Battery Swapping Architecture (Protocol Specifications)

DRAFT VERSION 2.4.2

28TH FEB 2018

Revision History	Issued on	Release Notes
Draft V1.0	1st Jul 2017	
Draft V2.3.3.	19th Aug 2017	
Draft V2.3.4.	15th Sep 2017	
Draft V2.4	30th Sep 2017	
Draft V2.4.1	21st Nov 2017	1.Updated explanation in required places and the flow 2.Correction of typo/errors 3.Included parameter for continuous storage of battery's ambient temperature every 15 mins 4.Charging protocol a. Changes in the message code for SAE J compliance b. Included new data types for battery. 5.Driving Protocol a. Included Vehicle start message b. Included messages for storing Open command data from vehicle to BMS 6.Included BLE specification 7.Included OMS-Bulk charging protocol.
Draft V2.4.2	28th Feb 2018	Major changes are as below

Protocol	#	Changes
Charging Protocol	**1**	One-time data request value corrected from 0xAA to timestamp
	2	Addition of parameters: battery measured current and battery measured voltage during charging stage
	3	Charging Protocol includes BIN authorization with OMS before proceeding with charging. This is explicitly mentioned

				4	Lock-smart mode parameter included in the battery log list (It is already included in the expected messages but was missing in the overall listing of battery log parameters)
			Driving Protocol	5	BMR Message description changed to add more clarity.
				6	VPV PGN number with missing one zero is corrected.
				7	In Section 4.5.1.2/ 3-byte count for VCA message corrected to 3 bytes.
				8	Inclusion of battery temperature as a parameter to VCU during driving stage
				9	Vehicle controller current from VCU to BMS added as a new parameter
				10	BMS message for Master BMS to Slave BMS communication included
			OMS Protocol	11	Addition of 'ConnectorId' parameter in SetAdminConnectorState message and ChargerParamConfiguration message
				12	Data type for few parameters included in 8.6
				13	Current charging stage enumeration included
				14	JSON for BT8, BT9 and BT10 included
				15	Changes for configuring periodicity of data log retrieval (also updated in Charging Protocol)
				16	Effective resistance of power path threshold value setting
				17	UFD included in Battery charging one-time data
			Driving & Charging Protocol	18	Units for current and voltages made common across protocols
				19	The reset of state machine and start from beginning of protocol after each suspension included
				20	Changes for LS batteries authorization in Driving and Charging protocol
				21	Firmware version as a part of protocol during handshake stage

Contents

1. Scope 6
2. Entities Involved 6
   2.1. Entity Identifiers 7
3. Charging Protocol 12
   3.1. Physical Layer 12
   3.2. Data Link Layer 12
   3.3. Application Layer 13
   3.4. Overall Charging Procedure 13
   3.5. Charging Protocol Stages 14
4. Driving Protocol 45
   4.1. Physical Layer 46
   4.2. Data Link Layer 46
   4.3. Application Layer 46
   4.4. Overall Driving Procedure 47
   4.5. Driving Protocol Stages 48
5. Data Logging Retrieval Logic 65
   5.1. TSG Packet Format – Response packet with data list 65
   5.2. TT Packet Format – Request packet for data 66
   5.3. TTD Packet Format – Response packet with data 66
   5.4. Timestamp Format for Periodic Data 67
6. BMS – HHD – CMS Protocol 67
   6.1. HHD Functionalities 67
   6.2. BLE Specifications 68
7. BMS Functionality Guidelines 69
8. Charger and OMS communication 70
   8.1. Abbreviation 70
   8.2. Message Types 70
   8.3. Messages Initiated by OMS 70
   8.4. Messages Initiated by Charger 72

8.5. Messages Syntax 74

8.6. Data Types 93

9. References 96

Annexure A: Protocol Version Matching Flow 97

A.1 Charging Protocol Version Matching 97

A.2 Driving Protocol Version Matching Flow 100

Annexure B: Data Log Message Flow Diagrams 103

B1. Charging Data Log messages 103

B2. Driving Data Log Messages 105

B3. Using Driving periodic data retrieval granularity and Charging periodic data retrieval granularity 108

Annexure C: Auto-address assignment flow 110

C1. Driving protocol 110

C2. Charging protocol 110

1. Scope

This standard specifies various communication protocols involved in the battery swapping architecture. Fig 1 shows the Battery Swapping Architecture and the various entities involved.

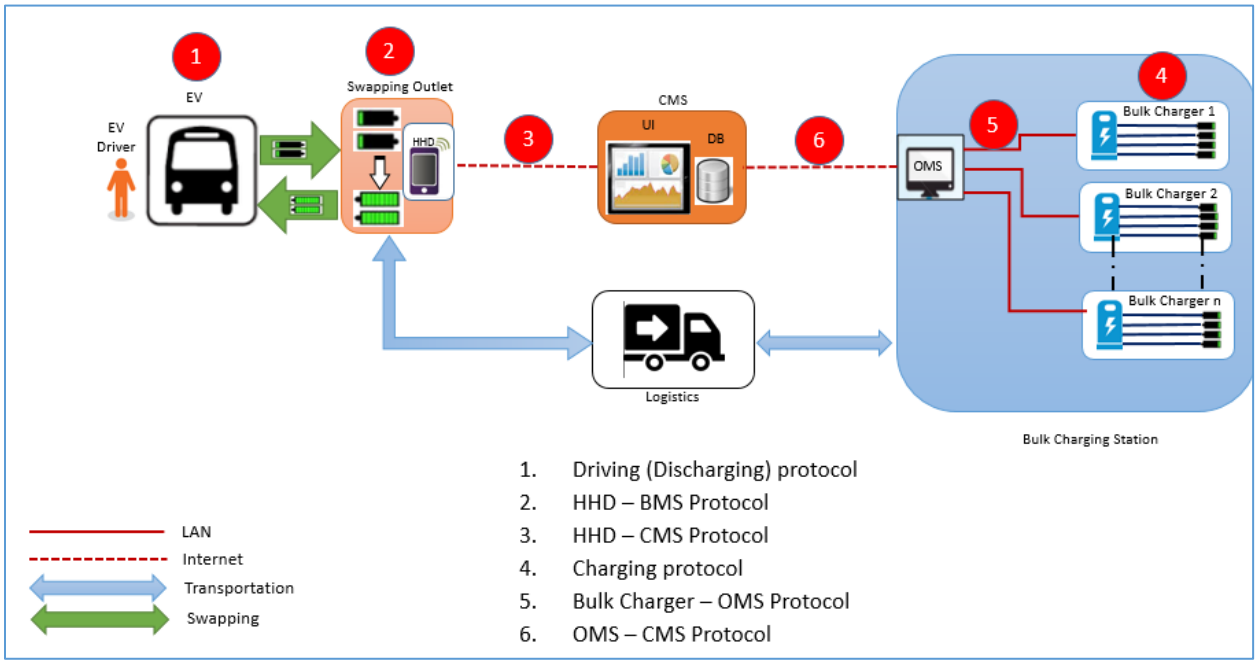


Fig 1: Battery Swapping Architecture

This document provides the draft specifications of the communication detailing of the following protocols:

1. Communication protocol between BMS and Bulk Charger – Charging Protocol
2. Communication protocol between BMS and the Electric Vehicle – Driving Protocol
3. Communication protocol between BMS and HHD
4. Communication protocol between HHD and CMS
5. Communication protocol between Bulk charger and OMS

2. Entities Involved

- EV: Assumed to be E-Rickshaw, E-auto or E-bus.
- EV Driver: Person who drives EV
- Battery Management System (BMS) and Battery modules
  - Discharged/Charged batteries received from /placed in Vehicle
- Swapping Outlet (SO): A small shop where battery swapping is done. Can be same as charging station.
- Hand-held device (HHD) in swapping outlets
  - A device to identify and assign batteries at swapping outlets
  - Used for calculating energy consumption units and billing
- Central Management System(CMS)
  - A cloud server to collect data, check & authorize operations from various entities

- Bulk Charging Station(BCS)
    - A Central station to where all the discharged batteries from various swapping outlets are brought for charging and charged batteries will be distributed
    - There could be multiple Bulk chargers per BCS and the charging function is done by Bulk Charger(BC)
- Operational Management System(OMS)
    - OMS - a local server to collect, check and authorize operations from all the bulk chargers in the bulk charging station
    - One OMS will be available per bulk charging station
    - OMS perform battery pairing. Battery pairing is required at the swapping outlet to help the operator in assigning charged batteries to the vehicle.

2.1. Entity Identifiers

2.1.1. Vehicle Identification Number – VIN (Existing Definition/

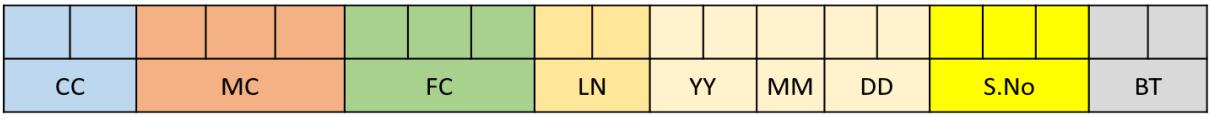
- VIN - 17-digit code and does not include alphabets I, O, Q to avoid confusion with numerals 1 and 0

Standard	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
ISO 3779	World manufacturer identifier			VDS						VIS							

- WMI - World Manufacturer Identifier [Digits 1-3]
    - Assigned by SAE (Society of Automotive Engineers) in US to countries and manufacturers
    - First two digit indicates country code: India country code: MA- ME
    - Third digit indicates manufacturer E.g. J- Ford India, L – Hyundai, 1 – Mahindra etc. E.g. WMI – MAL: India Hyundai, MB1: India Ashok Leyland etc.
- VDS - Vehicle Descriptor Section [Digits 4-9]
    - Carries manufacturer specific information like engine type, vehicle model, body type, transmission etc.
    - Besides letter I, O, Q not being used in VIN, letters U and Z are not used in VDS
    - Used according to local regulations to identify vehicle type and each manufacturer has unique system for using this field to identify the vehicle
- VIS - Vehicle Identifier Section [Digits 10-17]
    - Digit 10 – Model year of the vehicle
    - Digit 11 – Plant code of the manufacturer
    - Digit 12-17 – 6-digit Serial number of the vehicle

2.1.2. Battery Identification Number – BIN (Proposed)

- BIN Components – 20 Digits

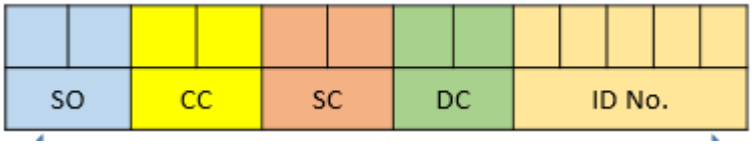


#	Components	Abbreviation	Description	Reference	Digits
1	Country code	CC	To identify the country	ISD country codes can be used Ref: https://countrycode.org/	2
2	Manufacturer Code	MC	To identify the manufacturer	To be assigned by some regulatory authority to identify different manufacturers. E.g. EXI – Exide, AMR-Amara raja	3
3	Factory code of the manufacturer	FC	To identify the factory code where the battery manufacturing is done	This could be manufacturer specific data to identify the factory	3
4	Line Number in the factory	LN	To identify the line number in the factory	This could be manufacturer specific data to identify the factory	2
5	Production Date	YYMMDD	Year, Month and date of production	Year: Offset to be 2017. E.g. 1 indicates 2017, 2 indicates 2018 and so on up to 99 Month: A-Jan, B-Feb, C-Mar…. J-Nov, K-Dec Date: 01, 02, 03… 30, 31.	2+1+2=5
6	Serial Number	S.No	Serial number of the battery	Three-digit serial number of the battery ranging from 1 to 4095(0X01H to 0xFFFH)	3
7	Battery Type	BT	Battery chemistry type	01H: lead acid battery;02H: nickel hydrogen battery; 03H: lithium iron phosphate battery; 04H: lithium manganite battery; 05H: cobalt based lithium battery; 06H: ternary material battery; 07H: polymer lithium-ion battery;08H: lithium ion battery; 09H: NMC (Lithium	2

				Nickel Manganese Cobalt Oxide)0AH: NCA (Lithium Nickel Cobalt Aluminum Oxide)0BH: Lithium titanate oxide (LTO)0CH: Lithium Nickel cobalt manganese FFH: other batteries	

2.1.3. Swapping Outlet Identifier – SOID (Proposed)

- SOID Components – 13 Digits

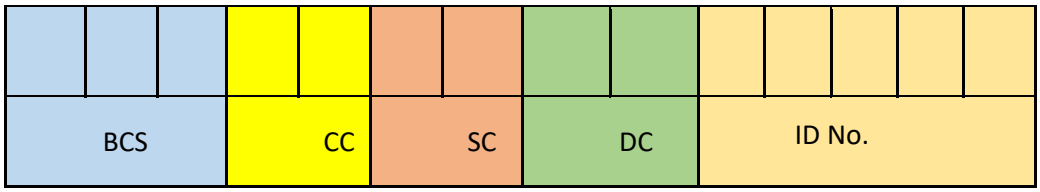


#	Components	Abbreviation	Description	Reference	Digits/ Bytes
1	Entity Name: Swapping Outlet	SO	To identify the entity name in the infrastructure	Taken from the first letters from each word of entity name	2
2	Country Code	CC	To identify the country	ISD country codes can be used Ref: https://countrycode.org/	2
3	State code	SC	To identify the state in which the SO is present. State codes used in vehicle number plate can be taken as reference. CMS will have list of state code. This would be captured during registration of SO with CMS and generated based on user's selection of location	Ref:https://en.wikipedia.org/wiki/Vehicle_registration_plates_of_India#Current_codes	2

4	District Code	DC	To identify the district of respective state in which the SO is present. CMS will have list of state-wise district code. This would be captured during registration of SO with CMS and generated based on user's selection of location	Ref: https://en.wikipedia.org/wiki/List_of_districts_in_India	2
5	ID Number	ID No.	5-digit number to identify the SO	Auto-generated 5-digit ID from CMS upon successful registration	5

2.1.4. Bulk Charging Station Identifier – BCSID (Proposed)

- BCSID – 14 Digits

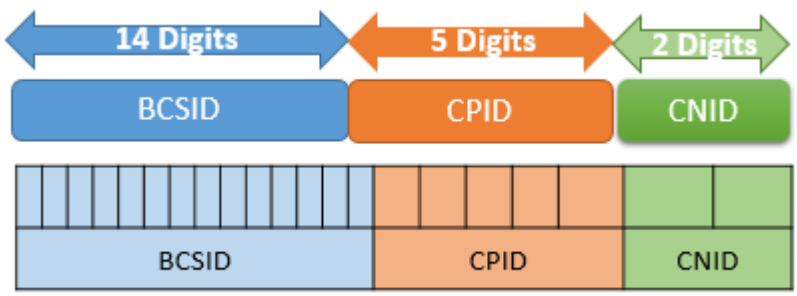


#	Components	Abbreviation	Description	Reference	Digits/Bytes
1	Entity Name: Bulk Charging Station	BCS	To identify the entity name in the infrastructure	Taken from the first letters from each word of entity name	3
2	Country Code	CC	To identify the country	ISD country codes can be used. Ref: https://countrycode.org/	2
3	State code	SC	To identify the state in which the BCS is present. State codes used in vehicle number plate can be taken as reference.	Ref:https://en.wikipedia.org/wiki/Vehicle_registration_plates_of_India#Current_codes	2

			CMS will have list of state code. This would be captured during registration of SO with CMS and generated based on user's selection of location		
4	District Code	DC	To identify the district of respective state in which the BCS is present. CMS will have list of state-wise district code. This would be captured during registration of SO with CMS and generated based on user's selection of location	Ref:https://en.wikipedia.org/wiki/List_of_districts_in_India	2
5	ID Number	ID No.	5-digit number to identify the BCS	Auto-generated 5-digit ID from CMS upon successful registration	5

2.1.5. Bulk Charger and Slot Identifiers – Proposed

Bulk Charger and slot identifier – 21 digits



- Each Bulk Charging Station (BCS) is assumed to have multiple chargers with 15/30/60 charging channels.
- Each charger would be referred as Charge Point(CP) and identified using CPID: CP followed by three-digit number

  E.g. CP001, CP002 ……CP00N

- Each slot/charger channel would be referred as Connector (CN) and identified using CNID by two-digit number.

  E.g. 0x01, 0x02, 0x03……

3. Charging Protocol

The Communication protocol between BMS and Bulk Charger will be referred as charging protocol.

- The charging function is done by bulk charger
- The communication network between bulk charger and battery adopts CAN 2.0B communication protocol
- During charging, the charger and BMS monitors parameters such as Voltage, current and temperature.
- The battery log data captured during driving is transferred to OMS through Bulk charger as a part of this protocol.
- Charging Protocol Version: 0.9.0

3.1. Physical Layer

Physical Layer conforming to this standard shall refer to ISO 11898-1:2003 and SAE J1939-14: 201612. The communication between charger and BMS in this standard shall use the CAN interface. The communication rate between charger and BMS would be 500 Kbit/s.

3.2. Data Link Layer

3.2.1. Frame Format

Equipment complying with this standard shall use 29-bit identifier of CAN extended frame, and the corresponding definition of each specific bit allocation shall meet the requirements as given in SAE J1939-21:2006.

3.2.2. Protocol Data Unit (PDU)

Each CAN data frame contains a single protocol data unit (PDU). The protocol data unit is composed of seven parts which respectively are priority, reserved bit, data page, PDU format, specific PDU, source address and data field.

3.2.3. PDU Format (PF)

In this standard, the PDU1 format defined in SAE J1939-21:2006 is used.

3.2.4. Parameter group number (PGN)

The second byte of PGN is PDU format (PF) value, and both high byte and low byte are 00H.

3.2.5. Functions of transport protocol

The transport of 9~1785-byte data between BMS and charger shall use the transport protocol function. The specific connection initialization, data transport and connection closing shall comply with

the provisions on message transport as given in 5.4.7 and 5.10 of SAE J1939-21:2006. As for the multi-frame message, the message period refers to the transport period for the whole data package.

3.2.6. Address allocation

Network address is used to guarantee the uniqueness of message identifier and to indicate the message source. The Charger address is fixed as 128(Dec) or 80H and battery address is assigned by the charger as a part of the protocol.

3.3. Application Layer

- The application layer is defined as set of parameters and parameter groups.
- Parameter group is numbered by PGN, and each node identifies the content of data packet according to PGN.
- Data are transported in the form of periodical transport and event-driven mode.
- In case that multiple PGN data need to be transmitted to realize one function, it requires receiving multiple PGN messages of this definition to judge the successful transmission of this function.
- The message options may be either mandatory or optional. If all the contents in the same frame of message are optional, such message may not be transported; if some contents in the same frame of message are optional, all the optional bits are transported in the format as specified in this standard or filled with 1; the invalid bit or field not specified in this standard is filled with 1.
- The length of message and content and format of mandatory item shall be transported as detailed in subsequent chapters.

3.4. Overall Charging Procedure

The Communication protocol between a Bulk Charger (BC) and Battery Management System (BMS) comprises of following stages, after the establishment of physical connection: Auto-address assignment Stage, Handshake Stage, Battery Authenticity Check Stage, Driving log Transfer Stage, Parameter Configuration Stage, Charging Stage, Charging log transfer stage and End-of-charging stage.

In each stage, if the charger or BMS does not receive message from the other party or does not receive correct message within the stipulated time limit, the waiting entity will timeout (timeout means failure to receive a complete data package or correct data package within specified time); unless otherwise specified, the timeout is all 5s. After timeout BMS or charger will send suspending message.

The figure below presents the overall charging process.

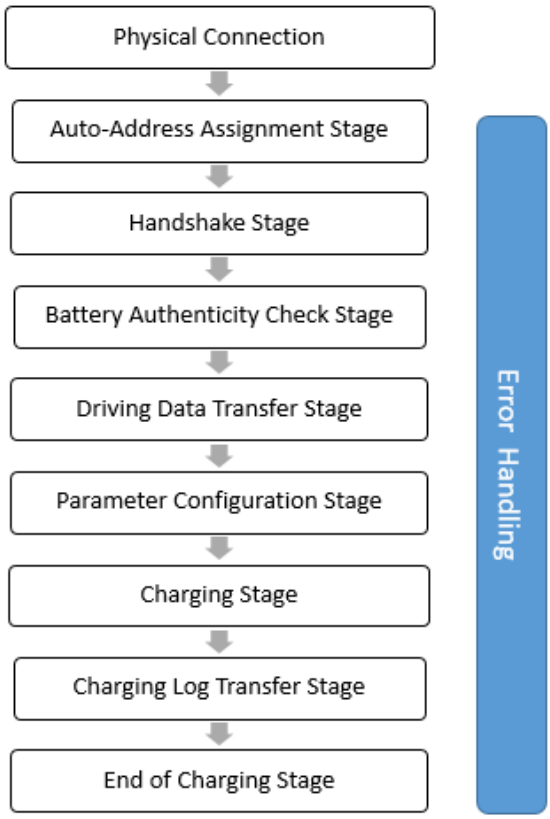


Fig: Overall Charging Process

3.5. Charging Protocol Stages

The bulk charger will be capable of charging 15/30/60 batteries simultaneously. Bulk charger will assign address automatically to each BMS as it gets plugged-in.

For all the parameters, the order of SPNs in the CAN communication is as per the order specified in the table itself. For any suspension during the protocol flow, the state machine will reset and start from the beginning of the protocol.

3.5.1. Auto address assignment stage

The BMS address will not be hard coded in them, instead when modules are connected to bulk charger (BC), BC will assign address to each module. BC should reserve the address in the range of 0x95 -0x185 for BMS addressing. BC's source address (SA) is defined as 128 (80H). By default, BMS should have default address as 254(0xFE) i.e. Null address.

- **Message1: Request for address claim from BMS**

BMS will send a request for address claim by generating a random number (RN1- say 2E2614D0) of 4 bytes in the data field. Bytes 1 to 4 will be used for this. Unused bytes in the data field will be filled with 0x00.

SA	DA	DATA Byte:01-04	DATA Byte:05	DATA Byte:06	DATA Byte:07	DATA Byte:08

FE (Null Address)	80 (BC Address)	2E2614D0	00	00	00	00

- **Message2: Broadcast response for address claim from BC**

BC broadcasts to CAN bus with the same random number (RN1- 2E2614D0) and allotted address (say 0x95). The allotted address will be available in the 5th byte of data field.

SA	DA	DATA Byte:01-04	DATA Byte:05	DATA Byte:06	DATA Byte:07	DATA Byte:08
80 (VCU Address)	FF (Broadcast Address)	2E2614D0	95	00	00	00

- **Message3: BMS confirmation request for allotted address from BMS**

BMS requests BC to confirm the usage of the allotted address by generating and sending another random number (RN2 – say 33AB7F30) and allotted address (0x95) to BC.

SA	DA	DATA Byte:01-04	DATA Byte:05	DATA Byte:06	DATA Byte:07	DATA Byte:08
FE (Null Address)	80 (BC Address)	33AB7F30	95	00	00	00

- **Message4: VCU confirmation response for allotted address from BC**

BC broadcasts to CAN bus with the random number (RN2), allotted address (0x95) and address status (0xAA: Success; 0xFF: Failure). The address status will be available in the 6th byte of data field. If the status from BC is a failure i.e. the address being allotted to some other BMS, then BMS has to repeat and start from Message1.

SA	DA	DATA Byte:01-04	DATA Byte:05	DATA Byte:06	DATA Byte:07	DATA Byte:08
80 (BC Address)	FF (Broadcast Address)	33AB7F30	95	AA	00	00

- **Message 5: BMS confirmation on allotted address**

BMS confirms to BC on the allotted address by sending the random number 2, allotted address and the confirmation status (0xAA: Success; 0XFF: Failure)

SA	DA	DATA Byte:01-04	DATA Byte:05	DATA Byte:06	DATA Byte:07	DATA Byte:08
FE (Null Address)	80 (BC Address)	33AB7F30	95	AA	00	00

The failure status would occur, if BMS is unable to assign the allotted address for some reasons. In this case, the BMS must repeat and start from Message1 to get an address assigned.

When BC sees a success status, it ensures that this address is not given to any other BMS.

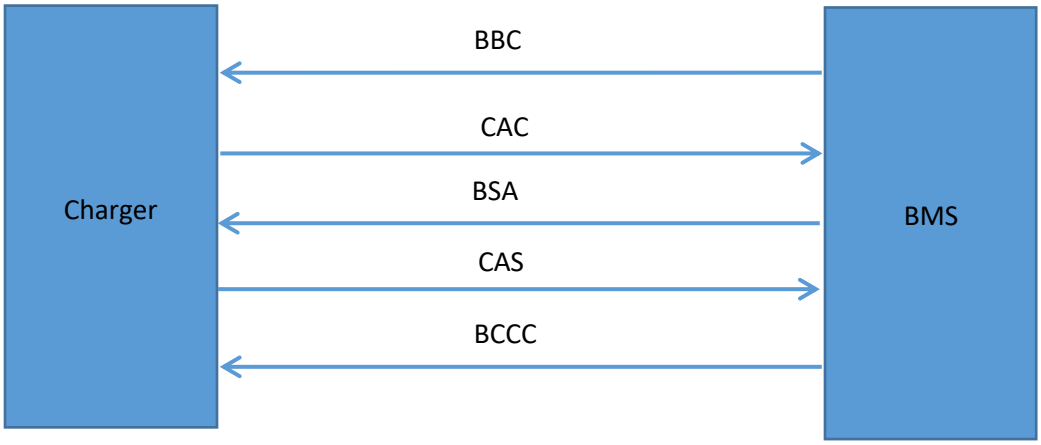
The address assignment for other BMS will happen in parallel in the same manner. The following use case is defined to elaborate on address assigning procedure under few conditions:

Case 1: Bulk Charger (BC) receiving the same random number from two BMS (Message1)

- BC will broadcast RN1+allotted address as defined in Message2.
- Both the BMS will receive this response and will send a confirmation request for allotted address with random number 2(RN2).
- BC will receive the message and by looking at the allotted address being common in the packets, it will send (Message4 with AA in byte 06) success to first BMS and failure(Message4 with 00 in byte 06) to other BMS.
- BMS receiving success status will continue with Message5 onwards and BMS receiving failure status will start from Message1
- Probability of occurrence of random number 2 being same for two BMS is very less and assumed to not occur.

Refer Annexure C for auto-address assignment flow

3.5.1.1. Message Flow



3.5.1.2. Messages

Message Code	Messages	Direction	PGN (HEX)	Priority	Size in Bytes	Time Period (ms)
BBC	Battery address claim request message	BMS To Charger	001000H	4	4	250

CAC	Charger broadcasts response for address claim message	Charger to BMS	002600H	4	5	250
BSA	BMS confirmation address request Message	BMS to Charger	002700H	4	5	250
CAS	Charger confirmation address response	Charger to BMS	002800H	4	6	250
BCC	BMS confirmation response on allotted address	BMS to Charger	001100H	4	6	250

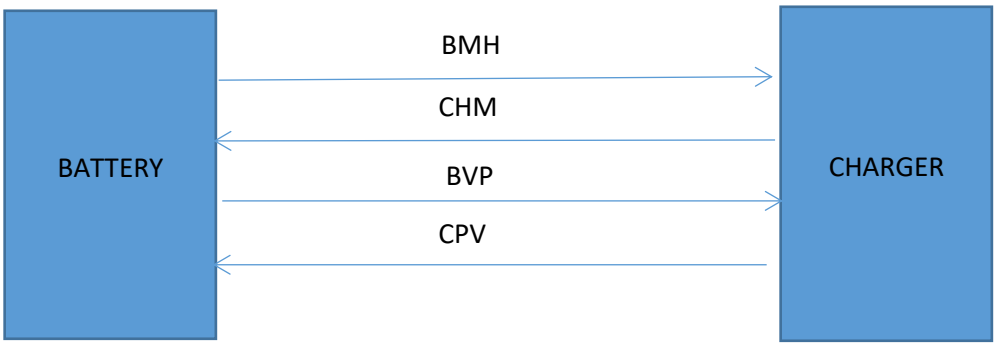
3.5.1.3. Parameters

Message Code	Parameter Name	Description	SPN in (Dec)	Size in Bytes	Delivery Option
BBC	Random Number 1	Random number 1 generated by BMS	346	4	Mandatory
CAC	Random Number 1	Random number received in BBC	289	4	Mandatory
	Allotted address	Address allotted by charger	290	1	Mandatory
BSA	Random Number 2	Random number 2 generated by BMS	291	4	Mandatory
	Allotted address	Allotted address in CAC	292	1	Mandatory
CAS	Random number 2	Random number 2 received in BSA message	347	4	Mandatory
	Allotted address	Allotted address by charger in CAC message	348	1	Mandatory
	Allotted status Success=0xAA; Failure=0xFF	Allotted address status. Charger confirming the address status as success or failure	349	1	Mandatory
BCC	Random Number 2	Random number 2 generated in BSA message	350	4	Mandatory
	Allotted address	Allotted address in CAC messages	351	1	Mandatory
	Acceptance status Success=0xAA; Failure=0xFF;	BMS acceptance status for allotted address	352	1	Mandatory

3.5.2. Handshake stage

In this stage, both charger and BMS communicates the charging protocol version between them. Annexure A1 provides details on the protocol version matching.

3.5.2.1. Message Flow



3.5.2.2. Messages

Message Code	Message Description	Source – Destination	PGN (HEX)	Priority	Data Length in Bytes	Message Period (ms)
BMH	Battery module Handshake Message	BMS to Charger	002900H	6	34	250
CHM	Charger Handshake Message	Charger to BMS	002A00H	6	6	250
BVP	Battery Protocol Version Confirmation Message	BMS to Charger	002B00H	6	3	250
CPV	Charger Protocol Version Acknowledgment Message	Charger to BMS	002C00H	6	3	250

3.5.2.3. Parameters

Message Code	Parameter Name	Description	SPN (Dec)	Size in Bytes	Delivery Option
BMH	BIN*	Unique battery identification number	293	20	Mandatory
	BMS communication protocol version	Communication protocol version number of BMS	294	3	Mandatory

	BMS Firmware version	Firmware version of BMS	2565	3	Mandatory
	UFD (Unique ID for drive)	Unique ID for drive assigned during battery issue in the swapping outlet	2566	8	Mandatory
CHM	Bulk Charger communication protocol version	Communication protocol version number of charger	295	3	Mandatory
	Charger firmware version	Firmware version of charger	2567	3	Mandatory
BVP	Confirmed version of BMS communication protocol	Based on charger's protocol version, BMS will confirm the version number	296	3	Mandatory
CPV	Protocol version acknowledgement	Acknowledgement on protocol version from BMS. Success:0xAA; Failure:0XFF	297	1	Mandatory
	Driving periodic data retrieval granularity**	Granularity of driving periodic (1 Sec) data retrieval	2572	1	Mandatory
	Charging periodic data retrieval granularity**	Granularity of charging periodic (1 Sec) data retrieval	2573	1	Mandatory

*Before continuing with the charging, charger must authorize battery by sending BIN number to OMS server. Refer Section 8.5.2.3 for the relevant message.

**The BMS will have to peruse this data to decide on the granularity of data that will be sent to charger. For e.g., if the time-stamp sent by charger to retrieve data is T and the granularity is G, then the BMS will have to provide the value of the corresponding type which has time-stamp T+G. Refer appendix B3 for an example illustration.

3.5.3. Battery authenticity check stage

Authenticity check is done in two stages:

Stage 1 – Authenticity check initiated by Charger

In this stage, a random number (RN1) is generated by charger and sent to BMS. BMS runs an algorithm and sends the result to charger (Result). Charger runs the same algorithm and based on the result, it authenticates the battery.

For internal testing purpose, Result = RN1 / 2 could be assumed with some random number as RN1.

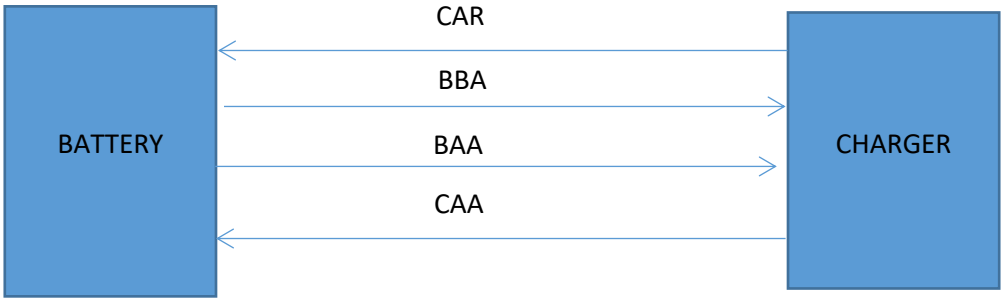
Stage2: Authenticity check initiated by Battery

In this stage, a random number (RN2) is generated by BMS and sent to charger. Charger runs an algorithm and sends the result to BMS (Result). BMS runs the same algorithm and based on the result, it authenticates the Charger.

For internal testing purpose, Result = RN2 / 2 could be assumed with some random number as RN2.

The proprietary algorithm will be provided by Energy Business company for both the stages: Authentication from Charger as well as from Battery as binaries. The BMS and Charger OEMs would have to develop their firmware using the same.

3.5.3.1. Message Flow



3.5.3.2. Messages

Message Code	Messages	Direction	PGN (HEX)	Priority	Size in Bytes	Time Period (ms)
CAR	Charger authenticity Request Message initiated by charger	Charger to BMS	002D00H	6	4	250
BBA	BMS authenticity Response Message for request initiated by charger	BMS to Charger	002E00H	6	4	250
BAA	BMS authenticity request message initiated by battery	BMS to Charger	001D00H	6	4	250
CAA	Charger authenticity response message for request initiated by battery	Charger to BMS	001E00H	6	4	250

3.5.3.3. Parameters

Message Code	Parameter Name	Description	SPN (DEC)	Size in Bytes	Delivery Option
CAR	Charger Random Number for Authentication	authenticity request initiated by Charger	298	4	Mandatory
BBA	BMS Response for Given Random Number	Battery authenticity response	299	4	Mandatory
BAA	Battery random number for authentication	Authenticity request initiated by battery	2568	4	Mandatory
CAA	Charger response for given random number	Authenticity response by charger for battery initiated request	2569	4	Mandatory

3.5.4. Driving log transfer stage

3.5.4.1. Driving Data Log

This section describes the data log parameters captured during driving. These data are transferred from BMS to charger during driving data transfer stage using defined packet formats. At the end of this stage, the BMS should clear all the driving data log to free the memory to enable BMS to store next session of driving data. BMS should make sure that data is cleared only after being completely transferred to the charger.

When all data is transferred for a given type or if there is no data available for transfer, BMS should respond with a packet with all '0's to indicate this.

These packets are formed by grouping various parameters. The charger will interpret the data based on the defined packet formats. Following groups of data are logged

- Battery related log
- Vehicle related log
- Vehicle related additional log
- Battery related additional log

There could be either one or multiple batteries (e.g. 2 to 3 batteries for 3W) in each vehicle and one among these batteries will act as Master BMS to provide required parameters to VCU. The assignment of Master BMS will be handled by VCU during the address assignment stage. Each BMS will have the mode as Master (0x01) or Slave(0x00) based on the assignment from VCU.

3.5.4.1.1. Driving Log Parameters

The parameters are logged either one-time at each ignition of the vehicle or periodically during running of the vehicle along with the timestamp.

- One-time parameters: All the one-time parameters are combined to form a packet. This single packet of defined size would be transferred from BMS to charger on one-time data request during driving data log transfer stage. This data is stored for every start of the vehicle.

- Periodic parameters: All periodic parameters are defined with granularities based on the data type they carry. They are combined to form a packet along with the timestamp for each record. These periodic data would be transferred from BMS to charger on periodic data request during driving data log transfer stage. The data are split up into multiple types and transferred based on the request type initiated by the charger. The timestamp for periodic data size would be 4 bytes and the format is defined in Section: 5.4.
- Additional Parameters: In addition to the above defined data, protocol allows extension to log additional battery related data. Additional data logging can be made for periodic or event based parameters. In case of event based parameters, the granularity should be '0'.
- The data logging is retrieved by defined packet structure and methods. This retrieval logic is detailed in Section 5.

3.5.4.1.2. Battery Data Log Parameters

Battery Data Log in each battery module						
#	**Parameter Name**	**Description**	**Size in Bytes**	**Parameter Nature**	**Nature**	**Log Size**
1	Battery Mode: Master(0x01) /Slave(0x00)	Mode showing whether battery was master /slave during driving. This parameter will help charger to retrieve additional vehicle data from BMS. BMS must keep this parameter set even if it has become slave when it had failed and resumed back	1	One-time	Vehicle Ignition	1
2	Driving Protocol Version	Driving Protocol Version	3	One-time	Vehicle Ignition	3
3	BIN	BIN number	20	One-time	Vehicle Ignition	20
4	VIN	VIN Number	17	One-time	Vehicle Ignition	17
5	Available energy	Available energy (w-hr.) in battery at start of vehicle	2	One-time	Vehicle Ignition	2
6	Number of cells	Number of cells in battery	1	One-time	Vehicle Ignition	1
7	Number of temp. sensors	Number of temperature sensors in battery	1	One-time	Vehicle Ignition	1
8	BMS Firmware Version	Firmware version of BMS	3	One-time	Vehicle Ignition	3
9	Lock-Smart Mode	Lock-smart mode: Drive mode 0x01/Charge mode 0x00	1	Periodic	15 mins	1

10	Available energy in w-Hr	Available energy in battery run in w-Hr	2	Periodic	1 Sec	2
11	Maximum current that battery can provide in Deci Amperes	Maxi. Current battery can provide at the instant of logging	2	Periodic	1 Sec	2
12	Battery instantaneous voltage in Deci volt	Battery instantaneous voltage	2	Periodic	1 Sec	2
13	Battery instantaneous current in Deci ampere	Battery instantaneous current	2	Periodic	1 Sec	2
14	Individual cell voltage in centi volt	Individual cell voltage and will be based on number of cells	2	Periodic	1 Sec	2* no. of cells
15	Individual Sensor temperature	Individual temperature of temp. sensors in 0.1°C. and will be based on sensors count	2	Periodic	1 Sec	2* no. of temperature sensors
16	Individual Balancing cell status	If cell is balanced '0' will have to be stored and '1' if cell is not balanced. Cell 0 status is stored in LSB and Cell 32 status is stored in MSB	4	Periodic	1 Sec	4
17	Suspending /Alert reasons	Whenever / BMS suspends or sends alerts	Based on the reason	Additional data - Event-based	0	Based on the reason
18	Battery Ambient temperature	Battery ambient temperature in 0.1°C.	2	Periodic	15 mins.	2

3.5.4.1.3. Battery Data Storage format in BMS

- Battery one-time data storage format in BMS

Parameter Name	Timestamp	Battery Mode: Master /Slave	Driving Protocol Version	BIN	VIN	No. of cells	No. of temp. sensors	Available energy	BMS Firmware Version
Size	4	1	3	20	17	1	1	2	3

- Battery periodic data storage format in BMS

BMS will have two set of records for storing various parameters. These records are defined based on the granularity of the parameter.

Record 1: All parameters with a granularity of 1 sec are grouped to form record 1

Parameter Name	Time Stamp	Available Energy	Max. current that battery can provide	Battery instantaneous voltage	Battery instantaneous current	Cell number 1	Cell Voltage 1	Cell number 2	Cell voltage 2
Size (Bytes	4	2	2	2	2	1			

Record 1 table continued …

Cell number n	Cell Voltage n	Temp. Sensor Number 1	Temp. Sensor Value 1	Temp. Sensor 2	Temp. Sensor Value 2	Temp. sensor number n	Temp. Sensor value n	Individual cell temperature[1-n]	Balancing current status
1								2* No of temperature sensors	4

Record 2: Event based suspending reason is formed as Record 2

Parameter Name	Timestamp	Suspending reason + Data
Size (Bytes)	4	Based on the reason

Record 3: Battery ambient temperature is formed as Record 3

Parameter Name	Timestamp	Lock-smart mode Drive mode:0x01 /charge mode :0x00	Battery ambient temperature
Size (Bytes)	4	1	2

3.5.4.1.4. Vehicle Data Log

Vehicle Data Log in Master Battery Module						
#	**Parameter Name**	**Description**	**Size**	**Parameter Nature**	**Granularity**	**Log Size**
1	VIN - Vehicle Identification Number	VIN Number	17	Vehicle Ignition	Once	17
2	Effective resistance of power path	The voltage across the cable connecting battery and the vehicle controller	2	Vehicle Ignition	Once	2
3	Odometer reading	Odometer reading in KM at the start of vehicle	4	Vehicle Ignition	Event	4
4	VCU Firmware Version	Firmware version of VCU	3	Vehicle Ignition	Event	3
5	Vehicle speed	Vehicle speed in Deci kmph during driving	2	Periodic	1 Sec	2
6	Odometer reading	Odometer reading	4	Periodic	1 Sec	4
7	Vehicle controller current	Vehicle controller current	2	Periodic	1 Sec	2
8	OC1	Open Command 1	2	Additional data - Periodic	10 Sec	2
9	OC2	Open Command 2	2	Additional data - Periodic	10 Sec	2
10	OC3	Open Command 3	2	Additional data - Periodic	60 Sec	2
11	OC4	Open Command 4	2	Additional data - Periodic	60 Sec	2
12	Suspending/ Alert reasons	Vehicle suspending/alert reasons	Based on the reason	Event	0	Based on the reason

3.5.4.1.5. Vehicle Data Storage format in BMS

- Vehicle one-time data storage format in BMS

Parameter Name	Timestamp	VIN - Vehicle Identification Number	Effective resistance of power path	Odometer reading	VCU Firmware Version
Size	4	17	2	4	3

- Vehicle periodic data storage format in BMS

BMS will have the following set of records for storing various parameters at various granularity levels.

Record 1:

Parameter Name	Timestamp	Vehicle Speed	Odometer Reading	Vehicle controller current
Size	4	2	4	2

- **Record 2:**

Parameter Name	Timestamp	OC1	OC2
Size	4	2	2

- **Record 3:**

Parameter Name	Timestamp	OC3	OC4
Size	4	2	2

- **Record 4:**

Parameter Name	Timestamp	Suspending reason + Data
Size	4	Based on the reason

3.5.4.1.6. Memory Calculation for driving data log

The approximate memory calculation for various data logs are calculated by assuming the no. of cells and no. of temperature sensors as 32.

- Battery Data Log Size

Battery Data Log	Per Sec/ Per 15 mins	Per Hour (B/Hr)	Per 15 Hours
One-time Data+ Timestamp	52	52	780
Periodic data 1 sec + timestamp (B/Sec)	154	554400	8316000
Periodic data 15 mins + timestamp (B/15 min)	7	28	420
Total (Bytes)	213	554480	8317200
Total (KB)		541.484375	8122.2656

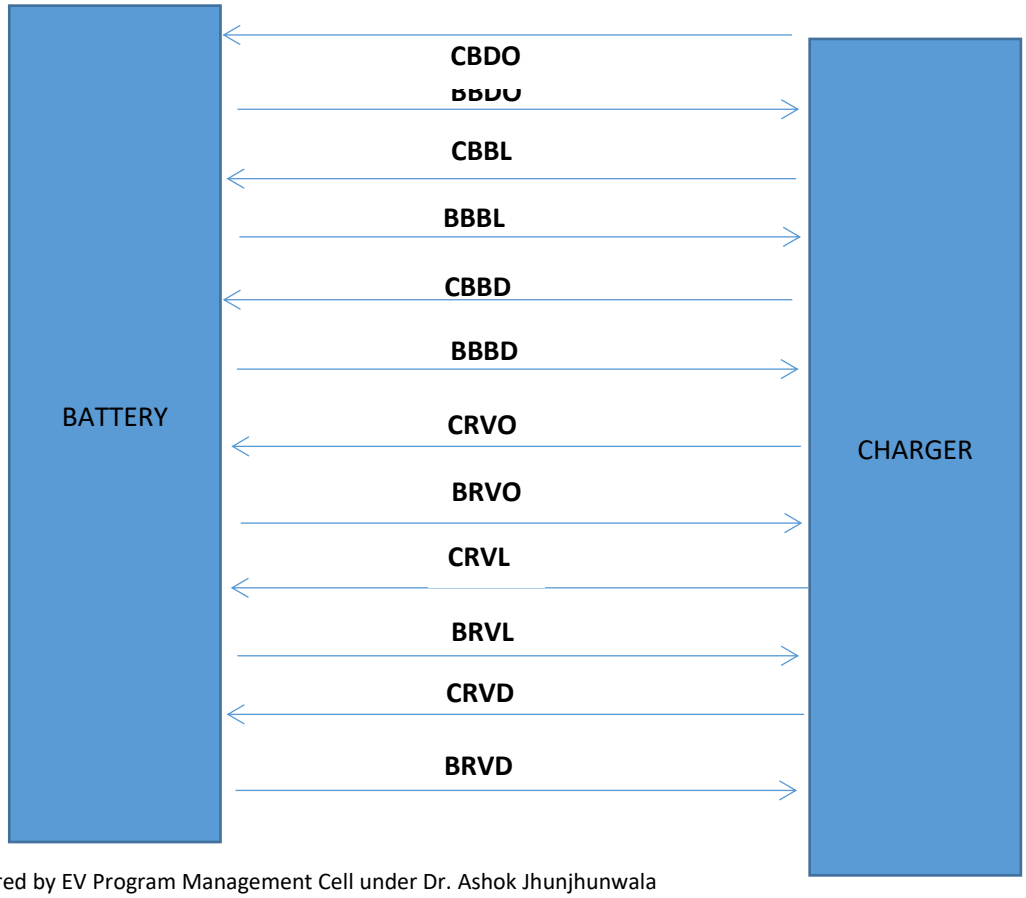
- Vehicle Data Log Size

Vehicle Data Log	Per Sec/ Per 10 Sec /Per 60 Sec	Per Hour (B/Hr)	Per 15 Hours
One-time Data+ Timestamp	30	30	450
Periodic data 1 sec + timestamp (B/Sec)	12	43200	648000
Periodic data 10 sec + timestamp (B/10 sec)	4	1440	21600
Periodic data 60 sec+ timestamp (B/60sec)	8	480	7200
Event-based data	10	10	150
Total (Bytes)	46	45160	677400
Total (KB)		44.10156	661.523438

- Total Data Log Size

Total Memory Log data size per 15 hours		
Battery Stats per 15 Hours	7959.15	KB/15 Hours
Vehicle stats per 15 hours	286.67	KB/15 Hours
Total	8245.88	KB/15 Hours

3.5.4.2. Message Flow



3.5.4.3. Messages

Message Code	Messages	Direction	PGN (HEX)	Priority	Size in Bytes	Time Period(ms)
CBDO	Charger Request battery discharge one-time data	Charger - Battery	002F00H	6	4	250
BBDO	Battery Response Battery discharge one-time data	Battery - Charger	003000H	6	52	250
CBBL	Charger request battery additional data list	Charger – Battery	003300H	6	1	1000
BBBL	Battery response additional data list in TSG format	Battery - Charger	003400H	6	5	250
CBBD	Charger request battery additional data	Charger – Battery	003500H	6	5	250
BBBD	Battery response battery additional data	Battery - Charger	003600H	6	14	250
CRVO	Charger Request vehicle onetime data	Charger - Battery	003700H	6	4	250
BRVO	Battery Response vehicle one-time data	Battery - Charger	003800H	6	30	250
CRVL	Charger request vehicle additional data list	Charger - Battery	003B00H	6	1	250
BRVL	Battery Response vehicle additional data list	Battery - Charger	003C00H	6	5	250
CRVD	Charger request vehicle additional data	Charger - Battery	003D00H	6	5	250
BRVD	Battery response vehicle additional data	Battery - Charger	003E00H	6	14	250

*For BT and VT types refer Section: 3.5.4.5.2. BBBL Packet format and Section: 3.5.4.5.6. BRVL Packet format respectively

3.5.4.4. Parameters

Message Code	Parameter Name	Description	SPN (DEC)	Size in Bytes	Delivery Option
CBDO	One-time battery data request Value: timestamp *	One-time data Request value	300	4	Mandatory
BBDO	One-time battery data Response Packet	Sends response packet in the format defined in the Section: BBDO packet format	301	52	Mandatory
CBBL	Additional battery data List request value: 0xAA	Charger requesting battery to send the data list.	304	1	Mandatory
BBBL	Additional battery data list Response	Battery response on data list. Response packet as defined in section BBBL packet format.	305	5	Mandatory
CBBD	Battery Data Request packet	Charger request packet as defined in section CBBD packet format	306	5	Mandatory
BBBD	Battery Data Response packet	Sends response packet in the format defined in the Section: BBBD packet format	307	14	Mandatory
CRVO	Vehicle one-time data Request Value: timestamp*	Charger Request Vehicle's one-time data Request value	308	4	Mandatory
BRVO	Vehicle one-time data Response Packet	Sends response packet in the format defined in the Section: BRVO packet format	309	30	Mandatory
CRVL	Vehicle additional data list request packet: Value 0xAA	Charger Request vehicle additional data list from BMS.	312	1	Mandatory
BRVL	Vehicle additional data lit response data	Battery Response packet as defined in Section BRVL packet format.	313	5	Mandatory
CRVD	Vehicle additional data request packet	Charger request packet as defined in section CRVD packet format	314	5	Mandatory

BRVD	Vehicle additional data response packet	Sends response packet in the format defined in the Section: BRVD packet format	315	14	Mandatory

Note: Refer Annexure B2 on detailed message flow diagrams

* By default, the first time-stamp should be 01/01/2017 00:00:00. The format of timestamp is as defined in Section: Timestamp format for periodic data

3.5.4.5. Packet formats

3.5.4.5.1. BBDO Packet Format

Parameter Name	Timestamp	Battery Mode	Driving Protocol Version	BIN	VIN	Number of cells	Number of temp. sensors	Available energy	BMS Firmware Version
Size	4	1	3	20	17	1	1	2	3
Example	0x17022906H	1	0x010101H	INAMRTNC1217A244898	19UYA31581L00000	32	32	1	0X00010109H

3.5.4.5.2. BBBL Packet Format

Parameter Name	**Type**	**Size(Bytes)**	**Granularity(sec)**
Size(Bytes)	1	2	2
Example for BT1	1	6	1

. For data retrieval of the stored battery log, the following types are defined:

Type Name	**Type Number**	**Size (Bytes)**	**Granularity (sec)**	**Parameters**
BT1	1	4+2	1	Timestamp+ Available Energy
BT2	2	4+2	1	Timestamp+ Maximum current that battery can provide
BT3	3	4+2	1	Timestamp+ Battery instantaneous voltage

BT4	4	4+2	1	Timestamp+ Battery instantaneous current
BT5	5	4+1+2	1	Timestamp + Cell number + Individual cell voltage
BT6	6	4+1+2	1	Timestamp + Sensor number + Individual sensor temp.
BT7	7	4+4	1	Timestamp + Balancing current status
BT8	8	4+2+4+4	0	Timestamp+ Error code+ Threshold value+ Breach value e.g. For excessive dis-charge current, the data will be Timestamp+0002(Error code) +discharge current(threshold value)+discharge current(breach value)
BT9*	9	4+2+17	0	Timestamp of BT8+ErrorCode of BT8+VIN number
BT10	10	4+1+2	900	Timestamp+Lock-smart mode + Battery ambient temperature

*This will be generated only for Error code 0001H – BIN-VIN mis-match

The periodic and additional data frame

3.5.4.5.3. CBBD Packet Format

Parameter Name	Type n	Timestamp
Size in Bytes	1	4
Example	1	16022906H

Note: By default, the first time-stamp should be 01/01/2017 00:00:00. The format of timestamp is as defined in Section: Timestamp format for periodic data

3.5.4.5.4. BBBD Packet Format

Parameter Name	Timestamp	Type n	Data
Size in Bytes	4	1	2
Example	16022906H	1	700

The charger would send the default timestamp along with requested type using CBBD packet. Based on the request type, BMS would send the data corresponding to that type using BBBDP packet. This response packet would contain the next timestamp to be sent in CBBD packet.

The first request packet would have the default timestamp with type 1 say BT1 with timestamp1. After receiving the response, charger would initiate request packet with timestamp received in response packet say timestamp2. In this way, all the transfer of BT1 would be completed and the charger will initiate the transfer with the next type say BT2 with default timestamp

When CBBD is received for BT5 and BT6 with say Timestamp1, BMS will send ‘n’ number of responses for that timestamp, where ‘n’ will be equal to number of cells for BT5 and number of temperature sensors for BT6.

For e.g. if the number of cells is 32, then BMS will send 32 responses for the timestamp received in CBBD.

3.5.4.5.5. BRVO Packet Format

Parameter Name	Timestamp	VIN - Vehicle Identification Number	Effective resistance of power path	Odometer Reading	VCU Firmware version
Size	4	17	2	4	3

3.5.4.5.6. BRVL Packet Format

Parameter Name	Type	Size(Bytes)	Granularity(sec)
Size(Bytes)	1 byte	2 bytes	2 bytes
Example for VT1	1	2	0

For data retrieval of the stored vehicle log, the following types are defined.

Type Description	Type Number	Size (Bytes)	Granularity (Secs)	Parameters
VT1	1	Based on suspending reason	0	Suspending reason+ Data(based on suspending data) e.g. For battery authenticity failure 4002(Error code)+Vehicle result + BMS random number
VT2	2	2	1	Vehicle speed
VT3	3	2	10	OC1
VT4	4	2	10	OC2
VT5	5	2	60	OC3
VT6	6	2	60	OC4
VT7	7	2	1	Odometer reading
VT8	8	2	1	Vehicle controller current

3.5.4.5.7. CRVD Packet Format

Parameter Name	Type n	Timestamp
Size in Bytes	1	4
Example	1	16022906H

Note: By default, the first time-stamp should be 01/01/2017 00:00:00. The format of timestamp is as defined in Section: Timestamp format for periodic data

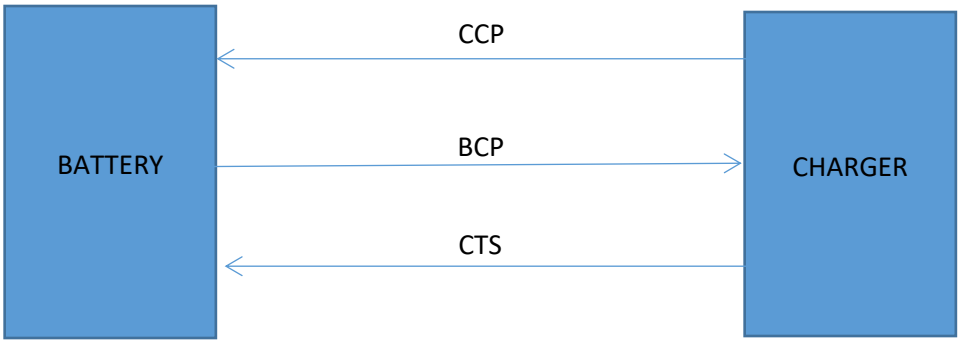
3.5.4.5.8. BRVD Packet Format or Additional Vehicle data log format

Parameter Name	Timestamp	Type n	Data
Size in Bytes	4	1	2
Example for OC1	16022906H	1	500

3.5.5. Parameter configuration stage

In this stage, charger and the BMS exchanges required parameter for charging the battery.

3.5.5.1. Message Flow



3.5.5.2. Messages

Message Code	**Messages**	**Direction**	**PGN (HEX)**	**Priority**	**Size in Bytes**	**Time Period (ms)**
CCP	Charger Charging Parameter Set Message	Charger to BMS	003F00H	6	6	250
BCP	Battery Charging Parameters Message	BMS to Charger	004000H	6	8	250
CTS	Charger Time synchronization Message(CTS)	Charger to BMS	004100H	6	7	250

3.5.5.3. Parameters

Message Code	Parameter Name	Description	SPN (DEC)	Size in Bytes	Delivery Option
CCP	Target SoC	Target SoC to be set in battery from charger	316	2	Mandatory
	Effective resistance of power path Threshold value	Threshold value for effective resistance of power path for setting in BMS	366	2	Mandatory
	Effective resistance of power path	Effective resistance of power path value	2574	2	Mandatory
BCP	Maximum charging voltage of battery pack	Battery pack charging voltage in Centi Volt	317	2	Mandatory
	Maximum charging current of battery pack	Battery maximum acceptable charging current in Centi Amperes	318	2	Mandatory
	Bulk charging current of battery pack	Initial estimated bulk charging (CC) current in Centi Amperes	319	2	Mandatory
	Charge termination current of battery pack	Battery pack CV phase charge termination current in mA	320	2	Mandatory
CTS	Charger Timestamp	Charger timestamp sent to battery to sync battery's timestamp. CTS timestamp as given in Section: CTS Timestamp Format	321	7	Mandatory

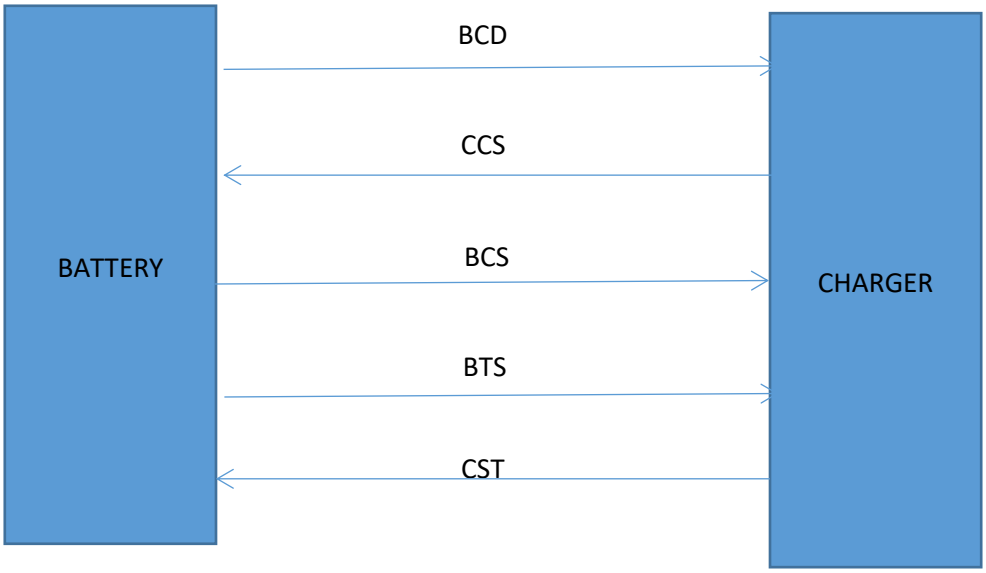
3.5.5.3.1. CTS Timestamp Format

Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Year		Month	Day	Hours	Minutes	Seconds

3.5.6. Charging stage

Throughout the charging stage, BMS will periodically send battery charging demand to the charger, and the charger will regulate the charging voltage and charging current according to battery charging demand to ensure the normal proceeding of charging process. During the charging process, charger and BMS will send their respective charging state.

3.5.6.1. Message Flow



3.5.6.2. Messages

Message Code	Messages	Direction	PGN (HEX)	Priority	Size in Bytes	Time Period (ms)
BCD	Battery charging demand Message(BCD)	BMS to Charger	004200H	4	5	1000
CCS	Charger Charging state message	Charger to BMS	004300H	4	4	1000
BCS	Battery Charging State Message	BMS to Charger	004400H	4	7	1000
BTS	Battery Suspending Message	BMS to Charger	004500H	2	10	250
CST	Charger Suspending Message	Charger to BMS	004600H	2	10	250

3.5.6.3. Parameters

Message Code	Parameter Name	Description	SPN (DEC)	Size in Bytes	Delivery Option
BCD	Battery current demand	Current requested by battery in Centi Amperes	322	2	Mandatory
	Battery voltage demand	Voltage requested by battery in Centi Volts	303	2	Mandatory
	Charging mode	Battery charging mode requested by battery CVC=0x01; CCC=0x02	323	1	Mandatory
CCS	Charger output voltage	Charger output voltage in Centi Volts	324	2	Mandatory
	Charger output current	Charger output current in Centi Amperes	325	2	Mandatory
BCS	Estimated remaining charging time	Estimated remaining charging time in minutes	326	2	Mandatory
	Current SOC	Battery pack instantaneous SOC in % with a rounded value	327	1	Mandatory
	Measured battery current	Measured battery current in Centi-Amperes	2570	2	Mandatory
	Measured battery voltage	Measured battery voltage in Centi-volts	2571	2	Mandatory
BTS	Normal, Warning or Error suspension	Battery suspending reasons	328	2	Mandatory
	BTS Threshold value	Threshold value for suspending parameter	329	4	Mandatory
	BTS Breach value	Breach value for suspending parameter	330	4	Mandatory
CST	Normal, Warning or Error suspension	Charger suspending reasons	331	2	Mandatory
	CST Threshold Value	Threshold value for suspending parameter	332	4	Mandatory
	CST Breach Value	Breach value for suspending parameter	333	4	Mandatory

3.5.6.4. BMS Suspending Reasons (BTS)

With two bytes, the error values could be from 1-65534 (0x1 – 0x FFFF). While error codes 0x1 – 0x7FFF are reserved for protocol specific, 0x8000 – 0xFFFF are available for manufacturer specific code. The table below summarizes the suspending / alert code range and allocated ranges.

#	Suspending /Alert code specific to	Range	
1	Overall range	0x0001-0xFFFF	
2	Protocol specific	0x001 – 0x7FFF	Battery Specific: 0x0001- 0x3FFF
			Charger Specific: 0x4000 – 0x7FFF
3	Manufacturer specific	0x8000 – 0xFFFF	Battery Specific:0x8000 – 0xBFFF Charger Specific:0xC000 - 0xFFFF

While suspending, BMS will give error code and threshold with breach value for that particular error code, if available. If there are no appropriate values could be sent on threshold and breach values, it has to be filled as FFFFH.

Suspension Type	**Reason**	**Hexa decimal Code**	**BTS Threshold Value**	**BTS Breach Value**
Normal Suspension	Reached the required SOC target value	0001H	FFFFH	FFFFH
Error Suspension	Battery Connection Check failure. This condition is generated when the connector effective resistance is greater than 25mohm	0002H	FFFFH	FFFFH
	Battery Authenticity failure	0003H	BMS random number challenge response	Charger random number challenge
	Charging current is over or greater than the battery demand current	0004H	Demand current	Charging current
	Charging voltage mis-match with battery demand voltage	0005H	Demand voltage	charging voltage
	Battery cell over-temperature	0006H	Threshold temp.	Excess temp.
Warning Suspension	BMS component over temperature	0007H	Threshold temp.	Excess temp.

	Battery over-temperature	0008H	Threshold temp.	Excess temp.
	Battery cell over-temperature	0009H	Cell number + Threshold temp.	Cell number + Excess temp.
Other Suspensions	Other suspensions	3FFEH	Threshold value	Breach value

3.5.6.5. Charger Suspending Reasons (CST)

The suspending/alert code for charger will be in the range of 0x4000-0x7FFF

While suspending, charger will give error/alert code and threshold with breach value for that particular code, if available. If there are no appropriate values could be sent on threshold and breach values, it must be filled as FFFFH.

Suspension Type	**Reason**	**Hexadecimal Code**	**CST Threshold Value**	**CST Breach Value**
Normal Suspension	Suspending due to reaching target SoC set by the charger	4001H	SoC	FFH
Error Suspension	BIN acknowledgement failure Result	4002H	FFH	FFH
	Battery authenticity failure Result	4003H	Charger Random number	Random number response
	Protocol version acknowledgment error	4004H	Charger protocol version	BMS Protocol version
	Battery parameters compatibility result failure	4005H	FFH	FFH
	Battery demand parameters compatibility result failure	4006H	FFH	FFH
	Emergency stop fault for charger	4007H	FFH	FFH
Warning Suspension	Internal temperature of charger is excessive	4008H	Threshold temp.	Excess temp.
Other Suspension	Other Suspension	7FFFH	Threshold value	Breach Value

3.5.7. Charging log transfer stage

After charging stage, both BMS and charger will enter charging log transfer stage. During this stage, BMS will send the charging statistical data captured during the charging process to the charger.

3.5.7.1. Charging Data Log

This section describes the data log parameters captured during charging. The log data is transferred from BMS to charger using defined packet formats. These packets are formed by grouping various parameters. The charger will interpret the data based on the defined packet formats.

3.5.7.1.1. *Charging Log Parameters*

The following table details the parameters logged during charging:

#	Parameter Name	Description	Parameter Nature	Granularity	Data Size(B)	Log Size/Sec
1	BIN	Unique battery Identification number	One-time	Once	20	20
2	Number of cells	Total number of cells in battery	One-time	Once	1	1
3	Number of temperature sensors	Total number of temp. sensors	One-time	Once	1	2
3	Start SOC	SoC at the start of charging	One-time	Once	2	2
4	End SOC	SoC at the end of charging	One-time	Once	2	2
5	Total energy received while charging	Total energy transferred in w-Hr	One-time	Once	2	2
6	Time required for current charging session	Time in seconds of last charging	One-time	Once	2	2
7	Charging life cycle number	The number of times the battery has undergone charging. This value will be available from BMS and BMS should increment this counter by one for each charging cycle and store the value	One-time	Once	2	2
8	Battery firmware version	Firmware version of battery during charging	One-time	Once	3	3

9	Charger firmware version	Firmware version of charger	One-time	Once	3	3
10	Balancing current status	Balancing current of each cell during charging stage	Periodic	10 Sec	4	4
11	Individual cell voltage	Individual cell voltage during charging stage	Periodic	10 Sec	2	2B * no. of cells
12	Individual cell temperatures	Individual cell temperatures in 0. 1°C.during charging stage	Periodic	10 Sec	2	2B * no. of temp. sensors

3.5.7.1.2. Battery Data Storage format in BMS

- Battery one-time data storage format in BMS

Parameter Name	Time stamp	BIN	No. of cells	No. of temp. sensors	Start SoC	End SoC	Total energy passed while charging	Time required for last charging session	Charging life cycle number	BMS Firmware Version	Charger Firmware Version
Size	4	20	1	1	2	2	2	2	2	3	3

- Battery periodic data storage format in BMS
  For a battery with 'n' cells and 't' temperature sensors, the data storage format could be as below:

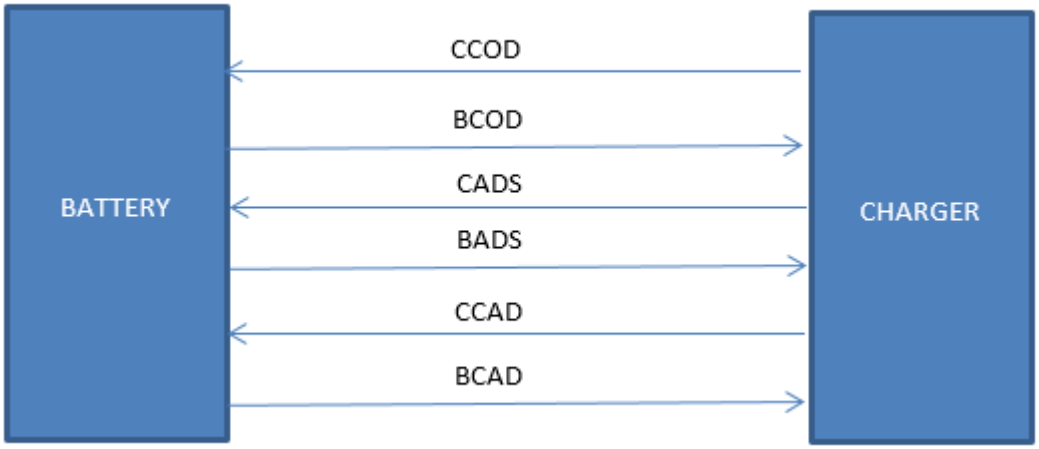
Parameter Name	Time Stamp	Balancing current status	Cell no.-1	Cell voltage-1	Cell no.-n	Cell voltage - n	Temp. sensor-1	Temp. sensor value-1	Temp. sensor -t	Temp. sensor value-t
Size (Bytes	4	4	1	2	1	2	1	2	1	2

3.5.7.1.3. Memory Calculation

One-time data in Bytes	35
Periodic data in Bytes/10 Sec	224
Periodic data in KB /Hour	78.75
Total KB / 2 Hours	192.5

*For e-auto, the no. of maximum cells assumed to be 32

3.5.7.2. MessageFlow



3.5.7.3. Messages

Message Code	Messages	Direction	PGN (HEX)	Priority	Size in Bytes	Time Period (ms)
CCOD	Charger request for Battery One-time data	Charger to BMS	004700H	6	1	250
BCOD	Battery one-time data Response	BMS to Charger	004800H	6	40	250
CADS	Charger request for Battery additional data list	Charger to BMS	004B00H	6	1	250
BADS	Battery additional data list response	BMS to Charger	004C00H	6	5	250
CCAD	Charger request for battery additional data	Charger to BMS	004D00H	6	5	250
BCAD	Battery additional data response	BMS to Charger	004E00H	6	Depends on the CT type	250

Note: Refer Annexure B1 on detailed message flow diagrams

3.5.7.4. Parameters

Message Code	Parameter Name	Description	SPN (DEC)	Size in Bytes	Delivery Option
CCOD	Request Value: timestamp*	One-time data Request value	334	1	Mandatory

BCOD	Response Packet	Sends response packet in the format defined in the Section: BCOD packet format	335	40	Mandatory
CADS	Additional data list: AAH	Request different data list from BMS	338	1	Mandatory
BADS	Response data list	Response packet as defined in Section BADS packet format	339	5 Bytes	Mandatory
CCAD	Request Packet	Charger request packet as defined in Section CCAD format	340	5	Mandatory
BCAD	Response data	Sends response packet in the format defined in the Section: BCAD packet format	341	Depends on the CT type	Mandatory

* Note: By default, the first time-stamp should be 01/01/2017 00:00:00. The format of timestamp is as defined in Section: Timestamp format for periodic data

3.5.7.5. Packet Formats

3.5.7.5.1. BCOD Packet Format

Parameter Name	Timestamp	BIN	No. of cells	No. of temp. sensors	Start SoC	End SoC	Total energy passed while charging	Time required for last charging session	Charging life cycle number	BMS Firmware Version	Charger Firmware Version
Size	4	20	1	1	2	2	2	2	2	3	3

3.5.7.5.2. BADS Packet Format

Parameter Name	**Type**	**Size(Bytes)**	**Granularity(sec)**
Size(Bytes)	1 byte	2 bytes	2 bytes
Example for CT1	1	2	10

The type, size and granularity will be identified from type number, size and granularity from the following table number. For data retrieval of the stored battery log during charging, the following types are defined:

Type Name	Type Number	Size(Bytes)	Granularity (Secs)	Parameters
CT1	1	4+4	10	Timestamp + Balancing current status
CT2	2	4+1+2	10	Timestamp + Cell number + Individual cell voltage
CT3	3	4+1+2	10	Timestamp + Sensor number + Individual sensor temp.

3.5.7.5.3. CCAD Packet Format

Parameter Name	Type n	Timestamp
Size in Bytes	1	4
Example for CT1	1	16022906H

Note: By default, the first time-stamp should be 01/01/2017 00:00:00. The format of timestamp is as defined in Section: Timestamp format for periodic data

3.5.7.5.4. BCAD Packet Format

When CCAD is received for CT2 and CT3 with say Timestamp1, BMS will send 'n' number of responses for that timestamp, where 'n' will be equal to number of cells for CT2 and number of temperature sensors for CT3.

For e.g. if the number of cells is 32, then BMS will send 32 responses for the timestamp received in CCAD.

Parameter Name	Timestamp	Data
Size in Bytes	4	3
Example for CT3	16022906H	500

3.5.8. End-of-Charging Stage

In this stage, if charging is successfully terminated, the battery lock-smart is changed from charge mode (0x00) to drive mode(0x01). Battery acknowledges the mode change. If the BMS sends a failure or if time-out occurs, then charger will indicate this as error in the suspension message.

3.5.8.1. Message Flow

3.5.8.2. Messages

Message Code	Message Description	Source – Destination	PGN (HEX)	Priority	Data Length in Bytes	Message Period (ms)
CCM	Charger Battery mode change message	Charger to BMS	004F00H	6	1	250
BCM	Battery mode change acknowledge Message	BMS to Charger	005000H	6	1	250

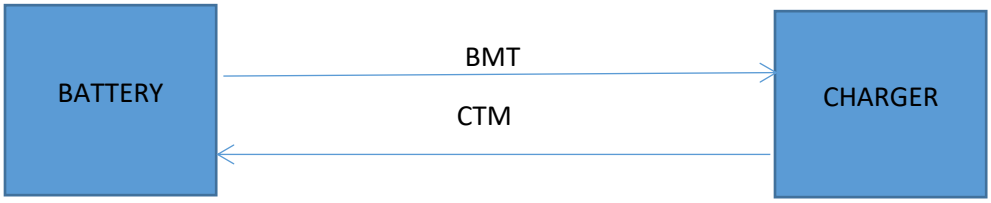
3.5.8.3. Parameters

Message Code	Parameter Name	Description	SPN (DEC)	Size in Bytes	Delivery Option
CCM	Battery lock-smart mode: Drive mode:0x01	Charger requesting battery to change mode from charge mode to drive mode	342	1	Mandatory
BCM	Battery mode acknowledge	Battery acknowledging mode change. Success: 0xAA; Failure: 0XFF	343	1	Mandatory

3.5.9. Time-out Messages

The time-out of messages during the whole charging process is communicated using these messages. This message will carry the second byte of PGN to indicate the timed-out message.

3.5.9.1. Message Flow



3.5.9.2. Messages

Message Code	Messages	Direction	PGN (HEX)	Priority	Size in Bytes	Time Period (ms)
BMT	BMS time-out Message	Battery to Charger	005100H	2	1	250
CTM	Charger time-out message	Charger to Battery	005200H	2	1	250

3.5.9.3. Parameters

Message Code	Parameter Name	Description	SPN (DEC)	Size in Bytes	Delivery Option
BMT	Second Byte of PGN Value of timed-out message	Battery Messages time-out	344	1	Mandatory
CTM	Second Byte of PGN Value of timed-out message	Charger Messages time-out	345	1	Mandatory

4. Driving Protocol

The communication protocol between Vehicle and BMS will be referred as “Driving Protocol”. This protocol will be divided into multiple stages based on the operation to be performed.

- When the batteries get placed inside the vehicle, the module address would be assigned by the vehicle controller automatically. VCU would reserve addresses in the range of 0x90 to 0x94 for addressing BMS modules.
- This auto-address assignment should happen only once when the battery gets placed and not on every start of the vehicle and is taken care by BMS. By default, BMS address would be 254(0xFE), Whenever vehicle is started, a vehicle start message is broadcasted to all BMS. When BMS receives this message, it checks its address and if it other than 254(0xFE), then it checks for its mode (Master/Slave) with vehicle and starts reacting based on the mode assigned by the VCU
- By default, the battery module with least address 0x90 will act as master and take care of the communication with vehicle controller as well as logging vehicle related data. If there is a failure in Master BMS, then vehicle would time-out for some message, suspend and stop as there would not be any communication with Master BMS.
- When the Vehicle is again started, BMS will have its address assigned so it will initiate the mode (Master/Slave) confirmation message to VCU. By sensing only two requests and will the Source address, VCU will identify the failure of Master BMS and will re-assign the next slave as Master.
- While assigning the Master BMS, VCU will inform the other module address as well as its role to all the BMS. This way all the BMS will come to know which the module’s address which will be acting as master BMS.
- Throughout the driving protocol, various parameters are logged at different time-intervals and based on events. Refer section, 3.5.4.1. for the details on the data and their format in which to be logged and the way they must be shared with charger.
- Vehicle related data log is done only in Master BMS to avoid duplicate data in all the batteries. This decision is made based on the battery mode (Master/Slave).
- When there is a Master BMS failure and when the slave gets assigned as Master, master BMS should retain a flag to indicate that it had acted as Master earlier. This would be required at the charging protocol during data retrieval.

4.1. Physical Layer

Physical Layer conforming to this standard shall refer to ISO 11898-1:2003 and SAE J1939-14: 201612. The communication between VCU and BMS in this standard shall use the CAN interface. The communication rate between vehicle and BMS would be 500 kbit/s.

4.2. Data Link Layer

4.2.1. Frame Format

Equipment complying with this standard shall use 29-bit identifier of CAN extended frame, and the corresponding definition of each specific bit allocation shall meet the requirements as given in SAE J1939-21:2006.

4.2.2. Protocol Data Unit (PDU)

Each CAN data frame contains a single protocol data unit (PDU). The protocol data unit is composed of seven parts which respectively are priority, reserved bit, data page, PDU format, specific PDU, source address and data field.

4.2.3. PDU Format (PF)

In this standard, the PDU1 format defined in SAE J1939-21:2006 is selected.

4.2.4. Parameter group number (PGN)

The second byte of PGN is PDU format (PF) value, and both high byte and low byte are 00H.

4.2.5. Functions of transport protocol

The transport of 9~1785-byte data between BMS and charger shall use the transport protocol function. The specific connection initialization, data transport and connection closing shall comply with the provisions on message transport as given in 5.4.7 and 5.10 of SAE J1939-21:2006. As for the multi-frame message, the message period refers to the transport period for the whole data package.

4.2.6. Address allocation

Network address is used to guarantee the uniqueness of message identifier and to indicate the message source. The Vehicle address is fixed as 129(Dec) or 81H and BMS address is assigned by the VCU as a part of the protocol.

4.3. Application Layer

- The application layer is defined in manner of parameters and parameter group.
- Parameter group is numbered by PGN, and each node identifies the content of data packet according to PGN.
- Data are transported in the form of periodical transport and event-driven mode.
- In case that multiple PGN data need to be transmitted to realize one function, it requires receiving multiple PGN messages of this definition to judge the successful transmission of this function.
- The message options may be either mandatory or optional. If all the contents in the same frame of message are optional, such message may be transported by filling 1The length of message and content and format of mandatory item shall be transported as detailed in subsequent chapters.

4.4. Overall Driving Procedure

The whole driving process comprises of four stages: Auto-address assignment stage, Handshake stage, Parameter exchange stage and driving stage. After the physical connection is established, the protocol starts the communication with auto-address assignment stage followed by handshake, parameter exchange and driving stages.

The auto-address assignment stage and the handshake stage will be repeated for the number of BMS present in the vehicle and for the other stages, VCU will communicate only with Master BMS. In Parameter exchange stage and driving stage, there will be an internal communication from Slave BMS to Master BMS to update the available energy. Master BMS then does a summation of the available energy and presents it to the Vehicle.

Whenever Slave BMS suspends due to some reasons, Master BMS will receive this and present it to VCU as suspension. Whenever Vehicle suspends it would broadcast this message to indicate to all the BMS. This is required for the slave BMS to stop sending the available energy data to Master BMS for summation

In each stage, if the vehicle or BMS does not receive message from the other party or does not receive correct message within the stipulated time limit, the waiting entity will timeout (timeout means failure to receive a complete data package or correct data package within specified time); unless otherwise specified, the timeout is all 5s. After timeout, vehicle or BMS will send suspending message.

4.4.1. Suspending /Alert Types

The suspending/alert messages are three types:

- Normal Suspension – Whenever vehicle stops / ignition is off
- Warning /Alerts – These alerts are intended to warn the vehicle/user of impending potential problems that could automatically stop the vehicle. It is advisable for the user to stop the vehicle so that the warning condition does not become an error. For e.g., if there is a battery over-temperature warning, stopping the vehicle (Normal Suspension) and waiting for some-time for the temperature to come down will prevent an abrupt stopping
- Error Suspension – The vehicle will stop after error and will not start until the error gets sorted out. Some errors like BIN-VIN mismatch are non-recoverable, while others like battery or component over-temperature can be overcome with some idle time

  Figure below presents the overall driving process.

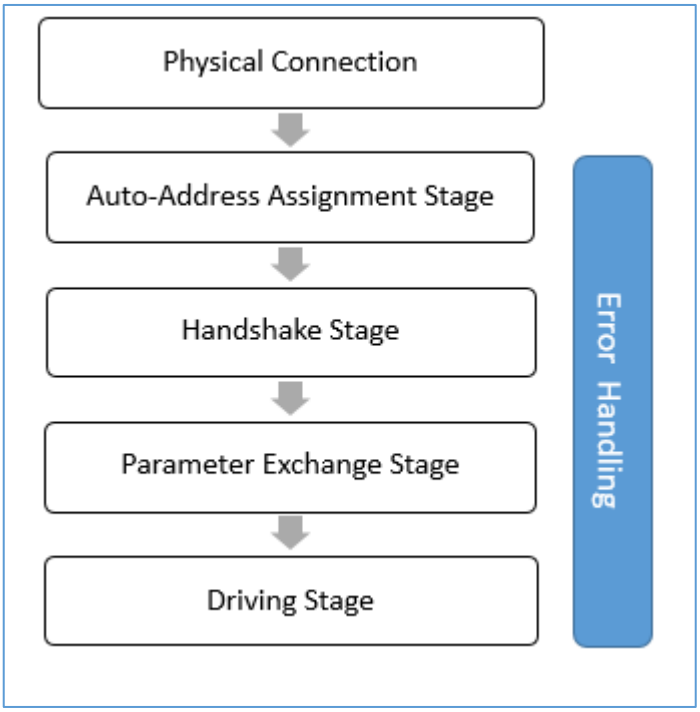


Figure: Driving Protocol Stages

4.5. Driving Protocol Stages

In driving protocol, VCU will communicate only with the master battery module in fetching the required information. Wherever required, the master module will get data from slave modules, do some calculation and present the data to the VCU. Similar condition is applicable for Error conditions also.

If there is an error in slave module, it will be communicated to Master module and master will suspend by stating the error code. If master BMS fails due to some condition, VCU will detect this through the battery mode re-iteration message from batteries and assign the next least addressed BMS as Master BMS and whenever Master BMS recovers, it would confirm its role as Master BMS with VCU and based on the response, it would start acting accordingly. Protocol has provision for extra message to handle this.
If the number of battery module count is less than the expected, then vehicle will indicate this as warning to the driver may be through dashboard

In the parameters table in each stage, the order of SPNs in the CAN communication is as per the order specified in the table itself. For any suspension during the protocol flow, the state machine will reset and start from the beginning of the protocol.

Driving Protocol version: 0.9.0.

4.5.1. Auto address assignment stage

VCU assigns addresses for all BMSes. VCU must reserve the address in the range of 0x90 -0x94 for BMS addressing irrespective of the number of batteries to be placed in the vehicle. VCU's source address (SA) is defined as 129 (81H).

Whenever Vehicle is started, vehicle broadcasts vehicle start message to BMS. BMS checks if its address is assigned, if yes then BMS confirms its role (Master/Slave) with vehicle and proceeds with handshake stage. If address is not assigned then BMS requests for an assignment

- **Message1: Indication of vehicle started from VCU**

VCU broadcasts vehicle started message to BMS. BMS checks for address assignment and if not sends Message 2 requesting for address claim else sends BMS will send a role (Master/Slave) confirmation request to VCU.

SA	DA	DATA Byte:01	DATA Byte:02	DATA Byte:03	DATA Byte:04	DATA Byte:05	DATA Byte:06	DATA Byte:07	DATA Byte:08
81 (VCU Address)	FF (Global Address)	AA				00	00	00	00

- **Message2: Request for address claim from BMS**

BMS will send a request for address claim by generating a random number (RN1- say 2E2614D0) of 4 bytes in the data field. Bytes 1 to 4 will be used for this. Unused bytes in the data field will be filled with 0x00.

SA	DA	DATA Byte:01-04	DATA Byte:05	DATA Byte:06	DATA Byte:07	DATA Byte:08
FE (Null Address)	81 (VCU Address)	2E2614D0	00	00	00	00

- **Message3: Broadcast response for address claim from VCU**

VCU broadcasts to CAN bus with the same random number (RN1- 2E2614D0) and allotted address (say 0x90). The allotted address will be available in the 5th byte of data field.

SA	DA	DATA Byte:01-04	DATA Byte:05	DATA Byte:06	DATA Byte:07	DATA Byte:08
81 (VCU Address)	FF (Broadcast Address)	2E2614D0	90	00	00	00

- **Message4: BMS confirmation request for allotted address from BMS**

BMS requests VCU to confirm the usage of the allotted address by generating and sending another random number (RN2 – say 33AB7F30) and allotted address (0x90) to VCU.

SA	DA	DATA Byte:01-04	DATA Byte:05	DATA Byte:06	DATA Byte:07	DATA Byte:08
FE (Null Address)	81 (VCU Address)	33AB7F30	90	00	00	00

- **Message5: VCU confirmation response for allotted address from VCU**

VCU broadcasts to CAN bus with the random number (RN2), allotted address (0x90) and address status (0xAA: Success; 0xFF: Failure). The address status will be available in the 6th byte of data field. If the status from VCU is a failure i.e. the address being allotted to some other BMS, then BMS must repeat and start from Message1.

When VCU broadcasts with random number, allotted address and address status in the CAN bus. Each BMS should keep a track on this message to identify the number of batteries present.

SA	DA	DATA Byte:01-04	DATA Byte:05	DATA Byte:06	DATA Byte:07	DATA Byte:08
81 (VCU Address)	FF (Broadcast Address)	33AB7F30	90	AA	00	00

- **Message 6: BMS confirmation on allotted address**

BMS confirms to VCU on the allotted address by sending the random number 2, allotted address and the confirmation status (0xAA: Success; 0XFF: Failure)

SA	DA	DATA Byte:01-04	DATA Byte:05	DATA Byte:06	DATA Byte:07	DATA Byte:08
FE (Null Address)	81 (VCU Address)	33AB7F30	90	AA	00	00

The failure status would occur, if BMS is unable to assign the allotted address for some reasons. In this case, the BMS must repeat and start from Message2 to get an address assigned.

When VCU sees a success status, it ensures that this address is not given to any other BMS

- **Message 7: VCU assigning master BMS**

VCU will assign BMS with least address as Master BMS. For e.g. it the batteries gets 0x90, 0x91 and 0x92 as their assigned addresses, then VCU assigns BMS with 0x90 as Master BMS. The other BMS will act as Slaves.

For Master BMS

SA	DA	DATA Byte:01	DATA Byte:02	DATA Byte:03	DATA Byte:04	DATA Byte:05	DATA Byte:06	DATA Byte:07	DATA Byte:08
81 (VCU Address)	90 (Master BMS)	01 (Master)	90 (Self Address)	90 (Master Address)	00	00	00	00	00

For Slave BMS1

SA	DA	DATA Byte:01	DATA Byte:02	DATA Byte:03	DATA Byte:04	DATA Byte:05	DATA Byte:06	DATA Byte:07	DATA Byte:08
81 (VCU Address)	91 (Slave BMS1)	00 (Slave)	91 (Self Address)	90 (Master Address)	00	00	00	00	00

For Slave BMS2

SA	DA	DATA Byte:01	DATA Byte:02	DATA Byte:03	DATA Byte:04	DATA Byte:05	DATA Byte:06	DATA Byte:07	DATA Byte:08
81 (VCU Address)	92 (Slave BMS1)	00 (Slave)	92 (Self Address)	90 (Master Address)	00	00	00	00	00

The address assignment for other BMS will happen in parallel in the same manner. In order to reduce the probability of multiple trials, each BMS should start with message 2 above after a random delay for a duration between 50 to 200 ms. The following use cases are defined to elaborate on address assigning procedure under few conditions:

Case 1: Probability of VCU receiving the same random number from two BMS (Message2)

- VCU will broadcast RN1+allotted address as defined in Message3.
- Both the BMS will receive this response and will send a confirmation request for allotted address with random number 2.
- VCU will receive the response and will send (Message5 with 0xAA in byte 06) success to first BMS and failure (Message5 with 0x00 in byte 06) to other BMS.
- BMS receiving success status will continue with Message6 onwards and BMS receiving failure status will start from Message2
- Probability of occurrence of random number 2 being same for two BMS is very less and assumed to not occur.

Case 2: Failure in Master BMS

Assuming there is a failure in Master BMS, VCU will time-out and suspend the drive. When the vehicle is again started, Master BMS is down without any communication, each BMS will confirm its role with VCU by sending a battery mode re-iteration message.

VCU will detect only two batteries are present and issue a warning to the driver and will continue with assigning the Slave with address 0x91 as Master and start with the protocol.

When master BMS had failed, VCU assigns SLAVE BMS 1 as Master:

SA	DA	DATA Byte:01	DATA Byte:02	DATA Byte:03	DATA Byte:04	DATA Byte:05	DATA Byte:06	DATA Byte:07	DATA Byte:08
81 (VCU Address)	91 (Slave BMS1)	01 (Master)	91 (Self Address)	91 (Master Address)		00	00	00	00

The same message will be issued to Slave BMS2 also.

During the run of the vehicle, when Master resumes from its failure mode, it will initiate the battery mode re-iteration message to VCU to check whether it is Master and VCU confirms its status by sending Message 6. Through Message 6, Master BMS will realize that it is not currently master and will start acting as slave. When Master BMS resumes and requests to confirms its status, VCU would send

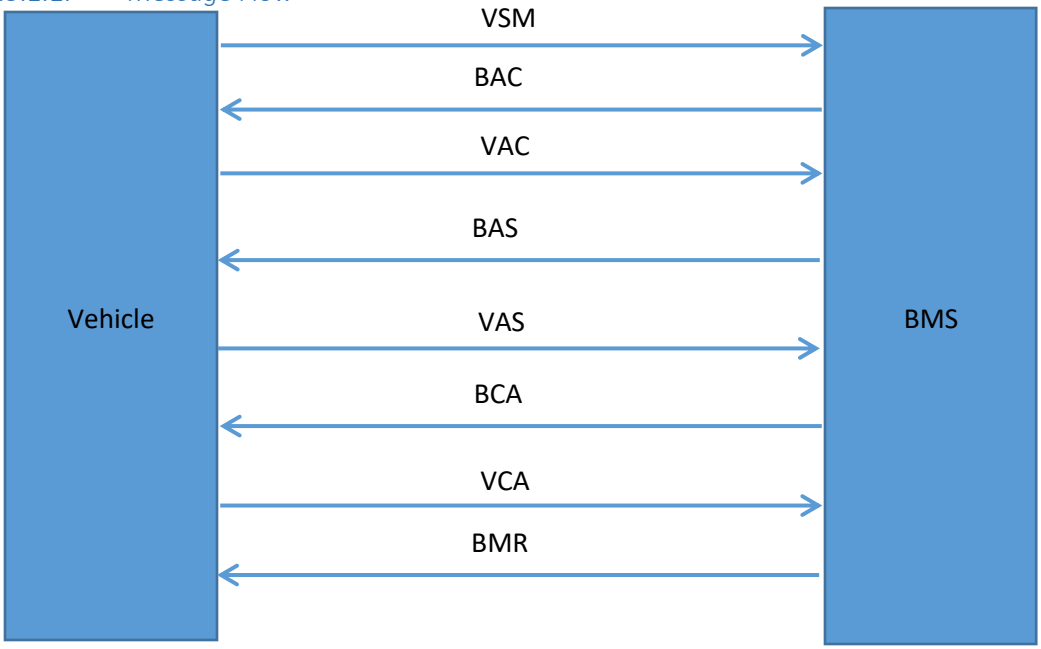
SA	DA	DATA Byte:01	DATA Byte:02	DATA Byte:03	DATA Byte:04	DATA Byte:05	DATA Byte:06	DATA Byte:07	DATA Byte:08
81 (VCU Address)	90 (Master BMS)	00 (Slave)	90 (Self Address)	91 (Master Address)	00	00	00	00	00

Case 4: When BMS resumes after a failure

When any BMS fails due to some reason, then immediately after recovery, it must check if its address is assigned and if not, it would claim by using address assignment procedure. Next it would check whether it was acting as Master BMS, if yes then it would send BMR message requesting the status of its role as Master BMS. Master BMS will send Message 6 to confirm whether it must act as Master or slave.

Refer Annexure C1 for auto-address assignment flow

4.5.1.1. Message Flow



4.5.1.2. Messages

Message Code	Messages	Direction	PGN (HEX)	Priority	Size in Bytes	Time Period (ms)
VSM	Vehicle start message	Vehicle to BMS	000F00H	4	1	250
BAC	Battery address claim request message	BMS to Vehicle	001200H	4	4	250
VAC	Vehicle broadcast response for address claim message	Vehicle to BMS	005700H	4	5	250
BAS	BMS confirmation address request Message	BMS to Vehicle	005800H	4	5	250
VAS*	VCU confirmation address response	Vehicle to BMS	005900H	4	6	250
BCA	BMS confirmation request on allotted address	BMS to Vehicle	005A00H	4	6	250
VCA	Vehicle Assigning Master BMS Message	Vehicle to BMS	001300H	4	3	250
BMR	BMS mode re-iteration Message	BMS to Vehicle	001400H	4	1	250

*VAS message shall be used by BMS to identify the number of batteries present in the vehicle. VCU broadcasts VAS with random number, allotted address and address status in the CAN bus. Each BMS should keep track of this message and the unique allotted address to decipher the number of batteries present.

BMS Mode Re-Iteration Message: This message will be used in two scenarios:

Scenario 1: Start of vehicle:

Whenever vehicle is started, it will send VSM message to indicate the start of vehicle to BMS. BMS checks if its address is assigned and if not, it would claim by using address assignment procedure by sending "BAC" message. If address is assigned, then BMS sends BMR message to check if it is Master or Slave. VCU will give the confirmation through VCA message. This would be repeated for each BMS in the vehicle

Scenario 2: Master BMS failure

Due to some reasons, when Master BMS fails to provide the required details to VCU, VCU will send Time-out message and suspend the drive.

As driver won't be aware of this, when he starts the vehicle again, VCU will receive BMR only from other BMS. As this count will be less than the expected count based on number of batteries in vehicle, vehicle will indicate this as warning to the driver.

As it had received only two requests from the BMS, by checking the source address, VCU identifies that Master BMS had failed and assigns next Slave BMS (say 91H) as Master and continues with the protocol.

The same steps are applicable when any BMS fails.

4.5.1.3. Parameters

Message Code	Parameter Name	Description	SPN (Dec)	Size in Bytes	Delivery Option
VSM	Vehicle start; Value=0xAA	Vehicle start parameter to BMS to indicate the ignition of vehicle	370	1	Mandatory
BAC	Random Number 1	Random number 1 generated by BMS	356	4	Mandatory
VAC	Random Number 1	Random number received in BAC	260	4	Mandatory
	Allotted address	Address allotted by VCU	261	1	Mandatory
BAS	Random Number 2	Random number 2 generated by BMS	262	4	Mandatory
	Allotted address	Allotted address in VAC	263	1	Mandatory
VAS	Random number 2	Random number 2 received in BAS message	357	4	Mandatory
	Allotted address	Allotted address by VCU in VAC message	358	1	Mandatory
	Allotted status Success=0xAA; Failure=0xFF	Allotted address status. VCU confirming the address status as success or failure	359	1	Mandatory
BCA	Random Number 2	Random number 2 generated in BAS message	360	4	Mandatory
	Allotted address	Allotted address in VAC messages	361	1	Mandatory
	Acceptance status Success=0xAA; Failure=0xFF;	BMS acceptance status for allotted address	362	1	Mandatory

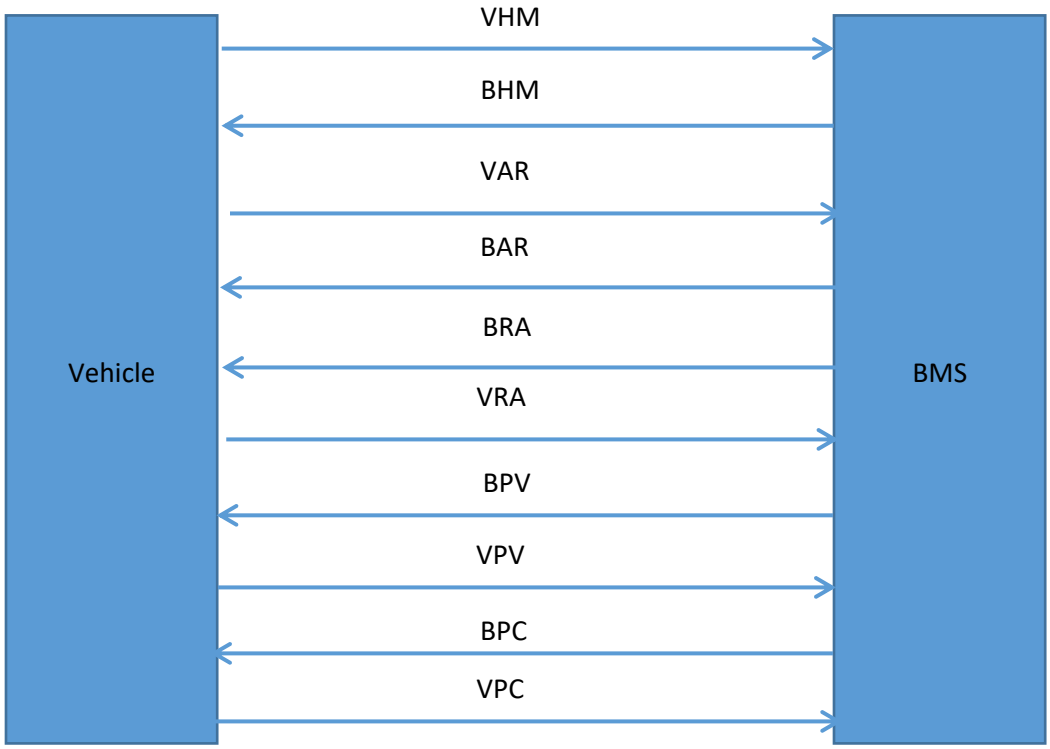
VCA	Master BMS Assignment with Master: 0xAA; Slave :0xFF;	VCA assigning the BMS with least address as Master BMS. By default, it would be 90H and incase of Master BMS failure, the next least address would be 91H	363	1	Mandatory
BMR	MBR assignment request, Value=0xAA;	BMS requesting Master status	364	1	Mandatory

4.5.2. Handshake Stage

In this stage, the BIN-VIN matching check is done to make sure that the VIN is programmed in Swapping outlet to work with this IBN is indeed the VIN in which the BMS is placed. This is followed by battery authenticity check by VCU and driving protocol version – compatibility check. Annexure A2 provides details on the protocol version matching.

In battery authenticity check, a random number is generated by VCU and sent to BMS. BMS runs an algorithm and sends the result to VCU. VCU runs the same algorithm and based on the result, it authenticates the battery. This battery authentication must be done with each battery module independently.

4.5.2.1. Message flow



4.5.2.2. Messages

Message Code	Messages	Direction	PGN (HEX)	Priority	Size in Bytes	Time Period (ms)
VHM	Vehicle Handshake Message(VHM)	Vehicle to BMS	005B00H	6	20	250
BHM	Battery Handshake Message(BHM)	BMS to Vehicle	005C00H	6	24	250
VAR	Authenticity Request Message initiated by Vehicle	Vehicle to BMS	005D00H	6	4	250
BAR	Authenticity Response Message by battery for vehicle initiated request	BMS to Vehicle	005E00H	6	4	250
BRA	Authenticity Request Message initiated by Battery	BMS to Vehicle	000E00H	6	4	250
VRA	Authenticity Response Message by Vehicle for battery initiated request	Vehicle to BMS	000F00H	6	4	250
BPV	BMS protocol version Message	BMS to Vehicle	005F00H	6	3	250
VPV	Vehicle Protocol Version Message	Vehicle to BMS	000600H	6	3	250
BPC	Battery Protocol Version Confirmation Message	BMS to Vehicle	006100H	6	3	250
VPC	Vehicle Protocol Version Acknowledgment Message	Vehicle to BMS	006200H	6	1	250

4.5.2.3. Parameters

Message Code	Parameter Name	Description	SPN in Dec	Size in Bytes	Delivery Option

VHM	VIN	VIN - Vehicle Identification Number	264	17	Mandatory
	VCU firmware version	Firmware version of VCU	336	3	Mandatory
BHM	BIN	BIN	265	20	Mandatory
	BMS firmware version	Firmware version of BMS	337	3	Mandatory
	VIN Acknowledgment Result	Acknowledgment result by BMS on receiving VIN. This step is to confirm that the battery modules are configured for the same VIN during swapping procedure. Success = 0xAA; Failure=0XFF	266	1	Mandatory
VAR*	Vehicle Random Number Challenge	Authenticity request number from vehicle check between Vehicle and BMS	267	4	Mandatory
BAR*	Battery Random Number Response	Authenticity response number from battery	268	4	Mandatory
BRA**	Battery random number request	Authenticity request number from battery	310	4	Mandatory
VRA**	Vehicle random number response	Authenticity response number from vehicle	311	4	Mandatory
BPV	BMS driving protocol version	Driving protocol version number of BMS e.g. 00H01H00H for version 1.0	269	3	Mandatory
VPV	Vehicle driving protocol version	Driving protocol version number of the vehicle	270	3	Mandatory
BPC	Confirmed version of BMS communication protocol	Based on vehicle's protocol version, BMS will confirm the version number	271	3	Mandatory
VPC	Protocol version acknowledgement	Acknowledgement on protocol version from BMS. Success = 0xAA; Failure=0XFF	272	1	Mandatory

Note: Refer Annexure A for protocol version matching scenarios

*Authenticity Check initiated by vehicle:

Random number (RN1) is generated by vehicle and sent to BMS. BMS runs an algorithm and sends the result to vehicle (Result). Vehicle runs the same algorithm and based on the result, it authenticates the battery.

For internal testing purpose, Result = RN1 / 2 could be assumed with some random number as RN1 and before final delivery, the proprietary algorithm provided by Energy Business Company has to be implemented.

**Authenticity Check initiated by Battery:

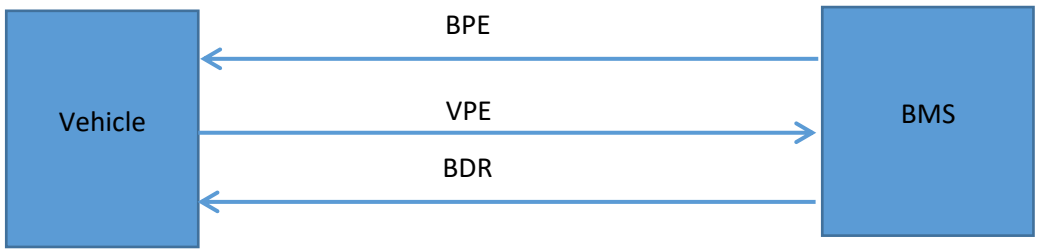
Random number (RN2) is generated by BMS and sent to Vehicle. Vehicle runs an algorithm and sends the result to BMS (Result). BMS runs the same algorithm and based on the result, it authenticates the Vehicle.

For internal testing purpose, Result = RN 2 / 2 could be assumed with some random number as RN2 . The proprietary algorithm will be provided by Energy Business company for both the stage: Authenticity check initiated by vehicle as well as from Battery as binaries. BMS and Vehicle OEMs would have to develop their firmware using the same.

4.5.3. Parameter Exchange Stage

During this stage, the parameters required for driving are exchanged between BMS and vehicle. The available energy will be provided by Master BMS. Master BMS will do all mathematical calculation and provide the required detail to vehicle

4.5.3.1. Message Flow



4.5.3.2. Messages

Message Code	Messages	Direction	PGN (HEX)	Priority	Size in Bytes	Time Period (ms)
BPE*	Battery Parameter Exchange Message	Master BMS to VCU and Slave BMS to Master BMS	006300H	6	2	250
VPE	Vehicle Parameter Exchange Message	Vehicle to BMS	006400H	6	6	250
BDR	Battery discharge ready message	BMS to Vehicle	001500H	6	1	250

* During parameter exchange stage, once a BMS is assigned as slave, it would keep sending the SoC*SoH value to Master BMS once. Master BMS will do the summation of all the SoC*SoH and present it as "available energy" to the vehicle.

4.5.3.3. Parameters

Message Code	Parameter Name	Description	SPN in Dec	Size in Bytes	Delivery Option
BPE	Available energy	Summation of (SOC*SOH) of battery modules is sent by the master module to the VCU. Slave battery modules should send their (SoC*SoH) to master module	273	2	Mandatory
VPE	Effective resistance of power path	Effective resistance of power path in milli-ohm	274	2	Mandatory
	Start meter reading	Odometer reading at the start of the vehicle in Km	275	4	Mandatory
BDR	Battery discharge ready signal. Value-0xAA	Master BMS giving the discharge ready signal to vehicle	365	1	Mandatory

4.5.4. Driving Stage

Throughout the driving stage, the BMS and VCU communicates periodically over the time period defined for respective messages. The BMS must send the available energy by maintaining some level of threshold to ensure that battery doesn't get drained to the end. This threshold value can be a proprietary parameter of battery manufacturer.

4.5.4.1. Message Flow

4.5.4.2. Messages

Message Code	Messages	Direction	PGN (HEX)	Priority	Size in Bytes	Time Period (ms)

BEV*	Battery Status Message	Master BMS to Vehicle & Slave BMS to Master BMS	006500H	4	8	500
VDS	Vehicle Status Message	Vehicle to BMS	006600H	4	4	1000
VPM	Vehicle Parameters Message	Vehicle to BMS	001600H	4	6	1000
BMS	Battery Master to Slave Message	Master BMS to Slave BMS	001900H	4	2	1000
VOC	Vehicle Open command 1 Data for logging	Vehicle to BMS	003900H	4	4	10000
VDL	Vehicle Open command 2 for data logging	Vehicle to BMS	003A00H	4	4	60000
BST	Battery Suspending /Alert Message	Master BMS to Vehicle & Slave BMS to Master BMS	006700H	2	10	500
VST	Vehicle Suspending / Alert Message	Vehicle to BMS	006800H	2	10	500

* During driving stage, Slave BMS should keep sending the SoC*SoH value to Master BMS every 50 ms. Master BMS will do the summation of all the SoC*SoH and present it as “available energy” to the vehicle.

4.5.4.3. Parameters

Message Code	Parameter Name	Description	SPN in Dec	Size in Bytes	Delivery Option
BEV	Battery maximum permissible current	Calculated by the master and sent to VCU by taking minimum value of the maximum current of individual modules multiplied by number of modules in Centi Ampere.	276	2	Mandatory
	Maximum permissible regeneration current	Calculated by the master and sent to VCU by taking minimum value of the maximum current of individual modules multiplied by number of modules in Centi Ampere	277	2	Mandatory
	Available energy	Master will do the summation of (SOC*SOH) of battery modules. Each module will send (SoC*SoH)	278	2	Mandatory

	Battery ambient temperature	Battery ambient temperature in 0.1°C.	353	2	Mandatory
VDS	Vehicle controller Voltage	Vehicle controller voltage in Centi volts	279	2	Mandatory
	Vehicle controller current	Vehicle controller current in centi-amperes	354	2	Mandatory
VPM	Vehicle Speed	Vehicle speed in Kmph	280	2	Mandatory
	Odometer reading	Odometer reading	355	4	Mandatory
BMS*	Battery instantaneous net current in Centi-Amp	Current sensed by master subtracted by current sensed by vehicle	366	2	Mandatory
VOC	OC1	Open Command 1	366	2	Optional
	OC2	Open Command 2	367	2	Optional
VDL	OC3	Open command 3	368	2	Optional
	OC4	Open command 4	369	2	Optional
BST	BMS Stops Discharging (BST)	BMS stops discharging with detailing normal, fault or error cause.	281	2	Mandatory
	BST Threshold Value	Parameter to carry the data related to Battery suspension reason with threshold value	282	4	Mandatory
	BST Breach Value	Parameter to carry the data related to Battery suspension reason with breach value	283	4	Mandatory
VST	Vehicle Stops (VST)	Vehicle stops with detailing normal, warning or error cause. This would be broadcasted over CAN bus to indicate that the vehicle is stopping	284	2	Mandatory
	VST Threshold Value	Parameter to carry the data related to vehicle suspension reason with threshold value	285	4	Mandatory
	VST Breach Value	Parameter to carry the data related to vehicle suspension reason with breach value	286	4	Mandatory

*BMS Message: Master BMS broadcasts the battery instantaneous net current (Current sensed by master subtracted by current sensed by vehicle) and all Slave BMSes can switch-off/disc-connect in case of unauthorized charging or disc-charging.

4.5.4.4. Battery Suspending/ Alert Reasons (BST)

With two bytes, the error values could be from 1-65534 (0x1 – 0x FFFF). While error codes 0x1 – 0x7FFF are reserved for protocol specific, 0x8000 – 0xFFFF are available for manufacturer specific code. The table below summarizes the suspending / alert code range and allocated ranges.

#	Suspending /Alert code specific to	Range	
1	Overall range	0x0001-0xFFFF	
2	Protocol specific	0x001 – 0x7FFF	Battery Specific: 0x0001- 0x3FFF
			Vehicle Specific: 0x4000 – 0x7FFF
3	Manufacturer specific	0x8000 – 0xFFFF	Battery Specific:0x8000 – 0xBFFF Charger Specific:0xC000 - 0xFFFF

While suspending, BMS will give error code and threshold with breach value for that particular error code, if available. If there is no appropriate values could be sent on threshold and breach values, it has to be filled as FFH.

Suspension Type	**Reason**	**Hexa-decimal Code**	**BST Threshold Value**	**BST Breach Value**
Error Suspension	BIN-VIN mis-match. Note: During this suspension battery has to store the mis-match VIN as additional value as type BT9, which would be retrieved in the charging protocol	0001H	FFH	FFH
	Discharging current is excess	0002H	Threshold discharging current	Discharging current above threshold
	Battery cell over temperature	0004H	Cell number + Threshold cell temp.	Cell number + Excess cell temp.
	Battery connector health check. This condition is generated when the connector effective resistance is greater than 25mohm	0005H	Threshold value	Excess value
	Vehicle authenticity check failure	0006H	BMS Random number challenge	VCU Random number

				challenge response
Warning /Alerts	BMS component over temperature	1001H	Threshold component temp.	Excess component temp.
	Battery cell over temperature	1002H	Threshold cell temp.	Excess cell temp.
	Battery over temperature	1003H	Threshold battery temp.	Excess battery temp.
Other Suspension	Other suspension	3FFEH	Threshold Value(if available)	Breach value(if available)

* BMS will keep monitoring the connector temperature towards a threshold value and suspend when it exceeds this value.

4.5.4.5. Vehicle Suspending Reasons (VST)

The suspending/alert code for vehicle will be in the range of 0x4000-0x7FFF

While suspending, vehicle will give error/alert code and threshold with breach value for that particular code, if available. If there are no appropriate values could be sent on threshold and breach values, it must be filled as FFFFH.

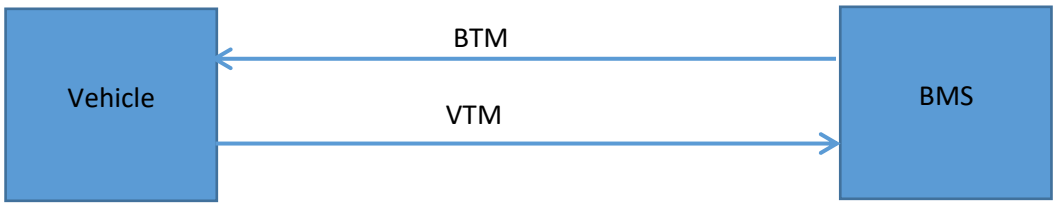
Suspension Type	**Reason**	**Hexa-decimal Code**	VST Threshold Value	VST Breach Value
Normal Suspension	Vehicle stopped by user	4001H	FFFFH	FFFFH
Error Suspension	Battery Authenticity Failure	4002H	Vehicle random number challenge	BMS random number challenge response
	Protocol version acknowledgment error	4003H	Protocol version	Protocol version
	Available energy of battery less than threshold value set by vehicle	4004H	Threshold energy	Available energy
Warning Suspension	Internal temperature of Vehicle controller is excessive	4005H	Threshold internal temp.	Excess internal temp.

	Vehicle component temperature is excess	4006H	Threshold component temp.	Excess component temp.
Other Suspension	Other suspension	7FFFH	Threshold value (If available)	Breach value (if available)

4.5.5. Time-out Messages

The time-out of messages during the whole driving process is communicated using these messages. This message will carry the second byte of PGN to indicate the timed-out message.

4.5.5.1. Message Flow



4.5.5.2. Messages

Message Code	Messages	Direction	PGN (HEX)	Priority	Size in Bytes	Time Period (ms)
BTM	Battery time-out Message	BMS to VCU	006900H	2	1	250
VTM	Vehicle Time-out Message	Vehicle to BMS	006A00H	2	1	250

4.5.5.3. Parameters

Message Code	Parameter Name	Description	SPN (Dec)	Size in Bytes	Delivery Option
BTM	Second Byte of PGN Value of timed-out message	Battery time-out messages	287	1	Mandatory
VTM	Second Byte of PGN Value of timed-out message	Vehicle time-out messages	288	1	Mandatory

5. Data Logging Retrieval Logic

The data is logged during charging and driving protocol. These data are retrieved during charging protocol in the corresponding stages. The data storage formats are provided in sections 3.5.4.1 and 3.5.7.1 for driving and charging log parameters respectively. This section details on the retrieval logic of the stored data.

- The transfer of periodic and additional data is done by the following steps: In the first message, charger initiates a request to BMS asking for the list of available data types, its size and granularity. The response is provided in Type-Size-Granularity (TSG) format with a packet size of 5 bytes. (Type-1 Byte, Size-2Bytes, Granularity-2Bytes). The end of the list is indicated by a "0" value in the TSG.
- After fetching all the list, charger initiates the transfer of type 1 by using Timestamp-Type (TT) packet format with a size of 3 bytes.
- By default, the first time-stamp should be 01/01/2017 00:00:00. The format of timestamp is as defined in Section: Timestamp format for periodic data.
- BMS parses to find the first data available for this type after the default timestamp. BMS responds with Timestamp-Type-Data (TTD) packet of size 6 bytes. This timestamp will be charger to initiate the request for the next packet of same type.
- Once all the data transfer is completed for type 1, charger repeats the same steps for other types.
- The data transfer stage is complete only after all types of data are transferred.
- The termination of TSG and TTD should be '0'. This is to indicate the end of list and no additional list/data is present.

5.1. TSG Packet Format – Response packet with data list

For fetching the data storage from BMS, a Type-Size-Granularity (TSG) packet is defined to fetch the list of data types available. For event based data, the granularity should be '0' and for periodic data, the granularity will carry the periodicity of the data. The TSG packet size is of 5 bytes

The type is of 1-byte length. Type 1 – Type 100 is reserved for protocol specific and Type 101 - Type 255 is for manufacturer specific. Size is of 2 bytes length.

Granularity is of 2 bytes and captures the periodicity of data log in seconds. This is shown in table below

The TSG packet format is defined as follows:

Type Number	Size in Bytes	Granularity in Secs.
1 byte	2 bytes	2 bytes

Example event list packet:

Type	Size (Bytes)	Granularity(Secs)
Available energy	2	0

Example periodic list packet:

Type	Size (Bytes)	Granularity(Secs)
1	2	10

5.2. TT Packet Format – Request packet for data

Request packet from charger for retrieving the data will follow Timestamp-Type (TT) packet format The TT packet format is defined as follows:

Timestamp	Type n
4 bytes	1 byte

The TT packet size is 5 Bytes.

Example Packets: Example additional data packet of vehicle

Timestamp	Type n
16022909H	1(for OC1)

5.3. TTD Packet Format – Response packet with data

Response packet from BMS for retrieving the data will follow the Timestamp-Type-Data (TTSGD) packet format. For event based data the granularity should be '0' and for periodic data, the granularity will carry the periodicity of the data. The TTD packet format is defined as follows:

Timestamp	Type n	Data
4 bytes	1 byte	2 Bytes

The TTD packet size would 9 Bytes plus the actual data size. The whole packet will be zero, if no data is present.

Example Packets:

Example additional data packet of vehicle

Timestamp	Type n	Size (Bytes)	Granularity(Secs)	Data
16022909H	1(for OC1)	2	10	500

Example additional data packet of battery for event-based data

Timestamp	Type n	Size (Bytes)	Granularity(Secs)	Data
16022906H	Available energy	2	0	1500

Example additional data packet of battery for periodicity-based data:

A sample packet is shown below for sending a data of voltage and current of each 2 bytes for a granularity of 10 Sec.

Timestamp	Type n	Size (Bytes)	Granularity (Secs)	Voltage (value in Deci volts)	Current (value in Deci Amps)
16022909H	T101	4	10	480	200

5.4. Timestamp Format for Periodic Data

Time Stamp Packet	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	1	0
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Time stamp	Day					Month				Year						Hour					Minutes						Seconds					

The above timestamp to be interpreted as: 02/12/2017 03:04:06. For the year component value of 1 is considered as 2017,2 as 2018 and so on.

6. BMS – HHD – CMS Protocol

A Handheld device will be used in the Swapping outlet for handling the swapping procedure. The swapping procedure will include removing of discharged batteries, placing charged batteries and handling the billing for the swapping. The BMS and the HHD app. Communicate using Bluetooth Low Energy(BLE)

6.1. HHD Functionalities

When a vehicle comes to the swapping outlet, the following sequence of operations occur.

1. The batteries would be removed from the Vehicle.
2. The MAC Id of the battery, which will be available as barcode / QR code on the battery, is scanned by the app.
3. The app connects to the scanned MAC Id through BLE (Bluetooth Low Energy) and then reads the available energy and the VIN (vehicle identification number) in which the battery was serving.
4. The mode of the battery is set to charging from driving. The BMS must change its address to 0xFF
5. The above three steps are repeated for all the batteries in the vehicle.
6. The charged batteries' BIN are scanned by the app as in Step 2.
7. The VIN of the vehicle is programmed in to the battery.
8. The above two steps are repeated for all the batteries that are to be placed in the vehicle.
9. The charged batteries' BIN and the VIN are updated to the CMS.
10. Then the individual available energies of the battery (step 3) and the VIN are sent to CMS to get the billing amount to be collected from the driver.
11. This amount will be collected from the driver using UPI-based payment.

6.2. BLE Specifications

This document provides the different characteristics available in the EV – Battery service in a device. If a device claims conformance to this service, all capabilities indicated as mandatory for this service shall be supported.

6.2.1. Service Dependency

This service has no dependencies on other GATT-based services

6.2.2. Byte Transmission Order

All Characteristics used with this service shall be transmitted with the least significant octet first (i.e. little endian)

6.2.3. Service Declaration

The service UUID shall be set as per the standards. Custom services, characteristics and descriptors preferably should have 128-bit UUID.

6.2.4. Service Characteristics

EV-Battery Service- UUID		**a377860c-0594-4377-9843-ede2281d3cbc**							
Characteristics Of above EV-Battery Service									
#	**Characteristic**	**Description**	**UUID**	**Mandatory / Optional**	**Read**	**Write**	**Notify**	**Indicate**	**Remarks**
1	Energy Available	Available energy in battery	b6060cf1-e288-4d85-855f-77162e8a4a47	M	Y	X	X	X	
2	BIN	Battery Identification Number	7837a8a3-936d-4009-b666-f2e9033e4ac6	M	Y	X	X	X	
3	VIN	Vehicle Identification Number	70d6e5bd-1289-4ff1-9eca-b20922eeb906	M	Y	X	X	X	
4	Lock-smart Mode	Smart-lock mode : Drive Mode/Charge mode	323c818f-28f0-4c5e-8cf8-d112a52ae7e1	M	Y	X	X	X	

5	Error Codes	Error Codes stored in battery	ef7f78b2-6835-423f-9b5c-71e340cc68cf	O	Y	X	X	X	
6	EV- Battery statistics	Packet to read the statistics from the battery	0cc59d16-be17-408a-bcb1-ed81ebca6f85	M	Y	X	X	X	Energy Available BIN VIN Smart-lock Mode Unique-id for drive (UFD)*
7	EV-Charged Battery VIN Association	Packet to program charged battery	dfa75d76-4078-47d3-9a03-b73495c52cb2	M	X	Y	X	X	Smart-lock Mode VIN Unique-id For Drive (UFD)*

*UFD- Unique-Id for Drive obtained from CMS

7. BMS Functionality Guidelines

1. Based on the lock-smart mode: Drive mode/Charge mode, battery should decide on the protocol to start with.
2. All the batteries should keep storing ambient temperature every 15 mins along with the timestamp and lock-smart mode. This will be recorded continuously by the battery in all the modes either driving or charging. The same will be retrieved during the charging protocol.
3. Master-Slave Communication in driving protocol:
    - Slave BMS will keep updating the available energy to Master BMS once during parameter exchange stage and for every 500 ms during driving stage.
    - When slave BMS is suspending it should send BST (Suspending message) to Master BMS with the reason for suspension and then suspend. Master BMS will get updated about the slave status and keep running the vehicle if the available energy is sufficient to run the vehicle
    - When the vehicle stops, all the batteries will receive VST from the vehicle and the BMS should to shut-off delivering energy.
4. During driving, if a battery fails due to some reason, then based on the role assigned by VCU, it should start communicating. For e.g. if a slave is assigned as Master then it should start representing the available energy to VCU and when a master becomes slave, it should keep updating the available energy to Slave BMS who is acting as Master now.
5. All vehicle related data are logged only by Master BMS to avoid duplicate logging in slave BMS. So, when a Master BMS which has logged vehicle data turns out to be slave due to some failure

and recovery, it should remember or store the battery mode as Master to enable the charging protocol to retrieve the vehicle log.

6. All the data logs to be cleared only during charging protocol at the end of the corresponding transfer stages

8. Charger and OMS communication

8.1. Abbreviation

OMS	Operation Management System
CIMS	Charging Infrastructure Management System
BMS	Battery Management System
HTTP	Hyper Text Transfer Protocol
TCP	Transmission Control Protocol
HHD	Hand Held(device)
CSV	Comma Separated Values
PDF	Portable Document Format
XML	extended Markup Language
GUI	Graphical User Interface
IP	Internet Protocol

Charger and OMS would communicate over a wired-Ethernet IP LAN. The OMS and Charger will have fixed IPs. The application/ /transport layer protocol would be HTTP 1.1/TCP using JSON.

8.2. Message Types

#	Message Type	Description
1	Monitoring and Management	Routing management messages
2	Diagnostics	Health check and diagnostic messages
3	Events and Alerts	Asynchronous events and alerts from Charger
4	Charging	Messages related to Charging
5	Discharging	Messages related to data transfer for Discharging
6	Software Management	Software update related messages
7	Discovery	To identify the chargers and Connectors
8	Heart Beat	Periodic heart beat messages from Charger to OMS

8.3. Messages Initiated by OMS

#	Msg. Types	Message name	Msg. Origin	Purpose
1	Discovery	GetChargerBasicConfig	OMS	To discover basic properties of the charger
2	Discovery	GetChargerAdvancedConfig	OMS	To discover detailed properties of the charger
3	Monitoring and Management	GetChargerAdminAndOperationalState	OMS	To retrieve the administrative state (Locked or Unlocked) and Operational state (Operational/Inservice Active or

				Non-operational/Out of service Failed)
4	Monitoring and Management	GetConnectorsDetails	OMS	To get the properties of all the connectors
5	Monitoring and Management	SetChargerAdminState	OMS	Enable or disable a charger to operate
6	Monitoring and Management	SetConnecterAdminState	OMS	Enable or disable a connector in a charger to operate
7	Monitoring and Management	StopTransaction	OMS	Stop a transaction on a connector id. If the connector id is zero, then stop transaction for all the connectors
8	Diagnostics	DoChargerSelfTest	OMS	Instruct charger to perform a self-test and report its health
9	Diagnostics	DoConnectorSelfTest	OMS	Instruct charger to perform a self-test of specified connector ids and report their health
10	Software Management	UpdateSoftware	OMS	Inform charger to update its software
11	Monitoring and Management	UpdateOMSIP	OMS	Bulk charger to update the IP address of OMS to send notifications for.
12	Monitoring and Management1	ChargerParamConfiguration	OMS	Bulk charger to update the parameters related to charging

8.3.1. Parameters - OMS Initiated Messages

The possible parameters for each of the messages in the above table is given in the table below

#	Message name	Parameters in Message	Parameters in Response
1	GetChargerBasicConfig	None	Make/Model/SWVersion/HWVersion/SerialNumber
2	GetChargerAdvancedConfig	None	Number of connectors, Connector ratings for each connector
3	GetChargerAdminAndOperationalState	None	AdminstrativeState: Enabled/Disabled OperationalState:Locked/Unlocked
4	GetConnectorsDetails	None	Connector Ids, Admin State, Operational State, Charging Capacity

5	SetChargerAdminState	Enabled/Disabled	Status code, Status Message, Status reason, Success/Failure/Deferred
6	SetConnecterAdminState	Enabled/Disabled	Status code, Status Message, Status reason, Success/Failure/Deferred
7	StopTransaction	Connector Id	Status code, status message, Status reason, Success/Failure/Deferred
8	DoChargerSelfTest	None	Status code, Status Message, Status reason, Success/Failure/Deferred
9	DoConnectorSelfTest	Connector Id	Status code, Status Message, Status reason, Success/Failure/Deferred
10	UpdateSoftware	Path of BIN file, optionally start date/time and end date/time between which update is to be done	Status code, Status Message, Status reason, Success/Failure/Deferred
11	Monitoring and Management	IPAddress and Port of new OMS	Status code, Status Message, Status reason, Success/Failure/Deferred
12	Monitoring and Management	TargetSoC to be set by charger in BMS, Driving periodic data retrieval granularity, Charging periodic data retrieval, Effective resistance of power path threshold value to be set by charger in BMS	Status code, Status Message, Status reason, Success/Failure/Deferred

8.4. Messages Initiated by Charger

#	Msg. Types	Message name	Msg. Origin	Purpose
1	Events and Alerts	BootedUp	Charger	To indicate to the OMS that the charger is booted up
2	Heart Beat	IAmAlive	Charger	A heart-beat message to indicate to the OMS that the charger is functioning. This is to be sent every N minutes, where N is received from OMS in response to BootedUp message
3	Charging	IsBatteryAllowedToCharge	Charger	Authentication of BIN by charger

4	Charging	BMSChargingOnetimeDataUpdate	Charger	Update on one-time data details of charging session of BMS
5	Charging	BMSChargingPeriodicDataUpdate	Charger	Update on periodic data details of charging sessions of BMS
6	Charging	ChargingSessionPeriodicUpdate	Charger	Inform OMS of progress of charging like Current stage of charging
7	Discharging	BMSDischargeOnetimeDataUpdate	Charger	Update OMS with the one-time data collected by BMS when it was in vehicle. (Should handle case if the battery was not used and gradually discharged)
8	Discharging	BMSDischargePeriodicDataUpdate	Charger	Update OMS with the periodic data collected by BMS when it was in vehicle. (Should handle case if the battery was not used and gradually discharged)
9	Discharging	VehicleDischargeOnetimeDataUpdate	Charger	Update OMS with the one-time vehicle data collected by BMS when it was in vehicle. (Should handle case if the battery was not used and gradually discharged)
10	Discharging	VehicleDischargePeriodicDataUpdate	Charger	Update OMS with the periodic vehicle data collected by BMS when it was in vehicle. (Should handle case if the battery was not used and gradually discharged)
11	Events and Alerts	NotifyChargerStatusChange	Charger	Inform OMS of any status change of charger based on any event not triggered by OMS
12	Events and Alerts	NotifyConnectorStatusChange	Charger	Inform OMS of any status change of connector in a charger based on any event not triggered by OMS
13	Diagnostics	ChargerSelfTestResult	Charger	Inform OMS of ChargerSelfTestResult
14	Diagnostics	ConnectorSelfTestResult	Charger	Inform OMS of ConnectorSelfTestResult
15	Software Management	NotifySoftwareUpdateStatus	Charger	Inform OMS of status of charger update

16	Charging	ChargingSessionCompletionUpdate	Charger	Inform OMS of status of charging session completion

8.4.1. Parameters - Charger Initiated Messages

The possible parameters for each of the messages in the above table is given in the table below

#	Message name	Parameters in Message	Parameters in Response
1	BootedUp	Make/Model/SWVersion/HWVersion/SerialNumber	HearbeatInterval, OK
2	IAmAlive	Serialnumber, ChargerAdminState ,ChargerOperationalState, List of Connectors' AdminStatus and ConnectorsOperationalStatus	OK
3	IsBatteryAllowedToCharge	SerialNumber, BIN	Yes/No
4	BMSChargingOnetimeDataUpdate	Details of one-time BMS data for the charging session	OK
5	BMSChargingPeriodicDataUpdate	Details of periodic BMS data for the charging session	OK
6	ChargingSessionPeriodicUpdate	Serial number, ChargingSessionId, connector id, Charging Stage, SoC (%),	OK
7	BMSDischargeOnetimeDataUpdate	Details of one-time BMS data log as per the Driving Protocol	
8	BMSDischargePeriodicDataUpdate	Details of periodic BMS data log as per the Driving Protocol	
9	VehicleDischargeOnetimeDataUpdate	Details of one-time Vehicle data log as per the Driving Protocol	
10	VehicleDischargePeriodicDataUpdate	Details of periodic vehicle data log as per the Driving Protocol	
11	NotifyChargerStatusChange	AdminState/OperationalStates of Charger	OK
12	NotifyConnectorStatusChange	AdminState, OperationalStates, Idle	OK
13	ChargerSelfTestResult	SerialNumber, self-test start time, self-test end time, self-test status code, self-test status message and self-test observations	OK
14	ConnectorSelfTestResult	SerialNumber, self-test start time, self-test end time, self-test status code, self-test status message and self-test observations	OK
15	NotifySoftwareUpdateStatus	Serial number, connector id, firmware path, upgrade status	OK
16	ChargingSessionCompletionUpdate	Update from charger on completion of a charging session	OK

8.5. Messages Syntax

This section provides the message syntax between the charger and OMS. The charger and OMS shall communicate using JSON/HTTP. All requests, responses and notifications shall have the following fields.

1	version	The version of the OMS-Charger communication protocol. The first version shall be 1.0.0
2	ts	Timestamp of request or response in the format YYYY-MM-dd HH:mm:ss.SSS where dd is date of month, MM is month of year, YYYY is year, HH is hours in 24 hour format, mm is minutes, ss is seconds and .SSS is milliseconds. All time related parameters will follow the same format.
3	operationname	the name of operation which this JSON carries

- Default URL: "https://<OMS IP Address>:<Port>/oms/<Operationname>/
  - E.g. https://10.9.x.x:8080/oms/bootedupnoti
- There will be no explicit response to notifications, unless otherwise noted, other than the standard 200 OK of HTTP header for acceptance or 403 for a message from a charger that is not registered at OMS.
- Administrative state can take values of Enabled/Disabled in response and Enable/Disable in requests
- Operational state can take values of Locked/Unlocked. A charger or connector will be locked if it is in use and unlocked if it is unused
- All Set or change operations on the chargers shall have the following fields
  - "statuscode": "statuscode" - statuscode should be zero if the operation is success and non-zero otherwise.
  - "statusmessage": "message" - If statuscode is 0, then status message should be "Success" else the string message for failure statuscode has to be filled. E.g.: Set Charger Admin State failed
  - "reason": "reason" - If statuscode is 0, then reason should be "Success" else the reason for should be filled. E.g. Charger is actively charging a battery

8.5.1. OMS Initiated Message Syntax

8.5.1.1. GetChargerBasicConfig

Messages	Request	Response	Remarks
GetChargerBasicConfig	{ "version": "version", "ts": "2017-10-22 08:14:51.443", "operationname": "getchargerbasicconfigreq" }	{ "make": "make", "model": "model", "swversion": "swver", "hwversion": "hwver", "serialnumber": "sernum", "version": "ver1", "ts": "2017-10-22 08:14:51.124", "operationname": "getchargerbasicconfigresp" }	

8.5.1.2. GetChargerAdvancedConfig

Messages	Request	Response	Remarks
GetChargerAdvancedConfig	{ "version": "version", "ts": "2017-10-22 08:14:51.151", "operationname": "getchargeradvancedconfigreq" }	{ "numofconn": "numofconn", "connrating": "connrating", "version": "version", "ts": "2017-10-22 08:14:51.297", "operationname": "getchargeradvancedconfigresp" }	Connector rating in kilowatts

8.5.1.3. GetChargerAdminAndOperationalState

Messages	Request	Response	Remarks
GetChargerAdminAndOperationalState	{ "version": "version", "ts": "2017-10-22 08:14:50.859", "operationname": "getchargeradminandoperationalstatereq" }	{ "administrativestate": "administrativestate", "operationalstate": "operationalstate", "version": "version", "ts": "2017-10-22 08:14:51.6", "operationname": "getchargeradminandoperationalstateresp" }	

8.5.1.4. GetConnectorsDetails

Messages	Request	Response	Remarks
GetConnectorsDetails	{ "version": "version", "ts": "2017-10-22 08:14:53.242", "operationname": "getconnectordetailsreq" }	{ "connectorslist": [{ "connectorid": "cid1", "adminstate": "adminstate1", "operationalstate": "ostate1", "chargingcapacity": "chcap1" }, { "connectorid": "cid2", "adminstate": "adminstate2", "operationalstate": "ostate2", "chargingcapacity": "chcap2" }], "version": "version", "ts": "2017-10-26 12:59:31.680", "operationname": "getconnectorsdetailsresp" }	Connectorslist has to be populated with details for each connector

8.5.1.5. SetChargerAdminState

Messages	Request	Response	Remarks

SetChargerAdminState	{
	“connectorid”:”cid1”,
	"status": "status",
	"version": "version",
	"ts": "2017-10-22 08:14:52.942",
	"operationname": "setchargeradminstatereq"
	}
		{
		"statuscode": "statuscode",
		"statusmessage": "message",
		"reason": "reason",
		"version": "version",
		"ts": "2017-10-22 08:14:53.91",
		"operationname": "setchargeradminstateresp"
		}
			In request status value can be Enable or Disable In reply, statuscode should be zero if the operation is success and non-zero otherwise.
			
			If statuscode is 0, then status message has to be "Success" else the string message for failure statuscode has to be filled. E.g: Set Charger Admin State failed.
			If statuscode is 0, then reason has to be "Success" else the reason for has to be filled. E.g. Charger is actively charging a battery

8.5.1.6. SetConnecterAdminState

Messages	Request	Response	Remarks

SetConnecterAdminState	{
	“connectorid”: connid”,
	"status": "status",
	"version": "version",
	"ts": "2017-10-22 08:14:53.242",
	"operationname": "setconnecteradminstatereq"
	}
		{
		"statuscode": "statuscode",
		"statusmessage": "message",
		"reason": "reason",
		"version": "version",
		"ts": "2017-10-22 08:14:53.392",
		"operationname": "setconnecteradminstateresp"
		}	

8.5.1.7. StopTransaction

Messages	Request	Response	Remarks

StopTransaction	{
	"connectorid": "connid",
	"version": "version",
	“ts": "2017-10-24 15:55:21 .515",
	"operationname": "stopchargingreq"
	}
		{
		"connector": "connector",
		"statuscode": "statuscode",
		"statusmessage": "statusmessage",
		"reason": "reason",
		"status": "status",
		"version": "version",
		"ts": "2017-10-24 15:59:45.208",
		"operationname": "stopchargingresp"
		}	

8.5.1.8. DoChargerSelfTest

Messages	Request	Response	Remarks
DoChargerSelfTest	{ "version": "version", "ts": "2017-10-22 08:14:50.563", "operationname": "dochargerselftestreq" }	{ "statuscode": "statuscode", "statusmessage": "message", "reason": "reason", "version": "version", "ts": "2017-10-22 08:14:50.711", "operationname": "dochargerselftestresp" }	

8.5.1.9. DoConnectorSelfTest

Messages	Request	Response	Remarks
DoConnectorSelfTest	{ “connectorid” : “connid” "version": "version", "ts": "2017-10-22 08:14:50.563", "operationname": "doconnectorselftestreq" }	{ "statuscode": "statuscode", "statusmessage": "message", "reason": "reason", "version": "version", "ts": "2017-10-22 08:14:50.711", "operationname": "doconnectorselftestresp" }	

8.5.1.10. UpdateSoftware

Messages	Request	Response	Remarks
UpdateSoftware	{ "binpath": "path", "startdate": "start", "enddate": "end", "version": "version", "ts": "2017-10-22 08:14:53.539", "operationname": "updatesoftwarereq" }	{ "statuscode": "statuscode", "statusmessage": "message", "reason": "reason", "version": "version", "ts": "2017-10-22 08:14:50.711", "operationname": "updatesoftwareresp" }	path is the location of firmware. Charger should download the software between start date and end date using FTP. The response only indicates the acceptance of the request and not the status of software upgrade itself which is notified via NotifySoftwareUpdateStatus

8.5.1.11. UpdateOMSIP

Messages	Request	Response	Remarks
UpdateOMSIP	{ "ipaddress": "ip", “port” : “port”, "version": "version", "ts": "2017-10-22 08:14:52.328", "operationname": "updateomsipreq" }	{ "statuscode": "statuscode", "statusmessage": "message", "reason": "reason", "version": "version", "ts": "2017-10-22 08:14:50.711", "operationname": "updateomsipresp" }	To change the OMS IP to a new IP

8.5.1.12. ChargerParamConfiguration

Messages	Request	Response	Remarks
ChargerParamConfiguration	{ “connectorid”:”cid1”, "targetsoc": "targetsoc", “chargingperiodicdataretgranularity”:” chargingperiodicdataretgranularity”, “drivingperiodicdataretgranularity”;” drivingperiodicdataretgranularity”, “effrespowerpath”:” effrespowerpath”, "version": "version", "ts": "2017-10-22 08:14:52.328", "operationname": "chargerparamconfiguration" }	{ "statuscode": "statuscode", "statusmessage": "message", "reason": "reason", "version": "version", "ts": "2017-10-22 08:14:54.711", "operationname": " chargerparamconfiguration " }	To set parameters in the charger. If it is for specific connector then connectorid carries the id number and if it is for all connectors, connectorid will be ‘0’. 1.TargetSoC to be set by charger in BMS 2. To set charging periodic data retrieval granularity for 1 sec data. By default the value is 1 second.

			3. To set driving periodic data retrieval granularity for 1 sec data. By default the value is 1 second.4.Effective resistance of power path threshold value – 25 mOhm default value.

8.5.2. Charger Initiated Message Syntax

8.5.2.1. BootedUp

Messages	Request	Response	Remarks

BootedUp	{
	"make": "make",
	"model": "model",
	"swversion": "swversion",
	"hwversion": "hwversion",
	"serialnumber": "sno",
	"version": "ver",
	"ts": "2017-10-22 08:16:18.807",
	"operationname": "bootedupnoti"
	}
		{“heartbeatinterval”: “heartbeatinterval”
		"version": "version",
		"ts": "2017-10-22 08:14:50.711",
		"operationname": "bootedupnotiresp"
		}
			Heartbeat interval in minutes

8.5.2.2. IAmAlive

Messages	Request	Response	Remarks
IAmAlive	{ "chargeradminstate": "chargeradminstatus", "chargeroperationalstate": "chargeroperationalstate", "connstslist": [{ "connectorid": "1", "adminstatus": "adminstatus", "operationalstatus": "operstatus" }, { "connectorid": "2", "adminstatus": "adminstatus", "operationalstatus": "operstatus" }], "serialnumber": "sno", "version": "version", "ts": "2017-10-24 15:52:00.642", "operationname": "iamalivenoti" }		Has to be repeated for all connectors

8.5.2.3. IsBatteryAllowedToCharge

Messages	Request	Response	Remarks
IsBatteryAllowedToCharge	{ "serialnumber": "sno", "bin": "bin", "version": "version", "ts": "2017-10-22 08:16:20.626", "operationname": "isbatteryallowedtochargereq" }	{ "bin": "bin", "statuscode": "statuscode", "statusmessage": "message", "reason": "reason", "version": "version", "ts": "2017-10-22 08:14:50.711", "operationname": "isbatteryallowedtochargeresp" }	Charger should charge the battery only if the status code is 0 i.e, success

8.5.2.4. BMSChargingOnetimeDataUpdate

Messages	Request	Response	Remarks

BMSChargingOneTimeDataUpdate	{
	"ufd": "ufd",
	"chargerfwver": "chargerfwver",
	"bmsfwver": "bmsfwver",
	"chargingstarttime": "2017-10-24 19:40:52.75",
	"chargingendtime": "2017-10-24 19:40:52.75",
	"serialnumber": "serialnumber",
	"connectorid": "connid",
	"bin": "bin",
	"numberofcells": "numofcells",
	"numberoftempsensors": "numoftempsensors",
	"startsoc": "startsoc",
	"endsoc": "endsoc",
	"chargingenergy": "chargingenergy",
	"chargingdurationinseconds": "chargeduration",
	"charginglifecyclenumber": "lifecyclenumber",
	"chargingsessionid": "chargingsessionid",
	"version": "version",
	"ts": "2017-10-24 19:40:52.75",
	"operationname": "chargingonetimedatanoti"
	}
			Charging energy is the energy taken for charging the battery
			UFD will be alphanumeric value written to the battery by HHD
			Chargerfwver is the firmware version of charger captured during charging
			Bmsfwver is the firmware version of BMS captured during charging
			sessionid is a value formed by concatenating serialnumber of charger , timestamp of start of charging session(after removing the space, - , : and . between date and time a and connectorid ie. serinalnumber.tsofstartofsession.cnid
			E.g: serialnumber= sno
			tsofstartofsession=YYYY-MM-dd HH:mm:ss.SSS
			Connector id = 4
			Then the charging session id would be sno.YYYYMMddHHmmssSSS.4

8.5.2.5. BMSChargingPeriodicDataUpdate

Messages	Request	Response	Remarks
BMSChargingPeriodicTimeDataUpdate	{ "chargingsessionid": "chargingsessionid",		Cellvoltagelist and temperature

	"serialnumber": "chargerserialnumber", "connectorid": "connectorid","sessionid": "sessionid", "cpdlist": [{ "ts": "2017-10-24 19:44:44.85", "balancingcurrentstatus": "balancingcurrentstatus", "cellvoltagelist": [{ "cellid": "cellid", "cellvoltage": "cellvoltage" }, { "cellid": "cellid", "cellvoltage": "cellvoltage" }], "temperaturelist": [{ "tempsensorid": "tempsensorid", "temperature": "temperature" }, { "tempsensorid": "tempsensorid", "temperature": "temperature" }] }, { "ts": "2017-10-24 19:44:44.85", "balancingcurrentstatus": "balancingcurrentstatus", "cellvoltagelist": [{ "cellid": "cellid", "cellvoltage": "cellvoltage" }, { "cellid": "cellid", "cellvoltage": "cellvoltage" }], "temperaturelist": [{ "tempsensorid": "tempsensorid", "temperature": "temperature" }, { "tempsensorid": "tempsensorid", "temperature": "temperature" }] }], "version": "version", “ts": "2017-10-24 19:44:44.74",		list has to be repeated for total number of cells and total number of temperature sensors respectively

	"operationname": "chargingperiodicdatanoti" }		

8.5.2.6. ChargingSessionPeriodicUpdate

Messages	Request	Response	Remarks
ChargingSessionPeriodicUpdate	{ "sessionid": "sessionid", "serialnumber": "chargerserialnumber", "connectorid": "connectorid", "currentstage": "stageasdefinedinchargingprotocol", "version": "version", "ts": "2017-10-22 23:24:45.399", "operationname": "chargingsessionnoti" }		

8.5.2.7. BMSDischargeOnetimeDataUpdate

Messages	Request	Response	Remarks
BMSDischargeOnetimeDataUpdate	{ "botdlist": [{ "datats": "2017-10-22 23:00:13.897", "batterymode": "battmode", "drivingprotocolversion": "drivver", "bin": "bin", "vin": "vin", "numofcells": "numofcells", "numoftemperaturesensors": "numoftempsens", "availableenergy": "availenergy", "bmsfwverdri":"bmsfwverdri" }, { "datats": "2017-10-22 23:00:13.941", "batterymode": "battmode", "drivingprotocolversion": "drivver", "bin": "bin", "vin": "vin", "numofcells": "numofcells", "numoftemperaturesensors": "numoftempsens", "availableenergy": "availenergy", "bmsfwverdri":"bmsfwverdri" }], "serialnumber": "chargerserialnumber",		bmsfwverdri is the version of BMS captured during driving protocol

	"connectorid": "connectorid", "sessionid": "sessionid", "version": "ver", "ts": "2017-10-22 23:00:13.941", "operationname": "bmsdischargeonetimedatanoti" }		

8.5.2.8. BMSDischargePeriodicDataUpdate

Messages	Request	Response	Remarks

BMSDischargePeriodicDataUpdate	{
	"bpdlist": [{
	"datats": "datats",
	"availableenergy":
	"availableenergy",
	"maxcurrent": "maxcurrent",
	"batteryinstvolt":
	"batteryinstvolt",
	"batteryinstcurrent":
	"batteryinstcurrent",
	"cellvoltagelist": [{
	"cellid":
	"cellid",
	"cellvoltage":
	"cellvoltage"
	},
	{
	"cellid":
	"cellid",
	"cellvoltage":
	"cellvoltage"
	}
	],
	"temperaturelist": [{
	
	"tempsensorid":
	"tempsensorid",
	
	"temperature":
	"temperature"
	},
	{
	
	"tempsensorid":
	"tempsensorid",
			Cellvoltagelist and temperature list has to be repeated for total number of cells and total number of temperature sensors respectively

	"temperature": "temperature" }], "balancingcurrentstatus": "balancingcurrentstatus" }], "batambtemplist": [{ "datats": "datats", "locksmartmode": "locksmartmode", "batteryambienttemperature": "batteryambienttemperature" }], "suspreasonlist": [{ "datats": "datats", "errorcode": "errorcode", "thresholdvalue": "thresholdvalue", "breachvalue": "breachvalue" }], "vinbinmismatchlist": [{ "datats": "datats", "errorcode": "errorcode", "vin": "vin" }], "serialnumber": "serialnumber", "connectorid": "connectorid", "sessionid": "sessionid", "version": "version", "ts": "ts", "operationname": "batteryperiodicdatanoti" }		

8.5.2.9. VehicleDischargeOnetimeDataUpdate

Messages	Request	Response	Remarks
VehicleDischargeOnetimeDataUpdate	{ "votdlist": [{ "datats": "2017-10-22 22:52:41.359", "vin": "vin", "powerpatheffres": "powerpatheffres", "odometer":"odometer ", "vcufwver": "vcufwver"		

	}, {
	"datats": "2017-10-22 22:52:41.388",
	"vin": "vin",
	"powerpatheffres":
	"powerpatheffres",
	"odometer":"odometer ",
	"vcufwver":"vcufwver"
	}],
	"serialnumber":
	"chargerserialnumber",
	"connectorid": "connectorid",
	"sessionid": "sessionid",
	"version": "ver",
	"ts": "2017-10-22 22:52:41.388",
	"operationname":
	"vehicledischargeonetimedatanoti"
	
	}		

8.5.2.10. VehicleDischargePeriodicDataUpdate

Messages	Request	Response	Remarks

VehicleDischargePeriodicDataUpdate	{
	"vpdlist": [{
	"somrList": [{
	
	"speedometerreading ":
	"speed",
	"odometer":"odometer",
	"datats": "2017-26-
	10 22:29:27.135"
	}, {
	
	"speedometerreading":
	"speed1",
	"odometer":"odometer1",
	"datats": "2017-26-
	10 22:29:27.135"
	}],
	"vsd1List": [{
	"datats": "2017-26-
	10 22:29:27.135",
	"oc1": "oc1",
	"oc2": "oc2"
	}, {
	"datats": "2017-26-
	10 22:29:27.135",
	"oc1": "oc1",
			Speed is in centikms/hr and odometer is in kms

	"oc2": "oc2"
	}],
	"vsd2List": [{
	"datats": "2017-26-10 22:29:27.135",
	"oc3": "oc4",
	"oc4": "oc4"
	}, {
	"datats": "2017-26-10 22:29:27.135",
	"oc3": "oc4",
	"oc4": "oc4"
	}],
	
	"vehcontrollerctList": [{
	"datats": "2017-26-10 22:29:27.135",
	"vehcontrollerct": "vehcontrollerct"
	}, {
	"datats": "2017-26-10 22:29:27.135",
	"vehcontrollerct": "vehcontrollerct"
	}]
	}],
	"serialnumber": "chargerserialnumber",
	"connectorid": "connectorid",
	"sessionid": "sessionid",
	"version": "version",
	"ts": "2017-26-10 22:29:27.101",
	"operationname": "vehicleperiodicdatanoti"
	}		

8.5.2.11. NotifyChargerStatusChange

Messages	Request	Response	Remarks

NotifyChargerStatusChange	{
	"serialnumber": "sno",
	"adminstate": "adminstate",
	"operationalstate":
	"operationalstate",
	"version": "version",
	"ts": "2017-10-22
	08:16:20.919",
	"operationname":
	"notifychargerstatuschangenoti"
	}		

8.5.2.12. NotifyConnectorStatusChange

Messages	Request	Response	Remarks

NotifyConnectorStatusChange	{
	"serialnumber": "sno",
	"connectorid": "cnid",
	"adminstate": "adminstate",
	"operationalstate":
	"operationalstate",
	"version": "version",
	"ts": "2017-10-22 08:16:20.919",
	"operationname":
	"notifyconnectorstatuschangenoti"
	}		

8.5.2.13. ChargerSelfTestResult

Messages	Request	Response	Remarks

ChargerSelfTestResult	{
	"serialnumber": "sno",
	"selfteststarttime" :
	"starttime"
	"selftestendtime" :
	"endtime"
	"selfteststatuscode" :
	"code"
	"selfteststatusmesg" :
	"mesg"
	
	"selfteststatusobservations
	: "observations"
	"version": "version",
	"ts": "2017-10-22
	08:16:20.919",
	"operationname":
	"chargerselftestnoti"
	}
			For a successful self test, the code should be zero. If this is not implemented, then the code should be 1.

8.5.2.14. ConnectorSelfTestResult

Messages	Request	Response	Remarks

ConnectorSelfTestResult	{
	"serialnumber": "sno",
	"connectorid": "cnid",
	"selfteststarttime" : "starttime"
	"selftestendtime" : "endtime"
	"selfteststatuscode" : "code"
	"selfteststatusmesg" : "mesg"
	
	"selfteststatusobservations : "observations"
	"version": "version",
	"ts": "2017-10-22 08:16:20.919",
	"operationname": "connectorselftestnoti"
	}
			For a successful self-test, the code should be zero. If this is not implemented, then the code should be 1.

8.5.2.15. NotifySoftwareUpdateStatus

Messages	Request	Response	Remarks

NotifySoftwareUpdateStatus	{
	"serialnumber": "sno",
	"path" : "pathoffirmware"
	"upgradestatus" : "upgradestatus"
	“swver”:”swver”,
	"version": "version",
	"ts": "2017-10-22 08:16:20.919",
	"operationname": "swupdatenoti"
	}		

8.5.2.16. ChargingSessionCompletionUpdate

Messages	Request	Response	Remarks
ChargingSessionCompletionUpdate	{ "sessionid": "sessionid", "serialnumber": "serialnumber", "connectorid": "connectorid",		

	"bin": "bin", "binwasmaster": "false", “ufd” : “ufd” "vin": "vin", "starttime": "2017-10-22 23:24:45.399", "endtime": "2017-10-22 23:24:45.399", "startsoc": "startsoc", "endsoc": "endsoc", "chargingcompletioncode": "chargingcompletioncode", "chargingcompletionmessage": "chargingcompletionmessage", "chargingenergy": "chargingenergy", "version": "version", "ts": "2017-10-22 23:24:45.399", "operationname": "chargingsessioncompnoti" }		

8.6. *Data Types*

#	Parameters	Data Type	Length	Units	Remarks
1	make	String	250		
2	model	String	250		
3	swversion	String	50		
4	hwversion	String	50		
5	serialnumber	String	250		
6	heartbeatinterval	Integer		Seconds	
7	chargeradminstate	String			Enumeration of Enabled or Disabled
8	chargeroperationalstate	String			Enumeration of Locked or UnLocked
9	chargingstarttime	String			Similar to ts format
10	chargingendtime	String			Similar to ts format
11	connectorid	Integer			This contains the id of the connector. Connector id is in the range of ‘1’ to ‘n’.
12	bin	String	50		BIN is currently 20 bytes. 50 is specified for future expansion
13	numberofcells	Integer			

14	numberoftempsensors	Integer			
15	startsoc	Float			
16	endsoc	Float			
17	chargingenergy	Float			kWh
18	chargingdurationinseconds	Integer			
19	charginglifecyclenumber	Integer			
20	chargingsessionid	String			sessionid is a value formed by concatenating serialnumber of charger , connectorid, timestamp of start of charging session(after removing the space, - , : and . between date and time a and connectorid ie. serinalnumber.cnid.tsofstartofsession.cnid Connector id has to represented in three digits. i.e connector id 4 would be represented as 004 E.g: serialnumber= sno tsofstartofsession=YYYY-MM-dd HH:mm:ss.SSS Then the charging session id would be sno.YYYYMMddHHmmssSSS
21	currentstage	String			Enumeration of Auto address assignment stage, Handshake stage, Battery authenticity check stage, Driving log transfer stage, Parameter configuration stage, Charging stage, Charging log transfer stage, End-of-charging stage
22	selfteststatuscode	Integer			selfstatuscode should be zero if the operation is success and non-zero otherwise.
23	selfteststatusmesg	String	500		If selfstatuscode is 0, then selfstatus message has to be "Success" else the string message for failure selfstatuscode has to be filled. E.g: Set Charger Admin State failed
24	selfteststatusobservations	String	500		Generic field to report on self-test

25	path	String	500		Path where the software binary can be downloaded from
26	upgradestatus	String	500		Enumeration of "Successfully upgraded" or "File fetch failed" or "Storing file locally failed" or "Upgrade failed" or "General failure - <Specifics that need to be notified to OMS>"
27	swver	String	50		New software version after the upgrade is done successfully
28	binwasmaster	String			Enumeration of "true" or "false"
29	vin	String	17		Vehicle Identification Number
30	chargingcompletioncode	Integer			Same as statuscode
31	chargingcompletionmessage	String	500		Same as status message
32	chargingenergy	Float		kWh	
33	balancingcurrentstatus	Integer	0 or 1		0 implies = balancing and 1 implies not balancing
34	cellid	Integer			
35	cellvoltage	Float			
36	tempsensorid	Integer			
37	temperature	Float			
38	batterymode	Integer	Enumeration of 0 or 1		1 implies master and 0 implies slave
39	drivingprotocolversion	String	50		
40	numofcells	Integer			
41	availableenergy	Float		kWh	
42	maxcurrent	Float			
43	batteryinstvolt	Float			
44	powerpatheffres	Float			
45	speedometerreading	Float			
46	odometerreading	Float			
47	oc1	Integer			
48	oc2	Integer			
49	oc3	Integer			
50	oc4	Integer			
51	t argetsoc	Integer			SoC of 80.5 has to be represented as integer 8050
52	port	Integer			Port number of OMS server
53	ip	String			IP address or hostname of OMS server

54	drivingperiodicdataretgranularity	Integer		Second	Granularity in Seconds. Default value: 1. Range : 1 to 60 i.e., should not be set to more than 60 seconds
55	chargingperiodicdataretgranularity	Integer		Second	Granularity in Seconds. Default value: 1. Range : 1 to 60 i.e., should not be set to more than 60 seconds
56	ufd	String	8		Written by HHD over BLE when battery is assigned to a vehicle in a swapping station.
57	effrespowerpath	Integer		mOhm	Threshold value of effective resistance of power path. Default value is 25mOhm
58	vehcontrollerct	Float			
59	odometer	Float			
60	locksmartmode	Integer			Charge Mode:0x00;Drive mode:0x01
61	batteryambienttemperature	Float			
62	errorcode	Integer			
63	thresholdvalue	Float			
64	breachvalue	Float			
65	chargerfwver	String	50		Charger firmware version
66	vehiclespeed	Float			Vehicle speed
67	bmsfwver	String	50		BMS firmware version during charging
68	bmsfwverdri	String	50		BMS firmware version during driving
69	vcufwver	String	50		VCU firmware version

9. References

SAE J1939 – Recommended practice for a serial control and communications vehicle network

SAE J1939 / 14_201612 - Physical layer, 500 Kbps

SAE J1939 / 21_200612 – Data link layer

SAE J1939 / 71_200112 – Vehicle application layer

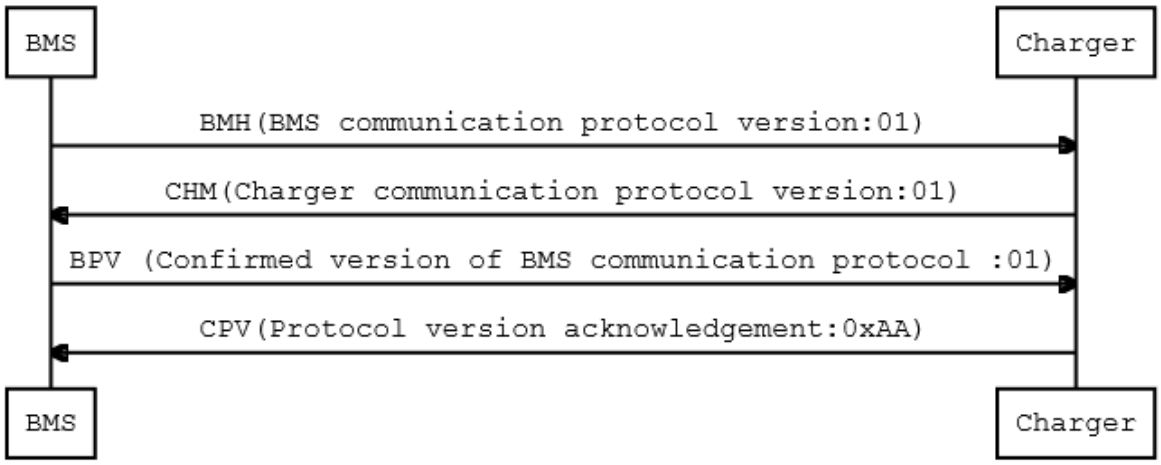
SAE J1939 / 73_199602 – Application Layer - Diagnostics

SAE J1939 DA_201707 – Digital Annex of serial control and communication heavy duty vehicle network data – Jul 2017

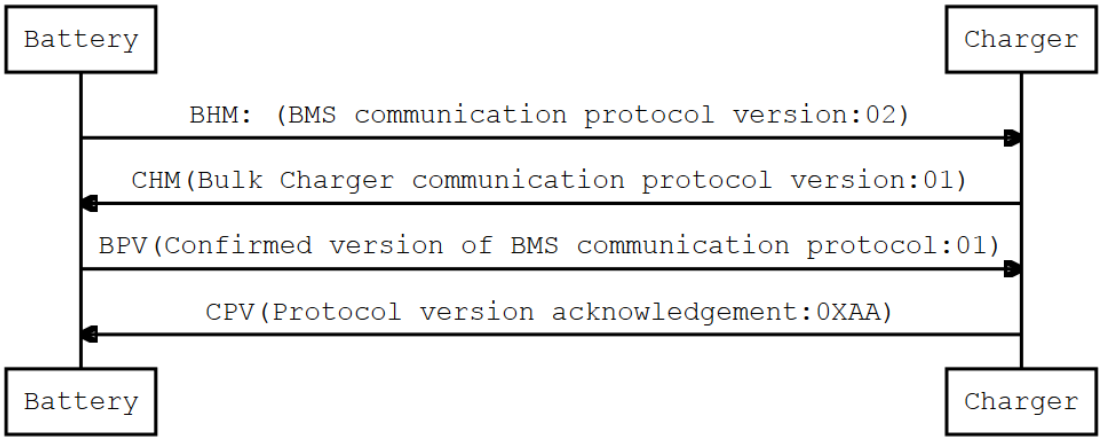
Annexure A: Protocol Version Matching Flow

A.1 Charging Protocol Version Matching

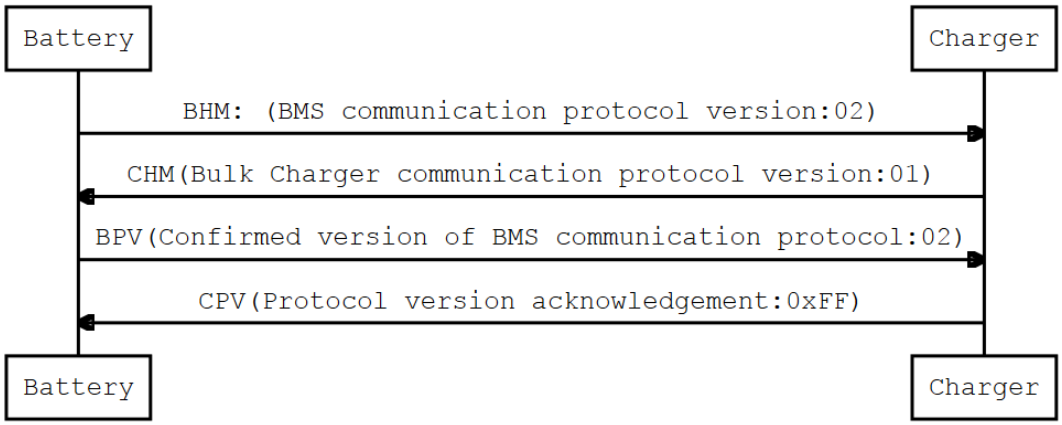
Case 1: BMS communication protocol version is '01' and Bulk Charger communication protocol version is '01'. BMS confirms the protocol version as 01. The result would be "**Success**" to indicate charger will also communicate with protocol version 01 . The same scenario would be applicable whenever both the entities are in the same version.



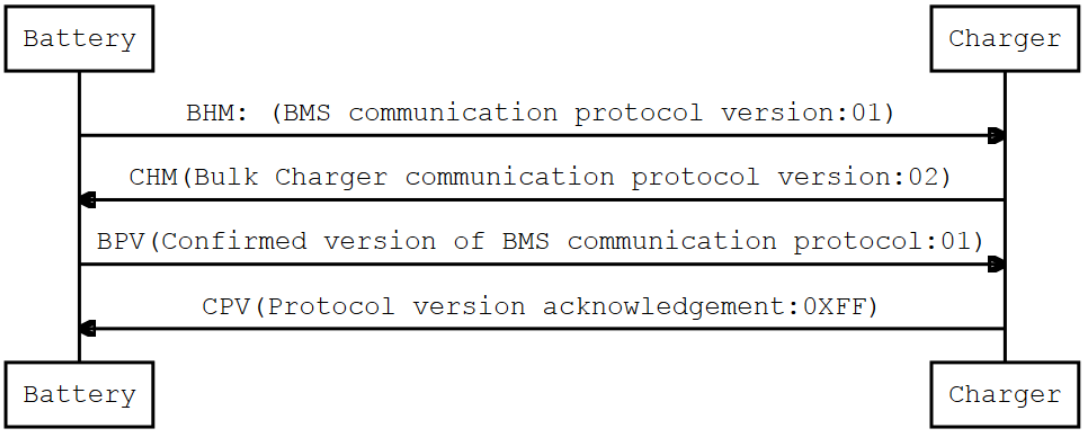
Case 2: BMS communication protocol version is '02' and Bulk Charger communication protocol version is '01'. BMS confirms protocol version as '01' only. The result would be "**Success**" if charger is capable of backward compatibility to communicate with protocol version '01' itself.



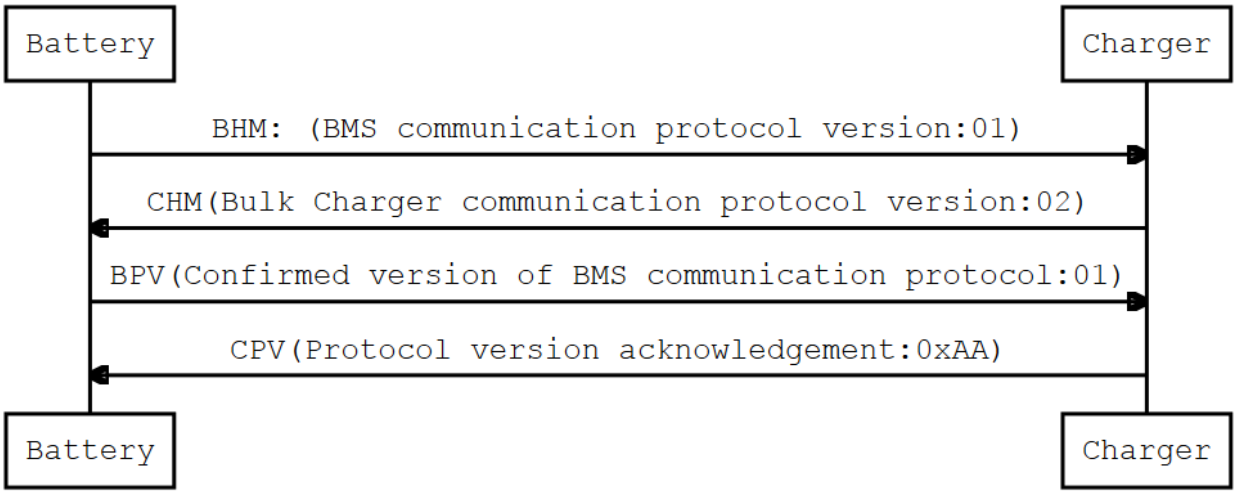
Case 3: BMS communication protocol version is '02' and Bulk Charger communication protocol version is '01'. BMS confirms protocol version as '02' only. The result would be "**Failure**" as charger is at lesser version than the version requested by BMS.



Case 4: BMS communication protocol version is '01' and Bulk Charger communication protocol version is '02'. BMS confirms the protocol version as '01'. The result would be "**Failure**" if the charger is not capable of backward compatibility of protocol with Version '01'.

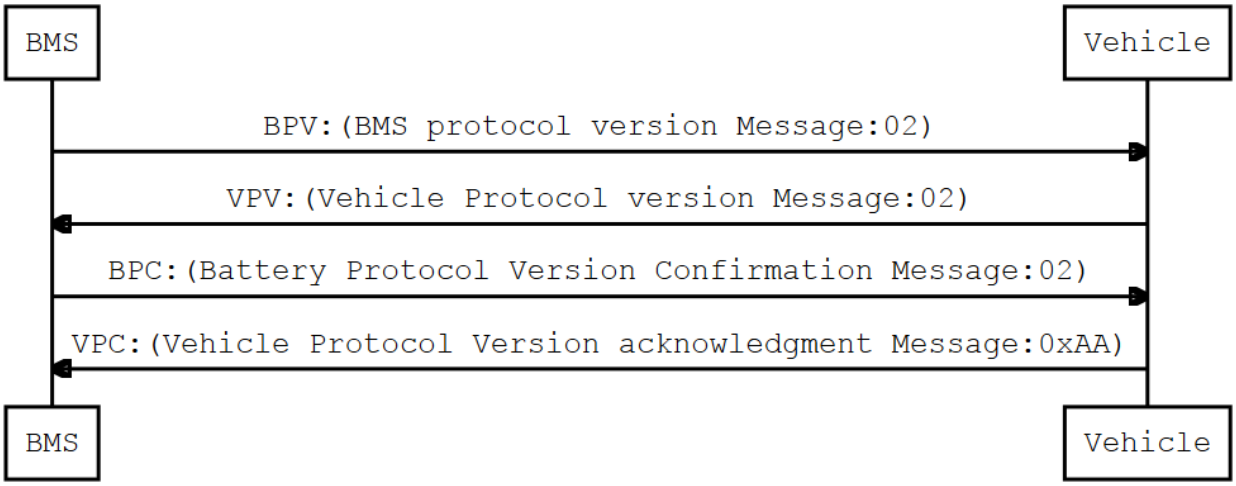


Case 5: BMS communication protocol version is '01' and Bulk Charger communication protocol version is '02'. BMS confirms with protocol version '01'. The result would be "**Success**" if charger is capable of backward compatible to communicate with protocol version 01.

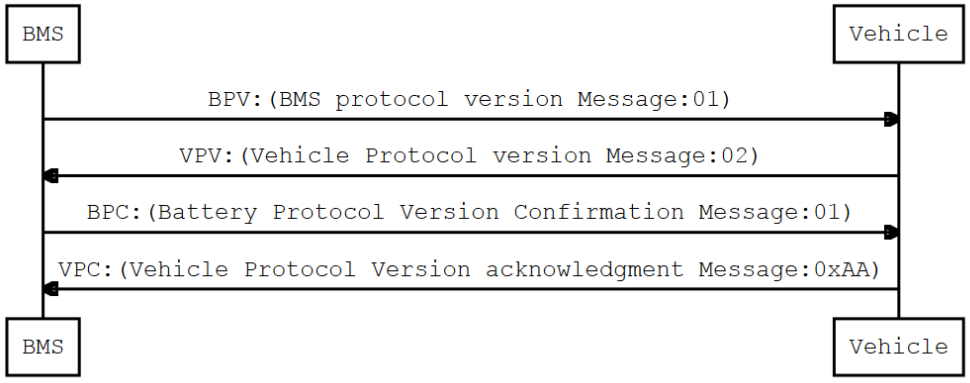

Battery
Charger
BHM: (BMS communication protocol version:01)
CHM(Bulk Charger communication protocol version:02)
BPV(Confirmed version of BMS communication protocol:01)
CPV(Protocol version acknowledgement:0xAA)
Battery
Charger

A.2 Driving Protocol Version Matching Flow

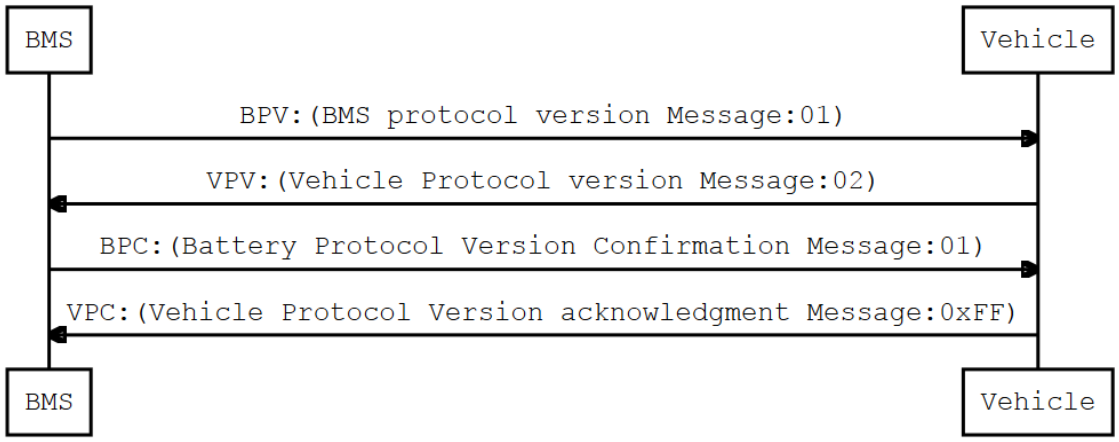
Case 1: BMS protocol version is '02' and Vehicle Protocol version is '02'. BMS confirms protocol version as 02. The result would be "**Success**" to indicate that vehicle will communicate with protocol version 02 with BMS. The same scenario is applicable whenever both the entities are in the same version.



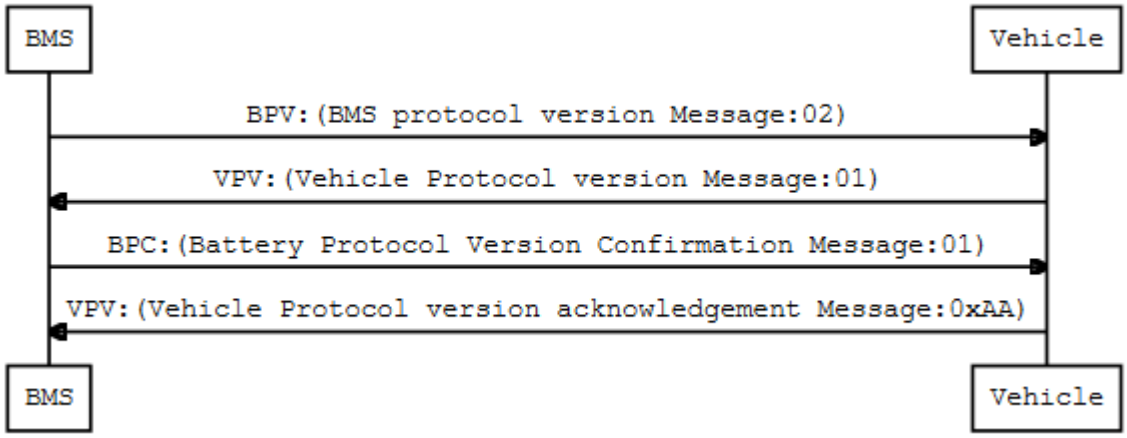
Case 2: BMS protocol version is '01' and Vehicle Protocol version is '02' .BMS confirms protocol version as 01 only. The result would be "**Success**" if the vehicle protocol is backward compatible to communicate with protocol version 01 with BMS.



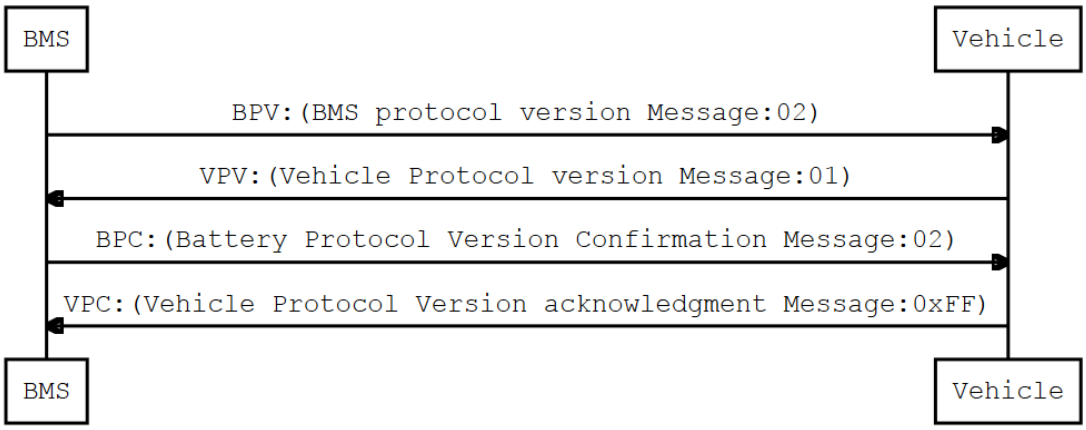
Case 3: BMS protocol version is '01' and Vehicle Protocol version is '02'. BMS confirms the version as '01'. The result would be "**Failure**" if vehicle is not capable of backward compatible to communicate with protocol version '01' with BMS.



Case 4: BMS protocol version is '02' and Vehicle Protocol version is '01 .BMS confirms the version as '01'. The result would be "**Success**" if vehicle version is backward compatible to communicate with protocol version '01' with BMS



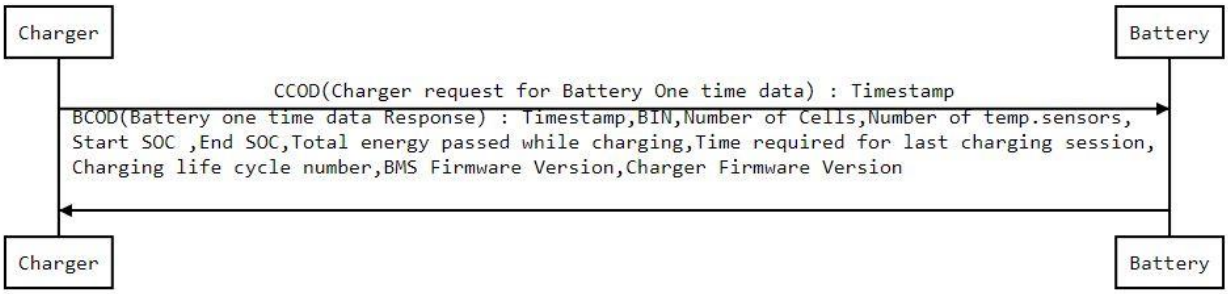
Case 5: BMS protocol version is '02' and Vehicle Protocol version is '01'. BMS confirms version as '02'. The result would be "**Failure**" if BMS protocol is not capable of backward compatible to communicate with protocol version '01' of vehicle.


BMS
Vehicle
BPV:(BMS protocol version Message:02)
VPV:(Vehicle Protocol version Message:01)
BPC:(Battery Protocol Version Confirmation Message:02)
VPC:(Vehicle Protocol Version acknowledgment Message:0xFF)
BMS
Vehicle

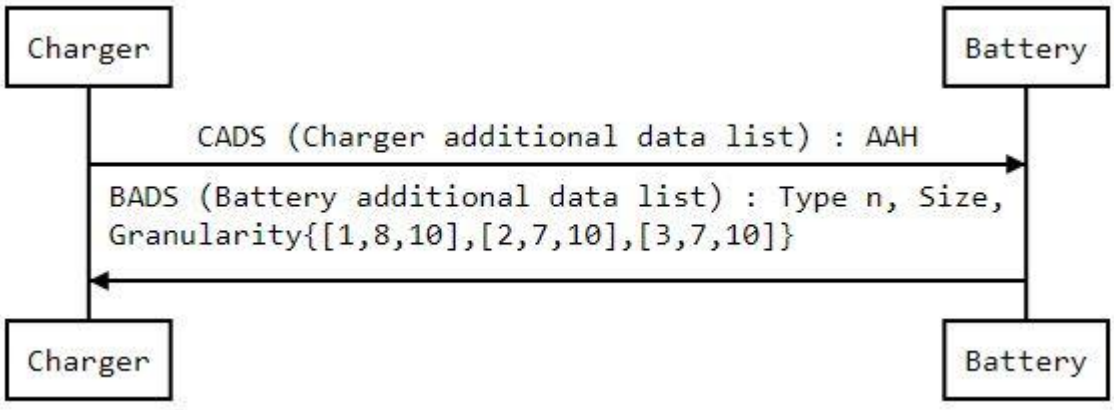
Annexure B: Data Log Message Flow Diagrams

B1. Charging Data Log messages

B1.1 Charger sends CCOD (Charger One time data) Request to the Battery. Battery replies with BCOD (Battery one time data) response. The corresponding BCOD Packet has Timestamp, BIN, Number of Cells, Number of Temperature Sensors, Start SOC, End SOC, Total energy passed while charging, Time required for Last charging, Charging Life cycle number.

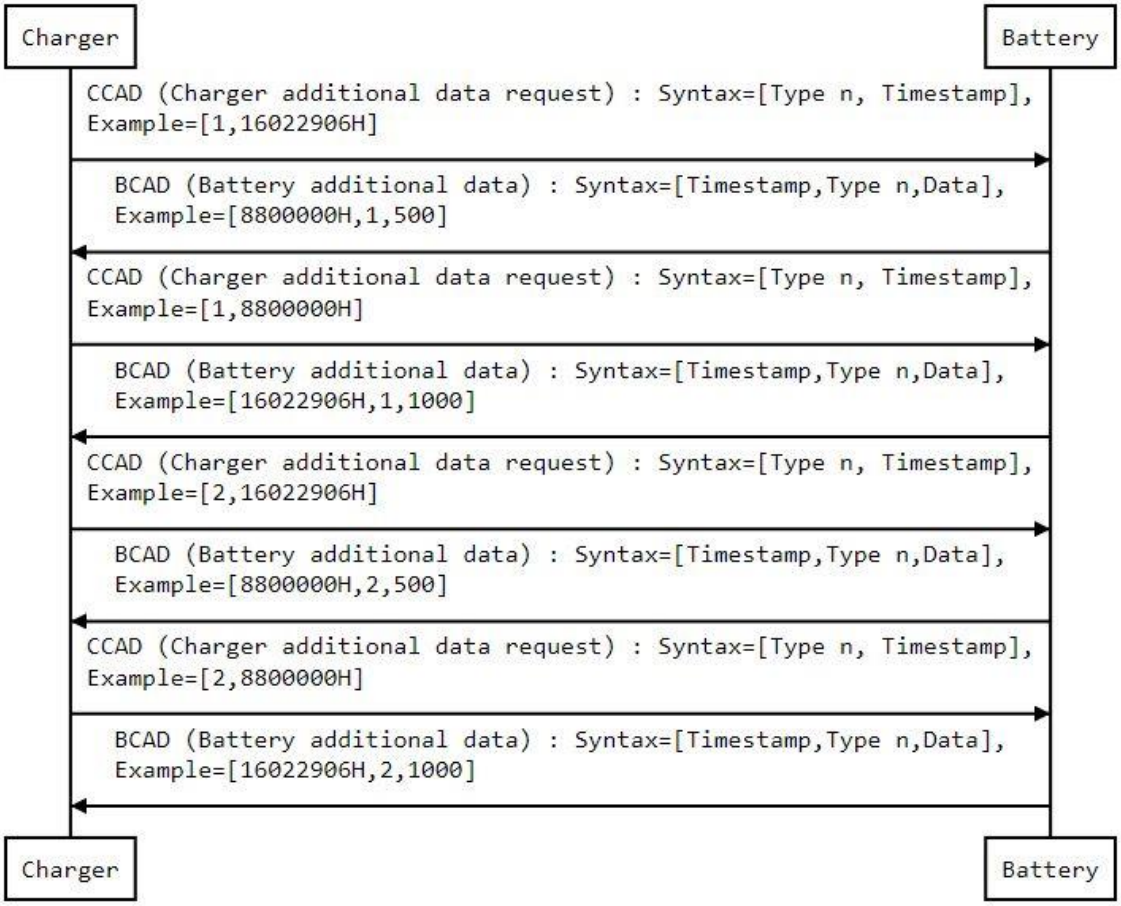


B1.2 Charger sends CADS (Charger additional data list) Request to the Battery. The CADS Request value is AAH and Battery replies with BADS (Battery additional data list) response. The corresponding BADS Packet has Type n, Size (Bytes), Granularity (Secs).For instance battery has 3 different size of data and different granularities. There will be 3 packets namely {[1,8,10], [2,7,10], [3,7,10]},So battery informs to charger that, it has three different types of data.



B1.3 Charger sends CCAD (Charger additional data request) Request to the Battery. The CCAD Request packet will have Type n and Timestamp. For instance CCAD requesting Type 1 and with default timestamp 16022906H . The Battery response with BCAD (Battery additional data) response packet. The Response packet will have Timestamp (8800000H), Type (one) and data (500). Next CCAD requesting

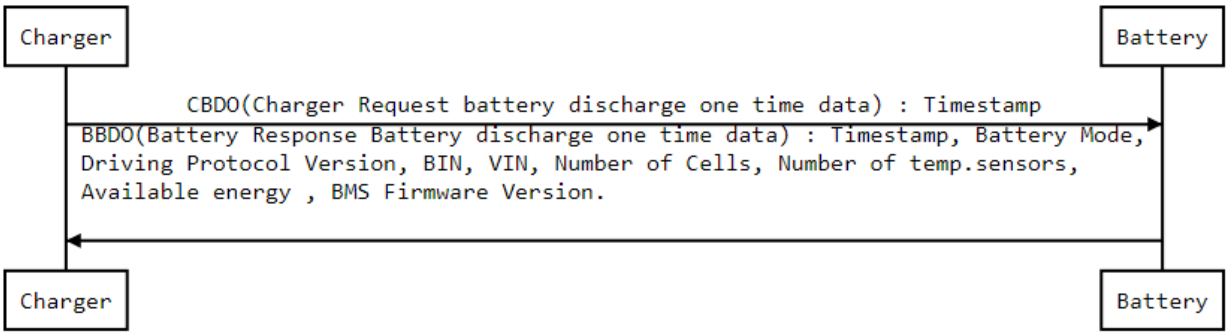
Type one data with teimstamp as 8800000H received from previous packet. The Battery response with BCAD (Battery additional data) response packet. The Response packet will have Timestamp (19023906H), Type (one), Size (2), Granularity (10) and data (1000).



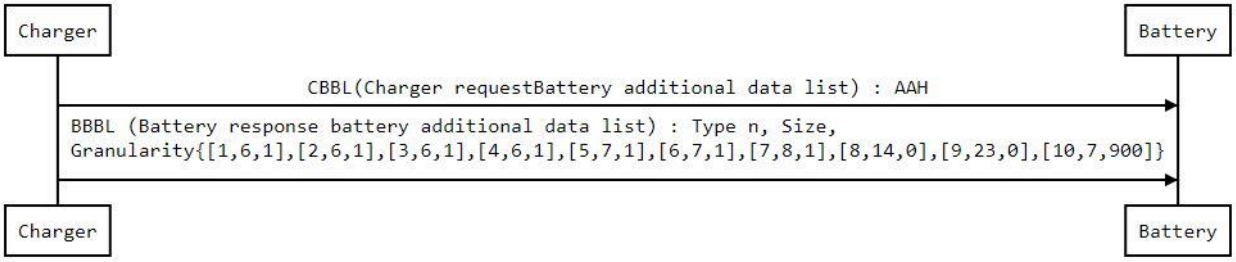
Similary the iterations will repeat for each type until the end of packet with “0” value is reached

B2. Driving Data Log Messages

B2.1 Charger sends CBDO (Charger request battery discharge onetime data) Request to the Battery. Battery replies with BBDO (Battery response battery discharge one time data). The corresponding BBDO (Battery request battery discharge onetime data) packet has

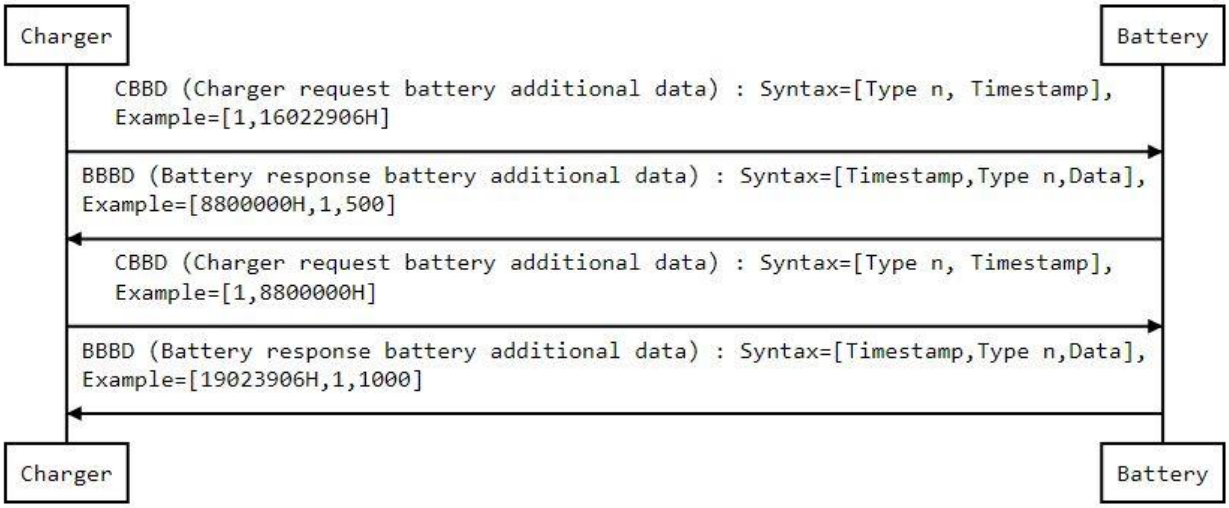


B2.2 Charger sends CBBL (Charger request Battery additional data list) Request to the Battery. CBBL Requests with value 0xAAH. The Battery replies with BBBL (Battery response battery additional data list) packet. The corresponding BBBL packet has Type n, Size, Granularity.



B2.3 Charger sends CBBD (Charger request battery additional data) Request to the Battery. CBBD Request has type n and Timestamp. The Battery replies with BBBD (Battery response battery additional data) packet. The corresponding BBBD packet has Timestamp, Type n and data.

For instance CBBD request type one with default timestamp (16022906H) and Battery replies with BBBD packet. The BBBD packet has timestamp (8800000H), type (one) and data (500). Next CBBD requests type one with timestamp as (8800000H) and Battery replies with BBBD packet. The BBBD packet has timestamp (19023906H), type (one) and data (1000).

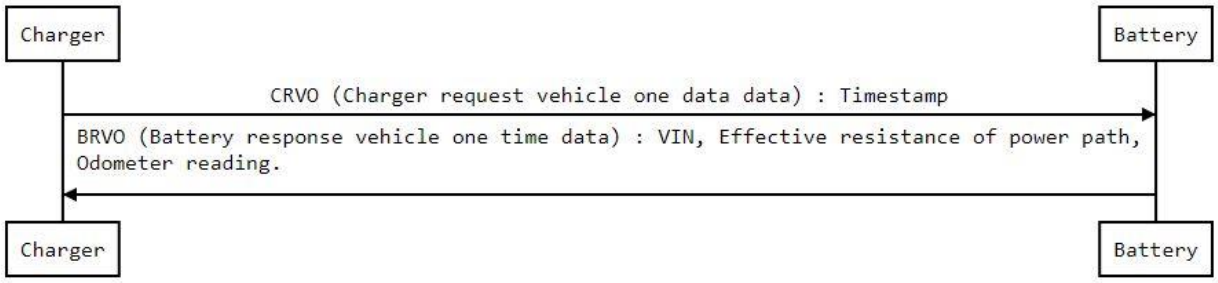


Similary the iterations will repeat for each type until the end of packet with “0” value is reached.

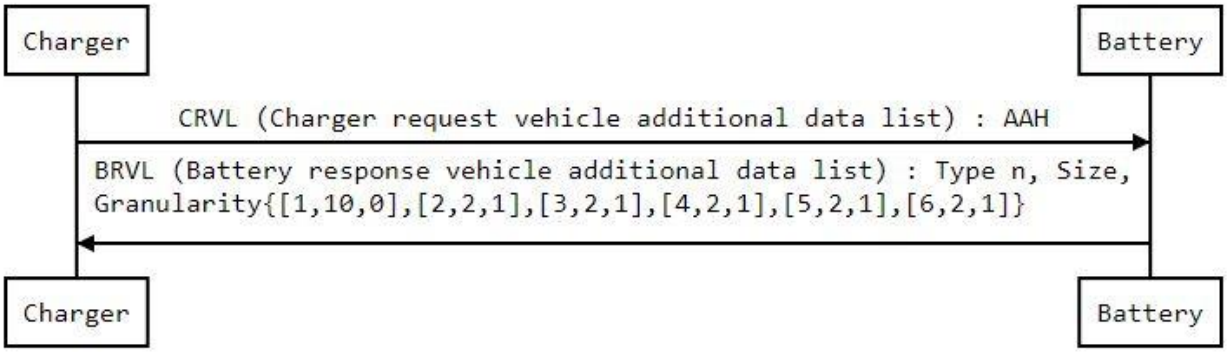
For battery data type 5 & 6 (BT5 and BT6), for a single request there would ‘n’ responses where ‘n’ equals the cell count and temperature sensor count for cell voltage and temperature sensor values respectively.

For e.g if there are 32 cells, then for type 5 request with timestamp1, there would be 32 responses. If ‘n’respons is not received then chanrger has to send the previous timestamp to retrive that record

B2.4 Charger sends CRVO (Charger request vehicle one data data) Request to the Battery. CRVO Requests with 0xAA value. The Battery responds with BRVO (Battery response vehicle one time data). The corresponding BBDP packet have

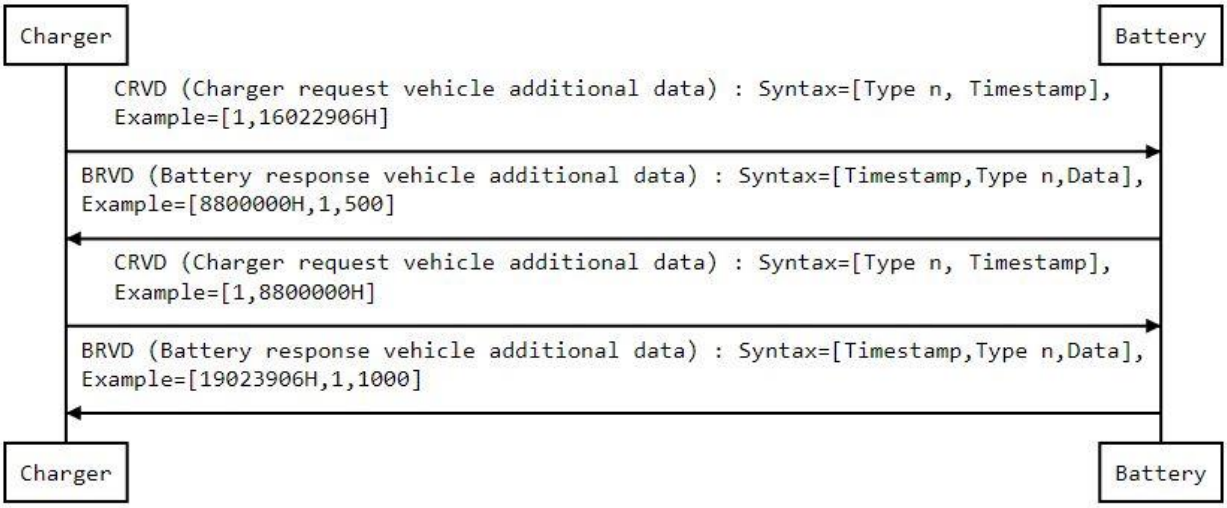


B2.5 Charger sends CRVL (Charger request vehicle additional data list) Request to the Battery. CRVL Requests with value 0xAAH. The Battery replies with BRVL (Battery response vehicle additional data list) packet. The corresponding BRVL packet has Type n, Size, Granularity. For instance, battery has 9 different size of data with different granularity.



B2.6 Charger sends CRVD (Charger request vehicle additional data) Request to the Battery. CRVD Request has type n and Timestamp. The Battery replies with BRVD (Battery response vehicle additional data) packet. The corresponding BRVD packet has Timestamp, Type n and data.

For instance, CRVD request type one with default timestamp (16022906H) and Battery replies with BRVD packet. The BRVD packet has timestamp (8800000H), type (one) and data (500). Next CRVD requests type one with timestamp as (8800000H) and Battery replies with BRVD packet. The BRVD packet has timestamp (19023906H), type (one) and data (1000).



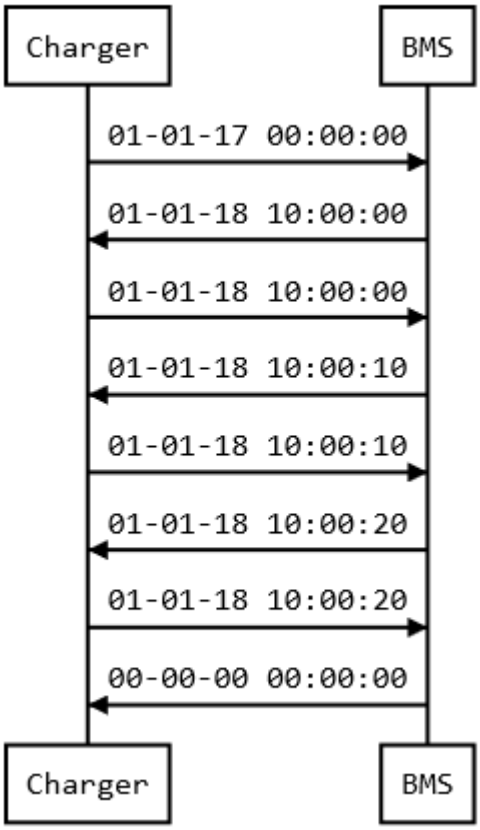
Similary the iterations will repeat for each type until the end of packet with “0” value is reached.

B3. Using Driving periodic data retrieval granularity and Charging periodic data retrieval granularity

The given table below provides the timestamps of data availability in BMS and the data which the BMS will have to send for different values of Driving and Charging periodic data retrieval granularity.

	Data retrieved for various granularity period in seconds		
BMS - Timestamps of data	Granularity-1	Granularity-5	Granularity-10
01-01-18 10:00:00	01-01-18 10:00:00	01-01-18 10:00:00	01-01-18 10:00:00
01-01-18 10:00:01	01-01-18 10:00:01	01-01-18 10:00:05	01-01-18 10:00:10
01-01-18 10:00:02	01-01-18 10:00:02	01-01-18 10:00:10	01-01-18 10:00:20
01-01-18 10:00:03	01-01-18 10:00:03	01-01-18 10:00:15	
01-01-18 10:00:04	01-01-18 10:00:04	01-01-18 10:00:20	
01-01-18 10:00:05	01-01-18 10:00:05		
01-01-18 10:00:06	01-01-18 10:00:06		
01-01-18 10:00:07	01-01-18 10:00:07		
01-01-18 10:00:08	01-01-18 10:00:08		
01-01-18 10:00:09	01-01-18 10:00:09		
01-01-18 10:00:10	01-01-18 10:00:10		
01-01-18 10:00:11	01-01-18 10:00:11		
01-01-18 10:00:12	01-01-18 10:00:12		
01-01-18 10:00:13	01-01-18 10:00:13		
01-01-18 10:00:14	01-01-18 10:00:14		
01-01-18 10:00:15	01-01-18 10:00:15		
01-01-18 10:00:16	01-01-18 10:00:16		
01-01-18 10:00:17	01-01-18 10:00:17		
01-01-18 10:00:18	01-01-18 10:00:18		
01-01-18 10:00:19	01-01-18 10:00:19		
01-01-18 10:00:20	01-01-18 10:00:20		

The same is shown below in a timeline diagram for granularity of 10 seconds

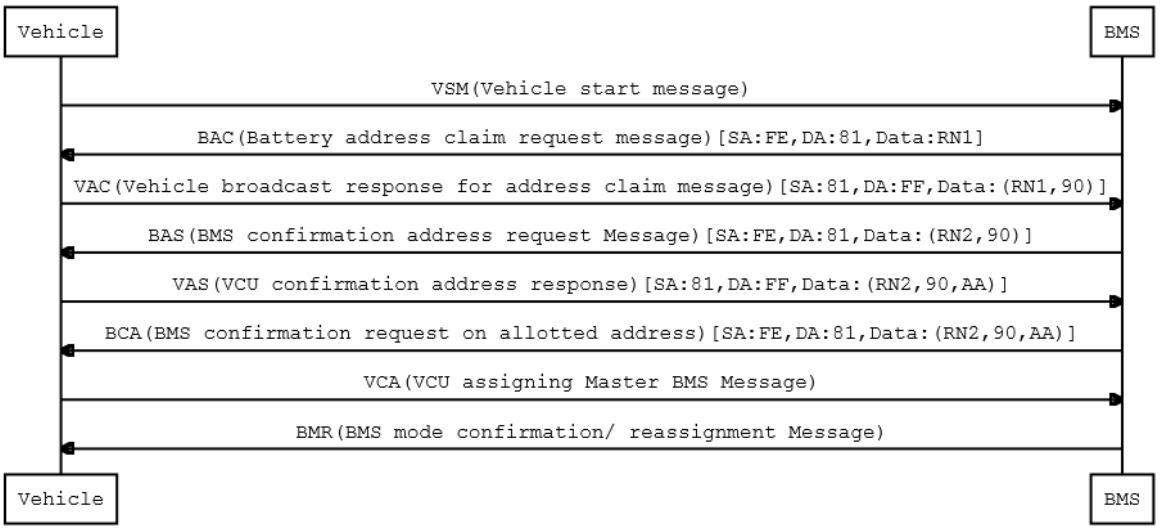

Charger
BMS
01-01-17 00:00:00
01-01-18 10:00:00
01-01-18 10:00:00
01-01-18 10:00:10
01-01-18 10:00:10
01-01-18 10:00:20
01-01-18 10:00:20
00-00-00 00:00:00
Charger
BMS

Annexure C: Auto-address assignment flow

C1. Driving protocol

The Vehicle sends a start message to BMS and on receiving VSM, BMS checks for address-assignment. If address is not assigned then it starts with address claim request (BAC) and the flow follows and ends at VCU assigning the Master among the BMS.This is repeated for each BMS.

If address is already assigned, then each BMS sends mode re-iterationmessage (BMR) to VCU and in turn VCU confirms the mode using VCA message,



C2. Charging protocol

In case of charging protocol, there will not be any start message from charger. BMS will complete address assignment by the procedure listed in the flow diagram below: