



# **User Guide**

## Babel Buster 2

### Model BB2-7010 BACnet Modbus Gateway Rev. 1.3 – Oct. 2015

© 2015 Control Solutions, Inc.

### **BB2-7010 User Guide Contents**

### 1 Introduction

- 1.1 How to Use This Guide
- 1.2 Overview of Gateway Devices
- 1.3 Important Safety Notice
- 1.4 Warranty

### 2 Connecting BB2-7010 for the First Time

3 Minimum BB2-7010 Gateway Setup

### 4 Using the BB2-7010 as a BACnet Server

### 5 Configuring BB2-7010 as a Modbus RTU Master

- 5.1 Modbus RTU Device Configuration
- 5.2 Modbus RTU Master Read Maps
- 5.3 Modbus RTU Master Write Maps
- 5.4 Modbus RTU Master Data Displayed Per Slave
- 5.5 Modbus RTU Errors

#### 6. Configuring BB2-7010 as a Modbus TCP Client

- 6.1 Modbus TCP Device Configuration
- 6.2 Modbus TCP Client Read Maps
- 6.3 Modbus TCP Client Write Maps
- 6.4 Modbus TCP Errors

#### 7 Using the BB2-7010 as a BACnet Client

- 7.1 BACnet Device Configuration
- 7.2 BACnet Client Read Maps
- 7.3 BACnet Client Write Maps
- 7.4 BACnet Errors

#### 8 Modbus RTU Slave Configuration

- 8.1 Modbus RTU Device Configuration
- 8.2 Modbus RTU Slave Register Mapping

#### 9 Modbus TCP Server Configuration

- 9.1 Modbus TCP Device Configuration
- 9.2 Modbus TCP Register Mapping
- 9.3 Modbus Virtual Device Register Mapping

### 10 SNMP Server Configuration

- 10.1 Local SNMP MIB
- 10.2 Trap Thresholds
- 10.3 SNMP Trap Destinations

#### 11 SNMP Client Configuration

- 11.1 SNMP Device Configuration
- 11.2 SNMP Client Read Maps
- 11.3 SNMP Client Write Maps
- 11.4 SNMP Errors

### 12 HTTP Client

### 13 BBMD Setup

- 14 Object Properties
- 14.1 Data Object Properties (Analog, Binary, Multi-state)
- 14.2 Device Object Properties
- 15 Trouble Shooting

### 16 WiFi Sensor Setup - PointSix, AirTest

16.1 Sensor Network Settings

- 16.2 Configuration
- 16.3 Sensor Data Page

# 17WiFi Sensor Setup - Veris Industries17.1Sensor Network Settings

- Sensor Network Settings
- 17.2 Configuration
- Sensor Data Page 17.3

### 18 WiFi Sensor Setup - Functional Devices RIB

- Sensor Network Settings 18.1
- 18.2 Configuration
- Sensor Data Page 18.3
- 18.4 Sensor Diagnostic Page

#### Appendix A Hardware Details

- Wiring A.1
- A.2 Front Panel LED Indicators
- RS-485 Line Termination and Bias A.3



# **1. Introduction**

## 1.1 How to Use This Guide

The first few sections of this user guide provide background information on how the gateway works, and an overview of the configuration process. The next several sections are guides for each of the tabs found in the web interface in the gateway which is accessed by opening a web browser and browsing to the IP address of the device.

You should at least read the overview sections to gain an understanding of how the gateway functions. You can use the remaining sections as reference material to look up as needed. There is a "Quick Help" section at the bottom of each web page in the gateway which is generally sufficient for quick reference in setting up the gateway.

## 1.2 Overview of Gateway Devices

The Babel Buster BB2-7010 is a BACnet to Modbus gateway. It may be used as BACnet IP client and server, Modbus TCP client and server, and Modbus RTU master or slave. The BB2-7010-02 may also be used as SNMP client and server. The BB2-7010-06, -07, and -08 models add gateway capability for specific types of WiFi sensors (see section 16, 17, 18).



The most common application for the BB2-7010 is interfacing a Modbus RTU device to a BACnet IP network. The BB2-7010 will automatically poll the Modbus RTU device, and store the content if its

registers in BACnet objects you assign. The BACnet system may then use standard BACnet services such as Read Property to access the content of the Modbus registers. The BB2-7010 will also accept COV subscriptions such that other devices will receive a COV notification when the content of a Modbus register changes.

The BB2-7010 can be also configured as a Modbus RTU slave. This is useful when a PLC wants to write data to the BB2-7010, thereby making the PLC's data available as BACnet object properties.

The BB2-7010 can be configured as a BACnet IP client. This means the BB2-7010 will be reading and writing properties in other BACnet devices, storing copies of their object's Present Value in the BB2-7010. The stored values may later be accessed by Modbus or SNMP.

## 1.3 Important Safety Notice

Proper system design is required for reliable and safe operation of distributed control systems incorporating any Control Solutions product. 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 ANY 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.

## 1.4 Warranty

This software and documentation 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 documentation or in the product(s) and/or the program(s) described in this documentation at any time. This product could include software bugs, technical inaccuracies, typographical errors, and the like. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the software.



# 2. Connecting the BB2-7010 for the First Time

Follow these steps to make the initial connection to the BB2-7010.

(a) Connect power. Apply +12 to +24VDC or 24VAC to the terminal marked "POWER", and common or ground the the terminal marked "GND".

(b) Connect a CAT5 cable between the RJ-45 jack on the top and your network switch or hub. You cannot connect directly to your PC unless you use a "crossover" cable.

(c) Apply power. A blue LED inside the case should light indicating power is present. If the yellow LED on the RJ45 jack is not on, check your Ethernet cable connections. Both green and yellow LEDs on the RJ45 jack will be on solid for a 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.

(d) The default IP address as shipped is 10.0.0.101. If your PC is not already on the 10.0.0.0 domain, you will need to add a route on your PC. Do this by opening a command prompt. First type "ipconfig" and note the IP address listed. This is your PC's IP address. Now type the command

route add 10.0.0.0 mask 255.255.255.0 1.2.3.4

but substitute your PC's IP address for 1.2.3.4.

This generally works, but if this fails, you will need to temporarily change your computer's IP address to a fixed address that starts with 10.0.0. and ends with anything but 101.



### Quick Help

Click any tab above to log in. If you are not already logged in, you will be asked for your user name and password. You will need these in order to log in.

To log out, simply close your browser. IMPORTANT: If you have made configuration changes that you want to save permanently, go to the System->Setup->Config File page and click "save". Changes made by clicking "update" are only temporary until you save changes permanently in your configuration file.

(e) Open your browser, and enter "http://10.0.0.101/" in the address window. You should see a page with the "Babel Buster BB2-7010" header shown above. From this point, you will find help on each page in the web site contained within the product.

(f) When you click on any of the page tabs such as System Setup, you will be asked for a user name and password. The default login is user name "system" with password "admin". You can also log in as "root" using password "buster". You should log in as "root" if you will be changing the IP address.

1001 BACNET-MODBUS 1001 NETWORK GATEWAY MODEL BB2-7010-01 MINNESOTA									
Data Objects	Modbus	BACnet	System Setup						
Setup	HT TP Clien	ıt 🚶							
Config File	BACnet IP Port	BBMD	Local Host	User					
This page allows you to ch accessing this server.	ange this device's IP a	ddress, and select whethe	er double registers are swapp	ed when returned to a remote client					
IP Address	192 168 1 64	192,168,1,64	Beview	IP					
Subnet Mask	255.255.255.0	255.255.255.0	Change						
Gateway	192.168.1.1	192.168.1.1							

(g) To can change the IP address of the BB2-7010, go to the Local Host page under System :: Setup. The following page should appear. Change the IP address, and subnet mask and gateway if applicable. Click Change IP to save the changes. The process of programming this into Flash takes around half a minute. The new IP address only takes effect following the next system restart or power cycle.

0001100 1100110 101101 0100101 0100101	achel Buste Achet-Modbus etwork gatewa odel bb2-7010-01	n 2 r		Control-Sol	UTIONS, INC. Minnesota
Data Objects	Modbus		BACnet	System Setup	
Setu	p	HTTP Client			
Config File	BACnet	P Port	BBMD	Local Host	User
This page allow	s you to manage conf	guration files.			
<u> </u>	Store configuration	to Flash file selecto	ed from directory, or to ne	w file if checked.	
Save	Local file directory	sootContig.xml		Delete	
Boot	Boot configuration E	BootConfig.xml		Confirm Rest	art
Upload	Upload Configuratio	n File	Browse		

(h) Most changes are stored in an XML configuration file in the device's Flash file system. Only a few are stored differently, and the IP address is one of those. Normally, clicking Update on any configuration page only stores that configuration information to a temporary RAM copy of the configuration file. To make your changes other than IP address permanent, you must click Save on the Config File page (System :: Setup :: Config File).



# 3. Minimum BB2-7010 Gateway Setup

The BB2-7010 requires only minimal configuration to be useful in its simplest form. First, you must assign a device instance to the BB2-7010, and you do this via the BACnet IP Port page. You may leave all other settings at their default. You could leave the device instance at its default as well. The only real requirement is that you do not duplicate device instances.

10011001101100 10011 Bacnet-modeus 10010 Network Gatewa Model BB2-7010-01	n 2 *			Contro	ol-Solu	TIONS, INC. Minnesota
Data Objects Modbus		BACnet		System Set	up	
Setup	HTTP Client					
Config File BACnet	IP Port	BBMD	J	Local Host	ſ	User
BACnet IP Settings: Device Instanc Port (default 0xBAC0 = 47808	■ 64 0 47808	]			Local Network S Save	Refresh Settings
Device Descriptio	BB2-7010 Mo	dbus to BA	Cnet IP Gatewa	iy		
Device Locatio	USA	1				
APDU Timeou APDU Segment Timeou	t 3000 t 5000	]	Datab	APDU Retries base Revision	4	
Object Count Limits	Input Objects	1	Output Objects		Value Objects	
Analo	150 . 20	[150]	10	[10]	10	[10]
Binar Multi-Stat	2 30 2 30	[30]	10	[10]	10	[10]
Allow fault self-reset without Ack	. 💌			Enable BBMD		

The number of each type of available object is indicated here. Initially, there will be only 10 AI's and 1 each of the other object types. There is a pool of objects that may be shared among the different object types. The number of objects available is displayed at the "System Capacities" link on the home page of

the device. The number will range from 300 to 1000 depending on model.

The number displayed next to the input window is the object count that has been requested and will take effect upon the next restart. To request a different number, enter that number and click Save. Restart (or power cycle) the device to make the new object allocations take effect.

The check box for "allow fault self-reset without ack" means object reliability code and fault status will return to normal automatically after recovery from a communication fault such as "no response" (reply timed out). If this box is not checked, a BACnet client must read the reliability code to acknowledge the problem before the status will be reset. If the box is checked, fault indications will simply go away when the fault goes away.

The check box for "enable BBMD" is used to enable the BBMD feature of the device. When enabled, additional configuration on the BBMD tab should also be set up.



# 4. Using the BB2-7010 as a BACnet Server

The BB2-7010 contains a set of BACnet objects whose only purpose is to store copies of data obtained from other devices. This copy of data may then be queried by different devices, or written to different devices by the BB2-7010 client functions.

The collection of 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 BB2-7010 also contains a Device object which is shared with router functions. All of the remaining objects noted here are not used by routing functions.

Data may be placed in the local objects by other devices writing to the BB2-7010, or by the BB2-7010 querying other devices. When the BB2-7010 is configured to query other devices, these operations are defined by "read maps" and "write maps" associated with the respective client function (e.g. BACnet client, Modbus TCP client, SNMP client).

The following pages illustrate the Analog Input object pages and the Binary Output object pages. The remaining object pages found in the BB2-7010 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 locall force values, assign units or names, etc.

110 101 010	1001 Babel Buster 2 BACNET-MODBUS 1001 NETWORK GATEWAY MODEL BB2-7010-01 MINNESOTA										
Data	Objects	Modbus		BACnet	<u> </u>	System Se	etup	<u> </u>			
	Analog	Binaŋ	y	🚺 Mult	i-State	<u> </u>		<u> </u>			
Input	Objects	Output Objects	s	Value Objects	s						
This pa Analog	This page displays data as presently found in the local objects maintained by this device. Analog Input Objects Showing objects from 1 Refresh <prev next=""></prev>										
Object	Object Name		Out of Service	Present Value	Reliability	Status	Units				
Object	Object Name Volts		Out of Service N	Present Value 121,298	Reliability 0	Status 0,0,0,0	Units no_units				
Object	Object Name Volts Amps		Out of Service N N	Present Value 121.298 2.19448	Reliability 0 0	Status 0,0,0,0 0,0,0,0	Units no_units no_units				
Object <u> 1</u> <u> 2</u> <u> 3</u>	Object Name Volts Amps Hertz		Out of Service N N	Present Value 121.298 2.19448 60.0112	Reliability 0 0 0	Status 0,0,0,0 0,0,0,0 0,0,0,0	Units no_units no_units no_units				
Object 1 2 3 4	Object Name Volts Amps Hertz Watts		Out of Service N N N N	Present Value 121.298 2.19448 60.0112 263.053	Reliability 0 0 0	Status 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0	Units no_units no_units no_units no_units				
Object 1 2 3 4 5	Object Name Volts Amps Hertz Watts Temperature		Out of Service N N N N N	Present Value 121.298 2.19448 60.0112 263.053 76.7660	Reliability 0 0 0 0 0	Status 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0	Units no_units no_units no_units no_units no_units				
Object 1 2 3 4 5 6 -	Object Name Volts Amps Hertz Watts Temperature Humidity		Out of Service N N N N N N	Present Value 121.298 2.19448 60.0112 263.053 76.7660 80.8409	Reliability 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Status 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0	Units no_units no_units no_units no_units no_units				
Object 1 2 3 4 5 6 7 2 2 2 2 2 2 3 4 5 6 2 2 2 2 3 4 5 6 2 2 2 3 4 5 6 7 7 7 7 7 7 7 7 7	Object Name Volts Amps Hertz Watts Temperature Humidity Rack Temp		Out of Service N N N N N N	Present Value 121.298 2.19448 60.0112 263.053 76.7660 80.8409 98.8300	Reliability 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Status 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0	Units no_units no_units no_units no_units no_units no_units				
Object 1 2 3 4 5 6 7 8 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	Object Name Volts Amps Hertz Watts Temperature Humidity Rack Temp Analog Input 8		Out of Service N N N N N N N	Present Value 121.298 2.19448 60.0112 263.053 76.7660 80.8409 98.8300 0.00000	Reliability 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Status 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0	Units no_units no_units no_units no_units no_units no_units no_units				
Object 1 2 3 4 5 6 7 8 9 10	Object Name Volts Amps Hertz Watts Temperature Humidity Rack Temp Analog Input 8 Analog Input 9 Analog Input 10		Out of Service N N N N N N N N N	Present Value 121.298 2.19448 60.0112 263.053 76.7660 80.8409 98.8300 0.00000 0.00000	Reliability 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Status 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0	Units no_units no_units no_units no_units no_units no_units no_units no_units				
Object 1 2 3 4 5 6 7 8 9 10 11	Object Name Volts Amps Hertz Watts Temperature Humidity Rack Temp Analog Input 8 Analog Input 9 Analog Input 10 Analog Input 11		Out of Service N N N N N N N N N N N	Present Value 121.298 2.19448 60.0112 263.053 76.7660 80.8409 98.8300 0.00000 0.00000 0.00000 0.00000	Reliability 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Status 0,0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0	Units no_units no_units no_units no_units no_units no_units no_units no_units no_units no_units no_units				
Object  1 2 3 4 5 6 7 8 9 10 11 12	Object Name Volts Amps Hertz Watts Temperature Humidity Rack Temp Analog Input 8 Analog Input 9 Analog Input 10 Analog Input 11 Analog Input 12		Out of Service N N N N N N N N N N N	Present Value 121.298 2.19448 60.0112 263.053 76.7660 80.8409 98.8300 0.00000 0.00000 0.00000 0.00000	Reliability 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Status 0,0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0	Units no_units no_units no_units no_units no_units no_units no_units no_units no_units no_units no_units				
Object 1 2 3 4 5 6 7 8 2 10 11 12 13	Object Name Volts Amps Hertz Watts Temperature Humidity Rack Temp Analog Input 8 Analog Input 9 Analog Input 10 Analog Input 11 Analog Input 12 Analog Input 13		Out of Service N N N N N N N N N N N N	Present Value 121.298 2.19448 60.0112 263.053 76.7660 80.8409 98.8300 0.0000 0.00000 0.00000 0.00	Reliability 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Status 0,0,0,0,0 0,0,0,0,0 0,0,0,0,0 0,0,0,0,	Units no_units no_units no_units no_units no_units no_units no_units no_units no_units no_units no_units no_units no_units				
Object  1 2 3 4 5 6 7 8 9 10 11 12 13 14	Object Name Volts Amps Hertz Watts Temperature Humidity Rack Temp Analog Input 8 Analog Input 9 Analog Input 10 Analog Input 11 Analog Input 13 Analog Input 14		Out of Service N N N N N N N N N N N N	Present Value 121.298 2.19448 60.0112 263.053 76.7660 80.8409 98.8300 0.0000 0.0000	Reliability 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Status           0,0,0,0	Units no_units no_units no_units no_units no_units no_units no_units no_units no_units no_units no_units no_units no_units no_units				

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 the remote Modbus register via the map indicated by the Device Link. The Modbus device will be polled at the rate specified by the Read Map.

Out of Service means polling of the Modbus register 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 Modbus poll unless the object is out of service.

Reliability codes may be any of the following (7010-01):

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

Reliability codes may be any of the following (7010-02):

- 80: Local device, configuration property fault
- 81: Faulty packet
- 82: BACnet IP client, device timeout
- 83: BACnet IP client, error returned by server
- 84: SNMP client, no response from agent
- 85: SNMP client, unable to parse data
- 86: SNMP client, reply does not match request

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

Device link will indicate BAC or TCP, followed by R for read or W for write, and a number which is the rule number in the table of read or write rules for mapping external devices or objects to this BACnet object. The designation R means read from external device, and W means write to external device.

Click on the AI map number to get to the expanded view of the AI object as illustrated below.

1001101101101101101101101101101101100110110010000									
Data Objects	Modbus	BACnet	System Setup						
Analog	Binary	Multi-State			J				
Input Objects	Output Objects	Value Objects							
This page displays data as	presently found in the	local registers maintained b	y this device.						
Analog Input # 1				Update <prev ne<="" td=""><td>ext≻</td></prev>	ext≻				
Name: Volts Reliability: 0	Status: 0,0,0,0 De	vice Link: <u>RTU R1</u> Out of §	Service:						
Object name Volts		Force Prese	ent Value 120.185						
COV increment: 0.00000	Units: no_units		~						

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 object name may be changed here. BACnet units may be selected. Initial COV increment may be entered. When any of these are changed, be sure to save the updated configuration by clicking Save on the Config File page under System Setup.

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 be reading the remote Modbus register via the map indicated by the Device Link. The Modbus device will be polled at the rate specified by the Read Map.

Out of Service means polling of the Modbus register 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 Modbus poll unless the object is out of service.

The Binary Output Object page is illustrated below.

0001 1100 101 0100 0000	100110110110110110110110110110110110110									
Data	Data Objects Modbus			BACnet		System Se	rtup			
	Analog	Binar	У	Mult	ti-State					
Input	Objects	Output Object	s	Value Object	s					
This pa	age displays data a:	s presently found	in the lo	cal registers mai	ntained by th	nis device.				
Binary C	)utput Objects		s	howing objects fr	om 1		Update <prev next=""></prev>			
Object	Object Name		Out of Service	Present Value	Reliability	Status	Text			
<u>1</u>	Relay 1 Out		N	Active	0	0,0,0,0	Contact Closed			
2	Binary Output 2		N	Inactive	0	0,0,0,0				
<u>3</u>	Binary Output 3		N	Inactive	0	0,0,0,0				
4	Binary Output 4		N	Inactive	0	0,0,0,0				
<u>5</u>	Binary Output 5		N	Inactive	0	0,0,0,0				
<u>6</u>	Binary Output 6		N	Inactive	0	0,0,0,0				
Z	Binary Output 7		N	Inactive	0	0,0,0,0				
<u>8</u>	Binary Output 8		N	Inactive	0	0,0,0,0				
2	Binary Output 9		N	Inactive	0	0,0,0,0				
<u>10</u>	Binary Output 10		N	Inactive	0	0,0,0,0				

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.

The destination of data for a Binary Output object will be writing the remote Modbus register via the map indicated by the Device Link. The Modbus device will be updated upon change of source data and/or periodically as defined by the Write Map.

The Binary Output object is commandable, 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 Modbus register. If all values are relinquished, the relinquish default value will be written to the Modbus register.

Out of service means the Modbus register will not be written. 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 Modbus register.

Click on the object number in the left-hand column to get to the expanded view of the Binary Output object:

00011001101 110011 01101 Babel Bacnet- 010010 Network Model BB2	Buster 2 modbus k Gateway -7010-01		Control S	SOLUTION	NS, INC. INNESOTA
Data Objects	Modbus	BACnet	System Setup		
Analog	Binary	Multi-State	<u> </u>	ſ	Ì
Input Objects	Output Objects	Value Objects	J		
This page displays data a	as presently found in the l	ocal objects maintained by	this device.		
Binary Output # 1				Update	<prev next=""></prev>
Name: Relay 1 Out Relia	ability: 0     Status: 0,0,0,0	D Device Link: <u>TCP W1</u>	Out of Service:		_
Object name Relay 1 Out	t	Force Pres	ent Value 🛛 Active 🛛 🝸	3> Active	/
Active Text: Contact Clos	ed	Inctive Text: C	ontact Open	2> NULL	
Relinguish Default Inactiv	/e 💌			3> Active	
				4> NULL	
				6> NULL	
Quick Help				7> NULL	
Quick help				9> NULL	
The object name, units, v the list. Click Refresh to u	value, and status are show update the page, or Updat	vn for the object number er te to accept changes,	tered at the top of the	F 10> NULL	ext to scroll through
<b>The shipts of the shipts of t</b>		and the same of the		11> NULL	
one object name may be configuration by clicking S	cnanged nere, state text Save on the Config File pa	may be entered, when an ige under System Setup.	y of these are changed,	13> NULL	ne updated
The destination of data fo	or a Binary Output object (	will be writing the remote M	odbus register via the n		he Device Link. The
Modbus device will be upo	dated upon change of sou	rce data and/or periodically	as defined by the Write	16> NULL	
The Binery Output chieft	is some postable incomin	vite beth a value and a v	I ray Inactive	at using The bishast	

The Binary Output object is commandable, meaning the BACnet client must write both a value and a plot fuctive jat value. The highest level value will be the one written to the Modbus register. If all values are relinquished, the relinquish default value will be written to the Modbus register.

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 object name may be changed here. State text may be entered. When any of these are changed, be sure to save the updated configuration by clicking Save on the Config File page under System Setup.

The destination of data for a Binary Output object will be writing the remote Modbus register via the map indicated by the Device Link. The Modbus device will be updated upon change of source data and/or periodically as defined by the Write Map.

The Binary Output object is commandable, 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 Modbus register. If all values are relinquished, the relinquish default value will be written to the Modbus register.

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 Modbus register will not be written. 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 Modbus register.



# 5. Configuring BB2-7010 as a Modbus RTU Master

The BB2-7010 can be a Modbus RTU master or slave. As a master you can read Modbus data from, or write Modbus data to, other Modbus slaves. The BB2-7010 will periodically poll the other Modbus devices according to register maps you set up. To read from a remote Modbus device, configure a Read Map. To write to a remote Modbus device, configure Write Map.

Data read from a remote device is stored in a local register when received. Data written to a remote device is taken from a local register when sent. The local registers are the same collection of registers that are accessible to other masters when operating as slave, and accessible to other Modbus TCP devices as a collection of holding registers.

## 5.1 Modbus RTU Device Configuration

Modbus device configuration for RTU really consists of port configuration, and includes setting the slave address if the BB2-7010 is functioning as Modbus slave.

00011001101 110011 101101 Bachet 010010Netwo Model Be NAA000010	Buster 2 MODBUS RK GATEWAY 52-7010-01		Con	TROL-SOLUTIO	ONS, INC. Minnesota
Data Objects	Modbus	BACnet	Syste	em Setup	
Modbus RT	U Data Modbu	s RTU Setup Modbus	s TCP Data	Modbus TCP Setup	
Local Device	RTU Read Map	RTU Write Map		J	
This page displays con	figuration parameters	for the Modbus RTU serial p	oort.		
Baud Rat	e 19200 💌 🛛 P.	arity 🛛 None, 1 Stop Bit 🛛 👻			Update
I am the	Master 💿	I am	a Slave 🔘		
Paremete	ers for RTU Master:	Parar	meters for RTU	Slave:	
Default P	oll Rate 5.000 Se	econds My A	ddress or Unit #	+0	
Timeout 🗌 Use F	1.UUU Seconds FC 5/6 instead of 15/1	6 for unit numbers (slave a	)ouble registers ddresses) starti	; are swapped ing at 0	

Select baud rate and parity from the drop down list. Click either Master or Slave buttons to select type of operation. Enter timing parameters or address as applicable. Click update to register your changes.

IMPORTANT: Set timeout to something long enough for the device. If too short, the gateway will not wait long enough for a response from the Modbus slave device, and the result will be a lot of "no response" errors from the device even though the device is perfectly functional.

If your slave/server device only supports function codes 5 and 6 for writing, check the Use FC 5/6 box. The default function codes are 15 and 16, which are most widely used. If you check the box, you should also enter a "starting at" unit # or slave address. This allows supporting both types of devices at the same time provided you assign slave addresses in two non-overlapping groups. (These settings do not apply if the BB2-7010 is the slave.)

The double register swap on this page only applies when the local device (the gateway you are configuring here) is functioning as a Modbus RTU slave.

The term "swapped" only applies to double or float formats. Modbus registers are, by definition, 16 bits of data per register. Access to 32-bit data, either 32-bit integer ("double"), or IEEE 754 floating point ("float"), is supported by the use of two consecutive registers. Modbus protocol is inherently "big endian", therefore, Modbus by the Module defaults to having the high order register first for double and float. If the low order register comes first on the device being accessed, check the "swapped" box.

If you have "swapped" turned around, you will quickly recognize it. If floating point data is reversed, a 1.0 becomes 2.2779508e-41, which simply rounds to zero. The pattern is not as predictable as the 1.0 example would suggest. A floating point 1.1 becomes negative 107609184. If 32-bit integer data is reversed, 1 becomes 65536.

000 110 101 010 100	10011001101101 1001101 Babel Buster 2 01101 BacNet-MODBUS 10010 NETWORK GATEWAY MODEL BB2-7010-01 000001									
Dat	ta Objects	Modbus		BACne	1	System	Setup			
Modbus RTU Data Modbus R				TU Setup	Modbus T	CP Data 👖	Modbus TCP 9	Setup 🚶	]	
Lo	Local Device RTU Read Map RTU Write Map									
Read devic	l remote registers into lo ses for processing here. •	ical registers. Click on map i	This p numb	age creates a er to see more	map entry th detail and ir	at reads data sert/delete rul	from one or moi es.	re remote M	odbus RTU serial	
				Showing	1 to 5	of 5	(	Update	<pre> Next &gt;</pre>	
Map #	Remote Type	Remote Register Foi	: rmat	Remote Register #	Remote Unit #	Scale	Local Object #		Name	
1	Holding Register 💌	Float	*	1019	2	0.00000	Al 1	Volts		
2	Holding Register 💌	Float	*	1163	2	0.00000	AI 2	Amps		
3	Holding Register 💌	Float	*	1033	2	0.00000	AI 3	Hertz		
4	Holding Register 💌	Float	*	1011	2	0.00000	AI 4	Watts		
<u>5</u>	None 💌	Integer	*	0	0	0.00000	0	—		

## 5.2 Modbus RTU Master Read Maps

Rule number simply tells you where you're at on the list of register maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Showing" box, then click Update.

Rules entered on this page only read data from remote devices. Go to the RTU Write Map to write data to those devices. The full parameter set is different for read versus write.

An abbreviated version of a list of rules is shown on this page. Any of the parameters shown may be changed here and registered by clicking the Update button. To view and/or modify the complete set of parameters, click on the map number in the left most column.

For each remote register to be read, enter the register type, format, number, and remote unit (slave address).

When the remote register is read, the data will be multiplied by the scale factor and written to the local register number given. The name is optional and used only for display purposes.

Selecting "none" for remote type effectively deletes the rule even though it will still appear in the list until deleted. Unused rules at the end of the list will always show none as the type.

Local register numbers are 1-999 for integer values, and 1001-1999 accessed as register pairs for floating point. If you try to enter an even number above 1001, you will get an error message. All floating point register pairs start on odd boundaries. All local registers are accessed via Modbus as holding registers.

Click on the map number in the left column of the tabular read map page (above) to get the expanded view of one read map at a time (below).

000110011011 110011 101101 Babel Bacnet- 010010 Networ Model BB2	Buster 2 MODBUS K GATEWAY 2-7010-01		Cont	ROL-SOLUTIO	ONS, INC. Minnesota
Data Objects	Modbus	BACnet	System	n Setup	
Modbus RTU	Data Modbus RT	U Setup Modbu	s TCP Data	Modbus TCP Setup	
Local Device	RTU Read Map	RTU Write Map			
This page creates a map	p entry that reads data fr	om a remote Modbus R	TU serial device t	for processing here.	
Map # 1				Upda	te < Prev Next >
Read Holding Register	as Float vfro	om register # 1019 Ily scale: 0.00000 an	at Unit # 2 nd offset: 0.0000	with doubles swappe	d 🗸
Save in local object # Al	1 named Volts		Repeat this p	process every 5.0	seconds.
Apply this default value:	0.00000 after 0	read failure(s).			
COV increment: 0.00000	Units: no_units		*		
# RTU Read Maps Enable	≥d: 5			Inse	ert Delete

Rule number simply tells you where you're at on the list of register maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Map #" box, then click Update.

For each remote register to be read, enter the register type, format, number, and remote unit (slave address).

When the remote register is read, data may be manipulated before being written to the local register. If a bit mask is entered (in hexadecimal), and the remote register type is signed or unsigned (16-bit data), the mask will be bit-wise logical AND-ed with the data, and the retained bits will be right justified in the result. The result will then be multiplied by the scale factor. The offset is then added and this final result is written to the local register number given. The name is optional and used only for display purposes.

The periodic poll time determines how often the remote register will be read. This number, if nonzero, will override the default poll time given in the Devices page for the remote device being read.

The default value will be stored into the local register after the given number of read failures if the fail count is non-zero. Setting the count to zero will disable the default, and the object will retain the most recent value obtained.

Delete will remove the rule number shown in the "Map #" box. Insert will insert a new rule before the rule number shown, and is used for placing rules between existing rules. It is not necessary to use Insert to add rules to the bottom of the list or to define any rule presently having zero for a source object or "none" for remote type.

Selecting "none" for remote type effectively deletes the rule even though it will still appear in the list until deleted. Unused rules at the end of the list will always show none as the type. If you wish to prevent these from being displayed, reduce the number of rules enabled.

The number of rules enabled simply limits the scope of rule review so that you do not have to review a lot of unused rules. If the displayed rules are used up and you need more, increase the enabled number.



## 5.3 Modbus RTU Master Write Maps

Rule number simply tells you where you're at on the list of register maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Showing" box, then click Update.

Rules entered on this page only write data to remote devices. Go to the Client Read Map to read data from those devices. The full parameter set is different for read versus write.

An abbreviated version of a list of rules is shown on this page. Any of the parameters shown may be changed here and registered by clicking the Update button. To view and/or modify the complete set of parameters, click on the map number in the left most column.

Data from the local register given will be multiplied by the scale factor before being written. For each remote register to be written, enter the register type, format, number, and remote unit (slave address).

Selecting "none" for remote type effectively deletes the rule even though it will still appear in the list until deleted. Unused rules at the end of the list will always show none as the type.

Local register numbers are 1-999 for integer values, and 1001-1999 accessed as register pairs for floating point. If you try to enter an even number above 1001, you will get an error message. All floating point register pairs start on odd boundaries. All local registers are accessed via Modbus as holding registers.

Click on the map number in the left column of the tabular write map page (above) to get the expanded view of one write map at a time (below).

0001100110110 110011 101101 Babel 101101 Bacnet-M 010010 Network Model BB2- 10000010	Buster 2 AODBUS GATEWAY 7010-01		Control So	DEUTIONS, INC. MINNESOTA
Data Objects	Modbus	BACnet	System Setup	
Modbus RTU D	ata 🔰 Modbus RTU S	etup Modbus TCP	Data 👖 Modbus TC	P Setup
Local Device	RTU Read Map	RTU Write Map		
This page creates a map of Map # 1	entry that writes data to a r	emote Modbus RTU seria	l device from data contair	Update <prev next=""></prev>
Read local object # AV1 Apply default value of 0.00 Write remote register 🗖 a	named Analog Value	1 d/or when 0.0 anged by 0.00000 or	seconds have elapsed wi	th no host update. nds have elapsed with no change.
Otherwise write remote regi Apply scale: 0.00000 an	ister unconditionally. In any	v event, when writing rem	ote register, apply local of ask: 0000 and bit	bject data as follows: fill: 0000
write Holding Register	🖌 as Integer 🛛 👻 to reg	ister # 19 at Uni	t # 7 with doubles	swapped 🗌
Repeat this process 💿 at Relinquish Default: 0.0000	least O no more than eve	ry 5.0 seconds. 10000 Units: NO_ur	its	~
# Client Write Maps Enable	d: 2			Insert Delete

Rule number simply tells you where you're at on the list of register maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Map #" box, then click Update.

The local register data may be written periodically, or when it changes, or both. To send upon change (send on delta), check the first box and enter the amount by which the local register must change before being written to the remote device. To guarantee that the remote register will be written at least

occasionally even if the data does not change, check the second box and enter some amount of time. This time period will be referred to as the "maximum quiet time".

Data from the local register may be manipulated before being written to the remote register. The local data is first multiplied by the scale factor. The offset is then added to it. If a bit mask is entered, and the remote register type is signed or unsigned (16-bit data), the mask will be bit-wise logical AND-ed with the data. The mask is right justified, then AND-ed with the data. The result is then left shifted back to the original position of the mask. In other words, the least significant bits of the original data will be stuffed at the position marked by the mask.

After the scaling and masking, the bit fill will be logically OR-ed into the result, but only if the mask was nonzero and was used. Both mask and fill are entered in hexadecimal.

Multiple local registers may be packed into a single remote register. To accomplish this, define two or more rules in sequence with the same remote destination. If the destination is the same, data types are 16-bit (integer or unsigned), bit masks are nonzero, and the rules are sequential, the results of all qualifying rules will be OR-ed together before being sent to the remote destination.

For the remote register to be written, enter the register type, format, number, and remote unit (slave address).

The repeat time may determine how often the remote register will be written. If send on delta and maximum quiet time are not checked above, clicking the "at least" button will establish a periodic update time. If send on delta is used and you wish to limit the network traffic in the event changes are frequent, click the "no more than" button and enter the minumum time that should elapse before another write to the remote device.

Delete will remove the rule number shown in the "Map #" box. Insert will insert a new rule before the rule number shown, and is used for placing rules between existing rules. It is not necessary to use Insert to add rules to the bottom of the list or to define any rule presently having zero for a source register or "none" for remote type.

Selecting "none" for remote type effectively deletes the rule even though it will still appear in the list until deleted. Unused rules at the end of the list will always show none as the type. If you wish to prevent these from being displayed, reduce the number of rules enabled.

The number of rules enabled simply limits the scope of rule review so that you do not have to review a lot of unused rules. If the displayed rules are used up and you need more, increase the enabled number.

## 5.4 Modbus RTU Master Data Displayed Per Slave

1100 1011 0100 1000	1001 Babel Buster 2 DILLO BACNET-MODBUS 1001 NETWORK GATEWAY MODEL BB2-7010-01 MINNESOTA									
Data Objects Modbus BA Cnet System Setup										
	Modbus RTU	I Data	Modbus RTU Setup	Modbus	s TCP Data	M	odbus TC	P Setup		Ì
RTUR	RTU Registers Error Codes									
This pa	ge displays data	to and from	registers in devices acce	ssed via th	e Modbus RT	U serial (	port.			
RTU Unit	t#2		s	howing 1	to 4 of	4		Update	< Pre	ev Next >
Dir.	Reg. Type	Remote Reg. #	Register Name	:	Local Object #	Hex	Update	Register Dat	а	Time since Last update
From	Holding Reg	01019	Volts		AI 1			121.484		3,180
From	Holding Reg	01163	Amps		AI 2			2.21056		3,100
From	Holding Reg	01033	Hertz		AI 3			60.0112		3.070
From	Holding Reg	01011	Watts		AI 4			265.243		3.090
RTU Unit	t#2 +	Unit - Un	it							

The values of Modbus registers that have been read from remote RTU serial devices is displayed here. One remote unit at a time is displayed. To display a different unit, change the RTU Unit #.

Simply click the Update button to view the most recent data. Enter a new value and check the Update box if the value should be changed when you click the Update button. Check the Hex box if you wish to view or enter values in hexadecimal (not recommended for floating point).

Click Update to view the most recent data values. Click "Prev" or "Next" to scroll through the list of registers. You may also enter a number in the "Showing" box to jump directly to a given register when Update is clicked.

## 5.5 Modbus RTU Errors

000 110 101 010 200	1001 Babel Buster 2 Bacnet-modeus 1001 Network Gateway Model BB2-7010-01 MINNESOTA										
Data	i Objects		Modbus	l l	BACnet	<u> </u>	System	Setup			
Modbus RTU Data Modbu			us RTU S	ietup 🚺 Ma	odbus TCP Dat	ia 🚺	Modbus TC	P Setup 🚶		l	
RTU Registers Error Codes			$\neg$					$\mathbf{r}$		٦	
This p	This page displays error codes encountered in processing reads and writes via the Modbus RTU serial port.										
				sł	nowing devices	from 1	]		Update	<pre> &lt; Prev</pre>	Next >
Unit #	Reset >	Read Error	Offending Read Map #	Reset >	Write Error	Offending Write Map #	Reset >	Total Messages	No Responses	CRC Errors	Exceptions
1		0/0	0		0/0	0		0	0	0	0
2		0/0	0		0/0	0		3744	0	0	0
3		0/0	0		0/0	0		0	0	0	0
4		0/0	0		0/0	0		0	0	0	0
5		0/0	0		0/0	0		0	0	0	0
6		0/0	0		0/0	0		0	0	0	0
7		0/0	0		5/0	1		32	32	0	0
8		0/0	0		0/0	0		0	0	0	0

The first occurrence of read and write errors are shown along with the map number that was being processed when the error occurred. Check the reset box and click update to clear it and possibly show the next error if there are more than one active error conditions.

A total count of all errors is also shown. This total is the sum of errors for all maps for this device. Check the reset box and click update to reset the counts. Click Update to view the most recent data values.

Error code indications of A/B indicate the following errors with the first number:

- 1 = Transaction ID out of sync
- 2 = Exception code returned by remote device
- 3 = Function code mismatch (bad packet)
- 4 = Inusfficient data (bad packet)
- 5 = No response from remote device, timed out
- 6 = CRC error in received packet

When A is code 2 indicating an exception code was returned, B indicates the exception as follows:

- 1 = Illegal function code
- 2 = Illegal data address (the requested register does not exist in the device)
- 3 = Illegal data value



# 6. Configuring BB2-7010 as a Modbus TCP Client

The BB2-7010 can be a Modbus client or server. As a client (master) you can read Modbus data from, or write Modbus data to, other Modbus servers (slaves). The BB2-7010 will periodically poll the other Modbus devices according to register maps you set up. The Modbus server (slave) devices that you will read/write are defined on the Devices page. To read from a remote Modbus device, configure a Read Map. To write to a remote Modbus device, configure Write Map.

Data read from a remote device is stored in a local data object when received. Data written to a remote device is taken from a local data object when sent. The local data objects are the same collection of objects that are accessible to other clients via the server map, and accessible to other BACnet devices via MS/TP or BACnet IP.

100110011011011011011011011011011011011									
Data Objects	Modbus	BACnet	Sys	tem Setup	1				
Modbus RT	'U Data 👖 Modb	us RTU Setup 🚶	Modbus TCP Data	Modbus TC	P Setup				
Devices	Client Read Ma	ap Client W	rite Map Ser	ver Map					
This page sets up the input and/or output (v	network address and ia the client read and	optional device parar client write maps). Ti	meters for a remote N he local device acts as	1odbus/TCP device s a Modbus maste	e that will be lir r to the remote	nked to for remote e devices listed below.			
Device #	1				Update	<pre> Next &gt;</pre>			
IP Address	192.168.1.144	Port 502 Local	Name: Rack Monito	or					
Unit (optional)	0	Use FC 5/6 instead	i of 15/16						
	📃 Swap Double Re	gisters		Connection Status					
Default Poll Period	2.0 Seconds			0					

## 6.1 Modbus TCP Device Configuration

The Modbus Devices page is illustrated above. Device number simply shows you where you are on the device list. Click "next" and "prev" to scroll through the list.

Remote Modbus/TCP devices to be accessed by this device are specified here. Enter the IP address of the remote device, a name to reference in other pages, a unit number, poll rate, and check "swapped" if appropriate. Then click "update".

If your slave/server device only supports function codes 5 and 6 for writing, check the Use FC 5/6 box. The default function codes are 15 and 16, which are most widely used.

The term "swapped" only applies to double or float formats. Modbus registers are, by definition, 16 bits of data per register. Access to 32-bit data, either 32-bit integer ("double"), or IEEE 754 floating point ("float"), is supported by the use of two consecutive registers. Modbus protocol is inherently "big endian", therefore, Modbus by the Module defaults to having the high order register first for double and float. If the low order register comes first on the device being accessed, check the "swapped" box.

If you have "swapped" turned around, you will quickly recognize it. If floating point data is reversed, a 1.0 becomes 2.2779508e-41, which simply rounds to zero. The pattern is not as predictable as the 1.0 example would suggest. A floating point 1.1 becomes negative 107609184. If 32-bit integer data is reversed, 1 becomes 65536.

Connection status will show a non-zero error code if there is a socket error. Possible errors include:

5 = Connection failed, unable to bind (usually means remote device not connected or not reachable)

- 81 = Connection in progress (means unsuccessful connect attempt, still trying)
- 95 = Network is unreachable
- 97 = Connection aborted
- 98 = Connection reset by peer
- 103 = Connection timed out
- 104 = Connection refused
- 107 = Host is unreachable

## 6.2 Modbus TCP Client Read Maps

1001 101 101 1001 Babel Buster 2 DILLO BACNET-MODBUS 1001 NETWORK GATEWAY MODEL BB2-7010-01 MINNESOTA										
Data Objects Modbus BACnet System Setup										
Modbus RTU	Modbus RTU Data Modbus RTU Setup Modbus TCP Data Modbus TCP Setup									
Devices	Client Read Map	Client V	Mrite Map Serve	ег Мар						
Read remote registers in processing here. Click o	nto local objects. This n map number to see	page creates a m more detail and i	ap entry that reads data insert/delete rules.	from one or n	nore remote Mo	dbus/TCP servers for				
		Showing	1 to 2 of 2		Update	<pre> Next&gt;</pre>				
Map Remote # Type	Remote Register Forma	Remote at Register#	Remote Device	Scale	Local Object #	Name				
1 Holding Register	🖌 Float 🗸	1001	Rack Monitor 💌	0.00000	AI 7	Rack Temp				
2 None	🖌 Integer 🖌	0	None 💌	0.00000	0	—				

Rule number simply tells you where you're at on the list of register maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Showing" box, then click Update.

Rules entered on this page only read data from remote devices. Go to the Client Write Map to write data to those devices. The full parameter set is different for read versus write.

An abbreviated version of a list of rules is shown on this page. Any of the parameters shown may be changed here and registered by clicking the Update button. To view and/or modify the complete set of parameters, click on the map number in the left most column.

For each remote register to be read, enter the register type, format, number, and location (device). The names in the device list are defined in the Devices page.

When the remote register is read, the data will be multiplied by the scale factor and written to the local object number given. The name is optional and used only for display purposes.

Selecting "none" for remote type effectively deletes the rule even though it will still appear in the list until deleted. Unused rules at the end of the list will always show none as the type.

Local Object is internally a coded number consisting of BACnet object type multiplied by 1000, then added to the object number starting from #1. These are translated into abbreviations that are easy to interpret on the web page as follows:

AI n = Analog Input #n AO n = Analog Output #n AV n = Analog Value #n BI n = Binary Input #n BO n = Binary Output #n BV n = Binary Value #n MI n = Multi-state Input #n MO n = Multi-state Output #n MV n = Multi-state Value #n

Object numbers start at #1. The maximum available number varies by object type, and these limits may be found on the System Capacities link from the home/index page (click graphic at top).

001100110110 10011Babel 01101Babel Bacnet-M 10010Network Model BB2-	Buster 2 AODBUS GATEWAY 7010-01		Control-S	SOLUTIONS, INC. MINNESOTA				
Data Objects	Modbus	BACnet	System Setup					
Modbus RTU D	ata 👔 Modbus RTU :	Setup 👖 Modbus 1	CP Data Modbus	TCP Setup				
Devices	Client Read Map	Client Write Map	Server Map					
This page creates a map entry that reads data from a remote Modbus/TCP server for processing here.								
Map # 1				Update < Prev Next >				
Read Holding Register	✓ as Float ✓ from	register # 1001	at Rack Monitor 👻					
Apply bit mask if applicable	e: 0000 then apply	scale: 0.00000 and	offset: 0.00000					
Save in local object # Al 7	named Rack Tem	p	Repeat this process eve	ry 2.0 seconds.				
Apply this default value:	0.00000 after 0 r	ead failure(s).						
Initial COV increment: 0.00	0000 Units: no_unit	S	*					
# Client Read Maps Enable	:d: 2			Insert Delete				

Rule number simply tells you where you're at on the list of register maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Map #" box, then click Update.

For each remote register to be read, enter the register type, format, number, and location (device). The names in the device list are defined in the Devices page.

When the remote register is read, data may be manipulated before being written to the local object. If a bit mask is entered (in hexadecimal), and the remote register type is signed or unsigned (16-bit data), the mask will be bit-wise logical AND-ed with the data, and the retained bits will be right justified in the result. The result will then be multiplied by the scale factor. The offset is then added and this final result is written to the local object number given. The name is optional and used only for display purposes.

The periodic poll time determines how often the remote register will be read. This number, if nonzero, will override the default poll time given in the Devices page for the remote device being read.

The default value will be stored into the local object after the given number of read failures if the fail count is non-zero. Setting the count to zero will disable the default, and the object will retain the most recent value obtained.

Delete will remove the rule number shown in the "Map #" box. Insert will insert a new rule before the rule number shown, and is used for placing rules between existing rules. It is not necessary to use Insert to add rules to the bottom of the list or to define any rule presently having zero for a source object or "none" for remote type.

Selecting "none" for remote type effectively deletes the rule even though it will still appear in the list until deleted. Unused rules at the end of the list will always show none as the type. If you wish to prevent these from being displayed, reduce the number of rules enabled.

Initial COV increment and period will only apply if a BACnet client subscribes to COV notification from the BACnet object assigned to this Modbus map. These properties may be overwritten by the BACnet client(s) at any time. The values shown here are initial values, not necessarily the current values. (Note: COV increment only applies to Analog objects, all changes are reported for Binary or Multistate objects.)

Units default to no\_units, but you may select any of the available BACnetEngineeringUnits values. This value will simply be read by the BACnet client when the units property is requested from the object this Modbus register maps to. The units have no bearing on calculations performed. You must select appropriate scale and offset values to make any required translation between Modbus units and BACnet units. Units are only valid for Analog objects.

The number of rules enabled simply limits the scope of rule review so that you do not have to review a lot of unused rules. If the displayed rules are used up and you need more, increase the enabled number.

## 6.3 Modbus TCP Client Write Maps

	1001 Babel Buster 2 0110 BacNet-Modbus 1001 Network Gateway Model BB2-7010-01 0000										
Data Objects Modbus			ibus B	ACnet	System Setup						
	Modbus RTU Data Modbus RTU Setup Modbus TCP Data Modbus TCP Setup										
De	vices	Clie	nt Read Map Cl	lient Write Map	Server M	ap )					
Write	e local registe 1 data contain	rs out to remo ed here. Click	te registers. This page cro on map number to see m	eates a map entry th tore detail and inser	nat writes data t/delete rules.	to one or more remote M	odbus/TCP servers				
			Sho	wing 1 to 2 of	f 2	Update	<pre> Next &gt;</pre>				
Map #	Local Object #	Scale	Remote Type	Remote Register Format	Remote Register #	Remote Device	Name				
1	BO 1	0.00000	Holding Register 💌	Integer 💌	15	Rack Monitor 👻	Relay 1 Out				
2	0	0.00000	None 💌	Integer 💌	0	None 💌	_				

Rule number simply tells you where you're at on the list of register maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Showing" box, then click Update.

Rules entered on this page only write data to remote devices. Go to the Client Read Map to read data from those devices. The full parameter set is different for read versus write.

An abbreviated version of a list of rules is shown on this page. Any of the parameters shown may be changed here and registered by clicking the Update button. To view and/or modify the complete set of parameters, click on the map number in the left most column.

Data from the local object given will be multiplied by the scale factor before being written. For each remote register to be written, enter the register type, format, number, and location (device). The names in the device list are defined in the Devices page. The name is optional and used only for display purposes.

Selecting "none" for remote type effectively deletes the rule even though it will still appear in the list until deleted. Unused rules at the end of the list will always show none as the type.

Local Object is internally a coded number consisting of BACnet object type multiplied by 1000, then added to the object number starting from #1. These are translated into abbreviations that are easy to interpret on the web page as follows:

Al n = Analog Input #n AO n = Analog Output #n AV n = Analog Value #n BI n = Binary Input #n BO n = Binary Output #n BV n = Binary Value #n MI n = Multi-state Input #n MO n = Multi-state Output #n MV n = Multi-state Value #n

Object numbers start at #1. The maximum available number varies by object type, and these limits may be found on the System Capacities link from the home/index page (click graphic at top).

100110011011001 10011 Bachel Buster 2 10110 Bachel Buster 2 10010 Network Gateway MODEL BB2-7010-01 10000									
Data Objects	Modbus	BACnet	System Se	tup					
Modbus RTU	Data 📕 Modbus R1	'U Setup 📗 Modbus	TCP Data M	odbus TCP Setup					
Devices	Client Read Map	Client Write Map	Server Map						
This page creates a map	o entry that writes data to	) one or more remote Mo	dbus/TCP servers fr	om data contained here.					
Map # 1				Update	<prev next=""></prev>				
Read local object # BO 1	named Relay10	ut							
Apply default value of $0$ .	00000 🗖 at power-up	and/or 🔲 when 0.0	seconds have (	elapsed with no host upd	late.				
Write remote register 📃	any time local object has	s changed by 0.00000	or 🗖 when 0.0	seconds have elaps	sed with no change.				
Otherwise write remote re	gister unconditionally. In	any event, when writing	remote register, ap	ply local object data as fo	ollows:				
Apply scale: 0.00000	and offset: 0.00000 T	hen if applicable, apply l	bit mask: 0000	and bit fill: 0000					
write Holding Register	💌 as Integer 💌 to	register #15 a	t 🛛 Rack Monitor 💌						
Repeat this process 💿 a	at least 🔘 no more than	every 2.0 secon	nds.						
Relinquish Default: 0.000	000 COV increment	: 0.00000 Units: 1	no_units	2	<b>~</b>				
# Client Write Maps Enab	led: 2			Insert	Delete				

Rule number simply tells you where you're at on the list of register maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Map #" box, then click Update.

The local object data may be written periodically, or when it changes, or both. To send upon change (send on delta), check the first box and enter the amount by which the local object must change before being written to the remote device. To guarantee that the remote register will be written at least occasionally even if the data does not change, check the second box and enter some amount of time. This time period will be referred to as the "maximum quiet time".

Data from the local object may be manipulated before being written to the remote register. The local data is first multiplied by the scale factor. The offset is then added to it. If a bit mask is entered, and the remote register type is signed or unsigned (16-bit data), the mask will be bit-wise logical AND-ed with the data. The mask is right justified, then AND-ed with the data. The result is then left shifted back to the original position of the mask. In other words, the least significant bits of the original data will be stuffed at the position marked by the mask.

After the scaling and masking, the bit fill will be logically OR-ed into the result, but only if the mask was nonzero and was used. Both mask and fill are entered in hexadecimal.

Multiple local objects may be packed into a single remote register. To accomplish this, define two or more rules in sequence with the same remote destination. If the destination is the same, data types are 16-bit (integer or unsigned), bit masks are nonzero, and the rules are sequential, the results of all qualifying rules will be OR-ed together before being sent to the remote destination.

For the remote register to be written, enter the register type, format, number, and location (device). The names in the device list are defined in the Devices page.

The repeat time may determine how often the remote register will be written. If send on delta and maximum quiet time are not checked above, clicking the "at least" button will establish a periodic update time. If send on delta is used and you wish to limit the network traffic in the event changes are frequent, click the "no more than" button and enter the minumum time that should elapse before another write to the remote device.

Delete will remove the rule number shown in the "Map #" box. Insert will insert a new rule before the rule number shown, and is used for placing rules between existing rules. It is not necessary to use Insert to add rules to the bottom of the list or to define any rule presently having zero for a source register or "none" for remote type.

Selecting "none" for remote type effectively deletes the rule even though it will still appear in the list until deleted. Unused rules at the end of the list will always show none as the type. If you wish to prevent these from being displayed, reduce the number of rules enabled.

Initial COV increment and period will only apply if a BACnet client subscribes to COV notification from the BACnet object assigned to this Modbus map. These properties may be overwritten by the BACnet client(s) at any time. The values shown here are initial values, not necessarily the current values. (Note: COV increment only applies to Analog objects, all changes are reported for Binary or Multistate objects.)

Units default to no\_units, but you may select any of the available BACnetEngineeringUnits values. This value will simply be read by the BACnet client when the units property is requested from the object this Modbus register maps to. The units have no bearing on calculations performed. You must select appropriate scale and offset values to make any required translation between Modbus units and BACnet units. Units are only valid for Analog objects.

Initial Relinquish Default may be set here, but may be overwritten by the BACnet client at any time. This window reflects the initial value, not the current value. (Note: Relinquish Default only applies to commandable Output objects, and does not apply to Input or Value objects.)

The number of rules enabled simply limits the scope of rule review so that you do not have to review a lot of unused rules. If the displayed rules are used up and you need more, increase the enabled number.

## 6.4 Modbus TCP Errors

100 110 101 010	1001 Babel Buster 2 0110 Bachet-Modeus 1001 Network Gateway Model BB2-7010-01 00000										
Data Objects Modbus				BACnet System Setup							
	Mod	lbus RTU Data	Modb	us RTU S	etup Ma	dbus TCP Dat	a	Modbus TC	P Setup		1
Modbus Server Error Codes									$\overline{\mathbf{v}}$		٦
This p	This page displays error codes encountered in processing Modbus Client reads and writes via the Modbus TCP connection(s).										
											Update
Device	Reset >	Read Error	Offending Read Map #	Reset >	Write Error	Offending Write Map #	Reset >	Total Messages	No Responses	Exceptions	
1		0/0	0		0/0	0		56952	0	0	
2		0/0	0		0/0	0		0	0	0	
3		0/0	0		0/0	0		0	0	0	
4		0/0	0		0/0	0		0	0	0	
5		0/0	0		0/0	0		0	0	0	
6		0/0	0		0/0	0		0	0	0	
7		0/0	0		0/0	0		0	0	0	
8		0/0	0		0/0	0		0	0	0	

The first occurrence of read and write errors are shown along with the map number that was being processed when the error occurred. Check the reset box and click update to clear it and possibly show the next error if there are more than one active error conditions.

A total count of all errors is also shown. This total is the sum of errors for all maps for this device. Check the reset box and click update to reset the counts. Click Update to view the most recent data values.

Error code indications of A/B indicate the following errors with the first number:

- 1 = Transaction ID out of sync
- 2 = Exception code returned by remote device
- 3 = Function code mismatch (bad packet)
- 4 = Inusfficient data (bad packet)
- 5 = No response from remote device, timed out
- 6 = CRC error in received packet

When A is code 2 indicating an exception code was returned, B indicates the exception as follows:

- 1 = Illegal function code
- 2 = Illegal data address (the requested register does not exist in the device)
- 3 = Illegal data value



# 7. Using the BB2-7010 as a BACnet Client

The BACnet client is used to query other BACnet devices, obtain their Present Value data, and store a copy of that data in the BB2-7030's own local objects. From there, the data may be accessed by Modbus TCP or SNMP devices, or other BACnet devices when application specific reasons make this approach more preferred than direct routing.

## 7.1. BACnet Device Configuration

Setting up the BACnet client consists of identifying one or more BACnet devices, then listing the objects that should be queried (whether read or written). The client configuration pages are illustrated below.

00011001101 10011 101101 Bacnet 010010 Netwo Model Be No00010	Butter -modbus rk gateway 52-7010-01	2		Cont	ROL-SOI		s, Inc. inesota	
Data Objects	Modbus		BACnet	System	) Setup			
BACnet IP C	lient Di	iagno stics	l	Ì				)
Devices	Client Rea	d Map	Client Write Map					
This page sets up the client write maps). The	device list for rer local device acts	note BACnet : as a BACnet	IP devices that will b : client to the remote	e accessed for re servers listed be	mote input and low.	/or output (via	a the client read	and
Device #	1					Update	<pre> N</pre>	ext >
Device Instance	8066	L	.ocal Name: BB2-7	010-02				
Default Poll Period	10.0 Seco	nds [	Default Write Priority	: 10				
Reply Timeout:	5.0 s	Seconds Ti	meouts: 0		Clear			
Address Binding:	💿 Dynamic (Wh	io-Is) 🔘 Sta	tic					
Device IP Address	192.168.1.178:4	7808	]		Clear Cao	che		

Device number simply shows you where you are on the device list. Click "next" and "prev" to scroll through the list.

Remote BACnet devices to be accessed by this device are specified here. Enter the Device Instance of the remote device, a name to reference in other pages, a poll rate, default reply timeout, and default write priority. Enter static address if applicable. Then click "update".

The gateway broadcasts a "who-is" looking for this device when a read or write map wants to use this device. When (if) it responds, its IP address or MS/TP mac address is listed here simply as a diagnostic. Timeouts resulting from inability to reach this device are tabulated on this page as well, and may be cleared by clicking the Clear button. To cause the who-is process to be repeated, click Clear Cache.

BACnet IP or MS/TP slave devices that to not support Who-Is/I-Am can still be supported here. When this is the case, enter the slave device's Mac address in the Static Mac window and check the 'No Who-Is' box. If located on a remote network via a router, enter the network number as DNet. This static entry effectively replies to the implied Who-Is.

To use a fixed static address, enter a single number for MS/TP MAC address. or an IP address optionally including port number. An example of IP address with port number would be 192.168.1.99:47808. The 47808 is the port number, and is separated from the IP address by a colon. Note that 47808 is the default 0xBAC0 port number. If no port number is given, the port configured on the BACnet IP Port page will be used (the BB2-7010's own port).

## 7.2. BACnet Client Read Maps

The client read maps tell the BB2-7010 which objects to read, from which BACnet devices. Click on the map number to view the full details of the read map.

100 110 101 010	1001 101101 1001 Babel Buster 2 Bacnet-Modbus 1001 Network Gateway Model BB2-7010-01 MINNESOTA										
Dat	ta Objects	Modbus	BACnet	System S	Setup						
	BACnet IP Client Diagnostics										
Dev	vices	Client Read	Map Client Write Map	ſ		J					
Read proce	l remote object data essing here. Click or	a into local object n map number to	s. This page creates a map entry see more detail and insert/delet	that reads data f æ maps.	from one or i	more remote BA	Cnet IP servers f	for			
			Showing 1 to :	3 of 3		Update	<pre> Ne&gt;</pre>	d >			
Map #	Remote Type	Remote Object #	Remote Device	Scale	Local Object #		Name				
1	Analog Input	♥ 1	BB2-7010-02 💌	0.00000	AI 5	Temper	rature	]			
2	Analog Input	✓ 2	BB2-7010-02 💌	0.00000	Al 6	Humidit	у				
3	None	✓ 0	None 💌	0.00000	0	—					

Map number simply tells you where you're at on the list of register maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Showing" box, then click Update.

Maps entered on this page only read data from remote devices. Go to the Client Write Map to write data to those devices. The full parameter set is different for read versus write.

An abbreviated version of a list of maps is shown on this page. Any of the parameters shown may be changed here and registered by clicking the Update button. To view and/or modify the complete set of parameters, click on the map number in the left most column.

For each remote object to be read, enter the object instance and type, and location (device). The names in

the device list are defined in the Devices page.

When the remote object is read, data may be manipulated before being written to the local object. The value will be multiplied by the scale factor. The final result is written to the local object number given. The name is optional and used only for display purposes.

Selecting "none" for remote type effectively deletes the map even though it will still appear in the list until deleted. Unused maps at the end of the list will always show none as the type.

Local Object is internally a coded number consisting of BACnet object type multiplied by 1000, then added to the object number starting from #1. These are translated into abbreviations that are easy to interpret on the web page as follows:

AI n = Analog Input #n AO n = Analog Output #n AV n = Analog Value #n BI n = Binary Input #n BO n = Binary Output #n BV n = Binary Value #n MI n = Multi-state Input #n MO n = Multi-state Output #n MV n = Multi-state Value #n

Object numbers start at #1. The maximum available number varies by object type, and these limits may be found on the System Capacities link from the home/index page (click graphic at top).

000110011011 110011 101101 Backet 010010 Networ Model BB2 10000010	Buster 2 MODBUS KGATEWAY 2-7010-01		Control-S	OLUTIONS, INC. MINNESOTA
Data Objects	Modbus	BACnet	System Setup	
BACnet IP Clie	ent Diagnostics			
Devices	Client Read Map	Client Write Map	, )	
This page creates a map	entry that reads data from	a remote BACnet IF	server for processing here.	
Мар # 1				Update < Prev Next >
Read property 85	instance # 1 of o	object type Analog	Input 💟	
Read from device BB2-70	010-02 🝸 using index 0			
Then apply scale: 0.0000	0 and offset: 0.00000			
Save in local object Al 5	named Temperat	ture	Repeat this process even	ry 10.0 seconds.
Apply this default value:	0.00000 after 0 1	read failure(s).		
# Client Read Maps Enable	ed: 3			Insert Delete

Rule number simply tells you where you're at on the list of object maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Map #" box, then click Update.

For each remote object to be read, enter the object instance and type, and location (device). The names in the device list are defined in the Devices page. Use index value of 0 if no index.

When the remote object is read, data may be manipulated before being written to the local object. The value will be multiplied by the scale factor, then the offset is added. The final result is written to the local object number given. The name is optional and used only for display purposes.

The periodic poll time determines how often the remote object will be read. This number, if nonzero, will override the default poll time given in the Devices page for the remote device being read.

The default value will be stored into the local object after the given number of read failures if the fail count is non-zero. Setting the count to zero will disable the default, and the object will retain the most recent value obtained.

Delete will remove the rule number shown in the "Map #" box. Insert will insert a new map before the map number shown, and is used for placing maps between existing maps. It is not necessary to use Insert to add maps to the bottom of the list or to define any map presently having zero for a source object or "none" for remote type.

Selecting "none" for remote type effectively deletes the map even though it will still appear in the list until deleted. Unused maps at the end of the list will always show none as the type. If you wish to prevent these from being displayed, reduce the number of maps enabled.

The number of maps enabled simply limits the scope of map review so that you do not have to review a lot of unused maps. If the displayed maps are used up and you need more, increase the enabled number.

## 7.3. BACnet Client Write Maps

The client read maps tell the BB2-7010 which objects to write, on which BACnet devices. Click on the map number to view the full details of the write map.

000 110 101 010 100	10011001101101101101101101101101100110										
Dat	Data Objects Modbus BACnet System Setup										
	BACnet IP Client Diagnostics										
De	vices	Clier	nt Read Map	Cli	ient Write Map	» <b>\</b>					
Write data	e local object ( contained he	data out to rem re. Click on ma	ote objects. This p p number to see m	age cr nore d	eates a map ( etail and inser	entry that writes data to one o t/delete rules.	r more remote BACnet IP servers from				
				Shov	ving 1 t	:0 2 of 2	Update <prev next=""></prev>				
Map #	Local Object #	Scale	Remote Type		Remote Object #	Remote Device	Name				
1	BO 2	0.00000	Binary Output	*	3	BB2-7010-02 💌	Relay 2				
2	0	0.00000	None	*	0	None 💌	_				

Map number simply tells you where you're at on the list of register maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Showing" box, then click Update.
Maps entered on this page only write data to remote devices. Go to the Client Read Map to read data from those devices. The full parameter set is different for read versus write.

An abbreviated version of a list of maps is shown on this page. Any of the parameters shown may be changed here and registered by clicking the Update button. To view and/or modify the complete set of parameters, click on the map number in the left most column.

Data from the local object given will be multiplied by the scale factor before being written. For each remote object to be written, enter the object instance and type, and location (device). The names in the device list are defined in the Devices page. The name is optional and used only for display purposes.

Selecting "none" for remote type effectively deletes the map even though it will still appear in the list until deleted. Unused maps at the end of the list will always show none as the type.

Local Object is internally a coded number consisting of BACnet object type multiplied by 1000, then added to the object number starting from #1. These are translated into abbreviations that are easy to interpret on the web page as follows:

AI n = Analog Input #n AO n = Analog Output #n AV n = Analog Value #n BI n = Binary Input #n BO n = Binary Output #n BV n = Binary Value #n MI n = Multi-state Input #n MO n = Multi-state Output #n MV n = Multi-state Value #n

Object numbers start at #1. The maximum available number varies by object type, and these limits may be found on the System Capacities link from the home/index page (click graphic at top).

1001 Bachel Buster 2 Bachet-Modeus 1001 Network Gateway Model BB2-7010-01 MINNESOTA
Data Objects Modbus BA Cnet System Setup
BACnet IP Client Diagnostics
Devices Client Read Map Client Write Map
This page creates a map entry that writes data to one or more remote BACnet IP servers from data contained here.
Map # 1 Update Verv Next>
Read local object BO 2 named Relay 2
Apply default value of 0.00000 at power-up and/or when 0.0 seconds have elapsed with no host update.
Write remote register 🗋 any time local object has changed by 0.00000 or 🗌 when 0.0 seconds have elapsed with no change.
Otherwise write remote register unconditionally. In any event, when writing remote register, apply local object data as follows:
Apply scale: 0.00000 and offset: 0.00000 Then, using index 0 and priority 0 proceed to
Write property 85 encoded as data type Enumerated 😒
Write to instance # 3 of object type Binary Output 💉 at device BB2-7010-02 💌
Repeat this process $\odot$ at least $\bigcirc$ no more than every $0.0$ seconds.
# Client Write Maps Enabled: 2 Insert Delete

Rule number simply tells you where you're at on the list of object maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Map #" box, then click Update.

The local object data may be written periodically, or when it changes, or both. To send upon change (send on delta), check the first box and enter the amount by which the local object must change before being written to the remote device. To guarantee that the remote object will be written at least occasionally even if the data does not change, check the second box and enter some amount of time. This time period will be referred to as the "maximum quiet time".

Data from the local object may be manipulated before being written to the remote register. The local data is first multiplied by the scale factor. The offset is then added to it.

For the remote object to be written, enter the object instance and type, index if applicable (leave at 0 if not), and priority to use of the object being written is commandable. The names in the device list are defined in the Devices page.

The repeat time may determine how often the remote object will be written. If send on delta and maximum quiet time are not checked above, clicking the "at least" button will establish a periodic update time. If send on delta is used and you wish to limit the network traffic in the event changes are frequent, click the "no more than" button and enter the minumum time that should elapse before another write to the remote device.

Delete will remove the rule number shown in the "Map #" box. Insert will insert a new map before the

map number shown, and is used for placing maps between existing maps. It is not necessary to use Insert to add maps to the bottom of the list or to define any map presently having zero for a source object or "none" for remote type.

Selecting "none" for remote type effectively deletes the map even though it will still appear in the list until deleted. Unused maps at the end of the list will always show none as the type. If you wish to prevent these from being displayed, reduce the number of maps enabled.

The number of maps enabled simply limits the scope of map review so that you do not have to review a lot of unused maps. If the displayed maps are used up and you need more, increase the enabled number.

## 7.4. BACnet Errors

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.

100110011011011011011011011011011011011						
Data Objec	ts	Modbus	BACnet	System Setup		
B	ACnet IP Client	Diagn	ostics	Ţ		Ì
Errors: Rea	ad Maps	Errors: Write M	taps V			
This page di	splays errors, if	any, reported b	by the remote server upon attem	pts to read from that device.		
					<-> Top	p Next >
Map #	Remote Type	Remote Object #	Remote Device	Name	Error Class	Error Code
		0			0	0
					P	leset Errors

Errors for BACnet IP client read maps are shown on this page. Only those maps with errors to report are listed. Refer to the code and class lists below for interpretation.

**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.

- 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



# 8. Configuring BB2-7010 as a Modbus RTU Slave

The BB2-7010 can be a Modbus RTU master or slave. As slave, the BB2-7010 will respond to another Modbus master and return data requested. The various objects in the BB2-7010 are accessed as holding registers, with register numbers calculated and based on object type and instance.

#### 8.1 Modbus RTU Device Configuration

Modbus device configuration for RTU really consists of port configuration, and includes setting the slave address if the BB2-7010 is functioning as Modbus slave.



Select baud rate and parity from the drop down list. Click either Master or Slave buttons to select type of operation. Enter timing parameters or address as applicable. Click update to register your changes.

The double register swap on this page only applies when the local device (the gateway you are configuring here) is functioning as a Modbus RTU slave. If the Modbus master expects least significant data to be in the first (lowest numbered) register, then check the "swap" box.

The term "swapped" only applies to double or float formats. Modbus registers are, by definition, 16 bits of data per register. Access to 32-bit data, either 32-bit integer ("double"), or IEEE 754 floating

point ("float"), is supported by the use of two consecutive registers. Modbus protocol is inherently "big endian", therefore, Modbus by the Module defaults to having the high order register first for double and float. If the low order register comes first on the device being accessed, check the "swapped" box.

If you have "swapped" turned around, you will quickly recognize it. If floating point data is reversed, a 1.0 becomes 2.2779508e-41, which simply rounds to zero. The pattern is not as predictable as the 1.0 example would suggest. A floating point 1.1 becomes negative 107609184. If 32-bit integer data is reversed, 1 becomes 65536.

### 8.2 Modbus RTU Slave Register Mapping

The mappings shown below are used when the BB2-7010 is treated as a Modbus RTU Slave. All objects are accessed as holding registers. Analog registers MUST be read as a register pair, and will return IEEE754 floating point.

#### Analog Input Object

Must be read/written as Floating Point (IEEE 754) register pair, most significant register first.

#### Analog Output Object

Must be read/written as Floating Point (IEEE 754) register pair, most significant register first.

#### Analog Value Object

Must be read/written as Floating Point (IEEE 754) register pair, most significant register first.

#### Binary Input Object

Read/written as a single holding register, any non-zero value written will result in bit set.

#### Binary Output Object

Read/written as a single holding register, any non-zero value written will result in bit

ANALOG INPUT	
Object	Modbus Registers
AI #1	1 (read 1,2 as pair)
AI #2	3 (read 3,4 as pair)
AI #3	5 (read 5,6 as pair)
AI #300	599 (read 599,600 as pair)

ANALOG OUTPUT	
Object	Modbus Registers
AO #1	1001 (read 1001,1002 as pair)
AO #2	1003 (read 1003,1004 as pair)
AO #3	1005 (read 1005,1006 as pair)
AO #100	1199 (read 1199,1200 as pair)

ANALOG VALUE	
Object	Modbus Registers
AV #1	2001 (read 2001,2002 as pair)
AV #2	2003 (read 2003,2004 as pair)
AV #3	2005 (read 2005,2006 as pair)
AV #100	2199 (read 2199,2200 as pair)

BINARY INPUT	
Object	Modbus Registers
BI #1	3001
BI #2	3002
BI #3	3003
BI #300	3300

BINARY OUTPUT	
Object	Modbus Registers
BO #1	4001
BO #2	4002

set.

BO #3	4003
BO #100	4100

#### **Binary Value Object**

Read/written as a single holding register, any non-zero value written will result in bit set.

Read/written as single holding register, treated as unsigned 16-bit integer.

Read/written as single holding register, treated as unsigned 16-bit integer.

BINARY VALUE	
Object	Modbus Registers
BV #1	5001
BV #2	5002
BV #3	5003
BV #100	5100

MULTI-STATE INPUT	
Object	Modbus Registers
MI #1	13001
MI #2	13002
MI #3	13003
MI #300	13300

MULTI-STATE OUTPUT	
Object	Modbus Registers
MO #1	14001
MO #2	14002
MO #3	14003
MO #100	14100

#### Multi-state Value Object

Read/written as single holding register, treated as unsigned 16-bit integer.

MULTI-STATE VALUE	
Object	Modbus Registers
MV #1	19001
MV #2	19002
MV #3	19003
MV #100	19100



# 9. Configuring BB2-7010 as a Modbus TCP Slave

The term "server" is often used to describe the Modbus TCP version of a Modbus slave. A server will provide data when a client asks for it. The concept of master/slave is less significant in Modbus TCP because any TCP device can be both master and slave at the same time, and there can be multiple "masters" on the network. That is in contrast with Modbus RTU where there can be only one master and multiple slaves, and each device must be one or the other.

The Modbus TCP server is simply a collection of registers that may contain data. The source of that data in the case of Babel Buster BB2-7010 can be any of several possible sources. It may be read from another Modbus device. Another Modbus device could have put it there by writing to the BB2-7010. The data could have been received by the BACnet client or BACnet server.

## 9.1 Modbus TCP Device Configuration

1001100110110 100113 101101 Bacnet-M 10010 Network Model BB2-1	Suster 2 DODBUS GATEWAY 7010-01		Control-Sc	DEUTIONS, INC. MINNESOTA
Data Objects	Modbus	BACnet	System Setup	
Setup	HTTP Client	l	l	
Config File	BACnet IP Port	BBMD	Local Host	User
This page allows you to ch accessing this server.	ange this device's IP ad	dress, and select whether	double registers are swapp	ed when returned to a remote client
IP Address	192.168.1.64	192.168.1.64	Review	P
Subnet Mask	255.255.255.0	255.255.255.0	Change	IP
Gateway	192.168.1.1	192.168.1.1		
HTTP Port	80 (default 80	)) 🗌 Disallow HTTP 🤇	uery Set Por	ts
Modbus Port	502 (default 5)	)2)		
Telnet Port	23 (default 2	3)		

The only local device configuration required for Modbus TCP is to set the IP address of the local device. The standard port for Modbus TCP is 502. This can be changed if necessary.

## 9.2 Modbus TCP Slave Register Mapping

The mappings shown below are used when the BB2-7010 is treated as a Modbus RTU Slave. All objects are accessed as holding registers. These mappings are also used when the BB2-7010 is treated as a Modbus TCP Server (slave). The local register numbers may be accessed as holding registers. In addition, if the Modicon mapping option is turned on (TCP only), binary points can be accessed as coils.

#### Analog Input Object

Must be read/written as Floating Point (IEEE 754) register pair, most significant register first.

#### Analog Output Object

Must be read/written as Floating Point (IEEE 754) register pair, most significant register first.

#### Analog Value Object

Must be read/written as Floating Point (IEEE 754) register pair, most significant register first.

#### Binary Input Object

Read/written as a single holding register, any non-zero value written will result in bit set.

#### **Binary Output Object**

Read/written as a single holding register, any non-zero value written will result in bit set.

#### **Binary Value Object**

Read/written as a single holding register, any non-zero

ANALOG INPUT	
Object	Modbus Registers
AI #1	1 (read 1,2 as pair)
AI #2	3 (read 3,4 as pair)
AI #3	5 (read 5,6 as pair)
AI #300	599 (read 599,600 as pair)

ANALOG OUTPUT	
Object	Modbus Registers
AO #1	1001 (read 1001,1002 as pair)
AO #2	1003 (read 1003,1004 as pair)
AO #3	1005 (read 1005,1006 as pair)
AO #100	1199 (read 1199,1200 as pair)

ANALOG VALUE	
Object	Modbus Registers
AV #1	2001 (read 2001,2002 as pair)
AV #2	2003 (read 2003,2004 as pair)
AV #3	2005 (read 2005,2006 as pair)
AV #100	2199 (read 2199,2200 as pair)

BINARY INPUT	
Object	Modbus Registers
BI #1	3001
BI #2	3002
BI #3	3003
BI #300	3300

BINARY OUTPUT	
Object	Modbus Registers
BO #1	4001
BO #2	4002
BO #3	4003
BO #100	4100

BINARY VALUE	
Object	Modbus Registers
BV #1	5001

value written will result in bit set.

BV #2	5002
BV #3	5003
BV #100	5100

#### Multi-state Input Object

Read/written as single holding register, treated as unsigned 16-bit integer.

MULTI-STATE INPUT	
Object	Modbus Registers
MI #1	13001
MI #2	13002
MI #3	13003
MI #300	13300

#### Multi-state Output Object

Read/written as single holding register, treated as unsigned 16-bit integer.

#### Multi-state Value Object

Read/written as single holding register, treated as unsigned 16-bit integer.

MULTI-STATE OUTPUT	
Object	Modbus Registers
MO #1	14001
MO #2	14002
MO #3	14003
MO #100	14100

MULTI-STATE VALUE	
Object	Modbus Registers
MV #1	19001
MV #2	19002
MV #3	19003
MV #100	19100



# 10. Using the BB2-7010-02 as an SNMP Server (Agent)

The BB2-7010-02 can act as an SNMP agent or server. You select which BACnet objects are to show up in the MIB, and the MIB is created dynamically as you fill out the list of objects. Once the MIB is created, any standard v1 or v2 SNMP manager can access the data. Integer data is most universally recognized by SNMP. Floating point support is available in the BB2-7010; however, floating point is not standardized and you should test compatibility.

11 10 01	11001101101101101101101101101101100110							
Da	ita Objects	Modbus	T I	BACnet		SNMP	System	
	SNMP Client	📔 Diagn	nostics		SNMP Agent			
Lo	cal MIB	Traps	ſs	Send Tra	ıps To			
This rem	device can function a ote SNMP Client can a	as an SNMP Agent Ilso write data usir	(or server) pr ng a Set. Note	oviding that th	local data to a s functionality	a remote SNMP Clier is the opposite of	nt when it sends a Get request. The the SNMP Client.	
			Sho	owing 1	to 6 of	6	Update	
Мар #	Loca SNMP (	l DID	Local Object #	s	ale Factor:	Local Value	Local Name	
1	1.3.6.1.4.1.3815.1	1.3.1.1.1.1.2.1	AL1	×1	*	80.7400	Analog Input 1	
2	1.3.6.1.4.1.3815.1	1.3.1.1.1.1.2.2	AI 2	x1	*	0.01208	Analog Input 2	
з	1.3.6.1.4.1.3815.1	1.3.1.1.1.1.2.3	AI 3	×1	*	0.00000	Analog Input 3	
4	1.3.6.1.4.1.3815.1	1.3.1.1.1.1.2.4	AI 4	x1	*	0.00000	Analog Input 4	
5	1.3.6.1.4.1.3815.1	1.3.1.1.1.1.2.5	AI 5	×1	*	0.00000	Analog Input 5	
6	1.3.6.1.4.1.3815.1	1.3.1.1.1.1.2.6	0	×1	*	0.00000		
Re	Reload SNMP Map # 1 Remove Insert Before							

**IMPORTANT:** The definition of Input versus Output object is from the perspective of the BACnet network. Therefore your SNMP client should Write to Input objects to provide input to BACnet, and Read from Output objects to receive output from BACnet. Attempting to write a BACnet Output object from SNMP will not work properly. You must think of your SNMP manager as the physical I/O being accessed from BACnet. If you want to make your SNMP manager write to an Output object on another BACnet device, use the BACnet client mapping to translate a local Input to remote Output on the BACnet side. Rule number simply tells you where you're at on the list of the local SNMP Agent's OID maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Showing" box, then click Update.

This page enables SNMP Get/Set to objects indicated on the above map list. The available local OID's are assigned automatically. You may select which local BACnet objects are mapped to these OID's. The only data type supported via the internal SNMP Agent is signed integer, therefore you must use scaling to provide real data as integers. This is an inherent limitation of SNMP which does not have any universally accepted method of transmitting floating point data.

Internal data is multiplied by the scale factor when read by your remote SNMP manager (client). Data written by your SNMP client is divided by the scale factor before being stored internally.

For each local object to be accessed by the remote SNMP Client, enter the local object number and scale factor. The local data and object name will be shown for reference. The data returned to the remote SNMP client will be the indicated local value multiplied by the scale factor, then truncated to integer. Enter an object number, then click Update to add the mapping to the list.

Objects are not immediately available when entered in the list above. When you have finished making changes, click the Reload SNMP button to clear and reload the MIB. The MIB is also automatically reloaded every time you restart this device.

Entering zero (none) for local object effectively deletes the rule even though it will still appear in the list until deleted. Unused rules at the end of the list will always show none as the type.

Local Object is internally a coded number consisting of BACnet object type multiplied by 1000, then added to the object number starting from #1. These are translated into abbreviations that are easy to interpret on the web page as follows:

Al n = Analog Input #n AO n = Analog Output #n AV n = Analog Value #n BI n = Binary Input #n BO n = Binary Output #n BV n = Binary Value #n MI n = Multi-state Input #n MO n = Multi-state Output #n MV n = Multi-state Value #n

Object numbers start at #1. The maximum available number varies by object type, and these limits may be found on the System Capacities link from the home/index page (click graphic at top).

00110011011 110011 101101 Bachel 101101 Bachet-A 010010 Network Model BB2	Butter 2 MODBUS GATEWAY -7010-02		Control	Solutions, In Minneso	NC. DTA
Data Objects	Modbus	BACnet	SNMP	System	
SNMP Client	Diagno stics	SNMP A	gent 🔶	l l	l
Local MIB	Traps	Send Traps To			
Each variable in the local for the given local object.	MIB can have an optiona	I trap associated with it	. If a trap is enabled, it i	is generated according to th	e following rule
oid # 1	Rule presently tests FA	LSE		Update < Pr	ev Next >
Read local object $f AI~1$ na	med Analog Input 1				
Event is TRUE if the value	is Greater than	✓ this value: 85.000			
Qualified by this hysteresi	is value: 0.00000 this	minimum On Time: 🛛	:00:00 this minimu	um Off Time: 0:00:00	
Trap on True	p on False Rep	eat Count: 0	Repeat Time:	0.0	

OID number simply tells you where you're at on the list of the local SNMP Agent's OID maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Showing" box, then click Update. You cannot proceed to a trap rule for an OID that has not been defined on the Local MIB page.

Select a comparison or test, and click the button for your choice of what the local register should be compared to. Then enter either the fixed value for threshold.

Quaifications are optional, and enabled only when values are nonzero. How hysteresis is applied depends on the comparison. For a test that becomes true if greater than, the test will not return to false until the local register is less than the test value by a margin of at least this hysteresis value. If a test becomes true if less than, it will not return to false until the local register is greater than the test value by a margin of at least this hysteresis value by a margin of at least this hysteresis value.

On time and off time, if specified, determine how long the condition must be true (on time) or false (off time) before the true or false response is actually taken. Times are given in HH:MM:SS format (hours, minutes, seconds).

The repeat count is the number of times the same trap will be sent when triggered. This number of traps will be sent at approximately 100 millisecond intervals. The repeat time is the delay period between re-transmissions of the trap, or series of traps as determined by the repeat count. Repeat time is in seconds. Example: If repeat count is set to 3, and repeat time is set to 60 seconds, then three trap messages will be sent in a burst and this burst will be repeated once every minute.

00011001101 1100112110 101101 Balen 101101 Bacnet 010010Netwo MODEL BI 0000010101	Buster 2 F-MODBUS RK GATEWAY B2-7010-02		Control	-Solutions	5, Inc. nesota
Data Objects	Modbus	BACnet	SNMP	System	
SNMP Clien	nt Diagnostics	SNMP Agen	t 📜		)
Local MIB	Traps	Send Traps To			
Traps are sent to the o this page.	devices selected via this pag	ge. Identification of the loca	al device as provideo	d to remote SNMP clien	ts is also entered on
Device #	1			Update	<pre> Next&gt;</pre>
IP Address	192.168.1.111				
System Name	Babel Buster BB2-7010 M	lodbus to BACnet IP Gatev	way		Update
System Location	USA				
System Contact	www.csimn.com				
Local Community	snickers				

Traps generated by this device will be sent to port 162 on each IP address listed above. The name, location, and contact listed above may be retrieved by the remote SNMP client. The local community is the name that must be used by the remote SNMP client to write to this device. The name "public" is accepted for reading.



# 11. Using the BB2-7010-02 as an SNMP Client (Manager)

The BB2-7010-02 has the ability to be an SNMP client. In "master/slave" terms, this would be the master. Configuring the SNMP client starts with defining one or more SNMP devices that will be queried. Then, like the other possible client functions in the BB2-7010-02, you set up read and write maps. A "read map" will use SNMP Get to query the device, and a "write map" will use SNMP Set to write to the device.

The SNMP Client configuration pages are illustrated below along with a summary of how to use them.

00011001101 110011 101101 Babel 010010 Bacnet 010010 Networ Model BB: 10000010	Buster 2 MODBUS K GATEWAY 2-7010-02		Control	Solutions, In Minneso	IC. DTA
Data Objects	Modbus	BACnet	SNMP	System	
SNMP Client	Diagnostics	SNMP Ag	ent 👔	l	
Devices	Client Read Map	Client Write Map			
This page sets up the n The local device acts as	etwork address for a remo an SNMP dient (manager)	e SNMP device that will l to the remote agents li:	oe linked to local objects sted below.	s via the client read and clie	nt write maps.
Device # 1				Update < Pr	ev Next >
IP Address 1	92.168.1.142 Local Local	Name: AddMe Jr.			
SNMP Version (	○v1 ⊙v2c				
SNMP Community r	nongoose				
Default Poll Period	2.0 Seconds		Devic 4	e Status Reset	

Device number simply shows you where you are on the device list. Click "next" and "prev" to scroll through the list.

Remote SNMP devices to be accessed by this device are specified here. Enter the IP address of the remote device, a name to reference in other pages, and a default poll rate. Then click "update".

This gateway expects to access SNMP devices via the standard port 161.

Connection status will show a non-zero error code if there is a socket error. Possible errors include:

- 5 = Connection failed, unable to bind (usually means remote device not connected or not reachable)
- 81 = Connection in progress (means unsuccessful connect attempt, still trying)
- 95 = Network is unreachable
- 97 = Connection aborted
- 98 = Connection reset by peer
- 103 = Connection timed out
- 104 = Connection refused
- 107 = Host is unreachable

0011001101 10011Babel 10110Babel 010010Networ 010010Networ 000001	Buster 2 MODBUS K GATEWAY 2-7010-02			Contro	L-Solu	fions, Inc. Minnesota
Data Objects	Modbus	ВА	Cnet	SNMP		System
SNMP Client	Diagnostics		SNMP Agent			
Devices	Client Read Map	Clie	ent Write Map	$\int$		
Read remote SNMP OIDs into local objects. This page creates a map entry that reads data from remote SNMP agents for processing here. Click on map number to see more detail and insert/delete rules.						
		Showi	ng 1 to 4 of 4	4	U	odate <prev next=""></prev>
Map #	Remote SNMP OID		Remo Devic	te :e	Local Object #	Local Object Name

1	1.3.6.1.4.1.3815.1.2.2.1.1.2.1.1.2.1	AddMe Jr. 💌	Al 1	Analog Input 1
2	1.3.6.1.4.1.3815.1.2.2.1.1.2.1.1.2.2	AddMe Jr. 💌	AI 2	Analog Input 2
<u>3</u>	1.3.6.1.4.1.3815.1.2.2.1.1.2.1.1.2.3	AddMe Jr. 💌	AI 3	Analog Input 3
<u>4</u>		None 💌	0	_

Rule number simply tells you where you're at on the list of OID maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Showing" box, then click Update.

Rules entered on this page only read data from remote devices. Go to the Client Write Map to write data to those devices. The full parameter set is different for read versus write.

An abbreviated version of a list of rules is shown on this page. Any of the parameters shown may be changed here and registered by clicking the Update button. To view and/or modify the complete set of parameters, click on the map number in the left most column.

For each remote OID to be read, enter the full SNMP OID and location (device). The names in the device list are defined in the Devices page.

The object name is optional and used only for display purposes, but is also returned as the object name to the remote BACnet client.

Entering zero (none) for local object effectively deletes the rule even though it will still appear in the list until deleted. Unused rules at the end of the list will always show none as the type.

Local Object is internally a coded number consisting of BACnet object type multiplied by 1000, then added to the object number starting from #1. These are translated into abbreviations that are easy to interpret on the web page as follows:

Al n = Analog Input #n AO n = Analog Output #n AV n = Analog Value #n BI n = Binary Input #n BO n = Binary Output #n BV n = Binary Value #n MI n = Multi-state Input #n MO n = Multi-state Output #n MV n = Multi-state Value #n

Object numbers start at #1. The maximum available number varies by object type, and these limits may be found on the System Capacities link from the home/index page (click graphic at top).

0001100110110 110011 101101 BACNET-M 10010 NETWORK MODEL BB2-7	Surter 2 DODBUS GATEWAY 7010-02		Control-S	Solutions, Inc. Minnesota
Data Objects	Modbus	BACnet	SNMP	System
SNMP Client	Diagnostics	SNMP Agent		
Devices	Client Read Map	Client Write Map		
This page creates a map e	ntry that reads data from a	remote SNMP agent for	processing here.	
Мар # 1				Update <prev next=""></prev>
Read OID 1.3.6.1.4.1.3815.	1.2.2.1.1.2.1.1.2.1	from AddMe Jr. 👻		
Then apply scale: 0.00000	and offset: 0.00000			
Save in local object # Al 1	named Analog Input	t1 R	epeat this process eve	ery 10.0 seconds.
Apply this default value: -1	1.00000 after 2 rea	ad failure(s).		
Initial COV increment: 0.00	1000 Initial COV Perio	d: 0 (sec.) U	nits: no_units	*
# Client Read Maps Enabled	d: 4			Insert Delete

Rule number simply tells you where you're at on the list of OID maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Map #" box, then click Update.

For each remote OID to be read, enter the full OID and location (device). The names in the device list are defined in the Devices page.

When the remote OID is read, data may be manipulated before being written to the local object. The result will be multiplied by the scale factor if any non-zero scale factor is given. The offset is then added and this final result is written to the local object number given. The name is optional and used only for display purposes (but will also be returned as the object name to the BACnet client).

The periodic poll time determines how often the remote OID will be read. This number, if nonzero, will override the default poll time given in the Devices page for the remote device being read.

The default value will be stored into the local object after the given number of read failures if the fail count

is non-zero. Setting the count to zero will disable the default, and the object will retain the most recent value obtained.

Delete will remove the rule number shown in the "Map #" box. Insert will insert a new rule before the rule number shown, and is used for placing rules between existing rules. It is not necessary to use Insert to add rules to the bottom of the list or to define any rule presently having zero for a source object or "none" for remote type.

Entering zero (for none) for local object effectively deletes the rule even though it will still appear in the list until deleted. Unused rules at the end of the list will always show none as the type. If you wish to prevent these from being displayed, reduce the number of rules enabled.

Initial COV increment and period will only apply if a BACnet client subscribes to COV notification from the BACnet object assigned to this SNMP client map. These properties may be overwritten by the BACnet client(s) at any time. The values shown here are initial values, not necessarily the current values. (Note: COV increment only applies to Analog objects, all changes are reported for Binary or Multistate objects.)

Units default to no\_units, but you may select any of the available BACnetEngineeringUnits values. This value will simply be read by the BACnet client when the units property is requested from the object this OID maps to. The units have no bearing on calculations performed. You must select appropriate scale and offset values to make any required translation between SNMP units and BACnet units. Units are only valid for Analog objects.

The number of rules enabled simply limits the scope of rule review so that you do not have to review a lot of unused rules. If the displayed rules are used up and you need more, increase the enabled number.

10011001 100110 101101 1011018 10010	abel Buster 2 CNET-MODBUS TWORK GATEWAY DEL BB2-7010-02		Cor	TROL SOLUTI	ONS, INC. MINNESOTA
Data Objects	Modbus	BACnet	SN	мр 🔰 Sy	stem
SNM	P Client Diagnosti	cs 🔰 SNMP /	\gent	J	l
Devices	Client Read Map	Client Write Ma			
Write local object here. Click on m	ts out to remote SNMP OIDs. Th ap number to see more detail	his page creates a map e and insert/delete rules.	entry that write	s data remote SNMP agen	ts from data contained
		Showing 1 t	o 2 of 2	Upd	ate < Prev Next >
Map Local # Object #	Remote SNMP OID	Rem Data 1	ote Type	Remote Device	Local Object Name
1 BO1	1.3.6.1.4.1.3815.1.2.2.1.1.2.	1.1.2.1 Unsigned	~	AddMe Jr. 💌	Binary Output 1
<u>2</u> 0		Undefined	*	None 💌	_

Rule number simply tells you where you're at on the list of OID maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Showing" box, then click Update.

Rules entered on this page only write data to remote devices. Go to the Client Read Map to read data from those devices. The full parameter set is different for read versus write.

An abbreviated version of a list of rules is shown on this page. Any of the parameters shown may be

changed here and registered by clicking the Update button. To view and/or modify the complete set of parameters, click on the map number in the left most column.

Data from the local object given will be multiplied by the scale factor before being written. For each remote OID to be written, enter the register type, format, number, and location (device). The names in the device list are defined in the Devices page. The name is optional and used only for display purposes.

Important note about data type: SNMP does not have a universally accepted representation for floating point. The most commonly used means of representing real data is scaled integers, and this method is supported by BB2-7010. IEEE 754 is not recognized as an SNMP standard and is not used. X.690 defines an encoding for real data, but it is inefficient and little used. A common recommendation is to use ASCII string representation of floating point data, and this method is supported by BB2-7010 (Octet String Num). Another known but application specific implementation is the ASN OPAQUE FLOAT used in netsnmp applications. This method is also supported by BB2-7010 but should be tested to confirm compatibility.

Selecting "none" for remote type effectively deletes the rule even though it will still appear in the list until deleted. Unused rules at the end of the list will always show none as the type.

Local Object is internally a coded number consisting of BACnet object type multiplied by 1000, then added to the object number starting from #1. These are translated into abbreviations that are easy to interpret on the web page as follows:

AI n = Analog Input #n AO n = Analog Output #n AV n = Analog Value #n BI n = Binary Input #n BO n = Binary Output #n BV n = Binary Value #n MI n = Multi-state Input #n MO n = Multi-state Output #n MV n = Multi-state Value #n

Object numbers start at #1. The maximum available number varies by object type, and these limits may be found on the System Capacities link from the home/index page (click graphic at top).

10011001101100110110011001100110011001	Mer 2 BUS TEWAY 0-02		Control Soi	UTIONS, INC. Minnesota
Data Objects Mo	odbus	BACnet	SNMP	System
SNMP Client	Diagnostics	SNMP Agent	l	
Devices	ient Read Map	Client Write Map		
This page creates a map entry	that writes data to rem	ote SNMP agents from da	ta contained here.	
Map # 1				Update < Prev Next >
Read local object # BO 1 r	named Binary Output 1	1		
Apply default value of 0.00000	at power-up and,	/or when 0.0 s	econds have elapsed with	no host update.
Write remote OID 🔲 any time l	local object has change	d by 0.00000 or 🔲 wh	en 0.0 seconds h	ave elapsed with no change.
Otherwise write remote OID unco	onditionally. In any eve	ent, when writing remote O	ID, apply local object data	a as follows:
Apply scale: 0.00000 and off	fset: 0.00000			
Write OID 1.3.6.1.4.1.3815.1.2.2	2.1.1.2.1.1.2.1	as Unsigned	👻 at 🛛 AddMe Jr. 💌	
Repeat this process 💿 at least	Ono more than ever	y 2.0 seconds.		
Initial COV increment: 0.00000	Initial COV Period	d: O (sec.) Uni	ts: no_units	~
Initial Relinquish Default: 0.000	000			
# Client Write Maps Enabled: 2				Insert Delete

Rule number simply tells you where you're at on the list of OID maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Map #" box, then click Update.

The local object data may be written periodically, or when it changes, or both. To send upon change (send on delta), check the first box and enter the amount by which the local object must change before being written to the remote device. To guarantee that the remote OID will be written at least occasionally even if the data does not change, check the second box and enter some amount of time. This time period will be referred to as the "maximum quiet time".

Data from the local object may be manipulated before being written to the remote OID. The local data is first multiplied by the scale factor. The offset is then added to it. The data is then sent to the remote SNMP agent. Enter the full OID to be written, the SNMP ASN data type to be written (select from list), and the location (device). The names in the device list are defined in the Devices page.

Important note about data type: SNMP does not have a universally accepted representation for floating point. The most commonly used means of representing real data is scaled integers, and this method is supported by BB2-7010. IEEE 754 is not recognized as an SNMP standard and is not used. X.690 defines an encoding for real data, but it is inefficient and little used. A common recommendation is to use ASCII string representation of floating point data, and this method is supported by BB2-7010 (Octet String Num). Another known but application specific implementation is the ASN OPAQUE FLOAT used in netsnmp applications. This method is also supported by BB2-7010 but should be tested to confirm compatibility.

The repeat time may determine how often the remote OID will be written. If send on delta and maximum quiet time are not checked above, clicking the "at least" button will establish a periodic update time. If send on delta is used and you wish to limit the network traffic in the event changes are frequent, click the "no more than" button and enter the minumum time that should elapse before another write to the remote device.

Delete will remove the rule number shown in the "Map #" box. Insert will insert a new rule before the rule number shown, and is used for placing rules between existing rules. It is not necessary to use Insert to add rules to the bottom of the list or to define any rule presently having zero/none for a source object.

Selecting "none" for remote type effectively deletes the rule even though it will still appear in the list until deleted. Unused rules at the end of the list will always show none as the type. If you wish to prevent these from being displayed, reduce the number of rules enabled.

Initial COV increment and period will only apply if a BACnet client subscribes to COV notification from the BACnet object assigned to this Modbus map. These properties may be overwritten by the BACnet client(s) at any time. The values shown here are initial values, not necessarily the current values. (Note: COV increment only applies to Analog objects, all changes are reported for Binary or Multistate objects.)

Units default to no\_units, but you may select any of the available BACnetEngineeringUnits values. This value will simply be read by the BACnet client when the units property is requested from the object this OID maps to. The units have no bearing on calculations performed. You must select appropriate scale and offset values to make any required translation between SNMP units and BACnet units. Units are only valid for Analog objects.

Initial Relinquish Default may be set here, but may be overwritten by the BACnet client at any time. This window reflects the initial value, not the current value. (Note: Relinquish Default only applies to commandable Output objects, and does not apply to Input or Value objects.)

The number of rules enabled simply limits the scope of rule review so that you do not have to review a lot of unused rules. If the displayed rules are used up and you need more, increase the enabled number.

100110011011001 10011Babel Buster 2 BACNET-MODBUS 10010NETWORK GATEWAY MODEL BB2-7010-02	Contro	L-Solutions, Inc. Minnesot	Ā
Data Objects Modbus BACn	et SNMP	System	
SNMP Client Diagnostics	SNMP Agent		
Errors: Read Maps Errors: Write Maps	ſ		
This page displays errors, if any, reported by the remote serv	er upon attempts to read from th	at device.	
		<b>&lt;&lt;</b> Top	Next >
Map Remote OID	Remote Device	Local Name	Error Code
			0
		R	eset Errors

Errors for SNMP client read maps are shown on this page. Only those maps with errors to report are listed. Refer to the code and class lists below for interpretation.

Common error codes for the SNMP client are as follows:

9 = No response from remote Agent (server)

- 10 = Unable to interpret data
- 11 = Reply does not match request

Other error codes are possible but improbable. Codes in the 80-120 range indicate socket errors; however, because SNMP uses UDP/IP, which is "connectionless", socket errors would indicate something internal is seriously broken.



# **12. HTTP Client**

The Babel Buster BB2-7010-01 can function as an HTTP Client to push data to a specially programmed remote server such as the web portal you find at www.logmydata.com. You may develop your own web server for this purpose. Contact Control Solutions via support ticket for guidance on developing such a web server application.

Once you have access to a web server application ready to receive data, you simply list the objects you wish to send to the server in the list illustrated below.

000110 110011 101101 010010 010000	Bachel B Bachet-Mo Network o Model BB2-7	ODBUS GATEWAY 010-01		Control-S	OLUTIONS, INC. MINNESOTA
Data Objec	ts	Modbus	BACnet	System Setup	
Se	etup	НТТРО	Client		
Data Points	s )		J	J	
This page di	splays a list of	points being logo	ged via the HTTP Client.		
				•	
			Showing points from		Update Prev Next>
Point #	Local Objec	t	Object Name		
1	Al 1		Volts		
2	AI 2		Amps		
3	AI 3		Hertz		
4	AI 4		Watts		
5	0				

There are additional settings that need to be configured on the Local Network page if the HTTP Client is used. This is the only time these settings need to be configured. You will need to provide the IP address(es) of your DNS Server(s). You will need to provide the URL of the web server that is to receive data, and the web page that the data is to be submitted to. The Client ID uniquely identifies this device when you have more than one reporting to the same web portal. The minimum additional settings are illustrated below.

1001 BACNET-M 1001 NETWORK MODEL BB2-7	ODBUS GATEWAY 1010-01		Control	SOLUTIONS, MINN	Inc. esota
Data Objects	Modbus	BACnet	System Setup		
Setup	HTTP Clier	nt 👔			Ĵ
Config File	BACnet IP Port	BBMD	Local Host	User	
accessing this server.					
IP Address	192.168.1.64	192.168.1.64	Re		
Subnet Mask	255.255.255.0	255.255.255.0	Ch	ange IP	
Gateway	192.168.1.1	192.168.1.1			
Static DNS1	68.87.77.130	68.87.77.130	S	et DNS	
Static DNS2	68.87.72.130	68.87.72.130			
Proxy Server			Se	t Proxy	
Proxy Port	0				
HTTP Port	80 (default	80) Disallow HTTP	Query S	et Ports	
Modbus Port	502 (default	502)			
Telnet Port	23 (default	23)			
			_		
Client Update Host	logmydata.net	En	able	pdate	
Log Page	/ireport.php				
Log Parameters		(optional	)		
Configuration Page					
Update Timeout	60 ci	ient ID 900064	HTTP Port 80		
Optime	1,02:29:30				



# **13. BBMD Configuration**

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

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 BB2-7010 only includes BBMD, not full routing. Use the BB2-7030 if you need full routing.

If you have a remote BB2-7010 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.

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

The screen shot below shows a BB2-7010 that has successfully registered with a BBMD at the IP address shown.

00011001101 110011 101101 Babel 101101 Bacnet 010010 Netword Model BB	Buster 2 -MODBUS RK GATEWAY 52-7010-01		Control-So	LUTIONS, Minni	Inc. esota
Data Objects	Modbus	BACnet	System Setup		
Setup	НТ ТР СІ	ient 👔	1		Ĵ
Config File	BACnet IP Port	BBMD	Local Host	User	
This page displays BAC	net IP port settings.				
Broadcast Distribu	tion Table			<u>Edit</u>	Refresh
	Broadca	st Address:Port	Broadcast Mask		
	192.16	3.1.191:47808	FFFFF00		
Foreign Device Tab	ble				Refresh
82	Local Dev	ice's Registration as a Fo	reign Device at Remote Location		
BBMD Time To I	Live (seconds) 600	(Zero disables for	reign registration)	ave	

Foreign Devices Registered Locally	Time to Live

BBMD is registered.

47808

BBMD IP Address, Port 173.11.32.82



# **14. Object Properties**

## 14.1 Data Object Properties (Analog, Binary, Multi-state)

The following properties are found in the Analog, Binary, and Multi-state types of Input, Output, and Value objects. Some properties apply only to certain object types as noted where applicable.

Property	Encoding
Object_Identifier (75)	BACnetObjectIdentifier
Object_Name (77) (W)	CharacterString "Analog Input <i>n</i> "
Object_Type (79)	BACnetObjectType ENUMERATED: analog-input (0) analog-output (1) analog-value (2) binary-input (3) binary-output (4) binary-value (5) device (8) multi-state-input (13) multi-state-output (14) multi-state-value (19)
Present_Value (85) (W)	REAL (analog objects) ENUMERATED (binary objects) Unsigned (multi-state objets) (no index) (priority required when writing commandable objects) (input objects 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 (64) crc error (65) exception, illegal function code (66) exception, illegal data address (67) exception, illegal data value (68)

	exception, code+65, rarely used (6979) configuration property fault (80) exception, code not recognized (81) BACnet client read/write timeout (82) BACnet client received error response from slave (83) SNMP client received no response from agent (84) SNMP client unable to parse data (85) SNMP client reply does not match request (86)
Out_Of_Service (81) (W)	BOOLEAN
COV_Increment (22) (W)	REAL (analog objects only)
Priority_Array (87)	BACnetPriorityArray (commandable objects only) SEQUENCE SIZE (16) OF BACnetPriorityValue REAL (each element, analog output objects) ENUMERATED (each element, binary output objects) Unsigned (each element, multi-state output objects)
Relinquish_Default (104) (W)	REAL (analog objects) ENUMERATED (binary objects) Unsigned (multi-state objets)
Polarity (84)	BACnetPolarity (binary objects only) ENUMERATED: normal(0)
Number_Of_States (74)	Unsigned (multi-state objects only)
Units (117)	BACnetEngineeringUnits (analog objects only)

## 14.2 Device Object Properties

The following properties are found in the Device object of the BB2-7010. In addition to standard Device properties.

Property	Encoding
Object_Identifier (75)	BACnetObjectIdentifier
Object_Name (77)	CharacterString
Object_Type (79)	BACnetObjectType ENUMERATED: device (8)
System_Status (112)	BACnetDeviceStatus
Vendor_Name (121)	CharacterString
Vendor_Identifier (120)	Unsigned16 (should always return 208)
Model_Name (70)	CharacterString
Fimrware_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



## **15. Trouble Shooting**

The following discussion focuses on Modbus Master functionality. Troubleshooting for the BACnet or SNMP clients will be very similar and you should be able to use the examples here to help trouble shoot.

#### 15.1 Modbus RTU Trouble Shooting

This discussion assumes you want the Babel Buster BB2-7010 to be the Modbus Master (most common use). Let's review the setup procedure for a single Modbus read map. We suggest starting with one register. Once you get that working, proceed to fill up the table.

First, go to the Local Device page and make sure you have the baud rate set, and parity (if any) selected. If you do not know what baud rate your Modbus device is set to, consult that manufacturers documentation before proceeding.

Make sure the Master button is clicked. Start with a liberally slow timeout, like 0.5 second just to be rather certain you do not have timeout problems. It is rare to see a piece of working equipment take longer than half a second to respond to a Modbus master. *Setting the timeout to zero, however, will guarantee failure* since the master will miss every reply by not waiting at all for it.

Data Objects	Modbus	BACnet	SNMP	System						
Modbus RTU	Data 🔰 Modbus RTU S	Setup Modbus TCP I	Data 👖 Modbus TCP S	Setup	Ì					
Local Device	RTU Read Map	RTU Write Map								
This page displays config	This page displays configuration parameters for the Modbus RTU serial port.									
Baud Rate	19200 - Parity N	one, 1 Stop Bit 🛛 👻		Update						
I am the M	aster 🍳	I am a Slav	e 🔘							
Paremeters	for RTU Master:	Parameters	for RTU Slave:							
Default Pol	I Rate 2.000 Seconds	My Address	or Unit # 0							
Timeout 1.	.000 Seconds	🗖 Double	registers are swapped							
🔲 Use FC	Use FC 5/6 instead of 15/16 for unit numbers (slave addresses) starting at 0									
🔲 Use FC	5/6 and 15/16 by count for	r unit numbers (slave addre	esses) starting at 0							

Next, go to the RTU Read Map page (below) . To get started, select a register type and format, a register number, a unit # (aka slave ID or slave address), and a local object number to store the data in. If any of the red check marks shown below are "none" or zero, you will get no action even attempted. Make sure the Unit # (slave ID or slave address) matches whatever you have your Modbus device set to. If you are uncertain what address it is set to, you need to consult the manufacturer's documentation for that

equipment before proceeding.

The following example shows the only non-zero entries required (the 5 check marks) to successfully read holding register #22 from unit #1 and store the data in Analog Input #10. Once these (or comparable) entries have been made, click the Update button.

Da	ıta Objects	Modbus	BACnet		SNMP	]	System					
Modbus RTU Data Modbus RTU Setup Modbus TCP Data Modbus TCP Setup												
Local Device RTU Read Map RTU Write Map												
Rea devi	Read remote registers into local registers. This page creates a map entry that reads data from one or more remote Modbus RTU serial devices for processing here. Click on map number to see more detail and insert/delete rules.											
	Showing 1 to 2 of 2 Update < Prev Next >											
Map #	Remote Type	Remote Register Formet	Remote Register #p	Remote Unit #p	Scale	Local Object #	Name					
1	Holding Register	Int 16-bit	22	1	0.00000	AI 10 🔨	Modbus Reg 22					
2	None -	Int 16-bit 👻	0	0	0.00000	0						

At this point, you can go to the data page (below) and see if you have data showing up. If you get no data, there is a problem. The confirmation that you are probably getting no data is the "time since last update". In this example, we see 95 seconds have elapsed. We are attempting to update every 2 seconds, so obviously data retrieval is not happening.

Data	Objects	Modbu	s BACn	et	SNMP			System	System		
	Modbus RTU	Data	Modbus RTU Setup	Modbus	TCP Data	М	odbus TCF	P Setup 🚶			
RTU Registers Error Codes											
This pa	This page displays data to and from registers in devices accessed via the Modbus RTU serial port.										
RTU Uni	t#1		si	howing 1	to 1 of 1	L		Update	<pre></pre>	v Next>	
Dir.	Reg. Type	Remote Reg. #	Register Name		Local Object #	Hex	Update	Register	Data	Time since Last update	
From	Holding Reg	00022	Modbus Reg 2	22	AI 10			0.00000		95.227	

If you are getting no data, check the Error Codes page (below). Here we see that the "No Responses" is about equal to the "Total Messages". This means we are not getting anything back from the Modbus slave. If you are certain all of the above setup is correct, the only conclusion you (or we) can come to at this point is that there is a wiring problem, or the slave is not responding or not configured correctly. Review wiring information, and check the slave configuration. If you get a high number of CRC errors, this is also an indication of likely wiring problem.

Data	i Objects		Modbus	1	BACnet	<u> </u>	SNMP		System	n	
	Mod	lbus RTU Data	Modb	us RTU S	ietup 🚺 Ma	odbus TCP Dat	ta 👔	Modbus TCF	9 Setup		Ì
RTU Registers Error Codes											
This p	age disp	lays error code	es encountered	in proces	sing reads and	d writes via the	Modbus	RTU serial po	rt.		
								~			
				s	nowing devices	from 1		(3)	Update	<pre></pre>	Next>
Unit #	Reset >	Read Error	Offending Read Map #	Reset >	Write Error	Offending Write Map #	Reset	Total Messages	No Responses	CRC Errors	Exceptions
1		5/0	1		0/0	0		222 🚝	221	0	0
2		0/0	0		0/0	0		0	0	0	0

If, instead of No Repsonses, the count you see is Exceptions, this means you are communicating just fine, but the slave is telling you that your request is incorrect. You are asking for a register number that does not exist, using the incorrect register type, etc. Something about configuration is not right if you get Exception errors.

If the Error Codes page is showing a problem, this will also be reflected by the BACnet object reliability code and status bits as illustrated below.

Data	Data Objects Modbus		]	BACnet S		SNMP		System	
	Analog	Binar	y	Mult	i-State			) (	Ì
Input Objects Value Objects									
This pa	ige displays data a	s presently found	in the lo	cal objects maint	ained by this	s device.			
Analog	Input Objects		s	howing objects fr	om 10		[	Refresh < Prev	Next>
Object	Object Name		Out of Service	Present Value	Reliability	Status	Units		
<u>10</u>	Modbus Reg 22		N	0.00000	64	0,1,0,0	no_units		

Once the problem is resolved and you are successfully receiving data, the BACnet object will reflect a reliability code of zero and the fault bits will be clear.

Data	Data Objects Modbus			BACnet		SNMP		System		
	Analog	Binar	У	Mult	i-State					Ĵ
Input	t Objects	Output Object	s	Value Objects	s )			$\mathbf{r}$	l	
This pa	age displays data	as presently found	in the lo	cal objects maint	ained by this	device.				
Analog	Input Objects		s	howing objects fr	om 10			Refresh	< Prev	Vext>
Object	Object Name		Out of Service	Present Value	Reliability	Status	Units			
<u>10</u>	Modbus Reg 22		N	45.000	0	0,0,0,0	no_units			

### 15.2 Modbus TCP Trouble Shooting

This discussion assumes you want the Babel Buster BB2-7010 to be the Modbus TCP Master. Let's review

the setup procedure for a single Modbus read map. We suggest starting with one register. Once you get that working, proceed to fill up the table.

First, go to the IP Network Devices page and make sure you have the IP of the intended Modbus TCP slave entered, along with a local name by which it will be referenced.

Data Objects	Modb	us	BACnet	SNMP		System		
Modbus RT	UData	Modbus RTU S	ietup 🚺 Modbi	us TCP Data 🔰	Modbus TCP 9	Setup		
Devices	Client	Read Map	Client Write Ma	ap Server	Мар	<u></u>		
This page sets up the via the client read and	network add client write (	lress and optional maps. The local de	device parameters avice acts as a Mod	for a remote Mod dbus master to the	bus/TCP device t remote devices	hat will be link listed below.	ked to local o	bjects
Device #	1				(	Update	< Prev I	Next>
IP Address	192.168.1.1	0 Port 502	Local Name:	Test Device				
Unit (optional)	0	🔲 Use FC 5/6 i	nstead of 15/16	Use FC 5/6	and 15/16 by co	ount		
	Low reg	ister is first for mu	ltiple registers		Connection S	tatus		
Default Poll Period	2.0	Seconds			0			

Next, go to the Click Read Map page (below). To get started, select a register type and format, a register number, a device (from list created above), and a local object number to store the data in. If any of the red check marks shown below are "none" or zero, you will get no action even attempted. Make sure the IP address (in the device list) matches whatever you have your Modbus device set to. If you are uncertain what IP address it is set to, you need to consult the manufacturer's documentation for that equipment before proceeding.

The following example shows the only non-zero entries required (the 5 check marks) to successfully read holding register #23 from "Test Device" and store the data in Analog Input #11. Once these (or comparable) entries have been made, click the Update button.

Da	ta Objects	Modbus	BACnet		SNMP			
Modbus RTU Data Modbus RTU			TU Setup	Modbus TCP Data	Modbus TCP	Setup	]	
De	vices	Client Read Map	Client V	Write Map Serv	/er Map			
Rea	Read remote registers into local objects. This page creates a map entry that reads data from one or more remote Modbus/TCP servers for processing here. Click on map number to see more detail and insert/delete rules.							
			Showing	1 to 2 of 2		Update	< Prev Next >	
Map #	Remote Type	Remote Register Format	Remote Register #	Remote Device 🍺	Scale	Local Object #	Name	
<u>1</u>	Holding Register 🕊	Int 16-bit	23	Test Device	0.00000	Al 11 🔨	TCP Reg 23	
2	None -	Int 16-bit 👻	0	None -	0.00000	0		

At this point, you can go to the Data Objects page (below) and see if you have data showing up. If you get no data, there is a problem, and the problem will be further indicated by the BACnet object reliability code being non-zero, and the fault status bit set. The meaning of the various possible reliability codes is listed in the Quick Help section at the bottom of each Data Object page. Code 64 means "no response" from the Modbus device.

Data	ata Objects Modbus		1	BACnet		SNMP	Syst	System	
	Analog	Binaņ	y	Mult	i-State				Ì
Input	Objects	Output Object	s	Value Objects	s )			)	
This pa	age displays data a	as presently found	in the lo	cal objects maint	ained by this	s device.			
Analog	Input Objects		s	howing objects fr	om 11		Refres	h < Prev	Next>
Object	Object Name		Out of Service	Present Value	Reliability	Status	Units		
<u>11</u>	TCP Reg 23		N	0.00000	64	0,1,0,0	no_units		

If you are getting no data, check the Error Codes page (below). Here we see that the "No Responses" is some number greater than total messages. Zero total messages means we never succeeded in making a TCP connection.

Data	Data Objects Modbus				BACnet	T T	SNMP		System	n	
	Mod	lbus RTU Data	Modb	us RTU S	etup 🚺 Ma	odbus TCP Dat	a	Modbus TC	P Setup 👔		l
Mod	ibus Serv	rer 🚺	Error Codes	$\neg$		ſ					٦
This p	This page displays error codes encountered in processing Modbus Client reads and writes via the Modbus TCP connection(s).										
									(C)		Update
Device	Reset >	Read Error	Offending Read Map #	Reset >	Write Error	Offending Write Map #	Reset >	Total Messages	No Responses	Exceptions	
1		0/5	1		0/0	0		0	7	0	

If you return to the TCP Devices page at this point, you may notice that the Connection Status is some non-zero value. Status zero means no problem has been detected. A non-zero code means there is a problem with the connection.

Data Objects	Modk	us	BACnet	SNMP		System		
Modbus RT	UData	Modbus RTU :	Setup 🚺 Modbu	us TCP Data 🔰	Modbus TCP 9	Setup		
Devices	Client	t Read Map	Client Write Ma	ap Server I	Map	<u></u>		
This page sets up the via the client read and	network add client write	dress and optional maps. The local d	device parameters evice acts as a Moo	for a remote Modb Ibus master to the	us/TCP device t remote devices	hat will be link listed below.	ed to local o	bjects
Device #	1				(	Update	< Prev	Vext>
IP Address	192.168.1.1	10 Port 502	2 Local Name:	Test Device				
Unit (optional)	0	🔲 Use FC 5/6	instead of 15/16	🔲 Use FC 5/6	and 15/16 by co	ount		
	Low reg	ister is first for m	ultiple registers		Connection S	tatus		
Default Poll Period	2.0	Seconds			5			

Connection status will show a non-zero error code if there is a socket error. The most common errors include:

5 = Connection failed, unable to bind (usually means remote device not connected or not reachable)

- 81 = Connection in progress (means unsuccessful connect attempt, still trying)
- 95 = Network is unreachable
- 97 = Connection aborted
- 98 = Connection reset by peer
- 103 = Connection timed out
- 104 = Connection refused
- 107 = Host is unreachable



## **16. WiFi Sensor Setup - PointSix, AirTest**

The BB2-7010-06 is an enhancement to the standard BB2-7010-02 which adds recognition of PointSix and AirTest WiFi sensors that transmit data to a UDP port which the user may configure. The sensor packets utilize the PointSix data packet format. Refer to PointSix documentation for setup of the sensor itself. The BB2-7010-06 will make the PointSix sensor data available as BACnet objects (and Modbus registers or SNMP OID's).



A standard generic WiFi access point is used to connect the WiFi sensors to the wired network. The access point should be placed where it has access to the wired network, but also where it will have good reception of the WiFi sensor signals. Typically, the BB2-7010-06 gateway will be located inside a control panel, but the WiFi access point will be located outside the control panel and potentially some distance from the panel.

#### 16.1 Sensor Network Settings

The WiFi sensors need to initially be configured using the sensor configuration utility. One of the required settings is the IP address of the server to which the sensor should send data. That IP address will be the address of the BB2-7010-06 gateway which is set on the Local Host page shown here. If NAT routers are placed between the sensors and the gateway, then use IP addresses provided by the IT technician that set up the router.
Data Objects	Modbus	BACnet	SNMP	System						
Setup	Sensors		Ì							
Config File	BACnet IP Port	BBMD	Local Host	User						
This page allows you to ch	This page allows you to change this device's IP address, and select whether double registers are swapped when returned to a remote client accessing this server.									
accessing uns server.										
IP Address	192.168.1.148	192.168.1.148	- Refresh -							
Subnet Mask	255.255.255.0	255.255.255.0	Change IP							
Gateway	192.168.1.1	192.168.1.1								
MAC Address	:00:40:9D:39:C1:A6									
HTTP Port	t 80 (default	80)	Set Port	]						
Uptime	1,17:19:37			1						
UDP Sensor Port	t b/b/ (default	: 6767)	Set UDP	J						
UPD Port Status	; 0									

The sensors send data to UDP port 6767 by default. To change that, go to the Network configuration page where the gateway's IP address is set, and select a different UDP port if desired.

# 16.2 Sensor Configuration

The screen illustrated below is used to set up recognition of the WiFi sensor packets.

1001 Babel Buster 2 1001 Bacnet-modeus 1001 Network Gateway Model BB2-7010-06 000001										
Data O	bjects	Modbus	BAC	net	SNMP		System			
	Setup	Sensor	s					Ì		
Senso	r Data	Sensor Setup				ſ				
WiFi sen	sors that will be rec	ognized by the s	ystem are enter	ed here.						
			Showin	g 1 to 1	5of 200	U	Ipdate <p< th=""><th>rev Next &gt;</th></p<>	rev Next >		
Sensor	Sensor S/N	Timeout (minutes)	Chan 1 Object #	Chan 1 Scale	Chan 1 Offset	Chan 2 Object #	Chan 2 Scale	Chan 2 Offset		
1	E1A3D018	15	Al 1	1.80000	32.0000	AI 2	1.80000	32.0000		
2	E22B4008	15	AI 3	0.00000	0.00000	AI 4	1.80000	32.0000		
3	E2044005	15	AI 5	1.80000	32.0000	0	0.00000	0.00000		
<u>4</u>		0	0	0.00000	0.00000	0	0.00000	0.00000		

Enter the 8-character hexadecimal serial number of the sensor packet whose data is to be received. Select local BACnet objects where data from channels 1 and 2 of the sensor should be stored. BACnet objects should be either Analog Input or Analog Value. Do not use any other object type. Scaling of raw data to

engineering units will be performed automatically based on the enumeration code returned by the sensor. The scale and offset values shown here are applied on top of the scaling native to the sensor. For example, sensors returning temperature will always return degrees Celsius. To convert to Fahrenheit instead, apply scale and offset as illustrated above.

The timeout, in minutes, is optional. If zero, the timeout will not be monitored. If non-zero, and the time since last transmission exceeds this time in minutes, then the BACnet objects assigned to this sensor will reflect a Fault status.

The sensors have high/low alarm capability. If the alarms are not being used, then the only setup needed is on the tabular list illustrated above. To set up alarms, click on the sensor number in the first column. This will take you to the extended alarm setup page illustrated below.

Data Objects	Modbus	BACn	BACnet SNMP			System	l	
Setup	Sensors	$\neg$						
Sensor Data	Sensor Setup	$\sum$						
WiFi sensors that will be r	ecognized by the system	are entered	d here.					
Sensor # 1						Update	<pre> I</pre>	Next >
Device	Serial Number E1	A3D018	Trans	mit Period	300			
	Timeout 15			Hysteresis	5			
Channel 1 Data	Data Object 🗛	1		Scale	1.80000			
				Offset	32.0000			
Channel 1 Alarms	Hig	h Alarm			Low Alarm			
	Setpoint Object 🛛		Setpo	oint Object	0			
	Setpoint Value 85.	.0000	Setp	oint Value	65.0000			
	Alarm Time 1		A	larm Time	1			
	Status Object Bl	1	Sta	tus Object	BI 2			
Channel 2 Data	Data Object 🗛	2		Scale	1.80000			
				Offset	32.0000			
Channel 2 Alarms	Hig	h Alarm			Low Alarm			
	Setpoint Object AC	01	Setpo	oint Object	AO 2			
	Setpoint Value 0.0	0000	Setp	oint Value	0.00000			
	Alarm Time 1		A	larm Time	1			
	Status Object Bl	3	Sta	tus Object	BI 4			

The same sensor setup information found on the tabular list is also available here, in addition to alarm settings. Enabling an alarm requires setting the Alarm Time to a non-zero value. User scaling entered here is applied to alarm setpoints, as well as additional internal scaling based on sensor type and its enumeration code. Therefore, the example here will display temperatures in Fahrenheit, and alarm setpoints will also then be entered as Fahrenheit.

The sensor Transmit Period is entered in seconds. The Hysteresis is entered as a raw count. This value is not scaled since the same hysteresis is applied to both sensor channels, which may have different units. The raw count is A/D count at the sensor. As an example, a single 'count' for an RTD sensor is 0.0977

degrees C. Refer to the sensor data sheet for actual scaling information. If transmit period and hysteresis are left as zero on this page, the gateway will send back whatever settings it got from the sensor for those parameters, which would have been initially set using the sensor configuration utility (independent of anything the BB2-7010-06 does).

Analog Input is the most suitable BACnet object for recording data values read by sensors. Analog Value would also be an appropriate choice, although with no particular advantage over Analog Input. Analog Output is not a suitable choice for sensor values since these objects are meant to be written to BB2-7010, not read from. Analog Input is typically a sensor object, while Analog Output is typically an actuator object.

Sensor alarm setpoints can be fixed values that are stored in the XML configuration file, or setpoints can be provided dynamically by an external system. Sensor alarm setpoints can be taken from an Analog Value object or Analog Output object. You would not use an Analog Input object for alarm setpoints since you cannot write to an Analog Input. The notation "Input" means input from sensor hardware, or input to the network. The notation "Output" means output to sensor hardware, as would be applicable for an alarm setpoint. The Analog Value object can be used without any requirement for command priority; however, at startup, the Analog Value present value properties in the BB2-7010 will all default to zero. If the BACnet client (typically a facility management system) is not quick enough to write setpoints to the BB2-7010, alarms will be cleared in the sensor the first time it reports in, and only restored at some later time when the client gets around to updating the setpoints. If you want the BB2-7010 to use known default alarm setpoint values at startup, use Analog Output objects for alarm setpoints, and enter the default alarm setting as the "Relinquish Default" value.

If an alarm setpoint object is given, it will be used, and will override any setpoint value entered on this page. If the setpoint object is left set to zero, then the setpoint value shown on this page will be used. All parameters on this page are stored in the XML configuration file when you click Save on the Config File page. (Forgetting to save the file will result in settings being lost the next time power is lost.) Whether a setpoint object or setpoint value is provided, in either case, the Alarm Time must be non-zero to enable the alarm.

The "alarm time" is the amount of time in sensor update periods (transmit periods) that the condition must exist before being reported as an alarm. Alarm time of zero is disabled (on this setup page). A zero will be transmitted as FF to the sensor, and all alarm time values offset by -1 when sent. This means alarm time of zero disables the alarm, alarm time of one enables the alarm with immediate reporting, and alarm time of two will delay the report by one sensor update period (shown as 300 seconds on the example above).

Alarms are not particularly useful if the alarm state is not reported. The means for reporting alarms states is via the status objects provided on the setup page. The most suitable object type is Binary Input, although any Input or Value object could be used. The alarm state value will always be either zero or one, and therefore Binary is a suitable type. Binary objects also have the option of associating user defined text strings with the active and inactive states. Changes in alarm state can be reported autonomously with BACnet COV subscriptions or SNMP traps. The alarm states can also be queried via Modbus.

### 16.3 Sensor Data Page

The screen illustrated below shows sensor data and additional diagnostic information. This page shows sensor data as received most recently. The data received will be displayed here regardless of whether the data has been assigned to any BACnet objects.

000 1100 1011 0100 0100	10011 Babel Buster 2 DILO Bachet-Modbus 1001 Network Gateway Model BB2-7010-06 MINNESOTA										
Data	Objects	Modb	us	BACnet	SNMP		System				
	Setup		Sensors					J			
Sens	or Data	Senso	r Setup								
This pa	ge displays data r	nost recer	ntly returned by WiFi :	sensors.							
WiFi Ser	isor List		Show	ing sensors from	1		Refresh	<prev next=""></prev>			
Sensor	Sensor S/N	Туре	IP	Chan 1 Data	Chan 2 Data	Life	Tx Count	Time since Tx			
1	E1A3D018	76	192:168:1:202	76.653847	75,774551	538d	2628	3.57m			
2	E1A3D024	76	192:168:1:203	1341.000000	69.285606	963d	19031	8.34m			
з	E2044005	29	192:168:1:204	2.087872	0.000000	289d	74319	2.67m			
4		00	0:0:0:0	0.000000	0.000000	DO	0				

The sensor type as reported by the sensor at the serial number shown will be displayed along with the IP address from which it was received. Scaled data is displayed, and this data should be echoed in the applicable BACnet object. The calculated remaining battery life based on data returned by the sensor is displayed. The total number of transmissions is indicated as well as time elapsed since the last transmission was received. Times are in seconds (s), minutes (m), hours (h) or days (d).



# **17. WiFi Sensor Setup - Veris Industries**

The BB2-7010-07 is an enhancement to the standard BB2-7010-02 which adds recognition of Veris Industries WiFi sensors that transmit data to a UDP or TCP port which the user may configure. The sensor packets utilize the Veris data packet format. Refer to Veris Industries documentation for setup of the sensor itself. The BB2-7010-07 will make the Veris sensor data available as BACnet objects (and Modbus registers or SNMP OID's).



A standard generic WiFi access point is used to connect the WiFi sensors to the wired network. The access point should be placed where it has access to the wired network, but also where it will have good reception of the WiFi sensor signals. Typically, the BB2-7010-07 gateway will be located inside a control panel, but the WiFi access point will be located outside the control panel and potentially some distance from the panel.

### 17.1 Sensor Network Settings

The WiFi sensors need to initially be configured using the sensor configuration utility. One of the required settings is the IP address of the server to which the sensor should send data. That IP address will be the address of the BB2-7010-07 gateway which is set on the Local Host page shown here. If NAT routers are placed between the sensors and the gateway, then use IP addresses provided by the IT technician that set up the router.



The sensors send data to UDP and TCP ports 6788 by default. To change that, go to the Network configuration page where the gateway's IP address is set, and select a different port if desired.

Port status is also shown on the network setup (Local Host) page. These should normally all be zero since they are error counts and error codes. Port status of zero means no error.

#### 17.2 Sensor Configuration

The screen illustrated below is used to set up recognition of the WiFi sensor packets.

1001 Babel Buster 2 DILO BACNET-MODBUS 1001 NETWORK GATEWAY MODEL BB2-7010-07 MINNESOTA										
Data (	)bjects	Modbus	BAC	net	SNMP		System			
	Setup	Senso	rs		Ì			l		
Senso	ır Data	Sensor Setup				ſ				
WiFi ser	nsors that will be reco	gnized by the s	system are entere	ed here.						
			Showing	g 1 to 1	5of 200		Jpdate < P	rev Next>		
Sensor	Sensor MAC	Timeout (minutes)		Chan 1 Object #	Chan 2 Object #	Chan 3 Object #	Chan 4 Object #			
1	74D850010001	2		AI 1	AI 2	AI 3	0			
2	74D850010002	2		AI 4	AI 5	0	0			
<u>3</u>	74D850010003	2		AI 6	AI 7	0	0			
4	00000000000	0		0	0	0	0			

Enter the 8-character hexadecimal MAC address of the sensor whose data is to be received. Select local BACnet objects where data from channels 1 through 4 of the sensor should be stored. BACnet objects should be either Analog Input or Analog Value. Do not use any other object type.

Enter a sensor timeout in minutes. If the BB2-7010-07 does not receive data from this sensor within this amount of time, the BACnet object's reliability property will be set to the non-zero error code indicating timeout. In addition, the optional timeout fault value will be placed into the object's present value property if selected on the extended setup page.

To access aditional setup parameters, click on the sensor number in the first column. This will take you to the extended setup page illustrated below.

Data Objects	Modbus	BACnet	SNMP		System	
Setup	Sensors		ļ			Ì
Sensor Data	Sensor Setup				$\int$	
WiFi sensors that will be r	ecognized by the syste	m are entered he	re.			
Sensor # 1					Update < Prev	Next >
Device	MAC Address 7	4D850010001	Timeout	2		
Channel 1 Data	Data Object 🗚	Al 1	Scale	1.80000		
	Timeout Fault 🔲 🛛	0.0000	Offset	32.0000		
Channel 2 Data	Data Object 🛛	AI 2	Scale	0.00000		
	Timeout Fault 🔲 🛛	0.0000	Offset	0.00000		
Channel 3 Data	Data Object 🛛	AI 3	Scale	0.00000		
	Timeout Fault 🔲 🛛	0.00000	Offset	0.00000		
Channel 4 Data	Data Object ()	)	Scale	0.00000		
	Timeout Fault 🔲 🛛	0.00000	Offset	0.00000		
Battery Voltage	Data Object ()	)				
RSSI	Data Object 🛛	)				

The same sensor setup information found on the tabular list is also available here, in addition to user scaling and fault values. The example shown here for AI 1 will display temperatures in Fahrenheit when actual data received was Celsuis temperature.

Analog Input is the most suitable BACnet object for recording data values read by sensors. Analog Value would also be an appropriate choice, although with no particular advantage over Analog Input. Analog Output is not a suitable choice for sensor values since these objects are meant to be written to BB2-7010, not read from. Analog Input is typically a sensor object, while Analog Output is typically an actuator object. Any data object set to zero will not have that data transferred to any BACnet object.

Timeout fault is optional. If the box is checked, and a value is entered in the adjacent window, that value will be placed into the data channel's object when the sensor is in fault, i.e., when the BB2-7010 has not received data from this sensor within the timeout window specified.

Battery voltage and signal strength readings reported by the sensor can optionally be provided as BACnet objects if provided. Analog Input or Analog Value object types are most suitable here.

### 17.3 Sensor Data Page

The screen illustrated below shows sensor data as received most recently. The data received will be displayed here regardless of whether the data has been assigned to any BACnet objects. This page also displays time elapsed since the last transmission was received. Times are in seconds (s), minutes (m), hours (h) or days (d).

1001 Babel Buster 2 DILO BACNET-MODBUS 1001 NETWORK GATEWAY MODEL BB2-7010-07 MINNESOTA										
Data O	bjects M	lodbus	BACnet		SNMP	System	·			
	Setup	Se	nsors				Ì			
Senso	r Data S	iensor Seti	ab 🔰	J						
This pag	e displays data most	recently re	turned by WiFi sensors							
WiFi Sens	or List		Showing sen	isors from 1		Refresh	<pre></pre>			
Sensor	Sensor MAC	Туре	Chan 1 Data	Chan 2 Data	Chan 3 Data	Chan 4 Data	Time since Tx			
1	74D850010001	0A,07	71.599998	40.000000	400.000000	0.000000	7.2s			
2	74D850010002	07,03	71.599998	40.000000	0.000000	0.000000	6.2s			
<u>3</u>	74D850010003	03,08	59.00000	0.000000	0.000000	0.000000	5.2s			
4	000000000000	00,00	0.00000	0.000000	0.000000	0.00000				

Click on the sensor number in the first column in the Sensor Data page to access additional sensor information illustrated below. This page reflects whatever information was transmitted by the respective sensor.

Data Objects	Modbus	BACnet	SNMI	Р	System	
Setup	Sensors		)			l
Sensor Data	Sensor Setup					
WiFi sensor statistical dat	a is displayed on this	page.				
Sensor # 1					Update	< Prev Next >
	MAC Address:	74D850010001	Type,Sub	type: 0A,07		
	Number of Sensors:	4	Software Ve	rsion: 2.01		
	Uptime:	0				
	Listen mode times:	224	Reconnect t	imes: 0		
	Packets sent:	140				
	Battery Voltage:	0.00		RSSI: -50.00		
	Tag error:	0	Conversion	error: 0		
	Sensor type error:	0				



# **18. WiFi Sensor Setup - Functional Devices RIB**

The BB2-7010-08 is an enhancement to the standard BB2-7010-02 which adds support for Functional Devices WiFi RIB sensors and relays. The data packets utilize the HTML/XML data format documented in the RIB user manual. Refer to Functional Devices documentation for setup of the sensor/relay itself. The BB2-7010-08 will make the RIB sensor data available as BACnet objects (and Modbus registers or SNMP OID's). The BB2-7010-08 will also allow control of the RIB relay from BACnet, Modbus, or SNMP networks.



A standard generic WiFi access point is used to connect the WiFi sensors to the wired network. The access point should be placed where it has access to the wired network, but also where it will have good reception of the WiFi sensor signals. Typically, the BB2-7010-08 gateway will be located inside a control panel, but the WiFi access point will be located outside the control panel and potentially some distance from the panel.

#### 18.1 Sensor Network Settings

The WiFi sensors need to initially be configured using the sensor configuration utility. One of the required settings is the IP address of the server to which the sensor should send data. That IP address will be the address of the BB2-7010-08 gateway which is set on the Local Host page shown here. If NAT routers are placed between the sensors and the gateway, then use IP addresses provided by the IT technician that set up the router.

Data Objects	Modbus	BACnet	SNMP	System	
Setup	Sensors	Ì	Ì		
Config File	BACnet IP Port	BBMD	Local Host	User	
This page allows you to ch accessing this server.	ange this device's IP	address, and select whether	r double registers are swapped	d when returned to a remote o	lient:
IP Address	192.168.1.33	192.168.1.33	- Refresh		
Subnet Mask	255.255.255.0	255.255.255.0	Change IP		
Gateway	192.168.1.1	192.168.1.1			
MAC Address	:00:40:9D:89:66:F6				
HTTP Port	: 80 (default	: 80)	Set Port		
Uptime	0,01:33:50				

The sensors send data to UDP port 6767 by default. To change that, go to the Network configuration page where the gateway's IP address is set, and select a different UDP port if desired.

### 18.2 Sensor Configuration

The screen illustrated below is used to set up recognition of the WiFi sensor packets.

0001 1100 1011 0100 0100	BACNET-MC NETWORK G MODEL BB2-70	obbus ATEWAY DIO-08			Contr	ol-Solu	TIONS, In Minnes	NC. DTA
Data	Objects	Modbus	BAC	net	SNMP		System	
	Setup	Sens	ors					l
Senso	or Data	Sensor Setup			$\mathbf{V}$	ſ		
WiFi se	nsors that will be reco	ognized by the	system are enter	ed here.				
			Showin	g 1 to 100	of 10		lpdate < P	rev Next>
Sensor	Sensor IP Addr	Poll Time (seconds)		DO 1 Object #	DI 1 Object #	UI 1 Object #	UI 2 Object #	
1	192.168.1.206	10		BO 1	BI 1	AI 1	AI 2	
2	192.168.1.207	10		BO 2	BI 2	AI 3	AI 4	
3	0.0.0.0	0		0	0	0	0	

Enter the IP address of the sensor that is to be queried. Select local BACnet objects to be associated with the RIB inputs and output. BACnet object type should be Binary Output for DO 1 (the relay). BACnet object type should be Binary Input for DI 1 (discrete input). BACnet object types can be Analog Input or Analog Value for the UI inputs if configured for analog values. BACnet object types can be Binary Input or Binary Value if UI inputs are configured as state inputs instead.

Enter a sensor poll time in seconds. The is the frequency at which the BB2-7010-08 will attempt to query the RIB. If the BB2-7010-08 is unable to connect to the RIB, the BACnet object's reliability property will be set to the non-zero error code indicating no connection. In addition, the optional connect fault value will be placed into the object's present value property if selected on the extended setup page.

To access aditional setup parameters, click on the sensor number in the first column. This will take you to the extended setup page illustrated below.

Data Objects	Modbus	BACnet	SNMP	System	
Setup	Sensors		Ì		Ĵ
Sensor Data	Sensor Setup				]
WiFi sensors that will be r	ecognized by the syster	n are entered here.			
Sensor # 1				Update < Prev	Next>
Device	IP Address 19	92.168.1.206	Poll Time (sec) 10		
	User Name				
	Password				
DO 1 Data	Data Object B	01			
DI 1 Data	Data Object B	11	Link Object BO 2	2	
	Connect Fault 🔲 0.	.00000			
UI 1 Data	Data Object A	11	Scale 0.000	000	
	Connect Fault 🔲 0.	.00000	Offset 0.000	000	
UI 2 Data	Data Object A	12	Scale 0.000	000	
	Connect Fault 🔲 0.	.00000	Offset 0.000	000	
Signal Strength	Data Object 🛛				

The same sensor setup information found on the tabular list is also available here, in addition to user scaling and fault values. Scale factors may be optionally applied to the analog sensor values. Data received from the sensor will be multiplied by the scale factor, then added to offset, before being placed into BACnet data objects.

User name and password should only be entered here if the RIB has been configured to use a password. If the RIB has been configured to require a password and it is not entered here, then the connection will fail.

The Link Object for the DI input is optional. The RIB allows the DI to be "bound" to the DO within the RIB. The Link Object here allows a DI from one RIB to be "bound" to the DO in a different RIB with the binding provided by the BB2-7010-08. Response time will be the poll time, meaning a change on a DI in one RIB will be propagated to the the linked DO in a different RIB the next time that sensor is polled. The Link Object should be a Binary Output object, and should be the same BO assigned to the DO in a different RIB. (Note: If the DO is bound to the DI via configuration within the same RIB, do not enter that binding here as the RIB will manage the binding internally in that case.)

Connect fault is optional. If the box is checked, and a value is entered in the adjacent window, that value will be placed into the data channel's object when the sensor is in fault, i.e., when the BB2-7010 was not able to connect to this sensor. If connection is successful on the next try, then the data received will be

placed into the BACnet data object and the reliability code will be cleared to zero.

Signal strength readings reported by the sensor can optionally be provided as aBACnet object if provided. Analog Input or Analog Value object types are most suitable here.

#### 18.3 Sensor Data Page

The screen illustrated below shows sensor data as received most recently. The data received will be displayed here regardless of whether the data has been assigned to any BACnet objects. This page also displays time elapsed since the last transmission was received. Times are in seconds (s), minutes (m), hours (h) or days (d).

1001 1001 1001 Bachel Buster 2 1011 Bachel Buster 2 1001 Network Gateway MODEL BB2-7010-08 1000 BB2-7010-08										
Data O	)bjects Modb	us BAC	inet	SNMP	Syste	m				
	Setup	Sensors								
Senso	r Data Senso	r Setup								
This pag WiFi Sens	ge displays data most rece sor List	ntly returned by WiFi sens Showing	sors. sensors from 1		Refresh	<pre>  Next&gt;</pre>				
Sensor	Sensor IP Addr	DO 1 State	DI 1 State	UI 1 Data	UI 2 Data	Time since update				
1	192.168.1.206	Active	Inactive	2.460000	4.730000	6.3s				
2	192.168.1.207	Inactive	Active	1.120000	3.310000	5.9s				
<u>3</u>	0.0.0.0									
<u>4</u>	0.0.0.0									
5	0.0.0.0									
<u>6</u>	0.0.0.0									
<u>Z</u>	0.0.0.0									
<u>8</u>	0.0.0.0									
2	0.0.0.0									
<u>10</u>	0.0.0.0									
Sensor pa	acket diagnostic page									

Click on the sensor number in the first column in the Sensor Data page to access additional sensor information illustrated below. This page reflects whatever information was transmitted by the respective sensor. The Reset button clears the error indication and may be used to see if the next retry returns the same error. The error code will automatically clear to zero (upon refresh of the page) if the connection was successfu.

Data Objects	Modbus	BACnet	SNMP	System		
Setup	Sensors		Ţ		J	
Sensor Data	Sensor Setup	<u></u>				
WiFi sensor additional information is displayed on this page.						
Sensor # 1				Update < Pre	v Next>	
	IP Address: 192.:	168.1.206				
	Device Version: v4.0.	9.1.0 WIUI	Software Version: 12556			
	Device Name: RIB B	BOX 1				
	Device Location: Lab					
	SSID: WiFis	Sensor				
	Signal Strength: 0					
	Connection error: 0/0					
	Re	eset				

### 18.4 Sensor Diagnostic Page

Click on the sensor diagnostic page link at the bottom of the sensor data page to access the page illustrated below. Enter a sensor number and click Update to select a sensor. Thereafter, each packet received in response to query will be displayed when the page is refreshed. This page is provided simply for diagnostic use.



WiFi Sensor Packet Log for Sensor #1

```
<relayState>ON</relayState>
<lastCommand>ON</lastCommand>
<digtalInputState>OPEN</digtalInputState>
<uilState>0-5V</uilState>
<ui1Value> 0.00 V</ui1Value>
<di1Value>OPEN</di1Value>
<ui2State>0-5V</ui2State>
<ui2Value> 0.00 V</ui2Value>
<di2Value>OPEN</di2Value>
<powerOnState>OFF</powerOnState>
<bindingState>UNBOUND</bindingState>
<deviceHostName>RIB BOX 1 </deviceHostName>
<deviceLocation>Lab</deviceLocation>
<deviceVersion>v4.0.9.1.0 WIUI</deviceVersion>
<bss>
<valid>0</valid>
<name>0</name>
<privacy>0</privacy>
<wlan>0</wlan>
<strength>0</strength>
</bss>
<scan>0</scan>
<ver>12556</ver>
<count>0</count>
<ssid>WiFiSensor</ssid>
```

# Appendix A Hardware Details

#### A.1 Wiring



Wire the BB2-7010 as illustrated above. Follow all conventional standards for wiring of EIA-485 networks when connecting the Modbus RTU EIA-485 (RS485) network. This includes use and termination of shield, termination of the network, and grounding.

IMPORTANT: Although EIA-485 (RS485) is thought of as a 2-wire network, you MUST include a third conductor connected to GND or common at each device so that all devices are operating at close to the same ground potential. Proper grounding of equipment should ensure proper operation without the third conductor; however, proper grounding often cannot be relied upon. If large common mode voltages are present, you may even need to insert optically isolated repeaters between EIA-485 devices.

Use standard CAT5 cables for Ethernet connections. Use control wire as applicable for local electrical codes for connecting the 24V (AC or DC) power supply.

Note that in addition to connecting power supply common to a GND terminal, you must also connect a GND terminal to earth ground in order to ensure proper ESD protection.

**BB2-7010-xx-232**: The standard BB2-7010 Modbus RTU port uses RS-485. The RS-232 version replaces the RS-485 transceiver with an RS-232 transceiver. The NET+/NET- terminals are replaced by TXD and RXD on the RS-232 version. TXD is data out from the BB2-7010, and RXD is data in to the gateway. Hardware handshake is not supported.

# A.2 Front Panel LED Indicators

Power-up LED behavior: All LEDs on front panel will turn on yellow or red for half a second, then all will turn on green for half a second. Then they will proceed to indicate as normally defined for the indicators.



The LED indicators behave as follows. Normally the reply/request LEDs reflect Modbus RTU activity. If the gateway is being used only for Modbus TCP (as SNMP gateway), then the same LEDs reflect TCP bahavior instead. The LEDs do not attempt to reflect both RTU and TCP activity since that would get confusing. If both RTU and TCP are used, the request/reply LEDs reflect RTU traffic while the Ethernet activity LED will indicate TCP traffic. To see TCP errors, one needs to look at the Errors page in the web UI.

DEV DATA	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.
DEV STATUS	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).

### A.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 white block next to the 3-pin header. Putting the shunt on the opposite 2 pins disables the option, and is simply a place to store the shunt.



The "Init" jumper on the server module should only be used when advised by tech support. Installing this jumper prior to power-up causes the server to go into firmware update mode.