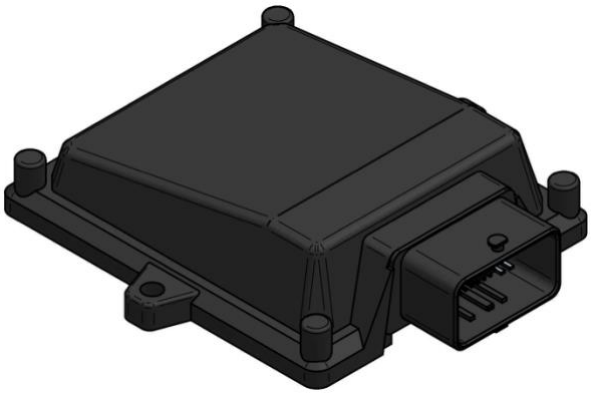
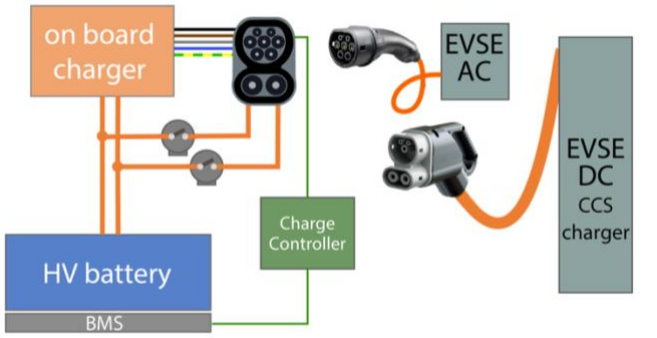
CCS charge controller user manual

Version 1.1 [18 februari 2026]



Fully integrated CCS-combo (ISO15118 / DIN1712) and AC (IEC61851) charge controller for use with EMUS BMS.

- Mode 1, 2, 3 and 4 charging
- Very low stand-by power consumption
- STM32F405 MCU
- 2x automotive CAN-bus
- 9...32V supply
- CP and PP wake-up
- Bidirectional input/ output wake-up pin
- 4x half-H bridge driver outputs
- 2x HV charge contactor drivers + feedback
- Charge lock actuator drivers + feedback
- 12V RGB LED driver
- 2x Analog inputs (stop charge and configurable)
- 3x PT1000 temperature inputs



This evaluation kit is intended for use for ENGINEERING DEVELOPMENT, DEMONSTRATION, OR EVALUATION PURPOSES ONLY and is not considered to be a finished end-product fit for general consumer use. Persons handling the product(s) must have electronics training and observe good engineering practice standards. As such, the goods being provided are not intended to be complete in terms of required design-, marketing-, and/or manufacturing-related protective considerations, including product safety and environmental measures typically found in end products that incorporate such semiconductor components or circuit boards. This evaluation kit does not fall within the scope of the European Union directives regarding electromagnetic compatibility, restricted substances (RoHS), recycling (WEEE), FCC, CE or UL, and therefore may not meet the technical requirements of these directives or other related directives.

You undertake your project at your own risk.

You should satisfy yourself that you are safely install and troubleshoot electrical systems and satisfy any legal requirements you may be subject to before beginning the project.

The information provided in this document and the support provided is intended as information only. MISMO tech and EV-Europe take no responsibility for how you use the information, nor any liability for injuries, or death, that may result from your actions.

You undertake your project at your own risk.

You should satisfy yourself that you are safely install and troubleshoot electrical systems and satisfy any legal requirements you may be subject to before beginning the project.

The information provided in this document and the support provided is intended as information only. MISMO·tech takes no responsibility for how you use the information, nor any liability for injuries, or death, that may result from your actions.

Table of contents

You undertake your project at your own risk.

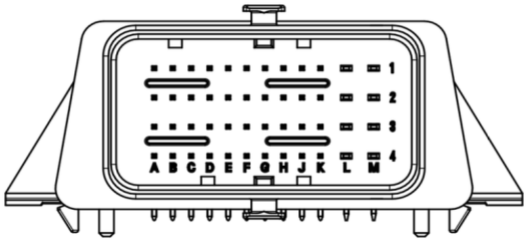
You should satisfy yourself that you are safely install and troubleshoot electrical systems and satisfy any legal requirements you may be subject to before beginning the project.

The information provided in this document and the support provided is intended as information only. MISMO·tech takes no responsibility for how you use the information, nor any liability for injuries, or death, that may result from your actions.

1 Pinout

	A	B	C	D	E	F	G	H	J	K	L	M	
1	CAN1 HI (BMS)	CAN1 HI	TEMP 1 (PT1000)	TEMP 3 (PT1000)	ANALOG IN 1 (Stop charge)	OUTPUT 1 (DC charging isol monitor disable)	POSITIVE CONTACTOR FEEDBACK	LED RED	LOCK +	POSITIVE CONTACTOR	SUPPLY VOLT (constant 9-32V)	GND	1
2	CAN1 LO	CAN1 LO	TEMP 1 RETURN	TEMP 3 RETURN	ANALOG IN 2	OUTPUT 2 (AC charger cooling enable)	CONTACTOR FEEDBACK RETURN	LED GREEN	LOCK -	CONTACTOR RETURN	WAKEUP (input + output)	PE*	2
3	CAN2 HI (service)	CAN2 HI	TEMP 2 (PT1000)			OUTPUT 3	NEGATIVE CONTACTOR FEEDBACK	LED BLUE	LOCK FB	NEGATIVE CONTACTOR		CP	3
4	CAN2 LO	CAN2 LO	TEMP 2 RETURN		PP DETECT (charge interlock)	OUTPUT 4	CONTACTOR FEEDBACK RETURN	LED RETURN	LOCK FB RETURN	CONTACTOR RETURN		PP	4

*PE needs an external solid connection to chassis GND!



2 Connector

MOLEX 48P Automotive connector	**Part nr.**
Wire harness female CMC Receptacle 48P	643201311
Dust cover/ cap for CMC Receptacle 48P	643201301
40x CP 0.6mm female terminal 0.5mm^2	643221039
40x CP 0.6mm female terminal 0.22-0.44mm^2	643221019
40x CP 0.6mm female terminal 0.75mm^2	643221029
8x CP 1.5mm female terminal 1.0mm^2- 2.0mm^2	643231039
8x CP 1.5mm female terminal 0.50mm^2 - 1.0mm^2	643231029
PCB through hole solder connector male	5007620481

Carefully read Molex CMC assembling manual before starting to connect wires!

https://www.molex.com/content/dam/molex/molex-dot-com/products/automated/en-us/applicationspecificationspdf/984/98420/AS-98420-002-001.pdf?inline

Use the correct crimping tool to avoid bad connection or issues with inserting the pins in the housing.

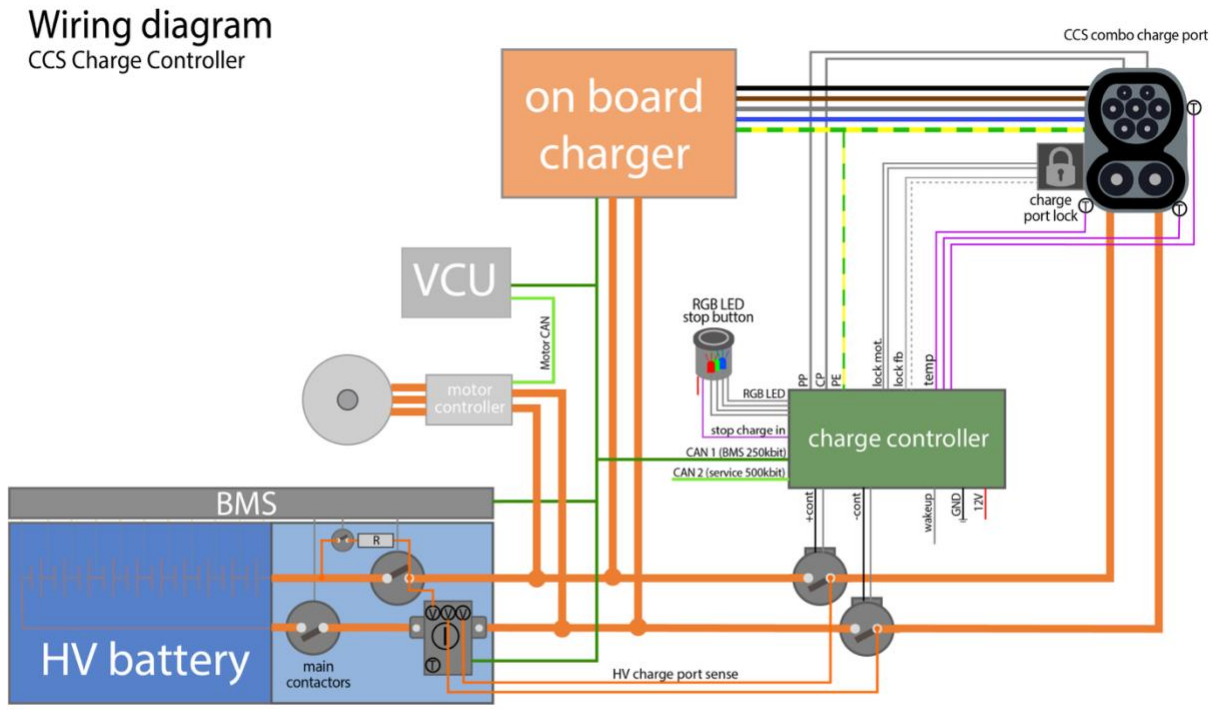
Check all pin connections with a multimeter after assembling.

You undertake your project at your own risk.

You should satisfy yourself that you are safely install and troubleshoot electrical systems and satisfy any legal requirements you may be subject to before beginning the project.

The information provided in this document and the support provided is intended as information only. MISMO·tech takes no responsibility for how you use the information, nor any liability for injuries, or death, that may result from your actions.

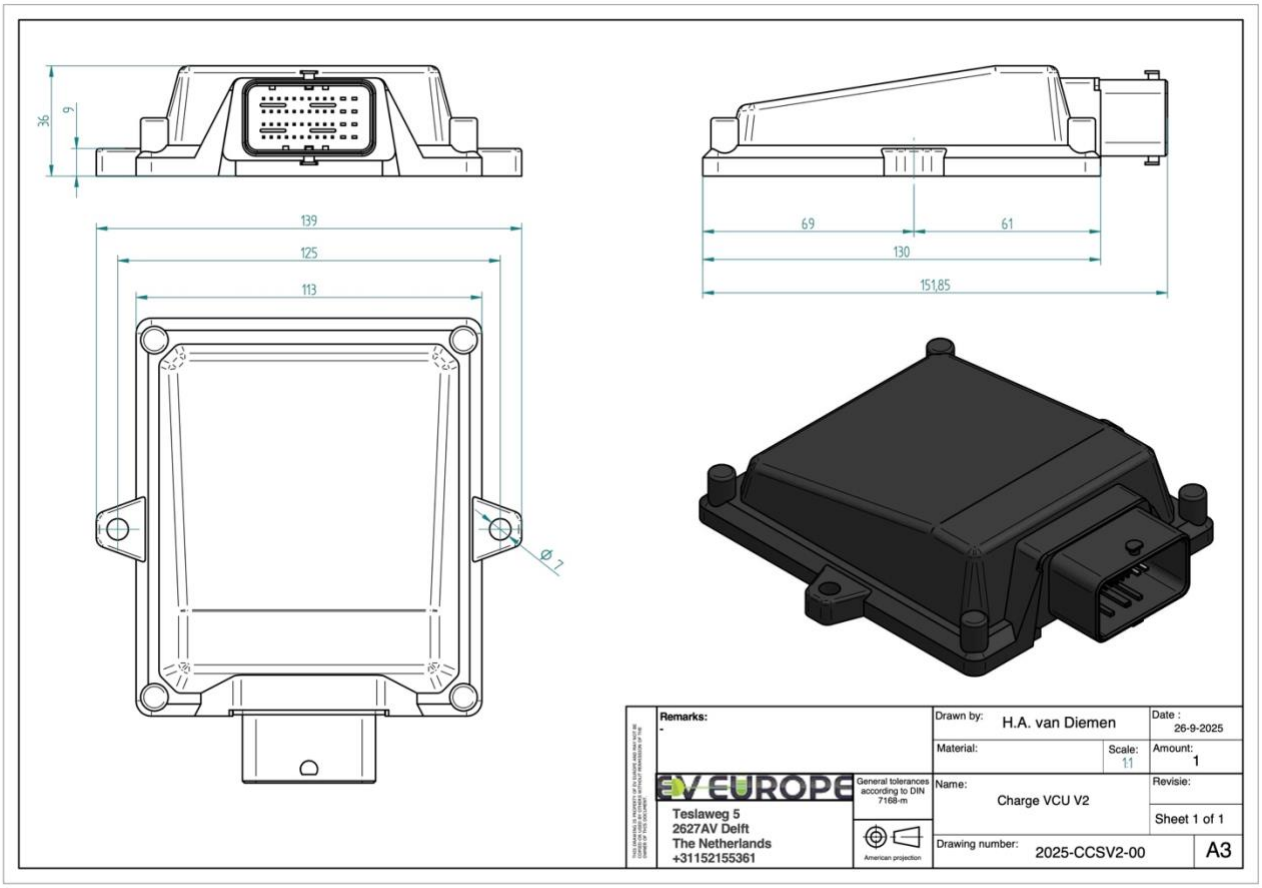
3 Wiring diagram



4 RGB LED status codes

- standby
- no charge
- connected initialisation
- charging active
- charging finished
- unlocked
- charge error

5 Mechanical drawing



The charge controller should be positioned close to the CCS inlet and kept away from high-power components, such as the on-board charger, motor controller, and high-power cables, to prevent electromagnetic interference (EMI) ensuring reliable operation.

6 Connections

6.1 Power supply

Connect to constant 9 to 32V battery.

Place a 10Amp fuse in the supply.

6.2 Ground

Connect to negative battery terminal / chassis.

PE needs an external solid connection to chassis GND!

6.3 Wake up

Bidirectional input - output pin.

Charge controller wakes up if…

- Charge plug is connected (PP is detected)
- CP PWM signal is detected
- Stop charge button is pressed
- Wake-up pin is pulled high by another device
- USB 5V power is detected (for SW update)

Charge controller pulls line high…

- for 10 seconds after one of the wake-up triggers
- for 1 minute during charging session initialization (PP is detect)
- while charging is active

In the normal configuration the charge controller goes back to sleep after charging is finished even if the cable stays plugged in. If you want to keep it awake until the cable is unplugged the solder jumper JP4 can be soldered.

If the motor controller needs the drive interlock signal over CAN bus the ignition signal should be used to wake up the charge controller to make sure the CAN messages are sent when the motor controller is enabled.

6.4 CAN bus

>> Avoid drop cables >30cm in the CAN bus.

>> Make sure you have two 120-ohm termination resistors connected on each CAN bus!
The on-board termination resistors can be enabled with a solder jumper on the PCB or the loop trough CAN pins in the connector can be connected to a 120-ohm resistor.

6.4.1 CAN 1 (250kbit)

The charge controller communicates on CAN1 with the BMS and the Isabelenhütte IVT sensor. during While CCS fast charging the charge controller emulates a J1939 charger.
During boot various BMS configuration parameters are requested which are used to calculate charge and discharge current/ voltage limiting curves. See Charge current limits section bellow.

6.4.2 CAN 2 (500kbit)

CAN2 is for service, debugging and CMS parameter change.

Some motor controllers like the Zonic need a drive interlock over CAN bus and can be connected to this bus. A separate CAN gateway between the BMS CAN and the motor controller CAN is recommended!

6.5 Temperature input

Tree inputs for a PT1000 thermistor are available.

6.6 Inputs

Two analog 0-32V inputs are available.

- Input 1: stop charge
- Input 2:

6.7 Outputs

Four 1A general purpose high or low side driver outputs are available.

- Output 1: DC charging active (disable on board insulation monitoring)
- Output 2: AC charger cooling enable
- Output 3:
- Output 4:
- **PP detect (drive interlock)** high side output to disable the motor when a plug is inserted. This output might also be used to enable the BMS.
  PP detect is switched even if the microcontroller is off.

6.8 USB-C connector

The USB-C connection can only be accessed if the automotive enclosure is removed and is for software updates only.

The charge controller is also powered and woken up from the USB 5V power.

6.9 Debug connector

STLINK-V3MINI STDC14 connector is for debugging and programming.

6.10 Charge port

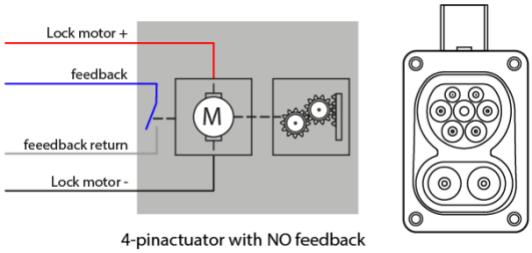
A DUOSIDA CCS2 charge port is recommended.

Communication wires PP (proximity pilot), CP (control pilot), PE (Earth)
and 2 or 3 temperature inputs need to be connected to the charge controller.

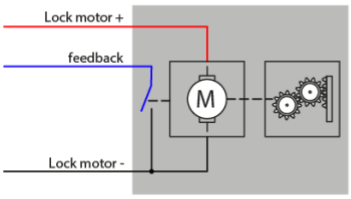
6.11 Charge port lock actuator

A 3 or 4-pin servo lock motor is recommended. Preferably no solenoid version!
Standard configuration is for NO feedback. (closed when locked)

In the standard configuration Phoenix inlets with 1k/10k feedback status resistors are not supported.



4-pinactuator with NO feedback



3-pin actuator with NO feedback

6.12 DC charge contactors

Use contactors with integrated economizer.

Feedback contact pins are available from charge controller hardware V1.1 onwards.

KILOVAC EV200 Series with built-in coil economizer and auxiliary contacts for feedback are recomended.
(12V or 24V version depending on system voltage)

max 4A inrush current

max 0.5A continuous current

You undertake your project at your own risk.

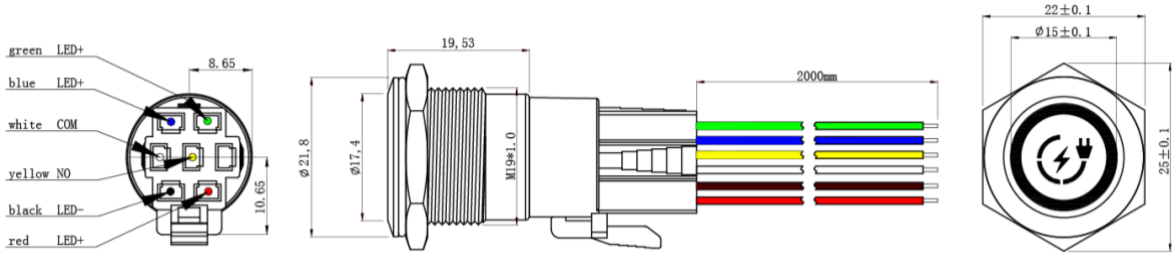
You should satisfy yourself that you are safely install and troubleshoot electrical systems and satisfy any legal requirements you may be subject to before beginning the project.

The information provided in this document and the support provided is intended as information only. MISMO·tech takes no responsibility for how you use the information, nor any liability for injuries, or death, that may result from your actions.

6.13 Stop push button and charge indication LED

The momentary switch can be connected to input 1 to terminate the charging process.

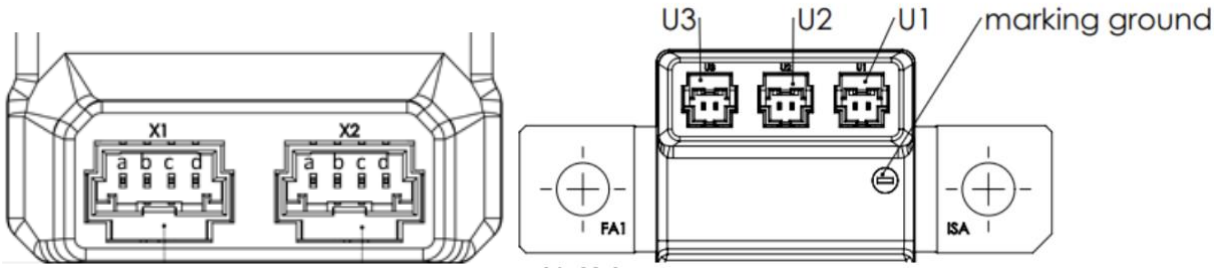
The RGB charge indicator LED should have a common cathode (negative) terminal and integrated series resistors for 12V / 24V DC.



6.14 DC charge port voltage sensing

It is recommended to measure the DC inlet voltage to compare the charger voltage and the pack voltage before closing the DC charge contactors. To detect a bad connection (high voltage drop) across the inlet or the cable or a welded contactor.

The Isabellenhütte IVT sensor with 3 voltage measure channels is recommended. When wired according to the schematics it can be used to measure the battery current, the voltage after the main battery contactors (zwischenkreis) and the charge port voltage.



X1 and X2

Pin	Description
a	Vcc
b	CAN L
c	CAN H
d	GND

Connector	Function	Pin
U1	DC bus +	Both
U2	Charge port -	Both
U3	Charge port +	Both

6.14.1 Isabellenhütte configuration

To change the CAN speed to 250kbit and enable periodic transmission of the voltages we need to send the following manual messages to the sensor.

Connect the sensor to a CAN network which only the PEAK CAN, the Isabellenhütte IVT sensor and two terminators. Use SavyCAN V221 or later to load the frame sender template shown below.

Each message needs to be sent once and wait for 0x511 reply. See Isabellenhütte manual for details.

Frame Sender

	En	Bus	ID	MsgName	Len	Ext	Rem	Data	Trigger
1		0	0x411	Stop	08			0x34 0x00 0x01 0x00 0x00 0x00 0x00 0x00	1m
2		0	0x411	Enable Voltage 1 periodic 60ms little endian	08			0x21 0x42 0x00 0x3C 0x00 0x00 0x00 0x00	1m
3		0	0x411	Enable Voltage 2 periodic 60ms little endian	08			0x22 0x42 0x00 0x3C 0x00 0x00 0x00 0x00	1m
4		0	0x411	Enable Voltage 3 periodic 60ms little endian	08			0x23 0x42 0x00 0x3C 0x00 0x00 0x00 0x00	1m
5		0	0x411	Enable Temperature periodic 1000ms little endian	08			0x24 0x42 0x08 0x3E 0x00 0x00 0x00 0x00	1m
6		0	0x411	Enable Wh periodic 60ms little endian	08			0x27 0x42 0x00 0x3C 0x00 0x00 0x00 0x00	1m
7		0	0x411	Store	08			0x32 0x00 0x00 0x00 0x00 0x00 0x00 0x00	1m
8		0	0x411	CAN speed 250kbit	08			0x3A 0x08 0x00 0x00 0x00 0x00 0x00 0x00	1m
9		0	0x411	Start	08			0x34 0x01 0x01 0x00 0x00 0x00 0x00 0x00	1m
10									

After programming load the Isabellenhuete IVT-S_LittleEndian.dbc (must be higher in the dbc list than EMUS dbc!) and check if you periodically receive the following messages and if the values are reasonable.

ID	Ext	Cnt	Dir	Bus	Len	ASCII	Data
0x521	0	817	Rx	0	6	.....	00 02 B9 B3 FF FF <IVT_Msg_Result_I> IVT_ID_Result_I: Vt_Result_I IVT_MsgCount_Result_I: 2 IVT_Result_I: -19.527 A IVT_Result_I_Channel_Error: 0 IVT_Result_I_Measurement_Error: 0 IVT_Result_I_OCS: 0 IVT_Result_I_System_Error: 0
0x522	0	409	Rx	0	6	.....	01 0D E1 81 05 00 <IVT_Msg_Result_U1> IVT_ID_Result_U1: Vt_Result_U1 IVT_MsgCount_Result_U1: 13 IVT_Result_U1: 360.929 V IVT_Result_U1_Channel_Error: 0 IVT_Result_U1_Measurement_Error: 0 IVT_Result_U1_OCS: 0 IVT_Result_U1_System_Error: 0
0x523	0	13...	Rx	0	6	.....	02 05 E4 FF FF FF <IVT_Msg_Result_U2> IVT_ID_Result_U2: Vt_Result_U2 IVT_MsgCount_Result_U2: 5 IVT_Result_U2: -0.028 V IVT_Result_U2_Channel_Error: 0 IVT_Result_U2_Measurement_Error: 0 IVT_Result_U2_OCS: 0 IVT_Result_U2_System_Error: 0
0x524	0	13...	Rx	0	6	.....	03 05 CF 81 05 00 <IVT_Msg_Result_U3> IVT_ID_Result_U3: Vt_Result_U3 IVT_MsgCount_Result_U3: 5 IVT_Result_U3: 360.911 V IVT_Result_U3_Channel_Error: 0 IVT_Result_U3_Measurement_Error: 0 IVT_Result_U3_OCS: 0 IVT_Result_U3_System_Error: 0
0x525	0	13...	Rx	0	6	.....	04 00 CA 00 00 00 <IVT_Msg_Result_T> IVT_ID_Result_T: Vt_Result_T IVT_MsgCount_Result_T: 0 IVT_Result_T: 20.2 ◆C IVT_Result_T_Channel_Error: 0 IVT_Result_T_Measurement_Error: 0 IVT_Result_T_OCS: 0 IVT_Result_T_System_Error: 0
0x528	0	13...	Rx	0	6	.....	07 00 01 FD FF FF <IVT_Msg_Result_Wh> IVT_ID_Result_Wh: Vt_Result_Wh IVT_MsgCount_Result_Wh: 0 IVT_Result_Wh: -0.767 kWh IVT_Result_Wh_Channel_Error: 0 IVT_Result_Wh_Measurement_Error: 0 IVT_Result_Wh_OCS: 0 IVT_Result_Wh_System_Error: 0

6.15 Hardware versions

PCB V1.4

- USB-C which is invertable

PCB V1.3

- fix schottky diode leakage (VBUS, Keep power on GPIO...)
- AI1 en AI2 protection diodes replaced
- contactor and lock feedback inputs with TVS diode protected
- New more accurate temperature with input rail to rail linear amplifier
- fix shared resistor for RGB LED which allows color mixing
- smaller RGB LED
- fix 5V LDO ENABLE not connected
- trough hole BOOT0 jumper
- Hardware version detect resistor 15K

PCB V1.2

- new temperature input amplifier circuit
- RGB status LED on PCB
- new 5V LDO
- BOOT0 solder jumper
- fix USB VBUS detect and wake-up from VBUS
- new ST-LINK debug connector STLINK-V3MINI_STDC14 (no more board-to-board connector)
- Hardware version detect resistor 12K

PCB V1.1

- wake-up from PP or CP pulse detection
- fix PP detect / wake-up op-amp circuit
- Hardware version detect resistor 10K

7 CAN bus communication

The following messages are transmitted and received on CAN1 (BMS CAN 250kbit).

See MISMOtech_VCU_V11.DBC or higher for CAN decoding and further details.

7.1 Transmitting CAN message

To understand why a certain charging speed is calculated the charge controller VCU broadcasts a CAN message with the charge/ discharge current limits and the reasons for it.

7.1.1 0x19B50409 Charge limits			**broadcasted every 100ms**
byte 0-1	Charge_Current_Limit	*0.1	Amps DC
byte 2	Charge_Current_Limit_Reason		enum
	0 = No limit		
	2 = Low Cell Temp		
	3 = High Cell Temp		
	4 = Low Cell Voltage		
	5 = High Cell Voltage		
	6 = Charging Station DC Limit		
	7 = User DC Fast Charge Limit		
	8 = Hard Coded DC Charge Limit		
	9 = BMS Fast Charge Current		
	10 = BMS Slow Charge Current		
	11 = BMS Charger Control Limit		
	12 = PP Cable Current Limit		
	13 = CP EVSE Current Available		
	14 = User AC Current Limit		
	15 = BMS Not Sending J1939 Charger Control Message		
	20 = Charging stopped		
	21 = Charging paused		
	22 = Emergency stop pressed		
	23 = BMS not responding		
	24 = Waiting for BMS parameters		
	25 = Battery Contactor Open		
	26 = Battery Contactor Going To Open		
	27 = BMS Charging Error		
	28 = BMS Warning Active		
	29 = BMS Protection Active		
	30 = BMS Warning Active		
	31 = Static Pilot (EVSE paused charge)		
	32 = No Pilot (Mode 1 charging 13A)		
	33 = Charging error		
	34 = Regen limit (when no cable connected)		
	255 = Signal not available		
byte 3	Charge_Current_Percentage		% of max charge current
byte 4-5	Charge_Voltage_Limit	*0.1	Volts DC
byte 6	bit 0-3: Battery_State		enum
	0 = Not_Ready		
	1 = Ready		
	2 = Ready_Reduced_Power		
	3 = Battery_Warning		
	4 = Waiting_For_Save_Shut_Down		
	5 = Emergency_Stop_Shutting_Down		
	6 = Emergency_Stop_Contactors_Open		
	14 = BMS_Not_Responding		
	15 = Signal_not_available		
	bit 4-5: Battery_Ready_To_Charge		enum
	0 = Not_Ready		
	1 = Ready		
	2 = Ready_with_reduction		
	3 = signal_not_available		
	bit 6-7: Charging_Cable_Connected		enum
	0 = Disconnected		
	1 = Connected		
	3 = Signal_not_available		

7.1.2 0x19B5040A Discharge limits			**broadcasted every 100ms**
byte 0-1	Discharge_Current_Limit	*0.1	Amps DC
byte 2	Discharge_Current_Limit_Reason		enum
	0 = No limit		
–	~~1 = High State of Charge~~	–	–
	2 = Low Cell Temp		
	3 = High Cell Temp		
	4 = Low Cell Voltage		
	5 = High Cell Voltage		
	6 = Charging Station DC Limit		
	22 = Emergency stop pressed		
	23 = BMS not responding		
	24 = Waiting for BMS parameters		
	25 = Battery Contactor Open		
	26 = Battery_Contactor_Going_To_Open		
	29 = BMS_Protection_Active		
	30 = BMS Warning Active		
	35 = Drive current limit		
	255 = Signal not available		
byte 3	Discharge_Current_Percentage		% of max discharge current
byte 4-5	Discharge_Voltage_Limit	*0.1	Volts DC
byte 6	bit 0-3: Battery_State		enum
	0 = Not_Ready		
	1 = Ready		
	2 = Ready_Reduced_Power		
	3 = Battery Warning		
	4 = Waiting_For_Save_Shut_Down		
	5 = Emergency_Stop_Shutting_Down		
	6 = Emergency_Stop_Contactors_Open		
	14 = BMS Not Responding		
	15 = Signal_not_available		
	bit 4-5: Battery_Ready_To_Discharge		enum
	0 = Not_Ready		
	1 = Ready		
	2 = Ready_with_reduction		
	3 = signal_not_available		
	bit 6-7: Charging_Cable_Connected		enum
	0 = Disconnected		
	1 = Connected (drive interlock)		
	3 = Signal_not_available		

7.1.3 0x15601060 Charge controller status 1			**broadcasted every second while plugged in**
byte 0	User_AC_Current_Limit		1...63 Amps AC
~~byte 1~~	~~Second_AC_Current_Limit~~	–	~~not used~~
byte 2	CP_charging_station_Limit		6...63 Amps AC
byte 3	PP_charging_cable_Limit		6...63 Amps AC
byte 4-5	Measured_AC_Current	*0.1	(not available with TC charger)
byte 6	Bit 0-3: CP_Pilot_Status		Enum
	0 = initialize		
	1 = A: static pilot		
	2 = B: AC limit, EV not ready (PWM)		
	3 = C: AC limit, EV ready		
	4 = D: AC limit, EV ready w. fan		
	5 = E: No pilot (no power)		
	6 = F: Pilot Error (-12V)		
	7 = PLC comm, EV not ready (PWM 5%)		
	8 = PLC comm, EV ready (PWM 5%)		
	Bit 4-7: AC_phase_count		0...3
byte 7:	bit 0: Charging_Cable_Detected		bool
	bit 1: Charging_Cable_Locked		bool
	bit 2: Charging_Allowed		bool
	bit 3: reserved		bool
	bit 4: Pilot_Error		bool
	bit 5: Stop_Charge		bool
	bit 6: Pause_Charge		bool
	bit 7: reserved		bool

7.1.4 0x15601260 Charge controller status 2			**broadcasted every second while plugged in**
byte 0	Remaining_Charging_Time_Estimation	*2	minutes
byte 1	User_DC_Current_Limit		Amps DC
byte 2	Charging_Station_DC_Current_Limit		Amps DC
byte 3	Actual_Charger_DC_Current		Amps DC
byte 4-5	Actual_Charger_DC_Voltage		Volts
byte 6	CCS_Charging_State		Enum
	0 = Standby		
	1 = Initialisation		
	2 = PreCharge		
	3 = EnergyTransfer		
	4 = Shutdown		
	9 = CableTest		
byte 7	Charger coolant temperature (AC) CCS inlet temperature (DC)	+55	°C

7.2 Receiving CAN message

The following charge control message can be sent from the user dashboard screen (HMI) or remote app cellular controller to control the charging process.

7.2.1 0x15602060 User charge control message			**Charge controller confirms parameters with same message as response**
byte 0	User_AC_Current_limit		1...63 Amps AC 0xFF=do not change
byte 1	Pause_Charging		0=charging 1=pause 0xFF=do not change
byte 2	Stop_Charging		0=charging 1=stop charge and unlock plug 0xFF=do not change
byte 3	Pause_Charge_SoC		1…99 pause SoC 0=100=fully charge 0xFF=do not change
byte 4/5	User_DC_Current_limit	*0.1	Amps DC 0=no limit 0xFFFF=do not change
byte 6/7	reserved		

The VCU is responding with the same message to confirm the change and share the current values.

Use 0xFF if a parameter should stay unchanged. Send message with all bytes 0xFF to get the current values from the VCU.

You undertake your project at your own risk.

You should satisfy yourself that you are safely install and troubleshoot electrical systems and satisfy any legal requirements you may be subject to before beginning the project.

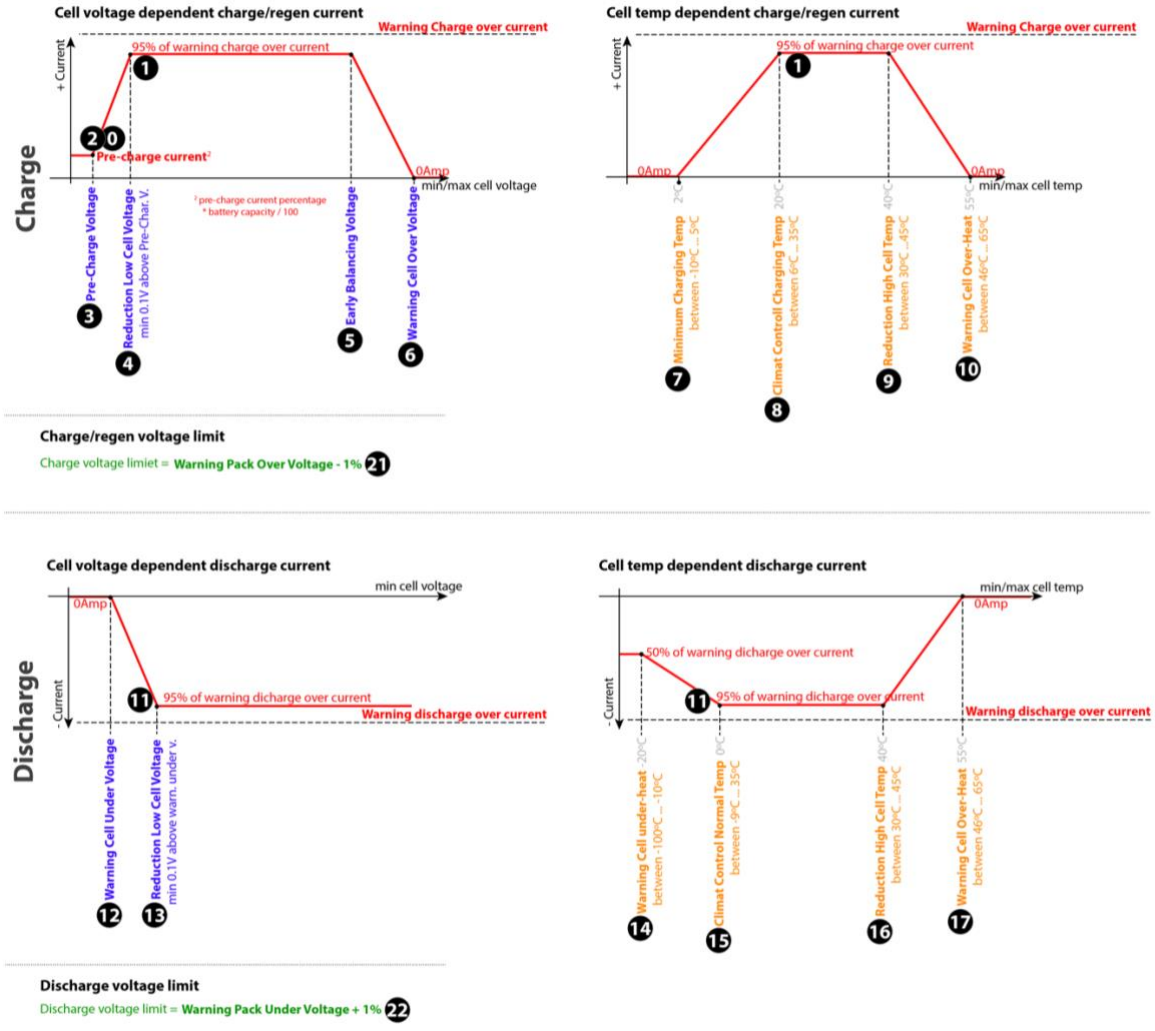
The information provided in this document and the support provided is intended as information only. MISMO·tech takes no responsibility for how you use the information, nor any liability for injuries, or death, that may result from your actions.

8 Charge current limits

For optimal battery safety, it's crucial to determine the appropriate maximum charging current limits. These limits vary based on factors like battery size, chemistry, type, health, temperature, voltage, and other system and application-specific considerations. Ensure accurate monitoring of cell voltage and temperature across the battery pack and implement effective thermal management to maintain cells within safe operational ranges. Adhere to the maximum ratings for charge inlets, wiring, and contactors. Charging with excessive currents or voltages can lead to rapid aging, permanent damage, or even cause a battery fire.

In EMUS BMS you can configure the maximum charging current. However, as EMUS lacks temperature and cell voltage-compensated charge current limits, which are essential for fast charging, these must be calculated in the VCU. To keep configuration simple, we use some EMUS parameters you can configure in the EMUS Control Panel for the compensated current limits.
Carefully study how the limiting curves are defined and configure the corresponding EMUS parameters. Note that the VCU only reads out the marked EMUS parameters during boot-up. Min and max cell voltage and temperatures is requested every second to recalculate the compensated charge and discharge current limits.

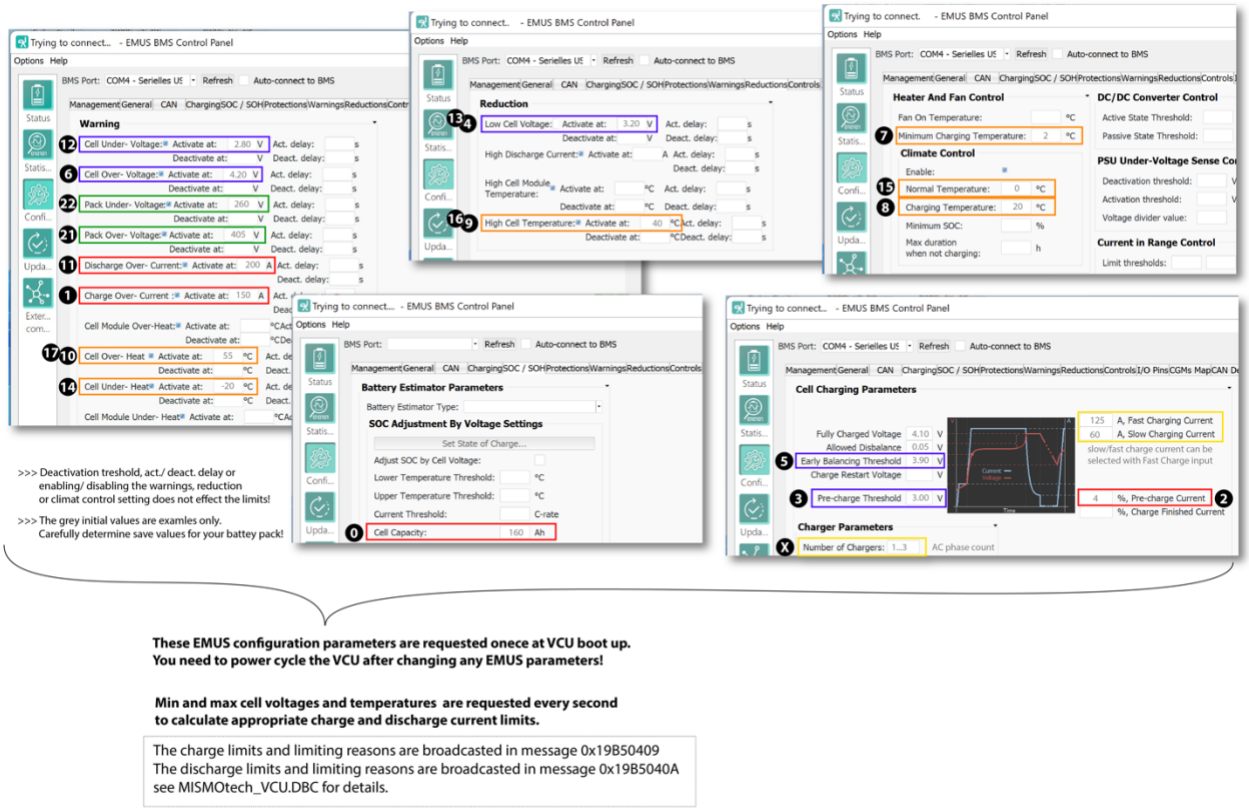
8.1 Temperature and voltage compensated charge / discharge limiting curves



You undertake your project at your own risk.

You should satisfy yourself that you are safely install and troubleshoot electrical systems and satisfy any legal requirements you may be subject to before beginning the project.

The information provided in this document and the support provided is intended as information only. MISMO·tech takes no responsibility for how you use the information, nor any liability for injuries, or death, that may result from your actions.



For debugging and to understand which limit is currently restricting the charging power connect a CAN bus logger and inspect the CAN protocol in section 0x19B50409 Charge limits or use the MISMOtech_VCU_V11.DBC or higher with SavvyCAN.

>> Make sure you power cycle the VCU after updating any of these EMUS parameters.

8.2 AC charge current limits

When AC charging, we get an AC current limit from the charging station (EVSE) called the pilot limit (CP) and a charging cable limit (PP). Because most on board chargers are controlled with a DC current request we need the AC phase count, grid voltage and charger efficiency to calculate the DC current limit.

```
DC_current = AC_voltage * CP_PP_current_limit * phase_count * charger_efficiency / DC_voltage
```

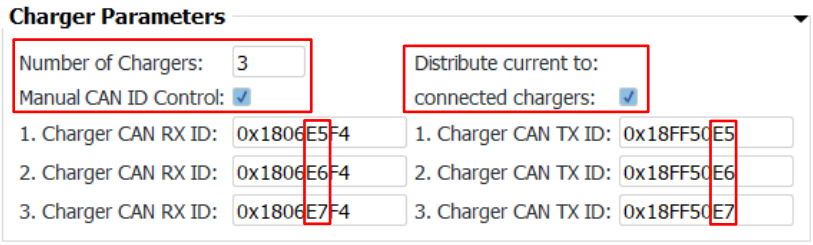
For the phase count we support 1 and 3-phase setups. If individual chargers are used for each phase, we count the charger CAN ID's which are broadcasting. **Each phase needs a different ID!**

3-phase chargers like the Infy share the connected phase count over CAN bus.

Chargers which work with the J1939 protocol like the TC chargers do not share the AC measurements which means we can only use a hard coded value for the AC voltage.
The VCU uses 230VAC and 95% charger efficiency if no other values are available.

EMUS charger configuration

Number of Chargers should correspond with the phases you're using. (1 or 3 is supported!)



1. Charger CAN RX ID = **0x1806E5F4**
2. Charger CAN RX ID = **0x1806E7F4**
3. Charger CAN RX ID = **0x1806E8F4**

1. Charger CAN TX ID = **0x18FF50E5**
2. Charger CAN TX ID = **0x18FF50E7**
3. Charger CAN TX ID = **0x18FF50E8**

From charge VCU firmware version 20240202 up we support flexible J1939 Source ID's.
Still each charger needs a unique RX ID in the range **0x180600F4** to **0x1806FFF4**
and a corresponding TX ID in the range **0x18FF5000** to **0x18FF50FF**.

You should enable "Distribute current to connected chargers".

Emus G1 Firmware version: 2.9.2
Prior to EMUS G1 firmware version 2.9.2 EMUS does always divide the current by the number of chargers configured instead of the number of chargers detected.

We recommend using EmusBMS_m1284p_v2.9.2_3_967404_J1939.img
with control panel EMUS_CP_2.11.0_4_967402_

BatteryBox Infy charger/inverter note: In special cases where a 3-phase charger with single J1939 message is used. You must configure EMUS with 3 chargers as explained above and manually set all 3 charger RX and TX CAN IDs to the same ID! Like this the VCU knows it's a 3-phase setup, but EMUS never detects more than one charger and does not divide the current.
(This method is not recommended and does not work if the charger also supports single phase mode because the phase count is fixed to 3!)

1. Charger CAN RX ID = **0x1806E5F4**
2. Charger CAN RX ID = **0x1806E5F4**
3. Charger CAN RX ID = **0x1806E5F4**

1. Charger CAN TX ID = **0x18FF50E5**
2. Charger CAN TX ID = **0x18FF50E5**
3. Charger CAN TX ID = **0x18FF50E5**

If the pause charge function is used it is necessary to set the max charging state durations to infinite because we will just ask 0A from the charger.

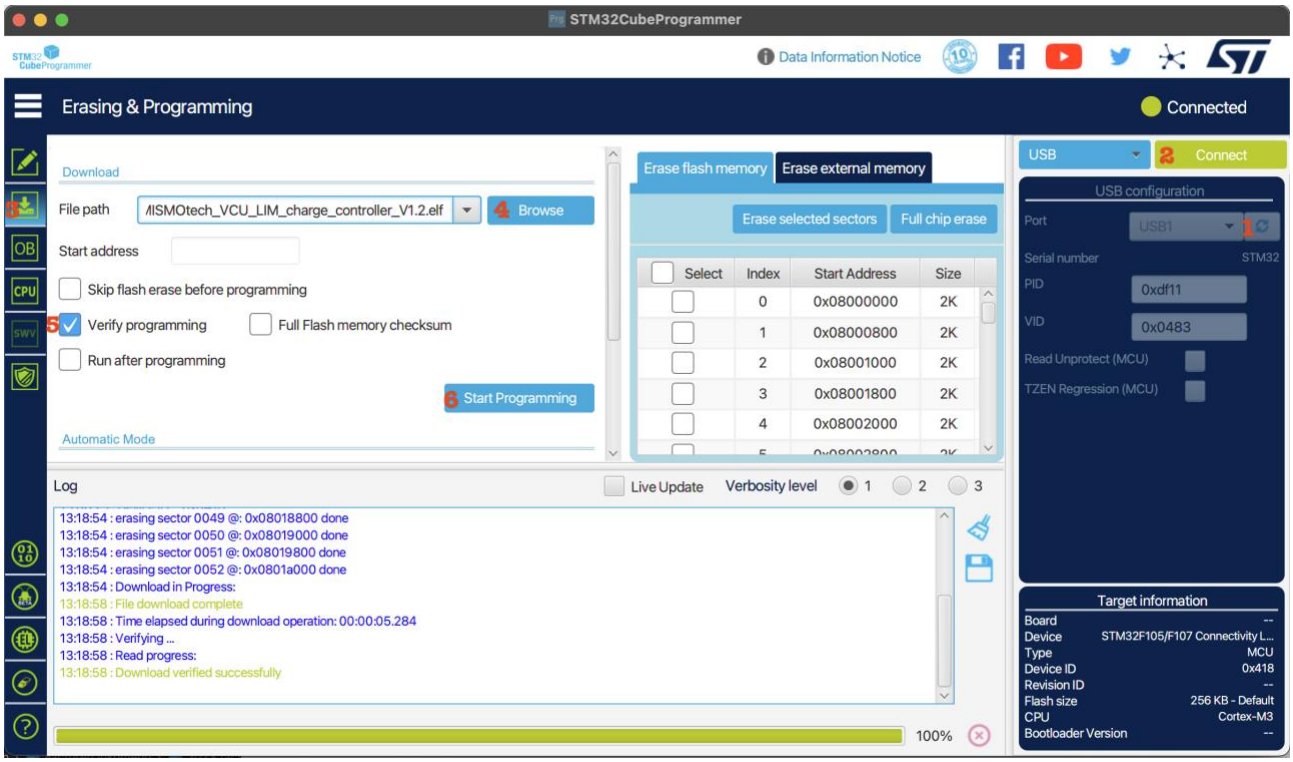
Max Main Charge Stage Duration = 0 (infinite)

Max Balancing Stage Duration = 0 (infinite)

9 Charge controller firmware update process

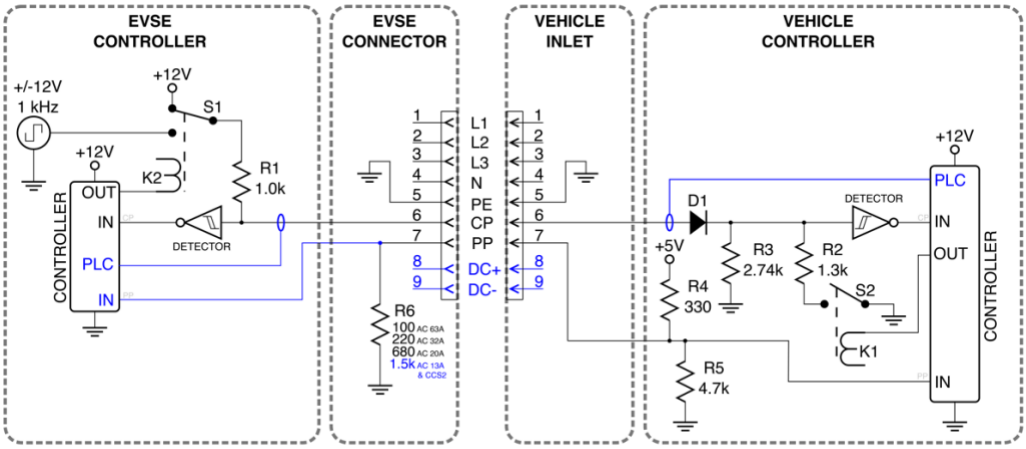
The VCU software can be updated with a USB-C cable. The controller also gets powered by the USB if no other power source is connected.

1. Download and install STM32CubeProgrammer for your operating system.
   https://www.st.com/en/development-tools/stm32cubeprog.html
2. Disconnect the 48-pin connector or remove the power supply.
3. Connect a USB-C cable from your computer to the controller.
   The on-board status LED should get blue if USB power is detected.
4. Select USB as the communication method in STM32CubeProgrammer.
   It sometimes takes a few seconds before the device is recognized in DFU mode, just press the refresh button a couple of times until you see USB1 device.
   >> Try a different USB cable if it does not work.
5. Press “Connect” to establish a connection to the controller.
6. Move to the Erasing & Programming tab on the left and press “Browse” to select the *.elf image you got with the new software.
7. Select “Verify programming”.
8. Make sure “Run after programming” is **not** selected.
9. Press “Start Programming” and wait until the process finishes.
   Do not disconnect the device during the download process!
10. You should get a few pop-ups with the “File download complete” and “Download Verified successfully” messages and your controller should be updated at this point, you can press “Disconnect”, remove the USB-C cable and reconnect the power supply / 12-pin connector.



10 Charge protocol description

This is a very simplified overview how the used charge protocols for AC and CCS charging work.



CCS2 / IEC61851 signaling circuit

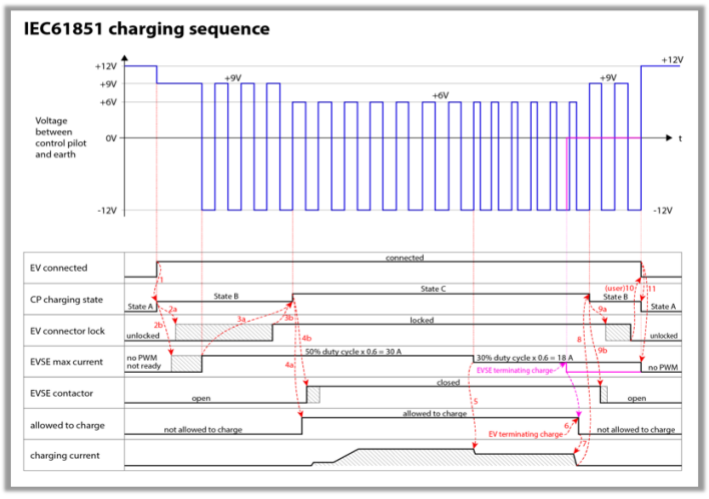
10.1 AC slow charge protocol [IEC61851]

A low-level protocol which allows the EVSE (charging station) and the EV to communicate.

By default, the EVSE outputs +12V on the CP pin, and when connected to an EV the signal will be reduced to 9V by a load resistor present in the EV. This signals the EVSE that the connector has been plugged into an EV. The EVSE then send a 1khz +12V to -12V square wave (PWM signal) and the duty cycle value corresponding to the maximum current it can deliver.
If the EV has locked the connector and is ready, it performs a handshake by changing the load resistance and dropping the positive PWM voltage to 6V, after which the EVSE opens the AC relays and charging can begin.

In Europe where untethered charging stations are allowed, the PP pin is used to detect the maximum power rating of the cable.

Charging status		Resistance, CP-PE	Voltage, CP-PE	Frequency
Status A	Standby	Open, or ∞ Ω	+12 V	DC
Status B	Vehicle detected	2740 Ω	+9 V | -12 V	1kHz PWM
Status C	Ready (charging)	882 Ω	+6 V | -12 V	1kHz PWM
	Charging paused from EVSE	882 Ω	+6 V	DC
Status D	Charging with ventilation (never used)	246 Ω	+3 V | -12 V	1kHz PWM
	Charging paused from EVSE	246 Ω	+3 V	DC
Status E	No power (EVSE shut off)	-	0 V	-
Status F	Error		-12 V	-



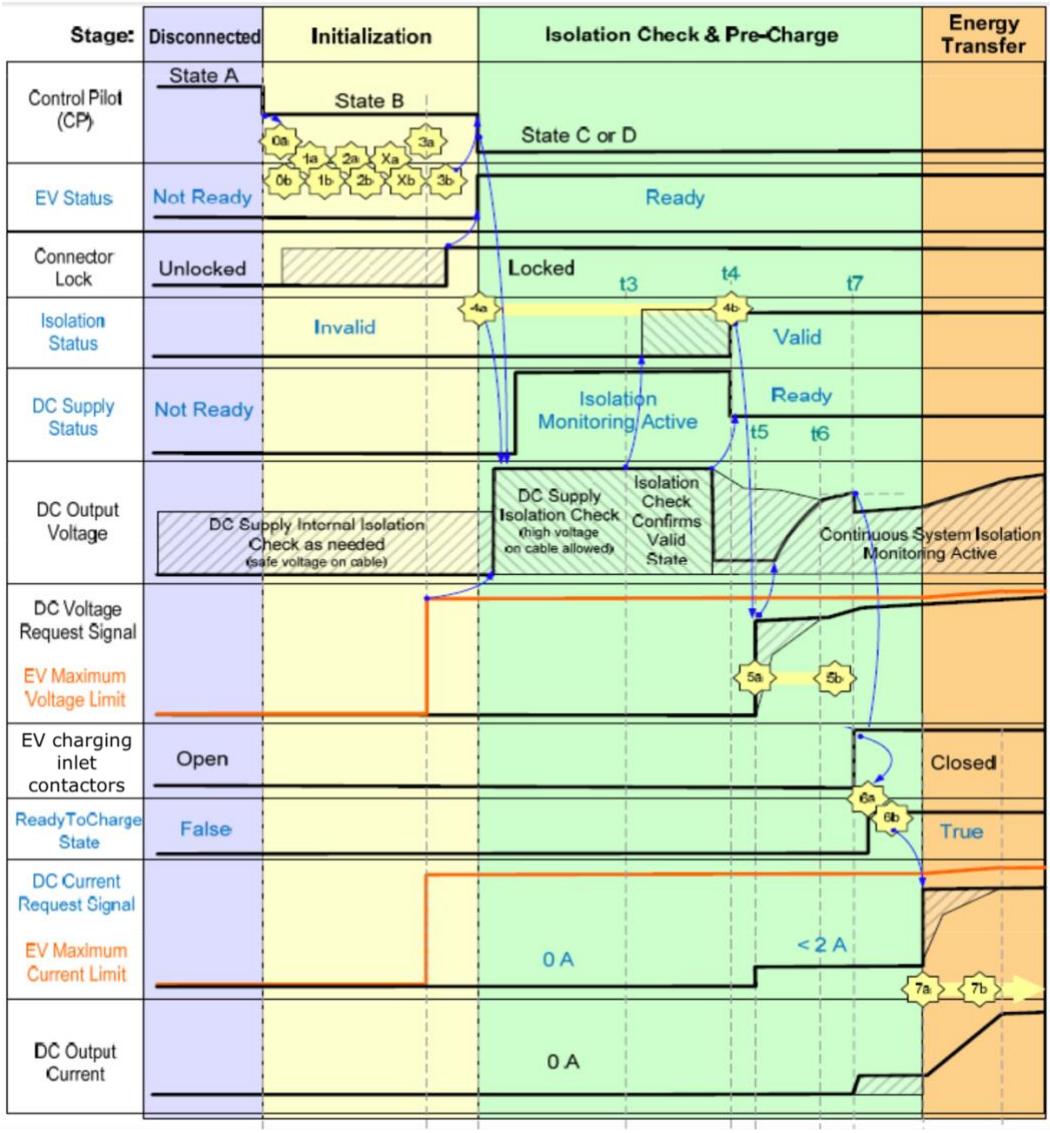
10.2 CCS DC fast charge protocol [ISO15118 / DIN1712]

ISO 15118 and DIN 70121 are quite complex high-level protocols which require a network connection over PLC (power line communication) on the CP pin.

Because the AC to DC converter (the charger) is in the charging station there are much more communication and safety steps required.

The newer ISO 15118-20 protocol supports more advanced features like bidirectional V2G charging and plug and charge which automatically handles the identification and payment process.

Illustration of charging sequence with a simplified architecture



You undertake your project at your own risk.

You should satisfy yourself that you are safely install and troubleshoot electrical systems and satisfy any legal requirements you may be subject to before beginning the project.

The information provided in this document and the support provided is intended as information only. MISMO·tech takes no responsibility for how you use the information, nor any liability for injuries, or death, that may result from your actions.

11 User manual revision history

Revised on	Version	Description	Approved by
23-08-2025	0.1	Initial document creation	ME
12-11-2025	1.0	Publish document	ME
18-02-2026	1.1	general information on first page recommended contactor information added Isabellenhuette pinout added hardware version information standard NO lock actuator instead of NC	ME