application NOTE

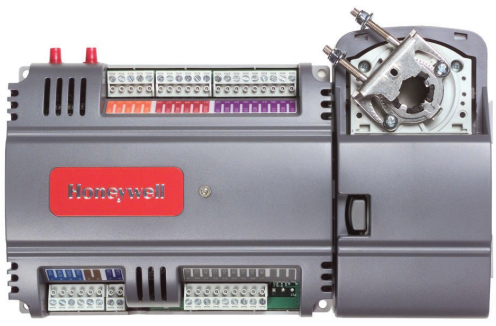
Adding Cube I/O Modules to BACnet Spyder Controllers

Honeywell Spyder VAV and unitary controllers are available with either a LON or BACnet interface. These controllers typically have 21 physical points but this point count can be expanded on the BACnet versions by adding BACnet Cube I/O modules instead of adding another Spyder. Cube I/O modules are low-cost, low-density DIN-rail modules with a variety of I/O point configurations. Installation can be accomplished using a Honeywell WEBstation-AX (Niagara workstation) with an installed Honeywell Spyder Tool for configuration and commissioning. Connecting the WEBstation to the Spyder is facilitated with a Contemporary Controls' BACnet MS/TP to BACnet/IP router. Once the Spyder is commissioned, the BACnet router can be disconnected.

With the standard-size Spyder models there are three which are BACnet MS/TP compliant. The PUB6438S is a unitary controller while both the PVB6436S and PVB6438S are VAV controllers. The PVB6436S has a resident damper actuator while the PVB6438NS requires an external actuator. In this example we are using a PVB6436S. All models can be expanded with the use of BACnet Cube I/O Modules.

For I/O expansion, we will use a BMT-DIO4/2 Cube I/O Module with four binary contract closure inputs and two relay outputs. The four input points can be used for additional Spyder inputs and the two output points can be used by the Spyder for additional outputs.

Cube I/O BMT-DIO4/2
BACnet MS/TP Mixed Module



Honeywell Spyder

Wiring

Figure 1 provides detail on connecting the various devices for this demonstration. A portable BASrouter is shown but a DIN-rail BASrouter will work as well. With the portable BASrouter, power from the router is derived from a USB port on a workstation or laptop computer. With the DIN-rail BASrouter, a 24VAC connection can be shared with the Spyder and Cube I/O from an adequately powered 24V control transformer. Make sure that COM (GND) connection integrity is observed as shown in figure 1.

Wire the BACnet connections as shown observing proper polarity of the MS/TP connection. For this bench test we will float the shield and ignore the need for a 120 ohm termination resistor.

Both the Cube I/O and Spyder utilize non-isolated EIA-485 transceivers while the BASrouters do not.

Therefore connect system common (SC) on the BASremote to a COM point. You can wire in switches for the Cube I/O or you can just use jumper wires. The two Cube I/O outputs have two LEDs which turn on when the output is true. This is what we will observe. Leave the two hand-off-auto switches in the middle position.

We need a workstation or laptop with Honeywell's WEBstation-AX software installed along with Honeywell's Spyder Tool. Ethernet on the computer can attach directly to the BASrouter or via an Ethernet switch.

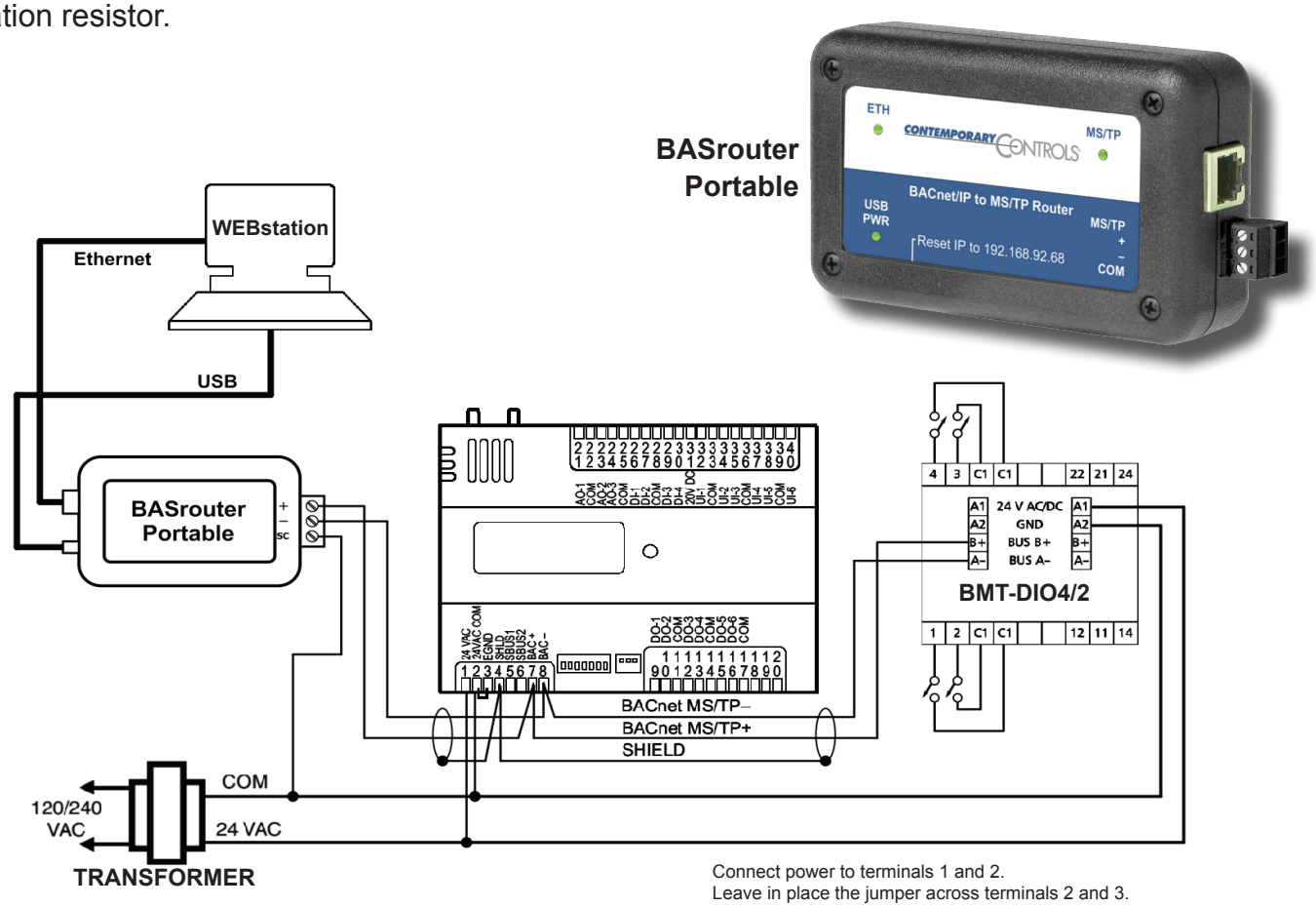


Figure 1 - Demonstration Wiring

Configuration

Configuration is necessary for all the components.

Cube I/O

The default baud rate is 9600 baud for the Cube I/O. Change the baud rate to 38,400 through the use of the two hexadecimal rotary switches X1 and X10. Refer to documentation on the BMT-DIO4/2 which can be found on the Contemporary Controls' website. First set X10 to "E" and then turn X1 to "C" which is the setting for 38,400. Next return X10 to "F". After one second you should see the red and green lights alternatively blinking. Next set the network address which in our case is 06. Set X10 to "0" and X1 to "6" and you are finished. There is a jumper below the faceplate. You can leave it in its default position of GND.

The Cube I/O module is set for BACnet device instance 421000+MAC address. Since the MS/TP MAC address we assigned is 06, the resulting BACnet device instance is 421006.

BASrouter

The BASrouter and the workstation (or laptop computer) must be on the same subnet. Make sure this is the case when assigning IP addresses to the workstation and BASrouter. Using the web browser on the workstation access the BASrouter home page. In our example the BASrouter is set for 10.0.0.237. If you can reach the BASrouter you are on the same subnet. Study the configuration in figure 2 at the right.

Device instances must be unique so assign a unique device instance. We need unique BACnet/IP and MS/TP network assignments. Usually the default "1" is good enough for the BACnet/IP. For the MS/TP network we assigned the value "2345." Set the baud rate to 38400 if it is not already selected. Make sure you save you changes. Your BASrouter is now configured.

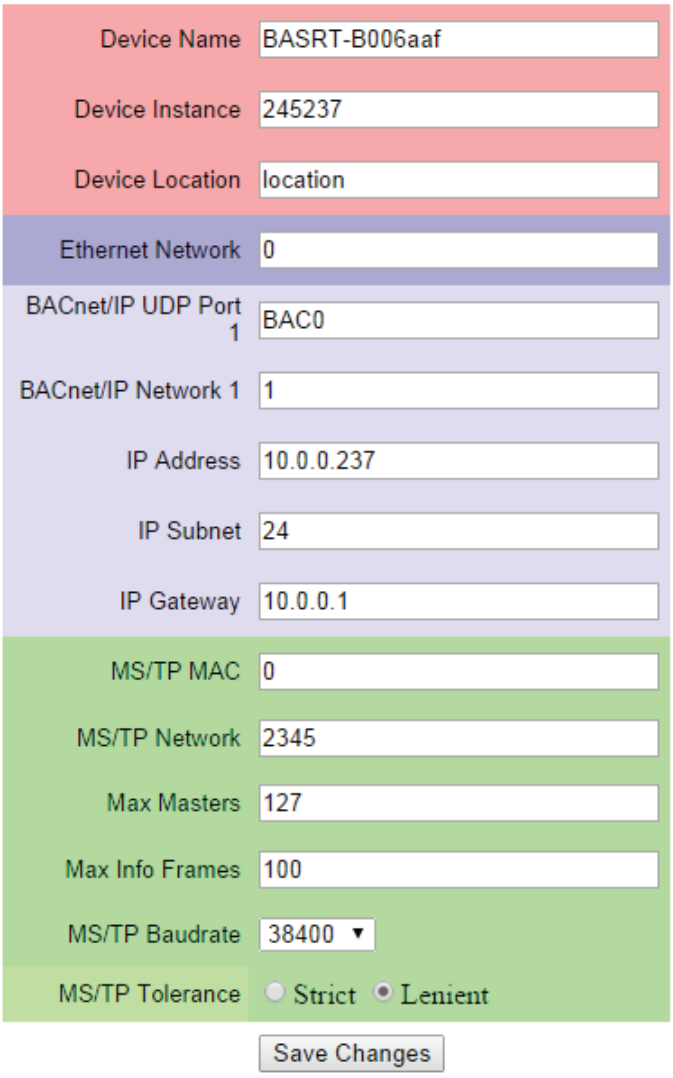
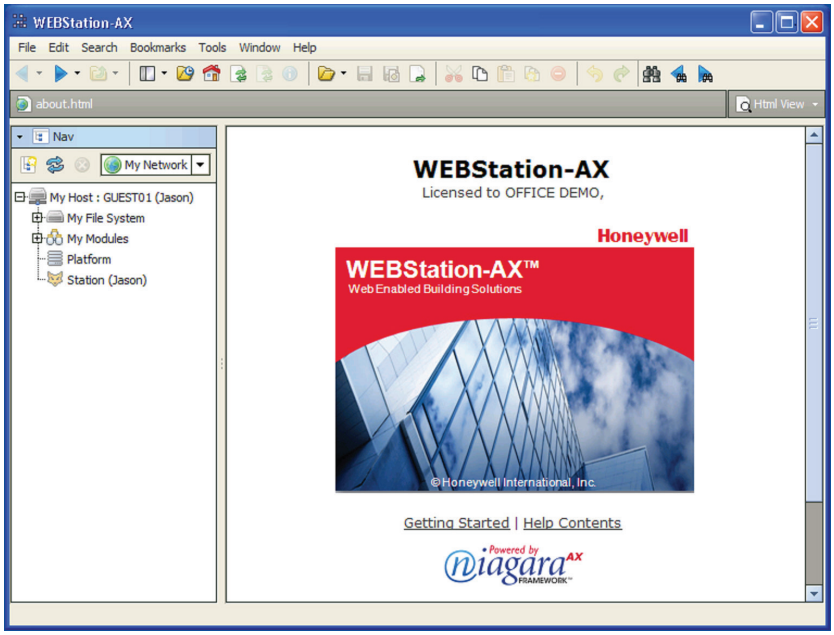


Figure 2 - BASrouter Configuration Screen

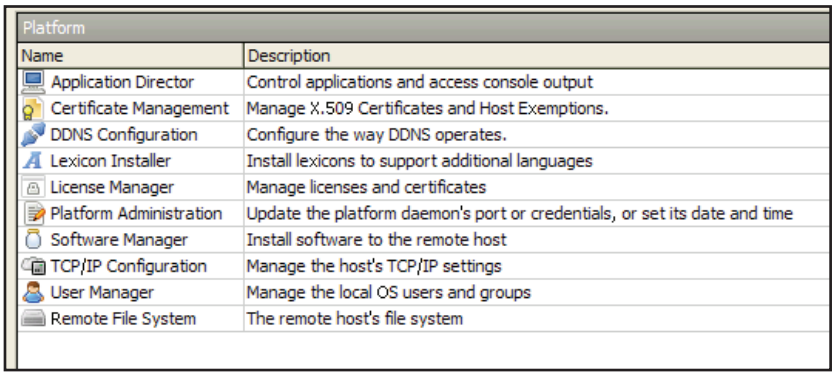
WEBstation

Any Spyder configuration, other then MAC address which is assigned by DIP switches, will occur through the WEBstation. The Spyder autobauds so it will Link up with the BASrouter once it sees network traffic.

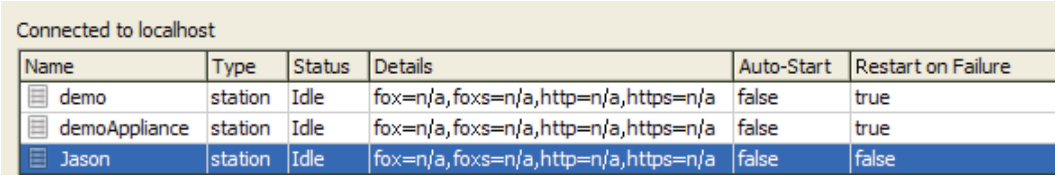


Next we need to start the WEBstation-AX program which should have already been installed on the workstation or laptop being used.

If the Navigation pane is not visible, go to Side Bar just below Bookmarks and bring it down. Expand the tree under My Host and double-click on Platform and enter your credentials for this workstation.



After successfully logging into Platform you will see a listing of Platform programs. Click on Application Director at the top which will show a list of pre-generated stations.

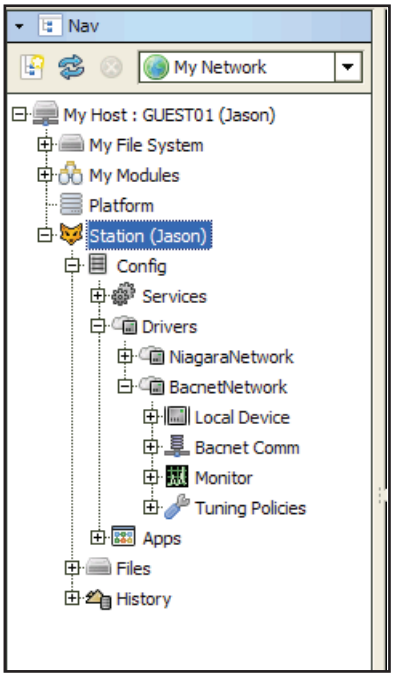


This application note assumes that a station has previously been created and only needs to be started or possibly restarted. If this is not the case, consult your WEBstation-AX documentation on creating a station. In our case we created a station called Jason that needs to be started. Highlight the station details and click on Start on the right side of the screen. Once the station is started you should notice a confirmation message in the main window.

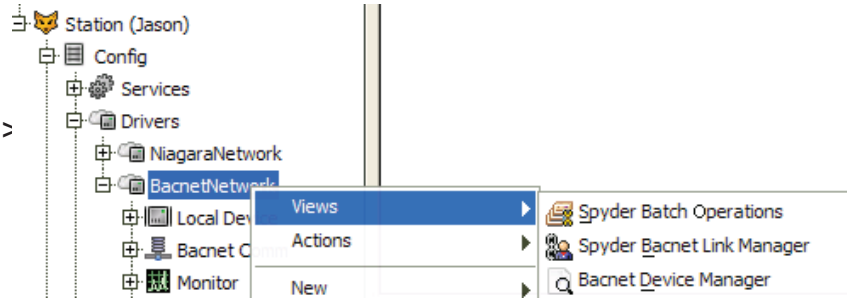
In the navigation pane on the left side click on Station (Jason) and enter the credentials that were created for this station.

Once you log into the station you can expand the navigation tree under Station (Jason).

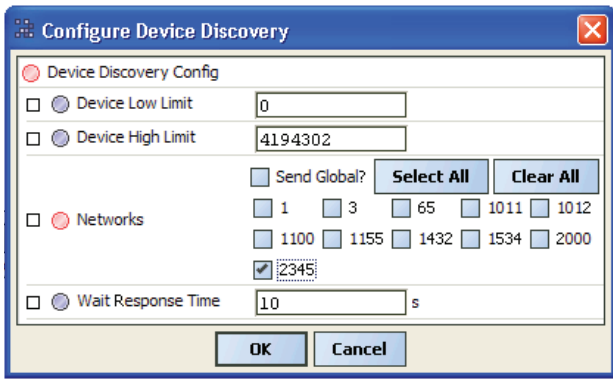
Expand the tree: Station > Config > Drivers > BacnetNetwork as shown in the figure to the right.



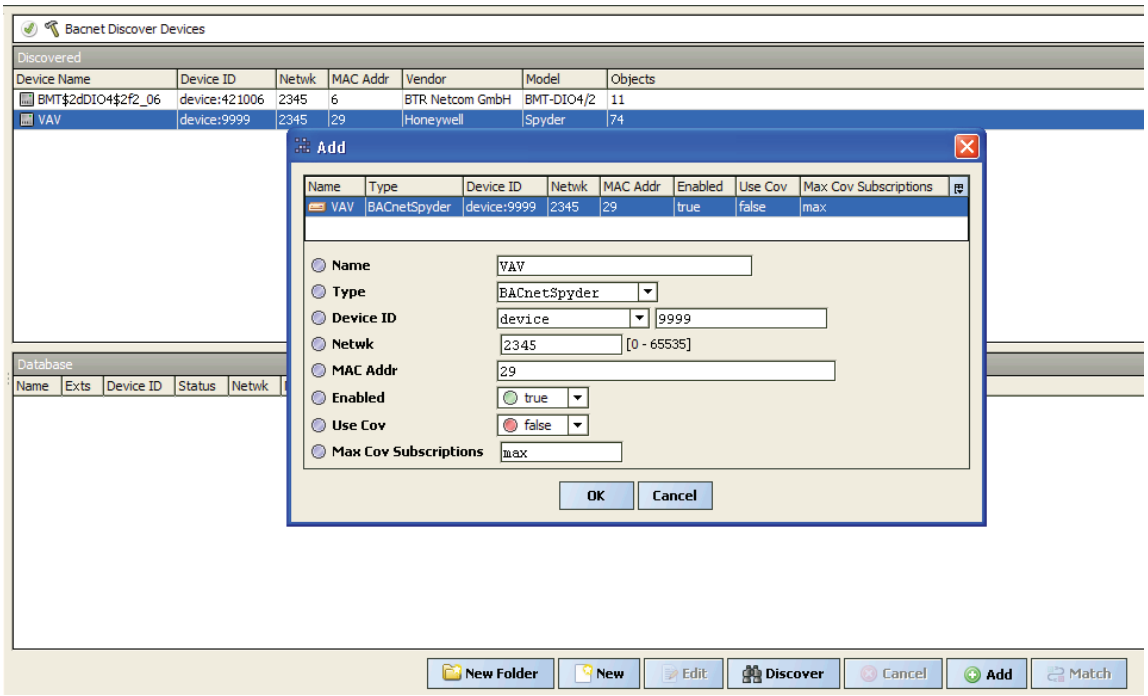
Next we need to discover our BACnet devices. Go to BacnetNetwork > Views > Bacnet Device Manager.



Once it is brought up click on Discover.

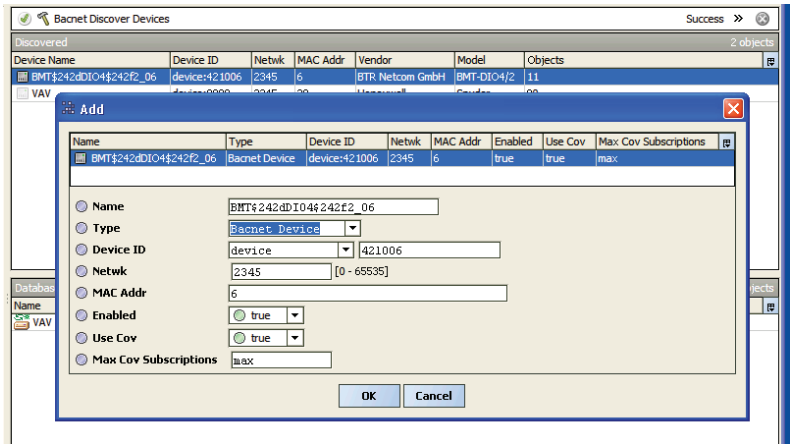


You will have a choice of networks to discover. We are only interested in network 2345 which is the MS/TP network we selected with the BASrouter. Therefore we cleared all other networks. Once this network is selected click OK.

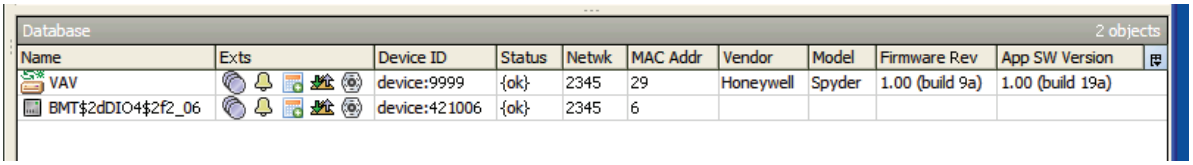


Two BACnet devices are discovered. Double-click the VAV entry which will bring up its device details. This is our Spyder controller. You can change the name of the Spyder during this step. Notice that the default Device ID is 9999 and the MAC address is 29. Click OK and it will be added to the Database pane.

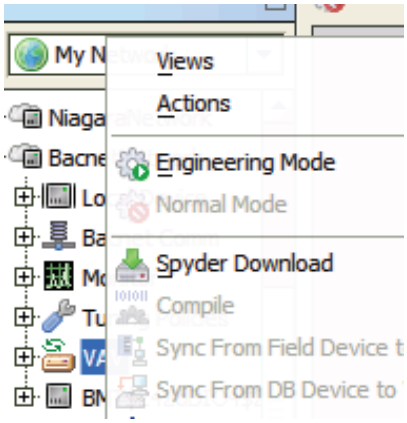
Now double-click on the other device which is the Cube I/O. The WEBstation may think it is a Spyder device so change the Type to Bacnet Device to avoid confusion. Retain all discovered data and click OK and it will be added to the Database pane.



Once your two devices are in the database they will appear in the tree under BacnetNetwork.

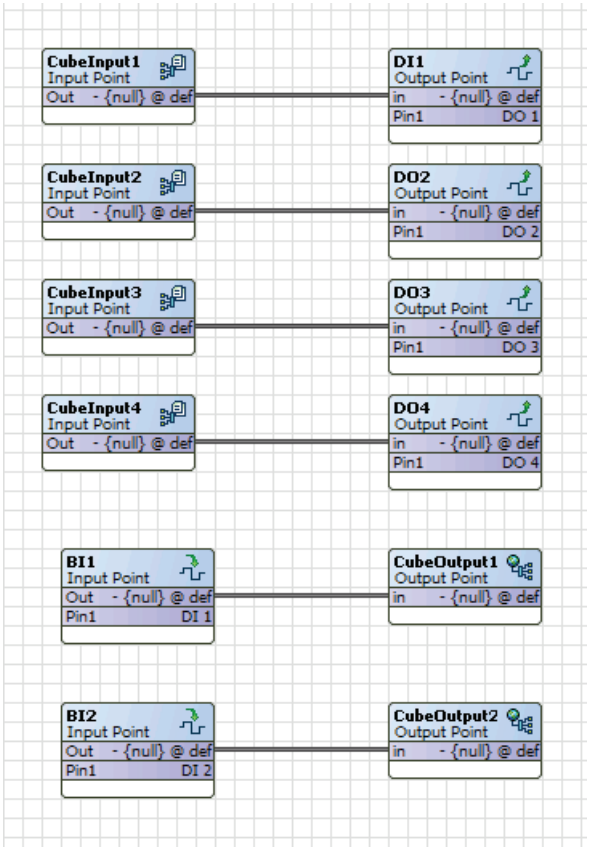


If the Palette is not being displayed on the left side bring it down by again going to Side Bar under the word Bookmark. Under Palette is the Open Palette icon. Click this to open up all the Palette options. Select HoneywellSpyderTool. If the Honeywell Spyder Tool cannot be found it will need to be installed in the WEBstation.



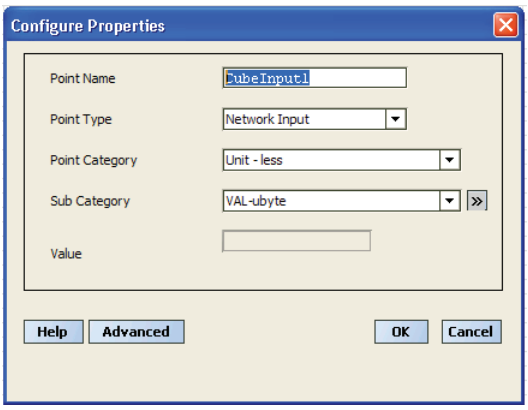
Right-click on VAV in the navigation pane and select Engineering Mode in order to reach the wiresheet.

The alternative to Engineering mode is Normal mode but in order to put down logic onto a wiresheet we need engineering mode.

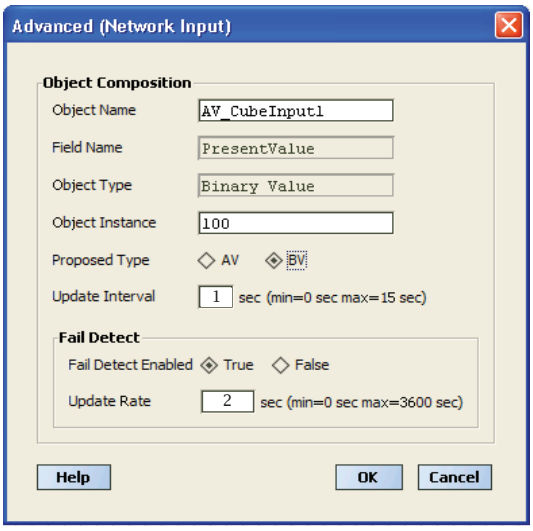


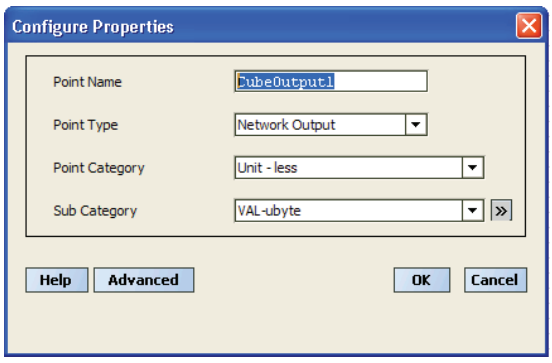
From the SoftwarePoints folder drag and drop four Network Input components onto the wiresheet and rename them CubeInput1, CubeInput2, CubeInput3 and Cubeinput4. From the same folder drag and drop two Network Output components and rename them CubeOutput1 and CubeOutput2. From the Physical point folder drag and drop two physical input points and four physical output points and link them as shown in the nearby figure. You can rename the Spyder physical points as shown for clarity. Our logic is simple. Each Cube I/O input point drives a corresponding Spyder output point while two Spyder input points individually drive two Cube I/O output points.

Next we need to configure our network variables beginning with the Network Inputs. Right-click each of the objects and select Configure Properties. Change the Point Category to Unit-less and the Sub Category to VAL-ubyte and the then click Advanced.

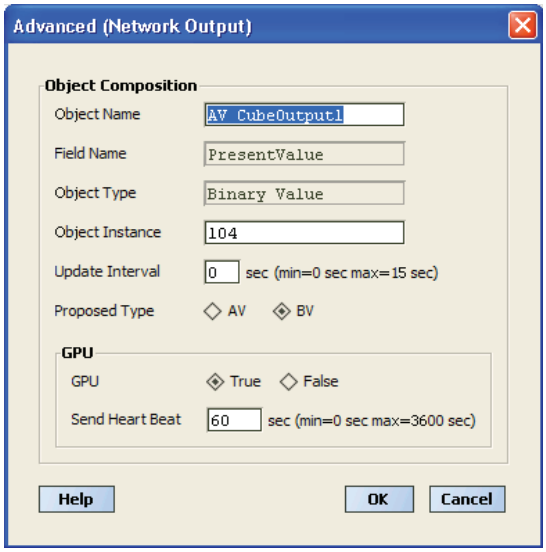


On the second screen change the Proposed Type to BV. Notice that the Object Name has been assigned for you as has the Object Instance. Change Update Interval to 1 and Update Rate to 2. Click OK on both screens and do the same procedure for all four Network Inputs. Unique object instances will automatically be assigned for you.





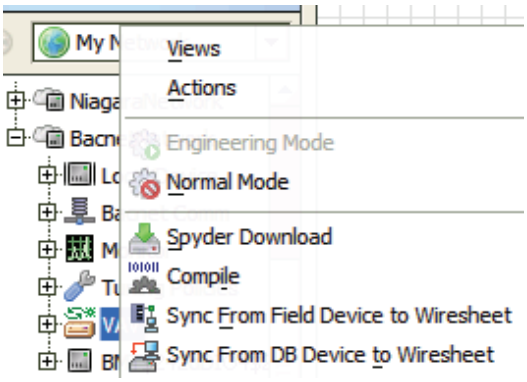
Next configure the two Network Outputs in a similar fashion although the screens are different. On the first screen again select Unit-less and VAL-ubyte and then click Advanced.



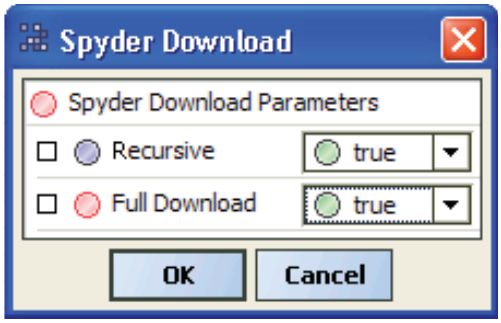
On the second screen change Proposed Type to BV while leaving all other settings unchanged. Click OK on both screens. Follow the same procedure for the other output.

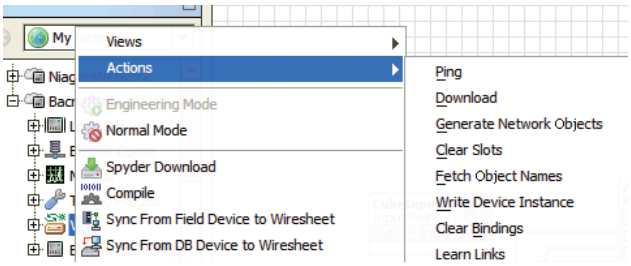
You can check the configuration for the Spyder physical inputs and outputs. Just make sure the proper pins are assigned. You can accept the default values.

The logic you just created along with the configurations are not in the Spyder. If you look at the Spyder icon (VAV) in the navigation pane you will notice a red down arrow on top of the icon meaning you must download the changes to the wiresheet by right clicking on the VAV icon and selecting Spyder Download. After any changes, you need to download the Spyder once you are finished to effect the changes.



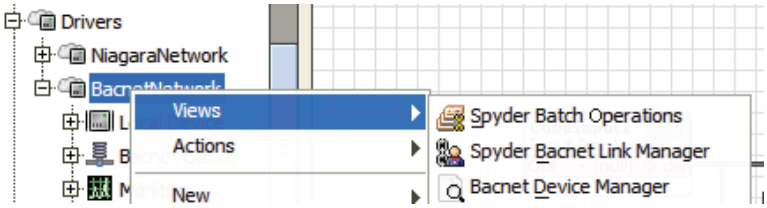
You will be asked to make a choice of a full download or an incremental download assuming you are appending an existing program. Since this is our first download select Full Download. Once download is completed the top of the VAV icon will turn green. When in doubt do a full download.



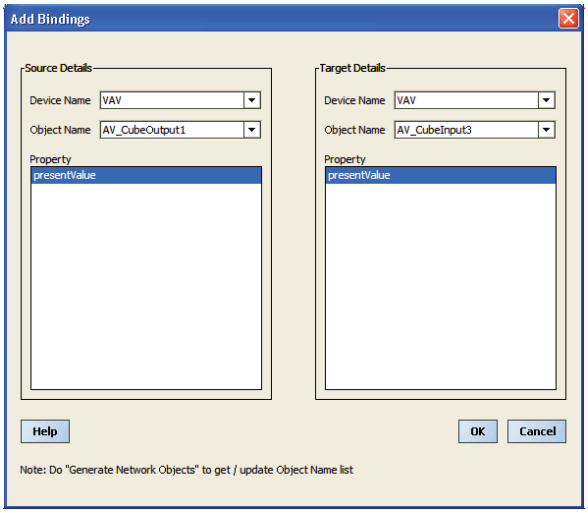


You are not finished. Next you need to again right-click VAV and then Actions and then click Generate Network Objects. You will not receive a confirmation of this action. Once this is completed we can do data sharing.

Go to BacnetNetwork > Views > Spyder Bacnet Link Manager.

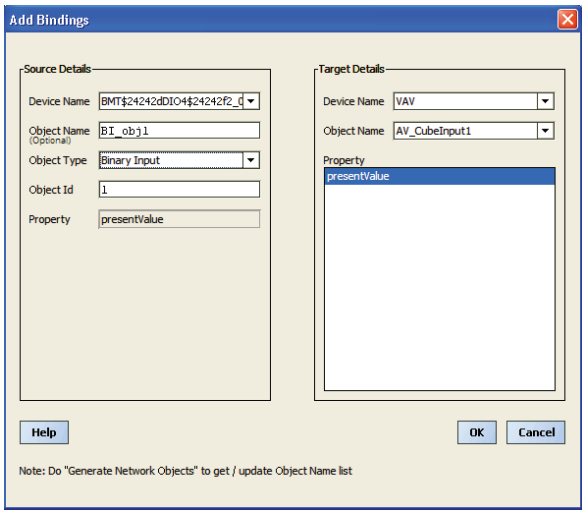


Click Add to get you to the Add Bindings window. You will notice a screen with some default values. For Source and Target columns you will see VAV displayed as the device which is our Spyder controller. On the Source side you will notice one of the network variables – in this case one of the network outputs we created. On the other side we see one of our network inputs we created. What this means is that the Spyder binds its network outputs as sources to third-party BACnet output targets and that third-party BACnet input sources bind to Spyder network input targets.

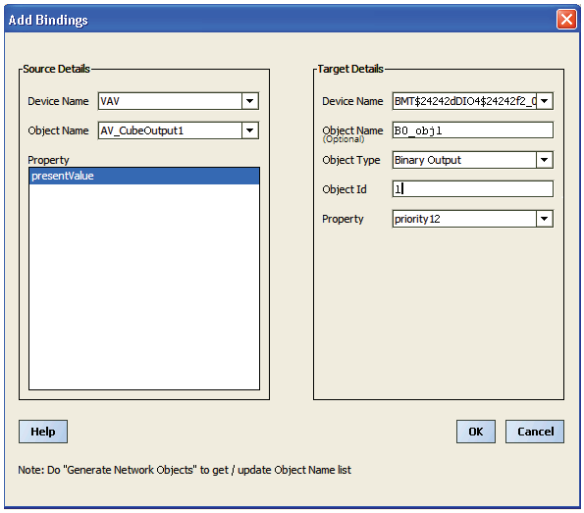


We will do inputs first. Under the Source section use the Device Name drop-down to find the Cube I/O. Use the Object Type drop-down to select binary input because the Cube I/O has four binary inputs with object identifiers from 1 to 4. Enter Object ID 1 and you will notice that the Object Name will change accordingly. Under the Target section use the Object Name drop-down to find AV_CubeInput1. Click OK and you will notice your first binding appearing as a New Link.

You need to keep adding links for the remaining three inputs assigning unique Object IDs that match the Cube I/O points. This process requires you to know the object IDs for the points being bound to Spyder network variables.



When doing the output side the device roles change. VAV will be the Source and the Cube I/O the target. There are only two physical outputs to the Cube I/O which are identified as BACnet object type Binary Output with Object identifiers 1 and 2. On the Spyder side you have AV_CubeOutput1 and AV_CubeOutput2. Enter the corresponding bindings. There is one other decision to make and that is the Priority property. It defaults to 12 but change it to 11. A value of 1 represents the highest priority while a value of 16 is the lowest.

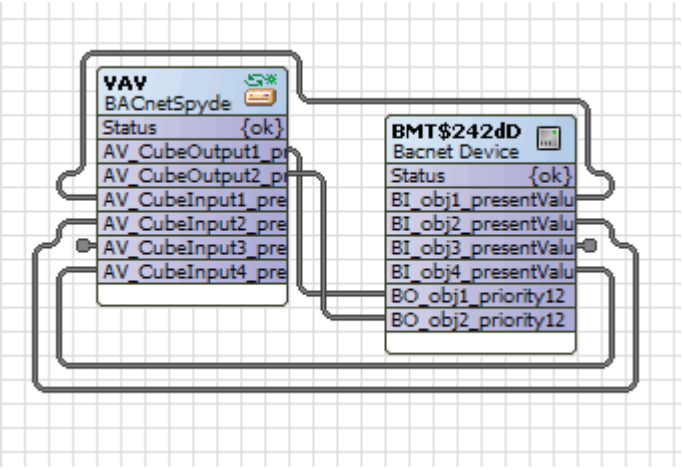


Once you complete the two other links your link list should look like the nearby image. Verify that this is what you want. If you made a mistake, delete the link and re-enter the data. Once you are finished click Bind.

A successful binding results in the Link Status column for all entries changing from New Link to Bound.

Link Status	Device Status	Source Device	Source Object	Source Property	Target Device	Target Object	Target Property	Poll/Push
New Link	Downloaded	BMT$24242dDIO4$24242f2_06	BI_obj1	presentValue	VAV	AV_CubeInput1	presentValue	Poll
New Link	Downloaded	BMT$24242dDIO4$24242f2_06	BV_obj2	presentValue	VAV	AV_CubeInput2	presentValue	Poll
New Link	Downloaded	BMT$24242dDIO4$24242f2_06	BI_obj3	presentValue	VAV	AV_CubeInput3	presentValue	Poll
New Link	Downloaded	BMT$24242dDIO4$24242f2_06	BI_obj4	presentValue	VAV	AV_CubeInput4	presentValue	Poll
New Link	Downloaded	VAV	AV_CubeOutput1	presentValue	BMT$24242dDIO4$24242f2_06	BO_obj1	priority 11	Push
New Link	Downloaded	VAV	AV_CubeOutput2	presentValue	BMT$24242dDIO4$24242f2_06	BO_obj2	priority 11	Push

If you go to Bacnet Network > Views > Wire Sheet you will see a graphical view of your bindings.



You can now try out your application program. Switching one of the input points on the Cube I/O will energize the Cube I/O output 2 while switching on all four input points will also energize Cube I/O output 1.

Some Hints

It is possible when doing your testing with the Cube I/O that the priority array for the two output points in the Cube I/O may get set with an unknown priority that is higher (lesser number) than the value 11 used in the application note. The result is that the Cube I/O appears not to set properly by the Spyder because the LEDs do not change state. If the Spyder outputs a value at priority 11, and in the Cube I/O is stored a value at a higher priority level, the output will not change. The simple solution is to power cycle the Cube I/O to clear out any saved priority level and bring it back online. The Spyder will then regain control of the Cube I/O.

When changing wire sheet logic involving network variables or when reconfiguring network variable properties, we found it beneficial to first unbind any network bindings. Binding network variables is the last step in the programming process and we feel that these bindings should be unbound as a first step when contemplating a change to any of the network variable logic. Once changes are made, you can generate your network variables again and then bind them as the final step. We also chose to do full downloads to the Spyder versus incremental downloads.

The Spyder solves its wire sheet about once a second. Network variables appear to be treated differently from that of internal I/O points. You will notice an Update Rate for the Network Input variables. We needed to change the default value of 60 seconds down to 2 seconds in order to achieve a reasonable response rate. In our example we are using internal Spyder physical inputs to drive Cube I/O output points via Spyder network output variables while using Cube I/O input points to drive Spyder physical output points via Spyder network input variables. The response times in the two scenarios were different. In the first instance, a change to a Spyder input resulted in a Cube I/O output change in about 2 seconds. In the second instance, a change to the Cube I/O input resulted in a change to the Spyder output in about 4 seconds. Considering the internal cycle time of the Spyder solving its logic and the fact that network commands are occurring over a lower speed MS/TP network, the overall response times seem acceptable.

This application note demonstrates that Cube I/O modules can be successfully deployed as expansion I/O for Spyder controllers.

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