# 40/60V, 100 Amps Management System for Lithium Ion Batteries

RoboteQ's BMS10x0 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 and 60V 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 BMS10x0 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) versions

- Centralized design. No cell boards – All BMS electronics contained within the unit
- 100A max current to load. 200A peak
- 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 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 Digital Output 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:139mm x W:113mm x H:29mm
- Weight: **250g**

## Ordering References

| Part Number | Number of Cells | Max Voltage | Bluetooth |
|---|---|---|---|
| BMS1040 | 6 - 10 | 40 | No |
| BMS1040BT | 6 - 10 | 40 | Yes |
| BMS1060 | 11 - 15 | 60 | No |
| BMS1060BT | 11 - 15 | 60 | Yes |

## System Architecture

The BMS10x0 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.
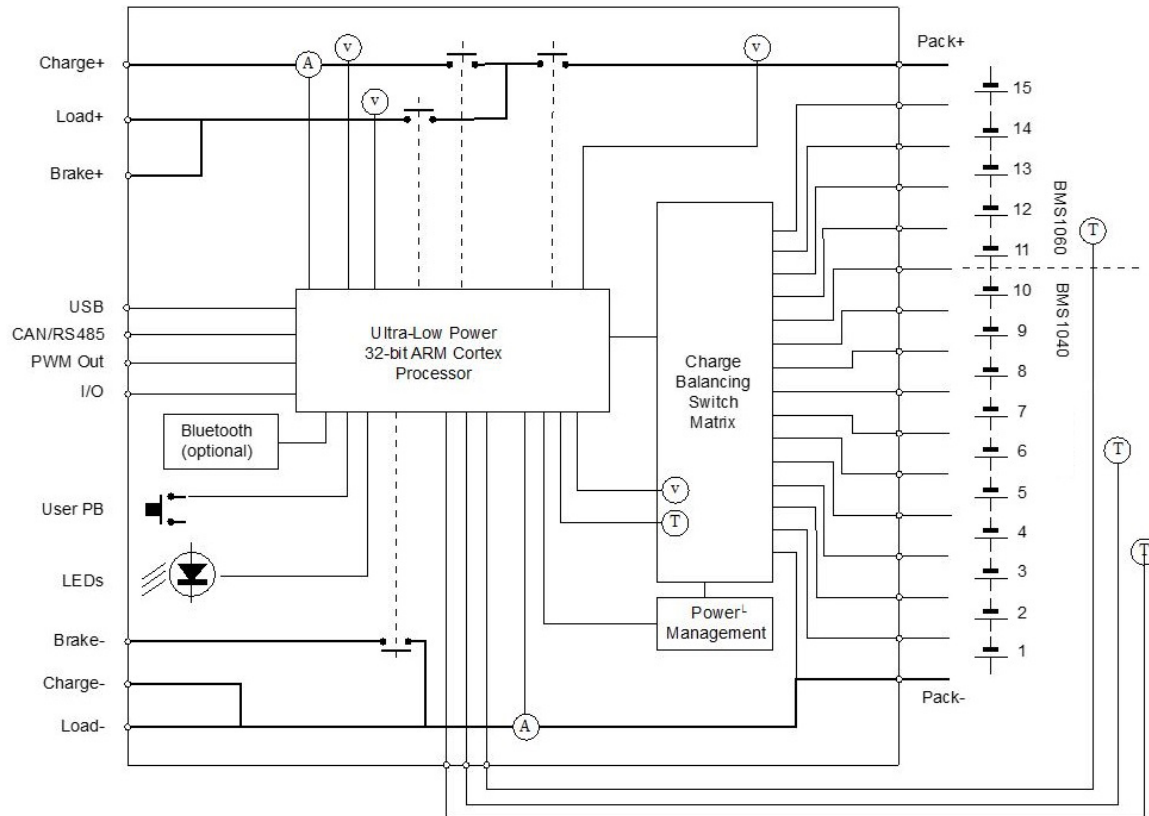


FIGURE 1. Balancing and Switches

## 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.
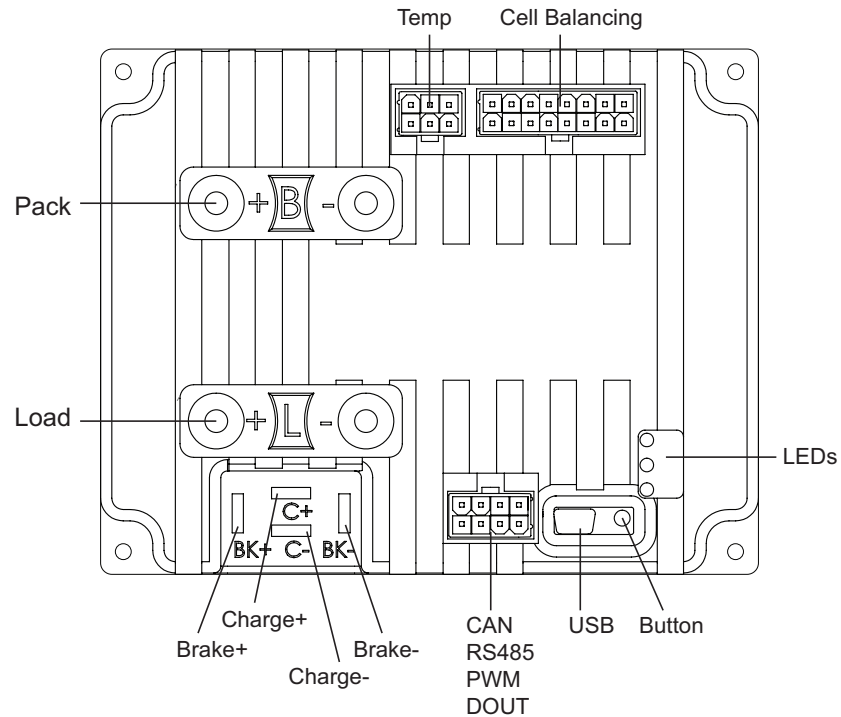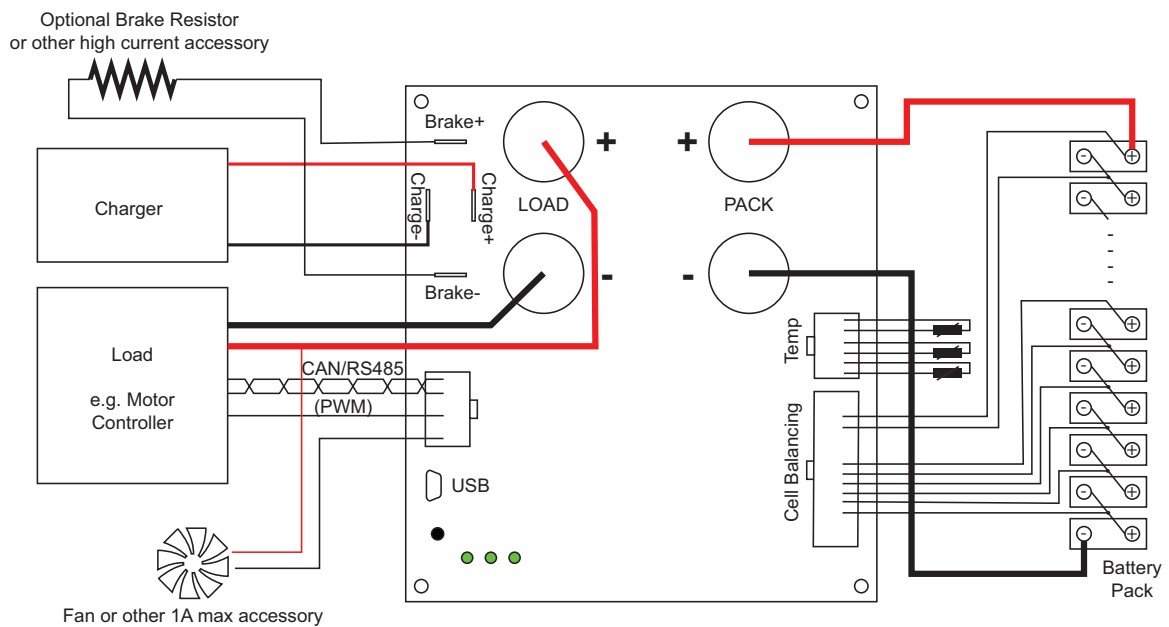


FIGURE 2. BMS Board Connectors



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 100A 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.

## Charge+/Charge-Connectors to Battery Charger

Two Faston connectors are used to connect the BMS to a charger. A 50A built-in switch will disconnect the charger from the battery when it is fully charged or if any unsafe condition occurs.

## 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.

## 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.
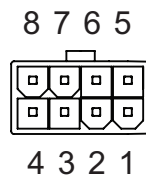
8 7 6 5

4 3 2 1

FIGURE 4. User IO Male Connector on BMS

TABLE 1.

| Pin | Signal | Description |
|-----|--------|-------------|
| 1 | CANH / RS485A | High level CAN bus line / RS485 Direct (+) |
| 2 | CANL / RS485B | Low level CAN bus line / RS485 Inverted (-) |
| 3 | DOUT1 | 1A Open Drain output 1 |
| 4 | DOUT2 | 1A Open Drain output 2 |
| 5 | VCC | Optocoupler VCC in - 3.3 to 5 V typical operation |
| 6 | PWM | Optocoupled PWM output |
| 7 | GND | Optocoupler GND in |
| 8 | NC | Reserved |

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 availalbe for connecting low-power accessories.

The BMS10x0 also implements a RoboteQ proprietary single wire PWM communication protocol for exchanging the most vital battery information with RoboteQ's motor controllers. 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

One push button and three LED indicators are provided for quick diagnostics such as charge level, communication, and other useful information. Different flashing patterns are used to code the various information type.

## USB Connector

A USB port is present for connecting the BMS to a PC running the dedicated PC Utilityfor 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 BMS10x0 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..
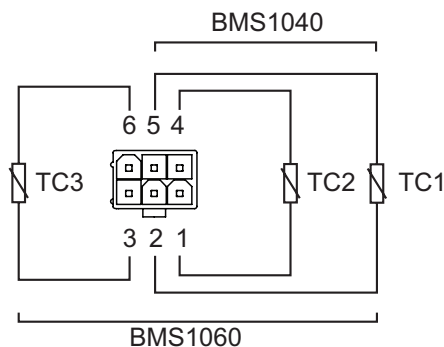


FIGURE 5. Temperature Thermistor connections

TABLE 2.

| Pin | Signals BMS1040 | Signals BMS1060 | Description |
|---|---|---|---|
| 1 | TC1P | TC1P | Thermistor 1 terminals |
| 4 | TC1N | TC1N | |
| 2 | TC2P | TC2P | Thermistor 2 terminals |
| 5 | TC2N | TC2N | |
| 3 | NC | TC33P | Thermistor 3 terminals |
| 6 | NC | TC3N | |

## 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.

| Cells | LiPo Minimum Empty | LiPo Nominal 80%-20% SoC | LiPo Maximum Full Charge | LiFe Minimum Empty | LiFe Nominal 80%-20% SoC | LiFe Maximum Full Charge | |
|---|---|---|---|---|---|---|---|
| 6 | 18 | 22.2 | 25.2 | 15 | 19.8 | 21.6 | BMS1040 |
| 7 | 21 | 25.9 | 29.4 | 17.5 | 23.1 | 25.2 | BMS1040 |
| 8 | 24 | 29.6 | 33.6 | 20 | 26.4 | 28.8 | BMS1040 |
| 9 | 27 | 33.3 | 37.8 | 22.5 | 29.7 | 32.4 | BMS1040 |
| 10 | 30 | 37 | 42 | 25 | 33 | 36 | BMS1040 |
| 11 | 33 | 40.7 | 46.2 | 27.5 | 36.3 | 39.6 | BMS1060 |
| 12 | 36 | 44.4 | 50.4 | 30 | 39.6 | 43.2 | BMS1060 |
| 13 | 39 | 48.1 | 54.6 | 32.5 | 42.9 | 46.8 | BMS1060 |
| 14 | 42 | 51.8 | 58.8 | 35 | 46.2 | 50.4 | BMS1060 |
| 15 | 45 | 55.5 | 63 | 37.5 | 49.5 | 54 | BMS1060 |

## Connecting the cells

A 16-pin connector is used for the necessary monitoring and balancing connections. The drawings in Figures 6-8, below, show how to connect 6 to 10 cell packs on the BMS1040, and 11 to 15 cells packs on the BMS 1060.
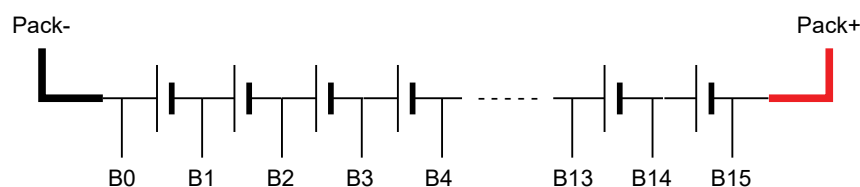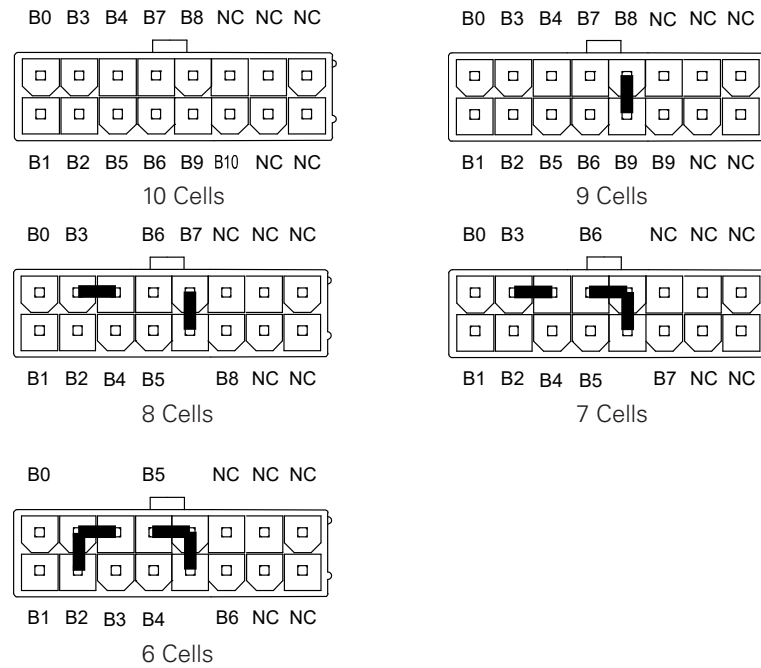


FIGURE 6. Battery Cell Configuration

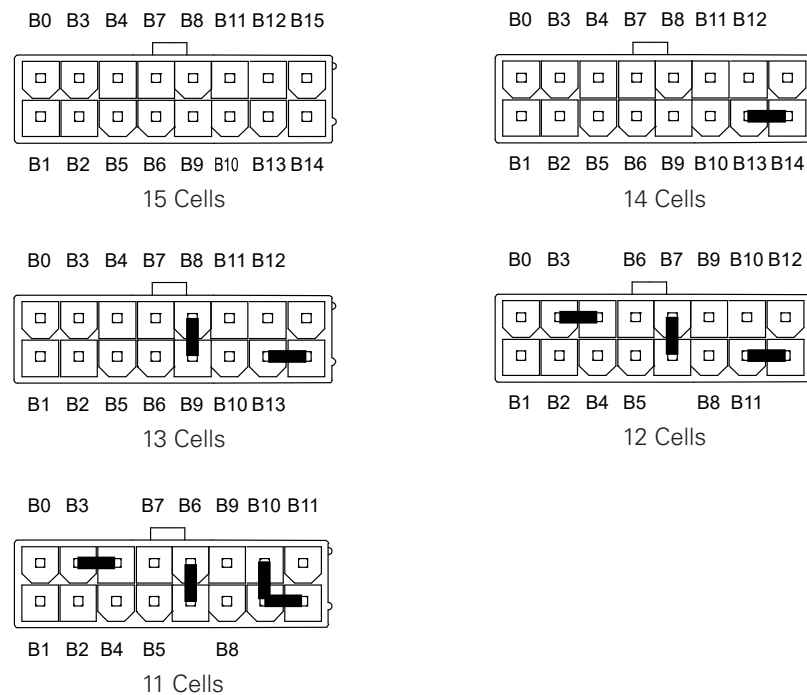FIGURE 7. Pin-out Connections for 6 to 10 Cell Packs



FIGURE 8. Pin-out Connections for 11 to 15 Cell Packs

## Absolute Maximum Ratings

TABLE 4.

| | | Model | Typ | Min | Max |
|---|---|---|---|---|---|
| $V_I$ 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 | | | 7.2 V |
| | B5 - B0, B10 - B6 | BMS1040 | | | 36 V |
| | B5 - B0, B10 - B6, B15 - B11 | BMS1060 | | | 36 V |
| | $TS_nP - TS_nN$, with n = 1, 2 | BMS1040 | | | 3.6 V |
| | $TS_nP - TS_nN$, with n = 1, 2, 3 | BMS1060 | | | 3.6 V |
| $V_I$ & $V_O$ Input and Output voltage | Pack+ - Pack- | BMS1040 | | | 40 V |
| | | BMS1060 | | | 80 V |
| | Load+ - Load- | BMS1040 | | | 40 V |
| | | BMS1060 | | | 80 V |
| | Aux+ - Aux- (Break Resistor) | BMS10x0 | | | 75 V |
| | Charge+ - Charge- | BMS1040 | | -0.3 V | 40 V |
| | | BMS1060 | | | 80 V |
| $I_I$ & $I_O$ Input and Output current | Pack+ - Pack- | BMS10x0 | | -200 A | 200 A |
| | Load+ - Load- | | | | |
| | Aux+ - Aux- (Break Resistor) | BMS1040 | | -60 A | 60 A |
| | | BMS1060 | | -60 A | 60 A |
| | Charge+ - Charge- | BMS1040 | | | 60 A |
| | | BMS1060 | | | 50 A |
| | Digital Output n - Pack- with n = 1, 2 | BMS10x0 | | | 1 A |
| $V_I$ & $V_O$ User Connections | Digital Output 1 - Pack- | BMS10x0 | | | 60 V |
| | Digital Output 2 - Pack- | | | | |
| | VCC - GND | | 3.3 V | 2.7 V | 5.5 V |
| | CANH / RS485A - Pack- | | | -4 V | 14 V |
| | CANL / RS485B - Pack- | | | | |
| | PWM - GND | | | 2.7 V | 15 V |

## BMS Operation

### Initial Configuration

RoboteQ's BMS is preconfigured with battery profiles for specific cell chemistry types. When the cell type is specified, voltage and temperature thresholds are automatically assigned. These parameters can then be adjusted to the user's needs, but 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.

## 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 BMS10x0 to work.

## 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 BMS10x0 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. Since the pack will most probably not be at exactly 50% SoC, the system will automatically adjust the measurements so that the actual State of Charge of the battery is aligned with the measured SoC, providing the user with an accurate indication.

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.

## Charging

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

Any configurable power supply may be used as a charger, as long as it can deliver adequate current at the necessary voltage. 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.
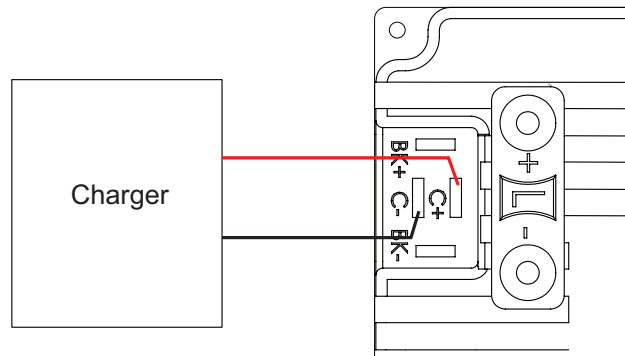


FIGURE 9. BMS Charge Configuration

## Brake Resistor

The BMS10x0 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.
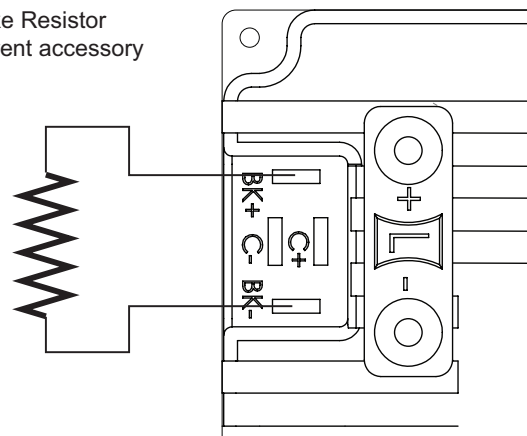


FIGURE 10. Connection to Brake Resistor or High Power Accessory

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
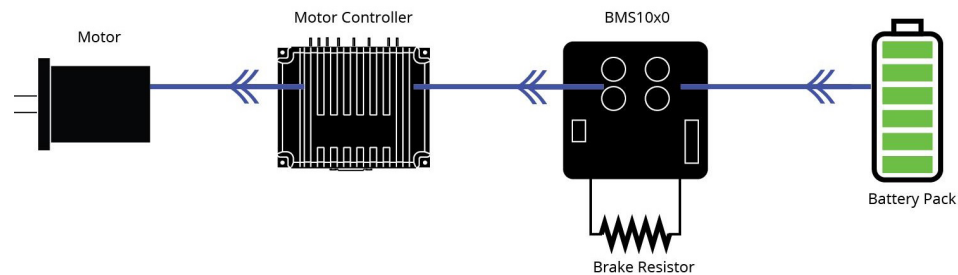
FIGURE 11. Current Flow from Pack to Load
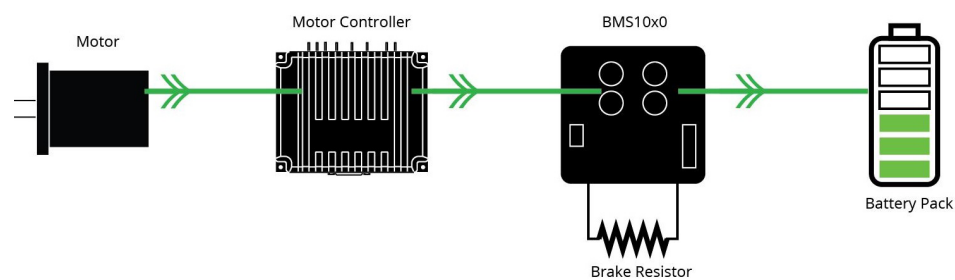
FIGURE 12. Regenerating Energy from Motor
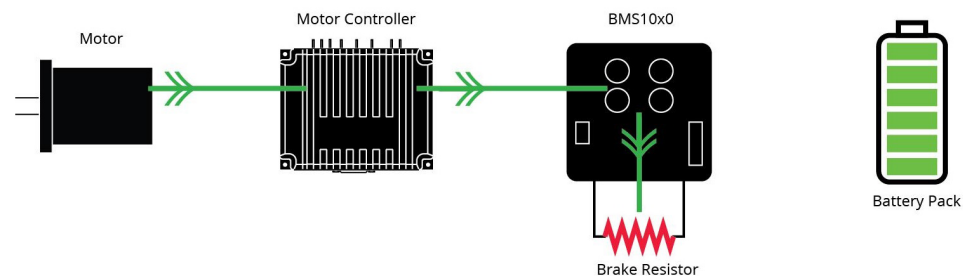
FIGURE 13. Excess Energy Out to Brake Resistor

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.
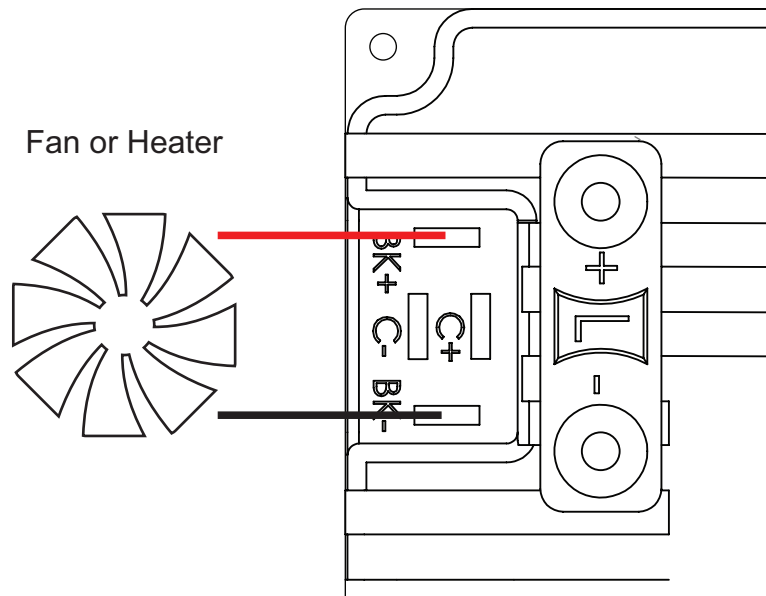


FIGURE 14. Configuration for Guaranteed Temperature Balance

## 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 BMS10x0 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 BMS10x0 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.

## Circuit Breaker

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 200 A peak current and 100 A continuous current. The BMS10x0 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 5.

| Command | Arguments | Description | Examples |
|---|---|---|---|
| ^THLD xx yy | Channel, Value | Set Response Time for Short | ^THLD 1 0<br>xx 1: Load, 2: Charge<br>yy 0: Quick<br>   1: Medium<br>   2: Slow<br>   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 |

## Temperature Sensors

An internal sensor periodically provides the user with the board temperature. Depending on the BMS10x0 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 BMS1040, two external sensors are supported, whereas the BMS1060 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.

## 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 6.

| 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.

TABLE 7.

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

## 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

As the name implies, this proprietary method uses a succession of variable duty-cycle pulses to carry vital predefined information. Using MultiPWM the battery's SoC, status flags and system state can be obtained. All data is continuously transmitted to RoboteQ's motor controllers and the values may be requested any time by the commands BMC, BMF and BMS.

Status Flag bytes consist of [SCRIPT_RUN, BALANCING, BAD_SOH, INTERNAL_FAULT, UV, OV, SC, OC]. The status represents user script running, that balancing is activated, state of health error, internal fault, under voltage, over voltage, short circuit, over current, respectively.

The 8 bits of the System State bytes are; cells are not properly connected, hard fault occurred, system is in safe lock state, system regenerates energy, discharging enabled, balancing enabled, charging enabled. These states are represented by the following, respectively: NOCONNECT, FAIL, LOCK, RGN, DCH, BALANCE, CH. Refer to Tables 8-10, below:

TABLE 8.

| Status Flags | Bit 7 | Bit 6 | Bit 5 | Bit 4 |
|--------------|-------|-------|-------|-------|
| | SCRIPT_RUN | BALANCING | BAD_SOH | INT_FAULT |
| | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| | UV | OV | SC | OC |

TABLE 9.

| System State | Bit 7 | Bit 6 | Bit 5 | Bit 4 |
|---|---|---|---|---|
| | - | NOCONNECT | FAIL | LOCK |
| | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | RGN | DCH | BALANCE | CH |

TABLE 10.

| Command | Arguments | Description | Examples |
|---|---|---|---|
| ?BMC | | Read BMS SoC | ?BMC |
| ?BMF | | Read BMS Status Flags | ?BMF |
| ?BMS | | Read BMS Operational State | ?BMS |

## 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 BMS10x0 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 bellow. 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.

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.

## 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 11.

| Command | Arguments | Description | Examples |
|---------|-----------|-------------|----------|
| ^RSBR xx | Bit Rate Option | Configure Serial Bit Rate and Inversion | ^RSBR 0<br>Xx 0: 115200<br>   1: 57600<br>   2: 38400<br>   3: 19200<br>   4: 9600<br>   5: Inv 115200<br>   6: Inv 57600<br>   7: Inv 38400<br>   8: Inv 19200<br>   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 BMS10x0 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 12.

| Command | Arguments | Description | Examples |
|---------|-----------|-------------|----------|
| ^DMOD xx | Modbus Mode | Set Modbus Mode | ^DMOD 4<br>xx 0: Off<br>   3: RS232 ASCII<br>   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 BMS10x0. 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.

## Status LEDs and Flashing Patterns

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



C  Communication LED
⏻  Power LED
⚠  Status LED

FIGURE 15. 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 16. Capacity LED Display

## Status LED

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.
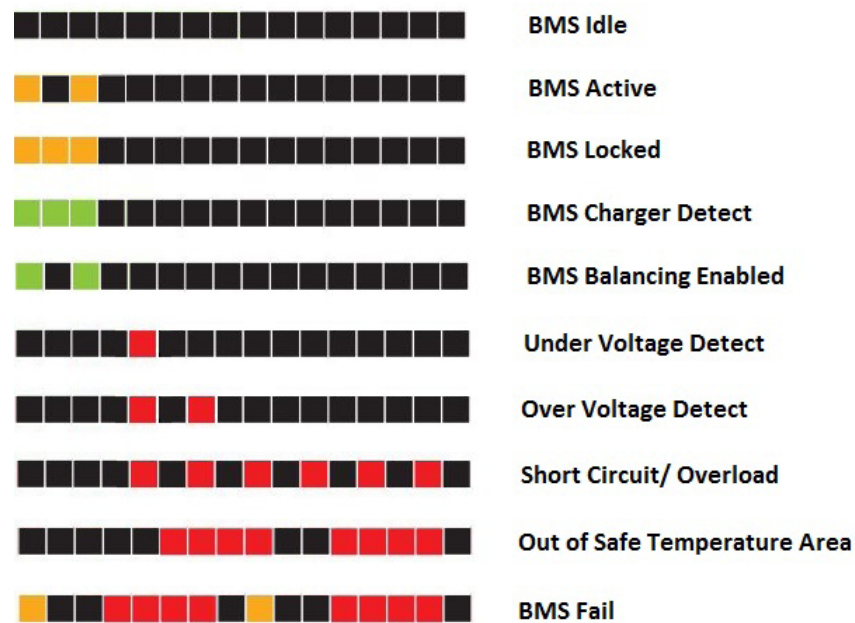


FIGURE 17. Status LED Display

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.

No USB, No CAN

USB Active, No CAN

No USB, CAN Enabled

USB Active, CAN Enabled
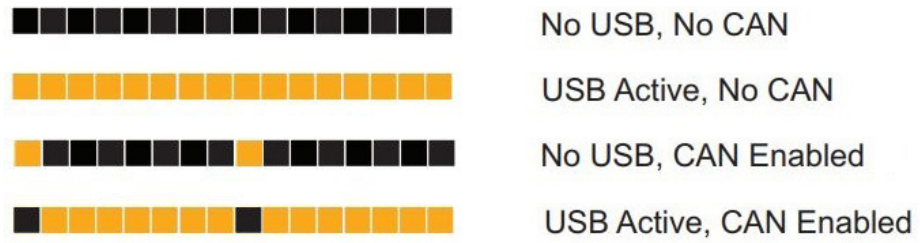
FIGURE 18. Communication LED Display

## Mechanical Dimensions

0.98"
(25mm)

0.25"
(6.3 mm)

0.3"
(7.6 mm)

0.16" (4.0mm)

FIGURE 19. BMS10x0 Side Dimensions

4.45" ( 113.0mm)

0.20" (5.0mm)

5.50" ( 139.7mm)

5.10" (129.7mm)

0.19" (5.0mm)

FIGURE 20. BMS10x0 Top Dimensions

# Appendix

## BMS Command Reference Summary

The BMS10x0 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 13-16, below.

## Real Time Queries

Real Time queries are commands used for reading BMS10x0 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.
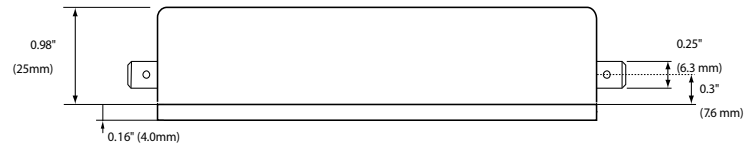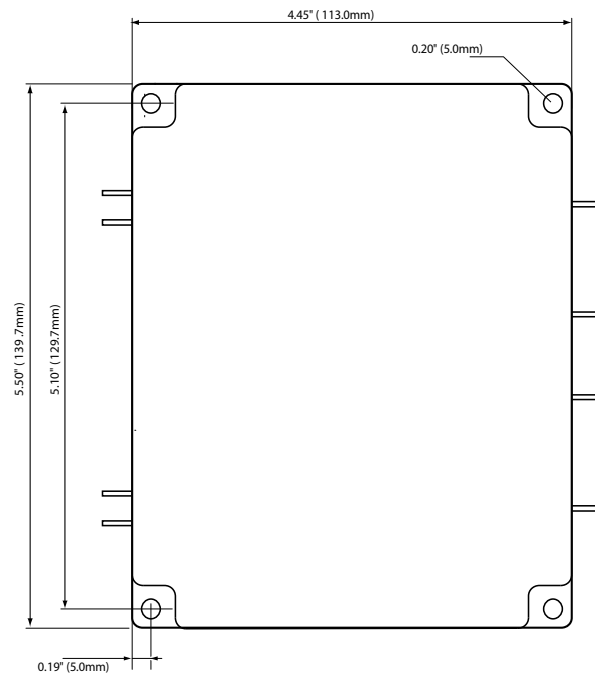
If the response is an array of values, a specific element of the array may be requested by indicating its index; for example ?A returns an array, whereas ?A 1 returns the first value of the array only.

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 13.

| Command | Arguments | Description | Examples |
|---|---|---|---|
| A | Index (optional) | Read Battery Amps, Ah, Max Ah, SoC, Charge Amps, Load Amps | ?A 4<br>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<br>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 BMS10x0 to perform an action. This type of command is executed by typing the "!" character before the name of the command. In Table 14, below, examples are shown for each command.

TABLE 14.

| 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) <br> !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 15, below, shows the list of supported configuration commands.

Be aware, however, it is far easier to use the BMS10x0 PC Utilitymenus, 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 15. Caption to come

| 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 <br> (Load, Quick) <br> ^THLD 2 3 <br> (Charge, Off) |
| ECHOF | Echo Status | Configure Echo | ^ECHOF 1 <br> 0: Echo On <br> 1: Echo Off |
| RSBR | Bit Rate Option | Configure Serial Bit Rate and Inversion | ^RSBR 0 <br> 0: 115200 <br> 1: 57600 <br> 2: 38400 <br> 3: 19200 <br> 4: 9600 <br> 5: Inv 115200 <br> 6: Inv 57600 <br> 7: Inv 38400 <br> 8: Inv 19200 <br> 9: Inv 9600 |

| Command | Arguments | Description | Examples |
|---|---|---|---|
| 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 |
| 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 |

| Command | Arguments | Description | Examples |
|---------|-----------|-------------|----------|
| CCHT | Chemistry Option | Cell Chemistry Type | ^CCHT 1<br>0: LiPo<br>1: LiFePO4<br>2: Custom |
| BPRC | Value mV | Balance Precision | ^BPRC 10 |
| ISRG | Aux Output Option | Aux Output Function | ^ISRG 0<br>0: Disable<br>1: Brake Resistor<br>2: Fan<br>3: Heater |
| CRES | Value in uOhm | Cell Internal Resistance | ^CRES 1<br>(0.001 mOhm) |
| STYP | Sensor Index, Type | Select Temperature Sensor Type | ^STYP 2 1<br>0: Disable<br>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<br>0: Off<br>3: RS232 ASCII<br>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 BMS10x0. The commands seen in Table 16, below, can be used by typing the "%" character before the command name.

TABLE 16.

| 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"<br>Response "OK" |