CODESYS – OPC UA Information models
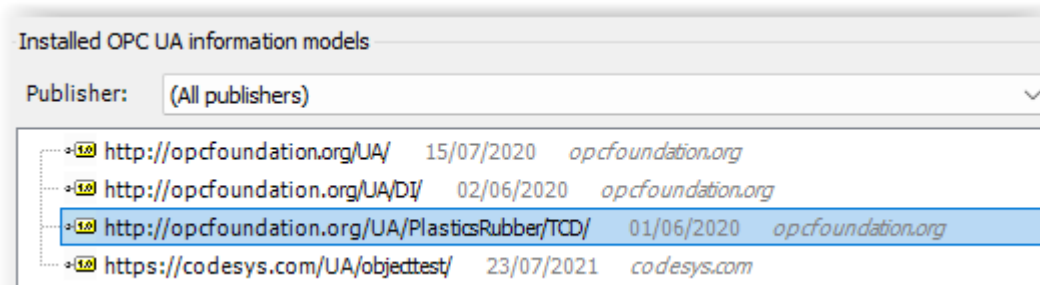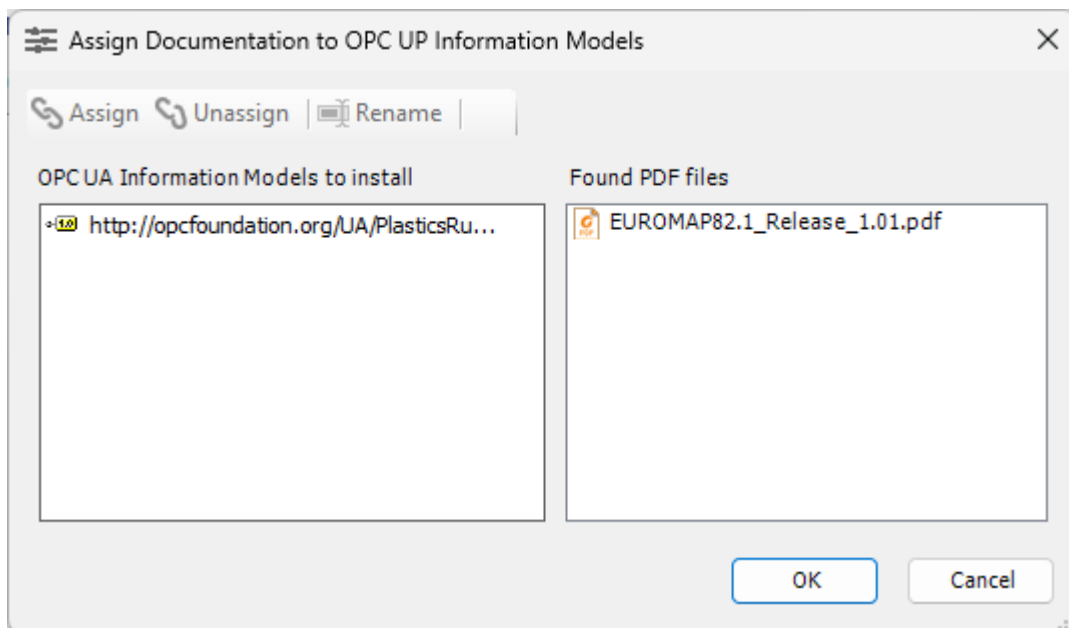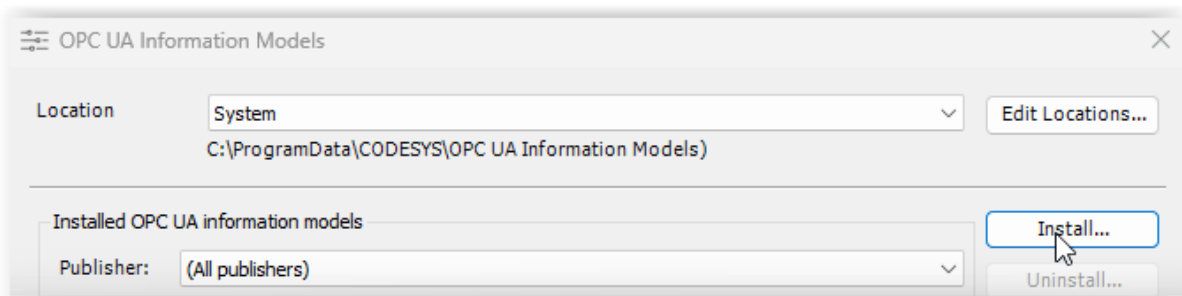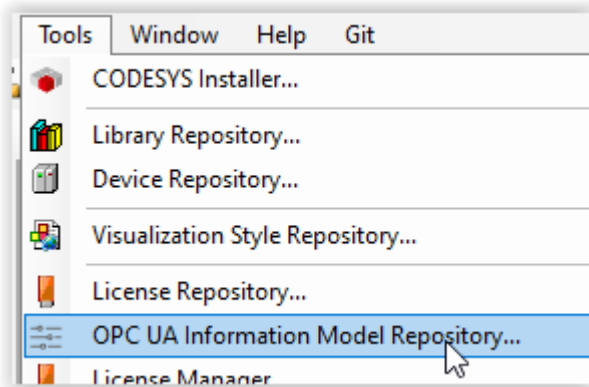
Install additional related (required) information models. E.g.:

https://reference.opcfoundation.org/PlasticsRubber/GeneralTypes/v103/docs/A

Devices

- *InformationModel*
  - Device (CODESYS Control Win V3 x64)
    - PLC Logic
      - **Application**
        - Communication Mar...
        - Library Manager
        - PLC_PRG (PRG)
      - Task Configuration
        - MainTask (IEC-...
          - PLC_PRG

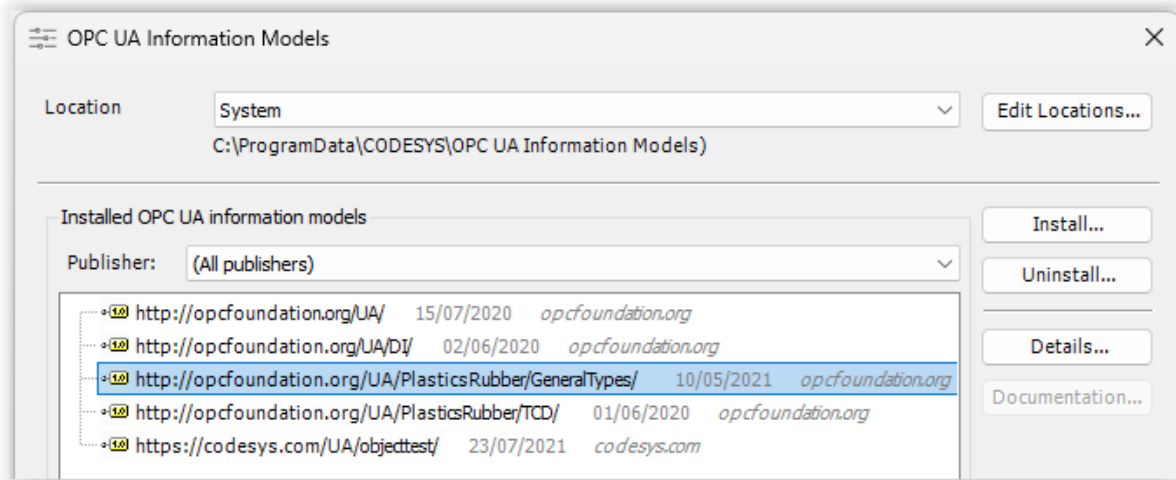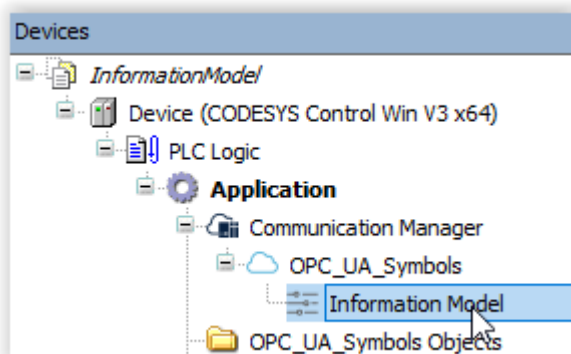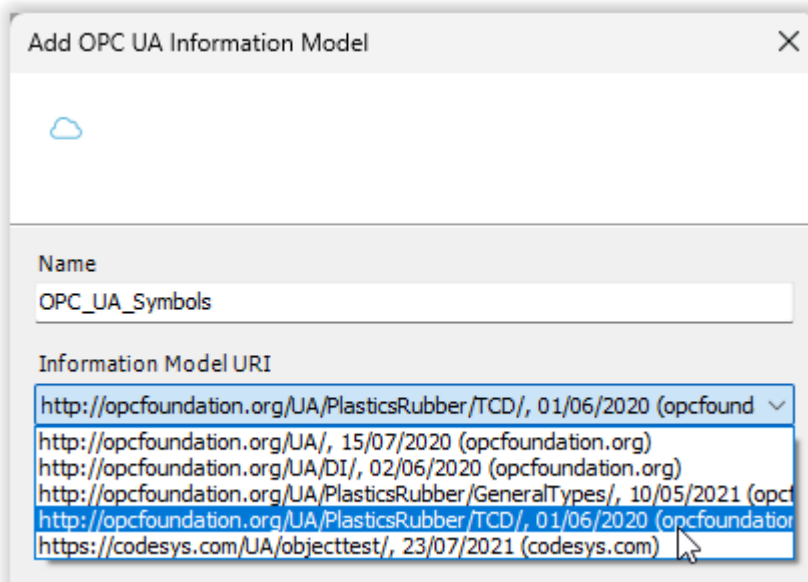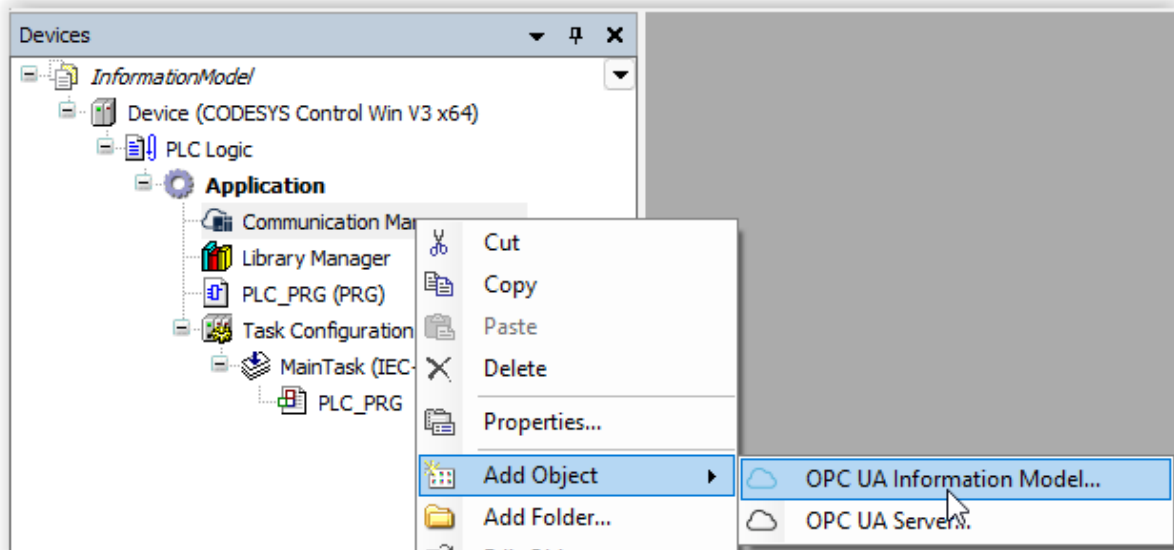|		
✂	Cut	
🗐	Copy	
📋	Paste	
✕	Delete	
🗐	Properties...	
🔳	Add Object	▶
📁	Add Folder...	

- ☁ OPC UA Information Model...
- ☁ OPC UA Servers...

Add OPC UA Information Model ✕

☁

Name

OPC_UA_Symbols

Information Model URI

http://opcfoundation.org/UA/PlasticsRubber/TCD/, 01/06/2020 (opcfound ⌄

http://opcfoundation.org/UA/, 15/07/2020 (opcfoundation.org)
http://opcfoundation.org/UA/DI/, 02/06/2020 (opcfoundation.org)
http://opcfoundation.org/UA/PlasticsRubber/GeneralTypes/, 10/05/2021 (opc
http://opcfoundation.org/UA/PlasticsRubber/TCD/, 01/06/2020 (opcfoundatio
https://codesys.com/UA/objecttest/, 23/07/2021 (codesys.com)

Devices

- *InformationModel*
  - Device (CODESYS Control Win V3 x64)
    - PLC Logic
      - **Application**
        - Communication Manager
          - OPC_UA_Symbols
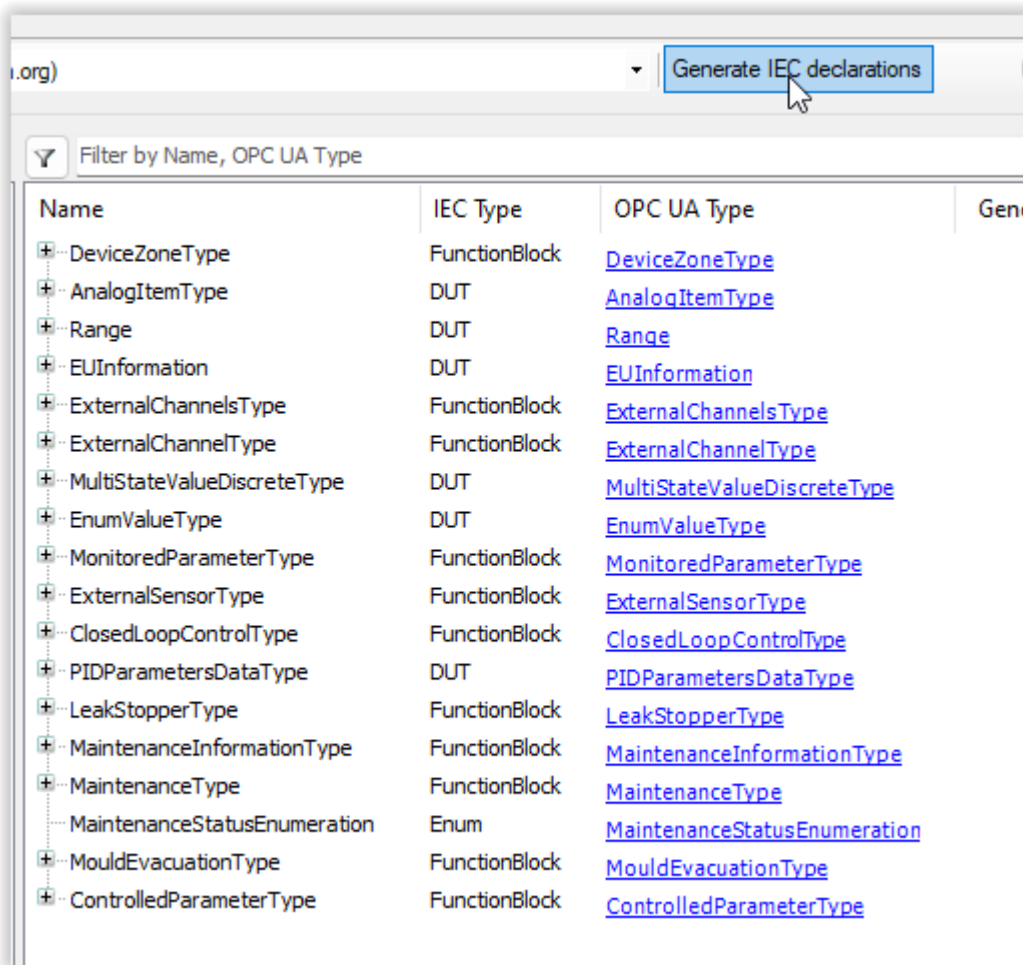   - Information Model
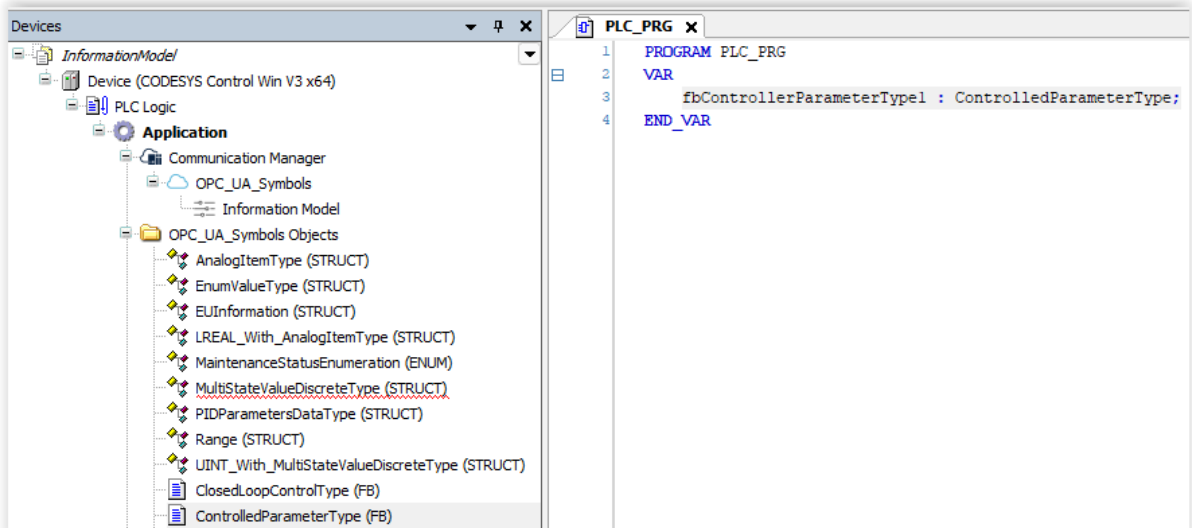   - OPC_UA_Symbols Objects
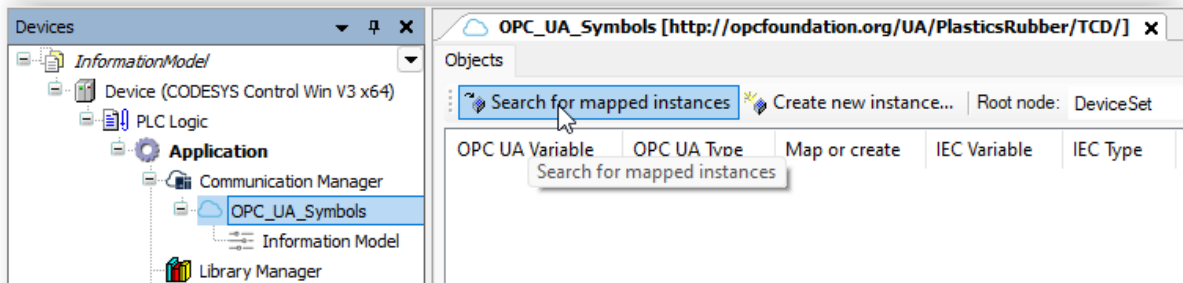
Drag a "Data model" to the right side:
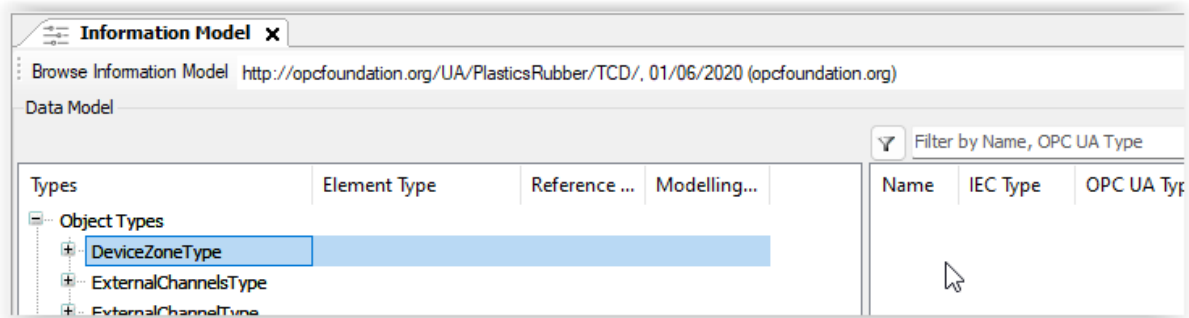
Then generate IEC objects for the "data model". Click on Generate IEC declarations:
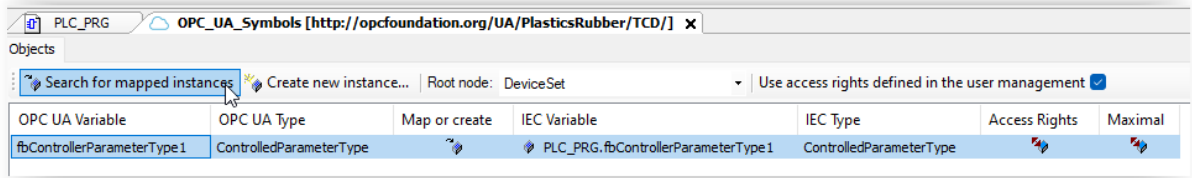

Now you could create an instance of a function block which is created.
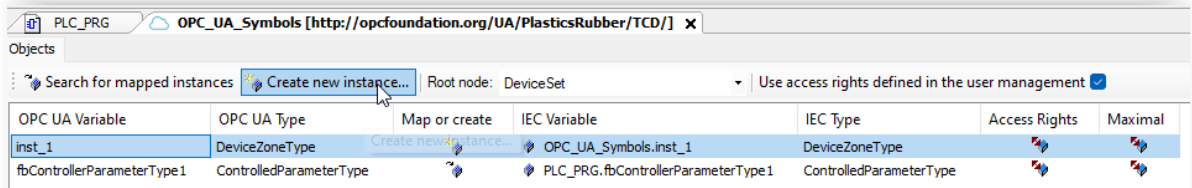This can be done somewhere in the application:

Then you can go back to the OPC_UP_Symbols and search for mapped instances:

This results in:

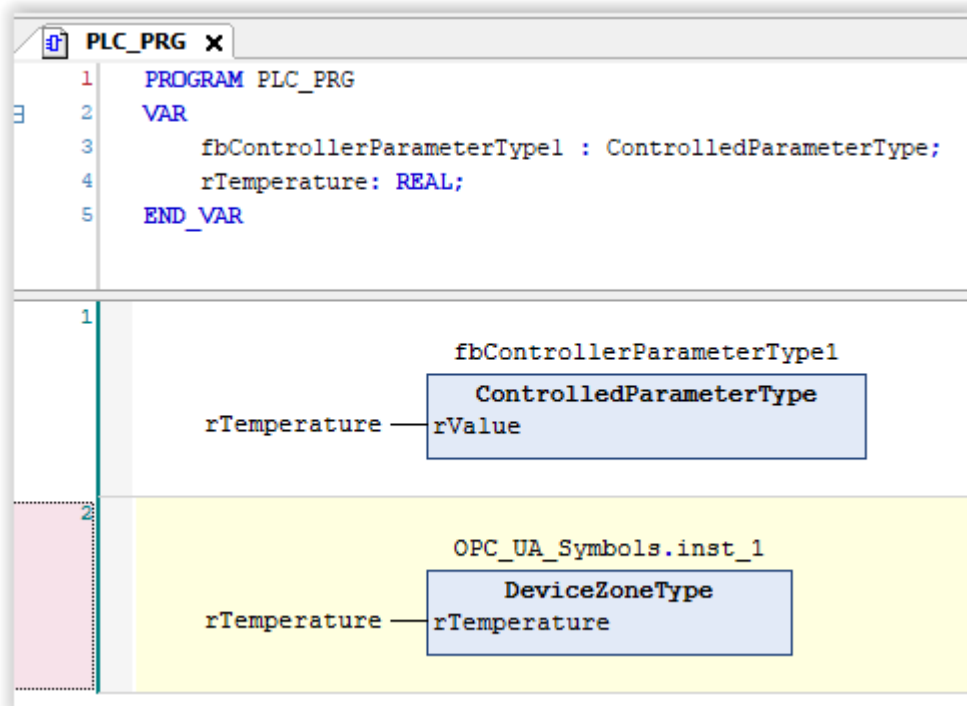You could also create a new instance in the OPC_UA_Symbols list:

You are allowed to modify the created instances to add your own specific functionality:

```
ControlledParameterType ✕
3    {attribute 'opcua.mapping.publicationdate' :=
4    {attribute 'opcua.mapping.nodeid' := 'nsu=htt|
5    FUNCTION_BLOCK ControlledParameterType
6    VAR_INPUT
7        rValue : REAL;
8    END_VAR
9    VAR
10       {attribute 'opcua.mapping.member.accessle|
11       ActualValue : LREAL_With_AnalogItemType;
12   END_VAR
13

1    ActualValue.value := rValue;
```

Or:

```
DeviceZoneType ✕
1    {attribute 'opcua.mapping.type'}
2    {attribute 'opcua.mapping.modeluri' := 'http:/
3    {attribute 'opcua.mapping.publicationdate' :=
4    {attribute 'opcua.mapping.nodeid' := 'nsu=http
5    FUNCTION_BLOCK DeviceZoneType
6    VAR_INPUT
7        rTemperature : REAL;
8    END_VAR
9    VAR
10       Temperature : ControlledParameterType;
11   END_VAR
12

1    Temperature(rValue:= rTemperature);
```

Then in your PLC application you can access these instances:

I got an error related to the following object which I manually changed:

After that I could build the application.

Then with OPC UaExpert, I could read the value:

Device.Application.PLC_PRG

Expression

⊞ ◆ fbControllerParameterType1
◆ rTemperature

1

```
                                              fbControllerParameterType1
                                          ┌──────────────────────────────┐
                                          │    ControlledParameterType     │
             rTemperature ┌──────────┐    │  rValue                        │
                          │    99    │────┤                                │
                          └──────────┘    └──────────────────────────────┘
```

Unified Automation UaExpert - The OPC Unified Architecture Client - NewProject*

File   View   Server   Document   Settings   Help

Project

Project
  Servers
    OPCUAServer@ESC-TIM-04
  Documents
    Data Access View

Data Access View

#	Server	Node Id	Display Name	Value
1	OPCUAServer@ES...	NS7|Opaque|0x01000...	ActualValue	99

Address Space

No Highlight

Root
  Objects
    DeviceSet
      CODESYS Control Win V3 x64
      DeviceFeatures
      fbControllerParameterType1
        ActualValue
      inst_1
        Temperature
          ActualValue
    DeviceTopology