BACnet: Writing of a Property (PresentValue)
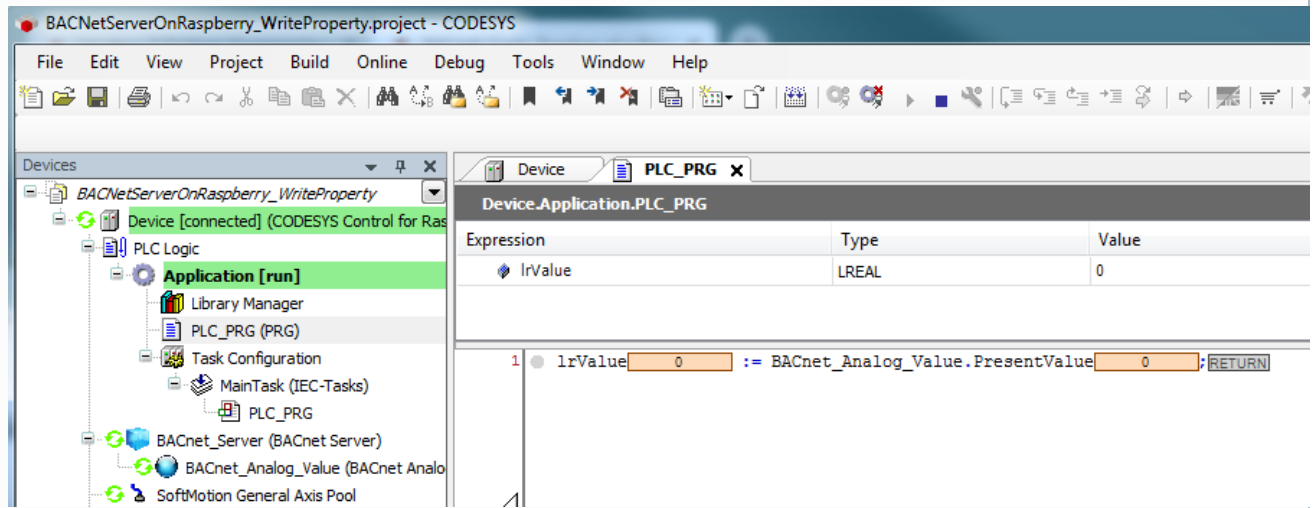
> ⓘ The BACnet server from the FAQ *BACnet: Cyclic Reading of a Property* is used as the server here.
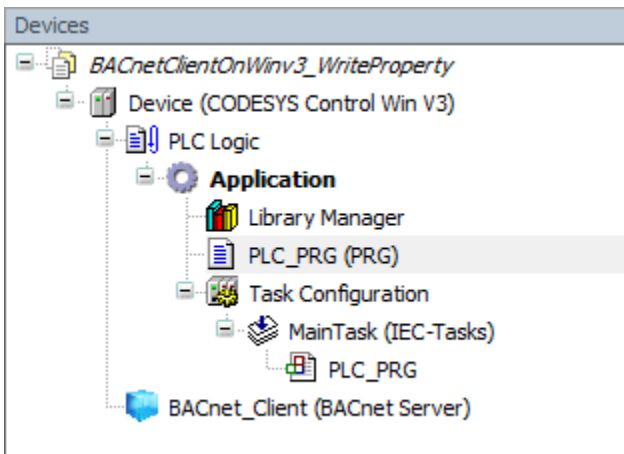>
> Reduce this one to reading the analog value.

Requirements for the server and client

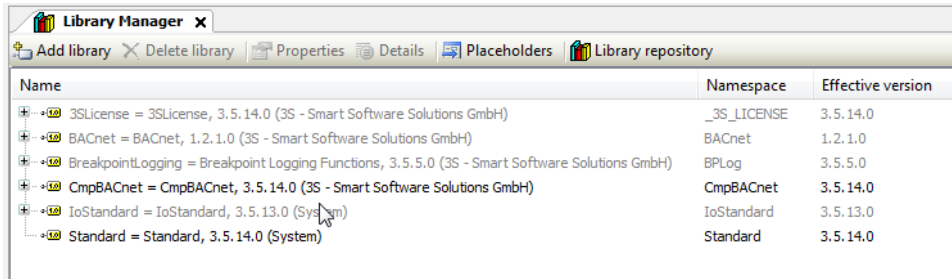Check that the following entries are present in the file *CODESYSControl.cfg*.

- Integration of the BACnet component:
  [ComponentManager]
  Component.[n+1]=CmpBACnet
- Integration of the INI file of the BACnet stack (pay attention to the syntax):
  [CmpBACnet]
  IniFile=bacstac**d**.ini

BACnet client project

- Create a "Standard project" and select *CODESYS Control Win V3* as the device.
- Define the target system by means of the network scan (see BACnet server).
- Insert a "BACNet Server" object into the device tree and rename it as "BACnet_Client".

- Open the *Library Manager* and add the following libraries: *CmpBACnet*

- Edit the *PLC_PRG* POU as follows:

Declaration

```
VAR
    fbWriteProperty : BACnet.BACnetClientWriteProperty;
    writePropVal    : BACnet.CmpBACnet.IEC_BACNET_REAL := 16.34;
    writePropCont   : BACnet.CmpBACnet.IEC_BACNET_PROPERTY_CONTENTS;
    xWriteExecute   : BOOL;
    writePrio       : CmpBACnet.IEC_BACNET_SIGNED := 16;
    xInitDone       : BOOL := FALSE;
END_VAR
```

Implemen
tation

```
IF NOT xInitDone THEN
    fbWriteProperty.RegisterToServer(BACnet_Client);
    fbWriteProperty(dwTargetDeviceNumber := 718,
                    objType := BACnet.CmpBACnet.IEC_BACNET_OBJECT_TYPE.OBJ_ANALOG_VALUE, objInst :
= 1,
                    propID := BACnet.CmpBACnet.IEC_BACNET_PROPERTY_ID.PROP_PRESENT_VALUE);
    xInitDone := TRUE;
ELSE
        writePropCont.buffer.pBuffer := ADR(writePropVal);
    writePropCont.buffer.nBufferSize := SIZEOF(writePropVal);
    writePropCont.nElements := 1;
    writePropCont.tag := BACnet.CmpBACnet.IEC_BACNET_DATA_TYPE.DATA_TYPE_REAL;

    fbWriteProperty(xExecute := xWriteExecute,
                    propertyContents := writePropCont,
                    nWritePriority := writePrio);
END_IF
```

- Download the project to the controller and set the *xWriteExecute* variable to *TRUE*.

The value is transferred to the server.

⚠ The *PresentValue* property is written with priorities from 1 to 16, where 16 is the lowest.

When you write a value with a higher priority to the server, it is retained until it is deleted or overwritten by an even higher priority.

During the delete operation, the value of the highest set priority is then output as PresentValue.

A priority is deleted by writing to it with the data type *DATA_TYPE_NULL*.