ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA159

INTEGRATING CONVENTIONAL FACILITIES SYSTEMS VIA BACnet*

S. B. Webb†, ORNL, Oak Ridge, USA

Abstract

Conventional Facility (CF) controls, such as those used for Heating, Ventilation, and Air Conditioning (HVAC) are often developed and operated independent of the accelerator control system using commercial Building Automation Systems (BAS). At the Spallation Neutron Source (SNS), these systems are fully integrated into the machine control system based on the Experimental Physics and Industrial Control System (EPICS) toolkit. This approach facilitates optimal machine performance and enhances reliability. BACnet (Building Automation and Control Networks) is the predominant communication protocol used in the building automation industry, thus inspiring SNS to develop a BACnet/IP software driver [4] for EPICS to enable this integration. This paper describes how SNS uses BACnet and standard EPICS tools to perform custom (complex) chiller sequencing and data collection to study the cooling system performance.

THE SNS CHILLED WATER SYSTEM

The central cooling plant at SNS [1] consist of four 1,200-ton Trane centrifugal chillers, four chilled water pumps, four condenser water pumps, and two cooling towers. The plant provides chilled water to accelerator equipment and non-accelerator equipment (e.g., office complex).

History

The entire chilled water system was originally controlled independent of the accelerator control system by proprietary vendor software. The software was Windows dependent and installed on a local desktop computer in a remote location, separate from Central Control Room (CCR). This remote computer was the only interface to the chilled water system, its alarms and historical data. The sample rate and number of samples of the historical data were restricted by the hardware limitations of the local desktop computer.

PROBLEMS

- Single operator interface to the chilled water system
- Not readily accessible to operations staff
- No direct control or monitoring of the system from EPICS
- Reliance on offsite vendor to resolve problems
- Limited historical data
- Unseen/unknown alarms

BUILDING AUTOMATION

Almost all commercial and industrial heating/cooling equipment is designed and built to work with building automation control systems, such as those provided by Johnson Controls, Honeywell, Trane, Carrier, etc.

The building automation industry standardizes a few communication protocols known as LonWorks, Modbus and BACnet [2] — BACnet being the most popular and the default protocol embedded in nearly all HVAC equipment.

EPICS SUPPORT FOR BUILDING AUTOMATION

Originally, EPICS [3] had no support for building automation. Any integration to HVAC equipment required external relays, instrumentation, and/or third-party gateway devices. This type of integration was slow and provided minimal datapoints from the system. This left some display screens incomplete and operators questioning certain events.

By developing an EPICS BACnet/IP driver [4], these HVAC systems can be fully integrated, directly monitored, and controlled from EPICS. This increases system flexibility, system performance, and improves system understanding and reliability.

INTEGRATION VIA BACNET

After developing the BACnet/IP driver [4], SNS fully integrated the chilled water system into EPICS, and have removed the vendor PC running proprietary software. SNS now manages their entire central cooling plant (chillers, pumps, and cooling towers) independently with an EPICS soft-IOC (Input Output Controller). The soft-IOC interfaces with the chillers directly via their native communication bus, BACnet and also communicates with an Allen-Bradley PLC that controls the pumps and the cooling towers. This made it possible to view and control the entire chilled water system from EPICS based OPIs (Operator Interface) in the CCR.

DRIVER DETAIL

Define a BACnet Device (st.cmd)

Each BACnet device requiring integration must be defined in an EPICS st.cmd file: '*drvBACnetDefineDevice("Chiller1", 21, "eth0", 47808)*'. The definition will include a user given EPICS name (Chiller1) that is unique in your st.cmd file for all devices defined, a BACnet device address/identifier (21), the network interface name of your ethernet card (eth0), and a port number (47808).

* Work supported by Oak Ridge National Laboratory for UT-Battelle, LLC, under contract DE-AC05-00OR22725 for the U.S. Department of Energy.
†webbsb@ornl.gov

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA159

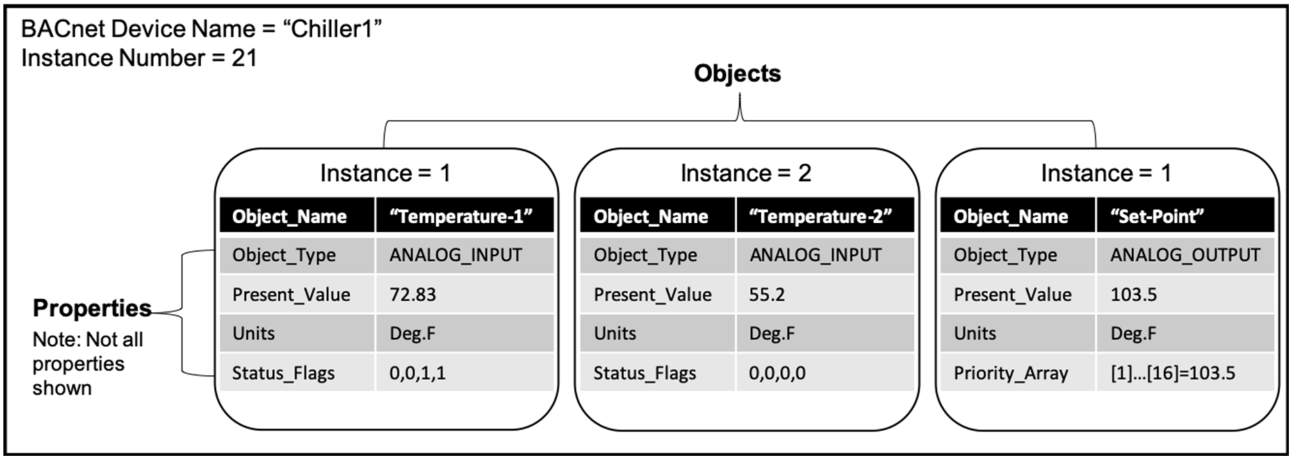

Figure 1: A BACnet device named "Chiller1" has two Analog-Input objects representing temperature values and one Analog-Output object representing a temperature set-point. Each temperature value can be uniquely identified by the object-type (ANALOG_INPUT or ANALOG_OUTPUT), object-instance (1-2) and object-property (Present_Value).

BACnet Objects and Properties

BACnet uses an object-oriented model for abstracting and representing device information. The object model consists of an object type, object instance, and object property. The object type represents the base datatype of the information, while the object instance uniquely identifies a specific object amongst multiple objects of the same type. The object property is a unique characteristic of the object. See Fig. 1 for an example of objects and properties.

EPICS Device Support

The DTYP field of the EPICS record must be set to "BACnet" and the INP field will contain a parameterized string where the first parameter is the device name that was defined in the st.cmd file. The second parameter is the object type, the third parameter is the object instance, and the fourth parameter is the property identifier. Note: the object type and property identifier are standard enumerations that are defined in the BACnet standard. For example, Analog_Input has an enumeration of zero (0), and the Present_Value property has an enumeration of eighty-five (85). Below is an example of an EPICS ai-record that would read the Present_Value property of an Analog_Input object, instance number two, from Chiller1.

```
record(ai, "Chiller1:Temperature-2")
{
 field(DTYP, "BACnet")
 field(INP, "@Chiller1 0 2 85")
 field(SCAN, "1 second")
}
```

Writing values to commandable BACnet objects and properties from EPICS is similar to reading with the excep-tion, that writes require a priority level in the range of 1–16 to be written with the value. 1 being the highest priority and 16 being the lowest priority. Below is an example of an EPICS ao-record that would write to the Present_Value property of an Analog_Output object, instance number one, of Chiller1, at priority level 16.

```
record(ao,"Chiller1:TemperatureSetpoint")
 {
   field(DTYP, "BACnet")
   field(OUT, "@Chiller1 1 1 85 P=16")
   field(SCAN, "Passive")
 }
```

Driver Execution

When the driver is executed it takes the network interface name and port number that were defined in *'drvBACnetDefineDevice("Chiller1", 21, "eth0", 47808)'* and creates a datagram socket and binds to the port.

Next the driver will need to learn the network address of each BACnet device defined. This dynamic binding process is accomplished by taking each device identifier (ID) defined in the st.cmd file and encoding it into a BACnet service request message called "Who-Is". The Who-Is request is broadcasted to all reachable BACnet networks and is received by all BACnet nodes on the network. If a node receives a Who-Is request and the node's device ID matches the ID that is encoded in the request, then the node replies back to the issuer of the request with an "I-Am" response. This response has the node's physical network address encoded. If the node's device ID does not match the device ID encoded in the Who-Is request, then the node discards the request and no response is issued.

Once the driver has learned the network address of the node, the driver can then send BACnet messages directly to the node.

Next the driver needs to determine which BACnet service it will use to read properties from the device: Read-Property (RP) service or Read-Property-Multiple (RPM) service. RP issues individual read requests for every object and property in the device. RPM optimizes and reduces the number of read requests by grouping several objects and properties into a single read request.

To make this determination the driver must first issue an RP-request to each device to read the Services-Supported

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA159

property of the device's Device_Object and see if the RPM flag is set to true or false.

If the RPM flag is false, this means that the device does not support the Read-Property-Multiple service and therefore individual RP requests for each of the objects and properties must be used.

However, If the RPM flag is true, the driver needs to figure out how many objects and properties can fit into a single RPM request. It must then construct as many RPM request as necessary to support the total number of objects and properties being read.

The size of an RPM request is restricted by a buffer limit in the device known as the Max-APDU-Length-Accepted property. The value of the Max-APDU-Length-Accepted property was encoded in the device's I-Am response to the driver during the dynamic binding process.

The driver must know the datatype and length of each property being read before it can construct an RPM request so that the size remains below the size limit set by the Max-APDU-Length-Accepted property of the device. In order to obtain the datatype and length information of the properties, the driver will iterate once through the list of objects and properties defined for each device and issue individual RP request for each of those properties. The datatype and length information are returned in the response to the RP request.

Now that the driver knows specifics about the data, the driver can efficiently construct RPM requests that are below the size limit stipulated by the device.

When writing data to commandable BACnet objects and properties, the EPICS driver uses the BACnet Write-Property service. The Write-Property service is only issued when the EPICS record is processed.

Synchronization between the driver thread and the device support thread is asynchronous.

MANAGING THE PLANT

Managing the chillers in EPICS is accomplished with an EPICS sequencer. The EPICS sequencer handles the input selections from operators for sequencing the chillers and the required automatic functions that happen without operator intervention.

Complex sequencing logic was created to stage the chillers off and on based on operator setpoints and cooling demands of the accelerator. Logic was also created for failover and redundancy of the chillers. This has increased reliability and machine performance. The flexibility of EPICS allows any process variable (PV) to be used for controlling the chillers as opposed to the vendor supplied software.

Using this method, control of the chillers is fully integrated with the larger SNS machine control system alongside all other accelerator subsystems. This integration allows operators to control and monitor the chillers from the CCR, chiller data to be archived and alarms to be configured to prompt operators to take action to handle off-normal conditions. Providing the chiller data in EPICS along with all other accelerator data then allows analysis that could, for example, correlate an equipment failure to a rise in temperature. The system no longer requires someone to check the status in another building of look at data from to two separate systems. See Fig. 2 for single chiller integration and Fig. 3 for chiller plant integration.

CONCLUSION

The EPICS BACnet driver [4] can now be extended for equipment designed for building automation control systems. The integration of CF systems has proven to be a factor in achieving the accelerator operational goals of 90% availability and 1.4MW beam on target. The ability to control, monitor, and react to CF-equipment issues without relying on outside vender support plays an important role in reducing accelerator downtime. CF PVs have alarms to indicate equipment issues in the central control room and allow operators to react accordingly. By archiving CF data, correlations between CF systems and accelerator systems can be derived and used for machine tuning to improve performance. SNS has integrated all CF subsystems into EPICS.

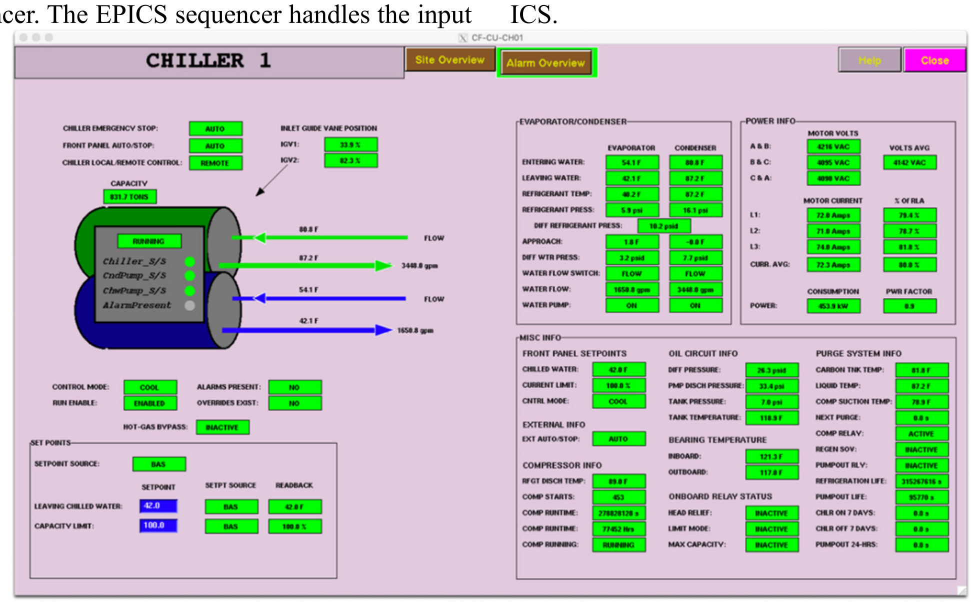

Figure 2: Trane centrifugal chiller with over 80 BACnet datapoints integrated into EPICS.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA159

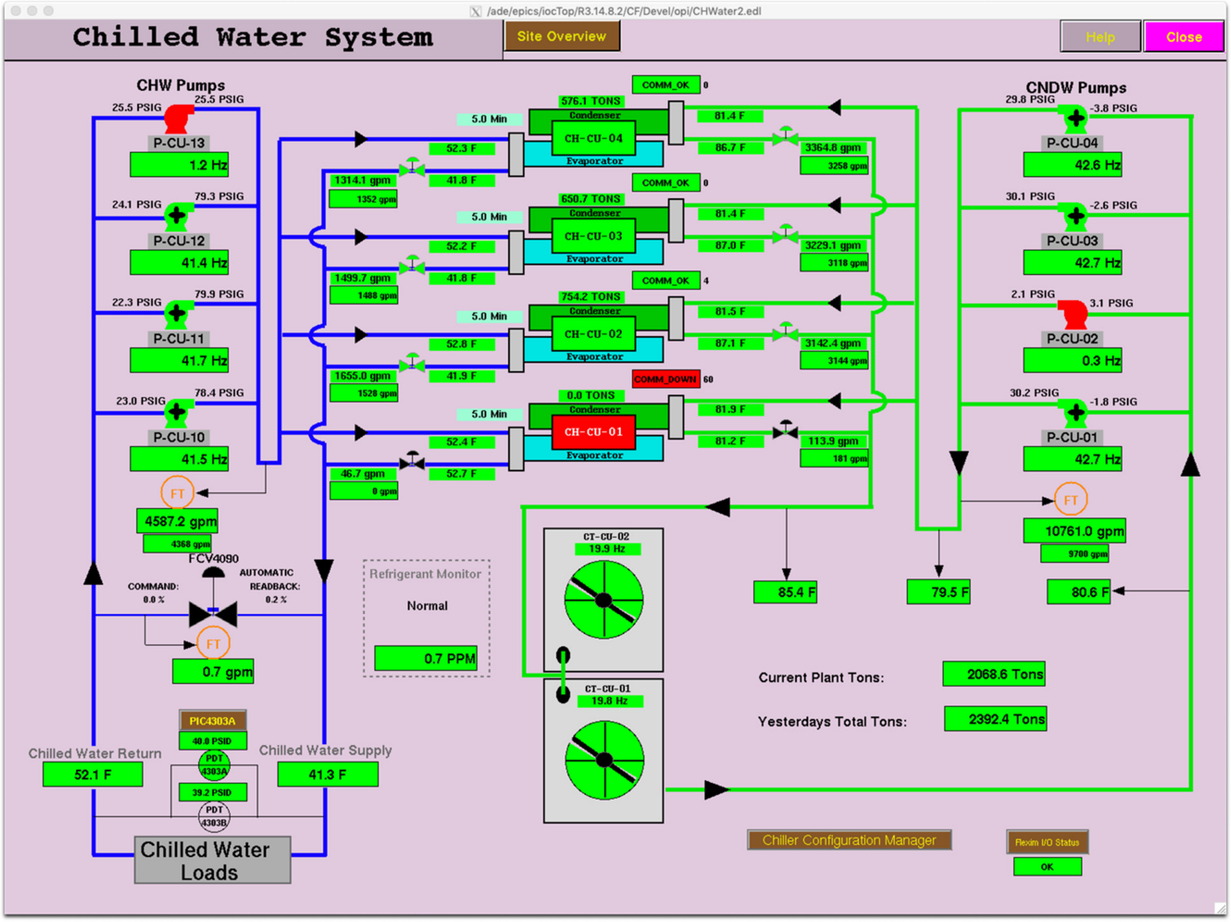

Figure 3: SNS chilled water system overview screen with BACnet data and PLC data.

REFERENCES

[1] SNS, `https://neutrons.ornl.gov/sns`

[2] BACnet, `http://www.bacnet.org`

[3] L.R. Dalesio, M.R. Kraimer, A.J. Kozubal, "EPICS Architecture," in *Proc. International Conference on Accelerator and Large Experimental Physics Control Systems*, Tsukuba, Japan, 1991, pp. 278-282.

[4] BACnet Driver, `https://controlssoftware.sns.ornl.gov/bacnet/index.html`