SIEMENS

Application Example • 07/2016

Modifying and Requesting SNMP Variables via S7 PN-CPU and SNMP Library

SIMATIC S7, SCALANCE

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

Warranty and Liability

Note

The Application Examples are not binding and do not claim to be complete with regard to configuration, equipment or any contingencies. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for the correct operation of the described products. These Application Examples do not relieve you of the responsibility of safely and professionally using, installing, operating and servicing equipment. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time and without prior notice. If there are any deviations between the recommendations provided in this Application Example and other Siemens publications – e. g. catalogs – the contents of the other documents shall have priority.

We do not accept any liability for the information contained in this document.
Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act ("Produkthaftungsgesetz"), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of fundamental contractual obligations ("wesentliche Vertragspflichten"). The compensation for damages due to a breach of a fundamental contractual obligation is, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.
Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of Siemens AG.

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 secure plants, systems, machines and networks against cyber threats it is necessary to implement (and to maintain continuously) a holistic, state-of-the-art Industrial Security concept. With this in mind, Siemens' products and solutions are only part of such a concept.

It is the client's responsibility to prevent unauthorized access to his plants, systems, machines and networks. Systems, machines and components should only be connected with the company's network or the Internet, when and insofar as this is required and the appropriate protective measures (for example, use of firewalls and network segmentation) have been taken.

In addition, Siemens' recommendations regarding appropriate protective action should be followed. For more information on Industrial Security, visit http://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them even more secure. Siemens explicitly recommends to carry out updates as soon as the respective updates are available and always only to use the current product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.

In order to always be informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at http://www.siemens.com/industrialsecurity.

Table of Contents

1 Task

Introduction

SNMP is a protocol that has established itself due to its modularity and versatility as a standard for the network management. Among the central tasks of a network management are

- monitoring of network components (network/configuration/status information, statistic data)
- the control and configuration of network components.

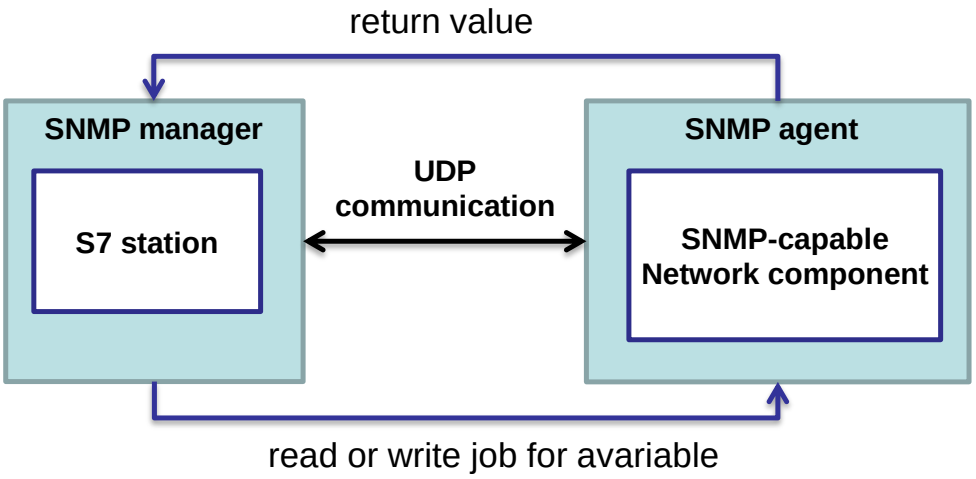
In order to enable these functions also for S7 stations, a “LSnmp” library with universal SNMP blocks was developed. With these blocks a simple system for requesting and controlling different SNMP variables of network components can be realized in the local network.

This application shows the application and use of this library.

Overview of the automation task

The figure below provides an overview of the automation task.

Figure 1-1



Description of the automation task

A S7 station with integrated Ethernet interface is to communicate with a network component via the SNMP protocol.

Here, the S7 station takes on the role of the SNMP manager and the network component that of the SNMP agent.

The aim of this solution is to realize the following scenarios:

- Requesting different SNMP variables from SNMP-capable network components.
- Writing of different SNMP variables in SNMP-capable network components.

The following variables are used as an example:

Table 1-1

Variable	OID Object
sysName Administratively assigned name for the managed node.	1.3.6.1.2.1.1.5.0
ipInReceives The total number of datagrams the device has received.	1.3.6.1.2.1.4.3.0
sysUpTime The time (in hundreds of a second) since the network management portion of the system was last initialized.	1.3.6.1.2.1.1.3.0
sysServices Value that indicates the services on this node.	1.3.6.1.2.1.1.7.0
syslocation Location name for the managed node.	1.3.6.1.2.1.1.6.0

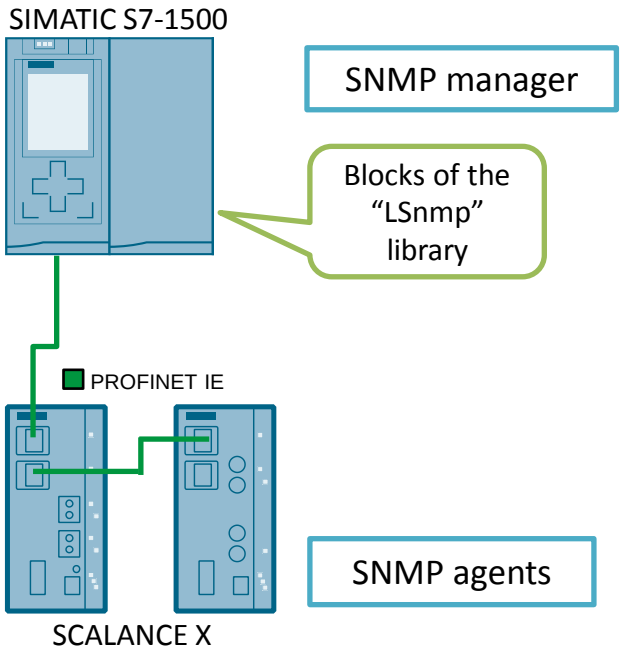
2 Solution

2.1 Solution overview

Schematic layout

The following figure shows the most important components of the solution with an S7-1500 PN CPU. Alternatively, the solution can also be implemented with another PN CPU that supports the UDP protocol (see \3\ in chapter 8).

Figure 2-1



Configuration

To demonstrate the network management via SNMP, a simple, local network is established.
A CPU with integrated PROFINET interface serves as network management station.

For monitoring, SCALANCE X modules are used. These modules have implemented a SNMP agent that makes it possible to detect the state of the network device and also to make own settings or trigger actions.

The SNMP scenarios are controlled via the user program of the CPU. For this purpose are

- the “LSnmp” library
  - blocks “SnmpGet” and “SnmpSet”,
  - global data blocks “SnmpGetParam” and “SnmpSetParam” with the necessary configuration data for the UDP connection, parameters for the SNMP variables to be requested/written and a storage area for the response/write data of the network components
- an individual function block to demonstrate the example application.

Note The SNMP block library and the respective documentation can be found on the same HTML page as this document (see \2\ in chapter 8).

Topics not covered by this application

This application does not include any basic information

- on the LAD/FBD/STL/SCL programming languages
- on the network components

Basic knowledge of these topics is assumed.

Validity

- Application of all current PROFINET CPUs from the SIMATIC product range that support the UDP protocol (see \3\ in chapter 8).
- STEP 7 V13 SP1

2.2 Description of the core functionality

2.2.1 The SNMP protocol

To realize simple network monitoring, this application uses the SNMP protocol.

The SNMP protocol is based on UDP and provides different data packet types for managing and controlling networks.

In turn, every SNMP agent provides a set of SNMP variables which the SNMP manager can request or write on (SNMP-MIB).

The sequence between SNMP manager and SNMP agents is always the same: The manager sends a request (GET/SET request) to the agent. It executes the command and sends back a response (GET response).

The following table shows the assignment of what data packet type is used for the requested scenarios:

Table 2-1

Scenario	SNMP data packet type
Request of SNMP variables	SNMP GET request/ GET response
Writing of SNMP variables	SNMP SET request/ GET response

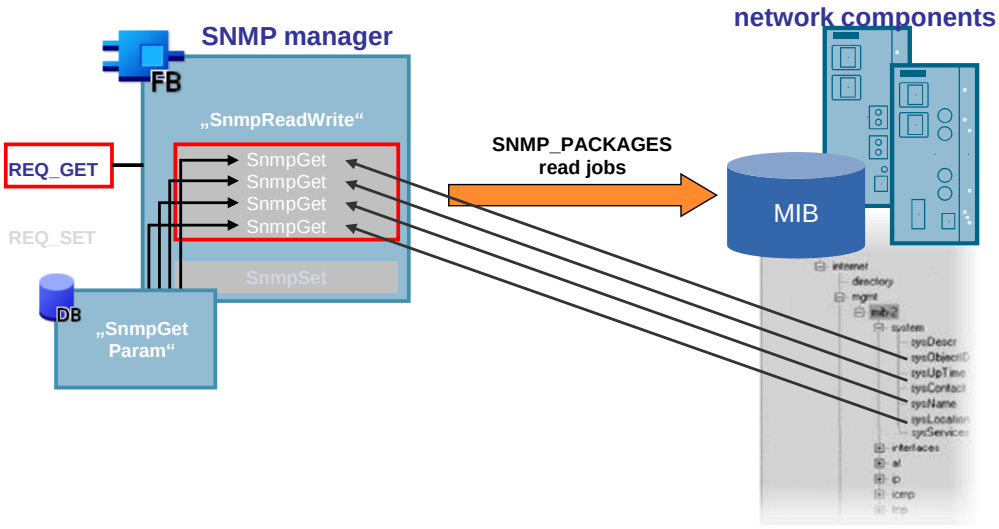
Note More information on the SNMP protocol can be found in chapter 3 (SNMP basics).

2.2.2 Overview of the scenarios of the example application

In this example, the required scenarios are implemented with a FB “SnmpReadWrite” function block (programmed in FBD) and with the aid of the universal SNMP library blocks (FB “SnmpSet” and FB “SnmpGet”).

Scenario “Requesting different SNMP variables”

Figure 2-2



This scenario shows how the SNMP manager (CPU) can request SNMP variables from a SNMP-capable network component (SCALANCE). For this purpose, the user sends defined read jobs (GET request) to the network components with the aid of the FB “SnmpReadWrite” user block.

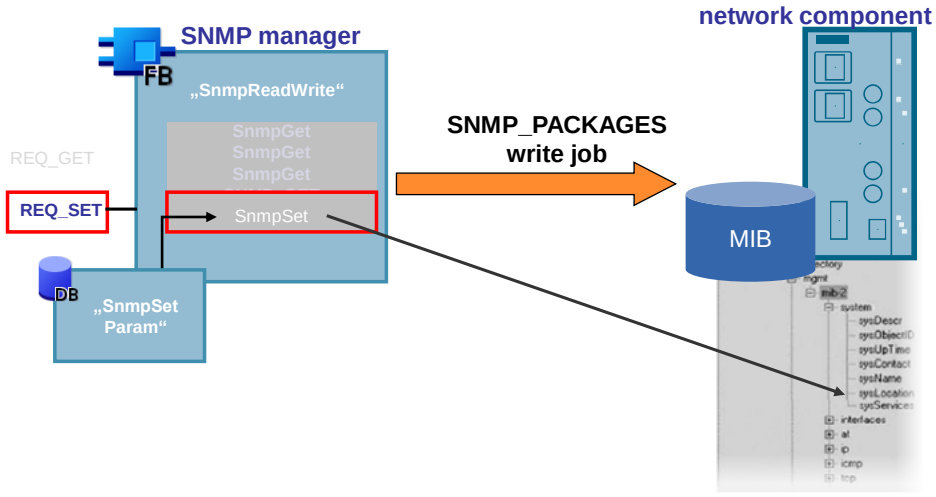
The information required for this purpose is stored in the “SnmpGetParam” data block. Here, in this example you read out the following variables:

Table 2-2

Variable	OID Object
sysName Administratively assigned name for the managed node.	1.3.6.1.2.1.1.5.0
ipInReceives The total number of datagrams the device has received.	1.3.6.1.2.1.4.3.0
sysUpTime The time (in hundreds of a second) since the network management portion of the system was last initialized.	1.3.6.1.2.1.1.3.0
sysServices Value that indicates the services on this node.	1.3.6.1.2.1.1.7.0

Scenario “Writing a SNMP variable”

Figure 2-3



This scenario shows how the SNMP manager (CPU) can write on a SNMP variable in a SNMP-capable network component (SCALANCE). For this purpose, the SNMP manager sends defined write job (SET request) to the network components with the aid of the FB “SnmpReadWrite” user block. The information required for this purpose is stored in the “SnmpSetParam” data block. Here, in this example the following variable is described:

Table 2-3

Variable	OID Object
syslocation Location name for the managed node.	1.3.6.1.2.1.1.6.0

2.3 Hardware and software components used

This application was created with the following components:

Hardware components

Table 2-4

Component	Qty.	Article number	Note
CPU 1516-3 PN/DP	1	6ES7516-3AN00-0AB0	Alternatively, any other S7 PN-CPU that supports the UDP protocol can also be used.
S7 MEMORY CARD	1	6ES7954-8LC01-0AA0	for example, 4 KB; size depends on your user program
Industrial Ethernet twisted pair cable	3	6XV1 850-2GH60	For the communication between controller and SCALANCE X
SCALANCE X202-2IRP	1	6GK5 202-2BB00-2BA3	Alternatively, any other SNMP-capable network component can also be used.
SCALANCE X202-2P IRP	1	6GK5 202-2BH00-2BA3	Alternatively, any other SNMP-capable network component can also be used.

Standard software components

Table 2-5

Component	Qty.	Remark
TIA V13 SP1	1	Or higher
Primary Setup Tool	See \5\ in chapter 8	

Example files and projects

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

Table 2-6

Component	Note
57249109_SNMP_RW_CODE_V3_0.zip	This zipped file includes a simple application example for the S7-1500.
57249109_SNMP_RW_ DOKU_V3_0_en.pdf	This document

3 SNMP basics

3.1 The SNMP protocol

Meaning and purpose of SNMP

SNMP – **S**imple **N**etwork **M**anagement **P**rotocol – is a UDP-based protocol that was specified for the administration of data networks and has established itself as the de facto standard for TCP/IP devices. For SNMP the UDP ports 161 and 162 are used.

The architecture of SNMP

In accordance to the architecture model of SNMP, a network is distributed as follows:

- Stations for the network management (management stations, SNMP manager)
- Network components

The management stations place requests to the network components to monitor and control them.

The network components (for example, routers, switches) have so called SNMP agents which are typically implemented as firmware function. The SNMP agents respond to the requests of the management stations.

The communication

The communication between management station and network components takes place via SNMP and comprises

- the transfer of relevant management data that was collected from the network components and stored in the MIB
- the control data for the network components that are defined in the MIB.

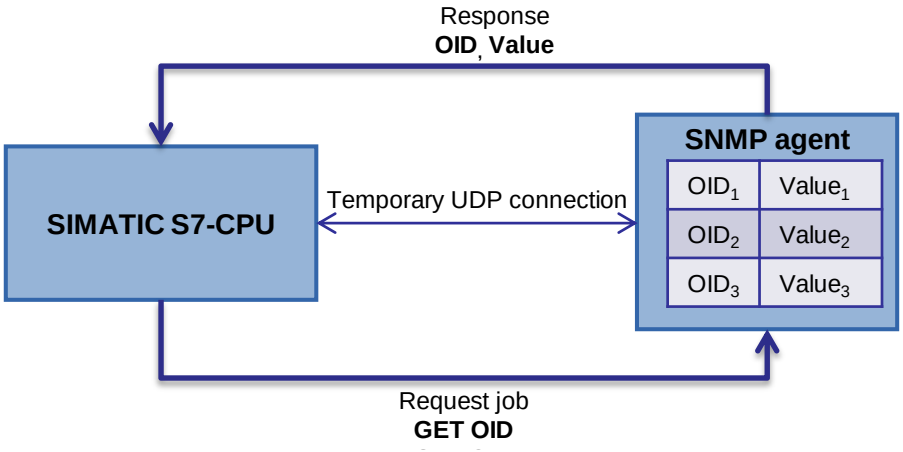
The communication takes place in the background and usually only puts insignificant stress on the network.

Forms of communication

The basic functionality of SNMP is mainly based on two different forms of communication.

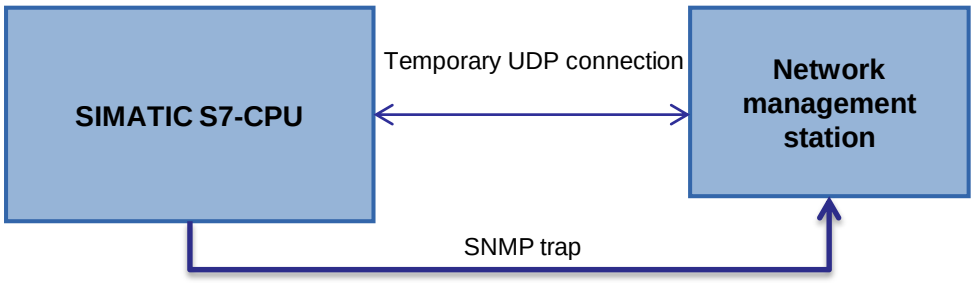
The first type of communication is bidirectional and according to the client/server model. Here, the management station (client) places a request to the monitored network component (server) which will be replied by it (response). This form of communication allows the monitoring (GET request) as well as the managing (SET request) of the network components.

Figure 3-1



For the other form of communication, messages ("traps") are sent unidirectional; only from the managed network component to the management station. This communication path enables the network component to quickly inform the management station, if unusual events have occurred.

Figure 3-2



3.2 Management Information Base – MIB

SNMP database

The information and data which the management station can request or write from the network component is stored in the so called MIB (Management Information Base).

The MIB is comparable with a database and provides a standardized, hierarchical data structure from different SNMP variables. The SNMP variables are structured in form of objects and referenced via an OID (Object Identifier). Each SNMP-capable network component is delivered with a collection of standard MIBs. If component-specific, non-standardized data is necessary for network monitoring, this data can be described by the manufacturers in so-called “private MIBs”.

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.

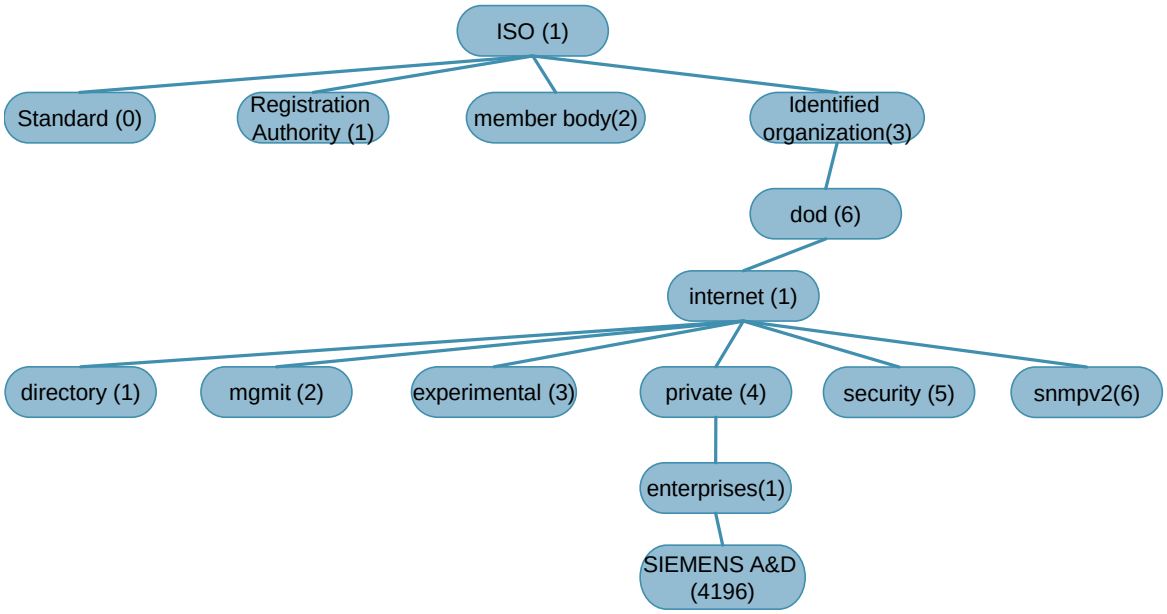
Object identifier

The OID (object identifier) describes the address of the MIB object in the hierarchical structure. 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 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 3-3



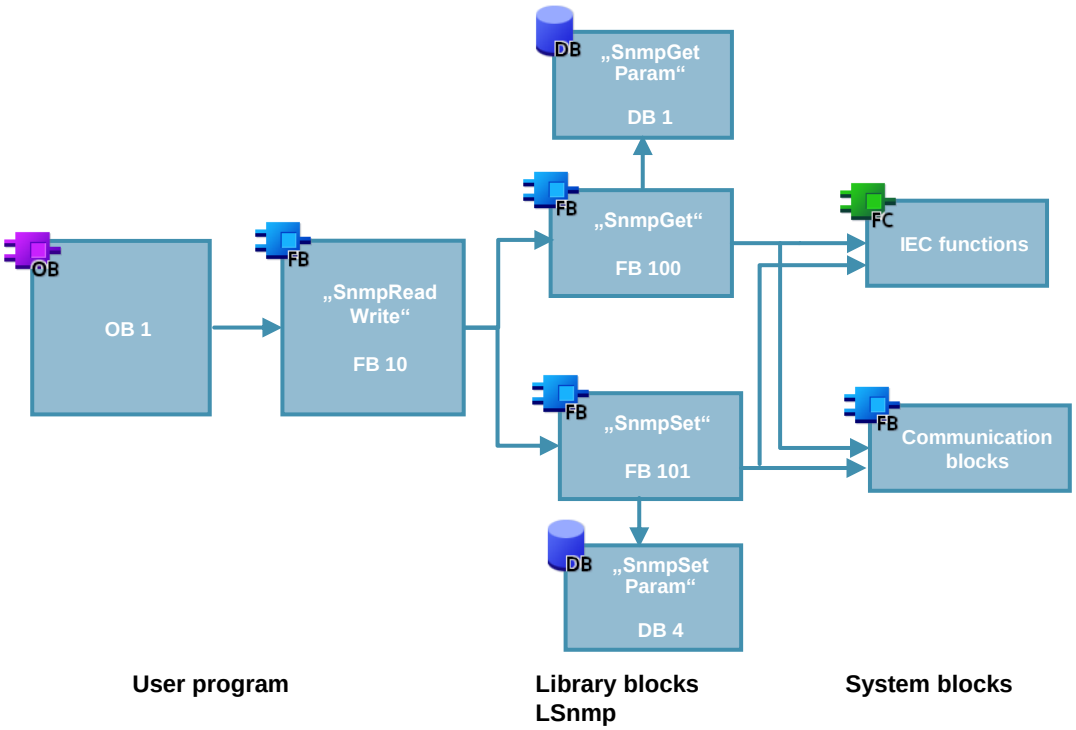
4 Functional Mechanisms of this Application

4.1 Program overview

Graphic display

The graphic below shows the program structure of the entire STEP 7 project.

Figure 4-1



Program blocks

The following blocks are used in detail for the example application:

Table 4-1

Block	Function	Comment
FB “SnmpReadWrite”	• Writes an SNMP variable with the “SnmpSet” SNMP library block • Request of different SNMP variables with the “SnmpGet” SNMP library block	Individual function block
FB “SnmpSet”	Controls an SNMP variable	Universal SNMP library block
FB “SnmpGet”	Request a SNMP variable	Universal SNMP library block
DB “SnmpSetParam”	Includes all required information to control a SNMP variable.	Universal SNMP library block
DB “SnmpGetParam”	Includes all required information to request SNMP variables.	Universal SNMP library block

Application block “SnmpReadWrite”

The “SnmpReadWrite” function block is the central application block and is called cyclically in OB1.

In order to demonstrate the requested scenarios

- request of different SNMP variables from SNMP-capable network components.
- writing of a SNMP variable from a SNMP-capable network component.

the function block calls uses “LSnmp” and internally calls the function blocks “SnmpGet” and “SnmpSet”.

Library “LSnmp”

The blocks “SnmpSet” and “SnmpGet” are part of the “LSnmp” SNMP library. For this library an own document exists with a detailed block description. For this reason there is no description of the library blocks below.

The blocks are already integrated in this example application.

Note The SNMP block library as well as a detailed library description can be found on the same HTML page as this document.
(see \2\ in chapter 8)

4.2 Requesting SNMP variables

Overview

The FB “SnmpReadWrite” function block sends read jobs to two network components after a trigger command and requests the value of four variables (see Table 4-2).

For this purpose, the “SnmpGet” library block is used with the respective “SnmpGetParam” data block.

In the “SnmpGetParam” data block, all communication and SNMP-relevant data is saved.

Figure 4-2

SnmpGetParam

Name	Datentyp
Static	
SNMP	"typeParamGetSet"
ipAddress	DWord
hwIdentifier	HW_ANY
connectionID	Word
localPort	Word
oID	String[254]
community	String[20]
returnValueType	Byte
returnValueLenght	Byte
returnValue	Array[1..255] of Byte

SNMP variables

The following SNMP variables are read out of the network components:

Table 4-2

Variable	OID Object	Network component
sysName Administratively assigned name for the managed node.	1.3.6.1.2.1.1.5.0	SCALANCE X202-2P IRT
ipInReceives The total number of datagrams the device has received.	1.3.6.1.2.1.4.3.0	SCALANCE X202-2P IRT
sysUpTime The time (in hundreds of a second) since the network management portion of the system was last initialized.	1.3.6.1.2.1.1.3.0	SCALANCE X202-2IRT
sysServices Value that indicates the services available on this node.	1.3.6.1.2.1.1.7.0	SCALANCE X202-2IRT

Flow chart

The graphic below describes how the FB “SnmpReadWrite” sends the four read jobs to the two devices to request the value of the above-listed variables.

Figure 4-3

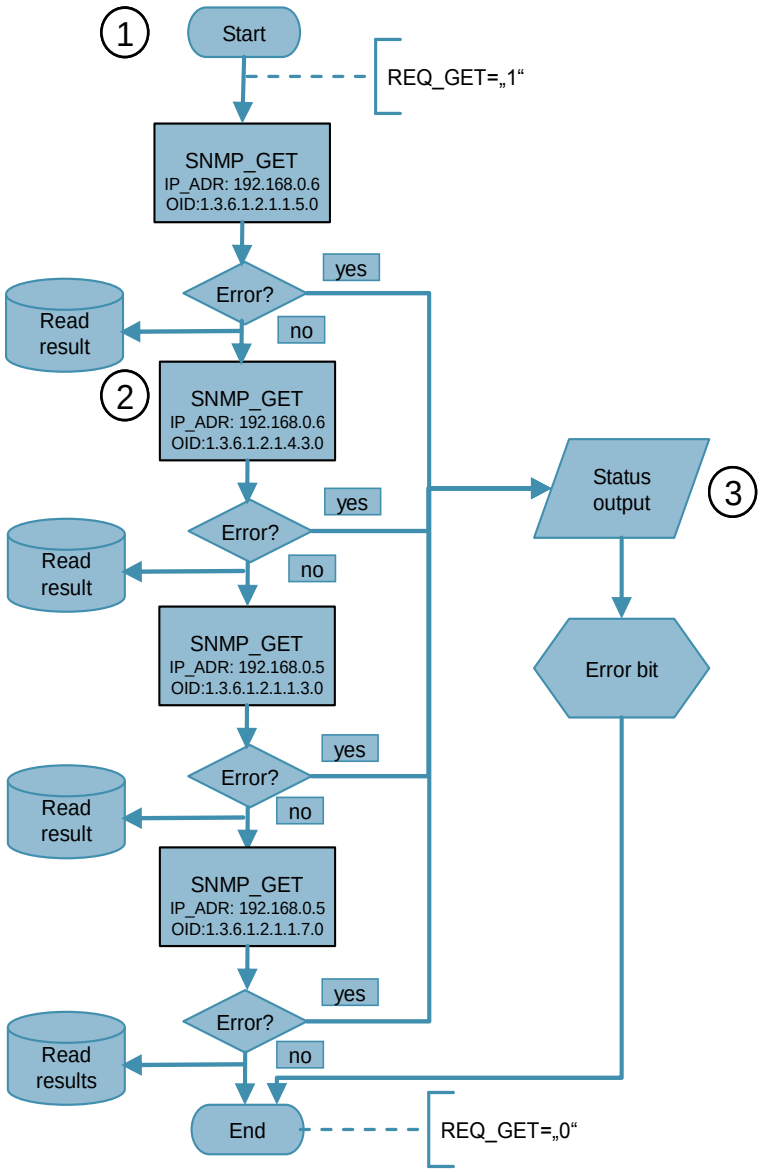


Table 4-3

No.	Description
1	The first function block is called with a positive edge on REQ_GET.
2	When there is a negative edge on the “BUSY” parameter and it has been processed without error, the next function block will be called. The received data is saved in DB “SnmpGetParam”.
3	If there are errors during the read process, the status messages are stored in the error area in DB “SnmpGetParam” and the read process is stopped.

4.3 Writing or controlling SNMP variable

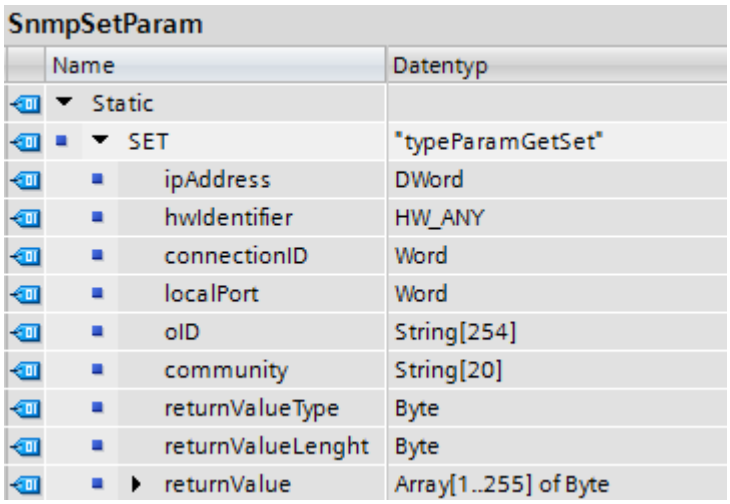
Overview

The FB “SnmpReadWrite” function block sends a write job to a network component after a trigger command and writes the value of a variable.

For this purpose, the “SnmpSet” SNMP library block is used with the respective “SnmpSetParam” data block.

In the “SnmpSetParam” data block, all communication and SNMP-relevant data is saved.

Figure 4-4



SNMP variable

The following SNMP variable is written in the network component:

Table 4-4

Variable	OID Object	Network component
syslocation Location name for the managed node.	1.3.6.1.2.1.1.6.0	SCALANCE X202-2P IRT

Flow chart

The following graphic shows a schematic overview of how the FB “SnmpReadWrite” sends a write job to the device to define the location where the device is located.

Figure 4-5

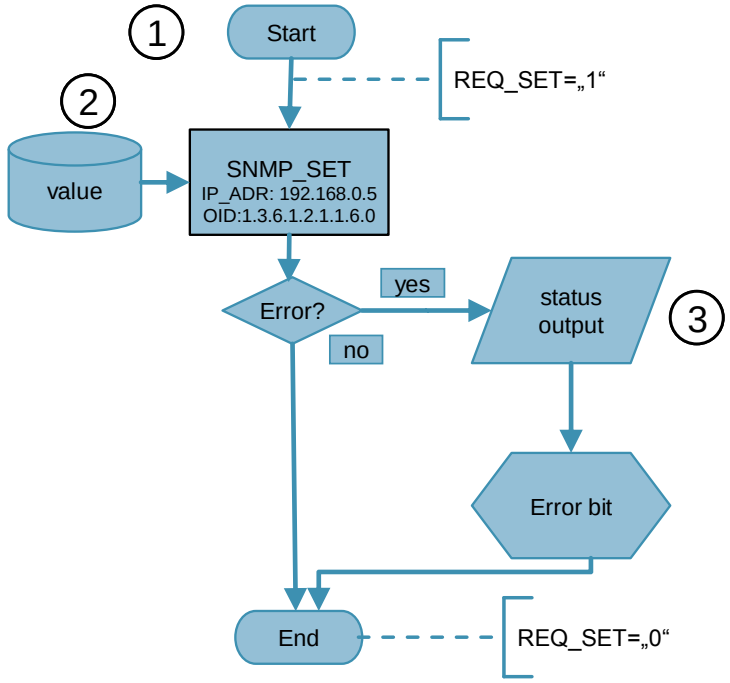


Table 4-5

No.	Description
1	The function block is called if there is a positive edge on REQ_SET.
2	The required parameters are saved in DB “SnmpSetParam”.
3	If there are errors during the write process, the status messages are stored in the error area in DB “SnmpSetParam” and the write process is stopped.

4.4 Error and status display

Each library bock (“SnmpGet”, “SnmpSet”) called in FB “SnmpReadWrite” has a STATUS output for error diagnostics.
Statements on logical errors and on error messages can be read out via this STATUS output which can occur during the communication between controller and network components.

Since the STATUS output is only available for one cycle, the values are saved in DB “SnmpGetParam” or DB “SnmpSetParam”.

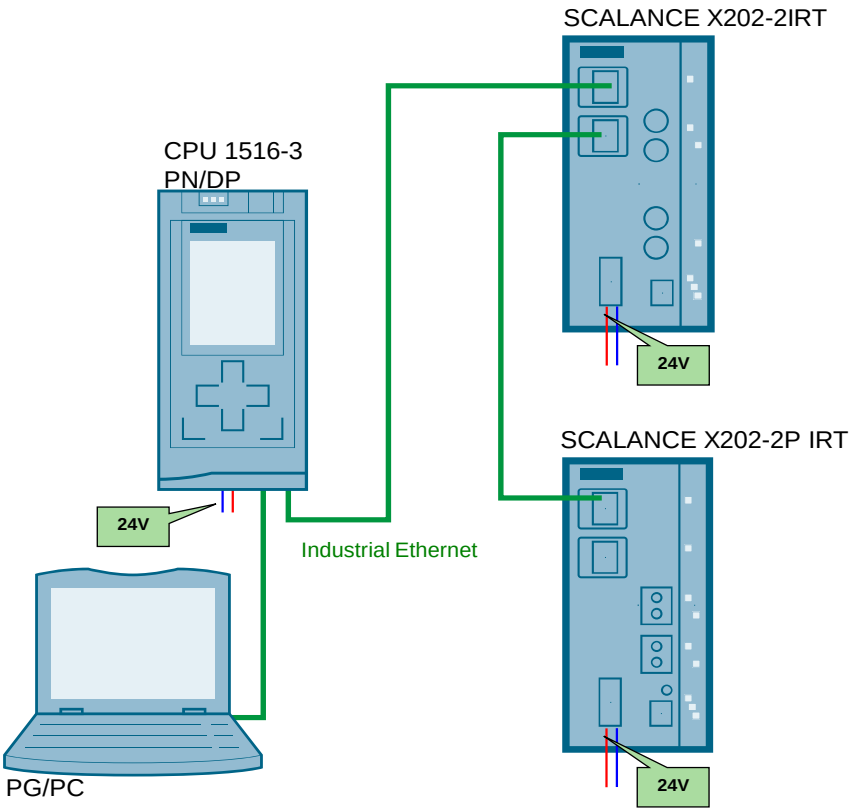
Note Information on error messages can be referred to in the library description “57249109_Library_SNMP_S7_CPU_DOKU_V3_0_en.pdf” (see \2\ in chapter 8).

5 Installation and Commissioning

5.1 Installing the hardware

The figure below shows the hardware configuration of the application on the example of a S7-1500 CPU.

Figure 5-1



Install the individual modules on a suitable module rack and connect them to 24V. Interconnect the following PROFINET interfaces with each other:

- PG/PC with the S7-CPU
- S7 CPU and SCALANCE X202-2IRT
- SCALANCE X202-2P IRT and SCALANCE X202-2IRT

NOTICE **Before you switch on the power supply, complete and check the installation!**

Note The installation guidelines for the installation of all components always have to be observed.

5.2 Installing the software

Engineering software

The engineering station is used as a configuration computer for the S7 station.

Before you start commissioning the application, you must install TIA Portal. Follow the instructions of the installation program.

Application software

The project is available on the HTML page from which you downloaded this document. Save the "**57249109_SNMP_RW_CODE_V3_0.zip**" project and unzip the folder.

5.3 Commissioning the application

5.3.1 IP address assignment

The following table provides an overview of the IP addresses used in this sample program.

Table 5-1

Module	IP address	PROFINET name
CPU 1516	192.168.0.1	-
PC/PG	192.168.0.3	-
SCALANCE X202-2IRT	192.168.0.5	X202-2
SCALANCE X202-2P IRT	192.168.0.6	X202-2P

Changing IP address of the PC

Change the network settings of the PC in accordance with Table 5-1. If your PG has an IWLAN interface, switch it off.

Changing IP address of the modules

Before you can load the STEP 7 project into the CPU and configure the modules, you have to change the IP address of the modules according to Table 5-1.

The Primary Setup Tool is suitable for assigning the IP address. You can carry out an address assignment with the Primary Setup Tool (PST) for SIMATIC NET network components, Ethernet CPs and network transitions.

Information on the PST can be found in the manual (see \5\ in chapter 8).

5.3.2 Configuring the network components

For this application, the SCALANCE X202-2IRT/SCALANCE X202-2P IRT is commissioned in two steps:

- Enable SNMP access
- Change PROFINET device name

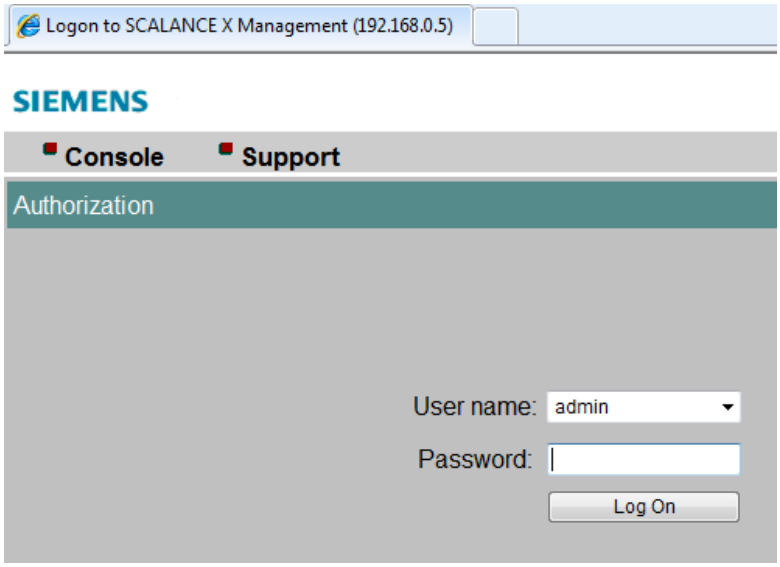
Note For general information on the configuration of the device, please refer to the configuration manual of the SCALANCE X200 devices. The aim of this manual is to enable you to correctly install, commission and operate the devices (see link \4\ and \6\ in chapter 8).

Opening web-based management

The SCALANCE devices are configured via the web-based management. Open an Internet Browser and enter the IP address of a SCALANCE module in the address line.

Log on as administrator.

Figure 5-2



Enabling SNMP access

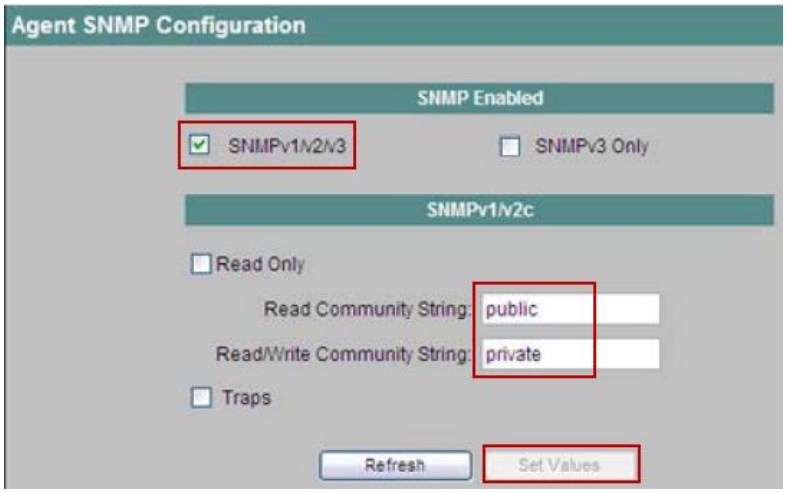
Go to the “Agent” menu and select the SNMP configuration overview.

Enable “SNMPv1/v2/v3” and enter the community names for the read and write access:

- For read access: public
- For write access: private

Accept the change with “Set Values”.

Figure 5-3



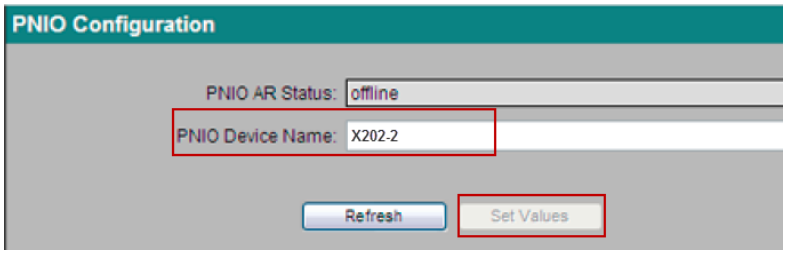
Adjusting PROFINET device name

For a PROFINET communication between CPU and network components a unique PROFINET name is crucial.

Select the PNIO configuration overview in the same menu item and enter the assigned PROFINET device name in the “PNIO Device Name” box in accordance with Table 5-1.

Accept the change with “Set Values”.

Figure 5-4



Note This step is necessary since the network components are integrated in the hardware configuration and a PROFINET communication has been configured. For a pure UDP communication a PROFINET communication is not a prerequisite.

5.3.3 Loading the TIA Portal project

Open the project extracted in the TIA Portal and load the project into the controller.

If the download process is completed without errors, start the controller.

6 Operating the Application

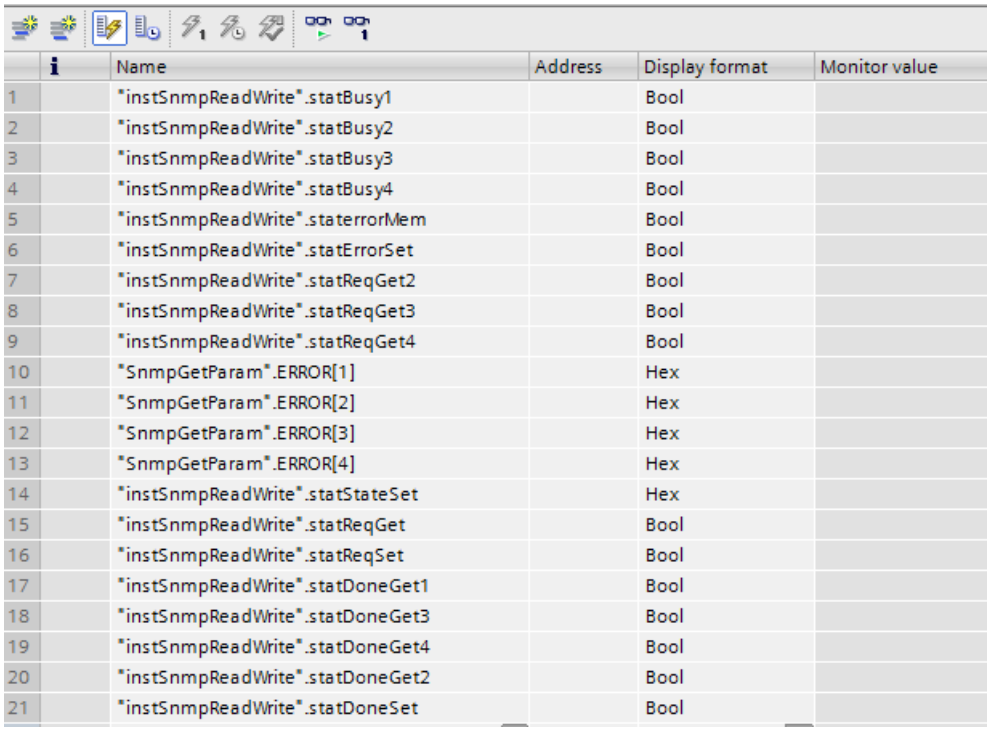
Overview

For convenient operation of the application the project has a “ReadWrite” watch table.

All variables that are required for the scenarios are integrated here:

- Trigger bits
- Status message
- Received data from the “request SNMP variables” scenario
  - Length of received data
  - Response data regarding the SNMP variables
- Data to be transmitted for the “write SNMP variable” scenario
  - Data type
  - Length
  - Information

Figure 6-1



	Name	Address	Display format	Monitor value
1	"instSnmpReadWrite".statBusy1		Bool	
2	"instSnmpReadWrite".statBusy2		Bool	
3	"instSnmpReadWrite".statBusy3		Bool	
4	"instSnmpReadWrite".statBusy4		Bool	
5	"instSnmpReadWrite".staterrorMem		Bool	
6	"instSnmpReadWrite".statErrorSet		Bool	
7	"instSnmpReadWrite".statReqGet2		Bool	
8	"instSnmpReadWrite".statReqGet3		Bool	
9	"instSnmpReadWrite".statReqGet4		Bool	
10	"SnmpGetParam".ERROR[1]		Hex	
11	"SnmpGetParam".ERROR[2]		Hex	
12	"SnmpGetParam".ERROR[3]		Hex	
13	"SnmpGetParam".ERROR[4]		Hex	
14	"instSnmpReadWrite".statStateSet		Hex	
15	"instSnmpReadWrite".statReqGet		Bool	
16	"instSnmpReadWrite".statReqSet		Bool	
17	"instSnmpReadWrite".statDoneGet1		Bool	
18	"instSnmpReadWrite".statDoneGet3		Bool	
19	"instSnmpReadWrite".statDoneGet4		Bool	
20	"instSnmpReadWrite".statDoneGet2		Bool	
21	"instSnmpReadWrite".statDoneSet		Bool	

To view the current values go online to your controller via the icon  .

.

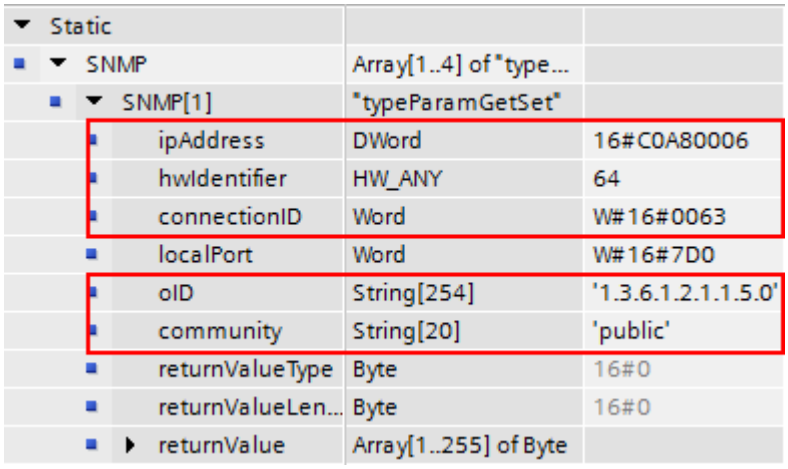
Preparation

If you deviate from the specified application configuration, you have to adjust the DB “SnmpGetParam” or “SnmpGetParam” to your requirements.

Possible reasons for a deviation:

- Use of other IP addresses for the network nodes.
- Differences at DEVICE ID parameter by using a different CPU.
- The specified connectionID is already in use in your project.
- You would like to write/read a different SNMP variable.

Figure 6-2



Static		
SNMP	Array[1..4] of "type...	
SNMP[1]	"typeParamGetSet"	
ipAddress	DWord	16#C0A80006
hwIdentifier	HW_ANY	64
connectionID	Word	W#16#0063
localPort	Word	W#16#7D0
oID	String[254]	'1.3.6.1.2.1.1.5.0'
community	String[20]	'public'
returnValueType	Byte	16#0
returnValueLen...	Byte	16#0
returnValue	Array[1..255] of Byte	

Change the parameters according to your specifications. Save the block and load it again into the CPU. For this purpose, use the
“Online > Download and reset PLC program” download function.

6.1 Requesting SNMP variables

Operation

The request of the status of the different variables is controlled via the instance DB, and the “InstSnmpReadWrite”.statReqGet” variable.

As soon as a positive edge is recognized, the defined SNMP variables are read out of the network components.

Connect online with the controller and set the appropriate variable to the value “true” in the variable table.

The four defined SNMP variables are requested successively.

Result

The read result is stored in the respective “SnmpGetParam”. SNMP[x].returnValue (x=1…4) areas.

If an error occurs during the reading of the SNMP variables, the error messages are saved in the error area in the DB “SnmpGetParam”, and the “statErrorMem” variable from the instance DB “InstSnmpReadWrite” is set.

6.2 Writing or controlling SNMP variable

Defining values

You have to change the following parameters if you want to write the “syslocation” SNMP variable with a different value than the specified one:

- Data type of the transferred values
- Length of the value
- Value

Figure 6-3

Static		
SET	"typeParamGetS...	
ipAddress	DWord	16#C0A80005
hwIdentifier	HW_ANY	64
connectionID	Word	W#16#0065
localPort	Word	W#16#07D0
oID	String[254]	'1.3.6.1.2.1.1.6.0'
community	String[20]	'private'
returnValueType	Byte	16#4
returnValueLenght	Byte	16#8
returnValue	Array[1..255] of Byte	

Change the parameters according to your specifications. Save the block and load it again into the CPU. For this purpose, use the “Online > Download and reset PLC program” download function.

Operation

The writing of the variables is controlled via the instance DB and the “InstSnmpReadWrite“.statReqSet” variable.

As soon as a positive edge is detected, the state machine is started and the setting up and sending of the set request packet is triggered.

Connect online with the controller and set the appropriate variable to the value “true” in the variable table.

If an error occurs during the writing of the SNMP variable, the error message is saved in the error area in the DB “SnmpSetParam”, and the “statErrorMem” variable from the instance DB “InstSnmpReadWrite” is set.

7 Glossary

MIB

The Management Information Base (abbrev.: MIB) describes the information that can be requested or modified via a network management protocol (for example SNMP). This information is called managed objects.

OID

In computer science, an object identifier (OID) is a globally unique identifier used to name an information object. An OID represents a node in a hierarchically-assigned namespace that is defined by the ASN.1 standard. Each node is uniquely identified by successive numbers that, starting at the root of the tree, indicate its position.

SNMP

SNMP means Simple Network Management Protocol. This protocol allows network administrators to manage network issues and analyze network problems.

UDT

The User Datagram Protocol (UDP) is a minimal, connectionless network protocol that belongs to the transport layer of the Internet protocol family. It is the task of UDP to transfer data that is transmitted over the Internet to the right application.

8 References

This list is by no means complete and only presents a selection of suitable information.

Table 8-1

	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Download page of this entry https://support.industry.siemens.com/cs/ww/de/view/57249109
\3\	Overview of communication services supported by PN-CPUs https://support.industry.siemens.com/cs/ww/en/view/18909487
\4\	Industrial Ethernet Switches SCALANCE X-200 Operating Instructions https://support.industry.siemens.com/cs/ww/en/view/25508728
\5\	PST manual https://support.industry.siemens.com/cs/ww/en/view/19440762
\6\	Industrial Ethernet Switches SCALANCE X-200 Configuration Manual https://support.industry.siemens.com/cs/ww/en/view/74609661
\7\	local_device_id https://support.industry.siemens.com/cs/ww/en/view/51339682

9 History

Table 9-1

Version	Date	Modifications
V1.0	04/2012	First version
V2.0	05/2014	Modification of program code, integration of the S7-1500
V3.0	07/2016	Modification of the program code; update to V 13 SP1