SIEMENS

Ingenuity for life

Sending of SNMP Traps with S7-PLCs

S7-300 / S7-1500 / SINEMA Server

https://support.industry.siemens.com/cs/ww/en/view/57249109

Siemens Industry Online Support

Legal information

Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG ("Siemens"). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.

Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.

By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

Other information

Siemens reserves the right to make changes to the application examples at any time without notice. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.

The Siemens terms of use (https://support.industry.siemens.com) shall also apply.

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 constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the Internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit https://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at: https://www.siemens.com/industrialsecurity.

Table of contents

1 Task

1.1 Overview

Introduction

Monitoring and analyzing industrial networks with PROFINET nodes is an important task of the automation technology. Consequent and transparent execution can prevent failures and production losses.

Network management systems such as the SINEMA server provide an overview of the networks and monitors them, using protocols such as ICMP, LLDP and SNMP.

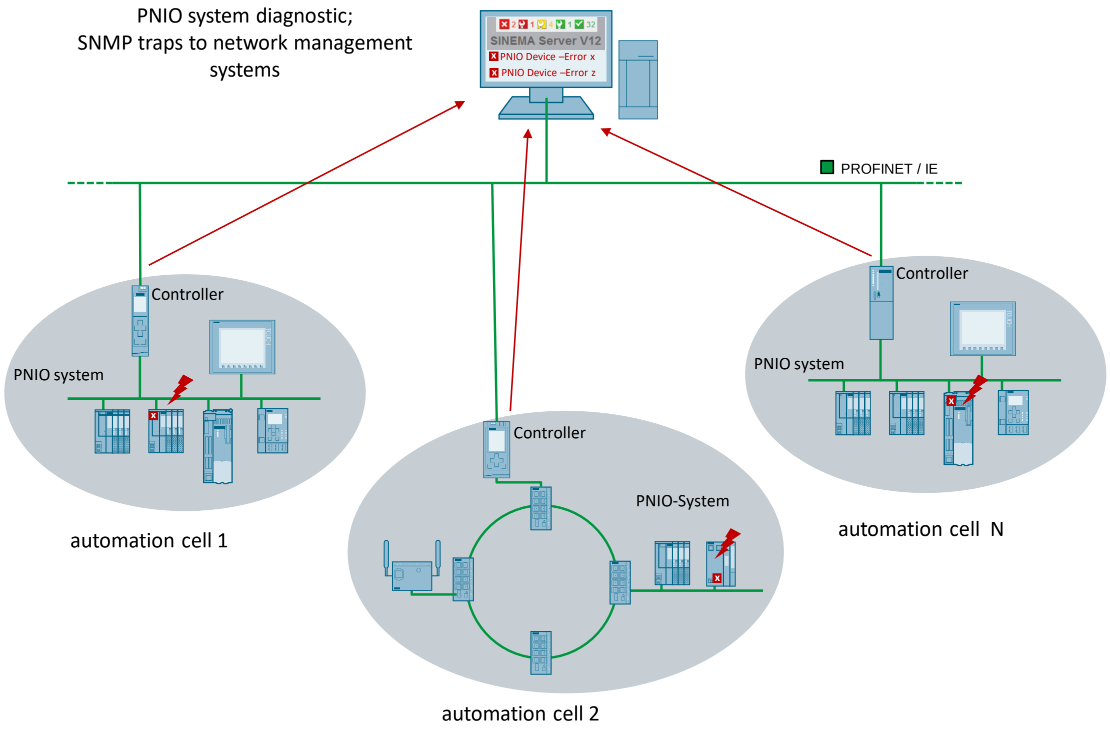
Network nodes are sending messages to the network management system via SNMP traps in order to display changes of the network status.

This application shows how user-defined traps can be sent from a S7 PLC. This makes it possible, for example, to send system diagnostic information of a PNIO device in place of the device to the network management system.

Overview of the automation task

The figure below provides an overview of the automation task.

Figure 1-1



1.2 Requirements

Table 1-1

Requirement	Explanation
Analysis of diagnostic interrupts of the distributed I/O with the help of mechanisms of the SIMATIC controllers.	General errors of the distributed I/O are detected and processed.
Preparing the diagnostic information and messages for display as SNMP trap.	The relevant information is extracted and prepared as SNMP trap from the errors detected on the distributed I/O.
Sending of the prepared messages to a network management system via SNMP traps across SNMPv1. The block used can send user-specific SNMP traps and is provided separately to the application example for S7-300 and S7-1500 in a library.	With the help of the following information a SNMP trap is sent: • IP address of the trap-triggering device • TRAP ID • OID • Variable description
For the PLCs of the series S7-300 and S7-1500, user programs with similar functions are to be provided.	The system architecture of the S7-1500 PLCs differs from the system architecture of the S7-300 PLCs. Accordingly, it is necessary to create independent user programs for each controller.

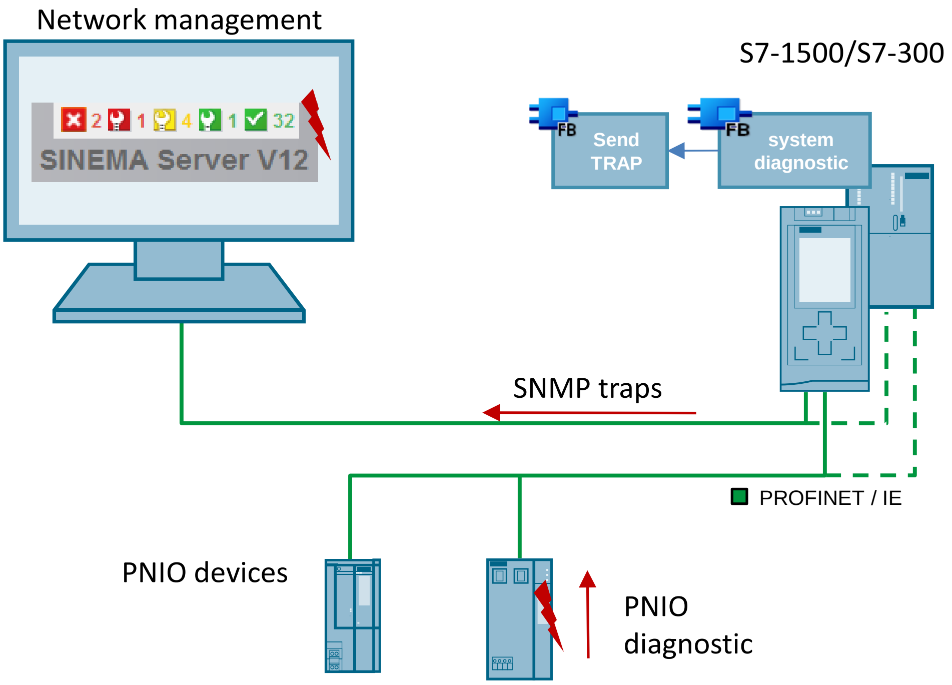
2 Solution

2.1 Overview

Schematic layout

The figure below shows a schematic overview of the most important components of the solution:

Figure 2-1



Note A user program for the S7-300 PLCs is provided. An explanation of the differences can be found in chapter 4.5. The following chapters describe mainly the behavior of the S7-1500 PLC.

Advantages

This application offers the user the following advantages:

- Provision of a universal communication block for sending user-specific traps to SNMPv1.
- Demonstration of analysis and processing of incoming diagnostic interrupts of the distributed I/O.

Topics not covered by this application

This application does not include a description of:

- STEP 7 V15 and STEP 7 V5.5
- the SCL/STL/LAD/FBD programming languages.
- the distributed I/O systems used (ET 200S, ET 200SP)

Basic knowledge of these topics is assumed.

2.2 Description of the core functionality

The example application consists logically from the parts outlined in the following subchapters.

2.2.1 Evaluating the diagnostic information

Figure 2-2

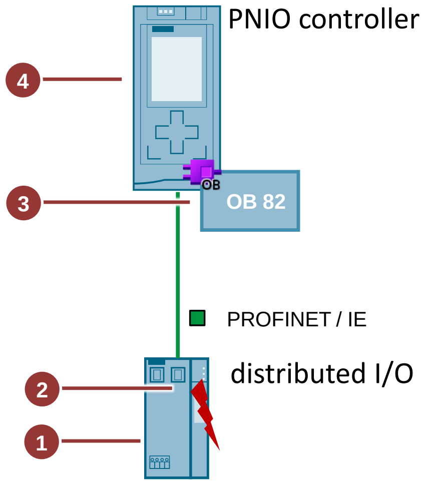


Table 2-1

No.	Process	Remark
1.	The distributed I/O is configured in a way so that a diagnostic interrupt is sent to the PNIO controller in the case of certain events.	The diagnostic interrupts that can be set, differ within the different distributed I/O systems. Example for a diagnostic interrupt: Channel monitoring of the input module (diagnostic interrupt in the event of wire break on the channel).
2.	A diagnostic event occurs (for example, wire break, missing supply voltage, etc.).	If there is a diagnostic event, the PNIO controller will report this to the PNIO controller.
3.	The OB82 (diagnostic interrupt) is called in the PLC based on the diagnostic event and, if programmed, run through.	With the call of the OB82 the diagnostic information is provided and stored for further processing (for example, HW-ID, …)

No.	Process	Remark
4.	For the sending of a SNMP trap • the incoming information for a coming diagnostic event in a cyclic OB is evaluated and prepared. • the IP address of the PNIO device that has caused the diagnostic interrupt is read out. • For an outgoing diagnostic event, the module is checked for trouble free operation and a trap with different TRAP ID than for the coming event is sent.	The trap sent by the PLC is detected as trap of the device by the network management system since the IP of the device is transferred in the frame.

2.2.2 Sending of SNMP traps

Figure 2-3

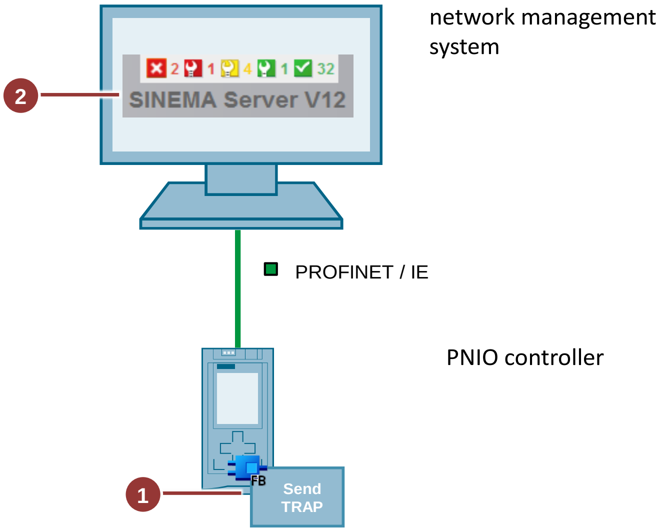
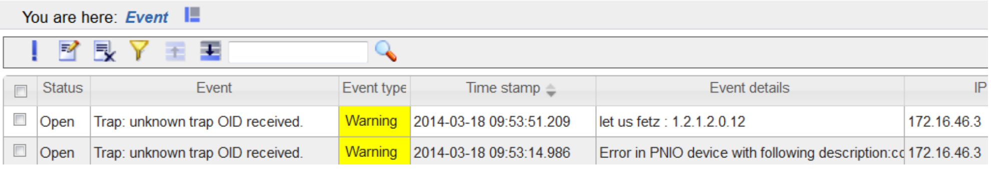


Table 2-2

No.	Process	Remark
1.	With the information from Figure 2-2 a SNMP trap frame is created and sent to the network management system.	The block can also be used self-sufficiently without the previous system diagnostic and send individual SNMP traps.
2.	Depending on the configuration of the network management system, the incoming trap is visually displayed or its information is prepared.	Figure 2-4 shows the display of unknown TRAP IDs in the SINEMA Server V12.

Figure 2-4



You are here: Event

□	Status	Event	Event type	Time stamp	Event details	IP
□	Open	Trap: unknown trap OID received.	Warning	2014-03-18 09:53:51.209	let us fetz : 1.2.1.2.0.12	172.16.46.3
□	Open	Trap: unknown trap OID received.	Warning	2014-03-18 09:53:14.986	Error in PNIO device with following description:cc	172.16.46.3

Figure 2-4 shows the input of an unknown trap on the SINEMA Server V12 network management system. The display of the traps depends on the network management system used.

2.3 Hardware and software components

2.3.1 Validity

This application is valid for

- STEP 7 V15.1
- STEP 7 V5.5 SP3
- S7-1500
- S7-300

2.3.2 Components used

The application was created using the following components:

Hardware components for PLC 1516-3 PN/DP

Table 2-3

Component	Qty.	Article number	Note
PS 60W 24/48/60VDC	1	6ES7 505-0RA00-0AB0	Alternatively, other 24 V power supplies can also be used.
PLC 1516-3 PN/DP	1	6ES7 516-3AN00-0AB0	Alternatively, other S7-1500 PLCs can also be used.
Engineering station	1		

Alternative hardware components for PLC 315-2 PN/DP

Component	Qty.	Article number	Note
PS 307 5A	1	6ES7 307-1EA00-0AA0	Alternatively, other 24 V power supplies can also be used.
PLC 315-2 PN/DP	1	6ES7 315-2EH14-0AB0	Alternatively, other S7-300 PLCs PROFINET interface can also be used.
Engineering station	1		

Hardware components of the distributed I/O

Component	Qty.	Article number	Note
IM 151-3 PN	1	6ES7 151-3AA23-0AB0	Head module of the ET 200S
PM-E DC24V	1	6ES7 138-4CA01-0AA0	
4DI x DC24V ST	1	6ES7 131-4BD01-0AA0	
IM 155-6 PN ST	1	6ES7 155-6AU00-0BN0	Head module of the ET 200SP
DI8 x DC24V ST	1	6ES7 131-6BF00-0BA0	
Server module	1	6ES7 193-6PA00-0AA0	

Software components

Table 2-4

Component	Qty.	Article number	Note
SIMATIC STEP 7 PROFESSIONAL V15.1	1	6ES7822-1AA03-0YA5	You can also use a higher version of SIMATIC STEP 7 PROFESSIONAL.
SIMATIC STEP 7 V5.5 SP3	1	6ES7810-4CC10-0YA5	
S7 SCL V5.3 SP6	1	6ES7811-1CC05-0YA5	

Example files and projects

The following list includes all files and projects that are used in this example.

Table 2-5

Component	Note
57249109_SNMP_Traps_CODE_V30.zip	This zip file contains the STEP 7 V15.1 project.
57249109_SNMP_Traps_DOC_V30_en.pdf	This document

3 Basics of the SNMP Protocol

Definition

The SNMP – Simple Network Management Protocol – is a UDP-based protocol that was specifically specified for the administration of data networks and which is based on the communication between SNMP manager and SNMP agents: The individual nodes in the network – network components or also terminal units – feature a SNMP agent that provides information, for example, network management software, in a structured form to the SNMP manager.

This structure is referred to as MIB – Management Information Base. In the network node, the SNMP agent is usually implemented as firmware functionality.

3.1 Mode of operation

Server-Client model

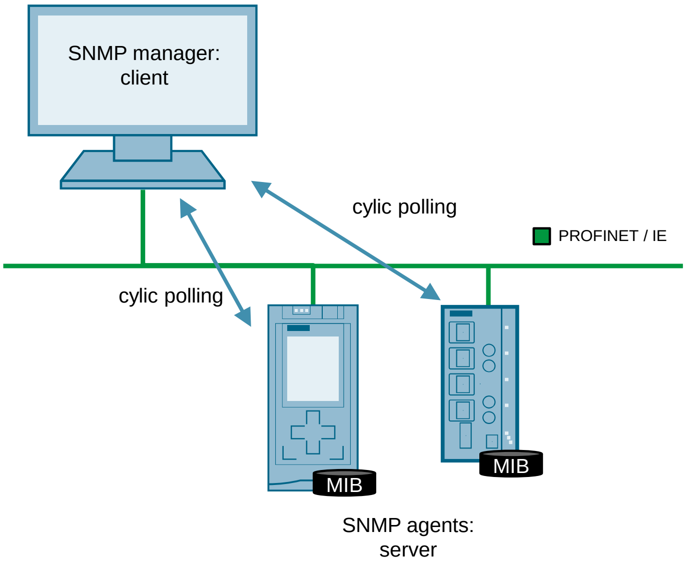
A network management solution based on SNMP operates according to the client/server model. The management station (SNMP manager, client) can request information from the agents (servers) to be controlled.

Cyclic polling

Requesting information from the SNMP agents is mainly carried out in cyclic requests. The MIB information is called from the management station and visualized, if required.

With SNMP, the nodes cannot just be monitored; instructions for controlling the devices are also possible. This includes, for example, the enabling or disabling of a port on a network component.

Figure 3-1



The communication between the agents and network management station is performed in the background and is usually only an insignificant stress on the network. However, for the configuration of the manager, it has to be observed that the poll intervals are not set too short.

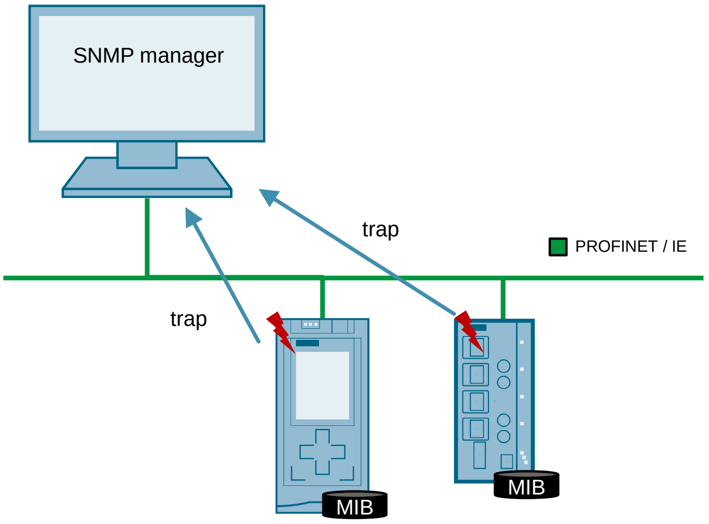
The following commands support the SNMP protocol for cyclic requests:

- GET REQUEST (requesting a data record)
- GET NEXT (requesting the following data record)
- SET REQUEST (changing a data record)
- GET RESPONSE (response to a GET request)

Acyclic event (trap)

The nodes of a network can also send states unprompted via what is known as traps, to the network management system. This has the advantage that changes in the status of the network arrive quickly at the management station and not only with the next polling cycle. More detailed information on the SNMP traps can be found in chapter 3.3.

Figure 3-2



SNMP agents:
event-controlled traps

The following message type is used by the SNMP protocol for acyclic information transfer:

- EVENT/TRAP (unprompted message of the agent to the manager)

3.2 Data storage in the agent

MIB – Management Information Base

An MIB describes the entirety of all SNMP objects (SNMP variables) located in the network. The variables are described in a language that is independent from the target system (ASN.1).

Figure 3-3

Standardized data

System information such as network statistic, counter, tables

Expanded standardized data

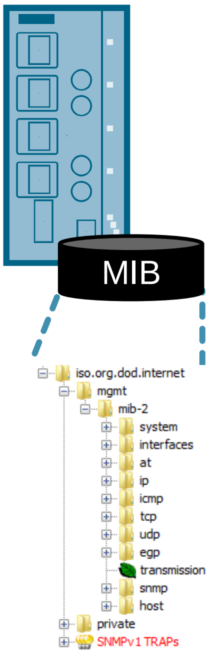
For example, data on network load (TMON) when switching

Device-specific data

For example, status of redundant voltage supply

Bridge MIB

For example, topological view by means of an "office tool"



In RFC 1155, a global MIB is defined whose variables are in parts supported by all SNMP agents.

If component-specific, non-standardized data is necessary for network monitoring, this data can be described by the manufacturers in "private MIBs".

On the support pages, Siemens AG provides private MIBs for many network components, for example, together with the firmware update.

Due to the cross-vendor standardization of MIBs and access mechanisms, even a heterogeneous network with components from different manufacturers can be monitored and controlled.

OID – Object identifier

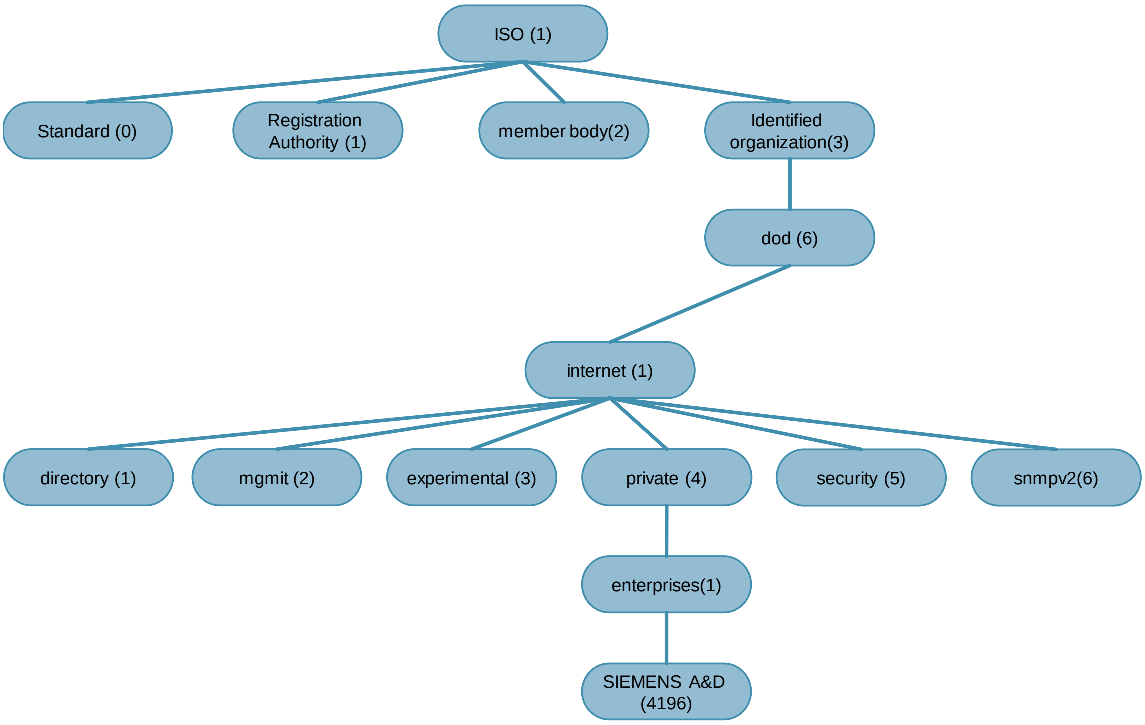
The OID (object identifier) uniquely describes the address of a MIB object. For standardized MIB objects, the address is predefined. Private MIB objects are always stored in the "enterprise" directory.

The addresses within this structure are set by the manufacturer. Only the manufacturer number must be registered.

The OIDs can be displayed as number chain or as ASCII string (see Figure 3-4).

The following graphic shows an excerpt from the MIB structure. The OID for the "SIEMENS A&D" node accordingly is: 1.3.6.1.4.1.4196.

Figure 3-4



3.3 SNMP traps

Description

The network management system has to execute each variable and status request individually via the already discussed mechanisms (GET, SET). In order to display events of the SNMP agent instantly in the SNMP manager, so called "traps" can be send via SNMP. The SNMP agents do not receive a receipt for sent traps.

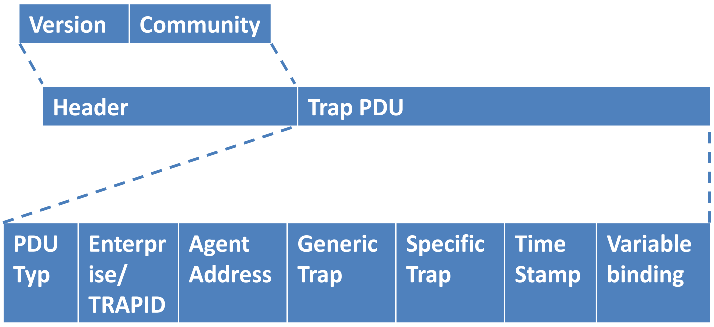
The advantage of sending a trap can be seen by the fact that the network management system displays special events in the network instantly, without the wait until the next routine information request.

Note In order to avoid more stress of the network, the sending of traps should only be done when required.

3.3.1 Frame structure

The trap frame has the following structure:

Figure 3-5



Note In "Variable binding" several OID variables for writing or with updated values etc. can be sent with the trap. The example project sends an additional string variable each to describe the trap.

The following table shows the detailed structure of a trap frame with the following information:

- SNMPv1
- Community: ‘public’
- (fictitious)TRAP ID: ‘1.2.7865’
- IP address agent: 172.16.46.13
- specific ID: 67
- Variable OID: ‘0.0’
- Variable string: ‘error’

Table 3-1 SNMP trap frame

Byte	Hex code	Description
1.	16#30	SNMP sequence
2.	16#2F	Length sequence (16#2f = 47 byte)
3.	16#02	SNMP version data type (02 – integer, 04 – byte, 05 – zero)
4.	16#01	SNMP version length
5.	16#00	SNMP version (00 - V1)
6.	16#04	Community string data type
7.	16#06	Community string length
8.	16#70	'p’
9.	16#75	'u'
10.	16#62	'b'
11.	16#6C	'l'
12.	16#69	'i'
13.	16#63	'c'
14.	16#A4	Trap PDU (A4 – trap)
15.	16#22	Length trap PDU
16.	16#06	Trap ID data type (06 – object)
17.	16#03	Trap ID length
18.	16#2A	1.2
19.	16#BD	1. Byte of 7865 (BER coding)
20.	16#39	2. Byte of 7865 (BER coding)
21.	16#40	IP data type
22.	16#04	IP length
23.	16#AC	172
24.	16#10	16
25.	16#2E	46
26.	16#0D	13
27.	16#02	Generic trap ID data type
28.	16#01	Generic trap length
29.	16#06	06 (company-specific)
30.	16#02	Specific trap data type
31.	16#01	Specific trap length
32.	16#43	67

Byte	Hex code	Description
33.	16#43	Data type of time stamp
34.	16#01	Length of time stamp
35.	16#0B	11
36.	16#30	Variable list type (30 – sequence)
37.	16#0C	List of variable length
38.	16#30	Variable type
39.	16#0A	Variable length
40.	16#06	OID variable data type (06 – object)
41.	16#01	OID variable length
42.	16#00	0.0
43.	16#04	Variable value data type (04 – byte)
44.	16#05	Variable length
45.	16#65	'e'
46.	16#72	'r'
47.	16#72	'r'
48.	16#6F	'o'
49.	16#72	'r'

3.3.2 Identification

The complete identification of a trap is realized via three fields in the SNMP frame:

- TRAP ID (syntax and structure identical to OID)
- generic trap (precise description of the TRAP ID)
- specific trap (company-specific ID)

The information mentioned is stored in the trap PDU of the frame.

Generic trap

For the value of the "generic trap" field, the following values are defined:

Table 3-2

Designation	Value
coldStart	0
warmStart	1
linkDown	2
linkUp	3
authenticationFailure	4
egpNeighborLoss	5
enterpriseSpecific	6

Specific trap and TRAP ID

If coding "6" (company-specific) is used as "generic trap", this company-specific trap can be specified in more detail by other ID details in the following field (specific TRAP).

Note

The user can influence the following information in the application example:

- TRAP ID
- specific TRAP (company-specific ID)

As specific TRAP, the following assignment was made in the example program:

- Incoming interrupt: specific TRAP = 67.
- Outgoing interrupt specific TRAP = 68.

4 Mode of operation

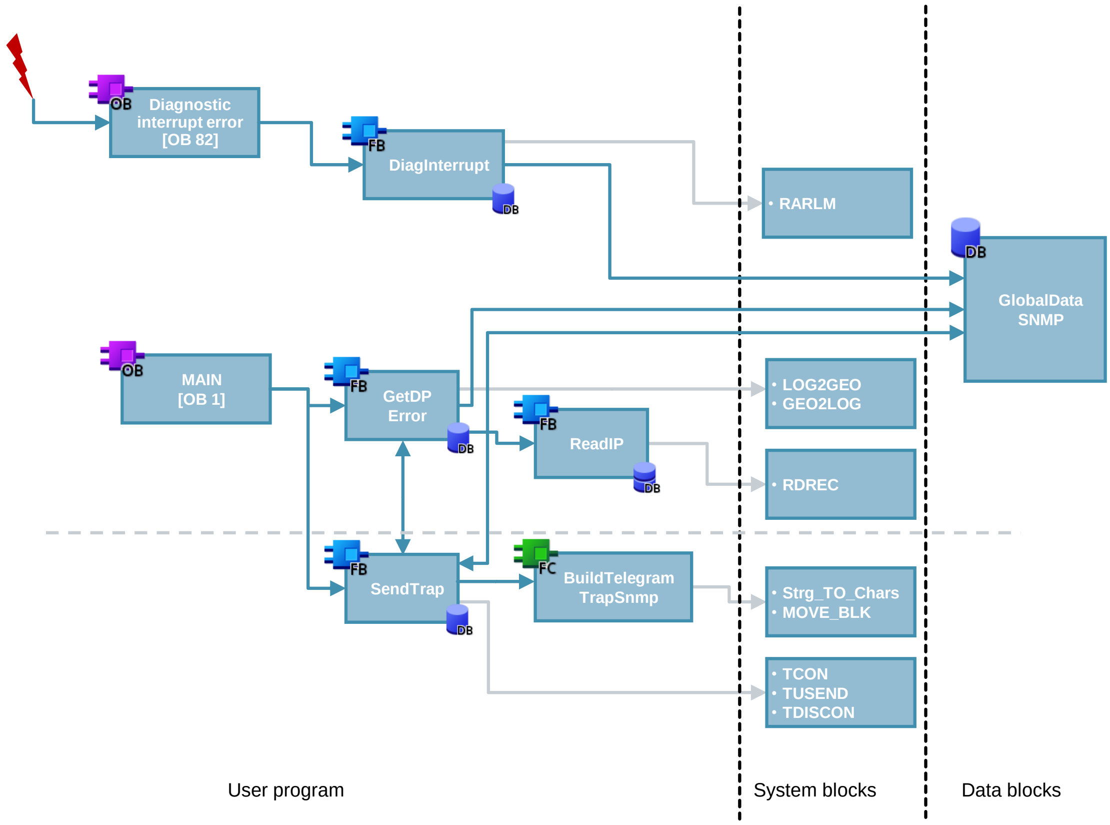
This chapter describes the mode of operation of the user program for the S7 1500 PLC in STEP 7 V15.1. Differences to S7-300 are listed in chapter 4.5.

The user program provided for STEP 7 V5.5 for the S7-300 is not described separately.

4.1 General overview

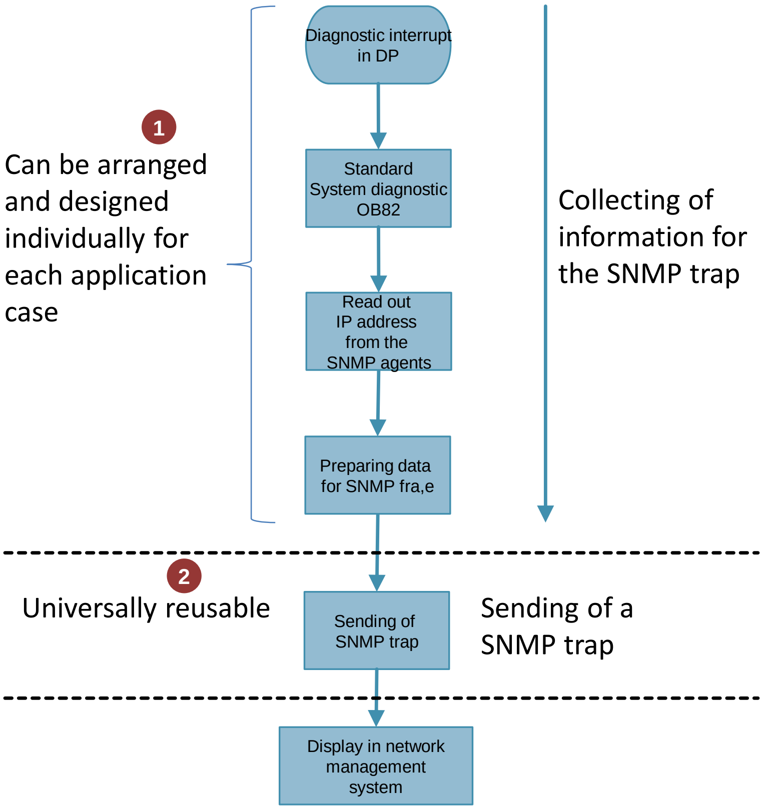
Program overview

Figure 4-1



Workflow

Figure 2-3


1
Can be arranged and designed individually for each application case
Diagnostic interrupt in DP
Standard System diagnostic OB82
Read out IP address from the SNMP agents
Preparing data for SNMP fra,e
Collecting of information for the SNMP trap
2
Universally reusable
Sending of SNMP trap
Sending of a SNMP trap
Display in network management system

Description

The program is divided into two parts:

Table 4-1

	Function	Blocks
1.	Detection of a diagnostic interrupt. Preparing the diagnostic information. Requirement: Diagnostic interrupts have to be configured in the distributed I/O.	• Diagnostic error interrupt (OB82) - DiagInterrupt (FB4) • Main (OB1) - GetDPError (FB2) - ReadIP (FB10)
2.	Sending of a SNMP trap.	• Main (OB1) - SendTrap (FB1) - BuildTelegramTrapSnmp (FC1)

You can use the "SendTRAP" (FB1) function block with the "BuildTelegramTrapSnmp" (FC1) function, independent from the other program parts in order to send user-defined SNMP traps.

Note

The application example sends different TRAP IDs for incoming and outgoing (for fault-free module) diagnostic interrupts.

This facilitates the differentiation of the traps in the network management system.

4.2 User-defined data types

Overview

In the user program the following PLC data types (UDTs) are used.

Table 4-2

S7-1500	S7-300	Usage
typeDPRalam	typeDPRalam	Includes a data record that is saved after a diagnostic interrupt for further processing.
typeDPRalamArray	typeDPRalamArray	Array of the typeDPRAlam data type.
typeIPaddress	typeIPaddress	Structure for storing an IP address.
typeTrapInformation	typeTrapInformation	Includes information for generating a SNMP trap frame.
typeTrapTelegram	typeTrapTelegram	Byte array to store the SNMP traps to be sent (part of typeTrapInformation).
provided by the system	GEOADDR	Structure for data for the use with the functions LOG2GEO/LOG_GEO and GEO2LOG/GEO_LOG
provided by the system	typeTaddrParam	Structure for the use with the FB TUSEND.
provided by the system	typeTconParam	Structure for the use with the FB TCON.
typeAinfo	-	To store the RALRM data.

4.2.1 PLC data type typeDPRalam

Usage

The typeDPRalam data type is used to store temporary data after the receipt of a diagnostic interrupt from OB82 in the example application. The evaluation of this data is via FB GetDPError.

Configuration

The following table illustrates the structure of the DP_RALRM data type.

Table 4-3

S7-1500	S7-300	Data type	Usage
newDP	newDP	Bool	TRUE, if new diagnostic information is available.
tInfo	-	TI_DiagnosticInterrupt	Information of the RALRM function.
aInfo	-	AINFO	Information of the RALRM function.
-	lowVlt	Bool	Information on the diagnostic event.
-	cnfgError	Bool	Information on the diagnostic event.
-	modStp	Bool	Information on the diagnostic event.
-	common	Bool	Information on the diagnostic event.
gone	gone	Bool	Shows the outgoing interrupt.
	laddr	Word	LADDR of the module with diagnostic event.

4.2.2 PLC data type typeTrapInformation

Usage

The typeTrapInformation PLC data type is used to record relevant information for a SNMP trap. It is interconnected at the function SendTrap and GetDPError.

Configuration

The data type is divided into two areas:

1. relevant information for the trap.
2. SNMP trap frame which can then be sent.

The following table shows the detailed configuration of the typeTrapInformation data type.

Table 4-4

S7-1500/S7-300	Data type	Usage
ipAgent	typeIPAddress	Includes the address of the agent from which the TRAP is triggered. Can be assigned with any IP address and is not necessarily the IP address of the PLC sending the trap.
trapID	String[125]	The TrapID of the trap (Syntax:'1.2.3').
specificID	Int	The company-specific ID of the trap.
oidBindingVar	String[125]	The OID of the connected variable (syntax: '1.2.3').
description	String[125]	The text of the connected variable.
telegram	Array of Byte	An array to record the SNMP trap frame.

4.3 Recording and preparing a diagnostic interrupt

The recording, evaluating and preparing of the diagnostic interrupt with the help of the SIMATIC diagnostic is divided in two parts:

1. Receipt of a diagnostic interrupt with the call of OB82 and read out of the diagnostic information provided by the system with the RALRM system function (integrated in FB DiagInterrupt) (see chapter 4.3.1).
2. Processing of the diagnostic information in a cyclic OB and preparation with the FB GetDPError (FB 2) for the FB SendTrap (FB1) to be able to send SNMP traps (see chapter 4.3.2).

Chapter 4.4 describes the sending of the data as SNMP trap.

4.3.1 Diagnostic Interrupt Error (OB82) and DiagInterrupt (FB4)

General

The diagnostic interrupt error (OB82) is called by the system (the S7 PLC) in the event of a diagnostic interrupt. The call occurs for incoming and outgoing interrupts.

Since OB82 is only called once after a diagnostic interrupt, the read out of diagnostic information provided by the system is not possible here. This task is performed by FB DiagInterrupt. This block is only enabled in OB82; further processing is done in the cyclic program flow.

Program workflow

Figure 4-4 PAP OB82

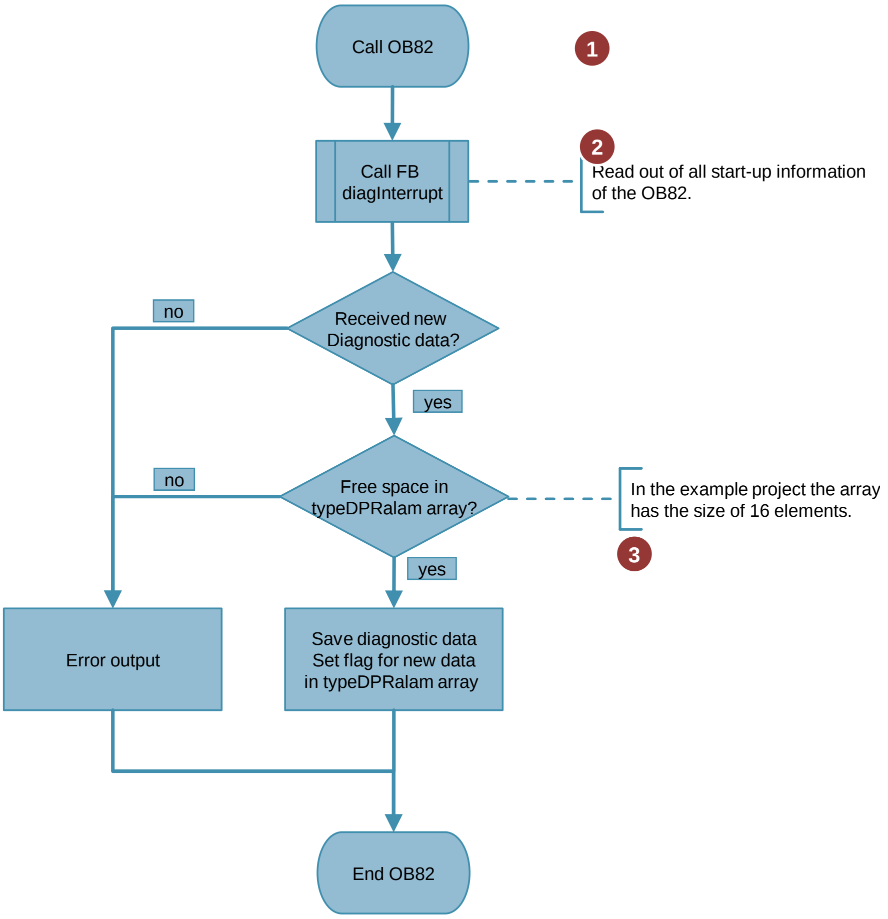

Call OB82
1
Call FB diagInterrupt
2
Read out of all start-up information of the OB82.
no
Received new Diagnostic data?
yes
no
Free space in typeDPRalam array?
In the example project the array has the size of 16 elements.
3
yes
Error output
Save diagnostic data Set flag for new data in typeDPRalam array
End OB82

Table 4-4

No.	Remark
1.	After a diagnostic interrupt has been triggered, the OB 82 is called automatically in S7 PLC and the FB DiagInterrupt is enabled. **Note** Diagnostic interrupts have to be enabled in the hardware configuration of the respective I/O devices.
2.	Based on the start information of the organization block and with the help of the RALRM system block, the FB DiagInterrupt reads out all available information.
3.	The information is temporarily stored in the GlobalDataSNMP global data block in an array. A flag signals that new diagnostic data is available.

4.3.2 GetDPError (FB2) and ReadIP (FB10)

General

The FB GetDPError (FB2) internally calls the FB ReadIP (FB10). The FB GetDPError (FB2)

- filters the collected data at the diagnostic interrupt.
- extracts the IP address of the triggering device.
- reads out and analyzes the data for sending a trap and stores it in the typeTrapInformation structure.

Block interface

Figure 4-5

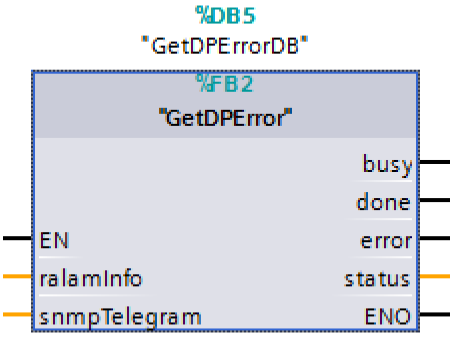


Table 4-5

Parameters	Data type	Description
ralamInfo	IN/OUT: typeDPRalam	Array from typeDPRalam: Signals a new diagnostic interrupt and contains its information (see chapter 4.2.1).
snmpTelegram	IN/OUT: typeTrapInformation	Structure for all relevant data for sending a SNMP trap (see chapter 4.2.2).
busy	OUT: Bool	TRUE, when the block currently processes diagnostic data.
done	OUT: Bool	TRUE for one cycle when the block has prepared diagnostic data for a SNMP trap.
error	OUT: Bool	TRUE, if an error occurred during processing.
status	OUT: DWORD	When ERROR=TRUE, a more precise error specification is pending on the STATUS output. See also Chapter 4.4.3.

Program workflow

Figure 4-6

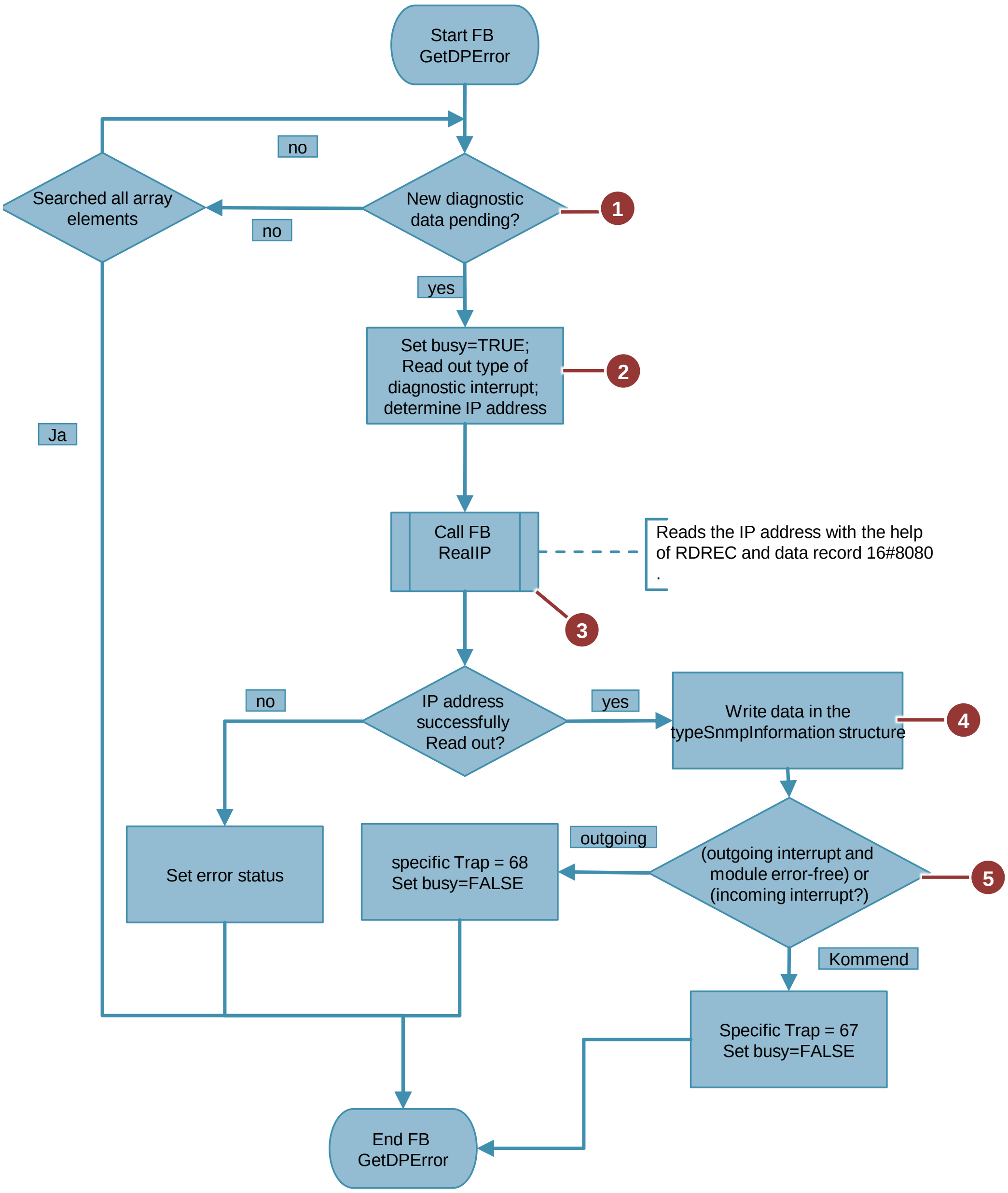

Start FB
GetDPError
no
Searched all array elements
New diagnostic data pending?
1
no
yes
Set busy=TRUE;
Read out type of diagnostic interrupt;
determine IP address
2
Ja
Call FB
RealIP
Reads the IP address with the help of RDREC and data record 16#8080
.
3
no
IP address successfully Read out?
yes
Write data in the typeSnmpInformation structure
4
Set error status
specific Trap = 68
Set busy=FALSE
outgoing
(outgoing interrupt and module error-free) or (incoming interrupt?)
5
Kommend
Specific Trap = 67
Set busy=FALSE
End FB
GetDPError

Table 4-6

No.	Action
1.	With the information from the diagnostic interrupt it is checked in the typeDPRalam array whether new diagnostic data (newDP = TRUE) is pending on an element.
2.	If a new diagnostic interrupt is pending, the precise error is detected. If the type of diagnostic error is not detected the description is set to ‘common’.
3.	The IP address of the device that caused the interrupt is read out.
4.	If the read out was successful, the information (IP address, TRAP ID, generic TRAP, OID, variable description) for sending a trap is stored in the typeTrapInformation structure.

No.	Action
5.	If it is • an incoming interrupt, the specific Trap ID is set to 67. • an outgoing interrupt and as a result a fault-free module, the specific Trap ID is set to 68. In the network management system this step is for the differentiation of the existing diagnostic interrupts.

4.4 Sending of a SNMP trap

4.4.1 Block description of FB SendTrap (FB1)

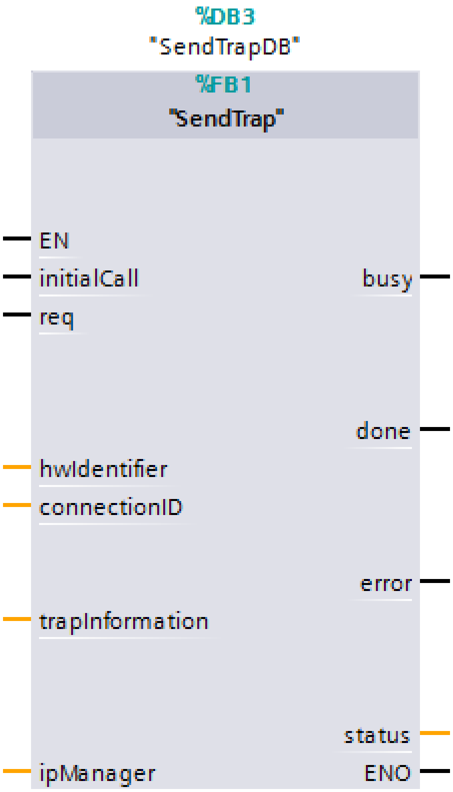
Description

The FB SendTrap (FB1) function block realizes the following functions:

1. Establishing a UDP connection to the SNMP manager.
2. Processing of the input information in the typeTrapInformation structure via the BuildTelegramTrapSnmp function to a SNMP trap frame.
3. Sending of the SNMP trap frame.
4. In the event of a fault: Disconnecting and establishing the UDP connection.

Call

Figure 4-7



Parameters

The block has the following inputs and outputs:

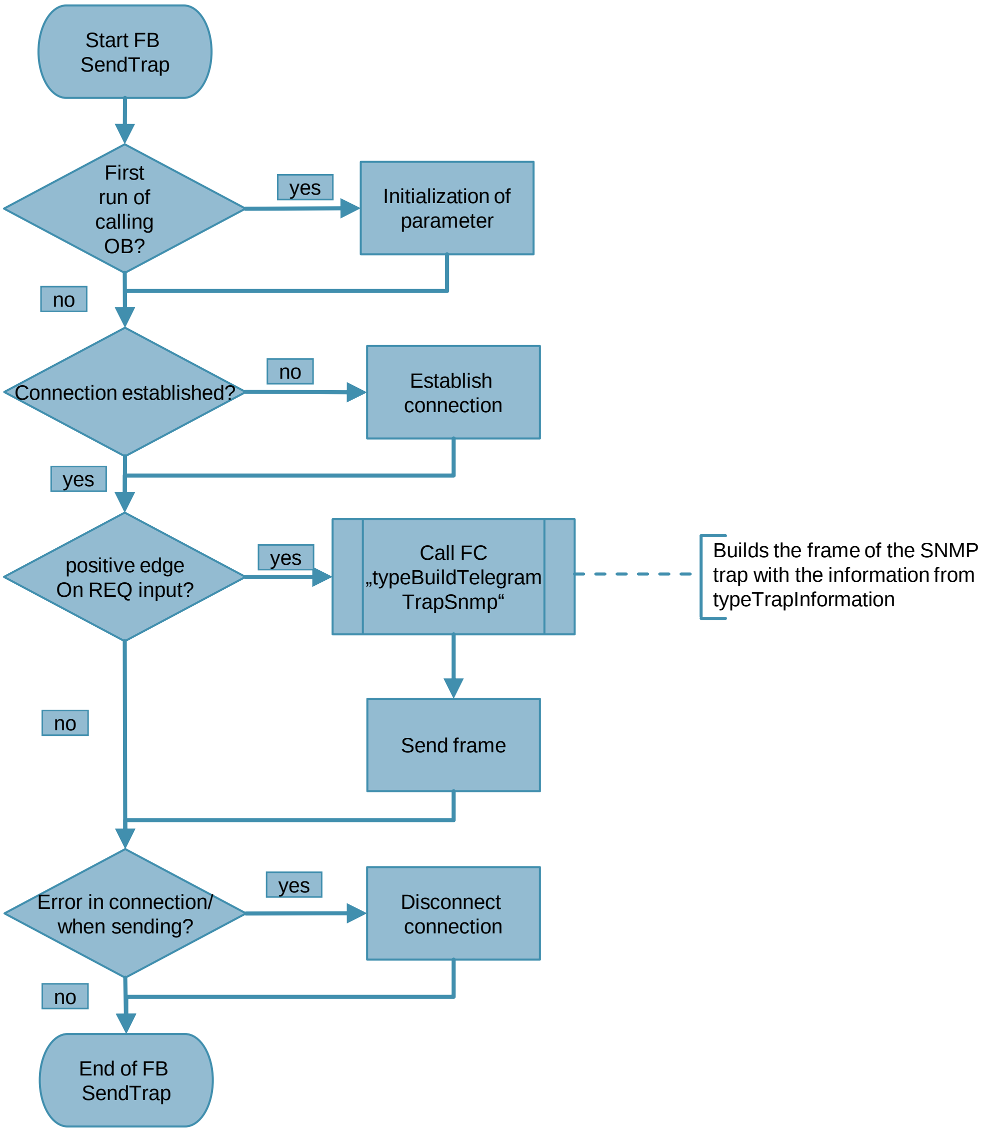
Table 4-7

Parameters	Data type	Description
initialCall	IN: Bool	Shows the first call of the FB. In the first run, the parameters are initialized.
req	IN: Bool	A positive edge triggers the sending of a SNMP trap to the address of the SNMP manager. Requirement: The block is not in status • BUSY = 1 or • ERROR = 1
hwidentifier	IN: HW_ANY	The hardware ID of the PROFINET interface of the PLC. You can use both interfaces of a PLC for the application. **Note** For a S7-300 this is the diagnostic address of the interface.
connectionId	IN: Bool	The connection ID for establishing the connection. **Note** The connection ID has to be unique across the project.
trapInformation	IN: typeTrapInformation (UDT)	Includes all information for the SNMP frame packet. See also chapter 4.2.2.
ipManager	OUT: typeIPAddress (UDT)	Specifies the IP address of the SNMP manager (for example, network management system).
busy	OUT: Bool	TRUE, if the block is currently busy. Is the case when • the parameters are initialized. • a connection with TCON is established. • a SNMP TRAP is sent via TUSEND. • the connection with TDISCON is disconnected.
done	OUT: Bool	TRUE for one cycle, when a SNMP Trap was sent.
error	OUT: Bool	TRUE, if an error is pending on the block.
status	OUT: DWORD	outputs an error code for =TRUE, see chapter 4.4.3.

4.4.2 Program workflow

The following plan shows the structure for the sending of a SNMP trap:

Figure 4-8


Start FB SendTrap
First run of calling OB?
yes
Initialization of parameter
no
Connection established?
no
Establish connection
yes
positive edge On REQ input?
yes
Call FC „typeBuildTelegramTrapSnmp“
Builds the frame of the SNMP trap with the information from typeTrapInformation
no
Send frame
Error in connection/ when sending?
yes
Disconnect connection
no
End of FB SendTrap

Note The T blocks (TCON, TUSEND, TDISCON) work asynchronously. Accordingly, the program flow chart should be viewed as schematic.

4.4.3 Error and status messages

Explanation

The used function blocks GetDPError and SendTrap use the STATUS output parameter. OB82 saves a status in the DB GlobalDataSnmp (DB2) in the case of an error.

The following table shows what values the STATUS output can take on and their meaning.

Values of the STATUS output

The values of the table are hexadecimal values. The pending ‘xxxx’ variables are each values returned by the functions (see table).

Example

Error code: 16#0180_9000

The call of the RALRM function was ended with error code 16#xx80_9000 → the logic start address on the block input is invalid.

Table 4-8

Error code	Description
01xx_xxxx	Error in FB DiagInterrupt: Call of the RALRM function completed with error.
0200_8123	The array for storing the DP_RALRM error is already full.
0300_xxxx	incorrect LADDR onLOG2GEO function.
0400_xxxx	incorrect specification on GEO2LOG.
05xx_xxxx	Error on RDREC when reading the IP address.
0600_xxxx	Error on RDSYSST when reading the local_device_id (only S7-300).
0700_xxxx	Error on TCON.
0800_xxxx	Error on TUSEND.
0900_xxxx	Error on TDISCON.
0A00_8124	Other positive edge at REQ during send job. → one job was lost.
0B00_xxxx	Incorrect specification on GEO2LOG.
0C00_xxxx	Error on ModuleStates.

4.5 Differences in configuration between S7-1500 PLC and S7-300 PLC

4.5.1 User program

Due to the different architecture, the function blocks between S7-1500 and S7-300 are different. Accordingly, attention has to be paid, to always use suitable blocks for the respective PLC from the projects/library.

Evaluation of diagnostic interrupt

S7-1500

Via the FB DiagnoseInterrupt function, information from the diagnostic interrupt is read out with the RALRM function.

S7-300

Just from the startup data of the OB82, basic information on sending the trap is already being read out. The FB DiagnoseInterrupt is not used.

Diagnostic interrupt

In the example project the following errors are detected:

Table 4-9

No.	S7-1500	S7-300
1.	Wire break	Low external voltage
2.	Low external voltage	Configuration error
3.	Short circuit	Module stop
4.	Group error (general)	Group error (general)

4.5.2 Open User Communication/local_device_id

The differences in the communication via the Open User Communication with the blocks TCON, TUSEND, TDISCON between a S7-1500 PLC and a S7-300 PLC can be found in the TIA Portal online help.

The user program realizes the different use of the communication blocks (T blocks).

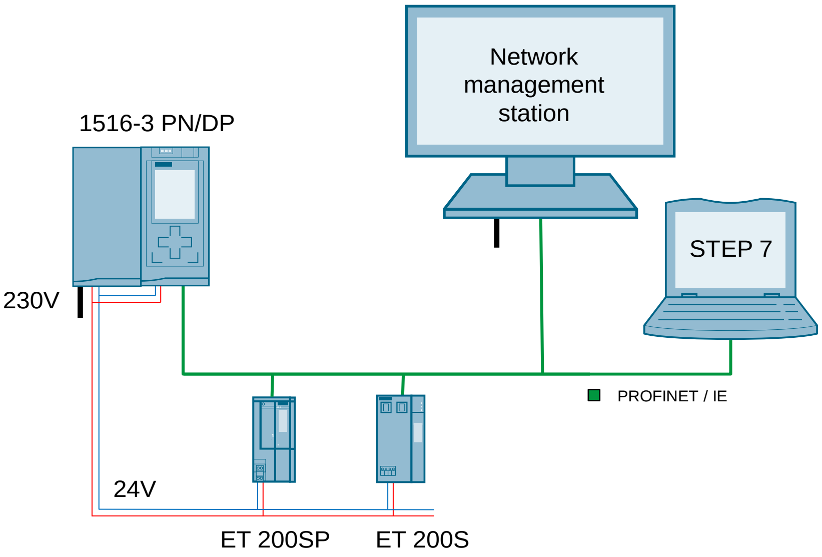
5 Installation and Commissioning

This chapter describes the configuration and the commissioning of the application. The commissioning is shown with a PLC 1516-3 PN/DP, the steps are the same for other PLCs.

5.1 Installing the hardware

The figure below shows the hardware configuration of the application.

Figure 5-1



Note The installation guidelines for SIMATIC systems must always be observed.

Table 5-1

No.	Action	Remark
1.	Fix the S7-PLC including power supply as well as the two distributed I/O stations on a suitable rack.	The rail with article number 6ES7590-1AE80-0AA0 is suitable for mounting.
2.	Connect the network management station with an IEC socket to the power supply.	
3.	Connect the individual components either in daisy chain or via a switch.	The distributed I/O stations ET 200S and ET 200SP have two ports on the interface. The S7-1500 PLC also has two ports on interface 1.
4.	Connect the individual components with the 24 volt connection of the power supply (see Figure 5-1).	
5.	Connect your engineering station with the Ethernet network either via a switch or also in daisy chain.	
6.	Connect the engineering station to the power supply.	

5.2 Installing the software

Install STEP 7 V15.1; see, for example, manual \3\.

Table 5-2

No.	Action
1.	Download the 57249109_SNMP_Traps_CODE_V30.zip file from the HTML page onto your engineering station and unzip the folder.
2.	Double-click on the file icon in the unzipped program folder. The project opens in TIA V15.1.

5.3 Commissioning

IP addresses used

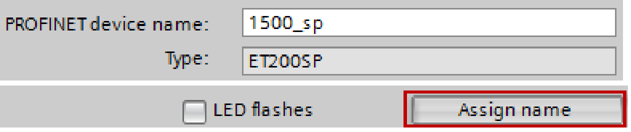
The following IP addresses are used as default values in the STEP 7 V15.1 project:

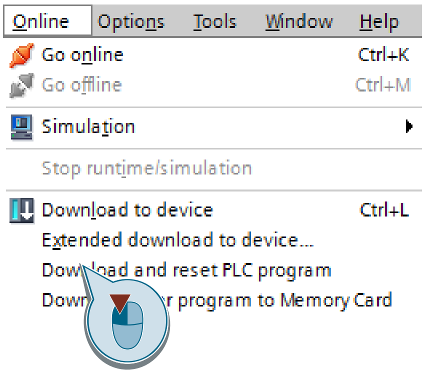
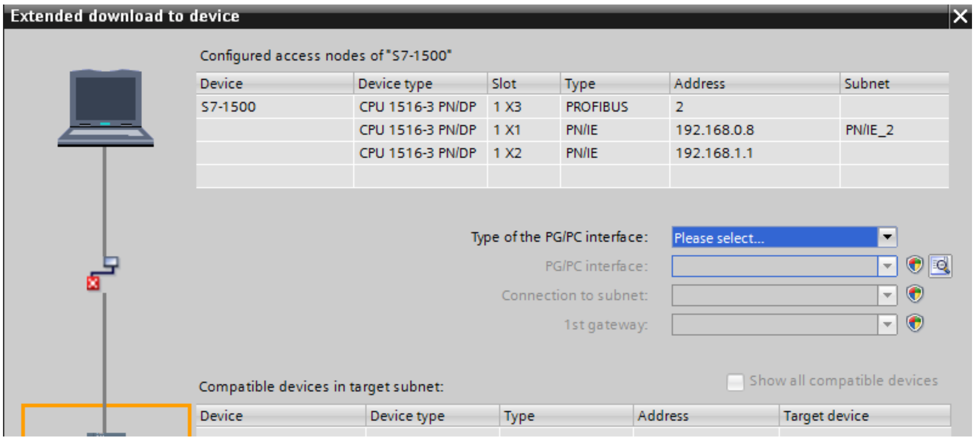
Table 5-3

Device	IP address
PLC 1516-3 PN/DP	172.16.46.13
PLC 315-2 PN/DP	172.16.46.14
ET 200S	172.16.46.3
ET 200SP	172.16.46.2

Commissioning

Table 5-4

No.	Action
1.	Search for your PNIO controller in "Online access > [YOUR_NETWORK_ADAPTER] > Update accessible devices". Go to "Online & diagnostics" and in the work area to "Functions > Assign IP address". Assign the PNIO controller • either (PLC 1516-3 PN/DP) the IP address 172.16.46.13, subnet mask: 255.255.0.0 • or (PLC 315-2 PN/DP) the IP address 172.16.46.14, subnet mask: 255.255.0.0 via the "Assign IP address" switch.
2.	Search for the ET 200SP and the ET 200S station in "Online access > [YOUR_NETWORK_ADAPTER] > Update accessible devices". The following points describe the procedure for the ET 200SP on the PLC 1516-3 PN/DP. The procedure for the ET 200S and furthermore the procedure for the PLC 315-2 PN/DP is analogous.
3.	Go to "Online & diagnostics" and in the work area to "Functions > Assign name". Assign the name "1500_sp" to the ET 200SP.  PROFINET device name: 1500_sp Type: ET200SP

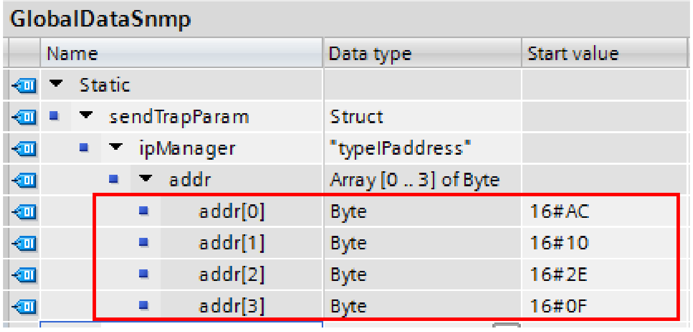
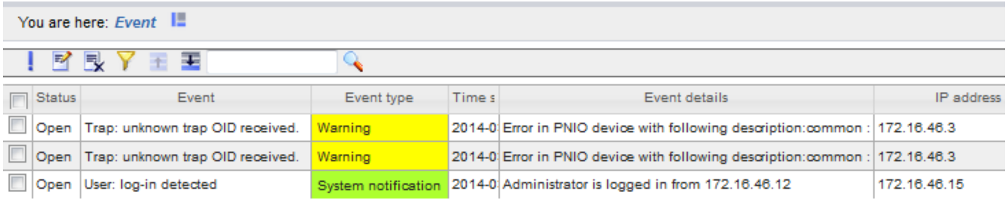
<table>
<tr><th>No.</th><th>Action</th></tr>
<tr><td>4.</td><td>Select the PLC you would like to download in the project tree and select "Online > Download and reset PLC program".

 </td></tr>
<tr><td>5.</td><td>If necessary, select the PC interface used by you. Select the PLC found by STEP 7 and click "Load".
You have to start the search for nodes manually via "start search" (only for TIA V15.1).
<b>Note</b>
If the IP address of your PC is not in the subnet of the node, STEP 7 automatically assigns a suitable IP address.

Extended download to device
Configured access nodes of "S7-1500"
<table>
<tr><th>Device</th><th>Device type</th><th>Slot</th><th>Type</th><th>Address</th><th>Subnet</th></tr>
<tr><td>S7-1500</td><td>CPU 1516-3 PN/DP</td><td>1 X3</td><td>PROFIBUS</td><td>2</td><td></td></tr>
<tr><td></td><td>CPU 1516-3 PN/DP</td><td>1 X1</td><td>PN/IE</td><td>192.168.0.8</td><td>PN/IE_2</td></tr>
<tr><td></td><td>CPU 1516-3 PN/DP</td><td>1 X2</td><td>PN/IE</td><td>192.168.1.1</td><td></td></tr>
</table>
Type of the PG/PC interface: Please select...
PG/PC interface:
Connection to subnet:
1st gateway:
Compatible devices in target subnet:
□ Show all compatible devices
<table>
<tr><th>Device</th><th>Device type</th><th>Type</th><th>Address</th><th>Target device</th></tr>
</table></td></tr>
<tr><td>6.</td><td>Load the program into your PLC via the menus that appear.</td></tr>
<tr><td>7.</td><td>Start the S7 PLC.
When the configuration is correct, the LEDs of the distributed I/O systems will light up green and the S7 PLC is in RUN mode.</td></tr>
</table>

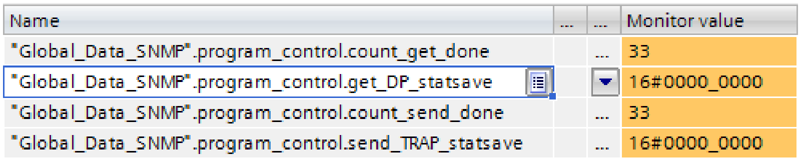
6 Operating the Application

6.1 Scenario A: Sending of a trap, triggered by a distributed I/O system

To send a trap that was triggered by a distributed I/O system from the PLC to a network management system, proceed as described in Table 6-1. The procedure for a S7-1500 PLC is identical for a S7-300 PLC.

Table 6-1

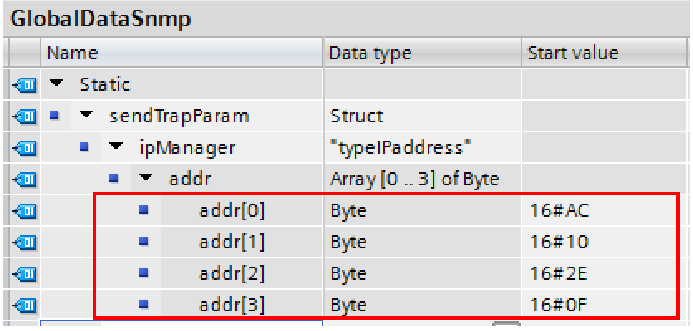
<table>
<tr><th>No.</th><th>Action</th></tr>
<tr><td>1.</td><td>Install and load the project as described in chapter 5.</td></tr>
<tr><td>2.</td><td>Open the DB GlobalDataSnmp and enter the IP address of the network management station as start value in the SendTrapParam.ipManager.addr parameter.

GlobalDataSnmp
<table>
<tr><th>Name</th><th>Data type</th><th>Start value</th></tr>
<tr><td>▼ Static</td><td></td><td></td></tr>
<tr><td>▼ sendTrapParam</td><td>Struct</td><td></td></tr>
<tr><td>▼ ipManager</td><td>"typeIPaddress"</td><td></td></tr>
<tr><td>▼ addr</td><td>Array [0 .. 3] of Byte</td><td></td></tr>
<tr><td>addr[0]</td><td>Byte</td><td>16#AC</td></tr>
<tr><td>addr[1]</td><td>Byte</td><td>16#10</td></tr>
<tr><td>addr[2]</td><td>Byte</td><td>16#2E</td></tr>
<tr><td>addr[3]</td><td>Byte</td><td>16#0F</td></tr>
</table>
<b>Note</b>
The IP has to be coded as hexadecimal value, for example, 16#AC = 172</td></tr>
<tr><td>3.</td><td>Reload the project into the PLC and restart the PLC.</td></tr>
<tr><td>4.</td><td>Create a diagnostic interrupt on a distributed I/O station. For example, by removing the supply voltage on the PM-E module of the ET 200S.</td></tr>
<tr><td>5.</td><td>The successful sending of the trap can be checked in your network management station (here on the example of SINEMA Server V12) or by a trace recording (for example via Wireshark).

You are here: <i>Event</i>
<table>
<tr><th>□</th><th>Status</th><th>Event</th><th>Event type</th><th>Time s</th><th>Event details</th><th>IP address</th></tr>
<tr><td>□</td><td>Open</td><td>Trap: unknown trap OID received.</td><td>Warning</td><td>2014-0</td><td>Error in PNIO device with following description:common :</td><td>172.16.46.3</td></tr>
<tr><td>□</td><td>Open</td><td>Trap: unknown trap OID received.</td><td>Warning</td><td>2014-0</td><td>Error in PNIO device with following description:common :</td><td>172.16.46.3</td></tr>
<tr><td>□</td><td>Open</td><td>User: log-in detected</td><td>System notification</td><td>2014-0</td><td>Administrator is logged in from 172.16.46.12</td><td>172.16.46.15</td></tr>
</table>
The columns "Status", "Event", "Event type" and "Time Stamp" are filled out in the SINEMA server, depending on the configuration of the SINEMA server and independent from the received trap frames.
If the trap ID is unknown, the "Event details" column includes the "Description" of the attached variables transferred with the frame, as well as the TRAP ID (cannot be seen in the screenshot).
The IP address is also taken from the trap frame.</td></tr>
</table>

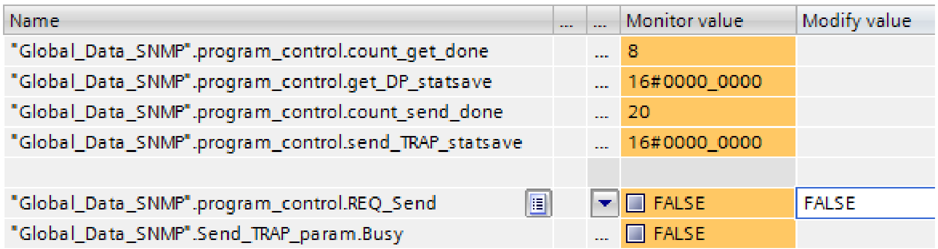
<table>
<tr><th>No.</th><th>Action</th></tr>
<tr><td>6.</td><td>The PLC now sends an automated trap to the network management station. If an error occurs, you can view the error code in the SNMP_Trap watch table.

<table>
<tr><th>Name</th><th>...</th><th>...</th><th>Monitor value</th></tr>
<tr><td>"Global_Data_SNMP".program_control.count_get_done</td><td></td><td>...</td><td>33</td></tr>
<tr><td>"Global_Data_SNMP".program_control.get_DP_statsave</td><td></td><td></td><td>16#0000_0000</td></tr>
<tr><td>"Global_Data_SNMP".program_control.count_send_done</td><td></td><td>...</td><td>33</td></tr>
<tr><td>"Global_Data_SNMP".program_control.send_TRAP_statsave</td><td></td><td>...</td><td>16#0000_0000</td></tr>
</table>
The screenshot shows 33 traps already sent – triggered by the distributed I/O. During this time, no errors occurred on the function blocks "GetDPError" and "SendTrap".
<b>Note</b>
In order to be able to send traps triggered by the distributed I/O, the variable has to be […].reqSend = TRUE at the time of the diagnostic interrupt in the example project.</td></tr>
</table>

6.2 Scenario B: Sending of a trap triggered via the watch table

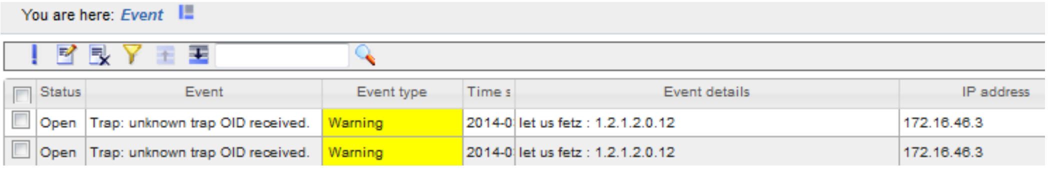
To send a manually triggered trap from the PLC to a network management station, proceed as described in Table 6-2. The procedure for a S7-1500 PLC is identical for a S7-300 PLC.

Table 6-2

<table>
<tr><th>No.</th><th>Action</th></tr>
<tr><td>1.</td><td>Install and load the project as described in chapter 5.</td></tr>
<tr><td>2.</td><td>Open the DB GlobalDataSnmp and enter the IP address of the network management station as start value in the SendTrapParam.ipManager.addr parameter.

GlobalDataSnmp
<table>
<tr><th>Name</th><th>Data type</th><th>Start value</th></tr>
<tr><td>Static</td><td></td><td></td></tr>
<tr><td>sendTrapParam</td><td>Struct</td><td></td></tr>
<tr><td>ipManager</td><td>"typeIPaddress"</td><td></td></tr>
<tr><td>addr</td><td>Array [0 .. 3] of Byte</td><td></td></tr>
<tr><td>addr[0]</td><td>Byte</td><td>16#AC</td></tr>
<tr><td>addr[1]</td><td>Byte</td><td>16#10</td></tr>
<tr><td>addr[2]</td><td>Byte</td><td>16#2E</td></tr>
<tr><td>addr[3]</td><td>Byte</td><td>16#0F</td></tr>
</table>
<b>Note</b>
The IP has to be coded as hexadecimal value, for example, 16#AC = 172</td></tr>
<tr><td>3.</td><td>Reload the project into the PLC and restart the PLC.</td></tr>
<tr><td>4.</td><td>Open the SNMP_Trap watch table and enter the values specified by you for the following variables:
<ul>
<li>Addr[0] (IP address of the device for which the trap is to be displayed.)</li>
<li>Addr[1] (IP address of the device for which the trap is to be displayed.)</li>
<li>Addr[2] (IP address of the device for which the trap is to be displayed.)</li>
<li>Addr[3] (IP address of the device for which the trap is to be displayed.)</li>
<li>TrapID</li>
<li>generic_ID</li>
<li>OID_BindingVar</li>
<li>Description</li>
</ul>
Finally, accept the values into the PLC by clicking ("Modify all selected values once and</td></tr>
</table>

<table>
<tr><th>No.</th><th>Action</th></tr>
<tr><td></td><td>now").</td></tr>
<tr><td>5.</td><td>Create a positive edge on variable [...].REQ_SEND. After successful sending, the […].countSendDone variable is incremented.

<table>
<tr><th>Name</th><th>...</th><th>...</th><th>Monitor value</th><th>Modify value</th></tr>
<tr><td>"Global_Data_SNMP".program_control.count_get_done</td><td></td><td>...</td><td>8</td><td></td></tr>
<tr><td>"Global_Data_SNMP".program_control.get_DP_statsave</td><td></td><td>...</td><td>16#0000_0000</td><td></td></tr>
<tr><td>"Global_Data_SNMP".program_control.count_send_done</td><td></td><td>...</td><td>20</td><td></td></tr>
<tr><td>"Global_Data_SNMP".program_control.send_TRAP_statsave</td><td></td><td>...</td><td>16#0000_0000</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>"Global_Data_SNMP".program_control.REQ_Send</td><td></td><td>▼</td><td>□ FALSE</td><td>FALSE</td></tr>
<tr><td>"Global_Data_SNMP".Send_TRAP_param.Busy</td><td></td><td>...</td><td>□ FALSE</td><td></td></tr>
</table>
The screenshot shows eight traps already sent – triggered by the distributed I/O. Furthermore, (20-8) 12 manually triggered traps were sent.

Note

In order to be able to send traps triggered by the distributed I/O, the variable has to be […].reqSend = TRUE at the time of the diagnostic interrupt in the example project.</td></tr>
<tr><td>6.</td><td>The successful sending of the trap can be checked in your network management station (here on the example of SINEMA Server V12) or by a trace recording.

You are here: *Event*
<table>
<tr><th>□</th><th>Status</th><th>Event</th><th>Event type</th><th>Time s</th><th>Event details</th><th>IP address</th></tr>
<tr><td>□</td><td>Open</td><td>Trap: unknown trap OID received.</td><td>Warning</td><td>2014-0</td><td>let us fetz : 1.2.1.2.0.12</td><td>172.16.46.3</td></tr>
<tr><td>□</td><td>Open</td><td>Trap: unknown trap OID received.</td><td>Warning</td><td>2014-0</td><td>let us fetz : 1.2.1.2.0.12</td><td>172.16.46.3</td></tr>
</table></td></tr>
</table>

7 Further Notes, Tips & Tricks, etc.

7.1 Adjusting the FB GetDPError block

General

If you want to adjust the TRAP ID, the error message, etc. to your requirements after the receipt of a diagnostic interrupt, this chapter will help you.

You can adjust the code of the FB GetDPError to your requirements. The following sections show how you

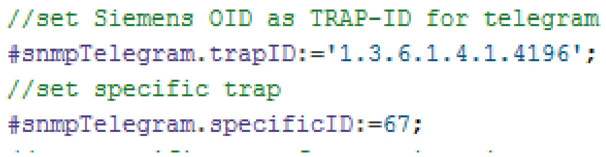
- change the TRAP ID after a distributed diagnostic interrupt.
- detect other diagnostic errors in the code.
- change the description of an error.

Changing the TRAP ID

Proceed as follows, to change the TRAP ID after a diagnostic interrupt:

Table 7-1

No.	Action
1.	Open the FB GetDPError.

2.	Move to the lines illustrated. This is where a TRAP ID is specified for all diagnostic interrupts (default: ‘1.3.6.1.4.2.4196.9’).
	
	`//set Siemens OID as TRAP-ID for telegram`
	`#snmpTelegram.trapID:='1.3.6.1.4.1.4196';`
	`//set specific trap`
	`#snmpTelegram.specificID:=67;`

3.	Change the plain text for the TRAP IDs of all interrupts according to your specifications. **Note** Of course, it may make sense to, for example, realize a differentiation of the interrupts and to assign different TRAP IDs to an IF instruction.

Detecting other diagnostic errors in the code

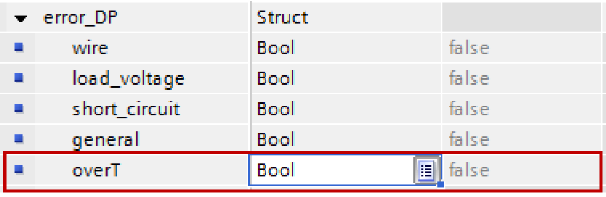
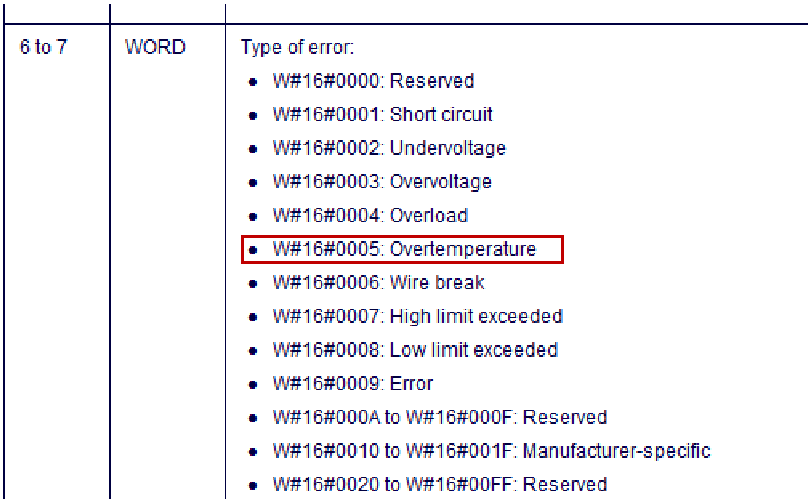
You can have other diagnostic errors detected in the code. In the example project the following errors are detected:

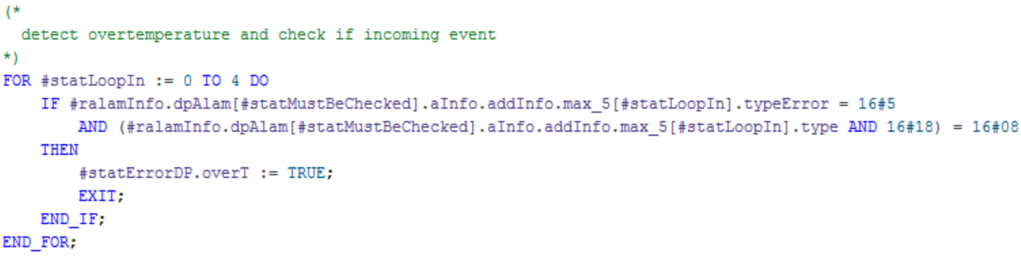
Table 7-2

No.	S7-1500	S7-300
1.	Wire break	Low external voltage
2.	Low external voltage	Configuration error
3.	Short circuit	Module stop
4.	Group error (general)	Group error (general)

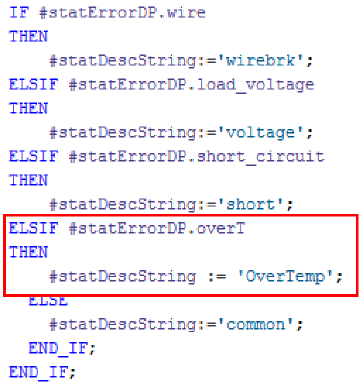
The following table shows you the procedure for the output of the "Overtemperature" diagnostic interrupt in the program of the S7-1500 as an example:

Table 7-3

No.	Action
1.	Open the FB GetDPError.
2.	Add another Boolean variable for the error to be detected (here: OverT) in the block declaration.  ▼ error_DP · Struct ■ wire · Bool · false ■ load_voltage · Bool · false ■ short_circuit · Bool · false ■ general · Bool · false ■ overT · Bool · false
3.	A description of the structure of the information AINFO can be found in the TIA Portal online help. Accordingly, request the "Overtemperature" bit in the program code. You can proceed the same way, as with the already existing request.  6 to 7 · WORD · Type of error: • W#16#0000: Reserved • W#16#0001: Short circuit • W#16#0002: Undervoltage • W#16#0003: Overvoltage • W#16#0004: Overload • W#16#0005: Overtemperature • W#16#0006: Wire break • W#16#0007: High limit exceeded • W#16#0008: Low limit exceeded • W#16#0009: Error • W#16#000A to W#16#000F: Reserved • W#16#0010 to W#16#001F: Manufacturer-specific • W#16#0020 to W#16#00FF: Reserved

4.	If the diagnostic bit is pending, set the Boolean "OverT" variable.
	
	`(*`
	`  detect overtemperature and check if incoming event`
	`*)`
	`FOR #statLoopIn := 0 TO 4 DO`
	`    IF #ralamInfo.dpAlam[#statMustBeChecked].aInfo.addInfo.max_5[#statLoopIn].typeError = 16#5`
	`        AND (#ralamInfo.dpAlam[#statMustBeChecked].aInfo.addInfo.max_5[#statLoopIn].type AND 16#18) = 16#08`
	`    THEN`
	`        #statErrorDP.overT := TRUE;`
	`        EXIT;`
	`    END_IF;`
	`END_FOR;`

<table>
<tr><th>No.</th><th>Action</th></tr>
<tr><td>5.</td><td>Add a further ELSIF request for the "overT" variable in the "set Description variable" section.



```
IF #statErrorDP.wire
THEN
    #statDescString:='wirebrk';
ELSIF #statErrorDP.load_voltage
THEN
    #statDescString:='voltage';
ELSIF #statErrorDP.short_circuit
THEN
    #statDescString:='short';
ELSIF #statErrorDP.overT
THEN
    #statDescString := 'OverTemp';
  ELSE
    #statDescString:='common';
  END_IF;
END_IF;
```
</td></tr>
<tr><td>6.</td><td>Save the project and load it into your PLC.
The "OverTemp" information is displayed as description of the trap in the event of an "Overtemperature" diagnostic interrupt.</td></tr>
</table>

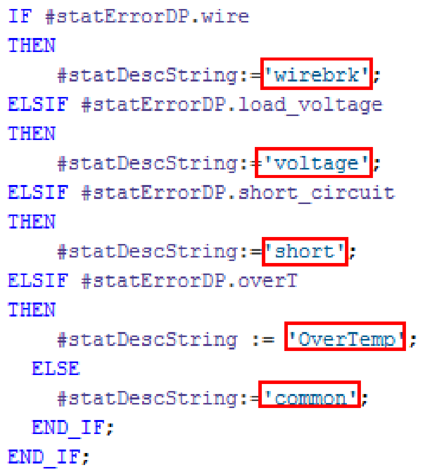
Changing the description of an error

The following table shows how the description of an error is changed in the program code.

Table 7-4

<table>
<tr><th>No.</th><th>Action</th></tr>
<tr><td>1.</td><td>Open the FB "GetDPError".</td></tr>
<tr><td>2.</td><td>

Navigate to the "set Description for variable" section. In an IF instruction the individual errors are requested here and their description is output.

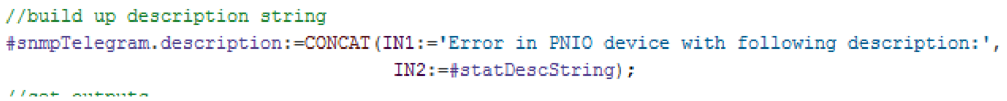


```
IF #statErrorDP.wire
THEN
    #statDescString:='wirebrk';
ELSIF #statErrorDP.load_voltage
THEN
    #statDescString:='voltage';
ELSIF #statErrorDP.short_circuit
THEN
    #statDescString:='short';
ELSIF #statErrorDP.overT
THEN
    #statDescString := 'OverTemp';
  ELSE
    #statDescString:='common';
  END_IF;
END_IF;
```

Change one of the descriptions according to your requirements.

</td></tr>
<tr><td>3.</td><td>

You can furthermore change the general description of the trap:



```
//build up description string
#snmpTelegram.description:=CONCAT(IN1:='Error in PNIO device with following description:',
                                  IN2:=#statDescString);
//get outputs
```

Note

Make sure that the two strings together are not longer than 125 characters.

</td></tr>
<tr><td>4.</td><td>Load the modified block into your PLC In the event of another diagnostic interrupt, the updated descriptions are used.</td></tr>
</table>

7.2 Calling several instances of the FBs SendTRAP in the user program

Advantage

The use of several instances of the FB SendTrap offers the following advantages:

- faster sending of the traps.
- no logic for determining several "TRAP triggers" necessary.

Boundary conditions

The use of several instances of the FBs SendTrap is possible under the following boundary conditions:

- free communication resources available (maximum number of connections depends on the PLC type).
- no use of the same memory area for the input and output variables of the FB.
- Use of different "connectionID"s on the block input.
- S7-300 PLC: firmware as of V3.2 (several connections via one port)

7.3 Displaying traps in the SINEMA Server V12

7.3.1 Sending of an unknown trap

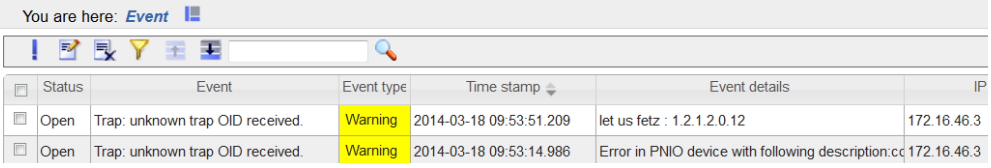
Setting up SINEMA Server V12

How to install and set up the SINEMA Server V12, can be read on the Siemens AG support pages (see chapter 8.2).

Displaying unknown trap

In the SINEMA Server V12 unknown traps are displayed directly in the log area.

Figure 7-1



You are here: *Event*

□	Status	Event	Event type	Time stamp	Event details	IP
□	Open	Trap: unknown trap OID received.	Warning	2014-03-18 09:53:51.209	let us fetz : 1.2.1.2.0.12	172.16.46.3
□	Open	Trap: unknown trap OID received.	Warning	2014-03-18 09:53:14.986	Error in PNIO device with following description:cc	172.16.46.3

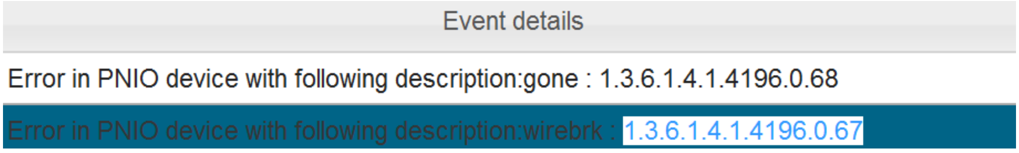
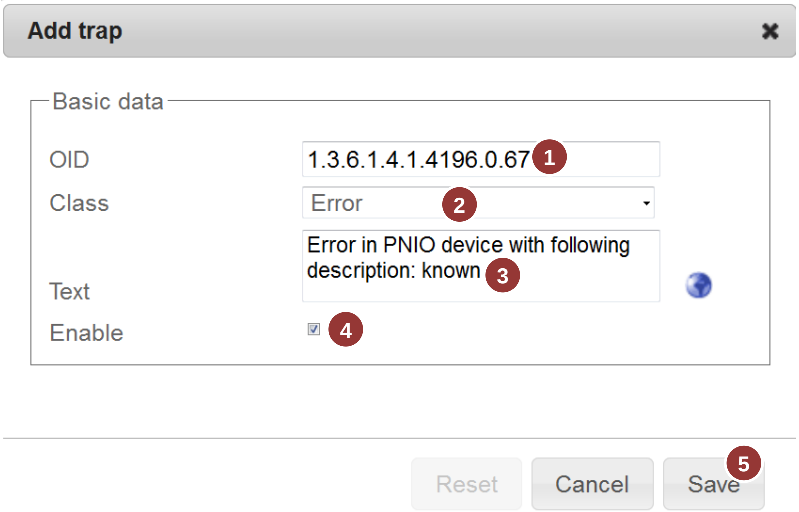
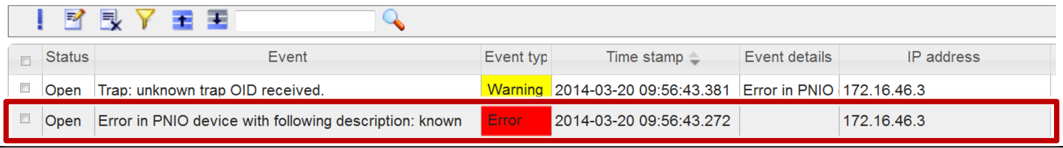
If a device with an already specified IP address is known in the network, the trap is additionally shown in the device log.

7.3.2 Defining trap and sending known trap

Defining trap

The following table describes the procedure to introduce an unknown trap ID to the SINEMA server.

Table 7-5

<table>
<tr><th>No.</th><th>Action</th></tr>
<tr><td>5.</td><td>Copy the trap ID of the unknown trap from the event details of the SINEMA server.

Event details
Error in PNIO device with following description:gone : 1.3.6.1.4.1.4196.0.68
Error in PNIO device with following description:wirebrk : 1.3.6.1.4.1.4196.0.67</td></tr>
<tr><td>6.</td><td>Go to "Administration > Event types" and open the dialog for creating a new trap by clicking "Add new trap".</td></tr>
<tr><td>7.</td><td>Add the OID to the copied OID (❶). Select a desired message class (❷). Supplement a descriptive text (❸).
To enable the trap, enable the "enable" ❹ checkbox and click "save" ❺.

Add trap ✖
Basic data
OID: 1.3.6.1.4.1.4196.0.67
Class: Error
Text: Error in PNIO device with following description: known
Enable: ☑
Reset | Cancel | Save
 </td></tr>
<tr><td>8.</td><td>If a trap with the entered OID is send again, it includes the entered classification (error) and the message text.

<table><tr><th>□</th><th>Status</th><th>Event</th><th>Event typ</th><th>Time stamp</th><th>Event details</th><th>IP address</th></tr><tr><td>□</td><td>Open</td><td>Trap: unknown trap OID received.</td><td>Warning</td><td>2014-03-20 09:56:43.381</td><td>Error in PNIO</td><td>172.16.46.3</td></tr><tr><td>□</td><td>Open</td><td>Error in PNIO device with following description: known</td><td>Error</td><td>2014-03-20 09:56:43.272</td><td></td><td>172.16.46.3</td></tr></table></td></tr>
</table>

8 Appendix

8.1 Service and support

Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

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

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

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers – ranging from basic support to individual support contracts. Please send queries to Technical Support via Web form:
www.siemens.com/industry/supportrequest

SITRAIN – Training for Industry

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that's tailored to the customer's specific needs.

For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page:
www.siemens.com/sitrain

Service offer

Our range of services includes the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contracts

You can find detailed information on our range of services in the service catalog web page:
support.industry.siemens.com/cs/sc

Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for Apple iOS, Android and Windows Phone:
support.industry.siemens.com/cs/ww/en/sc/2067

8.2 Links and literature

Table 8-1

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to this entry page of this application example https://support.industry.siemens.com/cs/ww/en/view/57249109
\3\	SIMATIC STEP 7 Basic/Professional V15.1 und SIMATIC WinCC V15.1 https://support.industry.siemens.com/cs/ww/de/view/109755202
\4\	SIMATIC PROFINET with STEP 7 V15 https://support.industry.siemens.com/cs/ww/de/view/49948856
\5\	PROFINET Industrial Ethernet System Manual https://support.industry.siemens.com/cs/ww/en/view/27069465
\6\	Monitoring and Control of Network Components with an S7 PN-PLC as an SNMP Manager https://support.industry.siemens.com/cs/ww/en/view/57249109
\7\	SINEMA Server https://support.industry.siemens.com/cs/ww/en/view/91198435

8.3 Change documentation

Table 8-2

Version	Date	Modifications
V1.0	04/2014	First version
V2.0	07/2016	Update to TIA Portal V13 SP1
V3.0	07/2019	Update to TIA Portal V15.1