MITSUBISHI ELECTRIC
Changes for the Better

IoT Gateway

IBHsoftec OPC UA Server function

Table of contents

1 Relevant manuals 4

2 Overview 5

2.1 System structure 5

2.2 Security 6

2.2.1 Encryption 6

2.2.2 Certificates 6

3 Commissioning 7

3.1 Software used 8

3.2 Prepare SD card 8

3.3 Prepare hardware 9

3.4 Set IP addresses for RD55UP12-V IBH 9

3.5 IoT Gateway OPC UA Server Setup 10

3.5.1 Launch web interface 10

3.5.2 Activate licence code 11

3.6 Backup / Restore of SD card 12

3.6.1 Create Backup 12

3.6.2 Restore the SD card 13

4 OPC UA server function 14

4.1 iQ-R CPU 14

4.1.1 Configure SLMP Connection 14

4.1.2 Export global variables 15

4.1.3 Add a controller via web interface 16

4.1.4 Insert CPU 16

4.1.5 Import XML file 17

4.1.6 Connect external OPC UA Client 19

4.2 Robot 22

4.2.1 Communication settings RT Toolbox3 22

4.2.2 Insert robot via web interface 23

4.2.3 Display in the external OPC UA Client 24

4.2.4 Add additional outputs 25

4.3 Inverter 27

4.3.1 A800/F800 30

4.3.2 E800-E 34

5 Diagnostics 37
5.1 Controller diagnostics 37
5.2 Client diagnostics 37
5.3 Network diagnostics 38
5.4 System log 40
6 OPC Client Function Example 43
7 MODBUS 45
7.1 Add MODBUS Configuration 45
7.2 Add Variables 46
7.3 Usage of MODBUS-Data 48
8 MQTT Settings 50
8.1 Create new Broker connection 50
8.2 Publish of values (Send data to Broker) 51
8.2.1 Create Topic 51
8.2.2 Create new variable (data to be sent) 53
8.2.3 Check data in Broker 54
8.3 Subscription to values (receive data from broker) 54
8.3.1 Create Topic 54
8.3.2 Create new variable 55
8.3.3 Function test of “Subscribe” 56
9 Teamviewer IoT Connection 57
9.1 PC preparations - connected to the Ethernet port of the control level 58
9.1.1 TeamViewer Software 58
9.1.2 IBHNet IoT setup 59
9.1.3 TeamViewer IoT Management Console 59
9.1.4 Connection setup 63
9.2 Access to controllers (CPUs / devices) that are connected to the ports of the control level. 66
9.2.1 iQ-R CPU 66
9.2.2 Robot 67
9.2.3 GOT 69

1 Relevant manuals

This "Quick Start Manual" only describes the necessary steps to put the hardware into operation and to parameterise the corresponding functions.

Further details can be found in the relevant manuals.

Manual name [manual number]	Description
MELSEC iQ-R C intelligent function module Linux Startup manual (Copyright© 2008-2020 Lineo Solutions, Inc. All rights reserved)	
MELSEC iQ-R C Intelligent Function Module User's Manual (Startup) [SH-081566ENG]	Specifications, procedure before operation, wiring, and operation examples of a C intelligent function module
MELSEC iQ-R C Intelligent Function Module User's Manual (Application) [SH-081567ENG]	Functions, input/output signals, buffer memory, parameter setting, and troubleshooting of a C intelligent function module
MELSEC iQ-R Module Configuration Manual [SH-081222ENG]	The combination of the MELSEC iQ-R series modules, common information on the installation/wiring in the system, and specifications of the power supply module, base unit, SD memory card, and battery
MELSEC iQ-R C Intelligent Function Module Programming Manual [SH-081568ENG]	Programming specifications and dedicated function libraries of a C intelligent function module
MELSEC iQ-R C Controller Module/C Intelligent Function Module Programming Manual (Data Analysis) [SH-081756ENG]	Programming specifications and dedicated function libraries for analysing the data of a C controller module and a C intelligent function module
CW Workbench/CW-Sim Operating Manual [SH-081373ENG]	System configuration, specifications, functions, and troubleshooting of CW Workbench/CW-Sime-Manual
GX Works3 Operating Manual [SH-081215ENG]	System configurations, parameter settings, and operation methods for the online function in GX Works3
GT Designer3 (GOT2000) Screen Design Manual [SH-081220ENG]	

2 Overview

Under the catchword "Industry 4.0", more and more new communication and information technologies are finding their way into industrial automation. In order to be able to implement these requirements efficiently and as quickly as possible, attempts must be made to reduce the high level of complexity through modularisation and standardisation.

In this context, OPC UA has proven to be a forward-looking communication standard. This is because OPC UA fulfils the Industry 4.0 requirement for independence from manufacturer, industry and operating system communication.

Since OPC UA not only transmits machine data, i.e. process values and measured values, but can also describe them semantically, OPC UA is becoming increasingly important when it comes to transferring machine data from controllers to higher-level systems.

One solution for reading process data from Mitsubishi Electric controllers and robots that do not have an OPC interface as standard is the RD55UP12-V module with installed IBH Link UA software, called IoT Gateway.

This solution is a server/client module. The OPC client function also enables OPC servers to exchange data with each other. For example, controllers or other devices with OPC UA from different manufacturers can exchange data via this function.

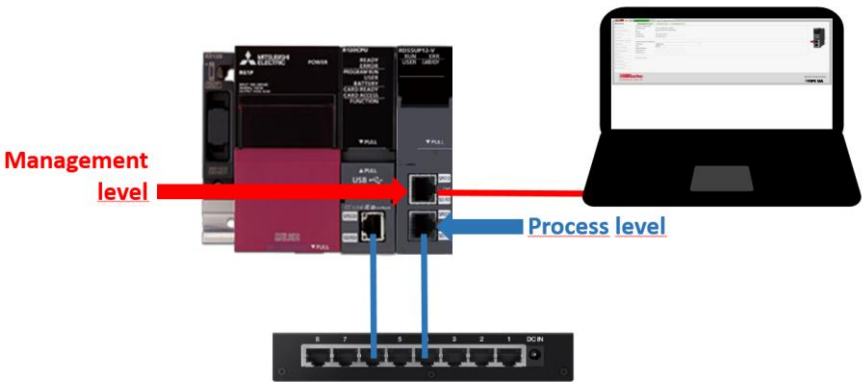
Communication with the controller is via TCP/IP.

The IoT Gateway has an Ethernet port for the machine level and an Ethernet port for the control level.

Only OPC accesses are possible on the control level. A firewall is integrated between the two levels. The ports are configured via the GX-Works3 programming software of the iQ-R CPU.

2.1 System structure

As a minimum configuration, the IoT Gateway always requires a base rack, an iQ-R CPU and the RD55UP12-V module. To establish a connection to several controllers, a switch is required at the process level.



2.2 Security

The IoT Gateway has a certificate management for secure communication. A corresponding software interface is supplied for this purpose, which can be displayed in any web browser. The configuration of the security levels and the administration of the certificates are carried out in this interface. This is based on the mechanisms defined by the OPC Foundation. OPC UA Security includes authentication and authorisation, encryption and data integrity through signing. In this way, the control system can be protected against uncontrolled access via a higher-level system.

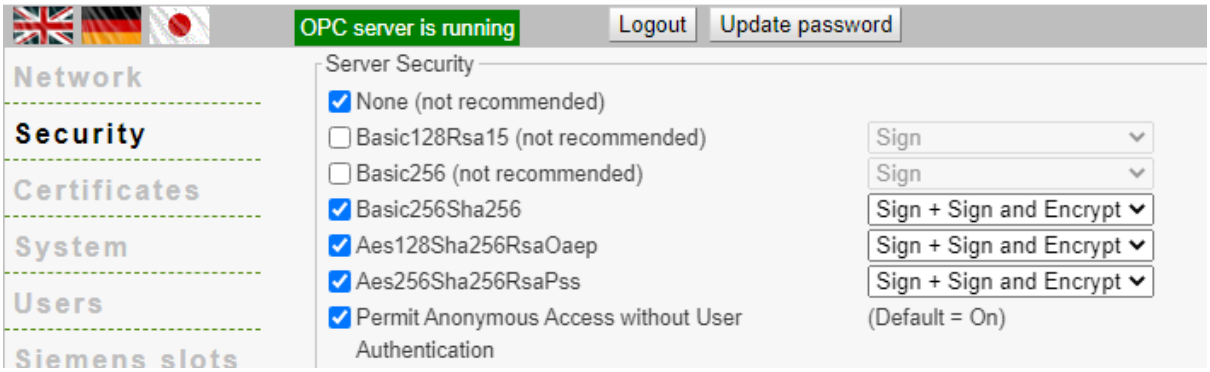
2.2.1 Encryption

There are different levels of encryption (Encrypt): None, Basic128Rsa15 and Basic256.

None	No encryption
Basic128Rsa15	128 bit encryption
Basic256	256 bit encryption

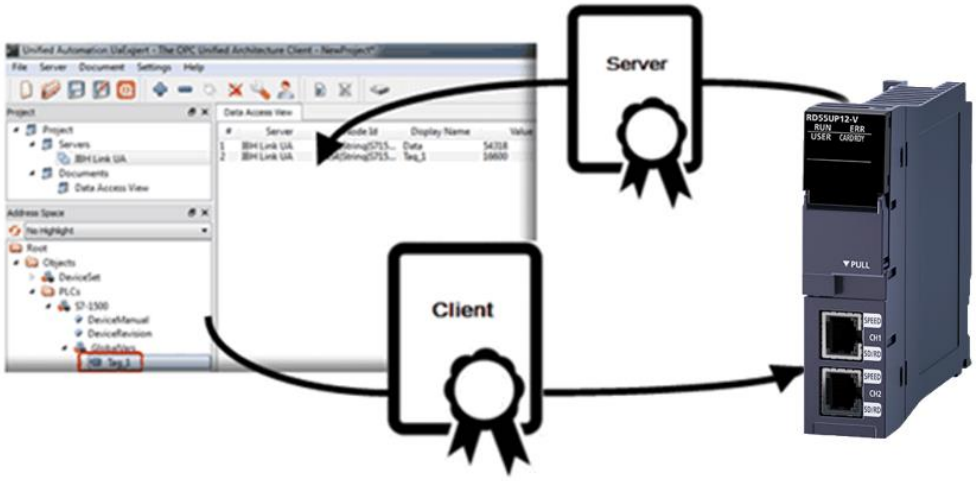
The IBH Link UA function supports the Level **Sign** and **SignAndEncrypt**

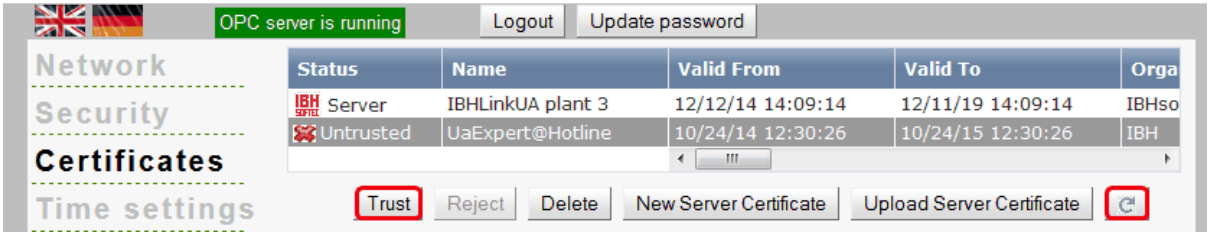
None	No security level, fastest data transmission
Sign	The messages contain signed certificates
Sign and Encrypt	The messages contain signed certificates and are encrypted
Sign + Sign and Encrypt	The messages may contain only signed certificates, or can be signed and are encrypted as well



2.2.2 Certificates

Another security level of OPC UA is the exchange of certificates. Communication is only possible when both server and client have been assigned a valid certificate by the respective partner.



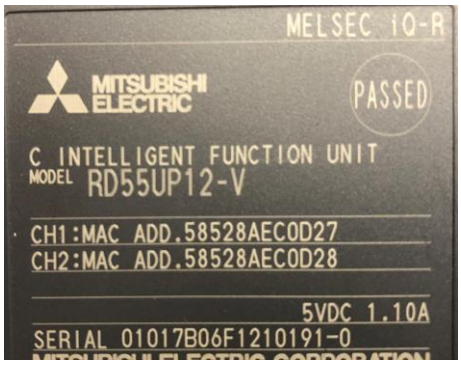


3 Commissioning

Each RD55UP12-V can be started as an IoT Gateway module. For this purpose, a corresponding image must be written to an SD card.

The image file contains the Linux operating system and the corresponding programme from IBH for the OPC UA server function.

In order to be able to use the OPC UA server function without restriction, it must be activated for the hardware used with a licence code created at IBH. For this purpose, the serial number and MAC addresses from the type plate of the module are required.



3.1 Software used

Product	Description
DDWIN (Download link: http://download.si-linux.co.jp/dd_for_windows/DDWin_Ver0998.zip)	Program for writing the IBHsoftec image file to an SD card
UaExpert (Download link: https://www.unified-automation.com/de/downloads/opc-ua-clients.html)	Free OPC UA client program for testing the server function
MELSOFT GX Works3 EU version	Programming tool for programming the iQ-R CPU
GT Designer3	Programming tool for HMIs of the GOT series
RT Toolbox3	Programming tool for robots
IBH OPCUA Edit https://download.ibhsoftec.com/neutral/IBHOPCUAEditor749Setup.exe	Offline Configuration tool for IoT Gateway

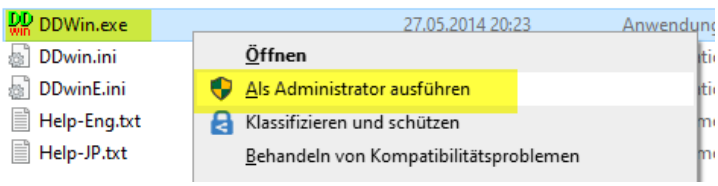
In addition, a web browser must be installed on the PC.

If you have received an already configured IoT Gateway from Mitsubishi Electric, the following setup chapters 3.2 to 3.5 are not required.

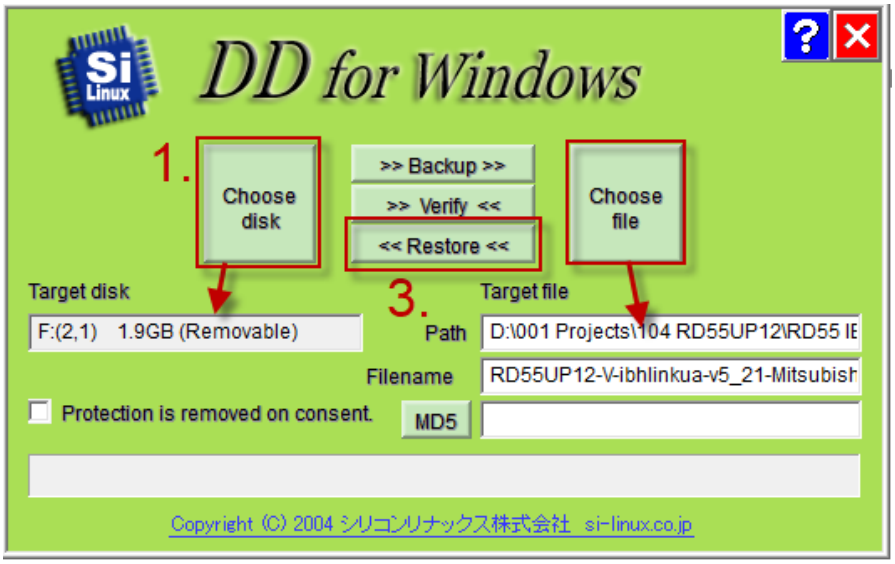
For your security, we recommend that you create a backup copy of the inserted SD card, as described in Chapter 3.6.

3.2 Prepare SD card

To write the image file with the Linux OS to an SD card, you need the programme "DDWIN" on a Windows computer. After downloading, the ZIP file must be unpacked, an SD card inserted into the PC and the file "DDWIN.EXE" executed as administrator.



1. Select the SD card in the PC under "Choose disk".
2. Select the Linux image file from IBHsoftec under "Choose file".
3. Select "<<Restore>>".



When writing to the SD card is finished, the "DDWIN" program can be closed and the SD card removed from the PC and inserted into the RD55UP12-V module.

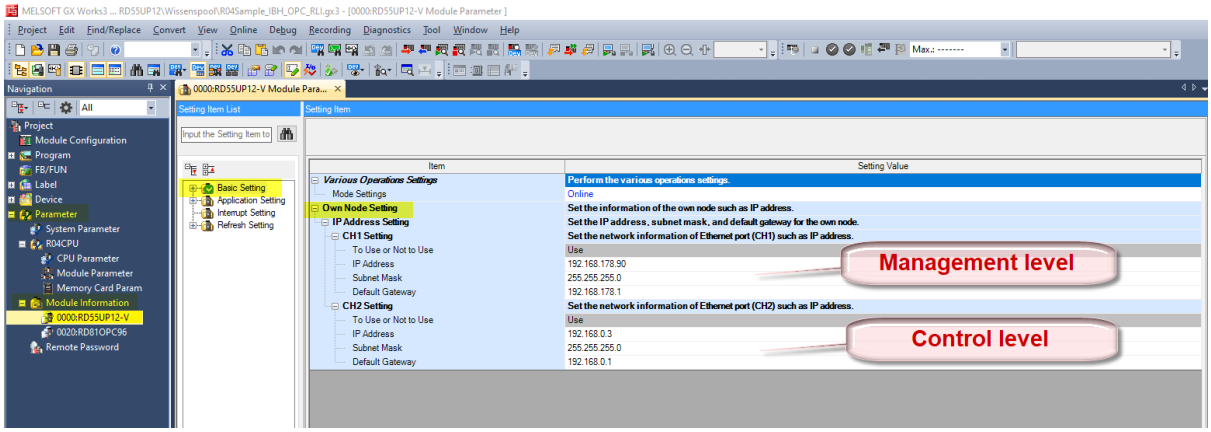
3.3 Prepare hardware

Unpack the selected iQ-R hardware, mount it to the base rack and connect it as instructed (refer to Mitsubishi Electric hardware manuals). The control unit can now be switched on.

3.4 Set IP addresses for RD55UP12-V IBH

First, you have to create a project in GW-Works3 that matches the hardware you have plugged in.

Under "Parameters" →"Module Information" →RD55UP12-V "→Basic Setting" →"Own node setting" the IP addresses for the control and process level can then be set.



Once all settings have been made, the new project is transferred to the CPU. The control unit must be switched off and on again once after the transfer.

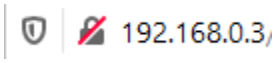
The channel "CH1" of the RD55 module is now in the address range of the management level and the channel "CH2" is in the range of the control level.

3.5 IoT Gateway OPC UA Server Setup

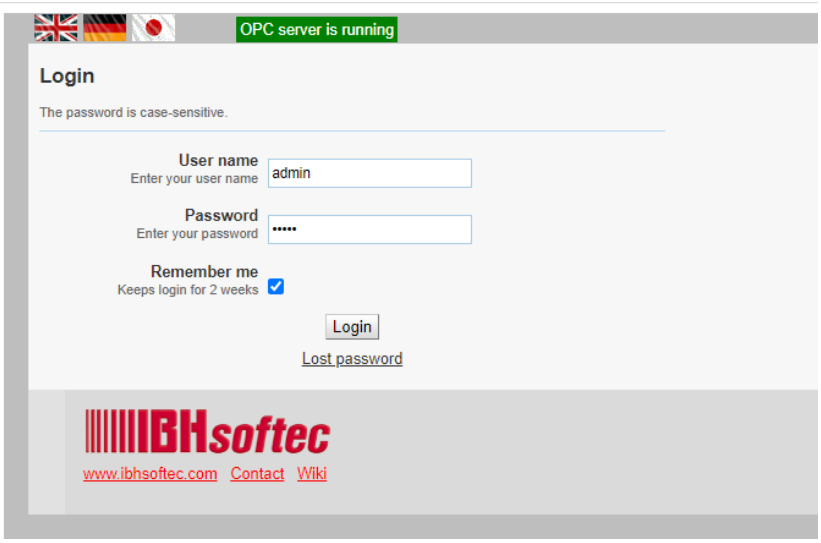
3.5.1 Launch web interface

Depending on the level via which the PC is connected to the IoT Gateway, the parameterisation interface can be called up by entering the corresponding IP address in a web browser.

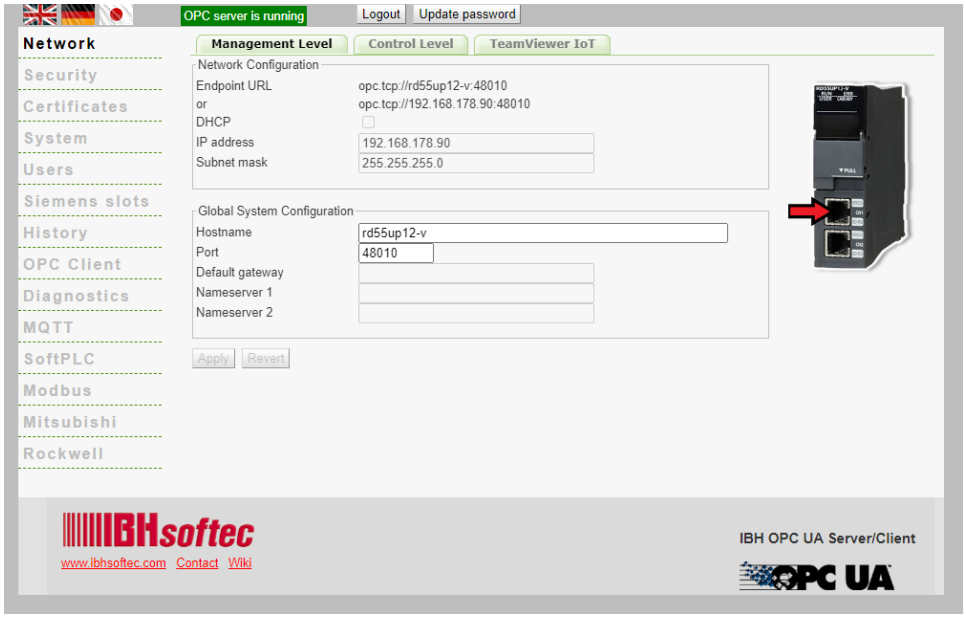
In the following example, the PC is connected via the process level, only "192.168.0.3" is entered in the browser.



The login screen is displayed. For a new system, the default user name and password are both "admin".



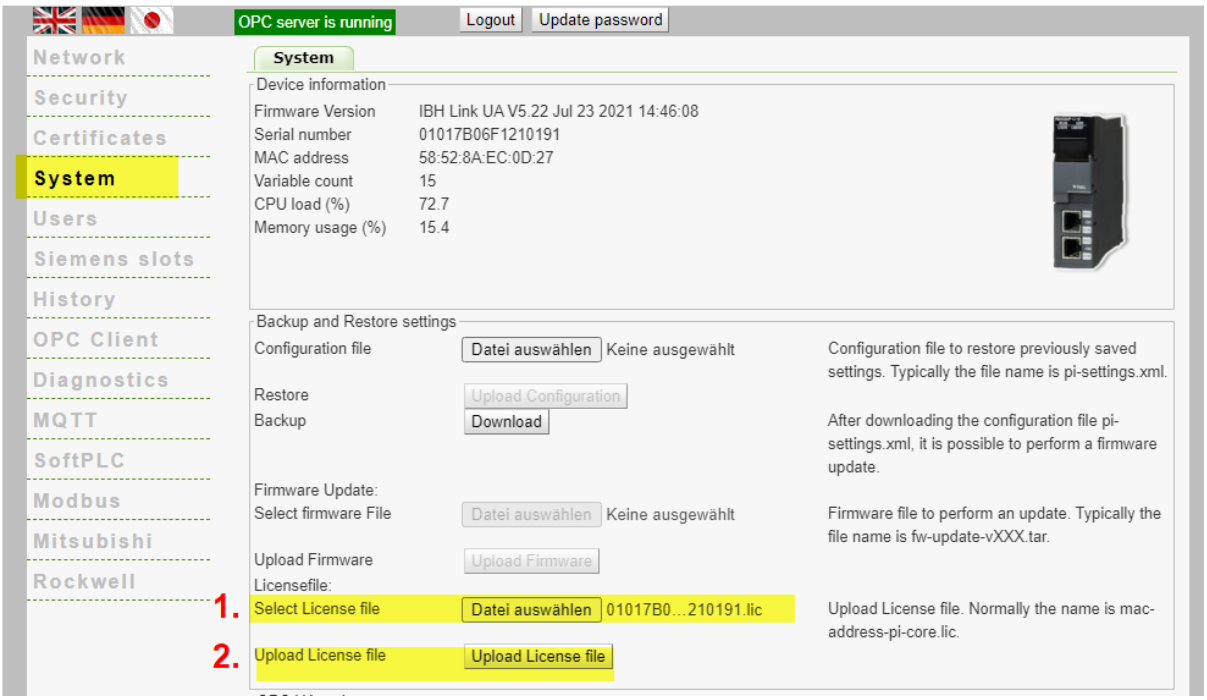
After pressing "Log in", you are in the configuration interface.



3.5.2 Activate licence code

After you have received the license file from IBHsoftec matching the module serial number and MAC addresses, this must be imported into the module once.

To do this, select the "System" menu in the configuration interface.



On this page, click on "Browse" at "Select licence file" to search for the corresponding file and then click on "Upload licence file" to copy it to the SD card.

After a restart of the system, the IoT Gateway can be used without restrictions.

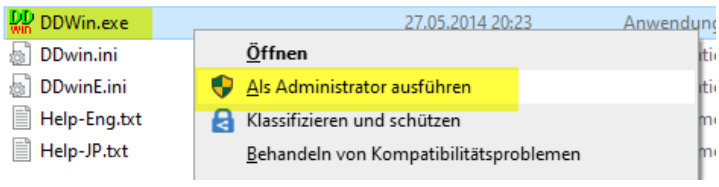
The basic set-up of the IoT Gateway is now complete. The individual functions are explained in the following chapters.

3.6 Backup / Restore of SD card

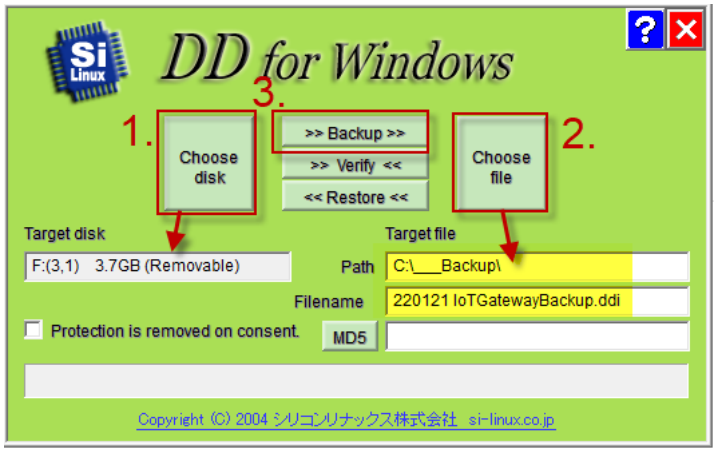
After all the necessary configurations have been made and the project functions have also been parameterized, a backup copy can be created from the SD card using the "DDWIN" software. This backup can only be used in the RD55UP12-V module with the hardware ID for which the IoT Gateway license was created.

3.6.1 Create Backup

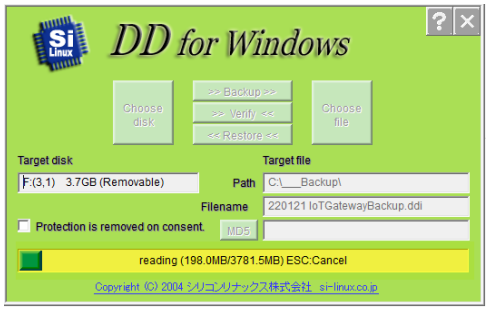
To create the backup, switch off the controller with the IoT Gateway and remove the SD card from the IoT Gateway module. Insert the SD card into the PC and run the "DDWIN.EXE" program as administrator.

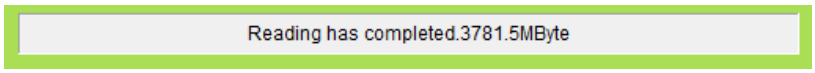


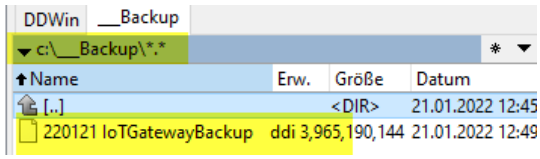
1. Select the SD card in the PC under "Choose disk"
2. Select a drive path under "Choose file" and enter the desired file name
3. Select „>>Backup>>“



The data will now be read from the SD card





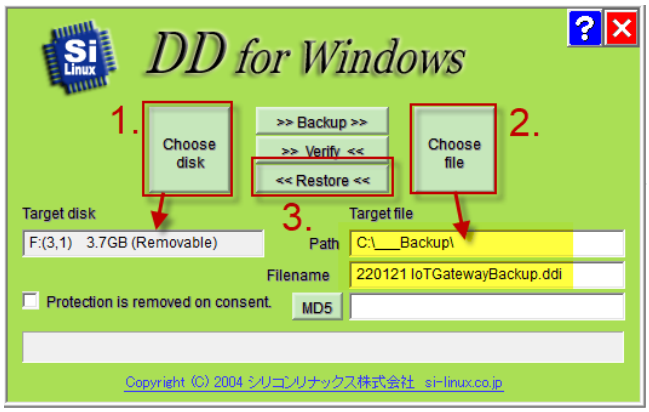


The SD card can now be reinserted into the IoT Gateway and the controller can be started.

3.6.2 Restore the SD card

To load the image onto a new SD card, the "DDWIN.EXE" program must be run as an administrator

Then insert a new SD card into the PC



1. Select the SD card in the PC under "Choose disk"
2. Select the previously created backup copy under "Choose file".
3. Select „<<Restore<<“

After the message "Writing has completed ..." the SD card can be plugged into the corresponding RD55UP12-V module and then the controller can be switched on.

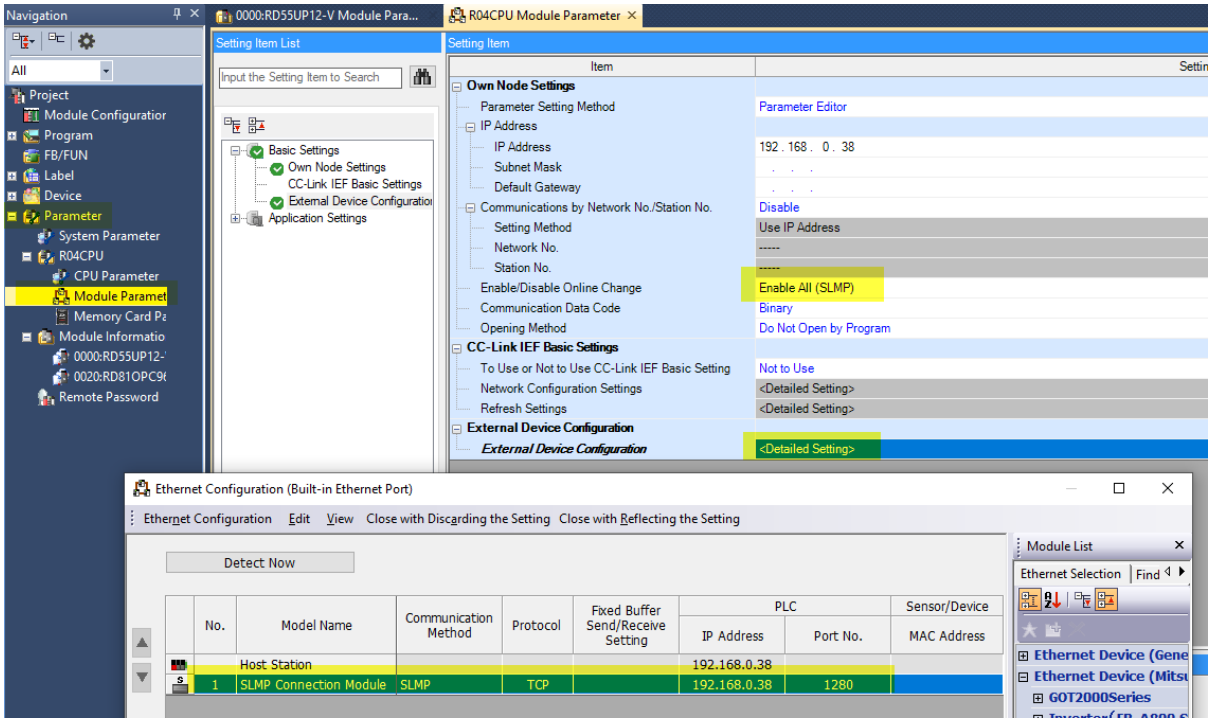
4 OPC UA server function

This chapter explains how to connect the IoT Gateway to controllers and make their global variables available as OPC UA tags for clients.

4.1 iQ-R CPU

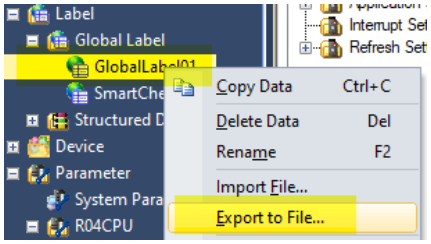
4.1.1 Configure SLMP Connection

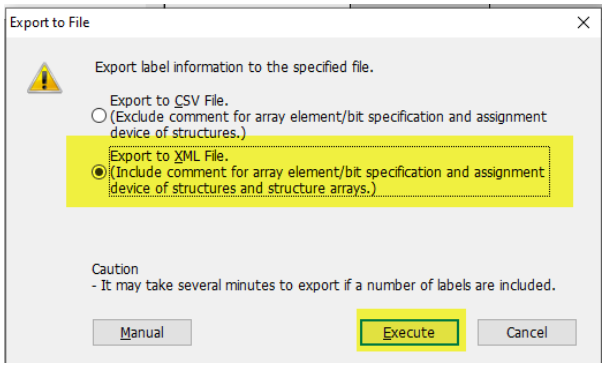
In order for the Mitsubishi controller to be accessible via OPC UA, the **SLMP** Connection Module must be inserted using the GX Works configuration software.

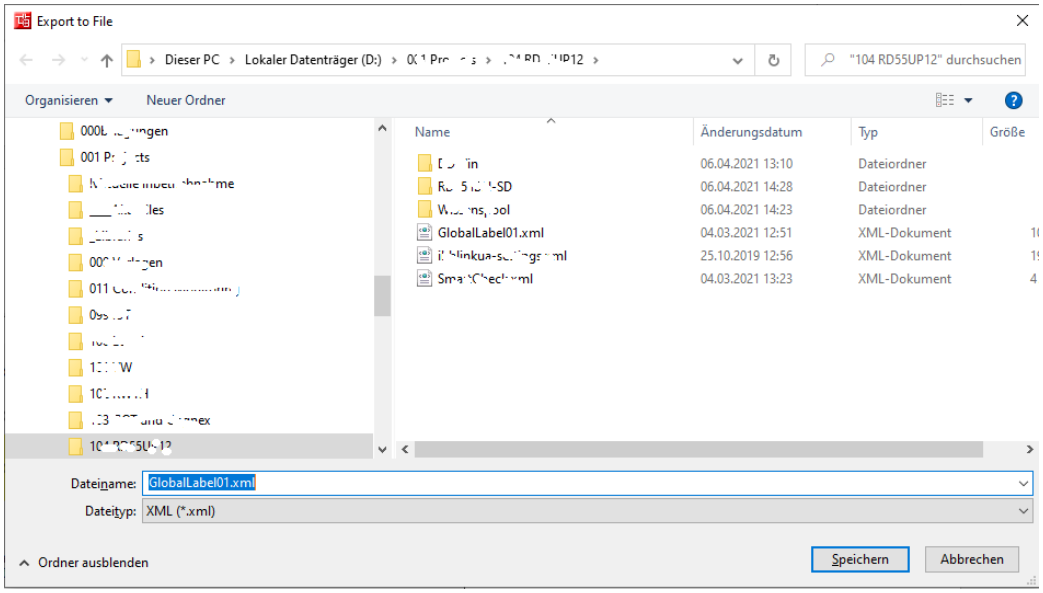


Important: In order for data to be written externally to the OPC UA tags, "Enable all (SLMP)" must be selected!

4.1.2 Export global variables

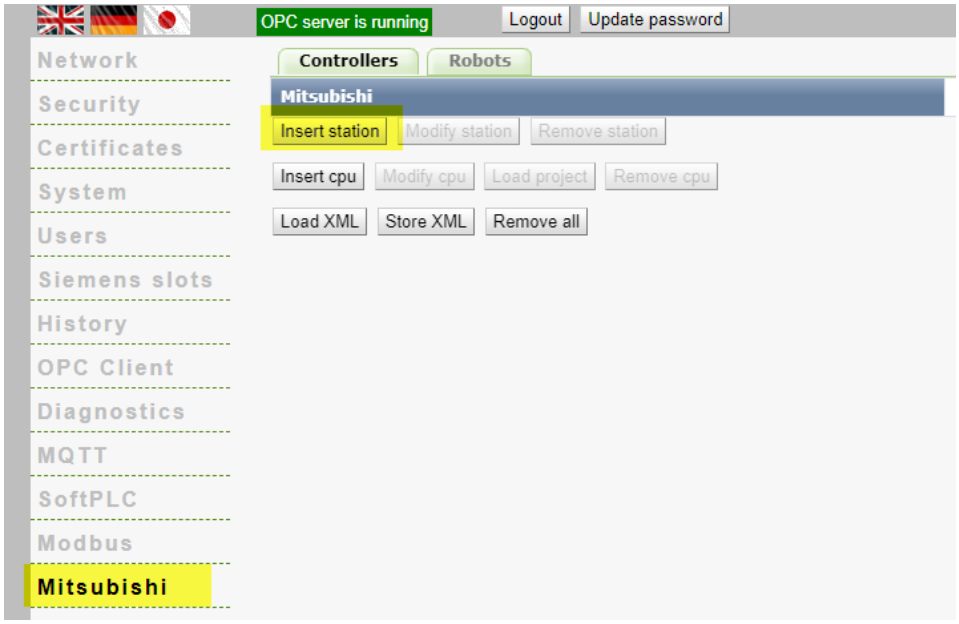


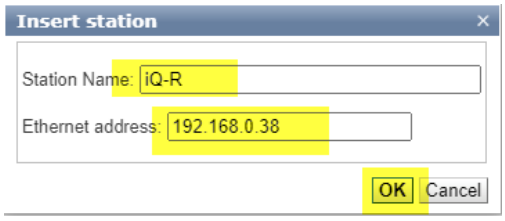




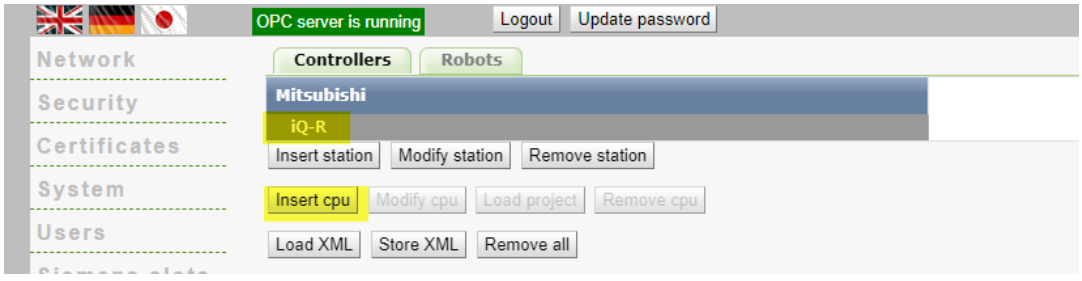
4.1.3 Add a controller via web interface

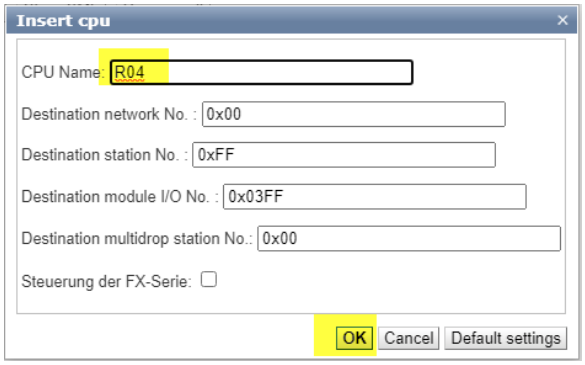
Insert station





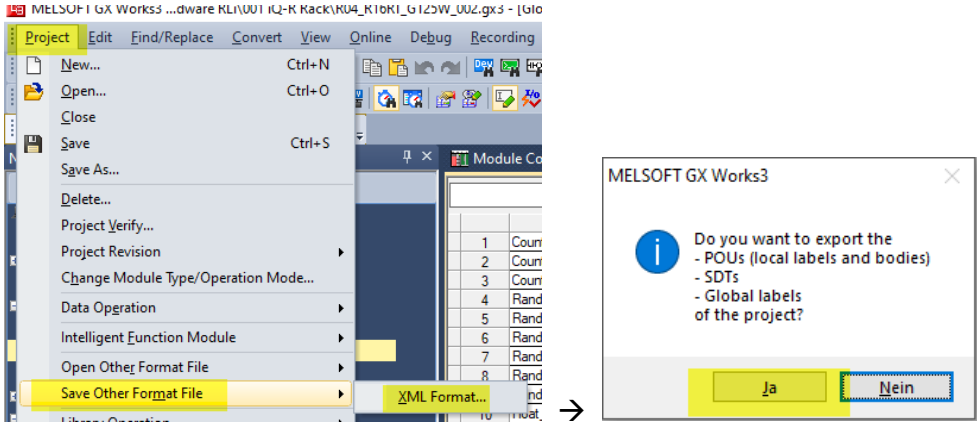
4.1.4 Insert CPU



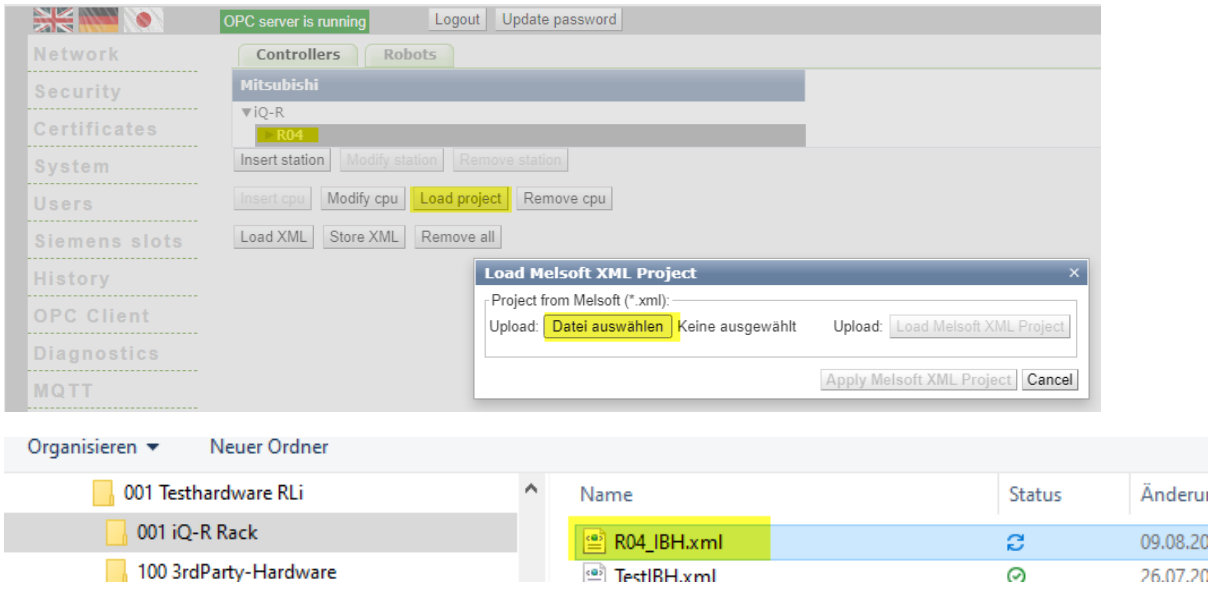


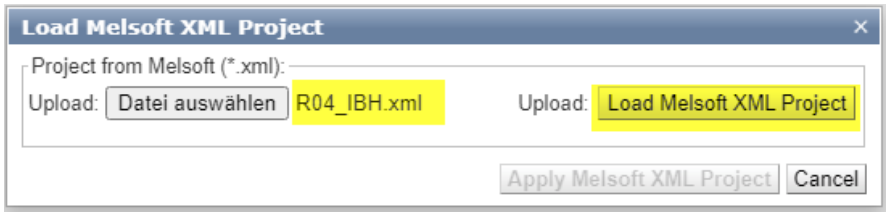
4.1.5 Import XML file

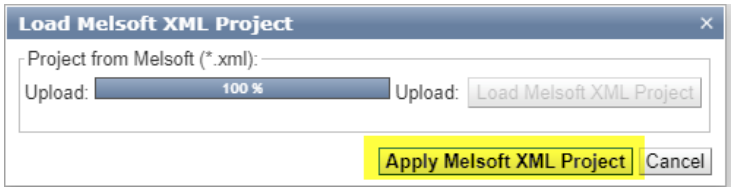
The European version of GX Works 3 offers the function to export the project as an XML file.

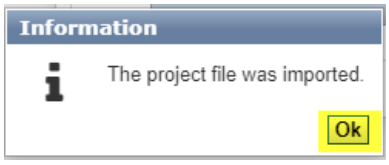


The file created by this function can be uploaded to the IoT Gateway web server.

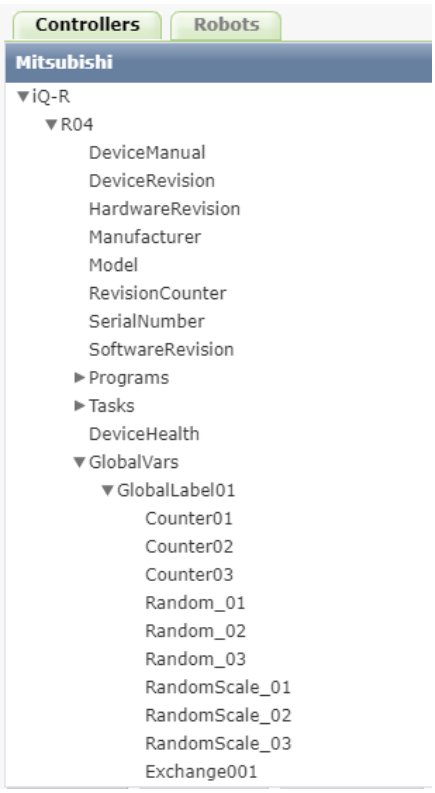








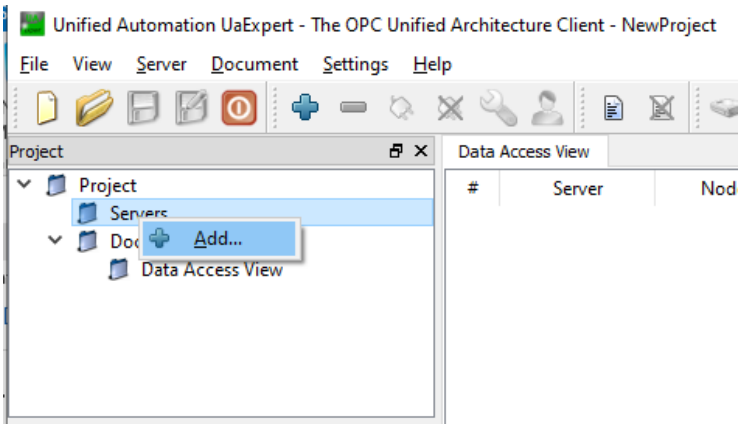
Now all global variables of the imported project are automatically available in the OPC UA Server.

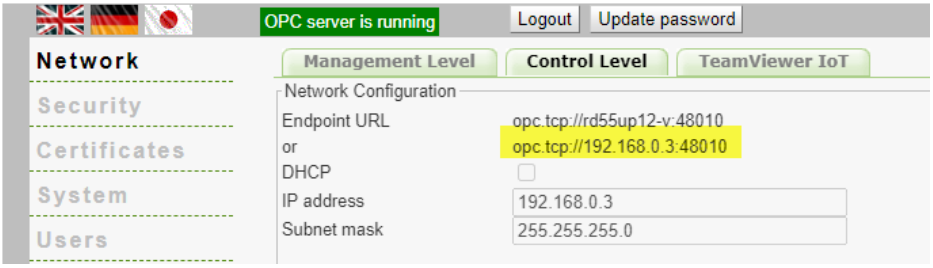


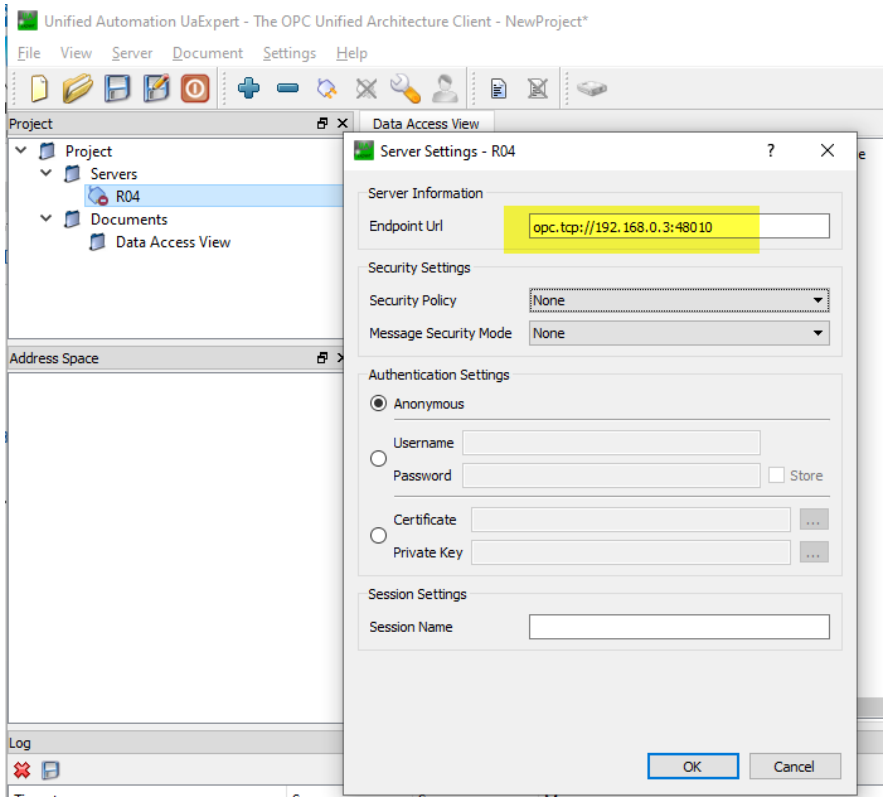
4.1.6 Connect external OPC UA Client

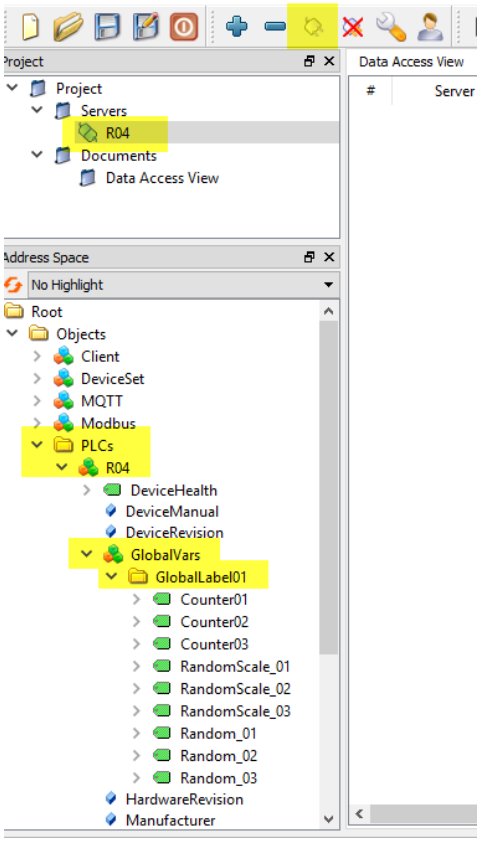
The free UaExpert is used here as a test client.

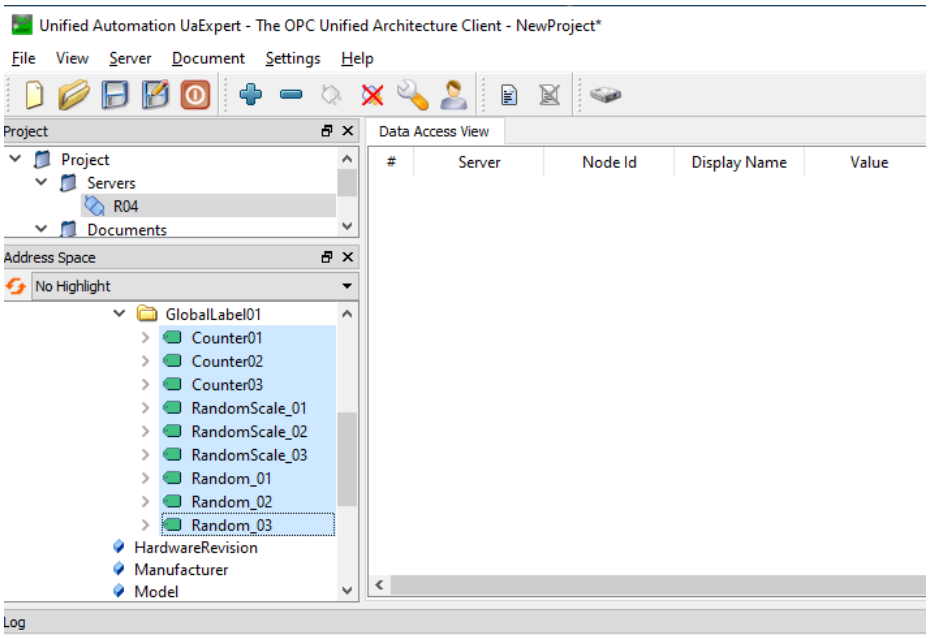
https://www.unified-automation.com/de/downloads/opc-ua-clients.html

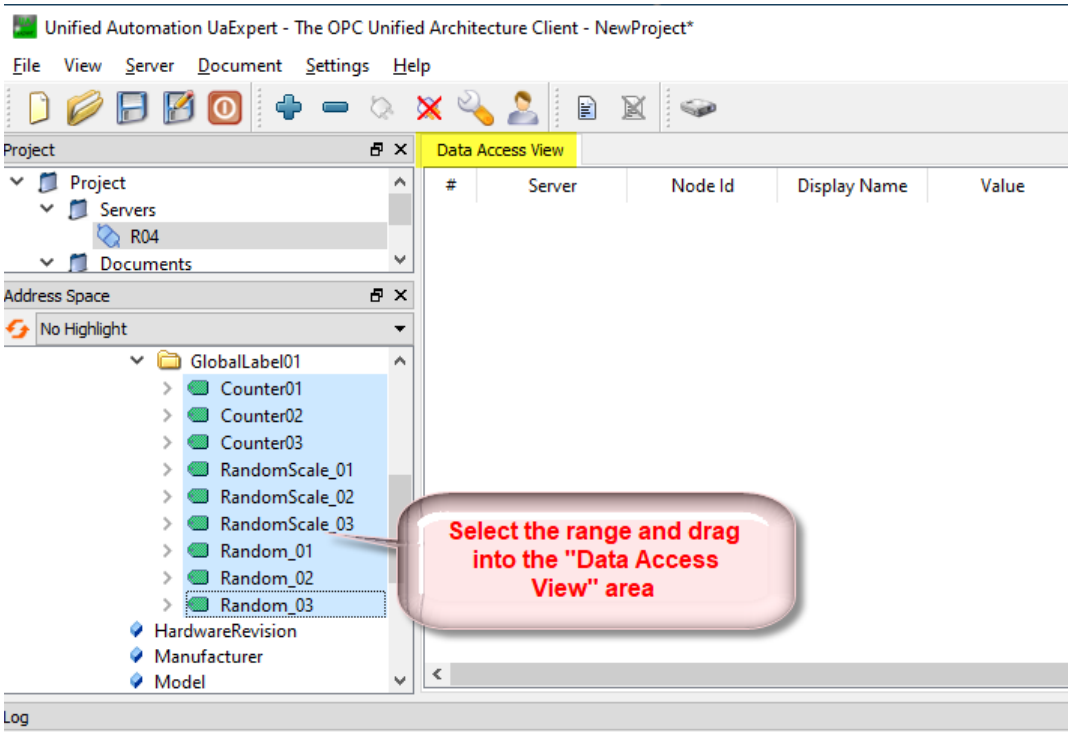


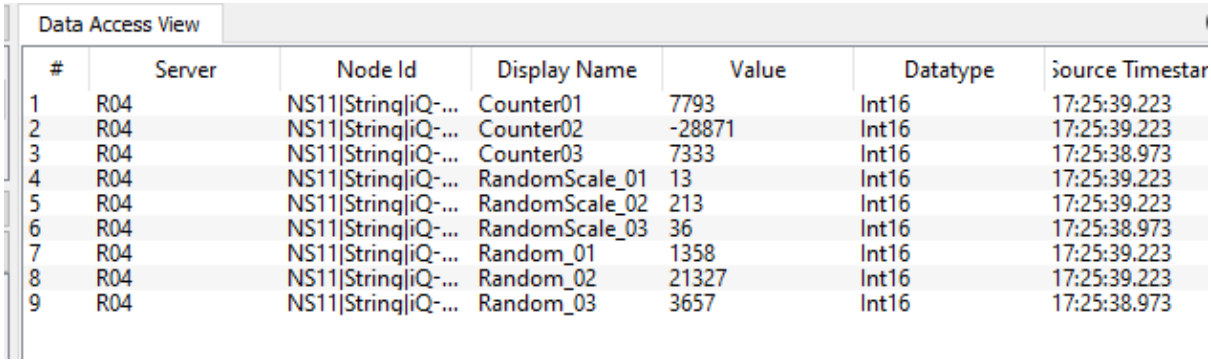





Project
Servers
R04
Documents
Data Access View
Address Space
No Highlight
Root
Objects
Client
DeviceSet
MQTT
Modbus
PLCs
R04
DeviceHealth
DeviceManual
DeviceRevision
GlobalVars
GlobalLabel01
Counter01
Counter02
Counter03
RandomScale_01
RandomScale_02
RandomScale_03
Random_01
Random_02
Random_03
HardwareRevision
Manufacturer
Data Access View
#
Server


Unified Automation UaExpert - The OPC Unified Architecture Client - NewProject*
File
View
Server
Document
Settings
Help
Project
Servers
R04
Documents
Address Space
No Highlight
GlobalLabel01
Counter01
Counter02
Counter03
RandomScale_01
RandomScale_02
RandomScale_03
Random_01
Random_02
Random_03
HardwareRevision
Manufacturer
Model
Data Access View
#
Server
Node Id
Display Name
Value
Log





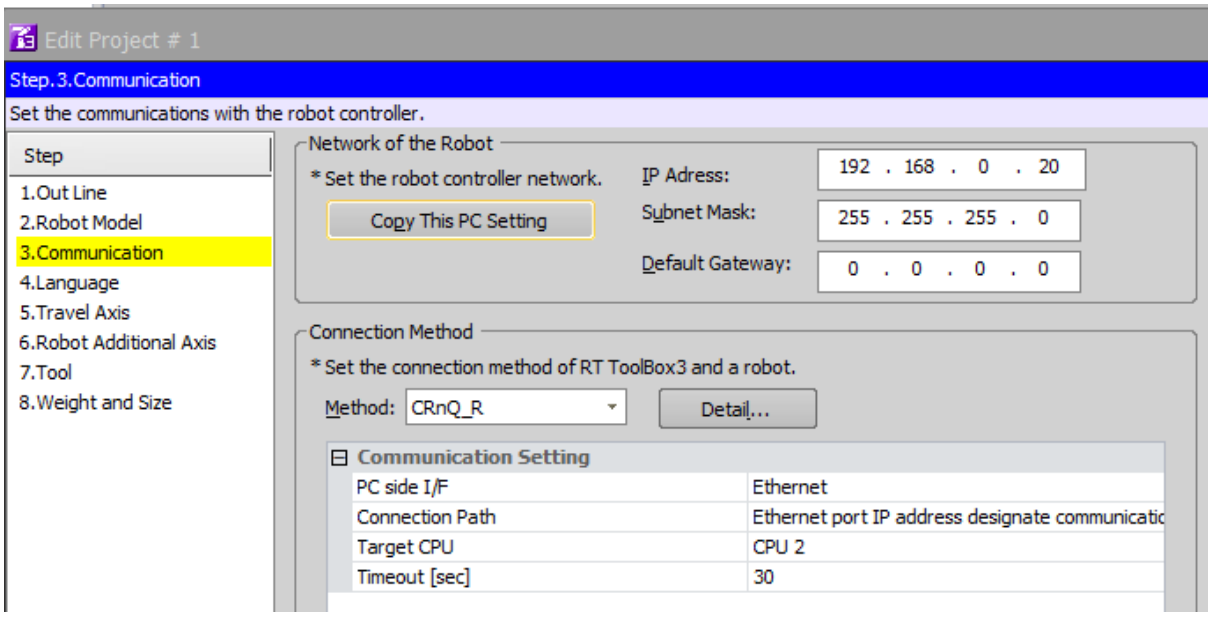
Data Access View

#	Server	Node Id	Display Name	Value	Datatype	Source Timestar
1	R04	NS11|String|iQ-...	Counter01	7793	Int16	17:25:39.223
2	R04	NS11|String|iQ-...	Counter02	-28871	Int16	17:25:39.223
3	R04	NS11|String|iQ-...	Counter03	7333	Int16	17:25:38.973
4	R04	NS11|String|iQ-...	RandomScale_01	13	Int16	17:25:39.223
5	R04	NS11|String|iQ-...	RandomScale_02	213	Int16	17:25:39.223
6	R04	NS11|String|iQ-...	RandomScale_03	36	Int16	17:25:38.973
7	R04	NS11|String|iQ-...	Random_01	1358	Int16	17:25:39.223
8	R04	NS11|String|iQ-...	Random_02	21327	Int16	17:25:39.223
9	R04	NS11|String|iQ-...	Random_03	3657	Int16	17:25:38.973

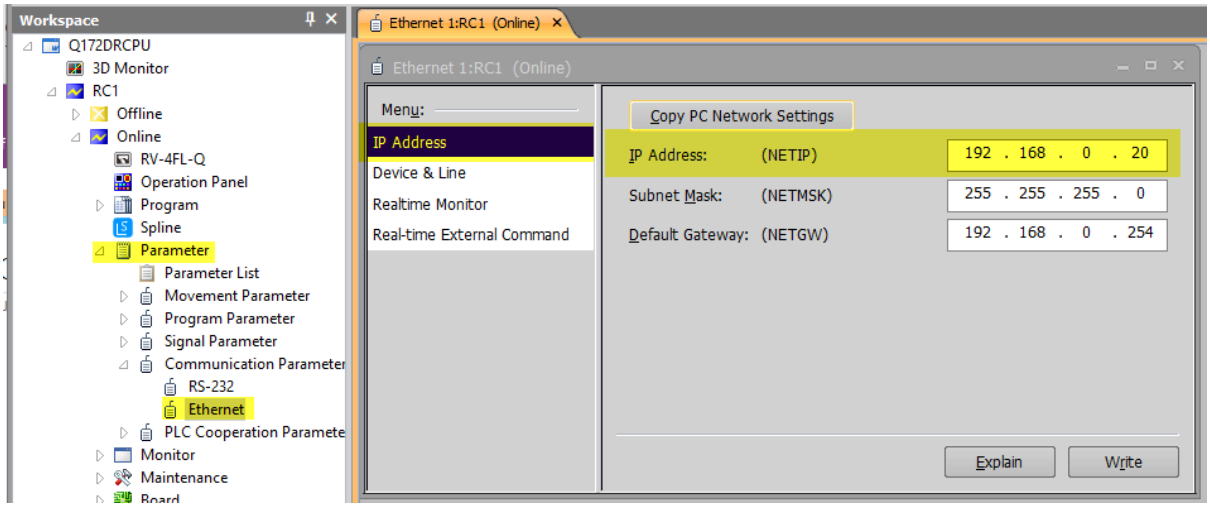
4.2 Robot

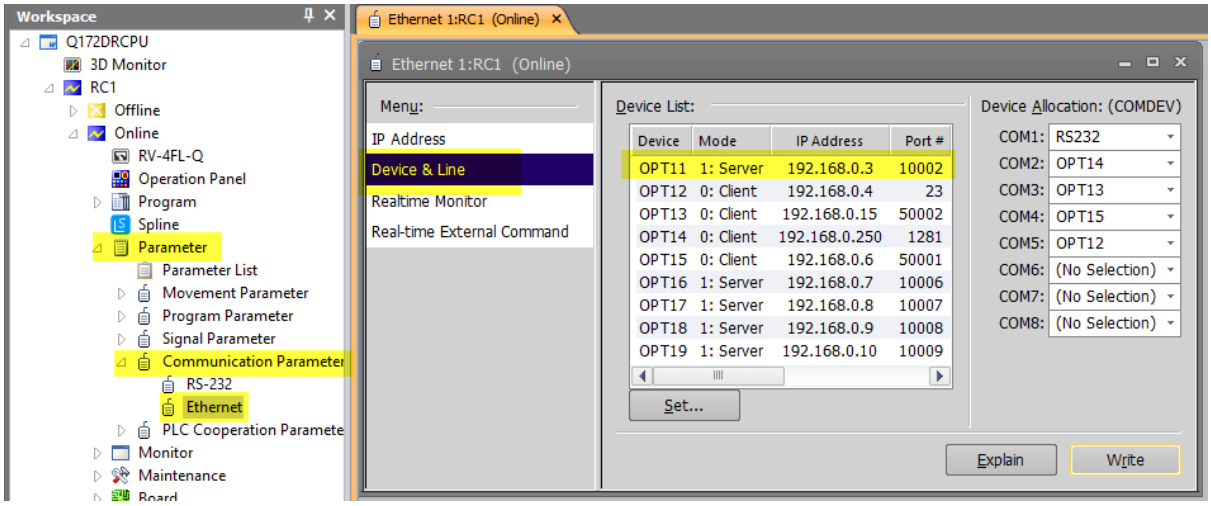
4.2.1 Communication settings RT Toolbox3

Project setting

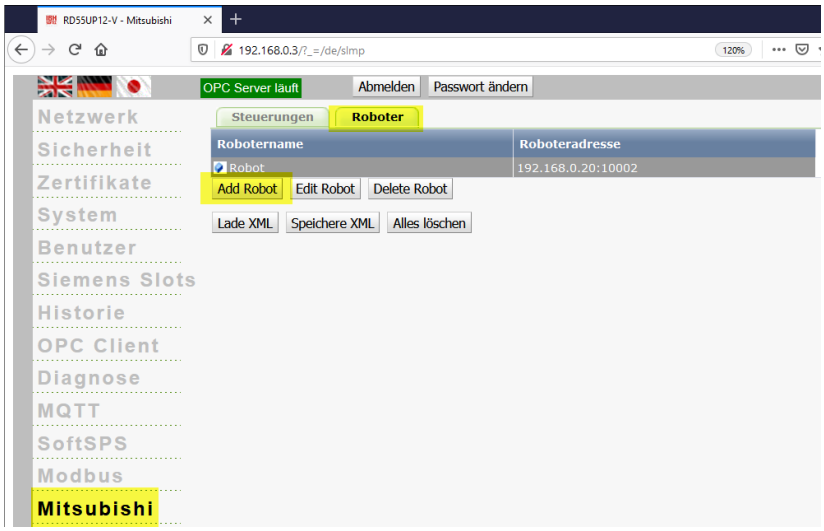


Parameter setting

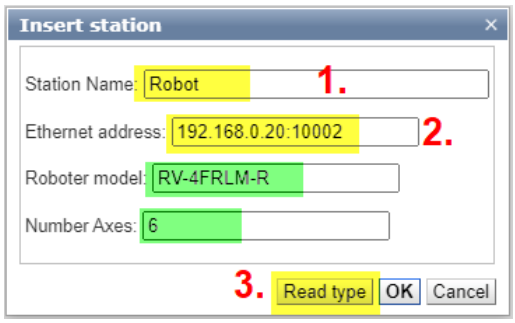




4.2.2 Insert robot via web interface

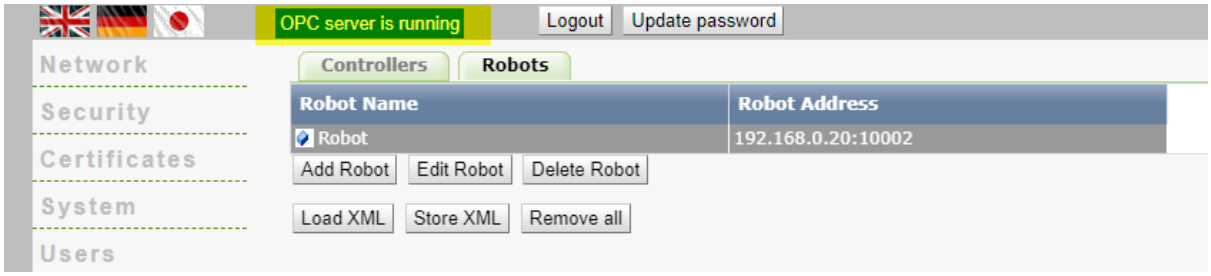


The station name (freely selectable) and the IP address of the robot with port number must be entered manually. If the Ethernet connection is OK, the robot model and the number of axes are entered automatically by pressing "Read type".



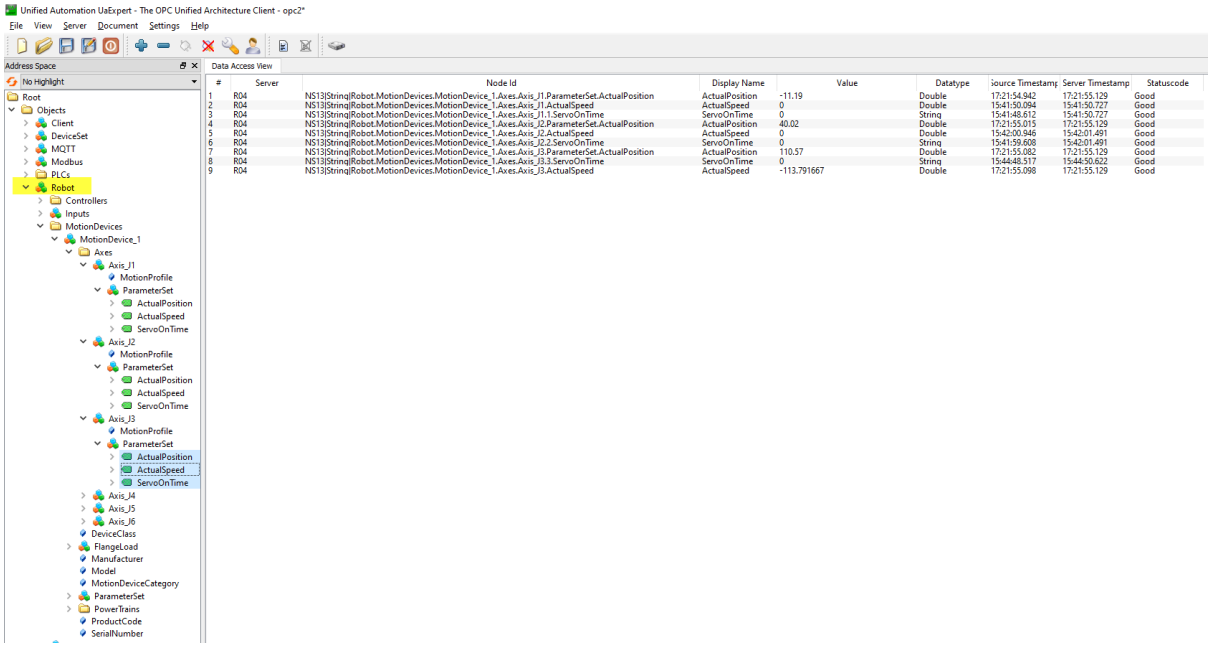
Then press "OK".

After a few seconds, the web interface will again show "OPC server running" in a green box.



4.2.3 Display in the external OPC UA Client

As soon as the server is running again, it can be selected in the Client Tool (here UaExpert) and you have access to all OPC UA tags from the robot that are defined in the OPC UA Companion Specification Part1.



4.2.4 Add additional outputs

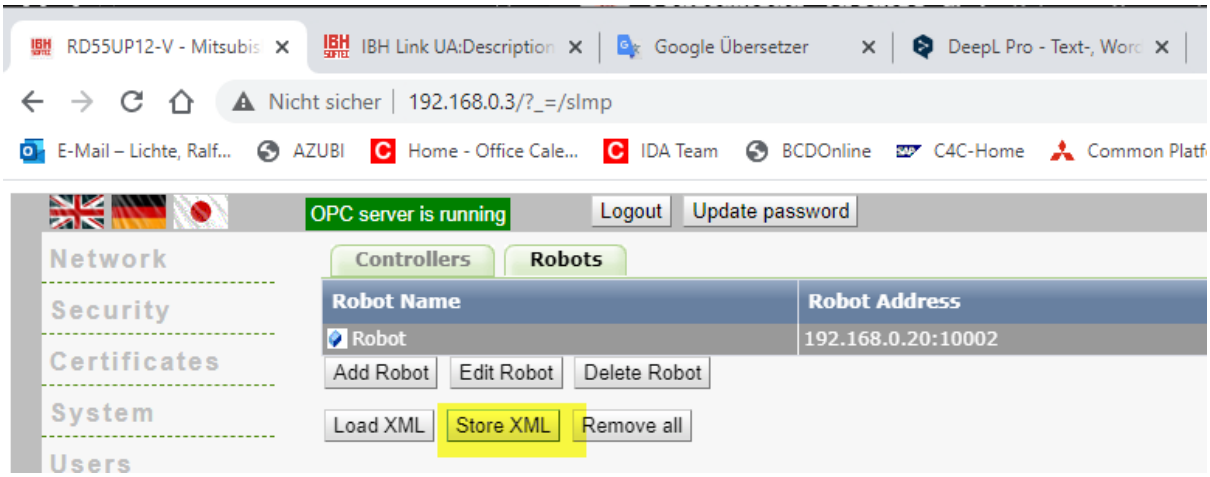
In the IoT Gateway OPC UA Server, the "Robot Companion Specification Part 1" is implemented first. In addition, the robot inputs and outputs 0 to 63 (M_In(0)/M_Out(0) to M_In(63)/M_Out(63)) can be accessed by default.

However, it is possible to add further inputs and outputs.

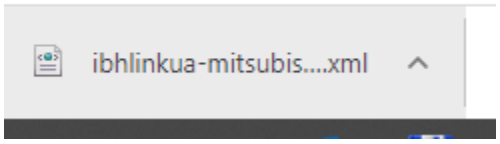
The following steps are necessary for this:

- Export XML file
- Add the required inputs or outputs to the XML file using a text editor.
- Load the XML file back into the RD55-IBH module (the server is then automatically restarted).

Export XML file



Depending on the browser, the file is then automatically saved in the "Download" folder or you are asked for a destination folder.



Edit XML file

Open the downloaded file with a text editor and scroll it to the desired "Inputs" or "Outputs" section.

```
<Outputs>
  <Output name="Outputs_0_15" address="0" type="UInt16"/>
  <Output name="Outputs_16_31" address="16" type="UInt16"/>
  <Output name="Outputs_32_47" address="32" type="UInt16"/>
  <Output name="Outputs_48_63" address="48" type="UInt16"/>
  <Output name="Output_0" address="0" type="Boolean"/>
  <Output name="Output_1" address="1" type="Boolean"/>
  <Output name="Output_2" address="2" type="Boolean"/>
  <Output name="Output_3" address="3" type="Boolean"/>
  <Output name="Output_4" address="4" type="Boolean"/>
  <Output name="Output_5" address="5" type="Boolean"/>
  <Output name="Output_6" address="6" type="Boolean"/>
  <Output name="Output_7" address="7" type="Boolean"/>
  <Output name="Output_8" address="8" type="Boolean"/>
  <Output name="Output_9" address="9" type="Boolean"/>
  <Output name="Output_10" address="10" type="Boolean"/>
```

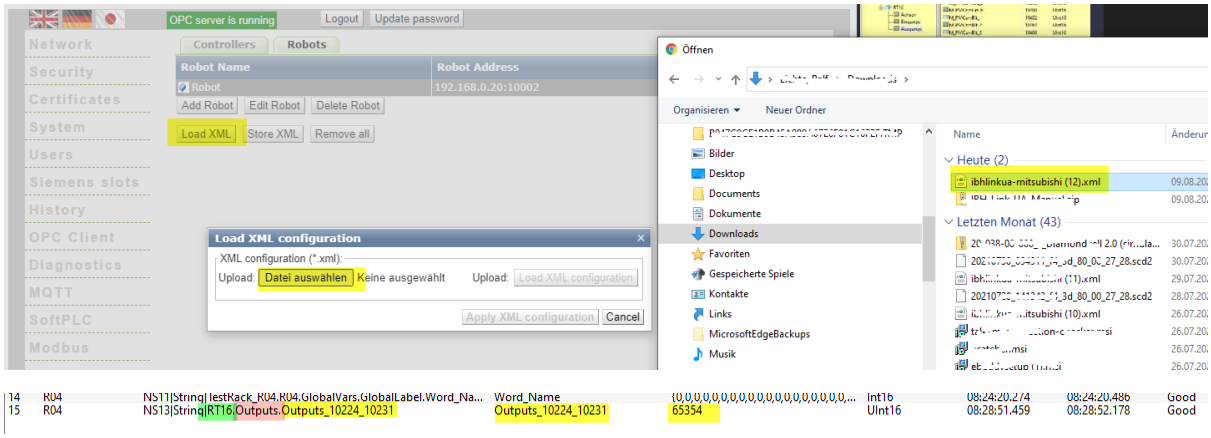
The required inputs or outputs can then be inserted here as new lines accordingly, e.g. for M_Outw(10224):

```
<Outputs>
  <Output name="Outputs_10224_10231" address="10224" type="UInt16"/>
  <Output name="Outputs_0_15" address="0" type="UInt16"/>
  <Output name="Outputs_16_31" address="16" type="UInt16"/>
  <Output name="Outputs_32_47" address="32" type="UInt16"/>
  <Output name="Outputs_48_63" address="48" type="UInt16"/>
  <Output name="Output_0" address="0" type="Boolean"/>
  <Output name="Output_1" address="1" type="Boolean"/>
  <Output name="Output_2" address="2" type="Boolean"/>
  <Output name="Output_3" address="3" type="Boolean"/>
```

Once all the desired entries have been made, the file is saved.

Load XML file

The modified file can now be loaded again in the RD55-IBH web server:



4.3 Inverter

The following chapter only deals with the connection with inverters of the 800 series.

Depending on the inverter type, the permissible address range of the variables that can be read via SLMP is different. This quick start guide only provides an overview using examples; details about the individual parameters and registers can be found in the corresponding manuals for the Mitsubishi Electric Inverter.

What they have in common, however, are the settings of the inverter parameters for Ethernet communication.

IP Address

The IP address of the inverter must be in the same network area as the control level (CH2) of the IoT gateway. In the quickstart example this area is at "192.168.0.xx", the inverter gets the address "2" for the A800/F800 Inverter and “11” for the E800 through the corresponding entries in parameters 1434 to 1437.

1434	Ethernet IP address 1	0 to 255	1	192	192
1435	Ethernet IP address 2	0 to 255	1	168	168
1436	Ethernet IP address 3	0 to 255	1	50	0
1437	Ethernet IP address 4	0 to 255	1	1	2

Other communication settings may have to be carried out, such as adapting the subnet mask (parameters 1438 to 1441).

1438	Subnet mask 1	0 to 255	1	255	255
1439	Subnet mask 2	0 to 255	1	255	255
1440	Subnet mask 3	0 to 255	1	255	255
1441	Subnet mask 4	0 to 255	1	0	0

For all other settings, please refer to the corresponding inverter manuals.

SLMP Connection

The communication between the IoT gateway and the inverter takes place via SLMP TCP / IP. To do this, the value 5012 or 5013 must be entered in one of the inverter parameters 1427 to 1429.

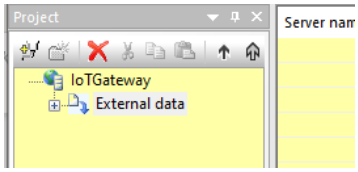
Pr.1427 to Pr.1429 setting	Application*1	Protocol*1	Number of connectable clients	Refer to page
502	Modbus/TCP	TCP/IP	3	38
5000	MELSOFT / FA product connection	UDP/IP	No limit	25
5001 (**Pr.1427** initial value)				
5002		TCP/IP	1	
5006		UDP/IP	No limit	
5007		TCP/IP	1	
5008		UDP/IP	No limit	
5010	SLMP	UDP/IP	No limit	26
5011				
5012		TCP/IP	1	
5013				
45237 (**Pr.1428** setting)	iQSS	UDP/IP	No limit	*2
9999 (**Pr.1429** initial value)	Unselected			—

In the example, parameter 1428 is set.

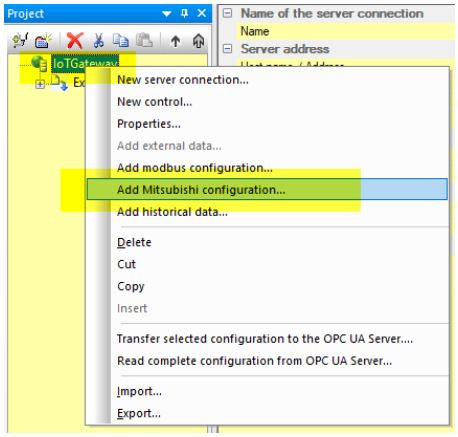
1427	Ethernet function selection 1	502,5000 to 5002,5006 to 5008,5010 to 5013,45237,61450,9999	1	5001	5001
1428	Ethernet function selection 2	502,5000 to 5002,5006 to 5008,5010 to 5013,45237,61450,9999	1	45237	5012
1429	Ethernet function selection 3	502,5000 to 5002,5006 to 5008,5010 to 5013,45237,61450,9999	1	9999	5008

Add Inverter to IoT Gateway

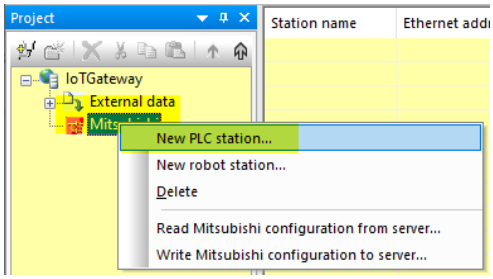
After the communication parameters have been set, the inverter can be registered as a station in the IoT gateway. For this we use the IBHsoftec software "OPCUAEdit". The IoT Gateway server has already been created here.



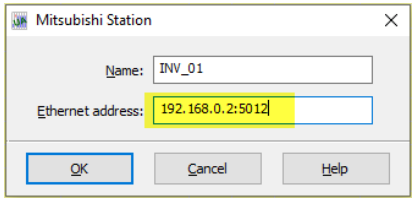
Right-click on the “IoTGateway” server and select “Add Mitsubishi configuration”.



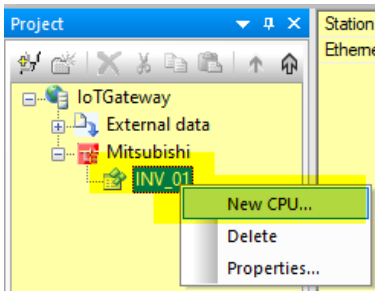
The inverter is then added as "New PLC station ...".



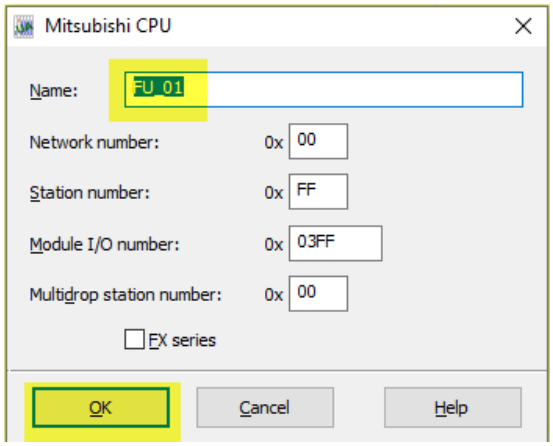
The name can be freely selected, the Ethernet address must correspond to the inputs of the inverter parameters.



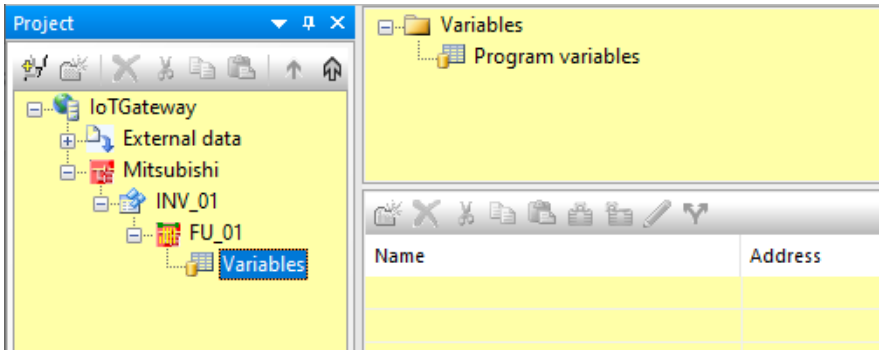
Now a "New CPU ..." has to be added

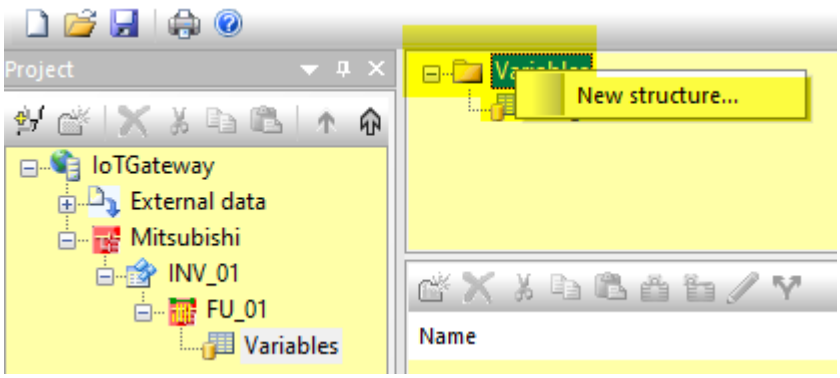


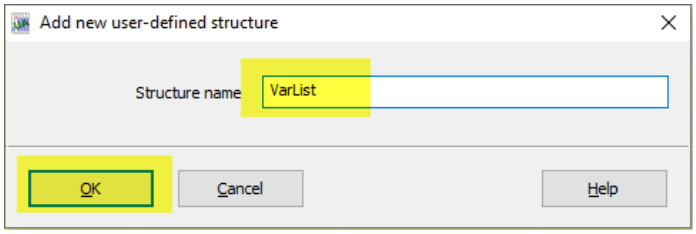
You only have to enter a freely selectable name and press "OK"



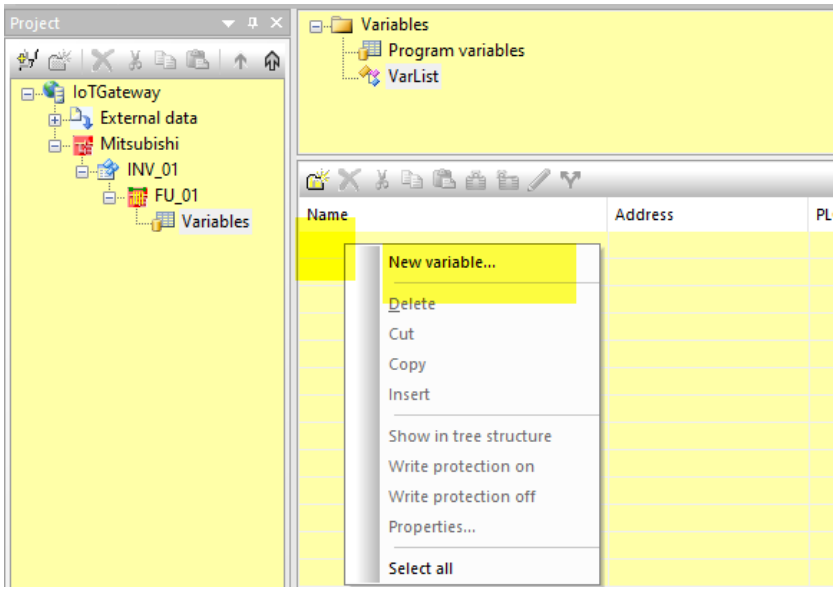
The variables are entered in a structure that has yet to be created.



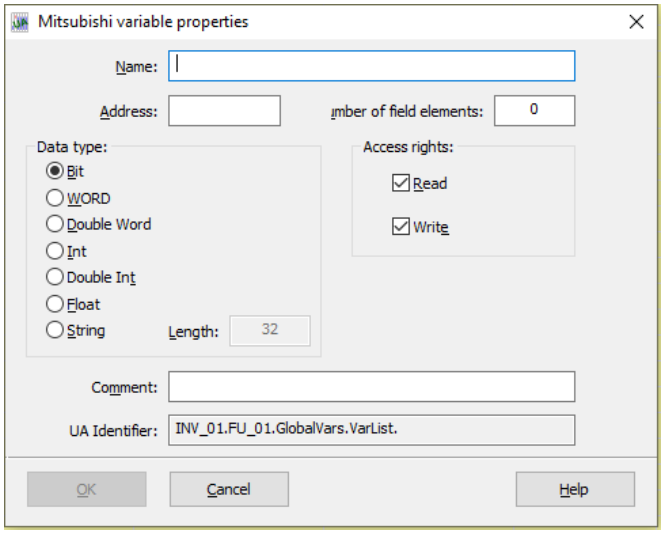




The desired variables can now be added.



The selection of the possible variables is now dependent on the inverter type



4.3.1 A800/F800

With the A800 / F800 inverters, the status can be monitored and controlled with certain special registers.

In order for this to be possible, the PLC function of the inverter must be activated

Below are a few examples; the complete list of special registers can be found in the relevant manual.

Device number	Name	Setting increments	Data example
SD1133	Output frequency monitor	0.01 Hz	Device content 6000→60.00 Hz
SD1134	Output current monitor	0.01 A	Device content 200→2.00 A
SD1135	Output voltage monitor	0.01 V	Device content 1000→10.0 V

Example: error history

	b15 to b8	b7 to b0
SD1136	Faults history 2	Faults history 1
SD1137	Faults history 4	Faults history 3
SD1138	Faults history 6	Faults history 5
SD1139	Faults history 8	Faults history 7

Newer

Older

Error code	Fault record
H00	No failure
H10	E.OC1
H11	E.OC2
H12	E.OC3
H20	E.OV1
H21	E.OV2
H22	E.OV3
H30	E.THT
H31	E.THM
H40	E.FIN
H50	E.IPF
H51	E.UVT
H52	E.ILF
H60	E.OLT
H61	E.SOT
H70	E.BE

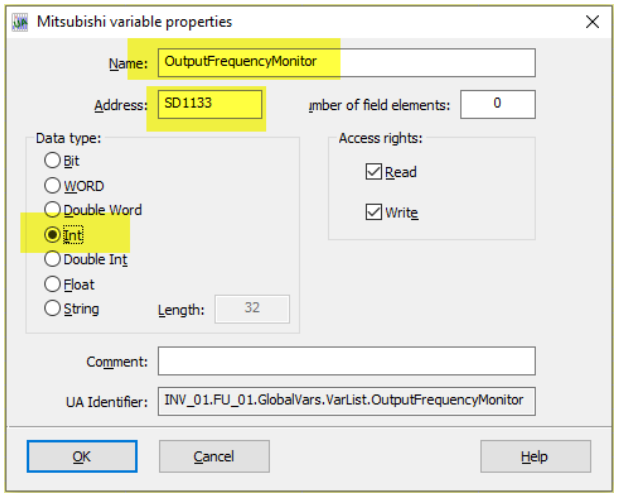
Error code	Fault record
H80	E.GF
H81	E.LF
H90	E.OHT
H91	E.PTC
HA0	E.OPT
HA1	E.OP1
HA4	E.16
HA5	E.17
HA6	E.18
HA7	E.19
HA8	E.20
HB0	E.PE
HB1	E.PUE
HB2	E.RET
HB3	E.PE2
HC0	E.CPU
HC1	E.CTE

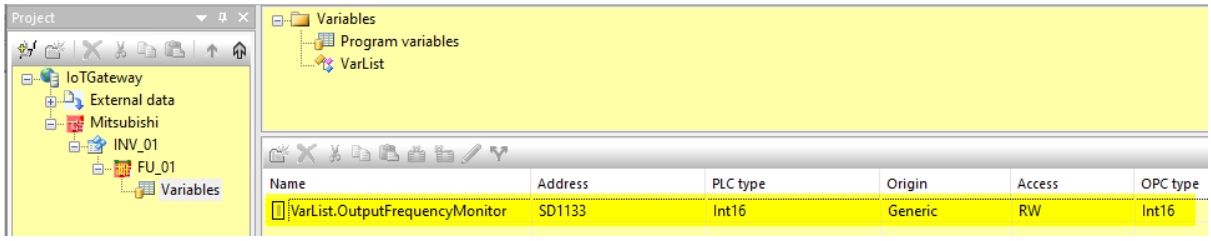
Error code	Fault record
HC2	E.P24
HC4	E.CDO
HC5	E.IOH
HC6	E.SER
HC7	E.AIE
HC8	E.USB
HC9	E.SAF
HCA	E.PBT
HD0	E.OS
HD1	E.OSD*1
HD2	E.ECT*1
HD3	E.OD*1
HD5	E.MB1*1
HD6	E.MB2*1
HD7	E.MB3*1
HD8	E.MB4*1
HD9	E.MB5*1

Error code	Fault record
HDA	E.MB6*1
HDB	E.MB7*1
HDC	E.EP*1
HDE	E.MP*1
HE1	E.IAH*2
HE4	E.LCI
HE5	E.PCH
HE6	E.PID
HF1	E.1
HF2	E.2
HF3	E.3
HF5	E.5
HF6	E.6
HF7	E.7
HFB	E.11*1
HFD	E.13

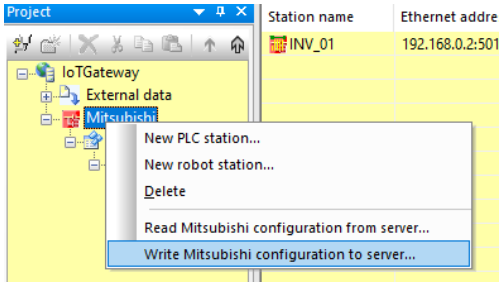
Entering the variable

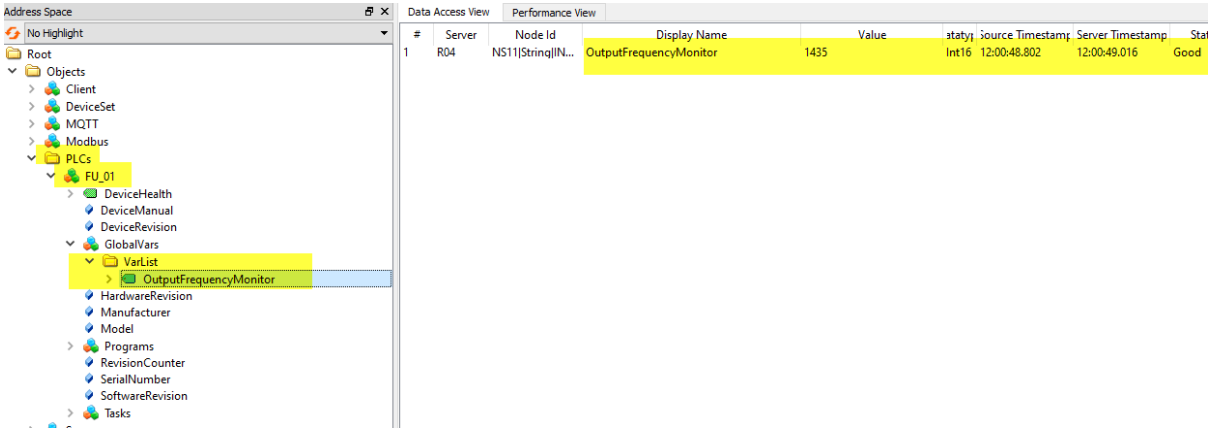
Example 1: output frequency





Right-click on the Mitsubishi station and then select "Write Mitsubishi configuration to server ..." to transfer the changes to the IoT gateway.



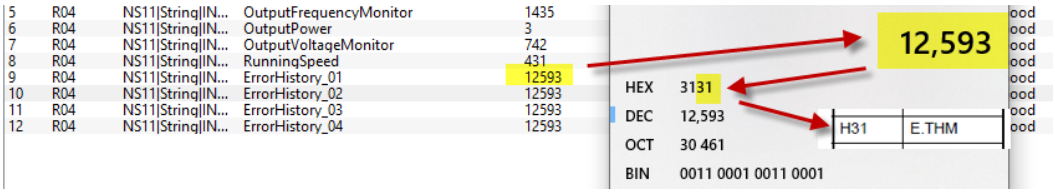


Further parameters can be added:

Name	Address	PLC type	Origin	Access	OPC type
InverterStatus	SD1151	Int16	Generic	RW	Int16
OutputFrequencyMonitor	SD1133	Int16	Generic	RW	Int16
OutputCurrentMonitor	SD1134	Int16	Generic	RW	Int16
OutputVoltageMonitor	SD1135	Int16	Generic	RW	Int16
InputPower	SD1160	Int16	Generic	RW	Int16
OutputPower	SD1161	Int16	Generic	RW	Int16
RunningSpeed	SD1153	Int16	Generic	RW	Int16
MotorTorque	SD1154	Int16	Generic	RW	Int16
ErrorHistory_01	SD1136	Int	Generic	RW	Int16
ErrorHistory_02	SD1137	Int	Generic	RW	Int16
ErrorHistory_03	SD1138	Int	Generic	RW	Int16
ErrorHistory_04	SD1139	Int	Generic	RW	Int16

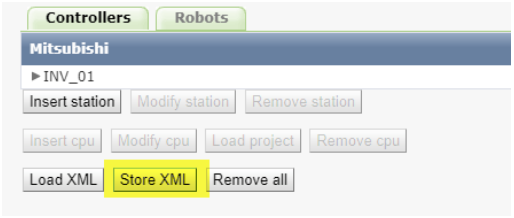
#	Server	Node Id	Display Name	Value	ata
1	R04	NS11|String|IN...	InputPower	3	Int
2	R04	NS11|String|IN...	InverterStatus	-32693	Int
3	R04	NS11|String|IN...	MotorTorque	0	Int
4	R04	NS11|String|IN...	OutputCurrentMonitor	63	Int
5	R04	NS11|String|IN...	OutputFrequencyMonitor	1435	Int
6	R04	NS11|String|IN...	OutputPower	3	Int
7	R04	NS11|String|IN...	OutputVoltageMonitor	742	Int
8	R04	NS11|String|IN...	RunningSpeed	431	Int
9	R04	NS11|String|IN...	ErrorHistory_01	12593	Int
10	R04	NS11|String|IN...	ErrorHistory_02	12593	Int
11	R04	NS11|String|IN...	ErrorHistory_03	12593	Int
12	R04	NS11|String|IN...	ErrorHistory_04	12593	Int

Example: Interpretation of the displayed value of the “Error History“:



XML-Ex-/Import

The configuration generated in the OPCUAEdit tool can be exported as an XML file in the IoT Gateway web server.



This generated XML file can then be edited in any text editor and then transferred back to the IoT Gateway using "Load XML".

Sample XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<Objects Version="1">
    <Device name="INV_01" URL="192.168.0.2:5012">
        <CPU name="FU_01" network="0" pcno="255" moduleio="1023" moduleno="0" hexio="1">
            <GlobalVars>
                <struct name="VarList">
                    <Variable name="InverterStatus" address="SD1151" type="Int16"/>
                    <Variable name="OutputFrequencyMonitor" address="SD1133" type="Int16"/>
                    <Variable name="OutputCurrentMonitor" address="SD1134" type="Int16"/>
                    <Variable name="OutputVoltageMonitor" address="SD1135" type="Int16"/>
                    <Variable name="InputPower" address="SD1160" type="Int16"/>
                    <Variable name="OutputPower" address="SD1161" type="Int16"/>
                    <Variable name="RunningSpeed" address="SD1153" type="Int16"/>
                    <Variable name="MotorTorque" address="SD1154" type="Int16"/>
                    <Variable name="ErrorHistory_01" address="SD1136" type="Int16"/>
                    <Variable name="ErrorHistory_02" address="SD1137" type="Int16"/>
                    <Variable name="ErrorHistory_03" address="SD1138" type="Int16"/>
                    <Variable name="ErrorHistory_04" address="SD1139" type="Int16"/>
                </struct>
            </GlobalVars>
        </CPU>
    </Device>
</Objects>
```

4.3.2 E800-E

With the E800 inverters, the status can be monitored and controlled with certain special SLMP Link registers.

In order for this to be possible, the PLC function of the inverter must be activated

Link Register

Parameter

Pr.	Register	Name	Read/write	Remarks
0 to 999	W0 to W999	For details on parameter names, refer to the parameter list in the FR-E800 Instruction Manual (Function).	Read/write	
C2 (902)	W902	Terminal 2 frequency setting bias (frequency)	Read/write	
C3 (902)	W4802	Terminal 2 frequency setting bias (analog value)	Read/write	Analog value (%) set in C3 (902)
	W4902	Terminal 2 frequency setting bias (terminal analog value)	Read	Analog value (%) of the voltage (current) applied to terminal 2
125 (903)	W903	Terminal 2 frequency setting gain (frequency)	Read/write	
C4 (903)	W4803	Terminal 2 frequency setting gain (analog value)	Read/write	Analog value (%) set in C4 (903)
	W4903	Terminal 2 frequency setting gain (terminal analog value)	Read	Analog value (%) of the voltage (current) applied to terminal 2
C5 (904)	W904	Terminal 4 frequency setting bias (frequency)	Read/write	
C6 (904)	W4804	Terminal 4 frequency setting bias (analog value)	Read/write	Analog value (%) set in C6 (904)
	W4904	Terminal 4 frequency setting bias (terminal analog value)	Read	Analog value (%) of the current (voltage) applied to terminal 4
126 (905)	W905	Terminal 4 frequency setting gain (frequency)	Read/write	
C7 (905)	W4805	Terminal 4 frequency setting gain (analog value)	Read/write	Analog value (%) set in C7 (905)
	W4905	Terminal 4 frequency setting gain (terminal analog value)	Read	Analog value (%) of the current (voltage) applied to terminal 4
C39 (932)	W4832	Terminal 4 bias (torque/magnetic flux)	Read/write	Analog value (%) set in C39 (932)
	W4932	Terminal 4 bias (torque/magnetic flux) (terminal analog value)	Read	Analog value (%) of the current (voltage) applied to terminal 4
C41 (933)	W4833	Terminal 4 gain (torque/magnetic flux)	Read/write	Analog value (%) set in C41 (933)
	W4933	Terminal 4 gain (torque/magnetic flux) (terminal analog value)	Read	Analog value (%) of the current (voltage) applied to terminal 4
C43 (934)	W4834	PID display bias analog value	Read/write	Analog value (%) set in C43 (934)
	W4934	PID display bias analog value (terminal analog value)	Read	Analog value (%) of the current (voltage) applied to terminal 4
C45 (935)	W4835	PID display gain analog value	Read/write	Analog value (%) set in C45 (935)
	W4935	PID display gain analog value (terminal analog value)	Read	Analog value (%) of the current (voltage) applied to terminal 4
1000 to 1499	W1000 to W1499	For details on parameter names, refer to the parameter list in the FR-E800 Instruction Manual (Function).	Read/write	

Inverter Status

Register	Monitor item	Read/write	Register	Monitor item	Read/write
W5001	Output frequency/speed	Read	W5040	PLC function user monitor 1	Read
W5002	Output current	Read	W5041	PLC function user monitor 2	Read
W5003	Output voltage	Read	W5042	PLC function user monitor 3	Read
W5005	Set frequency / motor speed setting	Read	W5045	Station number (CC-Link)	Read
W5006	Operation speed	Read	W5050	Energy saving effect	Read
W5007	Motor torque	Read	W5051	Cumulative energy saving	Read
W5008	Converter output voltage	Read	W5052	PID set point	Read
W5009	Regenerative brake duty	Read	W5053	PID measured value	Read
W5010	Electronic thermal O/L relay load factor	Read	W5054	PID deviation	Read
W5011	Output current peak value	Read	W5058	Option input terminal status 1 (for communication)	Read
W5012	Converter output voltage peak value	Read	W5059	Option input terminal status 2 (for communication)	Read
W5014	Output power	Read	W5060	Option output terminal status (for communication)	Read
W5015	Input terminal status	Read	W5061	Motor thermal load factor	Read
W5016	Output terminal status	Read	W5062	Inverter thermal load factor	Read
W5017	Load meter	Read	W5067	PID measured value 2	Read
W5018	Motor excitation current	Read	W5077	32-bit cumulative energy (lower 16 bits)	Read
W5020	Cumulative energization time	Read	W5078	32-bit cumulative energy (upper 16 bits)	Read
W5023	Actual operation time	Read	W5079	32-bit cumulative energy (lower 16 bits)	Read
W5024	Motor load factor	Read	W5080	32-bit cumulative energy (upper 16 bits)	Read
W5025	Cumulative power	Read	W5083	BACnet valid APDU counter	Read
W5032	Torque command	Read	W5091	PID manipulated amount	Read
W5033	Torque current command	Read	W5097	Dancer main speed setting	Read
W5038	Trace status	Read			

Preventive maintenance data

Register	Definition	Read/write	Remarks
W6000	Control method	Read	H02: V/F control H04: Advanced magnetic flux vector control H08: Real sensorless vector control H09: Vector control H18: PM sensorless vector control

Model information monitor

Register	Definition	Read/write	Remarks
W8001	Model (1st and 2nd characters)	Read	The inverter model can be read in ASCII code. "H20" (blank code) is set for blank area. Example) FR-E820-EPA: H46, H52, H2D, H45, H38, H32, H30, H2D, H45, H50, H41, H20...H20
W8002	Model (3rd and 4th characters)	Read	
W8003	Model (5th and 6th characters)	Read	
W8004	Model (7th and 8th characters)	Read	
W8005	Model (9th and 10th characters)	Read	
W8006	Model (11th and 12th characters)	Read	
W8007	Model (13th and 14th characters)	Read	
W8008	Model (15th and 16th characters)	Read	
W8009	Model (17th and 18th characters)	Read	
W8010	Model (19th and 20th characters)	Read	
W8011	Capacity (1st and 2nd characters)	Read	The capacity in the inverter model can be read in ASCII code. Data is read in increments of 0.1 kW, and rounds down to 0.01 kW increments. "H20" (blank code) is set for blank area. Example) 0.75K: " 7" (H20, H20, H20, H20, H20, H37)
W8012	Capacity (3rd and 4th characters)	Read	
W8013	Capacity (5th and 6th characters)	Read	

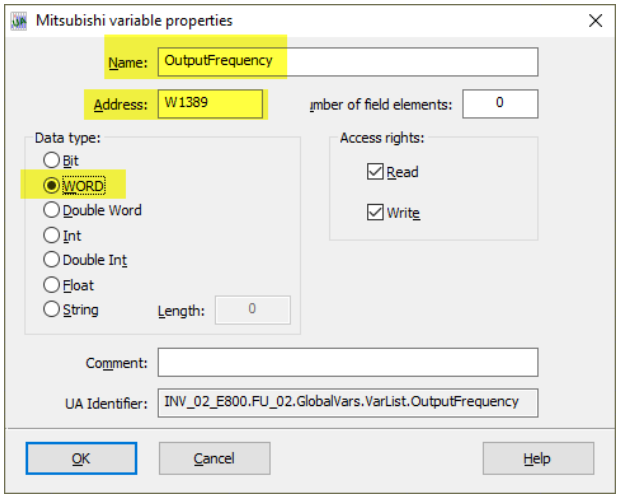
Serial number

Register	Definition	Read/write	Remarks
W8001	Model (1st and 2nd characters)	Read	The inverter model can be read in ASCII code. "H20" (blank code) is set for blank area. Example) FR-E820-EPA: H46, H52, H2D, H45, H38, H32, H30, H2D, H45, H50, H41, H20...H20
W8002	Model (3rd and 4th characters)	Read	
W8003	Model (5th and 6th characters)	Read	
W8004	Model (7th and 8th characters)	Read	
W8005	Model (9th and 10th characters)	Read	
W8006	Model (11th and 12th characters)	Read	
W8007	Model (13th and 14th characters)	Read	
W8008	Model (15th and 16th characters)	Read	
W8009	Model (17th and 18th characters)	Read	
W8010	Model (19th and 20th characters)	Read	
W8011	Capacity (1st and 2nd characters)	Read	The capacity in the inverter model can be read in ASCII code. Data is read in increments of 0.1 kW, and rounds down to 0.01 kW increments. "H20" (blank code) is set for blank area. Example) 0.75K: " 7" (H20, H20, H20, H20, H20, H37)
W8012	Capacity (3rd and 4th characters)	Read	
W8013	Capacity (5th and 6th characters)	Read	

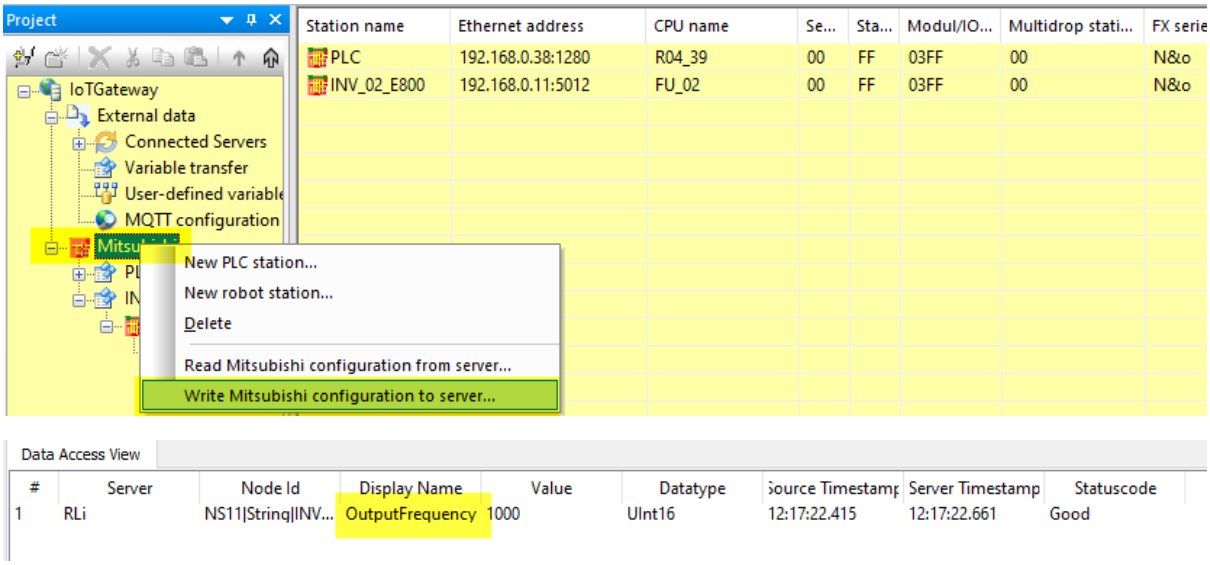
Details on the parameters can be found in the E800 manual “Instruction Manual (Communication)”.

Entering the variable

Example 1: output frequency



Right-click on the Mitsubishi station and then select "Write Mitsubishi configuration to server ..." to transfer the changes to the IoT gateway.



Further parameters can be added:

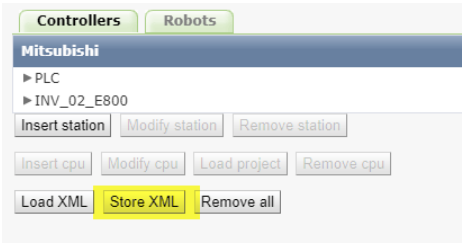
Name	Address	PLC type	Origin	Access	OPC type
VarList.OutputFrequency	W1389	Word	Generic	RW	UInt16
VarList.OutputCurrent	W1390	UInt16	Generic	RW	UInt16
VarList.OutputVoltage	W1391	UInt16	Generic	RW	UInt16
VarList.SetFrequency	W1392	UInt16	Generic	RW	UInt16
VarList.OperationSpeed	W1393	UInt16	Generic	RW	UInt16
VarList.MotorTorque	W1394	UInt16	Generic	RW	UInt16

Data Access View

#	Server	Node Id	Display Name	Value	Datatype	Source Timesta
1	RLi	NS11|String|INV...	MotorTorque	3197	UInt16	12:30:39.602
2	RLi	NS11|String|INV...	OperationSpeed	0	UInt16	12:18:54.685
3	RLi	NS11|String|INV...	OutputCurrent	3181	UInt16	12:30:42.786
4	RLi	NS11|String|INV...	OutputFrequency	1000	UInt16	12:30:37.286
5	RLi	NS11|String|INV...	OutputVoltage	0	UInt16	12:30:40.776
6	RLi	NS11|String|INV...	SetFrequency	0	UInt16	12:30:40.336

XML-Ex-/Import

The configuration generated in the OPCUAEdit tool can be exported as an XML file in the IoT Gateway web server.



This generated XML file can then be edited in any text editor and then transferred back to the IoT Gateway using "Load XML".

Sample XML file:

Beispiel-XML-Datei:

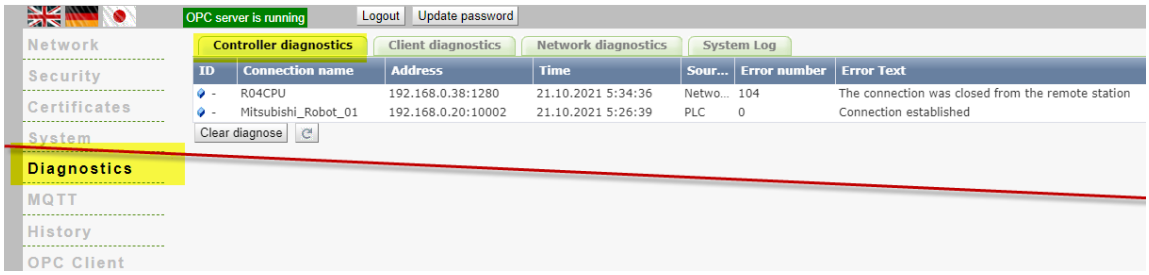
```
<?xml version="1.0" encoding="UTF-8"?>
<Objects Version="1">
<Device name="INV_02_E800" URL="192.168.0.11:5012">
    <CPU name="FU_02" network="0" pcno="255" moduleio="1023" moduleno="0" hexio="1">
        <GlobalVars>
            <struct name="VarList">
                <Variable name="OutputFrequency" address="W1389" type="UInt16"/>
                <Variable name="OutputCurrent" address="W1390" type="UInt16"/>
                <Variable name="OutputVoltage" address="W1391" type="UInt16"/>
                <Variable name="SetFrequency" address="W1392" type="UInt16"/>
                <Variable name="OperationSpeed" address="W1393" type="UInt16"/>
                <Variable name="MotorTorque" address="W1394" type="UInt16"/>
            </struct>
        </GlobalVars>
    </CPU>
</Device>
</Objects>
```

5 Diagnostics

The browser window “Diagnostics” has several tabs to display details about established or faulty connections.

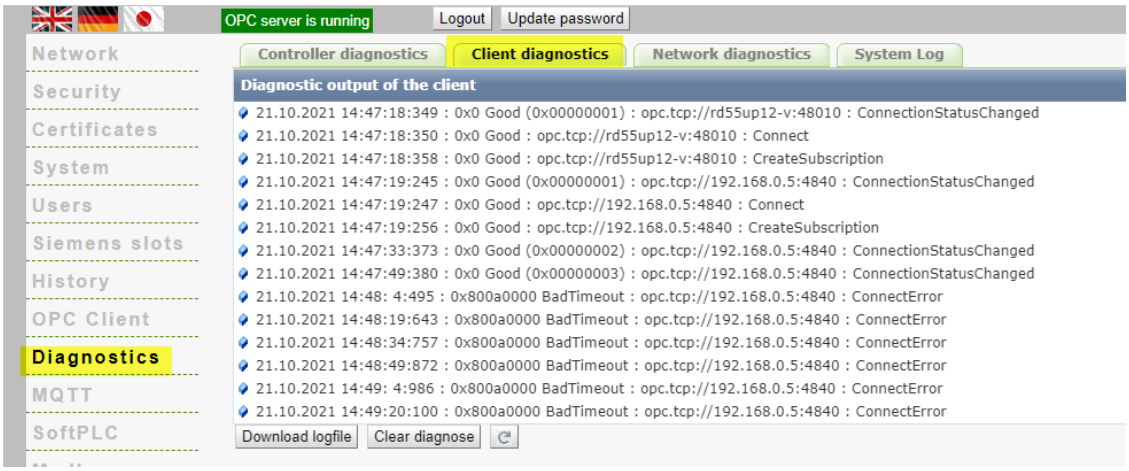
5.1 Controller diagnostics

The configured connections and their status (error-free / faulty) are displayed.

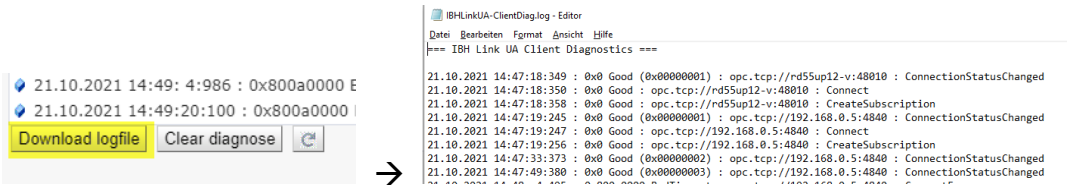


5.2 Client diagnostics

The current states of the configured OPC client connections (error-free / faulty) are displayed.

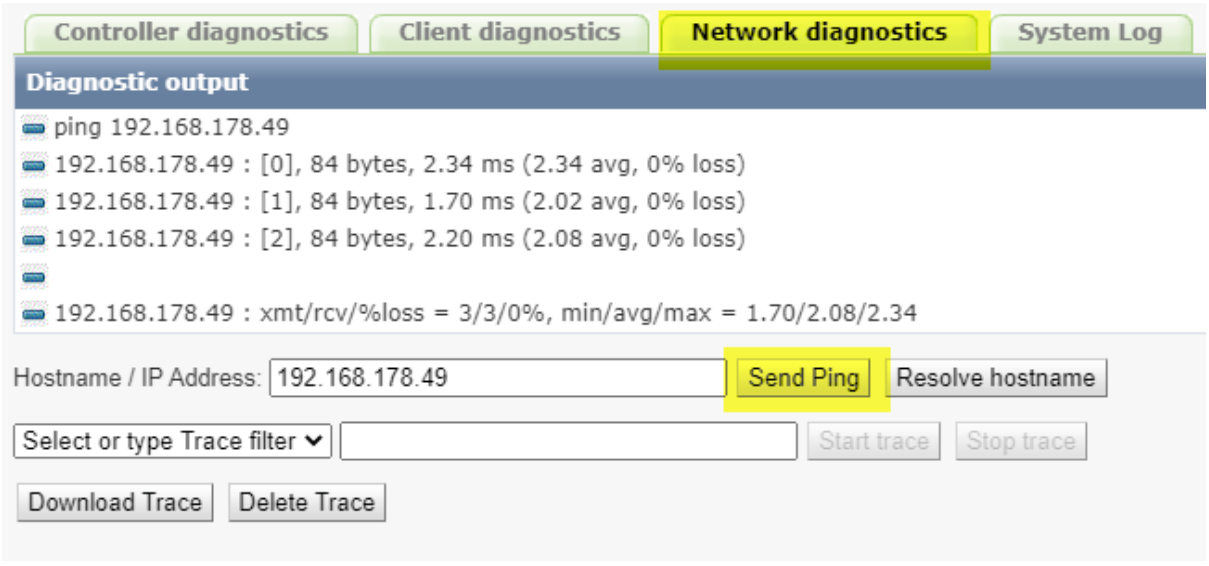


By clicking the “Download logfile” button, the saved states of the configured OPC client connections (error-free / faulty) can be displayed in an editor or saved as a text file.



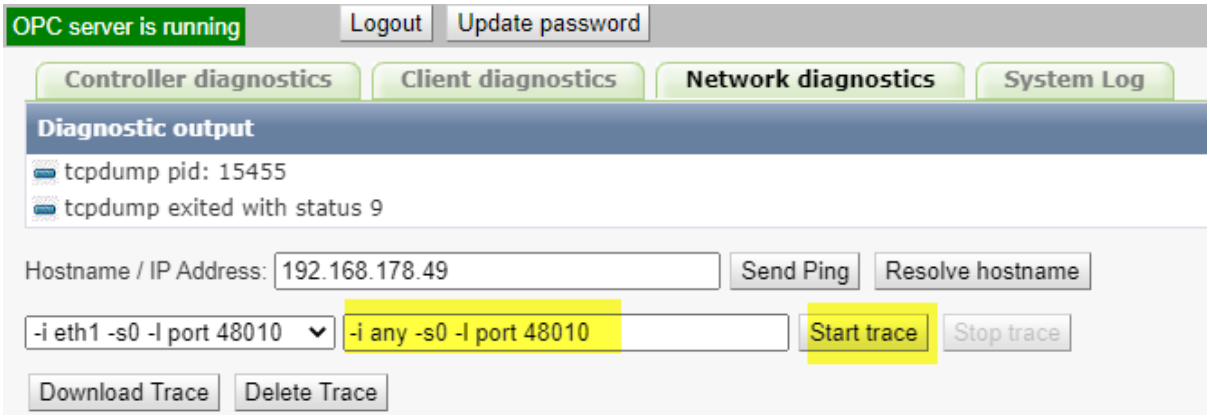
5.3 Network diagnostics

An ICMP (Internet Control Message Protocol) ping is sent to the specified IP address (host name) by clicking the “Send ping” button.



If the Wireshark diagnostic software is installed on the PC, a very extensive network analysis can be carried out.

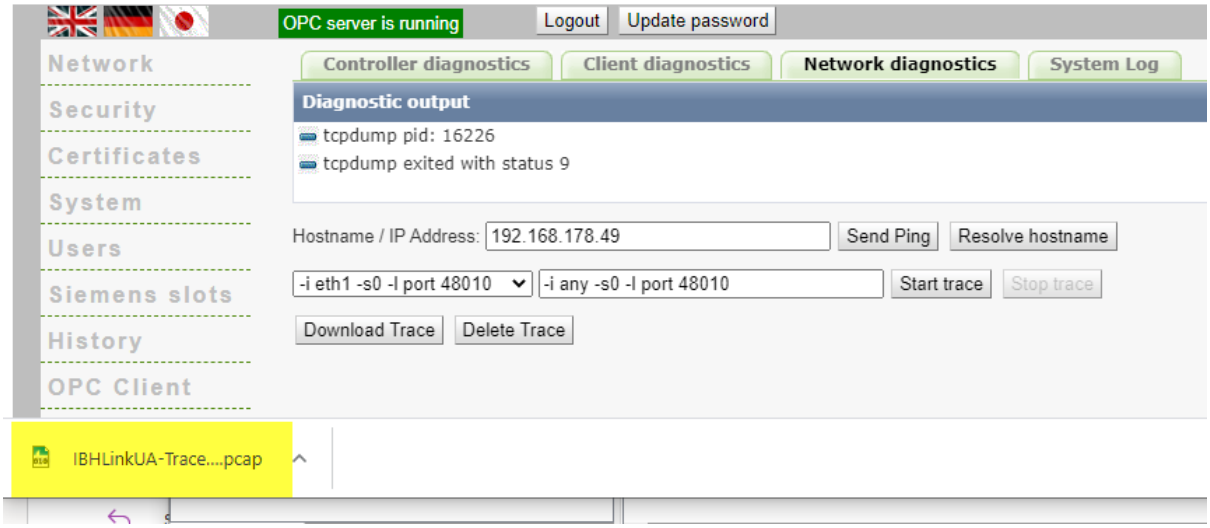
After selecting the ethernet interface and the port to be checked press the “Start trace” button.



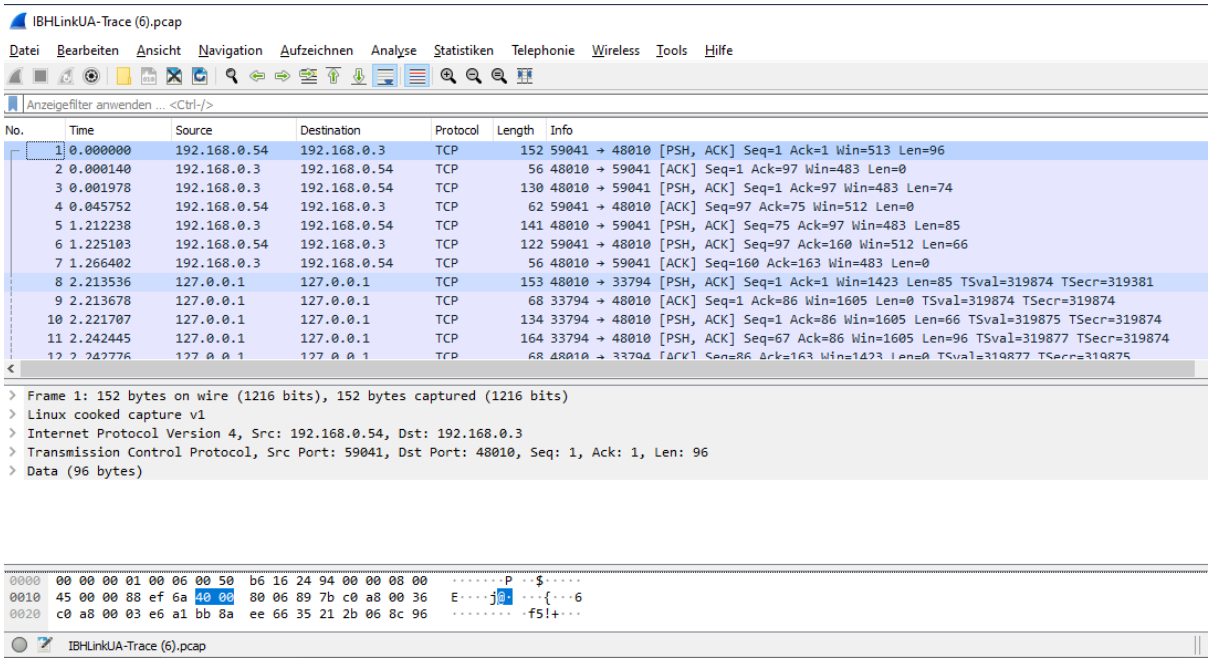
After stopping the trace, the recorded data is in a buffer inside the IoT Gateway.

The data can be downloaded to the connected PLC by pressing “Download trace”

Double clicking on the downloaded file starts Wireshark and shows the communication running on the selected interface.

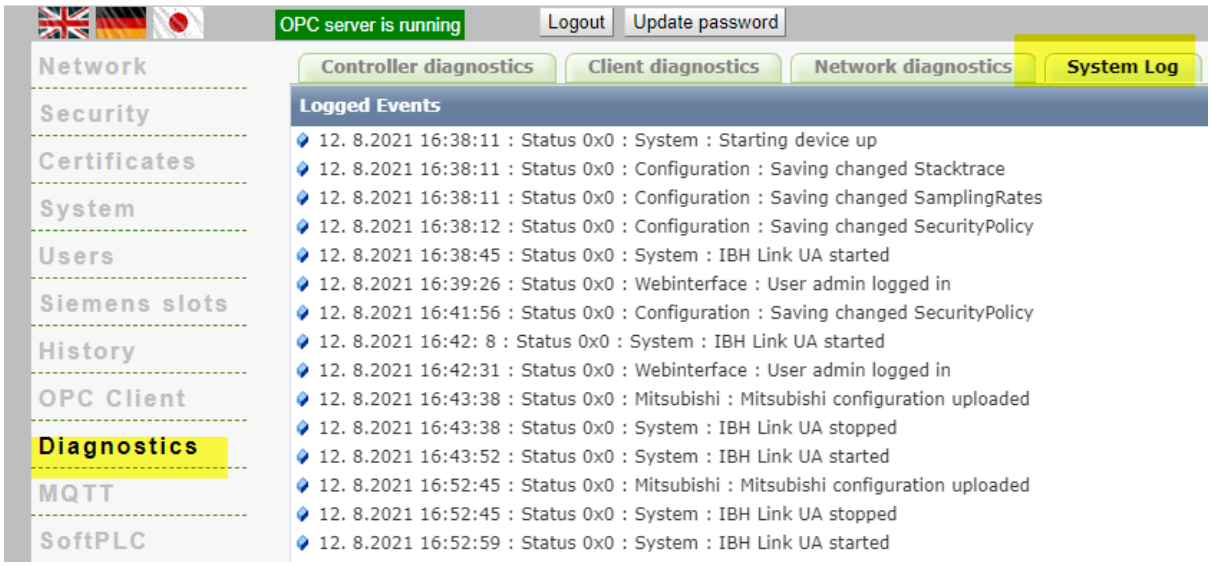


Since the evaluation of Wireshark-Trace requires some specialist knowledge, this diagnosis should be carried out in the event of a malfunction using the IBHsoftec hotline.

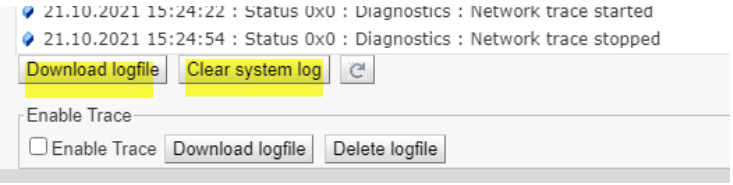


5.4 System log

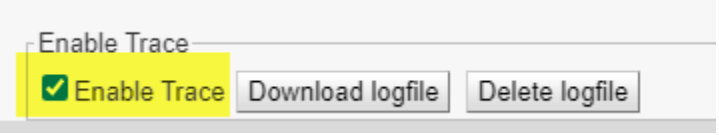
The IoT Gateway diagnosis creates a log file in which activities are recorded with a time stamp.



Buttons are provided to display the log file in an editor or to save it as a text file or to delete it. In the event of a malfunction, an analysis can be carried out using the IBHsoftec hotline.



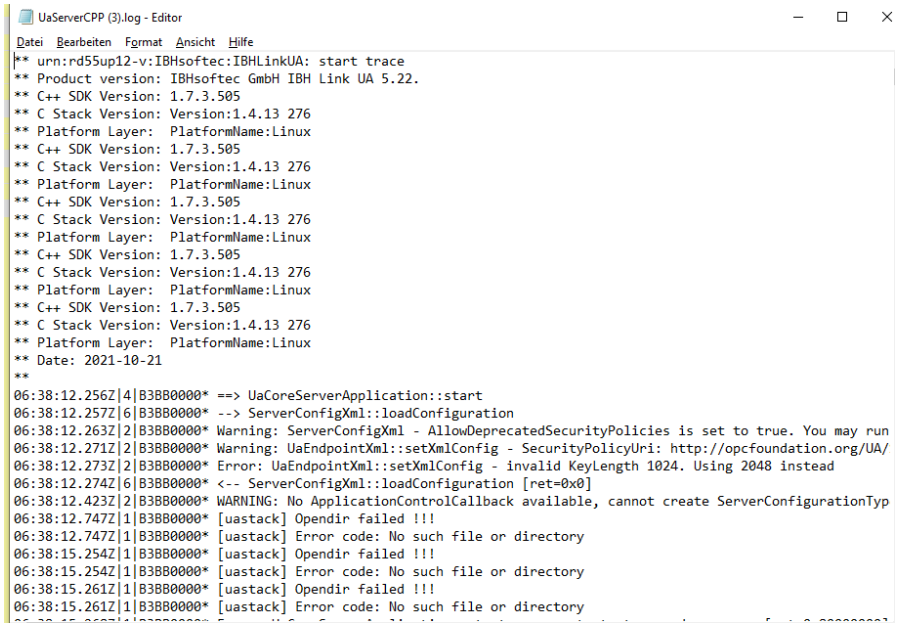
All the background operations of the IoT Gateway can be made visible by activating the “Enable Trace” checkbox.



After the event that should be evaluated has occurred, the current recorded data can be downloaded

and opened in a text editor.

To evaluate this information requires some specialist knowledge, this diagnosis should be carried out in the event of a malfunction using the IBHsoftec hotline.

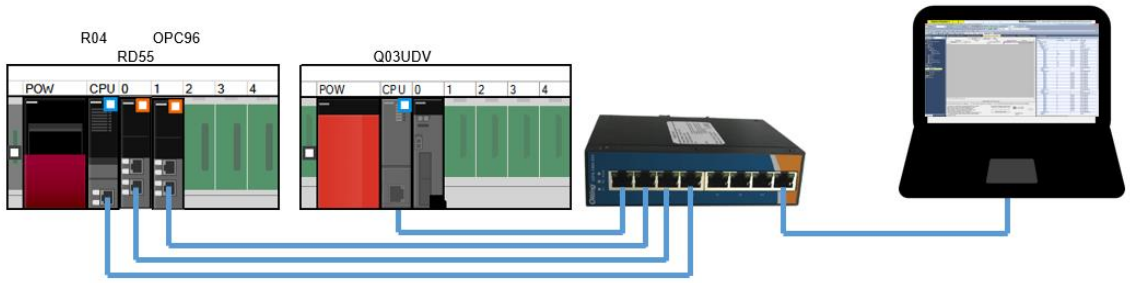


6 OPC Client Function Example

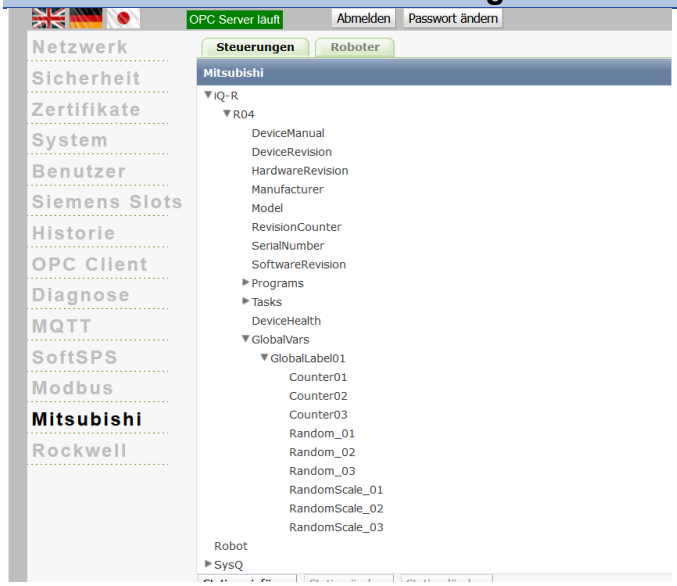
The IoT Gateway can also be used as an OPC UA client. This makes it possible to exchange data between different OPC UA servers and thus also between controllers from different manufacturers.

The following structure serves as an example:

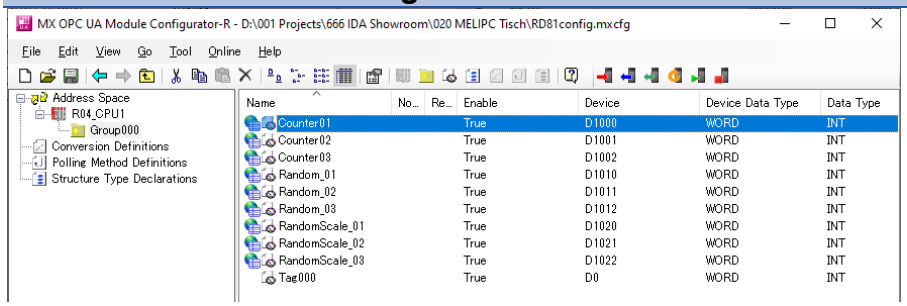
On the iQ-R rack there is an R04 CPU (192.168.0.38), an IoT Gateway (192.168.0.3) and an RD81OPC96 module (192.168.0.5). Both OPC UA servers on RD55 and RD81 are active.

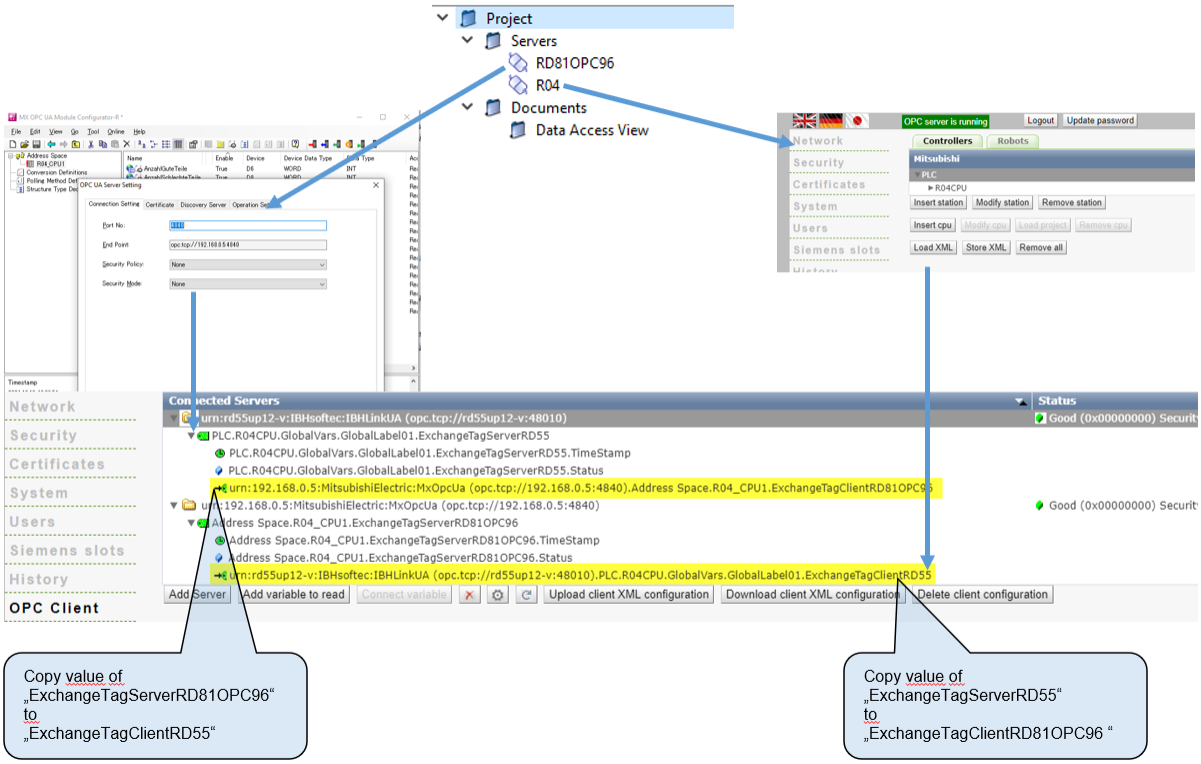


RD55UP12-V OPC Server Setting



RD81OPC96 Server Setting




Project
Servers
RD81OPC96
R04
Documents
Data Access View
Network
Security
Certificates
System
Users
Siemens slots
History
OPC Client
Connected Servers
Status
Add Server
Add variable to read
Upload client XML configuration
Download client XML configuration
Delete client configuration
Copy value of „ExchangeTagServerRD81OPC96“ to „ExchangeTagClientRD55“
Copy value of „ExchangeTagServerRD55“ to „ExchangeTagClientRD81OPC96 “

7 MODBUS

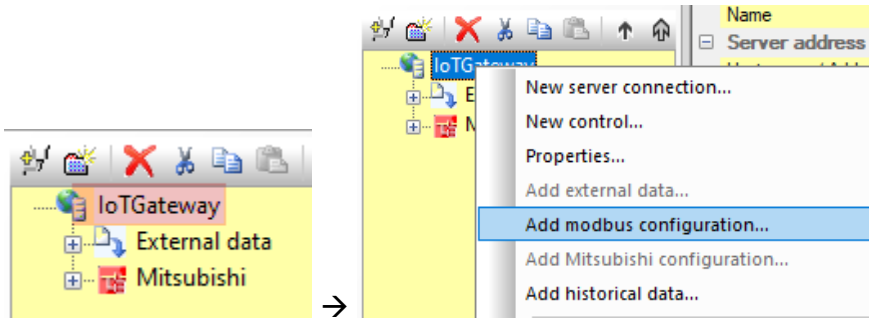
The IoT Gateway can connect to MODBUS servers and provide their data as OPC UA tags or communicate further via MQTT.

The configuration is done with the IBH OPC Editor software.

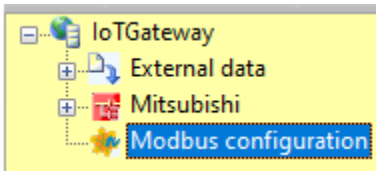
7.1 Add MODBUS Configuration

Right-clicking on an existing server configuration opens a menu window.

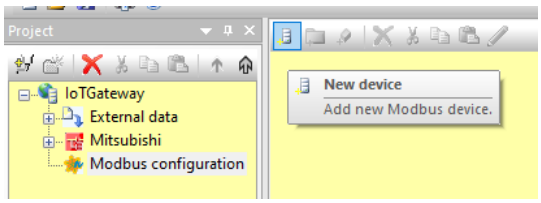
Click on "Add modbus configuration..." in this window.

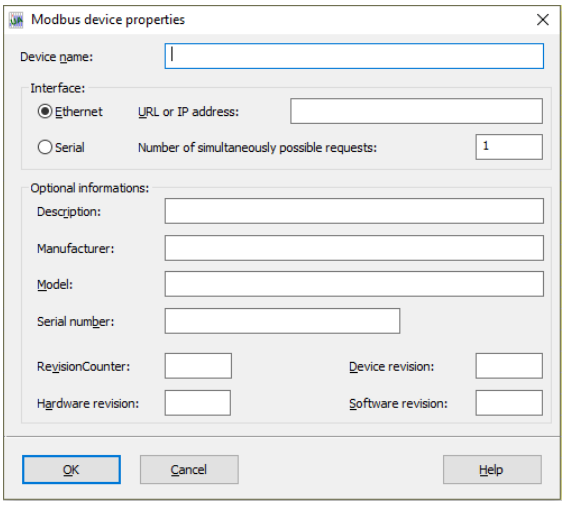


The new empty MODBUS configuration is displayed.

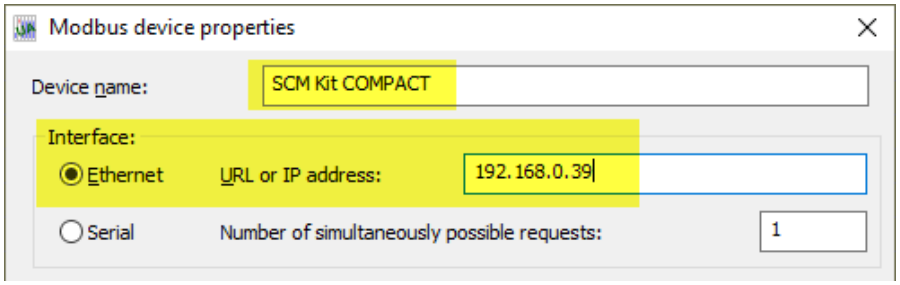


Clicking on the icon  opens the window for entering the MODBUS server connection data:

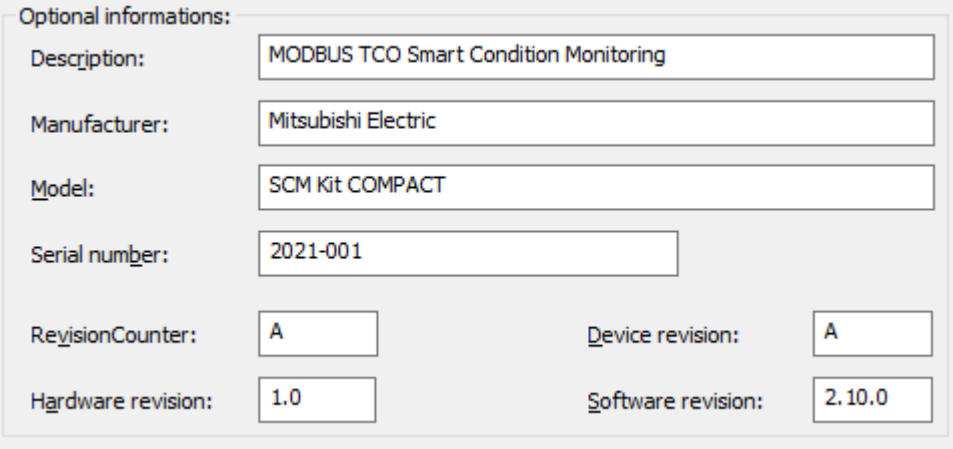




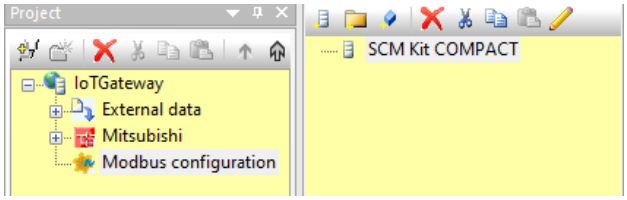
Only the connection name (freely selectable) and the connection data are required here.



The optional data is used for project documentation.

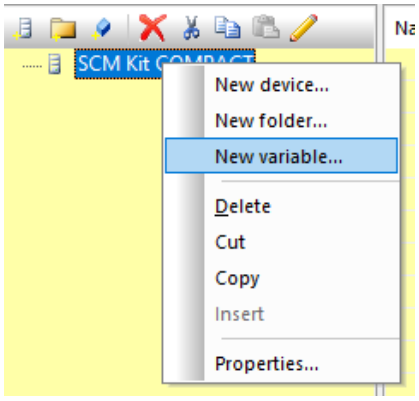


After clicking on "OK" the new MODBUS connection is displayed.

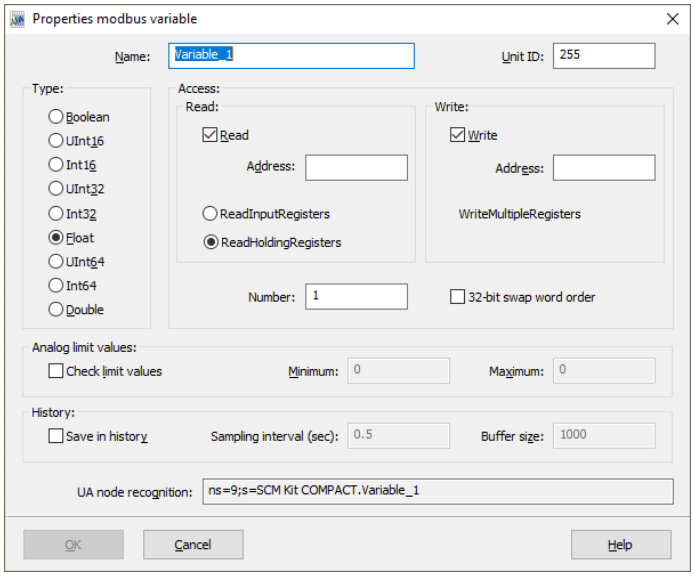


7.2 Add Variables

In the next window, which is opened by right-clicking on the desired connection, select "New variable..."



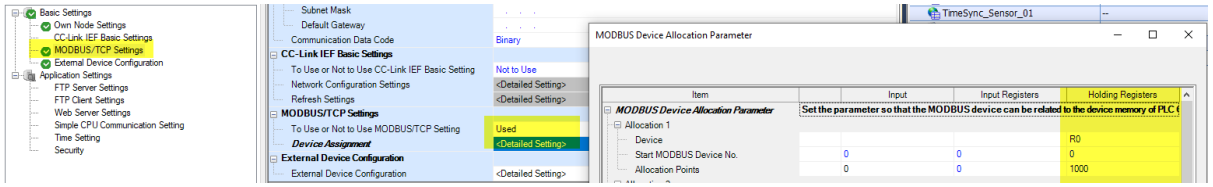
In this window you enter a name for the new variable and select the desired MODBUS data type with the corresponding MODBUS address.



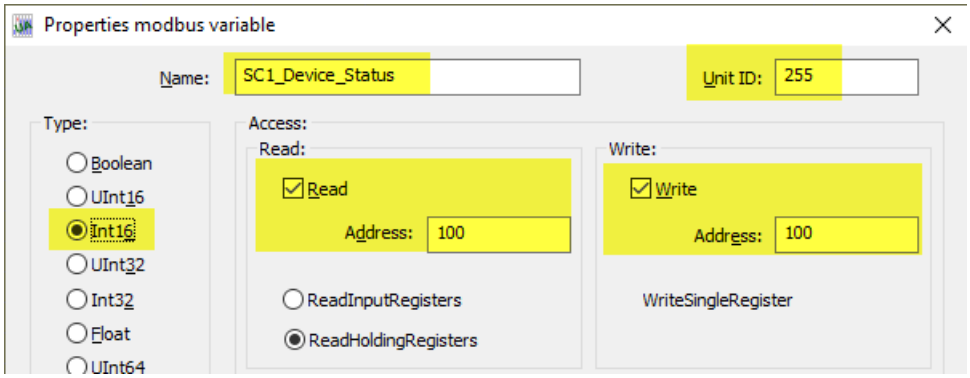
The IoT Gateway supports the following MODBUS data types:

- Inputs → Read only
- Coils → Read/Write
- Input registers → Read only
- Holdingregisters → Read/Write

In the following example, communication takes place with the MODBUS TCP server of an iQ-F controller, accessing holding registers. The address range of the holding registers is between 0 and 999

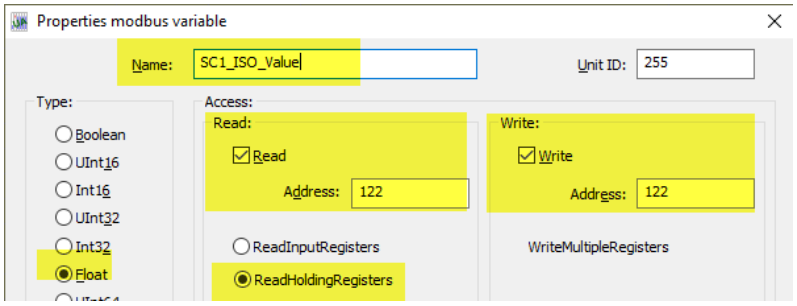


The first variable in the example is of type INT16, the data from holding register 100 is read/written



The unit ID is only relevant for a serial connection and specifies the slave number.

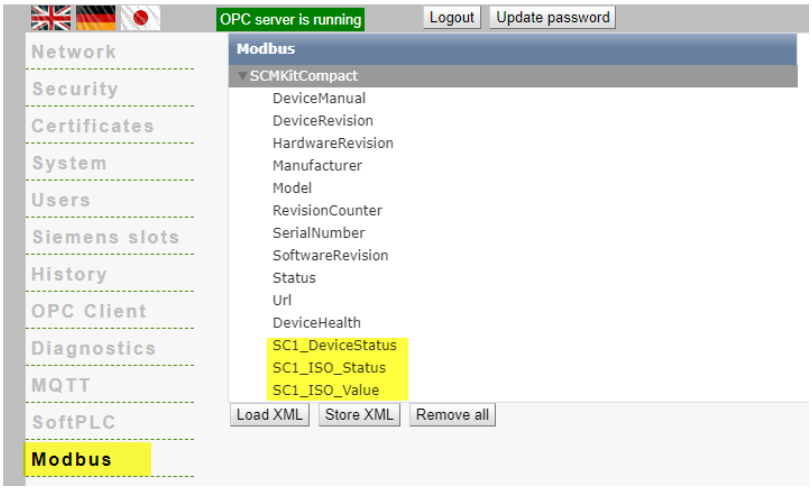
The second variable is of type FLOAT, 32bit.



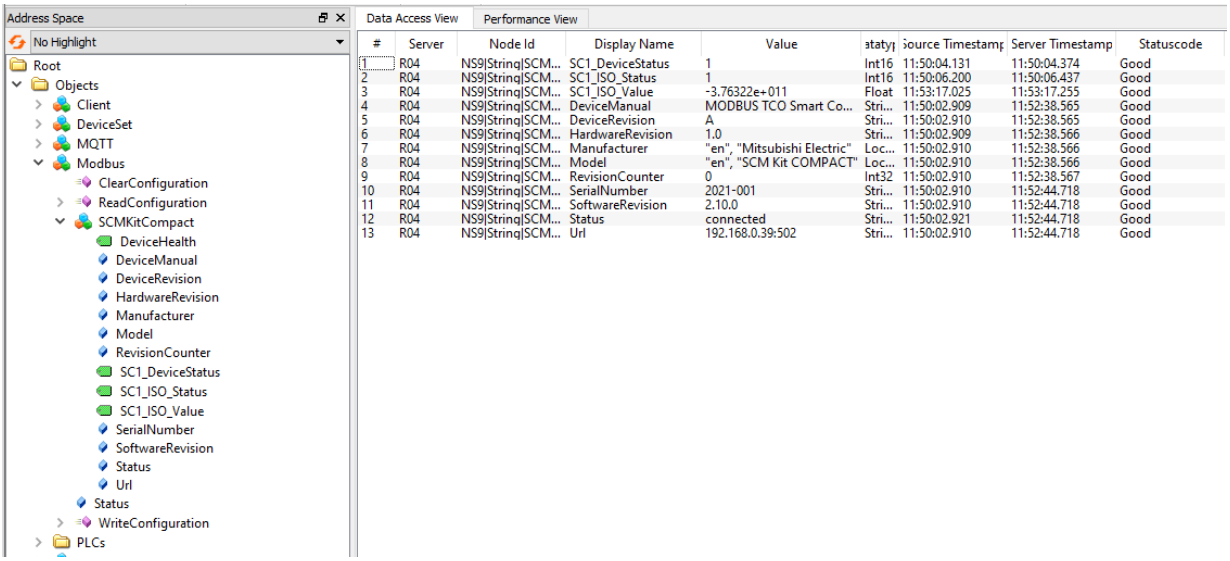
After all desired variables have been created, the configuration can be transferred to the IoT Gateway

7.3 Usage of MODBUS-Data

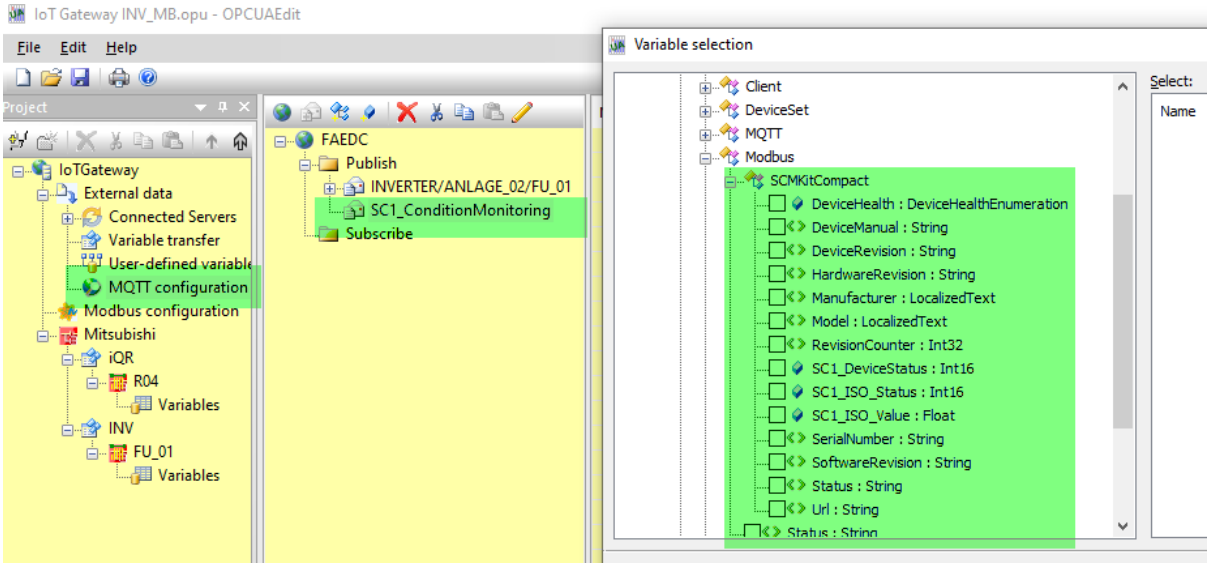
After the transfer to the IoT Gateway, the new MODBUS connection is visible in the web interface.



This means that the data is available to every OPC UA client, …



… and can also be sent to the IT world via MQTT



8 MQTT Settings

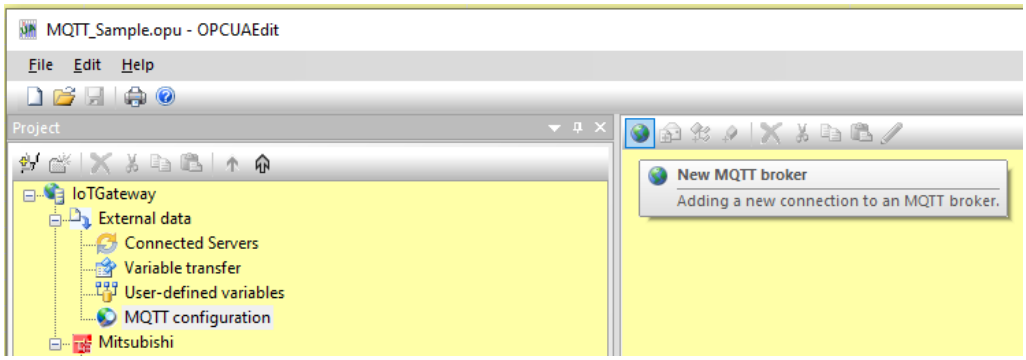
The IoT gateway supports the MQTT protocol to send data to the cloud or to receive it from the cloud.

In order to create an MQTT connection, access to an MQTT broker is required. The corresponding configuration takes place in the IBH OPCUAEdit tool.

Knowledge of the corresponding IT infrastructure and the operation of an MQTT broker must be available and are not part of this document.

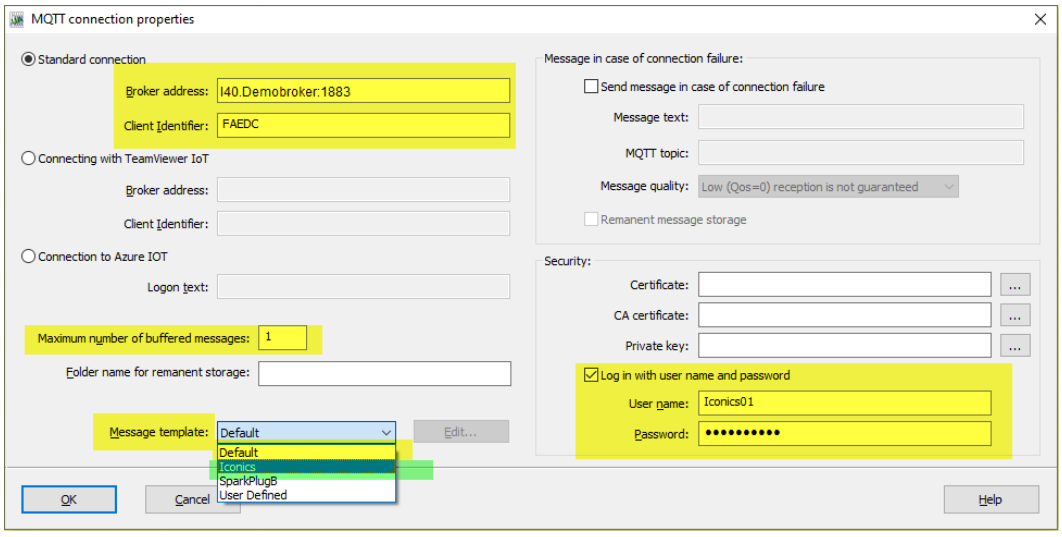
8.1 Create new Broker connection

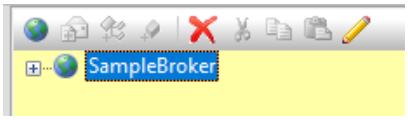
Open an existing OPCUA server configuration and select “New MQTT broker”



In the next window enter the broker IP address and the necessary login data and press “OK”.

In addition, the MQTT message format can be selected under "Message template". With "Default" the simple JSON format is used. If you want to communicate Iconics format, you can select it here accordingly.





Example Simple JSON-format:


```
{
  "LetzteAnzahlGuteTeile" : 3,
  "LetzteAnzahlSchlechteTeile" : 0,
  "LetzteAnzahlTeileGesamt" : 3,
  "LetzteAuftragsnummer" : 47110932,
  "LetzteEndZeit" : [ 2021, 12, 15, 10, 55, 45, 3 ],
  "LetzteProzentGuteTeile" : 100,
  "LetzteProzentSchlechteTeile" : 0,
  "LetzteSollStueckzahl" : 3,
  "LetzteStartZeit" : [ 2021, 12, 15, 10, 55, 32, 3 ],
  "LetzteTemperatur" : 21,
  "LetzteZykluszeitAvg" : 4667
}
```


Example Iconics-JSON-format:


```
{
  "timestamp" : 1639562267738,
  "values" : [ {
    "id" : "LetzteAnzahlGuteTeile",
    "v" : 3,
    "q" : 1,
    "t" : 1639562267738
  }, {
    "id" : "LetzteAnzahlSchlechteTeile",
    "v" : 2,
    "q" : 1,
    "t" : 1639562267738
  }, {
    "id" : "LetzteAnzahlTeileGesamt",
    "v" : 5,
    "q" : 1,
    "t" : 1639562267738
  }, {
    "id" : "LetzteAuftragsnummer",
    "v" : 47110933,
    "q" : 1,
    "t" : 1639562267738
  }, {
    "id" : "LetzteAuftragstext",
    "q" : 1,
    "t" : 1639562267738
  }, {
    "id" : "LetzteEndZeit",
    "v" : [ 2021, 12, 15, 10, 57, 47, 3 ],
```

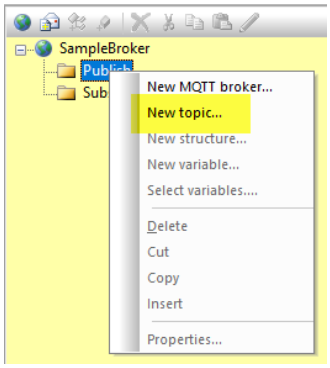

8.2 Publish of values (Send data to Broker)

Two steps are required to “publish” values:

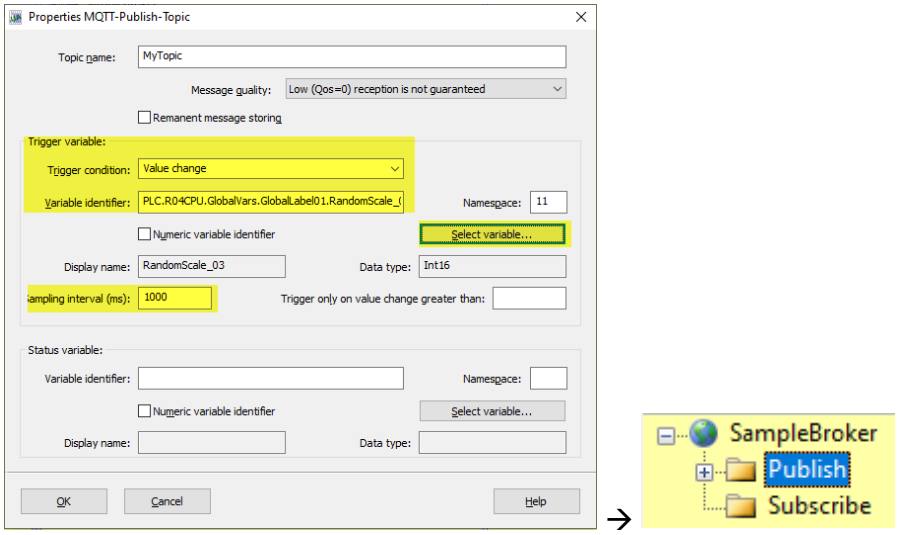
1. Create a new topic
2. Link the topic with data

8.2.1 Create Topic

Open the newly created “SampleBroker” by pressing, then right-click on “Publish” to open the menu and select “New topic”

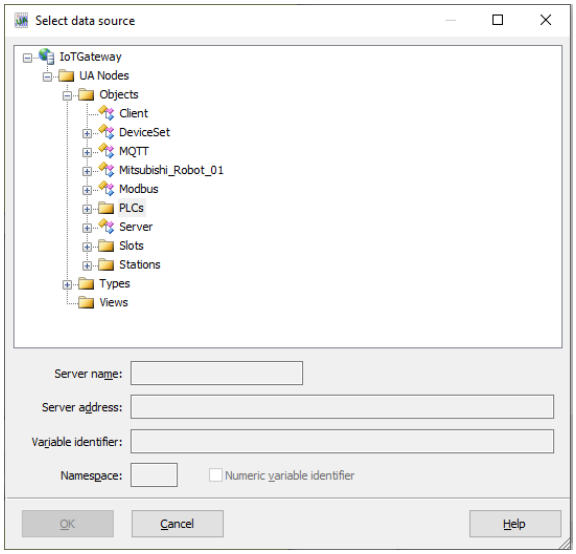


Fill in the properties of the “Publish-Topic” and press “OK”:



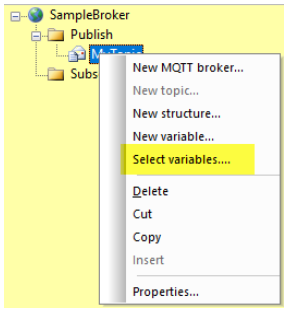
In this example, the selected variable is checked for changes every 1000ms and "Publish" is started if the condition is met.

The variable to be checked can be selected from the OPC UA server (UA nodes).

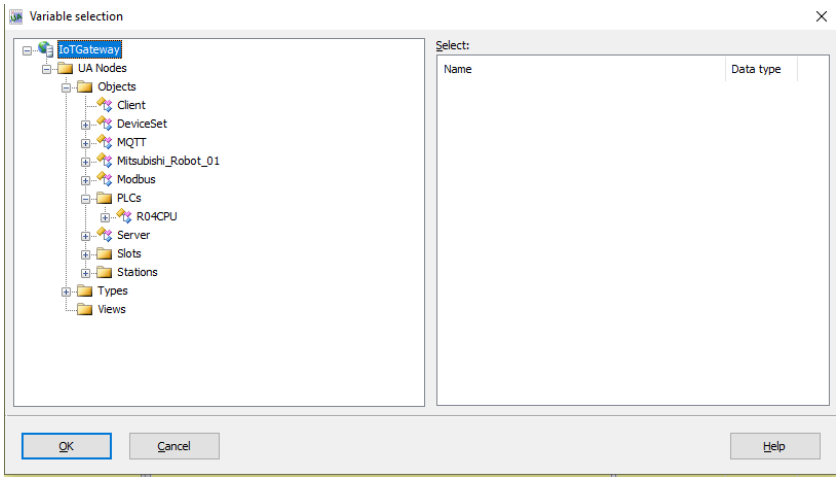


8.2.2 Create new variable (data to be sent)

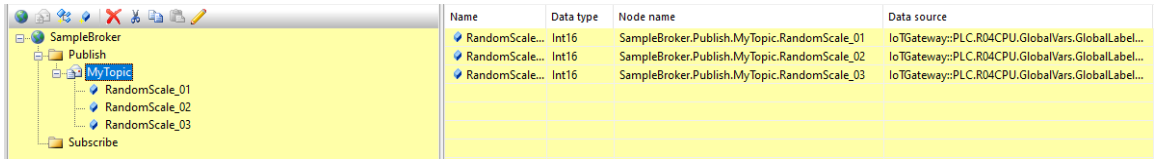
Open “Publish” and click the newly created topic, then right-click on the topic in the “Select variables…” menu



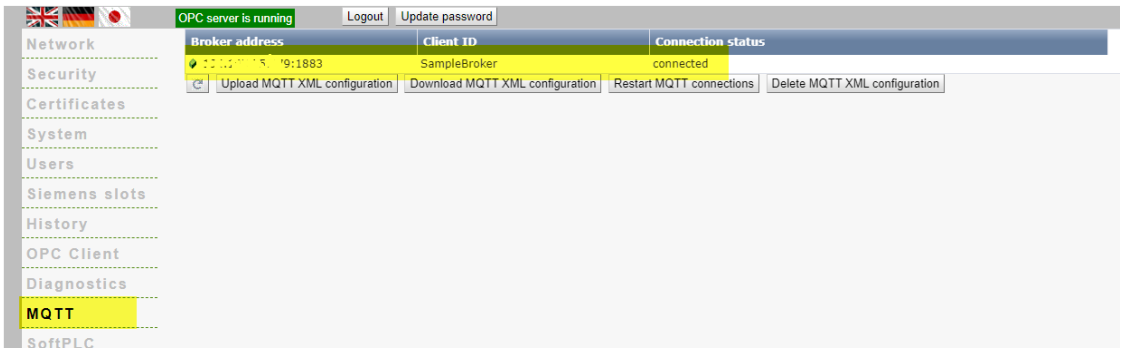
Select the required data in the next window and press “OK”



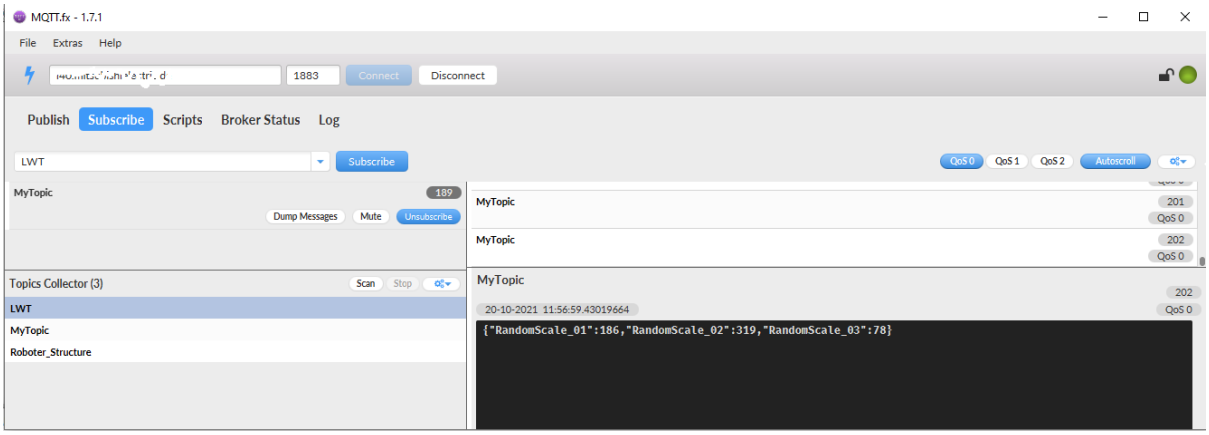
The selected data are then displayed in the active topic in the editor



Send the new setting to the IoT gateway and after the transfer has taken place, check whether the IoT gateway is connected to the broker.



8.2.3 Check data in Broker

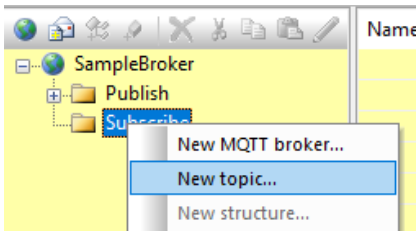


8.3 Subscription to values (receive data from broker)

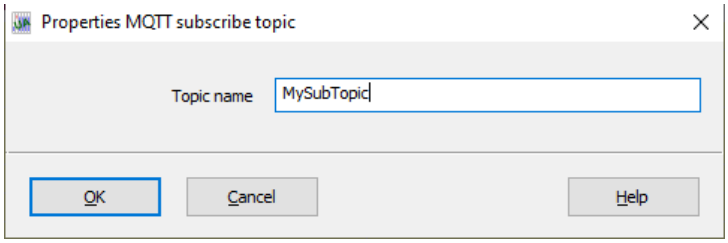
In the following example, the program “MQTT.fx” “publishes” data to a broker. The IoT gateway should "subscribe" this data and write it into the corresponding PLC register in order to reset various counters (Counter01 to Counter03) to "0".

8.3.1 Create Topic

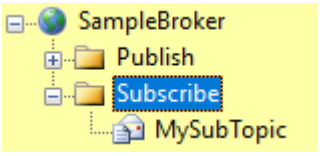
Open SampleBroker ”by pressing, then right-click on“ Subscribe ”and select“ New topic ”.



Enter a name for the new topic and confirm with “OK”.

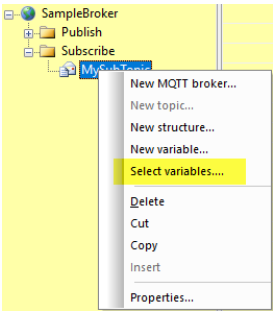


Pressing “⊞” next to “Subscribe” shows the new topic

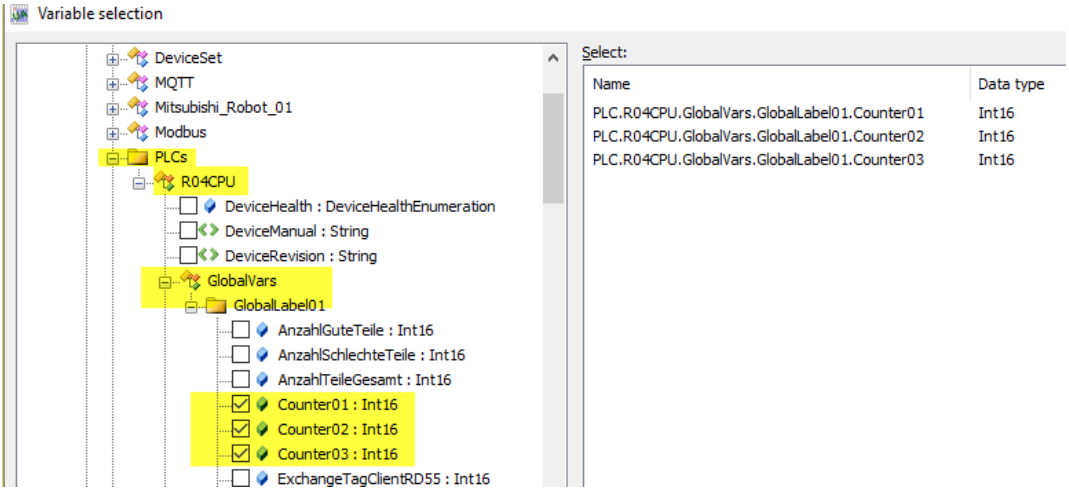


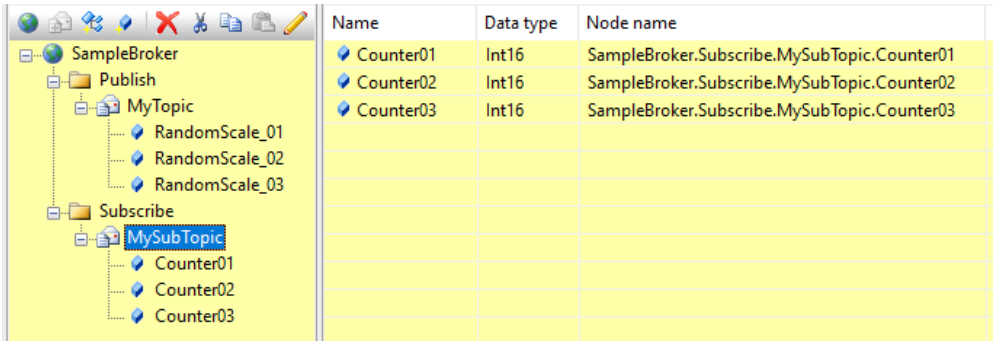
8.3.2 Create new variable

Select the new topic, right-click to open the menu and press “Select variables…”



Select the required data in the next window and confirm with “OK”



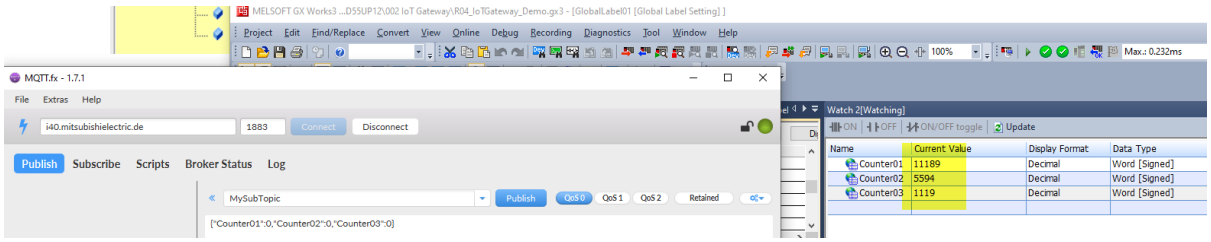


8.3.3 Function test of “Subscribe”

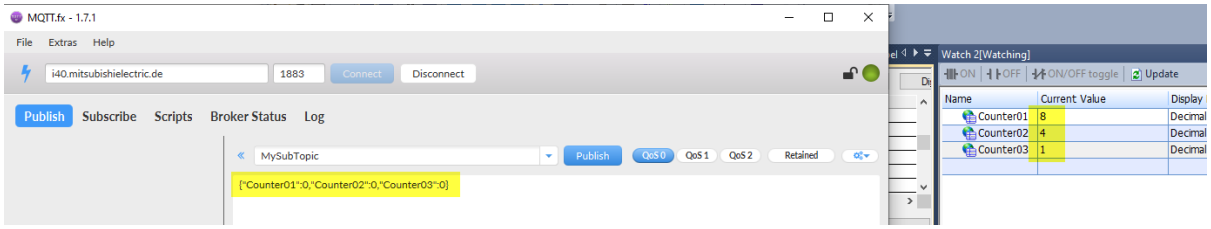
In the Sunscribe topic, the IoT Gateway expects a string in JSON format from the publisher, which must look like this:

{"Counter01":0,"Counter02":0,"Counter03":0}

Befor “Publish” on Broker-side

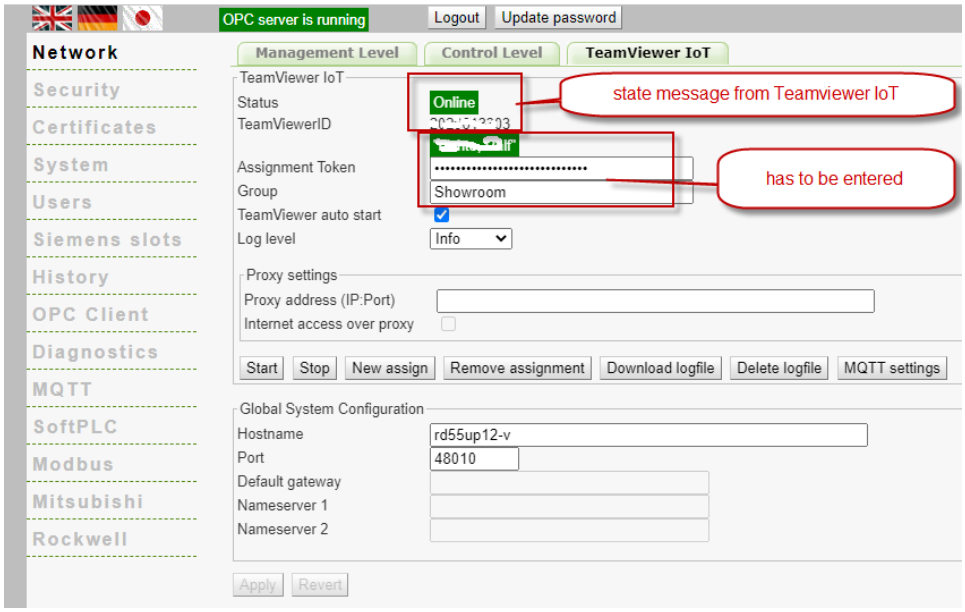


Short after “Publish” on Broker-side

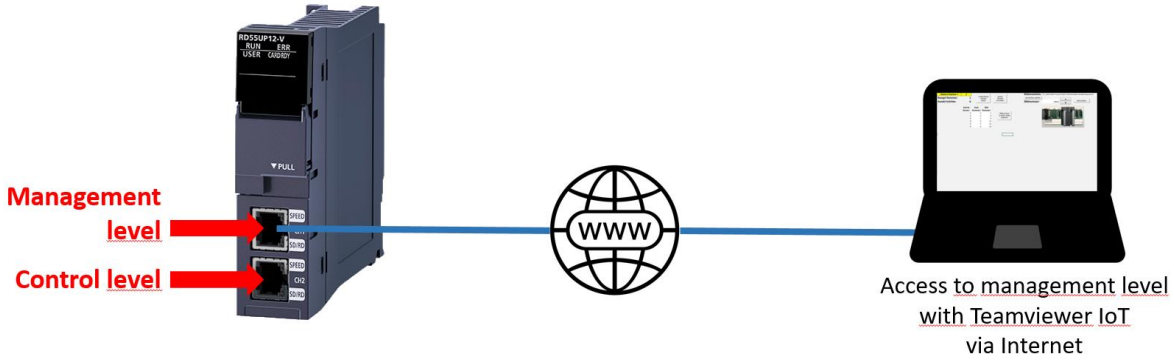


9 Teamviewer IoT Connection

A TeamViewer software is pre-installed in the IoT Gateway from firmware V 7.5. This offers the possibility of accessing almost all Mitsubishi Electric automation components at any time and from anywhere.
Complex modem solutions or the use of a PC on site are a thing of the past.



To establish a connection via ***TeamViewer-IoT,*** the Ethernet subnet of the control level must have access to the Internet.



The RD55UP12-V manages two firewall-separated subnet addresses, each with its own MAC address.

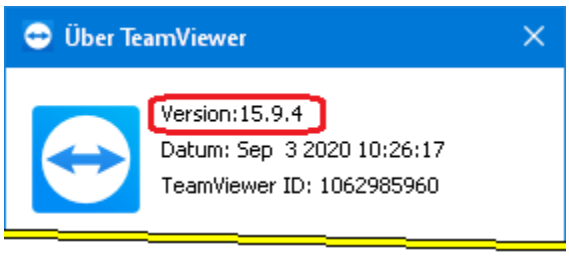
Level	Port	Subnet
Control level	CH 1	Must be different
Process level	CH 2	

9.1 PC preparations - connected to the Ethernet port of the control level

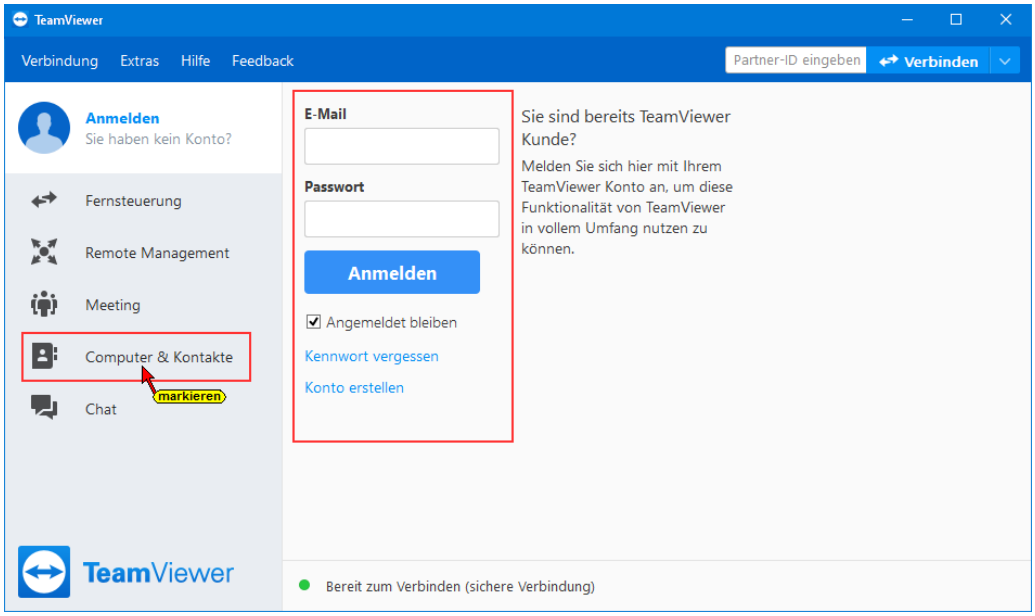
In order to use the access options of the pre-installed TeamViewer software, the conditions mentioned in the following sub-chapters must be fulfilled.

9.1.1 TeamViewer Software

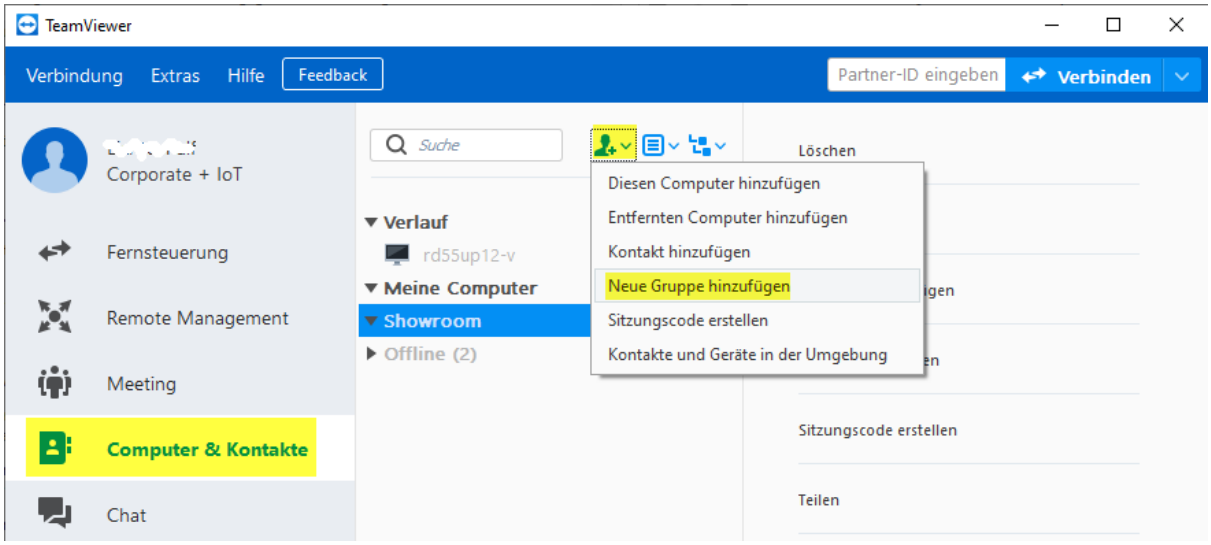
The PC that is to be used to access the external IBH Link UA must have TeamViewer software version 15.9.4 or newer installed.



A TeamViewer account with an appropriate licence must be ready to activate.



If not already present, a new group must be added under Computers and Contacts after logging in.



9.1.2 IBHNet IoT setup

Install the software IBHNet-IoT-Setup.exe on the PC. This software is available under

https://download.ibhsoftec.com/neutral/IBHNet-IoT-Setup.exe

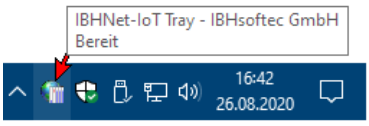
is available for download.

Install and start IBHNet-IoT software

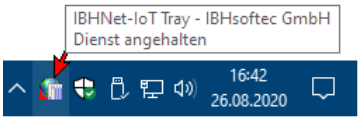
IBHNet-IoT-Setup.exe Execute the downloaded file.

Then run the ***IBHNet-IoT*** programme, which will start the ibhsoftec-agent-service.

The service is displayed as an ***IBHNet-IoT tray*** in the taskbar. If necessary, change the properties of the taskbar to display the icon.



If the icon shows a stopped service, start the service.

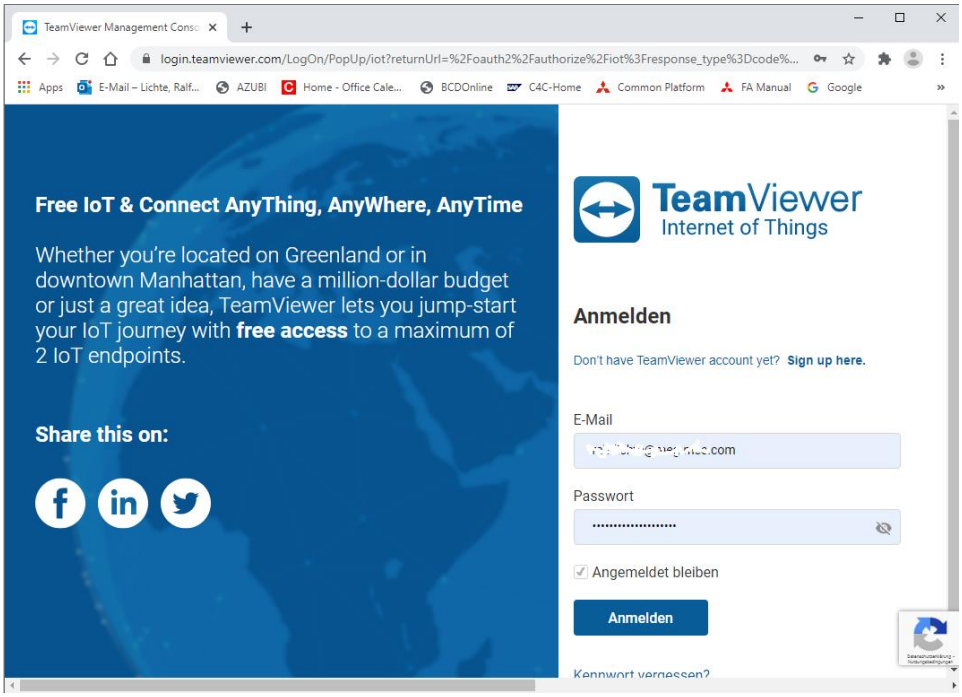


9.1.3 TeamViewer IoT Management Console

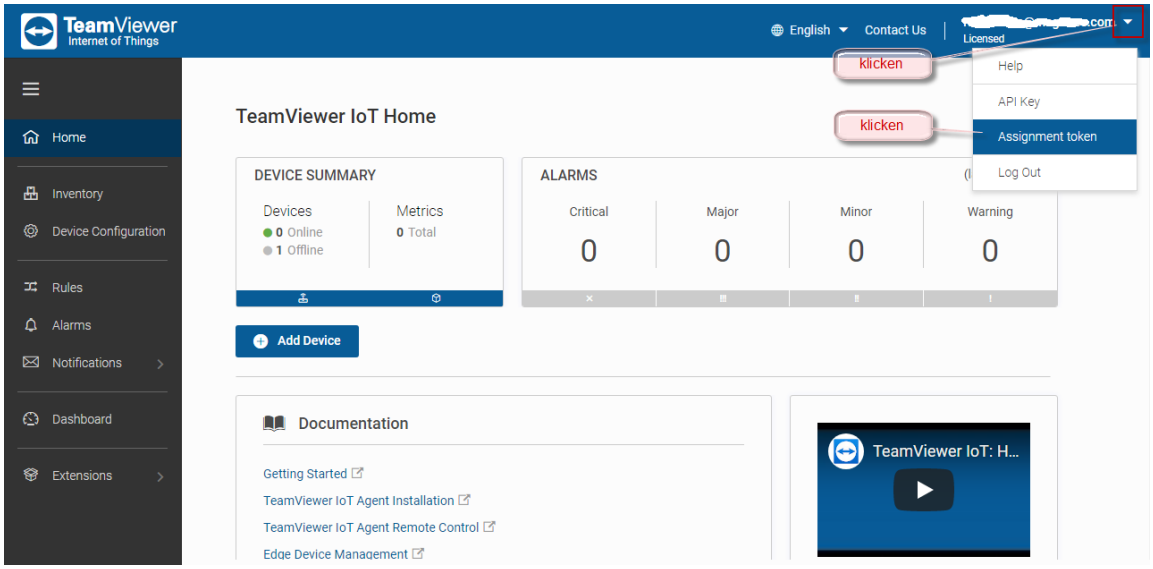
With the link

https://teamviewer-iot.com/en/

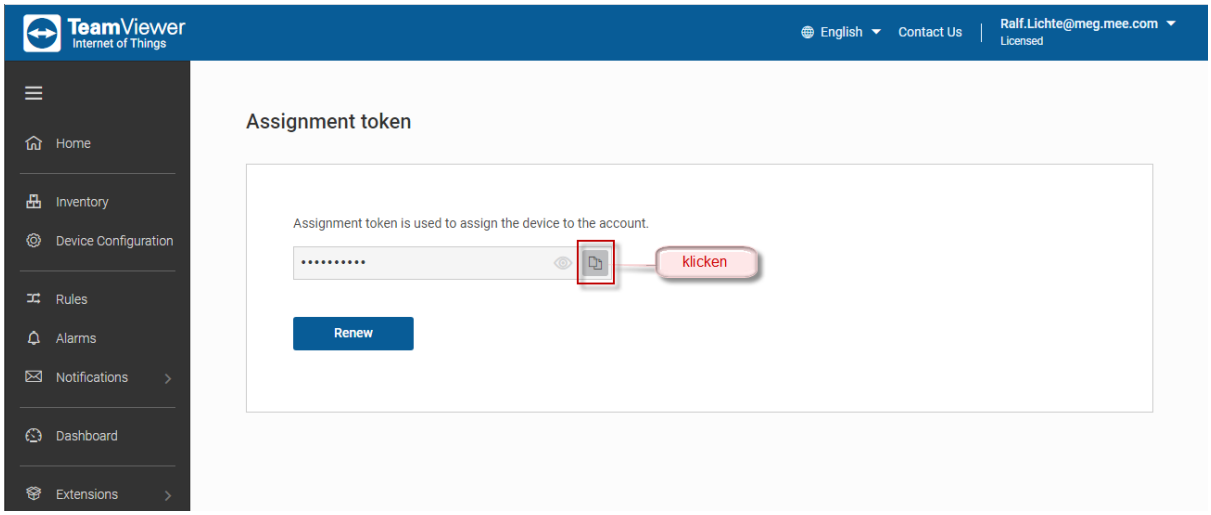
go to the ***TeamViewer Internet of Things*** login page and log in.



After logging into the **TeamViewer IoT Management Console,** open the ***Assignment token*** dialogue box.



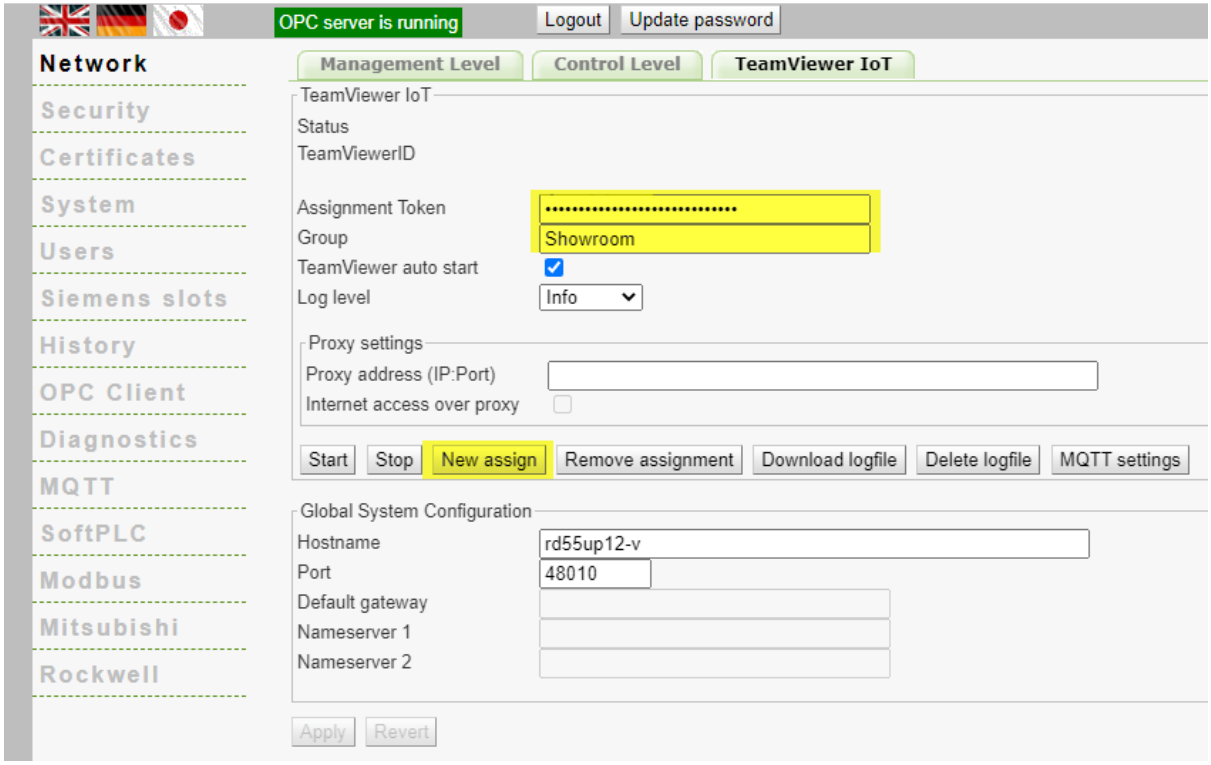
Click on the Copy symbol ***to copy the assignment token to*** the Windows clipboard.



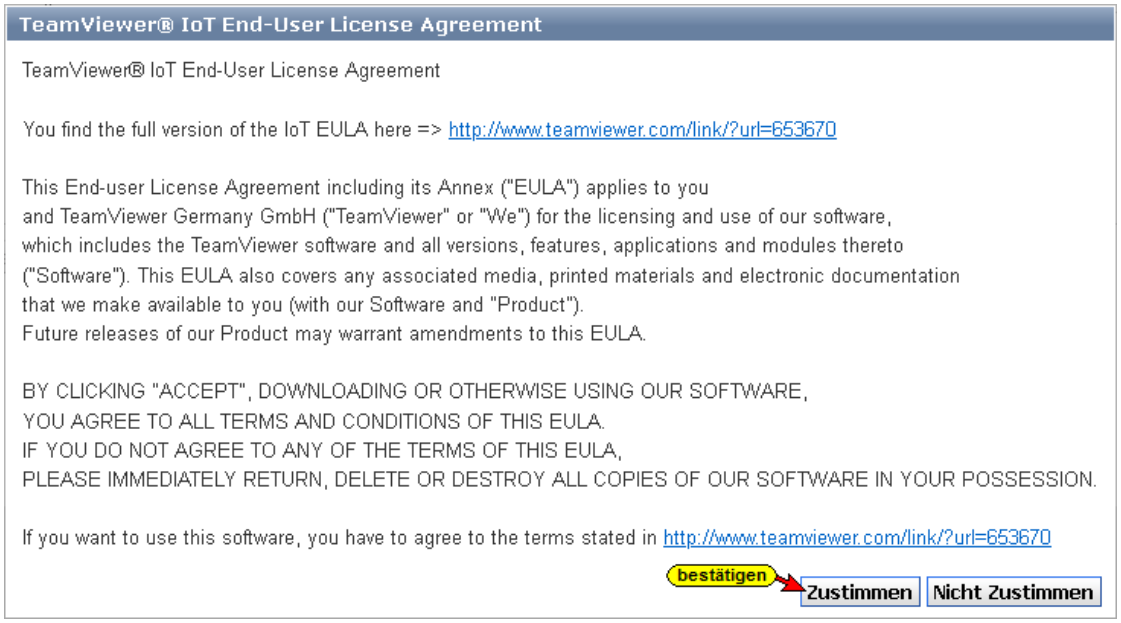
The **assignment token** must then be copied into the field of the same name in the **IBH Link UA** web browser window Network / TeamViewer IoT.

In addition, the group name previously created in TeamViewer must be entered and the check mark for **Automatically switch on TeamViewer** must be set.

Now you can click on **Reassign**

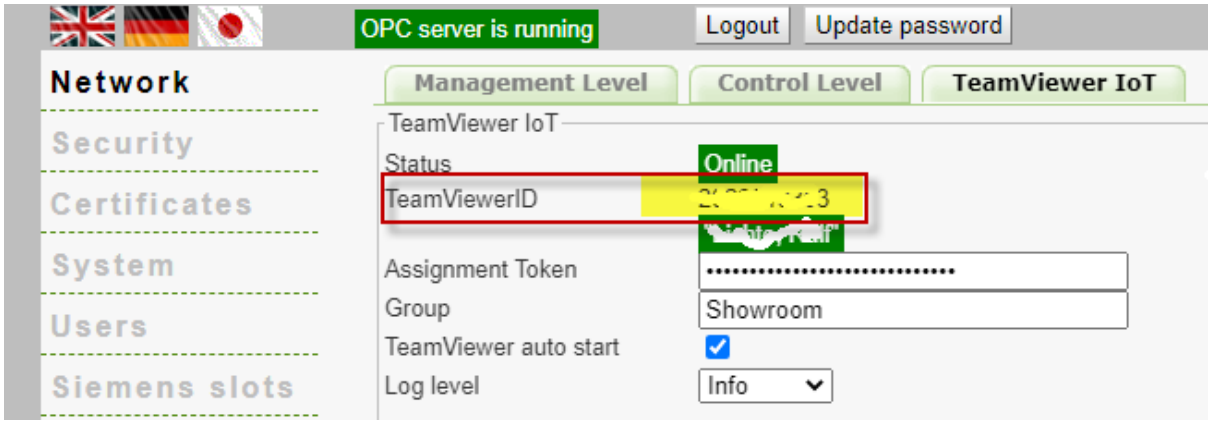


To apply the settings, the **TeamViewer IoT End User Licence Agreement** must be accepted by clicking the **Agree** button.



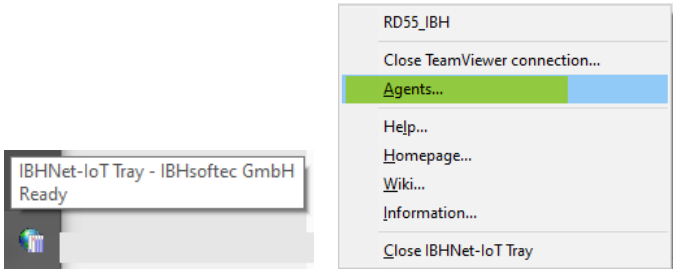
It may take some time for the **assignment token to be** sent to ***TeamViever.***

After successful login, the online connection to the TeamViewer IoT server is established.

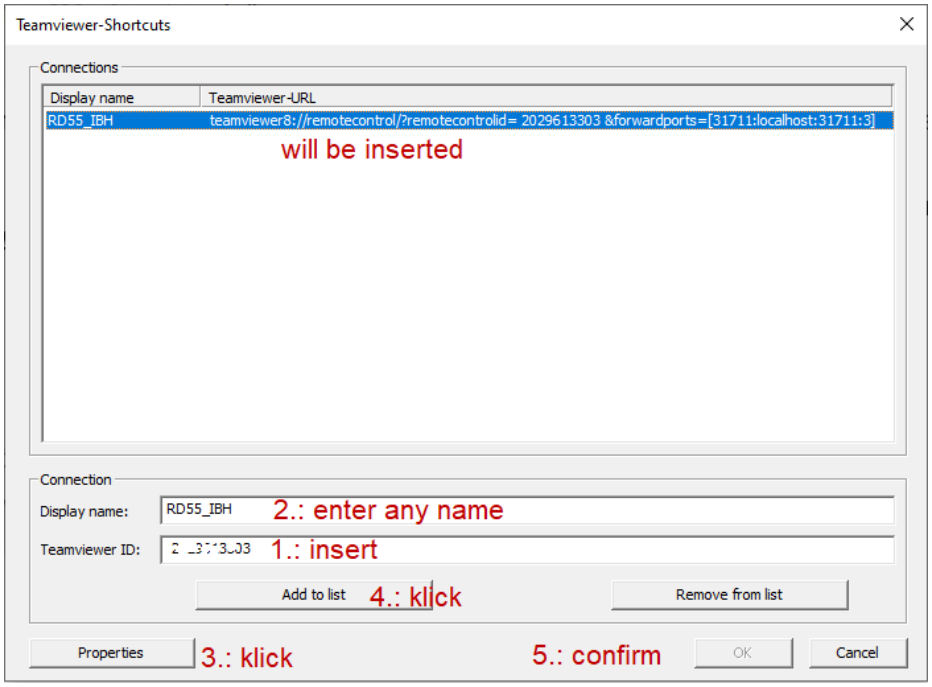


Double-click on the ID now entered by TeamViewer and copy it to the clipboard.

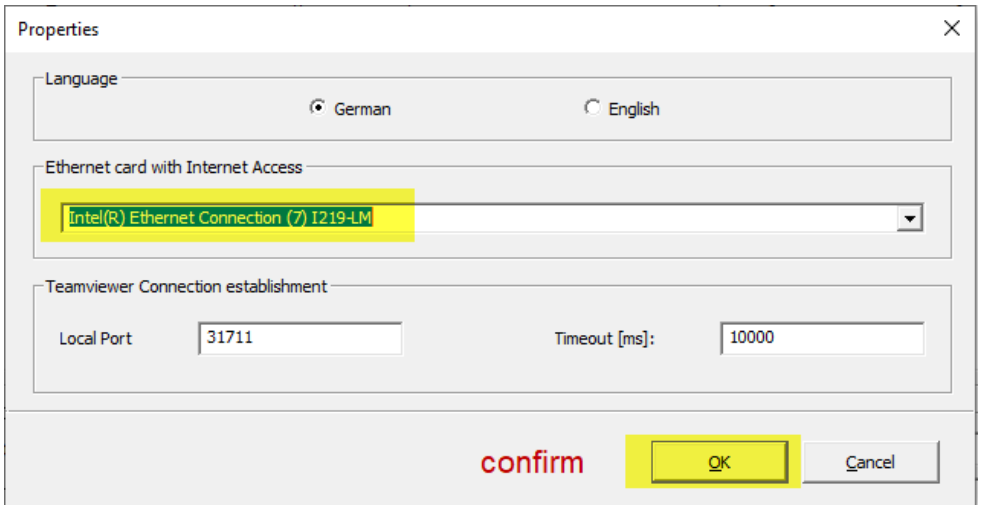
Right-click on the ***IBHNet-IoT Tray*** icon to open the context menu. The command ***Agents...*** opens the **TeamViewer Shortcuts** dialogue box.



Insert the TeamViewer ID number in the field with the same name. The display name is transferred to the ***TeamViewer account.*** This name can be used to establish a connection to the IBH Link UA via the internet.



Clicking the ***Properties*** button opens a dialogue box with the specification of the network card via which the IoT Gateway is connected.

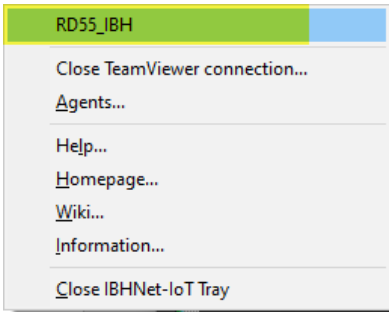


Click on the ***Add to list*** button to adopt the display name and the TeamViewer ID. Click ***OK*** to close the dialogue box.
The installation of **TeamViewer IoT** in the IoT Gateway is hereby completed.

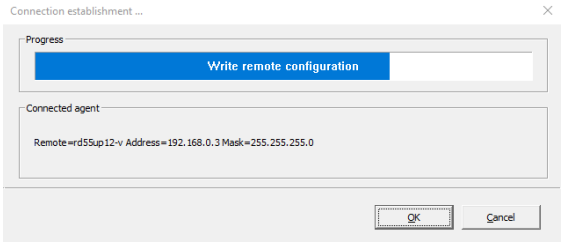
9.1.4 Connection setup

From any PC, a connection can be established via the Internet to the IoT Gateway and thus to the PLC controllers and other devices connected to the control level ports.

The ***IBHNet-IoT software*** must be installed on this PC, TeamViewer must be started and the login to the ***TeamViewer account*** must have taken place.
Right-click on the ***IBHNet-IoT Tray*** icon to open the context menu. The devices registered with the ***TeamViewer account*** are listed in the upper section of the context menu. Click on the desired device (RD55_IBH) to establish the connection.

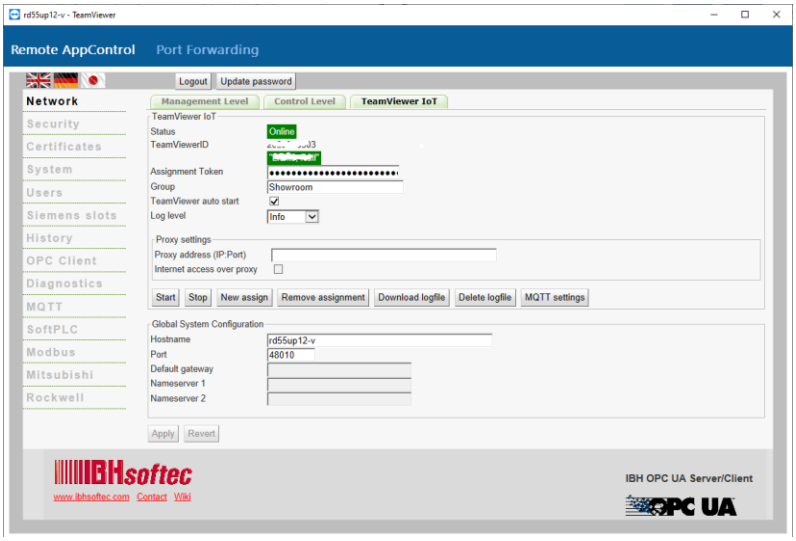


The establishment of the connection is displayed.



After the connection has been successfully established, the ***IBH Link IoT*** appears as a ***WireGuard tunnel*** under the network adapters of the PC.

From now on, all controllers and devices that are connected via the ***IBH Link UA*** control level are accessible.



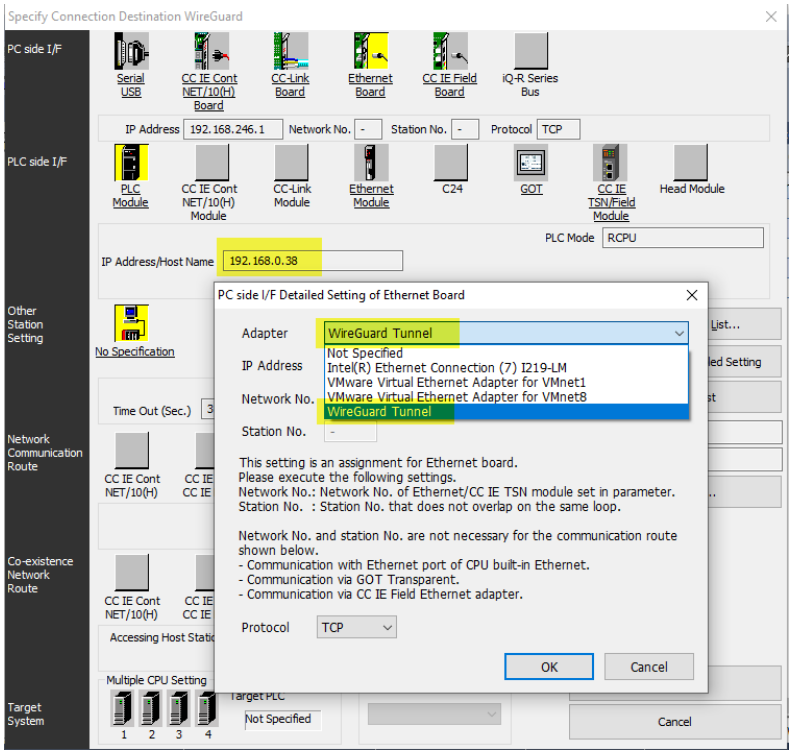
During the TeamViewer start-up process, it can happen that no connection is established and error messages are displayed. These error messages are to be closed with Cancel. The start process must be restarted.

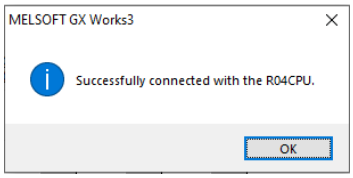
After the connection has been successfully established, the ***IBHNet-IoT tray*** icon in the taskbar changes. It gets an additional green marking.

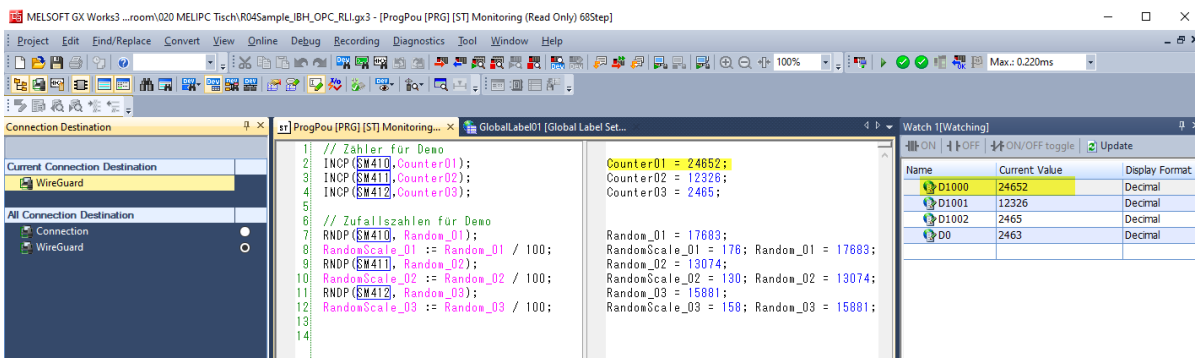
From the PC whose ***IBHNet-IoT tray*** icon displays the existing online connection, controllers (CPUs / devices) connected to the control level ports can be accessed with the corresponding software (programming system).

9.2 Access to controllers (CPUs / devices) that are connected to the ports of the control level.

9.2.1 iQ-R CPU



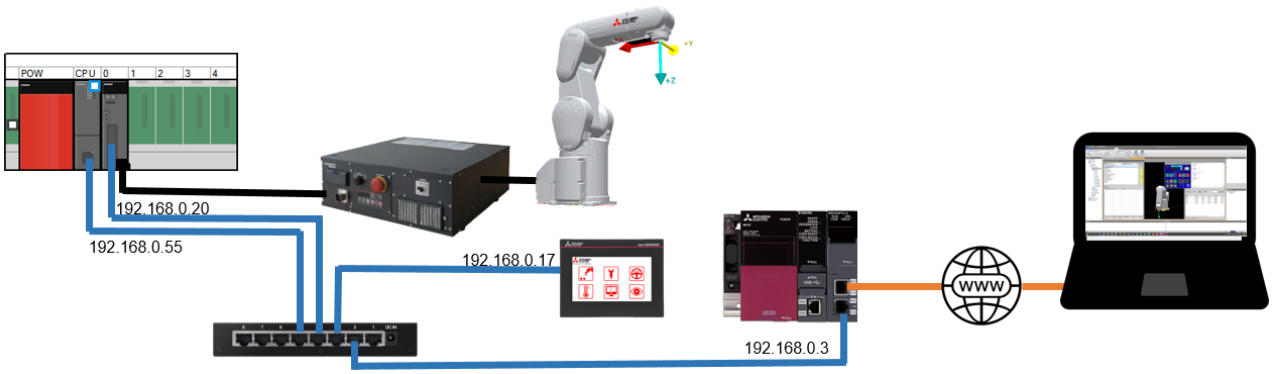




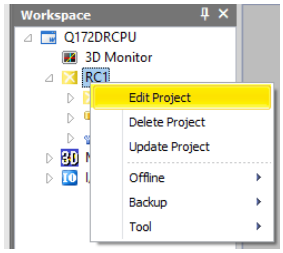
9.2.2 Robot

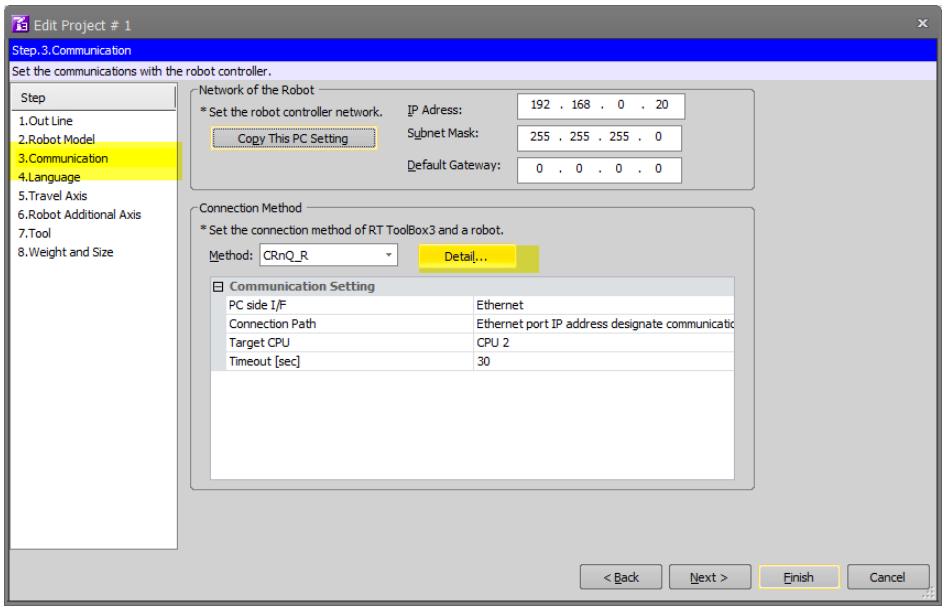
Example: Connection with Q172DRCPU

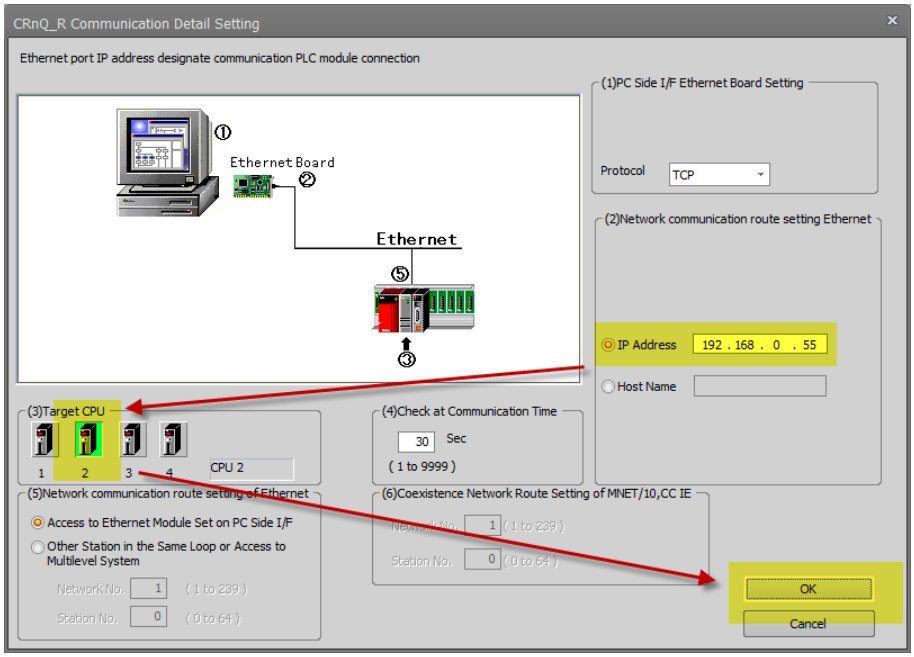
Hardware structure:

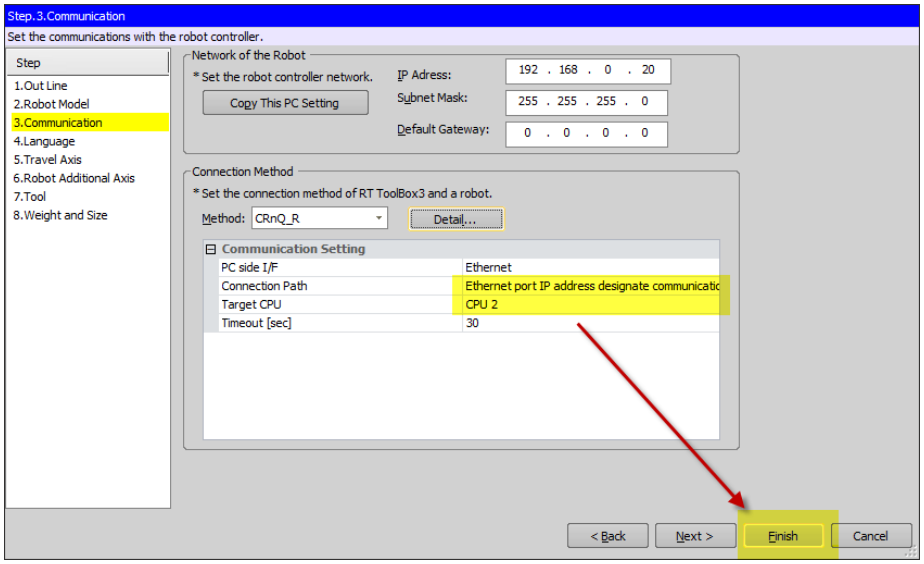


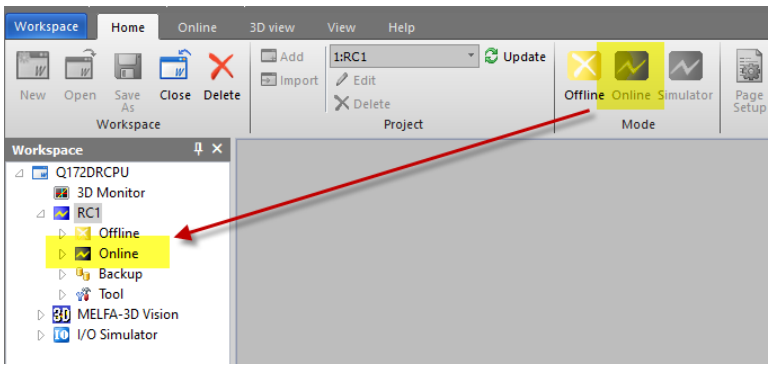
Communication setting in the RT Toolbox 3:






CRnQ_R Communication Detail Setting
Ethernet port IP address designate communication PLC module connection
(1)PC Side I/F Ethernet Board Setting
Protocol
TCP
Ethernet Board
Ethernet
(2)Network communication route setting Ethernet
IP Address
192 . 168 . 0 . 55
Host Name
(3)Target CPU
CPU 2
(4)Check at Communication Time
30
Sec
(1 to 9999)
(5)Network communication route setting of Ethernet
Access to Ethernet Module Set on PC Side I/F
Other Station in the Same Loop or Access to Multilevel System
Network No.
Station No.
(6)Coexistence Network Route Setting of MNET/10,CC IE
OK
Cancel


Step.3.Communication
Set the communications with the robot controller.
Step
1.Out Line
2.Robot Model
3.Communication
4.Language
5.Travel Axis
6.Robot Additional Axis
7.Tool
8.Weight and Size
Network of the Robot
* Set the robot controller network.
Copy This PC Setting
IP Adress:
192 . 168 . 0 . 20
Subnet Mask:
255 . 255 . 255 . 0
Default Gateway:
0 . 0 . 0 . 0
Connection Method
* Set the connection method of RT ToolBox3 and a robot.
Method:
CRnQ_R
Detail...
Communication Setting
PC side I/F
Ethernet
Connection Path
Ethernet port IP address designate communicatic
Target CPU
CPU 2
Timeout [sec]
30
< Back
Next >
Finish
Cancel


Workspace
Home
Online
3D view
View
Help
New
Open
Save As
Close
Delete
Workspace
Add
Import
1:RC1
Update
Edit
Delete
Project
Offline
Online
Simulator
Page Setup
Mode
Workspace
Q172DRCPU
3D Monitor
RC1
Offline
Online
Backup
Tool
MELFA-3D Vision
I/O Simulator

9.2.3 GOT

In the GT-Designer3, it is not necessary to select an IoT tunnel if one exists; entering the destination IP address of the GOT is sufficient: