

SIEMENS



FAQ • 01/2017

Module and Channel Diagnostics in the User Program

SIMATIC S7-1200 / S7-1500



<https://support.industry.siemens.com/cs/ww/en/view/109480387>

This entry originates from Siemens Industry Online Support. The conditions of use specified there apply (www.siemens.com/nutzungsbedingungen).

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.

Customers are responsible to prevent unauthorized access to their 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 the 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>.

Contents

1	Task	3
2	Basic Information	5
2.1	RALRM	5
2.1.1	TINFO.....	5
2.1.2	AINFO.....	5
2.1.3	Special Point Relating to SIMATIC S7-1200.....	7
3	Function	8
3.1	PLC Data Types	8
3.1.1	typeDiag	8
3.1.2	typeChannelType	8
3.1.3	typeChannel	9
3.1.4	typeChannelDiag.....	9
3.1.5	typeModuleDiag	9
3.2	Blocks	10
3.2.1	FB "1x00Diag82"	10
	Preparations	10
	Program flow chart for the channel diagnostics	13
3.2.2	FC "ModuleDiag"	15
	Extension of the module errors to be displayed.....	15
3.3	Download	16
3.3.1	Channel Diagnostics in the Central Configuration	16
3.3.2	Channel Diagnostics in the Central Structure (Flurry of Messages).....	17
3.3.3	Channel Diagnostics in the Distributed Structure via PROFINET IO.....	19
3.3.4	Module Diagnostics in the Distributed Structure via PROFINET IO.....	20

1 Task

Using the "RALRM" instruction you can receive alarms from central or distributed IO modules (DP slaves or PROFINET IO devices).

You implement module or channel diagnostics of IO modules that support diagnostics by calling the "RALRM" instruction in the diagnostic interrupt OB 82.

Note Further information about which modules support which diagnostics is available under ["What are the differences between the Basic \(BA\), Standard \(ST\), High Feature \(HF\) and High Speed \(HS\) modules of the ET 200SP and ET 200MP?"](#).

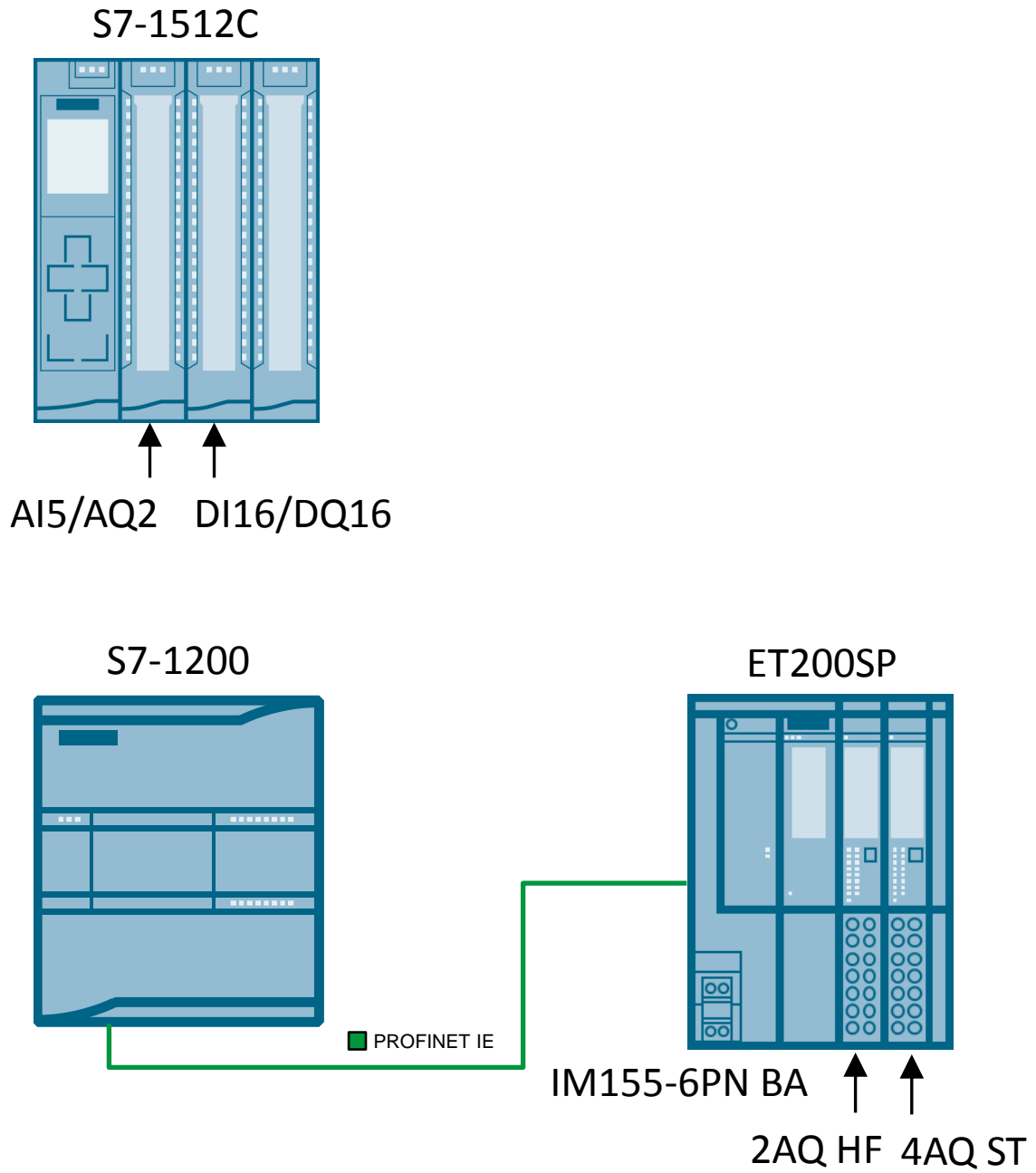
Note The OB 82 is called when the diagnostics status of the modules that support diagnostics changes. Existing errors are not recognized at startup.

Diagnostics via the SIMATIC S7-1200/S7-1500 is illustrated taking the example of the central IO and via PROFINET IO.

Diagnostics is illustrated as follows in the example project.

- The SIMATIC S7-1500 evaluates the channel diagnostics in the central configuration of the Compact CPU 1512C. In this example the submodule "AI5/AQ2" is evaluated. As an example for a flurry of messages the submodule "DI16/DQ16" is also evaluated.
- The SIMATIC S7-1200 evaluates the diagnostics in the distributed configuration of the PROFINET device ET 200SP. Here, channel diagnostics with the module "2AQ HF" and module diagnostics with the module "4AQ ST" are evaluated via the interface module "IM155-6PN BA".

Figure 1-1



© Siemens AG 2017. All rights reserved.

2 Basic Information

2.1 RALRM

Using the "RALRM" instruction you can receive alarms from central or distributed IO modules (DP slaves or PROFINET IO devices).

"RALRM" is called in the organization blocks and via the "TINFO" parameter it delivers the OB start information and via the "AINFO" parameter it delivers header information and additional alarm information.

2.1.1 TINFO

The data structure at the "TINFO" parameter depends on the OB in which the "RALRM" is called. For example, if "RALRM" is called in diagnostic error interrupt OB, you must use the "TI_DiagnosticInterrupt" data structure for "TINFO". The table below gives an overview of the data structures.

Table 2-1

OB no.	OB	TINFO
1, 123...	Cycle OB (program cycle)	TI_ProgramCycle
10	Time-of-day interrupt OB	TI_TimeOfDay
20	Time delay interrupt OB	TI_Delay
30	Cyclic interrupt OB (Cyclic interrupt)	TI_Cyclic
40	Process alarm OB (hardware interrupt)	TI_HWInterrupt
55	Status interrupt OB	TI_Submodule
56	Update alarm OB	
57	OB for manufacturer- or profile-specific interrupt (profile)	
61	Synchronous cycle OB	TI_SynchCycle
80	Time error interrupt OB	TI_TimeError
82	Diagnostic error interrupt OB	TI_DiagnosticInterrupt
83	Pull or plug of modules OB	TI_PlugPullModule
86	Rack or station failure OB	TI_StationFailure
91	MC Servo	TI_Classic
92	MC Interpolator	
100	Startup OB	TI_Startup
121	Programming error OB	TI_ProgIOAccessError
122	IO access error OB	

2.1.2 AINFO

You should independently create the data structures to be transferred at the AINFO parameter as given in the [description](#).

Note

A detailed description of the RALRM instruction is available [here](#) or in the Online Help of the TIA Portal.

The data structure is used in the project for PROFINET IO alarms or central IO alarms. This is structured as below.

Structure of the header information

The header information for PROFINET IO alarms or central IO alarms is stored in the PLC data type "typeHeader".

Table 2-2

Name	Data type	Description
blockType	Word	Block type
blockLength	UInt	Received block length in bytes (from "version" through to and including the last diagnostics event "diag")
version	Word	Version
interruptType	UInt	Identifier for the interrupt type
api	DWord	API (Application Process Identifier)
slotNumber	UInt	Slot number of the component that triggered the alarm
submoduleSlotNumber	UInt	Submodule slot number of the component that triggered the alarm
moduleIdentification	DWord	Module identification; unique information of the alarm source
submoduleIdentification	DWord	Submodule identification; unique information of the alarm source
interruptSpecifier	Word	Alarm specifier

Structure of the additional alarm information

The additional alarm information for PROFINET IO alarms or central IO alarms with the format identifier W#16#8000 (channel diagnostics) are stored in the PLC data type "typeAddInfo".

Table 2-3

Name	Data type	Description
formatIdentifier	Word	Format identifier for the structure of the following data as additional alarm information: W#16#8000: Channel diagnostics
diag	Array[0..233] of "typeAddDiag"	Array with the diagnostics events

The diagnostics events are stored in the PLC data type "typeAddDiag" and are 6 bytes in size.

Table 2-4

Name	Data type	Description
channelNumber	Int	Channel number of the component that triggered the alarm
channelErrorType	Byte	Channel and error types
dataFormat	Byte	Data format
errorCode	Word	Error type

The number of received diagnostic events can be calculated as follows.

$Number\ of\ diagnostics\ events = (Ainfo.header.blockLength - Offset) / |„typeAddDiag“|$

$Offset = \text{Byte of "Ainfo.header.version" up to finally „Ainfo.addInfo.formatIdentifier“} = 24$

$|„typeAddDiag“| = \text{Byte length of the PLC data type "typeAddDiag"} = 6$

$Number\ of\ diagnostics\ events = (Ainfo.header.blockLength - Offset) / |„typeAddDiag“|$

2.1.3 Special Point Relating to SIMATIC S7-1200

WARNING	The AINFO data structure must be in the non-optimized block access of the S7-1200. You ensure this by deselecting the "Optimized block access" attribute in the data block properties (Figure 2-1) or by creating the data structure in an FB with the "Set in IDB" option selected in the "Retain" column of the Properties (Figure 2-2).
----------------	--

Figure 2-1

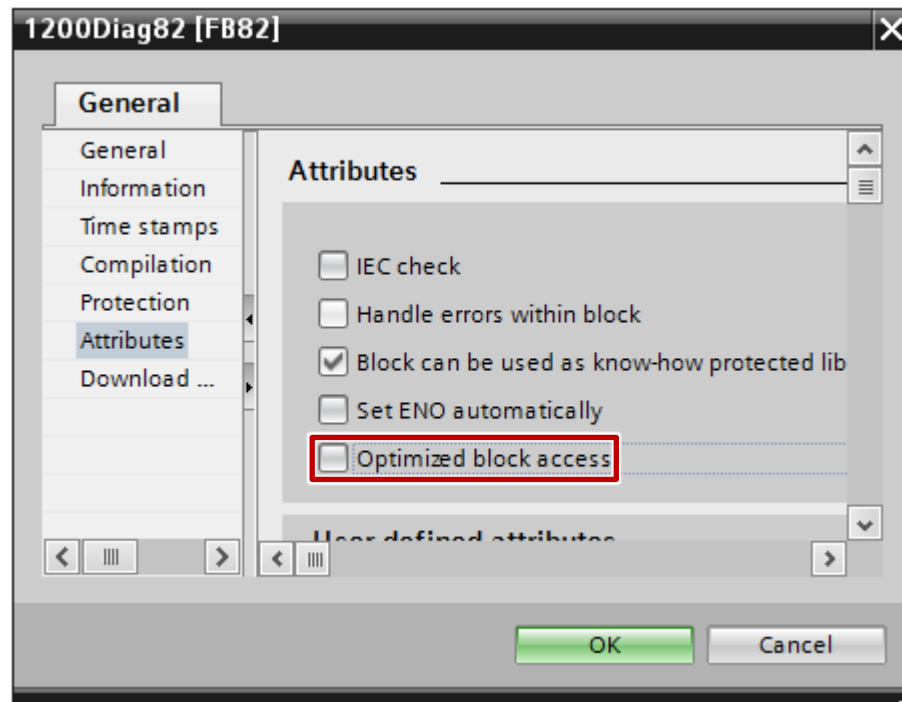


Figure 2-2

1500Diag82				1200Diag82			
Name	Data type	...	Retain	Name	Data type	...	Retain
Input				Input			
Output				Output			
InOut				InOut			
Static				Static			
statTInfo	TI_DiagnosticInterrupt		Non-retain	statTInfo	TI_DiagnosticInterrupt		Non-retain
statAInfo	"typeAiPnOrCentral"		Non-retain	statAInfo	"typeAiPnOrCentral"		Set in IDB

3 Function

3.1 PLC Data Types

The PLC data types below contain the most important information for evaluating the channel diagnostics.

3.1.1 typeDiag

Table 3-1

Name	Data type	Description
error	Boolean	Error occurred in the module.
multiError	Boolean	=TRUE, if multiple errors have occurred.
errorCounter	USInt	<ul style="list-style-type: none"> Number of channel errors in the module with "moduleDiagOnly" = FALSE Number of different channel errors in the module with "moduleDiagOnly" = TRUE
slotNumber	UInt	Slot number of the component that triggered the alarm
submoduleSlotNumber	UInt	Submodule slot number of the component that triggered the alarm
moduleDiagOnly	Boolean	The module supports only the module diagnostics (diagnostics channel "Ainfo.addInfo.channelNumber" = -32768).
input	"typeChannelType"	Diagnostics data of the inputs
output	"typeChannelType"	Diagnostics data of the outputs
inOut	"typeChannelType"	Diagnostics data of the outputs

3.1.2 typeChannelType

Table 3-2

Name	Data type	Description
channel	"typeChannel"	Channel diagnostics data ("moduleDiagOnly" = FALSE)
module	"typeModuleDiag"	Module diagnostics data ("moduleDiagOnly" = TRUE)

3.1.3 typeChannel

Table 3-3

Name	Data type	Description
chMax	SInt	Last channel of the module inputs and outputs
areaLenError	Boolean	Area length error (diagnostics channel "Ainfo.addInfo.channelNumber" is outside the permissible range of 0 to "chMax")
ch	Array[0..x] of "typeChannelDiag"	Field with the channel diagnostics data (x >= "chMax")

WARNING Select the upper limit x of the "ch" field to cover the number of channels to be diagnosed.

3.1.4 typeChannelDiag

Table 3-4

Name	Data type	Description
error	Boolean	Channel fault
errorCode	Word	Error type ("Ainfo.addInfo.errorCode" parameter)

Note The decoding of the error type is given in the [description of the AINFO parameter](#) for PROFINET IO alarms or the central IO alarms for the format identifier ("formatIdentifier") = 16#8000: Channel diagnostics.

3.1.5 typeModuleDiag

The PLC data type "typeModuleDiag" includes the error to be displayed for the module diagnostics.

Table 3-5

Name	Data type	Description
shortCircuit	Boolean	At least one channel of the module has a short circuit.
wireBreak	Boolean	At least one channel of the module has a wire break.
highLimitExceeded	Boolean	At least one channel of the module has exceeded a limit.
lowLimitExceeded	Boolean	At least one channel of the module has exceeded a limit.
noSupplyVoltage	Boolean	Power supply failure of the module

It is one of the most common errors.

With other expected module errors a corresponding Boolean entry per error must be inserted in the PLC data type and the error must be programmed in the FC "ModuleDiag" (see section [3.2.2](#)).

3.2 Blocks

3.2.1 FB "1x00Diag82"

The FB "1x00Diag82" decodes the diagnostics messages of a central or distributed IO module (PROFINET IO device).

It is called for each module to be diagnosed (identified by the hardware identifier) with its own instance and own data buffer storage ("diag") in the diagnostic interrupt OB 82.

Figure 3-1

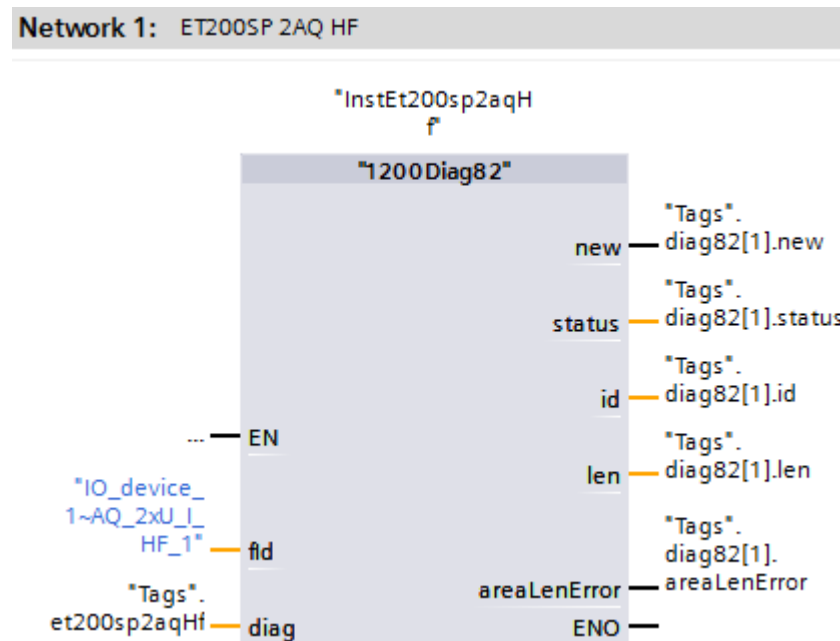


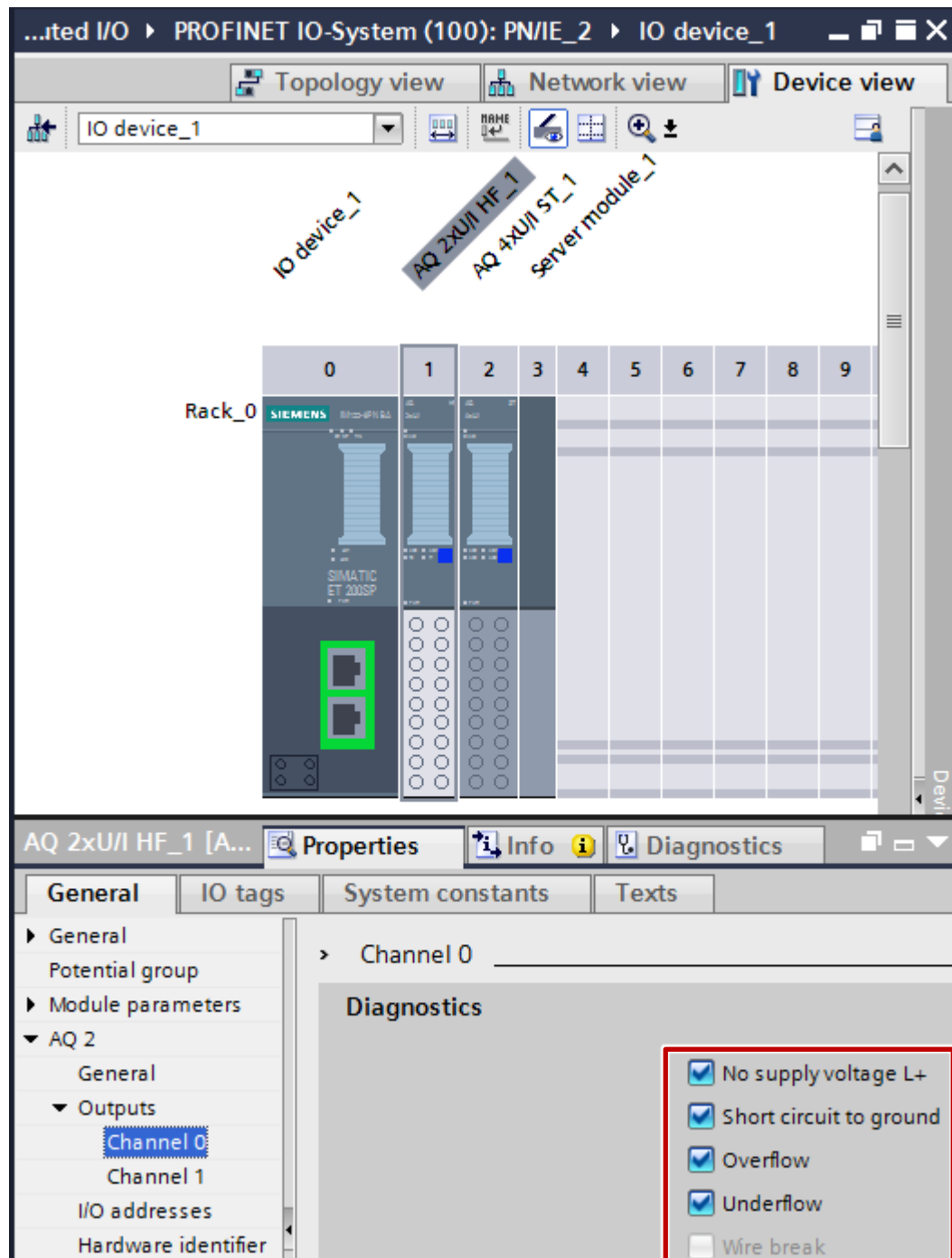
Table 3-6

	Name	Data type	Description
Input	fId	HW_IO	Hardware identifier of the module
Output	new	Boolean	A new alarm has been received.
	status	DWord	Error code of the "RALRM" instruction
	id	HW_IO	Hardware identifier of the module from which the alarm has been received.
	len	UInt	Length of the received alarm information
	areaLenError	Boolean	Area length error (channel diagnostics cannot be written to "diag" – area too small)
InOut	diag	"typeDiag"	Data buffer with the diagnostics data of the module

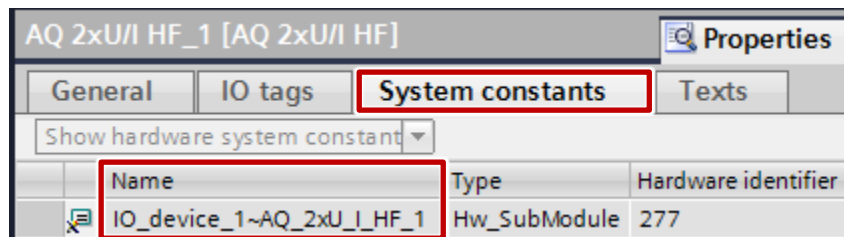
Preparations

1. In the Device View you select the device to be diagnosed.
2. Mark the module to be diagnosed and open the Properties.

3. Enable the desired diagnostics messages for the module to be diagnosed (shown here with the example of the distributed structure of the ET 200SP, module "AQ 2U/I HF", channel diagnostics).



4. Take the hardware identifier "fld" from the properties of the module to be diagnosed ("System constants" item).



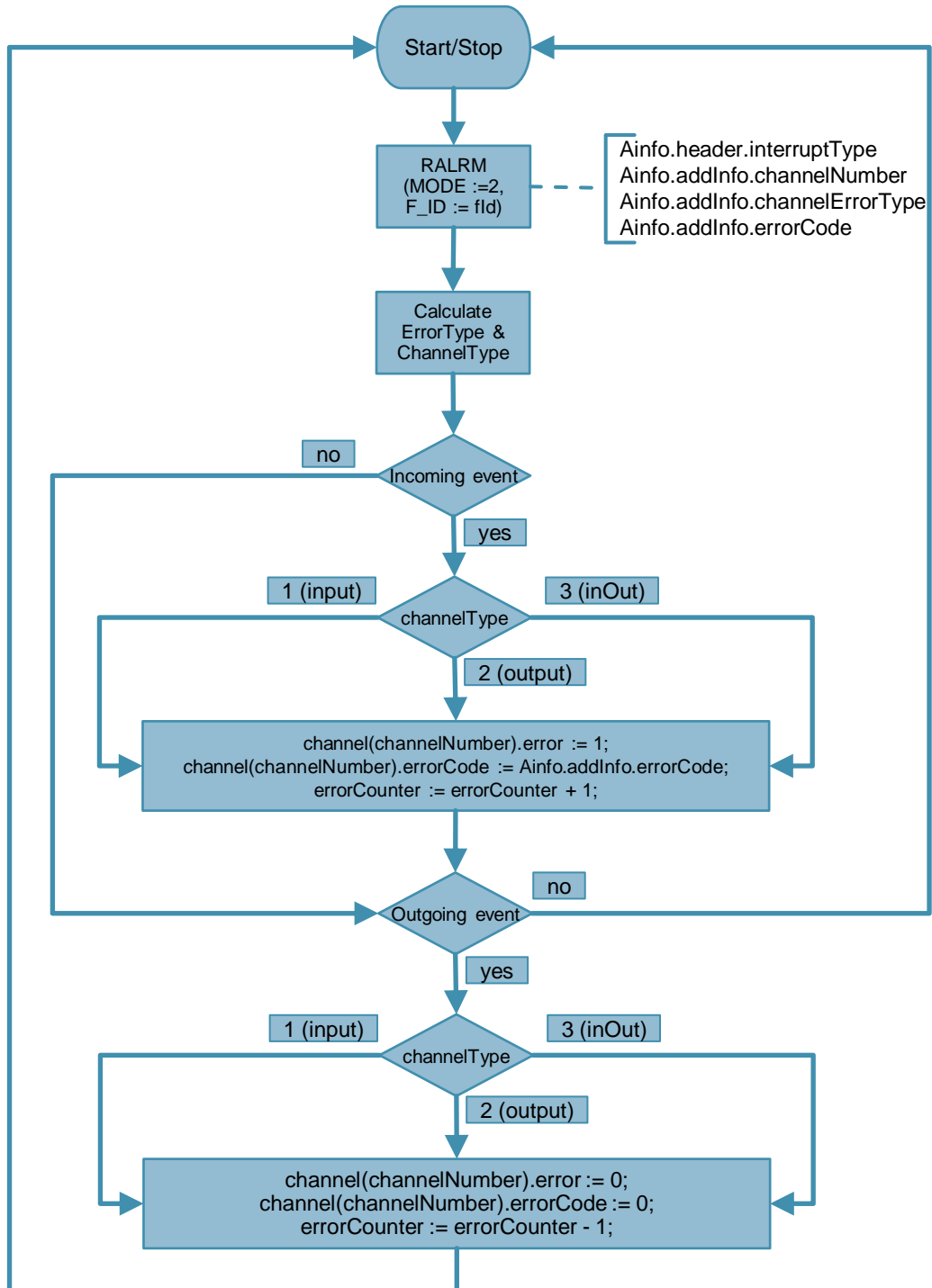
5. Select the upper limit "x" of the "ch" field in the PLC data type "typeChannel" so that the number of channels to be diagnosed (per channel type) is covered (see section 3.1.3).
6. Specify as start value the highest channel number "chMax" per channel type (Input, Output, InOut) in the data buffer with the diagnostics data of the module ("typeDiag"):

Name	Data type	Start value
Static		
diag82	Array[1..2] of "typeDiag82"	
et200sp2aqHf	"typeDiag"	
error	Bool	false
multiError	Bool	false
errorCounter	USInt	0
slotNumber	UInt	0
submoduleSlotNu...	UInt	0
moduleDiagOnly	Bool	false
input	"typeChannelType"	
output	"typeChannelType"	
channel	"typeChannel"	
chMax	SInt	1
areaLenError	Bool	false
ch	Array[0..4] of "typeChannelDiag"	
module	"typeModuleDiag"	
inOut	"typeChannelType"	

Note Point 6 is required for channel diagnostics of the ET 200S in the distributed structure via PROFINET IO. With the outgoing event of the last channel diagnostics all the channel errors (0 to "chMax") are deleted per channel type.

Program flow chart for the channel diagnostics

Figure 3-2



The diagnostic interrupt OB 82 is called at each incoming and each outgoing event of a channel of the connected modules that support diagnostics.

In the FB "1x00Diag82" the RALRM instruction is called in MODE 2 with specification of the module to be diagnosed (hardware identifier "fld"). When the alarm-triggering hardware ID matches, this delivers the OB start information and management information via the "TINFO" parameter, and header information and additional alarm information via the "AINFO" parameter.

When errors occur simultaneously, up to 234 events can be stored in the additional alarm information of the AINFO parameter.

The number of events occurred is determined as follows from the block length and further processed (see section [2.1.2](#)).

$$\text{Number of diagnostics events} = (\text{Ainfo.header.blockLength} - 24) / 6$$

Below are the most important parameters for sorting channel diagnostics.

Table 3-7

Name	Description
Ainfo.header.interruptType	Identifier for the interrupt type
Ainfo.addInfo.channelNumber	Channel number of the component that triggered the alarm
Ainfo.addInfo.channelErrorType	Channel and error types
Ainfo.addInfo.errorCode	Error type

The "Ainfo.addInfo.channelErrorType" parameter gives the "ErrorType" and "ChannelType" information.

With an incoming event ("ErrorType" = 1), depending on the "ChannelType" (1 = Input, 2 = Output and 3 = InOut), the relevant channel error is set, the error type "errorCode" is transferred and the number of channel errors is increased in the "errorCounter" module.

With an outgoing event ("ErrorType" > 1), depending on the "ChannelType" (1 = Input, 2 = Output and 3 = InOut), the relevant channel error is reset, the error type "errorCode" is deleted and the number of channel errors is decreased in the "errorCounter" module.

If the channel number "Ainfo.addInfo.channelNumber" produces the value -32768 (16#8000), the module to be diagnosed does not support channel diagnostics. Then the "moduleDiagOnly" bit is set and evaluation is made via the listed errors in the data type "typeModuleDiag" (see section [3.1.5](#)) and the FC "ModuleDiag" (see section [3.2.2](#)) module for module.

Note For channel diagnostics in the distributed configuration via PROFIBUS DP, the FB "1x00Diag82" must be changed according to the [description of the AINFO parameter](#).

Note Some modules do not support multiple channel diagnostics (SM1232 AQ, for example). In this case the diagnostic error interrupt OB is run through once for the first incoming channel error of the module. Another channel error is reported only when the first channel error has gone. With the counter ("ob82counter") of the diagnostic error interrupt OB you can check the number of alarm calls.

3.2.2 FC "ModuleDiag"

The FC "ModuleDiag" includes the program code for identifying the module diagnostics.

It is called as subprogram in the FB "1x00Diag82".

```
"ModuleDiag" (errorCode := #statAinfo.addInfo.errorCode,
              value := TRUE,
              diag := #diag.input.module,
              errorCounter := #diag.errorCounter);
```

Table 3-8

	Name	Data type	Description
Input	errorCode	Word	Error type ("Ainfo.addInfo.errorCode" parameter)
	value	Boolean	Value transfer: <ul style="list-style-type: none"> • TRUE with incoming error • FALSE with outgoing error
InOut	diag	"typeModuleDiag"	Pointer to PLC data type with the module errors
	errorCounter	USInt	Number of different channel errors in the module

Extension of the module errors to be displayed

If you want to show more errors with the module diagnostics, you have to extend the PLC data type "typeModuleDiag" by the errors to be displayed (see section [3.1.5](#)) and assign the corresponding error type "errorCode" to the error in the program code of the FC "ModuleDiag":

```
CASE WORD_TO_INT(#errorCode) OF
  1:
    #diag.shortCircuit := #value;

  6:
    #diag.wireBreak := #value;

  7:
    #diag.highLimitExceeded := #value;

  8:
    #diag.lowLimitExceeded := #value;

  17:
    #diag.noSupplyVoltage := #value;
END_CASE;
```

Note

The identifiers of the error types are given in the [description of the AINFO parameter](#) for PROFINET IO alarms or the central IO alarms for the format identifier ("formatIdentifier") = 16#8000: Channel diagnostics.

3.3 Download





The download includes a STEP 7 V14 project.

3.3.1 Channel Diagnostics in the Central Configuration

In the project, a CPU 1512C firmware V2.0 is created as an example for channel diagnostics in the central structure (submodule AI5/AQ2).

The function of the FB "1500Diag82" can be checked via the "Ai5Aq2" monitoring table. It is very easy to simulate the "Cable break" error (errorCode = 6) by triggering the unwired analog outputs.

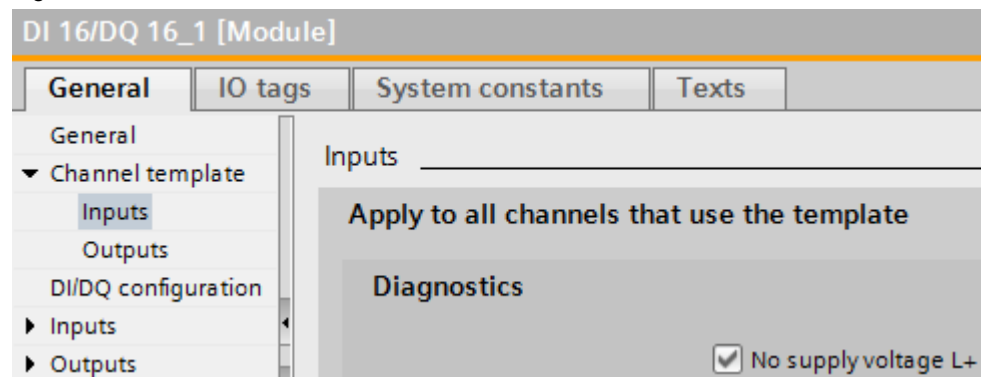
Figure 3-3

Name	Address	Display format	Monitor value	Modify value		
%QW0		Hex	16#1000	16#1000	<input checked="" type="checkbox"/>	
%QW2		Hex	16#1000	16#1000	<input checked="" type="checkbox"/>	
Tags.ob82counter		DEC	2		<input type="checkbox"/>	
Tags.diag82[1].new		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>	
Tags.diag82[1].status		Hex	16#0000_0000		<input type="checkbox"/>	
Tags.diag82[1].id		DEC	259		<input type="checkbox"/>	
Tags.diag82[1].len		DEC	34		<input type="checkbox"/>	
Tags.diag82[1].areaLenError		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>	
Tags.ai5Aq2.error		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>	
Tags.ai5Aq2.multiError		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>	
Tags.ai5Aq2.errorCounter		DEC	2		<input type="checkbox"/>	
Tags.ai5Aq2.slotNumber		DEC	1		<input type="checkbox"/>	
Tags.ai5Aq2.submoduleSlotNumber		DEC	8		<input type="checkbox"/>	
Tags.ai5Aq2.output.channel.chMax		DEC	1		<input type="checkbox"/>	
Tags.ai5Aq2.output.channel.areaLenError		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>	
Tags.ai5Aq2.output.channel.ch[0].error		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>	
Tags.ai5Aq2.output.channel.ch[0].errorCode		Hex	16#0006		<input type="checkbox"/>	
Tags.ai5Aq2.output.channel.ch[1].error		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>	
Tags.ai5Aq2.output.channel.ch[1].errorCode		Hex	16#0006		<input type="checkbox"/>	

3.3.2 Channel Diagnostics in the Central Structure (Flurry of Messages)

Channel diagnostics as flurry of messages is displayed in the central structure on the submodule AI16/DQ16. For this the power supply failure is diagnosed at the inputs.

Figure 3-4



For this the submodule must be supplied via the front connector with a DC 24V power supply (terminals 19 and 20) in compliance with the [connection and block diagram of the digital onboard IO](#).

The function of the FB "1500Diag82" can be checked via the "Di16Dq16" monitoring table (see [Figure 3-5](#)).

When you remove the front connector all 16 input channels display the error "Power supply failure of the module" (errorCode = 16#0011), although the OB 82 was only run through twice (ob82counter = 2). Here, when the RALRM instruction is called, 8 items of channel error information are stored in each case in the AINFO parameter (see DB "InstDi16Dq16".statAinfo.addInfo.diag), which are sorted in a FOR loop in the diagnostics structure.

Figure 3-5

Tags.ob82counter	DEC	2
Tags.diag82[2].new	Bool	<input checked="" type="checkbox"/> TRUE
Tags.diag82[2].status	Hex	16#0000_0000
Tags.diag82[2].id	DEC	260
Tags.diag82[2].len	DEC	76
Tags.diag82[2].areaLenError	Bool	<input type="checkbox"/> FALSE
Tags.di16Dq16.error	Bool	<input checked="" type="checkbox"/> TRUE
Tags.di16Dq16.multiError	Bool	<input checked="" type="checkbox"/> TRUE
Tags.di16Dq16.errorCounter	DEC	16
Tags.di16Dq16.slotNumber	DEC	1
Tags.di16Dq16.submoduleSlotNumber	DEC	9
Tags.di16Dq16.moduleDiagOnly	Bool	<input type="checkbox"/> FALSE
Tags.di16Dq16.input.channel.chMax	DEC+/-	16
Tags.di16Dq16.input.channel.areaLenError	Bool	<input type="checkbox"/> FALSE
Tags.di16Dq16.input.channel.ch[0].errorCode	Hex	16#0011
Tags.di16Dq16.input.channel.ch[1].errorCode	Hex	16#0011
Tags.di16Dq16.input.channel.ch[2].errorCode	Hex	16#0011
Tags.di16Dq16.input.channel.ch[3].errorCode	Hex	16#0011
Tags.di16Dq16.input.channel.ch[4].errorCode	Hex	16#0011
Tags.di16Dq16.input.channel.ch[5].errorCode	Hex	16#0011
Tags.di16Dq16.input.channel.ch[6].errorCode	Hex	16#0011
Tags.di16Dq16.input.channel.ch[7].errorCode	Hex	16#0011
Tags.di16Dq16.input.channel.ch[8].errorCode	Hex	16#0011
Tags.di16Dq16.input.channel.ch[9].errorCode	Hex	16#0011
Tags.di16Dq16.input.channel.ch[10].errorCode	Hex	16#0011
Tags.di16Dq16.input.channel.ch[11].errorCode	Hex	16#0011
Tags.di16Dq16.input.channel.ch[12].errorCode	Hex	16#0011
Tags.di16Dq16.input.channel.ch[13].errorCode	Hex	16#0011
Tags.di16Dq16.input.channel.ch[14].errorCode	Hex	16#0011
Tags.di16Dq16.input.channel.ch[15].errorCode	Hex	16#0011

3.3.3 Channel Diagnostics in the Distributed Structure via PROFINET IO

In the project an S7-1200 CPU firmware V4.1.3 is created as controller with ET 200SP (IM155-6 PN HF V3.2) for the diagnostics in the distributed PN IO structure. The channel diagnostics is displayed on the module 2AQ HF and can be checked via the "Et200sp2aqHf" monitoring table.

[Figure 3-6](#) shows the channel diagnostics taking the example of the simulated errors of "limit exceeded" (errorCode = 8) and "wire break" (errorCode = 6) by addressing the non-wired current output.

Figure 3-6

Name	Address	Display format	Monitor value	Modify value	
%QW64	Hex		16#8000	16#8000	<input checked="" type="checkbox"/>
%QW66	Hex		16#1000	16#1000	<input checked="" type="checkbox"/>
*Tags".ob82counter		DEC	2		<input type="checkbox"/>
*Tags".diag82[1].new		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>
*Tags".diag82[1].status		Hex	16#0000_0000		<input type="checkbox"/>
*Tags".diag82[1].id		Hex	16#0115		<input type="checkbox"/>
*Tags".diag82[1].len		DEC	34		<input type="checkbox"/>
*Tags".diag82[1].areaLenError		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
*Tags".et200sp2aqHf.error		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>
*Tags".et200sp2aqHf.multiError		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>
*Tags".et200sp2aqHf.errorCounter		DEC	2		<input type="checkbox"/>
*Tags".et200sp2aqHf.slotNumber		DEC	1		<input type="checkbox"/>
*Tags".et200sp2aqHf.submoduleSlotNumber		DEC	1		<input type="checkbox"/>
*Tags".et200sp2aqHf.moduleDiagOnly		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
*Tags".et200sp2aqHf.output.channel.chMax		DEC+/-	1		<input type="checkbox"/>
*Tags".et200sp2aqHf.output.channel.areaLenError		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
*Tags".et200sp2aqHf.output.channel.ch[0].error		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>
*Tags".et200sp2aqHf.output.channel.ch[0].errorCode		Hex	16#0008		<input type="checkbox"/>
*Tags".et200sp2aqHf.output.channel.ch[1].error		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>
*Tags".et200sp2aqHf.output.channel.ch[1].errorCode		Hex	16#0006		<input type="checkbox"/>

3.3.4 Module Diagnostics in the Distributed Structure via PROFINET IO

The module diagnostics is displayed on the module 4AQ ST and can be checked via the "Et200sp4aqSt" monitoring table.

Figure 3-7 shows the module diagnostics taking the example of the simulated errors:

- "highLimitExceeded"
Modify value = 16#8000 at the voltage output 0 (+/- 10V)
- "wireBreak"
Modify value = 16#1000 at the non-wired current output 1 (+/- 20mA)
- "lowLimitExceeded"
Modify value = 16#7FFF at the voltage output 2 (0..10V)

Figure 3-7

Name	Address	Display format	Monitor value	Modify value	
	%QW68	Hex	16#8000	16#8000	<input checked="" type="checkbox"/>
	%QW70	Hex	16#1000	16#1000	<input checked="" type="checkbox"/>
	%QW72	Hex	16#7FFF	16#7FFF	<input checked="" type="checkbox"/>
	%QW74	Hex	16#0000		<input type="checkbox"/>
"Tags".ob82counter		DEC	3		<input type="checkbox"/>
"Tags".diag82[2].new		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>
"Tags".diag82[2].status		Hex	16#0000_0000		<input type="checkbox"/>
"Tags".diag82[2].id		Hex	16#0116		<input type="checkbox"/>
"Tags".diag82[2].len		DEC	34		<input type="checkbox"/>
"Tags".diag82[2].areaLenError		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
"Tags".et200sp4aqSt.error		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>
"Tags".et200sp4aqSt.multiError		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>
"Tags".et200sp4aqSt.errorCounter		DEC	3		<input type="checkbox"/>
"Tags".et200sp4aqSt.slotNumber		DEC	2		<input type="checkbox"/>
"Tags".et200sp4aqSt.submoduleSlotNumber		DEC	1		<input type="checkbox"/>
"Tags".et200sp4aqSt.moduleDiagOnly		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>
"Tags".et200sp4aqSt.output.module.shortCircuit		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
"Tags".et200sp4aqSt.output.module.wireBreak		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>
"Tags".et200sp4aqSt.output.module.highLimitExceeded		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>
"Tags".et200sp4aqSt.output.module.lowLimitExceeded		Bool	<input checked="" type="checkbox"/> TRUE		<input type="checkbox"/>
"Tags".et200sp4aqSt.output.module.noSupplyVoltage		Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>
"InstEt200sp4aqSt".statTInfo.channel		DEC	32768		<input type="checkbox"/>
"InstEt200sp4aqSt".statAinfo.header.interruptType		DEC	1		<input type="checkbox"/>
"InstEt200sp4aqSt".statAinfo.addInfo.formatIdentifier		Hex	16#8000		<input type="checkbox"/>
"InstEt200sp4aqSt".statAinfo.addInfo.channelNumber		DEC+/-	-32768		<input type="checkbox"/>
"InstEt200sp4aqSt".statAinfo.addInfo.channelErrorType		Hex	16#48		<input type="checkbox"/>
"InstEt200sp4aqSt".statErrorType		DEC	1		<input type="checkbox"/>
"InstEt200sp4aqSt".statChannelType		DEC	2		<input type="checkbox"/>
"InstEt200sp4aqSt".statAinfo.addInfo.errorCode		Hex	16#0007		<input type="checkbox"/>

The "ST" (Standard) version of the ET 200SP only provides module diagnostics. As diagnostics channel number "Ainfo.addInfo.channelNumber" the RALRM instruction delivers the value -32768 when errors occur.

This value sets the "moduleDiagOnly" bit and signals that no channel diagnostics is available.

Each different error can only be signaled once, because with module diagnostics the diagnostic error interrupt OB is only run through for the first incoming and last outgoing error of an error type "errorCode".