allocortech inc.

Lohner Smart Battery ICD

601-0050-000

Revision B

August 2021

5391 Three Notch'd Rd.
Crozet, VA 22903

https://allocor.tech

Version History

Revision	Changes
A	Initial Draft (proposal phase)
B	Revised to match final design specifications

Introduction

The allocortech inc. Lohner Smart Battery is a 6S2P/8Ah Lithium battery in a ruggedized enclosure that contains support electronics for serial communication of status and commands and a monitored push button input for enabling the output. The battery system is designed to charge and balance the 6S2P pack when not being actively discharged.

Scope of this Document

This document covers the mechanical and electrical specifications of the allocortech inc. Lohner Smart Battery.

List of Abbreviations

CAN	Controller Area Network
LiPo	Lithium-Ion Polymer (Battery)
SoC	State of Charge
USART	Universal Synchronous Asynchronous Receiver Transmitter

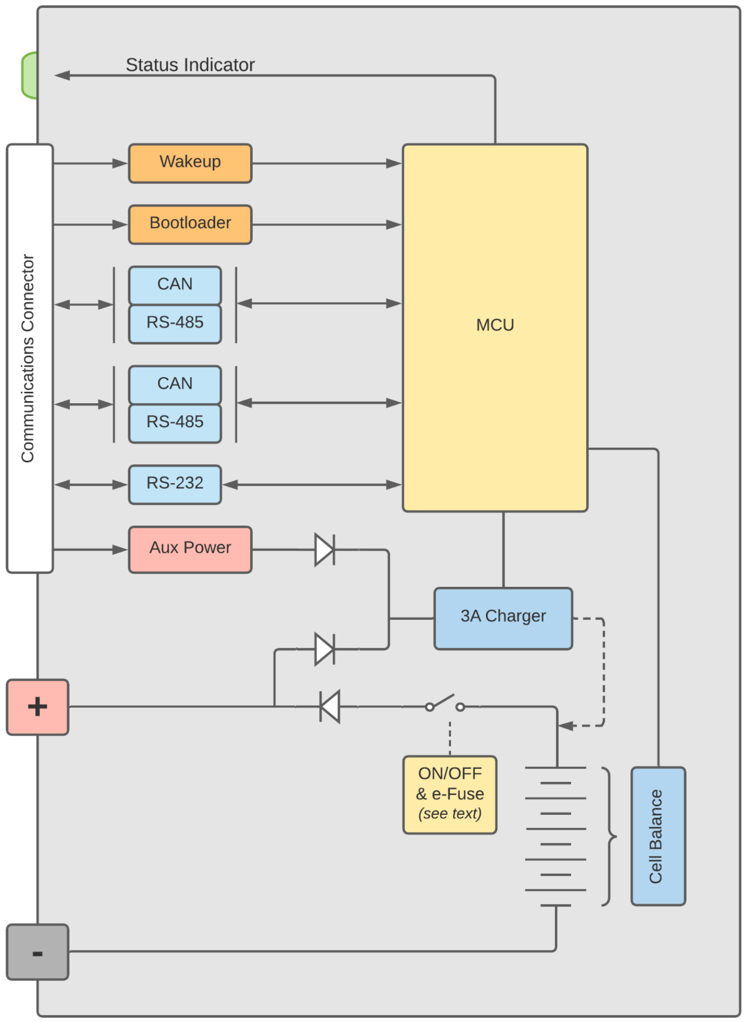
References

STMicroelectronics AN3155 USART protocol used in the STM32 bootloader

Texas Instruments bq24616 JEITA guideline compatible LiPo charger datasheet

Overview

The Lohner Smart Battery consists internally of a control PCB and a 6S2P/8Ah Lithium battery. A block diagram is shown below.



When the Lohner is at rest, the positive output terminal is electrically disconnected from the internal battery, described in this document as being in the "detached state". The MCU will operate in a low-power monitoring mode, checking for faults and monitoring the battery state-of-charge (SoC) and illuminating the externally visible fault indicator if a fault is detected. To conserve power at rest, communications interfaces are disabled.

When the Lohner is activated, via push button or via serial command, the positive output terminal will be electrically connected to the internal battery, described in this document as being the "attached state".

This primary On/Off switch also functions as a gross overload protection, and is designed to trip at approximately 200A. There are no physical fuses present in the primary battery output path.

Operation

The Lohner has five distinct operational modes:

1) Asleep: In this mode the Lohner CPU is primarily in a sleep mode, waking periodically to check the voltage of the battery. The internal battery is disconnected from the primary output terminal, and no external power is applied.
2) Awake - Unattached & Unpowered: In this mode the Lohner CPU is operating and communications are active, but no external power is supplied either from the primary output terminal or the secondary charging input, and the internal battery is disconnected from the primary output terminal.
3) Awake - Unattached & Powered: In this mode the Lohner CPU is operating and communications are active, external power is provided either on the primary output terminal or the secondary charging input, and the internal battery is disconnected from the primary output terminal. The internal battery may be charged in this state.
4) Awake - Attached & Unpowered: In this mode the Lohner CPU is operating and communications are active, but no external power is supplied either from the primary output terminal, or the secondary charging input, and the internal battery is connected to the primary output terminal.
5) Awake - Attached & Powered: In this mode the Lohner CPU is operating and communications are active, external power is provided either on the primary output terminal or the secondary charging input, and the internal battery is connected to the primary output terminal. The internal battery may be charged in this state.

There are two ways to transition the Lohner from being Asleep to being Awake:

1) Closing the circuit between the two “monitored contact closure” pins on the J1 connector. A momentary push button works well for this.
2) Providing external power, either to the primary output connector or the secondary charge input on the J1 connector.

There are two ways to transition the Lohner from being Awake to being Asleep:

1) If the Lohner is awakened by the use of the external push button, and no communications are received within a nominal timeout period.
2) The Lohner is awake and receives a “detach” command, and then no further communications are received within a nominal timeout period.

There are two ways to transition the Lohner to the attached state:

1) If the Lohner is awakened by the use of the external push button, it will automatically put itself into the attached state.
2) If the Lohner is Awake and in the detached state and receives the “attach” command.

The Lohner will stay awake under all conditions if external power is supplied through either the primary output connector or the secondary charging input.

While the Lohner is awake with the battery connected (attached) to the output terminal, the battery will remain connected to the output terminal *under all conditions* except the following two:

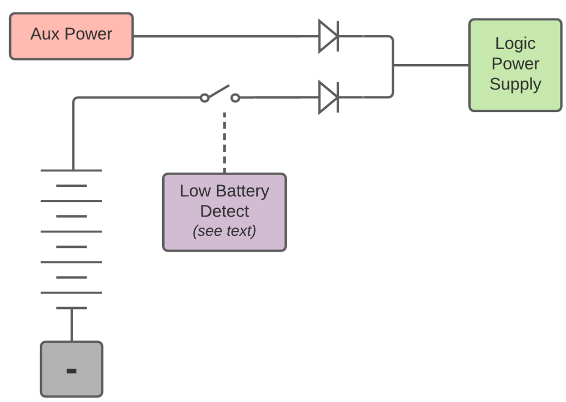
1) The battery becomes completely exhausted (to the point of internal battery damage) such that there is not enough energy left to power in the internal logic circuitry.
2) A disconnect (“detach”) command is received by the MCU.

To emphasize, once the battery is connected or attached to the output terminal, it will stay connected with no low-voltage safety interlocks or disconnects. **The battery may be drained to complete exhaustion and damage, and no hardware or software interlocks will interrupt this process.** This operation is by design and intended to provide as much energy from the battery as possible.

A state flow chart is provided in a subsequent section “Application State Machine”.

Sleep Mode Battery Protection

When the Lohner is sleeping, there is a hardware safety feature to prevent the battery from completely discharging while unattended. Again, this feature is disabled while the Lohner is in the attached state. The block diagram below indicates the power flow for the internal logic circuitry. The internal logic is powered by a low-current, high-efficiency supply, which draws directly from the internal battery, or externally supplied power, whichever has a greater voltage. If the Lohner is in sleep mode, disconnected from all external power sources, and the battery drops below about 22.2V (about 20% SoC), the battery will be disconnected from the logic power supply, and the unit will become dead / inoperative. At this point, external power must be supplied to regain MCU operation and control of the unit. This low-voltage cutoff protection is only available in sleep mode. Once the Lohner is awake and the battery is connected to the output terminal, all low-voltage safeguards are removed.



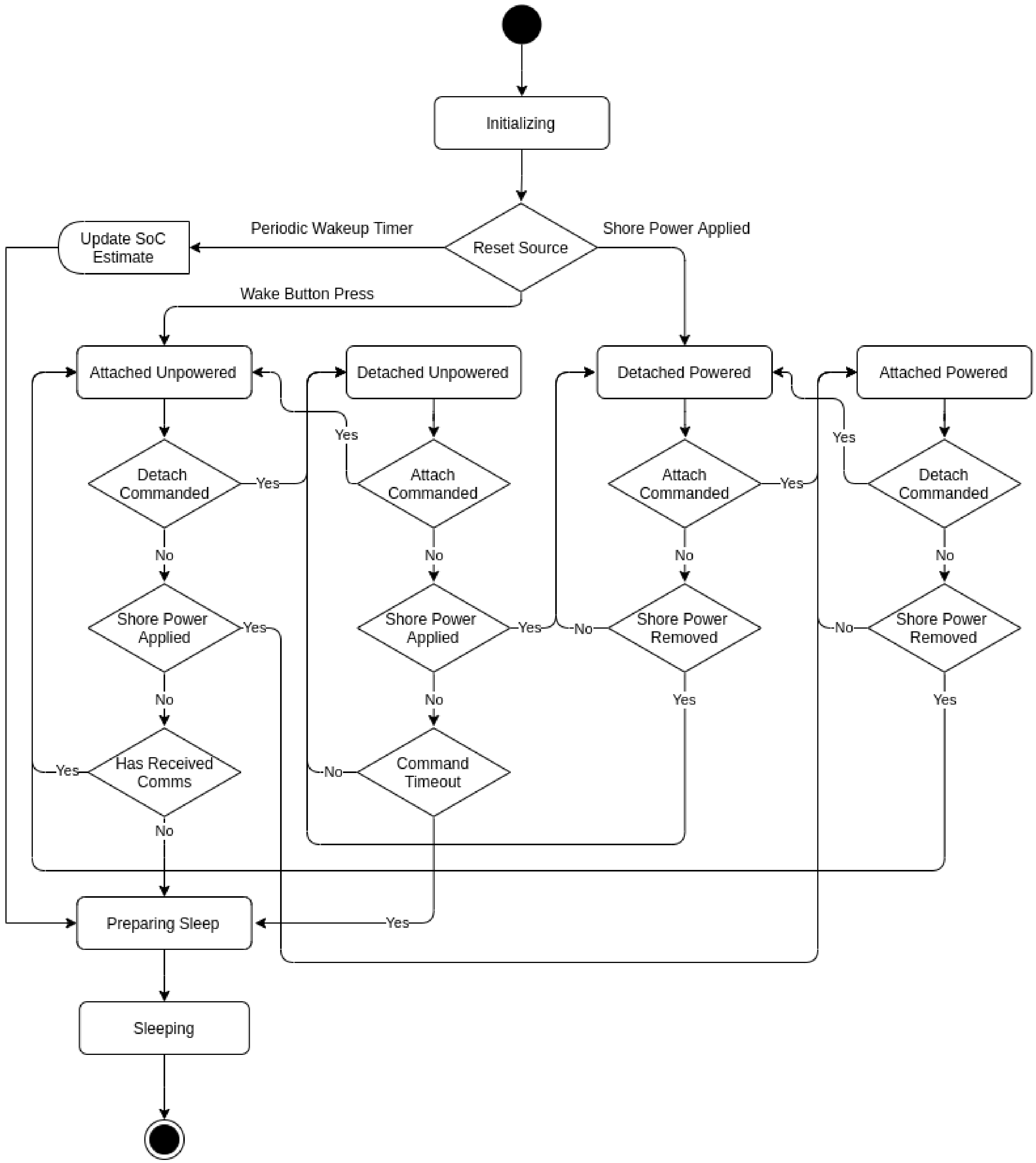
External Indicator

The Lohner has an externally visible, tri-color LED. The LED flashes at a slow rate when sleeping, and at a faster rate when awake. The following table describes the indicator patterns resulting from the various system states.

🟢										Sleep mode, Battery > 40%
🔴										Sleep mode, Battery <= 40%
										Battery < 20% (totally off)
🟢		🔵	🔵		🟢		🔵	🔵		Awake / Active, Battery > 40%
🔴		🔵	🔵		🔴		🔵	🔵		Awake / Active, Battery <= 40%

Application State Machine

The Lohner ships with software that operates per the state diagram shown below:



Software Update / Recovery

The Lohner software may be updated over the CAN or RS232 serial interfaces, using the allocortech bootloader tool, yubnub, which is available as part of the allocortech SDK.

If the CPU becomes software-locked where the allocortech bootloader is not responsive, or if the allocortech bootloader is not desired, the silicon-level STMicro bootloader may be activated by the following sequence:

1) Connect the “MCU Recovery” pin to GND on the J1 connector
2) Closing the circuit between the two “monitored contact closure” pins on the J1 connector (eg pressing the “wake” push button)

At this point the STM32 processor in the Lohner is reset into its internal bootrom. No software functions are operative until new software is loaded into the processor, or the battery is removed. The Sleep Mode Battery Protection is not guaranteed to operate correctly if the MCU is left in the STMicro bootrom mode.

Communications Protocol

The Lohner communicates using the allocortech **Y**et **A**nother **P**acket **P**rotocol.

Yet Another Packet Protocol (basics)

YAPP (**Y**et **A**nother **P**acket **P**rotocol) is a serial protocol developed by allocortech inc. for use in communication. The protocol is similar to many packet formatting protocols, and includes a header, packet payload and a CRC for data integrity checks. This document will not go through all of the details of the protocol, but a basic message is constructed as follows:

Header						Payload	CRC
SYNC[2]	Seq[1]	CTL[1]	ID[4]	Size[2]	RSVD[2]	*Size* Bytes	32-bit CRC

This same protocol is utilized over the CAN bus, with some limitations:

- ID field is limited to 11-bits
- Payloads over 8-bytes (or 64-bytes in the case of CAN FD) are segmented into multi-frame messages, which use some of the extended CAN ID space for control and payload re-construction

The payload can be constructed of custom encoded types, but are typically YAP encoded types:

- YapRaw (e.g. int/uint(8/16/32/64), float, double, bool
- YapFloatCompressed - float/double encoded on wire as uint8/16/32 with specified min/max as well as specific reserved values for nan/inf/overflow.

Parsers/handlers for CAN and a byte-stream as well as serialization/deserialization for the payload types are available in the allocore SDK, but if more detail is desired regarding the CAN protocol or explicit decoding of binary data, allocortech can provide more detail on request.

YAPP Messages

There are two message types that are sent from the attached system into the Lohner, and two message types that are transmitted from the Lohner unit that are intended to be consumed by the attached system. The unique identifiers for these messages are customizable, but the defaults will be listed below in the message definitions. Further, the detailed message definitions can be found in lohner_yap_msgs.h source file, which can be provided, or is included if you have purchased a license to the allocore SDK with your unit.

LohnerHelloMsg

The Lohner “Hello” message is a command sent from the main system to the Lohner, which has an empty payload, and is something of a “ping” type message which lets the Lohner know that a valid host is attached to the port, such that the system will start sending telemetry messages on this port, without changing the state of the unit. Connection can also be established with the LohnerCommandMsg, but will change Lohner operating state, which may not be desired simply to establish a connection (e.g. to look at data from a running system).

Yap ID: 0x04
Payload Size: 0-bytes

LohnerCommandMsg

The Lohner Command message is the main interface in which the system controls the Lohner unit. This includes *connecting/attaching* the battery to the LV bus, enabling/disabling the ability for the Lohner to charge the internal battery, and setting the max charging current. Further, there is a mechanism for clearing system faults with the command.

Yap ID: 0x0C
Payload Size: 5-bytes

Field	Type	Wire-Size	Range	Description
disconnect	Bool	uint8_t (1)	N/A	Detach battery from bus
charge_enable	Bool	uint8_t (1)	N/A	Allow charging of battery
clear_faults	Bool	uint8_t (1)	N/A	Clear system faults
max_charge_current	float	uint16_t(2)	0A-3A	Max charge current value

LohnerBatteryDataMsg

The Battery Data Message is likely the primary message of concern in the system, as it reports the state of the battery and system control, including energy available, voltages, temperatures, summarized currents, etc.

Yap ID: 0x0B
Payload Size: 38-bytes
Frequency: 10Hz

Field	Type	Wire-Size	Range	Description
Timestamp	U64	uint64_t(8)	N/A	Time since boot (nsec)
pack_voltage	float	uint16_t(2)	0V-50V	Battery Voltage
pack_current	float	uint16_t(2)	-10A-200A	Battery Current
est_rem_ah	float	uint16_t(2)	0Ah-100Ah	Battery State of Charge
bus_v	float	uint16_t(2)	0V-60V	Output Bus Voltage
soc_fract	float	uint8_t (1)	0-1.0	State of Charge fract
estimate[1]	custom	uint8_t (1)	Custom	Estimation State
attached	bool	uint8_t (1)	N/A	Is battery attached
charging	bool	uint8_t (1)	N/A	Is battery charging
cell0_voltage	float	uint16_t(2)	0V-5V	Cell 0 Voltage
cell1_voltage	float	uint16_t(2)	0V-5V	Cell 1 Voltage
cell2_voltage	float	uint16_t(2)	0V-5V	Cell 2 Voltage
cell3_voltage	float	uint16_t(2)	0V-5V	Cell 3 Voltage
cell4_voltage	float	uint16_t(2)	0V-5V	Cell 4 Voltage
cell5_voltage	float	uint16_t(2)	0V-5V	Cell 5 Voltage
cell0_temp	float	uint8_t (1)	-55°C-125°C	Cell 0 Temperature
cell1_temp	float	uint8_t (1)	-55°C-125°C	Cell 1 Temperature
cell2_temp	float	uint8_t (1)	-55°C-125°C	Cell 2 Temperature
cell3_temp	float	uint8_t (1)	-55°C-125°C	Cell 3 Temperature
cell4_temp	float	uint8_t (1)	-55°C-125°C	Cell 4 Temperature
cell5_temp	float	uint8_t (1)	-55°C-125°C	Cell 5 Temperature

[1]The estimate value is a custom enumeration which reports the estimation state of the battery. The supported valid values are the following:

- 0: Invalid Estimate
- 1: Estimate is valid, using OCV method
- 2: Estimate is valid, Coulomb Counting method

When the system is in the OCV state of estimation, it is considered most accurate since the accumulation of current sensor error has been removed.

LohnerTelemMsg

The Lohner Telem message is a detailed telemetry message that by default is sent from the device, but may be disabled if bus traffic is a concern. This message contains significant details of the internal operating conditions of the platform, including individual current sensors, temperature sensors, voltages at different points in the bus conditioning circuitry, etc. These values are not expected to be of great value to the system, except in troubleshooting issues encountered with the Lohner itself.

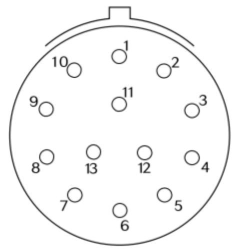
Yap ID: `0x07`
Payload Size: `76-bytes`
Frequency: `1Hz`

See Appendix B for details of this telemetry message.

Electrical Interface

Connector Pinouts

J1 - Communications Connector



Face view of receptacle D38999/24FB35PN.

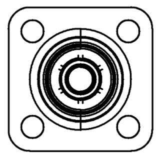
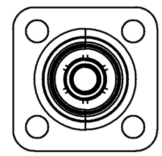
13x size 22D male pins, 3A each.

Recommended mate: D38999/26FB35SN

The communications connector has the following pin allocation:

Pin	Signal	Function
11	GND	Ground reference for communications
3	ENABLE (+)	Monitored Contact Closure (“Wake” push button)
4	ENABLE (-)	
10	D0 (+)	CAN FD or Half-Duplex RS485, MCU selectable at run-time
1	D0 (-)	
7	D1 (+)	CAN FD or Half-Duplex RS485, MCU selectable at run-time
8	D1 (-)	
12	GND	Ground for MCU Recovery
5	UART TX	RS232 TX (bootloader, debug)
9	UART RX	RS232 RX (bootloader, debug)
2	MCU Recovery	Bootloader Enable Pin
13	CHG B(-)	Secondary Charge Source
6	CHG B(+)	

J2/J3 - Primary Power Connection

Power (+)	Power (-)
	
5.7mm SurLok SLPRATPSR (Red)	5.7mm SurLok SLPRATPSB (Black)
Mate: SLPPAxxBSR	Mate: SLPPAxxBSB

The Primary Power Connections (+ and -) provide the main power output from the battery (50A sustained, 100A peak for 30 seconds) as well as being the primary source for charging the battery when external bus power is present.

JTAG Interface

The MCU has a 14-pin ST-Link v3 JTAG 14-pin debugger interface accessible inside the enclosure.

Pin	Name	Pin	Name
1	N/C	2	N/C
3	+3.3V	4	JTMS
5	GND	6	JTCK
7	GND	8	JTDO
9	N/C	10	JTDI
11	GND	12	nRST
13	3.3V TTL USART3 RX	14	3.3V TTL USART3 TX

Chassis Bonding

The chassis (mounting feet) are treated electrically as chassis. The Lohner (-) power terminal and all GND signals on the J1 connector are considered ground. Chassis and Ground are internally connected via 2,200pF Y-capacitor in parallel with a 4.7MΩ bleed resistor.

Electrical Ratings

Parameter	Value	Units
Internal Battery Battery Power Rating *See Appendix A for cell specifications*	6S2P LiPo 8,000	 mAh
Input Voltage Min Max Max for Charging	 22.1 36.0 28.0	 V V V
Output Current Sustained Maximum for 30 seconds	 50 100	 A A
Charge Current (programmable) Min Max	 0.0 (disabled) 3.0	 A A
Measurement Error * Output Current < 1A Output Current 1A to 100A Battery Voltage All other voltages	 < 4% < 1% <= 0.25% <= 2%	 Typical at 25°C

* Measurement errors represent designed-in and measured prototype performance, but are not characterized over temperature and are not 100% guaranteed.

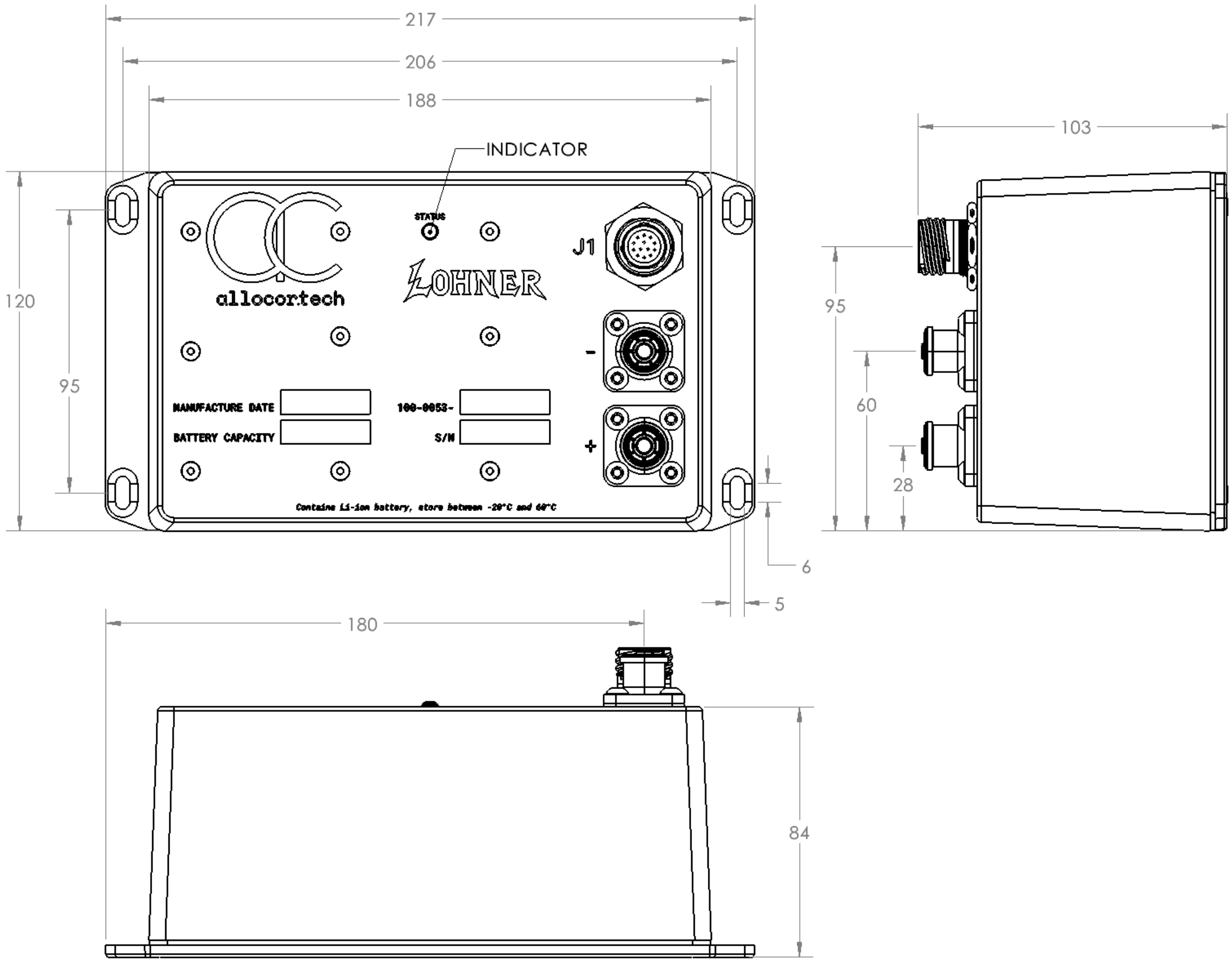
Environmental Ratings

Parameter	Value	Units
Storage Temperature Min Max	 -20 60	 °C °C
Operating Temperature * Min Max (at ±0A current)	 -20 60	 °C °C
Ingress Protection Designed to *(with all connectors mated)*	 IP65	

* The rated operating temperature is dictated by the limitations of the internal lithium polymer battery. Self heating of the battery and circuit board will reduce the effective safe maximum operating temperature as a time dependent function of the expected current draw. Lohner uses an on-PCB thermistor as an approximation for battery temperature, although as a manufacturing option a thermistor can also be placed directly on the battery. Charging of the battery will happen under JEITA temperature-based charging thresholds/limitations, details of which can be found in the TI bq24616 datasheet.

Mechanical Interface

The mechanical outline of the Lohner is given below.



Measurements given in mm

Attachment: 4mm or #8 screws on 95mm x 206mm square pattern
Materials: Aluminum Alloy
Finish, Base: cast aluminum (chromate conversion on request)
Finish, Enclosure: cast aluminum (chromate conversion on request),
Powder Coat, Blue Purple (Flip Flop), Powder Buy the Pound SK19811
Weight: 2,200 g

APPENDIX A - LiPo Cell Specification

The cell specifications are listed below for the 8Ah 6S2P battery used in the Lohner.

TECHNICAL DATA

CELL DIMENSIONS - 137 x 45 x 7.25mm

CELL WEIGHT - 93g

CELL CAPACITY @ 0.5C RATE - 4Ah (4,000mAh)

VOLTAGE - 3.7V (NOMINAL/STORAGE) ; 3.0V (MIN) ; 4.2V (MAX)

INTERNAL IMPEDANCE (1KHZ AC TYPICAL) - 2mΩ

POWER DENSITY @ Continuous Rating - 9,548 W/kg

STANDARD CHARGE RATE - 4A (1C MAX)

FAST CHARGE RATE - 20A (5C MAX)

MAXIMUM CONTINUOUS DISCHARGE - 240A (60C)

MAXIMUM PEAK DISCHARGE - 600A (150C)

CYCLE LIFE - Varies per application/usage/care

OPERATING TEMPERATURE - 0°C to 60°C

STORAGE TEMPERATURE - 0°C to 23°Cv

APPENDIX B - LohnerTelemMsg Detail

Field	Type	Wire-Size	Range	Description
Timestamp	U64	uint64_t(8)	N/A	Time since boot (nsec)
vbatt1_volts	float	uint16_t(2)	0V-33V	Batt1 (SW) Voltage Value
vbatt2_volts	float	uint16_t(2)	0V-33V	Batt2 (SW) Voltage Value
vbatt3_volts	float	uint16_t(2)	0V-33V	Batt3 (SW) Voltage Value
batt_volts	float	uint16_t(2)	0V-26.4V	V_{BATT} output voltage
vbus_a_volts	float	uint16_t(2)	0V-33V	V_{BUSA} pre-fuse voltage
Vbus_a_fused_volts	float	uint16_t(2)	0V-33V	V_{BUSA} fused voltage
vbus_b_volts	float	uint16_t(2)	0V-33V	V_{BUSB} pre-fuse voltage
vbus_b_fused_volts	float	uint16_t(2)	0V-33V	V_{BUSB} fused voltage
pos_va_iso_volts	float	uint16_t(2)	0V-75V	V+ Iso Supply Voltage
neg_va_iso_volts	float	uint16_t(2)	0V-75V	V- Iso Supply Voltage
pos_12v0_iso_volts	float	uint16_t(2)	0V-75V	12V Iso Supply Voltage
vbus_clamped_volts	float	uint16_t(2)	0V-33V	Internal clamped voltage
lv_ps_supply_volts	float	uint16_t(2)	0V-33V	Logic supply voltage
pos_5v0_volts	float	uint16_t(2)	0-6V	5V Rail voltage
pos_3v3_volts	float	uint16_t(2)	0V-6V	3.3V Rail voltage
vbus_prot_volts	float	uint16_t(2)	0V-33V	Protected V_{BUS} voltage
vbus_or_volts	float	uint16_t(2)	0V-33V	Internal OR'd supply volt
lv_ps_supply_amps	float	uint16_t(2)	0A-0.2A	Logic supply current
current_high_1	float	uint16_t(2)	0A-24.2A	I_{High1} measured current
current_high_2	float	uint16_t(2)	0A-24.2A	I_{High2} measured current
current_high_3	float	uint16_t(2)	0A-24.2A	I_{High3} measured current
current_high_4	float	uint16_t(2)	0A-24.2A	I_{High4} measured current
current_high_5	float	uint16_t(2)	0A-24.2A	I_{High5} measured current
current_low_amps	float	uint16_t(2)	0A-4.84A	High res current sense
batt_charge_amps	float	uint16_t(2)	0A-12A	Battery Charge Current
dac2_out1_volts	float	uint16_t(2)	0V-3V	DAC output volts
vsense	float	uint16_t(2)	-55°C-125°C	CPU Temperature
vrefint	float	uint16_t(2)	0V-3V	CPU Reference Voltage
*heater_temp_c	float	uint16_t(2)	-55°-125°C	Heater Temperature
pcb_temp_c	float	uint16_t(2)	-55°-125°C	PCB Temperature
*battery_temp_c	float	uint16_t(2)	-55°-125°C	Battery Temperature
cpu_usage_pct	float	uint8_t (1)	0%-100%	CPU Usage in percent
stack_usage_pct	float	uint8_t (1)	0%-100%	Lohner app stack usage
heap_usage_pct	float	uint8_t (1)	0%-100%	Lohner app heap usage
state	rsvd	uint8_t (1)	reserved	Internal state of app
max_charge_current	float	uint16_t(2)	0A-5A	Max charge current

** Values reported are likely invalid as these features are optional in the Lohner product*