Technical Support
support@nuvationenergy.com

Sales Inquiries
sales@nuvationenergy.com
855-261-0507

NUVATION BMS™

Communication Protocol Reference Guide

2017-12-22, Rev. 2.0

Table of Contents

1. Introduction

Thank you for choosing Nuvation BMS™

The Nuvation BMS™ is an enterprise-grade battery management system with support for various external communication protocols like Modbus RTU, Modbus TCP, and CANBus.

The Nuvation BMS is conformant with the MESA-Device/Sunspec Energy Storage Model. MESA (mesastandards.org) conformant products share a common communications interface that exposes all the data and control points required for operating an energy storage system. This enables the Nuvation BMS to be integrated with other MESA-conformant energy storage hardware or software without the need for custom middleware.

1.1. About this Guide

Nuvation BMS™ implements two standard communication protocols for battery monitoring and control - Modbus and CANbus.

This *Communication Protocol Reference Guide* provides instructions on how to setup and configure your Nuvation BMS to communicate over Modbus RTU, Modbus TCP, or CANBus.

> We thrive on your feedback and what we build is driven by your input. Please submit support tickets to support@nuvationenergy.com.

2. Modbus Protocol Support

2.1. Overview

Nuvation BMS™ implements the SunSpec battery models defined in the Modular Energy Storage Architecture (MESA) as the top-level Modbus interface to the product. These specifications are available for download at http://mesastandards.org/mesa-downloads/. Currently, the draft 3 version is implemented.

A good introduction to system-level MESA concepts can be found here: http://mesastandards.org/downloads/MESA-PCS-Specification_D2.pdf.

Nuvation BMS supports both Modbus RTU and Modbus TCP in the following products:

- Nuvation BMS™ High-Voltage Stack controller
- Nuvation BMS™ Low-Voltage Battery Controller
- Nuvation BMS™ Grid Battery Controller

2.1.1. Modbus RTU

This protocol is used in serial communications. The default configuration is as follows:

- Baud rate: 38400
- Parity: even
- Data bits: 8
- Stop bits: 1

The Modbus RTU slave address must be set through software configuration.

2.1.2. Modbus TCP

This protocol is used for communications over TCP/IP networks. The Stack controller, Battery Controller, Grid Battery Controller support a single Modbus TCP connection over port 502 for read and write access. Additionally, the Grid Battery Controller supports as many as 16 read-only Modbus TCP connections on port 11503.

2.2. Implemented MESA Models

The MESA standards contain a number of 'models' that can be implemented by vendors to describe a storage device at various levels of detail. The models implemented by Nuvation BMS are described below. Detailed register maps are provided by the standard in an Excel spreadsheet format here: http://mesastandards.org/wp-content/uploads/2015/10/Energy-Storage-Information-Models_D3-2015-10-26-Update.xlsx.

2.2.1. Common Model

This model primarily contains information to identify the device (e.g. manufacturer, model, serial number) as well as the version of software running on the device. A full description of the Common Model can be found in the SunSpec specification bundle.

2.2.2. S801

This model describes an energy storage device at the highest possible level. State of charge and overall alarm and warning states are found here. All mandatory points are implemented. The Modbus address of this model is `40070`.

2.2.3. S802

This model describes a battery storage device. At this level, the critical operational information includes the charge and discharge current limits. All mandatory points are implemented. The Modbus address of this model is `40094`.

2.2.4. S803

This model describes a lithium-ion battery in detail. Voltage, temperature, and current statistics are available at the pack and stack level within this model. All mandatory and most optional points are implemented. The Modbus address of this model is `40116`.

2.2.5. End Model

This model marks the end of the implemented Modbus address space.

2.3. MESA Model Structure and Nomenclature

This section is a clarification of terms used to describe a MESA model. It is used to understand the terminology in the spreadsheet presented in the previous section.

2.3.1. Points

All MESA models are a collection of points (i.e. Modbus registers). These points can be one or more Modbus registers in length. By definition, each Modbus register is 16 bits wide. For points that are larger than 16 bits, partial read accesses are not allowed. A Modbus read/write error is returned on such an access.

2.3.2. Fixed/Repeating Blocks

MESA models are described as collections of `Fixed` and `Repeating` blocks of points. A `Fixed` block is a set of points that is always defined and never changes in its size. A `Repeating` block describes a set of related points (i.e. usually for a string of batteries) of which there could be multiple instances of the Repeating blocks. The points within a repeating block are the same but these sets of blocks are concatenated sequentially.

For example in the 803 model, there is a set of repeating blocks that describe data for a particular stack/string of batteries. Accessing the 803 repeating block corresponds to using a stack/string index `(0, 1, 2, …)`` to access the desired repeating block.

For a single stack Nuvation BMS there is only one 803 repeating block. For a multi-stack Nuvation BMS there are 36 stack/string repeating blocks. If a stack/string is configured in the Nuvation BMS software to be installed, then accessing its 803 repeating block will provide a valid Modbus response.

For stack/strings that are not installed, accessing the corresponding repeating block will result in an unimplemented point response. Repeating blocks are taken into account in the length indicated in the model header.

2.3.3. Unimplemented Points

Any MESA point that is not implemented by a vendor will generate an unimplemented response. The response will be a valid Modbus read response but all point data returned will report unimplemented values. The unimplemented values vary by type as listed in the following table.

Table 1. Unimplemented Point Values

Type	Width (bits)	Unimplemented Value (hexadecimal)
signed int	16	0x8000
unsigned int	16	0xFFFF
signed int	32	0x80000000
unsigned int	32	0xFFFFFFFF
enumeration	16	0xFFFF
enumeration	32	0xFFFFFFFF

A write to a writeable MESA point that is unimplemented will generate a Modbus write error.

The Nuvation BMS MESA implementation has unimplemented points as follows:

Table 2. Unimplemented MESA Points

Model	Point Name	Description
801	Evt	Event status bit field
801	DisChaRate	Self Discharge Rate
801	MaxRsvPct	Maximum Reserve Percent
801	MinRsvPct	Minimum Reserve Percent
801	ChaSt	Charge Status
801	DerAlarmReset	Alarm Reset (read is unimplemented)
802	CycleCt	Cycle Count
802	WMaxDisChaRte	NamePlate Max Discharge Rate
802	SoH	State of Health (reported in 803 model per string)
802	BatReqPCSSt	PCS State Request
802	BatReqW	Battery Power Request
802	SoH_SF	State of Health Scale Factor
802	BatReqW_SF	Battery Power Request Scale Factor
803	StrEvt2	String Event 2
803	StrConFail	String Connection Fail

2.3.4. Scale Factors

All MESA points are integer values (signed or unsigned). To account for different range values beyond the data size (i.e. greater than 65535 for an unsigned 16 bit value) or some fractional value (i.e. 1.1), some MESA points have scale factors associated with them. The scale factor is another point within the model which contains a signed integer exponent of base 10 that scales a corresponding point value. For example a scale factor of 2 would result in multiplying the corresponding point by 100. Likewise a scale factor of -3 would result in a scale factor of 0.001. Refer to the prior MESA Draft 3 spreadsheet for the relationship between MESA points and their scale factors. All scale factors are fixed for a model and do not change in value.

2.4. Operational Cases for MESA

There are two main operational cases for the control of a Nuvation BMS over its MESA interface:

1. An external controller (sometimes called "site controller") is used to coordinate power control functions of the Nuvation BMS in conjuction with some other equipment (such as an inverter). This controller requires periodic and rapid responses of MESA point reads as well as some control over the operation of the Nuvation BMS (such as stack connectivity). If there is a loss of communication between this controller and the Nuvation BMS, Nuvation BMS will disconnect the stack(s) as a safety precaution.
2. An owner/operator of a battery system requires control of the Nuvation BMS to monitor the activity of the batteries and track battery usage and its charge/discharge activities. This information can then be used to characterize the usage of the battery management system and to validate battery warranties of a vendor.

Read and write Modbus TCP operations can be performed over the standard Modbus port `502` (only a single connection is supported on this port). In the scenario where the Nuvation BMS MESA interface must be accessed by multiple devices, the Grid Battery Controller provides a secondary port (`11503`) for read-only Modbus TCP operations.

These two operational cases will be discussed in detail in the following sections.

2.4.1. External Controller Communicating Over MESA Interface

An external controller typically polls Nuvation BMS battery control points at a rate of 2-4Hz. This controller reads data points required to manage current flow in the system. The following table summarizes the most important points an external controller may want to read from Nuvation BMS.

Table 3. MESA Points Read by an External Controller

Model	Block	Point Name	Address	Scale Factor	Purpose
801	Fixed	DERHb	40086	No	Nuvation BMS Heartbeat counter incremented every second
802	Fixed	Evt1	40101	No	Bit field of all faults/warnings of a BMS
802	Fixed	Vol	40105	Yes	External DC voltage of the battery system
802	Fixed	MaxBatACha	40106	Yes	Charge current limit calculated by BMS
802	Fixed	MaxBatADischa	40107	Yes	Discharge current limit calculated by BMS
803	Fixed	BTotDCCur	40127	Yes	Total DC current of the battery system

A controller may also want to command Nuvation BMS to perform certain actions, such as connecting/disconnecting all stacks/strings. The following table provides the different writeable points in the MESA interface for different control functions:

Table 4. MESA Points Written to by an External Controller

Model	Block	Point Name	Address	Purpose
801	Fixed	ControllerHb	40087	Heartbeat register for external controller
801	Fixed	DERAlarmReset	40088	Clears all latched alarms in the BMS
802	Fixed	BSetOperation	40110	Commands the Nuvation BMS to connect/disconnect all stacks/strings

When configured, the heartbeat controller can be used to update the watchdog timer of the Nuvation BMS on single-stack and multi-stack implementations. If the heartbeat point is not updated within the watchdog timer period, a fault will be generated on the Nuvation BMS stack(s) and their corresponding contactors will be opened. The value written to the heartbeat point must increase in value and roll over to zero when the 16 bit range limit is reached.

The clearing of alarms of the Nuvation BMS is accomplished by writing a value of `1` to the `DERAlarmReset` point. This point is erroneously identified as read only in the Mesa Draft 3 spreadsheet referenced earlier. Note that alarm will not be cleared if the alarm condition is still present.

Commanding all stacks/strings of a Nuvation BMS system to connect/disconnect is accomplished through the `BSetOperation` point. This point accepts the enumerated values for these connect/disconnect operations. Additional connection logic (such as separate precharge circuitry) is managed automatically by the Nuvation BMS when it is configured for such an operation.

2.4.2. External Nuvation BMS Monitoring Over MESA Interface

An external data logger may want to access a variety of data from the BMS. In general, a data logger will not actively manage a Nuvation BMS; normally, it will not initiate actions such as connecting a battery stack to the DC voltage bus or clearing faults. A data logger should connect to one of the read-only Modbus connections (if available) to allow the writable Modbus connection to be available for separate external control functions. The following table contains the MESA data points exposed by Nuvation BMS that could be collected for logging purposes.

Table 5. MESA Points Read by External Data Logger

Model	Block	Point Name	Address	Scale Factor	Purpose
801	Fixed	SoC	40081	Yes	BMS State of Charge
801	Fixed	DERHb	40086	No	BMS Heartbeat counter incremented every second
802	Fixed	Evt1	40101	No	Bit field of all faults/warnings of a BMS
802	Fixed	Vol	40105	Yes	External DC voltage of the battery system
802	Fixed	MaxBatACha	40106	Yes	Charge current limit calculated by BMS
802	Fixed	MaxBatADischa	40107	Yes	Discharge current limit calculated by BMS
802	Fixed	BSetOperation	40110	No	BMS requested connection state of all stacks/strings
803	Fixed	BConStrCt	40118	No	Number of stacks/strings with contactor closed
803	Fixed	BMaxCellVol	40119	Yes	Maximum cell voltage measured

Model	Block	Point Name	Address	Scale Factor	Purpose
803	Fixed	BMaxCellVolLoc	40120	No	Module/String location of maximum cell voltage
803	Fixed	BMinCellVol	40121	Yes	Minimum cell voltage measured
803	Fixed	BMinCellVolLoc	40122	No	Module/String location of minimum cell voltage
803	Fixed	BMaxModTmp	40123	Yes	Maximum module temperature
803	Fixed	BMaxModTmpLoc	40124	No	Module/String location for maximum module temperature
803	Fixed	BMinModTmp	40125	Yes	Minimum module temperature
803	Fixed	BMaxModTmpLoc	40126	No	Module/String location for minimum module temperature
803	Fixed	BTotDCCur	40127	Yes	Total DC current of the battery system
803	Fixed	BMaxStrCur	40128	Yes	Largest DC current reported by a stack/string
803	Fixed	BMinStrCur	40129	Yes	Smallest DC current reported by a stack/string
803	Repeat	StrSoC	40135+Index	No	State of charge for a stack/string
803	Repeat	StrSoH	40136+Index	Yes	State of health for a stack/string
803	Repeat	StrCur	40137+Index	Yes	Current of a stack/string
803	Repeat	StrMaxCellVol	40138+Index	Yes	Maximum cell voltage of a stack/string
803	Repeat	StrMinCellVol	40139+Index	Yes	Minimum cell voltage of a stack/string
803	Repeat	StrCellVolLoc	40140+Index	No	Location of min/max cell voltages of a stack/string
803	Repeat	StrMaxModTmp	40141+Index	Yes	Maximum module temperature of a stack/string
803	Repeat	StrMinModTemp	40142+Index	Yes	Minimum module temperature of a stack/string
803	Repeat	StrModTmpLoc	40143+Index	No	Location of min/max module temperatures of a stack/string
803	Repeat	StrEvt1	40144+Index	No	Alarms warnings and status bit field of a stack/string

The term Index in the Repeating block addresses used in the above table refers to a calculation of `Index = Stack Index * Length of Repeating block`. By definition, the 803 Repeating block is 16 Modbus registers in length.

2.5. Accessing MESA Models

MESA models are located contiguously in the Modbus address space starting at a base address of `40000`. The Common Model is always located first in this space. The End Model is always last and is used to denote the end of MESA Modbus registers. Each model located between the Common Model and the End Model has a numeric identifier as well as a length. A handy tool that can be used to explore the MESA Modbus registers for Nuvation BMS is `modpoll.exe`. It is available for free download at http://www.modbusdriver.com/modpoll.html.

Using `modpoll.exe`, the Common Model can be polled from a Stack controller or Grid Battery Controller using the following command (assuming the device has an IP address of `192.168.1.21`)

```
modpoll.exe -m tcp -0 -r 40000 -c 70 192.168.1.21

modpoll 3.4 - FieldTalk(tm) Modbus(R) Master Simulator
Copyright (c) 2002-2013 proconX Pty Ltd
Visit http://www.modbusdriver.com for Modbus libraries and tools.

Protocol configuration: MODBUS/TCP
Slave configuration...: address = 1, start reference = 40000 (PDU), count = 70
Communication.........: 192.168.1.21, port 502, t/o 1.00 s, poll rate 1000 ms
Data type.............: 16-bit register, output (holding) register table

-- Polling slave... (Ctrl-C to stop)
[40000]: 21365
[40001]: 28243
[40002]: 1
[40003]: 66
.
.
.
[40068]: 4660
[40069]: -32768
```

As another example, the complete S802 model for a system with one stack could be polled using the following command:

```
modpoll.exe -m tcp -0 -r 40094 -c 22 192.168.1.21

modpoll 3.4 - FieldTalk(tm) Modbus(R) Master Simulator
Copyright (c) 2002-2013 proconX Pty Ltd
Visit http://www.modbusdriver.com for Modbus libraries and tools.

Protocol configuration: MODBUS/TCP
Slave configuration...: address = 1, start reference = 40094 (PDU), count = 22
Communication.........: 192.168.1.21, port 502, t/o 1.00 s, poll rate 1000 ms
Data type.............: 16-bit register, output (holding) register table

-- Polling slave... (Ctrl-C to stop)
[40094]: 802
[40095]: 20
.
.
.
[40114]: -2
[40115]: -32768
To access the common model using Modbus RTU (assuming Nuvation BMS ™ is connected to serial
port COM1 and its address is 0x1):
modpoll.exe -m rtu -0 -r 40000 -c 70 -b 38400 COM1

modpoll 3.4 - FieldTalk(tm) Modbus(R) Master Simulator
Copyright (c) 2002-2013 proconX Pty Ltd
Visit http://www.modbusdriver.com for Modbus libraries and tools.

Protocol configuration: Modbus RTU
Slave configuration...: address = 1, start reference = 40000 (PDU), count = 70
Communication.........: COM1, 38400, 8, 1, even, t/o 1.00 s, poll rate 1000 ms
Data type.............: 16-bit register, output (holding) register table

-- Polling slave... (Ctrl-C to stop)
[40000]: 21365
[40001]: 28243
[40002]: 1
[40003]: 66
.
.
.
[40068]: 4660
[40069]: -32768
```

3. CAN Bus Protocol Support

3.1. Overview

Nuvation BMS uses a flexible CAN reporting implementation which maps BMS registers to CAN message identifiers. The external CAN protocol supports both individual and bulk reporting capabilities. Remote Transmission Requests (RTR) are not supported. The parameters for CAN are:

- Baud: 500 kbit/s
- CAN ID: 11-bit Identifier (Base frame format)
- CAN payload length: variable from 1 byte to 8 bytes based on register size

From a standard configuration, CAN reporting is enabled by editing the following settings:

Table 6. Modifications to Standard Configuration Settings

Register	Setting
sc_canbus.enabled	Set to 1 to enable CAN reporting.
sc_canbus_bulkrpt[0].numtoread	Set to 16x the value of the sc_linkbus.cicount setting.
sc_canbus_bulkrpt[1].numtoread	Set to 8x the value of the sc_linkbus.cicount setting.
sc_canbus.basecanaddress	This is the base CAN identifier. CAN identifiers are assigned sequentially for CAN reporting (see table below). The number of assigned CAN identifiers depends on the system configuration. If the default CAN identifier of 0x100 results in identifier conflicts with other devices on the bus (such as another Nuvation BMS), a different value may be chosen.

Using the standard configuration with the modifications listed above, Nuvation BMS would broadcast the following messages every 500ms.

Table 7. CAN IDs when Using the Standard Configuration

CAN ID	Message	Unit
0x100	Clock	Seconds
0x101	Stack Voltage	mV
0x102	Stack Current	mA
0x103	State of Charge	%
0x104	Depth of Discharge	mAhr
0x105	Maximum Cell Voltage	mV
0x106	Minimum Cell Voltage	mV
0x107	Average Cell Voltage	mV
0x108	Maximum Temperature	C
0x109	Minimum Temperature	C
0x10A	Average Temperature	C
0x10B	Overall Safe	Boolean
0x10C	Safe to Charge	Boolean
0x10D	Safe to Discharge	Boolean

CAN ID	Message	Unit
0x10E	Charge Current Limit	mA
0x10F	Charge Percent Limit	%
0x110	Discharge Current Limit	mA
0x111	Discharge Percent Limit	%
0x112	Stack Control Connection State	Enumeration
0x140 + 1	Cell 0 Voltage	mV
0x140 + 2	Cell 1 Voltage	mV
0x140 + 3	Cell 2 Voltage	mV
…	…	mV
0x140 + (16*N - 1)	Cell (16*N - 1) Voltage	mV
0x140 + (16*N)	Thermistor 0 Temperature	C
0x140 + (16*N + 1))	Thermistor 1 Temperature	C
0x140 + (16*N + 2)	Thermistor 2 Temperature	C
…	…	C
0x140 + (24*N - 1)	Thermistor (8*N - 1) Temperature	C

In the above table, `N` is the value of `sc_linkbus.cicount`. In many systems, not all the channels are enabled (i.e. some values in `cell[0:799].installed` and `therm[0:399].installed` are set to `0`). For example, a system based on sets of 12 cells has `cell[12:15:16:N].installed = 0`. Only the messages associated with enabled channels are broadcast.

Nuvation BMS CAN reporting is highly customizable. Consult the Configuration Settings section of the appropriate Firmware Reference Manual for more information.