INTERFACE SPECIFICATION (IF)*

Document Number: 710-xxxx

Project: IC Series CANOpen Interface Specification

Document Revision: 2.0

CANOpen Version: 2.6

Approvals:

Issue Date	Rev	CAN Rev	Author	Description of Change
2013 Aug 08	0.1		David Gong	Draft Release
2013 Aug 27	0.2		David Gong	Updates after initial review
2013 Sep 09	0.3		Russell Lewis	Updates after customer review
2013 Sep 24	0.4		Peter Drexel	• Added PDO configuration • Added reset of elapsed times on algo change Added reset of Ah counts on algo change
2013 Sep 27	0.5		Peter Drexel	• Change Charge Status (2006h) to Extended Charge Status • Moved Override Status into Extended Charge Status • Clarified TPDO 4 • Added PDO 4 bits to battery module device type • Added some further use cases • Added descriptions for Ah elapsed, Voltage request and battery status objects
2013 Oct 2	0.6		Peter Drexel	• Added fifth charger PDO containing voltage, current and extended charger status • Fourth charger PDO now contains Ah count and elapsed time only • Added charger ready status bit to extended charger status • Updated field definitions in extended charger status • Added default TPDO transmit periods • Added note regarding byte order differences between EMCY message and object 1003h JIRA reference SSP-4514
2013 Oct 4	0.7		David Gong	• Added DC Relay Test Fault
2013 Oct 7	0.8		Tony Manhas	• Added PDO 5 to table on page 11
2013 Oct 24	0.9		Peter Drexel	• Swapped fields in TPDO 4 and TPDO 5
2013 Oct 28	0.10		Peter Drexel	• Updated TPDO 5 COB-Ids • Added version of CANOpen Interface specified
2013 Dec 05	0.11		Peter Drexel	• Clarified serial number encoding/decoding

2014 Jan 10	0.12		Tony Manhas	• Corrected couple of items
2014 Jan 14	0.13		Russell Lewis	• Added description for the Ah return target object
2014 Jan 14	0.14	1.2	Russell Lewis	• Added a note about object 2002 to define some limitations
2014 Mar 04	0.15		Tony Manhas	• Added bit for algorithm reset to permanent algorithm after AC loss to object 2241. Added description for battery disconnect leading to algorithm reverting back to permanent algo.
2014 Mar 14	0.16		Tony Manhas	• Added JIRA reference for correction in object 1003h
2014 Mar 17	0.17		Tony Manhas	• Modified the document number.
2014 Mar 19	0.18		Tony Manhas	• Corrected a minor error with E-0-2-4 in the error mapping table.
2014 Mar 20	0.19		Tony Manhas	• Added missing alarms to the the error mapping table.
2014 Mar 20	1.0		Russell Lewis	• Added details on battery low and high thresholds • Minor wording modifications
2014 Apr 10	1.1		Tony Manhas	• Updated the Visio diagrams
2014 Apr 17	1.2		Tony Manhas	• Grayed out TPDO 5
2014 Sep 29	1.3		Russell Lewis	• Clarified wording in Elapsed Time signal description (2005h)
2014 Oct 10	1.4	1.3	David Gong	• Added alarm #33/34/35/36 in error object section
2014 Nov 19	1.5	1.4	Russell Lewis	• Removed Alarm code E005 Charger over temperature – The charger does not produce this alarm
2014 Dec 10	1.6	2.0	David Gong	• Added AC Voltage Object 2200 • Added Battery Voltage in Object 2101 • Modified PDO4 to use object 2101 instead of 2100, which was limited in range to 0 - 64V • Removed greyed out PDO5 sections • Added AC Connected, derating active and HW shutdown status bits to extended status (2006 object)
2015 Jan 09	1.7	2.0	Peter Drexel	• Corrected COB-IDs for Client SDO

				• Restored TPDO4 definition and added range note for object 2100h
2015 Apr 17	1.8	2.1	I Stroud	• Added Objects 2242, 2243 and 2244 to access installed algorithm version information
2015 Apr 22	1.9	2.1	I Stroud	• Added support for dynamically reconfiguring Transmit PDOs
2015 Apr 23	1.10	2.1	I Stroud	• Amended algorithm version information to use installed algo index rather than internal slot number. • Added support for dynamically reconfiguring Receive PDOs
2015 May 11	1.11	2.1	P Drexel	• Added cumulative counters object and description
2015 Jun 25	1.12	2.1	P.Drexel	• Changed PDO numbering to match EDS file
2015 Aug 20	1.13	2.1	P.Drexel	• Added Cable resistance object 2275h • Added Authentication object 2400h
2015 Oct 05	1.14	2.2	P.Drexel	• Updated Object 1003h error codes • Added Object 2060h
2015 Oct 27	1.15	2.2	P.Drexel	• Corrected fan error and button fault codes
2015 Dec 7	1.16	2.3	P.Drexel	• Added charge history retrieval objects 2E00h, 2E01h and 2E02h
2015 Dec 23	1.17	2.3	I. Stroud	• Added information about CAN configuration error E042.
2015 Dec 23	1.18	2.3	R Lewis	• Modified alarms to updated CANOpen error codes
2016 Jan 22	1.19	2.3	R Lewis	• Removing 2E02h as not required by customer
2016 Feb 12	1.20	2.4	P.Drexel	• Added Transmit/Receive PDO CRCs • Added Transmit/Receive PDO Sequence Counts
2016 Feb 22	1.21	2.4	T. Prettejohn	• Added descriptions for current charge data objects (200Bh through 2011h) • Added example graphs for current charge objects
2016 Feb 24	1.22	2.4	P.Drexel	• Grammatical corrections
2016 Mar 1	1.23	2.4	R Lewis	• SSP-5511 – Fixing length of 2271 in PDO mapping table
2016 Mar 3	1.24	2.4	P.Drexel	• Added PDO CRC and sequence count error alarms

2016 Mar 3	1.25	2.4	A. Midwinter	• Added ability to configure PDO communication parameters via NVS
2016 Mar 9	1.26	2.4	A. Midwinter	• Added battery disconnection alarm
2016 Mar 14	1.27	2.4	T. Prettejohn	• Changed state of charge – finishing from 95% to 90%
2016 Apr 11	1.28	2.4	T. Prettejohn	• Added latching behaviour for state of charge (6080h)
2016 Apr 12	1.29	2.4	T. Prettejohn	• Added Fault 006
2016 Apr 18	1.30	2.4	T. Prettejohn	• Added Wh counters (object 200Ah) • Updated behaviour for Ah counters (2009h) and Cycle Type (200Bh) for consistency
2016 May 6	1.31	2.4	A. Midwinter	• Added objects for reading charger mode (0x2250) and configuration number (0x2251), and requesting a SW reset (0x4000)
2016 May 13	1.32	2.5	P.Drexel	• Added object(s) for log upload
2016 Jun 2	1.33	2.5	P.Drexel	• Added invalid PDO length alarm
2016 Jun 8	1.34	2.5	T. Prettejohn	• Added objects for RX & TX error counters (2600h, 2601h)
2016 Jun 22	1.35	2.5	A. Midwinter	• Changed object 2500h to provide up-to-date cumulative counter values.
2016 Aug 9	1.36	2.5	P.Drexel	• Added object 2276h – voltage request 16-bit, Q8
2016 Aug 22	1.37	2.5	P.Drexel	• Added CANopen standard objects 1029h and 1F80h
2016 Sep 09	1.38	2.5	P.Drexel	• Updated objects 1F50h and 1F57h per actual implementation.
2016 Nov 21	1.39	2.5	A. Midwinter	• Added description of part number encoding from new serial number format.
2017 Oct 11	1.40	2.5	P.Drexel	• Clarified “Program Control – 1F51h” section
2018 Jan 08	2.0	2.6	P.Drexel	• Removed all object descriptions; Added them to EDS/HTML files • Updated local/remote mode descriptions; Added references to application notes • Removed program download details; added reference to reprogramming application note

				• Added reference to charge history record retrieval application note

Table of Contents

Table of Figures

1 Introduction

1.1 Purpose

This Interface specification details version 1.3 of the CANopen interfaces for the IC Series Charger Family. This document needs to be read alongside the CANopen Object Dictionary.

1.2 Scope

The scope of this document is to specify the CAN Interface for the charger. It shall define the protocol, signals and functionality that can be achieved through the interface along with configuration items that effect the behaviour.

1.3 Audience

This document is solely for use by engineering staff and management within Delta-Q Technologies and customers who want to interface with the IC Series product family using CANopen. It must not be distributed outside these organizations without prior written permission from Delta-Q Technologies

1.4 Definitions and Acronyms

Term	Definition
AC	Alternating current
Ah	Ampere · hours
Alarm	Alarms are typically external, user recoverable conditions that are preventing charging such as bad AC.
Battery Connection Cycle	Starts from the time the battery is connected to the charger and conditions are valid to charge till the time conditions are deemed not OK to charge
CANopen	CAN-based communications standard developed by the CiA
CiA	CAN in Automation
Charge Cycle	Starts from the time the battery is connected to the charger and conditions are valid to charge till the charge cycle completes or conditions are deemed not OK to charge whichever happens sooner.
COB-ID	Communication Object Id
DC	Direct current
Fault	Faults are internal charger errors that prevent charging

Term	Definition
NMT	Network Management
Not OK to Charge	AC removed for > configurable time (Default 10mins) or Battery disconnected Or Algorithm changed.
OSI	Open Systems Interconnection
PDO	Process data object
SDO	Service data object
Valid Charge Conditions	AC present, Battery Detected, in correct charger mode and not internal charger fault.

2 References

2.1 Reference Documents

Ref	Title
[DQT_CONF_CTR]	810-0003 (Configuration Control).pdf
[DQT_CHANGE_CTRL]	810-0022 (Change Management).pdf
[DQT_DEV_PROCESS]	810-0013 (Development Process).pdf
[CHARGER_EDS]	710-0185_EDS_IC_SERIES_CANOpen_V2.0
[CHARGER_EDS_HTML]	710-0186_HTML GLOSSARY_IC_SERIES_CANOpen_V2.0
740-0022	Application Note: Remote Control using CANopen
740-0024	Application Note: Local Mode using CANopen
740-xxxx	Application Note: Reprogramiming the Charger over CANopen
740-xxxx	Application Note: Charge History Record Retrieval

2.2 Notation

The following methods of notation are used:

Format	Meaning
1234h	Hexadecimal value
code [courier font]	Source code element (variable, function, macro, etc).
0101b	Binary value
123d	Decimal Value

All arrays are referenced via an index ranging from 0 to (length-1) (as opposed to 1 to length), unless otherwise stated.

When referencing bits within a byte, the most significant byte is labelled bit7, the least significant bit0.

3 Charger Configuration

There are a number of configuration parameters which affect how the charger behaves when controlled via CANopen. This section specifies those configuration items with the aim of providing enough information that customers reading this document who know what their charger's parameters are set to will understand the intended functionality of their charger.

3.1 Operating Mode Configurations

3.1.1 The charger can be configured to be in one of two different operating modes when under CANopen control and a further mode for monitoring only.Remote Control Mode

In remote control mode, the charger requires the presence of an external Battery Management System to start, stop and generally control the charging process.

3.1.2 For more details please refer to Application Note: 740-0022740-0022.Local Mode

In local mode, the charger normally has full control of the charge cycle. An external BMS may be connected. Usually the BMS will just monitor the charge cycle. However, the BMS also has the ability to override the charging operation should it choose to do so.

Local Override Mode

For details of the charger's behaviour when operating in Local Override mode please refer to Application Note:740-0024.

Monitoring

The purpose of this mode is to allow other CANopen devices to monitor the charging parameters.

4 External Interface Design

4.1 Interface Architecture

The vehicle controller, battery module and Delta-Q charger shall follow CiA 301, 418 and 419, which defines most of the interface, including all of the first six layers (Physical, Data Link, Network, Transportation, Session, and Presentation) of the OSI model, and most of the seventh layer (Application). This document fills out additional object details in the application layer.

In addition to the standard protocol the charger has support for

- Voltage set point Control
- Algorithm selection
- Setting an Ah target
- Reading the sensed current
- Reading charge cycle status information

4.2 Context Diagram

The typical use of the charger in this context is off-board, always connected to AC but it also supports on board applications. The user drives a vehicle to the charging station and plugs the charger into the vehicle. There are typically multiple chargers and multiple vehicles and there is no guarantee as to which charger is connected to which vehicle.

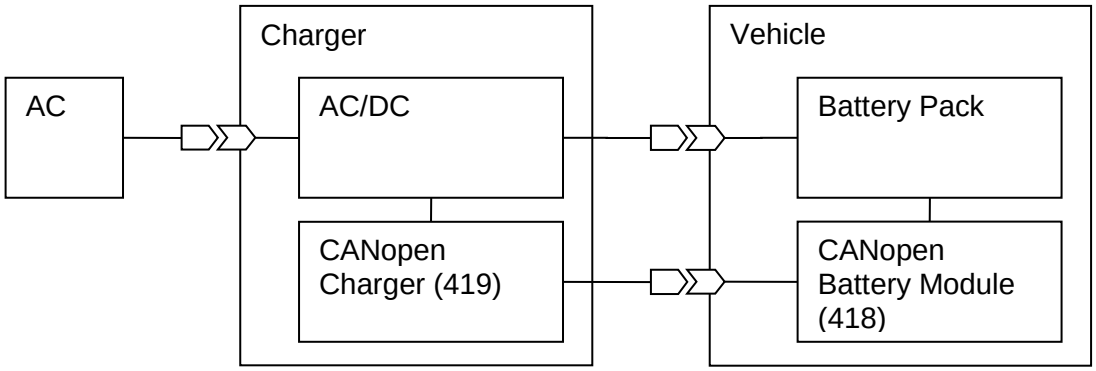


Figure 1 Context Diagram

4.3 CAN bus Configuration

IC Series charger role:	slave of the master/slave model.
# Of bits in Identifier:	11 bits.
Bit-rate:	125 kbps (configurable via USB)
Node ID of the charger:	10 (configurable via USB)
Node ID of the battery node:	1 (configurable via USB)
Use Dynamic SDO and SDO Manager:	FALSE (configurable via USB)
SDO COB-ID:	60B(rx) and 58B(tx) “Use Dynamic SDO…” is FALSE

PDO COB-ID: Always requested from the battery module via SDO.

The charger has been configured with a 2 second heartbeat consumer timeout. The charger has been configured with a 1 second heartbeat producer period.

4.3.1 Recommended Timeouts

Parameter	Timeout Value	NOTE
Boot up	5000 ms	
NMT	2000 ms	
SDO	2000 ms	2000ms functions correctly but is being optimised
PDO Cycle Time	TBD ms	2000ms functions correctly but is being optimised
Objects 1010h/1011h	2000ms	
SYNC cycle time	N/A	Charger does not support SYNC function
Charger Monitoring Heartbeat Timeout	2000ms	The charger produces HB messages every 1000ms

Table 1 Recommended Timeouts

4.3.2 Default PDO and SDO COB-IDs

The following table shows the default COB-IDs for the charger and battery module.

PDO/SDO	Object Index	Object Sub-index	Base COB-ID	Charger w/Node Id COB-ID	Battery Module w/Node Id COB-ID
Server SDO: Client -> Server	1200h	1	600h	60Ah	601h
Server SDO: Server -> Client	1200h	2	580h	58Ah	581h
Client SDO: Client -> Server	1280h	1	600h	60Ah	601h
Client SDO: Server -> Client	1280h	2	580h	58Ah	581h
RPDO 0	1400h	1	200h	20Ah	201h
RPDO 1	1401h	1	300h	30Ah	301h
RPDO 2	1402h	1	400h	40Ah	401h
RPDO 3	1403h	1	500h	50Ah	501h
TPDO 0	1800h	1	180h	18Ah	181h
TPDO 1	1801h	1	280h	28Ah	281h
TPDO 2	1802h	1	380h	38Ah	381h
TPDO 3*	1803h	1	480h	48Ah	481h

* The charger is not expecting to receive PDO 3.

4.4 Error Objects (EMCY and Object 1003h)

4.4.1 Emergency Write

The charger's faults and alarms are mapped to a subset of CiA 301 generic errors, and the Delta-Q specific fault or alarm code is stored in the first two bytes (16-bits) of the manufacturer-specific error code, which is used by the EMCY object and the object 1003h. A fault is an internal charger error condition (i.e., hardware issue). An alarm is a user recoverable condition. NOTE: Due to an idiosyncrasy in Delta-Q's CANopen stack, the 16-bit Delta-Q alarm/fault codes are stored in little-endian order in the EMCY message and big-endian order in the 1003h object.

The charger is configured to store the last 10 errors in a FIFO buffer.

Alarms have mostly been assigned the CiA error code of external error (9000_h) exceptions for charger temperature and charger initialisation alarms. Faults have all been assigned the CiA error code of CANopen device hardware (5000_h).

Delta-Q has its own numbering scheme for alarms and faults. The Delta-Q alarm and fault numbers appear in the most significant word of each 1003_h sub index. Fault numbers are prefixed with a C_h in the top most nibble and error numbers are prefixed with an 8_h in the top most nibble.

The following table illustrates the mapping and expected error code values.

Charger Code(s)	Description	CiA301 EMCY Code(s)	Object 1003h Error Code
E-0-0-1 E-0-2-1	Battery high voltage – Threshold as set by the algorithm profile in operation.	9000h External error – generic	01809000h 15809000h
E-0-0-2 E-0-2-2	Battery low voltage - Threshold as set by the algorithm profile in operation.	9000h External error – generic	02809000h 16809000h
E-0-0-3	Charge timeout caused by battery pack not reaching required voltage within safe time limit.	9000h External error – generic	03809000h
E-0-0-4	Battery could not meet minimum voltage	9000h External error – generic	04809000h
E-0-0-7	Battery amp hour limit exceeded	9000h External error – generic	07809000h
E-0-0-8	Battery temperature is out of range	9000h External error – generic	08809000h
E-0-1-2	Reverse polarity error	9000h External error – generic	0C809000h
E-0-1-3	Battery does not take current	9000h External error – generic	0D809000h
E-0-1-4	Number of Cells Invalid	1000h Generic error	0E801000h
E-0-1-5	End of Charge Voltage Not In Algo	1000h Generic error	0F801000h
E-0-1-6	Upgrade Failed	0000h error	10800000h*
E-0-1-7	USB error	0000h error	11800000h*
E-0-1-8	Storage error	0000h error	12800000h*
E-0-1-9	Incompatible software error	0000h error	13800000h*
E-0-2-0	Active Algo Not Set	1000h Generic error	14801000h

Charger Code(s)	Description	CiA301 EMCY Code(s)	Object 1003h Error Code
E-0-2-3	High AC voltage error (>270VAC)	9000h External error – generic	17809000h
E-0-2-4	Charger failed to initialize	1000h Generic error	18801000h
E-0-2-5	Low AC voltage oscillation error	9000h External error – generic	19809000h
E-0-2-6	USB Script Error	0000h error	1A800000h*
E-0-2-7	USB Over Current	0000h error	1B800000h*
E-0-2-8	Incompatible algorithm error	1000h Generic error	1C801000h
E-0-2-9	Communication CAN-bus error	9000h External error – generic	1D809000h
E-0-3-0	Communication battery module error	8130h Monitoring – Comms – Heartbeat Error	1E808130h
E-0-3-1	Reference out of range error	1000h Generic error	1F801000h
E-0-3-2	Communication heartbeat lost error	8130h Monitoring – Comms – Heartbeat Error	20808130h
E-0-3-3	Target voltage configuration too high	1000h Generic error	21801000h
E-0-3-4	Battery capacity configuration not set	1000h Generic error	22801000h
E-0-3-5	Target voltage configuration too low	1000h Generic error	23801000h
E-0-3-6	Battery temperature sensor not installed	9000h External error – generic	24809000h
E-0-3-7	CAN Download Failed	6000h SW Generic error	25806000h
E-0-3-8	Fan error	9000h External error – generic	26809000h
E-0-3-9	Button stuck down	1000h Generic error	27801000h
E-0-4-0	Fan Supply Voltage Low	1000h Generic error	28801000h
E-0-4-1	Software Internal Error	6000h SW Generic error	29806000h
E-0-4-2	CAN Configuration Error	6000h SW Generic error	2A806000h
E-0-4-3	PDO CRC Error	9000h External error – generic	2B809000h
E-0-4-4	PDO Sequence Count Error	9000h External error – generic	2C809000h
E-0-4-5	Battery Disconnected Alarm	9000h External error - generic	2D809000h
E-0-4-6	Invalid PDO Length	8210h Monitoring – Protocol – PDO Length Error	2E808210h

* These items will never appear in Object 1003h, but will appear in Object 2060h.

The following table illustrates the mapping and expected fault code values.

Charger Code(s)	Description	CiA301 EMCY Code(s)	Object 1003h Error Code
F-0-0-1	Output Stage Error	5000h CANopen Device Hardware	01C05000h
F-0-0-2 F-0-0-3	Input Stage Error	5000h CANopen Device Hardware	02C05000h 03C05000h
F-0-0-4	Current Measurement Error	5000h CANopen Device Hardware	04C05000h
F-0-0-5	DC Output Relay Test Error (High voltage across closed relay)	5000h CANopen Device Hardware	05C05000h
F-0-0-6	Output Current Error	5000h CANopen Device Hardware	0x500006C0

4.4.1.1 Emergency Write Structure

The following illustrates the frame structure for the emergency write.

EEC0	EEC1	ER	MSef 0	MSef 1	Not used	Not used	Not used

Where:
EEC is the CiA Emergency Error Code
ER is the Error Register
MSef is the Delta-Q 16 bit Error or Fault Code

4.4.2 Emergency Consumer

The charger shall also consume emergency events from the battery module. The behaviour varies slightly between a charger configured in local mode versus a charger configured in remote mode.

In both local and remote modes the handling of the special CiA 418 device profile specific error code 5010h (Temperature sensor failure) is different to the other error codes. This behaviour is specified in section **Error! Reference source not found. Error! Reference source not found.**.

For the non-temperature sensor failure error codes in local mode the charger shall terminate charging if the battery control module sends any error code other than the 8XXXh codes.

For the non-temperature sensor failure error codes in remote control mode the charger shall shutdown on receiving any error code.

These error codes shall be logged into datlog as and when they arrive.

4.5 Handshaking

4.5.1 Heartbeat Consumer

The charger shall “listen” for the battery module’s heartbeat message to determine if it is present on the network or not. The charger shall assume that the battery module is not on the network if it does not receive a heartbeat within the timeout specified in section 4.3. Whenever the charger detects a loss of connection, it shall re-configure its PDO COB-IDs upon reconnection, as described below. The charger shall support the heartbeat consumer object 1016h as described in CiA 301.

4.5.2 Heartbeat Producer

The charger shall transmit its heartbeat message using the node id and period in section 4.3. The charger shall support the heartbeat producer object 1017h as described in CiA 301.

4.5.3 PDO Producer

At start-up, the charger’s transmit PDOs (TPDOs) shall be disabled and not transmitted even if the charger is in the NMT operational state. The TPDOs shall be transmitted after their COB-IDs are configured (see below) and the charger has been commanded into the NMT operational state.

4.5.4 SDO Manager and Dynamic SDOs

By default, the charger shall expect the battery module to use a pre-configured node id (see section 4.3) and it shall NOT use dynamic SDOs to communicate with the battery module. However, if configured to use dynamic SDO (see section 4.3), the charger shall search for a node with device profile 418d (i.e., battery module) and register with an SDO Manager to acquire a dynamic SDO connection to the battery module per CiA 305. Subsequently, the charger shall configure its PDO COB-IDs as described below.

4.5.5 PDO COB-ID Configuration

CiA 419 “Device Profile for Battery Chargers” section 7.2 “Functional Description” specifies that the charger shall configure its PDO COB-IDs to correspond to those of the battery module. To do this, the charger shall send an SDO request for the battery module’s device type (1000h) and examine the response to determine which PDOs it supports as shown in the table below. The charger shall request COB-ID’s for RPDO 0 and TPDO 0 by default. TPDO 3 is a manufacturer specific PDO and is not part of the CIA 419 Profile. RPDO3 is not currently used by the charger.

Bits 31 – 22	Bit 21 Deprecated	Bit 20 RPDO 3 Supported	Bit 19 TPDO 2 Supported	Bit 18 TPDO 1 Supported	Bit 17 RPDO 2 Supported	Bit 16 RPDO 1 Supported	Bits 15 – 0 Device Profile Number
0h	N/A	1 – Y 0 – N	1 – Y 0 – N	1 – Y 0 – N	1 – Y 0 – N	1 – Y 0 – N	418d

For each supported PDO, the charger shall send an SDO request for the battery module communication parameter object’s (see table below for object index) COB-ID (sub-index 1h) and set its corresponding COB-ID to be the same. The COB-ID settings shall be volatile.

Charger PDO	Transmit Period (milliseconds)	Charger Communication Parameter Object Index	Battery Module PDO	Battery Module Communication Parameter Object Index
TPDO 0	200	1800h	RPDO 0	1400h
TPDO 1	200	1801h	RPDO 1	1401h
TPDO 2	200	1802h	RPDO 2	1402h
TPDO 3*	1000	1803h	RPDO 3*	1403h
RPDO 0	NA	1400h	TPDO 0	1800h
RPDO 1	NA	1401h	TPDO 1	1801h
RPDO 2	NA	1402h	TPDO 2	1802h
RPDO 3	NA	1403h	TPDO 3+	1803h

* The charger shall send PDO 3 if the battery module’s device type requires it.

+ The charger is not expecting to receive PDO 3 from the battery module.

Refer to section 10 for diagrams depicting the above process.

4.6 PDO Definitions

4.6.1 Default Configuration

It is possible to tell the charger to start transmitting its PDO messages via setting its NMT state to operational (see sections 4.5.3 and 4.5.5). This is standard CANopen functionality.

CiA 419 defines the three transmit PDO messages PDO0, PDO1 and PDO2. Additionally, Delta-Q has specified a further PDOs with the following information

Table 2: Charger TPDO 4

Field	Object Index	Object Sub-index
Charger Current	2002h	0
Battery Voltage Q8 UINT16	2100h*	0
Extended Charger Status	2006h	0

* On a 48V charger the range of this field may be exceeded. Use an SDO to read object 2101h or 6060h for an accurate voltage.

In addition there are four receive PDOs defined. Receive PDOs 0-2 are configured by default to match the requirements of CiA 419 “Device Profile for Battery Chargers”. PDO3 is currently disabled by default. Each transmit and receive PDO may be enabled and configured as required either dynamically (as specified below) or statically. For static configuration please contact Delta-Q.

4.6.2 Dynamic Reconfiguring the PDOs

All transmit and receive PDOs, PDO0 – PDO3, can be configured dynamically to each carry up to 8 separate variables. (Redefining PDO0, PDO1 or PDO2 will break compatibility with CiA 419). The total combination of variables carried by one PDO must however be no more than 8 bytes (64 bits).

To redefine a transmit PDO the following procedure should be followed:

1. Disable the PDO by setting bit 31 in the corresponding Communication Parameter COB-ID variable (Index 180x, Sub Index 1, where x = 1 - 4)
2. Set the Number of mapped objects to 0 (Index 1A0x, Sub Index 0). The mapping entries (Sub Index 1 to 8) can now be edited.
3. Update the mapping entries as required. Only variables that are flagged as “PDO Mapping” are allowed. Attempts to write an illegal value will be rejected. The mapping entry is a 32-bit value with the following fields:
   - Bits 31:16 Variable Index
   - Bits 15:8 Sub Index
   - Bits 7:0 Size of variable in bits
4. Set the Number of mapped objects to the required number between 1 and 8. The number set will be rejected if:
   - Number of entries was not originally 0 (all changes require the number of entries to be set to zero first)
   - Number of entries > 8
   - The included mappings include uninitialized (zero) entries

- The total bit count of the all selected entries is greater than 64

5. Enable the PDO by clearing bit 31 in the corresponding Communication Parameter COB-ID variable (Index 180x, Sub Index 1)

A receive PDO can be redefined in a similar manner only using indexes 0x140x and 0x160x.

5 Program Download

6 The program download feature of the Delta-Q charger is based upon the CiA Specification 302-part 3. See Application Note: 740-xxxx for detailed information.Charge History Record Retrieval

See Application Note: 740-xxxx for detailed information.

7 Example Use Case Sequence Diagrams (STILL IN DEVELOPMENT)

This Section aims to go through several use cases of using the charger when configured in the different modes.

The following configuration scenarios will be covered

- Local Mode with CAN Hold-off, Using Ah Target to Complete Charge, with a different default algo if No CAN available.
- Local Mode with CAN Hold-off, Using Normal Algorithm Completion[1].
- Remote mode

For each of these configurations the following Use Cases will be illustrated.

Use Case Name	Use Case description	Reference
	Normal Charge	
	Restart Charge – Via Set Active Algo	
	Complete Charge – Via Set Active Algo to Float 2241h	
	Terminate Charge – Via Battery Ready Status Object 6000h	
	Reduce Current and Voltage	
	Lose Battery Controller – before charge starts	
	Lose Battery Controller – while charging	
	CAN Bus Error	
	Never get CAN	
	Change Algo to a different algo	
	Set Ah target	

[1] Unless other commands are issued to terminate charge or to complete such as setting the algo to the float algorithm.

7.1.1 Normal Operation

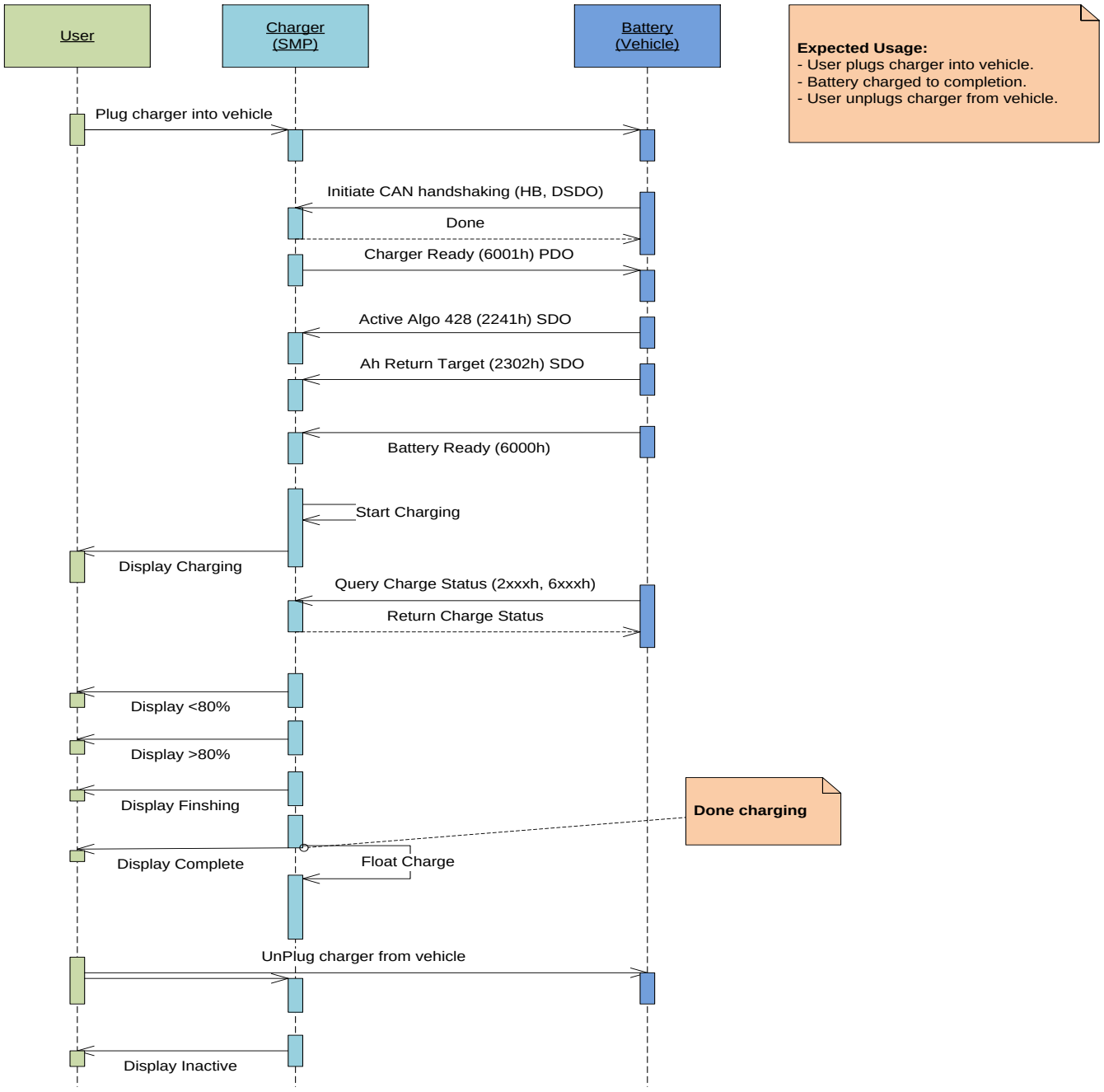


Figure 2: Sequence Diagram of Normal Operation

7.1.2 Stop Charge and Float

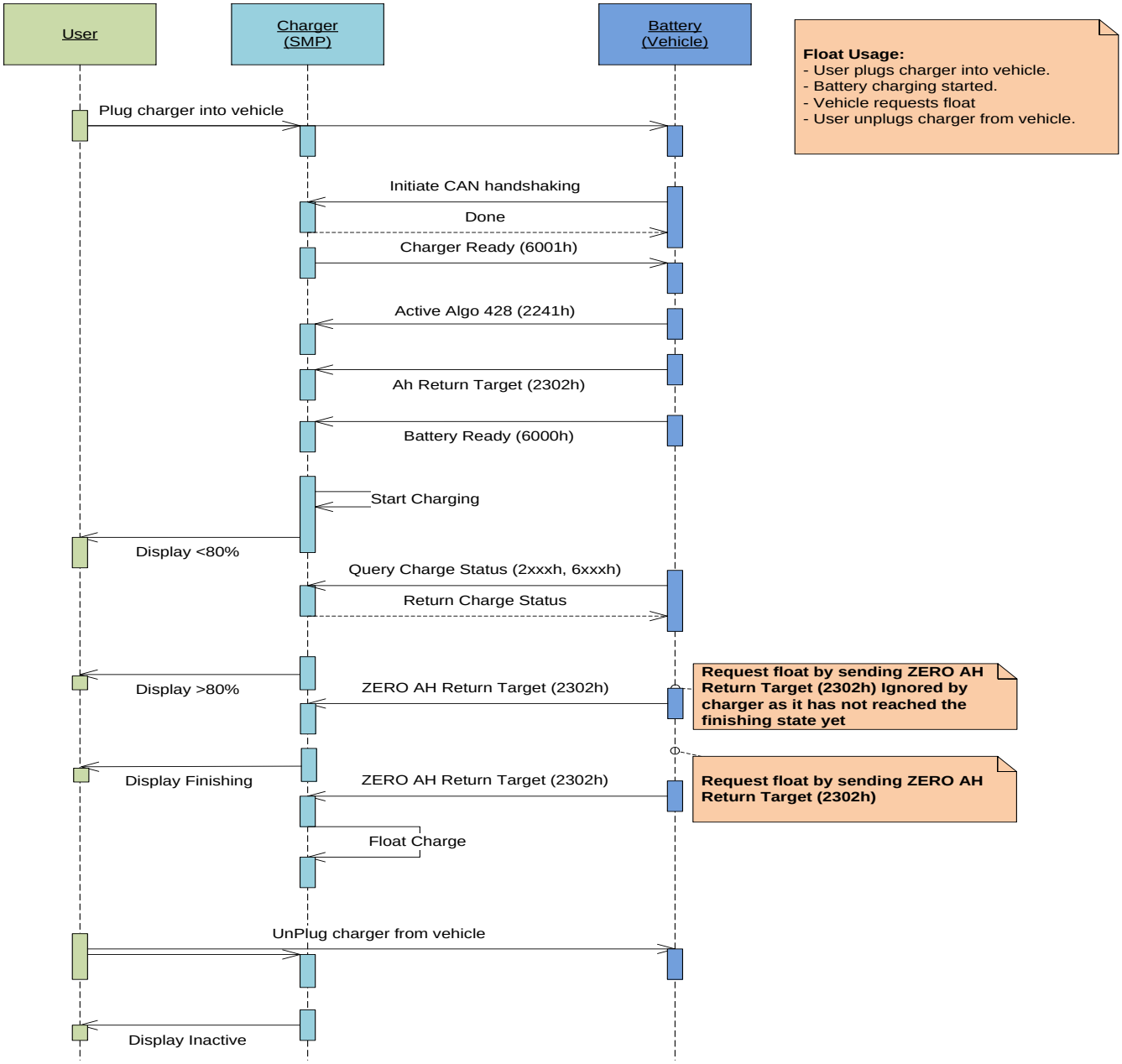


Figure 3 Sequence Diagram of Stop and Float

7.1.3 Terminate Charge Cycle

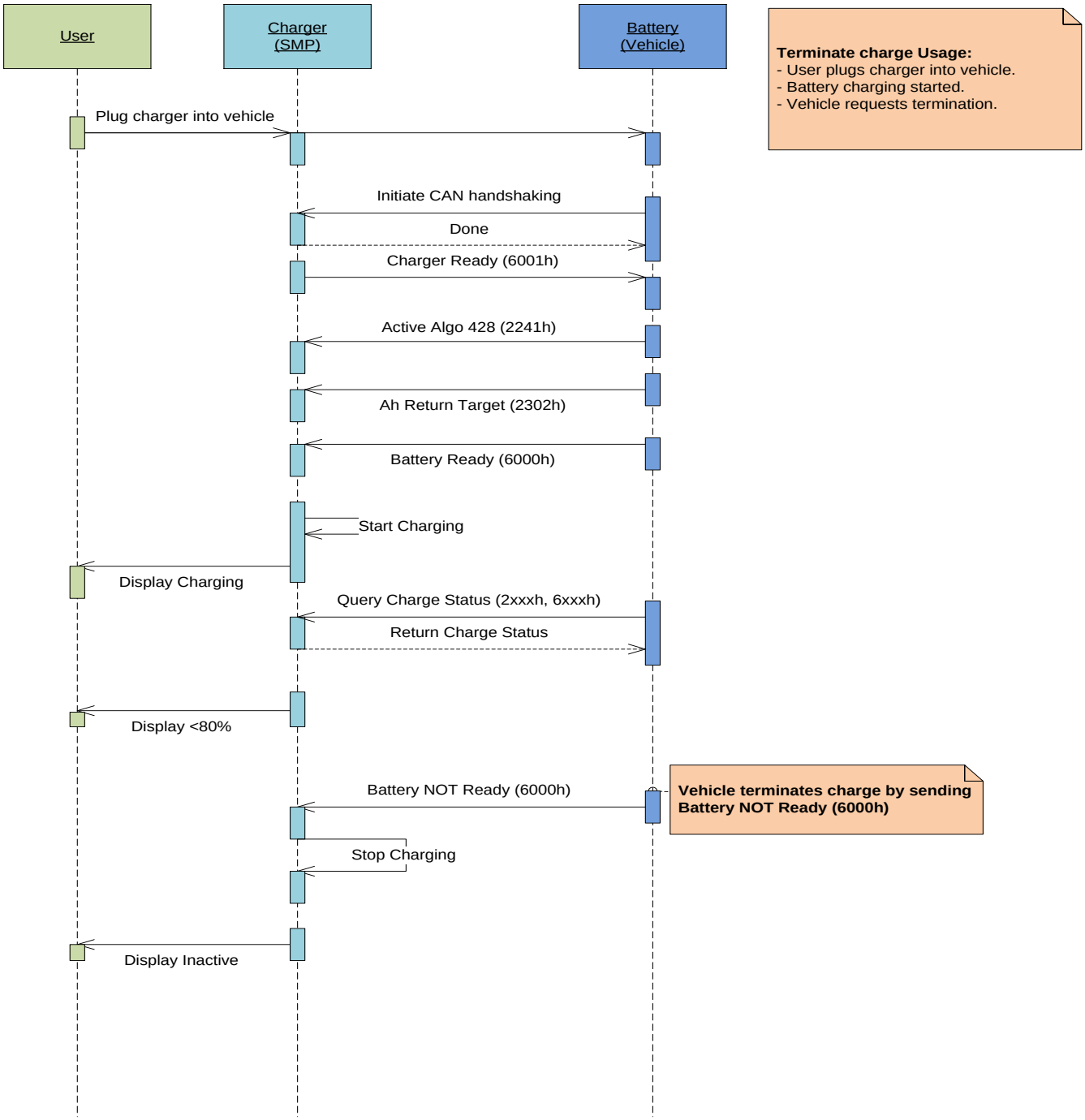


Figure 4 Sequence Diagram of Terminate Charge

7.1.4 Communication Failure Default

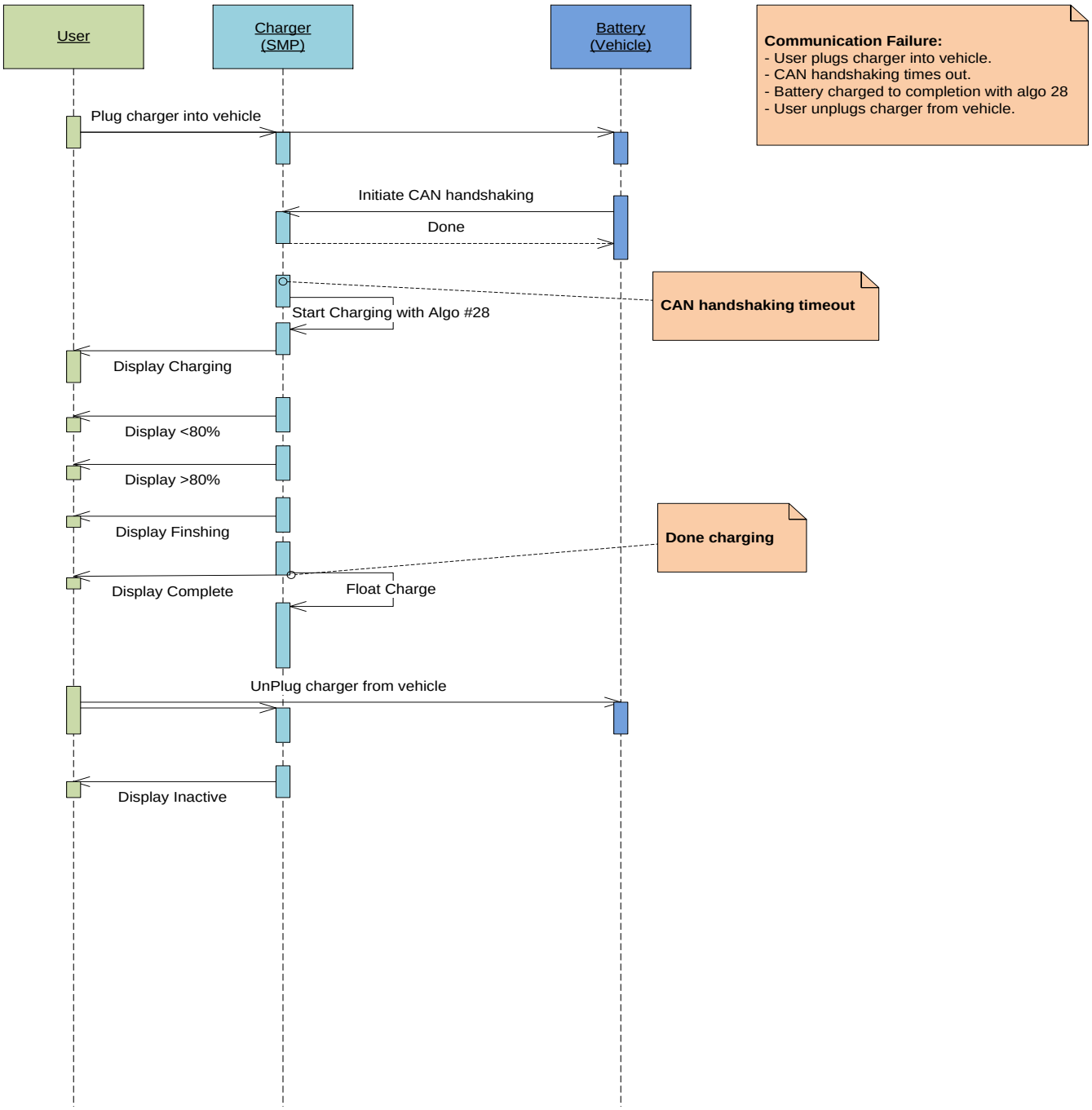


Figure 5 Sequence Diagram of Communication Failure and Default

7.1.5 Restart Charge Cycle

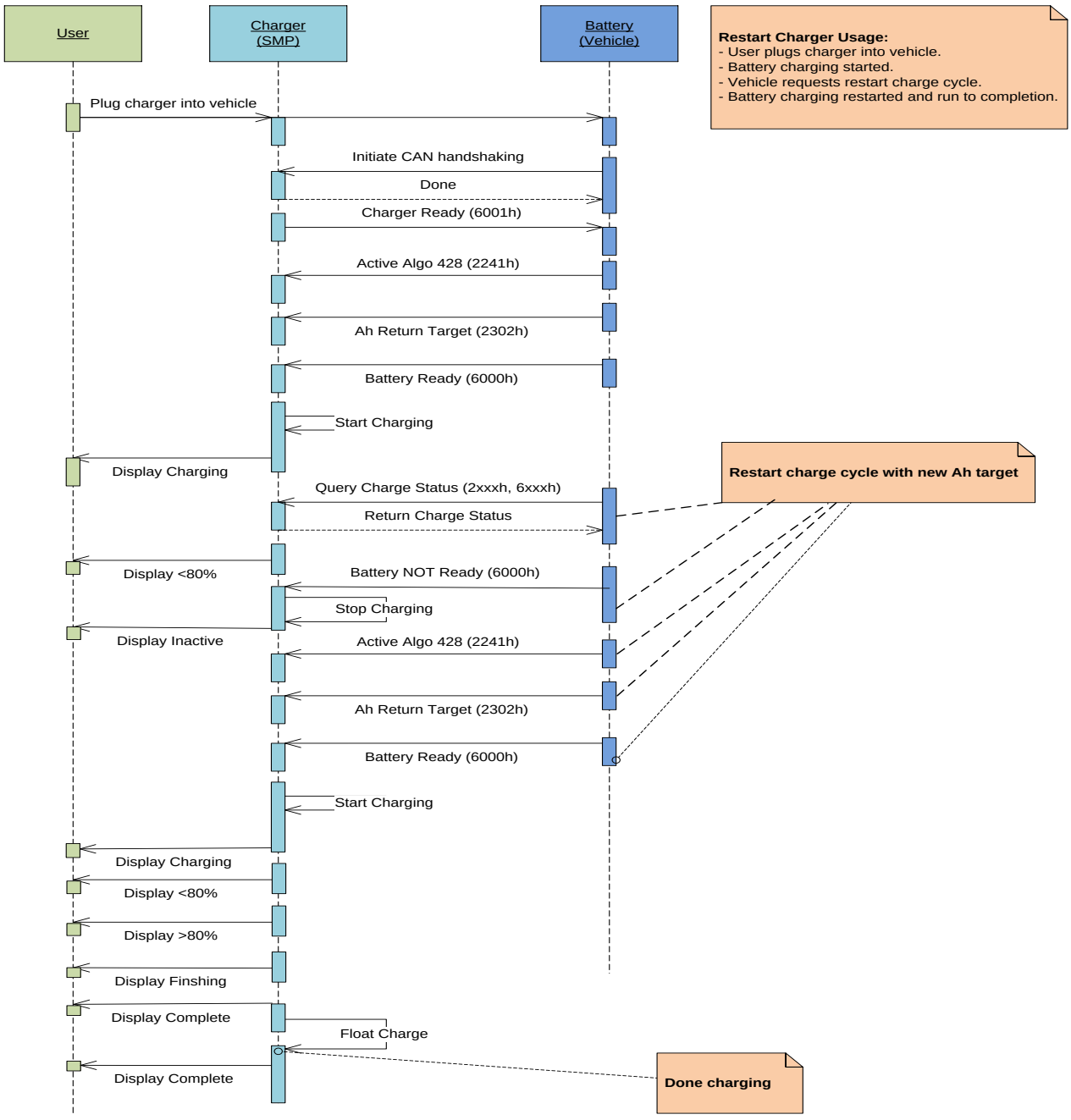


Figure 6 Sequence Diagram of Reset Charge Cycle

7.1.6 Communication Failure after Charge Termination

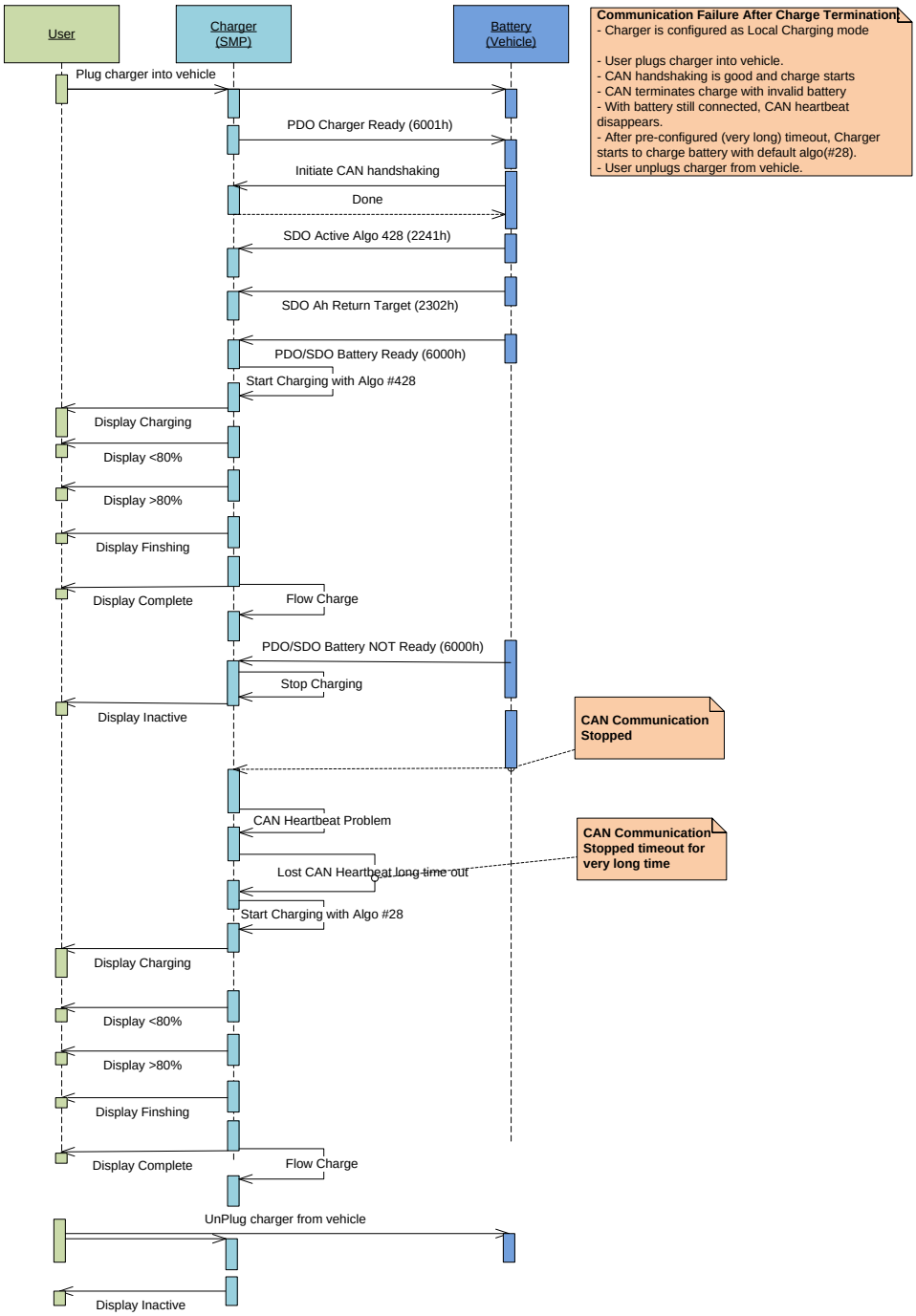


Figure 7 Sequence Diagram: Communication Failure after Charge Termination

7.1.7 Communication Restored

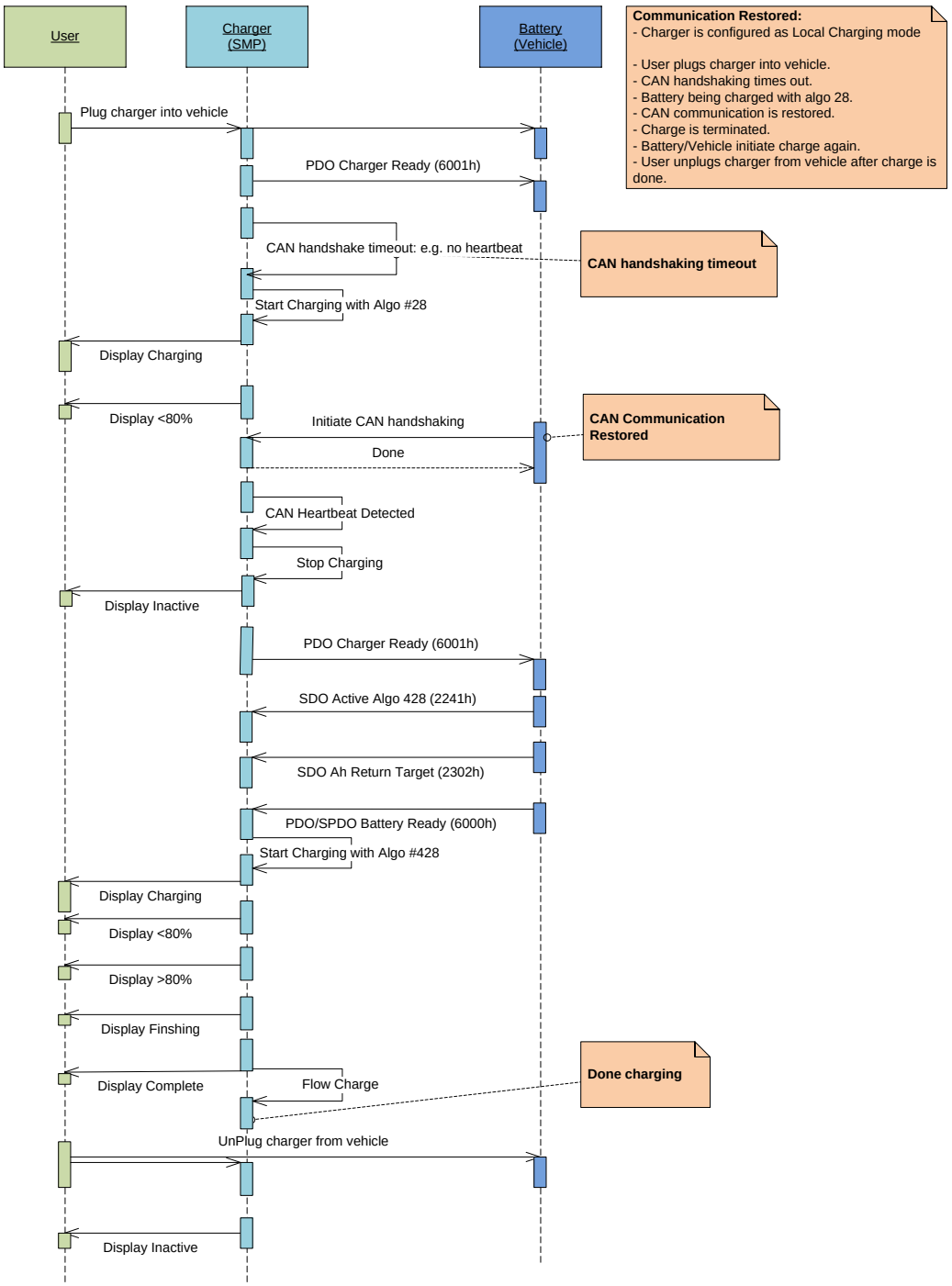


Figure 8 Sequence Diagram: Communication Restored

7.1.8 CAN Connected with Battery Unplugged

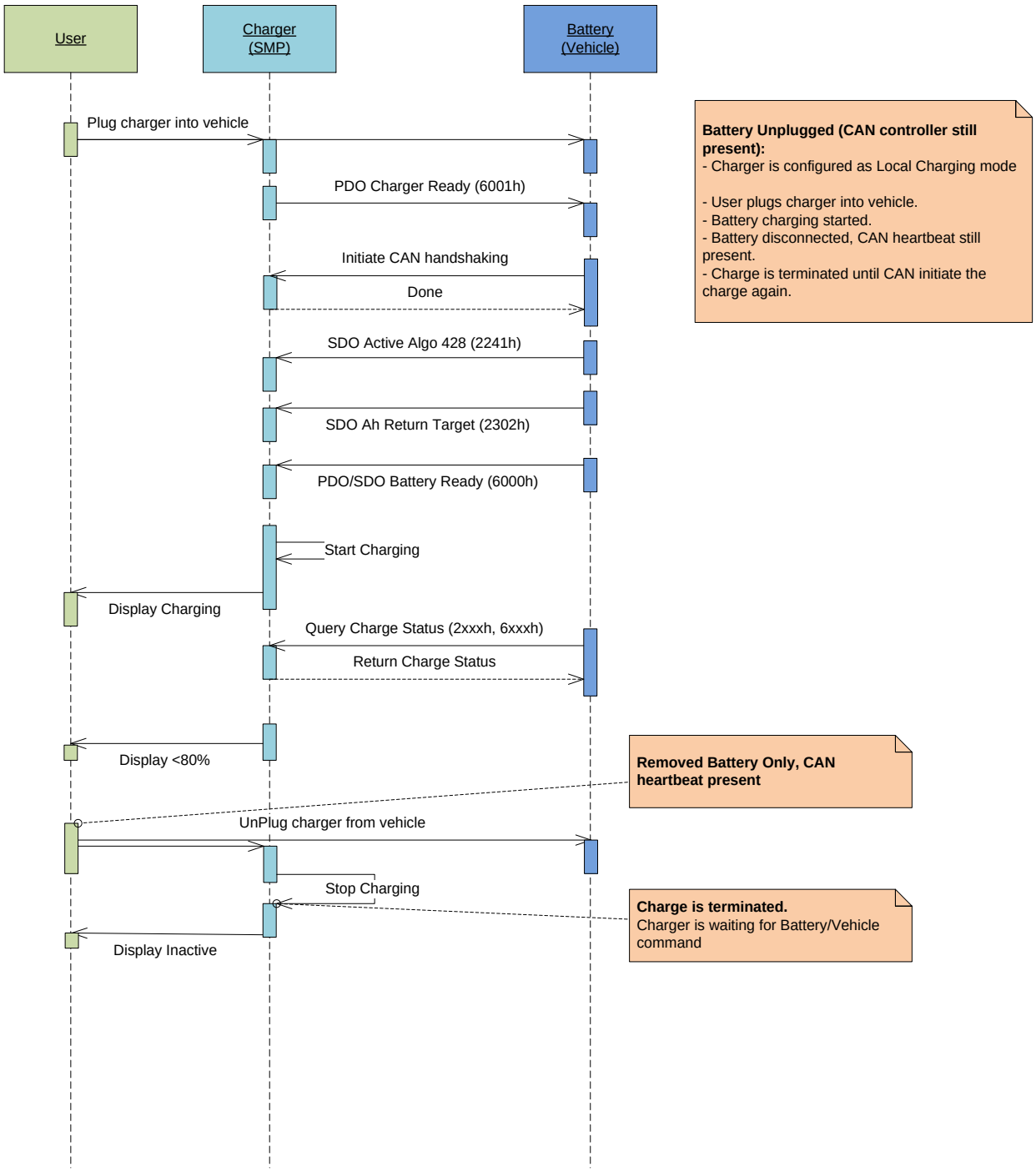


Figure 9 Sequence Diagram: CAN Connected with Battery Unplugged

8 Cycle vs. Cycle Related Parameters Update

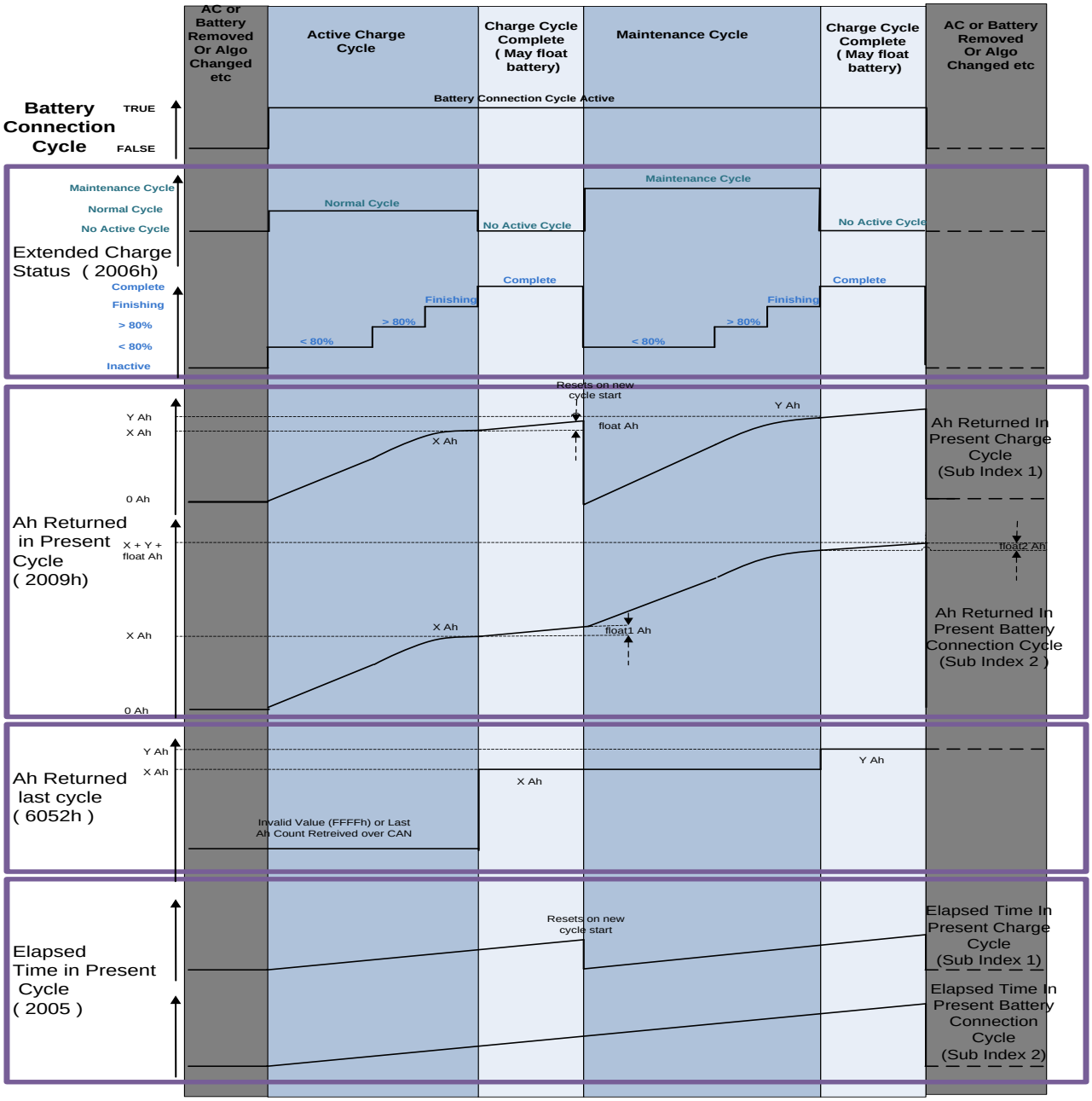
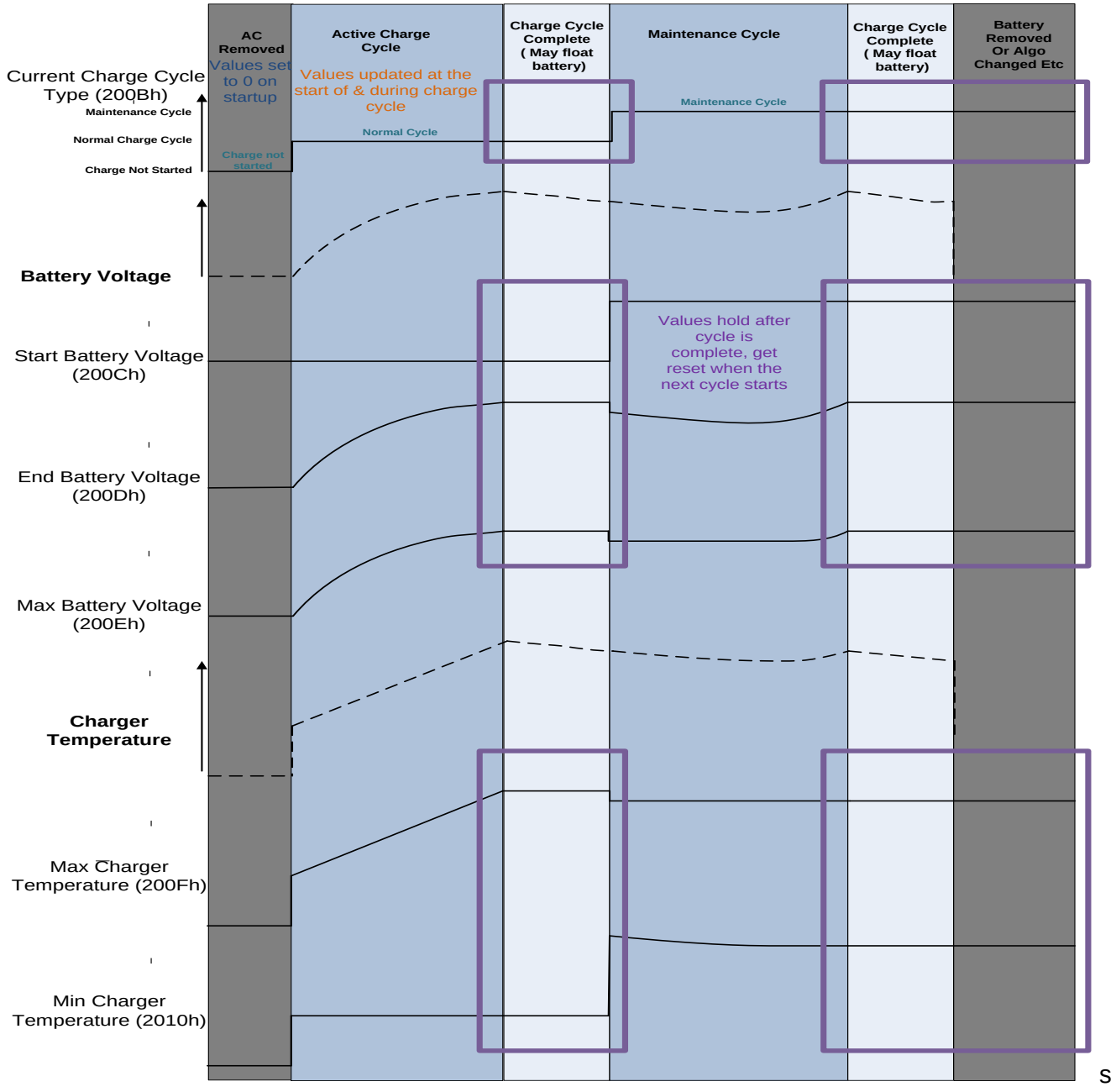


Figure 10 Cycle vs. Cycle Related Parameter Update

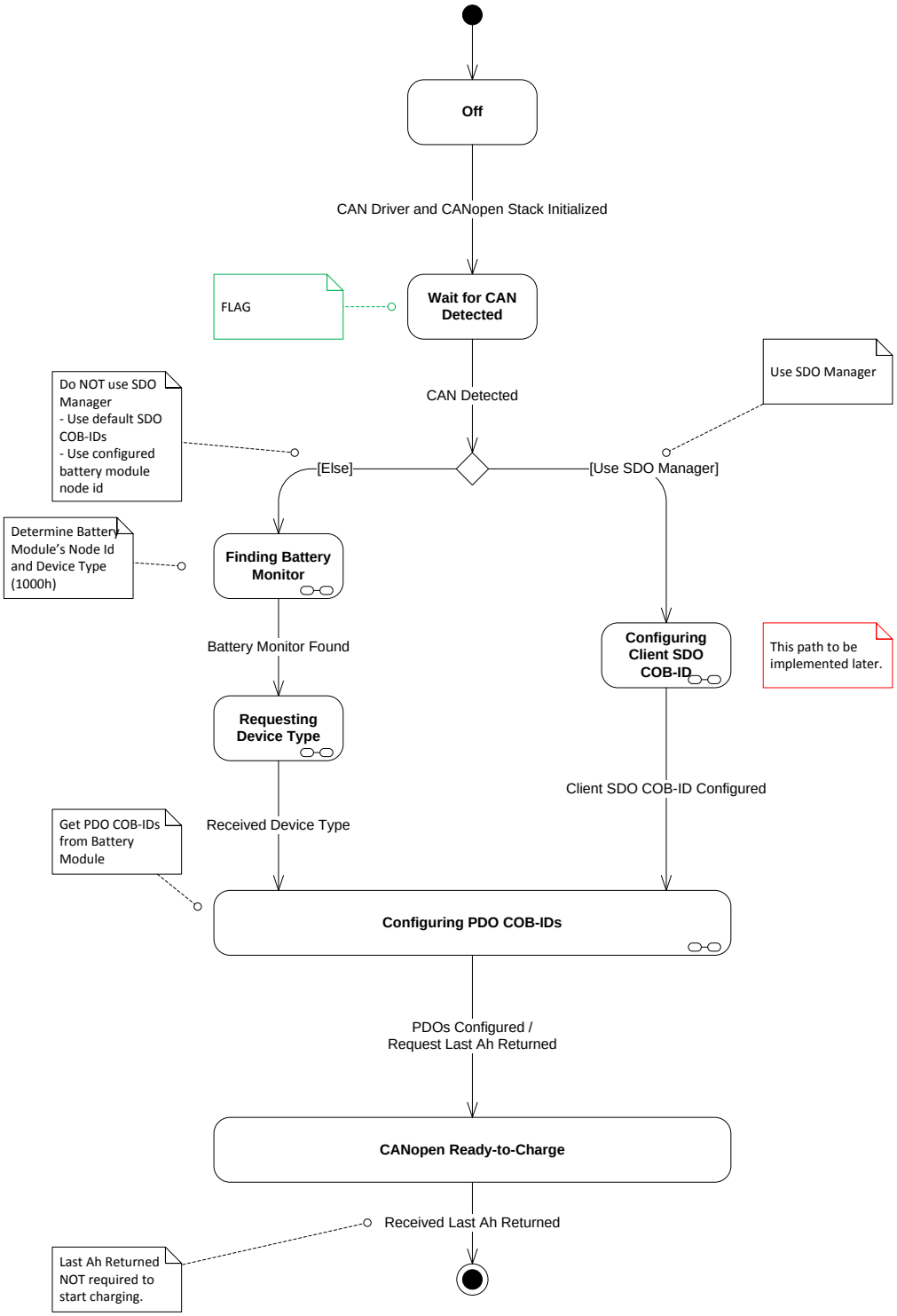
9 Current Charge Cycle Data Update



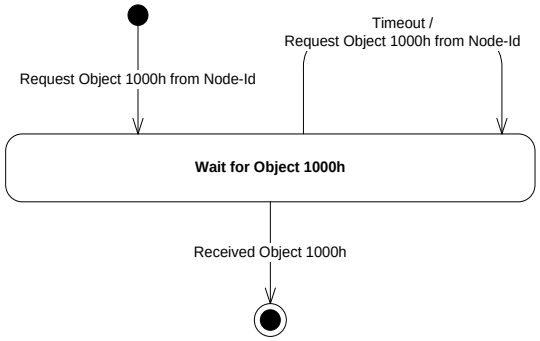
S

10 PDO Configuration State Machines

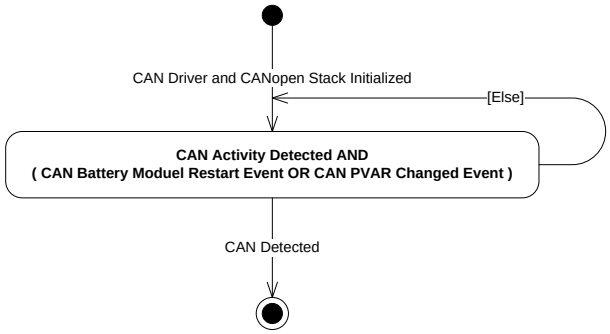
10.1 Top-level


Off
CAN Driver and CANopen Stack Initialized
Wait for CAN Detected
FLAG
CAN Detected
Do NOT use SDO Manager
- Use default SDO COB-IDs
- Use configured battery module node id
Use SDO Manager
[Else]
[Use SDO Manager]
Determine Battery Module's Node Id and Device Type (1000h)
Finding Battery Monitor
Battery Monitor Found
Configuring Client SDO COB-ID
This path to be implemented later.
Requesting Device Type
Client SDO COB-ID Configured
Get PDO COB-IDs from Battery Module
Received Device Type
Configuring PDO COB-IDs
PDOs Configured / Request Last Ah Returned
CANopen Ready-to-Charge
Received Last Ah Returned
Last Ah Returned NOT required to start charging.

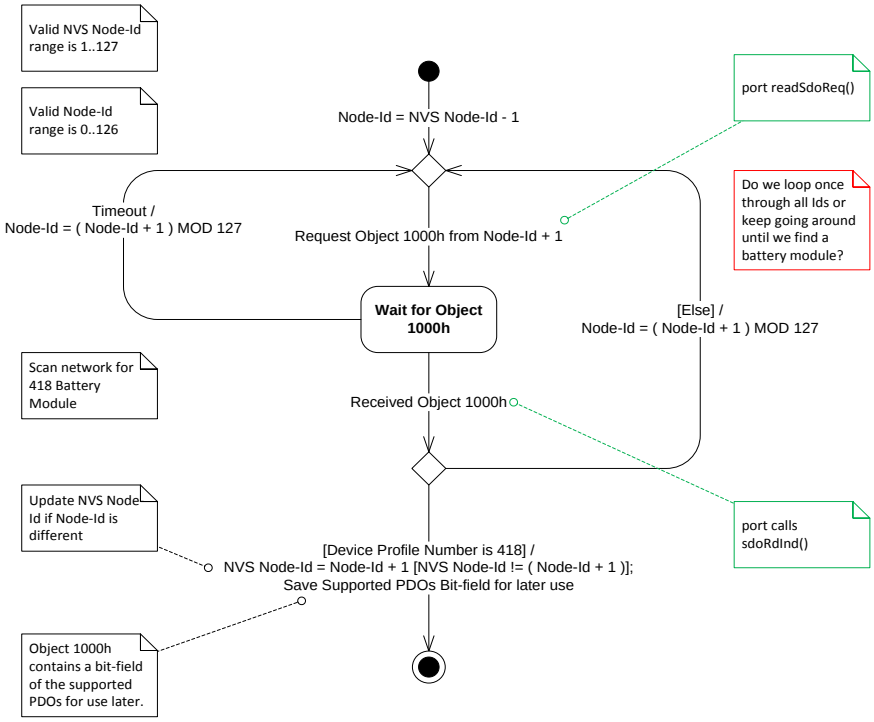
10.2 Requesting Device Type



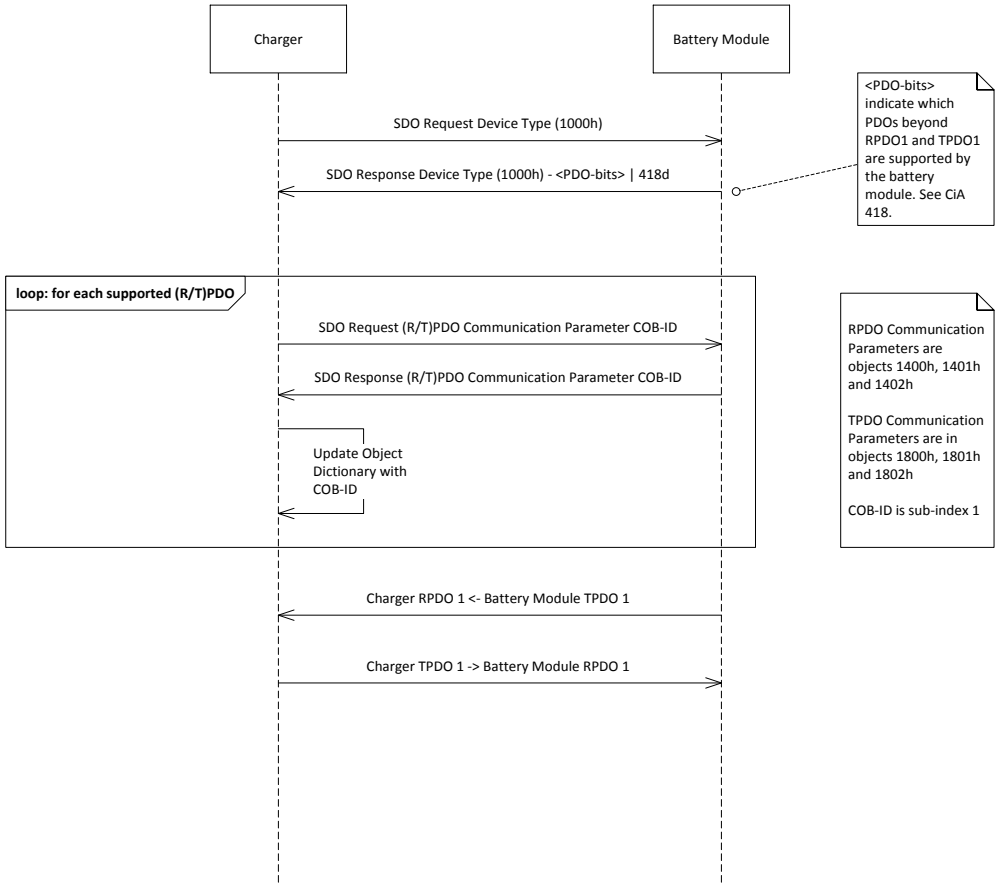
10.3 CAN Detected



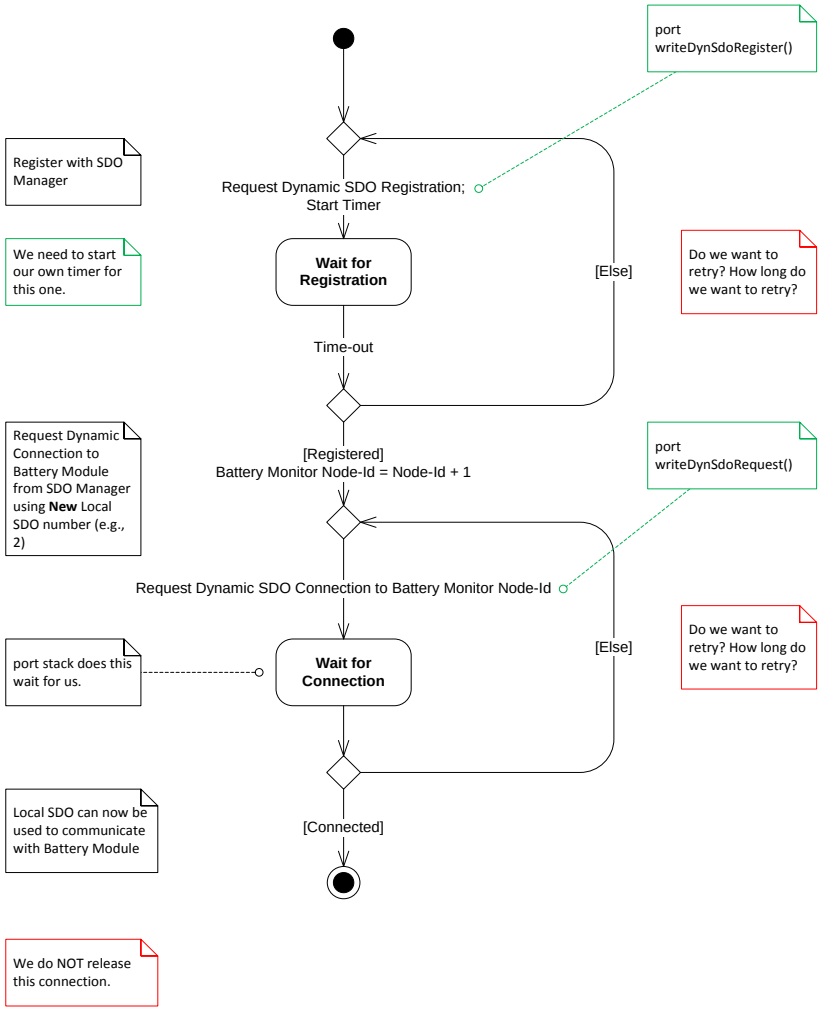
10.4 Finding Battery Module



10.5 Sequence



10.6 Configuring Client SDO COB-ID



10.7 Configuring PDO COB-IDs