



## CtBACnetIP

Driver Parameterization

*Doc. No. ENP8090  
Version: 2015-02-18*

**ASKOM®** and **Asix®** are registered trademarks of ASKOM Spółka z o.o., Gliwice. Other brand names, trademarks, and registered trademarks are the property of their respective holders.

All rights reserved including the right of reproduction in whole or in part in any form. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without prior written permission from the ASKOM.

ASKOM sp. z o. o. shall not be liable for any damages arising out of the use of information included in the publication content.

Copyright © 2015, ASKOM Sp. z o. o., Gliwice



ASKOM Sp. z o. o., ul. Józefa Sowińskiego 13, 44-121 Gliwice,  
tel. +48 32 3018100, fax +48 32 3018101,

<http://www.askom.com.pl>, e-mail: office@askom.com.pl

## Table of Contents

CtBACnetIP - driver for BACnet/IP protocol devices.....	4
---	---

# CtBACnetIP - driver for BACnet/IP protocol devices

## Driver Use

The CtBACnetIP driver is designed for data exchange between ASIX and devices with BACnet/ IP protocol interface.

The driver allows to read (and possibly write) the properties of the following BACnet standard objects:

AI	- analog input object,
AO	- analog output object,
AV	- analog value object,
AVG	- averaging object,
BI	- binary input object,
BO	- binary output object,
BV	- binary value object,
CAL	- calendar object,
DEV	- device object,
LP	- life-safety point object,
LZ	- life-safety zone object,
MI	- multi-state input object,
MO	- multi-state output object,
MV	- multi-state value object,
SCH	- schedule object,

and signal events transmitted from devices using telegrams of the UnconfirmedEventNotification and ConfirmedEventNotification type.

The driver has been designed according to ISO 16484-5: 2003.

Parameterization of CtBACnetIP driver is performed with the use of Architect module.

## Declaration of Transmission Channel

Declaration of the transmission channel operating according to the CtBACnetIP driver protocol requires adding a new channel with the following parameters to the *Current data* module:

**Standard** tab:

*Name*: logical name of the transmission channel  
*Driver*: BACnetIP

### **BACnetIP/Channel Parameters:**

<i>IP Address of the Computer</i>	- IP address of the Asix computer card;
<i>IP Address of the Device</i>	- device IP address;
<i>ID of the Device</i>	- device ID;
<i>Event Definitions File</i>	- name of the file containing definitions of events,
<i>Alarms Offset</i>	- offset added to the number of each alarm defined for the device in the event definition file;
<i>Write Priority</i>	- priority of controls for 'commandable' objects (AO, BO, MO, AV, BV, MV); allowable range of the parameter is 1 - 16. Default value is 16 (lowest priority);

<i>Port</i>	- computer port numer used to connect with a PLC; default value: 0 - port is provided by the Windows system;
<i>Alarm Number</i>	- alarm number reported to the alarm system when there is no communication in the channel;
<i>Log File</i>	- log file name; if not specified - messages will be written to the driver log;
<i>Log File Size</i>	- log file size in MB; default value: 10;
<i>Log of Telegrams</i>	- declaration of saving telegrams to the log file.

## EXAMPLE

Below is an example of the KANAL channel declaration. The device ID is set on 348380. Communication is by the network card with the address 10.10.105.1 and the device IP address is 10.10.105.212. Definitions of events generated by the PLC are in the file Dev348380.def. And the offset 100 is added to each alarm reported to the Asix system.

*Name: KANAL  
IP Address of the Computer : 10.10.105.1  
IP Address of the Device: 10.10.105.212  
ID of the Device: 348380  
Event Definitions File: Dev348380.def  
Alarms Offset: 100  
Write Priority: 14*

## Addressing the Variables

The driver supports only the following BACnet standard objects:

AI	- analog input object,
AO	- analog output object,
AV	- analog value object,
AVG	- averaging object,
BI	- binary input object,
BO	- binary output object,
BV	- binary value object,
CAL	- calendar object,
DEV	- device object,
LP	- life-safety point object,
LZ	- life-safety zone object,
MI	- multi-state input object,
MO	- multi-state output object,
MV	- multi-state value object,
SCH	- schedule object.

The syntax of symbolic address which is used for the variables belonging to the CtBACnetIP driver channel is presented below:

*objectType.instance[.property[.item]]*

where:

<i>objectType</i>	- BACnet object name (from the list above);
<i>instance</i>	- object instance number;
<i>property</i>	- object property ID;
<i>item</i>	- element of object property if the object property is an array (eg. element of 'date-;list' property of a calendar).

The syntax for SCHEDULE object is as follows:

*objectType.instance.property.dayNr.item.[VAL | TM]*

where:

<i>dayNr</i>	- day number of the week (Monday 0, Sunday 7),
<i>item</i>	- array element for a given day (indexed from 0),
<i>VAL</i>	- value of the 'value' parameter for the 'item' element,
<i>TM</i>	- value of the 'time' parameter for the 'item' element.

## EXAMPLE

An exemplary variable definitions:

*Name: X01*  
*Description: device\_max\_apdu\_length*  
*Address: DEV.62*  
*Channel: KAN*  
*Elements Count: 1*  
*Sample Rate: 1*  
*Conversion Function: NIC*

*Name: X02*  
*Description: analog-value\_1\_present\_value*  
*Address: AV.1.85*  
*Channel: KAN*  
*Elements Count: 1*  
*Sample Rate: 1*  
*Conversion Function: NIC\_FP*

*Name: X03*  
*Description: analog-value\_1\_object\_name*  
*Address: AV.1.77*  
*Channel: KAN*  
*Elements Count: 50*  
*Sample Rate: 1*  
*Conversion Function: NIC\_TEXT*

*Name: X04*  
*Description: binary-value\_1\_present\_value*  
*Address: BV.1.85*  
*Channel: KAN*  
*Liczba elementów: 1*  
*Sample Rate: 1*  
*Conversion Function: NIC*

*Name: X05*  
*Description: device\_local\_time*  
*Address: DEV.57*  
*Channel: KAN*  
*Elements Count: 20*  
*Sample Rate: 1*  
*Conversion Function: NIC\_TEXT*

*Name: X06*  
*Description: device\_local\_date*  
*Address: DEV.56*  
*Channel: KAN*  
*Elements Count: 20*  
*Sample Rate: 1*  
*Conversion Function: NIC\_TEXT*

*Name: X07*  
*Description: item nr 6 of calendar\_5*

*Address:* CAL.5.23.6  
*Channel:* KAN  
*Elements Count:* 30  
*Sample Rate:* 1  
*Conversion Function:* NIC\_TEXT

*Name:* Y20  
*Description:* 'value' elem. 4 środa SCH nr 1  
*Address:* SCH.1.123.2.4.VAL  
*Channel:* KAN  
*Elements Count:* 1  
*Sample Rate:* 1  
*Conversion Function:* NIC\_BYTE

*Name:* Y21  
*Description:* 'time' elem. 3 wtorek SCH nr 2  
*Address:* SCH.2.123.1.3.TM  
*Channel:* KAN  
*Elements Count:* 30  
*Sample Rate:* 1  
*Conversion Function:* NIC\_TEXT

### CALENDAR Object Variable Values

The CALENDAR object variable values are always expressed in ASCII strings and take one of the following formats:

<b>yyyy-mm-dd</b>	(date)
<b>yyyy-mm-dd - yyyy-mm-dd</b>	(date range)
<b>D.W.M</b>	(day, week, month)

where:

<b>yyyy</b>	- number of year
<b>mm</b>	- number of month
<b>dd</b>	- number of month day
<b>D</b>	- number of week day: 1 - Monday ... 7 - Sunday * - every day of week
<b>W</b>	- number of week in month: 1 - days from 1 to 7 2 - days from 8 to 14 3 - days from 15 to 21 4 - days from 22 to 28 5 - days from 29 to 31 6 - last 6 days of month * - every week of month
<b>M</b>	- number of month: 1 - January ... 12 - December * - every month

#### Notice:

When saving, it is allowed that the control value will be an empty ASCII string - such a control value means a request to delete the item from the calendar. If a deleted item from the calendar is not the last item, the next items will be shifted forward by one position (Calendar does not allow to save undefined items).

## SCHEDULE Object Variable Values

The SCHEDULE object variable values depend on the schedule parameter type.

For the parameter 'time' they are expressed by ASCII string of the following format:

**hh:mm:ss.zz**

where:

<b>hh</b>	- hour
<b>mm</b>	- minute
<b>ss</b>	- second
<b>zz</b>	- hundredths of a second

For the parameter 'value' they are expressed by the number of one of the following types:

bool  
unsigned  
signed  
real  
double

If the parameter 'value' has the value NULL in a PLC, the variable takes the value 0 and receives the status OPC\_QUALITY\_OUT\_OF\_SERVICE.

Notice:

When saving, it is allowed that the control value will be an empty ASCII string for the parameter 'time' - such a control value means a request to delete the item from the schedule (both 'time' as well as 'value'). If a deleted item from the schedule is not the last item, the next items will be shifted forward by one position (SCHEDULE does not allow to save undefined items).

## Signalling Events

The driver supports the 'ConfirmedEventNotification' and 'UnconfirmedEventNotification' telegrams. The way the telegram is interpreted depends on the 'Notify-Type' field value of a received telegram:

- a/ if the field 'Notify-Type' = 'alarm' and the field 'To-State' = 'normal', the driver signals the end of the Asix alarm,
- b/ if the field 'Notify-Type' = 'alarm' and the field 'To-State' != 'normal', the driver signals the start of the Asix alarm,
- c/ Notify-Type = 'event', the driver reports the start and the end of the Asix alarm (telegram content is treated as an event).

The telegram content about an event is converted into Asix system alarms based on event definitions read from an event definition file. The event definition file name is placed in the transmission channel declaration (*Event Definitions File*). No *Event Definitions File* parameter declared in the channel definition causes the telegrams about events in a given channel are ignored by the driver.

The *Alarms Offset* parameter (placed in the channel declaration) is added to each of the Asix alarm read from the event definition.

When using Asix.**Evo**, all the parameters of a given event are transferred to the alarm system.

When using **classical** Asix, only the parameters specified in the definition of a given event in the event definition file are transferred.

The syntax of event definition is as follows:

***asixAlarmNr, BacnetObject, eventType [,par0[,par1[,par2[,par3]]]]***

where:

***asixAlarmNr*** - alarm number in Asix system  
***BacnetObject*** - BACnet ObjectIdentifier and instanceNr of BACnet object that generated an alarm in a PLC,  
***eventType*** - name of BACnet alarm type (values assumed by the enumerator **BACnetEventType**):  
 CHANGE\_OF\_BITSTRING  
 CHANGE\_OF\_STATE  
 CHANGE\_OF\_VALUE  
 FLOATING\_LIMIT  
 OUT\_OF\_RANGE  
***parn*** - number of BACnet alarm parameter the value of which will be transferred to the Asix alarm system; parameter type: WORD or FLOAT (depending on the BACnet parameter number and alarm type); parameters are numerated starting from 0.

Notice:

1. Types of parameters for each alarm type is described in ISO 16484-5:2003 - **BACnetNotificationParameters** type.
2. The total number of bytes occupied by the values of the parameters transferred with an alarm to the Asix alarm system can not exceed 8 bytes (limitation of Asix alarm system). You can then transmit 2 parameters of FLOAT type, 2 parameters of DWORD type, 4 parameters of WORD type and so on.

Examples of event definitions:

10, AV.2, OUT\_OF\_RANGE, 0, 3

The meaning of the above definition:

alarm of OUT\_OF\_RANGE type coming from the object *analog-value instance 2* generates the Asix alarm no. 10. Together with the alarm the following parameter values are passed:

parameter nr 0	- exceeding-value	(FLOAT)
parameter nr 3	- exceeded limit	(FLOAT)

20, AV.3, FLOATING\_LIMIT, 0, 2

The meaning of the above definition:

alarm of FLOATING\_LIMIT type coming from the object *analog-value instance 3* generates the Asix alarm no. 20. Together with the alarm the following parameter values are passed:

parametr nr 0	- reference-value (FLOAT)
parametr nr 2	- setpoint-value (FLOAT)

## Control

For BACnet objects with the property 'priority-array' ('commandable' objects) the controls are performed using the commands transferring a control value and a priority. By default, the controls with priority are performed for the objects: AO, BO, MO, AV, BV, MV.

The priority value is declared separately for each of channels with the use of the parameter *Write Priority*. If the parameter is not used in the channel declaration, the default value shall be 16 (lowest priority).

The priority value defined for the channel is valid for all BACnet objects supported in that channel. That value can be overridden for any object by entering a desired priority value (from 1 to 16) into the virtual property with the ID 512 of the object. To achieve this functionality, you have to create an additional variable the symbolic address of which will appeal to the virtual property with the ID 512 of the object.

## Removing Controls

For the BACnet objects with the property 'priority-array' ('commandable' objects) you can remove an active control (placed in the array 'priority-array'). Removal of control consists in writing the value NULL into the array 'priority-array' at the position corresponding to the property with which controls of the object are performed.

To achieve the above functionality you need to create an additional variable the symbolic address of which refers to the virtual property with the ID 513 of the object. The operation of removal of control consists in writing any control value to a such variable.

## Driver Parameters

The driver configuration is defined in the *Current Data* module, in the channel operating according to CtBACnetIP driver, on the tabs **BACnetIP / Driver parameters**.

### **Log File**

Meaning: text log file to which driver status messages are stored; it is used for diagnostic purposes.

Parameter:

*log\_file\_name* - file name

Default value: file is not created

### **Events Log File**

Meaning: text log file to which messages about events transferred from PLCs are stored; it is used for diagnostic purposes.

Parameter:

*log\_file\_name* - file name

Default value: file is not created

### **Log File Size**

Meaning: option is used to determine the log file size defined by the *Log file* and *Event Log File* options.

Parameter:

*number* - log file size in MB

Default value: 10

**Log of Telegrams**

Meaning:

option allows writing the contents of messages transmitted between the driver and the controllers to the *Log File* (declared using the *Log File* option). The option should only be used during the Asix system start-up. The option value can be overridden by using the same option but in channel parameters.

Parameter:

YES / NO

Default value: by default, the option is disabled

 **Event Definitions Path**

Meaning:

option specifies the full path to the directory where the driver will search for event definition files declared in the channel declaration.

Parameter:

*path*

- path to the directory

Default value: no path declared

**EXAMPLE**

An exemplary declaration of driver parameters:

 *Log File*

d:\tmp\ CtBACnetIP\bacnet.log

 *Events Log File*

d:\tmp\ CtBACnetIP\events.log

 *Log File Size*

30

 *Log of Telegrams*

TAK

 *Event Definitions Path*

d:\tmp\ EventsDef