

# SPEED7 Library

OPL\_SP7-LIB | SW90HS0MA V10.001 | Manual

HB00 | OPL\_SP7-LIB | SW90HS0MA V10.001 | en | 24-02

Block library - EtherCAT Communication



YASKAWA Europe GmbH  
Philipp-Reis-Str. 6  
65795 Hattersheim  
Germany  
Tel.: +49 6196 569-300  
Fax: +49 6196 569-398  
Email: [info@yaskawa.eu](mailto:info@yaskawa.eu)  
Internet: [www.yaskawa.eu.com](http://www.yaskawa.eu.com)

Table of contents

1 General. . . . . 4

1.1 Copyright © YASKAWA Europe GmbH. . . . . 4

1.2 About this manual. . . . . 5

2 Important notes. . . . . 6

2.1 General. . . . . 6

2.2 Internally used blocks. . . . . 6

3 Include library. . . . . 7

3.1 Integration into Siemens SIMATIC Manager. . . . . 7

4 Block parameters. . . . . 8

4.1 RET\_VAL and BUSY for asynchronously operating blocks. . . . . 8

4.2 General and Specific Error Information RET\_VAL. . . . . 8

5 EtherCAT Communication. . . . . 11

5.1 SDO Communication. . . . . 11

5.1.1 FB 52 - SDO\_READ - Read access to Object Dictionary Area. . . . . 11

5.1.2 FB 53 - SDO\_WRITE - Write access to Object Dictionary Area. . . . . 15

# 1 General

## 1.1 Copyright © YASKAWA Europe GmbH

<b>All Rights Reserved</b>	<p>This document contains proprietary information of Yaskawa and is not to be disclosed or used except in accordance with applicable agreements.</p> <p>This material is protected by copyright laws. It may not be reproduced, distributed, or altered in any fashion by any entity (either internal or external to Yaskawa) except in accordance with applicable agreements, contracts or licensing, without the express written consent of Yaskawa and the business management owner of the material.</p> <p>For permission to reproduce or distribute, please contact: YASKAWA Europe GmbH, European Headquarters, Philipp-Reis-Str. 6, 65795 Hattersheim, Germany</p> <p>Tel.: +49 6196 569 300 Fax.: +49 6196 569 398 Email: <a href="mailto:info@yaskawa.eu">info@yaskawa.eu</a> Internet: <a href="http://www.yaskawa.eu.com">www.yaskawa.eu.com</a></p>
<b>EC conformity declaration</b>	<p>Hereby, YASKAWA Europe GmbH declares that the products and systems are in compliance with the essential requirements and other relevant provisions. Conformity is indicated by the CE marking affixed to the product.</p>
<b>Conformity Information</b>	<p>For more information regarding CE marking and Declaration of Conformity (DoC), please contact your local representative of YASKAWA Europe GmbH.</p>
<b>Trademarks</b>	<p>System 300S and SPEED7 are registered trademarks of YASKAWA Europe GmbH.</p> <p>EtherCAT is a registered trademark of Beckhoff Automation GmbH.</p> <p>SIMATIC, STEP, TIA Portal, S7-300 and S7-1500 are registered trademarks of Siemens AG.</p> <p>All Microsoft Windows, Office and Server products mentioned are registered trademarks of Microsoft Inc., USA.</p> <p>All other trademarks, logos and service or product marks specified herein are owned by their respective companies.</p>
<b>General terms of use</b>	<p>Every effort has been made to ensure that the information contained in this document was complete and accurate at the time of publishing. We cannot guarantee that the information is free of errors, and we reserve the right to change the information at any time. There is no obligation to inform the customer about any changes. The customer is requested to actively keep his documents up to date. The customer is always responsible for the deployment of the products with the associated documentation, taking into account the applicable directives and standards.</p> <p>This documentation describes all hardware and software units and functions known today. It is possible that units are described that do not exist at the customer. The exact scope of delivery is described in the respective purchase contract.</p>
<b>Document support</b>	<p>Contact your local representative of YASKAWA Europe GmbH if you have errors or questions regarding the content of this document. You can reach YASKAWA Europe GmbH via the following contact:</p> <p>Email: <a href="mailto:Documentation.HER@yaskawa.eu">Documentation.HER@yaskawa.eu</a></p>

## Technical support

Contact your local representative of YASKAWA Europe GmbH if you encounter problems or have questions regarding the product. If such a location is not available, you can reach the Yaskawa customer service via the following contact:

YASKAWA Europe GmbH,  
European Headquarters, Philipp-Reis-Str. 6, 65795 Hattersheim, Germany  
Tel.: +49 6196 569 500 (hotline)  
Email: support@yaskawa.eu

## 1.2 About this manual

### Objective and contents

The manual describes the block library '*EtherCAT Communication*':

- The manual is targeted at users who have a background in automation technology.
- The manual consists of chapters. Every chapter provides a self-contained description of a specific topic.
- The following guides are available in the manual:
  - An overall table of contents at the beginning of the manual
  - References with pages numbers

### Icons Headings

Important passages in the text are highlighted by following icons and headings:



#### **DANGER**

Immediate or likely danger. Personal injury is possible.



#### **CAUTION**

Damages to property is likely if these warnings are not heeded.



*Supplementary information and useful tips.*

## 2 Important notes

### 2.1 General



*In the following, you will find important notes, which must always be observed when using the blocks.*

### 2.2 Internally used blocks



#### CAUTION

The following blocks are used internally and must not be overwritten! The direct call of an internal block leads to errors in the corresponding instance DB! Please always use the corresponding function for the call.

FC/SFC	Designation	Description
FC/SFC 131	TSEND_	is used internally for FB 63
FC/SFC 132	TRECV_	is used internally for FB 64
FC/SFC 133	TCON_	is used internally for FB 65
FC/SFC 134	TDISCON_	is used internally for FB 66
FC/SFC 135	TUSEND_	is used internally for FB 67
FC/SFC 136	TURECV_	is used internally for FB 68
FC/SFC 192	CP_S_R	is used internally for FB 7 and FB 8
FC/SFC 196	AG_CNTRL	is used internally for FC 10
FC/SFC 198	USEND_	is used internally for FB 8
FC/SFC 198	URCV_	is used internally for FB 9
FC/SFC 200	AG_GET	is used internally for FB/SFB 14
FC/SFC 201	AG_PUT	is used internally for FB/SFB 15
FC/SFC 202	AG_BSEND	is used internally for FB/SFB 12
FC/SFC 203	AG_BRCV	is used internally for FB/SFB 13
FC/SFC 204	IP_CONF	is used internally for FB 55 IP_CONF
FC/SFC 205	AG_SEND	is used internally for FC 5 AG_SEND
FC/SFC 206	AG_RECV	is used internally for FC 6 AG_RECV
FC/SFC 253	IBS_ACCESS	is used internally for SPEED bus INTERBUS masters
SFB 238	EC_RWOD	is used internally for EtherCAT Communication
SFB 239	FUNC	is used internally for FB 240, FB 241

### 3 Include library

#### Block library EtherCAT Communication

The block library can be found for download in the 'Download Center' of [www.yaskawa.eu.com](http://www.yaskawa.eu.com) under 'Controls Library' as 'Block library EtherCAT Communication - SW90HS0MA'. The library is available as packed zip file. As soon as you want to use these blocks you have to import them into your project.



*Please always use the manual associated with your library. As long as there are no description-relevant changes, the version information in the manual can differ from those of the library and its files.*

The following block libraries are available

File	Description
EtherCATCom_S7_V0001.zip	<ul style="list-style-type: none"> <li>■ Block library for Siemens SIMATIC Manager.</li> <li>■ For use in Yaskawa CPUs or S7-300 CPUs from Siemens.</li> </ul>

#### 3.1 Integration into Siemens SIMATIC Manager

##### Overview

The integration into the Siemens SIMATIC Manager requires the following steps:

1. ➤ Load ZIP file
2. ➤ "Retrieve" the library
3. ➤ Open library and transfer blocks into the project

##### Load ZIP file

- Navigate on the web page to the desired ZIP file, load and store it in your work directory.

##### Retrieve library

1. ➤ Start the Siemens SIMATIC Manager with your project.
2. ➤ Open the dialog window for ZIP file selection via 'File → Retrieve'.
3. ➤ Select the according ZIP file and click at [Open].
4. ➤ Select a destination folder where the blocks are to be stored.
5. ➤ Start the extraction with [OK].

##### Open library and transfer blocks into the project

1. ➤ Open the library after the extraction.
  2. ➤ Open your project and copy the necessary blocks from the library into the directory "blocks" of your project.
- ➔ Now you have access to the blocks via your user application.



*Are FCs used instead of SFCs, so they are supported by the System 300S Yaskawa CPUs starting from firmware 3.6.0.*

## 4 Block parameters

### 4.1 RET\_VAL and BUSY for asynchronously operating blocks

#### Output parameters RET\_VAL and BUSY

For blocks that work asynchronously, the function execution extends over several block calls. Here, the output parameters RET\_VAL and BUSY indicate the status of the job execution:

- First call with REQ = 1
  - With free system resources and correct input parameters, BUSY is set to 1 and RET\_VAL returns W#16#7001.
  - If the system resources are occupied or the input parameters are faulty, BUSY is set to 0 and the corresponding error code is returned in RET\_VAL.
- Interim call
  - BUSY remains set to 1 and RET\_VAL returns W#16#7002. This signals that the job is still being processed.
- Last call
  - After error-free execution, BUSY is set to 0 and RET\_VAL returns 0. Please note that some blocks report the number of transmitted data via RET\_VAL. For more information, please refer to the corresponding block descriptions.
  - In the event of an error, BUSY is set to 0 and the RET\_VAL returns the corresponding error code.

#### Input parameter REQ

The REQ input parameter is used exclusively to initiate a job:

- REQ = 1 in the corresponding block executes a job that is not yet active.
- REQ is not evaluated with each subsequent call of the block.

#### Correlation of the parameters

Call number	Call type	REQ	RET_VAL	BUSY
1	First call	1	W#16#7001	1
			Error code in the event of an error	0
2 ... n-1	Interim call	not relevant	W#16#7002	1
n	Last call	not relevant	W#16#0000 <sup>1</sup>	0
			Error code in the event of an error	

1) For some blocks, the number of transmitted data - see corresponding block description.

### 4.2 General and Specific Error Information RET\_VAL

#### Overview

The return value *RET\_VAL* of a system function provides one of the following types of error codes:

- A *general error code*, that relates to errors that can occur in anyone SFC.
- A *specific error code*, that relates only to the particular SFC.

Although the data type of the output parameter *RET\_VAL* is integer (INT), the error codes for system functions are grouped according to hexadecimal values.

If you want to examine a return value and compare the value with the error codes, then display the error code in hexadecimal format.



**RET\_VAL (Return value)**

The table below shows the structure of a system function error code:

Bit	Description
7 ... 0	Event number or error class and single error
14 ... 8	Bit 14 ... 8 = "0": <b>Specific error code</b> The specific error codes are listed in the descriptions of the individual SFCs. Bit 14 ... 8 > "0": <b>General error code</b> The possible general error codes are shown
15	Bit 15 = "1": indicates that an error has occurred.

**Specific error code**

This error code indicates that an error pertaining to a particular system function occurred during execution of the function.

A specific error code consists of the following two numbers:

- Error class between 0 and 7
- Error number between 0 and 15

Bit	Description
3 ... 0	Error number
6 ... 4	Error class
7	Bit 7 = "1"
14 ... 8	Bit 14 ... 8 = "0"
15	Bit 15 = "1": indicates that an error has occurred.

**General error codes  
RET\_VAL**

The parameter *RET\_VAL* of some SFCs only returns general error information. No specific error information is available.

The general error code contains error information that can result from any system function. The general error code consists of the following two numbers:

- A parameter number between 1 and 111, where 1 indicates the first parameter of the SFC that was called, 2 the second etc.
- An event number between 0 and 127. The event number indicates that a synchronous fault has occurred.

Bit	Description
7 ... 0	Event number
14 ... 8	Parameter number
15	Bit 15 = "1": indicates that an error has occurred.

**General error codes**

The following table explains the general error codes associated with a return value. Error codes are shown as hexadecimal numbers. The x in the code number is only used as a placeholder. The number represents the parameter of the system function that has caused the error.

Error code	Description
8x7Fh	Internal Error. This error code indicates an internal error at parameter x. This error did not result from the actions if the user and he/she can therefore not resolve the error.
8x01h	Illegal syntax detection for an ANY parameter.
8x22h	Area size error when a parameter is being read.
8x23h	Area size error when a parameter is being written. This error code indicates that parameter x is located either partially or fully outside of the operand area or that the length of the bit-field for an ANY-parameter is not divisible by 8.
8x24h	Area size error when a parameter is being read.
8x25h	Area size error when a parameter is being written. This error code indicates that parameter x is located in an area that is illegal for the system function. The description of the respective function specifies the areas that are not permitted for the function.
8x26h	The parameter contains a number that is too high for a time cell. This error code indicates that the time cell specified in parameter x does not exist.
8x27h	The parameter contains a number that is too high for a counter cell (numeric fields of the counter). This error code indicates that the counter cell specified in parameter x does not exist.
8x28h	Orientation error when reading a parameter.
8x29h	Orientation error when writing a parameter. This error code indicates that the reference to parameter x consists of an operand with a bit address that is not equal to 0.
8x30h	The parameter is located in the write-protected global-DB.
8x31h	The parameter is located in the write-protected instance-DB. This error code indicates that parameter x is located in a write-protected data block. If the data block was opened by the system function itself, then the system function will always return a value 8x30h.
8x32h	The parameter contains a DB-number that is too high (number error of the DB).
8x34h	The parameter contains a FC-number that is too high (number error of the FC).
8x35h	The parameter contains a FB-number that is too high (number error of the FB). This error code indicates that parameter x contains a block number that exceeds the maximum number permitted for block numbers.
8x3Ah	The parameter contains the number of a DB that was not loaded.
8x3Ch	The parameter contains the number of a FC that was not loaded.
8x3Eh	The parameter contains the number of a FB that was not loaded.
8x42h	An access error occurred while the system was busy reading a parameter from the peripheral area of the inputs.
8x43h	An access error occurred while the system was busy writing a parameter into den peripheral area of the outputs.
8x44h	Error during the n-th ( $n > 1$ ) read access after an error has occurred.
8x45h	Error during the n-th ( $n > 1$ ) write access after an error has occurred. This error code indicates that access was denied to the requested parameter.

## 5 EtherCAT Communication

### 5.1 SDO Communication

#### 5.1.1 FB 52 - SDO\_READ - Read access to Object Dictionary Area

##### Description

With this block, you will have read access to the object directory of the EtherCAT slave stations and EtherCAT master. The block operates asynchronously, that is, processing covers multiple FB calls. Start the job by calling FB 52 with REQ = 1. The job status is displayed via the output parameters BUSY and RETVAL. The record set transmission is completed when the output parameter BUSY = FALSE.

The error handling happens with the parameters ERROR, ERROR\_ID and RETVAL  
 → [‘RET\\_VAL and BUSY for asynchronously operating blocks’...page 8.](#)

##### Parameters

Parameter	Declaration	Data type	Description
REQ	IN	BOOL	REQ = 1: activates the SDO access at rising edge.
ID	IN	WORD	Logical base address of the EtherCAT slave station respectively master in the hardware configuration.  With an output module bit 15 must be set (example for address 5: ID:=DW#16#8005). With a combination module you have to set the lower one of the two addresses.
INDEX	IN	WORD	Index of the object for the SDO access.
SUBINDEX	IN	BYTE	Sub index of the object for the SDO access.
COMPL_ACCESS	IN	BOOL	This parameter defines whether only a single sub-index, or the entire object is to be read.
MLEN	IN	INT	Maximum length of the data to be read.
VALID	OUT	BOOL	indicates that a new record set was received and is valid.
BUSY	OUT	BOOL	This parameter indicates the status of the SDO access. <i>BUSY</i> = 1: SDO access is not yet terminated.
ERROR	OUT	BOOL	<i>ERROR</i> = 1: A read error has occurred.
RETVAL	OUT	INT	Return value (0 = OK)
ERROR_ID	OUT	DWORD	Bus specific error code. If there was an error during the SDO access, the SDO abort error code (EtherCAT error code) can be found here.
LEN	OUT	INT	Length of the read data.
RECORD	IN_OUT	ANY	Area of the read data.



Please note that the data transferred to RECORD are not in a temporary area.

**Special features at  
COMPL\_ACCESS (CompleteAccess)**

With the activation of the parameter *COMPL\_ACCESS* the following is to be considered:

- With *COMPL\_ACCESS* = true only *SUBINDEX* 0 or 1 is allowed! Otherwise you will get an error message.
- With *COMPL\_ACCESS* = true for *SUBINDEX* 0 2 bytes are read, because *SUBINDEX* 1 has an offset of 2 byte.

**RETVAL (return value)**

In addition to the module specific error codes, which are listed here, also the general error codes for FC/SFC as return value are possible. ➔ ['General and Specific Error Information RET\\_VAL'...page 8](#)

RETVAL	Description	Error code in <i>ERROR_ID</i>
0x7000	First call with REQ = 0, Job was not triggered.	no
0x7001	First call with REQ = 1, Job was triggered, Block ist ready for data transfer.	yes
0x7002	Intermediate call: Job is being processed.	no
0x8090	I/O address range is different from I/O address range in SPEED7 EtherCAT Manager	yes
0x80A0	Negative acknowledgement while reading the module.	yes
0x80A1	Negative acknowledgement while writing the module.	yes
0x80A3	General protocol error.	yes
0x80A5	Internal error.	Value = 0: no Value ≠ 0: yes
0x80A7	Module is occupied (Timeout).	yes
0x80A9	Feature not supported by the module.	yes
0x80AA	Module reports a manufacturer-specific error in its application.	yes
0x80B0	Data record not known in module / Illegal data record number.	yes
0x80B4	Module reports access to an invalid area.	yes
0x80B5	Module not ready.	yes
0x80B6	Module denies access.	yes
0x80B7	Module reports an invalid range for a parameter or value.	yes
0x80B8	Module reports an invalid parameter.	yes
0x80B9	Module reports an invalid type: Buffer too small (reading subsets is not possible).	yes
0x80C2	The module currently processes the maximum possible jobs for a CPU.	yes
0x80C3	The required operating resources are currently occupied.	no
0x80C4	Internal temporary error: Job could not be carried out.	yes
0x80C5	Module not available.	yes
0x80D2	Error on reading an SDO due to wrong call parameters.	yes

**ERROR\_ID**

On a *RETVAL* more information can be found in the *ERROR\_ID* if available. Otherwise *ERROR\_ID* is 0.

Internal error	Description
0x00000000	No error
0x98110001	Feature not supported
0x98110002	Invalid Index
0x98110003	Invalid Offset
0x98110005	Invalid Size
0x98110006	Invalid Data
0x98110007	Not ready
0x98110008	Busy
0x9811000A	No Memory left
0x9811000B	Invalid Parameter
0x9811000C	Not Found
0x9811000E	Invalid state
0x98110010	Timeout
0x98110011	Open Failed
0x98110012	Send Failed
0x98110014	Invalid Command
0x98110015	Unknown Mailbox Protocol Command
0x98110016	Access Denied
0x98110024	Slave error
0x9811002D	Ethernet link cable disconnected
0x98110031	No mailbox support

CoE Error codes	Description	CoE slave abort code
0x98110040	SDO: Toggle bit not alternated	0x05030000
0x98110041	SDO protocol timed out	0x05040000
0x98110042	SDO: Client/server command specifier not valid or unknown	0x05040001
0x98110043	SDO: Invalid block size (block mode only)	0x05040002
0x98110044	SDO: Invalid sequence number (block mode only)	0x05040003
0x98110045	SDO: CRC error (block mode only)	0x05040004
0x98110046	SDO: Out of memory	0x05040005
0x98110047	SDO: Unsupported access to an object	0x06010000
0x98110048	SDO: Attempt to read a write only object	0x06010001
0x98110049	SDO: Attempt to write a read only object	0x06010002
0x9811004A	SDO: Object does not exist in the object dictionary	0x06020000
0x9811004B	SDO: Object cannot be mapped to the PDO	0x06040041
0x9811004C	SDO: The number and length of the objects to be mapped would exceed PDO length	0x06040042
0x9811004D	SDO: General parameter incompatibility reason	0x06040043

CoE Error codes	Description	CoE slave abort code
0x9811004E	SDO: General internal incompatibility in the device	0x06040047
0x9811004F	SDO: Access failed due to an hardware error	0x06060000
0x98110050	SDO: Data type does not match, length of service parameter does not match	0x06070010
0x98110051	SDO: Data type does not match, length of service parameter too high	0x06070012
0x98110052	SDO: Data type does not match, length of service parameter too low	0x06070013
0x98110053	SDO: Sub-index does not exist	0x06090011
0x98110054	SDO: Value range of parameter exceeded (only for write access)	0x06090030
0x98110055	SDO: Value of parameter written too high	0x06090031
0x98110056	SDO: Value of parameter written too low	0x06090032
0x98110057	SDO: Maximum value is less than minimum value	0x06090036
0x98110058	SDO: General error	0x08000000
0x98110059	SDO: Data cannot be transferred or stored to the application	0x08000020
0x9811005A	SDO: Data cannot be transferred or stored to the application because of local control	0x08000021
0x9811005B	SDO: Data cannot be transferred or stored to the application because of the present device state	0x08000022
0x9811005C	SDO: Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of an file error)	0x08000023
0x9811005D	SDO: Unknown code	unknown
0x9811010E	Command not executed	Slave is not present at the bus

## 5.1.2 FB 53 - SDO\_WRITE - Write access to Object Dictionary Area

### Description

With this block, you will have write access to the object directory of the EtherCAT slave stations and EtherCAT master. The block operates asynchronously, that is, processing covers multiple FB calls. Start the job by calling FB 53 with REQ = 1. The job status is displayed via the output parameters BUSY and RETVAL. The record set transmission is completed when the output parameter BUSY = FALSE.

The error handling happens with the parameters ERROR, ERROR\_ID and RETVAL  
 → [‘RET\\_VAL and BUSY for asynchronously operating blocks’...page 8.](#)

### Parameters

Parameter	Declaration	Data type	Description
REQ	IN	BOOL	REQ = 1: activates the SDO access at rising edge.
ID	IN	WORD	Logical base address of the EtherCAT slave station respectively master in the hardware configuration.  With an output module bit 15 must be set (example for address 5: ID:=DW#16#8005). With a combination module you have to set the lower one of the two addresses.
INDEX	IN	WORD	Index of the object for the SDO access.
SUBINDEX	IN	BYTE	Sub index of the object for the SDO access.
COMPL_ACCESS	IN	BOOL	This parameter defines whether only a single sub-index, or the entire object is to be written.
LEN	IN	INT	Maximum length of the data to be written.
DONE	OUT	BOOL	indicates that a new record set was written.
BUSY	OUT	BOOL	This parameter indicates the status of the SDO access.  <i>BUSY</i> = 1: SDO access is not yet terminated.
ERROR	OUT	BOOL	<i>ERROR</i> = 1: A write error has occurred.
RETVAL	OUT	INT	Return value (0 = OK)
ERROR_ID	OUT	DWORD	Bus specific error code. If there was an error during the SDO access, the SDO abort error code (EtherCAT error code) can be found here.
LEN	OUT	INT	Length of the data to be written.
RECORD	IN_OUT	ANY	Area of the data to be written.



Please note that the data transferred to RECORD are not in a temporary area.

### Special features at COMPL\_ACCESS (CompleteAccess)

With the activation of the parameter *COMPL\_ACCESS* the following is to be considered:

- With *COMPL\_ACCESS* = true only *SUBINDEX* 0 or 1 is allowed! Otherwise you will get an error message.
- With *COMPL\_ACCESS* = true for *SUBINDEX* 0 2 bytes are written, because *SUBINDEX* 1 has an offset of 2 bytes.

**RETVAL (return value)**

In addition to the module specific error codes, which are listed here, also the general error codes for FC/SFC as return value are possible. ➔ [‘General and Specific Error Information RET\\_VAL’...page 8](#)

RETVAL	Description	Error code in <i>ERROR_ID</i>
0x7000	First call with REQ = 0, Job was not triggered.	no
0x7001	First call with REQ = 1, Job was triggered, Block ist ready for data transfer.	yes
0x7002	Intermediate call: Job is being processed.	no
0x80A0	Negative acknowledgement while reading the module.	yes
0x80A1	Negative acknowledgement while writing the module.	yes
0x80A3	General protocol error.	yes
0x80A5	Internal error.	Value = 0: no
		Value ≠ 0: yes
0x80A7	Module is occupied (Timeout).	yes
0x80A9	Feature not supported by the module.	yes
0x80AA	Module reports a manufacturer-specific error in its application.	yes
0x80B0	Data record not known in module / Illegal data record number.	yes
0x80B4	Module reports access to an invalid area.	yes
0x80B5	Module not ready.	yes
0x80B6	Module denies access.	yes
0x80B7	Module reports an invalid range for a parameter or value.	yes
0x80B8	Module reports an invalid parameter.	yes
0x80B9	Module reports an invalid type: Buffer too small (writing subsets is not possible).	yes
0x80C2	The module currently processes the maximum possible jobs for a CPU.	yes
0x80C3	The required operating resources are currently occupied.	no
0x80C4	Internal temporary error: Job could not be carried out.	yes
0x80C5	Module not available.	yes
0x80D2	Error on reading an SDO due to wrong call parameters.	yes

**ERROR\_ID**

On a *RETVAL* more information can be found in the *ERROR\_ID* if available. Otherwise *ERROR\_ID* is 0.

Internal error	Description
0x00000000	No error
0x98110001	Feature not supported
0x98110002	Invalid Index
0x98110003	Invalid Offset
0x98110005	Invalid Size



Internal error	Description
0x98110006	Invalid Data
0x98110007	Not ready
0x98110008	Busy
0x9811000A	No Memory left
0x9811000B	Invalid Parameter
0x9811000C	Not Found
0x9811000E	Invalid state
0x98110010	Timeout
0x98110011	Open Failed
0x98110012	Send Failed
0x98110014	Invalid Command
0x98110015	Unknown Mailbox Protocol Command
0x98110016	Access Denied
0x98110024	Slave error
0x9811002D	Ethernet link cable disconnected
0x98110031	No mailbox support

CoE Error codes	Description	CoE slave abort code
0x98110040	SDO: Toggle bit not alternated	0x05030000
0x98110041	SDO protocol timed out	0x05040000
0x98110042	SDO: Client/server command specifier not valid or unknown	0x05040001
0x98110043	SDO: Invalid block size (block mode only)	0x05040002
0x98110044	SDO: Invalid sequence number (block mode only)	0x05040003
0x98110045	SDO: CRC error (block mode only)	0x05040004
0x98110046	SDO: Out of memory	0x05040005
0x98110047	SDO: Unsupported access to an object	0x06010000
0x98110048	SDO: Attempt to read a write only object	0x06010001
0x98110049	SDO: Attempt to write a read only object	0x06010002
0x9811004A	SDO: Object does not exist in the object dictionary	0x06020000
0x9811004B	SDO: Object cannot be mapped to the PDO	0x06040041
0x9811004C	SDO: The number and length of the objects to be mapped would exceed PDO length	0x06040042
0x9811004D	SDO: General parameter incompatibility reason	0x06040043
0x9811004E	SDO: General internal incompatibility in the device	0x06040047
0x9811004F	SDO: Access failed due to an hardware error	0x06060000
0x98110050	SDO: Data type does not match, length of service parameter does not match	0x06070010
0x98110051	SDO: Data type does not match, length of service parameter too high	0x06070012

CoE Error codes	Description	CoE slave abort code
0x98110052	SDO: Data type does not match, length of service parameter too low	0x06070013
0x98110053	SDO: Sub-index does not exist	0x06090011
0x98110054	SDO: Value range of parameter exceeded (only for write access)	0x06090030
0x98110055	SDO: Value of parameter written too high	0x06090031
0x98110056	SDO: Value of parameter written too low	0x06090032
0x98110057	SDO: Maximum value is less than minimum value	0x06090036
0x98110058	SDO: General error	0x08000000
0x98110059	SDO: Data cannot be transferred or stored to the application	0x08000020
0x9811005A	SDO: Data cannot be transferred or stored to the application because of local control	0x08000021
0x9811005B	SDO: Data cannot be transferred or stored to the application because of the present device state	0x08000022
0x9811005C	SDO: Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of an file error)	0x08000023
0x9811005D	SDO: Unknown code	unknown
0x9811010E	Command not executed	Slave is not present at the bus