NUVATION ENERGY

Technical Support
support@nuvationenergy.com

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

Communication Protocol

Reference Manual

Document ID: NE-RM-005 | Revision: 2.2, 2020-05-06

Table of Contents

1. Introduction

Thank you for choosing Nuvation Energy BMS.

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

Nuvation Energy BMS is conformant with the MESA-Device/Sunspec Energy Storage Model (Draft 3). MESA (http://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 Nuvation Energy 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 Energy BMS implements two standard communication protocols for battery monitoring and control - Modbus and CANbus.

This *Communication Protocol: Reference Manual* provides instructions on how to setup and configure Nuvation Energy 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.

1.2. Units

A standard set of types/units has been adopted for use within Nuvation Energy BMS for the measurements and configuration settings. Unless otherwise noted, the units used within the firmware should be assumed as defined below.

Table 1. Standard Types and Units

Type	Units	Variable Type	Application
Address	none	uint32	BMS register address
Boolean	false=0 true=1	uint8	Logic value
Charge	mAh	int32	Depth of discharge and throughput
Count	number	uint32	A postive count of some event
Crc16	none	uint16	Used to verify persistent data
Current	milliamperes	int32	Stack and pack currents
Energy	Wh	uint32	Energy capacity of a stack
Int64	value	int64	Signed 64 bit integer
ImpedanceMili	mOhms	uint32	Resistance of a stack
IpAddress	IP	uint32	IP4 address for a stack
LogLevel	none	uint32	Log level applied the BMS firmware
MicroOhms	μOhms	uint32	Resistance of a cell
Microseconds	microseconds	uint64	Time measurement or period
Percentage	%	uint8	Percentage of a full scale value (e.g. SoC)
PowerW	W	int32	Rated power of a stack
SoftwareId	enum	uint32	Enumeration for type of Cell Interface
Temperature	degrees Celsius	int16	Thermistor temperatures
String	none	char[8]	Storage for 8 characters
UInt16	value	uint16	Unsigned 16 bit integer
UInt64	value	uint64	Unsigned 64 bit integer
Voltage	millivolts	int32	Cell and stack voltages

Every register within the firmware has an associated type that defines the expected units for that register.

A negative value for *Current* and *Charge* represents a charge direction while a positive value represents a discharge direction with respect to the cell/stack.

2. Modbus Protocol Support

2.1. Overview

Nuvation Energy BMS implements the SunSpec battery models defined in the Modular Energy Storage Architecture (MESA) as the top-level Modbus interface to the product. Specifically the BMS implements the MESA Draft 3 Storage models (800 Series)

These specifications are available for download at http://mesastandards.org/mesa-device.

> MESA Draft 3 is currently implemented on Nuvation Energy BMS. Be sure to download the correct draft of the standard from the MESA website.

A good introduction to the benefits of the Open MESA standards is summarized here: http://mesastandards.org/why-mesa.

A system-level summary of all of the MESA standards can be found here: http://mesastandards.org/mesa-standards.

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

- High-Voltage Stack Controller
- Low-Voltage Battery Controller
- Nuvation Energy 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. The default slave address value is 1.

> A slave address of zero is used as a broadcast address and should not be used as the Modbus RTU slave address.

2.1.2. Modbus TCP

This protocol is used for communications over TCP/IP networks. Stack Controller, Battery Controller, and Grid Battery Controller support a single Modbus TCP connection over port 502 for read and write access. Additionally, 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 Energy BMS are described in the sections below.

Detailed register maps for all Draft 3 models are found in the document *MESA-Energy-Storage-Information-Models_Draft_3.xlsx* found at the Nuvation Energy technical resources located at https://www.nuvationenergy.com/technical-resources.

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

For stacks or 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 2. 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 MESA implementation on Nuvation Energy BMS has unimplemented points as follows:

Table 3. 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

Model	Point Name	Description
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 Nuvation Energy BMS over its MESA interface:

1. An external controller (sometimes called "Energy Storage Controller") is used to coordinate power control functions of the BMS in conjunction 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 BMS (such as stack connectivity). If there is a loss of communication between this controller and the BMS, the BMS will disconnect the stack(s) as a safety precaution.
2. An owner/operator of a battery system requires control of the 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 MESA interface must be accessed by multiple devices, 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 Energy 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 the BMS.

Table 4. MESA Points Read by an External Controller

Model	Block	Point Name	Address	Scale Factor	Purpose
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
803	Fixed	BTotDCCur	40127	Yes	Total DC current of the battery system

A controller may also want to command Nuvation Energy 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 5. 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 Nuvation Energy BMS to connect/disconnect all stacks/strings

When configured, the heartbeat controller can be used to update the watchdog timer of Nuvation Energy 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 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 Nuvation Energy 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 Energy 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 pre-charge circuitry) is managed automatically by the BMS when it is configured for such an operation.

2.4.2. External Nuvation Energy 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 Nuvation Energy 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 the BMS that could be collected for logging purposes.

Table 6. 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
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

Model	Block	Point Name	Address	Scale Factor	Purpose
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 Energy 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 Energy 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 Energy BMS uses a flexible CAN reporting implementation which maps BMS software registers to CAN message identifiers. The interface is provided on the Nuvation Energy High-Voltage BMS and Nuvation Energy Low-Voltage BMS products, but not on the Grid Battery Controller. It is based on the CAN 2.0 standard, with the exception that Remote Transmission Requests (RTR) are not supported.

The parameters for the CAN interface 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

3.2. CAN Reporting and Commands

The CAN interface provides the following capabilities through configuration:

- Reporting of up to 64 individual registers within the battery management system.
- Bulk-reporting of registers that represent sets of measurements (i.e. cell voltages and temperatures)
- Creation of customizable commands to the battery management system to implement actions such as:
  - connecting/disconnecting the battery to the DC bus
  - clearing faults and/or warnings
  - updating the controller heartbeat (i.e. watchdog)

These capabilites are described in detail within the `CAN Bus` configuration section of the *Software Reference Manual*.

3.3. Data Format

The protocol is based on the CAN message frames. A reference for the CAN message frames can be found at https://en.wikipedia.org/wiki/CAN_bus.

3.4. Data Size

The size of the register data transmitted/received in a CAN frame will depend of the *variable type* of the register. The *type* and corresponding *variable type* for all registers used in battery management system can be found in Section 1.2. For example a variable type of `int32` represents a signed 32 bit integer and is 4 bytes in length. All register types can be read from the Registers page as discussed in the Operator Interface Manual.

From time to time Nuvation Energy will make updates to Nuvation Energy BMS in response to changes in available technologies, client requests, emerging energy storage standards, and other industry requirements. The product specifications in this document, therefore, are subject to change without notice.