IEC-60870-5-104

Slave Communication Protocol

Summary

1. GENERAL INFORMATION

1.1 Summary

Communication Driver Name: IEC8705104S

Current Version: 2016.2.1

Implementation DLL: T.ProtocolDriver.IEC8705104S.dll

Protocol: IEC-60870-5-104 Slave standard protocol

Interface: TCP/IP

Description: The IEC870504S protocol implements communication with client stations compatible with this protocol, acting as a slave station (server)..

Clients types supported: Any IED compatible with IEC-60870-5-104.

Communication block size: Maximum 253bytes;

Protocol Options: Counters for sending protocol control messages.

Multi-threading: User defined, five threads per node by default.

Max number of nodes: User defined

PC Hardware requirements: Standard PC Ethernet interface board;

PC Software requirements: Action.NET system.

1.2 Supported Object Data (ASDUs)

The protocol uses the same ASDUs defined for IEC-60870-5-101, as well as the same object data types. The major difference is that it is only targeted network orientation, using TCP/IP as the transport layer.

M_SP_NA: 1 - Single-point information ;

M_DP_NA: 3 - Double-point information ;

M_ST_NA: 5 - Step position;

M_BO_NA: 7 - Bitstring with 32 bits ;

M_ME_NA: 9 - Measured value, normalized ;

M_ME_NB: 11 - Measured value, scaled value ;

M_ME_NC: 13 - Measured value Float;

M_IT_NA: 15 - Integrated totals ;

C_SC_NA: 45 - Single command ;

C_DC_NA: 46 - Double command ;

C_RC_NA: 47 - Regulating step command ;

C_SE_NA: 48 - Set point command, normalized value ;

C_SE_NC: 50 - Set point command, 32 bits floating point ;

C_BO_NA: 51- Write 32 bits Bitstring;

Also, all the variant of the ASDUs above with 56 bits timestamp. The codes above are used when enrolling points, but the variant with date and timestamp obtained from the tags contained in memory at the moment are used when sendind unsolicited changes.

1.3 General Operation

This communication module implements The IEC-60870-5-104 protocol in Slave mode communicating with IEDs that use the same protocol in Master mode. Several parameterizations are available to accommodate different profiles of protocol implementations.

Slave has the following operating sequence:

- When starting (or after closing the Tcp-Ip socket), reaches a DISCONNECTED state (with the socket on a LISTENING state), waiting for a Tcp-Ip connection from a client;
- After accepting a client TCP/IP CONNECT, waits for a "Start of data transmission – STARTDT". Until the receiving of this message, stays blocked on an ESTABLISHED state, not answering or sending any messages;
- On receiving of the STARTDT message, answers with confirmation and becomes ready to receive and send any of the implemented messages, on a STARTED state;
- Sends unsolicited messages containing data of objects that had their field states changed;
- For each k (a user setting parameter) sent messages, or after a period of time without messages sent, waits for an acknowledgment message numbered with the sequence number of the last message sent. In case this message of not received, goes to the state UNCONF STOPPED;
- Always responds to Test Frame messages with confirmation if requested;

This module answers to variable reading – analog and digital -, event transmission and command execution requests. The implementation has the following characteristics:

- Responds to cyclical reading requests (general sampling) of digital points (simple or double) and analog;
- Sends unsolicited states change messages of digital points and analog measures. In events generation dead band and buffer time should be regarded;
- Uses 56 bits timestamp tag;
- Accepts commands for single or double digital points;
- Accepts Direct or Select Before Operate commands;
- Implements point quality analysis (QDS);

2. CHANNEL SETTINGS

2.1 Protocol Options

t0 - Timeout of Connection establishment(s) – Maximum waiting time, in seconds, for a client TCP/IP connection establishment into the LISTENNING port. After this time, this driver actively closes TCP/IP socket and restart it to LISTENNING state. Allowed values lies between 1 and 255.

t1 - Timeout of send or test APDUs(s) - Maximum acceptable time, in seconds, for the slave to send regular or test APDUs after receiving the START DT sending confirmation. Allowed values lie between 1 and 255.

t2 - Timeout for ack in case of no data(s)- Maximum waiting time, in seconds, for a pending acknowledgement before sending an acknowledgement for the last received message. A message with the sequence number of last received. Values from 1 to 255 are allowed. Besides, t2 must be inferior to t1.

t3 - Timeout for send test frames(s) - Maximum waiting time, in seconds, for the arrival of any information (in case of a TCP-IP connection already established) before sending a TEST-FR. The Values are allowed are from 1 to 255.

Maximum Changes to send a message -To improve communication module performance by sending changes in analog measurements, you can define this number as the maximum number of changes that must be accumulated to be sent in a single message, instead of sending a measure in each message. The number considered good is 30 measures, which is the default value.

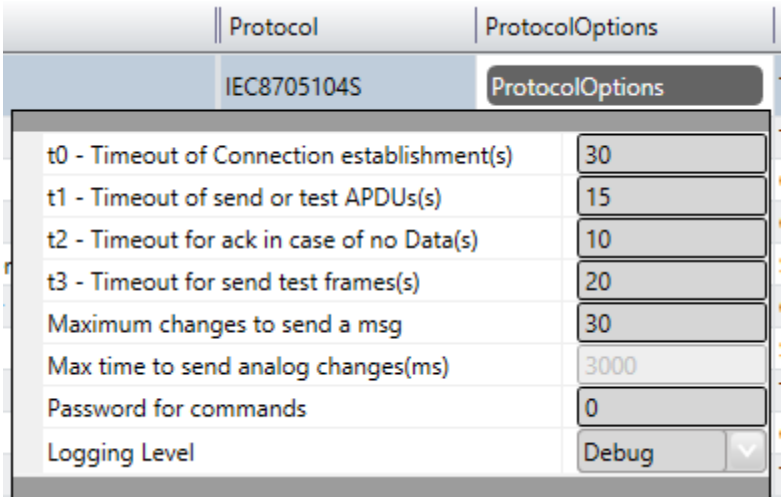
Max time to send analog changes (ms) - This time defines the maximum wait for sending a message with changes in analog measurements. If, since the start of accumulation of measurements for the same message, this time expires before the arrival of the number of measures defined above, this module will send the message with the measures that have already arrived. This time is set to 3 seconds.

Password for commands: In order to increase the security in sending commands, normally initiated only by a change in the state of a tag, it is possible to specify in the Client modules a password of up to 9 digits for the command. Here in this server module you must specify the password used by this Server module to generate the command for the Client module that will actually send the command to the field. This password must be the same as that used by the module sender of this command.

Logging Level – You can choose from this list the logging mode created by the communication module.

Logging level	Debug	All messages are registered in the LOG.
	Info	Only Info, Warning and Error messages are logged in the LOG.

	Warning	Only Warning and Error messages are registered in the LOG.
	Error	Only Error messages are registered in the LOG.



2.2 Settings

Listenning Port: Port number used for listening clients connections attempts. The standard defines port **2404**, by default. User can use custom port numbers.

NodeConnections: Defines the maximum number of parallel requests that could be sent to each node (asynchronous communication).

3. NODES SETTINGS

Each node is a server station (IED). User may set a single station per channel.

3.1 Parameters

CommonAddress- Application Layer address.

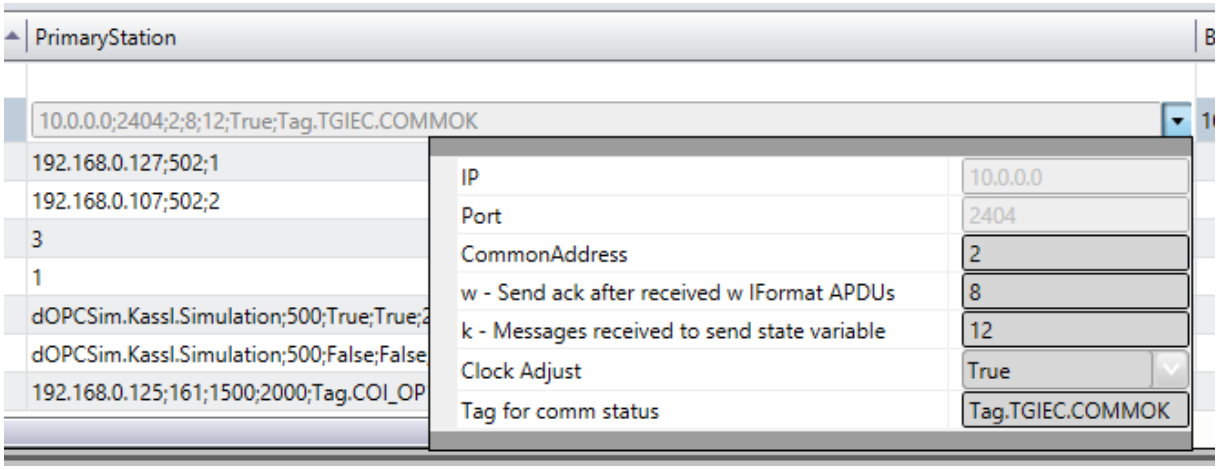
w – Send ack after received w IFormatAPDUs – Number of information messages sent spontaneously to client until it sends an "acknowledgment" with the sequence number of last message it received. Allowed values lies between 1 and 32767.

k – Messages received to send state variable – Maximum allowed number of pending acknowledgements before this slave stops sending new messages. The IEC standard recommends that **w** is, at most, two thirds of the **k** value. Values allowed are between 1 and 32767.

Clock Adjust – Can be set as "True" to adjust the clock on this server compute or "False" to make no adjustment. This module will adjust clock by changing the machine time to match the one that came on a synchronization message received. For this to be effective, the master IED must send a time that comes, for example, from a GPS.

Tag for Comm status - In this field can be indicated the name of an existing tag in the project to receive indication of success / failure in communication from a functional point of view. The module waits a maximum of Timeout milliseconds (defined in Protocol Options, as t2 above) for receiving a request from the client. In failure case, the module places the value of this tag in ZERO. In case of success puts the value in ONE.

Backup Station – The same settings made to the main station can be made to one backup workstation (alternative IED) if the there is one in the facility.



4. POINTS SETTINGS

4.1 General

The points can be input or output. The input points, points that are acquired by the protocol, have basically, two main parameters: the point type and address. The output points, used for remote controls have an additional address field parameter to specify an output operation. In a given IED or Node all addresses are unique no matter the kind of the point.

4.2 Points Type

ActionNet in Slave mode implements:

- Receiving date and time for synchronizing;
- General Interrogation request send;
- Sending of unsolicited information frames due to the data changes in remote IED.
- Time tag (56 bits length);
- Receiving single or double digital Point Commands;
- Select Before Operate Command;
- Point Value Quality analyzing (QDS);

The implemented point types are defined by the data objects defined in the IEC standard, presented below:

M_SP_NA: 1 - Single-point information

Simple binary input point, value 0 or 1. The variant sent by the server is with timetag M_SP_TB (= 30) when sent spontaneously, or M_SP_NA_1 itself in the answers to the General Interrogations. In registration only this type is used.

M_DP_NA: 3 - Double-point information;

Dual input point, which can assume states 0 to 3. Normally used in the signaling of states of switches and circuit breakers. The variant sent by the server is with timetag M_DP_TB (= 31) when sent spontaneously, or M_DP_NA itself in the answers to the General Inquiries. In registration only this type is used.

M_ST_NA: 5 - Step position;

Step or step value, in the range from -64 to +63, mainly used for transformer tap position or other position information. The variant sent by the server is with "timetag" when sent spontaneously, or the M_ST_NA itself in the answers to the General Inquiries. In registration only this type is used.

M_BO_NA: 7 - Bitstring with 32 bits;

Binary state information as a 32-bit string. No manipulation is done by the driver. The setting is treated as a long number. The variant sent by the server is with "timetag" when sending spontaneously M_BO_TB (= 33) or M_BO_NA itself in the answers to the General Inquiries. In registration only this type is used.

M_ME_NA: 9 - Measured value, normalized;

Standard analogue 16-bit signal measurement. Value between -32768 and + 32767. It is calculated as a real number between 0 and 1 before being assigned to the real-time tag. Scaling must be used to reproduce the value in the engineering unit. The variant sent by the server is the same without timetag M_ME_NA for both the changes and the replies to the General Queries. In registration only this type is used.

M_ME_NB: 11 - Measured value, scaled value;

Scalar analog measurement used for transmitting analogue quantities. Also 16-bit, value between -32768 and 32767. The variant sent by the server is the same without timetag M_ME_NB for both changes and replies to the General Queries. In registration only this type is used.

M_ME_NC: 13 - Measured value short floating point;

Analog measurement in fractional real number format, used for transmission of analogue quantities. The measurements are 32-bit fields in the IEEE STD 754 format, which implements floating-point numbers. The variant sent by the server is the same without timetag M_ME_NC for both changes and replies to the General Queries. In registration only this type is used.

M_IT_NA: 15 - Integrated totals;

Full analogue measurement with signal. Measures with 32 bits integer. The variant sent by the server is the same without timetag M_IT_NA for both changes and replies to the General Queries. In registration only this type is used.

C_SC_NA: 45 - Single command;

Command for single point (1 bit). Command details can be chosen by clicking the button to the right of the field. You can also enter directly the number which is the command code resulting from the choice of details. Each point will be statically parameterized in the POINTS table, so that one point must be set for opening and one for closing one-bit keys.

C_DC_NA: 46 - Double command;

Command for double point (2 bits). Command details can be chosen by clicking the button to the right of the field. You can also enter directly the number which is the command code resulting from the choice of details. Each point will be parameterized statically in the POINTS table, so that one point must be configured for opening and another for closing keys with two-bit signaling.

C_RC_NA: 47 - Regulating step command;

Command for step control normally used to send pulses up or down transformer "tap" switches. Command details can be chosen by clicking the button to the right of the field. You can also enter directly the number which is the command code resulting from the choice of details. Each point will be parameterized statically in the POINTS table, so that one point must be configured to go up and another to lower the position of the tap.

C_SE_NA: 48 - Set point command, normalized value;

For sending set points of 16 bits, normalized to IEDS that support this type of command.

The value to be sent is what is at the moment as the value of the "tag" whose address was sent in the command.

C_SE_NC: 50 - Set point command, short floating point value;

For sending 32-bit set points, in IEEE STD 764 floating point format, for IEDs that support this type of command. The value to be sent is what is at the moment as the value of the "tag" whose address was sent in the command.

C_BO_NA: 51- Write Bitstring 32-bit

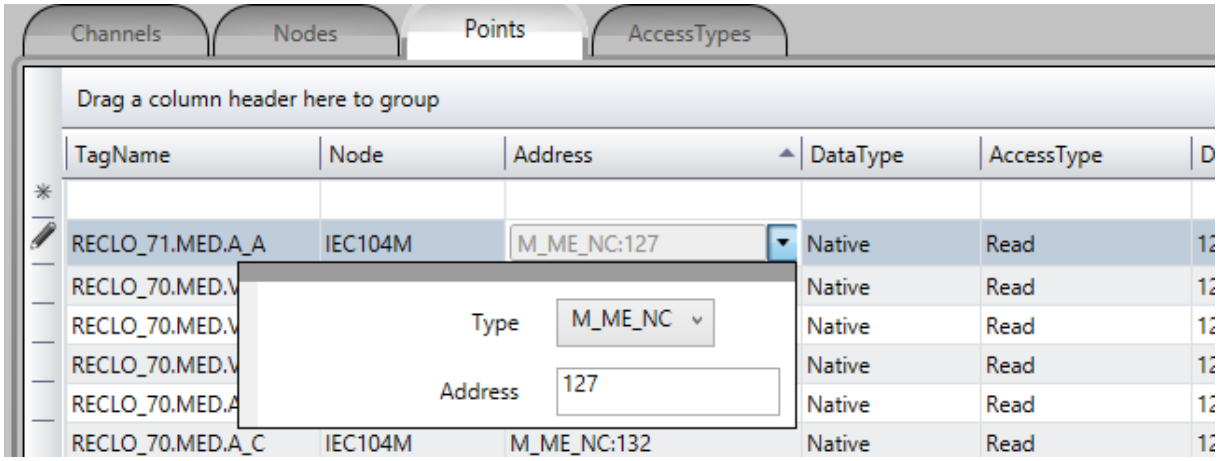
For writing the IED server a binary state information as a 32bit chain. No manipulation is done by the driver. The setting is treated as a long unsigned number. The value to be sent is what is at the moment as the value of the "tag" whose address was sent in the command. The type of the tag must be "long" or AnalogInt, this is a 32-bit integer.

4.3 Point Address

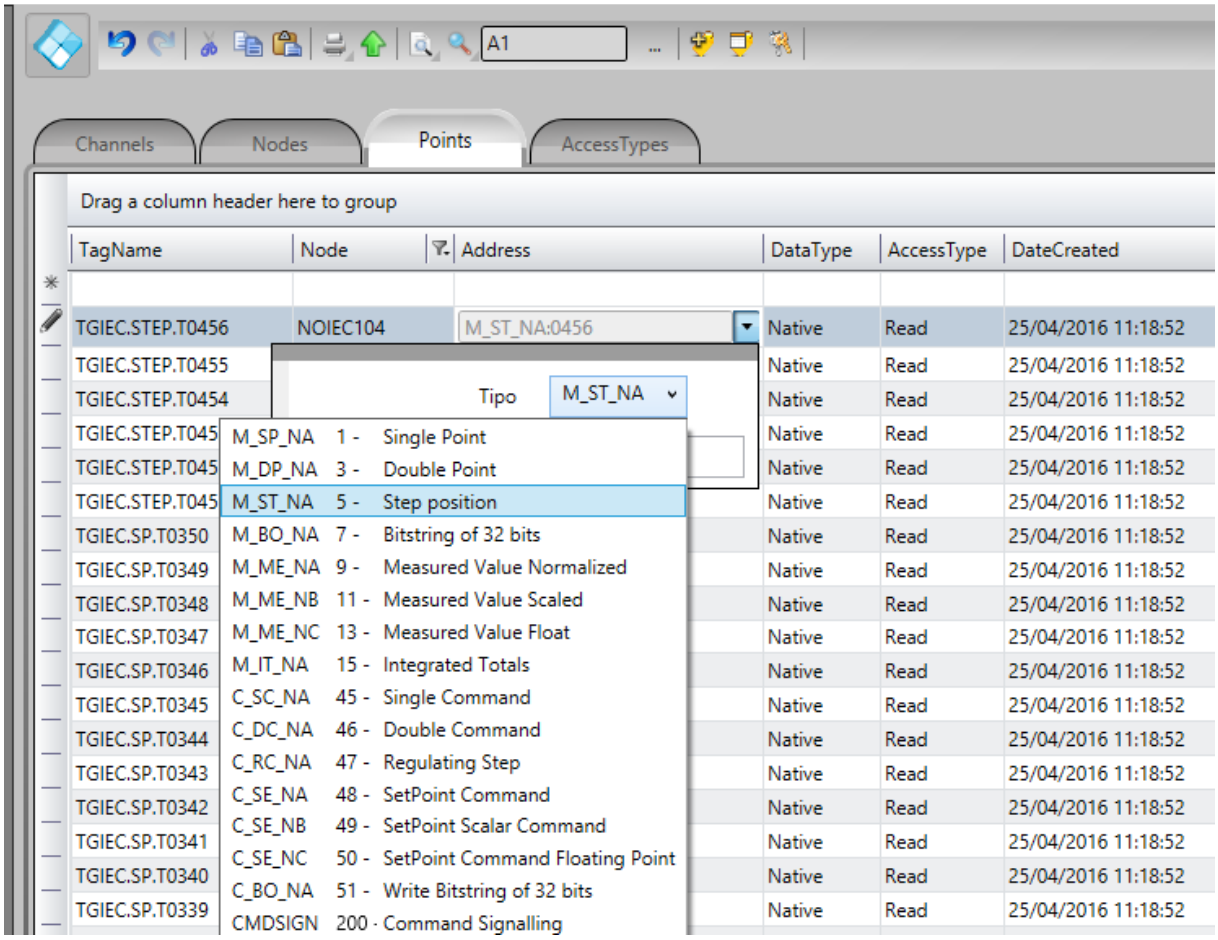
The completion of point addresses is done in the engineering environment, in **Edit> Devices> Points.**

The **Address** field to be filled in the point registration is what the standard calls "Information Object Address." It is a number of 3 bytes. For a given IED (node) it must be unique.

As the figure below, a click on the row of the address column opens a window to select the type and address of the point. A click on the type opens a window with all types of points supported:



To select **Type:**



4.4 Command Parameter

The command parameter is only for commands types and is a one-byte codes, which details what and how the IED should executes the command. In this implementation, as user register a point with type as command output, this field shows up to the user. If one already knowns the code, may just type it in the field. If not, click on the button to the right of the window to be displayed a dialog to choose the actions and details of commands.

4.4.1 Configuração do Parâmetro

The codes generated by choosing the items in the window parameter setting command are formed by calculating the sum of two parts (A and B), with the first part indicating action, and the second indicating details of the transaction, as defined below:

For Single Command C_SC_NA:

0 = Turn off (A)

1 = Turn on (A)

0 = No detail (B)

4 = Short Pulse (B)

8 = Long Pulse (B)

12 = Persistent Signal (B)

For Double Command C_DC_NA:

1= Turn off (A)

2= Turn on (A)

0 = No detail (B)

4 = Short Pulse (B)

8 = Long Pulse (B)

12= Persistent Signal (B)

For Voltage Regulation C_RC_NA:

1 = Down (A)

2 = Up (A)

0 = No detail (B)

4 = Short Pulse (B)

8 = Long Pulse (B)

12 = Persistent Signal (B)

The remaining options are a **Select** command - just select the device to be controlled, or an **Execute** command – which means sending the action command itself. In case of **Select**, we should add 128 to the code so far obtained by the sum of the parts A and B.

Example: code = 9, in a simple command means ***Long Pulse*** to ***Turn on*** the remote device;

4.4.2 Use of the parameter in the Server protocol (Slave)

When receiveing a master (cliente) command, the server does its execution according to the parameter sento n the message. **The parameter defined on the data base of the server is not used.**, and can be set in any way desired by the user.

The behaviour of the server when executing a command is as follows:

Select / Execute

SELECT – There will be no execution per se, ie. there will be no alteration on the server's memory. A message will be sent to the log (Trace), indicating the mode SELECT, if the output point was actually found on the server. In case the point doesn't exist, a message of error "POINT NOT FOUND" will appear on the log.

EXECUTE- The command will be executed normally and a message of it will appear on the log, indicating EXECUTE.

Details Options – B Option

0 – No detail – There will be only a "toggle" on the state of the command destiny point (if zero, goes to 1 and if 1 goes to zero), any that it may be the value of part A.

4 – Short Pulse – The value of part A will be placed into the destiny point, kept this way for 100 ms, and then restored to its original value.

8 – Long Pulse – The value of part A will be placed into the destiny point, kept this way for 1000 ms, and then restored to its original value.

12 – Persistent Signal – The value of part A will be placed into the destiny point and kept this way.

In order to set the Action.NET with output parameters, follow the procedure below:

(1) One click over the right border of the address shows three command parameters, in the command tab:

a. Type

b. Address

c. Command parameter

And the command options:

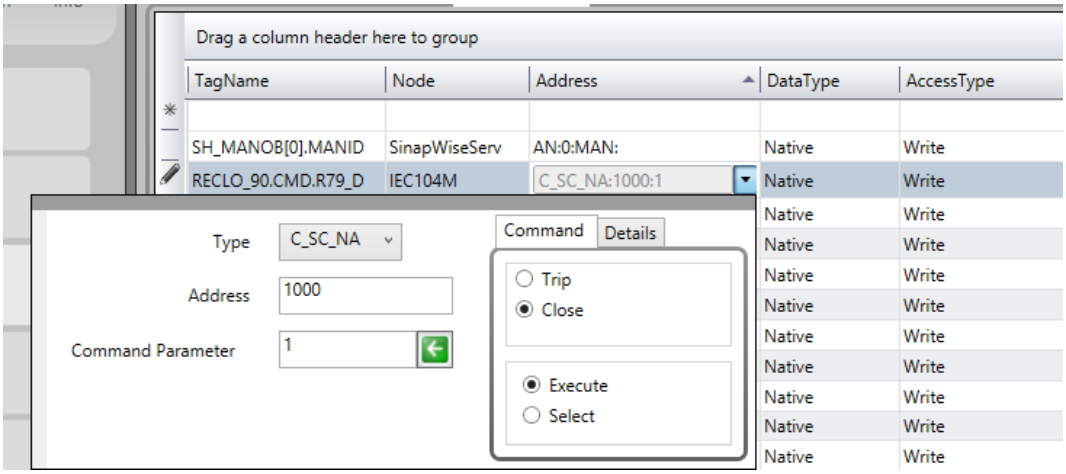
a. Trip

b. Close

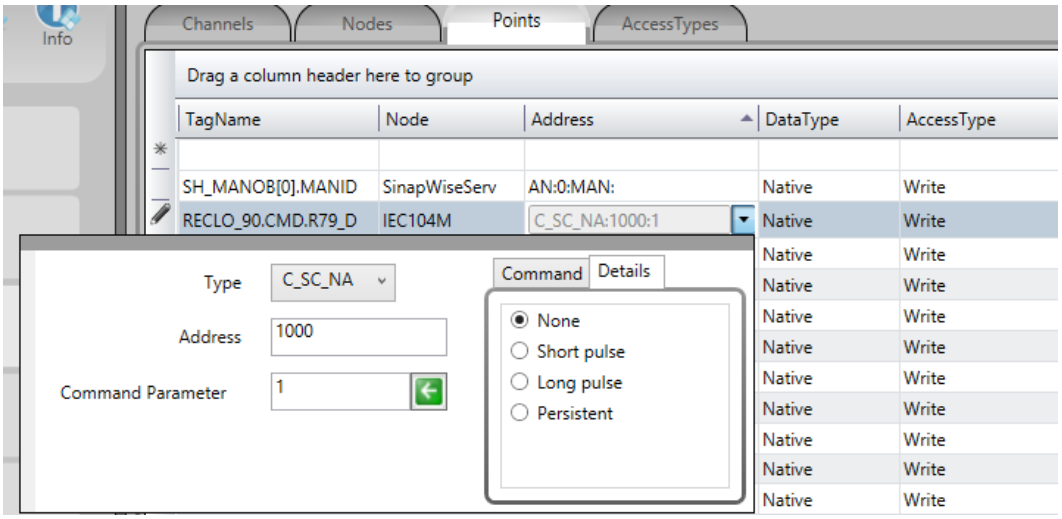
c. Execute

d. Select

(2) Select the desired options and when pressing the left arrow (), the binary value corresponding to the selection will be loaded in the command parameter:



If detailing the type of signal to be sent is necessary, before pressing the left arrow click on details and, as in the figure below, select the type of the output signal:



4.5 Access Type

Since it is a slave (server) communication module, there is need for a few specific characteristics in order to parameterize the **Access Type** field of the **Points** table:

For Reading type points:

M_SP_NA: 1 - Single-point information ;

M_DP_NA: 3 - Double-point information ;

M_ST_NA: 5 - Step position;

M_BO_NA: 7 - Bitstring with 32 bits ;

M_ME_NA: 9 - Measured value, normalized ;

M_ME_NB: 11 - Measured value, scaled value ;

M_ME_NC: 13 - Measured value Float;

M_IT_NA: 15 - Integrated totals ;

The Access Type must be defined with:

ReadOnStartup= On;

ReadPooling= Always;

ReadPoolongRate: 500 mili

WriteEnable = On

WriteEvent= Changed;

AccepUnsolictited = On;

For Command type points:

C_SC_NA: 45 - Single command ;

C_DC_NA: 46 - Double command ;

C_RC_NA: 47 - Regulating step command ;

C_SE_NA: 48 - Set point command, normalized value ;

C_SE_NC: 50 - Set point command, 32 bits floating point ;

C_BO_NA: 51- Write Bitstring de 32 bits

The Access Type must be defined with:

ReadPooling = Never;

WriteEnable = On

WriteEvent= Changed;