Download without reinitialization"
2. Insert newly defined tags in an existing block
3. Download expanded block to the controller

The newly defined tags are initialized. The existing tags retain their current value.

Prerequisite: a memory reserve must have been defined for the block beforehand and the block with this memory reserve must have downloaded to the CPU.
5 Task

In this chapter, the program from chapter "SCE_EN_032-500 Analog Values" will be expanded to include a data block that centrally provides the parameters for the two functions "MOTOR_SPEEDCONTROL" [FC10] and "MOTOR_SPEEDMONITORING" [FC11].

6 Planning

The data management and setpoint setting for the "MOTOR_SPEEDCONTROL" [FC10] and "MOTOR_SPEEDMONITORING" [FC11] functions will be carried out using the global data block "SPEED_MOTOR" [DB2].

This will be added to the "032-500_Analog_Values" project. This project must be retrieved from the archive beforehand.

In the "Main" [OB1] organization block, the two functions "MOTOR_SPEEDCONTROL" [FC10] and "MOTOR_SPEEDMONITORING" [FC11] must then be connected with the tags from global data block "SPEED_MOTOR" [DB2].

6.1 Global data block for speed control and speed monitoring of the motor

Speed setpoint and actual speed value will be created in Real data format (32-bit floating-point number) as the first tags in the "SPEED_MOTOR" [DB2] data block. The speed setpoint is thereby given the start value + 14 U/min

A structure (Struct) 'Positive_Speed' will then be created for monitoring the positive speed limits.

This structure contains the 2 tags 'Threshold_Error' (start value + 15 rpm) and 'Threshold_Warning' (start value + 10 rpm) in Real data format (32-bit floating-point number) and the 2 tags 'Error' and 'Warning' in Bool data format (binary number).

The structure (Struct) 'Positive_Speed' will then be inserted again as a copy and renamed to 'Negative_Speed' for monitoring the negative speed limits.

The 'Threshold_Error' tag is given the start value - 16 rpm and the 'Threshold_Warning' tag the start value - 14 rpm.
6.2 Technology diagram

Here you see the technology diagram for the task.

Figure 3: Technology diagram

Figure 4: Control panel
### 6.3 Reference list

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

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

<table>
<thead>
<tr>
<th>DO</th>
<th>Type</th>
<th>Identifier</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q 0.2</td>
<td>BOOL</td>
<td>-Q3</td>
<td>Conveyor motor -M1 variable speed</td>
</tr>
<tr>
<td>QW 64</td>
<td>BOOL</td>
<td>-U1</td>
<td>Manipulated value speed of the motor in 2 directions +/- 10V corresponds to +/- 50 rpm</td>
</tr>
</tbody>
</table>

**Legend for reference list**

- **DI** | Digital Input
- **AI** | Analog Input
- **I**  | Input
- **NC** | Normally Closed
- **NO** | Normally Open
- **DO** | Digital Output
- **AO** | Analog Output
- **Q**  | Output
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_032-500_Analog_Values_R1508.zap13" project from chapter "SCE_EN_032-500 Analog Values", 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.

(→ Project → Retrieve → Select a .zap archive → Open)

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

(→ Target directory → OK)
→ Save the opened project under the name 032-600_Global_Data_Blocks.

(→ Project → Save as … → 032-600_Global_Data_Blocks → Save)
7.2 Create the global data block "SPEED_MOTOR"

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

(→ CPU_1516F [CPU 1516F-3 PN/DP] → Add new block)

→ Select in the next dialog and rename your new block to: "SPEED_MOTOR". Select 'Global DB' as the type. The number '2' will be automatically assigned. Select the "Add new and open" check box. Click "OK".

(→ Name: SPEED_MOTOR → Type: Global DB → Add new and open → OK)
The “SPEED_MOTOR” data block is automatically displayed. Start by creating the ‘Speed_Setpoint’ and ‘Speed_Actual_Value’ tags shown here with their associated comments. Select ‘Real’ as the data type. Also set a start value of 10.0 rpm for the ‘Speed_Setpoint’.

(→ Speed_Setpoint → Real → 14.0 → Speed_Actual_Value → Real)

**Note:** Be sure to use the correct data types.

Next we create a tag structure ‘Struct’ so it can be duplicated later.

(→ Struct)
→ Name the structure 'Positive_Speed' and enter a comment.

(→ Positive_Speed)

Create the tags for the speed monitoring below the structure as shown here.

Note: Be sure to use the correct data types.
→ Then select the structure and copy it.

(→ Copy)

→ Paste the copied structure below the 'Positive_Speed' structure again.

(→ Paste)
 Rename the new structure to 'Negative_Speed' and enter a comment.

\(\rightarrow\) Negative_Speed

Do not forget to click \(\text{Save project}\). The finished global data block "SPEED_MOTOR" [DB2] is shown below. Check to verify that Retain is selected and the corresponding start value is entered for all tags. The data will thus be retained in the data block even after a power failure or a STOP/START of the CPU. The check boxes for 'Accessible from HMI' and 'Visible in HMI' should also all have a check mark so that all tags in future expansions of this project will be accessible by the visualization systems (HMI). We will select the 'Setpoint' check box only for the default values in our data block.

\(\rightarrow\)

\textbf{Note:} The use of setpoints is described further below in the step-by-step instructions of the module.
7.3 Access to data of the data block in the organization block

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

→ Delete the temporary tags in Main [OB1] that are no longer needed. Only the Boolean tag 'Motor_Speed_Control_Ret_Val' is still needed.

(→ Delete)
→ Have the "SPEED_MOTOR" [DB2] data block and the "Main" [OB1] organization block displayed side by side by clicking the '-split' icon to vertically split the editor area.

→ Use drag-and-drop to move the tags needed for the interconnection from the "SPEED_MOTOR" [DB2] data block onto the connections of the called functions and function blocks in the "Main" [OB1] organization block. First we move the 'Speed_Actual_Value' tag onto the 'Actual_speed' output of the "MOTOR_SPEEDMONITORING" [FC11] block.

(→ Speed_Actual_Value)
→ Also connect the other contacts in Network 1 with tags from the “SPEED_MOTOR” [DB2] data block as shown here.

→ Connect the contacts in Network 2 with tags from the “SPEED_MOTOR” [DB2] data block as shown here.
Connect the contacts in Network 3 with tags from the "SPEED_MOTOR" [DB2] data block as shown here.
7.4 Save and compile the program

→ To save your project, click the button in the menu. To compile all blocks, click the "Program blocks" folder and select the icon for compiling in the menu.

(→ Save project → Program blocks → )

→ The "Info", "Compile" area shows which blocks were successfully compiled.

<table>
<thead>
<tr>
<th>Description</th>
<th>Go to</th>
<th>Errors</th>
<th>Warnings</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU1516F</td>
<td></td>
<td>0</td>
<td>0</td>
<td>7:26:42 AM</td>
</tr>
<tr>
<td>Program blocks</td>
<td></td>
<td>0</td>
<td>0</td>
<td>7:26:42 AM</td>
</tr>
<tr>
<td>SPEED_MOTOR (D82)</td>
<td></td>
<td>0</td>
<td>0</td>
<td>7:26:42 AM</td>
</tr>
<tr>
<td>Main (OB1)</td>
<td></td>
<td>0</td>
<td>0</td>
<td>7:26:44 AM</td>
</tr>
<tr>
<td>Blocks was successfully compiled</td>
<td></td>
<td></td>
<td></td>
<td>7:26:46 AM</td>
</tr>
</tbody>
</table>
7.5 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.
7.6 Monitor/modify values in data blocks

- The desired block must be open for monitoring the tags of a downloaded data block. The monitoring can then be activated/deactivated by clicking the icon.

(\(\rightarrow\) SPEED_MOTOR [DB2] \(\rightarrow\))

- In the ‘Monitor value’ column, the values currently available in the CPU can be monitored.
If your right-click on one of the values, the 'Modify' dialog for modifying this value opens.

(→ Modify → Modify value: 14.0 → OK)

7.7 Initialize setpoints/reset start values

→ The setpoints can be initialized by clicking the icon. For the tags whose 'Setpoint' check box is selected , the start value will then be applied as the current value.
All start values can be reset by clicking the icon.
7.8 Snapshots in data blocks

If you click the 'Snapshot' icon, a snapshot of the monitored values can be taken in order to apply these values as start values or to transfer them back to the CPU later.
Alternatively, values from the snapshot can be applied by clicking the '...' icon for all values or by clicking the '...' icon for the start values only. Only the setpoints are needed here in most cases.
To transfer data captured in the snapshot back to the CPU, you must click '→'.

(→)
If you want to overwrite all setpoints with the start values, you can initiate this by clicking '→'. The values in the CPU for which the 'Setpoint' check box was not selected are thereby retained.

(→ →)
7.9 Expand data block and download it without reinitialization

To enable 'Download without reinitialization' for the "SPEED_MOTOR" [DB2] data block, you must 'Offline' and then open the properties of the data block. (→ Offline → SPEED_MOTOR [DB2] → Properties)

Select the 'Optimized block access' check box in the properties under 'General', 'Attributes'. (→ General → Attributes → Optimized block access)
→ Assign a ‘Retentive memory reserve’ to the data block for ‘Download without reinitialization’.
(→Download without reinitialization → Retentive memory reserve → 10 bytes → OK)

→ Download your "SPEED_MOTOR" [DB] data block to the controller again and select ' 

(→SPEED_MOTOR [DB] → Go online)
→ Then click the 'Download without reinitialization' icon to activate download without reinitialization and confirm the safety prompt with 'OK'.

(→ Download without reinitialization → OK)
Next add any tag in your data block

(Name: Value_test → Data type: Real → Start value: 99)

Download your "SPEED_MOTOR" [DB] data block to the controller again.

(SPEED_MOTOR [DB] → Download)
If you click '□' to monitor the block again, you will see that the monitored values have not been overwritten with the start values.

7.10 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”.

(→ Project → Archive → TIA Portal project archive → 032-600_Global_Data_Blocks.... → Save)
## 8 Checklist

<table>
<thead>
<tr>
<th>No.</th>
<th>Description</th>
<th>Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data block SPEED_MOTOR [DB2] successfully created.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Program changes made in Main [OB1].</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Compiling successful and without error message</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Download successful and without error message</td>
<td></td>
</tr>
</tbody>
</table>
| 5   | 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 |           |
| 6   | Sensor part at end of conveyor activated (-B7 = 1)  
(>Q3 = 0  
(after 2 seconds) |           |
| 7   | Briefly press the automatic stop pushbutton (-S2 = 0)  
-Q3 = 0 |           |
| 8   | Activate EMERGENCY OFF (-A1 = 0)  
-Q3 = 0 |           |
| 9   | Manual mode (-S0 = 0)  
-Q3 = 0 |           |
| 10  | Switch off station (-K0 = 0)  
-Q3 = 0 |           |
| 11  | Cylinder not retracted (-B1 = 0)  
-Q3 = 0 |           |
| 12  | Speed > Motor_speed_monitoring_error_max  
-Q3 = 0 |           |
| 13  | Speed < Motor_speed_monitoring_error_min  
-Q3 = 0 |           |
| 14  | Project successfully archived |           |
9 Exercise

9.1 Task – Exercise

In this exercise a global data block "MAGAZINE_PLASTIC" [DB3] will be created additionally. The setpoint and actual value of the counter for the plastic parts will be specified and displayed in this data block.

A connectable input for the setpoint setting and an output for displaying the actual value will also be added to the "MOTOR_AUTO" [FB1] function block.

9.2 Technology diagram

Here you see the technology diagram for the task.

Figure 5: Technology diagram

Figure 6: Control panel
## 9.3 Reference list

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

<table>
<thead>
<tr>
<th>DI</th>
<th>Type</th>
<th>Identifier</th>
<th>Function</th>
<th>NC/NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>I 0.0</td>
<td>BOOL</td>
<td>-A1</td>
<td>Return signal emergency stop OK</td>
<td>NC</td>
</tr>
<tr>
<td>I 0.1</td>
<td>BOOL</td>
<td>-K0</td>
<td>Main switch &quot;ON&quot;</td>
<td>NO</td>
</tr>
<tr>
<td>I 0.2</td>
<td>BOOL</td>
<td>-S0</td>
<td>Mode selector manual (0)/ automatic (1)</td>
<td>Manual = 0&lt;br&gt;Auto = 1</td>
</tr>
<tr>
<td>I 0.3</td>
<td>BOOL</td>
<td>-S1</td>
<td>Pushbutton automatic start</td>
<td>NO</td>
</tr>
<tr>
<td>I 0.4</td>
<td>BOOL</td>
<td>-S2</td>
<td>Pushbutton automatic stop</td>
<td>NC</td>
</tr>
<tr>
<td>I 0.5</td>
<td>BOOL</td>
<td>-B1</td>
<td>Sensor cylinder -M4 retracted</td>
<td>NO</td>
</tr>
<tr>
<td>I 1.0</td>
<td>BOOL</td>
<td>-B4</td>
<td>Sensor part at slide</td>
<td>NO</td>
</tr>
<tr>
<td>I 1.3</td>
<td>BOOL</td>
<td>-B7</td>
<td>Sensor part at end of conveyor</td>
<td>NO</td>
</tr>
<tr>
<td>IW64</td>
<td>BOOL</td>
<td>-B8</td>
<td>Sensor actual value speed of the motor +/−10V corresponds to +/− 50 U/min</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DO</th>
<th>Type</th>
<th>Identifier</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q 0.2</td>
<td>BOOL</td>
<td>-Q3</td>
<td>Conveyor motor -M1 variable speed</td>
</tr>
<tr>
<td>QW 64</td>
<td>BOOL</td>
<td>-U1</td>
<td>Manipulated value speed of the motor in 2 directions +/− 10V corresponds to +/− 50 rpm</td>
</tr>
</tbody>
</table>

**Legend for reference list**

<table>
<thead>
<tr>
<th>DI</th>
<th>Digital Input</th>
<th>DO</th>
<th>Digital Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Analog Input</td>
<td>AO</td>
<td>Analog Output</td>
</tr>
<tr>
<td>I</td>
<td>Input</td>
<td>Q</td>
<td>Output</td>
</tr>
<tr>
<td>NC</td>
<td>Normally Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NO</td>
<td>Normally Open</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## 9.4 Planning

Plan the implementation of the task on your own.
## 9.5 Checklist – Exercise

<table>
<thead>
<tr>
<th>No.</th>
<th>Description</th>
<th>Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data block MAGAZINE_PLASTIC [DB3] successfully created.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Program changes made in MOTOR_AUTO [FB1].</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Program changes made in Main [OB1].</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Compiling successful and without error message</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Download successful and without error message</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Sensor part at end of conveyor activated (-B7 = 1) → -Q3 = 0 (after 2 seconds)</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Briefly press the automatic stop pushbutton (-S2 = 0) → -Q3 = 0</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Activate EMERGENCY OFF (-A1 = 0) → -Q3 = 0</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Manual mode (-S0 = 0) → -Q3 = 0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Switch off station (-K0 = 0) → -Q3 = 0</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Cylinder not retracted (-B1 = 0) → -Q3 = 0</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Speed &gt; Motor_speed_monitoring_error_max → -Q3 = 0</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Speed &lt; Motor_speed_monitoring_error_min → -Q3 = 0</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Project successfully archived</td>
<td></td>
</tr>
</tbody>
</table>
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