

SIEMENS

SIMOTION

Supplement for the ET 200S frequency converter

Function Manual

Preface

Fundamental safety
instructions

1

Description

2

Programming

3

Parameterization

4

Application example

5

Alarm processing

6

Appendix

A

04/2014

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

 DANGER
indicates that death or severe personal injury will result if proper precautions are not taken.

 WARNING
indicates that death or severe personal injury may result if proper precautions are not taken.

 CAUTION
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

 WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Preface

Contents of this Function Manual

This **document** is a component of the **SIMOTION Programming - References** documentation package.

This documentation is a supplement to the SIMATIC *ET 200S FC Frequency Converter Operating Instructions (03/2005 Edition)*.

This documentation is supplied in electronic form with SIMOTION SCOUT.

This manual describes how you can control the ET 200S frequency converter (order no.: 6SL3244-0SA00-1AA0) from a SIMOTION device using the **_ET200S_FC_control** function block.

Function block

The **_ET200S_FC_control** function block is used for the communication between the SIMOTION system and the ET 200S frequency converter and is a component of the command library of the "SIMOTION SCOUT" engineering system.

SIMOTION Documentation

An overview of the SIMOTION documentation can be found in the SIMOTION Documentation Overview document.

This documentation is included as electronic documentation in the scope of delivery of SIMOTION SCOUT. It comprises ten documentation packages.

The following documentation packages are available for SIMOTION V4.4:

- SIMOTION Engineering System Handling
- SIMOTION System and Function Descriptions
- SIMOTION Service and Diagnostics
- SIMOTION IT
- SIMOTION Programming
- SIMOTION Programming - References
- SIMOTION C
- SIMOTION P
- SIMOTION D
- SIMOTION Supplementary Documentation

Hotline and Internet addresses

Additional information

Click the following link to find information on the the following topics:

- Ordering documentation / overview of documentation
- Additional links to download documents
- Using documentation online (find and search manuals/information)

<http://www.siemens.com/motioncontrol/docu>

Please send any questions about the technical documentation (e.g. suggestions for improvement, corrections) to the following e-mail address:
docu.motioncontrol@siemens.com

My Documentation Manager

Click the following link for information on how to compile documentation individually on the basis of Siemens content and how to adapt it for the purpose of your own machine documentation:

<http://www.siemens.com/mdm>

Training

Click the following link for information on SITRAIN - Siemens training courses for automation products, systems and solutions:

<http://www.siemens.com/sitrain>

FAQs

Frequently Asked Questions can be found in SIMOTION Utilities & Applications, which are included in the scope of delivery of SIMOTION SCOUT, and in the Service&Support pages in **Product Support**:

<http://support.automation.siemens.com>

Technical support

Country-specific telephone numbers for technical support are provided on the Internet under **Contact**:

<http://www.siemens.com/automation/service&support>

Table of contents

	Preface.....	3
1	Fundamental safety instructions.....	7
1.1	General safety instructions.....	7
1.2	Industrial security.....	8
2	Description.....	9
2.1	General.....	9
2.2	Product description.....	10
2.3	Installation and connection.....	12
2.4	Inserting the ET 200S frequency converter into a SIMOTION project.....	13
3	Programming.....	15
3.1	Integrating the function block in the user project.....	15
3.2	Addressing the ET 200S frequency converter.....	17
4	Parameterization.....	19
4.1	Function block _ET200S_FC_control.....	19
4.2	Calling the function block.....	25
5	Application example.....	29
5.1	General.....	29
5.2	Sequence of the application example.....	31
6	Alarm processing.....	35
6.1	Overview.....	35
6.2	Diagnostic alarms.....	37
A	Appendix.....	39
A.1	SIMOTION and SIMATIC names.....	39
A.2	List of abbreviations.....	40
	Index.....	41

Fundamental safety instructions

1.1 General safety instructions

 WARNING
Risk of death if the safety instructions and remaining risks are not carefully observed If the safety instructions and residual risks are not observed in the associated hardware documentation, accidents involving severe injuries or death can occur. • Observe the safety instructions given in the hardware documentation. • Consider the residual risks for the risk evaluation.

 WARNING
Danger to life or malfunctions of the machine as a result of incorrect or changed parameterization As a result of incorrect or changed parameterization, machines can malfunction, which in turn can lead to injuries or death. • Protect the parameterization (parameter assignments) against unauthorized access. • Respond to possible malfunctions by applying suitable measures (e.g. EMERGENCY STOP or EMERGENCY OFF).

1.2 Industrial security

Note

Industrial security

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens' products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit <http://www.siemens.com/industrialsecurity>.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit <http://support.automation.siemens.com>



WARNING

Danger as a result of unsafe operating states resulting from software manipulation

Software manipulation (e.g. by viruses, Trojan horses, malware, worms) can cause unsafe operating states to develop in your installation which can lead to death, severe injuries and/or material damage.

- Keep the software up to date.
Information and newsletters can be found at:
<http://support.automation.siemens.com>
- Incorporate the automation and drive components into a state-of-the-art, integrated industrial security concept for the installation or machine.
For more detailed information, go to:
<http://www.siemens.com/industrialsecurity>
- Make sure that you include all installed products into the integrated industrial security concept.

Description

2.1 General

Overview

This manual describes the differences in the operation of the ET 200S FC (FC = Frequency Converter) in the SIMOTION system compared with operation in the SIMATIC system.

Note

This manual is a supplement to the SIMATIC *ET 200S FC Frequency Converter* Operating Instructions (03/2005 Edition).

This documentation is supplied in electronic form with SIMOTION SCOUT.

Requirements

The following are required for the standard function described in this document:

- SIMOTION SCOUT, V4.1 SP1 or higher
- SIMOTION Kernel, V4.1 SP1 or higher
- ET 200S frequency converter with ICU24, software release V1.02 or higher

Note

The functions associated with the **_ET200S_FC_control** function block can only be used on the ET 200S frequency converter with ICU (order no.: 6SL3244-0SA00-1AA0).

2.2 Product description

ET 200S frequency converter

The ET 200S FC is a modular frequency converter that is completely embedded in the distributed I/O system of the ET 200S.

During generator operation of the connected motor, the ET 200S FC feeds back energy into the supply system.

Function block

The **_ET200S_FC_control** function block is provided for the cyclic communication and the parameter transfer of the ET 200S FC with SIMOTION. This function block is described in this manual.

Note

The **_ET200S_FC_control** function block **cannot** be used to control the ET 200S FC (order no.: 6SL3244-0SA00-1AA1). Standard telegram 1 should be used for this frequency converter.

Functionality

The functionality of the **_ET200S_FC_control** function block and the ET 200S FC is the same as when used in the SIMATIC S7 automation system. There are differences in the execution of the data transfer and the handling of the FB.

Possible applications

The ET 200S FC can be operated in a distributed ET 200S station via PROFIBUS DP on a SIMOTION controller.

Several ET 200S frequency converters can be operated in one ET 200S station. The ET 200S interface module IM 151-1 is used for the connection to PROFIBUS DP.

The following figure shows a distributed configuration with SIMOTION C2xx.

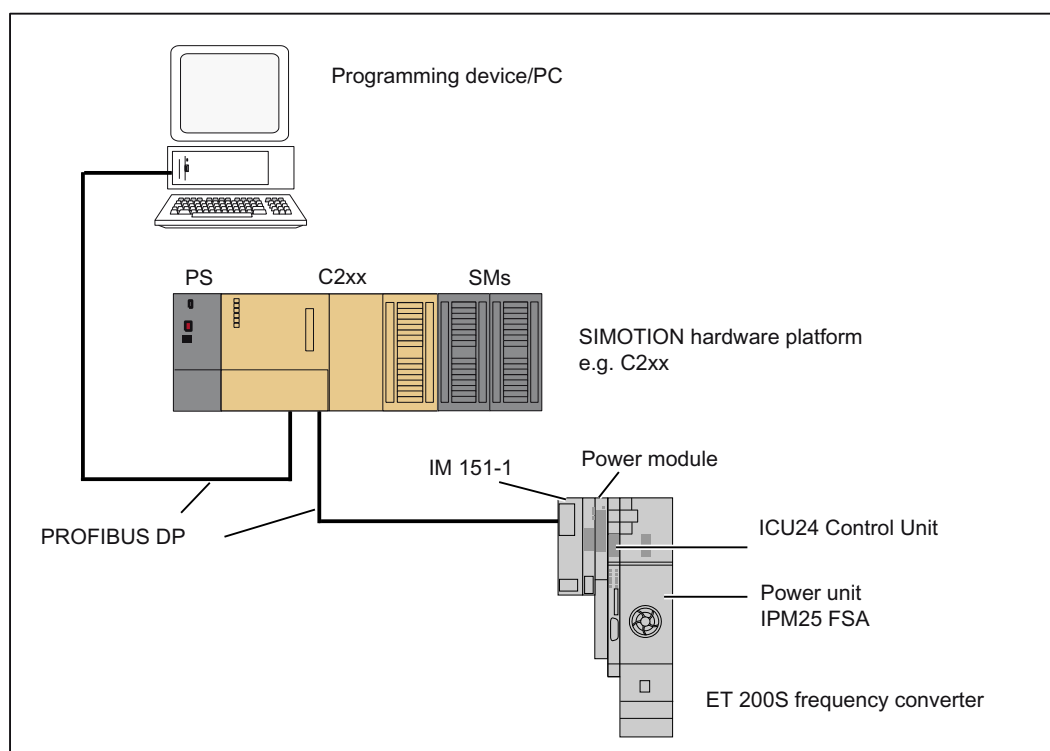


Figure 2-1 Connection of the ET 200S frequency converter to the SIMOTION C2xx device (example)

2.3 Installation and connection

You must perform the following steps to be able to operate the ET 200S frequency converter using the SIMOTION system:

1. Install and connect the ET 200S interface module (IM) with the components of the ET 200S FC.
2. Establish the PROFIBUS connection between the IM and the SIMOTION device.
3. Set the PROFIBUS DP node address on the IM.
4. Switch on the terminating resistor at the first and last bus node.

Note

The commissioning of the ET 200S frequency converter can be performed either via

- the STARTER functionality integrated in the SIMOTION SCOUT engineering system (via "Insert single drive"), or
- or the STARTER engineering software.

For a description of commissioning and steps 1 to 4, see the *SIMATIC ET 200S FC Frequency Converter Operating Instructions* (03/2005 Edition).

5. Insert the ET 200S FC into the SIMOTION project (refer to Chapter Inserting the ET 200S frequency converter into a SIMOTION project (Page 13)).
6. Parameterizing the ET 200S FC.
To parameterize the module, go to **HW Config**. Double-click the relevant module to open the **Properties** dialog box. Select the **Parameters** tab to access the parameterization interface. There you can assign parameters for your module.
See the *SIMATIC ET 200S FC Frequency Converter Operating Instructions* (03/2005 Edition).
7. Link the function block to the SIMOTION project (refer to Chapter Integrating the function block in the user project (Page 15)).

2.4 Inserting the ET 200S frequency converter into a SIMOTION project

Requirements

The following requirements must be met:

1. You have created a project in SIMOTION SCOUT and inserted a rack with a SIMOTION device in the hardware configuration.
2. You have configured a PROFIBUS subnet.

Note

Consult the *SIMOTION SCOUT* online help to learn how to create a project and configure a PROFIBUS subnet.

Inserting the frequency converter

1. In SIMOTION SCOUT, open the "User Projects" dialog box with the "Project" > "Open" menu command. In this dialog box, select your project and confirm with "OK".
2. Open **HW Config**.
3. In the "HW Config" window, open the **hardware catalog** with the "View" > "Catalog" menu command.
4. In the hardware catalog, open the "PROFIBUS DP" folder and the "ET 200S" subfolder and select, for example, the **IM 151-1 Standard** interface module (order no.: 6ES7 151-1AA04-0AB0 or follow-on module).
5. Use drag-and-drop to move the "IM 151-1 Standard" interface module onto the PROFIBUS subnet of your SIMOTION device.
The "Properties - PROFIBUS IM 151-1 Standard Interface" dialog box opens. In this dialog box, select the address you set on the IM 151-1 (see *ET 200S Distributed I/O Device Manual*) and confirm your selection with "OK".
The selected IM 151-1 Standard I/O device is inserted into the project.
6. The inserted I/O device still has to be equipped with the following modules: PM-D power module and ET 200S FC frequency converter, consisting of ICU24 closed-loop control module.
Open the "Motor starter" folder and the "PM" subfolder under "IM 151-1 Standard" in the hardware catalog. Select the **PM-D 24VDC** power module.
Insert the frequency converter by selecting the **ICU24** closed-loop control module (order no. 6SL3244-0SA00-1AA0) in the "Frequency converter" folder under "IM151-1 Standard" (order no.: 6ES7 151-1AA04-0AB0) in the hardware catalog.

Note

You can disable or enable diagnostic alarms via parameter p8452 in the ET 200S frequency converter.

7. Save and compile your project.

Programming

3.1 Integrating the function block in the user project

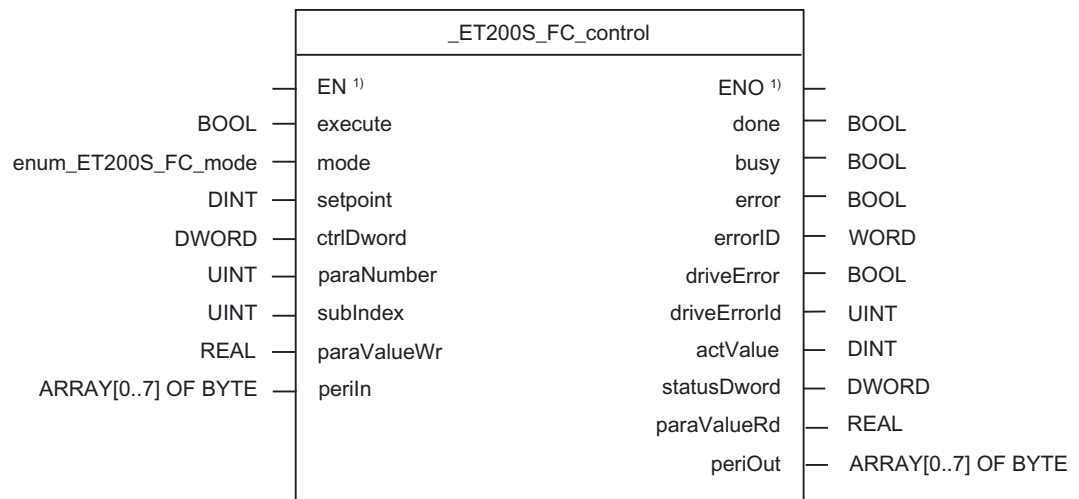
Creating the FB instance in the user project

The **_ET200S_FC_control** function block is a component of the command library of the SIMOTION SCOUT engineering system. To work with the block, an instance of the function block must first be created in the user project.

Example:

```
VAR_GLOBAL
...
    myFC_control      : _ET200S_FC_control;    // create instance of FB _ET200S_FC_control
...
END_VAR
```

Call (LAD representation)



1) LAD-specific parameters

Application example

The application example is included on the "SIMOTION Utilities & Applications" CD-ROM and is available for various SIMOTION hardware platforms.

3.1 Integrating the function block in the user project

The "SIMOTION Utilities & Applications" CD-ROM is provided free of charge with SIMOTION SCOUT.

3.2 Addressing the ET 200S frequency converter

Overview

Communication between the SIMOTION device and the ET 200S FC takes place through direct access to the I/O. I/O variables are used to address the direct read/write access to the I/O.

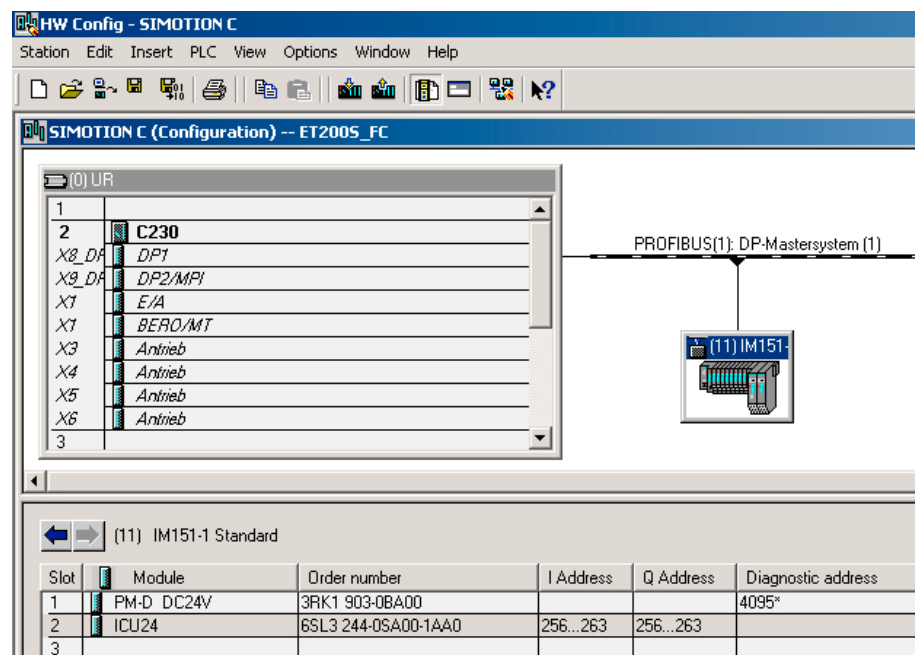
Creating I/O variables

For the cyclic data transfer with the ET 200S frequency converter, you must create an I/O variable in the symbol browser for each I/O input and output. When creating the I/O variable, enter the configured address from the hardware configuration in the "I/O address" column (see example below). Enter a field length of 8 in the "Field length" column.

With these specifications, an I/O variable of type ARRAY[0..7] of BYTE is created in the SIMOTION project for cyclic data transfer.

Example:

Determining addresses from HW Config:



Slot	Module	Order number	I Address	Q Address	Diagnostic address
1	PM-D DC24V	3RK1 903-0BA00			4095*
2	ICU24	6SL3 244-0SA00-1AA0	256...263	256...263	
3					

Figure 3-1 Addresses in HW Config

Creating the I/O variables in the symbol browser:

	Name	I/O address	Read only	Data type	Field length
1	myperiIn	PIB 256		Array	8
2	myperiOut	PQB 256		Array	8

Figure 3-2 Addressing with I/O variable

The I/O addresses in the symbol browser shown in the figure have the following meanings:

- "PIB" = Peripheral Input Byte
- "PQB" = Peripheral Output Byte

Parameter transfer

The I/O variable for the I/O inputs is transferred to the **periIn** input parameter.

The prepared data for the I/O outputs are supplied by the function block at the **periOut** output parameter. The **periOut** output parameter must be assigned to the I/O variable for the I/O outputs.

Note

When booting the firmware of the ET 200S frequency converter from the MMC card, it may take more than 500 ms for the ET 200S frequency converter to ramp up.

To avoid I/O area access errors when the frequency converter has not yet ramped up, accesses to the I/O inputs and outputs of the frequency converter must be performed with the **_getSafeValue** and **_setSafeValue** system functions (see sample program **UNIT E_FC_ctr**).

Note

For additional information, refer to:

- SIMOTION SCOUT online help
- Programming Manuals for the relevant programming languages, such as:
 - SIMOTION ST, Structured Text Programming Manual
 - SIMOTION MCC, Motion Control Chart Programming Manual
 - SIMOTION LAD/FBD, Ladder Diagram and Function Block Diagram Programming Manual

These documents are shipped with SIMOTION SCOUT in electronic form.

Parameterization

4.1 Function block `_ET200S_FC_control`

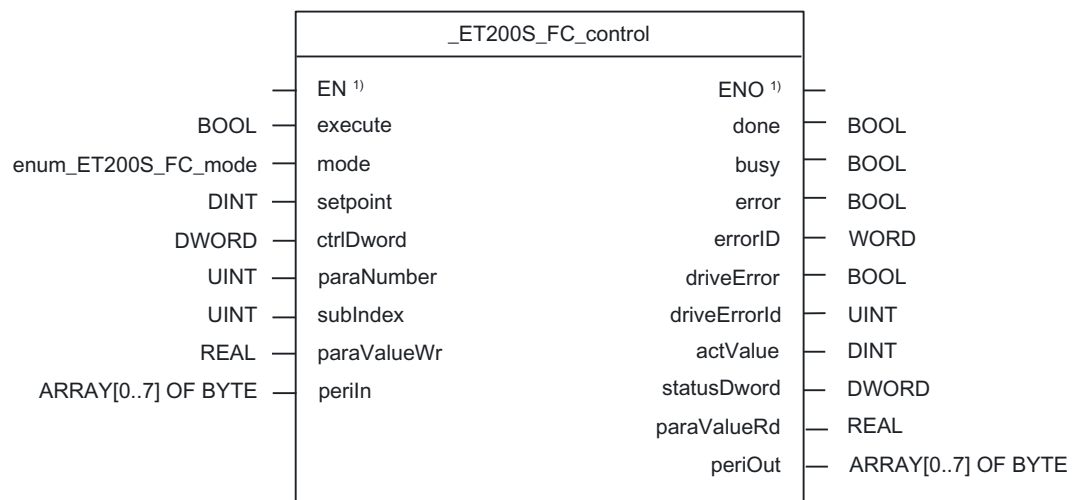
Task

You can control the ET 200S FC via your user program with the FB `_ET200S_FC_control`.

The FB `_ET200S_FC_control` supports the following functions:

- Drive control:
 - Sending control words
 - Receiving status words
- Diagnostics:
 - Displaying drive faults
 - Acknowledging drive errors
 - Displaying parameterization errors in the data exchange
- Reading and writing of drive parameters

Call (LAD representation)



1) LAD-specific parameters

Parameter description

The following table contains all the parameters of the function block `_ET200S_FC_control`.

Table 4-1 Parameters of the FB `_ET200S_FC_control`

Name	Type ¹⁾	Data type	Default	Meaning
execute	IN	BOOL	FALSE	Start of a data transfer from/to the ET 200S FC with rising edge
mode	IN	enum_ET200S_FC_mode (READ, WRITE, DATA1, DATA2, RESET_ERR)	READ	Selection of data transfer type • READ: Read individual parameter • WRITE: Write individual parameter • DATA1: Data set 1 is set active (p0810 to 0.0) with rising edge on parameter execute. • DATA2: Data set 2 is set active (p0810 to 1.0) with rising edge on parameter execute. • RESET_ERR: Acknowledgement of parameterization and drive errors
perIn	IN	ARRAY [0 to 7] of BYTE	8 (16#00)	Transfer I/O inputs of ET 200S FC to FB
setpoint	IN	DINT	0	Normalized setpoint without units for the ET 200S FC, transferred cyclically ⁴⁾
ctrlDword	IN	DWORD	16#00000000	Control words (STW 1 and STW 2) of the ET 200S FC, transferred cyclically ⁵⁾
paraNumber	IN	UINT	0	Parameter number to be read or written
subIndex	IN	UINT	0	Index of an indexed parameter for the ET 200S FC
paraValueWr	IN	REAL	0.0	Parameter value to be written
done	OUT	BOOL	FALSE	TRUE = Read/write parameter was processed without errors
busy	OUT	BOOL	FALSE	TRUE = Read/write parameter active
error	OUT	BOOL	FALSE	TRUE = parameterization error
errorID	OUT	WORD	16#0000	Error number of the parameterization error (error number 0..108, or 16#0000..16#006C) ³⁾ Note: If error = TRUE and errorID = 16#0000 , the following error is pending: "Invalid parameter number"
driveError	OUT	BOOL	FALSE	TRUE = Drive error is pending (refer to the driveErrorId parameter)
driveErrorId	OUT	UINT	0	Reason for the error (decimal format) Value corresponds to parameter r0947 (errors) ²⁾
periOut	OUT	ARRAY [0 to 7] of BYTE	8 (16#00)	Prepared FB data for I/O outputs of the ET 200S FC
actValue	OUT	DINT	0	Display of the parameter value of p2051.1 (process data on field bus) ²⁾
statusDword	OUT	DWORD	16#00000000	Status words (ZSW 1 and ZSW 2) of the ET 200S FC ²⁾
paraValueRd	OUT	REAL	0.0	Read parameter value

¹⁾ Parameter types: IN = input parameter, OUT = output parameter

²⁾ See the *SIMATIC ET 200S FC Frequency Converter* Parameter Manual (03/2005 Edition).

³⁾ See the *SIMATIC ET 200S FC Frequency Converter* Operating Instructions (03/2005 Edition).

⁴⁾ Calculation in the converter:

$$\text{Converter set frequency} = \frac{\text{Value in setpoint} \cdot \text{reference frequency in drive parameter p2000}}{\text{Normalizing value}}$$

Example 1: - Normalizing value = 16,384 (4000 hex)

- p2000 = 50 Hz

- Desired set frequency = 20 Hz

$$\text{Setpoint} = \frac{20 \text{ Hz} \cdot 16,384}{50 \text{ Hz}} = 6,553$$

Example 2: - Normalizing value = 16,384 (4000 hex)

- Speed (50 Hz Δ max. speed Δ 1,350 rpm)

- Desired speed = 1,000 rpm

$$\text{Setpoint} = \frac{1,000 \text{ rpm} \cdot 16,384}{1,350 \text{ rpm}} = 12,136$$

Note

You can find the normalizing value in the *SIMATIC ET 200S FC Frequency Converter* Parameter Manual (03/2005 Edition) under parameter p2000.

The **execute** parameter is only used to start acyclic data transfer from/to the frequency converter.

When the frequency converter is switched on, the enables must be set in the control word and the setpoint (**setpoint**) takes immediate effect.

The effect of the control word depends on the parameterization in the frequency converter.

⁵⁾ Bit assignment of **ctrlDword** (control words 1 and 2 for the ET 200S FC):

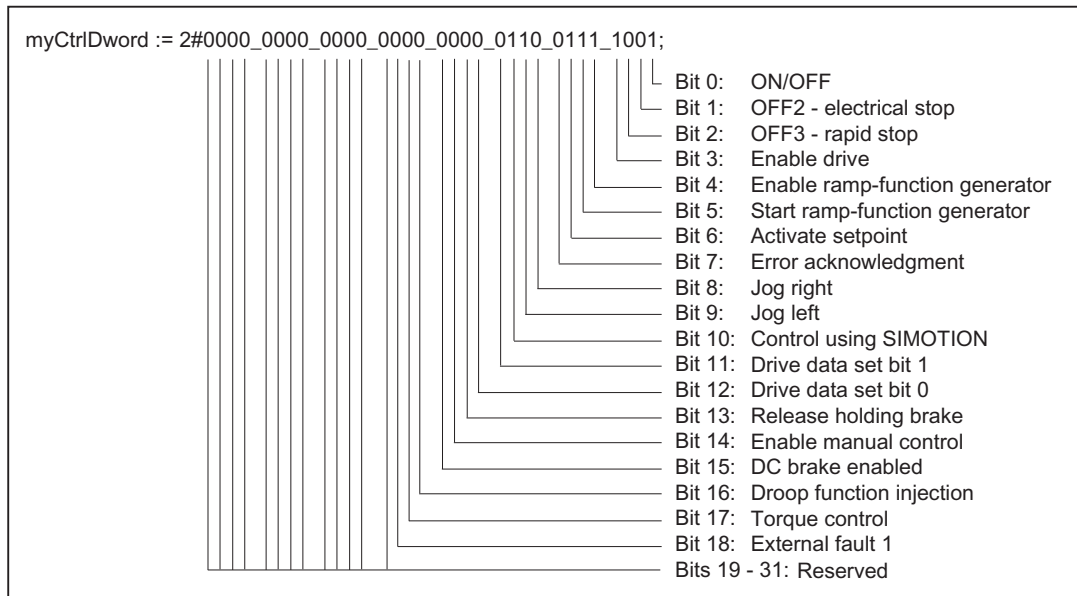


Figure 4-1 Meaning of control bits (in STW 1 and STW 2 for the ET 200S FC)

Description of functions

When the `_ET200S_FC_control` function block is called, the cyclic data is written and updated and jobs (such as reading/writing parameter) can be carried out.

The values in the input parameters **perIn**, **setpoint**, and **ctrlDword** are transmitted cyclically - regardless of the **busy**, **done**, **execute**, and **mode** parameters - and the output parameters **periOut**, **actValue**, and **statusDword** are updated cyclically.

Jobs are started with a rising edge on the input parameter **execute**. The type of data transfer is selected in the input parameter **mode**.

You can choose from the following options in the input parameter **mode**:

- READ: Individual parameters are read from the ET 200S FC (factory setting).
- WRITE: Individual parameters are written to the ET 200S FC.
- DATA1: The drive parameter p0810 is set to 0.0 (command data set 1 of the drive active).
- DATA2: The drive parameter p0810 is set to 1.0 (command data set 2 of the drive active).
- RESET_ERR: Resetting of parameterization and drive errors.

A job is started with a rising edge on the input parameter **execute** and the output parameters set to **done** = **FALSE** and **busy** = **TRUE**.

If **mode** = **WRITE** is selected, the parameters **paraNumber**, **subIndex**, and **paraValueWr** are accepted with a rising edge on the input parameter **execute**. An active data transfer is indicated by **busy** = **TRUE**. When a job is completed without errors, the following parameters are displayed: **busy** = **FALSE**, **done** = **TRUE**, and **error** = **FALSE**. When a job is completed with errors (parameterization errors), the following parameters are displayed: **busy** = **FALSE**, **done** = **FALSE**, and **error** = **TRUE**. The error number of the parameterization error is displayed in output parameter **errorID**. Parameterization errors are acknowledged with a rising edge on the input parameter **execute** and **mode** = **RESET_ERR** selected. If acknowledgement is

successful, the output parameter is set to **done = TRUE** and the **error** and **errorID** parameters are reset.

The drive parameter p0810 is set to 0.0 (command data set 1 of the drive active) with a rising edge on the input parameter **execute** and **mode = DATA1** selected. This drive parameter is set to 1.0 (command data set 2 of the drive active) with a rising edge on the input parameter **execute** and **mode = DATA2** selected. This means that control command acceptance can be switched from "USS on RS232" to "Fieldbus", if the relevant P0700 setting is made.

Note

If another job (e.g. reading parameters when a write job is active) is started when **busy = TRUE**, this is disregarded. A new job is only executed when **busy = FALSE**.

Note

When booting the firmware of the ET 200S frequency converter from the MMC card, it may take more than 500 ms for the ET 200S frequency converter to ramp up.

To avoid I/O area access errors when the frequency converter has not yet ramped up, accesses to the I/O inputs and outputs of the frequency converter must be performed with the **_getSafeValue** and **_setSafeValue** system functions (see sample program **UNIT E_FC_ctr** on the "SIMOTION Utilities & Applications" CD-ROM).

Graphic overview of the functionality

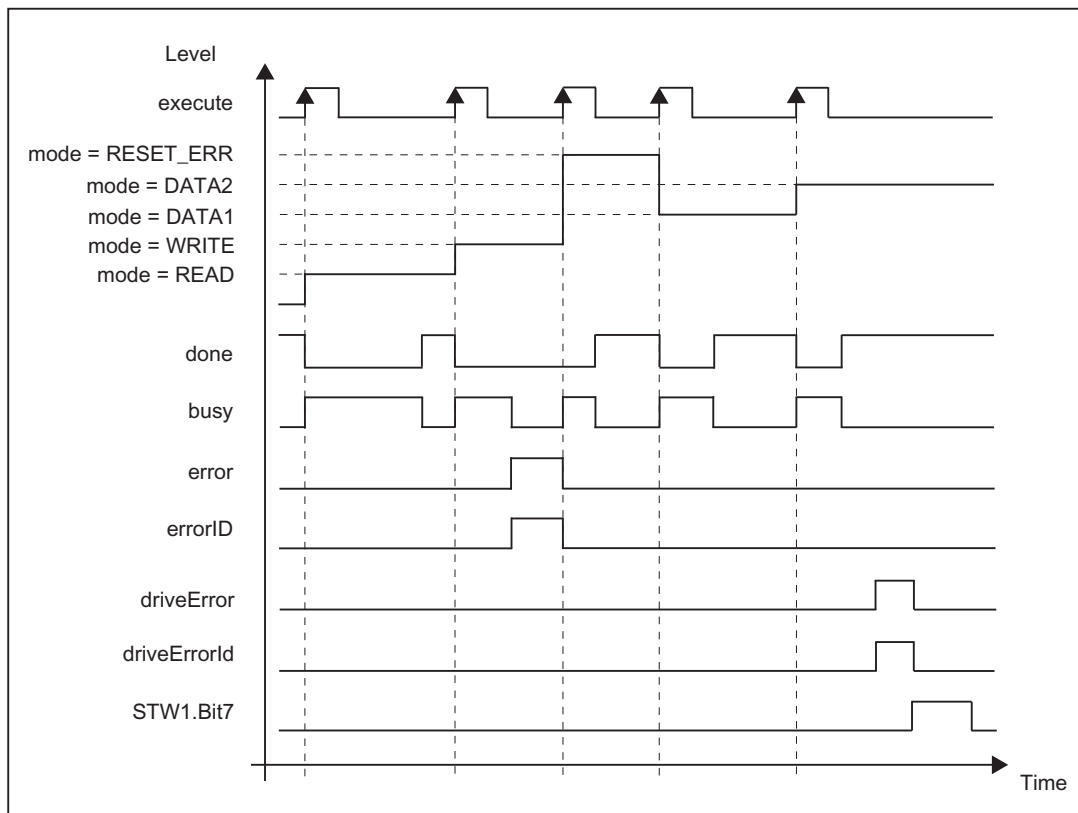


Figure 4-2 Signal propagation diagram

Task integration (call)

The `_ET200S_FC_control` FB is designed to be called in a cyclic task and must be called in this task during each task cycle. The execution of a job can take several cycles. The user decides which cyclical task the `_ET200S_FC_control` FB is called in. There are no restrictions with respect to this in the `_ET200S_FC_control` FB. The call can be performed in all cyclic tasks. A fixed time frame (e.g. `IPOSynchronousTask`) is not required for processing.

A separate instance of the `_ET200S_FC_control` function block must be created for the control and communication tasks of each frequency converter used.

Error messages

The drive errors of the ET 200S FC are output at the output parameters `driveError` and `driveErrorId` of the `_ET200S_FC_control` FB. The output parameter `driveErrorId` displays the associated error number (from r0947). For a description of the error numbers, see the *SIMATIC ET 200S FC Frequency Converter Parameter Manual* (03/2005 Edition).

Rectify the cause of the drive error. Drive errors are acknowledged with a rising edge on the input parameter `execute` and `mode = RESET_ERR` selected.

Parameterization errors are signaled at the output parameters `error` and `errorID`, and reset with a rising edge on input parameter `execute` and `mode = RESET_ERR` selected.

4.2 Calling the function block

Procedure

Proceed as follows to work with the **_ET200S_FC_control** function block in your user project (the numbers shown in the following program segment correspond to the steps listed below):

1. Create an instance of the function block.
2. Call instance of the function block.
3. Transfer input parameters.
4. The output parameters of the function block are accessed with <instance name of FB>.<name of output parameter>.
5. The data for the I/O outputs prepared by the FB are assigned to the I/O variables.

Note

The program segment is an extract from the supplied application example. The application example is included on the "SIMOTION Utilities & Applications" CD-ROM and is available for various SIMOTION hardware platforms.

The "SIMOTION Utilities & Applications" CD-ROM is provided free of charge with SIMOTION SCOUT.

If you want to control several frequency converters, you must create a new variable for the FB instances with a new name for each module used.

Separate data structures have been created for the input and output parameters in the example to provide a better overview.

Call example

```

UNIT E_FC_ctrl;

INTERFACE

TYPE
    ...                                // see example
END_TYPE

VAR_GLOBAL
    myFC_control      : _ET200S_FC_control;    // create instance of FB _ET200S_FC_control      (1
    myFC_control_In   : Struct_FC_control_In;  // input parameter of FB _ET200S_FC_control      )
    myFC_control_Out  : Struct_FC_control_Out; // output parameter of the FB _ET200S_FC_control
    myRetValSetSafe   : enumSetAndGetSafeValue; // return value from system function _setsafevalue

    ...                                // see example

```

4.2 Calling the function block

```

END_VAR

PROGRAM ExampleET200S_FC;                                // Program in BackgroundTask

END_INTERFACE

IMPLEMENTATION

PROGRAM ExampleET200S_FC
// Program in BackgroundTask
// call _getSafeValue -> Test I/O-variables after powerOn Reset
myRetValGetSafe := _getSafeValue (
    variable      := myPeriIn,
    accessMode    := DEFAULT_VALUE,
    getValue      := myTmpPeriIn
);
IF (myRetValGetSafe = OK)                                // powerOn ready
THEN
    // CALL FB INSTANCE
    myFC_control (                                       (2
                                                         )
    periIn        := myTmpPeriIn,                        // I/O peripheral input          (3
    execute       := myFC_control_In.execute,           // execute / start the data transfer          )
    mode          := myFC_control_In.mode,              // select the mode
    paraNumber    := myFC_control_In.paraNumber,        // parameter number to read or write
    subIndex      := myFC_control_In.subIndex,          // index of the parameter
    paraValueWr   := myFC_control_In.paraValueWr,       // value of the parameter to write
    setpoint      := myFC_control_In.setpoint,          // set the setpoint
    ctrlDword     := myFC_control_In.ctrlDword          // set the control word
    );

    // get output parameter of the FB _ET200S_FC_control (4
                                                         )

    myFC_control_Out.done          := myFC_control.done;           // ready
    myFC_control_Out.busy          := myFC_control.busy;           // at work
    myFC_control_Out.error         := myFC_control.error;          // error occurred
    myFC_control_Out.errorID       := myFC_control.errorID;        // number of error
    myFC_control_Out.driveError    := myFC_control.driveError;     // drive error
    myFC_control_Out.driveErrorId  := myFC_control.driveErrorId;   // number of the drive error
    myFC_control_Out.actValue      := myFC_control.actValue;       // per default the
                                                                    // actual speed value
    myFC_control_Out.statusDword   := myFC_control.statusDword;    // the status word
    myFC_control_Out.paraValueRd   := myFC_control.paraValueRd;    // value of the

```

```

// parameter to read
END_IF

// write peripheral output (5
myRetValSetSafe := _setSafeValue (
    variable := myPeriOut, // I/O peripheral output
    value := myFC_control.periOut,
    accessMode := NO_CHANGE
);

END_PROGRAM // end Program in BackgroundTask

END_IMPLEMENTATION
```


Application example

5.1 General

Task

The application example shows how you can control the ET 200S frequency converter with the aid of the **_ET200S_FC_control** function block. You can carry out the following functions with this example:

- Reset parameterization and drive errors
- Read parameter
- Write parameter
- Activate command data set 1
- Activate command data set 2
- Start the motor at a specified speed
- Stop a running motor

Requirements

The following requirements must be met:

- You have commissioned the ET 200S frequency converter (order no.: 6SL3244-0SA00-1AA0) with the connected motor.
For information on commissioning the frequency converter, see the SIMATIC *ET 200S FC Frequency Converter* Operating Instructions (03/2005 Edition).
This documentation is supplied in electronic form with SIMOTION SCOUT.
- You have completed the addressing of the ET 200S frequency converter (refer to chapter Addressing the ET 200S frequency converter (Page 17)).

Hardware platform

The application example is available for various SIMOTION hardware platforms.

Note

If the application example is not available for your SIMOTION hardware platform, you must adapt the hardware configuration.

Calling the application example

The application example can be found on the "SIMOTION Utilities & Applications" CD-ROM. The "SIMOTION Utilities & Applications" CD-ROM is provided free of charge with SIMOTION SCOUT.

1. Dearchive and open the project containing the application example.
2. Check the hardware configuration, e.g. the PROFIBUS DP addresses.
3. **Save** and **compile** the example project. Then, you can download the example to the SIMOTION device and switch to RUN mode.

Adapting the application example

The configuration in the example and its available hardware must be adapted.

The following options are available:

1. You can adapt the configuration of the example to the available hardware.
2. You can adapt the configuration of the hardware to the example.

You can adapt the configuration in the example to the available hardware.

You have assigned a module address $\neq$ 256 in **HW Config**.

1. In the symbol browser, change the I/O address to the value you assigned in **HW Config**:

	Name	I/O address	Read only	Data type	Field length
1	☒ myperiin	PIB 256	☒	Array	8
2	☒ myperiout	PQB 256	☐	Array	8

Figure 5-1 Adapting the I/O address

2. Enter the same value in the **myModuleAddr** variable of the application example.

Note

When booting the firmware of the ET 200S frequency converter from the MMC card, it may take more than 500 ms for the ET 200S frequency converter to ramp up.

To avoid I/O area access errors when the frequency converter has not yet ramped up, accesses to the I/O inputs and outputs of the frequency converter must be performed with the **_getSafeValue** and **_setSafeValue** system functions (see sample program **UNIT E_FC_ctrl** on the "SIMOTION Utilities & Applications" CD-ROM).

5.2 Sequence of the application example

Relevant variables in the application example

Table 5-1 Overview of the relevant variables

Symbol	Data type	Initial value	Meaning
myCommand	enumCommand (CMD_READ, CMD_WRITE, CMD_RESET_ERR, CMD_USE_DATA1, CMD_USE_DATA2, CMD_START, CMD_STOP)	CMD_READ	Selection of command to be executed
myCmdExecute	BOOL	FALSE	TRUE = Start selected command
myCmdDone	BOOL	FALSE	TRUE = Error-free command processing
myCmdError	INT	0	Error specification
myParaNumber	UINT	0	Number of parameter to be written
mySubIndex	UINT	0	Index of the parameter number
myParaValueWr	REAL	0.0	Parameter value to be written
myParaValueRd	REAL	0.0	Read parameter value
mySetpoint	INT	500	Speed specification in [rpm]
myMaxRevolution	INT	1,350	Maximum speed of the motor in [rpm]

Command processing

Proceed as follows to activate command processing in the sample program:

1. In the project navigator, under **Programs** select the **E_FC_ctrl Unit**.
2. Open the symbol browser in SIMOTION SCOUT.
3. Select a command in the symbol browser by setting the **myCommand** variable to the command to be executed.
4. Parameterize the command-specific variable.
5. Start the selected command by setting the **myCmdExecute** variable to **TRUE**.
6. Wait until error-free command execution is indicated by **myCmdDone = TRUE**. An error during command execution is specified in the **myCmdError** parameter.
7. You can now call additional commands.

The functions contained in the application example are described below.

Example: Resetting drive and parameterization errors

The function block displays drive errors at the output parameters **myFC_control.driveError = TRUE** and **myFC_control.driveErrorId ≠ 16#0000**; these must be acknowledged before the next command is called. Drive errors are reset with bit 7 in the control word **ctrlDword**.

Parameterization errors are indicated at the output parameters **myFC_control.error** = **TRUE** and **myFC_control.errorID** = **16#0000**.

Example: You have set an invalid parameter number (**myParaNumber** = 9999).

To acknowledge the parameterization error, you must set the variables **myCommand** and **myCmdExecute** to **CMD_RESET_ERR** and **TRUE** respectively.

The output parameters are then reset with **myFC_control.error** = **FALSE** and **myFC_control.errorID** = **16#0000**.

Example: Read parameter

Proceed as follows to read a parameter:

1. Set the **myCommand** variable to **CMD_READ** in the symbol browser.
2. Assign the **myParaNumber** variable the parameter number to be read (e.g. "1120", startup time).
3. Set the **mySubIndex** variable to "0".
4. Start the command with **myCmdExecute** = **TRUE**.
The parameter value read from the ET 200S FC is output in the **myParaValueRd** variable.

Example: Write parameter

Proceed as follows to write a parameter:

1. Set the **myCommand** variable to **CMD_WRITE** in the symbol browser.
2. Assign the **myParaNumber** variable the parameter number to be written (e.g. "1120", startup time).
3. Set the **mySubIndex** variable to "0".
4. Enter the parameter value to be written (e.g. "50") in the **myParaValueWr** variable.
5. Start the command with **myCmdExecute** = **TRUE**.
The value entered in the **myParaValueWr** variable is transmitted to the ET 200S FC.

Example: Activate command data set 1

Proceed as follows to activate command data set 1:

1. Set the **myCommand** variable to **CMD_USE_DATA1** in the symbol browser.
2. Start the command with **myCmdExecute** = **TRUE**.

Example: Activate command data set 2

Proceed as follows to activate command data set 2:

1. Set the **myCommand** variable to **CMD_USE_DATA2** in the symbol browser.
2. Start the command with **myCmdExecute** = **TRUE**.

Example: Start the motor at a specified speed

Proceed as follows to get the connected motor to turn at a speed of 500 rpm:

1. Enter the value "500" in the **mySetpoint** variable.
2. Set the **myCommand** variable to **CMD_START** in the symbol browser.
3. Start the command with **myCmdExecute = TRUE**.
The input parameter **myFC_Control_In.setpoint** is set to the speed parameterized in the **mySetpoint** variable accordingly and transferred to the input parameter **setpoint**.
All enables in control word 1 are set. The motor starts to turn.
After the speed parameterized in the **mySetpoint** variable is reached, the command is ended with **myCmdDone = TRUE**.

Note

When starting the motor with a rising edge on bit 0 in STW 1, bit 8 and bit 9 in STW 1 (jog right/left) must have the signal level **FALSE**.

Assignment of **ctrlDword** (STW 1 and STW 2 of ET 200S FC) when starting the motor:

CMD_START	: // start motor	16#0000 047F hex or binary: 0000 0100 0111 1111
CMD_JOG_RIGHT	: // motor jog. Right	16#0000 057E hex or binary: 0000 0101 0111 1110
CMD_JOG_LEFT	: // motor jog. Left	16#0000 067E hex or binary: 0000 0110 0111 1110

Example: Stop a running motor

Proceed as follows to stop a running motor:

1. Set the **myCommand** variable to **CMD_STOP** in the symbol browser.
2. Start the command with **myCmdExecute = TRUE**.
Bit 0 in control word 1 (On/Off) is reset. The motor begins to stop.
After reaching the actual speed < 1 rpm, the command is terminated with **myCmdDone = TRUE**.

Note

Bit 0, bit 8, and bit 9 in STW 1 must have the signal level **FALSE** when stopping a motor.

Assignment of **ctrlDword** (STW 1 and STW 2 of ET 200S FC) when stopping the motor:

CMD_STOP	: // stop motor	16#0000 047E hex or binary: 0000 0100 0111 1110
-----------------	-----------------	--

Alarm processing

6.1 Overview

Sequence of the alarm processing

Pending error messages are processed and evaluated differently than in a SIMATIC system. Diagnostic alarms are not enabled by default. Activate the alarms for the relevant module in the hardware configuration (refer to Chapter Inserting the ET 200S frequency converter into a SIMOTION project (Page 13)).

If you have parameterized diagnostic alarms, then you should program the alarm processing sequence according to the principle presented below.

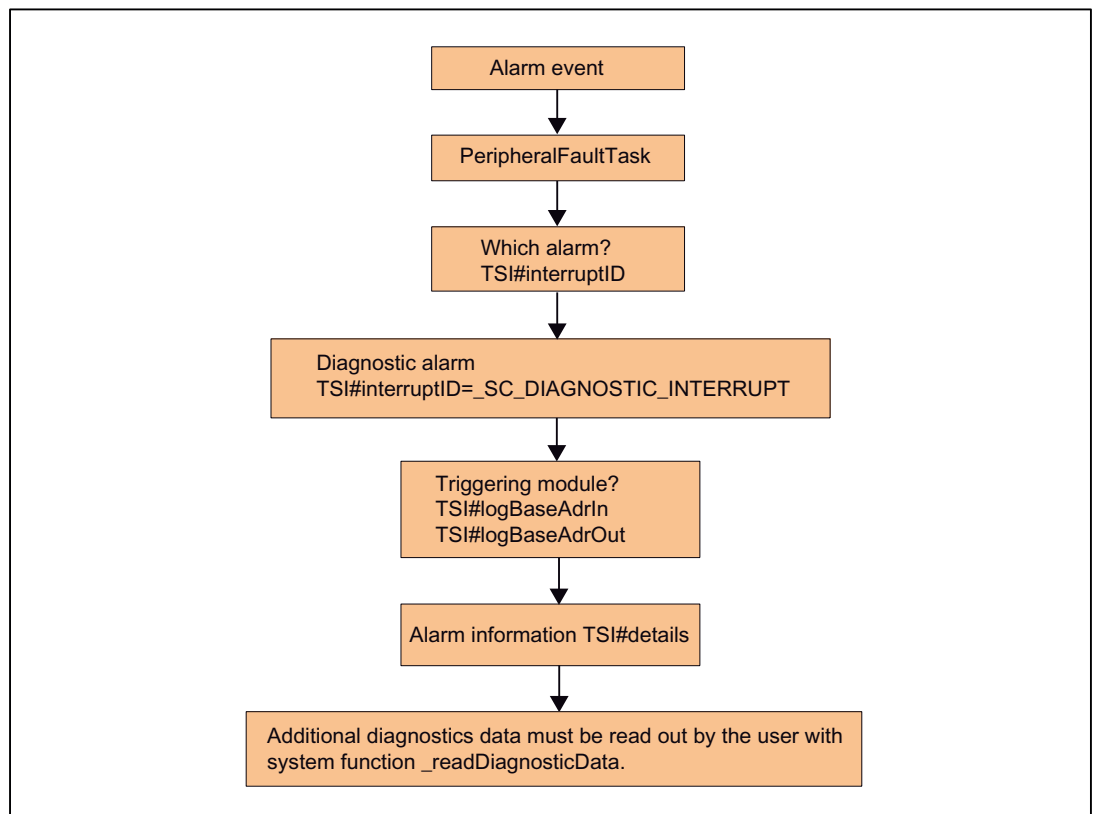


Figure 6-1 Alarm processing for the frequency converter

Additional references

A detailed description of the system functions can be found in the *SIMOTION System Function/ Variable Devices* Parameter Manual. This document is shipped with SIMOTION SCOUT in electronic form.

Detailed information on DP slave diagnostics can be found in the SIMATIC Manual titled *ET 200S Distributed I/O System*.

Alarm evaluation

Alarms sent by the I/O are evaluated in the **PeripheralFaultTask**. When the **PeripheralFaultTask** is started, the **Taskstartinfo** is made available; you must evaluate this in the user program. To do this, assign a program to the **PeripheralFaultTask** in the execution system (see sample program **Unit_E_FC_ctr**).

The **Taskstartinfo** of **PeripheralFaultTask** is comparable to the local data of OB 82 in the SIMATIC system.

Table 6-1 Meaning of the Taskstartinfo

Task	TSI		Meaning
PeripheralFaultTask	DT	TSI#startTime	Start time of the task
	UDINT	TSI#interruptID	Identifies the triggering event: • _SC_PROCESS_INTERRUPT • _SC_DIAGNOSTIC_INTERRUPT • _SC_STATION_DISCONNECTED • _SC_STATION_RECONNECTED
	DINT	TSI#logBaseAdrIn	Logical start address if a hardware interrupt (PRAL) or a diagnostic interrupt (DAL) was caused by an input range on the module, otherwise _SC_INVALID_ADDRESS
	DINT	TSI#logBaseAdrOut	Logical start address if a hardware interrupt (PRAL) or a diagnostic interrupt (DAL) was caused by an output range on the module, otherwise _SC_INVALID_ADDRESS
	DINT	TSI#logDiagAdr	Diagnostic address of a DP slave if the interrupt was caused by a CPU stop or recovery of the associated DP slave, otherwise _SC_INVALID_ADDRESS
	DWORD	TSI#details	Detail information (bit fields)

6.2 Diagnostic alarms

Diagnostic alarms enable operational messages (hardware faults) of the ET 200S FC to be detected in the SIMOTION device.

Definition of a diagnostic alarm

If the user program is to respond to an internal or external error, you can set the parameters for a diagnostic alarm that will interrupt program execution in the SIMOTION device.

Events that trigger a diagnostic alarm

The criteria (events) for triggering diagnostic alarms in a SIMOTION system correspond to those in a SIMATIC system.

For a more detailed description, refer to the section titled "Diagnostics and monitoring functions" in the SIMATIC *ET 200S FC Frequency Converter* Operating Instructions (03/2005 Edition).

Responses to a diagnostic alarm

If a diagnostic alarm is issued, the following occurs:

1. The group error LED (SF) lights up on the ET 200S frequency converter.
2. Diagnostic data is written to the **TS#details** variable in the Taskstartinfo of **PeripheralFaultTask**. The program assigned to the **PeripheralFaultTask** is executed.
3. The SIMOTION device goes into STOP mode if a program has not been assigned to the **PeripheralFaultTask**.
4. The group error LED (SF) goes out as soon as the error has been remedied.

Bit assignment

The **TS#details** variable is assigned in the same way as in a SIMATIC system.

Note

For a more detailed description, refer to the section titled "Diagnostics and monitoring functions" in the SIMATIC *ET 200S FC Frequency Converter* Operating Instructions (03/2005 Edition).

Appendix

A.1 SIMOTION and SIMATIC names

The table below contains a comparison of SIMOTION and SIMATIC names.

Name in the SIMOTION system, V4.1 and higher (command library in SCOUT)	Name in the SIMATIC system
Function block parameter	
_ET200S_FC_control	ET200S_FC_DRIV
execute	REQ
mode	-
periIn	-
setpoint	FREQ_SET
ctrlDword	CTRL
paraNumber	PAR_ADDRESS
subIndex	PAR_INDEX
paraValueWr	WRITE_DATA_VALUE
done	-
busy	BUSY
error	PAR_ERR
errorID	PAR_ERR_NO
driveError	ERR_NO_VALID
driveErrorId	ERR_NO
periOut	-
actValue	FREQ
statusDword	STAT
paraValueRd	READ_DATA_VALUE

A.2 List of abbreviations

Table A-1 Abbreviations

Abbreviation	Meaning
DP	Distributed I/O
FB	Function Block
FC	Frequency Converter
IN	Input parameters
IM	Interface Module
I/O	Input/Output
LAD	Ladder diagram
LED	Light Emitting Diode
OUT	Output parameters
PG/PC	Programming Device / Personal Computer
PIB	Peripheral Input Byte
PM	Power Module
PQB	Peripheral Output Byte
PS	Power Supply
ST	Structured Text
STW	Control word
ZSW	Status word

Index

A

- Addressing
 - Example:, 17
 - I/O variable, 17
- Alarm processing
 - Bit assignment, 37
 - Diagnostic alarm, 37
 - Evaluation, 36
 - Programming the sequence, 35
- Application example
 - Aligning, 30
 - Call, 30

C

- Call example, 25
- Command processing, 31
- Connection, 12

D

- Diagnostic alarm
 - Definition, 37
 - Responses, 37
 - Triggering events, 37

E

- Error messages, 24
- ET 200S frequency converter, 10
 - Addressing, 17
 - Connection example, 10
 - Inserting in project, 13
 - Possible applications, 10
 - Product description, 10
- Example:
 - Activate command data set 1, 32
 - Activate command data set 2, 32
 - Read parameter, 32
 - Resetting parameterization errors, 32
 - Start the motor at a specified speed, 33
 - Stop a running motor, 33
 - Write parameter, 32

F

- FB_ET200S_FC_control
 - Application example, 29
 - Call example, 25
 - Creating an instance, 15
 - Error messages, 24
 - Functions, 22
 - Parameter, 20
 - Task, 19
 - Task integration, 24

I

- Installation, 12
- Integrating the function block, 15

P

- Parameter transfer, 18
- Possible applications, 10

R

- References, 3
- Requirements, 9

S

- SIMOTION and SIMATIC names, 39

T

- Taskstartinfo, 36

