

DNNF012

DNNF020

TCP/IP or UDP Integration of uniVision Products into Control Systems



Interface Protocol

Table of Contents

1. Use for Intended Purpose	3
2. Network Overview	4
3. Settings in uniVision	5
4. TIA Sample Program	11
4.1 Receiving Process Data from TCP Device	11
4.2 Receiving Process Data from UDP Device	15
4.3 Sending LIMA Commands via TCP/IP and Receiving LIMA Answers	18
5. TwinCAT3 Sample Programs	24
5.1 Receiving Process Data from TCP Device	25
5.2 Receiving Process Data from UDP Device	27
5.3 Sending LIMA Commands via TCP/IP and Receiving LIMA Answers	29
6. Rockwell Sample Programs	32
6.1 Receiving Process Data from the TCP Device	32
6.2 Receiving Process Data from the UDP Device	34
6.3 Sending LIMA Commands via TCP/IP and Receiving LIMA Responses	36

1. Use for Intended Purpose

The instructions show, for example, integration of uniVision products into various control environments via the TCP/IP or UDP interface. These instructions are in addition to the control sample programs and show, among other things, which changes are necessary for a different network configuration or for a different number of characters transmitted via TCP/IP or UDP.

The following uniVision products can be integrated in this way:



uniVision
all-in-one software

Image Analysis



Smart Camera

Profile Analysis



Smart 2D/3D Profile Sensor



VisionSystem2D



Control Unit with 2D/3D Profile Sensors

The sample programs are available for the following control environments:

- Siemens PLC S7-1200 with TIA Portal V15
- Beckhoff TwinCAT3
- SPS 1769-L18ERM-BB1B from Allen-Bradley with Studio 5000 Logix Designer V32

Depending on the control environment, the sample program contains a different scope of functions. In general, the following functions are possible in the control sample programs:

- Receiving process data from the TCP device
- Receiving process data from the UDP device
- Sending LIMA commands (e.g. trigger commands) via TCP/IP and receiving LIMA answers

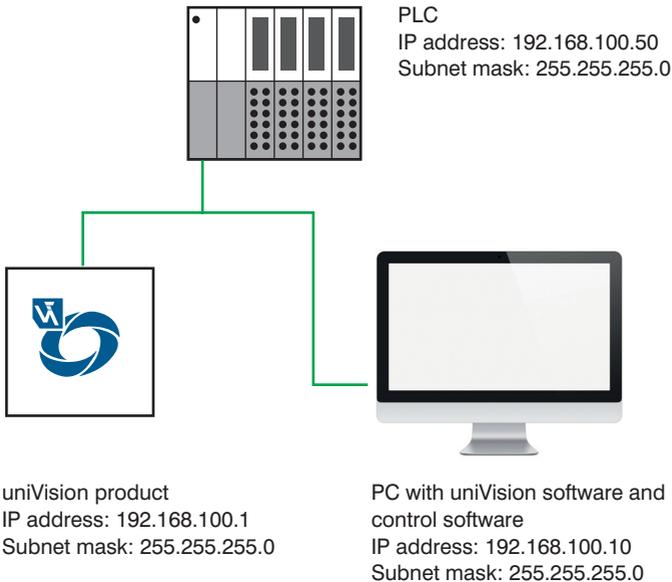


NOTE!

The control sample programs are supported starting with uniVision version 2.4.0.

2. Network Overview

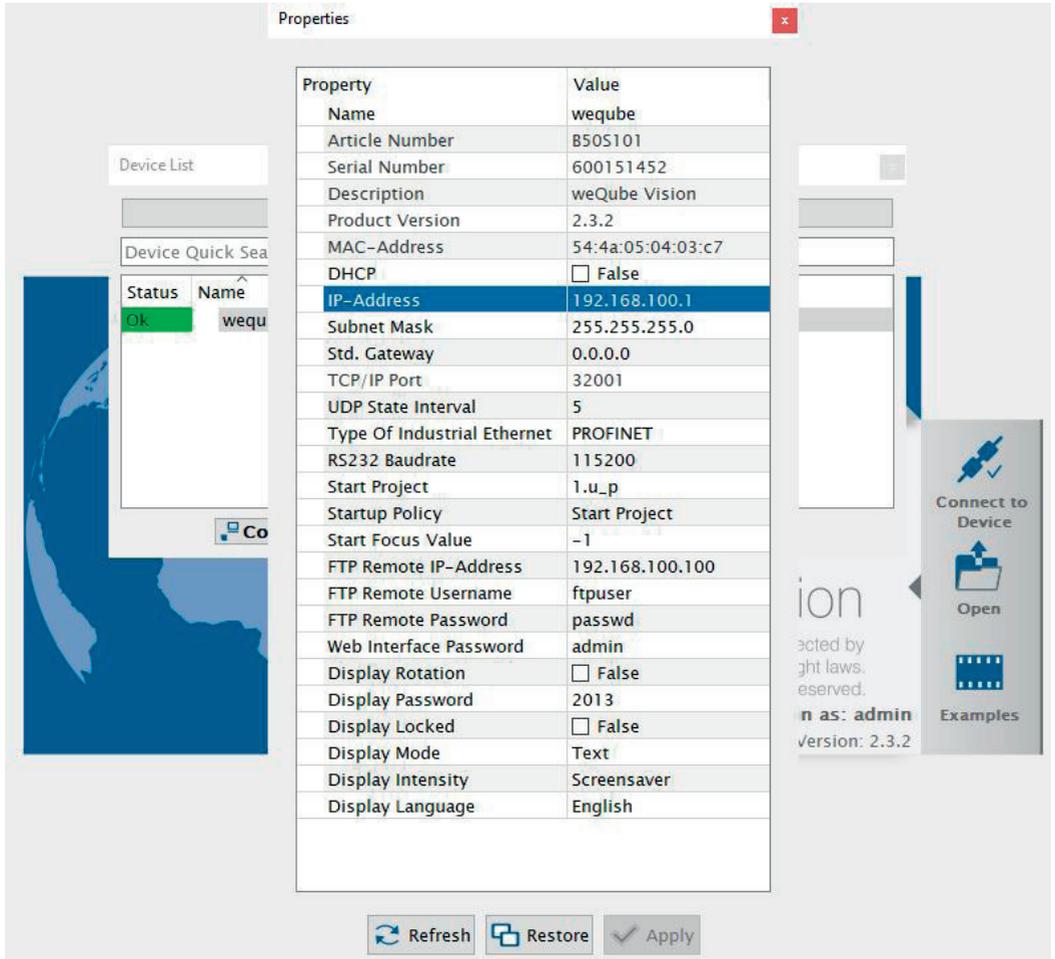
The uniVision product, the PLC and PC with uniVision software, and the control software must be on the same network. The following network settings are used in the sample program.



3. Settings in uniVision

The following steps are necessary to start the uniVision software:

1. Install and open uniVision software for Windows (article number: DNNF020)
2. Set up the network configuration of the uniVision product via the uniVision software. To do this, select the uniVision product in the device list and click Properties.



The screenshot shows the 'Properties' dialog box for a device named 'weqube'. The 'IP-Address' field is highlighted in blue. To the left, a 'Device List' window is partially visible, showing a table with columns 'Status' and 'Name'. Below the main table, there are 'Refresh', 'Restore', and 'Apply' buttons.

Property	Value
Name	weqube
Article Number	B505101
Serial Number	600151452
Description	weQube Vision
Product Version	2.3.2
MAC-Address	54:4a:05:04:03:c7
DHCP	<input type="checkbox"/> False
IP-Address	192.168.100.1
Subnet Mask	255.255.255.0
Std. Gateway	0.0.0.0
TCP/IP Port	32001
UDP State Interval	5
Type Of Industrial Ethernet	PROFINET
RS232 Baudrate	115200
Start Project	1.u_p
Startup Policy	Start Project
Start Focus Value	-1
FTP Remote IP-Address	192.168.100.100
FTP Remote Username	ftpuser
FTP Remote Password	passwd
Web Interface Password	admin
Display Rotation	<input type="checkbox"/> False
Display Password	2013
Display Locked	<input type="checkbox"/> False
Display Mode	Text
Display Intensity	Screensaver
Display Language	English

3. Double-click to connect to the uniVision product and load a template onto the product.

- Set the trigger mode to software or trigger in order to later use the LIMA interface via TCP/IP and send trigger commands to the uniVision device.

The screenshot shows the uniVision software interface. At the top, the 'Navigator' window displays a tree view of modules under 'Module Application'. The 'Device Camera' module is selected. Below the Navigator is a 'Start Assistant' button. Below that is a table of properties and their values.

Property	Value	
Light Internal	<input checked="" type="checkbox"/>	⚙️
Light External	<input type="checkbox"/>	⚙️
Rotate Input Image	<input type="checkbox"/>	⚙️
Create HSV Image	<input checked="" type="checkbox"/>	⚙️
Create RGB Image	<input type="checkbox"/>	⚙️
Create Raw Image	<input type="checkbox"/>	⚙️
Create BGRA Image	<input type="checkbox"/>	⚙️
Exposure Time [us]	200	⚙️
Focus Position [steps]	141	⚙️
Auto Focus	<input type="checkbox"/>	⚙️
Light Current [%]	20	⚙️
Light Mode	Flash Light	⚙️
Trigger Mode	Trigger	⚙️

- In order to send process data via TCP/IP or UDP, the TCP or UDP device must also be available in the project tree and configured accordingly.



NOTE!

The TCP device and UDP device for communicating with the control system are already preconfigured in the template. Alternatively, a new project can be created and the TCP or UDP device added manually to the project from the toolbar.

4. Any character count, preamble, separator and postamble can be configured on the TCP or UDP device. In addition, the output mode should be set to “Formatted” in order to define a fixed character count. This makes it easier to read out the process data on the control system.

Navigator ☰ ×

- ▼ Module Application
 - >  Device Camera
 - >  Module Region
 - >  Module Threshold
 - >  Module Counter
 - >  Device IO Unit
 - >  Device TCP
 - >  Device UDP
 - >  Add Module

Property	Value	⚙
<i>Process Time [us]</i>	1000	⚙
<i>Module State</i>	0	⚙
<i>Output</i>	+0027958,+0005748,+0016000,+0000000,+0035302,+0004300,+0001448,+0000000,+0006000;	
Preamble		⚙
Postamble	;	⚙
Delimiter	,	⚙
String Count	9	⚙
<i>Output Mode</i>	Formatted	⚙
<i>Error Handling</i>	Value Substitution	⚙
Connections	5	⚙
TCP Port	32002	⚙
Blocking Mode	<input type="checkbox"/>	⚙

5. If the output mode is set to “Formatted”, the character count for the various data types can be configured under “Formatting options”.



NOTE!

In the example, a total of eight characters are used for “integral numbers” and “floating point numbers” (incl. sign and comma). A character is used for bool data type results.

Property	Value
<i>Digits Before Comma</i>	4
<i>Digits After Comma</i>	2
<i>Print +</i>	<input checked="" type="checkbox"/>

6. The character count should also be selected for the replacement value defined under troubleshooting. In the example, eight characters are also used for the error replacement value.



The screenshot shows a software interface with a 'Navigator' window on the left and a 'Property' table below it.

The Navigator window contains a tree view with the following structure:

- Module Application
 - Device Camera
 - Module Region
 - Module Threshold
 - Module Counter
 - Device IO Unit
 - Device TCP
 - String Count
 - Error Handling**
 - Formatting Options
 - Integer
 - Floating Point
 - Boolean
 - Device UDP
 - Add Module

The 'Property' table below the Navigator has the following content:

Property	Value
<i>Substitute STRING Types by</i>	Error###

- The total number of characters sent via TCP or UDP can be determined under “Output” on the TCP or UDP device.

The screenshot shows the uniVision software interface. The Navigator panel on the left displays a tree view of modules under 'Module Application'. The 'Device TCP' module is selected and highlighted. Below the Navigator is the Property panel, which lists various configuration parameters for the selected module.

Property	Value
<i>Process Time [us]</i>	1000
<i>Module State</i>	0
<i>Output</i>	+0027958,+0005748,+0016000,+0000000,+0035302,+0004300,+0001448,+0000000,+0006000;
Preamble	
Postamble	;
Delimiter	,
String Count	9
<i>Output Mode</i>	Formatted
<i>Error Handling</i>	Value Substitution
Connections	5
TCP Port	32002
Blocking Mode	<input type="checkbox"/>

- Save the project on the uniVision device and store it as a starter project in the device's properties.

4. TIA Sample Program

The TIA sample program is created with a Siemens PLC S7-1200 with TIA Portal V15. It includes the following use cases:

- Receiving process data from the TCP device
- Receiving process data from the UDP device
- Sending LIMA commands (e.g. trigger commands) via TCP/IP and receiving LIMA answers

4.1 Receiving Process Data from TCP Device

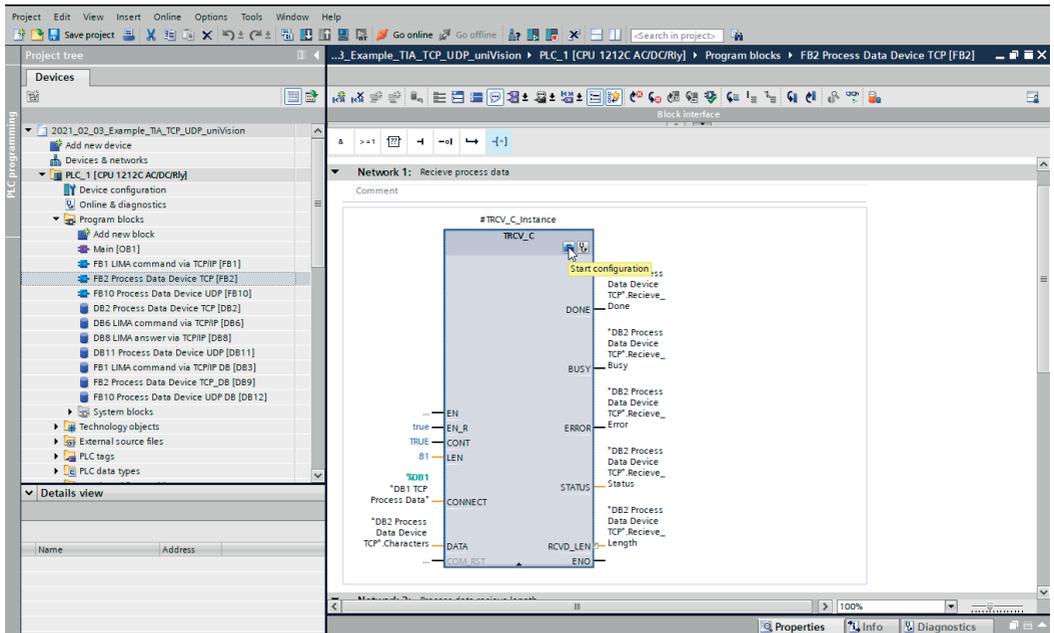
The TIA sample program is created with the following network setting for the uniVision product:

- IP address: 192.168.100.1
- Subnet mask: 255.255.255.0

The TCP process data is sent via port 32002 by default.

If a different network setting or another port is used on the uniVision product, the sample program must be adapted accordingly.

To do this, open the function module “FB2 Process Data Device TCP” and click on “Start Configuration” on network 1 “Receive process data”.



The screenshot displays the Siemens TIA Portal interface for configuring the 'FB2 Process Data Device TCP' block. The main workspace shows the block diagram for 'Network 1: Receive process data'. The block is labeled '#TRCV_C_Instance' and 'TRCV_C'. A yellow callout box points to the 'Start configuration' button on the block. The diagram shows the following connections:

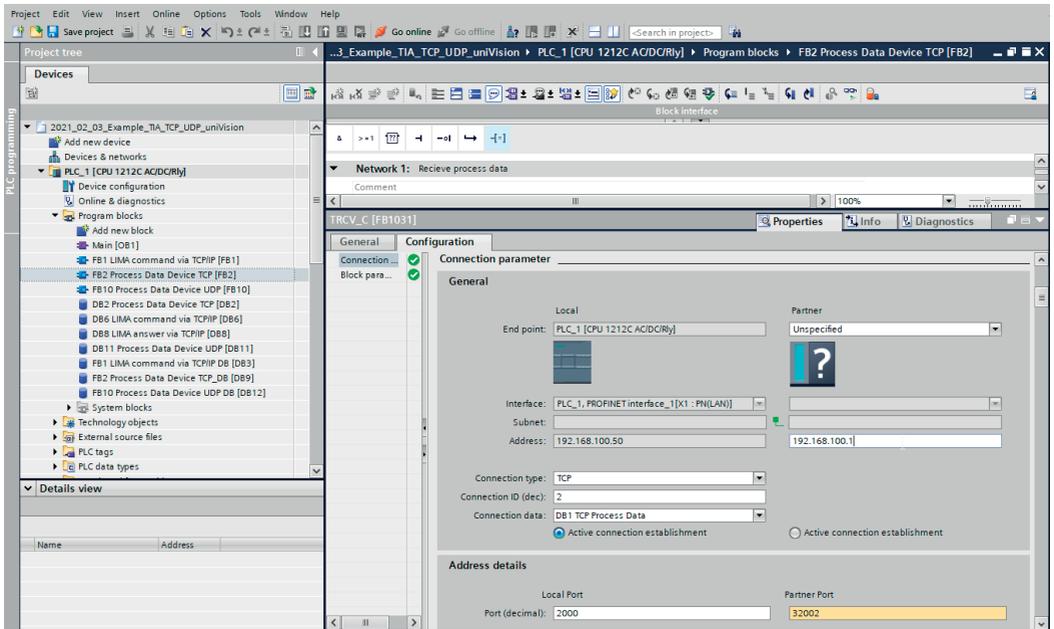
- Inputs:**
 - EN: true
 - EN_R: TRUE
 - COAT: 81
 - LEN: DB1 TCP Process Data
 - CONNECT: DB2 Process Data Device TCP Characters
 - DATA: DB2 Process Data Device TCP Characters
 - COM_RST: COM_RST
- Outputs:**
 - DONE: Done
 - BUSY: Busy
 - ERROR: Error
 - STATUS: Status
 - RCVD_LEN: Length
 - ENO: ENO

The left sidebar shows the project tree with the following structure:

- 2021_02_03_Example_TIA_TCP_UDP_uniVision
 - Add new device
 - Devices & networks
 - PLC_1 [CPU 1212C AC/DC/Rly]
 - Device configuration
 - Online & diagnostics
 - Program blocks
 - Add new block
 - Main [OB1]
 - FB1 LIMA command via TCP/IP [FB1]
 - FB2 Process Data Device TCP [FB2]**
 - FB10 Process Data Device UDP [FB10]
 - DB2 Process Data Device TCP [DB2]
 - DB6 LIMA command via TCP/IP [DB6]
 - DB8 LIMA answer via TCP/IP [DB8]
 - DB11 Process Data Device UDP [DB11]
 - FB1 LIMA command via TCP/IP DB [DB3]
 - FB2 Process Data Device TCP_DB [DB9]
 - FB10 Process Data Device UDP DB [DB12]
 - System blocks
 - Technology objects
 - External source files
 - PLC tags
 - PLC data types

The bottom of the window shows the 'Details view' with columns for Name and Address, and a status bar with 'Properties', 'Info', and 'Diagnostics' buttons.

Enter the IP address and port under “Partner”.



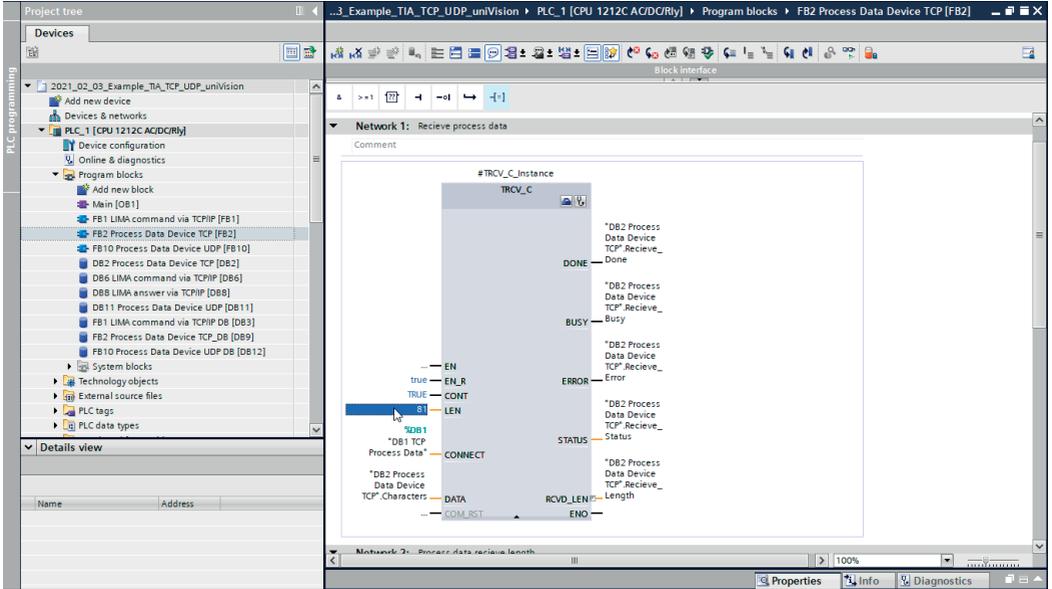
The TIA sample program is created for process data with a length of 81 characters. If a different character count is required, the sample program must be adapted accordingly.

NOTE!

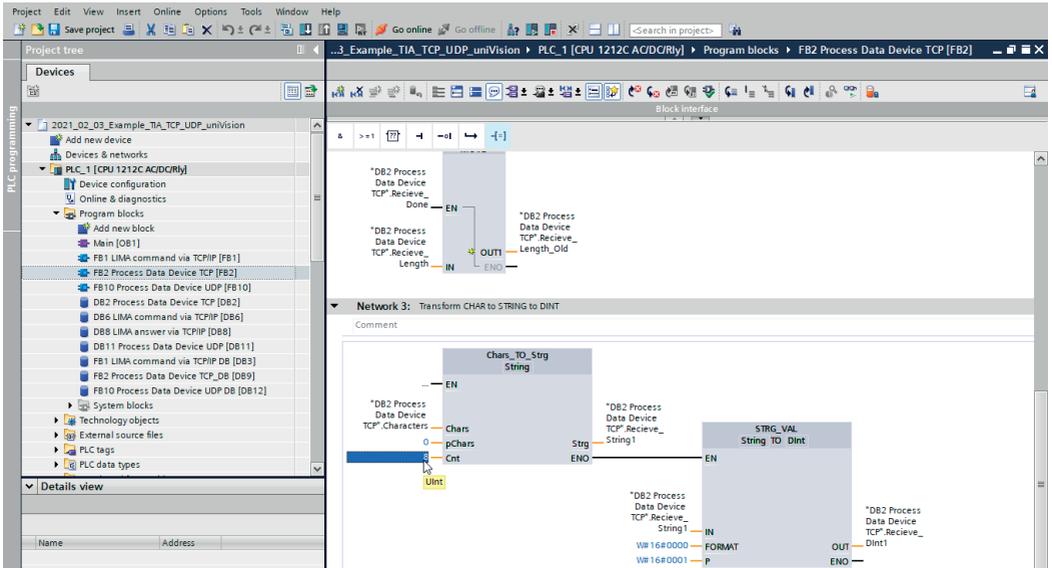


The total number of characters sent via TCP as process data can be determined in the uniVision software on the TCP device under “Output” (see section 3 “Settings in uniVision” on page 5). Preamble, separator and postamble as well as prefixes must be included in the character count!

To do this, adjust the character count under “LEN” on network 1 “Receive process data”.



The sample program also includes direct conversion of the characters into an integer (DINT) for the first string. The number of characters or data type for the first integer can be changed as desired.



Compile the sample program, load it onto the control system and connect it online.

The “DB2 Process Data Device TCP” data block receives the process data sent by the TCP device. The data is received as individual characters (Char).

The screenshot shows the configuration of the DB2 Process Data Device TCP data block. The table below represents the data block structure:

Name	Data type	Start value	Monitor value	Retain	Accessible f...	Write...	Visible in ...	Setpoint
Static								
Characters	Array[0..100...]							
Characters[0]	Char	'0'						
Characters[1]	Char	'0'						
Characters[2]	Char	'5'						
Characters[3]	Char	'5'						
Characters[4]	Char	'5'						
Characters[5]	Char	'5'						
Characters[6]	Char	'0'						
Characters[7]	Char	'0'						
Characters[8]	Char	'0'						
Characters[9]	Char	'9'						
Characters[10]	Char	'9'						
Characters[11]	Char	'9'						
Characters[12]	Char	'5'						
Characters[13]	Char	'5'						
Characters[14]	Char	'5'						
Characters[15]	Char	'0'						
Characters[16]	Char	'0'						
Characters[17]	Char	'0'						
Characters[18]	Char	'9'						
Characters[19]	Char	'9'						
Characters[20]	Char	'5'						
Characters[21]	Char	'5'						
Characters[22]	Char	'5'						
Characters[23]	Char	'5'						
Characters[24]	Char	'0'						
Characters[25]	Char	'0'						
Characters[26]	Char	'0'						
Characters[27]	Char	'9'						
Characters[28]	Char	'9'						

For the first string, conversion to another data type is shown on the DINT for example purposes.

The screenshot shows the configuration of the DB2 Process Data Device TCP data block with various data types. The table below represents the data block structure:

Name	Data type	Start value	Monitor value	Retain	Accessible f...	Write...	Visible in ...	Setpoint
Static								
Characters	Array[0..100] of Char							
Receive_Done	Bool	false	FALSE					
Receive_Busy	Bool	false	TRUE					
Receive_Error	Bool	false	FALSE					
Receive_Status	Word	16#0	16#7D02					
Receive_Length	Int	0	0					
Receive_Length_Old	Int	0	81					
Receive_String1	String		'-0009#81'					
Receive_Dint1	Dint	0	5081					

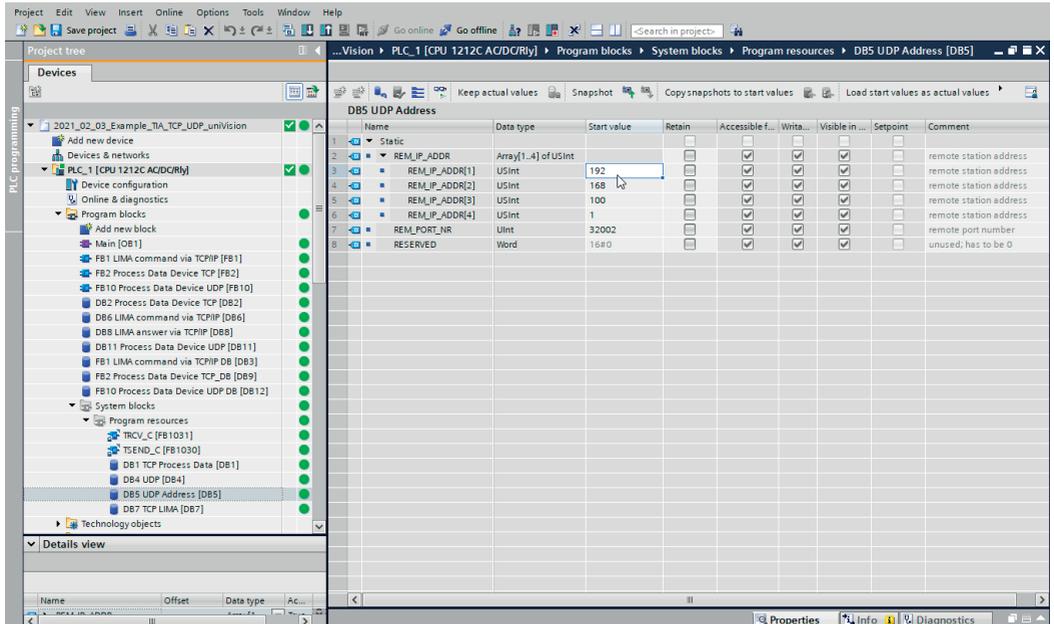
4.2 Receiving Process Data from UDP Device

The TIA sample program is created with the following network setting for the uniVision product:

- IP address: 192.168.100.1
 - Subnet mask: 255.255.255.0
- UDP process data is sent via port 32002.

If a different network setting is used on the uniVision product, the sample program must be adapted accordingly.

To do this, open the data block “DB5 UDP Address” under “System blocks” and enter the IP address of the uniVision product under “REM_IP_ADDR”.



Name	Data type	Start value	Retain	Accessible f...	Write...	Visible in ...	Setpoint	Comment
Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
REM_IP_ADDR	Array[1..4] of USint		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	remote station address
REM_IP_ADDR[1]	USint	192	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	remote station address
REM_IP_ADDR[2]	USint	168	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	remote station address
REM_IP_ADDR[3]	USint	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	remote station address
REM_IP_ADDR[4]	USint	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	remote station address
REM_PORT_NR	UInt	32002	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	remote port number
RESERVED	Word	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	unused; has to be 0

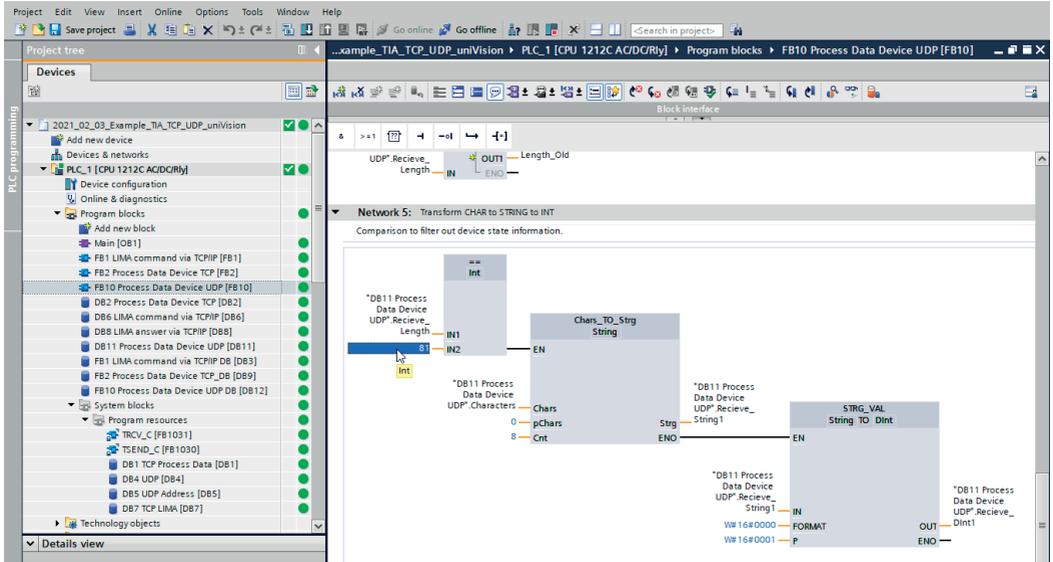
The TIA sample program is created for process data with a length of 81 characters. If a different character count is required, the sample program must be adapted accordingly.

NOTE!

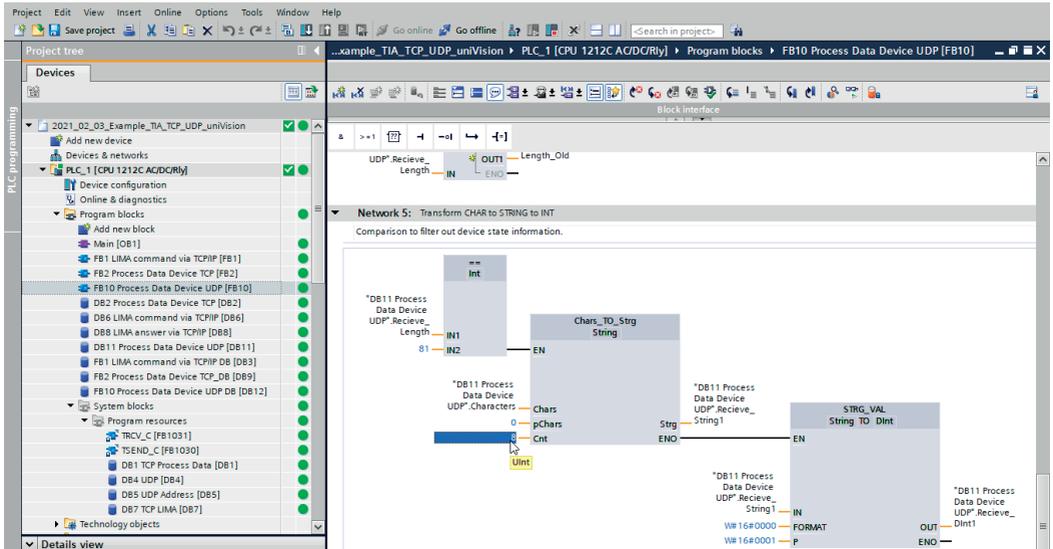


The total number of characters sent via UDP as process data can be determined in the uniVision software on the UDP device under “Output” (see section 3 “Settings in uniVision” on page 5). Preamble, separator and postamble as well as prefixes must be included in the character count!

To do this, adjust the character count under "IN2" in the function module "FB10 Process Data Device UDP" on network 5 "Transform CHAR to STRING to INT".

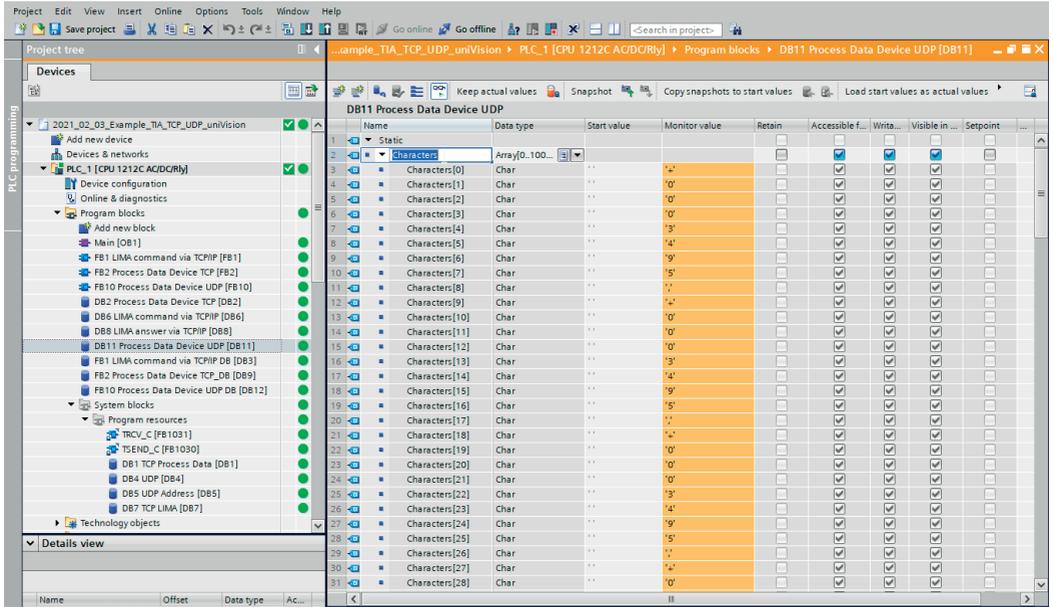


The sample program also includes direct conversion of the characters into an integer (DINT) for the first string. The number of characters or data type for the first integer can be changed as desired.



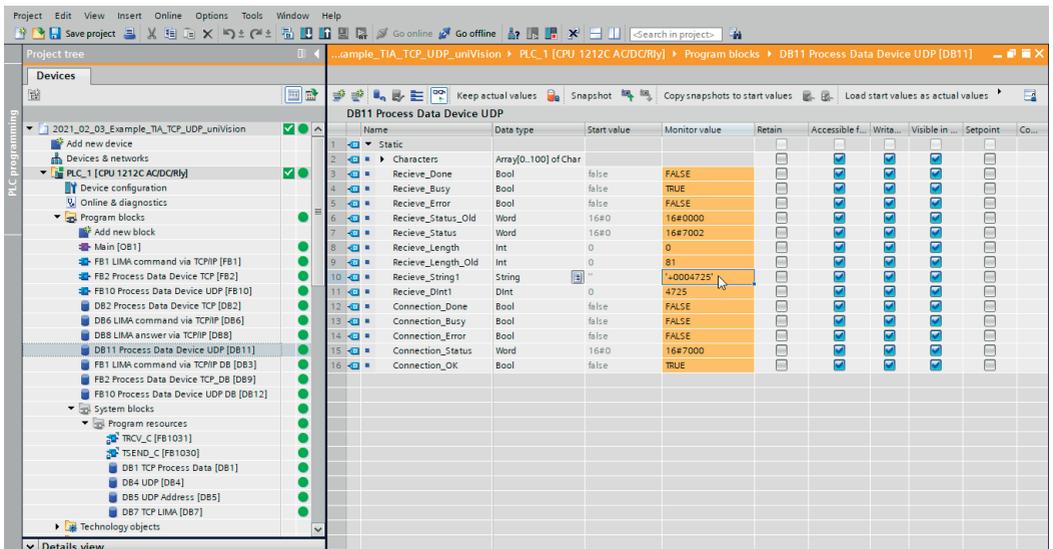
Compile the sample program, load it onto the control system and connect it online.

The “DB11 Process Data Device UDP” data block receives the process data sent by the UDP device. The data is received as individual characters (Char).



Name	Data type	Start value	Monitor value	Retain	Accessible f...	Writes	Visible in ...	Setpoint
Static	Array[0..100]							
Characters[0]	Char	''	'.'					
Characters[1]	Char	''	'.'					
Characters[2]	Char	''	'0'					
Characters[3]	Char	''	'0'					
Characters[4]	Char	''	'3'					
Characters[5]	Char	''	'4'					
Characters[6]	Char	''	'9'					
Characters[7]	Char	''	'5'					
Characters[8]	Char	''	'7'					
Characters[9]	Char	''	'2'					
Characters[10]	Char	''	'2'					
Characters[11]	Char	''	'0'					
Characters[12]	Char	''	'0'					
Characters[13]	Char	''	'3'					
Characters[14]	Char	''	'4'					
Characters[15]	Char	''	'9'					
Characters[16]	Char	''	'5'					
Characters[17]	Char	''	''					
Characters[18]	Char	''	'.'					
Characters[19]	Char	''	'0'					
Characters[20]	Char	''	'0'					
Characters[21]	Char	''	'0'					
Characters[22]	Char	''	'9'					
Characters[23]	Char	''	'2'					
Characters[24]	Char	''	'9'					
Characters[25]	Char	''	'5'					
Characters[26]	Char	''	''					
Characters[27]	Char	''	'.'					
Characters[28]	Char	''	'0'					

For the first string, conversion to another data type is shown on the DINT for example purposes.

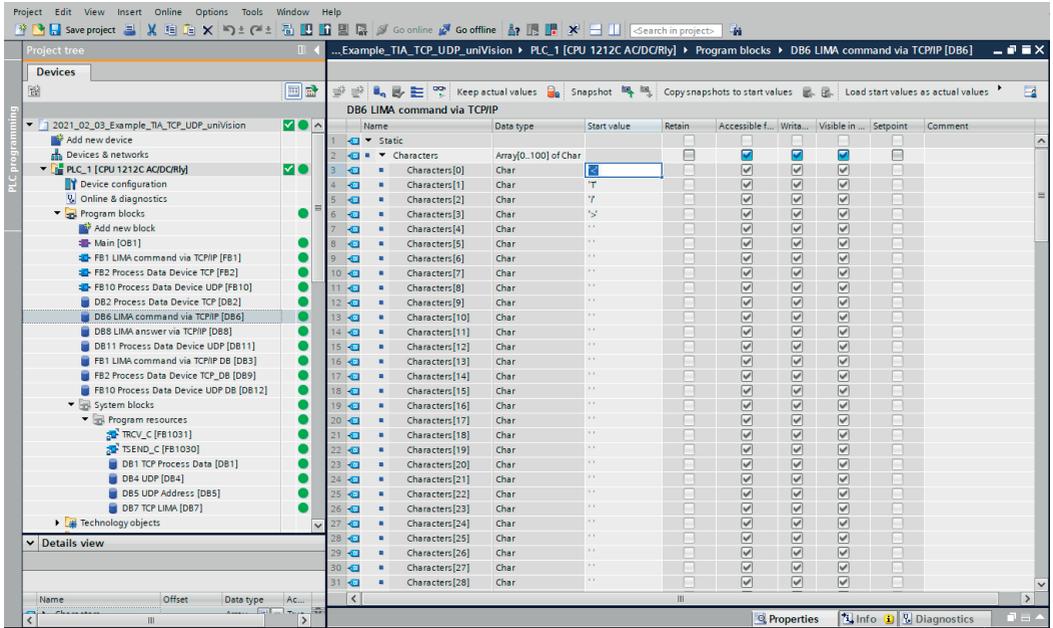


Name	Data type	Start value	Monitor value	Retain	Accessible f...	Writes	Visible in ...	Setpoint
Static	Array[0..100] of Char							
Recvie_Done	Bool	false	FALSE					
Recvie_Busy	Bool	false	TRUE					
Recvie_Error	Bool	false	FALSE					
Recvie_Status_Old	Word	16#0	16#0000					
Recvie_Status	Word	16#0	16#7002					
Recvie_Length	Int	0	0					
Recvie_Length_Old	Int	0	81					
Recvie_String1	String		'16#004725'					
Recvie_Dint1	Dint	0	4725					
Connection_Done	Bool	false	FALSE					
Connection_Busy	Bool	false	FALSE					
Connection_Error	Bool	false	FALSE					
Connection_Status	Word	16#0	16#7000					
Connection_OK	Bool	false	TRUE					

4.3 Sending LIMA Commands via TCP/IP and Receiving LIMA Answers

LIMA commands can be sent via the TCP/IP interface. In the sample program, a trigger command is sent to the uniVision product, which triggers an image or profile recording. Details on the commands available can be found in the LIMA interface protocol. It is available in the download area of the uniVision product detail page (<https://www.wenglor.com/product/DNNF020>).

The LIMA command must be entered with individual characters under “DB6 LIMA command via TCP/IP”. <T/> must be sent for the trigger command.

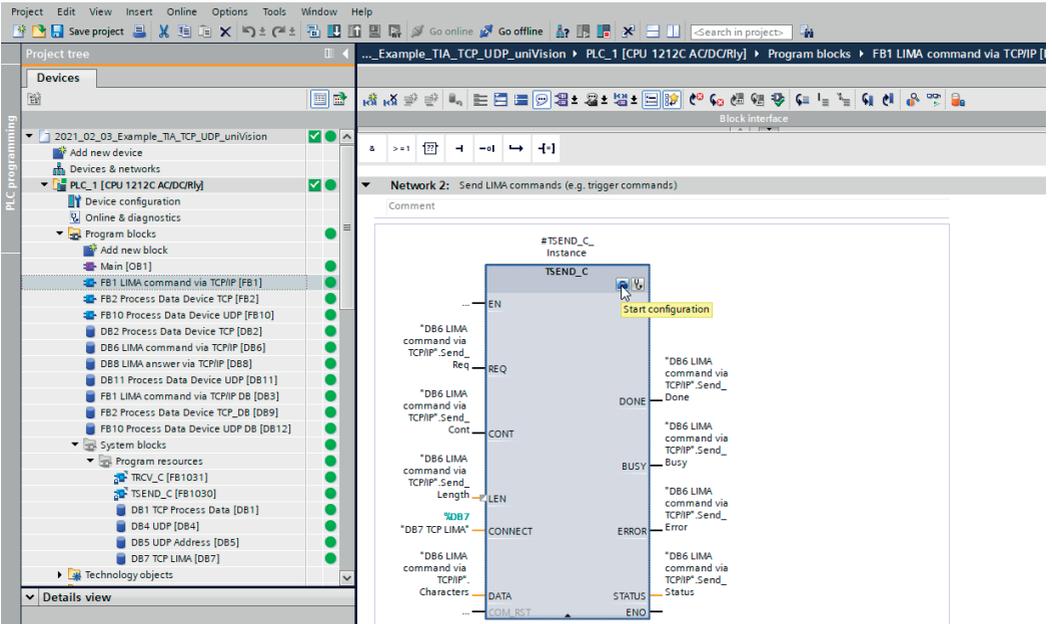


The TIA sample program is created with the following network setting for the uniVision product:

- IP address: 192.168.100.1
 - Subnet mask: 255.255.255.0
- LIMA commands are sent via port 32001.

If a different network setting is used on the uniVision product, the sample program must be adapted accordingly.

To do this, open the function module “FB1 LIMA command via TCP/IP” and click on “Start Configuration” on network 2 “Send LIMA commands (e.g. trigger commands)”.

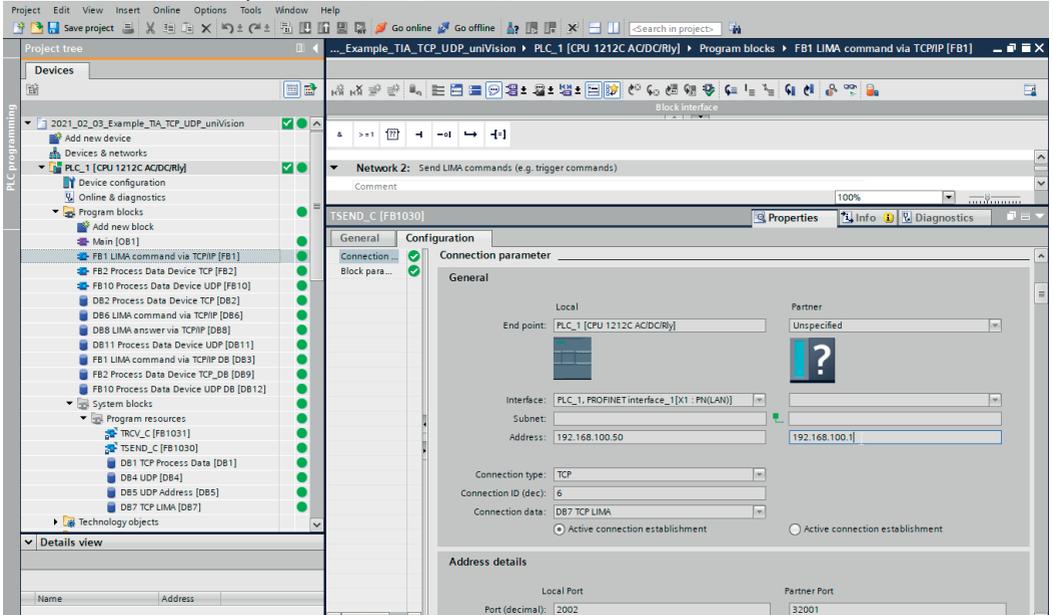


The screenshot displays the SIMATIC Manager interface. On the left, the 'Project tree' shows the hierarchy: 2021_02_03_Example_TIA_TCP_UDP_uniVision > PLC_1 [CPU 1212C AC/DC/Rly] > Program blocks > FB1 LIMA command via TCP/IP. The 'Main [OB1]' block is selected, and the 'FB1 LIMA command via TCP/IP [FB1]' block is highlighted in the 'Program blocks' list.

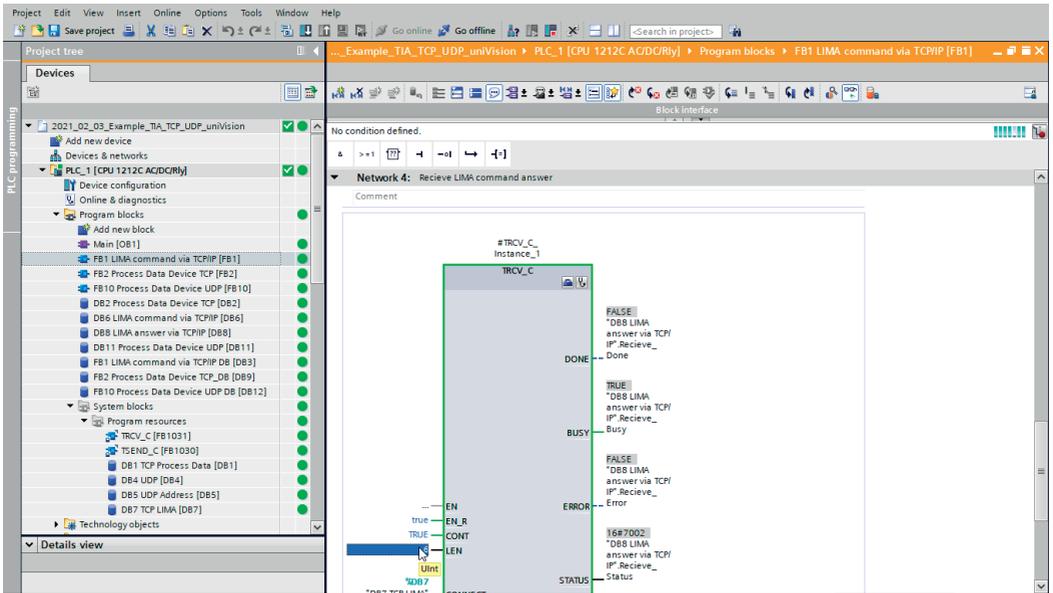
The main window shows the 'Network 2: Send LIMA commands (e.g. trigger commands)' configuration. A 'Start configuration' button is visible on the right side of the network diagram. The diagram shows the following connections:

- EN**: Start signal
- REQ**: "DB6 LIMA command via TCP/IP".Send_Req
- CONT**: "DB6 LIMA command via TCP/IP".Cont
- LEN**: "DB7 TCP LIMA".Length
- CONNECT**: "DB7 TCP LIMA".CONNECT
- DATA**: "DB6 LIMA command via TCP/IP".Characters
- EQM_BST**: End of message
- DONE**: "DB6 LIMA command via TCP/IP".Send_Done
- BUSY**: "DB6 LIMA command via TCP/IP".Send_Busy
- ERROR**: "DB6 LIMA command via TCP/IP".Send_Error
- STATUS**: "DB6 LIMA command via TCP/IP".Send_Status
- ENO**: End of network

Enter the IP address and port 32001 under “Partner”.



Similarly, click on “Start Configuration” on network 4 “Receive LIMA command answer” and enter the IP address and port 32001 again. In addition, the character count of the LIMA answers must be entered on network 4 under “LEN”. The trigger command answer contains 6 characters (<TOK/>).



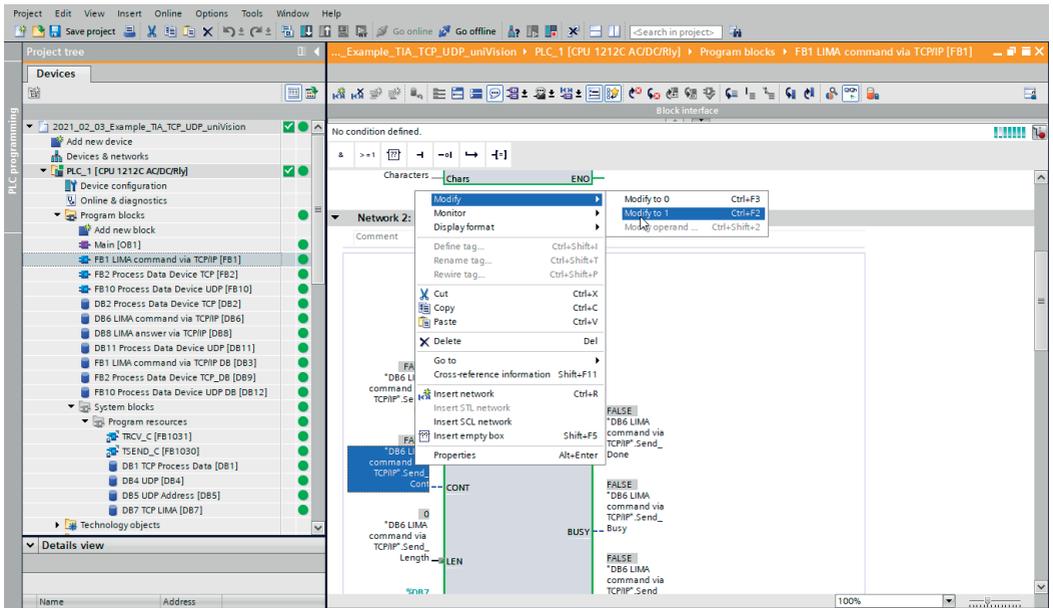
Compile the sample program, load it onto the control system and connect it online.

To send the LIMA command, first establish the connection to the uniVision product. To do this, open the function module “FB1 LIMA command via TCP/IP” and set “Send LIMA commands (e.g. trigger commands)” CONT to 1 on network 2.

NOTE!



The connection can only be established if port 32001 is available for the control system. Depending on the product or operating mode of the uniVision software, port 32001 is also required by the uniVision software (e.g. in editing mode). In this case, the uniVision software must disconnect so that the connection can be established via the control system.



The LIMA command is sent to the uniVision device by setting REQ to 1.

The screenshot displays the SIMATIC Manager interface. The main window shows a ladder logic network with the following elements:

- REQ**: A normally open contact representing the LIMA command request.
- DONE**: A coil representing the completion of the LIMA command.
- BUSY**: A coil representing the busy status of the LIMA command.

A context menu is open over the 'REQ' contact, with the following options visible:

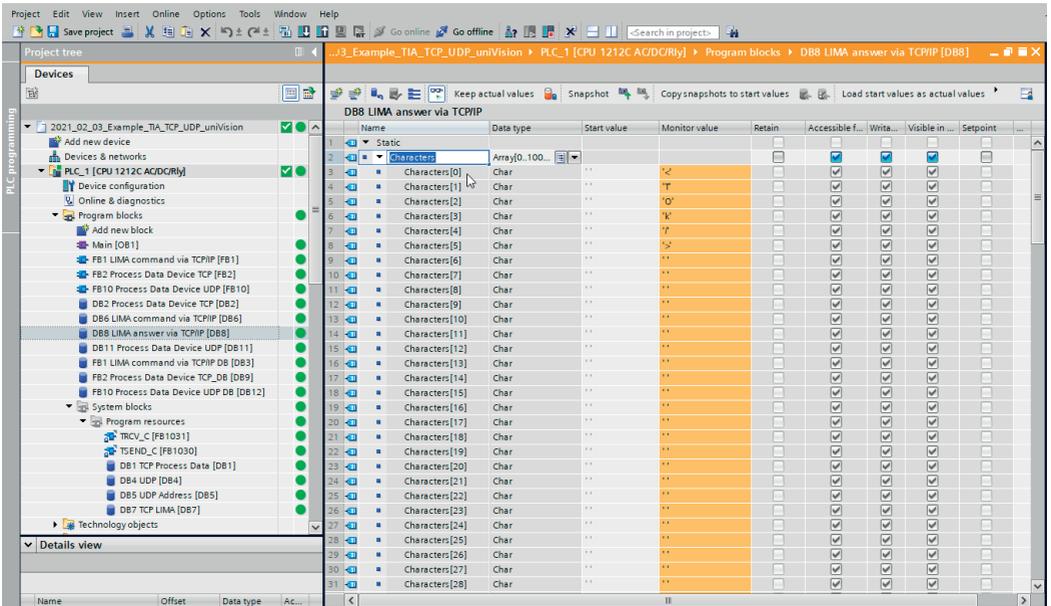
- Modify to 0 (Ctrl+F3)
- Modify to 1 (Ctrl+F2)
- Modify operand... (Ctrl+Shift+2)
- Define tag... (Ctrl+Shift+I)
- Rename tag... (Ctrl+Shift+T)
- Rewire tag... (Ctrl+Shift+P)
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Delete (Del)
- Go to (Shift+F11)
- Cross-reference information (Shift+F11)
- Insert network (Ctrl+R)
- Insert STL network
- Insert SCL network
- Insert empty box (Shift+F5)
- Properties (Alt+Enter)

NOTE!



The LIMA command is reset in the sample program immediately after sending so that only one image or profile is taken from the uniVision product. The associated results for the trigger can be received via the process data. For example, the execution counter can be used to check when the results are available.

The LIMA answer can be received in the data block “DB8 LIMA answer via TCP/IP”. For the trigger command, <Tok/> is sent by the uniVision product in response to a successful execution of the trigger command.



The screenshot shows the SIMATIC Manager interface with the project tree on the left and the variable declaration table for 'DB8 LIMA answer via TCP/IP' in the center. The table lists 29 character variables from 'Characters[0]' to 'Characters[28]'. The 'Data type' column for all entries is 'Char'. The 'Monitor value' column contains asterisks (**). The 'Retain' column has checkboxes, with the first 10 rows checked. The 'Accessible f...' column has icons for various access permissions. The 'Visible in ...' column has icons for visibility in different views. The 'Setpoint' column has checkboxes, with the first 10 rows checked.

Name	Data type	Start value	Monitor value	Retain	Accessible f...	Visible in ...	Setpoint
1	Static						
2	Characters	Array(0..100...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	Characters[0]	Char	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	Characters[1]	Char	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	Characters[2]	Char	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	Characters[3]	Char	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	Characters[4]	Char	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	Characters[5]	Char	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	Characters[6]	Char	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	Characters[7]	Char	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	Characters[8]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	Characters[9]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	Characters[10]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	Characters[11]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15	Characters[12]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	Characters[13]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	Characters[14]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18	Characters[15]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19	Characters[16]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
20	Characters[17]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
21	Characters[18]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
22	Characters[19]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
23	Characters[20]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
24	Characters[21]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
25	Characters[22]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
26	Characters[23]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
27	Characters[24]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
28	Characters[25]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
29	Characters[26]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
30	Characters[27]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
31	Characters[28]	Char	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

5. TwinCAT3 Sample Programs

The TwinCAT3 sample programs for UDP and TCP include the following use cases:

- Receiving process data from the TCP device (in the TCP sample program)
- Receiving process data from the UDP device (in the UDP sample program)
- Sending LIMA commands (e.g. trigger commands) via TCP/IP and receiving LIMA response (in the TCP sample program)

In the example, the following network configuration is used:

- PC with TwinCAT3:
 - IP address: 192.168.100.181
 - Subnet mask: 255.255.255.0
- uniVision product:
 - IP address: 192.168.100.1
 - Subnet mask: 255.255.255.0



NOTE!

To do this, the latest TwinCAT3 version must be installed, including the TF6310 TC3 TCP/IP module. For details, please contact Beckhoff support.

5.1 Receiving Process Data from TCP Device

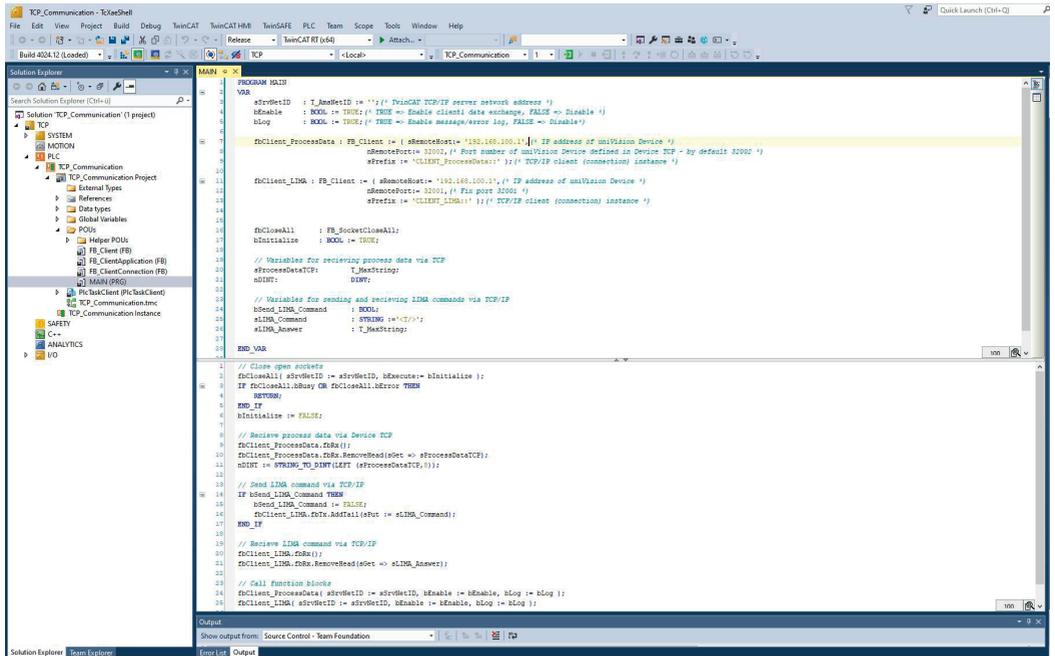
The sample program is created with the following network setting for the uniVision product:

- IP address: 192.168.100.1
- Subnet mask: 255.255.255.0

The TCP process data is sent via port 32002 by default.

If a different network setting or another port is used on the uniVision product, the sample program must be adapted accordingly.

To do this, enter the IP address of the uniVision product under “sRemoteHost” and the port under “nRemotePort” in the MAIN of TCP_Communication under fbClient_ProcessData.



```

PROGRAM MAIN
VAR
  sRemoteID : T_AnnetID := ''; (* TwinCAT TCP/IP server network address *)
  bEnable   : BOOL := TRUE; (* TRUE => Enable client data exchange, FALSE => Disable *)
  bLog      : BOOL := TRUE; (* TRUE => Enable message/error log, FALSE => Disable *)
END_VAR

fbClient_ProcessData : FB_Client := ( sRemoteHost := '192.168.100.1', (* IP address of uniVision Device *)
                                     sRemotePort := 32002, (* Port number of uniVision Device defined in Device TCP - by default 32002 *)
                                     sPrefix := 'CLIENT_ProcessData', (* TCP/IP client (connection) instance *)
                                     sRemoteHost := '192.168.100.1', (* IP address of uniVision Device *)
                                     sRemotePort := 32002, (* Port number of uniVision Device *)
                                     sPrefix := 'CLIENT_LINA', (* TCP/IP client (connection) instance *)
                                     sFCloseAll : FB_SocketCloseAll := ( sFCloseAll := FB_SocketCloseAll;
                                                                       bInitialize : BOOL := TRUE;
                                                                       // Variables for receiving process data via TCP
                                                                       sProcessDataTCP : T_Message := ' ';
                                                                       sData : STRING := ' ';
                                                                       // Variables for sending and receiving LINA commands via TCP/IP
                                                                       sLINA_Command : STRING := ' ';
                                                                       sLINA_Answer : STRING := ' ';
                                                                       sLINA_Answer : T_Message := ' ');
                                     END_VAR

// Close open sockets
sFCloseAll(sRemoteID := sRemoteID, bResource := Initialize);
IF sFCloseAll.sBusy OR sFCloseAll.sError THEN
  RETURN;
END_IF
bInitialize := FALSE;

// Receive process data via Device TCP
fbClient_ProcessData.sData := sProcessDataTCP;
nDIPT := sProcessDataTCP.Len;

// Send LINA command via TCP/IP
IF sLINA_Command THEN
  fbClient_LINA.sData := sLINA_Command;
END_IF

// Receive LINA command via TCP/IP
fbClient_LINA.sData := sLINA_Answer;

// Call function blocks
fbClient_ProcessData(sRemoteID := sRemoteID, bEnable := bEnable, bLog := bLog);
fbClient_LINA(sRemoteID := sRemoteID, bEnable := bEnable, bLog := bLog);
  
```

The sample program also includes direct conversion of the first eight characters into an integer (DINT) for the first string. The number of characters or data type can be changed as desired.

Enable the sample program, log in and start it. The process data sent by the TCP device appears under the variables "sProcessDataTCP". The data for the first DINT appears under "nDINT".

The screenshot displays the SIMATIC Manager interface for a project named "TCP_Communication". The left pane shows the project tree with the "MAIN (PDR)" folder selected. The main workspace is divided into two sections: a variable declaration table and a ladder logic editor.

Expression	Type	Value	Prepared value	Address	Comment
sSrvNetID	T_AnsNetID	-			TwinCAT TCP/IP server network address
isEnable	BOOL	TRUE			TRUE => Enable client_data exchange, FALSE => Enable mess. error log, FALSE ...
sMsg	STRING	TRUE			
fbClient_ProcessData	FB_Client				TCP/IP client (connection) instance
fbClient_LINA	FB_Client				TCP/IP client (connection) instance
fbClientInst	FB_SocketClient				
bInitialize	BOOL	FALSE			
sProcessDataTCP	T_MemString	+0594672,+001437L,+0594672,+0594672,+0594672,+0594672,+0594672,+0594672;+0594672;+0594672;			Variables for receiving process data via TCP
nDINT	DINT	0x0020			
bSend_LINA_Command	BOOL	FALSE			Variable for sending ... receiving LINA com...
sLINA_Command	STRING	<TCP>			
sLINA_Answer	T_MemString	-			


```

1 // Close open sockets
2 bCloseAll := sSrvNetID == sSrvNetID; bExecute[FALSE] := bInitialise[FALSE];
3 bCloseAll := bCloseAll AND bCloseAll.bError[FALSE] AND bCloseAll.bError[FALSE];
4 bCloseAll := bCloseAll AND bCloseAll.bError[FALSE] AND bCloseAll.bError[FALSE];
5 bCloseAll := bCloseAll AND bCloseAll.bError[FALSE] AND bCloseAll.bError[FALSE];
6 bCloseAll := bCloseAll AND bCloseAll.bError[FALSE] AND bCloseAll.bError[FALSE];
7 bCloseAll := bCloseAll AND bCloseAll.bError[FALSE] AND bCloseAll.bError[FALSE];
8 bCloseAll := bCloseAll AND bCloseAll.bError[FALSE] AND bCloseAll.bError[FALSE];
9 bCloseAll := bCloseAll AND bCloseAll.bError[FALSE] AND bCloseAll.bError[FALSE];
10 // Receive process data via Device TCP
11 fbClient_ProcessData.fbData.RemoveHead(sMsg := sProcessDataTCP[0594672..]);
12 nDINT := sMsg[0..7];
13 // Send LINA command via TCP/IP
14 bSend_LINA_Command := bSend_LINA_Command;
15 fbClient_LINA.data.AddTail(sPut := sLINA_Command + sMsg);
16 bSend_LINA_Command := bSend_LINA_Command;
17 bSend_LINA_Command := bSend_LINA_Command;
18 bSend_LINA_Command := bSend_LINA_Command;
19 // Receive LINA command via TCP/IP
20 fbClient_LINA.fbData();
21 fbClient_LINA.fbData.RemoveHead(sMsg := sLINA_Answer);
22 fbClient_LINA.fbData.RemoveHead(sMsg := sLINA_Answer);
23 // Call function blocks
24 fbClient_ProcessData(sSrvNetID := sSrvNetID, sSrvNetID := sSrvNetID, sSrvNetID := sSrvNetID, sSrvNetID := sSrvNetID);
25 fbClient_LINA(sSrvNetID := sSrvNetID, sSrvNetID := sSrvNetID, sSrvNetID := sSrvNetID, sSrvNetID := sSrvNetID);
26 bExecute := bExecute;

```

The bottom of the screenshot shows the "Error List" pane, which is currently empty, indicating that the code is error-free.

5.2 Receiving Process Data from UDP Device

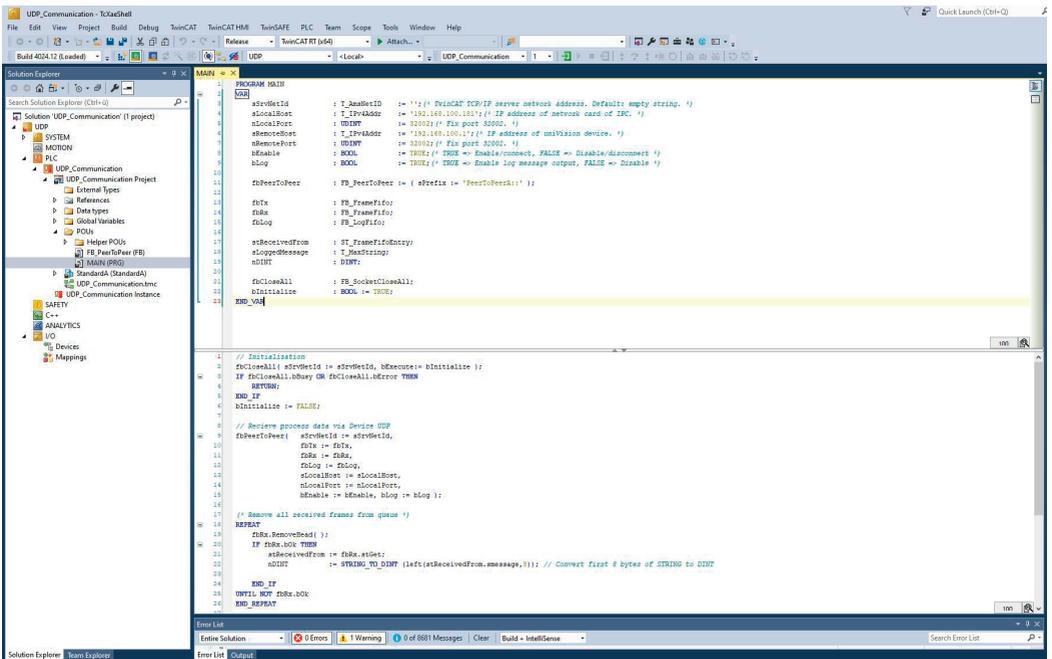
The sample program is created with the following network setting for the uniVision product:

- IP address: 192.168.100.1
- Subnet mask: 255.255.255.0

UDP process data is sent via port 32002.

If a different network setting is used on the uniVision product, the sample program must be adapted accordingly.

To do this, enter the IP address of the uniVision product in the MAIN of UDP_Communication under the variables "sRemoteHost".



```

1  PROGRAM MAIN
2  END
3
4  sRemoteID    := ''; (* TwinCAT TCP/IP server network address. Default: empty string. *)
5  sLocalHost   := '192.168.100.11'; (* IP address of network card of PLC *)
6  sLocalPort   := 32002; (* Fix port 32002. *)
7  sRemoteHost  := '192.168.100.1'; (* IP address of uniVision device. *)
8  sRemotePort  := 32002; (* Fix port 32002. *)
9  sEnable      := TRUE; (* TRUE => Enable/connect, FALSE => Disable/disconnect *)
10 sLog         := TRUE; (* TRUE => Enable log message output, FALSE => Disable *)
11
12 sBPeerToPeer := ( sPrefix := 'PeerToPeer!' );
13
14 sData        := FB_FramePInfo;
15 sData        := FB_FramePInfo;
16 sLog         := FB_LogPInfo;
17
18 sReceivedFrom := ST_FramePInfo;
19 sLogMessage    := T_String;
20 sDINT          := DINT;
21
22 sCloseAll    := FB_SocketCloseAll;
23 sInitialize  := BOOL := TRUE;
24
25 END_MAIN
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
25
```

The sample program also includes direct conversion of the first eight characters into an integer (DINT) for the first string. The number of characters or data type can be changed as desired.

Enable the sample program, log in and start it. The process data sent by the UDP device appears under the variables "stReceivedFrom" -> "sMessage". The data for the first DINT appears under "nDINT".

The screenshot displays the Siemens TwinCAT3 environment. The top window shows the 'MAIN (Online)' configuration for the 'UDP_UDP_Communication.MAIN' project. Below it, the ladder logic code is visible, and the bottom window shows the 'Output' console.

Expression	Type	Value	Prepared value	Address	Comment
stvNetId	T_AnsiStr	-			TwinCAT TCP/IP serve. stwork address. Def...
sLocalHost	T_IPv4Addr	192.168.100.187			IP address of network card of IPC.
sLocalPort	UDINT	32002			Fix port 32002.
sRemoteHost	T_IPv4Addr	192.168.100.1			IP address of un/vision device.
sRemotePort	UDINT	32002			Fix port 32002.
sNetId	BOOL	TRUE			TRUE => EnableNet. FALSE => DisableNet.
sLog	BOOL	TRUE			TRUE => Enable log. age output, FALSE => ...
stPeerToPeer	FB_PeerToPeer				
stTx	FB_FrameTo				
stRx	FB_FrameTo				
stLog	FB_LogTo				
stReceivedFrom	ST_FrameToEntry				Remote address. stin. staining an (IPv4) L...
stSocketAddr	ST_SocketAddr				UDP packet data
sMessage	T_MacString	+0256095,+0256095,+0256095,+0256095,+0256095,+0256095,+0256095,+0256095;			
sLoggedMessage	T_MacString	PeerToPeer::Connectless(UDP) socket created:192.168.100.187:32002			
nDINT	DINT	256095			
stClassId	FB_SocketClass				
stInitialize	BOOL	FALSE			

```

1 // Initialize
2 @ FCIOseAll := stvNetId <> '' && sRemoteHost <> '' && sRemotePort <> '' && sLocalHost <> '' && sLocalPort <> '' && sNetId <> '' && sLog <> '' && stPeerToPeer <> '' && stTx <> '' && stRx <> '' && stLog <> '' && stReceivedFrom <> '' && stSocketAddr <> '' && sMessage <> '' && sLoggedMessage <> '' && nDINT <> '' && stClassId <> '' && stInitialize <> ''
3 IF FCIOseAll THEN
4   sNetId := TRUE
5   sLog := TRUE
6   stPeerToPeer := TRUE
7   stTx := TRUE
8   stRx := TRUE
9   stLog := TRUE
10  stReceivedFrom := stvNetId
11  stSocketAddr := sLocalHost & ':' & sLocalPort
12  stSocketAddr := sRemoteHost & ':' & sRemotePort
13  stSocketAddr := sLocalHost & ':' & sLocalPort
14  stSocketAddr := sLocalHost & ':' & sLocalPort
15  stSocketAddr := sLocalHost & ':' & sLocalPort
16  stSocketAddr := sLocalHost & ':' & sLocalPort
17 // Remove all received frames from queue
18 SBRPRAT
19 @ FBx_RemoveHead();
20 IF FBx_IsEmpty THEN
21   stReceivedFrom := stvNetId
22   nDINT := 0
23   stClassId := FB_SocketClass
24   stInitialize := TRUE
25 END SBRPRAT

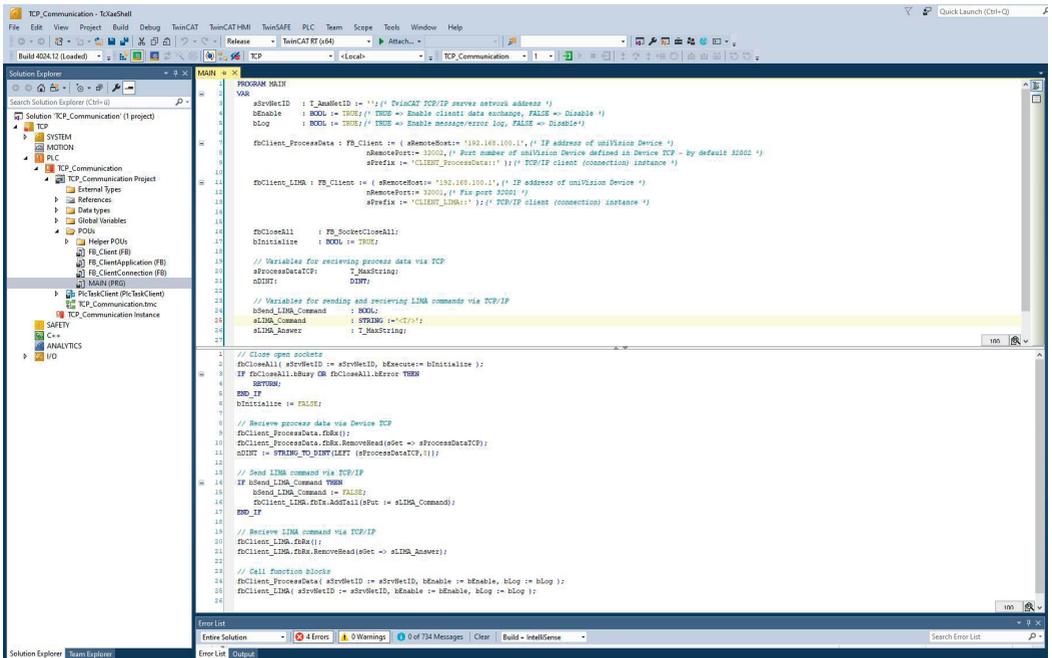
```

Output console shows: Show output from: Source Control - Team Foundation

5.3 Sending LIMA Commands via TCP/IP and Receiving LIMA Answers

LIMA commands can be sent via the TCP/IP interface. In the sample program, a trigger command is sent to the uniVision product, which triggers an image or profile recording. Details on the commands available can be found in the LIMA interface protocol. It is available in the download area of the uniVision product detail page (<https://www.wenglor.com/product/DNNF020>).

The LIMA command must be entered in the MAIN of TCP_Communication under “sLIMA_Command”. <T/> must be sent for the trigger command.



```

TCP_Communication - TCaseShell
File Edit View Project Build Debug TwinCAT TwinCATMM TwinSAFE PLC Team Scope Tools Window Help
Release TwinCAT RT (64) Attach...
Built 404.12 (Loaded) TCP Local TCP_Communication 1
Solution Explorer (Ctrl+Q)
TCP_Communication (1 project)
SYSTEM
MOTION
PLC
TCP_Communication
TCP_Communication Project
External Types
References
Data types
Global Variables
PODs
Helper PODs
FB_Client (FB)
FB_ClientApplication (FB)
FB_ClientConnection (FB)
MAIN (FB)
PictaClient (PicTactClient)
TCP_CommunicationIinc
TCP_Communication Instance
C++
SAFETY
ANALYTICS
ID

PROGRAM MAIN
VAR
  sDrvNetID := ''; (* DriveNet TCP/IP server network address *)
  bEnable := BOOL := TRUE; (* TRUE => Enable client's data exchange, FALSE => Disable *)
  bLog := BOOL := TRUE; (* TRUE => Enable message/error log, FALSE => Disable *)

  sClient_ProcessData := ( sRemoteHost := '192.168.100.1', (* IP address of uniVision Device *)
    sRemotePort := 32002, (* Port number of uniVision Device defined as Device TCP - by default 32002 *)
    sPrefix := 'CLIENT_ProcessData:'); (* TCP/IP client (connection) instance *)

  sClient_LIMA := ( sRemoteHost := '192.168.100.1', (* IP address of uniVision Device *)
    sRemotePort := 32001, (* Port post 32001 *)
    sPrefix := 'CLIENT_LIMA:'); (* TCP/IP client (connection) instance *)

  sCloseAll := FB_SocketCloseAll;
  sInit := TRUE;

  // Variables for receiving process data via TCP
  sProcessDataTCP := T_ManString;
  sDINT := DINT;

  // Variables for sending and receiving LIMA commands via TCP/IP
  sSend_LIMA_Command := BOOL;
  sLIMA_Command := STRING := '<T/>';
  sLIMA_Answer := T_ManString;

  // Close open sockets
  sCloseAll() sDrvNetID := sDrvNetID, bExecute:= bInititalize;
  IF sCloseAllLibary OR sCloseAllLibary THEN
    sSYSTEM;
  END_IF
  sInititalize := FALSE;

  // Receive process data via Device TCP
  sClient_ProcessData.sfx();
  sClient_ProcessData.sfx.RemoveHead(sdet := sProcessDataTCP);
  sDINT := SYSTEM_TO_DINTLEFT (sProcessDataTCP);

  // Send LIMA command via TCP/IP
  IF sSend_LIMA_Command THEN
    bSend_LIMA_Command := FALSE;
    sClient_LIMA.sfxTa.AddTail(sfx := sLIMA_Command);
  END_IF

  // Receive LIMA command via TCP/IP
  sClient_LIMA.sfx();
  sClient_LIMA.sfx.RemoveHead(sdet := sLIMA_Answer);

  // Call function blocks
  sClient_ProcessData() sDrvNetID := sDrvNetID, bEnable := bEnable, bLog := bLog;
  sClient_LIMA() sDrvNetID := sDrvNetID, bEnable := bEnable, bLog := bLog;
END

```

Error List: Entire Solution - 4 Errors 0 Warnings 0 of 734 Messages Clear Build - IntelliSense Search Error List

The sample program is created with the following network setting for the uniVision product:

- IP address: 192.168.100.1
- Subnet mask: 255.255.255.0

LIMA commands are sent via port 32001.

If a different network setting is used on the uniVision product, the sample program must be adapted accordingly.

To do this, enter the IP address of the uniVision product under “sRemoteHost” in fbClient_LIMA.

Enable the sample program, log in and start it.

NOTE!



Connection from the control unit to the uniVision product can only be established if port 32001 is available for the control system. Depending on the product or operating mode of the uniVision software, port 32001 is also required by the uniVision software (e.g. in editing mode). In this case, the uniVision software must disconnect so that the connection can be established via the control system.

The LIMA command is sent to the uniVision product by setting “bSend_LIMA_Command” to TRUE. The command may only be sent once, not sent constantly, so that only one image or profile is recorded. A new command must not be sent until the LIMA answer to the previous command has been received.

6. Rockwell Sample Programs

The Rockwell sample programs for process data and LIMA include the following application cases:

- Receiving process data from the TCP device (in the sample program Example_Rockwell_ProcessData.ACD)
- Receiving process data from the UDP device (in the sample program Example_Rockwell_ProcessData.ACD)
- Sending LIMA commands (e.g., trigger commands) via TCP/IP and receiving the LIMA response (in the sample program Example_Rockwell_LIMA.ACD)

In the example, the following network configuration is used:

- PLC:
 - IP address: 192.168.100.70
 - Subnet mask: 255.255.255.0
- uniVision product:
 - IP address: 192.168.100.1
 - Subnet mask: 255.255.255.0



NOTE!

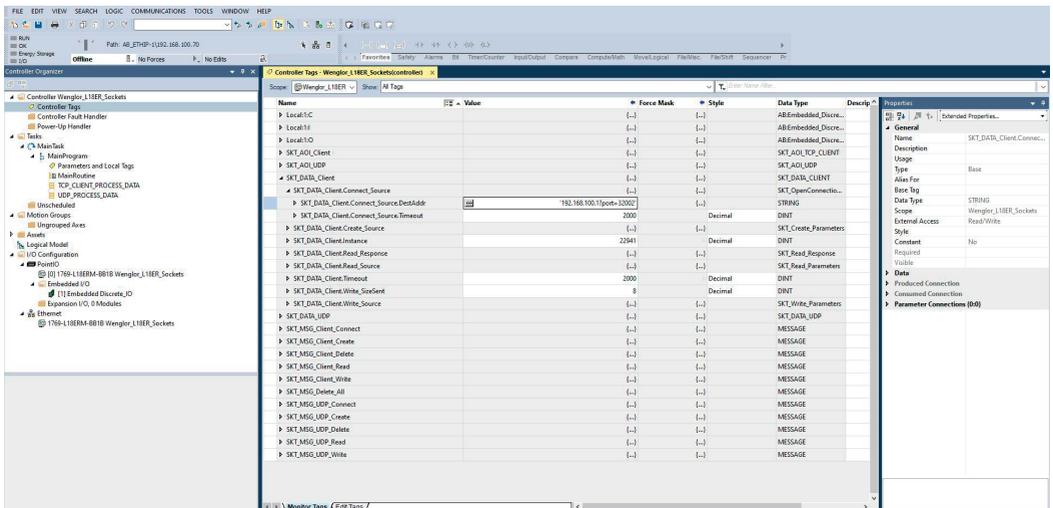
The sample program is created with an Allen-Bradley 1769-L18ERM-BB1B PLC using Studio 5000 Logix Designer V32.

6.1 Receiving Process Data from the TCP Device

The sample program Example_Rockwell_ProcessData.ACD is created with the following network setting for the uniVision product:

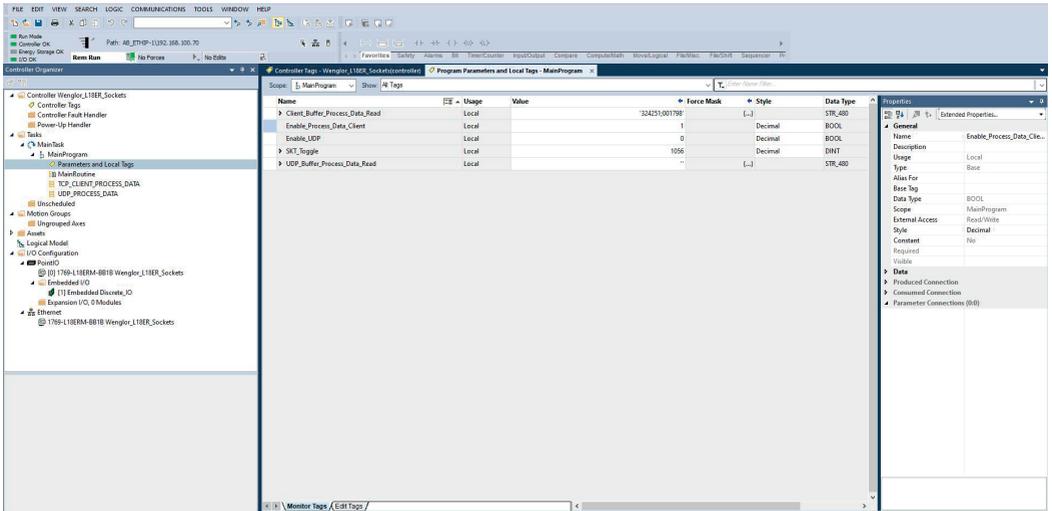
- IP address: 192.168.100.1
- Subnet mask: 255.255.255.0

The TCP process data are sent via port 32002 by default.



Transfer the sample program to the controller and go online.

The TCP connection is established by activating the value Enable_Process_Data_Client under Parameters and Local Tags. The process data sent by the TCP device appear under the Client_Buffer_Process_Data_Read.



The screenshot shows the 'Program Parameters and Local Tags' window in SIMATIC Manager. The table below represents the data shown in the window:

Name	Usage	Value	Force Mask	Style	Data Type
Client_Buffer_Process_Data_Read	Local	124251.001708	[...]		STR_480
Enable_Process_Data_Client	Local	1		Decimal	BOOL
Enable_UDP	Local	0		Decimal	BOOL
SRV_Suppe	Local	1056		Decimal	DINT
UDP_Buffer_Process_Data_Read	Local	-	[...]		STR_480

The right-hand pane shows the 'Properties' window for the 'Enable_Process_Data_Client' tag, with the 'General' tab selected. The 'Name' is 'Enable_Process_Data_Client', 'Description' is 'Local', 'Type' is 'Base', and 'Data Type' is 'BOOL'. Other properties like 'Scope', 'External Access', and 'Style' are also visible.

6.2 Receiving Process Data from the UDP Device

The sample program Example_Rockwell_ProcessData.ACD is created with the following network setting for the uniVision product:

- IP address: 192.168.100.1
- Subnet mask: 255.255.255.0

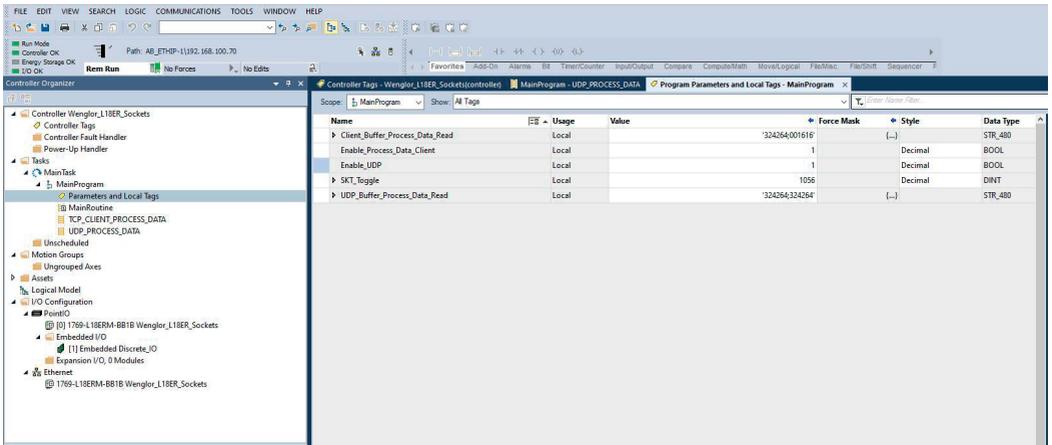
The UDP process data are sent via port 32002.

If a different network setting is used on the uniVision product, the sample program must be adapted accordingly. To do so, open the controller tags and enter the IP address under SKT_DATA_UDP.Connect_Source.DestAddr.

Name	Value	Force Mask	Style	Data Type	Description
Local:IC		(-)	(-)	AB:Embedded_Discrete	
Local:IS		(-)	(-)	AB:Embedded_Discrete	
Local:IO		(-)	(-)	AB:Embedded_Discrete	
SKT_AOI_Client		(-)	(-)	SKT_AOI_CLIENT	
SKT_AOI_UDP		(-)	(-)	SKT_AOI_UDP	
SKT_DATA_Client		(-)	(-)	SKT_DATA_CLIENT	
SKT_DATA_UDP		(-)	(-)	SKT_DATA_UDP	
SKT_DATA_UDP.Connect_Source		(-)	(-)	SKT_OpenConnectio...	
SKT_DATA_UDP.Connect_Source.DestAddr	192.168.100.1			STRING	
SKT_DATA_UDP.Connect_Source.Timeout	2000			Decimal	
SKT_DATA_UDP.Create_Source		(-)	(-)	SKT_Create_Parameters	
SKT_DATA_UDP.Instance	20875			Decimal	
SKT_DATA_UDP.Read_Response		(-)	(-)	SKT_Read_Response	
SKT_DATA_UDP.Read_Source		(-)	(-)	SKT_Read_Parameters	
SKT_DATA_UDP.Timeout	2000			Decimal	
SKT_DATA_UDP.Write_SizeGain	24			Decimal	
SKT_DATA_UDP.Write_Source		(-)	(-)	SKT_Write_Parameters	
SKT_MSG_Client.Connect		(-)	(-)	MESSAGE	
SKT_MSG_Client.Create		(-)	(-)	MESSAGE	
SKT_MSG_Client.Delete		(-)	(-)	MESSAGE	
SKT_MSG_Client.Read		(-)	(-)	MESSAGE	
SKT_MSG_Client.Write		(-)	(-)	MESSAGE	
SKT_MSG_Delete_All		(-)	(-)	MESSAGE	
SKT_MSG_UDP.Connect		(-)	(-)	MESSAGE	
SKT_MSG_UDP.Create		(-)	(-)	MESSAGE	
SKT_MSG_UDP.Delete		(-)	(-)	MESSAGE	
SKT_MSG_UDP.Read		(-)	(-)	MESSAGE	
SKT_MSG_UDP.Write		(-)	(-)	MESSAGE	

Transfer the sample program to the controller and go online.

To receive the UDP process data, activate the value Enable_UDP under Parameters and Local Tags. The process data sent by the UDP device appear under UDP_Buffer_Process_Data_Read.



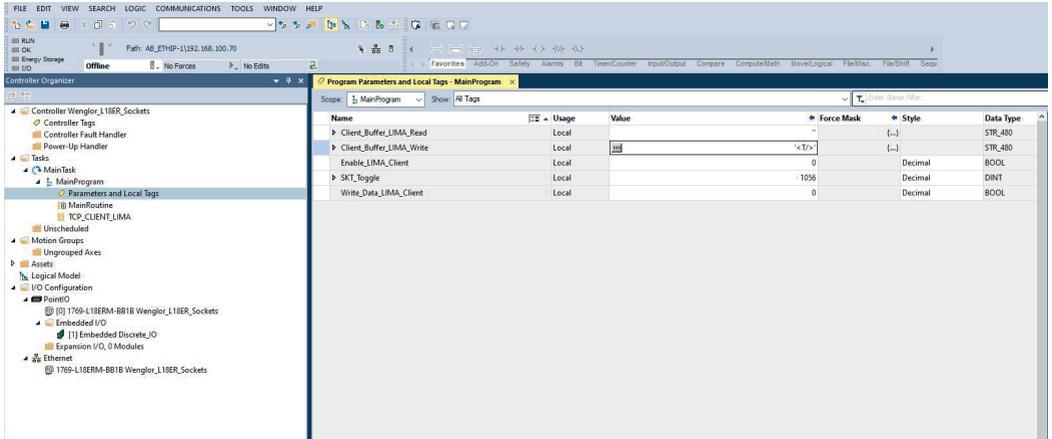
The screenshot shows the Wenglor software interface with the following configuration details:

- Controller Tags - Wenglor_L18ER_Sockets(controller):**
 - Client_Buffer_Process_Data_Read: Local, Value: '324264,001616', Force Mask: 1, Style: (-), Data Type: STR_480
 - Enable_Process_Data_Client: Local, Value: 1, Force Mask: 1, Style: Decimal, Data Type: BOOL
 - Enable_UDP: Local, Value: 1056, Force Mask: 1056, Style: Decimal, Data Type: DINT
 - SKT_Toggle: Local, Value: '324264,324264', Force Mask: (-), Style: (-), Data Type: STR_480
 - UDP_Buffer_Process_Data_Read: Local, Value: '324264,324264', Force Mask: (-), Style: (-), Data Type: STR_480
- Program Parameters and Local Tags - MainProgram:**
 - MainRoutine
 - TCP_CLIENT_PROCESS_DATA
 - UDP_PROCESS_DATA
 - Unscheduled
 - Motion Groups
 - Ungrouped Axes
 - Assets
 - Logical Model
 - I/O Configuration
 - PrintIO
 - 1769-L18ERM-8B1B Wenglor_L18ER_Sockets
 - Embedded I/O
 - 1 Embedded Discrete_IO
 - Expansion I/O, 0 Modules
 - Ethernet
 - 1769-L18ERM-8B1B Wenglor_L18ER_Sockets

6.3 Sending LIMA Commands via TCP/IP and Receiving LIMA Responses

LIMA commands can be sent via the TCP/IP interface. In the sample program Example_Rockwell_LIMA.ACD, a trigger command is sent to the uniVision product, which triggers an image or profile recording. Details on the commands available can be found in the LIMA interface protocol. It is available in the download area of the uniVision product detail page (<https://www.wenglor.com/product/DNNF020>).

The LIMA command must be entered under Client_Buffer_LIMA_Write under Parameters and Local Tags. <T/> must be sent for the trigger command.

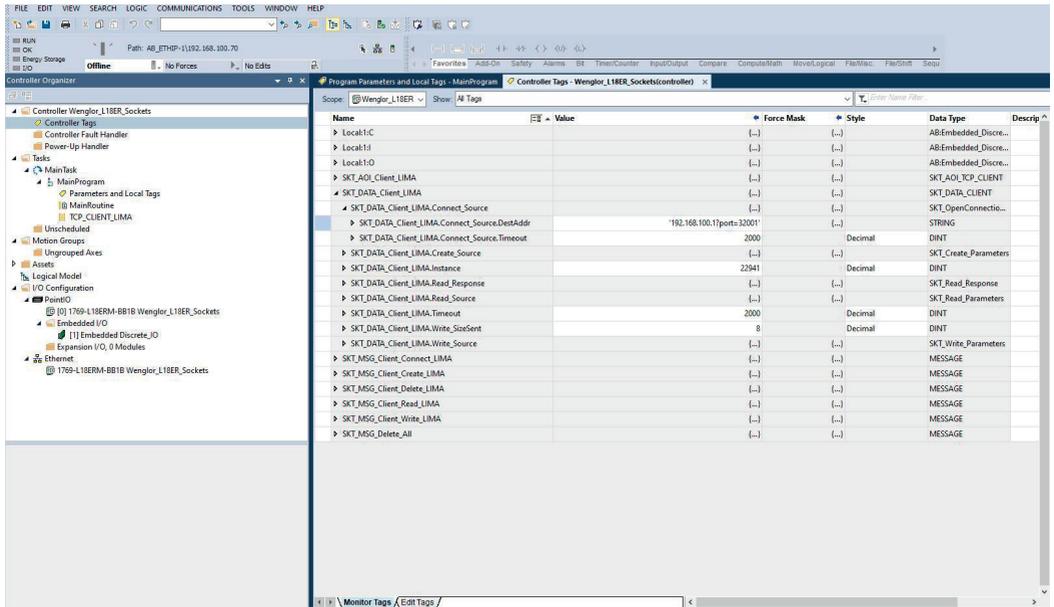


The sample program is created with the following network setting for the uniVision product:

- IP address: 192.168.100.1
- Subnet mask: 255.255.255.0

LIMA commands are sent via port 32001.

If a different network setting is used on the uniVision product, the sample program must be adapted accordingly. To do so, open the controller tags and enter the IP address under SKT_DATA_Client_LIMA.Connect_Source.DestAddr.



Name	Value	Force Mask	Style	Data Type	Descrip
Local:LC		(...)	(...)	AB:Embedded_Discr...	
Local:LI		(...)	(...)	AB:Embedded_Discr...	
Local:LO		(...)	(...)	AB:Embedded_Discr...	
SKT_AOI_Client_LIMA		(...)	(...)	SKT_AOI_TCP_CLIENT	
SKT_DATA_Client_LIMA		(...)	(...)	SKT_DATA_CLIENT	
SKT_DATA_Client_LIMA.Connect_Source		(...)	(...)	SKT_OpenConnectio...	
SKT_DATA_Client_LIMA.Connect_Source.DestAddr	192.168.100.17ports:32001	(...)	(...)	STRING	
SKT_DATA_Client_LIMA.Connect_Source.Timeout	2000	(...)	(...)	Decimal	DINT
SKT_DATA_Client_LIMA.Create_Source		(...)	(...)	Decimal	DINT
SKT_DATA_Client_LIMA.Instance	2941	(...)	(...)	Decimal	DINT
SKT_DATA_Client_LIMA.Read_Response		(...)	(...)	SKT_Read_Response	
SKT_DATA_Client_LIMA.Read_Source		(...)	(...)	SKT_Read_Parameters	
SKT_DATA_Client_LIMA.Timeout	2000	(...)	(...)	Decimal	DINT
SKT_DATA_Client_LIMA.Write_SizeSent	8	(...)	(...)	Decimal	DINT
SKT_DATA_Client_LIMA.Write_Source		(...)	(...)	SKT_Write_Parameters	
SKT_MSG_Client_Connect_LIMA		(...)	(...)	MESSAGE	
SKT_MSG_Client_Create_LIMA		(...)	(...)	MESSAGE	
SKT_MSG_Client_Delete_LIMA		(...)	(...)	MESSAGE	
SKT_MSG_Client_Read_LIMA		(...)	(...)	MESSAGE	
SKT_MSG_Client_Write_LIMA		(...)	(...)	MESSAGE	
SKT_MSG_Delete_All		(...)	(...)	MESSAGE	

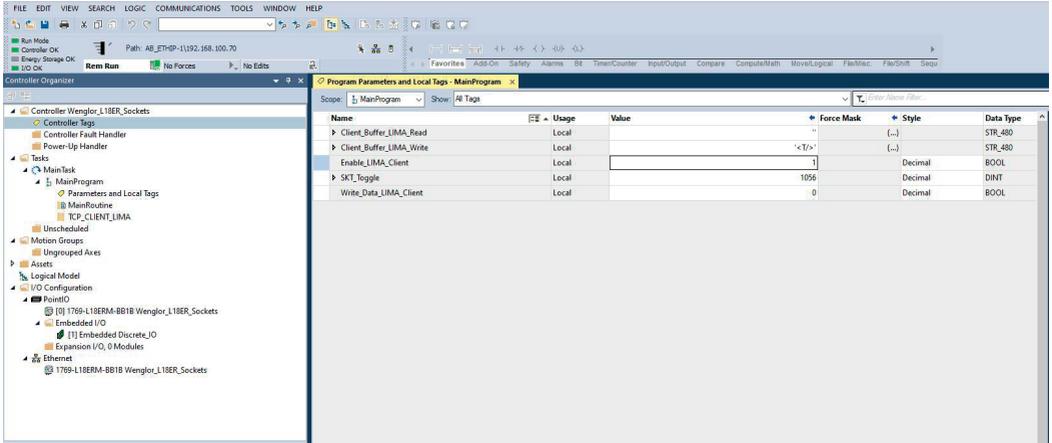
Transfer the sample program to the controller and go online.

NOTE!



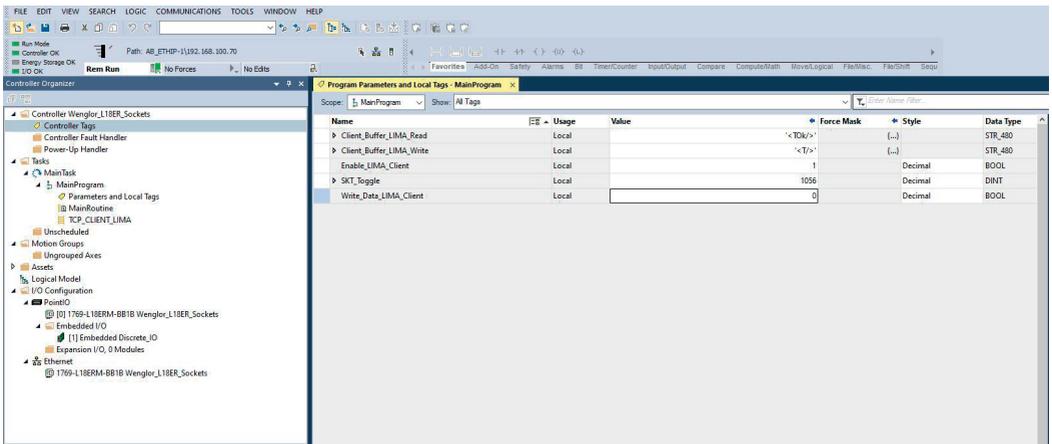
A connection from the controller to the uniVision product can be established only if port 32001 is available for the controller. Depending on the product or the mode of operation of the uniVision software, port 32001 may also be required by the uniVision software (e.g., in editing mode). In this case, the connection via the uniVision software may have to be disconnected so that the connection can be established via the controller.

The TCP connection is established by activating the value `Enable_LIMA_Client` under Parameters and Local Tags.



The LIMA command is sent to the uniVision product by activating `Write_Data_LIMA_Client`. The command may be sent once only and must not be permanently set so that only one image or profile is recorded. A new command must not be sent until the LIMA response to the previous command has been received.

The LIMA response is contained under `Client_Buffer_LIMA_Read`. For the trigger command, `<TOK/>` is sent by the uniVision product in response to a successful execution of the trigger command.



In addition, after data recording and evaluation, the new process data are also available via TCP (see the sample program `Example_Rockwell_ProcessData.ACD`). The run counter can be used, for example, to check if new results are available.

