



Babel Buster 3 BACnet–Modbus Network Gateway User Guide

Relevant for model BB3-7101-TZ

Compatible exclusively with the Temperzone UC8

Revision	Date	Notes
1	June 2023	Initial Release

Contents

1	Introduction	4
1.1	Application of the BB3-7101-TZ	4
2	Connecting Gateway for the First Time	5
2.1	Connectors and Indicators	5
2.2	Opening the Web User Interface	6
2.3	Changing the Gateway IP Address	7
3	Local Objects	7
3.1	Behaviour of Input vs Output Objects	7
3.2	Local Objects Reliability Codes	8
3.3	Analog Objects	8
3.3.1	Analog Input Objects	8
3.3.2	Analog Output Objects	9
3.3.3	Analog Value Objects	10
3.4	Binary Objects	11
3.4.1	Binary Input Objects	11
3.4.2	Binary Output Objects	11
3.4.3	Binary Value Objects	11
3.5	Multistate Objects	13
3.5.1	Multistate Input Objects	13
3.5.2	Multistate Output Objects	13
3.5.3	Multistate Value Objects	13
4	Configuring Gateway as a BACnet Device	15
4.1	Device Object Parameters	15
4.2	Network Settings	16
4.3	BACnet Client Diagnostics	17
4.3.1	BACnet Error Code List	18
4.4	Configuring Gateway as a BACnet Server	19
4.4.1	Server Configuration	19
4.4.2	Accessing Local Objects	19
4.5	Configuring BBMD	21
4.5.1	Registering as a Foreign Device	22
4.5.2	Allowing Other Devices to Register Locally	23
4.5.3	Broadcast Distribution Table	24
4.5.4	User Login Password	24

5	Troubleshooting	25
5.1	BACnet IP Troubleshooting	25
5.1.1	Most Common Problems	25
5.1.2	Using Network Discovery Tool.....	25
6	Appendices.....	29
6.1	Appendix A: Hardware Details.....	29
6.1.1	Wiring.....	29
6.1.2	Front Panel LED Indicators.....	30
6.1.3	RS-485 Line Termination & Bias	31
6.2	Appendix B: BACnet Object Property Codes	31
6.3	Appendix C: SSL Certificates for Secure Web (HTTPS).....	40
6.3.1	X.509 Auto-Certificate Generation.....	40

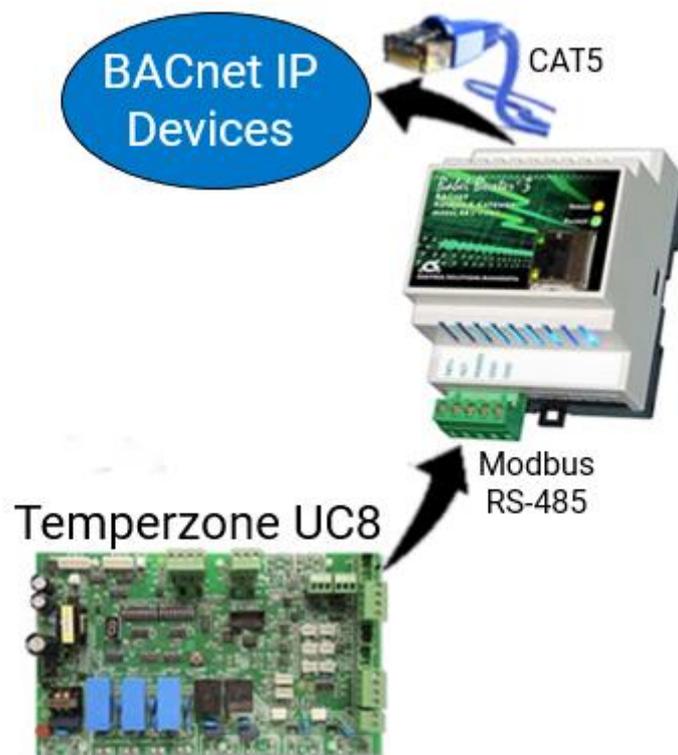
1 Introduction

1.1 Application of the BB3-7101-TZ

The Babel Buster 3: BB3-7101-TZ is a BACnet gateway device that converts the Temperzone unit's UC8 Modbus RTU communication to BACnet IP protocol. This manual provides instructions for operating and setting up the BB3-7101-TZ to communicate with Temperzone units using BACnet IP communication protocol.

The BB3-7101-TZ gateway serves as a BACnet IP client for a Modbus RTU master or slave.

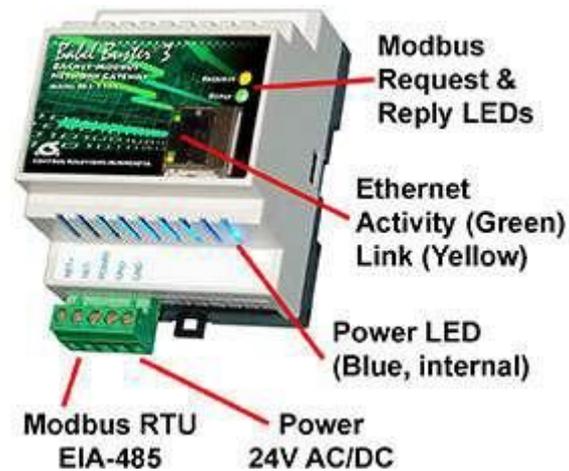
The BB3-7101-TZ can be considered as a data server with multiple network ports that have access to the internal database. The BB3-7101-TZ is not a router – BACnet devices cannot interact directly with Modbus devices or vice versa. Both BACnet and Modbus have access to the internal data objects which are updated according to rules defined by Temperzone.



2 Connecting Gateway for the First Time

2.1 Connectors and Indicators

Follow these steps to connect to the BB3-7101-TZ.



1. Connect power cables:

Apply 12-24VDC or 24VAC to the terminal marked "POWER".

Apply common or ground to one of the terminals marked "GND".

2. Connect a CAT5 cable between the RJ-45 jack on the gateway and your network switch or hub.

Note: You may be unable to connect directly to your PC unless you use a "crossover" cable (or your PC supports auto-MDX, which many newer laptops do).

3. Apply power.

A blue LED inside the case should light, showing power is present.

If the link LED (yellow) on the RJ-45 (CAT5) jack is not illuminated, check your Ethernet cable connections. Both link (yellow) and activity (green) LEDs on the RJ-45 jack will be illuminated solidly for a short time during boot-up. The entire bootup process will take 1-2 minutes, during which time you will not be able to connect with a browser.

Refer to **6.1 Appendix A: Hardware Details** for additional detail pertaining to connections and indicators as well as optional internal jumper settings.

2.2 Opening the Web User Interface

The default IP address as shipped is 10.0.0.101. Open your browser and enter “http://10.0.0.101/” in the address bar. You should see a page with the “Babel Buster 3 & Temperzone” header as shown below. The model should show “BB3-7101-TZ”. From this point, you will find help on each page on the website contained within the product.

If your PC is not already on the 10.0.0.x domain, and you are unable to connect, you may need to temporarily change your computer’s IP address to a static IP address “10.0.0.x” where “x” is any number except 101. Your subnet mask should also be set to 255.255.255.0.



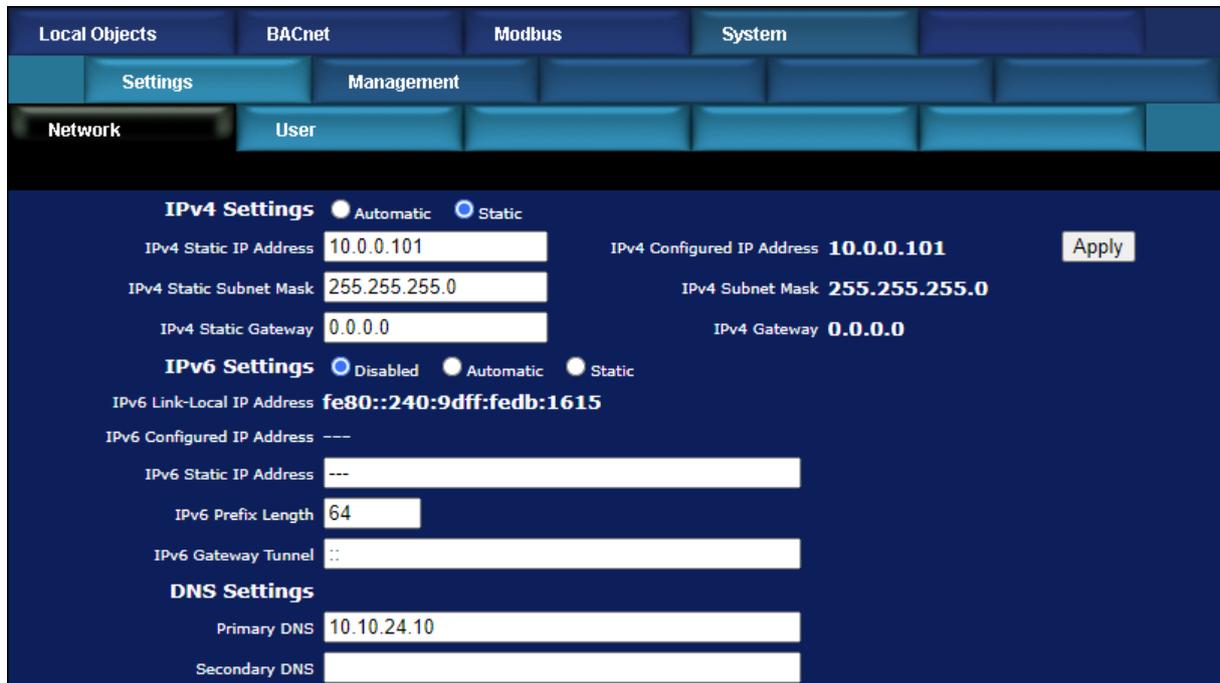
When you click on any of the page tabs such as ‘System’, you will be asked for a username and password. The only login as shipped is:

Username: **system**

Password: **admin**

2.3 Changing the Gateway IP Address

To change the IP address of the gateway, go to the 'System' tab and navigate to the 'Network' page under 'Settings'. The following page should appear.



The screenshot shows the 'Network' settings page. At the top, there are tabs for 'Local Objects', 'BACnet', 'Modbus', and 'System'. Under 'System', there are sub-tabs for 'Settings' and 'Management'. The 'Network' sub-tab is selected. The page is divided into three sections: IPv4 Settings, IPv6 Settings, and DNS Settings. The IPv4 Settings section has radio buttons for 'Automatic' and 'Static', with 'Static' selected. It includes input fields for 'IPv4 Static IP Address' (10.0.0.101), 'IPv4 Static Subnet Mask' (255.255.255.0), and 'IPv4 Static Gateway' (0.0.0.0). To the right, it shows 'IPv4 Configured IP Address' (10.0.0.101), 'IPv4 Subnet Mask' (255.255.255.0), and 'IPv4 Gateway' (0.0.0.0). There is an 'Apply' button. The IPv6 Settings section has radio buttons for 'Disabled', 'Automatic', and 'Static', with 'Disabled' selected. It includes input fields for 'IPv6 Link-Local IP Address' (fe80::240:9dff:fedb:1615), 'IPv6 Configured IP Address' (---), 'IPv6 Static IP Address' (---), 'IPv6 Prefix Length' (64), and 'IPv6 Gateway Tunnel' (::). The DNS Settings section includes input fields for 'Primary DNS' (10.10.24.10) and 'Secondary DNS' (---).

Change the IP address, subnet mask and gateway if applicable. Click Change IP to save the changes. Programming this into flash memory takes around half a minute. The new IP address will only take effect following the next system restart or power cycle.

3 Local Objects

3.1 Behaviour of Input vs Output Objects

When considering input and output values you should think of the BACnet system as the frame of reference rather than the BB3-7101-TZ device.

An **input** will receive input to the system from the device and so be a read only value. The value will be displayed in either analog form, binary form or multistate form.

An **output** will send data from the system to the device and so be a writable value. The value shall be provided as either analog form, binary form or multistate form.

An **object** can be either an input or an output and is generally both read and write. The values will be read or written in either analog form, binary form or multistate form.

3.2 Local Objects Reliability Codes

Reliability codes may be any of the following:

Modbus client/master, no response from slave	(64)
Modbus client/master, crc error	(65)
Modbus exception, illegal function code	(66)
Modbus exception, illegal data address	(67)
Modbus exception, illegal data value	(68)
Modbus exception, code+65, rarely used	(69..79)
Local device, configuration property fault	(80)
Faulty Modbus packet	(81)
BACnet IP client, device timeout	(82)
BACnet IP client, error returned by server	(83)

Status flags A, B, C & D indicate the following, 0 meaning not true, 1 meaning true:

A = in alarm

B = fault

C = overridden

D = out of service

3.3 Analog Objects

3.3.1 Analog Input Objects

The source of data for an Analog Input object will typically be reading from the UC8 Controller. These values are Read Only. They may appear editable, however there is no way to save any changes made on site.

The screenshot displays a web-based configuration interface for an Analog Input object. At the top, there is a navigation bar with tabs for 'Local Objects', 'BACnet', 'Modbus', and 'System'. Under 'Local Objects', there are sub-tabs for 'Analog', 'Binary', 'Multi-State', and 'Actions'. The 'Analog' sub-tab is active, and within it, 'Input Objects' is selected. The main configuration area shows 'Analog Input # 1' with a text input field containing '1'. Below this, there are several configuration options: 'Reliability: 0', 'Status: 0,0,0,0', 'Device Link: RTU_R1', 'Out of Service: ', and 'Deconfigure: '. The 'Object name' field contains 'U1_ID', and the 'Present Value' field contains '210.0000'. The 'Description' field contains 'Unit1 Controller ID'. The 'COV increment' field contains '0.00', and the 'Units' dropdown menu is set to 'no_units'. At the bottom right, there are three buttons: 'Update', '< Prev', and 'Next >'.

The object name, units, value, and status are shown for the object number entered at the top of the page. Click 'Prev/Next' to scroll through the list. Click 'Refresh' to update the page.

The object may be set Out of Service by checking that box and clicking Update. The present value may be changed by entering a value, checking Force, and clicking Update.

The source of data for an Analog Input object will typically be reading from some other BACnet or Modbus device via the map indicated by the Device Link. The mapped device will be polled at the rate specified by the Read Map.

Out of Service means polling of the mapped remote device will stop. While out of service, the present value may be written by the BACnet client. Data may be forced via this web page at any time but will be overwritten by the next mapped client update unless the object is out of service.

Device link will indicate RTU, followed by R for read or W for write, and a number which is the map number in the table of read or write maps for mapping to this BACnet object. The designation R means read from a UC8 controller, and W means write to a UC8 controller.

3.3.2 Analog Output Objects

The destination of data for an Analog Output object will be from a UC8 controller. These may appear editable, however there is no way to save any changes made on site.

The object name, units, value, and status are shown for the object number entered at the top of the page. Click Prev/Next to scroll through the list. Click Refresh to update the page.

The destination of data for an Analog Output object will be writing to the UC8 controller via the map indicated by the Device Link. The UC8 controller will be updated upon change of source data and/or periodically as defined by the Write Map.

The Analog Output object is command able, meaning the BACnet client must write both a value and a priority level for that value. The highest-level value will be the one written to the remote device (if one is mapped). If all values are relinquished, the relinquish default value will be written to the remote device.

To set an output object manually from this page, check the Force box, enter a value in the Present Value window, and select a priority level to assign to your forced value.

Then click Update. To return a given priority level to NULL, simply type the word NULL in the Present Value window, check Force, and click Update.

Out of service means the mapped remote device will not be written to. Values written by the BACnet client will be retained, but only applied when this object is placed back in service. At that time, the highest priority value will be written to the remote device.

Reliability codes indicate errors as itemized on the tabular object list.

3.3.3 Analog Value Objects

Analog Value objects may be both a source and destination for a UC8 controller. These may appear editable, however there is no way to save any changes made on site.

The screenshot displays the configuration page for an Analog Value Object. At the top, there is a navigation bar with tabs for 'Local Objects', 'BACnet', 'Modbus', and 'System'. Under the 'BACnet' tab, there are sub-tabs for 'Analog', 'Binary', 'Multi-State', and 'Actions'. The 'Value Objects' sub-tab is active. Below the navigation, there is a header for 'Analog Value # 2' with 'Update', '< Prev', and 'Next >' buttons. The main configuration area includes fields for 'Reliability: 65', 'Status: 0,1,0,0', 'Device Link: RTU R70 RTU W3', 'Out of Service: [checkbox]', and 'Deconfigure: [checkbox]'. The 'Object name' is 'U1_REQ_FAN', 'Force' is [checkbox], and 'Present Value' is '0.00'. The 'Description' is 'Unit1 Request Indoor Fan Speed'. 'COV increment' is '0.00' and 'Units' is 'percent'.

The object name, units, value, and status are shown for the object number entered at the top of the page. Click Prev/Next to scroll through the list. Click Refresh to update the page.

The object may be set Out of Service by checking that box and clicking Update. The present value may be changed by entering a value, checking Force, and clicking Update.

Analog Value objects may be both a source and destination. The source of data for the Value object will be reading from a remote device when associated with a Read Map. The destination of data for the Value object will be writing to a UC8 Controller when associated with a Write Map.

The Value object may be simultaneously associated with both Read and Write maps pointing to the same remote device object. When this Value object receives new data (from any source), this data will be written to the mapped remote device before any subsequent read from the same device. Thus, the Value data is not discarded by the read operation before the new data can be written.

3.4 Binary Objects

3.4.1 Binary Input Objects

The source of data for a Binary Input object will typically be reading from a UC8 controller. These may appear editable, however there is no way to save any changes made on site.

The object name, units, value, and status are shown for the object number entered at the top of the page. Click Prev/Next to scroll through the list. Click Refresh to update the page.

The object may be set Out of Service by checking that box and clicking Update. The present value may be changed by entering a value, checking Force, and clicking Update.

The source of data for a Binary Input object will typically be reading from A UC8 controller via the map indicated by the Device Link. The UC8 controller will be polled at the rate specified by Read Map.

Check 'Deconfigure' and click Update to erase configuration for this object.

3.4.2 Binary Output Objects

There are no binary objects operational in the program. These may appear editable, however there is no way to save any changes made on site.

Object	Object Name	Object Description	Out of Service	Present Value	Reliability	Status	Text
1	Binary Output 1		N	Inactive	0	0,0,0,0	
2	Binary Output 2		N	Inactive	0	0,0,0,0	
3	Binary Output 3		N	Inactive	0	0,0,0,0	
4	Binary Output 4		N	Inactive	0	0,0,0,0	
5	Binary Output 5		N	Inactive	0	0,0,0,0	
6	Binary Output 6		N	Inactive	0	0,0,0,0	
7	Binary Output 7		N	Inactive	0	0,0,0,0	
8	Binary Output 8		N	Inactive	0	0,0,0,0	
9	Binary Output 9		N	Inactive	0	0,0,0,0	
10	Binary Output 10		N	Inactive	0	0,0,0,0	

3.4.3 Binary Value Objects

Binary Value objects may be both a source and destination for the UC8 controller. These may appear editable, however there is no way to save any changes made on site.

Local Objects	BACnet	Modbus	System
Analog	Binary	Multi-State	Actions
Input Objects	Output Objects	Value Objects	

Binary Value # Update < Prev Next >

Reliability: 64 Status: 0,1,0,0 Device Link: [RTU_R97](#) [RTU_W10](#) Out of Service: Deconfigure:

Object name: Force Present Value:

Description:

Active Text: Inctive Text:

The object name, units, value, and status are shown for the object number entered at the top of the page. Click Prev/Next to scroll through the list. Click Refresh to update the page.

The object may be set Out of Service by checking that box and clicking Update. The present value may be changed by entering a value, checking Force, and clicking Update.

Binary Value objects may be both a source and destination for a UC8 controller. The source of data for the Value object will be reading from a UC8 controller when associated with a Read Map. The destination of data for the Value object will be written to a UC8 controller when associated with a Write Map. If a UC8 controller is mapped, the UC8 links are displayed above. You may click on either link to view the respective mapping.

The Value object may be simultaneously associated with both Read and Write maps pointing to the same UC8 controller object. When this Value object receives new data, this data will be written to the mapped UC8 controller before any subsequent read from the UC8 controller. Thus, the Value data is not discarded by the read operation before the new data can be written.

3.5 Multistate Objects

3.5.1 Multistate Input Objects

The source of data for a Multistate Input object will typically be reading from a UC8 controller.

The object name, value, and status are shown for the object number entered at the top of the page. Click Prev/Next to scroll through the list. Click Refresh to update the page.

The object may be set Out of Service by checking that box and clicking Update. The present value may be changed by entering a value, checking Force, and clicking Update.

3.5.2 Multistate Output Objects

There are no Multistate output objects operational in the program. These may appear editable, however there is no way to save any changes made on site.

Object	Object Name Object Description	Out of Service	Present Value	State Text	Reliability	Status
1	Multi-state Output 1	N	1	---	0	0,0,0,0

3.5.3 Multistate Value Objects

There are no Multistate value objects operational in the program. These may appear editable, however there is no way to save any changes made on site.

Local Objects		BACnet	Modbus	System		
Analog		Binary	Multi-State	Actions		
Input Objects	Output Objects	Value Objects				
Multi-State Value Objects		Showing objects from <input type="text" value="1"/>		Refresh	< Prev	Next >
Object	Object Name Object Description	Out of Service	Present Value	State Text	Reliability	Status
<u>1</u>	Multi-state Value 1	N	1	---	0	0,0,0,0
<u>2</u>	Multi-state Value 2	N	1	---	0	0,0,0,0
<u>3</u>	Multi-state Value 3	N	1	---	0	0,0,0,0
<u>4</u>	Multi-state Value 4	N	1	---	0	0,0,0,0
<u>5</u>	Multi-state Value 5	N	1	---	0	0,0,0,0
<u>6</u>	Multi-state Value 6	N	1	---	0	0,0,0,0
<u>7</u>	Multi-state Value 7	N	1	---	0	0,0,0,0
<u>8</u>	Multi-state Value 8	N	1	---	0	0,0,0,0
<u>9</u>	Multi-state Value 9	N	1	---	0	0,0,0,0
<u>10</u>	Multi-state Value 10	N	1	---	0	0,0,0,0

4 Configuring Gateway as a BACnet Device

4.1 Device Object Parameters

The identity of the gateway as a BACnet device is entered on this page, along with other device object parameters.

The screenshot shows the 'BACnet Settings' configuration page. The page is divided into sections: 'BACnet Device Settings' and 'Local Network Settings'. The 'BACnet Device Settings' section includes fields for Device Instance (20800), Port (47808), Device Object Name (Babel Buster BB3-7101), Device Description, Device Location (St. Paul, Minnesota), APDU Timeout (3000), APDU Retries (0), APDU Segment Timeout (5000), Database Revision (1), Local Command Priority (10), and Backlog Limit (0). There are also four checkboxes: 'Allow fault self-reset without Ack.', 'Disable Segmentation.', 'Disable self-restart upon communications loss', and 'Disable Modbus offset.' The 'Local Network Settings' section includes a 'Network Number' field set to 0, with a note '(0 = Local)'. A 'Save' button is located at the top right of the settings area.

Enter a device instance from 1 to 4,194,303. Enter a port number (note that 47808 is the standard port expected by most BACnet IP devices).

The device object name, description, and location are entered here. The device object name is expected to be unique to the entire BACnet network. Standard BACnet timeout and retry values are also entered on this page. These values are stored in a special area of non-volatile memory.

Network Number is optional, and normally left set to zero here. Normally, BACnet routers will take care of any network number translation needed. If you leave this network number entry set to zero, then this BB3-7101-TZ WILL respond to any local broadcast message. If you enter a network number here, then this device WILL NOT respond to local broadcast messages and will only respond to remote broadcast messages or global broadcast messages. The default setting is zero, and BTL certification was done with the default setting of zero - no other number is needed here for normal typical use.

Local command priority is used when the result of a Calculate or Copy rule is written to a commandable object. It is also used if the result of a client read map is saved to a local commandable object, although this would not be recommended. Output objects are commandable. Client read maps should store results in input or value objects, while client write maps take their data from value or output objects. In other words, output objects should not be used for input.

Check the "allow" check box to allow faults to self-reset. These faults are those conditions indicated by a non-zero reliability code in any of the data objects (see list on data objects pages). Normally an external client needs to read the reliability code to acknowledge the fault before it will automatically reset. By checking the "allow" check box, faults will

automatically self-reset without acknowledgement. This is required any time the client does not periodically read reliability codes but does check fault status - a behavior known to be common to BMS front ends.

Check the "disable self-restart" box to disable self-restart upon communication loss. If this box is not checked, this gateway will restart itself to auto-recover if communications with devices has started and then stopped.

By default, when Modbus reads values that are mapped to a Multi-State object, the values are offset by one. Multi-State objects are not allowed to contain a value of zero per BACnet protocol standard. Since Modbus doesn't often deal very well with zero being prohibited, values are offset by one. Therefore, where Modbus sees zero, the BACnet Multi-State object will see one. This offset only applies to Multi-State objects. You can disable this offset by checking the "Disable Modbus offset" box and saving the setting. Just be aware that with the offset disabled, Modbus will not know the difference between zero and one because any attempt to write zero to the Multi-State object will be forced to one. You can safely disable the offset when only reading Multi-State objects, but use extra caution when writing to Multi-State objects from Modbus with the offset disabled.

Click Save to store. This store process will take a little while as these parameters are being saved to non-volatile memory. A change in port number will not take effect until the next system restart.

4.2 Network Settings

The two most important things that must be unique on the BACnet IP network are device instance, and IP address. The IP address is set on the Network page.

The screenshot displays the 'Network' settings page. At the top, there are navigation tabs for 'Local Objects', 'BACnet', 'Modbus', and 'System'. Below these are sub-tabs for 'Settings' and 'Management'. The 'Network' tab is selected, and within it, the 'User' sub-tab is active. The main content area is divided into three sections: 'IPv4 Settings', 'IPv6 Settings', and 'DNS Settings'.
- **IPv4 Settings:** Radio buttons for 'Automatic' and 'Static' are present, with 'Static' selected. Fields include 'IPv4 Static IP Address' (10.0.0.101), 'IPv4 Static Subnet Mask' (255.255.255.0), and 'IPv4 Static Gateway' (0.0.0.0). On the right, 'IPv4 Configured IP Address' (10.0.0.101), 'IPv4 Subnet Mask' (255.255.255.0), and 'IPv4 Gateway' (0.0.0.0) are displayed. An 'Apply' button is located to the right of the configured values.
- **IPv6 Settings:** Radio buttons for 'Disabled', 'Automatic', and 'Static' are present, with 'Disabled' selected. Fields include 'IPv6 Link-Local IP Address' (fe80::240:9dff:fedb:1615) and 'IPv6 Configured IP Address' (---).
- **DNS Settings:** Fields for 'Primary DNS' (10.10.24.10) and 'Secondary DNS' (---) are shown.

Select either Static or Automatic for IPv4. To change the Static IP address of this device, enter the address, subnet mask, and gateway, then click Apply.

Select Automatic to specify that DHCP should be used to obtain an IP address upon power-up. IP address change will take effect upon next power cycle.

The above screenshot is only a portion of the Network setup page, and is the only part of the Network page that is required for BACnet IP. The remainder of the Network page is discussed in Section 11.3.

The Web User Interface is accessible via IPv6; however, the BACnet IP Client does not yet support IPv6. The only demonstrated version of BACnet IP over IPv6 does not use actual IPv6 addresses - it uses Virtual MAC addresses (VMAC) and address translation tables. The VMAC approach allows IPv6 to coexist with original IPv4 devices. As of the BTL testing of this device, test specifications for BACnet IP over IPv6 were not available and thus IPv6 support is not included in the BTL tested device.

4.3 BACnet Client Diagnostics

If errors are detected in the course of reading or writing other BACnet objects via the client's maps, they will be indicated on the errors pages.

The screenshot shows a web interface with a navigation menu at the top containing 'Local Objects', 'BACnet', 'Modbus', and 'System'. Under 'BACnet', there are sub-menus for 'Local Device', 'BACnet Client', 'Diagnostics', and 'BBMD'. The 'Diagnostics' menu is active, showing 'Errors: Read Maps' and 'Errors: Write Maps'. Below the menu is a table with the following data:

Map #	Remote Type	Remote Object #	Remote Device	Name	Error Class	Error Code
--	---	0		---	0	0

Navigation buttons '<< Top' and 'Next >' are visible to the right of the table, and a 'Reset Errors' button is at the bottom right.

Errors for BACnet client read/write maps are shown on these pages. Only those maps with errors to report are listed. Refer to the code and class lists below for interpretation. In the illustration above, error class 1 says the error refers to "object" and the code says "unknown object". In other words, AV 6125 does not exist in the device shown.

Proprietary class 82, code 0, is generated locally indicating a timeout, no response received from remote server. All other codes listed below are returned by the remote server.

4.3.1 BACnet Error Code List

```
0 = ERROR_CLASS_DEVICE
1 = ERROR_CLASS_OBJECT
2 = ERROR_CLASS_PROPERTY
3 = ERROR_CLASS_RESOURCES
4 = ERROR_CLASS_SECURITY
5 = ERROR_CLASS_SERVICES

/* valid for all classes */ 0 = ERROR_CODE_OTHER

/* Error Class - Device */
2 = ERROR_CODE_CONFIGURATION_IN_PROGRESS
3 = ERROR_CODE_DEVICE_BUSY
25 = ERROR_CODE_OPERATIONAL_PROBLEM

/* Error Class - Object */
4 = ERROR_CODE_DYNAMIC_CREATION_NOT_SUPPORTED
17 = ERROR_CODE_NO_OBJECTS_OF_SPECIFIED_TYPE
23 = ERROR_CODE_OBJECT_DELETION_NOT_PERMITTED
24 = ERROR_CODE_OBJECT_IDENTIFIER_ALREADY_EXISTS
27 = ERROR_CODE_READ_ACCESS_DENIED
31 = ERROR_CODE_UNKNOWN_OBJECT
36 = ERROR_CODE_UNSUPPORTED_OBJECT_TYPE

/* Error Class - Property */
8 = ERROR_CODE_INCONSISTENT_SELECTION_CRITERION
9 = ERROR_CODE_INVALID_DATA_TYPE
32 = ERROR_CODE_UNKNOWN_PROPERTY
37 = ERROR_CODE_VALUE_OUT_OF_RANGE
40 = ERROR_CODE_WRITE_ACCESS_DENIED
41 = ERROR_CODE_CHARACTER_SET_NOT_SUPPORTED
42 = ERROR_CODE_INVALID_ARRAY_INDEX
44 = ERROR_CODE_NOT_COV_PROPERTY
45 = ERROR_CODE_OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED
47 = ERROR_CODE_DATATYPE_NOT_SUPPORTED
50 = ERROR_CODE_PROPERTY_IS_NOT_AN_ARRAY

/* Error Class - Resources */
18 = ERROR_CODE_NO_SPACE_FOR_OBJECT
19 = ERROR_CODE_NO_SPACE_TO_ADD_LIST_ELEMENT
20 = ERROR_CODE_NO_SPACE_TO_WRITE_PROPERTY

/* Error Class - Security */
1 = ERROR_CODE_AUTHENTICATION_FAILED
6 = ERROR_CODE_INCOMPATIBLE_SECURITY_LEVELS
12 = ERROR_CODE_INVALID_OPERATOR_NAME
15 = ERROR_CODE_KEY_GENERATION_ERROR
26 = ERROR_CODE_PASSWORD_FAILURE
28 = ERROR_CODE_SECURITY_NOT_SUPPORTED
30 = ERROR_CODE_TIMEOUT
```

```

/* Error Class - Services */
5 = ERROR_CODE_FILE_ACCESS_DENIED
7 = ERROR_CODE_INCONSISTENT_PARAMETERS
10 = ERROR_CODE_INVALID_FILE_ACCESS_METHOD
11 = ERROR_CODE_ERROR_CODE_INVALID_FILE_START_POSITION
13 = ERROR_CODE_INVALID_PARAMETER_DATA_TYPE
14 = ERROR_CODE_INVALID_TIME_STAMP
16 = ERROR_CODE_MISSING_REQUIRED_PARAMETER
22 = ERROR_CODE_PROPERTY_IS_NOT_A_LIST
29 = ERROR_CODE_SERVICE_REQUEST_DENIED
43 = ERROR_CODE_COV_SUBSCRIPTION_FAILED
46 = ERROR_CODE_INVALID_CONFIGURATION_DATA
48 = ERROR_CODE_DUPLICATE_NAME
49 = ERROR_CODE_DUPLICATE_OBJECT_ID

```

4.4 Configuring Gateway as a BACnet Server

4.4.1 Server Configuration

The BB3-7101-TZ contains a set of BACnet objects whose only purpose is to store copies of data obtained from the UC8 controllers. This copy of data may then be queried by different devices.

The only configuration needed to use the BB3-7101-TZ as a BACnet server is to set the Device instance on the BACnet page. The device should also be given an object name that will be unique on the entire network. Configuring the gateway as a BACnet Device is described in more detail in Section 4.

The screenshot displays the 'BACnet Settings' configuration page. The interface is dark-themed with white text and input fields. At the top, there are navigation tabs for 'Local Objects', 'BACnet', 'Modbus', and 'System'. Below these, there are sub-tabs for 'Local Device', 'BACnet Client', 'Diagnostics', and 'BBMD'. The 'BACnet Settings' tab is active. The main content area is titled 'BACnet Device Settings' and 'Local Network Settings'. It contains several input fields and checkboxes. The 'Device Instance' field is set to 20800. The 'Port (default 0xBAC0 = 47808)' field is set to 47808. The 'Network Number' field is set to 0, with a note '(0 = Local)'. The 'Device Object Name' field is set to 'Babel Buster BB3-7101'. The 'Device Location' field is set to 'St. Paul, Minnesota'. The 'APDU Timeout' field is set to 3000. The 'APDU Retries' field is set to 0. The 'APDU Segment Timeout' field is set to 5000. The 'Database Revision' field is set to 1. The 'Local Command Priority' field is set to 10. The 'Backlog Limit' field is set to 0. There are four checkboxes: 'Allow fault self-reset without Ack.' (unchecked), 'Disable Segmentation.' (unchecked), 'Disable self-restart upon communications loss' (unchecked), and 'Disable Modbus offset.' (unchecked). A 'Save' button is located in the top right corner of the configuration area.

4.4.2 Accessing Local Objects

The collection of local objects includes Analog, Binary, and Multi-State types of objects, and includes Input, (commandable) Output, and (writeable) Value types of each of those objects. The BB3-7101-TZ also contains a Device object which is configured in the above screen.

Data may be placed in the local objects by other devices writing to the BB3-7101-TZ, or by the BB3-7101-TZ querying other devices. When the BB3-7101-TZ is configured to query other devices, these operations are defined by “read maps” and “write maps” associated with the respective client function (BACnet client).

The following pages illustrate the Analog Input object pages and the Binary Output object pages. The remaining object pages found in the BB3-7101-TZ are virtually identical, and are not replicated here.

Each object page initially comes up as a table of object data. Click on the object number in the left-hand column to expand the view of that object and access the windows that let you locally force values, assign units, etc.

Local Objects		BACnet	Modbus	System		
Analog		Binary	Multi-State	Actions		
Input Objects		Output Objects	Value Objects			
Analog Input Objects		Showing objects from <input type="text" value="1"/>		Refresh	< Prev	Next >
Object	Object Name Object Description	Out of Service	Present Value	Reliability	Status	Units
1	U1_ID Unit1 Controller ID	N	210.0000	0	0,0,0,0	no_units
2	U1_SW Unit1 SW Version	N	618.0000	0	0,0,0,0	no_units
3	U1_T_OC Unit1 Temp: Outdoor Coil	N	-100.0000	0	0,0,0,0	degrees_celsius
4	U1_T_IC Unit1 Temp: Indoor Coil	N	-100.0000	0	0,0,0,0	degrees_celsius
5	U1_T_OA Unit1 Temp: Outdoor Ambient	N	14.54000	0	0,0,0,0	degrees_celsius
6	U1_T_SL Unit1 Temp: Suction Line	N	11.65000	0	0,0,0,0	degrees_celsius
7	U1_T_DL Unit1 Temp: Discharge Line	N	69.03000	0	0,0,0,0	degrees_celsius
8	U1_T_DEI Unit1 Temp: De-Ice Sensor	N	33.59000	0	0,0,0,0	degrees_celsius
9	U1_T_EV Unit1 Temp: Evaporating	N	4.600000	0	0,0,0,0	degrees_celsius
10	U1_T_CO Unit1 Temp: Condensing	N	39.00000	0	0,0,0,0	degrees_celsius
11	U1_T_UC Unit1 Temp: Controller	N	23.74000	0	0,0,0,0	degrees_celsius
12	U1_SH_S Unit1 SH: Suction Side	N	7.050000	0	0,0,0,0	degrees_kelvin
13	U1_SH_D Unit1 SH: Discharge Side	N	30.03000	0	0,0,0,0	degrees_kelvin
14	U1_SH_C Unit1 SH: Combined	N	7.050000	0	0,0,0,0	degrees_kelvin
15	U1_P_SL Unit1 Pressure: Suction Line	N	835.0000	0	0,0,0,0	kilopascals

The object name, units, value, and status are shown for a list of objects starting with the number entered at the top of the page. Click Prev/Next to scroll through the list. Click on the object number in the first column to change name, units, COV, and out-of-service state.

The source of data for an Analog Input object will be reading from the UC8 controller. Click on the object number in the first column for more detail including the link to any client map providing data to this object.

Reliability codes may be any of the following:

- Modbus client/master, no response from slave (64)
- Modbus client/master, crc error (65)
- Modbus exception, illegal function code (66)
- Modbus exception, illegal data address (67)
- Modbus exception, illegal data value (68)
- Modbus exception, code+65, rarely used (69..79)

- Local device, configuration property fault (80)
- Faulty Modbus packet (81)
- BACnet IP client, device timeout (82)
- BACnet IP client, error returned by server (83)

Click on an Object number in the first column of maps to get the expanded view of that object as follows:

The object name, units, value, and status are shown for the object number entered at the top of the page. Click Prev/Next to scroll through the list. Click Refresh to update the page, or Update to accept changes.

The source of data for an Analog Input object will typically be reading from UC8 controller via the map indicated by the Device Link.

Check 'Deconfigure' and click Update to erase configuration for this object.

Object	Object Name Object Description	Out of Service	Present Value	Reliability	Status	Units
1	U1_CNTRL_REG_RESET Unit1 Control Register Reset	N	0.00	0	0,0,0,0	no_units

The object name, value, and status are shown for a list of objects starting with the number entered at the top of the page. Click Prev/Next to scroll through the list. Click on the object number in the first column to change name or out-of-service state.

4.5 Configuring BBMD

BBMD stands for BACnet Broadcast Management Device. Messages such as "Who-Is" and "I-Am" are broadcast. Most routers do not pass broadcast messages along. The BBMD solves this problem by explicitly directing broadcast messages to a specific IP address.

The BBMD Settings page appears as shown above when no part of BBMD support is enabled, as is the case when shipped. Do not enable BBMD if you are not aware of needing it and/or do not understand how BBMD works. The three elements of BBMD support are discussed in the following sections, and their use is often mutually exclusive, meaning you will often need only one of the three elements.

The screenshot shows the 'BBMD Settings' page. At the top, there are navigation tabs: Local Objects, BACnet, Modbus, System, Local Device, BACnet Client, Diagnostics, and BBMD. The 'BBMD Settings' tab is active, and the 'Edit BDT' button is highlighted. Below the navigation is a 'Refresh' button. The main content area is divided into three sections:

- Broadcast Distribution Table (BDT):** A table with columns 'Broadcast Address:Port' and 'Broadcast Mask'. Both columns contain '---'.
- Foreign Devices Registered Locally:** A table with columns 'Foreign Devices Registered Locally' and 'Time to Live'. Both columns contain '---'.
- Local Device's Registration as a Foreign Device at Remote Location:** This section contains:
 - 'Enable BBMD' checkbox:
 - 'BBMD Time To Live (seconds)' input field: (Zero disables foreign registration)
 - 'Save' button
 - 'BBMD IP Address, Port' input fields: and . The status 'BBMD is not registered.' is displayed.

4.5.1 Registering as a Foreign Device

If you have a remote BB3-7101-TZ that needs to connect via router, including NAT router, to a local network, use Foreign Device Registration. There will typically be a master device, such as operator station or other front end, that includes BBMD. The IP address of this device is the one that should be given as the BBMD address for foreign device registration.

This screenshot is identical to the one above, but with the following changes in the 'Local Device's Registration as a Foreign Device at Remote Location' section:

- 'Enable BBMD' checkbox:
- 'BBMD Time To Live (seconds)' input field: (Zero disables foreign registration)
- 'BBMD IP Address, Port' input fields: and . The status 'BBMD is not registered.' is displayed.

To enable BBMD processing, check the "Enable BBMD" box. This applies to foreign device registration. The broadcast distribution table functions regardless of whether foreign device registration is enabled.

If the BB3-7101-TZ should register as a foreign device with another BBMD, then the port, time-to-live, and IP address of the remote BBMD must be given. The local BBMD will attempt to register with the remote BBMD whose address is given.

Disable this device's attempts to register elsewhere, but allow other devices to register here, by setting time to live to zero with BBMD enabled.

4.5.2 Allowing Other Devices to Register Locally

The BB3-7101-TZ can be the BBMD that other devices register with. The screenshot below shows that three other devices have registered with this BBMD, and broadcast messages will now be sent explicitly to these locations. In this case, there are NAT routers between this local device and the three remote devices. While they are all on physically separate local networks, they will appear as a single BACnet network even if the local networks are miles apart. The local BACnet client will be able to communicate with these remote BACnet devices as a result of the foreign registration.

Note that foreign registration only provides communication with a single remote device. If communicating with an entire remote network of BACnet devices is the intent, then a full BACnet router is required, and BBMD would be handled by the BACnet router (disable everything on this BBMD Settings page if connected via a BACnet router).

To allow foreign devices to register with this device, but not have this device register elsewhere, check Enable BBMD, but enter zero for BBMD Time To Live. This enables BBMD but disables this device's attempt to register somewhere else.

4.5.3 Broadcast Distribution Table

A Broadcast Distribution Table (BDT) defines a list of IP addresses that the BBMD should send broadcast messages to. It is important to note that a BBMD only forwards broadcast messages. It does not do full routing. If you are attempting to connect two networks across a NAT router, you must get a full BACnet Router to accomplish this. For this reason, the BDT has limited usefulness when only BBMD is present. The BB3-7101-TZ only includes BBMD, not full routing.

Broadcast distribution will result in device discovery, but you will not be able to read/write properties in the remote device without full routing. Foreign device registration via a router does result in being able to fully communicate with the foreign device from the local network.

Broadcast Address : Port		Broadcast Mask	
10.0.0.101	47808	24	FFFFFF00
0.0.0.0	0	0	00000000

Once the table has been initialized, it will appear on the BBMD Settings page as illustrated below.

Broadcast Distribution Table (BDT)	Broadcast Address:Port	Broadcast Mask
	10.0.0.101:47808	FFFFFF00

4.5.4 User Login Password

Default Login:

Username: **system**

Password: **admin**

5 Troubleshooting

5.1 BACnet IP Troubleshooting

5.1.1 Most Common Problems

The most common problem is "no response" or timeout. The most common cause of this problem for BACnet IP is a network configuration problem, such as incorrect IP address or IP address that cannot be reached as configured. The problem sometimes lies outside the BB3-7101-TZ and may require consulting with the IT personnel responsible for the network if on a large network.

The subnet mask determines what part of the IP address constitutes the domain, and all devices on the same network must be on the same domain before they can communicate.

Obviously two devices being assigned the same IP address is going to cause trouble. If you can communicate at all with a device having a duplicate IP address, it will be intermittent, and potentially erratic as the other device having the same IP address may be responding to your queries.

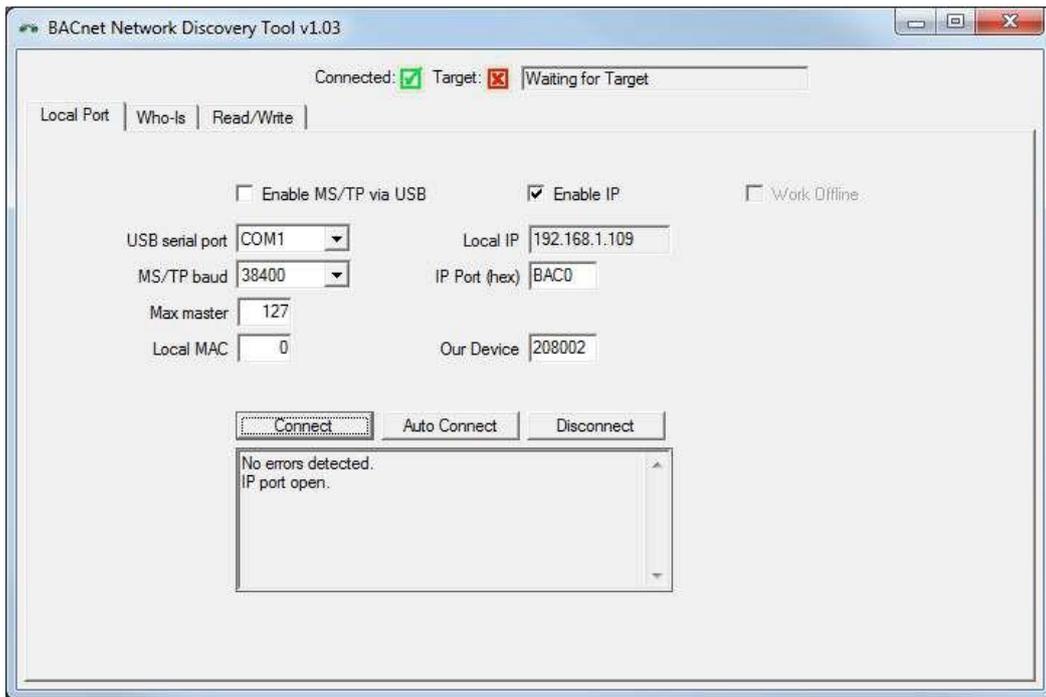
If you are connecting via one or more routers, then everything that applies to routing issues will apply to your device. A complete discussion of NAT routing, BACnet routing, etc.

Once the BB3-7101-TZ is communicating BACnet IP, then next area for possible concern is with the BACnet client. If the gateway is supposed to be polling other IP devices, but the data does not appear correct, the first thing to check is the reliability code. Any reliability code other than zero is a problem. Refer to the list at the bottom of any of the Data Objects pages for explanation of the non-zero codes. If the reliability code indicates that an error was returned by the server (meaning the other BACnet device you are trying to query), then refer to the BACnet Diagnostics page for additional error information.

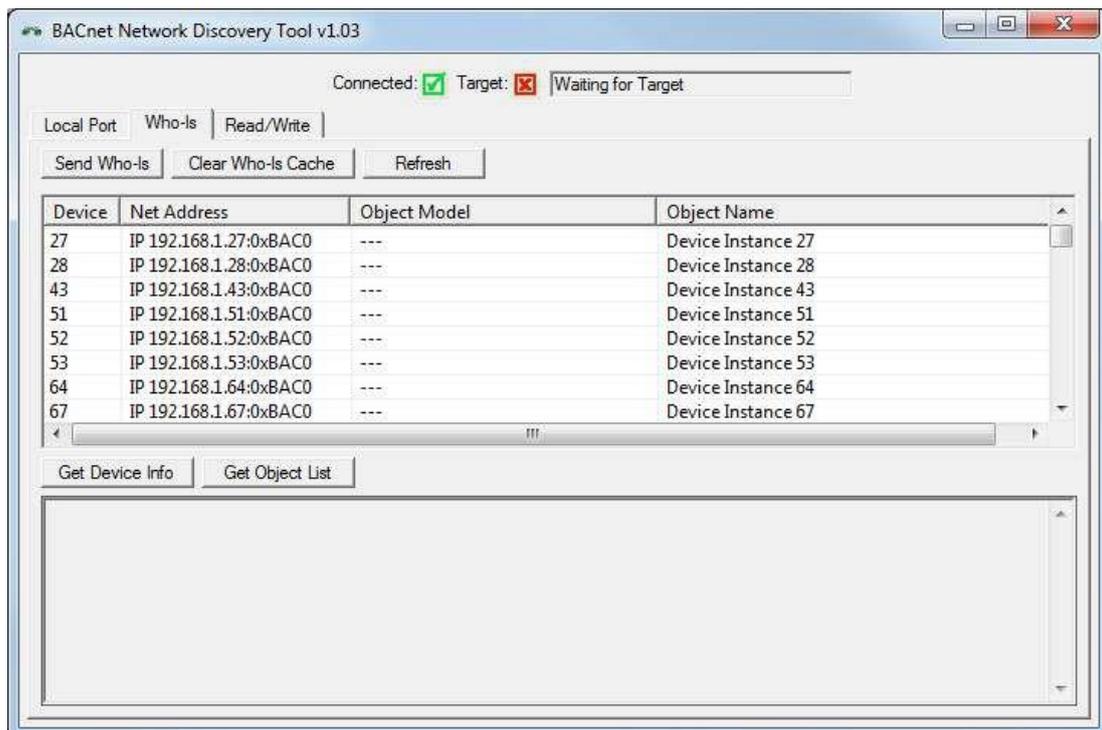
5.1.2 Using Network Discovery Tool

Control Solutions has created a Network Discovery Tool to perform simple diagnostics on BACnet devices and networks. It works with BACnet IP using your PC's Ethernet connection - assuming your PC is connected to the BACnet IP network.

Simple check Enable IP, and click Connect to begin.

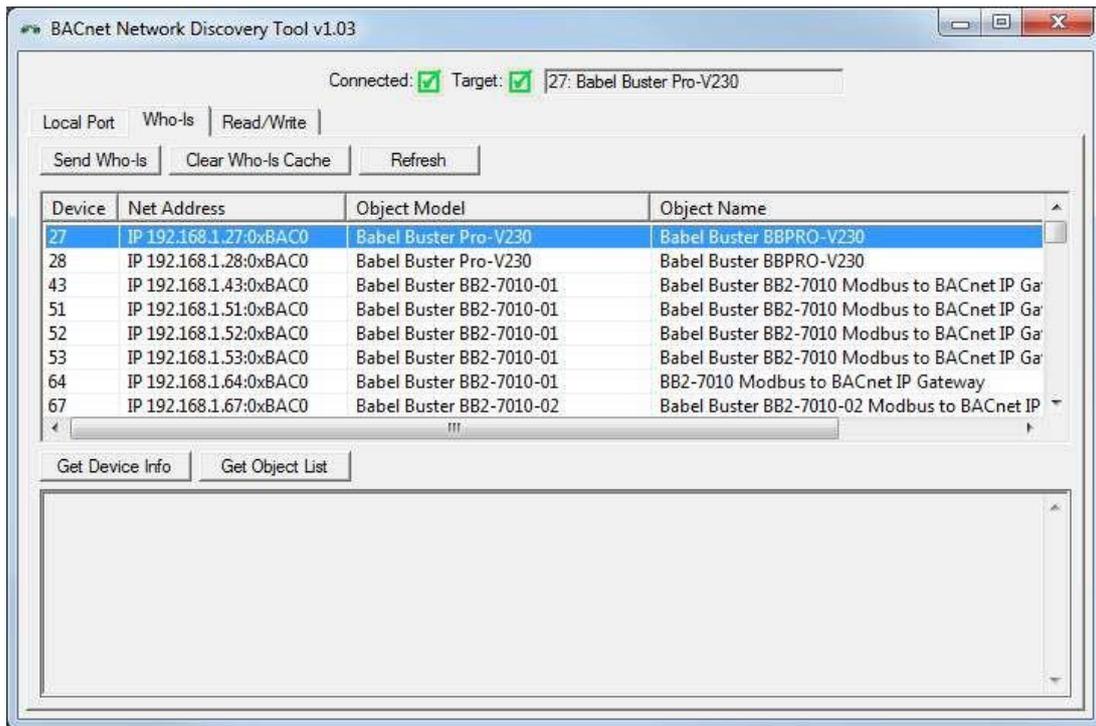


Once connected, go to the Who-Is page. Usually, by the time you get there, the results of the first automatic Who-Is are already displayed.

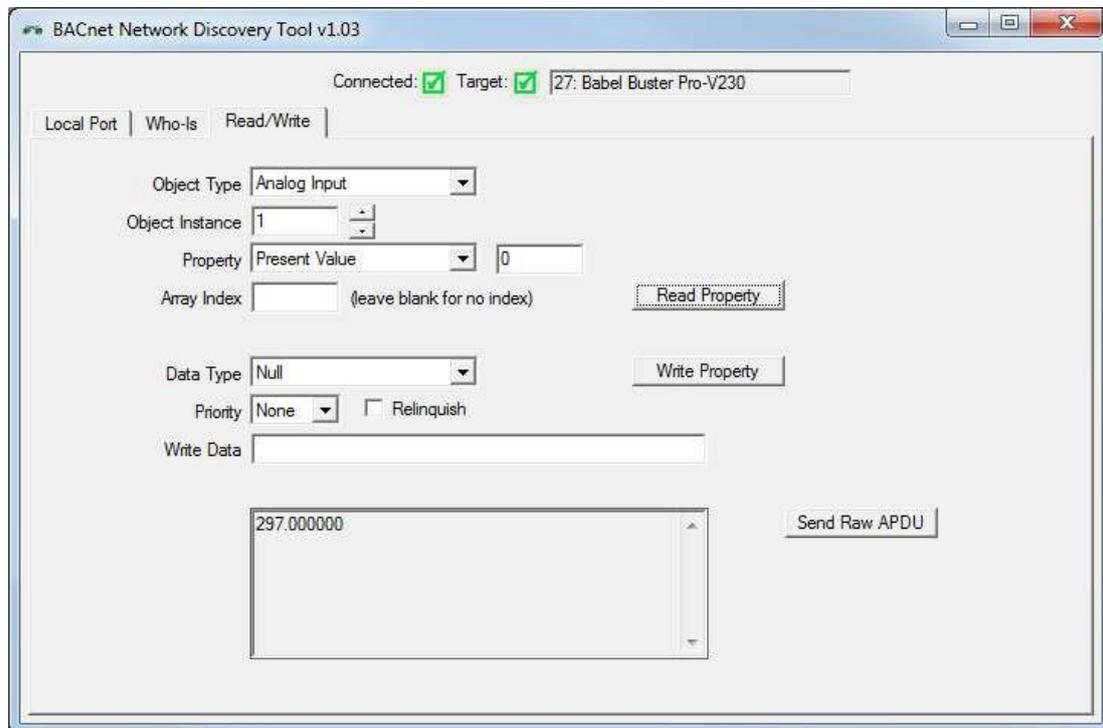


Click the Refresh button to cause the discovery tool to query every responding device to read object model and device object name from each of them.

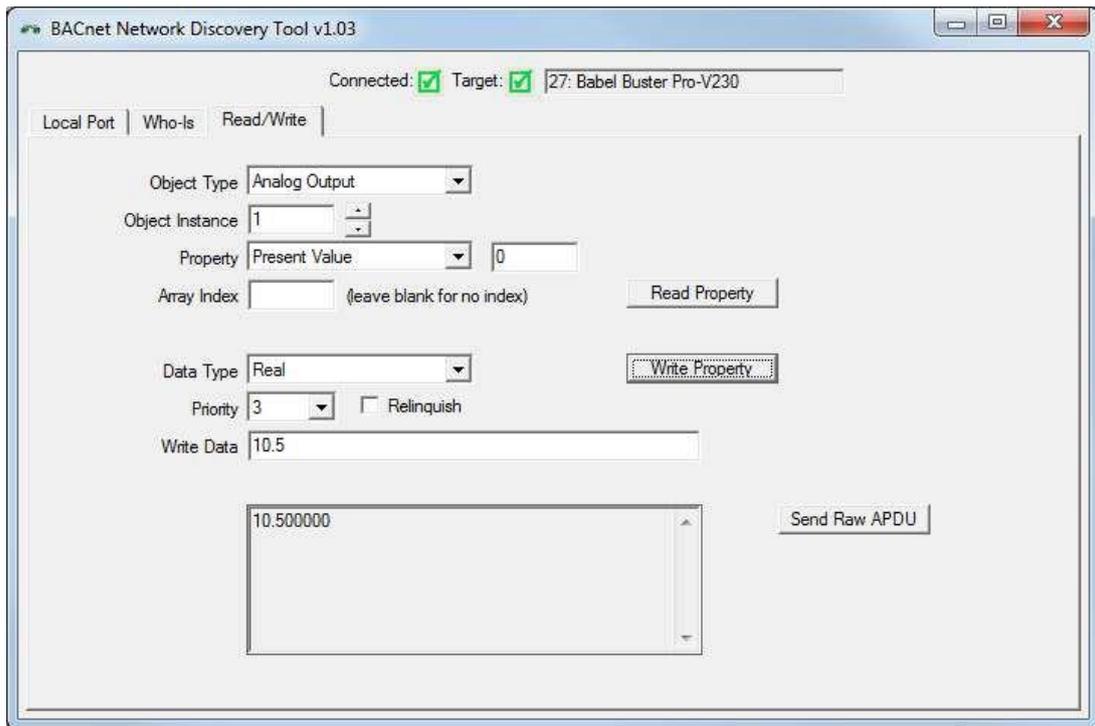
Double click on the device you wish to query further. It will now appear as the Target.



You can read and write properties in any of the standard objects typically used in any Control Solutions device and in most other devices. Select object type, instance, and property to read data by clicking the Read Property button.



In addition to selecting the same parameters you would for reading, select data type, priority if writing to a commandable object, and data value to write that property by clicking the Write Property button.

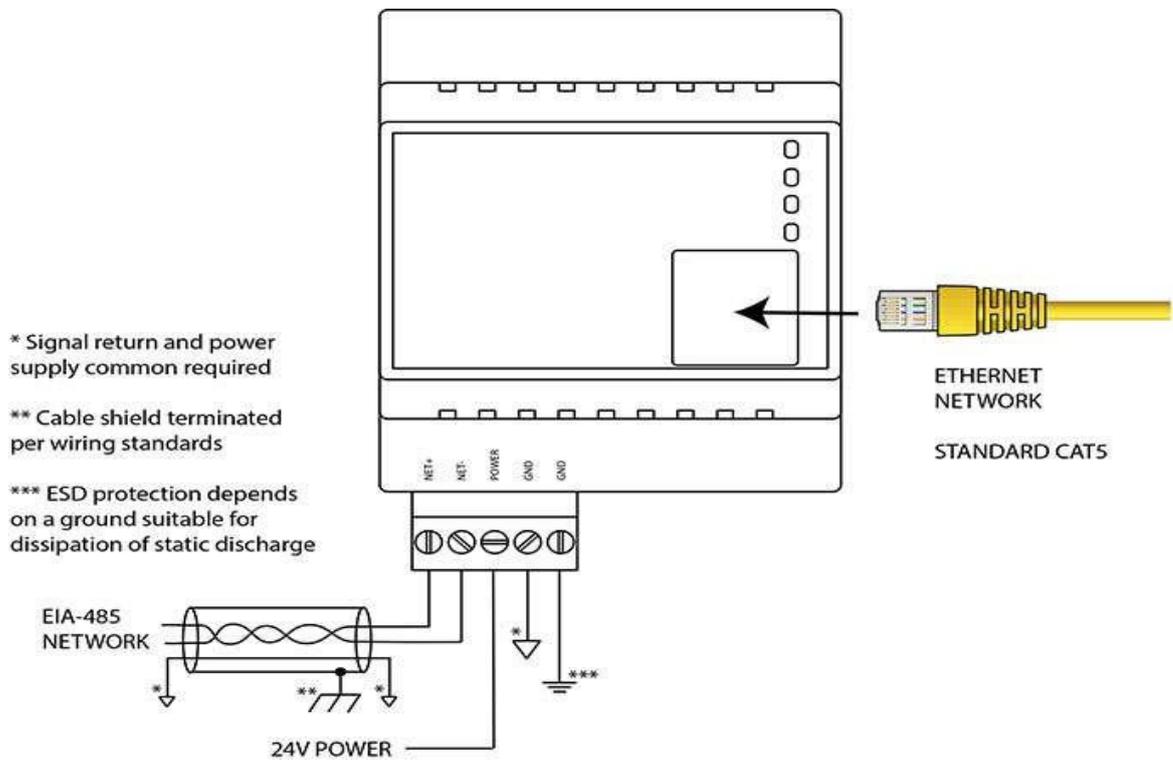


6 Appendices

6.1 Appendix A: Hardware Details

6.1.1 Wiring

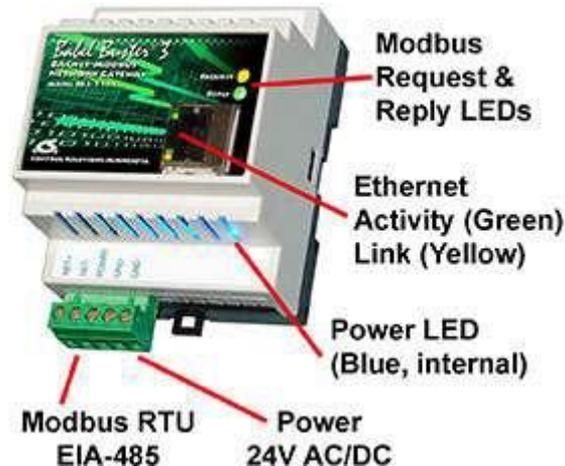
Wiring for the Babel Buster BB3-7101-TZ is illustrated below.



6.1.2 Front Panel LED Indicators

Power-up LED behaviour:

1. On power up, the 'Reply' LED will remain illuminated on solid red for about 20 seconds.
2. Following this the 'Request' and 'Reply' LEDs will do a "lamp test" where 'Request' is illuminated yellow and 'Reply' is illuminated red simultaneously for about 1 second, and then both Request and Reply turn green simultaneously for about 1 second.
3. The LEDs will then begin to operate according to their normal functionality.



Babel Buster BB3-7101-TZ 'Request' and 'Reply' LEDs reflect Modbus RTU traffic, and the Ethernet activity LED will indicate Ethernet/TCP traffic in general.

Babel Buster BB3-7101-TZ LEDs indicate as follows (LEDs are bi-colour):

Request	Flashes yellow each time a request is sent when operating as Modbus Master, or each time a request is received when operating as Modbus Slave.
Reply	Operating as Modbus Master, flashes green each time a good response is received, or red when an error code is received, the request times out, or there is a flaw in the response such as CRC error. Operating as Modbus Slave, flashes green each time a good response is sent, or red if an exception code is sent (meaning the received request resulted in an error).
Ethernet Activity	Green LED is illuminated solidly during portions of the boot-up process, and then flashes briefly when Ethernet network traffic is detected.
Ethernet Link	Yellow LED indicates an Ethernet link is present. This indicator will illuminate if a link is present regardless of processor or network activity. If not illuminated, check network wiring.
Status	Blue LED (internal) illuminated any time power is present and internal power supply is functioning.

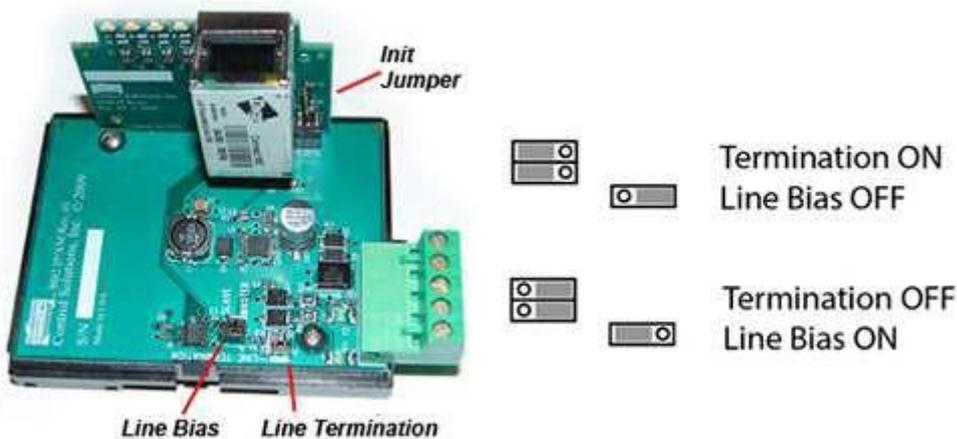
6.1.3 RS-485 Line Termination & Bias

Enable line termination only when this device is placed at the end of the network. Termination should only be enabled at two points on the network, and these two points must be specifically the end points.

Enable line bias when needed. Line bias should only be enabled at one point on the network, and does not have to be the end point. Line bias holds the line in a known neutral state when no devices are transmitting. Without bias, the transition from offline to online by a transmitter can look like a false start bit and cause loss of communication.

The line conditioning options are enabled when the respective shunt is moved to the position indicated by the diagrams below.

Jumper locations for Babel Buster BB3-7101-TZ:



6.2 Appendix B: BACnet Object Property Codes

BACnet property type codes may be found in your copy of the BACnet protocol specification, ANSI/ASHRAE Standard 135. That document is copyrighted, but the C enumeration shown below for reference is taken from open source code available under GPL at <http://sourceforge.net>, and provides essentially the same information (copyrighted by Steve Karg, licensed under GPL as noted at <http://sourceforge.net>).

typedef enum

```
PROP_ACKED_TRANSITIONS = 0,  
PROP_ACK_REQUIRED = 1,  
PROP_ACTION = 2,  
PROP_ACTION_TEXT = 3,  
PROP_ACTIVE_TEXT = 4,  
PROP_ACTIVE_VT_SESSIONS = 5,  
PROP_ALARM_VALUE = 6,  
PROP_ALARM_VALUES = 7,  
PROP_ALL = 8,  
PROP_ALL_WRITES_SUCCESSFUL = 9,
```

PROP_APDU_SEGMENT_TIMEOUT = 10,
PROP_APDU_TIMEOUT = 11,
PROP_APPLICATION_SOFTWARE_VERSION = 12,
PROP_ARCHIVE = 13,
PROP_BIAS = 14,
PROP_CHANGE_OF_STATE_COUNT = 15,
PROP_CHANGE_OF_STATE_TIME = 16,
PROP_NOTIFICATION_CLASS = 17,
PROP_BLANK_1 = 18,
PROP_CONTROLLED_VARIABLE_REFERENCE = 19,
PROP_CONTROLLED_VARIABLE_UNITS = 20,
PROP_CONTROLLED_VARIABLE_VALUE = 21,
PROP_COV_INCREMENT = 22,
PROP_DATE_LIST = 23,
PROP_DAYLIGHT_SAVINGS_STATUS = 24,
PROP_DEADBAND = 25,
PROP_DERIVATIVE_CONSTANT = 26,
PROP_DERIVATIVE_CONSTANT_UNITS = 27,
PROP_DESCRIPTION = 28,
PROP_DESCRIPTION_OF_HALT = 29,
PROP_DEVICE_ADDRESS_BINDING = 30,
PROP_DEVICE_TYPE = 31,
PROP_EFFECTIVE_PERIOD = 32,
PROP_ELAPSED_ACTIVE_TIME = 33,
PROP_ERROR_LIMIT = 34,
PROP_EVENT_ENABLE = 35,
PROP_EVENT_STATE = 36,
PROP_EVENT_TYPE = 37,
PROP_EXCEPTION_SCHEDULE = 38,
PROP_FAULT_VALUES = 39,
PROP_FEEDBACK_VALUE = 40,
PROP_FILE_ACCESS_METHOD = 41,
PROP_FILE_SIZE = 42,
PROP_FILE_TYPE = 43,
PROP_FIRMWARE_REVISION = 44,
PROP_HIGH_LIMIT = 45,
PROP_INACTIVE_TEXT = 46,
PROP_IN_PROCESS = 47,
PROP_INSTANCE_OF = 48,
PROP_INTEGRAL_CONSTANT = 49,
PROP_INTEGRAL_CONSTANT_UNITS = 50,
PROP_ISSUE_CONFIRMED_NOTIFICATIONS = 51,
PROP_LIMIT_ENABLE = 52,
PROP_LIST_OF_GROUP_MEMBERS = 53,

PROP_LIST_OF_OBJECT_PROPERTY_REFERENCES = 54,
PROP_LIST_OF_SESSION_KEYS = 55,
PROP_LOCAL_DATE = 56,
PROP_LOCAL_TIME = 57,
PROP_LOCATION = 58,
PROP_LOW_LIMIT = 59,
PROP_MANIPULATED_VARIABLE_REFERENCE = 60,
PROP_MAXIMUM_OUTPUT = 61,
PROP_MAX_APDU_LENGTH_ACCEPTED = 62,
PROP_MAX_INFO_FRAMES = 63,
PROP_MAX_MASTER = 64,
PROP_MAX_PRES_VALUE = 65,
PROP_MINIMUM_OFF_TIME = 66,
PROP_MINIMUM_ON_TIME = 67,
PROP_MINIMUM_OUTPUT = 68,
PROP_MIN_PRES_VALUE = 69,
PROP_MODEL_NAME = 70,
PROP_MODIFICATION_DATE = 71,
PROP_NOTIFY_TYPE = 72,
PROP_NUMBER_OF_APDU_RETRIES = 73,
PROP_NUMBER_OF_STATES = 74,
PROP_OBJECT_IDENTIFIER = 75,
PROP_OBJECT_LIST = 76,
PROP_OBJECT_NAME = 77,
PROP_OBJECT_PROPERTY_REFERENCE = 78,
PROP_OBJECT_TYPE = 79,
PROP_OPTIONAL = 80,
PROP_OUT_OF_SERVICE = 81,
PROP_OUTPUT_UNITS = 82,
PROP_EVENT_PARAMETERS = 83,
PROP_POLARITY = 84,
PROP_PRESENT_VALUE = 85,
PROP_PRIORITY = 86,
PROP_PRIORITY_ARRAY = 87,
PROP_PRIORITY_FOR_WRITING = 88,
PROP_PROCESS_IDENTIFIER = 89,
PROP_PROGRAM_CHANGE = 90,
PROP_PROGRAM_LOCATION = 91,
PROP_PROGRAM_STATE = 92,
PROP_PROPORTIONAL_CONSTANT = 93,
PROP_PROPORTIONAL_CONSTANT_UNITS = 94,
PROP_PROTOCOL_CONFORMANCE_CLASS = 95,
/ deleted in version 1 revision 2 */*
PROP_PROTOCOL_OBJECT_TYPES_SUPPORTED = 96,

PROP_PROTOCOL_SERVICES_SUPPORTED = 97,
PROP_PROTOCOL_VERSION = 98,
PROP_READ_ONLY = 99,
PROP_REASON_FOR_HALT = 100,
PROP_RECIPIENT = 101,
PROP_RECIPIENT_LIST = 102,
PROP_RELIABILITY = 103,
PROP_RELINQUISH_DEFAULT = 104,
PROP_REQUIRED = 105,
PROP_RESOLUTION = 106,
PROP_SEGMENTATION_SUPPORTED = 107,
PROP_SETPOINT = 108,
PROP_SETPOINT_REFERENCE = 109,
PROP_STATE_TEXT = 110,
PROP_STATUS_FLAGS = 111,
PROP_SYSTEM_STATUS = 112,
PROP_TIME_DELAY = 113,
PROP_TIME_OF_ACTIVE_TIME_RESET = 114,
PROP_TIME_OF_STATE_COUNT_RESET = 115,
PROP_TIME_SYNCHRONIZATION_RECIPIENTS = 116,
PROP_UNITS = 117,
PROP_UPDATE_INTERVAL = 118,
PROP_UTC_OFFSET = 119,
PROP_VENDOR_IDENTIFIER = 120,
PROP_VENDOR_NAME = 121,
PROP_VT_CLASSES_SUPPORTED = 122,
PROP_WEEKLY_SCHEDULE = 123,
PROP_ATTEMPTED_SAMPLES = 124,
PROP_AVERAGE_VALUE = 125,
PROP_BUFFER_SIZE = 126,
PROP_CLIENT_COV_INCREMENT = 127,
PROP_COV_RESUBSCRIPTION_INTERVAL = 128,
PROP_CURRENT_NOTIFY_TIME = 129,
PROP_EVENT_TIME_STAMPS = 130,
PROP_LOG_BUFFER = 131,
PROP_LOG_DEVICE_OBJECT = 132, */* The enable property is
renamed from log-enable in Addendum b to ANSI/ASHRAE 135-
2004(135b-2) */*
PROP_ENABLE = 133,
PROP_LOG_INTERVAL = 134,
PROP_MAXIMUM_VALUE = 135,
PROP_MINIMUM_VALUE = 136,
PROP_NOTIFICATION_THRESHOLD = 137,
PROP_PREVIOUS_NOTIFY_TIME = 138,

PROP_PROTOCOL_REVISION = 139,
PROP_RECORDS_SINCE_NOTIFICATION = 140,
PROP_RECORD_COUNT = 141,
PROP_START_TIME = 142,
PROP_STOP_TIME = 143,
PROP_STOP_WHEN_FULL = 144,
PROP_TOTAL_RECORD_COUNT = 145,
PROP_VALID_SAMPLES = 146,
PROP_WINDOW_INTERVAL = 147,
PROP_WINDOW_SAMPLES = 148,
PROP_MAXIMUM_VALUE_TIMESTAMP = 149,
PROP_MINIMUM_VALUE_TIMESTAMP = 150,
PROP_VARIANCE_VALUE = 151,
PROP_ACTIVE_COV_SUBSCRIPTIONS = 152,
PROP_BACKUP_FAILURE_TIMEOUT = 153,
PROP_CONFIGURATION_FILES = 154,
PROP_DATABASE_REVISION = 155,
PROP_DIRECT_READING = 156,
PROP_LAST_RESTORE_TIME = 157,
PROP_MAINTENANCE_REQUIRED = 158,
PROP_MEMBER_OF = 159,
PROP_MODE = 160,
PROP_OPERATION_EXPECTED = 161,
PROP_SETTING = 162,
PROP_SILENCED = 163,
PROP_TRACKING_VALUE = 164,
PROP_ZONE_MEMBERS = 165,
PROP_LIFE_SAFETY_ALARM_VALUES = 166,
PROP_MAX_SEGMENTS_ACCEPTED = 167,
PROP_PROFILE_NAME = 168,
PROP_AUTO_SLAVE_DISCOVERY = 169,
PROP_MANUAL_SLAVE_ADDRESS_BINDING = 170,
PROP_SLAVE_ADDRESS_BINDING = 171,
PROP_SLAVE_PROXY_ENABLE = 172,
PROP_LAST_NOTIFY_TIME = 173,
PROP_SCHEDULE_DEFAULT = 174,
PROP_ACCEPTED_MODES = 175,
PROP_ADJUST_VALUE = 176,
PROP_COUNT = 177,
PROP_COUNT_BEFORE_CHANGE = 178,
PROP_COUNT_CHANGE_TIME = 179,
PROP_COV_PERIOD = 180,
PROP_INPUT_REFERENCE = 181,
PROP_LIMIT_MONITORING_INTERVAL = 182,

```

PROP_LOGGING_DEVICE = 183,
PROP_LOGGING_RECORD = 184,
PROP_PRESCALE = 185,
PROP_PULSE_RATE = 186,
PROP_SCALE = 187,
PROP_SCALE_FACTOR = 188,
PROP_UPDATE_TIME = 189,
PROP_VALUE_BEFORE_CHANGE = 190,
PROP_VALUE_SET = 191,
PROP_VALUE_CHANGE_TIME = 192, /* enumerations 193-206 are new
*/
PROP_ALIGN_INTERVALS = 193,
PROP_GROUP_MEMBER_NAMES = 194,
PROP_INTERVAL_OFFSET = 195,
PROP_LAST_RESTART_REASON = 196,
PROP_LOGGING_TYPE = 197,
PROP_MEMBER_STATUS_FLAGS = 198,
PROP_NOTIFICATION_PERIOD = 199,
PROP_PREVIOUS_NOTIFY_RECORD = 200,
PROP_REQUESTED_UPDATE_INTERVAL = 201,
PROP_RESTART_NOTIFICATION_RECIPIENTS = 202,
PROP_TIME_OF_DEVICE_RESTART = 203,
PROP_TIME_SYNCHRONIZATION_INTERVAL = 204,
PROP_TRIGGER = 205,
PROP.UTC_TIME_SYNCHRONIZATION_RECIPIENTS = 206, /*
enumerations 207-211 are used in Addendum d to ANSI/ASHRAE
135-2004 */
PROP_NODE_SUBTYPE = 207,
PROP_NODE_TYPE = 208,
PROP_STRUCTURED_OBJECT_LIST = 209,
PROP_SUBORDINATE_ANNOTATIONS = 210,
PROP_SUBORDINATE_LIST = 211, /* enumerations 212-225 are used
in Addendum e to ANSI/ASHRAE 135-2004 */
PROP_ACTUAL_SHED_LEVEL = 212,
PROP_DUTY_WINDOW = 213,
PROP_EXPECTED_SHED_LEVEL = 214,
PROP_FULL_DUTY_BASELINE = 215, /* enumerations 216-217 are
used in Addendum i to ANSI/ASHRAE 135-2004 */
PROP_BLINK_PRIORITY_THRESHOLD = 216,
PROP_BLINK_TIME = 217, /* enumerations 212-225 are used in
Addendum e to ANSI/ASHRAE 135-2004 */
PROP_REQUESTED_SHED_LEVEL = 218,
PROP_SHED_DURATION = 219,
PROP_SHED_LEVEL_DESCRIPTIONS = 220,

```

PROP_SHED_LEVELS = 221,
PROP_STATE_DESCRIPTION = 222, */* enumerations 223-225 are used
in Addendum i to ANSI/ASHRAE 135-2004 */*
PROP_FADE_TIME = 223,
PROP_LIGHTING_COMMAND = 224,
PROP_LIGHTING_COMMAND_PRIORITY = 225, */* enumerations 226-235
are used in Addendum f to ANSI/ASHRAE 135-2004 */*
PROP_DOOR_ALARM_STATE = 226,
PROP_DOOR_EXTENDED_PULSE_TIME = 227,
PROP_DOOR_MEMBERS = 228,
PROP_DOOR_OPEN_TOO_LONG_TIME = 229,
PROP_DOOR_PULSE_TIME = 230,
PROP_DOOR_STATUS = 231,
PROP_DOOR_UNLOCK_DELAY_TIME = 232,
PROP_LOCK_STATUS = 233,
PROP_MASKED_ALARM_VALUES = 234,
PROP_SECURED_STATUS = 235, */* enumerations 236-243 are used in
Addendum i to ANSI/ASHRAE 135-2004 */*
PROP_OFF_DELAY = 236,
PROP_ON_DELAY = 237,
PROP_POWER = 238,
PROP_POWER_ON_VALUE = 239,
PROP_PROGRESS_VALUE = 240,
PROP_RAMP_RATE = 241,
PROP_STEP_INCREMENT = 242,
PROP_SYSTEM_FAILURE_VALUE = 243, */* enumerations 244-311 are
used in Addendum j to ANSI/ASHRAE 135-2004 */*
PROP_ABSENTEE_LIMIT = 244,
PROP_ACCESS_ALARM_EVENTS = 245,
PROP_ACCESS_DOORS = 246,
PROP_ACCESS_EVENT = 247,
PROP_ACCESS_EVENT_AUTHENTICATION_FACTOR = 248,
PROP_ACCESS_EVENT_CREDENTIAL = 249,
PROP_ACCESS_EVENT_TIME = 250,
PROP_ACCESS_RULES = 251,
PROP_ACCESS_RULES_ENABLE = 252,
PROP_ACCESS_TRANSACTION_EVENTS = 253,
PROP_ACCOMPANIED = 254,
PROP_ACTIVATION_TIME = 255,
PROP_ACTIVE_AUTHENTICATION_POLICY = 256,
PROP_ASSIGNED_ACCESS_RIGHTS = 257,
PROP_AUTHENTICATION_FACTOR_INPUT_LIST = 258,
PROP_AUTHENTICATION_FACTORS = 259,
PROP_AUTHENTICATION_POLICY_LIST = 260,

PROP_AUTHENTICATION_POLICY_NAMES = 261,
PROP_AUTHORIZATION_MODE = 262,
PROP_BELONGS_TO = 263,
PROP_CREDENTIAL_DISABLE = 264,
PROP_CREDENTIAL_STATUS = 265,
PROP_CREDENTIALS = 266,
PROP_CREDENTIALS_IN_ZONE = 267,
PROP_DAYS_REMAINING = 268,
PROP_ENTRY_POINTS = 269,
PROP_EXIT_POINTS = 270,
PROP_EXPIRY_TIME = 271,
PROP_EXTENDED_TIME_ENABLE = 272,
PROP_FAILED_ATTEMPT_EVENTS = 273,
PROP_FAILED_ATTEMPTS = 274,
PROP_FAILED_ATTEMPTS_TIME = 275,
PROP_FORMAT_CLASS_SUPPORTED = 276,
PROP_FORMAT_TYPE = 277,
PROP_LAST_ACCESS_EVENT = 278,
PROP_LAST_ACCESS_POINT = 279,
PROP_LAST_CREDENTIAL_ADDED = 280,
PROP_LAST_CREDENTIAL_ADDED_TIME = 281,
PROP_LAST_CREDENTIAL_REMOVED = 282,
PROP_LAST_CREDENTIAL_REMOVED_TIME = 283,
PROP_LAST_USE_TIME = 284,
PROP_LOCKDOWN = 285,
PROP_LOCKDOWN_RELINQUISH_TIME = 286,
PROP_MASTER_EXEMPTION = 287,
PROP_MAX_FAILED_ATTEMPTS = 288,
PROP_MEMBERS = 289,
PROP_MASTER_POINT = 290,
PROP_NUMBER_OF_AUTHENTICATION_POLICIES = 291,
PROP_OCCUPANCY_COUNT = 293,
PROP_OCCUPANCY_COUNT_ENABLE = 294,
PROP_OCCUPANCY_COUNT_EXEMPTION = 295,
PROP_OCCUPANCY_LOWER_THRESHOLD = 296,
PROP_OCCUPANCY_LOWER_THRESHOLD_ENFORCED = 297,
PROP_OCCUPANCY_STATE = 298,
PROP_OCCUPANCY_UPPER_LIMIT = 299,
PROP_OCCUPANCY_UPPER_LIMIT_ENFORCED = 300,
PROP_PASSBACK_EXEMPTION = 301,
PROP_PASSBACK_MODE = 302,
PROP_PASSBACK_TIMEOUT = 303,
PROP_POSITIVE_ACCESS_RULES = 304,
PROP_READ_STATUS = 305,

```

PROP_REASON_FOR_DISABLE = 306,
PROP_THREAT_AUTHORITY = 307,
PROP_THREAT_LEVEL = 308,
PROP_TRACE_FLAG = 309,
PROP_TRANSACTION_NOTIFICATION_CLASS = 310,
PROP_USER_EXTERNAL_IDENTIFIER = 311,
/* enumerations 312-313 are used in Addendum k to ANSI/ASHRAE
135-2004 */
PROP_CHARACTER_SET = 312,
PROP_STRICT_CHARACTER_MODE = 313,
/* enumerations 312-313 are used in Addendum k to ANSI/ASHRAE
135-2004 */
PROP_BACKUP_AND_RESTORE_STATE = 314,
PROP_BACKUP_PREPARATION_TIME = 315,
PROP_RESTORE_PREPARATION_TIME = 316,
/* enumerations 317-323 are used in Addendum j to ANSI/ASHRAE
135-2004 */
PROP_USER_INFORMATION_REFERENCE = 317,
PROP_USER_NAME = 318,
PROP_USER_TYPE = 319,
PROP_USES_REMAINING = 320,
PROP_VENDOR_FORMAT_IDENTIFIER = 321,
PROP_ZONE_FROM = 322,
PROP_ZONE_TO = 323,
/* enumerations 324-325 are used in Addendum i to ANSI/ASHRAE
135-2004 */
PROP_BINARY_ACTIVE_VALUE = 324,
PROP_BINARY_INACTIVE_VALUE = 325
/* The special property identifiers all, optional, and
required */
/* are reserved for use in the ReadPropertyConditional and */
/* ReadPropertyMultiple services or services not defined in
this standard. */
/* Enumerated values 0-511 are reserved for definition by
ASHRAE. */
/* Enumerated values 512-4194303 may be used by others subject
to the */
/* procedures and constraints described in Clause 23. */

```

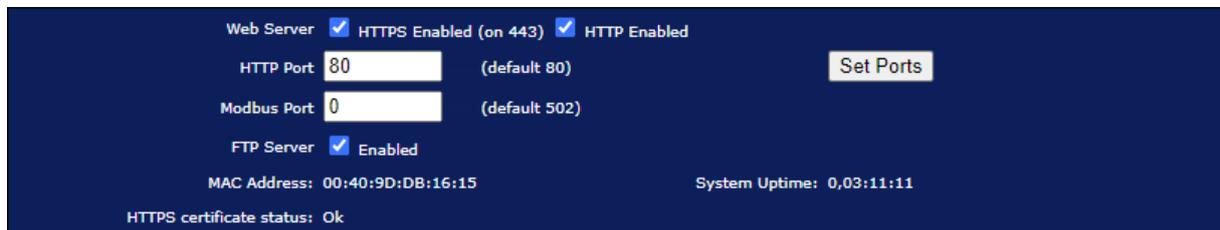
6.3 Appendix C: SSL Certificates for Secure Web (HTTPS)

The secure web server (HTTPS) requires SSL certificates in order to establish secure connections. The HTTPS certificates are only required if HTTPS is enabled on the Network configuration page in the Babel Buster BB3-7101-TZ.

6.3.1 X.509 Auto-Certificate Generation

The Babel Buster BB3-7101-TZ Gateway will automatically generate X.509 certificates if no external certificates are found or could not be loaded correctly. These will be generated one time and saved in the Flash file system for subsequent reuse.

When the self-generated X.509 certificates are in use, this will be indicated at the bottom of the Network configuration page.



The screenshot shows a configuration interface with a dark blue background. At the top, it displays 'Web Server' with two checked checkboxes: 'HTTPS Enabled (on 443)' and 'HTTP Enabled'. Below this, there are two input fields: 'HTTP Port' with the value '80' and '(default 80)', and 'Modbus Port' with the value '0' and '(default 502)'. A 'Set Ports' button is located to the right of the HTTP Port field. Further down, 'FTP Server' is shown with a checked checkbox and the label 'Enabled'. At the bottom left, the 'MAC Address' is '00:40:9D:DB:16:15', and at the bottom right, the 'System Uptime' is '0,03:11:11'. At the very bottom, the 'HTTPS certificate status' is 'Ok'.