CODESYS STORE

SNMP Library

This library makes it possible to read device information of printers, routers etc. via the SNMP protocol. The library provides function blocks to send and receive SNMP messages. The package contains the SNMP library and example applications.

The library "SNMP Library" is now part of the product IIoT Libraries SL and is no longer available as single product.

Product description

Licensing:

Workstation License

This library makes it possible to read device information of printers, routers etc. via the SNMP protocol. The library provides function blocks to send and receive SNMP messages. The package contains the SNMP library and example applications.

Supported SNMP features:

- SNMP GET: Request a value.
- SNMP GET_NEXT: Request a value and get the OID from the next value.
- SNMP Agent: Enables devices to request values from the control (agent).
- SNMP TRAP: Send and receive TRAP/INFORM telegrams.
- SNMP SET: Set values via SNMP

The package `SNMPLibrary.package` contains following components:

- SNMPLibrary (Namespace: SNMP)
- Project with example applications
- CHM help
- Product datasheet

The library `SNMPLibrary` is divided into four categories (see folder `Function Blocks`):

`Agent`: Function blocks of the SNMP agent.
`GET`: Function block to request values.
`Trap`: Function blocks to send and receive TRAP messages.
`SET`: Function block to set SNMP values.

Function blocks of the category "Agent"

The function block SNMP_AGENT is the base of the SNMP agent. The function block receives GET and GET_NEXT messages and sends the value of the requested OID to the requesting device. Values can be registered under an OID by the function blocks SNMP_STRING and SNMP_DINT.

The example application `SNMPAgentExample` of the project `SNMPExample.project` shows how to use the components.

Function block SNMP_Agent:

Inputs:

`sOwnIP`: IP address of the control (own IP)
`uiPort`: SNMP port, default: 161, UDP
`xExecute`: Starts the agent

Outputs:

`xDone`: True, if the action has successfully completed
`xError`: True, if an error occurred

xBusy: True during the action is executed

Function block SNMP_STRING:

Function block to register a STRING value.

Inputs:

pSnmpAgent: Pointer to SNMP_Agent to register the value
sOID: The OID to request the value (e.g. 1.3.6.1.4.1.2001)
sValue: The value to request (STRING)

Function block SNMP_DINT:

Function block to register a DINT value.

Inputs:

pSnmpAgent: Pointer to SNMP_Agent to register the value
sOID: The OID to request the value (e.g. 1.3.6.1.4.1.2001)
diValue: The value to request (DINT)

Function blocks of the category "GET"

A SNMP request can be sent via the function block SNMP_GET_REQUEST.

The example applications SNMGetNextExample and SNMPTimerExample of the project SNMPExample.project show how to use the function block.

Function block SNMP_GET_REQUEST:

Inputs:

sHost: IP address of the device (also called agent)
sOwnIP: IP address of the control (own IP address)
asOIDs: OIDs to request
iNumberOfOIDs: Number of OIDs in asOIDs
eRequestType: GET or GET_NEXT
uiPeerPort: Port of the sender, default: 161, UDP
uiSendPort: Port of the receiver, default: 161, UDP
xExecute: Opens a UDP socket and starts the requests.
sCommunity: "Community" of the request.
xClosePeer: Rising edge closes the UDP socket.

Outputs:

aSNMPValues: Result of the request. The result is stored in an array of SNMPValue structures. Generally the result has two elements. The first element contains the requested OID (case GET) or the next OID (case GET_NEXT). The second element contains the value.
iSize: Size of the array aSNMPValues
abResponse: Result as byte array (BER encoded)
xDone: True, if the request was successful
xError: True, if an error occurred
eError: Error code, if an error occurred
xBusy: True during a request is executed

Function blocks of the category "SET"

SNMP values can be set via the function block SNMP_SET.

The example application SNMSetExample of the project SNMPExample.project shows how to use the function block.

Function block SNMP_SET:

Inputs:

sHost: IP address of the device (also called agent)
sOwnIP: IP address of the control (own IP address)
snmpVarBindings: Array of SNMPVarBinding. The structure SNMPVarBinding contains the OID and the value.

iNumberOfVarBindings: Number of SNMPVarBinding in snmpVarBindings
uiPeerPort: Port of the sender, default: 161, UDP
uiSendPort: Port of the receiver, default: 161, UDP
xExecute: Opens a UDP socket and starts the requests.
sCommunity: “Community” of the request.
xClosePeer: Rising edge closes the UDP socket.

Outputs:

aSNMPValues: Result of the request. The result is stored in an array of SNMPValue structures.
iSize: Size of the array aSNMPValues
abResponse: Result as byte array (BER encoded)
bySNMPErrorStatus: SNMP error status of the request
xDone: True, if the request was successful
xError: True, if an error occurred
eError: Error code, if an error occurred
xBusy: True during a request is executed

Function blocks of the category “Trap”

The category “Trap” contains function blocks to send and receive TRAP/INFORM messages.

The example applications SNMPTrapReceiver and SNMPTrapSender of the project SNMPExample.project show how to use the components.

Function block SNMP_TRAP_RECEIVER:

Function block to receive TRAP/INFORM messages.

Inputs:

sOwnIP: IP address of the control (own IP)
uiPort: SNMP port, default: 162, UDP
xExecute: Activates the receiver

Outputs:

xDone: True, if the action has successfully completed
xError: True, if an error occurred
xBusy: True during the action is executed
xReceived: True, if a trap was received
aSNMPValues: Values of the received TRAP
iSize: Number of received values.
sSenderIP: IP-Address of the sender
sEnterprise: Enterprise-OID of the sender
udiTimestamp: Timestamp of the sender

Function block SNMP_TRAP_SENDER:

Function block to send TRAP/INFORM messages.

Inputs:

sHost: IP address of the receiver
sOwnIP: IP address of the sender (own IP)
uiPeerPort: Port of the sender, default: 162, UDP
uiSendPort: Port of the receiver, default: 162, UDP
sEnterprise: OID of the event
snmpVarBindings: Array of SNMPVarBinding. The struct SNMPVarBinding contains the OID and the value.
iNumberOfVarBindings: Number of structures in snmpVarBindings.
sCommunity: “Community” of the request.
xClosePeer: Rising edge closes the UDP socket.
bGenericTrapType: Generic type, default: 6 (enterprise specific)
bSpecificTrapType: Specific type, default: 0
udiTimestamp: Timestamp, default: 0
eTrapType: Type of the request (V1_Trap, V2_Trap, Inform). Inform messages are confirmed by sending an acknowledgment to the sender.
uiTimeout: Timeout in milliseconds (only relevant for Inform messages).

`xExecute`: Opens a UDP socket and starts the requests.

Outputs:

`xDone`: True, if the action has successfully completed
`xError`: True, if an error occurred
`xBusy`: True during the action is executed

Structure SNMPValues

SNMP Values are stored in the structure `SNMPValue`:

`byType`: Data type of the value. The data types are defined in the global variable list.
`aValue`: Result as byte array (BER encoded)
`iLength`: Length of the array `aValue`
`psValue`: Pointer to the result as string
`pliValue`: Pointer to the result as `LINT` (only for integer types)

Structure “SNMPVarBinding”

The structure `SNMPVarBinding` contains an OID and the corresponding value.

`SNMPVarBinding`:

`oid`: Object Identifier
`value`: Value of the OID

Example SNMPExample.project / Application SNMPAgentExample

The example shows how to use the function block `SNMP_AGENT` and how to register values. The visualization displays the status of the agent *(see figure 1)*.

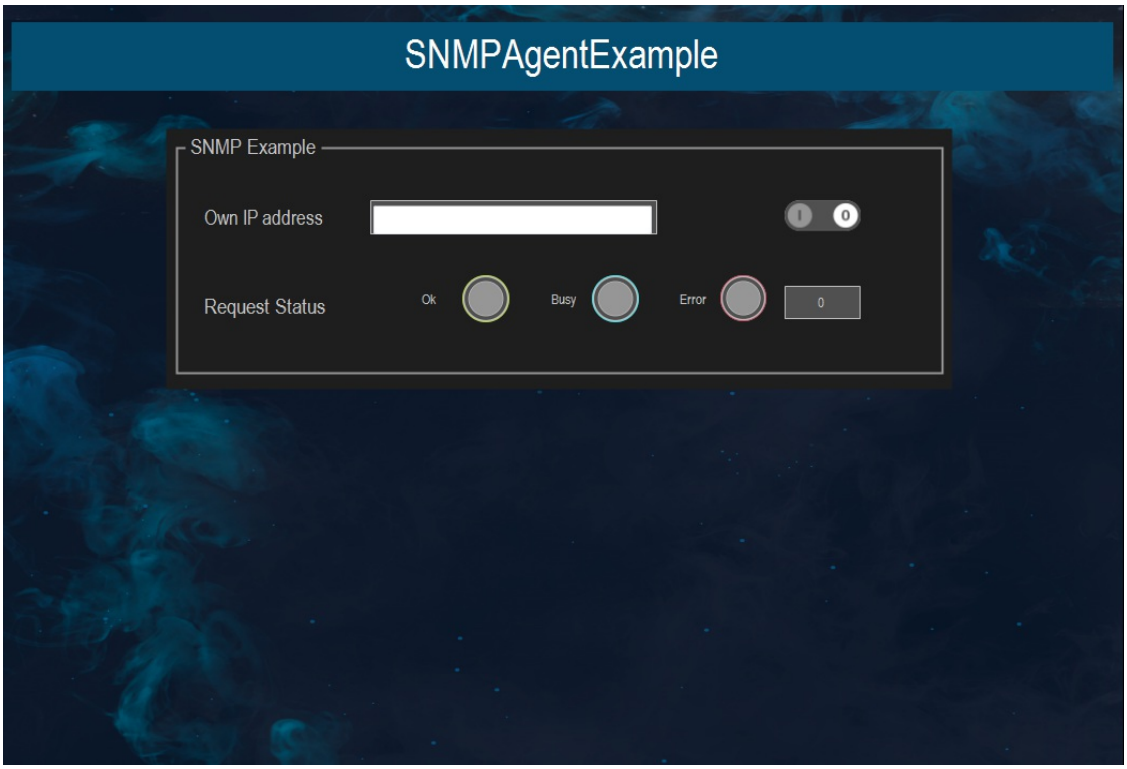

Figure 1: SNMPAgentExample

Example SNMPExample.project / Application SNMPGetNextExample

This example shows how to send a GET_NEXT request via the function block `SNMP_GET_REQUEST`. The result of the request will be displayed in the visualization *(see figure 2)*.

Please note that the value of an OID is stored in the next data set. The IP addresses and the OID can be configured via the visualization.

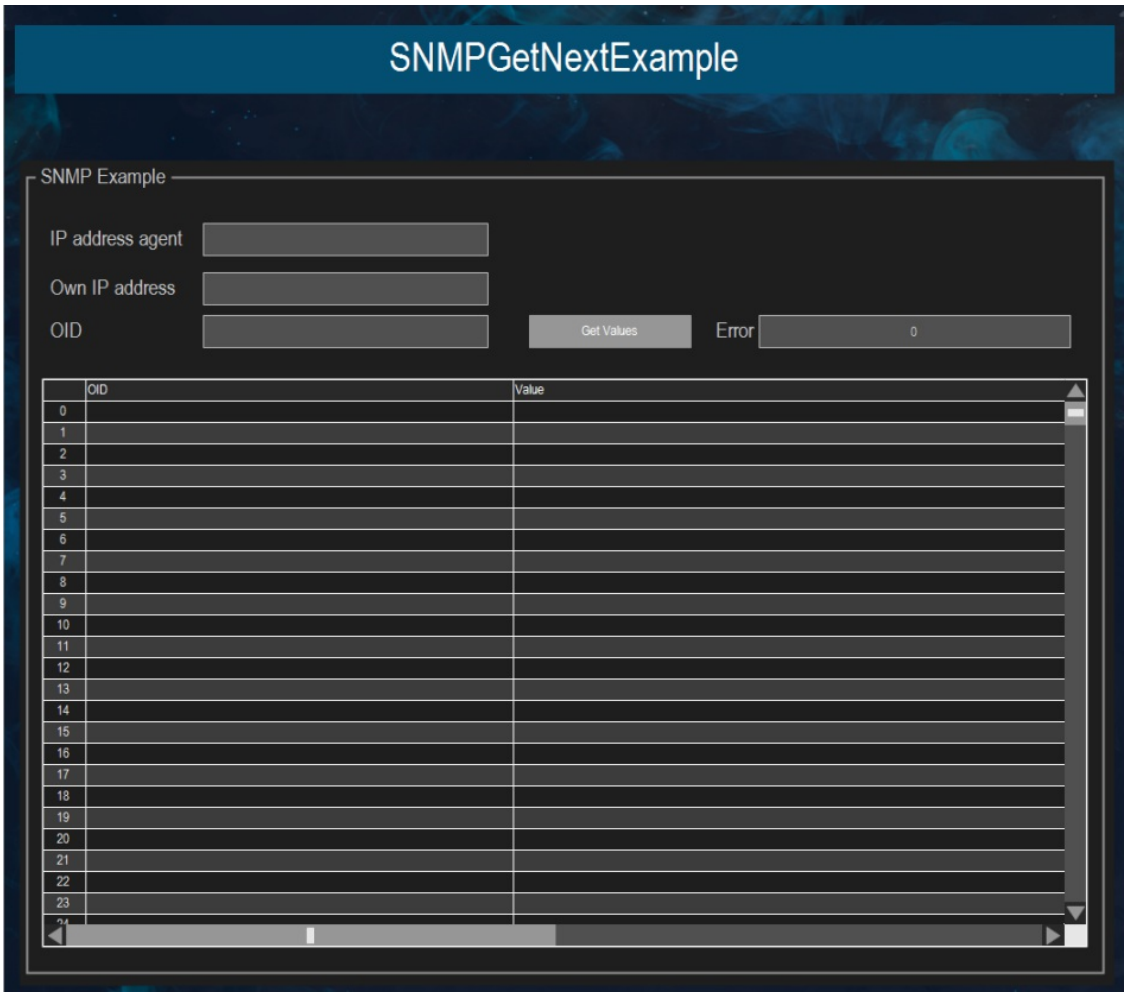

Figure 2: SNMPGetNextExample

Example SNMPExample.project / Applikation SNMPMutiGetExample

This example shows how to request multiple OID values in one request.

Example SNMPExample.project / Application SNMPTimerExample

This example shows how to send a GET request via the function block `SNMP_GET_REQUEST`. The result of the request will be displayed in the visualization *(see figure 3)*.

The IP addresses and the OIDs of the request are configured in the CSV-file `"c:\temp\SNMPCOnfig.csv"`. The first column contains the IP address of the device, the second column contains the OID (separator is semicolon). The last line must contain a line feed. The configuration and the values are displayed in the visualization. If the switch “Start timer” is on, the values will be requested every 20 seconds.

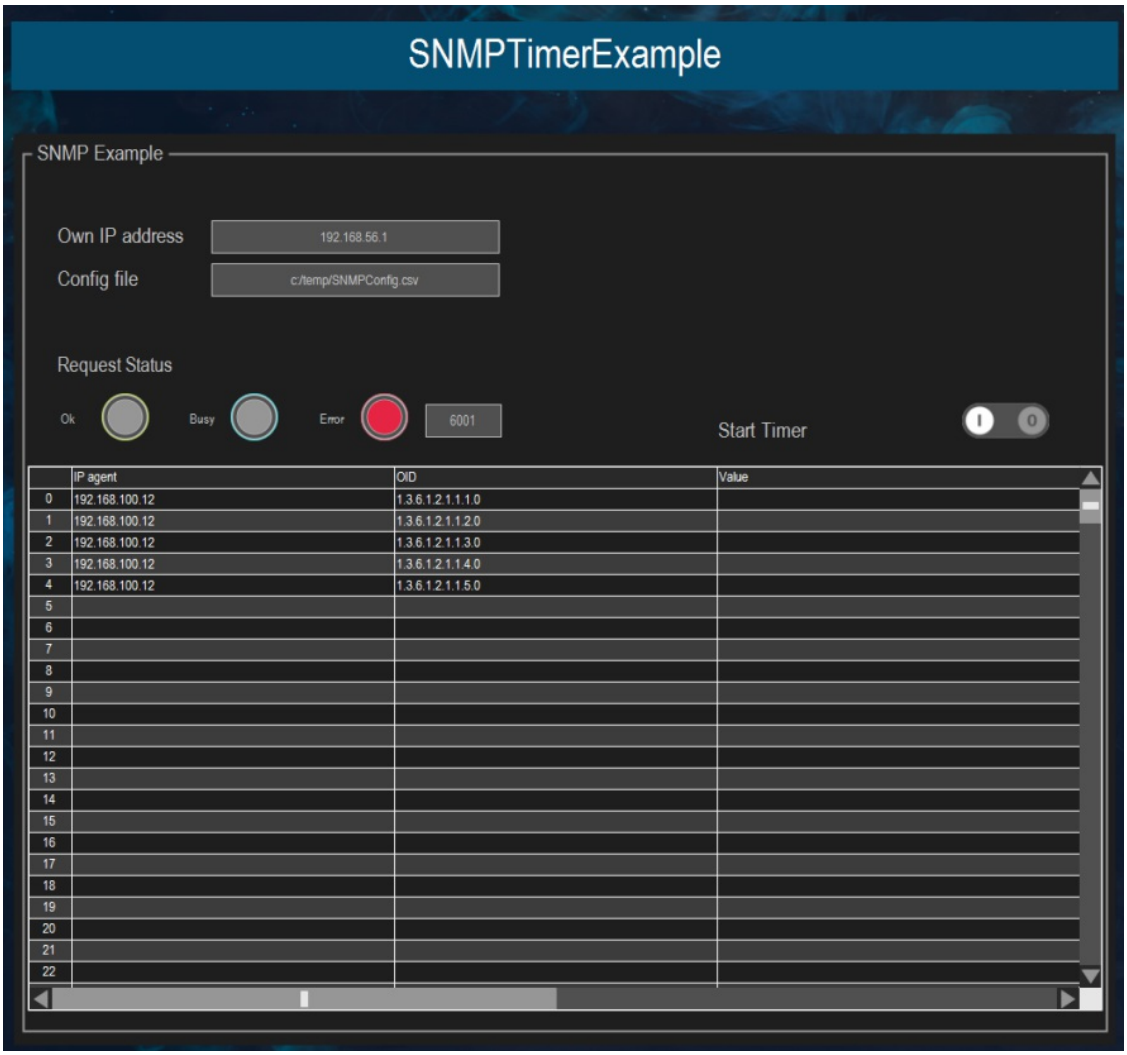

Figure 3: SNMPTimerExample

Example SNMPExample.project / Application SNMPSetExample

This example shows how to send a SET request via the function block `SNMP_SET` *(see figure 4)*.

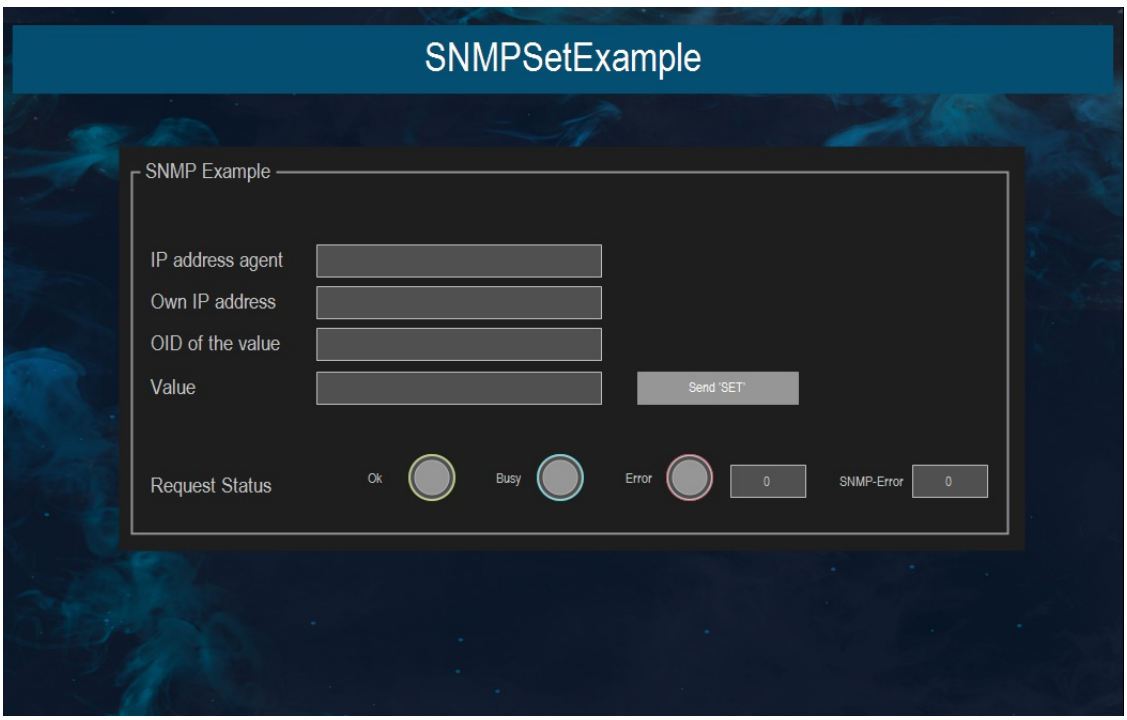

Figure 4: SNMPSetExample

Example SNMPExample.project / Application SNMPTrapReceiver

This example shows how to receive TRAP messages by the function block `SNMP_TRAP_RECEIVER`. The received values are displayed in a table *(see figure 5)*.

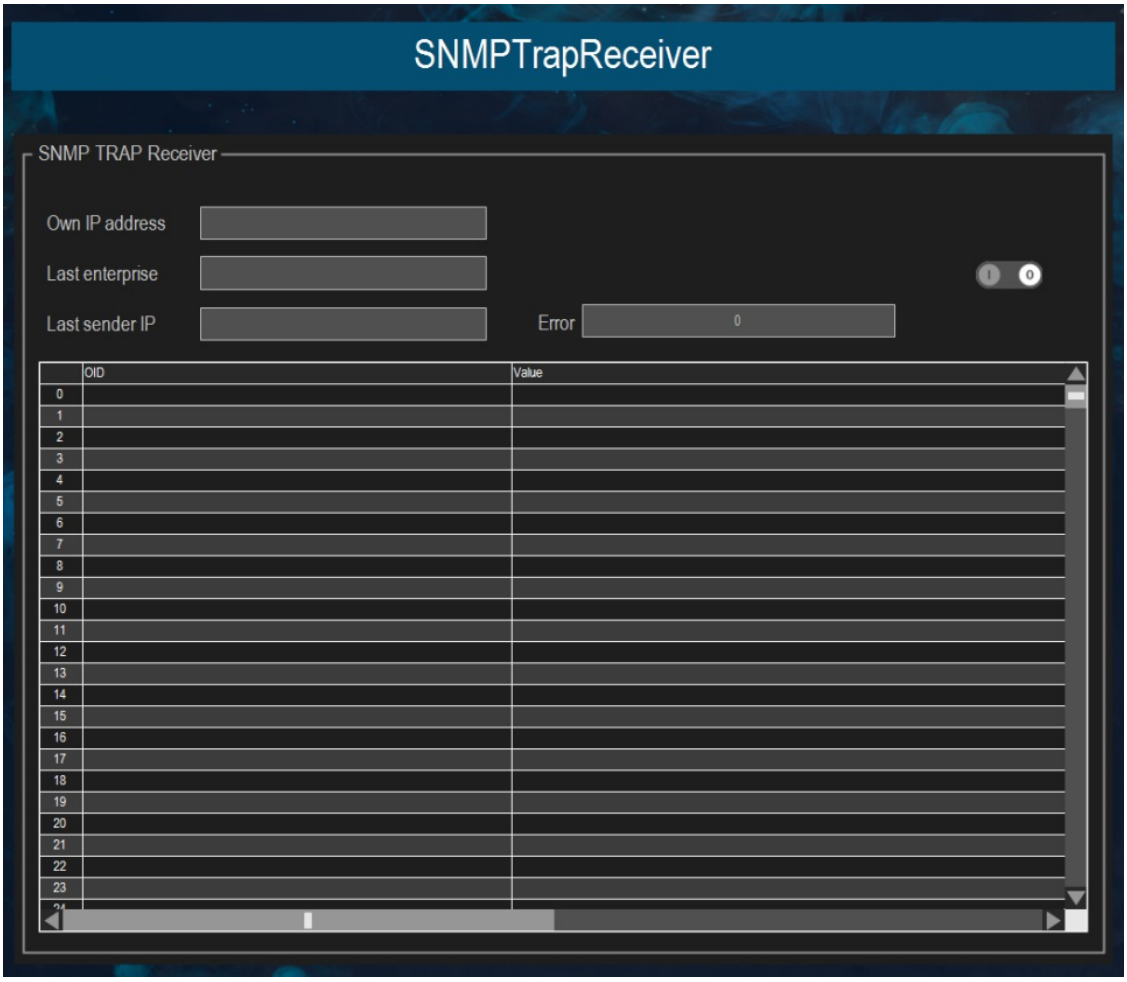

Figure 5: SNMPTrapReceiver

Example SNMPExample.project / Application SNMPTrapSender

The example shows how to send TRAP messages via the function block `SNMP_TRAP_SENDER` *(see figure 6)*.

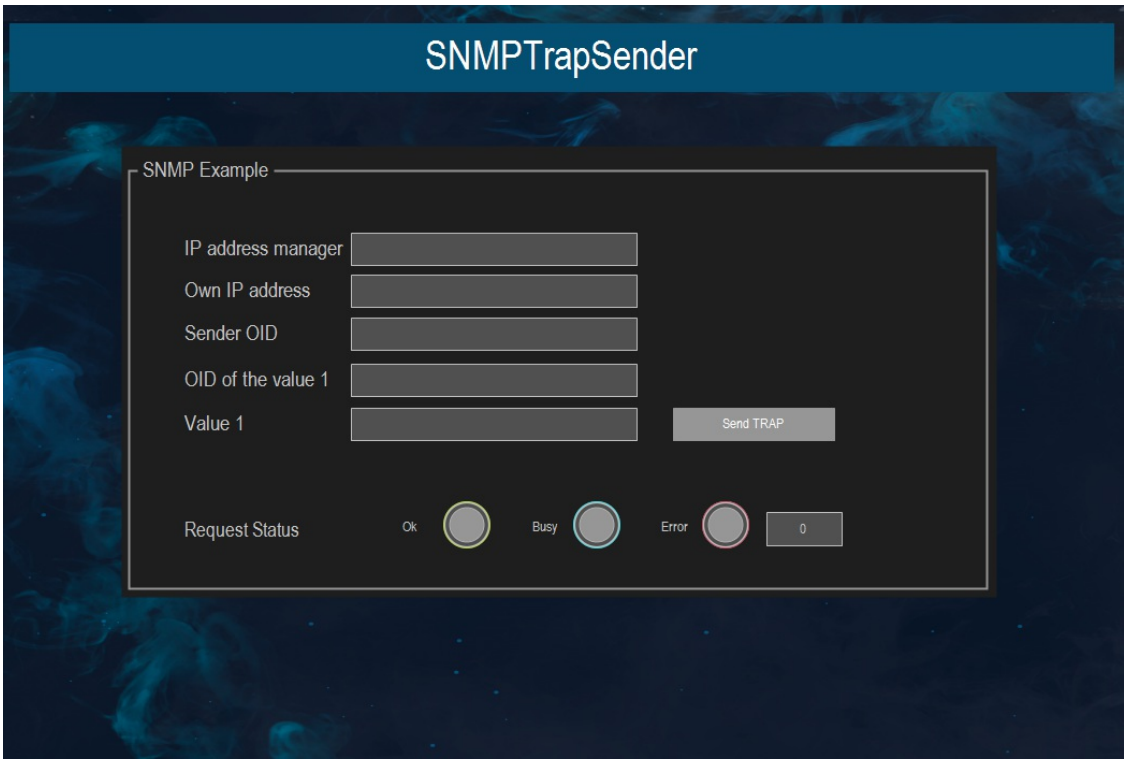

Figure 6: SNMPTrapSender

General information

Supplier:

CODESYS GmbH
Memminger Strasse 151
87439 Kempten
Germany

Support:

https://support.codesys.com

Item:
SNMP Library
Item number:
2111000005
Sales:

CODESYS Store
https://store.codesys.com

Included in delivery:
CODESYS Package with library and example project

System requirements and restrictions

Programming System	CODESYS Development System V3.5.14.0 or higher
Runtime System	CODESYS Control V3.5.9.0 or higher
Supported Platforms/ Devices	All Note: Use the project “Device Reader” to find out the supported features of your device. “Device Reader” is available for free in the CODESYS Store.
Additional Requirements	-
Restrictions	Supported SNMP versions: SNMP V1, SNMP V2c
Licensing	Workstation License: The license can be used on the workstation on which the CODESYS Development System is installed and executed. Licenses are activated on a software-based license container (soft container), which is permanently connected to the workstation. Alternatively the license can be stored on a CODESYS Key (USB-Dongle). By replugging the CODESYS Key, the license can be used on any other workstation.
Required Accessories	CODESYS Key for CODESYS < 3.5.14.0

Note: Not all CODESYS features are available in all territories. For more information on geographic restrictions, please contact sales@codesys.com.
Note: Technical specifications are subject to change. Errors and omissions excepted. The content of the current online version of this document applies.