SIEMENS

Ingenuity for life

ET 200SP Motor Starter: Reading and writing data sets with HMI connection

ET 200SP / Motor starter:

https://support.industry.siemens.com/cs/ww/de/view/109750305

Siemens
Industry
Online
Support

Warranty and Liability

Note

The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These Application Examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice.
If there are any deviations between the recommendations provided in these Application Examples and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.
Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act ("Produkthaftungsgesetz"), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract ("wesentliche Vertragspflichten"). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.
Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of the Siemens AG.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.
In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions only form one element of such a concept.
Customer is responsible to prevent unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place.
Additionally, Siemens' guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit http://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.
To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under http://www.siemens.com/industrialsecurity.

Table of Contents

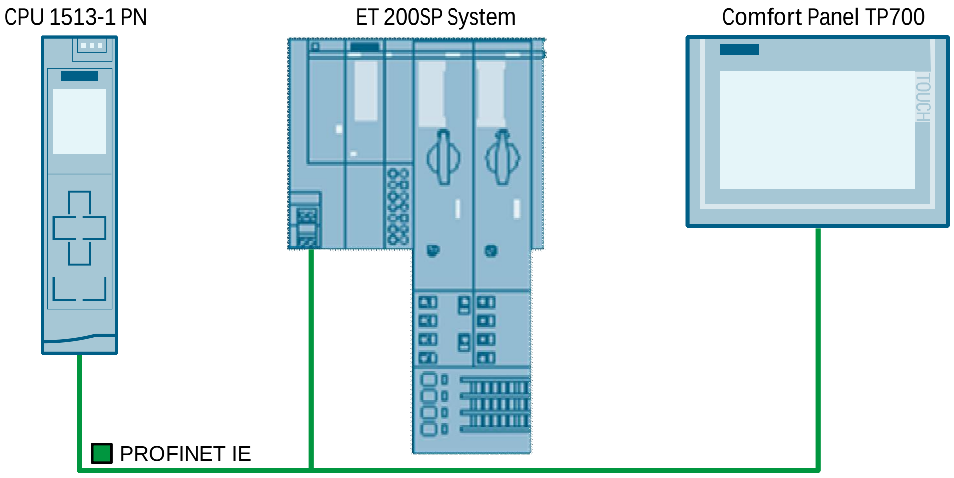
1 Introduction

1.1 Overview

This application example offers you the possibility to acyclically query and, if necessary, change all available data sets and parameters of the ET200SP motor starter (3RK1308-0**00-0CP0) via a panel and during operation.

The following figure shows the setup of the components used in the application example.

Figure 1-1 setup of the application example



1.2 Mode of operation

The parameters of the ET 200SP motor starter can be read or written via data sets. These data sets are read via the RDREC function and written via the WRREC function.

More information on the setup of the different data structures can be found in the manual of the ET 200SP motor starter (see /5/).

In this application example, reading and writing the data sets is exemplary demonstrated by using two different motor starter types. The functions are executed with a direct starter 0.9-3 A (3RK1 308-0AC00-0CP0) and a reversing starter 0.3-1 A (3RK1 308-0BB00-0CP0). The functions of the application example can be executed for all types of the ET 200SP motor starter (also F starters).

To facilitate the use of the data sets, this application example uses the data types for the ET 200SP motor starter. These data types can be found in the Industry Online Support (see /6/).

For the understanding of the application example, the following knowledge is assumed and therefore not further explained in the application example:

- TIA Portal and WinCC configuration software
- Basics of STEP 7 programming
- Programming in SCL

1.3 Components used

This application example was created with the following hardware and software components:

Table 1-1 Hardware components

Component	Number	Article number	Note
CPU 1513-1 PN	1	6ES7513-1AL00-0AB0	FW V1.8; Alternatively, any other S7-1500 CPU as of firmware FW V1.8 can also be used.
4MB Memory Card	1	6ES7954-8LC02-0AA0	
ET 200SP IM 155-6 PN ST	1	6ES7155-6AU00-0BN0	FW V3.3
ET 200SP Base Unit, Typ A0	1	6ES7193-6BP00-0BA0	Empty Base Unit, due to system setup of the ET 200SP motor starter
ET 200SP Base Unit Cover	1	6ES7133-6CV15-1AM0	Cover for the empty Base Unit
ET 200SP Motor starter DS 0.9-3A HF	1	3RK1308-0AC00-0CP0	FW V1.0
ET 200SP Base Unit for motor starter MS1	1	3RK1908-0AP00-0AP0	Base Unit with 24V and feed-in system 400V
ET 200SP Motor starter RS 0.3-1A HF	1	3RK1308-0BB00-0CP0	FW V1.1
ET 200SP Base Unit for motor starter MS4	1	3RK1908-0AP00-0DP0	Base Unit without feed-in
ET 200SP energy bus cover	1	3RK1908-1DA00-2BP0	
ET 200SP Motor starter 3DI/LC module	2	3RK1908-1AA00-0BP0	
ET 200SP server module	1	6ES7193-6PA00-0AA0	
SIMATIC Comfort Panel TP700 Comfort	1	6AV2124-0GC01-0AX0	FW V14.0.1.0

Table 1-2 Software components

Component	Number	Article number	Note
STEP 7 PROFESSIONAL V14 SP1	1	6ES7822-1..04-..	Alternatively, a smaller package is also possible.
WinCC Engineering V14 SP1	1	6AV210.-....4-0	Can be used with WinCC Comfort and higher.

This application example consists of the following components:

Table 1-3

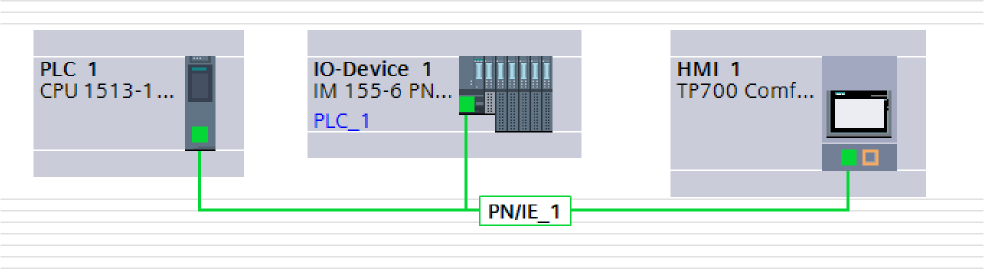
Component	File name	Note
Project	109750305_ET200SPMS_Rd_Wr_PROJ_V11.zip	Created with V14 SP1
Library	109750305_ET200SPMS_Rd_Wr_LIB_V11.zip	
Documentation	109750305_ET200SPMS_Rd_Wr_DOC_V11_en.pdf	This document

2 Engineering

2.1 Setup of control system

The control system setup from the network view in TIA Portal is exemplary shown in the following figure and may vary from your configuration.

Figure 2Structure of control system



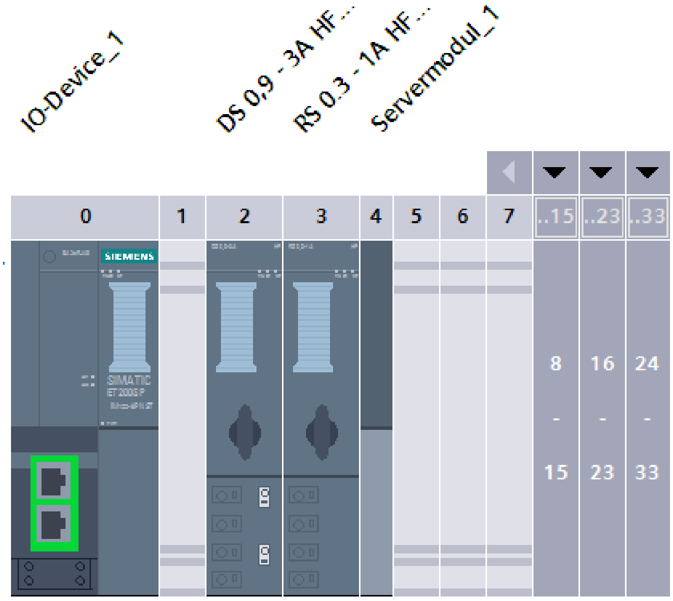
2.2 Configuration and settings

Trouble-free use of the ET 200SP motor starter requires that you use an empty module in front and on the left of the first motor starter.

More information about the system structure with the ET 200SP motor starter can be found in the “SIMATIC ET 200SP Distributed I/O System” system manual(\7\).

The empty module must be regarded as empty slot also in the hardware configuration.

Figure 2-1 Hardware configuration ET 200SP System



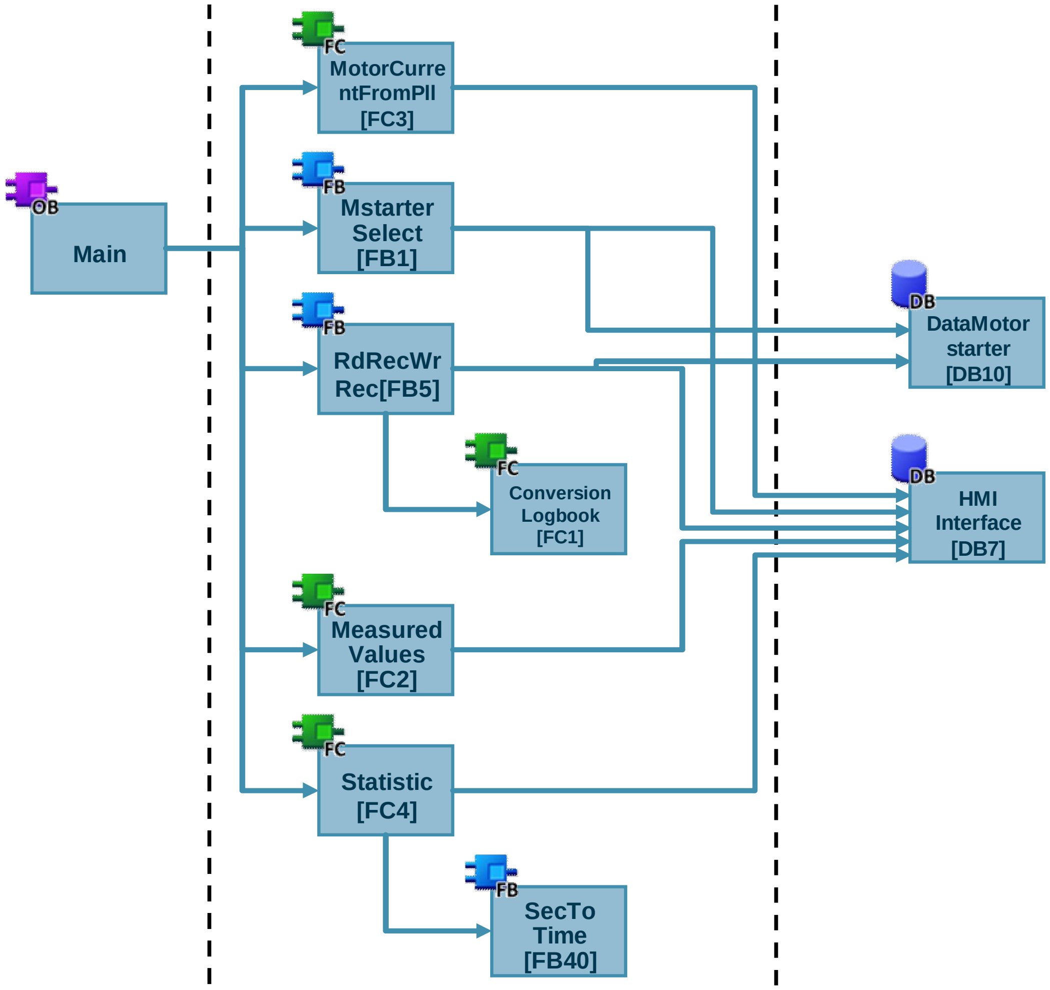
3 Block description

3.1 Program structure

To be able to integrate the functions of the application example into your project, the "109750305_ET200SPMS_Rd_Wr_LIB_V1.zip" block library
And the "109750305_ET200SPMS_Rd_Wr_PROJ_V1.zip" project is available to you as download (\2\).

The program sequence depends on the number of the motor starters used in the system and needs to be adjusted to your setup. This example is built with two ET 200SP motor starters.

Figure 3-1 Program structure



Note The block numbers can be changed.

The “109750305_ET200SPMS_Rd_Wr_LIB_V1.zip” block library offers the following contents:

- HMI_1: this is the complete HMI project
- PLC data types: in this folder all data types created for the project have been integrated
- PLC program blocks: this folder contains all the program blocks, FBs, FCs and DBs necessary for the creation of this application example
- PLC tags: Contains the created tag chart for the example

To facilitate the use of the data sets, this application example uses the data types for the ET 200SP motor starter. These data types can be found in the Industry Online Support (see /6/).

The “typeDataMotorStarter” data type has been newly created for the example. In this data type, all available data types for the ET 200SP motor starter have been integrated. It serves to assign the data of the ET 200SP motor starter, that are stored in “DataMotorstarter[DB10]”.

The “typeFpLogbook” data type has also been newly created for the example and consists of 21 entries which represent the scope of the logbooks and are nested in the “typeDataMotorStarter” data type. The entries of the “typeFpLogbook” data type again subdivide into a “typeFpLogbookEntry” data type. It contains the information of object number, day, hours, minutes and seconds which are calculated in the “ConversionLogbook [FC1]” and stored on the “DataMotorstarter [DB10]” data block.

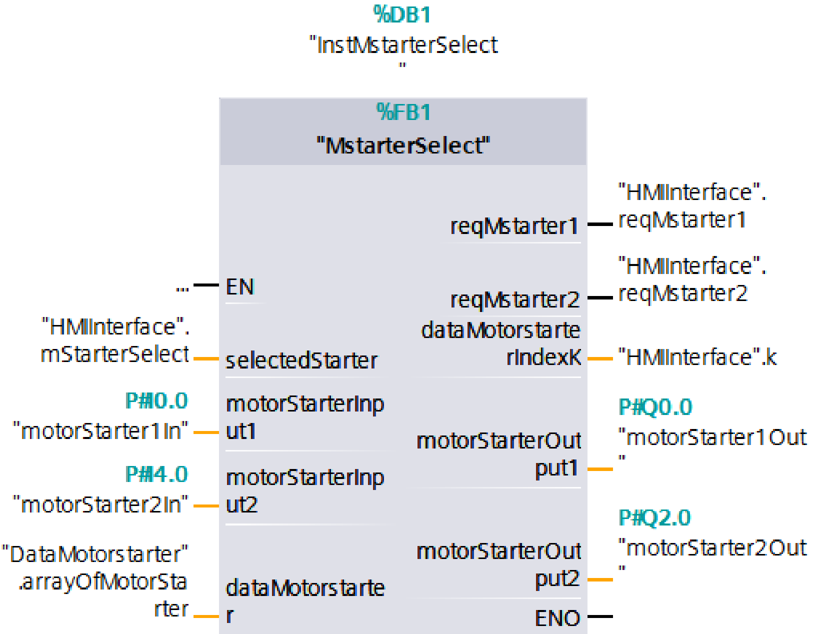
In the following chapters, the program blocks are explained.

3.1.1 “MstarterSelect [FB1]”

Via the “MstarterSelect [FB1]” function block, the information from the HMI, which motor starter is currently selected at the panel, is transmitted. Additionally, the process image of the inputs is read in in this block.

The following figure shows the call of the “MstarterSelect” block.

Figure 3-2 Interconnected “MstarterSelect” FB1



The activation of the process image of the outputs is done separately in the OB1, so that there will not be any double accesses of the outputs in the further course of your application, see chapter 3.1.7 “Controlling the process image of the “.

The “MstarterSelect” function block has the following inputs and inputs/outputs:

Table 3-1 Inputs and inputs/outputs of the “MstarterSelect” FB

Name	Data type	Description
selectedStarter	Int	Here, the panel reads in the information about which motor starter has been selected.
motorStarterInput1	"LPD_type Motor StarterIn"	Input address of the respective ET 200SP motor starter. In this example, the process image of the inputs for the motor starter has been stored in the global tag chart on the “motorStarter1/2In” tags with the “LPD_typeMotorStarterIn” data type. The status of the inputs is stored on the “dataMotorstarter” output tag in the block.
motorStarterInput2	"LPD_type Motor StarterIn"	
dataMotorstarter	Array[0..1] of “typeDataMotorStarter”	Contains the information of the motor starters that are stored after the RDREC or WRREC request.

The “MstarterSelect” block has the following outputs:

Table 3-2 Outputs of “MstarterSelect” FB

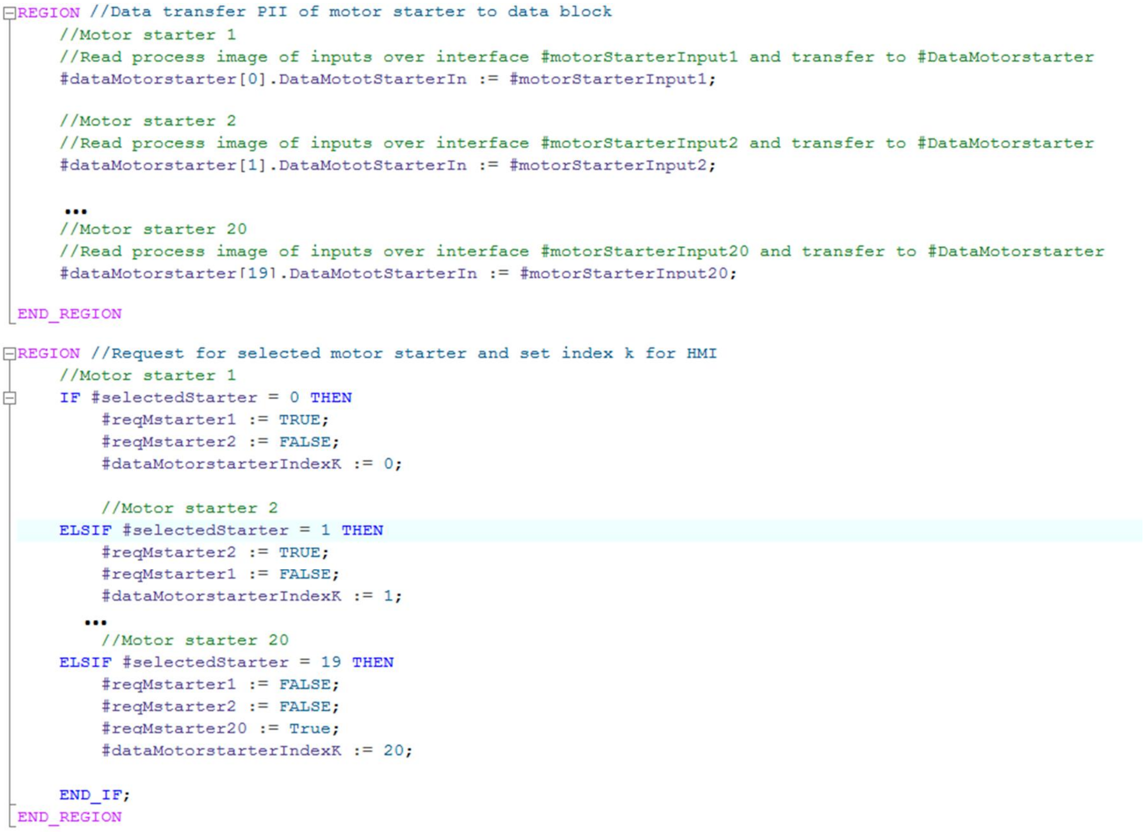
Name	Data type	Description
reqMstarter1	Bool	Serve as trigger, so that upon selecting a motor starter at the HMI, all data sets are read out once consecutively and the current information from the data sets are displayed on the HMI.
reqMstarter2	Bool	
dataMotorstarter IndexK	Int	Index for the HMI. This way, the data of the selected motor starter are accessed via a multiplex tag at the HMI.

Note

If you use more than two ET 200SP motor starters in your system setup, you need to extend the block with further inputs and outputs.

The following figure exemplary shows, how the program needs to be adjusted for the use of 20 ET 200SP motor starters.

Figure 3-3 Exemplary program extension “MstarterSelect” [FB1]



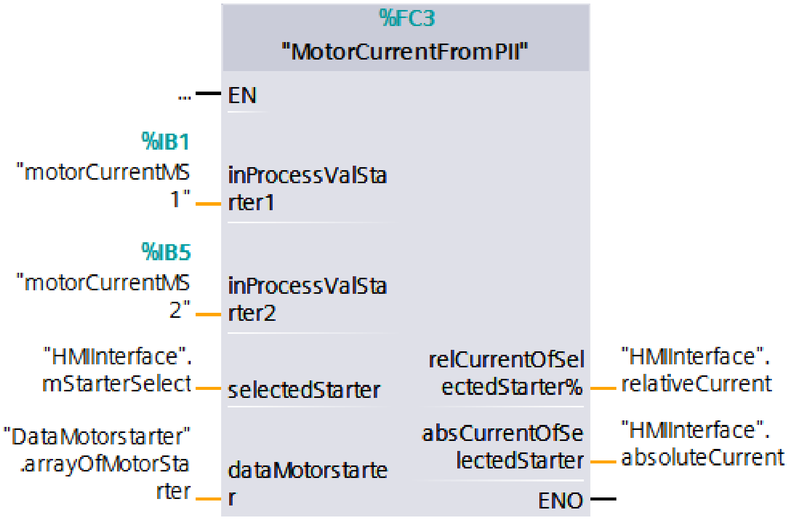
```
REGION //Data transfer PII of motor starter to data block
    //Motor starter 1
    //Read process image of inputs over interface #motorStarterInput1 and transfer to #DataMotorstarter
    #dataMotorstarter[0].DataMototStarterIn := #motorStarterInput1;

    //Motor starter 2
    //Read process image of inputs over interface #motorStarterInput2 and transfer to #DataMotorstarter
    #dataMotorstarter[1].DataMototStarterIn := #motorStarterInput2;

    ...
    //Motor starter 20
    //Read process image of inputs over interface #motorStarterInput20 and transfer to #DataMotorstarter
    #dataMotorstarter[19].DataMototStarterIn := #motorStarterInput20;

END_REGION

REGION //Request for selected motor starter and set index k for HMI
    //Motor starter 1
    IF #selectedStarter = 0 THEN
        #reqMstarter1 := TRUE;
        #reqMstarter2 := FALSE;
        #dataMotorstarterIndexK := 0;

        //Motor starter 2
    ELSIF #selectedStarter = 1 THEN
        #reqMstarter2 := TRUE;
        #reqMstarter1 := FALSE;
        #dataMotorstarterIndexK := 1;
      ...
        //Motor starter 20
    ELSIF #selectedStarter = 19 THEN
        #reqMstarter1 := FALSE;
        #reqMstarter2 := FALSE;
        #reqMstarter20 := True;
        #dataMotorstarterIndexK := 20;

    END_IF;
END_REGION
```

3.1.2 “MotorCurrentFromPII[FC3]”

With the function “MotorCurrentFromPII[FC3]”, a real motor current is calculated by the 6bit values in the process image of the motor starter’s inputs, which display the current motor current in percentage. (PII stands for Process Image of Inputs)

The Figure below shows the call of the “MotorCurrentFromPII” function.

Figure 3-4 Interconnected “MotorCurrentFromPII” FC3


%FC3
"MotorCurrentFromPII"
...
EN
%IB1
"motorCurrentMS 1"
inProcessValStarter1
%IB5
"motorCurrentMS 2"
inProcessValStarter2
"HMIInterface".mStarterSelect
selectedStarter
relCurrentOfSelectedStarter%
"HMIInterface".relativeCurrent
"DataMotorstarter".arrayOfMotorStarter
dataMotorstarter
absCurrentOfSelectedStarter
"HMIInterface".absoluteCurrent
ENO

The function has the following inputs and inputs/outputs:

Table 3-3 Inputs and inputs/outputs of the “MotorCurrentFromPII” FC

Name	Data type	Description
inProcessValStarter1	Byte	Address of input byte that reflects the motor current from the process image of the motor starter’s inputs.
inProcessValStarter2	Byte	
selectedStarter	Int	Here, the information from the “HMIInterface” DB is read out, which motor starter is currently active.
dataMotorstarter	Array[0..1] of "typeData MotorStarter"	Contains the information of the motor starters that are stored after the RDREC or WRREC request.

The function has the following outputs:

Table 3-4 Outputs of "MotorCurrentFromPII” FC

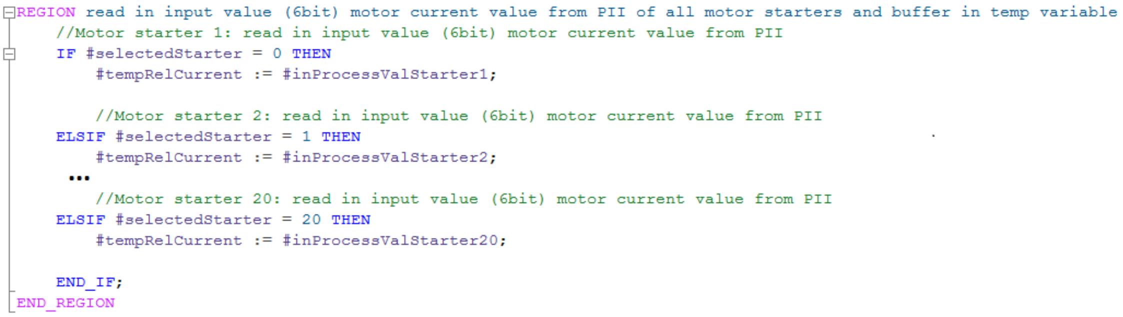
Name	Data type	Description
relativCurrentOf SelectedStarter%	Real	Current value of the selected motor starter in percent.
absCurrentOf SelectedStarter	Real	Absolute current value of the selected motor starter.

Note

> If you use more than two ET 200SP motor starters in your system setup, you need to extend the block with further inputs and outputs.

The following figure exemplary shows, how the program needs to be adjusted for the use of 20 ET 200SP motor starters.

Figure 3-5 Exemplary program extension “MotorCurrentFromPII” [FC3]



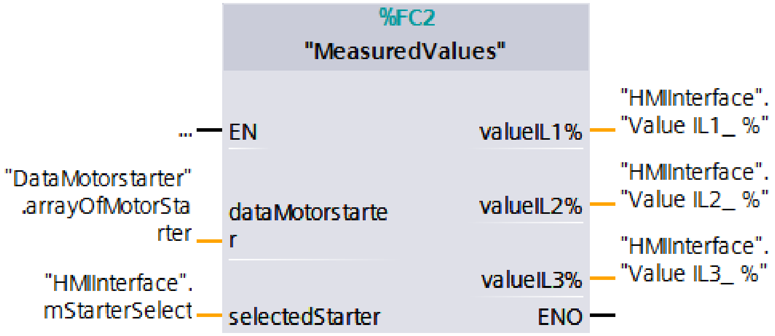
```
REGION read in input value (6bit) motor current value from PII of all motor starters and buffer in temp variable
    //Motor starter 1: read in input value (6bit) motor current value from PII
    IF #selectedStarter = 0 THEN
        #tempRelCurrent := #inProcessValStarter1;

        //Motor starter 2: read in input value (6bit) motor current value from PII
    ELSIF #selectedStarter = 1 THEN
        #tempRelCurrent := #inProcessValStarter2;
    ...
        //Motor starter 20: read in input value (6bit) motor current value from PII
    ELSIF #selectedStarter = 20 THEN
        #tempRelCurrent := #inProcessValStarter20;

    END_IF;
END_REGION
```

3.1.3 “MeasuredValues [FC2]”

This block serves to convert the phase current that is read from the data set 94 (reading measured values). The value of each phase is stored as byte in the data set and converted as real value in the function.

The following figure shows the call of the “Measured Values” function.

Figure 3-6 Interconnected “MeasuredValues” FC2


%FC2
"MeasuredValues"
...
EN
"DataMotorstarter".arrayOfMotorStarter
dataMotorstarter
"HMIInterface".mStarterSelect
selectedStarter
valueIL1%
"HMIInterface"."Value IL1_ %"
valueIL2%
"HMIInterface"."Value IL2_ %"
valueIL3%
"HMIInterface"."Value IL3_ %"
ENO

The function has the following inputs and inputs/outputs:

Table 3-5 Inputs and inputs/outputs of the “MeasuredValues” FC

Name	Data type	Description
dataMotorstarter	Array[0..1] of "typeData MotorStarter"	Contains the information of the motor starters that are stored after the RDREC or WRREC request.
selectedStarter	Int	Here, the information from the “HMIInterface” DB is read out, which motor starter is currently active.

The function has the following outputs:

Table 3-6 Outputs of “MeasuredValues” FC

Name	Data type	Description
valueIL1	Real	Supplies the converted phase current of the respective motor phase.
valueIL2	Real	
valueIL3	Real	

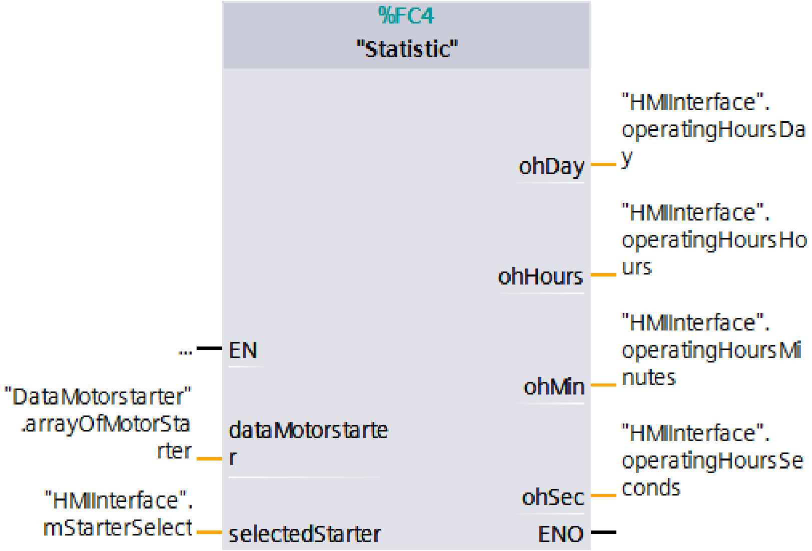
3.1.4 “Statistic [FC4]” and “SecToTime [FB40]”

In the “Statistic [FC4]” block, the operating hours counter (which is read from the data set 95 as second value) is converted into days, hours, minutes and seconds. The conversion is done via the “SecToTime [FB40]” function block that is called up in this block. The basic function of the “SecToTime [FB40]” block was taken from the “MillisecToTime” function(\8\).

The converted values are transferred to the HMI via the “HMIInterface” DB.

The following figure shows the interconnected “Statistic” block.

Figure 3-7 Interconnected “Statistic” FC4


%FC4
"Statistic"
"HMIInterface".operatingHoursDay
ohDay
"HMIInterface".operatingHoursHours
ohHours
...
EN
"HMIInterface".operatingHoursMinutes
ohMin
"DataMotorstarter".arrayOfMotorStarter
dataMotorstarter
"HMIInterface".operatingHoursSeconds
ohSec
"HMIInterface".mStarterSelect
selectedStarter
ENO

The block has the following inputs:

Table 3-7 Inputs and inputs of “Statistic” FB

Name	Data type	Description
dataMotorstarter	Array[0..1] of "typeData MotorStarter"	Contains the information of the motor starters that are stored after the RDREC or WRREC request.
selectedStarter	Int	Here, the information from the “HMIInterface” DB is read out, which motor starter is currently active.

The block has the following outputs:

Table 3-8 Outputs of “Statistic” FB

Name	Data type	Description
ohDay	Int	supplies the calculated day value from the operating hours counter
ohHours	Int	supplies the calculated hour value from the operating hours counter
ohMin	Int	supplies the calculated second value from the operating hours counter
ohSec	Int	supplies the calculated minute value from the operating hours counter

3.1.5 “RdRecWrRec [FB5]”

The main function of the read/write example of the data sets is realized via the “RdRecWrRec [FB5]” function block. The data sets are read via the RDREC function and written via the WRREC function.

The following data sets are read:

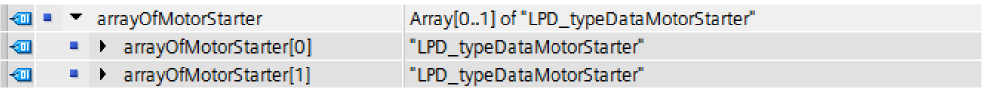
- Logbooks (Data set 72, 73 and 75)
- Diagnosis (Data set 92)
- Measured values (Data set 94)
- Statistical data (Data set 95)
- Parameters (Data set 201 and 203)

This block needs to be called up separately for each motor starter in the system. The information that are read from/ written to the data sets are stored in an array in the “DataMotorstarter [DB10]” data block. For each motor starter, there is a separate field reserved in the array.

This means that in this example, an array with two fields is created in DB10. If, for example, you use 20 ET 200SP in your setup, you create an array with 20 fields (“Array [0..19] of "typeDataMotorStarter”).

The following figure shows the “DataMotorstarter” DB of the application example.

Figure -3-8 Array DataMotorstarter DB10

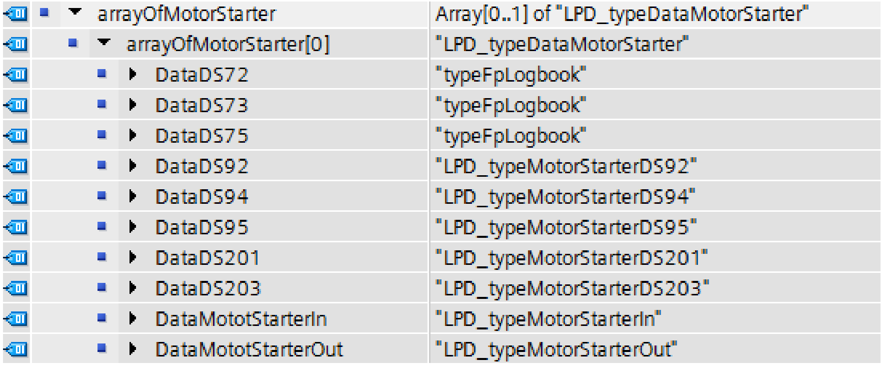


arrayOfMotorStarter	Array[0..1] of "LPD_typeDataMotorStarter"
arrayOfMotorStarter[0]	"LPD_typeDataMotorStarter"
arrayOfMotorStarter[1]	"LPD_typeDataMotorStarter"

The array consists of a compound data type with the name “typeDataMotorStarter”.

The following figure shows the setup of the “typeDataMotorStarter” data type.

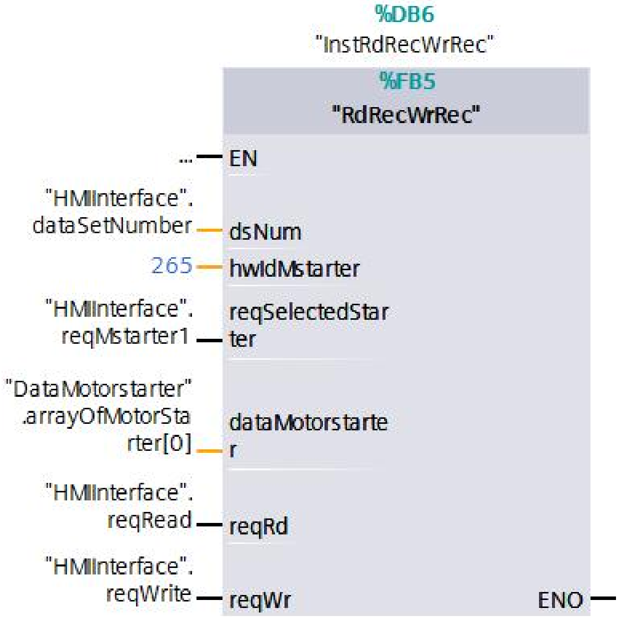
Figure -3-9 Setup of the array DataMotorstarter DB10



arrayOfMotorStarter	Array[0..1] of "LPD_typeDataMotorStarter"
arrayOfMotorStarter[0]	"LPD_typeDataMotorStarter"
DataDS72	"typeFpLogbook"
DataDS73	"typeFpLogbook"
DataDS75	"typeFpLogbook"
DataDS92	"LPD_typeMotorStarterDS92"
DataDS94	"LPD_typeMotorStarterDS94"
DataDS95	"LPD_typeMotorStarterDS95"
DataDS201	"LPD_typeMotorStarterDS201"
DataDS203	"LPD_typeMotorStarterDS203"
DataMototStarterIn	"LPD_typeMotorStarterIn"
DataMototStarterOut	"LPD_typeMotorStarterOut"

The following figure shows the call of the “RdRecWrRec” data block.

Figure 3-10 Interconnected “RdRecWrRec” FB5

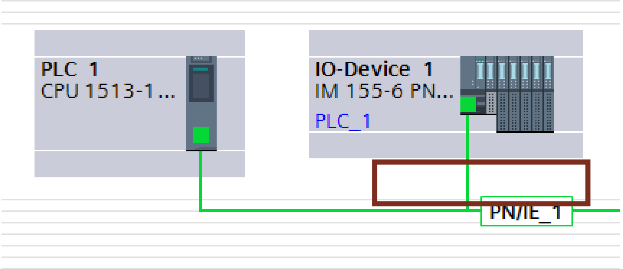


%DB6
"InstRdRecWrRec"
%FB5
"RdRecWrRec"

... — EN
"HMIInterface".dataSetNumber — dsNum
265 — hwIdMstarter
"HMIInterface".reqMstarter1 — reqSelectedStarter
"DataMotorstarter".arrayOfMotorStarter[0] — dataMotorstarter
"HMIInterface".reqRead — reqRd
"HMIInterface".reqWrite — reqWr
ENO —

The “RdRecWrRec” block has the following inputs and inputs/outputs:

Table 3-9 Inputs and inputs/outputs of the “MstarterSelect” FB

Name	Data type	Description
dsNum	Int	Transfers the data set number for manual reading or writing from the HMI.
hwIdMstarter	HW_IO	Hardware ID of motor starter that is assigned to the corresponding motor starter in the hardware configuration.  PLC_1 CPU 1513-1 ... IO-Device_1 IM 155-6 PN... PLC_1 PN/IE_1
reqSelectedStarter	Bool	Serves as trigger to consecutively read out all data sets of the desired motor starter and to always receive current information from the data sets.
dataMotorstarter	"typeData Motor Starter"	Contains the information of the corresponding motor starter that are stored after the RDREC or WRREC request. Here, it is switched to the corresponding array of the created motor starter.
reqRd	Bool	Serves as trigger for a manual read request.
reqWr	Bool	Serves as trigger for a manual write request.

Note

The number of simultaneous diagnostic data calls via the “RDREC” system function used in the function block is limited, depending on the CPU used. The number of possible simultaneous calls can the found in the CPU’s manual (\3\).

Reference value for CPU 1500: 10 simultaneous calls.

3.1.6 “ConversionLogbook [FC1]”

During the read-out of the motor starter’s logbooks, the operating hours of the device and an object number for each event are read. The operating hours counter is specified as second value. To make a precise statement as to when an event took place, this second value is converted into days, hours, minutes and seconds in the “ConversionLogbook[FC1]” function. After the read-out of the logbooks DS72, DS73 and DS75, the block is called up in the “RdRecWrRec [FB5]” function block.

3.1.7 Controlling the process image of the outputs

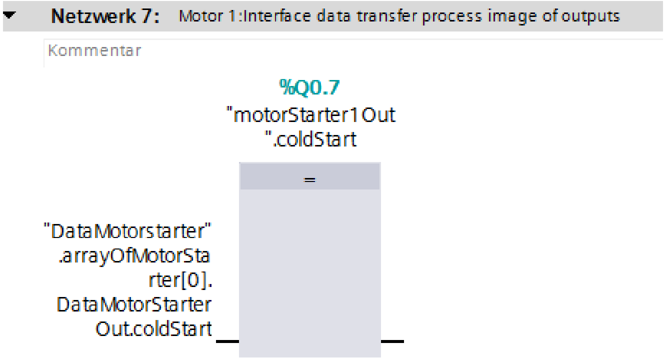
In order to make sure that there are no simultaneous accesses to the process image of the ET 200SP motor starters’ outputs from multiple points, the outputs are individually described in OB1. This way, you can integrate your additional switching conditions at this interface for your application.

In this example, the process image of the outputs for the motor starter has been stored in the global tag chart on the “motorStarter1/2Out” tags with the “LPD_typeMotorStarterOut” data type.

To be able to switch the outputs via the HMI image “motor control”, the outputs of the used motor starter need to be switched in OB1. If you use more than two ET 200SP motor starters in your system setup, you need to extend the switching accordingly.

The following figure shows the interconnection of the process image of the outputs, using the example of “motorStarter1Out”.

Figure 3-11 Exemplary switching of the process image of the outputs



Netzwerk 7: Motor 1:Interface data transfer process image of outputs

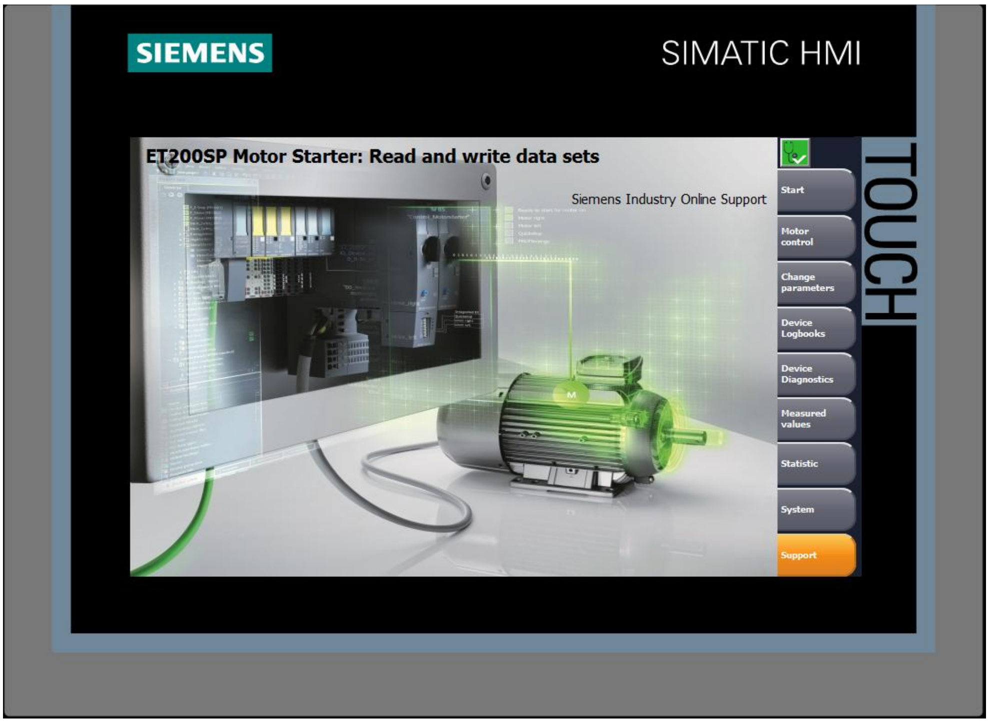
Kommentar

4 Operating the application example

4.1 Overview

The following figure shows the start screen of the HMI.

Figure 4-1 Start screen on HMI



The HMI project consists of 6 screens. These screens display the read information from the data sets of the used ET 200SP motor starter. Besides the system diagnostics window, a start, a system and a support screen are also integrated.

In the following, it will be described in detail how to operate the individual screens.

4.2 Operation

4.2.1 Toolbar

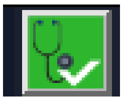
The individual screens are selected via the toolbar on the right. It is possible to access the following screens from anywhere:

- Start screen
- Motor control
- Change parameters
- Device logbooks
- Device diagnosis
- Measured values
- Statistics
- System
- Support

4.2.2 System diagnostics window

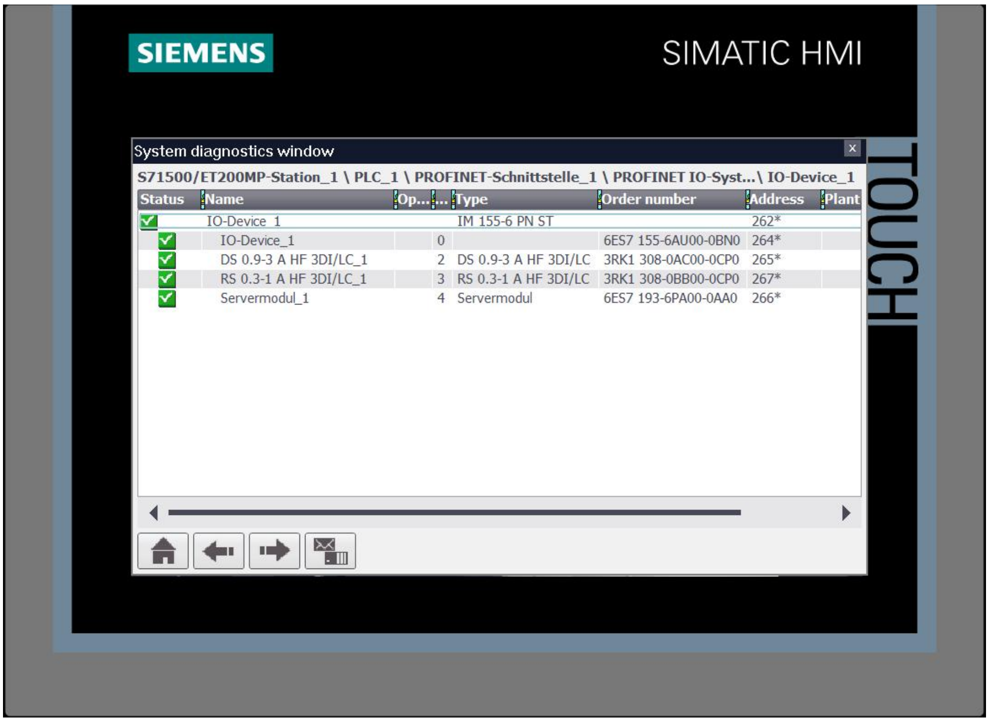
A diagnostics button to call up the system diagnostics window is integrated in the toolbar.

Figure 4-2 Button to call up the system diagnostics



The System diagnostics window offers you an overview of all available devices in your plant. You navigate directly to the cause of the error and to the respective device. You have access to all diagnostics-capable devices which you have configured in the "Devices & Networks" editor.

Figure 4-3 Extract from the “System diagnostics window”



S71500/ET200MP-Station_1 \ PLC_1 \ PROFINET-Schnittstelle_1 \ PROFINET IO-Syst...\ IO-Device_1

Status	Name	Op...	...	Type	Order number	Address	Plant
✓	IO-Device 1			IM 155-6 PN ST		262*	
✓	IO-Device_1		0		6ES7 155-6AU00-0BN0	264*	
✓	DS 0.9-3 A HF 3DI/LC_1		2	DS 0.9-3 A HF 3DI/LC	3RK1 308-0AC00-0CP0	265*	
✓	RS 0.3-1 A HF 3DI/LC_1		3	RS 0.3-1 A HF 3DI/LC	3RK1 308-0BB00-0CP0	267*	
✓	Servermodul_1		4	Servermodul	6ES7 193-6PA00-0AA0	266*	

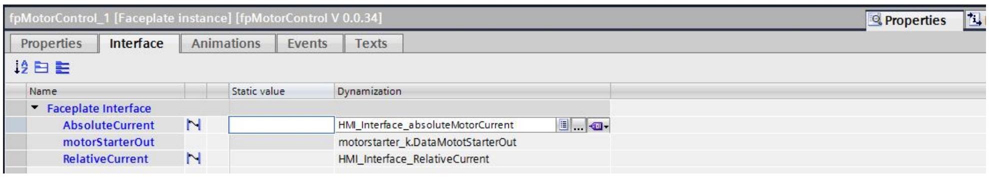
4.2.3 Using faceplates

In this application example, the contents of the HMI screens are generated as faceplates. A faceplate is a block that makes it easier for you to connect the individual process tags to the HMI. You also get the advantage of being able to use the faceplate for several panels and to adjust it for each panel size.

Further information on faceplates can be found in the Industry Online Support(\9\).

To connect the process tags to the faceplate, click on the faceplate and go to “Interface”. Now, all process tags to be connected are displayed.

Figure 4-4 Interface for the faceplate



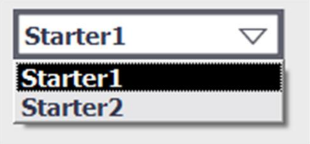
fpMotorControl_1 [Faceplate instance] [fpMotorControl V 0.0.34]

Name	Static value	Dynamization
Faceplate Interface		
AbsoluteCurrent		HMI_Interface_absoluteMotorCurrent
motorStarterOut		motorstarter_k.DataMototStarterOut
RelativeCurrent		HMI_Interface_RelativeCurrent

4.2.4 Selecting the motor starter

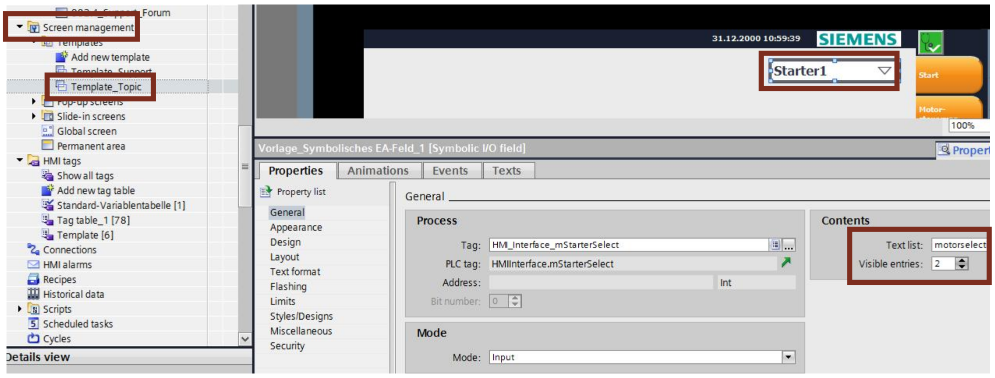
To receive the correct values of the desired motor starter, you need to select a motor starter after selecting an HMI screen. To do this, there is a dropdown menu in the upper right corner.

Figure 4-5 Selecting motor starter 1



To expand the display and integrate further motor starters into this selection field, and to adjust the names for your application, the dropdown menu can be adjusted at “Screen templates > Template_Topic > Properties.

Figure 4-6 Editing the dropdown menu



Vorlage_Symbolisches EA-Feld_1 [Symbolic I/O field]

General

Process

Tag: HMI_Interface_mStarterSelect

PLC tag: HMIInterface.mStarterSelect

Address: Int

Bit number: 0

Mode

Mode: Input

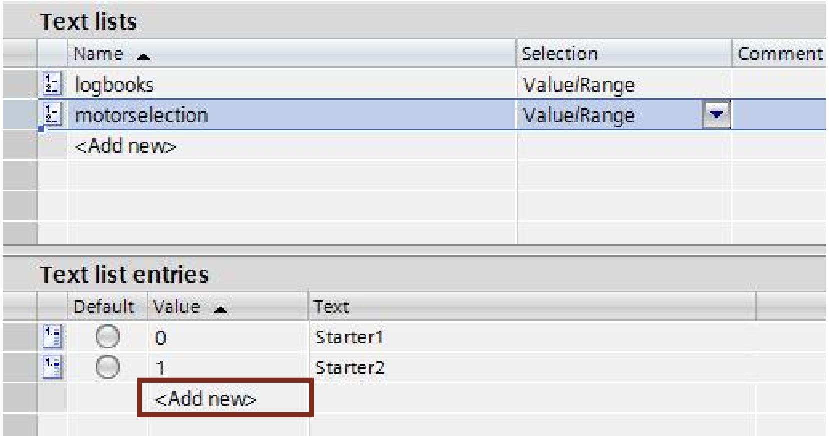
Contents

Text list: motorselect

Visible entries: 2

If you now open the text list “motorselection”, you can add further motor starters to the selection by adding new entries to the text list (“Add new”).

Figure 4-7 Expanding the “motorselection” text list



Text lists

Name ▲	Selection	Comment
logbooks	Value/Range	
motorselection	Value/Range	
<Add new>		

Text list entries

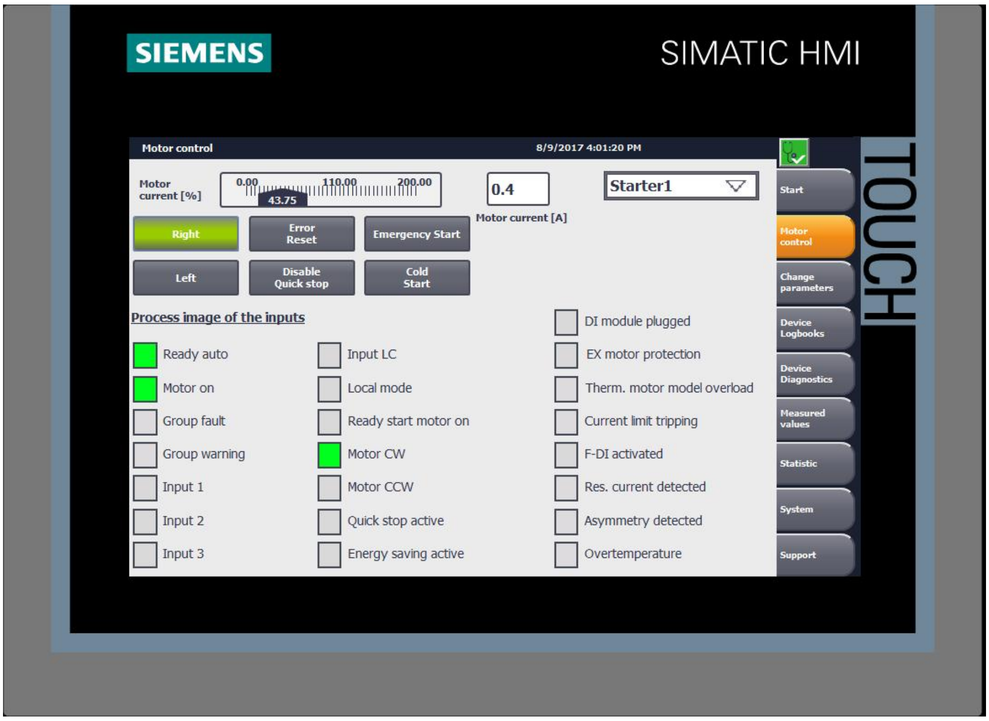
Default	Value ▲	Text
○	0	Starter1
○	1	Starter2
	<Add new>	

The value for “Starter1” = 0 and “Starter2” = 1 is stored in the “HMIInterface” DB in the “mStarterSelect” tags and interconnected at several program blocks.

4.2.5 Motor control

The following figure shows the HMI screen “Motor control”.

Figure 4-8 HMI screen “Motor control”



Motor control 8/9/2017 4:01:20 PM

Motor current [%] 0.00 110.00 200.00 43.75

0.4 Motor current [A]

Starter1

Process image of the inputs

🟩 Ready auto
🟩 Motor on
□ Group fault
□ Group warning
□ Input 1
□ Input 2
□ Input 3

□ Input LC
□ Local mode
□ Ready start motor on
🟩 Motor CW
□ Motor CCW
□ Quick stop active
□ Energy saving active

□ DI module plugged
□ EX motor protection
□ Therm. motor model overload
□ Current limit tripping
□ F-DI activated
□ Res. current detected
□ Asymmetry detected
□ Overtemperature

In the HMI screen “motor control”, the process image of the motor starter’s inputs as well as the process image of the outputs is shown. This provides you with the status of the selected motor starter and allows you to activate it manually, or reset an error. Additionally, the current motor current is displayed in % and as real value in [A] in the upper screen area.

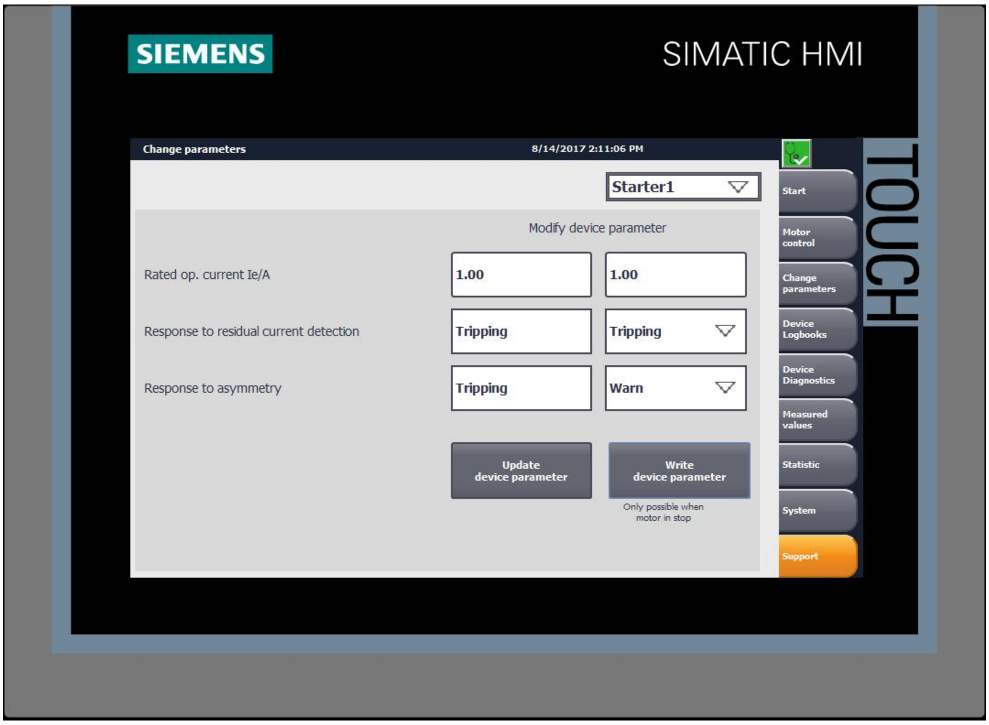
Note

The process image of the inputs and outputs are connected to the panel as multiplex tags (DataMotorstarter_k.DataMotorStarterIn, DataMotorstarter_k.DataMotorStarterOut). Please note that the “Symbolic multiplexing” is available only for TIA Portal version V 14 and higher.

4.2.6 Change parameters

The following figure shows the HMI screen “Change parameters”.

Figure 4-9 HMI screen “Change parameters”



Change parameters 8/14/2017 2:11:06 PM

Starter1

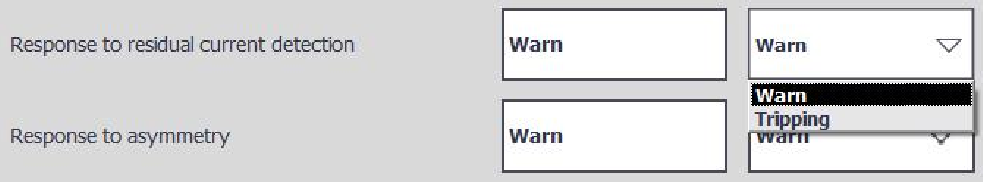
Modify device parameter

Rated op. current Ie/A	1.00	1.00
Response to residual current detection	Tripping	Tripping
Response to asymmetry	Tripping	Warn

In the HMI screen “Change parameters”, you have the option of adjusting the rated operational current I_e. The entered value of the rated operational current needs to be in the value range of the selected motor starter, otherwise, the parameter will not be accepted. This means that, for an ET 200SP motor starter DS 0.9-3A (3RK1 308-0AC00-0CP0), only values between 0.9 - 3 can be entered. Parameters cannot be written during motor operation. For a parameter change, the motor needs to be stopped and the local manual operation needs to be deactivated.

Additionally, the parameters “Response to residual current detection” and “Response to asymmetry” can be changed between “Warn” and “Tripping”.

Figure 4-10 Changing the parameters “Residual current detection” and “Asymmetry”



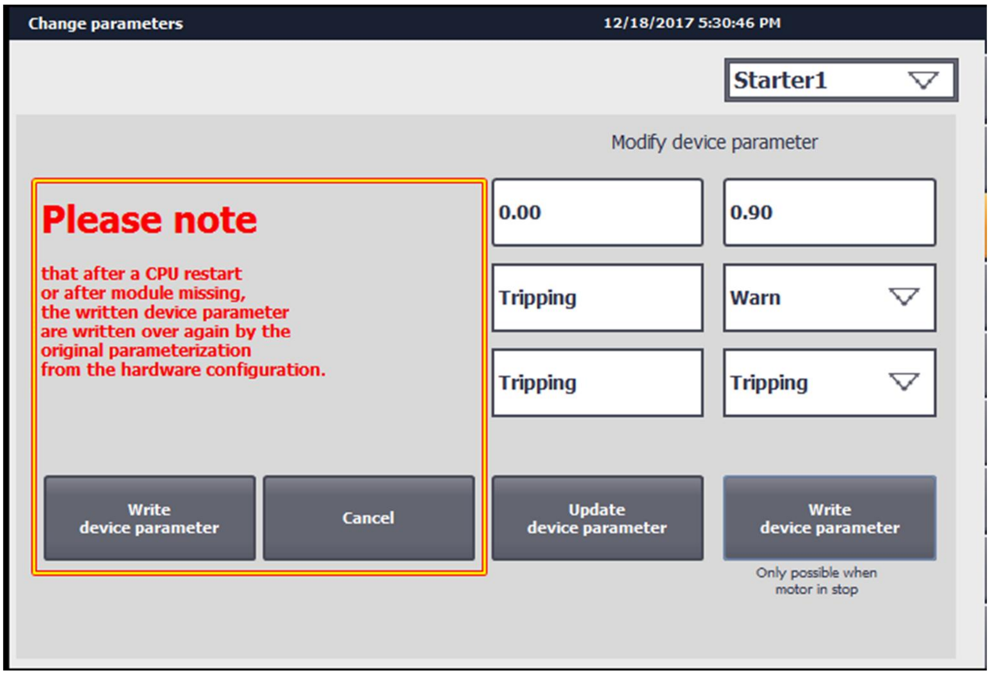
Response to residual current detection	Warn	Warn
Response to asymmetry	Warn	Warn

Note

> Please note that after a CPU restart or after module missing, the written device parameter are written over again by the original parameterization from the hardware configuration.

You will see this note if you want to write the parameter.

Abbildung 4-11



Change parameters 12/18/2017 5:30:46 PM

Starter1

Modify device parameter

Please note

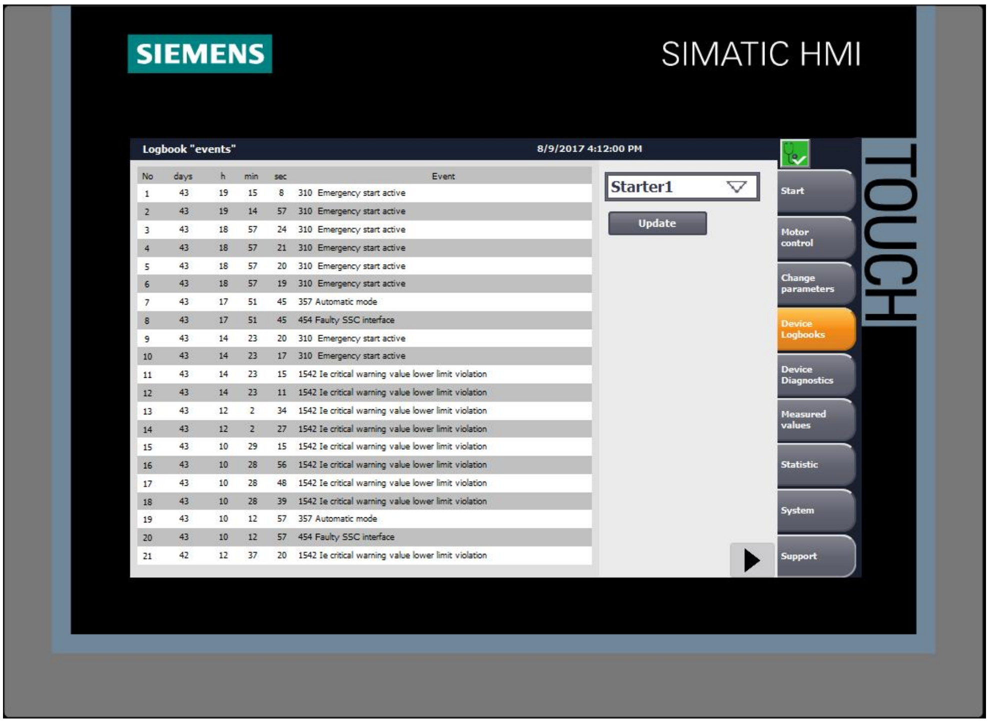
that after a CPU restart
or after module missing,
the written device parameter
are written over again by the
original parameterization
from the hardware configuration.

0.00	0.90
Tripping	Warn
Tripping	Tripping

4.2.7 Device logbooks

The following figure shows the HMI screen “Device logbooks”.

Figure 4-12 HMI screen “Device logbooks - Logbook ‘events’”



Logbook "events" 8/9/2017 4:12:00 PM

No	days	h	min	sec	Event
1	43	19	15	8	310 Emergency start active
2	43	19	14	57	310 Emergency start active
3	43	18	57	24	310 Emergency start active
4	43	18	57	21	310 Emergency start active
5	43	18	57	20	310 Emergency start active
6	43	18	57	19	310 Emergency start active
7	43	17	51	45	357 Automatic mode
8	43	17	51	45	454 Faulty SSC interface
9	43	14	23	20	310 Emergency start active
10	43	14	23	17	310 Emergency start active
11	43	14	23	15	1542 Ie critical warning value lower limit violation
12	43	14	23	11	1542 Ie critical warning value lower limit violation
13	43	12	2	34	1542 Ie critical warning value lower limit violation
14	43	12	2	27	1542 Ie critical warning value lower limit violation
15	43	10	29	15	1542 Ie critical warning value lower limit violation
16	43	10	28	56	1542 Ie critical warning value lower limit violation
17	43	10	28	48	1542 Ie critical warning value lower limit violation
18	43	10	28	39	1542 Ie critical warning value lower limit violation
19	43	10	12	57	357 Automatic mode
20	43	10	12	57	454 Faulty SSC interface
21	42	12	37	20	1542 Ie critical warning value lower limit violation

There are 3 different logbooks for the ET 200SP motor starter:

- DS72 logbook device errors
- DS73 logbook triggering operations
- DS75 logbook events

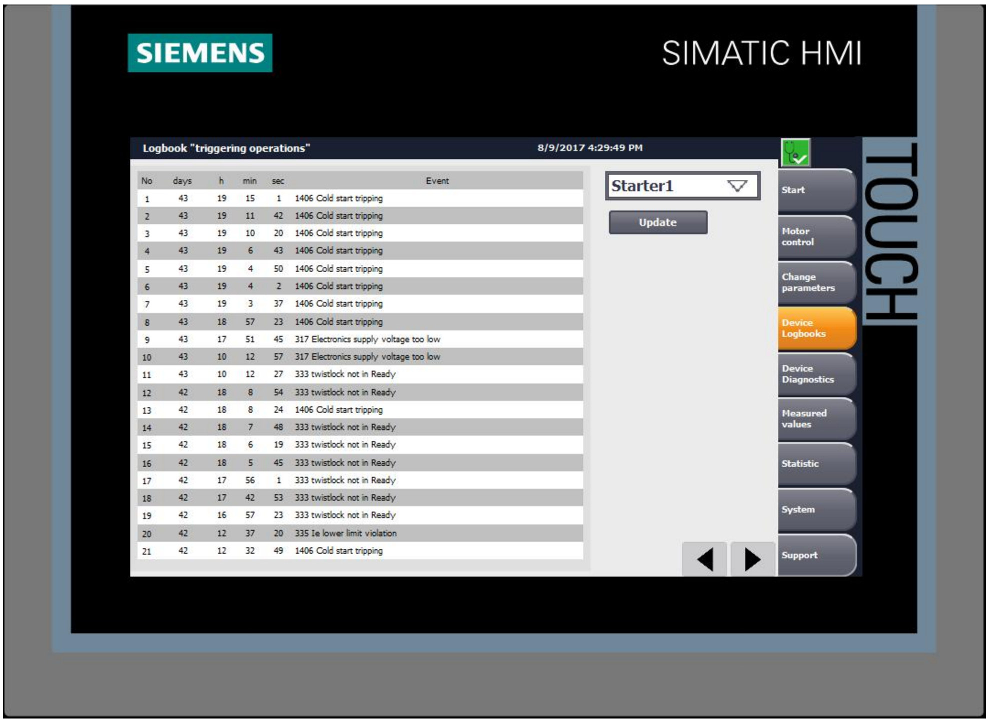
In the HMI screen “Device logbooks”, all 3 variants can be found. The logbooks can be navigated through via the arrow keys in the lower right corner. For each entered event, the time is additionally displayed in days, hours, minutes and seconds. This is the operating hours counter in the device, which means that the exact time of when an event occurred in the device will be recorded.

In the logbooks “Events” and “Triggering operations”, most entries will usually be found, which is why the view starts with the logbook “Events”, followed by the logbook “Triggering operations” and finally the logbook “Device errors”. It may well be that there are no entries in the logbook “Device errors”. Here, errors that occur in the hardware are entered.

Note The logbook entries are connected to the panel as multiplex tags. Please note that the “Symbolic multiplexing” is available only for TIA Portal version V 14 and higher.

The following figure shows the HMI screen “Device logbooks”.

Figure 4-13 HMI screen “Triggering operations”



Logbook "triggering operations"

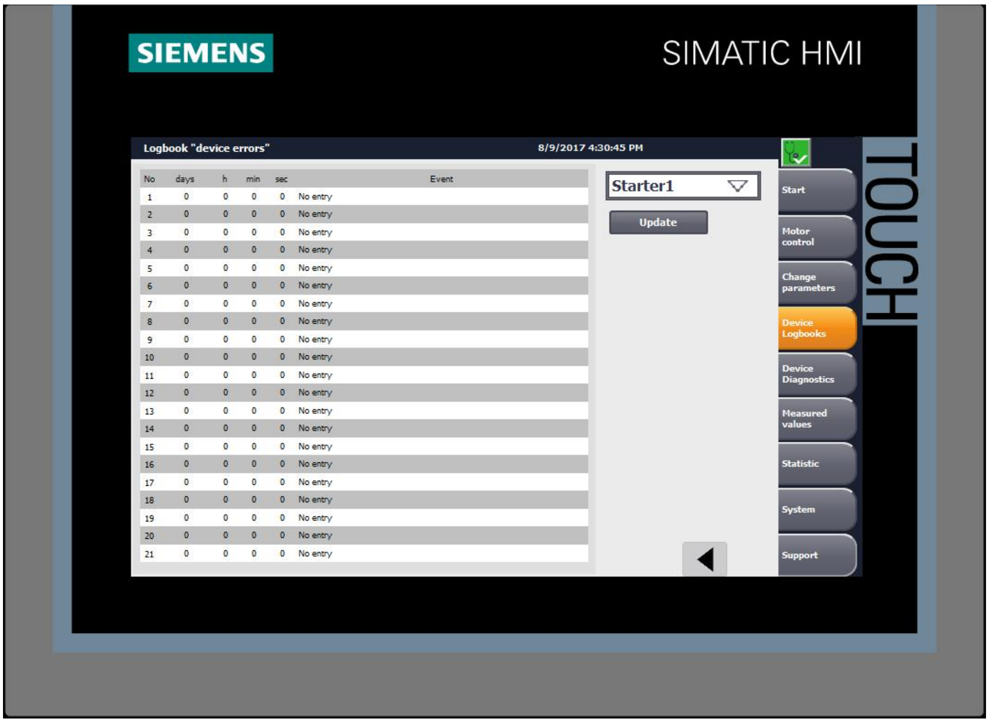
8/9/2017 4:29:49 PM

No	days	h	min	sec	Event
1	43	19	15	1	1406 Cold start tripping
2	43	19	11	42	1406 Cold start tripping
3	43	19	10	20	1406 Cold start tripping
4	43	19	6	43	1406 Cold start tripping
5	43	19	4	50	1406 Cold start tripping
6	43	19	4	2	1406 Cold start tripping
7	43	19	3	37	1406 Cold start tripping
8	43	18	57	23	1406 Cold start tripping
9	43	17	51	45	317 Electronics supply voltage too low
10	43	10	12	57	317 Electronics supply voltage too low
11	43	10	12	27	333 twistlock not in Ready
12	42	18	8	54	333 twistlock not in Ready
13	42	18	8	24	1406 Cold start tripping
14	42	18	7	48	333 twistlock not in Ready
15	42	18	6	19	333 twistlock not in Ready
16	42	18	5	45	333 twistlock not in Ready
17	42	17	56	1	333 twistlock not in Ready
18	42	17	42	53	333 twistlock not in Ready
19	42	16	57	23	333 twistlock not in Ready
20	42	12	37	20	335 Ie lower limit violation
21	42	12	32	49	1406 Cold start tripping

Starter1

The following figure shows the HMI screen “Device logbook during a device error”.

Figure 4-14 HMI screen logbook “Device errors”



Logbook "device errors"

8/9/2017 4:30:45 PM

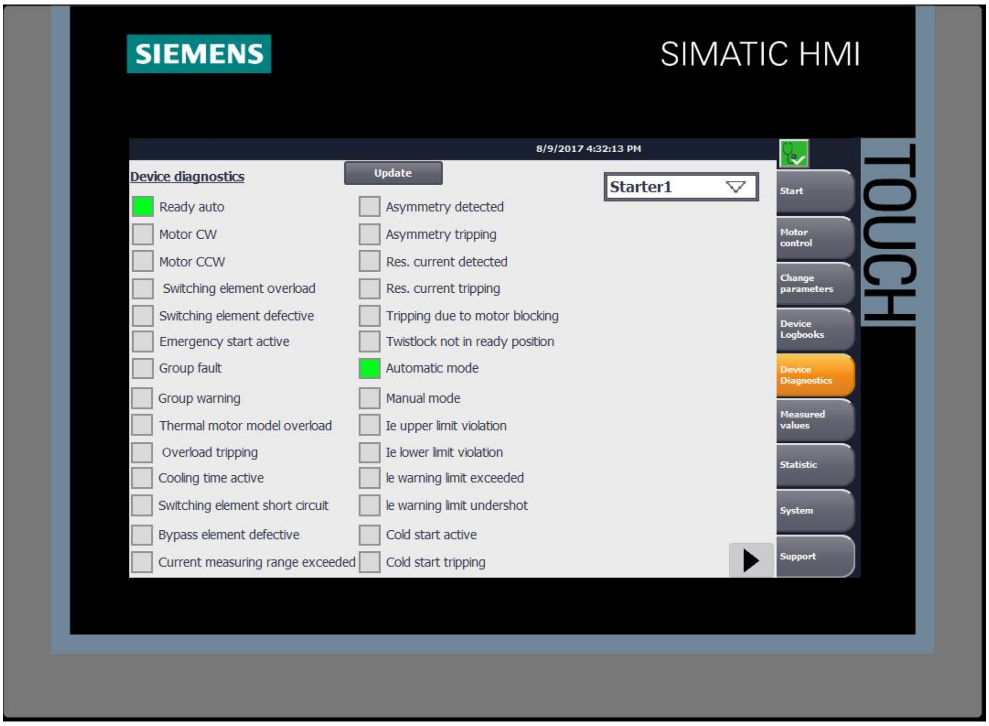
No	days	h	min	sec	Event
1	0	0	0	0	No entry
2	0	0	0	0	No entry
3	0	0	0	0	No entry
4	0	0	0	0	No entry
5	0	0	0	0	No entry
6	0	0	0	0	No entry
7	0	0	0	0	No entry
8	0	0	0	0	No entry
9	0	0	0	0	No entry
10	0	0	0	0	No entry
11	0	0	0	0	No entry
12	0	0	0	0	No entry
13	0	0	0	0	No entry
14	0	0	0	0	No entry
15	0	0	0	0	No entry
16	0	0	0	0	No entry
17	0	0	0	0	No entry
18	0	0	0	0	No entry
19	0	0	0	0	No entry
20	0	0	0	0	No entry
21	0	0	0	0	No entry

Starter1

4.2.8 Device diagnostics

The following figure shows part 1 of the HMI screen “Device diagnostics”.

Figure -4-15 HMI screen “Device diagnostics” 1



8/9/2017 4:32:13 PM

Device diagnostics

Starter1

- 🟩 Ready auto
- □ Motor CW
- □ Motor CCW
- □ Switching element overload
- □ Switching element defective
- □ Emergency start active
- □ Group fault
- □ Group warning
- □ Thermal motor model overload
- □ Overload tripping
- □ Cooling time active
- □ Switching element short circuit
- □ Bypass element defective
- □ Current measuring range exceeded
- □ Asymmetry detected
- □ Asymmetry tripping
- □ Res. current detected
- □ Res. current tripping
- □ Tripping due to motor blocking
- □ Twistlock not in ready position
- 🟩 Automatic mode
- □ Manual mode
- □ Ie upper limit violation
- □ Ie lower limit violation
- □ Ie warning limit exceeded
- □ Ie warning limit undershot
- □ Cold start active
- □ Cold start tripping

Just like the logbooks, the HMI screen “Device diagnostics” consists of 3 individual screens. Via the arrow keys in the lower right corner, you can navigate between the screens.

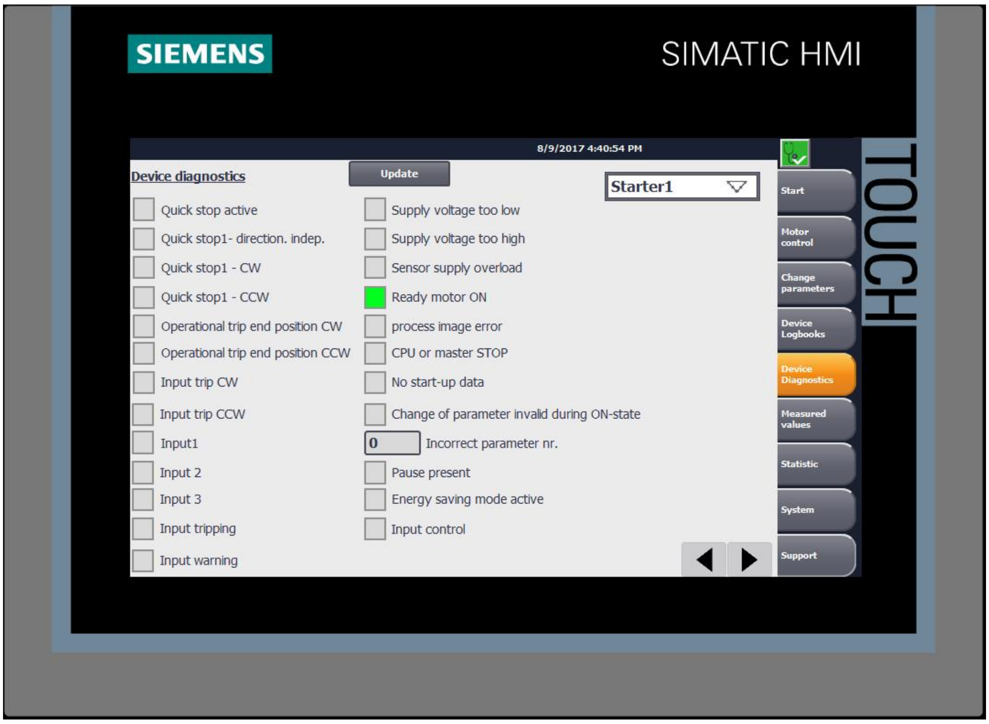
In these screens, all diagnoses integrated into the ET 200SP motor starter are displayed. In the last screen of the “Device diagnostics”, the diagnosis of the ET 200SP F-motor starter will be specifically addressed.

Note

The process value of the device diagnosis is connected to the panel as multiplex tag (DataMotorstarter_k.DataDS92). Please note that the “Symbolic multiplexing” is available only for TIA Portal version V 14 and higher.

The following figure shows part 2 of the HMI screen “Device diagnostics”.

Figure -4-16 HMI screen “Device diagnostics” 2



Device diagnostics

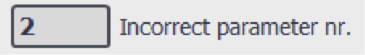
Starter1

- □ Quick stop active
- □ Quick stop1- direction. indep.
- □ Quick stop1 - CW
- □ Quick stop1 - CCW
- □ Operational trip end position CW
- □ Operational trip end position CCW
- □ Input trip CW
- □ Input trip CCW
- □ Input1
- □ Input 2
- □ Input 3
- □ Input tripping
- □ Input warning

- □ Supply voltage too low
- □ Supply voltage too high
- □ Sensor supply overload
- 🟩 Ready motor ON
- □ process image error
- □ CPU or master STOP
- □ No start-up data
- □ Change of parameter invalid during ON-state
- 0 Incorrect parameter nr.
- □ Pause present
- □ Energy saving mode active
- □ Input control

From the DS92, a faulty parameter number can also be read. Here, numerical values that reflect object numbers are entered. The interpretation can be done with the device manual of the ET 200SP motor starter (\5\).

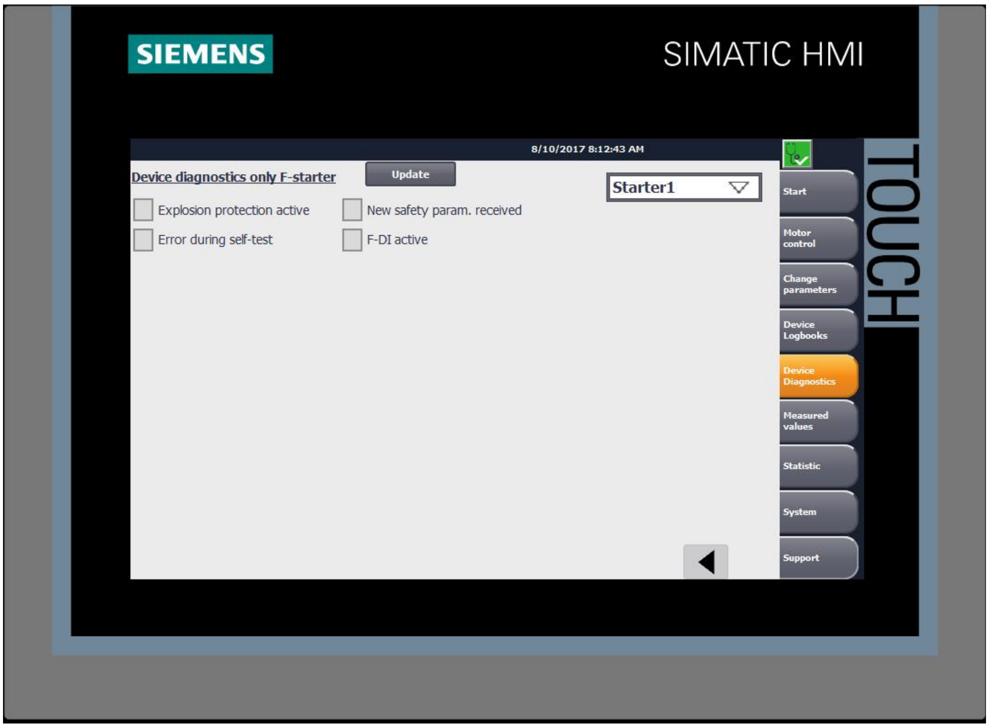
Figure 4-17 Output of a faulty parameter number



2 Incorrect parameter nr.

The following figure shows the HMI screen “Device diagnostics” of the F-motor starter.

Figure -4-18 HMI screen “Device diagnostics F-motor starter”



8/10/2017 8:12:43 AM

Device diagnostics only F-starter

Starter1

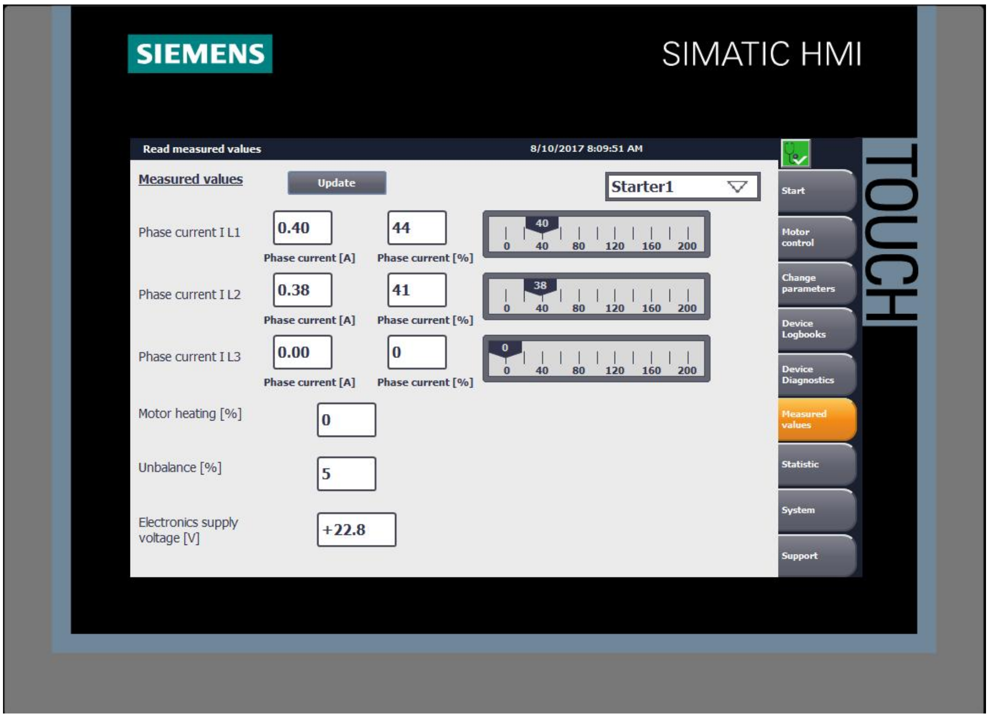
□ Explosion protection active □ New safety param. received

□ Error during self-test □ F-DI active

4.2.9 Measured values

The following figure shows the HMI screen “Measured values”.

Figure 4-19 HMI screen “Read measured values”



Read measured values

8/10/2017 8:09:51 AM

Measured values

Starter1

Phase current I L1: 0.40 Phase current [A]; 44 Phase current [%]; 40

Phase current I L2: 0.38 Phase current [A]; 41 Phase current [%]; 38

Phase current I L3: 0.00 Phase current [A]; 0 Phase current [%]; 0

Motor heating [%]: 0

Unbalance [%]: 5

Electronics supply voltage [V]: +22.8

In the HMI screen “Measured values”, the information from the data set 94 are displayed. The different phase currents can be viewed in % as well as in [A] so that you can assess the occupancy rate of the individual subordinate actuators.

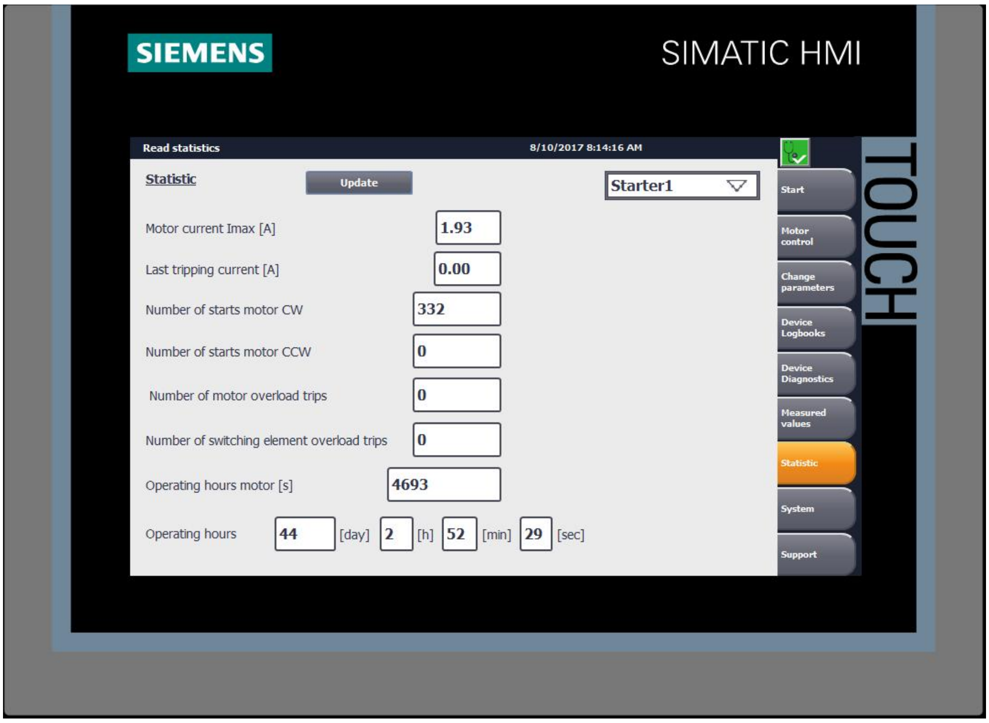
The displayed data are not displayed cyclically, however, they can be updated via the “Update” button.

Note If you integrate a cyclic evaluation of the data into your project, please note that this affects the cycle time of the OB1.

4.2.10 Statistics

The following figure shows the HMI screen “Statistics”.

Figure 4-20 HMI screen “Read statistics”



Read statistics 8/10/2017 8:14:16 AM

Statistic Starter1

Motor current Imax [A] 1.93

Last tripping current [A] 0.00

Number of starts motor CW 332

Number of starts motor CCW 0

Number of motor overload trips 0

Number of switching element overload trips 0

Operating hours motor [s] 4693

Operating hours 44 [day] 2 [h] 52 [min] 29 [sec]

In the HMI screen “Statistics”, the information from data set 95 are displayed. It is evaluated, how often the motor starter has been activated (in clockwise or anti-clockwise rotation), how high the maximum motor current in [A] is, and different tripping conditions are also counted. The operating hours counter of the device is specified in days, hours, minutes and seconds.

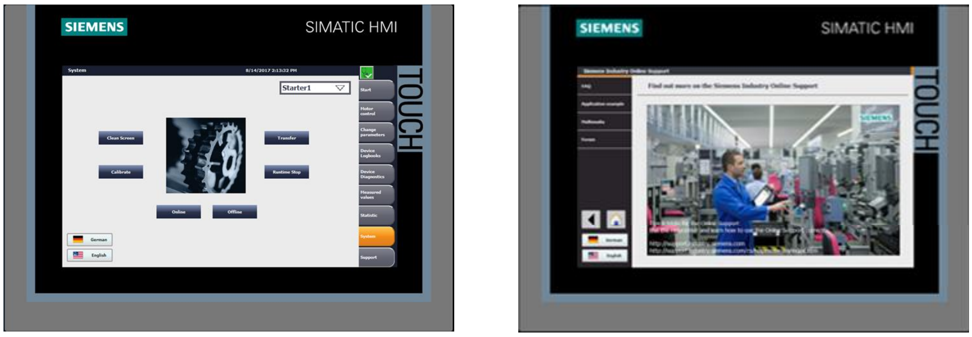
The displayed data are not displayed cyclically, however, they can be updated via the “Update” button.

Note If you integrate a cyclic evaluation of the data into your project, please note that this affects the cycle time of the OB1.

4.2.11 System & support

The following figure shows the HMI screens “System” and “Support”.

Figure 4-21 HMI screens "System" and "Support"



The HMI screens "System" and "Support” are integrated into the project as assistance. No functions of the ET 200SP motor starter are displayed in these screens.

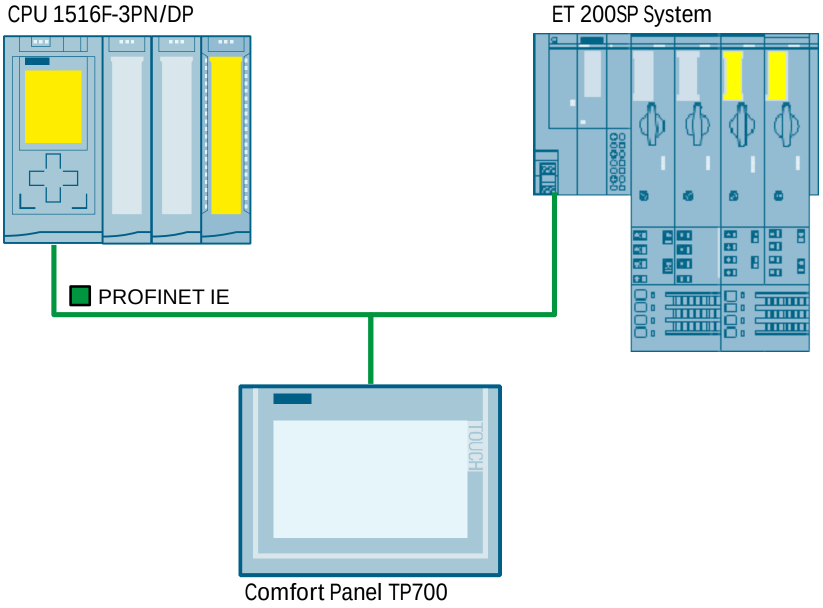
Under “Support” and “System”, you also have the option of switching between the German and English language.

5 Other setup options

The setup shown in this application example is not mandatory. Your system may vary from this setup and can be structured individually. Two possible variants are described below:

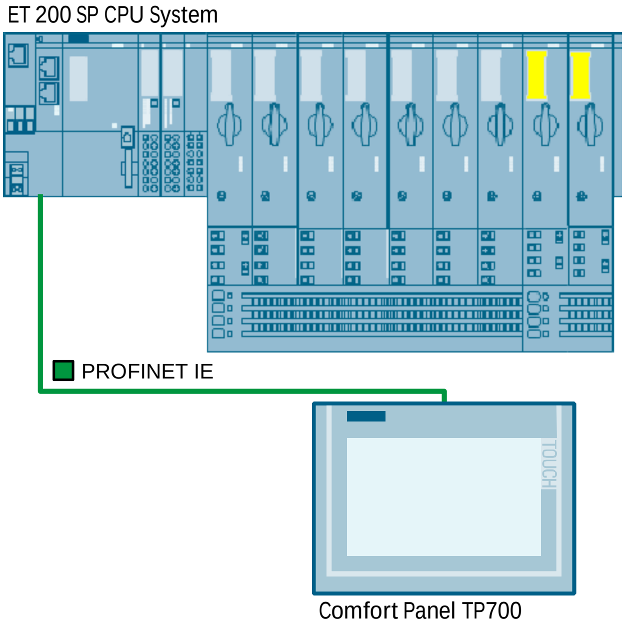
The following figure shows a variant with an S7-1500 F-CPU as well as standard and failsafe motor starters

Figure 5-1 Example: Standard and Failsafe ET 200SP motor starters with S7-1500 F-CPU



The following figure shows a variant with an ET 200SP CPU as well as standard and failsafe motor starters

Figure 5-2 Example: Standard and failsafe ET 200SP motor starters centrally on ET 200SP CPU



6 Appendix

6.1 Service & support

Industry Online Support

Do you have any questions or need support?

Siemens Industry Online Support offers access to our entire service and support know-how as well as to our services.

Siemens Industry Online Support is the central address for information on our products, solutions and services.

Product information, manuals, downloads, FAQs and application examples – all information is accessible with just a few mouse clicks at https://support.industry.siemens.com

Technical Support

Siemens Industry's Technical Support offers quick and competent support regarding all technical queries with numerous tailor-made offers – from basic support right up to individual support contracts.

Please address your requests to the Technical Support via the web form: www.siemens.com/industry/supportrequests

Service offer

Our service offer comprises, among other things, the following services:

- Product Training
- Plant Data Services
- Spare Parts Services
- Repair Services
- On Site and Maintenance Services
- Retrofit and Modernization Services
- Service Programs and Agreements

Detailed information on our service offer is available in the Service Catalog: https://support.industry.siemens.com/cs/sc

Industry Online Support app

Thanks to the "Siemens Industry Online Support" app, you will get optimum support even when you are on the move. The app is available for Apple iOS, Android and Windows Phone. https://support.industry.siemens.com/cs/ww/en/sc/2067

6.2 Links and literature

Table 1-6

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to the entry page of the application example https://support.industry.siemens.com/cs/ww/de/view/109750305
\3\	Limitation for the active jobs https://support.industry.siemens.com/cs/de/en/view/15364459
\4\	Application example for ET 200SP Motor starter Safe Shutdown of ET 200SP Motor Starters https://support.industry.siemens.com/cs/ww/en/view/109748128
\5\	Device Manual - SIMATIC ET 200SP Motor Starter https://support.industry.siemens.com/cs/ww/en/view/109479973
\6\	Libraries with PLC data types (LPD) for STEP 7 (TIA Portal) and S7-1200/S7-1500 https://support.industry.siemens.com/cs/ww/en/view/109482396
\7\	SIMATIC ET 200SP Distributed I/O System https://support.industry.siemens.com/cs/ww/en/view/58649293
\8\	Description for “MillisecToTime” FB. https://support.industry.siemens.com/cs/ww/en/view/42637538
\9\	Using faceplates. https://support.industry.siemens.com/cs/ww/en/view/68014632

6.3 Change documentation

Table 6-1

Version	Date	Modifications
V1.0	09/2017	First version
V1.1	12/2017	Add note in HMI Change of Faceplate and Screen "Change parameter" and FB "RdRecWrRec" and DB "HMIInterface".