power developers

EVC - Electrical Vehicle Charger

Interface description

Applicable for: **V2G500V30A** – 10kW EVC (Bidirectional, Vehicle 2 Grid)

Revision	Date	Comment	Author
1.0	20-09-2018	Start version V3.xx [base: 1.1a] Added temp error Switch Off Reason to decimal value	R.Baardman
1.0a	5-03-2019	Comment: module disable after OVP; added voltage input to (+9V … +30V); changed status_ext to correct values Added heatsink temp,	R.Baardman
1.0b	26-03-2019	Update: 0x2104, foldback	R.Baardman
1.0c 1.0d	11-04-2019 02-05-2019	Updated various	R.Baardman
1.0e	06-11-2019	Updated to version V3.28	R.Baardman
1.1	15-01-2020	Revision for hardware versions V2.12	M. Wolleswinkel
1.11	30-01-2020	Revision for hardware versions V2.20	M. Wolleswinkel
1.12	20-02-2020	Minor additions	M. Wolleswinkel
1.13	20-04-2020	Minor changes	M. Wolleswinkel
1.14	06-07-2020	Added optional RCD feature	M. Wolleswinkel
1.15	20-07-2020	Added DC undervoltage setpoint	M. Wolleswinkel

Change overview since latest version (1.14)

- Added ‘DC undervoltage setpoint’ (see 4.2.14)

Contents

1 Introduction

1.1 Purpose and scope

This document describes the hardware interfacing and CAN interfacing used to control the EVC power module.

1.2 Definitions, acronyms, and abbreviations

EVC	Electric Vehicle Charger
V2G	Vehicle to Grid
V2H	Vehicle to Home
Bidi	Bidirectional charger (V2G)
COB ID	CANbus Identifier
OVP	Over Voltage Protection
UVP	Under Voltage Protection
OTP	Over Temperature Protection
CC	Constant Current
CV	Constant Voltage
NSP	Network and Safety Processor
DCM	DC Microprocessor

2 Hardware overview

2.1 Aux power supply input/output

The Aux power input +12V / 24V (voltage range: +9V … +30V) @100mA max. is used to power the CANbus interface. Caution: only one power supply should be used per CANbus system, so 1 power supply will power all (stacked) modules within the same CANbus network. Note: if 5 units are stacked, use a power supply of at least 500mA.

2.2 Hardware connection

In Figure 1 the hardware connections of the BiDi 10kW module.

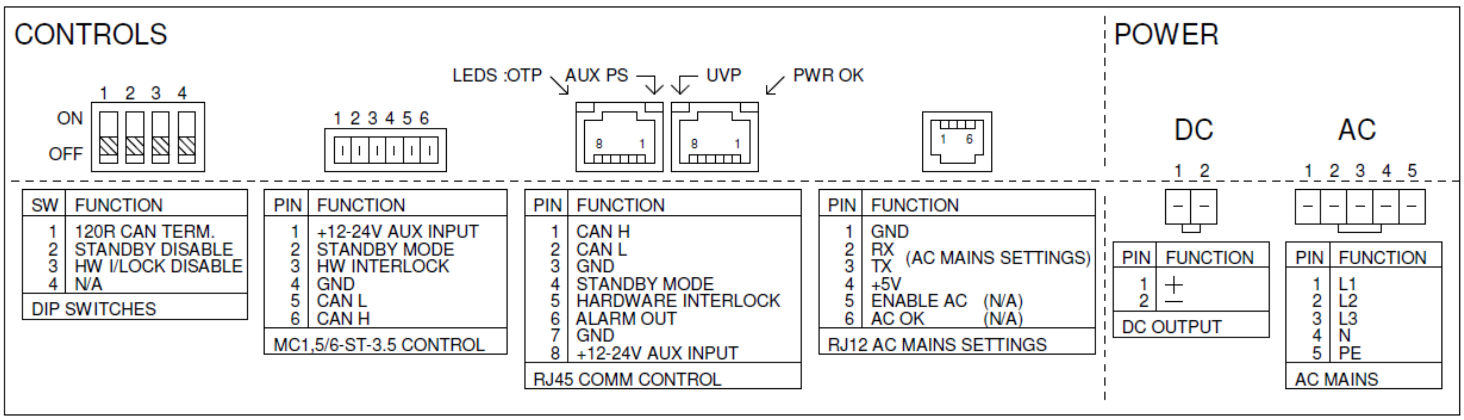

Figure 1: Bidi 10kW hardware connection overview

2.2.1 ENABLE pin (interlock)

The hardware ENABLE pin or ‘interlock’ is the hardware signal required to enable the module. A signal of at least 12V on this pin is required to enable the module. If the voltage on this pin is lower or floating, the module assumes there is a fault in the controller and immediately disables its output power. This signal can for instance be used in series with an emergency stop button.

2.2.2 Standby Mode

The standby mode pin is the hardware signal to disable the module to reduce standby losses when not used. The module is in standby mode when the pin is low, thus ground in applied. In standby mode the mains relay is switched OFF, reducing the power consumption to <2W and CANbus communication is not possible.
The module is active when the pin is high, thus 12V applied. When the module is active the mains relay is switched ON, power consumption is approx. 17W.

2.2.3 Alarm output

The Alarm output is an open collector output with an 100K pull-up resistor, 9…24V / OVP function.

2.2.4 Dip switch options

1. 120R CANbus terminator. ON = 120R
2. Disable ext. Standby mode (always on). ON = always ON (no standby)
3. Disable ext. HW Enable (Interlock) (always on). ON = always ON
4. **Reserved** (keep in OFF position)

2.3 CANbus connection and cable

The unit has 2 RJ45 connectors for CANbus connection with physical layer according ISO 11898-2. See pin-out below. When stacking multiple modules both RJ45 connectors can be used to link the units together, since the connectors are parallel wired.

The CANbus cable should be a twisted pair with an impedance of 120 ohm (UTP Ethernet cable will do).

2.4 CANbus termination

The module has an on-board CANbus bus termination resistor of 120Ω. It can be switched ON and OFF by setting the dipswitch 1. Switching it towards the 'ON' marking will enable the 120Ω bus termination. The location of the CANbus bus termination jumper is right behind the CANbus connectors on the PCB.

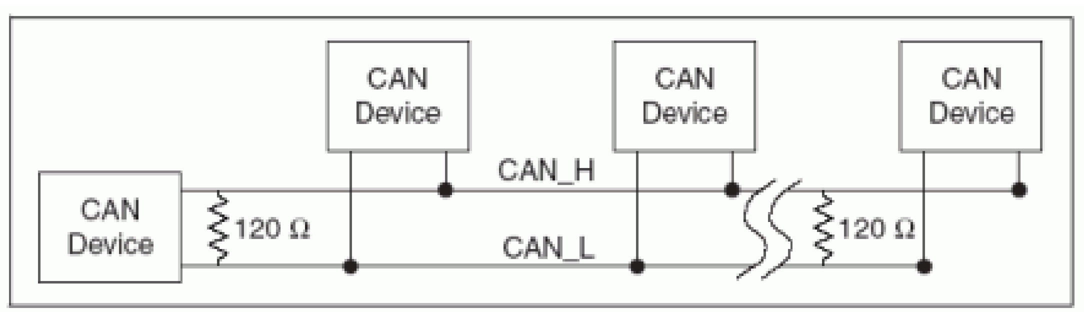

Figure 2: CANbus termination

Note: A CANbus network needs 2 termination resistors in its network; normally at each end of the bus. Make sure no more (or less) termination resistors are used!

2.5 LED status

The status LED's are located in the corners of the RJ45 (CANbus) connectors.

2.5.1 Left-left yellow: Over Temperature Protection (OTP).

This LED indicates that the module is in Over Temperature Protection.
CANbus termination

2.5.2 Left-right green: 12V Auxiliary supply present.

This LED indicates that the CANbus power supply is present on this board.

2.5.3 Right-left yellow: Mains Under Voltage Protection.

This LED indicated that the AC grid voltage is too low.

2.5.4 Right-right green: Charger Active

This LED indicates that the module is active. That is, the hardware and software is enabled, and no system faults.

2.6 CANbus hardware interface

Example of a typical CANbus interface.

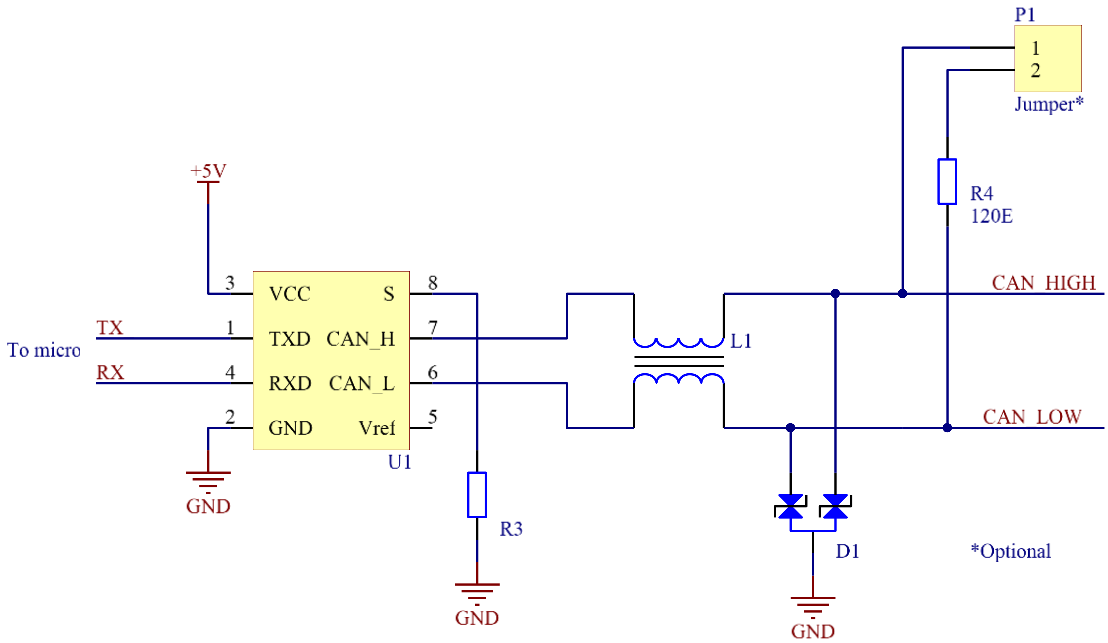

Figure 3: Typical CAN hardware interface

3 CANbus Communication protocol

3.1 CAN communication settings

This paragraph describes part of the physical layer of the communication.
Host CAN parameters:

Parameter	Setting
Default baud rate	500 kbps
Sample moment	75%

3.2 CANbus Addressing

The host must be able to select / address multiple Power Modules. For this purpose, the CAN-node-id's are used. This table defines which CANopen ID is (to be) assigned to which device.

For communicating with the power modules:
Host should address power modules via address-> 0x600+ID
Power module will answer via address -> 0x580+ID

Module (server)	Node ID (EVC)	CANID
<Broadcast>	0x00	0x600
Power Module 1	0x01	0x601
Power Module 2	0x02	0x602
Power Module 3	0x03	0x603
Power Module 4…79	0x04… 0x4F	0x604… 0x64F
Power Module 48	**0x30**	**0x630 [default]**

The default Node ID, out of the factory, is 0x30, so the CANID 0x630 (reply: 0x5B0).

Important notice:
According to the CANopen specification ID 0x7E6 to 0x7FF are unused in the pre-defined connection set. PRE reserves ID 0x7FE and 0x7FF and should therefore not be used.

3.3 CANbus Connectivity

The module requires CANbus messages on a regular basis. The module will switch off after 1000 milliseconds if it has not received a message (addressed to that module). Send CANbus message to each module frequently, but at minimum once every 1000ms. It is advised to send a message to each module at least every 500ms.

4 CANopen

This chapter describes the implemented CANopen-like stack for communication with the host. No network functionality is implemented; just the CANopen message layout and corresponding send and receive behaviour. This way no CANopen stack is needed for the master/server.

4.1 CANopen data frames

The CANopen SDO messages are constructed as follows:

ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
0x6**UU**	8	CD	OD Index		SI	Data			

ID = 0x6**UU** where **UU** is the node ID
DLC = Data Length, is always 8
CD = Command Byte
OD Index = Object Dictionary Index
SI = SubIndex
Data = data

4.1.1 Command bytes (CD)

Where Command Byte for a 32-bit read or write operation is:

SDO Write Master	Value	Data
Master reads from Slave	0x40	
Master writes to Slave	0x23	4 bytes
Master writes to Slave	0x27	3 bytes
Master writes to Slave	0x2B	2 bytes
Master writes to Slave	0x2F	1 byte

Where Command Byte for a 32-bit Slave response is:

SDO Answer Slave	Value	Data
Successful read response	0x43	4 bytes
Successful read response	0x47	3 bytes
Successful read response	0x4B	2 bytes
Successful read response	0x4F	1 byte
Successful write response	0x60	
Abort operation	0x80	

(Slave will respond with the identical message structure as Master)

Note: OD and data are transmitted as ‘Little Endian’, thus low value first!

4.1.1.1 Abort operation

The CANopen stack reports known failure codes when objects are read/written, when this is not allowed or possible. The module sends failure codes back to the sender, who is responsible for the correct communication. Sender can then adjust the commands to fit the specification in this document. The Abort message has a Command byte value of 0x80.

Implemented failure codes are:

Failure code	Value	Explanation
Not existing object	0x06020000	Object does not exist in the object dictionary
Write only object	0x06010001	Attempt to read a write-only object
Read only object	0x06010002	Attempt to write a read-only object
Error local control	0x08000021	Data cannot be transferred or stored to the application because of local control
Invalid command specifier	0x05040001	Client/Server command specifier not valid or unknown (wrongly constructed CAN message)

The first three failure codes are self-explanatory and can be easily diagnosed (e.g. don’t write to a read-only object).
The “Error local control” is an internal state of the module and could be caused by a wrong size in the “Command byte (CD)”, e.g. attempt to write to a 16-bit object with a 32-bit value.

Example: Write 0x2101 (Read-only object)

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	2B 01 21 00 00 00 00 00	Write attempt (data = 0)
RX	5B0	8	80 01 21 00 02 00 01 06	Attempt to write a read only object

Tx:
ID = 0x630
DLC = 0x8
CD = 0x2B
OD Index = 0x2101
SI = 0x0
Data = 0x00000000

Rx:
ID = 0x5B0
DLC = 0x8
CD = 0x80 – Abort!
OD Index = 0x2101
SI = 0x0
Data = 0x06010002

4.2 CANopen objects

The CANopen objects are divided in 2 object sections; the standard objects and the extended objects. The **standard objects are mandatory** and thus needed to monitor and control the power module. The extended objects are not needed to control the module, but can be used for extra monitoring and debugging. Due to the complexity of the power module, some internal (which are not further explained) objects are available, which are generally used for debugging purposes. They can be found in Appendix chapter 9.1.1.

4.2.1 Standard object overview

Object index	SI	Description	r/w	unit	size / type	
Communication Profile Area						
1008	4	Device name	r	string	4	
1009	4	Hardware version	r	string	4	
100A	4	Software version	r	string	4	
1018	4	Identity	r	-	uint32	
Manufacturer Specific Profile Area - Control / Monitor						
2100		Module enable	r/w	-	uint16	
2101		Module status	r	-	uint16	
2102		Module status extended	r	-	uint32	
2104		Module temperature (highest)	r	0.1C	int16	
2105		UAC / input	r	0.1V	uint16	
2106		IAC / input	r	0.1A	uint16	
2107		UDC / output	r	0.1V	uint16	
2108		IDC / output	r	0.1A	int16	
2109		UDC / output setpoint	r/w	0.1V	uint16	
210A		IDC / output setpoint	r/w	0.1A	int16	
210D		UBUS	r	0.1V	uint16	
210E		ISOLAR input on bus (option)	r	0.1A	uint16	
210F		UDC undervoltage setpoint	r/w	0.1V	uint16	
211F		UAUX internal	r	0.1V	uint16	
2121		UAC / input L1	r	0.1V	uint16	
2122		UAC / input L2	r	0.1V	uint16	
2123		UAC / input L3	r	0.1V	uint16	
2124		IAC / input L1	r	0.1V	uint16	
2125		IAC / input L2	r	0.1V	uint16	
2126		IAC / input L3	r	0.1V	uint16	
2130		Warning status	r	bit	uint16	
2132		Error source	r	bit	uint16	
2136		Temperature ambient	r	0.1C	int16	
2137		Temperature secondary	r	0.1C	int16	
2138		Temperature primary	r	0.1C	int16	

2139		Temperature transformer	r	0.1C	int16	
2140		Temperature AC heatsink	r	0.1C	int16	
2149		UDC setpoint (real)	r	0.1V	uint16	
214A		IDC setpoint (real)	r	0.1A	int16	
214B		Available power (charge)	r	W	int32	
214C		Available power (V2G)	r	W	int32	
214D		Max DC current (charge)	r/w	0.1A	int16	
214E		Max DC current (V2G)	r/w	0.1A	int16	
2150	0-5	Switch-off reason	r	array[5]	uint32	
2151		Module uptime	r	sec	uint32	
2152	0-5	Switch-off timestamp	r	array[5]	uint32	
2241		Time to reconnection	r	sec	uint32	
2242		Reconnection reset reason	r	custom	uint32	
2552	1-5	AC switch-off reason	r	array[5]	uint32	
2553	1-5	AC switch-off timestamp	r	array[5]	uint32	
2FFF		Restart module	w	-	uint32	
Manufacturer Specific Profile Area - Configuration						
2FF0		Set node id - Subindex: new node id - Param: serial number	w	-	uint32	

4.2.2 Object 0x2100: Power Module Enable

Bit	Description	Value
0	Power module enable	0: Disable; 1: Enable
1…15	*Reserved, should be 0*	

4.2.3 Object 0x2101: Power Module Status

Bit	Description	Value	Note
0	Charger on	0: OFF; 1: ON	
1	Power error	0: OK; 1: ERROR	See 1
2	Input over voltage detect	0: OK; 1: ERROR	
3	Input under voltage detect	0: OK; 1: ERROR	
4	Output over voltage detect	0: OK; 1: ERROR	
5	Output under voltage detect	0: OK; 1: ERROR	
6	*Reserved, should be 0*		
7	Over temperature detect	0: OK; 1: ERROR	
8	Uaux error, UV / OV	0: OK; 1: ERROR	See 2
9	*Reserved, should be 0*		
10	Mode	0: CHARGE; 1: V2G	

11	Grid error (AC voltage, frequency, phase)	0: OK; 1: ERROR	
12	HW interlock error	0: OK; 1: ERROR	
13	Service mode enabled	0: OFF; 1: ON	See 3
14…15	*Reserved, should be 0*		

(1) When the module encounters a power error (overtemperature, overvoltage, etc.), the module needs to be disabled (object 0x2100) before it can be enabled again.
(2) When an Uaux error occurs, all DC measurements are not valid (and will read '0')
(3) When the module is in service mode, the power module cannot be enabled

4.2.4 Object 0x2102: Power Module Status Extended

Bit	Description	Value	Note
0	Error UAC L1	0: OK; 1: ERROR	NSP
1	Error UAC L2	0: OK; 1: ERROR	NSP
2	Error UAC L3	0: OK; 1: ERROR	NSP
3	Error IAC L1	0: OK; 1: ERROR	NSP
4	Error IAC L2	0: OK; 1: ERROR	NSP
5	Error IAC L3	0: OK; 1: ERROR	NSP
6	Error FAC L1	0: OK; 1: ERROR	NSP
7	Error FAC L2	0: OK; 1: ERROR	NSP
8	Error FAC L3	0: OK; 1: ERROR	NSP
9	Error, no communication with NSP	0: OK; 1: ERROR	NSP
10…20	*reserved, should be 0*		
21	Error phase	0: OK; 1: ERROR	AC
22	Error Vh	0: OK; 1: ERROR	AC
23	Error Vc	0: OK; 1: ERROR	AC
24	Error AC status	0: OK; 1: ERROR	AC
25	Test mode	0: OK; 1: TESTMODE	
26	Error communication AC	0: OK; 1: ERROR	AC
27	Error communication MPPT1	0: OK; 1: ERROR	MPPT1
28	Error communication MPPT2	0: OK; 1: ERROR	MPPT2
29…31	*reserved, should be 0*		

4.2.5 Object 0x2104: Power Module Temperature

This read-only object contains the highest temperature of all temperature measurements in the module, in steps of 0.1°C.

4.2.6 Object 0x2105: Power Module AC Input Voltage

This read-only object contains the measured average input voltage of the three AC phases (line-neutral), in steps of 0.1V

4.2.7 Object 0x2106: Power Module AC Input Current

This read-only object contains the measured average input current of the three AC phases, in steps of 0.1A

4.2.8 Object 0x2107: Power Module DC Output Voltage

This read-only object contains the measured output voltage, in steps of 0.1V.

4.2.9 Object 0x2108: Power Module DC Output Current

This read-only object contains the measured output current, in steps of 0.1A. Note that this value has a sign, which shows whether the DC current is delivered **by** the module (>0) or **to** the module (<0)

4.2.10 Object 0x2109: Power Module DC Output Voltage Setpoint

This read/write object contains the output voltage setpoint in steps of 0.1V.

4.2.11 Object 0x210A: Power Module DC Output Current Setpoint

This read/write object contains the output current setpoint, in steps of 0.1A.

4.2.12 Object 0x210D: Power Module DC Bus Voltage

This read-only object contains the internal DC bus voltage, in steps of 0.1V.

4.2.13 Object 0x210E: Solar Input Current (option)

This read-only object contains the solar current delivered into the DC bus, in steps of 0.1A.

4.2.14 Object 0x210F: Power Module DC Undervoltage Setpoint

This read/write object contains the output undervoltage setpoint in steps of 0.1V.

4.2.15 Object 0x2121: Power Module AC Input Voltage L1

This read-only object contains the measured L1 AC input voltage (line-neutral).

4.2.16 Object 0x2122: Power Module AC Input Voltage L2

This read-only object contains the measured L2 AC input voltage (line-neutral).

4.2.17 Object 0x2123 Power Module AC Input Voltage L3

This read-only object contains the measured L3 AC input voltage (line-neutral).

4.2.18 Object 0x2124: Power Module AC Input Current L1

This read-only object contains the measured L1 AC input current.

4.2.19 Object 0x2125: Power Module AC Input Current L2

This read-only object contains the measured L2 AC input current.

4.2.20 Object 0x2126: Power Module AC Input Current L3

This read-only object contains the measured L3 AC input current.

4.2.21 Object 0x2130: Warning status

Bit	Description	Value
0	*Reserved*	0: OFF; 1: ON
1	Derating by **temperature** active. *The current setpoint is decreasing due to high temperature.*	0: OFF; 1: ON
2	Derating by **maximum power** active. *The current setpoint is decreasing due to high power. Max power is exceeded if **Udc * Idc > max. rated power** (e.g. 6kW or 10kW)*	0: OFF; 1: ON
3	*Reserved*	
4	Derating by **over-frequency** *The current setpoint is decreasing due to high AC frequency*. *This parameter is determined by the active grid code.*	0: OFF; 1: ON
5	Derating by **under-frequency** *The current setpoint is decreasing due to low AC frequency*. *This parameter is determined by the active grid code.*	0: OFF; 1: ON
6	Ramp rate limits current setpoint due to **reconnection** *The current setpoint is not equal to the desired setpoint due to a previous grid disconnect event. After this, the active grid code determines a maximum rate at which the output power may be increased.*	0: OFF; 1: ON
7	Ramp rate limits current setpoint due to **active power rate** *The current setpoint is not equal to the desired setpoint due to the defined active power rate. This parameter is determined by the active grid code*	0: OFF; 1: ON
8	No power available due to **converter halt** *The converter is not allowed / able to deliver power due to an internal halt event (e.g. grid voltage dip)*	0: OFF; 1: ON
9	Derating by **AC current limit** *The current setpoint is decreasing due to the maximum allowed current on the AC (grid) side. This can be caused by low grid voltage.*	0: OFF; 1: ON
10..15	*Reserved*	

4.2.22 Object 0x2132: Error source

For extensive root cause analysis of error conditions, this object can contain extra information regarding the source of certain errors.

Bit	Description	Value
0	*Reserved*	
1	Overvoltage was detected on **internal BUS**	0: OFF; 1: ON
2	Overvoltage was detected on **DC output**	0: OFF; 1: ON
3…15	*Reserved*	

4.2.23 Object 0x2136: Temperature ambient

This is the ‘ambient’ temperature, where the sensor is fitted on the PCB (not near local heat sources). Measured in steps of 0.1 °C

4.2.24 Object 0x2137: Temperature secondary

This is the temperature measured on the heatsink of the internal DC bus side of the converter. The temperature in steps of 0.1 °C.

4.2.25 Object 0x2138: Temperature primary

This is the temperature measured on the heatsink of the DC output side of the converter. The temperature in steps of 0.1 °C.

4.2.26 Object 0x2139: Temperature transformer

This is the temperature measured on the transformer of the main converter. The temperature in steps of 0.1 °C.

4.2.27 Object 0x2139: Temperature heatsink AC

This is the temperature measured on the heatsink of the AC converter. The temperature in steps of 0.1 °C.

4.2.28 Object 0x2149: UDC setpoint (real)

This read-only object contains the real (current) setpoint of **UDC** that is used by the module. Due to hardware / software restrictions the value of the desired setpoint set in object 0x2109 can be modified by the module. E.g. when UDC is set to 800V (not within allowed range!) the current UDC setpoint is reduced to the maximum allowable voltage, which is ultimately reflected with this object (0x2149).

4.2.29 Object 0x214A: IDC setpoint (real)

This read-only object contains the real (current) setpoint of **IDC** that is used by the module. Due to hardware / software restrictions the value of the desired setpoint set in object 0x210A can be modified by the module. E.g. when module is derating due to temperature, the actual setpoint of IDC will be lower than the desired setpoint, which is ultimately reflected with this object (0x214A).

4.2.30 Object 0x214B: Available power (charge mode)

This read-only object contains the available power if the module is in charge mode. The power can for instance be limited by temperature and grid conditions. Note that depending on the active grid-code, the available power can even become inverted (meaning that although in charge mode, the module is required to deliver power to the grid)

4.2.31 Object 0x214C: Available power (V2G mode)

This read-only object contains the available power if the module is in V2G mode. The power can for instance be limited by temperature and grid conditions. Note that depending on the active grid-code, the available power can even become inverted (meaning that although in V2G mode, the module is required to draw power from the grid)

4.2.32 Object 0x214D: Maximum DC current (charge)

This read/write object contains the maximum current allowed in charge mode on the DC side of the module. This should be used to prevent the module from requesting higher currents than the connected battery/load can handle due to state-of-charge or other constrictions. By default, this value contains the maximum current setpoint in charge mode (=28 A). In steps of 0.1A.

4.2.33 Object 0x214E: Maximum DC current (V2G)

This read/write object contains the maximum current allowed in V2G mode on the DC side of the module. This should be used to prevent the module from requesting higher currents than the connected battery/load can handle due to state-of-charge or other constrictions. By default, this value contains the maximum current setpoint in V2G mode (=-28 A, ***mind the sign!***). In steps of 0.1A.

4.2.34 Object 0x2150: Switch OFF reason

This read-only object contains (since the last power cycle) the last 5 reasons why the module has disabled. This can be useful for root cause analysis in case of shutdown.

SI	Description	Value
0	Last known entry	0: OFF; 1: ON
1…4	Known entry last-SI	0: OFF; 1: ON

#	Description	Value
0	User switch off (disabled via CANbus)	0x00000001
1	Hardware Interlock disabled	0x00000002
2	Reboot request	0x00000004
3	Grid error (detected by NSP, grid code dependent)	0x00000008
4	CANbus time out (see 3.3)	0x00000010
5	Overvoltage on AC	0x00000020
6	Undervoltage on AC	0x00000040
7	Overvoltage on DC	0x00000080
8	*Reserved*	0x00000100
9	Overtemperature	0x00000200
10	Overvoltage on internal BUS	0x00000400
11	*Reserved*	0x00000800
12	Over- or undervoltage on internal auxiliary supply	0x00001000
13	NSP (communication) error	0x00002000
14	AC overcurrent protection error	0x00004000
15…31	*Reserved*	

4.2.35 Object 0x2151: Module uptime

This read-only object contains the time in seconds since the module was powered on. This value will only be reset to 0 when the power modules internal controller is not powered anymore (e.g. external AC switch off or AC blackout)

4.2.36 Object 0x2152: Switch OFF reason timestamp

This read-only object contains (since the last power cycle) the relative timestamp of each of the last 5 reasons why the module has disabled.

SI	Description	Value
0…4	Timestamp (SI 0 is the most recent)	Seconds since start-up when switch-off occurred

4.2.37 Object 0x2241: Time to reconnection

This read-only object contains the time in seconds until the module will (re)connect to the grid. If this value is 0, the module is already connected. If the value does not count down but remains at its grid-code defined (re)connection time (e.g. 60 seconds for VDE), there is some grid or system error which makes the unit unable to start the (re)connection countdown.

4.2.38 Object 0x2242: Reconnect reset reason

This read-only object contains the reason that caused the (re)connection timer (see 4.2.37) to reset to its grid-code defined (re)connection time.
The mapping of this value is equivalent to that of the switch off reason and can be found in Table 1: Switch Off Reasons

4.2.39 Object 0x2552: AC switch-off reason

See section 6.2

4.2.40 Object 0x2553: AC switch-off timestamp

See section 6.2

4.2.41 Object 0x2FFF: Restart

This write-only object will restart the module.

4.3 SDO object details – Configuration

4.3.1 Object 0x2FF0: Configuration Node ID

With this object the node ID of this unit can be changed. After writing to this node, the node ID is changed instantly.
The factory default for the node ID is 0x30, so the unit is listening to 0x630 (and will reply with 0x5B0). When using more than one module in the same CANbus network, each module needs to have a unique node ID to prevent loss of data and possible failure of the CANbus. The broadcast node ID is 0x00, thus CAN ID 0x600.

When setting up a CANbus system with multiple power modules, make sure only one module is connected to the CANbus.

1. Issue a broadcast 'Identity' object 0x1018 with Node ID 0x00, thus 0x600 as broadcast CAN ID.
2. The connected unit will reply with CAN ID 0x5B0 and its identity, the serial number.
3. Construct a new CAN message with CAN ID 0x630, object 0x2FF0, Sub Index (SI) will be the new node ID (e.g. 0x31) and the serial number received in step 2.

In the next example it is shown how it looks on the CANbus when changing the ID from 30 to 31 (resp. 0x630 to 0x631). Keep in mind that the serial number will be different for each module.

Example: Read 0x1018

Direction	ID (hex)	Length	Data (hex)	Comment
TX	600	8	40 18 10 04 00 00 00 00	Read node ID
RX	580	8	43 10 18 04 02 00 0B 0C	Reply: serial number

Example: Write 0x2FF0

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	23 F0 2F 31 02 00 0B 0C	Write new node ID 31
RX	5B1	8	60 F0 2F 31 00 00 00 00	Reply: serial number

5 BiDi power module usage

This chapter describes the usage of the BiDi module, starting the converter, setting current, setting output voltage. Note that the following instructions is a ‘good practice’ guide of controlling of the BiDi module and focused to prevent / limit overshoot and undesirable (output) behaviour.
Every system is different and may require a different setup.

DISCLAIMER
Please make sure below instructions and values suit your system and do not exceed maximum ratings!

5.1 Example EV charger setup

The following figure gives an example of how the modules should generally be used in an EV charger setup.

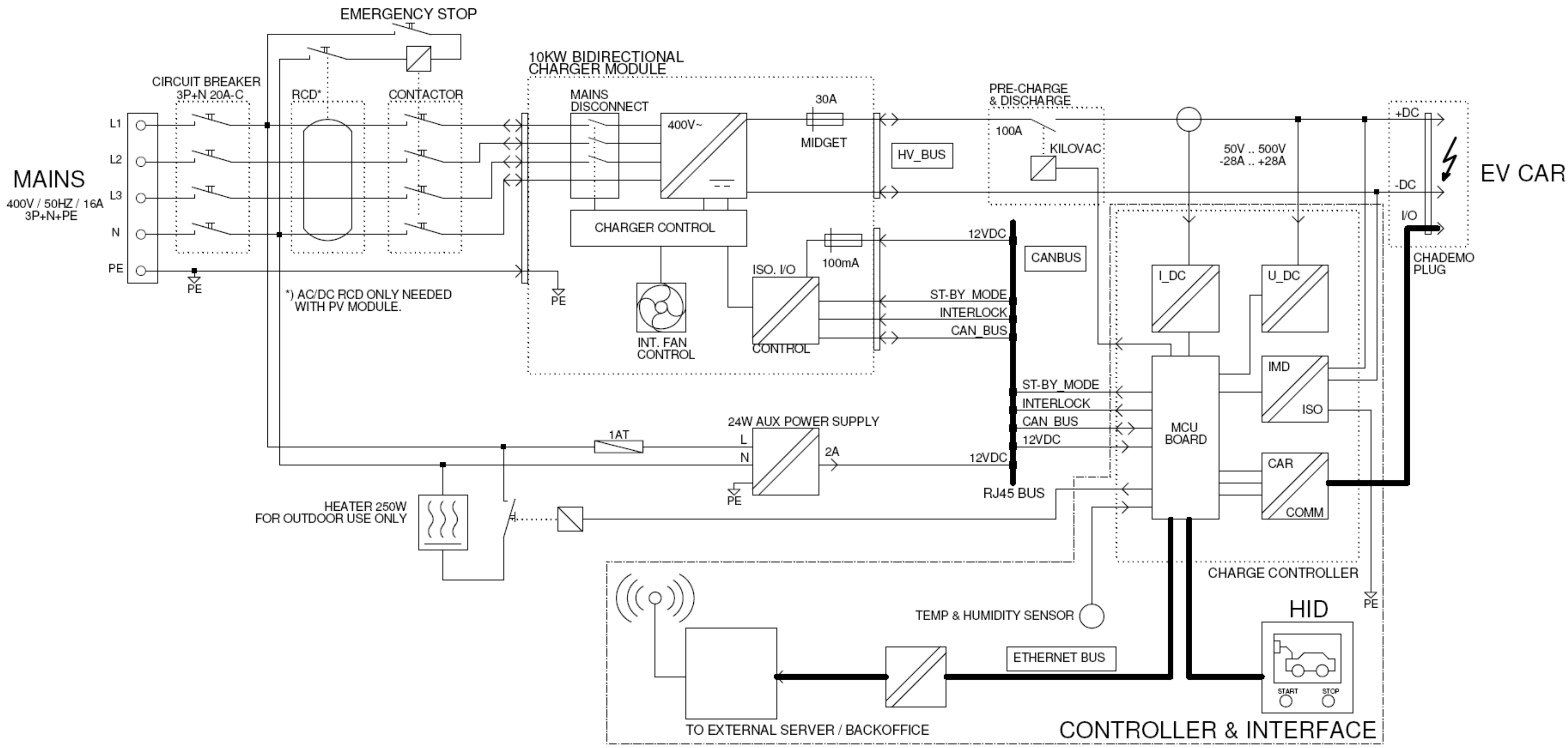

Figure 4: Example EV charger setup

5.2 Controlling the power module

Generally, the module can be controlled in the following way.

1. Isolation test (if required) according to CHAdeMO or CCS
2. Pre charge
3. Start Charge / V2G
4. Stop Charge / V2G
5. Discharge (optional)

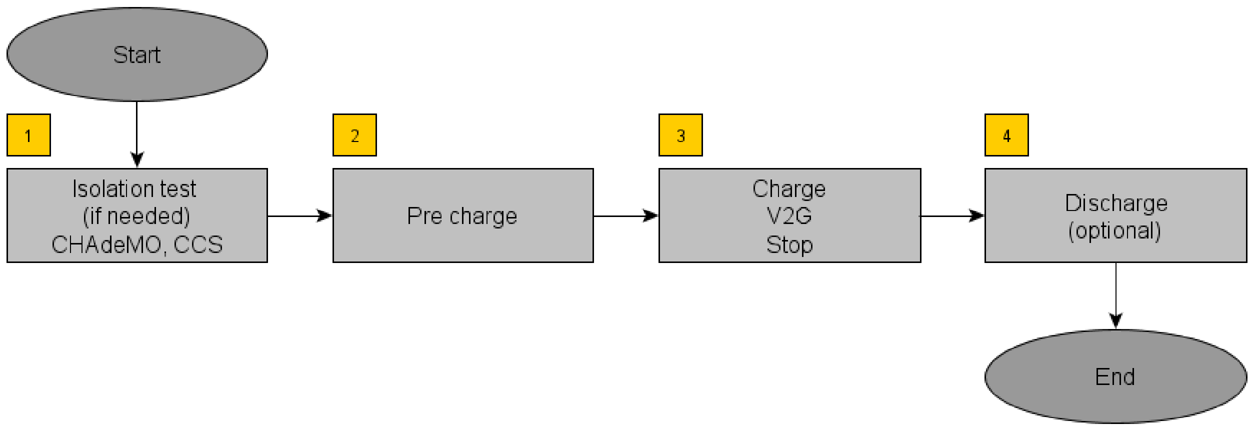

Figure 5: Controlling overview

5.2.1 Isolation test (if required)

1. Send **'Enable'** command and set the output current setpoint to **1A**.
2. Wait 2 sec
3. Set the output voltage setpoint to **500V** (after 600ms the isolation test is performed, e.g. CHAdeMO, CCS)
4. Wait 800 ms (= 600ms + 200ms isolation test)

Note: the isolation test is performed by the charge controller, the module does not do any test itself!

5. Set the output current setpoint to **-1A** (this discharges the output quickly)
6. Send '**Disable'** if the isolation test did not pass, else keep the module enabled continue to pre charging stage

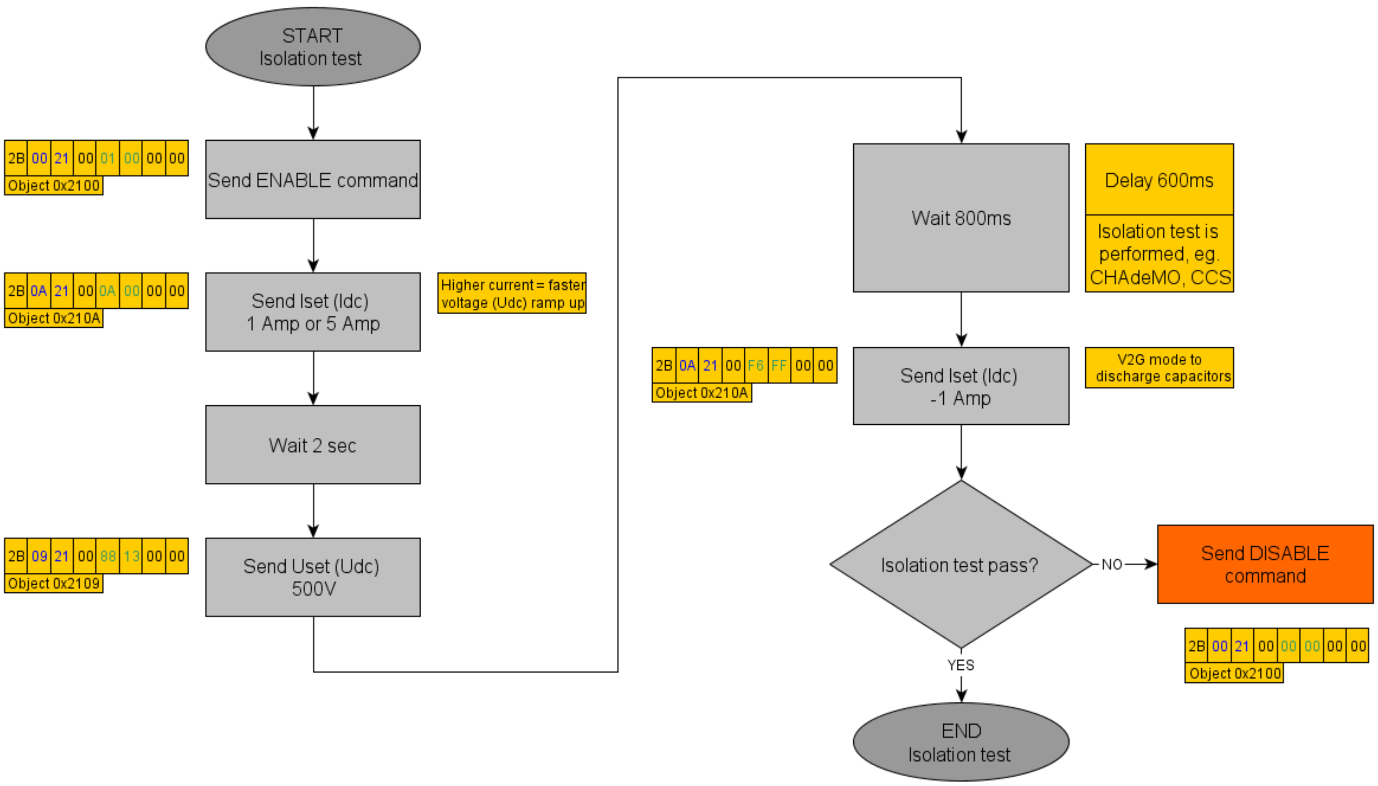

Figure 6: Isolation test

5.2.2 Pre-charge

Pre-charging is required to prevent high inrush currents when connecting the battery to the power module.

WARNING
Incorrect or lack of pre-charging can damage the power module due to very high inrush current from the battery to the module. Make sure you implement the pre-charge procedure correctly in your application to avoid damage to the power module.

1. Cable pre-charge; DC contactors are open (the battery is not connected)
   (Send 'Enable' command if no isolation test is required)
2. Set the output current setpoint to **1A**
3. Set the output voltage setpoint to (**battery voltage – 5V)**
   The voltage by the module must be slightly lower to prevent the module from charging instantaneously when the contactors close.
4. When the output voltage is stabilised at the setpoint (battery voltage – 5V), close the DC contactors.

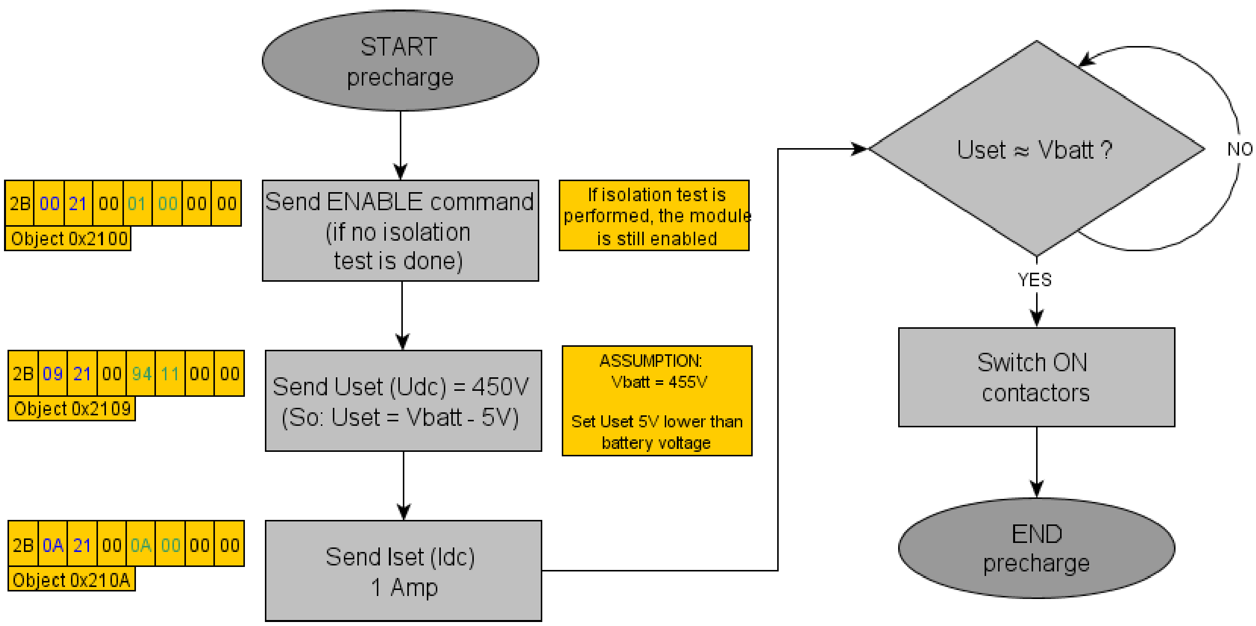

Figure 7: Precharge

5.2.3 Start Charge / V2G

1. Ensure that the pre-charge sequence is complete
2. Set the output current setpoint to **0A**
3. Set the output voltage setpoint to the **maximum battery voltage allowed**
4. Ramp up the output current setpoint to the desired setpoint
   - For charging mode -> send a **positive** current setpoint
   - For discharging/V2G mode -> send a **negative** current setpoint

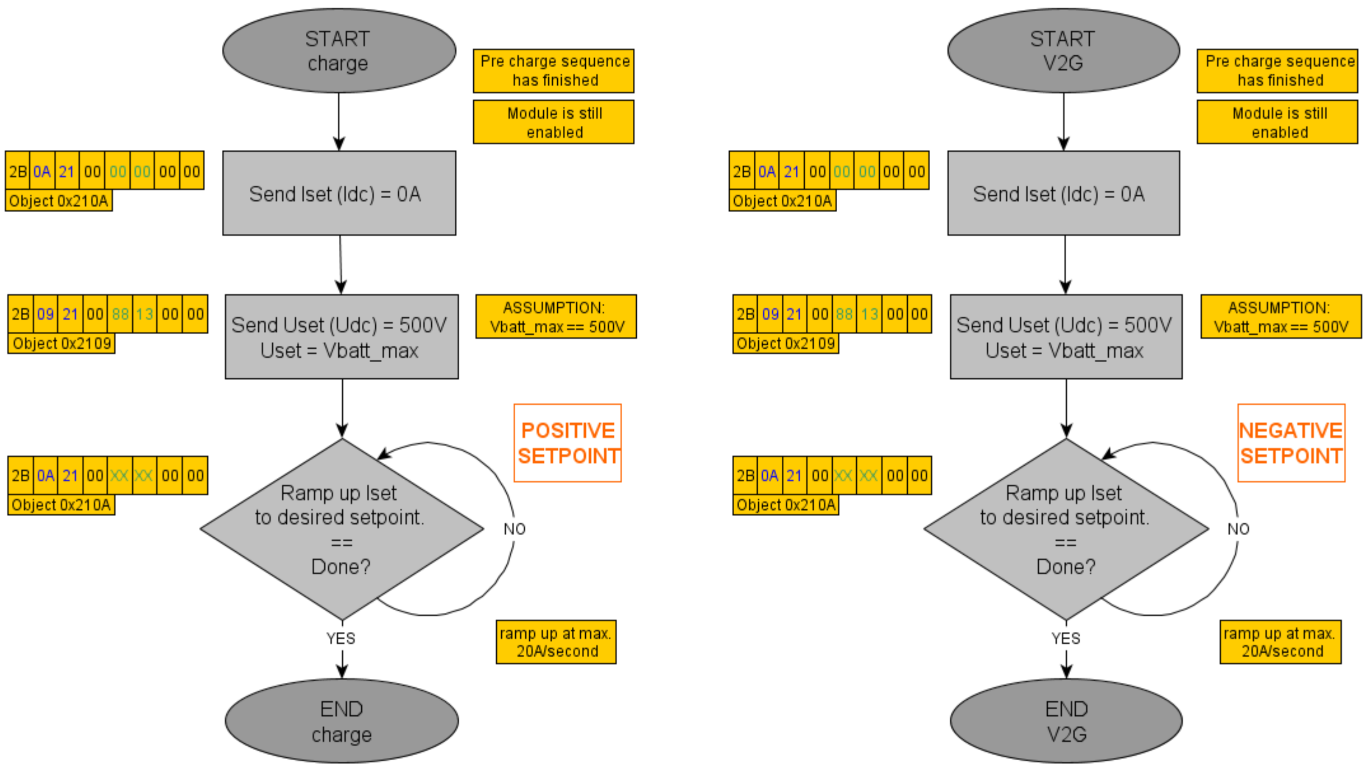

Figure 8: Charge & V2G mode

5.2.4 Stop Charge / V2G (without cable discharge)

1. To stop (dis)charging, send ‘**Disable**’ command
2. Set the output current setpoint to **0A**
3. Set the output voltage setpoint to **0V**

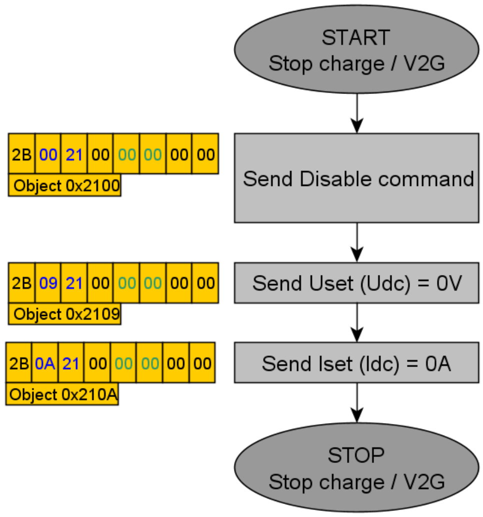

Figure 9: Stop charge & V2G

5.2.5 Stop Charge / V2G with cable discharge

1. Set the output current setpoint to **0A**
3. Open the DC contactors
4. Set the output current setpoint to **-1A** (this will discharge the output)
5. When the output voltage is low enough (e.g. lower than 50V), send **‘Disable’** command

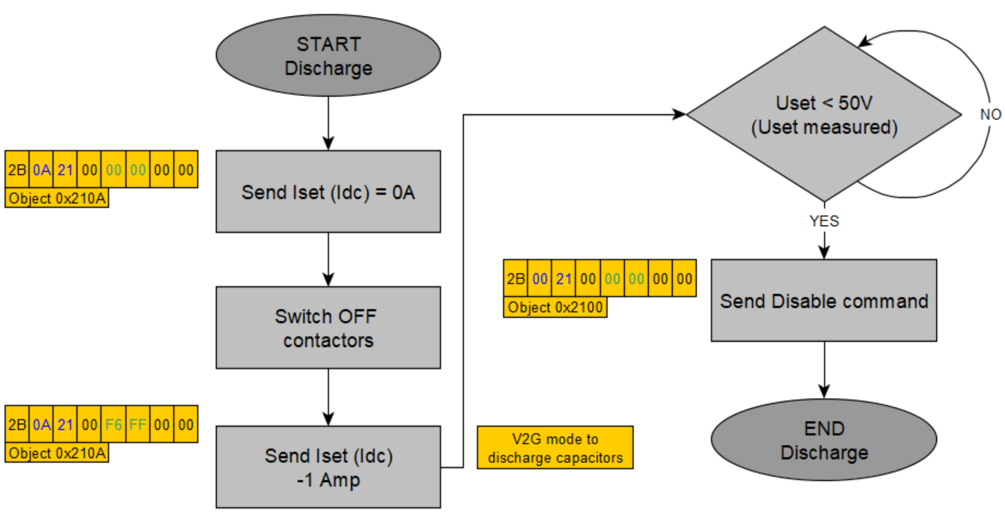

Figure 10: Discharge

5.3 Multiple power modules (stacking)

Multiple power modules can be stacked in parallel and connected to the same CANbus. When using multiple modules at lower power, it is advantageous to only use a subset of the available modules to improve overall system efficiency.

6 Special features

6.1 Derating (see also 4.2.21)

Derating (limiting output power) is a safety mechanism to protect the module from damaging or breaking down. The status can be read from CANopen object 0x2130 (see chapter 4)

6.1.1 Derating by temperature (optional, not enabled by default)

When the module is not able to keep the temperature below the maximum allowed temperature (e.g. the fans are blocked or defective or the intake air is too hot to cool effectively), the module will switch off to prevent damage.

Alternatively, to extend the operating range for higher temperatures (albeit with limited output power), derating by temperature can be configured for the module. This will cause the unit to derate when coming close to the maximum allowable temperature before switching off when that maximum temperature is exceeded. This is a factory programmed feature and can be requested when ordering the modules. For more info, contact PRE.

6.1.2 Derating by maximum power

When the module has a high setpoint in voltage (UDC) and current (IDC) and the maximum rated module power is exceeded (10kW), the maximum current (IDC) will decrease (foldback) to a level within the maximum specified output power.
Lowering the voltage and/or current, where the product of UDC and IDC is not greater than the maximum output power, will result in a recovered output current IDC.

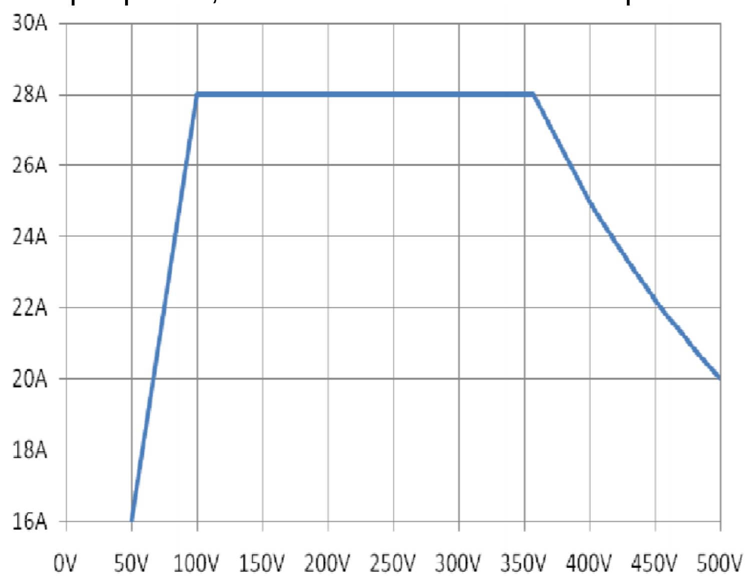

Figure 11: Derating by maximum power

6.2 Last switch-off reasons

The module stores its Switch Off Reason (SOR). As long as the module remains powered, the module will store its last 5 SOR's. E.g. this can be a user disable, a CANbus time-out, AC fault etc. The last SOR and the previous SOR need to be unique, e.g.: when the unit is disabled 2x in a row, only 1 is stored. This prevents erasing the SOR array to be flushed with the same SOR. See also

In addition, the last 5 Switch Off Failures (SOF, reasons for AC relays to disconnect) will be stored (object 0x2552) with a relative time-stamp (object 0x2553).

Switch off failure (SOF) – 0x2552

SI	Description	Value
1…5	Switch off failure (SI 1 is the most recent)	See Table 1

Switch off failure timestamp – 0x2553

SI	Description	Value
1…5	Timestamp (SI 1 is the most recent)	Seconds since start-up when SOF occurred

Every SOF report has 6 additional values available as shown in Table 1. Note that these errors are volatile and are not preserved when the device is power cycled.

SOR OBJ: 0x2552	**Error**	**Field 1** OBJ: 0x2218 SI: 1	**Field 2** OBJ: 0x2218 SI: 2	**Field 3** OBJ: 0x2218 SI: 3	**Field 4** OBJ: 0x2218 SI: 4	**Field 5** OBJ: 0x2218 SI: 5	**Field 6** OBJ: 0x2218 SI: 6
	SYSTEM:						
0x0001	Settings Error	-	-	-	-	-	-
0x0002	Settings Write Enabled	-	-	-	-	-	-
0x0003	WDT Error	-	-	-	-	-	-
0x0004	AC Trip	-	-	-	-	-	-
0x0005	Self-test	-	-	-	-	-	-
0x0006	Startup low	-	-	-	-	-	-
0x0007	Leakage ACDC	-	-	-	-	-	-
	DIR phase 1:						
0x0021	UVP Stage 1	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0022	UVP Stage 2	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0023	OVP Stage 1	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0024	OVP Stage 2	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0025	UFP Stage 1	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0026	UFP Stage 2	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0027	OFP Stage 1	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0028	OFP Stage 2	UAC1	UAC2	UAC3	F1	ROCOF	-

0x0029	ROCOF	UAC1	UAC2	UAC3	F1	ROCOF	-
0x002A	PhaseDeviation	UAC1	UAC2	UAC3	PH1	PH2	PH3
0x002B	LOM	UAC1	UAC2	UAC3	F1	ROCOF	-
0x002C	DC injection	UAC1	UAC2	UAC3	IDC1	IDC2	IDC3
	DIR phase 2:						
0x0041	UVP Stage 1	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0042	UVP Stage 2	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0043	OVP Stage 1	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0044	OVP Stage 2	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0045	UFP Stage 1	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0046	UFP Stage 2	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0047	OFP Stage 1	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0048	OFP Stage 2	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0049	ROCOF	UAC1	UAC2	UAC3	F1	ROCOF	-
0x004A	PhaseDeviation	UAC1	UAC2	UAC3	PH1	PH2	PH3
0x004B	LOM	UAC1	UAC2	UAC3	F1	ROCOF	-
0x004C	DC injection	UAC1	UAC2	UAC3	IDC1	IDC2	IDC3
	DIR phase 3:						
0x0061	UVP Stage 1	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0062	UVP Stage 2	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0063	OVP Stage 1	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0064	OVP Stage 2	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0065	UFP Stage 1	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0066	UFP Stage 2	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0067	OFP Stage 1	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0068	OFP Stage 2	UAC1	UAC2	UAC3	F1	ROCOF	-
0x0069	ROCOF	UAC1	UAC2	UAC3	F1	ROCOF	-
0x006A	PhaseDeviation	UAC1	UAC2	UAC3	PH1	PH2	PH3
0x006B	LOM	UAC1	UAC2	UAC3	F1	ROCOF	-
0x006C	DC injection	UAC1	UAC2	UAC3	IDC1	IDC2	IDC3

Table 1: Switch Off Reasons

7 Troubleshooting

This document provides troubleshooting procedures for the BiDi modules. It provides the procedures for troubleshooting hardware and software related problems. It also includes details for aiding troubleshooting by utilizing diagnostic and trace procedures.

7.1 General troubleshooting

You can use this troubleshooting as a checklist. No cross references are made.

7.1.1 Power

-Check if there is 230V / 400V (unless otherwise stated) AC connected, otherwise there can be no CANbus communication.
-Check that the neutral of the AC mains is properly connected

7.1.2 No CANbus communication

- Check the CANbus CAN High and CAN Low are not swapped
- Check if there is a 12V supply connected to power the CANbus (galvanically isolated)
- If the STANDBY pin is not controlled by the charge controller, use the DIP switch (see dip switches)

7.1.3 CANbus data, but no reply to messages

- Check you have the right node ID. If you get data from the module when sending a ‘broadcast’ message, but no reply with e.g. node ID 30, the module has a different node ID (default CAN node 30 -> CANID 0x630).

7.1.4 Module doesn’t switch ON

- Check if hardware Enable (Interlock) is high (12V), this is a safety line. The module will never enable without this pin high (see also dip switches).
- Check if Standby is switched to ON (see dip switches).

7.1.5 Module switches ON, then OFF

- Make sure you send CANbus data with an interval of at least <1000ms. The interval is also a ‘keep alive’ function. Better is to send CANbus data every 500ms.

7.1.6 Module does not make low UDC setpoint

This applies for modules without load. UDC setpoint is set to 100V, but if module not enabled or no IDC setpoint, the module will not 'make' UDC voltage. Otherwise the module will make its 100V UDC.

7.1.7 Module error handling

7.1.7.1 Module is Disabled -> Enabled

If the Status and Extended Status error bits are not set, the module can switch to power ON state. When either of the error bits are set, first remove the error source.

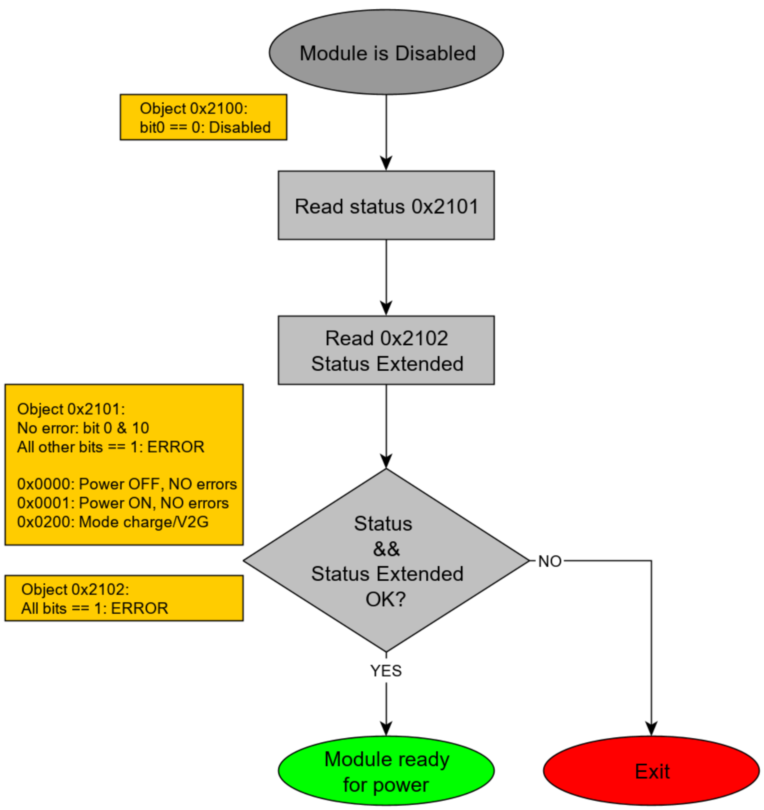

Figure 12: Error handling: Disabled -> Enabled

7.1.7.2 Module is Enabled -> Disabled

If the module is Enabled and then Disables, read the Status and SOR registers for root cause analysis (and log the faults). Note that an explicit 'Disable' command has to be send to the module after a power error.

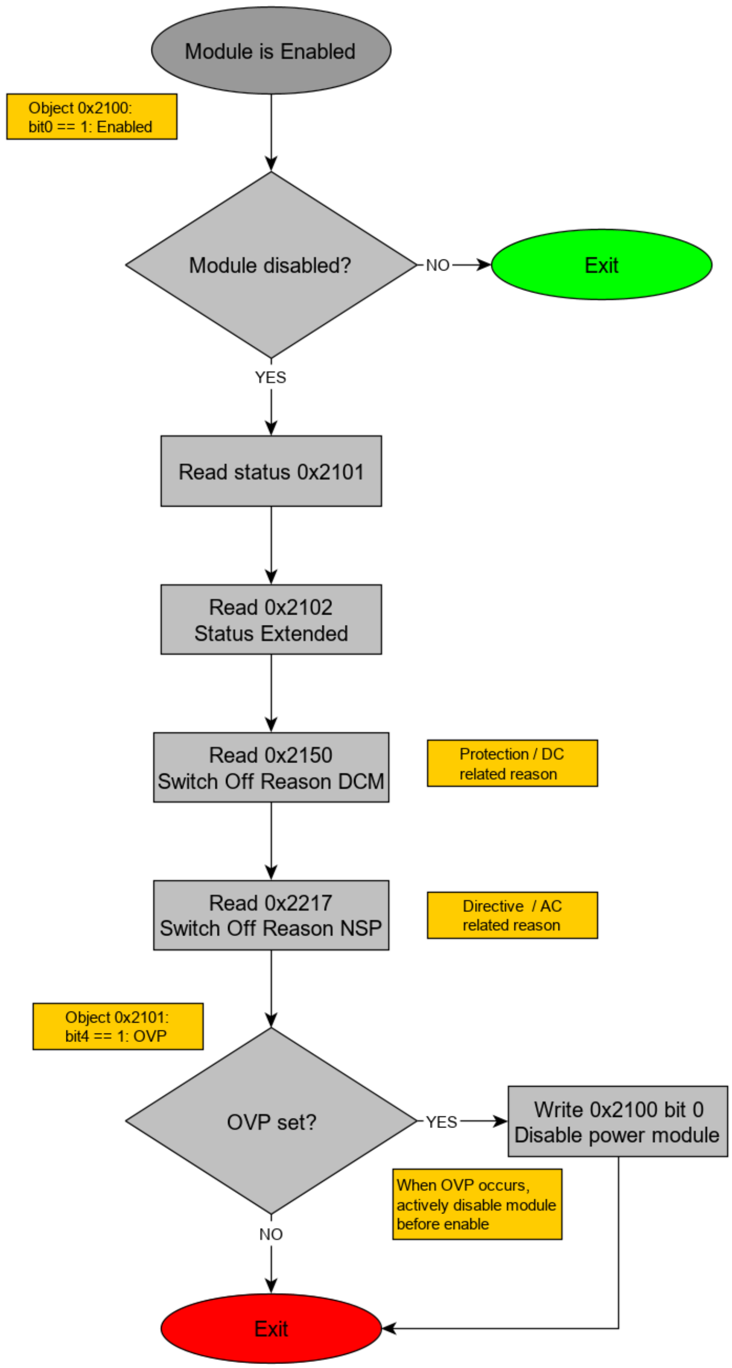

Figure 13: Error handling: Enabled -> Disabled

8 Examples

8.1 CANopen examples

See these references for CANopen messages.
Green is data, blue is object.

8.1.1 Object 0x2100: Power Module Enable

Example: Read 0x2100

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	40 00 21 00 00 00 00 00	Read 'enable'
RX	5B0	8	4B 00 21 00 01 00 00 00	Reply: module enabled

Example: Write 0x2100 (Enable module)

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	2B 00 21 00 01 00 00 00	Enable module = 1
RX	5B0	8	60 00 21 00 00 00 00 00	Acknowledge

8.1.2 Object 0x2101: Power Module Status

Example: Read 0x2101

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	40 01 21 00 00 00 00 00	Read status
RX	5B0	8	4B 01 21 00 01 00 00 00	Reply: charger on

Example: Write 0x2101

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	2B 01 21 00 00 00 00 00	Write attempt (data = 0)
RX	5B0	8	80 01 21 00 02 00 01 06	Attempt to write a read only object

8.1.3 Object 0x2104: Power Module Temperature

Example: Read 0x2104

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	40 04 21 00 00 00 00 00	Read temperature
RX	5B0	8	4B 04 21 00 FC 00 00 00	Reply: 25.2 °C

8.1.4 Object 0x2105: Power Module AC Input Voltage

Example: Read 0x2105

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	40 05 21 00 00 00 00 00	Read AC input
RX	5B0	8	4B 05 21 00 B4 0F 00 00	Reply: 402.0 V

8.1.5 Object 0x2106: Power Module AC Input Current

Example: Read 0x2106

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	40 06 21 00 00 00 00 00	Read status
RX	5B0	8	4B 06 21 00 96 00 00 00	Reply: 15.0 A

8.1.6 Object 0x2107: Power Module DC Output Voltage

Example: Read 0x2107

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	40 07 21 00 00 00 00 00	Read DC output voltage
RX	5B0	8	4B 07 21 00 7C 15 00 00	Reply: 550.0 V

8.1.7 Object 0x2108: Power Module DC Output Current

Example: Read 0x2108

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	40 08 21 00 00 00 00 00	Read DC output current
RX	5B0	8	4B 08 21 00 5A 00 00 00	Reply: 9.0 A

8.1.8 Object 0x2109: Power Module DC Output Voltage Setpoint

Example: Read 0x2109

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	40 09 21 00 00 00 00 00	Read V setpoint
RX	5B0	8	4B 09 21 00 88 13 00 00	Reply: 500.0 V

Example: Write 0x2109

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	2B 09 21 00 88 13 00 00	Set V = 500.0 V
RX	5B0	8	60 09 21 00 00 00 00 00	Acknowledge

8.1.9 Object 0x210A: Power Module DC Output Current Setpoint

Example: Read 0x210A

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	40 0A 21 00 00 00 00 00	Read A setpoint
RX	5B0	8	4B 0A 21 00 5A 00 00 00	Reply: 9.0 A

Example: Write 0x210A – CHARGE mode

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	2B 0A 21 00 5A 00 00 00	Set A = 9.0 A
RX	5B0	8	60 0A 21 00 00 00 00 00	Acknowledge

Example: Write 0x210A - V2G mode

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	2B 0A 21 00 A6 FF 00 00	Set A = -9.0 A
RX	5B0	8	60 0A 21 00 00 00 00 00	Acknowledge

8.1.10 Object 0x210B: Power Module AC max Input Current Setpoint

Example: Read 0x210B

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	40 0B 21 00 00 00 00 00	Read A setpoint
RX	5B0	8	4B 0B 21 00 5A 00 00 00	Reply: 9.0 A

Example: Write 0x210B

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	2B 0B 21 00 5A 00 00 00	Set A = 9.0 A
RX	5B0	8	60 0B 21 00 00 00 00 00	Acknowledge

8.1.11 Object 0x210D: Power Module DC Bus Voltage

This object contains the DC bus voltage, in steps of 0.1V.

Example: Read 0x210D

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	40 0D 21 00 00 00 00 00	Read A setpoint
RX	5B0	8	4B 0D 21 00 B8 0B 00 00	Reply: 300.0 V

Write 0x210E -> NOT ALLOWED

8.1.12 Object 0x210E: Solar Output Current

This object contains the Solar output current, in steps of 0.1A.

Example: Read 0x210E

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	40 0E 21 00 00 00 00 00	Read A setpoint
RX	5B0	8	4B 0E 21 00 5A 00 00 00	Reply: 9.0 A

Write 0x210E -> NOT ALLOWED

8.1.13 Object 0x2FFF: Reset

When this object is written, a reset of the node is performed.

Example: Read 0x2FFF

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	40 FF 2F 00 00 00 00 00	Read attempt
RX	5B0	8	80 FF 2F 00 01 00 01 06	Attempt to read a write only object

Example: Write 0x2FFF

Direction	ID (hex)	Length	Data (hex)	Comment
TX	630	8	23 FF 2F 00 00 00 00 00	Write object (data = 0)
RX	5B0	8	60 FF 2F 00 00 00 00 00	Acknowledge

9 Appendix

9.1.1 Extended object overview

These objects are not necessary for normal operation and therefore not further explained.

PRE reserves the right to change or update these objects in the future.

Object index	SI	Description	r/w	unit	size / type
220A		OBJ_NSP_SOFTW_VERSION	r	custom	uint32
2210		OBJ_NSP_STATUS	r	bits	uint32
2212		OBJ_NSP_SYS_ERROR	r	bits	uint32
2213		OBJ_NSP_PH1_DIRECTIVE_ERROR	r	bits	uint32
2214		OBJ_NSP_PH2_DIRECTIVE_ERROR	r	bits	uint32
2215		OBJ_NSP_PH3_DIRECTIVE_ERROR	r	bits	uint32
2217		OBJ_NSP_NSP_SOR	r	bits	uint32
2218	1-6	OBJ_NSP_SOR_STORED_VALUES	r	table	uint32
2220		OBJ_NSP_U_IN_230V_L1	r	0.1V	uint16
2221		OBJ_NSP_U_IN_230V_L1	r	0.1V	uint16
2222		OBJ_NSP_U_IN_230V_L1	r	0.1V	uint16
2223		OBJ_NSP_I_IN_L1	r	0.1A	uint16
2224		OBJ_NSP_I_IN_L2	r	0.1A	uint16
2225		OBJ_NSP_I_IN_L3	r	0.1A	uint16
2226		OBJ_NSP_F_IN_L1	r	0.01Hz	uint16
2227		OBJ_NSP_F_IN_L2	r	0.01Hz	uint16
2228		OBJ_NSP_F_IN_L3	r	0.01Hz	uint16
2229		OBJ_NSP_PH_IN_L1	r	0.1°	uint16
222A		OBJ_NSP_PH_IN_L2	r	0.1°	uint16
222B		OBJ_NSP_PH_IN_L3	r	0.1°	uint16
2240		OBJ_NSP_FB_U_OFP_V2X_THROTTLE	r	0.01%	uint32
2243		OBJ_NSP_FB_U_OFP_CHARGE_THROTTLE	r	0.01%	uint32
2260		OBJ_ACP_PH_IN_L1	r	°	uint16
2261		OBJ_ACP_PH_IN_L2	r	°	uint16
2262		OBJ_ACP_PH_IN_L3	r	°	uint16
2263		OBJ_ACP_VC_ELCO	r	V	uint16
2264		OBJ_ACP_VT_TAP	r	V	uint16
2265		OBJ_ACP_VH_ENABLE	r	V	uint16
2266		OBJ_ACP_STATE	r	bits	uint32