BAFCON

BACnet Protocol Programming

OVERVIEW

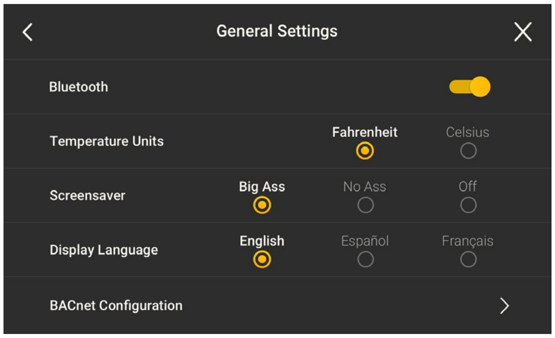
BAFCon is a BACnet MS/TP adapter for Big Ass Fans products that do not have native BACnet support, and it presents itself on the BACnet network as a Device. It also creates objects to interact with connected fans and lights, as well as with the BAFCon itself.

BAFCon is a BACnet master, and can therefore be discovered on the network by other masters, including BACnet MS/TP explorer applications.

Change of Value (COV) is not supported by BAFCon, and all points shall be polled by masters that are interested in their values.

CONFIGURATION OPTIONS

BAFCon ships with default values for MAC address, baud rate, and device instance number. These values can be viewed and edited in the main menu, under General > BACnet Configuration.



General Settings

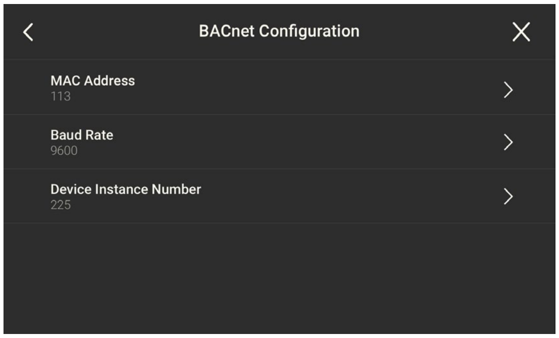
Bluetooth: on

Temperature Units: ● Fahrenheit ○ Celsius

Screensaver: ● Big Ass ○ No Ass ○ Off

Display Language: ● English ○ Español ○ Français

BACnet Configuration >



BACnet Configuration

MAC Address: 113 >

Baud Rate: 9600 >

Device Instance Number: 225 >

MAC Address

Every BACnet device on a network must have a unique MAC address. BAFCon is a BACnet master, and as such must have a MAC address between 0 and 127. Upon initial boot or after applying factory resets, BAFCon generates a unique MAC address. The MAC address is created by XOR-ing the three 32 bit words of the 96 bit unique identifier to get a seed. This seed is used to initialize the random utility. The MAC address is the lower 7 bits of the first random number returned by the rand function. Given that there is such a small range of available MAC addresses, it cannot be guaranteed that this address will be unique. Therefore, this address can also be manually adjusted on the BACnet Settings screen if necessary/desired.

Baud Rate

BAFCon supports all Baud rates given in the BACnet standard (9600, 19200, 38400, 57600, 76800, 115200). The default BAUD rate is 9600.

BAFCon Device Object ID

BACnet object IDs are composed of an object type and object instance number. The BACnet instance number is 22 bits in length.

The default BAFCon Device Instance Number is generated by calling the rand function after the MAC address. The maximum Device Instance Number is 4,194,302.

The Instance Number of any Device Object must be unique on the BACnet network. Although collisions will be rare when applying this scheme, the scheme does not guarantee uniqueness of the Device Instance Number on the BACnet network. Therefore, the Device Instance Number can be manually adjusted on the BACnet Settings screen if necessary/desired.

Note: Instance number 4194303 is disallowed by the standard.

Other Object IDs

Since multiple identical fans can be connected to BAFCon's Modbus port, it is necessary to ensure that the objects associated with them have unique object IDs within the BAFCon BACnet device. Modbus addresses for fans are guaranteed to be unique between 1 and 247. 100 instance numbers are reserved for objects of each type on each fan. The fan at Modbus address 1 reserves instance numbers of 100–199, fan 2 reserves instance numbers 200–299, and so on.

BAFCon reserves instance numbers 0–99 for objects that are embedded or directly connected.

SUPPORTED OBJECTS

Overview of Supported Objects

Object Name	Object ID
BAFCon-<Instance Number>	See Device Object ID
The following properties are present when the 0–10 V output is enabled.	
ZttDeviceType	MULTI-STATE_INPUT : 1
ZttPower	BINARY_VALUE : 1
ZttAutoEnable (fans only)	BINARY_VALUE : 2
ZttLevelPercent	ANALOG_VALUE : 1
The following are global properties.	
AutoIdealTemperature	ANALOG_VALUE : 10
ActualTemperature	ANALOG_INPUT : 10
UserPasscode	CHARACTERSTRING_VALUE : 10
AdminPasscode	CHARACTERSTRING_VALUE : 11
UserPasscodeEnable	BINARY_VALUE : 10
AdminPasscodeEnable	BINARY_VALUE : 11
The following properties are repeated for each Modbus connected fan.	
Fan <Modbus Addr> - FanType	MULTI-STATE_INPUT : Modbus Addr * 100 + 1
Fan <Modbus Addr> - Power	BINARY_VALUE : Modbus Addr * 100 + 1
Fan <Modbus Addr> - CommandedSpeedPercent	ANALOG_VALUE : Modbus Addr * 100 + 1

Fan <Modbus Addr> - ActualSpeedPercent	ANALOG_INPUT : Modbus Addr * 100 + 1
Fan <Modbus Addr> - IsForward	BINARY_VALUE : Modbus Addr * 100 + 2
Fan <Modbus Addr> - ResetFaults	BINARY_VALUE : Modbus Addr * 100 + 3
Fan <Modbus Addr> AutoEnable	BINARY_VALUE : Modbus Addr * 100 + 4
Fan <Modbus Addr> - ActiveFault	MULTI-STATE_INPUT : Modbus Addr * 100 + 2
Fan <Modbus Addr> - Name	CHARACTERSTRING_VALUE: Modbus Addr * 100 + 1

Individual Object Details

Property Identifier	**Value**
Object_Name	BAFCon-<Instance Number>
Object_Type	DEVICE
Instance_Number	See Device Object Instance Number
System_Status	OPERATIONAL
Vendor_Name	Big Ass Fans
Vendor_Identifier	1219
Model_Name	BAFCon
Firmware_Revision	BACnet stack's version
Application_Software_Version	BAFCon's firmware version
Protocol_Version	1
Protocol_Revision	19
Protocol_Services_Supported	Supported as per BACnet spec

Protocol_Object_Types_Supported	Supported as per BACnet spec
Object_List	Supported as per BACnet spec
Max_APDU_Length_Accepted	1476
Segmentation Supported	NO_SEGMENTATION
APDU Timeout	3000
Number_Of_APDU_Retries	0
Device Address Binding	N/A
Database Revision	Supported as per BACnet spec
Property List	Supported as per BACnet spec
Object_Name	ZttDeviceType$_2$
Object_Type	MULTI-STATE_INPUT
Instance_Number	See Overview of Supported Objects table
Present_Value	Supported as per BACnet spec
Status Flags	Supported as per BACnet spec
Event State	NORMAL
Reliability (Optional)	Supported as per BACnet spec
Out_Of_Service	Supported as per BACnet spec
Number Of States	6
State Text (optional)	{ “Light”, “AirEye”,

	“Essence”, “Pivot 2.0”, “Isis”, “Custom Fan”, “Hornet” }
Property_List	Supported as per BACnet spec

Property Identifier	**Value**
Object_Name	ZttPower
Object_Type	BINARY_VALUE
Instance_Number	See Overview of Supported Objects table
Present_Value	Supported as per BACnet spec
Status Flags	Supported as per BACnet spec
Event State	NORMAL
Reliability (Optional)	Supported as per BACnet spec
Out_Of_Service	Supported as per BACnet spec
Property_List	Supported as per BACnet spec

Property Identifier	**Value**
Object_Name	ZttAutoEnable
Object_Type	BINARY_VALUE
Instance_Number	See Overview of Supported Objects table
Present_Value	Supported as per BACnet spec

Status Flags	Supported as per BACnet spec
Event State	Supported as per BACnet spec
Reliability (Optional)	Supported as per BACnet spec
Out_Of_Service	Supported as per BACnet spec
Property_List	Supported as per BACnet spec

Property Identifier	**Value**
Object_Name	ZttLevelPercent
Object_Type	ANALOG_VALUE
Instance_Number	See Overview of Supported Objects table
Present_Value	Supported as per BACnet spec
Status Flags	Supported as per BACnet spec
Event State	Supported as per BACnet spec
Reliability (Optional)	Supported as per BACnet spec
Out_Of_Service	Supported as per BACnet spec
Units	Percent (98)
Property_List	Supported as per BACnet spec

Property Identifier	**Value**
Object_Name	AutoIdealTemperature
Object_Type	ANALOG_VALUE
Instance_Number	See Overview of Supported Objects table

Present_Value	Supported as per BACnet spec
Status Flags	Supported as per BACnet spec
Event State	NORMAL
Reliability (Optional)	Supported as per BACnet spec
Out_Of_Service	Supported as per BACnet spec
Units	Supported through Unit Selection
Property_List	Supported as per BACnet spec
Max_Present_Value	Supported as per BACnet spec
Min_Present_Value	Supported as per BACnet spec

Property Identifier	**Value**
Object_Name	ActualTemperature
Object_Type	ANALOG_INPUT
Instance_Number	See Overview of Supported Objects table
Present_Value	Supported as per BACnet spec
Status Flags	Supported as per BACnet spec
Event State	NORMAL
Reliability (Optional)	Supported as per BACnet spec
Out_Of_Service	Supported as per BACnet spec
Units	Supported through Unit Selection
Property_List	Supported as per BACnet spec

Property Identifier	**Value**
Object_Name	AdminPasscode
Object_Type	CHARACTERSTSRING_VALUE
Instance_Number	See Overview of Supported Objects table
Present_Value	Supported as per BACnet spec
Description	“0000–9999, default 0000"
Status Flags	Supported as per BACnet spec
Property_List	Supported as per BACnet spec
Property Identifier	**Value**
Object_Name	UserPasscode
Object_Type	CHARACTERSTSRING_VALUE
Instance_Number	See Overview of Supported Objects table
Present_Value	Supported as per BACnet spec
Description	“0000–9999, default 0000"
Status Flags	Supported as per BACnet spec
Property_List	Supported as per BACnet spec
Property Identifier	**Value**
Object_Name	AdminPasscodeEnable
Object_Type	BINARY_VALUE
Instance_Number	See Overview of Supported Objects table

Present_Value	Supported as per BACnet spec
Status Flags	Supported as per BACnet spec
Property_List	Supported as per BACnet spec

Property Identifier	**Value**
Object_Name	UserPasscodeEnable
Object_Type	BINARY_VALUE
Instance_Number	See Overview of Supported Objects table
Present_Value	Supported as per BACnet spec
Status Flags	Supported as per BACnet spec
Property_List	Supported as per BACnet spec

Property Identifier	**Value**
Object_Name	FanType-<Modbus Addr>
Object_Type	MULTI-STATE_INPUT
Instance_Number	See Overview of Supported Objects table
Present_Value	Supported as per BACnet spec
Status Flags	Supported as per BACnet spec
Event State	NORMAL
Reliability (Optional)	Supported as per BACnet spec
Out_Of_Service	Supported as per BACnet spec
Number Of States	5

State Text (optional)	{
	1 : “Powerfoil X”,
	2 : “Powerfoil D”,
	3 : “PF8/Basic 6”,
	4: “Powerfoil B”,
	5: ”Cool Space 3”
	}

Property_List	Supported as per BACnet spec

Property Identifier	**Value**
Object_Name	Power-<Modbus Addr>
Object_Type	BINARY_VALUE
Instance_Number	See Overview of Supported Objects table
Present_Value	Supported as per BACnet spec
Status Flags	Supported as per BACnet spec
Event State	NORMAL
Reliability (Optional)	Supported as per BACnet spec
Out_Of_Service	Supported as per BACnet spec
Property_List	Supported as per BACnet spec

Property Identifier	**Value**
Object_Name	CommandedSpeedPercent-<Modbus Addr>
Object_Type	ANALOG_VALUE
Instance_Number	See Overview of Supported Objects table
Present_Value	Supported as per BACnet spec

Status Flags	Supported as per BACnet spec
Event State	NORMAL
Reliability (Optional)	Supported as per BACnet spec
Out_Of_Service	Supported as per BACnet spec
Units	Percent (98)
Property_List	Supported as per BACnet spec
Max_Present_Value	Supported as per BACnet spec
Min_Present_Value	Supported as per BACnet spec

Property Identifier	**Value**
Object_Name	ActualSpeedPercent-<Modbus Addr>
Object_Type	ANALOG_INPUT
Instance_Number	See Overview of Supported Objects table
Present_Value	Supported as per BACnet spec
Status Flags	Supported as per BACnet spec
Event State	NORMAL
Reliability (Optional)	Supported as per BACnet spec
Out_Of_Service	Supported as per BACnet spec
Units	Percent (98)
Property_List	Supported as per BACnet spec

Property Identifier	**Value**

Object_Name	IsForward-<Modbus Addr>
Object_Type	BINARY_VALUE
Instance_Number	See Overview of Supported Objects table
Present_Value	Supported as per BACnet spec
Status Flags	Supported as per BACnet spec
Event State	NORMAL
Reliability (Optional)	Supported as per BACnet spec
Out_Of_Service	Supported as per BACnet spec
Property_List	Supported as per BACnet spec
Property Identifier	**Value**
Object_Name	ResetFaults-<Modbus Addr>
Object_Type	BINARY_VALUE
Instance_Number	See Overview of Supported Objects table
Present_Value	Supported as per BACnet spec
Status Flags	Supported as per BACnet spec
Event State	NORMAL
Reliability (Optional)	Supported as per BACnet spec
Out_Of_Service	Supported as per BACnet spec
Property_List	Supported as per BACnet spec
Property Identifier	**Value**

Object_Name	AutoEnable-<Modbus Addr>
Object_Type	BINARY_VALUE
Instance_Number	See Overview of Supported Objects table
Present_Value	Supported as per BACnet spec
Status Flags	Supported as per BACnet spec
Event State	NORMAL
Reliability (Optional)	Supported as per BACnet spec
Out_Of_Service	Supported as per BACnet spec
Property_List	Supported as per BACnet spec

Property Identifier	**Value**
Object_Name	ActiveFault-<Modbus Addr>
Object_Type	MULTI-STATE_INPUT
Instance_Number	See Overview of Supported Objects table
Present_Value	Supported as per BACnet spec
Status Flags	Supported as per BACnet spec
Event State	NORMAL
Reliability (Optional)	Supported as per BACnet spec
Out_Of_Service	Supported as per BACnet spec
Number Of States	12

State Text (optional)	{
	“No Fault”
	“External Fault”,
	“Over Voltage”,
	“Overload”,
	“Ground Fault”,
	“Low Voltage”,
	“Internal Fault”,
	“AC Input Phase Loss”,
	“Rotor Control”,
	“Over Temperature”,
	“Current Mismatch”,
	“Impact Detected”
	}

Property_List	Supported as per BACnet spec

Property Identifier	**Value**
Object_Name	FanAddr-<Modbus Addr> Name
Object_Type	CHARACTERSTSRING_VALUE
Instance_Number	See Overview of Supported Objects table
Present_Value	Supported as per BACnet spec
Description	“Fan Name, 1-32 characters”
Status Flags	Supported as per BACnet spec
Property_List	Supported as per BACnet spec

USAGE EXAMPLES

Power on a fan

Write a “1” to the **Present_Value** property of the **Power** object of the target fan.

Set Fan Speed

Write a value between 1 and 100 to the **Present_Value** property of the **CommandedSpeedPercent** object of the target fan.

Clear Faults

Write a “1” to the **Present_Value** property of the **ResetFaults** object of the target fan. The **Present_Value** of the **ResetFaults** object and the **ActiveFault** object will be set to “0”.

Get Fault Status

Read the **Present_Value** property of the **ActiveFault** object of the target fan. The actual fault number will be returned.

Change Direction

Write a “0” to the **Present_Value** of the **IsForward** object of the target fan to set the direction to reverse, or write a “1” to set the direction to forward.

Change Name of a Fan

Write “Desired Fan Name” (up to 32 characters) to the **Present_Value** property of the **Name** object of the object of the target fan.

USAGE RECOMMENDATIONS

Reading Properties

Properties can be read using *readProperty* or *readPropertyMultiple* data requests. The latter allows for multiple properties to be read with a single request.

Important *readPropertyMultiple* Limitations

BAFCon limits *readPropertyMultiple* to a maximum of 16 explicit properties per request, due to packet size limits. If more than 16 explicit properties are requested, BAFCon will respond with an *outOfResources* error.

BAFCon does support *readPropertyMultiple "all"* requests for all properties. However, if the resulting response is too large due to a large quantity of properties present, BAFCon will not respond to the request. The recommendation is to read single properties using *readProperty,* or use *readPropertyMultiple* with <=16 properties per request.

Reading the Object List for the BAFCon Device

There are cases where a single property request can lead to BAFCon not responding due to the large packet size required to provide the data. A request for the *object-list* for the BAFCon device is one instance where this can occur if there are a large quantity of fans connected. In general, if there are more than 8 fans attached, a request for the entire object list will fail (BAFCon will not respond) due to packet size limitations. In this case, when requesting the object list, the object-list property identifier should be used with an included *property array index* to get objects individually.