APPLICATION NOTE

App. Note Code: 1M-C

BACnet to Modbus / Modbus to BACnet Protocol Conversion

CAMPBELL SCIENTIFIC®
WHEN MEASUREMENTS MATTER

BACnet to Modbus Protocol Conversion

This application note describes using Alerton's BCM-MDBS with Campbell Scientific's PakBus dataloggers to convert protocol between Modbus and BACnet. Information in the note refers to wiring between the datalogger and the BCM-MDBS, programming the datalogger as the slave, and programming the BCM-MDBS as the master.

1. General Information

Alerton equipment is typically used to provide automated temperature control of HVAC systems in large buildings. The protocol used by Alerton for BACnet communication is BACTalk. The building control products they produce are divided into two categories: field controllers and building controllers. Building controllers (BCMs) are the top end of the system and communicate directly with Envision software from Alerton. Field controllers (VLCs) communicate with Building controllers and directly control devices in the HVAC system. There are a variety of BCMs that have different functions. The BCMs required for protocol conversion are a BCM-PWS, BCM-ETH, and BCM-MDBS (see Figure 1).

Figure 1. The BCM-PWS, BCM-ETH, and BCM-MDBS modules.

1.1 BCM-PWS

The BC-PWS provides power to the other modules.

1.2 BCM-ETH

This module allows Alerton's Envision software to communicate with the BCM-MDBS module.

1.3 BCM-MDBS

This module is the converter between BACnet and Modbus protocols. Conceptually, it is a bank of registers that are accessible by both Modbus equipment and BACnet equipment, allowing data to be passed between the two devices. The BCM-MDBS is the Modbus master. It uses descriptions of analog values and boolean values to formulate Modbus RTU commands used to read and write the Modbus registers of the datalogger. In round robin fashion, it will repeatedly update register blocks. The module consists of 1000 analog values (AVs) and 1000 boolean values (BVs).

Figure 2. To wire the BCM-PWS (left), BCM-ETH (center), and BCM-MDBS (right), lift their hinged lids. Wire them in series.

2. Wiring

Lift the lids on the BCM-PWS, BCM-ETH, and BCM-MDBS modules and wire them in series (see Figure 2).

815 W. 1800 N., Logan, UT 84321-1784 (435) 753-2342

The datalogger connects to the BCM-MDBS via an SC110 cable (PN 17856). Connect the pigtail wires of the SC110 to a COM port on the datalogger (see Figure 3). The datalogger COM port must match the COM port specified in the datalogger program.

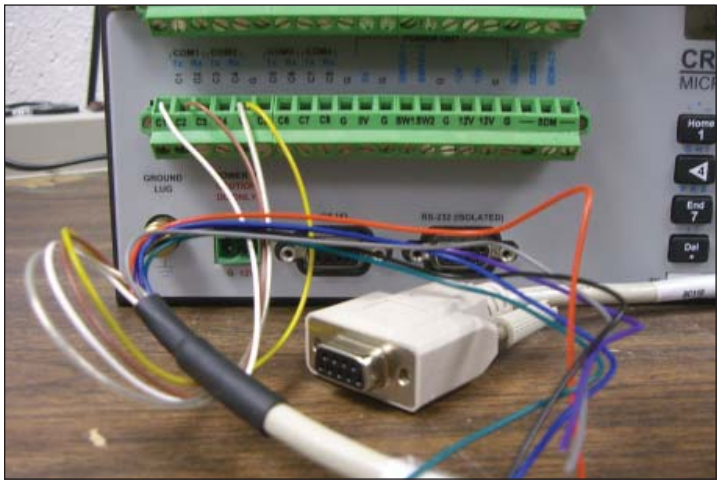

Figure 3: The SC110 cable is wired to COM1 on a CR3000 Micrologger. The datalogger COM port must match the COM port designated in the datalogger program.

Attach the SC110's DB9 female connector to the 9-pin port on the back of the BCM-MDBS (see Figure 4). The Ethernet cable connects to the back of the BCM-ETH as shown in Figure 4.

Figure 4: At bottom left, the SC110's DB9 female connector is attached to the BCM-MDBS. At center, a yellow Ethernet cable is connected to the Ethernet port on the BCM-ETH. The BCM-PWS connects to a 24 Vac source.

815 W. 1800 N., Logan, UT 84321-1784 (435) 753-2342

3. Programming Alerton Devices

3.1 Configuration

Use HyperTerminal to do the following:

1. Open the panel on the BCM to be configured, and use the DB-9 port to connect the module to a computer with a null modem cable.
2. Open HyperTerminal and create a new connection. Enter the port settings as shown in Figure 5.
3. Once the settings have been entered, select 'OK' and press enter. A window similar to the window shown in Figure 6 should appear.
4. Enter the password (the default password is 'pass').

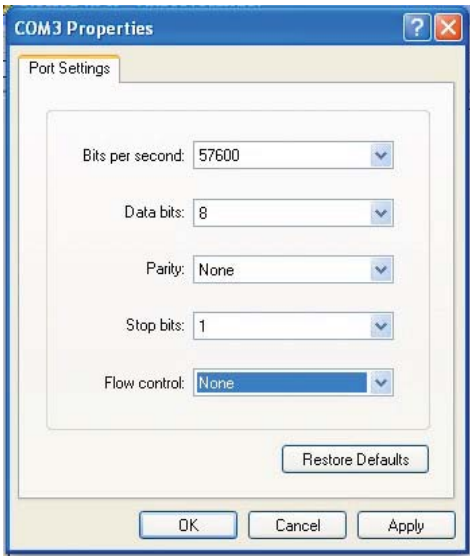

Figure 5: HyperTerminal Port settings.

5. After logging in, configure the BCM-ETH as follows.
   a. Give it a unique device instance.
   b. Enter a network number. This number must match the network number of all other devices running on the same network.
   c. Can use the defaults for the other settings although you may want to set the correct date.

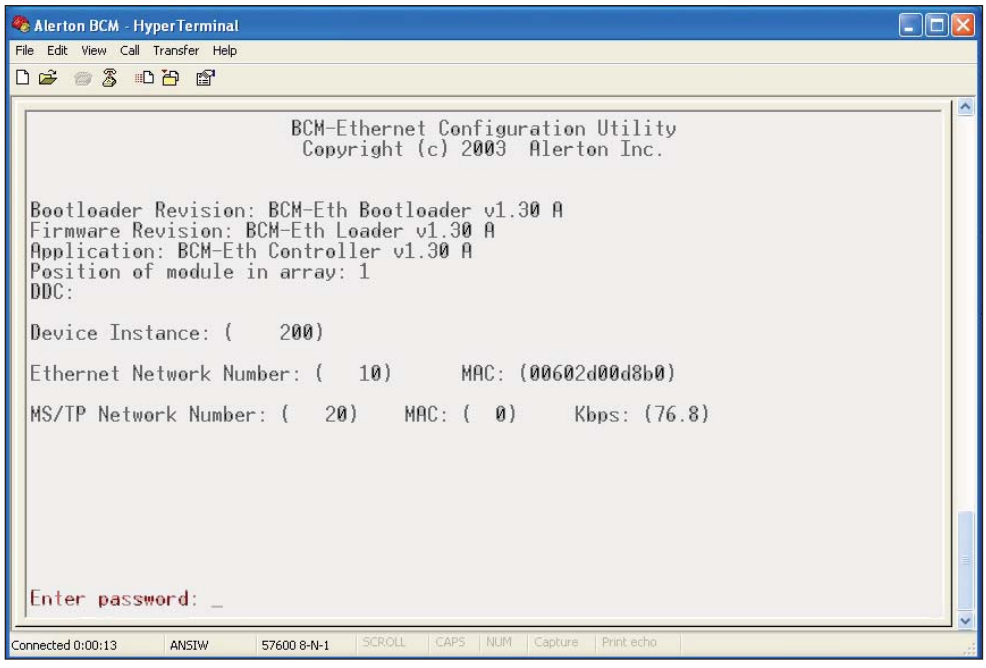

Figure 6: Established connection to a BCM-ETH.

815 W. 1800 N., Logan, UT 84321-1784 (435) 753-2342

6. Configure the BCM-MDBS as follows:
   a. Give it a unique device instance.
   b. Enter a network number that matches the network number of all other devices running on the same network.
   c. Under Modbus protocol, use the settings shown in Figure 7.
   d. Can use the defaults for the other settings although you may want to set the correct date.

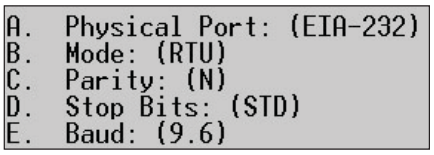

Figure 7: BCM-MDBS Modbus settings.

Alerton equipment is programmed using a DDC. However, communication does not require DDC programming. The Envision software are used to configure descriptions for all AVs and BVs to be passed between protocols. These descriptions enable communication.

1. Do the following to add configured devices on Envision software:
   a. After opening the Envision software and connecting the Ethernet cables to both your computer and the BCM-ETH, go to 'BACTalk' -> 'Device Manager'.
   b. Click the 'Device Scan...' button and a new window should open.
   c. In the new window, click the 'Scan...' button. This will generate a list of all connected devices.
   d. Select the device you wish to add, and click the 'Save To Table' button.
   e. Close both windows. The devices are now accessible with Envision software.
2. Do the following to enter descriptions for AV's and BV's:
   a. After opening Envision software, edit a display by going to 'Edit' -> 'Current display', then go to 'Place Item' -> 'Pushbutton selector'.
   b. Place a button on the display and double click it to bring up the settings for the button.
   c. In the 'Button type' field, select 'Device template'.
   d.. Enter the 'Device inst' you specified for the BCM-MDBS during device configuration.
   e. For the other settings, use the values shown in Figure 8.

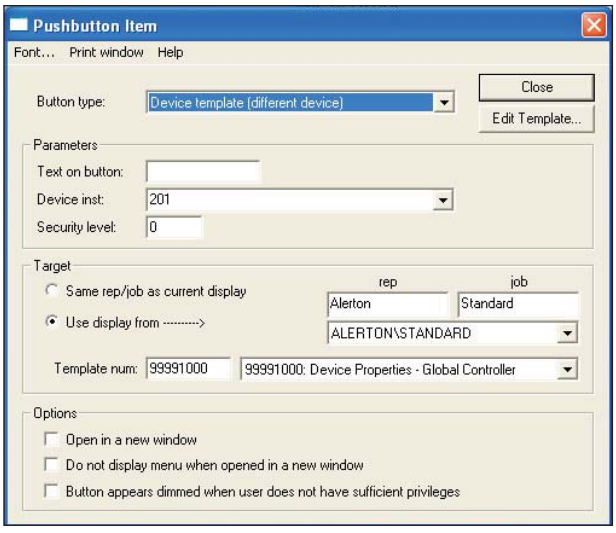

Figure 8: Pushbutton settings.

815 W. 1800 N., Logan, UT 84321-1784 (435) 753-2342

f. Close the window, then save the display and exit edit mode (Edit -> Exit Edit Mode).

2. Push the button that was just created, and then press the AV's button near the top right corner of the screen that opens. All of the AV values are shown with their descriptions on their right. To enter a description for an AV, click the button next to the description field. A new window with a text field should open.

4. The syntax for the description is as follows:

   I|O|W|T|A,<Slave Address>,<Reference>,<Register or Coil Number>, <Extended Memory File Number>,F|Fb|Fl,L,Lb|Ll,BU<nn>, <Real Number1>,<Real Number2>,M<nn><Descriptor>

   Optional fields (identified below) do not require a place holding comma if left out. Spaces are not allowed.

   Below are brief descriptions of each field. Full descriptions are provided in the BCM manual (LTBT-TM-BCMIOG, Pg. 65).

   a. ***I|O|W|T|A*: *I*** is the input to the BCM. ***O*** is the output from the BCM. ***W, T, and A*** are alternatives to ***O*** that are not necessary for communication with the datalogger.

   b. <***Slave Address***>: This address is the Modbus slave number you assigned to your datalogger.

   c. <***Reference***>**:** Enter ***4*** for analog values, or ***0*** for boolean values.

   d. <***Register or Coil Number***>: For analog values, the register number is ***2 * array position.*** For example, if you wanted to read or write to ***array position 1***, the register number is 2 (i.e., 2*1=2).

   e. <***Extended Memory File Number***>**:** This optional field is not needed for datalogger communication.

   f. ***F|Fb|Fl:*** This optional field refers to little endian or big endian. When passing analog values to and from the datalogger, ***Fl*** is required.

   g. ***The rest of the fields are optional and not necessary.***

   Examples of description are provided below:

   a. ***'I,25,4,8,Fl':*** This description reads registers 7 and 8 of the floating point array (array position 4) in a datalogger with Modbus slave number 25 and stores the value in the AV corresponding to the description.

   b. ***'O,3,4,10,Fl':*** This description writes the value stored in the corresponding AV to registers 9 and 10 of the floating point array (array position 5) in a datalogger with Modbus slave number 3.

   c. ***'I,14,0,15***': This description reads array position 15 of the boolean array in a datalogger with Modbus slave number 14 and stores the value in the BV corresponding to the description.

   d. ***'O,1,0,2'***: This description writes the value stored in the corresponding BV to array position 2 in a datalogger with Modbus slave number 1.

4. Verify that everything is working correctly by watching the AV value fields next to the description fields. They will update promptly.

815 W. 1800 N., Logan, UT 84321-1784 (435) 753-2342

3. Programming the Datalogger

No extra configuration is required for the datalogger. To program the datalogger, insert the ModBusSlave instruction in the datalogger program:. The ModBusSlave instruction has the following syntax:

ModBusSlave (ComPort, BaudRate, ModBusAddr, ModBusVariable, Boolean Variable)

Where,

BaudRate: Must match the baud rate set during configuration of the BCM-MDBS,

ModBusAddr: Must be the same number as the Modbus slave number specified in the AV description.

Registers are updated continuously as fast as the communication rate will allow repetition of the cycle. At 9600 baud, the load would consume approximately 1% of the datalogger's processing resources.

3.1 Example

Below is an example of the ModBusSlave instruction used to communicate with BCM.

ModBusSlave (Com1,9600,3,IOAnalog(),IOBoolean())

The above example uses COM1 on the datalogger to communicate with the BCM. The baud rate is 9600; the Modbus slave number is 3; the floating point array the BCM will read from and write to is IOAnalog(), and the boolean array the BCM will read from and write to is IOBoolean().

NOTE

This example is only a portion of the datalogger program. For more information about creating a datalogger program, refer to your datalogger manual, software manual, or CRBasic help.

815 W. 1800 N., Logan, UT 84321-1784 (435) 753-2342