

PRELIMINARY

Babel Buster 485XL User Guide



**Control Solutions, Inc.
2179 Fourth Street
White Bear Lake, MN 55110**

www.csimn.com

Control Solutions, Inc.
Babel Buster® 485XL
LonWorks to BACnet MS/TP Gateway

User Guide
Rev. 1.01 • Sept. 2007

This manual corresponds with firmware version 1.01.

IMPORTANT SAFETY CONSIDERATIONS:

Proper system design is required for reliable and safe operation of distributed control systems incorporating Babel Buster series gateways and other such devices. It is extremely important for the user and system designer to consider the effects of loss of power, loss of communications, and failure of components in the design of any monitoring or control application. This is especially important where the potential for property damage, personal injury, or loss of life may exist. By using the Babel Buster series gateway or any other Control Solutions, Inc., product, the user has agreed to assume all risk and responsibility for proper system design as well as any consequence for improper system design.

© 2007 Control Solutions, Inc.

BACnet® is a registered trademark of American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE). Babel Buster® is a registered trademark of Control Solutions, Inc., Minnesota, USA. All other trademarks mentioned in this document are the property of their respective owners. Information in this document is subject to change without notice and does not represent a commitment on the part of Control Solutions, Inc. This document is provided “as is,” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Control Solutions may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time. This product could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the publication.

Babel Buster 485XL Overview

The Babel Buster 485XL BACnet Gateway supports connecting a LonWorks device to a BACnet MS/TP network. It does not support connecting a BACnet device to a LonWorks network. The 485XL gateway is a polling device only, and is not bound to other LonWorks devices. The typical LonWorks network management is not required. It is assumed that putting a LonWorks device on a BACnet network means there is no LonWorks network to manage. A fixed number of LonWorks devices is supported. The 485XL gateway is a BACnet slave (server) only, and does not include a BACnet client.

There is a one to one mapping of LonWorks network variables to BACnet objects, except when the LonWorks variable is a structure variable containing multiple data values. In this case, the single network variable translates to multiple BACnet objects.

Use an Analog Input object to read a LonWorks network variable (NV) defined as a Network Variable Output (NVO) at the remote device. If the Analog Input object is placed out of service, it can be used to write to a LonWorks NV defined as a LonWorks Network Variable Input (NVI) at the remote device.

Use an Analog Value object to provide the most effective write access to a NVI. A write to Present Value requires a priority level, and this value is placed into the object's priority array. The result of processing the priority array according to BACnet standard will be written to the NVI any time the Present Value is written. To reiterate, the highest priority value is written to the NVI, and the result of processing the priority array is returned as the Present Value read from the BACnet object.

Use a Binary Input to read an NVO for which the result is a simple binary state. The state will be 1 if the value of the NVO is nonzero, regardless of the size of the NVO.

Use a Binary Value object works the same as the Analog Value object, except that it applies to a single bit or on/off state.

Communication problems between the gateway and LonWorks device will be logged and reported via the Reliability property of the affected object. Vendor specific Reliability codes indicate "no response", etc. If the object is in fault as a result of communication problems, the Status Flags will indicate Fault, and the Event State will also indicate Fault.

Babel Buster 485XL Capacities:

LonWorks devices: 20

Analog Input objects: 100

Analog Value objects: 5

Binary Input objects: 40

Binary Value objects: 10

Initial Setup or Setup Recovery

The gateway's station ID, device ID, and baud rate for the BACnet MS/TP port are set by writing properties to the device object. This of course presumes that you know what they are initially. By holding the SERVICE button down (see note that follows), you force the station ID to one (1), device ID to one (1), and baud rate to 9600. This is referred to as "init mode". It is assumed that you will make these one-time device settings while the gateway is not connected to the rest of your BACnet network, and is instead connected to a PC running BACbeat or Control Solutions proprietary Babel Buster BAConfig tool.

Note: The init mode button on the 485XL functions differently than the 485XM. On the 485XM, you hold the button down while powering up the gateway. The SERVICE button on the 485XL serves two functions, one being the service button for the Neuron Chip in the gateway. The secondary function is to cause entry into init mode. You must allow the 485XL to boot up (red LED inside goes off), then hold the service button down until the red LED comes back on.

General Discussion of Object Behavior

Writeable Input Objects

Input objects are normally read-only. They become write-only objects if placed out of service. Normal operation performs periodic LonWorks polls to read the configured Network Variable (NV), and place the contents in the Present Value buffer of the BACnet object.

Writes to an Input object will normally not be accepted. However, if the object is placed out of service, you may now write the Present Value. The value written will be periodically written to the configured LonWorks NV. While out of service, reading the BACnet Present Value will simply echo what was written. To determine if the contents were successfully written to the LonWorks device, you need to check the Reliability property.

When transitioning an Input object to or from out-of-service, the object may be flagged as having an NV failure. Since you cannot write to an NVO at the remote device, any attempt to do so will cause a failure until the correct direction is established for the object.

Value Objects

Value objects are read-write objects, but the LonWorks network variable is treated as read or write, but never both at the same time. Value objects provide full priority array functionality in controlling the contents of the LonWorks NV. A priority level must be included when writing the Present Value of a Value object. The highest priority level not yet relinquished will be written to the LonWorks NV each time the Present Value is updated. If all priorities are relinquished, the relinquish default value is written to the LonWorks NV when the last level is relinquished. The result of processing the priority array will be placed in the value object's Present Value buffer for subsequent BACnet

PRELIMINARY

reading. If the object is placed out of service, the Present Value may be written directly, and the result is a direct write to the LonWorks NV (which must be an NVI at the remote device).

Upon transitioning a Value object to out-of-service, the most recent priority based update will remain in effect in the LonWorks NV until the next time Present Value is written. The new value will be written to LonWorks at that time. While out-of-service, values may continue to be written to the priority array by writing Present Value with a priority. You must write Present Value with no priority to do a direct write while out-of-service (writes with priority will simply be placed in the priority array without writing to LonWorks). Upon transition to in-service, the highest priority non-relinquished value found in the priority array will be written to the LonWorks NVI at the remote device.

The properties that are recognized for each type of object supported in Babel Buster 485XL are listed below, by object type. The BACnet property name is given in the left column, and its property ID value is in parenthesis after the name. All listed properties are readable, and those having (W) after the property ID are writeable. The property type is listed in the right column, including applicable enumeration values.

ANALOG INPUT

Object_Identifier (75)	BACnetObjectIdentifier
Object_Name (77)	CharacterString “Analog Input <i>n</i> ”
Object_Type (79)	BACnetObjectType ENUMERATED: analog-input (0) analog-value (2) binary-input (3) binary-value (5) device (8)
Present_Value (85) (W)	REAL (no index, no priority) (writeable only when out of service)
Status_Flags (111)	BACnetStatusFlags BIT STRING: fault(1), out-of-service(3)
Event_State (36)	BACnetEventState ENUMERATED: normal(0), fault(1)
Reliability (103)	BACnetReliability ENUMERATED: normal(0) <i>Vendor specific:</i> no response, device query fail (64) no response, NV fail (67) configuration property fault (80)
Out_Of_Service (81) (W)	BOOLEAN
Units (117)	BACnetEngineeringUnits
<u><i>Vendor Specific Object Properties:</i></u>	
Object_Map_Use (701) (W)	BIT STRING: (1) map enabled (2) NV direction, 0=NVI (read remote NVO), 1=NVO (write remote NVI) (3) grouped – is field in struct (4) lock – this field triggers struct update
Special_Service_Request (702) (W)	BIT STRING: (1) default to out of service at power-up (2) invoke write at power-up if writeable object (3) set BACnet Present Value to default if

PRELIMINARY

LonWorks communications fails

Node Index (703) (W)	ENUMERATED 1..MAX_NODE Index into device table for this object
NV Index (704) (W)	Unsigned NV index within device
SNVT Index (705) (W)	OCTET STRING (size 4) (1) SNVT index, 0=raw (2) NVT_CAT if raw (3) Byte offset if raw (4) Bit offset if raw
Scale Array (706) (W)	Integer Index: 1..3 (1) Scale A (2) Scale B (3) Scale C
Default_Value (707) (W)	REAL
Reference_String (708) (W)	CharacterString

ANALOG VALUE

Object_Identifier (75)	BACnetObjectIdentifier
Object_Name (77)	CharacterString “Analog Value <i>n</i> ”
Object_Type (79)	BACnetObjectType ENUMERATED: analog-input (0) analog-value (2) binary-input (3) binary-value (5) device (8)
Present_Value (85) (W)	REAL (no index)
Status_Flags (111)	BACnetStatusFlags BIT STRING: fault(1), out-of-service(3)
Event_State (36)	BACnetEventState ENUMERATED: normal(0), fault(1)
Reliability (103)	BACnetReliability ENUMERATED: normal(0) <i>Vendor specific:</i> no response, device query fail (64) no response, NV fail (67) configuration property fault (80)
Out_Of_Service (81) (W)	BOOLEAN
Priority_Array (87)	BACnetPriorityArray SEQUENCE SIZE (16) OF BACnetPriorityValue REAL (each element)
Relinquish_Default (104) (W)	REAL
Units (117)	BACnetEngineeringUnits
<u><i>Vendor Specific Object Properties:</i></u>	
Object_Map_Use (701) (W)	BIT STRING: (1) map enabled (2) NV direction, 0=NVI (read remote NVO), 1=NVO (write remote NVI) (3) grouped – is field in struct (4) lock – this field triggers struct update

PRELIMINARY

Special_Service_Request (702) (W)	BIT STRING: (1) default to out of service at power-up (2) invoke write at power-up if writeable object (3) set BACnet Present Value to default if LonWorks communications fails
Node Index (703) (W)	ENUMERATED 1..MAX_NODE Index into device table for this object
NV Index (704) (W)	Unsigned NV index within device
SNVT Index (705) (W)	OCTET STRING (size 4) (1) SNVT index, 0=raw (2) NVT_CAT if raw (3) Byte offset if raw (4) Bit offset if raw
Scale Array (706) (W)	Integer Index: 1..3 (1) Scale A (2) Scale B (3) Scale C
Default_Value (707) (W)	REAL
Reference_String (708) (W)	CharacterString

BINARY INPUT

Object_Identifier (75)	BACnetObjectIdentifier
Object_Name (77)	CharacterString “Binary Input <i>n</i> ”
Object_Type (79)	BACnetObjectType ENUMERATED: analog-input (0) analog-value (2) binary-input (3) binary-value (5) device (8)
Present_Value (85) (W)	BOOLEAN (no index, no priority) (writeable only when out of service)
Status_Flags (111)	BACnetStatusFlags BIT STRING: fault(1), out-of-service(3)
Event_State (36)	BACnetEventState ENUMERATED: normal(0), fault(1)
Reliability (103)	BACnetReliability ENUMERATED: normal(0) <i>Vendor specific:</i> no response, device query fail (64) no response, NV fail (67) configuration property fault (80)
Out_Of_Service (81) (W)	BOOLEAN
Polarity (84)	BACnetPolarity ENUMERATED: normal(0)
Units (117)	BACnetEngineeringUnits
<u><i>Vendor Specific Object Properties:</i></u>	
Object_Map_Use (701) (W)	BIT STRING: (1) map enabled (2) NV direction, 0=NVI (read remote NVO), 1=NVO (write remote NVI) (3) grouped – is field in struct (4) lock – this field triggers struct update
Special_Service_Request (702) (W)	BIT STRING:

PRELIMINARY

- (1) default to out of service at power-up
- (2) invoke write at power-up if writeable object
- (3) set BACnet Present Value to default if LonWorks communications fails

Node Index (703) (W)	ENUMERATED 1..MAX_NODE Index into device table for this object
NV Index (704) (W)	Unsigned NV index within device
SNVT Index (705) (W)	OCTET STRING (size 4) (1) SNVT index, 0=raw (2) NVT_CAT if raw (3) Byte offset if raw (4) Bit offset if raw
Scale Array (706) (W)	Integer Index: 1..3 (1) Scale A (2) Scale B (3) Scale C
Default_Value (707) (W)	REAL
Reference_String (708) (W)	CharacterString

BINARY VALUE

Object_Identifier (75)	BACnetObjectIdentifier
Object_Name (77)	CharacterString “Binary Value <i>n</i> ”
Object_Type (79)	BACnetObjectType ENUMERATED: analog-input (0) analog-value (2) binary-input (3) binary-value (5) device (8)
Present_Value (85) (W)	BOOLEAN (no index)
Status_Flags (111)	BACnetStatusFlags BIT STRING: fault(1), out-of-service(3)
Event_State (36)	BACnetEventState ENUMERATED: normal(0), fault(1)
Reliability (103)	BACnetReliability ENUMERATED: normal(0) <i>Vendor specific:</i> no response, device query fail (64) no response, NV fail (67) configuration property fault (80)
Out_Of_Service (81) (W)	BOOLEAN
Priority_Array (87)	BACnetPriorityArray SEQUENCE SIZE (16) OF BACnetPriorityValue BOOLEAN (each element)
Relinquish_Default (104) (W)	BOOLEAN
Units (117)	BACnetEngineeringUnits
<u><i>Vendor Specific Object Properties:</i></u>	
Object_Map_Use (701) (W)	BIT STRING: (1) map enabled (2) NV direction, 0=NVI (read remote NVO), 1=NVO (write remote NVI) (3) grouped – is field in struct (4) lock – this field triggers struct update

PRELIMINARY

Special_Service_Request (702) (W)	BIT STRING: (1) default to out of service at power-up (2) invoke write at power-up if writeable object (3) set BACnet Present Value to default if LonWorks communications fails
Node Index (703) (W)	ENUMERATED 1..MAX_NODE Index into device table for this object
NV Index (704) (W)	Unsigned NV index within device
SNVT Index (705) (W)	OCTET STRING (size 4) (1) SNVT index, 0=raw (2) NVT_CAT if raw (3) Byte offset if raw (4) Bit offset if raw
Scale Array (706) (W)	Integer Index: 1..3 (1) Scale A (2) Scale B (3) Scale C
Default_Value (707) (W)	REAL
Reference_String (708) (W)	CharacterString

DEVICE

Object_Identifier (75)	BACnetObjectIdentifier
Object_Name (77)	CharacterString
Object_Type (79)	BACnetObjectType
System_Status (112)	BACnetDeviceStatus
Vendor_Name (121)	CharacterString
Vendor_Identifier (120)	Unsigned16 (should always return 208)
Model_Name (70)	CharacterString
Firmware_Revision (44)	CharacterString
Application_Software_Version (12)	CharacterString
Protocol_Version (98)	Unsigned
Protocol_Revision (139)	Unsigned
Protocol_Services_Supported (97)	BACnetServicesSupported
Protocol_Object_Types_Supported (96)	BACnetObjectTypesSupported
Object_List (76)	BACnetARRAY[N] of BACnetObjectIdentifier
Max_APDU_Length_Accepted (62)	Unsigned
Segmentation_Supported (107)	BACnetSegmentation
APDU_Timeout (11)	Unsigned
Number_Of_APDU_Retries (73)	Unsigned
Device_Address_Binding (30)	List of BACnetAddressBinding
Database_Revision (155)	Unsigned
<u><i>Vendor Specific Object Properties:</i></u>	
Domain Length (751) (W)	ENUMERATED: 0, 1, 3, or 6 (byte length)

PRELIMINARY

Domain ID (752) (W)	OCTET STRING (size 6)
Subnet (753) (W)	Unsigned (Note: Node nos. assigned from 1) Index: 1..127 (Writing subnet initiates update of entire domain table entry in Neuron Chip.)
Neuron ID (754) (W)	OCTET STRING (size 6) Index: 1..MAX_NODE
Program ID (755) (W)	OCTET STRING (size 8) Index: 1..MAX_NODE
Install Node (756) (W)	ENUMERATED 1..MAX_NODE Selects this node for service pin install if !=0. Filtered by SPID if SPID is nonzero. Note: This results in altering the domain table of the installed node.
Wink Node (757) (W)	ENUMERATED 1..MAX_NODE
Query Node (758) (W)	BOOLEAN Index: 1..MAX_NODE Reads result of last set domain request Write invokes new attempt to set domain for node
Query NV Status (759) (W)	ENUMERATED 1..MAX_OBJECT Forces update of object reliability
Initiate Node Discovery (760) (W)	ENUMERATED Values: 0=no action, 1=initiate Discovery looks for SPID in device table that does not already have NID assigned. Use property 755 to tell gateway what type of device to search for. Use property 754 to zero out respective NID if not already zero. SPID wild card is 0xFF in first byte. DO NOT use if network interface such as i.LON 100 is attached. The first responding undiscovered node will have its domain table altered.

PRELIMINARY

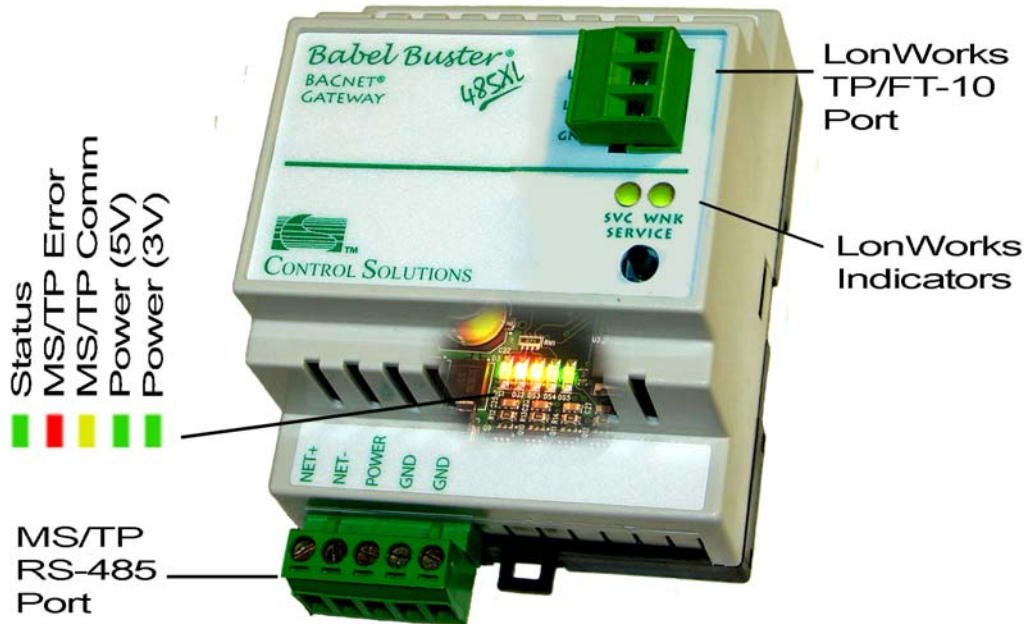
MSTP_Port_Baud_Rate (1201) (W)	ENUMERATED: 9600(0), 19200(1), 38400(2), 76800(3)
Station_ID (1202) (W)	Unsigned
Device_ID (1203) (W)	Unsigned
Max Masters (1204) (W)	Unsigned
Reinit_Password (1205) (WO)	CharacterString Note: This property is write-only, and is only writeable in INIT mode.

NOTE: Changes to port settings (properties 1201-1205) will take effect only upon issuing a Reinit Device command to the device. Upon receiving the Reinit Device, the gateway will commit these changes to EEPROM, and then reset itself so the new settings can take effect.

NOTE: The port settings will only be re-written if a WARM START command is issued. A cold start will only reset the device (same as power-up reset).

INDICATORS

The LED indicators showing LonWorks activity are on the front of the gateway. Additional diagnostic LEDs are inside the unit and may be viewed through the vent slots on the lower side.



LonWorks Indicators

SVC – Green, remains on while service button is pressed, and follows standard Service Pin LED behavior as defined for LonWorks Neuron Chips otherwise.

SVC – Green/Yellow. (a) Alternates green/yellow rapidly for 10 seconds if a wink command is received by the gateway from the LonWorks network. (b) Flashes green when a successful NV update or NV fetch response is received; (c) Flashes yellow when a failed NV update or NV fetch attempt is detected; (d) is off if no activity.

BACnet and System Indicators

Status – Green, flashes each time the MS/TP token is passed. This is effectively a BACnet activity indicator.

MS/TP Error – Red, flashes each time a request is received for which an error is returned to the requesting client.

MS/TP Comm – Yellow, flashes each time a request is received for which a good response is returned to the requesting client.

Power – Green, remains on if power is present. The gateway base board has two power supplies, a 5V and a 3.3V supply. Only the 5V supply is used in this application, but the 3.3V indicator will still be present.

WIRING

Connect the RS-485 network connections to the (+) and (-) terminals where shown. Connect power to the “power” terminal, and power common/ground to the “ground” terminal.

Power may be 10-30VDC, or 12-24VAC. This gateway uses a half wave rectifier which means it is not required to have a dedicated isolation transformer (ungrounded AC) as with other gateways in the Control Solutions gateway family that use a full wave bridge rectifier. There is no separate AC common on the Babel Buster 485XL. One side of AC will be grounded.

ERROR CODES

Error codes returned via BACnet consist of an error class and code. The most common errors are listed below with class in parenthesis followed by code. Some applications will interpret these for you and provide a text description.

(property) code 9	Invalid data type
(property) code 32	Unknown property
(property) code 37	Value out of range
(property) code 40	Write access denied
(property) code 42	Invalid array index
(client) code 30	Timeout
(client) code 31	Unknown device
(object) code 31	Unknown object
(service) code 7	Inconsistent parameters
(service) code 10	Invalid access method
(service) code 29	Service request denied
(abort) code 4	Segmentation not supported
(reject) code 4	Invalid tag
(security) code 26	Password failure
(resource) code 0	Resource error – “other”