PG-131-103-BX Gamewell FCI to BACnet IP Protocol Converter

PG-131-103-BX is highly powerful, superior, completely configurable and productive Building & Industrial Automation gateway for integrators to effortlessly interface devices to networks in commercial buildings and industrial plants.

PG-131-103-BX Gateway model supports Gamewell FCI and BACnet IP protocols. It is a Bi-directional Converter that can be configured as a Client on Gamewell FCI side and a Server/Client on BACnet IP protocol interface.

When configured as a Gamewell FCI client, the PG-131-103-BX can read data from your Gamewell FCI fire panel and publish it as BACnet IP data. Also, it can write commands sent from the BACnet IP side to the Gamewell FCI fire panel.

When configured as a BACnet IP client, the PG-131-103-BX can read data from your BACnet IP devices and publish it as Gamewell FCI type data. Also, it can write commands sent from the Gamewell FCI side to the BACnet IP devices.

The PG-131-103-BX can be configured to behave as a server on BACnet IP interfaces. This mode is useful when data exchange is required between a Gamewell FCI client (for eg. SCADA) and a BACnet IP client (for eg. a Building Management System).

The PG-131-103-BX can be configured to behave as a client on both Gamewell FCI and BACnet IP interfaces.

PG-131-103-BX gateways have benefitted system integrators worldwide with its powerful line of gateways. Additionally, PG-131-103-BX gateway runs the same protocol conversion software on a productive and cost efficient platform backed by the experience, engineering expertise and technically proven support that integrators have come to expect from PG-131-103-BX.

Features

- Ability to interface upto 1000 points
- DIN rail mount optional
- DIP switches to select baud rate or node ID on the fly
- Multi-configuration capability
- BACnet COV support for fast data communication while reducing the traffic over a BACnet network

www.protoconvert.com
sales@protoconvert.com

Specifications

Environment	Operating Temperature: -40 to 75° C (-40 to 167°F)	
	Relative Humidity:5-90% RH non-condensing	
Power Requirements	9-30 VDC or 12-24 VAC	
	Current Draw @ 12V about 250Ma	
Physical Dimensions(HxWxD)	4.5x2.9x1.6 in. (11.5x7.4x4.1 cm)	
	0.4 lbs (0.2 Kg)	
Other	Configuration/Diagnostic utilities	
	Capacity: 1000 points	
	Table,Wall or DIN rail mount	
Communication Interfaces	RS-485	1
	RS-485 or RS-232	1
	Ethernet 10Base-T, 100BASE-T[2]	1
	Mbus	-
	KNX	-
	LonWorks	-
Approvals	TUV Approved to UL 916 and CSA C22.2 standards	
	BTL and LonMark certified	
	LonMark Certified	
	RoHS Compliant	
	GOST-R Certified	
	CE and FCC	

BACnet IP Protocol Driver Description

Driver Name: BACnet/IP	Connection type:	Internet Protocol (IP)
	Ethernet Speeds Supported:	10Base-T, 100BASE-T[2]
	BBMD SUPPORTED:	Yes(Not supported on client connections)
	Foreign Device:	Not Supported for client
	Registration:	Connections
PG-131-103-BX AS A BACnet IP CLIENT		
Read Operations Supported	**Properties Supported**	**Comments and Limitations**
Read Property	Present Value	Store value in Data Array location after scaling has been applied
	Out_Of_Service	When using a Complex Data Object, the OOS property is fully supported. Return FALSE when not OOS or when using standard Data Arrays
	Units	Returns Units as specified in the Map Descriptor
	Reliability	When using a Complex Data Objects, returns “Unreliable Other” when the Node is offline, or when the data is old. Returns FALSE if the Node is online or when using Standard Data Arrays
	Priority_Array	Returns Priority_Array of Map Descriptor
	Unsupported	This property is supported
	Protocol_Object_Type_Supported	This property is supported
	Protocol_Services_Supported	This property is supported

	Database_Revision	This property is supported and will change if a new configuration is downloaded to the FS
	Max_Master	This property is supported for the BACnet /MSTP DLL option
	Max_Info_Frames	This property is supported for the BACnet/MSTP DLL option
	Relinguish_Default	Returns Relinguish _Default
Read Property Multiple	As for Read Property	Transactions can be defined to read multiple objects and properties in a single ReadPropertyMultiple operation.
	ALL	Read Property Multiple of the ALL property is NOT supported
Write Operations Supported	**Properties Supported**	**Comments and Limitations**
Write Property	Present Value	Send value in Data Array location after scaling has been applied
Write Property Multiple		
PG-131-103-BX AS A BACnet IP SERVER		
DEVICE OBJECT		
Read Operations Supported	**Properties Supported**	**Comments and Limitations**
Read Property	Object_Identifier	Returns Object _ID with Node_ID as Object Instance
	Object_Name	Returns Node Name
	Object_Type	Returns Device Object type
	System_Status	Returns Normal
	Vendor_Name	Returns PG-10XX Technologies

	Vendor_Identifier	Returns 37
	Model_Name	Returns PG-10XX model
	Firmware_revision	Returns Kernel Version
	Application_sw_version	Returns DCC version
	Protocol_Version	Returns version 1
	Protocol_Revision	Returns version 1
	Protocol_Services_Supported	This property is supported
	Protocol_Object_Type_Supported	This property is supported
	Protocol_Object_List	Returns a list of objects defined in the PG-10XX
	Max_APDU_Length_Accepted	For PG-10XX,the MAX APDU length for BACnet MSTP is 480 bytes and for BACnet IP/BACnet Eth 1497 bytes
	Segmentation_Supported	Returns Segmantation NOT Supported
	APDU_Timeout	Returns the value as defined by the Node's "Timeout" parameter
	APDU_Retries	Returns the value as defined by the Node's "Retries" parameter
	Device_Address_Bindings	Returns an empty list
	Max_Master	This property is supported for the BACnet/MSTP DLL option
	Max_info_Frames	This property is supported for the BACnet/MSTP DLL option
	Description	This property is supported
	Database_Revision	This property is supported and will change if a new configuration is downloaded to the PG-10XX
Read Property Multiple	Same properties as Read Property	Read Property Multiple is fully supported. Multiple objects with multiple properties can

		be specified
Write Operations Supported	**Properties Supported**	**Comments and Limitation**
Write Property	Max_Master	This Property is supported for the BACnet /MSTP DLL option
	Max_info_Frames	This Property is supported for the BACnet /MSTP DLL option
Write Property Multiple	Max_Master	This Property is supported for the BACnet /MSTP DLL option
	Max_info_Frames	This Property is supported for the BACnet /MSTP DLL option
Analog Input Object		
Read Operations Supported	**Properties Supported**	**Comments and Limitations**
Read Property	Object_Identifier	No Limitations
	Object_Name	Returns Map Descriptor Name
	Object_Type	Returns Analog Input Object Type
	Present_Value	Returns value in Data_Array after scaling has been applied
	Status_Flags	When using Complex Data Objects returns the FAULT and OUT_OF_SERVICE fields as indicated in section 12.2.7 of the BACnet specification. When using standard Data Arrays returns FALSE for all bits.
	Event_State	No Limitations
	Reliability	When using a Complex Data Objects, returns Unreliable Other when the Node is offline, or when the data is old. Returns FALSE if the node is

		online or when using Standard Data Arrays
	Out_Of_Service	Fully supported when using a Complex data Object. Returns FALSE when not OOS or when using standard Data Arrays
	Description	This property is supported
	Units	Returns Units as specified in the Map Descriptor
Read Property Multiple	Same properties as Read Property	Read Property Multiple is fully supported. Multiple objects with Multiple properties can be specified
Write Operations Supported	**Properties Supported**	**Comments and Limitations**
Write Property	Present_Value	Writing to the Present Value is allowed if the Object is OOS
Write Property Multiple		
Data Sharing Operations Supported	**Properties Supported**	**Comments and Limitations**
SubscribeCOV	Present_value	Subscription storage is non-volatile
COVNotification	Present_value	Confirmed and Unconfirmed
Alarm and Event Operations Supported	**Properties Supported**	**Comments and Limitations**
EventNotification	Present_Value,Status	Confirmed and Unconfirmed
AcknowledgeAlarm		No limitations
Analog Output Object, Analog Value Object		

Read Operations Supported	Properties Supported	Comments and Limitations
Read Property	Object_Identifier	No Limitations
	Object_Name	Returns “Map Descriptor Name”
	Object_Type	Returns Analog Output Object type
	Present_Value	Returns value in Data Array after scaling has been applied
	Status_Flags	When using Complex Data Objects returns the FAULT and OUT_OF_SERVICE fields as indicated in section 12.2.7 of the BACnet specification. When using standard Data Arrays returns FALSE for all bits
	Event_State	No Limitations
	Reliability	When using a Complex Data Objects, returns “Unreliable Other” when the Node is offline, or when the data is old. Returns FALSE if the Node is online or when using Standard Data Arrays
	Out_Of_Service	Fully supported when using a Complex Data Object. Returns FALSE when not OOS or when using standard Data Arrays
	Units	Returns Units as specified in the Map Descriptor
	Priority_Array	Returns Priority_Array of Map Descriptor
	Description	This property is supported
	Relinguish_Default	Returns Religuish _Default
Read Property Multiple	Same properties as Read Property	Read Property Multiple is fully supported. Multiple objects

		with multiple properties can be specified
Write Operations Supported	Properties Supported	Comments and Limitations
Write Property Write Property Multiple	Present_Value	When using Complex Data Objects and OOS is TRUE, then the write will not cause a write-through operation to the Server side. If the OOS is FALSE or when using standard Data Arrays then writes will always cause a write-through operation to the Server side
Data Sharing Operations Supported	**Properties Supported**	**Comments and Limitations**
SubscribeCOV	Present_Value	Subscription storage is non-volatile
COVNotification	Present_Value	Confirmed and Unconfirmed
Alarm and Event Operations Supported	**Properties Supported**	**Comments and Limitations**
Event Notification	Present_Value, Status	Confirmed and Unconfirmed
AcknowledgeAlarm		No Limitations
Binary Input Object		
Read Operations Supported	**Properties Supported**	**Comments and Limitations**
Read Property	Object_Identifier	No Limitations
	Object_Name	Returns “Map Descriptor Name”
	Object_Type	Returns Analog Input Object type
	Present_Value	Returns the binary value in the

		data array
	Status_Flags	When using Complex Data Objects returns the FAULT and OUT_OF_SERVICE fields as indicated in section 12.2.7 of the BACnet specification. When using standard Data Arrays returns FALSE for all bits
	Event_State	No Limitations
	Reliability	When using a Complex Data Objects, returns “Unreliable Other” when the Node is offline, or when the data is old. Returns FALSE if the Node is online or when using Standard Data Arrays
	Out_Of_Service	Fully supported when using Complex Data Object. Returns FALSE when not OOS or when using standard Data Arrays
	Polarity	Always returns “Normal”
	Active_Text	Returns Active Text as specified on the Map Descriptor
	Description	This property is supported
	Inactive_Text	Returns Inactive Text as specified on the Map Descriptor
Read Property Multiple	Same properties as Read Property	Read property Multiple is fully supported. Multiple objects with multiple properties can be specified
Write Operations Supported	**Properties Supported**	**Comments and Limitations**
Write Property	Present_Value	Writing to the Present Value is

Write Property Multiple		allowed if the Object is OOS
Data Sharing Operations Supported	**Properties Supported**	**Comments and Limitations**
SubscribeCOV	Present_Value	Subscription storage is non-volatile
COVNotification	Present_Value	Confirmed and Unconfirmed
Alarm and Event Operations Supported	**Properties Supported**	**Comments and Limitations**
Event Notification	Present_Value, Status	Confirmed and Unconfirmed
AcknowledgeAlarm		No Limitations
Binary Output Object, Binary Value Object		
Read Operations Supported	**Properties Supported**	**Comments and Limitations**
Read Property	Object_Identifier	No Limitations
	Object_Name	Returns “Map Descriptor Name”
	Object_Type	Returns Analog Input Object type
	Present_Value	Returns the binary value in the data array
	Status_Flags	When using Complex Data Objects returns the FAULT and OUT_OF_SERVICE fields as indicated in section 12.2.7 of the BACnet specification. When using standard Data Arrays returns FALSE for all bits
	Event_State	No Limitations

	Reliability	When using a Complex Data Objects, returns “Unreliable Other” when the Node is offline, or when the data is old. Returns FALSE if the Node is online or when using Standard Data Arrays
	Out_Of_Service	Fully supported when using Complex Data Object. Returns FALSE when not OOS or when using standard Data Arrays
	Priority_Array	Returns Priority_Array of Map Descriptor
	Religuish_Default	Returns Current Relinguish_Default
	Description	This property is supported
	Active_Text	Returns Active Text as specified on the Map Descriptor
	Inactive_Text	Returns Inactive Text as specified on the Map Descriptor
Read Property Multiple	Same properties as Read Property	Read property Multiple is fully supported. Multiple objects with multiple properties can be specified
Write Operations Supported	**Properties Supported**	**Comments and Limitations**
Write Property Write Property Multiple	Present_Value	When using Complex Data Objects and OOS is TRUE, then the write will not cause a write-through operation to the downstream side. If the OOS is FALSE or when using standard Data Arrays when writes will always cause a write-through operation to the downstream

		side
Data Sharing Operations Supported	**Properties Supported**	**Comments and Limitations**
SubscribeCOV	Present_Value	Subscription storage is non-volatile
COVNotification	Present_Value	Confirmed and Unconfirmed
Alarm and Event Operations Supported	**Properties Supported**	**Comments and Limitations**
Event Notification	Present_Value, Status	Confirmed and Unconfirmed
AcknowledgeAlarm		No Limitations
Multiple State Input Object		
Read Operations Supported	**Properties Supported**	**Comments and Limitations**
Read Property	Object_Identifier	No Limitations
	Object_Name	Returns “Map Descriptor Name”
	Object_Type	Returns Analog Input Object type
	Present_Value	Returns unsigned Integer value in the data array
	Status_Flags	When using Complex Data Objects returns the FAULT and OUT_OF_SERVICE fields as indicated in section 12.2.7 of the BACnet specification. When using standard Data Arrays returns FALSE for all bits
	Event_State	No Limitations
	Reliability	When using a Complex Data Objects, returns “Unreliable Other” when the Node is

		offline, or when the data is old. Returns FALSE if the Node is online or when using Standard Data Arrays
	Description	This property is supported
	Out_Of_Service	When using a Complex Data Object, the OOS property is fully supported. Returns FALSE when not OOS or when using standard Data Arrays
	Number_Of_State	When using a Complex Data Object, returns the number of states defined. When using Standard Data Arrays returns the value of 5
	State_Text	When using Complex Data Objects returns the State Text Strings defined. When using Standard Data Arrays return "State_X" where "X" is the value stored in Data_Array and could be 0 to 4
Read Property Multiple	Same properties as Read Property	Read property Multiple is fully supported. Multiple objects with multiple properties can be specified
Write Operations Supported	**Properties Supported**	**Comments and Limitations**
Write Property	Present_Value	Writing to the Present Value is allowed if the Object is OOS
Write Property Multiple		
Data Sharing Operations Supported	**Properties Supported**	**Comments and Limitations**
SubscribeCOV	Present_Value	Subscription storage is non-volatile

COVNotification	Present_Value	Confirmed and Unconfirmed
Alarm and Event Operations Supported	**Properties Supported**	**Comments and Limitations**
Event Notification	Present_Value, Status	Confirmed and Unconfirmed
AcknowledgeAlarm		No Limitations
Multi-State Output Object, Multi-State Value Object		
Read Operations Supported	**Properties Supported**	**Comments and Limitations**
Read Property	Object_Identifier	No Limitations
	Object_Name	Returns “Map Descriptor Name”
	Object_Type	Returns Analog Input Object type
	Present_Value	Returns unsigned Integer value in the data array
	Status_Flags	When using Complex Data Objects returns the FAULT and OUT_OF_SERVICE fields as indicated in section 12.2.7 of the BACnet specification. When using standard Data Arrays returns FALSE for all bits
	Event_State	No Limitations
	Reliability	When using a Complex Data Objects, returns “Unreliable Other” when the Node is offline, or when the data is old. Returns FALSE if the Node is online or when using Standard Data Arrays
	Out_Of_Service	Fully supported when using a Complex Data Object. Returns FALSE when not OOS or when using standard Data Arrays
	Number_Of_State	When using a Complex Data

		Object, returns the number of states defined. When using Standard Data Arrays returns the value of 5
	State_Text	When using Complex Data Objects returns the State Text Strings defined. When using Standard Data Arrays return "State_X" where "X" is the value stored in Data_Array and could be 0 to 4
	Description	This property is supported
	Priority_Array	Returns Priority_Array of Map Descriptor
	Religuish_Default	Returns Relinguish_Default
Read Property Multiple	Same properties as Read Property	Read property Multiple is fully supported. Multiple objects with multiple properties can be specified
Write Operations Supported	**Properties Supported**	**Comments and Limitations**
Write Property Write Property Multiple	Present_Value	When using Complex Data Objects and OOS is FALSE or when using standard data arrays, writes will trigger a write through operation to client side
Data Sharing Operations Supported	**Properties Supported**	**Comments and Limitations**
SubscribeCOV	Present_Value	Subscription storage is non-volatile
COVNotification	Present_Value	Confirmed and Unconfirmed
Alarm and Event	**Properties Supported**	**Comments and Limitations**

Operations Supported		
Event Notification	Present_Value, Status	Confirmed and Unconfirmed
AcknowledgeAlarm		No Limitations
Notification Class Object		
Read Operations Supported	**Properties Supported**	**Comments and Limitations**
Read Property	Object_Identifier	No Limitations
	Object_Name	Returns “Map Descriptor Name”
	Object_Type	Returns Notification Class Object type
	Description	No Limitations
	Notification_Class	No Limitations
	Priority	No Limitations
	Ack_Required	No Limitations
	Description	This Property is supported
	Recipient List	No Limitations
Read Property Multiple	Same properties as Read Property	Read property Multiple is fully supported. Multiple objects with multiple properties can be specified
Write Operations Supported	**Properties Supported**	**Comments and Limitations**
Write Property	Recipient_List	RecipientList storage is non-volatile
Write Property Multiple		
AddList	RecipientList	Used to subscribe to Alarm and Event Notifications

Unsupported Functions And Data Types
BACnet Object Type not Supported
Averaging Object
Calendar Object
Command Object
Event Enrollment Object
File Object
Group Object
Life Safety Point Object
Life Safety Zone Object
Loop Object
Notification Class Object unsupported on Client side only
Program Object
Schedule Object
BACnet Services not Supported
Alarm and Event Services unsupported on Client side only
File Access Services
Virtual Terminal Services
COV and EventNotification services are not supported for BACnet
MSTP on the ProtoCessor
For BACnet MSTP , PTP and Arcnet , COV services are disabled by default and may be enabled by setting the Node_Option property to COV_Enable in the Nodes section configuration file.

PG-131-103-BX Gamewell FCI Protocol Driver Description

Formal Driver Type
Serial
Passive Client

Connection Information	
Connection type:	RS-232 or RS-485 (with converter)
Baud Rates:	Gamewell Panel: 2400 Driver: 110;300;600;1200;2400;4800;9600;19200;28800;38400;57600;115200
Data Bits:	Gamewell Panel: 8 Driver: 7,8
Stop Bits:	Gamewell Panel: 1 Driver: 1,2
Parity:	Gamewell Panel: None Driver: Odd, Even, None
Multidrop Capability	No

Data Type Supported

Data Type	Type of Information Stored	Notes
Any	Stores Status Information	Status:????. Stores non-zero value for any not –normal status.
Alarms	Stores Status Information	Status:ALARM sets array non-zero. NORMAL sets array to zero.
Faults	Stores Status Information	Status:FAULT sets array non-zero. NORMAL sets array to zero.
Events	Stores Status Information	Status:EVENT sets array non-zero. NORMAL sets array to zero.
Bus	Stores Status Information	Status:BUS sets array non-zero. NORMAL sets array to zero
Comm	Stores Status Information	Status:COMM sets array non-zero. NORMAL sets array to zero.
Control	Stores Status Information	Status:CONTROL sets array non-zero. NORMAL sets array to zero.
Ack	Stores Status Information	Status:ACK sets array non-zero. NORMAL sets array to zero.
Signal Silence	Stores Status Information	Status:SIG SIL sets array non-zero. NORMAL sets array to zero.
Troubles	Stores Status Information	Status:FAULT sets array non-zero. NORMAL sets array to zero. Status:EVENT and Action contains 'Supv. Event in ' sets array non-zero. Or Status:SUPV and any action.
Supervisories	Stores Status Information	NORMAL sets array to zero.
Action_Numbers	Stores Action information	Value based on contents of 'Action' Field

Action_Bits	Stores Action information	Sets bit whose offset is based on contents of 'Action' Field
Dump	Dump's ignored messages for user review.	

Supports:

Gamewell FCI E3 Series
Gamewell FCI 7100 Series
Gamewell-FCI 7200
Gamewell Serial Driver

Communications Functions

Supported Functions at a glance:

Listen. Driver listens passively for status messages, parses them looking for Node, Status, Circuit/Device and action information and stores data based on this information.

Write. Driver can send an Ack, Silence and Reset command. (To be provided in a later release.)

Supervision Query / Response

The driver clears its data arrays when the following messages is received.
Status:NORMAL 08/31/95 16:23
System Idle

Driver Limitations and Unsupported Features

The driver stores a value representing the type of status message received. A table of status types vs. Values is provided in the driver manual. Each message is inspected for circuit/device information. If none is present the message is assumed to report a status event for the panel. If the one or both are present then the circuit /device number is used to determine the storage location.

The driver can store a value to represent the status of a point (device / Circuit / panel) and/or a value to represent the 'action' that caused the most recent message to be sent.

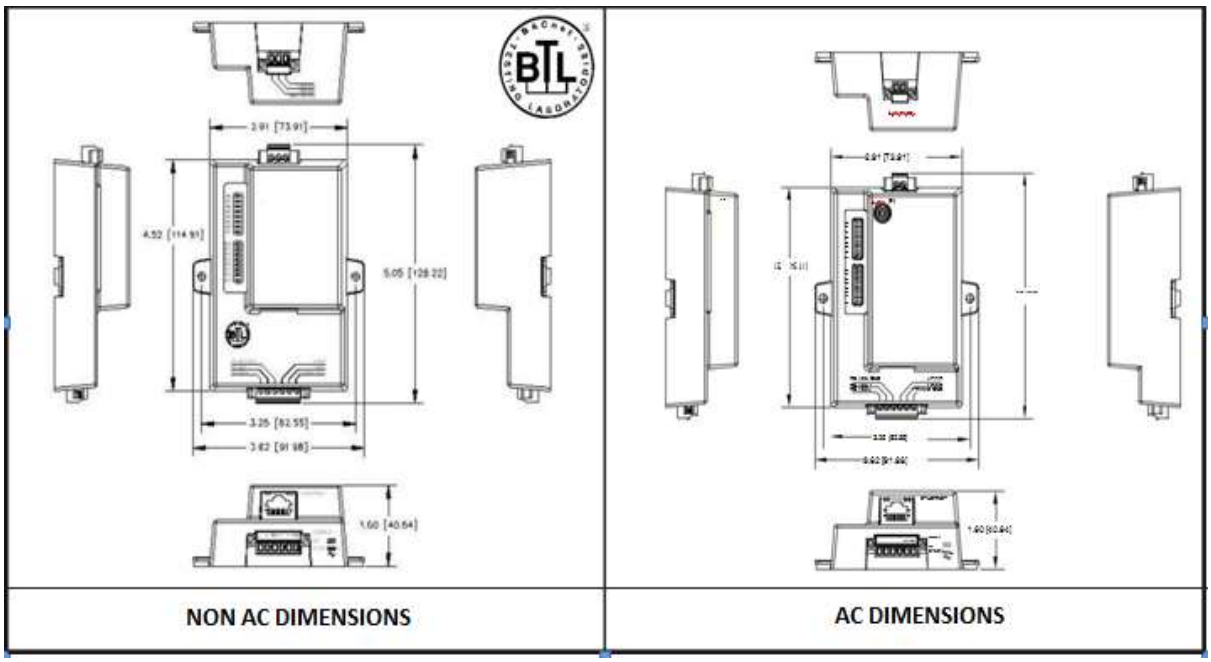
For messages reporting a status event for a circuit / device the driver uses only the device number to determine the location to store the indicating value.

The driver does not maintain an event / alarm history.

The value zero will be used to represent normal

The driver is programmed with a list of status types and action types that it recognizes. In the event that unrecognized information is found, the driver will store special value to indicate this. The driver provides a method which allows the user to extend the list of recognized status types and actions.

Dimensions:



NON AC DIMENSIONS	AC DIMENSIONS