Battery Management System

For Lithium Ion Batteries Up to 75V/100A

RoboteQ
BMS1040BT
Battery Management System
200A Peak, 100A Continuous Load
50A Max Charge - Bluetooth
Max Voltage: 45V

RoboteQ’s BMS10x0x is a battery management and protection system for building cost-effective, ultra-efficient and high current power sources using Lithium battery cells.

The BMS connects to an array of battery cells at one end, and to a user load at the other. Available in a 40V, 60V and 75V versions, it is optimized for 6-cell to 15-cell battery packs.

Precision voltage sensors monitor the voltage of every cell. Precise, built-in current sensors keep track of the current flowing in and out of the pack, load and charge, maintaining an accurate image of the battery’s State of Charge and State of Health. Balancing takes places during battery charge.

The BMS includes a separate connector for battery charging. The charger’s voltage and current are monitored and an internal switch will automatically stop the charging when the battery is full or if a fault condition occurs. This allows the load to be controlled separately.

The BMS10x0x supports regenerative loads, such as motor controllers in electric vehicles or mobile robot application. A built-in, bidirectional, high power solid-state switch will disconnect the regenerative current source from the cells in case the later are fully charged and cannot accept more current. A secondary, built-in high power switch will connect the auxiliary power port to an optional external brake resistor, in case of braking while the batteries are full.

Charge/discharge current, voltages, internal and battery temperatures are continuously monitored. The internal switch will disconnect the load in case of overload, over or under voltage, or over temperature. The BMS will act as a programmable circuit breaker in case of a short circuit or continuous overload. The BMS is highly configurable: temperature limits, voltage thresholds, and current overload levels can all be set and changed using a provided PC Utility. The utility can also be used to view and chart in real time the BMS internal parameters while the system is running. This provides the user with useful data during system development and tuning.

State of Charge, voltage, current, temperature and other vital parameters are available on a front connector using selectable communication methods. This includes USB, CANbus, RS485 and PWM, or wirelessly via an optional Bluetooth module. A push button and LEDs are also provided for quick visualization of battery charge level and system status.

Applications

- Personal mobility system
- Mobile robots, ROVs and AGVs
- Electric vehicles, scooters, e-bikes
- Electric boats
- Electric forklifts
- Storage for wind/solar generators
- Telecom tower backup

Features List

- Configurable for all Lithium battery types
- Can manage 6 to 10 cells (40V) or 11 to 15 cells (60V/75V) versions
- Centralized design. No cell boards – All BMS electronics contained within the unit
- 100A/75A/100A (40V/60V/75V versions) max current to load.
- Automatic, intelligent cell voltage balancing during charge
- Advanced State of Charge monitoring and management for optimal battery life
- State of Health monitoring and reporting

- Built-in bi-directional current sensors for monitoring flow from/to battery pack, load and charger.
- Built in temperature sensor. Connector for up to three external temperature sensors
- Built-in 100A/75A/100A (40V/60V/75V versions) switch with under voltage and overvoltage Shutoff
- A built-in circuit breaker function in case of short circuit and continuous overload
- Separate independent input and internal switches for charger
- Connection to optional brake resistor, heater or to other high current accessory
- Two 1A open drain Digital Outputs for activating Fan, Cooling Circuit, Contactor or to another low current device
- CANbus, RS485, or PWM Output for communication of battery data and system status
- Optional Bluetooth transceiver for Battery Monitoring via Smartphone or PC Utility
- LED indicators for status and battery charge indication
- High current screw terminals to battery and load
- Automotive grade, locking connectors to battery cells and to the I/O
- Built in clock/calendar for data logging time stamp
- Built-in programming language for scripting advanced power management strategies
- Direct, one-wire interface with RoboteQ motor controllers
- USB connector and PC Utility for setup, testing and monitoring
- Ultra-Low power consumption. Will not drain batteries during storage and stand-by
- Aluminum bottom plate for conduction cooling with ABS cover
- Field upgradeable firmware
- Dimensions: L: 145mm x W: 115mm x H: 23mm
- Weight: **250g**

Ordering References

Part Number	Number of Cells	Max Voltage	Bluetooth
BMS1040x	6 - 10	40	No
BMS1040xBT	6 - 10	40	Yes
BMS1060	11 - 15	75	No
BMS1060BT	11 - 15	75	Yes
BMS1060A	11 - 15	75	No
BMS1060ABT	11 - 15	75	Yes

Revision History

Date	Version	Changes
April 14,2020	1.2	New BMS10x0A features added
May 23, 2019	1.1	Minor fixes
	1.0	Initial release

System Architecture

The BMS10x0x uses an ultra-low power consumption, 32-bit Processor to monitor voltages, amperage and temperature (shown as V, A and T, as seen in the diagram in Figure 1, below) to control the internal balancing array and solid-state switches. The balancing array is designed to extract charge from multiple higher voltage cells during the charge cycle and smoothing the uneven stresses on the battery. A power management block ensures that any system or subsystem is turned off when not needed. This results in the lowest possible power consumption in order to keep the battery from needlessly discharging when not used.

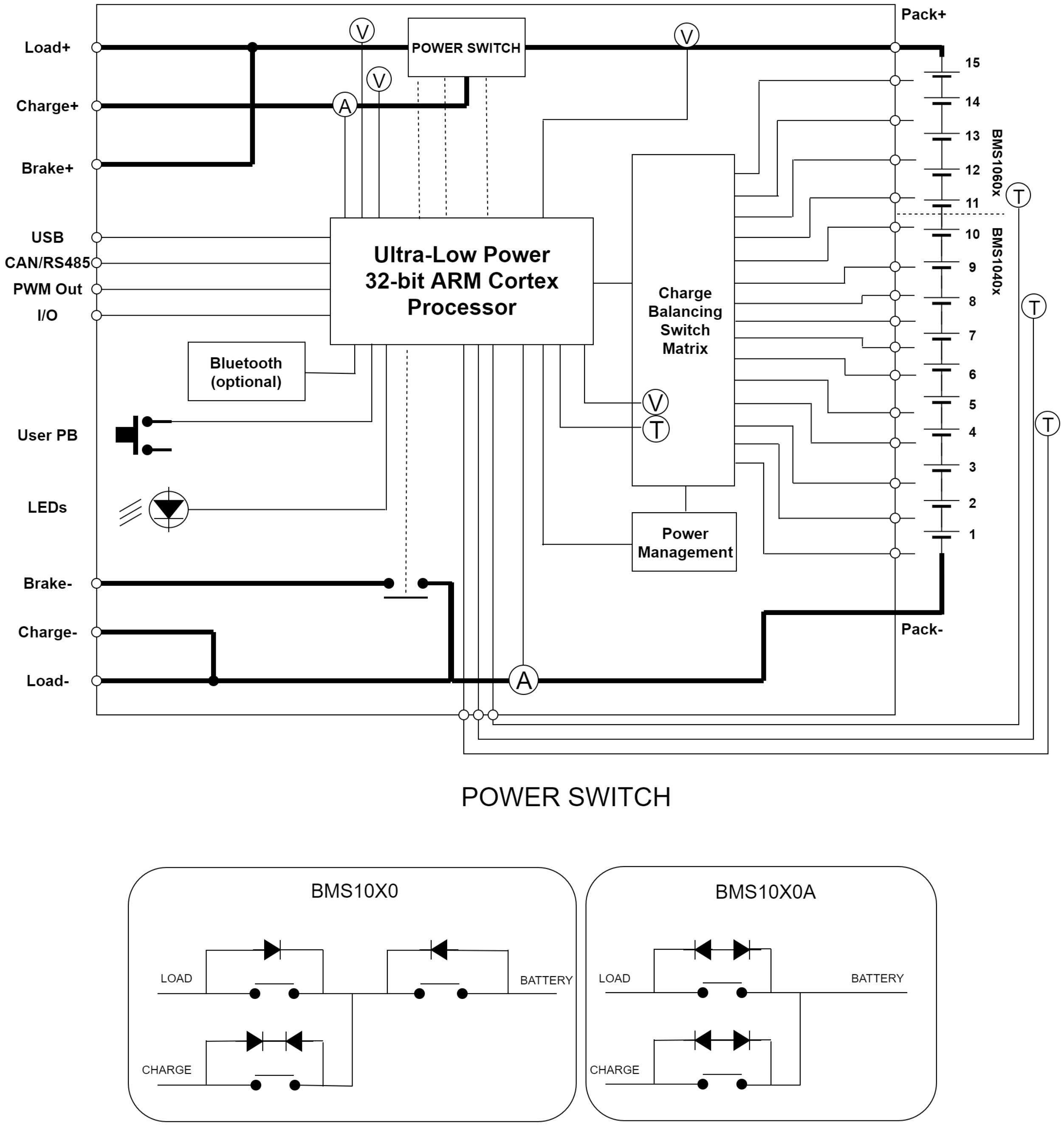


FIGURE 1. Balancing and Switches

Important Warning

Please make sure that before any firmware update only the USB, and the Battery Cell Connector is attached to the device for BMS1040 and BMS1060 only, not the "A" models.

User Connections

The BMS electronics board is mounted on top of an aluminum cooling plate and under an ABS plastic cover. Figure 2, below, shows the position of its connectors.

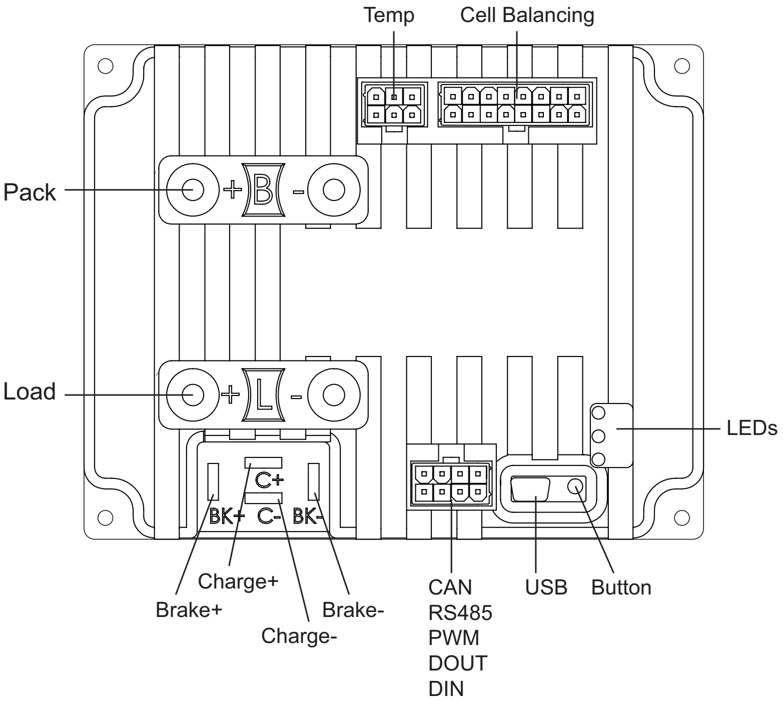


FIGURE 2. BMS Board Connectors

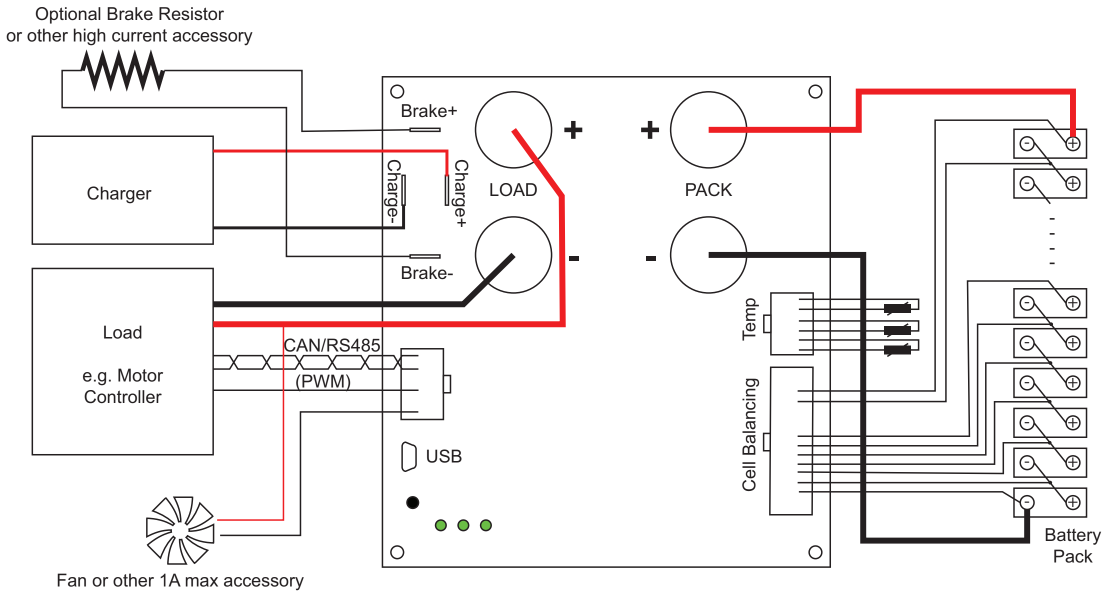


FIGURE 3. BMS Connections and Power Flow

Load+/Load-Power Connectors to Load

Two high current screw terminals are used to connect to the user load. A 75 A for BMS1060, a 100A for BMS1060A and 100A for BMS1040x switch built into the BMS will isolate these connections from the battery in case of over/under voltage, short circuit, over current, over temperature or other unsafe conditions. Current is allowed to flow back into the battery from regenerative loads if operating in the safe operation area. The power connectors support M5 x 10mm-14mm screws.

Charge+/Charge-Connectors to Battery Charger

Two Faston connectors are used to connect the BMS to a charger. A 35 A for BMS1060 and 50 A for all other models of BMS built-in switch will disconnect the charger from the battery when it is fully charged or if any unsafe condition occurs. The Faston connectors are 0.250" (6.35 mm) wide and 0.032" (0.81 mm) thick.

Brake+/Brake-Connectors to External Resistor

These Faston connections can be used to connect a brake resistor to absorb regenerative current from the main connectors when the batteries are already fully charged or an unsafe condition has occurred. The brake resistor is externally connected by the user. Alternatively, these connections can be used for attaching other high-current elements like a battery heater or fans. The Faston connectors are 0.250" (6.35 mm) wide and 0.032" (0.81 mm) thick.

User I/O

An 8-pin connector, as shown in Figure 4, contains the signal needed by the BMS to communicate with other parts of the system. Use Molex 2 rows x 4-pin, 4.2mm pitch connector reference 39-01-2080, or equivalent.

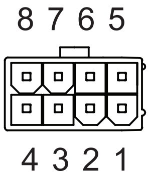


FIGURE 4. User IO Male Connector on BMS

TABLE 1. User IO Pin Signals

Pin	Signal	Models	Description
1	CANH / RS485A	All models	High level CAN bus line / RS485 Direct (+)
2	CANL / RS485B	All models	Low level CAN bus line / RS485 Inverted (-)
3	DOUT1	All models	1A Open Drain output 1
4	DOUT2	All models	1A Open Drain output 2
5	VCC	10x0	Optocoupler VCC input - 3.3V to 5V typical operation
		10x0A	5V Output/ 15mA for external wake up signal*
6	PWM	10x0	Optocoupled PWM output
		10x0A	Non-Optocoupled PWM output
7	GND	10x0	Optocoupler GND in
		10x0A	Non-Optocoupled GND in
8	DIN2	10x0	Optocoupled Digital Input
		10x0A	Non-Optocoupled Digital Input
* changed Vcc input of user connector to 5V out, with 15mA current limitation in order to drive directly the wakeup through an external mechanical switch.			

The CAN and RS485 pins can be connected to a twisted pair cable. The BMS can be configured to operate in either of these communication modes.

Two Open Drain Digital Outputs are available for connecting low-power accessories.

One digital input (DIN2) is available on the user connector. The DIN2 is electrically isolated from the BMS board only for 10x0 models by an optocoupler. The digital input available on the user connector is defined as DIN2 because the on board user button is defined as DIN1.

The BMS10x0x also implements a RoboteQ proprietary single wire PWM communication protocol for exchanging the most vital battery information with RoboteQ's motor controllers. For BMS10x0 when using the PWM mode, external ground and 5V supply must be connected to the GND and VCC pins. (See MultiPWM in BMS operation)

Push-Button and LEDs

The push-button is considered to be the digital input 1 (DIN1) of the BMS system and its action is configured by default as Operating Mode Control. With this function, the user is able to switch the BMS's mode of operation between Active, Idle and Storage mode.

USB Connector

A USB port is present for connecting the BMS to a PC running the dedicated PC Utility for configuring the BMS's various parameters (voltage, current, temperature thresholds, etc.), monitoring real-time operating values, capture/save logs and editing user scripts.

Pack+/Pack-Battery Connection

The Pack+ and Pack- are high-current screw terminals connected to the battery pack's mains; positive and negative. Current up to 100A can be transferred to the battery during charge or regeneration and from the battery during discharge. A 16-pin connector on the BMS10x0x is used for connecting each individual cell's terminals for monitoring and balancing.

Temperature Thermistor Connector

The 6-pin connector provides the connections to up to three external 10K NTC thermistors for monitoring temperature in the battery pack. Thermistors can be connected without attention to polarity. Use Molex 2 rows x 3, 4.2mm pitch connector reference 0039012060, or equivalent. For recommended NTC model see Page 18 - Temperature Sensors.

Thermistor Connector Pins

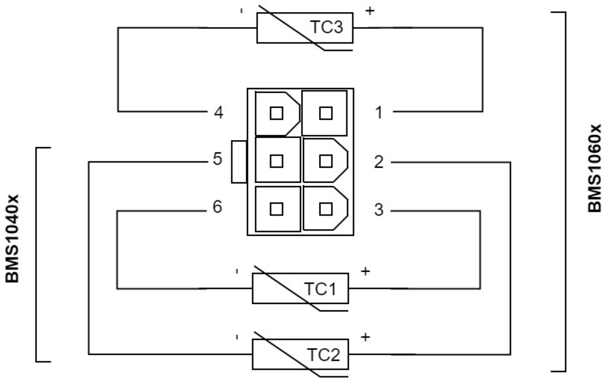


FIGURE 5. Temperature Thermistor connections

TABLE 2. Thermistor Inputs

	Signals		
Pin	**BMS1040x**	**BMS1060x**	**Description**
1	NC	TC3P	Thermistor 3 terminals
4	NC	TC3N	
2	TC2P	TC2P	Thermistor 2 terminals
5	TC2N	TC2N	
3	TC1P	TC1P	Thermistor 1 terminals
6	TC1N	TC1N	

Cell Connections

In order for the system to successfully measure, monitor and balance each individual cell in the battery pack, the cells must be properly connected to the BMS.

Cell Chemistry and Voltages

Table 3, below, shows the expected output voltage available for the load, at different States of Charge, for different cell arrangements and chemistry. Each of these values can be adjusted in the BMS configuration to different operating/charge conditions.

TABLE 3. Number of Cells vs Voltage

	LiPo			**LiFe**			
	Minimum	**Nominal**	**Maximum**	**Minimum**	**Nominal**	**Maximum**	
Cells	**Empty**	**80%-20% SoC**	**Full Charge**	**Empty**	**80%-20% SoC**	**Full Charge**	
6	18	22.2	25.2	15	19.8	21.6	**BMS1040x**
7	21	25.9	29.4	17.5	23.1	25.2	
8	24	29.6	33.6	20	26.4	28.8	
9	27	33.3	37.8	22.5	29.7	32.4	
10	30	37	42*	25	33	36	
11	33	40.7	46.2	27.5	36.3	39.6	**BMS1060x**
12	36	44.4	50.4	30	39.6	43.2	
13	39	48.1	54.6	32.5	42.9	46.8	
14	42	51.8	58.8	35	46.2	50.4	
15	45	55.5	63	37.5	49.5	54	
Voltages per cell	**3.0**	**3.7**	**4.2**	**2.5**	**3.2**	**3.65**	
***Warning: Only BMS1040A can hadle 10cell LiPo Batteries with state of charge up to 100%**							

Connecting the cells

A 16-pin connector is used for the necessary monitoring and balancing connections. Use Molex 2 rows x 8-pin, 4.2mm pitch connector reference 39-01-2160, or equivalent. The drawings in Figures 6-8, below, show how to connect 6 to 10 cell packs on the BMS1040x, and 11 to 15 cells packs on the BMS1060x.

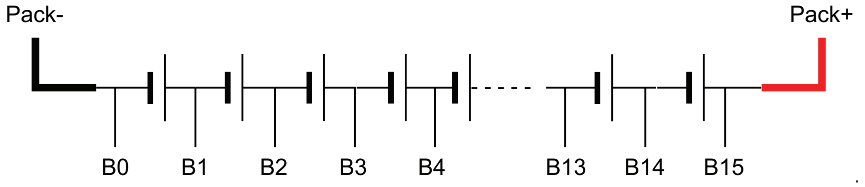


FIGURE 6. Battery Cell Configuration

For achieving higher battery pack capacity, battery cells may be connected in parallel.

Important Warning

It is extremely important for all the cells being paralleled to be at the same state of charge. If not, due to the very low internal resistance of the cells a very high current will flow from the higher charged cells to the lower. This is a very dangerous scenario since it may lead to damaged cells or serious injuries to the user.

When connecting cells in parallel, like when in series, the cells must be of the same characteristics (Ah, max current on charge, max current on discharge, internal resistance, etc). For optimizing the pack's performance and longevity, it is preferred to use cells from the exact same manufacturing batch.

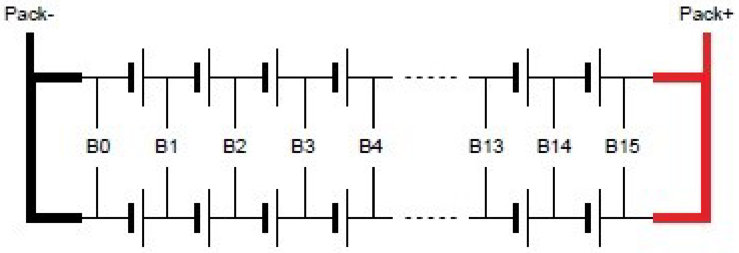


FIGURE 7. Cells in Parallel

For wiring the Molex connector, please consult Figure 8 for BMS1040x and Figure 9 for BMS1060x.

The black lines displayed on the connectors indicate jumper wires. The marked slots on the connector must be shorted with each other like shown by the black lines for the BMS to work properly.

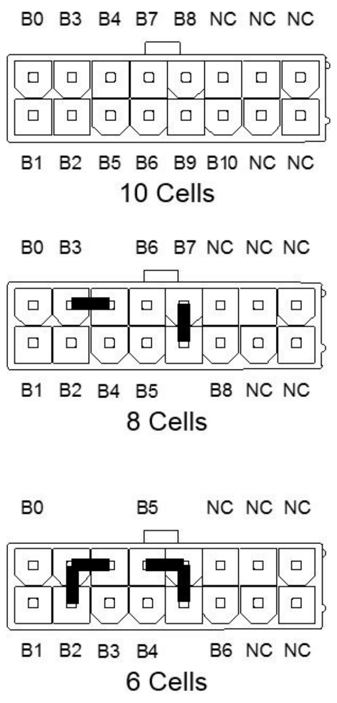


FIGURE 8. Pin-out Connections for 6 to 10 Cell Packs

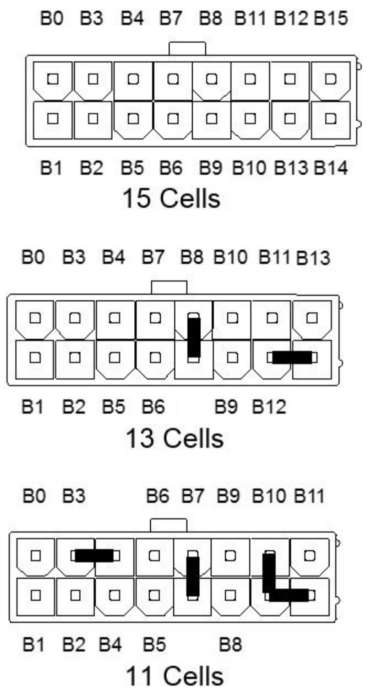


FIGURE 9. Pin-out Connections for 11 to 15 Cell Packs

Absolute Maximum Ratings

TABLE 4. Absolute Maximum Ratings

Description	Pin Name	Model	Typ	Min	Max
V_I & V_O Input and Output voltage	Pack+ - Pack-	BMS1040			40 V
		BMS1060			80 V
	Load+ - Load-	BMS1040			40 V
		BMS1060			75 V
	Aux+ - Aux- (Brake Resistor)	BMS10x0			75 V
	Charge+ - Charge-	BMS1040		-0.3 V	40 V
		BMS1060			75 V
V_I Input voltage	B_{n+1} - B_n, with n = 0…10	BMS1040		-0.3 V	7.2 V
	B_{n+1} - B_n, with n = 0…15	BMS1060			
	B5 - B0, B10 - B6	BMS1040			36 V
	B5 - B0, B10 - B6, B15 - B11	BMS1060			
	TS_nP - TS_nN, with n = 1, 2	BMS1040			3.6 V
	TS_nP - TS_nN, with n = 1, 2, 3	BMS1060			
V_I & V_O User Connections	Digital Output n – Pack- with n = 1, 2	BMS10x0x			60 V
	CANH / RS485A - Pack-			-4 V	12 V
	CANL / RS485B - Pack-				
	VCC - GND	BMS10x0 (input)	3.3 V	2.7 V	5.5 V
		BMS10x0A (output)	5 V	0 V (>15mA)	5.1 V
	PWM - GND	BMS10x0	VCC		
		BMS10x0A	5 V	4.9 V	5.1 V
	DIN2 - GND	BMS10x0x		3.3V	5.5V
I_I & I_O Input and Output current	Pack+ - Pack- Continuous *Peak (10s.)*	BMS1040x			100 A *150 A*
		BMS1060			75 A *150 A*
		BMS1060A			100A *150 A*
	Load+ - Load- Continuous *Peak (10s.)*	BMS1040			100 A *150 A*
		BMS1060			75 A *150 A*
	Charge+ - Charge-	BMS1040x			50 A
		BMS1060			35 A
		BMS1060A			50 A
	Aux+ - Aux- (Brake Resistor)	BMS10x0x			60 A
	Digital Output n – Pack- with n = 1, 2	BMS10x0x			1 A
Balancing	Balancing Current				300mA

BMS Operation

Push-Button Operation

When a DIN (Push Button by default) is configured in Operating Mode Control, the LED patterns are by-passed and Mode Control LEDs are displayed when the input is triggered.

Mode Control LED indications:

Power LED (Green) – Turns off and remains off as long as the DIN is triggered.

Communication LED (Orange) – Turns on and remains on as long as the DIN is triggered.

Status LEDs (Green/Red) – Green and Red turn off and remain off for the first 1000 ms.

Green and Red turn both on after 1000 ms (indicating that a reset will occur upon release).

Green turns off after 3 seconds and Red will perform a blinking pattern becoming faster as time pass until 10 seconds (then BMS will enter storage mode).

In this mode 5 actions are available.

- Short Tap (100 ms) and no current based errors triggered: The BMS will switch from Active to Idle state and vice versa.
- Short Tap (100 ms) and any current based error triggered: The current based errors will be cleared without switching operating state (Active/Idle).
- Short Tap (>1 ms and < 1 s) and BMS in storage mode: The BMS will wake up and enter Active mode.
- Long Press and Release (>1 s and < 10 s): The BMS will be reset upon release.
- Very Long Press (>10 s): From BMS firmware version 1.3 and above the BMS will be reset and en¬ter storage mode.

Details about other DIN available actions may be found in the Digital Input section at page.

Wake Up Function

In BMS10x0 the wake up function is available only from the user button, while in BMS10x0A it is possible to be applied through external mechanical switch using the DIN2.

TABLE 5.Wake Up function Table

	User Button	COM	DIN2
BMS10x0	●	○	○
BMS10x0A	●	●	●

Status LEDs and Flashing Patterns

The controller is equipped with three LEDs indicating information regarding Power, Status and Communication at a two second interval.

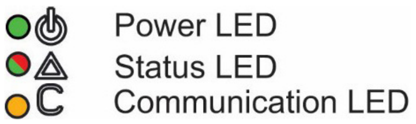


FIGURE 10. Status LEDs

Power LED

After the controller is powered up the Power LED will be activated, indicating that the controller is ON. The battery's State of Charge is mirrored on the Power LED duty cycle.

The current capacity is indicated by four different patterns. The system will inform the user that the battery is over 75% of the total capacity, over 50%, over 25% or below 25%. For example, over 75% is indicated by 100% of Power LED's duty cycle ON, over 50% with 75% duty cycle ON etc.

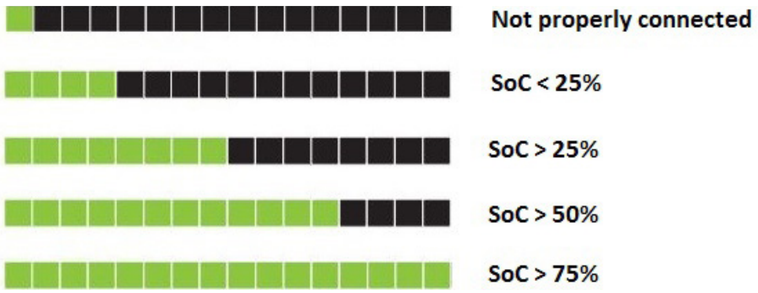


FIGURE 11. Capacity LED Display

Status LED

Note that every time the BMS is initialized, maybe due to power up, after reset or when entering and exiting low power mode, a blinking Red & Green LED pattern will be displayed.

When the user button is tapped the BMS will toggle between Active and Idle state. In Idle state for efficiency purposes no LED indications are used.

When the user button is pressed the LED patterns are temporarily disabled and the COM LED (Orange LED) will light up for us long as the user button stays pressed. For long presses over 1 second another orange LED will light up indicating a warning, that a reset will occur on release. If the button is still pressed for longer than a couple of seconds, a red LED will start blinking indicating that the BMS will enter low power mode if the button is not released. In case the user intents to force the BMS into a low power mode, then the button should be kept pressed for 10 seconds, until all LEDs turn OFF. Then a reset will occur and after the reset LED pattern is displayed, all LEDs will be disabled and the BMS will enter a low power state.

In BMS firmware version 1.3 the status LED patterns have been adjusted to the following, (see FIGURE 12)

The flashing pattern provides operating and exception status information. The Status LED is indicated by a GREEN and RED LED. The first half of the duty cycle includes operating information using the GREEN light. The second half indicates hazardous conditions detected in regards to the pack or the board.

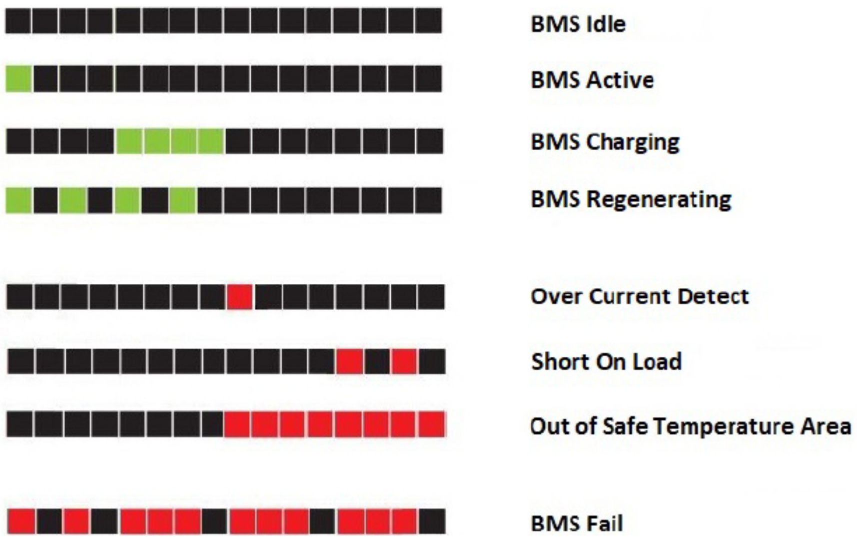


FIGURE 12. Status LED Display

Additional status information may be obtained by monitoring the controller with the PC Utility.

Communication LED

The communication LED gives information about CANbus and USB communication status.

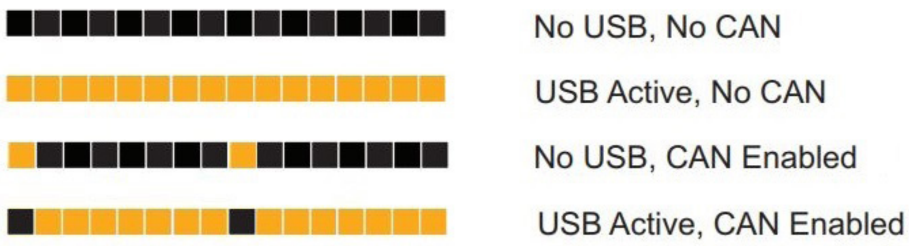


FIGURE 13. Communication LED Display

Configuration Before Use

RoboteQ's BMS is preconfigured with battery profiles for specific cell chemistry types. When the number of cells and the cell type are specified, voltage and temperature thresholds are automatically assigned. These parameters can then be adjusted to the user's needs, but must remain within the Battery's Manufacturer's published specifications. Nominal Ah capacity and maximum C-rate need to be set for an accurate gauge indication and battery protection.

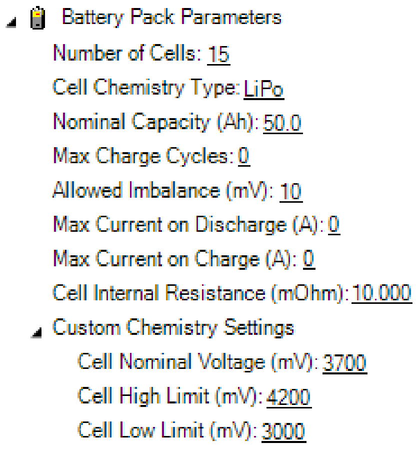
Considering balancing, the precision can be adjusted, minimizing cell voltage differences. Take care when altering the 10 mV default value because very low precision values may fail to satisfy the end of balancing condition.

The BMS has by default the auxiliary output switch function set to Disabled. This means that the state of the switch does not automatically change and script may be written to control it.

Choosing the predefined option Brake Resistor, the BMS will automatically handle the switch implementing an "excess energy dumping" algorithm. The auxiliary output may also be defined as a Fan or a Heater. Their functionality is also automatically handled by the BMS.

BMS initial configurations may be accessed in Utility's Setup tab via USB connection.

If the configurations are made via console or scripting the commands in the following table may be used.



- Battery Pack Parameters
  - Number of Cells: 15
  - Cell Chemistry Type: LiPo
  - Nominal Capacity (Ah): 50.0
  - Max Charge Cycles: 0
  - Allowed Imbalance (mV): 10
  - Max Current on Discharge (A): 0
  - Max Current on Charge (A): 0
  - Cell Internal Resistance (mOhm): 10.000
  - Custom Chemistry Settings
    - Cell Nominal Voltage (mV): 3700
    - Cell High Limit (mV): 4200
    - Cell Low Limit (mV): 3000

TABLE 6. Low Level Battery Pack Configuration Commands

Command	Arguments	Description	Example
		Configuration Commands	
^BTNC uu	Value	Configure number of cells in the pack	^BTNC 15
^CCHT uu	Type	Select the cell chemistry type used	^CCHT 1 0: LiPo 1: LiFePO4 2: Custom
^NOMC uu	Value * 10	Configure the nominal Ah rating of the battery, to set Ah = 50.0 the value of this command must be 500	^NOMC 500
^BPRC uu	Value	Configure the allowed imbalance of the cells in mV	^BPRC 10
^BAHT uu	Value	Max continuous rated current of the battery during discharge	^BAHT 50
^BALT uu	Value	Max continuous rated current of the battery during charging	^BALT 50
^CRES uu	Value * 1000	Nominal internal resistance of the cells in mOhm	^CRES 1000
^CNOV uu	Value	Define nominal operating voltage of the cells in mV (When custom chemistry)	^CNOV 3200
^BCHT uu	Value	Configure max operating voltage of the cells in mV (When custom chemistry)	^BCHT 3650
^BCLT uu	Value	Configure min operating voltage of the cells in mV (When custom chemistry)	^BCLT 2500

BMS Consumption

The BMS has two operating modes, normal operation and low power mode. The BMS will always be initialized in the normal operation mode after power up.

Normal Operation

To switch between Idle and Active state a short tap of the user button is required. For switching between states no errors should have been triggered, else the first short tap on the user button will clear any errors triggered by a current related issue. In that case a second tap will be needed to switch between states.

The BMS is always initialized on Active state upon power up.

For minimizing power consumption also on Active state, the user may configure the BMS LED indications to be turned off. This is recommended in cases that the BMS will be integrated in closed solutions in which the BMS cannot be seen by the user.

This may be configured using ^VIND cc command in the console, Setconfig(_VIND, cc) via scripting or configuring LED Indications option in Utility's Setup tab. The parameter cc may be 0 (Disabled) or 1 (Enabled).

The LED Indications are enabled by BMS's default configuration.

In this mode the BMS may be either in an Active or an Idle state. In both states the BMS consumption is almost the same, with the Idle state being slightly more energy efficient due to deactivation of the LED indications.

Idle – All power switches are disabled and the LED indications are deactivated. The BMS does not have power on the Load, Auxiliary or Charger terminal and it is the safest way to make adjustments on the power connections. Consumption is slightly less than when in Active state.

Active – All power switches are enabled for automatic or manual handling (depends on the configuration at Switch Control Mode). By default, the LED indications are enabled allowing easy and fast monitoring of the systems status. In this mode power from the battery can be delivered to the load and charging of the battery is allowed.

Consumption Table

All optional peripherals, CAN, RS485, Digital Input and Pulse Output, are disabled by default.

When any of the peripherals are enabled, the BMS will automatically handle the internal Peripheral VCC to provide power only to the necessary functional blocks.

Based on the following table we may calculate the BMS peripherals' power consumption.

If no peripherals are used the total BMS power consumption is given by the Main VCC Base Consumption plus each LED's consumption. To calculate each LED's power consumption the duty cycle must be taken into consideration since most LEDs will remain off during most of the time of each pattern in normal circumstances.

When any of the peripherals is used, the Peripheral VCC Base Consumption must be added. Then, the actual BMS consumption is given by adding the Additional Consumption of each dependent peripheral enabled to the total.

TABLE 7. Consumption Table

	Base Consumption (mA)	**Additional Consumption of Dependent Peripherals if Enabled (mA)**			
Main VCC (1)	11.8	LED Green	LED Red	LED Orange	
		1.35	1.47	1.48	
Peripheral VCC (2)	8	CAN	RS485	MultiPWM	DIN
		3.36	2.74	1.8	0

(1) Main VCC is always enabled in Normal Operation

(2) Peripheral VCC is enabled when at least one of the dependent peripherals is enabled

Storage Mode

In this mode BMS consumption drops down to 3 mA. All switches are forced to be disabled, communication is deactivated and scripting is paused. All configurations, RAM and backup domain are preserved.

When in Storage Mode none of the BMS operations is available and communication is not supported. For resuming any BMS activity the system must be waken up to Normal Operation Mode.

Exiting Low Power Mode – To exit low power mode, a short tap on the user button is needed. This will trigger the wake-up interrupt, resetting the BMS and initializing it into to Normal Operation.

Entering Low Power Mode – To force the BMS into Low Power Mode, a long, 10 second, press of the user button is needed. This will reset the BMS and initialize it into Low Power Mode.

The system may also enter Low Power Mode automatically, in case at least one the cells drops below minimum allowed voltage for more than 5 minutes.

Important Warning

The BMS is powered by the battery pack itself, which means there will always be some energy drained from the battery even when no load is connected. Consumption depends on BMS power mode.

Balancing

Balancing is the process of eliminating the voltage differences between battery cells that is created by the different rate of charge and discharge of each cell. Every battery that consists of multiple cells in series needs to be balanced for maximizing its capacity, making all of its energy available and prolonging the battery's life. The voltage differences between the cells of an unbalanced battery grow in time and uneven stresses are exerted on the cells. This phenomenon leads to more strain on the weakest cell in the chain leading the battery to fail faster than expected.

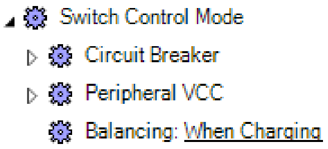
Generally, balancing is performed during the charge cycle by enabling a low value resistor across the cells that have higher voltage, and thus diverting the charge current to lower voltage cell.

Balancing is activated automatically and the user does not have to know or learn any balancing strategies for the BMS10x0x to work.

While balancing is mostly desired during the charge cycle, there are cases that balancing is necessary when not charging. An example of a scenario like this would be short and frequent charging cycles. In this case the BMS will not be connected on the charger sufficient time for the balancing to restore the uneven cells. When balancing is configured as "Always" instead of "When Charging", the BMS will even out the cells as long as the State of Charge of the battery is above 90%.

Using Automatic Balancing Options

By default the BMS balancing option is set to "When Charging". This option may be accessed using the BMS utility's Setup tab in Switch Control Mode list. The two options "When Charging" and "Always" may be chosen from the Balancing parameter.

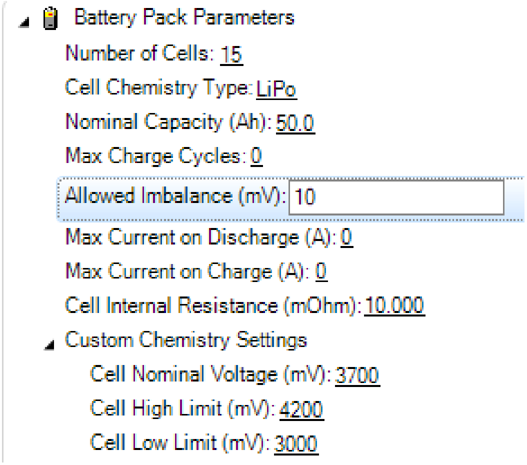


Switch Control Mode
Circuit Breaker
Peripheral VCC
Balancing: When Charging

When using console to configure this parameter, the ^SWMAN 8 ww command should be used. By setting the value of ww to 0, the BMS will balance the cells only in the presence of a charger. Setting ww value to 1, will allow the BMS to balance the cells even if the charger has been disconnected from the system for as long as the SoC is above 90%.

For configuring via scripting, Setconfig(_SWMAN, 8, ww) command may be used.

In all cases, if all the cells are within the configured Allowed Imbalance range, the balancing is paused.



Battery Pack Parameters
Number of Cells: 15
Cell Chemistry Type: LiPo
Nominal Capacity (Ah): 50.0
Max Charge Cycles: 0
Allowed Imbalance (mV): 10
Max Current on Discharge (A): 0
Max Current on Charge (A): 0
Cell Internal Resistance (mOhm): 10.000
Custom Chemistry Settings
Cell Nominal Voltage (mV): 3700
Cell High Limit (mV): 4200
Cell Low Limit (mV): 3000

Using User Controlled Balancing

With this option, the user may force the BMS to perform balancing even if none of the previously mentioned conditions are met. For safety reasons, when user-controlled balancing is applied, the SoC must be above 50%, or else the BMS will ignore the request.

The real time command for user-controlled balancing is !SWS 8 ww. When ww is set to 0, only BMS automatic handling of the balancing procedure is applied. If the ww value is set to 1, the BMS will force balancing the cells until the SoC drops below 50% or all the cells reach the Allowed Imbalance threshold.

Via scripting Setcommand(_SWS, 8, ww) may be used.

Note that whether user-controlled balancing is applied or not, the balancing option set at Balancing configuration parameter is still applied. When the BMS generates a balancing request, the user's request is ignored.

The user may only force balancing when the BMS is not automatically handling balancing, which means that the user cannot force a no-balancing request when the BMS is balancing the cells but they may force a balancing request when the BMS is not balancing the cells.

TABLE 8. Low Level Balancing Commands

Command	Arguments	Description	Example
		Run Time Commands	

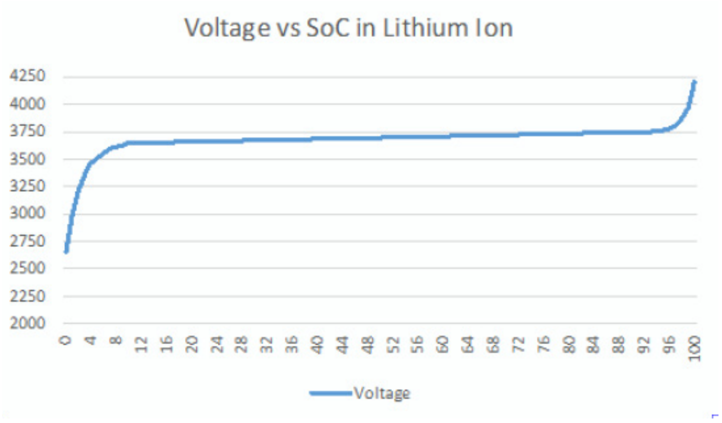
!SWS uu ww	uu = 8
	ww = 0, 1
		When 0 force balancing is not applied, when 1 balancing is forced	!SWS 8 1

		Configuration Commands	

^SWMAN uu ww	uu = 8
	ww = 0, 1
		When 0 balancing only occurs when charger available, when 1 balancing is applied also when above 90% SoC	^SWMAN 8 1

State of Charge

SoC is crucial in protecting the battery. Due to the charge and discharge behavior of the LiOn and LiFePO chemistries, measuring the individual cell voltage is not sufficient for specifying the state of charge of the battery. The voltage output for this type of cell shows very small changes when charging or discharging until reaching full charge or discharge.



The state of charge is defined as a percentage of the Ah measurement taken from the BMS10x0 versus the real Ah capacity of the pack recalculated in every full charge/discharge cycle.

When it is first connected, the BMS10x0x considers the battery at 50% SoC, assigning the nominal Ah/2 capacity as the energy left in the pack and the nominal max Ah capacity as the real measured capacity. At the end of the first full charge cycle, since the initial 50% SoC assumption was not totally accurate (real initial conditions of the battery are unknown), the BMS will set a reference point for the 100% State of Charge. For the rest charge-discharge cycles the BMS will monitor the energy flowing in and out of the battery maintaining an accurate SoC indication using Coulomb Counting.

The Ah counting is maintained during both charge and discharge cycle adapting through time to the actual capacity limits, while the battery ages.

The SoC is calculated based on two parameters, the Coulomb Counting measurement and the minimum and maximum cell voltages. Since the cell voltages are not an accurate indication of State of Charge while the battery operates in the nominal state, Coulomb Counting is considered the dominant indicator. When at least one of the cells reaches near the minimum or maximum cut off thresholds, where the voltage change is significant depending on the energy stored in the battery, this cell's voltage becomes the dominant indicator for the battery's SoC. Observing the SoC allows the user to know when the pack will be isolated from the load (0% SoC) and when it will be isolated from the charger (100% SoC).

A time variable calculates in real time an estimation of battery's depletion time in minutes during discharging. During charging, the time variable provides the time in minutes that is still needed for the battery to be fully charged.

In summary, the State of Charge is determined as a percentage of the current stored energy versus the maximum battery capacity while operating in the nominal operating area which is around 10% to 90% SoC, or based on the cell voltages when below 10% or above 90%.

The State of Charge is readable via USB, CANbus, RS485, PWMOut and an optional Bluetooth module.

TABLE 9. State of Charge Command

Command	Arguments	Description
	Query Commands	
?BSC	-	Returns state of charge of the battery in per cent (%) formatting.

Charging

The easiest and safest charging method for lithium cells is the so-called CC-CV, meaning Constant Current – Constant Voltage.

Any current limited power supply may be used as a charger, as long as it can deliver adequate current at the necessary voltage. An example charger option is RPB-1600-48 for 50V applications.

For configuring the charger's voltage, please refer to Cell Chemistry and Voltages section (Page 7, Table 3) and set the charger according to the max voltage for the corresponding cell number and chemistry. The current should be set lower or equal to the maximum current during charge specified by the cell manufacturer. In case this value cannot be obtained, usually it is safe to consider the maximum current during charge as the nominal cell capacity in Ah divided by 1 h. For example, a 10 Ah battery would safely operate at up to 10 Ah/ 1 h = 10 A during charge.

Remember to configure your BMS system using the specified thresholds to ensure that your battery operates in the safe operating area.

When the BMS system and the charger are properly configured, the charging process will automatically begin after connecting the respective terminals. At the beginning, the charger's voltage will drop delivering 100% of the max charge current. While the pack is being charged its voltage will at some point reach the nominal fully charged value and the current will start to drop. The BMS system will continuously monitor and protect the pack from any dangerous event.

If any of the cell voltages exceeds the maximum threshold or the pack is over heated, the BMS system will pause the charging process allowing balancing to even out the differences between the cells. When the system recognizes that the charging speed is low, meaning slow voltage changes during charging, the BMS will activate the passive balancing algorithm, extracting energy from the higher voltage cells so that the lower voltage cells can catch up.

If for any reason the current exceeds the configured thresholds or in case a short circuit event occurs, the BMS will automatically enable the circuit breaker to deactivate the dangerous channel.

In a case where the charger parameters are not set with caution and the cells' voltage safe operating area is violated, the BMS will pause the process. Depending on the conditions under which the charging was paused, different rest times will be set. Balancing will still be applied as long as the BMS recognizes a charger to be connected. For example, if one of the cells hits the overvoltage threshold while still charging with a fast speed (current close to maximum nominal current during charging), the system will let the battery rest for a longer time than if it were charging with a slow speed, allowing balancing to operate for more time before charging again. When charging at slow speeds the battery is not stressed even when operating near the maximum thresholds. However, if it charges with a high current, an unbalanced cell's voltage may rise above the threshold very quickly because of the stresses.

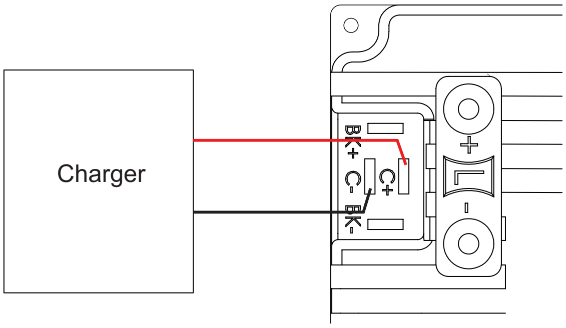


FIGURE 14. BMS Charge Configuration

TABLE 10. Charger Voltage vs Number of Cells

	Maximum Charging Voltage			
Cells	**LiPo**	**LiFe**	**Custom**	
6	25.2	21.6	6 x Cell High Threshold	**BMS1040x**
7	29.4	25.2	7 x Cell High Threshold	
8	33.6	28.8	8 x Cell High Threshold	
9	37.8	32.4	9 x Cell High Threshold	
10	42*	36	10 x Cell High Threshold	
11	46.2	39.6	11 x Cell High Threshold	**BMS1060x**
12	50.4	43.2	12 x Cell High Threshold	
13	54.6	46.8	13 x Cell High Threshold	
14	58.8	50.4	14 x Cell High Threshold	
15	63	54	15 x Cell High Threshold	
*Only BMS1040A can hadle 10cell LiPo Batteries with state of charge up to 100%				

When calculating the charger's voltage, we may also use this formula:

Charger Voltage = nn x (Cell High Threshold – cc)

With nn being the number of cells used in the pack, Cell High Threshold being the absolute maximum voltage threshold of the cell chemistry used and cc being a safety value between 50-150 mV.

When the charger's voltage is set a bit lower than the absolute maximum threshold of the battery, the cells will maintain a better health in the long run but the pack will not be charged up to 100%. Near 100% state of charge voltage and the energy stored have a linear relationship, which means that if the cc value is chosen bigger than 150 mV the maximum energy stored might drop below 95%.

Usually, by choosing the cc between 50 mV and 150 mV, when designing the charging system, will provide the benefits of prolonged battery health and around 98% the total capacity of the battery during charge and discharge cycles.

Brake Resistor

The BMS10x0x is equipped with a pair of terminals designed to be connected to an external resistor. Placing an external resistor is optional but recommended when driving a regenerative load such as a motor. The Brake Resistor is used to release the excess regenerative energy from the load into the environment as heat in case the current is out of the safe operating boundaries, the battery is full, or in any other unsafe conditions. The resistor is automatically paralleled with the load when BMS's algorithm recognizes harmful excess energy is returning into the pack.

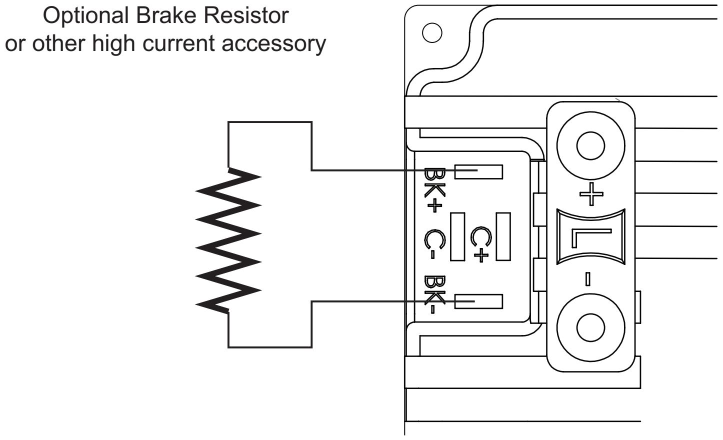


FIGURE 15. Connection to Brake Resistor or High Power Accessory

The BK+ terminal is connected internally to the Load+ terminal. The BK- goes to a MOSFET transistor which will internally connect to the Load- terminal when the resistor is enabled.

The figures below show the current flow under the various load and regeneration conditions

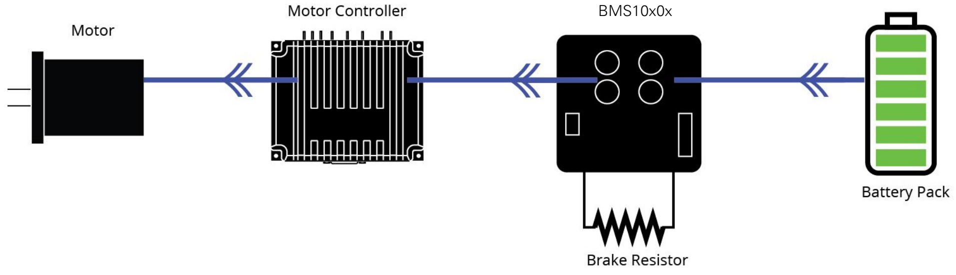


FIGURE 16. Current Flow from Pack to Load

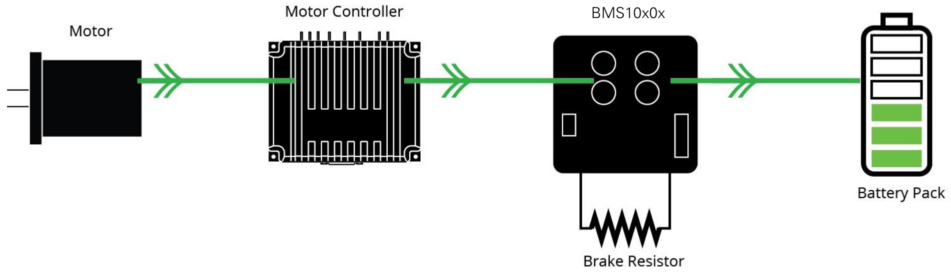


FIGURE 17. Regenerating Energy from Motor

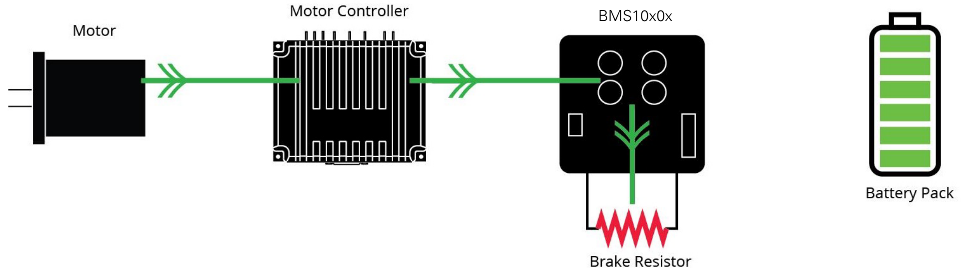


FIGURE 18. Excess Energy Out to Brake Resistor

By default, the Brake output is deactivated and may be controlled by user command or script. The BMS supports Microbasic scripting that can be written using the PC Utility accompanying the product.

If a Brake resistor is not needed, two other predefined implementations may be chosen: a cooling fan or a heater function. These features utilize the temperature thresholds configured in the utility. This guarantees that the pack is kept warm or cool enough during charging in a significantly cold environment or makes charge/discharge safer in hot environments.

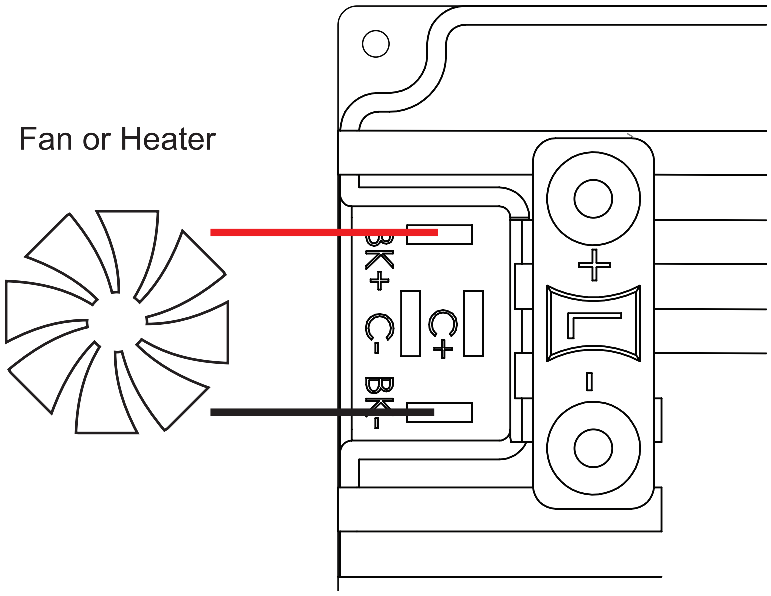


FIGURE 19. Configuration for Guaranteed Temperature Balance

State of Health

The State of Health in a battery is a vague measurement to infer its wellbeing. In other words, there is no exact way to specify how many years or days the battery has left until one of its cells fail.

The BMS10x0x provides the user with the total Charge/Discharge cycles counted from the first time the pack was connected to the system. Based on the manufacturer's specifications as a reference, this may be used as a first indication for the battery's State of Health. Not all cell manufactures provide the client with statistics about the typical endurance of their cells. This is due to the fact that it requires many tests and depends on the circumstances under which the cells operate.

Another indicator is the nominal capacity vs the actual capacity of the battery. As the cells age their capacity drops. The BMS10x0x corrects the max capacity variable when charging and discharging. It also informs the user when the real capacity is significantly lower that the nominal. When measured capacity reaches 50% the nominal value, the State of Health is considered to be 0%. SoH 50% corresponds to real capacity and 75% the nominal. If any of the cells are defined as "dead", SoH is also considered to be 0%. A cell is considered "dead" when it is unable to store energy any more or has dangerous behavior, like constantly overheating during Charge or Discharge. A "dead" cell would make the system cut off either the Charge or Discharge process long before the other cells reach their limits. As such, the battery will not be able to delivery nor store energy.

The battery's charge/discharge cycles are counted as the number of times the battery delivered energy equal to its nominal capacity. For example, a 10 Ah battery is considered to have made a charge/discharge when the total of 10 Ah was delivered. In other words, if a battery that is 100% charged reaches 90%, this gives a total of 10% the nominal capacity. If the battery is then charged back to 100% and is discharged to 90%, we would have a total of 20% the nominal capacity delivered. When this process is repeated 10 times, the total energy delivered would be equal to 100% the nominal capacity, thus one full battery cycle.

Another example with different percentages is the following: A 100% SoC battery is discharged to 70% (30% the nominal capacity delivered), then charged back to 90% and discharged to 20% (70% delivered), which again gives a total of 100% the nominal capacity delivered, meaning one full battery cycle.

TABLE 11. State of Health Command

Command	Arguments	Description
	Query Commands	
?SOH	-	Returns state of health of the battery

Circuit Breaker

All BMS circuit breaker switches are embedded inside the BMS board. By default all power switches Pack In, Load Out, Charge Bidirectional and Auxiliary Out are configured to be automatically handled by the BMS.

Using the Switch Control Mode any of the above mentioned switches may be configured as Manual. This way the corresponding switch will be left for the user to handle. This feature is ideal for users who need to implement a personalized BMS strategy for charging, discharging or secondary devise handling (auxiliary devices).

Using Switch Control Mode

All switches are automatically handled by default. If any of the switches is configured as Manual, then the BMS has no control over the switch in normal operation. Note that in case of a dangerous condition the BMS will still disable the necessary switches to prevent damage to the battery or the itself.

An example of operating in manual control is the following. If Charge is set to Manual, the BMS will not automatically enable the charger switch even if a charger is connected (normally the BMS enables charging switch when charger is present). This way the user may implement a script on when the charging process is initiated. In case the charger is connected with inverted polarity and the user tries to enable the charging switch while in Manual mode, a Inverse Charger error will be trigger and the BMS will automatically force the Charger switch to turn off until the error is cleared. When the error has been handled the BMS will return the switch control to the user.

To configure the switch mode the utility may be used.

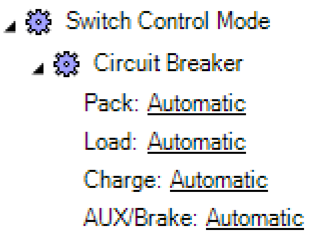
If the configuration is performed via console, the command is ^SWMAN uu ww and via scripting Setconfig(_SWMAN, uu, ww). The value of uu defines the configured switch and ww defines the switch's function.

uu = 1 (Pack), 2 (Load), 3 (Charge), 4 (Aux)

ww = 0 (Automatic), 1 (Manual)

Controlling the Switches

After having set the switch mode in Manual, monitoring and control of the switch state is performed by the SWS command. When the ?SWS uu command is used (SWS used as query command), the system will return the user assigned switch state for the uu switch. If the returned value is 0, the user switch state is requested to be disabled. For a returned value of 1, the user has requested the switch to be enabled.
Via scripting Getvalue(_SWS, uu) command may be used.



Switch Control Mode
Circuit Breaker
Pack: Automatic
Load: Automatic
Charge: Automatic
AUX/Brake: Automatic

For assigning a new switch state !SWS uu nn command is used. The nn parameter may be 0 (turn off switch) or 1 (turn on switch).

Via scripting Setcommand(_SWS, uu, nn) may be used.

uu = 1 (Pack), 2 (Load), 3 (Charge), 4 (Aux)

nn = 0 (Switch Off), 1 (Switch On)

Some important notes. If the switch is not set in manual control the SWS command will still accept the user requests but they will not be applied on the switches until they are configured in Manual control. For example, if !SWS 3 1 command is sent and the Charge switch is in automatic control, the switch state will not be affected by the user. However, the ?SWS 3 command will respond 1, indicating that the user has requested for this switch to be enabled when in Manual control.

For monitoring the actual current state of the BMS switches the ?BMS uu command should be used, explained in a different section of this datasheet.

TABLE 12. Low Level Switch Control Commands

Command	Arguments	Description
		Configuration Commands

^SWMAN uu ww	uu = 1,…,4
	ww = 0, 1
		Uu = 1 (Pack), 2 (Load), 3 (Charge), 4 (Aux)
		Ww = 0 (Automatic), 1 (Manual)

		Real Time Commands

!SWS uu ww	uu = 1,…,4
	ww = 0, 1
		Uu = 1 (Pack), 2 (Load), 3 (Charge), 4 (Aux)
		Ww = 0 (Switch OFF), 1 (Switch ON)

		Query Commands
?SWS uu	uu = 1,…,4	Uu = 1 (Pack), 2 (Load), 3 (Charge), 4 (Aux)

The Circuit Breaker is one of the most important functions of the BMS10x0 and that is because the first priority in any case is the protection of the battery and BMS board. The Pack, Load, Charger and Brake Resistor can all be connected and isolated separately. The circuit is fully operational by way of implementing internally all the necessary components. While the user may control the switches using the Microbasic script, the protection actions are automatically triggered by default if any of the configured thresholds is exceeded.

The board is designed to withstand up to 150 A peak and 100 A continuous current (BMS1040X, BMS1060A) or up to 150 A peak and 75 A continuous current (BMS1060). The BMS10x0x is highly configurable and by configuring the battery's parameters the system can protect the cells from different over current conditions during charging or discharging and different response times can be set for acting on a short circuit event.

Response time is also configurable for over current protection. After an overload error has been triggered the system will either lock or retry after the configured delay time to connect to the channel for the configured number of retry times.

Protection measures are also taken when Over/Under Voltage or Over/Under Temperature events occur, always with respect to the configured values.

TABLE 13. Low Level Amp Trigger Configuration Commands

Command	Arguments	Description	Examples
^THLD xx yy	Channel, Value	Set Response Time for Short	^THLD 1 0 xx 1: Load, 2: Charge yy 0: Quick 1: Medium 2: Slow 3: Disabled
^ATRIG xx yy	Channel, Amps	Amp Trigger Level	^ATRIG 1 150
^ATGD xx yy	Channel, Delay ms	Amp Trigger Action Delay	^ATGD 1 500
^MXRTR xx yy	Channel, Number Rtr	Max Retry Times	^MXRTR 2 0
^REDL xx yy	Channel, Time ms	Delay Between Retries	^REDL 2 1000

Peripheral Power

In BMS1040xBT and BMS1060xBT a Bluetooth module is included. The module's power supply may be switch off for power management reasons but this is left to handled by the user. In Switch Control Mode, the option Peripheral VCC will be set to Manual by default.

TABLE 14. Low Level Bluetooth VCC Control

Command	Arguments	Description
Configuration Commands		

^SWMAN uu ww	uu = 6
	ww = 0, 1
		Uu = 6 (Bluetooth VCC)
		Ww = 0 (Always OFF), 1 (Manual)

Real Time Commands		

!SWS uu ww	uu = 6
	ww = 0, 1
		Uu = 6 (Bluetooth VCC)
		Ww = 0 (Switch OFF), 1 (Switch ON)

Query Commands		
?SWS uu	Uu = 6	Uu = 6 (Bluetooth VCC state)

In all above cases, each switch state is set by the user using the real time command SWS. This command will force the switch state to be Enabled or Disabled depending if a 0 or an 1 is set at the corresponding bit band index of each switch.

Temperature Sensors

An internal sensor periodically provides the user with the board temperature. Depending on the BMS10x0x model, up to three external sensors can be used to measure temperatures. It is recommended that the thermistors be positioned in equally increments on the pack. The temperature thresholds are configurable so, as long as the values set define a safe operating area, the Circuit Breaker will do the job. In the BMS1040x, two external sensors are supported, whereas the BMS1060x has three.

The temperature sensors should be NTC thermistors of 10k resistance. A sensor example is NTCALUG03A103G by Vishay, digikey number BC2385-ND, or any similar sensor.

TABLE 15. Read Temperature Values Command

Command	Arguments	Description
		Query Commands
?T uu	uu = 1,…,4	1 (Read internal BMS temperature), 2 (TC1), 3 (TC2), 4 (TC3)

Dout1 & Dout2

The BMS provides two general purpose digital outputs suitable for any application. For example, driving a small fan for cooling the pack, the board or any other system and as a warning indication driving either a buzzer or an alert LED.

When using the digital output, users must use caution. Due to its purpose (ex. activating a buzzer under some conditions like undervoltage) no automatic protection handling is performed. This means that even if the pack has an undervoltage indication, the Douts will not be automatically disabled.

The digital outputs may be assigned one of the predefined functions or be assigned as never automatically activated. Their behavior may then be specified in the user script.

The predefined activation statements are: Never, Over Temperature, Voltage, or Short/Overload. When Never is selected, the outputs can be activated by User Commands, or Scripting

The state of the digital outputs is visualized in RoboteQ's PC utility. Green LED indicates that the digital output is activated and Grey that it is disabled. For user's convenience, each digital output may be assigned a custom name based on the functionality, there by being easily distinguished and monitored.

TABLE 16. Read Digital Output State Command

Command	Arguments	Description	Examples
?DO		Read Current Digital Outputs	?DO

For configuring the digital output two parameters are provided, the logical state of the Dout and the condition. By default, the digital port is set as normal logic (active HIGH, deactivated LOW) and the activation condition is disabled. This way a user script may be implemented depending on the desired functionality.

Furthermore, if one of the three predefined functions are set, the BMS system will automatically assign the desired output state based on the measurements and the active level parameter. For example, if Dout is set to be active HIGH on over temperature, the system will set the output high if the user defined temperature limit is exceeded. It will set the output to LOW as long as no over temperature has occurred or when an over temperature error has been cleared. The BMS resets when the temperature drops below the over temperature value minus hysteresis.

The Douts are designed as pull-down switches, which means that when the switch is turned on the Dout terminal is connected to the packs negative terminal. This way the user may connect as a power supply to the LED, fan, buzzer, etc., a secondary battery, take power directly from the main battery, or use the load output or auxiliary output. If an external power source is utilized, users must make sure that the external GND is connected to the negative terminal of the pack.

Important Warning

Dout1 & Dout2 are capable to accept constant current up to 1 A. Since these outputs are unfused, please be aware of the driving load rating. Unwanted damages could be done for non-proper operation.

Connecting Resistive Loads to Outputs

Resistive or other non-inductive loads can be connected simply as shown in the diagram below.

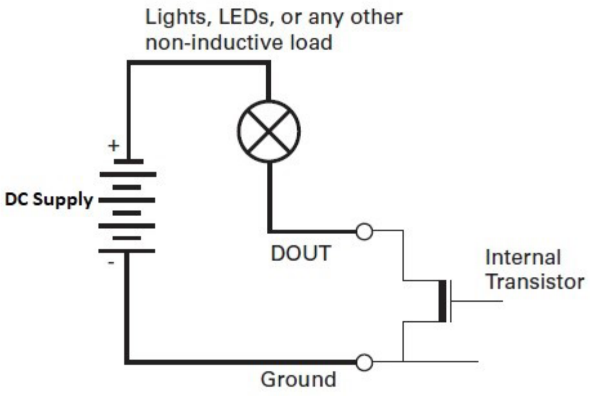


FIGURE 20. Connecting resistive loads to Dout pins

Connecting Inductive loads to Outputs

The diagrams on Figure 21 show how to connect a relay, solenoid, valve, small motor, or other inductive load to a Digital Output.

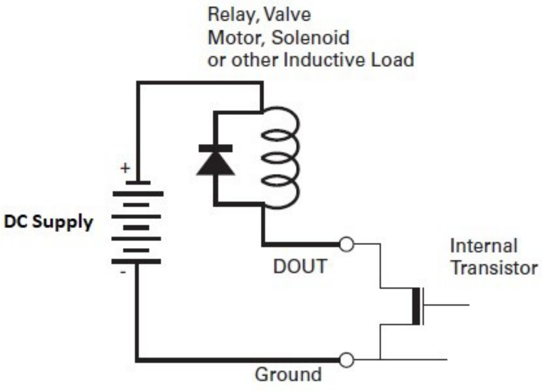


FIGURE 21. Connecting inductive loads to Dout pins

Choosing the DC Supply For Digital Outputs

Battery Pack Positive Terminal – The load may be powered directly from the positive terminal of the battery pack. In this case, even if the BMS detects a hazardous condition, the load connected on the digital output will be fully handled by the user.

This scenario is preferred if the load is used to handle the causes of an error or informing the user about the situation. Possible applications, enable a fan in case of over temperature, LED or a buzzer indication in case of short circuit, etc.

On the other hand, caution must be taken when extracting power from the battery bypassing the BMS. In case of under voltage, short circuit and other similar issues, the BMS will not be able to disable the load connected on the digital output and this might lead to damage on the battery or the system.

Main Load Positive Terminal – By using the BMS's main load output as a power source to the secondary load connected on the digital output is a safer implementation. In case of an error the BMS will disable the power on the load's side, disabling the secondary load as well. Important is that even in this case the BMS will not automatically handle the digital input's state.

This is the ideal implementation if the secondary load will not be needed in cases of errors.

External Supply – An external power supply could also be used, as long as the voltage does not exceed the battery's voltage. An important note when using an external power supply is to make sure to use a common GND with the BMS.

Example of external power supplies are 5V generated from a motor controller, the output of a DC/DC converter, etc.

TABLE 17. Low Level Digital Output Configuration Commands

Command	Arguments	Description	Examples
^DOA xx yy	Output Index, Trigger	Digital Output Active When	^DOA 1 0 xx 1: Dout1 2: Dout2 yy 0: Never 1: Over Temp 2: Over Volt 3: Overload/Short
^DOL xx yy	Output Index, Level	Digital Output Level Logic	^DOL 1 1

Digital Inputs

Two digital inputs are available in the BMS. Digital input 1 is the on board push button and digital input 2 is an optocoupled digital input for BMS10x0 versions and non optocoupled for BMS10x0A versions on the User O/I connector. The default functionality of DIN1 is Operating Mode Control and DIN2 is disabled by default for minimizing power consumption.

The DIN2 logical states may be configured to be activated when High or Low. When DIN2 Active Level is configured High, GND will be captured as a logical 0 (deactivated) and 3.3-5 V as a logical 1 (activated). If the Active Level is configured Low, GND will be captured as a logical 1(activated) and 3.3-5 V on the input as a logical 0 (deactivated).

Wake Up Function - In BMS10x0 the wake up function is available only from the user button, while in BMS10x0A it is possible to be applied through external mechanical switch using the DIN2.

General Purpose Input - In this mode the state of the input is available but no action is defined. This is the ideal mode for implementing personalized functions for logically High and Low states of the inputs using the build-in scripting language.

Operating Mode Control - This is the default option for DIN1. This section is described at Push-Button and LEDs section.

Emergency Stop - When the DIN is activated the BMS will stay in Idle mode. When deactivated the BMS will operate in active state.

Load Switch Control - If Load has been configured in Switch Control Mode as Manual, when the DIN is activated the Load switch will be enabled. When deactivated no power will be available on the load. If the Load switch is configured as Automatic in Switch Control Mode, this option has no effect.

Charge Switch Control - If Charger has been configured in Switch Control Mode as Manual, when the DIN is activated the Charger switch will be enabled. When deactivated the charger switch will remain disabled. If the Charger switch is configured as Automatic in Switch Control Mode, this option has no effect.

Clear Over Current Error - On activation over current on load and over current on charger errors are cleared. For safety reasons the input on the DIN for this option should be a short pulse, to avoid staying in an activated state if multiple over current errors occur.

Clear Short Circuit Error - Short circuit on load and inverse current errors may be cleared with this option. The error flags are cleared upon activation of the DIN and it is important for safety reasons to use a short pulse for activating the DIN, to avoid clearing multiple short circuit triggers. The DIN should be normally deactivated when using this option.

Clear OC and SHC Errors - Both Clear Over Current Error and Clear Short Circuit Error applied when using this option. For safety reasons DIN input should be activated using a short pulse like mention in the above two cases.

Run Script - When this option is used the stored script will be executed from the beginning for as long as the input is activated. When the DIN is deactivated the script will be reset and paused.

TABLE 18. Low Level Digital Input Configuration Commands

Command	**Arguments**	**Description**
		Configuration Commands

^DINA uu ww	uu = 1, 2
	ww = 0,…,9
		Uu = 1 (User button – DIN1), 2 (DIN2)
		Ww = 0 (Disabled)
		1 (General Purpose Input)
		2 (Operating Mode Control)
		3 (Emergency Stop)
		4 (Load Switch Control)
		5 (Charge Switch Control)
		6 (Clear Over Current Error)
		7 (Clear Short Circuit Error)
		8 (Clear OC and SHC Errors)
		9 (Run Script)

^DINL uu ww	uu = 2
	ww = 0, 1
		Uu = 2 (DIN2)
		Ww = 0 (Active when High)
		1 (Active when Low)

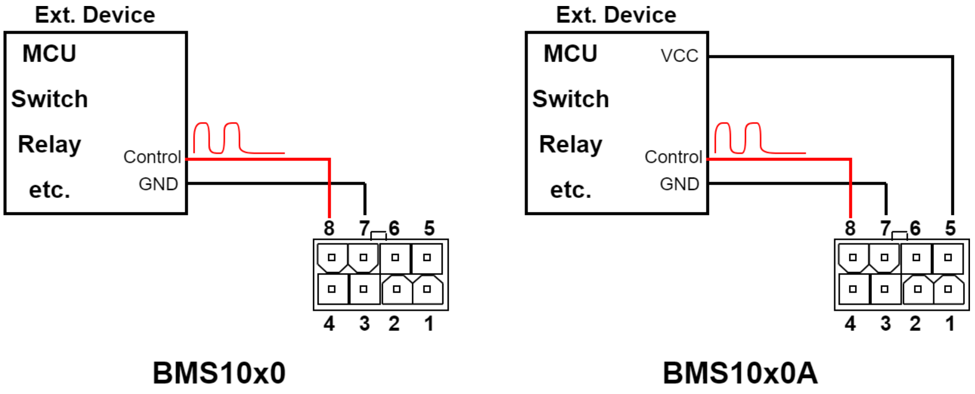


FIGURE 22. Connection of external signal to DIN2

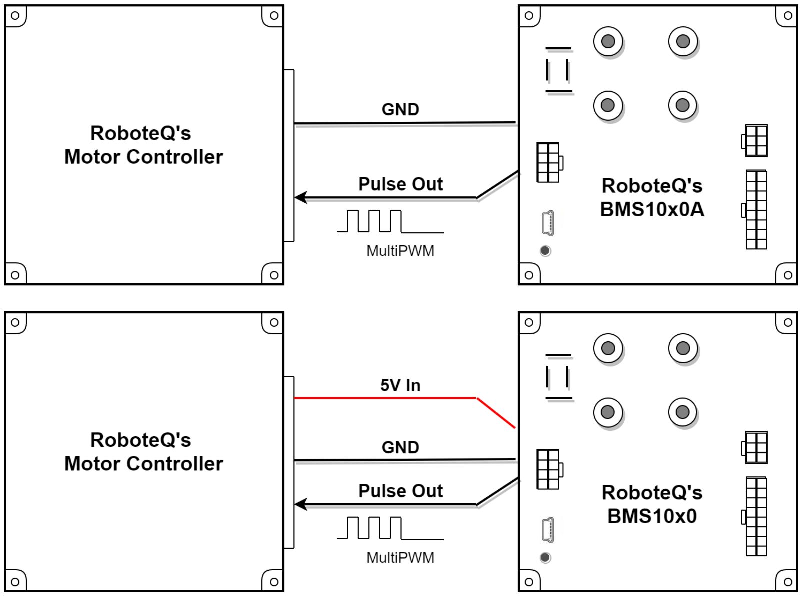
Scripting

Scripting is a common feature in every RoboteQ product and documentation can be found on RoboteQ's website (www.roboteq.com), including examples and tutorials.

With scripting all system variables can be accessed and manipulated. Custom strategies can be implemented controlling the circuit breaker, digital outputs, auxiliary output, dynamically adjusting protection thresholds etc.

MultiPWM

MultiPWM mode is the simplest way for interfacing with RoboteQ's motor controller. As the name implies, this proprietary method uses a succession of variable duty-cycle pulses to carry the AmpHours stored in the battery, BMS status, switch states and State of Charge (SoC in %) information.



Any of the controller's pulse input can be configured as a MultiPWM input. The diagram below shows how simple this one-wire interfacing is.

Enabling MultiPWM Communication

BMS10x0x has MultiPWM mode disabled in its factory default configuration. To enable MultiPWM transmission, the pulse output option must be set to RoboteQ MultiPWM on the BMS when using the PC utility. When configuring via the console use

```
^PWMM 1
```

To enable pulse output in MultiPWM mode.

Accessing Sensor Information

Once enabled, the pulses are sent continuously by the sensor 100 times per second. The pulses are captured and parsed by the motor controller as they arrive. A real time mirror image of BMS data is then present inside the controller. From there the BMS information can be read using serial, USB, CAN or MicroBasic scripts like any other of the controller's operational parameters.

The following Motor Controller queries are available for reading the captured sensor data.

?BMC : Read Ah in battery

?BMF: Read BMS status flags

?BMS: Read BMS switch states

?BSC: Read battery state of charge in per cent

Reading Ah

?BMC or GetValue(_BMC)

Returns the battery's stored Ah.

Reading BMS Status Flags

?BMF or GetValue(_BMF)

Returns the BMS status flags. Each flag is assigned one of the 8 bits constructing the received byte. For extracting each flag consult note (*1) presenting the flag-bit mapping.

Reading BMS switch states

?BMS or GetValue(_BMS)

Returns the states of all BMS switches. Each switch may be enabled or disabled. For extracting each switch's state consult note (*2) presenting the switch-bit mapping.

Reading BMS SoC

?BSC or GetValue(_BSC)

Returns the battery's state of charge in per cent. The received value will be in the range 0-100.

TABLE 19. MultiPWM Configuration and Transmitted Data

Command	Arguments	Description
Configuration Commands		
^PWMM uu	uu 0, 1	Uu = 0 (Disabled), 1 (Roboteq MultiPWM)
Query Commands		
?BMC	-	Ah stored in battery
?BMF	-	BMS status (*1) flags
?BMS	-	BMS switch states (*2)
?BSC	-	State of charge of the battery in per cent

BMC, BMF, BMS and BSC are also available commands in Roboteq motor controllers when connected on a BMS via MultiPWM. Upon establishing connection these values will be always available and up to date on both BMS and controller.

(*1) BMS Status: Consists of important status and faults flags. After converting the received value in binary formatting the bit mapping is the following. If any of the following flags have been triggered the corresponding bit will be set (1).

TABLE 20. BMS Status - BMF Command Response

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Internal Fault	Config Error	Bad State of Health	Short Load or Inv Charger	Over Current Error Set	Amp Trigger Set	Over or Under Voltage Error Set	Unsafe Temperature

(*2) BMS Switch States: Includes information about the BMS switch states. After converting the received value in binary formatting the bit mapping is the following. When each corresponding bit is 1 means that the switch is Enabled (ON) and when that bit is 0 that the switch is Disabled (OFF).

TABLE 21. BMS Switch States - BMS Command Response

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Rsvd	Rsvd	Bluetooth Vcc Switch	Rsvd	Brake Resistor / Aux Switch	Charger Switch	Load Switch	Pack Switch

USB

USB is the most reliable method for communication with the BMS. When users connect the system to a PC, the PC Utility will automatically recognize the BMS model. At that time all internal variables can be accessed. Using the USB connection, real time monitoring of the system can be achieved utilizing the RUN tab in the utility. All switch states, voltages, and current flow are visually presented and internal variables can be plotted on the real time graph.

Via a USB connection, user's scripts can be uploaded to the BMS from the scripting window. Custom configurations can be made and stored in the system's internal memory manually or by browsing through the configuration tree.

CANbus Communication

Four CAN operating modes are available in RoboteQ's BMS10x0x system:

1. RawCAN
2. MiniCAN
3. CANopen
4. RoboCAN

Detailed documentation of all four CAN operating modes can be found in RoboteQ's CAN manual. A short description of the four modes is listed below. Be aware when using RS485 since the communication lines are shared with CAN. When CANbus is enabled the RS485 is automatically disabled. To enabled again RS485, CAN mode must be set as Off.

RawCAN is a low-level operating mode giving total read and write access to CAN frames. It is recommended for use in low data rate systems that do not hold to any specific standard. CAN frames are typically built and decoded using the MicroBasic scripting language.

MiniCAN is greatly simplified subset of CANopen, allowing, within limits, the integration of the controller into an existing CANopen network. This mode requires MicroBasic scripting to prepare and use the CAN data.

CANopen is the full Standard from CAN in Automation (CIA), based on the DS302 specification. This is the best mode to use if full compliance with the CANopen standard is a primary requisite.

For all BMS command IDs mapped at CANOpen consult the BMS EDS file.

Detailed information about Roboteq's CANOpen implementation may be found at Roboteq's Motor Controller User Manual in the corresponding section.

RoboCAN is a RoboteQ proprietary meshed networking protocol allowing multiple RoboteQ devices to operate together as a single system. This protocol is extremely simple and lean, yet practically limitless in its abilities. It is the preferred protocol to use by users who want to make multiple controllers work together with the minimal effort.

MiniCAN TPDOs

TABLE 22. BMS MiniCAN TPDOs

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
TPDO1	State	Status	Ah	SoC	Time Left Mins		Pack Volt	
TPDO2	VAR1				VAR2			
TPDO3	VAR3		VAR4		VAR5		VAR6	
TPDO4	VAR7		VAR8		VAR9		VAR10	

For all variables that are either 16 or 32 bit the reconstruction process is the following.

For 16-bit variables: MSByte = Byte_{i+1} and LSByte = Byte_i.

Example: VAR3 = Byte2 << 8 | Byte1
VAR4 = Byte4 << 8 | Byte3 etc.

For 32-bit variables the process is similar.

Example: VAR1 = Byte4 << 24 | Byte3 << 16 | Byte2 << 8 | Byte1

Ah: Corresponds to the integer part of your stored Ah in your pack. For 25.2 Ah you get 25 (decimal).

Status: Consists of important status and faults flags. After converting the received value in binary formatting the bit mapping is the following.

TABLE 23. Status Byte Bit Mapping

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Internal Fault	Config Error	Bad State of Health	Short Load or Inv Charger	Over Current Error Set	Amp Trigger Set	Over or Under Voltage Error Set	Unsafe Temperature

State: Includes information about the BMS switch states. After converting the received value in binary formatting the bit mapping is the following.

TABLE 24. State Byte Bit Mapping

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Rsvd	Rsvd	Bluetooth Vcc Switch	Rsvd	Brake Resistor / Aux Switch	Charger Switch	Load Switch	Pack Switch

Marked with 0 are the disabled switches and with 1 the switches that are currently enabled.

Note that charger's switch is bidirectional (if disabled no current may flow) whereas Pack switch will only cut current flowing into the pack and Load switch will only cut current going to the Load.

MiniCAN RPDOs

TABLE 25.BMS MiniCAN RPDOs

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
RPDO1	VAR11				VAR12			
RPDO2	VAR13				VAR14			
RPDO3	VAR15		VAR16		VAR17		VAR18	
RPDO4	BVar 33-40	BVar 41-48	BVar 49-56	BVar 57-64				

CANOpen

Service Data Object (SDO) Read/Write Messages

Runtime queries and runtime commands can be sent to the controller in real-time using the expedited SDO messages.

SDO messages provide generic access to Object Dictionary and can be used for obtaining parameter values on an irregular basis due to the excessive network traffic that is generated with each SDO request and response message.

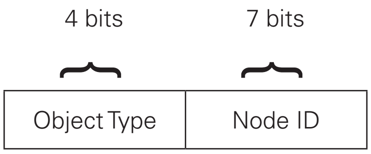
The list of commands accessible with SDO messages can be found in the "Object Dictionary" section.

Transmit Process Data Object (TPDO) Messages

Transmit PDO (TPDO) messages are one of the two types of PDO messages that are used during operation.

TPDOs are runtime operating parameters that are sent automatically on a periodic basis from the controller to one or multiple nodes. TPDOs do not alter object data; they only read internal controller values and transmit them to the CAN bus.

TPDOs are identified on a CANopen network by the bit pattern in the 11-bit header of the CAN frame.



- TPDO1: 0x180 + Node ID
- TPDO2: 0x280 + Node ID
- TPDO2: 0x380 + Node ID
- TPDO2: 0x480 + Node ID

CANopen allows up to four TPDOs for any node ID. Unless otherwise specified in the product datasheet, TPDO1 to TPDO4 are used to transmit up to 8 user variables which may be loaded with any operating parameters using MicroBasic scripting.

Each of the 4 TPDOs can be configured to be sent at user-defined periodic intervals which can be configured using the PC utility or through CTPS configuration command.

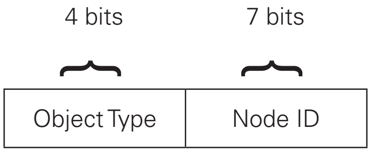
TABLE 26. BMS CANOpen TPDOs

TPDO	Object Index-Sub	Size	Object Mapped
TPDO 1	0x213A-00	U8	BSC – Battery state of charge.
	0x2141-00	U8	BMC – BMS state of charge.
	0x2142-00	U8	BMF – BMS status flags.
	0x2143-00	U8	BMS – BMS operational state.
TPDO 2	0x2106-01	S32	User VAR 1.
	0x2106-02	S32	User VAR 2.
TPDO 3	0x2106-03	S32	User VAR 3.
	0x2106-04	S32	User VAR 4.
TPDO 4	0x2106-05	S32	User VAR 5.
	0x2106-06	S32	User VAR 6.

- U8: unsigned 8-bits.
- S32: signed 32-bits.

Receive Process Data Object (RPDO) Messages

RPDOs are configured to capture runtime data destined to the controller. RPDOs are CAN frames identified by their 11-bit header.



- TPDO1: 0x200 + Node ID
- TPDO2: 0x300 + Node ID
- TPDO2: 0x400 + Node ID
- TPDO2: 0x500 + Node ID

Roboteq CANopen implementation supports RPDOs. Unless otherwise specified in the product's datasheet, data received using RPDOs are stored in 4 user variables from where they can be processed using MicroBasic scripting.

TABLE 27. BMS CANOpen RPDOs

RPDO	Object Index-Sub	Size	Object Mapped
RPDO 1	0x2005-09	S32	User VAR 9.
	0x2005-0A	S32	User VAR 10.
RPDO 2	0x2005-0B	S32	User VAR 11.
	0x2005-0C	S32	User VAR 12.
RPDO 3	0x2005-0D	S32	User VAR 13.
	0x2005-0E	S32	User VAR 14.
RPDO 4	0x2005-0F	S32	User VAR 15.
	0x2005-10	S32	User VAR 16.
• S32: signed 32-bits.			

Object Dictionary

The CANopen dictionary shown in this section is subject to change. The CANopen EDS file can be downloaded from the roboteq web site.

The Object Dictionary given in the table below contains the runtime queries and runtime commands that can be accessed with SDO/PDO messages during controller operation.

Runtime Commands

Index	Sub (hex)	Entry Name	Type & Access	Command Name
0x2005	01-vv (1)	Set user integer variable.	S32 WO	VAR
0x2008	00	Set all digital out bits.	U8 WO	DS
0x2009	00	Set individual digital out bit.	U8 WO	D1
0x200A	00	Reset individual digital out bit.	U8 WO	D0
0x200C	00	Emergency shutdown.	U8 WO	EX
0x200D	00	Release shutdown.	U8 WO	MG
0X2015	00-bb (2)	Set user Boolean variable.	U8 WO	B
0x2017	00	Save configuration to flash.	U8 WO	EES
0x2018	00	Run MicroBasic script.	U8 WO	R

Runtime Queries

Index	Sub (hex)	Entry Name	Type & Access	Command Name
0x2100	01	Read Amps	S16 RO	A
0x2106	01-vv (1)	Read user integer variable.	S32 RO	VAR
0x210D	01-03+cc(3)	Read battery voltage (01), load voltage (02), charge voltage (03) and each cell voltage (04, 05, …)	U16 RO	V
0x210E	00	Read all digital inputs.	U32 RO	D
0x210F	00-01+tt (4)	Read MCU temperature (01) and each transistor temperature (02, 03, …).	S8 RO	T
0x2111	00	Read status flags.	U8 RO	FS
0x2112	00	Read fault flags.	U8 RO	FF
0x2113	00	Read current digital outputs.	U8 RO	DO
0x2115	00-bb (2)	Read user Boolean variable.	U8 RO	B
0x2119	00	Read time.	S32 RO	TM
0x2133	00	Read script checksum.	U32 RO	SCC
0x2134	00	Read if node is alive.	U8 RO	ICL
0x2137	01-04	Read firmware ID (Version, Month, Day, Year).	U16 RO	FIN
0x2139	00	Read capacity runtime.	U16 RO	CRT
0x213A	00	Read battery's state of charge.	U8 RO	BSC
0x213B	00-ss (5)	Read internal switch control.	U8 RO	SWS
0x2140	00	Read the total charge/discharge cycles.	U16 RO	CHD
0x2141	00	Read BMS state of charge.	U8 RO	BMC
0x2142	00	Read BMS status flags.	U8 RO	BMF
0x2143	00	Read BMS operational state.	U8 RO	BMS
0x2144	00	Read battery's state of health.	U8 RO	SOH

(1) vv: maximum number of user integer variables.
(2) bb: maximum number of user Boolean variables.
(3) cc: number of battery cells.
(4) tt: number of internal temperature sensors.
(5) ss: number of internal switches.

RS485

RS485 is implemented for a half duplex communication user configurable baud rate. Since the communication lines are shared with CAN, RS485 cannot be used while CAN is enabled.

RS485 is automatically enabled when CAN mode is set to Off.

User need to be aware when wiring the RS485 communication lines. It is important to keep in mind that no matter which line is named A or B (these are the usual identifications used for RS485 lines) the positive line should be connected to the positive and negative to negative. Example: two devices may have the RS485 lines marked as A+ & B- and A- & B+ respectively. In this case A+ should be connected to B+ of the other device and B- to A-.

In RoboteQ's products, RS485A line is the Direct (+) line and RS485B the Inverted (-).

TABLE 28. Low Level RS845 Bit Rate Configuration

Command	Arguments	Description	Examples
^RSBR xx	Bit Rate Option	Configure Serial Bit Rate and Inversion	^RSBR 0 Xx 0: 115200 1: 57600 2: 38400 3: 19200 4: 9600 5: Inv 115200 6: Inv 57600 7: Inv 38400 8: Inv 19200 9: Inv 9600

Modbus ASCII

Modbus is a serial communication protocol designed mainly for industrial environments. It is open sourced and easy to set up and maintain. This make it an ideal communication protocol for industrial applications.

The BMS10x0x supports the Modbus ASCII protocol over the RS485 communication lines. To enable Modbus, use the ^dmod 4 command and ^dmod 0 to disable it.

More information on RoboteQ's Modbus solution may be found in the Modbus documentation uploaded in the official web page (www.roboteq.com).

TABLE 29. Low Level Modbus Configuration Commands

Command	Arguments	Description	Examples
^DMOD xx	Modbus Mode	Set Modbus Mode	^DMOD 4 xx 0: Off 3: RS232 ASCII 4: RS485 ASCII
^MNOD xx	Node ID	Set Modbus Node ID	^MNOD 1

Bluetooth BLE

RoboteQ offers the option of an embedded low energy Bluetooth module in the BMS10x0x. With Bluetooth, all variables are accessible using our Smartphone application which is a simplified version of the PC Utility. The mobile app is a very useful feature for monitoring the battery's status and configuring all BMS parameters. Since the system may not always be easily accessible the app is ideal for quick system status reviews and troubleshooting in case of errors.

In the case of a failure condition, users may minimize their exposure to hazardous situations by inspecting the cause of the damage before having to, for example, get close to a shorted system or problematic cells that may be dangerous.

Real Time Clock

The BMS features a real time clock allowing the user to keeping track of time. This feature may be used for marking time stamps when logging BMS parameters or events, triggering events on a specific time during the day, week or month, during specific periods of time etc.

An example of an application like these would be for the BMS to save once a month the maximum available capacity of the battery into flash. This could be used to keep track of the rate of degradation of the battery.

Setting Time

When configuring the Year only the 2 last digits of the desired year value should be set. Example, 2018 should be %STIME 6 18, for 2019 use %STIME 6 19 etc.

Command	Arguments	Description	Example
%STIME uu vv	uu = 1,…,6 1: Hours 2: Mins 3: Secs 4: Date 5: Month 6: Year (2018 is value 18) vv = new value	Configure date and time settings.	%STIME 1 12 %STIME 6 18

Reading Time

When reading the time stamp via the console ?TM command may be used to get the whole time-stamp in Hours : Mins : Sec : Date : Month : Year formatting, or ?TM uu to get each individual value.

Even though when configuring the time only the last 2 digits are used in Year configuration, when reading, Year will be returned written in full.

Command	Arguments	Description	Example
?TM uu	uu = 1,…,6 1: Hours 2: Mins 3: Secs 4: Date 5: Month 6: Year	Returns the date/time values.	?TM 6 + TM = 2018

Mechanical Dimensions

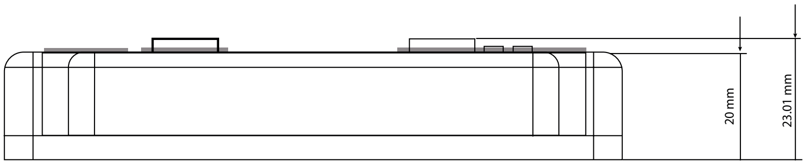


FIGURE 23. BMS10x0 Side Dimensions

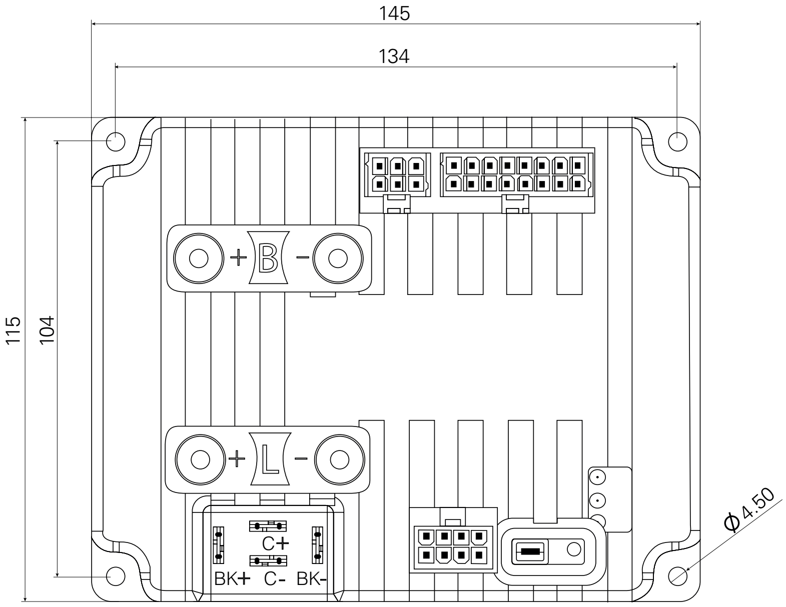


FIGURE 24. BMS10x0 Top Dimensions

Appendix

BMS Command Reference Summary

The BMS10x0x is highly configurable and accepts a number of commands via the USB port and optional Bluetooth interfaces. This is useful for reading operational data, setting configuration sending commands and performing maintenance. The commands are categorized in four groups depending on their functionality and can be found in Tables 30-33, below.

Real Time Queries

Real Time queries are commands used for reading BMS10x0x and battery data, such as Digital inputs, Voltage, Amperage, etc. These commands are executed by first typing the "?" character followed by the name of the command.

If the response is an array of values, a specific element of the array may be requested by indicating its index; for example ?A returns an array, whereas ?A 1 returns the first value of the array only.

Each time a query is executed, it is stored in a history buffer and may therefore be automatically repeated at a periodic rate using the "#" character with the following syntax:

#	repeat last query in queue
# nn	repeat last queries ever nn ms. Example: **# 100** to execute one query from the history queue every 100ms
# C	clear queue

TABLE 30. Real Time Query Commands

Command	Arguments	Description	Examples
A	Index (optional)	Read Battery Amps, Ah, Max Ah, SoC, Charge Amps, Load Amps	?A 4 A=SoC
VAR	Index (optional)	Read User Integer Variable	?VAR
V	Index (optional)	Read Battery, Load, Charger and Cell Voltages (up to 15 cell voltages, 18 value array)	?V 4 V=Cell_1_Voltage
D		Read All Digital Inputs	?D
DI		Read Individual Digital Inputs	?DI
T		Read Internal and External Temperatures	?T
FS		Read Status Flags	?FS
B		Read User Boolean Variable	?B
DO		Read Current Digital Outputs	?DO
TM		Read Time	?TM
LK		Read Lock Status	?LK
FID		Read Firmware ID String	?FID
CL	Index (optional)	Read RoboCAN Active Nodes	?CL
CAN	Index (optional)	Read CAN Raw Received Data	?CAN
CF		Read CAN Received Frames	?CF
UID		Read MCU Unique ID	?UID
CRT		Read Time Left (Until full charge or depletion)	?CRT
CHD		Read Charge Pack Cycles	?CHD
BMC		Read BMS SoC	?BMC
BMF		Read BMS Status Flags	?BMF
BMS		Read BMS Operational State	?BMS

Real Time Commands

A Real Time Command is used for instructing the BMS10x0x to perform an action. This type of command is executed by typing the "!" character before the name of the command. In Table 30, below, examples are shown for each command.

TABLE 31. Real Time Commands

Command	Arguments	Description	Examples
VAR	Index, Value	Read User Integer Variable	!VAR 3 125
DS	Binary Value	Set all Digital Out bits	!DS 3
D1	Output Bit Band Index	Set Individual Digital Out bits	!D1 1
D0	Output Bit Band Index	Reset Individual Digital Out bits	!D0 1
R		Run MicroBasic Script	!R
EX		Emergency Disconnect	!EX
MG		System Reconnect	!MG
B	BB Index, Bit Value	Set User Boolean Variable	!B 2 1 (Set BB 2) !B 2 0 (Res BB 2)
CS		Send Raw CAN Frame	!CS
EES		Save Config to Flash	!EES

Configuration Commands

Configuration commands are used to read or modify System configuration parameters. They begin with the "~" character for reading and the "^" character for writing. Table 31, below, shows the list of supported configuration commands.

Be aware, however, it is far easier to use the BMS10x0x PC Utility menus, which are especially designed for inspection and changing configurations.

If changing manually, remember to save the new configuration to flash with the %EESAV. Otherwise, the BMS will revert to the previously active configuration.

TABLE 32. Configuration Commands.

Command	Arguments	Description	Examples
EE	Index, Signed 16 Bit Value	User EE Storage	^EE 3 536
THLD	Channel, Value	Set Response Time for Short	^THLD 1 0 (Load, Quick) ^THLD 2 3 (Charge, Off)
ECHOF	Echo Status	Configure Echo	^ECHOF 1 0: Echo On 1: Echo Off
RSBR	Bit Rate Option	Configure Serial Bit Rate and Inversion	^RSBR 0 0: 115200 1: 57600 2: 38400 3: 19200 4: 9600 5: Inv 115200 6: Inv 57600 7: Inv 38400 8: Inv 19200 9: Inv 9600
DINL	Input Index, Active Level	Digital Input Active Level (If inverted)	^DINL 2 1 (Din 2 inverted)
DOA	Output Index, Trigger	Digital Output Active When	^DOA 1 0 0: Never 1: Over Temp 2: Over Volt 3: Overload/Short
DOL	Output Index, Level	Digital Output Level Logic	^DOL 1 1
ATRIG	Channel, Amps	Amp Trigger Level	^ATRIG 1 150
ATGD	Channel, Delay ms	Amp Trigger Action Delay	^ATGD 1 500
BRUN		MicroBasic Auto Start	^BRUN 1

Command	Arguments	Description	Examples
CEN	CAN Mode Option	CAN On	^CEN 0 0: Disabled 1: CANOpen 2: MiniCAN 3: RawCAN 4: RoboCAN
CNOD	Address	CAN Node Address	^CNOD 6
CBR	Bit Rate Option	CAN Bit Rate	^CBR 3 0: 1000k 1: 800k 2: 500k 3: 250k 4: 125k
CHB	Heartbeat ms	CAN HeartBeat	^CHB 100
CAS		CAN PDO Autostart	^CAS 1
CLSN	Listen node	CAN Listening COB	^CLSN 4 0: Disabled 1-127: Nodes
CSRT	SendRate ms	CAN SendRate	^CSRT 100
CTPS	TPDO number, SendRate ms	CAN TPDO SendRate	^CTPS 1 100 (if ms = 0 no transmit)
SCRO	Print Output Option	Select Print Output Port for Scripting	^SCRO 0 0: Last used 1: Serial 2: USB
BEE	Index, Value	User BEE Storage	^BEE 1 55
CTT		CAN Open Type	
BCHT	Cell Index, mV	Cell threshold high	^BCHT 1 4200
BCLT	Cell Index, mV	Cell threshold low	^BCLT 1 3200
BTHT	Sensor Index, Celsius	Temperature Max on Discharge	^BTHT 2 55
BTLT	Sensor Index, Celsius	Temperature Max on Charge	^BTLT 3 40
BAHT	Amps	Pack Amp Rating on Discharge	^BAHT 50
BALT	Amps	Pack Amp Rating on Charge	^BALT 15
NOMC	Ah*10	Battery Nominal Capacity	^NOMC 100 (100 = 10 Ah)
CNOV	Value mV	Nominal Cell Voltage	^CNOV 3700
CCHT	Chemistry Option	Cell Chemistry Type	^CCHT 1 0: LiPo 1: LiFePO4 2: Custom
BPRC	Value mV	Balance Precision	^BPRC 10

Command	Arguments	Description	Examples
ISRG	Aux Output Option	Aux Output Function	^ISRG 0 0: Disable 1: Brake Resistor 2: Fan 3: Heater
CRES	Value in uOhm	Cell Internal Resistance	^CRES 1 (0.001 mOhm)
STYP	Sensor Index, Type	Select Temperature Sensor Type	^STYP 2 1 0: Disable 1: 10k NTC
MXRTR	Channel, Number Rtr	Max Retry Times	^MXRTR 2 0
REDL	Channel, Time ms	Delay Between Retries	^REDL 2 1000
MINOP	Sensor Idx, Celsius	Minimum Operating Temperature	^MINOP 2 -10
DMOD	Modbus Mode	Set Modbus Mode	^DMOD 4 0: Off 3: RS232 ASCII 4: RS485 ASCII
MNOD	Node ID	Set Modbus Node ID	^MNOD 1

Maintenance Commands

Maintenance commands are used for performing maintenance functions on RoboteQ’s BMS10x0x. The commands seen in Table 32, below, can be used by typing the “%” character before the command name.

TABLE 33. Maintenance Commands

Command	Arguments	Description	Examples
RESET	Key	Reset Controller	%RESET 321654987
SLD	Key	Load Microbasic scripts	%SLD 321654987
EERST	Key	Restore Factory Defaults	%EERST 321654987
EESAV		Save Parameters to EEPROM	%EESAV
EELD		Load Parameters from EEPROM	%EELD
CLSAV	Key	Calibrations Values to EEPROM	%CLSAV 321654987
CLRST	Key	Load Factory Default Configurations	%CLRST 321654987
LK		Lock Configuration Access	%LK
UK		Unlock Configuration Access	%UK
STIME	Hours Mins Secs	Set Time	
DFU	Key	Enter Firmware Update via USB	%DFU 321654987
BTCMD	BT Command String	Send BT Configuration Commands	%BTCMD “AT” Response “OK”