



by Schneider Electric

BACnet – ModBUS Gateway Library User Guide

Publisher: HVAC Solution Center, Alpago (BL) – Italy

Authors: Federico Marcassa, Pierpaolo Armeli

Doc. Version: v0.9





Table of Contents

1. BACnet – ModBUS Gateway Library	6
1.1. Overview	6
1.2. Description	6
1.3. Function Blocks and Programs	7
1.3.1. List of the FBs and Progs	7
1.3.2. BACnet_ModBUS_SL_AV	7
1.3.3. BACnet_ModBUS_SL_BV	8
1.3.4. BACnet_ModBUS_SL_MV	9
1.3.5. BACnet_ModBUS_SL_AI	9
1.3.6. BACnet_ModBUS_SL_BI	10
1.3.7. BACnet_ModBUS_SL_MSI	11
1.3.8. BACnet_ModBUS_SL_Sc_REAL	12
2. Publisher's Info	13



Safety Information and Good Practices

Before You Begin

General

The products specified in this document have been tested under actual service conditions. Of course, your specific application requirements may be different from those assumed for this and any related examples described herein. In that case, you will have to adapt the information provided in this and other related documents to your particular needs. To do so, you will need to consult the specific product documentation of the hardware and/or software components that you may add or substitute for any examples specified in this documentation. Pay particular attention and conform to any safety information, different electrical requirements and normative standards that would apply to your adaptation.

© 2018 Eliwell Controls Srl. All rights reserved.

WARNING

REGULATORY INCOMPATIBILITY

Be sure that all equipment applied and systems designed comply with all applicable local, regional and national regulations and standards

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Note

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Eliwell Controls Srl for any consequences arising out of the use of this material. A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved. Failure to observe this information can result in injury or equipment damage.

The use and application of the information contained herein require expertise in the design and programming of automated control systems. Only the user or integrator can be aware of all the conditions and factors present during installation and setup, operation, and maintenance of the machine or process, and can therefore determine the automation and associated equipment and the related safeties and interlocks which can be effectively and properly used. When selecting automation and control equipment, and any other related equipment or software, for a particular application, the user or integrator must also consider any applicable local, regional or national standards and/or regulations.

Some of the major software functions and/or hardware components used in the proposed architectures and examples described in this document cannot be substituted without significantly compromising the performance of your application. Further, any such substitutions or alterations may completely invalidate any proposed architectures, descriptions, examples, instructions, wiring diagrams and/or compatibilities between the various hardware components and software functions specified herein and in related documentation. You must be aware of the consequences of any modifications, additions or substitutions.



A residual risk, as defined by EN/ISO 12100-1, Article 5, will remain if:

- it is necessary to modify the recommended logic and if the added or modified components are not properly integrated in the control circuit;
- you do not follow the required standards applicable to the operation of the machine, or if the adjustments to and the maintenance of the machine are not properly made (it is essential to strictly follow the prescribed machine maintenance schedule);
- the devices connected to any safety outputs do not have mechanically-linked contacts.

CAUTION

EQUIPMENT INCOMPATIBILITY

Read and thoroughly understand all device and software documentation before attempting any component substitutions or other changes related to the application examples provided in the document

Failure to follow these instructions can result in injury, or equipment damage.

Start-Up and Test

Before using electrical control and automation equipment after design and installation, the application and associated functional safety system must be subjected to a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such testing be made and that enough time is allowed to perform complete and satisfactory testing.

CAUTION

EQUIPMENT OPERATION HAZARD

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices
- Remove tools, meters, and debris from equipment.

Failure to follow these instructions can result in injury, or equipment damage.

Verify that the completed system, including the functional safety system, is free from all short circuits and grounds, except those grounds installed according to local regulations. If high-potential voltage testing is necessary, follow the recommendations in equipment documentation to help prevent injury or equipment damage.



Operations and Adjustments

Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly installed and operated.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the hands and other parts of the body are free to enter the pinch points or other hazardous areas where serious injury can occur. Software products alone cannot protect an operator from injury. For this reason, the software cannot be substituted for or take the place of point-of-operation protection.

WARNING

UNGUARDED MACHINERY CAN CAUSE SERIOUS INJURY

- Do not use this software and related automation equipment on equipment which does not have point-of -operation protection.
- Do not reach into machinery during operation.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the examples and implementations suggested herein. It is sometimes possible to adjust the equipment incorrectly and this produce unsatisfactory or unsafe operation. Always use the manufacturer instructions as a guide to functional adjustments. Personnel who have access to these adjustments must be familiar with the equipment manufacturer instructions and the machinery used with the electrical equipment.

Only those operational adjustments actually required by the machine operator should be accessible to the operator. Access to other controls should be restricted to help prevent unauthorized changes in operating characteristics.



BACnet – ModBUS Gateway Library

1.1. Overview

The current documentation (v0.9) is related to the **BACnet ModBUS Gateway library v.0.9.1**. The library is tested/supported on **FreeStudio 3.9.1** and greater versions.

1.2. Description

The *BACnet ModBUS Gateway* library contains function blocks, which are additional to the ones included in the *BACnet Addons* library, that can specifically be used to implement the BACnet to ModBUS SL Gateway functionalities. In particular, they manage the synchronization of different types of values between the BACnet and the ModBUS side.

The **BACnet Addons** and the **ModBUS Addons libraries are prerequisites** for the *BACnet ModBUS Gateway* library to be used.

The **key functionalities** of the current library version are:

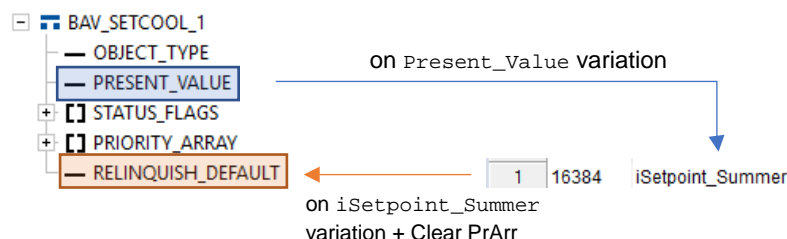
- Link contiguous registers / variables on a Slave with contiguous BACnet Objects (up to 4 registers). The supported types are:
 - INT variable target container link with an Analog Value;
 - BOOL variable target container link with a Binary Value;
- Link a single register on a Slave (UINT variable target container) with a Multi State Value.
- Set the Present Value of Analog, Binary and Multi State Input BACnet Objects, based on the ModBUS slave status.
- Write the REAL output of a BACnet Schedule into a REAL variable of a ModBUS Slave.

The **targets** supported by this library version are:

- **Advance** with BIOS 596.6 and greater;
- **Advance 4DIN** with BIOS 668.6 and greater.

The reference about how to implement the library is the BACnet to ModBUS SL Gateway Application Note.

The **priorities between the BACnet and the ModBUS side** are managed in the same way the *usiMode 2* of the BACnet Addons library FBs acts: the BACnet_ModBUS_SL_XX function blocks write the value of the slave register, on variation, on the Relinquish Default and make a one-time Priority Array clear. On variation of the Present Value, this is written on the slave register. This way of working is synthetized in the schematic below.





1.3. Function Blocks and Programs

In this chapter, the function blocks and programs that are part of the library are described, and for each one an example of use is reported.

1.3.1. List of the FBs and Progs

The **function blocks (FB)** included in the library are the following:

Type	Name	Description
FB	BACnet_ModBUS_SL_AV	Link a register on the slave (INT target container) with an Analog Value (up to 4 contiguous registers/AVs)
FB	BACnet_ModBUS_SL_BV	Link a register on the slave (BOOL target container) with a Binary Value (up to 4 contiguous registers/AVs)
FB	BACnet_ModBUS_SL_MV	Link a register on the slave (UINT target container) with a Multi State Value
FB	BACnet_ModBUS_SL_AI	This function block sets the Present_Value of an Analog Input BACnet Object based on the ModBUS slave status.
FB	BACnet_ModBUS_SL_BI	This function block sets the Present_Value of a Binary Input BACnet Object based on the ModBUS slave status.
FB	BACnet_ModBUS_SL_MSI	This function block sets the Present_Value of a Multi State Input BACnet Object based on the ModBUS slave status.
FB	BACnet_ModBUS_SL_Sc_REAL	Write the REAL output of a BACnet Schedule into a REAL variable of a Slave.

1.3.2. BACnet_ModBUS_SL_AV

This function block allows to read contiguous registers (max 4) on the Slave and links them to contiguous AVs on the BACnet side. The contiguity Slave side and BACnet Objs side has to be coincident. A different ordering correlation of the registers/objects between the two sides is not supported.

The interaction between the BACnet side and the PLC side follows what in the *link* FBs (e.g. BACnet_RelDef_AV_link_iEE) of the BACnet Addons library is called *usiMode* = 2. This means that, on every value variation on the ModBUS side, the Priority Array is cleared to regain priority over the BACnet side. The ModBUS value is updated based on the Present Value.

Input

Name	Type	Description
xEn	BOOL	Used to enable/disable the BACnet/ModBUS Gateway Functionality
i_sysBACnet_AV_FirstIndex	INT	Index of the first AV to link. It is considered from the list of the defined AVs per Slave. It starts from 0.
uiContiguousRegsN	UINT	Number of contiguous registers/AVs to link. Max 4 registers supported.
ptr_strKA_Slave	@strModbusRTUslave	strKeepAlive input type related to the Slave
uiADR_Slave_FirstVariable	UINT	Address of the first variable to link on the Slave.
rScaler	REAL	Scaling factor. Default is 1.0. It cannot be 0.0.
xRelDef_E2	BOOL	RelDef in EEPROM / not in EEPROM.



xRestoreRelDef_OFF	BOOL	If TRUE, the RelDef value is written on the Slave on a xEn transition from TRUE to FALSE. Default is FALSE.
--------------------	------	-------------------------------------------------------------------------------------------------------------

Output

Name	Type	Description
wStatus	WORD	Bit 0: Config Error

1.3.3. BACnet_ModBUS_SL_BV

This function block allows to read contiguous registers (max 4) on the Slave and links them to contiguous BVs on the BACnet side. The contiguity Slave side and BACnet Objects side has to be coincident. A different ordering correlation of the registers/objects between the two sides is not supported.

The interaction between the BACnet side and the PLC side follows what in the *link* FBs (e.g. *BACnet_RelDef_AV_link_iEE*) of the BACnet Addons library is called *usiMode* = 2. This means that, on every value variation on the ModBUS side, the Priority Array is cleared to regain priority over the BACnet side. The ModBUS value is updated based on the Present Value.

Input

Name	Type	Description
xEn	BOOL	Used to enable/disable the BACnet/ModBUS Gateway Functionality
i_sysBACnet_BV_FirstIndex	INT	Index of the first AV to link. It is considered from the list of the defined AVs per Slave. It starts from 0.
uiContiguousRegsN	UINT	Number of contiguous registers/AVs to link. Max 4 registers supported.
ptr_strKA_Slave	@strModbusRTUslave	strKeepAlive input type related to the Slave
uiADR_Slave_FirstVariable	UINT	Address of the first variable to link on the Slave.
xRelDef_E2	BOOL	RelDef in EEPROM / not in EEPROM.
xRestoreRelDef_OFF	BOOL	If TRUE, the RelDef value is written on the Slave on a xEn transition from TRUE to FALSE. Default is FALSE.

Output

Name	Type	Description
wStatus	WORD	Bit 0: Config Error



1.3.4. BACnet_ModBUS_SL_MV

This function block allows to read a register on the Slave and links it to a MV on the BACnet side.

The interaction between the BACnet side and the PLC side follows what in the *link* FBs (e.g. BACnet_RelDef_AV_link_iEE) of the BACnet Addons library is called *usiMode* = 2. This means that, on every value variation on the ModBUS side, the Priority Array is cleared to regain priority over the BACnet side. The ModBUS value is updated based on the Present Value.

Input

Name	Type	Description
xEn	BOOL	Used to enable/disable the BACnet/ModBUS Gateway Functionality
i_sysBACnet_MV_Index	INT	Index of the first AV to link. It is considered from the list of the defined AVs per Slave. It starts from 0.
ptr_strKA_Slave	@strModbusRTUslave	strKeepAlive input type related to the Slave
uiADR_Slave_Variable	UINT	Address of the first variable to link on the Slave.
xRelDef_E2	BOOL	RelDef in EEPROM / not in EEPROM.
iOffset	INT	INT offset that is summed to the input value
xRestoreRelDef_OFF	BOOL	If TRUE, the RelDef value is written on the Slave on a xEn transition from TRUE to FALSE. Default is FALSE.

Output

Name	Type	Description
wStatus	WORD	Bit 0: Config Error

1.3.5. BACnet_ModBUS_SL_AI

This FB acts as BACnet_AI_SetPV FB of the BACnet Addons library, with the following additional feature: the slave status is taken into account to set the BACnet Object OUT_OF_SERVICE and Reliability values.

Refert to the BACnet_AI_SetPV description for information about the inputs, the outpus and how the FB works.

The ptr_strKA_Slave input is COMPULSORY.

Input

Name	Type	Description
xEn	BOOL	
ptrAI_BACnet	@BACNET_ANALOG_VALUE	Pointer to the Analog Input BACnet Object (use ADR function).
iValue	INT	INT value to be written in PV. If iValue = 32767, it is disabled and rValue is used to set the PV instead. It is multiplied by rScaler to calculate the PV.
rValue	REAL	REAL Value to be written in PV. Enabled only if iValue = 32767. It is multiplied by rScaler to calculate the PV.



rScaler	REAL	Scaling factor. Default is 1.0. It cannot be 0.0. NOTE: it is multiplied by the iValue or the rValue to determine the PV.
xLocalPV_Upd	BOOL	Forces the local PV Update of the interested BACnet variable. By default, it is set to FALSE.
ptr_strKA_Slave	@strModbusRTUslave	strKeepAlive input type related to the Slave

Output

Name	Type	Description
rPV	REAL	It returns the Present Value.
wStatus	WORD	Bit 0: OK (IEC code operating properly). Bit 1: Out Of Service. Bit 2: Configuration Error.

1.3.6. BACnet_ModBUS_SL_BI

This FB acts as BACnet_BI_SetPV FB of the BACnet Addons library, with the following additional feature: the slave status is taken into account to set the BACnet Object OUT_OF_SERVICE and Reliability values.

Refert to the BACnet_BI_SetPV description for information about the inputs, the outpus and how the FB works.

The ptr_strKA_Slave input is COMPULSORY.

Input

Name	Type	Description
xEn	BOOL	
ptrBI_BACnet	@BACNET_BINARY_VALUE	Binary Value Object Address (use ADR function)
xValue	BOOL	EEPROM Parameter Address (use ADR function)
xLocalPV_Upd	BOOL	Forces the local PV Update of the interested BACnet variable. By default, it is set to FALSE.
xClearAlways	BOOL	If TRUE, the PrArr is cleared at each execution, if required, in order for the Present_Value to reflect the Input Value. Default if FALSE.
ptr_strKA_Slave	@strModbusRTUslave	strKeepAlive input type related to the Slave

Output

Name	Type	Description
xPV_current	BOOL	BOOL Present Value
wStatus	WORD	Bit 0: OK (IEC code operating properly). Bit 1: Out Of Service. Bit 2: Configuration Error. Bit 4: Priority Active (override from supervisor).



1.3.7. BACnet_ModBUS_SL_MSI

This FB acts as BACnet_MSI_usiSetPV FB of the BACnet Addons library, with the following additional feature: the slave status is taken into account to set the BACnet Object OUT_OF_SERVICE and Reliability values.

Refert to the BACnet_MSI_usiSetPV description for information about the inputs, the outpus and how the FB works.

The ptr_strKA_Slave input is COMPULSORY.

Use a TO_USINT block in case you are using a different data type in your application for the input value.

Input

Name	Type	Description
xEn	BOOL	
ptrMSI_BACnet	@BACNET_MULTI_STATE_VALUE	Pointer to the Multi State Input BACnet Object (use ADR function).
usiValue	USINT	USINT Value to be written in PV, scaled by summing the iOffset to it. Its value must be such that the sum diValue + iOffset is in the range 1... Number_Of_States.
iOffset	INT	INT offset that is summed to the input value.
iWrongInputValue	INT	Value that is written on the Present Value in case of wrong input value (outside the allowed range of the object). Its value MUST be in the range 1 ... Number_Of_States.
xLocalPV_Upd	BOOL	Forces the local PV Update of the interested BACnet variable. By default, it is set to FALSE.
ptr_strKA_Slave	@strModbusRTUslave	strKeepAlive input type related to the Slave

Output

Name	Type	Description
udiPV	UDINT	It returns the Present Value.
wStatus	WORD	Bit 0: OK (IEC code operating properly). Bit 1: Out Of Service. Bit 2: Configuration Error.



1.3.8. *BACnet_ModBUS_SL_Sc_REAL*

This FB allows to write the REAL output of a BACnet Schedule into a REAL variable of a Slave (two registers are written).

Input

Name	Type	Description
xEn	BOOL	Used to enable/disable the BACnet/ModBUS Gateway Functionality
i_sysBACnet_Sc_Index	INT	Index of the first AV to link. It is considered from the list of the defined AVs per Slave. It starts from 0.
ptr_strKA_Slave	@strModbusRTUslave	strKeepAlive input type related to the Slave
uiADR_Slave_Variable	UINT	Address of the remote variable to link on the Slave.

Output

Name	Type	Description
wStatus	WORD	Bit 0: Config Error



Publisher's Info

The **publisher** of this library is the **HVAC Solution Center** based in Alpago (BL), Italy.
Its main goal is to work on machine architecture solutions, software libraries and application notes.
Its **members** are:

- **Pierpaolo Armeli**
pierpaolo.armeli@schneider-electric.com
- **Federico Marcassa**
federico.marcassa@schneider-electric.com