**PCAN Methods - Detailed Calculation Document**

**1. Device Name (device\_name)**

* **Raw Data: Typically received in two parts from two different CAN messages with the ID 0x1CECFFC0.**
* **Process:**
  + **The raw hexadecimal data is split across two hex strings.**
  + **These hex strings are converted to ASCII characters.**
  + **The two ASCII strings are concatenated to form the device name.**
* **Example:**
  + **Hex Strings: 0142542D37303933 and 0239415048010021**
  + **ASCII Strings: BT-7093 and 9APH!**
  + **Result: BT-70939APH!**

**2. Serial Number (serial\_number)**

* **Raw Data: Received in a single CAN message with the ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a single hexadecimal value.**
  + **This hex value is then converted to a decimal number, representing the serial number.**
* **Example:**
  + **Hex Data: ['0xc4', '0x5']**
  + **Swapped Hex: 0x05c4**
  + **Decimal Value: 1476**
  + **Result: 1476**

**3. Manufacturer Date (manufacturer\_date)**

* **Raw Data: Received from a CAN message with ID 0x1CEBFFC0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number representing the manufacturer date.**
* **Example:**
  + **Hex Data: ['0x3', '0x20']**
  + **Swapped Hex: 0x2003**
  + **Decimal Value: 8195**
  + **Result: 8195**

**4. Manufacturer Name (manufacturer\_name)**

* **Raw Data: Similar to device\_name, this is typically received in two parts from CAN messages with ID 0x1CECFFC0.**
* **Process:**
  + **The raw hexadecimal data is split across two hex strings.**
  + **These hex strings are converted to ASCII characters.**
  + **The two ASCII strings are concatenated to form the manufacturer name.**
* **Example:**
  + **Hex Strings: 014272656E2D5472 and 026F6E6963730100**
  + **ASCII Strings: Bren-Tr and onics**
  + **Result: Bren-Tronics**

**5. Firmware Version (firmware\_version)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The major version is extracted from the first byte.**
  + **The minor version is extracted from the second byte.**
  + **The patch number and build number are extracted by swapping the corresponding bytes.**
  + **These values are then concatenated into a string representing the firmware version.**
* **Example:**
  + **Hex Data: ['0x02', '0x01', '0x07', '0x00', '0x22', '0x00']**
  + **Firmware Version: 2.1.7.34**

**6. Battery Status (battery\_status)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The raw data is converted to a 16-bit binary string.**
  + **This binary string is interpreted as individual flags, each representing a different status (e.g., Overcharged Alarm, Terminate Charge Alarm, etc.).**
* **Example:**
  + **Binary Data: 1000000000000000**
  + **Flags: {'Overcharged Alarm': 1, 'Terminate Charge Alarm': 0, '...': ...}**
  + **Result: 1000000000000000**

**7. Cycle Count (cycle\_count)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number representing the cycle count.**
* **Example:**
  + **Hex Data: ['0x04', '0x00']**
  + **Swapped Hex: 0x0004**
  + **Decimal Value: 4**
  + **Result: 4**

**8. Design Capacity (design\_capacity)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
  + **The capacity is calculated by multiplying the decimal value by 40 and dividing by 1000.**
* **Example:**
  + **Hex Data: ['0xA5', '0x0A']**
  + **Swapped Hex: 0x0AA5**
  + **Decimal Value: 2725**
  + **Design Capacity: (2725 \* 40) / 1000 = 109.0 Ah**
  + **Result: 109.0 Ah**

**9. Design Voltage (design\_voltage)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
  + **The voltage is calculated by dividing the decimal value by 1000.**
* **Example:**
  + **Hex Data: ['0x70', '0x62']**
  + **Swapped Hex: 0x6270**
  + **Decimal Value: 25200**
  + **Design Voltage: 25200 / 1000 = 25.2 mV**
  + **Result: 25.2 mV**

**10. Remaining Capacity (remaining\_capacity)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
  + **The capacity is calculated by multiplying the decimal value by 40 and dividing by 1000.**
* **Example:**
  + **Hex Data: ['0x9D', '0x09']**
  + **Swapped Hex: 0x099D**
  + **Decimal Value: 2461**
  + **Remaining Capacity: (2461 \* 40) / 1000 = 98.44 Ah**
  + **Result: 98.44 Ah**

**11. Temperature (temperature)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
  + **The temperature in Celsius is calculated by converting Kelvin to Celsius.**
* **Example:**
  + **Hex Data: ['0xD5', '0x0B']**
  + **Swapped Hex: 0x0BD5**
  + **Decimal Value: 3029**
  + **Temperature: (3029 / 10) - 273.15 = 29.8°C**
  + **Result: 29.8°C**

**12. Current (current)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
  + **The current is calculated by multiplying the decimal value by 40 and dividing by 1000.**
* **Example:**
  + **Hex Data: ['0x89', '0x00']**
  + **Swapped Hex: 0x0089**
  + **Decimal Value: 137**
  + **Current: (137 \* 40) / 1000 = 5.48 A**
  + **Result: 5.48 A**

**13. Voltage (voltage)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
  + **The voltage is calculated by dividing the decimal value by 1000.**
* **Example:**
  + **Hex Data: ['0x53', '0x70']**
  + **Swapped Hex: 0x7053**
  + **Decimal Value: 28755**
  + **Voltage: 28755 / 1000 = 28.8 mV**
  + **Result: 28.8 mV**

**14. Average Current (avg\_current)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
  + **The average current is calculated by multiplying the decimal value by 40 and dividing by 1000.**
* **Example:**
  + **Hex Data: ['0x5E', '0x00']**
  + **Swapped Hex: 0x005E**
  + **Decimal Value: 94**
  + **Average Current: (94 \* 40) / 1000 = 3.76 A**
  + **Result: 3.76 A**

**15. Charging Current (charging\_current)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
  + **The charging current is calculated by multiplying the decimal value by 40 and dividing by 1000.**
* **Example:**
  + **Hex Data: ['0xE2', '0x04']**
  + **Swapped Hex: 0x04E2**
  + **Decimal Value: 1250**
  + **Charging Current: (1250 \* 40) / 1000 = 50.0 A**
  + **Result: 50.0 A**

**16. Full Charge Capacity (full\_charge\_capacity)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
  + **The full charge capacity is calculated by multiplying the decimal value by 40 and dividing by 1000.**
* **Example:**
  + **Hex Data: ['0x24', '0x0A']**
  + **Swapped Hex: 0x0A24**
  + **Decimal Value: 2596**
  + **Full Charge Capacity: (2596 \* 40) / 1000 = 103.84 Ah**
  + **Result: 103.84 Ah**

**17. Charging Voltage (charging\_voltage)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
  + **The charging voltage is calculated by dividing the decimal value by 1000.**
* **Example:**
  + **Hex Data: ['0xD8', '0x72']**
  + **Swapped Hex: 0x72D8**
  + **Decimal Value: 29400**
  + **Charging Voltage: 29400 / 1000 = 29.4 mV**
  + **Result: 29.4 mV**

**18. At Rate Time to Full (at\_rate\_time\_to\_full)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
  + **The time is calculated by dividing the decimal value by 1000.**
* **Example:**
  + **Hex Data: ['0xFF', '0xFF']**
  + **Swapped Hex: 0xFFFF**
  + **Decimal Value: 65535**
  + **At Rate Time to Full: 65535 / 1000 = 65.5 minutes**
  + **Result: 65.5 minutes**

**19. At Rate Time to Empty (at\_rate\_time\_to\_empty)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
  + **The time is calculated by dividing the decimal value by 1000.**
* **Example:**
  + **Hex Data: ['0xFF', '0xFF']**
  + **Swapped Hex: 0xFFFF**
  + **Decimal Value: 65535**
  + **At Rate Time to Empty: 65535 / 1000 = 65.5 minutes**
  + **Result: 65.5 minutes**

**20. At Rate OK Text (at\_rate\_ok\_text)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
  + **The text is determined based on whether the decimal value is 0 (No) or not (1 for Yes).**
* **Example:**
  + **Hex Data: ['0x01', '0x00']**
  + **Swapped Hex: 0x0001**
  + **Decimal Value: 1**
  + **Result: Yes**

**21. At Rate (at\_rate)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
  + **The at rate is calculated by multiplying the decimal value by 40 and dividing by 1000.**
* **Example:**
  + **Hex Data: ['0x00', '0x00']**
  + **Swapped Hex: 0x0000**
  + **Decimal Value: 0**
  + **Result: 0.0 A**

**22. Charging Battery Status (charging\_battery\_status)**

* **Raw Data: Derived from the current field.**
* **Process:**
  + **If the current value is positive, the battery status is "Charging".**
  + **If the current value is negative, the battery status is "Discharging".**
  + **If the current value is zero, the battery status is "Idle".**
* **Example:**
  + **Current Value: 5.48 A**
  + **Result: Charging**

**23. Relative State of Charge (rel\_state\_of\_charge)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
* **Example:**
  + **Hex Data: ['0x5B', '0x00']**
  + **Swapped Hex: 0x005B**
  + **Decimal Value: 91**
  + **Result: 91%**

**24. Absolute State of Charge (abs\_state\_of\_charge)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
* **Example:**
  + **Hex Data: ['0xFF', '0xFF']**
  + **Swapped Hex: 0xFFFF**
  + **Decimal Value: 65535**
  + **Result: 65535%**

**25. Run Time to Empty (run\_time\_to\_empty)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
  + **The time is calculated by dividing the decimal value by 10.**
* **Example:**
  + **Hex Data: ['0x4F', '0x00']**
  + **Swapped Hex: 0x004F**
  + **Decimal Value: 79**
  + **Run Time to Empty: 79 / 10 = 7.9 minutes**
  + **Result: 7.9 minutes**

**26. Average Time to Empty (avg\_time\_to\_empty)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
  + **The time is calculated by dividing the decimal value by 10.**
* **Example:**
  + **Hex Data: ['0xFF', '0xFF']**
  + **Swapped Hex: 0xFFFF**
  + **Decimal Value: 65535**
  + **Average Time to Empty: 65535 / 10 = 6553.5 minutes**
  + **Result: 6553.5 minutes**

**27. Average Time to Full (avg\_time\_to\_full)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
  + **The time is calculated by dividing the decimal value by 1000.**
* **Example:**
  + **Hex Data: ['0xFF', '0xFF']**
  + **Swapped Hex: 0xFFFF**
  + **Decimal Value: 65535**
  + **Average Time to Full: 65535 / 1000 = 65.5 minutes**
  + **Result: 65.5 minutes**

**28. Maximum Error (max\_error)**

* **Raw Data: Received in a single CAN message with ID 0x18FED0C0.**
* **Process:**
  + **The first two bytes are swapped and combined to form a hexadecimal value.**
  + **This hex value is converted to a decimal number.**
* **Example:**
  + **Hex Data: ['0x60', '0x00']**
  + **Swapped Hex: 0x0060**
  + **Decimal Value: 96**
  + **Result: 96%**