

Accessing BACnet Objects as Any Modbus Register You Like

All types of BACnet objects in a BB2-7010 are accessed as Modbus holding registers when the gateway is a Modbus slave. The holding register numbers are documented in the user guide in section 8.2 for Modbus RTU or section 9.2 for Modbus TCP (register mapping is same for both RTU and TCP). However, you do have the option of assigning any Modbus register number you like and any Modbus register type you like to any BACnet object by using the Server Map.

We will look at three different ways to look at Binary Outputs in the following examples, but the same principle applies to all other object types. By default, without using the server map, Binary Outputs will show up at holding registers 4001, 4002, etc., as illustrated below.



This page displays data as presently found in the local registers maintained by this device.

Binary Output Objects Showing objects from 1

Object	Object Name Object Description	Out of Service	Present Value	Reliability	Status	Text
1	My Binary Output 1 Description of BO 1	N	Active	0	0,0,0,0	BO 1 Active
2	My Binary Output 2 Description of BI 2	N	Inactive	0	0,0,0,0	BI 2 Inactive
3	My Binary Output 3 Description of BI 3	N	Inactive	0	0,0,0,0	BI 3 Inactive

Now let's say we want to read the Binary Outputs as Modbus coils. To do that, we start by going to the Server Map. This page is found under Modbus TCP settings, but applies to Modbus RTU as well. To enable mapping using Modicon register numbering (required in order to access objects as coils), check both of the boxes illustrated at the bottom of this page. Then simply enter the Modicon address and assign the BACnet objects as illustrated below.



Data Objects
Modbus
BACnet
System Setup

Modbus RTU Data
Modbus RTU Setup
Modbus TCP Data
Modbus TCP Setup

Devices
Client Read Map
Client Write Map
Server Map

Create remote client's custom view of local registers. This page sets up the register map for the virtual Modbus/TCP server. This map is also referred to as the "user map". This allows you to remap the default server register map to match any layout you wish, including matching the map found in other equipment.

Showing 1 to 4 of 4 Update < Prev Next >

Map #	Mapped Register #	Mapped Register Format	Local Object #	Scale Factor	Offset	Bit Field	Fill	Name
1	1	Unsigned Integer ▾	BO 1	0.000000	0.000000	0000	0000	My Binary Output 1
2	2	Unsigned Integer ▾	BO 2	0.000000	0.000000	0000	0000	My Binary Output 2
3	3	Unsigned Integer ▾	BO 3	0.000000	0.000000	0000	0000	My Binary Output 3
4	0	None ▾	0	0.000000	0.000000	0000	0000	---

User Map Enabled
 Map is Exclusive
 Swap Double Registers
 Zero fill null registers
 Use Modicon mapping

Custom Registers Enabled: 4 Insert Delete

We are now reading the same Binary Output objects as Modbus coils, as illustrated below.



This page displays data as presently found in the local registers maintained by this device.

Binary Output Objects Showing objects from

Object	Object Name Object Description	Out of Service	Present Value	Reliability	Status	Text
1	My Binary Output 1 Description of BO 1	N	Active	0	0,0,0,0	BO 1 Active
2	My Binary Output 2 Description of BI 2	N	Inactive	0	0,0,0,0	BI 2 Inactive
3	My Binary Output 3 Description of BI 3	N	Inactive	0	0,0,0,0	BI 3 Inactive

ge. Click Prev/Next

he Device Link. The

at value. The
value will be written to

If we wanted to read the Binary Outputs as Modbus status inputs (discrete inputs) instead, we would enter the Modicon register numbers for status inputs instead, as illustrated below.



Data Objects
Modbus
BACnet
System Setup

Modbus RTU Data
Modbus RTU Setup
Modbus TCP Data
Modbus TCP Setup

Devices
Client Read Map
Client Write Map
Server Map

Create remote client's custom view of local registers. This page sets up the register map for the virtual Modbus/TCP server. This map is also referred to as the "user map". This allows you to remap the default server register map to match any layout you wish, including matching the map found in other equipment.

Showing to 4 of 4 Update < Prev Next >

Map #	Mapped Register #	Mapped Register Format	Local Object #	Scale Factor	Offset	Bit Field	Fill	Name
1	10001	Unsigned Integer ▾	BO 1	0.000000	0.000000	0000	0000	My Binary Output 1
2	10002	Unsigned Integer ▾	BO 2	0.000000	0.000000	0000	0000	My Binary Output 2
3	10003	Unsigned Integer ▾	BO 3	0.000000	0.000000	0000	0000	My Binary Output 3
4	0	None ▾	0	0.000000	0.000000	0000	0000	---

Custom Registers Enabled:
 User Map Enabled Map is Exclusive
 Swap Double Registers Zero fill null registers
 Use Modicon mapping

Insert Delete

Although potentially not useful since you cannot write to a Modbus status input, we illustrate reading the same Binary Outputs as status inputs below.



This page displays data as presently found in the local registers maintained by this device.

Binary Output Objects Showing objects from

Object	Object Name Object Description	Out of Service	Present Value	Reliability	Status	Text
<u>1</u>	My Binary Output 1 Description of BO 1	N	Active	0	0,0,0,0	BO 1 Active
<u>2</u>	My Binary Output 2 Description of BI 2	N	Inactive	0	0,0,0,0	BI 2 Inactive
<u>3</u>	My Binary Output 3 Description of BI 3	N	Inactive	0	0,0,0,0	BI 3 Inactive

ModScan32 - ModSca1

File Connection Setup View Window Help

Address:
 Device Id:
 Number of Polls:
 Valid Slave Responses:

Length:
 MODBUS Point Type:

10001: <1>
 10002: <0>
 10003: <0>

For Help, press F1 Polls: 5 Resps: 5

ge. Click Prev/Next

he Device Link. The

at value. The

value will be written to

You can mix any number of Modbus register types and register numbers using Modicon notation, and can assign the Modbus registers to any of the BACnet objects in the gateway. Be sure to read the Quick Help section at the bottom of the Server Map page for more information about setting up the various parameters for each Server Map entry.

Article ID: 39

Created On: Tue, Jul 25, 2017 at 9:50 PM

Last Updated On: Tue, Jul 25, 2017 at 9:50 PM