NUVATION ENERGY

Technical Support
support@nuvationenergy.com

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

NUVATION BMS™

Grid Battery Controller

Software Reference Manual

2017-12-22, Rev. 1.2

Table of Contents

Introduction

This manual describes the software configuration settings used within your Nuvation BMS ™ Grid Battery Controller (GBC). The GBC is used in conjunction with multiple Stack Controllers or Battery Controllers to provide pack-level monitoring and aggregation for multi-stack battery packs.

This document is designed as a companion to the example configuration file provided by Nuvation. The sections in this document are designed to correspond in order and content to the layout of the example configuration file. This enables efficient cross referencing between the descriptions in this document and actual configuration examples.

Note: This document applies to the Nuvation BMS™ Ampere 17.12 release (firmware version 4.79.0). Content may be inaccurate or incomplete for other versions.

Background

Terminology and technical concepts critical to the operation and configuration of Nuvation BMS ™ are presented in this section.

Battery Topology

Energy Storage Systems are hierarchical in nature. Nuvation Energy has adopted the following definitions for battery pack topology:

Cell

A Cell is the smallest unit of energy storage distinguishable by the BMS. One Cell, as defined from the perspective of the BMS, may actually consist of one or more electrochemical cells connected in parallel. This subtlety is reflected in the nomenclature for completeness. For example, a "1p" Cell refers to a single electrochemical cell, while a "2p" Cell refers to two electrochemical cells connected together in parallel. From the perspective of the BMS, these topologies appear identical except for the capacity of the Cells.

Group

A Group is a set of Cells connected in series and managed together. For example, 12 "1p" Cells in series are referred to as a "12s1p" Group, while 16 "2p" Cells in series are referred to as a "16s2p" Group. Grouping of Cells is highly application-specific and is defined in how BMS hardware interfaces are physically wired up to Cells.

Stack

A Stack is one or more Groups connected in series. For example, five "14s2p" Groups connected in series are referred as a "5g14s2p" Stack. This Stack may also be described as a "70s2p" Stack.

Bank

A Bank is one or more stacks connected in parallel. For example, three "5g14s2p" Stacks are referred to as a "3x5g14s2p" Bank or simply a "3x70s2p" Bank.

Pack

A Pack is one or more Banks connected in series.

Register Data Model

Nuvation BMS™ implements all data storage and processing using two important software building blocks

Register

A register is the fundamental unit of data storage within the system. Each register has a unique name and associated type that defines how the value is interpreted. Registers range in size from as small as one byte up to as large as eight bytes.

Component

A component combines a set of related registers with processing rules that operate on those registers to implement a particular BMS function for the system. A given component may have many instances throughout the system. In this case, its associated registers will have the same number of instances.

Complex behavior within the system is achieved by connecting multiple components together, either through configuration or through hard-wired connections in the firmware itself.

Configuration for a system is completely determined by the state of all configuration registers present within the system. Configuration registers are persisted in non-volatile memory and are loaded automatically upon reset.

External protocols are implemented by mapping (and in some cases aggregating) the appropriate BMS registers to CAN bus messages or Modbus registers.

Index versus Location

Internally to the BMS firmware, all register indexing is zero-based. That is, if multiple instances of the same register are present, the first instance is always indexed at zero. This convention is reflected in all register expressions and configuration files.

Operator-facing tools such as the Operator Interface or the MESA Modbus models use one-based location identifiers to refer to physical, countable entities. For example, the location

of the first cell within a stack is defined as CI 1, Cell 1 (it is the first cell in the first CI). The index of this first cell is defined as zero within the firmware.

Register Expressions

Registers are accessed by name in Nuvation BMS™ tools and configuration. Each register also has a unique address that is used internally within the BMS.

Single Register Instance

This expression is used when assigning to or reading from a single register in the system and is of the form

```
component_name.register_name
```

Where

- component_name is the name of the component within the system
- register_name is the name of the register within the component

Range of Register Instances

These expressions build on the single register references by adding an additional range expression in square brackets

```
component_name[range_expression].register_name
```

The range_expression may take any of the following forms

- `index` - a single instance of component_name.register_name at index. Note that `cell[0].voltage` is equivalent to `cell.voltage`.
- `start_index:end_index` - all instances from start_index through end_index. The expression `cell[0:3].voltage` expands into

  ```
  cell[0].voltage
  cell[1].voltage
  cell[2].voltage
  cell[3].voltage
  ```

- `start_index:end_index:block_length` - all instances from start_index through end_index within a repeating block of block_length across all instances of the register. The expression `cell[0:3:16].voltage` expands into

  ```
  cell[0].voltage
  cell[1].voltage
  cell[2].voltage
  cell[3].voltage
  cell[16].voltage
  cell[17].voltage
  cell[18].voltage
  ```

```
cell[19].voltage
cell[32].voltage
cell[33].voltage
cell[34].voltage
cell[35].voltage
cell[N-16].voltage
cell[N-15].voltage
cell[N-14].voltage
cell[N-13].voltage
```

where N is the total number of instances of the register cell.voltage within the system.

- `start_index:end_index:block_length:block_count` - all instances from start_index through end_index within a repeating block of block_length repeated block_count times. The expression `cell[0:3:16:2].voltage` expands into

```
cell[0].voltage
cell[1].voltage
cell[2].voltage
cell[3].voltage
cell[16].voltage
cell[17].voltage
cell[18].voltage
cell[19].voltage
```

In this case, only the first 2 blocks of 16 instances are included, rather than all blocks of 16 instances.

All Register Instances

A compact syntax can be used to expand to all instances of a given register within the system. The expression

```
component_name[*].register_name
```

expands to all instances of component_name.register_name within the system. For example, the expression `cell[*].voltage` expands into

```
cell[0].voltage
cell[1].voltage
cell[2].voltage
cell[3].voltage
cell[4].voltage
cell[5].voltage
cell[6].voltage
cell[7].voltage
cell[8].voltage
cell[9].voltage
cell[10].voltage
cell[11].voltage
```

```
cell[12].voltage
cell[13].voltage
cell[14].voltage
cell[15].voltage
cell[16].voltage
.
.
.
cell[N-3].voltage
cell[N-2].voltage
cell[N-1].voltage
```

where N is the total number of instances of the register cell.voltage within the system.

Register Address

In some cases, it is necessary to use the address of a register as a configuration value for another register in the system. This is required when assigning input and output pins to functions within the BMS, for example.

The expression

```
@component_name.register_name
```

expands to the address of the register in the system. The single-instance range expression may be used for register addresses. For example

```
@component_name[index].register_name
```

expands to the address of component_name.register_name at index in the system.

Configuration Files

Configuration is stored externally to Nuvation BMS™ in a plain-text file. This file defines the state of configuration registers as required for a particular system.

The Operator Interface provides tools for uploading and downloading configuration files to and from Nuvation BMS™ as a way to set or retrieve the state of all configuration registers.

The configuration file format is plain ASCII text with the following syntax

- Any lines starting with a leading '#' are treated as comments.
- Each non-comment line is treated as a register assignment statement.

A register assignment takes on the form `register_expression = value` where

- register_expression is one of the valid Register Expressions previously defined.
- value is either a numerical constant, quoted string, IP address, or a valid Register Address.

Any standard text editor can be used to edit configuration files (e.g. Windows Notepad, Notepad++, etc.).

Configuration Settings

This section highlights the most important settings required to achieve a working Grid Battery Controller (GBC) configuration for your system.

To operate your GBC in a pack containing one or more Stack Controllers or Battery Controllers, each SC or BC must meet the following requirements:

- It must be running the same firmware version as the GBC you are using.
- It must be configured for the battery stacks in use in the pack.
- It must be assigned a unique network address on the ETH1 (Internal) GBC network.

Please refer to the *Single Stack Operator Interface Manual* for instructions on upgrading, configuring, and assigning addresses at the stack level. Refer to the *Multi-Stack Operator Interface Manual* for instructions on upgrading firmware or loading configuration for a Grid Battery Controller.

The most important aspects of a GBC configuration file are discussed in the following sections.

GBC Network Settings

The Ethernet IP addressing used by each stack in the pack must match the network configuration of the GBC.

The recommended network topology has each SC or BC connected to an Ethernet switch. This switch is then connected to the GBC on the internal Ethernet port (labeled ETH1).

The GBC ETH1 interface has the following settings by default:

- IP: 192.168.4.10
- Subnet: 255.255.255.0
- Gateway: 192.168.4.1

Each stack is assigned a static IP address on the GBC internal 192.168.4.xxx subnet. The GBC Ethernet bridges must be configured to match this IP address assignment. There is one instance of the Ethernet bridge component for each stack in the pack.

gbc_eth_bridge.ip_address

- The IP address of the stack. This is set to the SC or BC IP address.

gbc_eth_bridge.port

- The port of the stack request bridge. This should be set to 8080.

gbc_eth_bridge.publish_port

- The publish port of the stack publish bridge. This should be set to 8081.

Pack Topology

Battery pack topology configuration is required to specify which stacks are connected in each bank as well as the number of banks within the pack. The following registers must be configured for each stack in the pack (the pack_stack component instances correspond in order to the gbc_eth_bridge component instances).

pack_stack.installed

- Flag indicating if this stack is installed. Set to 1 if the stack is present.

pack_stack.enabled

- Flag indicating if this stack is enabled. Set to 1 to actually use the stack.

pack_stack.bank_index

- The index of the bank where the stack is located. Set to 0 if only one back is in use.

Stacks that are installed are considered physically present in the system. The GBC will attempt to connect to and communicate with the SCs for these stacks using the corresponding gbc_eth_bridge configuration.

A stack must be enabled before the GBC will aggregate that stack's data into pack-level view or connect the stack to the DC bus.

Control Settings

The GBC includes a pack connection routine that sequences the connection and disconnection of all enabled stacks in a pack. A configurable delay period is used when connecting multiple stacks to the DC bus. This is the settling time required after one stack is connected but before the next stack is requested to connect. Configuration of this connection routine is achieved through the following registers.

pack_control.enabled

- Set to 1 to enable automatic control of stack connection
- Set to 0 to require manual control of stack connection

pack_control.stack_delay

- Set to the settling delay in microseconds

By default, this feature is disabled. When disabled, stacks must be connected and disconnected individually through the Operator Interface or via one of the supported Communication Protocols.

Warning: The GBC pack connection routine currently does not perform any validation of stack voltages or SOC before connecting stacks. Unless each stack is using a precharge circuit designed with this behavior in mind, use of the pack connection routine is not recommended.

Operational Limits

The operational limits of the pack are primarily defined by the configuration at the stack level. The GBC aggregates the limits and signals from each stack to produce overall pack-level limits and signals automatically.

A number of additional triggers are present within the GBC itself that define fault thresholds at the pack level.

Communication Safety

Pack level communication must be stable and error free to ensure the battery is safely operating. Faults are tripped if any of the installed Ethernet bridges experience connectivity issues. Fault behavior is tuned using the following trigger registers.

gbc_fault_eth_con.thresh
- The threshold that is tripped when a bridge fault occurs.

gbc_fault_eth_con.time_hyst
- The microsecond duration that the fault must be present before it trips.

gbc_fault_eth_con.end_time_hyst
- The microsecond duration that the fault must be cleared before it resets.

Stack Watchdog

The GBC will periodically update the watchdogs timers (sc_wdt) of all stacks that are connected to the GBC. If an SC becomes disconnected from the GBC, this local SC watchdog can be configured to trip a fault at the stack level.

gbc_sc_wdt_reset.period
- The period at which all connected SC watchdogs will be reset.

gbc_sc_wdt_reset.enable
- Set this to 1 to enable the update of stack watchdogs by the GBC.

Communication Protocols

The Grid Battery Controller supports the following interfaces for connection with external systems:

- 10/100/1000 Ethernet for Modbus TCP and Operator Interface connectivity

- RS-485 for Modbus RTU

RS-485 Modbus RTU

The slave device address used by the BMS for Modbus RTU may be customized as required.

gbc_modbus_rtu.device_address

- Set to the desired Modbus RTU slave device address