SIEMENS

Ingenuity for life

SNMP Blocks for S7 PN-PLCs for Diagnosing and Control of Network Components

Library / SIMATIC S7-PLCs

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

Siemens
Industry
Online
Support

Legal information

Use of application examples

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

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

Disclaimer of liability

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

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

Other information

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

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

Security information

Siemens provides products and solutions with Industrial Security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

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

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

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

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

Table of contents

1 Library Overview

What will I get?

This document describes the "LSnmp" block library for STEP TIA Portal. The block library provides you with tested code with clearly defined interfaces. They can be used as a basis for the task you want to implement.

This document describes

- all blocks pertaining to the block library
- the functionality implemented through these blocks

Furthermore, this documentation shows possible fields of application and helps you integrate the library into your TIA project using step-by-step instructions.

Scope of validity of the library

This library is valid for:

- STEP 7 TIA V15.1
- SIMATIC S7-1200 PLCs V4.0 or higher
- SIMATIC S7-1500 PLCs
- SIMATIC S7-400 PN/DP
- SIMATIC S7-300 PN/DP (firmware 2.5 or higher)

1.1 Application scenario

Overview

The status of SNMP-capable network components is monitored and, if necessary, controlled by network management systems (for example, SINEMA Server) via SNMP (Simple Network Management Protocol).

The blocks of the "LSnmp" library make it possible for a SIMATIC S7-PLC with PROFINET interface to request information from the network components as simple SNMP manager and also to control it, if required.

Note

Internally, SNMP is based on the connectionless UDP. Using the function blocks from the standard library, TUSEND (FB67) and TURCV (FB 68), data up to 1472 bytes can be sent and received via UDP.

The blocks of this library support sending and receiving SNMP messages whose overall length must not exceed 486 bytes.

The user data length for strings is limited to 255 bytes.

Schematic layout

The figure below shows a possible configuration where the SNMP blocks of the library can be used.

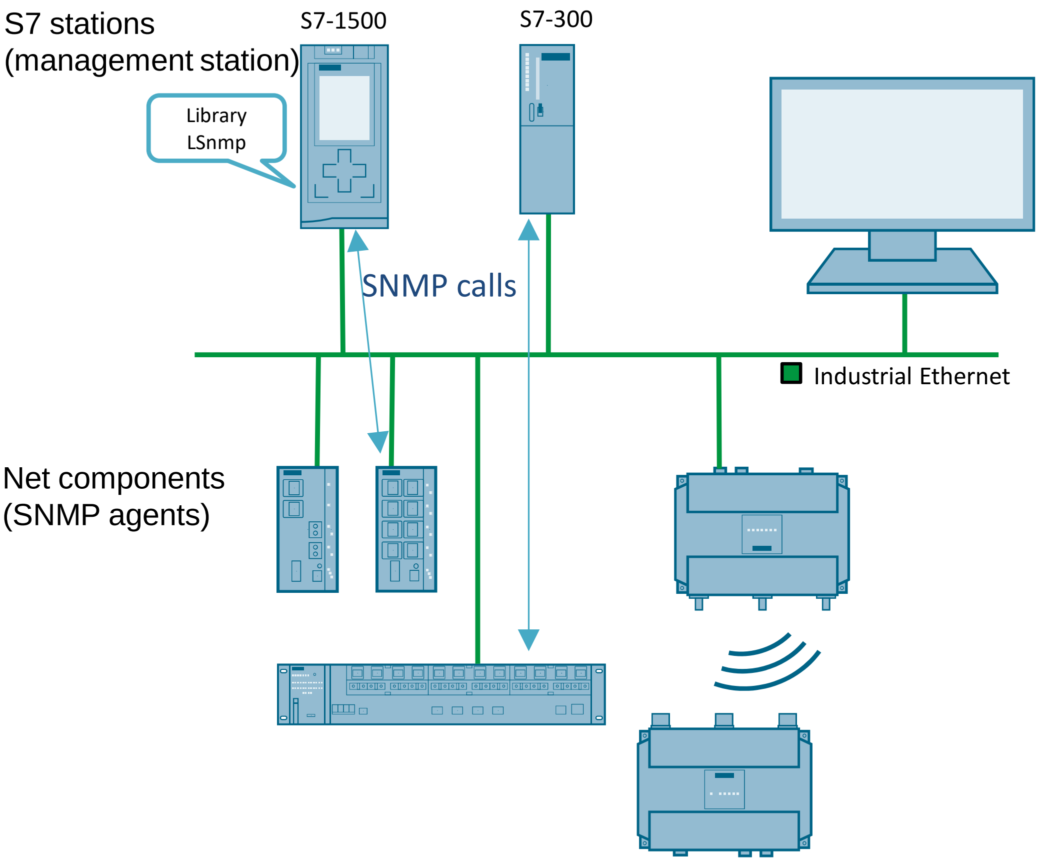
An S7 station can, for example,

- request or determine the current transmission power from an IWLAN access point to find out which IWLAN clients are currently logged on.
- request the link status of a port from a switch.
- switch the digital output of an IWLAN client.

Note These SNMP variables must exist in the private MIB ("Management Information Base") of the device (see chapter 4.2) or in the general MIB 2.

Note Examples for using SNMP blocks can be found on the HTML page of this entry (see chapter 4.2).

Figure 1-1



1.2 Functions

In order to enable a SNMP communication between a SIMATIC S7-PLC and the network component, the function blocks from the "LSnmp" library are required.

The following table describes the core functions of the function blocks.

Table 1-1

Function	Description	SNMP version
SnmpGet	Request of a SNMP variable from a SNMP agent (get request command).	SNMPv1
SnmpSet	Changing a SNMP variable of a SNMP agent (set request command).	SNMPv1
SnmpGetNext	Expanding the get request; Enables an automatic execution and request of the following objects within an OID subtree.	SNMPv1
SnmpGetBulk	Expanding the GetNext request; Makes the request of large data volumes of a SNMP agent with only one response frame possible.	SNMPv2
SwitchIO	Includes the functions "SnmpGet" and "SnmpSet" for switching the digital output of an IWLAN client.	SNMPv1

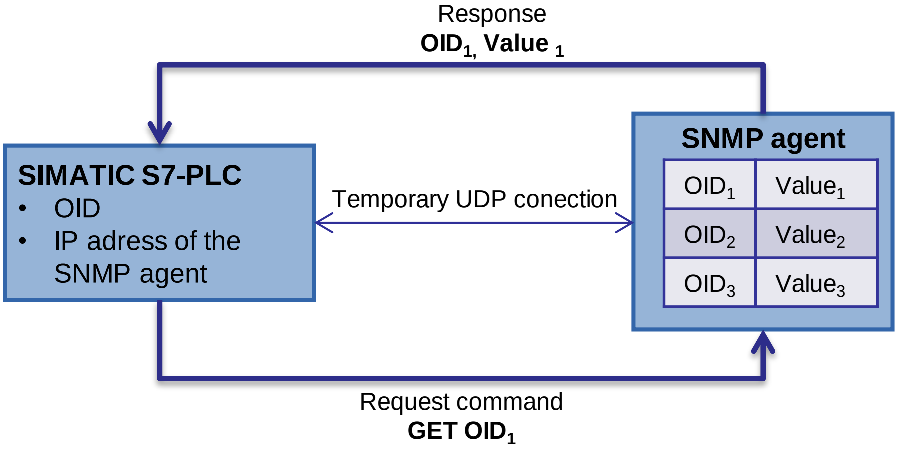
A temporary UDP/IP connection to the SNMP agent is established for each function.

Sequence for the get request command

For the "SnmpGet" function, the S7-PLC (SNMP manager) establishes an UDP connection to the SNMP agent and sends a "get request command". The frame includes the OID of the SNMP variables to be requested.

After the receipt of a response, the connection is disconnected again.

Figure 1-2

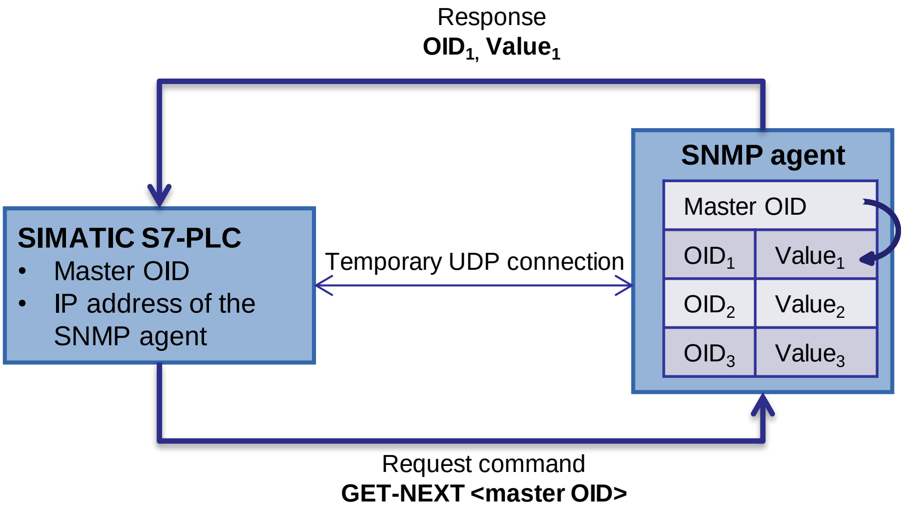


Sequence for the GetNext request command

For the "SnmpGetNext" and "SnmpGetBulk" function, the S7-PLC (SNMP manager) establishes an UDP connection to the SNMP agent and sends a "GetNext request command". The frame includes the previous OID (mainly master OID of the subtree) of the SNMP variables to be requested.

After the execution of the OID subtree, the connection is disconnected.

Figure 1-3

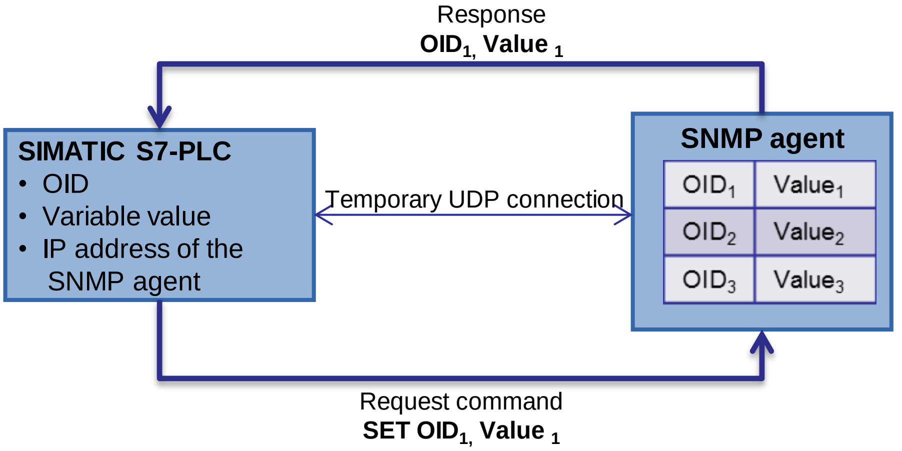


Sequence for the set request command

For the "SnmpSet" function, the S7-PLC establishes an UDP connection to the SNMP agent and sends a "SetRequest command". The frame includes the OID of the SNMP variables to be changed as well as the new variable value.

After the receipt of a response, the connection is disconnected again.

Figure 1-4



1.3 Hardware and software requirements

Hardware

For the hardware, the following requirements are necessary:

Table 1-2

No.	Component	Article number
1	SIMATIC PLC	• PLC15xx • PLC12xx (as of version 4) • PLC S7-300/400/ ET 200S-PLC - with integrated PROFINET interface - Support of the UDP protocol - Support of open communication blocks (T blocks)
2	Network components that are SNMP capable	For example, SCALANCE X202 2IRT

Software

The V15.1 configuration software is used as software

2 Blocks of the Library

What will you learn here?

This chapter explains the blocks of the "LSnmp" library.

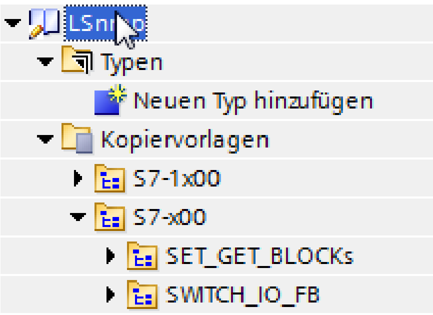
The library includes two main folders:

- **S7-x00** for the application with a S7-300/400, ET 200S PLC, WinAC
- **S7-1500/S7-1200** for the application with a S7-1x00.

The content of the folders is identical; only the program code is optimized for the respectively used controllers.

For each controller there are two folders available, as shown by the following figure

Figure 2-1



LSnmp
Typen
Neuen Typ hinzufügen
Kopiervorlagen
S7-1x00
S7-x00
SET_GET_BLOCKs
SWITCH_IO_FB

- **SET_GET_Blocks** includes the basic functions "SnmpGet", "SnmpGetBulk", "SnmpGetNext" and "SnmpSet" as well as the global data blocks that are required for the configuration of the function blocks.
- **SWITCH_IO_FB** includes the "SwitchIO" application block as well as a global data block that is required for the configuration of the function block.

2.1 Block list

The table below lists all blocks belonging to the "LSnmp" library.

Table 2-1

Icon	Classification	Folder
SnmpGet	In-house development	SET_GET_Blocks
SnmpGetBulk	In-house development	SET_GET_Blocks
SnmpGetNext	In-house development	SET_GET_Blocks
SnmpSet	In-house development	SET_GET_Blocks
SnmpGetParam	In-house development	SET_GET_Blocks
SnmpSetParam	In-house development	SET_GET_Blocks
SnmpGetBulkParam	In-house development	SET_GET_Blocks
typeParamGetSet	In-house development	SET_GET_Blocks
typeParamGetBulk	In-house development	SET_GET_Blocks
typeParamGetBulkResponseData	In-house development	SET_GET_Blocks
SwitchIO	In-house development	SWITCH_IO_FB
SwitchIOParam	In-house development	SWITCH_IO_FB
typeParamSwitchIO	In-house development	SWITCH_IO_FB

2.2 Explanation of the blocks from SET_GET_Blocks

2.2.1 FB "SnmpGet"

Overview

The "SnmpGet" block is a configurable function block for reading SNMP variables. All information that this block requires is stored in the global "SnmpGetParam" data block of type "typeParamGetSet" (information see chapter 2.2.3) and transferred to the block as input/output parameter.

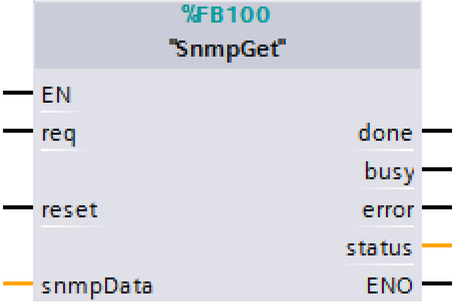
Function principle

The "SnmpGet" block sends an SNMP get request command to the SNMP agent in the network component. The network component responds with a SNMP get response command that contains the requested data or an error message.

Illustration and configuration

The FB "SnmpGet" is called as follows:

Figure 2-2



The "SnmpGet" block includes the following input parameters:

Table 2-2

Parameter	Data type	Description
EN	BOOL	Enable input Relevant only in the FBD and LAD view.
req	BOOL	When there is a positive edge, the variable will be read.
reset	BOOL	When there is a positive edge, a reset of the UDP connection parameters will be triggered.

The SnmpGet block has an input and output parameter:

Table 2-3

Parameter	Data type	Description
snmpData	"typeParamGetSet"	Specifying a data structure; transferring all relevant information for the UDP connection and SNMP variable. Storage for the response data of the net component (for example, return value of the variable etc.) (See chapter 2.2.3).

The "SnmpGet" block includes the following output parameters:

Table 2-4

Parameter	Data type	Description
done	BOOL	TRUE when the last job was processed without errors. Only TRUE for one cycle.
busy	BOOL	Set to TRUE when the "SnmpGet" block is active. Assumes the FALSE status as soon as the operation is completed or an error occurs. Only TRUE for one cycle.
error	BOOL	TRUE if an error occurs when processing the routine. Only TRUE for one cycle.
status	DWORD	Status, if ERROR=TRUE. Only active for one cycle.
ENO	BOOL	Enable output. Relevant only in FBD and LAD representation.

Status and error displays

The "status" parameter can assume the following error states:

Table 2-5

Status	Meaning	Remedy/notes
16#00008101	Previous job is not yet complete. (You have started a new REQ operation, although BUSY was still active)	• Restart the REQ operation
16#00138102	The SNMP packet is too large for sending	• Check and change the size of the packet • Restart the REQ operation
16#00138103	The OID is not supported	• Check and change the MIB object (OID) • Restart the REQ operation
16#00138106	unknown generation error	• Restart the REQ operation
16#00008107	The length of the received "VALUE" variable exceeds 255 characters. Only 255 characters were transmitted.	
16#00008108	Watchdog timer has expired.	• Check the communication between controller and network components • Check and change the community string (must match the one in the configuration of the remote partner for read access) • Check the IP address of the remote partner • Restart the REQ operation
16#00008109	The overall length of the GETResponse packet exceeds 486 characters. Only the first 486 characters were transmitted.	

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

Note

Errors with a different status are errors of the subordinate communication blocks:

TCON:	**DW#16#0001xyyy**
TUSEND:	**DW#16#0010xyyy**
TURCV:	**DW#16#0011xyyy**
TDISCON:	**DW#16#0012xyyy**

The specific error information (coded in xyyy) can be found in the online help of the respective communication block.

If a communication error occurs, the PLC must be restarted after the elimination of the error (for example, after changing an incorrect parameter).

2.2.2 FB "SnmpGetNext"

Overview

In principle, the FB "SnmpGetNext" block works like the "SnmpGet" block, with the exception that it enables the reading of the subsequent SNMP variables within a SNMP object tree.

All information that this block requires is stored in the global "SnmpGetParam" data block of type "typeParamGetSet" (information see chapter 2.2.3) and transferred to the block as input/output parameter.

Function principle

The "SnmpGetNext" block uses the SNMP GetNext request command and facilitates the request of subsequent objects of a MIB subtree. The objects within a MIB subtree are characterized by the fact that all objects have the same master OID and that they can only be distinguished based on the suffixes.

The GetNext command is primarily used for the execution of a table or a table column and works as follows:

The first call of GetNext goes to the master OID. The SNMP agent does not respond with the return values of the requesting OID (here: the master OID) - as is the case with the get request command, but with the OID (master OID + Suffix$_1$) and the return value (Value$_1$) of the following object. The next call of GetNext is now sent to the receiving OID (master OID + Suffix$_1$). The response is the next following OID (master OID + Suffix$_2$) and its return value (Value$_2$).

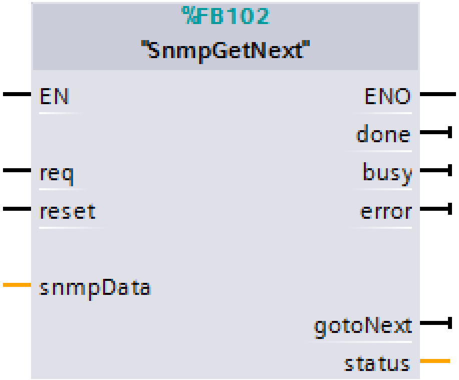
The execution of the MIB subtree is automatic and until the end of the subtree has been reached.

This is implemented via a "while" loop in the program, which is executed until the master OID of the SNMP variable changes.

Illustration and configuration

The FB "SnmpGetNext" is called as follows:

Figure 2-3



In terms of configuration, the FB "SnmpGetNext" is similar of the "SnmpGet" block, so that the identical parameters do not have to be explained in the following.

The "goToNext" output parameter is new. If there is a positive edge, it signals another run through the loop and thus the reading of the subsequent SNMP object.

Use this edge to transfer the currently stored response data to another memory area. Otherwise this data will be overwritten by the next request.

2.2.3 FB "SnmpGetBulk"

Overview

The FB "SnmpGetBulk" block always works like the "SnmpGetNext" block. It is used to minimize the network transfer since it allows the efficient reading of large data volumes with only one single response frame.

All information that this block requires is stored in the global "SnmpGetBulkParam" data block of type "typeParamGetBulk" (information see chapter 2.2.3) and transferred to the block as input/output parameter.

Function principle

The "SnmpGetBulk" block uses the SNMP GetBulk request command and therefore requires SNMPv2 – a further development from SNMPv1.

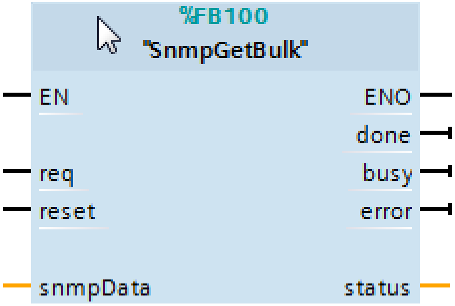
The GetBulk command executes several GetNext requests internally and returns the event in one single response frame. The number of GetNext commands can be specified via a configurable repeat factor in the GetBulk frame.

The return value of all requested objects are collected in one single response frame and transferred. Thus, it is the job of the "SnmpGetBulk" block to restructure the received data stream in the return values of the individual objects and to provide it split at the output parameter.

Illustration and configuration

The FB "SnmpGetBulk" is called as follows:

Figure 2-4



In terms of configuration the FB "SnmpGetBulk" is identical with the "SnmpGet" block, so that it does not have to be explained anymore.

2.2.4 FB "SnmpSet"

Overview

The "SnmpSet" block is a configurable function block for writing SNMP variables. All information that this block requires is stored in the global "SnmpGetParam" data block of type "typeParamGetSet" (information see chapter 2.2.3) and transferred to the block as input/output parameter.

Function principle

The "SnmpSet" block sends a write job for a SNMP variable to the SNMP agent in the network component via the SNMP GetRequest command.

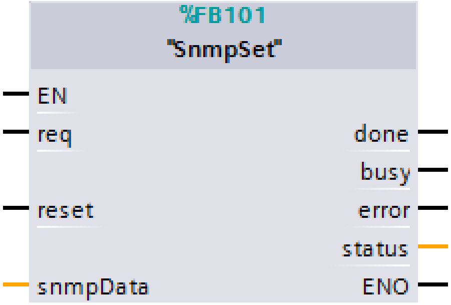
In the SetResponse response frame, the block receives the result of the write job from the network component.

If this read-in variable does not match the written value, an error will be output. The write job has to be transmitted again.

Illustration and configuration

The FB "SendSet" is called as follows:

Figure 2-5



The "SnmpSet" block includes the following input parameters:

Table 2-6

Parameter	Data type	Description
EN	BOOL	Enable input. Relevant only in the FBD and LAD view.
req	BOOL	When there is a positive edge, the variable will be written.
reset	BOOL	When there is a positive edge, a reset of the UDP connection parameters will be triggered.
snmpData	"typeParamGetSet"	Specifying a data structure; transferring all relevant information for the UDP connection and SNMP variable. Transfer of the information to the SNMP variables to be written. (See chapter 2.2.8).

Output parameters

The "SnmpSet" block includes the following output parameters:

Table 2-7

Parameter	Data type	Description
done	BOOL	TRUE when the last job was processed without errors. Only TRUE for one cycle
busy	BOOL	Set to TRUE when the "SnmpSet" block is active. Assumes the FALSE status as soon as the operation is completed or an error occurs. Only TRUE for one cycle
error	BOOL	TRUE if an error occurs when processing the routine. Only TRUE for one cycle
status	WORD	Status, if ERROR=TRUE. Only active for one cycle.
ENO	BOOL	Enable output. Only relevant in FBD and LAD representation

Status and error displays

The "status" parameter can assume the following error states:

Table 2-8

Status	Meaning	Remedy/notes
16#00008101	Previous job is not yet complete. (You have started a new REQ operation, although BUSY was still active)	• Restart the REQ operation
16#00138102	The SNMP packet is too large for sending	• Check and change the size of the packet • Restart the REQ operation
16#00138103	The OID is not supported or OID is only supported for read access	• Check and change the MIB object (OID) • Restart the REQ operation
16#00138104	Incorrect data type or value (only possible as a response to SET packets)	• Check and change the MIB object (OID) or Value_Type • Restart the REQ operation
16#00138106	unknown generation error	• Restart the REQ operation
16#00008108	Watchdog timer has expired.	• Check the communication between controller and network components. • Check the IP address of the remote partner. • Check the community string (must match the one in the configuration of the remote partner for read and write access). • Restart the REQ operation
16#00008109	The get response value does not match the set request value.	• Restart the REQ operation
16#00008110	The length of the get response value does not match the length of the set request value.	• Restart the REQ operation

Note

Errors with the **16#0013xyyy** status are SNMP protocol errors!

Note

Errors with a different status are communication block errors:

TCON: **DW#16#0001xyyy**

TUSEND: **DW#16#0010xyyy**

TURCV: **DW#16#0011xyyy**

TDISCON: **DW#16#0012xyyy**

The specific error information (coded in xyyy) can be found in the online help of the respective communication block.
If a communication error occurs, the PLC must be restarted after the elimination of the error (for example, after changing an incorrect parameter).

2.2.5 DB "SnmpGetParam"

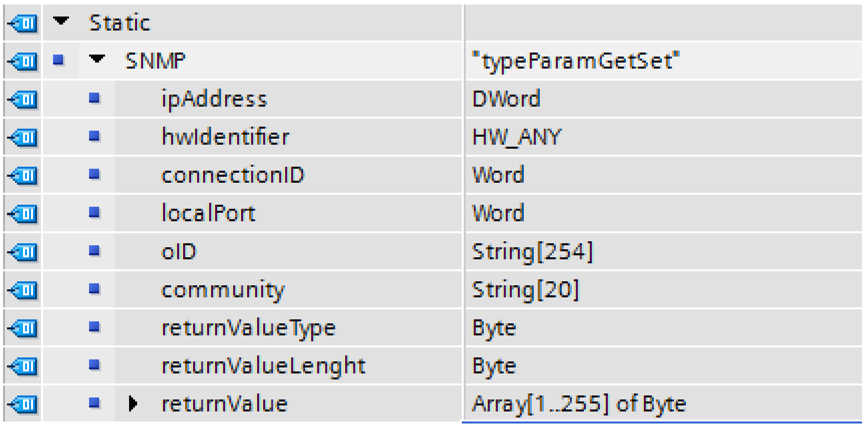
The DB "SnmpGetParam" is a global data block and includes

- the configuration data for the UDP connection of the network component,
- parameters for the SNMP variables to be requested,
- storage area for the response data of the network components.

All this information is stored in a defined data structure.
The PLC data type "typeParamGetSet" is used as template.

The defined data structure can be used multiply in the program.

Figure 2-6



▼ 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

Note

The structure of the "typeParamGetSet" PLC data type is made up of several components.

A detailed description is available in chapter 2.2.8.

2.2.6 DB "SnmpGetBulkParam"

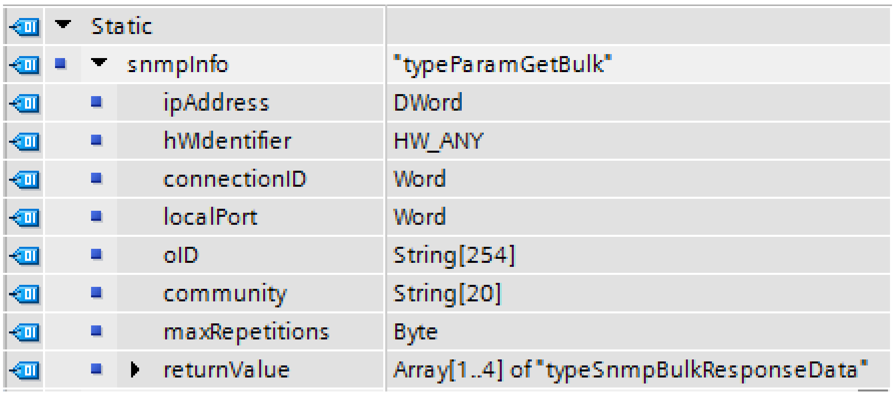
The DB "SnmpGetBulkParam" is a global data block and includes

- the configuration data for the UDP connection of the network component,
- parameters for the SNMP variables to be requested
- storage area for the response data of the network components.

All this information is stored in a defined data structure.
The PLC data type "typeParamGetBulk" is used as template.

The defined data structure can be used multiply in the program.

Figure 2-7



▼ Static	
▼ snmpInfo	"typeParamGetBulk"
ipAddress	DWord
hWIdentifier	HW_ANY
connectionID	Word
localPort	Word
oID	String[254]
community	String[20]
maxRepetitions	Byte
▶ returnValue	Array[1..4] of "typeSnmpBulkResponseData"

Note The structure of the PLC data type "typeParamGetBulk" is made up of several components.

A detailed description is available in chapter 2.2.9.

2.2.7 DB "SnmpSetParam"

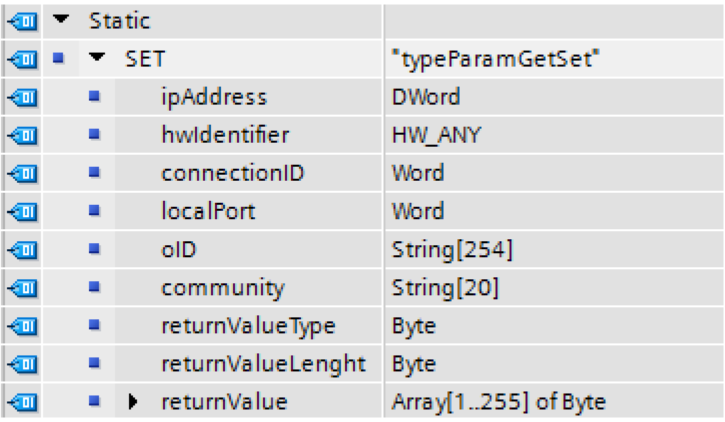
The "SnmpGetParam" is a global data block and includes

- the configuration data for the UDP connection of the network component,
- parameters for the SNMP variable to be written.

All this information is stored in a defined data structure.
The PLC data type "typeParamGetSet" is used as template.

The defined data structure can be used multiply in the program. In "SnmpSetParam" an array with four data types was created as an example and thus enabling a writing of the four different SNMP variables.

Figure 2-8



▼ Static	
▼ SET	"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

Note

The structure of the "typeParamGetSet" PLC data type is made up of several components.

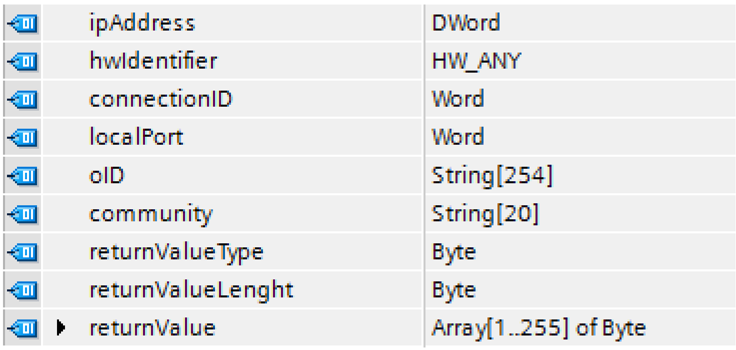
A detailed description is available in chapter 2.2.8.

2.2.8 UDT "typeParamGetSet"

Overview

The "typeParamGetSet" PLC data type is a defined data structure that is used several times in the program and which is used as template for creating the global data blocks "SnmpGetParam" and "SnmpSetParam".
The structure of the PLC data type is made up of several components.

Figure 2-9



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

Configuration data for the UDP connection

The blocks "SnmpGet" and "SnmpSet" require special information for establishing the UDP connection to the network component. The following variables in the PLC data type are responsible for it:

Table 2-9

Parameter	Description
ipAdress	IP address of the network component
hwIdentifier	Valid for the **S7-x00** library folder: local_device_id of the S7-PLC type (see chapter 4.2). Valid for the **S7-1x00** library folder: Hardware ID of the PROFINET interface of the S7-1x00 PLC.
connectionId	The connection ID for the SNMP block that is necessary for the UDP connection is invariably set to the value **W#16#62**. If you want to configure other open communication connections, in addition to the UDP connection, you have to select different connection IDs (value range: W#16#0001 to W#16#0FFF) and change the default value.
localPort	The local port no of the UDP connection is configured to value **W#16#7D0**. If you want to configure other open communication connections, in addition to the UDP connection, you have to select different LOCAL_PORT no. for these connections and change the default value.

Parameters of the SNMP variables

To read ("SnmpGet") or write ("SnmpSet") SNMP variables, information on SNMP variables is required. The following variables in the PLC data type are responsible for it:

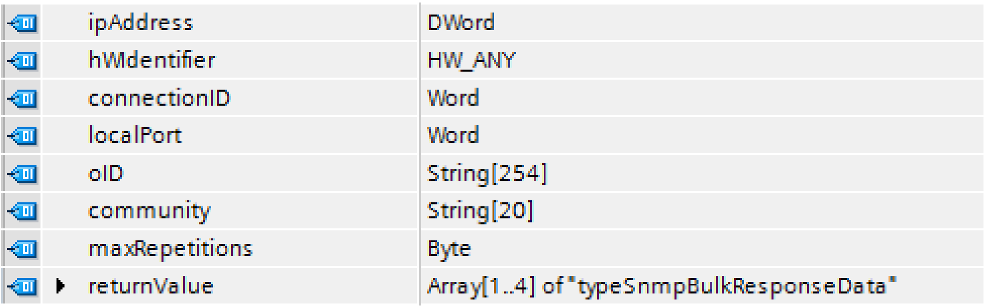
Parameter	Description
oID	Object identifier of the SNMP variable in SNMP format **(for example, "1.3.6.1.2.1.1.4.0")**. The OID object can be found in the general (RFC1213 II: MIB II) or private MIB file of the device (see chapter 4.2).
community	In most cases, "private" is selected as the community name for read access and for the write access "private" is selected. This value has to match the community name which is selected in the configuration of the network component.
returnValueType	Data type of the SNMP variable: 02: Integer, 04: String, 41: Counter, 43: Timeticks For the read access ("SnmpGet") the type of the SNMP variable is determined automatically and entered here. For the write access ("SnmpSet") the type of the SNMP variable has to be configured.
returnValueLenght	Length of the SNMP variable. For the read access ("SnmpGet") the length of the SNMP variable is determined automatically and entered here. For the write access ("SnmpSet") the length of the SNMP variable has to be configured.
returnValue	ARRAY OF BYTE. The length of the array is limited to 255 bytes. For read access ("SnmpGet") the response data of the SNMP variable is entered here. For write access ("SnmpSet") the data has to be entered here with which the SNMP variable is to be written.

2.2.9 UDT "typeParamGetBulk"

Overview

The "typeParamGetBulk" PLC data type is a defined data structure for the "GetBulk" function that is used as template for creating the "SnmpGetBulkParam" global data block.
The structure of the PLC data type is made up of several components.

Figure 2-10



ipAddress	DWord
hWIdentifier	HW_ANY
connectionID	Word
localPort	Word
oID	String[254]
community	String[20]
maxRepetitions	Byte
▸ returnValue	Array[1..4] of "typeSnmpBulkResponseData"

Configuration data for the UDP connection

Just as the blocks "SnmpGet", "SnmpGetNext" and "SnmpSet" the "SnmpGetBulk" also requires special information in order to establish the UDP connection to the network component. Since this does not differ from the parameters of the other functions, there will be no description. Details can be found in chapter 2.2.8.

Parameters of the SNMP variables

To read or write SNMP variables, information on SNMP variables is also required here. In addition to the parameters known from chapter 2.2.8 the "SnmpGetBulk" block requires a repeat factor and a return range as array of the data type "typeSnmpBulkResponseData".

The additional variables in the PLC data type are responsible as follows:

Parameter	Description
maxRepetitions	Specification how many successive objects of an OID tree are to be read.
returnValue	This array is of the "typeSnmpBulkResponseData" data type and includes a field for storing the return information such as data type, length and value for each read object. The size of the array has to match the "maxRepetitions" parameter.

2.2.10 Call environment of the SNMP blocks

The SNMP blocks must be called cyclically. This can be done either in OB1 or alternatively in a time interrupt OB.

Detailed application examples of using the SNMP blocks are available on the HTML page from which you downloaded this document.

Note The GET, GETNEXT and SET operation is monitored using an internal timer. If these operations are not completed with a positive result within the monitoring time, an error will be output.

WATCH_DOG_TIME default value: 4 sec.

If you want to change the time, you can enter the value directly in the instance data block of the function (#WATCH_DOG_TIME constant).

2.3 Explanation of the blocks from SWITCH_IO_FB

2.3.1 FB "SwitchIO"

Overview

The "SwitchIO" block is a block that is programmed in SCL to demonstrate an application of the "SnmpGet" and "SnmpSet" blocks.

The "SwitchIO" user block enables a SIMATIC S7-PLC to switch and read the digital output of SCALANCE W devices via SNMP.

All information that this block requires is stored in the global "SwitchIOParam" data block of type "typeParamSwitchIO" (information see chapter 2.2.3) and transferred to the block as input parameter.

Function principle

The "SwitchIO" block realizes two functions

- Requesting the state of the digital output of a SCALANCE W module.
- Switching the digital output of a SCALANCE W module.

To demonstrate these scenarios, the function block uses the basic functions from the SET_GET_Blocks folder and internally calls the function blocks "SnmpGet" and "SnmpSet".

To request the status of the digital output, a SNMP GetRequest command is sent to the IWLAN component after a trigger command with the "SnmpGet" block.
It responds to the request with a GetResponse frame that contains the desired status (output is set ("true"), output is not set ("false")).

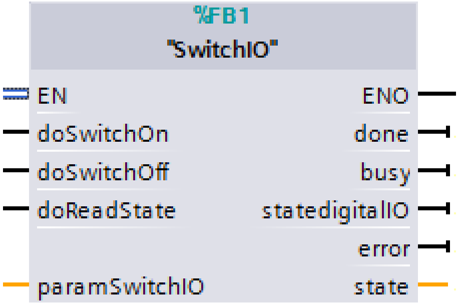
To switch the digital output, a SNMP SET packet is established after the after trigger command that includes the desired value (set output, reset output). The write job is sent to the IWLAN components via the "SnmpSet" block.

It responds to the request with a GetResponse fame that includes the result of the write job. If the read-in variable matches the overwritten value, the switching operation was successful.

Illustration and configuration

The FB "SwitchIO" is called as follows:

Figure 2-11



The "SwitchIO" block includes the following input parameters:

Table 2-10

Parameter	Data type	Description
EN	BOOL	Enable input Relevant only in the FBD and LAD view.
doSwitchOn	BOOL	The digital output is set with a positive edge.
doSwitchOff	BOOL	The digital output is reset if there is a positive edge.
doReadState	BOOL	The state of the digital output is read with a positive edge.
paramSwitchIO	"typeParamSwitchIO"	Specifying a data structure; Transferring all relevant information for the UDP connection. (See chapter 2.3.3)

The "SwitchIO" block includes the following output parameters:

Table 2-11

Parameter	Data type	Description
done	BOOL	TRUE when the last job was processed without errors. Only TRUE for one cycle.
busy	BOOL	Set to TRUE when the "SwitchIO" block is active. Assumes the FALSE status as soon as the operation is completed or an error occurs. Only TRUE for one cycle.
statedigitalIO	BOOL	Status of the digital output of the last action TRUE: digital output "ON" FALSE: digital output "OFF"
error	BOOL	TRUE if an error occurs when processing the routine. Only TRUE for one cycle.
status	DWORD	Status, if ERROR=TRUE. Only active for one cycle.
ENO	BOOL	Enable output. Relevant only in FBD and LAD representation.

Static variables and constants

The "SwitchIO" block is only designed for requesting and switching the digital output of a SCALANCE W component. Some relevant information for the UDP connection and SNMP variable is therefore firmly defined. The following table lists the most important static variables and constants:

Table 2-12

Parameter	Data type	Description
statParamGetSet	"typeParamGetSet"	The blocks "SnmpGet" and "SnmpSet" require special information for establishing the UDP connection to the network component. The UDT is configured at runtime with the required data.
statOID	STRING[40]	The OID for the digital output is 1.3.6.1.4.1.4329.20.1.1.1.1.39.1.3.1.6.1. This value is transferred to the UDT "typeParamGetSet" at runtime.
COMM_SET COMM_GET	STRING[10]	As community name, "public" is selected for the read access and "private" for write access. These values are transferred, depending on the scenario, to the UDT "typeParamGetSet" at runtime.
LOCAL_PORT	WORD	The local port no of the UDP connection is configured to value **W#16#7D0**. This value is transferred to the UDT "typeParamGetSet" at runtime.
instSnmpGet	"SnmpGet"	Blocks "SnmpGet"
instSnmpSet	"SnmpSet"	Block "SnmpSet"
ENO	BOOL	Enable output. Relevant only in FBD and LAD representation.

The following variables from the UDT "typeParamGetSet" are directly written at runtime:

- returnValue[1]:= 1 (reset digital output) or 2 (set digital output)
- returnValueType:= 2
- returnValueLenght:= 1

The parameter IP address, ConnectionID and DeviceID are configured via the global "ParamSwitchIO" data block at runtime.

Status and error displays

The "status" parameter can assume the following error states:

Table 2-13

Status	Meaning	Support/remarks
16#0000081A	The parameters "doSwitchOn" and "doSwitchOff" are simultaneously active.	• Reset these parameters • Restart the operation
Errors with the **16#01xyyyyy** status are errors that occur when the digital output is switched on. Errors with the **16#10xyyyyy** status are errors that occur when the digital output is switched off. Errors with the **16#11xyyyyy** status are errors that occur when the digital output is read.		
Information on the errors that can have different status can be read in chapter 2.2.1 and 2.2.4.		

2.3.2 DB "SwitchIOParam"

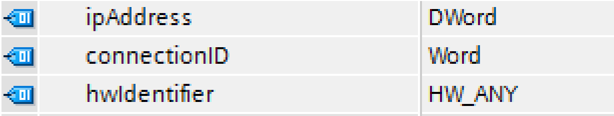
The DB "SwitchIOParam" is a global data block and includes the yet missing configuration data for the UDP connection for the network component:

- IP address
- Connection ID
- DeviceID

All this information is stored in a defined data structure.
The PLC data type "typeParamSwitch" is used as template.

The defined data structure can be used multiply in the program.

Figure 2-12



ipAddress	DWord
connectionID	Word
hwIdentifier	HW_ANY

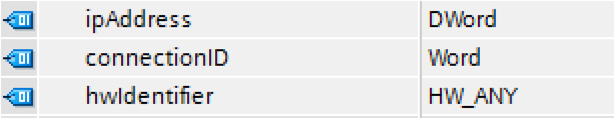
Note

> The structure of the PLC data type "typeParamSwitch" is made up of several components.
>
> A detailed description can be found in the following chapter 2.3.3.

2.3.3 UDT "typeParamSwitch"

Overview

The PLC data type "typeParamSwitch" is a defined data structure that is used several times in the program and is used as template for creating the global "SwitchIOParam" data block.
The structure of the PLC data type is made up of several components.



ipAddress	DWord
connectionID	Word
hwIdentifier	HW_ANY

Configuration data for the UDP connection

The blocks "SnmpGet" and "SnmpSet" require special information for establishing the UDP connection to the network component. The following variables in PLC data type are responsible for this and are transferred to the "typeParamGetSet" data structure at runtime.

Table 2-14

Parameter	Description
ipAdress	IP address of the network component
connectionId	The connection ID for the SNMP block that is necessary for the UDP connection is invariably set to the value **W#16#62**. If you want to configure other open communication connections, in addition to the UDP connection, you have to select different connection IDs (value range: W#16#0001 to W#16#0FFF) and change the default value.
hwIdentifier	Valid for the **S7-x00** library folder: DEVICE_ID of the S7 PLC type (see chapter 4.2). Valid for the **S7-1x00** library folder: Hardware ID of the PROFINET interface of the S7-1500 PLC.

3 Working with the Library

What will you learn here?

This chapter consists of instructions for integrating the "LSnmp" library into your STEP 7 project and instructions for using the library blocks.

3.1 Integrating the library into STEP 7

The table below lists the steps for integrating the "LSnmp" library into your STEP 7 project. Subsequently, you can use the blocks of "LSnmp" library.

Note The following section assumes that a TIA project exists.

Preparation

Table 3-1

No.	Action
1.	The library is available on the HTML page from which you downloaded this document. Save the library to your hard drive.
2.	Retrieve the library with the help of a data compression program.

Opening the library

Table 3-2

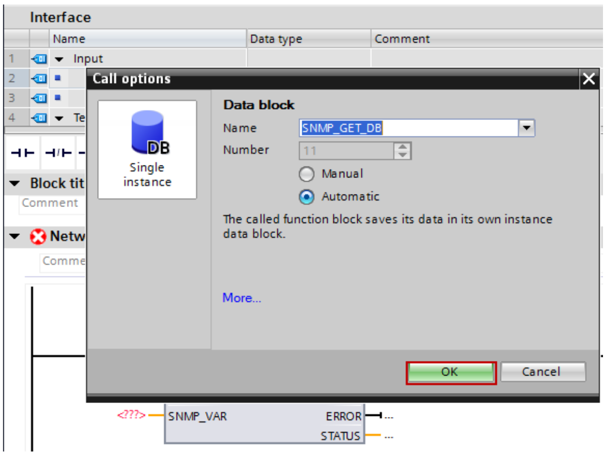
No.	Action	Note
1.	Open your TIA project with TIA V15.1 and go to the project view.	
2.	Click the icon in the "Global library" tab.	The "Open global library" dialog opens.
3.	Select the path in which the library is saved.	
4.	Click on "Open". The "LSnmp" library opens.	

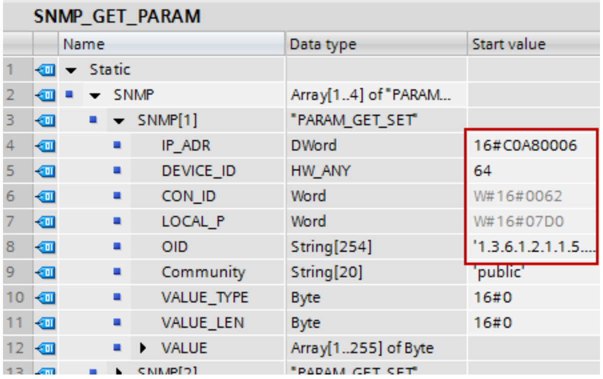
3.2 Integrating the library blocks into TIA V15.1

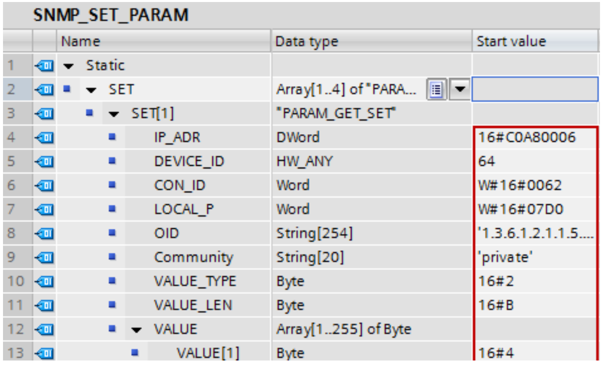
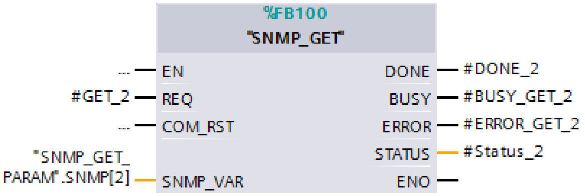
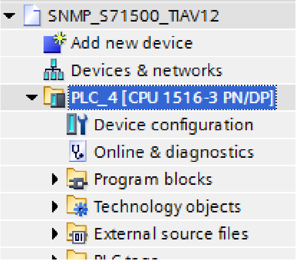
Below, you will find the steps how to integrate the "LSnmp" library into your TIA project.

Note The following instruction shows how to integrate the blocks on the example of SET_GET_Blocks. The integration of the blocks from the other folders is analogous.

Table 3-3

<table>
<tr><th>No.</th><th>Action</th><th>Note</th></tr>
<tr><td>1.</td><td>Open the <b>Master copies</b> folder in the library and here the subfolder <b>S7-x00</b> or <b>S7-1x00</b> &gt; <b>SET_GET_Blocks</b>.</td><td>The <b>S7-x00</b> subfolder includes the blocks for a S7-300/400/ET200 PLC or WinAC.
The <b>S7-1x00</b> subfolder includes the blocks for a S7-1500 or S7-1200.</td></tr>
<tr><td>2.</td><td>Drag the copy templates via drag-and-drop to the desired place in the TIA project.</td><td>The final destination for the templates "SnmpGet", "SnmpGetNext", "SnmpGetBulk", "SnmpSet", "SnmpGetParam", "SnmpGetBulkParam" and "SnmpSetParam" is the program folder.
The PLC data type "typeParamGetSet" , "typeParamGetBulk" and "typeSnmpBulkResponseData" is inserted into the respective PLC_Data type folder.</td></tr>
<tr><td>3.</td><td>From the templates a copy is created in the location of use.</td><td></td></tr>
<tr><td>4.</td><td>Open the OB1 organization block from your program folder and if required, create a new network.</td><td></td></tr>
<tr><td>5.</td><td>Select the "SnmpGet" program code and drag it via drag-and-drop into the network.</td><td></td></tr>
<tr><td>6.</td><td>The block requires an instance data block. The TIA portal automatically suggests a number and name. Confirm the dialog with OK.</td><td>
Call options
Single instance
Data block
Name: SNMP_GET_DB
Number: 11
○ Manual
● Automatic
The called function block saves its data in its own instance data block.
More...
OK Cancel
Interface
Name
Data type
Comment
Input
Block tit
Comment
Netw
Comme
&lt;???&gt;
SNMP_VAR
ERROR
STATUS

</td></tr>
<tr><td>7.</td><td>Proceed with "SnmpSet" as described in the previous step.</td><td></td></tr>
<tr><td>8.</td><td>Open DB "SnmpGetParam" or "SnmpSetParam" and enter your desired values (IP address, DeviceID and OID).</td><td>
SNMP_GET_PARAM
<table>
<tr><th></th><th>Name</th><th>Data type</th><th>Start value</th></tr>
<tr><td>1</td><td>Static</td><td></td><td></td></tr>
<tr><td>2</td><td>SNMP</td><td>Array[1..4] of "PARAM...</td><td></td></tr>
<tr><td>3</td><td>SNMP[1]</td><td>"PARAM_GET_SET"</td><td></td></tr>
<tr><td>4</td><td>IP_ADR</td><td>DWord</td><td>16#C0A80006</td></tr>
<tr><td>5</td><td>DEVICE_ID</td><td>HW_ANY</td><td>64</td></tr>
<tr><td>6</td><td>CON_ID</td><td>Word</td><td>W#16#0062</td></tr>
<tr><td>7</td><td>LOCAL_P</td><td>Word</td><td>W#16#07D0</td></tr>
<tr><td>8</td><td>OID</td><td>String[254]</td><td>'1.3.6.1.2.1.1.5....</td></tr>
<tr><td>9</td><td>Community</td><td>String[20]</td><td>'public'</td></tr>
<tr><td>10</td><td>VALUE_TYPE</td><td>Byte</td><td>16#0</td></tr>
<tr><td>11</td><td>VALUE_LEN</td><td>Byte</td><td>16#0</td></tr>
<tr><td>12</td><td>VALUE</td><td>Array[1..255] of Byte</td><td></td></tr>
<tr><td>13</td><td>SNMP[2]</td><td>"PARAM_GET_SET"</td><td></td></tr>
</table></td></tr>
</table>

No.	Action	Note
9.	When working with FB "SnmpSet", you also have to enter the type, length and value of the variable in "SnmpSetParam".	
10.	Reopen the OB1 and assign all required formal parameters of the blocks with values.	
11.	Select your PLC in the project overview and compile the entire project via the icon .	

SNMP_SET_PARAM

	Name	Data type	Start value
1	Static		
2	SET	Array[1..4] of "PARA...	
3	SET[1]	"PARAM_GET_SET"	
4	IP_ADR	DWord	16#C0A80006
5	DEVICE_ID	HW_ANY	64
6	CON_ID	Word	W#16#0062
7	LOCAL_P	Word	W#16#07D0
8	OID	String[254]	'1.3.6.1.2.1.1.5....
9	Community	String[20]	'private'
10	VALUE_TYPE	Byte	16#2
11	VALUE_LEN	Byte	16#B
12	VALUE	Array[1..255] of Byte	
13	VALUE[1]	Byte	16#4

%FB100
"SNMP_GET"

... — EN
#GET_2 — REQ
... — COM_RST
"SNMP_GET_PARAM".SNMP[2] — SNMP_VAR
DONE — #DONE_2
BUSY — #BUSY_GET_2
ERROR — #ERROR_GET_2
STATUS — #Status_2
ENO —

3.3 Downloading the blocks to the S7 PLC

Make sure that your PC/PG, the S7 PLC and the network components are in the same subnet.

Select your PLC in the project overview and load the entire project via the appropriate menu icon into your PLC.

4 Appendix

4.1 Service and support

Industry Online Support

Do you have any questions or need assistance?

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

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

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

Technical Support

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

SITRAIN – Training for Industry

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

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

Service offer

Our range of services includes the following:

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

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

Industry Online Support app

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

4.2 Links and literature

Table 4-1

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to this entry page of this application example https://support.industry.siemens.com/cs/ww/en/view/57249109
\3\	Private MIBs: SCALANCE X, SCALANCE W and SNMP OPC Profile http://support.automation.siemens.com/WW/view/en/22015045
\4\	Information to "Local_Device_id" http://support.automation.siemens.com/WW/view/en/51339682
\5\	Examples of using the SNMP blocks http://support.automation.siemens.com/WW/view/en/57249109

4.3 Change documentation

Table 4-2

Version	Date	Modifications
V1.0	17.04.2012	First version
V2.0	17.04.2014	Complete revision of the blocks; integration in TIA V12; blocks for S7-1500
V3.0	21.07.2016	Additional SNMP Blocks "SnmpGetBulk" and "SnmpGetNext"; Update to TIA V13 SP1
V4.0	07/2019	Update for TIA Portal V15.1