

SIEMENS

Ingenuity for life

Industry Online Support

Home

SNMP blocks for SIMATIC S7 PN CPUs for the diagnostics and control of network components

SCALANCE X/W/M/ STEP 7 V15.1, STEP 7 V16/ V4.0

<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

	Legal information	2
1	Library overview	4
1.1	General	5
1.2	Functions	6
1.3	Hardware and software requirements	9
2	Function blocks of the library "LSnmp"	10
2.1	The library block "LSnmp_Get"	12
2.1.1	Functionality	12
2.1.2	Program flow chart	12
2.1.3	Call structure and parameter interface	13
2.2	The library block "LSnmp_GetNext"	15
2.2.1	Functionality	15
2.2.2	Call structure and parameter interface	15
2.3	The library block "LSnmp_GetBulk"	17
2.3.1	Functionality	17
2.3.2	Call structure and parameter interface	17
2.4	The library block "LSnmp_Set"	19
2.4.1	Functionality	19
2.4.2	Program flow chart	19
2.4.3	Call structure and parameter interface	20
2.5	The library block "LSnmp_SwitchIO"	22
2.5.1	Functionality	22
2.5.2	Program flow chart	23
2.5.3	Call structure and parameter interface	24
2.6	The library block "LSnmp_SendTrap"	26
2.6.1	Functionality	26
2.6.2	Program flow chart	26
2.6.3	Call structure and parameter interface	27
2.7	The PLC data type "LSnmp_typeParamGetSet"	29
2.8	The PLC data type "LSnmp_typeParamGetBulk"	31
2.9	The PLC data type "LSnmp_typeBulkResponseData"	32
2.10	The PLC data type "LSnmp_typeParamSwitchIO"	33
2.11	The PLC data type "LSnmp_typeParamSendTrap" S7-1500/S7-1200 CPU	34
2.12	The PLC data type "LSnmp_typeParamSendTrap" S7-300/S7- 400 CPU	36
2.13	The PLC data type "LSnmp_typeTrapInfo" S7-300/S7-400 CPU	37
2.14	Signal Flow Chart of the library function blocks	38
2.15	Status and error displays	39
3	Activating SNMP in network components	41
4	Working with the library	43
4.1	Integrating the library in STEP 7	43
4.2	Integrating the library block "LSnmp_Get" into STEP 7	43
4.3	Updating the library "LSnmp"	49
5	Appendix	51
5.1	Service and Support	51
5.2	Links and Literature	52
5.3	Change Documentation	52

1 Library overview

The status of SNMP-capable network components is monitored and controlled, if applicable, via SNMP (Simple Network Management Protocol) by network management systems (e.g. SINEC NMS).

The blocks of the library "LSnmp" also allow a SIMATIC S7-CPU with a PROFINET interface to retrieve information from the network components and control them where applicable as a simple SNMP manager.

The "LSnmp" library contains two folders with the respective types:

- S7_1xxx
for controlling and monitoring SNMP-capable network components of a S7-1500/S7-1200 PN-CPU.
- S7_xxx
for controlling and monitoring SNMP-capable network components of a S7-300/S7-400 PN-CPU.

The core concern of the document is the description of the application of SNMP modules and how their features are implemented.

In addition, this documentation shows possible application areas and helps you integrate the library into your STEP 7 project with step-by-step instructions.

Scope of the library

The library is valid for:

- STEP 7 V16
- STEP 7 V15.1, update 3
- SIMATIC S7-1200 CPUs V4.0 or higher
- SIMATIC S7-1500 CPUs
- SIMATIC S7-400 PN/DP
- SIMATIC S7-300 PN/DP (firmware 2.5 and higher)
- SNMPv1/SNMPv2

Note

This document describes the function of the modules in the STEP 7 V16 library "LSnmp" for the S7-1500/S7-1200 CPU.

The differences from the S7-300/S7-400 CPU are explained in the relevant sections.

This description also applies to STEP 7 V15.1 library "LSnmp."

1.1 General

An S7 station can use the library “LSnmp” e.g.:

- to query the current transmission power of an IWLAN access point
- to query the link status of a port from a switch
- to send user-defined traps to the network management system (e.g. SINEC NMS)
- to switch the digital output of an IWLAN client

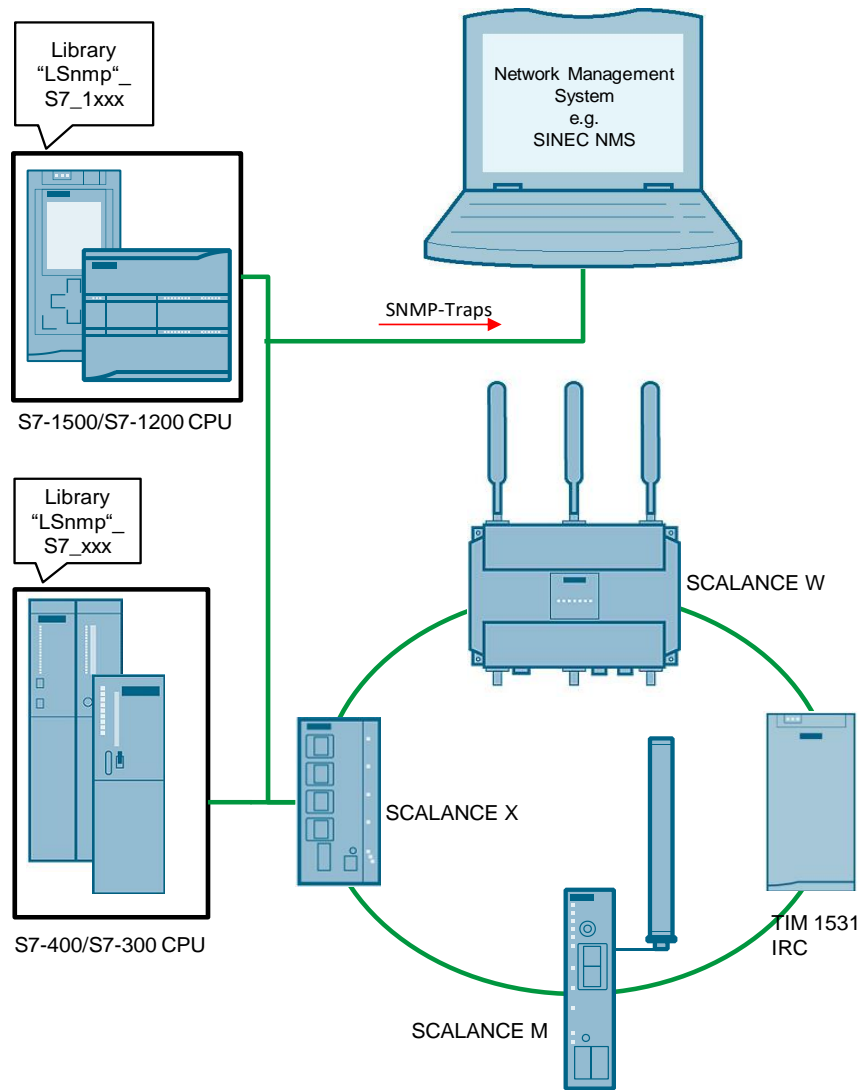
Note

These variables must be present in the device’s private MIB (see [3](#)) or in the general MIB 2.

Diagram

The following figure shows one possible system configuration in which the SNMP blocks of the “LSnmp” library can be used.

Figure 1-1



1.2 Functions

Overview

To allow SNMP communication between a SIMATIC S7-1500/S7-1200 CPU and the network component, the function blocks from the library "LSnmp"_S7_1xxx are needed.

To allow SNMP communication between a SIMATIC S7-300/S7-400 CPU and the network component, the function blocks from the library "LSnmp"_S7_xxx are needed.

Core functions of the function blocks

The following table describes the core functions of the function blocks in the library "LSnmp."

Table 1-1

Function	Description
LSnmp_Get	Requests an SNMP variable from an SNMP agent (SNMPv1)
LSnmp_GetNext	Automatically runs through and retrieves successive objects within an OID subtree (SNMPv1)
LSnmp_GetBulk	Retrieves large volumes of data for an SNMP agent with only one response telegram (SNMPv2)
LSnmp_Set	Modifies an SNMP variable of an SNMP agent (SNMPv1)
LSnmp_SwitchIO	Switches the digital output of an IWLAN client (SNMPv1)
LSnmp_SendTrap	Sends user-defined traps to the network management system

Note

To keep the network load low, the connectionless UDP protocol is used to send messages. The agent and the manager communicate (requests/responses) on port 161.

Port 162 is required for sending TRAP messages.

The following SNMP variables are used as examples:

Table 1-2

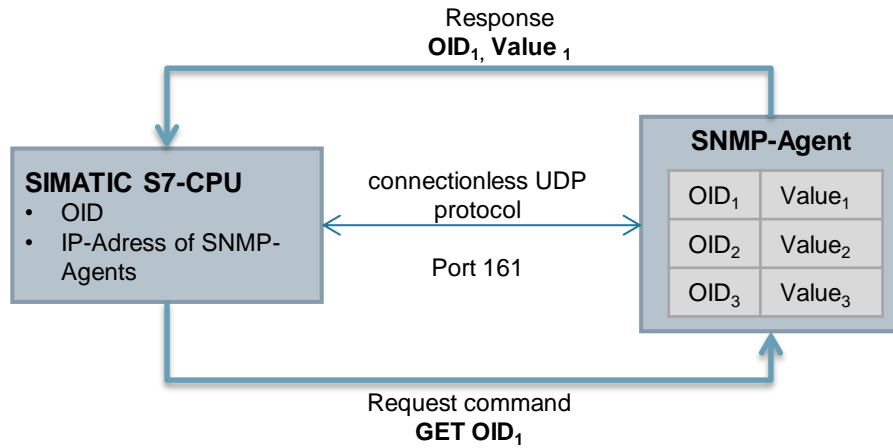
Variable	OID object
sysName (Read-Write, String) Name allocated to managed nodes by administration	1.3.6.1.2.1.1.5.0
sysServices (Read-Only, Integer) Value that describes the services available on this node	1.3.6.1.2.1.1.7.0
Syslocation (Read-Write, String) Place name for managed nodes	1.3.6.1.2.1.1.6.0
sysUpTime (Read-Only, TimeTicks) Time (in hundredths of a second) since System NM part was last initialized	1.3.6.1.2.1.1.3.0

Process when requesting an SNMP variable (Get request command)

When an SNMP variable is requested, the S7 CPU (SNMP Manager) sends a Get request command to the SNMP agent via the connectionless UDP protocol. The telegram contains the OID of the SNMP variable to be retrieved.

After receiving the response, the UDP end point will be switched off.

Figure 1-2

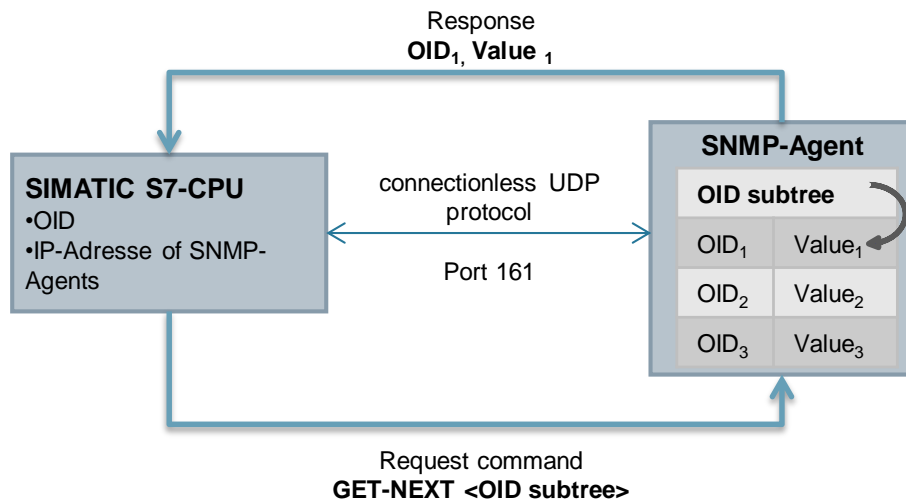


Process during the GetNext request (Get Next request command)

With the GetNext request (GetBulk request), the S7 CPU (SNMP Manager) sends a GetNext request command to the SNMP agent via the connectionless UDP protocol. The telegram contains the preceding OID (usually the stem OID of the subtree) of the SNMP variable to be retrieved (OID₁, Value₁).

After running through the OID subtree, the UDP end point will be switched off.

Figure 1-3

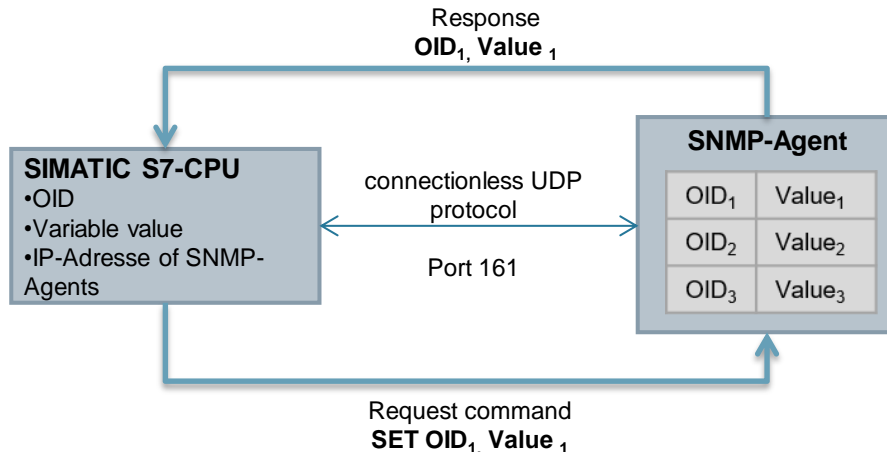


Process during the Write of a SNMP variable to be retrieved (Set request command)

When writing an SNMP variable, the S7 CPU sends a Set request command to the SNMP agent via the connectionless UDP protocol. The telegram contains the OID of the SNMP variable to be retrieved and the new variable value.

After receiving the response, the UDP end point will be switched off.

Figure 1-4



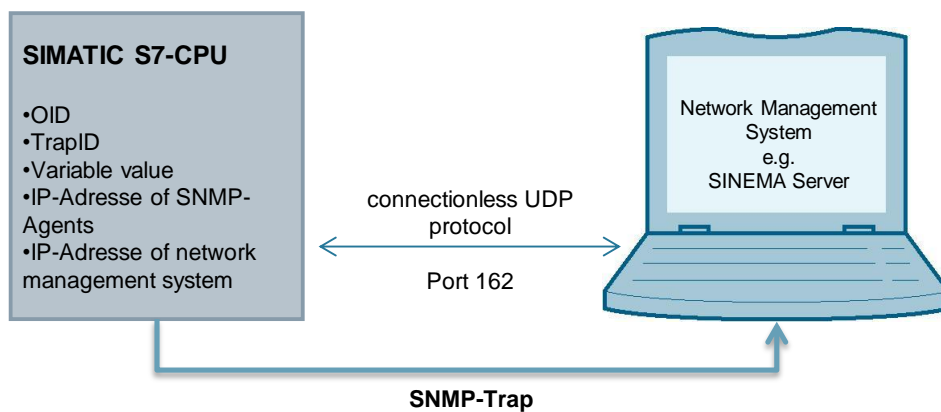
Process for sending traps to the network management system

When sending an SNMP trap, the S7 CPU sends the prepared telegram to the network management system via the connectionless UDP protocol. The telegram contains:

- the IP address of the device that triggers the trap
- the IP address of the network management system
- the TRAP ID
- the OID
- the variable value

After sending the telegram, the UDP end point will be switched off.

Figure 1-5



1.3 Hardware and software requirements

Requirements for this library

In order to use the functionality of the library described here, the following hardware and software requirements must be met:

Hardware

The following requirements for the hardware must be met:

Table 1-3

No.	Components	Order number
1	SIMATIC CPU	CPU15xx CPU12xx (version 4 and higher) CPU S7-300/S7-400/ET 200S-CPU <ul style="list-style-type: none"> • with integrated PROFINET interface • Supports UDP protocol • Supports open communication blocks (T-blocks)
2	Network components that are SNMP-capable	e.g. SCALANCE M-800

Software

Table 1-4

Components	Qty.	Item number
STEP 7 Professional V16	1	6ES7823-1AE06-0YA5

2 Function blocks of the library "LSnmp"

Introduction

This section explains the most blocks in the library "LSnmp."

The "LSnmp" library contains the blocks and the necessary PLC data types for SNMP communication between an S7 CPU and the network components.

The library contains two folders with the respective types:

- **S7-1xxx**
This folder contains the SNMP blocks and the necessary PLC data types for application with an a S7-1500/S7-1200 CPU:
 - "LSnmp_Get"
 - "LSnmp_GetNext"
 - "LSnmp_GetBulk"
 - "LSnmp_Set"
 - "LSnmp_SwitchIO"
 - "LSnmp_SendTrap"
 - "LSnmp_typeParamGetSet"
 - "LSnmp_typeParamGetBulk"
 - "LSnmp_typeBulkResponseData"
 - "LSnmp_typeParamSwitchIO"
 - "LSnmp_typeParamSendTrap"
- **S7-xxx**
This folder contains the SNMP blocks and the necessary PLC data types for application with an a S7-300/S7-400 CPU:
 - "LSnmp_Get"
 - "LSnmp_GetNext"
 - "LSnmp_GetBulk"
 - "LSnmp_Set"
 - "LSnmp_SwitchIO"
 - "LSnmp_SendTrap"
 - "LSnmp_typeParamGetSet"
 - "LSnmp_typeParamGetBulk"
 - "LSnmp_typeBulkResponseData"
 - "LSnmp_typeParamSwitchIO"
 - "LSnmp_typeParamSendTrap"
 - "LSnmp_typeTrapInfo"

The contents of the folder are identical; only the program code for the respective type of control system is optimized.

Note The SNMP blocks must be activated in cycles. This can be done in the OB1. If you activate multiple SNMP blocks in OB1, you must select the respective different connection IDs (value range: W#16#0001 to W#16#0FFF).

Note If using an S7-300/400, the value of an OID element is limited to a maximum of 16383.

2.1 The library block “LSnmp_Get”

Overview

The “LSnmp_Get” block is a programmable function block for reading SNMP variables. All information required by this block is stored in a global data block in a variable of the type “LSnmp_typeParamGetSet” (see [section 2.7](#)) and delivered to the block as an input/output parameter.

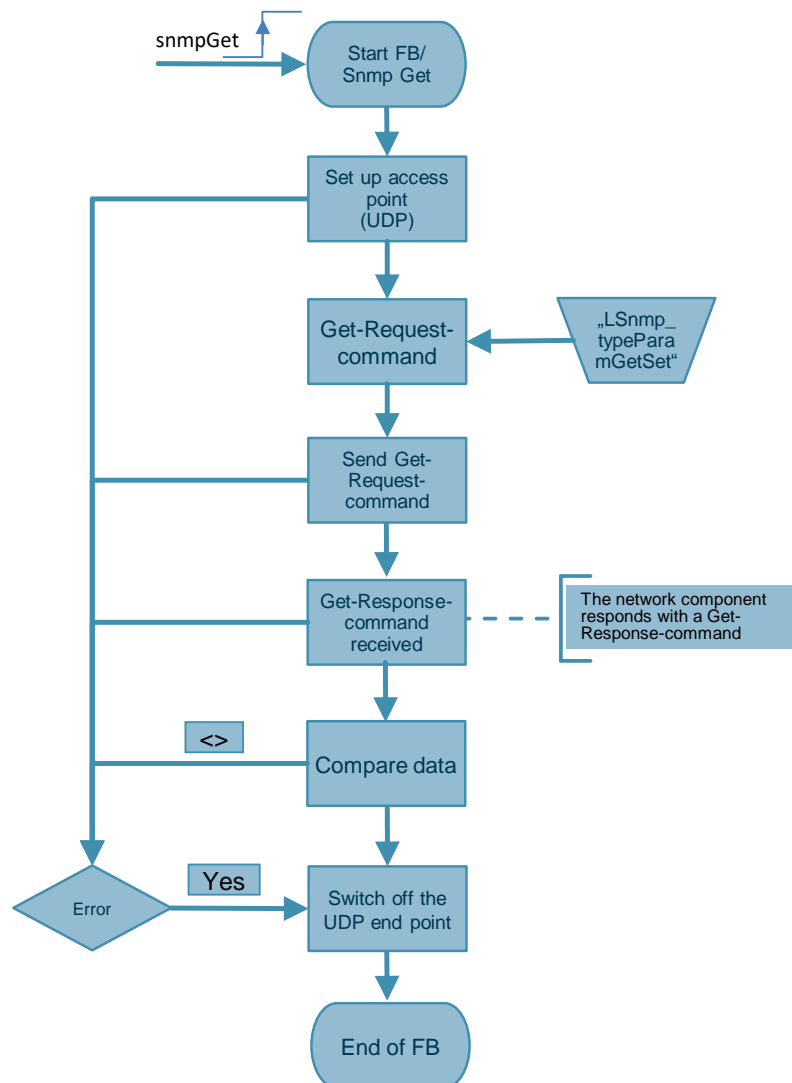
2.1.1 Functionality

The block “LSnmp_Get” sends an SNMP Get request command to the SNMP agent in the network component. The network component responds with an SNMP Get response command that contains the requested data or an error message.

2.1.2 Program flow chart

The following chart shows the process for reading an SNMP variable.

Figure 2-1



2.1.3 Call structure and parameter interface

The following figure shows the call interface of the library block "LSnmp_Get."

Figure 2-2

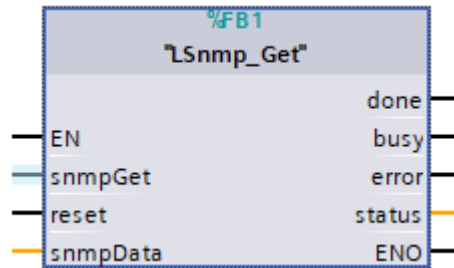


Table 2-1

	Parameters	Data type	Description
Input	EN	BOOL	Enable input Only in FBD and KOP view.
	snmpGet	BOOL	In the event of positive edge, the variable is read
	reset	BOOL	In the event of positive edge, the connection resources are enabled
InOut	snmpData	"LSnmp_typeParamGetSet"	Specifying a data structure; <ul style="list-style-type: none"> Transfer of all relevant information for setting up the UDP access point and for the SNMP variable Storage space for the response data of the network component (e.g. return value of the variable, etc.) (See Section 2.7)

2 Function blocks of the library "LSnmp"

Output	done	BOOL	TRUE, if the last request has been processed without errors FALSE: by falling edge at "snmpGet" or positive edge at "reset"
	busy	BOOL	TRUE, if the block "LSnmp_Get" is active Adopts the FALSE state as soon as the process is completed or an error occurs
	error	BOOL	TRUE if an error occurs during the execution of the routine FALSE: by falling edge at "snmpGet" or a positive edge at "reset"
	status	DWORD	Status, if ERROR=TRUE. By falling edge at "snmpGet" or positive edge at "reset," the pending error that must be corrected by the user can be acknowledged
	ENO	BOOL	Enable output Only relevant in FDP and LAD representation

2.2 The library block "LSnmp_GetNext"

Overview

The block "LSnmp_GetNext" essentially works like the block "LSnmp_Get," except that it allows successive SNMP variables within an SNMP object tree to be read.

All information required by this block is stored in a global data block in a variable of the type "LSnmp_typeParamGetSet" (see [section 2.7](#)) and delivered to the block as an input/output parameter.

2.2.1 Functionality

The block "LSnmp_GetNext" uses the SNMP GetNext request command and makes it easier to query successive objects in an MIB subtree. The objects within an MIB subtree are characterized by the fact that all objects have the same master OID and differ only in the suffix.

The GetNext command is primarily used for running through a table or table column and functions as follows.

The first GetNext call goes to the master OID. The SNMP agent does not respond like the Get request command with the return values of the requested OID (here: the master OID), but with the OID (master OID + suffix₁) and the return value (value₁) of the following object. The next GetNext call will now be set at the received OID ((master OID + suffix₁). The response to this is the successive OID (master OID +suffix₂) and its return value (value₂).

The MIB subtree is run through automatically until the end of the subtree is reached.

In terms of programming technology, it must be implemented via a while loop that runs until the master OID of the SNMP variable changes.

2.2.2 Call structure and parameter interface

The following figure shows the call interface of the library block "LSnmp_GetNext." Figure 2-3

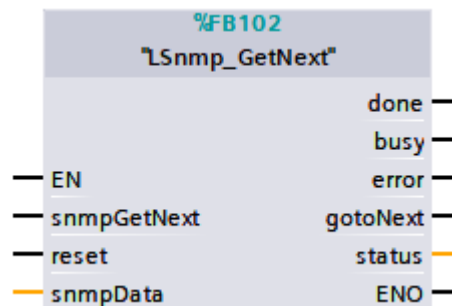


Table 2-2

Parameters		Data type	Description
Input	EN	BOOL	Enable input Only relevant in FBD and LAD view
	snmpGetNext	BOOL	In the event of positive edge, the variable is read
	reset	BOOL	In the event of positive edge, the connection resources are enabled
InOut	snmpData	"LSnmp_typeParamGetSet"	Specifying a data structure; <ul style="list-style-type: none"> Transfer of all relevant information for setting up the UDP access point and for the SNMP variable Storage space for the response data of the network component (e.g. return value of the variables, etc.) (See Section 2.7).
Output	done	BOOL	TRUE, if the last request has been processed without errors FALSE: by falling edge at "snmpGetNext" or positive edge at "reset"
	busy	BOOL	TRUE, if the block "LSnmp_GetNext" is active Adopts the FALSE state as soon as the process is completed or an error occurs
	error	BOOL	TRUE if an error occurs during the execution of the routine. FALSE: by falling edge at "snmpGetNext" or a positive edge at "reset"
	gotoNext	BOOL	If positive edge, signals another loop pass and with it, the reading of the subsequent SNMP object Note: Use this edge to transfer the currently stored response data to another storage area. Otherwise it will be overwritten by the next query
	status	DWORD	Status, if ERROR=TRUE By falling edge at "snmpGetNext" or positive edge at "reset," the pending error that must be corrected by the user can be acknowledged
	ENO	BOOL	Enable output Only relevant in FDP and LAD representation

2.3 The library block "LSnmp_GetBulk"

Overview

The block "LSnmp_GetBulk" basically operates like the Block "LSnmp_GetNext." It serves to minimize the network transfer, as it allows large volumes of data to be read out efficiently with only one response telegram. All information required by this block is stored in a global data block in a variable of the type "typeParamGetBulk" (see [section 2.8](#)) and delivered to the block as an input/output parameter.

2.3.1 Functionality

The block "LSnmp_GetBulk" uses the SNMP GetBulk request command and requires SNMPv2 for this - an extension of SNMPv1. The GetBulk command internally runs multiple GetNext queries and returns the result in a single response telegram. The number of GetNext commands can be given in the GetBulk telegram by means of a programmable repetition factor. The return values of all queried objects are compiled in a single response telegram and transferred. Consequently, the task of the block "LSnmp_GetBulk" is to structure the data stream received back into the return values of the individual objects and provide it in split form on the output parameter.

2.3.2 Call structure and parameter interface

The following figure shows the call interface of the library block "LSnmp_GetBulk."
Figure 2-4

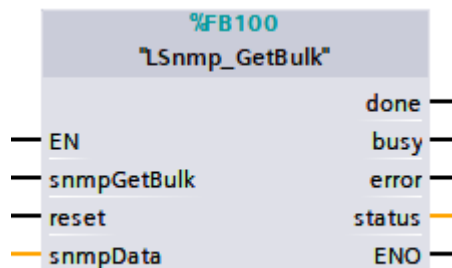


Table 2-3

Parameters		Data type	Description
Input	EN	BOOL	Enable input Only relevant in FBD and LAD view
	snmpGetBulk	BOOL	In the event of positive edge, the variable is read
	reset	BOOL	In the event of positive edge, the connection resources are enabled
InOut	snmpData	"LSnmp_typeParamGetBulk"	Specifying a data structure; <ul style="list-style-type: none"> Transfer of all relevant information for setting up the UDP access point and for the SNMP variable Storage space for the response data of the network component (e.g. return value of the variables, etc.) (See section 2.8)
Output	done	BOOL	TRUE, if the last request has been processed without errors. FALSE: by falling edge at "snmpGetBulk" or positive edge at "reset"
	busy	BOOL	TRUE, if the block "LSnmp_GetBulk" is active. Adopts the FALSE state as soon as the process is completed or an error occurs
	error	BOOL	TRUE if an error occurs during the execution of the routine FALSE: by falling edge at "snmpGetBulk" or a positive edge at "reset"
	status	DWORD	Status, if ERROR=TRUE By falling edge at "snmpGetBulk" or positive edge at "reset," the pending error that must be corrected by the user can be acknowledged
	ENO	BOOL	Enable output Only relevant in FDP and LAD representation

2.4 The library block "LSnmp_Set"

Overview

The "LSnmp_Set" block is a programmable function block for writing SNMP variables. All information required by this block is stored in a global data block in a variable of the type "LSnmp_typeParamGetSet" (see [section 2.7](#)) and delivered to the block as an input/output parameter.

2.4.1 Functionality

The block "LSnmp_Set" sets a Write job via the SNMP Set request command for an SNMP variable to the SNMP agent in the network component.

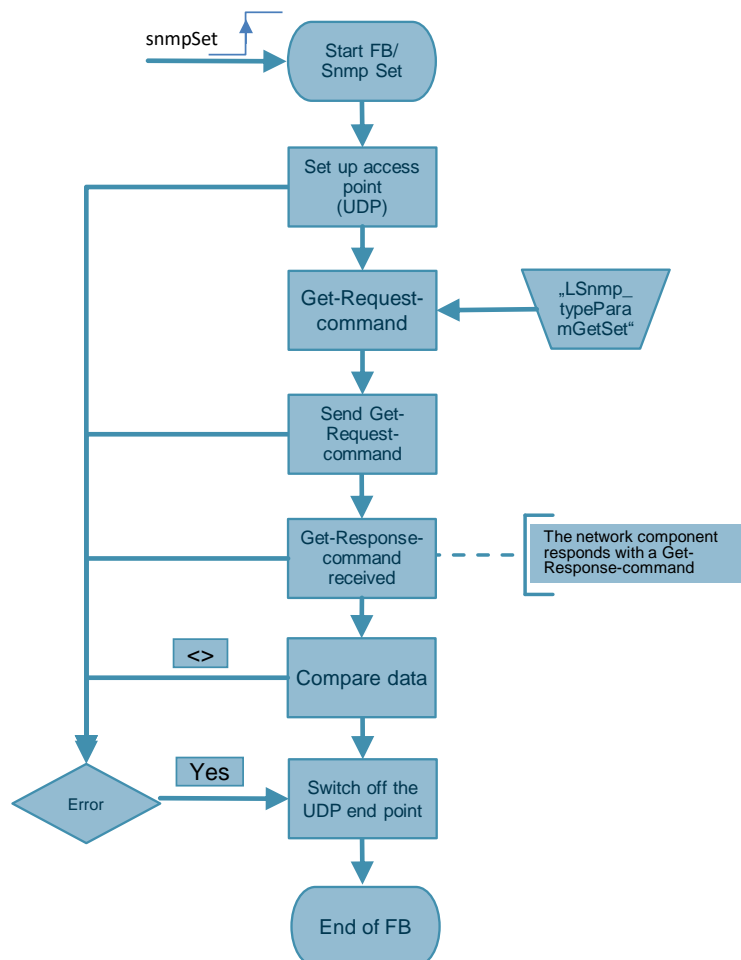
In the set response telegram, the block contains the result of the Write job from the network component.

If the variable read in does not match the written value, an error will be read out. The Write job must be set again.

2.4.2 Program flow chart

The following chart shows the process for writing an SNMP variable.

Figure 2-5



2.4.3 Call structure and parameter interface

The following figure shows the call interface of the library block "LSnmp_Set."

Figure 2-6

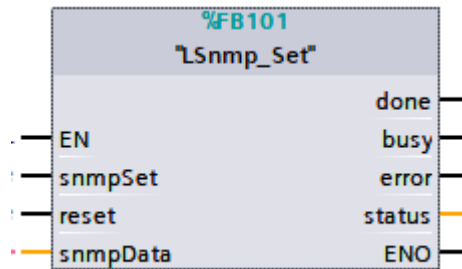


Table 2-4

Parameters		Data type	Description
Input	EN	BOOL	Enable input Only relevant in FBD and LAD view
	snmpSet	BOOL	In the event of positive edge, the variable is written
	reset	BOOL	In the event of positive edge, the connection resources are enabled
InOut	snmpData	"LSnmp_typeParamGetSet"	Specifying a data structure; <ul style="list-style-type: none"> Transfer of all relevant information for setting up the UDP access point and for the SNMP variable Transfer of information for the SNMP variables to be written (See section 2.7)

2 Function blocks of the library "LSnmp"

Output	done	BOOL	TRUE, if the last request has been processed without errors FALSE: by falling edge at "snmpSet" or positive edge at "reset"
	busy	BOOL	TRUE, if the block "LSnmp_Set" is active Adopts the FALSE state as soon as the process is completed or an error occurs
	error	BOOL	TRUE if an error occurs during the execution of the routine FALSE: by falling edge at "snmpSet" or a positive edge at "reset"
	status	DWORD	Status, if ERROR=TRUE By falling edge at "snmpSet" or positive edge at "reset," the pending error that must be corrected by the user can be acknowledged
	ENO	BOOL	Enable output Only relevant in FDP and LAD representation

2.5 The library block "LSnmp_SwitchIO"

Overview

The block "LSnmp_SwitchIO" is a block programmed in SCL to demonstrate an application of the blocks "LSnmp_Get" and "LSnmp_Set."

The user block "LSnmp_SwitchIO" allows a SIMATIC S7 CPU to switch and read the digital output of SCALANCE W devices via SNMP.

All information required by this block is stored in a global data block in a variable of the type "LSnmp_typeParamSwitchIO" (see [section 2.10](#)) and delivered to the block as an input parameter.

2.5.1 Functionality

The block "LSnmp_SwitchIO" implements two functions:

- Queries the status of the digital output of a SCALANCE W device
- Switches the digital output of a SCALANCE W device

To demonstrate these scenarios, the "LSnmp_SwitchIO" internally calls up the function blocks "LSnmp_SnmpGet" and "LSnmp_SnmpSet."

Queries the status of the digital output of a SCALANCE W device

To query the status of the digital output, an SNMP Get Request command is sent to the IWLAN component with the block "LSnmp_SnmpGet" after a trigger command.

This responds to the query with a Get Response telegram that contains the requested status.

Switches the digital output of a SCALANCE W device

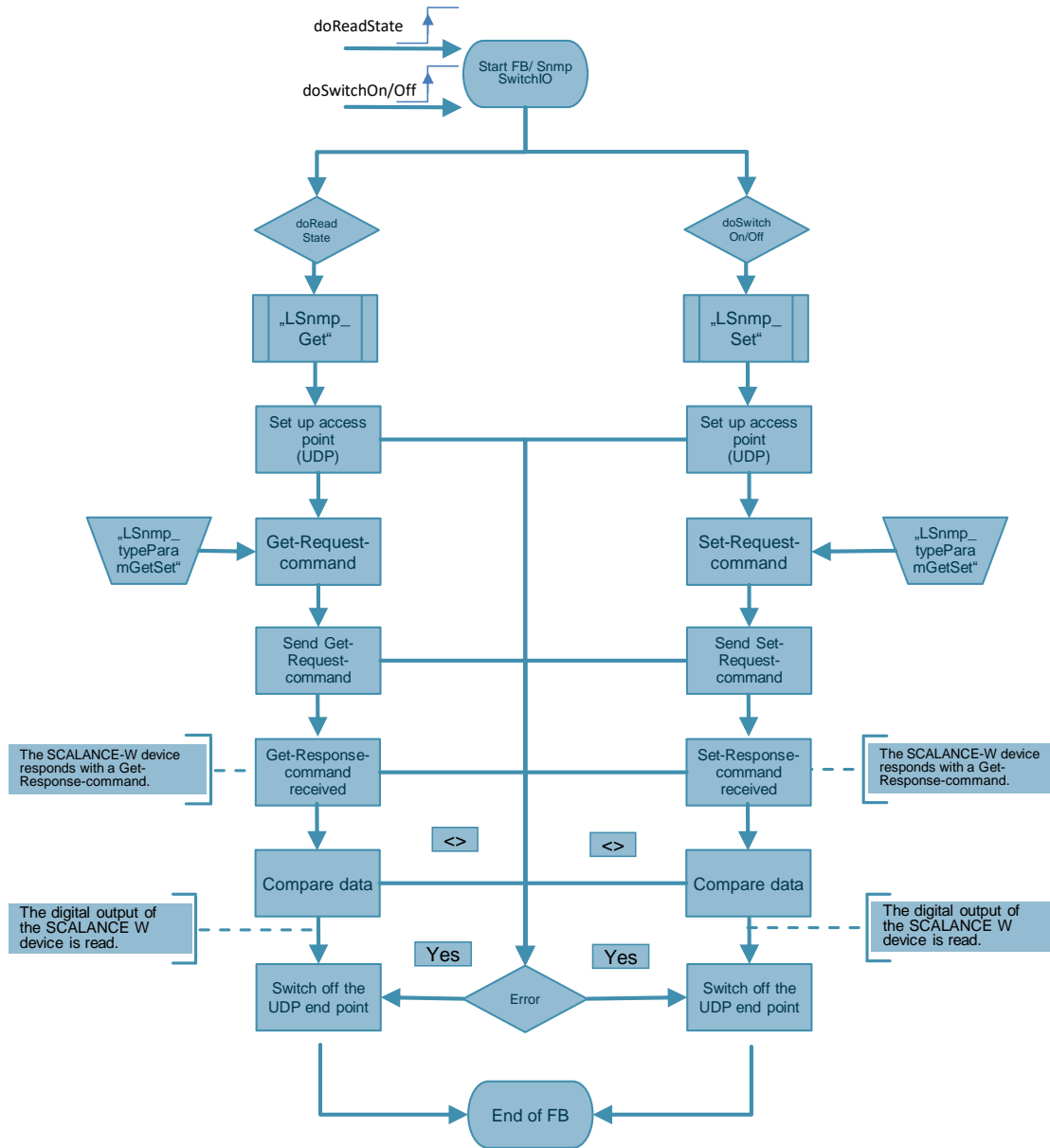
To switch the digital output, after a trigger command, an SNMP Set packet is set up that contains the requested value (set output, reset output). The Write Job is set on the IWLAN component via the block "LSnmp_SnmpSet."

This responds to the query with a Set Response telegram that contains the result of the Write job. If the variable read in matches the written value, the switching process was successful.

2.5.2 Program flow chart

The following chart shows the process for reading an SNMP variable.

Figure 2-7



2.5.3 Call structure and parameter interface

The following figure shows the call interface of the library block "LSnmp_SwitchIO."
Figure 2-8

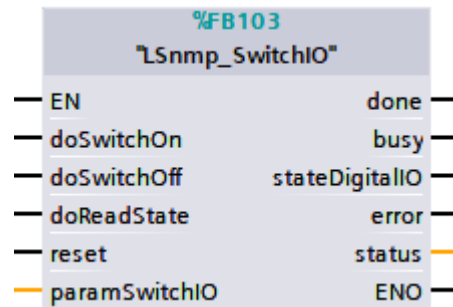


Table 2-5

	Parameters	Data type	Description
Input	EN	BOOL	Enable input Only relevant in FBD and LAD view
	doSwitchOn	BOOL	In the event of positive edge, the digital output is set
	doSwitchOff	BOOL	In the event of positive edge, the digital output is reset
	doReadState	BOOL	In the event of positive edge, the status of the digital output is read
	reset	BOOL	In the event of positive edge, the connection resources are enabled
InOut	snmpData	"LSnmp_typeParamSwitchIO"	Specifying a data structure; Transfer of all relevant information for setting up the UDP access point and for the SNMP variable (See section 2.10)
Output	done	BOOL	TRUE, if the last request has been processed without errors FALSE: <ul style="list-style-type: none"> By falling edge at "doSwitchOn," if "doSwitchOn" was the last action By falling edge at "doSwitchOff," if "doSwitchOff" was the last action By falling edge at "doReadState," if "doReadState" was the last action By falling edge at "reset"

2 Function blocks of the library "LSnmp"

	busy	BOOL	TRUE, if the block "LSnmp_SwitchIO" is active Adopts the FALSE state as soon as the process is completed or an error occurs
	stateDigitalIO	BOOL	Status of digital output of last action: TRUE: digital output "ON" FALSE: digital output "OFF"
	error	BOOL	TRUE if an error occurs during the execution of the routine FALSE: <ul style="list-style-type: none"> • By falling edge at "doSwitchOn," if "doSwitchOn" was the last action • By falling edge at "doSwitchOff," if "doSwitchOff" was the last action • By falling edge at "doReadState," if "doReadState" was the last action • By falling edge at "reset"
	status	DWORD	Status, if ERROR=TRUE <ul style="list-style-type: none"> • By falling edge at "doSwitchOn," if "doSwitchOn" was the last action • By falling edge at "doSwitchOff," if "doSwitchOff" was the last action • By falling edge at "doReadState," if "doReadState" was the last action • By falling edge at "reset" the pending error that must be corrected by the user can be acknowledged
	ENO	BOOL	Enable output Only relevant in FDP and LAD representation

2.6 The library block "LSnmp_SendTrap"

Overview

SNMP agents (e.g. an S7 CPU) send messages to the network management system via SNMP traps to display changes in the network status.

The "LSnmp_SendTrap" block is a programmable function block for sending SNMP trap telegrams. All information required by this block is stored in a global data block in a variable of the type "LSnmp_typeParamSendTrap" (see [section 2.11](#) for the S7-1500/S7-1200 CPU and [section 2.12](#) for the S7-300/S7-400 CPU) and delivered to the block as an input/output parameter.

2.6.1 Functionality

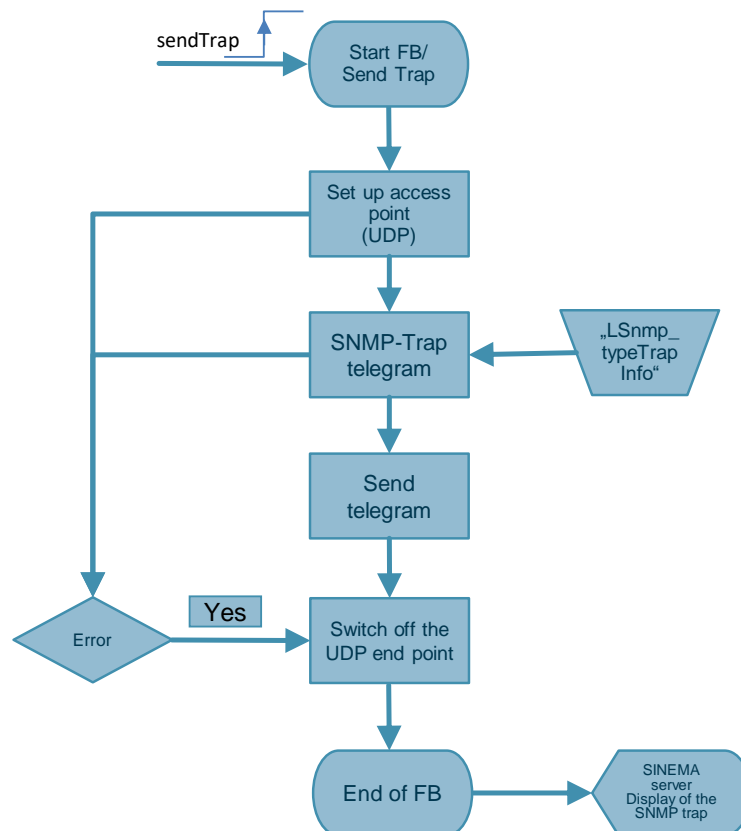
The "LSnmp_SendTrap" block implements the following functions:

1. Setting up the UDP access point for communication between the SNMP Manager and the SNMP agent
2. Processing the input information in the structure "LSnmp_typeTrapInfo" into an SNMP trap telegram
3. Sending the SNMP trap telegram
4. Switching off the UDP end point

2.6.2 Program flow chart

The following chart shows the process for sending an SNMP trap.

Figure 2-9



Note The SNMP agents do not receive acknowledgment for sent traps.

Note To prevent heavy network load, traps should only be sent as needed.

2.6.3 Call structure and parameter interface

The following figure shows the call interface of the library block "LSnmp_SendTrap."

Figure 2-10

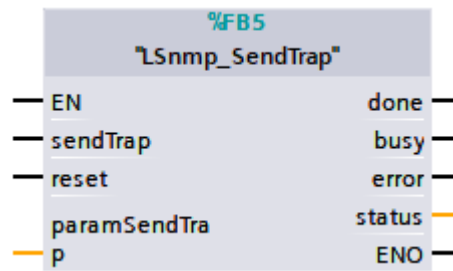


Table 2-6

	Parameters	Data type	Description
Input	EN	BOOL	Enable input Only relevant in FBD and LAD view
	snmpSendTrap	BOOL	In the event of positive edge, the trap telegram is sent
	reset	BOOL	In the event of positive edge, the connection resources are enabled
InOut	paramSendTrap	"LSnmp_typeParamSendTrap"	Contains all information for the SNMP telegram packet. <ul style="list-style-type: none"> • S7-1500/S7-1200 CPU (see section 2.11) • S7-300/S7-400 CPU (see section 2.12)

2 Function blocks of the library "LSnmp"

Output	done	BOOL	TRUE, if the last request has been processed without errors. FALSE: by falling edge at "snmpSendTrap" or positive edge at "reset"
	busy	BOOL	TRUE, if the block "LSnmp_SendTrap" is active Adopts the FALSE state as soon as the process is completed or an error occurs
	error	BOOL	TRUE if an error occurs during the execution of the routine FALSE: by falling edge at "snmpSendTrap" or a positive edge at "reset"
	status	DWORD	Status, if ERROR=TRUE By falling edge at "snmpSendTrap" or positive edge at "reset," the pending error that must be corrected by the user can be acknowledged
	ENO	BOOL	Enable output Only relevant in FDP and LAD representation

2.7 The PLC data type "LSnmp_typeParamGetSet"

Overview

The PLC data type "LSnmp_typeParamGetSet" is a defined data structure used multiple times in the program.

This serves as a template for creating global data blocks for transferring all relevant information to the function blocks "LSnmp_Get," "LSnmp_GetNext," and "LSnmp_Set."

The structure of the data type PLC consists of several components.

Figure 2-11

LSnmp_typeParamGetSet		
	Name	Data type
1	ipAddress	DWord
2	hwIdentifier	HW_ANY
3	connectionID	Word
4	localPort	Word
5	watchDogTime	Time
6	oID	String[254]
7	community	String[20]
8	returnValueType	Byte
9	returnValueLenght	Byte
10	returnValue	Array[1..255] of Byte

Configuration data for setting up the UDP access point

The blocks "LSnmp_Get," "LSnmp_GetNext," and "LSnmp_Set" require special information for setting up the UDP access point for communication with the network component. The following variables in the PLC data type are responsible for this:

Table 2-7

Parameters	Description
ipAddress	IP address of the network components; The IP address must be coded as a hexadecimal value, e.g. 16#AC = 172.
hwIdentifier	Hardware identification of the PROFINET interface for the S7-1500/S7-1200 CPU; Note: S7-300/S7-400 CPUs deviceID >> is the local_device_id of the S7-300/S7-400 CPU (see 4)
connectionID	The SNMP block connection ID required for setting up the UDP access point; Note: If you want to plan further open communication connections in addition to the UDP connection, you must select the respective different connection IDs (value range: W#16#0001 to W#16#0FFF).
localPort	The local port number of the UDP connection; If you want to plan further open communication connections in addition to the UDP connection, you must select the respective different local port numbers.
watchDogTime	Monitoring time of the processes; Default value:=4 s

SNMP variable parameters

To read ("LSnmp_Get", "LSnmp_GetNext") or write ("snmpSet") SNMP variables, registration information on the SNMP variables is needed. The following variables in the PLC data type are responsible for this.

Table 2-8

Parameters	Description
oid	Object identifier of the SNMP variable to be retrieved in SNMP format (for example, 1.3.6.1.2.1.1.4.0); The OID object can be found in the general (RFC1213: MIB II) or in the private MIB file of the device (see 3)
community	In most cases, "public" is chosen as the community name for read access and "private" for write access. This value must match the community name chosen in the project planning for the network component (see section 3).
returnValueType	Data type of SNMP variable: 02: Integer, 04: String, 41: Counter, 43: Timeticks During the Read access ("LSnmp_Get", "LSnmp_GetNext"), values of SNMP variable type are automatically determined and entered here. For the Write access ("LSnmp_Set"), the type of SNMP variable must be programmed.
returnValueLength	Length of the SNMP variable; During the Read access ("LSnmp_Get", "LSnmp_GetNext"), the length of the SNMP variable is automatically determined and entered here. For the Write access ("LSnmp_Set"), the length of the SNMP variable must be programmed.
returnValue	ARRAY OF BYTE: The array length is limited to 255 bytes. During the Read access ("LSnmp_Get", "LSnmp_GetNext"), the response data of the SNMP variable is entered here. For the Write access ("LSnmp_Set"), the data with which the SNMP variable should be described must be entered here.

2.8 The PLC data type "LSnmp_typeParamGetBulk"

Overview

The PLC data type "LSnmp_typeParamGetBulk" is a defined data structure that serves as a template for creating global data blocks for transferring all relevant information to the function block "LSnmp_GetBulk."

The structure of the data type PLC consists of several components.

Figure 2-12

LSnmp_typeParamGetBulk			
		Name	Data type
1		ipAddress	DWord
2		hwIdentifier	HW_ANY
3		connectionID	Word
4		localPort	Word
5		watchDogTime	Time
6		oid	String[254]
7		community	String[20]
8		maxRepetitions	Byte
9		returnValue	Array[1..4] of "LSnmp_typeBulkResponseData"

Configuration data for setting up the UDP access point

The block "LSnmp_GetBulk" requires special information for setting up the UDP access point for communication with the network component. The following variables in the PLC data type are responsible for this.

Table 2-9

Parameters	Description
ipAddress	IP address of the network components; The IP address must be coded as a hexadecimal value, e.g. 16#AC = 172.
hwIdentifier	Hardware identification of the PROFINET interface for the S7-1500/S7-1200 CPU; Note: S7-300/S7-400 CPUs deviceID: is the local_device_id of the S7-300/S7-400 CPU (see 4).
connectionID	The SNMP block connection ID required for setting up the UDP access point; Note: If you want to plan further open communication connections in addition to the UDP connection, you must select the respective different connection IDs (value range: W#16#0001 to W#16#0FFF).
localPort	The local port number of the UDP connection; If you want to plan further open communication connections in addition to the UDP connection, you must select the respective different local port numbers.
watchDogTime	Monitoring time of the processes; Default value: 4 s

SNMP variable parameters

To read (“LSnmp_GetBulk”) SNMP variables, information on the SNMP variables is needed. The following variables in the PLC data type are responsible for this.

Table 2-10

Parameters	Description
oid	Object identifier of the SNMP variable to be retrieved in SNMP format (for example, 1.3.6.1.2.1.1.4.0); The OID object can be found in the general (RFC1213: MIB II) or in the private MIB file of the device (see 3)
community	In most cases, “public” is chosen as the community name for read access and “private” for write access. This value must match the community name chosen in the project planning for the network component (see section 3)
maxRepetitions	Information on how many consecutive objects in an OID tree should be read.
returnValue	This array is of the data type "LSnmp_typeBulkResponseData" (see section 2.9) and for each read object includes a field for storing the return information such as data type, length, and value. The size of the array must match the parameter “maxRepetitions.”

2.9 The PLC data type “LSnmp_typeBulkResponseData”

To read SNMP variables, information on the SNMP variables is needed. In addition to the parameters already know from [section 2.8](#), the block “LSnmp_GetBulk” requires a return range as an array of the data type “LSnmp_typeBulkResponseData.”

Figure 2-13

LSnmp_typeBulkResponseData	
Name	Data type
1 returnValueType	Byte
2 returnValueLenght	Byte
3 returnValue	Array[1..255] of Byte

Table 2-11

Parameters	Description
returnValueType	Data type of SNMP variable; 02: Integer, 04: String, 41: Counter, 43: Timeticks During the Read access (“LSnmp_GetBulk”), values of SNMP variable type are automatically determined and entered here.
returnValueLength	Length of the SNMP variable; During the Read access (“LSnmp_GetBulk”), the length of the SNMP variable is automatically determined and entered here.
returnValue	ARRAY OF BYTE: The array length is limited to 255 bytes During the Read access (“LSnmp_GetBulk”), the response data of the SNMP variable is entered here.





2.10 The PLC data type "LSnmp_typeParamSwitchIO"

Overview

The PLC data type "LSnmp_typeParamSwitchIO" is a defined data structure that serves as a template for creating global data blocks for transferring all relevant information to the function block "LSnmp_SwitchIO."

The structure of the data type PLC consists of several components.

Figure 2-14

LSnmp_typeParamSwitchIO			
		Name	Data type
1		ipAddress	DWord
2		connectionGetID	Word
3		connectionSetID	Word
4		hwIdentifier	HW_ANY

Configuration data for setting up the UDP access point

The blocks "LSnmp_Get" and "LSnmp_Set" accessed internally require special information for setting up the UDP access point for communication with the network component. The following variables in the PLC data type are responsible for this and are transferred to the data structure "LSnmp_typeParamSwitchIO" during run time.

Table 2-12

Parameters	Description
ipAddress	IP address of the network components; The IP address must be coded as a hexadecimal value, e.g. 16#AC = 172.
connectionGetID	The connection ID required for setting up the UDP access point for the SNMP block "LSnmp_Get" accessed internally; Note: If you want to plan further open communication connections in addition to the UDP connection, you must select the respective different connection IDs (value range: W#16#0001 to W#16#0FFF).
connectionSetID	The connection ID required for setting up the UDP access point for the SNMP block "LSnmp_Set" accessed internally; Note: If you want to plan further open communication connections in addition to the UDP connection, you must select the respective different connection IDs (value range: W#16#0001 to W#16#0FFF). The connectionSetID and connectionGetID must be different.
hwIdentifier	Hardware identification of the PROFINET interface of the S7-1500/S7-1200 CPU; Note: S7-300/S7-400 CPUs deviceID >> is the local_device_id of the S7-300/S7-400 CPU (see 4)

2.11 The PLC data type "LSnmp_typeParamSendTrap" S7-1500/S7-1200 CPU

Overview

The PLC data type "LSnmp_typeParamSendTrap" is a defined data structure that serves as a template for creating global data blocks for transferring all relevant information to the function block "LSnmp_SendTrap" for an S7-1500/S7-1200 CPU. The structure of the data type PLC consists of several components.

Figure 2-15

LSnmp_typeParamSendTrap			
	Name	Data type	
1	hwIdentifier	HW_ANY	
2	connectionID	CONN_OUC	
3	ipManager	DWord	
4	ipAgent	Array[0..3] of Byte	
5	trapID	String[125]	
6	specificID	Int	
7	oidBindingVar	String[125]	
8	description	String[125]	
9	telegram	Array[0..400] of Byte	

Configuration data for setting up the UDP access point

The block "LSnmp_SendTrap" requires special information for setting up the UDP access point for communication with the network component. The following variables in the PLC data type are responsible for this.

Table 2-13

Parameters	Description
hwIdentifier	Hardware identification of the PROFINET interface for the S7-1500/S7-1200 CPU; Note: S7-300/S7-400 CPUs deviceID: local_device_id of the S7-300/S7-400 CPU (see V4)
connectionID	The SNMP block connection ID required for setting up the UDP access point; Note: If you want to plan further open communication connections in addition to the UDP connection, you must select the respective different connection IDs (value range: W#16#0001 to W#16#0FFF).
ipManager	IP Address of the network management system (e.g. SINEC NMS); The IP address must be coded as a hexadecimal value, e.g. 16#AC = 172.
ipAgent	IP address of the SNMP agent (the S7 CPU); The IP address must be coded as a hexadecimal value, e.g. 16#AC = 172.

SNMP variable parameters

Information on the SNMP traps is required to send SNMP traps. The following variables in the PLC data type are responsible for this.

Table 2-14

Parameters	Description
trapID	The trap ID of the trap; Syntax and structure identical to OID (e.g. '1.2.3.6.1.4.2.4196')
specificID	The company-specific ID of the trap, e.g.: <ul style="list-style-type: none"> Incoming alarm: specific TRAP = 67 Outgoing alarm: specific TRAP = 68
oidBindingVar	The OID of the connected variables (syntax: '1.2.3...'); Note: With traps it is also possible that no variable binding is sent with it ('0.0'). In this case, the trap ID is considered sufficient information.
Description	Text of the connected variables
Telegram	An array for mounting the SNMP trap telegram

Note

Multiple OID variables can be sent with the trap for description or with updated values, etc., under "Variable Binding." The function block "LSnmp_SendTrap" sends an additional string variable "description" to describe each trap.

2.12 The PLC data type "LSnmp_typeParamSendTrap" S7-300/S7-400 CPU

The PLC data type "LSnmp_typeParamSendTrap" is a defined data structure that serves as a template for creating global data blocks for transferring all relevant information to the function block "LSnmp_SendTrap" for an S7-300/S7-400 CPU. The structure of the data type PLC consists of several components.

Figure 2-16

LSnmp_typeParamSendTrap		
	Name	Data type
1	deviceID	Byte
2	connectionID	Word
3	ipManager	DWord
4	trapInfo	"LSnmp_typeTrapInfo"

The block "LSnmp_SendTrap" requires special information for setting up the UDP access point for communication with the network component and to send an SNMP trap. The following variables in the PLC data type are responsible for this.

Table 2-15

Parameters	Description
deviceID	The local_device_id of the S7-300/S7-400 CPU (see 4)
connectionID	The SNMP block connection ID required for setting up the UDP access point; Note: If you want to plan further open communication connections in addition to the UDP connection, you must select the respective different connection IDs (value range: W#16#0001 to W#16#0FFF)
ipManager	IP Address of the network management system (e.g. SINE NMS); The IP address must be coded as a hexadecimal value, e.g. 16#AC = 172
trapInfo	Structure for all relevant data to send an SNMP trap (see section 2.13)

2.13 The PLC data type "LSnmp_typeTrapInfo" S7-300/S7-400 CPU

Information on the SNMP traps is required to send an SNMP trap. In addition to the parameters already known from [section 2.12](#), the block "LSnmp_SendTrap" for the S7-300/S7-400 CPU requires the structure of the type "LSnmp_typeTrapInfo."

Figure 2-17

LSnmp_typeTrapInfo		
	Name	Data type
1	ipAgent	Array[0..3] of Byte
2	trapID	String[125]
3	specificID	Int
4	oidBindingVar	String[125]
5	description	String[125]
6	telegram	Array[0..400] of Byte

Information on the SNMP traps is required to send SNMP traps. The following variables in the PLC data type "LSnmp_TrapInfo" are responsible for this.

Table 2-16

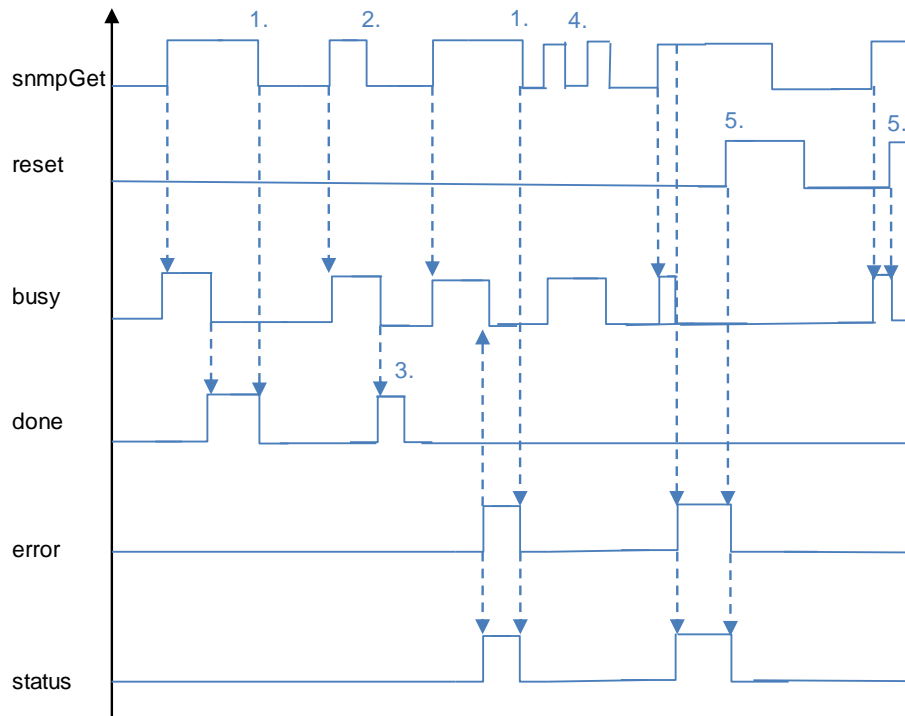
Parameters	Description
ipAgent	IP address of the SNMP agent (the S7 CPU); The IP address must be coded as a hexadecimal value, e.g. 16#AC = 172
trapID	The trap ID of the trap; Syntax and structure identical to OID (e.g. '1.2.3.6.1.4.2.4196')
specificID	The company-specific ID of the trap, e.g.: <ul style="list-style-type: none"> Incoming alarm: 67 Outgoing alarm: 68
oidBindingVar	The OID of the connected variables (syntax: '1.2.3...'); Note: With traps it is also possible that no variable binding is sent with it ('0.0'). In this case, the trap ID is considered sufficient information.
Description	Text of the connected variables
Telegram	An array for mounting the SNMP trap telegram

Note Multiple OID variables can be sent with the trap for description or with updated values, etc., under "Variable Binding." The function block "LSnmp_SendTrap" sends an additional string variable "description" to describe each trap.

2.14 Signal Flow Chart of the library function blocks

The following figure shows the graphic representation of the output parameters of a Get job. This representation is identical for all processes (GetNext, GetBulk, Set, SwitchIO, and SendTrap).

Figure 2-18



1. "done," "error," and "status" are reset with falling edge at "snmpGet."
2. The functionality of the FB is not stopped with falling edge at "snmpGet."
3. If "snmpGet" is already FALSE, then "done," "error," and "status" will only be due for one cycle.
4. A new job is triggered with a rising edge at "snmpGet" while the block is still in processing (busy = TRUE). The old job is finished with the parameters due for the job start. The new job is ignored.
5. The processing of a job is interrupted with a rising edge at "reset." This will immediately interrupt remaining job processing. All output parameters will be reset. The UDP end point will be switched off.

2.15 Status and error displays

For an error diagnostic, evaluate the output "status" of the function blocks of the library "LSnmp." This will give you statements about logical errors and error messages that could occur during communication between the control system and the network components.

You will find a list of possible error messages in the following table.

Table 2-17

Status	Meaning	Remedy/notes
16#0001xyyy	TCON error messages	A description of communication errors can be found in the STEP 7 online help. <ul style="list-style-type: none"> • Change the incorrect parameters • Start the reset process
16#0010xyyy	TUSEND error messages	
16#0011xyyy	TURCV error messages	
16#0012xyyy	TDISCON error messages	
16#00138102	The SNMP packet is too large to send.	<ul style="list-style-type: none"> • Check and change the size of the packet • Restart the process
16#00138103	OID not supported or the OID is only supported for read access.	<ul style="list-style-type: none"> • Check and change the MIB object (OID) • Restart the process
16#00138104	Incorrect data type or value (only possibly as a response to SET packets)	<ul style="list-style-type: none"> • Check and change the MIB object (OID) or the "returnValueType" • Restart the process
16#00138105	The OID is only supported for read access.	<ul style="list-style-type: none"> • Check and change the MIB object (OID) • Restart the process
16#00138106	unknown generation error	<ul style="list-style-type: none"> • Restart the snmpGet process
16#00008107	The length of the received variable "returnValue" exceeds 255 characters. Only 255 characters will be transmitted.	-
16#00008108	Watchdog timer has expired.	<ul style="list-style-type: none"> • Check communication between the control system and network components • Check and change community string (it must match the remote partner for read access in the project planning) • Check IP address of the remote partner • Restart the process
16#00008109	The full length of the GETResponse packet exceeds 486 characters. Only the first 486 characters will be transmitted.	-

Note Errors with the status 16# **0013**xyyy are protocol errors from SNMP.

Note The blocks in this library support sending and receiving SNMP messages that may not exceed 486 bytes in their full length. The useful data length for strings is limited to 255 bytes.

3 Activating SNMP in network components

For the S7 CPU to exchange SNMP data with a network component (e.g. SALANCE M, SCALACE W), make the following settings via the web-based management page of the network component:

- Enable SNMP access
- Enter community string for read access
- Enter community string for write access

Note

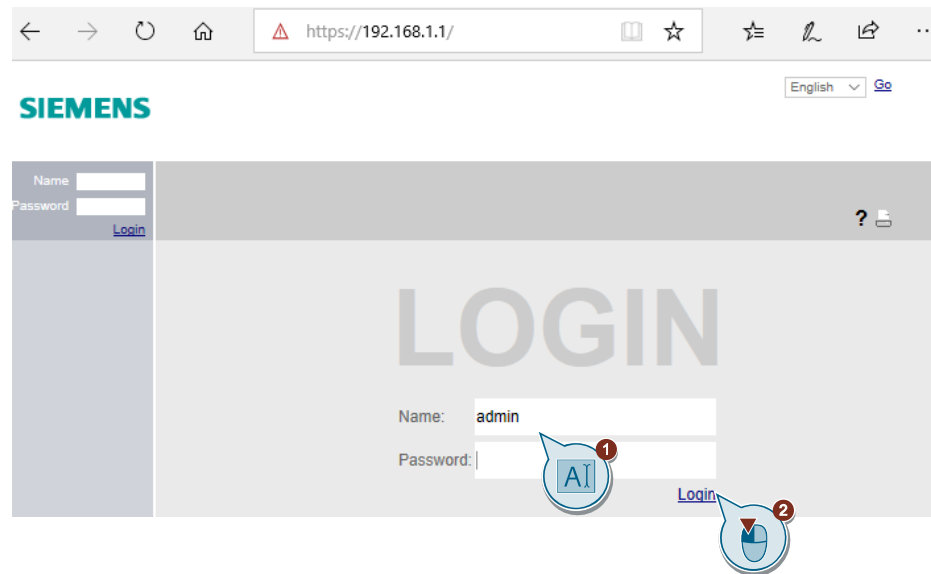
General information about device configuration can be found in the project planning handbook for SCALANCE M devices (see [5](#)).

The SCALANCE devices are configured with Web Based Management:

1. Open an Internet browser and enter the IP address of a SCALANCE module into the address bar.



2. Log on as Administrator.



3. Switch to the menu "System > SNMP > General" ("System > SNMP > General").



4. Enable "SNMPv1/v2c/v3."

Simple Network Management Protocol (SNMP) General

General Traps v3 Groups v3 Users

SNMP: SNMPv1/v2c/v3

SNMPv1/v2c Read Only

5. Enter the community names for read and write access:

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

Simple Network Management Protocol (SNMP) General

General Traps v3 Groups v3 Users

SNMP: SNMPv1/v2c/v3

SNMPv1/v2c Read Only

SNMPv1/v2c Read Community String: public

SNMPv1/v2c Read/Write Community String: private

Note:

These values must match the values entered in the "community" parameters of the PLC data types "LSnmp_typeParamGetSet" and "LSnmp_typeParamGetBulk."

6. Accept the change by clicking "Update" ("Set Values").

Simple Network Management Protocol (SNMP) General

General Traps v3 Groups v3 Users

SNMP: SNMPv1/v2c/v3

SNMPv1/v2c Read Only

SNMPv1/v2c Read Community String: public

SNMPv1/v2c Read/Write Community String: private

SNMPv1 Traps

SNMPv1/v2c Trap Community String: public

SNMPv3 User Migration

SNMP Engine ID: 80.00.10.e9.03.00.1b.1b.80.4e.90

Set Values Refresh

4 Working with the library

This section consists of instructions on how to integrate the library "LSnmp" into your STEP 7 V16 project and instructions on using the library blocks.

Note This description also applies to working with the library "LSnmp" in STEP 7 V15.1.

4.1 Integrating the library in STEP 7

Below are the steps to integrate the library "LSnmp" into your STEP 7 Project. Afterwards you can use the blocks of the library "LSnmp."

Note The existence of a STEP 7 project is assumed in the following.

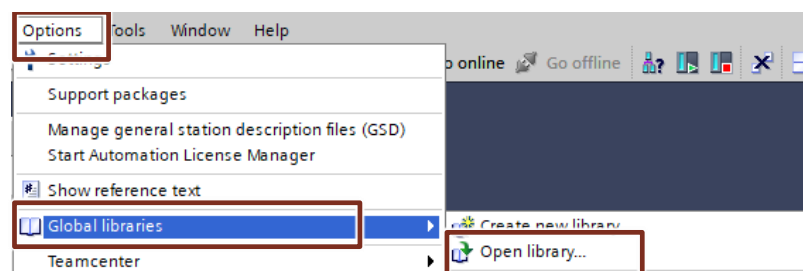
1. The library can be found on the HTML page from which you downloaded this document [2](#). Save the library "57249109_SNMPS7_CPU_LIB_V16_V40.zip" to your hard drive.
2. Unzip the library.

4.2 Integrating the library block "LSnmp_Get" into STEP 7

Note The library blocks "LSnmp_Set," "LSnmp_GetNext," "LSnmp_GetBulk," "LSnmp_SwitchIO," and "LSnmp_SendTrap" are integrated in the same way.

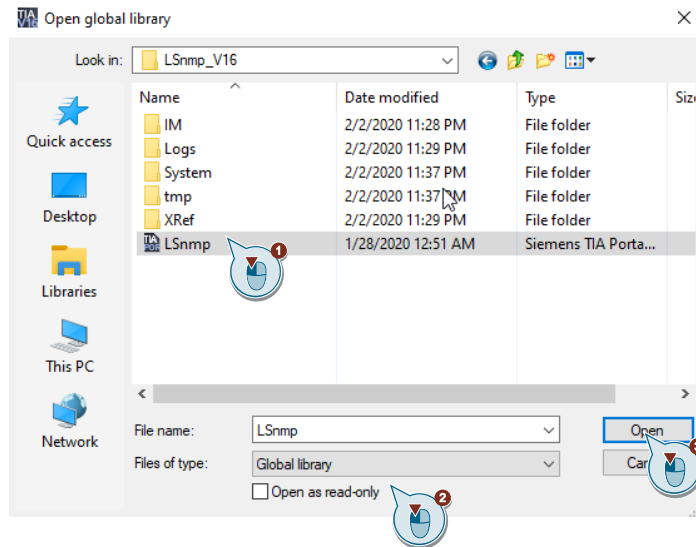
Below are the steps to integrate the elements of the library "LSnmp" into your STEP 7 V16 project. Afterwards you can use the blocks of the library.

1. Open the already existing STEP 7 V16-project.
2. In the toolbar of the "Global libraries" palette, click on "Open global library" or in the "Options" menu, click on the command "Global libraries > Open library."



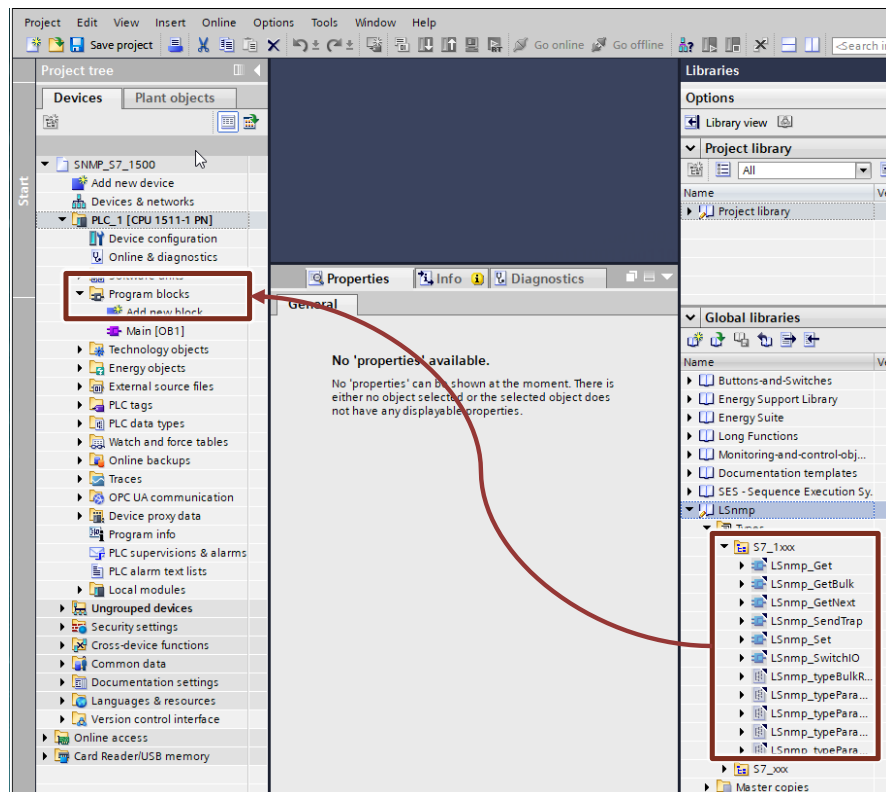
The "Open global library" dialog is opened.

3. Select the global library “LSnmp.”



The library “LSnmp” will open.

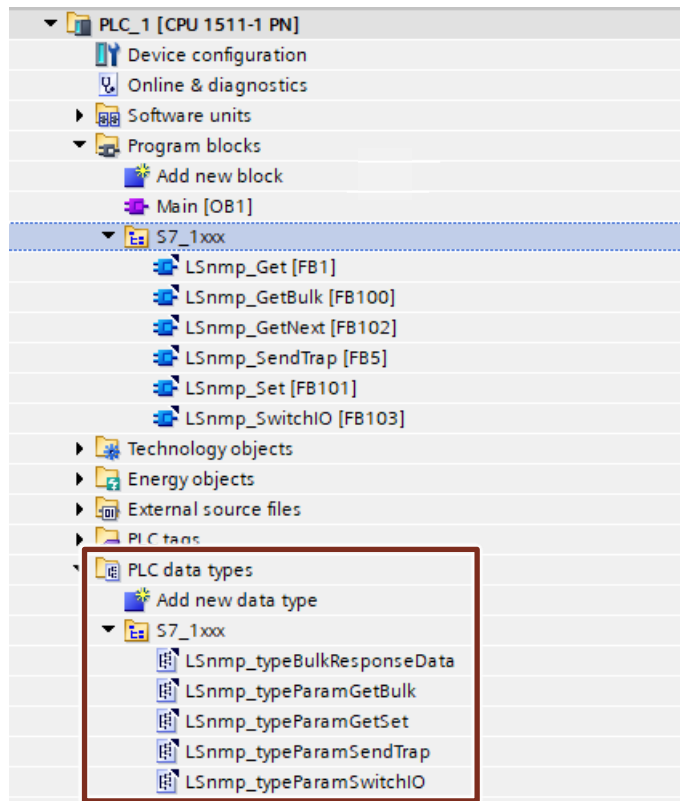
4. Drag the blocks to the folder “S7_1xxx” (for the S7-1500/S7-1200 CPU) in the folder “Program blocks” in the planned S7 CPU.



Note:

If you have planned an S7-300/S7-400 CPU, then drag the blocks in the folder “S7_ xxx” to the folder “Program blocks” in the S7-300/S7-400 CPU.

- The PLC data types are automatically copied to the "PLC data types" folder.



- Create a new global data block with the following variables:
 - the PLC data type "LSnmp_typeParameterGetSet" for call "LSnmp_Get" in OB1
 - the input parameters for the call in OB1
 - the output parameters for the call in OB1

snmpData			
	Name	Data type	Sta
1	Static		
2	snmpGetData	"LSnmp_typeParamGetSet"	
3	snmpGet	Bool	fal
4	resetGet	Bool	fal
5	doneGet	Bool	fal
6	busyGet	Bool	fal
7	errorGet	Bool	fal
8	statusGet	DWord	16

Note:

If using an S7-1500/S7-1200, you must use an "optimized module access."

7. Assign values to all parameters required for reading SNMP variables (see [section 2.7](#)):

- ipAddress
- hwIdentifier
- connectionID
- localPort
- watchDogTime
- oID (see [table 1-2](#))
- community

Note:

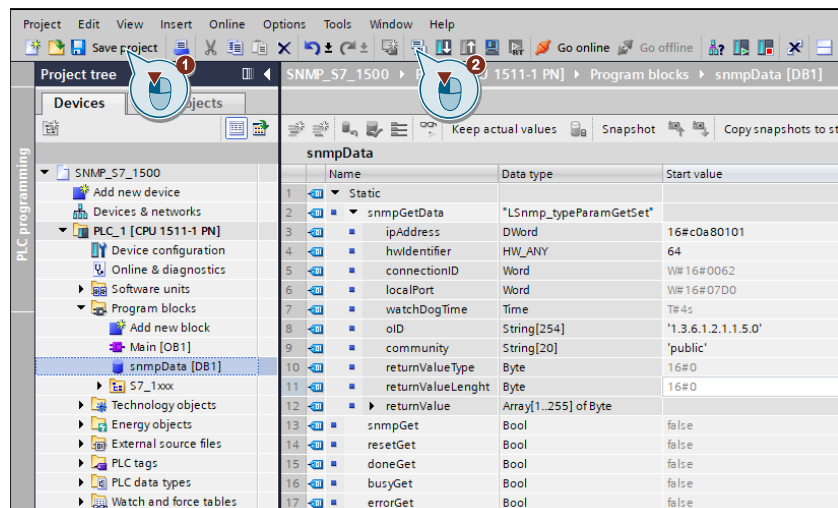
This value must match the value entered in Web Based Management ("Community String") of the network component (see [section 3, step 5](#)).

Name	Data type	Start value
Static		
snmpGetData	"LSnmp_typeParamGetSet"	
ipAddress	DWord	16#c0a80101
hwIdentifier	HW_ANY	64
connectionID	Word	W#16#0062
localPort	Word	W#16#07D0
watchDogTime	Time	T#4s
oID	String[254]	'1.3.6.1.2.1.1.5.0'
community	String[20]	'public'
returnValueType	Byte	16#0
returnValueLenght	Byte	16#0
returnValue	Array[1..255] of Byte	

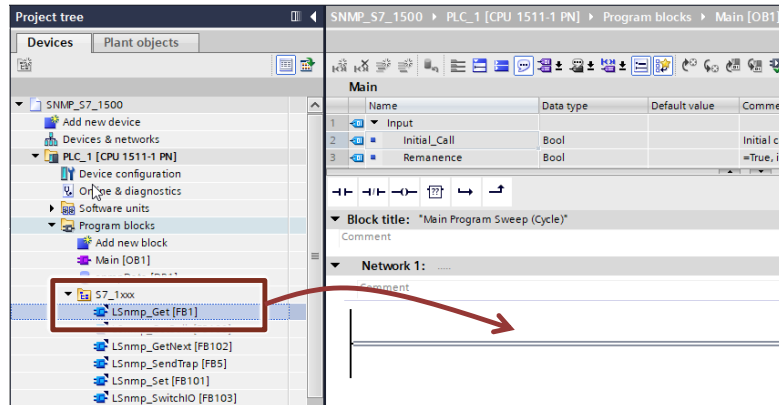
Note:

During the Read access ("LSnmp_Get"), the type, length, and value of SNMP variable are automatically determined and entered here.

8. Save and compile the newly created data block.



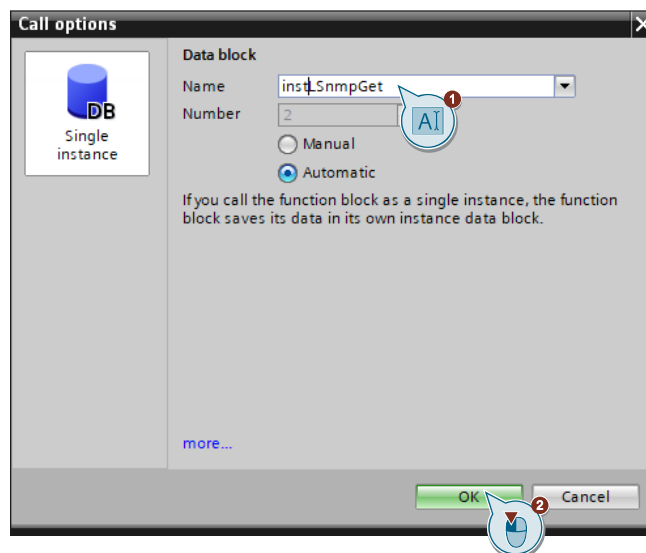
- Open the organization block OB1 in the folder "Program blocks" of your device and drag the function block "LSnmp_Get" to the network of your choice.



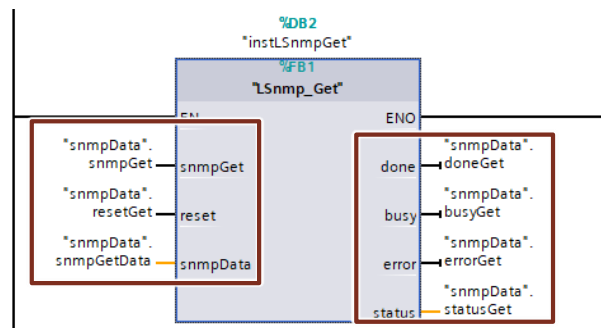
Note:

The SNMP blocks must be activated in cycles. This can be done in the OB1, for example. If you activate multiple SNMP blocks in OB1, you must select the respective different connection IDs (value range: W#16#0001 to W#16#0FFF).

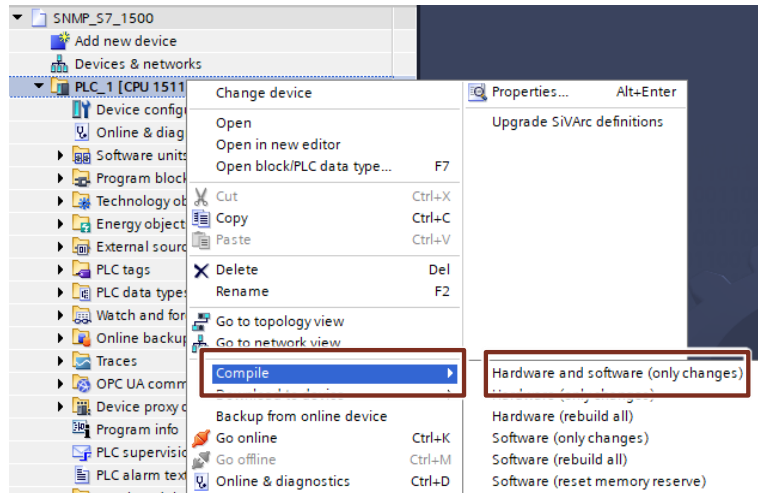
- Specify the name of the corresponding instance data block. Exit the dialog with "OK."



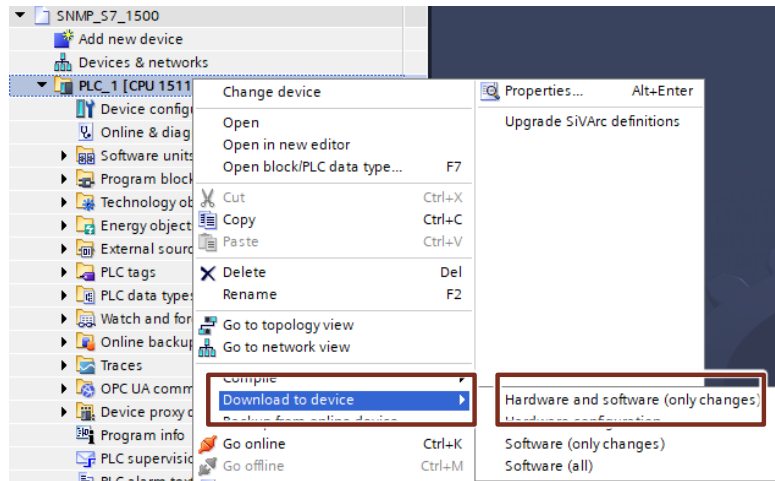
- Assign values to all parameters required. Take the values from the newly created data block.



12. Compile the hardware and software of the S7 station. To do this, right-click the device in project navigation and select the menu “Compile > Hardware and software (only changes).”



13. Load the new project into your controller.



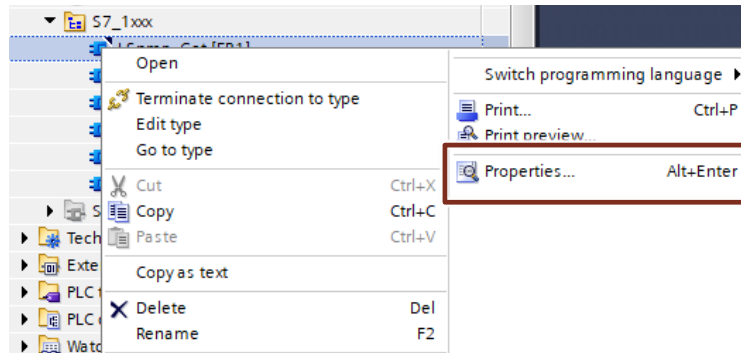
Note:

Your programming device, the S7 CPU, and the network components must always be in the same subnetwork.

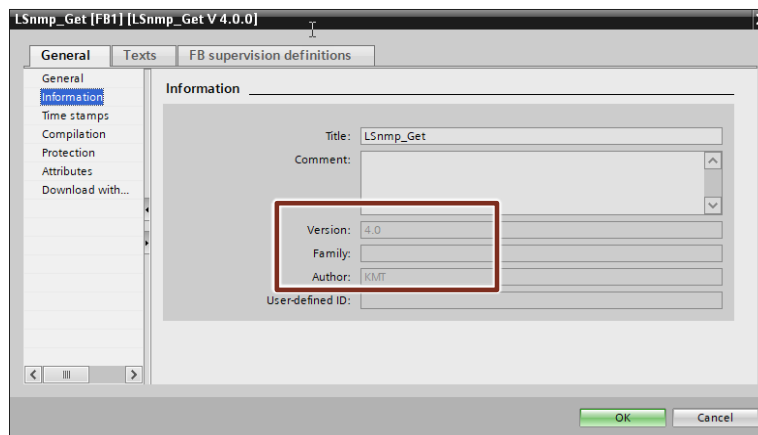
4.3 Updating the library “LSnmp”

The following instructions show:

- how to check whether the library is up to date
 - how to integrate a newer version of the library “LSnmp” into your STEP 7 project
1. Open the TIA portal and compare the current version number of each element in the library with the latest version from the Siemens Industry Online Support portal.
 - a. Right-click on the blocks in the library in project navigation. Select the "Object properties" option from the context menu.



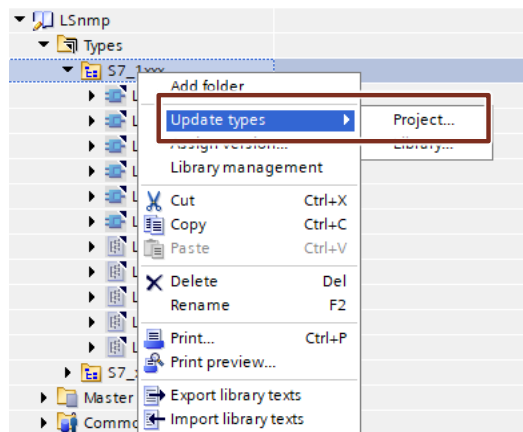
- b. Switch to the “Information” tab in the overlaid Properties window.



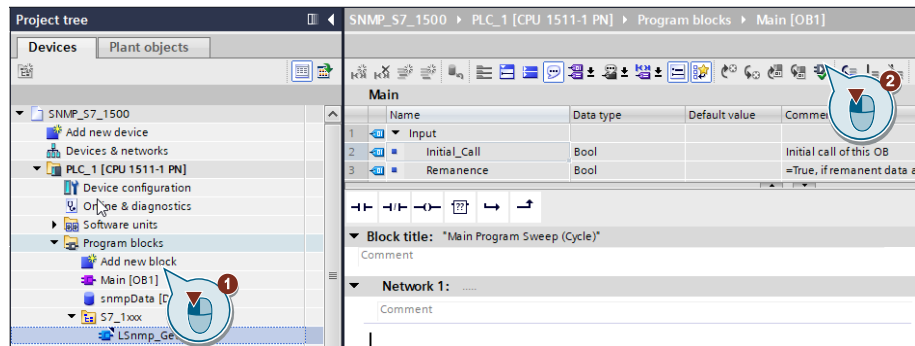
2. To update the library, integrate the current library as described in [section 4.2](#).
3. Open the new library as described in [section 4.2](#) steps 1 through 5.
4. Right-click on the library “LSnmp.”

4 Working with the library

5. Select the context menu “Update types > Project.”



6. The blocks are now updated. Click “Update inconsistent block calls” in the toolbar in OB1 to update or regenerate the instance DBs.



7. Compile the software after you have added the updated blocks to your STEP 7 Project and load the project.
8. The library update is now complete.

5 Appendix

5.1 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 iOS and Android:

support.industry.siemens.com/cs/ww/en/sc/2067

5.2 Links and Literature

Table 5-1

No.	Subject
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to the article page of the Application Example https://support.industry.siemens.com/cs/ww/en/view/57249109
\3\	Private MIBs: SCALANCE W and SNMP OPC Profile https://support.industry.siemens.com/cs/ww/de/view/35842319
\4\	Information about "Local_device_id" https://support.industry.siemens.com/cs/ww/en/view/51339682
\5\	SIMATIC NET: Industrial Remote Communication Remote Networks SCALANCE M-800 Web Based Management https://support.industry.siemens.com/cs/ww/de/view/109751635
\6\	SIMATIC NET: Network Management Diagnostics and Project Planning with SNMP https://support.industry.siemens.com/cs/ww/en/view/103949062

5.3 Change Documentation

Table 5-2

Version	Date	Change
V1.0	April 17, 2012	First version
V2.0	April 17, 2014	Complete revision of blocks; Integration into TIA V12; Blocks for S7-1500
V3.0	July 25, 2016	Additional SNMP blocks "SnpGetBulk" and "snmpGetNext"; Update to TIA V13 SP1
V4.0	February 5, 2020	Complete revision of blocks; Integration into STEP 7 V15.1 and V16