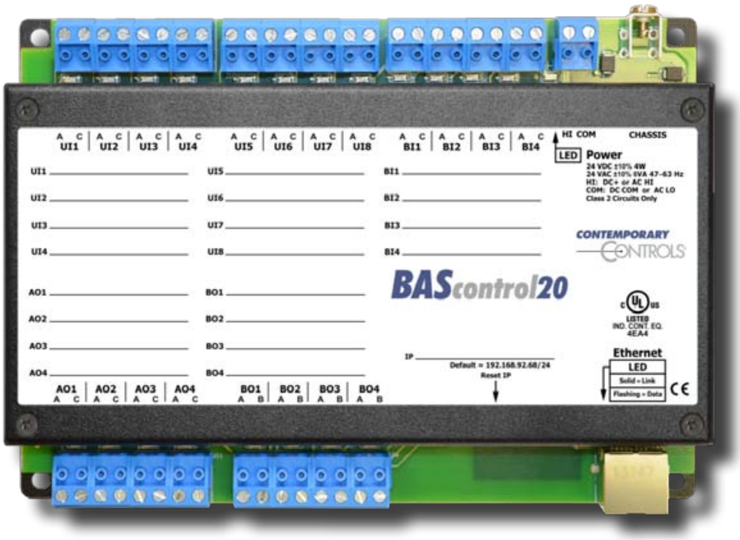
application NOTE

Sedona Framework™ Demo Box

Introduction

The Sedona Framework Demo Box provides a convenient means of familiarizing a person with the capabilities of Sedona Framework and the BAScontrol20 — a 20-point BACnet/IP Sedona Field Controller. For those without access to the demo box, one can easily be constructed following the information in this application note. Just by following the instructions in this application note, a person can learn much about the BAScontrol20 or Sedona Framework in general.

The BAScontrol20 is a freely-programmable direct digital controller (DDC) with an Ethernet connection to an IP network. It complies with the BACnet B-ASC device profile and includes a Sedona Framework virtual machine (SVM) allowing it to be programmed with tools such as Niagara Workbench or a third-party Sedona programming tool. Using a visual drag-and-drop methodology, a systems integrator can assemble components onto a wire sheet to create applications. Once developed the application program remains stored on the BAScontrol20 and executes by way of the resident SVM. The BAScontrol20 utilizes Sedona 1.2 components from Tridium as well as custom components from Contemporary Controls. Components reside in kits and the equivalent kits must exist in both the BAScontrol20 and the Workbench tool. Required Workbench kits are available from Contemporary Controls' web site.



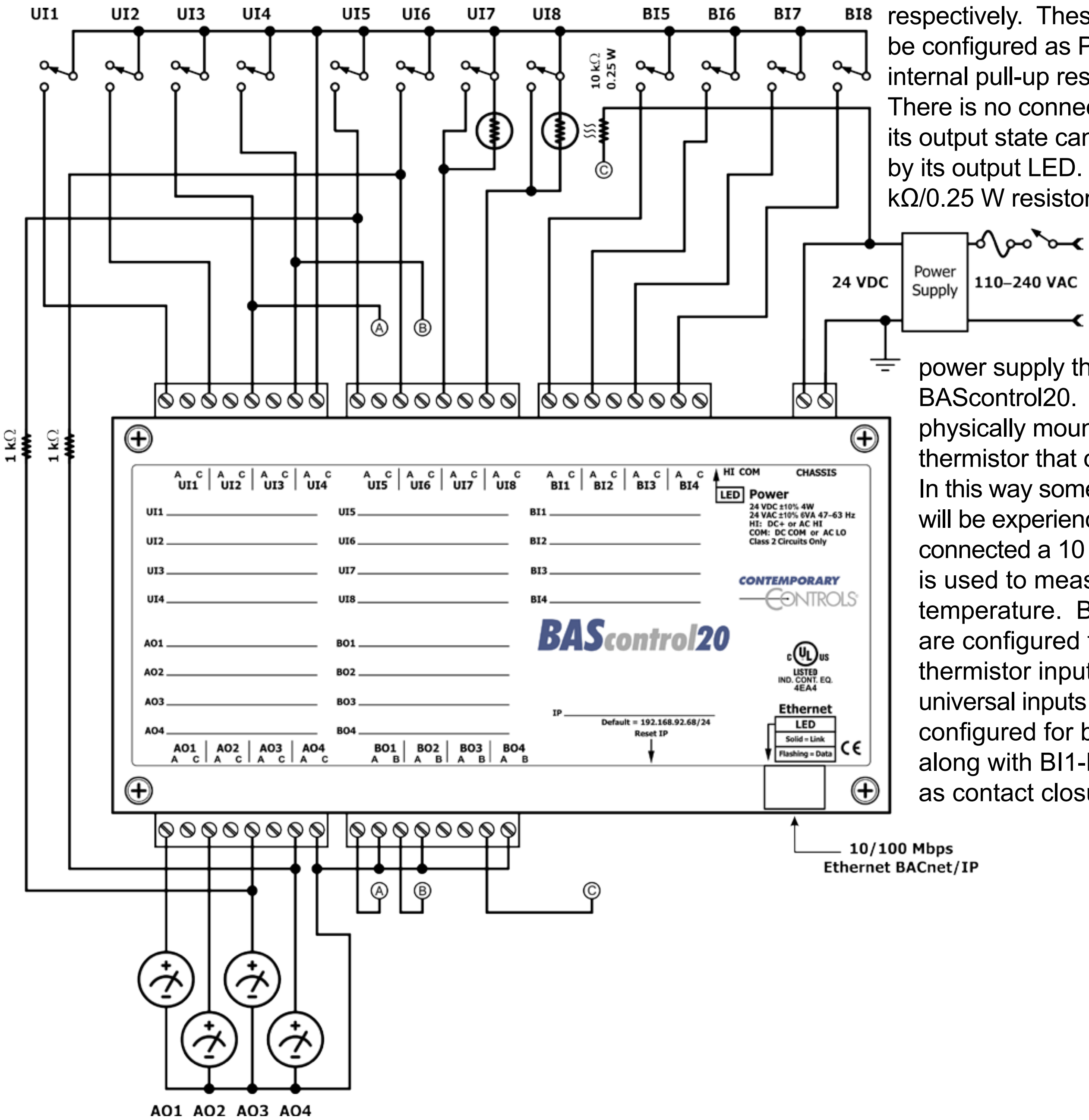
For convenience to the user, the demo box has been pre-wired and pre-programmed so that the user can experience the execution of programs loaded into the BAScontrol20. The pre-loaded application can be modified at will to test different sequences which can be saved to an attached Workbench tool. The default application can be reloaded if necessary. For those interested in constructing a demo box, the pre-loaded Sedona application is available for download from Contemporary Controls' web site. For those without a demo box who just want to test some applications, all IO points — including virtual points — can be forced using web pages thereby allowing most applications to be tested without the need for physical input and output devices.

BASautomation®

Making Connections to the IO

The BAScontrol20 provides eight universal inputs (UI1–UI8), four binary inputs (BI1–BI4), four analog outputs (AO1–AO4) and four binary outputs (BO1–BO4). The model used happens to have relay binary outputs. Of these 20 IO points, only the universal inputs require configuration. Inputs are located at the top of the unit and the outputs at the bottom. In order to provide as much flexibility as possible, it was decided to install switches on all input points because it is possible to configure all the inputs as contact closure inputs. Each universal input and each binary input has a center-off toggle switch that has two other positions — momentary ON and maintained ON. In either of the ON conditions, the input is shorted to common thereby turning the point LED indicator on and presenting a True input state to the Sedona logic.

All four analog outputs (AO1–AO4) have attached 0–10 VDC voltmeters for reading each output. AO3 and AO4 output signals are fed back to universal inputs UI5 and UI6 respectively via 1 kΩ resistors which are used to protect the analog outputs from a switch closure to common. UI5 and UI6 will assume the default condition of analog inputs. Binary outputs BO1 and BO2 are each tied to inputs UI3 and UI4 respectively. These two inputs will be configured as Pulse inputs with internal pull-up resistors enabled. There is no connection to BO3 but its output state can be determined by its output LED. BO4 feeds a 10 kΩ/0.25 W resistor to the +24 VDC power supply that powers the BAScontrol20. This resistor is physically mounted near a 10 kΩ thermistor that connects to UI8. In this way some thermal feedback will be experienced. UI7 also has connected a 10 kΩ thermistor that is used to measure ambient temperature. Both UI7 and UI8 are configured for Type III 10k thermistor inputs. The remaining universal inputs — UI1, UI2 — are configured for binary inputs and along with BI1-BI4 are all treated as contact closure inputs.

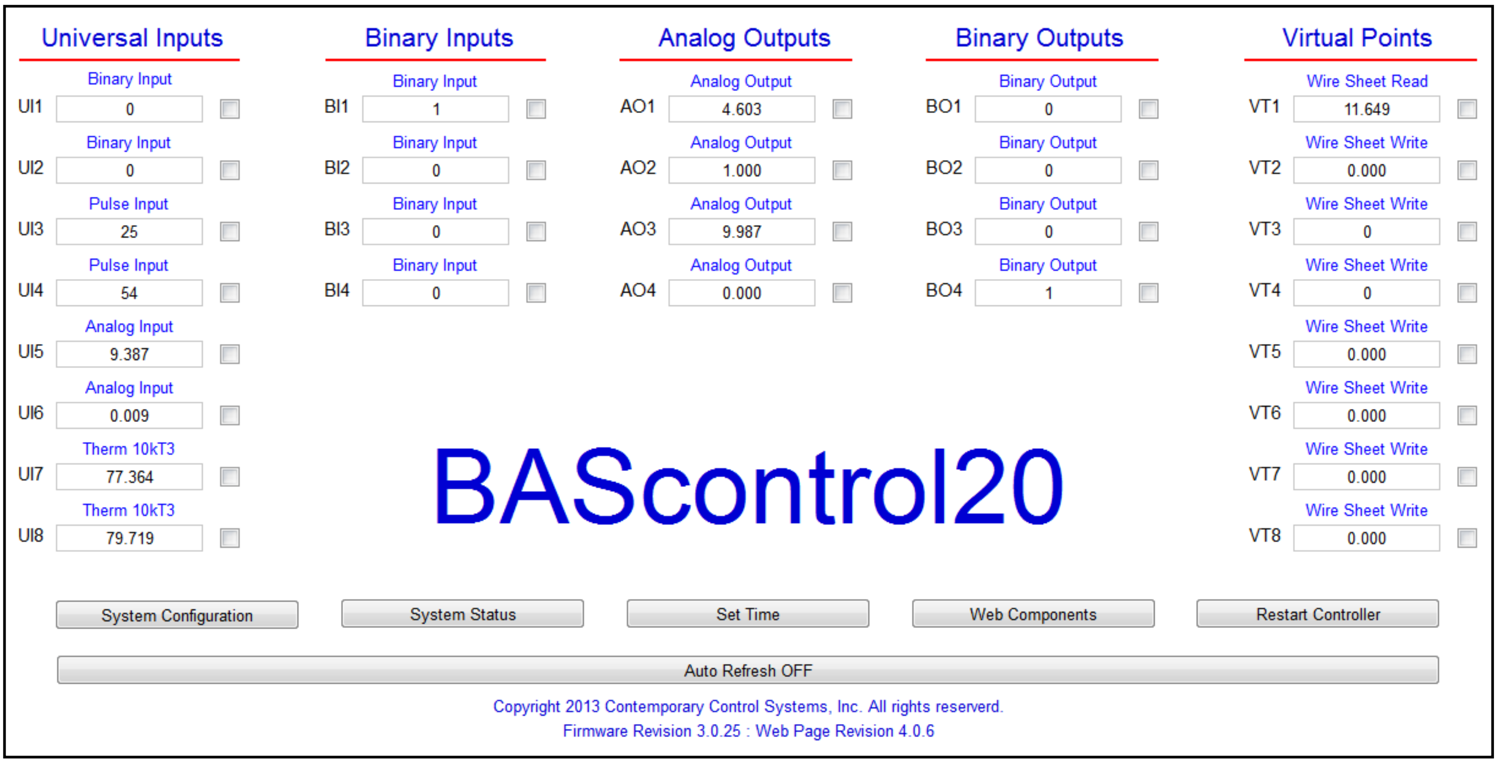


Configuration

Accessing the Main Web Page

Configuring the universal inputs, virtual points and setting the real-time clock is via web pages using a standard web browser. A connection is made to the BAScontrol20 10/100 Mbps Ethernet port to a web browser using a common CAT5 Ethernet cable. The BAScontrol20 supports auto-negotiation and auto-MDIX. The LED next to the Ethernet port should flash once a connection is made. The default IP address of the BAScontrol20 is 192.168.92.68/24. Make sure the web browser is on the same subnet as the BAScontrol20. If successful, there will be a request for a Username and Password. Enter the default settings of admin/admin. At that point the main BAScontrol20 web page should be visible.

The main web page displays the status of the BAScontrol20's 20 real IO points plus eight virtual points. At the bottom of the screen is an Auto Refresh button that allows for the periodic updating of the displayed data. If this feature is not wanted, just toggle the Auto Refresh button. There are four buttons just above the Auto Refresh button. Click on the System Configuration button to begin the configuration process.



The main web page provides an overview of the 20 real IO points plus the eight virtual points. Access to other configuration pages is possible through the main page. Point forcing is possible.

Configuration

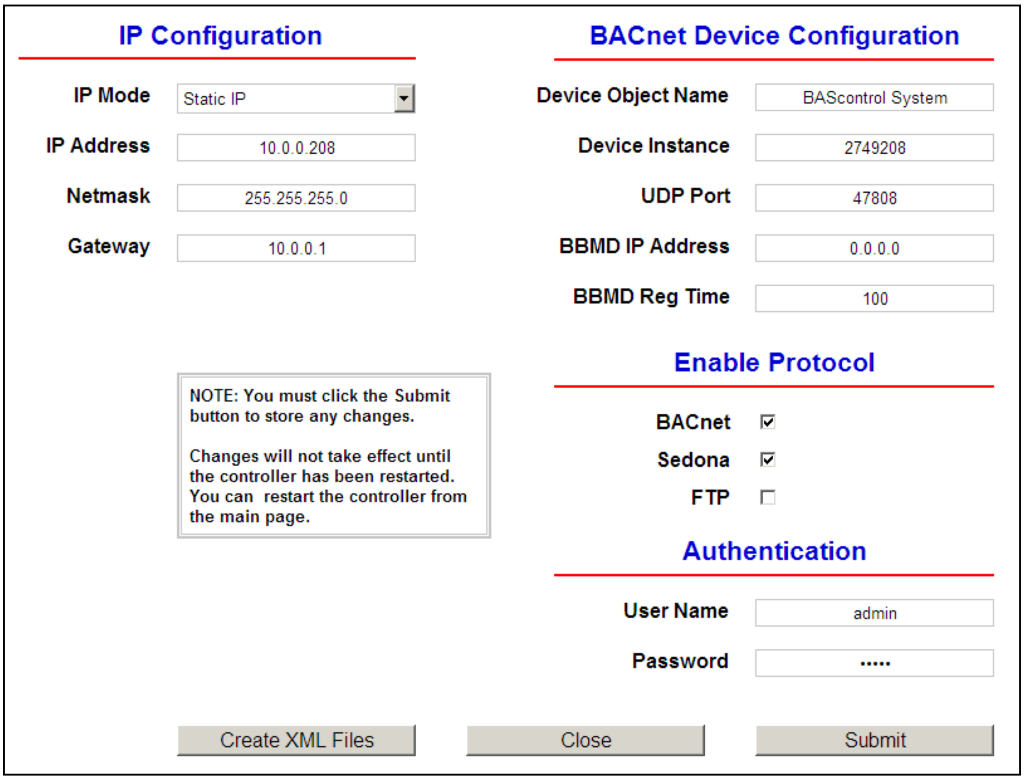
System Configuration

The system configuration page has several sections including the IP Configuration section. Here you can change the IP settings if so desired. A static IP address is selected by default but the address can be changed by entering the new IP address, netmask and gateway settings. However, the BAScontrol20 also supports DHCP client so by selecting this mode a DHCP server on the attached network will supply the necessary IP settings for you. A gateway address setting is necessary if the NTP time server is going to be accessed or if the BAScontrol20 is to reside on a sub-netted BACnet/IP network.

On the same screen there is a BACnet Device Configuration that must be set if BACnet is to be used. The Device Object Name must be unique BACnet internetwork wide so change the name to something more meaningful without duplication. The Device Instance default is 2749. This number likewise needs to be unique so immediately change it by appending a two- or three-digit variable to the end of this number and incrementing the value as more devices are added. However, the complete number can be changed to your choosing. The UDP Port should probably not be changed. If this unit is to operate over a sub-netted BACnet Internetwork, then the location of the BBMD device must be entered. The BAScontrol20 supports foreign device registration (FDR). The BBMD Reg Time is probably fine as is.

You have a choice of protocols — BACnet, Sedona and FTP. If FTP is selected, BACnet and Sedona are de-selected. If you want a freely-programmable controller, select Sedona. If you choose BACnet and not Sedona, you can still use the BAScontrol20 as a BACnet/IP IO controller. If you want to ignore BACnet and only use Sedona, you can make that selection. By selecting both BACnet and Sedona you gain a freely-programmable Sedona controller plus the ability to function as a BACnet/IP server device compliant with the B-ASC BACnet device profile. When both protocols are selected and using a BACnet client, you can read both the BAScontrol20 real inputs and real outputs but you cannot write to real outputs if the enable slot on a particular AO or BO is true. This slot present on each output component provides Sedona precedence when controlling outputs. If the slot is false, BACnet can control this point. The BAScontrol20 will ignore the BACnet initiated output commands to individual output components that are enabled. With Sedona disabled on the System Configuration page, BACnet gains precedence when writing outputs. The BAScontrol20 supports up to eight virtual points that can be manipulated by a BACnet client if BACnet is enabled. There is no output writing restrictions on the virtual points.

You can also change your Username and Password. Once all changes are made click on Submit to save changes and then Close. Any change to the System Configuration settings must then be followed by a controller re-boot. On the main controller page, click Restart Controller to initiate a re-boot.



The System Configuration page allows for IP and BACnet device configuration. Select the desired protocol(s) from this page.

Configuration

Recovery Mode

In the event that you forget your IP settings or user name or password there is a way to recover. On the right hand side of the BAScontrol20 under the cover is a small push button. Depress this button until the LEDs for UI1–UI4 and AO1–AO4 begin to alternately flash. This indicates recovery mode. Using the default IP settings and user name/password connect the web browser to the BAScontrol20 and view the System Configuration screen. Your original settings will still be displayed except for the password. Enter a new password and make any changes desired and then click Submit and then Close. At this point you are still in recovery mode. On the main web page click on Restart Controller and the unit will come back with the submitted settings running its application having exited recovery mode.

Setting the Time

From the main web page there is a button that can be clicked to set the time. A System Time screen displays the current time along with settings to set the time. Time can be set manually or automatically using a time server if access to the Internet is available. Make sure that if a time server is to be accessed that the proper gateway address has be set in the IP configuration in order for the controller to reach out to the Internet.

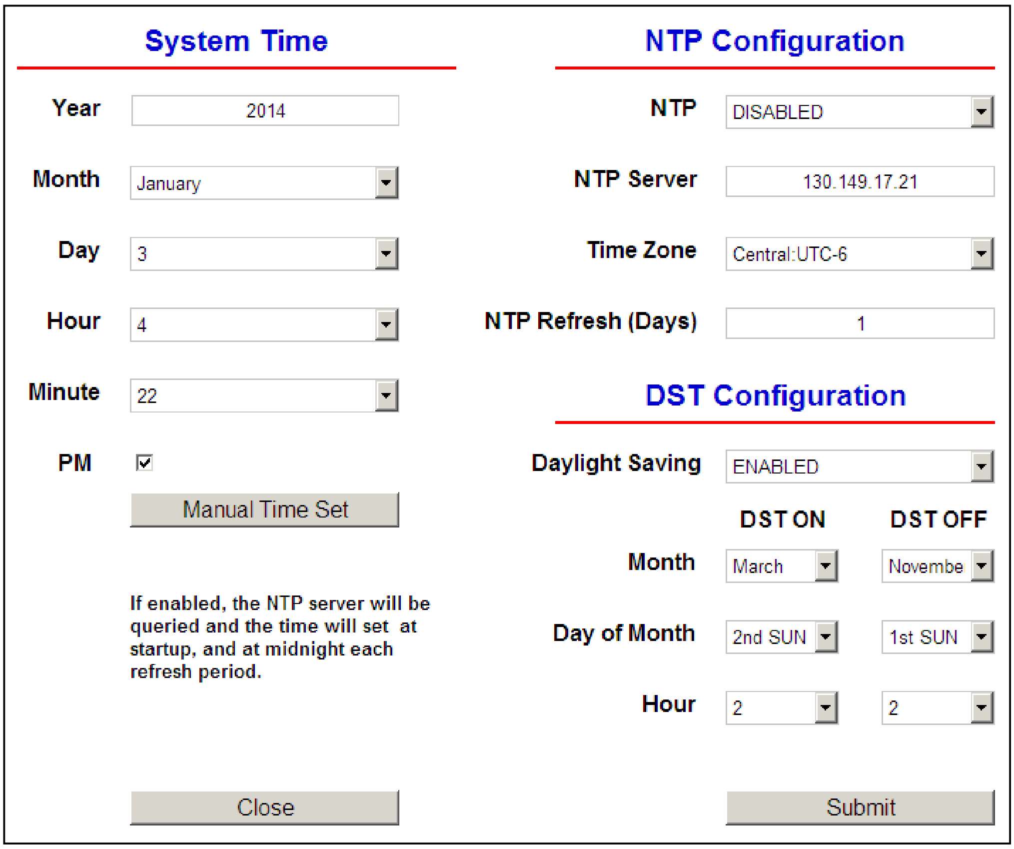
If Network Time Protocol (NTP) is disabled, then a Manual Time Set button will appear. Make the time settings manually and click Manual Time Set. Note that this screen refreshes automatically so you will see the actual time changing once set.

If Daylight Savings Time (DST) is to be supported, you can make the appropriate settings for the beginning and ending of DST. Enable DST and then click the Submit button below the settings.

If NTP is to be used to set the time, enable this function at the top-right of the screen. A time server IP address has been provided for convenience. Make sure you select the Time Zone for the location of the controller. Leave the NTP Refresh (Days) to 1. Verify the DST settings if DST is to be enabled before clicking Submit at the bottom of the screen. Note that NTP does not account for DST. When using NTP, the system time is updated upon power up, upon a changeover to NTP and at every 24 hour increment based upon the setting of NTP Refresh.

System Time utilizes a real-time clock that is backed-up with a Super-cap in the event of a power failure. Time is maintained up to seven days without power to the controller. This is important to know if manual setting of time is used. With NTP, the time is reset upon power-up.

Setting the time is important. Other than time zone, any time setting directly affects the Sedona time component in the DateTimeServiceStd kit which is used in scheduling events. Time zone must be configured using the Workbench tool.

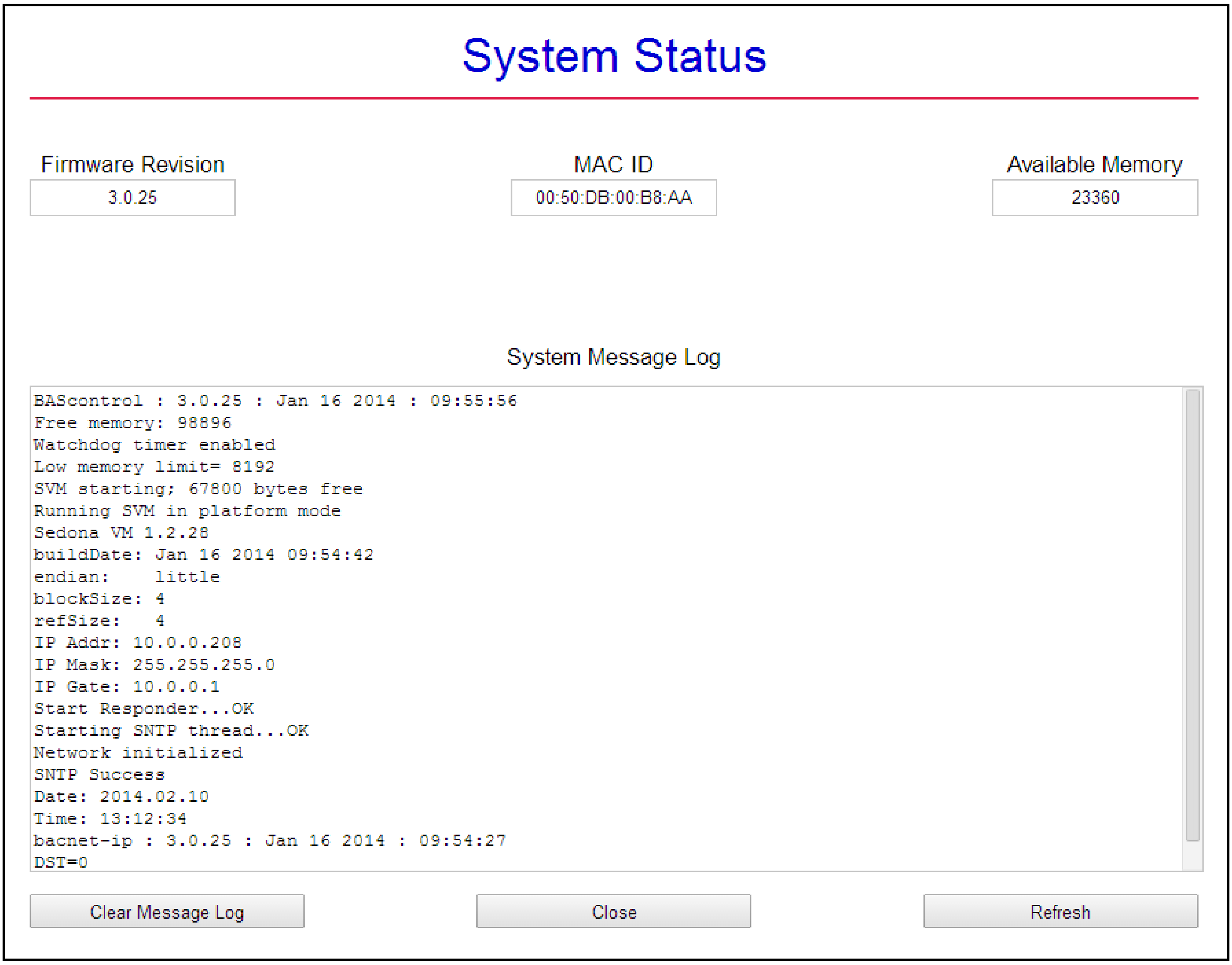


System time can be set manually or through an NTP server assuming that Internet access exists. The real-time clock is backed up through the use of a Super-cap.

Configuration

Accessing the System Status Screen

Another screen that is accessible from the main web page is the System Status Screen. This screen provides a message log when the controller first boots. It is information that is provided to Contemporary Controls' technical support staff if troubleshooting is necessary. However, there are three meaningful fields at the top of the screen that could prove useful. On the top-left is the firmware revision number of the controller. In the middle is the MAC address of the Ethernet port, and on the top-right is an indication of available memory for applications. The available memory will vary with controller activity and the opening and closing of web pages. A more meaningful memory reading can be found in the BASC20PlatformService custom component located on the Sedona wire sheet under App>Services.



The System Status page could be helpful for troubleshooting.
The available memory indicator will change as web pages are loaded.

Configuration

Configuring the Universal Inputs

From the main web page all real and virtual IO points can be configured for BACnet by just clicking on the particular point. However, the universal inputs require additional configuration besides BACnet. Once all the points are configured and saved, there is usually no need to reconfigure them unless points are added or changed.

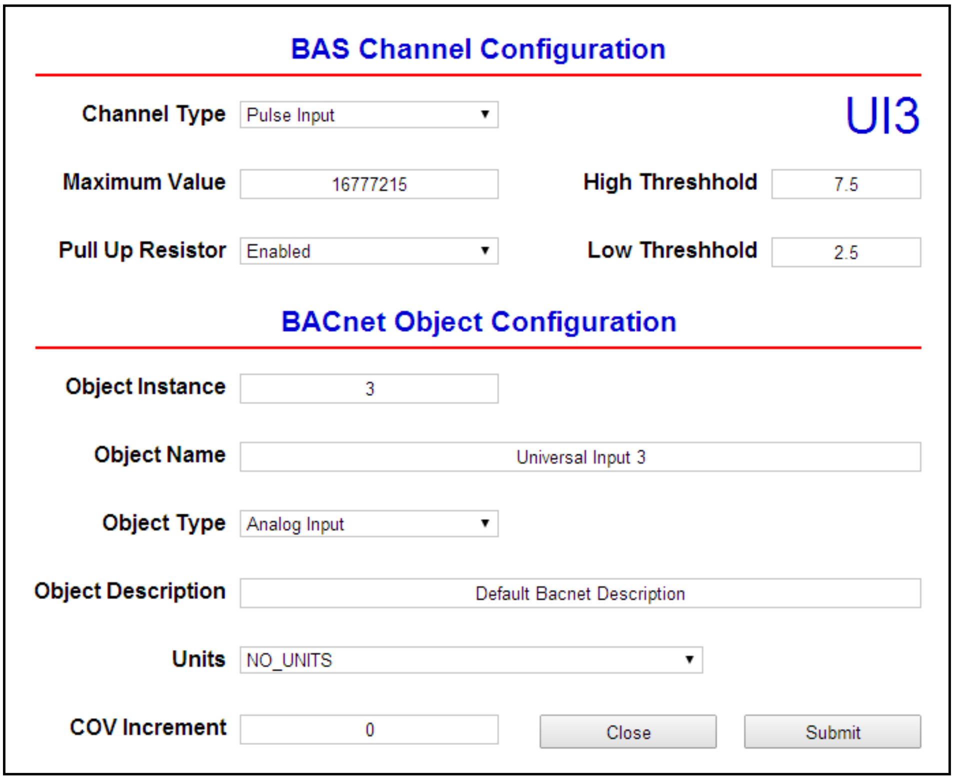
The unit's universal inputs UI1–UI8 assume the default 0–10 VDC analog input state from the factory. For this application UI5 and UI6 are to be analog inputs so they need not be changed. However, the BACnet settings can be changed from the default settings. Click on either the UI5 or UI6 point on the main screen which will lead you to the point's configuration screen. If you are interested in configuring the BACnet portion of the point you can do it now. The Object Instance and Object Name are predefined to ensure that they are unique. The Object Instance cannot be changed but the Object Name can. Just make sure the Object Name is unique within the device. The Object Description can be changed as well with no restrictions on choice. The Units can be selected from a drop-down. The COV increment can be specified but it is the BACnet client that would establish COV communication with the BAScontrol20. If no COV is anticipated leave this value at zero. The BAScontrol20 will allow up to two analog and two binary COV subscriptions to a BACnet client. Once you are finished, click on Submit and close the screen.

UI1 and UI2 are to set for contact closures so click on those points from the main screen and a channel configuration screen will appear for each point. On the drop-down menu select Binary Input. The COV increment and Units are meaningless on a binary variable for BACnet settings but the remainder of the BACnet settings can be made or the defaults accepted.

UI3 and UI4 are to be set for pulse so on their configuration screens select Pulse Input on the drop-down. You will notice that additional information is requested. The maximum value of the pulse accumulator is displayed but this valued can be reduced if desired. The pulse transducer can have an active output between 0–10 VDC or it can have a passive output requiring power to excite its output. In that case enable the Pull Up Resistor option in the drop-down. For this application we need a pull-up resistor on both channels because we are sensing simple passive contact closures. You can leave the default threshold settings as is. Make any desired BACnet changes and click Submit and Close.

The final configuration is for the two thermistors attached to UI7 and UI8. For these devices select Therm 10KT3 to indicate Type III 10 kΩ thermistor curve. You can specify either Celsius or Fahrenheit and what you select will appear in the BACnet section. You can also apply a Temperature Offset if the thermistor is reading higher or lower than determined by a calibrated source. Enter the correction value and click Submit after any BACnet changes are made. Then close the screen.

Of the eight universal inputs, only UI1–UI4 can be used as retentive pulse accumulators meaning that in the event of a power loss their value will be retained up to seven days through the use of a Super-cap.



The System Status page could be helpful for troubleshooting.
The available memory indicator will change as web pages are loaded.

Configuration

Configuring Non-Universal IO Points

Once universal inputs are configured, what remain of real IO points are BI1–BI4, AO1–AO4 and BO1–BO4. The only configuration required of these points is the BACnet settings. As with the universal inputs, the BACnet object instance and Object name are predefined but the Object name can be changed as long as it is unique within the device. Entering an Object description is optional as is Units and COV Increment.

Configuring Virtual Points

In the CControls_BASC20_IO kit are eight virtual point components (VT1–VT8) that are used by a BACnet client to send and receive intermediate data to and from the BAScontrol20. By intermediate data we mean that the data is neither input data nor output data but something in between real inputs and real outputs. It could be set point or reset data intended for the wire sheet or calculated or status information generated by the wire sheet. Although BACnet allows for the reading of the BAScontrol20 real input and output points – and under certain conditions the writing of real output points – the virtual points have no reading or writing restrictions. Virtual points are treated by BACnet as either a binary variable (BV) or analog variable (AV) while real points appear as binary inputs (BI), analog inputs (AI), binary outputs (BO) or analog outputs (AO). The BAScontrol20 logic engine reads the state of its inputs (AI and BI) and outputs (AO and BO), executes logic, and then sets outputs (AO and BO) accordingly. In a similar manner, a BACnet client can "read" the BAScontrol20's real inputs and will attempt to "write" to the BAScontrol20's real outputs. AVs and BVs are a bit different in that they can be configured to be either an input to the BACnet client (read) or an output from the BACnet client (write). Therefore we need to establish rules for the use of AVs and BVs.

If a BACnet client is to read intermediate data from the Sedona wire sheet, this is no different from accessing data from an input component on the wire sheet. We would call this "reading from the wire sheet" or Wire Sheet Read. The VT on the wire sheet would have a channel type (Chn Type) of "float out" or "binary out."

If a BACnet client is to write intermediate data to the Sedona wire sheet, this is no different from logic on the wire sheet writing to an output component. We would call this "writing to the wire sheet" or Wire Sheet Write. The VT on the wire sheet would have a channel type (Chn Type) of "float in" or "binary in."

Like universal inputs, virtual points are configured via web pages that are accessible from the main web page. Click on the particular virtual point to gain access to its configuration page. Select either Binary Variable or Analog Variable and check the button called Read from Wire Sheet if the intent is for the BACnet client to read this point. If the intent is for the BACnet client to write to this point, check the button called Write to Wire Sheet. You will notice that the heading of the virtual points will be either Wire Sheet Read or Wire Sheet Write depending if the BACnet client is going to be reading wire sheet data or writing wire sheet data.

All eight virtual components are retentive upon power loss up to seven days. This allows a BACnet client command to be retained even if power is lost to the controller. Their backup is accomplished using a Super-cap.

Configuring Web Components

Access to web component configuration is through the main web page. Web components provide a means of interacting with the Sedona wire sheet via a web browser versus using a Workbench tool. These are custom components developed by Contemporary Controls which are provided in the CControls_BASC20_Web kit. Configuring the 48 web components is accomplished from eight web pages. Each web component can be configured as a wire sheet input or a wire sheet output using a drop-down box. For every web component (WC), a description and value can be entered on the web page. If the component is configured as a wire sheet input (float, integer or binary), limits can be assigned for the variable's range. If the entered value is within the limits, it will be applied to the web component on the wire sheet. This eliminates the need to add limit logic on the wire sheet. For wire sheet outputs, limits are ignored. The description field is only used as an aid to the systems integrator in understanding the function of the component.

Configuration

Web Components

<PREV NEXT>

	Description	Value	Wire Sheet	Min	Max
WC17	Monday Duration	480	Input	0	1439
WC18	Tuesday Occupied	540	Input	0	1439
WC19	Tuesday Duration	480	Input	0	1439
WC20	Wednesday Occupied	540	Input	0	1439
WC21	Wednesday Duration	480	Input	0	1439
WC22	Thursday Occupied	540	Input	0	1439
WC23	Thursday Duration	480	Input	0	1439
WC24	Friday Occupied	540	Input	0	1439

Auto Refresh ON Close Submit

Up to 48 web components are available for data exchange between web pages and wire sheets without the need of a programming tool. These components are used for scheduling.

Forcing IO Points from the Main Web Page

There is one feature available on the main web page that could be useful during checkout but must be done with great care. Both input and output points can be forced to states and values different from program generated values. Looking at the main web page, it is possible to both read and write values for the 20 real IO points and eight virtual points. There is no issue with reading points – just writing points. Just to the right of the value field is a checkbox that when hovered over says Click to Force Channel. To change a value, check this box before making a value change. This override value will remain until the checkbox is unchecked.

BACnet Discovery Tool

To examine the BACnet device and object properties of the BAScontrol20, Contemporary Controls' BACnet Discovery Tool (BDT) or any other BACnet tool can be used. BDT can be downloaded from the Contemporary Controls' web site at no charge. Not only can you discover the BAScontrol20's device object with its properties, you can read the virtual objects provided in the Demo Box application programming.

Programming

Accessing the Sedona logic

To access the Sedona logic you will need Niagara Workbench or a compatible tool. The proper kits must be installed on the workbench tool.

Go to File>Open>Open Device. If you do not see Open Device, you will need to first load in Sedona in the Workbench tool. Contemporary Controls hosts a video on how this is accomplished. Assume for the moment that Open Device can be found. Clicking on this option opens up a window. In the Connect window enter the IP address of the Sedona device such as 10.0.0.208. Enter the Username and Password. The default is admin/admin. Click Remember These Credentials and then click OK.

You should see on the right screen the word App. Double-click App which will bring you a list. At the bottom of the list is the word Sheet. Click on Sheet and you will see the main wiresheet with Contemporary Controls IO kit components already placed.

Default Wire Sheet

A BAScontrol20 shipped from the factory has a default wire sheet populated with custom components from Contemporary Controls' CControls_BASC20_IO kit. These include universal IO points UI1–UI8, binary inputs BI1–BI4, binary outputs BO1–BO4 and analog outputs AO1–AO4. Inputs are located on the left and outputs on the right. You will notice that on the universal inputs UI1–UI8, the channel type (Chn Type) will reflect the configuration accomplished through web pages.

Both the binary outputs and analog output components have an Enable pin which by default is disabled. This prevents any Sedona logic to manipulate an output until this pin is enabled. This then allows a BACnet client to directly control a physical BAScontrol20 output when using the BAScontrol20 in a BACnet/IP IO controller mode. When using the BAScontrol20 in freely-programmable mode (Sedona), the enable pin on any output component intended to be used with the Sedona logic should be set to true thereby preventing a BACnet client from controlling this output.

The Demo Box application expands the default wire sheet to include virtual points and folders.

Pre-loaded Sedona Application (Demo Box)

A Sedona application called Demo Box has been pre-loaded into the BACcontrol20 that will execute once power is applied. If it is not loaded, it can be obtained from the Contemporary Controls' web site. This program can be freely changed at any time to assist in learning the product. However, if the program is changed and if it is desired to save the changes it must be saved by right-clicking App, then Actions and then Save. This is accomplished in the Navigation Pane under the Sedona (Demo Box) directory. This only saves the program on the BAScontrol20. If you want a copy saved on your Workbench tool do the following.

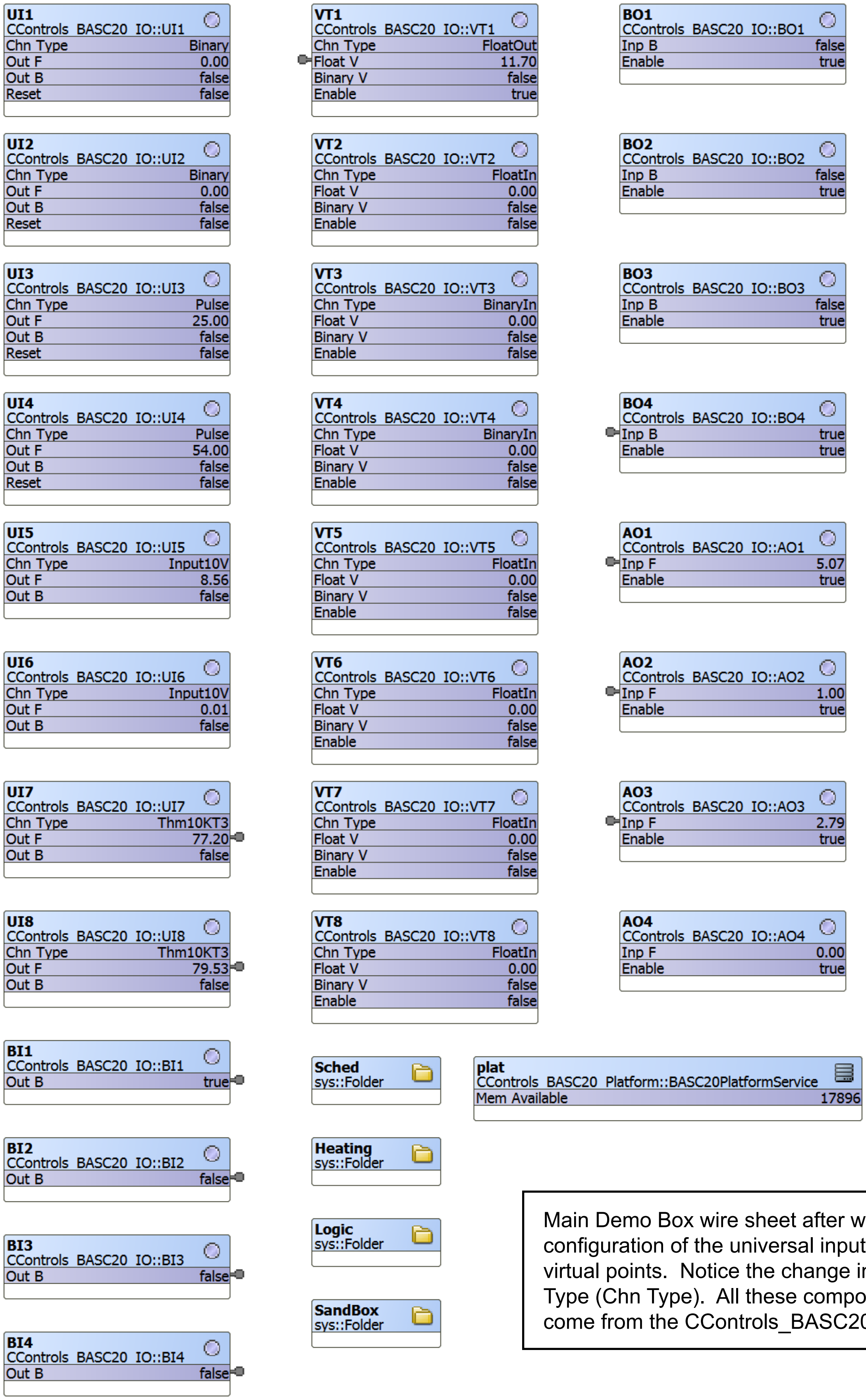
Click on Sedona Tools just under Sedona (Demo Box) in the Navigation Pane. Then click on Application Manager and then Get. You will be presented with a long file name with the words Demo Box in the name. You can change it or accept it. Complete the actions and your program is saved.

If you do not have the Demo Box program, download it from the Contemporary Controls' web site and store it in the directory for saved Sedona programs. On the computer with your Workbench tool, go to your main drive and find your Niagara folder. Then drill down to niagara-version>sedona>store>apps. Put the program there.

To place the program into the controller, click on Sedona Tools just under Sedona (Demo Box) in the Navigation Pane. Then click on Application Manager and then Put. Select the Demo Box program from the list and complete the instructions. You will be prompted to restart the controller. Remember that by doing a Put or a Get, no web page data is loaded or saved. This is accomplished with a different procedure.

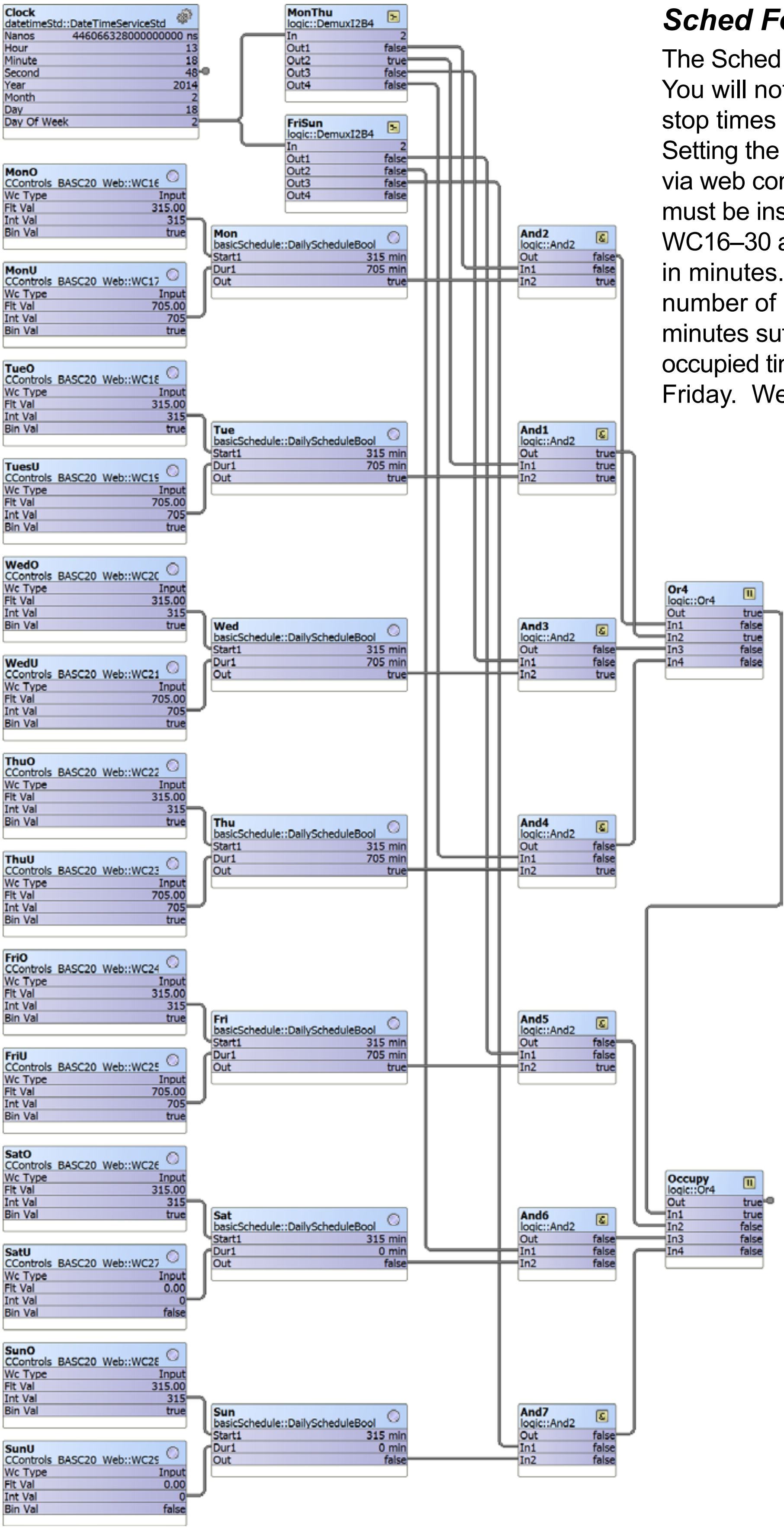
The Demo Box program has all 20 physical IO points and eight virtual points residing on the wire sheet regardless if they are used or not. They are arranged with inputs on the left, outputs on the right with virtual points in the middle because they could be considered as inputs or outputs. If they are not loaded by default, they can be dragged and dropped from the CControls_BASC20_IO kit under the Sedona Palette. The actual applications or portions of the actual applications should be in folders. These folders can by placed from the sys kit. Four folders have been pre-loaded — Sched, Heating, Logic and SandBox. All real outputs have been enabled for Sedona use. What follows is information about the logic in the various folders.

Programming — Main Demo Box Wire Sheet



Main Demo Box wire sheet after web page configuration of the universal inputs and virtual points. Notice the change in Channel Type (Chn Type). All these components come from the CControls_BASC20_IO kit.

Programming — Schedule Folder



Sched Folder

The Sched folder contains logic for a seven-day schedule. You will notice that each day can have flexible start and stop times but there is only one output — called Occupy. Setting the occupied times and duration of the periods is via web components so the CControls_BASC20_Web kit must be installed on the controller. Web components WC16–30 are used for this purpose and must have values in minutes. For occupied times, the value should be the number of minutes past midnight. For duration times minutes suffice. The schedule has been arbitrarily set for occupied times between 9:00am–5:00pm Monday through Friday. Weekend times are set for unoccupied status.

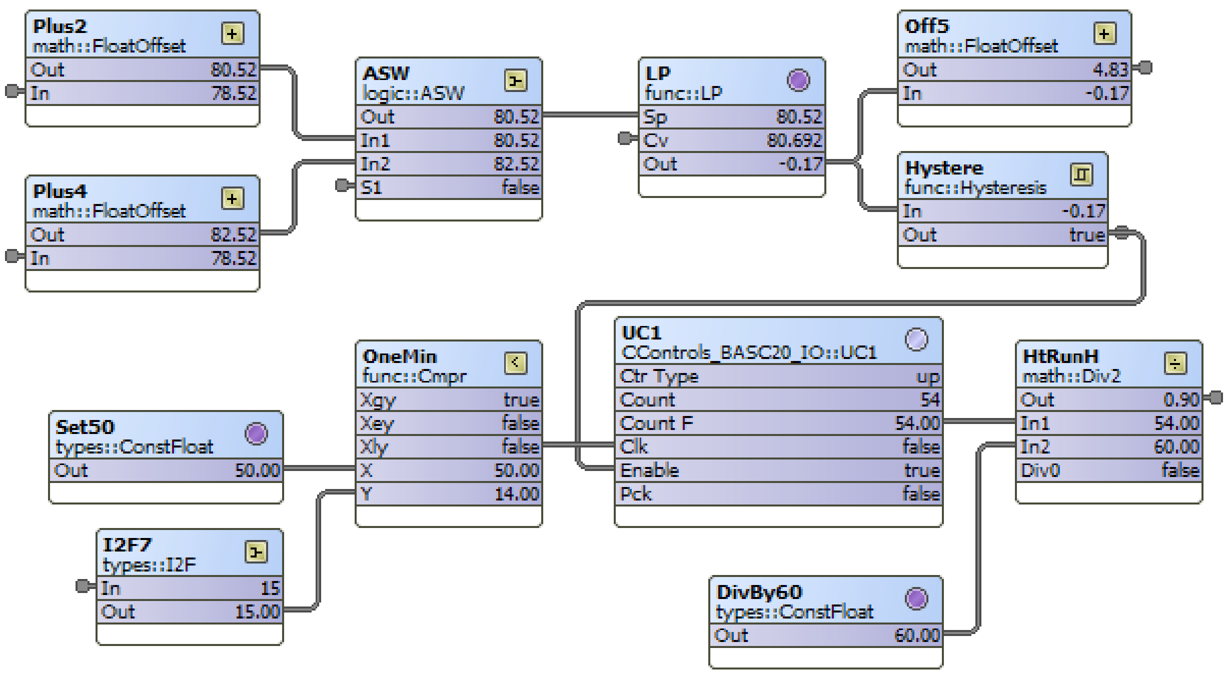
Programming — Heating Folder

Heating Folder

In the heating folder is a simple single loop PID controller driving one digital point (B4). The nubs indicate that the signal is connected to other wire sheet pages such as the main wire sheet or wire sheets inside folders. UI7 on the main wire sheet measures ambient temperature (space temperature) and the intent of this application is to have temperature UI8 track the ambient temperature with either a two-degree positive offset (Plus2) or a four-degree positive offset (Plus4) depending upon the state of the Occupy signal from the Sched folder. These two offset components accept the output from UI7 and feed an analog switch (ASW) which does the switching. During occupied times the setpoint is four degrees above ambient space temperature otherwise it is only two degrees above ambient. UI8 is connected to a thermistor that is thermally bonded to a 10 kΩ resistor that is excited by the 24 VAC supply through a relay contact from BO4. When BO4 is energized, the resistor generates heat that is sensed by the thermistor. This signal becomes the controlled variable to the loop component (LP). The analog output of the loop component is sent to a hysteresis component that converts the signal to binary. You can observe the performance of the control by changing the Kp, Ki and Kd tuning parameters on the loop component as well as the trip points on the hysteresis component.

To obtain a better understanding of the actual output of the loop component, the float output is first sent to an offset component set to +5 (Off5). This provides a bias of 5 V to the output signal which is then sent to analog output AO1 connected to a voltmeter. The jitter you see in the meter is the PID solving its equation at the rate set in the loop component (default is 1000 ms). Although bias can be applied to the loop component itself eliminating the need for the offset component, this bias will go to zero if a nonzero Ki parameter is set in the loop component. Output limits of ±5 have been set in the loop component to restrict the output swing.

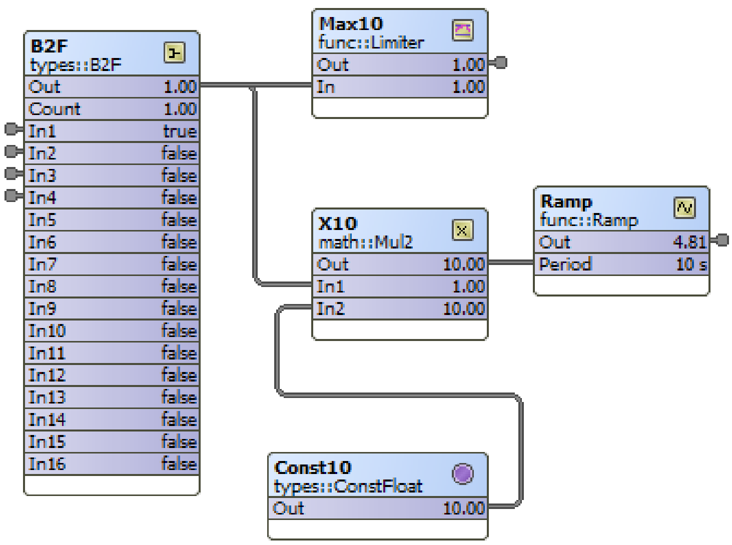
The amount of time in hours that the control calls for heat is measured. A one-minute pulse rate is created by reading the seconds time slot on the clock component in the Sched folder. Since the seconds time is an integer it must first be converted to a float for use by the comparator (OneMin). The other input of the comparator is the arbitrary constant 50. A pulse is created with the number of seconds exceeds 50 remaining true until the seconds time returns to zero. This pulse increments universal counter UC1 assuming the heating is being called. Although the output of the counter is accumulated minutes, the actual monitored time is in hours using a divider (HtRunH) and a constant set to 60 (DivBy60). The accumulated running time in hours is then sent to virtual point VT1 for reading by a BACnet client. VT1 had to be first configured as a wire sheet read from the main web page. UC1 has retentive memory so if power is lost for less than seven days there will be no loss in accumulated running time.



Programming — Logic Folder

Logic Folder

The logic folder contains some very simple logic to demonstrate how binary input points can control analog variables within the wire sheet. A binary to float (B2F) component has four binary input connections (BI1–BI4). BI1 is attached to In1 and BI4 is connected to In4 on the component. Therefore this bank of four switches can be treated as a binary coded switch with BI1 representing the least significant bit (LSB). With all switches set to true, the output of the component would be 15. This output feeds a limiter (Max10) which is set to limit the output to ten so that we do not pin the voltmeter connected to AO2 driven by the limiter. At the same time, the output from the B2F component is sent to a multiplier (X10) that multiplies the output by ten while controlling the period of a triangular ramp (Ramp). This Ramp has its output value limited to ten volts so that it can drive another voltmeter attached to AO3 without pinning the output. By manipulating the four switches, the binary to analog conversion can be observed on meter AO2 and the period of oscillation of the ramp can be observed on meter AO3.



SandBox Folder

The Sandbox folder is empty. It was placed on the main wire sheet to encourage the user to experiment with different applications using the standard Tridium components and the Contemporary Controls components from the various kits. Connect points in the application using the IO from the main wire sheet or even from points in other folders. Do not forget that applications that are intended to be saved must first be saved on the controller (App>Save) and then saved in the Workbench tool (Application Manager>Get).

Conducting Simple Exercises

You do not even need Workbench to try things out on the Demo Box. Using the pre-programmed applications and a web browser you can learn much about how Sedona works and how the BAScontrol20 functions. Using BDT, you can read input and output points on the BAScontrol20 as well as the one virtual point that is pre-programmed. What follows is one example of a simple exercise that can be accomplished with just a web browser.

Retentive Pulse Accumulators

Switches U3 and U4 are directly connected to UI3 and UI4 respectively. Both inputs are configured as pulse accumulators. While observing the main web page and with Auto Refresh ON, toggle either of these switches and observe the counting. The accumulators should increment. Go to the configuration page for each of these points and change the maximum value to test how the counters rollover when they reach the maximum value. Write down the accumulated values for both counters and then cycle power to the demo box. If you reestablish connection to the main web page you should notice that the values did not change.

The Demo Box was constructed using a standard carrying case.

United States
Contemporary Control Systems, Inc.
2431 Curtiss Street
Downers Grove, IL 60515
USA

Tel: +1 630 963 7070
Fax:+1 630 963 0109

info@ccontrols.com
www.ccontrols.com

China
Contemporary Controls (Suzhou) Co. Ltd
11 Huoju Road
Science & Technology
Industrial Park
New District, Suzhou
PR China 215009

Tel: +86 512 68095866
Fax: +86 512 68093760

info@ccontrols.com.cn
www.ccontrols.asia

United Kingdom
Contemporary Controls Ltd
14 Bow Court
Fletchworth Gate
Coventry CV5 6SP
United Kingdom

Tel: +44 (0)24 7641 3786
Fax:+44 (0)24 7641 3923

info@ccontrols.co.uk
www.ccontrols.eu

Germany
Contemporary Controls GmbH
Fuggerstraße 1 B
04158 Leipzig
Germany

Tel: +49 341 520359 0
Fax: +49 341 520359 16

info@ccontrols.de
www.ccontrols.eu