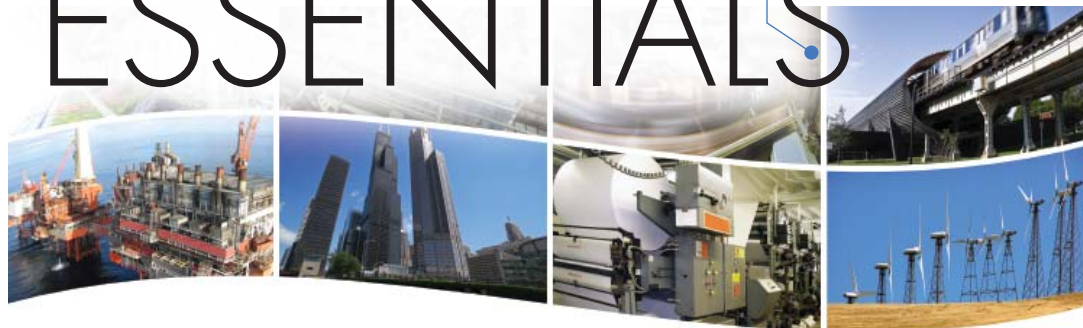


ESSENTIALS



© Contemporary Control Systems, Inc.

Mapping of Modbus Registers to BACnet® Objects Using the BAS Remote

People have asked “what advantage does BACnet equipment have over Modbus equipment?” The simple answer is that BACnet equipment is discoverable while Modbus equipment is not. What this means is that BACnet devices are modeled as a collection of objects that are “network visible.” By being able to view these BACnet objects and by understanding their pre-defined properties, much can be learned about the equipment. With Modbus, you will need to consult a user’s manual to understand the meaning of the Modbus registers. However, there is a way of creating BACnet objects from Modbus registers using the BAS Remote and some off-line configuration programs. Because Modbus equipment is popular, being able to integrate these devices into a “single-seat BACnet system” is important.

Modbus Register Addressing

Modbus data is considered to be segregated into four memory blocks—coils, discrete inputs, input registers and holding registers. Each memory location in each block can be accessed by a 16-bit address. Discrete inputs and coils are considered to be 1-bit registers while input registers and holding registers are 16-bit. A 32-bit register would require two memory locations. Addressing practice has been to use five-digit decimal references with the leading digit signifying a memory block as shown in Table 1. This approach is simpler to understand but it only addresses 10,000 points or registers instead of the 65,536 which are possible. However, there are still plenty of points and registers to work with so this representation was chosen for Modbus mapping.

Memory Block	Bits	Access	Address Range
Coils	1	Read/Write	00001-09999
Discrete inputs	1	Read-Only	10001-19999
Input registers	16	Read-Only	30001-39999
Holding registers	16	Read/Write	40001-49999

Table 1—Common decimal representation of Modbus registers.

Modbus Device Addressing

Modbus slaves are addressed from 1-247 while address 0 reserved for the broadcast address. The BAS Remote functions as the Modbus master and therefore has no address assignment. There can be several Modbus devices attached to the MB bus on the BAS Remote so each one must be assigned a unique Modbus address. The range of addresses has been restricted from 10-39 because BACnet instance numbers will be automatically assigned based upon Modbus device address and Modbus register address to greatly reduce configuration time. This also avoids potential conflicts with any BAS Remote internal I/O.

Modbus Function Codes

In order to act upon the data within the Modbus memory blocks a number of function codes have been defined. However, not all Modbus devices support all the available function codes. The BAS Remote supports the following function codes shown in Table 2 which are adequate for accessing common Modbus devices.

Code	Description
1	Read coils
2	Read discrete inputs
3	Read holding registers
4	Read input registers
5	Write single coil
6	Write single register
15	Write multiple coils
16	Write multiple registers

Table 2—The BAS Remote supports the most common Modbus function codes

BACnet Objects

Looking at the four types of Modbus data and available BACnet objects it is obvious that coils should be binary outputs (BO) and discrete inputs should be binary inputs (BI). Holding registers could be just about anything so assigning them as analog outputs (AO) makes sense because they can be read or written. Input registers can only be read so they will be represented as an analog input (AI) which is read-only.

BACnet Object Properties

The next step is to determine the required properties that must be supported for these four objects and to learn which properties must be configured. The abbreviated conformance code table (Table 3) shows what properties are required and what are optional.

Property	BI	BO	AI	AO
Object Identifier	R	R	R	R
Object Name	R	R	R	R
Object Type	R	R	R	R
Present Value	R	W	R	W
Status Flags	R	R	R	R
Event State	R	R	R	R
Out of Service	R	R	R	R
Polarity	R	R	NA	NA
Units	NA	NA	R	R
Priority Array	NA	R	NA	R
Relinquish Default	NA	R	NA	R
COV Increment	NA	NA	O	O

Table 3—Abbreviated Conformance Code table.

O means optional; **R** means required to be read; **W** means required to be read and written; **NA** means not applicable to this object

The Object Identifier must be unique within the BAS Remote regardless of the number of identical Modbus devices that are attached to the MB bus. Likewise, the Object Name must be unique within the BAS Remote. The Object Type will be set accordingly to represent the register as BI, BO, AI or AO. The Present Value represents the actual value of the point being read or written to by the BAS Remote. Status Flags, Event State and Out of Service need not be configured. They are handled by the BAS Remote. Polarity is a required property for binary points. It was decided to preset all polarities to NORMAL so as to not invert the state of the Modbus point. Units for analog points need to be identified but it is possible to indicate no units. For output points the Priority Array and Relinquish Default must be set. Priority Array is set when the Present Value is written. The Present Value assumes the value of the Relinquish Default when there is a null Priority Array.

The BAS Remote supports Change of Value (COV), so the COV increment must be set on analog points that will be subscribed.

The first issue is to create a unique Object Instance for each Modbus register in each attached Modbus device. It was decided to combine the Modbus Device Address with the Modbus register address. Modbus addresses have been forced to reside within the 10-39 range so this number will precede the register address. For example, there is a Modbus device at address 10 and it has holding registers 40001 through 40005. Five objects would be created with Object Instances 1040001 through 1040005. If we attach a second identical Modbus device at the next Modbus device address, the resulting Object Instances will range from 1140001-1140005. The Object Instance is combined with the Object Type to create a unique Object Identifier.

The next issue is the Object Name which must be unique. Any unique name would work but it is recommended to use the register description for the corresponding Modbus register. For example, if register 40001 means the “Voltage phase A to neutral,” that will be the name. However, if a second identical Modbus device is added, it will have the same name for that register which must be avoided. The best way to make them unique is to append a sequence number to all Object Names in order to ensure their uniqueness. This can be accomplished automatically by using the Project Builder tool from Contemporary Controls.

Configuration Tools

Contemporary Controls has developed two off-line programs that will assist in configuring Modbus register to BACnet mapping. This first is called the Modbus Profile Generator and the second is the Project Builder. Both can be downloaded for free from the company’s web site. Once the project is built, the resulting file can be uploaded to the BAS Remote for execution.

Sedona Framework™ and Modbus Devices

With the BAS Remote 3.0, a Sedona Virtual Machine (SVM) exists in the unit that will execute function block programs developed with Sedona Workbench. Not only will BAS Remote internal I/O appear as Sedona objects, the Modbus registers mapped to BACnet will as well. By using the Modbus expansion port on the BAS Remote, the total I/O count being controlled by Sedona Framework can increase greatly beyond the BAS Remote’s internal I/O.