San Telequip (P) Ltd.,
504/505 Deron Heights, Baner Road, Baner
Pune 411045, Mah, India
Phone : +91 20-27293456, 8390069393, 9764027070
email : info@santelequip.com

USER MANUAL FOR BMS PROTOCOL GATEWAY

San Telequip Pvt. Ltd.
504/505, Deron Heights, Baner Road,
Pune 411045, India
www.santelequip.com

San Telequip (P) Ltd.,
504/505 Deron Heights, Baner Road, Baner
Pune 411045, Mah, India
Phone : +91 20-27293456, 8390069393, 9764027070
email : info@santelequip.com

Table of Contents

1 INTRODUCTION

BMS protocol gateway is a protocol translator between Serial-Serial, Serial-Ethernet devices using MODBUS RTU, MODBUS TCP, BACnet MSTP, BACnet IP, SNMP, Metasys N2, Lon Works®, DNP3 Eth. protocol. Depending on the model, the gateway has one or two serial ports and one Ethernet port. The Gateway needs to be configured as per engineering details of field equipment's to be connected to the gateway.

2 SPECIFICATIONS

2.1 Interfaces

Serial Interface	
SERIAL PORT	: 1 RS-485, 2 Wire, 3 Pin Screw type terminals. (For GW BMS 01, GW BMS 02 & GW BMS 05 Series)
SERIAL PORT	: 2 RS-485, 2 Wire, 3 Pin Screw type terminals. (For GW BMS 01-02, GW BMS 02-02 GW BMS 02-02R, GW BMS MM, GW BMS SS and GW BMS 04 Series)
LonWorks Port	: 1 LON Connector 3 pin Screw type terminal (For GW BMS 03 only)
LAN	
Ethernet	: 10/100Mbps – Auto detecting
No. of Ports	: 1 Port
Connector	: RJ45
Indications	: LED's for Power, RS485 Tx and Rx
Power Supply	
Power supply	: 24V DC, Range 18 to 30 VDC, 100mA
Connector	: 3 Pin Screw type
Environmental	
Operating Temp.	: -40°C to 85°C
Relative Humidity	: 5-90% RH, non-condensing
Dimensions	: L x W x H: 46.5 x 84.5x106.5 mm (For GW BMS 01, GW BMS 02, GW BMS 01 02, GW BMS 02 02, GW BMS 02 02R, GW BMS 04 Series and GW BMS 05 Series) L x W x H: 50 x 101 x 105 mm (For GW BMS 03)

2.2 DIP Switch setting

DIP switch setting is used to enable termination on RS485 Bus. DIP switch is available on top side of the GW BMS GATEWAY. Below are DIP switches positions.

For GW BMS 01, GW BMS 02 and GW BMS 05 Series.

Serial Port	Switch 1	Switch2
S1	ON	OFF

For GW BMS 01 02, GW BMS 02 02, GW BMS 02-02R and GW BMS 04 Series

Serial Port	Switch 1	Switch2
R1	ON	OFF
R2	OFF	ON

3 OPERATION

BMS protocol gateway operates for conversion of protocol as per the driver configuration file loaded into it. Depending on driver it can be operated in following modes –

1) Modbus RTU - BACnet IP	GW BMS 01
2) Modbus RTU – BACnet IP (2- RS 485 ports)	GW BMS 01-02
3) BACnet MSTP – BACnet IP	GW BMS 02
4) BACnet MSTP – BACnet IP -Router (2-RS 485 ports)	GW BMS 02 02R
5) LonWorks – BACnet IP	GW BMS 03
6) Modbus RTU - Metasys N2 BACnet MSTP - Metasys N2 Metasys N2 – BACnet IP	GW BMS 04
7) SNMP to BACnet IP SNMP to Modbus TCP BACnet IP to Modbus TCP Ethernet IP to Modbus RTU/Modbus TCP DNP3 Eth. To Modbus TCP	GW BMS 05
8) Modbus Master Master Gateway	GW BMS MM
9) Modbus Slave Slave Gateway	GW BMS SS

BMS protocol gateway can support mentioned protocol vice versa.

4 Configuration of BMS Gateway

4.1 Overview

The GW BMS GATEWAY has a web-browser based User Interface, and uses a combination of technologies and devices to provide a platform that the user can interact with for the tasks of easily gathering and producing information.

4.2 Use of WEB GUI

1. Check the status and diagnostics of a GW BMS Gateway including such information as network settings, connection info, node information, map descriptors, and error messages
2. Monitor a working Gateway's internal data and parameters
3. Change or update a Gateway's internal data and parameters
4. Transfer files to and from a Gateway
5. Delete files on a Gateway
6. Change the Gateway's IP address
7. Restart a Gateway

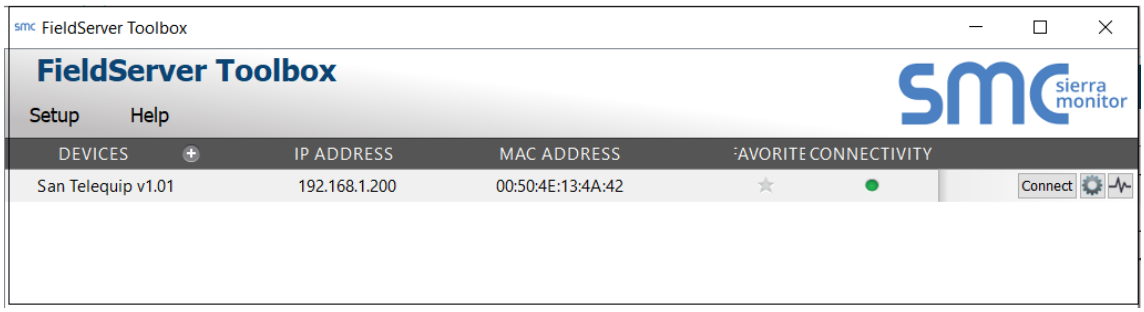
4.3 Using the Web Browser to launch GUI (Graphic User Interface)

Connect the Gateway to pc through straight Ethernet cable and power on the unit. Type the IP address of gateway directly into the web browser and press enter, GUI will be launching as shown GUI Landing Page as below.

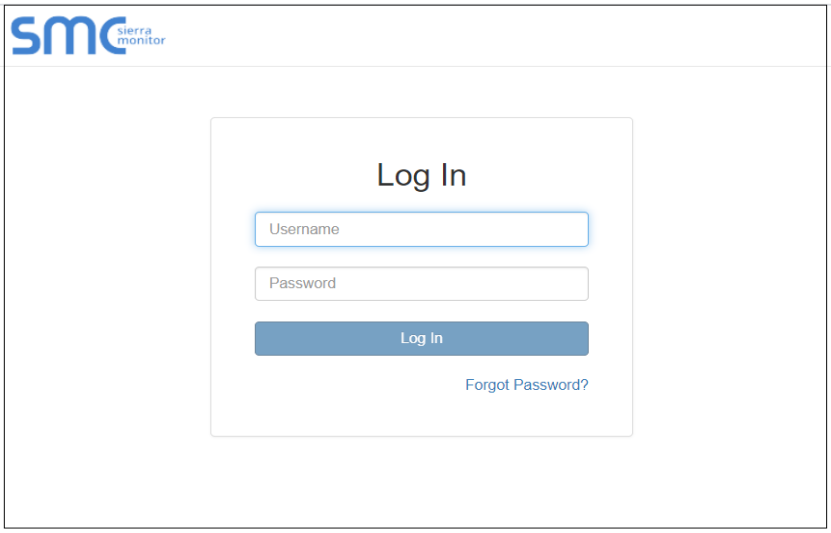
4.4 Using the FieldServer Toolbox

If browser is not able to connect to the gateway, use FS toolbox. Install FS Toolbox software utility on this PC. Refer below link to download FS Tool box.
http://www.santelequip.com/download/BMS%20gateway/Fieldserver%20toolbox/FieldServer-Toolbox-1.05aA-Setup.zip

Use the FS Toolbox to find the GW BMS GATEWAY, and launch the GUI. FS tool box will automatically discover the gateway on the network and show below screen. Default IP address of GW BMS Gateway is **192.168.1.200**.



WEB GUI Landing Page
When the login screen appears, put in the Username (default is "ADMIN") and the Password (Spassword@123).



NOTE: A user has 5 attempts to login then there will be a 10-minute lockout. There is no timeout on the GW BMS GATEWAY to enter a password.

4.5 NAVIGATION TREE

After successful login to GW BMS GATEWAY configuration page will be open.

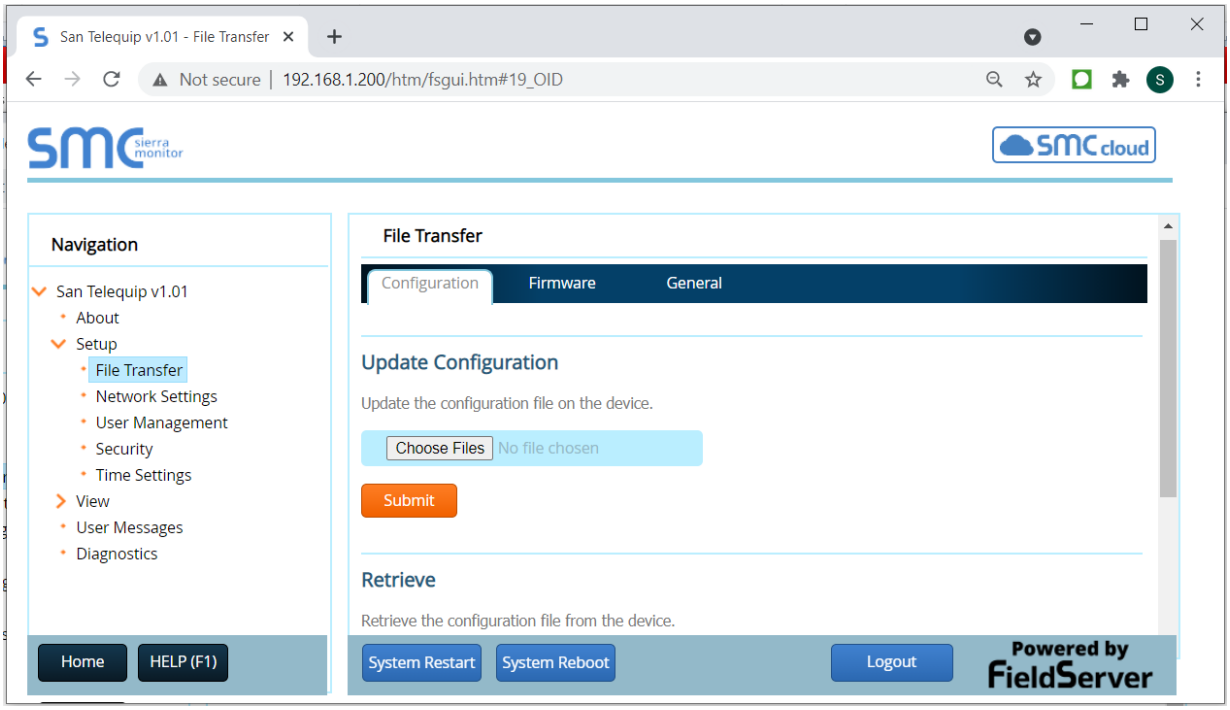
4.5.1 Root

The root of the navigation tree allows the user to check the status of the gateway, including the configuration code, version, memory, gateway type and more. Under "Settings" the user has access to important network information. The name of the root is specified in the Gateways Configuration file under the Title Keyword.

4.5.2 About

Allows the user to check current firmware of the gateway plus version identification of the interface and skin. Skin is either the default gateway template or it can be a specific template specified by the owner.

4.5.3 Setup



4.5.4 File Transfer

There are 3 types of files that can be transferred, namely Configuration Files, Firmware and Miscellaneous (general) files.

4.5.4.1 Configuration Files

Configuration files have a .csv extension, and are used to configure the gateway for its specific application.

- To Update the configuration file

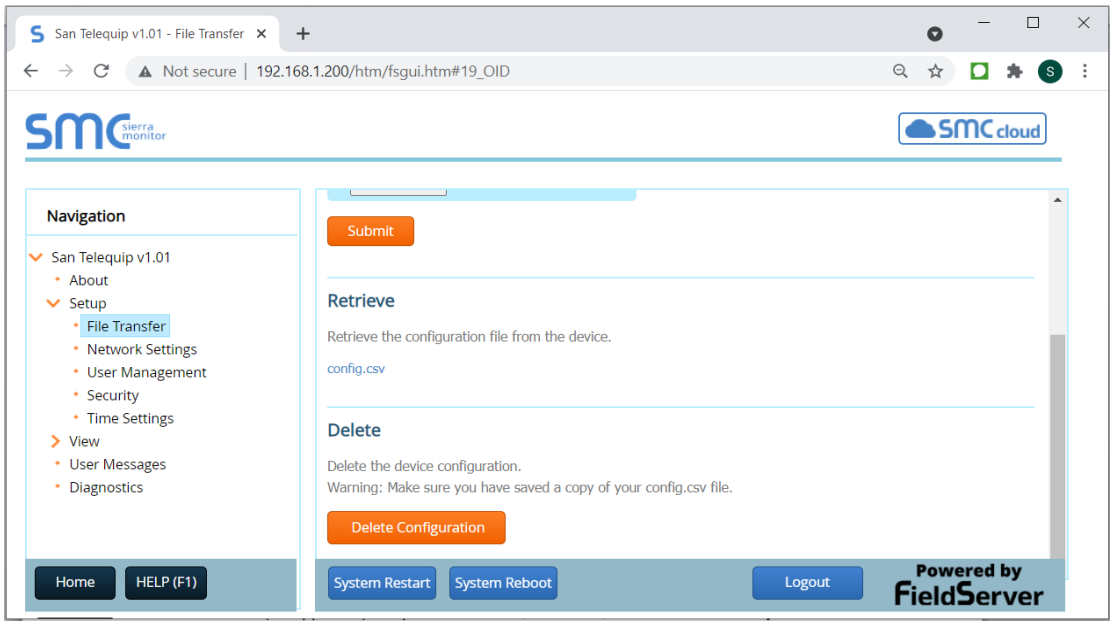
To update the Gateway's configuration file, click the browse button and select the configuration file (.csv). Click open, and submit. Wait until the message "Configuration update complete" appears, and click the System Restart button to activate the new configuration file.

- To Retrieve the configuration file

For making changes to the configuration file: Retrieve the file, edit it, save the updated file and update the file

- To Delete the configuration file

To temporarily disable the Gateway's protocol communications, the configuration can be deleted. The Gateway needs to be restarted to activate the changes.
Warning: This action cannot be undone - make sure you have saved a copy of your configuration file.



4.5.4.2 Firmware Files

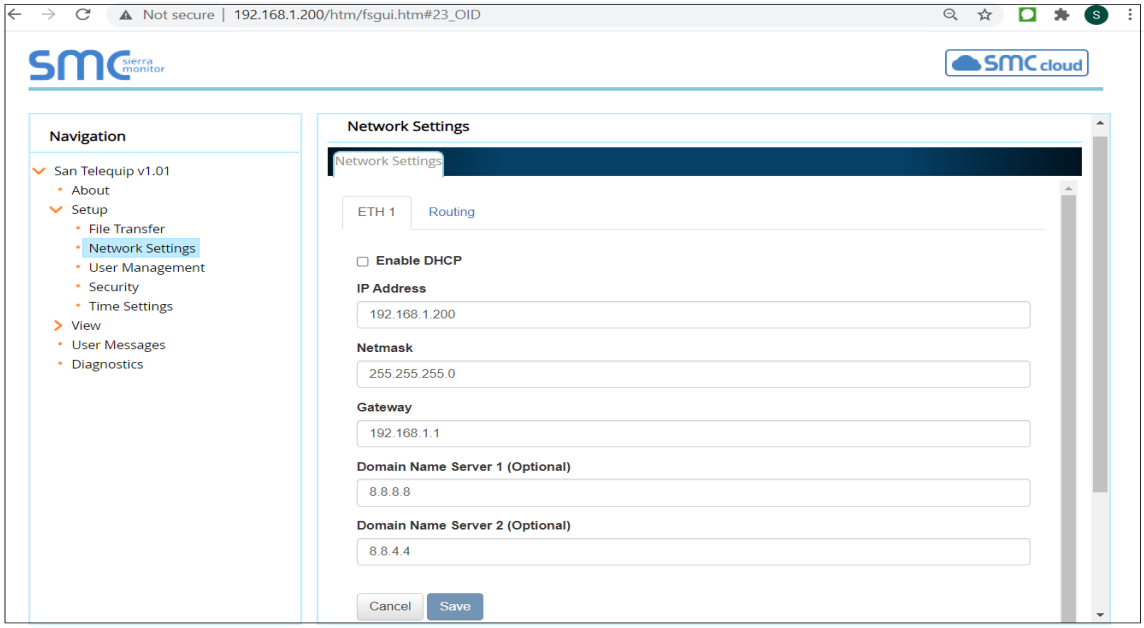
The Gateway Firmware contains the application program commonly referred to as the DCC or the PCC. This program contains the protocol drivers applicable to the application and the FieldServer Operating System Kernel.
A Firmware update is only required when updated files are received from service support. Firmware files have a .bin extension.

4.5.4.3 General (Other) Files

Other files can be updated, the procedure for updating these is the same as for configuration files, but the update needs to be made in the “General” update section.

4.5.5 Network Settings

On the Network Settings Page, the Ethernet adaptor settings of the Gateway can be changed. The N1 and N2 (if supported) adaptor IP address, Netmask, and default Gateway can be changed by entering values in the applicable fields and clicking on the Update IP settings Button.
NOTE: The Gateway has to be restarted for any changed settings to take effect. Also note that enabling the DHCP client on any adaptor will cause the static IP address settings to be overruled by a DHCP server on the network.
The Gateway’s built-in DHCP server can be enabled to establish easy connection for Support purposes. Set the laptop or computer to automatically obtain an IP address to use this feature. Note that the FieldServer DHCP server periodically checks for other DHCP servers on the network and will disable itself should any other DHCP servers exist on the network. This mode of operation is because the Gateway DHCP server is strictly for support purposes and does not have all the features of a commercial DHCP server. Setting the default gateway IP address to your network gateway will ensure the BMS converter is reachable on the internet.



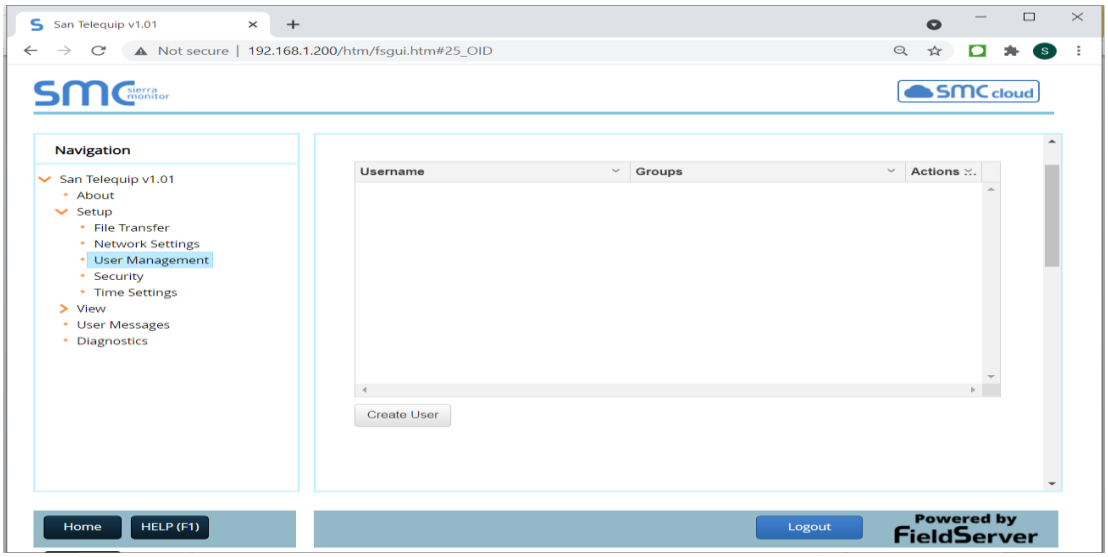
Routing is used only if there are two Ethernet ports.

4.5.6 User Management

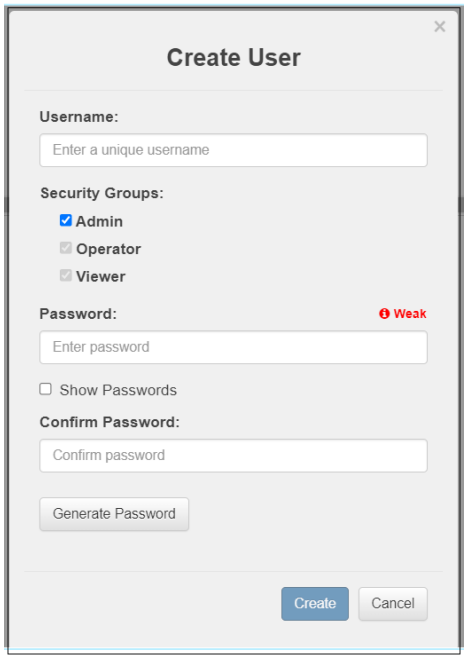
Access to the Gateway is restricted by username and password. There are 2 access levels defined by 3 User types: Admin, Operator and Viewer.
Default Username: **ADMIN** and Password: **Spassword@123**

User Types:
Admin – Can modify and view any settings on the GW BMS GATEWAY.
Operator – Can only make changes to the writable device points on the GW BMS GATEWAY.
Viewer – Can only view settings/readings on the GW BMS GATEWAY.

NOTE: If the passwords are lost contact to San Telequip team.

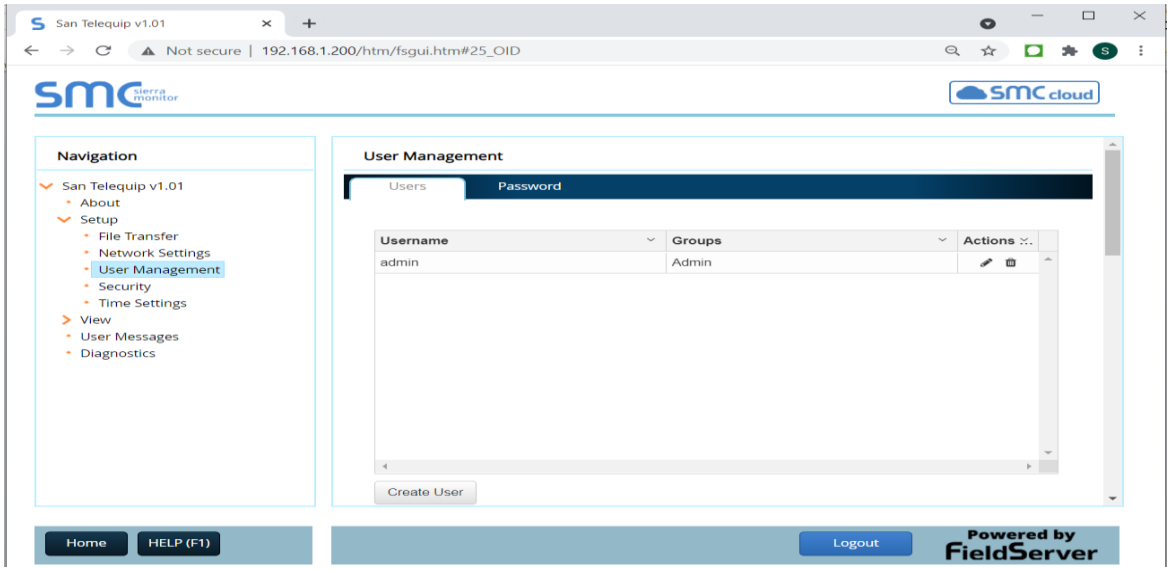


Click on Create User.



Enter the new User fields: Name, Security Group and Password and Click the Create button. Once the Success message appears, click OK.

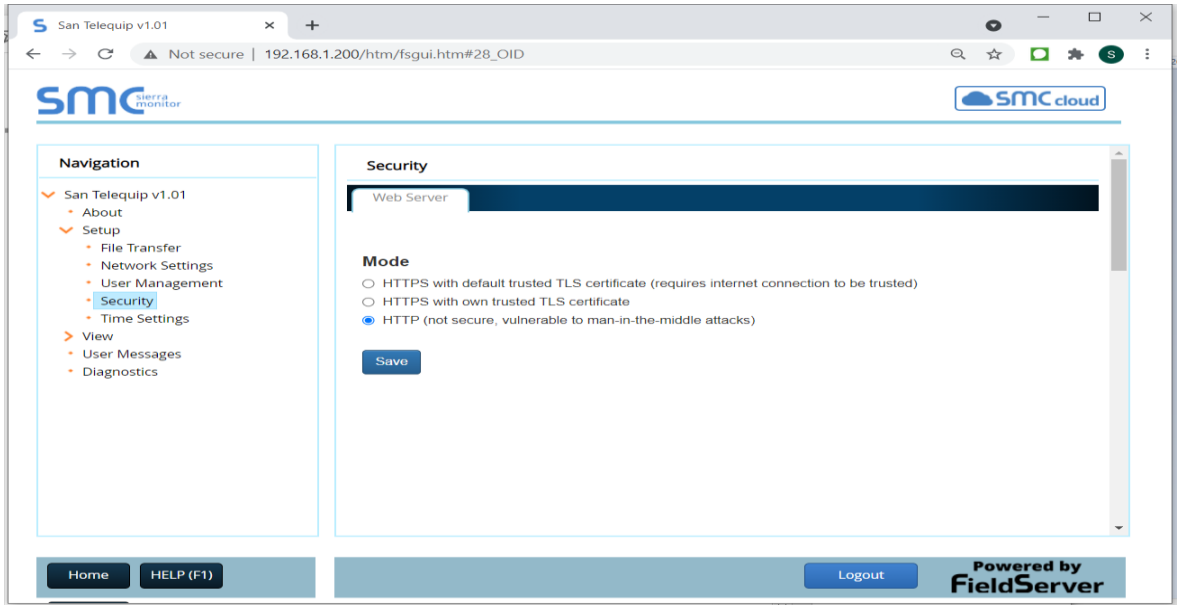
NOTE: Passwords must be at least 10 characters long. An algorithm automatically checks the password entered and notes the level of strength on the top right of the Password text field



Do not delete the default entry. Customer can modify the default entry if required.

4.5.7 Security

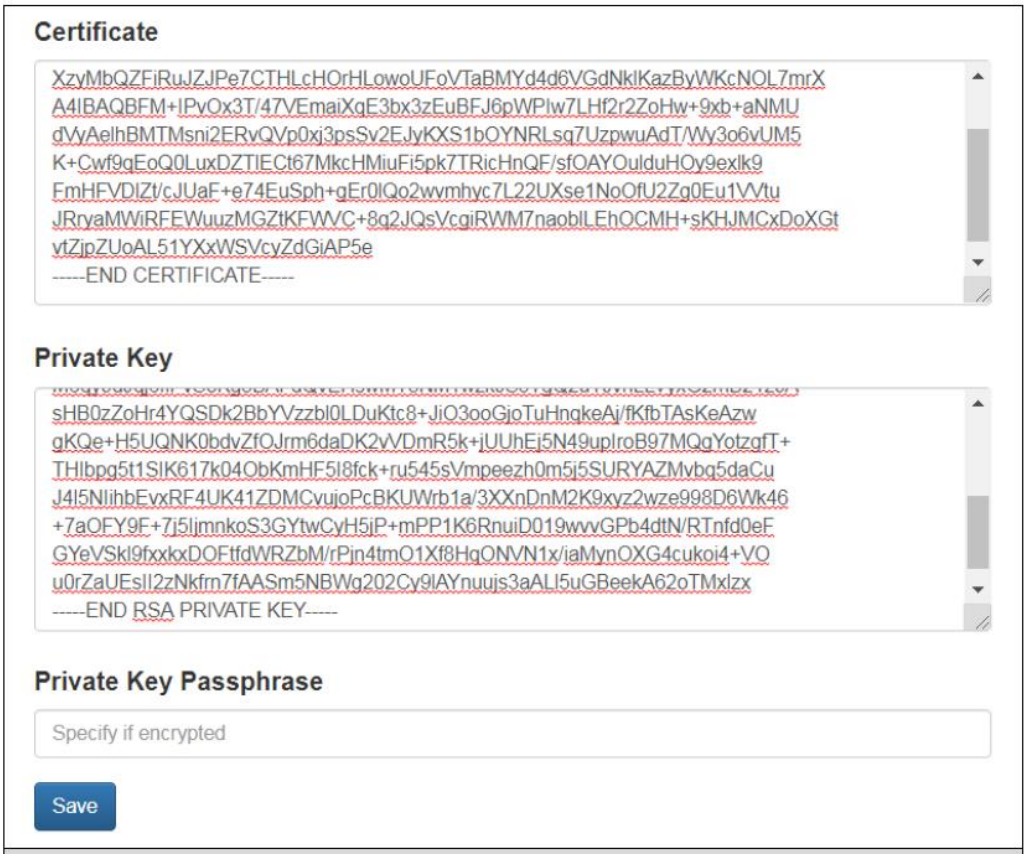
In security option customer have option to select http for HTTP and HTTPS.
By default it's selected to HTTP. Select desired security mode.



If HTTPS with own trusted TLS certificate is selected

This is the recommended selection and the most secure. Please contact your IT department to find out if you can obtain a TLS certificate from your company before proceeding with the Own Trusted TLS Certificate option.

Once this option is selected, the Certificate, Private Key and Private Key Passphrase fields will appear under the mode selection.



Copy and paste the Certificate and Private Key text into their respective fields. If the Private Key is encrypted type in the associated Passphrase.

Click Save.

A “Redirecting” message will appear. After a short time, the WEB GUI will open.

If HTTPS with Default Untrusted Self-Signed TLS Certificate is selected

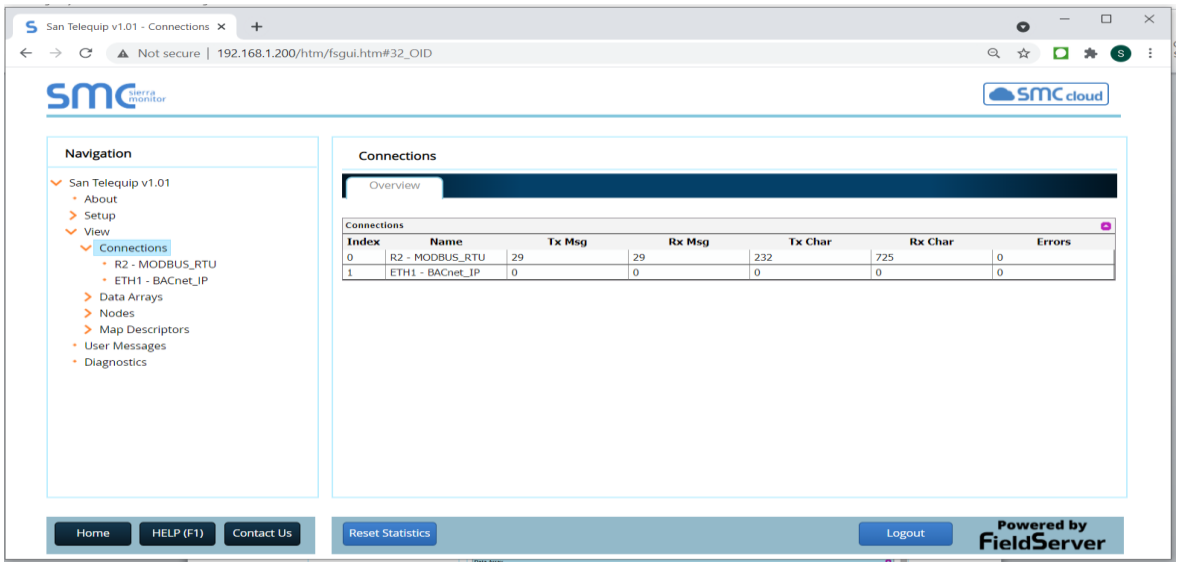
Select one of these options and click the Save button.

A “Redirecting” message will appear. After a short time, the WEB GUI will open.

4.6 View

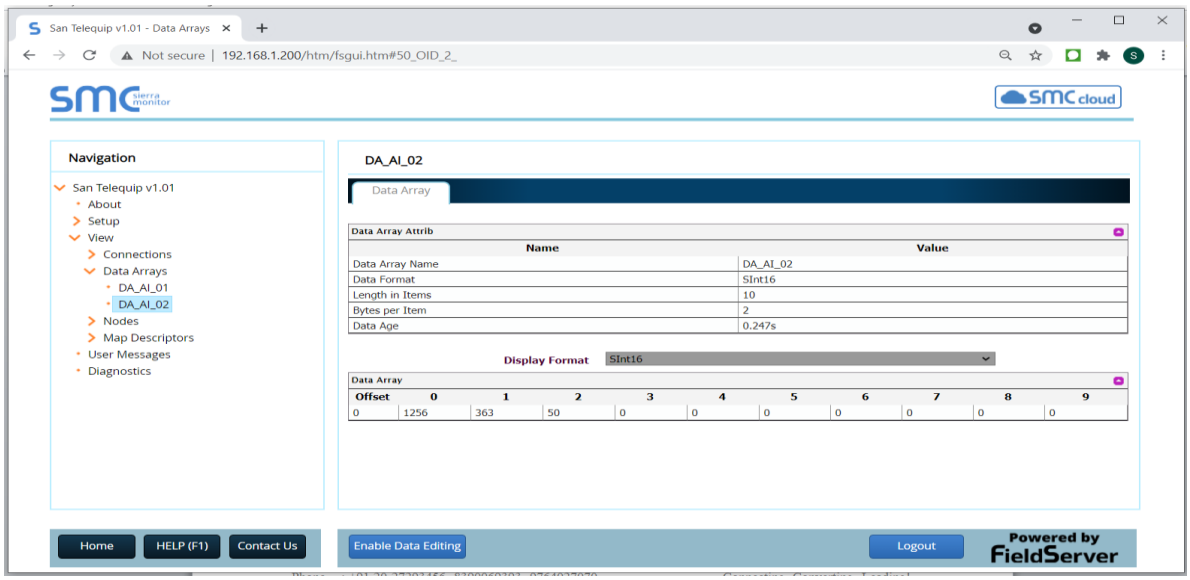
4.6.1 Connections

The Connections screen supplies information on communication between the Gateway and remote devices. A number of aspect screens are available, including settings, info stats and error stats. The information on these screens cannot be changed, and is for viewing only.



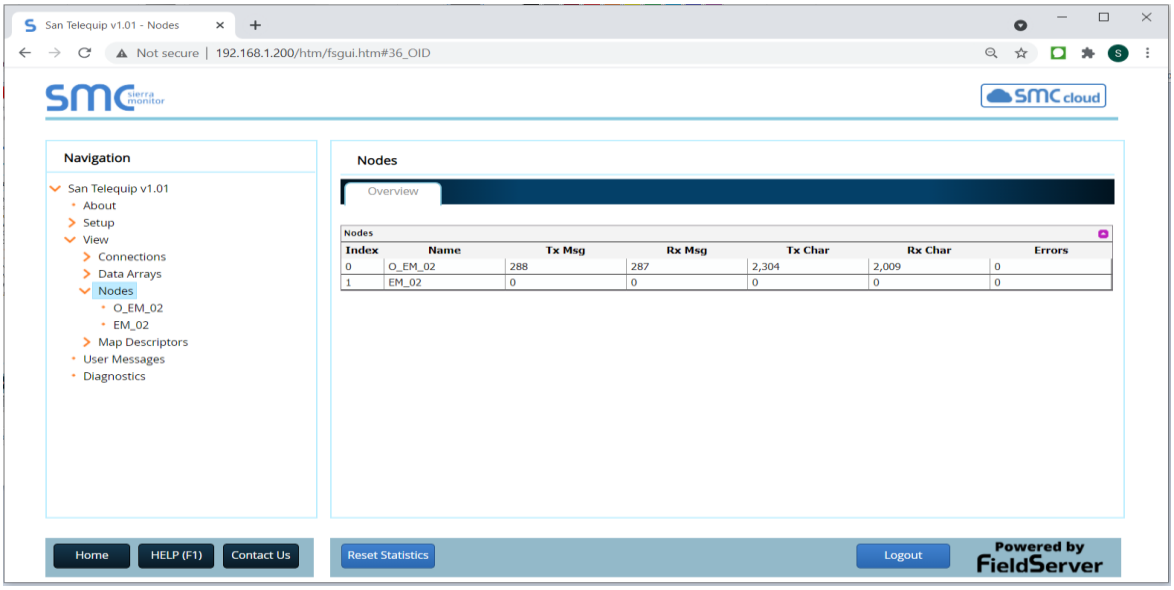
4.6.2 Data Arrays

The Data Arrays screens can be used to view the values in Data Arrays. The values can be changed by clicking on the “Enabled Grid”- button, and changing the value in the data array grid. Note that if values are being written into the Array by a driver, then any modifications made by grid editing will be overridden.



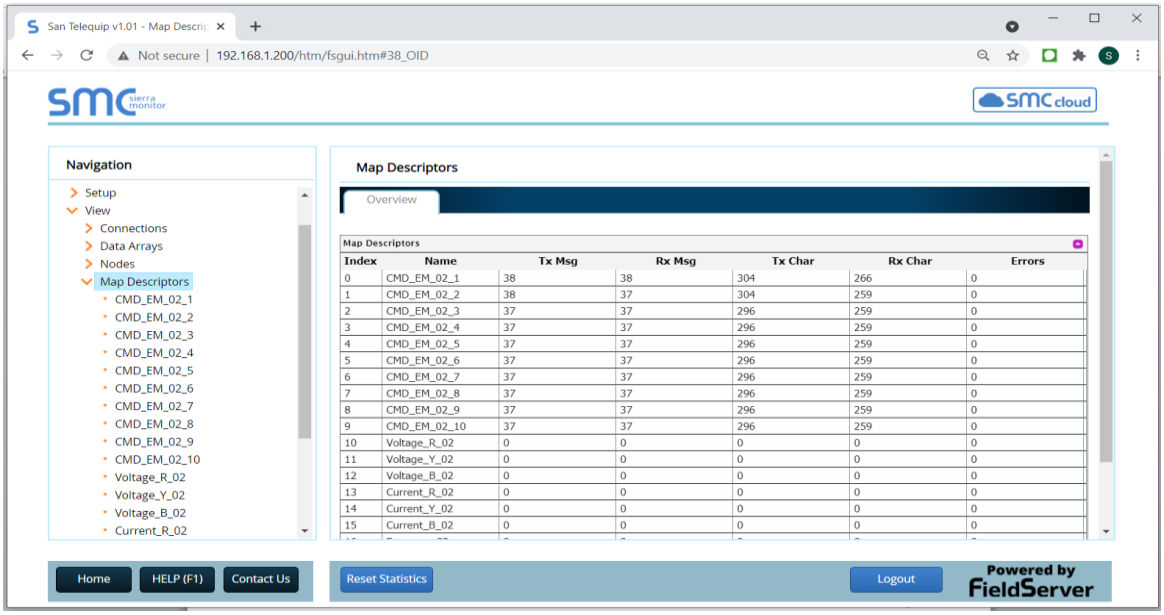
4.6.3 Nodes

On the Nodes screens information about the remote devices on each connection can be viewed. A number of aspect screens are available, including settings, status, info stats and error stats. The information on these screens cannot be changed and is for viewing only.



4.6.4 Map Descriptors

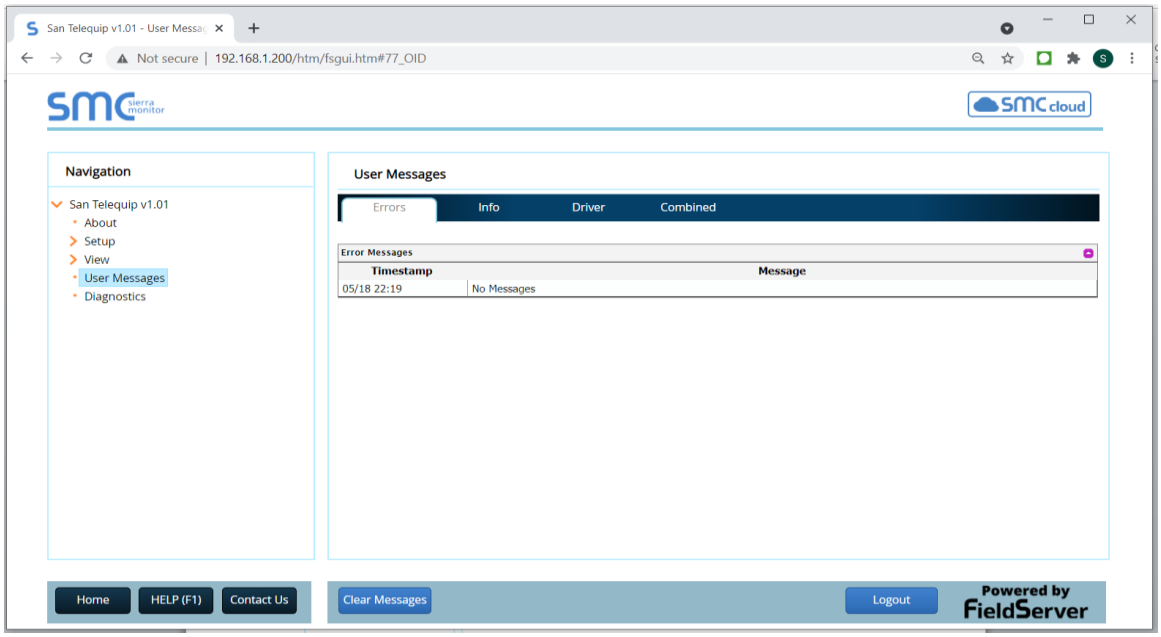
On the Map Descriptors screens information on each individual Map Descriptor can be viewed. Several aspect screens are available, including settings, status, info stats and error stats. The information on these screens cannot be changed and is for viewing only.



4.6 User Messages

The user message screens display Gateway messages generated by drivers and the operating system.

User messages on the “Error”- screen usually indicate some problem with the configuration or communication and should be attended to. User messages of an informational type will be displayed on the “Info”- screen, and no user action is usually required. Messages generated by protocol drivers will be displayed on the “Driver”- screen. These messages convey protocol specific information that can be useful for field integration purposes. Finally, the “Combined”- screen contains all messages chronologically from all the above message screens.

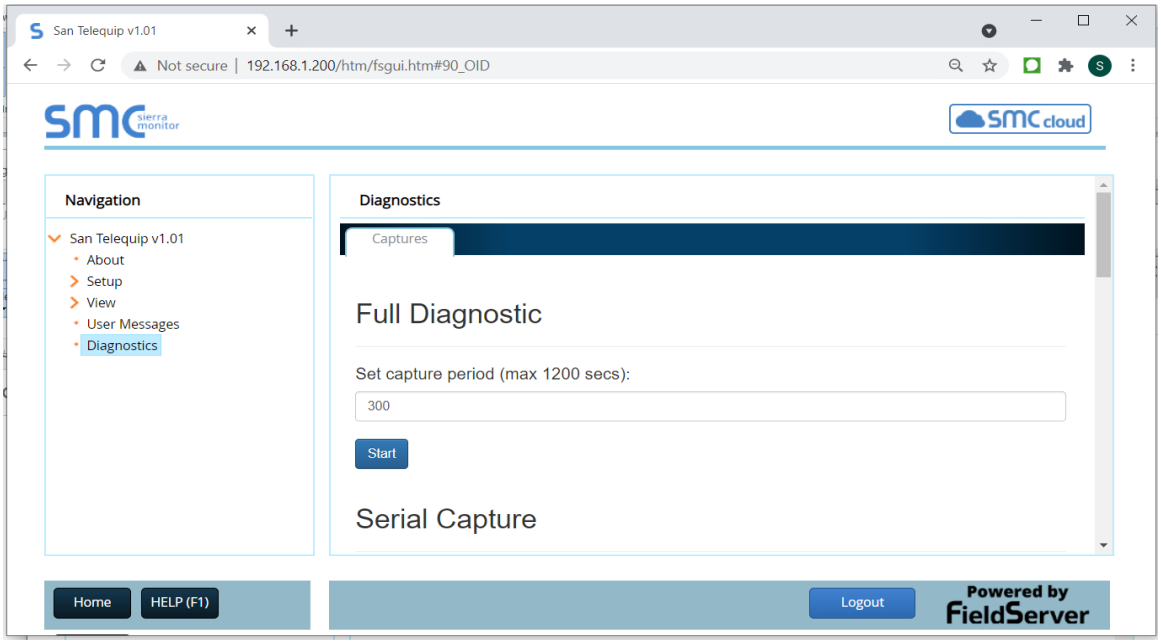


4.7 Taking a GW BMS GATEWAY Diagnostic Capture

When there is a problem on-site that cannot easily be resolved, perform a Diagnostic Capture

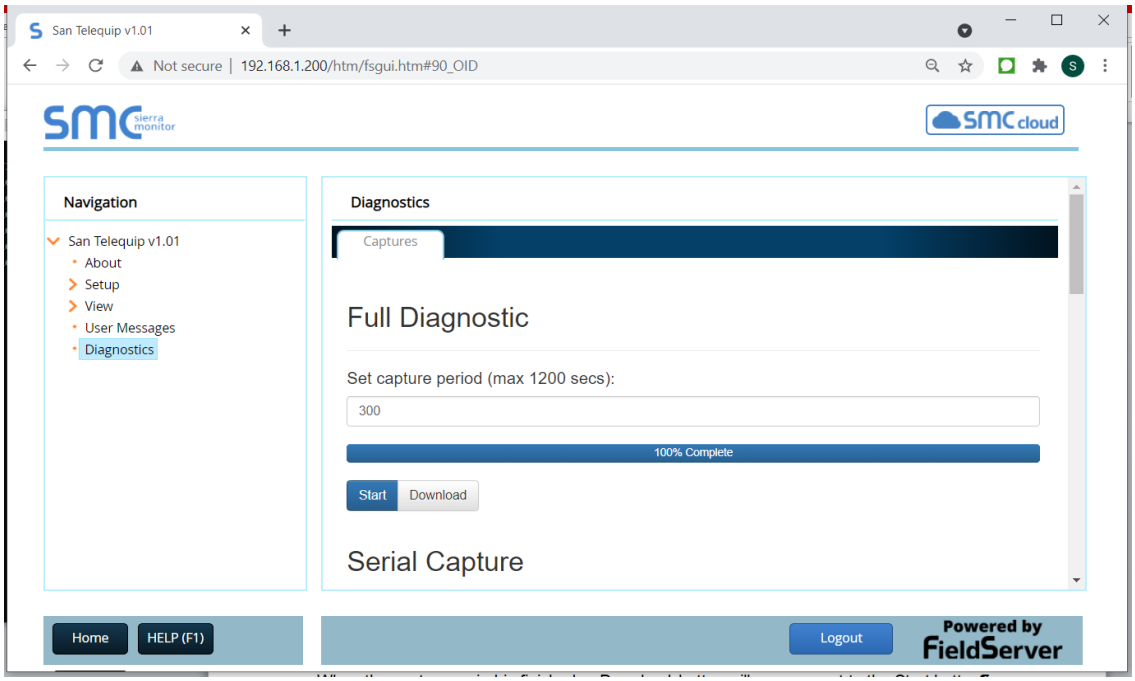
Access the GW BMS GATEWAY Diagnostics page via one of the following methods:

- Open the GW BMS GATEWAY FS-GUI page and click on Diagnostics in the Navigation panel
- Open the FieldServer Toolbox software and click the diagnose icon of the desired device



Go to Full Diagnostic and select the capture period.
Click the Start button under the Full Diagnostic heading to start the capture.

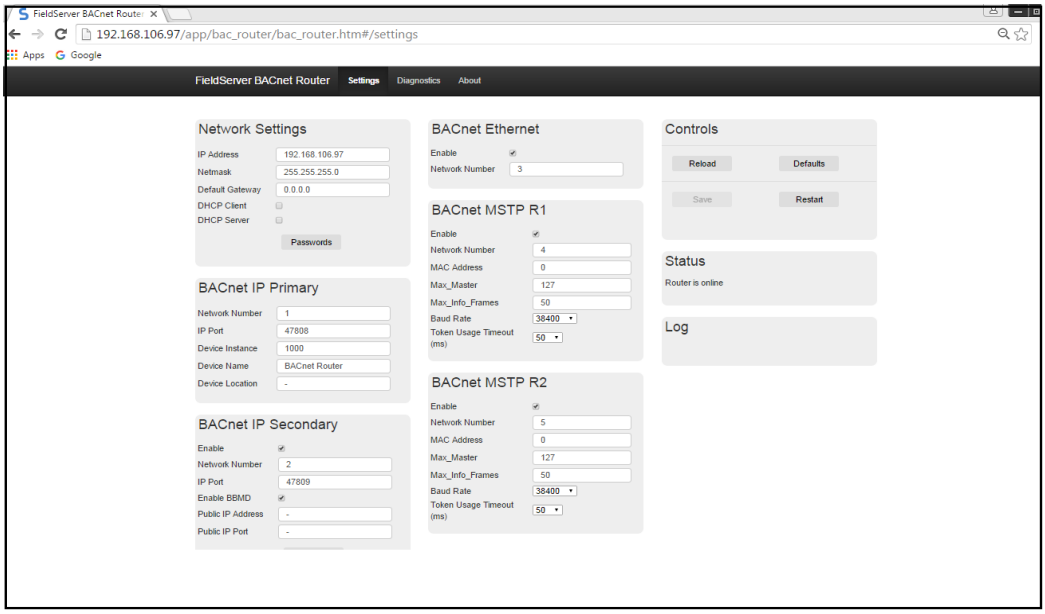
When the capture period is finished, a Download button will appear next to the Start button



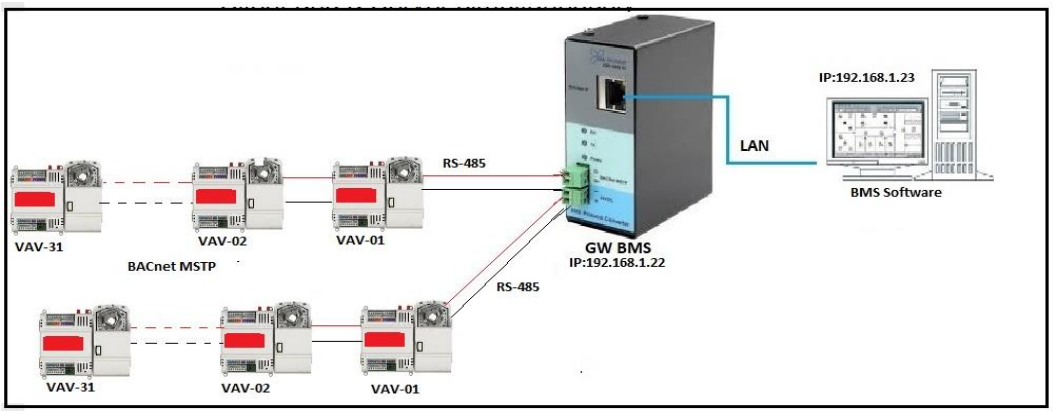
Click Download for the capture to be downloaded to the local PC.
Email the diagnostic zip file to santelequip technical support team.

5 PROCEDURE TO CONFIGURE BACnet Router GW BMS 02-02R

1. Connect the GWBMS02-02R to the network using straight patch chord. The PC and gateway should be on same subnet.
2. Open a web browser on PC and put IP address of BACnet router.
   E.g. If gateway IP address is 192.168.1.200, put this IP into browser. (PC IP should be 192.168.1.X).Web page settings will open as shown in below snapshot.
3. To change the IP addresses of router go to Network Setting Option. Put new IP and press System restart.
4. To set the BACnet MSTP-1 Port baud rate, make changes in 'BACnet MSTP R2'Setting.
5. To set the BACnet MSTP-2 Port baud rate, make changes in 'BACnet MSTP R1' Setting.
6. After all configuration changes, press system restart.



5.1 BACnet Router GW BMS 02-02R Connection Diagram

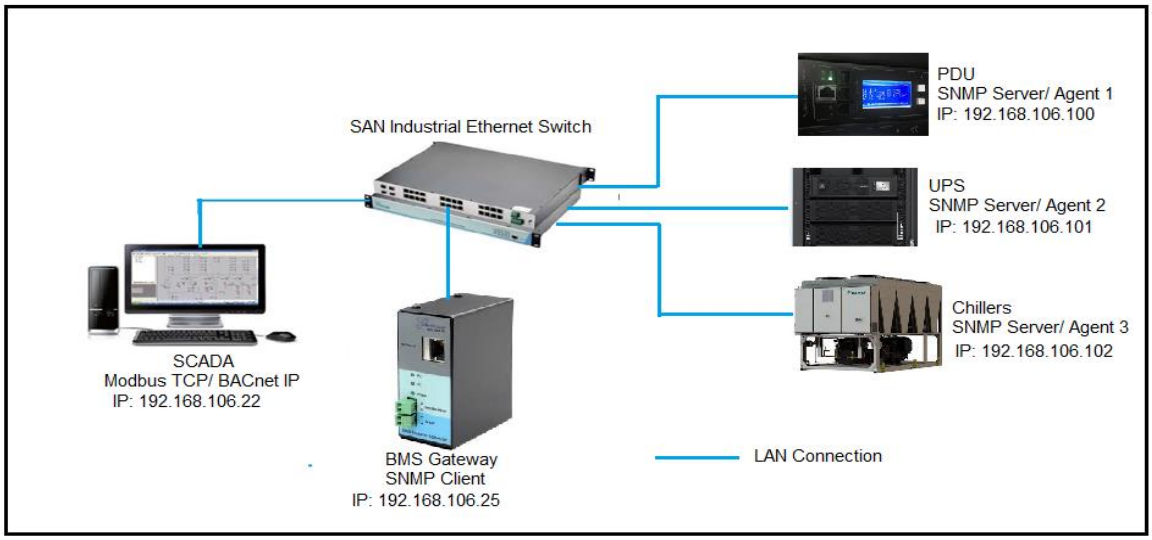


6 SNMP FUNCTIONALITY

Note: For GW BMS 05 unit.

6.1 Using BMS gateway as a SNMP client

Following figure shows connection details when gateway is used as a SNMP client.

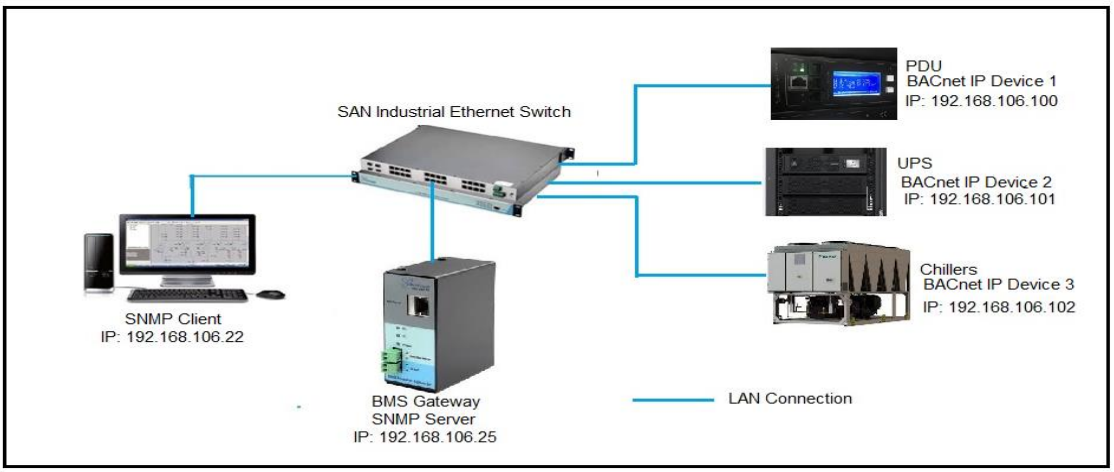


When gateway is used as a SNMP client, it will communicate with SNMP agents/servers connected on the network and provide data to the SCADA on different protocol (e.g. BACnet IP/ Modbus TCP). In order to establish communication with the SNMP servers, configuration file needs to be downloaded in the gateway. This configuration file will contain parameters like IP addresses of SNMP servers and mapping details. Each SNMP server will have it's own MIB file. The object identifiers mentioned in this MIB file will be used for mapping device parameters.

6.2 Using BMS gateway as a SNMP server

Following figure shows connection details when gateway is used as SNMP server.
BACnet IP is shown as other non-SNMP protocol.

Gateway will communicate with BACnet IP devices using node IDs and object IDs of these devices which are configured through CSV file downloaded in the gateway. After configuration of the gateway, MIB file is generated through ruinet utility. This MIB file is loaded into SNMP client software to establish connection with the gateway. After completion of above setup, SNMP client station can communicate to the BACnet IP devices.



7 CONNECTION DETAILS

NAME	DETAILS
24 V AC/ DC (+,- and E)	Connect 24 V AC/DC supply
Ethernet port/BACnet IP	Host / field protocol support (BACnet IP in GW BMS 01, GW BMS 02, GW BMS 03, GW BMS 01-02, GW BMS 02-02R and SNMP in GWBMS05). & Also used for diagnostics purpose.
D+ and D-	Connect D+ and D- of Field devices.

8 LED INDICATIONS

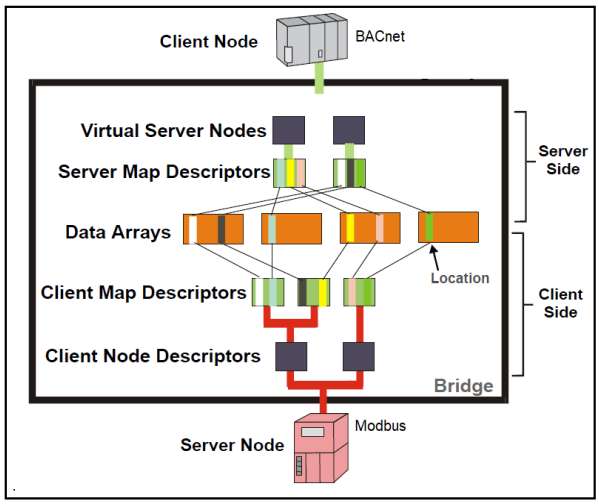
LED name	Meaning
POWER	Unit is powered on.
TX	Channel is transmitting data.
RX	Channel is receiving data.
10/100M	ON-100 Mbps (100Base Tx)
	OFF-10Mbps
LINK/ACT	ON- Network Connected
	OFF- No network connected
	Blinking- Activity on the Ethernet port

9 CSV configuration

9.1 CSV File Overview

Reference to the information submitted by customer the configuration file will be prepared. The default driver configuration file (CONFIG.CSV) for Modbus RTU to BACnet IP combination ordered is loaded into the GW BMS GATEWAY and can be retrieved using the Graphical User Interface. Use this file as a template when creating new configuration files to ensure that the edited file takes the correct form. A detailed explanation of the configuration file is as follows:

9.2 GW BMS Gateway Connection Philosophy



The GW BMS Gateway functions as a bridge between two or more different Nodes (see Figure above). The information is gathered by the Client side of the GW BMS Gateway from the Server Nodes via a Serial Port, Ethernet port or plug-in card. Nodes may use different protocols and even different communication busses. The GW BMS Gateway thus acts as a Client and a Server simultaneously.

For example, consider a Modbus PLC with a set of 10 high alarms in address 00001 to 00010. A Map Descriptor is allocated to fetch Data Objects from Modbus address 00001 length 10 and save this data to a Data Array named PLC1, offset 20. The high alarm for sensor number 5 on PLC1 is thus stored in Data Array PLC1; offset 24 (the fifth location starting at offset 20).

A Client using BACnet IP protocol can be configured to access the GW BMS Gateway and read the Data Array.

9.3 Example of CSV file Modbus RTU to BACnet IP

9.3.1 Steps to prepare CSV File

- Define Common Information
- Define Data Arrays
- Define Client Side connections
- Configure Client side Nodes
- Configure Client side Map descriptor
- Define Server Side Connection
- Configure Server Side Nodes
- Configure Server Side Map descriptor

9.3.2 Details required to configuring CSV for Modbus RTU to BACnet IP conversion

Sr No	Details Required
1	Device ID of devices to be connected on 1st RS485 port.
2	Device ID of devices to be connected on 2nd RS485 port.
3	Modbus register addresses of parameters & name to be mapped.
4	Communication parameters - baud rate, data bits, stop bits and parity.
5	IP address and subnet mask to be set to the gateway.

The CSV file is editable in notepad and excel. Below CSV Example is for Modbus RTU to BACnet IP conversion. For Example 2 VFDs are connected to R1 and R2 port GW BMS Gateway respectively. The Modbus mapping of the VFD is as follows.

Device ID of VFD connected on R1 port: 1
Device ID of VFD connected on R2 port: 2
Baud rate: 9600, Parity: None, Data Bits:8, Stop Bits:1

Parameter Name	Modbus Address
Speed	40001
Frequency	40002
Current	40003
Power	40004
DC Bus Voltage	40005
Output Voltage	40006
Run Hours	40007
Kwh	40008
KVA	40009
KVAR	40010

San Telequip (P) Ltd.,
504/505 Deron Heights, Baner Road, Baner
Pune 411045, Mah, India
Phone : +91 20-27293456, 8390069393, 9764027070
email : info@santelequip.com


```
//=============================================================================
//
//    Customer      : XYZ
//    Location      : Pune
//    Configured By : ABC
//    Date          : 10/05/2021
//
//=============================================================================
//
//    Common Information
//
Bridge
Title              , System_Node_Id ,Network_Number
San Telequip v1.01 ,    1000        ,   5

//=============================================================================
//
//    Data Arrays
//
Data_Arrays
Data_Array_Name , Data_Format , Data_Array_Length
DA_AI_01        , Int16       ,         10
DA_AI_02        , Int16       ,         10

//=============================================================================
//
//    Client Side Connections
//
Connections
Port , Baud , Parity , Data_Bits , Stop_Bits , Protocol
R1   , 9600 ,  None  ,     8     ,      1    , Modbus_RTU
R2   , 9600 ,  None  ,     8     ,      1    , Modbus_RTU

//=============================================================================
//
//    Client Side Nodes
//
Nodes
Node_Name    , Node_ID , Protocol    , Port
O_Device_01  ,     1   , Modbus_RTU ,  R1
O_Device_02  ,     2   , Modbus_RTU ,  R2

//=============================================================================
//
//    Client Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name   , Address , Length , Scan_Interval
CMD_Device_01_1     ,   DA_AI_01      ,       0           , Rdbc     ,O_Device_01  ,   40001 ,   10   , 1.000s

CMD_Device_02_1     ,   DA_AI_02      ,       0           , Rdbc     ,O_Device_02  ,   40001 ,   10   , 1.000s
```

San Telequip (P) Ltd.,
504/505 Deron Heights, Baner Road, Baner
Pune 411045, Mah, India
Phone : +91 20-27293456, 8390069393, 9764027070
email : info@santelequip.com


```
//================================================================================
//
//    Server Side Connections
//

Connections
Adapter , Protocol
N1      , Bacnet_IP

//================================================================================
//
//    Server Side Nodes
//

Nodes
Node_Name    , Node_ID , Protocol  , Srv_Offline_Method
VFD_01       ,  1001   , Bacnet_IP , Always_Respond
VFD_02       ,  1002   , Bacnet_IP , Always_Respond
```

```
//================================================================================
//
//    Server Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name    , Data_Type , Object_ID
Speed_01            ,   DA_AI_01      ,        0          , Server   ,  VFD_01      ,     AV    ,     1
Frequency_01        ,   DA_AI_01      ,        1          , Server   ,  VFD_01      ,     AV    ,     2
Current_01          ,   DA_AI_01      ,        2          , Server   ,  VFD_01      ,     AV    ,     3
Power_01            ,   DA_AI_01      ,        3          , Server   ,  VFD_01      ,     AV    ,     4
Dc bus voltage_01   ,   DA_AI_01      ,        4          , Server   ,  VFD_01      ,     AV    ,     5
Output Voltage_01   ,   DA_AI_01      ,        5          , Server   ,  VFD_01      ,     AV    ,     6
Run hours_01        ,   DA_AI_01      ,        6          , Server   ,  VFD_01      ,     AV    ,     7
Kwh_01              ,   DA_AI_01      ,        7          , Server   ,  VFD_01      ,     AV    ,     8
KVAR_01             ,   DA_AI_01      ,        8          , Server   ,  VFD_01      ,     AV    ,     9
KVAR_01             ,   DA_AI_01      ,        9          , Server   ,  VFD_01      ,     AV    ,     10

Speed_02            ,   DA_AI_02      ,        0          , Server   ,  VFD_02      ,     AV    ,     1
Frequency_02        ,   DA_AI_02      ,        1          , Server   ,  VFD_02      ,     AV    ,     2
Current_02          ,   DA_AI_02      ,        2          , Server   ,  VFD_02      ,     AV    ,     3
Power_02            ,   DA_AI_02      ,        3          , Server   ,  VFD_02      ,     AV    ,     4
Dc bus voltage_02   ,   DA_AI_02      ,        4          , Server   ,  VFD_02      ,     AV    ,     5
Output Voltage_02   ,   DA_AI_02      ,        5          , Server   ,  VFD_02      ,     AV    ,     6
Run hours_02        ,   DA_AI_02      ,        6          , Server   ,  VFD_02      ,     AV    ,     7
Kwh_02              ,   DA_AI_02      ,        7          , Server   ,  VFD_02      ,     AV    ,     8
KVAR_02             ,   DA_AI_02      ,        8          , Server   ,  VFD_02      ,     AV    ,     9
KVAR_02             ,   DA_AI_02      ,        9          , Server   ,  VFD_02      ,     AV    ,     10
```

```
Data_Array_Low_Scale , Data_Array_High_Scale , Node_Low_Scale , Node_High_Scale
          0          ,        100            ,       0       ,     100
          0          ,        100            ,       0       ,     100
          0          ,        100            ,       0       ,     100
          0          ,        100            ,       0       ,     100
          0          ,        100            ,       0       ,     100
          0          ,        100            ,       0       ,     100
          0          ,        100            ,       0       ,     100
          0          ,        100            ,       0       ,     100
          0          ,        100            ,       0       ,     100
          0          ,        100            ,       0       ,     100

          0          ,        100            ,       0       ,     100
          0          ,        100            ,       0       ,     100
          0          ,        100            ,       0       ,     100
          0          ,        100            ,       0       ,     100
          0          ,        100            ,       0       ,     100
          0          ,        100            ,       0       ,     100
          0          ,        100            ,       0       ,     100
          0          ,        100            ,       0       ,     100
          0          ,        100            ,       0       ,     100
          0          ,        100            ,       0       ,     100
```

9.3.3 Description of the CSV file

The file begins with general information. It is contain Customer name, Site location, CSV file Configured by and Date of configuration.

```
//================================================================================
//
//    Customer      : XYZ
//    Location      : Pune
//    Configured By : ABC
//    Date          : 10/05/2021
//
//================================================================================
```

Lines beginning with // are comments and It is do not affect the configuration.

9.3.3.1 Common Information

The Common Information Section allows for the determination of parameters It is not directly related to any of the connections.

```
//================================================================================
//
//    Common Information
//
Bridge
Title              , System_Node_Id ,Network_Number
San Telequip v1.01 ,    1000        ,   5

//================================================================================
```

Title: The Title appears on the top of FS-GUI line Screen. It may be used to indicate the configuration and the relevant Customer/project.

System_Node_ID: The Value of this Parameter is dependent on the context of drivers.
e.g. BACnet - Used as the MAC address.

Network_Number: Specify Network Number to GW BMS GATEWAY where the protocol (BACnet) required. Default value Network_Number is 5. The Network Number should be unique to each BACnet device connected on network. Do not assign same network number to the GW BMS Gateway connected on network.

9.3.3.2 Data Array

Data Arrays are buffers for storage of data to be passed between protocols. It is necessary to declare the data format of each of the Data Arrays to facilitate correct storage of the relevant data.

```
//================================================================================
//
//    Data Arrays
//
Data_Arrays
Data_Array_Name , Data_Format , Data_Array_Length
DA_AI_01        , Int16       ,         10
DA_AI_02        , Int16       ,         10

//================================================================================
```

Data_Array_Name: Its function to Provide name for Data Array. It can support Up to 15 alphanumeric characters.

Data_Format: It is used to provide data format. Each Data Array can only take on one format. Supported data formats are Float, Bit, Byte, Uint16, Uint32, Sint16, Sint32, and UInt64.

Data_Array_Length: It's is depends on Number of Data Objects. It must be larger than the Number of parameters are required to map.

If more than 2 slave device connected do following changes in Data Array.

- Add more Data Arrays. The Number of Data Arrays should match with number of Modbus device connected.
- Each Data Array should have unique name. Do not repeat data array name.

9.3.3.3 Client Side Connections

The Client Side Connections Section contains the parameters that describe the nature of the physical connection to the Server Nodes.

```
//================================================================================
//
//    Client Side Connections
//
Connections
Port , Baud , Parity , Data_Bits , Stop_Bits , Protocol
R1   , 9600 ,  None  ,     8     ,      1    , Modbus_RTU
R2   , 9600 ,  None  ,     8     ,      1    , Modbus_RTU

//================================================================================
```

Port: Specify which port the device is connected to the GW BMS GATEWAY. Serial ports are defined by R1 and R2 and Ethernet Port is specifying by N1.

.

Baud rate: Specify the baud rate of Serial Port. Supported Baud rates are 110-115200. 9600 is default baud rate.

Parity: Specify the parity of serial port. Supported Parity is None, Even and Odd. None is default Parity.

Data_Bits: Specify the Data_bits. Supported data bits are 7,8. Default data bits are 8.

Stop Bits: Specify the stop bits. Supported stop bits are 1, 2. Default stop bits are 1.

Protocol: Specify protocol used on serial port. Supported protocols on serial port are Modbus RTU, BACnet MSTP, Metasys N2. Etc.

9.3.3.4 Client side nodes

The Client Side Nodes Section defines the logical connection parameters for the Server Nodes communicating with the GW BMS GATEWAY.

```
//=========================================================================================
//
//    Client Side Nodes
//
Nodes
Node_Name     , Node_ID , Protocol    , Port
O_Device_01   ,     1   , Modbus_RTU ,  R1
O_Device_02   ,     2   , Modbus_RTU ,  R2

//=========================================================================================
```

Node_Name: Used to provide name for Node. It support Up to 32 alphanumeric characters.

Node_ID: Device ID of the slave devices**.**

Protocol: Specify the slave devices protocol

Port: Specify which port the slave device is connected to the GW BMS GATEWAY.

If more than two Modbus device connected do following changes in client side nodes. Maximum 32 devices are connected on RS485 port of GW BMS GATEWAY.

- Add more nodes. The Number of Client side nodes in CSV should match with number of slave device connected.
- Define each node with unique name.
- Define Device ID. The Device ID of devices connected on same serial port must be unique. The Device ID can repeat on Different serial port (R1 and R2).

9.3.3.5 Client side map descriptor

The Map Descriptor Section contains parameters that describe the address details required to move data between the GW BMS GATEWAY and an external device and the nature of the data transfer.

```
//=========================================================================
//
//    Client Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name   , Address , Length , Scan_Interval
CMD_Device_01_1     ,   DA_AI_01      ,       0           , Rdbc     ,O_Device_01  ,   40001 ,   10   , 1.000s

CMD_Device_02_1     ,   DA_AI_02      ,       0           , Rdbc     ,O_Device_02  ,   40001 ,   10   , 1.000s

//=========================================================================
```

Map_Descriptor_Name: Specify the Name of this Map Descriptor. It support Up to 32 alphanumeric characters.

Data_Array_Name: Name of Data Array where data is to be stored in the GW BMS GATEWAY.

Data_Array_Offset: Specify the Starting location in Data Array.

Function: Function of Client Map Descriptor. Rdbc is used for read and write both operation.

Node_Name: Name of Node to fetch data from. Define same name as configured in Client side nodes.

Address: Starting address of the slave device. Specify the address of the parameters which needs to configure. GW BMS GATEWAY supports Coil, Input status, Input Register, Holding Register Modbus data types.

Length: Length of Map Descriptor. Numbers are address needs to read from starting address. The length is depends on how many number of registers are needs from 40001. As shown in Client side connection length is defined as 10. It means 40001-40010 registers data will be stored in Data Array.

Scan Interval: Specify Rate at which data is polled this is a timing parameter.

The Modbus address is linked to data array as shown below:

Modbus Address	Data Array	Data Array Offset
40001	DA_AI_01	0
40002	DA_AI_01	1
40003	DA_AI_01	2
40004	DA_AI_01	3
40005	DA_AI_01	4
40006	DA_AI_01	5
40007	DA_AI_01	6
40008	DA_AI_01	7
40009	DA_AI_01	8
40010	DA_AI_01	9

If the Modbus registers are not in sequence do configuration as below.
For Example: 40001-40005, 40012, 40021-40023, 40050-40055 register needs to configure.
Notice how data array offset is changing for every query.

```
//=============================================================================
//
//    Client Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name   , Address , Length , Scan_Interval
CMD_Device_01_1     ,   DA_AI_01      ,       0           , Rdbc     ,O_Device_01  ,   40001 ,   5    , 1.000s
CMD_Device_01_2     ,   DA_AI_01      ,       5           , Rdbc     ,O_Device_01  ,   40012 ,   1    , 1.000s
CMD_Device_01_3     ,   DA_AI_01      ,       6           , Rdbc     ,O_Device_01  ,   40021 ,   3    , 1.000s
CMD_Device_01_4     ,   DA_AI_01      ,       9           , Rdbc     ,O_Device_01  ,   40050 ,   5    , 1.000s
```

9.3.3.6 Server Side Connection

The Server Side Sections are functionally the same as their Client Side equivalents, except that Server parameters are being defined.

```
//=============================================================================
//
//     Server Side Connections
//

Connections
Adapter , Protocol
N1      , Bacnet_IP

//=============================================================================
```

Adapter: Adapter definition applies to defining network and GW BMS GATEWAY connections. Ethernet port of GW BMS GATEWAY is defined as N1 in CSV file configuration.

Protocol: The protocol for the network connected to N1 port. GW BMS GATEWAY Support BACnet IP, Modbus TCP, SNMP, Ethernet IP, DNP3 Ethernet protocols on RJ45 port.

9.3.3.7 Server Side Nodes

```
//=============================================================================
//
//     Server Side Nodes
//

Nodes
Node_Name     , Node_ID , Protocol   , Srv_Offline_Method
VFD_01        ,  1001   , Bacnet_IP , Always_Respond
VFD_02        ,  1002   , Bacnet_IP , Always_Respond

//=============================================================================
```

Node_Name: Provide name for Node. This Node Name is Discover on BACnet Client. It supports Up to 31 alphanumeric characters.

Node_ID: Specify a Node ID to BACnet node. The legal value for this field is 0 – 4194303.

Protocol: Specify the protocol.

Srv_Offline_Method: It is used to Controls the system status property of the device object. It keep GW BMS GATEWAY online incase field stop communicating.

To add more nodes configure at BACnet IP side configure the

9.3.3.8 Server Side Map Descriptors

The Server Side Map Descriptor Section contains parameter which needs to discovered on BACnet IP client for under each BACnet Node.

```
//=============================================================================
//
//    Server Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name    , Data_Type , Object_ID
Speed_01            ,   DA_AI_01      ,        0          , Server   ,  VFD_01      ,      AV    ,     1
Frequency_01        ,   DA_AI_01      ,        1          , Server   ,  VFD_01      ,      AV    ,     2
Current_01          ,   DA_AI_01      ,        2          , Server   ,  VFD_01      ,      AV    ,     3
Power_01            ,   DA_AI_01      ,        3          , Server   ,  VFD_01      ,      AV    ,     4
Dc bus voltage_01   ,   DA_AI_01      ,        4          , Server   ,  VFD_01      ,      AV    ,     5
Output Voltage_01   ,   DA_AI_01      ,        5          , Server   ,  VFD_01      ,      AV    ,     6
Run hours_01        ,   DA_AI_01      ,        6          , Server   ,  VFD_01      ,      AV    ,     7
Kwh_01              ,   DA_AI_01      ,        7          , Server   ,  VFD_01      ,      AV    ,     8
KVAR_01             ,   DA_AI_01      ,        8          , Server   ,  VFD_01      ,      AV    ,     9
KVAR_01             ,   DA_AI_01      ,        9          , Server   ,  VFD_01      ,      AV    ,     10

Speed_02            ,   DA_AI_02      ,        0          , Server   ,  VFD_02      ,      AV    ,     1
Frequency_02        ,   DA_AI_02      ,        1          , Server   ,  VFD_02      ,      AV    ,     2
Current_02          ,   DA_AI_02      ,        2          , Server   ,  VFD_02      ,      AV    ,     3
Power_02            ,   DA_AI_02      ,        3          , Server   ,  VFD_02      ,      AV    ,     4
Dc bus voltage_02   ,   DA_AI_02      ,        4          , Server   ,  VFD_02      ,      AV    ,     5
Output Voltage_02   ,   DA_AI_02      ,        5          , Server   ,  VFD_02      ,      AV    ,     6
Run hours_02        ,   DA_AI_02      ,        6          , Server   ,  VFD_02      ,      AV    ,     7
Kwh_02              ,   DA_AI_02      ,        7          , Server   ,  VFD_02      ,      AV    ,     8
KVAR_02             ,   DA_AI_02      ,        8          , Server   ,  VFD_02      ,      AV    ,     9
KVAR_02             ,   DA_AI_02      ,        9          , Server   ,  VFD_02      ,      AV    ,     10
```

```
Data_Array_Low_Scale , Data_Array_High_Scale , Node_Low_Scale , Node_High_Scale
          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100

          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100
          0          ,         100           ,       0       ,      100
```

Map_Descriptor_Name: Specify name to each parameters which needs to discover on BACnet.

Data_Array_Name: Define the Data Array name which is assigned to particular devices as described in section 9.3.5.

Data_Array_Offset: Define a Data array offset to each parameter according to define in section 9.3.5.

Function: Specify a function to BACnet node. GW BMS GATEWAY is configured as a server on BACnet IP.

Node_Name: Specify a Node Name defined in section 9.3.7.

Data_Type: Specify Data_Type for each parameter. Data Types supported by GW BMS GATEWAY are AI, AV, AO, BI, BV, BO, MI, MV, MO, NC and LSP. Refer Section 10.5 for more information.

Object ID: Specify an Object ID to each parameter. GW BMS GATEWAY support 0-4194303 range for object ID.

Units: Specify Units to each parameter.

Data_Array_Low_Scale: It is used to scaling zero in Data Array. A default value of this field is 0.

Data_Array_High_Scale: It is used to Scaling max in Data Array. A default value of this field is 100.

Node_Low_Scale: It is used to Scaling zero in Connected Node. A default value of this field is 100.

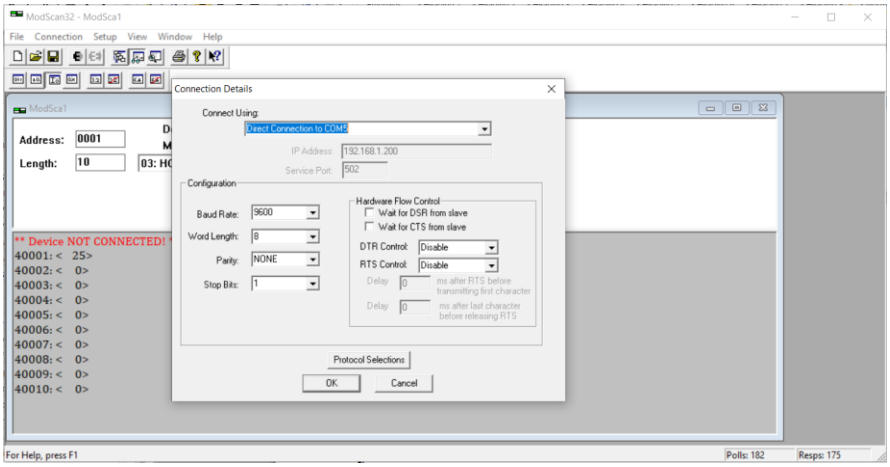
Node_High_Scale: It is used to Scaling max in Connected Node. A default value of this field is 100.

Scaling Formula in GW BMS GATEWAY
BACnet Value = (Node_High_Scale / Data_Array_High_Scale)*Modbus Value.

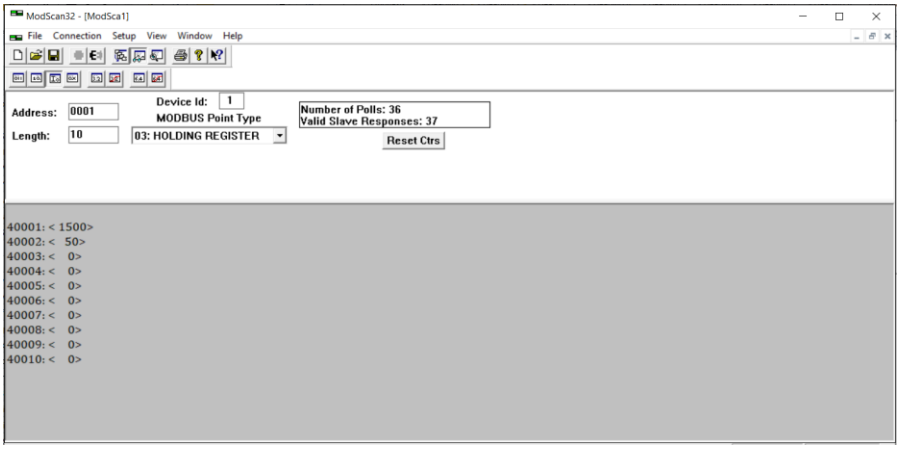
9.3.4 Check Modbus Device Communication on Modscan

To check Modbus RTU device communication Modscan software is useful. USB to RS485 converter is required to check Modbus Communication. Following are the steps to check Modbus Communication.

- Connect RS485 port of USB to RS485 converter to Modbus Slave device RS485 port and USB port to PC/Laptop. Check the COM port detected in PC.
- Open Modscan and Configure COM port with serial settings as shown below screenshot.



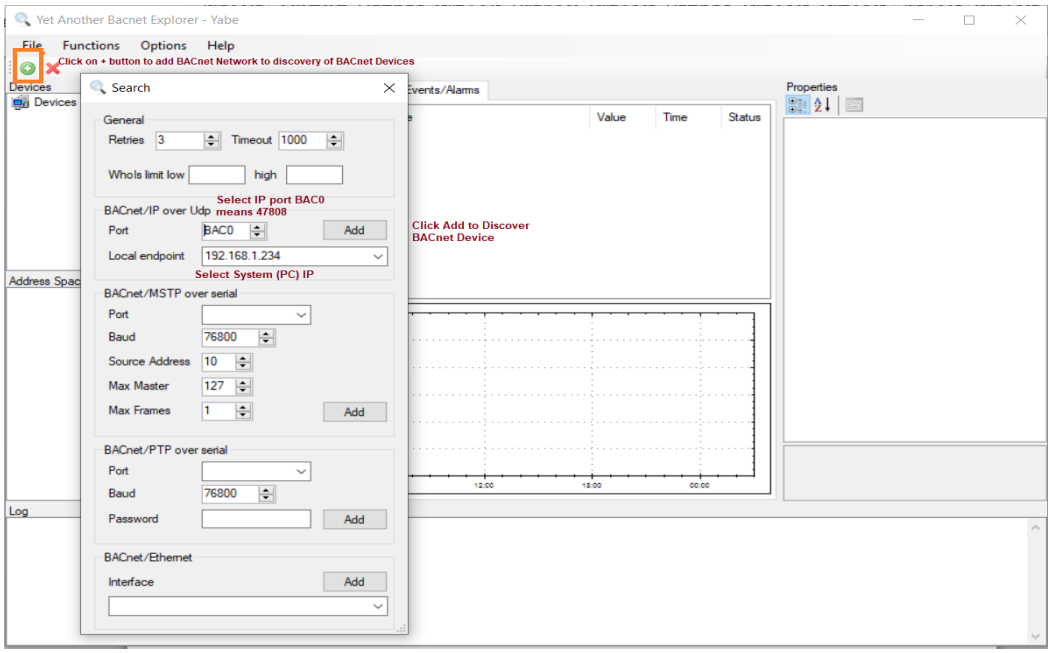
- Click on OK and check communication.
- Select appropriate Modbus address Device ID and check the parameter values.



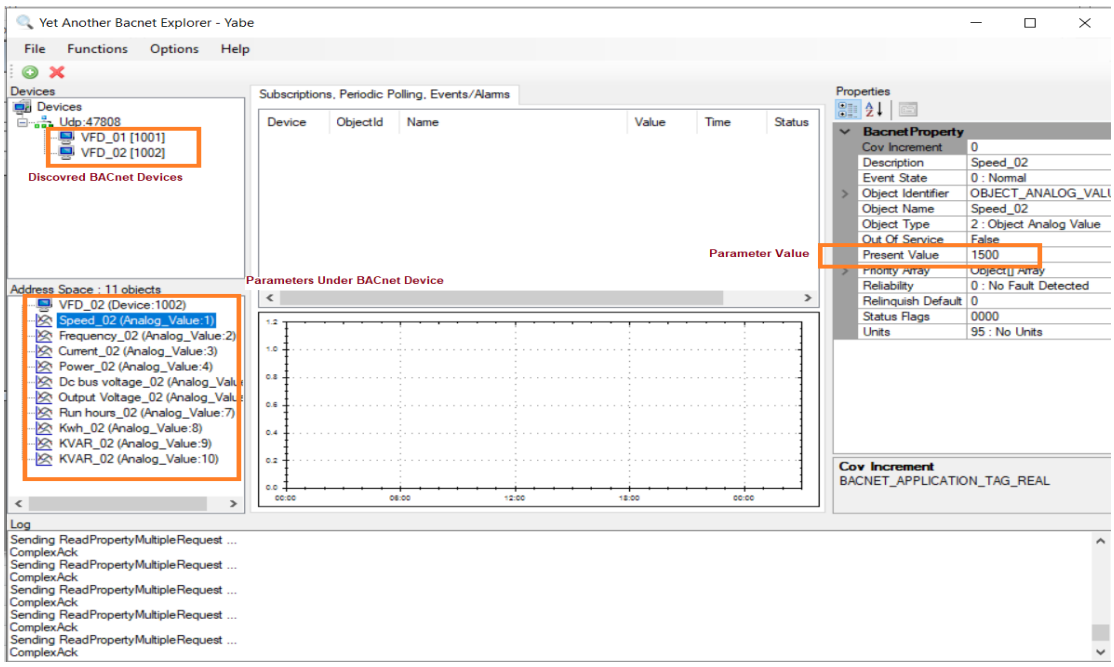
9.3.5 Check GW BMS Gateway BACnet IP communication

YABE software is useful to check BACnet Communication. Following are the steps to check BACnet IP communication.

- Connect straight Ethernet cable in between GW BMS Gateway and PC/Laptop.
- Set up IP address of GW BMS gateway and PC IP address on same subnet.
- Open Yabe Software
- Add BACnet network
- Select IP Port and IP address of Laptop.
- After all settings click on Add as shown in below Screenshots.



- After Click on Add Button the BACnet device will discover automatically.
- Click on Discovered device and check that parameters.
- Check all parameters value.



9.3.6 Troubleshooting Tips

9.3.6.1 If no communication over RS485 (Modbus RTU)

- Check Configured register mapping details of CSV file.
- Check BUS Communication parameters (Baud Rate, Parity, Data Bits & Stop bits)
- Check BUS Connection. All devices should connected in Daisy chain topology
- Check Connection on Port. All the 'A' terminals must be connected together and all the 'B' terminals must be connected together respectively
- Check Type of Cable Used. The cable to be used is shielded twisted pair
- Check Earthing. The cable shield must be properly earthed only in one point
- Check Cable Length not more than1.2Km? Use our RS485 Repeater
- Add Cable termination resistor (120 ohm) to reduce signal reflections

9.3.6.2 Debugging of BACnet Connection

- If duplicate Object Instances are configured in the GW BMS GATEWAY, the second call of the Instance will overwrite the first one. This may cause a BACnet Object to be "lost."
- When using the GW BMS GATEWAY as a BACnet Server, ensure the GW BMS GATEWAY Subnet Mask matches the Subnet Mask of the BACnet Client. Otherwise, communications are very slow and eventually stop altogether.
- The network number assign to each GE BMS must be unique else its result communication failure at BACnet client.

9.4 Example of CSV file BACnet MSTP to BACnet IP

9.4.1 Steps to prepare CSV File

- Define Common Information
- Define Data Arrays
- Define Client Side connections
- Configure Client side Nodes
- Configure Client side Map descriptor
- Define Server Side Connection
- Configure Server Side Nodes
- Configure Server Side Map descriptor

9.4.2 Details required to configuring CSV for BACnet MSTP to BACnet IP conversion

Sr No	Details Required
1	MAC IDs of BACnet MSTP devices to be connected on 1st RS485 port.
2	MAC IDs of BACnet MSTP devices to be connected on 2nd RS485 port.
3	Mapping details (Object IDs) of BACnet MSTP devices and their type [e.g. Analog inputs (AI), Binary inputs (BI)].
4	Communication parameters - baud rate, data bits, stop bits and parity.
5	Parameters name to be mapped.

Below CSV example is for VAV integration. 2 VAVs are connected to R1 and R2 port GW BMS Gateway respectively. The BACnet MSTP details of the VAV are as follows.

MAC ID of VFD connected on R1 port: 1
MAC ID of VFD connected on R2 port: 2
Baud rate: 38400, Parity: None, Data Bits:8, Stop Bits:1

Parameter Name	BACnet MSTP Details
Temp. Set point	AV:1
Room Temp	AI:2
Volume	AI:3
Damper Position	AV:4
ON/OFF Status	BI:5

```
//=============================================================================
//
//    Customer      : XYZ
//    Location      : Pune
//    Configured By : ABC
//    Date      : 10/05/2021"
//
//=============================================================================
//
//    Common Information
//   Date :
Bridge
Title               , System_Node_Id , Network_Number
San Telequip v1.01a ,     105        ,     6

//=============================================================================
//
//    Data Arrays
//
Data_Arrays
Data_Array_Name , Data_Format , Data_Array_Length
DA_AV_01        ,     Float   ,     10
DA_AV_02        ,     Float   ,     10

//=============================================================================
//
//    Client Side Connections
//
Connections
Port ,   Baud  , Parity  , Data_Bits , Stop_Bits , Protocol    , Connection_Type
R1   ,  38400  ,  None   ,     8     ,     1     ,BACnet_MSTP  , MSTP_Master_Mode
R2   ,  38400  ,  None   ,     8     ,     1     ,BACnet_MSTP  , MSTP_Master_Mode

//=============================================================================
//
//    Client Side Nodes
Nodes
Node_Name ,MAC_Address    , Protocol    , Port, Retry_Interval , APDU_Timeout , APDU_Retries
T_VAV_01  ,1,BACnet_MSTP  ,     R1      , 0.100s               , 5.000s       ,    1
T_VAV_02  ,2,BACnet_MSTP  ,     R2      , 0.100s               , 5.000s       ,    1
```

```
//=============================================================================
//    Client Side Map Descriptors
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset ,  Function , Node_Name , Data_Type , Object_ID , Property      , Scan_Interval
CMD_VAV_01_1        ,DA_AV_01         ,       0           ,   Rdbc     , T_VAV_01  ,    AV     ,     1     , Present_Value , 1.000s
CMD_VAV_01_2        ,DA_AV_01         ,       1           ,   Rdbc     , T_VAV_01  ,    AI     ,     2     , Present_Value , 1.000s
CMD_VAV_01_3        ,DA_AV_01         ,       2           ,   Rdbc     , T_VAV_01  ,    AI     ,     3     , Present_Value , 1.000s
CMD_VAV_01_4        ,DA_AV_01         ,       3           ,   Rdbc     , T_VAV_01  ,    AV     ,     4     , Present_Value , 1.000s
CMD_VAV_01_5        ,DA_AV_01         ,       4           ,   Rdbc     , T_VAV_01  ,    BI     ,     5     , Present_Value , 1.000s

CMD_VAV_02_1        ,DA_AV_02         ,       0           ,   Rdbc     , T_VAV_02  ,    AV     ,     1     , Present_Value , 1.000s
CMD_VAV_02_2        ,DA_AV_02         ,       1           ,   Rdbc     , T_VAV_02  ,    AI     ,     2     , Present_Value , 1.000s
CMD_VAV_02_3        ,DA_AV_02         ,       2           ,   Rdbc     , T_VAV_02  ,    AI     ,     3     , Present_Value , 1.000s
CMD_VAV_02_4        ,DA_AV_02         ,       3           ,   Rdbc     , T_VAV_02  ,    AV     ,     4     , Present_Value , 1.000s
CMD_VAV_02_5        ,DA_AV_02         ,       4           ,   Rdbc     , T_VAV_02  ,    BI     ,     5     , Present_Value , 1.000s

//=============================================================================
//
//    Server Side Connections
//
Connections
Adapter , Protocol
N1      , Bacnet_IP

//=============================================================================
//    Server Side Nodes
//
Nodes
Node_Name    , Node_ID , Protocol  , Srv_Offline_Method
VAV_01       ,  1001   , Bacnet_IP , Always_Respond
VAV_02       ,  1002   , Bacnet_IP , Always_Respond
```

```
//=============================================================================================
//
//     Server Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Data_Type , Object_ID , Units     ,
Temp_Setpoint       ,    DA_AV_01     ,       0           ,  Server  ,  VAV_01   ,    AV     ,     1     , Deg-C     ,
Room_Temp.          ,    DA_AV_01     ,       1           ,  Server  ,  VAV_01   ,    AI     ,     2     , Deg-C     ,
Volume              ,    DA_AV_01     ,       2           ,  Server  ,  VAV_01   ,    AI     ,     3     , No-Units  ,
Damper Position     ,    DA_AV_01     ,       3           ,  Server  ,  VAV_01   ,    AV     ,     4     , percent   ,
ON/OFF Status       ,    DA_AV_01     ,       4           ,  Server  ,  VAV_01   ,    BI     ,     5     , No-Units  ,

Temp_Setpoint       ,    DA_AV_02     ,       0           ,  Server  ,  VAV_02   ,    AV     ,     1     , Deg-C     ,
Room_Temp.          ,    DA_AV_02     ,       1           ,  Server  ,  VAV_02   ,    AI     ,     2     , Deg-C     ,
Volume              ,    DA_AV_02     ,       2           ,  Server  ,  VAV_02   ,    AI     ,     3     , No-Units  ,
Damper Position     ,    DA_AV_02     ,       3           ,  Server  ,  VAV_02   ,    AV     ,     4     , percent   ,
ON/OFF Status       ,    DA AV 02     ,       4           ,  Server  ,  VAV 02   ,    BI     ,     5     , No-Units  ,
```

```
Data_Array_Low_Scale , Data_Array_High_Scale , Node_Low_Scale , Node_High_Scale
        0            ,         100           ,        0       ,    100
        0            ,         100           ,        0       ,    100
        0            ,         100           ,        0       ,    100
        0            ,         100           ,        0       ,    100
        0            ,         100           ,        0       ,    100

        0            ,         100           ,        0       ,    100
        0            ,         100           ,        0       ,    100
        0            ,         100           ,        0       ,    100
        0            ,         100           ,        0       ,    100
        0            ,         100           ,        0       ,    100
```

9.4.3 Description of the CSV file

The file begins with general information. It is contain Customer name, Site location, CSV file Configured by and Date of configuration.

9.4.3.1 Common Information

The Common Information Section allows for the determination of parameters it is not directly related to any of the connections.

```
//================================================================================
//
//    Common Information
//   Date :
Bridge
Title               , System_Node_Id , Network_Number
San Telequip v1.01a ,      105       ,      6

//================================================================================
```

Title: The Title appears on the top of FS-GUI line Screen. It may be used to indicate the configuration and the relevant Customer/project.

System_Node_ID: The Value of this Parameter is dependent on the context of drivers. For.BACnet MSTP - Used as the MAC address. This value is always less than 127.

Network_Number: Specify Network Number to GW BMS GATEWAY where the protocol (BACnet) required. Default value Network_Number is 5. The Network Number should be unique to each BACnet device connected on network.

9.4.3.2 Data Array

Data Arrays are buffers for storage of data to be passed between protocols. It is necessary to declare the data format of each of the Data Arrays to facilitate correct storage of the relevant data.

```
//================================================================================
//
//    Data Arrays
//
Data_Arrays
Data_Array_Name , Data_Format , Data_Array_Length
DA_AV_01        ,     Float   ,     10
DA_AV_02        ,     Float   ,     10

//================================================================================
```

Data_Array_Name: Its function to Provide name for Data Array. It can support Up to 15 alphanumeric characters.

Data_Format: It is used to provide data format. Each Data Array can only take on one format. Supported data formats are Float, Bit, Uint16, Uint32, Sint16, Sint32, Byte.

Data_Array_Length: It's is depends on Number of Data Objects. It must be larger than the Number of parameters are required to map.

9.4.3.3 Client Side Connections

The Client Side Connections Section contains the parameters that describe the nature of the physical connection to the Server Nodes.

```
//================================================================================
//
//    Client Side Connections
//
Connections
Port ,   Baud  , Parity  , Data_Bits , Stop_Bits , Protocol    , Connection_Type
R1   ,  38400  ,  None   ,     8     ,     1     ,BACnet_MSTP  , MSTP_Master_Mode
R2   ,  38400  ,  None   ,     8     ,     1     ,BACnet_MSTP  , MSTP_Master_Mode

//================================================================================
```

Port: Specify which port the device is connected to the GW BMS GATEWAY. Serial ports are defined by R1 and R2 and Ethernet Port is specifying by N1.

Baud rate: Specify the baud rate of Serial Port. Supported Baud rates are 9600, 19200, 38400, 76800. 9600 is default baud rate.
Parity: Specify the parity of serial port. Supported Parity is None, Even and Odd. None is default Parity.

Data_Bits: Specify the Data_Bits. Supported data bits are 7, 8. Default data bits are 8.

Stop Bits: Specify the stop bits. Supported stop bits are 1, 2. Default stop bits are 1.

Protocol: Specify protocol used on serial port. Supported protocols on serial port are Modbus RTU, BACnet MSTP, and Metasys N2. Etc.

Connection_Type: Specify if the connection must be in Master_Mode or in Slave_Mode

9.4.3.4 Client side nodes

The Client Side Nodes Section defines the logical connection parameters for the Server Nodes communicating with the GW BMS GATEWAY.

```
//================================================================================
//
//    Client Side Nodes
Nodes
Node_Name ,MAC_Address    , Protocol    , Port , Retry_Interval , APDU_Timeout , APDU_Retries
T_VAV_01  ,     1         ,BACnet_MSTP  ,  R1  , 0.100s         , 5.000s       ,    1
T_VAV_02  ,     2         ,BACnet_MSTP  ,  R2  , 0.100s         , 5.000s       ,    1

//================================================================================
```

Node_Name: This Used to provide name for Node. It support Up to 32 alphanumeric characters.

MAC_Address / Node_ID: Device ID/ MAC_ID of the slave devices**.**

Protocol: Specify the slave devices protocol

Port: Specify which port the slave device is connected to the GW BMS GATEWAY.

Retry_Interval: The amount of time in seconds that the GW BMS Gateway should wait before retrying a poll after a timeout has occurred.

APDU_Timeout: The time in milliseconds between retransmissions of an APDU requiring acknowledgement for which no acknowledgment has been received.

APDU_Retries: The maximum number of times that an APDU shall be retransmitted.

9.4.3.5 Client side map descriptor

The Map Descriptor Section contains parameters that describe the address details required to move data between the GW BMS GATEWAY and an external device and the nature of the data transfer.

```
//================================================================================
//
//    Client Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset ,  Function , Node_Name , Data_Type , Object_ID , Property      , Scan_Interval
CMD_VAV_01_1        ,DA_AV_01         ,       0           ,   Rdbc     , T_VAV_01  ,    AV     ,     1     , Present_Value , 1.000s
CMD_VAV_01_2        ,DA_AV_01         ,       1           ,   Rdbc     , T_VAV_01  ,    AI     ,     2     , Present_Value , 1.000s
CMD_VAV_01_3        ,DA_AV_01         ,       2           ,   Rdbc     , T_VAV_01  ,    AI     ,     3     , Present_Value , 1.000s
CMD_VAV_01_4        ,DA_AV_01         ,       3           ,   Rdbc     , T_VAV_01  ,    AV     ,     4     , Present_Value , 1.000s
CMD_VAV_01_5        ,DA_AV_01         ,       4           ,   Rdbc     , T_VAV_01  ,    BI     ,     5     , Present_Value , 1.000s

CMD_VAV_02_1        ,DA_AV_02         ,       0           ,   Rdbc     , T_VAV_02  ,    AV     ,     1     , Present_Value , 1.000s
CMD_VAV_02_2        ,DA_AV_02         ,       1           ,   Rdbc     , T_VAV_02  ,    AI     ,     2     , Present_Value , 1.000s
CMD_VAV_02_3        ,DA_AV_02         ,       2           ,   Rdbc     , T_VAV_02  ,    AI     ,     3     , Present_Value , 1.000s
CMD_VAV_02_4        ,DA_AV_02         ,       3           ,   Rdbc     , T_VAV_02  ,    AV     ,     4     , Present_Value , 1.000s
CMD_VAV_02_5        ,DA_AV_02         ,       4           ,   Rdbc     , T_VAV_02  ,    BI     ,     5     , Present_Value , 1.000s

//================================================================================
```

Map_Descriptor_Name: Specify the Name of this Map Descriptor. It support Up to 32 alphanumeric characters.

Data_Array_Name: Name of Data Array where data is to be stored in the GW BMS GATEWAY.

Data_Array_Offset: Specify the Starting location in Data Array.

Function: Function of Client Map Descriptor. Rdbc is used for read and write both operations.

Node_Name: Name of Node to fetch data from.

Data_Types: Specify the data type of parameter. GW BMS Gateway supports AI, AO, AV, BI, BO, BV, MI, MO, MV, LSP and custom Data types.

Object_ID: Specify the object ID of parameter. GW BMS Gateway Supports 0-4194303 range for object types.

Property: Specify The BACnet property to be read. Present_Value property contains the present value of the Input / Output / Value.

Scan_Interval: Specify Rate at which data is polled this is a timing parameter.

9.4.3.6 Server Side Connection

The Server Side Sections are functionally the same as their Client Side equivalents, except that Server parameters are being defined. Refer Section 9.3.3.6 for detailed information.

```
//=============================================================================
//
//    Server Side Connections
//

Connections
Adapter , Protocol
N1      , Bacnet_IP

//=============================================================================
```

9.4.3.7 Server Side Nodes

Refer Section 9.3.3.7 for detailed information.

```
//=============================================================================
//
//    Server Side Nodes
//

Nodes
Node_Name     , Node_ID , Protocol   , Srv_Offline_Method
VAV_01        ,  1001   , Bacnet_IP , Always_Respond
VAV_02        ,  1002   , Bacnet_IP , Always_Respond

//=============================================================================
```

9.4.3.8 Server Side Map Descriptors

Refer Section 9.3.3.8 for detailed information.

```
//=============================================================================
//
//    Server Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Data_Type , Object_ID , Units    ,
Temp_Setpoint       ,   DA_AV_01      ,      0            ,  Server  ,  VAV_01   ,    AV     ,     1      , Deg-C    ,
Room_Temp.          ,   DA_AV_01      ,      1            ,  Server  ,  VAV_01   ,    AI     ,     2      , Deg-C    ,
Volume              ,   DA_AV_01      ,      2            ,  Server  ,  VAV_01   ,    AI     ,     3      , No-Units ,
Damper Position     ,   DA_AV_01      ,      3            ,  Server  ,  VAV_01   ,    AV     ,     4      , percent  ,
ON/OFF Status       ,   DA_AV_01      ,      4            ,  Server  ,  VAV_01   ,    BI     ,     5      , No-Units ,

Temp_Setpoint       ,   DA_AV_02      ,      0            ,  Server  ,  VAV_02   ,    AV     ,     1      , Deg-C    ,
Room_Temp.          ,   DA_AV_02      ,      1            ,  Server  ,  VAV_02   ,    AI     ,     2      , Deg-C    ,
Volume              ,   DA_AV_02      ,      2            ,  Server  ,  VAV_02   ,    AI     ,     3      , No-Units ,
Damper Position     ,   DA_AV_02      ,      3            ,  Server  ,  VAV_02   ,    AV     ,     4      , percent  ,
ON/OFF Status       ,   DA AV 02      ,      4            ,  Server  ,  VAV 02   ,    BI     ,     5      , No-Units ,
```

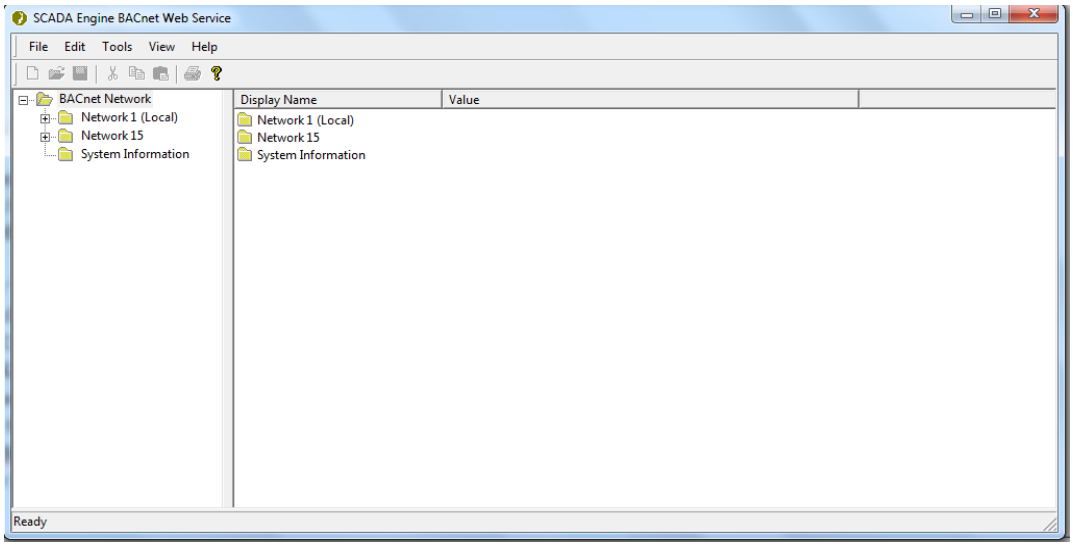
```
Data_Array_Low_Scale , Data_Array_High_Scale , Node_Low_Scale , Node_High_Scale
        0            ,         100           ,        0      ,    100
        0            ,         100           ,        0      ,    100
        0            ,         100           ,        0      ,    100
        0            ,         100           ,        0      ,    100
        0            ,         100           ,        0      ,    100

        0            ,         100           ,        0      ,    100
        0            ,         100           ,        0      ,    100
        0            ,         100           ,        0      ,    100
        0            ,         100           ,        0      ,    100
        0            ,         100           ,        0      ,    100
```

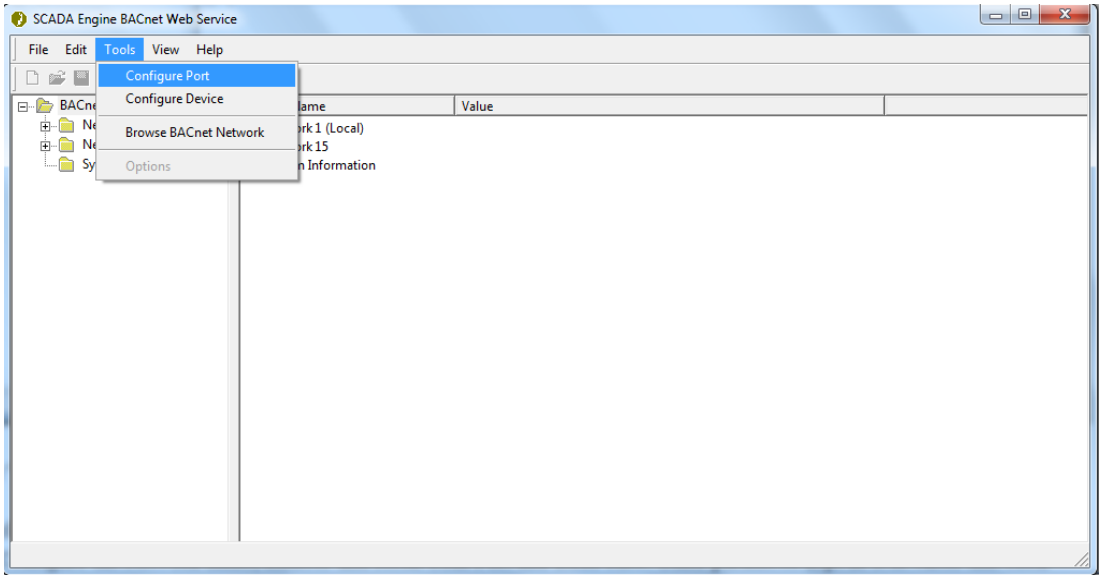
9.4.4 Check MSTP Device communication on MSTP tool (BACnet Web Service)

To check BACnet device communication on BACnet Web Service software is useful. USB to RS485 converter is required to check MSTP Communication.To check MSTP device on BACnet web service follow the below procedure:

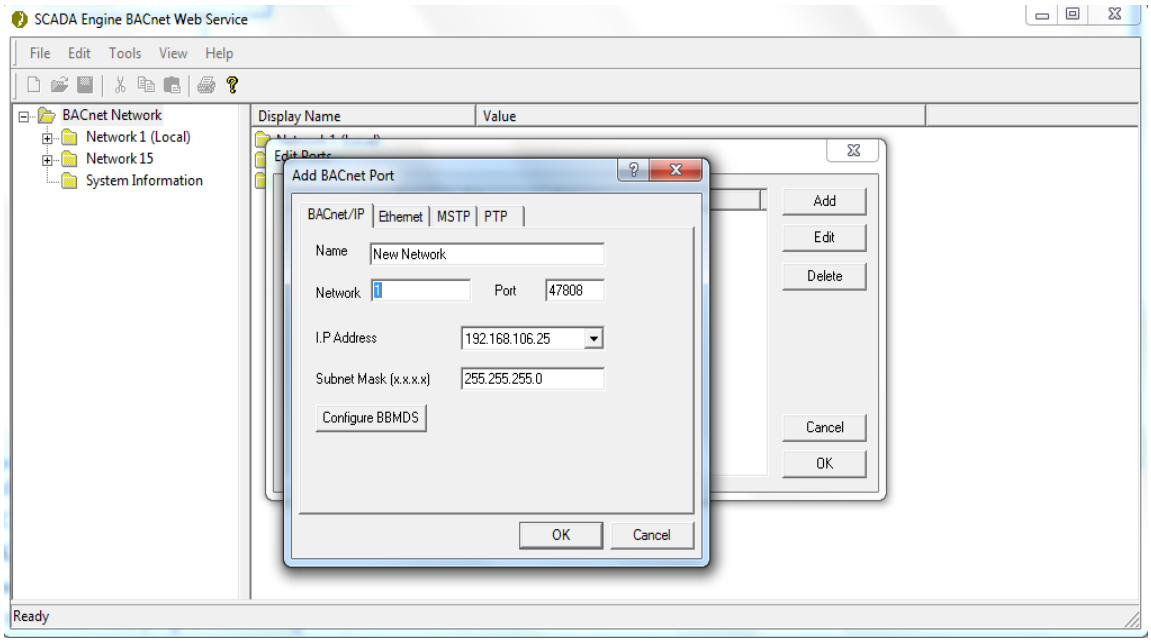
1. Connect the MSTP device to PC through USB to RS 485 Converter.
2. Check the com port in Device Manager.
3. Open the BACnet web service tool. Wait for 5sec and click on ‘OK’ button. Following window will be open.



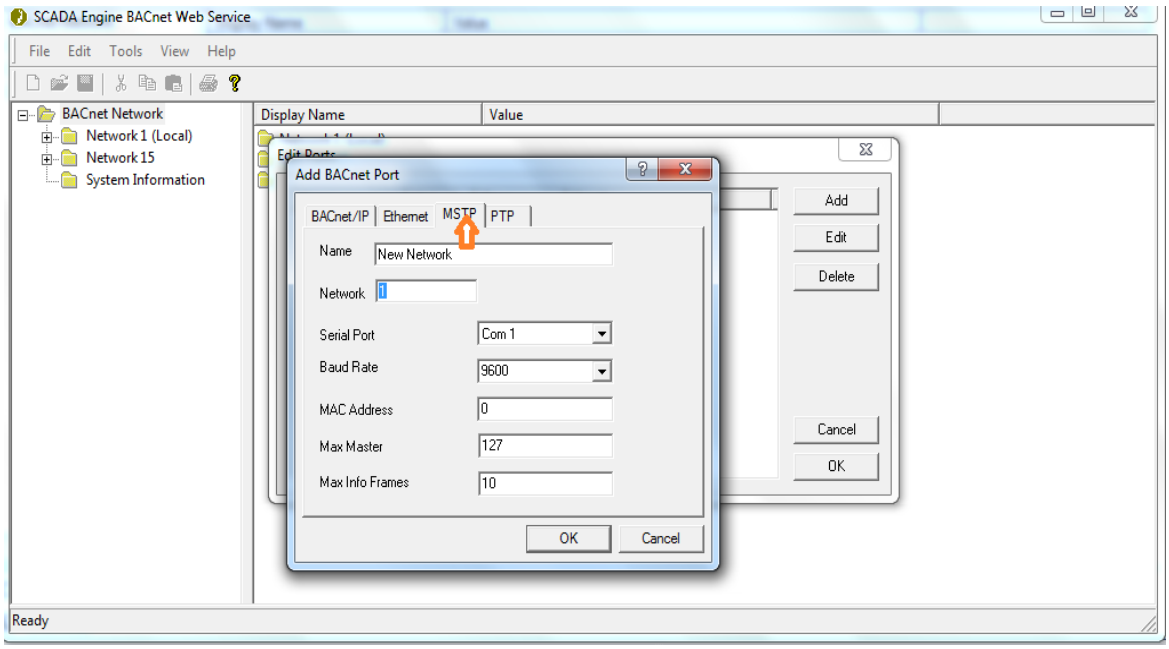
4. Select ‘Tool’ option from Menu bar. Then click on Configure Port. Delete previous setting.



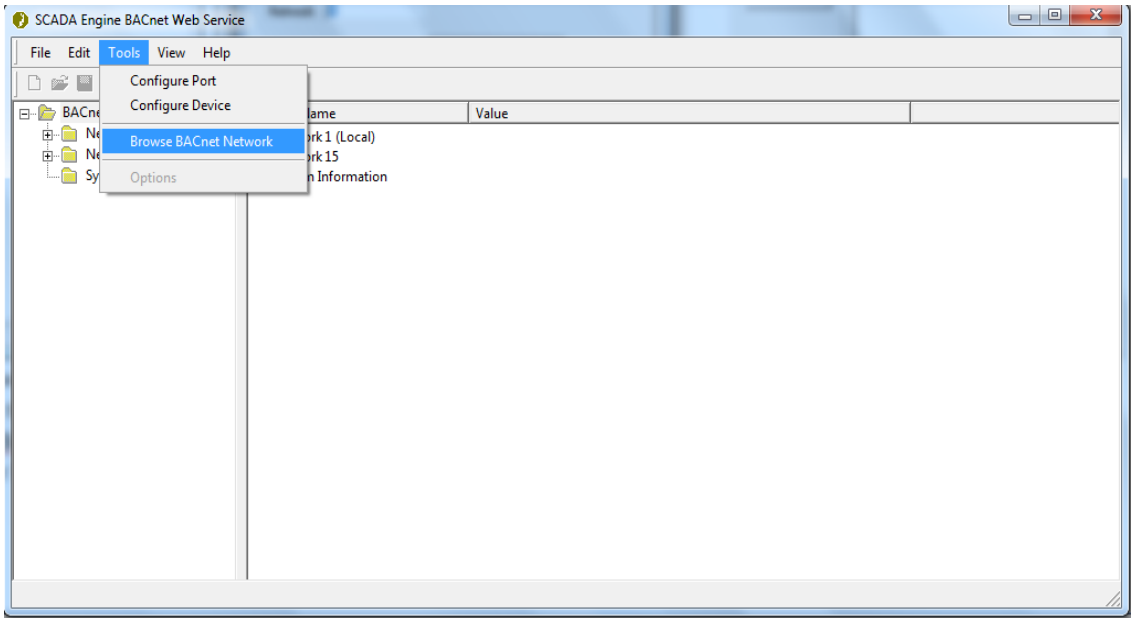
5. Then click on ‘ADD’ option. Select the ‘MSTP’ menu.



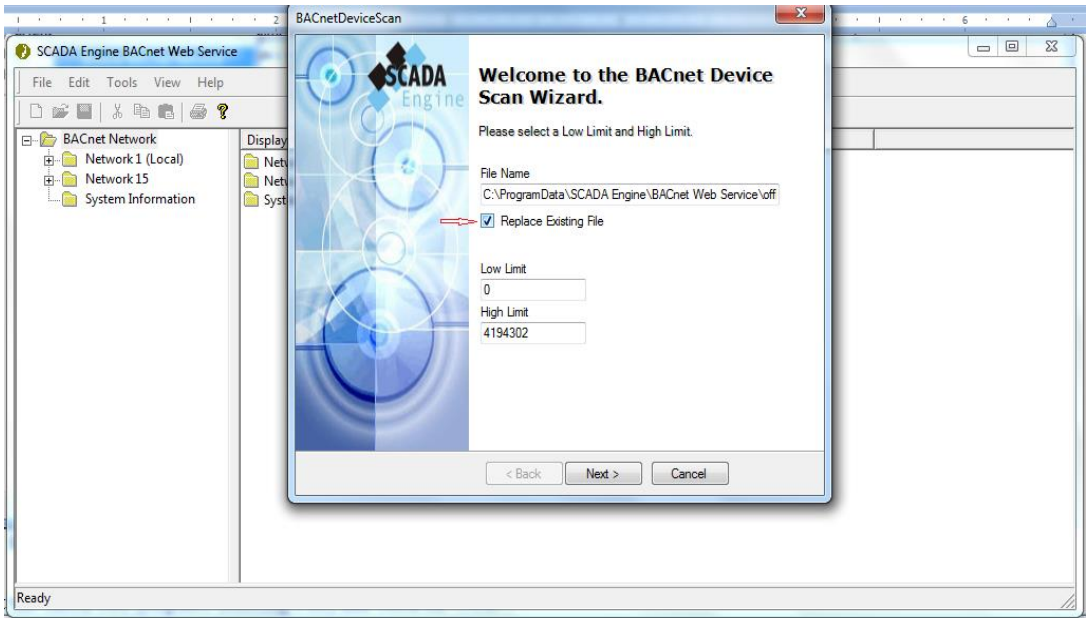
6. Then Configure the Network name, number and serial Com Port and Baud Rate. Other parameter kept as it is. After that click on ‘OK’ button.



7. Then select ‘Tool’ from menu bar and then click on ‘Brows BACnet Network’.



8. Then following window will be open. Select the check box (Replace Existing File) and click on ‘Next’.



9. Then BACnet web service tool start to scanning the devices.
10. After Scanning completed click on Next Button.
11. Then Click on Network. From BACnet network tree.
12. Then all VAV parameters will be displayed below the Network option in Tree.

9.4.5 Troubleshooting Tips

9.4.5.1 If no communication over RS485 (BACnet MSTP)

- Check and verify Configured MC ID, Data Type, Object ID details of CSV file.
- Check BUS Communication parameters (Baud Rate, Parity, Data Bits & Stop bits)
- Check BUS Connection. All devices should connected in Daisy chain topology
- Check Connection on Port. All the 'A' terminals must be connected together and all the 'B' terminals must be connected together respectively
- Check Type of Cable Used. The cable to be used is shielded twisted pair
- Check Earthing. The cable shield must be properly earthed only in one point
- Check Cable Length not more than1.2Km? Use our RS485 Repeater
- Add Cable termination resistor (120 ohm) to reduce signal reflections

9.5 Example of CSV file LonWorks to BACnet IP

9.5.1 Steps to prepare CSV File

- Define Common Information
- Define Data Arrays
- Define Client Side connections
- Configure Client side Nodes
- Configure Client side Map descriptor
- Define Server Side Connection
- Configure Server Side Nodes
- Configure Server Side Map descriptor

9.5.2 Details required to configuring CSV for LonWorks to BACnet IP conversion

Sr No	Details Required
1	XIF file of LonWorks device.
2	SNVT parameter names to be mapped of LonWorks device.
3	Total number of LonWork's devices to be mapped.
4	Neuron ID of LonWorks device.
5	User manual and installation documents of LonWorks device.

Below CSV example is for VRF integration. 1 VRF connected to LON port of GW BMS Gateway. The LonWorks details of the VRF are as follows.

Neuron ID of VRF: 43304920300

Parameter Name	SNVT Type	SNVT Index
Minimum_Send_Time	SNVT_time_sec	0
Send_Heartbeat	SNVT_time_sec	1
Send_On_Delta_Tempearature	SNVT_temp_p	2
Delay_Start_Up_Time	SNVT_time_sec	3
Range_Maximum	SNVT_count	4

San Telequip (P) Ltd.,
504/505 Deron Heights, Baner Road, Baner
Pune 411045, Mah, India
Phone : +91 20-27293456, 8390069393, 9764027070
email : info@santelequip.com


```
//=========================================================================
//    Customer       : XYZ
//    Location       : Pune
//    Configured By : ABC
//    Date           : 26/05/2021
//
//=========================================================================
//
//    Common Information
//
Bridge
System_Node_Id , Title
127            ,"""":D01:S01:San Telequip"""

//=========================================================================
//
//    Data Arrays
//
Data_Arrays
Data_Array_Name , Data_Format , Data_Array_Length
DA_AI_01        , Float       ,    26

//=========================================================================
//
//    Client Side Connections
//
Connections
Adapter  , Protocol
Lonworks , Lonworks

//=========================================================================
//
//    Client Side Nodes
//
Nodes
Node_Name , Node_ID , Protocol , Adapter  , Neuron_Id
Lon_01    ,    1    , Lonworks , Lonworks ,43304920300

//=========================================================================
//
//    Client Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Lon_Function , Function , Node_Name , SNVT_Index , SNVT_Type        , Scan_Interval
nviOnOff_01         , DA_AI_01        ,      0            , NMFETCHC     , Rdbc     , Lon_01    ,     0      , SNVT_time_sec    , 1.000s
nviOnOff_02         , DA_AI_01        ,      1            , NMFETCHC     , Rdbc     , Lon_01    ,     1      , SNVT_time_sec    , 1.000s
nviOnOff_03         , DA_AI_01        ,      2            , NMFETCHC     , Rdbc     , Lon_01    ,     2      , SNVT_temp_p      , 1.000s
nviOnOff_04         , DA_AI_01        ,      3            , NMFETCHC     , Rdbc     , Lon_01    ,     3      , SNVT_time_sec    , 1.000s
nviOnOff_05         , DA_AI_01        ,      4            , NMFETCHC     , Rdbc     , Lon_01    ,     4      , SNVT_count       , 1.000s
```

```
//=============================================================================
//
//    Server Side Connections
//

Connections
Adapter , Protocol
N1      , Bacnet_IP

//=============================================================================
//
//    Server Side Nodes
//

Nodes
Node_Name  , Node_ID , Protocol  , Srv_Offline_Method
B20_VRF_01 , 1001    , Bacnet_IP , Always_Respond
```

```
//=========================================================================
//
//    Server Side Map Descriptors
//
//Sensor Outputs
Map_Descriptors
Map_Descriptor_Name        , Data_Array_Name , Data_Array_Offset , Function , Node_Name   , Data_Type ,Object_ID , Units    ,
Minimum_Send_Time          , DA_AI_01        ,      0            , Server   , B20_VRF_01 ,   AI      ,  1       , hours    ,
Send_Heartbeat             , DA_AI_01        ,      1            , Server   , B20_VRF_01 ,   AI      ,  2       , No-Units ,
Send_On_Delta_Tempearature , DA_AI_01        ,      2            , Server   , B20_VRF_01 ,   AI      ,  3       , Deg-C    ,
Delay_Start_Up_Time        , DA_AI_01        ,      3            , Server   , B20_VRF_01 ,   AI      ,  4       , No-Units ,
Range_Maximum              , DA_AI_01        ,      4            , Server   , B20_VRF_01 ,   AI      ,  5       , No-Units ,
```

```
Data_Array_Low_Scale , Data_Array_High_Scale , Node_Low_Scale , Node_High_Scale
        0            ,        100            ,        0       ,   100
        0            ,        100            ,        0       ,   100
        0            ,        100            ,        0       ,   100
        0            ,        100            ,        0       ,   100
        0            ,        100            ,        0       ,   100
```

9.5.3 Description of the CSV file

The file begins with general information. It is contain Customer name, Site location, CSV file Configured by and Date of configuration.

9.5.3.1 Common Information

Refer Section 9.4.3.1 for detailed information.

```
//================================================================================
//
//    Common Information
//
Bridge
System_Node_Id , Title
127            ,"""D01:S01:San Telequip"""

//================================================================================
```

9.5.3.2 Data Array

Refer Section 9.4.3.2 for detailed information.

```
//================================================================================
//
//    Data Arrays
//
Data_Arrays
Data_Array_Name , Data_Format , Data_Array_Length
DA_AI_01        , Float       ,    26

//================================================================================
```

9.5.3.3 Client Side Connections

The Client Side Connections Section contains the parameters that describe the nature of the physical connection to the Server Nodes.

```
//================================================================================
//
//    Client Side Connections
//
Connections
Adapter  , Protocol
Lonworks , Lonworks

//================================================================================
```

Adapter: Define the GW BMS Gateway LonWorks port.

Protocol: Specify protocol used.

9.5.3.4 Client Side nodes

The Client Side Nodes Section defines the logical connection parameters for the Server Nodes communicating with the GW BMS GATEWAY.

Node Name: This Used to provide name for Node. It support Up to 32 alphanumeric characters.

```
//================================================================================
//
//    Client Side Nodes
//
Nodes
Node_Name , Node_ID , Protocol , Adapter  , Neuron_Id
Lon_01    ,    1    , Lonworks , Lonworks ,43304920300

//================================================================================
```

Node_ID: Define: Specify Node ID to configured Lon Node.

Protocol: Specify the protocol.

Adapter: Specify the adapter

Neuron ID: Specify the Neuron ID of LonWorks Device.

9.5.3.5 Client side map descriptor

The Map Descriptor Section contains parameters that describe the address details required to move data between the GW BMS GATEWAY and an external device and the nature of the data transfer.

```
//================================================================================
//
//    Client Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Lon_Function , Function , Node_Name , SNVT_Index , SNVT_Type       , Scan_Interval
nviOnOff_01         , DA_AI_01        ,      0            , NMFETCHC     , Rdbc     , Lon_01    ,     0      , SNVT_time_sec   , 1.000s
nviOnOff_02         , DA_AI_01        ,      1            , NMFETCHC     , Rdbc     , Lon_01    ,     1      , SNVT_time_sec   , 1.000s
nviOnOff_03         , DA_AI_01        ,      2            , NMFETCHC     , Rdbc     , Lon_01    ,     2      , SNVT_temp_p     , 1.000s
nviOnOff_04         . DA_AI_01        .      3            . NMFETCHC     . Rdbc     . Lon_01    .     3      . SNVT_time_sec   . 1.000s
//================================================================================
```

Map_Descriptor_Name: Specify the Name of this Map Descriptor. It support Up to 32 alphanumeric characters.

Data_Array_Name: Name of Data Array where data is to be stored in the GW BMS GATEWAY.

Data_Array_Offset: Specify the Starting location in Data Array.

Lon_Function: Specify the LonWorks function. NMFETCHC is the preferred Lon Function for most applications. NMFETCHC function is same as RDBC function in Modbus RTU (Section 9.3.3.5) configuration.

Function: Specify the function. RDBC is used for read and write operation.

Node_Name: Name of Node to fetch data from.

SNVT_Index: Specify the SNVT_Index to relevant map descriptor.

SNVT_Type: Specify the SNVT type to relevant map descriptor**.** GW BMS Gateway support following SNVT types.

San Telequip (P) Ltd.,
504/505 Deron Heights, Baner Road, Baner
Pune 411045, Mah, India
Phone : +91 20-27293456, 8390069393, 9764027070
email : info@santelequip.com

SNVT Type Number	SNVT Type Name	SNVT Type Number	SNVT Type Name
1	SNVT_amp	83	SNVT_state
2	SNVT_amp_mil	84	SNVT_time_stamp
3	SNVT_angle	85	SNVT_zerospan
4	SNVT_angle_vel	86	SNVT_magcard
5	SNVT_btu_kilo	87	SNVT_elapsed_tm
6	SNVT_btu_mega	88	SNVT_alarm
7	SNVT_char_ascii	89	SNVT_currency
8	SNVT_count	90	SNVT_file_pos
9	SNVT_count_inc	91	SNVT_muldiv
10	SNVT_date_cal	92	SNVT_obj_request
11	SNVT_date_day	93	SNVT_obj_status
12	SNVT_date_time	94	SNVT_preset
13	SNVT_elec_kwh	95	SNVT_switch
14	SNVT_elec_whr	96	SNVT_trans_table
15	SNVT_flow	97	SNVT_override
16	SNVT_flow_mil	98	SNVT_pwr_fact
17	SNVT_length	99	SNVT_pwr_fact_f
18	SNVT_length_kilo	100	SNVT_density
19	SNVT_length_micr	101	SNVT_density_f
20	SNVT_length_mil	102	SNVT_rpm
21	SNVT_lev_cont	103	SNVT_hvac_emerg
22	SNVT_lev_disc	104	SNVT_angle_deg
23	SNVT_mass	105	SNVT_temp_p
24	SNVT_mass_kilo	106	SNVT_temp_setpt
25	SNVT_mass_mega	107	SNVT_time_sec
26	SNVT_mass_mil	108	SNVT_hvac_mode
27	SNVT_power	109	SNVT_occupancy
28	SNVT_power_kilo	110	SNVT_area
29	SNVT_ppm	111	SNVT_hvac_overid
30	SNVT_press	112	SNVT_hvac_status
31	SNVT_res	113	SNVT_press_p
32	SNVT_res_kilo	114	SNVT_address
33	SNVT_sound_db	115	SNVT_scene
34	SNVT_speed	116	SNVT_scene_cfg
35	SNVT_speed_mil	117	SNVT_setting
36	SNVT_str_asc	118	SNVT_evap_state
37	SNVT_str_int	119	SNVT_therm_mode
38	SNVT_telcom	120	SNVT_defr_mode
39	SNVT_temp	121	SNVT_defr_term
40	SNVT_time_passed	122	SNVT_defr_state
41	SNVT_vol	123	SNVT_time_min
42	SNVT_vol_kilo	124	SNVT_time_hour
43	SNVT_vol_mil	125	SNVT_ph
44	SNVT_volt	126	SNVT_ph_f
45	SNVT_volt_dbmv	127	SNVT_chlr_status
46	SNVT_volt_kilo	128	SNVT_tod_event
47	SNVT_volt_mil	129	SNVT_smo_obscur
48	SNVT_amp_f	130	SNVT_fire_test
49	SNVT_angle_f	131	SNVT_temp_ror
50	SNVT_angle_vel_f	132	SNVT_fire_init
51	SNVT_count_f	133	SNVT_fire_indcte
52	SNVT_count_inc_f	134	SNVT_time_zone
53	SNVT_flow_f	135	SNVT_earth_pos
54	SNVT_length_f	136	SNVT_reg_val
55	SNVT_lev_cont_f	137	SNVT_reg_val_ts

SNVT Type Number	SNVT Type Name	SNVT Type Number	SNVT Type Name
56	SNVT_mass_f	138	SNVT_volt_ac
57	SNVT_power_f	139	SNVT_amp_ac
58	SNVT_ppm_f	143	SNVT_turbidity
59	SNVT_press_f	144	SNVT_turbidity_f
60	SNVT_res_f	145	SNVT_hvac_type
61	SNVT_sound_db_f	146	SNVT_elec_kwh_l
62	SNVT_speed_f	147	SNVT_temp_diff_p
63	SNVT_temp_f	148	SNVT_ctrl_req
64	SNVT_time_f	149	SNVT_ctrl_resp
65	SNVT_vol_f	150	SNVT_ptz
66	SNVT_volt_f	151	SNVT_privacyzone
67	SNVT_btu_f	152	SNVT_pos_ctrl
68	SNVT_elec_whr_f	153	SNVT_enthalpy
69	SNVT_config_src	154	SNVT_gfci_status
70	SNVT_color	155	SNVT_motor_state
71	SNVT_grammage	156	SNVT_pumpset_mn
72	SNVT_grammage_f	157	SNVT_ex_control
73	SNVT_file_req	158	SNVT_pumpset_sn
74	SNVT_file_status	159	SNVT_pump_sensor
75	SNVT_freq_f	160	SNVT_abs_humid
76	SNVT_freq_hz	161	SNVT_flow_p
77	SNVT_freq_kilohz	162	SNVT_dev_c_mode
78	SNVT_freq_milhz	163	SNVT_valve_mode
79	SNVT_lux	164	SNVT_alarm_2
80	SNVT_ISO_7811	165	SNVT_state_64
81	SNVT_lev_percent	166	SNVT_nv_type
82	SNVT_multiplier		

Scan Time: Specify Rate at which data is polled this is a timing parameter.

9.5.3.6 Server Side Connection

The Server Side Sections are functionally the same as their Client Side equivalents, except that Server parameters are being defined. Refer Section 9.3.3.6 for detailed information.

```
//=============================================================================
//
//    Server Side Connections
//

Connections
Adapter , Protocol
N1      , Bacnet_IP

//=============================================================================
```

9.5.3.7 Server Side Nodes

Refer Section 9.3.3.7 for detailed information.

```
//=============================================================================
//
//    Server Side Nodes
//

Nodes
Node_Name   , Node_ID , Protocol   , Srv_Offline_Method
B20_VRF_01 , 1001    , Bacnet_IP , Always_Respond

//=============================================================================
```

9.5.3.8 Server Side Map Descriptors

Refer Section 9.3.3.8 for detailed information.

```
//===============================================================================
//
//    Server Side Map Descriptors
//
//Sensor Outputs
Map_Descriptors
Map_Descriptor_Name        , Data_Array_Name , Data_Array_Offset , Function , Node_Name    , Data_Type ,Object_ID , Units    ,
Minimum_Send_Time          , DA_AI_01        ,       0           , Server   ,  B20_VRF_01 ,   AI      ,  1       , hours    ,
Send_Heartbeat             , DA_AI_01        ,       1           , Server   ,  B20_VRF_01 ,   AI      ,  2       , No-Units ,
Send_On_Delta_Tempearature , DA_AI_01        ,       2           , Server   ,  B20_VRF_01 ,   AI      ,  3       , Deg-C    ,
Delay_Start_Up_Time        , DA_AI_01        ,       3           , Server   ,  B20_VRF_01 ,   AI      ,  4       , No-Units ,
Range_Maximum              , DA_AI_01        ,       4           , Server   ,  B20_VRF_01 ,   AI      ,  5       , No-Units ,
```

9.5.4 Troubleshooting Tips

9.5.4.1 If no communication over LonWorks

- Check physical connection of LonWork device with GW BMS Gateway.
- Check and verify the Neuron ID, SNVT Index and SNVT type configured in CSV file.

9.6 Example of CSV file Modbus TCP to BACnet IP

9.6.1 Steps to prepare CSV File

- Define Common Information
- Define Data Arrays
- Define Client Side connections
- Configure Client side Nodes
- Configure Client side Map descriptor
- Define Server Side Connection
- Configure Server Side Nodes
- Configure Server Side Map descriptor

9.6.2 Details required to configuring CSV for Modbus TCP to BACnet IP conversion

Sr No	Details Required
1	IP address of Modbus TCP device
2	Modbus TCP slave device ID.
3	Modbus register addresses of parameters to be mapped.

Below CSV Example is for Modbus TCP to BACnet IP conversion. 1 VFD is connected to Ethernet Switch and GW BMS Gateway is also connected to Ethernet switch. The Modbus mapping of the VFD is as follows.

Device ID of VFD: 1
IP Address of VFD: 192.168.1.25

IP Address of BMS gateway: 192.168.1.200
IP address of VFD and GW BMS Gateway should be on same subnet.

Parameter Name	Modbus Address
Speed	40001
Frequency	40002
Current	40003
Power	40004
DC Bus Voltage	40005
Output Voltage	40006
Run Hours	40007
Kwh	40008
KVA	40009
KVAR	40010

```
//=============================================================================
//
//    Customer      : XYZ
//    Location      : Pune
//    Configured By : ABC
//    Date          : 10/05/2021
//
//=============================================================================
//
//    Common Information
//
Bridge
Title              , System_Node_Id ,Network_Number
San Telequip v1.01 ,    1000        ,   5

//=============================================================================
//
//    Data Arrays
//
Data_Arrays
Data_Array_Name , Data_Format , Data_Array_Length
DA_AI_01        , Int16       ,10

//=============================================================================
//
//    Client Side Connections
//
Connections
Adapter ,  Protocol
N1      ,  Modbus/TCP

//=============================================================================
//
//    Client Side Nodes
//
Nodes
Node_Name   , Node_ID ,  Protocol   , Adapter ,IP_Address
CMD_VFD_01  ,    1    , Modbus/TCP  ,    N1   ,192.168.1.25

//=============================================================================
//
//    Client Side Map Descriptors
//

Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name  , Address , Length , Scan_Interval
CMD_VFD_01          ,   DA_AI_01      ,       0           ,   Rdbc   , CMD_VFD_01 ,  40001  ,   10   , 1.000s

//=============================================================================
```

```
//=============================================================================
//
//    Server Side Connections
//

Connections
Adapter , Protocol
N1      , Bacnet_IP

//=============================================================================
//
//    Server Side Nodes
//

Nodes
Node_Name    , Node_ID , Protocol  , Srv_Offline_Method
VFD_01       ,  1001   , Bacnet_IP , Always_Respond

//=============================================================================
//
//    Server Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name    , Data_Type , Object_ID , Units     ,
Speed_01            ,   DA_AI_01      ,       0           , Server   ,  VFD_01      ,     AV    ,    1      ,  No-units ,
Frequency_01        ,   DA_AI_01      ,       1           , Server   ,  VFD_01      ,     AV    ,    2      ,  Hz       ,
Current_01          ,   DA_AI_01      ,       2           , Server   ,  VFD_01      ,     AV    ,    3      ,  Amps     ,
Power_01            ,   DA_AI_01      ,       3           , Server   ,  VFD_01      ,     AV    ,    4      ,  KW       ,
Dc bus voltage_01   ,   DA_AI_01      ,       4           , Server   ,  VFD_01      ,     AV    ,    5      ,  Volts    ,
Output Voltage_01   ,   DA_AI_01      ,       5           , Server   ,  VFD_01      ,     AV    ,    6      ,  Volts    ,
Run hours_01        ,   DA_AI_01      ,       6           , Server   ,  VFD_01      ,     AV    ,    7      ,  No-units ,
Kwh_01              ,   DA_AI_01      ,       7           , Server   ,  VFD_01      ,     AV    ,    8      ,  No-units ,
KVAR_01             ,   DA_AI_01      ,       8           , Server   ,  VFD_01      ,     AV    ,    9      ,  No-units ,
KVAR_01             ,   DA_AI_01      ,       9           , Server   ,  VFD_01      ,     AV    ,    10     ,  No-units ,
```

```
Data_Array_Low_Scale , Data_Array_High_Scale , Node_Low_Scale , Node_High_Scale
          0          ,         100           ,       0       ,     100
          0          ,         100           ,       0       ,     100
          0          ,         100           ,       0       ,     100
          0          ,         100           ,       0       ,     100
          0          ,         100           ,       0       ,     100
          0          ,         100           ,       0       ,     100
          0          ,         100           ,       0       ,     100
          0          ,         100           ,       0       ,     100
          0          ,         100           ,       0       ,     100
          0          ,         100           ,       0       ,     100
          0          ,         100           ,       0       ,     100
```

9.6.3 Description of the CSV file

The file begins with general information. It is contain Customer name, Site location, CSV file Configured by and Date of configuration. Refer Section 9.3.3 for detailed information.

9.6.3.1 Common Information

The Common Information Section allows for the determination of parameters It is not directly related to any of the connections.

```
//=============================================================================
//
//    Common Information
//
Bridge
Title              , System_Node_Id ,Network_Number
San Telequip v1.01 ,    1000        ,   5

//=============================================================================
```

Refer Section 9.3.3.1 for detailed information.

9.6.3.2 Data Array

Data Arrays are buffers for storage of data to be passed between protocols. It is necessary to declare the data format of each of the Data Arrays to facilitate correct storage of the relevant data. Refer Section 3.3.3.2 for detailed information.

```
//================================================================================
//
//    Data Arrays
//
Data_Arrays
Data_Array_Name , Data_Format , Data_Array_Length
DA_AI_01        , Int16       ,10

//================================================================================
```

9.6.3.3 Client Side Connections

The Client Side Connections Section contains the parameters that describe the nature of the physical connection to the Server Nodes.

```
//================================================================================
//
//    Client Side Connections
//
Connections
Adapter ,  Protocol
N1      ,  Modbus/TCP

//================================================================================
```

Adapter: Adapter definition applies to defining network and GW BMS GATEWAY connections. Ethernet port of GW BMS GATEWAY is defined as N1 in CSV file configuration.

Protocol: Specify the protocol used.

9.6.3.3 Client Side Nodes

The Client Side Nodes Section defines the logical connection parameters for the Server Nodes communicating with the GW BMS GATEWAY.

```
//================================================================================
//
//    Client Side Nodes
//
Nodes
Node_Name   , Node_ID ,  Protocol   , Adapter ,IP_Address
CMD_VFD_01  ,    1    , Modbus/TCP  ,    N1   ,192.168.1.25

//================================================================================
```

Node_Name: Used to provide name for Node. It support Up to 32 alphanumeric characters.

Node_ID: Device ID of the slave devices**.**

Protocol: Specify the slave devices protocol

Adapter: Specify which Adapter the slave device is connected to the GW BMS GATEWAY.

IP Address: Specify the IP address of Modbus TCP slave devices.

9.6.3.4 Client Side Map Descriptor

The Map Descriptor Section contains parameters that describe the address details required to move data between the GW BMS GATEWAY and an external device and the nature of the data transfer. Refer Section 9.3.3.4 for detailed information.

```
//================================================================================
//
//    Client Side Map Descriptors
//

Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name  , Address , Length , Scan_Interval
CMD_VFD_01          ,   DA_AI_01      ,       0           ,   Rdbc   , CMD_VFD_01 ,  40001  ,   10   , 1.000s

//================================================================================
```

9.6.3.5 Server Side Connection

The Server Side Sections are functionally the same as their Client Side equivalents, except that Server parameters are being defined. Refer Section 9.3.3.5 for detailed information.

```
//================================================================================
//
//    Server Side Connections
//

Connections
Adapter , Protocol
N1      , Bacnet_IP

//================================================================================
```

9.6.3.6 Server Side Nodes

Refer Section 3.3.3.6 for detailed Information.

```
//================================================================================
//
//    Server Side Nodes
//

Nodes
Node_Name     , Node_ID , Protocol   , Srv_Offline_Method
VFD_01        ,  1001   , Bacnet_IP , Always_Respond

//================================================================================
```

9.6.3.7 Server Side map Descriptor

The Server Side Map Descriptor Section contains parameter which needs to discovered on BACnet IP client for under each BACnet Node. Refer Section 3.3.3.7 for detailed information.

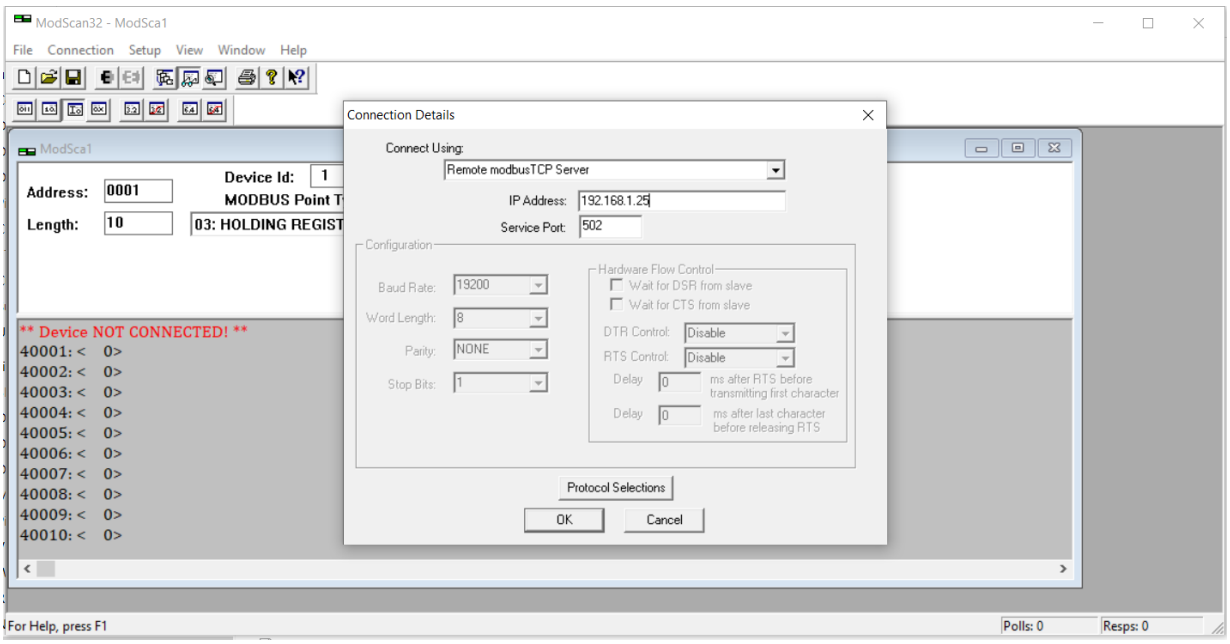
```
//===============================================================================
//
//    Server Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name    , Data_Type , Object_ID , Units     , Data_Array_Low_Scale , Data_Array_High_Scale , Node_Low_Scale , Node_High_Scale
Speed_01            ,   DA_AI_01      ,        0          , Server   , VFD_01      ,     AV    ,    1      , No-units ,          0         ,          100          ,         0       ,         100
Frequency_01        ,   DA_AI_01      ,        1          , Server   , VFD_01      ,     AV    ,    2      , Hz       ,          0         ,          100          ,         0       ,         100
Current_01          ,   DA_AI_01      ,        2          , Server   , VFD_01      ,     AV    ,    3      , Amps     ,          0         ,          100          ,         0       ,         100
Power_01            ,   DA_AI_01      ,        3          , Server   , VFD_01      ,     AV    ,    4      , KW       ,          0         ,          100          ,         0       ,         100
Dc bus voltage_01   ,   DA_AI_01      ,        4          , Server   , VFD_01      ,     AV    ,    5      , Volts    ,          0         ,          100          ,         0       ,         100
Output Voltage_01   ,   DA_AI_01      ,        5          , Server   , VFD_01      ,     AV    ,    6      , Volts    ,          0         ,          100          ,         0       ,         100
Run hours_01        ,   DA_AI_01      ,        6          , Server   , VFD_01      ,     AV    ,    7      , No-units ,          0         ,          100          ,         0       ,         100
Kwh_01              ,   DA_AI_01      ,        7          , Server   , VFD_01      ,     AV    ,    8      , No-units ,          0         ,          100          ,         0       ,         100
KVAR_01             ,   DA_AI_01      ,        8          , Server   , VFD_01      ,     AV    ,    9      , No-units ,          0         ,          100          ,         0       ,         100
KVAR_01             ,   DA_AI_01      ,        9          , Server   , VFD_01      ,     AV    ,    10     , No-units ,          0         ,          100          ,         0       ,         100
```

9.6.4 Check Modbus TCP Device Communication on Modscan

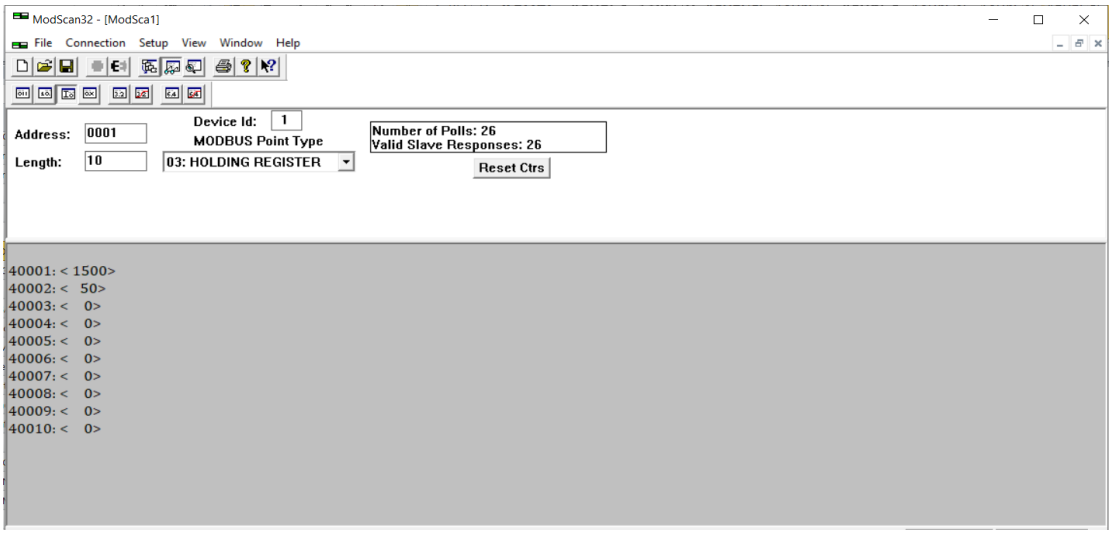
To check Modbus TCP device communication Modscan software is useful. Following are the steps to check Modbus Communication.

- Connect Modbus TCP Slave device and PC/Laptop to Ethernet switch or directly one to one. Set PC and Modbus TCP device IP on same subnet.

Open Modscan and Configure IP settings as shown below screenshot.



- Click on OK and check communication.
- Select appropriate Modbus address Device ID and check the parameter values.



9.6.4 Troubleshooting Tips

9.6.4.1 If No Communication over Modbus TCP

- Check Configured CSV file details of Modbus TCP device such as IP address, Slave device ID, Modbus Register addressing.
- Check GW BMS Gateway IP and Modbus TCP device IP. It should on same subnet.
- Check Network Configuration Avoid Duplicate IP address on the local area network (LAN).
- Avoid Low quality Ethernet cable
- Check Switch Port
- Sometimes switch ports are configured to be disabled or turned off by the IT network administrator
- Check speed settings

9.7 Example of CSV file SNMP to BACnet IP

9.7.1 Steps to prepare CSV File

- Define Common Information
- Define Data Arrays
- Define Client Side connections
- Configure Client side Nodes
- Configure Client side Map descriptor
- Define Server Side Connection
- Configure Server Side Nodes
- Configure Server Side Map descriptor

9.7.2 Details required to configuring CSV for SNMP to BACnet IP conversion

Sr No	Details Required
1	Number of SNMP devices to be connected to the gateway and their IP addresses
2	MIB file of SNMP device.
3	OID list of parameters to be mapped into the gateway.

Below CSV Example is for SNMP to BACnet IP conversion. 1 EM and GW BMS Gateway is connected to Ethernet switch. The Modbus mapping of the EM is as follows.

IP Address of EM: 192.168.1.24
IP Address of BMS gateway: 192.168.1.200

Parameter Name	OID
Kwh	1.3.6.1.4.1.2606.5.5.1.0
Frequency	1.3.6.1.4.1.2606.5.5.2.0

```
//=============================================================================
//
//    Customer      : XYZ
//    Location      : Pune
//    Configured By : ABC
//    Date          : 10/05/2021
//
//=============================================================================
//
//    Common Information
//
Bridge
Title              , System_Node_Id ,Network_Number
San Telequip v1.01 ,    1000        ,   5

//=============================================================================
//
//    Data Arrays
//
Data_Arrays
Data_Array_Name , Data_Format , Data_Array_Length
DA_AI_01        , Int16       ,         10

//=============================================================================
//
//    Client Side Connections
//
Connections
Adapter , Protocol
N1      , SNMP-STD

//=============================================================================
//
//    Client Side Nodes
//
Nodes
Node_Name , Protocol , Adapter , IP_Address
Dev_01    , SNMP-STD , N1      , 192.168.1.24

//=============================================================================
//
//    Client Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , SNMP_OID                , Length , Scan_Interval
CMD_Dev_01_01       , DA_AI_01        ,      0            , Rdbc     , Dev_01    , 1.3.6.1.4.1.2606.5.5.1.0 ,    1   , 1.000s
CMD_Dev_01_02       , DA_AI_01        ,      1            , Rdbc     , Dev_01    , 1.3.6.1.4.1.2606.5.5.2.0 ,    1   , 1.000s

//=============================================================================
```

```
//=============================================================================
//
//    Server Side Connections
//

Connections
Adapter , Protocol
N1      , Bacnet_IP

//=============================================================================
//
//    Server Side Nodes
//

Nodes
Node_Name    , Node_ID , Protocol  , Srv_Offline_Method
EM_01        ,  1001   , Bacnet_IP , Always_Respond

//=============================================================================
//
//    Server Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name    , Data_Type , Object_ID , Units     , Data_Array_Low_Scale
KWh_01              ,   DA_AI_01      ,       0          , Server   ,  EM_01      ,     AV    ,    1      ,  No-units ,          0
Frequency_01        ,   DA_AI_01      ,       1          , Server   ,  EM_01      ,     AV    ,    2      ,  Hz       ,          0
```

```
, Data_Array_High_Scale , Node_Low_Scale , Node_High_Scale
,        100            ,        0      ,     100
,        100            ,        0      ,     100
```

9.7.3 Description of the CSV file

The file begins with general information. It is contain Customer name, Site location, CSV file Configured by and Date of configuration. Refer Section 9.3.3 for detailed information.

9.7.3.1 Common Information

The Common Information Section allows for the determination of parameters it is not directly related to any of the connections.

```
//=============================================================================
//
//    Common Information
//
Bridge
Title              , System_Node_Id ,Network_Number
San Telequip v1.01 ,    1000        ,   5

//=============================================================================
```

Refer Section 9.3.3.1 for detailed information.

9.7.3.2 Data Array

Data Arrays are buffers for storage of data to be passed between protocols. It is necessary to declare the data format of each of the Data Arrays to facilitate correct storage of the relevant data. Refer Section 3.3.3.2 for detailed information.

```
//================================================================================
//
//    Data Arrays
//
Data_Arrays
Data_Array_Name , Data_Format , Data_Array_Length
DA_AI_01        , Int16       ,10

//================================================================================
```

9.7.3.3 Client Side Connections

The Client Side Connections Section contains the parameters that describe the nature of the physical connection to the Server Nodes.

```
//================================================================================
//
//    Client Side Connections
//
Connections
Adapter , Protocol
N1      , SNMP-STD

//================================================================================
```

Adapter: Adapter definition applies to defining network and GW BMS GATEWAY connections. Ethernet port of GW BMS GATEWAY is defined as N1 in CSV file configuration.

Protocol: Specify the protocol used.

9.7.3.3 Client Side Nodes

The Client Side Nodes Section defines the logical connection parameters for the Server Nodes communicating with the GW BMS GATEWAY.

```
//================================================================================
//
//    Client Side Nodes
//
Nodes
Node_Name , Protocol , Adapter , IP_Address
Dev_01    , SNMP-STD , N1      , 192.168.1.24

//================================================================================
```

Node_Name: Used to provide name for Node. It support Up to 32 alphanumeric characters.

Protocol: Specify the slave devices protocol

Adapter: Specify which Adapter the slave device is connected to the GW BMS GATEWAY.

IP Address: Specify the IP address of Modbus TCP slave devices.

9.7.3.4 Client Side Map Descriptor

The Map Descriptor Section contains parameters that describe the address details required to move data between the GW BMS GATEWAY and an external device and the nature of the data transfer.

```
//=============================================================================
//
//    Client Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , SNMP_OID                , Length , Scan_Interval
CMD_Dev_01_01       , DA_AI_01        ,      0            , Rdbc     , Dev_01    , 1.3.6.1.4.1.2606.5.5.1.0 ,    1   , 1.000s
CMD_Dev_01_02       , DA_AI_01        ,      1            , Rdbc     , Dev_01    , 1.3.6.1.4.1.2606.5.5.2.0 ,    1   , 1.000s

//=============================================================================
```

Map_Descriptor_Name: Specify the Name of this Map Descriptor. It support Up to 32 alphanumeric characters.

Data_Array_Name: Name of Data Array where data is to be stored in the GW BMS GATEWAY.

Data_Array_Offset: Specify the Starting location in Data Array.

Function: Function of Client Map Descriptor. Rdbc is used for read and write both operation.

Node_Name: Name of Node to fetch data from.

SNMP_OID: Specify the OID of the Parameters**.**

Scan Interval: Specify Rate at which data is polled this is a timing parameter.

9.7.3.5 Server Side Connection

The Server Side Sections are functionally the same as their Client Side equivalents, except that Server parameters are being defined. Refer Section 9.3.3.5 for detailed information.

```
//=============================================================================
//
//    Server Side Connections
//

Connections
Adapter , Protocol
N1      , Bacnet_IP

//=============================================================================
```

9.7.3.6 Server Side Nodes

Refer Section 3.3.3.6 for detailed Information.

```
//=============================================================================================
//
//     Server Side Nodes
//

Nodes
Node_Name     , Node_ID , Protocol   , Srv_Offline_Method
EM_01         ,   1001    , Bacnet_IP , Always_Respond

//=============================================================================================
```

9.7.3.7 Server Side map Descriptor

The Server Side Map Descriptor Section contains parameter which needs to discovered on BACnet IP client for under each BACnet Node. Refer Section 3.3.3.7 for detailed information.

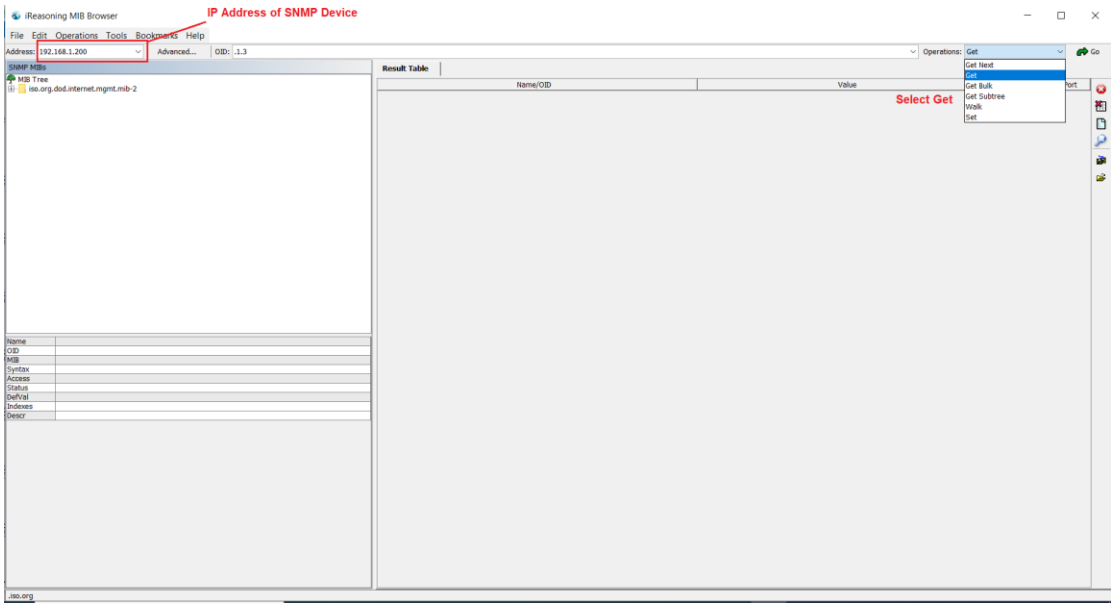
```
//=============================================================================
//
//    Server Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name    , Data_Type , Object_ID , Units     , Data_Array_Low_Scale , Data_Array_High_Scale , Node_Low_Scale , Node_High_Scale
KWh_01              ,   DA_AI_01      ,       0          , Server   , EM_01      ,     AV    ,    1      , No-units ,          0          ,       100             ,       0        ,    100
Frequency_01        ,   DA_AI_01      ,       1          , Server   , EM_01      ,     AV    ,    2      , Hz       ,          0          ,       100             ,       0        ,    100
```

9.6.4 Check SNMP Device Communication on MIB Browser

To check SNMP device communication MIB Browser software is useful. Following are the steps to check Communication.

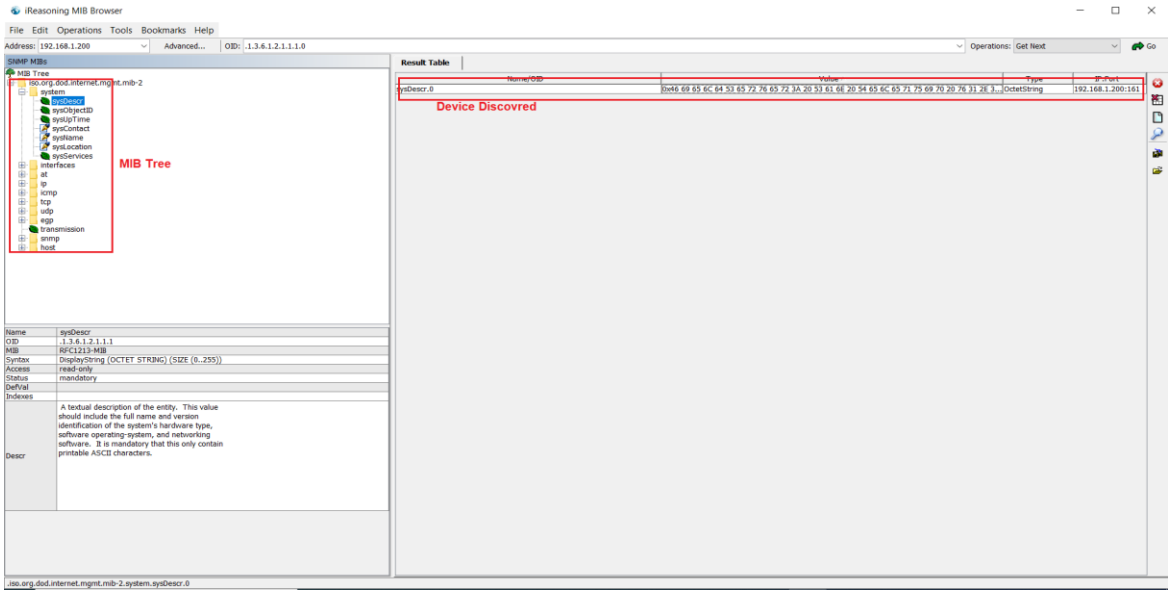
- Connect SNMP Slave device and PC/Laptop to Ethernet switch or directly one to one. Set PC and SNMP device IP on same subnet.

Open MIB Browser and Configure IP settings as shown below screenshot.



- After click on get SNMP device will be automatically discovered as shown below.

- Double click on the discovered device.
- The MIB tree will be discovered automatically.



9.7.4 Troubleshooting Tips

9.7.4.1 If No Communication over SNMP

- Check Configured CSV file details of SNMP device such as IP address, OID.
- Check PC GW BMS Gateway IP and Modbus TCP device IP. It should on same subnet.
- Check Network Configuration Avoid Duplicate IP address on the local area network (LAN).
- Avoid Low quality Ethernet cable
- Check Switch Port
- Sometimes switch ports are configured to be disabled or turned off by the IT network administrator
- Check speed settings

10 Additional Information

10.1 BACnet Communication Methods

GW BMS GATEWAY Support Poll and COV method to communicating on BACnet IP. By default Poll is selected. To select COV below change need to do in CSV file on server side node section**.**

```
//===============================================================================
//
//     Server Side Nodes
//

Nodes
Node_Name     , Node_ID , Protocol   , Srv_Offline_Method   , Node_Option
VFD_01        ,  1001   , Bacnet_IP , Always_Respond       , COV_Enable
VFD_02        ,  1002   , Bacnet_IP , Always_Respond       , COV_Enable

//===============================================================================
```

10.2 BBMD

GW BMS GATEWAY Support BBMD functionality for BACnet Communication. A BBMD (BACnet Broadcast Management Device) is used to allow devices on different subnets to communicate to one another. BACnet/IP requires that a BBMD be defined on every subnet. To Enable BBMD Below changes need to do in CSV file.

```
//================================================================================
//
//    Server Side Connections
//

Connections
Adapter , Protocol    , Connection_Type
N1      , Bacnet_IP   , BBMD

//================================================================================
```

Also Broadcast Distribution Table (BDT) needs to update is GW BMS GATEWAY. The Broadcast distribution table is looks as below**.**

```
//Bdt.ini
//The format of this table must be:
//
//BBMD IP_Address , BBMD port , BBMD subnet Mask
10.25.126.78      , 47808     , 255.255.255.255
10.25.6.59        , 47808     , 255.255.255.255
```

The file must be created and downloaded to the GW BMS GATEWAY using the FS-GUI:

- Enter the IP Address of the GW BMS GATEWAY into a web browser
- Choose the ‘Setup’ option in the Navigation Tree and Select ‘File Transfer’
- Choose the ‘General’ tab
- Click on the ‘Browse’ button and select bdt.ini
- Click on ‘Submit’
- When it says “Configuration update complete”, click on the ‘System Restart’ button

NOTE: BBMD operation is not required if there is already another BBMD on the subnet – there can only be one BBMD per subnet.

10.3 IP Port

Specify the UDP port that will be used to communicate with other BACnet Client devices. Default IP port in GW BMS GATEWAY is 47808. To change it below change need to do in CSV file.

```
//================================================================================
//
//    Server Side Connections
//

Connections
Adapter , Protocol    , IP_PORT
N1      , Bacnet_IP   , 47809

//================================================================================
```

10.4 BACnet Data Types

GW BMS GATEWAY support following BACnet data types.

Values	BACnet Number	Description
AI	0	ANALOG_INPUT
AO	1	ANALOG_OUTPUT
AV	2	ANALOG_VALUE
BI	3	BINARY_INPUT
BO	4	BINARY_OUTPUT
BV	5	BINARY_VALUE
MI	13	MULTIPLE_INPUT
MO	14	MULTIPLE_OUTPUT
MV	19	MULTIPLE_VALUE
NC	15	NOTIFICATION_CLASS
LSP	21	LIFE_SAFETY_POINT
DEVICE	8	DEVICE

10.5 How to apply Scaling factor to original value

Scalling factor is required to divide original value by any number.
For Example: Frequency parameter is configured at BACnet side. The Original value coming from Modbus slave device is 500 and it has to show as 50.0 on BACnet.
Following changes needs to do in CSV for this change.

```
//=============================================================================
//
//    Server Side Map Descriptors
//
Map_Descriptors
Map Descriptor Name , Data Array Name , Data Array Offset , Function , Node Name    , Data Type , Object ID , Units
Frequency_01        ,   DA_AI_01      ,       1           , Server   ,  VFD_01      ,     AV    ,    2      ,  Hz
```

```
 , Data Array Low Scale , Data Array High Scale , Node Low Scale , Node High_Scale
 ,          0           ,       100             ,        0        ,     10
```

Scaling Formula in GW BMS GATEWAY
BACnet Value = (Node_High_Scale / Data_Array_High_Scale)*Modbus Value.

10.6 Units for BACnet IP

Used to specify engineering units to interpret data if used. Will display a dash if not used. This field is protocol dependent and used in BACnet IP and BACnet MSTP Protocols.

San Telequip (P) Ltd.,
504/505 Deron Heights, Baner Road, Baner
Pune 411045, Mah, India
Phone : +91 20-27293456, 8390069393, 9764027070
email : info@santelequip.com

Unit	Variation 1	Variation 2	Type
amperes	amps	A	Electrical
ampere-seconds			Energy
amperes-per-meter			Electrical
amperes-per-square-meter			Electrical
ampere-square-hours			Energy
ampere-square-meters			Electrical
bars			Pressure
becquerels			Other
btus			Energy
btus-per-hour			Power
btus-per-pound			Enthalpy
btus-per-pound-dry-air			Enthalpy
candelas			Light
candelas-per-square-meter			Light
centimeters			Length
centimeters-of-mercury			Pressure
centimeters-of-water			Pressure
cubic-feet			Volume
cubic-feet-per-day			Volumetric Flow
cubic-feet-per-hour			Volumetric Flow
cubic-feet-per-minute			Volumetric Flow
cubic-feet-per-second			Volumetric Flow
cubic-meters			Volume
cubic-meters-per-day			Volumetric Flow
cubic-meters-per-hour			Volumetric Flow
cubic-meters-per-minute			Volumetric Flow
cubic-meters-per-second			Volumetric Flow
currency1			Currency
currency10			Currency
currency2			Currency
currency3			Currency
currency4			Currency
currency5			Currency
currency6			Currency
currency7			Currency
currency8			Currency
currency9			Currency
cycles-per-hour			Frequency
cycles-per-minute			Frequency
days			Time
decibels			Electrical
decibels-a			Other
decibels-millivolt			Electrical
decibels-volt			Electrical
degree-days-Celsius			Temperature
degree-days-Fahrenheit			Temperature
degrees-angular			Other
degrees-Celsius	Deg-C	Deg_C	Temperature
degrees-Celsius-per-hour			Other
degrees-Celsius-per-minute			Other
degrees-Fahrenheit	Deg-F	Deg_F	Temperature
degrees-Fahrenheit-per-hour			Other
degrees-Fahrenheit-per-minute			Other
degrees-Kelvin	Deg-K	Deg_K	Temperature
degrees-Kelvin-per-hour			Temperature
degrees-Kelvin-per-minute			Temperature
degrees-phase			Electrical
delta-degrees-Fahrenheit			Temperature
delta-degrees-Kelvin			Temperature
farads			Electrical
feet			Length
feet-per-minute			Velocity
feet-per-second			Velocity
foot-candles			Light

San Telequip (P) Ltd.,
504/505 Deron Heights, Baner Road, Baner
Pune 411045, Mah, India
Phone : +91 20-27293456, 8390069393, 9764027070
email : info@santelequip.com

Unit	Variation 1	Variation 2	Type
grams			Mass
grams-of-water-per-kilogram-dry-air			Humidity
grams-per-cubic-centimeter			Other
grams-per-cubic-meter			Other
grams-per-gram			Other
grams-per-kilogram			Other
grams-per-liter			Other
grams-per-milliliter			Other
grams-per-minute			Mass Flow
grams-per-second			Mass Flow
grams-per-square-meter			Other
gray			Other
hectopascals			Pressure
henrys			Electrical
hertz	Hz		Frequency
horsepower	HP		Power
hours			Time
hundredths-seconds			Time
imperial-gallons			Volume
imperial-gallons-per-minute			Volumetric Flow
inches			Length
inches-of-mercury			Pressure
inches-of-water			Pressure
joule-per-hours			Power
joules			Energy
joule-seconds			Other
joules-per-cubic-meter			Other
joules-per-degree-Kelvin			Entropy
joules-per-kilogram-degree-Kelvin			Entropy
joules-per-kilogram-dry-air			Enthalpy
kilobecquerels			Other
kilo-btus			Energy
kilo-btus-per-hour			Power
kilograms	kg		Mass
kilograms-per-cubic-meter			Other
kilograms-per-hour			Mass Flow
kilograms-per-kilogram			Other
kilograms-per-minute			Mass Flow
kilograms-per-second			Mass Flow
kilohertz	kHz		Frequency
kilohms			Electrical
kilojoules			Energy
kilojoules-per-degree-Kelvin			Entropy
kilojoules-per-kilogram			Energy
kilojoules-per-kilogram-dry-air			Enthalpy
kilometers			Length
kilometers-per-hour			Velocity
kilopascals	Kpa		Pressure
kilovolt-ampere-hours			Energy
kilovolt-ampere-hours-reactive			Energy
kilovolt-amperes	kilovolt-amps	KVA	Electrical
kilovolt-amperes-reactive	KVAR		Electrical
kilovolts			Electrical
kilowatt-hours	kWh		Energy
kilowatt-hours-per-square-foot			Other
kilowatt-hours-per-square-meter			Other
kilowatt-hours-reactive			Energy
kilowatts	kW		Power
liters			Volume
liters-per-hour			Volumetric Flow
liters-per-minute			Volumetric Flow
liters-per-second			Volumetric Flow
lumens			Light
luxes			Light
megabecquerels			Other

San Telequip (P) Ltd.,
504/505 Deron Heights, Baner Road, Baner
Pune 411045, Mah, India
Phone : +91 20-27293456, 8390069393, 9764027070
email : info@santelequip.com

Unit	Variation 1	Variation 2	Type
mega-btus			Energy
megahertz	MHz		Frequency
megajoules			Energy
megajoules-per-degree-Kelvin			Entropy
megajoules-per-kilogram-dry-air			Enthalpy
megajoules-per-square-foot			Other
megajoules-per-square-meter			Other
megavolt-ampere-hours			Energy
megavolt-ampere-hours-reactive			Energy
megavolt-amperes	megavolt-amps		Electrical
megavolt-amperes-reactive	MVAR		Electrical
megavolts			Electrical
megawatt-hours	MWh		Energy
megawatt-hours-reactive			Energy
megawatts	MW		Power
megohms			Electrical
meters			Length
meters-per-hour			Velocity
meters-per-minute			Velocity
meters-per-second			Velocity
meters-per-second-per-second			Acceleration
micrograms-per-cubic-meter			Other
micrograms-per-liter			Other
microgray			Other
micrometers			Length
microsiemens			Electrical
microsieverts			Other
microsieverts-per-hour			Other
miles-per-hour			Velocity
milliamperes	milliamps		Electrical
millibars			Pressure
milligrams			Mass
milligrams-per-cubic-meter			Other
milligrams-per-gram			Other
milligrams-per-kilogram			Other
milligrams-per-liter			Other
milligray			Other
milliliters			Volume
milliliters-per-second			Volumetric Flow
millimeters			Length
millimeters-of-mercury			Pressure
millimeters-of-water			Pressure
millimeters-per-minute			Velocity
millimeters-per-second			Velocity
milliohms			Electrical
million-standard-cubic-feet-per-day			Volumetric Flow
million-standard-cubic-feet-per-minute			Volumetric Flow
millirems			Other
millirems-per-hour			Other
milliseconds			Time
millisiemens			Electrical
millisieverts			Other
millivolts			Electrical
milliwatts			Power
minutes			Time
minutes-per-degree-kelvin			Other
mole-percent			Other
months			Time
nanograms-per-cubic-meter			Other
nephelometric-turbidity-unit			Other
newton			Force
newton-meters			Torque
newton-seconds			Other
newtons-per-meter			Other
no-units	No_Units	None	Other

San Telequip (P) Ltd.,
504/505 Deron Heights, Baner Road, Baner
Pune 411045, Mah, India
Phone : +91 20-27293456, 8390069393, 9764027070
email : info@santelequip.com

Unit	Variation 1	Variation 2	Type
ohm-meters			Electrical
ohm-meter-squared-per-meter			Electrical
ohms			Electrical
parts-per-billion			Other
parts-per-million			Other
pascals			Pressure
pascal-seconds			Other
percent			Other
percent-obscuration-per-foot			Other
percent-obscuration-per-meter			Other
percent-per-second			Other
percent-relative-humidity	% RH; %RH	Percent RH; Percent RH	Humidity
per-hour			Frequency
per-mille			Other
per-minute			Other
per-second			Other
pH			Other
pounds-force-per-square-inch	PSI	pounds-force-per-sq-inch	Pressure
pounds-mass			Mass
pounds-mass-per-day			Volumetric Flow
pounds-mass-per-hour			Mass Flow
pounds-mass-per-minute			Mass Flow
pounds-mass-per-second			Mass Flow
power-factor	PF		Electrical
psi-per-degree-Fahrenheit			Other
radians			Other
radians-per-second			Other
revolutions-per-minute			Other
seconds	Secs	S	Time
siemens			Electrical
siemens-per-meter			Electrical
sieverts			Other
square-centimeters			Area
square-feet			Area
square-inches			Area
square-meters			Area
square-meters-per-Newton			Other
standard-cubic-feet-per-day			Volumetric Flow
teslas			Electrical
therms			Energy
thousand-cubic-feet-per-day			Volumetric Flow
thousand-standard-cubic-feet-per-day			Volumetric Flow
ton-hours			Energy
tons			Mass
tons-per-hour			Mass Flow
tons-refrigeration			Power
us-gallons	Gallons		Volume
us-gallons-per-hour			Volumetric Flow
us-gallons-per-minute	GPM		Volumetric Flow
volt-ampere-hours			Energy
volt-ampere-hours-reactive			Energy
volt-amperes	Volt-Amps	VA	Electrical
volt-amperes-reactive	VAR		Electrical
volts	voltage		Electrical
volts-per-degree-Kelvin			Electrical
volts-per-meter			Electrical
volt-square-hours			Energy
watt-hours	wH		Energy
watt-hours-per-cubic-meter			Other
watt-hours-reactive			Energy
watts	W		Power
watts-per-meter-per-degree-Kelvin			Other
watts-per-square-foot			Light
watts-per-square-meter			Light
watts-per-square-meter-degree-kelvin			Other
webers			Electrical
weeks			Time
years			Time

10.7 Modbus reading Data Types

When a Modbus slave device needs to pass a 32-Bit Integer or a 32-Bit Float, it splits the float into two 16-bit Integers and maps it to consecutive registers. The following data types read the 2 consecutive registers and auto combines them into a 32-Bit Integer or Float before storing it in a Data Array.

- Float_Reg (32-Bit IEEE 754 Floating Point in Holding Register FC03)
- 32Bit_Reg (32-Bit Integer in Holding Register FC03)
- Input_Float (32-Bit IEEE 754 Floating Point in Input Register FC04)
- Input_Reg_32Bit (32-Bit Integer in Input Register FC04)
- Float_Reg_Swap (32-Bit IEEE 754 Floating Point with swapped format in Holding Register FC03)
- 32Bit_Reg_Swap (32-Bit Integer with swapped format in Holding Register FC03)
- Input_Float_Swap (32-Bit IEEE 754 Floating Point with swapped format in Input Register FC04)
- Input_Reg_32Bit_Swap (32-Bit Integer with swapped format in Input Register FC04)

Below changes needs to implement in CSV for this data types.

```
//================================================================================
//
//    Data Arrays
//

Data_Arrays
Data_Array_Name , Data_Format , Data_Array_Length
DA_AI_01        ,   Float     ,        10
DA_AI_02        ,   UInt32    ,        10
```

```
//=============================================================================
//
//    Client Side Map Descriptors
//

Map_Descriptors,,,,,,,,,,
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name   , Data_Type          , Address , Length , Scan_Interval
CMD_EM_01           ,DA_AI_01         ,        0          , Rdbc     ,  Device_01  , Float_Reg          ,  40001  ,   1    , 1.000s
CMD_EM_02           ,DA_AI_02         ,        0          , Rdbc     ,  Device_01  , 32Bit_Reg          ,  40003  ,   1    , 1.000s
CMD_EM_03           ,DA_AI_01         ,        1          , Rdbc     ,  Device_01  , Input_Float        ,  30001  ,   1    , 1.000s
CMD_EM_04           ,DA_AI_02         ,        1          , Rdbc     ,  Device_01  , Input_Reg_32Bit    ,  30003  ,   1    , 1.000s
CMD_EM_05           ,DA_AI_01         ,        2          , Rdbc     ,  Device_01  , Float_Reg_Swap     ,  40005  ,   1    , 1.000s
CMD_EM_06           ,DA_AI_02         ,        2          , Rdbc     ,  Device_01  , 32Bit_Reg_Swap     ,  40007  ,   1    , 1.000s
CMD_EM_07           ,DA_AI_01         ,        3          , Rdbc     ,  Device_01  , Input_Float_Swap   ,  30005  ,   1    , 1.000s
CMD_EM_08           ,DA_AI_02         ,        3          , Rdbc     ,  Device_01  , Input_Reg_32Bit_Swap,  30007  ,   1    , 1.000s
```

10.8 64-Bit Integer and Float Data Types

When a Modbus slave device needs to pass a 64-Bit Integer or a 64-Bit Float, it splits the float into four 16-bit Integers and maps it to consecutive registers. The following data types read the 4 consecutive registers and auto combines them into a 64-Bit Integer or Float, before the scaling is applied (to keep the decimal precision) and stores the scaled value in a Data Array. When serving the value to the output protocol, the reverse scaling needs to be applied.

- 64Bit_Reg (64-Bit Integer in Holding Register FC03)
- Double_Reg (64-Bit IEEE 754 Floating Point in Holding Register FC03)
- Input_Reg_64bit (64-Bit Integer in Input Register FC04)

- Input_Double (64-Bit IEEE 754 Floating Point in Input Register FC04)

Below changes needs to implement in CSV for this data types.

Modbus RTU:

```
//=============================================================================
//
//    Data Arrays
//

Data_Arrays
Data_Array_Name , Data_Format , Data_Array_Length
DA_AI_01        ,    Float    ,      10
DA_AV_01        ,    UInt32   ,      10

//=============================================================================
```

```
//=========================================================================
//
//   Client Side Map Descriptors
//

Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function   , Node_Name , Data_Type          , Address , Length , Scan_Interval
CMD_Dev_01          ,    DA_AI_01     ,        0          ,   Rdbc     ,  Dev_01   ,  64Bit_Reg        ,  40001  ,    1   , 1.000s
CMD_Dev_02          ,    DA_AV_01     ,        0          ,   Rdbc     ,  Dev_01   ,  Double_Reg       ,  40003  ,    1   , 1.000s
CMD_Dev_03          ,    DA_AI_01     ,        1          ,   Rdbc     ,  Dev_01   ,  Input_Reg_64bit  ,  30001  ,    1   , 1.000s
CMD_Dev_04          ,    DA_AV_01     ,        1          ,   Rdbc     ,  Dev_01   ,  Input_Double     ,  30003  ,    1   , 1.000s
```

```
Data_Array_High_Scale , Node_Low_Scale , Node_High_Scale
      1               ,       0        ,    1000000000
      1               ,       0        ,    1000000000
      1               ,       0        ,    1000000000
      1               ,       0        ,    1000000000
```

BACnet IP:

```
//=============================================================================================
//
//     Server Side Nodes
//

Nodes
Node_Name     , Node_ID , Protocol   , Srv_Offline_Method
VFD_01        ,  1001   , Bacnet_IP , Always_Respond

//=============================================================================================
```

```
//=========================================================================
//
//    Server Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name    , Data_Type , Object_ID , Units     ,
Import_Kwh_01       ,   DA_AI_01      ,       0          , Server   ,  VFD_01      ,     AI    ,    1      ,  Kwh      ,
KVAR_01             ,   DA_AI_01      ,       1          , Server   ,  VFD_01      ,     AI    ,    2      ,  No-Units ,
Power_01            ,   DA_AI_01      ,       2          , Server   ,  VFD_01      ,     AI    ,    3      ,  KW       ,
Export_Kwh_01       ,   DA_AI_01      ,       3          , Server   ,  VFD_01      ,     AI    ,    4      ,  Kwh      ,
```

```
, Data_Array_Low_Scale , Data_Array_High_Scale , Node_Low_Scale , Node_High_Scale
,          0           ,          1            ,        0       ,     1000000000
,          0           ,          1            ,        0       ,     1000000000
,          0           ,          1            ,        0       ,     1000000000
,          0           ,          1            ,        0       ,     1000000000
```

10.9 Modbus Address Type

GW BMS GATEWAY support 3 types of Modbus address Type.

- ADU
- PDU
- Modicon_5 digit

By Default Modicon_5 digit is selected.

If Node parameter Address_Type is set as ADU or PDU, then Data_Type must be specified as follows.

For Address_Type ADU:

Address range	Data_Type	Function Code (Write)	Function Code (Read)
0 - 65535	FC01	15	1
0 – 65535	FC02	n/a.	2
0 – 65535	FC04	n/a.	4
0 – 65535	FC03	16	3

For Address_Type PDU:

Address range	Data_Type	Function Code (Write)	Function Code (Read)
1 - 65536	Coil	15	1
1 – 65536	Discrete_Input	n/a.	2
1 – 65536	Input_Register	n/a.	4
1 - 65536	Holding_Register	16	3

For Address_Type Modicon_5digit:

When a Modbus address range is specified, a particular Data Type is implied. The defaults are shown below.

Address range	Data_Type	Function Code (Write)	Function Code (Read)
00001 - 09999	Coil	5,15	1
10001 - 19999	Discrete_Input	n/a.	2
30001 - 39999	Input_Register	n/a.	4
40001 - 49999	Holding_Register	6,16	3

- If writing multiple registers, the write function code will be 16.
- If writing multiple coils, the write function code will be 15.
- If writing a single register, the write function code will be 6.
- If writing a single coil, the write function code will be 5.

Configuration Example:

```
//=============================================================================================
//
//    Client Side Nodes
//
Nodes
Node_Name     , Node_ID , Protocol    , Port   , Address_Type
O_Device_01   ,     1   , Modbus_RTU ,  R1    ,  ADU
O_Device_02   ,     2   , Modbus_RTU ,  R2    ,  ADU

//=============================================================================================
```

```
//=========================================================================
//
//    Client Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name   , Data_Type             , Address , Length , Scan_Interval
CMD_Device_01_1     ,   DA_AI_01      ,       0           , Rdbc     ,O_Device_01  ,  Coil                , 1     ,  1    , 1.000s
CMD_Device_01_2     ,   DA_AI_01      ,       1           , Rdbc     ,O_Device_01  ,  Discrete_Input      , 1     ,  1    , 1.000s
CMD_Device_01_3     ,   DA_AI_01      ,       2           , Rdbc     ,O_Device_01  ,  Holding_Register    , 1     ,  1    , 1.000s
CMD_Device_01_4     ,   DA_AI_01      ,       3           , Rdbc     ,O_Device_01  ,  Input_Register      , 1     ,  1    , 1.000s

//=========================================================================
```

10.10 Ethernet IP Connection Type (CON_Typ) Configuration

GW BMS Gateway Support Explicit, Unconnected and Implicit EIP connections type and GW BMS Gateway configurable as a EIP Client or EIP Server

10.10.1 CON_Typ Configuration When GW BMS Gateway is EIP Client

10.10.1.1 Explicit Type

Within Ethernet IP, the explicit message connection can be thought of as a client/server relationship. The GW BMS Gateway as a Client asks or requests the information from a server, such as a PLC field device, and the server sends the requested information back to the GW BMS gateway. The Information request by GW BMS Gateway is configure in CSV file.

```
//=============================================================================
//
//    Client Side Connections,,
//

Connections
Adapter , Protocol,
N1      , Ethernet/IP,

//=============================================================================
//
//    Client Side Nodes,,
//

Nodes
Node_Name ,Node_ID  , Protocol    , Adapter , IP_Address
EIP_01    ,    2    , Ethernet/IP , N1      ,169.254.122.2

//=============================================================================
//
//    Client Side Map Descriptors,,
//

Map_Descriptors,,
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , EIP_CON_TYP , Node_Name , EIP_PATH
CMD_AI_01           ,   DA_AI_01      ,     0             , Rdbc     , EXPLICIT    , EIP_01    ,   1 0
```

```
   , EIP_SERVICE      , EIP_TAG_NAME  ,Address , Length , Scan_Interval
   , DATA_TABLE_READ ,   Read_Data     ,   0     ,    20    , 1.000s
```

IP Address: IP address of Ethernet IP Device.

EIP_CON_TYP: Define the type of data transfer required. The GW BMS Gateway CON_TYP should match with the Field devices CON_TYP.

EIP_PATH: EIP Path Used to provide the path to the CPU in certain PLC's. EIP Paths vary and are dependent on the structure of the network. Refer below table for EIP paths of some of the PLC.

Device	Typical Path
Direct AB	1 0
AB ENI module	3 1
AB ControlLogix 1756-L55 (With network card 1756-ENBT/A)	1 1 or 1 0
CompactLogix ENI (1769-L31 using the 1761-NET-ENI)	3 1
CompactLogix Direct Connection (P/N 1769-L35E)	1 1 or 1 0
If not required by remote device (ex. Rockwell Micro820SD Controller)	Not_used

EIP_Service: EIP Service defined the action (Read/Write) to be performed. For read its value is **Data_Table_Read** and for write value is **Data_Table_Write**.

EIP_TAG_NAME: The Tag Name Defined in Ethernet IP configuration of field device. The Tag name of GW BMS Gateway and field device should match.

Address: Put the address set in field device.

10.10.1.2 Unconnected Type

```
//=========================================================================
//
//    Client Side Map Descriptors
//

Map_Descriptors,,
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , EIP_CON_TYP , Node_Name , EIP_Class
CMD_AI_01           ,   DA_AI_01      ,     0             , Rdbc     , Unconnected , EIP_01    ,   10
```

```
, EIP_SERVICE     , EIP_Attribute  ,Address , Length , Scan_Interval
, Get_Attrib      ,  3             ,  1     ,   20   , 1.000s
```

EIP_SERVICE: EIP Service defined the action (Read/Write) to be performed. For read its value is **Get_Attrib** and for write value is **Set_Attrib**.

EIP_Attribute: Configure the Class associate with the Class configured. By Default EIP_Attribute is 0 and it support 0-255 values range.

EIP Class: Define the class of data. EtherNet/IP is an object orientated protocol. The Object Oriented structure therefore allows for classes, instances, attributes and services. The ‘data types’ listed below are to be considered as the objects supported in the protocol. Each of these has attributes that have been supported to differing degrees.

10.10.1.3 Implicit Type

```
//=========================================================================
//
//    Client Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , EIP_CON_TYP , Node_Name ,EIP_Class ,Address ,Length , Scan_Interval
CMD_AI_Imp_01       ,    DA_AI_02     ,     0             ,  Rdbc    ,  implicit   , EIP_01    ,    4     ,  101   ,   20  ,   1.000s
```

11 Drivers Information of Protocols supported by GW BMS Gateway

11.1 Modbus RTU / Modbus ASCII/ Modbus TCP

The Modbus RTU, Modbus ASCII and Modbus TCP drivers allow the GW BMS Gateway to transfer data to and from devices using Modbus RTU / Modbus ASCII / Modbus TCP protocol respectively. Data can be transferred over RS 232/ RS-485/ RJ45. The GW BMS Gateway can emulate either a Server or Client.

11.1.1 Connection Information

Connection Type: RS-232 or RS-485 (two wire, half-duplex)
Baud Rates: 110 – 115200, standard baud rates only

Data Bits: 7, **8**
Parity: Even, Odd, **None**
Multidrop Capability: Yes

11.1.2 Communication Functions

Data Types Supported	
Data Type	**Comments**
ASCII	8-bit character
Digital	Digital
Float	32-bit IEEE floating point
Long	Unsigned 32-bit integer
Signed	Signed 16-bit integer
Slong	Signed 32-bit integer
Unsigned	Unsigned 16-bit integer
Input_Reg_64bit	Long input data type
64Bit_Reg	Long data type
Input_Double	Double-precision 64-bit IEEE 754 floating point input data type
Double_Reg	Double-precision 64-bit IEEE 754 floating point data type

Function Codes Supported	
Function Codes	**Description**
01	Read Discrete Output Status (0xxxx)
02	Read Discrete Input Status (1xxxx)
03	Read Output Registers (4xxxx)
04	Read Input Registers (3xxxx)
05	Force Single Coil (0xxxx)
06	Preset Single Register (4xxxx)
15	Force Multiple Coils (0xxxx)
16	Preset Multiple Registers (4xxxx)
17	Report Slave ID

11.2 BACnet IP/ BACnet MSTP/ BACnet Ethernet/ BACnet PTP

The BACnet® suite of drivers is designed to work with the GW BMS Gateway products. One or more drivers using different Data Link Layer options could be configured to act as a gateway between BACnet systems and RTU, SCADA's and PLC's using a wide variety of protocols.

11.2.1 Connection Information

BACnet/IP
Connection Type: Internet Protocol (IP)
Ethernet Speeds Supported: 10Base-T, 100Base-T
BBMD Supported: Yes (not supported on connections where GW BMS Gateway is a client)
Foreign Device Registration: Not supported for client Connections

BACnet MS/TP (Master and Slave Operation)
Connection Type: RS-485 (Two Wire, Half Duplex)
Baud Rates: 9600, 19200, 38400, 76800
Data Bits: 7,8
Stop Bits: 1,2
Parity: Odd, Even, None
Multidrop Capability: Yes

11.2.2 Data Types Supported

GW BMS Gateway Data Types	BACnet Object Type	Optional Properties Supported
AI	Analog Input Object	Reliability, Description
AO	Analog Output Object	Reliability, Description, Max_Pres_Value, Min_Pres_Value
AV	Analog Value Object	Reliability, Description
BI	Binary Input Object	Reliability, Description, Active_Text, Inactive_Text
BO	Binary Output Object	Reliability, Description, Active_Text, Inactive_Text
BV	Binary Value Object	Reliability, Description, Active_Text, Inactive_Text
MI	Multi-state Input Object	Reliability, Description, State_Text
MO	Multi-state Output Object	Reliability, Description, State_Text
MV	Multi-state Value Object	Reliability, Description, State_Text
Device	Device Object	Location, Description, UTC Offset, Active COV Subscriptions, Database Revision
Trend Log		Description, Log_Interval
LSP	Life Safety Point Object	Description

11.2.3 GW BMS Gateway as a 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 MS/TP DLL option.
	Max_Info_Frames	This Property is supported for the BACnet MS/TP DLL option.
	Relinquish_Default	Returns Relinquish_Default.
	Mode	This property is supported.
	Tracking_Value	This property is supported.
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.
	Mode	This property is supported.
	Tracking_Value	This property is supported.
Write Property Multiple	Present Value	Send value in Data Array location after scaling has been applied.

11.2.4 GW BMS Gateway as a Server

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 MS/TP DLL option.
	Max_Info_Frames	This Property is supported for the BACnet MS/TP DLL option.
	Relinquish_Default	Returns Relinquish_Default.
	Mode	This property is supported.
	Tracking_Value	This property is supported.
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	Max_Master	These properties are supported for the BACnet MS/TP DLL option.
	Max_Info_Frames	
	Object_Name	
Write Property Multiple	Max_Master	
	Max_Info_Frames	

11.3 LonWorks

The LonWorks driver allows the GW BMS Gateway to transfer data to and from devices using LonWorks protocol. Data transfer occurs via TP/FT10 twisted pair interface with an exhaustive list of protocols including Modbus, BACnet etc. Data transfer is via 2 basic functional blocks, Input and Output allowing Float and Word SNVT data types.

The GW BMS Gateway LonWorks driver can facilitate up to 4096 Network Variables, which can be of the Standard Network Variable Types (SNVT) and/or User-defined Network Variable Types (UNVT). The GW BMS Gateway LonWorks device can be used with explicit and/or implicit addressing and can be bound to a maximum of 15 other LonWorks nodes. The GW BMS Gateway can handle a maximum of 3000 explicitly addressed nodes. The GW BMS Gateway currently supports a default of 63 network variable aliases to avoid network variable connection constraints.

The GW BMS Gateway can transfer data (Network Variables) in two ways:

- It can poll (request data from) other devices at a regular interval.
- It can send Network Variable Updates:
  - At a regular interval
  - When the data has changed

In throttled mode using minimum and maximum send time and change on delta parameters

11.3.1 Connection Information

Connection Type: FTT-10 Free Topology Network Transceiver
Baud Rates: 78125 bps (bits per second)
Hardware Interface: Built in LonWorks FTT-10 interface
Additional information on cabling and junction boxes that may be used in twisted pair LonWorks networks are detailed in the
following Echelon publication:
http://downloads.echelon.com/support/documentation/bulletin/005-0023-01O_Jbox_wiring.pdf

11.3.2 Communication Functions

Data Types Supported	
GW BMS Gateway Data Types	Description (or Device Data Type)
Integers (long, short, signed, unsigned)	SNVTs* and UNVTs can be presented, stored and moved into any FieldServer data type. **NOTE: For supported SNVTs, see the LonWorks manual.**
Float	
Byte	
Bit	

Read Operations Supported	
GW BMS Gateway as a Client	GW BMS Gateway as a Server
Polled Network variables:	Polled Network variables:
Send Network variable fetch	Respond to Network variable fetch
Send Network variable poll	Respond to Network variable poll

Write (Control) Operations Supported	
GW BMS Gateway as a Client	GW BMS Gateway as a Server
Send Network variable updates	Accept Network variable updates

Unsupported Functions and Data Types	
Function	Reason
Programming Messages	FieldServer is a data transfer device therefore programming messages are not required.
Direct Memory Read / Writes Under User Control	The driver uses the Echelon MIP which handles direct memory read and writes.
LonMark File Transfer Protocol	The driver does not support reading and writing remote configuration properties implemented as files. The driver doesn't support the LonMark File transfer protocol commonly used to access these remote files.

11.4 SNMP-STD

The SNMP-STD driver allows the GW BMS Gateway to transfer data to and from devices over Ethernet using the **SNMP Version V1 or V2c** protocol. The GW BMS Gateway can emulate a Server (SNMP Agent) or Client (NMS Network Management Station).

The GW BMS Gateway provides a generic MIB (Management Information Base) file that sets out the OID (Object Identifiers) structure. The GW BMS Gateway Enterprise ID is 6347. A selection of standard MIB-2 OID's are supported to allow interaction with popular Network Management packages.

When configured as an SNMP Agent (Server) the SNMP-STD driver allows SNMP Get, GetNext (walk) and Set commands to access Data Arrays using the Integer type. The SNMP v1 protocol does not make provision for Floats.

The SNMP-STD driver can send SNMP traps. The structure for SNMP Traps is provided in the GW BMS Gateway's generic MIB file.

The GW BMS Gateway also supports custom MIBs and the automatic generation of the MIB file. It supports setting a custom enterprise ID, object names and custom traps or informs. In custom configurations the GW BMS Gateway supports various data types as specified in the Data Types Supported section.

When configured as a Client, the GW BMS Gateway can read objects from the Server using Get, GetNext (walk) or GetBulk commands. The GetBulk command is very useful to transfer large amounts of data. The GW BMS Gateway can update objects in Agent using the Set command. The GW BMS Gateway can accept any trap or inform as long as all the objects in the message are encoded with a full OID.

11.4.1 Connection Information

Connection Type: Ethernet
Ethernet Speeds Supported: 10Base-T, 100Base-T

11.4.2 Data Types Supported

Many Network Management systems poll these variables to connect to the SNMP Agent.

Standard Configuration	
GW BMS Gateway Data Type	**Description (or Device Data Type)**
Integer	Signed integer (8 - 32 bits)
Octet String	Character strings (0 -255 characters)
TimeTicks	Timer values in 1/100ths of a second

Custom Configuration	
GW BMS Gateway Data Type	**Description (or Device Data Type)**
Integer	Signed integer (8 - 32 bits)
Octect String	Character string (0 -255 characters)
Displaystring	Null terminated character string (0 -255 characters)
Integer32	Signed integer (8 - 32 bits)
Counter	Unsigned integer (8 - 32 bits)
Counter32	Unsigned integer (8 - 32 bits)
Counter64	Unsigned integer (8 - 64 bits)
Gauge	Unsigned integer (8 - 32 bits)
Gauge32	Unsigned integer (8 - 32 bits)
Unsigned32	Unsigned integer (8 - 32 bits)
Bits	32bit integer representing 32 states
Timeticks	Time value in 1/100th of a second

11.4.3 MIB-2 variable supported

Many Network Management systems poll these variables to connect to the SNMP Agent.

OID	Description (or Device Data Type)
1.3.6.1.2.1.1.1	sysDescr
1.3.6.1.2.1.1.2	sysObjID
1.3.6.1.2.1.1.3	sysUpTime
1.3.6.1.2.1.1.4	sysContact
1.3.6.1.2.1.1.5	sysName
1.3.6.1.2.1.1.6	sysLocation

11.4.4 Read Operations Supported

GW BMS Gateway as a Client	GW BMS Gateway as a Server
get-request	get-request
get-next-request/ SNMP Walk	get-next-request/ SNMP Walk
get-bulk-request/ Bulk Walk	get-bulk-request

11.4.5 Write (Control) Operations Supported

GW BMS Gateway as a Client	GW BMS Gateway as a Server
set-request	set-request

11.4.6 Unsolicited Operations Supported

Traps are event notifications and do not require acknowledgements. Inform-Request is a service that keeps sending V2-Traps until events get acknowledged.

Standard Configuration	
GW BMS Gateway as a Client	**GW BMS Gateway as a Server**
N/A	SnmpV1-Trap

Custom Configuration	
GW BMS Gateway as a Client	**GW BMS Gateway as a Server**
SnmpV1-Trap	SnmpV1-Trap
SnmpV2-Trap	SnmpV2-Trap
Inform-Request	Inform-Request

11.5 Ethernet IP

The EtherNet/IP driver allows the GW BMS Gateway to transfer data to and from devices over Ethernet using the EtherNet/IP protocol. The GW BMS Gateway can emulate either a Server or Client. The EtherNet/IP driver uses port 44818 by default.

EtherNet/IP uses CIP (Control and Information Protocol), the common network, transport and application layers also shared by ControlNet and Device Net. EtherNet/IP then makes use of standard Ethernet and TCP/IP technology to transport CIP communications packets. The result is a common, open application layer on top of open and highly popular Ethernet and TCP/IP protocols.

11.5.1 Connection Information

Connection Type: Ethernet
Ethernet Speeds Supported: 10Base-T, 100Base-T

11.5.2 Connection Type Supported

Connection Type	Support Details
Unconnected Messages	Unconnected messages are supported to objects mentioned above.
Explicit Messages	Both client and server support Explicit Messages to all supported objects.
Implicit Messages Using EDS File	Implicit Messages are supported. **NOTE: Electronic Data Sheets (EDS) are simply ASCII files that describe how a device can be used on an EtherNet/IP network. It describes the objects, attributes and services available in the device.**

11.5.3 Data Type Supported

GW BMS Gateway Data Type	Description (or Device Data Type)	GW BMS Gateway Data Type	Description (or Device Data Type)
Identity – Class Code 0x01	**Attributes Supported:** One instance supported (0x01) **Attributes List:** Vendor ID Device Type Product Code Device Revision Status Serial Number Device Description (text) **Services Supported:** Get_Attribute_All Get_Attribute_Single	Register – Class Code 0x07	**Attributes Supported:** Class Instance Support (0x00) **Class Attributes:** 0x02 (Max Instance) Two instances supported (0x01 and 0x02) **Attribute List:** Status Flag Direction (read/write) Size of Data (bits) **Services Supported:** Get_Attribute_Single
Message Router – Class Code 0x02	**Attributes Supported:** One instance supported (0x01) **Attribute List:** Max Connections **Services Supported:** Get_Attribute_Single	Discrete Input Point – Class Code 0x08	No visible interface currently
		Discrete Output Point – Class Code 0x09	No visible interface currently
Assembly – Class Code 0x04	**Attributes Supported:** Class Instance Support (0x00) **Class Attributes:** 0x02 (Max Instance) Two instances supported (0x0100 and 0x0101) **Attribute List:** Member List Not Supported Data **Services Supported:** Get_Attribute_Single	Analog Input Point – Class Code 0x0A	**Attributes Supported:** Class Instance Support (0x00) **Class Attributes:** 0x02 (Max Instance) Two instances supported (0x01 and 0x02) **Attribute List:** Number of Attributes Not Supported Analog value (UINT16) not supported Vendor ID **Services Supported:** Get_Attribute_Single
Connection Manager – Class Code 0x06	Forward Open Service Forward Close Service		
Analog Output Point – Class Code 0x0B	**Attributes Supported:** Class Instance Support (0x00) **Class Attributes:** 0x02 (Max Instance) Two instances supported (0x01 and 0x02) **Attribute List:** Number of Attributes Not Supported Analog value (UINT16) not supported Vendor ID **Services Supported:** Set_Attribute_Single Get_Attribute_Single	EtherNet Link Object – Class Code 0xF6	**Attributes Supported:** One instance supported (0x01) **Attribute List:** Interface Speed Interface Flags Physical Address Interface Counters Media Counters **Services Supported:** Get_Attribute_Single
TCP/IP Interface Object – Class Code 0xF5	**Attributes Supported:** One instance supported (0x01) **Attribute List:** Status Configuration Capability Configuration Control Physical Link Object Interface Configuration Host Name **Services Supported:** Get_Attribute_Single	Data Table Object – Private Object	**Attributes Supported:** This object does not support instances or attributes but uses the data table structure, and associated tags, in Logix5000 PLC's. **Services Supported:** CIP Read Data

11.5.4 Read Operations Supported

GW BMS Gateway as a Client (Scanner)	GW BMS Gateway as a server (Adapter)
Get_Attribute_Single – Service Code 0x0E	Get_Attribute_Single – Service Code 0x0E
Data_Table_Read – Service Code 0x4C	Get_Attribute_All – Service Code 0x01
	Data_Table_Read – Service Code 0x4C

11.5.5 Write (Control) Options Supported

GW BMS Gateway as a Client (Scanner)	GW BMS gateway as a server (Adapter)
Set_Attribute_Single – Service Code 0x10	Set_Attribute_Single – Service Code 0x10
Data_Table_Write – Service Code 0x4D	Data_Table_Write – service code 0x4D

11.6 Metasys N2

The Metasys® N2 by Johnson Controls network supports communications with a diverse range of devices. Many N2 compatible devices use their own version of the protocol and care must be taken to ensure that the device of interest is covered by the GW BMS Gateway implementation.

11.6.1 Connection Information

Connection Type: RS-485 (two wire, half-duplex)
Baud Rates: 9600 (N2 standard)
Data Bits: 8
Stop Bits: 1
Parity: None
Multidrop Capability: Yes

11.6.2 Data Types Supported

Data Type	Description (Device Data Type)
Analog_Input	Analog Input (AI)
Digital_Input	Binary Input (BI)
Analog_Output	Analog Output (AO)
Digital_Output	Binary Output (BO)
Float_Reg	Internal Float value (ADF)
Integer	Internal Integer value (ADI)
Byte	Internal Byte value (BD)

12 Questions and Answers

Q: Incase GW BMS GATEWAY IP Lost, How to find GW BMS GATEWAY IP?
A: Default IP of gateway is 192.168.1.200 subnet 255.255.255.0. If you forgot IP address then use FieldServer toolbox to find Gateway IP. You can download Fieldserver toolbox from below

link.
http://www.santelequip.com/download/BMS%20gateway/Fieldserver%20toolbox/FieldServer-Toolbox-1.05aA-Setup.zip

Q: How to access Web-GUI of BMS gateway?
A: Connect an Ethernet cable between the PC and BMS Gateway or connect the BMS Gateway and the PC to the Hub/switch using a straight CAT5 cable so that PC and BMS Gateway are on same network. Once you get ping to BMS gateway IP, BMS gateway easily accessible through Ruinet or Web-GUI.

Q: What are the maximum connection distance using RS-485 devices?
A: The maximum distance of RS-485 without using a repeater is 4000 feet (1220 meters) at baud rates up to 9600. Extend that distance by adding an RS-485 Repeater or Optically Isolated Repeater every 4000 feet.

Q: How many devices can connect on RS-485?
A: RS-485 devices can be connected to a computer and multi-dropped at various locations in a network style configuration. On RS-485 32 devices can be connecting. To Increase Number of devices more than 32 RS-485 repeater needs to use.

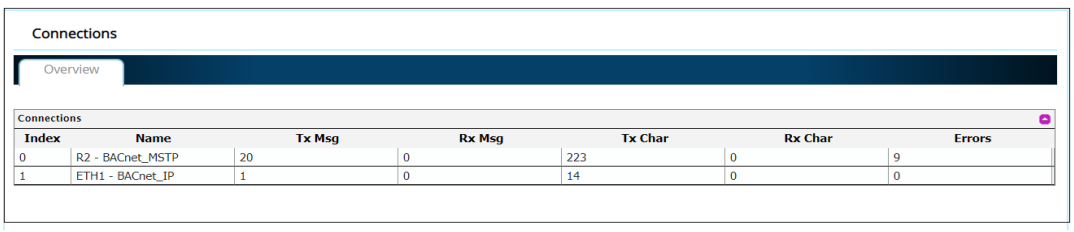
Q: Why do I need a RS-485 repeater for more than 32 devices?
A: A standard RS-485 node has a rated input impedance of 12Kohms. A standard RS-485 transmitter can't drive more than 32 devices and one pair of 120 ohm termination resistors. Simply adding an isolated repeater allows another 32 nodes to be connected.

Q: How do I connect my computer to an RS-485 device?
A: You can use an RS-232 to RS-485 converter or USB to RS-485 converter.

Q: How to Identify Power, RS-485 Connectors on BMS gateway for wiring purpose?
A: All ports related information available on Front side Sticker of BMS Gateway. Always do connections on correct port. There is a possibility that BMS gateway hardware will damage in case of incorrect wiring connections.

Q: What are the 3rd party software's that can use to check various protocols data.
A: We use various software's for testing and troubleshooting purpose. For Modbus Modscan is useful, For BACnet Yabe, BACeye, BDT tool etc. are useful; For SNMP MIB Browser is usefull.

Q: How to Diagnose and Solve BACnet MSTP Connection Problems on a BMS Gateway?
A: Open WEB GUI and go to view



In the screen we see these numbers: Tx Msg = 20, Tx Char = 223 This means that the BMS Gateway is properly transmitting data.
Now see Rx Msg = 0, Rx Char = 0 This shows that zero messages are being received into the BMS Gateway, through BACnet MS/TP.
This is means of the MSTP connection failure. Tx Messages are being transmitted, but there is no Rx (response) Messages being received
Solution: Check communications parameters Baud rate, Parity, Data Bits and Stop bits. Check your BACnet MSTP wiring into the BMS Gateway. Try changing the + and – wires and test again.

Q: How to troubleshoot LonWorks communication problem?
A: Check Neuron ID of LonWorks devices. The Neuron ID of field LON devices should match with IDs configured in CSV file. Check BUS termination as per recommend by Echelon. Check A wire that is loose in a screw terminal? A loose junction where a cable has been spliced or extended.

12.1 Common Mistake made by customer during configuring CSV file

- Use .xls for CSV Configuration which is not supported in GW BMS Gateway for configuration. Always Use .csv file.
- Use old file to upgrade the configuration file. In order to upgrade csv file first upload the CSV file from GW BMS Gateway and the made the changes.
- Adding unsupported description for units in BACnet IP protocol. Refer section 10.4 for description of all the supported units.
- Configured wrong information of slave device which result no communication.