OPC UA Server

1. General

The iQunet UNIX Server runs an embedded OPC UA (unified architecture) Server. OPC UA extends the standard, and highly successful, OPC communication protocol. It enables data acquisition and allows that data to be modelled and communicated between the plant floor and the enterprise reliably and securely. OPC UA is future-ready, easy to configure, and maintain.

2. How to access the OPC UA Server?

2.1. OPC UA Server address

All recorded data can be extracted via the built-in OPC UA Server. The OPC UA Server always listens on port 4840 regardless if the connection is made via cable, hotspot or WiFi.

If you use the hotspot connection, this will be 192.168.42.1:4840.

If you use another connection type, you need to use the IP address of the iQunet Unix server (xxx.xxx.xx.xx:4840). The server's IP address is handed out by your DHCP server. The recommended way of operation is to setup your DHCP server to provide a static lease. The current IP address of the server in the network can easily been found by clicking the 3 bars below the iQunet logo in the sensor dashboard. Select "Ethernet-802.3" in the left pane and the IP address will appear.

It is also possible to set a static IP address in the "Ethernet-802.3" panel. The server will then start a virtual network interface and will operate from 2 simultaneous IP addresses (the static IP address and the regular DHCP lease). You can then either use the static IP address or the DHCP address for the connection to the OPC UA Server.

2.2 OPC UA Server monitoring

After a power restart of the iQunet system it will take a few minutes to allow the server to startup and populate the address space from the database. Thereafter an internal process monitor becomes active which surveys the internal daemons every few seconds. When a problem is detected, a cleanup of the affected processes is performed. This cleanup is reported to the supervisor which can then restart the faulty component. The OPC UA Server is one of the 11 monitored subcomponents.

The supervisor has a local frontend running on port 9001 (http://xxx.xxx.xx.xx:9001 where xxx.xxx.xx.xx is the server's IP address). Username and password are 'admin'. Please note that only the highest-level master processes are reported here.

2.3 OPC UA variable nodes addressing

There are 3 ways to access variables in OPC:

- via string ID's,
- via numeric node-ID's
- and via browsing.

All 3 options will be explained in further detail below. Which variable nodes are available depends on the type of sensor since they are added dynamically to the address space. Due to the collision of variable names when multiple sensors are present in the OPC UA server, we only support browsing and numeric node-ID's.

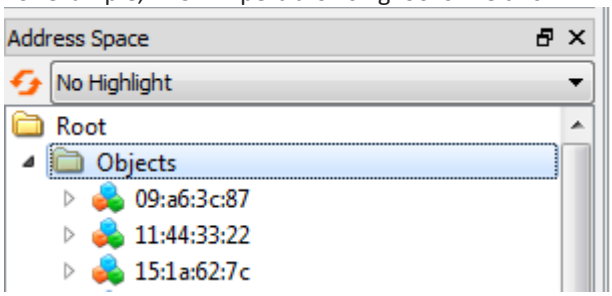
String ID's

The (deprecated) OPC DA way to access variables is through string ID's, where the variable path is encoded into a string with dot separators like for example "ns=2;s=my_sensor.tree.encoded.variablename". We don't support this old style of accessing any more.

Numeric node-ID's
An alternative way is direct access via the UINT32 numeric node identifier like for example "ns=2;i='12345678'". In practice only 31 bits are used to prevent overflow errors in some client software systems. The exact representation depends on your client software. The client extracts the namespace and 32-bit ID and sends this to the OPC server.

Browsing
The preferred way of accessing variables is via browsing. iQunet's OPC UA server supports browsing which means that your client software can browse the address space as follows:
[get node 'root' (ns=0)] -> [Objects (ns=0)]-> [get sensor children with ns=2] -> [get Sensor 'ab:cd:de:ef'] -> [get Variable 'mmsRmsX']

For example, in UA Expert browsing looks like this:

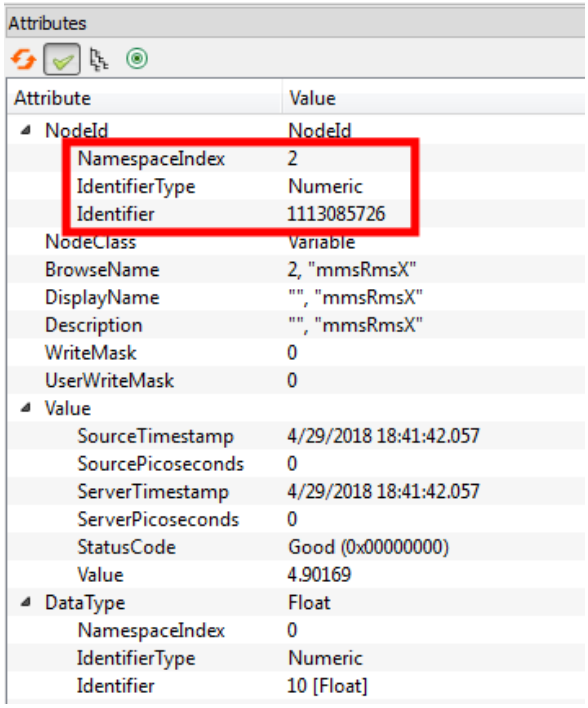


The server will send back an ua-node struct object containing the numeric node-ID (as described above). Your client can cache and use this numeric node-ID for fast direct access in all future requests in the same session.

In the figure below, you can see that our server supports node identifiers of the type "Numeric".

The node below can thus be accessed as follows:

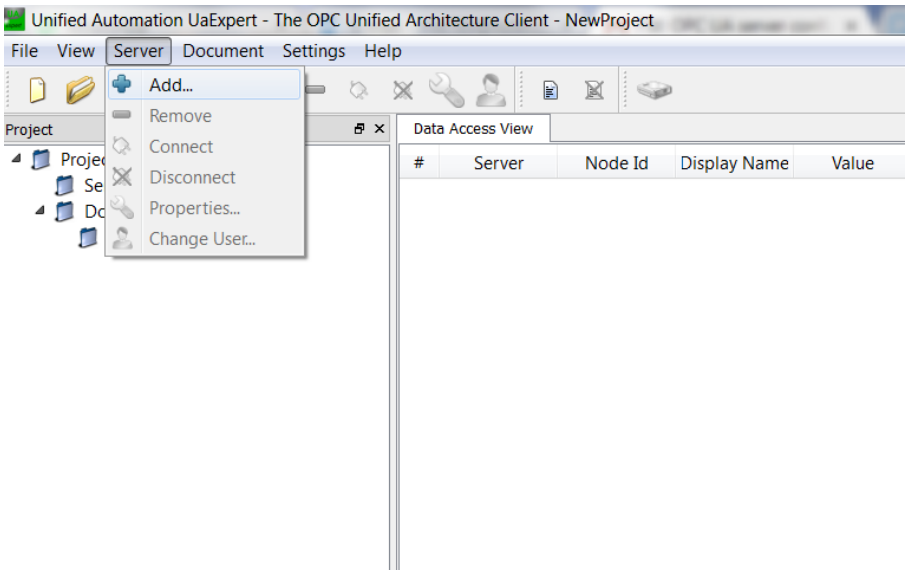
- via browsing: Root->Objects->'15:1a:62:7c'->'mmRmsX'
- or via direct access: ns=2;i=1113085726.



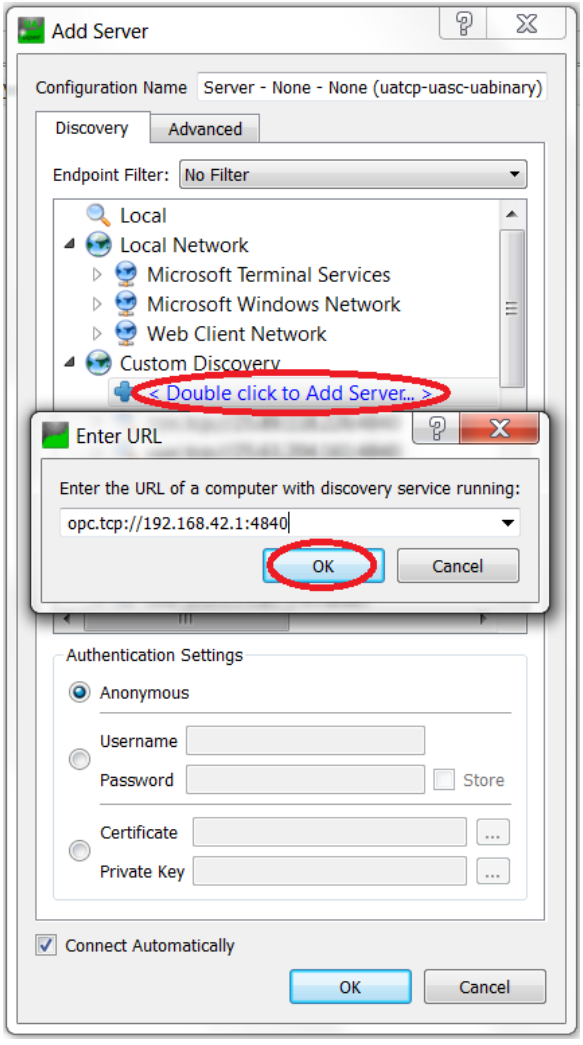
The OPC UA standard requires this numeric node-ID to remain static only for 1 single server session. The client software must then re-request the node-ID mapping for each session. Our server will however remap the same ID's to the same endpoints to support client software with incorrect caching.

2.4 Example using UA Expert

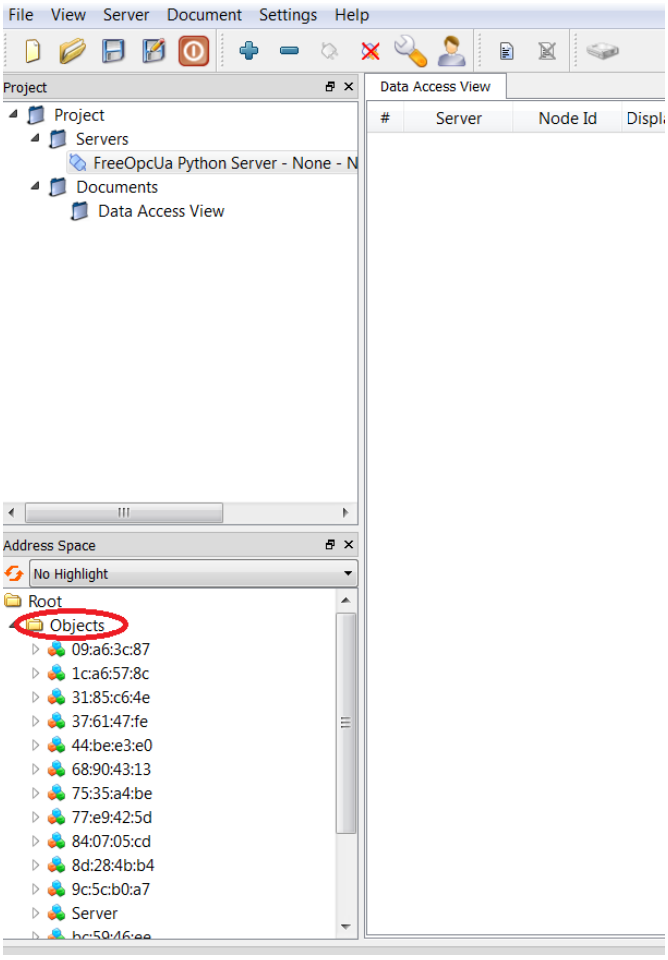
To extract data via OPC, you can use UAExpert for example.
Open UA Expert and click on Server → Add.



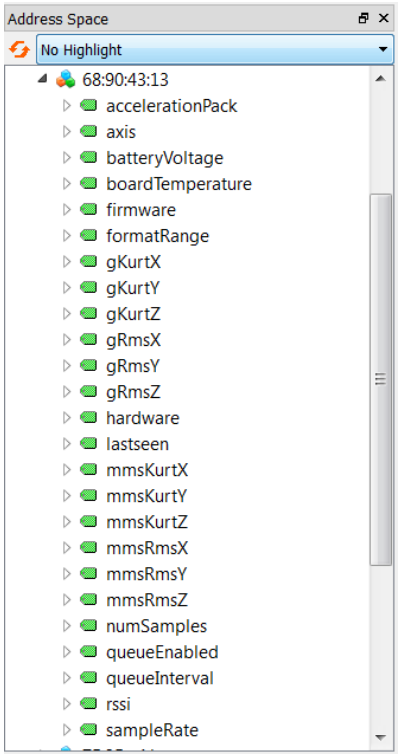
Double click on "Double click to Add Server" and fill out the IP address behind opc.tcp://. Click OK.



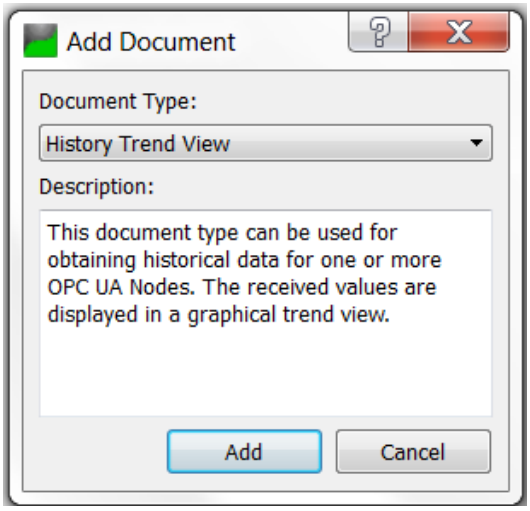
Select the added server in the server list. All sensors connected to this server will appear in the Address Space.



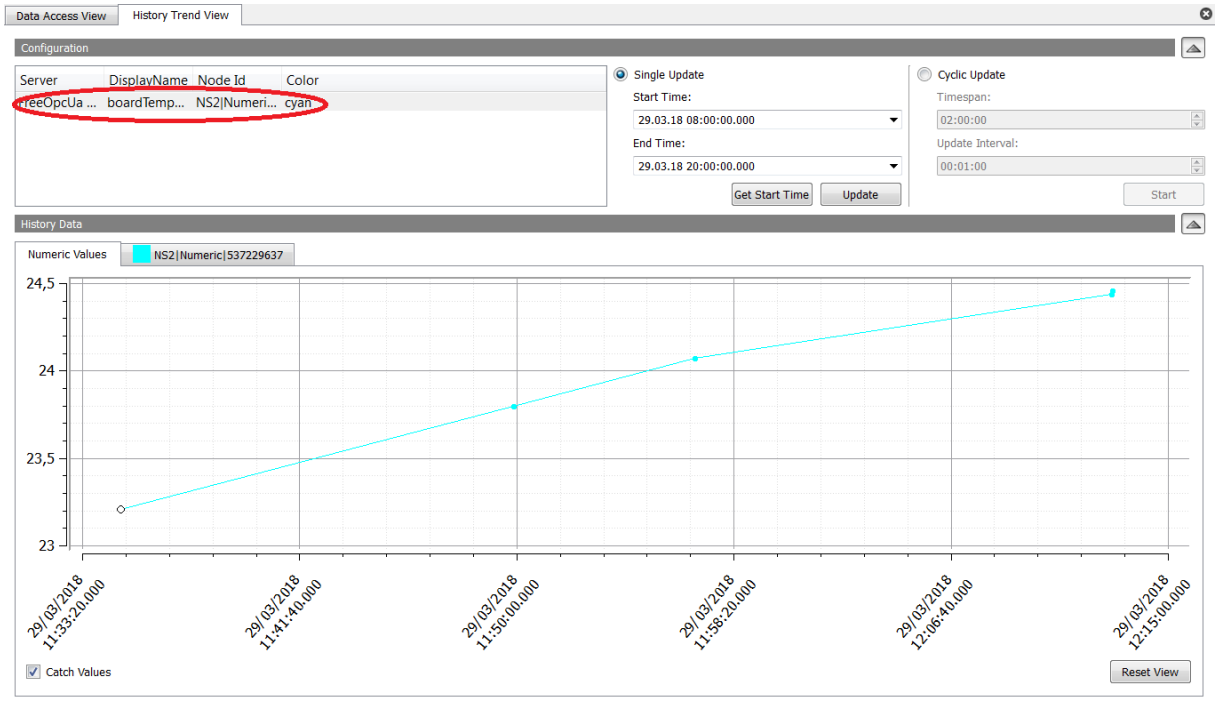
Click on the macId of the sensor to see all possible attributes of the sensor.



Add a document to inspect for example the board temperature data (Document → Add). Select 'History Trend View' as the document type and click 'Add'.



Drag the boardTemperature attribute of the sensor to the configuration window.



Temperature read-out is possible via either a single update that extracts all data values in between two points of time at once or via a cyclic update that extracts all data over the set timespan every set time interval (update interval).

The accelerationPack attribute contains the raw vibration data. The accelerationPack format is as follows:

1/ numSamples: n = #samples
2/ accelArray: rawSample[0:n-1]
3/ sampleRate: e.g. 400 = 400Hz
4/ formatRange: e.g. 4 = +/-4g (hardware setting of the accelerometer IC)
5/ offset: unused, 0 (hardware offset of the accelerometer IC)
6/ encoded_axis: X = 0, Y = 1, Z = 2
7/ prescaler: unused (only used when no compression in debug mode)

8/ compression: unused (0 = no compression in debug mode, 1 = compression)
You will see that the first 7 samples of the accelArray (at the start of each measurement) show a transient response due to the start-up behavior of the compression algorithm. Since a Hanning window is used for the calculation of the DFT and RMS, this behavior will be automatically suppressed and has thus no further impact.

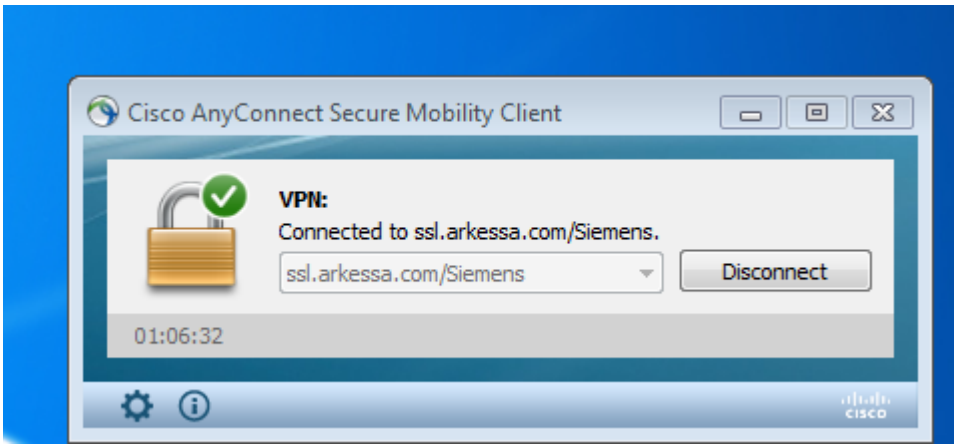
The conversion of the accelArray to g units is as follows:

Conversion of rawSample[0:n-1] to [g]:

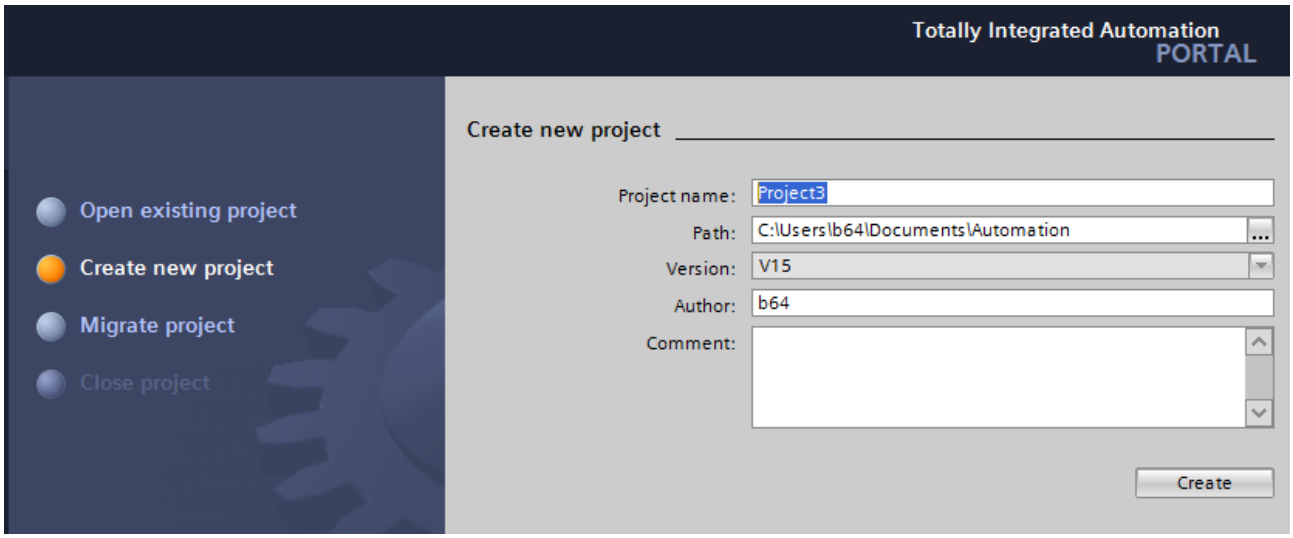
gSample = rawSample[0:n-1]/512.0*formatRange [g]
gTimes = [0:n-1]/sampleRate [sec]

2.5 Example using TIA Portal V15 WinCC RT Advanced

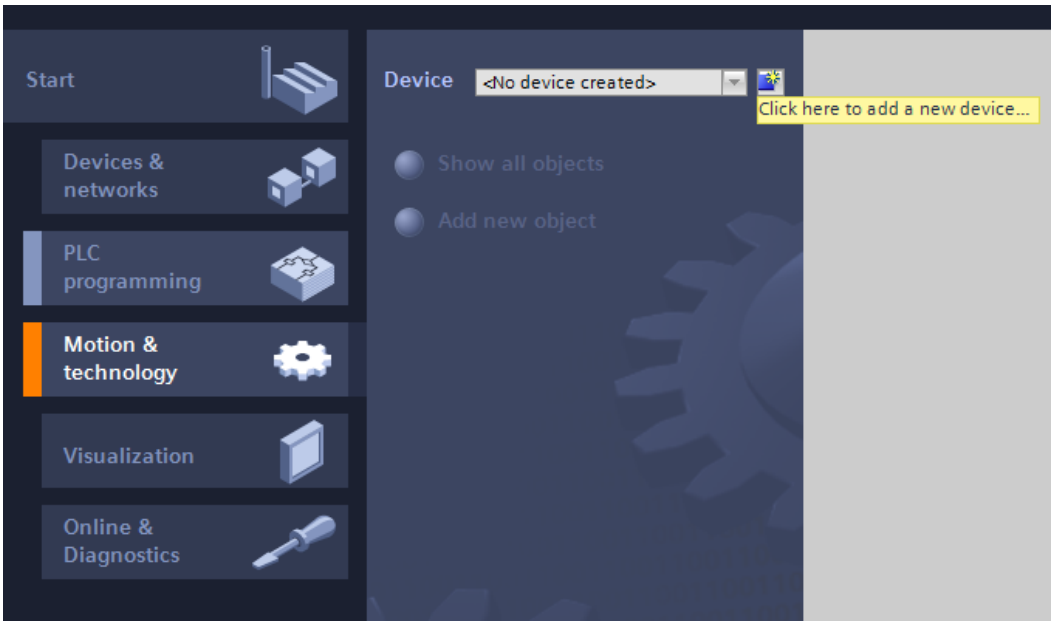
Connect to the iQunet server. In this example, the connection is set-up via VPN with IP address 10.50.29.1. Note that other connection methods can be used as well. See the support page on the iQunet website (FAQ) for more information.



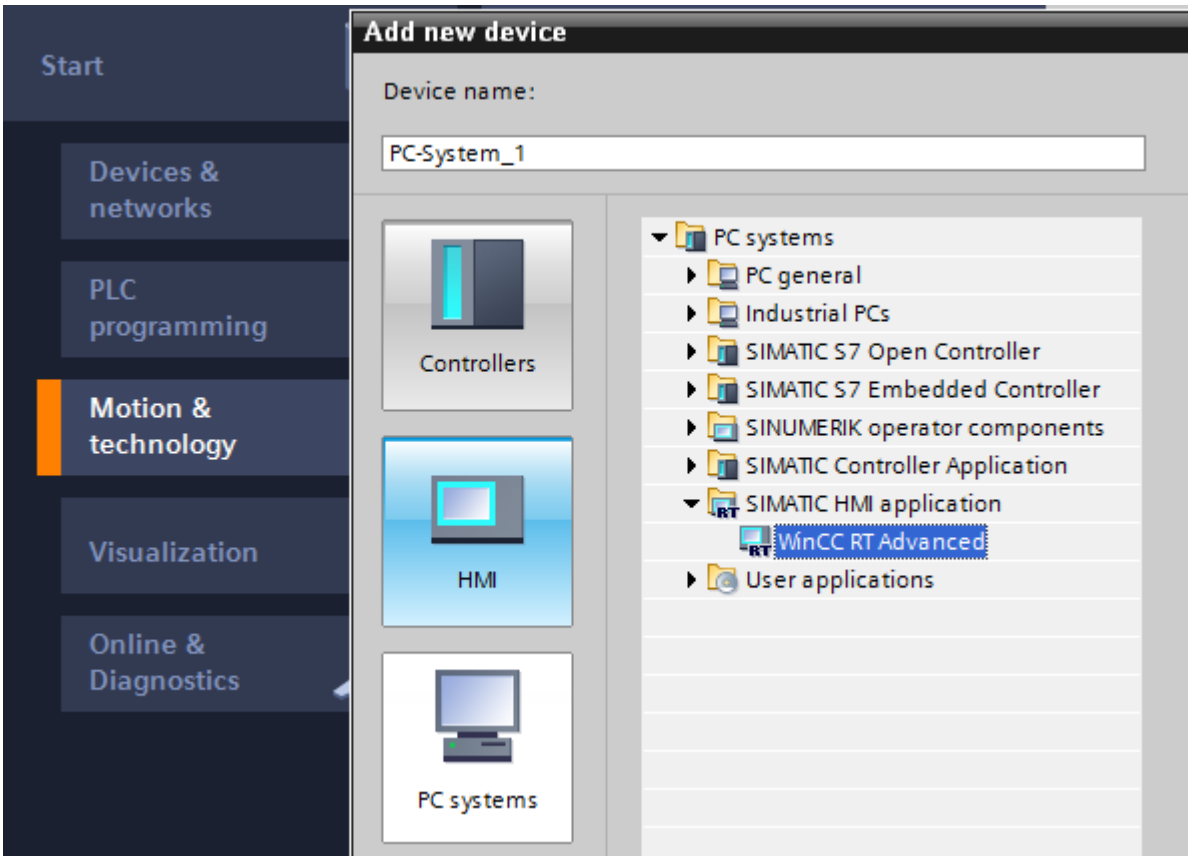
Open TIA Portal V15 and select “Create new project”.



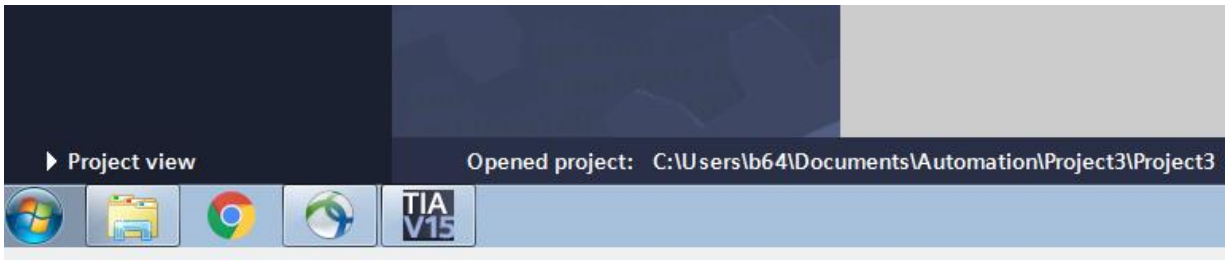
Select “Motion & technology” and click on the button on the right to add a new device.



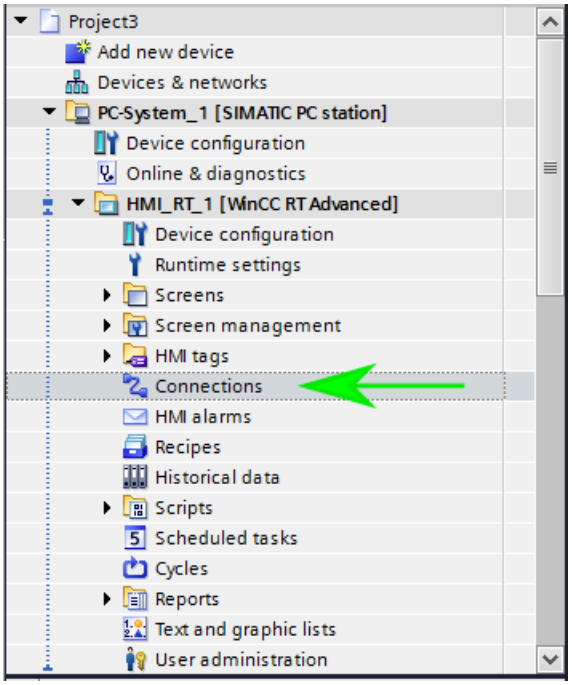
Select PC systems → SIMATIC HMI application → WinCC RT Advanced and click OK.



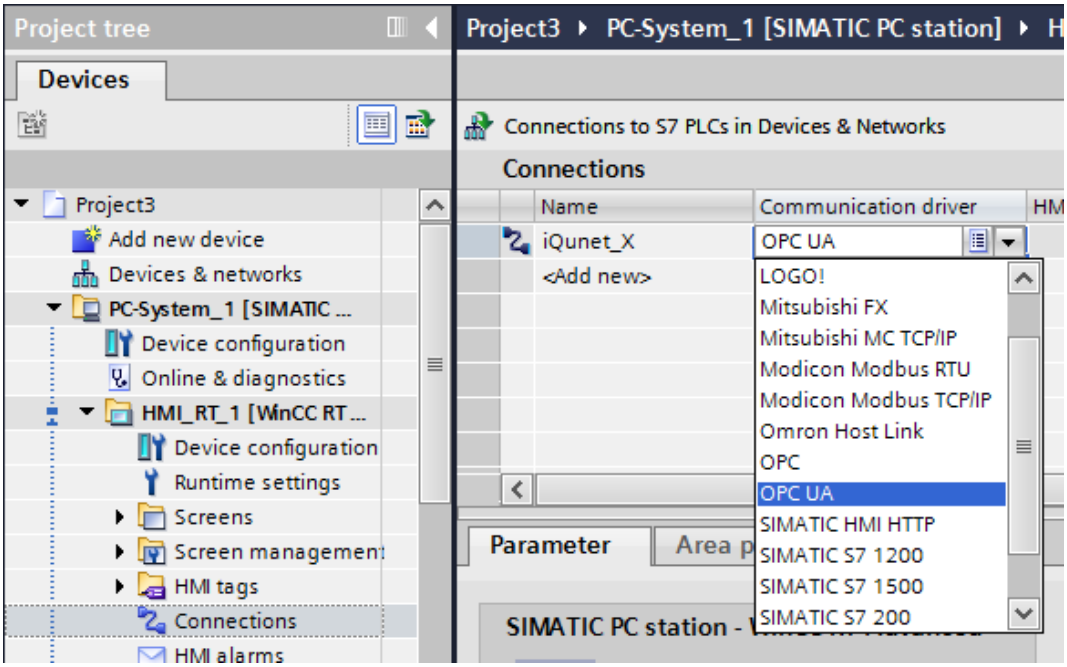
Open the project view.



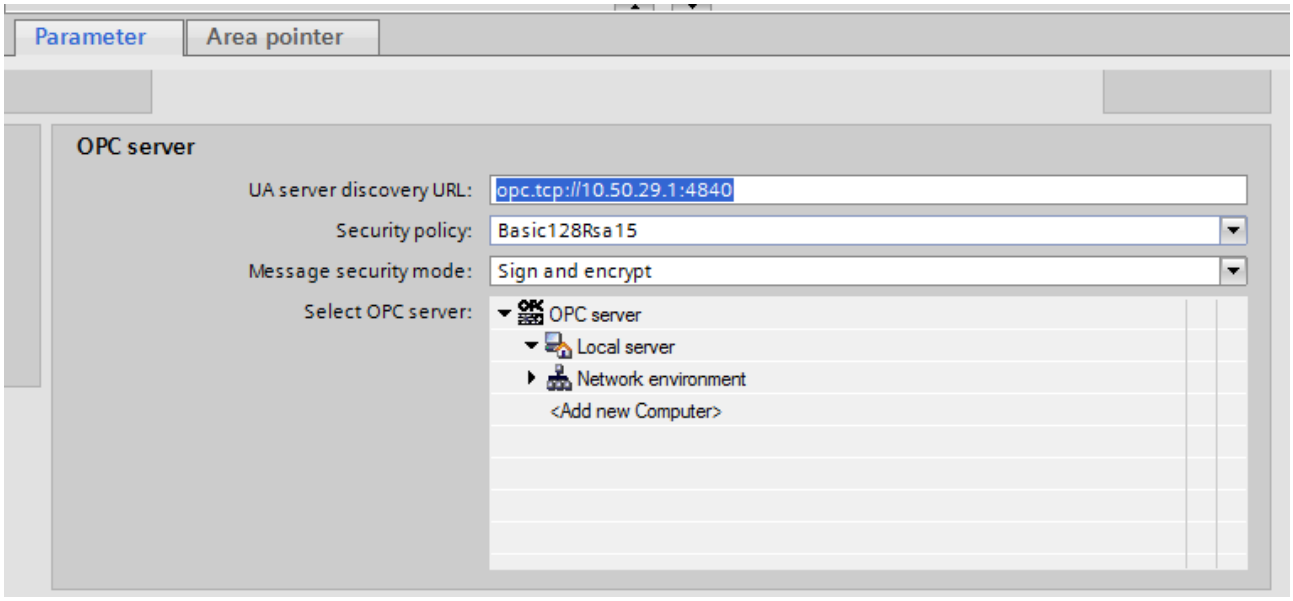
Select PC-System_1 → HMI_RT_1 → Connections.



Select “Add new” and add a new connection. Select “OPC UA” as the communication driver.

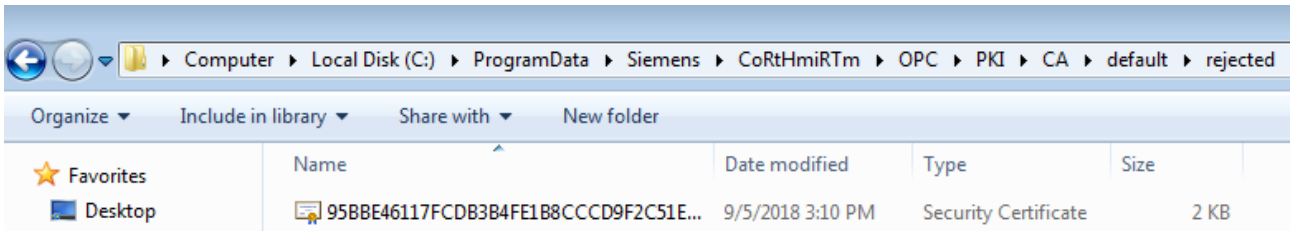


Fill out the UA server discovery URL and the security settings.
url = opc.tcp:// 10.50.29.1:4840
policy = Basic128Rsa15
mode = Sign & encrypt



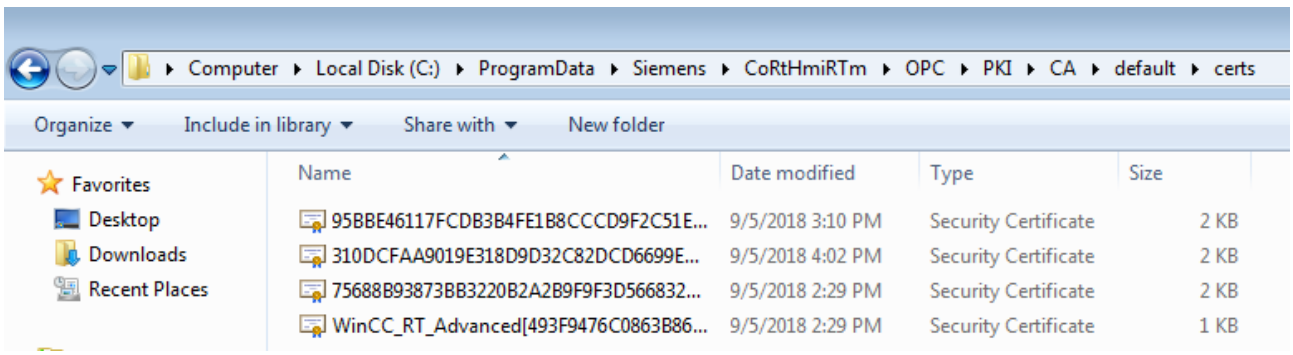
Accept the server certificates. Move the iQunet server certificate from the 'rejected' folder to the 'certs' folder. The procedure is explained in the following link:
https://support.industry.siemens.com/cs/attachments/63481236/63481236_Part5_RT_Advanced_Server_und_Panel_Client_en.pdf.

Move the certificate from C:\ProgramData\Siemens\CoRtHmiRTm\OPC\PKI\CA\default\rejected

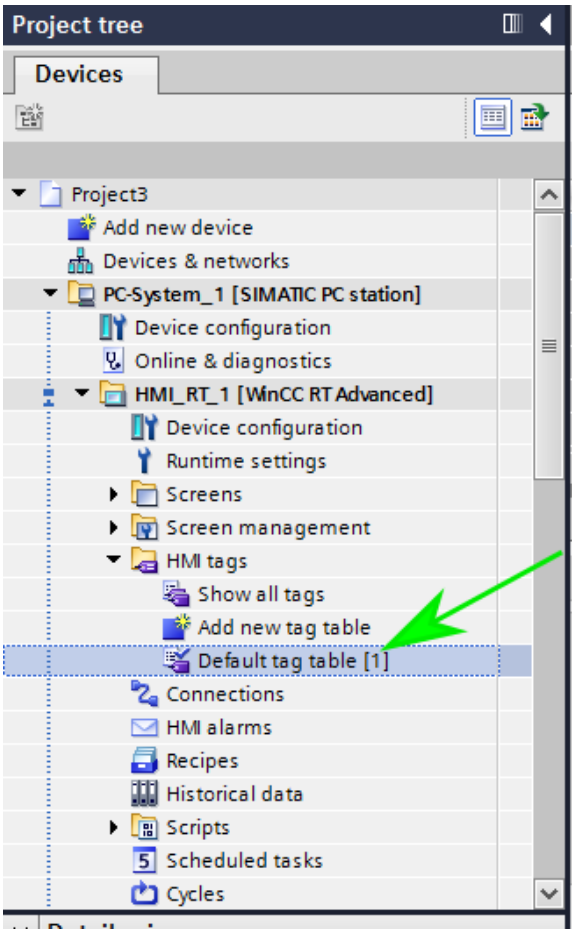


to

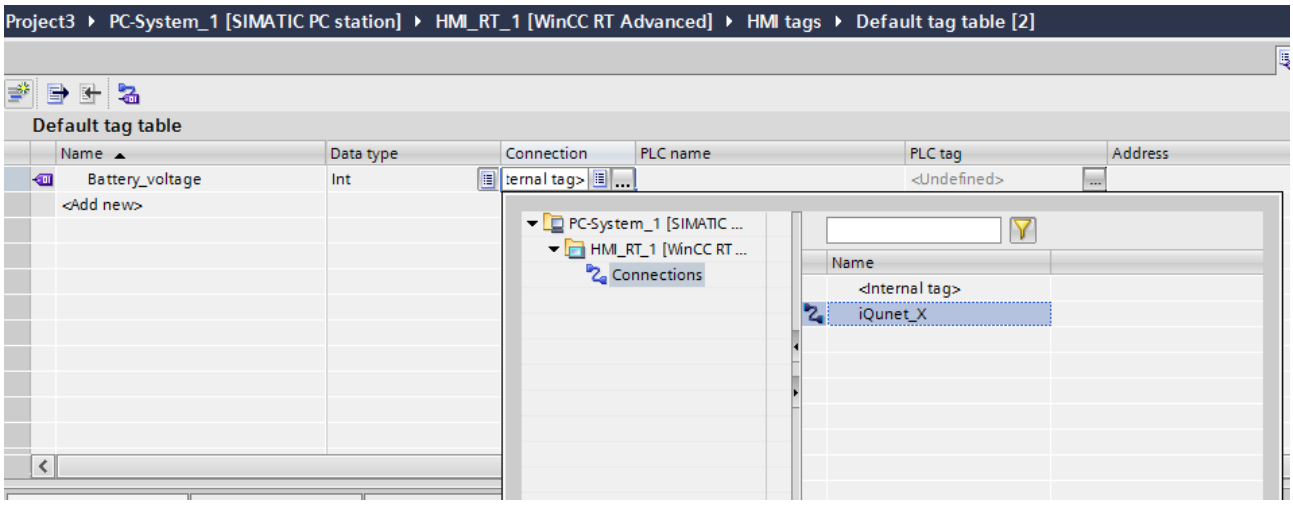
C:\ProgramData\Siemens\CoRtHmiRTm\OPC\PKI\CA\default\certs.



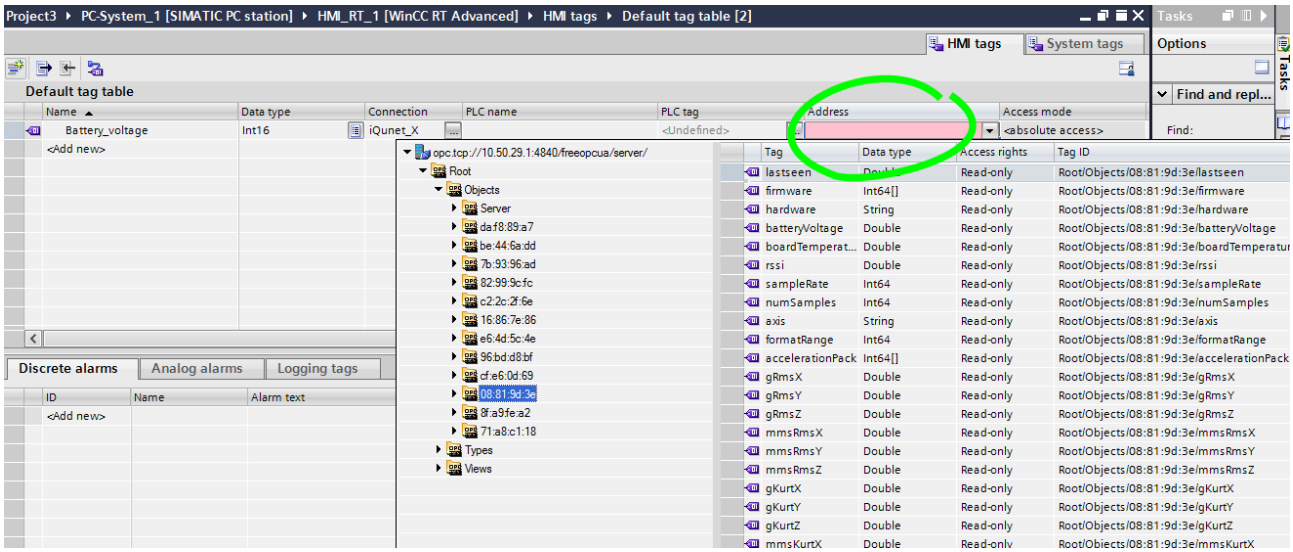
Set-up the HMI tags. Select “Default tag table [1]” under HMI tags.



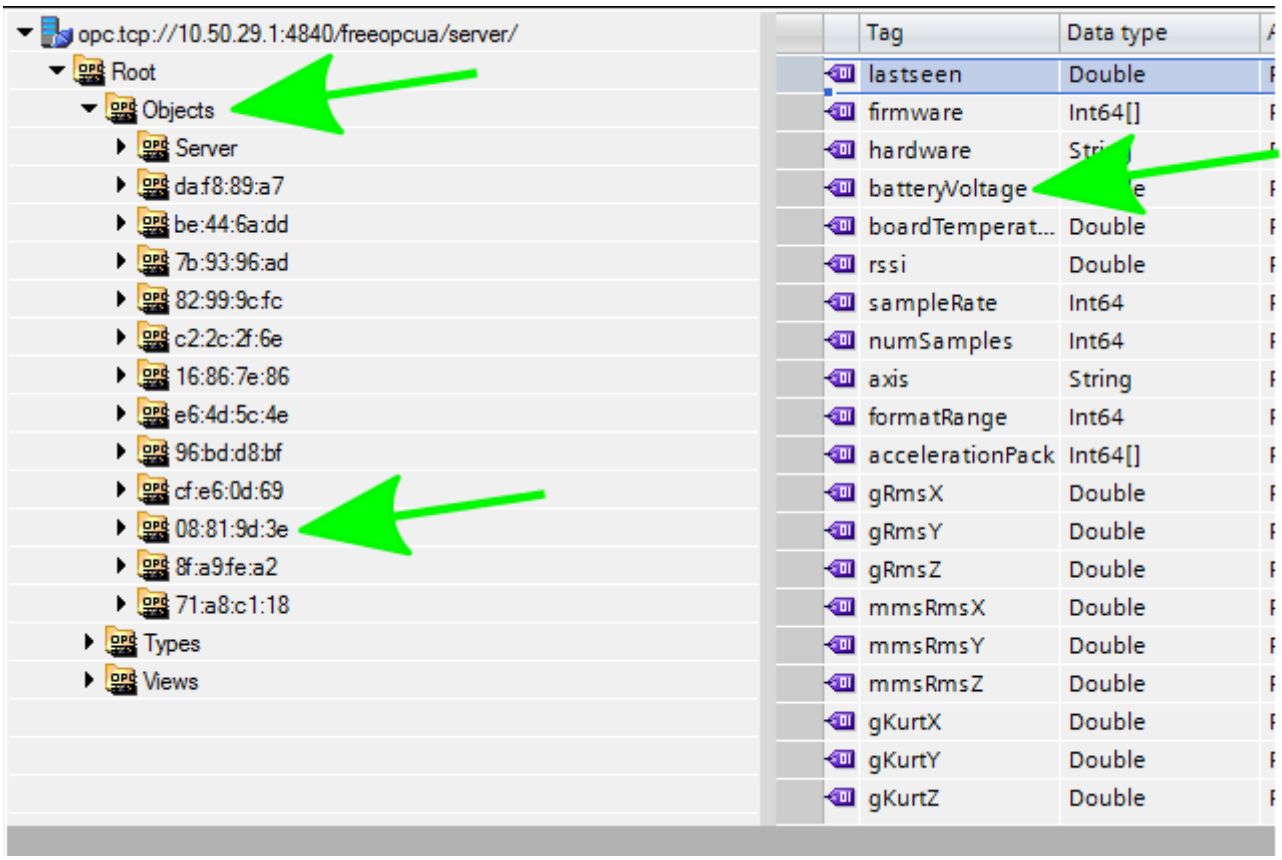
Add a new tag → Connection iQunet_X (as set above).

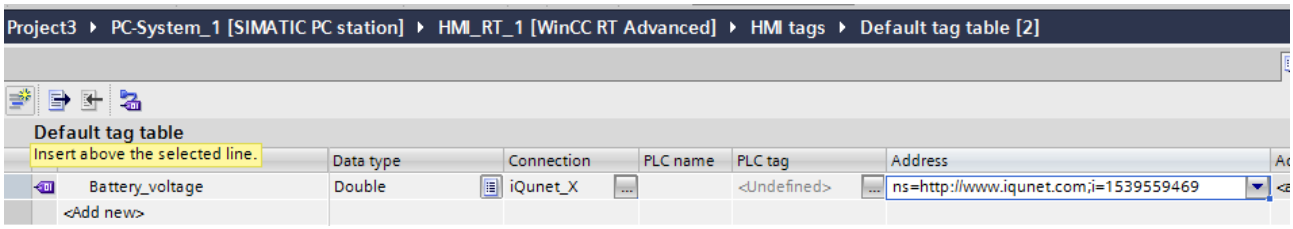


Choose the Address. Browse to the Objects node and select the sensor's macId.

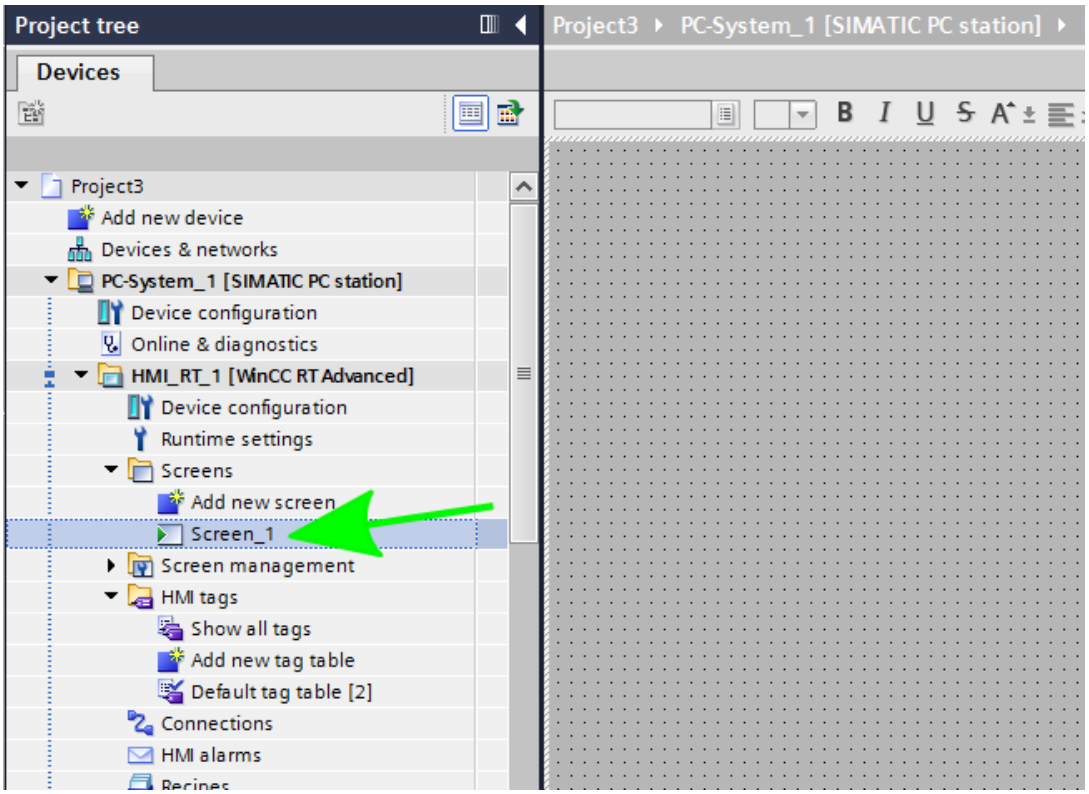


Select the sensor attribute or tag you want to observe.

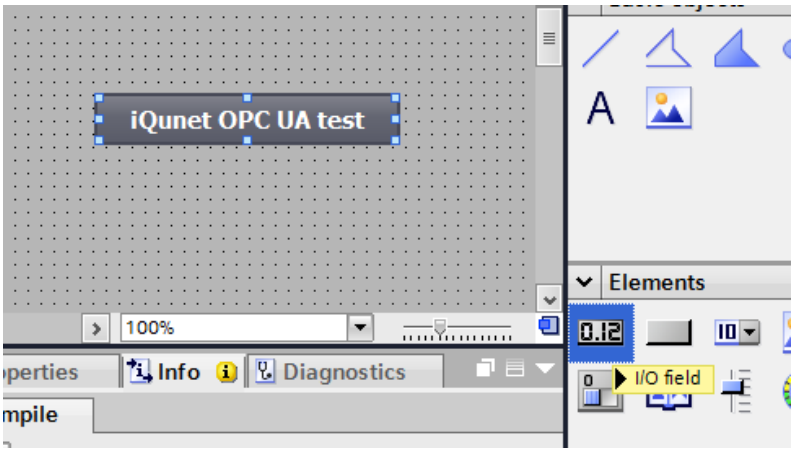


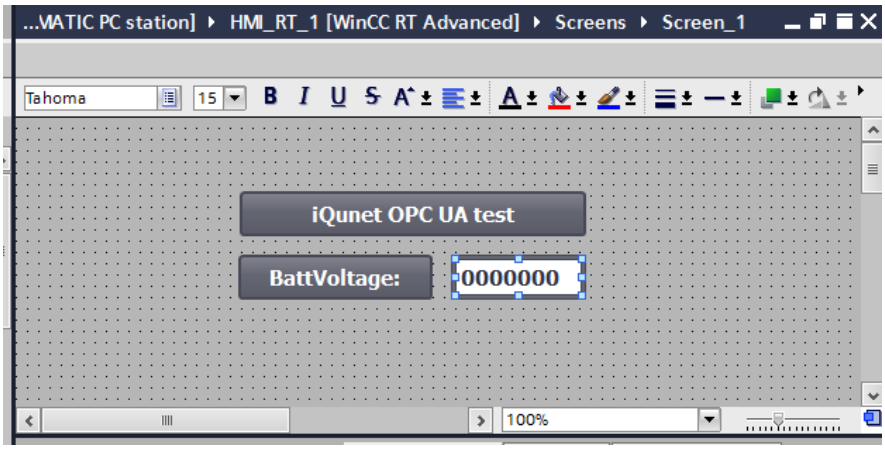


Add a new screen using the Screens node.

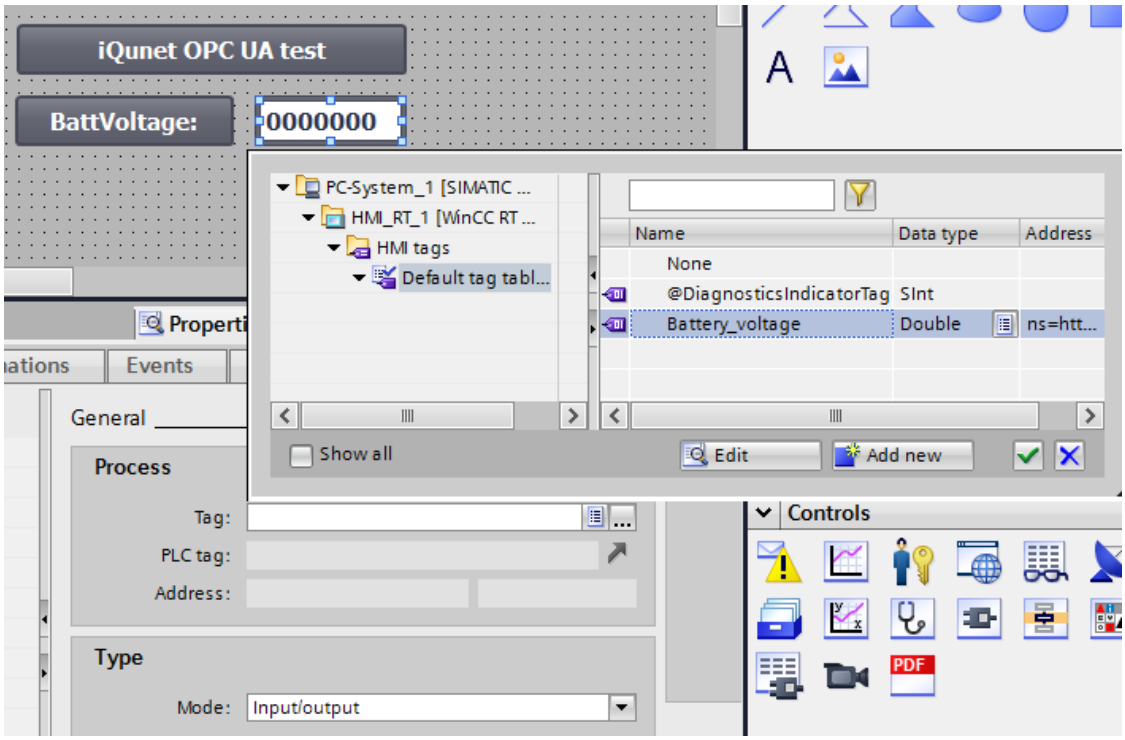


Add a new I/O widget from the Elements groupbox to the layout pane.

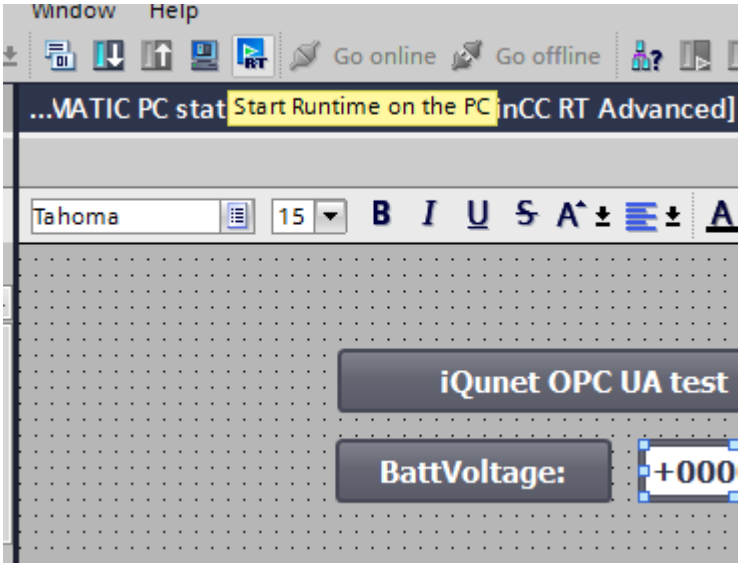




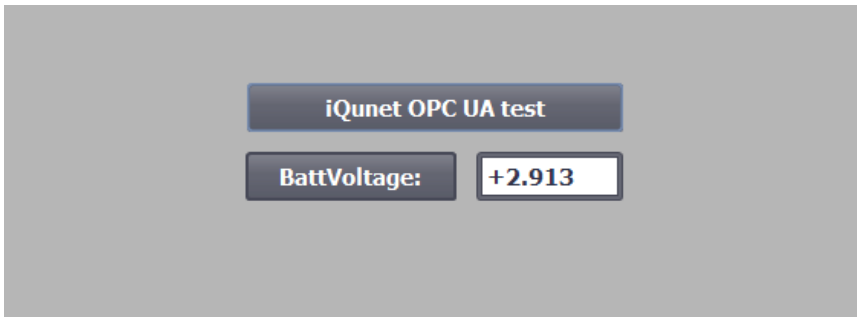
Select the HMI tag.



Compile and run.



Check if the OPC UA connection works.



Remark: please note that there is a certificate bug in older TIA WinCC Advanced Version V15 versions (and OPC Scout). You need service pack 1 for this to be fixed in TIA WinCC.

2.6 Python/Matlab

For OPC UA communication in Python the OPCUA library can be used (https://python-opcua.readthedocs.io/en/latest/index.html). You can find 2 example Python scripts on our Github page (https://github.com/iqunet/sern).

Matlab also offers an extension or toolbox to read data directly from OPC UA (https://nl.mathworks.com/products/opc.html).