BASR

Versatile Building Automation Appliance

BAS Remote

User Manual

# TD040300-0MF

(for firmware version 3.1.x)

CONTEMPORARY CONTROLS®

Trademarks

Contemporary Controls and CTRLink are registered trademarks of Contemporary Control Systems, Inc. BACnet is a registered trademark of the American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. Powered by Sedona Framework is a trademark of Tridium, Inc. Other product names may be trademarks or registered trademarks of their respective companies.

Copyright

Contemporary Control Systems, Inc.	Tel:	+1-630-963-7070
2431 Curtiss Street	Fax:	+1-630-963-0109
Downers Grove, Illinois 60515 USA	E-mail:	info@ccontrols.com
	WWW:	http://www.ccontrols.com
Contemporary Controls Ltd	Tel:	+44 (0)24 7641 3786
14 Bow Court	Fax:	+44 (0)24 7641 3923
Fletchworth Gate	E-mail:	info@ccontrols.co.uk
Coventry CV5 6SP UK	WWW:	http://www.ccontrols.co.uk
Contemporary Controls Ltd	Tel:	+44 (0)24 7641 3786
14 Bow Court	Fax:	+44 (0)24 7641 3923
Fletchworth Gate	E-mail:	info@ccontrols.co.uk
Coventry CV5 6SP UK	WWW:	http://www.ccontrols.eu
Contemporary Controls GmbH	Tel:	+49 (0)341 520359 0
Fuggerstraße 1 B	Fax:	+49 (0)341 520359 16
D-04158 Leipzig Deutschland	E-mail:	info@ccontrols.de
	WWW:	http://www.ccontrols.eu

Disclaimer

Contemporary Control Systems, Inc. reserves the right to make changes in the specifications of the product described within this manual at any time without notice and without obligation of Contemporary Control Systems, Inc. to notify any person of such revision or change.

WARNING — This is a Class A product as defined in EN55022.
In a domestic environment this product may cause radio interference
in which case the user may be required to take adequate measures.

1 Table of Contents

2 Introduction

The **BAS Remote Master** provides the ultimate in flexibility. It can be used for Expansion I/O at remote locations where an Ethernet connection exists. Its built-in router and gateway capability addresses unique integration needs where more than one communications protocol is involved. It can function as a controller with its resident Sedona Virtual Machine. Powered by a Linux engine, the **BAS Remote Master** can operate as BACnet/IP or Modbus TCP remote I/O, Sedona controller, Modbus Serial to Modbus TCP router, Modbus Serial to BACnet gateway, and Modbus master to attached Modbus slaves – all at the same time. The BAS Remote also supports web services. You can customize the unit via custom web pages that utilize special java applets provided in the device. A 10/100 Mbps Ethernet port allows connection to IP networks and popular building automation protocols such as Modbus TCP, BACnet/IP, and Sedona SOX. Six universal I/O points and two relay outputs can be configured through resident web pages using a standard web browser and without the need of a special programming tool. A 2-wire Modbus serial port can greatly expand the I/O count with built-in routing to Modbus TCP clients. If BACnet mapping is preferred, the unit incorporates a Modbus serial to BACnet/IP gateway. Additional universal I/O can be achieved with the simple addition of **BAS Remote Expansion** modules. **The BAS Remote Master PoE** has the same capabilities as the **BAS Remote Master** except it is powered over the Ethernet connection — thereby providing a "One Cable Solution."

Both the **BAS Remote Master** and **BAS Remote Expansion** modules have the same I/O capability. Six universal input/output points are provided on each module. Depending upon configuration, each unit can accommodate a contact closure from a digital point, a thermistor, voltage or current analog input from a field transmitter or supervisory controller. Analog inputs can range from 0–5 VDC, 0–10 VDC or 0–20 mA. Inputs can be scaled to accommodate ranges such as 1–5 VDC, 2–10 VDC, and 4–20 mA. Input point resolution is 10-bits.

Type II and III 10 kΩ thermistor calibration curves are resident in the BAS Remote. Single-point calibration of temperature is accomplished using the internal web server. Inputs can accept pulse trains in the range of 0–40 Hz (50% duty cycle) to measure flow rates.

Analog outputs can be 0–10 VDC or 0–20 mA. However, scaling for 2–10 V, 0–5 V, 1–5 V and 4–20 mA is possible. LED indicators identify the state of I/O points. Output point resolution is 12 bits.

There are two relay outputs available with both normally open (NO) and normally closed (NC) contacts. The relay output rating is 30 VAC/VDC, 2A.

There are two non-isolated 2-wire EIA-485 Expansion ports on the Master module. The downstream port (DN) is intended for Expansion modules while the Modbus (MB) port functions as either a Modbus TCP router or Modbus serial to BACnet gateway — allowing for the attachment of Modbus 2-wire or 3-wire EIA-485 serial devices. On BAS Remote Expansion modules, the two ports are marked UP and DN, and are dedicated for communication with the **BAS Remote Master** module and other Expansion modules.

All field connectors are removable — making field replacement of units quick and simple.

A single RJ-45 shielded connector provides the 10/100 Mbps twisted-pair Ethernet connection. The unit supports auto-negotiation of data rate and duplex. A resident web server facilitates commissioning and troubleshooting. Configuration is accomplished via Ethernet. Java must be enabled in the browser used to access the BAS Remote.

Power for the BAS Remote can be derived from a 24 VAC Class 2 transformer capable of delivering 17 VA or from a 24 VDC power supply capable of at least 10 W. Since the unit incorporates a half-wave rectified power supply, attached I/O points and the power supply can share a common ground. Therefore, the BAS Remote can be powered by the same control transformer used to power other half-wave rectified control equipment. The BAS Remote can be DIN-rail mounted into a control panel. If panel mounting is required, use the supplied mounting tabs.

The BAS Remote conforms to the BACnet/IP standard and therefore allows field I/O to be directly accessed via Ethernet without the need of a router. A standard web browser with Java enabled is used for commissioning and troubleshooting. The BAS Remote adheres to the BACnet Application Specific Controller (B-ASC) profile.

2.1 Features and Benefits

Versatile Control Device — remote I/O, router, gateway and controller

- Web-page configuration
- BACnet/IP Remote I/O
- Modbus TCP Remote I/O
- Modbus Serial to Modbus TCP Router
- Modbus Serial to BACnet/IP Gateway
- Modbus Master to Attached Modbus Slaves
- *Powered by Sedona* Controller
- Power over Ethernet (PoE)
- Java Applets for customisable web pages
- Web Services

Flexible Input/Output — expandable by adding modules

- Six universal input/output points — web-page configurable
- Two relay outputs
- Thermistor, voltage, current, contract closure and pulse inputs
- Voltage, current and relay outputs
- 2-wire Modbus Serial Expansion port
- 2-wire Expansion port for up to three Expansion modules

2.2 Software

The provided CD-ROM contains:

- This User Manual
- A copy of the Installation Guide that is packaged with the product
- Additional information of interest

2.3 Product Image (Master Module)

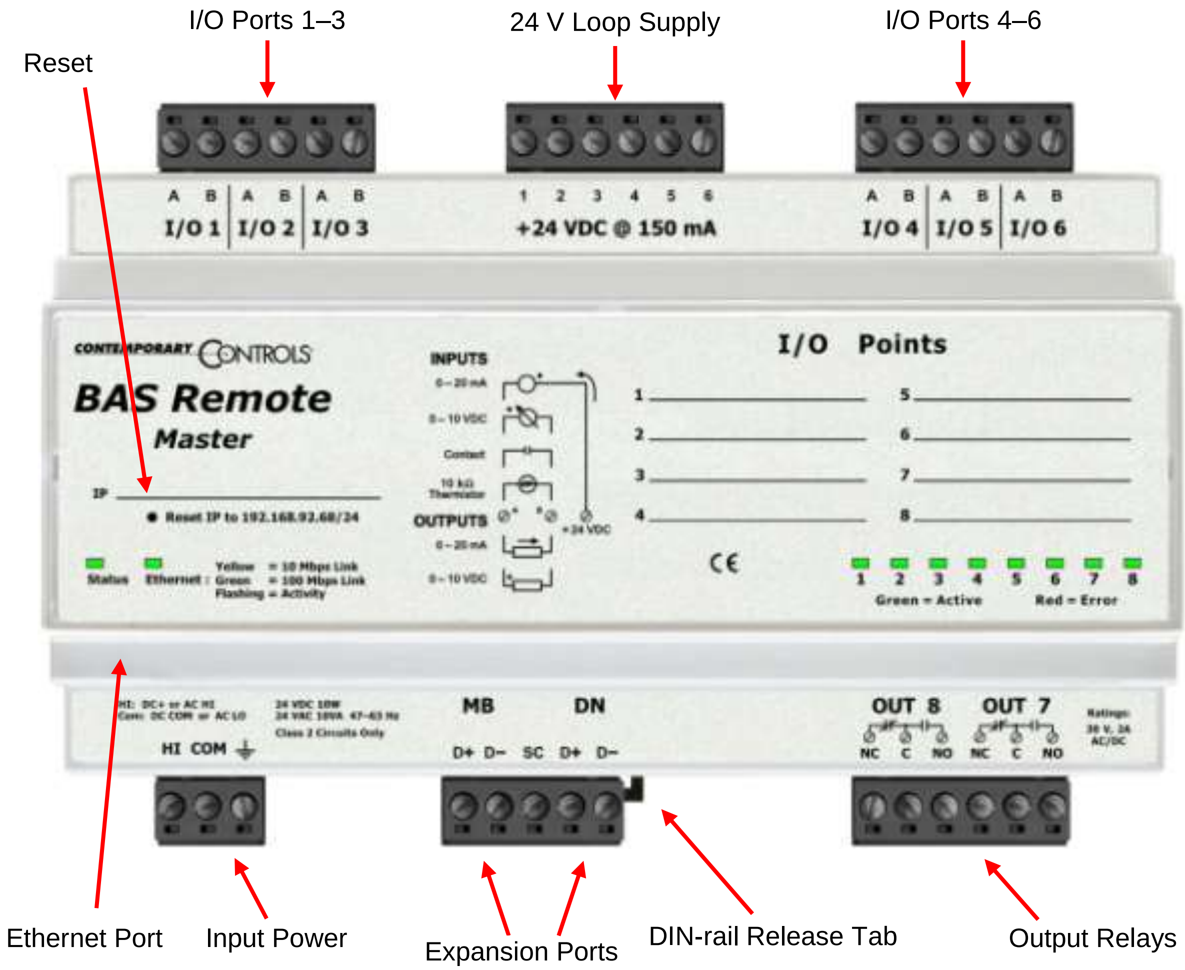


Figure 1 — BAS Remote Master Module Main Features

3 Specifications

3.1 Universal Input/Outputs — Channels 1–6

Configured As	Limits
Analog Input	0–5 VDC, 0–10 VDC or 0–20 mA scalable by user. 10-bit resolution. Input impedance 100 kΩ on voltage and 250 Ω on current.
Temperature Input	Type II or Type III thermistors +40°F to +110°F (+4.4°C to +44°C)
Contact closure input	Excitation current 2 mA. Open circuit voltage 24 VDC. Sensing threshold 0.3 VDC. Response time 20 ms.
Pulse input	0–10 VDC scalable by user. User adjustable threshold. 40 Hz maximum input frequency with 50% duty cycle.
Analog Output	0–10 VDC or 0–20 mA scalable by user. 12-bit resolution. Maximum burden 750 Ω when using current output.

3.2 Relay Outputs — Channels 7–8

Limits
Form “C” contact with both NO and NC contacts available. 30 VAC/VDC 2 A. Class 2 circuits only.

3.3 Communications

Protocol	Data Link and Physical Layers
BACnet/IP — Master only	ANSI/IEEE 802.3 10/100 Mbps Ethernet. 10BASE-T, 100BASE-TX, auto-negotiation of speed and duplex. Auto-MDIX. 100 m maximum segment length. Default IP address is 192.168.92.68/24.
Modbus TCP — Master only	
Expansion Bus (UP/DN)	Modified Modbus serial protocol. 2-wire non-isolated EIA-485 57.6 kbaud. Maximum segment length 100 m.
Modbus Serial — Master only (MB Bus)	Modbus serial ASCII or RTU protocol. 2-wire non-isolated EIA-485. 2.4, 4.8, 9.6, 19.2, 38.4, 57.6, 115.2 kbps. Max segment length 100 m. Jumper selectable bias and termination.

3.4 Protocol Compliance

Protocol	Compliance
BACnet/IP	ASHRAE 135-2008 annex J. Application specific controller device profile B-ASC.
Modbus TCP	Modbus Application Protocol Specification V1.1b Dec. 28, 2006, Modbus.org. Modbus Messaging on TCP/IP Implementation Guide V1.0b October 24, 2006, Modbus.org.
Modbus serial	Modbus over Serial Line Specification and Implementation Guide V1.02 December 20, 2006, Modbus.org.

3.5 Power Requirements

Item	Limits
Input power	Master module: 24 VAC/VDC ± 10%, 47–63 Hz, 17 VA Master PoE module: 48 VDC ± 10% 10 W Expansion module: 24 VAC/VDC ± 10%, 47–63 Hz, 17 VA
Loop supply	+24 VDC nom, 150 mA maximum

3.6 General Specifications

Item	Description
Protection	All inputs and outputs (except for relay outputs and communications ports) are over-voltage protected up to 24 VAC and short-circuit protected.
Environmental	Operating temperature 0° to +60°C. Storage temperature –40°C to +85°C. Relative humidity 10 to 95%, non-condensing.
Weight	0.6 lbs. (0.27 kg).

3.7 LED Indicators

LED Indicator	Indication
I/O 1–6 configured as Analog input	Green: > 1% of range, otherwise off
I/O 1–6 configured as Temperature input	Green: sensor detected Red: open
I/O 1–6 configured as Contact input	Green: contact closed, otherwise off
I/O 1–6 configured as Pulse input	Green: pulse sensed, otherwise off
I/O 1–6 configured as Analog output	Green: commanded output Red: expected output not within 40 mV on voltage or 0.2 mA on current
Status	Red: device in reset Green flashing: booting up Green: running application Green flashing: Modbus serial activity after application is running — *Master only*
Ethernet — Master module only	Yellow: 10Mbps; flashes with activity Green: 100 Mbps; flashes with activity
Network — Expansion module only	Green flashing: Expansion bus activity

3.8 Electromagnetic Compatibility

Standard	*Test Method*	*Description*	*Test Levels*
EN 55024	EN 61000-4-2	Electrostatic Discharge	6 kV contact
EN 55024	EN 61000-4-3	Radiated Immunity	10 V/m, 80 MHz to 1 GHz
EN 55024	EN 61000-4-4	Fast Transient Burst	1 kV clamp & 2 kV direct
EN 55024	EN 61000-4-5	Voltage Surge	1 kV L-L & 2 kV L-Earth
EN 55024	EN 61000-4-6	Conducted Immunity	10 V (rms)
EN 55024	EN 61000-4-11	Voltage Dips & Interruptions	1 Line cycle, 1–5 s @100% dip
EN 55022	CISPR 22	Radiated Emissions	Class A
EN 55022	CISPR 22	Conducted Emissions	Class B
CFR 47, Part 15	ANSI C63.4	Radiated Emissions	Class A

3.9 Field Connections

Terminal	Universal I/Os 1 – 3
I/O 1 A	Universal I/O point 1 high
I/O 1 B	Universal I/O point 1 low
I/O 2 A	Universal I/O point 2 high
I/O 2 B	Universal I/O point 2 low
I/O 3 A	Universal I/O point 3 high
I/O 3 B	Universal I/O point 3 low

Terminal	Universal I/Os 4 – 6
I/O 4 A	Universal I/O point 4 high
I/O 4 B	Universal I/O point 4 low
I/O 5 A	Universal I/O point 5 high
I/O 5 B	Universal I/O point 5 low
I/O 6 A	Universal I/O point 6 high
I/O 6 B	Universal I/O point 6 low

Terminal	Relay Outputs
OUT 8 NC	Output 8 normally-closed contact
OUT 8 C	Output 8 common
OUT 8 NO	Output 8 normally-open contact
OUT 7 NC	Output 8 normally-closed contact
OUT 7 C	Output 8 common
OUT 7 NO	Output 8 normally-open contact

Terminal	+24 VDC @ 150 mA Loop Supply
1	+24 VDC
2	+24 VDC
3	+24 VDC
4	+24 VDC
5	+24 VDC
6	+24 VDC

Terminal	Expansion Ports — Master Module	Terminal	Expansion Ports — Expansion Module
MB-D+	Modbus Serial Bus positive terminal	**UP-D+**	Upstream Expansion positive terminal
MB-D–	Modbus Serial Bus negative terminal	**UP-D–**	Upstream Expansion negative terminal
SC	Modbus signal common	**SC**	Not used
DN-D+	Downstream Expansion positive terminal	DN-D+	Downstream Expansion positive terminal
DN-D–	Downstream Expansion negative terminal	DN-D–	Downstream Expansion negative terminal

Terminal	Power — Master Module*	Power — Expansion Module
HI	High AC or DC +	High AC or DC +
COM	AC or DC common	AC or DC common
Earth	Optional earthing connection	No connection

* BASR-M model only

3.10 Ordering Information

Model	Description
BASR-8M	**BAS Remote Master** with eight I/O points
BASR-8X	BAS Remote Expansion with eight I/O points
BASR-8M/P	**BAS Remote Master** PoE with eight I/O points

3.11 Dimensional Drawing

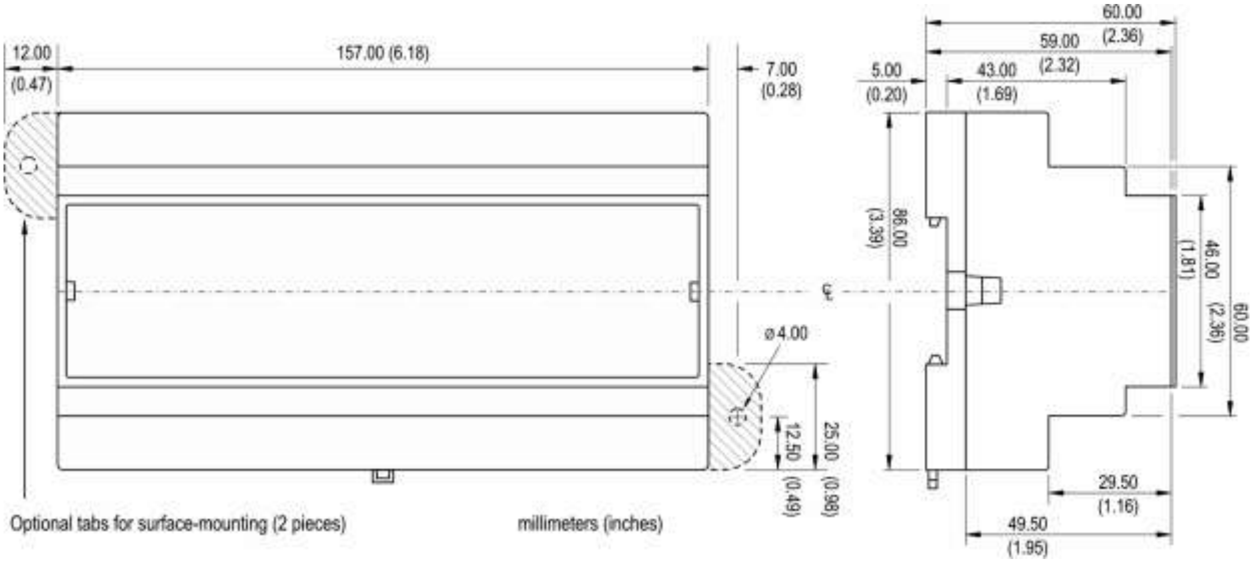
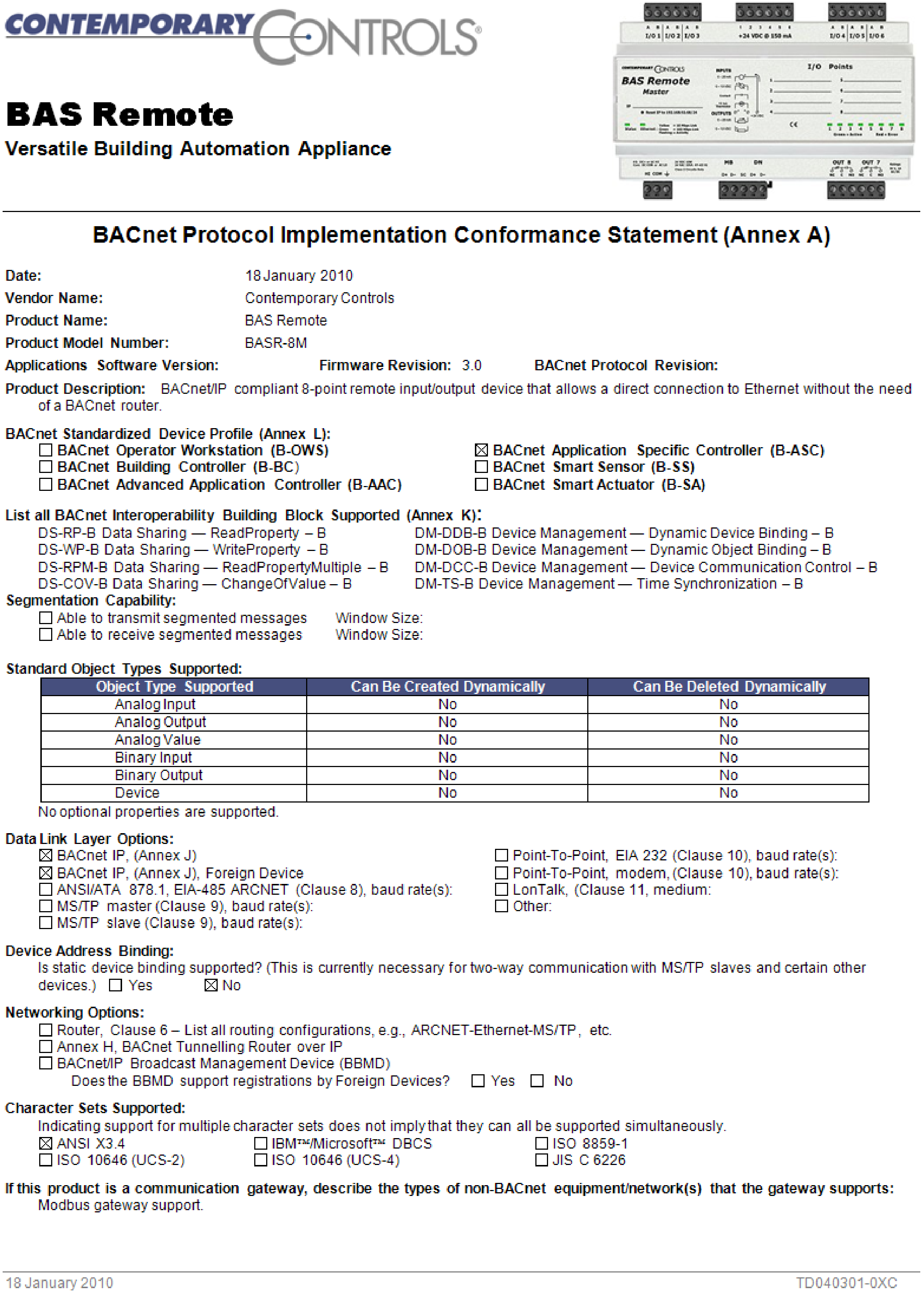


Figure 2 — BAS Remote Dimensions

3.12 PICS Statement



CONTEMPORARY CONTROLS

BAS Remote

Versatile Building Automation Appliance

BACnet Protocol Implementation Conformance Statement (Annex A)

Date: 18 January 2010
Vendor Name: Contemporary Controls
Product Name: BAS Remote
Product Model Number: BASR-8M
Applications Software Version: **Firmware Revision:** 3.0 **BACnet Protocol Revision:**

Product Description: BACnet/IP compliant 8-point remote input/output device that allows a direct connection to Ethernet without the need of a BACnet router.

BACnet Standardized Device Profile (Annex L):

☐ **BACnet Operator Workstation (B-OWS)**
☐ **BACnet Building Controller (B-BC)**
☐ **BACnet Advanced Application Controller (B-AAC)**
☒ **BACnet Application Specific Controller (B-ASC)**
☐ **BACnet Smart Sensor (B-SS)**
☐ **BACnet Smart Actuator (B-SA)**

List all BACnet Interoperability Building Block Supported (Annex K):

DS-RP-B Data Sharing — ReadProperty – B
DS-WP-B Data Sharing — WriteProperty – B
DS-RPM-B Data Sharing — ReadPropertyMultiple – B
DS-COV-B Data Sharing — ChangeOfValue – B
DM-DDB-B Device Management — Dynamic Device Binding – B
DM-DOB-B Device Management — Dynamic Object Binding – B
DM-DCC-B Device Management — Device Communication Control – B
DM-TS-B Device Management — Time Synchronization – B

Segmentation Capability:

☐ Able to transmit segmented messages Window Size:
☐ Able to receive segmented messages Window Size:

Standard Object Types Supported:

Object Type Supported	Can Be Created Dynamically	Can Be Deleted Dynamically
Analog Input	No	No
Analog Output	No	No
Analog Value	No	No
Binary Input	No	No
Binary Output	No	No
Device	No	No

No optional properties are supported.

Data Link Layer Options:

☒ BACnet IP, (Annex J)
☒ BACnet IP, (Annex J), Foreign Device
☐ ANSI/ATA 878.1, EIA-485 ARCNET (Clause 8), baud rate(s):
☐ MS/TP master (Clause 9), baud rate(s):
☐ MS/TP slave (Clause 9), baud rate(s):
☐ Point-To-Point, EIA 232 (Clause 10), baud rate(s):
☐ Point-To-Point, modem, (Clause 10), baud rate(s):
☐ LonTalk, (Clause 11, medium:
☐ Other:

Device Address Binding:

Is static device binding supported? (This is currently necessary for two-way communication with MS/TP slaves and certain other devices.) ☐ Yes ☒ No

Networking Options:

☐ Router, Clause 6 – List all routing configurations, e.g., ARCNET-Ethernet-MS/TP, etc.
☐ Annex H, BACnet Tunnelling Router over IP
☐ BACnet/IP Broadcast Management Device (BBMD)
Does the BBMD support registrations by Foreign Devices? ☐ Yes ☐ No

Character Sets Supported:

Indicating support for multiple character sets does not imply that they can all be supported simultaneously.

☒ ANSI X3.4
☐ ISO 10646 (UCS-2)
☐ IBM™/Microsoft™ DBCS
☐ ISO 10646 (UCS-4)
☐ ISO 8859-1
☐ JIS C 6226

If this product is a communication gateway, describe the types of non-BACnet equipment/network(s) that the gateway supports:

Modbus gateway support.

18 January 2010 TD040301-0XC

4 Installation

The BAS Remote is intended to be mounted in an industrial enclosure or wiring closet on 35-mm DIN-rail or panel-mounted with screws (not provided). The panel-mounting tabs are packaged in a plastic bag within the shipping box. To use these tabs, Figure 3 illustrates how the two studs of each tab are press fitted into their respective holes in opposing corners of the case.

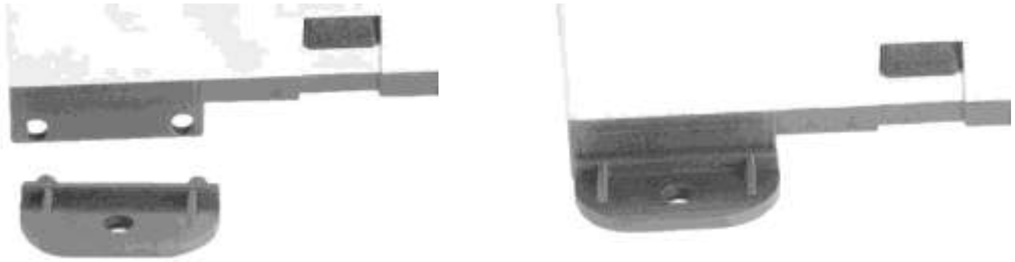


Figure 3 — Attaching Panel-Mounting Tabs

4.1 Power Supply

The power source for the internal supply is applied via the three terminals labelled *Earth*, *COM*, and *HI*. *Earth* allows external connection to earth if better EMC compliance is needed. *COM* is for the power source return and also serves as the BAS Remote common ground connection. Primary 24 VAC/VDC (± 10%) power is applied to *HI* and *COM*. *HI* connects to a diode accomplishes half-wave rectified power — while providing reverse input voltage protection.

Internally, the BAS Remote provides the 24 VDC loop supply to power external devices attached as inputs to the BAS Remote — you do not need a separate loop supply. Since the BAS Remote can source current via its analog outputs, an internal source of 24 VDC is provided for powering outputs. Collectively, the sum of input and output power cannot exceed 150 mA.

Maximum current draw for any I/O channel is 20 mA — yielding a total draw of 120 mA for all six channels. Analog output current sources from the same internal supply, so an external source of 24 VDC is unneeded — ***but a return common is***. Six +24 VDC pins are present to serve external transmitters, so they do not need a separate loop supply. However, the power supply must serve only its own BAS Remote module.

The BAS Remote requires 24 VDC or VAC from a source via a three-pin removable keyed connector. The proper connections for various power options are shown in Figure 4. Note that the **BAS Remote Master PoE** derives its power from the Ethernet cable and therefore no internal connection exists for the HI pin on the input power connector.

The recommended size for power conductors is 16–18 AWG (solid or stranded). Ground is directly connected to zero volts. Input connections are reverse-polarity protected.

NOTE: This device is intended for use with Class 2 circuits.

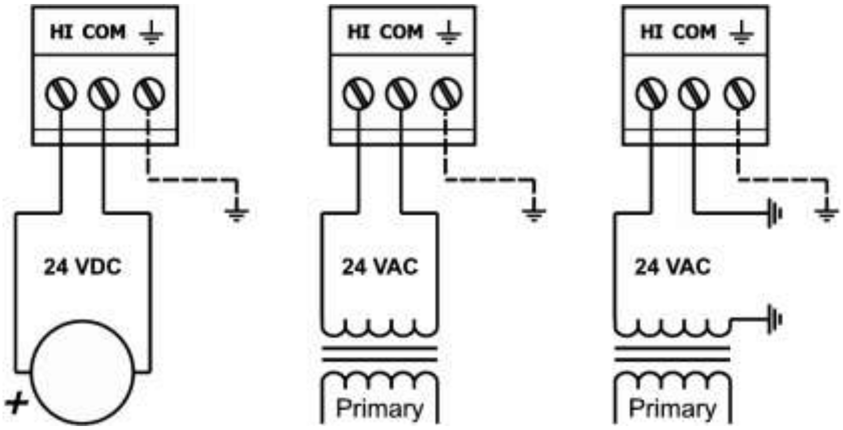


Figure 4 — Power Options

WARNING: Powering devices can present hazards. Read the next two sections carefully.

4.1.1 Power Supply Precautions

Internally, the BAS Remote utilizes a half-wave rectifier and therefore can share the same AC power source with other half-wave rectified devices. Sharing a common DC power source is also possible. Sharing AC power with full-wave rectified devices is NOT recommended. Full-wave rectified devices usually require a dedicated AC power source that has a secondary elevated above ground. Both secondary connections are considered HOT. AC power sources that power several half-wave devices have a common secondary connection called COMMON, LO, or GROUND. This connection might be tied to earth. The other side of the secondary is considered the HOT or HI side of the connection. Connect the HOT side of the secondary to the *HI* input on the BAS Remote and the LO side to *COM* on the BAS Remote. All other half-wave devices sharing the same AC power source need to follow the same convention. When using a DC power source, connect its positive terminal to the *HI* input on the BAS Remote and the negative terminal to *COM* on the BAS Remote. Reversing polarity to the BAS Remote will not damage the BAS Remote.

WARNING: Devices powered from a common AC source could be damaged if a mix of half-wave and full-wave rectified devices exist. If you are not sure of the type of rectifier used by another device, do not share the AC source with it.

4.1.2 Limited Power Sources

The BAS Remote should be powered by a limited power source complying with the requirements of the National Electric Code (NEC) article 725 or other international codes meeting the same intent of limiting the amount of power of the source. Under NEC article 725, a Class 2 circuit is that portion of the wiring system between the load side of a Class 2 power source and the connected equipment. For AC or DC voltages up to 30 volts, the power rating of a Class 2 power source is limited to 100 VA. The transformer or power supply complying with the Class 2 rating must carry a corresponding listing from a regulatory agency such as Underwriters Laboratories (UL).

4.2 Connecting Expansion Equipment

Input/output points beyond those available from the **BAS Remote Master** module can be increased by adding BAS Remote Expansion modules or by attaching Modbus serial devices to the MB bus. The *MB* port is used for connecting to 2-wire Modbus serial devices while the *DN* port is used for connecting to BAS Remote Expansion modules. Both ports are non-isolated EIA-485 compatible.

When installing equipment, make a record that identifies the power source, equipment locations, IP and MAC ID numbers, protocol in use, baud rate, cable colour coding, etc. — anything that will be helpful for future staff.

4.2.1 BAS Remote Expansion Module Connections

Expansion modules are intended to occupy positions to the right or left of the Master module on the same DIN-rail or on additional DIN-rails within the same control panel. In this situation only a short 2-wire twisted-pair cable is needed for making connections between *DN* on the Master module and *UP* on the first Expansion module. Up to three Expansion modules can attach to the Master module using a daisy-chain wiring scheme. The second Expansion module has its *UP* port connected to the preceding Expansion module's *DN* port. The last Expansion module will have a vacant *UP* port. The *D+* terminal on one device must attach to the *D+* terminal on the other. The same applies to the *D–* terminals. Bias and termination exists on the *UP* terminals. See Figure 6 for wiring details. For short connections, unshielded cable can be used. Expansion modules are automatically assigned Modbus addresses beginning with 2 based upon its position to the Master within the daisy-chain wiring.

4.2.1.1 Bias and Termination

Each Master and Expansion module has two Expansion ports. The Master ports are labelled *MB* and *DN*; the Expansion ports are *UP* and *DN*. These are shared buses where only one device drives the bus at any one time. When no device is driving the bus, the bus floats. To prevent noise from being interpreted as data, the bus must be biased to a valid state. (The Modbus Serial specification calls this polarization.) With no data on the bus, the *D+* pin is biased to be more positive than the *D–* pin. Bias is applied at only one point on the bus: the Master provides bias internally on its *MB* port, the Expansion does so on its *UP* port.

4.2.1.2 Communicating from Master to Expansion Modules

The Master uses the downstream port *DN* to communicate to the upstream port *UP* on the Expansion. If additional Expansion modules are used, they are cascaded such that the *DN* port of the Expansion module nearest the Master is connected to the *UP* port on the added Expansion module. Commands received by an Expansion module's *UP* port are relayed to its *DN* port while being read by the module itself. Similarly, a response received at the *DN* port is transferred to the *UP* port — eventually arriving at the Master. Thus all connections (Master-Expansion and Expansion-Expansion) are point-to-point with termination and bias in each *UP* transceiver. *DN* ports have termination only.

4.2.2 Modbus Serial Bus Connections

The Modbus serial Expansion port (MB) on the **BAS Remote Master** module is non-isolated EIA-485 compatible. When connecting to other non-isolated devices, care must be exercised to ensure that all non-isolated Modbus devices share the same ground reference (*COM*) with the **BAS Remote Master** module. This is usually accomplished by sharing the same power source. Configure the Modbus baudrate and protocol using the BAS Remote Modbus port web page.

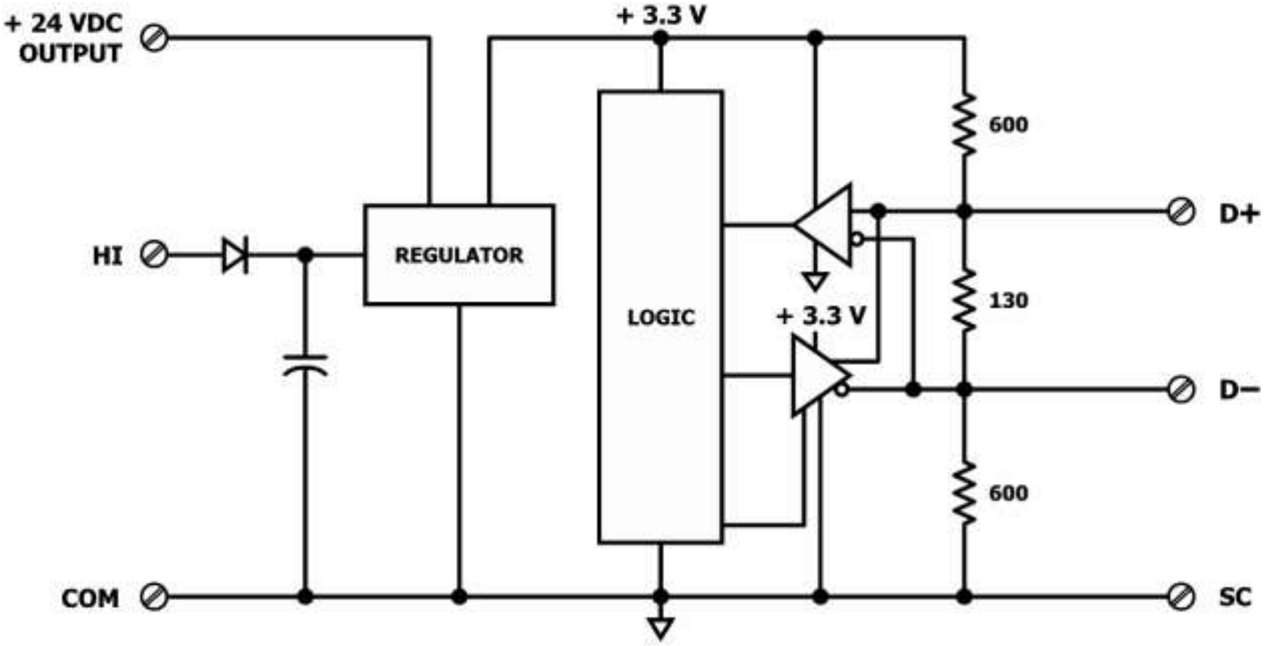


Figure 5 — Internal Termination and Bias

When connecting to an isolated 3-wire Modbus device, the signal common of the isolated device must be connected to the *SC* pin between the *MB* and *DN* ports. This ties the two reference points together for reliable communications. Refer to Figure 6 for wiring details.

Modbus serial device can only be attached to the *MB* port on the Master module. Refer to Figure 5 for details on the bias and termination network present on the *MB* port. Together, these resistors approximate one 120 Ω terminating resistor. Terminal *D+* represents the more positive connection for the EIA-485 Modbus serial network while *D–* represents the less positive connection. Make corresponding connections to Modbus serial devices. The last device on the bus should have applied bias and termination or just termination. A shielded twisted-pair cable should be used with interconnecting devices. Connect the shields together and attached to chassis at only one point. Refer to Figure 6 for wiring details.

4.2.3 Cabling Considerations

When attaching cables to the BAS Remote, Table 1 should be considered.

Function	Signalling and Data Rate	Minimum Required Cable	Maximum Segment Distance
Ethernet	10BASE-T 10 Mbps	Category 3 UTP	100 m (328 ft)
Ethernet	100BASE-TX 100 Mbps	Category 5 UTP	100 m (328 ft)
I/O	Unspecified	Solid: 16–22 AWG Stranded: 16–18 AWG	Unspecified
Expansion	Unspecified	Belden® 9841 or equivalent*	100 m (328 ft)
Modbus	Varied	Belden® 3106A or equivalent*	100 m (328 ft)

Table 1 — Cabling Considerations

* If using shielded cable, connect to chassis at only one point.

NOTE: Wire size may be dictated by electrical codes for the area where the equipment is being installed. Consult local regulations.

Observe in Table 1 that 10BASE-T segments can successfully use Category 3, 4 or 5 cable — but 100BASE-TX segments ***must*** use Category 5 cable. Category 5e cable is highly recommended as the minimum for new installations.

The Ethernet port of the BAS Remote employs Auto-MDIX technology so that either straight-through or crossover cables can be used to connect to the network.

5 Field Connections

5.1 Sample BAS Remote Wiring Diagram

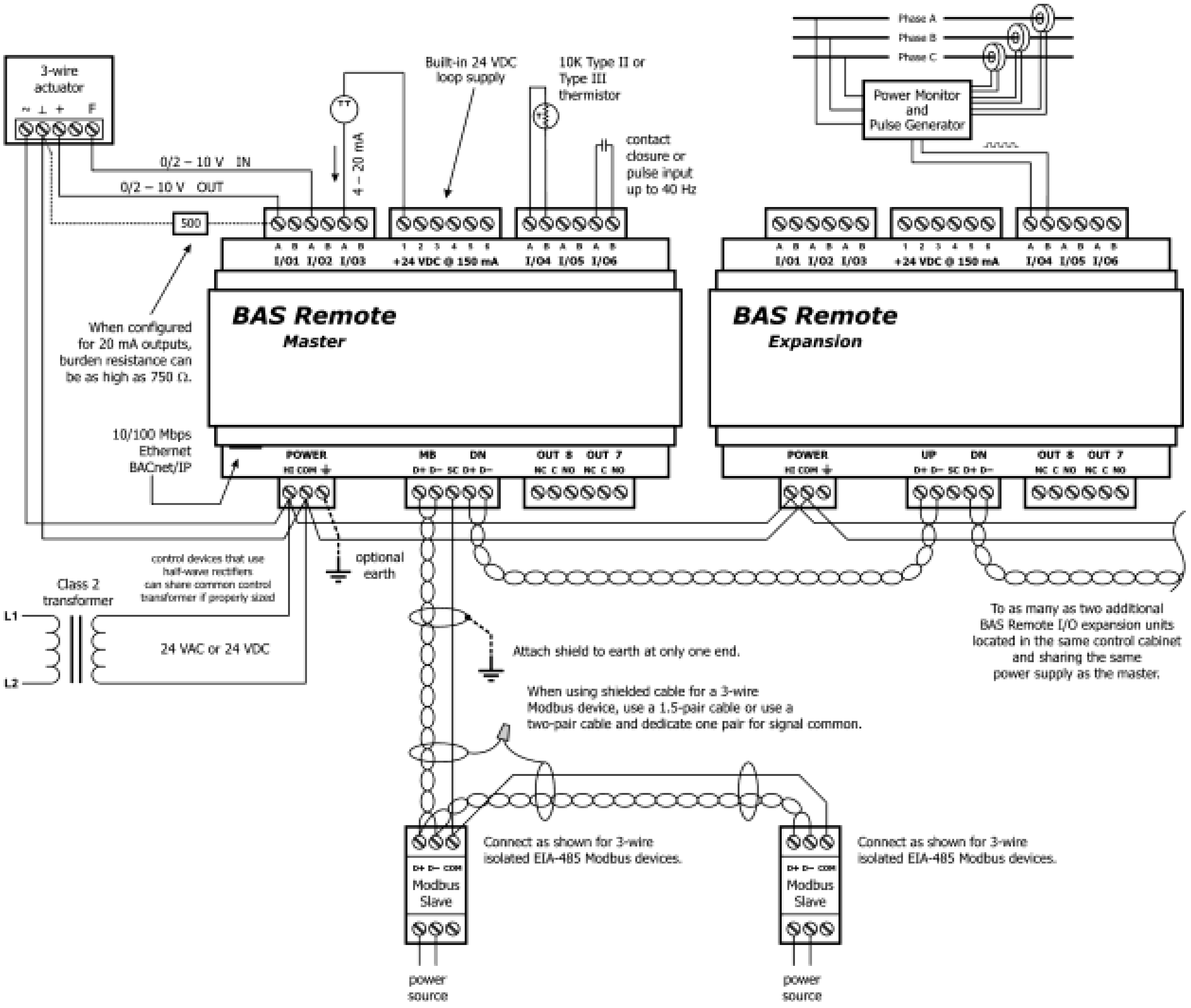

3-wire actuator
~ ⊥ + F
0/2 – 10 V IN
0/2 – 10 V OUT
500
TT
4 – 20 mA
Built-in 24 VDC loop supply
10K Type II or Type III thermistor
contact closure or pulse input up to 40 Hz
Phase A
Phase B
Phase C
Power Monitor and Pulse Generator
A B A B A B
I/01 I/02 I/03
1 2 3 4 5 6
+24 VDC @ 150 mA
I/04 I/05 I/06
BAS Remote
Master
BAS Remote
Expansion
When configured for 20 mA outputs, burden resistance can be as high as 750 Ω.
10/100 Mbps Ethernet BACnet/IP
POWER
HI COM ⏚
MB
DN
D+ D– SC D+ D–
OUT 8 OUT 7
NC C NO NC C NO
UP
DN
optional earth
control devices that use half-wave rectifiers can share common control transformer if properly sized
Class 2 transformer
L1
L2
24 VAC or 24 VDC
Attach shield to earth at only one end.
When using shielded cable for a 3-wire Modbus device, use a 1.5-pair cable or use a two-pair cable and dedicate one pair for signal common.
To as many as two additional BAS Remote I/O expansion units located in the same control cabinet and sharing the same power supply as the master.
D+ D– COM
Modbus Slave
Connect as shown for 3-wire isolated EIA-485 Modbus devices.
D+ D– COM
Modbus Slave
Connect as shown for 3-wire isolated EIA-485 Modbus devices.
power source
power source

Figure 6 — Sample BAS Remote Wiring Diagram

Wire Channels 1–6 so the most positive wire goes to the "A" terminal and the most negative wire to the "B" terminal.

The wiring options for Channels 1–6 are shown in Figure 7. For each case in which polarity matters, proper polarity is indicated.

Considerations in making field connections for various types of input and output devices are discussed in the following pages.

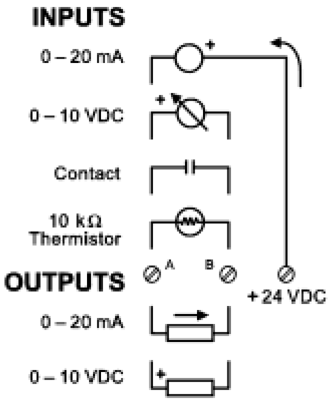


Figure 7 — I/O Options (Channels 1–6)

5.2 Thermistors

The BAS Remote has built-in calibration curves for 10 kΩ Type II or Type III thermistors. These devices have a non-linear with a negative coefficient of resistance to temperature and provide a nominal resistance of 10 kΩ at 25°C. Using the web server, configure an input for either Type II or Type III thermistor. As shown in Figure 8, connect the two-wire thermistor to points A and B. Polarity is not an issue. If averaging of temperature is desired, connect multiple thermistors in a series-parallel combination so that the nominal resistance remains at 10 kΩ as shown. Make sure that all devices are of the same type. The effective range of temperature measurement is from +40° to +110°F (+4.4° to +44°C). An open input results in a fault condition that produces a red LED indication for that channel.

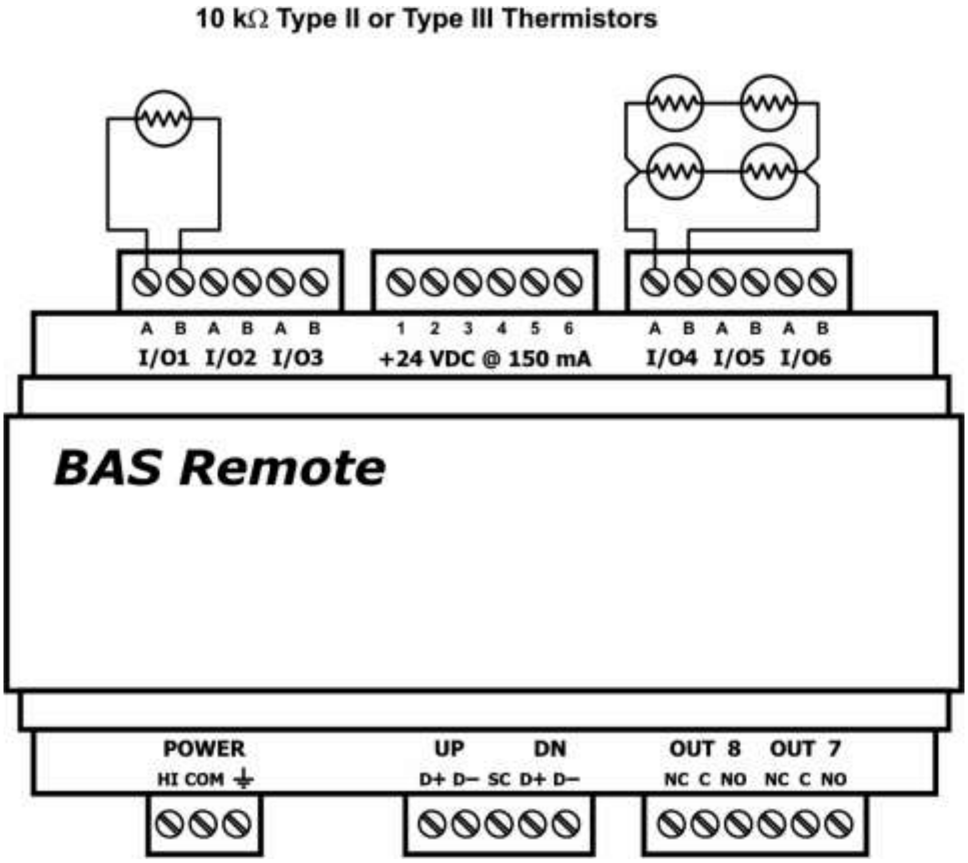


Figure 8 — Thermistor Connections

5.3 Contact Closure

The BAS Remote can sense the make or break of a contact from a relay or push-button. The contacts being sensed must be absent of any applied source of energy, and be rated for low-voltage, low-current switching. The BAS Remote will provide the electrical energy to be sensed. Using the web server, configure an input for contact closure. As shown in Figure 9, simply connect the contacts between points *A* and *B*. For common mechanical contacts, polarity is not an issue. The open-circuit voltage is 24 VDC and the short-circuit current is 2 mA.

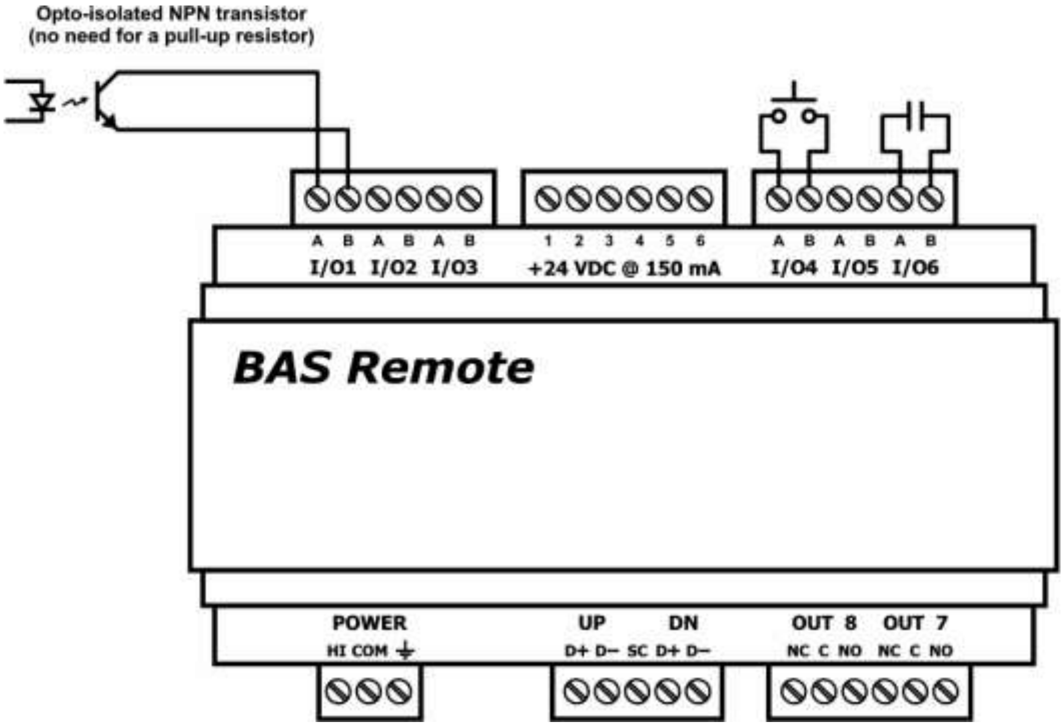


Figure 9 — Contact Closure Connections

For solid-state switches, there are further concerns. It is recommended that a solid-state device have an opto-isolated open-collector NPN transistor output stage with a collector-emitter output voltage (Vce) of at least 30 V. Output sinking current should be greater than 5 mA. The collector-emitter saturation voltage should be less than 0.2 V when sinking 2 mA. The emitter should be connected to point *B* and the collector to point *A* which is the more positive point. This polarity must be observed when using solid-state devices. When an input is configured for a contact closure, the BAS Remote sets the low-threshold to 2 V and the high-threshold to 3 V. When a contact is made or the solid-state switch is on (resulting in a saturated output), the voltage at point *A* is close to zero volts. The corresponding LED for that channel will be on. If the contact is opened or the solid-state switch is turned off, the voltage at point B will quickly begin to rise towards 24 V. Once the voltage passes the 3 V high-threshold, the input channel will sense the “off” state. To return to the “on” state, this voltage needs to return to 2 V. The one-volt difference is called *hysteresis*. There is no need to add an external pull-up resistor when using a contact closure input.

Contact closure inputs are sampled every 10 ms and for a change of state to be recognized, the input state must be stable for two consecutive samples. Therefore, contact closure response is from 20–30 ms.

5.4 Pulse Inputs

A variation on contact closure inputs is pulse inputs. In this situation speed is critical so the input filtering that limits the time response is removed. When an input is configured for Pulse Input, a pulse rate up to 40 Hz can be measured, assuming a 50% duty cycle. The pulse device could have an opto-isolated open-collector NPN transistor output stage like the one identified under Contact Closure, or it could provide an active sinusoidal output signal that needs to be detected. Data can be in the form of *frequency* or *pulse count*.

The Pulse Input voltage range is 0–10 VDC and the installer can set both the low-threshold and high-threshold on the Pulse Input web page. The difference in the two thresholds is the hysteresis. You can detect *sinusoidal* input signals by setting the high threshold below the positive peak and the low threshold above the negative peak. Setting the two thresholds well toward the centre of the sinusoidal waveform (rather than near its peaks) offers some noise immunity. It is not necessary for the input signal to swing from zero to 10 V. Any substantial swing within this range can be detected. The input impedance using Pulse Input is 100 kΩ. Connect the output of the pulse device to point *A* and the common to BAS Remote common as shown in Figure 10.

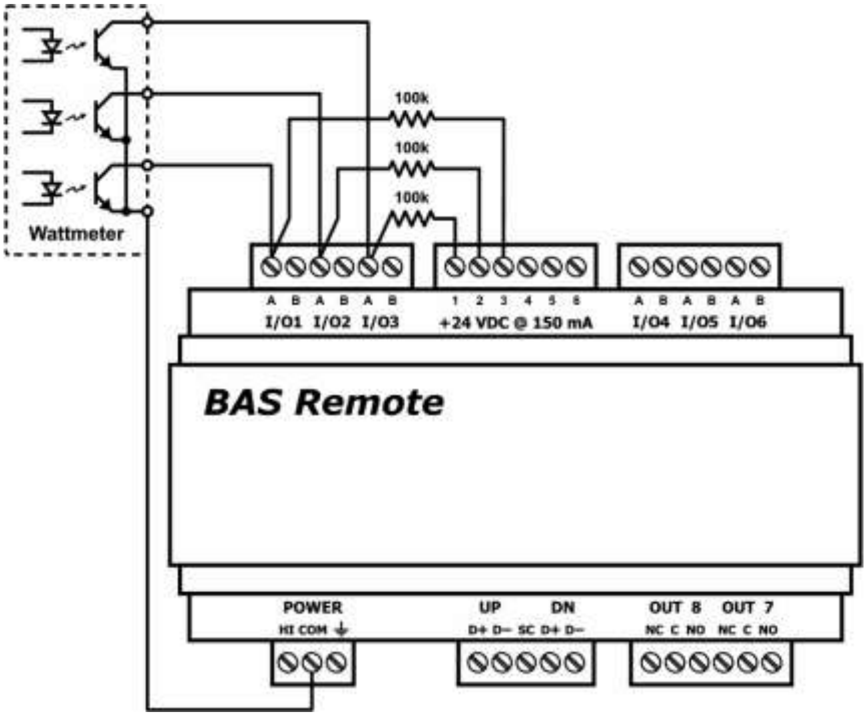


Figure 10 — Pulse Input Connections

The pulse output could be sinusoidal with no DC offset so the BAS Remote could experience both positive and negative excursions of the signal. The BAS Remote can only detect positive voltages so the negative excursions will be ignored. It is still possible to detect the input signal by only sensing the positive excursions.

When interfacing to a pulse device that has an opto-isolated open-collector output, a pull-up resistor must be added to the device output. In Figure 10, a 3-phase wattmeter has three opto-isolated open-collector outputs, each requiring an external pull-up resistor. Since each of the opto-isolators is rated for 60 VDC, install a 100 kΩ pull-up resistor between each output and the +24 V loop supply. The common of the opto-isolators connects to the common of the BAS Remote. Since each BAS Remote input has a 100 kΩ input impedance, the resulting voltage divider sets the off-state voltage to 12 V. Even though the BAS Remote input range is 0–10 VDC, this will not harm it. Set the two thresholds to 2 V and 3 V. The threshold points on digital signals are not critical. Consult the pulse device manual for more guidance.

5.5 Analog Input

An analog input can measure voltage in the range of 0–5 VDC or 0–10 VDC or it can measure current in the range of 0–20 mA. Transmitters that produce an elevated “zero” such as 1–5 VDC, 2–10 VDC or 4–20 mA can be measured as well. Using the web page, configure the input for either voltage or current and select an appropriate range. Scaling the input is accomplished by assigning the low and high points to engineering units. When set as a voltage input, the input impedance is 100 kΩ and for a current input, the impedance is 250 Ω.

With voltage measurement, connect the more positive voltage to point *B* and the less positive to BAS Remote common as shown in Figure 11. On three-wire devices such as damper actuators, the output signal is referenced to the damper’s power supply common. That common must be at the same reference as the BAS Remote common. Notice the connections in the diagram. In this situation it is only necessary to attach the transmitter output to point *A* on the BAS Remote input.

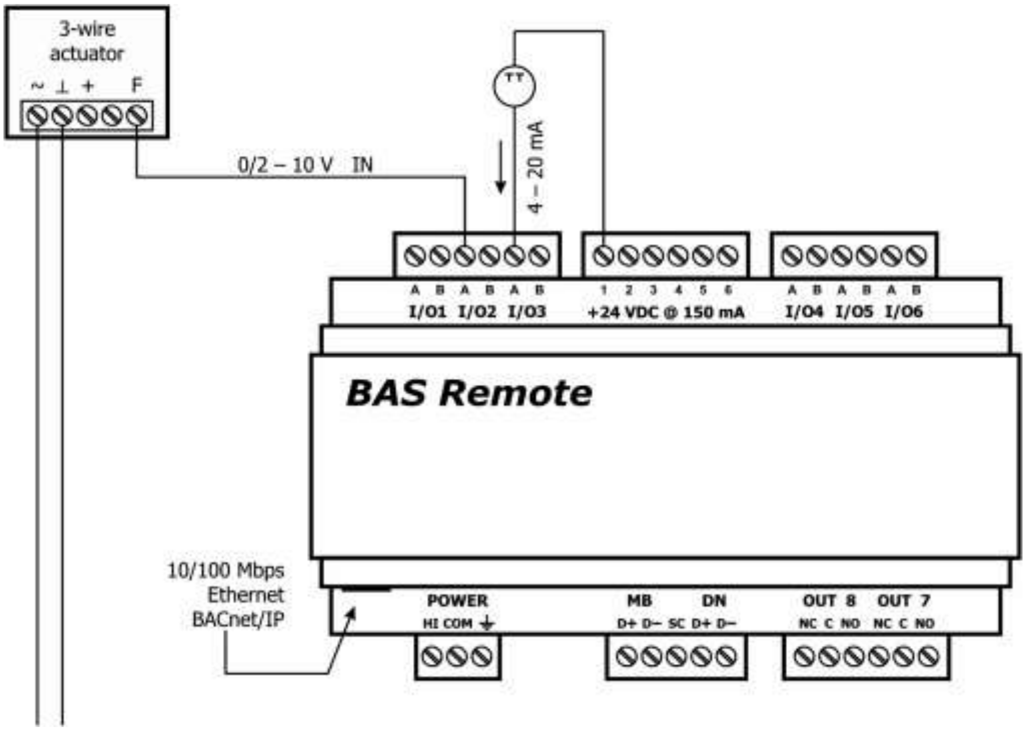


Figure 11 — Analog Input Connections

When measuring current, remember the BAS Remote sinks current to ground. A 250 Ω impedance is effectively applied between points *A* and *B* on the input. To measure current, it must be driven into point *A* with respect to point *B*. For two-wire current transmitters, the more positive point on the transmitter attaches to the +24 V on the BAS Remote loop supply or it can attached to an external loop supply as long as that loop supply has a common connection with the BAS Remote. The less positive connection is made to point *A* on the input.

Care should be exercised when connecting to a three-wire current transmitter. These are usually non-isolated devices between the power source and signal output. The BAS Remote will sink current from its input to ground so the transmitter must source current from a positive potential to ground. If the three-wire transmitter works in this manner, it can be accommodated.

Four-wire transmitters usually have isolation between power supply and signal output so their output stage can usually be treated as a two-wire transmitter.

5.6 Analog Output

Either voltage in the range of 0–10 VDC or current in the range of 0–20 mA can be outputted by assigning analog outputs. Configure an output using a web page. Select the appropriate range. For DC voltage, the output voltage is applied to point A with respect to common. For DC current, the output current is sourced from point A to common so there is no need for a loop supply. A current output can source up to 20 mA into a resistive load not exceeding 750 Ω. Verify the burden that a current output device will present. The BAS Remote cannot generate enough voltage to drive loads with higher resistance.

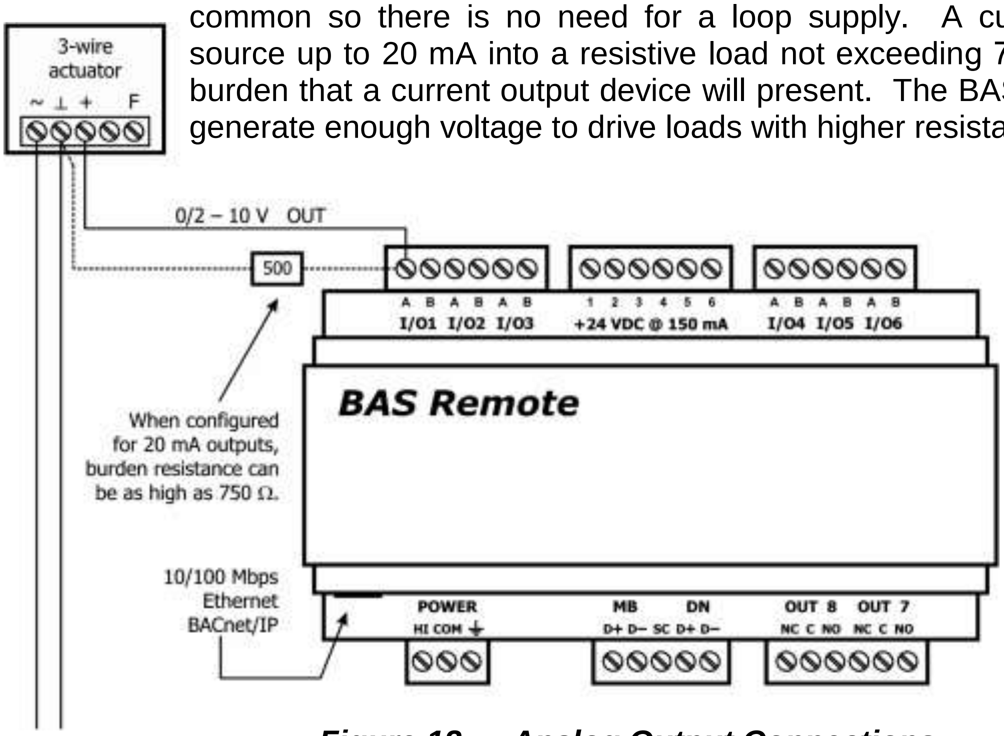


Figure 12 — Analog Output Connections

Figure 12 illustrates connections to a three-wire damper actuator. The damper requires a 0–10 V command signal which can easily be accomplished by the BAS Remote. However, if a current output is desired it is possible to set the BAS Remote analog output to 4–20 mA and install an external 500 Ω resistor that will convert the 4–20 mA signal to 2–10 V.

6 Operation

6.1 General Considerations

Configuration is accomplished while the unit is connected to a computer running a web browser (Java-enabled) that accesses the unit's built-in web server.

6.1.1 Ethernet Port on the Master Module

Auto-Negotiation

The Ethernet port on the **BAS Remote Master** unit offers full auto-negotiation. A single cable links two Ethernet devices. When these devices auto-negotiate, the data rate will be 100 Mbps only if both are capable of that speed. Likewise, full-duplex will only be selected if both can support it. If only one device supports auto-negotiation, then it will default to half-duplex mode and match the data rate of the non-auto-negotiating device.

Auto-MDIX (Auto-Crossover)

The Ethernet port offers Auto-MDIX. When interconnecting two Ethernet devices, a straight-through cable or crossover cable can be used — but if one device uses Auto-MDIX, the cable wiring does not matter; Auto-MDIX adjusts for either type.

Reset Switch

To reset the Master module to its default values of the IP address (192.168.92.68), gateway address (192.168.92.1) and netmask (/24 or 255.255.255.0), use a paperclip or similar tool to press the reset switch (see Figure 1 for location) while the unit is repowered and until the reboot is complete. Once the *Status* LED stops flashing, release the reset switch. The unit will now use the default values.

6.1.2 LEDs

To aid in troubleshooting, several LEDs have been provided.

The **Status LED** flashes green during boot up — then glows solid green while operation is fault-free. If a fault occurs, the LED glows solid red. On the Master module, this LED flashing green (after boot up) indicates Modbus serial activity.

The Master module has an **Ethernet LED** that glows green when properly linked to equipment operating at 100 Mbps (yellow for 10 Mbps) and indicates activity by flashing.

The Expansion module has a **Network LED** that flashes green to indicate data transfers.

I/O LEDs 1–8 follow the behaviour described in the chart below **:**

If the I/O channel is …	Green indicates …	Red indicates …
a **Relay** output	the coil is energized.	*(not used for relay output.)*
an **Analog** output	the command is greater than zero.	10% deviation from command
a **Contact** input	the contact is made.	*(not used for contact input.)*
a **Pulse** input	the input state changed.	*(not used for pulse input.)*
a **Thermistor**	current flow is detected	No current flow detected
an **Analog** input	the signal is greater than 1% of span.	*(not used for analog input.)*

6.1.3 Accessing and Configuring the Web Server on the Master Unit

6.1.3.1 Web Browser

The Master contains an interactive web server, accessible from any Internet-compatible PC on the local network. It is compatible with recent versions of Internet Explorer (5.0 or later, suggested) or Netscape Navigator (7.1 or later, required). It is factory-programmed with a default IP address of 192.168.92.68 and a Class C subnet mask of 255.255.255.0. Once configured, changing the BAS Remote IP address is strongly encouraged.

6.1.3.2 Initial Access

The hardware arrangement for initially setting the Master IP address appears in Figure 13. The PC should be temporarily disconnected from the Ethernet LAN in case the Master's default address matches that of a device on the existing LAN. The procedure for altering the IP address creates a ***temporary LAN*** composed of nothing but the Master, the PC used to configure it and a CAT5 cable connecting the two. Since the Master supports Auto-MDIX, either straight-through or crossover cable can be used.

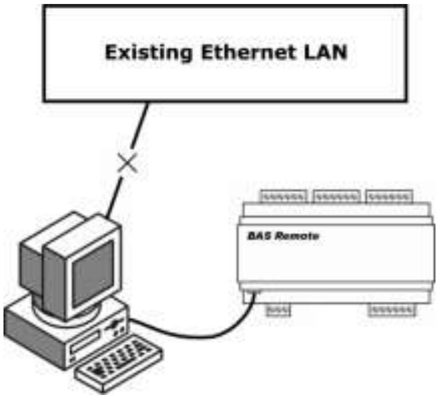
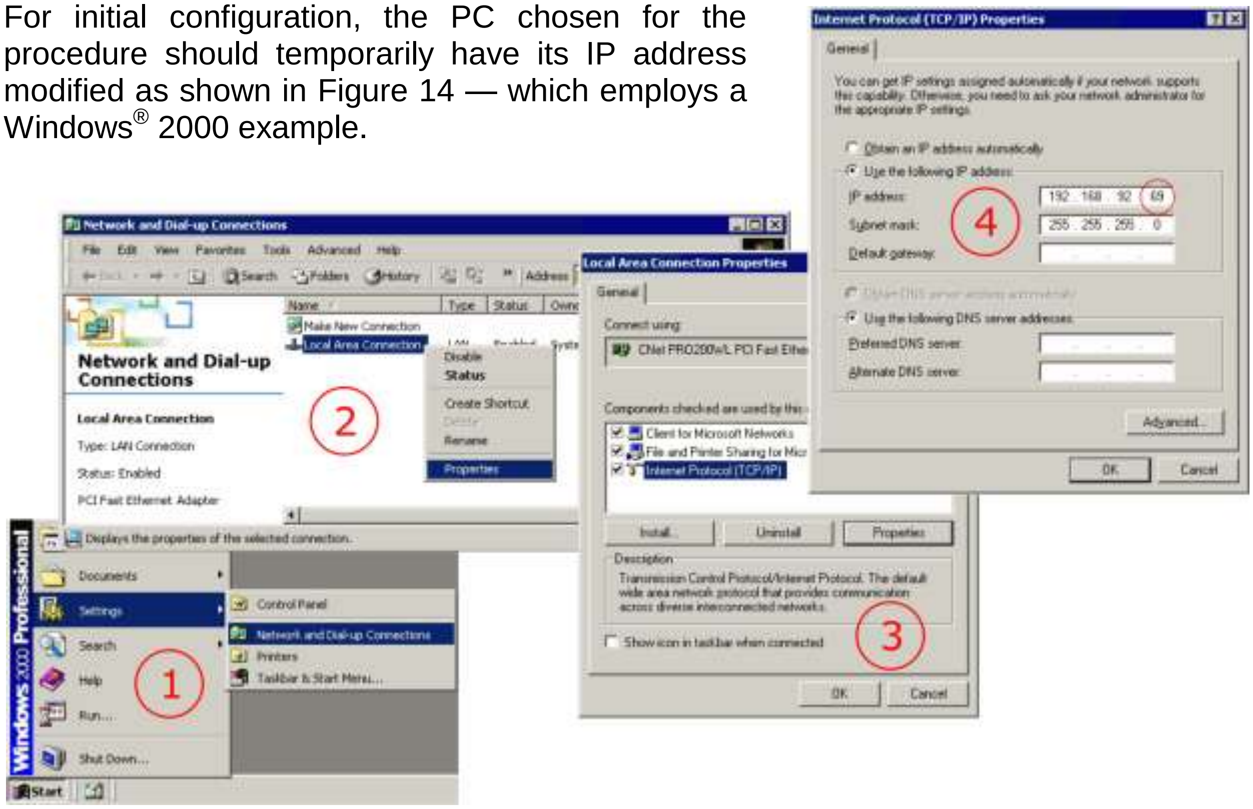


Figure 13 — Setup for Initial IP Address Configuration by Web Browser

For initial configuration, the PC chosen for the procedure should temporarily have its IP address modified as shown in Figure 14 — which employs a Windows® 2000 example.



Network and Dial-up Connections

Local Area Connection

Type: LAN Connection

Status: Enabled

PCI Fast Ethernet Adapter

Local Area Connection — LAN — Enabled — System

Displays the properties of the selected connection.

Local Area Connection Properties

Connect using:

CNet PRO200WL PCI Fast Ethe

Components checked are used by this

☑ Client for Microsoft Networks

☑ File and Printer Sharing for Micr

☑ Internet Protocol (TCP/IP)

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

□ Show icon in taskbar when connected

Internet Protocol (TCP/IP) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 92 . 69

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

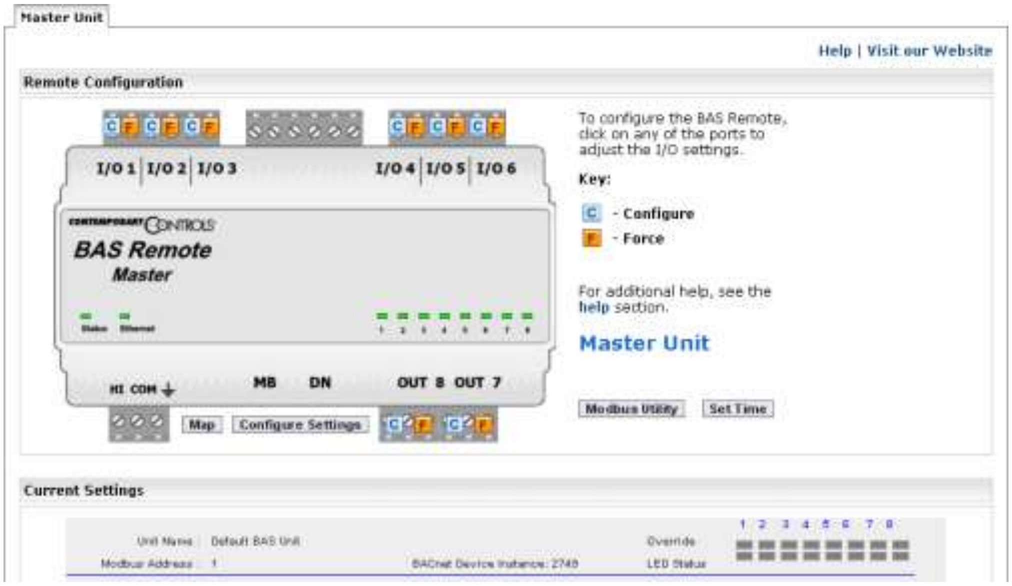
Preferred DNS server: . . .

Alternate DNS server: . . .

Figure 14 — Steps for Changing the IP Address of the PC Used for Setup

Figure 14 suggests an IP address for the PC of 192.168.92.**69**, but the final quad of the address could be any value from 1 to 254 — **except for 68** which is used by the Master. After the IP address of the PC has been set to the same LAN as the Master, a browser can access the Master's default IP address. The Master does **not** support DHCP.

Figure 15 displays just the *relevant upper portion* of the screen that appears when you access the Master. Just beneath the device image is a link named "Configure Settings". Clicking this link opens another window where you can configure the values discussed in the next four sections (see Figures 17–20).

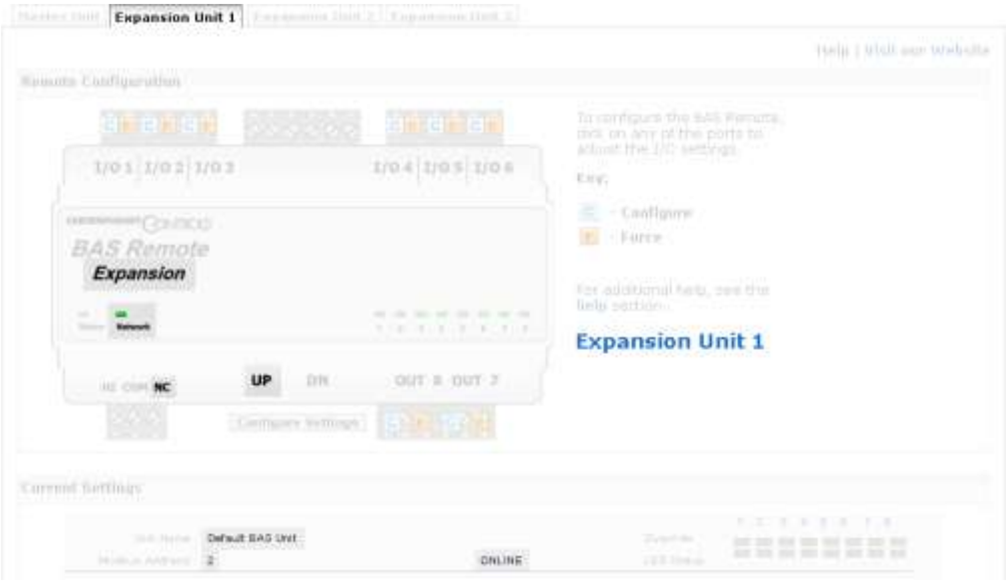


To configure the BAS Remote, click on any of the ports to adjust the I/O settings.

For additional help, see the help section.

Unit Name : Default BAS Unit
Modbus Address : 1
BACnet Device Instance: 2748
Override
LED Status

Figure 15 — Master Main Page (Partial View)



To configure the BAS Remote, click on any of the ports to adjust the I/O settings.

For additional help, see the help section.

Unit Name : Default BAS Unit
Modbus Address : 2
ONLINE
Override
LED Status

Figure 16 — Expansion Main Screen (Partial View)

Figure 16 displays a view of a main screen (for one of three Expansion units) that is faded — ***except*** for those elements that ***differ*** from the Master screen.

6.1.3.3 System Configuration

Four tabs exist on the configuration screen. Select System if it is not already highlighted.

System Name Give your system any name you wish.

IP Address Changing the default value of 192.168.92.68 is recommended.

Subnet Mask The default value of 255.255.255.0 is adequate for most users.

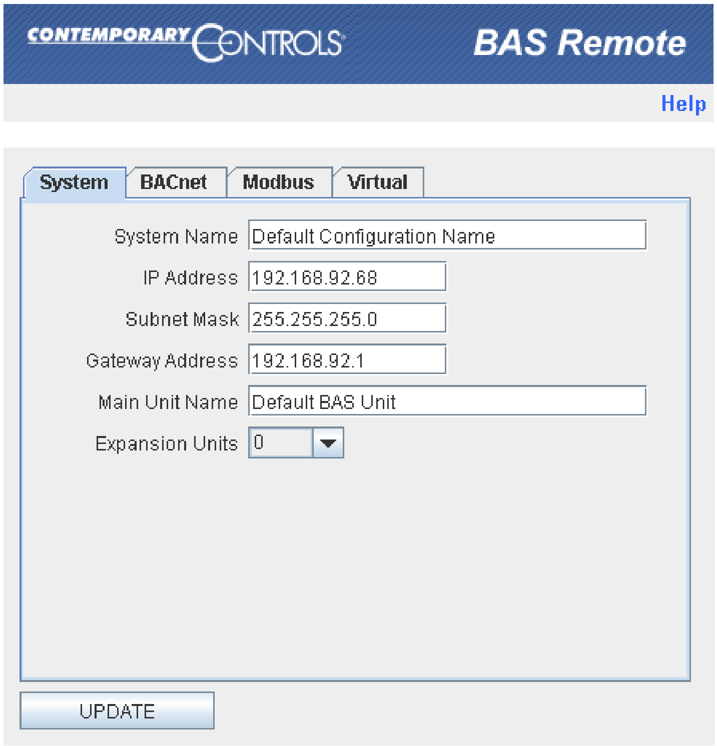
Gateway Address If your Ethernet LAN has a gateway (router) enter its address here.

Main Unit Name Give your Master module any name you wish.

Expansion Units Choose from the default of 0 to as many as 3 units. This value will set the number of tabs that you see atop the main screen.

NOTE: Whether you are configuring *System, BACnet, Modbus or Virtual* values, clicking "UPDATE" will **save** your values to internal memory, but you must **reboot** the Master module by cycling power before the new values will **apply**!

After the Master has been given its initial configuration, it will be ready for use in the full original Ethernet network. The temporary Ethernet network constructed in Figure 13 should be dismantled and the PC re-configured to restore its original IP address.



System Name: Default Configuration Name
IP Address: 192.168.92.68
Subnet Mask: 255.255.255.0
Gateway Address: 192.168.92.1
Main Unit Name: Default BAS Unit
Expansion Units: 0

Figure 17 — Configuring System in the Master Module

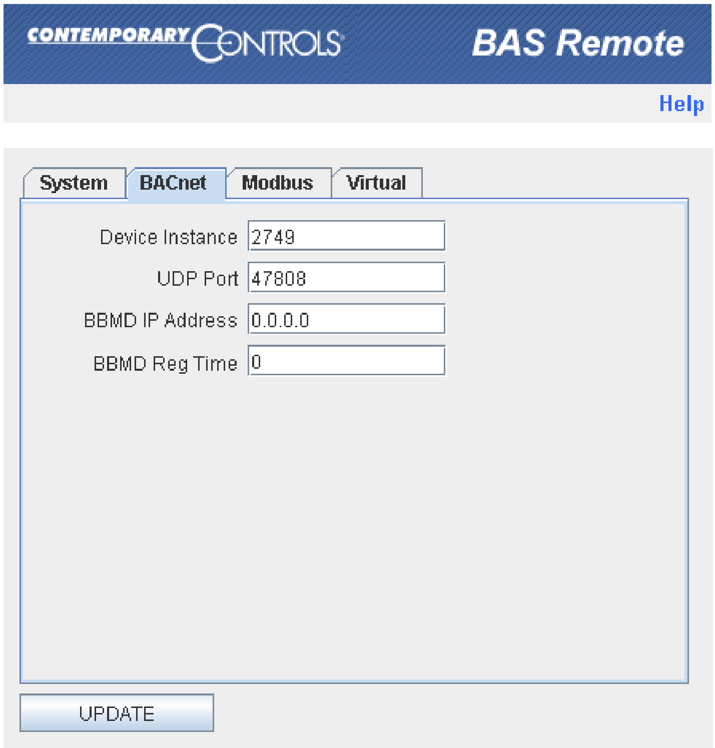
6.1.3.4 BACnet Configuration

Device Instance This is a 22-bit value (0–4,194,303). Each BACnet network device must have a unique device instance. The default value is 2749.

UDP Port The default value of 0xBAC0 should usually not be changed.

BBMD IP Address Enter the address of the BBMD with which the BAS Remote will perform Foreign Device Registration (FDR) if the BBMD is not located on the same subnet as the BAS Remote.

BBMD Reg Time Enter the interval (in seconds) between successive foreign device registrations.



Device Instance 2749
UDP Port 47808
BBMD IP Address 0.0.0.0
BBMD Reg Time 0

Figure 18 — Configuring BACnet in the Master Module

6.1.3.5 Modbus Configuration

Modbus configuration (see Figure 19) is only required if Modbus serial devices are attached to the MB bus. Communication settings between the Master module and the Expansion modules are preset and cannot be changed.

Modbus Address

Address 1 is pre-assigned to the Master module while Expansion modules are assigned addresses 2, 3 and 4 assuming Expansion units exist. The actual Expansion address depends upon the order of cabling to the Master. In addition to address 1, the Master module is also pre-assigned addresses 8 and 9 for internal functions. It is best to assume that addresses 1 through 9 are reserved for BAS Remote equipment so assign addresses from 10 to 247 to attached Modbus devices.

Baudrate

All Modbus devices attached to the MB bus must have the same baud rate. Baud rates range from 2,400 to 115,200. The default is 19,200.

Protocol

All Modbus devices attached to the MB bus must use the same protocol. The choices are RTU and ASCII. The default is RTU.

Parity

All Modus devices attached to the MB bus must use the same parity. The choices are NONE, ODD or EVEN. The default is EVEN.

Command Timeout

Set how long the Master will wait for a slave to respond. The range is 50–3000 ms with a default of 1,000. If a device fails to respond, it is placed into an offline queue. Devices in this queue are checked every offline poll period.

Inter Scan Delay

Set the desired delay between each poll cycle from 100-30,000 ms. The default is 2,000.

Offline Poll Period

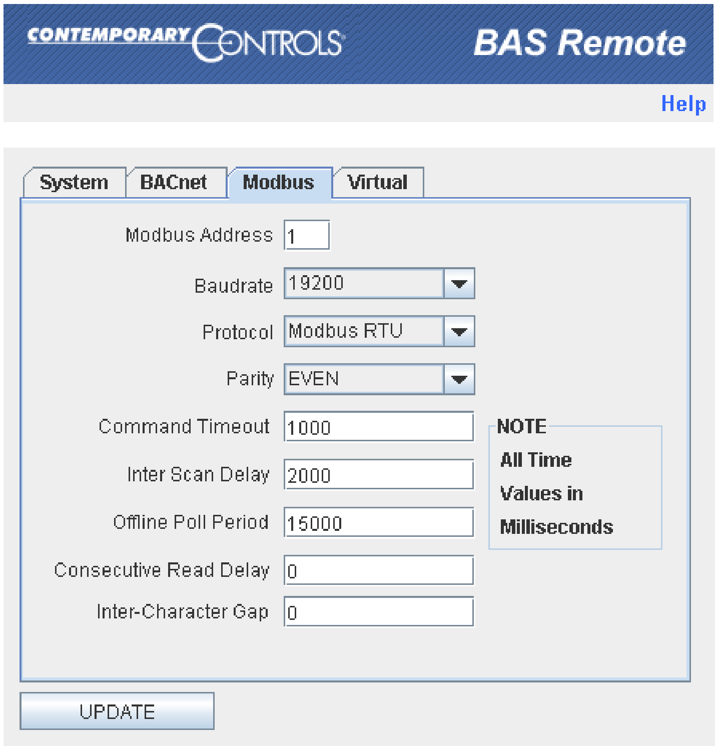
Set how often the Master will check to see if a slave device is back online. The range is 2,000-30,000 ms. The default is 15,000.

Consecutive Read Delay

Sets the delay between back-to-back accesses of the same slave device. The range is 0-100 ms and the default is 0.

Inter-Character Gap

This maximum (1.5 character times) is what Modbus allows between characters. If a device exceeds the allowable time, set this to the highest possible value for the device. The default of 0 is recommended, but settings between 0 and 10 ms are possible.



Modbus Address 1

Baudrate 19200

Protocol Modbus RTU

Parity EVEN

Command Timeout 1000

Inter Scan Delay 2000

Offline Poll Period 15000

Consecutive Read Delay 0

Inter-Character Gap 0

NOTE

All Time Values in Milliseconds

__Figure 19 — Configuring Modbus in the Master Module__

6.1.3.6 Virtual Point Configuration

The BAS Remote can store data independent of I/O Channel values. These stored values are called *Virtual Points* because they are not required to be related to physical I/O points. These virtual points can be treated as Modbus registers, BACnet objects or Sedona Framework components. See Section 7.3.2 for more information.

Instance This read-only number is created from the *BACnet Type*. It is of the form ***8xyyyy***, where the ***x*** defaults to ***4*** (register) but after the *Apply Object Changes* button is clicked it will become ***0*** if the Type is *Binary Output* or ***1*** if the Type is *Binary Input*. The value of ***yyyy*** is automatically incremented as objects are added to the list of Virtual Points in the left side of Figure 20. This virtual object can be written or read via its Modbus/TCP address. Its address will be 8 and the register number will be YYYY. The binary input/output can be written/read via 0YYY or 1YYY.

Object Name Give the Virtual Point any name you wish. This will be the BACnet object name for this virtual object.

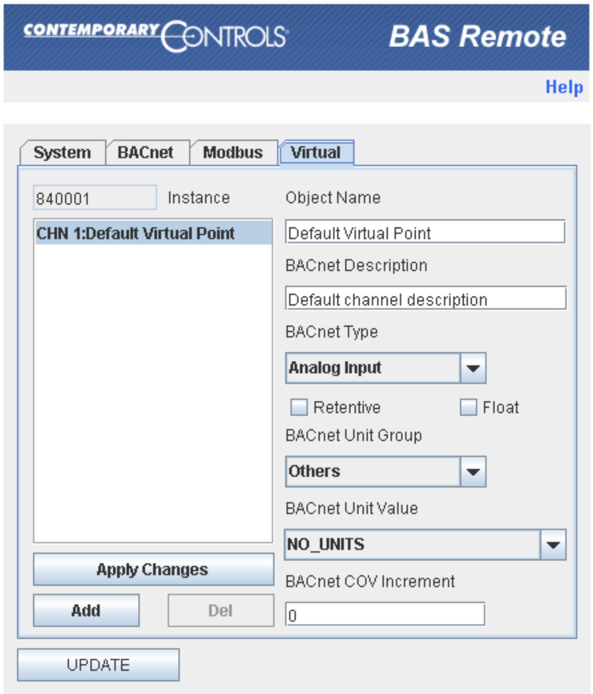
BACnet Description Give the BACnet channel any name you wish.

BACnet Type Select the Type from the drop-down menu. Below, check the “Retentive” box to preserve counts despite a power failure. “Float” indicates how this virtual object will be accessed via Modbus/TCP. If checked, this object will be accessed as a 32-bit float value. If unchecked, it will be accessed as a 16-bit register. **NOTE:** Analog Value, Analog Input and Binary Input types will not be writeable via BACnet. Analog output and Binary Input can be written via BACnet.

BACnet Unit Group Select the Unit Group from the drop-down menu.

BACnet Unit Value Select the Unit Value from the drop-down menu.

BACnet COV Increment Once the channel value changes by this amount (up or down), a COV message is sent to subscribers.



Instance: 840001

CHN 1:Default Virtual Point

Object Name: Default Virtual Point

BACnet Description: Default channel description

BACnet Type: Analog Input

□ Retentive □ Float

BACnet Unit Group: Others

BACnet Unit Value: NO_UNITS

BACnet COV Increment: 0

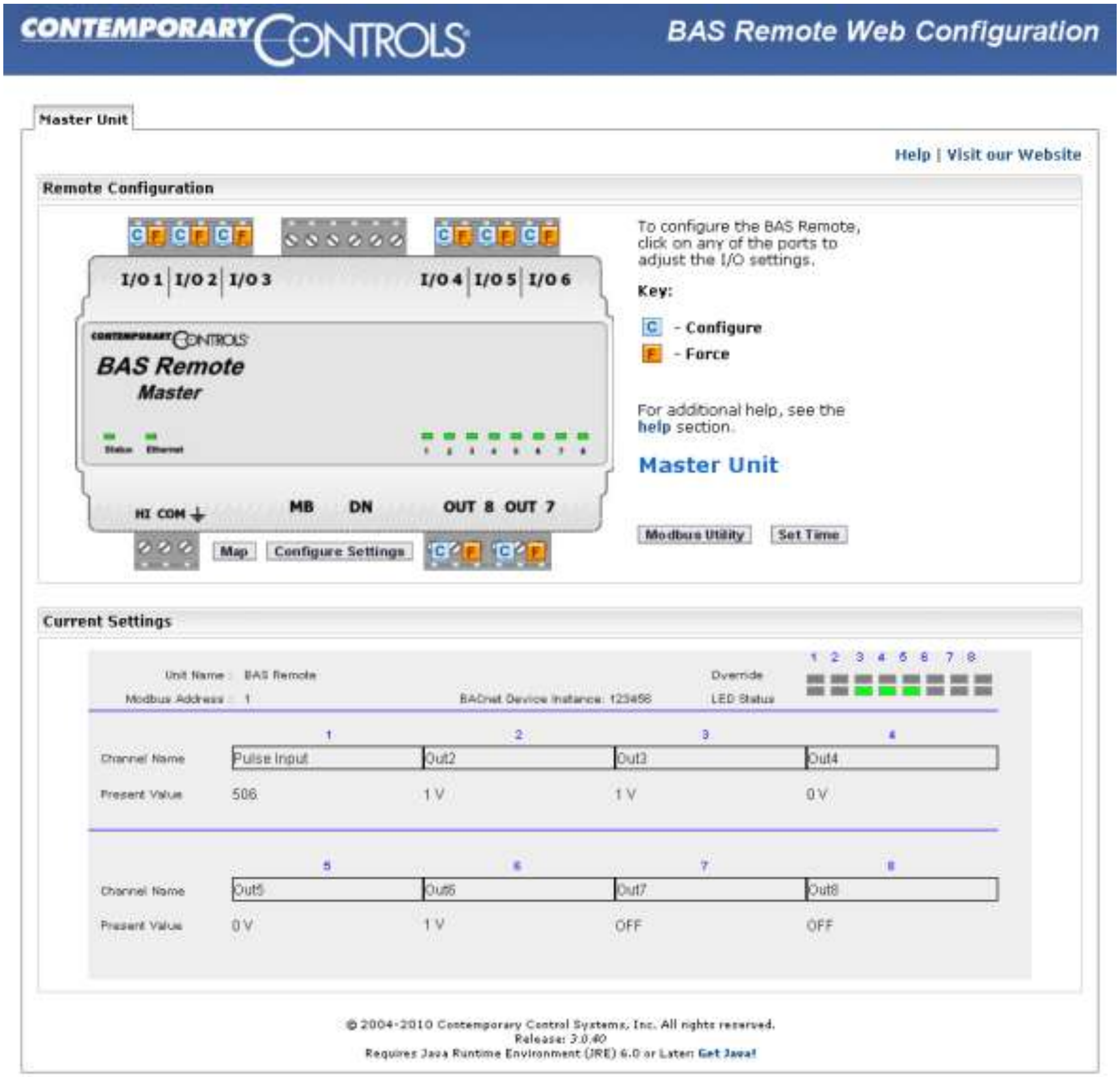
Figure 20 — Configuring Virtual Points in the Master Module

6.1.4 Web Server Screen Overview

From the **Web Server Screen** (Figure 21) you can configure all I/O channels, view their status or force them as part of a commissioning process. (The figure uses the Master unit as an example, but the Expansion unit appears very similar and functions the same.) Immediately beneath the right side of the banner, the following links are displayed:

Help displays a new window (Figure 23) with context-sensitive information.

Visit Our Website links to our WWW home page for further helpful information.



To configure the BAS Remote, click on any of the ports to adjust the I/O settings.

For additional help, see the help section.

Unit Name : BAS Remote

Modbus Address : 1

BACnet Device Instance: 123456

	1	2	3	4
Channel Name	Pulse Input	Out2	Out3	Out4
Present Value	506	1 V	1 V	0 V

	5	6	7	8
Channel Name	Out5	Out6	Out7	Out8
Present Value	0 V	1 V	OFF	OFF

© 2004-2010 Contemporary Control Systems, Inc. All rights reserved.
Release: *3.0.40*
Requires Java Runtime Environment (JRE) 6.0 or Later: Get Java!

Figure 21 — Web Server Page

The *Configure Settings* button (just below the device image) provides access to the basic device configuration fields already discussed (Figure 17).

The BAS Remote image includes graphic representations of each I/O Channel. Each of the 8 channels has two icons (C and F) that open additional windows used for *configuring* or *forcing* each Channel. The use of these links is discussed in Section 6.3.

Other buttons located beneath and to the right of the device image are:

Map	This accesses the **Modbus Mapping** function (Section 7.5) which shows the status of all mapped objects — including mapped Modbus variables, virtual objects and all points within the BAS Remote.
Modbus Utility	This accesses the **Modbus Utility** function (Section 7.6).
Set Time	This accesses the Set Time window where you can manually set the values of time and date.

The large grey section at the bottom of the screen labelled *Current Settings* is the device "Monitor" — a read-only display of information for the BAS Remote module currently selected. The tab in **bold face** near the top of the screen indicates the module selected. Figure 22 below is an example in which Expansion Unit 2 has been selected.

NOTE: The number of tabs displayed is determined by the number of Expansion Units selected in the Master Configuration Screen (Figure 17).

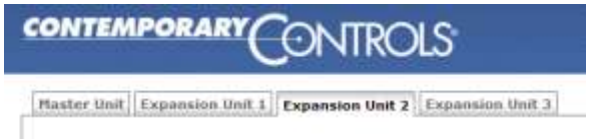


Figure 22 — The Tab in Bold Face Indicates the Selected Module

In the upper-left portion of the Monitor shown in Figure 21, two values (*Unit Name* and *Modbus Address*) are displayed for the currently selected module.

To the right of the Modbus Address value is a box which reports one of two possible pieces of information — depending on the type of BAS Remote module currently selected. When the Master is selected, the box will display the *BACnet Device Instance* of the Master. If an Expansion module is selected, the box will report the module's status: It will be *ONLINE* if its connection to the Master is valid or *OFFLINE* if the connection is invalid. An *OFFLINE* report usually means the Expansion module cannot communicate with the Master due to a cabling issue.

In the upper-right portion of the Monitor, the status of each channel (point) is reported with the *Override* indicators and the *LED Status* indicators. All of these LEDs are refreshed at the rate of once per second.

The lower portion of the Monitor displays the *Channel Names* and their *Present Values*.

6.1.5 On-Screen Help

There are several screens used in the BAS Remote. The upper-right portion of each screen displays a ***Help*** option. Clicking this option launches another browser window with helpful information about the current screen.

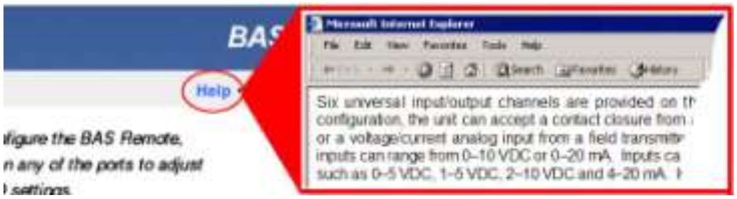


figure the BAS Remote,
n any of the ports to adjust
l settings.

Six universal input/output channels are provided on th
configuration, the unit can accept a contact closure from
or a voltage/current analog input from a field transmitte
inputs can range from 0–10 VDC or 0–20 mA. Inputs ca
such as 0–5 VDC, 1–5 VDC, 2–10 VDC and 4–20 mA.

Figure 23 — Help Window

6.2 Input/Output Channels (I/O)

6.2.1 Universal I/Os

Six identical universal I/O channels allow any mix of inputs and outputs requiring only two connections labelled **A** and **B**. Pin **A** is always more positive than pin **B**. Channels are labelled *I/O 1* through *I/O 6* and are divided equally between two six-pin terminal blocks. One terminal block is for I/O 1–3 and the other is for I/O 4–6. Configuring is done via the built-in web server. Input resolution is 10 bits; output resolution is 12 bits.

Schottky barrier diodes protect the electronics from over-voltage faults on inputs **A** and **B**. Also, a PPTC (***p***olymeric ***p***ositive ***t***emperature ***c***oefficient) resettable fuse protects terminal **B** from over-voltage when driven to ground.

Pin **A** can be an input or output. Pin **B** can be an input or ground. A D/A converter is used for generating analog voltage or current outputs, and for providing excitation current for contact closure sensing and for thermistor measurement.

In current output mode, pin **B** is grounded. The output burden applied to pins **A** and **B** can range from 0 to 750 Ω. Since the internal burden is 250 Ω, the output voltage at pin **A** can range from 0–20 volts when driving 20 mA.

When measuring input voltages, pin **A** receives input while pin **B** is held at ground. Any DC voltage in the range of 0–5 V or 0–10 V can be measured.

When measuring current, pin **B** is unused and the input on pin **A** sees a 250 Ω load.

To sense contact closure, 2 mA is output at pin **A** while pin **B** is grounded. Then the voltage at pin **A** is measured. Any value below 0.3 volts (150 Ω) is considered a closed circuit. Dry-contact or solid-state switches being sensed must withstand an open-source voltage of 24 VDC and a current of 2 mA. For solid-state switches, the most positive connection is at pin **A** and a saturation voltage under 0.3 V is required.

A 10 kΩ thermistor is applied between pins **A** and **B**. Thermistors are non-linear heat sensing devices with a negative temperature coefficient of resistance. At nominal room temperature (77 °F), the resistance of a Type II or Type III thermistor is 10 kΩ. Both have curves with an accuracy of ±0.36 °F from 32 °F to 158 °F. Because higher resistance thermistors (such as 10 kΩ) introduce error due to the self-heating effect, lower thermistor current is used — thus minimizing self-heat and measurement inaccuracy.

6.2.2 Relay Outputs

There are two independent SPDT relay outputs. For each output, both the **NO** and **NC** contacts are brought out to a six-pin terminal block. Contacts are rated at *2A* at *30* V*AC* and *2A* at *30 VDC.* Wiring to the BAS Remote should only be Class 2. To control higher voltages, the safer approach is to connect the coils of Class 2 interposing relays to the contacts of the BAS Remote and have the contacts of the interposing relays connect to the Class 1 circuits. These interposing relays should be further from the BAS Remote and closer to the Class 1 equipment.

6.3 Channel Configuring

To configure a channel, access the Web Server Page, click on the C icon for the channel of interest and make adjustments in the new screen that appears. Your selected channel is confirmed by the large number on the left side of the new screen. As Figure 24 shows, clicking on the *Save* button is confirmed by the button briefly changing from light blue to dark blue. If you attempt to set an illegal value, the button will not change colour.

6.3.1 Analog Voltage Input Configuring

You can define any channel 1–6 as "*INPUT: 0–10V Analog*" or "*INPUT: 0–5V Analog*" (As an example, Figure 24 uses Channel 1 and 0–10V). Such a channel can accept an input voltage in the range of 0–10 volts or 0–5 volts. The channel BACnet type will be Analog Input. On this screen, you can adjust these parameters:

BACnet Unit Group The *Electrical* default can be set to any option in the list.

Channel Name You can rename the channel using no more than 63 characters. This will be the object name for this channel (followed by the channel number). For example, "Analog Input-1".

BACnet Unit Group The *Electrical* default can be set to any option in the list.

BACnet Unit Value The *VOLTS* default can be set to any option in the list. The available options depend on the BACnet Unit Group you specify.

BACnet COV Increment Once the channel value changes by this amount (up or down), a COV message is sent to subscribers.

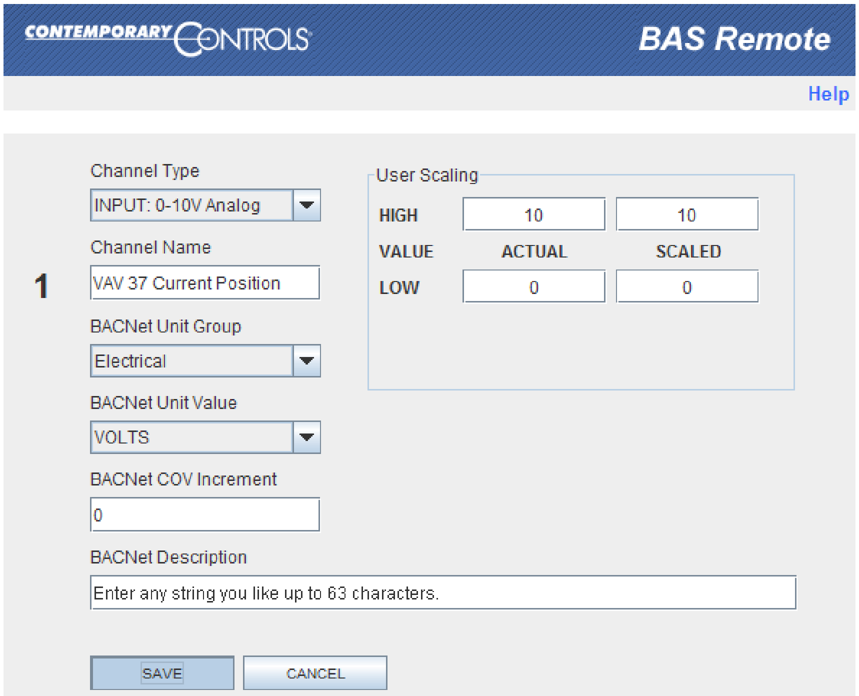
BACnet Description You can enter any string using no more than 63 characters.

ACTUAL HIGH This specifies the highest value within the range.

ACTUAL LOW This specifies the lowest value within the range.

SCALED HIGH You can set a physical value corresponding to the high value.

SCALED LOW You can set a physical value corresponding to the low value.



Channel Type: INPUT: 0-10V Analog

Channel Name: VAV 37 Current Position

BACNet Unit Group: Electrical

BACNet Unit Value: VOLTS

BACNet COV Increment: 0

BACNet Description: Enter any string you like up to 63 characters.

User Scaling

VALUE	ACTUAL	SCALED
HIGH	10	10
LOW	0	0

Figure 24 — Analog Input Configuration

6.3.2 Analog Voltage Output Configuring

You can set any channel 1–6 as type “*OUTPUT: 0–10V Analog*” as shown in Figure 25, using Channel 1 as an example. Such a channel can supply an output voltage in the range of 0–10 V. The channel BACnet type will be Analog Output. On this screen, you can adjust any of the following parameters.

Channel Name You can rename the channel using no more than 63 characters. This will be the object name for this channel (followed by the channel number). For example, “Analog Output-1”.

BACnet Unit Group The *Electrical* default can be set to any option in the list.

BACnet Unit Value The *VOLTS* default can be set to any option in the list. The available options depend on the BACnet Unit Group you specify.

BACnet COV Increment Once the channel value changes by this amount (up or down), a COV message is sent to subscribers.

BACnet Description You can enter any string using no more than 63 characters.

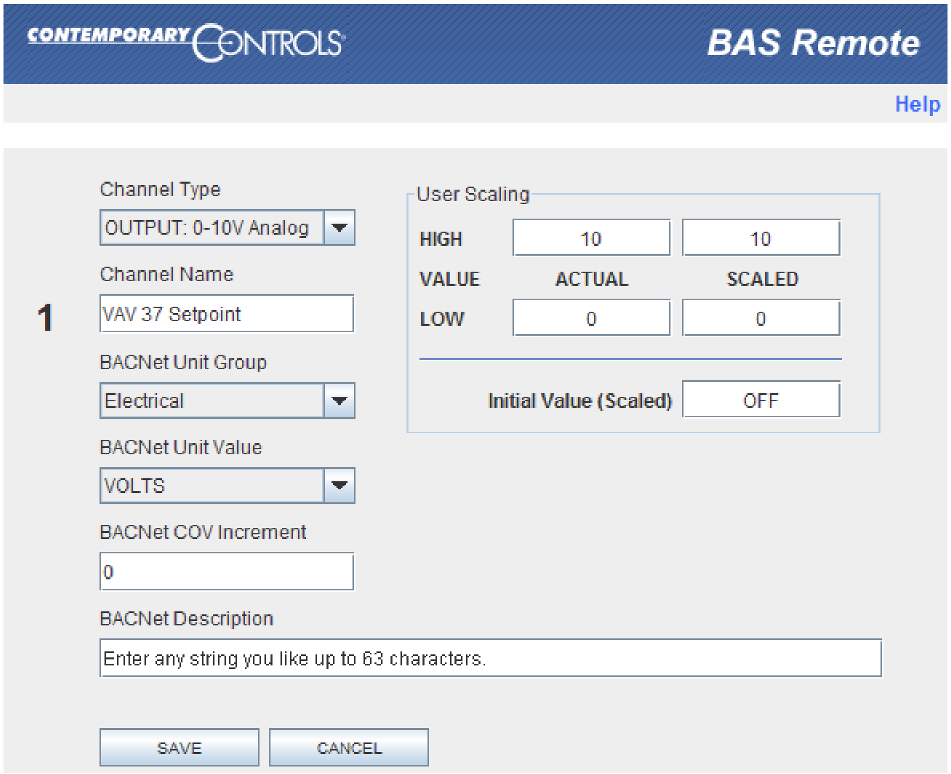
ACTUAL HIGH This specifies the highest value within the range.

ACTUAL LOW This specifies the lowest value within the range.

SCALED HIGH You can set a physical value corresponding to the high value.

SCALED LOW You can set a physical value corresponding to the low value.

Initial Value (Scaled) The *Off* default outputs 0, but you can preset a *Scaled* value.



Channel Type: OUTPUT: 0-10V Analog

Channel Name: VAV 37 Setpoint

BACNet Unit Group: Electrical

BACNet Unit Value: VOLTS

BACNet COV Increment: 0

BACNet Description: Enter any string you like up to 63 characters.

User Scaling

HIGH	10	10
VALUE	ACTUAL	SCALED
LOW	0	0

Initial Value (Scaled): OFF

Figure 25 — Analog Output Configuration

6.3.3 Binary Input Configuring

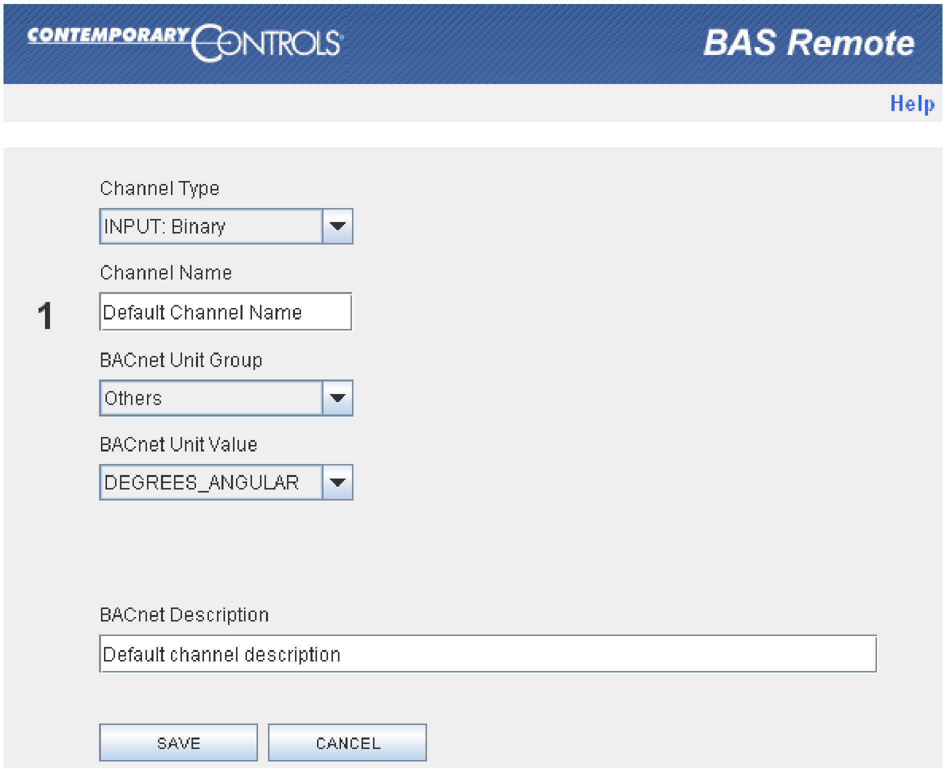
You can define any channel 1–6 as type “*INPUT: Binary*” as shown in Figure 26 which uses Channel 1 as an example. The channel BACnet type will be Binary Input. On this screen, you can adjust the following parameters.

Channel Name You can name the channel using no more than 63 characters. This will be the object name for this channel (followed by the channel number). For example, “Binary Input-1”.

BACnet Unit Group The *Others* default can be set to any option in the list.

BACnet Unit Value The *NO_UNITS* default can be set to any option in the list. The available options depend on the **BACnet Unit Group** you specify.

BACnet Description You can enter any string using no more than 63 characters.



1

Channel Type: INPUT: Binary

Channel Name: Default Channel Name

BACnet Unit Group: Others

BACnet Unit Value: DEGREES_ANGULAR

BACnet Description: Default channel description

Figure 26 — Binary Input Configuration

6.3.4 Current Input Configuring

You can define any channel 1–6 as type “*INPUT: 0–20 mA*“ as shown in Figure 27 which uses Channel 1 as an example. Such a channel accepts an input current in the range of 0–20 mA. The channel BACnet type will be Analog Input. On this screen, you can adjust any of the following parameters.

Channel Name You can rename the channel using no more than 63 characters. This will be the object name for this channel (followed by the channel number). For example, “Current Input-1”.

BACnet Unit Group The *Electrical* default can be set to any option in the list.

BACnet Unit Value The *MILLIAMPERES* default can be set to any option in the list. The available options depend on the BACnet Unit Group you specify.

BACnet COV Increment Once the channel value changes by this amount (up or down), a COV message is sent to subscribers.

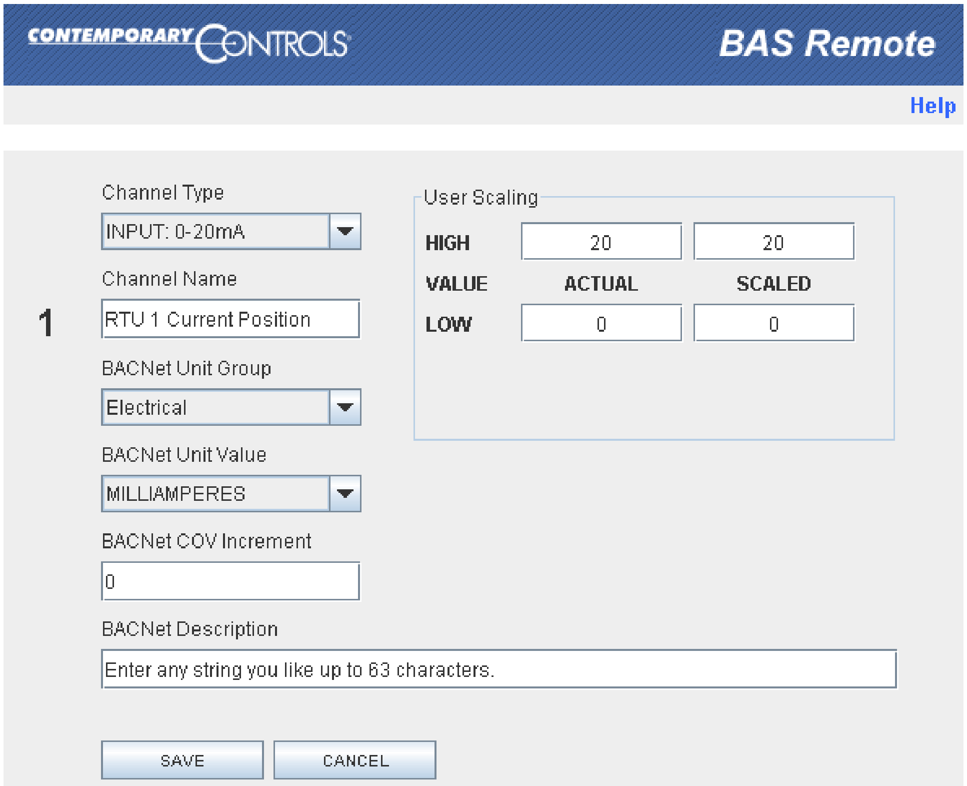
BACnet Description You can enter any string using no more than 63 characters.

ACTUAL HIGH This specifies the highest value within the range.

ACTUAL LOW This specifies the lowest value within the range.

SCALED HIGH You can set a physical value corresponding to the high value.

SCALED LOW You can set a physical value corresponding to the low value.



1

Channel Type: INPUT: 0-20mA

Channel Name: RTU 1 Current Position

BACNet Unit Group: Electrical

BACNet Unit Value: MILLIAMPERES

BACNet COV Increment: 0

BACNet Description: Enter any string you like up to 63 characters.

User Scaling

HIGH	20	20
VALUE	ACTUAL	SCALED
LOW	0	0

Figure 27 — Current Input Configuration

6.3.5 Current Output Configuring

You can define any channel 1–6 as type "*OUTPUT: 0–20 mA*" as shown in Figure 28 which uses Channel 1 as an example. Such a channel provides an output current in the range of 0–20 mA. The channel BACnet type will be Analog Output. On this screen, you can adjust any of the following parameters.

Channel Name You can rename the channel using no more than 63 characters. This will be the object name for this channel (followed by the channel number). For example, "Current Input-1".

BACnet Unit Group The *Electrical* default can be set to any option in the list.

BACnet Unit Value The *MILLIAMPERES* default can be set to any option in the list. The available options depend on the BACnet Unit Group you specify.

BACnet COV Increment Once the channel value changes by this amount (up or down), a COV message is sent to subscribers.

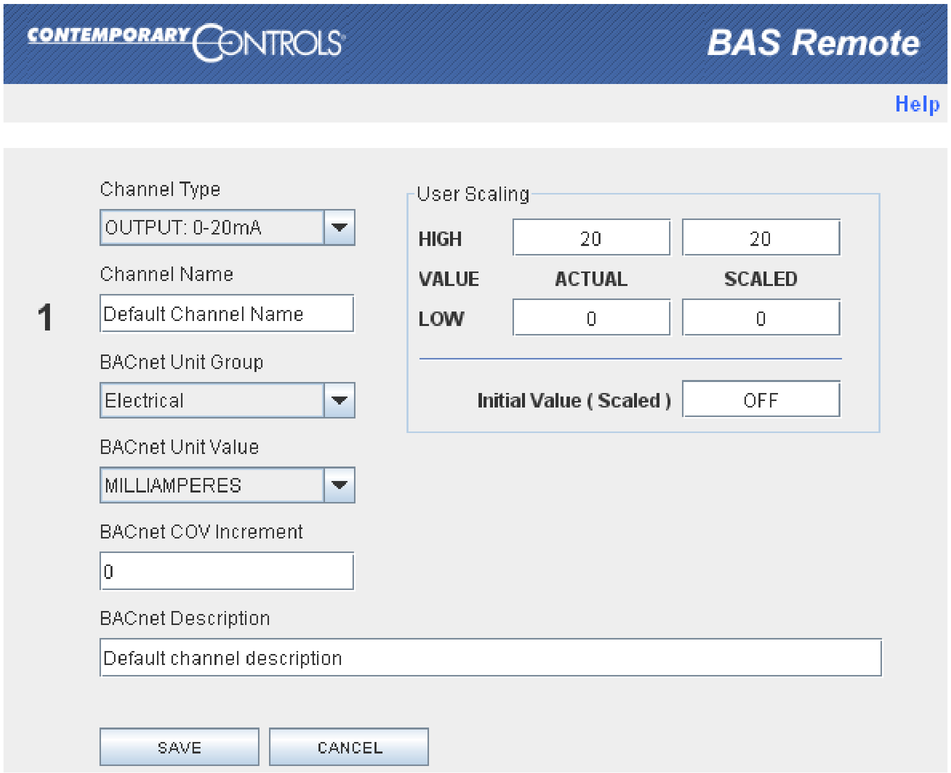
BACnet Description You can enter any string using no more than 63 characters.

ACTUAL HIGH This specifies the highest value within the range.

ACTUAL LOW This specifies the lowest value within the range.

SCALED HIGH You can set a physical value corresponding to the high value.

SCALED LOW You can set a physical value corresponding to the low value.



1

Channel Type: OUTPUT: 0-20mA

Channel Name: Default Channel Name

BACnet Unit Group: Electrical

BACnet Unit Value: MILLIAMPERES

BACnet COV Increment: 0

BACnet Description: Default channel description

User Scaling

HIGH	20	20
VALUE	ACTUAL	SCALED
LOW	0	0

Initial Value (Scaled): OFF

Figure 28 — Current Output Configuration

6.3.6 Thermistor Input Configuring

You can define any channel 1–6 as a Type II or III thermistor input as shown in Figure 29 which uses a Type 3 thermistor and Channel 1 as an example. The channel BACnet type will be Analog Input. On this screen, you can adjust any of the following parameters.

Channel Type You can select a Type II or Type III thermistor profile.

Channel Name You can name the channel using no more than 63 characters. This will be the object name for this channel (followed by the channel number). For example, “Thermistor Input-1”.

BACnet Unit Group The *Temperature* default can be set to any option in the list.

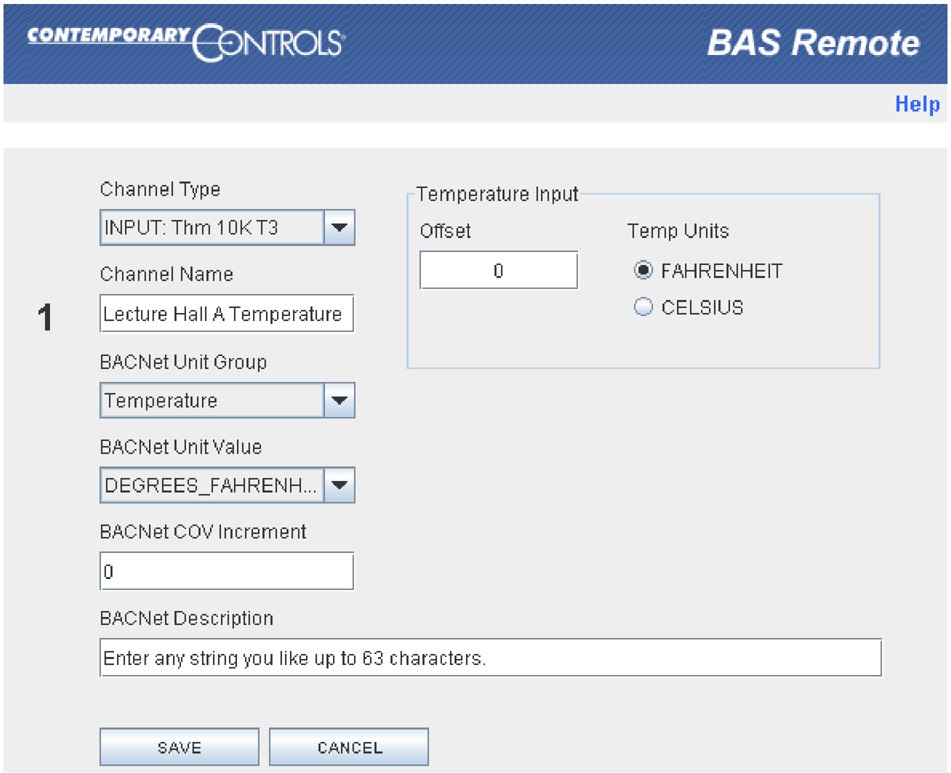
BACnet Unit Value Set the *DEGREES_FAHRENHEIT* default to any option in the list. Available options depend on the BACnet Unit Group you specify.

BACnet COV Increment Once the channel value changes by this amount (up or down), a COV message is sent to subscribers.

BACnet Description You can enter any string using no more than 63 characters.

Offset You can specify a (+) or (–) number of degrees offset — if, for example, the thermistor in use is known to be delivering an inaccurate reading. Either type of thermistor can normally measure from 40º and +110º Fahrenheit.

Temp Units The *Fahrenheit* default scale can be changed to *Celsius*.



1

Channel Type: INPUT: Thm 10K T3

Channel Name: Lecture Hall A Temperature

BACNet Unit Group: Temperature

BACNet Unit Value: DEGREES_FAHRENH...

BACNet COV Increment: 0

BACNet Description: Enter any string you like up to 63 characters.

Temperature Input

Offset: 0

Temp Units: ● FAHRENHEIT ○ CELSIUS

Figure 29 — Thermistor Input Configuration

6.3.7 Pulse Input Configuring

You can define any channel 1–6 as type “*INPUT: Pulse*” as shown in Figure 30 which uses Channel 1 as an example. Such a channel can accept a pulse train in the range of 0–40 Hz. The channel BACnet type will be Analog Value.

NOTE: The value reported by this channel via BACnet will be a 32-bit float as this is an Analog Value. To provide accurate values when counting input pulses, this value will rollover at 2^{24}–1.

To reset the pulse input value, you can (via BACnet) take the channel out of service and write a desired value — such as 0 — and then put the channel back in service to allow pulse counting to continue. This value can also be reset by writing a new value to it using one of the following four methods:

- via the map web page — displayed by clicking the *Map* button (Figure 15)
- via Modbus/TCP
- via Web Services
- via a custom webpage (which can write to the instance associated with this channel)

On the screen of Figure 30, you can adjust any of the following parameters.

Channel Name You can rename the channel using no more than 63 characters. This will be the object name for this channel (followed by the channel number). For example, “Pulse Input-1”.

BACnet Unit Group The *Others* default can be set to any option in the list.

BACnet Unit Value Set the *NO_UNITS* default to any option in the list. The available options depend on the BACnet Unit Group you specify.

BACnet COV Increment Once the channel value changes by this amount (up or down), a COV message is sent to subscribers.

BACnet Description You can enter any string using no more than 63 characters.

ACTUAL HIGH This specifies the highest value within the range.

ACTUAL LOW This specifies the lowest value within the range.

SCALED HIGH You can set a physical value corresponding to the high value.

SCALED LOW You can set a physical value corresponding to the low value.

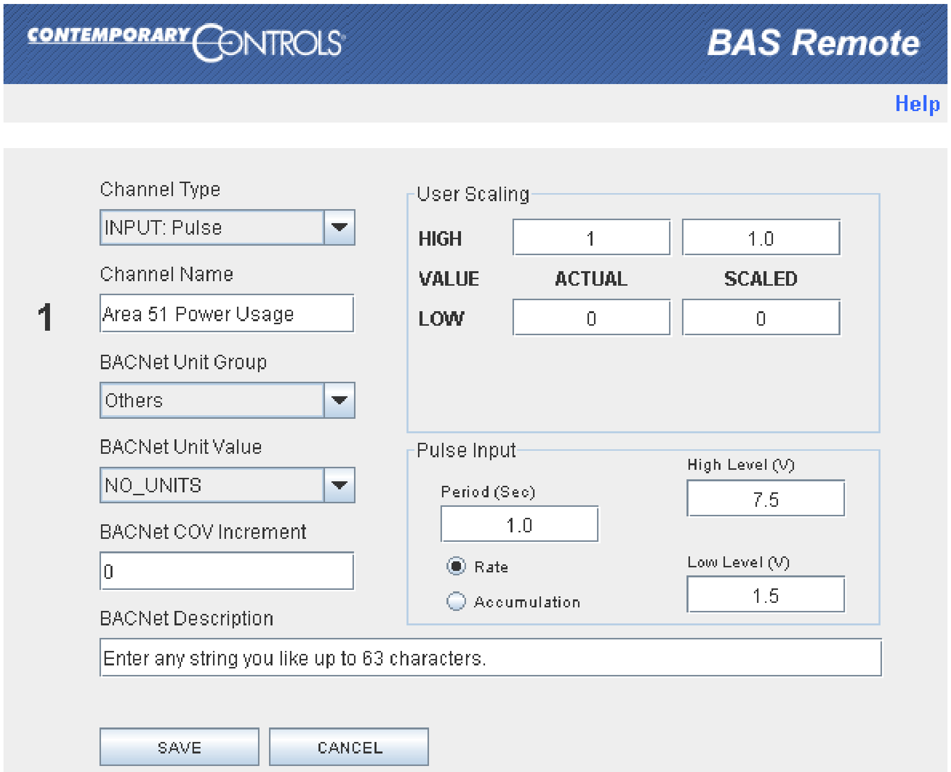
Period If the “Rate” option has been selected, this specifies the period in seconds — otherwise, this field is not present on the screen.

Rate This specifies that the rate of the input is being obtained.

Accumulate This specifies that the input pulses are being accumulated (absolute count) — with no limit to the time during which pulses are counted.

High Level (V) You can specify a value from 10 down to (but more than) the “Low Level”.

Low Level (V) You can specify a value from 0 up to (but less than) the “High Level”.



1

Channel Type: INPUT: Pulse

Channel Name: Area 51 Power Usage

BACNet Unit Group: Others

BACNet Unit Value: NO_UNITS

BACNet COV Increment: 0

BACNet Description: Enter any string you like up to 63 characters.

User Scaling

VALUE	ACTUAL	SCALED
HIGH	1	1.0
LOW	0	0

Pulse Input

Period (Sec): 1.0

● Rate

○ Accumulation

High Level (V): 7.5

Low Level (V): 1.5

Figure 30 — Pulse Input Configuration

6.3.8 Relay Output Configuring

Channels 7 and 8 are fixed as type “*OUTPUT: Relay*” as shown in Figure 31 which uses Channel 8 as an example. Each channel provides a relay contact rated at 30 VAC/DC, 2A. Each relay has a normally-open and a normally-closed set of contacts. The channel BACnet type will be Binary Output. On this screen, you can adjust the following parameters.

Channel Name You can name the channel using no more than 63 characters. This will be the object name for this channel (followed by the channel number). For example, “Relay Output-1”.

BACnet Unit Group The *Others* default can be set to any option in the list.

BACnet Unit Value The *NO_UNITS* default can be set to any option in the list. The available options depend on the BACnet Unit Group you specify.

BACnet Description You can enter any string using no more than 63 characters.

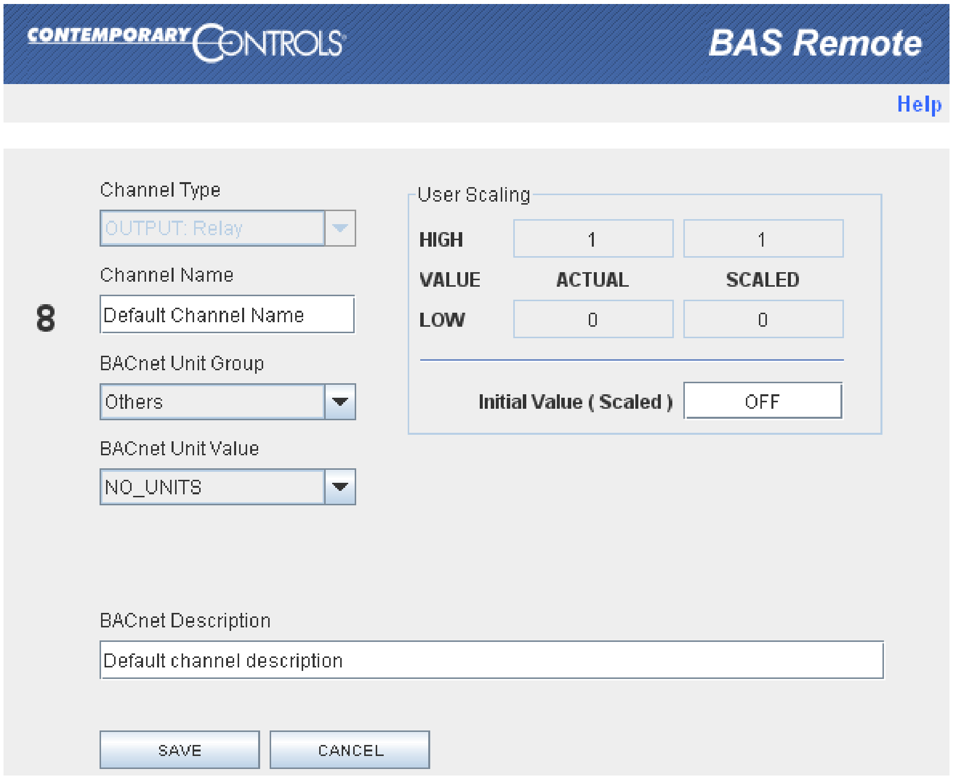
ACTUAL HIGH The default value of *1* is unchangeable.

ACTUAL LOW The default value of *0* is unchangeable.

SCALED HIGH The default value of *1* is unchangeable.

SCALED LOW The default value of *0* is unchangeable.

Initial Value (Scaled) The *Off* default outputs 0, but you may set a *0* or a *1*.



Channel Type: OUTPUT: Relay

Channel Name: Default Channel Name

BACnet Unit Group: Others

BACnet Unit Value: NO_UNITS

User Scaling

HIGH	1	1
VALUE	ACTUAL	SCALED
LOW	0	0

Initial Value (Scaled) OFF

BACnet Description: Default channel description

Figure 31 — Relay Output Configuration

6.4 Channel Forcing

To force a channel, access the Web Server Page, click on the F icon for the channel of interest and make adjustments in the new screen that appears. Your selected channel is confirmed by the large number on the left side of the new screen. If you apply an override value, the override condition will be indicated by a simulated LED on the main Web Server Screen — as described in Section 6.1.4. Also the BACnet status flag overridden will be true for the BACnet object which represents this channel. A forced channel will show up as “offline” in Sedona.

For any channel type selected, the “Forcing” screen reports information for the following read-only fields:

Channel Type This reports the type that you have defined for this channel.

Channel Name This reports the name that you have specified for this channel. This will be the object name for this channel (followed by the channel number). For example, “Analog Input-1”.

Status This reports any of the following conditions:

Status OK
Settings Applied
Output Overload
Input Shorted or Open
Input Open
Input Shorted
Range High
Range Low

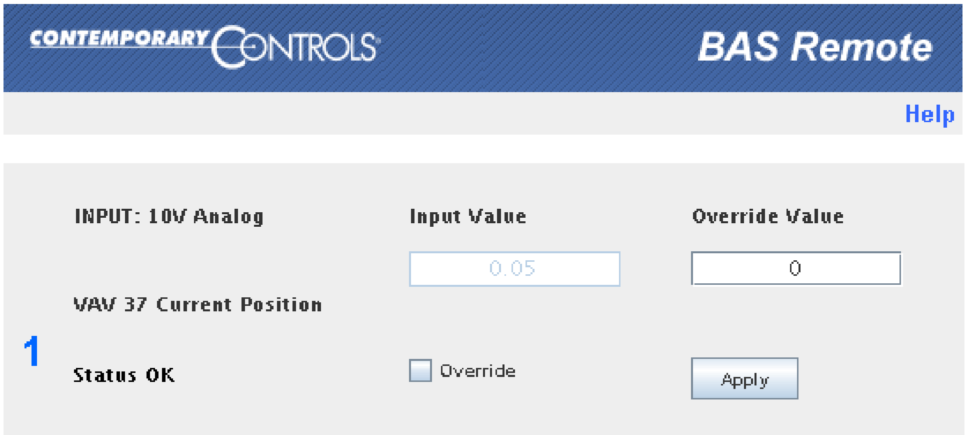
6.4.1 Analog Input Forcing

In addition to *type*, *name* and *status* — this screen displays the following fields.

Input Value This reports the *Input Value* (read-only).

Override Value You can specify an *Override Value*.

Override Checking this box will put the *Override Value* in effect — after the *Apply* button is clicked. To remove the *Override Value*, uncheck the *Override* checkbox and click the *Apply* button.



INPUT: 10V Analog

Input Value: 0.05

Override Value: 0

VAV 37 Current Position

1

Status OK

□ Override

Apply

Figure 32 — Analog Input Forcing

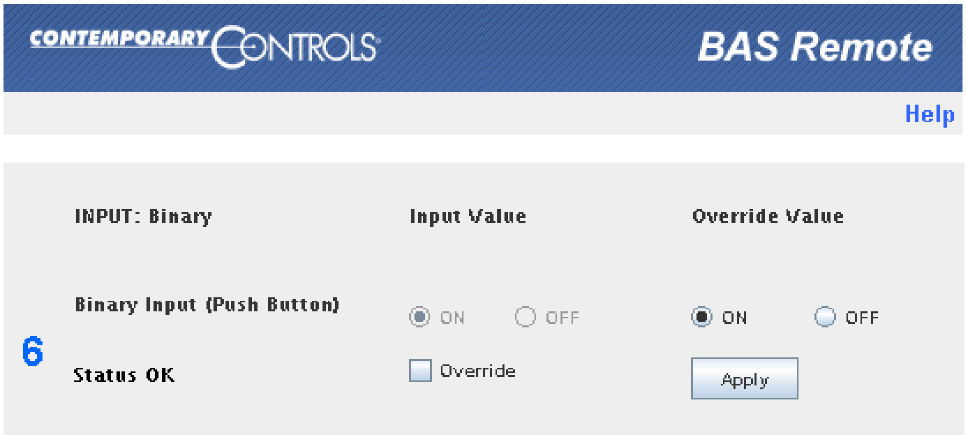
6.4.2 Binary Input Forcing

In addition to *type*, *name* and *status* — this screen displays the following fields.

Input Value This reports the *Input Value* (read-only) as *ON* or *OFF*.

Override Value You can specify an *Override Value* as *ON* or *OFF*.

Override Checking this box will put the *Override Value* in effect — after the *Apply* button is clicked. To remove the *Override Value*, uncheck the *Override* checkbox and click the *Apply* button.



INPUT: Binary	Input Value	Override Value
Binary Input (Push Button)	● ON ○ OFF	● ON ○ OFF
6 Status OK	□ Override	Apply

Figure 33 — Binary Input Forcing

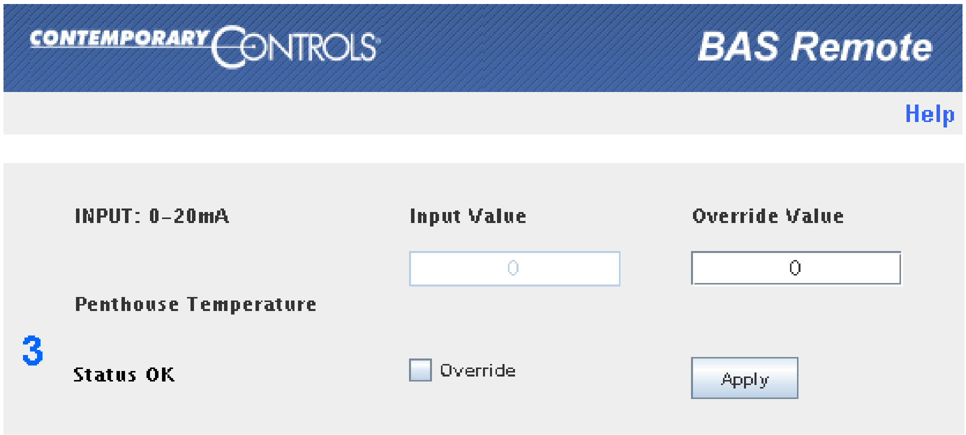
6.4.3 Current Input Forcing

In addition to *type*, *name* and *status* — this screen displays the following fields.

Input Value This reports the *Input Value* (read-only).

Override Value You can specify an *Override Value*.

Override Checking this box will put the *Override Value* in effect — after the *Apply* button is clicked. To remove the *Override Value*, uncheck the *Override* checkbox and click the *Apply* button.



INPUT: 0-20mA	Input Value	Override Value
	0	0
Penthouse Temperature		
3 Status OK	□ Override	Apply

Figure 34 — Current Input Forcing

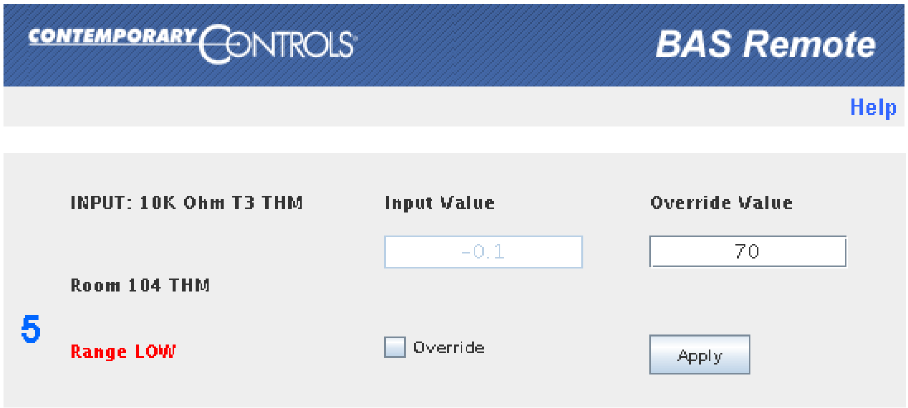
6.4.4 Thermistor Input Forcing

In addition to *type*, *name* and *status* — this screen displays the following fields.

Input Value This read-only temperature value will be meaningless if no thermistor is connected as in Figure 8 (and *Status* will report *Range LOW*).

Override Value You can specify an *Override Value*.

Override Checking this box will put the *Override Value* in effect — after the *Apply* button is clicked. To remove the *Override Value*, uncheck the *Override* checkbox and click the *Apply* button.



INPUT: 10K Ohm T3 THM

Room 104 THM

5

Range LOW

Input Value: -0.1

□ Override

Override Value: 70

Figure 35 — Thermistor Input Forcing

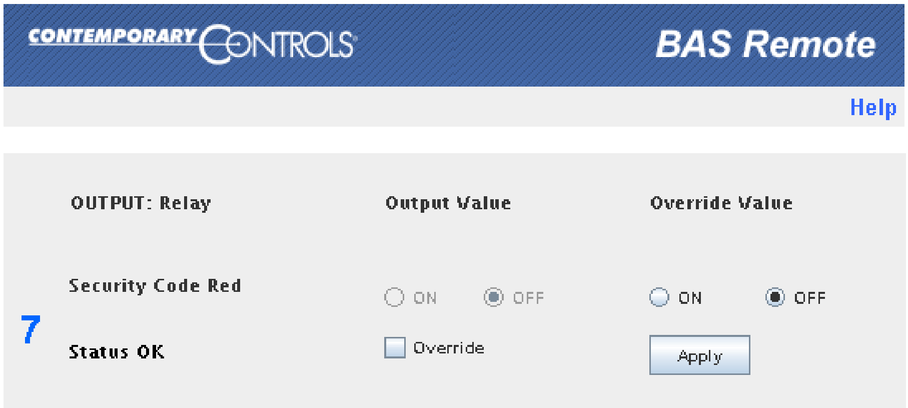
6.4.5 Relay Output Forcing

In addition to *type*, *name* and *status* — this screen displays the following fields.

Input Value This reports the *Input Value* (read-only) as *ON* or *OFF*.

Override Value You can specify an *Override Value* as *ON* or *OFF*.

Override Checking this box will put the *Override Value* in effect — after the *Apply* button is clicked. To remove the *Override Value*, uncheck the *Override* checkbox and click the *Apply* button.



OUTPUT: Relay

Security Code Red

7

Status OK

Output Value: ○ ON ● OFF

□ Override

Override Value: ○ ON ● OFF

Figure 36 — Relay Output Forcing

6.5 Instance Numbers

In several locations in the manual we instruct the use of instance numbers. The instance numbers of the 8 channels of I/O within the BAS Remote will equal their channel numbers. The instance number of the 1st connected expansion module will be 9. The last instance number of channel 8 be 32 (assuming three expansion modules are connected). The instance numbers of virtual objects will be provided in the configuration screen. The instance number of Modbus mapped I/O will be provided in the project.

6.6 Flash File System and Custom Web Pages

An onboard 1 MB of Flash is set aside for your use. Using the default IP address as an example, your Flash area is accessible at ***http://192.168.92.68/user/***. A User ID of "admin" and Password of "ctrlink" allow you to access the user folder, but you cannot create subfolders within the user folder.

Upload your custom webpage files to the BAS Remote by placing them in the folder named:

/data/user

The names of your html files must be ***user.html***, ***user1.html*** ... ***user9.html***. The other file names can be whatever you want (up to 255 characters).

By uploading web pages and images to your Flash space, you can view the BAS Remote status in a graphical format of your own design. Via these custom webpage Java applets, you can access the current status of a channel, virtual object or mapped Modbus variable. **NOTE:** The part of each applet file name shown below in red *will vary by release date*.

- The ***Binary Applet*** displays status by a coloured square which will be red for a value of 0 and green for a value of 1. The size of the displayed square in pixels is specified in your HTML code — with a maximum size of 500 x 500. The applet has two parameters: *Instance* (the instance number of the object to be read) and *TimerMsDelay* (the time between reads of this value on the BAS Remote). The applet file name is:

  binary-applet-2010.03.08.08.48.29.jar

- The ***Channel Set Applet*** writes to a channel or instance using the value specified in the text box. The applet has one parameter: *Instance*. The applet file name is:

  channel-set-applet-2010.03.08.08.48.29.jar

- The ***Channel View Applet*** reads a channel or instance. The applet has two parameters: *Instance* and *TimerMsDelay*. The applet file name is:

  channel-view-applet-2010.03.08.08.48.29.jar

After studying an example web page (***http://192.168.92.68/user.html***) that shows you how to use the three applets, you can erase it and replace it with your own design.

6.7 Web Services

The BAS Remote provides a simple Web Service to support reading/writing of channels, virtual objects and mapped Modbus variables. On receiving a write request, it writes data to the data server for changing the I/O status of the BAS Remote. On receiving a read request, it communicates with the data server and returns the status of the requested channel or Modbus variable.

A *Web Services Description Language* (WSDL) file in the BAS Remote documents the services provided. The file is loaded at **http://192.168.92.68/basremote.wsdl** where the default IP address of 192.168.92.68 is replaced by the actual IP address of the BAS Remote. This allows applications such as Visual Basic, Visual C++, etc. to read/write BAS Remote channels and Modbus mapped variables.

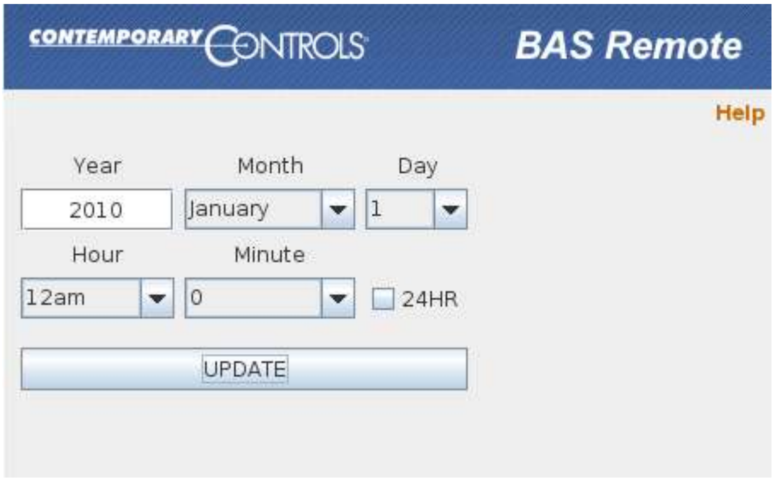
An example Visual Basic program which demonstrates use of the BAS Remote's Web Service functionality will be available on the following webpage:

www.ccontrols.com/exe/brwsdemo.zip

6.8 Set Time

The **BAS Remote Master** has a **Set Time Screen** (Figure 37) which allows you to set the unit's time and date. This feature is most useful in Sedona control strategies.

To manually adjust the date and time, click the *Set Time* button on the **Web Server Screen** (see Figure 21) and the screen of Figure 37 appears. After you have set the current date and time, click the *Update* button when done.



Year: 2010
Month: January
Day: 1
Hour: 12am
Minute: 0
□ 24HR

Figure 37 — Set Time Screen

The *Time* and *Date* can be read via Telnet. The time can also be set by the BACnet *Time Synchronization* service (Section 7.2.7.8).

7 Appendix

7.1 Troubleshooting Ethernet Connectivity of the Master Unit

When troubleshooting Ethernet connectivity for the **BAS Remote Master** unit, as with any IP device, first try pinging the IP address of the Master. The ***ping*** function is typically run as a command line operation in a DOS window. This simple test, performed from any IP-enabled end station, quickly determines if the unit is accessible via the TCP/IP protocol. The ping result (positive or negative) will guide the nature of subsequent troubleshooting efforts.

7.2 BACnet Object Model

The BAS Remote complies with ANSI/ASHRAE Standard 135-2008.

7.2.1 Device

Property Identifier	Property Datatype	Code	Value
Object_Identifier	BACnetObjectIdentifier	R	(Device, Instance 2749)
Object_Name	CharacterString	R	“BAS System Building 1”
Object_Type	BACnetObjectType	R	DEVICE
System_Status	BACnetDeviceStatus	R	(OPERATIONAL)
Vendor_Name	CharacterString	R	“Contemporary Controls”
Vendor_Identifier	Unsigned16	R	245
Model_Name	CharacterString	R	“BASR-8M”
Firmware_Revision	CharacterString	R	“1.0”
Application_Software_Version	CharacterString	R	“1.0”
Protocol_Version	Unsigned	R	2
Protocol_Revision	Unsigned	R	
Protocol_Services_Supported	BACnetServicesSupported	R	(List of Services)
Protocol_Object_Types_Supported	BACnetObjectTypesSupported	R	(List of Object Types)
Object_List	BACnetARRAY[N]of ..Identifier	R	(List of all the objects)
Max_APDU_Length_Accepted	Unsigned	R	1476
Segmentation_Supported	BACnetSegmentation	R	(NO SEGMENT)
APDU_Timeout	Unsigned	R	(3000 MSEC)
Number_Of_APDU_Retries	Unsigned	R	0
Device_Address_Binding	List of BACnetAddressBinding	R	
Database_Revision	Unsigned	R	1

Object Identifier (comprised of the object type *DEVICE* and an *instance number*) must be unique within the complete BACnet network. The default instance (2749) is changed during commissioning. *Object Name* can be set to some meaningful description (e.g.: device location). The remaining fields are set by the manufacturer.

7.2.2 Analog Input

Property Identifier	Property Datatype	Code	Remarks
Object_Identifier	BACnetObjectIdentifier	R	
Object_Name	CharacterString	R	
Object_Type	BACnetObjectType	R	
Present_Value	REAL	R	
Status_Flags	BACnetStatusFlags	R	Status flags will display the status of the channel fault = false for normal conditions. Overridden = false if the channel is not forced. If the channel is a thermistor input and the thermistor is disconnected, the fault flag will be true.
Event_State	BACnetEventState	R	Always normal. Check status flags for real condition.
Out_Of_Service	BOOLEAN	R/W	
Units	BACnetEngineeringUnits	R	

7.2.3 Analog Output

Property Identifier	Property Datatype	Code	Remarks
Object_Identifier	BACnetObjectIdentifier	R	
Object_Name	CharacterString	R	
Object_Type	BACnetObjectType	R	
Present_Value	REAL	W	
Status_Flags	BACnetStatusFlags	R	Status flags will display the status of the channel fault = false for normal conditions. Overridden = false if the channel is not forced.
Event_State	BACnetEventState	R	Always normal. Check status flags for real condition.
Out_Of_Service	BOOLEAN	R/W	
Units	BACnetEngineeringUnits	R	
Priority Array	BACnetPriorityArray	R	
Relinquish_Default	REAL	R	

7.2.4 Analog Value*

Property Identifier	Property Datatype	Code	Remarks
Object_Identifier	BACnetObjectIdentifier	R	
Object_Name	CharacterString	R	
Object_Type	BACnetObjectType	R	
Present_Value	REAL	W	
Status_Flags	BACnetStatusFlags	R	Status flags will display the status of the channel fault = false for normal conditions. Overridden = false if the channel is not forced.
Event_State	BACnetEventState	R	Always normal. Check status flags for real condition.
Out_Of_Service	BOOLEAN	R/W	
Units	BACnetEngineeringUnits	R	
Priority Array	BACnetPriorityArray	R	
Relinquish_Default	REAL	R	
COV_Increment	REAL	R	

* After the channel is configured as a Pulse Input, its object type will be Analog Value.

7.2.5 Binary Input

Property Identifier	Property Datatype	Code	Remarks
Object_Identifier	BACnetObjectIdentifier	R	
Object_Name	CharacterString	R	
Object_Type	BACnetObjectType	R	
Present_Value	BACnetBinaryPV	R	
Status_Flags	BACnetStatusFlags	R	Status flags will display the status of the channel fault = false for normal conditions. Overridden = false if the channel is not forced.
Event_State	BACnetEventState	R	Always normal. Check status flags for real condition.
Out_Of_Service	BOOLEAN	R/W	
Polarity	BACnetPolarity	R	

7.2.6 Binary Output

Property Identifier	Property Datatype	Code	Remarks
Object_Identifier	BACnetObjectIdentifier	R	
Object_Name	CharacterString	R	
Object_Type	BACnetObjectType	R	
Present_Value	BACnetBinaryPV	W	
Status_Flags	BACnetStatusFlags	R	Status flags will display the status of the channel fault = false for normal conditions. Overridden = false if the channel is not forced.
Event_State	BACnetEventState	R	Always normal. Check status flags for real condition.
Out_Of_Service	BOOLEAN	R	
Polarity	BACnetPolarity	R	

BAS Remotes comply with the BACnet Application Specific Controller (B-ASC) profile of the six possible standardized BACnet devices. A B-ASC device can do the following:

Data Sharing

- Ability to provide the values of any of its BACnet objects
- Ability to allow modification of some or all of its BACnet objects by another device

Alarm and Event Management

- No requirement

Scheduling

- No requirement

Trending

- No requirement

Device and Network Management

- Ability to respond to queries about its status
- Ability to respond to request for information about any of its objects
- Ability to respond to communication control messages

Based upon the above requirements, a B-ASC must comply with the following BACnet Interoperability Building Blocks (BIBBs).

B-ASC	BIBB
Data Sharing	DS-RP-B
	DS-WP-B
	DS-RPM-B
	DS-COV-B
Device and Network Management	DM-DDB-B
	DM-DOB-B
	DM-DCC-B

7.2.7 BIBBs

BIBBs are collections of one or more BACnet services between devices on a BACnet network. An "A" device is generally the user of the service or client while the "B" device is the provider of the service or the server. These references are necessary when understanding the BIBBs.

7.2.7.1 DS-RP-B Data Sharing — ReadProperty — B

The BAS Remote functions as the B device and is a provider of data to an A device.

BACnet Service	Initiate	Execute
ReadProperty		X

7.2.7.2 DS-WP-B Data Sharing — WriteProperty — B

The BAS Remote functions as the B device and allows a value to be changed by the A device.

BACnet Service	Initiate	Execute
WriteProperty		X

7.2.7.3 DS-RPM-B Data Sharing — ReadPropertyMultiple — B

The BAS Remote functions as the B device and is a provider of data to an A device.

BACnet Service	Initiate	Execute
ReadPropertyMultiple		X

7.2.7.4 DS-COV-B Data Sharing — ChangeOfValue — B

The BAS Remote functions as the B device and is a provider of data to an A device.

BACnet Service	Initiate	Execute
SubscribeCOV	X	
ConfirmedCOVNotification		X
UnconfirmedCOVNotification		X

7.2.7.5 DM-DDB Device Management — Dynamic Device Binding — B

The BAS Remote functioning as the B device provides information about its device attributes and responds to requests to identify itself.

BACnet Service	Initiate	Execute
Who-Is		X
I-Am	X	

7.2.7.6 DM-DOB-B Device Management — Dynamic Object Binding — B

The BAS Remote functioning as the B device provides address information about its objects upon request.

BACnet Service	Initiate	Execute
Who-Has		X
I-Have	X	

7.2.7.7 DM-DCC-B Device Management — Device Communication Control — B

The BAS Remote functions as a B device and responds to communication control exercised by an A device.

BACnet Service	Initiate	Execute
DeviceCommunicationControl		X

7.2.7.8 DM-TS-B Device Management — Time Synchronization – B

The BAS Remote functions as a B device and interprets time synchronization messages from the A device.

BACnet Service	Initiate	Execute
TimeSynchronization		X

Time Synchronization can set the *Time* and *Date* within the **BAS Remote Master**. Before these parameters are set with this service (or through the **Set Time Screen** in Section 6.8), the *Time Set* virtual object (instance 910001) will be *false*. Once these parameters are set, the *Time Set* virtual object will be *true*.

7.3 Sedona for Control

Tridium has made available the open-source Sedona Framework with which devices can provide user programmable control. Control scheme programming is done using Tridium's Sedona Workbench or Niagara Workbench.

The BAS Remote can execute the Sedona Virtual Machine (SVM) to implement control schemes designed with Sedona Workbench and downloaded to the BAS Remote via the SVM. The SVM executes these two files:

an scode file containing the Sedona application

an SAB file containing the control scheme

The Sedona application is a program (written in the Sedona language) that contains executable code for each Sedona component. The SAB file determines how the components are executed.

The SAB file included with the BAS Remote contains a basic structure similar to that provided by Tridium (although blank). This structure provides a folder that contains a wiresheet. You double click this folder to begin your control scheme design.

As you design your scheme, all changes are immediately sent to the BAS Remote for execution and for monitoring via Sedona Workbench. This only affects the currently executing control scheme stored in RAM. If you store your control scheme to non-volatile memory, it will over-write the SAB file in the BAS Remote.

7.3.1 Getting Started with a Sedona Tool

The best way to begin is to connect either a ***Sedona Workbench*** or ***Niagara Workbench*** tool to a BAS Remote using an Ethernet connection. The BAS Remote is shipped with a pre-installed SVM, a ***kits.scode*** file and an ***app.sab*** file that represents a blank wire sheet. If you are using Niagara Framework you will need version 3.5.24 or later. Expand the Niagara folder loaded on a Windows PC and you will find a Sedona folder. In this folder is a kits folder with additional folders — one of which should be a BAS Remote folder. Also in the Sedona folder is a manifests folder that should also have a BAS Remote folder. Both of these are necessary before using the workbench with the BAS Remote. You will find instructions and necessary files on the controls website, at:

www.ccontrols.com/tech/sedona.htm

Sedona Framework is IP-based so it is necessary to gain access to the BAS Remote over the same IP-subnet where the BAS Remote resides. Using your web browser, access the BAS Remote webpage. In the navigation pane example shown in Figure 38, the BAS Remote is located at private address ***192.168.92.68***. Pointing your browser to this location brings up the BAS Remote webpage which verifies that the PC and the BAS Remote are properly communicating over the same subnet. You should be able to reach the BAS Remote with the Sedona or Niagara workbench as well.

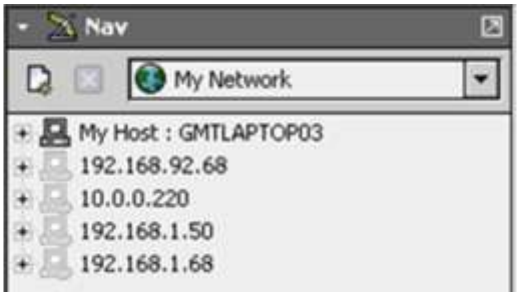
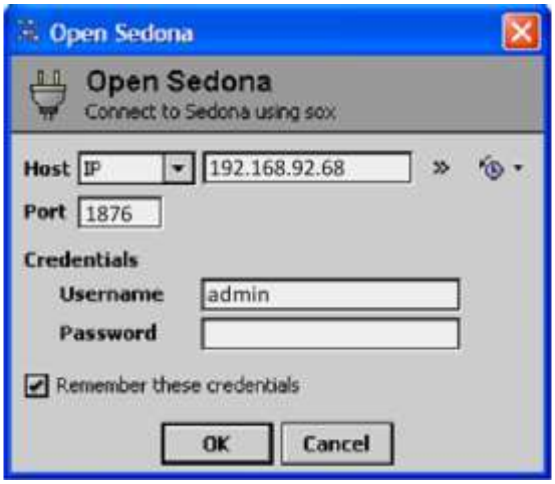


Figure 38 — Workbench Navigation Screen

Launch Sedona or Niagara Workbench. If the BAS Remote was accessed earlier, you should see its IP address in the navigation pane (192.168.92.68 in our example shown in Figure 38). However, if this is the first time the BAS Remote is being accessed, use the menu bar to execute the following options:

File > Open > Open Sedona (sox)

You should see an ***Open Sedona*** dialog box. As shown in Figure 39, the ***Host*** setting should be "IP" and the ***Port*** setting should be "1876". In the box to the right of "IP" enter the IP address of the BAS Remote (which is 192.168.92.68 in this example). You must enter a ***Username*** and ***Password*** for the tool. In the example we accepted the default username "admin", then entered no password — and finally we checked the box to "Remember these credentials".



Host IP 192.168.92.68

Port 1876

Credentials

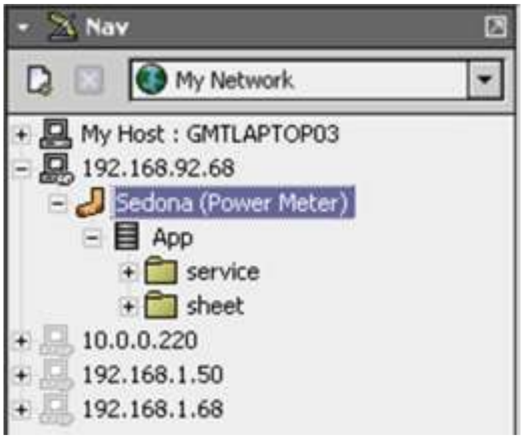
Username admin

Password

☑ Remember these credentials

Figure 39 — Open Sedona Dialog

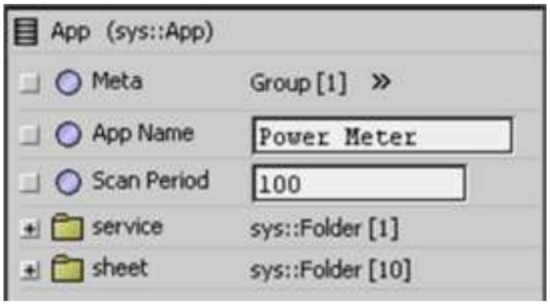
If you are successful in connecting, there will be no error messages. On the left side under the navigation pane (as shown in Figure 40), expand the computer icon that lists the BAS Remote IP address then expand some more by clicking the ***Sedona sox icon***. This will open the navigation tree so that you can see the ***App*** (application) directory. Click on App and you will see a property sheet as shown in Figure 41. On the lower left side, the Sedona Palette will be filled in. To save the wiresheet to BAS Remote non-volatile memory, right click on App and select Actions and then Save.



- My Host : GMTLAPTOP03
- 192.168.92.68
  - Sedona (Power Meter)
    - App
      - service
      - sheet
- 10.0.0.220
- 192.168.1.50
- 192.168.1.68

Figure 40 — Expanded Navigation Tree

On the property sheet you can enter a name for your application under "App Name". You will also notice a "Scan Period" setting with a default time of 100 which is in milliseconds. The "Scan Period" indicates how often the Sedona logic is solved which is once every scan period. During its sleep time, the BAS Remote CPU is allowed to do other tasks. By increasing the "Scan Period", Sedona logic is solved less frequently — allowing other tasks (such as the web server) to perform work.



App (sys::App)

Meta	Group [1] »
App Name	Power Meter
Scan Period	100
service	sys::Folder [1]
sheet	sys::Folder [10]

Figure 41 — Sample Property Sheet

On the navigation pane of Figure 40, the expansion of the ***App*** directory allows you to see two folders named “service” and “sheet”. Click on “sheet” to obtain a blank Wire Sheet. This is where components will be assembled into an application. Under Sedona Palette, select the “control” container to see all the components that are included in the control kit. The “basremote” option displays the BAS Remote kit. To connect the BAS Remote component to the appropriate channel or virtual object, you will need to configure its instance number. These screens are illustrated in Figure 40. To access the components, simply drag one of the components onto the wire sheet. You will be required to provide a unique seven character ASCII name for the component. Information on the contents of the various kits and Sedona Framework in general can be found on our website at:

www.ccontrols.com/basautomation/sedona.htm

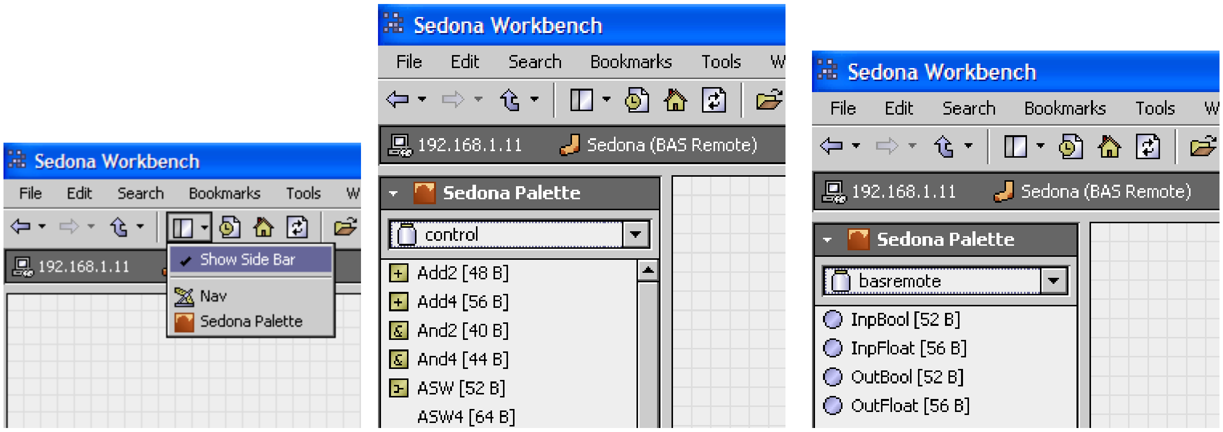


Figure 42 — Accessing the Sedona Palette and the BAS Remote Kit

7.3.2 Virtual Points

Sedona creates a necessity for *Virtual Points* that do not represent real I/O. Virtual Points are placed in the Sedona control scheme and allow applications or devices to access a scheme. An example of a Virtual Point is a temperature set point.

There are five types of Virtual Points:

analog-input
analog-output
analog-value
binary-input
binary-output

Virtual Points are visible via BACnet, Modbus/TCP, custom webpage java applets and Web services — and can be used to share data between Modbus/TCP, BACnet and Web services systems.

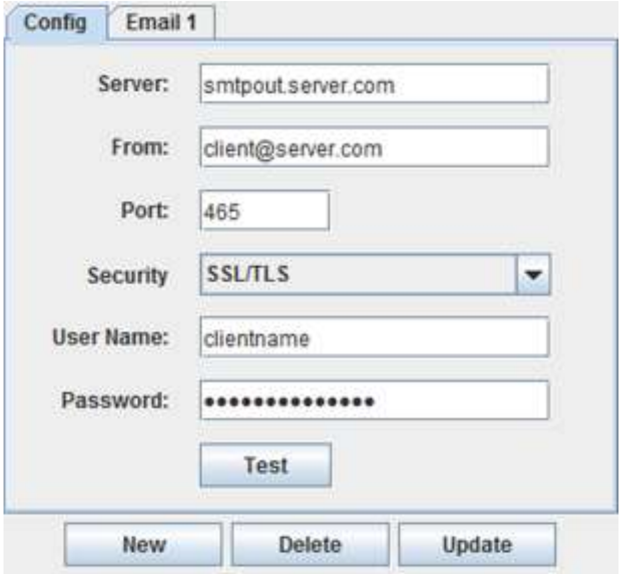
Each Virtual Point has a unique instance number in one of the ranges below:

840001–840050	Analog Inputs, Analog Outputs and Analog Value
810001–810025	Binary Inputs
800001–800025	Binary Outputs

Virtual points are always present, support COV and have a 0 value if not used.

7.3.3 Email Alerts

You can configure the BASremote to transmit an email alert based on a Sedona event. Figure 43 displays your server configuration options.



Server: smtpout.server.com

From: client@server.com

Port: 465

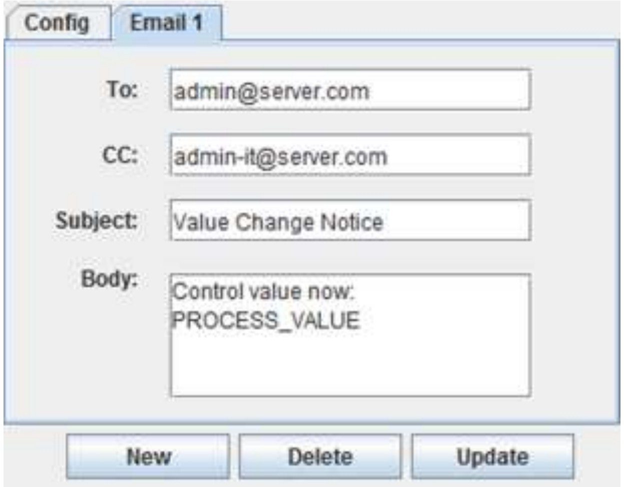
Security SSL/TLS

User Name: clientname

Password: •••••••••••••••

Figure 43 — Sedona Email Configuration O

Figure 44 displays your options for message delivery, subject and triggering event. If desired, you can specify additional emails to trigger on different events and/or target different email recipients.



To: admin@server.com

CC: admin-it@server.com

Subject: Value Change Notice

Body: Control value now:
PROCESS_VALUE

Figure 44 — Sedona Email Detail Screen

7.4 Modbus Server Operation

The **BAS Remote Master** functions as a Modbus TCP server accepting commands from a Modbus TCP client through reserved port 502. As a Modbus TCP server, the **BAS Remote Master** and its attached Expansion modules appear as Modbus serial devices with each module assigned a unique Modbus device address from 1 through 4 with addresses 8 and 9 reserved for the Master. In a similar fashion, third-party Modbus serial devices can be attached to the **BAS Remote Master**'s MB port allowing these devices the ability to be accessed by a Modbus TCP client. All attached serial devices must be assigned unique valid Modbus addresses that do not conflict with those assigned to the BAS Remote modules.

From a Modbus serial point-of-view, the BAS Remote is the Modbus master within the system, but it actually functions as a proxy to Modbus TCP clients. Since the **BAS Remote Master** has its own I/O points, it functions as both the Modbus master and Modbus slave. All BAS Remote Expansion modules function as Modbus slaves as do devices attached to the **BAS Remote Master**'s MB Expansion port. Enabling the **BAS Remote Master** to function as a Modbus TCP server does not preclude the **BAS Remote Master** and its Expansion modules to simultaneously function as BACnet devices.

The easiest way to understand Modbus server operation is to consider the **BAS Remote Master** as also functioning as the Modbus master for attached Modbus slaves. As a master, it does not require a Modbus address. Only Modbus slaves have addresses. The **BAS Remote Master**, receives requests from one or more Modbus TCP clients attached to the Ethernet network. The **BAS Remote Master** receives these requests and sends them to the attached Modbus slave devices as commands. From the Modbus slave's point-of-view, these commands came from the **BAS Remote Master**, and the slave is unaware that a Modbus TCP server even exists.

The **BAS Remote Master** also has some I/O points that are considered Modbus slave points at Modbus address 1. Each of the attached BAS Remote Expansion modules require a single Modbus slave address. If a Modbus command is addressed to any of the BAS Remote modules, it is sent to the appropriate module for execution. If the address is outside the range of BAS Remote module addresses, the command is then sent to the MB port assuming that the intended Modbus slave is connected to this serial Expansion bus. The command is unmodified as it passes through the **BAS Remote Master**. If there is no response from the slave, the **BAS Remote Master** will send a GATEWAY_TIMEOUT message to the Modbus TCP client. It should be remembered that the **BAS Remote Master** utilizes slave addresses 8 and 9 for internal functions and should not be used by any attached Modbus serial devices.

7.4.1 Modbus Register Organization

While the Modbus Data model consists of four types – discrete input, coils, input registers and holding registers, the BAS Remote I/O points appear as holding registers with four holding registers assigned to each channel beginning with register location 40001. For each point there is status, two inputs, and one output register. There are eight points of I/O per module, and the register locations are the same in both the Master and Expansion modules. It is the Modbus device address that distinguishes points on one module versus another. Holding registers represent 16-bit read/write words, but in the Modbus application protocol they are sent as two-byte transfers. However, the input data registers for each I/O point are represented as 32-bit registers regardless if they represent an analog or digital point. On input registers, the least significant word occupies the odd address while the most significant word occupies the even address. Output data registers require only one register per point. What follows are the register locations for universal I/O points 1–6 and for relay outputs 7 and 8.

Channel	Status	Input	Output
1	40001	40101-40102	40201
2	40002	40103-40104	40202
3	40003	40105-40106	40203
4	40004	40107-40108	40204
5	40005	40109-40110	40205
6	40006	40111-40112	40206
7	40007	40113-40114	40207
8	40008	40115-40116	40208

Table 2 — Register Mapping for Input/Output Channels 1–8

7.4.1.1 Status Registers

Each of the eight BAS Remote I/O points has one 16-bit status register. Register location 40001 corresponds to channel 1 while register 40008 corresponds to channel 8. The value of the status register depends upon how the I/O point is configured. For example, an out-of-range indication only applies to those channels configured for thermistor inputs. A similar situation applies to analog voltage outputs and analog voltage inputs. All channels operating within their limits will report a 0x00 when reading the lower 8-bits of the status register.

Register	Bits	Value	Description
4000X	0:7	0	Channel operating within limits
4000X	0:7	2	Analog current output has no load
4000X	0:7	6	Thermistor value out of range
4000X	0:7	9	Analog output deviates by ±60 mV
4000X	8:11	1	Channel LED is red
4000X	8:11	2	Channel LED is green
4000X	8:11	3	Channel LED is off
4000X	12:15	0	Channel is in automatic operation
4000X	12:15	2	Channel value is being overridden

Table 3 — Status Registers

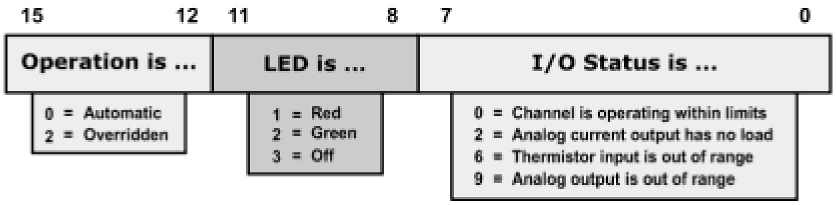

15
12
11
8
7
0
Operation is ...
LED is ...
I/O Status is ...
0 = Automatic
2 = Overridden
1 = Red
2 = Green
3 = Off
0 = Channel is operating within limits
2 = Analog current output has no load
6 = Thermistor input is out of range
9 = Analog output is out of range

Figure 45 — Status Register Depicted Graphically

7.4.1.2 Read Channel Data Registers

Two 16-bit registers are assigned to each of the eight channels for reading input data. Adding an extra register was done to accommodate accumulator values when measuring pulse inputs. The least significant word resides at the register base address while the most significant word resides one register location above the base address. The base register address for channel 1 is 40101 so its most significant word is at 40102. Similarly, channel 8's base address is at 40115 so its most significant word is at 40116.

Register	Bits	Value	Description
401YX	0:15	0x0000 to 0x1388	0-5000 mV on 0-5 V scale*
401YX	0:15	0x0000 to 0x2710	0-10000 mV on 0-10 V scale*
401YX	0:15	0x0000 to 0x07D0	0-2000 hundredths of mA on current
401YX	0:15	0xFEA2 to 0x044C	-350 to 1110 tenths of °F (2's comp.)
401YX	0:15	0xFE8E to 0x01B8	-370 to 440 tenths of °C (2's comp.)
401YX	0:15	0x0000	Contact off (relay de-energized)
401YX	0:15	0x0001	Contact on (relay energized)

* Resolution on 0–5 V scale is 5 mV, and on the 0–10 V scale 10 mV.

Table 4 — Read Channel Data Registers Least Significant Word

Register	Bits	Value	Description
401YX	0:15	0x0000 to 0xFFFF	0-16,777,215 counts
401YX+1	0:15	0x0000 to 0x00FF	

Table 5 — Read Channel Data Registers Accumulator

7.4.1.3 Write Channel Data Registers

Unlike input data registers, output data registers only require one register per channel. Output channel 1 is located at register location 40201 and output channel 8 is located at register location 40208. Since channel 8 is a relay output, outputting a 0x0001 to register 40208 will energize the relay coil.

Register	Bits	Value	Description
4020X	0:15	0x0000 to 0x2710	0-10000 mV on voltage
4020X	0:15	0x0000 to 0x07D0	0-2000 tenths of mA on current
40207	0:15	0x0000	De-energize channel 7 relay
40207	0:15	0x0001	Energize channel 7 relay
40208	0:15	0x0000	De-energize channel 8 relay
40208	0:15	0x0001	Energize channel 8 relay

Table 6 — Write Channel Data Registers

7.4.2 Modbus Function Codes

There are many Modbus function codes but the BAS Remote only responds to three which involve holding registers. These function codes are described below. If other than one of these three function calls are received from the Modbus TCP client, an ILLEGAL_FUNCTION error message is returned by the BAS Remote. The **BAS Remote Master** does not monitor the function calls sent to the MB Expansion bus.

7.4.2.1 03 (0x03) Read Holding Registers

This function code is used to read the contents of a contiguous block of holding registers within the **BAS Remote Master** or BAS Remote Expansion module.

7.4.2.2 06 (0x06) Write Single Register

This function code is used to write a single holding register to the **BAS Remote Master** or BAS Remote Expansion module.

7.4.2.3 16 (0x10) Write Multiple Registers

This function code is used to write a block of contiguous registers (1 to 123 registers) to a **BAS Remote Master** or BAS Remote Expansion module.

7.5 Mapping Modbus Variables to BACnet/IP

7.5.1 General Mapping Principles

The BAS Remote acts as a *gateway* between Modbus devices and BACnet/IP devices. It is the master on the Modbus network and a server on the BACnet/IP network. It will poll for specified Modbus variables (coils or registers) and will map these to specified BACnet/IP objects.

Modbus addresses: When communication occurs between a Modbus TCP host and the slaves devices attached to the BAS Remote, you can use all legal Modbus addresses (except that 1–9 are always reserved for the **BAS Remote Master** module). Due to the large range of instance numbers in, BACnet, it was decided to have the BAS Remote normally map the Modbus address/register or coil to a 22-bit BACnet instance using the first 2 positions as the Modbus device address (10–39) and the next 5 digits as the Modbus register or coil number. For example, 1040001 will be device 10 and register 40001. This gives a range of 1,000,001 to 3,949,999. The maximum number of mapped Modbus variables is 1000.

7.5.2 Mapping Scheme

Each Modbus register is read and converted into a BACnet floating point value. Each coil is converted to a binary value. All such values are readable and some are writeable. A value is accessible in the *present_value* property of the object being mapped. A *present_value* that is written will be passed to the Modbus device only if its priority is sufficient to force a change.

Each Modbus variable maps to one BACnet object as follows:

	Registers		**Coils**	
Modbus Variable **:**	read/write	read-only	read/write	read-only
	maps to	maps to	maps to	maps to
BACnet Object **:**	*analog-output*	*analog-input*	*binary-output*	*binary-input*

To accomplish mapping, the **BAS Remote Master** module is loaded (via FTP) with a user-generated CSV file (**bas_cfg.csv**) that specifies the following:

Modbus Device Address (ID)
Modbus Variable Number
Modbus Variable Type (Register or Coil)

BACnet Instance
BACnet Object Type
BACnet Object Name

The Modbus register types are the following:

16-bit signed
16-bit unsigned

32-bit signed
32-bit unsigned
32-bit float

The 32-bit values are constructed from two consecutive 16-bit registers that have their high and low words defined.

7.5.3 Project Builder

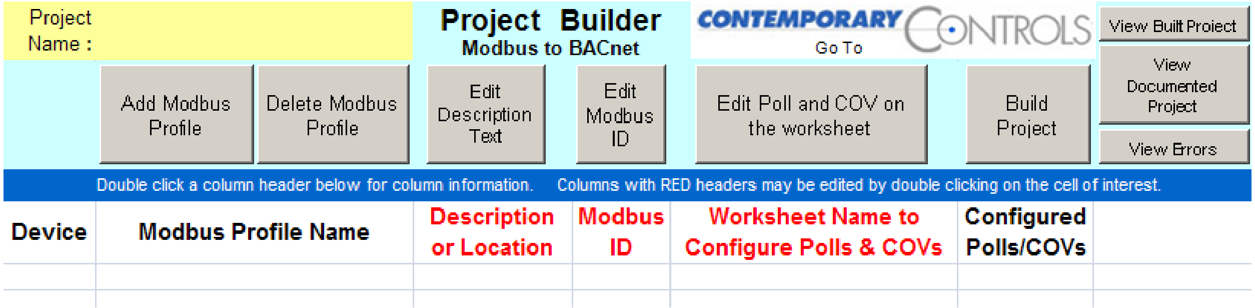
Microsoft® Excel® spreadsheets have been developed to provide you with a means of loading Modbus device profiles into the BAS Remote. You can access these files at …

www.ccontrols.com/basautomation/m2bsoft/

… where you will find the *Project Builder* file and various Modbus Device Profiles for downloading to any convenient location on your local hard drive. You will use these files to build a *Project* that will contain the Modbus Device Profiles you choose to install. The first step in the process is to download:

M2B_ProjectBuilder_A1.xls

When you save and open the above file, the following Excel worksheet appears:



Double click a column header below for column information. Columns with RED headers may be edited by double clicking on the cell of interest.

Device	Modbus Profile Name	Description or Location	Modbus ID	Worksheet Name to Configure Polls & COVs	Configured Polls/COVs

Figure 46 — Project Builder Worksheet (Excel)

As the above sheet loads, the *Project Name* is checked. If none is found, a dialog appears and prompts you to enter a Project Name which will then appear in the upper left portion of the sheet. Because this will also be the name of the Project *file* that will be saved with an XLS extension in a new directory of the same name, you must only use characters that are allowed in Windows filenames. After the Project Name has been accepted (this may take several seconds), you may work with the Project Builder. Double clicking a column header (just below the blue row) will display information about that column. (**Red** column headers identify columns that you can edit.)

The next step in building a project is to add one or more Modbus Device Profile(s) to the Project Builder. If you have a valid Internet connection, clicking the Contemporary Controls logo in the top row of the worksheet will take you to the same web page from which you downloaded the Project Builder file. At this page you can access the list of online profiles and download any profile that your project needs. These files can be saved to any location on your hard drive, but it is convenient if they are in the same location as your Project Builder file.

After the desired profiles have been downloaded, click the *Add Modbus Profile* button in Figure 46. A window will appear and prompt you for a file to load. Navigate to the location where you saved the file(s) that you downloaded, and select a file. After specifying the file to be loaded, that device's *Modbus to BACnet Profile Sheet* (*M2BPS*) will appear — then a dialog will open and prompt you to *Enter Device Description or Location* for this profile that you are adding to your project. You can enter up to 12 characters. (Leaving this field blank is allowable but not recommended.) After specifying the Device Description or Location, another dialog will open and prompt you for the *Modbus Slave ID* (the Modbus address of the device in question). This dialog box also displays the available address values from which you may choose. Make your choice and click *OK*. After the Modbus address value has been accepted, the profile sheet closes and you once again see the Project Builder sheet of Figure 46.

Note: What was the top-most blank line of the Project Builder worksheet will now be numbered and populated with data from the profile you loaded. The easiest way to edit data for Column 3, 4 or 5 is to double click the cell in question. But in a project with many devices, you may prefer to click one of the *Edit* buttons (above the blue row) then use the Device number from Column 1 to specify the device to be edited. The Project Builder columns are discussed below.

Note: Although Excel identifies columns by *letters*, in this discussion the columns of the Project Builder worksheet are identified by *numerals*. This makes it is easier to distinguish references to a column in the Project Builder worksheet from references to a column in a Profile worksheet (where columns are identified by letters).

Column 1 *(Device)* This is an automatically generated number. Each Device number corresponds to a profile that has been added to the Project. You must reference this number if you use one of the *Edit* buttons to edit this device's profile or delete it from your project. If you simply double click a cell to be edited, this Device number is not needed.

Column 2 *(Modbus Profile Name)* This name was specified in the Modbus Profile and cannot be changed.

Column 3 *(Description or Location)* This was the text describing the device description or location that you entered when you loaded the profile.

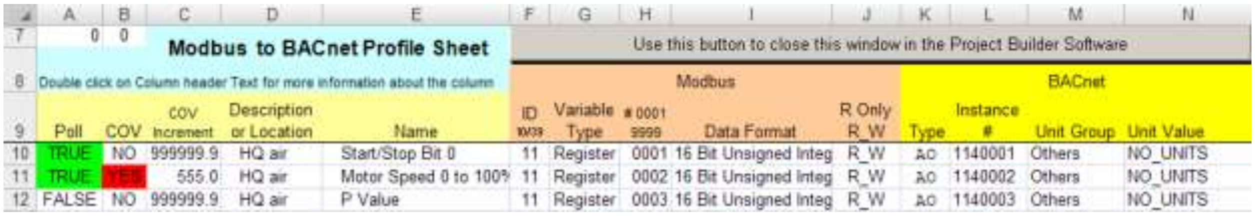
Column 4 *(Modbus ID)* This unique value (10–39) is the Modbus Slave address that you specified when the device profile was loaded.

Column 5 *(Worksheet Name to Configure Poll & COVs)* This is a combination of the values from Columns *3*, 2 and 4. Double clicking this cell (or the button atop Column 5) opens the *M2BPS* for this device as a new editable sheet in your project and the *Worksheet Name* appears on the worksheet tab.

Column 6 *(Configured Polls/COVs)* This reports how many Polls and COVs are configured in each *M2BPS*. Initially, this column will display "0/0" for each profile that you have added to your project. (By default, each Modbus Profile will have the *Poll* function set to *FALSE* (off) for each of its variables.)

7.5.3.1 Editing Modbus to BACnet Profile Sheets

After you add a *Modbus to BACnet Profile Sheet* (*M2BPS)* to your project, you must edit it and set the *Poll* function to *TRUE* for each variable that you want polled. A *M2BPS* ***must be edited*** before its associated device can communicate via the BAS Remote. A sample *M2BPS* is shown in Figure 47.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
7	0	0	Modbus to BACnet Profile Sheet			Use this button to close this window in the Project Builder Software								
8	Double click on Column header Text for more information about the column					Modbus					BACnet			
9	Poll	COV	COV Increment	Description or Location	Name	ID [illegible]	Variable Type	# 0001 9999	Data Format	R Only R_W	Type	Instance #	Unit Group	Unit Value
10	TRUE	NO	999999.9	HQ air	Start/Stop Bit 0	11	Register	0001	16 Bit Unsigned Integ	R_W	AO	1140001	Others	NO_UNITS
11	TRUE	YES	555.0	HQ air	Motor Speed 0 to 100%	11	Register	0002	16 Bit Unsigned Integ	R_W	AO	1140002	Others	NO_UNITS
12	FALSE	NO	999999.9	HQ air	P Value	11	Register	0003	16 Bit Unsigned Integ	R_W	AO	1140003	Others	NO_UNITS

Figure 47 — Sample Modbus to BACnet Profile Sheet (Excel)

Note: The *M2BPS* in Figure 47 has standard column and row headings, but rows 1–6 are not shown. This is because rows 1–6 have been reserved for special use and are therefore hidden. The list of profile variables begins with row 10. The various features of this sheet are explained below.

Cell A7 This is the number of profile variables which have *Poll* set *TRUE* **as reported by the Project Builder** (Figure 46). Changes made in the profile sheet of Figure 47 will not be reported in Cell A7 until the sheet is closed, causing Column 6 of the Project Builder to be updated. Subsequently reopening the view in Figure 47 will display the updated value.

Cell B7 This is the number of profile variables which have *COV* set *TRUE* **as reported by the Project Builder.** Like Cell A7, changes made in the profile sheet of will not be reported in Cell B7 until the sheet is closed, and Column 6 of the Project Builder is updated. Subsequently reopening the view in Figure 47 will display the updated value.

Column A Set the drop-down menu option to TRUE if you want to include this variable (row) in your project. **Note:** If you do not set any row to *Poll* = *TRUE*, this profile cannot become part of your built project.

Column B If you set a variable (row) to be polled, set this option to *YES* if you want a COV message sent to the subscribing BACnet device. If you set this option to *YES* but do not set *Poll* = *TRUE* for the same variable (row), this option will have no effect.

Column C If you set a cell in Column B to *YES*, you are prompted for the COV increment value that must occur before the COV message will be sent. If a Column B cell is left at *NO*, the Column C default value will stay at *999999.9*.

Column D If you entered a Description or Location when the profile was loaded, it registered in Column 3 of the Project Builder and also appears in this column. **Note:** Any cell in Column D can be edited, but the entire column will be reset by any edits to Column 3 in the Project Builder worksheet. Therefore, if you wish add special text to a variable, do it in Column E.

Column E This is pre-filled with the variable name supplied by the manufacturer of the device. Any cell in this column can be individually edited if you wish rename the variable to something more meaningful in your installation.

Column F This is the *Modbus ID* (address) you set when the profile loaded. You can edit a cell in this column, **but don't**. Remember, every variable (row) in the profile of a single device must have the same value in this column.

Column G The *Variable Type* may be either a Register or Coil.

Column H The *Variable Number* is a value from 0001 through 9999.

Column I The *Data Format* is one of those described Section 7.5.2.

Column J This specifies the variable as Read-Only or Read-Write.

Column K This is the *BACnet Object* that corresponds to the *Modbus Variable* as described in Section 7.5.2.

Column L The *BACnet Instance* is a calculated value that you cannot edit. It is the concatenation of the values from Columns F, G and H. For example, a Modbus register 40001 (*Variable Type* 4 + *Variable Number* 0001) at address 10 will be mapped to *BACnet Instance* 1040001.

Column M Many standard BACnet *Unit Groups* are supported in the drop-down menu.

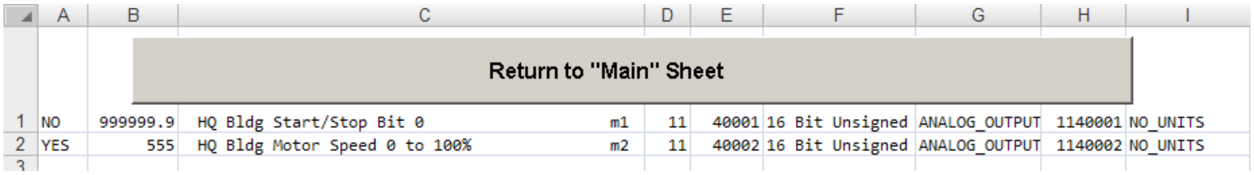
Column N The *Unit Value* options depend on your selection in Column M.

Row 9 Double clicking any cell in this row will display helpful information about the column in question.

7.5.3.2 Building and Viewing a Project

After you have edited each *M2BPS* that you have added to your project, you are ready to build the project by clicking the *Build Project* button. Once you click this button, the Project Builder will create a CSV file that you will later load into the BAS Remote. Building the CSV can take some time — the more devices in your project, the longer it will take. Each device profile will take a few seconds to render into CSV. When the build is complete, a new window will open and confirm the build.

After the project is built, see the results by clicking the *View Built Project* button in the upper-right portion of Figure 46. In response to your click, a new worksheet opens to display the contents of the CSV file that was just built. A sample screen is shown in Figure 48 and is discussed below.



	A	B	C	D	E	F	G	H	I
1	NO	999999.9	HQ Bldg Start/Stop Bit 0 m1	11	40001	16 Bit Unsigned	ANALOG_OUTPUT	1140001	NO_UNITS
2	YES	555	HQ Bldg Motor Speed 0 to 100% m2	11	40002	16 Bit Unsigned	ANALOG_OUTPUT	1140002	NO_UNITS
3									

Figure 48 — Sample Built Project

Each row in the worksheet displays how one *Modbus Variable* has been mapped to a *BACnet Object*. The information in the columns is as follows:

Column A This is how the *COV* flag was set — either *NO* or *YES*.

Column B This is the chosen *COV Increment* value (*999999.9* if *COV* is set to *NO*).

Column C This BACnet *Object Name* is a concatenation of two values from the *M2BPS* (*Description or Location* and *Name*) and a third value generated by Project Builder. This third value takes the form of *mxxxx*, which is the letter *m* plus four digits appended to assure that the *Object Name* is unique.

Column D This is the *Modbus Address* (*ID*).

Column E This is the Modbus Register value.

Column F This is the Modbus Register Type (number format).

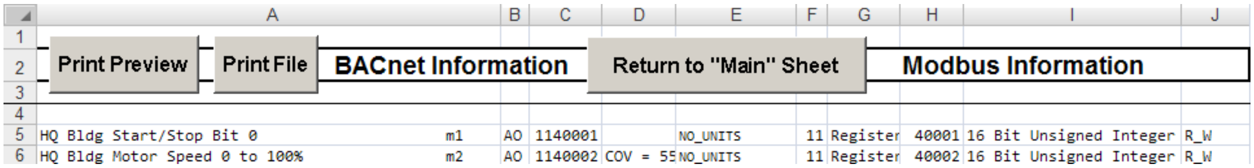
Column G This is the *BACnet Object* to which the *Modbus Variable* was mapped.

Column H This is the *BACnet Instance* that was created by Project Builder.

Column I This is the unit of measure that was selected.

Note: Six more columns (J–O) are present, but are reserved for future use.

You can also print a report of your project. To do this, click the *View Documented Project* button in the upper-right portion of Figure 46. In response to your click, a new worksheet opens to display the most important parameters of the project that was just built. Unlike the CSV file of Figure 48 which has parameters sequenced for proper mapping, the sample Documented Project of Figure 49 has parameters displayed in a sequence more meaningful to humans. As shown below, typical *Print Preview* and *Print* functions have their respective buttons on the left of the screen.



	A		B	C	D	E	F	G	H	I	J
5	HQ Bldg Start/Stop Bit 0	m1	AO	1140001		NO_UNITS	11	Register	40001	16 Bit Unsigned Integer	R_W
6	HQ Bldg Motor Speed 0 to 100%	m2	AO	1140002	COV = 55	NO_UNITS	11	Register	40002	16 Bit Unsigned Integer	R_W

Figure 49 — Sample Documented Project for Printing

7.5.3.3 Loading Your Project into the BAS Remote

Once your **bas_cfg.csv** file is ready, use an FTP client to copy it to the BAS Remote in the following directory:

/data/config/

The User Name is ***admin*** and the Password is ***ctrlink***. Once the **bas_cfg.csv** file has been copied to the BAS Remote, reboot the unit to implement your mapping scheme.

7.5.4 Viewing the Mapping Status in the BAS Remote

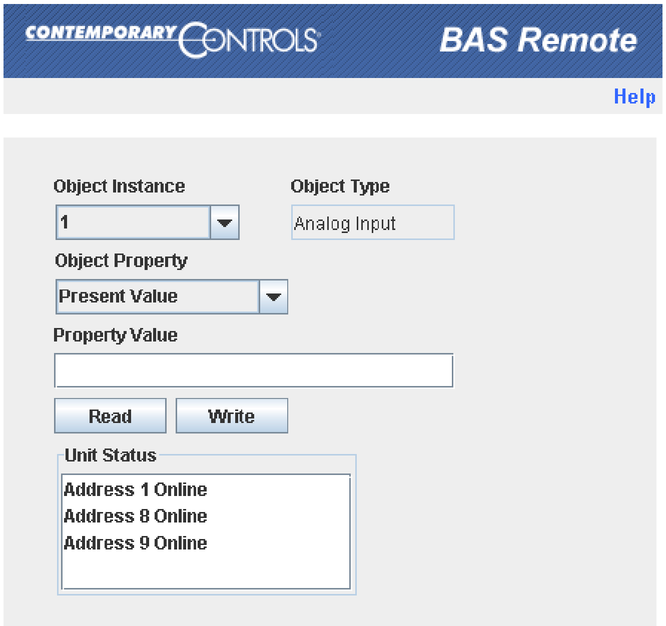
To view the Mapping in the BAS Remote, click on the *Map* button on the Web Server Page (just under the device image in Figure 21). Then a screen ***similar*** to that of Figure 50 appears. But your screen will display more “Address” lines in the *Unit Status* box — where each additional line corresponds to one of the Modbus addresses of your map.

This window not only displays the Modbus mapped variables, it also displays all instances in the BAS Remote — for example, the 8 I/O channels of the BAS Remote and many of their parameters. It also allows you to view or change the preset value of most instances within the BAS Remote. This is particularly useful when you want to clear a pulse counter. It also displays the BACnet status of each instance. Having the full list of instances is handy when doing Sedona programming.

Note the following considerations:

- The screen of Figure 50 displays the situation of a BAS Remote with no Expansion Modules installed. If one Expansion Module were installed and working, the *Unit Status* box would list “Address 2 Online”. If two Expansion Modules were installed, the *Unit Status* window would list “Address 2 Online” and “Address 3 Online”. And you would see “Address 4 Online”, if you had installed a third Expansion Module.
- If a Modbus device (or an Expansion Module) has been installed but is reported as “Offline” because the Master Module has detected a communication failure and listed the questionable device as “Offline”. In such a case, you may notice that periodically the “Offline” report ***briefly*** changes to “Online”. This brief “Online” report is due to the Master Module attempt to contact the questionable device every scan cycle. Due to other considerations, the actual frequency of this “Online” report cannot be predicted — thus, you should expect brief “Online” reports at irregular intervals (but no more often than the scan cycle set by the parameters in Section 6.1.3.5).

Because Addresses 5, 6 and 7 are reserved for future use, you will not see these three addresses displayed in the *Unit Status* box.



Object Instance: 1

Object Type: Analog Input

Object Property: Present Value

Property Value:

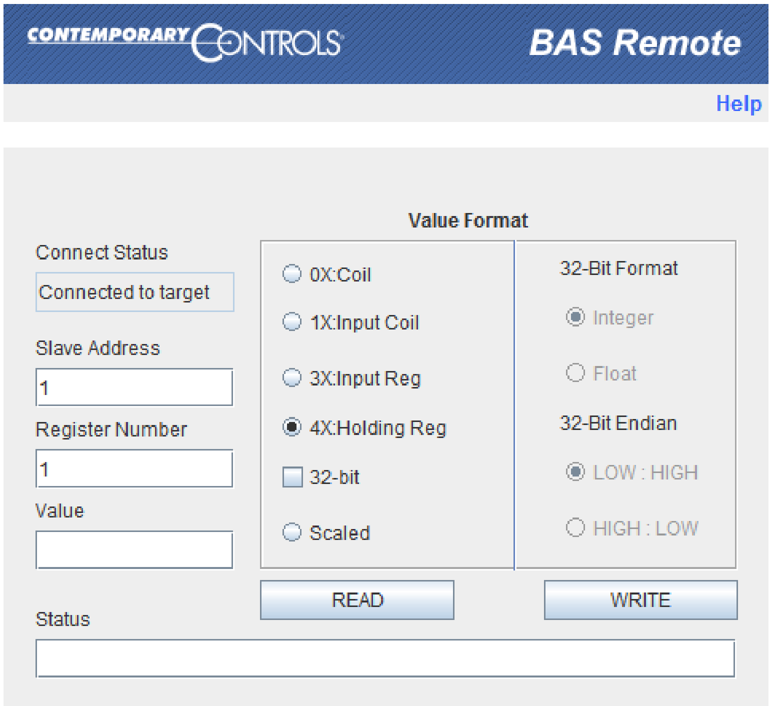
Unit Status

Address 1 Online
Address 8 Online
Address 9 Online

Figure 50 — Map Screen (Unpopulated)

7.6 Modbus Utility

This is an embedded utility that allows direct access to the Modbus registers within the **BAS Remote Master**, BAS Remote Expansion modules and any attached Modbus serial slave devices. It functions as a Modbus TCP client and it is an excellent utility to verify that the BAS Remote is communicating with any attached slave devices. With a web browser and an IP connection to the BAS Remote you can remotely examine and alter various Modbus register values. Access the utility by clicking the *Modbus Utility* button on the **BAS Remote Master** web page. Once the connection is made, the following screen appears.



Connect Status: Connected to target

Slave Address: 1

Register Number: 1

Value:

Status:

Value Format

○ 0X:Coil
○ 1X:Input Coil
○ 3X:Input Reg
● 4X:Holding Reg
□ 32-bit
○ Scaled

32-Bit Format
● Integer
○ Float

32-Bit Endian
● LOW : HIGH
○ HIGH : LOW

Figure 51 — Modbus Utility Screen

As shown above, this screen has 5 fields, 9 radio buttons and a pair of read/write buttons. The five fields are:

Connect Status It initially reports "Connected to target" then awaits your first register access. As this is a Java applet, it executes on your computer. When it successfully connects to the BAS Remote, it will report "connected to target".

NOTE: If you have uploaded a bas-cfg.csv file to the BAS Remote, you will only be able to read/write the Modbus values of the devices listed in the file. You will still be able to read/write all the Modbus registers/coils of devices not listed in the CSV file.

Slave Address This is ***1*** by default. Enter the Modbus address of the device you wish to access. To avoid erroneous results, it must be one of the following:

- the **BAS Remote Master**
- a BAS Remote Expansion module
- a Modbus slave device attached to the Modbus Expansion port of the **BAS Remote Master**

Register Number This is ***1*** by default. Enter ***only the least three significant digits*** of the address of the register you wish to access. The two most significant digits will be added ***automatically*** with a value that depends on whether you click the *Read* button or the *Write* button.

Value This field is used to ***read or write*** values for the register specified in the *Register Number* field. When writing a value, match the number format of the selected radio button or an error will result. If an error occurs, its description will be displayed in this field.

Status This read-only field reports the status of the most recent Modbus action. If your attempt to read or write a register is successful, the reported status is "Success" and the *Value* field displays the value in question. If the attempt to access a register is unsuccessful, the *Value* field reports the type of error that was generated.

The five radio buttons ***on the left*** specify which number format is displayed in the *Value* field. The choices are:

0X:Coil Use this button if the addressed device is a *Binary Output* that has only two possible state values (0 or 1) — as is the case for a coil.

1X:Coil Use this button if the addressed device is a *Binary Input* that has only two possible state values (0 or 1) — as is the case for a coil.

4X:16-Bit This button displays the specified 16-bit register value in raw-number format.

4X:32-Bit This button displays the specified 32-bit value of a register pair in raw-number format and enables the four radio buttons ***on the right*** (see below).

Scaled If the specified register represents a BAS Remote I/O point, engineering units can be displayed by selecting this button. But if the register is not one of the BAS Remote I/O points, an error will result.

The four radio buttons ***on the right*** specify the number format of a 32-bit value. The choices are:

Integer This button specifies a whole value with no fractional content.

Float This button specifies a single-precision floating point value.

LOW:HIGH This specifies *Little Endian* format whereby the ***low*** word occupies the base register and the *high* word occupies the next-higher register.

HIGH:LOW This specifies *Big Endian* format whereby the ***high*** word occupies the base register and the *low* word occupies the next-higher register.

Read and *Write* buttons Clicking one of these buttons causes the specified action to occur.

7.7 Linux License

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author’s protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program”, below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License.

However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.
To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

> *<one line to give the program’s name and a brief idea of what it does.>*
> *Copyright (C) <year> <name of author>*
>
> *This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.*
>
> *This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.*
>
> *You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.*

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

> *Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w’. This is free software, and you are welcome to redistribute it under certain conditions; type `show c’ for details.*

The hypothetical commands `show w’ and `show c’ should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w’ and `show c’; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

> *Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision’ (which makes passes at compilers) written by James Hacker.*
>
> *<signature of Ty Coon>, 1 April 1989*
> *Ty Coon, President of Vice*

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.