Document Number:  ENG2021002

# InSight CANbus Specification

| | |
|---|---|
| Author | Anna |
| Revision | H |
| Date | 9/27/2021 |

## Table of Contents

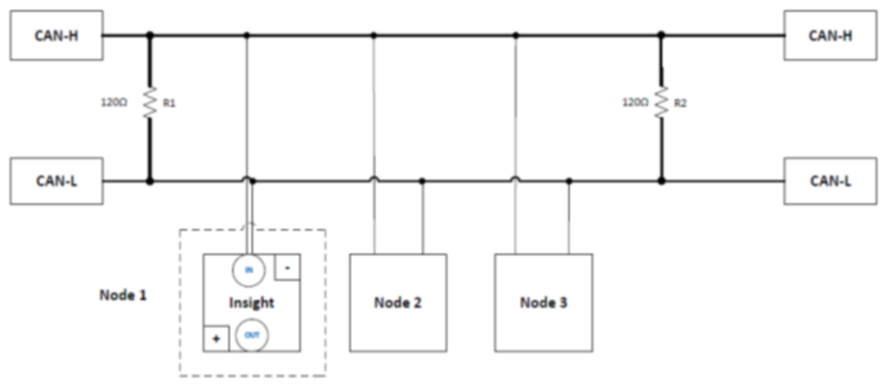



powerpros@relionbattery.com 855.931.2466 relionbattery.com

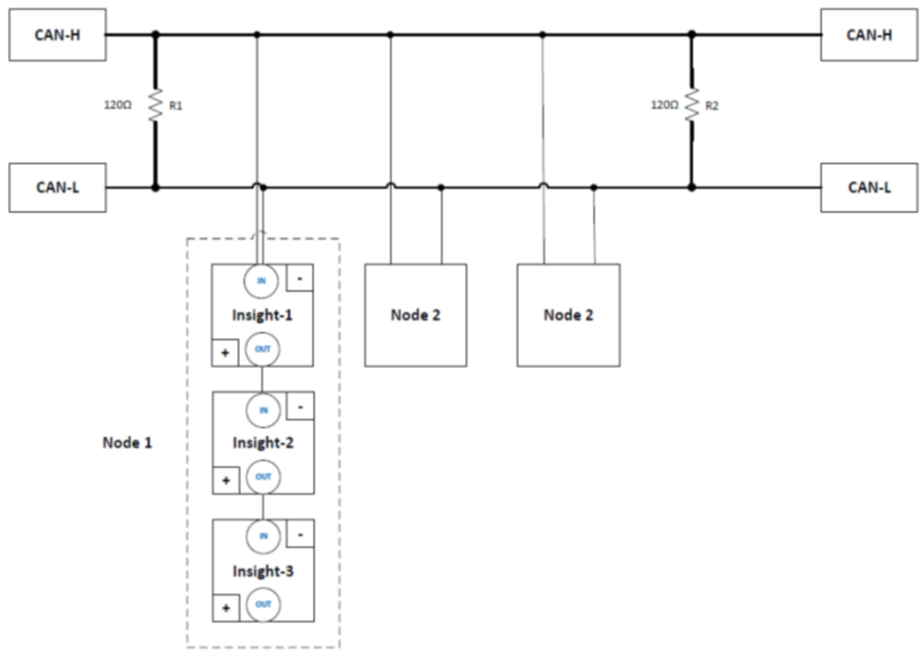
## 1. PURPOSE

This document gives the details of the hardware, and CANbus details. It also contains instructions on how to query the batteries via CAN.

### 1.1 CANbus Connections – Single Battery:




### 1.2 CANbus Connections – Multiple Batteries:



## 2. SCOPE

RELiON InSight Series has built in CAN communication. This is the CAN protocol document for 48V-030-GC2, 24V-030-GC2 and 24V-030-GC2

## 3. RESPONSIBILITIES

This document is for engineering and technical support use only, it is not intended for customer use.

## 4. PREREQUISITES

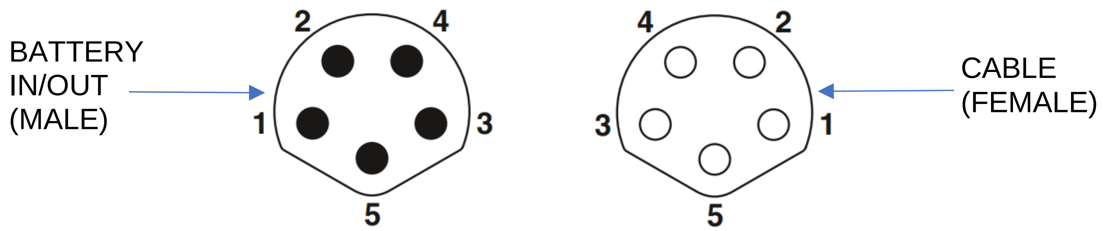
### 4.1 Hardware:

120Ω termination resistors needs to be present on the CAN network for proper operation. The termination resistors must be provided by the system and not built into batteries. Ensure that the cables and connectors have a characteristic impedance of 120Ω. The RELiON insight series fuel gauge (Part Number BI-3140R) has the 120 Ohm termination resistor.

### 4.2 CANbus Details

- CAN2.0A
- Identifier: 11-bit
- Pack Broadcast ID: 0x5FF
- Bit Rate: 250kbps
- Multi-Byte value form: Little-Endian

**48V-030-GC2 - Battery Connectors (M8-P5):**
The Input connector is located on the negative side of the battery, and the output connector is located on the positive side of the battery.




| Pin Number | CAN Input Connector (on battery negative) | CAN Output Connector (on battery positive) | CANbus Cable Wires (2-foot Cable) |
|---|---|---|---|
| **1** | Reserve | Reserve | Green |
| **2** | CANbus Address IN | CANbus Address IN | Blue |
| **3** | CAN-L | CAN-L | Black |
| **4** | CAN-H | CAN-H | Yellow |
| **5** | P- (Battery Negative) | P- (Battery Negative) | Red |

powerpros@relionbattery.com 855.931.2466 relionbattery.com

## 5. MESSAGE LAYOUT

**Message Format:**

| ID0 - ID10 | DLC | DATA0 | DATA1 | DATA2 | DATA3 | DATA4 to DATAn | DATAm |
|---|---|---|---|---|---|---|---|
| Message ID | Data Length Code | Command ID | Continue sequential number | Command Type | Source Address | CANbus data | Checksum |

**▬ ID0-ID10:**

Broadcast ID: 0x5FF (automatically broadcasted)
Message ID: 0x500 (data returned from BMS when requested)

**▬ DLC, Data Length Code:**

Data length 0 bytes to 8 bytes.

**▬ DATA0:**

Command ID for available data. For example, 0x09 for pack voltage.

**▬ DATA1:**

For data that is less than 1 byte = 0x81
For data that is greater than 8 bytes:

- The first set of data is assigned 0x01
- The second set of data is assigned 0x02
- For the last data set, DATA1 will be assigned 0x8(bytes)
  - For example: if there are three data sets, the last data set will be assigned 0x83

**▬ DATA2:**

| Command Type | Value |
|---|---|
| CAN word read | 0x01 |
| CAN block read | 0x02 |
| CAN communication failure | 0x0B |
| CAN bus broadcast | 0x21 |
| Checksum error | 0x0C |

**Data3:**
Source address of the broadcast ID or message ID. The address is automatically assigned. For example, battery one is assigned an address of 0x01, battery two is assigned 0x02, and so on until 0xFF is reached.

Broadcasted Frame: single battery

| Battery No. | ID | DLC | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
|---|---|---|---|---|---|---|---|---|---|---|
| One | 0x5FF | 8 | 0x20 | 0x81 | 0x21 | **0X01** | Alarm1 | Alarm2 | Status | SOC |

Broadcasted Frame: multiple batteries

| Battery No. | ID | DLC | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
|---|---|---|---|---|---|---|---|---|---|---|
| One | 0x5FF | 8 | 0x20 | 0x81 | 0x21 | **0X01** | Alarm1 | Alarm2 | Status | SOC |
| Two | 0x5FF | 8 | 0x20 | 0x81 | 0x21 | **0X02** | Alarm1 | Alarm2 | Status | SOC |
| Three | 0x5FF | 8 | 0x20 | 0x81 | 0x21 | **0X03** | Alarm1 | Alarm2 | Status | SOC |

Request Message Frame: For battery voltage (0x09) as an example for a single battery

| Battery No. | Type | ID | DLC | D0 | D1 | D2 | D3 | D4 | D5 | D6 |
|---|---|---|---|---|---|---|---|---|---|---|
| One | Tx | 0x001 | 5 | 0x09 | 0x81 | 0x01 | 0x00 | 0x00 | -- | -- |
| One | Rx | 0x005 | 7 | 0x09 | 0x81 | 0x01 | **0X01** | Voltage | | checksum |

Request Message Frame: For battery voltage (0x09) as an example for multiple batteries

| Battery No. | Type | ID | DLC | D0 | D1 | D2 | D3 | D4 | D5 | D6 |
|---|---|---|---|---|---|---|---|---|---|---|
| **One** | Tx | 0x001 | 5 | 0x09 | 0x81 | 0x01 | 0x00 | 0x00 | -- | -- |
| **Two** | Tx | 0x002 | 5 | 0x09 | 0x81 | 0x02 | 0x00 | 0x00 | -- | -- |
| **Three** | Tx | 0x003 | 5 | 0x09 | 0x81 | 0x03 | 0x00 | 0x00 | -- | -- |
| **One** | Rx | 0x005 | 7 | 0x09 | 0x81 | 0x01 | **0X01** | Battery 1 Voltage | | checksum |
| **Two** | Rx | 0x005 | 7 | 0x09 | 0x81 | 0x01 | **0X02** | Battery 2 Voltage | | checksum |
| **Three** | Rx | 0x005 | 7 | 0x09 | 0x81 | 0x01 | **0X03** | Battery 3 Voltage | | checksum |

**DATA4 to DATAn:**
Field data

**DATAm:**
A Checksum Exclusive OR Operation of DATA3 to DATAn fields (See appendix C)

## 6. BROADCAST DATA

### 6.1 Broadcasted Data Frames:

The BMS broadcasts five data frames that communicate relevant information for applications every three seconds. The sixth frame, 0x7F8, is only used for internal purposes. **All multi byte data is read in little endian ex. pack voltage would be read 0xD3D2**

| ID | DLC | D0* | D1** | D2 | D3 | D4 | D5 | D6 | D7 |
|---|---|---|---|---|---|---|---|---|---|
| 0x5FF | 8 | 0x20 | 0x81 | 0x21 | 0X01 | Alarm1 | Alarm2 | Status | SOC |
| 0x5FF | 8 | 0x30 | 0X01 | SOC | SOH | Max Disch. Current ((1,500)/10) = 150A | | Max Regen Current ((1,000)/10)=100A | |
| 0x5FF | 8 | 0x31 | 0X01 | Pack Voltage | | Cycle Count | | Not Used | Not Used |
| 0x5FF | 8 | 0x32 | 0X01 | Temperature 1 | | Temperature 2 | | Temperature 3 | |
| 0x5FF | 8 | 0x33*** | 0X01 | Low byte current 1 | Low byte current 2 | High byte current 1 | High byte current 2 | Not Used | Not Used |
| 0x7F8 | 7 | This field is for inter-battery communication only. | | | | | | | |

* For field 0x20 (D0), D3 refers to the battery number.

**For fields 0x30 to 0x33, D1 refers to the battery number.

***See appendix B for Current Readings details.

### 6.2 Bit Definition for 0x20:

| | | | |
|---|---|---|---|
| **Alarm1** | Bit7 | 0x80 | Pack low-voltage alarm |
| | Bit6 | - | Reserved |
| | Bit5 | 0x20 | Over-temperature protection during discharge |
| | Bit4 | 0x10 | Overload protection |
| | Bit3 | - | Reserved |
| | Bit2 | - | Reserved |
| | Bit1 | 0x02 | Cell low voltage alarm |
| | Bit0 | - | Reserved |
| **Alarm2** | Bit7 | - | Not Used |
| | Bit6 | 0x40 | Over-current alarm during discharge |
| | Bit5 | 0x20 | Pack under-voltage protection |
| | Bit4 | 0x10 | Cell under-voltage protection |
| | Bit3 | - | Not Used |
| | Bit2 | - | Reserved |
| | Bit1 | 0x02 | Over-temperature alarm |
| | Bit0 | - | Not Used |
| **Status** | Bit7 | - | Not Used |
| | Bit6 | 0x40 | Fully charged status (1 = during charge & SOC = 100%) |
| | Bit5 | 0x20 | Heating-element ON |
| | Bit4 | - | Not Used |
| | Bit3 | - | Not Used |
| | Bit2 | 0x04 | Discharge Current Detected |
| | Bit1 | 0x02 | Charging Current Detected |
| | Bit0 | - | Not Used |

# 7. REQUESTING MESSAGES

## 7.1 Message Construction:

Requesting data from the BMS. The Message ID of battery one is 0x001 and the Message ID of battery two is 0x002, and so on until 0x5FF is reached.
Follow the message layout for D1 and D2.

**Example Pack voltage of single battery:**

| ID | DLC | D0 | D1 | D2 | D3 | D4 |
|---|---|---|---|---|---|---|
| 0x001 | 5 | 0x09 | 0x81 | 0x01 | 0X00 | 0X00 |

**Example Pack voltage of multiple batteries:**

| Battery No. | ID | DLC | D0 | D1 | D2 | D3 | D4 |
|---|---|---|---|---|---|---|---|
| **One** | 0x001 | 5 | 0x09 | 0x81 | 0x01 | 0X00 | 0X00 |
| **Two** | 0x002 | 5 | 0x09 | 0x81 | 0x02 | 0X00 | 0X00 |

## 7.2 Frequently requested messages: (see appendix A for all messages)

| CAN Messages and Responses | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Message Description** | **Type** | **ID** | **DLC** | **D0** | **D1** | **D2** | **D3** | **D4** | **D5** | **D6 (CRC)** |
| **Pack Voltage** | Tx | 0x001 | 5 | 09 | 81 | 01 | 00 | 00 | | |
| | Rx | 0x500 | 7 | 09 | 81 | 01 | 01 | 37 | CD | FB |
| **Cell Temperature** | Tx | 0x001 | 5 | 08 | 81 | 01 | 00 | 00 | | |
| | Rx | 0x500 | 7 | 08 | 81 | 01 | 01 | A1 | 0B | AB |
| **Relative State of Charge** | Tx | 0x001 | 5 | 0D | 81 | 01 | 00 | 00 | | |
| | Rx | 0x500 | 7 | 0D | 81 | 01 | 01 | 46 | 00 | 47 |
| **Charging Current** | Tx | 0x001 | 5 | 14 | 81 | 01 | 00 | 00 | | |
| | Rx | 0x500 | 7 | 14 | 81 | 01 | 01 | 10 | 27 | 36 |
| **Cycle Count** | Tx | 0x001 | 5 | 17 | 81 | 01 | 00 | 00 | | |
| | Rx | 0x500 | 7 | 17 | 81 | 01 | 01 | 08 | 01 | 09 |

| Cell Voltages | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Message Description** | **Transmission Type** | **ID** | **DLC** | **D0** | **D1** | **D2** | **D3** | **D4** | **D5** | **D6** | **D7** |
| **Serial Cell Voltages** | Tx | 0x001 | 5 | 28 | 81 | 02 | 00 | 00 | | | |
| | Rx | 0x500 | 8 | 28 | 1 | 2 | 1 | 20* | 8 | 0D | 5 |
| | Rx | 0x500 | 8 | 28 | 2 | 2 | 0D | 4 | 0D | E8 | 0C |
| | Rx | 0x500 | 8 | 28 | 3 | 2 | 5 | 0D | 6 | 0D | 9 |
| | Rx | 0x500 | 8 | 28 | 4 | 2 | 0D | 4 | 0D | 4 | 0D |
| | Rx | 0x500 | 8 | 28 | 5 | 2 | 6 | 0D | 6 | 0D | 17 |
| | Rx | 0x500 | 8 | 28 | 6 | 2 | 0D | 2 | 0D | 3 | 0D |
| | Rx | 0x500 | 8 | 28 | 87 | 2 | 2 | 0D | 7 | 0D | D8** |

*2 times Number of cells in hex.
**check sum
Boxed in area is all 16 cell voltages read from left to right in sets of two, reordered in little endian.
Ex: Cell 1 0D08, Cell 2 0D05, …. Cell 16 0D07

### 7.3 CANbus Reset:

The CANbus reset command is a global command that is received by all the batteries and is sent to ID 0x7F8. Only the CANbus is reset during the reset sequence. The battery continues to operate as normal. The reset cycle time is 3.15 seconds. The table below shows how the command should be constructed.

| ID | DLC | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
|---|---|---|---|---|---|---|---|---|---|
| 0x7F8 | 6 | 10 | 81 | 21 | 22 | 55 | 77 | | |

### 7.4 Sleep Timer:

The sleep timer function has a range of 20 to 160 hours. The command sequence requires three CAN IDs to be accessed (0x50, 0xD0, and 0x52). The sequence for a 48 hour sleep timer is shown below. D4 in frame DO (highlighted in green), sets the time. D6 in frame DO is the CRC of bytes D3 and D4. This should be done one battery at a time.

| ID | DLC | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
|---|---|---|---|---|---|---|---|---|---|
| 0x001 | 7 | 50 | 81 | 11 | 22 | 00 | 00 | 22 | |
| 0x001 | 7 | D0 | 81 | 11 | 22 | 30 | 00 | 12 | |
| 0x001 | 7 | 52 | 81 | 11 | 22 | 00 | 00 | 22 | |

To verify timer use.

| ID | DLC | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
|---|---|---|---|---|---|---|---|---|---|
| 0x001 | 5 | D0 | 81 | 01 | 22 | 22 | | | |

## 8. DEFINITIONS

- State of Charge (SOC)
- State of Health (SOH)
- Over Voltage Protection (OVP)
- Under Voltage Protection (UVP)
- Data Length Code (DLC)

## 9. REVISION HISTORY

| Rev | Change Summary | Effective Date | Author | Reviewer/Approved |
|---|---|---|---|---|
| **G** | Added software fields for new SW | 03/05/2021 | Ronald P. | |
| **H** | Update for software 2A-06 and other improvements | 09/27/2021 | Anna Kirchan | Pradeep Jaltota |

# APPENDIX

## A. MESSAGE DEFINITIONS

| CANbus Read Messages | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Pack ID** | **Param ID** | **Parameter Description** | **Size** | **DLC** | **Data Type** | **Range** | **Unit** |
| 0x001 | 0x06 (High) | Pack Voltage (high byte) | Word | 5 | Unsigned Int | 65536-131071 | mV |
| | 0x09 (Low) | Pack Voltage (low byte) | | | | 0-65535 | |
| 0x001 | 0x08 | Battery Temperature | Word | 5 | Unsigned Int | -40C - 120C (value - 2731)/10= tempC | C |
| 0x001 | 0x0D | Relative SOC (Remaining Capacity / Full Charge Capacity) | Word | 5 | Unsigned Int | 0-100 | % |
| 0x001 | 0x0E | Absolute SOC (Remaining Capacity / Design Capacity) | Word | 5 | Unsigned Int | 0-100 | % |
| 0x001 | 0x1E (High) | Remaining Capacity (returns the predicted remaining capacity) | Word | 5 | Unsigned Int | 65536-131071 | mAh |
| | 0x0F (Low) | | | | | 0-65535 | |
| 0x001 | 0x1F (High) | Full Charge Capacity (returns the predicted remaining capacity value when it is fully charged) | Word | 5 | Unsigned Int | 65536-131071 | mAh |
| | 0x10 (Low) | | | | | 0-65535 | |
| 0x001 | 0x14 | Desired Charging Current | Word | 5 | Unsigned Int | 0-65535 | mA |
| 0x001 | 0x15 | Desired Charging Voltage | Word | 5 | Unsigned Int | 0-65535 | mV |
| 0x001 | 0x17 | Cycle Count | Word | 5 | Unsigned Int | N/A | N/A |
| 0x001 | 0x1D (High) | Design Capacity (returns the capacity value (mAh) of the batery pack | Word | 5 | Unsigned Int | 65536-131071 | mAh |
| | 0x18 (Low) | | | | | 0-65535 | |

| CANbus Read Messages - Continued | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Pack ID** | **Param ID** | **Parameter Description** | | **Size** | **DLC** | **Data Type** | **Range** | **Unit** |
| 0x001 | 0x19 | Design Voltage | | Word | 5 | Unsigned Int | 0-65535 | mV |
| 0x001 | 0x28 | Cell Voltages | | Block | 5 | Unsigned Int | 0-65535 | mV |
| | | Byte 0 | (No. of cells) x 2 | | | | | |
| | | byte2:byte1 | Vcell1 | | | | | |
| | | byte4:byte3 | Vcell2 | | | | | |
| | | byte6:byte5 | Vcell3 | | | | | |
| | | … | … | | | | | |
| | | byte32-byte31 | Vcell16 | | | | | |
| 0x001 | 0x29 | Temperatures | | Block | 5 | Unsigned Int | -40C - 120C (value - 2731)/10= tempC | C |
| | | Byte 0 | (No. of cells) x 2 | | | | | |
| | | byte2:byte1 | Temp. 1 | | | | | |
| | | byte4:byte3 | Temp. 2 | | | | | |
| | | byte6:byte5 | Temp. 3 | | | | | |
| 0x001 | 0x2A* | Battery Current | | Block | 5 | Unsigned Int | 0-65535 | mA |
| | | Byte 0 | (No. of cells) x 2 | | | | | |
| | | byte2:byte1 | Low Current | | | | | |
| | | byte4:byte3 | High Current | | | | | |
| 0x001 | 0x23 | Software Version 1 | | Block | 5 | Unsigned Int | N/A | N/A |
| | | Byte 0 (length) | 0x11 or 0x12 | | | | | |
| | | Byte 1 | MainPack Major | | | | | |
| | | Byte 16 | MainPack Minor | | | | | |
| | | Byte 17 | (106C-CANbus) Major | | | | | |
| 0x001 | 0xEE | Software Version 2 | | Block | 5 | Unsigned Int | N/A | N/A |
| | | Byte 0 (length) | 0x00 | | | | | |
| | | Byte 3 | (106C-CANbus) Minor | | | | | |

**Note about Parameter Description:** Bytes listed here are representative of the byte number the whole block of data, not location in one word in a data block.

See example on next page.

Ex: Below is some of the cell voltage Rx message with the data bytes numbers according to how they are referred to in the parameter description

| Transmission Type | ID | DLC | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
|---|---|---|---|---|---|---|---|---|---|---|
| **Rx** | 0x500 | 8 | 28 | 1 | 2 | 1 | Byte 0 | Byte 1 | Byte 2 | Byte 3 |
| **Rx** | 0x500 | 8 | 28 | . | . | . | . | . | . | . |
| **Rx** | 0x500 | 8 | 28 | 87 | 2 | Byte 29 | Byte 30 | Byte 31 | Byte 32 | D8 |

## B. CURRENT READING

Current is broadcasted as 32-bit signed integer, the most significant bit (MSB) is the sign bit .

Current = Signed Int32(D5D4D3D2)

**Request Message Current Example:**

| ID | DLC | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
|---|---|---|---|---|---|---|---|---|---|
| 0x001 | 5 | 0x2A | 0x81 | 0x02 | 0x00 | 0x00 | | | |
| 0x500 | 8 | 0x2A | 0x01 | 0x02 | 0x01 | 0x04 | 0xB2 | 0x08 | 0x00 |
| 0x500 | 5 | 0x2A | 0x82 | 0x02 | 0x00 | 0xBF | | | |

Current =Signed Int32(0x00 00 08 B2) = 2226 mA

1. Read bytes 5 and 6 in little endian => 08B2
2. Read byte 7 from first signal and byte 3 from second in little endian = >0000
3. Put together => 0000 08B2
4. Convert to decimal => 2226 mA

**Broadcasted Message Current Example:**

| ID | DLC | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
|---|---|---|---|---|---|---|---|---|---|
| 0x5FF | 8 | 0x33 | 0x01 | 0x48 | 0xF4 | 0xFF | 0xFF | 0xFF | 0xFF |

Current =Signed Int32(0x FFFF F448) = -3000 mA (discharge)

1. Read bytes 2 and 3 in little endian => F448
2. Read byte 4 and 5 in little endian = >FFFF
3. Put together => FFFF F448
4. Do a 2s complement => 0000 0BB8 (See appendix D)
5. Convert to decimal => 3000mA

## C. CHECK SUM

The Check sum is an Exclusive OR Operation (XOR) of DATA3 to DATAn fields.

Examples: Looking at the Pack Voltage message, DATA3 to DATAn correspond to D3-D5 to the value of D6 should be equal to XOR of (D3, D4 and D5)

| Message Description | Type | ID | DLC | D0 | D1 | D2 | D3 | D4 | D5 | D6 (CRC) |
|---|---|---|---|---|---|---|---|---|---|---|
| **Pack Voltage** | Tx | 0x001 | 5 | 09 | 81 | 01 | 00 | 00 | | |
| | Rx | 0x500 | 7 | 09 | 81 | 01 | 01 | 37 | CD | FB |

This is computed as follows: convert hexadecimal to binary

| **D3 hex** | | **D4 hex** | | **D5 hex** | | **=** | **D3 bin** | | **D4 bin** | | **D5 bin** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 3 | 7 | C | D | = | 0000 | 0001 | 0011 | 0111 | 1100 | 1101 |

XOR operation (compare all bits if there is an odd number of 1s = 1 if an even #/no 1s = 0)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| D3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| D4 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| D5 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| **XOR** | **1** | **1** | **1** | **1** | **1** | **0** | **1** | **1** |

Convert XOR to hex and compare to D6

| XOR bin | | = | XOR hex | | D6 | |
|---|---|---|---|---|---|---|
| 1111 | 1011 | = | F | B | F | B |

## D. 2S COMPLEMENT

2s complements are done because the values are signed.
Using the first example of discharge current reading FFFE F10C
**Long way of computing:** Convert to binary, invert all values (NOT operation 1=0 and 0=1)
Add 1 and convert back to Hexadecimal

| **Hex** | **F** | | | | **F** | | | | **F** | | | | **E** | | | | **F** | | | | **1** | | | | **0** | | | | **C** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bin** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **NOT** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| **+** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 1 |
| **Bin** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **1** | **1** | **1** | **0** | **1** | **1** | **1** | **1** | **0** | **1** | **0** | **0** |
| **Hex** | **0** | | | | **0** | | | | **0** | | | | **1** | | | | **0** | | | | **E** | | | | **F** | | | | **4** | | | |

**Simpler way:** Take the maximum value possible with 8 bits (FFFF FFFF) and subtract the received value and add 1.

| | F | F | F | F | F | F | F | F |
|---|---|---|---|---|---|---|---|---|
| - | F | F | F | E | F | 1 | 0 | C |
| **+** | | | | | | | | 1 |
| **=** | **0** | **0** | **0** | **1** | **0** | **E** | **F** | **4** |

powerpros@relionbattery.com 855.931.2466 relionbattery.com