

Protocol Specific

- Modbus
- IEC 60870-5-101
- IEC 60870-5-103
- IEC 60870-5-104

Modbus

Modbus is a communications protocol originally published by Modicon (now Schneider Electric) in 1979 for use with its programmable logic controllers (PLCs). Simple and robust, it has since become a de facto standard communication protocol, and it is now a commonly available means of connecting industrial electronic devices.



Info about protocol

Address:

- **IP address** - every device in Ethernet have physical address (only for Ethernet)
- **Station address** - every slave (client) device have a logical address.
- **Function** - function type
- **Address** - information object address.

Telegram Structure

Modbus RTU

Request

| Name | Bytes | Function |
|-------------|-------|------------------|
| Station | 0 | Station address |
| Function | 1 | Function code |
| Address Hi | 2 | Starting address |
| Address Lo | 3 | |
| Quantity Hi | 4 | Quantity |
| Quantity Lo | 5 | |
| CRC | 6 | CRC check |

Response

| Name | Bytes | Function |
|----------|-------------------|-----------------|
| Station | 0 | Station address |
| Function | 1 | Function code |
| Bytes | 2 | Data bytes |
| Data | 3... | Data |
| CRC | Depending of data | CRC check |

Modbus RTU/ASCII

Request

| Name | Char | Function |
|------|------|----------|
|------|------|----------|

| | | |
|-------------|---|-----------------------------|
| Start | 0 | : 0x3A |
| Station | 1 | Station address |
| Function | 2 | Function code |
| Address Hi | 3 | Starting address |
| Address Lo | 4 | |
| Quantity Hi | 5 | Quantity |
| Quantity Lo | 6 | |
| LRC | 7 | LRC check |
| End | 8 | ASCII values of 0x0D & 0x0A |
| End | 9 | |

Response

| Name | Char | Function |
|----------|-------------------|-----------------------------|
| Start | 0 | : 0x3A |
| Station | 1 | Station address |
| Function | 2 | Function code |
| Bytes | 3 | Data bytes |
| Data | 4... | Data |
| LRC | Depending of data | LRC check |
| End | Depending of data | ASCII values of 0x0D & 0x0A |

Modbus TCP

Request

| Name | Bytes | Function |
|------------------------|-------|---|
| Transaction Identifier | 0 | For synchronization |
| Transaction Identifier | 1 | |
| Protocol Identifier | 2 | Zero for Modbus/TCP |
| Protocol Identifier | 3 | |
| Length | 4 | Number of remaining bytes in this frame |
| Length | 5 | |
| Station | 6 | Station address |
| Function | 7 | Function code |
| Address Hi | 8 | Starting address |
| Address Lo | 9 | |
| Quantity Hi | 10 | Quantity |
| Quantity Lo | 11 | |

Response

| Name | Bytes | Function |
|------|-------|----------|
|------|-------|----------|

| | | |
|------------------------|------|---|
| Transaction Identifier | 0 | For synchronization |
| Transaction Identifier | 1 | |
| Protocol Identifier | 2 | Zero for Modbus/TCP |
| Protocol Identifier | 3 | |
| Length | 4 | Number of remaining bytes in this frame |
| Length | 5 | |
| Station | 6 | Station address |
| Function | 7 | Function code |
| Bytes | 8 | Data bytes |
| Data | 9... | Data |

Functions

Standard MODBUS functions

| Dec | Description | Direction | Support |
|-----|----------------------------------|-----------|---------|
| 1 | Read Coils | Monitor | Yes |
| 2 | Read Discrete Inputs | Monitor | Yes |
| 3 | Read Holding Registers | Monitor | Yes |
| 4 | Read Input Registers | Monitor | Yes |
| 5 | Write Single Coil | Control | Yes |
| 6 | Write Single Register | Control | Yes |
| 7 | Read Exception Status | Monitor | No |
| 8 | Diagnostic | Monitor | No |
| 11 | Get Com Event Counter | Monitor | No |
| 12 | Get Com Event Log | Monitor | No |
| 15 | Write Multiple Coils | Control | Yes |
| 16 | Write Multiple Registers | Control | Yes |
| 17 | Report Slave ID | Monitor | No |
| 20 | Read File Record | Monitor | No |
| 21 | Write File Record | Control | No |
| 22 | Mask Write Register | Control | No |
| 23 | Read/Write Multiple Registers | Both | No |
| 24 | Read FIFO Queue | Monitor | No |
| 43 | Read Device Identification | Monitor | No |
| 43 | Encapsulated Interface Transport | Monitor | No |

Settings

| MASTER | | | |
|---|---------------|--|--|
| <div>Address</div> <div>Slave address: <input type="text" value="1"/></div> <div>Timeouts</div> <div>ScanRate(ms): <input type="text" value="1000"/></div> | | Modbus TCP | Modbus Serial (ASCII included) |
| | Slave address | Address of the device which data is read from | Address of the device which data is read from |
| | Scan Rate(ms) | Interval between requests to data | Interval between requests to data |
| SLAVE | | | |
| <div>Address</div> <div> <div>Select all</div> <div>Clear all</div> <div> <input checked="" type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 </div> </div> | Address | Select addresses of slaves to simulate | Select addresses of slaves to simulate |
| <div>Value</div> <div>Default value: <input type="text" value="0"/></div> | Value | Default value which slaves will return from all registers. | Default value which slaves will return from all registers. |

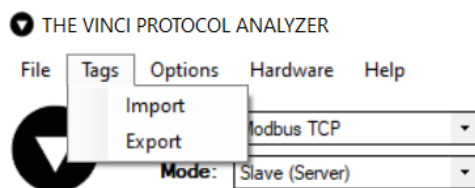
 Default values are overridden by created Tags.

Functions

Tags

This function allows user to created named points. After points created, user can send it manually or set reply checkbox to automatic reply.

- To export Tags to csv file: **Tags -> Export -> Save file dialog appear**
- To import Tags from csv file: **Tags -> Import -> Open file dialog appear**



Creating Tag

In Modbus protocol tags are mostly used for Slaves to simulate specific data registers, although they can be used in Master to write data to slave registers or format received data from Slave devices.

There are two ways of creating tags:

1. Add button in the tag menu.

The image shows a window titled 'Tags'. At the top is a text input field labeled 'Name'. Below this field are three buttons stacked vertically: 'Add', 'Edit', and 'Delete'. A black arrow points from the left towards the 'Add' button.

2. Left mouse button double click value in statistic tab.

Slave

If the simulation mode is **Slave** a modbus tag creation window should look like this and have these parameters.

- **Name** - user-friendly tag name.
- **Type** - the function to be used. *This means that if The Vinci software gets a request with the 03 function type and there is a tag created with that type and it matches the slave address and data address The Vinci software will respond to the request with the value that is set as the tag value.*
- **Slave** - Slave address of tag.
- **Address** - Address of slave register.
- **Format** - Format to store or read data in.
- **Value** - Value of the register.

The image shows a dialog box titled 'Tag'. It contains the following fields and controls:

- Name:** A text input field.
- Type:** A dropdown menu with 'Read Holding Registers (3)' selected.
- Slave:** A text input field with the value '1'.
- Address:** A text input field with the value '1'.
- Format:** A dropdown menu with 'Signed' selected.
- Value:** A text input field with the value '0'.
- At the bottom, there are two buttons: 'Save' and 'Cancel'.

Master

If the simulation mode is **Master** a modbus tag creation window should look like the one shown below and have these parameters.

In this case since this is the **Master** tag creation window it has two additional buttons that give the master the ability

to write data to the Slave device. For the selected type the buttons are **Write 5** and **Write 15** correlating to the modbus types.

In **Master** applications tags are mostly used to format data read from Slaves since data reading from devices is done using **Jobs**.

i For types 1 and 2 the buttons will be **Write 5** and **Write 15**.

i For types 3 and 4 the buttons will be **Write 6** and **Write 16**

- **Name** - user-friendly tag name.
- **Type** - the type of data.
- **Slave** - Slave address of tag.
- **Address** - Address of slave register.
- **Format** - Format to store or read data in.
- **Value** - Value of the register.

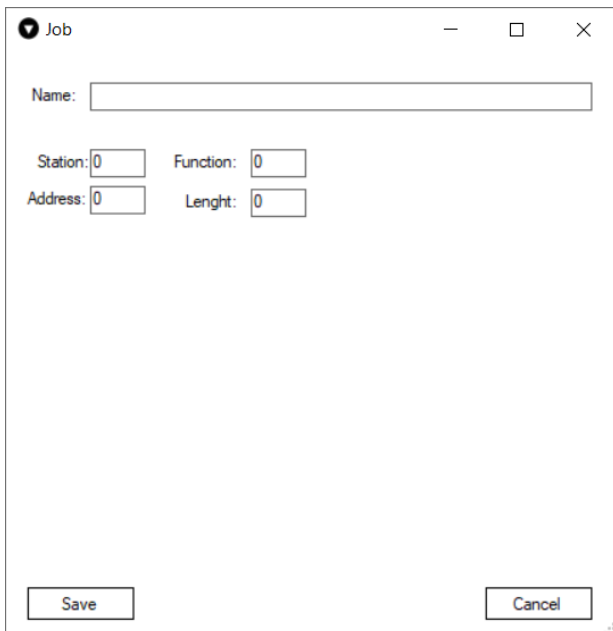
The image shows a 'Tag' configuration window. It has a title bar with a checkmark icon and the word 'Tag'. Inside, there is a 'Name:' label followed by a text input field. Below that is a 'Type:' label followed by a dropdown menu showing 'Read Coil Status (1)'. Then there are 'Slave:' and 'Address:' labels, each followed by a text input field containing the number '1'. Below these is a checkbox labeled 'Value'. At the bottom of the window, there are four buttons: 'Write 5', 'Write 15', 'Save', and 'Cancel'. The 'Save' and 'Cancel' buttons are at the very bottom, while 'Write 5' and 'Write 15' are above them.

Jobs

Jobs are only available in **Master** simulations. What jobs are meant for is reading data from the Modbus slave device. They can send requests for big chunks of data in a single request. Then the data that slave responds with can be formatted using tags.

Job has these parameters:

- **Name** - user-friendly job name.
- **Station** - Modbus **Slave address** to read data from.
- **Function** - the function to be used to read data.
- **Address** - slave address to begin reading data from.
- **Length** - how many bytes will be read.



A dialog box titled "Job" with a standard Windows window frame (minimize, maximize, close buttons). It contains the following fields:

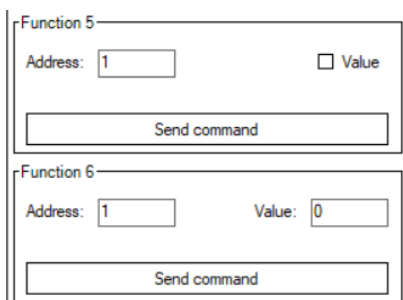
- Name:** A text input field.
- Station:** A numeric input field with the value "0".
- Function:** A numeric input field with the value "0".
- Address:** A numeric input field with the value "0".
- Lenght:** A numeric input field with the value "0".

At the bottom of the dialog are two buttons: "Save" on the left and "Cancel" on the right.

Command


Commands are only available in **Master** simulations. Commands are used to send data to **Slave** devices. They serve the same purpose as **Write 5** and **Write 6** buttons in tags.

Although, commands will send data to the **Slave address** configured in the **Settings** tab whereas tags will send the data to the **Slave address** configured in the tag configuration.



Two stacked panels for configuring Modbus commands:

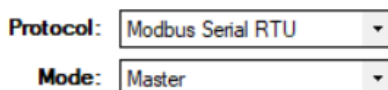
- Function 5:**
 - Address:** Input field with value "1".
 - ☐ **Value** checkbox.
 - Send command** button.
- Function 6:**
 - Address:** Input field with value "1".
 - Value:** Input field with value "0".
 - Send command** button.

 When using commands make sure to enter the desired Slave Address in the settings tab.

Setup

To setup an Modbus simulation it is fairly straightforward.

1. Select Modbus and the mode.

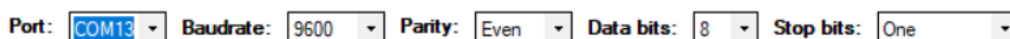


Configuration for Modbus protocol and mode:

- Protocol:** Dropdown menu showing "Modbus Serial RTU".
- Mode:** Dropdown menu showing "Master".

 There are three different Modbus modes: **Modbus TCP**, **Modbus Serial RTU**, **Modbus Serial ASCII**.

2. Select Serial Port settings according to your device specification.



Configuration for serial port settings:

- Port:** Dropdown menu showing "COM13".
- Baudrate:** Dropdown menu showing "9600".
- Parity:** Dropdown menu showing "Even".
- Data bits:** Dropdown menu showing "8".
- Stop bits:** Dropdown menu showing "One".

- 2.1 If **Modbus TCP** is used then the **IP** address and **Port** will have to be selected.

IP:

Port:

3. Select settings in the settings tab according to your device, preference and selected mode.

Address

Slave address:

Timeouts

ScanRate(ms):

4. Press the green **START** button and the simulation should start. If everything was done correctly The Vinci software should establish communication with the Modbus device which you can monitor in the console tab.

Protocol:

Mode:



IEC 60870-5-101

IEC 60870-5-101 is a protocol for power system monitoring and controlling. Mostly used for communication between substations and control centers over radio.



Info about protocol

Telegram Structure

Teleram format with fixed length

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|--------------------------|-----|------------|------------|---------------|---|---|---|
| 0 | Start byte | | | | | | | |
| 1 | RES | PRM | FCB ACD | FCV DFC | Function code | | | |
| 2 | Link address (1-2 bytes) | | | | | | | |
| 3 | Checksum | | | | | | | |
| 4 | Stop byte | | | | | | | |

Telegram format with variable length

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|--------------------------|-----|------------|------------|---------------|---|---|---|
| 0 | Start byte | | | | | | | |
| 1 | Length | | | | | | | |
| 2 | Length | | | | | | | |
| 4 | Start byte | | | | | | | |
| 5 | RES | PRM | FCB ACD | FCV DFC | Function code | | | |
| 6 | Link address (1-2 bytes) | | | | | | | |
| 7... | ASDU | | | | | | | |

- **RES** - Reserved
- **PRM** - 1 if master, 0 if slave

PRM = 1

- **FCB** - alternating bit for successive services per station
- **FCV** - (if FCV=1 FCB enabled)

PRM = 0

- **ACD** - access demand (if ACD=1 there are class 1 data)
- **DFC** - data flow control (if DFC=1 further messages may cause data overflow)

ASDU - Application Service Data Unit

Function Code

PRM=1

| Dec | Frame type | Service function | FCV |
|-----|---------------------------|---|-----|
| 0 | SEND/CONFIRM expected | Reset of remote link | 0 |
| 1 | SEND/CONFIRM expected | Reset of user process | 0 |
| 2 | SEND/CONFIRM expected | Reserved | - |
| 3 | SEND/CONFIRM expected | User data | 1 |
| 4 | SEND/REPLY expected | User data | 0 |
| 5 | | Reserved | - |
| 6 | | Reserved | - |
| 7 | | Reserved | - |
| 8 | REQUEST for access demand | Expected response specifies access demand | 0 |
| 9 | REQUEST/RESPOND expected | Request status of link | 0 |
| 10 | REQUEST/RESPOND expected | Request user data class 1 | 1 |
| 11 | REQUEST/RESPOND expected | Request user data class 2 | 1 |
| 12 | | Reserved | - |
| 13 | | Reserved | - |
| 14 | | Reserved | - |
| 15 | | Reserved | - |

PRM=0

| Dec | Frame type | Service function |
|-----|------------|---------------------------------------|
| 0 | CONFIRM | ACK: positive acknowledgment |
| 1 | CONFIRM | NACK: message not accepted, link busy |
| 2 | | Reserved |
| 3 | | Reserved |
| 4 | | Reserved |
| 5 | | Reserved |
| 6 | | Reserved |
| 7 | | Reserved |
| 8 | RESPOND | User data |
| 9 | RESPOND | Requested data not available |
| 10 | | Reserved |
| 11 | RESPOND | Status of link |
| 12 | | Reserved |

| | | |
|----|--|----------|
| 13 | | Reserved |
| 14 | | Reserved |
| 15 | | Reserved |

Type identification

Standard IEC 60870-5-101 data types[1-255]

- [1-127] - standard definition
- [128-135] - reserved for routing of messages
- [136-255] - for special use

| Dec | Type | Description | Direction | Support |
|---------------------|-----------|---|-----------|---------|
| Process information | | | | |
| 1 | M_SP_NA_1 | Single-point information | Monitor | Yes |
| 2 | M_SP_TA_1 | Single-point information with time tag | Monitor | Yes |
| 3 | M_DP_NA_1 | Double-point information | Monitor | Yes |
| 4 | M_DP_TA_1 | Double-point information with time tag | Monitor | Yes |
| 5 | M_ST_NA_1 | Step position information | Monitor | Yes |
| 6 | M_ST_TA_1 | Step position information with time tag | Monitor | Yes |
| 7 | M_BO_NA_1 | Bit string of 32 bit | Monitor | Yes |
| 8 | M_BO_TA_1 | Bit string of 32 bit with time tag | Monitor | Yes |
| 9 | M_ME_NA_1 | Measured value, normalized value | Monitor | Yes |
| 10 | M_ME_TA_1 | Measured value, normalized value with time tag | Monitor | Yes |
| 11 | M_ME_NB_1 | Measured value, scaled value | Monitor | Yes |
| 12 | M_ME_TB_1 | Measured value, scaled value wit time tag | Monitor | Yes |
| 13 | M_ME_NC_1 | Measured value, short floating point number | Monitor | Yes |
| 14 | M_ME_TC_1 | Measured value, short floating point number with time tag | Monitor | Yes |
| 15 | M_IT_NA_1 | Integrated totals | Monitor | Yes |
| 16 | M_IT_TA_1 | Integrated totals with time tag | Monitor | Yes |
| 17 | M_EP_TA_1 | Event of protection equipment with time tag | Monitor | Yes |
| 18 | M_EP_TB_1 | Packed start events of protection equipment with time tag | Monitor | Yes |
| 19 | M_EP_TC_1 | Packed output circuit information of protection equipment with time tag | Monitor | Yes |
| 20 | M_PS_NA_1 | Packed single point information with status change detection | Monitor | Yes |
| 21 | M_ME_ND_1 | Measured value, normalized value without quality descriptor | Monitor | Yes |
| 30 | M_SP_TB_1 | Single-point information with time tag CP56Time2a | Monitor | Yes |

| | | | | |
|---------------------------|-----------|--|---------|-----|
| 31 | M_DP_TB_1 | Double-point information with time tag CP56Time2a | Monitor | Yes |
| 32 | M_ST_TB_1 | Step position information with time tag CP56Time2a | Monitor | Yes |
| 33 | M_BO_TB_1 | Bit string of 32 bit with time tag CP56Time2a | Monitor | Yes |
| 34 | M_ME_TD_1 | Measured value, normalized value with time tag CP56Time2a | Monitor | Yes |
| 35 | M_ME_TE_1 | Measured value, scaled value with time tag CP56Time2a | Monitor | Yes |
| 36 | M_ME_TF_1 | Measured value, short floating point number with time tag CP56Time2a | Monitor | Yes |
| 37 | M_IT_TB_1 | Integrated totals with time tag CP56Time2a | Monitor | Yes |
| 38 | M_EP_TD_1 | Event of protection equipment with time tag CP56Time2a | Monitor | Yes |
| 39 | M_EP_TE_1 | Packed start events of protection equipment with time tag CP56Time2a | Monitor | Yes |
| 40 | M_EP_TF_1 | Packed output circuit information of protection equipment with time tag CP56Time2a | Monitor | Yes |
| 45 | C_SC_NA_1 | Single command | Control | Yes |
| 46 | C_DC_NA_1 | Double command | Control | Yes |
| 47 | C_RC_NA_1 | Regulating step command | Control | Yes |
| 48 | C_SE_NA_1 | Set-point Command, normalized value | Control | Yes |
| 49 | C_SE_NB_1 | Set-point Command, scaled value | Control | Yes |
| 50 | C_SE_NC_1 | Set-point Command, short floating point number | Control | Yes |
| 51 | C_BO_NA_1 | Bit string 32 bit command | Control | Yes |
| 58 | C_SC_TA_1 | Single command with time tag CP56Time2a | Control | Yes |
| 59 | C_DC_TA_1 | Double command with time tag CP56Time2a | Control | Yes |
| 60 | C_RC_TA_1 | Regulating step command with time tag CP56Time2a | Control | Yes |
| 61 | C_SE_TA_1 | Measured value, normalized value command with time tag CP56Time2a | Control | Yes |
| 62 | C_SE_TB_1 | Measured value, scaled value command with time tag CP56Time2a | Control | Yes |
| 63 | C_SE_TC_1 | Measured value, short floating point number command with time tag CP56Time2a | Control | Yes |
| 64 | C_BO_TA_1 | Bit string of 32 bit command with time tag CP56Time2a | Control | Yes |
| System information | | | | |
| 70 | M_EI_NA_1 | End of Initialization | Monitor | Yes |
| 100 | C_IC_NA_1 | Interrogation command | Control | Yes |
| 101 | C_CI_NA_1 | Counter interrogation command | Control | Yes |
| 102 | C_RD_NA_1 | Read command | Control | Yes |
| 103 | C_CS_NA_1 | Clock synchronization command | Control | Yes |
| 104 | C_TS_NA_1 | Test command | Control | Yes |
| 105 | C_RP_NA_1 | Reset process command | Control | Yes |

| | | | | |
|----------------------|-----------|---|---------------|----|
| 106 | C_CD_NA_1 | Delay acquisition command | Control | No |
| 107 | C_TS_TA_1 | Test command with time tag CP56Time2a | Control | No |
| Parameter | | | | |
| 110 | P_ME_NA_1 | Parameter of measured values, normalized value | Control | No |
| 111 | P_ME_NB_1 | Parameter of measured values, scaled value | Control | No |
| 112 | P_ME_NC_1 | Parameter of measured values, short floating point number | Control | No |
| 113 | P_AC_NA_1 | Parameter activation | Control | No |
| File transfer | | | | |
| 120 | F_FR_NA_1 | File ready | File transfer | No |
| 121 | F_SR_NA_1 | Section ready | File transfer | No |
| 122 | F_SC_NA_1 | Call directory, select file, call file, call section | File transfer | No |
| 123 | F_LS_NA_1 | Last section, last segment | File transfer | No |
| 124 | F_FA_NA_1 | ACK file, ACK section | File transfer | No |
| 125 | F_SG_NA_1 | Segment | File transfer | No |
| 126 | F_DR_TA_1 | Directory | File transfer | No |

Cause of transmission

Standard IEC 60870-5-101 cause of transmission [0-63]

| Dec | Description |
|-----|---|
| 1 | Periodic, cyclic |
| 2 | Background interrogation |
| 3 | Spontaneous |
| 4 | Initialized |
| 5 | Interrogation or interrogated |
| 6 | Activation |
| 7 | Confirmation activation |
| 8 | Deactivation |
| 9 | Confirmation deactivation |
| 10 | Termination activation |
| 11 | Return information caused by a remote command |
| 12 | Return information caused by a local command |
| 13 | File transfer |
| 20 | Interrogated by general interrogation |
| 21 | Interrogated by interrogation group 1 |
| 22 | Interrogated by interrogation group 2 |
| 23 | Interrogated by interrogation group 3 |
| 24 | Interrogated by interrogation group 4 |
| 25 | Interrogated by interrogation group 5 |
| 26 | Interrogated by interrogation group 6 |
| 27 | Interrogated by interrogation group 7 |
| 28 | Interrogated by interrogation group 8 |
| 29 | Interrogated by interrogation group 9 |
| 30 | Interrogated by interrogation group 10 |
| 31 | Interrogated by interrogation group 11 |
| 32 | Interrogated by interrogation group 12 |

| | |
|----|---|
| 33 | Interrogated by interrogation group 13 |
| 34 | Interrogated by interrogation group 14 |
| 35 | Interrogated by interrogation group 15 |
| 36 | Interrogated by interrogation group 16 |
| 37 | Interrogated by counter general interrogation |
| 38 | Interrogated by interrogation counter group 1 |
| 39 | Interrogated by interrogation counter group 2 |
| 40 | Interrogated by interrogation counter group 3 |
| 41 | Interrogated by interrogation counter group 4 |
| 44 | Type Identification unknown |
| 45 | Cause unknown |
| 46 | ASDU address unknown |
| 47 | Information object address unknown |

Settings

Structure

| | Monitor | Master | Slave |
|--------------------|--------------------|--------------------|--------------------|
| LINK size in bytes | LINK size in bytes | LINK size in bytes | LINK size in bytes |
| COT size in bytes | COT size in bytes | COT size in bytes | COT size in bytes |
| ASDU size in bytes | ASDU size in bytes | ASDU size in bytes | ASDU size in bytes |
| IOA size in bytes | IOA size in bytes | IOA size in bytes | IOA size in bytes |

Address

| | Monitor | Master | Slave |
|--------------|----------|-----------------------|--------------------|
| Link address | Not used | Remote device address | Own system address |

Timeouts (ms)

| | Monitor | Master | Slave |
|-------------------|------------------------------------|------------------------------------|------------------------------------|
| Reading data | Waiting data in serial port buffer | Waiting data in serial port buffer | Waiting data in serial port buffer |
| Pause before send | Not used | Pause before send data | Pause before send data |

Parameters

| | Monitor | Master | Slave |
|------------------|----------|----------|--|
| Send End of ini. | Not used | Not used | Send end of initialization TI 70 (M_EI_NA_1) |

| | | | | |
|--|-----------------------------------|----------|----------|---|
| | Auto ack. control commands | Not used | Not used | Auto acknowledge system commands (TI: 100, 103) |
| | Auto ack. system commands | Not used | Not used | Auto acknowledge commands |

System

For all system functions user can set custom address:

APDU

ASDU:
☐ Test

Originator:

General Interrogation

This function will send telegram Type-identification = 100 (C_IC_NA_1)

General interrogation

Send

QOI:

QOI - qualifier of interrogation [0...255]

- 20 - Station interrogation
- 21 - Interrogation of group 1
- 22 - Interrogation of group 2
- 23 - Interrogation of group 3
- 24 - Interrogation of group 4
- 25 - Interrogation of group 5
- 26 - Interrogation of group 6
- 27 - Interrogation of group 7
- 28 - Interrogation of group 8
- 29 - Interrogation of group 9
- 30 - Interrogation of group 10
- 31 - Interrogation of group 11
- 32 - Interrogation of group 12
- 33 - Interrogation of group 13
- 34 - Interrogation of group 14
- 35 - Interrogation of group 15
- 36 - Interrogation of group 16

Counter Interrogation

This function will send telegram Type-identification = 101 (C_CI_NA_1)

Counter interrogation

Send

FRZ:

RQT:

FRZ - freeze[0..3]

- 0 - Station interrogation
- 1 - Interrogation of group 1
- 2 - Interrogation of group 2
- 3 - Interrogation of group 3

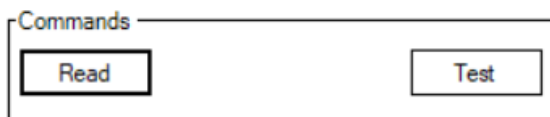
RQT - request[0..63]

- 1 - Counter group 1
- 2 - Counter group 2
- 3 - Counter group 3
- 4 - Counter group 3
- 5 - General request

Commands

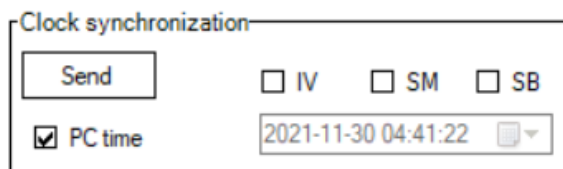
Read command will send telegram Type-identification = 102 (C_RD_NA_1)

Test command will send telegram Type-identification = 104 (C_TS_NB_1)

A dialog box titled "Commands" with a light blue header. It contains two buttons: "Read" and "Test".

Clock synchronization

This function will send telegram Type-identification = 103 (C_CS_NA_1)

A dialog box titled "Clock synchronization" with a light blue header. It contains a "Send" button, three checkboxes labeled "IV", "SM", and "SB", a checked checkbox labeled "PC time", and a text field showing the date and time "2021-11-30 04:41:22" with a dropdown arrow.

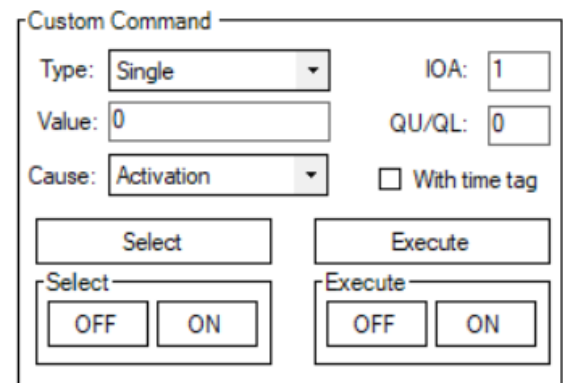
If "PC time" checkbox is checked, then the PC time will be sent. If it's not checked user can set time manually.

Time tag status bits:

- **IV** - invalid time
- **SM** - Summer/Winter
- **SB** - Substitute

Custom Commands

This function allows user to send commands to the slave device.

A dialog box titled "Custom Command" with a light blue header. It contains several fields: "Type" (dropdown menu set to "Single"), "IOA" (text field with "1"), "Value" (text field with "0"), "QU/QL" (text field with "0"), "Cause" (dropdown menu set to "Activation"), and a checkbox "With time tag". Below these are two buttons: "Select" and "Execute". Under "Select" are two buttons: "OFF" and "ON". Under "Execute" are two buttons: "OFF" and "ON".

Tags

This function allows user to created named points. After points created user can send it manually or set reply checkbox to automatic reply.

- To export Tags to csv file: **Tags -> Export -> Save file dialog appear**
- To import Tags from csv file: **Tags -> Import -> Open file dialog appear**

There are two ways of creating tags:

1. Create tag button.
2. Double click a signal with the left mouse button in the statistic tab.

Main parameters:

- **Name** - user-friendly tag name
- **Asdu** - Identifier of the device
- **IoA** - Identifier of values from the device.
- **Type** - the type of value.

Here is an example image of the tag window with the **M_SP_TB_1 (30)** type selected. Each type has different options that can be configured when sending data. For example this type depicted in the picture below can send a value **Off** or

On and it also is time-tagged. The user in this case can either select a specific time that they have in mind or just mark the PC checkbox and The Vinci software will automatically send the current PC time. As you can see the Value box in this example is greyed out that is because this tag is created on a **master** simulation, and this type doesn't support writing to slave.

Tag

Name:

Type:

Asdu: Ioa: Value:

Quality:

☐ BL ☐ SB ☐ NT ☐ IV ☐ OV

Time:

☐ PC

Setup

To setup an IEC 60870-5-101 simulation it is fairly straightforward.

1. Select IEC 60870-5-101 and the mode.

Protocol:

Mode:

2. Select Serial Port settings according to your device specification.

Port: Baudrate: Parity: Data bits: Stop bits:

3. Select settings in the settings tab according to your device and preference.

| Settings | Console | Statistic |
|--|---------|-----------|
| Structure | | |
| Link size in bytes: <input type="text" value="2"/> | | |
| COT size in bytes: <input type="text" value="2"/> | | |
| ASDU size in bytes: <input type="text" value="2"/> | | |
| IOA size in bytes: <input type="text" value="3"/> | | |
| Timeouts | | |
| Reading data: <input type="text" value="2000"/> | | |
| Pause before send: <input type="text" value="1000"/> | | |
| Address | | |
| Link address: <input type="text" value="1"/> | | |

4. Press the green **START** button and the simulation should start. If everything was done correctly The Vinci software should establish communication with the IEC 60870-5-101 device which you can monitor in the console tab.

Protocol:

Mode:

START

IEC 60870-5-103

IEC 60870-5-103 is a protocol for power system monitoring and controlling. Mostly used for communication between protection devices and devices of a control system in a substation (RTU) over fiber optics.



Info about protocol

Telegram Structure

Telegram format with fixed length

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|--------------------------|-----|------------|------------|---------------|---|---|---|
| 0 | Start byte | | | | | | | |
| 1 | RES | PRM | FCB ACD | FCV DFC | Function code | | | |
| 2 | Link address (1-2 bytes) | | | | | | | |
| 3 | Checksum | | | | | | | |
| 4 | Stop byte | | | | | | | |

Telegram format with variable length

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|-------------------------|-------------------|------------|------------|---------------|---|---|---|
| 0 | Start byte | | | | | | | |
| 1 | Length | | | | | | | |
| 2 | Length | | | | | | | |
| 4 | Start byte | | | | | | | |
| 5 | RES | PRM | FCB ACD | FCV DFC | Function code | | | |
| 6 | Link address | | | | | | | |
| 7 | Type identification | | | | | | | |
| 8 | SQ | Number of objects | | | | | | |
| 9 | Cause of transmission | | | | | | | |
| 10 | ASDU address | | | | | | | |
| 11 | Function type | | | | | | | |
| 12 | Information number | | | | | | | |
| 13... | Information elements... | | | | | | | |
| x | Checksum | | | | | | | |
| x | Stop byte | | | | | | | |

- **RES** - Reserved
- **PRM** - 1 if master, 0 if slave

PRM = 1

- **FCB** - alternating bit for successive services per station

- **FCV** - (if **FCV**=1 **FCB** enabled)

PRM = 0

- **ACD** - access demand (if **ACD**=1 there are class 1 data)
- **DFC** - data flow control (if **DFC**=1 further messages may cause data overflow)

Function Code

PRM=1

| Dec | Frame type | Service function | FCV |
|-----|---------------------------|---|-----|
| 0 | SEND/CONFIRM expected | Reset of remote link | 0 |
| 1 | SEND/CONFIRM expected | Reset of user process | 0 |
| 2 | SEND/CONFIRM expected | Reserved | - |
| 3 | SEND/CONFIRM expected | User data | 1 |
| 4 | SEND/REPLY expected | User data | 0 |
| 5 | | Reserved | - |
| 6 | | Reserved | - |
| 7 | | Reserved | - |
| 8 | REQUEST for access demand | Expected response specifies access demand | 0 |
| 9 | REQUEST/RESPOND expected | Request status of link | 0 |
| 10 | REQUEST/RESPOND expected | Request user data class 1 | 1 |
| 11 | REQUEST/RESPOND expected | Request user data class 2 | 1 |
| 12 | | Reserved | - |
| 13 | | Reserved | - |
| 14 | | Reserved | - |
| 15 | | Reserved | - |

PRM=0

| Dec | Frame type | Service function |
|-----|------------|---------------------------------------|
| 0 | CONFIRM | ACK: positive acknowledgment |
| 1 | CONFIRM | NACK: message not accepted, link busy |
| 2 | | Reserved |
| 3 | | Reserved |
| 4 | | Reserved |
| 5 | | Reserved |
| 6 | | Reserved |

| | | |
|----|---------|------------------------------------|
| 7 | | Reserved |
| 8 | RESPOND | User data |
| 9 | RESPOND | NACK: requested data not available |
| 10 | | Request user data class 1 |
| 11 | RESPOND | Request user data class 2 |
| 12 | | Reserved |
| 13 | | Reserved |
| 14 | | Reserved |
| 15 | | Reserved |

Type identification

Standard IEC 60870-5-103 data types[1-255]

- [1-31] - standard definition
- [32-255] - for special use

| Dec | Description | Direction | Support |
|-----|--|-----------|---------|
| 1 | Time-tagged message | Monitor | Yes |
| 2 | Time-tagged message with relative time | Monitor | Yes |
| 3 | Measurands I | Monitor | Yes |
| 4 | Time-tagged measurands with relative time | Monitor | Yes |
| 5 | Identification | Monitor | Yes |
| 6 | Clock synchronization | Both | Yes |
| 7 | General interrogation | Control | Yes |
| 8 | End of general interrogation | Monitor | Yes |
| 9 | Measurands II | Monitor | Yes |
| 10 | Generic data | Both | No |
| 11 | Generic identification | Monitor | No |
| 20 | General command | Control | Yes |
| 21 | Generic command | Control | No |
| 23 | List of recorded disturbances | Monitor | No |
| 24 | Order for disturbance data transmission | Control | No |
| 25 | Acknowledgment for disturbance data transmission | Control | No |
| 26 | Ready for transmission of disturbance data | Monitor | No |
| 27 | Ready for transmission of a channel | Monitor | No |
| 28 | Ready for transmission of tags | Monitor | No |
| 29 | Transmission of tags | Monitor | No |
| 30 | Transmission of disturbance values | Monitor | No |
| 31 | End of transmission | Monitor | No |

Cause of transmission

Standard IEC 60870-5-103 cause of transmission [0-255]

- [0] - not used
- [1-63] - standard definition
- [64-255] - for special use

| Dec | Description |
|-----|-------------|
|-----|-------------|

| | |
|----|---|
| 1 | Spontaneous |
| 2 | Cyclic |
| 3 | Reset frame count bit (FCB) |
| 4 | Reset communication unit (CU) |
| 5 | Start/ restart |
| 6 | Power ON |
| 7 | Test mode |
| 8 | Time synchronization |
| 9 | General interrogation |
| 10 | End of general interrogation |
| 11 | Return information caused by a remote command |
| 12 | Return information caused by a local command |
| 20 | Command "ACK positive" |
| 21 | Command "ACK negative" |
| 31 | Transmission disturbance data |
| 40 | Generic write command with ACK positive |
| 41 | Generic write command with ACK negative |
| 42 | Generic read command data valid |
| 43 | Generic read command data invalid |
| 44 | Generic write conformation |

Settings

| Timeouts (ms) | | | | |
|---|----------------------------------|------------------------------------|------------------------------------|------------------------------------|
| <div>Timeouts (ms)</div> <div> <div>Reading data: <input type="text" value="1000"/></div> <div>Pause before send: <input type="text" value="100"/></div> </div> | | Monitor | Master | Slave |
| | Reading data | Waiting data in serial port buffer | Waiting data in serial port buffer | Waiting data in serial port buffer |
| | Pause before send | Not used | Pause before send data | Pause before send data |
| Address | | | | |
| <div>Address</div> <div>Link address: <input type="text" value="1"/></div> | | Monitor | Master | Slave |
| | Link | Not used | Remote device address | Own system address |
| | ASDU | Not used | Remote device address | Own system address |
| Commands ack. | | | | |
| <div>Parameters</div> <div> <input checked="" type="checkbox"/> Auto ack. control commands <input checked="" type="checkbox"/> Auto ack. system commands </div> | | Monitor | Master | Slave |
| | Auto ack. system commands | Not used | Not used | Auto acknowledge system commands |

| | | | | |
|--|-----------------------------------|----------|----------|---------------------------|
| | Auto ack. control commands | Not used | Not used | Auto acknowledge commands |
|--|-----------------------------------|----------|----------|---------------------------|

System

For all system functions user can set custom address:

APDU

ASDU:

General Interrogation

This function will send telegram Type-identification = 7

General interrogation

Scan:

Clock synchronization

This function will send telegram Type-identification = 103 (C_CS_NA_1)

Clock synchronization

☐ IV ☐ SM ☐ SB

☒ PC time

If "PC time" checkbox is checked, then the PC time will be sent. If it's not checked user can set time manually.

Time tag status bits:

- **IV** - invalid time
- **SM** - Summer/Winter
- **SB** - Substitute

General Command

This function allows user to send command to slave device.

General Command

FUN: INF: Rti:

Tags

This function allows user to created named points. After points created user can send it manually or set reply checkbox to automatic reply.

- To export Tags to csv file: **Tags -> Export -> Save file dialog appear**
- To import Tags from csv file: **Tags -> Import -> Open file dialog appear**

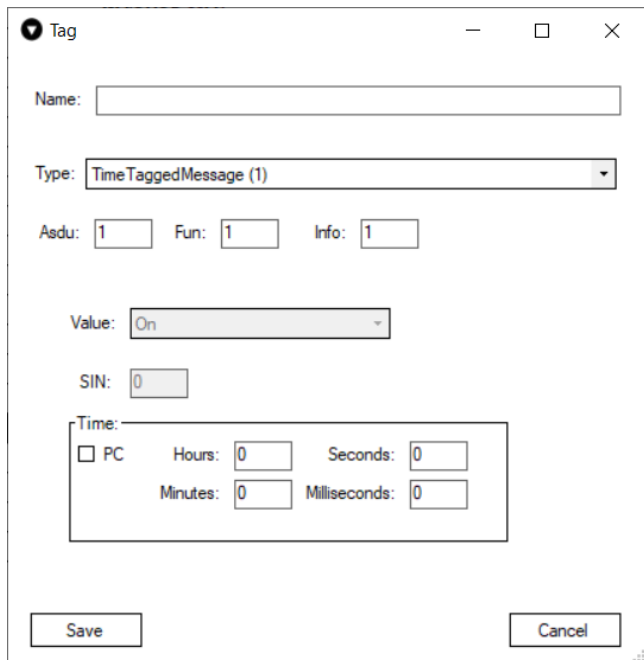
There are two ways of creating tags:

1. Create tag button.
2. Double click a signal with the left mouse button in the statistic tab.

Main parameters:

- **Name** - user-friendly tag name
- **Type** - the type of value.
- **Asdu** - Identifier of the device
- **Fun** - function number
- **Info** - Identifier of values from the device.

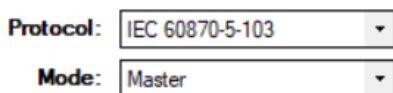
Here is an example image of the tag window with the **TimeTaggedMessage(1)** type selected. Each type has different options that can be configured when sending data. For example this type depicted in the picture below can send a value **Off** or **On** and it also is time-tagged. The user in this case can either select a specific time that they have in mind or just mark the PC checkbox and The Vinci software will automatically send the current PC time. As you can see the **Value** box in this example is greyed out that is because this tag is created on **amaster** simulation, and this type doesn't support writing to slave.



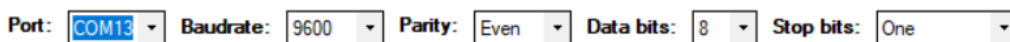
Setup

To setup an IEC 60870-5-103 simulation it is fairly straightforward.

1. Select IEC 60870-5-103 and the mode.



2. Select Serial Port settings according to your device specification.



3. Select settings in the settings tab according to your device and preference.

Timeouts

Reading data:

100

Pause before send:

100

Address

Link address:

1

4. Press the green **START** button and the simulation should start. If everything was done correctly The Vinci software should establish communication with the IEC 60870-5-103 device which you can monitor in the console tab.

Protocol:

IEC 60870-5-103

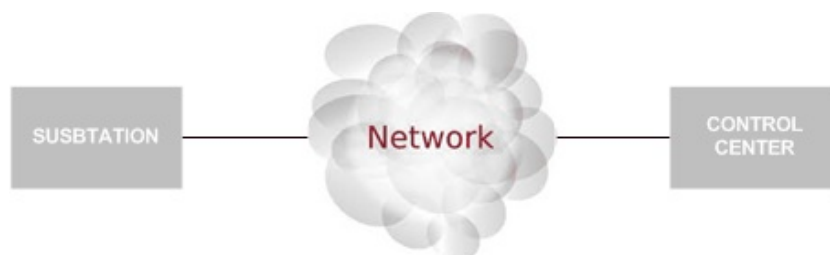
Mode:

Master

START

IEC 60870-5-104

IEC 60870-5-104 is a protocol for power system monitoring and controlling. Mostly used to communication between substations and control centers over Ethernet (Fiber optics, 2/3/4G, ...). IEC 60870-5-104 protocol is an extension of IEC 60870-5-101 protocol with the changes in transport, network, link and physical layer services to suit the complete network access.



Info about protocol

Telegram Structure

Teleram format with fixed length

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|-----------------|---|---|---|---|---|---|---|
| 0 | Start byte | | | | | | | |
| 1 | Length of APDU | | | | | | | |
| 2 | Control field 1 | | | | | | | |
| 3 | Control field 2 | | | | | | | |
| 4 | Control field 3 | | | | | | | |
| 5 | Control field 4 | | | | | | | |

Telegram format with variable length

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|-----------------|---|---|---|---|---|---|---|
| 0 | Start byte | | | | | | | |
| 1 | Length of APDU | | | | | | | |
| 2 | Control field 1 | | | | | | | |
| 3 | Control field 2 | | | | | | | |
| 4 | Control field 3 | | | | | | | |
| 5 | Control field 4 | | | | | | | |
| 6... | ASDU | | | | | | | |
| | | | | | | | | |

- **APCI** - Application Protocol Control Information (First 6 bytes)
- **APDU** - Application Protocol Data Unit (All variable length telegram)
- **ASDU** - Application Service Data Unit

Type identification

Standard IEC 60870-5-104 data types[1-255]

- [1-127] - standard definition
- [128-135] - reserved for routing of messages
- [136-255] - for special use

| Dec | Type | Description | Direction | Support |
|---------------------|-----------|---|-----------|---------|
| Process information | | | | |
| 1 | M_SP_NA_1 | Single-point information | Monitor | Yes |
| 2 | M_SP_TA_1 | Single-point information with time tag | Monitor | Yes |
| 3 | M_DP_NA_1 | Double-point information | Monitor | Yes |
| 4 | M_DP_TA_1 | Double-point information with time tag | Monitor | Yes |
| 5 | M_ST_NA_1 | Step position information | Monitor | Yes |
| 6 | M_ST_TA_1 | Step position information with time tag | Monitor | Yes |
| 7 | M_BO_NA_1 | Bit string of 32 bit | Monitor | Yes |
| 8 | M_BO_TA_1 | Bit string of 32 bit with time tag | Monitor | Yes |
| 9 | M_ME_NA_1 | Measured value, normalized value | Monitor | Yes |
| 10 | M_ME_TA_1 | Measured value, normalized value with time tag | Monitor | Yes |
| 11 | M_ME_NB_1 | Measured value, scaled value | Monitor | Yes |
| 12 | M_ME_TB_1 | Measured value, scaled value with time tag | Monitor | Yes |
| 13 | M_ME_NC_1 | Measured value, short floating point number | Monitor | Yes |
| 14 | M_ME_TC_1 | Measured value, short floating point number with time tag | Monitor | Yes |
| 15 | M_IT_NA_1 | Integrated totals | Monitor | Yes |
| 16 | M_IT_TA_1 | Integrated totals with time tag | Monitor | Yes |
| 17 | M_EP_TA_1 | Event of protection equipment with time tag | Monitor | Yes |
| 18 | M_EP_TB_1 | Packed start events of protection equipment with time tag | Monitor | Yes |
| 19 | M_EP_TC_1 | Packed output circuit information of protection equipment with time tag | Monitor | Yes |
| 20 | M_PS_NA_1 | Packed single point information with status change detection | Monitor | Yes |
| 21 | M_ME_ND_1 | Measured value, normalized value without quality descriptor | Monitor | Yes |
| 30 | M_SP_TB_1 | Single-point information with time tag CP56Time2a | Monitor | Yes |

| | | | | |
|----|-----------|--|---------|-----|
| 31 | M_DP_TB_1 | Double-point information with time tag CP56Time2a | Monitor | Yes |
| 32 | M_ST_TB_1 | Step position information with time tag CP56Time2a | Monitor | Yes |
| 33 | M_BO_TB_1 | Bit string of 32 bit with time tag CP56Time2a | Monitor | Yes |
| 34 | M_ME_TD_1 | Measured value, normalized value with time tag CP56Time2a | Monitor | Yes |
| 35 | M_ME_TE_1 | Measured value, scaled value with time tag CP56Time2a | Monitor | Yes |
| 36 | M_ME_TF_1 | Measured value, short floating point number with time tag CP56Time2a | Monitor | Yes |
| 37 | M_IT_TB_1 | Integrated totals with time tag CP56Time2a | Monitor | Yes |
| 38 | M_EP_TD_1 | Event of protection equipment with time tag CP56Time2a | Monitor | Yes |
| 39 | M_EP_TE_1 | Packed start events of protection equipment with time tag CP56Time2a | Monitor | Yes |
| 40 | M_EP_TF_1 | Packed output circuit information of protection equipment with time tag CP56Time2a | Monitor | Yes |
| 45 | C_SC_NA_1 | Single command | Control | Yes |
| 46 | C_DC_NA_1 | Double command | Control | Yes |
| 47 | C_RC_NA_1 | Regulating step command | Control | Yes |
| 48 | C_SE_NA_1 | Set-point Command, normalized value | Control | Yes |
| 49 | C_SE_NB_1 | Set-point Command, scaled value | Control | Yes |
| 50 | C_SE_NC_1 | Set-point Command, short floating point number | Control | Yes |
| 51 | C_BO_NA_1 | Bit string 32 bit command | Control | Yes |
| 58 | C_SC_TA_1 | Single command with time tag CP56Time2a | Control | Yes |
| 59 | C_DC_TA_1 | Double command with time tag CP56Time2a | Control | Yes |
| 60 | C_RC_TA_1 | Regulating step command with time tag CP56Time2a | Control | Yes |
| 61 | C_SE_TA_1 | Measured value, normalized value command with time tag CP56Time2a | Control | Yes |
| 62 | C_SE_TB_1 | Measured value, scaled value command with time tag CP56Time2a | Control | Yes |
| 63 | C_SE_TC_1 | Measured value, short floating point number command with time tag CP56Time2a | Control | Yes |
| 64 | C_BO_TA_1 | Bit string of 32 bit command with time tag CP56Time2a | Control | Yes |

| System information | | | | |
|--------------------|-----------|---|---------------|-----|
| 70 | M_EI_NA_1 | End of Initialization | Monitor | Yes |
| 100 | C_IC_NA_1 | Interrogation command | Control | Yes |
| 101 | C_CI_NA_1 | Counter interrogation command | Control | Yes |
| 102 | C_RD_NA_1 | Read command | Control | Yes |
| 103 | C_CS_NA_1 | Clock synchronization command | Control | Yes |
| 104 | C_TS_NA_1 | Test command | Control | Yes |
| 105 | C_RP_NA_1 | Reset process command | Control | Yes |
| 106 | C_CD_NA_1 | Delay acquisition command | Control | No |
| 107 | C_TS_TA_1 | Test command with time tag CP56Time2a | Control | No |
| Parameter | | | | |
| 110 | P_ME_NA_1 | Parameter of measured values, normalized value | Control | No |
| 111 | P_ME_NB_1 | Parameter of measured values, scaled value | Control | No |
| 112 | P_ME_NC_1 | Parameter of measured values, short floating point number | Control | No |
| 113 | P_AC_NA_1 | Parameter activation | Control | No |
| File transfer | | | | |
| 120 | F_FR_NA_1 | File ready | File transfer | No |
| 121 | F_SR_NA_1 | Section ready | File transfer | No |
| 122 | F_SC_NA_1 | Call directory, select file, call file, call section | File transfer | No |
| 123 | F_LS_NA_1 | Last section, last segment | File transfer | No |
| 124 | F_FA_NA_1 | ACK file, ACK section | File transfer | No |
| 125 | F_SG_NA_1 | Segment | File transfer | No |
| 126 | F_DR_TA_1 | Directory | File transfer | No |

| | | | | |
|-----|-----------|----------------------|---------------|----|
| 127 | F_SC_NB_1 | Request archive file | File transfer | No |
|-----|-----------|----------------------|---------------|----|

Cause of transmission

Standard IEC 60870-5-101 cause of transmission [0-63]

| Dec | Description |
|-----|---|
| 1 | Periodic, cyclic |
| 2 | Background interrogation |
| 3 | Spontaneous |
| 4 | Initialized |
| 5 | Interrogation or interrogated |
| 6 | Activation |
| 7 | Confirmation activation |
| 8 | Deactivation |
| 9 | Confirmation deactivation |
| 10 | Termination activation |
| 11 | Return information caused by a remote command |
| 12 | Return information caused by a local command |
| 13 | File transfer |
| 20 | Interrogated by general interrogation |
| 21 | Interrogated by interrogation group 1 |
| 22 | Interrogated by interrogation group 2 |
| 23 | Interrogated by interrogation group 3 |
| 24 | Interrogated by interrogation group 4 |
| 25 | Interrogated by interrogation group 5 |
| 26 | Interrogated by interrogation group 6 |
| 27 | Interrogated by interrogation group 7 |
| 28 | Interrogated by interrogation group 8 |

| | |
|----|---|
| 29 | Interrogated by interrogation group 9 |
| 30 | Interrogated by interrogation group 10 |
| 31 | Interrogated by interrogation group 11 |
| 32 | Interrogated by interrogation group 12 |
| 33 | Interrogated by interrogation group 13 |
| 34 | Interrogated by interrogation group 14 |
| 35 | Interrogated by interrogation group 15 |
| 36 | Interrogated by interrogation group 16 |
| 37 | Interrogated by counter general interrogation |
| 38 | Interrogated by interrogation counter group 1 |
| 39 | Interrogated by interrogation counter group 2 |
| 40 | Interrogated by interrogation counter group 3 |
| 41 | Interrogated by interrogation counter group 4 |
| 44 | Type Identification unknown |
| 45 | Cause unknown |
| 46 | ASDU address unknown |
| 47 | Information object address unknown |

Settings

Structure

COT size in bytes: 2

ASDU size in bytes: 2

IOA size in bytes: 3

| Structure | | |
|--------------------|------------------------|--|
| | Master, Slave, Monitor | |
| COT size in bytes | COT size in bytes | |
| ASDU size in bytes | ASDU size in bytes | |
| IOA size in bytes | IOA size in bytes | |

Timeouts

t0 in seconds

Timeout for the establishment of the connection with the server.

| Timeouts (ms) | | |
|---------------|--|----------|
| | Master | Slave |
| t0 in seconds | Timeout for the establishment of the connection with the server. | Not used |

| | | | |
|--|-------------------------------------|--|---|
| <div> <div>t0 in seconds: <input type="text" value="30"/></div> <div>t1 in seconds: <input type="text" value="15"/></div> <div>t2 in seconds: <input type="text" value="10"/></div> <div>t3 in seconds: <input type="text" value="20"/></div> </div> | t1 in seconds | This parameter defines the time in seconds that Master waits maximum for an acknowledge from the slave. | This parameter defines the time in seconds that slave waits maximum for an acknowledge from the master. |
| | t2 in seconds | A S-format frame will be sent at the latest after this time starting from the last received telegram from the slave. | A S-format frame will be sent at the latest after this time starting from the last received telegram from the master. |
| | t3 in seconds | A Test frame will be sent at the latest after this time starting from the last received telegram from the slave. | A Test frame will be sent at the latest after this time starting from the last received telegram from the master. |
| | Windows | | |
| <div> <div>Windows</div> <div>RWT (w) size: <input type="text" value="8"/></div> <div>SWT (k) size: <input type="text" value="12"/></div> </div> | | Master | Slave |
| | w size | This parameter indicates the number of received I frames after the S-Frame will be send. | This parameter indicates the number of received I frames after the S-Frame will be send |
| | k size | Maximum I-frames send until acknowledgment. | Not used |
| SLAVE Parameters | | | |
| <div> <div>Parameters</div> <div> <input checked="" type="checkbox"/> Send End of ini. on start up <input checked="" type="checkbox"/> Auto ack. U-Frame <input checked="" type="checkbox"/> Auto ack. control commands <input checked="" type="checkbox"/> Auto ack. system commands </div> </div> | | Slave | |
| | Send End of ini. on start up | Send end of initialization TI 70 (M_EI_NA_1) | |
| | Auto ack. U-Frame | Auto ack. U-Frame. | |
| | Auto ack. control commands | Auto acknowledge commands | |
| | Auto ack. system commands | Auto acknowledge system commands (TI: 100, 103) | |
| MASTER Parameters | | | |
| <div> <div>Parameters</div> <div> <input checked="" type="checkbox"/> Send Start DT on start up <input checked="" type="checkbox"/> Auto ack. Test Frame </div> </div> | | Master | |
| | Send Start DT on start up | Send Start DT on startup | |
| | Auto ack. Test Frame | Auto ack. Test frame | |

System

For all system functions user can set custom address:

APDU

ASDU:

☐ Test

Originator:

General Interrogation

This function will send telegram Type-identification = 100 (C_IC_NA_1)

General interrogation

Send

QOI:

QOI - qualifier of interrogation [0...255]

- 20 - Station interrogation
- 21 - Interrogation of group 1

- 22 - Interrogation of group 2
- 23 - Interrogation of group 3
- 24 - Interrogation of group 4
- 25 - Interrogation of group 5
- 26 - Interrogation of group 6
- 27 - Interrogation of group 7
- 28 - Interrogation of group 8
- 29 - Interrogation of group 9
- 30 - Interrogation of group 10
- 31 - Interrogation of group 11
- 32 - Interrogation of group 12
- 33 - Interrogation of group 13
- 34 - Interrogation of group 14
- 35 - Interrogation of group 15
- 36 - Interrogation of group 16

Counter Interrogation

This function will send telegram Type-identification = 101 (C_CI_NA_1)

Counter interrogation

FRZ:
RQT:

FRZ - freeze[0..3]

- 0 - Station interrogation
- 1 - Interrogation of group 1
- 2 - Interrogation of group 2
- 3 - Interrogation of group 3

RQT - request[0..63]

- 1 - Counter group 1
- 2 - Counter group 2
- 3 - Counter group 3
- 4 - Counter group 3
- 5 - General request

Commands

Read command will send telegram Type-identification = 102 (C_RD_NA_1)

Test command will send telegram Type-identification = 104 (C_TS_NB_1)

Commands

Clock synchronization

This function will send telegram Type-identification = 103 (C_CS_NA_1)

Clock synchronization

☐ IV
☐ SM
☐ SB

☒ PC time

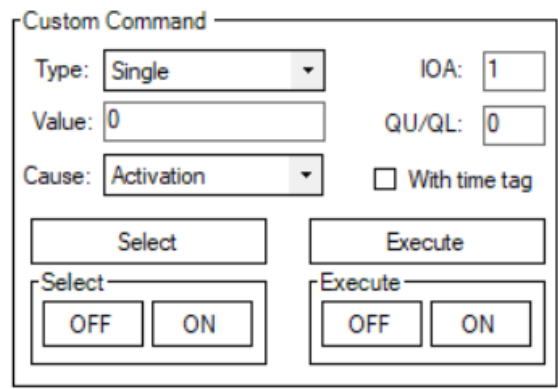
If "PC time" checkbox is checked, then the PC time will be sent. If it's not checked user can set time manually.

Time tag status bits:

- **IV** - invalid time
- **SM** - Summer/Winter
- **SB** - Substitute

Custom Commands

This function allows user to send commands to the slave device.



Custom Command

Type: IOA:

Value: QU/QL:

Cause: ☐ With time tag

Select

Select

Execute

Channel

With these functions a user has the ability to send any U or S frame telegram.



U-frame

- **Start DT act** - Send *Start Data terminal* activation
- **Start DT cnf** - Send *Start Data terminal* confirmation
- **Stop DT act** - Send *Stop Data terminal* activation
- **Stop DT cnf** - Send *Stop Data terminal* confirmation
- **Test Frm act** - Send *Test Frame* activation
- **Test Frm cnf** - Send *Test Frame* confirmation



S-frame

S-Frame ack - Send S-Frame. User can specify acknowledgment telegram count in text box.

Tags

This function allows user to created named points. After points created user can send it manually or set reply checkbox to automatic reply.

- To export Tags to csv file: **Tags -> Export -> Save file dialog appear**
- To import Tags from csv file: **Tags -> Import -> Open file dialog appear**

There are two ways of creating tags:

1. Create tag button.
2. Double click a signal with the left mouse button in the statistic tab.

Main parameters:

- **Name** - user-friendly tag name
- **Asdu** - Identifier of the device
- **loa** - Identifier of values from the device.
- **Type** - the type of value.

Here is an example image of the tag window with the **M_SP_TB_1 (30)** type selected. Each type has different options that can be configured when sending data. For example this type depicted in the picture below can send a value **Off** or **On** and it also is time-tagged. The user in this case can either select a specific time that they have in mind or just mark the PC checkbox and The Vinci software will automatically send the current PC time. As you can see the Value box in this example is greyed out that is because this tag is created on a **master** simulation, and this type doesn't support writing to slave.

Tag

Name:

Type:

Asdu: Ioa: Value:

Quality:

☐ BL ☐ SB ☐ NT ☐ IV ☐ OV

Time:

☐ PC

Setup

To setup an IEC 60870-5-104 simulation it is fairly straightforward.

1. Select IEC 60870-5-104 and the mode.

Protocol:

Mode:

2. Select Ethernet settings to connect to device. Set the **IP** and the **Port**. (Default port: 2404)

IP:

Port:

3. Select settings in the settings tab according to your device and preference.

Structure

COT size in bytes:

ASDU size in bytes:

IOA size in bytes:

Parameters

☒ Send Start DT on start up

☒ Auto ack. Test Frame

Timeouts

t0 in seconds:

t1 in seconds:

t2 in seconds:

t3 in seconds:

Windows

RWT (w) size:

SWT (k) size:

4. Press the green **START** button and the simulation should start. If everything was done correctly The Vinci software should establish communication with the IEC 60870-5-104 device which you can monitor in the console tab.

Protocol:

Mode:

