FAQ • 09/2014

Evaluating Diagnostics Data in the User Program

S7-300 / S7-400 CPU

http://support.automation.siemens.com/WW/view/en/101906697

This entry is from the Siemens Industry Online Support. The general terms of use (http://www.siemens.com/terms_of_use) apply.

Security information

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 (cell protection concept, for example) 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.

Contents

1 Introduction

In this document we describe how, with diagnostics-compatible modules, you evaluate a change of the diagnostics status in the user program of the CPU.

If a diagnostics-compatible module, for which you have enabled the diagnostics alarm, detects a change in its diagnostics status, it sends a diagnostics alarm request to the CPU:

- There is a fault or a component must be serviced or both (incoming event).
- There is no longer a fault and no component must be serviced anymore (outgoing event).

The operating system of the CPU then calls the diagnostics alarm OB (OB82).

If a diagnostics alarm is triggered, the defective, diagnostics-compatible module automatically enters 4 bytes of diagnostics data and its start address in the start information of the diagnostics alarm OB (OB82) and in diagnostics buffer. In this way you receive information about when the fault occurred and in which module.

You can evaluate other diagnostics data of the defective, diagnostics-compatible module (channel on which the fault occurred and type of fault) with an appropriate program in the diagnostics alarm OB (OB 82).

You can use the "RDSYSST" instruction to read out module diagnostics data and the "WR_USMSG" instruction to write this data to the diagnostics buffer.

Note The technical data of the module includes information about whether the module supports diagnostics functions and whether the diagnostics data can be read out.

2 Enable the Diagnostics Alarm and Add the Diagnostics Alarm OB (OB82) to the User Program of the CPU

This example shows you how to read out, evaluate and visualize the diagnostics data of a digital input module, SM321 (6ES7321-7BH01-0AB0), for example, in the user program of the S7-300 CPU.

Overview

The following figure shows the hardware configuration of the sample program. The sample program was created with STEP 7 Professional V13.

Figure 2-1

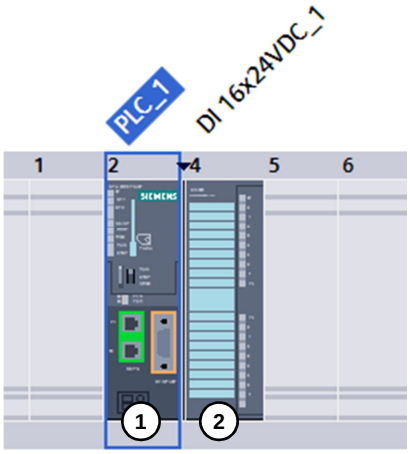


Table 2-1

No.	Component	Article number	Note
1	CPU315-2 PN/DP	6ES7315-2EH14-0AB0	Alternatively you can use any S7-300 CPU.
2	SM321, 16DE, DC24V	6ES7321-7BH01-0AB0	Alternatively you can use any input and output module that supports diagnostics.

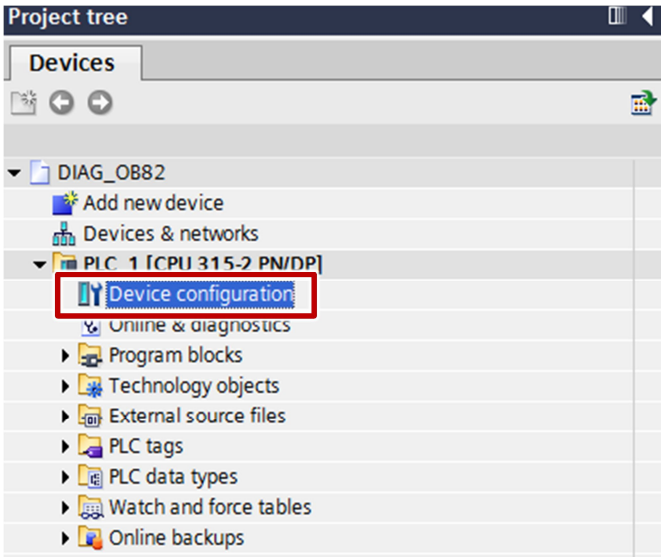
Note The manual below gives information about which S7-300 modules support diagnostics functions.

http://support.automation.siemens.com/WW/view/en/8859629

Enable the diagnostics alarm in the digital input module

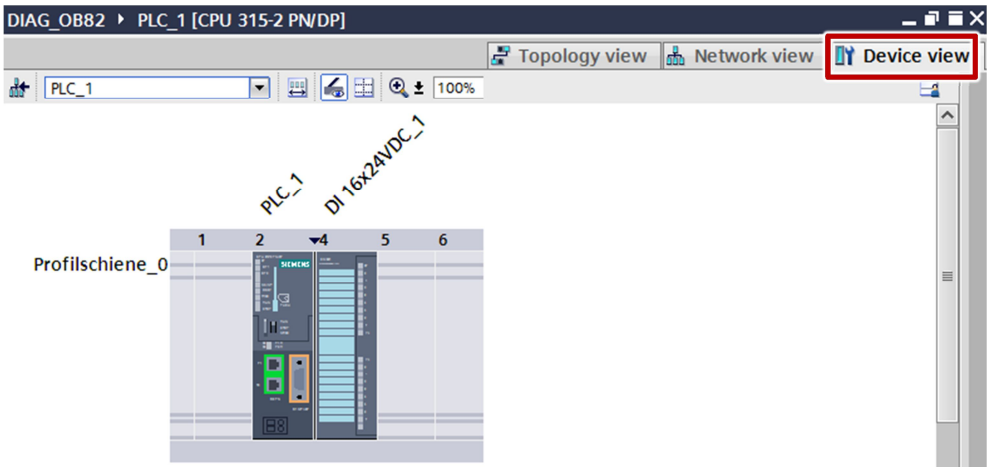
In the project tree you double-click the "Device configuration" entry in the device folder of the S7-300 CPU. The Devices and Networks editor opens in the working area.

Figure 2-2



In the Devices and Networks editor you open the "Device view" tab. The configuration of the S7-300 is displayed.

Figure 2-3

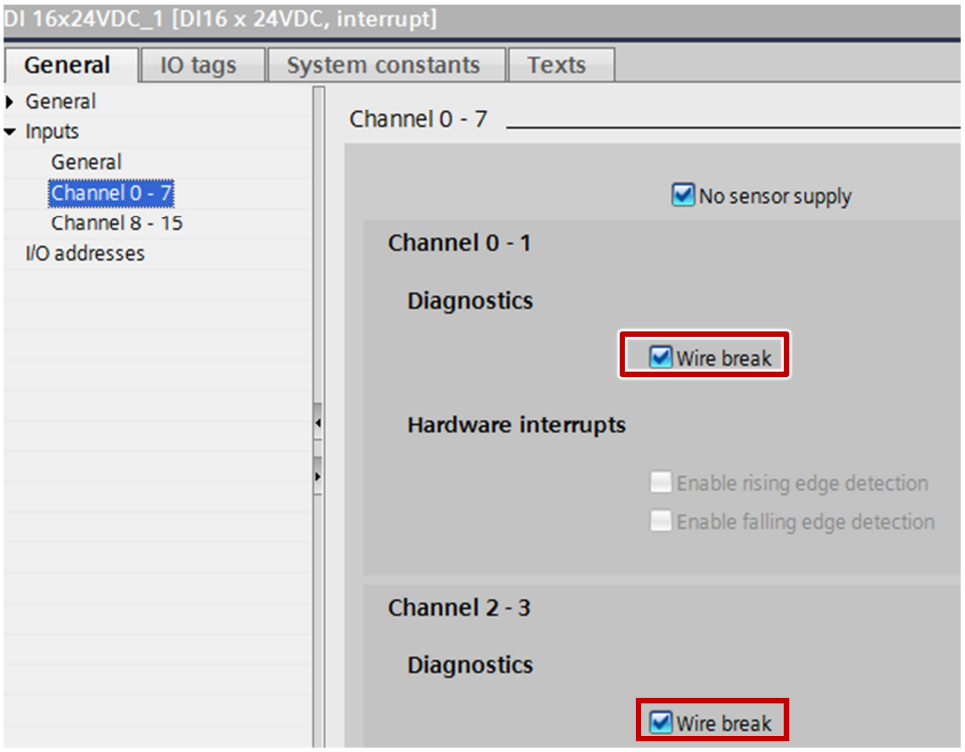


Mark the digital input module. The properties of the digital input module are displayed in the inspector window.

In the "General" tab you navigate to "Inputs > Channel 0 – 7". Enable the "Wire break" diagnostics alarm for channels 0 to 7.

In the "General" tab you navigate to "Inputs > Channel 8 – 15". Enable the "Wire break" diagnostics alarm for channels 8 to 15.

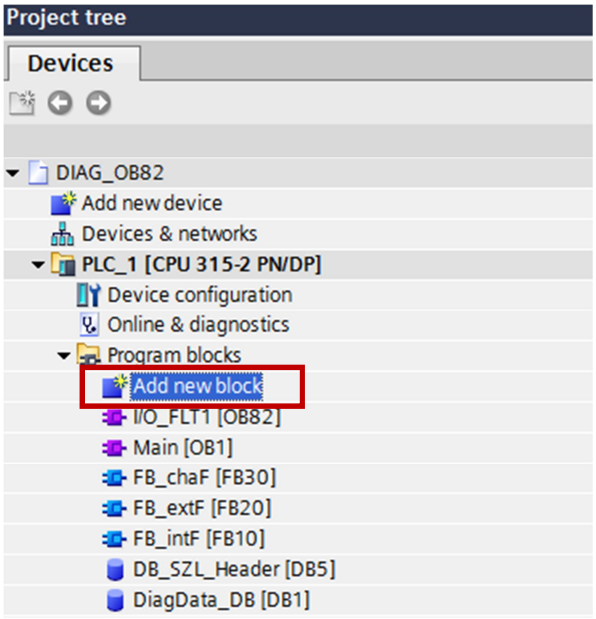
Figure 2-4



Add the Diagnostics Alarm OB (OB82) to the User Program of the S7-300 CPU

In the project tree you double-click the "Add new block" entry in the "Program blocks" folder of the S7-300 CPU. The "Add new block" dialog opens.

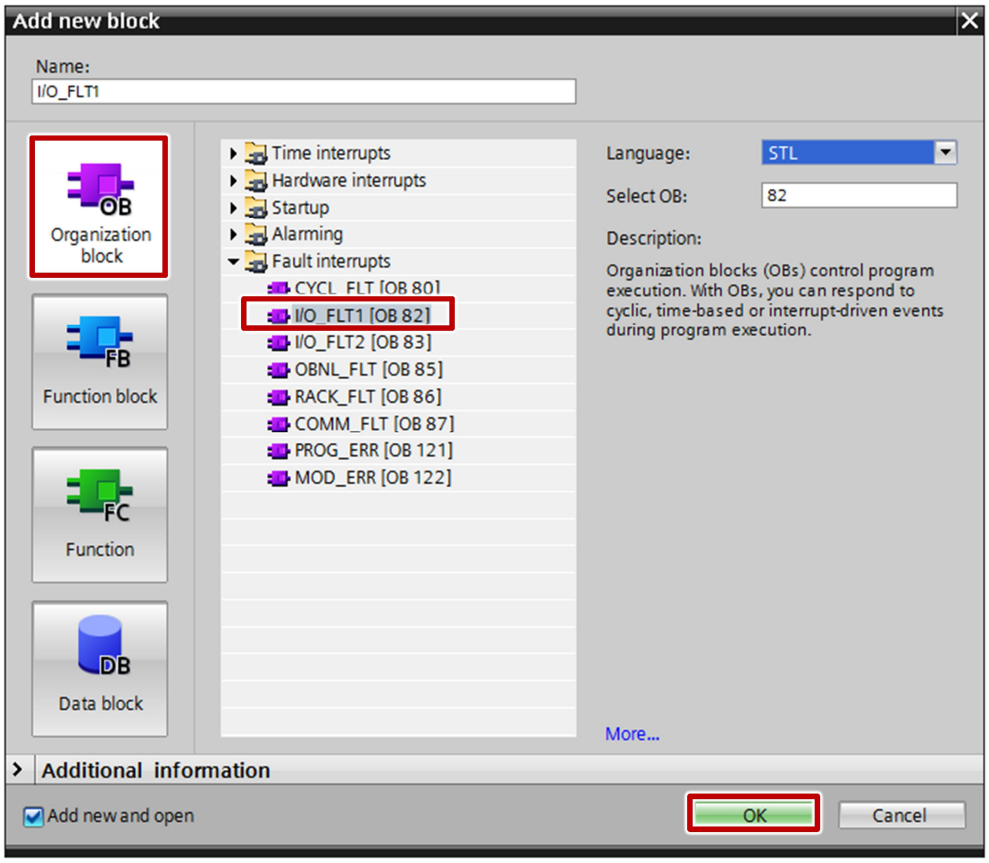
Figure 2-5



Click the "Organization block" button and under "Fault interrupts" you select the diagnostics alarm OB (OB82).

Click the "OK" button. The diagnostics alarm OB (OB82) is added to the user program of the S7-300 CPU.

Figure 2-6

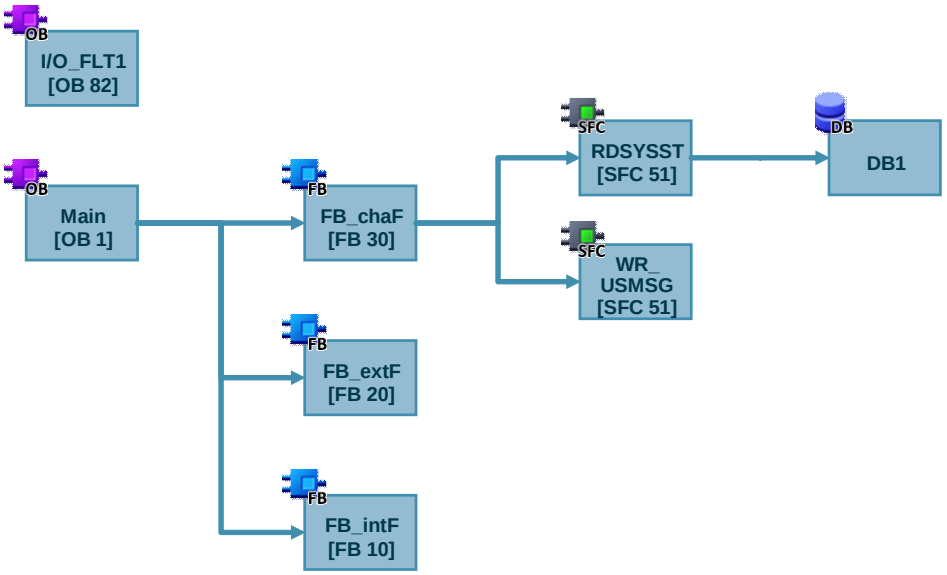


3 User Program

3.1 Overview

The figure below shows an overview of the user program.

Figure 3-1

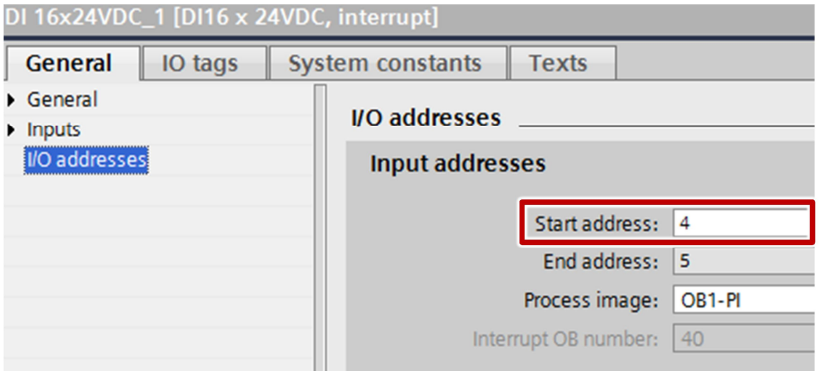


3.2 Diagnostics Alarm OB (OB82)

You need the module address of the digital input module for evaluating the diagnostics data in the diagnostics alarm OB (OB82).

Open Properties of the digital input module to determine the module address. Go to the "General" tab and navigate to "I/O addresses". The start address of the inputs is the module address. In this example the start address of the inputs and thus the module address is 4.

Figure 3-2



In the program of the diagnostics alarm OB (OB82) it is evaluated whether the diagnostics alarm has been triggered by an internal or external fault of the digital input module.

If there is an internal fault, the "OB82_INT_FAULT" tag of the diagnostics alarm OB (OB82) is set to "1". Then the marker M20.1 "internal_failure" is set to "1" in the OB82. When the marker M20.1 "internal_failure" is set to "1", the FB10 "FB_intF" is called and executed in the OB1.

If there is an external fault, the "OB82_EXT_FAULT" tag of the diagnostics alarm OB (OB82) is set to "1". Then the marker M20.2 "external_failure" is set to "1" in the OB82. When the marker M20.2 "external_failure" is set to "1", the FB20 "FB_extF" is called and executed in the OB1.

A channel fault is an external fault. There is additional diagnostics information about this external fault which can be read out with the "RDSYSST" instruction. If there is a channel fault, the "OB82_PNT_INFO" tag of the diagnostics alarm OB (OB82) is set to "1". Then the marker M20.0 "channel_failure" is set to "1" in the OB82. When the marker M20.0 "channel_failure" is set to "1", the FB30 "FB_chaF" is called and executed in the OB1. The "RDSYSST" instruction is called in the FB30 "FB_chaF" to read the diagnostics information. When the diagnostics information has been read, the M20.0 "channel_failure" is reset.

Figure 3-3

```
      L     4                     //module address of the digital input module
      L     #OB82_MDL_ADDR
      ==I
      SPBN  END                   // if the module address of the OB82 is 4, do not jump

// here we do everything for module on address 4
// b#16#39 for entering the diagnostic, b#16#38 for leaving the diagnostic

      L     b#16#39
      L     #OB82_EV_CLASS
      ==I
      SPBN  MOOK                  //if no entering diagnostic > module is ok

//here we do everything for module on address 4 an entering diagnostic

      U     #OB82_EXT_FAULT
      S     "external_failure"    //external fault marker
      SPBN  NEXF



      U     #OB82_INT_FAULT
      S     "internal_failure"


      U     #OB82_PNT_INFO
      SPBN  END                   //jump not if there is more channel informations available

      S     "channel_failure"
      SPA   END


NEXF: NOP 0
//here you may react to internal faults
      SET
      =     "internal_failure"
      SPA   END


MOOK: NOP 0
//reset all markers if no entering diagnostic > the module is ok
      CLR
      =     "channel_failure"
      =     "internal_failure"
      =     "external_failure"
      SPA   END

END:  NOP 0
      BEA
```

3.3 OB1

When the marker M20.0 "channel_failure" is set to "1", the FB30 "FB_chaF" is called and executed.

3.4 FB30 "FB_chaF"

3.4.1 Read Diagnostics Data with the System Function SFC51 "RDSYSST"

In this example, the "RDSYSST" instruction is called in the FB30 "FB_chaF" with the SZL_ID w#16#B3 and access is made to the digital input module with module address 4 which set the diagnostics alarm. The read process is executed immediately.

Figure 3-4

```
CALL   RDSYSST
       REQ          :=TRUE
       SZL_ID       :=w#16#00B3
       INDEX        :=w#16#4
       RET_VAL      :="Tag_3"
       BUSY         :="Tag_4"
       SZL_HEADER   :="DB_SZL_Header".SZL_HEADER
       DR           :=P#DB1.DBX0.0 BYTE 100
```

Input parameters of the "RDSYSST" instruction

The "RDSYSST" instruction has the following input parameters.

Table 3-1

Input parameters	Data type	Description
REQ	BOOLEAN	REQ=1: Trigger for processing
SZL_ID	WORD	SZL_ID of the parts list or of the excerpt of the parts list The SZL_ID w#16#B3 is used in this example. With this SZL_ID all the diagnostics data of a module (diagnostics data record DS1) is read. An overview of the SZL_IDs for the S7-300 CPUs and ET 200 CPUs is available in the operations list in Entry ID 31977679. An overview of the SZL_IDs for the S7-400 CPUs is available in the STEP 7 Online Help and in the manual "STEP 7 Professional V13" in Entry ID 89515142.
INDEX	WORD	Type or number of an object in a parts list When you read out the SZL_ID w#16#B3, you specify the module address at the INDEX parameter. In this example it is the start address of the inputs.

Output parameters of the "RDSYSST" instruction

The "RDSYSST" instruction has the following output parameters.

Table 3-2

Output parameters	Data type	Description
RET_VAL	INT	If an error occurs during processing of the instruction, the RET_VAL parameter receives an error code.
BUSY	BOOLEAN	TRUE: Read procedure has not yet finished
SZL_HEADER	STRUCT	The structure of the data structure "SZL_HEADER" is given in section 3.4.2.
DR	ANY	Target area for the read SZL parts list or the read excerpt of SZL parts list. • If you have only read out the header information of an SZL parts list, you must not evaluate DR, but only SZL_HEADER. • Otherwise the product of LENTHDR and N_DR indicates how many bytes were entered in DR. The structure of the data structure "SZL_HEADER" is given in section 3.4.2.

3.4.2 Structure of the Data Structure "SZL_HEADER"

The figure below shows the structure of the data structure "SZL_HEADER".

Figure 3-5

DB_SZL_Header

	Name	Data type	Offset	Start value
1	Static			
2	SZL_HEADER	Struct	0.0	
3	LENTHDR	Word	0.0	16#0
4	N_DR	Word	2.0	16#0

LENTHDR is the length of a data record of the SZL parts list or of the excerpt of the SZL parts list.

- If you have only read out the header information of an SZL parts list, N_DR contains the number of the associated data records available.
- Otherwise N_DR contains the number of data records transferred to the target area.

3.4.3 Diagnostics Data Structure

Table 3-3 shows the structure and content of the diagnostics data of a module.

The diagnostics data of a module is in the data records 0 and 1 of the system data area.

- The data record 0 contains 4 bytes of diagnostics data which describes the current status of a signal module.
- Data record 1 has the following content:
  - the 4 bytes of diagnostics data that are also in data record 0 and
  - the module-specific diagnostics data.

If an error occurs, the corresponding bit is set to "1".

Table 3-3

Byte	Bit	Meaning	Data record
0	0	Module fault	0 and 1
	1	Internal fault	
	2	External fault	
	3	Channel fault	
	4	No external auxiliary voltage	
	5	No front connector	
	6	No parameters	
	7	Incorrect parameters in the module	
1	0-3	Module class: 0101: Analog module 0000: CPU 1000: Function module 1100: CP 1111: Digital module 0011: DP standard slave 1011: I slave 0100: IM	0 and 1
	4	Channel information available	
	5	User information available	
	6	Diagnostics alarm of proxy	
	7	Maintenance required (only for PROFINET IO)	
2	0	Memory module incorrect or missing	0 and 1
	1	Communication fault	
	2	Operating state	
	3	Cycle time monitoring addressed	
	4	Internal module power supply failure	
	5	Empty battery	
	6	Total backup failure	
	7	Maintenance request (only for PROFINET IO)	
3	0	Extension device failure	0 and 1
	1	Processor failure	
	2	EPROM fault	
	3	RAM fault	
	4	ADU/DAU fault	
	5	Fuse failure	
	6	Lost process alarm	
	7	Reserved	
4	0-6	Channel type: B#16#70: digital input B#16#72: digital output B#16#71: analog input B#16#73: analog output B#16#74: FM POS B#16#75: FM REG	1

Byte	Bit	Meaning	Data record
		B#16#76: FM COUNT B#16#77: FM TECHNO B#16#78: FM NCU B#16#79 to B#16#7D: reserved B#16#7E: US300 B#16#7F: reserved	
	7	Other channel type available: 0: no 1: yes	
5	0-7	Number of diagnostics bits that a module outputs per channel	1
6	0-7	Number of channels of same type in a module	1
7	0	Channel fault Channel 0 / Channel group 0 (digital input channel 0 and 1, for example)	1
	1	Channel fault Channel 1 / Channel group 1 (digital input channel 2 and 3, for example)	
	2	Channel fault Channel 2 / Channel group 2 (digital input channel 4 and 5, for example)	
	3	Channel fault Channel 3 / Channel group 3 (digital input channel 6 and 7, for example)	
	4	Channel fault Channel 4 / Channel group 4 (digital input channel 8 and 9, for example)	
	5	Channel fault Channel 5 / Channel group 5 (digital input channel 10 and 11, for example)	
	6	Channel fault Channel 6 / Channel group 6 (digital input channel 12 and 13, for example)	
	7	Channel fault Channel 7 / Channel group 7 (digital input channel 14 and 15, for example)	
…		Channel-specific diagnostics data (see Table 3-4)	1

Structure of Channel-specific Diagnostics Data

Table 3-4 shows the structure of the channel-specific diagnostics data. The channel-specific diagnostics data contains information about a wire break on a channel or channel group, for example. The example shows the structure of the diagnostics byte for a digital input channel.

Table 3-4

Bit	Meaning	Remarks
0	Configuration/parameter fault	Can be reported with the instruction "WR_USMSG" (SFC52) and EVENTN=w#16#8x70
1	Ground fault	Can be reported with the instruction "WR_USMSG" (SFC52) and EVENTN=w#16#8x71

Bit	Meaning	Remarks
2	P short circuit (encoder)	Can be reported with the instruction "WR_USMSG" (SFC52) and EVENTN=w#16#8x72
3	M short circuit	Can be reported with the instruction "WR_USMSG" (SFC52) and EVENTN=w#16#8x73
4	Wire break	Can be reported with the instruction "WR_USMSG" (SFC52) and EVENTN=w#16#8x74
5	No encoder supply	Can be reported with the instruction "WR_USMSG" (SFC52) and EVENTN=w#16#8x75
6	0	
7	0	

To detect a wire break on a digital input channel in the user program you evaluate Bit 4 of the diagnostics byte. If Bit 4 is set to "1", there is a wire break.

3.4.4 Detect Wire Break on Channel 0 and/or Channel 1 of a Digital Input Module

The diagnostics data is stored in data block DB1. . In this example the diagnostics data read out (data record 1) has a length of 16 bytes. This information is in the data structure "SZL_HEADER" (see Figure 3-6).

Figure 3-6

DB_SZL_Header

	Name	Data type	Offset	Start value	Monitor value
1	Static				
2	SZL_HEADER	Struct	0.0		
3	LENTHDR	Word	0.0	16#0	16#0010
4	N_DR	Word	2.0	16#0	16#0001

The channel-specific diagnostics data is 8 bytes long and is stored in data block DB1 starting at byte 8.

Table 3-5

Data block DB1	Description
Byte 0-7	Diagnostics bytes 0 to 7 of data record 1
Byte 8	Diagnostics byte for digital input channels 0 and 1
Byte 9	Diagnostics byte for digital input channels 2 and 3
Byte 10	Diagnostics byte for digital input channels 4 and 5
Byte 11	Diagnostics byte for digital input channels 6 and 7
Byte 12	Diagnostics byte for digital input channels 8 and 9
Byte 13	Diagnostics byte for digital input channels 10 and 11
Byte 14	Diagnostics byte for digital input channels 12 and 13
Byte 15	Diagnostics byte for digital input channels 14 and 15

To detect a wire break on channel 0 and/or channel 1 of a digital input module you evaluate Bit 0 in Byte 7 and Bit 4 in Byte 8 of the diagnostics data read out (data record 1). If both bits are set to "1", there is a wire break on channel 0 and/or channel 1. In that case the user-defined diagnostics event "Wire break" is written to the diagnostics buffer of the CPU (see below). The data word #INFO1 is set to the value 0 (for channel 0) and the data word #INFO2 is set to the value 1 (for channel 1).

Figure 3-7

```
      U     %DB1.DBX7.0
      U     %DB1.DBX8.4

      S     "wire_break_0_1"

      U     "wire_break_0_1"
      SPBN  FBCE

      L     0
      T     #INFO1

      L     1
      T     #INFO2


      CALL  WR_USMSG
         SEND     :=TRUE
         EVENTN   :=w#16#8174
         INFO1    :=#INFO1
         INFO2    :=#INFO2
         RET_VAL  :=#varRET_VAL
```


3.4.5 SFC52 "WR_USMSG"

With system function SFC52 "WR_USMSG" (write user element to diagnostics buffer) you write a user-defined diagnostics event to the diagnostics buffer. Furthermore, you can send the associated diagnostics message to all the users logged on for that (by setting the input parameter SEND = TRUE).

Figure 3-8

```
      CALL  WR_USMSG
         SEND     :="wire_break_0_1"
         EVENTN   :=w#16#8174
         INFO1    :=#INFO1
         INFO2    :=#INFO2
         RET_VAL  :=#varRET_VAL
```

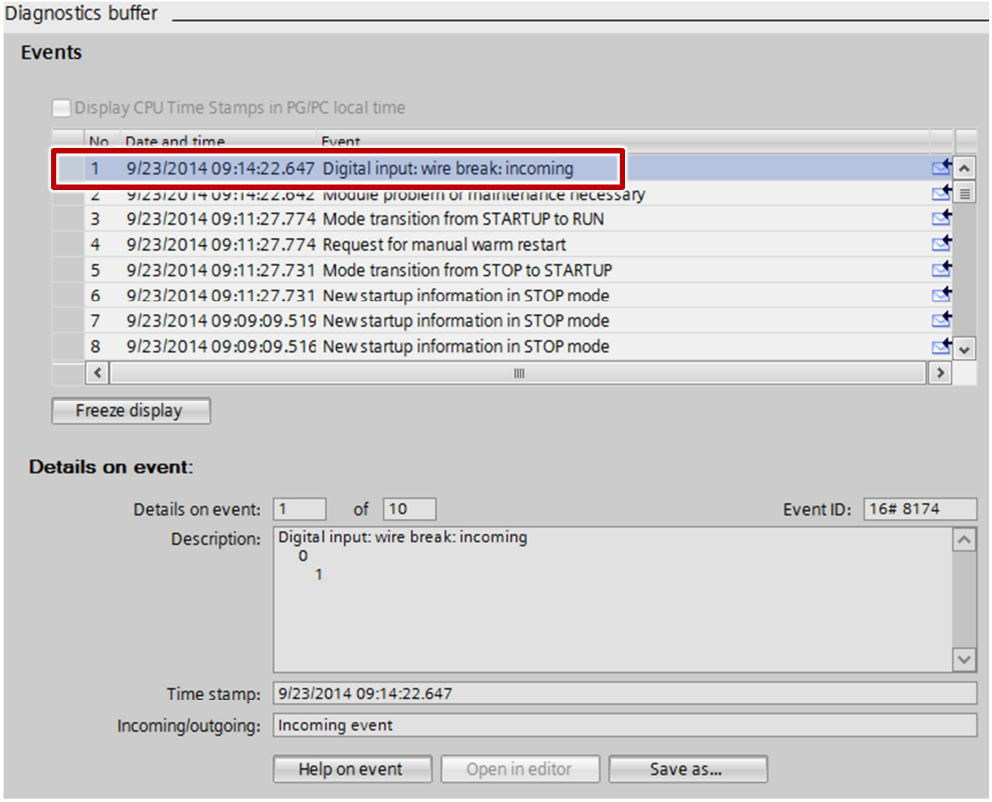
The "WR_USMSG" instruction has the following input parameters.

Table 3-6

Input parameters	Data type	Description
SEND	BOOLEAN	Enable sending of user-defined diagnostics message to all logged on users.
EVENTN	WORD	Event ID With event ID w#16#8174 the wire break is entered as incoming event in the diagnostics buffer of the CPU.
INFO1	ANY	Additional information 1 word long
INFO2	ANY	Additional information 2 words long

The figure below shows the user-defined event that is written to the diagnostics buffer of the CPU if there is a wire break on channel 0 and/or channel 1.

Figure 3-9



3.5 FB20 "FB_extF"

The FB20 "FB_extF" is called and executed in OB1 if an external fault is signaled by the digital input module. Here you can react accordingly to an external fault.

3.6 FB10 "FB_intF"

The FB10 "FB_intF" is called and executed in OB1 if an internal fault is signaled by the digital input module. Here you can react accordingly to an internal fault.