

# Anwendungshinweis CODESYS Control SL BACnet MS/TP

BACnet MS/TP via BACnet-MS/TP-Router MBS UBR-01 Mk II

CODESYS® ist ein eingetragenes Warenzeichen. Druckfehler, Irrtümer und technische Änderungen vorbehalten. Vervielfältigung (auch nur auszugsweise) und Weitergabe nur nach vorheriger Genehmigung. Bitte beachten Sie: Nicht alle CODESYS-Funktionen sind in allen Ländern verfügbar. Weitere Informationen zu diesen länderspezifischen Einschränkungen erhalten Sie unter [support@codesys.com](mailto:support@codesys.com).

# INHALT

	Seite	
<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Wichtige Eigenschaften der für dieses Beispiel ausgewählten Produkte</b>	<b>4</b>
<b>3</b>	<b>Einige grundlegende Informationen zu BACnet-MS/TP</b>	<b>4</b>
<b>4</b>	<b>Grundlegende Schritte bei der Integration</b>	<b>5</b>
4.1	BACnet-MS/TP-Installation	5
4.2	BACnet-MS/TP-Konfiguration	5
4.3	Auswahl BACnet-MS/TP-Geräteadresse	5
4.4	Integration der BACnet-Datenpunkte in die Applikation	5
<b>5</b>	<b>Konkrete Schritte bei der Integration anhand des Beispiels</b>	<b>6</b>
5.1	BACnet-MS/TP-Installation anhand des Beispiels	6
5.2	BACnet-MS/TP-Konfiguration und Inbetriebnahme anhand des Beispiels	6
5.3	Integration der BACnet-Datenpunkte in die Applikation anhand des Beispiels	9

# 1 Einleitung

CODESYS ist eine Automatisierungsplattform basierend auf einer Entwicklungsoberfläche unter Windows sowie Laufzeitumgebungen für kundenspezifische bzw. Standardgeräte wie PCs oder Boards mit ARM/Linux. Die Projektierung von Applikationen erfolgt nach dem internationalen Standard IEC 61131-3 ist auch für Aufgaben in der Gebäudeautomation einsetzbar.

BACnet ist eines der wichtigsten Kommunikationsprotokolle im Bereich Gebäudeautomation. Das Produkt CODESYS BACnet SL erweitert die Funktionalität einer SPS, um damit Applikationen im Bereich Gebäudeautomation standardkonform zu ANSI/ASHRAE Standard 135-2012 realisieren zu können – direkt integriert in das CODESYS Development System.

CODESYS BACnet SL kann für SPS mit CODESYS Control SL lizenziert werden und ist im CODESYS Store zu finden: <https://store.codesys.com/codesys-bacnet-sl.html>.

CODESYS BACnet erlaubt zwar grundsätzlich die Nutzung von BACnet-MS/TP, die konkrete Anpassung an eine spezielle Hardware (UART) muss aber zwingend durch den Steuerungshersteller (der CODESYS Control einsetzt) erfolgen. Daher unterstützt CODESYS BACnet SL beim Einsatz auf CODESYS Control SL ausschließlich BACnet-IP – nicht jedoch BACnet-MS/TP.

Die Integration von BACnet-MS/TP-Geräte mit CODESYS Control SL und CODESYS BACnet SL ist ohne viel Aufwand mit einem BACnet-MS/TP-Router möglich.

In diesem Beispiel wird eine solche Integration mit einem Produkt der MBS GmbH – dem „UBR-01 Mk II“ (<https://www.mbs-solutions.de/ubr-01-mk-ii-bacnet-router>) - und BACnet-MS/TP-Feldbus-Modulen von romutec Steuer- und Regelsysteme GmbH – RDC600 (<https://www.romutec.de/Produkte/Kommunikation/RDC-BACnet.html>) demonstriert.



Template:

## 2 Wichtige Eigenschaften der für dieses Beispiel ausgewählten Produkte

### UBR-01 Mk II

Der Universal-BACnet-Router UBR-01 ermöglicht die Umsetzung der BACnet-Netzwerk-Topologien ISO 8802-2 (auch als BACnet/Ethernet bekannt), BACnet/IP und MS/TP (serielle BACnet-Netzwerke auf Basis RS485).

Wichtige Eigenschaften des UBR-01:

- Integrierter Webserver zur Konfiguration und Analyse
- Keine beweglichen Teile wie Lüfter oder ähnliches
- integrierte und schaltbare Netzwerk- und Bias-Widerstände
- Slave-Proxy Mode / Auto-Slave-Detection
- Unterstützung internationaler Sprachpakete

### Romutec RDC600

Die RDC600-Serie beinhaltet Feldbusmodule zur Steuerung von Datenpunkten über RS485 Protokoll BACnet MS / TP. Wichtige Eigenschaften der RDC600-Serie:

- Praxistaugliche Mischung aus festem / konfigurierbarem I/O
- Integrierte, lokale Bedienebene im Schaltschrank und optional in der Schaltschranktür
- Sowohl zentrale als auch dezentrale Verwendung mit den für BACnet-MS/TP möglichen Verkabelungslängen bis 1200 m
- Inbetriebnahme der Feldgeräte auch ohne Bus-Kommunikation und DDC-System
- Parametrierung sämtlicher Einstellungen über BACnet-Objekte

Andere BACnet-Router (wie Contemporary Controls BASrouter <https://www.ccontrols.de/basautomation/basrouter.htm>) ermöglichen eine vergleichbare Vorgehensweise bei der Integration.

## 3 Grundlegende Informationen zu BACnet-MS/TP

BACnet-MS/TP "**M**aster-**S**lave/**T**oken-**P**assing" ist eines der im BACnet-Standard definierten BACnet-Datenlink-Protokolle basierend auf EIA-485 (als physikalische Schicht). Grundprinzip von BACnet-MS/TP ist das sogenannte "Token-Passing" – ein "Token" wird von einem Gerät zum anderen übertragen. Ein Gerät kann erst nach Erhalt dieses Tokens kommunizieren. BACnet-MS/TP unterscheidet dabei zwischen zwei Arten von Geräten – „Master“ und „Slave“.

BACnet-MS/TP wird kaum als allgemeiner BACnet-Backbone verwendet, da BACnet-MS/TP für diesen Zweck eine zu geringe Datenrate aufweist. In der Regel wird BACnet-IP als allgemeine BACnet-Backbone eingesetzt.

BACnet-MS/TP unterstützt Kabellängen bis 1200 m ohne zusätzliche Geräte (wie Repeater etc.) und ermöglicht so eine verteilte Automatisierung.

BACnet-MS/TP-Geräte sind kostengünstige Alternativen für Anwendungen, die nur geringe Anforderungen an Ein-Ausgabe-Datenraten aufweisen und bei denen BACnet-IP-Geräte meist aus Kostengründen nicht eingesetzt werden (Gerätekosten, Verkabelungskosten, Störfestigkeit ohne besondere Maßnahmen).

## 4 Grundlegende Schritte bei der Integration

Folgende Schritte sind für eine Integration notwendig:

1. BACnet-MS/TP-Installation
2. Konfiguration und Inbetriebnahme von BACnet-MS/TP
3. Integration der BACnet-Datenpunkte in die Applikation (der BACnet-MS/TP-Feldbus-Module und ggf. des BACnet-MS/TP-Routers)

### 4.1 BACnet-MS/TP-Installation

Die Verkabelung muss entsprechend den Vorgaben des BACnet-Standards erfolgen – siehe "ANSI/ASHRAE Standard 135-2012" - Kapitel 9.2 „Physical Layer“.

Wichtige Aspekte:

- Zweidraht "twisted pair"
- Polung (+/-) beachten
- Abschlusswiderstände 120 Ohm beidseitig des Netzwerksegmentes
- Ein oder zwei Paar Bias-Widerstände 510 Ohm innerhalb des Netzwerksegmentes – bevorzugt an den Enden des Netzwerksegmentes
- Ggf. lokale Bias-Widerstände 47 kOhm an einigen Geräten
- Begrenzte Anzahl von Geräten im Netzwerksegment (in der Regel max. 32)

### 4.2 Konfiguration von BACnet-MS/TP

#### Auswahl Master/Slave

Slave-Geräte haben einen Vorteil: Je mehr Slave-Geräte (d. h. je weniger Master) auf dem BACnet-MS/TP-Bus existieren, umso weniger Token-Übergaben finden statt. Dies beschleunigt die BACnet-MS/TP-Kommunikation. Nachteilig ist, dass Slave-Geräte für den Aufbau einer Kommunikation einen Master im Netzwerk benötigen. Daher werden die meisten BACnet-MS/TP-Geräte als Master ausgeführt oder zumindest defaultmäßig als Master konfiguriert.

### 4.3 Auswahl einer Geräteadresse für BACnet-MS/TP

Ein Master im BACnet-MS/TP-Bus sendet zyklisch Anfragen an die nächsten potenziellen Master zwischen der eigenen Adresse und der nächsten bekannten Master-Adresse aus, um zu prüfen, ob ein neuer Master in diesem Bereich zum BACnet-MS/TP-Bus hinzugefügt wurde. Dabei muss bis zu einem Timeout gewartet werden, um sicherzustellen, dass kein neuer Master in diesem Bereich hinzugefügt wurde. Daher sollten Adresslücken in den BACnet-MS/TP-Geräteadressen bei Master-Geräten vermieden werden, um die ohnehin geringe Datenrate nicht teilweise für unnötige Abfragen zu verwenden. Die Einstellung "MaxMaster" erlaubt eine Einschränkung des aktiven Adressbereichs.

### 4.4 Integration der BACnet-Datenpunkte in die Applikation

Datenpunkte der BACnet-MS/TP-Feldbus-Module sowie des BACnet-MS/TP-Routers werden vom jeweiligen Gerät als als BACnet(-Server)-Objekte repräsentiert. Die Properties dieser Objekte können mit den CODESYS-BACnet-Funktionsblöcken BACnetClientReadProperty und BACnetClientWriteProperty gelesen bzw. geschrieben werden. BACnet-COV-/Event-Subscriptions sowie eine ggf. notwendige Geräteerkennung (device discovery) sind bei Bedarf ebenso unter Nutzung von CODESYS-BACnet-Funktionsblöcken möglich.

Template:

## 5 Konkrete Schritte bei der Integration anhand des Beispiels

### 5.1 BACnet-MS/TP-Installation anhand des Beispiels

Im Beispiel werden zwei BACnet-MS/TP-Feldbus-Module RDC601 verwendet.  
Diese verfügen über:

- 8 digitale Eingänge
- 6 digitale Ausgänge als Relaisausgänge mit konfigurierbarer Handebene
- 4 analoge Eingänge für verschiedene Fühler und Spannung.

Diese werden als BACnet-BinaryInput, -BinaryOutput und -AnalogInput repräsentiert.

Der BACnet-MS/TP-Router UBR-01 Mk II verfügt über integrierte, mit DIP-Schalter aktivierbare Abschluss- und Bias-Widerstände. Sofern der UBR-01 Mk II an einem Ende des Netzwerksegmentes platziert wird, kann der Abschlusswiderstand zur Terminierung eingesetzt werden. Das andere Ende des Netzwerksegmentes muss in der Verdrahtung terminiert werden.

### 5.2 Konfiguration und Inbetriebnahme von BACnet-MS/TP anhand des Beispiels

Der BACnet-MS/TP-Router UBR-01 Mk II wird grundsätzlich als Master betrieben.  
Die beiden BACnet-MS/TP-Feldbus-Module RDC601 werden ebenso als Master konfiguriert.  
Dazu wird eine BACnet-MS/TP-Geräteadresse im Bereich von 1...127 gewählt.

Im Beispiel wurden die BACnet-MS/TP-Geräteadressen wie folgt festgelegt:

- UBR-01 – Adresse 1
- RDC601 – Adressen 2 und 3

Die Konfiguration der BACnet-MS/TP-Geräteadresse der RDC601 erfolgt über Drehkodierschalter am Gerät.

Die Konfiguration der BACnet-MS/TP-Geräteadresse des UBR-01 erfolgt via Web-Browser.  
Hierfür ist zunächst eine Ethernet-Verbindung zum UBR-01 aufzubauen. Details hierzu im Benutzerhandbuch.

Nach erfolgter Anmeldung am Webinterface des UBR-01 kann dieser konfiguriert werden:

MBS UBR-01 | Mk II | UBR

ALLGEMEIN BACNET DIAGNOSE HILFE Benutzer: admin

### Allgemein

- Übersicht
- Angaben
- IP-Netzwerk
- Systemzeit
- Benutzer
- Datensicherung
- Update
- Neustart

### IP-Netzwerk Einstellungen

Stand von: Montag, 26.10.2020 17:05:49



Parameter	Wert	
<b>Netzwerk-Adapter LAN1</b>		
Die IP-Adresse:	192 . 168 . 1 . 50	
Netzmaske:	255 . 255 . 255 . 0	
<b>Default gateway</b>		
Gateway:	192 . 168 . 1 . 1	<input type="button" value="Löschen"/>
<b>Advanced IP-Routing</b>		
<b>Typ</b>	<b>Adapter</b>	<b>IP</b>
<b>Netmask</b>	<b>Gateway</b>	<b>Status</b>
<b>Editieren</b>		
<input type="button" value="Neu"/>		
<b>Netzwerk Name</b>		
Hostname:	ubr	
Nameserver 1:		<input type="button" value="Löschen"/>
Nameserver 2:		<input type="button" value="Löschen"/>
<b>Dienste</b>		
Webserver Zugriff:	http und https (80/443) v	
aktiviere SSH Zugriff:	<input checked="" type="checkbox"/>	
<input type="button" value="Speichern"/>		

### Festlegung des Router-Modus

MBS UBR-01 | Mk II | UBR

ALLGEMEIN **BACNET** DIAGNOSE HILFE Benutzer: admin

### BACnet

- Einstellungen
- Device-Objekt

### BACnet Datenverbindungen

Stand von: Montag, 26.10.2020 17:07:28



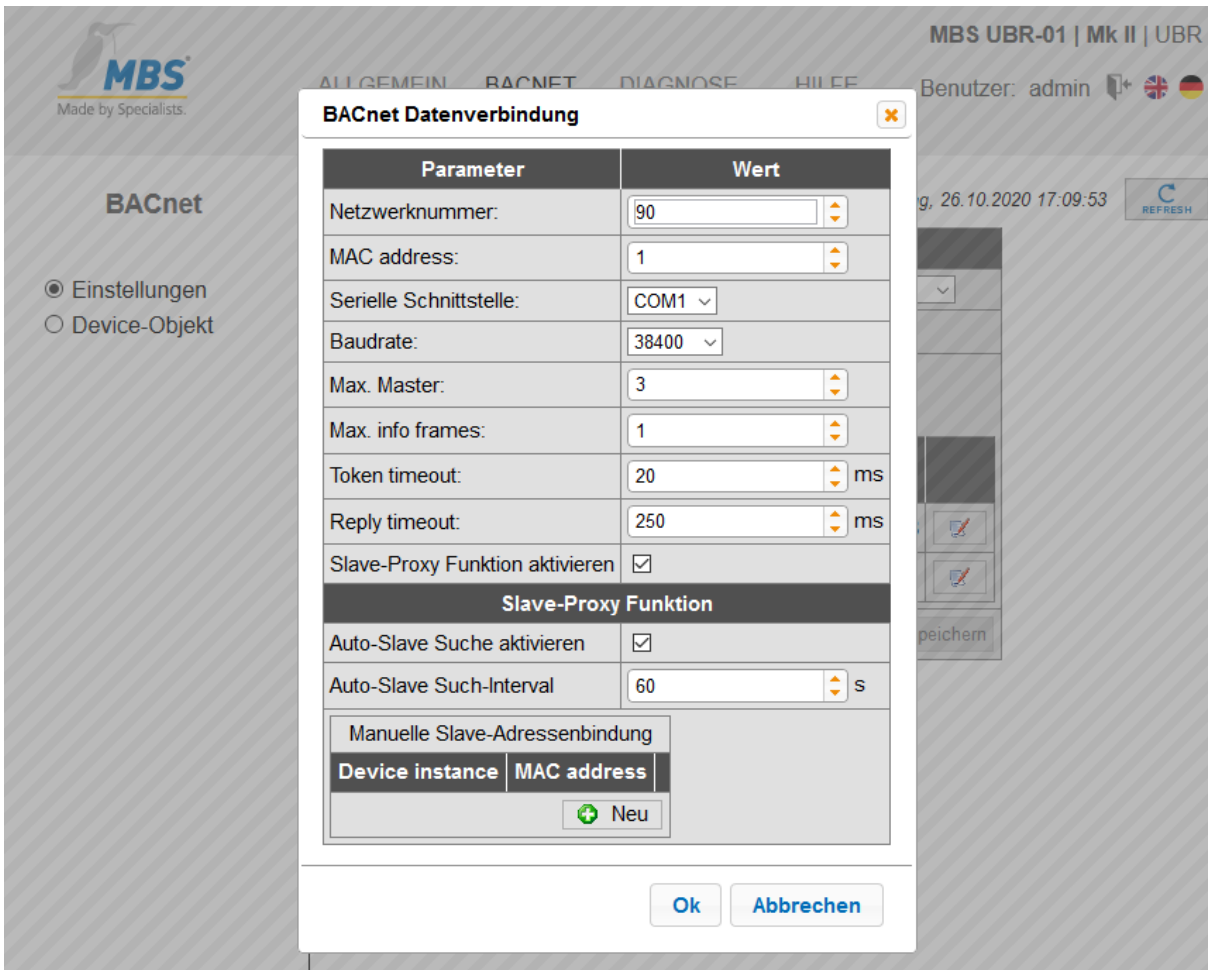
Eigenschaft	Wert		
Router Modus:	BACnet IP and MS/TP v		
Passwort DCC/RD:	ubr		
Datenverbindungen:			
Port-Id	Netzwerk-Nummer	Link-Typ	Angaben
1	92	ip	Mode:ip LAN1 UDP-Port:47808 <input type="button" value="Editieren"/>
2	90	mstp	MAC:1 COM1 Baudrate:38400 <input type="button" value="Editieren"/>
<input type="button" value="Speichern"/>			

Template:

### Konfiguration von

- BACnet-Netzwerknummer (des BACnet-MS/TP-Segmentes)
- BACnet-MS/TP-Geräteadresse
- Schnittstellenparameter

- MaxMaster



### Diagnose der BACnet-MS/TP-Geräte



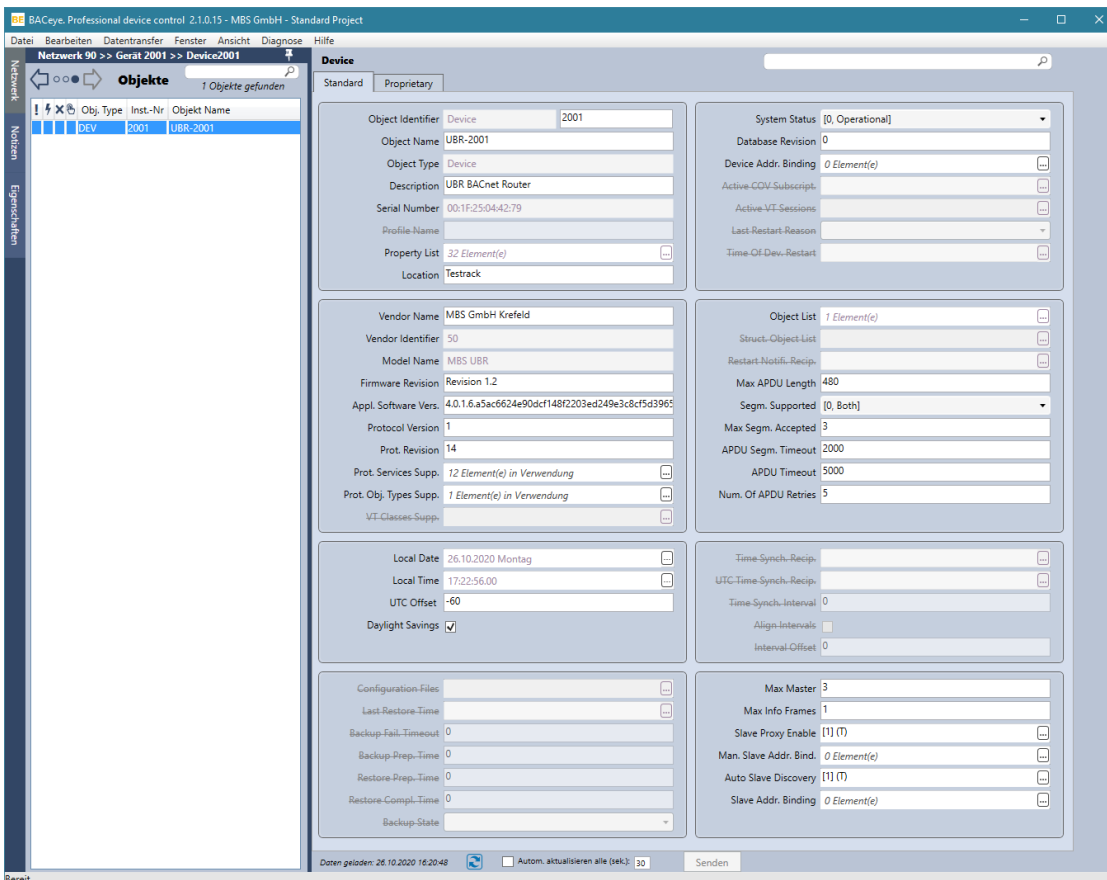
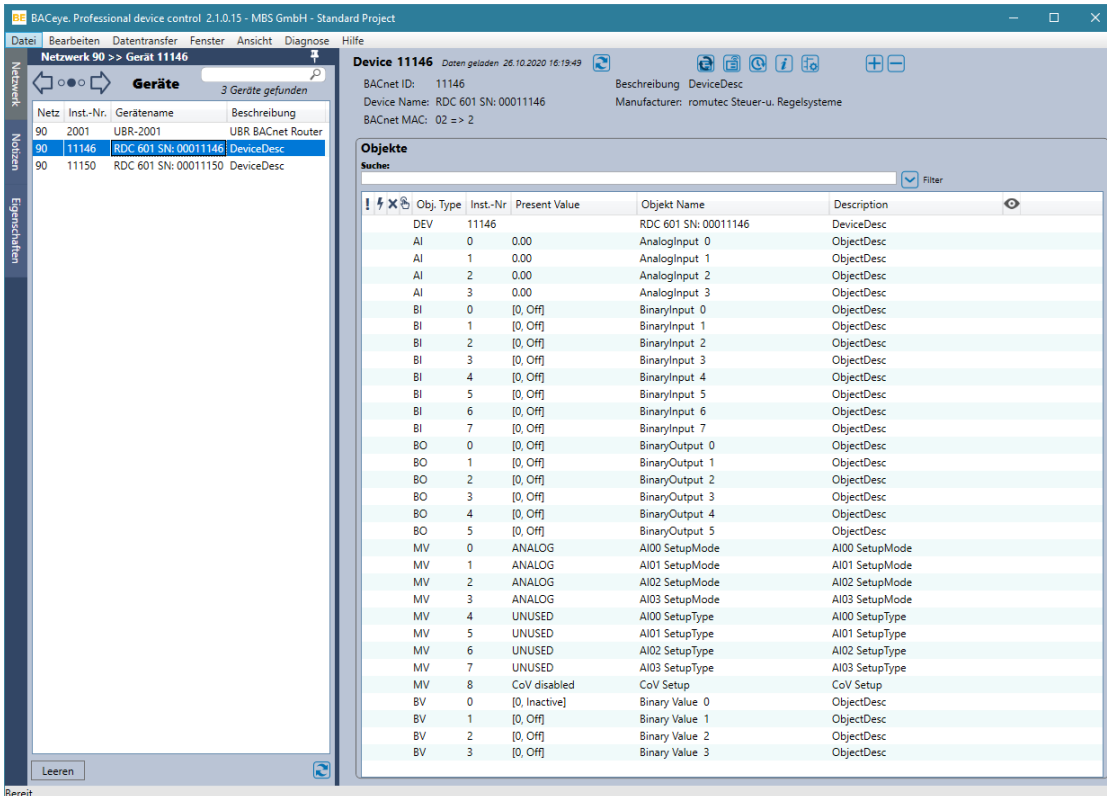
Der UBR-01 verfügt über folgende für die Inbetriebnahme wichtige Funktionen:

- Routing-Statistik
- Datalink-Statistik
- Paketlogs

Template:

Nach erfolgter Konfiguration und Inbetriebnahme sind die BACnet-Geräte nun im BACnet-IP-Netzwerk verfügbar und können mit einem BACnet-Explorer (im Beispiel BACeye der MBS GmbH) bearbeitet werden.



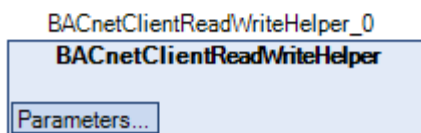
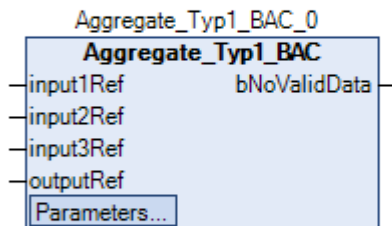
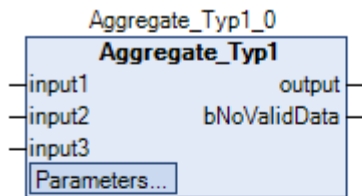


Template:

### 5.3 Integration der BACnet-Datenpunkte in die Applikation anhand des Beispiels

Im Beispiel wird ein fiktives HVAC-Aggregat modelliert, das optional mit einem BACnet-Anbindungs-„Wrapper“ versehen werden kann. Eine solche Vorgehensweise beim Design von CODESYS-Bibliotheken ist dann sinnvoll, wenn Aggregat-Implementierungen in Bibliotheken nicht unmittelbar mit den Ein- oder Ausgängen von Prozessdaten eines spezifischen Kommunikationsprotokolls verbunden werden sollen, sondern wahlweise mit

den Prozessdaten anderer Ein- und Ausgänge verwendet werden sollen, beispielsweise mit den Prozessdaten lokaler I/Os oder verschiedener Kommunikationsprotokolle wie BACnet, MODBUS oder CANopen.



Der Funktionsblock Aggregate\_Typ1 stellt die Aggregat-Implementierung dar, die Ein- und Ausgänge der Prozessdaten sind als VAR\_INPUT und VAR\_OUTPUT deklariert.

Der Funktionsblock Aggregate\_Typ1\_BAC stellt einen „BACnet-Wrapper“ für einen Aggregate\_Typ1 dar, die Verweise auf konkrete BACnet-Datenpunkte sind als VAR\_INPUT deklariert.

Der Funktionsblock BACnetClientReadWriteHelper vereinfacht das Lesen / Schreiben von BACnet-Objekt-Properties mittels BACnetClientReadProperty und BACnetClientWriteProperty.

```

1 // Read an analog input from BACnet fieldbus device.
2 // Returns TRUE, if done.
3 METHOD ReadAnalogInput : BOOL
4   VAR_INPUT
5     inputRef : BACnetObjectPropertyReference;
6   END_VAR
7   VAR_IN_OUT
8     readProp : BACnet.BACnetClientReadProperty;
9     data : AnalogProcessData;
10  END_VAR
11  VAR
12    objIdUsed : BACnet.CmpBACnet.IEC_BACNET_OBJECT_ID;
13    timeStamp : SYSTIME;
14  END_VAR
15
16  ReadAnalogInput := FALSE;
17
18  // configure readProp and trigger xExecute
19  IF NOT readProp.xExecute AND NOT readProp.xAborted AND NOT readProp.xDone THEN
20    SysTimeGetUs(timeStamp);
21    IF timeStamp < _errorTimeStamp + udiErrorDownTime * 1000 THEN
22      RETURN;
23    END_IF
24    readProp.dwTargetDeviceNumber := inputRef.devInstance;
25    readProp.objType := inputRef.objId.typ;
26    readProp.objInst := inputRef.objId.instNumber;
27    readProp.propID := inputRef.propID;
28    readProp.nIndex := inputRef.nIndex;
29    readProp.xExecute := TRUE;
30    IF verbose THEN
31      LogInfo(CONCAT('trigger read value from ',
32        BACnet.PropertyAddrString(readProp.dwTargetDeviceNumber, inputRef.objId, readProp.propID)));
33    END_IF
34  IF readProp.xExecute THEN
35    IF readProp.xDone THEN
36      readProp.xExecute := FALSE;
37      data.bDataValid := TRUE;
38      data.rValue := BACnet.GetRealFromContents(readProp.result);
39      _errorTimeStamp := 0;
40      objIdUsed.typ := readProp.objType;
41      objIdUsed.instNumber := readProp.objInst;
42      ReadAnalogInput := TRUE;
43      IF verbose THEN
44        LogInfo(CONCAT(CONCAT(CONCAT('got value from ',
45          BACnet.PropertyAddrString(readProp.dwTargetDeviceNumber, objIdUsed, readProp.propID)), ' '), TO_STRING(data.rValue)));
46      END_IF
47    IF readProp.xError THEN
48      readProp.xExecute := FALSE;
49      data.bDataValid := FALSE;
50      SysTimeGetUs(_errorTimeStamp);
51      objIdUsed.typ := readProp.objType;
52      objIdUsed.instNumber := readProp.objInst;
53      LogWarning(CONCAT(CONCAT(CONCAT('failed to get value from ',
54        BACnet.PropertyAddrString(readProp.dwTargetDeviceNumber, objIdUsed, readProp.propID)), ' eStatus='), TO_STRING(readProp.eStatus)));
55    END_IF
56  END_IF
57  readProp();

```

Die Implementierung des „BACnet-Wrappers“ für Aggregate\_Typ1\_BAC ist damit sehr einfach zu realisieren.

```

1 // Library element Aggregate_Typ1_BAC
2 //
3 // Aggregate_Typ1_BAC is a BACnet-„wrapper“ for Aggregate_Typ1 to read/write sensor/actuator/status values from/to BACnet fieldbus devices.
4 //
5 FUNCTION_BLOCK Aggregate_Typ1_BAC
6   VAR_INPUT_CONSTANT
7     // Fictitious aggregate identifier
8     aggregateId : UINT := 0;
9   END_VAR
10  // Some aggregate parameters ...
11  // Limit to check input1 against.
12  rLimit : REAL := 42.0;
13  END_VAR
14  VAR_INPUT
15    // object-property-reference for inputs/output
16    input1Ref : BACnetObjectPropertyReference;
17    input2Ref : BACnetObjectPropertyReference;
18    input3Ref : BACnetObjectPropertyReference;
19    outputRef : BACnetObjectPropertyReference;
20  END_VAR
21  VAR_OUTPUT
22    bNoValidData : BOOL;
23  END_VAR
24  VAR
25    _aggregate : Aggregate_Typ1;
26    // We do work with one BACnetClient(Read/Write)Property function block per input / output data here.
27    // Of course it's possible to share one BACnetClientReadProperty for all input data, and one BACnetClientWriteProperty for all output data.
28    // Other options could be chosen based on number of input / output data etc.
29    _readInput1 : BACnet.BACnetClientReadProperty;
30    _input1 : AnalogProcessData;
31    _inputHelper : BACnetClientReadWriteHelper;
32
33    _readInput2 : BACnet.BACnetClientReadProperty;
34    _input2 : BinaryProcessData;
35    _input2Helper : BACnetClientReadWriteHelper;
36
37    _readInput3 : BACnet.BACnetClientReadProperty;
38    _input3 : BinaryProcessData;
39    _input3Helper : BACnetClientReadWriteHelper;
40  END_VAR
41  _aggregate(aggregateId:=aggregateId, rLimit:=rLimit, input1:=_input1, input2:=_input2, input3:=_input3, output=>_output, bNoValidData=>bNoValidData);
42
43  // Simple measure to deal with invalid output data: check if we eventually try to write data to Binary[OutputValue].
44  // In this case write Out_of_Service accordingly.
45  // In real world implementations this needs to be more sophisticated, but this would leave the scope of this example.
46  IF (outputRef.objId.typ = BACnet.CmpBACnet.IEC_BACNET_OBJECT_TYPE_OBJ_BINARY_OUTPUT OR
47    outputRef.objId.typ = BACnet.CmpBACnet.IEC_BACNET_OBJECT_TYPE_OBJ_BINARY_VALUE) AND
48    outputRef.propID = BACnet.CmpBACnet.IEC_BACNET_PROPERTY_ID_PROP_PRESENT_VALUE THEN
49    _writeOutofService := TRUE;
50  END_IF
51  _outputHelper.WriteBinaryOutput(outputRef, _output, _writeOutofService);
52  IF _writeOutofService THEN
53    _outofServiceHelper.WriteOutofService(outputRef, bNoValidData, _writeOutofService);
54  END_IF

```

Template:

Weitere Details können dem beiliegenden Beispiel entnommen werden welches unter <https://forge.codesys.com/prj/codesys-example/appnotebacnmstp/home/Home/> zu finden ist.