WHITE PAPER

Rohit Goel
digital applications associate, processors

Punya Prakash
business manager, processors

Texas Instruments

Data communication protocol for control networks enabling automated buildings

Introduction to BACnet

A building automation system encompasses a highly complex network of control systems. It is critical that a self-governing intelligent protocol be established to determine and institute a robust communication system. The concept of automated buildings has been in discussion for several years; however, there is no true definition or well-accepted industry standard for this. BACnet has been under active development since 1987, but it was only in 1995 that a standard for ***BACnet*** *(building and automation control network) was officially instated.*

BACnet is a data communications protocol for building automation and control networks. It aids in intelligent building systems and helps achieve their control and data analysis goals. This ***ASHRAE****,* ***ANSI****, and* ***ISO*** *standard protocol is designed to facilitate communication of building automation and control systems for applications such as heating, ventilating, and air-conditioning control, lighting control, access control, and fire detection systems and their associated equipment. This protocol standard provides the infrastructure for the various systems to exchange information and is agnostic to the building service that it renders. To maximize the efficiency of these connected building systems, competent communication between the various building blocks is essential.*

This white paper discusses how the BACnet protocol provides the infrastructure for automated buildings. Details of the functionality of the software protocol library and use cases on the Texas Instruments (TI) processor are also outlined in this paper.

Topology of a standard building automation system

Building automation systems are deployed across all levels ranging from small building segments to larger building establishments. Building automation system (BAS) is a communication network infrastructure that manages various building services. The key to an effective BAS is having a ubiquitous system that can be deployed to serve new and old building technology as well as small and large commercial facilities. Typically, the complex building automation network of devices includes a primary and secondary bus that is connected to various nodes in the system:

- Building management service
- Building control systems
- Zone controllers
- End nodes

For more details on a standard building automation system please refer to the **BAS white paper**.

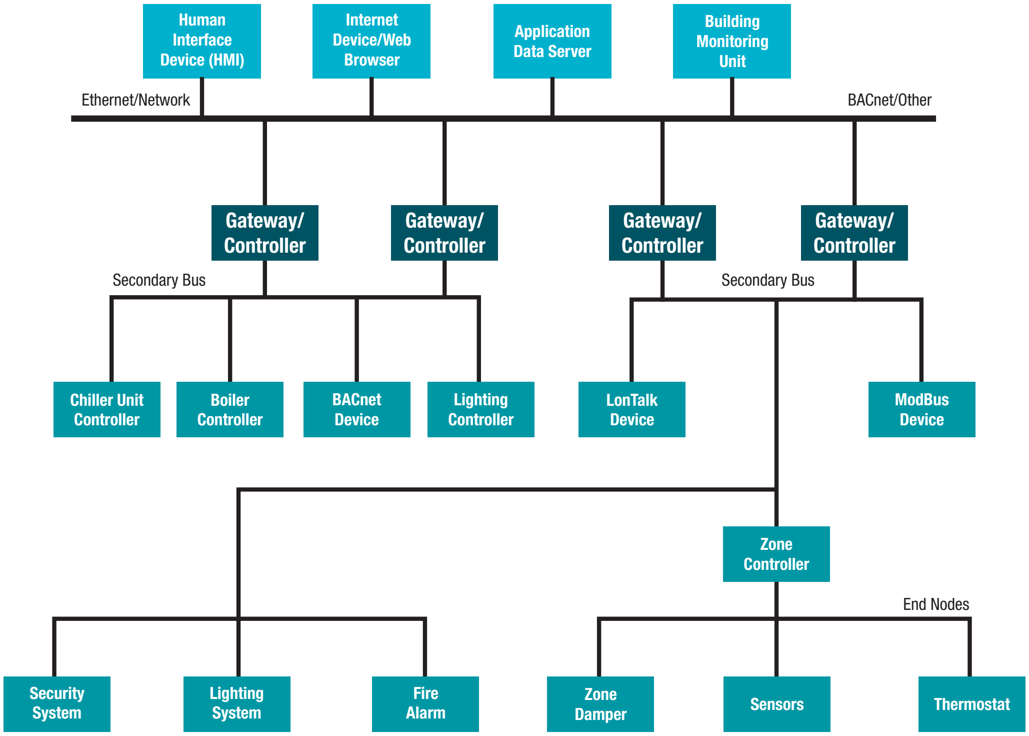


Figure 1. Typical building automation system topology

Other prominent communication protocols used in building automation systems are:

- The LonMark standard is based on the proprietary communications protocol LonTalk. The LonTalk protocol establishes a set of rules to manage communications between devices, while LonWorks defines the content and structure of the information that is exchanged between them. Like BACnet, LonWorks has been accepted and adopted by the international standards organizations (ANSI/CEA 709.1 and IEEE 1473-L).
- Modbus® is a truly open standard and is one of the most widely used protocols in the industrial manufacturing environment. Its messaging structure establishes master-slave, client-server communications between devices. A relatively smaller percentage of installations are Modbus certified.

Why BACnet?

Systems based on Modbus tend to be smaller with highly centralized controls. The other problem with Modbus is that its concept for data access is “memory file” oriented which places the burden of understanding and structuring information on the controller that is asking for data.

Each of the protocols has advantages and disadvantages and must be selected based on the needs of the facility and its ability to support a particular protocol. However, BACnet is the first choice where the system is not required to have limited size, scope and interoperability requirements.

BACnet technology

The BACnet protocol institutes a standard way to send and receive messages using wired or wireless standard protocols. BACnet data is exchanged between only devices that have BACnet-enabled applications. The BACnet standard defines a standard way to communicate via data link/physical layers: Ethernet, RS-485, RS-232 and other protocol standards such as ARCNET and LonTalk. In addition, the BACnet standard also defines a standard way to communicate using UDP, IP and HTTP (web services).

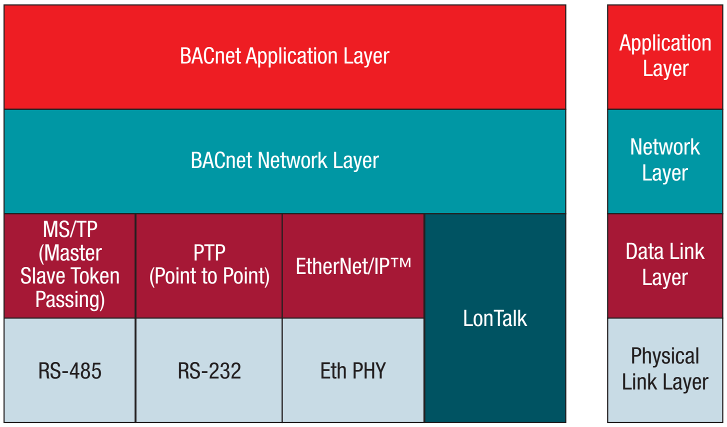


Figure 2. Condensed view of the BACnet layer stack

Compared to the seven-layer OSI model, BACnet implements a four-layer collapsed architecture as shown in figure 2. It consists of following layers:

- Application layer that provides communication services
- Network layer handles network-to-network communication
- Data link layer handles device-to-device communication within a network
- Physical layer converts electrical signals sent over the medium into data

Being an open protocol, BACnet is becoming increasingly adopted across various building device manufacturers. This is a critical feature since it provides a common foundation to enable building systems and devices from different manufacturers to interoperate. This directly drives a lower system cost as there are now minimal investments in delivering custom drivers and protocols. The level of intelligence required in a building is determined by the occupant and is not necessarily standard across the establishment; this protocol is effectively a glue logic which connects the various devices together. Since its inception, it has been designed to provide a standard for building automation.

Key benefits of the standard BACnet protocol

- Single primary management service for all the networks of systems
- Can be implemented in devices of any size ranging from microcontroller to embedded processors
- Infrastructure for easy system expansion providing flexibility to add more devices from multiple vendors

- Potential to integrate all – old and new – building automation and control functions
- Interoperability

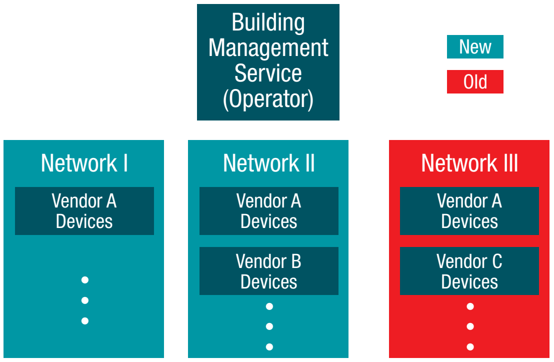


Figure 3. Demonstration of flexibility and interoperability of a standard BACnet protocol network

Figure 3 demonstrates combinations of devices from various vendors, each of these devices are deployed in different networks. New networks I and II and an older network III are all operated by one primary operator exhibiting the primary benefit of the BACnet protocol solution.

When using BACnet, the communication of each device is replaced by a standard set of communication rules signifying a common language. This ensures that each device looks the same on the network. All the information required to describe a device in the network is represented in the form of an object. This information could be physical inputs, outputs or even software processes – thus establishing a common network view. The services manage the requests and interoperability of the devices while a selection of diverse network layers enable the transport system.

BACnet network profiles and device configuration

Network profile

Depending on the scale of the BAS, the network of BACnet devices could be managed using a simple network, or in a complex grid structure.

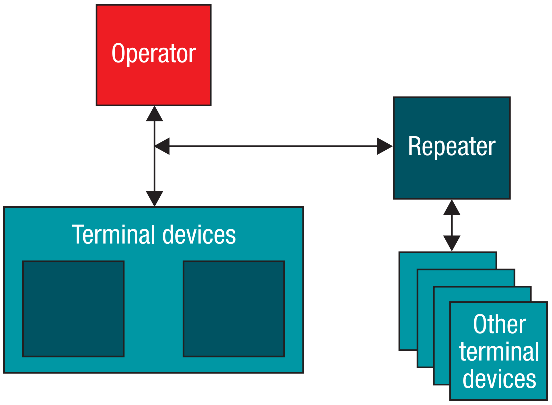


Figure 4. A simple BACnet network of automated devices

- **Simple network**

  A single network demonstrates a simple grid of terminal devices using one distinct communication medium, such as RS-485. These terminal devices can be further expanded in the same network using repeaters.

- **Compound network**

  A BACnet router is used to connect multiple "single" networks to establish a complex network grid. Here the router is used to bridge data packets from one set of devices to either the operator or the other network devices.

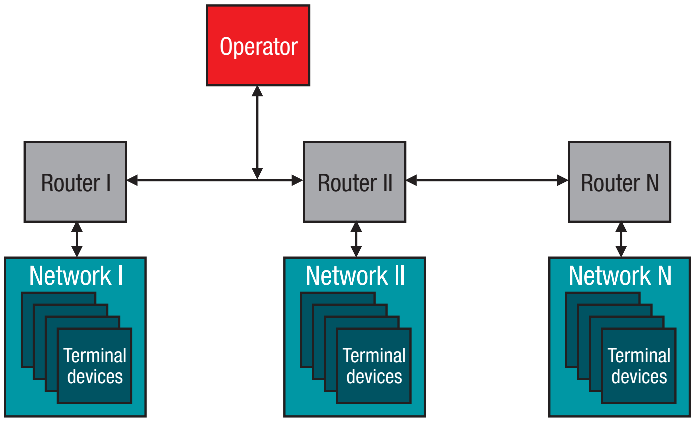
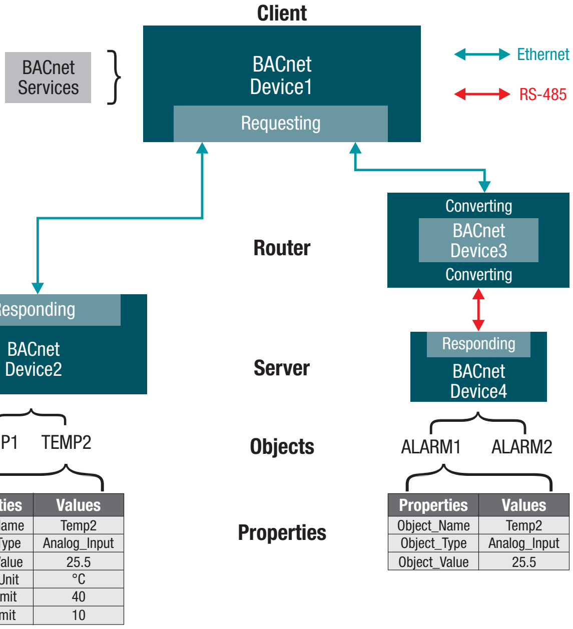


Figure 5. A compound grid of automated devices connecting multiple networks

Device configuration

Each device in the BACnet network can be configured based on its core application. Depending on the application requirement, the device can be configured as a client, server or a router.



Properties	Values
Object_Name	Temp2
Object_Type	Analog_Input
Object_Value	25.5
Object_Unit	°C
High_Limit	40
Low_Limit	10

Properties	Values
Object_Name	Temp2
Object_Type	Analog_Input
Object_Value	25.5

Figure 6. An example of standard BACnet device configurations

- **Client**

  A device configured as a client is the master controller; it initiates a service request to the multiple nodes, including the server or router. This service request could be to retrieve or modify server data. The client is also responsible to configure the server to set the threshold for external events.
- **Server**

  This slave device services the requests initiated by the master client. Depending on the type of service the process is equipped to optionally respond to the client. In case a pre-configured external event is triggered, the server is also equipped to send alarm events to the client.
- **Router**

  In a complex network, a router is used to connect multiple BACnet devices. This network of devices could be a set of client-server or server-server systems. A router is also used to cross network mediums (e.g., RS-485 to Ethernet).

BACnet modules

The data structures used in a device to store information are confined to the host device. In order to exchange information with another device using this protocol, there must be a "network-visible" representation of the information that is standardized. BACnet uses an object-oriented structure of objects, properties and services which allows devices to be modelled in a way that they are network visible.

Device modelling

With device modelling, a physical BACnet device appears as a collection of objects with properties that represent data and functionality. Objects represent physical or virtual information, control algorithms, special application and calculations. Some of the commonly used BACnet objects are analog input (AI), analog output (AO), binary input (BI) and binary output (BO).

There are currently 50 predefined BACnet objects and the numbers continue to grow as further requirements are recognized.

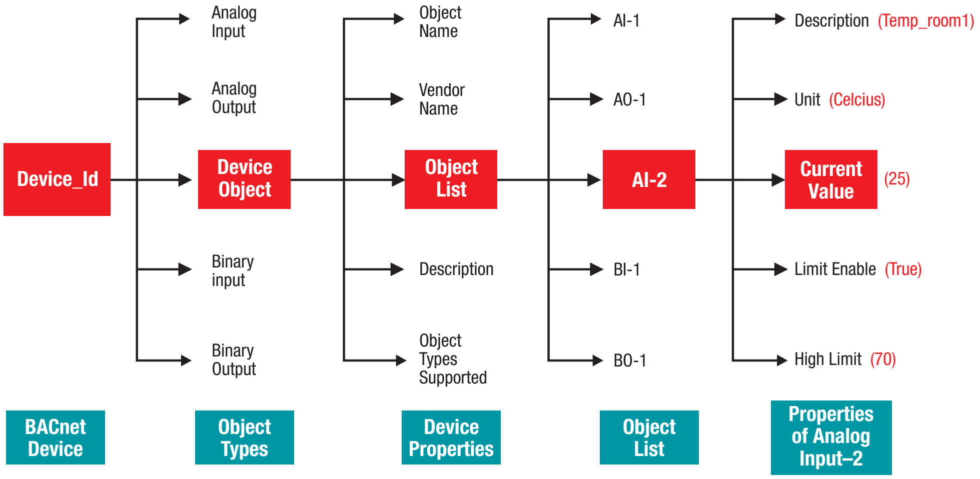


Figure 7. Device modelling representation of a BACnet device

Each object type has a defined set of properties. The values of individual properties characterize the object. The number and type of properties vary with the object types.

For each device:

- Instance number of a device must be unique within a complete BACnet Internetwork.
- Each device must have at least one object – Device object.

Services

Services are the commands issued by the BACnet devices for reading and manipulating information at the application layer. These services are requested in the form of PDU (Protocol Data Unit) messages. The structure of these PDUs depend on the type of services, the standard services are:

- Confirmed request
- Unconfirmed request
- Acknowledgement
- Error
- Reject

Confirmed services are unicast messages and are expected to be acknowledged while unconfirmed services are usually broadcast messages and don't need any response back from the receiver.

The BACnet device, to which a service is requested, has a handler defined for each type of service. Upon receiving a service request the BACnet device will pass the request to an appropriate handler which will process it further. Currently there are 37 defined services which are grouped under five main categories:

- Alarm and event services
- File access services
- Object access services
- Remote device management services
- Virtual terminal services

A BACnet device is not required to support all services except one service. The mandatory "read_property" service must be supported by every BACnet device.

Application

The object-oriented structure of BACnet allows the user to create any BAS application with minimal effort by utilizing the wide variety of readily available services and objects. Based on the scale and scope of BAS users can also configure the number of BACnet device instances, objects and properties in their application. In case the system needs to be expanded, new device instances or their objects can be simply added at an application level.

An Application running at the top level of the building management system controls all the other devices. It has to take inputs from all the lower devices and will have to process multiple inputs from multiple devices. Applications are complex and must be written with the overall system in mind. It is non-trivial to develop applications that do not cause contradictions in operations and processing delays. These types of applications are time consuming to develop.

BACnet provides various features specific to building automation which can reduce the development time of an application significantly. Based on user settings, BACnet coordinates the routing of data effectively from remote devices to the central unit. Users can decide how and when to retrieve information from other devices by scheduling and prioritizing different events in the system.

Some of these features are:

- **Device and network management**

  Device and network management consists of the establishment and exchange of operational characteristics. It allows BACnet devices to discover other BACnet devices, discover objects within devices, establish and re-establish communications, synchronize time and re-initialize a device's program.
- **Alarm and event management**

  Alarm and event management defines the exchange of data based on pre-defined alarm limits or event triggers. The event or alarm may require human intervention and acknowledgement. Alarms and events may also be logged and summaries generated.
- **Trending**

  Trending allows BACnet devices to enable trend collections and request trend data from other BACnet devices.
- **Scheduling**

  Scheduling allows BACnet devices to establish, and edit schedules in BACnet devices so that control can be coordinated based on dates and times.

BACnet software architecture

The **BACnet protocol stack library** provides a BACnet application layer, network layer and media access (MAC) layer communications services. It is an open source, royalty-free library for embedded systems supporting Windows®, Linux™ and additional operating systems. Source code of the stack is written in C and is designed to be portable across many compilers and architectures. The stack includes BACnet examples for the server, router and client which can be used in scripts.

The BACnet operating stack is enabled in each device in the network at the board level and in all operator interface software packages. Device drivers for all the hardware interfaces to be used in the system must be enabled in the operating system kernel. Customized BACnet objects for a specific device can be bound to these drivers at the application layer. Initial settings for different communication medium such as RS-485, Ethernet and others are done at the data link layer of the stack. The BACnet application layer is independent

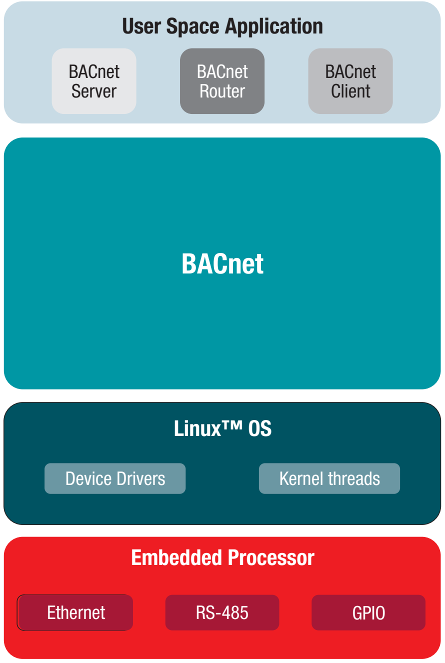


Figure 8. High-level software architecture

of transport layer such as MSTP, IP, Ethernet and even PTP and ARCnet. This enables the user to enhance the bandwidth and performance by exploiting the resources such as multi-threading and multiple simultaneous connections.

System automation

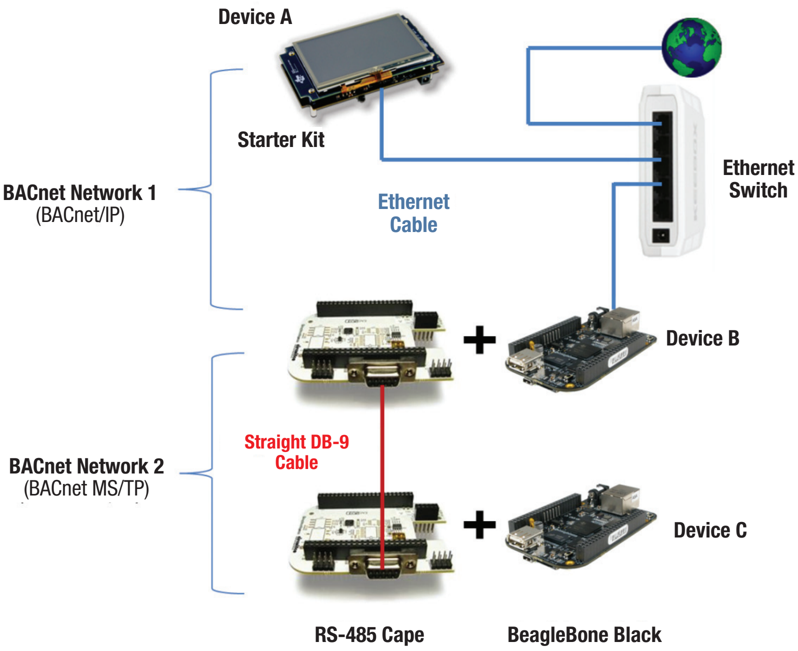


Figure 9. TI processor set-up for BACnet integration and validation

Set-up

The Sitara™ AM335x processors are used for test and validation of the BACnet functionality. Here Sitara hardware platforms are used – one each for client, server and router application. The set-up is connected as shown in Figure 9 on the previous page.

- Starter kit with Ethernet port addressed as Device A
- Two BeagleBone Black boards with one Ethernet port and one PRU-based RS-485 cape on each board – addressed as Device B and Device C respectively. For more details on this cape refer to: [link for RS-485 cape]

Integration

Device A based on the AM335x Starter Kit is configured as BACnet client that sends the service request in the form of Protocol Data Unit (PDU) packets through Ethernet medium. The client application configures the Ethernet port and sets the network number as zero. It also initializes all the handlers that can process the responses back from the server.

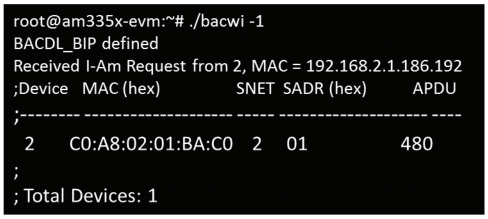
Device B based on BeagleBone with the RS-485 cape is running the BACnet router application which allows messages to cross network mediums. The application configures the network address of Ethernet as port one and RS-485 as port two and initializes a routing table for both the networks. It extracts the application data from an incoming packet and uses the rest to get network information such as source address and destination address. Then, using the routing table, a destination port is determined and the extracted information is used to form a new packet which is sent over that port. Multiple threads are running on the processor to handle packets from each port.

Device C based on BeagleBone with RS-485 is running the BACnet server application which sets the network number as zero and device instance as two for this device. It configures all the objects and their properties to be supported by this device. For testing, four analog inputs are configured and linked to TI ADC's input. Finally, the device initializes all the handlers for the service requests from the client.

The IP addresses for Device A and Device B are assigned in the same sub-net.MAC address of RS-485 port of Device B is set by default as zero as it is a router. Since, there are only two devices in the MS/TP network, the maximum number of masters in the MS/TP network is set as "2". MAC address of Device C is sequentially numbered as one in order to optimize the bandwidth. Both the devices are acting as master because the router has to be a master node and slave cannot respond to any broadcast message.

Validation

A client application needs to establish communication with the devices it seeks to communicate with. To do this, a who-is command is broadcast by Device A using the command `bacwi –1` on a serial terminal. Device A gets I-am response and binds the address of Device C to its address cache. Network address is printed on the client terminal like this:

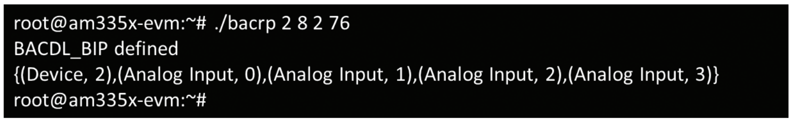


```
root@am335x-evm:~# ./bacwi -1
BACDL_BIP defined
Received I-Am Request from 2, MAC = 192.168.2.1.186.192
;Device   MAC (hex)            SNET  SADR (hex)          APDU
;-------- -------------------- ----- -------------------- ----
  2       C0:A8:02:01:BA:C0    2     01                   480
;
; Total Devices: 1
```

Figure 10. Execution of BACnet whois service request

Once the location of device C is found it is queried for other services.

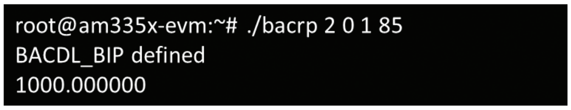
Device A reads the list of objects supported by Device C.



```
root@am335x-evm:~# ./bacrp 2 8 2 76
BACDL_BIP defined
{(Device, 2),(Analog Input, 0),(Analog Input, 1),(Analog Input, 2),(Analog Input, 3)}
root@am335x-evm:~#
```

The above figure shows that the device C supports one device object and four Analog Input objects.

Device A reads the current value AI-1 of Device C which is linked to TI_ADC5.



```
root@am335x-evm:~# ./bacrp 2 0 1 85
BACDL_BIP defined
1000.000000
```

Conclusion

Continuously evolving smart building technology challenges the building owners and vendors to develop and provide solutions according to the latest technology in the industry. The BACnet protocol provides a strong foundation to enable the different entities of an automated building solution. This convergence facilitates the control technology to provision the management of energy-efficient systems optimizing the operating conditions of the building devices.

These automated solutions are a valued business proposition for the building management service provider and occupants alike. With easy integration and validated system solutions, TI processors demonstrate a highly flexible solution to realize smart building technology.

Important Notice: The products and services of Texas Instruments Incorporated and its subsidiaries described herein are sold subject to TI's standard terms and conditions of sale. Customers are advised to obtain the most current and complete information about TI products and services before placing orders. TI assumes no liability for applications assistance, customer's applications or product designs, software performance, or infringement of patents. The publication of information regarding any other company's products or services does not constitute TI's approval, warranty or endorsement thereof.

Sitara is a trademark of Texas Instruments. All other trademarks are the property of their respective owners.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265