VECTOR

User Manual
vSECC Controllers
Supply Equipment Communication Controllers

Version 3.7

Contents

1 Introduction

In this chapter you will find the following information:

1.1 About This User Manual

1.1.1 How to find information quickly

This user manual provides you with the following access help:

- At the beginning of each chapter, you will find a summary of its contents
- The header indicates the current chapter of the manual
- The footer shows the manual's version
- At the end of the manual, you will find a glossary to look up used technical terms and abbreviations

1.1.2 Conventions

The two tables below show the notation and icon conventions used throughout this manual.

Style	Utilization
bold	Fields/blocks, user/surface interface elements, window- and dialog names of the software **[OK]** Buttons in square brackets **File|Save** Notation for menus and menu commands
`Source Code`	File and directory names, source code, class and object names, object attributes and values
`Hyperlink`	Hyperlinks and references
Emphasis	Terms with special emphasis

Symbol	Utilization
	This icon indicates notes and tips that facilitate your work.
	This icon warns of dangers that could lead to damage.
	This icon indicates step-by-step instructions.
	This icon indicates an introduction to a specific topic.

1.1.3 Certification

Vector Informatik GmbH is certified under ISO 9001. The ISO standard is a globally recognized standard. Details can be found at the Vector website.

1.1.4 Warranty

We reserve the right to modify the contents of the documentation or the software without notice. Vector disclaims all liabilities for the completeness or correctness of the contents and for damages which may result from the use of this documentation.

1.1.5 Service, Support and Disposal

You can issue a support or hardware repair request online at vector.com/support or in our Vector Customer Portal at portal.vector.com.
You can get through to our Support hotline by calling +49 (0)711 80670-200.

Please also revise the Vector KnowledgeBase for Frequently Asked Questions.

Furthermore, E-Learnings about the vSECC Controllers are available.
At https://elearning.vector.com you will find all the courses relevant for commissioning and integration. Access can be requested using the invoice number.

If you want to return the device, please remove all things that were not part of the original delivery, e.g. SD cards, and send it back to:

> Vector Informatik GmbH
> Dept. CPL4
> Motorstr. 56
> 70499 Stuttgart
> Germany

Observe the national regulations and laws for the disposal of the device. Ask your supplier if you are not sure how to dispose the device. Within the European Community, the Directive on Waste Electrical and Electronic Equipment (WEEE Directive) and the Directive on the Restriction of the Use of Certain Hazardous Substances in Electrical and Electronic Equipment (RoHS Directive) apply.

1.1.6 Trademarks

All brand names in this documentation are either registered or non-registered trademarks of their respective owners.

1.2 Important Notes

1.2.1 Safety Instructions and Hazard Warnings

Caution: In order to avoid personal injuries and damage to property, you must read and understand the following safety instructions and hazard warnings prior to installation and use of the product. Provide this documentation (manual) to every user of the product.

1.2.2 Proper Use and Intended Purpose

The Supply Equipment Communication Controllers are used for communication between the vehicle's charge controller and the supply equipment via the charging cable and CCS or CHAdeMO (only vSECC) plug connections provided for this purpose. Based on the received and transmitted information, messages for interaction with the supply equipment operator and for controlling the power electronics are exchanged with other components of the supply equipment. The controller also sends messages to the back end of the charging station, the Charging Station Management System.

Caution: The product is designed for permanent, fixed installation in closed control cabinets and stationary charging equipment. The installation environment must be dry and protected from the weather.

Caution: Only specifically qualified, trained and authorized personnel is allowed to install, set up, configure and operate the product to prevent accidents from hazardous electrical voltage or electrical power. Access to operating products must be limited to authorized personnel at any time. The housing of the vSECC must always be assembled during operation. The device may only be used with appropriate connectors. The connectors of the vSECC Controllers may only be used and operated within the specified range, the information in the manual must be observed.

The product can be integrated into an existing IT infrastructure. The configuration of the respective parameters and IT security is the responsibility of the customer.

Caution: vSECC Controllers contain components and circuits that communicate with other components and circuits that can store and transform energy. The user has to take care of the resulting dangers and make a separate risk assessment. The devices may only be operated within the specified temperature range.

Caution: Electrical safety and data security of the Supply Equipment must be assured by separate means and is not in scope of the product. In particular, effective measures must be taken to avoid damage and injury caused by overload or short circuit in the electric power installation independent from the vSECC Controllers.

Caution: Neither the monitoring of residual current and insulation, relay monitoring (main conductor), especially sticking of the conductors; nor the cooling function (use of the temperature sensors for monitoring), monitoring of battery and wire and the performance limits in the vehicle; nor the monitoring of power electronics incl. contactors (especially emergency shutdown devices) is in the scope of the product and must therefore be assured by separate means.

1.2.3 Foreseeable Misuse

Caution: vSECC Controllers do not comply with the directive 2014/34/EU and must therefore not be used in explosion critical areas. Operation or installation in mobile equipment without adequate protection against weather and moisture is not allowed. The electrical safety of the supply equipment is not in the scope of the vSECC Controllers' functionality and must be assured independently by suitable measures such as insulation monitoring, residual current detection, overload protection and circuit breaker. It is not permitted to use the device for purposes other than controlling the charging communication. Interventions or changes to the hardware are not permitted. The vSECC Controllers may only be installed and operated by qualified and instructed personnel, who is familiar with the contents of this document and must have access to it at all times.

1.2.4 Hazards

Caution: Supply Equipment operates under high voltage which could also occur at the product in case of failure and cause heavy injury and damage. Wrong configuration and/or operation of the product may cause failures of the Supply Equipment leading to personal injury or damage to property.
Comply with safety standards and public regulations which are relevant for the operation of the system. Before you can operate the system in public areas, it should be tested on a site which is not accessible to the public and specifically prepared for performing tests in order to reduce hazards.

1.2.5 Disclaimer

Caution: Claims based on defects and liability claims against Vector are excluded to the extent damages or errors are caused by improper use of the controller or use not according to its intended purpose. The same applies to damages or errors arising from incorrect mounting, insufficient training or lack of experience of personnel using the controller.

It is not allowed to open the housing of the vSECC Controllers. Claims will not be accepted after the housing was opened.

1.2.6 Open-Source Licenses

vSECC Software includes several open source software tools. This open source software is governed by the terms and conditions of the applicable open source license. You are bound to the terms and conditions of the applicable open source license in connection with your use and distribution of the open source software in this product.

A complete list of open source software modules and their respective licenses can be found in the provided `ThirdPartyLicenses.html` file.

Upon request, we will provide the applicable GPL/LGPL source code files via the Vector Portal for a nominal cost to cover provisioning as allowed under the GPL. This offer is valid for 3 years.

1.3 vSECC Controllers at a Glance

The vSECC Family consists of various Supply Equipment Communication Controllers (SECC) and is designed to be used in smart charging applications.

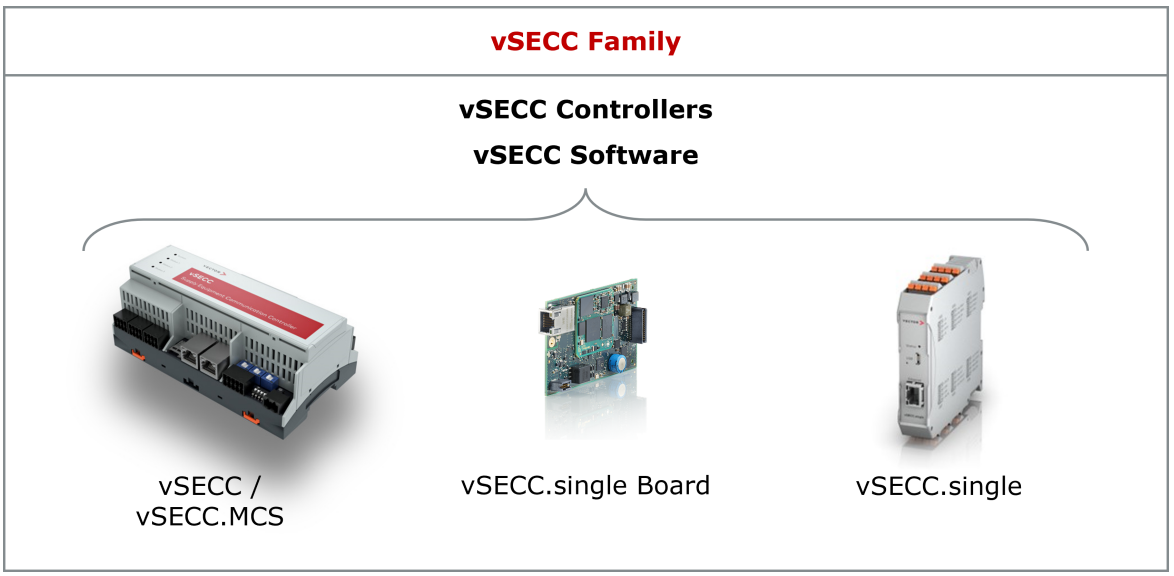


Figure 1: The vSECC Family

The vSECC Controllers designate the hardware products, which all share the same vSECC Software. Different software options are available to choose from. Today, the vSECC Software is capable of DC (bidirectional) and MCS charging; and PWM-based AC charging. The **vSECC Software** is responsible for the communication between an Electric Vehicle (EV), a Charging Station Management System (CSMS), the Power Electronics (PE) and peripherals.

The **vSECC** is designed for handling up to two CCS (Type 1 or 2) / NACS DC or AC (Type 1 or 2) charge points in parallel, or one CCS / NACS / AC and one CHAdeMO charge point. Moreover, the vSECC can control charging stations for roof-mounted or inverted pantograph charging. All the future options (e.g. GB/T) will be available via a pure software update.

The **vSECC.MCS** is designed for handling one MCS and one CCS (Type 1 or 2) / NACS DC charge points in parallel.

The **vSECC.single Board** manages one CCS DC / NACS or AC charge point and is designed for an highly integrated solution, e.g. for wallboxes, mobile chargers and small charge point outlets. The connectors of the vSECC.single Board's interfaces are to be placed on a so-called base board (as shown in Figure 5), which can be developed by our customers (for the integrated solution).

The **vSECC.single** handles the charging communication for one CCS DC / NACS or AC charge point. It includes the vSECC.single Board and the so-called base board, all placed into a handy housing for mounting on a DIN rail.

The large number of practical interfaces make the vSECC Controllers widely applicable for the rapid implementation of intelligent charging stations and wallboxes.
For CCS / NACS and roof-mounted pantograph charging, the communication to the EV is established by Control Pilot (CP) basic signaling (IEC 61851 and SAE J1772) and Power Line Communication (PLC) according to DIN SPEC 70121 and ISO 15118-2, -3 and -20. For MCS charging, Automotive Ethernet Communication (10Base-T1S, as specified in ISO 15118-10) with ISO 15118-20 is used. Bi-directional power transfer as defined in the ISO 15118-20 is also possible with the vSECC Controllers. Moreover, the **vSECC** realizes CHAdeMO charging via CAN or inverted pantograph charging according to OppCharge or SAE J3105 via Ethernet and a Wireless Access Point (WAP).
For the communication to the back end, e.g. for load management, the vSECC Controllers require an Open Charge Point Protocol (OCPP) 1.6J or 2.0.1 compliant CSMS, such as Vector's vCharM.
The vSECC Controllers provide the possibilities for identification at the charging station with External Identification Means (EIM), Autocharge and Plug & Charge (PnC; only for DC) as standardized in ISO 15118-2.

For some functionalities, e.g. the usage of BPT or PnC, a license is required. Please get in touch with your sales contact for details.

The vSECC Controllers provide a variety of communication interfaces, which can also be used to control the Power Electronics.
Vector has specified the Power Electronics Protocols over Ethernet WebSocket or CAN (PEP-WS and PEP-CAN) , which will be delivered along with the controller. Moreover, custom protocols can be implemented on the vSECC Controller.
The "Configurable Customer Interface" enables the custom use of the vSECC Controller's hardware interfaces. Peripheral devices such as RFID readers, energy meters, insulation monitors etc. can be flexibly connected using the browser-based low-code programming tool Node-RED in a vSECC Software Container. This offers maximum flexibility in the selection of peripheral devices with ease of use.
Furthermore, it is possible to program, install and execute own functionality in a container on the vSECC Controllers, as part of the "vSE Developer Program".

The "vSE Developer Program" is currently in a prototypical stage and only for pilot customers.

1.3.1 vSECC Interfaces Overview

A top-level connection scheme of the vSECC is shown in Figure 2.

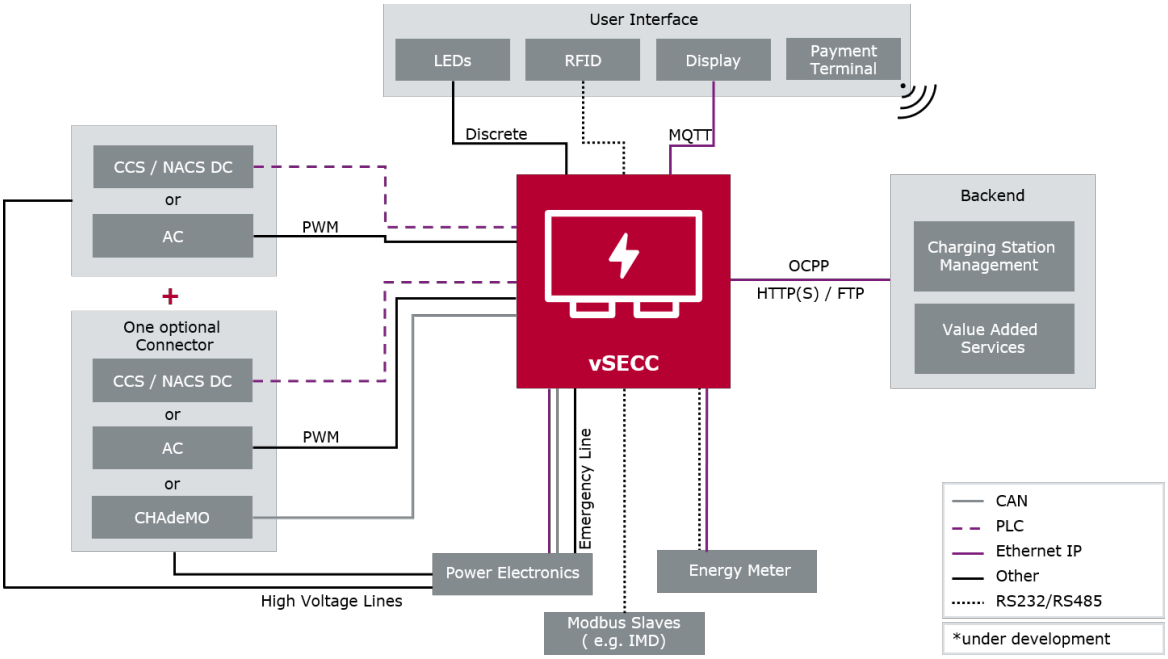


Figure 2: vSECC connection scheme

The hardware overview shown in Figure 6 in chapter 2.2 includes all connectors available with the fully populated vSECC. With the current version of the vSECC, the following connectors can be used:

- X300: CHAdeMO Charging Connector
- X301: Analog Inputs (e.g. Temperature Sensors)
- X302: CCS Charging Connector 2
- X303: CCS Charging Connector 1
- X304: Safety Outputs
- X305: Serial Communication (2x CAN, RS232, RS485)
- X306: Digital In-/Outputs, Start (CHAdeMO) and Stop Buttons, Pantograph Control
- X307: Power Supply Connector
- ETH1: RJ45 Ethernet Connector 1
- ETH2: RJ45 Ethernet Connector 2

The connectors are described more in detail in chapter 2.2.

1.3.2 vSECC.MCS Interfaces Overview

A top-level connection scheme of the vSECC.MCS is shown in Figure 3.

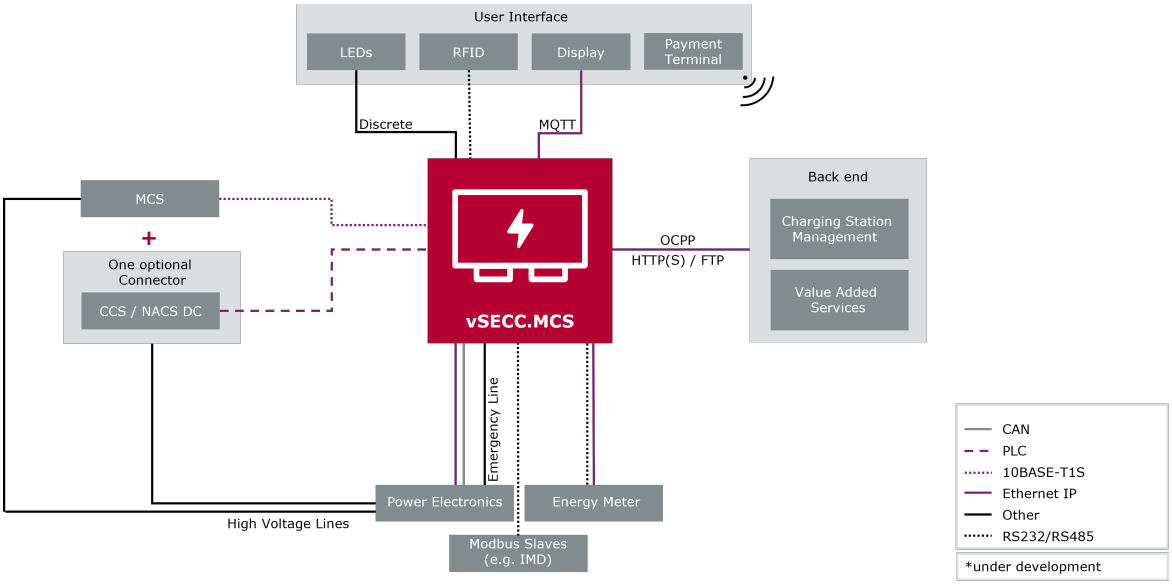


Figure 3: vSECC.MCS connection scheme

The hardware overview shown in Figure 18 in chapter 3.2 includes all connectors available with the fully populated vSECC.MCS. With the current version of the vSECC.MCS, the following connectors can be used:

- X200: MCS Charging Connector, Safety Output
- X301: Analog Inputs (e.g. Temperature Sensors)
- X302: CCS Charging Connector
- X304: Safety Output for CCS
- X305: Serial Communication (CAN, RS232, RS485)
- X306: Digital In-/Outputs, Stop Buttons
- X307: Power Supply Connector
- ETH1: RJ45 Ethernet Connector 1
- ETH2: RJ45 Ethernet Connector 2

The connectors are described more in detail in chapter 3.2.

1.3.3 vSECC.single Board Interfaces Overview

A top-level connection scheme of the vSECC.single Board is shown in Figure 4.

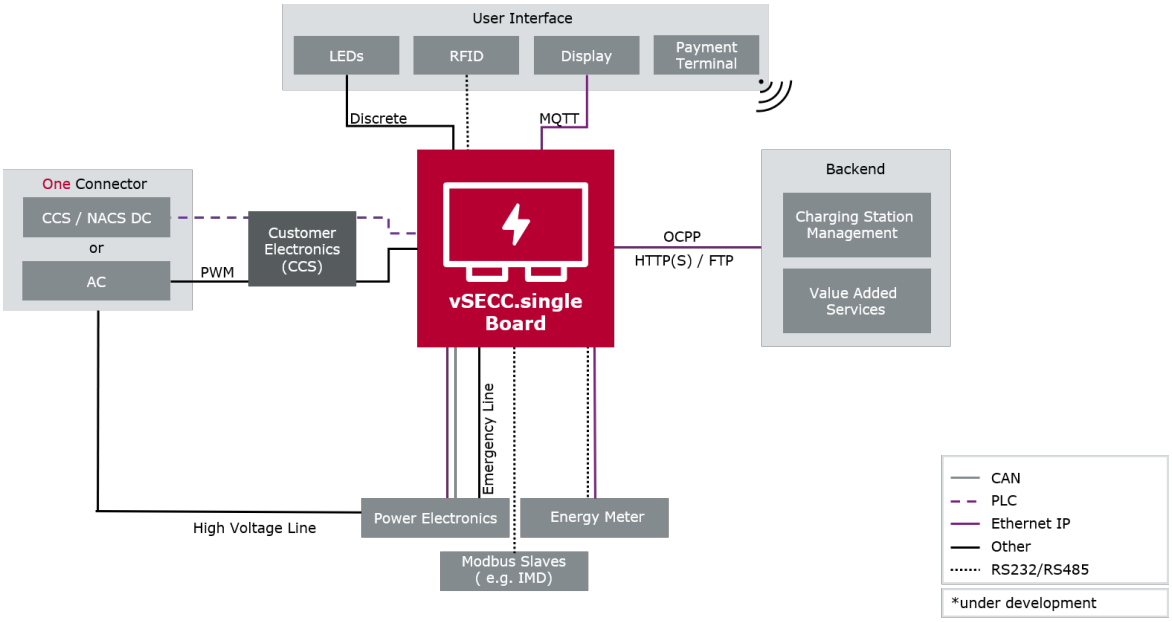


Figure 4: vSECC.single Board connection scheme

The hardware overview shown in Figure 28 in chapter 4.2 shows the connectors, including those that connect the vSECC.single Board to a so-called base board. With the current version of the vSECC.single Board, the following functionality is given for the connector and the respective pins:

- Connector X300:
  - Power Supply
  - Digital In-/Outputs
  - Analog Inputs
  - Serial Communication (1x CAN, RS232, RS485)
  - Safety Output
- Connector X301:
  - CCS Charging Communication
  - Temperature Sensors
- RJ45 Ethernet Connector

The connectors are described more in detail in chapter 4.2.

1.3.4 vSECC.single Interfaces Overview

A top-level connection scheme of the vSECC.single is shown in Figure 5.

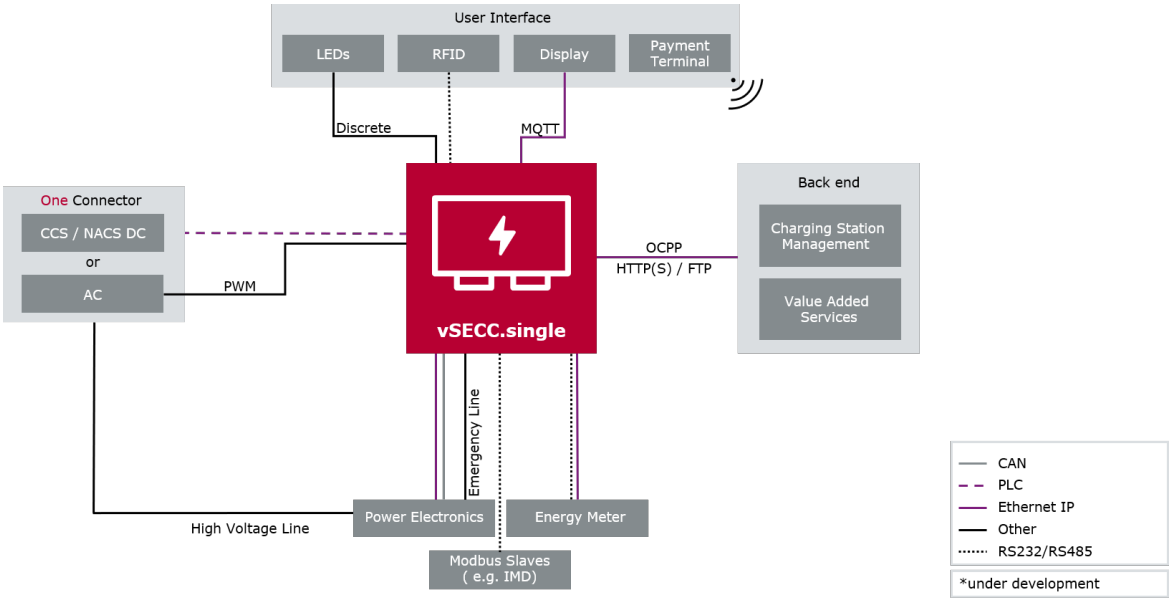


Figure 5: vSECC.single connection scheme

The hardware overview shown in Figure 32 in chapter 5.2 includes all connectors available with the fully populated vSECC.single.
With the current version of the vSECC.single, the following connectors can be used:

- X301: Digital/Analog Inputs
- X302: CCS Charging Communication
- X303: Safety Output
- X304: Analog Temperature Inputs
- X306: Power Supply, Serial Communication (RS232)
- X307: Serial Communication (1x CAN, RS485)
- ETH: RJ45 Ethernet Connector

The connectors are described more in detail in chapter 5.2.

1.4 vSECC Software Features

1.4.1 CCS Type 2 / NACS DC charging

- IEC 61851 Control Pilot for basic communication
- DIN SPEC 70121 High Level Communication
- ISO 15118-2/-3 High Level Communication (DC only)
- Authentication via External Identification Means (EIM)
- Load leveling based on power electronics limits and CSMS charging schedules (received via OCPP interface)
- Renegotiation process according to ISO 15118-2
- Charge pause according to ISO 15118-2 and ISO 15118-3:
  - EVSE wakeup by EV CP state BCB toggle
  - EV wakeup by EVSE CP state by PWM signal change 100% → 5% and, if necessary BEB toggle
  - EV wakeup can also be triggered via the MQTT interface
- IEC Control Pilot signal and basic signalling (PWM) is only used for High Level Communication
- Configurable EnergyTransferModes according to ISO 15118-2:
  DC_EXTENDED / DC_COMBO_CORE / DC_CORE / DC_UNIQUE

1.4.2 CCS Type 1 DC Charging

- SAE J1772 Control Pilot for basic communication
- Same high level communication features as in CCS Type 2 DC charging
- SAE Control Pilot signal and basic signalling (PWM) is only used for High Level Communication
- SAE Proximity Pin (PP) handling for controlled emergency shutdown
- Diagnostics of DIP-switch settings, if PP supervision is active

1.4.3 MCS Charging (vSECC.MCS only)

- Enable MCS charging with vSECC.MCS
- Use of 10Base-T1S for basic communication as specified in ISO 15118-10
- Support of ISO 15118-20 High Level Communication with Amendment for MCS
- Removal of SLAC
- Usage of new communication pins Charge Enable & Insertion Detection as specified in IEC 61851-23-3
- Encryption with TLS 1.3

1.4.4 Prototypical PWM-based AC charging

- Charging communication (Basic Signalling) according to IEC 61851 and SAE J1772
- 1- or 3-phase Mode 3 Case C charging (with fixed cable)
- No support of High Level Communication, authorization mechanisms, OCPP charging profiles and native energy meter support
- Contactors control and feedback via digital I/Os
- Maximum charging current configurable via Web-UI

1.4.5 Prototypical ISO 15118-20 charging

- Charging communication according to ISO 15118-20 (DC only)
- Support of Dynamic Control Mode for uni- and bidirectional charging
- Support for Bidirectional Power Transfer (BPT)
  - Setting the dynamic setpoint for (dis-)charging via OCPP device model variable and via Web Interface
  - The EV's min/max (dis)charging limits are published on MQTT for managing the discharge
- Support of Scheduled Control Mode for unidirectional charging
  - Support of Charging Schedules, Schedule Renegotiation and Sidestream Communication (used for ScheduleExchange messages)

1.4.6 Prototypical CHAdeMO 0.9.1 and v1.2 charging (Option, only for vSECC)

- Charging communication according to CHAdeMO 0.9.1 and v1.2 protocol specification
- Handling of CHAdeMO Start and Stop Buttons

1.4.7 Prototypical Inverted Pantograph charging (Option, only for vSECC)

- Charging communication according to OppCharge v1.3.0 protocol specification based on ISO/IEC DIS 15118-2
- **C**harging communication according to SAE J3105 (Prototype)
- Pantograph Control via digital I/Os or PEP-WS
- **R**FID Pairing of buses supported

1.4.8 Value Added Services (Internet)

- The vSECC software can be configured to offer internet access via ISO 15118 Value Added Services (VAS)
- Supports the service announcement internet access for TCP port 80 and 443
- The vSECC Controller supports the vehicle to set a global dynamic IPv6 address via Neighbor Discovery for IP version 6 (RFC 4861)
- The vSECC Controller supports VAS back ends with a fixed IPv6 address
- The IP Routing tables are shown in the Web-UI

1.4.9 Transport Layer Security

- Support of TLS 1.2 for ISO 15118-2 High-Level Communication Encryption
- Support of TLS 1.3 for ISO 15118-20 High-Level Communication Encryption
- Certificate Management in Web Interface
- TLS Debugging Mode for Development

1.4.10 Charging Schedules

- Charging Schedules are configurable as static PowerMaxLimit or provided by CSMS
- Default Charging Profiles provided by CSMS are persisted as required by OCPP
- Duration of Default Charging Profile is configurable
- The Composite Schedule is published on MQTT, to understand what is currently limiting the charging

1.4.11 Secure Operating System

- Secure boot mechanisms to run only signed and verified software on the device
- Usage of hardware related security mechanisms to recognize modified software
- Linux based operating system

1.4.12 Symmetric Key and Certifcate Handling

- Creation of certificate signing requests (CSR)
- V2G EVSE leaf certificate installation (including private key creation) through CSR using OCPP
- CSMS Root certificate installation and deletion using OCPP
- Installing client certificates sent by the CSMS
- Usage of the secure storage for private keys and certificates that are used for client certificate authentication

1.4.13 Hardware IEC CP/PP or CE/ID supervision

- Creation of certificate signing requests (CSR)
- Proximity Pin (PP) and Control Pilot (CP) (for CCS) / Charge Enable (CE) and Insertion Detection (ID) (for MCS) as dedicated hardware function to monitor and shutdown in emergency case
- Normally Open (relay based, potential-free) switching output

1.4.14 Stop Charging

- Graceful stop of charging by CSMS via OCPP
- Graceful stop of charging by power electronics via PEP
- Graceful stop of charging by physical stop buttons via digital inputs
- Graceful stop of charging via MQTT topic

1.4.15 Remote Start of Charging Sequence

- Allows starting a charging sequence remotely while the EV is plugged-in
- The remote start can be triggered by using the OCPP 1.6 message ”RemoteStartTransaction” or OCPP 2.0.1 ”RequestStartTransactionRequest” or via the MQTT interface
- A remote start causes a transition from CP state B to E and back, simulating unplugging and plugging in the EV

1.4.16 CSMS connectivity (OCPP)

- Supported CSMS protocols:
  - OCPP 2.0.1
  - OCPP 1.6J
- WebSocket based connection according to OCPP 2.0.1: Part 4 – JSON and OCPP-J 1.6 Specification over WebSocket is supported
- OCPP security profiles 1 and 2 are supported (OCPP 2.0.1: Part 2 – Section 1.3 ”Security Profiles”)
- The ”Basic Implementation of OCPP 2.0” as defined in the OCPP 2.0 standard (OCPP 2.0.1: Part 0 – Introduction) is supported
- Monitoring of the WebSocket connection status via WebSocket ping to allow faster detection of connection losses to CSMS **NEW!**
- Fallback from TxProfile to TxDefaultProfile when CSMS switching offline **NEW!**

1.4.17 OCPP 2.0.1 Use Cases

- It is possible to update the charging station credentials (OCPP 2.0.1 use cases A01 – A02)
- It is possible to upgrade the charging station's security profile (OCPP 2.0.1 use case A05)
- It is possible to boot the charging station (OCPP 2.0.1 use cases B01 – B04)
- It is possible to configure the charging station via a CSMS (OCPP 2.0.1 use cases B05 – B08)
- It is possible to set a new network connection profile (OCPP 2.0.1 use cases B09 – B10)
- It is possible to reset the charging station (OCPP 2.0.1 use cases B11, B12)
- It is possible to authorize a driver using RFID, Plug & Charge, a start button and ISO 15118 External Identification Means (EIM). Offline authorization of an Unknown Token is possible (OCPP 2.0.1 use cases C01, C02, C07, C08, C15; *certificate-based authorization is not a feature of OCPP 1.6J specification*)
- It is possible to remotely authorize a driver through the CSMS, e.g. by using credit card information or a smartphone app (OCPP 2.0.1 use cases C03, C05, F01, F02)
- It is possible to start and stop transactions also while the charging station is offline and end the charging process (OCPP 2.0.1 use cases E01 – E09, E11 – E15)
- It is possible to remotely stop transactions and the charging (OCPP 2.0.1 use cases F03, F04)
- It is possible to remotely trigger messages (OCPP 2.0.1 use case F06)
- It is possible to change and report the availability of an EVSE and its connectors (OCPP 2.0.1 use cases G01 – G04). **NEW!** also on charging station level
- It is possible to use reservations (OCPP 2.0.1 use cases H01 – H04) **NEW!**
- It is possible to provide tariff and cost information to the customer (OCPP 2.0.1 use cases I02 - I04, P01)
- It is possible to send transaction related meter values (OCPP 2.0.1 use case J02)
- It is possible to perform General Smart Charging and Renegotiation (OCPP 2.0.1 use cases K01 – K02, K05 – K07, K10, K16, K17)
- It is possible to perform firmware updates (OCPP 2.0.1 use case L02) via HTTP/HTTPS and FTP protocol
- It is possible to delete certificates from a charging station and to install CA certificates (OCPP 2.0.1 use cases M04, M05)
- It is possible to upload log files (OCPP 2.0.1 use case N01) via HTTP/HTTPS and FTP protocol
- It is possible to set and clear monitors and monitoring events (OCPP 2.0.1 use cases N04, N06 – N08; *not a feature of OCPP 1.6J specification*)
- It is possible to set a display message (OCPP 2.0.1 use case O01 – O02)

- **NEW!** It is possible to reserve charging station/connectors (OCPP 2.0.1 use cases H01 – H04)
- Any non-supported messages are rejected

1.4.18 OCPP 1.6J Messages

The following messages of the OCPP 1.6J specification are supported:

- ”Core” Profile
  - Authorize
  - BootNotification (Meter S/N & Type **NEW!**)
  - ChangeAvailability
  - ChangeConfiguration
  - DataTransfer (charging cost only, or external handling via MQTT **NEW!**)
  - GetConfiguration
  - Heartbeat
  - MeterValues (added more measurands **NEW!**)
  - RemoteStartTransaction
  - RemoteStopTransaction
  - Reset (Differentiation hard/soft reset **NEW!**)
  - StartTransaction
  - StatusNotification (added failure reporting on charging station level **NEW!**)
  - StopTransaction (added more measurands **NEW!**)
- ”Firmware Management” Profile
  - GetDiagnostics
  - DiagnosticsStatusNotification
  - FirmwareStatusNotification
  - UpdateFirmware
- ”Smart Charging” Profile
  - ClearChargingProfile
  - SetChargingProfile
- ”Remote Trigger” Profile
  - TriggerMessage
- ”Security” Profile
  - CertificateSigned
  - DeleteCertificate
  - ExtendedTriggerMessage
  - GetLog
  - InstallCertificate
  - LogStatusNotification
  - SignCertificate

1.4.19 Enhancing OCPP 1.6J Status Notification with Vendor-specific Information

- Customized information can be reported to the CSMS via the OCPP 1.6J StatusNotification Message
- The fields “errorCode”, “info” and “vedorErrorCode” can be filled via MQTT
- Possibility to send Status Notifications for the entire charging station and each charge point
- **NEW!** Possibility to report a failure on charging station level via MQTT

1.4.20 NEW! Enhancing OCPP Measurands with Vendor-specific Information

- Customized measurands can be reported to the CSMS via the OCPP 1.6J MeterValues/StopTransaction Message
- Customized measurands can be reported to the CSMS via the OCPP 2.0.1 TransactionEvent Message
- Customized measurands can be provided via MQTT
- Possibility to configure the measurands via WebUI

1.4.21 AutoCharge for identification of vehicles

- Pre-stage to Plug & Charge (ISO 15118) with simple EVCC-ID identification
- Vehicles can be identified by their EVCC-ID (MAC address of the EVCC) at the CSMS
- Vehicles can be authorized for charging by sending their EVCC-ID to the CSMS
- Authorization can be turned on/off in the vSECC Controllers’ configuration file
- If a vehicle is not authorized to charge, no charging transaction is started
- If no connection to a CSMS is established, the vehicle is assumed to be unauthorized and no charging will take place

1.4.22 Authorization

- RFID readers, an external MQTT client or Remote Start via CSMS are possible methods for authorizing a charging session
- Hardware support of the following RFID readers via RS232:
  - MCRN2-RS232 RFID reader from Minova
  - TWN4 MultiTech (2) Series from Elatec
  - One RFID reader for all charging connectors is supported
  - Interface to custom RFID Readers can be developed in Configurable Customer Interface
  - Flexible choice of various authorization sequences (first hold RFID card, first plug-in vehicle, etc.)
- MQTT interface for external authorization with custom hardware
  - Supports all types of OCPP 2.0.1 tokens

  - Token can be preauthorized or passed to the CSMS for authorization
  - Enables an individual and flexible authorization sequence
- For Remote Authorization, ConnectorID can be assigned after successful authorization
- Transactions can be stopped by presenting the same RFID token again
- Transactions get stopped automatically if the authorized token expires or is blocked by the CSMS during the charging process

1.4.23 Prototypical Authorization with Plug & Charge

- Support of Plug & Charge acc. to ISO 15118-2
- Authorization of vehicles by Contract Certificate
- Contract Certificate validation by CSMS
- EVSE Leaf Certificate installation possible via OCPP and Web Interface

1.4.24 Usage of Payment Terminals

- Support of cloud-based payment terminals with no direct interface to vSECC Controller
- Direct interface to Payment Terminals can be developed in Configurable Customer Interface
- Tariffs and the cost for charging are provided via OCPP and published on MQTT broker for display
- Compliant with German Eichrecht
- Possibility to inform the driver with an everyday language message and provide a machine-readable tariff to the energy meter at the same time

1.4.25 MQTT Interface

- MQTT broker with subscribe and publish access via ethernet interfaces
- Basic charging information is published on MQTT broker
- Digital/analog/temperature input states are published on MQTT broker
- Message to be displayed on HMI (OCPP Display Message) can be sent from the CSMS
- Outdated information on MQTT topics get reset with an empty string **NEW!**
- Ability to Report an OCPP Connector, EVSE or CS as Unavailable to the CSMS **NEW!**
- Topics to expose message contents for OCPP 1.6 DataTransfer & StatusNotification **NEW!**

1.4.26 TCP-based Power electronics control

- Connection to power electronic via physical Ethernet
- Interface to Power Electronics can be developed in Configurable Customer Interface
- Alternatively, usage of Vector's protocol specification PEP-WS 1.8
- Communication based on WebSocket connection with JSON data exchange
- Monitoring of the PEP-WebSocket connection status via WebSocket ping to allow faster detection of connection losses to PE **NEW!**

1.4.27 CAN-based Power electronics control

- Connection to up to 2 power electronics via physical CAN (one CAN-bus)
- Interface to Power Electronics can be developed in Configurable Customer Interface
- Alternatively, communication based on Vector's protocol specification PEP-CAN 1.4
- CAN-IDs and baudrate are configurable

1.4.28 Dynamic Switching of Power Electronics modules

- Allows switching between multiple power electronics for the vSECC's two connectors
- Updated PEP specification to inform the power electronics about changes in charging profiles
- This is only possible using OCPP 2.0.1

1.4.29 Modbus Gateway

- Usage of Modbus RTU devices, connected to vSECC Controller's RS485 interface, via Modbus TCP
- Variable Modbus RTU settings on RS485
- Modbus TCP is accessible via both Ethernet interfaces (vSECC only)
- RS485 / Modbus interface can be developed in Configurable Customer Interface

1.4.30 Energy Meter Support

- Virtual energy meter: Voltage and current are taken from power electronics values to calculate power and energy
- Native support of LEM DCBM 400/600 energy meter
- Support of AST DC Meter via Configurable Customer Interface
- Interfacing with energy meters is possible via MQTT broker
- Development of energy meter interface (e.g. RS 485) in Configurable Customer Interface
- Forward of signed meter values (OCMF[1]) for LEM & MQTT energy meter to support German Eichrecht

[1] OCMF: Open Charge Metering Format

1.4.31 Configurable Customer Interface

- Enables the custom use of the vSECC Controller's hardware interfaces independent of vSECC Software
- Peripheral devices such as RFID readers, energy meters, insulation monitors etc. can be flexibly connected using the browser-based low-code programming tool Node-RED
- Offers flexibility in the selection of peripheral devices with ease of use. Such tool offers speed and accessibility but can also present limitations. Kindly, refer to the Application Note for vSECC CCI on the recommended practices using Node-RED from our Vector Customer Portal at portal.vector.com.
- HTML page for display can be developed on the vSECC Controller
- Functionality previously running on another controller can be developed and executed on vSECC Controller
- Templates and application examples are included and make it easy to get started
- Easy provisioning of the container with data export function and possibility to upload single configuration files

1.4.32 Web-based Device Management

- The vSECC Controllers can be easily and intuitively configured via a local web front end
- The vSECC Controller runs a HTTP server that allows accessing the web-based configuration via Ethernet interfaces and a normal web
- Configuration of the controller is also possible via Python tool or REST API
- The configuration can be downloaded and uploaded to other vSECC Controllers
- The local time, date and time zone can be set and synchronized with various sources
- Dashboard shows important information and status of the charging station for monitoring
- The network setting can be updated
- The password can be changed
- The device can be rebooted
- Firmware updates can be installed
- A software container can be installed and managed
- Licenses (to use features that require a license) can be installed
- The hardware interfaces' (e.g. network) settings can be updated
- Log files can be downloaded and cleared
- Certificates (e.g. for TLS encryption) can be installed
- The vSECC Controller allows to open a remote support interface to vector's support servers on customer request

1.4.33 Firmware Update

- Secure firmware updates of the vSECC Controller (accepts only signed updates) are possible via:
  - CSMS with OCPP 1.6 and OCPP 2.0.1 use case L02
  - Web Interface
- Supported download methods: HTTP/HTTPS, FTP
- Firmware of any device in the charging station can be updated through the vSECC Controller
  - vSECC Controller downloads and distributes the firmware images via REST API
  - Notification via MQTT about the availability of a new firmware file

1.4.34 Log File Management

- Log Files can be downloaded and cleared in the Web Interface
- Log File upload to CSMS with OCPP 1.6 and according to OCPP 2.0.1 use case N01
- Supported upload methods: HTTP/HTTPS, FTP
- vSECC Software notifies the CSMS of the current upload status

1.4.35 HLC Logging

- Logging of whole high level communication (HLC) between vSECC Controller and EV into trace files
- HLC logging is supported for all licensed connectors (PLC, MCS, OppCharge and CHAdeMO)
- Trace files can be interpreted e.g. by CANoe and Wireshark
- Activation and deactivation via web interface
- Logfiles are rotated after running for about 2 days (20 MB size)

1.5 Scope of Delivery

Each delivery consists of a certain number of controllers, as specified in the order, the Safety Instructions and a link to the User Manual.

1.6 International Certification

In the following, country-specific certificates and information is listed. The official documents can be found in Appendix A.

As the vSECC.single Board is no stand-alone controller, it is not certified by Vector.

1.6.1 CE

1.6.2 UKCA

1.6.3 FCC

FCC ID vSECC: 2AXYRVSECC

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.
This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:
(1) This device may not cause interference.
(2) This device must accept any interference, including interference that may cause undesired operation of the device.

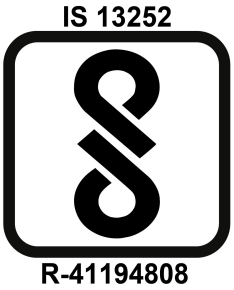
1.6.4 cCSAus (”UL certification”)

1.6.5 ICES

Vector’s self-declaration of compliance with ISED’s ICES standard applicable:

CAN ICES-003(B) / NMB-003(B)

1.6.6 BIS



Manufacturer:	Vector Informatik GmbH	
Model Name:	vSECC	vSECC.single
Model No.	20006	20011
Electrical Power:	Input 18-30V; <5A	Input 11-13V; <6A
Brand:	Vector	
Country of Origin:	Germany	

1.6.7 KC

Company Name:	Vector Informatik GmbH	
Product Name:	Supply Equipment Communication Controller (IMI61)	
Manufacturer:	Vector Informatik GmbH	
Country of Manufacture:	Germany	
Model Name:	vSECC	vSECC.single
Registration No.	R-R-VeC-vSECC	R-R-VeC-vSECCsingle
Date of Manufacture:	11-2020	03-2023

2 Installation Guide vSECC

In this chapter you will find the following information:

2.1 Physical Mounting

The vSECC is equipped with a mounting bracket which allows for an easy installation on a top-hat rail.

2.2 Electrical Connections

For wiring examples showing different use cases, please refer to Appendix F.

Figure 6 shows the vSECC from above. Each connector is described in detail below.

Figure 6: vSECC connector overview

We recommend purchasing the **vSECC Connector Kit**, please consult your Vector sales contact for details.
Alternatively, the following connector types can be directly obtained from **Phoenix Contact** for connecting to the vSECC:

vSECC Connector	Phoenix Contact Connector Name	Order Key
X301	DFMC 1,5 / 10-ST-3,5	1790182
X300, X302, X303	DFMC 1,5 / 4-ST-3,5	1790124
X304	DFMC 1,5 / 3-ST-3,5	1790111
X305	DFMC 1,5 / 5-ST-3,5	1790137
X306	DFMC 1,5 / 13-ST-3,5	1790218
X307	FKCN 2,5 / 2-ST-5,08	1754568

Caution: Please make sure that no pulling forces are applied on the wiring harness or connectors to make sure that no connector is getting pulled out.

2.2.1 X300 - CHAdeMO

1 CHD SEQ1	3 CHRG PER	5 LATCH OUT	7 GBT CC1
2 CHD SEQ2	4 PROX DET	6 LATCH IN	8 PE

Figure 7: vSECC connector: X300

This connector is used for the CHAdeMO charging interface. The pins have the following functional assignment and must be connected to the respective circuit of the charging cable.

- Pin 1, CHD SEQ1: CHAdeMO Charge Sequence Signal 1
- Pin 2, CHD SEQ2: CHAdeMO Charge Sequence Signal 2
- Pin 3, CHRG PER: Vehicle Charge Permission
- Pin 4, PROX DET: CHAdeMO Connector Proximity Detection
- Pin 5, LATCH OUT: Latch Control (output)
- Pin 6, LATCH IN: Latch Monitoring (input)
- Pin 8, PE: Protective Earth

Pin 7 is intended for GB/T or ChaoJi functionality and will be used for planned features coming with future software releases.

2.2.2 X301 - Analog In and Temperature Sensor Connectors

1 0-10V 2	3 AGND	5 AGND	7 TEMP 8	9 AGND	11 TEMP 6	13 AGND	15 TEMP 4	17 AGND	19 TEMP 2
2 0-10V 1	4 TEMP 9	6 AGND	8 TEMP 7	10 AGND	12 TEMP 5	14 AGND	16 TEMP 3	18 AGND	20 TEMP 1

Figure 8: vSECC connector: X301

This connector is used for both analog input signals and external temperature sensors. See section 8.18.7 and 8.18.8 for details and a mapping of PEP-identifiers to connector pins.

2.2.3 X302 - CCS Charging Connector 2

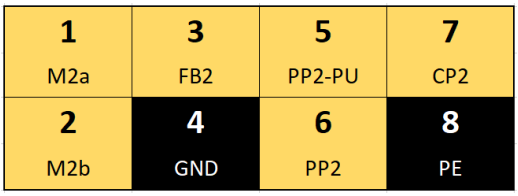


Figure 9: vSECC connector: X302

This connector is used for CCS Charging at charging port 2 which requires the following pins:

> Pin 6, PP2: Proximity Pin for SAE J1772 Proximity Detection (only used for CCS Type 1).
> Pin 7, CP2: Control Pilot line which corresponds to the respective pin of the second CCS connector.
> Pin 8, PE: Protective Earth for CCS connector 2.

The following pins may be used in the future. For now, they are ignored:

> Pin 1, M2a: Required for AC-charging.
> Pin 2, M2b: Required for AC-charging.
> Pin 3, FB2: Required for AC-charging.
> Pin 5, PP2-PU: Not used.

Please be aware of the naming: The connector X302 which has the lower number corresponds to the logical CCS connector 2.

Vector recommends protecting the power line communication from interference.

2.2.4 X303 - CCS Charging Connector 1

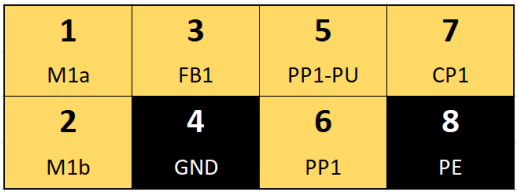


Figure 10: vSECC connector: X303

This connector is used for CCS Charging at charging port 1 which requires the following pins:

> Pin 6, PP1: Proximity Pin for SAE J1772 Proximity Detection (only used for CCS Type 1).

> Pin 7, CP1: Control Pilot line which corresponds to the respective pin of the first CCS connector.

> Pin 8, PE: Protective Earth for CCS connector 1.

The following pins may be used in the future. For now, they are ignored:

> Pin 1, M1a: Required for AC-charging.

> Pin 2, M1b: Required for AC-charging.

> Pin 3, FB1: Required for AC-charging.

> Pin 5, PP1-PU: Not used.

Please be aware of the naming: The connector X303 which has the higher number corresponds to the logical CCS connector 1.

Vector recommends protecting the power line communication from interference.

2.2.5 X304 - Safety Outputs

1 REL1b	**3** REL2b	**5** REL3b
2 REL1a	**4** REL2a	**6** REL3a

Figure 11: vSECC connector: X304

This connector is used for safety purposes. It provides access to specialized outputs that add a layer of safety. They are intended to connect to the respective inputs of the power electronics circuitry. Please see the following paragraph on safety outputs, loss detection and CP/PP supervision for a general explanation of this mechanism.

The three safety outputs REL1, REL2 and REL3 serve the following safety functions:

> Pin 1 + 2, REL1: Safety output for IEC/SAE Connector 1 (CP and optionally PP) and Inverted Pantograph

> Pin 3 + 4, REL2: Safety output for IEC/SAE Connector 2 (CP and optionally PP) and GB/T

> Pin 5 + 6, REL3: Safety output for CHAdeMO

The two pins corresponding to each output are wired such that they are short-circuited if everything is fine and the respective output may be energized.
If the outlet must not be energized, the electric circuit remains open between the a and b pin.

In order to use the CCS Connector 2 with REL2 safety pins, the GB/T loss detection must be disabled by switching the corresponding DIP switches to ”ON”. See section 2.3.2 for details.

In compliance to the SAE J1772 requirements, the Proximity Detection needs to be activated for CCS Type 1 DC charging. With the activation, the PP of a connector influences the corresponding safety output. To activate the Proximity Detection for a connector, the corresponding DIP switches must be switched to ”OFF”. See also section 2.3.2 for details.

2.2.6 Safety Outputs: Loss Detection, Control Pilot / Proximity Pin Supervision

The following details apply to the IEC 61851 Control Pilot line used for AC and DC charging. The general principle holds, too, for IEC AC charging and the GB/T equivalent (AC and DC).

The IEC 61851 standard imposes strict safety requirements on the charging process and power supply monitoring. The charging process is controlled by the electric vehicle which sets a specific CP state. Four state categories exist: Ax, Bx, Cx and Dx. Energy transfer is allowed only in state categories Cx and Dx. In some cases (e.g. for CCS Type 1 connectors), a PP supervision is also required to prevent energy transfer when the PP signal is not valid.
In order to enforce this, the vSECC Controllers provide a logical output called *CP/PP supervision*.
This output controls the power electronics’ ability to energize its outlet. Conceptually, a logical AND conjunction exists in the power electronics between the Power Electronics Communication Controller’s (PECC) control input and the CP/PP supervision: The power electronics is able to close its contactors if and only if the CP/PP supervision allows it, i.e. the CP state category is Cx or Dx and the PP signal is valid (if applicable).
See Figure 12 for an illustration of this principle.

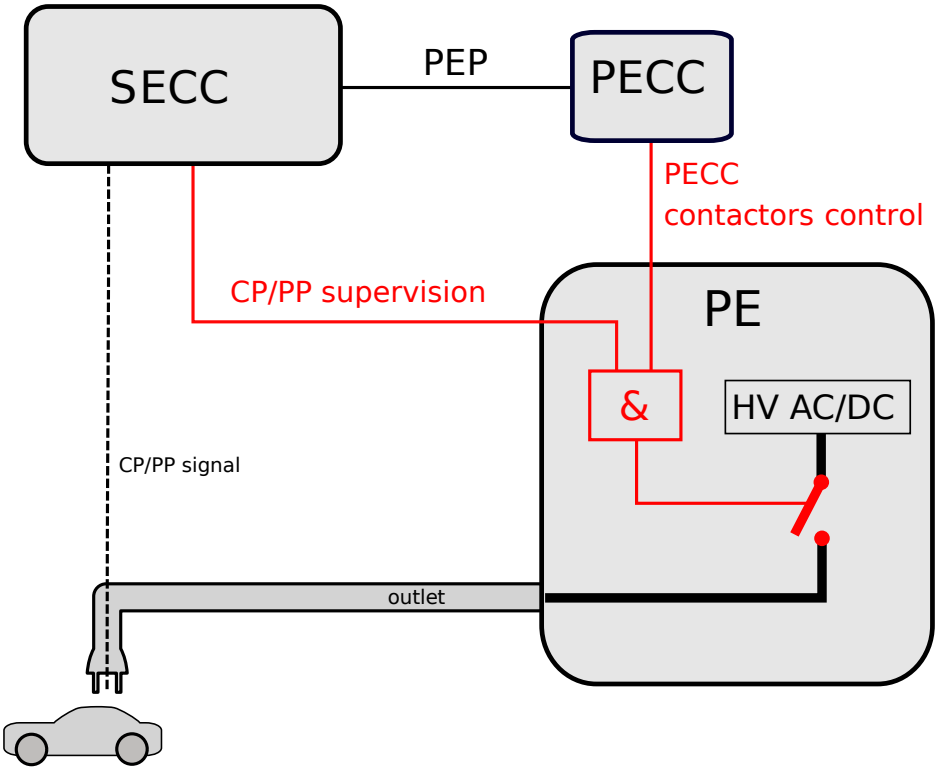


Figure 12: vSECC safety output

Control Pilot / Proximity Pin supervision: The EV communicates the charging state via the CP signal to the vSECC Controller. Depending on this state and the validity of the PP signal, the power electronics may or may not energize its outlet. The CP and PP signals are processed and provided as safety output directly to the power electronics. There, a logical AND conjunction of the input from the PECC and the vSECC safety output controls the high-voltage DC module (HVDC) output.

Caution: Please be aware that the power electronics is responsible for derating the provided current in accordance to IEC 61851 and SAE J1772, e.g. in case of an emergency shutdown.

2.2.7 X305 - CAN / Serial Interfaces

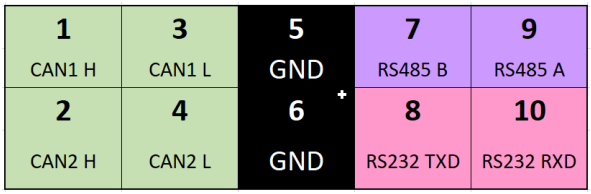


Figure 13: vSECC connector: X305

> CAN 1 is used when a charging connector is configured for CHAdeMO.
> CAN 2 is used for controlling a Power Electronics, e.g. via the PEP-CAN protocol (see the provided protocol description for further reference under vector.com/vsecc/documentation).
> RS232 is used for connecting an RFID reader (see section 8.10.4).
> RS485 is used to connect Modbus RTU slaves to the Modbus gateway (see section 8.20).

For information regarding termination, see section 2.3.2.

RS485 pin polarity: the noninverting pin is X305.9 (RS485 A) and the inverting pin is X305.7 (RS485 B).

2.2.8 X306 - Digital In and Digital Out Connectors, Connector Start/Stop Buttons, Pantograph Control

1 +24V	3 CN2 START	5 PANTO DOWN	7 CN1 STOP	9 DIN1	11 OUT15	13 OUT13	15 OUT11	17 OUT9	19 OUT7	21 OUT5	23 OUT3	25 OUT1
2 CN2 STOP	4 PANTO ERR	6 PANTO UP	8 CN1 START	10 PANTO CTRL	12 OUT14	14 OUT12	16 OUT10	18 OUT8	20 OUT6	22 OUT4	24 OUT2	26 GND

Figure 14: vSECC connector: X306

This connector is used for both digital input and digital output signals. See Section 8.18.5 and 8.18.6 for details and a mapping of PEP-identifiers to connector pins.

> Pin 1 outputs 24V.
> Pin 2-9 are active high digital inputs.
> Pin 10-25 are active high digital outputs.
> Pin 26 is the ground pin (GND).

In addition, four pins are used for buttons that can be used to start or stop a charging session.

Starting a charging session via button is currently only available for CHAdeMO.

See Section 8.4 for more details on the CHAdeMO charging process and for the implementation of the CHAdeMO **[EMERGENCY STOP]** button.

> Pin 8 (CN 1 START) is used for the Connector 1 **[START]** button (feature not yet available)
> Pin 7 (CN 1 STOP) is used for the Connector 1 **[STOP]** button.
> Pin 3 (CN 2 START) is used for the CHAdeMO **[START]** button.
> Pin 2 (CN 2 STOP) is used for the Connector 2/CHAdeMO **[STOP]** button.

When using Inverted Pantograph, four pins can be used for directly controlling a pantograph. Refer to Section 8.5 for more details on the Inverted Pantograph charging process and the options of controlling a pantograph.

> Pin 4 (PANTO ERR) is used as input to signal an error to the vSECC which prevents charging.
> Pin 5 (PANTO DOWN) is used as input to signal the vSECC, that the pantograph is in lower position.
> Pin 6 (PANTO UP) is used as input to signal the vSECC, that the pantograph is in upper position.
> Pin 10 (PANTO CTRL) is used as output to request moving the pantograph up (logical low) or down (logical high).

When using CCS AC charging, four pins are used for AC switch control and feedback. See Section 8.2 for more details on the AC charging charging process.

> Pin 12 (OUT14) is used as output to control the AC contactors of connector 1.
> Pin 11 (OUT15) is used as output to control the AC contactors of connector 2.
> Pin 4 (PANTO ERR) is used as input signal for contactors feedback of connector 1.
> Pin 9 (DIN1) is used as input signal for contactors feedback of connector 2.

2.2.9 X307 - Power Supply Connector

Figure 15: vSECC connector: X307

This connector is used for the supply voltage of $24\,\mathrm{V}$. The current drawn is typical below $300\,\mathrm{mA}$, though while booting the vSECC can draw up to $1.5\,\mathrm{A}$. So the power supply should provide at least $1.5\,\mathrm{A}$. Please refer to Section 10.1 for further details on power consumption.

Caution: Pressing the button above the X307 connector may cause a factory reset of the vSECC. See Section 2.3.1 for details.

2.2.10 ETH1 - Ethernet 1 (Back End / Inverted Pantograph)

This connector is used to connect network entities such as a Charging Station Management System (CSMS / Back End) or the Power Electronics Communication Controller (PECC) to the vSECC. If Value Added Services are used, this port must be used to connect the VAS-backend to the vSECC Controller. If the vSECC is configured for Inverted Pantograph, ETH1 must be used for connection to the Wi-Fi access point.

2.2.11 ETH2 - Ethernet 2

This connector is used to connect network entities to the vSECC in the same manner as it is possible with ETH1. The second port allows a higher flexibility, e.g., regarding network segmentation.

2.3 Buttons and Switches

2.3.1 Factory Reset Button

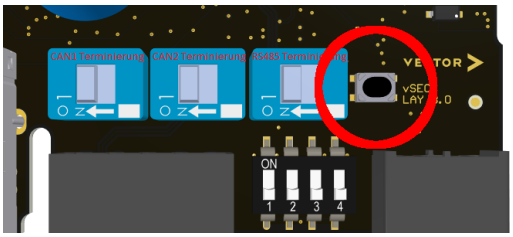


Figure 16: vSECC reset button in the lower right corner (top view)

This button is used to reset the configuration to the factory defaults. See Section 9.1 for details.

This functionality is given from vSECC version 1.3.0 upwards.

2.3.2 DIP Switches

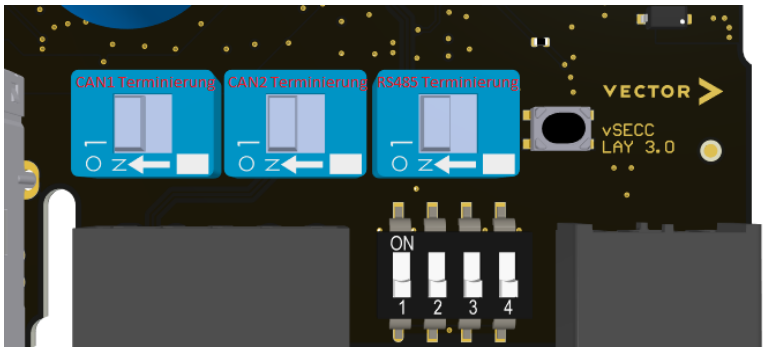


Figure 17: vSECC connector: DIP Switches

In the lower right corner, close to connectors X305 and X307, DIP switches allow the configuration of the termination for serial communication busses and the hardware supervision for the safety outputs.
The three blue switches allow the activation and deactivation of terminating resistors. All three enable the termination if switched to the left (on) and disable the termination if switched to the right.

- Left switch: CAN1 termination
- Middle switch: CAN2 termination
- Right switch: RS485 termination

In addition, four more switches reside between the X305 and X307 connectors.

Caution: These allow the deactivation of some safety-related functions. Disabling safety features may cause harm or serious injuries.

The safety supervision can be deactivated for some of the functions by flipping the respective switch to the **[ON]** position (upwards). If a function supervision has been deactivated, it is not considered for the result provided at the respective safety output.

The Proximity Pin is only relevant for CCS Type 1 and the switch for the respective connector must be set to **[OFF]** to enable CCS Type 1. For CCS Type 2, it must be set to **[ON]**.

Make sure to configure the matching CCS Type via the Web Interface. If PP loss detection is activated and CCS Type 2 is configured, the safety output will prevent energy transfer. If PP loss detection is deactivated and CCS Type 1 is configured, the connector will stay inoperative.

- Switch 1: PP1 loss detection deactivation
- Switch 2: PP2 loss detection deactivation
- Switch 3: CC1 loss AC (GB/T) detection deactivation

> Switch 4: CC1 loss DC (GB/T) detection deactivation

The supervision of CP1 and CP2 is always active and cannot be deactivated.

2.4 Use Cases: vSECC in Different Scenarios

This section details the electrical connections required for the most common use cases. Note that additional configuration may be required, e.g., setting the correct back end URI. Please use the Web Interface or an already connected CSMS to configure the vSECC (see Section 7.3). The use cases could be combined easily.

2.4.1 Use Case 1: vSECC Stand-alone Operation, CCS Charging Ready

The goal is to be able to start up the vSECC.

1. Mount the vSECC such that no cable is bent and electrical short-circuits are impossible.
2. Connect the X303 charging connector according to the pin descriptions depicted in Section 2.2.4. This plug relates to the first charging port.
3. Connect the X302 charging connector according to the pin descriptions depicted in Section 2.2.3. This plug relates to the second charging port.
4. Use the DIP switches to configure CCS Type 1 or Type 2 as described in Section 2.3.2.
5. Connect the ETH1 Ethernet port to an Ethernet network providing DHCP. This allows the configuration of the vSECC via the Web Interface.
6. Connect the X307 power supply connector. Take care of the correct polarity. Ensure that 24 V and at least 1.5 A are provided.
7. Open the Web Interface and configure CCS Type 1 or Type 2 for each connector accordingly (in Section **Configuration / Vehicle / Connector Type**).

It is important that the DIP switches are configured to match the CCS Type as configured in the Web Interface. If the DIP switch for PP loss detection is activated and CCS Type 2 is configured, the safety output will prevent energy transfer. If the DIP switch for PP loss detection is deactivated and CCS Type 1 is configured, the connector will stay inoperative.

The vSECC starts up as soon as the power supply is connected. The System LED (see Section 9.3) blinks green as long as the start-up is running. After the vSECC has finished initialization, the System LED turns green constantly.
The vSECC is now ready to be configured, e.g. for charging simulation purposes.

2.4.2 Use Case 2: vSECC with Power Electronics

The goal is to use a power electronics circuitry together with the vSECC.

1. Follow the Use Case 1 instructions above. Make sure that you do not connect the power supply yet.
2. Connect one of the Ethernet ports to an Ethernet network, which is providing access to the power electronics communication controller (PECC). This connection is used to control the power electronics via PEP-WS.
3. Connect the X304 safety output connector. Make sure that the pins for REL1 and REL2 are connected to the appropriate inputs of the power electronics itself. REL1 corresponds to the first charging connector and REL2 to the second.
4. Flip the DIP Switches 3 and 4 to the **[ON]** position. This disables the CC1 AC and CC1 DC loss detection.
5. (Optional) Connect the X301 (analog and temperature inputs) and X306 (digital inputs and outputs) connectors. This is necessary for the PECC to get input values or control digital outputs through PEP-WS. See Section 8.18.4 for the PEP identifiers that correspond to each pin.
6. Connect the X307 power supply connector. Take care of the correct polarity. Ensure that 24 V and at least 1.5 A are provided.

After the vSECC has started up, set the correct power electronics URI for both connectors using the Web Interface or CSMS.

2.4.3 Use Case 3: vSECC with CSMS

The goal is to use a Charging Station Management System (CSMS) to configure and manage the vSECC.

1. Follow the Use Case 1 instructions above. Make sure that you do not connect the power supply yet.
2. Connect one of the Ethernet ports to an Ethernet network, which is providing access to the CSMS.
3. Use the Web Interface (see Section 7) to set the correct back end URI and possibly login credentials.
4. Connect the X307 power supply connector. Take care of the correct polarity. Ensure that 24 V and at least 1.5 A are provided.

After the vSECC has started up, the vSECC tries to connect repeatedly to the CSMS using the configured URI and credentials.

When no Power Electronics are running and the PE Configuration in the Web Interface is not on *Simulation*, the vSECC appears in vCharM as **Inoperative**.

2.4.4 Use Case 4: vSECC and Roof-mounted Pantograph Charging

The goal is to be able to use the vSECC in a roof-mounted pantograph charging scenario.

Roof-mounted pantograph charging (”Panto-Up”) is supported by the vSECC. Regarding the vSECC, the physical connections, communication interfaces and procedures do not differ between normal CCS operation and roof-mounted pantograph charging. In particular, the Control Pilot (CP) handling, the SLAC procedure and High-Level Communication are the same. In this use-case the pantograph (including all its moving parts) is mounted on top of the vehicle. It is controlled by a separate device setup.
Please note that both the charging station and the EV must have the appropriate devices and controllers installed.

1. Follow the Use Case 2 instructions above.
2. Connect the X307 power supply connector. Take care of the correct polarity and sufficient power as defined in the technical data section.

The vSECC starts up as soon as the power supply is connected. The System LED (see Section 9.3) blinks green as long as the startup is running. After the vSECC has finished initialization, the System LED turns green constantly.

3 Installation Guide vSECC.MCS

In this chapter you will find the following information:

3.1 Physical Mounting

The vSECC.MCS is equipped with a mounting bracket which allows for an easy installation on a top-hat rail.

3.2 Electrical Connections

For wiring examples showing different use cases, please refer to Appendix G.

Figure 18 shows the vSECC.MCS from above. The MCS connector is described in detail below.

Figure 18: vSECC.MCS connector overview

We recommend purchasing the **vSECC Connector Kit**, please consult your Vector sales contact for details. Since the basis for the vSECC.MCS is the vSECC, the connectors are the same, with the additional X200 MCS connector. Please consult chapter 2.2 for details about the other connectors. Alternatively, the following connector types can be directly obtained from **Phoenix Contact** for connecting to the vSECC.MCS:

vSECC Connector	Phoenix Contact Connector Name	Order Key
X301	DFMC 1,5 / 10-ST-3,5	1790182
X302, X200	DFMC 1,5 / 4-ST-3,5	1790124
X304	DFMC 1,5 / 3-ST-3,5	1790111
X305	DFMC 1,5 / 5-ST-3,5	1790137
X306	DFMC 1,5 / 13-ST-3,5	1790218

X307	FKCN 2,5 / 2-ST-5,08	1754568

Caution: Please make sure that no pulling forces are applied on the wiring harness or connectors to make sure that no connector is getting pulled out.

3.2.1 X200 - MCS Charging Connector

1 PE	3 CE	5 REL1b	7 TRX-P
2 PE	4 ID	6 REL1a	8 TRX-N

Figure 19: vSECC.MCS connector: X200

This connector is used for MCS Charging which requires the following pins:

- Pin 1 + 2, PE: Protective Earth for MCS connector.
- Pin 3, CE: Charge Enable Pin for the MCS Charge Enable circuit.
- Pin 4, ID: Insert Detection Pin to indicate the insertion state of the MCS connector.
- Pin 5 + 6, REL1b and REL1a: Safety Output for CE and ID supervision.
- Pin 7 + 8, TRX-P and TRX-N: Digital communication Pins for 10Base-T1S according to ISO 15118-10.

This connector is also used for safety purposes when CCS charging. It provides access to specialized outputs that add a layer of safety. REL1 is intended to connect to the respective inputs of the power electronics circuitry. The two pins (REL1b + REL1a) are wired such that they are short-circuited if everything is fine and the respective output may be energized.
If the outlet must not be energized, the electric circuit remains open between the a and b pin.

3.2.2 X301 - Analog In and Temperature Sensor Connectors

1 0-10V 2	3 AGND	5 AGND	7 TEMP 8	9 AGND	11 TEMP 6	13 AGND	15 TEMP 4	17 AGND	19 TEMP 2
2 0-10V 1	4 TEMP 9	6 AGND	8 TEMP 7	10 AGND	12 TEMP 5	14 AGND	16 TEMP 3	18 AGND	20 TEMP 1

Figure 20: vSECC.MCS connector: X301

This connector is used for both analog input signals and external temperature sensors. See section 8.18.7 and 8.18.8 for details and a mapping of PEP-identifiers to connector pins.

3.2.3 X302 - CCS Charging Connector

1 n/a	3 n/a	5 PP-PU	7 CP
2 n/a	4 GND	6 PP	8 PE

Figure 21: vSECC.MCS connector: X302

This connector is used for CCS Charging which requires the following pins:

> Pin 6, PP: Proximity Pin for SAE J1772 Proximity Detection (only used for CCS Type 1).
> Pin 7, CP: Control Pilot line which corresponds to the respective pin of the CCS connector.
> Pin 8, PE: Protective Earth for CCS connector.

The following pins may be used in the future. For now, they are ignored:

> Pin 5, PP-PU: Not used.

Vector recommends protecting the power line communication from interference.

3.2.4 X304 - Safety Outputs

1 n/a	3 REL2b	5 n/a
2 n/a	4 REL2a	6 n/a

Figure 22: vSECC.MCS connector: X304

This connector is used for safety purposes. It provides access to specialized outputs that add a layer of safety. They are intended to connect to the respective inputs of the power electronics circuitry. Please see the following paragraph on safety outputs, loss detection and CP/PP supervision for a general explanation of this mechanism.

The safety output REL2 serve the following safety functions:

> Pin 3 + 4, REL2: Safety output for IEC/SAE Connector (CP and optionally PP)

The two pins corresponding to each output are wired such that they are short-circuited if everything is fine and the respective output may be energized.

If the outlet must not be energized, the electric circuit remains open between the a and b pin.

In order to use the CCS Connector with REL2 safety pins, the GB/T loss detection must be disabled by switching the corresponding DIP switches to ”ON”. See section 3.3.2 for details.

In compliance to the SAE J1772 requirements, the Proximity Detection needs to be activated for CCS Type 1 DC charging. With the activation, the PP of a connector influences the corresponding safety output. To activate the Proximity Detection for a connector, the corresponding DIP switches must be switched to ”OFF”. See also section 3.3.2 for details. For details about loss detection and CP/PP supervision, please visit chapter 2.2.6.

3.2.5 X305 - CAN / Serial Interfaces

1 CAN1 H	3 CAN1 L	5 GND	7 RS485 B	9 RS485 A
2 CAN2 H	4 CAN2 L	6 GND	8 RS232 TXD	10 RS232 RXD

Figure 23: vSECC.MCS connector: X305

> CAN 1 may be used in the future. For now, it is ignored.
> CAN 2 is used for controlling a Power Electronics, e.g. via the PEP-CAN protocol (see the provided protocol description for further reference under vector.com/vsecc/documentation).
> RS232 is used for connecting an RFID reader (see section 8.10.4).
> RS485 is used to connect Modbus RTU slaves to the Modbus gateway (see section 8.20).

For information regarding termination, see section 3.3.2.

RS485 pin polarity: the noninverting pin is X305.9 (RS485 A) and the inverting pin is X305.7 (RS485 B).

3.2.6 X306 - Digital In and Digital Out Connectors

1 24V	3 DIN7	5 DIN5	7 CN1 STOP	9 DIN1	11 OUT15	13 OUT13	15 OUT11	17 OUT9	19 OUT7	21 OUT5	23 OUT3	25 OUT1
2 CN2 STOP	4 DIN6	6 DIN4	8 DIN2	10 OUT16	12 OUT14	14 OUT12	16 OUT10	18 OUT8	20 OUT6	22 OUT4	24 OUT2	26 GND

Figure 24: vSECC.MCS connector: X306

This connector is used for both digital input and digital output signals. See Section 8.18.5 and 8.18.6 for details and a mapping of PEP-identifiers to connector pins.

> Pin 1 outputs 24V.
> Pin 2-9 are active high digital inputs.
> Pin 10-25 are active high digital outputs.
> Pin 26 is the ground pin (GND).

In addition, two pins are used for buttons that can be used to stop a charging session.

> Pin 7 (CN1 STOP) is used for the MCS Connector **[STOP]** button.
> Pin 2 (CN2 STOP) is used for the CCS Connector **[STOP]** button.

3.2.7 X307 - Power Supply Connector

Figure 25: vSECC.MCS connector: X307

This connector is used for the supply voltage of 24 V. The current drawn is typical below 300 mA, though while booting the vSECC.MCS can draw up to 1.5 A. So the power supply should provide at least 1.5 A. Please refer to Section 10.1 for further details on power consumption.

Caution: Pressing the button above the X307 connector may cause a factory reset of the vSECC.MCS. See Section 3.3.1 for details.

3.2.8 ETH1 - Ethernet 1 (Back End)

This connector is used to connect network entities such as a Charging Station Management System (CSMS / Back End) or the Power Electronics Communication Controller (PECC) to the vSECC.MCS. If Value Added Services are used, this port must be used to connect the VAS-backend to the vSECC Controller.

3.2.9 ETH2 - Ethernet 2

This connector is used to connect network entities to the vSECC.MCS in the same manner as it is possible with ETH1. The second port allows a higher flexibility, e.g., regarding network segmentation.

Vector recommends using shielded Ethernet cables to connect the network entities.

3.3 Buttons and Switches

3.3.1 Factory Reset Button

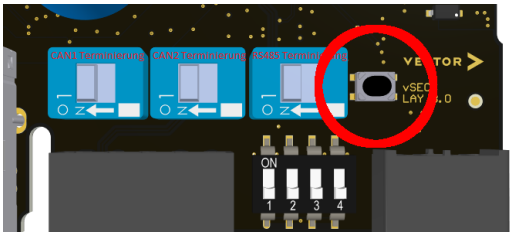


Figure 26: vSECC.MCS reset button in the lower right corner (top view)

This button is used to reset the configuration to the factory defaults. See Section 9.1 for details.

3.3.2 DIP Switches

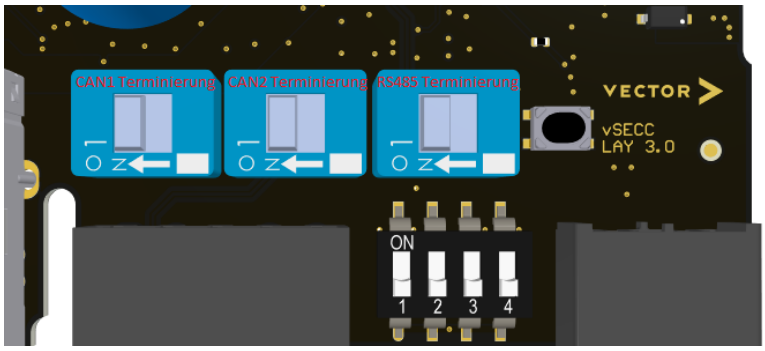


Figure 27: vSECC.MCS connector: DIP Switches

In the lower right corner, close to connectors X305 and X307, DIP switches allow the configuration of the termination for serial communication busses and the hardware supervision for the safety outputs.
The three blue switches allow the activation and deactivation of terminating resistors. All three enable the termination if switched to the left (on) and disable the termination if switched to the right.

- Left switch: CAN1 termination
- Middle switch: CAN2 termination
- Right switch: RS485 termination

In addition, four more switches reside between the X305 and X307 connectors.

Caution: These allow the deactivation of some safety-related functions. Disabling safety features may cause harm or serious injuries.

Caution: The safety supervision could only be deactivated for CCS (second charging port). These switches are not relevant for MCS safety functions. The MCS safety functions on the first charging port are always enabled.

The safety supervision can be deactivated for some of the functions by flipping the respective switch to the **[ON]** position (upwards). If a function supervision has been deactivated, it is not considered for the result provided at the respective safety output.

The Proximity Pin is only relevant for CCS Type 1 and the switch for the respective connector must be set to **[OFF]** to enable CCS Type 1. For CCS Type 2, it must be set to **[ON]**.

Make sure to configure the matching CCS Type via the Web Interface. If PP loss detection is activated and CCS Type 2 is configured, the safety output will prevent energy transfer. If PP loss detection is deactivated and CCS Type 1 is configured, the connector will stay inoperative.

> Switch 1: is not used
> Switch 2: PP loss detection deactivation
> Switch 3: CC1 loss AC (GB/T) detection deactivation
> Switch 4: CC1 loss DC (GB/T) detection deactivation

The supervision of CP is always active and cannot be deactivated.

3.4 Use Cases: vSECC.MCS in Different Scenarios

This section details the electrical connections required for the most common use cases. Note that additional configuration may be required, e.g., setting the correct back end URI. Please use the Web Interface or an already connected CSMS to configure the vSECC.MCS (see Section 7.3). The use cases could be combined easily.

3.4.1 Use Case 1: vSECC.MCS Stand-alone Operation, MCS Charging Ready

The goal is to be able to start up the vSECC.MCS.

1. Mount the vSECC.MCS such that no cable is bent and electrical short-circuits are impossible.
2. Connect the X200 charging connector (MCS) according to the pin descriptions depicted in Section 3.2.1. This plug relates to the first charging port.
3. Connect the ETH1 Ethernet port to an Ethernet network providing DHCP. This allows the configuration of the vSECC.MCS via the Web Interface.
4. Connect the X307 power supply connector. Take care of the correct polarity. Ensure that 24 V and at least 1.5 A are provided.
5. Open the Web Interface and configure the MCS connector (in Section **Configuration / Vehicle / Connector Type**).

3.4.2 Use Case 2: vSECC.MCS Stand-alone Operation, MCS and CCS Charging Ready

The goal is to be able to start up the vSECC.MCS with both connectors.

1. Mount the vSECC.MCS such that no cable is bent and electrical short-circuits are impossible.
2. Connect the X200 charging connector (MCS) according to the pin descriptions depicted in Section 3.2.1. This plug relates to the first charging port (MCS).
3. Connect the X302 charging connector (CCS) according to the pin descriptions depicted in Section 3.2.3. This plug relates to the second charging port (CCS).
4. Use the DIP switches to configure CCS Type 1 or Type 2 as described in Section 2.3.2.
5. Connect the ETH1 Ethernet port to an Ethernet network providing DHCP. This allows the configuration of the vSECC.MCS via the Web Interface.
6. Connect the X307 power supply connector. Take care of the correct polarity. Ensure that 24 V and at least 1.5 A are provided.
7. Open the Web Interface and configure MCS for connector 1 and CCS Type 1 or Type 2 for connector 2 (in Section **Configuration / Vehicle / Connector Type**).

The vSECC.MCS starts up as soon as the power supply is connected. The System LED (see Section 9.3) blinks green as long as the start-up is running. After the vSECC.MCS has finished initialization, the System LED turns green constantly.
The vSECC.MCS is now ready to be configured, e.g. for charging simulation purposes.

3.4.3 Use Case 3: vSECC.MCS with Power Electronics

The goal is to use a power electronics circuitry together with the vSECC.MCS.

1. Follow the Use Case 1 instructions above. Make sure that you do not connect the power supply yet.
2. Connect one of the Ethernet ports to an Ethernet network, which is providing access to the power electronics communication controller (PECC). This connection is used to control the power electronics via PEP-WS.
3. Connect the X200 safety output connector. Make sure that the pins for REL1 are connected to the appropriate inputs of the power electronics itself.
4. Flip the DIP Switches 3 and 4 to the **[ON]** position.This disables the CC1 AC and CC1 DC loss detection.
5. (Optional) Connect the X301 (analog and temperature inputs) and X306 (digital inputs and outputs) connectors. This is necessary for the PECC to get input values or control digital outputs through PEP-WS. See Section 8.18.4 for the PEP identifiers that correspond to each pin.
6. Connect the X307 power supply connector. Take care of the correct polarity. Ensure that 24 V and at least 1.5 A are provided.

After the vSECC.MCS has started up, set the correct power electronics URI for both connectors using the Web Interface or CSMS.

3.4.4 Use Case 4: vSECC.MCS with CSMS

The goal is to use a Charging Station Management System (CSMS) to configure and manage the vSECC.MCS.

1. Follow the Use Case 1 instructions above. Make sure that you do not connect the power supply yet.
2. Connect one of the Ethernet ports to an Ethernet network, which is providing access to the CSMS.
3. Use the Web Interface (see Section 7) to set the correct back end URI and possibly login credentials.
4. Connect the X307 power supply connector. Take care of the correct polarity. Ensure that 24 V and at least 1.5 A are provided.

After the vSECC.MCS has started up, the vSECC.MCS tries to connect repeatedly to the CSMS using the configured URI and credentials.

When no Power Electronics are running and the PE Configuration in the Web Interface is not on *Simulation*, the vSECC.MCS appears in vCharM as **Inoperative**.

The vSECC.MCS starts up as soon as the power supply is connected. The System LED (see Section 9.3) blinks green as long as the startup is running. After the vSECC.MCS has finished initialization, the System LED turns green constantly.

4 Installation Guide vSECC.single Board

In this chapter you will find the following information:

4.1 Physical Mounting

The vSECC.single Board is designed to be mounted on a base board PCB. Further details can be found in the mechanical drawings in Appendix C.

4.2 Electrical Connections

Figure 28 shows the top view of the vSECC.single Board with the available connectors. Each connector is described in detail below.

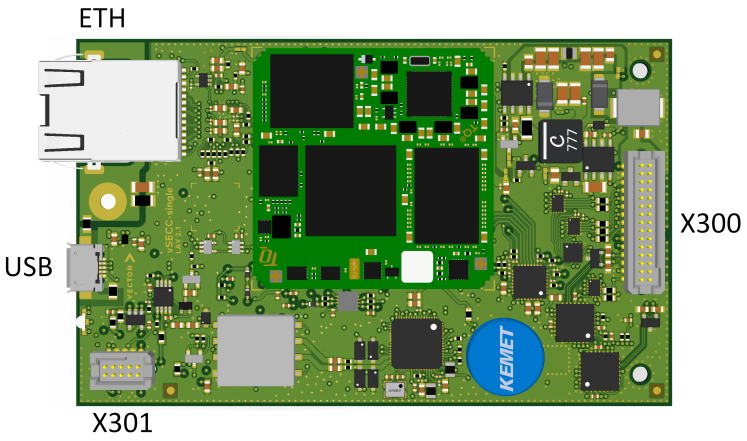


Figure 28: vSECC.single Board Top View

We recommend purchasing the following connector types from **Samtec** for connecting to the vSECC.single Board:

Connector	Manufacturer	Part Number
X300	Samtec	SFM-115-02-L-D-A
X301	Samtec	SFM-105-02-L-D-A

4.2.1 Connector X300

Connector X300 contains pins for the following functional blocks:

- Supply power to the vSECC.single Board
- Digital inputs and outputs
- Analog inputs
- Safety output from hardware supervision circuit
- Serial communication

The whole pinout is shown in the table below, the usage is described in the following sections.

PIN	Signal Name	PIN	Signal Name
1	VCC_SUP	2	VCC_SUP
3	GND	4	GND
5	VCC_LOGIC	6	DIO1
7	DIO2	8	DIO3
9	DIO4	10	DIO5
11	DIO6	12	DIO7
13	DIO8	14	DIO9
15	DIO10	16	VCC_ADC
17	GND_ADC	18	AIN1
19	AIN2	20	AIN3
21	AIN4	22	SAFETY_OUT
23	CAN1_HIGH	24	RS485_A
25	CAN1_LOW	26	RS485_B
27	RS232_RXD_TTL	28	CAN2_HIGH
29	RS232_TXD_TTL	30	CAN2_LOW

4.2.2 Connector X300 - Power Supply Connection

The vSECC.single Board must be powered with 12 V by the attached base board. Connect both pin X300.1 and pin X300.2 with the supply voltage. Connect both pin X300.3 and X300.4 with the corresponding ground.

These connectors are used for the supply voltage of 12 V. The current drawn is typical below 400 mA, though while booting the vSECC.single Board can draw up to 500 mA. Please refer to Section 11.1 for further details on power consumption.

4.2.3 Connector X300 - Digital Inputs/Outputs

The vSECC.single Board offers 10 digital inputs/outputs on pin X300.6 through X300.15. In order to enable usage of opto-couplers or a level shifter, an output of a matching supply voltage (VCC_LOGIC at pin X300.5) for these circuits is provided.
When using CCS AC charging, two pins are used for AC switch control and feedback. See Section 8.2 for more details on the AC charging charging process.

> Pin 6 (DIO1) is used as output to control the AC contactors of connector 1.

> Pin 11 (DIO6) is used as input signal for contactors feedback of connector 1.

.

4.2.4 Connector X300 - Analog Inputs

4 analog inputs are available on pin X300.18 through X300.21. Use pin X300.17 to connect GND for the analog circuitry. If you intend to supply power to the connected sensor, use pin 300.16 (VCC_ADC).

4.2.5 Connector X300 - Serial Communication

Caution: CAN2 currently cannot be used on the vSECC.single Board.

The vSECC.single Board offers the following serial communication interfaces:

- CAN1 on X300.23 and X300.25 with fixed termination, for communication with the power electronics
- CAN2 on X300.28 and X300.30 with fixed termination, currently not in use
- RS232 on X300.27 and X300.29, for connection of supported RFID-Readers (see section 8.10.4)
- RS485 on X300.24 and X300.26, for connection of Modbus RTU slaves to the Modbus gateway (see section 8.20)

RS485 pin polarity: the noninverting pin is X300.24 (RS485_A) and the inverting pin is X300.26 (RS485_B).

4.2.6 Connector X300 - Safety Output

This output is used for safety purposes. It provides access to a specialized output that adds a layer of safety. It is intended to connect to the respective input of the power electronics circuitry.
Please be aware that the vSECC.single Board also includes supervision of two temperature sensors as described in Section 4.2.7. For further information on proper wiring please refer to Section 4.4 and Section 11.5.
Please see Section 2.2.6 on loss detection and CP/PP supervision for a general explanation of this mechanism.

4.2.7 Connector X301

Connector X301 contains all signals for charging communication and the temperature sensors. The whole pinout is shown in the table below, the detailed description is located in the following sections.

PIN	Signal Name	PIN	Signal Name
1	PP	2	CP

3	PP_PU	4	GND
5	GND	6	GND
7	GND_TEMP	8	GND_TEMP
9	AIN_TEMP_2	10	AIN_TEMP_1

For DC charging, the following pins are needed:

> Pin X301.1, PP: Proximity Pin for SAE J1772 Proximity Detection (only used for CCS Type 1).
> Pin X301.2, CP: Control Pilot line which corresponds to the respective pin of the CCS connector.
> Pin X301.4, GND: Must be connected to Protective Earth of the CCS connector.

Furthermore, as the vSECC.single Board does not allow to disable PP supervision by setting a switch or something similar, the following connections have to be made for CCS Type 2:

> Connect PP_PU (X301.3) with PP (X301.1).
> Add a resistor of $142\ \Omega$ between PP and GND (X301.4-6).

In contrast to the vSECC, the vSECC.single Board includes supervision of two temperature sensors directly with the hardware supervision circuitry. Thus, the safety output will only be active if the temperature sensors connected to the AIN_TEMP_1 (X301.10) and AIN_TEMP_2 (X301.9) are in the specified range (see Section 11.5).

Vector recommends protecting the power line communication from interference.

4.2.8 Ethernet

This connector is used to connect network entities such as a Charging Station Management System (CSMS / Back End) or the Power Electronics Communication Controller (PECC) to the vSECC. This connector can also be used to configure the vSECC.single Board via the web configuration interface.

4.2.9 USB

Caution: The USB connector is currently not in use.

4.3 Factory Reset Button

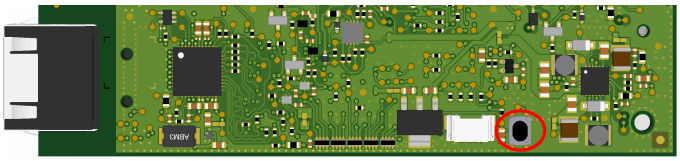


Figure 29: Factory Reset Button on vSECC.single Board

The factory reset button is located on the bottom of the vSECC.single Board. This button is used to reset the configuration to the factory defaults. See Section 9.1 for details.

4.4 Wiring Examples

In this section, some wiring examples for the vSECC.single Board in common use cases are shown.

4.4.1 CCS Type 1 Charging

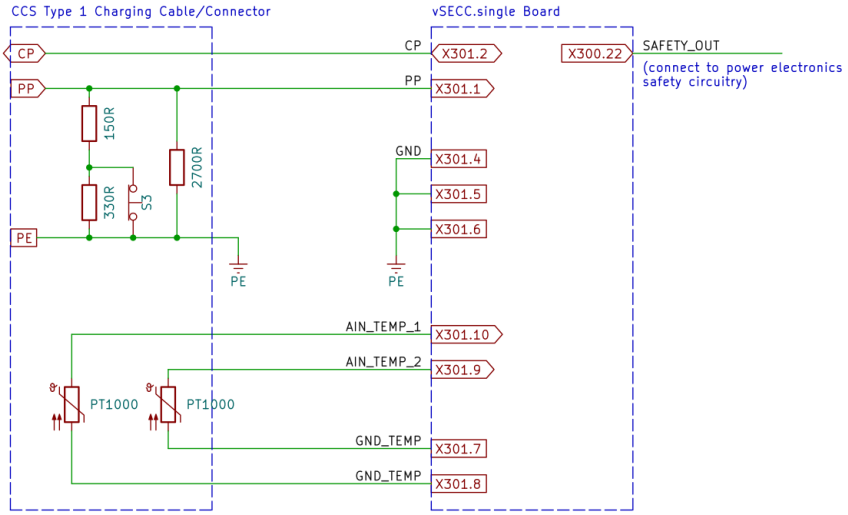


Figure 30: Wiring Diagram vSECC.single Board CCS Type 1

In order to use a CCS Type 1 charging connector, the following connections are necessary:

- CP: Connect X301.2 of the vSECC.single Board to the CP pin of the charging cable.
- PP: Connect X301.1 of the vSECC.single Board to the PP pin of the charging cable (usually, the resistors between PP and PE are included in the connector assembly).
- PE: Please make sure that the PE of the charging cable is connected to pins X301.4 through X301.6.
- Temperature sensors: Please make sure to connect the temperature sensors from the connector assembly to X301.10 and X301.9 and use X301.7 and X301.8 for GND.

4.4.2 CCS Type 2 Charging

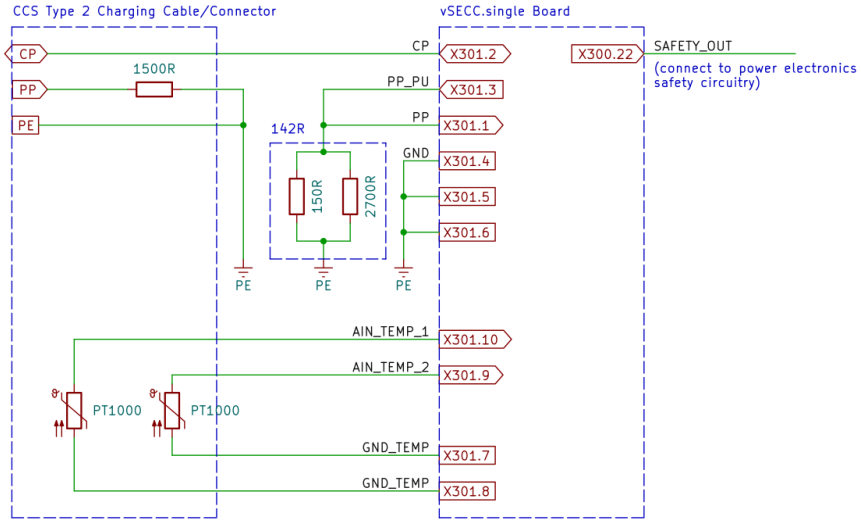


Figure 31: Wiring Diagram vSECC.single Board CCS Type 2

In order to use a CCS Type 2 charging connector, the following connections are necessary:

> CP: Connect X301.2 of the vSECC.single Board to the CP pin of the charging cable.

> PP: Check if the connector assembly has an internal resistor for PE. You may have to connect the resistor to PE manually or even provide the $1.5\,\mathrm{k}\Omega$ resistor on your base board. Also make sure to deactivate the PP supervision as shown in the example.

> PE: Please make sure that the PE of the charging cable is connected to pins X301.4 through X301.6.

> Temperature sensors: Please make sure to connect the temperature sensors from the connector assembly to X301.10 and X301.9 and use X301.7 and X301.8 for GND.

5 Installation Guide vSECC.single

In this chapter you will find the following information:

5.1 Physical Mounting

The vSECC.single is equipped with a mounting bracket which allows for an easy installation on a top-hat rail. The vSECC.single Board is mounted through the connector interface X300 and X301.

Caution: The vSECC.single must be installed in such a way that it cannot be touched from the outside.

5.2 Electrical Connections

Figure 32 shows the top view of the vSECC.single with the available connectors. Each connector is described in detail below.

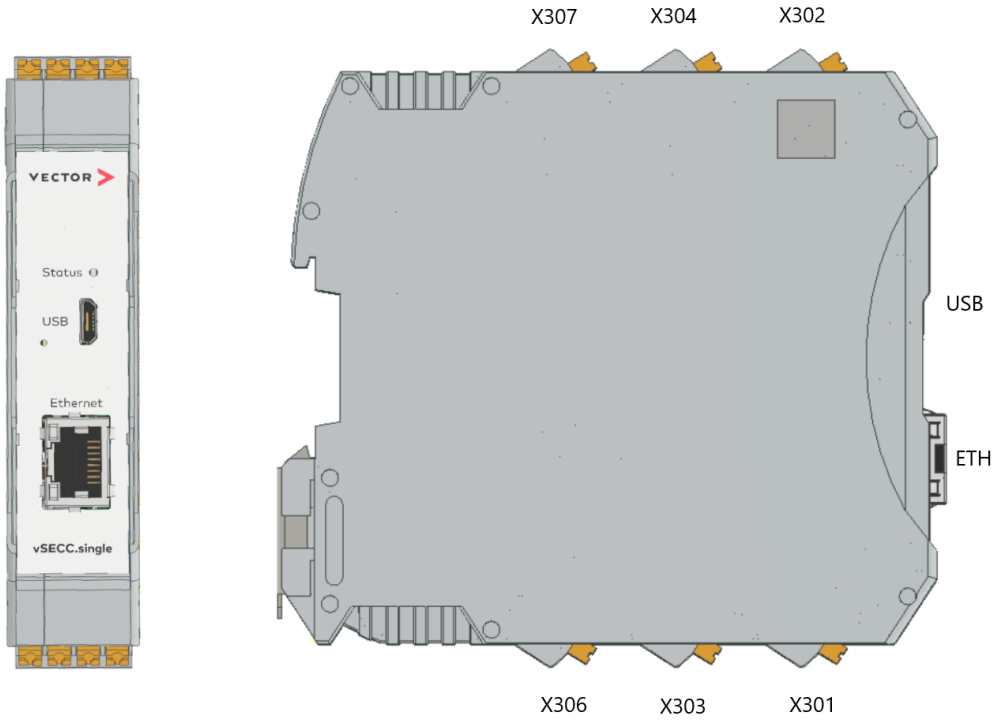


Figure 32: vSECC.single Front and Side View

All connectors on the vSECC.single are from the same manufacture type and ready to use for wiring. Following connnectors are populated on the vSECC.single:

Connector	Manufacturer	Part Number	Description
X301	Phoenix	2200319	4-pin, pitch: 5mm
X302	Phoenix	2200320	4-pin, pitch: 5mm
X303	Phoenix	2200319	4-pin, pitch: 5mm
X304	Phoenix	2200320	4-pin, pitch: 5mm
X306	Phoenix	2200319	4-pin, pitch: 5mm

X307	Phoenix	2200320	4-pin, pitch: 5mm

The whole pinout is shown in the figure below, the detailed description is located in the following sections.

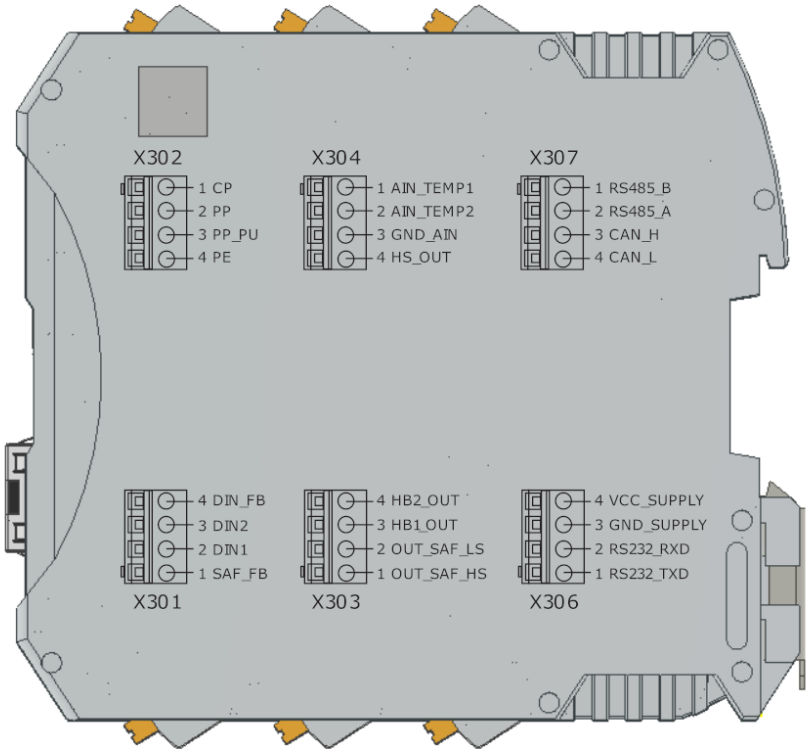


Figure 33: vSECC.single Pinout

5.2.1 Connector X301

Connector X301 contains all signals for digital inputs.

PIN	Signal Name
1	SAF_FB
2	DIN1
3	DIN2
4	DIN_FB

This connector is used for safety feedback and digital inputs, which requires the following pins:

- Pin 1, SAF_FB: Safety feedback
- Pin 2, DIN1: Digital input 1
- Pin 3, DIN2: Digital input 2
- Pin 4, DIN_FB: Digital input feedback

When using CCS AC charging, two pins are used for AC switch control and feedback (Connector X301 and X303). See Section 8.2 for more details on the AC charging charging process.

> Pin 2 (DIN1) is used as input signal for contactors feedback of connector 1.

5.2.2 Connector X302

Connector X302 contains all signals for charging communication.

PIN	Signal Name
1	CP
2	PP
3	PP_PU
4	PE

This connector is used for CCS Charging. Today, only DC charging is supported, which requires the following pins:

> Pin 1, CP: Control Pilot line which corresponds to the respective pin of the CCS connector.
> Pin 2, PP: Proximity Pin for SAE J1772 Proximity Detection (only used for CCS Type 1).
> Pin 3, PP_PU: Only used for CCS Type 2.
> Pin 4, PE: Protective Earth for CCS connector.

Furthermore, as the vSECC.single does not allow to disable PP supervision by setting a switch or something similar, the following connections have to be made for CCS Type 2:

> Connect PP_PU (X302.3) with PP (X302.2).
> Add a resistor of $142\,\Omega$ between PP (X302.2) and PE (X302.4).

Vector recommends protecting the power line communication from interference.

5.2.3 Connector X303

Connector X303 contains all signals for safety output (high side and low side switch) and digital outputs.

PIN	Signal Name
1	OUT_SAF_HS
2	OUT_SAF_LS

3	HB1_OUT
4	HB2_OUT

This connector is used for safety output and digital outputs:

> Pin 1, OUT_SAF_HS: Safety output (high side switch). For further information on proper wiring please refer to Section 5.4.
> Pin 2, OUT_SAF_LS: Safety output (low side switch). For further information on proper wiring please refer to Section 5.4.
> Pin 3, HB1_OUT: Half bridge 1 output
> Pin 4, HB2_OUT: Half bridge 2 output

When using CCS AC charging, two pins are used for AC switch control and feedback (Connector X301 and X303). See Section 8.2 for more details on the AC charging charging process.

> Pin 3 (HB1_OUT) is used as output to control the AC contactors of connector 1.

5.2.4 Connector X304

Connector X304 contains all signals for temperature sensors and digital outputs.

PIN	Signal Name
1	AIN_TEMP1
2	AIN_TEMP2
3	GND_AIN
4	HS_OUT

This connector is used for external temperature senors and digital outputs.

> Pin 1, AIN_TEMP1: Analog input for temperature sensor 1.
> Pin 2, AIN_TEMP2: Analog input for temperature sensor 2.
> Pin 3, GND_AIN: Ground for analog input temperature sensor 1 and 2.
> Pin 4, HS_OUT: High side switch output

In contrast to the vSECC, the vSECC.single includes supervision of two temperature sensors directly with the hardware supervision circuitry. Thus, the safety output will only be active if the temperature sensors connected to the AIN_TEMP_1 (X304.1) and AIN_TEMP_2 (X304.2) are in the specified range (see Section 12.4).

5.2.5 Connector X306

Connector X306 contains all signals for power supply connection and RS232 serial communication.

PIN	Signal Name
1	RS232_TXD
2	RS232_RXD
3	GND_SUPPLY
4	VCC_SUPPLY

> RS232 on X306.1 and X306.2, for connection of supported RFID-Readers (see section 8.10.4).

> The vSECC.single is typically powered with 12 V. Connect pin X306.4 with the supply voltage. Connect pin X306.3 with the corresponding ground. These connectors are used for the supply voltage of 12 V. The current drawn is typical below 400 mA, though while booting the vSECC.single can draw up to 500 mA. Please refer to Section 12.1 for further details on power consumption.

Caution: For EMC reasons, the maximum cable length for RS232 and RS485 should be 3 meters.

5.2.6 Connector X307

Connector X307 contains all signals for RS485 and CAN communication.

PIN	Signal Name
1	RS485_B
2	RS485_A
3	CAN_H
4	CAN_L

The vSECC.single offers the following serial communication interfaces:

> CAN on X307.3 and X307.4 with fixed termination, for communication with the power electronics.

> RS485 on X307.1 and X307.2, for connection of Modbus RTU slaves to the Modbus gateway (see section 8.20).

RS485 pin polarity: the noninverting pin is X307.2 (RS485_A) and the inverting pin is X307.1 (RS485_B).

Caution: For EMC reasons, the maximum cable length for RS485 and RS232 should be 3 meters.

5.2.7 Ethernet

This connector is used to connect network entities such as a Charging Station Management System (CSMS / Back End) or the Power Electronics Communication Controller (PECC) to the vSECC.single. This connector can also be used to configure the vSECC.single via the web configuration interface.

Caution: For EMC reasons, please use shielded Ethernet cables only.

5.2.8 USB

Caution: The USB connector is currently not in use.

5.3 Factory Reset Button

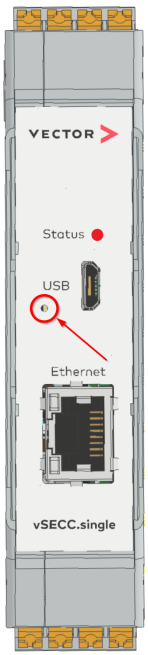


Figure 34: Factory Reset Button on vSECC.single

The factory reset button is located behind the small hole next to the USB port of the vSECC.single. This button is used to reset the configuration to the factory defaults. See Section 9.1

for details.

5.4 Wiring Examples

In this section, some wiring examples for the vSECC.single in common use cases are shown.

5.4.1 CCS Type 1 Charging

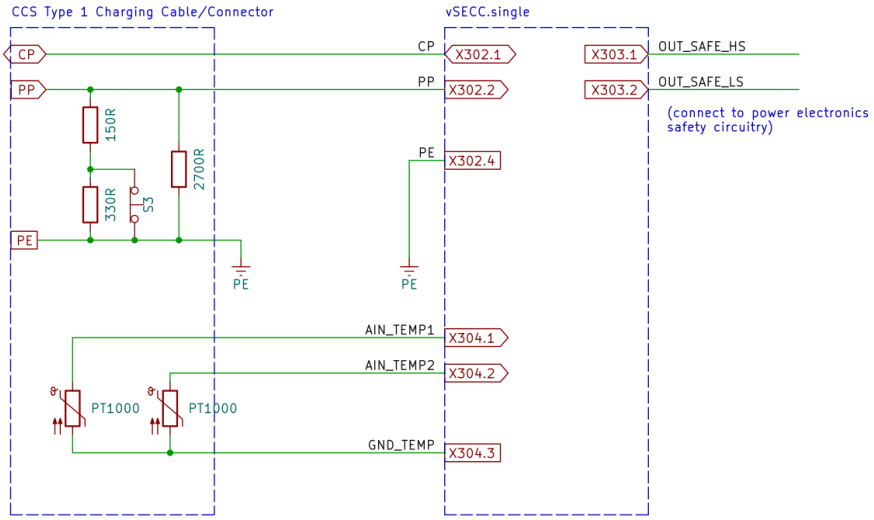


Figure 35: Wiring Diagram vSECC.single CCS Type 1

In order to use a CCS Type 1 charging connector, the following connections are necessary:

> CP: Connect X302.1 of the vSECC.single to the CP pin of the charging cable.

> PP: Connect X302.2 of the vSECC.single to the PP pin of the charging cable (usually, the resistors between PP and PE are included in the connector assembly).

> PE: Please make sure that the PE of the charging cable is connected to pin X302.4.

> Temperature sensors: Please make sure to connect the temperature sensors from connector assembly to X304.1 and X304.2 and use X304.3 for GND.

5.4.2 CCS Type 2 Charging

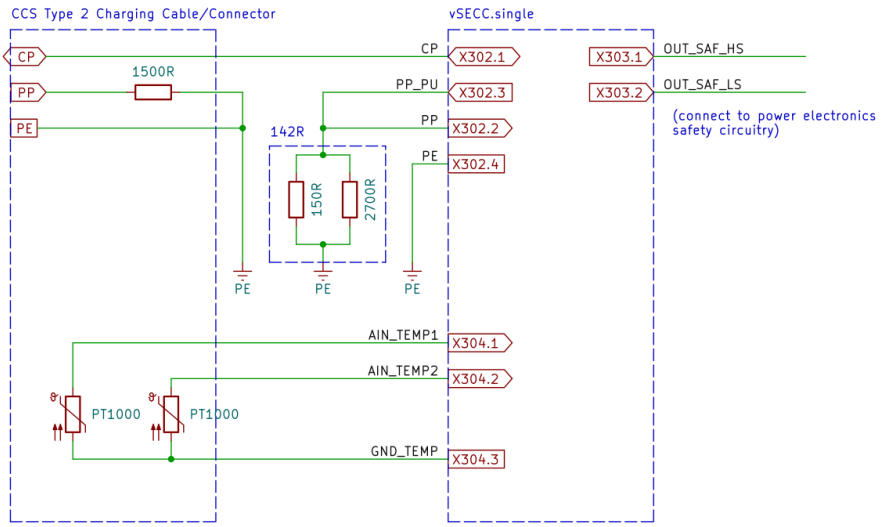


Figure 36: Wiring Diagram vSECC.single CCS Type 2

In order to use a CCS Type 2 charging connector, the following connections are necessary:

> CP: Connect X302.1 of the vSECC.single to the CP pin of the charging cable.

> PP: Check if the connector assembly has an internal resistor for PE. You may have to connect the resistor to PE manually or even provide the $1.5\,\mathrm{k}\Omega$ resistor on your base board. Also make sure to deactivate the PP supervision as shown in the example.

> PE: Please make sure that the PE of the charging cable is connected to pin X302.4.

> Temperature sensors: Please make sure to connect the temperature sensors from the connector assembly to X304.1 and X304.2 and use X304.3 for GND.

6 Installation Guide vSECC.single +70°C

In this chapter you will find the following information:

6.1 Physical Mounting

The vSECC.single +70°C is equipped with a mounting bracket which allows for an easy installation on a top-hat rail. The vSECC.single +70°C Board is mounted through the connector interface X300 and X301.

Caution: The vSECC.single +70°C must be installed in such a way that it cannot be touched from the outside.

6.2 Electrical Connections

Figure 37 shows the top view of the vSECC.single +70°C with the available connectors. Each connector is described in detail below.

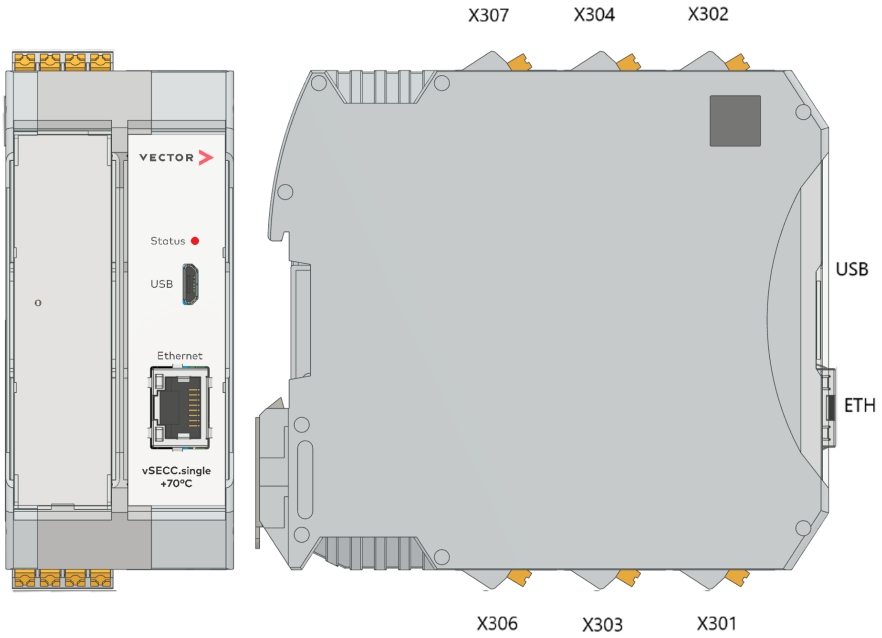


Figure 37: vSECC.single +70°C Front and Side View

All connectors on the vSECC.single +70°C are from the same manufacture type and ready to use for wiring. Following connnectors are populated on the vSECC.single +70°C:

Connector	Manufacturer	Part Number	Description
X301	Phoenix	2200319	4-pin, pitch: 5mm
X302	Phoenix	2200320	4-pin, pitch: 5mm
X303	Phoenix	2200319	4-pin, pitch: 5mm
X304	Phoenix	2200320	4-pin, pitch: 5mm
X306	Phoenix	2200319	4-pin, pitch: 5mm
X307	Phoenix	2200320	4-pin, pitch: 5mm

The whole pinout is shown in the figure below, the detailed description is located in the following sections.

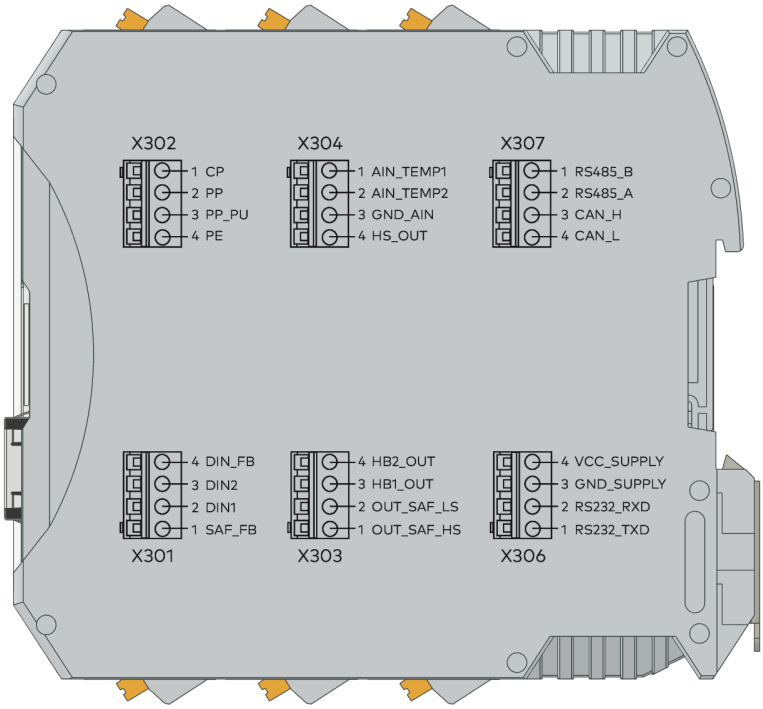


Figure 38: vSECC.single +70°C Pinout

6.2.1 Connector X301

Connector X301 contains all signals for digital inputs.

PIN	Signal Name
1	SAF_FB
2	DIN1
3	DIN2
4	DIN_FB

This connector is used for safety feedback and digital inputs, which requires the following pins:

> Pin 1, SAF_FB: Safety feedback

> Pin 2, DIN1: Digital input 1

> Pin 3, DIN2: Digital input 2

> Pin 4, DIN_FB: Digital input feedback

When using CCS AC charging, two pins are used for AC switch control and feedback (Connector X301 and X303). See Section 8.2 for more details on the AC charging charging process.

> Pin 2 (DIN1) is used as input signal for contactors feedback of connector 1.

6.2.2 Connector X302

Connector X302 contains all signals for charging communication.

PIN	Signal Name
1	CP
2	PP
3	PP_PU
4	PE

This connector is used for CCS Charging. Today, only DC charging is supported, which requires the following pins:

> Pin 1, CP: Control Pilot line which corresponds to the respective pin of the CCS connector.

> Pin 2, PP: Proximity Pin for SAE J1772 Proximity Detection (only used for CCS Type 1).

> Pin 3, PP_PU: Only used for CCS Type 2.

> Pin 4, PE: Protective Earth for CCS connector.

Furthermore, as the vSECC.single +70°C does not allow to disable PP supervision by setting a switch or something similar, the following connections have to be made for CCS Type 2:

> Connect PP_PU (X302.3) with PP (X302.2).

> Add a resistor of $142\,\Omega$ between PP (X302.2) and PE (X302.4).

Vector recommends protecting the power line communication from interference.

6.2.3 Connector X303

Connector X303 contains all signals for safety output (high side and low side switch) and digital outputs.

PIN	Signal Name
1	OUT_SAF_HS
2	OUT_SAF_LS
3	HB1_OUT
4	HB2_OUT

This connector is used for safety output and digital outputs:

> Pin 1, OUT_SAF_HS: Safety output (high side switch). For further information on proper wiring please refer to Section 5.4.

> Pin 2, OUT_SAF_LS: Safety output (low side switch). For further information on proper wiring please refer to Section 5.4.
> Pin 3, HB1_OUT: Half bridge 1 output
> Pin 4, HB2_OUT: Half bridge 2 output

When using CCS AC charging, two pins are used for AC switch control and feedback (Connector X301 and X303). See Section 8.2 for more details on the AC charging charging process.

> Pin 3 (HB1_OUT) is used as output to control the AC contactors of connector 1.

6.2.4 Connector X304

Connector X304 contains all signals for temperature sensors and digital outputs.

PIN	Signal Name
1	AIN_TEMP1
2	AIN_TEMP2
3	GND_AIN
4	HS_OUT

This connector is used for external temperature senors and digital outputs.

> Pin 1, AIN_TEMP1: Analog input for temperature sensor 1.
> Pin 2, AIN_TEMP2: Analog input for temperature sensor 2.
> Pin 3, GND_AIN: Ground for analog input temperature sensor 1 and 2.
> Pin 4, HS_OUT: High side switch output

In contrast to the vSECC, the vSECC.single +70°C includes supervision of two temperature sensors directly with the hardware supervision circuitry. Thus, the safety output will only be active if the temperature sensors connected to the AIN_TEMP_1 (X304.1) and AIN_TEMP_2 (X304.2) are in the specified range (see Section 12.4).

6.2.5 Connector X306

Connector X306 contains all signals for power supply connection and RS232 serial communication.

PIN	Signal Name
1	RS232_TXD
2	RS232_RXD
3	GND_SUPPLY
4	VCC_SUPPLY

> RS232 on X306.1 and X306.2, for connection of supported RFID-Readers (see section 8.10.4).
> The vSECC.single +70°C is typically powered with 12 V. Connect pin X306.4 with the supply voltage. Connect pin X306.3 with the corresponding ground. These connectors are used for the supply voltage of 12 V. The current drawn is typical below 400 mA, though while booting the vSECC.single +70°C can draw up to 500 mA. Please refer to Section 12.1 for further details on power consumption.

Caution: For EMC reasons, the maximum cable length for RS232 and RS485 should be 3 meters.

6.2.6 Connector X307

Connector X307 contains all signals for RS485 and CAN communication.

PIN	Signal Name
1	RS485_B
2	RS485_A
3	CAN_H
4	CAN_L

The vSECC.single +70°C offers the following serial communication interfaces:

> CAN on X307.3 and X307.4 with fixed termination, for communication with the power electronics.
> RS485 on X307.1 and X307.2, for connection of Modbus RTU slaves to the Modbus gateway (see section 8.20).

RS485 pin polarity: the noninverting pin is X307.2 (RS485_A) and the inverting pin is X307.1 (RS485_B).

Caution: For EMC reasons, the maximum cable length for RS485 and RS232 should be 3 meters.

6.2.7 Ethernet

This connector is used to connect network entities such as a Charging Station Management System (CSMS / Back End) or the Power Electronics Communication Controller (PECC) to the vSECC.single +70°C. This connector can also be used to configure the vSECC.single +70°C via the web configuration interface.

Caution: For EMC reasons, please use shielded Ethernet cables only.

6.2.8 USB

Caution: The USB connector is currently not in use.

6.3 Factory Reset Button

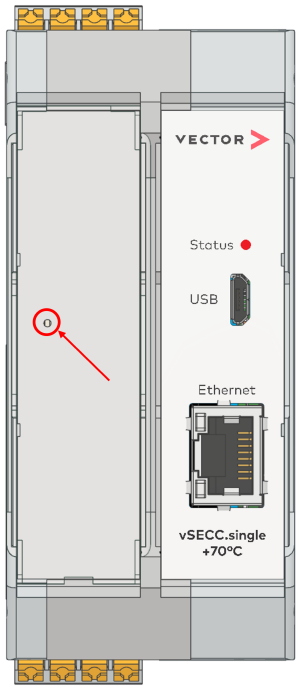


Figure 39: Factory Reset Button on vSECC.single +70°C

The factory reset button is located behind the small hole next to the USB port of the vSECC.single +70°C. This button is used to reset the configuration to the factory defaults. See Section 9.1 for details.

6.4 Wiring Examples

In this section, some wiring examples for the vSECC.single +70°C in common use cases are shown.

6.4.1 CCS Type 1 Charging

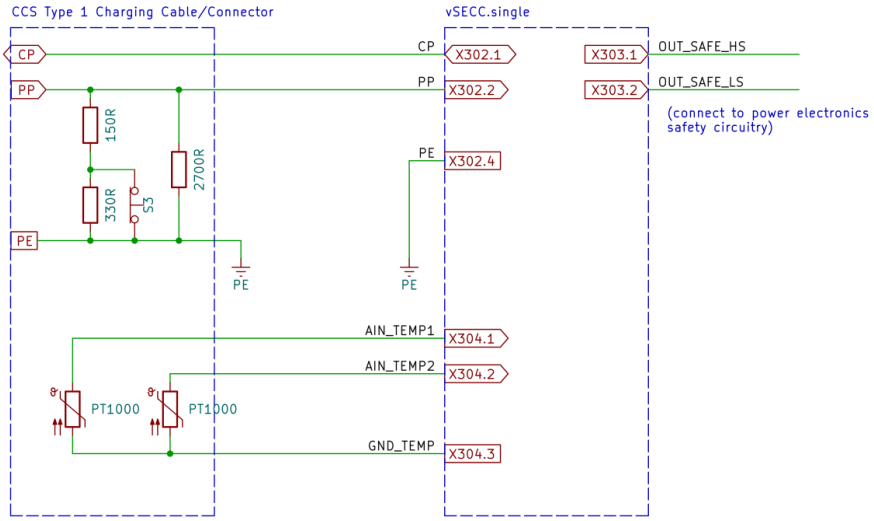


Figure 40: Wiring Diagram vSECC.single +70°C CCS Type 1

In order to use a CCS Type 1 charging connector, the following connections are necessary:

- CP: Connect X302.1 of the vSECC.single +70°C to the CP pin of the charging cable.
- PP: Connect X302.2 of the vSECC.single +70°C to the PP pin of the charging cable (usually, the resistors between PP and PE are included in the connector assembly).
- PE: Please make sure that the PE of the charging cable is connected to pin X302.4.
- Temperature sensors: Please make sure to connect the temperature sensors from connector assembly to X304.1 and X304.2 and use X304.3 for GND.

6.4.2 CCS Type 2 Charging

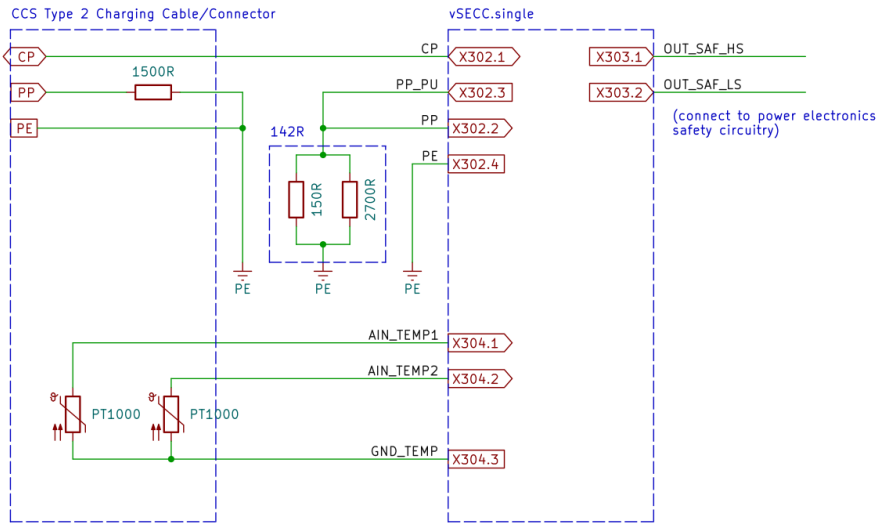


Figure 41: Wiring Diagram vSECC.single +70°C CCS Type 2

In order to use a CCS Type 2 charging connector, the following connections are necessary:

> CP: Connect X302.1 of the vSECC.single +70°C to the CP pin of the charging cable.

> PP: Check if the connector assembly has an internal resistor for PE. You may have to connect the resistor to PE manually or even provide the $1.5\,\mathrm{k}\Omega$ resistor on your base board. Also make sure to deactivate the PP supervision as shown in the example.

> PE: Please make sure that the PE of the charging cable is connected to pin X302.4.

> Temperature sensors: Please make sure to connect the temperature sensors from the connector assembly to X304.1 and X304.2 and use X304.3 for GND.

7 Configuration Guide

In this chapter you will find the following information:

The vSECC Controller can be configured through various mechanisms:

> in the provided web interface
> with a Python tool
> via a RESTful API (see chapter 7.9)
> by exchanging OCPP messages with a CSMS (see chapter 7.10.1).

Because connecting to a CSMS usually requires setting its address first, the initial configuration setup takes place using the web interface.

The web interface was completely reworked in Software Version 3.0. The configuration is migrated. Please note that the variable names changed. Please verify your configuration after the migration.

To connect to the web configuration interface, open a web browser and enter the vSECC Controller's IP address. Upon delivery, the IP address of the vSECC's right Ethernet port is set to 192.168.3.11. This will take you to the login page as shown in Figure 42. How to change the IP Address is described in chapter 7.3.4.

The credentials required for accessing the web interface for the first time consist of
username `"admin"`
password `"rootpassword"`.
On older firmware versions a browser login dialog appears instead of the vSECC login page, the username there is `"root"`.

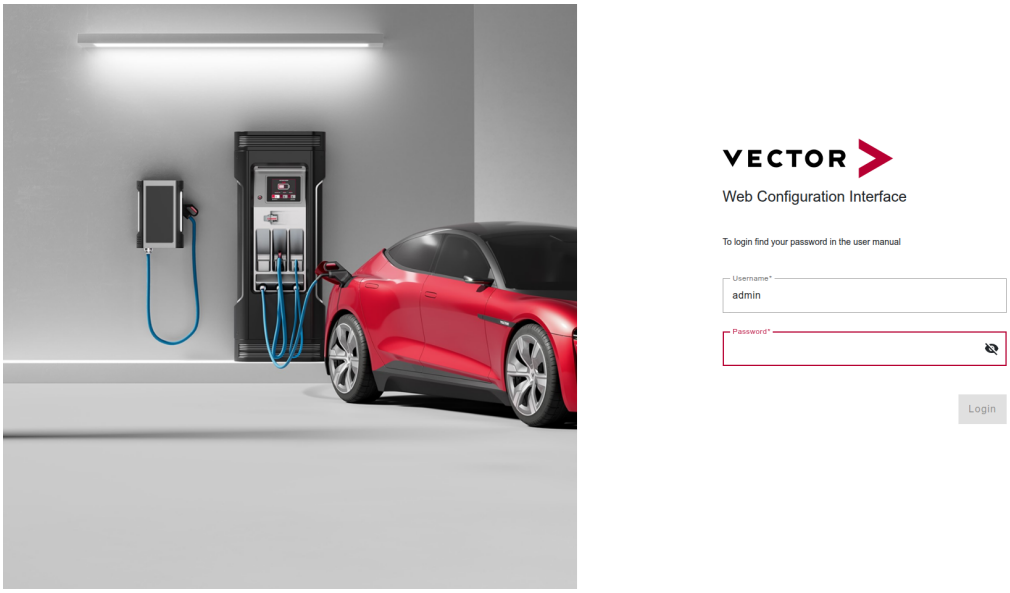


Figure 42: Login Page

After the first login, you are asked to set a new password.

The web interface is designed in a user-friendly manner. After changing the password, the **Dashboard** appears as shown in Figure 43. The vertical menu on the left guides you step-by-step through the **Configuration** of the vSECC Controller. Software containers are shown and can be managed in the **Container** part. To assist technical support, the vSECC Software's log files can be downloaded from the Controller within the **Logging** section. Moreover, **Certificates** for encrypted communication can be managed.
In the horizontal menu, the internal **Time and Date** of the vSECC Controller can be set by clicking on the time field. **Network Settings** can be accessed via the Network icon and **General Settings** via the gearwheel icon on the top right side.

7.1 Dashboard

The Dashboard provides an overview of the connected controller and status information. Please note that the displayed information depends on the installed licenses and configuration (e.g., one or two licensed connectors on the vSECC). Some boxes can be expanded by clicking the icon in their top right corner.

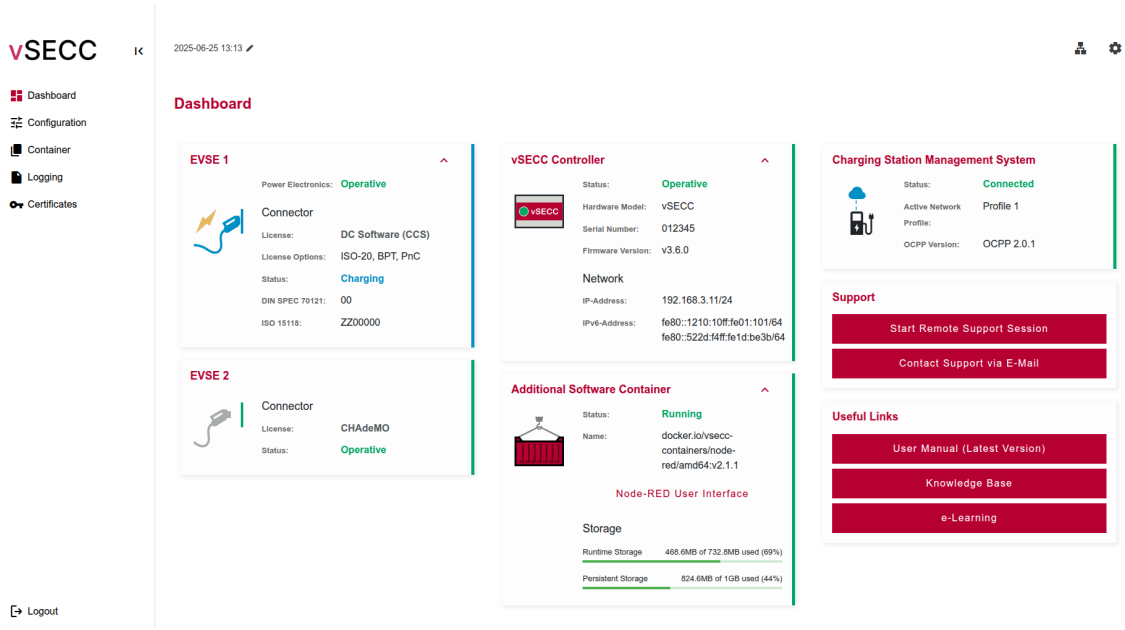


Figure 43: Status Dashboard

General information such as the installed Firmware Version and Serial Number is displayed in the vSECC Controller box in the middle.

Information about installed **Licenses** is shown in the individual EVSE boxes. How to upload new licenses is described in chapter 7.8.5.
System status information for different components is indicated by colored borders on the right-hand side of the boxes. This view helps monitor a charging process and provides a quick overview of the system configuration and health.
Furthermore, direct links to **Support** are available:

> via E-Mail, with important information automatically inserted.

> via Remote Support Session, by providing the Vector Support access to your vSECC Controller.

How to establish a remote support session is described in chapter 8.1.

Finally, **Useful Links** are provided, e.g., to the Vector Knowledge Base and the e-Learning Platform.

7.2 Time and Date Settings

The current local time and date of the vSECC Controller is shown in the horizontal menu bar (see Figure 44).

The vSECC Controller does not have a valid time in the delivery state. The local time must be updated first.

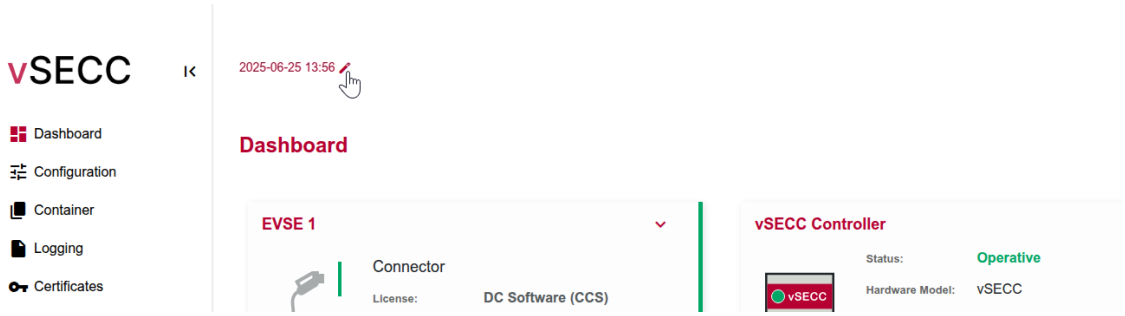


Figure 44: Time and Date Setting

The vSECC Controller's internal time can be synchronized with different mechanisms. Possible values for time synchronization are **OCPP Heartbeat, Network Time Protocol (NTP)** and **Local Host Time**.
The device time will be updated by clicking on **[Save and Synchronize Time]**. Then, the current time, date and time zone settings of the local host will be transferred to the vSECC Controller and synchronized with the internal real time clock.

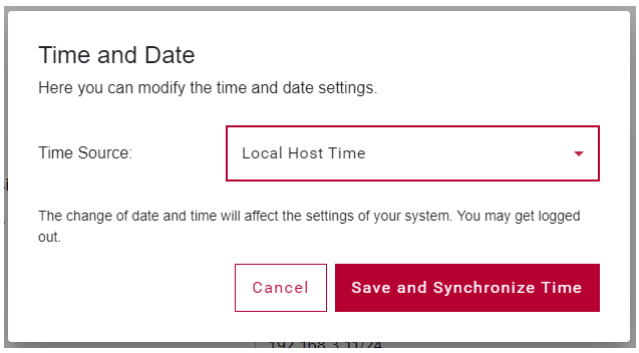


Figure 45: Setting the Time Source

Caution: After changing time and date settings, a reboot of the vSECC Controller is necessary. After a restart, the changes are successfully applied and active.

When setting the Time Source to **OCPP Heartbeat**, the vSECC Controller's internal time is set by the CSMS it is connected to. The local time settings may not be appropriate if the vSECC Controller is not connected to a CSMS. Also, it may be necessary to set the time in order to be able to use certificates for the connection to the CSMS and meet their validity time span.

When setting the Time Source to **Network Time Protocol (NTP)**, the vSECC Controller's local time will be synchronized with a network protocol server. A connection to this server is required for successful synchronization.

The internal real time clock ensuring accurate timekeeping even when the main power supply is lost. The real time clock is an independent timer. The battery backup activates when the main power supply is absent. The backup battery pin connects to an external super-capacitor. It powers the real time clock during power loss. This ensures that the real time clock remains functional and synchronized after boot up. See Section 10.10 (vSECC), Section 12.7 (vSECC.single) and Section 11.8 (vSECC.single Board) for details.

7.3 Configuration

The scrollable configuration page guides through the configurable parameters relevant for tailoring the vSECC Controller to the desired functionality of the charging station. All variables are described directly in the web interface, by hovering over a configuration variable. Only editable and relevant configuration parameters are displayed. I.E. if a license for Inverted Pantograph is active, fields for configuring OppCharge and SAE J3105 will appear. **Expert functions** provide functionality for in-depth configuration, e.g. to adjust timings.

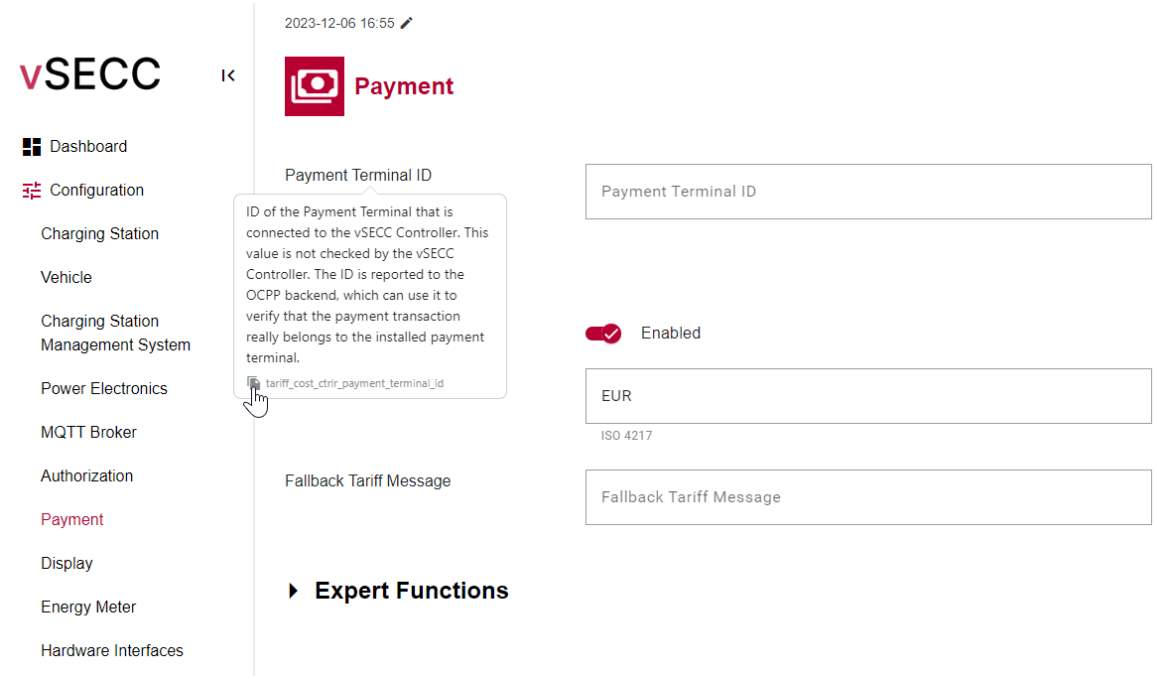


Figure 46: Hover provides explanation of the configuration variable and the technical variable name

When using the vSECC Controller for the first time, it is advisable to use the web interface for configuring. In series production, a provisioning script can be used that automatically runs a defined configuration. Hovering over the variable in the web interface provides the technical variable name (copyable), which is to be used for writing the provisioning script.

After configuring a vSECC Controller, the configuration can be downloaded as a yaml-file by pressing **[Download Config File]** and uploaded to other vSECC Controllers by pressing **[Upload Config File]**. The file includes the set variables of the ”Configuration”, ”Log Level”, ”Time and Date” and the Ethernet settings of the ”Hardware Interfaces”. Please note that the Firmware and Container Images are not part of the Config File. This enables an easy configuration of multiple vSECC Controllers. Furthermore, the REST API can be used for automatic configuration.
The configuration subpages are clustered by communication functionality - e.g. to the vehicle, the CSMS, peripheral devices such as the energy meter, etc.
Use the **[Save]** button in the lower right corner to save your changes to the configuration. Please note that the software may be automatically restarted to apply the configuration, if necessary. When changing the page without saving before, you are notified that there are unsaved changes. You can choose to stay on the page by pressing **[Cancel]**, to change the page and **[Discard Changes]** or **[Save and Apply]**.

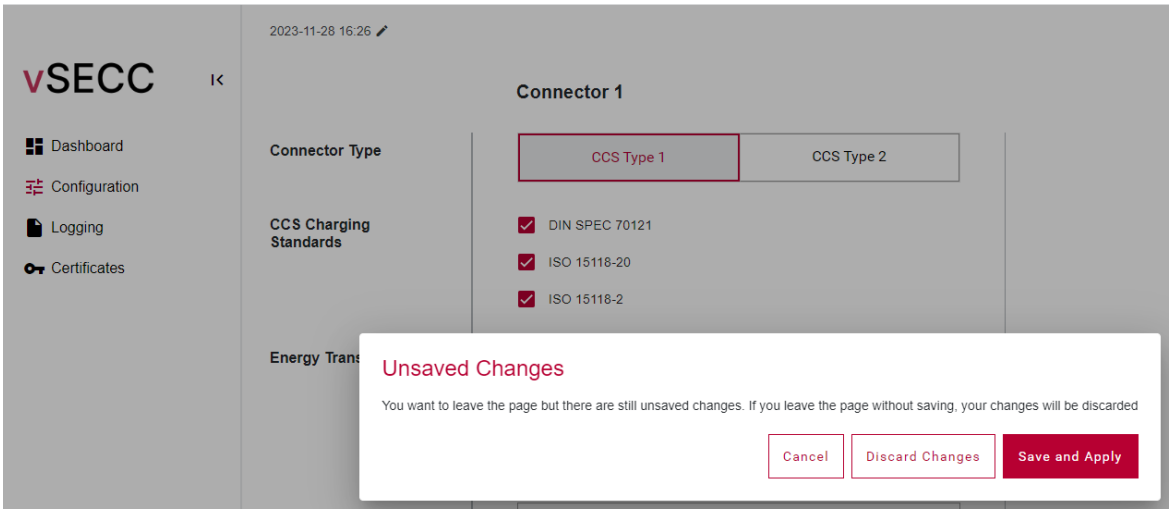


Figure 47: Dialog reminding of unsaved changes when changing a page

Caution: Restarting the vSECC Software will immediately disrupt all ongoing transactions, regardless of the application’s current state. Use this functionality with caution!

The respective sections in the User Guide describe their required configuration parameters to be set in the web interface.
In the following, some important settings to quickly get started with the first charging session are described.

7.3.1 Vehicle

The vSECC Controller comes pre-configured and can be adapted to the design of your charging station. Therefore, select the **Connector Type** and the **Charging Standards** to be used and provide an identifier for the **EVSE ID**.

The **EVSE ID** is validated according to the respective standard:

ISO-15118 Standard The EVSE ID must be at least 7 digits long and follow the format `<country code - 2 alpha digits> <optional separator - *> <operator code - 3 alphanumeric digits> <optional separator - *> <type - E or 0><charging station code - 1-30 alphanumeric digits or * (but not as leading character)>`. The fallback ID is `ZZ00000`.

DIN Standard The ID is validated as a numeric code up to 32 digits without leading zeros as specified in the DIN SPEC 91286. Furthermore, the vSECC Controller accepts the usage of `*` as separator. The fallback ID is `00`.

7.3.2 Charging Station Management System

Enabling CSMS Connection OCPP communication to a Charging Station Management System (CSMS) can be enabled by using the ”Connection” switch. As soon as it is enabled, various settings are exposed to configure the configuration details. It is important to note that only the communication to the CSMS is switched on or off. Even when the CSMS connection is switched off, the controller’s operation is still influenced by the concepts of OCPP. For instance authorization, if enabled, still affects how charging sessions can be authorized and the charging session still follows the OCPP concept of transactions.

The vSECC Controller supports the following OCPP versions which can be selected in the web interface:

- OCPP 1.6J
- OCPP 2.0.1

Disabling the connection via the WebUI overwrites the following configuration variables to provide an unrestrictive experience. It is possible to manually reconfigure these settings again, if a different behavior is desired.

- Authorization is disabled.
- AlignedDataCtrlr and SampledDataCtrlr are disabled. This means no measurands are sampled for transactions.
- The OCPP Version is defaulted to 2.0.1.
- The transaction start and stop point is set to `PowerPathClosed`.

When the CSMS connection is disabled via the provisioning tool, these defaults will not be applied automatically. Only values explicitly set in the provisioning file will be applied.

Please note that not all features of newer OCPP versions are available in older versions of the standard. You can find more information about the supported features in Section 1.4.16
.

The ”Multiple Connection Profiles” switch allows you to enable an advanced connection mechanism with ”Network Connection Profiles”, which is described in detail in a later section. It is recommended to disable this feature during initial setup.

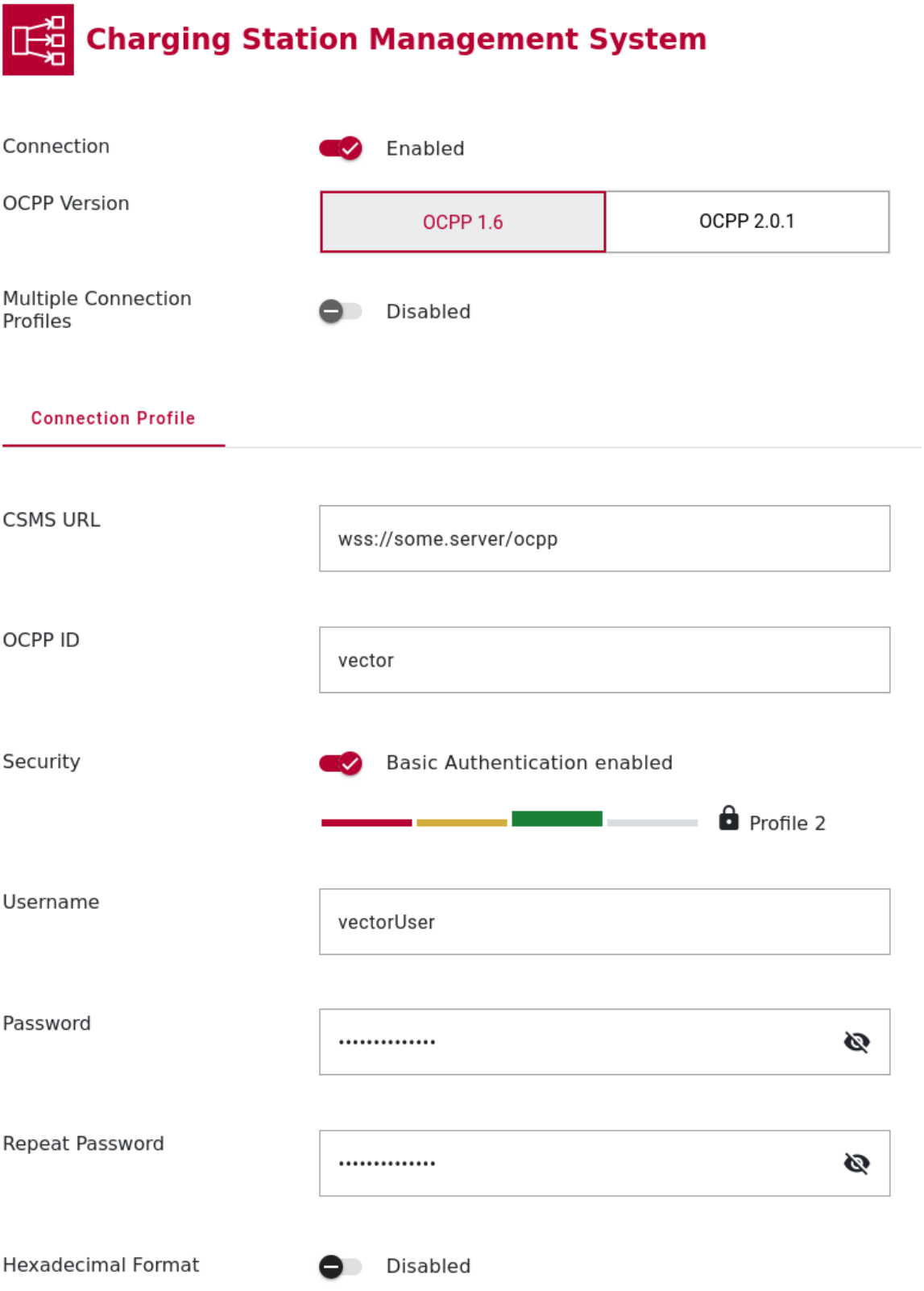


Figure 48: Configuration for using a Charging Station Management System

Connection Details Successful CSMS connections require at least a CSMS URL and OCPP ID.

The OCPP ID is appended to the CSMS URL and serves as a unique identifier of this charging station at the CSMS. For example:

> CSMS URL: ”wss://some.server/ocpp”
> OCPP ID: ”vector”
> Final URL: ”wss://some.server/ocpp/vector”

Security Profiles The vSECC Controller supports the OCPP security profiles 0, 1 and 2. Security profile 3 is not supported. The security profiles have the following definitions in accordance with the OCPP standard:

> 0: No HTTP Basic Authentication. No TLS.
> 1: HTTP Basic Authentication without TLS.
> 2: HTTP Basic Authentication with TLS.

The web interface displays the currently active security profile. It depends on whether Basic Authentication is enabled and whether a secure protocol (https/wss) is part of the configured CSMS URL.

Authentication to a CSMS The vSECC Controller may authenticate to a CSMS by using **HTTP Basic Authentication** credentials.

In order to use the HTTP Basic Authentication, a **Username** and the corresponding **Password** have to be entered. If OCPP 1.6 is used and the password is in hexadecimal, the corresponding field should be enabled. Moreover, a CSMS Root Certificate must be uploaded in the **Certificates** Section, as described in chapter 7.6.1.

VPN and APN The vSECC Controller will reject any network connection profiles sent by the CSMS that include VPN and APN settings, as this functionality is not currently supported.

Network Connection Profiles Network connection profiles are standardized by OCPP 2.0.1 use cases B09 and B10. They allow a CSMS to maintain a list of connection details that can be sorted by priority. This enables both migration and fallback behavior.

Network connection profiles are an advanced feature. They are typically managed by the CSMS and not required on the initial setup. Usually, there is no need to edit them through the web interface of the charging station. However, the vSECC Controller does provide this functionality to allow the user to adjust the settings of the profiles if needed.

Network connection profiles are stored in slots. The vSECC Controller provides three configurable slots and one read-only fallback slot. The writeable slots can be configured using either the web interface or the CSMS. Configuration via the web interface is explained in the following section. Configuration through the CSMS is done with the SetNetworkProfileRequest in OCPP 2.0.1.

Configuring Network Connection Profiles The ”Multiple Connection Profiles” switch must be enabled in order to unlock the functionality of using multiple profile slots.

Several tabs become visible upon activation. Each tab represents a single profile. Profiles 1, 2, and 3 are user-configurable. The fallback slot is read-only. It represents the last successful connection. As the name implies, it serves as a fallback if a connection cannot be established to any of the other profiles. The fallback slot may be hidden from the tabs if your vSECC Controller has not previously connected to any CSMS.

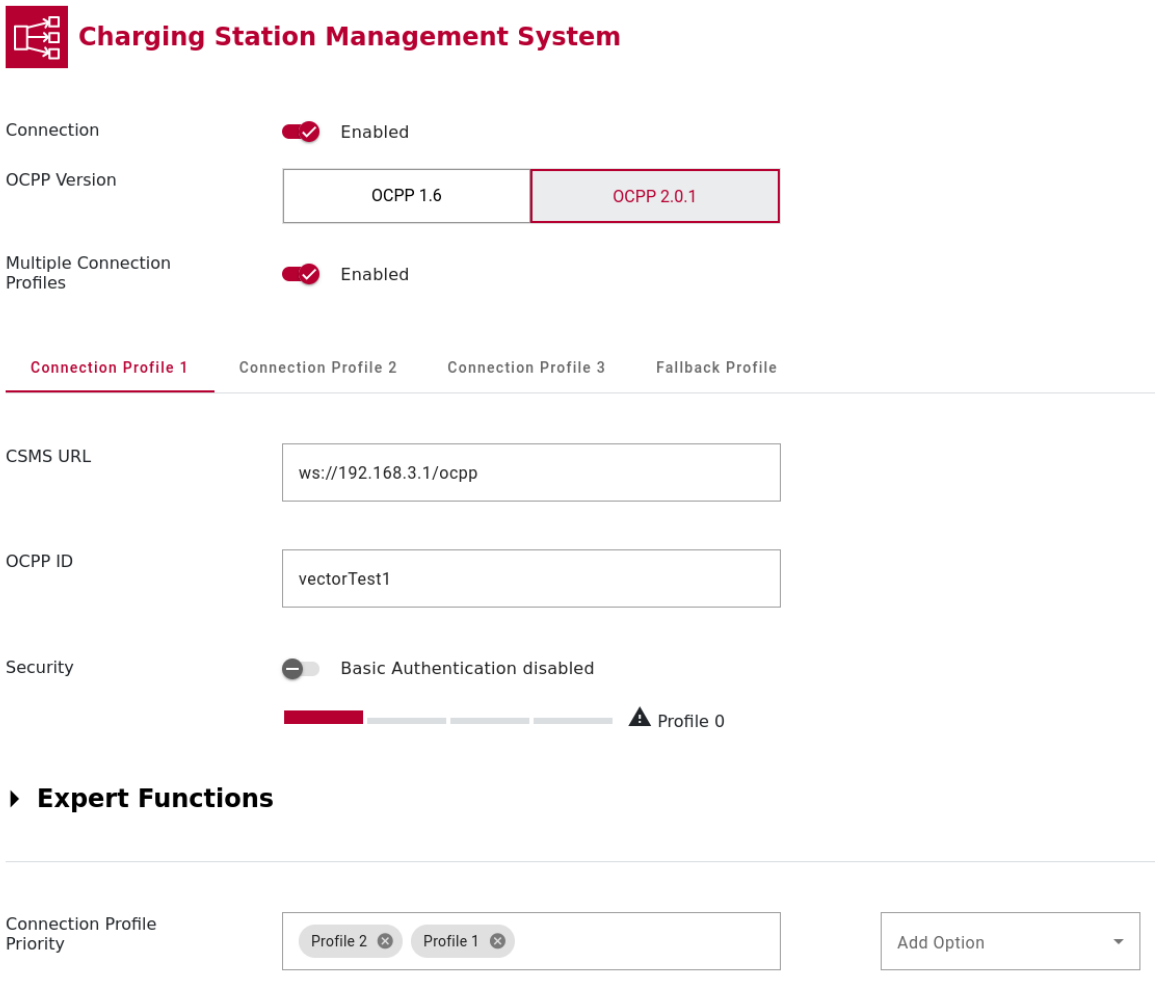


Figure 49: Multiple Network Connection Profile Tabs

Profiles can be ordered using the Connection Profile Priority field. When the vSECC Controller attempts to connect to a CSMS, profiles to the left will be given priority. The priority mechanism is described in more detail in the next section. Profiles can also be removed

from the priority list using the ”X”, which will prevent them from being used in future connection attempts. Profiles can also be (re-)added using the ”Add Option” drop-down menu. For a profile to be added, it must contain at least a CSMS URL and an OCPP ID. Please be advised that profiles removed from the priority list are not deleted. Rather, they remain intact but become unused.

Profile Iteration Logic The following sections provide a detailed overview of how the vSECC Controller uses network connection profiles. This knowledge is not necessary for initial setup and may not even become relevant during operation. However, it can be useful when trying to understand how exactly the vSECC Controller handles certain edge cases.

Profile Slots

Each network connection profile is stored in a ”slot”. These slots appear as instances of the ”NetworkConfiguration” component within the OCPP device model. Each slot contains a set of variables which describes the connection details of the stored profile. Slots are identified by their instance name, which must be parsable as an integer (e.g. ”1”).

Fallback Slot

The fallback slot is identified by its instance name which is set to ”-1”. The fallback slot always contains information about the latest successful connection. In case you are connected to a CSMS, it therefore also represents the currently active connection. The contents of the fallback slot are maintained automatically by the vSECC Controller and should not be altered by another party. As the name implies, it serves as a fallback if no connection can be established with any of the other slots.

Please be advised that the information stored in the fallback slot will remain valid until a new, successful connection is established. You may choose to clear the contents of the fallback slot by altering the device model directly. For example, before provisioning the charging station, you may wish to clear any remaining maintenance or provisioning CSMS information.

Connecting to a CSMS

The required variable ”NetworkConfigurationPriority” underneath the ”OCPPCommCtrlr” in the device model plays a crucial role in the slot iteration logic. It represents an ordered list of the network connection profile slots that the vSECC Controller shall attempt to connect to. In our example, a total of three customizable slots exist, which are named ”1”, ”2” and ”3” respectively. Additionally, the mandatory fallback slot ”-1” exists. The NetworkConfigurationPriority is set to [”1”, ”3”, ”2”] which means that slot ”1” has the highest priority and slot ”2” has the lowest.

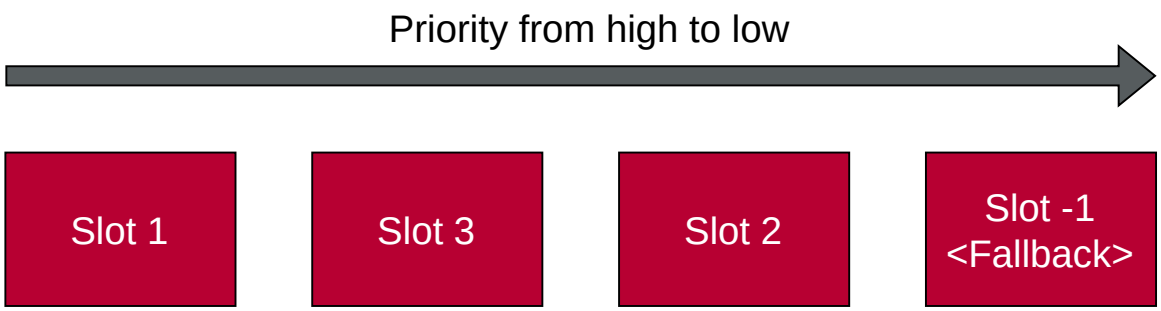


Figure 50: Network Connection Profile Priority

After initialization, the vSECC Controller starts with the slot of the highest priority (”1”) and executes these steps:

- Read the network connection profile’s contents from the slot.
- Check if the profile is valid:
  - OcppCsmsUrl must not be empty.
  - OcppVersion must be set to OCPP16 or OCPP20.
  - OcppTransport must be set to JSON.
  - ”SecurityProfile” of the slot must not be lower than the global ”SecurityProfile” underneath the ”SecurityCtrlr”.
  - For slots with security profile 2: Required root certificates must exist.
- If the profile is invalid: Transition to the next slot in the priority list. Otherwise: continue.
- Update global variables:
  - Set the ”ActiveNetworkProfile” variable underneath the ”OCPPCommCtrlr” to the slot’s identifier (e.g. ”1”).
  - Set the ”Identity” and ”BasicAuthPassword” variables underneath the ”SecurityCtrlr” to the slot’s ”Identity” and ”BasicAuthPassword”.
  - Set the ”SecurityProfile” variable underneath the ”SecurityCtrlr” to the slot’s ”SecurityProfile”.
  - Set the ”DefaultMessageTimeout” variable underneath the ”OCPPCommCtrlr” to the slot’s ”MessageTimeout”.
- Pass the profile’s information to the WebSocket to start a connection attempt.

If the connection fails:

- Check if ”NetworkProfileConnectionAttempts” have been exceeded. If yes, proceed to the next slot in the priority list (more details below). If not, remain on this slot. The slot is NOT read again from the device model because its contents cannot be altered by a CSMS while there is no active connection.
- Pass the profile’s information to the WebSocket again to start the next connection attempt.

If the connection succeeds:

> Store the profile in the fallback slot as it now represents the latest successful connection.

> Reset the connection attempts.

Slot Iteration

If a slot is invalid or has reached its maximum amount of connection attempts, the vSECC Controller moves towards the next slot in the priority list. In this example, slot ”1” has reached its maximum attempts. Therefore, slot ”3” is now used for the following connection attempts.

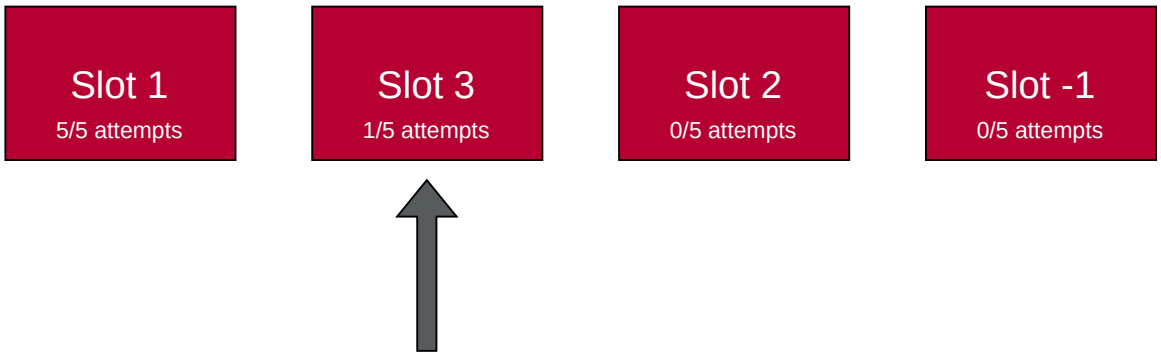


Figure 51: Active Slot 2

The vSECC Controller will continue to iterate through all available slot until it reaches the end of the prioritized list. If no connection can be established with the lowest priority slot as well, the vSECC Controller will use the fallback (”-1”) slot which contains the information of the latest successful connection attempt.

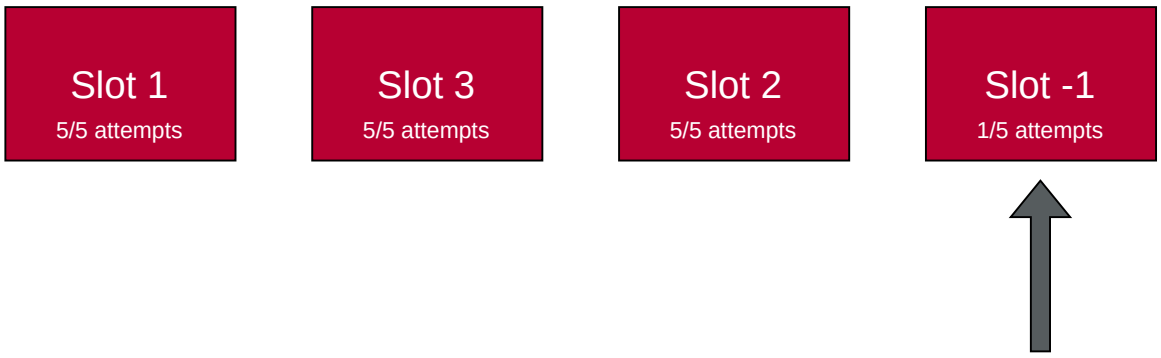


Figure 52: Active Fallback Slot

If the fallback slot also fails to establish a connection and reaches its maximum attempts, the vSECC Controller starts its iteration logic at the highest-priority slot again.

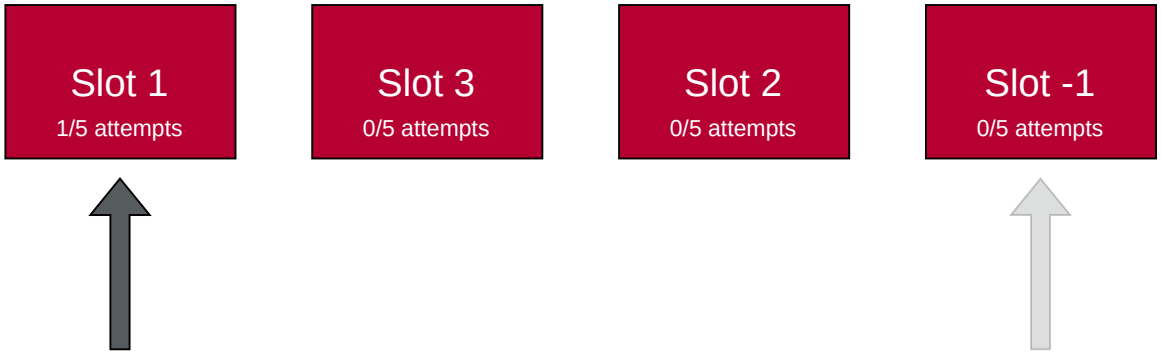


Figure 53: Iteration starts at the front after fallback attempts have expired

Should the NetworkConfigurationPriority be altered, the vSECC Controller will apply the updated priority on the next connection attempt. This may be caused by a restart of the controller or a disconnect from the currently active CSMS connection. Applying the new priority means that the vSECC Controller will restart its iteration loop at the slot with the now highest priority and reset its connection attempts.

Global Variables

As already mentioned in previous sections, some of the active slot's contents are mirrored to a set of global variables. This is the proposed way of ensuring backwards compatibility in OCPP 2.1.

- SecurityCtrlr/Identity
  - If read: Returns the identity (basic auth username) of the active slot.
  - If written: Overwrites each slot's identity with the given value.
- SecurityCtrlr/BasicAuthPassword
  - If read: Returns the basic auth password of the active slot.
  - If written: Overwrites each slot's basic auth password with the given value.
- SecurityCtrlr/SecurityProfile
  - If read: Returns the security profile of the active slot.
  - If written: No action. Recommended mutability is read-only.
- OcppCommCtrlr/DefaultMessageTimeout
  - If read: Returns the message timeout of the active slot.
  - If written: No action. Recommended mutability is read-only.

Active Network Profile

The ActiveNetworkProfile variable underneath the OcppCommCtrlr signals which slot is currently active:

- 0 -> No valid profile exists
- -1 -> Fallback slot is active
- <any other integer> -> Slot is active

Should the active slot (e.g. ”2”), which is currently connected to the CSMS, be modified, the ActiveNetworkProfile will be set to the fallback slot (”-1”). The rationale behind this approach is to ensure that if the CSMS is configured to read the ActiveNetworkProfile variables (i.e., ”-1”) and access the slot with the same instance name (also ”-1”), it will retrieve the connection details that are currently in use and will function as expected. In contrast, if the ActiveNetworkProfile were to remain at slot ”2,” the CSMS would read the new, altered slot, which has not yet been applied.

Reconnection

In the event that the vSECC Controller loses connection with a CSMS, it will attempt to recover the connection by restarting from the slot with the highest priority. For example, if the vSECC Controller is connected to slot ”3” with the 2nd highest priority and it then loses connection, it will start its first reconnection attempt with slot ”1” which has the highest priority.

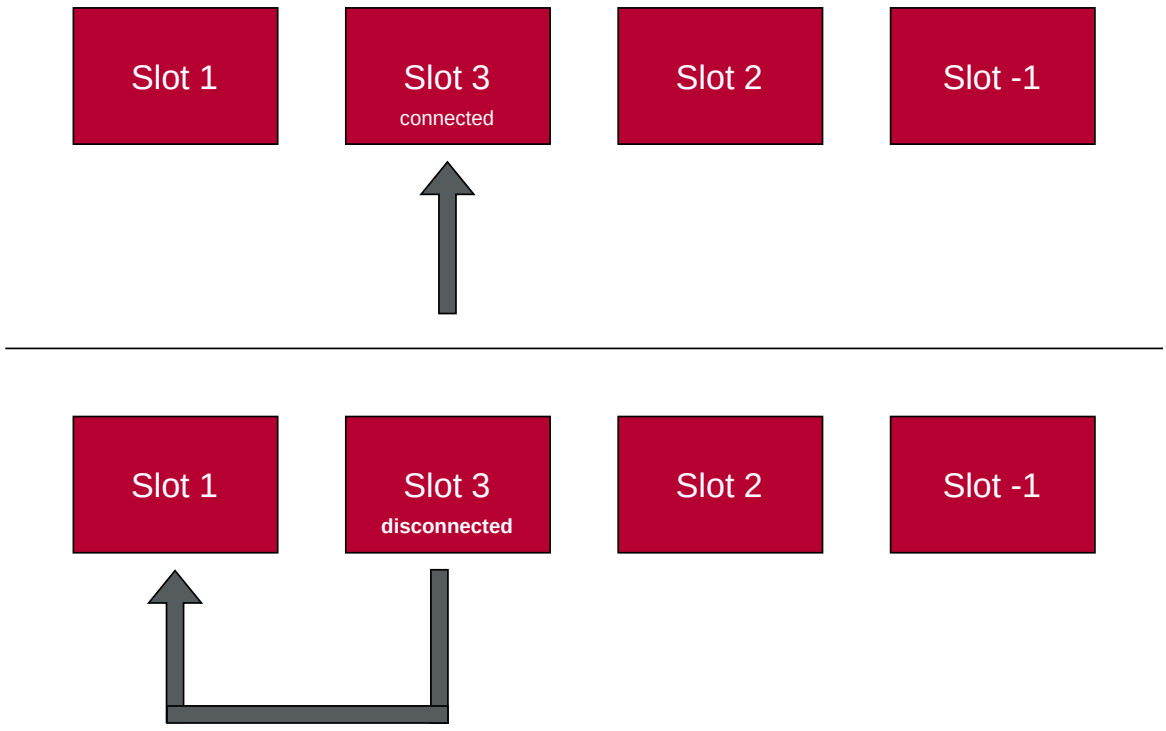


Figure 54: Disconnect Iteration

Installation

The CSMS can install new network connection profiles in OCPP 2.0.1 by using the SetNetworkProfileRequest. A set of validations is performed every time the CSMS attempts to set a new profile:

- OcppCsmsUrl must not be empty.
- OcppVersion must be set to OCPP16 or OCPP20.
- OcppTransport must be set to JSON.
- ”SecurityProfile” of the slot must not be lower than the global ”SecurityProfile” underneath the ”SecurityCtrlr”.
- Security profile must not be set to 3.
- For slots with security profile 2: Required certificates must exist.
- APN and VPN information must not be part of the profile.

If any of the validations fail, the profile gets rejected.

Please note that even if the CSMS chooses to overwrite the profile in the currently active slot, the changes will not be applied until the next (re-)connection attempt.

Security Profile

According to OCPP 2.0.1’s use case A05 it shall not be possible to use a network profile with a security level lower than the one of the previously used profile. Therefore, setting the NetworkConfigurationPriority to a slot containing a lower security profile with a SetVariablesRequest will be rejected. Also, installing a profile with the SetNetworkProfileRequest with a lower security level will be rejected. Users may ”downgrade” to a lower security level by setting the SecurityCtrlr’s SecurityProfile variable directly.

If a profile with a higher security level gets installed, all profiles containing a lower security level are removed from the NetworkConfigurationPriority and prevented from being set again via SetNetworkProfileRequest and SetVariables on NetworkConfigurationPriority. The slot itself remains untouched and its contents do not get cleared.

Ignored fields

The following fields are ignored when the vSECC Controller applies a network connection profile:

- ocppVersion
- ocppTransport

SetVariables Access

Due to the nature of the profile's content being stored in the device model, this data is also accessible via the SetVariablesRequest. However, the validation logic and proper internal handling is only applied when the CSMS uses the SetNetworkProfileRequest as specified in the OCPP 2.0.1 standard. This limitation only applies to the slot's contents (i.e. everything underneath the NetworkConfiguration component) and not to variables that affect all profiles such as the NetworkConfigurationPriority.

OCPP 1.6

OCPP 1.6 is not equipped with the functionality to handle network connection profiles. It does not provide a uniform method for establishing the URL that a SECC should connect to. When connected to an OCPP 1.6 CSMS, the same connection profile mechanisms are applied as if connected to an OCPP 2.0.1 CSMS. The OCPP 1.6 CSMS may choose to alter the profiles' values within the device model. Please note though that the ChangeConfiguration.Req will not be rejected even if it puts the slot into an invalid state. This validation logic is only applied when using network connection profiles in OCPP 2.0.1 with the SetNetworkProfileRequest.

7.3.3 Power Electronics

There are various ways how to communicate with power electronics. To understand how to interface to power electronics, please refer to chapter 8.18.

7.3.4 Hardware Interfaces

The Hardware Interfaces section allows to change parameters for CAN and serial communication. The network configuration (IPv4 Settings) has its own **Network Settings** page that can be accessed by the network icon on the top right menu.

7.4 Container Management

The vSECC Controllers allow runnig customer-specific software in a containerized environment. Upon first delivery, no container is pre-installed. How to work with the container is described in the User Guide chapter 8.27. To manage the container and the used data, open the **Container** section in the navigation bar.

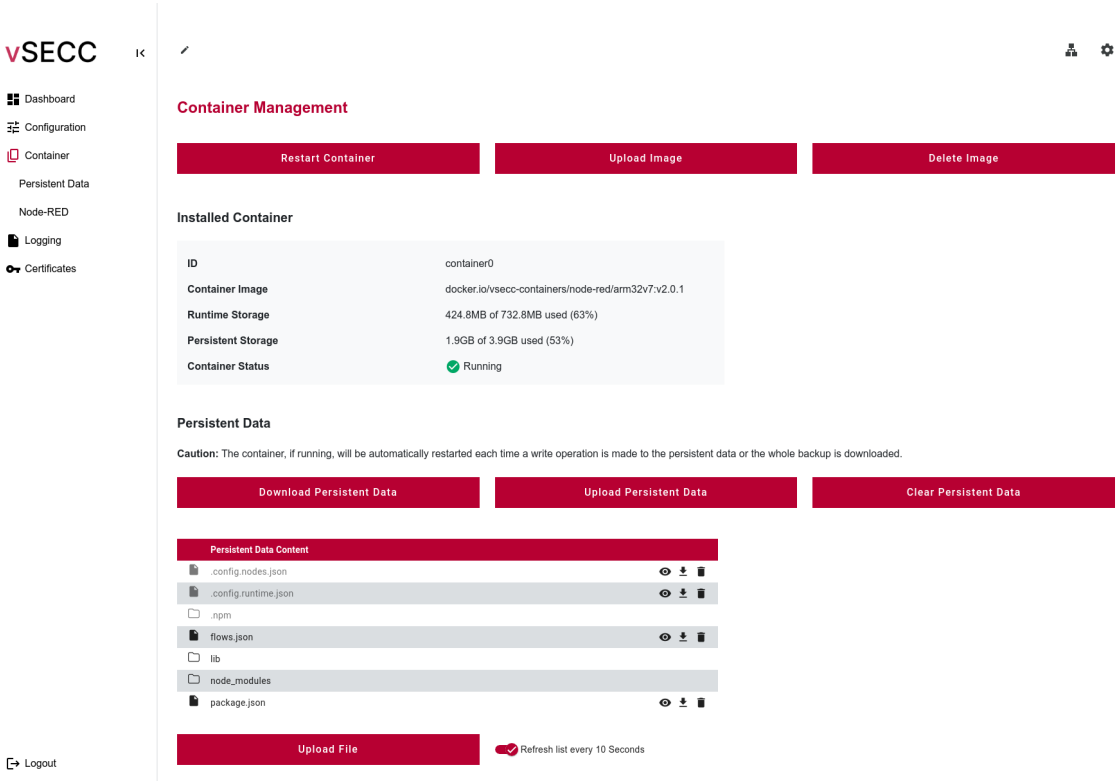


Figure 55: Container Management Section

The web interface allows to install, remove or restart a container. Additionally, the persistent storage can be modified by either uploading or deleting single files, uploading a backup or clearing all persistent data.

7.4.1 Container Handling

The **Container Management** section provides the following options to handle the actual container:

- Container installation
- Restarting the installed container
- Removing the container alongside with the persistent data

Container Installation The installation of a container is done by pressing **[Upload Image]**. It follows the same steps like performing a firmware update. The vSECC Controllers only allow to install signed containers in the ”vSECC Container Image Bundle” (`*.vCIB`) format. After downloading the `*.vCIB` file from the Vector Portal, select the Image and press **[Upload Image]**.

Caution: A valid date is required for container installation, otherwise the installation will fail. The current local time and date of the vSECC Controller is shown in the horizontal menu bar. For time and date settings refer to chapter 7.2.

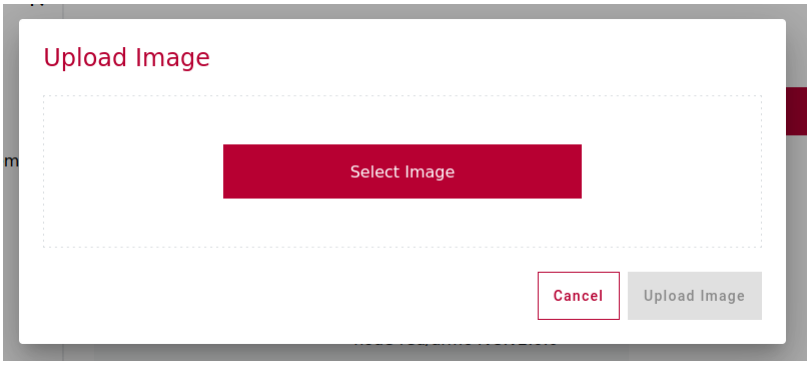


Figure 56: Container Image Upload

Caution: Depending on the file size of the container image, the installation may take several minutes. Please make sure not to interrupt the connection or reload the page until the successfull installation is reported.

The installed container with the current status and the storage use is displayed in the **Installed Container** section.

Installed Container

ID	container0
Container Image	docker.io/vsecc-containers/node-red/arm64v8:v2.0.0
Runtime Storage	450.8MB of 732.8MB used (67%)
Persistent Storage	742.8MB of 1.9GB used (41%)
Container Status	Running

Figure 57: Installed Container Section

Restart Container Pressing **[Restart Container]** will clear the container environment to the state when the container was installed. This can be used to clear data stored by the container inside the `rootfs`.

Remove Container Pressing **[Delete Image]** will remove the installed container and delete all data stored by the container in the `/data` folder.

7.4.2 Persistent Data Handling

A persistent data location is provided for the container. This allows to store data that persist through reboots and updates of the vSECC Controllers or of the container itself.

Persistent Data Content The content of the persistent data storage is shown in the **Persistent Data Content** list. The list only shows the first level of files and folders and is intended to provide an overview. It is not possible to create, delete or navigate through (sub-) folders. However, it is possible to download or delete all files shown in the file list. Files considered by the vSECC Controller as text files can be viewed using the **Eye** button next to the file name. The list is refreshed automatically when the slider next to the **[Upload File]** button is activated, to reflect possible changes done by the container during runtime.

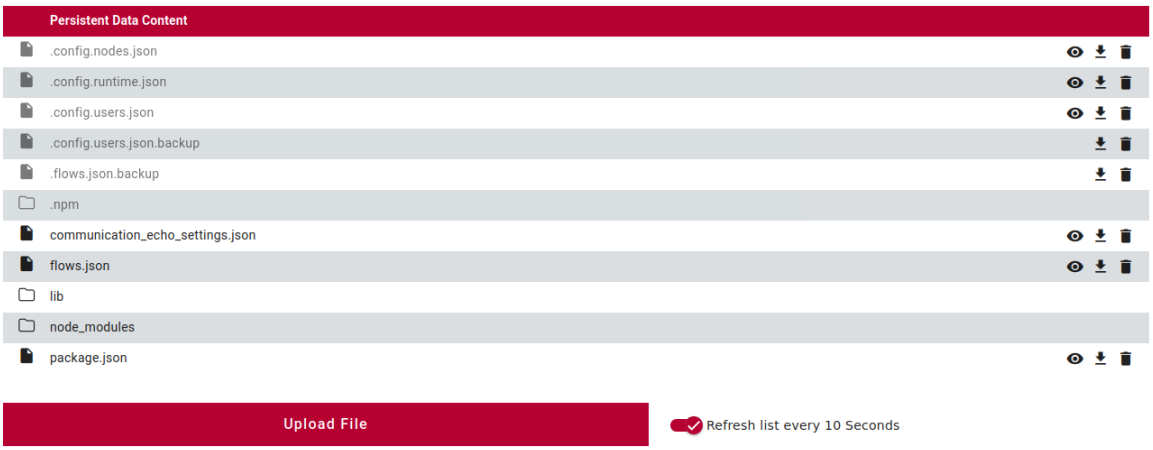


Figure 58: Container Image Upload

Upload a Single File Pressing **[Upload File]** opens a pop-up to uplaod a single file, which may contain e.g. settings for the container, to the persistent data storage. Press **[Select File]**, select a file to be uploaded in the file picker and press **[Upload File]**.

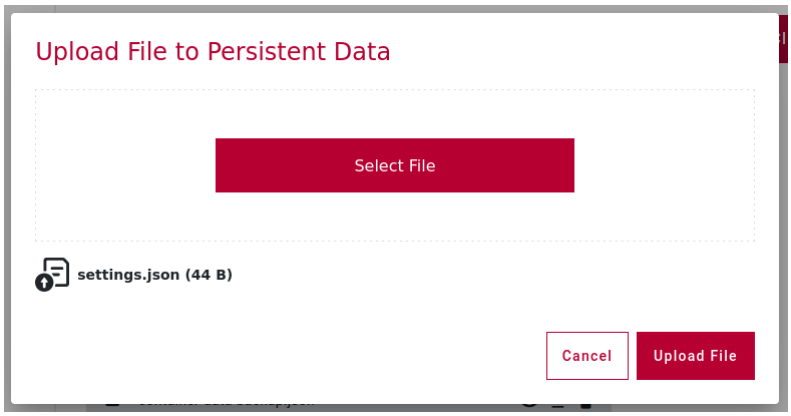


Figure 59: Single File Upload

Download Persistent Storage Backup It is possible to download a backup of the entire persistent data storage by pressing **[Download Persistent Data]**.

Caution: In order to avoid changes to the persistent data storage during creation of the backup, the container is stopped before and started after creation of the backup.

Upload Persistent Storage Backup A backup can be installed by pressing **[Upload Persistent Data]** and selecting the backup to be installed. When installing a backup, the container is stopped before and restarted again after installation.

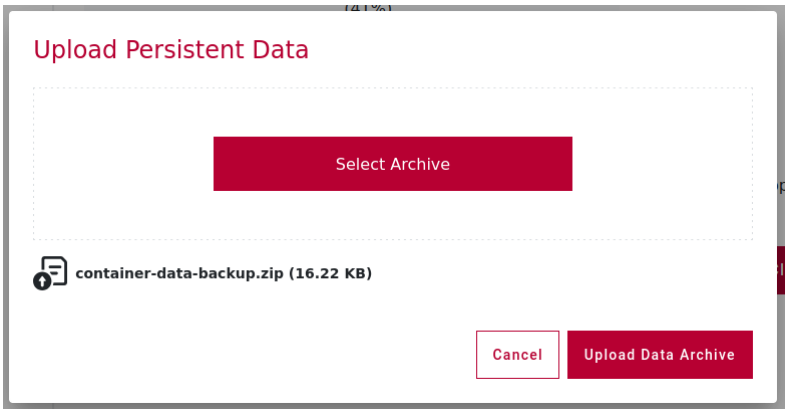


Figure 60: Upload Persistent Data Pop-Up

Clear Persistent Storage Pressing **[Clear Persistent Data]** clears all the data stored in the `/data` folder by the container.

Caution: The container will be restarted (without clearing the `rootfs`) when the `/data` folder is cleared.

After the container has been restarted, the `/data` folder may be populated with files by the running application again. If using Node-RED, this will always be the case.

When using the Configurable Customer Interface container, the above mentioned mechanism should only be used to revert to a factory preset state (without doing a factory reset for the whole vSECC Controller).

7.4.3 Accessing the Container

After logging into the web interface, the container web front-end (if available) can be accessed under the URL `http(s)://<ip of vsecc>/container`. If you are not yet logged in, you will be automatically redirected to the login page.

If you are developing your own container using the vSE Developer Program, the web front-end of your container needs to be served on port 1880 to use the `/container` route.

Access to Node-RED GUI If a Node-RED container is installed, a button is shown in the navigation bar that will redirect to the Node-RED user interface.

The GUI can also be accessed under:
`<IP-Address of vSECC Controller>/node-red/`
e.g. `http://192.168.3.11/node-red/`.
Note: If tunneling the connection through a reverse proxy, the trailing slash in the URL must not be omitted.

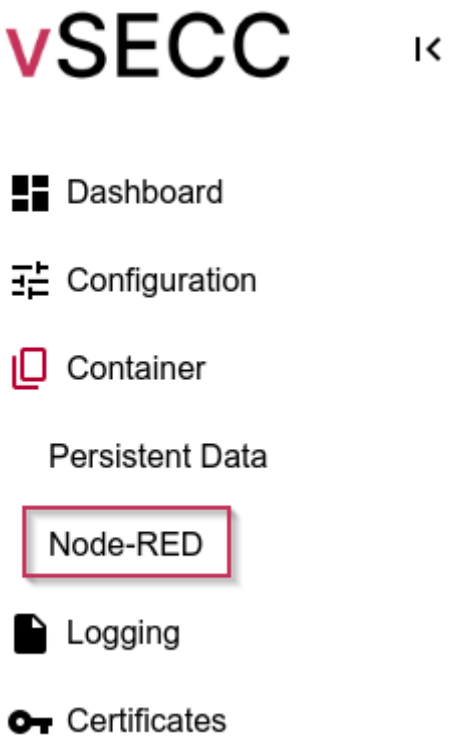


Figure 61: Shortcut to Node-RED GUI

In addition, the following URLs are accessable in the Node-RED container:

> `http(s)://<ip of vsecc>/container`
  Route to the Node-RED admin interface (same as `/node-red/`)

> `http(s)://<ip of vsecc>/node-red/ui`
  Shows the Node-RED Dashboard (no authentication needed)

> `http(s)://<ip of vsecc>/node-red/ws`
  For websocket connections (no authentication needed)

7.5 Logging

To assist technical support, the vSECC Software provides comprehensive logging.
The log level can be set in the **Settings** section. Based on the selected value, the associated messages and messages with higher severity are logged. Consequently, for a smaller log file, **Warn** or **Error** should be used.
To assist analysis of the EV communication, trace logs of the high level communication with the EV can be activated in the **HLC Logging** section. Further details are found in chapter 7.5.1.
Log files can be displayed, downloaded and deleted from the vSECC Controller in the **Log File Management** section. Alternatively, they can be downloaded via OCPP, which is described in chapter 7.10.1.

The log files are stored for 1 week on the vSECC Controller and then deleted automatically.

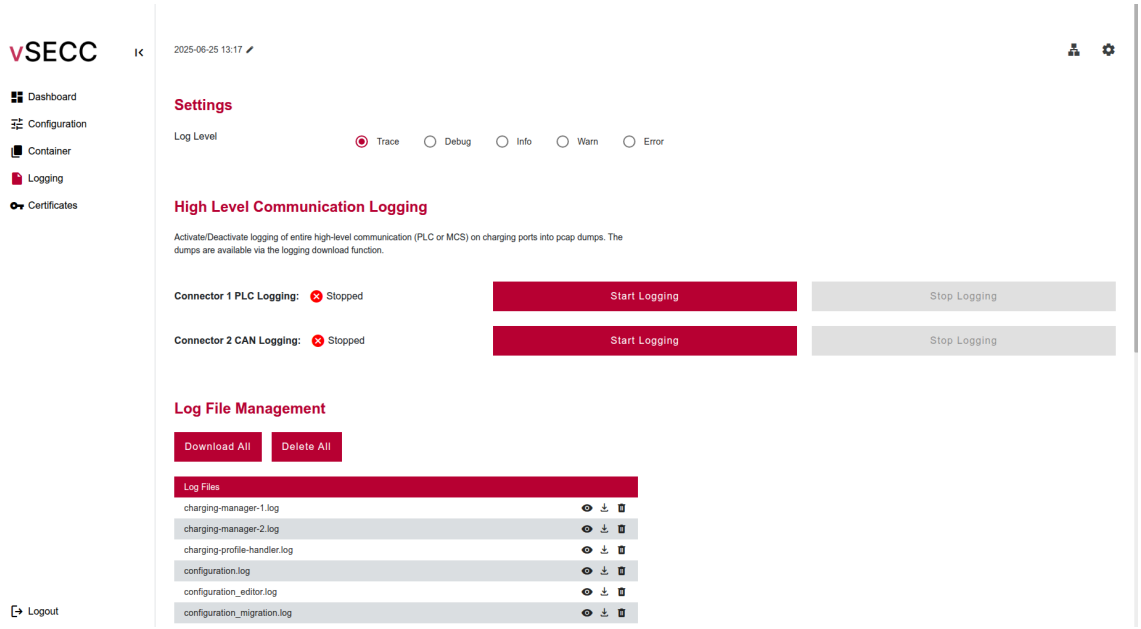


Figure 62: Log File Management

When viewing a log file, an Autoscroll mode is activated to monitor the currently running charging process (see Figure 63). This can be deactivated .
To download log files using the web interface, press the **[Download All]** button. Single log files can be downloaded by clicking on the respective download icon.
To delete log files, use the **[Delete All]** button. Single log files can be deleted by clicking on the respective delete icon.

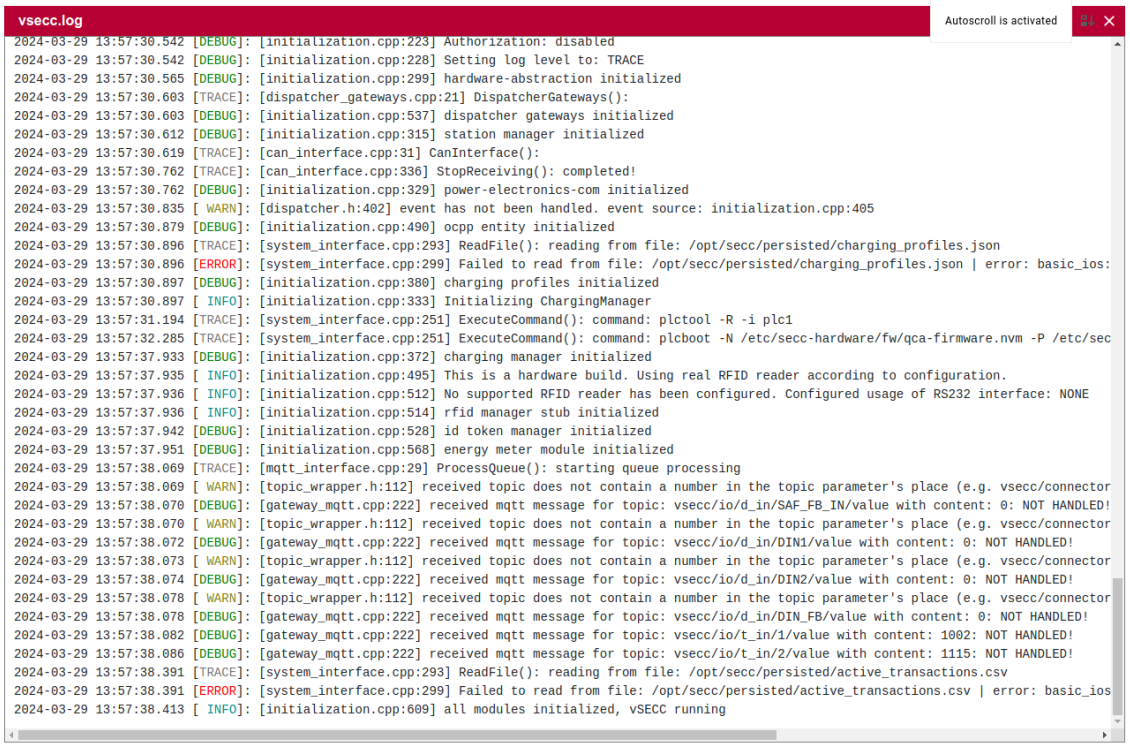


Figure 63: Autoscroll Mode when Viewing a Log File

7.5.1 HLC Logging

To assist analysis of the EV communication, the vSECC Software allows to write trace logs of the high level communication (HLC) between charging station and EV. The communication is written into pcap trace files on the vSECC Controller. Later, the files can be downloaded via the web interface and opened with tools such as CANoe or Wireshark to analyze the complete high level traffic.

The supported physical interfaces are based on the licensed connectors. Logging of unlicensed connectors is not supported.

The following example shows a MCS (Megawatt Charging System) license on connector 1 and a PLC (Power Line Communication) license on connector 2.

To start the trace logging of the PLC high level communication using the web interface, press the **[Start Logging]** next to the desired connector. The status indication changes from **[Stopped]** to **[Running]**, as shown in Figure 64.
After the activation of the trace logging, initiate a charging session.
To deactivate the trace logging, press the **[Stop Logging]** button.

To download the trace files use the download log files feature as described in 7.5. A new .pcap file should appear in the log files list. Depending on the used physical interface, the file is prefixed with **mcs**, **plc**, **can** or **eth** and the connector number. Rotation of the log files will suffix the pcap files with a index number starting at 0.

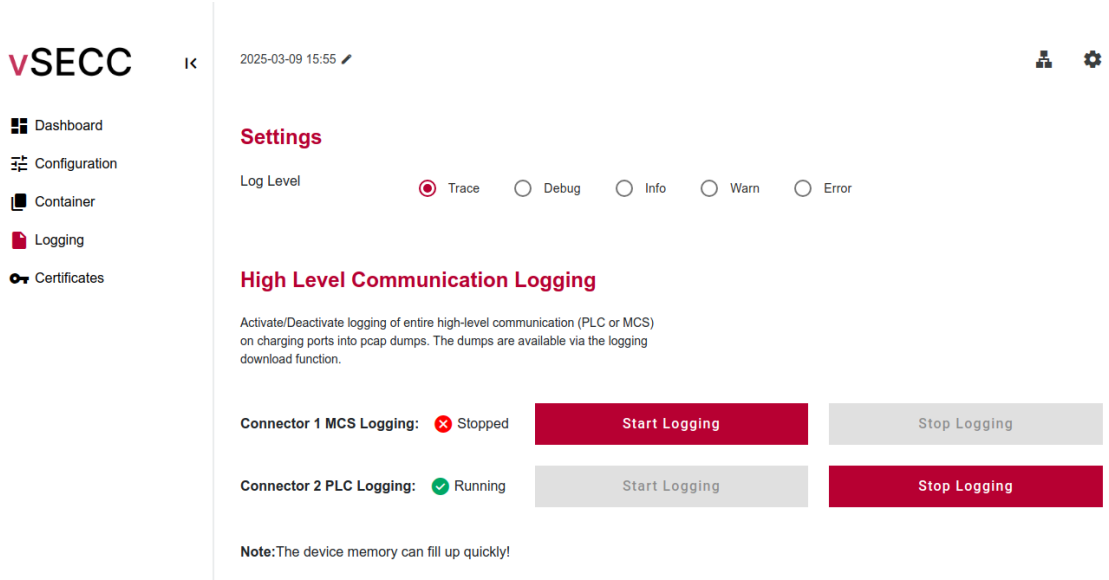


Figure 64: Starting of HLC Logging for PLC at Connector 2

Caution: Use this feature only for development purposes or to analyze issues. As long as the HLC logging is activated, the complete HLC traffic is written to the vSECC Controller's memory and could fill up the entire available space for log files. Therefore, the log files are rotated after running for about 2 days (20 MB size). Nevertheless, the activated feature reduces the lifetime of the device's memory. Always turn off the logging if you do not need it. To reduce the used space, clean up the log files after you have downloaded them.

7.6 Certificates

This section provides certificate management functionality, e.g. for CSMS Root CA Certificates, EVSE Leaf Certificates or OEM Root Certificates. The installed certificates are shown as depicted in Figure 65.

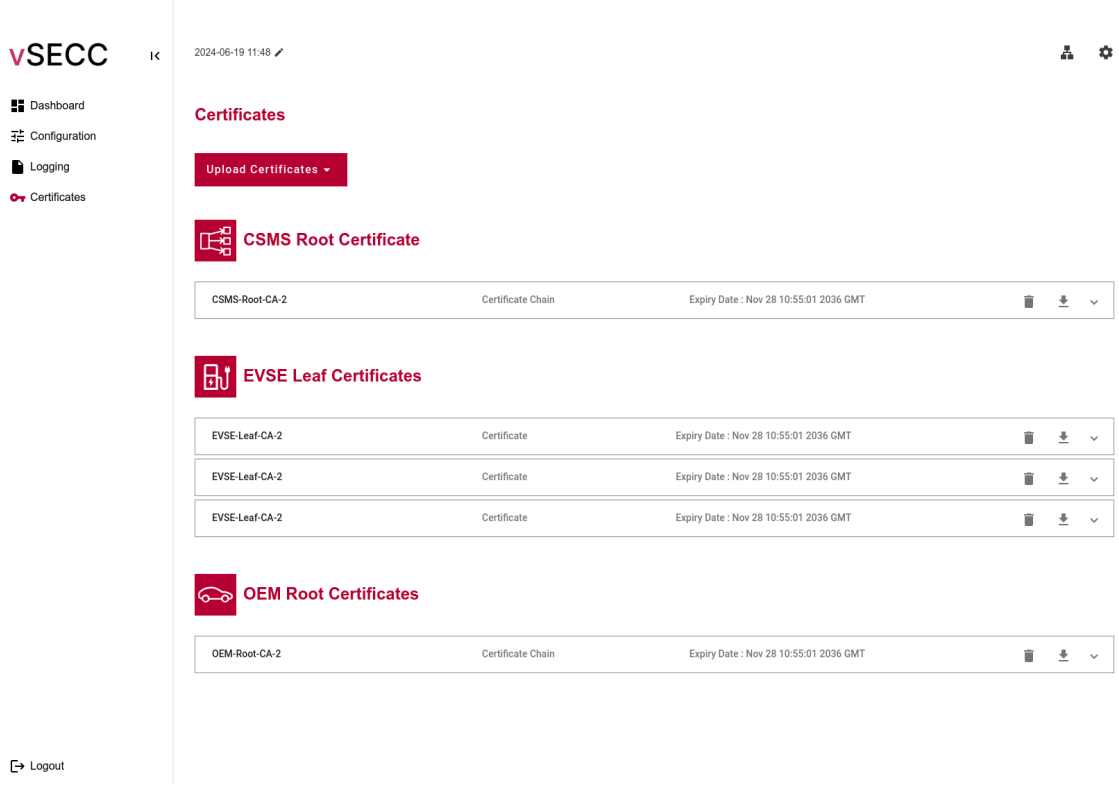


Figure 65: Installed Certificates on vSECC Controller

7.6.1 Installing Root Certificate Authorities (CAs)

When connecting to a CSMS using a secured TLS connection, the vSECC Controller uses its installed root certificate authorities (Root-CAs) for verifying the server's certificate chain. The vSECC Controller already comes with the mozilla root certificate store pre-installed. Additional Root-CAs can be installed either by using the web interface or via OCPP.

To install a Root-CA, press the **[Upload Certificates]** button as seen in Figure 66. Select **CSMS Root Certificate** from the drop down menu, then press the **[Select Certificate]** button to select a certificate you would like to install. Pressing the **[Upload]** button finishes

the installation of the certificate. Please keep in mind that root certificates installed through the web interface are not taken into account when verifying Certificate Signing Requests (CSRs). Instead, they are only used for verifying a server's certificate during the websocket connection's TLS handshake.

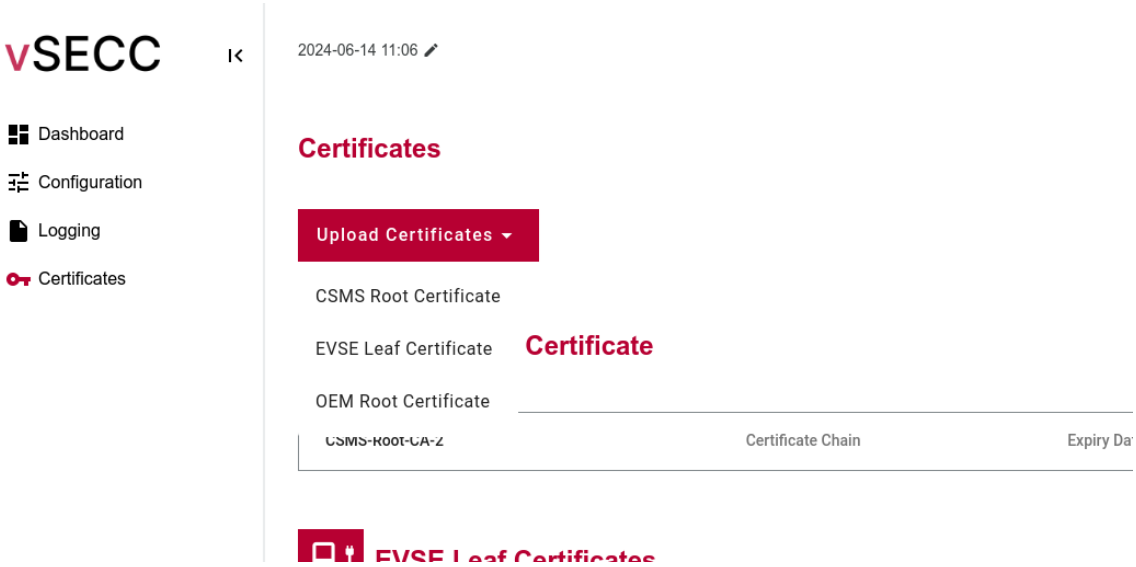


Figure 66: Installing Root-CAs

Furthermore, installing a Root-CA is possible via OCPP. This is described in chapter 7.10.1.

7.6.2 Installing EVSE Leaf Certificates

To establish a secured TLS connection between EV and EVSE, a leaf certificate (chain) has to be provided. The leaf certificate will be validated against its corresponding private key during upload. After uploading the private key once, it is possible to upload new certificates signed by the private key without uploading the key again.
The vSECC Controller does not provide any EVSE certificates by default. It supports only **one** EVSE leaf certificate (chain's) private key at the same time for TLS 1.2, one for TLS 1.3 primary curve and one for the secondary curve. Any new upload will replace the old certificate and private key. When uploading a certificate chain, the certificate file has to contain the leaf certificate and all sub certificates.

Caution: The uploaded private key is neither transferred nor stored in a secure manner yet. Use this for development purpose only. Don't use for public or productive applications.

To install an EVSE leaf certificate (chain), select **EVSE Leaf Certifcate** from the drop down menu. Select the TLS version and purpose and then press the **[Select Certificate]** and **[Select Private Key]** buttons to select a certificate (chain) and the corresponding key you would like to install, as seen in Figure 67. Pressing the **[Upload]** button finishes the installation of the certificate chain and validates it against the provided key. After the upload, the current installed certificate appears in the Certificates Overview.

Certificates according to X.509 are supported. The certificate has to be signed according to ISO 15118-2 requirement [V2G2-006].

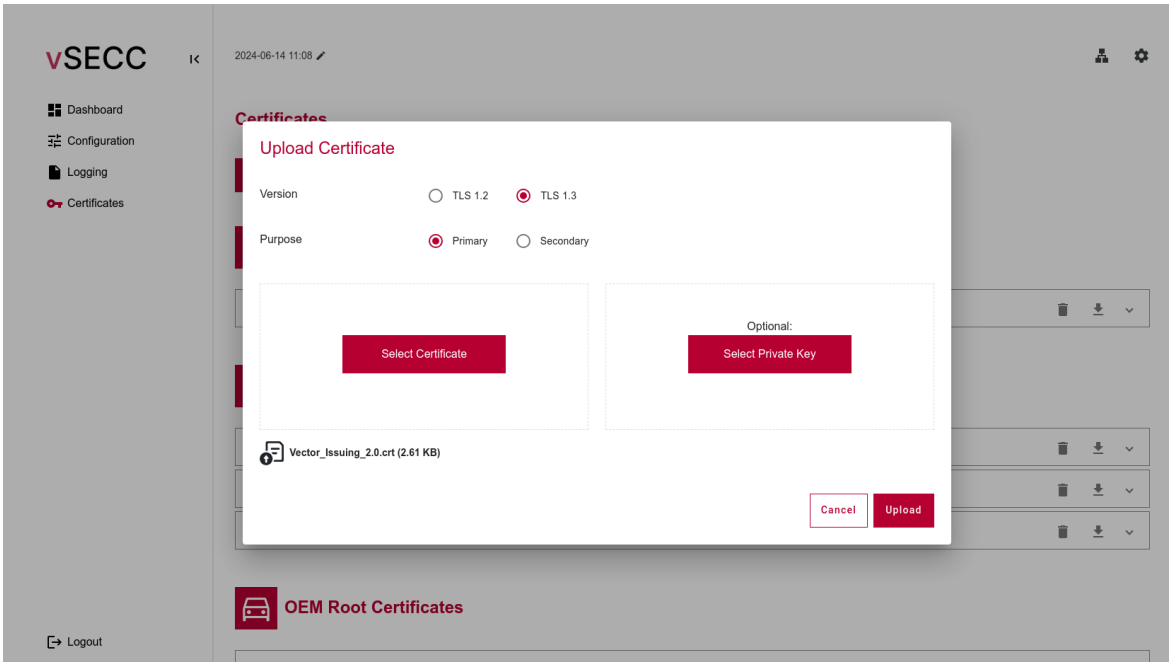


Figure 67: Installing an EVSE leaf certificate

7.7 Network Settings

In the top right corner, next to the General Settings, the Network Settings for the Ethernet interfaces **[ETH1]** and **[ETH2]** (vSECC only) can be changed in the web interface as shown in Figure 68. In this section you can also find information about the IPv4 and IPv6 routing that you may need to set up **Value Added Services** correct in your environment (see Section 8.6).

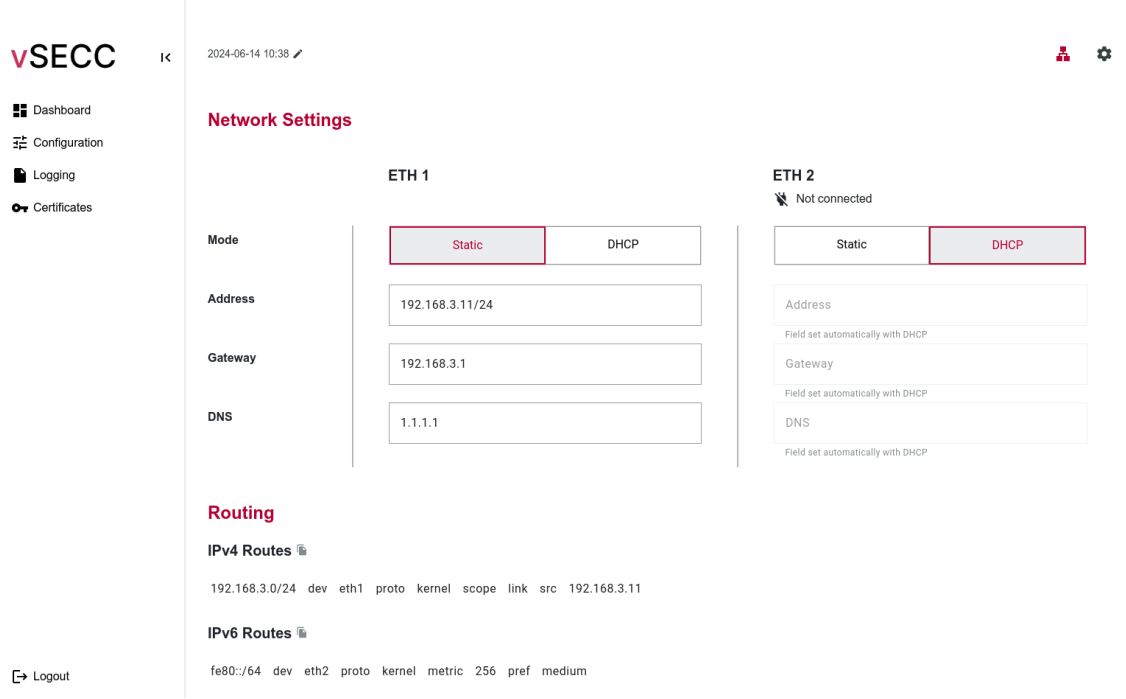


Figure 68: Change network settings

7.7.1 Network Interface Settings

For the Ethernet interfaces you can choose between the modes **[Static]** and **[DHCP]**.

The default configuration modes are:
For ETH1 **[Static]** with the IP address 192.168.3.11.
For ETH2 **[DHCP]**.

If the IP address of ETH1 is changed, the web interface can only be reached via the new IP address. The settings are applied after pressing the **[Save]** button. Confirming the settings by pressing **[Apply and Restart]** will restart the network interfaces.

Caution: If you change the mode or IP address of ETH1, the web interface is no longer reachable under 192.168.3.11. Remember to use the new assigned IP address to reach the web interface.

The execution of a factory reset (see 2.3.1) will restore the factory default network settings:

> Interface ETH1: Static 192.168.3.11/24

> Interface ETH2: DHCP

7.7.2 Routing Information

In this section you can find the routing information for **IPv4** and **IPv6**, that are configured on the system.
The lists are displayed as the linux ip route command shows them. You can copy the whole output to the clipboard using the copy icon next to the headline.

7.8 General Settings

In the top right corner of the web interface, the General Settings can be accessed. The password to access the web interface can be changed in the **User Management** section. The **System** can be rebooted / restarted. The **Firmware Update** section allows to upload and run an update of the vSECC Software. In the future, license files can be updated in the **License Update** section. Today, the installed licenses are shown here, additionally to the Dashboard. Finally, **Scripts** (signed by Vector) can be uploaded and run for support purposes.

7.8.1 User Management

Changing the default password of the web interface user is possible via the section shown in Figure 69.
First, enter the new password in the field **Password** and repeat it. If both input fields match, the new password will be set by pressing the **[Save]** button.

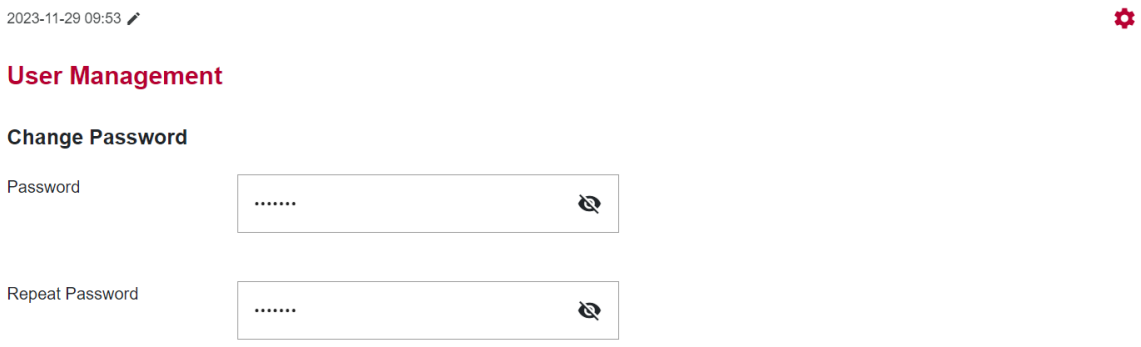


Figure 69: Changing the Web Interface Password

7.8.2 System

Here, the vSECC Controller can be rebooted (hard reset). Alternatively, the vSECC Software can be restarted (soft reset).
Pushing **[Reboot vSECC]** causes a reboot of the whole system. The last boot reason is shown on the System Dashboard at the Controller Status.

Rebooting the system may take up to two minutes.

Pushing **[Restart Software]** causes a shutdown and restart of the vSECC Software Application. This is equivalent to the functionality of enabling the ”Maintenance Mode”, that was used before vSECC Software Version 3.0.

Figure 70: Rebooting or Restarting the Controller

7.8.3 Firmware Update

A new firmware can be installed on the vSECC Controller either via the web interface or via an OCPP-based charging station management system like vCharM (see chapter 7.10.4).
You can see the currently installed Firmware Version.
To update the firmware, press **[Upload Firmware]**, then select the new firmware update file that you have stored on your PC by pressing on the **[Select Firmware]** button. Pressing **[Upload Firmware]** initiates the update process. The entire update process including the upload takes about 2 minutes. Please do not turn off the device while the update is running.

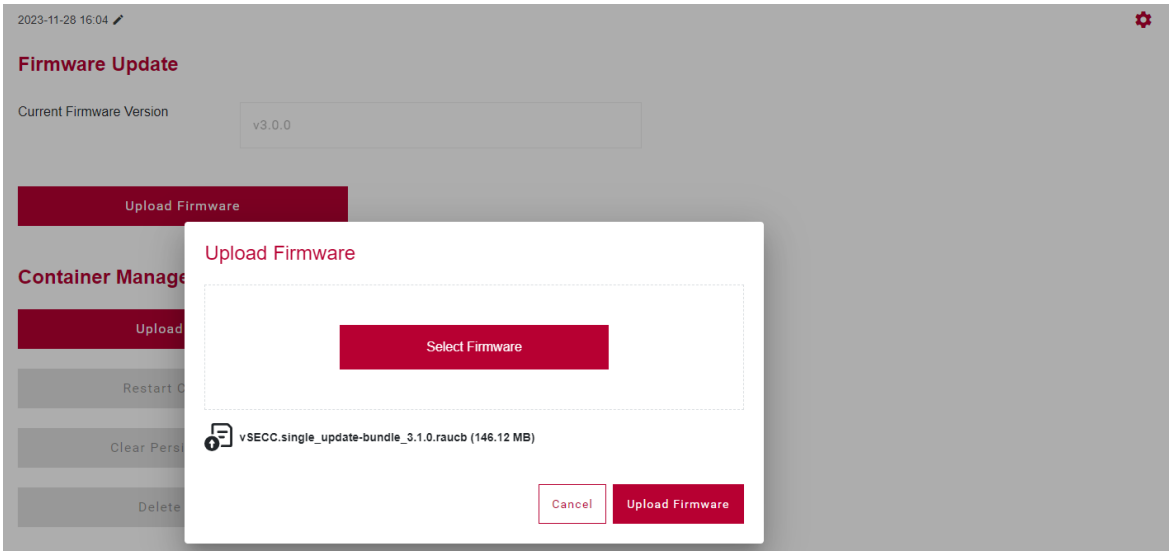


Figure 71: Uploading a new Firmware

After a successful upload, the vSECC Controller must be rebooted. With the reboot, the device boots into the new installed firmware.

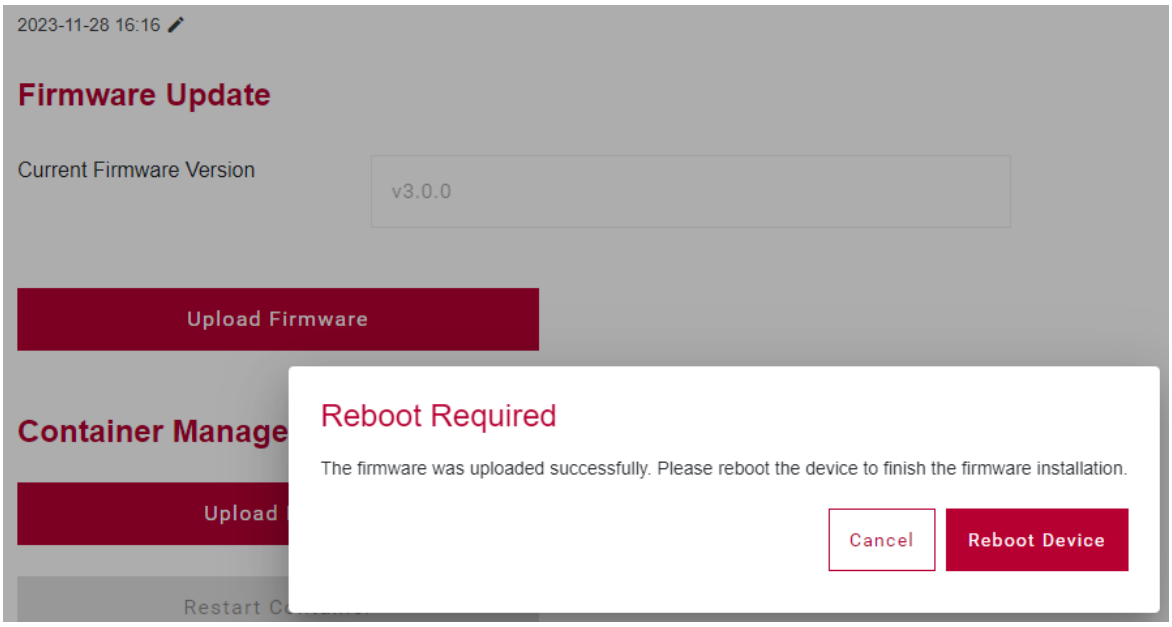


Figure 72: Reboot Request after Successful Firmware Upload

Caution: By installing a new firmware, an active container environment will be stopped. This will clear the container environment to the state when the container was installed (the `rootfs` of the container will be reset). For container configuration and usage refer to chapters 7.4 and 8.27.

7.8.4 Firmware Update of External Peripherals

Firmware updates of external peripherals can be installed through the vSECC Controller either via the WebUI or via an OCPP-based charging station management system like vCharM (see chapter 7.10.4).
An external peripheral that wants to consume updates can either directly use MQTT and the REST API as described below to monitor for new update files and download them. Alternatively a container on the vSECC Controller (see 7.4) can be used to monitor for new updates and redirect the relevant files to the appropriate external peripheral.

Update via CSMS Updating external peripherals via OCPP uses the same mechanism as described in section 7.10.4.
Arbitrary update files can be sent to the vSECC Controller from the CSMS and the vSECC Controller will automatically decide how to handle the file. If the vSECC Controller detects that the update file is a suitable vSECC Controller update, it will perform the update on itself as described in 7.10.4. Otherwise the vSECC Controller stores the update file and announces its presence to all external peripherals.

Only a single device can be updated via OCPP simultaneously.

The file size limit is 500MB, but due to hardware limitation no larger than 200MB is recommended.

A new pending update file is announced by the vSECC Controller through a message to the MQTT topic `vsecc/firmware_external/update_pending` containing the file name of the update file. The stored update file can be retrieved from the vSECC Controller through the REST API (see 7.9) route `GET /firmware/external/{name}`, where the `{name}` fragment must be replaced with the update file name.
To report the update installation status back to the CSMS, a message must be sent to the MQTT topic `vsecc/firmware_external/report_status`. This message must containing the filename of the firmware update file and the current installation status (”INSTALLING”, ”REBOOTING”, ”FAILED” or ”SUCCESS”). If no such message is sent within a configurable timeout window, the vSECC Controller will assume that no external peripheral has consumed the update and will report an installation failure back to the CSMS. The timeout duration can be configured through the **External Peripheral Firmware Update Timeout** configuration variable.
The update file is automatically deleted from the vSECC Controller after the external peripheral has reported a successful/failed installation or the timeout has expired.

Update via WebUI The WebUI allows to upload multiple update files under the general settings section below the vSECC controller update section. (Fig. 73)
Immediately upon each upload of an update file, the name of the file is published to the MQTT topic `vsecc/firmware_external/update_pending`. An update file can be downloaded through the REST API (see 7.9) route `GET /firmware/external/{name}`, where the `{name}` fragment must be replaced with the update file name.
The list of available files can be viewed and modifed in the WebUI or through the REST API route `GET /firmware/external`. Files uploaded over the WebUI are NOT automatically deleted from the vSECC Controller after installation. They can be deleted manually in the WebUI or through the REST API routes `DELETE /firmware/external/{name}` (for individual files) or `DELETE /firmware/external` (for all files at once).

The update installation status reported by an external peripheral is NOT displayed in the WebUI.

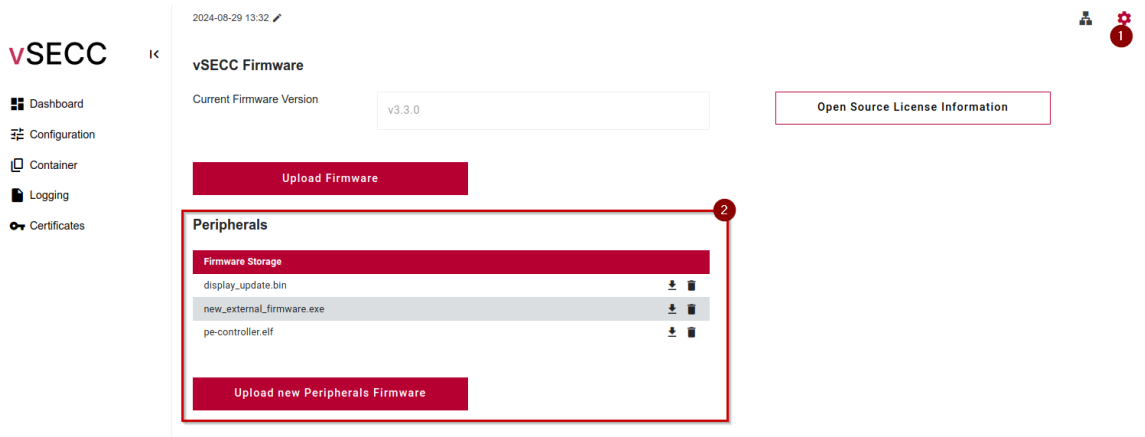


Figure 73: Upload external peripherals update files

7.8.5 License Update

To use certain features, a special license is required. To purchase a license and receive the license file, please get in touch with your sales contact.
In order to install the license and check if it has been correctly installed, perform the following steps:

> Upload the license file, as described in the Email that delivers the license
> Restart the vSECC Software at least once, so the license gets read by the software
> Check if the shown licenses match the purchased licenses.

The validity of the license is only checked within the vSECC Software, not during the upload in the web interface. If an uploaded license is invalid is shown in the the log file vsecc.log. When you upload a license and afterwards something does not work, please revise the log file.

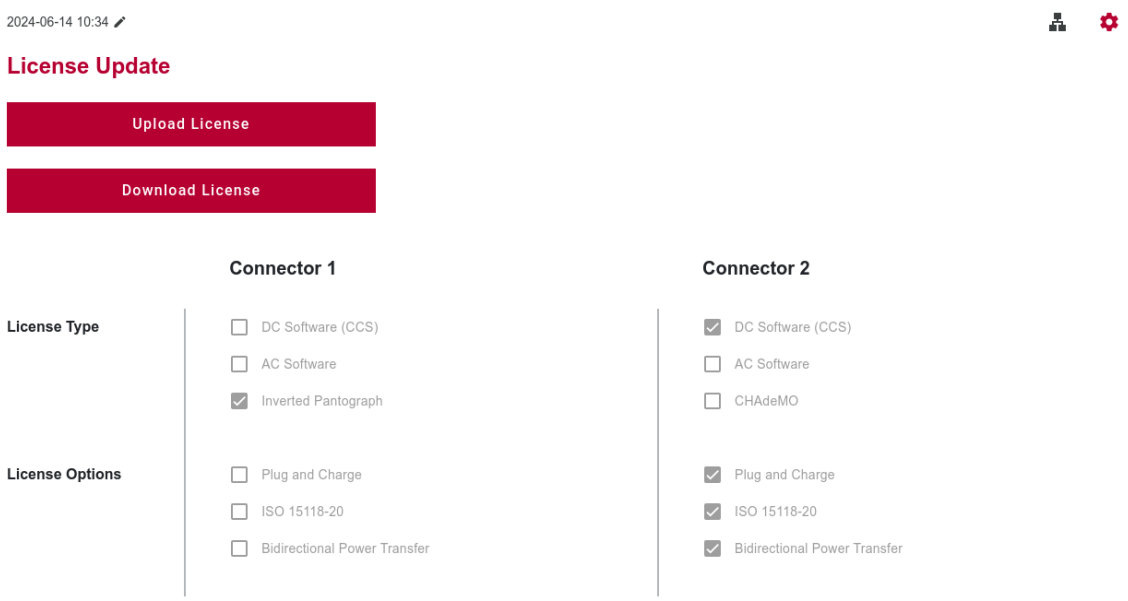


Figure 74: Installed Licenses are Shown in Web Interface

7.9 Configuration via RESTful API

The vSECC Controller can also be configured using a RESTful API interface. It is specified using the OpenAPI v3.0.3 standard, this comes with a generated Swagger UI API GUI available on the vSECC Controller.

For more information about the Swagger UI and OpenAPI, visit
`https://spec.openapis.org/oas/v3.0.3`
`https://swagger.io/tools/swagger-ui`

The GUI can be accessed under:
<IP-Address>/vsecc-api
e.g. `http://192.168.3.11/vsecc-api`

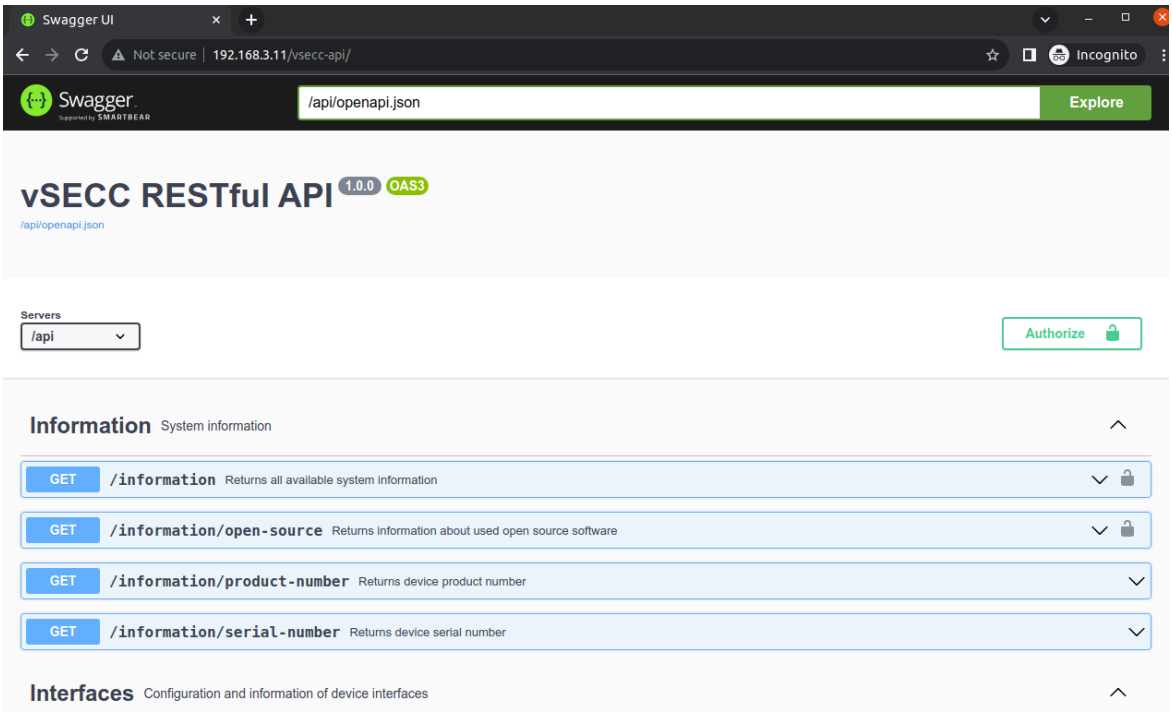


Figure 75: Generated OpenAPI Swagger UI

7.9.1 Using the Swagger UI

The Swagger UI shows all available REST routes. Most routes require authorization, this is annotated by the small padlock on the right side as seen in Figure 76, the padlock is only shown if authorization is required. How to authorize is described in chapter 7.9.2.

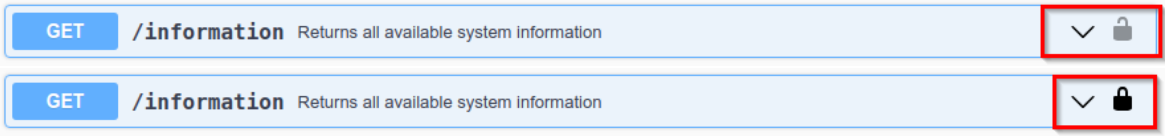


Figure 76: Swagger UI Authorization Status

When authorized, the GUI can help you creating requests or getting information. Please perform the following steps.

1. Choose any route and open its drop down
2. Click on **[Try it out]** (Fig. 77)
3. Click on **[Execute]** and wait for the response (Fig. 78).

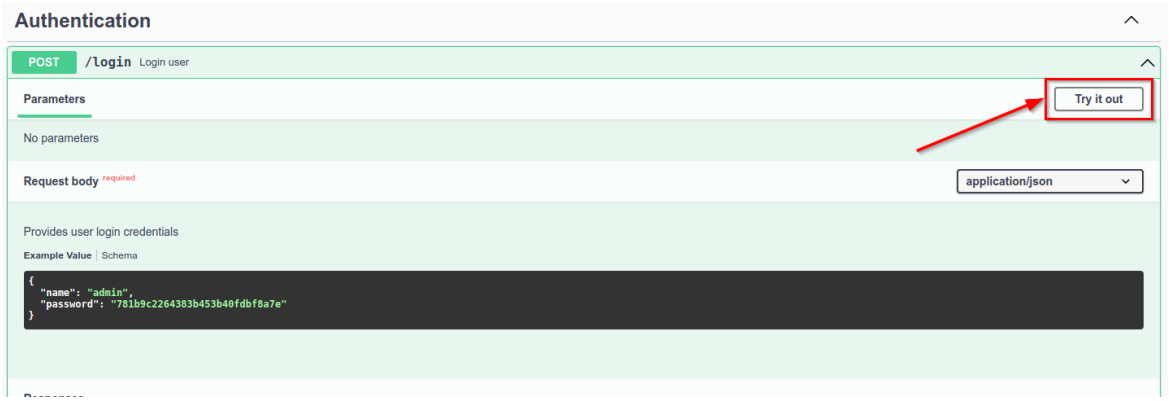


Figure 77: **[Try it out]** Button to Enable the REST Client Functionality

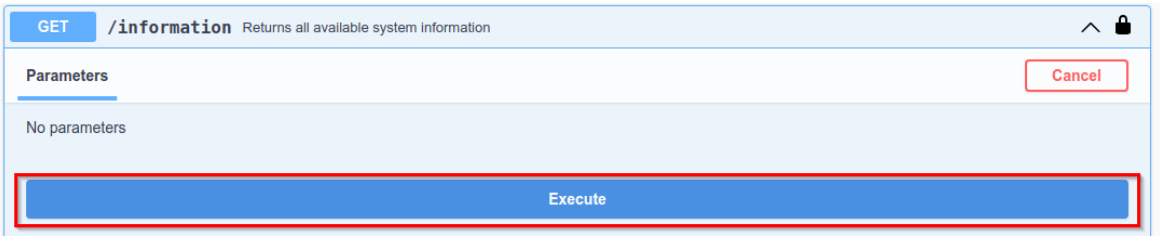


Figure 78: **[Execute]** Button to Send API Request

As shown in Figure 79 As shown in Figure 50, the GUI will generate a cURL command and the associated URL to use in shell scripts **(1)**. Below, the Response body **(2)** and Response header are shown.

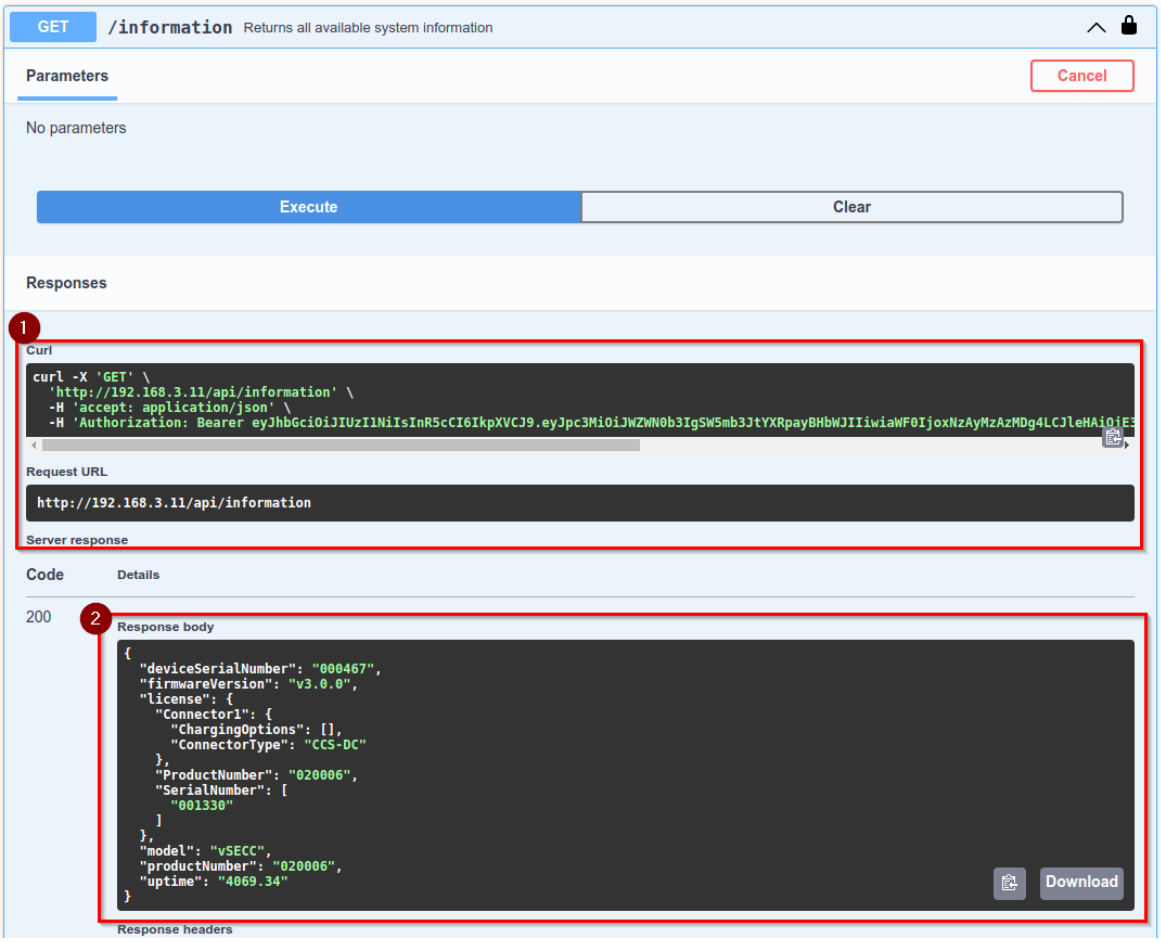


Figure 79: (1) Generated cURL Command and Requested URL; (2) Received Response Body

7.9.2 Authorization

Currently the REST API only supports authorization via JWT[1]. The token can be obtained using the `Login` route belonging to **Authentication**. Please follow these steps to receive a valid JWT.

The credentials required for accessing the web interface for the first time consist of
username "`admin`"
password "`rootpassword`".

[1] https://datatracker.ietf.org/doc/html/rfc7519

On First Login

1. Navigate to the `/login` route belonging to **Authentication** and open the drop down
2. Click on **[Try it out]** (Fig. 77)
3. Enter the login credentials for accessing the web interface in the request body **(1)** (Fig. 80)
4. Click on **[Execute] (2)** and wait for the response
5. Go to the response body and copy the JWT to your clipboard (Fig. 81)
6. Navigate to the `/users/credentials` route belonging to **Configuration**
7. Click on the padlock of the route, this will open an authorization dialog window
8. Paste the JWT into the *pwReset* input field and click **[Authorize]** and **[Close]** (Fig. 83)
9. Open the drop down of the route
10. Click on **[Try it out]**
11. Enter a new Password in the Request body **(1)** (Fig. 84)
12. Click on **[Execute]** and wait for the response. You should now be authorized.

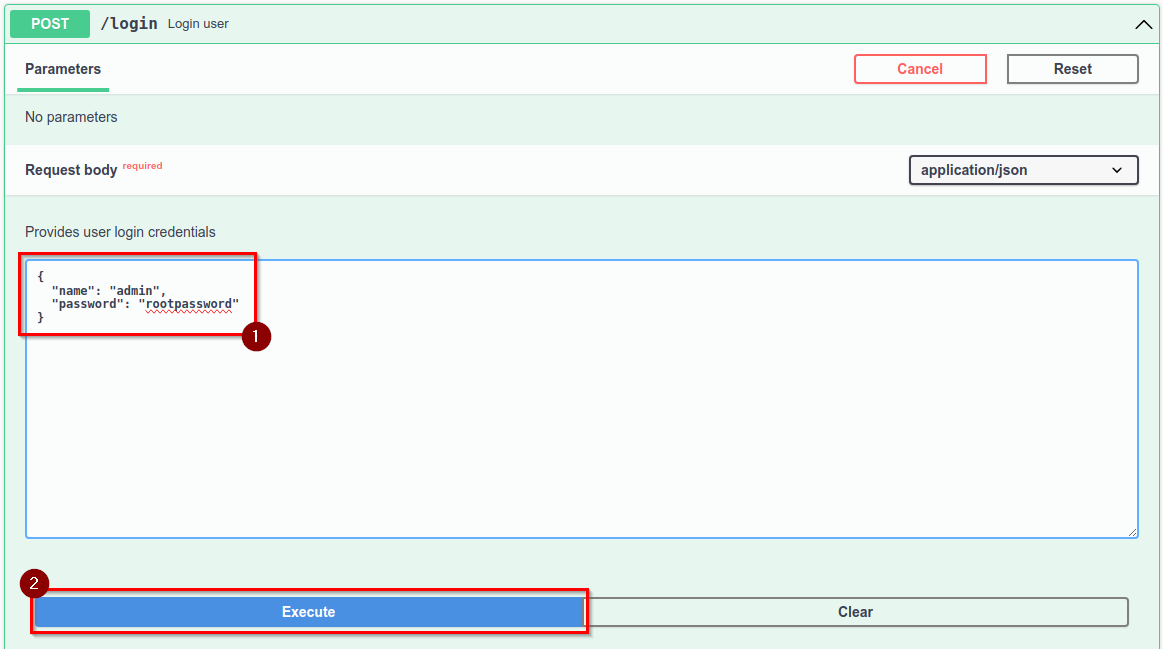


Figure 80: Entering the Credentials for Login

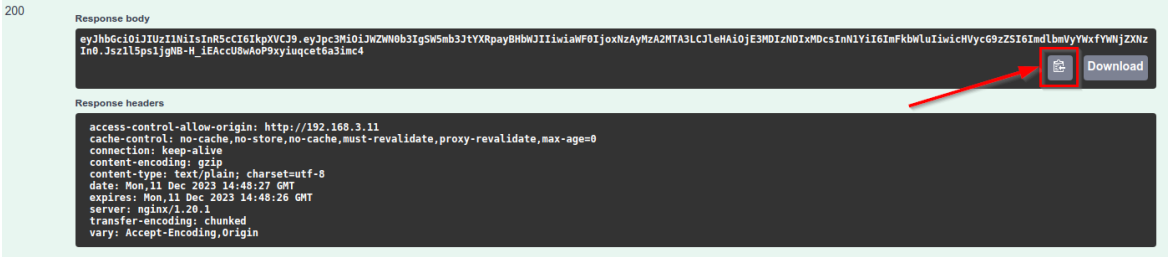


Figure 81: Copy JWT for the First Login

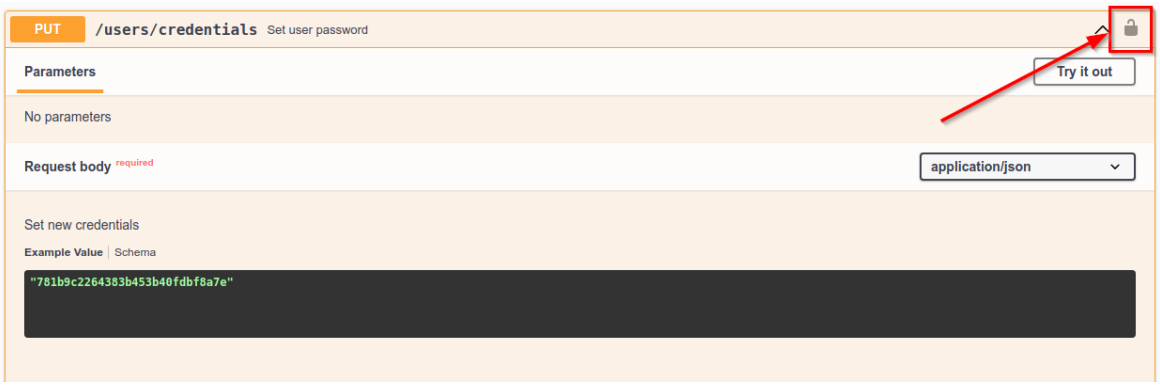


Figure 82: Setting the Password on the First Login

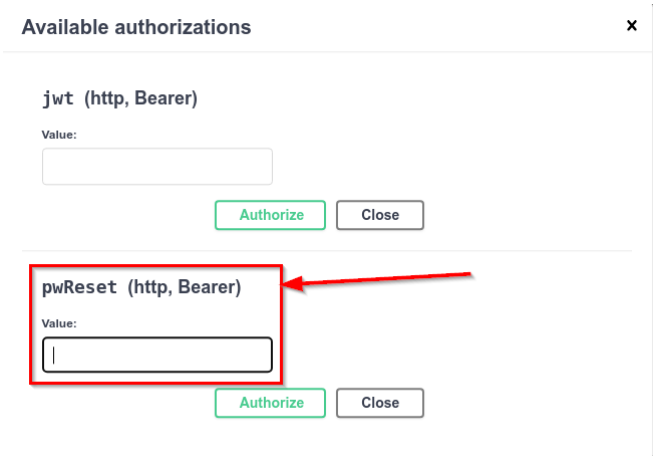


Figure 83: Paste JWT to Change the Password on the First Login

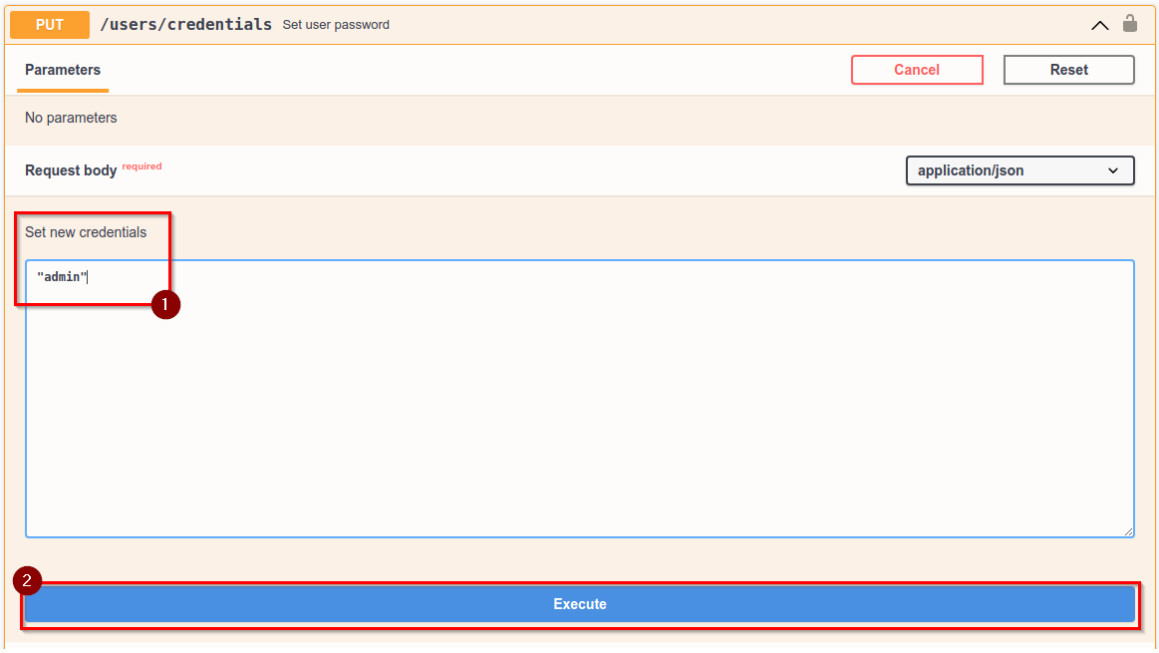


Figure 84: Setting a New Password

Normal Login

1. Navigate to the `/login` route belonging to **Authentication** and open the drop down
2. Click on **[Try it out]** (similar to fig. 77)
3. Enter your credentials in the Request body **(1)** (Fig. 85)
4. Click on **[Execute] (2)** and wait for the response
5. Go to the Response body and copy the JWT to your clipboard (similar to fig. 81)
6. Click on any open padlock. This will open the authorization window.
7. Paste the copied token into the *jwt* input field and click **[Authorize]** and **[Close]** (Fig. 86)
8. All padlocks should now be locked.

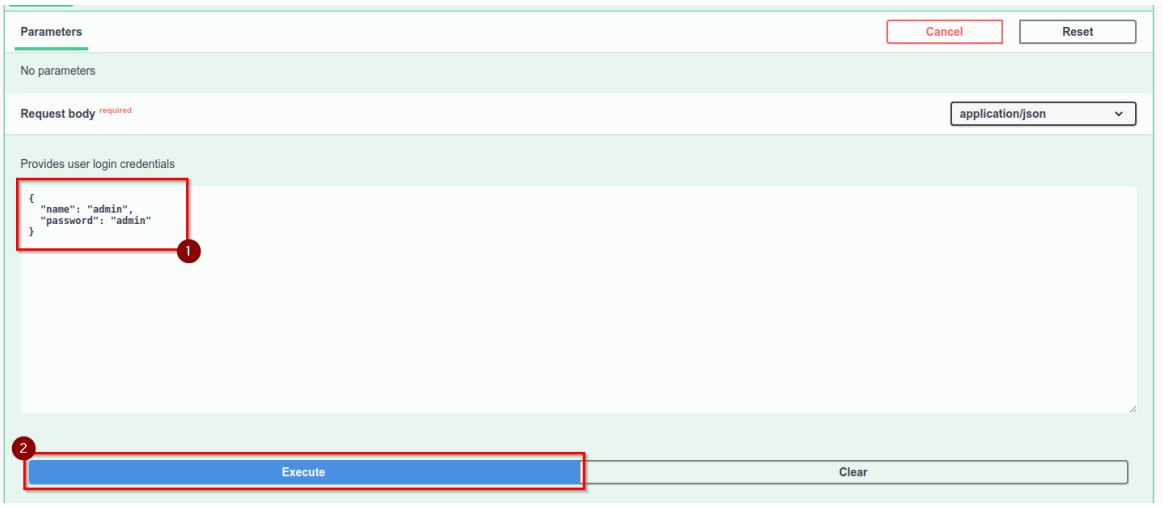


Figure 85: Normal login

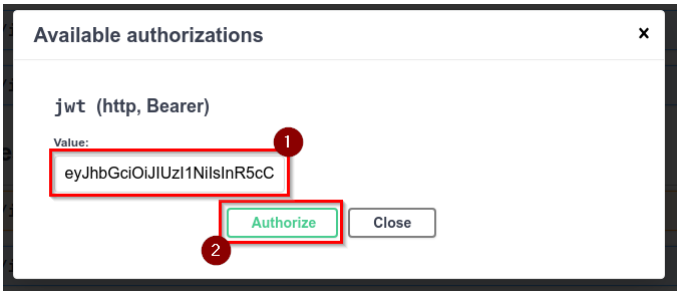


Figure 86: Past jwt normal login

7.10 Web Interface Features available via CSMS

7.10.1 Configuration

As an alternative to the web interface, the vSECC Controller can also be configured by using a CSMS. The vSECC Software uses the OCPP 1.6 messages `GetConfiguration` and `ChangeConfiguration` / OCPP 2.0.1 messages `GetVariables, GetBaseReport` and `SetVariables` for configuration data exchange with the CSMS. For further information about the structure and usage of those messages, please refer to appropriate OCPP specification.
Inside Vector's CSMS solution vCharM for example, the vSECC Controller's variables are presented as shown in Figure 87. The information about the available variables is gathered automatically when the vSECC Controller establishes its connection. Changes to any variables are sent to the vSECC Controller, where they are validated and then applied to its configuration.

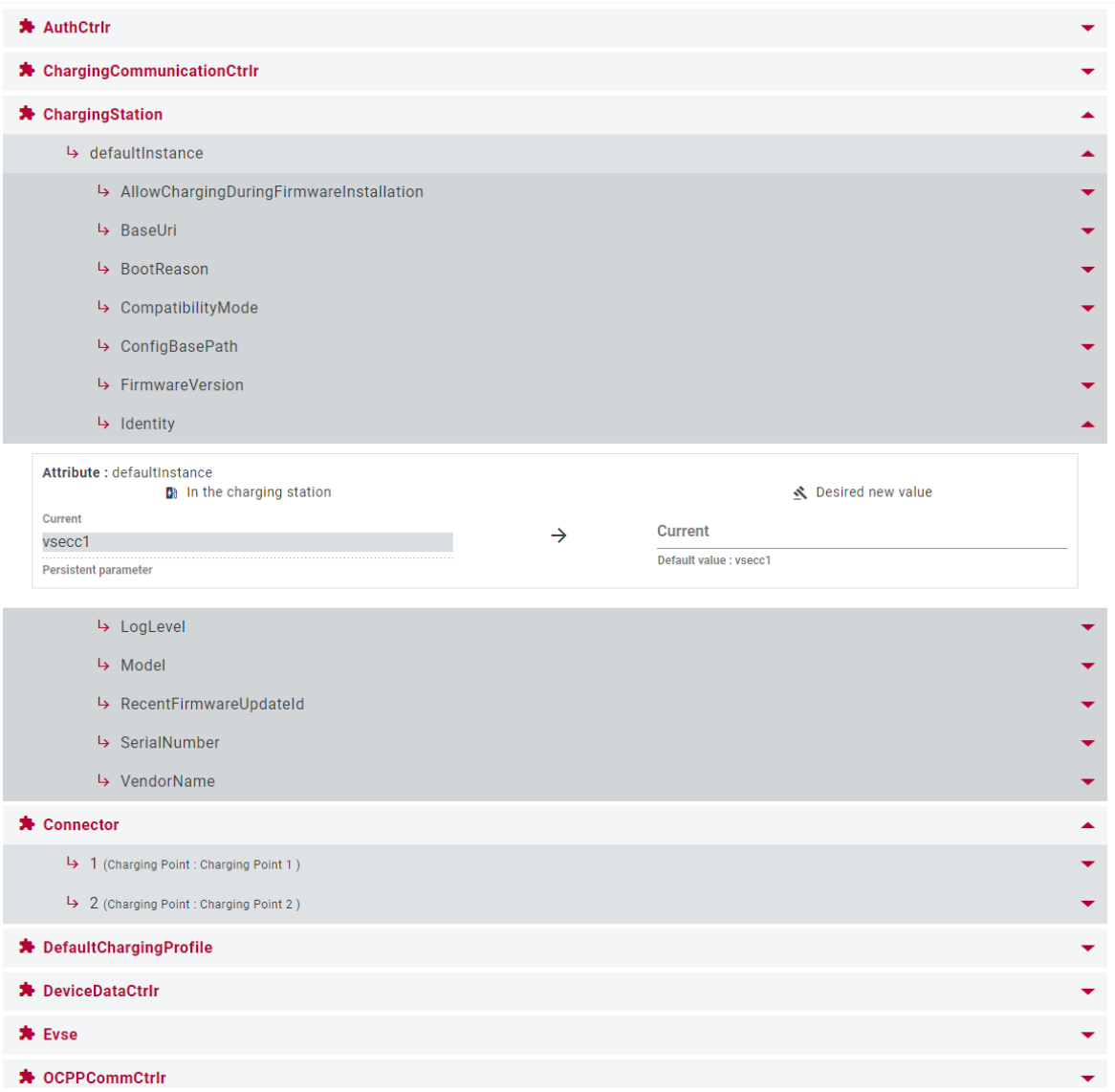


Figure 87: vCharM Charging Station Configuration

Reporting of configuration variables using OCPP 1.6J OCPP 1.6J does not support the concept of a device model to report a charging station's configuration as specified within OCPP 2.x. However, the CSMS still has the possiblity to request the configured variables using the `GetConfiguration.Req` message.

If a variable within the device model can be mapped to a specified OCPP 1.6J variable, the vSECC Software will report the variable under that name. If a variable within the device model does not have an equivalent within OCPP 1.6J or is custom to the vSECC Controller's configuration, the variable is reported using the following naming scheme:

- Component (max. 15 characters)
- Component Instance or EVSE-ID (max. 3 characters)
- Variable (max. 25 characters)
- Variable Instance (max. 3 characters)

All parts are separated using forward slashes (`/`). Empty parts, for example if no instance name exists, map to empty strings. The slash is not ommitted in that case. Some examples:

- Component "AuthCtrlr" with variable "Enabled":
  "AuthCtrlr//Enabled/"
- Component "Connector" of EVSE "1" with variable "ConnectorType":
  "Connector/1/ConnectorType/"
- Component "SmartChargingCtrlr" with variable "LimitChangeSignificance":
  "SmartChargingCt//LimitChangeSignificance/"
- Component "SampledDataCtrlr" with variable "TxUpdatedInterval" which maps to an OCPP 1.6J specified variable:
  "MeterValueSampleInterval"

OCPP 1.6J Configuration Keys The following configuration keys of the OCPP 1.6J specification are supported:

- AllowOfflineTxForUnknownId
- ConnectionTimeOut
- GetConfigurationMaxKeys
- HeartbeatInterval
- MeterValuesSampledData
- MeterValueSampleInterval
- MinimumStatusDuration
- NumberOfConnectors
- StopTransactionOnEVSideDisconnect
- StopTransactionOnInvalidId
- StopTxnSampledData

- TransactionMessageAttempts
- TransactionMessageRetryInterval
- WebSocketPingInterval
- AuthorizationKey
- CpoName

7.10.2 Log File Management

Downloading log files via OCPP uses the `GetLog` messages. The vSECC Software will compress its log files into an archive and upload it to the URI specified by the CSMS. The URI must point to a directory and not a file. A URI pointing to a file is interpreted to rename the uploaded file by some file transfer protocols. This is not intended, as the vSECC Controller is responsible to choose a name and notify the CSMS about it.
For further information about the structure and usage of those messages, please refer to the appropriate OCPP specification. In vCharM for example, the log files can be requested via the **Request Logs** dialog. The uploaded log files can then be retrieved from vCharM and downloaded as a local copy, as can be seen in Figures 88 and 89.
If the target server of the log file upload requires credentials, they can be set in the **HTTP Basic Authentication / Logfile Upload** Section of the Charging Station Management System Configuration. If the CSMS sends its own credentials embedded in the download URL, the configuration variables are ignored. For further details on credential options, please refer to subsection 7.10.4. For use with vCharM, set the configuration variables for **HTTP Basic Authentication / Logfile Upload** equal to **HTTP Basic Authentication / CSMS Connection**.

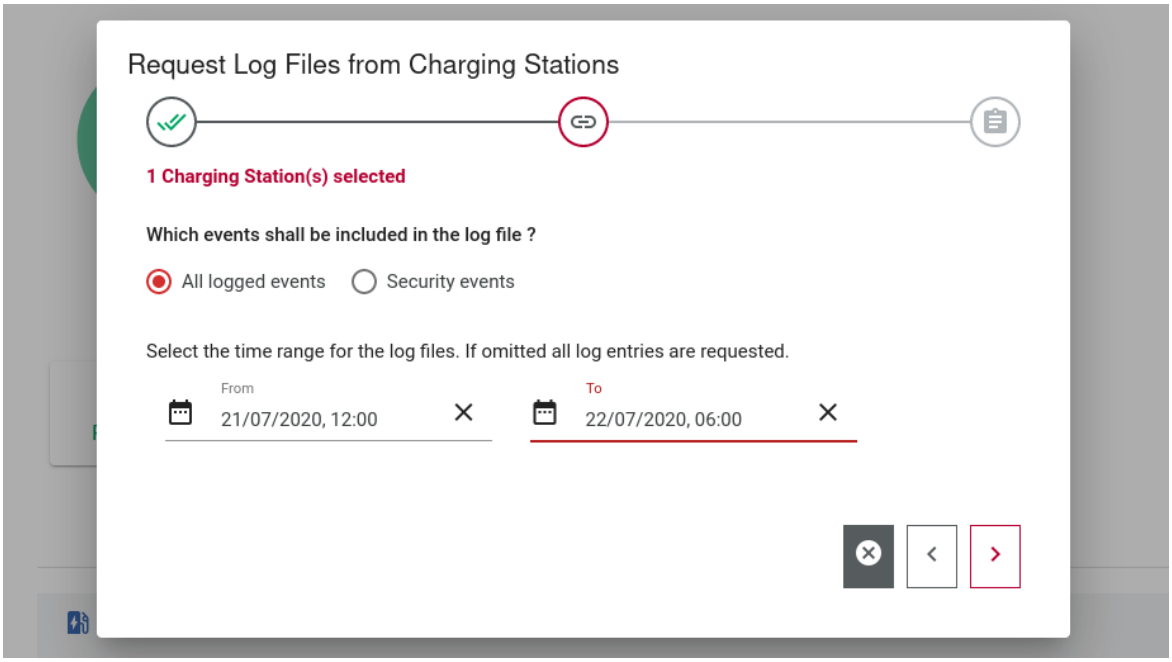


Figure 88: vCharM Requesting Log Files

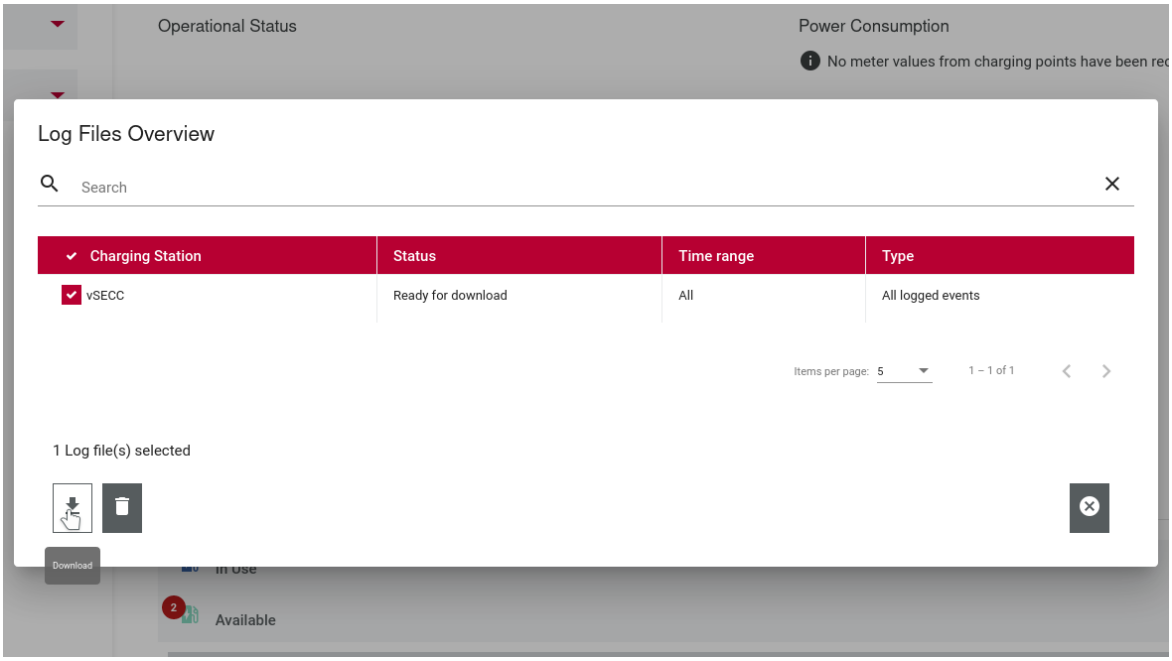


Figure 89: vCharM Downloading Log Files

7.10.3 Certificate Management

Installing a Root-CA via OCPP uses the `InstallCertificate` messages. The vSECC Controller will store the transmitted certificate inside its trusted storage. The `DeleteCertificate` OCPP messages can be used for deleting a previously installed certificate. For further information about the structure and usage of those messages, please refer to the appropriate OCPP specification. Root certificates installed via OCPP are used for both the websocket connection's TLS handshake and verifying Certificate Signing Requests (CSRs).

7.10.4 Firmware Update via CSMS

Updating the vSECC Controller's firmware via OCPP utilizes the `UpdateFirmware` messages. The firmware will be downloaded from the URI specified inside the CSMS' request. After a successful download, it will then install the firmware update and reboot. For further information about the structure and usage of those messages, please refer to the appropriate OCPP specification.
Inside vCharM, a firmware update can be requested via the **Update Firmware** dialog as seen in Figure 90. You can then upload the firmware file or specify the file's location using an URI. After scheduling the update to a certain point in time, the request is sent to the vSECC Controller, which initiates the update process.
If the server where the firmware update file is located requires credentials, they can be set in the
HTTP Basic Authentication / Firmware Update Section of the Charging Station Management System Configuration. If the CSMS sends its own credentials embedded in the firmware URL, the configuration variables are ignored.
Special characters like "@" must be encoded in URL embedded credentials. Credentials set in the configuration must not be encoded. Anonymous logins are possible by leaving

credentials in the URL and the configuration variables empty. In the case of FTP, setting the user to ”anonymous” via the URL credentials allows to explicitly override configuration credentials in order to log into the server anonymously. Non-default ports can be chosen by an appropriate URL modification, too.
For use with vCharM, set the configuration variables for **HTTP Basic Authentication / Firmware Update** equal to **HTTP Basic Authentication / CSMS Connection**.

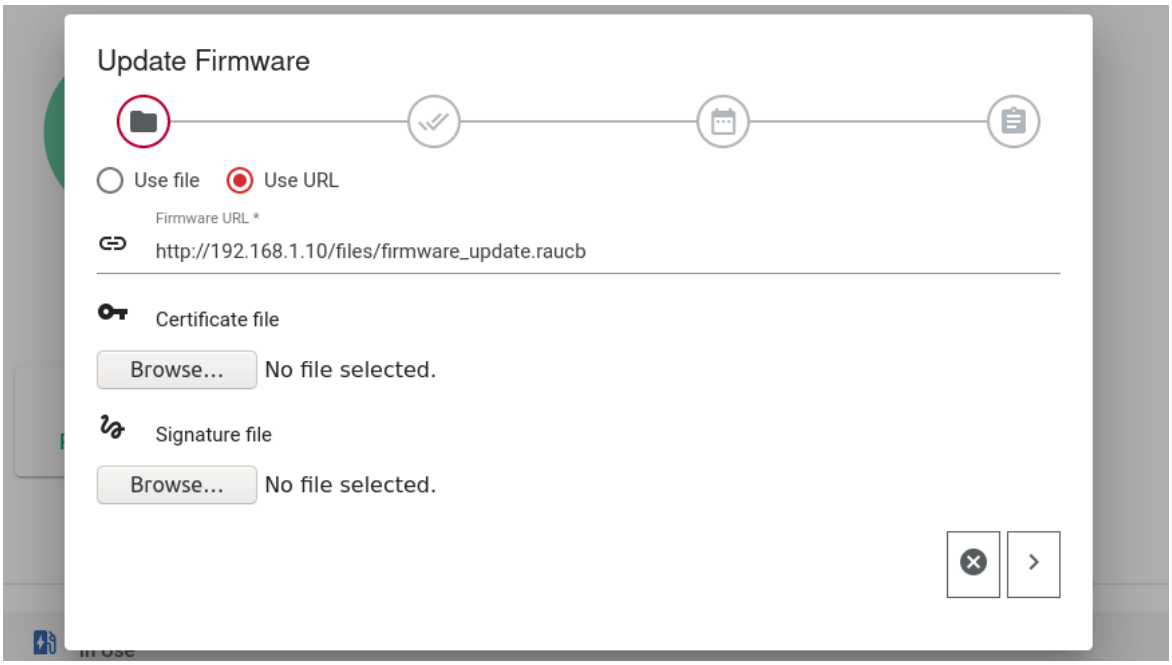


Figure 90: vCharM Firmware Update

7.10.5 EVSE Topology

The configuration’s structure is based on OCPP’s 3-tier model which is also used by vCharM. This model describes the charging infrastructure on a logical level, consisting of three elements: *Charging Station*, *EVSE* and *Connector*.

Charging Station The term charging station describes a physical system where an EV can be charged. Each vSECC Controller corresponds to one charging station. This relationship is based on a unique charging station OCPP ID for each charging station. Figure 91 shows the edit menu of the charging station ”vSECC” which consists of two EVSEs with one connector each.

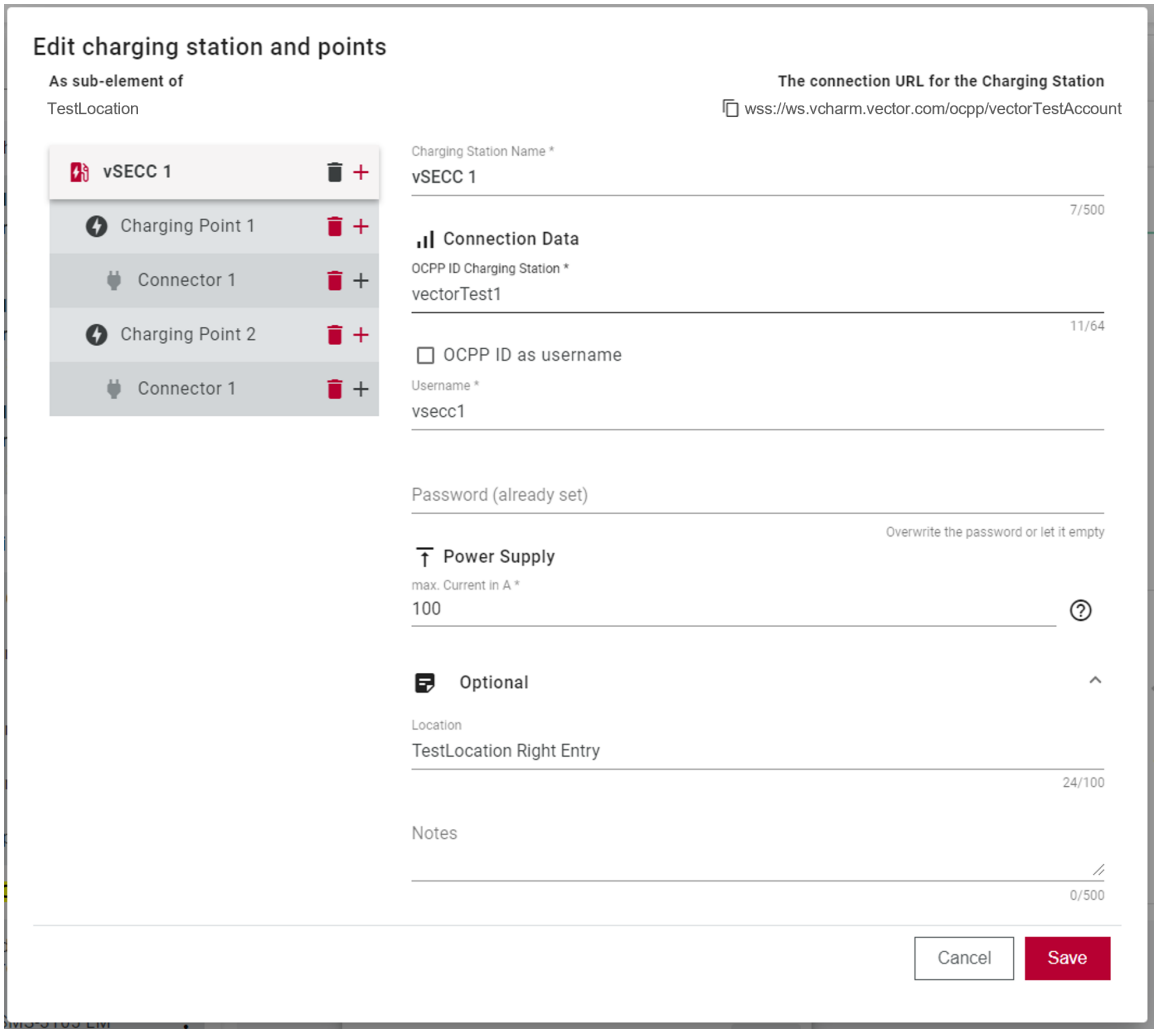


Figure 91: vCharM Charging Station Editing

EVSE An EVSE is defined by its ability to deliver energy to one EV at a time. A charging station can be connected to one or more EVSEs. Since the 3-tier model operates on a logical level, no assumptions are made about the physical hardware mapping. For example, the EVSE might be integrated into the charging station device itself. However, it could also be placed in a separate power electronics casing outside the charging station.

Connector The term connector describes an electrical outlet on a charging station. It is connected to a single EVSE. An EVSE can have multiple connectors attached to it, e.g. one CCS and one CHAdeMO compliant outlet. However, an EVSE will always use only one of its connectors exclusively. The complete 3-tier model is visualized in Figure 92.

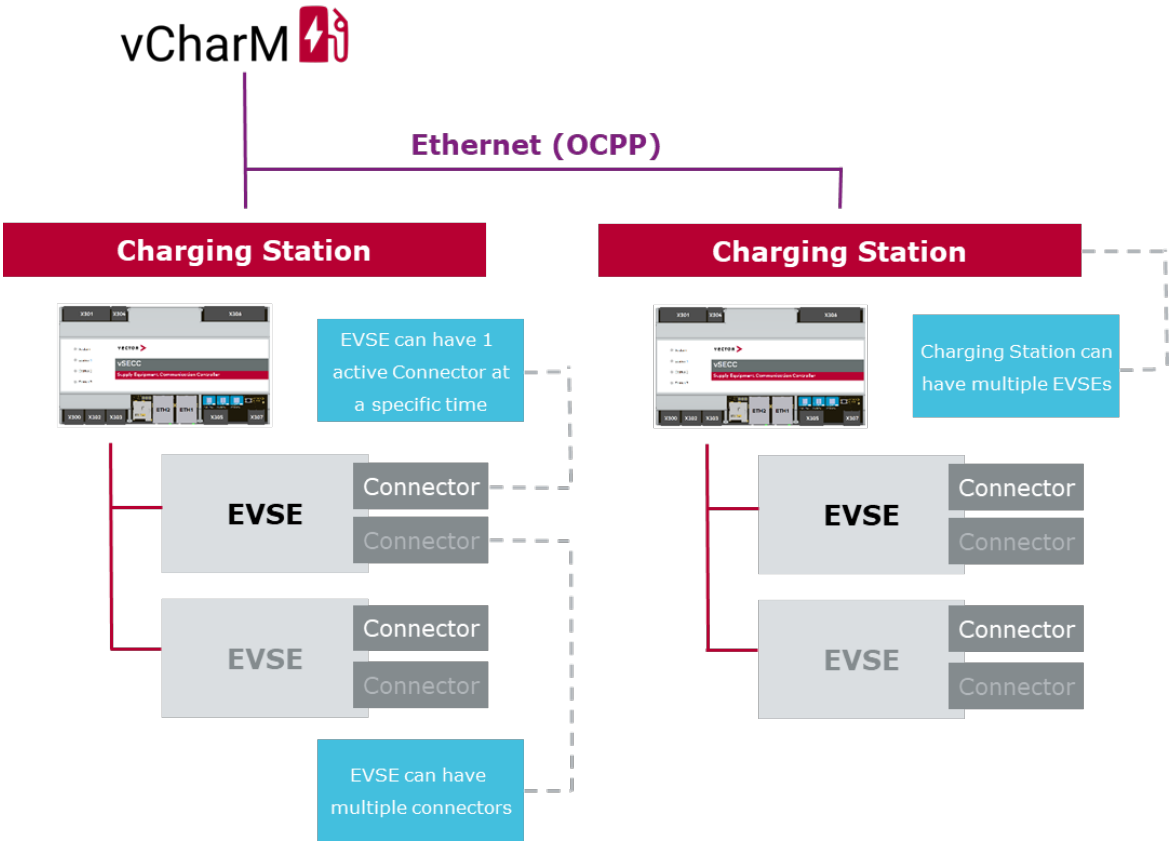


Figure 92: Overview according to OCPP 2.0: Part 1

8 User Guide

In this chapter you will find the following information:

In this chapter, you will find step-by-step instructions how to use the features.

8.1 Remote Support

This feature is only usable if the Vector support requests you to use it.

In the case that Vector's support team needs access to the device via a terminal connection, you can start a remote support session via the web interface **Dashboard**. The start of such a session opens a connection to Vector's support servers and allows access to the device for the support team.

Prerequisites for a remote support session

> The device has internet access (via routing: IP, Gateway, DNS set by a DHCP server)
> Connection to a remote (internet) host on TCP port 22 is possible (without any kind of proxy or firewall in-between)
> Vector's support team has sent you a key and port

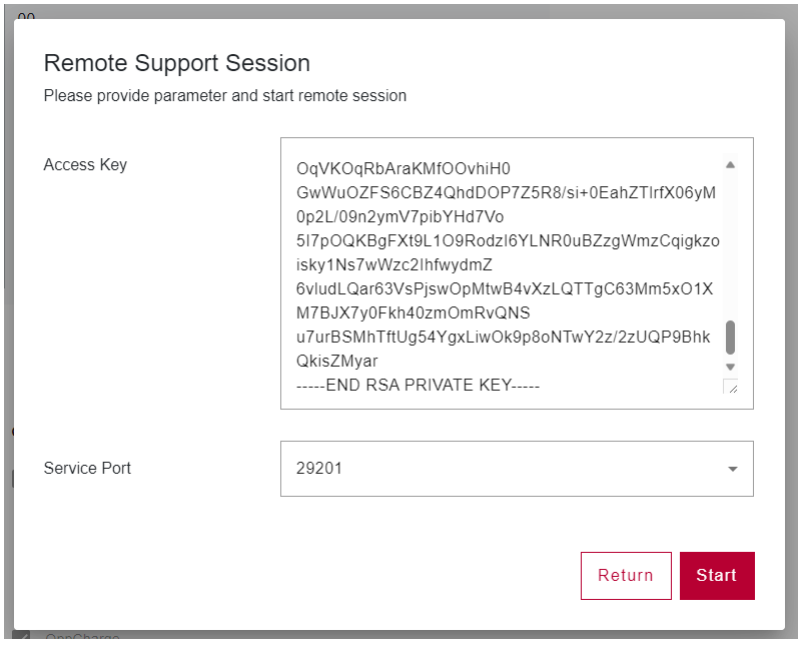


Figure 93: Start a remote support session

When Vector's support team wants to activate this feature, you will get a key in ASCII format that needs to be copied into the text field **Access Key**. Furthermore, a port number is provided to you, that needs to be selected in the drop down menu **Service Port**. Finally, a support tunnel is opened by pushing the button **[Start]**.
It is shown that the remote support session is running, when the button turns green as depicted in figure 94.

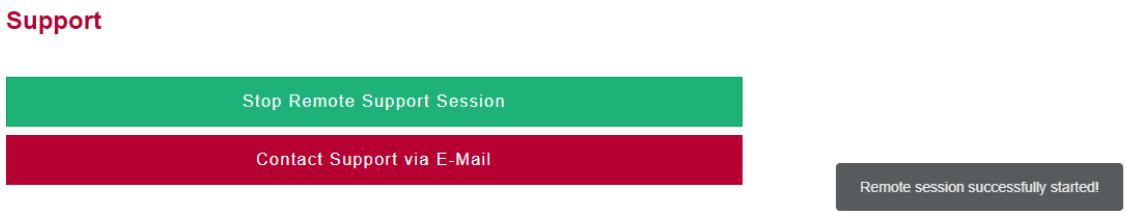


Figure 94: A Running Remote Support Session

8.2 AC Charging (PWM-based) Prototype

To use this feature you need a separate license. Please consult your Vector sales contact for details.

8.2.1 General Remarks

AC charging via Basic Signalling support is prototypical. It is possible to start a charging process by plugging in an EV, set the maximum charging current and stop the charging process via a stop button or MQTT (see Section 8.9 for details).
Other features known from DC charging are not supported yet, such as authorization mechanisms, charging profiles, native energy meter support and other CSMS-related features.

8.2.2 AC Maximum Power

AC charging via Basic Signalling according to IEC 61851-1 uses the duty cycle of the PWM signal of the Control Pilot line to communicate readiness and maximum charging current to the EV. This maximum current per phase is configured in the web configuration interface in the **Configuration / Vehicle** section or by setting the variable **ac_charging_setpoint_1/2** via the Provisioning Tool. Negative values are not allowed because the standard does not support discharging. Note: The EV is free to draw less power at any instant.

8.2.3 Power Electronics / Contactors Control

AC charging via Basic Signalling does not require a power electronics in the sense that a connection via PEP-WS or PEP-CAN is established to the PECC and target values are communicated.
Instead, only a very basic set of functionality is required: The vSECC Controller actuates one set of contactors per connector by setting a digital output pin HIGH or LOW. HIGH means ”close the contactors”, LOW means ”open the contactors”. On this control level, no distinction is made between 1 or 3 phase AC charging.
In addition to this control output, a feedback pin is read. This feedback is expected to follow the output within a maximum of 1000 ms. It is considered a severe fault if it does not follow. The charging process is then aborted, the control pin is set to LOW (i.e., contactors should open) and further charging is prevented by setting CP state F and the whole connector

inoperative. This is reflected on the respective MQTT topic, too.
Note that the safety requirements are met in the same way as with DC charging: The CP supervision output works for AC charging, too. See Section 2.2.6 for details.

8.3 ISO 15118-20 BPT Prototype

To use this feature you need a separate license. Please consult your Vector sales contact for details.

For performing bi-directional power transfer (BPT) as standardized in the ISO 15118-20, the following prerequisites must be met:

- A license for ISO 15118-20 and BPT must be installed on the vSECC Controller
- ISO 15118-20 must be selected in the **Configuration / Vehicle / CCS Charging Standards** in the web interface
- The power electronics must be able to discharge
- The PECC must implement the new fields in the `response-configuration` and `info-chargingSession` messages for PEP-WS or the `PeccLimits` and `Charging-SessionInfo` messages for PEP-CAN
- An EV supporting ISO 15118-20 BPT with TLS 1.2 / TLS 1.3 (certificates required) or without TLS must be available
- A CSMS connection with OCPP 1.6 or 2.0.1 should be used, if you want to remotely control the BPT setpoint (charge/discharge power)

8.3.1 BPT Dynamic Setpoint

In BPT dynamic control mode (scheduled mode currently not supported), the vSECC Controller instead of the EV computes the target values for how much power is charged/discharged to/from the EV. This value can be influenced by a CSMS via the `V2XCharging-Ctrlr/Setpoint` (OCPP 2.0.1) device model variable.
In OCPP 1.6 the variable is called `V2XChargingCtrl/1/Setpoint/` for the first connector. **Please note** the missing ”r” of ”Ctrlr”.
Negative values will represent discharging. The value will persist through restarts of the vSECC Controller.
The vSECC Controller will then send `request-targetValues` messages to the PECC based on the setpoint and the limits given by the EV and the PECC.
The setpoint’s unit is watts.

8.3.2 Power Electronics

The power electronics must have the capability to discharge the EV. Furthermore, the PECC must report the corresponding charge and discharge limits via the `request-configuration` or `PeccLimits` messages.
In a scenario where the PECC takes control of the charging/discharging process, the target values provided by the vSECC Controller can be ignored. The vSECC Controller will not check if the target values are applied or if the EV limits (provided in the `info-chargingSession` or `ChargingSessionInfo` messages) are respected, this is the responsibility of the PECC.
To determine if the connected EV is in dynamic control mode, the `chargeMode` field of the `info-chargingSession` /`ChargingSessionInfo` message contains the `dynamicBpt` value. In scheduled control mode (only for ISO 15118-2 and DIN SPEC 70121 at the moment), the target values are coming from the EV and must be respected.

8.4 CHAdeMO Support (vSECC only)

This section provides an overview of the CHAdeMo charging sequence, the currently supported features and safety considerations.

Caution: The current support for CHAdeMO charging is experimental. Thus, unexpected behavior may occur and safety mechanisms could not work properly. Use this feature with caution.

CHAdeMO is not yet supported on the vSECC.single.

8.4.1 General Behavior of a CHAdeMO Charging Station

A charging station supporting CHAdeMO has at least three buttons: The **[START]**, **[STOP]** and **[EMERGENCY STOP]** button.

If an EV connects to the station, the charging process is not started automatically. Although the station detects the EV's proximity, the charging process initialization commences only after the **[START]** button has been pressed.

The **[STOP]** button could be pressed anytime. This is intended as a graceful stop of the charging process. The two buttons **[START]** and **[STOP]** are connected through the X306 connector. See Section 2.2.8 for the correct pin usage.

The **[EMERGENCY STOP]** button is not connected to the vSECC itself, see the following section 8.4.2 for more details.

8.4.2 Safety Considerations, Emergency Stop Button

On a very coarse level, two safety mechanisms must be supported for CHAdeMO charging: The latch locking and the emergency stop.

Latch Locking: The latch locking is a means to secure the charging plug to the socket of the EV. It is implemented by a mechanism that locks the plug physically in place. This connector lock is controlled and monitored by the charging station.

Emergency Stop: Pressing the **[EMERGENCY STOP]** button must lead to a fast shutdown of the whole charging process, especially of the power supply. This shutdown must occur within tight timing boundaries.
Therefore, the **[EMERGENCY STOP]** must be connected directly to the power electronics. In case of a button press, the PE handles the shutdown of the power supply by itself and informs the vSECC of this incident through the Power Electronics Protocol (PEP). This is done by sending a PEP `status message` with the value set to INOPERATIVE to the vSECC. The vSECC then stops the charging process communication and disables the respective output (i.e., the EVSE that belongs to the output that has been shut down is rendered inoperative, too. This inoperative state is then communicated to the CSMS.).

This state exists until the power electronics sends another `status message` containing the OPERATIVE value. This indicates that the PE is able to provide power again.

8.5 Inverted Pantograph Support (vSECC only)

This section provides an overview of the Inverted Pantograph feature and pantograph control.
The vSECC supports both OppCharge and SAE J3105 if a license for Inverted Pantograph has been obtained.
Which communication standards are offered to the vehicle can be configured via the web interface (**Configuration / Vehicle / Inverted Pantograph Charging Standards**).
Inverted Pantograph is only available on charging port 1, which means that the control pilot line needs to be connected to the corresponding pin (see Section 2.2.4) and the Wi-Fi access point for vehicle communication needs to be connected to Ethernet port ETH1 (see Section 2.2.10).

Caution: The current support for OppCharge and SAE J3105 charging is experimental. Thus, unexpected behavior may occur and safety mechanisms may not work properly. Use this feature with caution.

Inverted Pantograph is not yet supported on the vSECC.single (Board).

8.5.1 Pantograph Control

The pantograph is controlled via four signals: One output signal to request changes of the pantograph position and three input signals to inform the vSECC about the state of the pantograph.

- `panto_control (output)`: This signal is used to request pantograph movement: `0` for up, `1` for down.
- `panto_up (input)`: This signal, when set to `1`, informs the vSECC that the pantograph is in its retracted endposition.
- `panto_down (input)`: This signal, when set to `1`, informs the vSECC that the pantograph is in its extended endposition.
- `panto_error (input)`: This signal, when set to `1`, informs the vSECC that charging is not allowed (e.g.: vehicle not in position, wind speed limit exceeded, mechanical problems, …).

If both `panto_up` and `panto_down` are set to `0`, the vSECC assumes that the pantograph is currently moving. A charging session can only be started if the pantograph is in its upper endposition and no pantograph error is signaled.
There are three possibilities to configure pantograph control via the web interface. The parameter can be found under **Configuration / Vehicle / Inverted Pantograph Control**

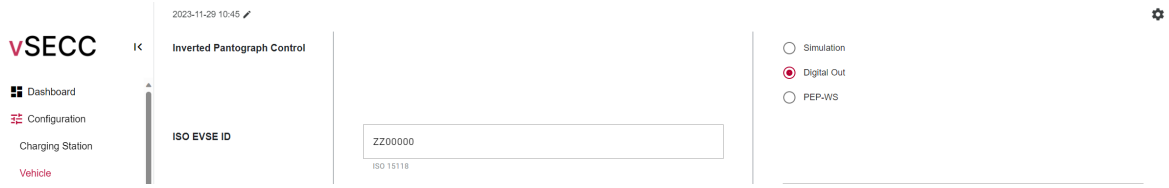


Figure 95: PantographControlType

- **Simulation:** For development only. The pantograph is simulated internally: it moves down during `CableCheck` and up again during `SessionStop`.
- **Digital Out:** The pantograph is controlled via digital inputs and outputs of the vSECC. Refer to Section 2.2.8 for wiring information.
- **PEP-WS:** The pantograph is controlled via virtual inputs and outputs of the PEP-WS protocol (this option is not available via PEP-CAN). The `getInput` message must be sent regularly with the `panto_control` identifier to get notified about requests to move the pantograph. Use the `setOutput` message to inform the vSECC about changes regarding the state of the pantograph. Make sure to send all three input signals at startup, so the vSECC knows the state of the pantograph. Refer to Section 8.18.9 for details about the virtual input/output identifiers.

8.5.2 RFID Pairing

SAE J3105 has native support for RFID pairing: The communication will loop in the Authorization stage until the pairing has been confirmed or rejected.
The implementation of the RFID reader has to be done on a customer device or could be implemented via the Configurable Customer Interface.
The matching has to be done by the customer based on e.g. the EVCC-ID that is transmitted via MQTT or PEP-WS. The result of the matching has to be provided via the `rfid_pairing` MQTT topic (refer to Appendix J).
The values of the `rfid_pairing` topic have the following semantics:

- `PENDING`: No vehicle detected / Pairing in progress.
- `OK`: Pairing successful, reset to `PENDING` after vehicle has left.
- `FAILED`: Pairing failed, reset to `PENDING` after vehicle has left.

OppCharge has no native support for RFID pairing, but there are two ways to achieve a similar behavior.

1. Enable **use_rfid_pairing_for_oppcharge** in the vSECC configuration. This enables the same RFID Pairing mechanism described above also for OppCharge. But you need to make sure to publish the pairing result before CableCheck to prevent movement of the pantograph.
2. Set the `panto_error` signal high as soon as the RFID pairing has failed

8.5.3 SAE J3105

The current prototypical implementation of the SAE J3105 feature has the following limitations:

- Response Code `FAILED_RFIDPairing` not supported yet, currently the vSECC just sends `FAILED`.
- No support of prioritization. Vehicle is always prioritized.
- No support of ”Offline behavior” as specified in SAE J3105.

8.6 Value Added Services

The vSECC Controllers support internet access via value-added services (VAS) as specified in the ISO 15118 standard. The service ”internet access” is announced in the service discovery phase of the charging session. The vSECC Controller by itself does not provide any VAS-back ends/endpoints. Therefore, it only acts as a router between the EV and an existing VAS-Endpoint. The VAS-backend must be connected to the ETH1 network interface. (see Section 2.2.10) See the following enumeration for a brief overview over the prerequisites. For further information consult the subsequent description.

Caution: The vSECC Controller only supports IPv6 routing. IPv4 is not supported.

8.6.1 Prerequisites

> Customer provided VAS-Back End
> VAS is activated on the vSECC Controller using the web interface or a CSMS
> Existing IPv6 route between the vSECC Controller and the VAS-Back End
> TLS encrypted V2G communication between EV and vSECC Controller according to ISO 15118
> The EV obtains an IPv6 address according to ISO 15118 utilizing SLAAC and NDP
> The vehicle connects to the VAS-Back End via IPv6

Caution: No DNS is provided.

8.6.2 Networking

The vSECC Controller should be connected to a network with an IPv6 capable router. This router should provide an IPv6 prefix to the vSECC Controller via NDP router advertisement. Using this address, the VAS-Endpoint can be reached by the vSECC Controller.

8.6.3 Activate VAS via Web Interface

Value Added Services can be toggled under **Configuration / Vehicle / Value Added Services**). Also make sure that **Transport Layer Security** is enabled.

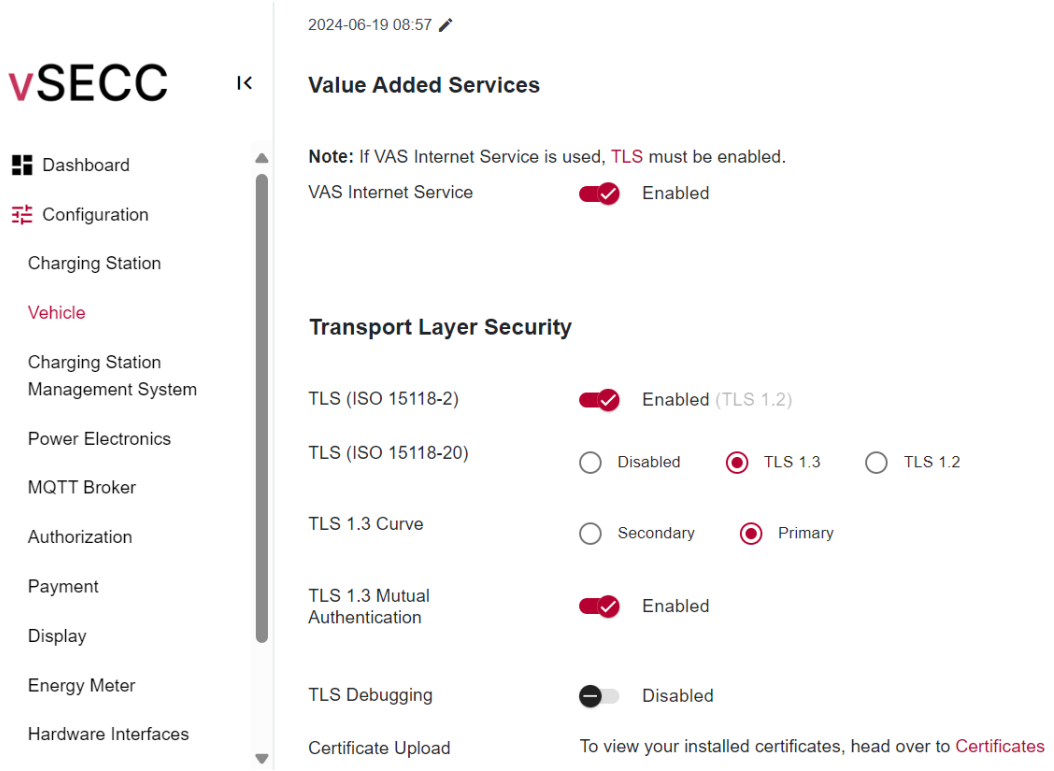


Figure 96: Value Added Services and Transport Layer Security activation via Web Interface

8.6.4 TLS certificate for V2G

An encrypted high level communication is mandatory to offer the internet access services as well as the Plug & Charge (PnC) authorization service (see ISO 15118-2). For this to work, a certificate chain with the public key and the corresponding private key needs to be installed on the vSECC Controller. The installation of the certificate chain and key is possible in two ways:

> Via OCPP using a CSR.

> Via the Web Interface by uploading a certificate chain and private key file. This process is described in Section 7.6.1. Select **EVSE Leaf Certificate** from the drop-down menu when pressing **[Upload Certificates]**. Both the certificate chain (including the public key) and private key must be given in (human-readable) PEM format. The certificate chain has the following structure: The most specific certificate, i.e. leaf certificate of the EVSE, is placed first, followed by other certificates higher up in the hierarchy of CAs. Please see Figure 65 in chapter 7.6.2.

8.7 Transport Layer Security (TLS)

8.7.1 TLS Communication

Certain charging standards support encryption of the high-level charging communication (HLC). Both the SECC and EV must support the respective charging standard and the corresponding TLS version. ISO 15118-2 allows either no encryption or TLS 1.2. ISO 15118-20 requires TLS 1.3 on both sides. Several configuration variables exist that control the vSECC Controller's behavior regarding this encryption. They can be set in the Web Interface section **Configuration / Vehicle / Transport Layer Security** as shown in Figure 96.

TLS HLC can only be established when the respective EVSE leaf certificate chain is installed on the vSECC Controller. See Section 7.6.2 for details on how to install a certificate.

Installed TLS 1.2 certificates cannot be used for TLS 1.3 connections.

The following configuration variables influence charging according to ISO 15118.

TLS (ISO 15118-2) `(charging_communication_ctrlr_tls_enabled)` When enabled, TLS is offered as an option to the EV when charging according to ISO 15118-2. A TLS communication is established if the EV wants to use TLS, the EV has the respective V2G Root CA certificate installed and the EVSE certificate chain is valid.

TLS (ISO 15118-20) `(iso_20_tls_version)` Defines which encryption is used when charging according to ISO 15118-20. The standard mandates that TLS 1.3 shall be used.

Other options supported by the vSECC Software are **Disabled** (meaning no TLS connection is used for ISO 15118-20 charging) and **TLS 1.2**.
The respective certificate chain for the TLS version is used. Not using TLS 1.3 is intended for testing purposes only and should not be used for production!

TLS 1.3 Curve `(iso_20_tls_1_3_elliptic_curve)` ISO 15118-20 requires that the SECC shall be able to switch the elliptic curve used for TLS 1.3 connection via a configuration option. Note that a TLS 1.3 certificate chain is curve-specific. In order to switch between primary and secondary curve, a certificate chain must be installed for the selected curve.

TLS 1.3 Mutual Authentication `(iso_20_use_mutual_authentication)` When enabled, mutual authentication is used for ISO 15118-20 charging and the EV is requested to present its own certificate chain at the TLS handshake.
This means that not only the SECC certificate chain is checked against a Root CA certificate installed on the EV, but also the EV-presented chain is checked against all Root CA certificates installed on the SECC. Zero or more of these *OEM Root Certificates* can be installed on the vSECC Controller.
When disabled, the EV is not asked for a certificate and no check occurs.
ISO 15118-20 requires mutual authentication (mTLS). Note that disabling mTLS is not standard-compliant and should not be used in production!

TLS Debugging `(security_ctrlr_tls_debugging_enabled)` Controls whether the *ClientRandom* and *MasterSecret* are logged. See Section 8.7.2 below.

8.7.2 TLS Debugging

Caution: This feature is only for debugging/development purposes and should never be enabled in a production environment.

This feature assists the debugging of TLS-encrypted communication. When enabled, the *ClientRandom* and *MasterSecret* values of the TLS connection are written into the Charging Manager log file. Furthermore, they are sent out as a UDP packet in the NSSKeyLog format. This information can be used to decrypt the PLC traces with e.g. CANoe or Wireshark.

TLS debugging is available for TLS 1.2 communication only.

8.8 Charging Schedules (Charging Profiles)

Since the available power for charging may change over time, charging schedules are advertised to the connected EV. These schedules can either be set to statically contain the power electronics static power limit, or to be computed from charging profiles provided by a CSMS via the OCPP interface. If configured to use charging profiles provided by a CSMS, the vSECC Controller will persist default charging schedules created from default charging profiles, and use these in case no transaction specific charging profile could yet be retrieved from the CSMS. When generating a charging schedule from charging profiles provided by the CSMS, gaps are filled with entries advertising the power electronics static power limit. The duration of charging schedules is defined by the departure time provided by the EV. Otherwise, it can be configured in the web interface **Vehicle / Expert Functions** by the configuration variable **Duration of SASchedule**, or using the Provisioning Tool variable **sa_schedule_duration_hours**. Scheduled power values by the CSMS are capped at the power electronics static power limit.

On the reception of a charging profile issued by the CSMS, the vSECC Controller behaves in the following way:

> During ISO 15118-2 charging, the renegotiation process is triggered and the new schedule is advertised to the EV. If the charging is paused, an EV wakeup is executed beforehand

> During DIN SPEC 70121 charging, the communicated EVSE power limit is adapted to the updated schedule values

> During CHAdeMO charging, the charging schedule is not communicated to the EV, the most recently created charging schedule is used to schedule charging power

> During Inverted Pantograph charging, the charging schedule is not communicated to the EV, the most recently created charging schedule is used to schedule charging power

> For AC charging, charging profiles are not yet considered

During a charging session, the vSECC Controller computes the target power from the following values and communicates it to the power electronics to adapt accordingly:

> The currently scheduled charging power

> The maximum power the EV reported to support

> The power electronics static power limit

For further details how a specific charging schedule is considered at derating, please refer to appendix H.

Charging Pause (Sleep) and Wake-Up

An EV with ISO 15118 support may request a charging pause which may enable the opportunity for power saving functionality. This is often the case if a charging schedule contains periods with 0 W scheduled power.

A charging session is paused by stopping it according to the procedure defined by the standard used for communication between EV and EVSE. In addition, one of the last messages sent by the EV triggers the pause sequence for both the vehicle and the EVSE. During a pause, the communication is reduced to a minimum by terminating the high-level communication session completely. The EV may decide to turn off the PLC device, too.

A previously paused charging session is resumed either by the EV or EVSE. This may happen due to the expiration of the sleep period or due to a new charging profile sent by the CSMS. The EV triggers the resumption by executing a BCB toggling sequence of the Control Pilot (CP) line. The EVSE detects this CP state change and sets the PWM duty cycle to 5% to trigger and force the setup of a high-level communication. The EV is then expected to resume the session by using the same SLAC key (NMK) as previously. In turn, the EVSE triggers the resumption by skipping the BCB toggling and directly setting the duty cycle accordingly. The setup of the high-level communication is the same as it would be without the pause period except for the session ID provided by the EV. Normally, this ID would be regenerated for each HLC setup. When resuming a paused session, though, this ID is expected to remain the same across the two V2G sessions. If the wake-up procedure does not complete in time, a reset of the EV is done. The EVSE executes a BEB toggling sequence to force the EV to restart the whole charging session initialization. This toggle can also be triggered by an external device using the MQTT interface (see ”Reinit EV to start charging” in Annex J). In this case the session ID is not expected to remain the same.

Communication of computed schedule via MQTT

If an entity connected to the vSECC Controller via MQTT or inside the Configurable Customer Interface is interested in the currently active charging schedule, it can subscribe to the Charging Profile Composite Schedule topic (see Annex J). It is published whenever a charging session is running and the schedule is updated.
The following example shows a typical scenario where no charging profiles are installed from the CSMS and only the power electronics static limit of 35kW applies. The `charging_profile_id` of ”-1” represents the power electronics static limit. The duration of the schedule covers the default setting of 48h (172800s).
The content of the topic then looks like this:

```
{
    "periods": [
        {
            "charging_profile_id": -1,
            "power_limit_watts": 35000,
```

```
            "start_seconds": 0,
            "stop_seconds": 172800
        }
    ],
    "schedule_start": "2024-03-27 11:27:39"
}
```

The contained JSON has the following format:

- **periods:** Array with one or more entries of the current charging schedule.
  - **charging_profile_id:** The ID of the CSMS charging profile from which this entry is generated. If the ID is ”-1”, it means that the power electronics static limit is applied.
  - **power_limit_watts:** The power limit applied in this period.
  - **start_seconds:** The start of this period relative to `schedule_start`.
  - **stop_seconds:** The stop of this period relative to `schedule_start`.
- **schedule_start:** Point in time when this schedule starts.

Automatic Deletion on CSMS Connection Loss

If enabled, all installed Transaction-specific charging profiles (with ChargingProfilePurpose set to *TxProfile*) get deleted automatically by the vSECC Controller as soon as the CSMS connection has been lost. Please note that charging profiles do not get reinstalled by the vSECC Controller after the connection has been restored. If this behavior is desired, the CSMS is expected to send the respective profiles again.
This automatic deletion is turned off by default. It could be turned on by setting the boolean configuration variable **delete_tx_profiles_on_connection_loss** accordingly.
The time that must pass between the minimum downtime up until the deletion takes place is configurable via the configuration variable **delete_tx_profiles_on_connection_loss_delay**. Its default value is 60 seconds.

8.9 Stop Charging

A running charging session can be gracefully terminated by the vSECC Controller using the following means:

- Physical button connected to the vSECC (not yet available on vSECC.single)
- Power electronics message via PEP-WS or PEP-CAN
- CSMS message
- MQTT topic publication

8.9.1 Physical Button (vSECC and vSECC.MCS only)

This feature can be enabled via the **Stop Button** configuration option for the respective connector in the Web Interface. For physical wiring details on connector X306, refer to Section 2.2.8 (vSECC) or Section 3.2.6 (vSECC.MCS).

8.9.2 Power Electronics

To stop charging via power electronics, use the `stopCharging` message via PEP-WS or PEP-CAN. Refer to the PEP manuals for details.

8.9.3 CSMS

A charging session can be stopped via CSMS by sending a `RequestStopTransactionRequest` message.

8.9.4 MQTT

Publishing to the MQTT topic `vsecc/connector/{evse_id}/preview/stop_charging` triggers a charging stop. The placeholder `{evse_id}` denotes the respective outlet ID. The actual topic for outlet 1 would look like this: `vsecc/connector/1/preview/stop_charging`. The message content is not used in any way, it is recommended to use the empty string.

8.10 Authorization

The vSECC Controller supports authorizing a transaction with multiple methods and options, which can be configured in the web interface section **Configuration / Authorization**. Besides automatic vehicle-based authorization (**MAC Address** and **EMAID**), three methods how a token can be presented manually exist and will be explained in the following sections:

- Via RFID from readers directly connected to the controller
- Via MQTT from an external MQTT client (e.g. from a system board where the RFID reader is connected)
- Via OCPP messages RequestStartTransactionRequest (OCPP 2.0.1) and RemoteStart.req (OCPP 1.6) from the CSMS.

Figure 97 shows the possible configuration options. **Authorization** can be disabled completely, which skips token validation during transaction setups. The **Authorization Options** select which token types are allowed. They are mostly aligned to the token types defined in OCPP 2.0.1.

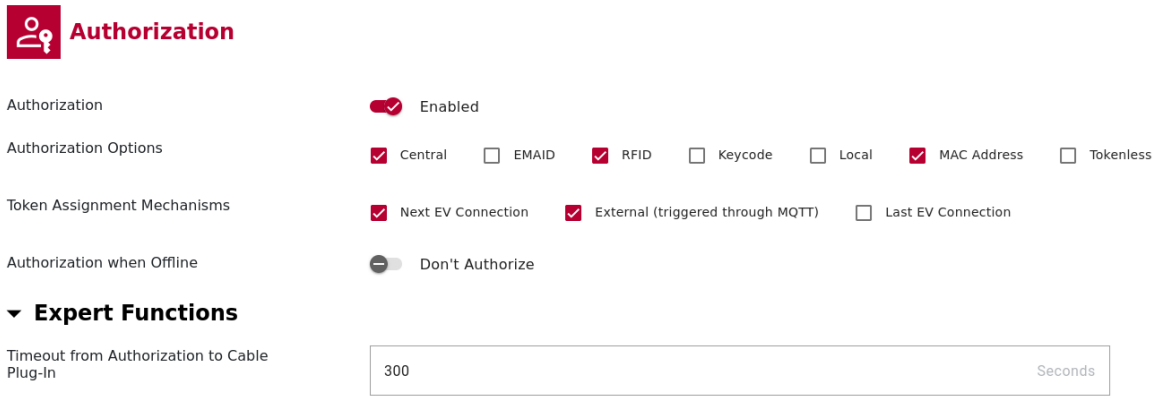


Figure 97: Authorization Settings in the Web Interface

The processed tokens themselves also follow the structure of OCPP 2.0.1's `IdTokenType` and can either be initially unassigned or already assigned to a particular EVSE. Multiple assignment mechanisms are supported to customize the authorization sequence to preference.

8.10.1 Authorization Sequence

The vSECC Controller supports flexible authorization sequences configurable in the web interface as seen in figure 97. Three **Token Assignment Mechanisms** can be selected. They define how tokens are assigned to EVSEs. Figure 98 shows internal decisions when a token assignment is attempted.

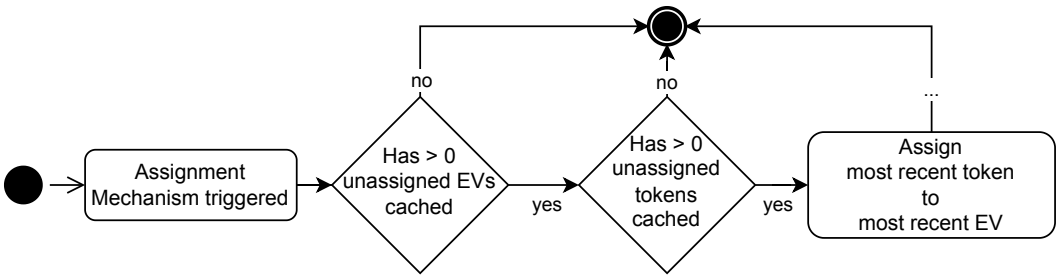


Figure 98: Token Assignment Decision Flow

It is possible to use multiple or all mechanisms at once. Consider that **Last EV Connection** and **Next EV Connection** are automatically evaluated. They will assign a token at the soonest opportunity, possibly before an **External (triggered through MQTT)** assignment is received.

Next EV Connection targets the situation when a driver first presents his RFID token and then plugs in the vehicle. This mechanism foresees that the presented token will be assigned to the EVSE with the next EV connection. The driver must then connect the vehicle at one of the available EVSEs within **Timeout from Authorization to Cable Plug-In** sec-

onds. The timeout can be set in the web interface section **Configuration / Authorization / Expert Functions**. If no EV gets connected within the timeout, the pending token is discarded. Figure 99 shows what happens when an EV is connected in this configuration.

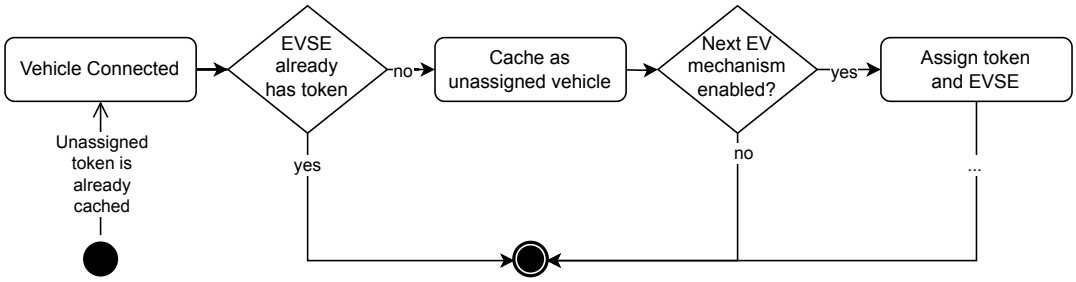


Figure 99: Next EV Connection Assignment Flow

If multiple tokens are presented before any EV is connected, the assignment uses the most recent token. The next EV and its EVSE would be assigned to the second most recent token. This can be seen in figure 100, where two tokens are presented and validated by the CSMS before the first EV is connected.

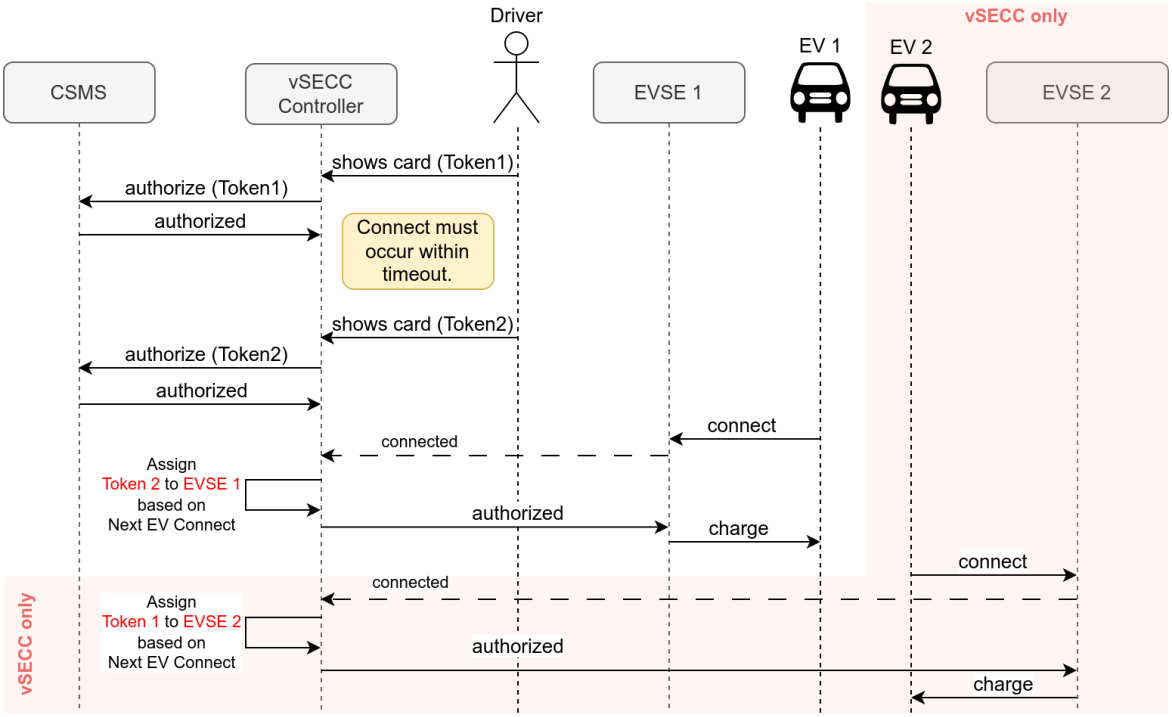


Figure 100: Next EV Connection with Multiple Tokens Queued

Last EV Connection targets the situation when a driver first plugs in the vehicle and then presents his RFID token. This mechanism assigns the token to the EVSE with the most-recent EV connection. If multiple EVs are connected but not yet authorized, the token will be assigned to the EVSE with the most recent connection. The next token will be assigned to the second most recent connection. If no EV is already connected and no other mechanisms are enabled it will not be possible to assign this token. Figure 101 shows what

happens when a token is presented in this configuration.

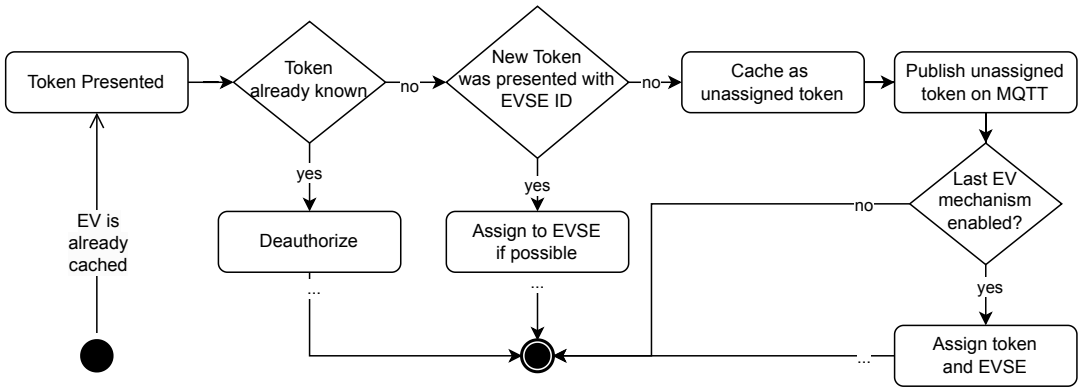


Figure 101: Last EV Connection Assignment Flow

The user may try again after either manually deauthorizing their token by presenting it a second time or by waiting until the **Timeout from Authorization to Cable Plug-In** expires. This scenario is shown in figure 102.

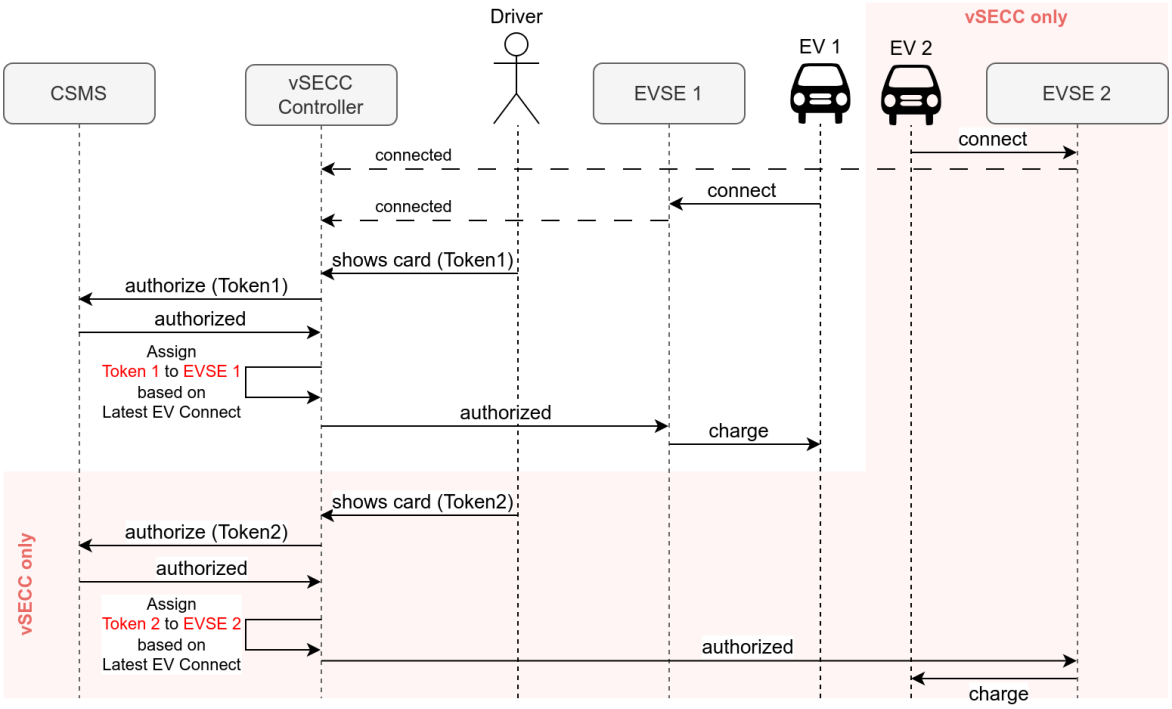


Figure 102: Latest EV Connection with Multiple EVs Queued

The third option is **External**, which assigns tokens to EVSEs based on requests sent over MQTT. This allows the most explicit control over the assignment for operators and is detailed in 8.10.5.

8.10.2 Relation to OCPP Transactions

OCPP transactions are significantly connected to the authorization sequence. Depending on the start points and assignment policies configured, as seen in Figure 103, your chosen authorization sequence may lead to multiple transactions being started.

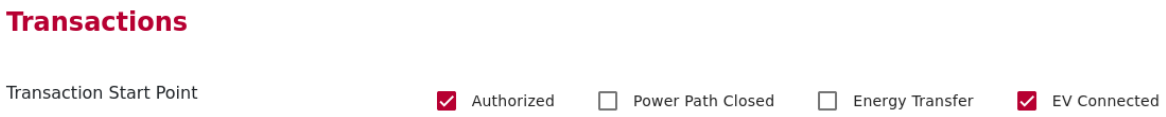


Figure 103: Authorized and EV Connected Configured as Transaction Start Points

Two transactions cannot be merged into a single transaction. For example, consider the following sequence with TxStartPoints ”EVConnected” and ”Authorized” configured and the usage of external token assignment:

> Driver connects EV at EVSE 1. Transaction TX1 starts because start point ”EVConnected” is triggered.
> Driver presents token at RFID reader. Token gets authorized by the CSMS. Transaction TX2 starts because start point ”Authorized” is triggered.
> Driver selects EVSE 1 through external token assignment after authorization, e.g. by using the charging station’s UI. **This will not be allowed by the vSECC Controller because the EV and token are already assigned to two independent transactions TX1 and TX2**.

This issue can be circumvented by utilizing a restricted set of TxStartPoints (e.g., exclusively ”Authorized,” ”EVConnected,” or ”PowerPathClosed”) or by using the token assignment policies ”Last EV Connection” or ”Next EV Connection” instead of external assignment.

It is not recommended to combine start points **Authorized** and **EV Connected**.

8.10.3 Related MQTT Topics

Multiple MQTT topics related to authorizations exists and are detailed in appendix J.

> `add_token`: Present a token to the vSECC Controller.
> `assign_token`: Assign a token to an EVSE at the vSECC Controller.
> `charging_authorization_state`: EVSE specific authorization state.
> `ev_authorization_token`: Token identifier that was previously assigned to the specific EVSE.
> `authorization_token_status`: Meta information for token.
> `unassigned_tokens`: Currently unassigned tokens.

8.10.4 Local Authorization via RFID

Tokens can be presented via RFID to authorize transactions. Additionally, the transaction can also be stopped by presenting the same RFID token which started the transaction. By the RFID Reader detected token identifiers are published on MQTT and can be used e.g. to show on the display that the token was successfully read. To avoid repeated MQTT publishments of the same token, ongoing detections of the same tag are ignored until a different tag is presented or the tag was removed from the reader.

If you want to make use of RFID authorization, make sure to add ”RFID” as an authorization option in the web interface section **Configuration / Authorization**. You may also turn off other authorization options such as **MAC Address** if you want to authorize only using RFID tokens. Assignment mechanisms must be chosen according to your preferred authorization sequence. Without assignment, the RFID tokens can not be used to authorize for charging.

Supported RFID Readers

At this moment, two RFID readers are supported to directly interface to the vSECC Controllers. Furthermore, custom RFID Readers can be integrated via the Configurable Customer Interface. For more details, see chapter 8.27. The directly supported devices are MCRN2 of Minova GmbH [1] and TWN4 Multitech (2) Series of Elatec GmbH [2].

Both readers support RS232 and can be connected to the corresponding pins of the vSECC Controller (for vSECC see X305 in Section 2.2.7; for vSECC.single Board X300 in Section 4.2.5). An example circuit for connecting an RFID reader with the vSECC Controller is shown in Appendix F).

To configure the vSECC Controller to use the connected reader, the RS232 interface needs to be configured in the web interface section **Configuration / Hardware Interfaces**. Possible values to use the Plug & Play RFID Readers are **Minova** and **Elatec**. If a custom RFID Reader should be used with the Configurable Customer Interface, the RS232 interface needs to be passed into the **Container**.

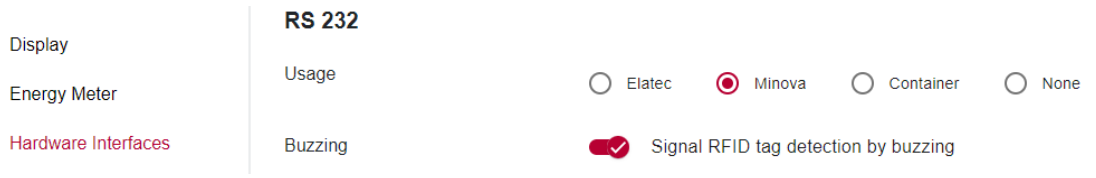


Figure 104: RS232 Interface Settings in the Web Interface

Before usage, the TWN4 Multitech reader has to be flashed with the right firmware set. These are provided by Elatec in their ”DevPack” [3]. Compatibility was tested with:

[1] https://minova-rfid.com/en/mcrn2-nfc-rfid-rs232.html
[2] https://www.elatec-rfid.com/int/product-detail/twn4-multitech-2
[3] https://www.elatec-rfid.com/int/twn4-dev-pack

> ”TWN4_xCx450_STD204_Multi_CDC_Standard.bix”

The firmware can be flashed onto the TWN4 reader with the ”AppBlaster” tool, also included in Elatec’s DevPack.

For the Minova MCRN2 reader it is possible to configure if the reader should play an acoustic signal (buzzing) after detecting an RFID tag. Therefore, enable the **Buzzing** as shown in 104. The Minova buzzing is suppressed like the MQTT publishments for repeated detections of the same tag. Other configuration is not necessary.

Deauthorization

If a driver has previously authorized and started a transaction using an RFID token and presents the same token again, the vSECC Software will deauthorize that token and stop the charging sequence. If the token was authorized but no transaction started, presenting the token a second time will also deauthorize the pending token.

8.10.5 External Authorization via MQTT

To support a wider range of RFID readers and other authorization hardware, the MQTT interface can be used to pass tokens. The MQTT client must reside in the Configurable Customer Interface. Hence, the tokens are also forwarded in the Container to the vSECC Controller’s MQTT interface. For more details about the Configurable Customer Interface, see chapter 8.27.
These tokens can already be preauthorized by the custom hardware, or they are being authorized by the CSMS. Preauthorized tokens should only be used if the CSMS connection is completely disabled. Online transactions with preauthorized token may run into different issues. For instance, transactions initiated with preauthorized tokens may be rejected immediately by the CSMS after resolving a temporary connection loss.
All token types available in OCPP 2.0.1 can be used for the external authorization. If no CSMS is used, only preauthorized tokens or of type ”NoAuthorization” can be used. See section J for a detailed description of the MQTT topic.

Figure 105 visualizes how the `add_token` MQTT topic can be used to authorize and deauthorize an ID token. If **External (triggered through MQTT)** is chosen as assignment mechanism, the MQTT topic `assign_token` must be used. Otherwise the authorization sequence must include connecting a vehicle to trigger **Next EV Connection** or **Last EV Connection** assignment. The figure additionally shows how the charging station can gather information about the authorization status by using the `unassigned_tokens`, `charging_authorization_state`, `ev_authorization_token` and `authorization_token_status` MQTT topics.

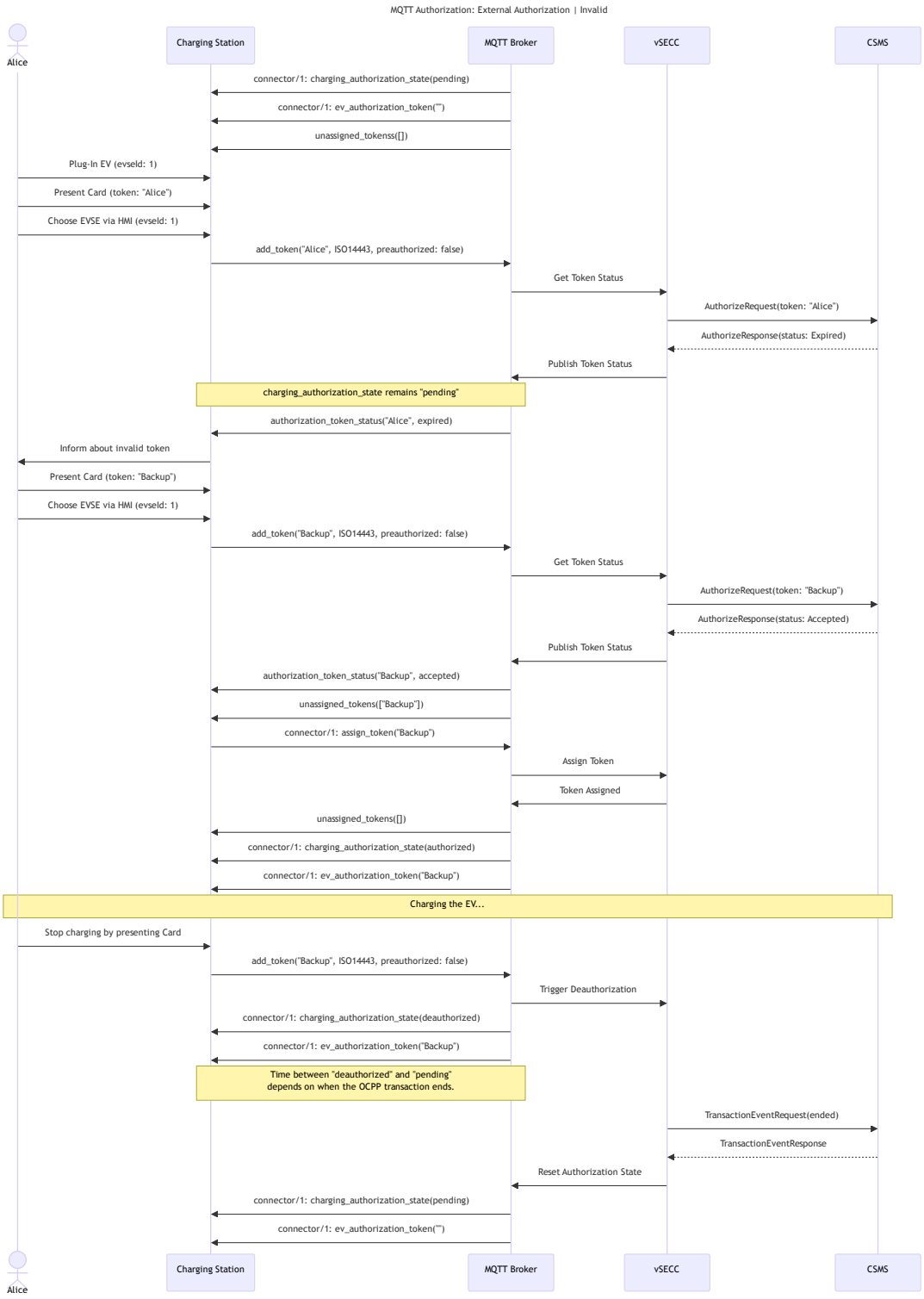


Figure 105: External Authorization via MQTT

8.10.6 Remote Token Presentation via CSMS

Tokens can be presented by the CSMS via RequestStartTransactionRequest in OCPP 2.0.1 and RemoteStart.req inOCPP 1.6. These messages may include an EVSE ID. If the ID is omitted, the token must be assigned via the configured assignment mechanism similarly to tokens presented via RFID or MQTT. Otherwise, if the message contains a particular EVSE ID, the token is immediately assigned to this EVSE.

8.10.7 MAC Address Authorization

OCPP 1.6 does not consider different authorization token types like RFID or MAC address. To help differentiating those types, it is possible to configure the vSECC Contoller to add a prefix ”VID:” to a MAC address token. This behavior can be toggled in the web interface under **Configuration** / **Authorization** / **Expert Functions** if ”MAC address” is selected as **Authorization Option** and OCPP 1.6 is configured for CSMS communication (see 7.3.2).

8.11 Plug & Charge (PnC)

To use this feature you need a separate license. Please consult your Vector sales contact for details.

8.11.1 Preconditions

The following preconditions must be met for the vSECC Controller to use PnC functionality:

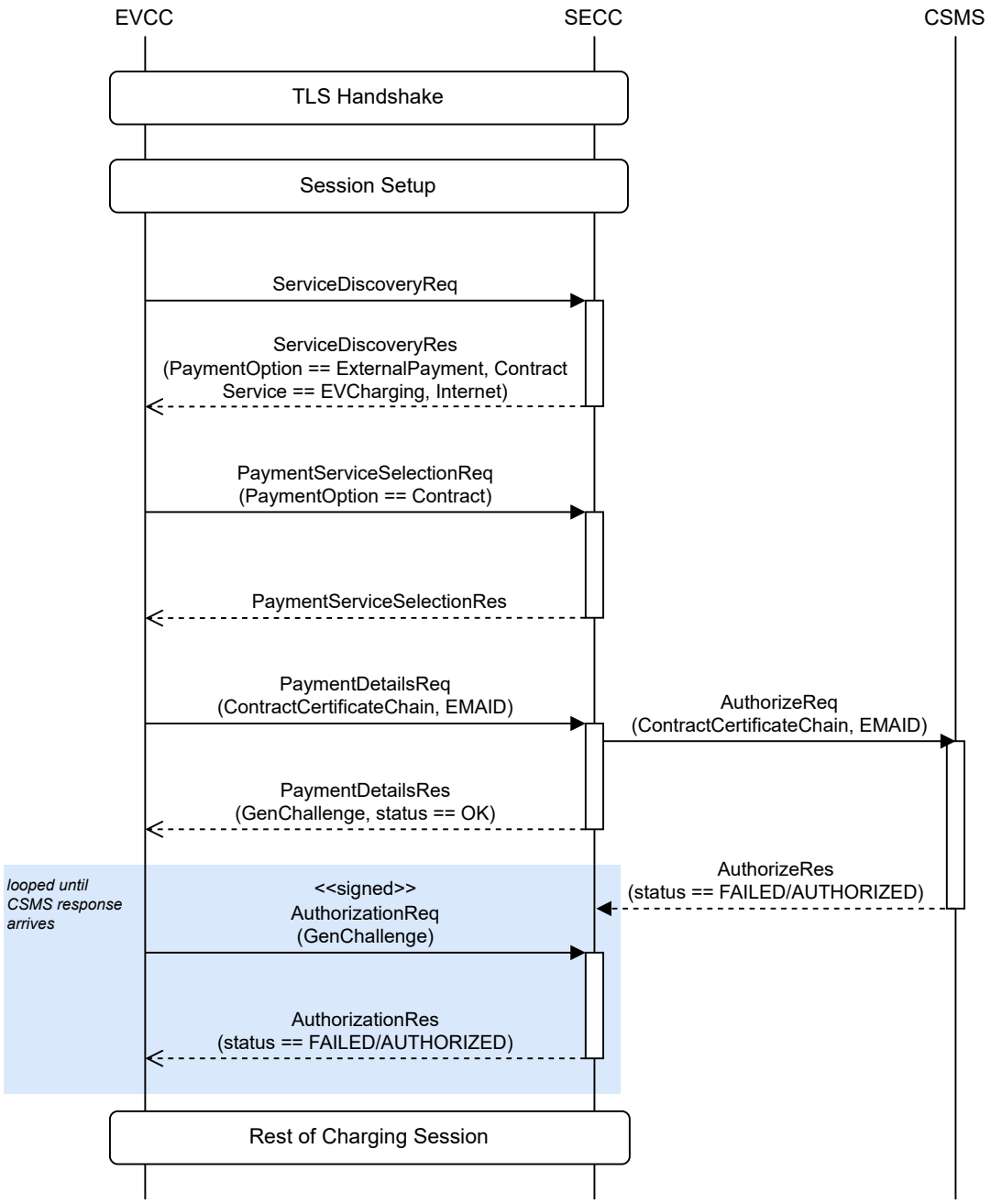
- The license installed on the vSECC Controller supports PnC (see Section 7.8.5 on how to install a license).
- PnC functionality is enabled in the web interface in Section **Configuration / Vehicle / Plug & Charge**.
- Usage of the ISO 15118-2 charging standard is enabled in the web interface in Section **Configuration / Vehicle / CCS Charging Standards**. In addition, ISO 15118-2 is selected as charging standard by the EV.
- TLS is enabled for charging sessions in the web interface in Section **Configuration / Vehicle / Transport Layer Security**. In addition, TLS is selected as security option by the EV.
- Communication to a CSMS is enabled in the web interface in Section **Configuration / Charging Station Management System / Connection**. The connection to the CSMS is established.
- EV Authorization is enabled in the web interface in Section **Configuration / Authorization** and **EMAID** is selected in the **Authorization Options**.
- No other authorization process is completed (e.g. via Autocharge or RFID). PnC is offered as payment/authorization option to the EV only if the authorization is still pending. If another authorization option has successfully completed beforehand, only EIM is available and PnC is not advertised to the EV. If another authorization option resulted in a negative response (deauth/blocked), PnC is also not advertised.
- A signed and valid V2G EVSE leaf certificate with its corresponding certificate chain and the according private key is present on the vSECC Controller, see Section 8.6.4.

In addition, the EV must support ISO 15118-2 Plug & Charge and have the according root certificate installed for verification of the EVSE certificate.
Furthermore, the CSMS must have the ability to verify the contract certificate which is provided by the EV and forwarded by the vSECC Controller.

8.11.2 PnC communication sequence

The good case scenario of charging with authorization via Plug & Charge contains the following steps (some common steps are omitted for simplicity). Refer to diagram 8.11.2 for a visualization.

1. The EV is plugged in, performs the SLAC process and a PLC link is established.
2. The EV sends a `SECCDiscoveryReq` message which indicates support of TLS usage, the vSECC Controller also indicates support of TLS in the response message.
3. A TLS 1.2 connection is established between the EV and the vSECC Controller, the EV checks the EVSE leaf certificate in the process (”TLS handshake”).
4. The EV sends a SupportedAppProtocolRequest which states support of ISO 15118-2 charging communication as preferred choice.
5. The vSECC Controller responds with the confirmation of charging according to ISO 15118-2 (”Session Setup”).
6. The EV queries the EVSE for its available services in the ServiceDiscovery phase. The vSECC Controller responds with PnC (”Contract”) as one of the available payment options.
7. During PaymentServiceSelection the EV selects PnC as payment and thus authorization mode.
8. The EV sends its contract certificate and the corresponding certificate chain in the `PaymentDetailsRequest` message. The vSECC Controller forwards the contract certificate to the CSMS, which verifies the certificate.
9. In the `PaymentDetailsResponse` message the vSECC Controller asks the EV to sign a challenge value (called ”GenChallenge”) as proof that it is the owner of the contract certificate.
10. The EV sends the signed challenge value to the vSECC Controller in the AuthorizationRequest. The vSECC Controller then verifies the signature of the challenge with the public key of the EV it received in the PaymentDetailsRequest.
11. The Authorization messages are repeated until a response from the CSMS arrives. Upon a positive response from the CSMS, the vSECC Controller continues the charging communication by responding with a positive `AuthorizationResponse` and the charging process is started.
12. The charging session is carried out normally.



8.11.3 Error cases

The following events prevent or abort a charging session using authorization via PnC:

The EV is already authorized via other means before the charging session reaches authorization mode selection In this case authorization was already done via external identification means (EIM). Then, authorization via PnC is not offered to the EV but only EIM. This is to prevent unintentional authorization via PnC if the user already chose another method.

The EV does not support TLS In this case authorization via PnC is still offered to the EV. However, the EV does not support and thus not select PnC/Contract as payment option. Instead, the EV selects EIM (External Identification Means). If a positive authorization via EIM is provided, the charging session may continue. Otherwise, the charging is aborted. The vSECC Controller offers EIM in every case, regardless of the PnC settings and preconditions.

The EV selects EIM instead of PnC In this case authorization via PnC is offered to the EV. However, the EV does not select PnC/Contract as payment option. Instead, the EV selects EIM (External Identification Means). If a positive authorization via EIM is provided, the charging session may continue. Otherwise, the charging is aborted. The vSECC Controller offers EIM in every case, regardless of the PnC settings and preconditions.

The EV provides a signed challenge that cannot be verified In this case the vSECC Controller assumes that the EV does not have the private key to the contract certificate it offered. Therefore authorization via PnC is considered to have failed and the vSECC Controller will terminate the charging session immediately. Afterwards, the EV may perform a BCB toggle to reestablish a new session.

The CSMS reports that the contract verification failed In this case authorization via PnC is considered as failed and the vSECC Controller will terminate the charging session. With ISO 15118-2, no other authorization means are carried out after a negative authorization response. The EV may perform a BCB toggle to reestablish a new session.

No connection to the CSMS is established In this case authorization via PnC is not offered to the EV. Authorization with other external identification means is still possible. Note that these other external identification means may have preconditions not described here. See the respective sections, e.g. Section 8.10.4 for RFID.

8.11.4 Limitations

Currently, the implementation of PnC has the following limitations:

- ISO 15118-20 is not supported.
- Contract certificate installation or update on the EV is not supported. It is not advertised as available service in the ServiceDiscovery phase.
- SalesTariffs according to ISO 15118 are not supported.
- The vSECC Controller does not send energy meter information to the EV nor does it verify the signed response from the EV (MeterInfo record and MeteringReceiptReq/Res handling).

> Offline contract validation: The contract certificate provided by the EV is not validated on the vSECC Controller (offline validation). Instead, it is forwarded to the CSMS (online validation).

8.12 Usage of Payment Terminals

The vSECC Controllers support the usage of so-called Cloud-based Payment Terminals. There is no direct interface between the payment terminal and the vSECC Controller. Instead, the Payment backend and the OCPP backend are interfacing. The concept is shown in the following picture.

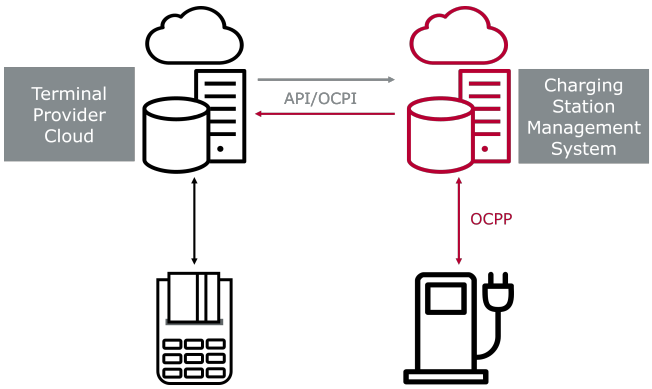


Figure 106: Communication Interface with Payment Terminals

Still, it is important to show the EV driver the tariff and cost of charging e.g. on a display. How this can be achieved is described in the following.

8.12.1 Tariff and Cost

The tariff and cost implementation is based on the OCPP & California Pricing Requirements. A tariff always concerns the whole charging station and is structured like ”**1.23 EUR/kWh**”, i.e., a **value** followed by **currency** and the unit ”**/kWh**”. Specific tariffs for individual customers, vehicles, charging times, idle fees, reservation fees or connectors are not supported, yet. Still, these mixed tariffs could be realized (not in the case of Eichrecht), since the CSMS is calculating the final cost and could include such a tariff structure in the calculation. Anyhow, these mixed tariffs would not be displayed to the driver through the vSECC Controller. With OCPP 2.0.1, the charging station tariff can be set in the web interface in Section **Configuration / Payment / Tariff** via the configuration variable **Tariff Fallback Message** as shown in 107. The configuration value is published on the MQTT bus during start-up of the vSECC Controller and every time it is updated by the CSMS. Two flavors of the same information are published: The raw `TariffFallbackMessage` without any further processing (MQTT Topic *vsecc/tariff_raw*) and the tariff in processed form (MQTT topic *vsecc/tariff*). This processed form is necessary to be compliant with the restrictions set by the German Eichrecht for ad-hoc payment, since the tariff of a charging session must be part of the data signed by the energy meter.
The vSECC Software uses a Regular Expression (regex) to process/extract the tariff from the full message. The extracted tariff must match the structure mentioned above, with a

dot-separated decimal value with 2 decimal places at most. This way, the charging station can inform the driver with an everyday language message and provide a machine-readable tariff to the energy meter at the same time. The Regex may be configured in the same web interface section in the **Expert Functions** via the **Regular Expression for Tariff Extraction** field.

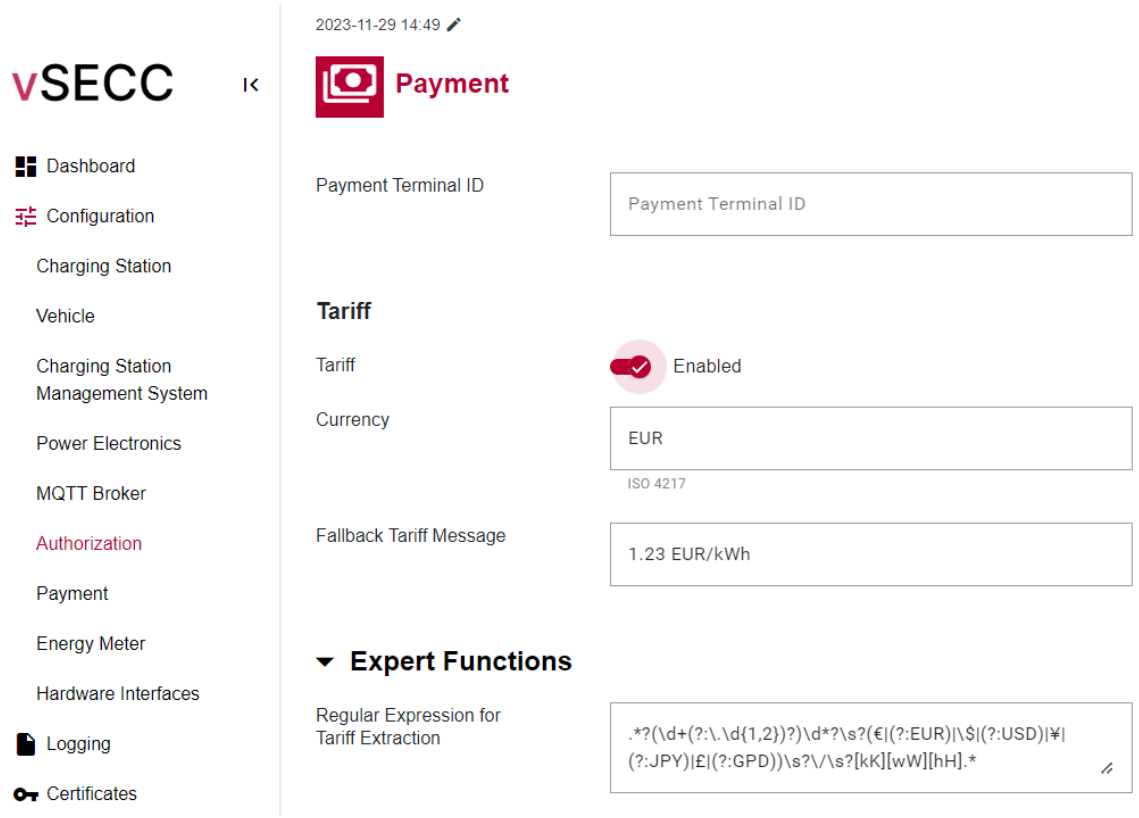


Figure 107: Tariff configuration in the Web Interface

Only 3-letter codes defined by ISO 4217 or currency symbols of the US-Dollar, Euro, Sterling or Japanese Yen are supported by the processing of the `TariffFallbackMessage` value. This limitation is independent from the currencies expected in the configurable regex. Supported currency symbols are replaced with their respective 3-letter code before further use. The raw tariff string is always published to a separate topic, even if the tariff extraction failed and is not published as a result.
With OCPP 1.6, the `TariffFallbackMessage` is replaced by the configuration key `DefaultPrice`. This unspecified key is part of the OCA's recommendation in *OCPP & California Pricing Requirements*. For the vSECC Software, 1.6 DefaultPrice is simply an alias for 2.0.1 TariffFallbackMessage and therefore behaves identically.

The transaction-specific costs are calculated by the CSMS. In the case of OCPP 2.0.1, they are sent to the charging station via `TransactionEventResponse`s. For OCPP 1.6 it is necessary that the CSMS implements custom data transfer messages according to *OCPP & California Pricing Requirements*. See subsection 8.14 for further details. The costs are combined with the currency set for the charging station and then published on the MQTT bus in the topic *vsecc/connector/evse_id/ocpp/cost*. On the bus, there are no differences between running and total cost.

8.13 MQTT Broker

The vSECC Controller provides an MQTT broker to exchange data with external MQTT clients. This interface provides the possibility to read current charging parameters, control all digital in and digital out connectors (Figure 14) and read analog values like voltage and temperature (Figure 8). For example this could be used to attach a human machine interface (HMI)-controller with a graphical user interface (GUI). All available MQTT topics can be found in Appendix J.

8.13.1 Configuration

The MQTT broker is available after the startup of the device is completed. An external connection to the MQTT broker is only possible with valid credentials.
Changing the default user name and password of the MQTT Interface is possible via the web interface as shown in Figure 108.
First, enter the new user name in the field **Username**. Afterwards enter and repeat the new password in the field **(Repeat) Password**. If the user name is available and the input fields for password match, the new user name and password will be set by pressing the button **[Save]** and confirming.

Caution: The vSECC Controller is coming without a valid user name and password for MQTT. A valid user name and password must first be set. The communication with external MQTT clients is not possible without these settings. Only one credential set can be configured at the same time. If a new user with corresponding password information is written, the old one will be deleted.

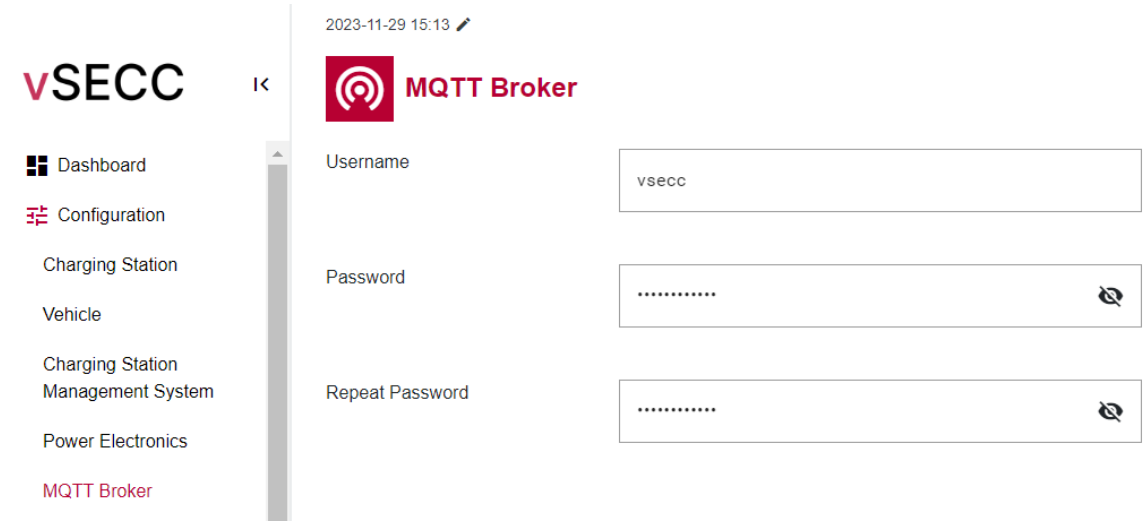


Figure 108: Change MQTT Interface Password

The execution of a factory reset (see chapter 2.3.1) will delete the MQTT credentials. After a factory reset the communication with external MQTT clients is not possible anymore until a new user name and password is set.

- Supported protocol versions: MQTT 5.0 / 3.1.1 / 3.1
- Port: 1883
- Allowed open connections: 50
- Message size limit: 20 MB

The IP address of the MQTT broker is configured with the vSECC Controller's IP configuration in the web interface. One can choose between the connectors ETH1/2 for the MQTT broker. The MQTT broker running on the vSECC Controller will listen on the MQTT standard port 1883.

Caution: A connection to the MQTT interface is not possible without user name and password settings.

8.13.2 Charging Session Topic Reset

Some MQTT messages are valid only for the duration of a specific charging session, such as the State of Charge (SoC) or the EVCCID. These topics are published as soon as the respective value is available or changes, and cleared (i.e. reset) by publishing the empty string after the charging session has ended. For some of these topics the empty string is also published once during startup. The topics handled by this mechanism are listed in Appendix J.
Please note the special case of "charging_session_state": Its standard-specific content is reset at the end of CCS charging sessions. It remains in this state up until a new session is started. For CHAdeMO sessions, the reset also happens at the end of a charging session. In contrast to CCS sessions tough, after the reset happened, either "connector_inoperative" or "state_b_ev_connected" are published. This is normal and expected behavior.

8.14 OCPP DataTransfer

The vSECC Software supports custom messages via OCPP's `DataTransfer` functionality in one of the following ways, according to the configured behavior:

- Internal handling of DataTransfer requests inspired by the Open Charge Alliance's proposal OCPP & California Pricing Requirements
- DataTransfer request and response messages are forwarded to/from MQTT for custom handling

The forwarding of DataTransfer messages to/from MQTT can be enabled in the web interface as shown in Figure 109. Forwarding will disable internal handling of DataTransfer messages.

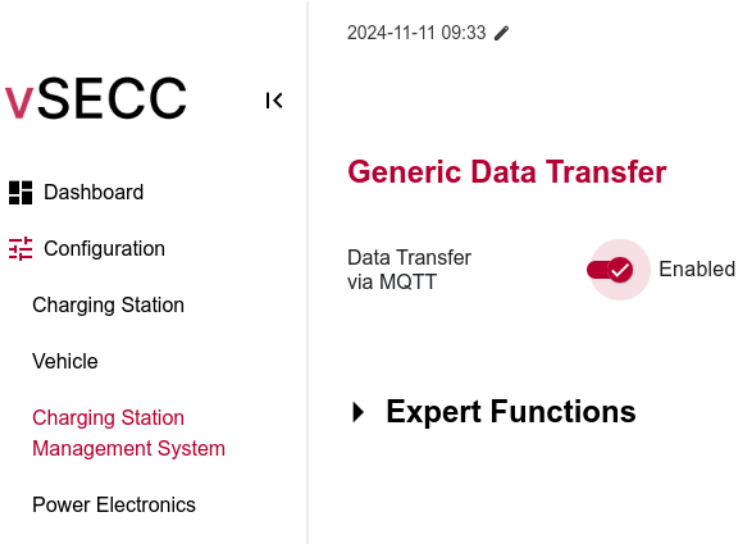


Figure 109: Configure OCPP DataTransfer message handling

8.14.1 Internal handling of DataTransfer messages according to OCPP & California Pricing Requirements

If forwarding of DataTransfer messages via MQTT is disabled and OCPP 1.6 is used for communication with the CSMS, the vSECC Controller supports the following DataTransfer messages for the vendor identifier ”org.openchargealliance.costmsg”:

- FinalCost
- RunningCost

Only a subset of the proposed message attributes is supported by the vSECC Software: An integer `transactionId` and a decimal number `cost`. The messages can be used to present cost at the charging station.

8.14.2 Forwarding of DataTransfer requests and responses via MQTT

When enabled, the vSECC Controller will not react to any DataTransfer request or response messages received from the CSMS, but instead forward them to MQTT. It will also forward request and response messages from MQTT to the CSMS. In case of an error, the vSECC Controller will publish an error notification via MQTT as well. Data published on these MQTT topics must be in the respective JSON format. In case no connection to a CSMS is established, messages to the CSMS will be buffered. Once a connection is established, buffered messages will be sent. On application restart the buffer is cleared.
Appendix M.1 visualizes different goodcase and error scenarios regarding the communication via MQTT and OCPP. Please refer to appendix N for the respective JSON schemas.
In this configuration, a custom application for OCPP DataTransfer message handling must subscribe to the respective MQTT topics of the vSECC Controllers’ MQTT broker. It needs to handle incoming DataTransfer request and/or response messages and publish corresponding response and/or request messages to the respective topics.
MQTT topics regarding DataTransfer requests from the CSMS to the charging station and responses to the CSMS:

- `vsecc/ocpp_data_transfer/received_request_from_csms`
  Requests received from the CSMS are published on this topic by the vSECC Controller
- `vsecc/ocpp_data_transfer/data_transfer_from_csms_notification`
  If an error occurred when a DataTransfer response was provided to be sent to the CSMS, or no response was provided in time, a notification is published on this topic
- `vsecc/ocpp_data_transfer/send_response_to_csms`
  Response messages published on this topic will be forwarded to the CSMS

MQTT topics regarding DataTransfer requests to the CSMS from the charging station and responses from the CSMS:

- `vsecc/ocpp_data_transfer/received_response_from_csms`
  Responses received from the CSMS are published on this topic by the vSECC Controller
- `vsecc/ocpp_data_transfer/send_request_to_csms`
  Request messages published on this topic will be forwarded to the CSMS
- `vsecc/ocpp_data_transfer/data_transfer_to_csms_notification`
  If an error occurred when a DataTransfer request was provided to be sent to the CSMS, or no response was received in time, a notification is published on this topic

The vSECC will detect and notify the following errors:

- Timeout, if a request was sent to the CSMS, and it did not respond in time.
- Timeout, if a request from the CSMS was received but not answered in time. This is reported if the CSMS sends another request, which indicates its timeout was triggered. It is also reported if twice the time of a timeout for sent requests has passed.
- Invalid Format, if a request or response message for sending to the CSMS was published, but it did not match the respective JSON schema.
- Transfer Already Ongoing, if a request to the CSMS was published, but a response for the last request is still expected.
- No Transfer Ongoing, if a response to the CSMS was published, but the last request was already answered.

8.15 External Measurands

The vSECC Controller offers the option to send externally measured values to the CSMS. This works by accepting special messages over MQTT, buffering and filtering them internally, and sending them over OCPP at the configured instant.

8.15.1 Communication Entities and Message Sequence

The communication sequence is as follows: An actor external to the vSECC Controller publishes an MQTT message containing a JSON object to the topic `vsecc/connector/evse_id/ocpp/external_measurand`.
The vSECC Controller then validates the incoming message against a JSON schema. The respective JSON could be found in Section J.3. It is similar to the data structure defined by the OCPP 2.0.1 specification in Chapter *2.41. (SampledValueType)*. The message is then added to an internal queue of items which get sent to the CSMS.
All messages queued so far are sent to the CSMS with the next *TransactionEventRequest*. Information measured or provided by the vSECC Controller internally (e.g. the State of Charge/SOC) is put into the same queue. After the queue elements have been sent, the internal queue is flushed. This means that a specific information received over MQTT is not

sent to the CSMS multiple times automatically. For it to be included in the next *TransactionEventRequest*, too, it must be added to the queue again by publishing to the topic mentioned above.
Valid messages received over MQTT contain at least a timestamp and a value. Other attributes, such as the measurand or phase, are optional. Two such messages are considered to be of the same kind if their measurand, location and phase are the same. Only one message of a kind is queued. This means that if a message has already been queued and a new one of the same kind is received, the old message (with all its mandatory and optional attributes, including the value and timestamp) is replaced by the new message.
OCPP 1.6 and OCPP 2.0.1 differ in the allowed units for external measurands. OCPP 2.0.1 allows arbitrary strings while OCPP 1.6 defines a specific set (refer to *OCPP 1.6, Section 7.45. (UnitOfMeasure)*). Futhermore, OCPP 1.6 allows the measurands *Temperature* and *RPM* while OCPP 2.0.1 does not. A message received over MQTT with an attribute set to a value which is not allowed by the OCPP standard currently used for the CSMS connection is dropped and NOT sent to the CSMS.

The queue is flushed after a transaction has ended.

Filtering according to the configuration occurs. Data is sent to the CSMS only if the respective measurand is configured to be sent at the specific *eventType* (*Started*, *Updated* or *Ended*). Web-UI menu: *Charging Station Management System / Sampled Meter Values / Send Measurands*). When using OCPP 1.6, Energy.Active.Import.Register is mandatory and it will be always sampled.

Messages received over MQTT on the topic above are processed only if MQTT is enabled as source for measurands (Configuration variable: `sampled_data_ctrlr_enable_mqtt_measurands`, Web-UI menu: *Charging Station Management System / Meter Values / Measurands Source*).

8.16 Clock Aligned Meter Values

The vSECC Controller offers the option to send clock aligned measured values to the CSMS. The clock aligned meter values feature ensures that meter values are recorded and reported at regular intervals, aligned with the clock. Currently only the Energy.Active.Import.Register measurand could be sampled clock aligned. There are two possibilities to sample the Energy.Active.Import.Register clock aligned.

- **Transaction independent**: If the *Aligned Interval* is set to e.g. 900 seconds (= 15 minutes) the vSECC Controller samples the configured measurands every quarter of an hour e.g. 10:00/10:15/10:30 and so on. This results every 15 minutes in a *MeterValuesRequest* of *context: (Sample.Clock)*. This message will also be reported during an active transaction.

> **Transaction dependent**: If the *Aligned Interval at Transaction End* is set to e.g. 900 seconds and a transaction starts at 10:58 o'clock the vSECC Controller samples the configured measurands every quarter of an hour e.g. 11:00/11:15/11:30 and so on during the active transaction. At the end of the transaction each clock aligned sampled value in the configured interval is provided in the *TransactionEventRequest* of *eventType (Ended)*.

Only the measurand Energy.Active.Import.Register is sampled clock aligned.

Data is sent to the CSMS only if the respective switch *Enable Aligned Data* is activated and one of the configurable intervals (*Aligned Interval* or *Aligned Interval at Transaction End*) is larger than zero seconds. Web-UI menu: *Charging Station Management System / Clock Aligned Meter Values*

8.17 Display Message

OCPP 2.0.1 provides the possibility to send messages from the CSMS that should be shown on a display in the charging station. The vSECC Software supports displaying messages by forwarding the raw OCPP payloads from the CSMS onto the MQTT bus. Section J describes the corresponding topics.
The usage of the Display Message functionality can be enabled in the web interface section **Configuration / Display**, as shown in Figure 110. If it is disabled, the vSECC Software will reject any incoming display messages from the CSMS.

Caution: In Release 3.0, it is required that the configuration variable "`display_message_ctrlr_available`" is set to *"true"*. In the default configuration, it was set to *"false"*. In version 3.0, there is no way to set this variable in the web interface. It must be set either via CSMS, Provisioning Tool (delivered through the portal) or REST-API (see chapter 7.9). In version 3.1, this was fixed.

Figure 110: Configuring the Usage of the OCPP Display Message

The vSECC Software supports the following

> **Formats**: ASCII, HTML, URI, UTF-8

> **Priorities**: Always in Front, In Front, Normal Cycle

The supported priority and format configurations are used to reject messages with unsupported options. A validation if the content itself matches the specified options is not performed. Other OCPP requirements and checks are also not supported.
Finally, accepted requests are forwarded as a whole to the MQTT bus. This allows the potential topic subscriber to consider so far ignored options of the request. Nonetheless, a subscriber cannot reject display messages, as the vSECC Software already accepted the request at this point.

8.18 Power Electronics

Please note that to date, some features are only available on the vSECC, and not yet on the vSECC.single.

For each connector of the vSECC Controller there are three configuration options in the web interface for communicating with a power electronics, as shown in 112:

> **Websocket:** Connection via PEP-WS protocol over Ethernet (see Section 2.2.10). Websocket must be set when developing custom PE-communication in the Configurable Customer Interface.

> **Simulation:** For development only! This mode simulates a power electronics internally.

> **CAN:** Connection via PEP-CAN protocol over CAN (see Section 2.2.7).

Some features are only available via PEP-WS:

> Inverted pantograph control via virtual I/Os

8.18.1 Websocket

Refer to the PEP-WS document provided with the Documentation package for usage instructions.
Multiple connections on a single IP address and port are possible, as long as the endpoint is different, e.g. *http://192.168.1.5:80/PE1* and *http://192.168.1.5:80/PE2*.
The configuration variable **Configuration Poll Interval** (under **Configuration / Power Electronics / Timings**) controls when the configuration is requested from the power electronics and how limit values are communicated to the EV. When set to `0` (default value), the configuration is only requested once, when the communication to the power electronics is established. When set to a positive value, the configuration is requested regularly, but the values are not updated during a charging session. The last requested configuration right before a charging session is then valid throughout the whole session.

Caution: To avoid hiccups in the EV communication, respond to `request` messages in a timely manner.

8.18.2 CAN

The vSECC Controller is able to control its power electronics via CAN; the vSECC on port CAN2 (see Section 2.2.7), the vSECC.single Board on CAN1 (see Section 4.2.5).
Refer to the PEP-CAN document provided with the Documentation package for usage instructions. Base addresses define the CAN IDs of the messages, e.g. base address 0x300 for the first connector means that its *VehicleStatus* message has CAN ID 0x301. Make sure that the configured base addresses do not cause the CAN IDs to overlap. The baud rate and base addresses of the CAN communication can be configured via Web Interface.
The general charging sequence is identical to the PEP-WS protocol and described in detail in the PEP-WS document. But there are some differences between both protocols which are described in Section 1.4 of the PEP-CAN document.
Since the values of the signals in the CAN frames only have a fixed resolution, factors are used to scale them to a usable range. For some use cases (e.g. Megawatt Charging) the default value ranges are not sufficient and need to be changed. Therefore, the factors for all values representing voltage, current or power can be configured.
When CAN is selected as protocol for a connector, these factors can be modified in the web interface in the **Expert Functions** section of **Power Electronics** configuration. Here they are grouped by the message purposes **Charging Session Info**, **PECC Limits** and **Status**.

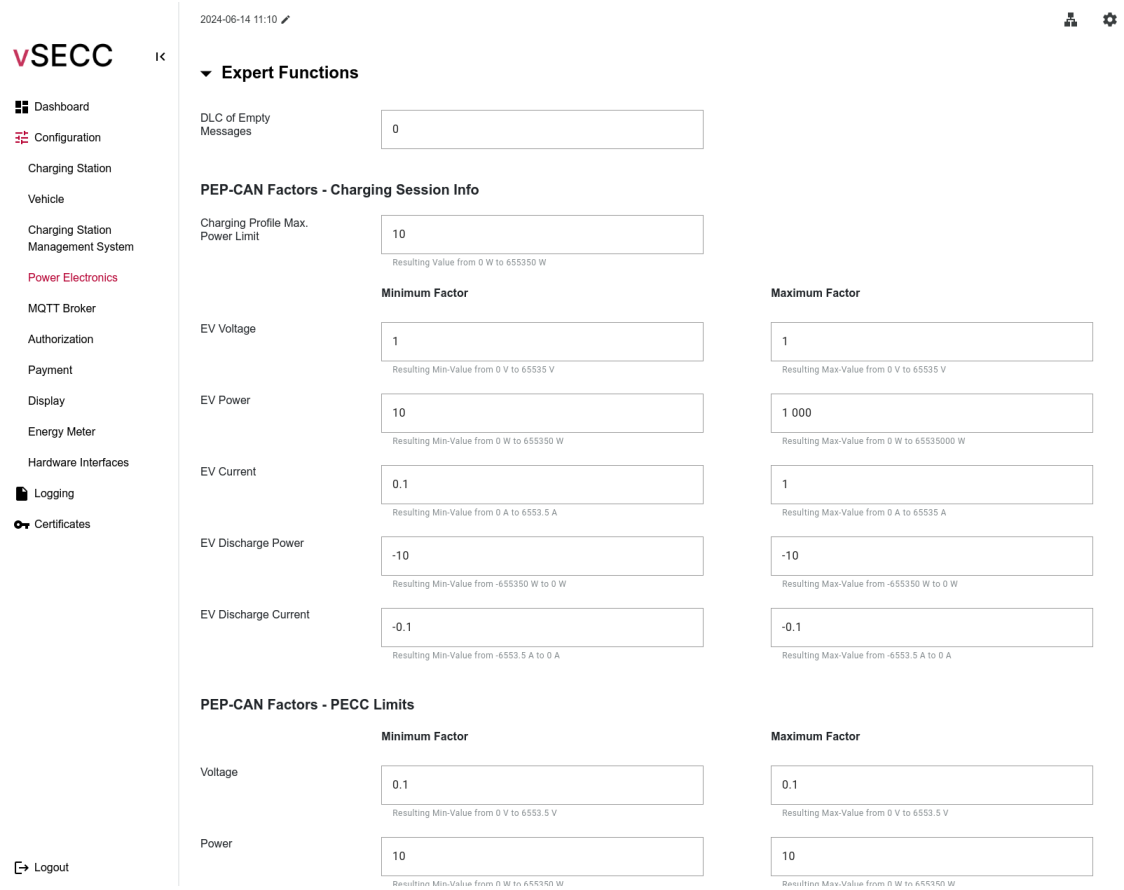


Figure 111: PEP-CAN Factors Modification

Additionally, the factors can be set using the provisioning tool. The naming scheme is: `pep_can_factor_` followed by the name of the signal from the PEP-CAN document. A list

of all configuration variable names can be found in the `config-variables` file as part of the provisioning tool release.

8.18.3 Simulation

When simulating a power electronics, max values can be configured via the Web Interface:

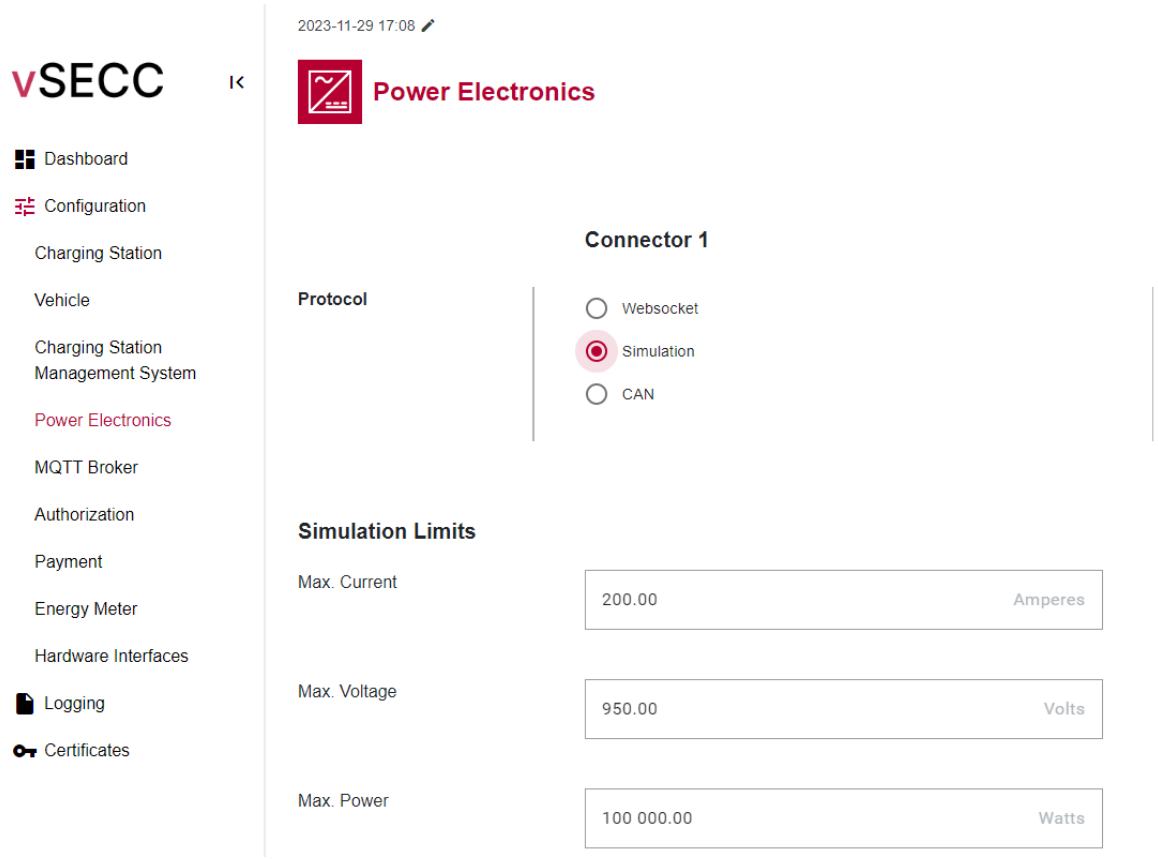


Figure 112: Power Electronics Simulation

8.18.4 PEP Input/Output Identifiers

The vSECC Controllers offer a set of ports which can be controlled over Ethernet via the Power Electronics Protocol (PEP-WS) `getInput` and `setOutput` messages. The ports can also be controlled over CAN using PEP-CAN and its `DigitalOuts1`, `DigitalOuts2`, `DigitalIns`, `AnalogIns1`, `AnalogIns2` and `AnalogIns3` messages.

Refer to the PEP-WS and PEP-CAN documents provided with the Documentation package for usage instructions.

8.18.5 Digital Out

The digital output ports can be set either with the PEP-WS `setOutput` message or PEP-CAN `DigitalOuts1` and `DigitalOuts2` messages. Valid values are `0` for logical low, and `1` for logical high.

Refer to Section K for more details about digital out PEP-CAN and PEP-WS identifiers.

8.18.6 Digital In

The digital inputs ports can be read with the PEP-WS `getInput` message and PEP-CAN `DigitalIns` message. Return values are `0` for logical low, and `1` for logical high.

Refer to Section K for more details about digital in PEP-CAN and PEP-WS identifiers.

8.18.7 Analog In

The ports can be read with the PEP-WS `getInput` message and PEP-CAN `AnalogIns3` message.

Refer to Section K for more details about analog in PEP-CAN and PEP-WS identifiers.

8.18.8 Temperature In

The temperature inputs can be read with the PEP-WS `getInput` message and PEP-CAN `AnalogIns1`, `AnalogIns2` and `AnalogIns3` messages.

Refer to Section K for more details about temperature in PEP-CAN and PEP-WS identifiers.

8.18.9 Virtual In/Outs (vSECC only)

The virtual I/Os can be used to control a pantograph, when the **Inverted Pantograph Control** in the **Configuration / Vehicle** Section is set to **PEP-WS**. Refer to Section 8.5 for more details. The virtual inputs can be read with the PEP-WS `getInput` message, the outputs written with the PEP-WS `setOutput` message.

Make sure to poll the `panto_control` input regularly to get notified when a pantograph movement is requested. Use the `panto_up`, `panto_down` and `panto_error` outputs to notify the vSECC about pantograph state changes.

- `panto_control (input)`: Requested pantograph position, `0` for up, `1` for down
- `panto_up (output)`: Set to `1`, when the pantograph is in its upper endposition
- `panto_down (output)`: Set to `1`, when the pantograph is in its lower endposition
- `panto_error (output)`: Set to `1`, when charging is not allowed (e.g.: vehicle not in position, wind speed limit exceeded, mechanical problems, …)

8.19 Power Electronics Dynamic Limits

There are several ways, how the Power Electronics (PE) is able to communicate limits (voltage, current, power) to the SECC and also the EV.
In general, there are two sorts of limits:

Static limits: Representing the physical limits of the PE. These are communicated to the EV in the ChargeParameterDiscovery message for CCS/Inverted Pantograph. These also apply to CHAdeMO.

Dynamics limits: These represent currently applicable limits of the PE caused by e.g. temperature or environmental constraints. They are communicated to the EV in the CurrentDemand/ChargeLoop for CCS/Inverted Pantograph. These are not applicable to CHAdeMO.

The **static** limits can be set via the `response-configuration` message for PEP-WS (and changed during operation, if a polling interval is configured). For PEP-CAN the limits can be set at any time via the `PECC-Limits` CAN frames.
The **dynamic** limits can be set via the `info-dynamicLimits` message for PEP-WS. This feature is not available when using the PEP-CAN protocol.
If no dynamic limits are configured, static limits are applied. Dynamic limits persist across charging sessions. Once set, they are only reset, when the connection to the PECC is lost. They can not be unset via the `info-dynamicLimits` message, but can be set to the same values as the static limits. Dynamic limits should not be higher than the respective static limits.

It is recommended to use either dynamic limits to change limits during CurrentDemand or the `response-configuration` message using a ConfigPollInterval greater than 0. If dynamic limits are used, the limits cannot be changed via config polling.

8.20 Modbus Gateway

RS485 pin polarity: in the vSECC product family, the RS485 pin labled A always marks the noninverting pin, whereas B marks the inverting pin.

The vSECC Controllers provide a Modbus TCP to RTU gateway. One or multiple Modbus TCP clients (masters) can connect via one of the Ethernet connectors to one or multiple Modbus RTU slaves, they are wired on the RS485 interface (see section 2.2.7). Thus, the vSECC Controllers allow to connect peripherals like an energy meter or isolation monitor to the RS485 interface and make it available to the power electronics or other modules over Modbus TCP. This eliminates the need for a dedicated device in the charging station which was was previously needed to make Modbus RTU slaves available on the Ethernet communication.

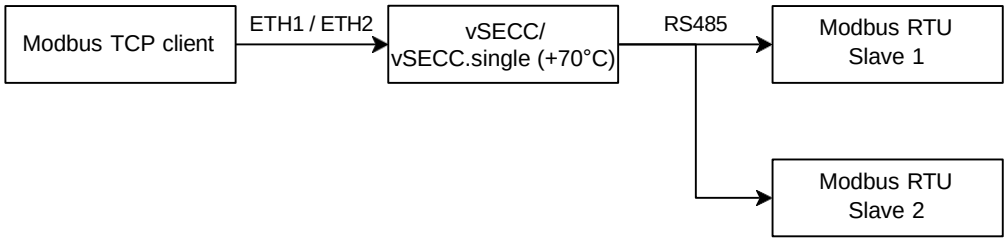


Figure 113: Modbus Gateway Overview

8.20.1 Configuration

The Modbus TCP to RTU gateway is configured in the **Configuration / Hardware Interfaces / RS 485** section of the web interface. To use it, the settings must be switched to **Modbus Bridge**. It is also possible to use the RS485 interface in the container. The container usage is described in section 8.27.

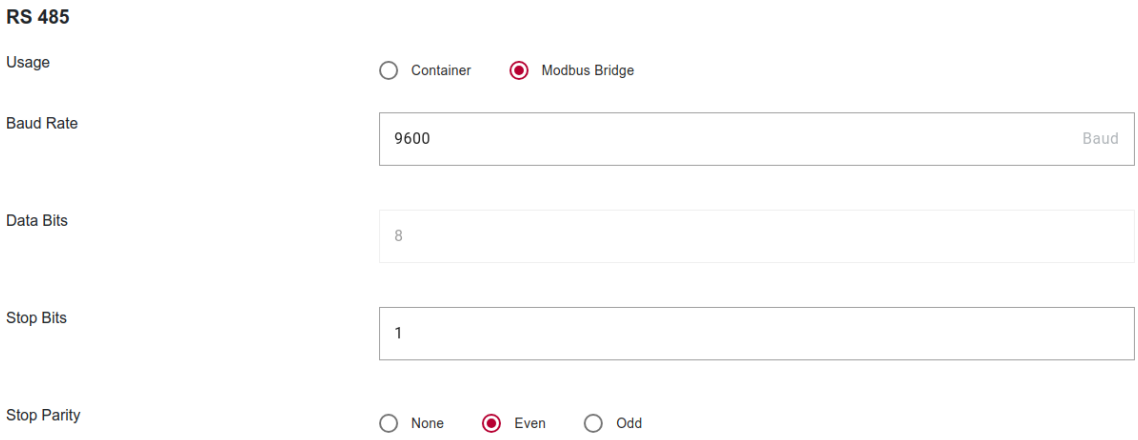


Figure 114: RS485 Settings for Modbus TCP to RTU Gateway in the Web Interface

The default RS485 settings are:

- Baud Rate: 9600
- Data Bits: 8
- Stop Bits: 1
- Stop Parity: even

Caution: Changing these parameters requires a restart. After the parameters are saved, the software will automatically be restarted to apply all changes.

On the TCP side, the Modbus IP address is configured with the vSECC Controller's IP configuration in the Web Interface (**Configuration / Hardware Interfaces**). One can choose between the connectors ETH1/2 for the Modbus TCP to RTU gateway. The Modbus TCP gateway running on the vSECC Controller will listen on the suggested TCP port 502.

8.21 Energy Meter

The vSECC Controller supports energy meters in order to provide readings to a CSMS. On start and end of an OCPP transaction, signed and unsigned meter readings are sent to the CSMS in the OCPP 1.6 `MeterValues` or OCPP 2.0.1 `TransactionEventRequest` message. For the signed meter readings, OCMF (Open Charge Metering Format) is used. Updates containing unsigned readings are also sent regularly based on the `MeterValueSampleInterval` (OCPP 1.6) / `SampledDataTxUpdatedInterval` (OCPP 2.0.1) set by the CSMS.
A ”virtual” energy meter inside the vSECC Software provides a calculated energy value based on the voltage and current values provided by the power electronics. This can be used for a simple estimation of the transferred energy.

At this moment, the vSECC Controllers come along with a direct interface to the LEM DCBM 400/600 energy meters[1]. Furthermore, example flows for interfacing to the AST DC Meter are available for use in the Configurable Customer Interface[2]. To provide billing capabilities, a dedicated energy meter per charging connector can be connected via Ethernet. If the two Ethernet ports are already used for e.g. PE and CSMS connection, it is recommended to use an Ethernet Switch.
Another physical energy meter can be used together with the MQTT interface of the vSECC Software. Thus, a custom implementation towards the desired energy meter or an energy meter with a MQTT interface is required. This can be realized in the Configurable Customer Interface.
Energy meters can be configured in the **Configuration / Energy Meter** Section via the **Type of Energy Meter** variable, as shown in Figure 115. For use with the Configurable Customer Interface (e.g. for the AST DC Meter), **Custom** must be selected.

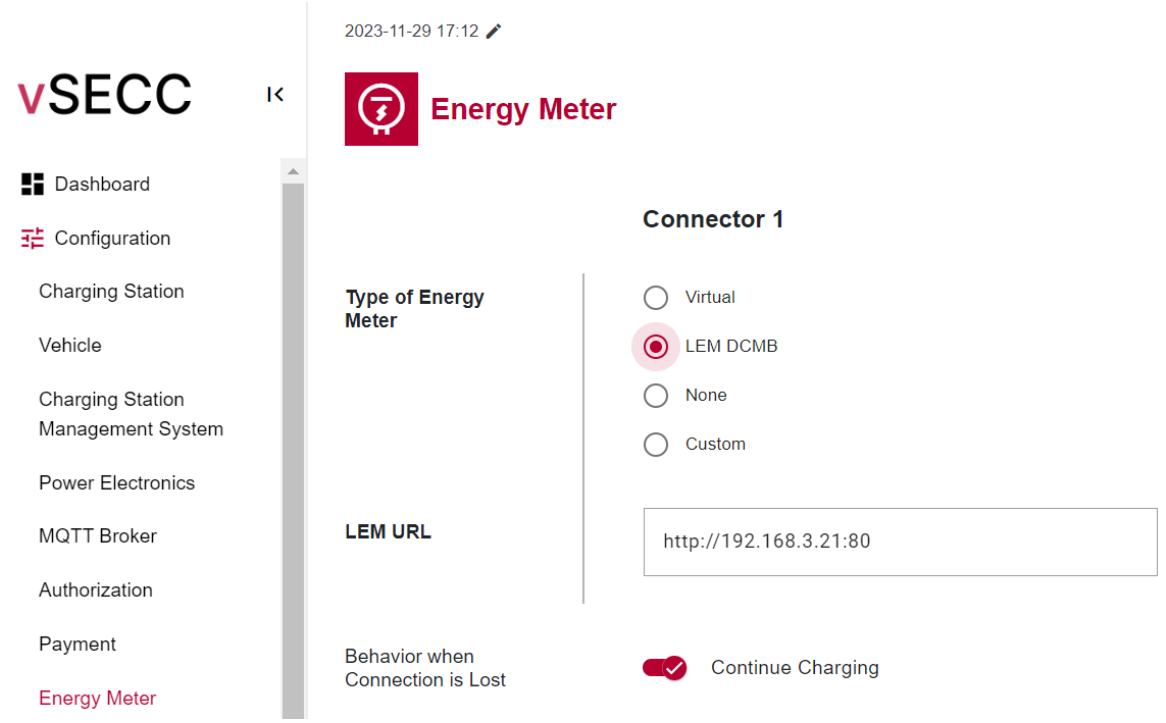


Figure 115: Energy Meter configuration in the Web Interface

[2]https://www.ast-international.com/en.products.dc-meter.html

The following prerequisites must be met for the energy meter to work:

- The **Transaction Start Point** and **Transaction Stop Point** in the web interface in Section **Configuration / Charging Station Management System / Transactions** are recommended to be set to either:
  - TxStartPoint **EV Connected**, TxStopPoint **EV Connected**
  - TxStartPoint **EV Connected**, TxStopPoint **EV Connected and Authorized**
  - TxStartPoint **Power Path Closed**, TxStopPoint **Power Path Closed**
- **Transaction Stop Points Authorized** or **Power Path Closed** should only be used if it can be guaranteed that no deauthorization can happen during charging (e.g. if no RFID reader is connected and the CSMS does not send `RemoteStop` (OCPP 1.6) / `RequestStopTransaction` (OCPP 2.0.1) messages)
- **Meter Values Measurands (At Transaction Start, During Transaction, At Transaction End)** in the web interface in Section **Configuration / Charging Station Management System / Meter Values** must include **Energy Active Import Register** for the readings to be transmitted to the CSMS at the respective time points.

The model and serial number for the fiscal metering can be set in the **Configuration / Charging Station** section. When set, these values are included in the OCPP 1.6 `BootNotification` message and the OCPP 2.0.1 device model.

8.21.1 Use of LEM DCBM 400/600

To use the LEM energy meter, its address must be configured via the **LEM URL** variable as shown in Figure 115. The Format should be `http://<IP Address>:<Port>`, e.g. `http://192.168.3.21:80` (HTTPS is currently not supported).
Via the **Behavior when Connection is Lost** toggle the behavior, what happens when the connection to the meter is broken while charging, is set. It can be chosen if the corresponding connector should become inoperative until the connection is re-established. Otherwise charging sessions are not aborted and new sessions can be established, but no meter readings are sent to the CSMS.

Caution: Make sure that the time synchronization on the LEM meter is valid, e.g. by configuring NTP on the meter.

Additional steps are required to fulfill the requirements of the German Eichrecht. The virtual energy meter cannot be used for Eichrecht-compliant billing.

8.21.2 Use of MQTT Interface

MQTT can be used to interface any energy meter to the vSECC Controller by providing the glue code between the API of the desired energy meter and the MQTT interface described in this chapter.
A standard message flow of a successful charging session can be found in Annex L. The corresponding MQTT topics can be found in Annex J.
For simplicity, the following description only contains MQTT topics for connector 1, replace the EVSE ID accordingly for connector 2.

Caution: Please note that the vSECC Controller expects certain messages in order to continue in the charging process. These are described with ”**must**”.

Startup On startup, the vSECC Controller is ready to receive MQTT messages as soon as it publishes `ready` on the **vsecc/readiness** topic. It keeps the charging connector inoperative by publishing `energy_meter_unavailable` to **vsecc/connector/1/status/components/report_failure**. Once the energy meter is ready, it **must** publish `energy_meter_unavailable` to **vsecc/connector/1/status/components/report_resolution** in order to make the connector operative.

Measurement Status The energy meter **must** report its measurement status at startup and then whenever it changes. This is done via the **vsecc/connector/1/em/measurement_status** topic. Possible values are `not_running` and `running`.

Unsigned Readings The energy meter needs to publish unsigned readings at regular intervals, the last published value is sent to the CSMS whenever meter values are transmitted. The values must be sent to the topic **vsecc/connector/1/em/unsigned_reading** with the format `<Timestamp>|<Import reading>|<Export reading>`. The timestamp must be according to RFC3339, readings are in kWh with a dot as decimal separator, import for charging, export for discharging.

Signed Readings Right after the start and end of a charging session, the vSECC Controller requests a signed reading that **must** be answered by the energy meter to ensure a proper flow of information between vSECC Controller, CSMS and energy meter. The request is published by the vSECC Controller in `get_reading_signed` via the **vsecc/connector/1/em/measurement_control** topic. If the energy meter provides signed readings, the response must be in the format `OCMF|<JSON1>|<JSON2>|{”PUBKEY”:”<PUBKEY>”}`. If no signed readings are supported, the request must still be answered with `NOT_SUPPORTED`.

Charging Session The vSECC Controller requests the start of an energy meter measurement with `start_measurement` via the **vsecc/connector/1/em/measurement_control** topic. Then it waits until the energy meter **must** report `running` as measurement status. Right after the start of the measurement, a signed reading is requested (which **must** be answered with a signed (i.e. `NOT_SUPPORTED`) reading). If specific information regarding the charging session, transaction, user, etc. is required to start the measurement, e.g. to include in the OCMF-tuple for Eichrecht, see Annex J for more MQTT topics to subscribe to. At the end of the charging session, the vSECC Controller requests the measurement stop with `stop_measurement` via the **vsecc/connector/1/em/measurement_control** topic. Finally, after the charging session has stopped, another signed reading is requested. Please note that further unsigned readings received after the signed reading, are not forwarded to the CSMS anymore.

Error Handling If there are any issues with the energy meter and the charging session should stop, make sure to publish `energy_meter_unavailable` to **vsecc/connector/1/status/components/report_failure**. This will stop the charging session and prevent further sessions until the failure is resolved. If the charging session should continue even if there is an issue with the energy meter, make sure to respond to the commands sent by the vSECC Controller.

8.21.3 Prerequisites for vSECC Controllers to fulfil German Eichrecht

Caution: Vector can provide no guarantee of the completeness of the prerequisites with regards to the charge controller to adhere to the German measurement and calibration law. Please advice a conformity assessment body.

When charging the electric vehicle, the driver or owner should only pay for the amount of energy that was consumed. The basis to ensure that is provided by the German measurement and calibration law. Therefore, when charging stations are built up in Germany and the charging is not for free, they must be compliant with the German ”Eichrecht”.
Today, a certification of the charging station (with a vSECC Controller) by a conformity assessment body can be achieved under the following conditions:

1. **Measuring Capsule**
   The vSECC Controller is not considered part of the so-called ”measuring capsule”, since it only forwards a data structure from the energy meter to the CSMS without modifying it. When the vSECC Controller has started the transaction, the energy meter starts accumulating energy for this transaction (identified with a unique transaction identifier chosen by the energy meter). When the transaction is stopped, the energy meter sends the data readouts in OCMF format in response. The whole data structure is authenticated by a signature mechanism of the energy meter.
   Please refer to the DCBM Operation Manual [1] for further details on this mechanism.

[1] https://www.lem.com/en/file/10314/download

The vSECC Software then forwards these signed data readouts to the CSMS, which can verify them and use the values e.g. for billing purposes. For more details on the ”Schalt-Mess-Koordination”, please refer to the documentation in Appendix P.

2. **Choice of energy meter**
   Since the vSECC Controller is not allowed to modify any parameters of the energy meter or the data structure of the meter readouts, in order not to be part of the measuring capsule, the energy meter must be chosen with one of the following parameters:
   > **A fixed-value cable compensation**
   The energy meter must be chosen in accordance with the cable resistance. E.g. for a resistance value of 2mΩ, the energy meter with the reference DCBM_N**2**D_-x0x0_0000 must be chosen. Other possible variants are DCBM_N**3**D_x0x0_0000 for 4mΩ cable compensation, or DCBM_N**4**D_x0x0_0000 for 6mΩ.
   > **Using 4-wire-measurement**
   Together with a suitable charging cable, the LEM energy meter offers the possibility to use four-wire measurement to measure the energy as close to the EV as possible. Therefore, the energy meter with the reference DCBM_N**1**D_x0x0_0000 must be chosen.

For more information on the certification requirements, please take a look at the Application Note ”Building charging stations compliant to German Measurement and Calibration Law with the vSECC Controllers” on vector.com/vsecc/documentation.

8.22 Failure Handling

The vSECC Controller is used together with other system components such as the power electronics communication controller (PECC) or the energy meter. When one of these components does not work properly, the other components should react on this failure and handle the exceptional situation gracefully. The set of possible situations include network connection losses, electrical problems (short to ground, power loss) or software bugs which lead to unintended behavior. Possible reactions to these situations include, among others, reporting, retries, restarts, graceful stops of the charging session and preventing new charging sessions until the situation is resolved.

The vSECC Controller employs its own failure handling: If a failure is detected internally by or reported externally to the vSECC Controller, it is added to the set of currently *”active failures”*.
If a failure has been resolved and this change is detected by or reported to the vSECC Controller, it is removed from the set of currently active failures.
As long as the set of active failures is empty, the vSECC Controller operates normally and allows charging (if not disabled explicitly via OCPP). If the set is not empty, the vSECC Controller switches its availability to *inoperative*, which is reported to the CSMS, stops active charging sessions (if any) and prevents new charging sessions.
This mechanism allows multiple failures to be present at the same time and tracking of resolutions independently of other, still ongoing failures.

In this context, a failure is determined by a specific identifier given as string. This identifier is used as handle to reference to this exact failure and shall be unique. Active failures, their resolution and the set of active failures are communicated via the following MQTT topics:

> vsecc/connector/{evse_id}/status/components/active_failures
> vsecc/connector/{evse_id}/status/components/report_failure
> vsecc/connector/{evse_id}/status/components/report_resolution

Similar to OCPP, *evse_id = 0* has a special meaning here: It represents failures on the charging station level. Reporting a failure on *evse_id 0* reports the whole charging station as *Faulted* towards OCPP. This also leads to all connectors becoming inoperative as a result, until the failure has been resolved. The vSECC Controller does not publish failures on the charging station level by itself, but the MQTT mechanism can be used to trigger this behavior.
An external component such as the PECC or energy meter may subscribe to the active failures and publish its own failures to the *report_failure* and *report_resolution* topics. It is recommended to publish a unique string identifying the respective failure in snake_case.

Publishing directly to *active_failures* breaks the vSECC-internal mechanism and is heavily discouraged.

The following failures may be set by the vSECC Controller. It is recommended to use other string values for customer-specific failures.

Failure	Description
energy_meter_unavailable	Energy meter unavailable.
energy_meter_daemon_not_running	The backend/interface of the energy meter is not running (only valid for LEM/Virtual energy meter).
energy_meter_daemon_exited	The backend/interface of the energy meter exited the connection (only valid for LEM/Virtual energy meter).
cm_qca_failure	Due to an error in the powerline communication, the QCA chip is flashed again. This will lead to an inoperative connector while trying to recover.
cm_min_peak	After connecting the EV, vSECC checks whether the diode in the vehicle is active. Measured voltage should be between 0 V and 12 V. When the diode is inactive, it is possible to measure voltages down to -12 V.

panto_init	Stays active until all pantograph signals have received initial valid values.
cm_state_e	If 0 V is measured at the CP pin, CP is in state E.
ccs_1_dip_switch	Configure PP pin monitoring via dip switch to implement the safety relay.
pe_connection	Power electronics connection fault.
pe_inoperative	Power electronics inoperative.
pe_isolation_fault	Power electronics isolation fault.
evse_triggered_emergency_shutdown	Shutdown which is triggered by the EVSE.
connector_lock	Only valid for CHAdeMO
hardware_software_error	Initializations failure of I/Os at the start of the vSECC app.

8.22.1 CCS DC connector behaviour

In error cases, the vSECC Controller CCS DC connectors behave according to IEC-61851-23.

EV initiated emergency shutdown (unexpected CP state) If a CCS DC charging session is terminated unexpectedly by the EV via an unexpected CP state, the vSECC Controller will switch the CP duty cycle on the according connector to 100% PWM, as defined in IEC-61851-23 ED2, Table CC.13, scenario 1. The according connector of the vSECC controller will stay operative and therefore available for further charging sessions if the EV performs a BCB toggle or on the next plug in.

EVSE initiated emergency shutdown due to PECC issue If the communication to the PECC failed or the PECC reports to be inoperative, the vSECC Controller also initiates an emergency shutdown immediately. The according connector then becomes inoperative. In this case the CP duty cycle will be set to 100%, usually resulting in state B1, according to IEC-61851-23 ED2, Table CC.13, scenario 2. This state will persist until plug-out. Upon plug-out, if unresolved active failures are still present, the CP duty cycle will change to 0% (CP state F) to indicate the connector to be inoperative, until all active failures are resolved. The according PECC error event ”pe_connection” / ”pe_inoperative” will be published via MQTT in the active_failures topic for the according connector. Additionally the active failure ”evse_triggered_emergency_shutdown” is published, it will be resolved by the vSECC Controller once the EV is plugged out. Once the according PECC is available and operative again, and the ”emergency_shutdown” failure is resolved by plug-out or by a third actor via MQTT, the charging connector will become operative again.

EVSE initiated shutdown due to active failure reported via MQTT If an active failure was reported via MQTT, the vehicle on the according connector will be asked to terminate the charging session. Once the charging session is terminated, the according connector's CP duty cycle will be set to 0% (State F). Upon resolution of all active failures, and if no other reason prevents the charging connector from being operative, the connector will leave CP state F and be available for further charging sessions.

Unexpected TCP or TLS connection termination If an EV unexpectedly closes the TCP or TLS connection or unexpectedly terminates the V2G communication, the vSECC behaves the same as if an unexpected CP state was detected: The according connector's CP duty cycle will be set to 100% (State X1) and the charging session will be aborted. The connector is available for further charging sessions if the EV performs a BCB toggle, or on the next plug in.

Connector set inoperative by CSMS If a CCS DC connector of the vSECC Controller is set to inoperative via the CSMS, a currently active charging session on this connector will not be interrupted, but once the charging session is finished, the connector's CP duty cycle will be set to 0% (State F).

8.23 OCPP Transaction Persistence

If a CSMS connection is configured, the vSECC Controller handles OCPP transactions. A running transaction must be terminated by the vSECC Controller, even in the case of power outages or other failure conditions leading to a restart or reboot of the vSECC Controller. In order to satisfy this requirement, OCPP transaction information is stored in non-volatile memory. If the vSECC Software starts and detects that a transaction has not been terminated correctly yet, it sends a `TransactionEventRequest` with the `eventType` set to *Ended* for each open transaction.

Only the `TransactionEventRequest (eventType =` *Ended)* message is sent to the CSMS. Other messages belonging to an ongoing transaction are neither recovered nor (re-)sent to the CSMS.

8.24 OCPP Reservations

The vSECC Controllers support the OCPP reservation feature. To enable this feature, select the corresponding configuration options in the `Reservation` section of the web interface (see Figure 116).
The vSECC Controller handles the reservation requests from the CSMS automatically. If a specific EVSE is reserved, the corresponding connector is blocked until the reservation is cancelled or expires. A reservation without an EVSE-ID does not necessarily lead to a connector being blocked. This only happens when blocking is necessary to fulfill the reservation. If an EVSE becomes unavailable, the reservations are recalculated, and this can cause reservation cancelation. Blocked means that authorization is blocked for any user that does not match the reservation criteria at this connector.
The MQTT topic `vsecc/connector/{evse_id}/status/reserved` informs about the reservation status of the connector. It can be used, for example, to display the reservation status on the HMI.

Figure 116: OCPP Reservation configuration options in the Web Interface

8.25 OCPP Availability

There are several ways how the availability of a connector, EVSE or the charging station can be influenced:

ChangeAvailability: The CSMS can change the availability via the `ChangeAvailability` message.

Reservations: The CSMS may influence the availability by triggering reservations.

Failure Handling: Faulted components influence the availability. See Chapter 8.22 for details.

MQTT: Availability can also be modified through MQTT using the `vsecc/connector/{evse_id}/status/set_availability` topic.
See Annex J for details on the topic.

Similar to OCPP, *evse_id = 0* has a special meaning here when used in the MQTT topic: It represents the availability on the charging station level.
The current availability is published to `vsecc/connector/{evse_id}/status/availability`.

Availability changes persist across reboots.

8.26 Status Notification

The Status Notification message is used in OCPP to report a status change or an error within the charging station.

8.26.1 OCPP 1.6

Reporting a status change or an error are possible for the components *Charging station* or *Connector*. By default, the vSECC Software uses the fields `connectorId, errorCode, status` and `timestamp` when reporting changes. In addition, vendor-specific fields such as `vendorId, vendorErrorCode`, and/or `info` can be included. The vSECC Controllers support the reporting of additional, vendor-specific information to the CSMS in the Status Notification message. Therefore, the vendor-specific information is to be provided via MQTT. The default fields are still filled-in by the vSECC Controller. See section J for a detailed description of the MQTT topics:

> `vsecc\status\report_status_information`

> `vsecc\connector\{evse_id}\status\report_status_information`

Every time something is published on one of these MQTT topics, sending an OCPP Status Notification message is triggered.

8.26.2 OCPP 2.0.1

OCPP 2.0.1 specifies several Use Cases for Monitoring (in combination with the OCPP Device Model) as an alternative to Status Notifications. Status messages can only be used for the *Connector* component. Other components such as *EVSE* or *Charging Station* are always reported with Monitoring Events. The choice of how to report connector changes can be configured via the web interface Section **Configuration / Charging Station Management System / Expert Functions**. Vendor-specific fields as in OCPP 1.6 are not available.

8.27 Customization Possibilities with Software Container Solution

8.27.1 Introduction

The vSECC Controllers allow runnig customer-specific software in a containerized environment. Vector provides the **Configurable Customer Interface**, an installable container with the open-source solution Node-RED (for low-code programming software using a browser-based editor) running. Such tool offers flexibility and accessibility but can also present limitations. Kindly, refer to the Application Note for vSECC CCI on the recommended practices using Node-RED from our Vector Customer Portal at portal.vector.com. Further information regarding Node-RED can be also found at the project website. Alternativly, Vector currently establishes the **vSE Developer Program**, in which it provides a development kit to create and deploy own containers.

The ”vSE Developer Program” is currently in a prototypical stage and only available for pilot customers.

Only one container can be installed and running at the same time.

8.27.2 Data Storage

Container related data is stored in two different places. The container with the Root File System itself is stored in the Container Runtime Storage. This storage is limited to 750 MB in total. Please make sure not to exceed this space when writing data into the `rootFS` of your container during runtime. To clear data stored by the container inside the `rootfs`, press **[Restart Container]** in the Container Management Section of the web interface.
Also, don’t use it to store persistent data. Persistent data can be stored in the `/data` folder. All data stored by the container in the `/data` folder can be removed alongside the installed container by pressing **[Delete Image]** in the Container Management Section of the web interface.
Pressing **[Clear Persistent Data]** in the Container Management Section of the web interface clears all the data stored in the `/data` folder by the container.
Furthermore, you are able to access the log files of the vSECC Controller with the `/secclog` directory in your container.

Caution: Filling all the space of the Persistent Data Storage can seriously harm the functionality of the vSECC Controller. Please make sure the storage is never completely used, in addition please clear logging data. Be aware that the Persistent Data Storage is shared with the internal software of the vSECC Controller, which will also fill up by the storage.

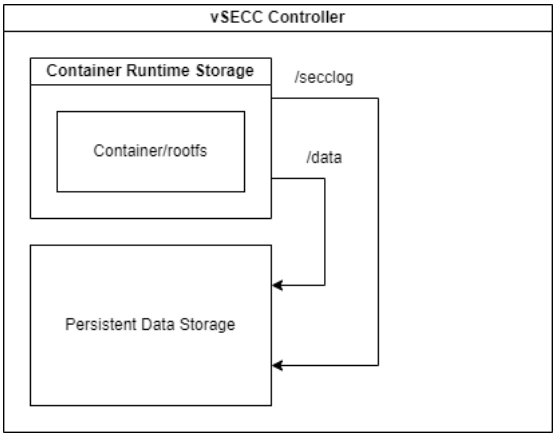


Figure 117: Container Storage Concept

When using the Configurable Customer Interface container image provided by Vector, the persistent data (e.g. Node-RED flows) is already stored in the Persistent Data Storage.

8.27.3 Managing Containers

How to manage a container in the web interface is described in the Configuration Guide in chapter 7.4.

8.27.4 Networking

Networking for the container is only possible with IPv4.

A virtual ethernet network interface called veth1 is available in the container. This network interface is connected to a virtual network interface on the vSECC Controller's host system. The IP address of the container is always set to `172.17.0.2/24`. This can not be changed. Please make sure that the physical network interfaces are not configured to use the `172.17.0.0/24` Subnet. The vSECC Controller's host IP address is fixed to `172.17.0.1/24`. External network endpoints can connect to the container using reserved ports shown in the table below.

External Port	Container Port	Description
25022	22	Port forwarding for well-known SSH port
25080	80	Port forwarding for well-known HTTP port
25443	443	Port forwarding for well-known HTTPS port
25030 - 25039	25030 - 25039	Port forwarding for general purpose (TCP only)
25040 - 25049	25040 - 25049	Port forwarding for general purpose (UDP only)

Caution: All exported ports are forwarded unprotected into the container. It is essential to restrict the access to the network interfaces in physical and in logical/firewalled way. The ethernet interfaces must not connect to any unfirewalled network.

To establish a connection to an external network participant from the container, make sure the network participant is in the same subnet as the vSECC Controller, or configure a default gateway as described in Section 7.3.4.

Caution: Inside the container, no name resolution over DNS is configured by default. For Node-RED, this implies that it is not possible to install extensions.

8.27.5 CAN

It is possible to use the CAN interface of the vSECC Controller with software in the container. Proprietary CAN messages can be sent and received via the socket CAN node functionality of the container. The interface / device name for communication with CAN in the container is called **vxcan1**. The mapping of the corresponding physical interface is described in section 8.27.9 ff.

The CAN interface can be used in the container even if PEP-CAN is used on the same connector.

The CAN TX queue for **vxcan0** is limited to 100 CAN frames. This is the number of messages queued for transmission or awaiting transmission completion.

The baud rate of the CAN interface is configured in the **Configuration / Hardware Interfaces / CAN** section of the web interface. The baud rate settings are the same for container CAN and PEP communication (if PEP-CAN is used).

Figure 118: CAN Baud Rate Settings in the Web Interface

Caution: Please make sure not to interfere with the PEP communication if PEP-CAN is used.

8.27.6 RS232

It is possible to use the RS232 interface of the vSECC Controller with software in the container. RS232 serial messages can be sent and received via the serial node functionality of the container. The interface / device name for serial communication in the container is called **/dev/ttymxc4** or **/dev/ttymxc3**, dependent which vSECC Controller is used. The mapping of the corresponding physical interface and the usage of the device name is described in section 8.27.9 ff.

The usage of the RS232 interface is configured in the **Configuration / Hardware Interfaces / RS 232** section of the web interface. The default setting therefore is **None**. For RS232 usage in the container, the settings must be switched to **Container**. All other RS232 settings like baud rate, data bits, parity and stop bits must be done within the container (e.g. in Node-RED).

Figure 119: RS232 Usage Configuration in Web Interface

The RS232 settings are configured directly in the container (e.g. in Node-RED).

Caution: Changing this parameter requires a reboot. This will automatically be done, when the parameter is stored. When a device is set (or removed) to be used in the container, the container will be re-created on startup. This will delete all data in the container that is not stored in the persistent storage.

8.27.7 RS485

RS485 pin polarity: in the vSECC product family, the RS485 pin labled A always marks the noninverting pin, whereas B marks the inverting pin.

It is possible to use the RS485 interface of the vSECC Controller with software in the container. RS485 serial messages can be sent and received via the serial node functionality of the container. The interface / device name for serial communication in the container is called **/dev/ttymxc3** or **/dev/ttymxc2**, dependent which vSECC Controller is used. The mapping of the corresponding physical interface and the usage of the device name is described in section 8.27.9 ff.

The usage of the RS485 interface is configured in the **Configuration / Hardware Interfaces**

/ RS 485 section of the web interface. The default setting therefore is **Modbus Bridge**. For RS485 usage in the container, the settings must be switched to **Container**. All other RS485 settings like baud rate, data bits, parity and stop bits must be done within the container (e.g. in Node-RED).

Figure 120: RS485 Usage Configuration in Web Interface

The RS485 settings are configured directly in the container (e.g. in Node-RED).

Caution: Changing this parameter requires a reboot. This will automatically be done, when the parameter is stored. When a device is set (or removed) to be used in the container, the container will be re-created on startup. This will delete all data in the container that is not stored in the persistent storage.

8.27.8 Introduction to the example flows in Node-RED as part of the Configurable Customer Interface

The Configurable Customer Interface Feature is realized with Node-RED.

Caution: The provided flows are only an example for reference. Vector can not guarantee proper operation of the charging station using the provided example flows. It's up to the customer to properly implement, verify and test the flows that are running on the vSECC Controller. Furthermore, please be aware that safe operation has to be ensured using the provided hardware supervision and further measures that are needed according to IEC 61851 and local regulations.

Vector offers an example flow to get familiar with Node-RED and interfacing with the vSECC Controller. To view the Node-RED interface navigate to the `<IP-Address of vSECC Controller>/node-red/` URL (e.g. `http://192.168.3.11/node-red/`). When first opening the Node-RED interface, you will see a welcome message as shown in Figure 121. If you are not yet logged in, you will be automatically redirected to the login page 7.4.

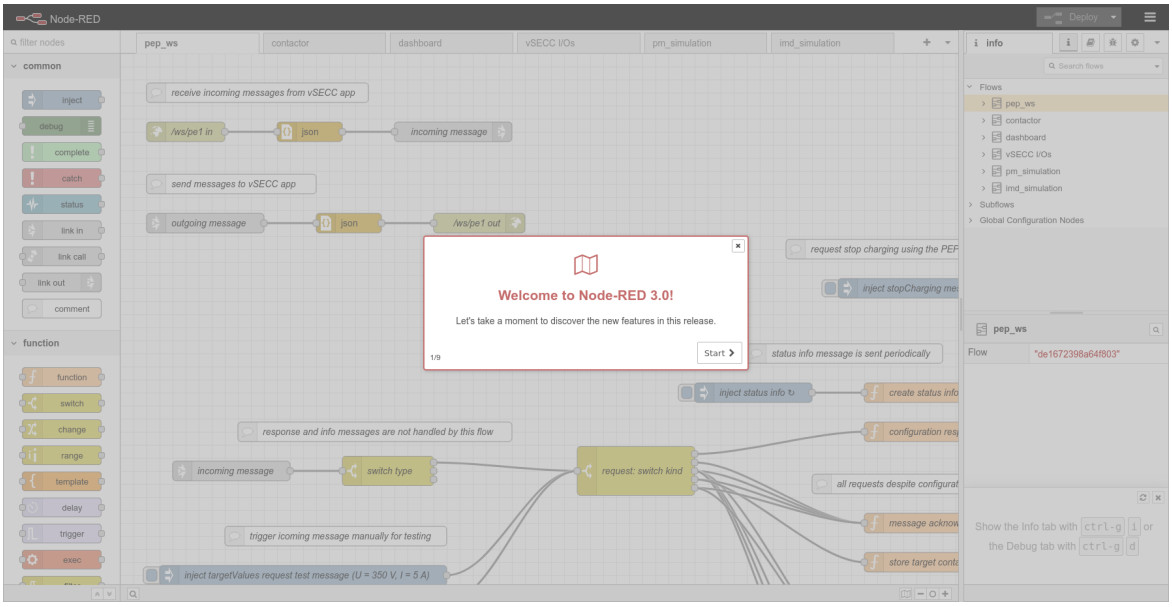


Figure 121: Node-RED welcome page

The provided example contains the following flows:

- **pep_ws**: handles the communication with the vSECC Software application
- **pm_simulation**: simulates a power module by reading in the target values from pep_ws and storing them as measured values
- **imd_simulation**: simulates an insulation monitoring device
- **contactor**: shows how to monitor different data sources to create an additional signal to actuate the contactors
- **vSECC I/Os**: provides an example flow on how to read inputs and set outputs using MQTT
- **dashboard**: an example dashboard displaying relevant data during a charging session. The dashboard can be accessed on `http://<ip of vsecc>/node-red/ui`.

Furthermore, example hardware implementation flows, e.g. of the LEM Energy Meter, can be downloaded from the portal.
The interaction between the different flows is done by storing the data in the global Node-RED context. Revise the User Guide[2] fore more details. Furthermore, a message with a link out (that can be linked inside other flows) is provided. The message is sent when values are updated. Therefore, each flow determines whether it wants to read out data from other flows periodically, when an internal event occurs or when the values of a module have been updated. The architecture of the flows that are relevant for charging communication is shown in Figure 122.

[2] https://nodered.org/docs/user-guide/context

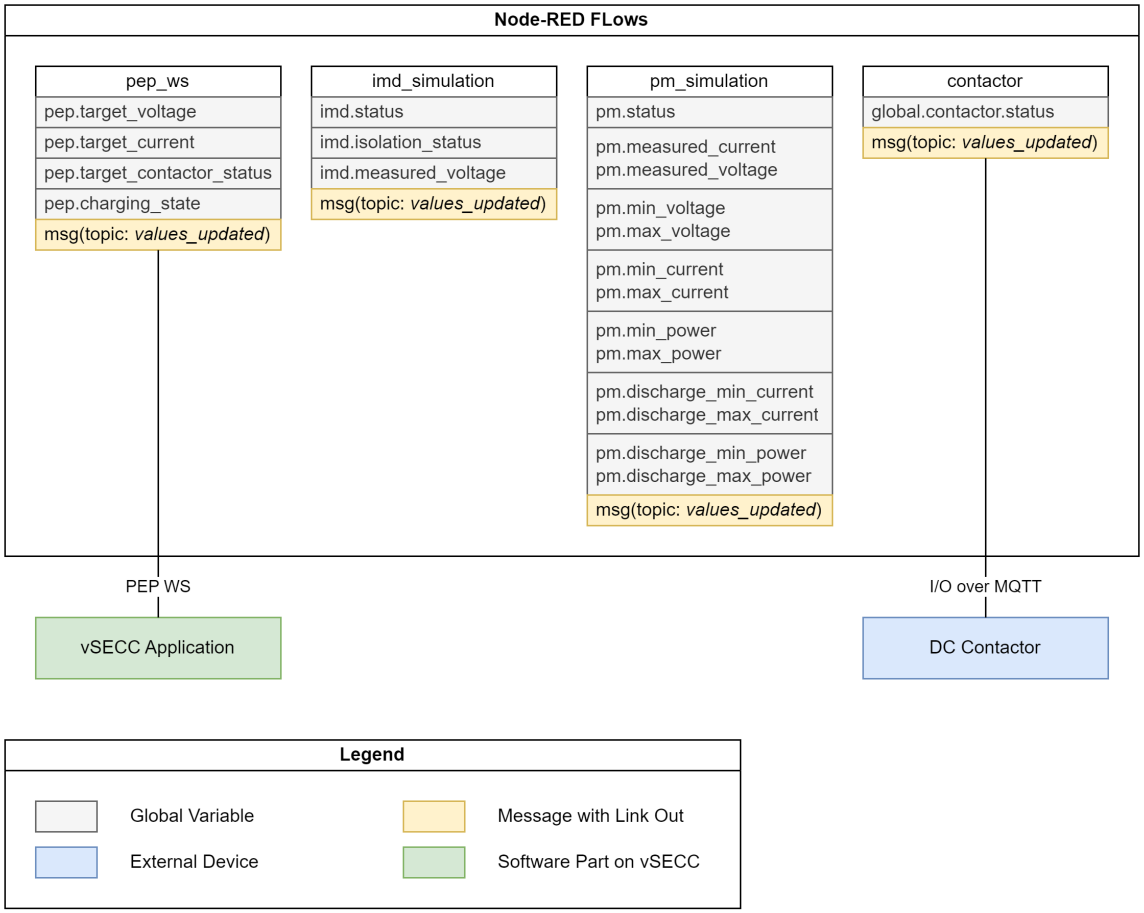


Figure 122: Node-RED flow architecture

The interface of the vSECC Software Application to the Node-RED Flows is via PEP-WS and MQTT. In order to use the example flows on the vSECC Controllers, the following configuration needs to be done in the web interface:

> Credentials for MQTT must be set (see Section 8.13)
> The **Protocol** of the **Configuration / Power Electronics** must be set to **Websocket**
> The **PECC URL** of the **Configuration / Power Electronics / Websocket** must be set to `http://192.168.3.11/node-red/ws/pe1`

Furthermore, in the MQTT configuration node in Node-RED, the credentials for connecting to the MQTT Broker must be set as shown in Figure 123 and described below:

1. Select the **config** tab on the panel on the right
2. Double-click the **vsecc-mqtt** configuration node
3. Open the **Security** tab
4. Enter the credentials previously configured in the vSECC Controller's web interface
5. Click **Update**
6. Deploy the changes by clicking the **Deploy** button

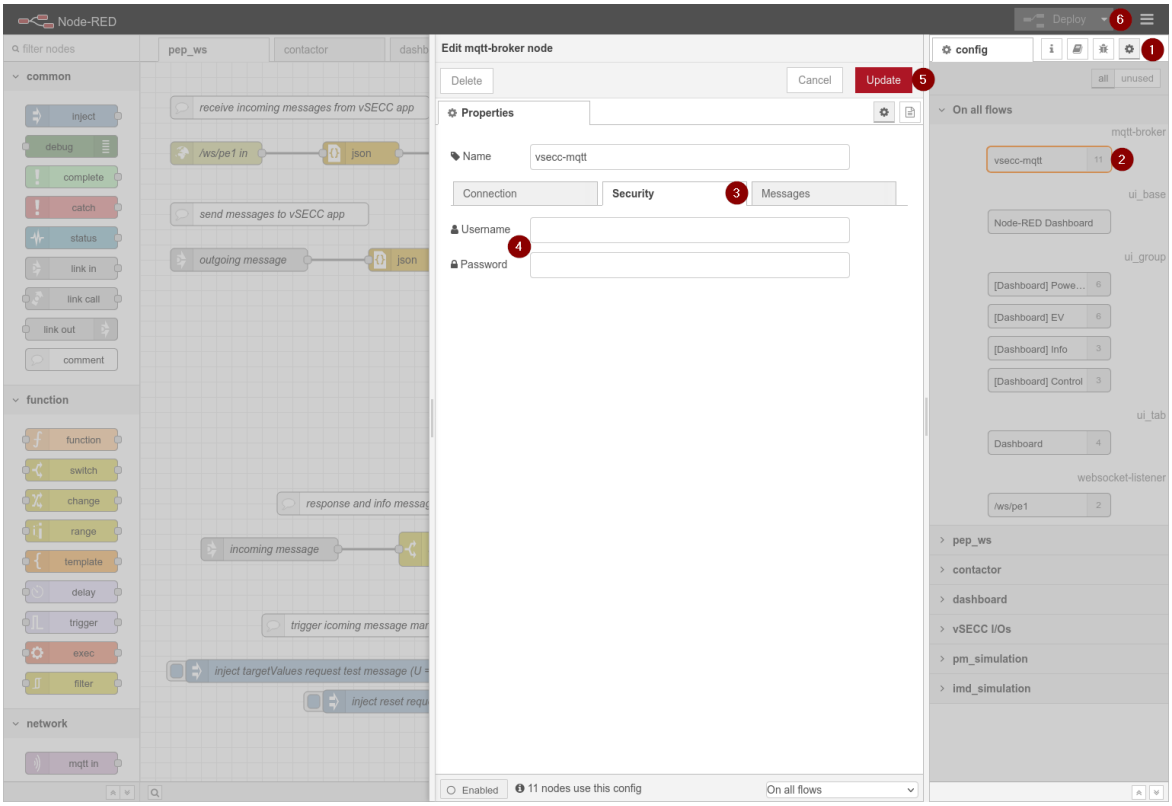


Figure 123: Configuring the MQTT node in Node-RED

Further hints about the flow implementation can be found in comments inside the provided flows.

8.27.9 Quick Reference: Devices and Interfaces on vSECC and vSECC.MCS

Interface/Device Name in Container	Corresponding Physical Interface	Notes
veth1	ETH1, ETH2	Internal IP address of container: 172.17.0.2, internal IP address of vSECC 172.17.0.1
ttymxc3	RS485 (X305.7, X305.9)	Please make sure to enable termination resistor
ttymxc4	RS232 (X305.8, X305.10)	
vxcan1	CAN2 (X305.2, X305.4)	Can be used in container even if PEP-CAN is used on the same connector

8.27.10 Quick Reference: Devices and Interfaces on vSECC.single Board

Interface/Device Name in Container	Corresponding Physical Interface	Notes
veth1	Ethernet	Internal IP address of container: 172.17.0.2, internal IP address of vSECC.single Board: 172.17.0.1
ttymxc2	RS485 (X300.24, X300.26)	
ttymxc3	RS232 (X300.27, X300.29)	
vxcan1	CAN1 (X300.23, X300.25)	Can be used in container even if PEP-CAN is used on the same connector

8.27.11 Quick Reference: Devices and Interfaces on vSECC.single

Interface/Device Name in Container	Corresponding Physical Interface	Notes
veth1	Ethernet	Internal IP address of container: 172.17.0.2, internal IP address of vSECC.single: 172.17.0.1
ttymxc2	RS485 (X307.1, X307.2)	
ttymxc3	RS232 (X306.1, X306.2)	
vxcan1	CAN (X307.3, X307.4)	Can be used in container even if PEP-CAN is used on the same connector

8.28 Disable ISO15118-2 Renegotiation

When charging according to ISO15118-2, a renegotiation may occur. The rough sequence is as follows:

1. ISO15118-2 demands that a change of charge parameters must be negotiated between the EV and EVSE. Among others, the charge parameters contain the charging schedule which is derived from the charging profiles.
2. The charging session is set up, said parameters are exchanged and confirmed in the ChargeParameterDiscovery (CPD) message pair. The charging session is then started by the EV by sending a PowerDeliveryRequest.
3. EV and EVSE are in the charge loop consisting of frequent exchanges of CurrentDemandRequests (sent by the EV) and CurrendDemandResponses (sent by the EVSE).
4. A new charging profile is sent by the CSMS to the EVSE via OCPP. This profile may change the maximum power the EV is allowed to charge with.
5. The EVSE sets a specific flag in its CurrentDemandResponses which requests the EV to trigger a renegotiation.
6. The EV interrupts the CurrentDemand message exchange and the charge parameters are negotiated again. This renegotiation leads to another exchange of a possibly updated charging schedule.
7. The EV starts the charging session again and both entities enter the charge loop.

Although not required by the ISO15118-2 standard, some EVs open their HVDC contactors during a renegotiation, while other EVs allow only a small and limited number of renegotiations during a single charging session. This happens mainly to prolong the life of the contactors by reducing the number of open/close operations.

When the charging power is changed very frequently by sending new charging profiles to the vSECC, a lot of renegotiations occur. With such EVs, this may lead to increased contactor wear or to charging session abortions.
A configuration option exists to disable ISO15118-2 renegotiations completely. While charging profiles sent over OCPP are still accepted and respected when sending control data to the power electronics (target values), the EV doesn't get the new charging profile because no renegotiation happens. This in turn may lead to unwanted behavior because a charging power limit communicated through a schedule at the begin of a charging session cannot be increased but only be reduced during the charging session (except through a renegotiation). To accommodate for this, the vSECC sends a special charging schedule to the EV if renegotiations are disabled. This special schedule consists of the power electronics' maximum values (communicated via PEP) as limits.

Please note: Some EVs abort the charging session if the values for power/current values from the schedule communicated in the CPD differ too much from the actually measured values. To mitigate this, the vSECC sets the optional values EVSEMaximumVoltageLimit, EVSEMaximumCurrentLimit and EVSEMaximumPowerLimit in the CurrentDeman-

dResponse message according to the charging profile received via OCPP (and NOT according to the dummy profile from the CPD).

8.29 Session Suspension with 0W Charging Profiles

This feature is a workaround for vehicles which do not properly support renegotiation or are limited to DIN SPEC 70121 (see chapter 8.28 for more details).
When enabled, the charging communication stops as soon as a 0W period in the charging schedule starts. Once the 0W period ends, the vSECC Controller tries to wake up the EV with a BEB toggle.
It might be also useful to configure a timeout to wait in ChargeParameterDiscovery/ScheduleExchange for a charging profile, if TxProfiles must be applied before power is requested from the power electronics.

This feature is available only for DIN SPEC 70121, ISO 15118-2 and ISO 15118-20

8.30 Controllable Delay at the Beginning of Charging Session (CPD/SE)

It may be necessary to delay the continuation of a charging session after a communication channel to the EV has been established before the HVDC connection is live. Depending on the charging standard, this is possible during the ChargeParameterDiscovery (DIN70121, ISO15118-2, SAE J3105) or ScheduleExchange (ISO15118-20) loop.
The vSECC publishes the maximum remaining time in seconds over MQTT on the topic `vsecc/connector/evse_id/status/waiting_for_power_electronics_ready`.
The charging session is continued when this time reaches zero or if the vSECC receives an empty string on the topic `vsecc/connector/evse_id/status/report_power_electronics_ready`, whichever comes first.
After the published remaining loop time is not valid anymore, be it due to an error or the charging session being continued, the empty string is published.
This feature is disabled by default and could be enabled via the WebUI (Section Configuration/Vehicle/Expert Functions) or REST-API. The boolean configuration variable is called `wait_for_power_electronics_ready_command`.

8.31 Send Custom Error Codes To CSMS

In general, an error code provides information about an error or a significant event that has occurred within the charging system, which is then communicated to a backend via OCPP. While errors that can be detected by the vSECC Controller are reported automatically, the MQTT interface offers the possibility to send any type of custom error code to a backend. In principle, the handling of an error code can be divided into three phases.

> **Phase 1: Detect error and translate into error code**
The vSECC Controller provides data through various interfaces, such as MQTT and

PEP. Errors can be derived from this data by the user. The error code is defined by the user and selected by the user according to an significant event or an error. Any information not provided by the vSECC Controller must be independently recorded. For more information on the MQTT topics and their description, please refer to appendix J and O.

Caution: The vSECC Controller does not interpret data for the purpose of translating it into error codes

The vSECC Controller acts as an information provider.

> **Phase 2: Transmit error code to vSECC Controller**
Once an error has been detected, it should be communicated to a backend. The vSECC Controller has subscribed to two topics for this purpose:
  > **ChargingStation events:**
  `vsecc/status/report_status_information`
  > **EVSE/Connector specific events:**
  `vsecc/connector/{evse_id}/status/report_status_information`

> **Phase 3: vSECC Controller creates and sends OCPP message**
After receiving a `report_status_information` via MQTT, a message is generated by the vSECC Controller and sent to the backend. The generated message depends on the active OCPP version. In the case of OCPP 1.6, a StatusNotificationRequest is sent. The fields `vendorId`, `vendorErrorCode`, and/or `info` are filled with the information from the `report_status_information`. Additional information is provided in section 8.26. In OCPP 2.0.1, this message is replaced by a `NotifyEventRequest`, which includes the fields `techCode` and `techInfo`. We define the ”techInfo” to contain a semicolon separated list of the ChargePointErrorCode, vendor ID and the info. The ”techCode” contains the vendor error code. A `NotifyEventRequest` must reference a device model variable. To ensure interoperability and reduce complexity, we limit error reporting to the standard components ”ChargingStation”, ”EVSE”, and ”Connector”, using a variable named `Problem`. This method eliminates the need for proprietary components for each error type and aligns with the components and variables specified in OCPP 2.0.1.

Example sequence

Phase 1: A high temperature is detected at power electronics (PE).

Phase 2: Publish JSON String for EVSE 1 to
`vsecc/connector/1/status/report_status_information`

```
1 {
2   "errorCode": "HighTemperature",
3   "vendorId": "Error Corp.",
4   "vendorErrorCode": "VX04",
5   "info": "PE temperature: 90C"
```

```
}
```

Phase 3: OCPP message result generated by vSECC Controller depending on active OCPP version.

OCPP1.6 StatusNotificationRequest:

```
{
  "connectorId": 1,
  "errorCode": "HighTemperature",
  "info": "PE temperature: 90C",
  "status": "Available",
  "timestamp": "2025-06-24T14:36:26.572Z",
  "vendorErrorCode": "VX04",
  "vendorId": "Error Corp."
}
```

OCPP2.0.1 NotifyEventRequest:

```
{
  "eventData": [
  {
    "actualValue": "true",
    "component": {
      "evse": {
        "id": 1,
        "connectorId": 1
      },
      "name": "Connector"
    },
    "eventId": 20,
    "eventNotificationType": "HardWiredNotification",
    "techCode": "VX04",
    "techInfo": "HighTemperature;Error Corp.;PE temperature: 90C",
    "timestamp": "2025-06-24T14:23:51Z",
    "trigger": "Alerting",
    "variable": {
          "name": "Problem"
    }
  }
  ],
  "generatedAt": "2025-06-24T14:23:51.112Z",
  "seqNo": 0
}
```

9 Service Guide

In this chapter you will find the following information:

9.1 Reset Factory Defaults

The vSECC Controllers are equipped with a reset button, see Section 2.3.1 (vSECC), Section 3.3.1 (vSECC.MCS), Section 4.3 (vSECC.single Board) and Section 5.3 (vSECC.single) for its exact location.

To reset the device to the factory settings perform the following actions:

1. Power off the device (remove power plug)
2. Press and hold the reset button
3. Power on the device
4. Keep on holding the button until
   > vSECC and vSECC.MCS: All 4 LEDs on the housing blink 4 times
   > vSECC.single (+70°C and Board): The LED blinks red 4 times
5. Release the button

After the reset, the LEDs behave as follows:

> vSECC and vSECC.MCS: All 4 LEDs remain solid red

> vSECC.single (+70°C and Board): The LED remains solid orange (red and green both active)

Then, please wait until the System LED is solid green. Now, your vSECC Controller is fully operational with the factory settings in place.

Caution: Pressing the reset button deletes all custom configuration data permanently, including certificates and timezone settings. If possible, make a backup prior to the reset.

9.2 Firmware Update

The process how to update the firmware is described in detail in chapter 7.8.3.

9.3 Status LEDs

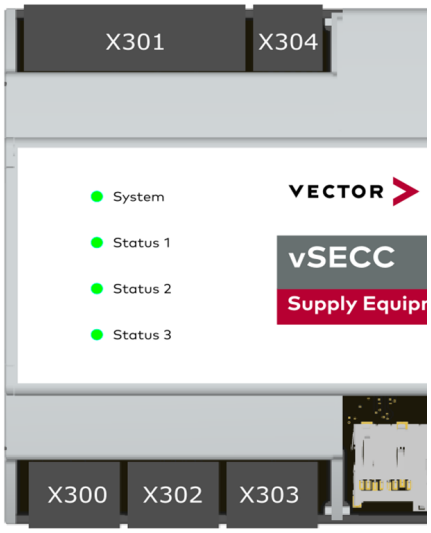


Figure 124: vSECC / vSECC.MCS

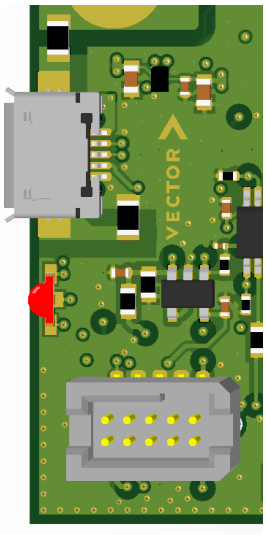


Figure 125: vSECC.single Board

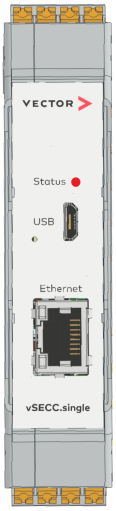


Figure 126: vSECC.single

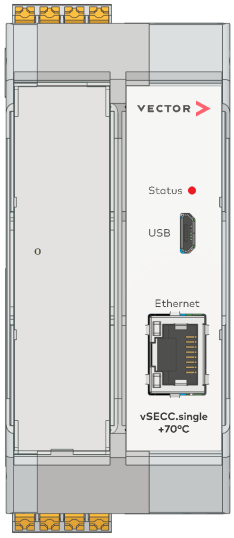


Figure 127: vSECC.single +70°C

The vSECC Controllers are equipped with a System LED that indicates the status of the controller. The vSECC / vSECC.MCS has three additional LEDs representing the status of the respective charging port. Each LED has multiple possible states, as described in detail below.

System LED

This LED shows the overall system status of the vSECC Controller:

- Off: No power is provided.
- Orange, flashing: Currently not used.
- Orange, continuously: Early boot phase.
- Green, flashing: Late boot phase.
- Green, continuously: Initialization is complete and the vSECC Controller is now running.
- Red, flashing: Indicates the special events reboot and factory reset.
- Red, continuously: The vSECC Controller is in maintenance mode.

Please note that the **vSECC / vSECC.MCS**'s System LED may briefly flash green when power is first connected. This should be ignored.

Caution: While the vSECC.single Board indicates the end of the boot process, it may take a few seconds until the controller is fully operational and starts communication with the power electronics, EV or the CSMS.

Charging Port LEDs (vSECC and vSECC.MCS only)

Three Status LEDs as shown in Figure 124 indicate the current state of the respective charging port.

vSECC: The LED `Status 1` corresponds to connector X303 (CCS 1), `Status 2` corresponds to connector X302 (CCS 2), and `Status 3` corresponds to connector X300 (CHAdeMO).

vSECC.MCS: The LED `Status 1` corresponds to connector X200 (MCS), `Status 2` corresponds to connector X302 (CCS), and `Status 3` is not used and reserved for future use.

Following states are possible for vSECC and vSECC.MCS:

- Off: Connector works, but currently no EV is connected.
- Green, flashing: The connected EV is currently charging.
- Green, lit continuously: An EV is connected.
- Red, flashing: Currently not used.
- Red, continuously: The connector is inoperative. This may be due to a setting in the CSMS or an internal error.

9.4 Reporting Security Issues

Security issues can be reported via Vector's PSIRT. For more information and best results please follow the Coordinated Vulnerability Disclosure policy defined in our security manual. The security manual is shipped alongside the user manual.

10 Technical Data vSECC

In this chapter you will find the following information:

10.1 General

Parameter	Min.	Typ.	Max.	Unit
Supply voltage V_{in}	18	24	30	V
Power consumption at $24\,V$ > Idle, communication to back end and power electronics active > Charging CCS on two spots, communication to back end and power electronics active > Charging CCS and CHAdeMO simultaneously, communication to back end and power electronics active > 5 digital ouputs at max load, communication to back end and power electronics active > All digital ouputs at max load, CHD_SEQ at max load, CHD_LATCH at $25\,\Omega$, CPU at $100\,\%$ load		 7 7 36 30 107		 W W W W W
Current draw during boot-up			1.5	A
Temperature range	-40		70	°C
Humidity	non-condensing			
Altitude (above sea level)	0 - 2000			m
Dimensions (length x width x depth)	161.6 x 89.7 x 60.7			mm
Total weight	approx. 276			g
IP protection class	20			

10.2 Digital Inputs

6 general purpose digital inputs + 2 inputs used for CHAdeMO start/stop buttons (IEC 61131-2 Type 1 & 3 compatible, active high):

Parameter	Min.	Typ.	Max.	Unit
Input voltage		24		V
Switching thresholds > High to Low > Low to High		 4.4 6.0		 V V
Current draw per input	2.1		2.6	mA

10.3 Digital Outputs

16 general purpose digital outputs (active high):

Parameter	Min.	Typ.	Max.	Unit
Output voltage (High)		V_{in} - 1	V_{in}	V
Output current per channel			200	mA
Total output current all channels			3.2	A
Each output is overcurrent and short-circuit protected				

10.4 Analog Inputs

2 general purpose analog inputs:

Parameter	Min.	Typ.	Max.	Unit
Input voltage	0		10	V
Resolution	12			bit

The analog inputs are not calibrated on every single device and therefore not usable for high-precision measurements.

10.5 Temperature Inputs

9 temperature sensor inputs, optimized for usage with PT-1000 temperature sensors:

Parameter	Min.	Typ.	Max.	Unit
Driven output current		400		μA
Resolution	16			bit

The temperature inputs are not calibrated on every single device and therefore not usable for high-precision measurements.

10.6 Safety Outputs

3 isolated relays (normally opened):

Parameter	Min.	Typ.	Max.	Unit
Rated current			100	mA
Switching voltage (DC)			30	V
Delay time		3.6	5	ms

10.7 Serial Communication

10.7.1 CAN1 (CHAdeMO)

CAN port with switchable bus termination:

Parameter	Min.	Typ.	Max.	Unit
Baudrate		500		kbit/s

10.7.2 CAN2 (Power Electronics)

CAN port with switchable bus termination:

Parameter	Min.	Typ.	Max.	Unit
Baudrate		500		kbit/s

10.7.3 RS485 (Modbus TCP to RTU gateway)

Configuration parameter:

Parameter	Min.	Typ.	Max.	Unit
Baudrate		9600		kbit/s
Number of data bits		8		
Number of stop bits		1		
Parity mode		even		

10.8 CCS Connectors

10.8.1 Full Bridge Out

Output voltage switchable via software (overcurrent and short-circuit protected):

Parameter	Min.	Typ.	Max.	Unit
> 24 V mode selected		V_{in}- 1.7	V_{in}	V
> 12 V mode selected	10	12	14	V
Output current for 2 seconds			2	A

10.8.2 Full Bridge Feedback

Parameter	Min.	Typ.	Max.	Unit
Output resistance	0		15	Ω
> Connector locked		11		$k\Omega$
> Connector unlocked		1		$k\Omega$

10.8.3 Control Pilot

2 control pilot pins (designed according to IEC 61851):

Parameter	Min.	Typ.	Max.	Unit
Output voltage				
> On state		12		V
> Off state		-12		V
Frequency	0.98	1	1.02	kHz
Duty cycle accuracy	+/- 5			μs

10.9 CHAdeMO Sequence Circuit

Sequence circuit signals (designed in accordance with CHAdeMO v0.9.1 and v1.2.0 ED2):

Parameter	Min.	Typ.	Max.	Unit
Charge sequence signal 1				
> On state voltage	11.65	12	12.35	V
> Driven continuous output current			2	A

Charge sequence signal 2				
> Open drain output				
> Maximum input voltage			0.35	V
> Maximum continuous input current			2	A
Connector proximity detection				
> Pull-down resistor		200		Ω
Vehicle charge permission				
> Pull-up resistor		1		kΩ
> External leakage current			2	mA
Latch Out				
> On state voltage	11.65	12	12.35	V
> Output Current			500	mA
Latch In				
> Input Current	200		500	mA

10.10 Real Time Clock (RTC)

The backup battery pin connects to an external super-capacitor. It powers the real time clock during power loss. This ensures that the real time clock remains functional and synchronized after boot up:

Parameter	Min.	Typ.	Max.	Unit
Super-Capacitor voltage			3.1	V
RTC current (during power loss)		1.5		μA
Battery Backup time			24	h

11 Technical Data vSECC.single Board

In this chapter you will find the following information:

11.1 General

Parameter	Min.	Typ.	Max.	Unit
Supply voltage V_{in}	11	12	13	V
Power consumption at $12\,V$ > idle, communication to back end and power electronics active		5		W
Temperature range	-40		70	°C
Humidity	non-condensing			
Altitude (above sea level)	0 - 2000			m
PCB Dimensions (further details in mechanical drawing)	94.10 x 58.00			mm
Total weight	approx. 48			g

11.2 Digital IO's

10 software-programmable general purpose digital IO's with VCC_LOGIC power supply output for opto-couplers and level shifters:

11.2.1 Digital Inputs

Parameter	Min.	Typ.	Max.	Unit
Input voltage		3.3	3.6	V
Switching thresholds				
> High to Low			0.8	V
> Low to High	2.0			V
Current draw per input	-8		8	µA

11.2.2 Digital Outputs

Parameter	Min.	Typ.	Max.	Unit
Output voltage (High)		3.2		V
Output voltage (Low)		0.1		V
Output current per channel			3	mA
Each output short-circuit (to GND) protected				

11.2.3 VCC_LOGIC Power Supply

Parameter	Min.	Typ.	Max.	Unit
Output Voltage	3.27		3.55	V
Output Current			30	mA

11.3 Analog Inputs

4 general purpose analog inputs:

Parameter	Min.	Typ.	Max.	Unit
Input voltage	0		5	V
Resolution	12			bit
VCC_ADC can be used as reference voltage.				

11.4 Temperature Inputs

2 temperature sensor inputs, optimized for usage with PT-1000 temperature sensors, supervised by safety output:

Parameter	Min.	Typ.	Max.	Unit
Measurement current	970	1000	1030	μA
Measurement range	815		1600	Ω
Resolution	12			bit

11.5 Safety Output

One digital output. Supervision of CP, PP and temperature sensors always active. Output is **OFF** when energy transfer is not allowed (CP State in states A/B/F/E and/or temperature inputs not in range). Output is **ON** when energy transfer is allowed (CP state in states C/D and temperature inputs in range).

Caution: In order to deactivate supervision of PP (e.g. when using CCS Type 2 Connector), connect PP_PU (X301.2) with PP (X301.1) and add a Resistor of $142\,\Omega$ between PP and GND (X301.4-6).

Caution: You can not use a pull-up on this output.

Parameter	Min.	Typ.	Max.	Unit
Max. output current			10	mA
Output voltage (active)	3.14	3.3		V
Delay time (CP loss, PP loss)		3.6	5	ms
Upper treshold temperature input (resistance)	1333		1359	Ω
> Equivalent temperature	86.3		93.1	°C
Lower threshold temperature input (resistance)	802		816	Ω
> Equivalent temperature	-50.3		-46.7	°C

11.6 Serial Communication

11.6.1 CAN1 (Power Electronics)

CAN port with fixed bus termination:

Parameter	Min.	Typ.	Max.	Unit
Baudrate		500		kbit/s

11.6.2 RS485 (Modbus TCP to RTU gateway)

Configuration parameter:

Parameter	Min.	Typ.	Max.	Unit
Baudrate		9600		kbit/s
Number of data bits		8		
Number of stop bits		1		
Parity mode		even		

11.7 CCS Connector Control Pilot

1 control pilot pin (designed according to IEC 61851):

Parameter	Min.	Typ.	Max.	Unit
Output voltage				
> On state		12		V
> Off state		-12		V

Frequency	0.98	1	1.02	kHz
Duty cycle accuracy	+/- 5			μs

11.8 Real Time Clock (RTC)

The backup battery pin connects to an external super-capacitor. It powers the real time clock during power loss. This ensures that the real time clock remains functional and synchronized after boot up:

Parameter	Min.	Typ.	Max.	Unit
Super-Capacitor voltage			3.3	V
RTC current (during power loss)		0.25		μA
Battery Backup time			240	h

12 Technical Data vSECC.single

In this chapter you will find the following information:

12.1 General

Parameter	Min.	Typ.	Max.	Unit
Supply voltage V_{in}	11	12	13	V
Power consumption at $12\,V$ > idle, communication to back end and power electronics active		5		W
Operating temperature	-40		50	°C
Storage temperature	-40		70	°C
Humidity	non-condensing			
Altitude (above sea level)	0 - 2000			m
PCB Dimensions (further details in mechanical drawing)	114.50 x 99.00			mm
Total weight	approx. 140			g

12.2 Digital IO's

12.2.1 SAF_FB (Safety Feedback)

The SAF_FB (Safety Feedback) is defined as input signal at connector X301.1. This signal is connected with the vSECC.single Board's digital input DIO5 through the Samtec connector:

Parameter	Min.	Typ.	Max.	Unit
Input voltage		3.3	V_{in}	V
Switching threshold		1.57		V
Current draw		3.3		mA

Caution: The digital input (SAF_FB) is populated with a pull up resistor to 3.3V. When the connector X301.1 is not connected, the voltage of the digital input is pulled up to the level of 3.3V. In this case the digital input DIO5 shows a high value. When the connector X301.1 is connected to GND, the digital input DIO5 goes to a low value.

12.2.2 DIN1, DIN2 (Digital Input 1 & 2), DIN_FB (Digital Input Feedback)

DIN1 (Digital Input 1), DIN2 (Digital Input 2) and DIN_FB (Digital Input Feedback) are defined as input signals. They are connected with the vSECC.single Board's DIOs through the Samtec connector as follows:

> DIN1 (X301.2) is connected to digital input DIO6
> DIN2 (X301.3) is connected to digital input DIO7
> DIN_FB (X301.4) is connected to digital input DIO9

Parameter	Min.	Typ.	Max.	Unit
Input voltage		5	V_{in}	V
Switching threshold		1.57		V
Current draw		0.51		mA

Caution: The digital inputs (DIN1, DIN2 and DIN_FB) are populated with a pull up resistor to 5V. When the connector is not connected, the voltage of each digital input is pulled up to the level of 5V. In this case the digital input shows a high value. When the connector is connected to GND, the digital input goes to a low value.

12.2.3 HB1_OUT (Half Bridge Output 1), HB2_OUT (Half Bridge Output 2)

HB1_OUT (Half Bridge Output 1) and HB2_OUT (Half Bridge Output 2) are defined as output signals. They are connected with the vSECC.single Board's DIOs through the Samtec connector as follows:

> HB1_OUT (X303.3) is configured via digital output DIO1
> HB2_OUT (X303.4) is configured via digital output DIO4

Parameter	Min.	Typ.	Max.	Unit
Output voltage (High)		V_{in} - 0.75	V_{in}	V
Switching (Delay time)				
> High to Low		1		μs
> Low to High		1		μs
Rated current per channel		1		A
Each output is short-circuit proof				

12.2.4 HS_OUT (High Side Switch Output)

HS_OUT (High Side Switch Output) is defined as output signal at connector X304.1. This signal is connected with the vSECC.single Board's digital output DIO10 through the Samtec connector:

Parameter	Min.	Typ.	Max.	Unit
Output voltage (High)		V_{in} - 0.5	V_{in}	V
Switching (Delay time)				
> High to Low	39	94	235	μs
> Low to High	39	94	235	μs
Rated current per channel		2		A
Each output is short-circuit proof				

12.3 Temperature Inputs

Two temperature sensor inputs, optimized for usage with PT-1000 temperature sensors, supervised by safety output:

Parameter	Min.	Typ.	Max.	Unit
Measurement current	970	1000	1030	μA
Measurement range	815		1600	Ω
Resolution	12			bit

12.4 Safety Output

Two outputs are available for Safety Output: OUT_SAF_HS (Output Safety High Side; X303.1) and OUT_SAF_LS (Output Safety Low Side; X303.2). The supervision of CP, PP and temperature sensors is always active. The output is **OFF** when energy transfer is not allowed (CP State in states A/B/F/E and/or temperature inputs not in range). The output is **ON** when energy transfer is allowed (CP state in states C/D and temperature inputs in range).

Parameters of OUT_SAF_HS (Output Safety High Side):

Parameter	Min.	Typ.	Max.	Unit
Output voltage (High)		V_{in} - 0.5	V_{in}	V
Switching (Delay time)				
> High to Low	39	94	235	µs
> Low to High	39	94	235	µs
Rated current per channel		2		A
Each output is short-circuit proof				

Parameters of OUT_SAF_LS (Output Safety Low Side):

Parameter	Min.	Typ.	Max.	Unit
Output voltage (High)			0.35	V
Switching (Delay time)				
> High to Low	12	38	76	µs
> Low to High	20	65	130	µs
Rated current per channel		2		A
Each output is short-circuit proof				

Caution: In order to deactivate supervision of PP (e.g. when using CCS Type 2 Connector), connect PP_PU (X302.3) with PP (X302.2) and add a Resistor of $142\,\Omega$ between PP (X302.2) and PE (X302.4).

Parameter	Min.	Typ.	Max.	Unit
Delay time (CP loss, PP loss)		3.6	5	ms
Upper treshold temperature input (resistance)	1333		1359	Ω
> Equivalent temperature	86.3		93.1	°C
Lower threshold temperature input (resistance)	802		816	Ω
> Equivalent temperature	-50.3		-46.7	°C

12.5 Serial Communication

12.5.1 CAN1 (Power Electronics)

CAN port with fixed bus termination:

Parameter	Min.	Typ.	Max.	Unit
Baudrate		500		kbit/s

12.5.2 RS485 (Modbus TCP to RTU gateway)

Configuration parameter:

Parameter	Min.	Typ.	Max.	Unit
Baudrate		9600		kbit/s
Number of data bits		8		
Number of stop bits		1		
Parity mode		even		

12.6 CCS Connector Control Pilot

One control pilot pin (designed according to IEC 61851):

Parameter	Min.	Typ.	Max.	Unit
Output voltage				
> On state		12		V
> Off state		-12		V
Frequency	0.98	1	1.02	kHz
Duty cycle accuracy	+/- 5			μs

12.7 Real Time Clock (RTC)

The backup battery pin connects to an external super-capacitor. It powers the real time clock during power loss. This ensures that the real time clock remains functional and synchronized after boot up:

Parameter	Min.	Typ.	Max.	Unit
Super-Capacitor voltage			3.3	V
RTC current (during power loss)		0.25		μA
Battery Backup time			240	h

13 Technical Data vSECC.single +70°C

In this chapter you will find the following information:

13.1 General

Parameter	Min.	Typ.	Max.	Unit
Supply voltage V_{in}	11	12	13	V
Power consumption at $12\,V$ > idle, communication to back end and power electronics active		5		W
Operating temperature	-40		70	°C
Storage temperature	-40		70	°C
Humidity	non-condensing			
Altitude (above sea level)	0 - 2000			m
PCB Dimensions (further details in mechanical drawing)	114.50 x 99.00			mm
Total weight	approx. 366			g

13.2 Digital IO's

13.2.1 SAF_FB (Safety Feedback)

The SAF_FB (Safety Feedback) is defined as input signal at connector X301.1. This signal is connected with the vSECC.single +70°C Board's digital input DIO5 through the Samtec connector:

Parameter	Min.	Typ.	Max.	Unit
Input voltage		3.3	V_{in}	V
Switching threshold		1.57		V
Current draw		3.3		mA

Caution: The digital input (SAF_FB) is populated with a pull up resistor to 3.3V. When the connector X301.1 is not connected, the voltage of the digital input is pulled up to the level of 3.3V. In this case the digital input DIO5 shows a high value. When the connector X301.1 is connected to GND, the digital input DIO5 goes to a low value.

13.2.2 DIN1, DIN2 (Digital Input 1 & 2), DIN_FB (Digital Input Feedback)

DIN1 (Digital Input 1), DIN2 (Digital Input 2) and DIN_FB (Digital Input Feedback) are defined as input signals. They are connected with the vSECC.single +70°C Board's DIOs through the Samtec connector as follows:

> DIN1 (X301.2) is connected to digital input DIO6
> DIN2 (X301.3) is connected to digital input DIO7
> DIN_FB (X301.4) is connected to digital input DIO9

Parameter	Min.	Typ.	Max.	Unit
Input voltage		5	V_{in}	V
Switching threshold		1.57		V
Current draw		0.51		mA

Caution: The digital inputs (DIN1, DIN2 and DIN_FB) are populated with a pull up resistor to 5V. When the connector is not connected, the voltage of each digital input is pulled up to the level of 5V. In this case the digital input shows a high value. When the connector is connected to GND, the digital input goes to a low value.

13.2.3 HB1_OUT (Half Bridge Output 1), HB2_OUT (Half Bridge Output 2)

HB1_OUT (Half Bridge Output 1) and HB2_OUT (Half Bridge Output 2) are defined as output signals. They are connected with the vSECC.single +70°C Board's DIOs through the Samtec connector as follows:

> HB1_OUT (X303.3) is configured via digital output DIO1
> HB2_OUT (X303.4) is configured via digital output DIO4

Parameter	Min.	Typ.	Max.	Unit
Output voltage (High)		V_{in} - 0.75	V_{in}	V
Switching (Delay time)				
> High to Low		1		µs
> Low to High		1		µs
Rated current per channel		1		A
Each output is short-circuit proof				

13.2.4 HS_OUT (High Side Switch Output)

HS_OUT (High Side Switch Output) is defined as output signal at connector X304.1. This signal is connected with the vSECC.single +70°C Board's digital output DIO10 through the Samtec connector:

Parameter	Min.	Typ.	Max.	Unit
Output voltage (High)		V_{in} - 0.5	V_{in}	V
Switching (Delay time)				
> High to Low	39	94	235	μs
> Low to High	39	94	235	μs
Rated current per channel		2		A
Each output is short-circuit proof				

13.3 Temperature Inputs

Two temperature sensor inputs, optimized for usage with PT-1000 temperature sensors, supervised by safety output:

Parameter	Min.	Typ.	Max.	Unit
Measurement current	970	1000	1030	μA
Measurement range	815		1600	Ω
Resolution	12			bit

13.4 Safety Output

Two outputs are available for Safety Output: OUT_SAF_HS (Output Safety High Side; X303.1) and OUT_SAF_LS (Output Safety Low Side; X303.2). The supervision of CP, PP and temperature sensors is always active. The output is **OFF** when energy transfer is not allowed (CP State in states A/B/F/E and/or temperature inputs not in range). The output is **ON** when energy transfer is allowed (CP state in states C/D and temperature inputs in range).

Parameters of OUT_SAF_HS (Output Safety High Side):

Parameter	Min.	Typ.	Max.	Unit
Output voltage (High)		V_{in} - 0.5	V_{in}	V
Switching (Delay time)				
> High to Low	39	94	235	µs
> Low to High	39	94	235	µs
Rated current per channel		2		A
Each output is short-circuit proof				

Parameters of OUT_SAF_LS (Output Safety Low Side):

Parameter	Min.	Typ.	Max.	Unit
Output voltage (High)			0.35	V
Switching (Delay time)				
> High to Low	12	38	76	µs
> Low to High	20	65	130	µs
Rated current per channel		2		A
Each output is short-circuit proof				

Caution: In order to deactivate supervision of PP (e.g. when using CCS Type 2 Connector), connect PP_PU (X302.3) with PP (X302.2) and add a Resistor of $142\,\Omega$ between PP (X302.2) and PE (X302.4).

Parameter	Min.	Typ.	Max.	Unit
Delay time (CP loss, PP loss)		3.6	5	ms
Upper treshold temperature input (resistance)	1333		1359	Ω
> Equivalent temperature	86.3		93.1	°C
Lower threshold temperature input (resistance)	802		816	Ω
> Equivalent temperature	-50.3		-46.7	°C

13.5 Serial Communication

13.5.1 CAN1 (Power Electronics)

CAN port with fixed bus termination:

Parameter	Min.	Typ.	Max.	Unit
Baudrate		500		kbit/s

13.5.2 RS485 (Modbus TCP to RTU gateway)

Configuration parameter:

Parameter	Min.	Typ.	Max.	Unit
Baudrate		9600		kbit/s
Number of data bits		8		
Number of stop bits		1		
Parity mode		even		

13.6 CCS Connector Control Pilot

One control pilot pin (designed according to IEC 61851):

Parameter	Min.	Typ.	Max.	Unit
Output voltage				
> On state		12		V
> Off state		-12		V
Frequency	0.98	1	1.02	kHz
Duty cycle accuracy	+/- 5			μs

13.7 Real Time Clock (RTC)

The backup battery pin connects to an external super-capacitor. It powers the real time clock during power loss. This ensures that the real time clock remains functional and synchronized after boot up:

Parameter	Min.	Typ.	Max.	Unit
Super-Capacitor voltage			3.3	V
RTC current (during power loss)		0.25		μA
Battery Backup time			240	h

A Conformity Declarations

A.1 vSECC

VECTOR

EC Declaration of Conformity

according to directive 2014/30/EU (EMC)
according to directive 2011/65/EU (RoHS)
according to directive 2012/19/EU (WEEE)

CE

The manufacturer

Vector Informatik GmbH

Ingersheimer Straße 24

70499 Stuttgart

herewith declares that the following product
vSECC Supply Equipment Communication Controller (Art. No. 20006)

complies with the essential requirements of the above directives, when used for its intended purpose. This declaration also comprises the delegated directive (EU) 2015/863.
The sole responsibility for issuing this Declaration of Conformity is with Vector.

Following harmonized standards have been applied:

EN 61000-6-2:2005 + AC: 2005	Generic standards – Immunity standard for industrial environments
EN 61000-6-3:2007 + A1:2011 + AC:2012	Generic standards - Emission standard for residential, commercial and light-industrial environments
EN 61000-3-2:2014	Limits - Limits for harmonic current emissions (equipment input current <= 16 A per phase)
EN 61000-3-3:2013	Limits - Limitation of voltage changes, voltage fluctuations and flicker in public low voltage supply systems, for equipment with rated current <16 A per phase and not subject to conditional connection
EN IEC 63000:2018	Technical documentation for the assessment of electrical and electronic products with respect to the restriction of hazardous substances

Place: Stuttgart

Date: 2020-09-30

Sign. Thomas Beck

Managing Director Dr. Thomas Beck

Vector Informatik GmbH
Ingersheimer Str. 24
70499 Stuttgart · Deutschland
Tel.: +49 711 80670-0
Fax: +49 711 80670-111
www.vector.com

BW Bank Stuttgart
IBAN: DE20 6005 0101 0002 2245 85 BIC: SOLADEST600
Deutsche Bank Stuttgart
IBAN: DE87 6007 0070 0161 4080 00 BIC: DEUTDESSXXX
Handelsregister Stuttgart HRB 17317

Managing Directors:
Dr. Thomas Beck
Dr. Stefan Krauß
Thomas Riegraf

VECTOR

UKCA Declaration of Conformity

UK CA

In accordance with UK Government Guidance
The Electromagnetic Compatibility Regulations 2016: 2016 No 1091 (EMC)
The Restriction of the Use of Hazardous Substances
in Electrical and Electronic Equipment Regulations 2012: 2012 No 3032 (RoHS)

The manufacturer

Vector Informatik GmbH

Ingersheimer Straße 24

70499 Stuttgart, Germany

herewith declares that the following product
vSECC Supply Equipment Communication Controller (Art. No. 20006)

complies with the essential requirements of the above-mentioned relevant UK Statutory Instruments and their amendments, when used for its intended purpose.
The sole responsibility for issuing this Declaration of Conformity is with Vector.

The following standards have been applied:

EN 61000-6-2:2005 / AC:2005	Generic standards – Immunity standard for industrial environments
EN 61000-6-3:2007 / A1:2011 / AC:2012	Generic standards – Emission standard for residential, commercial and light-industrial environments
EN 61000-3-2:2014	Limits - Limits for harmonic current emissions (equipment input current <= 16 A per phase)
EN 61000-3-3:2013	Limits - Limitation of voltage changes, voltage fluctuations and flicker in public low voltage supply systems, for equipment with rated current <16 A per phase and not subject to conditional connection
EN IEC 63000:2018	Technical documentation for the assessment of electrical and electronic products with respect to the restriction of hazardous substances

Place: Stuttgart

Date: 2023-06-16

Sign. Thomas Beck

Managing Director Dr. Thomas Beck

Vector Informatik GmbH
Ingersheimer Str. 24
70499 Stuttgart · Deutschland
Tel.: +49 711 80670-0
Fax: +49 711 80670-111
www.vector.com

BW Bank Stuttgart
IBAN: DE20 6005 0101 0002 2245 85 BIC: SOLADEST600
Deutsche Bank Stuttgart
IBAN: DE87 6007 0070 0161 4080 00 BIC: DEUTDESSXXX
Handelsregister Stuttgart HRB 17317

Managing Directors:
Dr. Thomas Beck
Dr. Stefan Krauß
Thomas Riegraf

TCB

GRANT OF EQUIPMENT AUTHORIZATION

Certification

Issued Under the Authority of the

Federal Communications Commission

By:

TCB

ACB, Inc.
6731 Whittier Avenue Suite C110
McLean, VA 22101

Date of Grant: 02/16/2021

Application Dated: 02/11/2021

Vector Informatik GmbH
Ingersheimer Str. 24
Stuttgart, 70499
Germany

Attention: Thomas Beck

NOT TRANSFERABLE

EQUIPMENT AUTHORIZATION is hereby issued to the named GRANTEE, and is VALID ONLY for the equipment identified hereon for use under the Commission's Rules and Regulations listed below.

FCC IDENTIFIER: 2AXYRVSECC
Name of Grantee: Vector Informatik GmbH
Equipment Class: Part 15 Class B Digital Device
Notes: Communication Controller for Charging Stations

Grant Notes	FCC Rule Parts	Frequency Range (MHZ)	Output Watts	Frequency Tolerance	Emission Designator
	15B	-			

Certificate of Compliance

Certificate: 80136659 **Master Contract:** 302522

Project: 80136659 **Date Issued:** 2022-10-12

Issued To: **Vector Informatik GmbH**
Ingersheimer Str. 24
Stuttgart, Baden-Württemberg, 70499
Germany

Attention: Rebekka Jentzsch

The products listed below are eligible to bear the CSA Mark shown with adjacent indicators 'C' and 'US' for Canada and US or with adjacent indicator 'US' for US only or without either indicator for Canada only.

Issued by: Markus Hackl
Markus Hackl

C US

PRODUCTS
CLASS - C386266 - INFORMATION TECHNOLOGY EQUIPMENT (CSA 62368-1)
CLASS - C386296 - INFORMATION TECHNOLOGY EQUIPMENT (ANSI/UL 62368-1) - Certified to US Standards

Communication Controller for Charging Stations, model vSECC, build-in unit, Class III
Ratings:
18 – 30 Vdc / 5 A

DQD 507 Rev. 2019-04-30 Page 1

2C68-73D3-11B0-CA74

방송통신기자재등의 적합등록 필증

Registration of Broadcasting and Communication Equipments

상호 또는 성명 *Trade Name or Registrant*	주식회사 벡터코리아아이티
기자재명칭(제품명칭) *Equipment Name*	Supply Equipment Communication Controller
기본모델명 *Basic Model Number*	vSECC
파생모델명 *Series Model Number*	
등록번호 *Registration No.*	R-R-VeC-vSECC
제조자/제조(조립)국가 *Manufacturer/Country of Origin*	Vector Informatik GmbH / 독일
등록연월일 *Date of Registration*	2020-10-14
기타 *Others*	

위 기자재는 「전파법」 제58조의2 제3항에 따라 등록되었음을 증명합니다.

It is verified that foregoing equipment has been registered under the Clause 3, Article 58-2 of Radio Waves Act.

2020년(Year) 10월(Month) 14일(Day)

국립전파연구원장 국립전파연구원장인

Director General of National Radio Research Agency

※ 적합등록 방송통신기자재는 반드시 **"적합성평가표시"** 를 부착하여 유통하여야 합니다.
위반시 과태료 처분 및 등록이 취소될 수 있습니다.

भारतीय मानक ब्यूरो
(उपभोक्ता मामले, खाद्य एवं सार्वजनिक वितरण मंत्रालय, भारत सरकार)
BUREAU OF INDIAN STANDARDS
(Ministry of Consumer Affairs, Food & Public Distribution, Govt. of India)
मानक : पथप्रदर्शक :

मानक भवन, 9 बहादुर शाह जफ़र मार्ग, नई दिल्ली – 110002
Manak Bhavan, 9 Bahadur Shah Zafar Marg, New Delhi – 110002
दूरभाष/Phone: +91-11-23230856/2323010131/23233375/23239402
ई-मेल/E-mail: registration@bis.gov.in
वेबसाईट/Website: https://bis.gov.in/, https://www.crsbis.in/BIS/

Our Ref: Registration/CRS 2021-4099/R-41194808 Date:23-06-2021

Subject : Licence Document

MANUFACTURING UNIT :	Vector Informatik Gmbh INGERSHEIMER STR.24 D 70499 STUTTGART, GERMANY BADEN WURTTEMBERG,Germany-70499 Thomas.Riegraf@vector.com 4971180670108	

Dear Sir,

1.With reference to your Application, we are pleased to inform you that it has been decided to grant you licence as per details given below :

Product Category :	Automatic Data Processing Machine
Product Name :	Controller Appliance (ADPM)
IS NO :	IS 13252(PART 1):2010/ IEC 60950-1 : 2005
Brand (As Declared by Manufacturer) :	vector
Model :	[Brand -> vector, Models -> vSECC]
Factory Address :	INGERSHEIMER STR.24 D 70499 STUTTGART, GERMANY BADEN WURTTEMBERG,Germany-70499

2. The Licence is being granted for your unit located at the address and for the brand and models mentioned at serial no 1 above.

3. The number assigned to this Licence is **R-41194808** which has been made operative from **23-06-2021** and is valid upto **22-06-2023** . The Licence Number should invariably be referred to in your future correspondence.

4. The rights and privileges under the licence shall not be exercised by any other factory / organization at any other location. This licence is not transferable. In the event of shifting of the manufacturing machinery from the registered premises to some other place use of the Licence Number shall be stopped and BIS shall be informed.

5. The licensee shall comply with the provisions of the Act, rules and regulations framed thereunder and as amended from time to time.

6. The licensee shall follow the guidelines for the use of Standard Mark and labeling requirements as per Annex-I.

7. The licensee shall not use the licence in any manner which contravenes the provisions of Act, rules and regulations framed thereunder and as amended from time to time.

8. Upon expiry of validity, stoppage or suspension or cancellation of licence, you shall discontinue forthwith the self declaration of conformity to the relevant Indian Standard(s) and withdraw all promotional and advertising matter which contains any reference thereto.

9. As per your declaration, **Shripad Kanni, Local Productline Manager, VECTOR INFORMATIK INDIA PRIVATE LIMITED(Address- 5th Floor, Office No. 11 to 14, Tara Height,, Old Pune Mumbai Road, Wakdewadi, Pune, Maharashtra, 411003,NA)** is your authorized Indian representative. Any intended change in the name of the Indian representative ought to be brought to our notice immediately along with requisite fees and document.

10. For renewal of licence, the licensee shall have to apply to BIS three months in advance before expiration of the licence and application form for renewal is available on BIS website

11. The licence is not transferable. Kindly acknowledge receipt of this letter.

Thanking you,

Yours faithfully,
(Peeyush Prakash)
Sc. C
Telfax : +91-11-23230856
E-mail: registration@bis.gov.in

Note: This is a system generated letter. Hence signature is not required.
To verify authentication of letter, kindly scan the QR code on this letter.

A.2 vSECC.single

VECTOR

EU Declaration of Conformity

according to directive 2014/30/EU (EMC)
according to directive 2011/65/EU (RoHS)
according to directive 2012/19/EU (WEEE)

CE

The manufacturer

Vector Informatik GmbH

Ingersheimer Straße 24

70499 Stuttgart

herewith declares that the following product
vSECC.single Supply Equipment Communication Controller (Art. No. 20011)

complies with the essential requirements of the above directives, when used for its intended purpose. This declaration also comprises the delegated directive (EU) 2015/863.
The sole responsibility for issuing this Declaration of Conformity is with Vector.

Following harmonized standards have been applied:

EN 61000-6-2:2005 + AC: 2005	Generic standards – Immunity standard for industrial environments
EN 61000-6-3:2007 + A1:2011 + AC:2012	Generic standards - Emission standard for residential, commercial and light-industrial environments
EN 61000-3-2:2014	Limits - Limits for harmonic current emissions (equipment input current <= 16 A per phase)
EN 61000-3-3:2013	Limits - Limitation of voltage changes, voltage fluctuations and flicker in public low voltage supply systems, for equipment with rated current <16 A per phase and not subject to conditional connection
EN IEC 63000:2018	Technical documentation for the assessment of electrical and electronic products with respect to the restriction of hazardous substances

Place: Stuttgart

Date: 2023-05-05

Sign. Thomas Beck

Managing Director Dr. Thomas Beck

Vector Informatik GmbH
Ingersheimer Str. 24
70499 Stuttgart · Deutschland
Tel.: +49 711 80670-0
Fax: +49 711 80670-111
www.vector.com

BW Bank Stuttgart
IBAN: DE20 6005 0101 0002 2245 85 BIC: SOLADEST600
Deutsche Bank Stuttgart
IBAN: DE87 6007 0070 0161 4080 00 BIC: DEUTDESSXXX
Handelsregister Stuttgart HRB 17317

Managing Directors:
Dr. Thomas Beck
Dr. Stefan Krauß
Thomas Riegraf

VECTOR

UKCA Declaration of Conformity

UK CA

In accordance with UK Government Guidance
The Electromagnetic Compatibility Regulations 2016: 2016 No 1091 (EMC)
The Restriction of the Use of Hazardous Substances
in Electrical and Electronic Equipment Regulations 2012: 2012 No 3032 (RoHS)

The manufacturer

Vector Informatik GmbH

Ingersheimer Straße 24

70499 Stuttgart, Germany

herewith declares that the following product
vSECC.single Supply Equipment Communication Controller (Art. No. 20011)

complies with the essential requirements of the above-mentioned relevant UK Statutory Instruments and their amendments, when used for its intended purpose.
The sole responsibility for issuing this Declaration of Conformity is with Vector.

The following standards have been applied:

EN 61000-6-2:2005 / AC:2005	Generic standards – Immunity standard for industrial environments
EN 61000-6-3:2007 / A1:2011 / AC:2012	Generic standards – Emission standard for residential, commercial and light-industrial environments
EN 61000-3-2:2014	Limits - Limits for harmonic current emissions (equipment input current <= 16 A per phase)
EN 61000-3-3:2013	Limits - Limitation of voltage changes, voltage fluctuations and flicker in public low voltage supply systems, for equipment with rated current <16 A per phase and not subject to conditional connection
EN IEC 63000:2018	Technical documentation for the assessment of electrical and electronic products with respect to the restriction of hazardous substances

Place: Stuttgart

Date: 2023-05-05

Sign. Thomas Beck

Managing Director Dr. Thomas Beck

Vector Informatik GmbH
Ingersheimer Str. 24
70499 Stuttgart · Deutschland
Tel.: +49 711 80670-0
Fax: +49 711 80670-111
www.vector.com

BW Bank Stuttgart
IBAN: DE20 6005 0101 0002 2245 85 BIC: SOLADEST600
Deutsche Bank Stuttgart
IBAN: DE87 6007 0070 0161 4080 00 BIC: DEUTDESSXXX
Handelsregister Stuttgart HRB 17317

Managing Directors:
Dr. Thomas Beck
Dr. Stefan Krauß
Thomas Riegraf

VECTOR

FCC-Self-Declaration of Conformity

Supplier's Declaration of Conformity

47 CFR § 2.1077 Compliance Information

Unique Identifier: Art. No. 20011 - vSECC.single

Vector North America Inc.

39500 Orchard Hill Place, Suite 500
Novi, Michigan
48375
Phone: +1 248-449-9290; email: sales@us.vector.com

FCC Compliance Statement (products subject to Part 15)

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:
(1) This device may not cause harmful interference, and
(2) this device must accept any interference received, including interference that may cause undesired operation.

Place: Stuttgart
Date: 2023-05-05

Sign. Thomas Beck
Managing Director Dr. Thomas Beck

Place: Novi, Michigan
Date: 2023-05-05

Sign. Tony Mascolo
Local Managing Director Tony Mascolo

Vector Informatik GmbH
Ingersheimer Str. 24
70499 Stuttgart · Deutschland
Tel.: +49 711 80670-0
Fax: +49 711 80670-111
www.vector.com

BW Bank Stuttgart
IBAN: DE20 6005 0101 0002 2245 85 BIC: SOLADEST600
Deutsche Bank Stuttgart
IBAN: DE87 6007 0070 0161 4080 00 BIC: DEUTDESSXXX
Handelsregister Stuttgart HRB 17317

Managing Directors:
Dr. Thomas Beck
Dr. Stefan Krauß
Thomas Riegraf

Certificate of Compliance

Certificate:	80149809	**Master Contract:**	302522
Project:	80149809	**Date Issued:**	2023-03-21

Issued To: **Vector Informatik GmbH**
Ingersheimer Str. 24
Stuttgart, Baden-Württemberg, 70499
Germany

Attention: Anna Marilena Hesse

The products listed below are eligible to bear the CSA Mark shown with adjacent indicators 'C' and 'US' for Canada and US or with adjacent indicator 'US' for US only or without either indicator for Canada only.

Issued by: Markus Hackl
Markus Hackl

C US

PRODUCTS
CLASS - C386266 - INFORMATION TECHNOLOGY EQUIPMENT (CSA 62368-1)
CLASS - C386296 - INFORMATION TECHNOLOGY EQUIPMENT (ANSI/UL 62368-1) - Certified to US Standards

Communication controller, model vSECC.single, Class III equipment, built in equipment, Ratings 11 Vdc to 13 Vdc.

DQD 507 Rev. 2019-04-30 Page 1

6A81-3D74-1299-368C

방송통신기자재등의 적합등록 필증

Registration of Broadcasting and Communication Equipments

상호 또는 성명 Trade Name or Registrant	**주식회사 벡터코리아아이티**
기자재명칭(제품명칭) Equipment Name	**Supply Equipment Communication Controller**
기기부호/추가 기기부호 Equipment code /Additional Equipment code	**IMI61**
기본모델명 Basic Model Number	**vSECC.single**
파생모델명 Series Model Number	
등록번호 Registration No.	**R-R-VeC-vSECCsingle**
제조자/제조국가 Manufacturer/Country of Origin	**Vector Informatik GmbH/독일**
등록연월일 Date of Registration	**2023-05-14**
기타 Others	

위 기자재는 「전파법」 제58조의2 제3항에 따라 등록되었음을 증명합니다.

It is verified that foregoing equipment has been registered under the Clause 3, Article 58-2 of Radio Waves Act.

2023년(Year) 05월(Month) 15일(Day)

국립전파연구원장 국립전파연구원장인

Director General of National Radio Research Agency

※ 적합등록 방송통신기자재는 반드시 **"적합성평가표시"** 를 부착하여 유통하여야 합니다.
위반시 과태료 처분 및 등록이 취소될 수 있습니다.

भारतीय मानक ब्यूरो
(उपभोक्ता मामले, खाद्य एवं सार्वजनिक वितरण मंत्रालय, भारत सरकार)
BUREAU OF INDIAN STANDARDS
(Ministry of Consumer Affairs, Food & Public Distribution, Govt. of India)

मानक भवन, 9 बहादुर शाह जफ़र मार्ग, नई दिल्ली – 110002
Manak Bhavan, 9 Bahadur Shah Zafar Marg, New Delhi – 110002
दूरभाष / Phone: +91-11-23230856/2323010131/23233375/23239402
ई–मेल / E-mail: registration@bis.gov.in
वेबसाईट / Website: https://bis.gov.in/, https://www.crsbis.in/BIS/

Our Ref: REGISTRATION/CRS 2021-4099/R-41194808 Date:05-12-2023

Inclusion Id: 75523

Subject :Inclusion of Additional Model(s)

MANUFACTURING UNIT :	Vector Informatik Gmbh INGERSHEIMER STR.24 D 70499 STUTTGART, GERMANY BADEN WURTTEMBERG,Germany-70499 Thomas.Riegraf@vector.com 4971180670108	

Dear Sir,

1. This has reference to your request for inclusion of models of **"Automatic Data Processing Machine" as per IS 13252(Part 1):2010/ IEC 60950-1 : 2005 in Licence No. R-41194808** already granted to you which is valid upto 22-06-2025.

2. It is intimated that the additional Models as per details given below have been agreed to be included in your scope of Licence. **R-41194808 w.e.f. 05-12-2023:**

Product Category	Automatic Data Processing Machine
Product Name	Controller Appliance (ADPM)
IS No.	IS 13252(Part 1):2010/ IEC 60950-1 : 2005
Brand (As Declared by Manufacturer):	vector
Inclusion of Additional Models (**w.e.f. 05-12-2023**)	[Brand -> vector, Models -> vSECC.single]
Factory Address	INGERSHEIMER STR.24 D 70499 STUTTGART, GERMANY BADEN WURTTEMBERG,Germany-70499

3. Other terms and conditions of the licence shall remain same.

4. This letter is being issued with the approval of competent authority.

Kindly acknowledge receipt of this letter.

Thanking you,

Yours faithfully,
(Aurosmita Kabiraj)
Scientist D
Telfax : +91-11-23230856
E-mail: registration@bis.gov.in

Note: This is a system generated letter. Hence signature is not required.
To verify authentication of letter, kindly scan the QR code on this letter.

For details information on BIS, consult the e-BIS Portal (www.manakonline. in).
Please use BIS CARE APP for verification of ISI-marked goods and hallmarked gold jewellery.

A.3 vSECC.single +70°C

VECTOR

EU Declaration of Conformity

CE

Acc. directive 2014/30/EU (EMV)
Acc. directive 2011/65/EU (RoHS)

The manufacturer

Vector Informatik GmbH

Ingersheimer Straße 24

70499 Stuttgart

herewith declares that the following product
vSECC.single +70ºC Supply Equipment Communication Controller (Art. Nr. 20022)

complies with the essential requirements of the above directives, when used for its intended purpose. This declaration also comprises the delegated directive (EU) 2015/863.
The sole responsibility for issuing this Declaration of Conformity is with Vector.

Following harmonized standards have been applied:

EN 61000-6-2:2005 / AC:2005	Generic standards – Immunity standard for industrial environments
EN 61000-6-3:2007 / A1:2011 / AC:2012	Generic standards – Emission standard for residential, commercial and light-industrial environments
EN IEC 63000:2018	Technical documentation for the assessment of electrical and electronic products with respect to the restriction of hazardous substances

Place: Stuttgart

Date: 2025-05-23

Sign. Thomas Beck

Managing Director Dr. Thomas Beck

Vector Informatik GmbH
Ingersheimer Str. 24
70499 Stuttgart · Deutschland
Tel.: +49 711 80670-0
Fax: +49 711 80670-111
www.vector.com

BW Bank Stuttgart
IBAN: DE20 6005 0101 0002 2245 85 BIC: SOLADEST600
Deutsche Bank Stuttgart
IBAN: DE87 6007 0070 0161 4080 00 BIC: DEUTDESSXXX
Handelsregister Stuttgart HRB 17317

Managing Directors:
Dr. Thomas Beck
Dr. Matthias Traub

FCC-Self-Declaration of Conformity

Supplier's Declaration of Conformity

47 CFR § 2.1077 Compliance Information

Unique Identifier: Art. No. 20022 - vSECC.single +70°C

Vector North America Inc.

39500 Orchard Hill Place, Suite 500
Novi, Michigan
48375
Phone: +1 248-449-9290; email: sales@us.vector.com

FCC Compliance Statement (products subject to Part 15 Subpart B – Unintentional Radiators)

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:
(1) This device may not cause harmful interference, and
(2) this device must accept any interference received, including interference that may cause undesired operation.

Place: Stuttgart
Date: 2025-05-23

Sign. Thomas Beck
Managing Director Dr. Thomas Beck

Place: Novi, Michigan
Date: 2025-05-23

Sign. Tony Mascolo
Local Managing Director Tony Mascolo

Vector Informatik GmbH
Ingersheimer Str. 24
70499 Stuttgart · Deutschland
Tel.: +49 711 80670-0
Fax: +49 711 80670-111
www.vector.com

BW Bank Stuttgart
IBAN: DE20 6005 0101 0002 2245 85 BIC: SOLADEST600
Deutsche Bank Stuttgart
IBAN: DE87 6007 0070 0161 4080 00 BIC: DEUTDESSXXX
Handelsregister Stuttgart HRB 17317

Managing Directors:
Dr. Thomas Beck
Dr. Matthias Traub

Certificate of Compliance

Certificate:	80238631	**Master Contract:**	302522
Project:	80238631	**Date Issued:**	2025-05-19
Issued to:	**Vector Informatik GmbH** **Ingersheimer Str. 24** **Stuttgart, Baden-Württemberg** **70499** **Germany** **Attention:** Aymen Awadni	**Issued by:**	*Mustafa Emre Çarkacı* Mustafa Emre Çarkacı

The products listed below are eligible to bear the CSA Mark shown with adjacent indicators 'C' and 'US' for Canada and US or with adjacent indicator 'US' for US only or without either indicator for Canada only.

C US

PRODUCTS

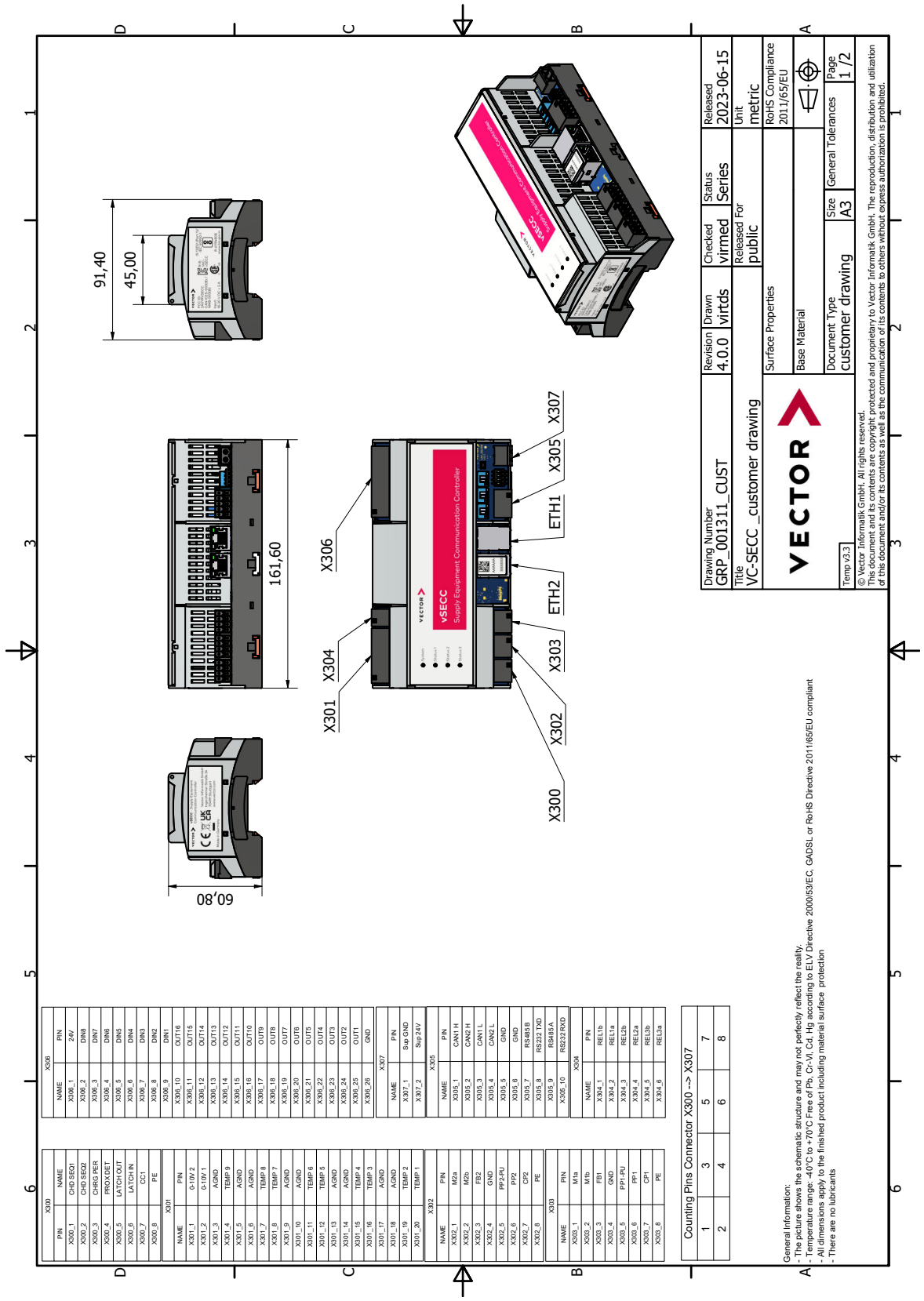
Class 3862 66 INFORMATION TECHNOLOGY EQUIPMENT - (CSA 62368-1)
Class 3862 96 INFORMATION TECHNOLOGY EQUIPMENT - (ANSI/UL 62368-1) - Certified to US Standards

Communication Controller, Class III Equipment

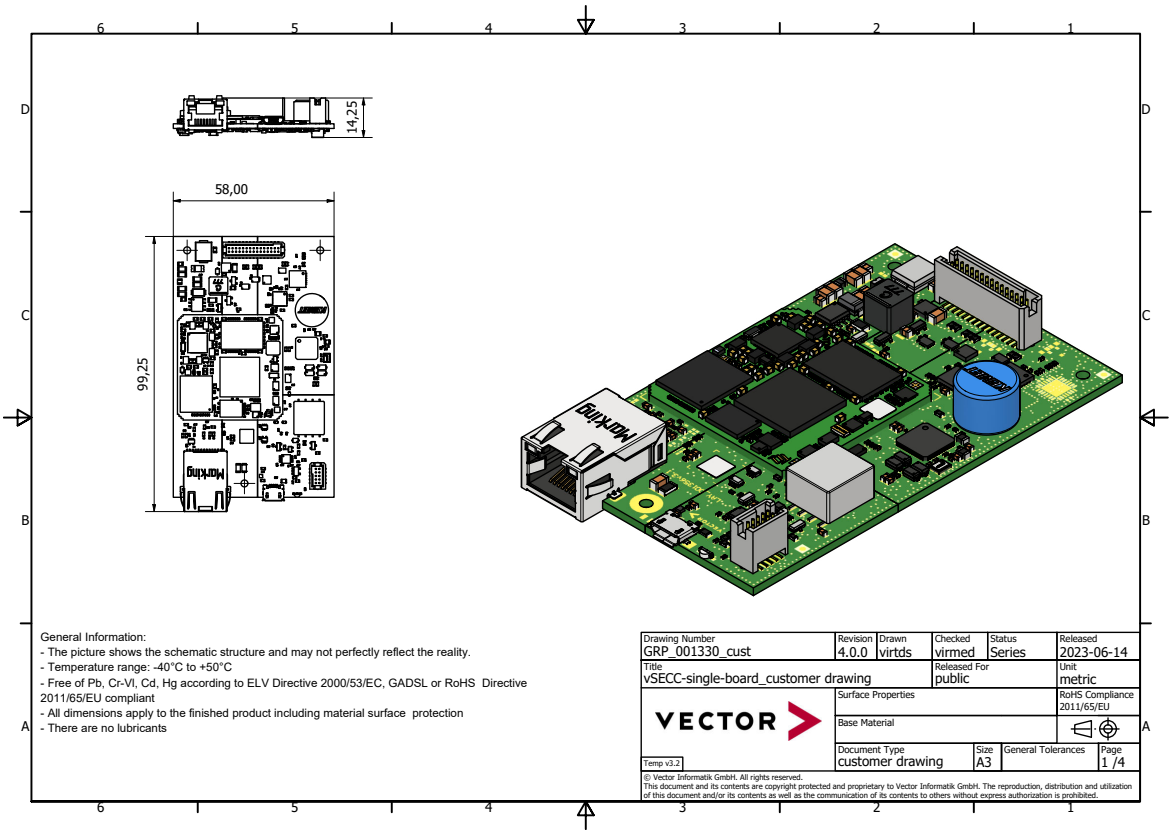
Model(s)	Rated Voltage (VDC)	Rated Current (Adc)	Operating Temp. (°C)
vSECC.single +70°C	11-13 V DC	max. 6 A	-40°C to +70°C

QD-1397 Rev 2019-04-30 Page 1

B vSECC Mechanical Drawing

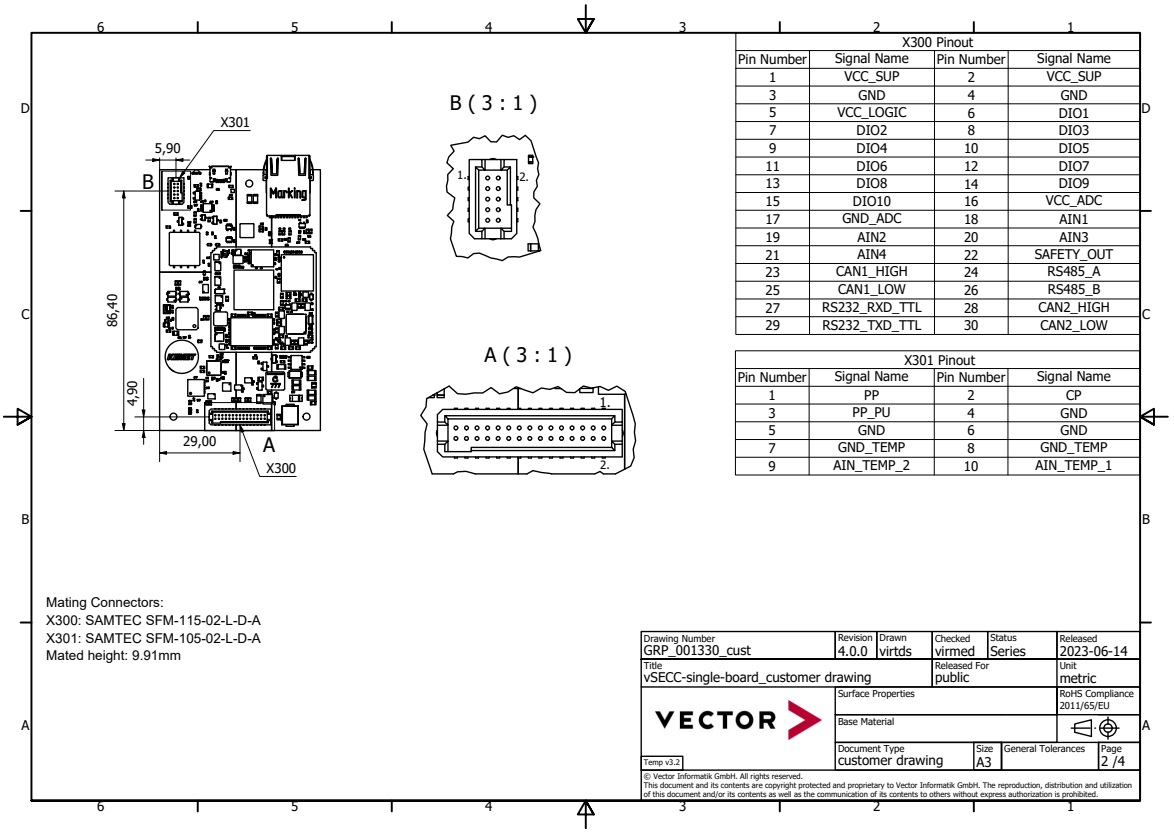


C vSECC.single Board Mechanical Drawing



General Information:
- The picture shows the schematic structure and may not perfectly reflect the reality.
- Temperature range: -40°C to +50°C
- Free of Pb, Cr-VI, Cd, Hg according to ELV Directive 2000/53/EC, GADSL or RoHS Directive 2011/65/EU compliant
- All dimensions apply to the finished product including material surface protection
- There are no lubricants

Drawing Number	Revision	Drawn	Checked	Status	Released
GRP_001330_cust	4.0.0	virtds	virmed	Series	2023-06-14
Title			Released For		Unit
vSECC-single-board_customer drawing			public		metric
VECTOR	Surface Properties				RoHS Compliance 2011/65/EU
	Base Material				
Temp v3.2	Document Type	Size	General Tolerances		Page
	customer drawing	A3			1 /4

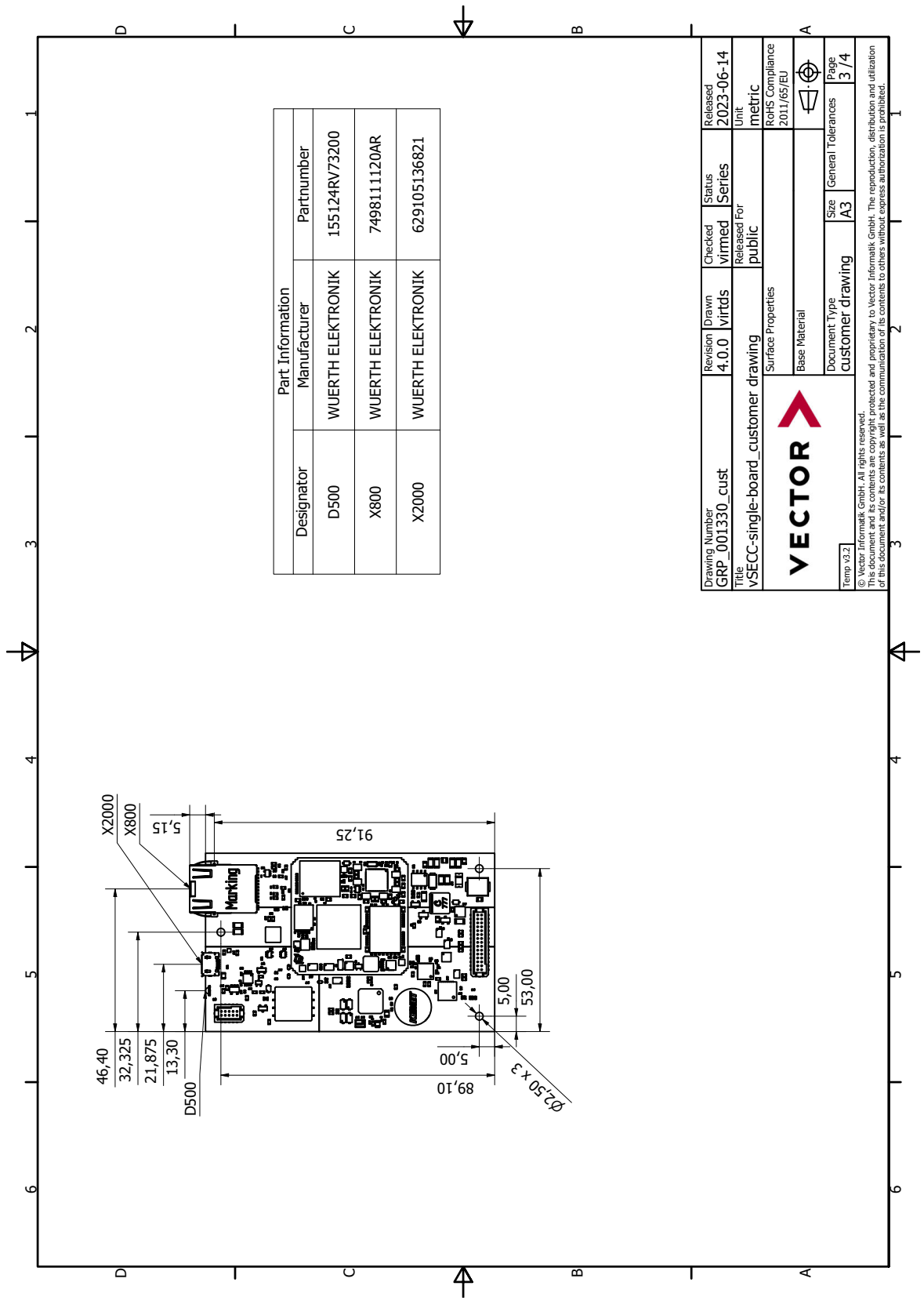


X300 Pinout			
Pin Number	Signal Name	Pin Number	Signal Name
1	VCC_SUP	2	VCC_SUP
3	GND	4	GND
5	VCC_LOGIC	6	DIO1
7	DIO2	8	DIO3
9	DIO4	10	DIO5
11	DIO6	12	DIO7
13	DIO8	14	DIO9
15	DIO10	16	VCC_ADC
17	GND_ADC	18	AIN1
19	AIN2	20	AIN3
21	AIN4	22	SAFETY_OUT
23	CAN1_HIGH	24	RS485_A
25	CAN1_LOW	26	RS485_B
27	RS232_RXD_TTL	28	CAN2_HIGH
29	RS232_TXD_TTL	30	CAN2_LOW

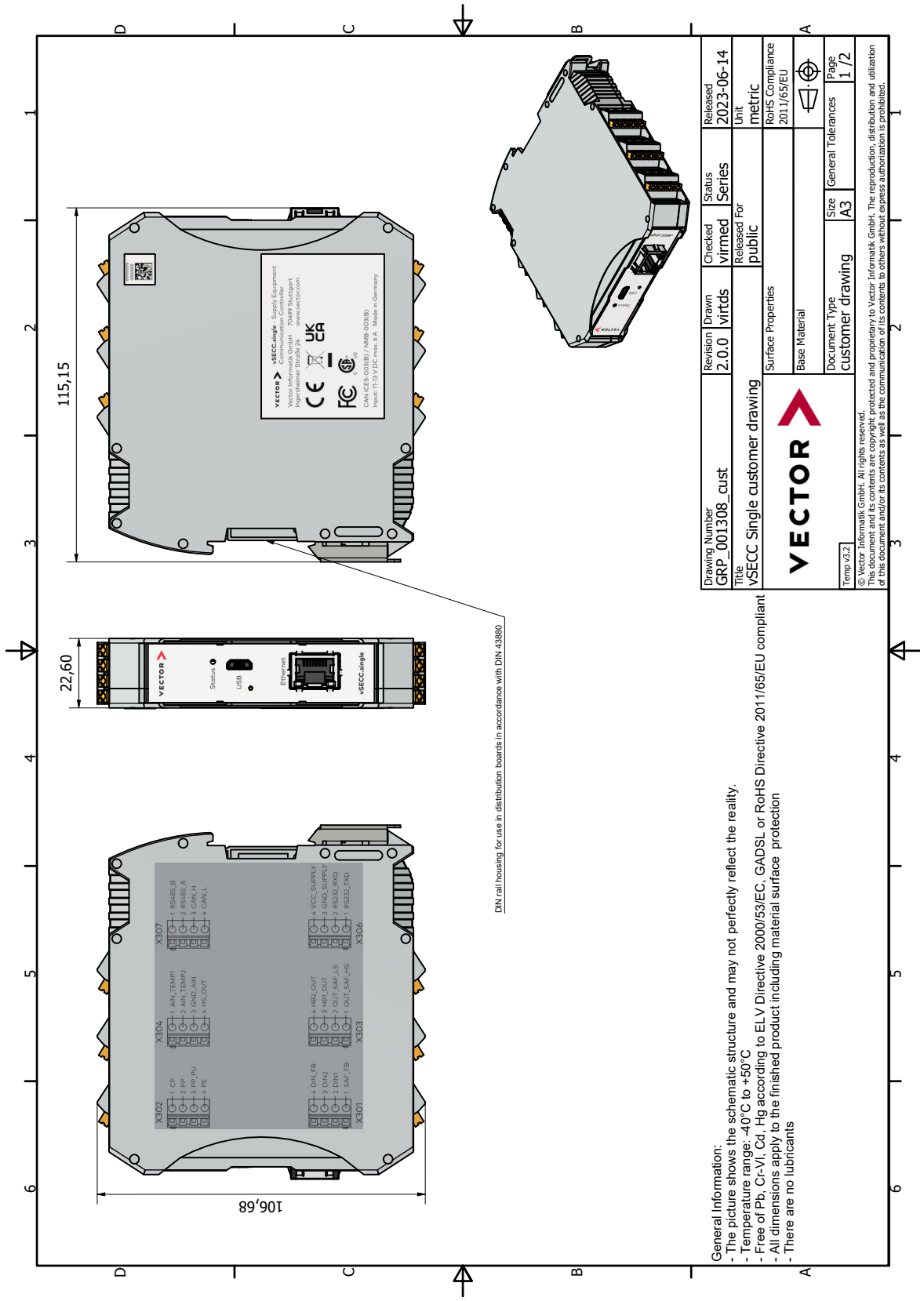
X301 Pinout			
Pin Number	Signal Name	Pin Number	Signal Name
1	PP	2	CP
3	PP_PU	4	GND
5	GND	6	GND
7	GND_TEMP	8	GND_TEMP
9	AIN_TEMP_2	10	AIN_TEMP_1

Mating Connectors:
X300: SAMTEC SFM-115-02-L-D-A
X301: SAMTEC SFM-105-02-L-D-A
Mated height: 9.91mm

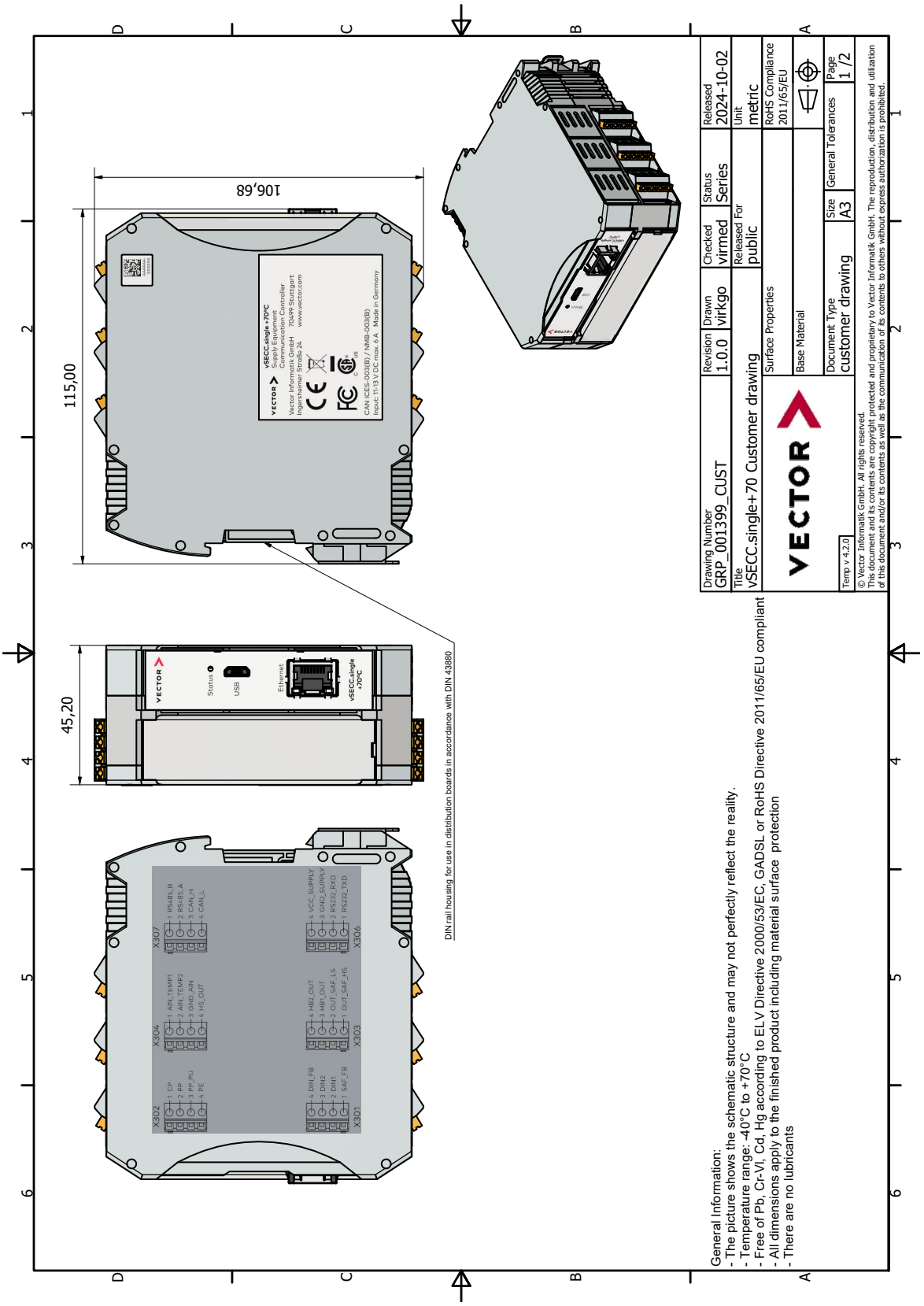
Drawing Number	Revision	Drawn	Checked	Status	Released
GRP_001330_cust	4.0.0	virtds	virmed	Series	2023-06-14
Title			Released For		Unit
vSECC-single-board_customer drawing			public		metric
VECTOR	Surface Properties				RoHS Compliance 2011/65/EU
	Base Material				
Temp v3.2	Document Type	Size	General Tolerances		Page
	customer drawing	A3			2 /4


Part Information
Designator
Manufacturer
Partnumber
D500
WUERTH ELEKTRONIK
155124RV73200
X800
WUERTH ELEKTRONIK
7498111120AR
X2000
WUERTH ELEKTRONIK
629105136821
Drawing Number
GRP_001330_cust
Revision
4.0.0
Drawn
virtds
Checked
virmed
Status
Series
Released
2023-06-14
Title
vSECC-single-board_customer drawing
Released For
public
Unit
metric
Surface Properties
RoHS Compliance
2011/65/EU
Base Material
Document Type
customer drawing
Size
A3
General Tolerances
Page
3 /4
Temp v3.2
VECTOR
© Vector Informatik GmbH. All rights reserved.
This document and its contents are copyright protected and proprietary to Vector Informatik GmbH. The reproduction, distribution and utilization of this document and/or its contents as well as the communication of its contents to others without express authorization is prohibited.
X2000
X800
5,15
91,25
Marking
46,40
32,325
21,875
13,30
D500
5,00
89,10
5,00
53,00
Ø2,50 x 3

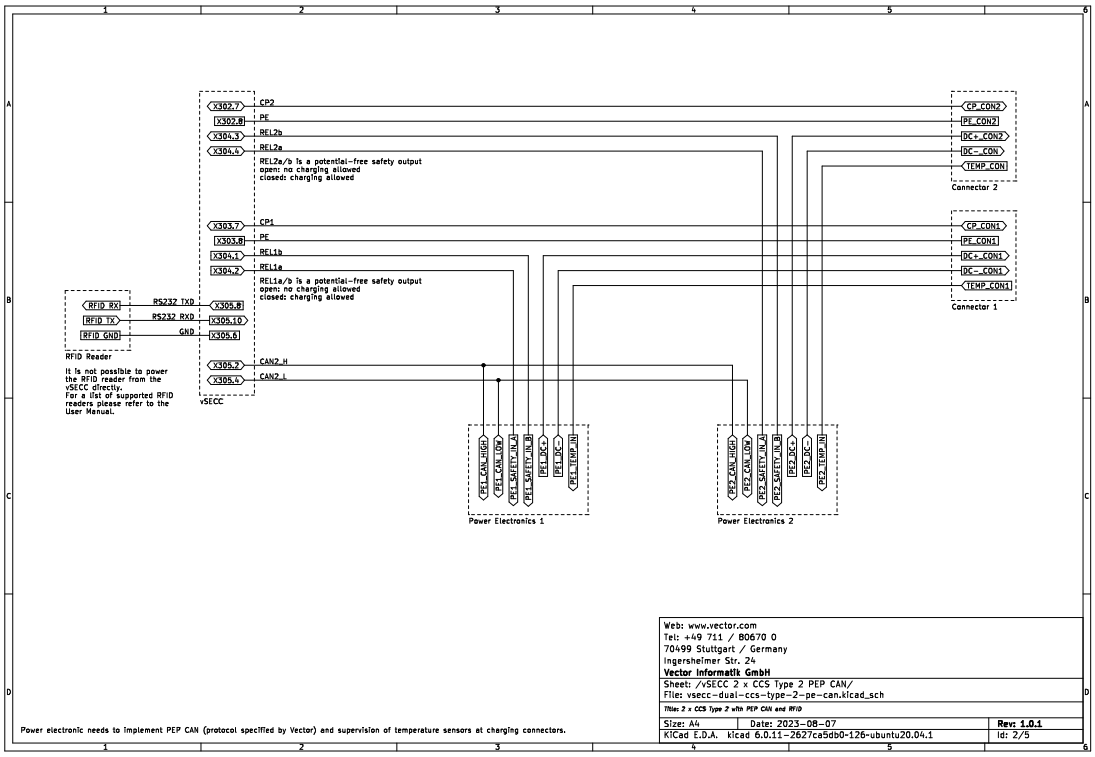
D vSECC.single Mechanical Drawing

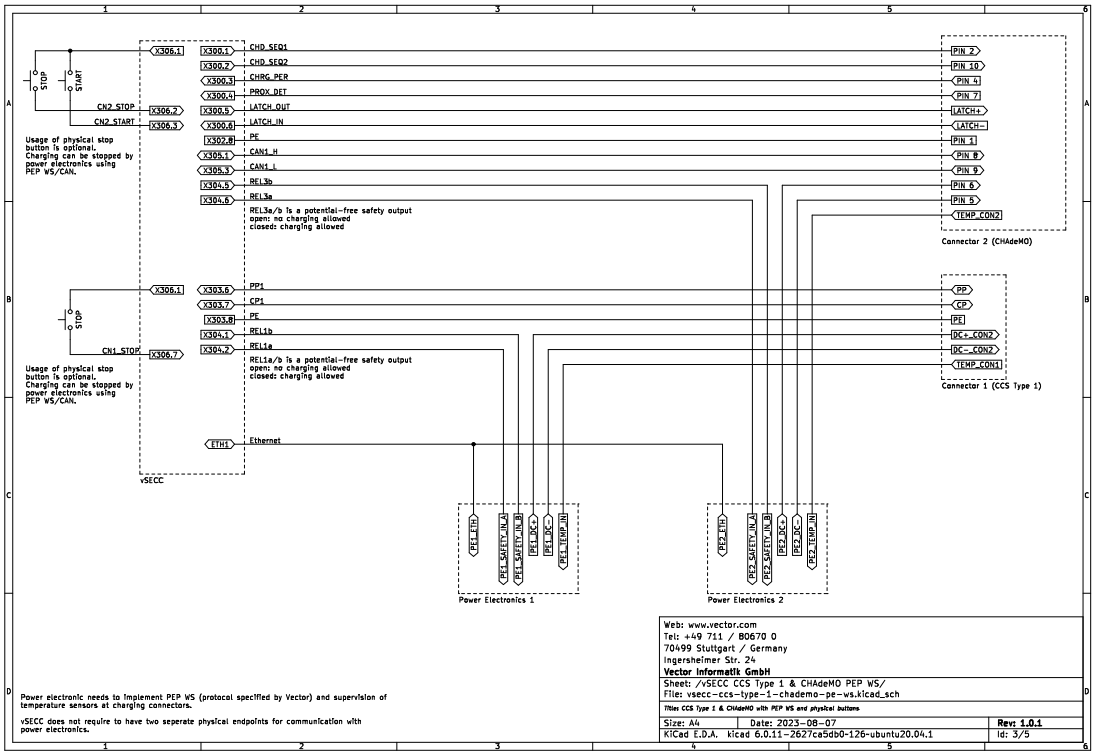


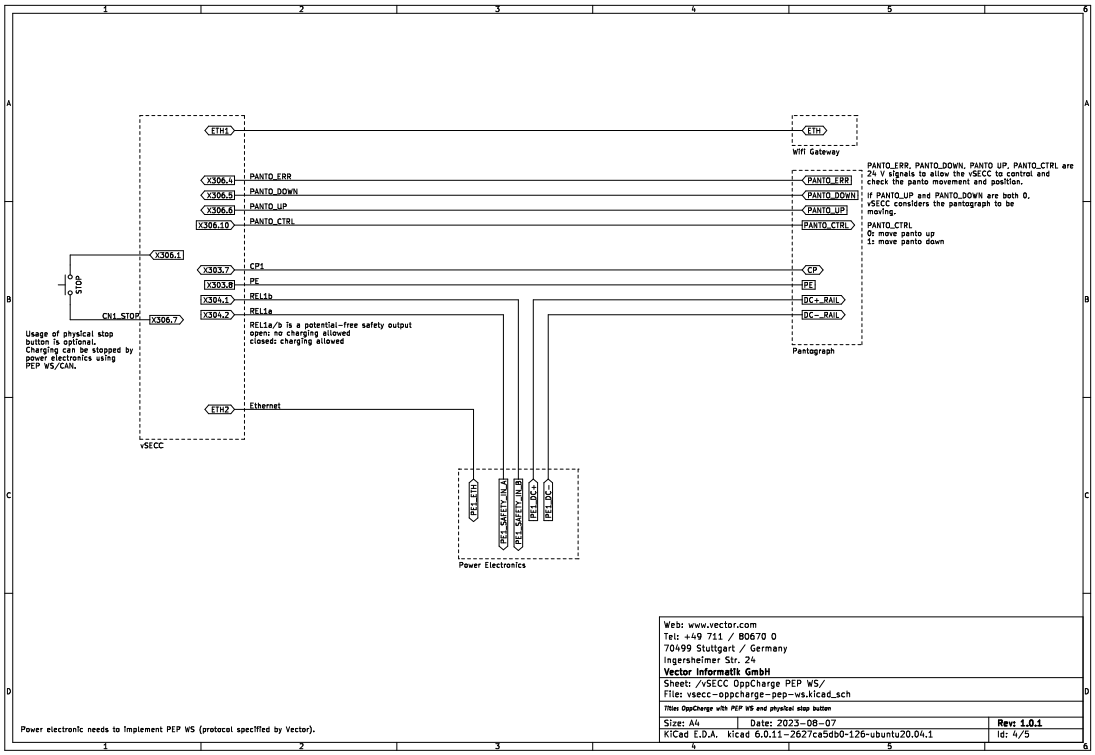
E vSECC.single +70°C Mechanical Drawing

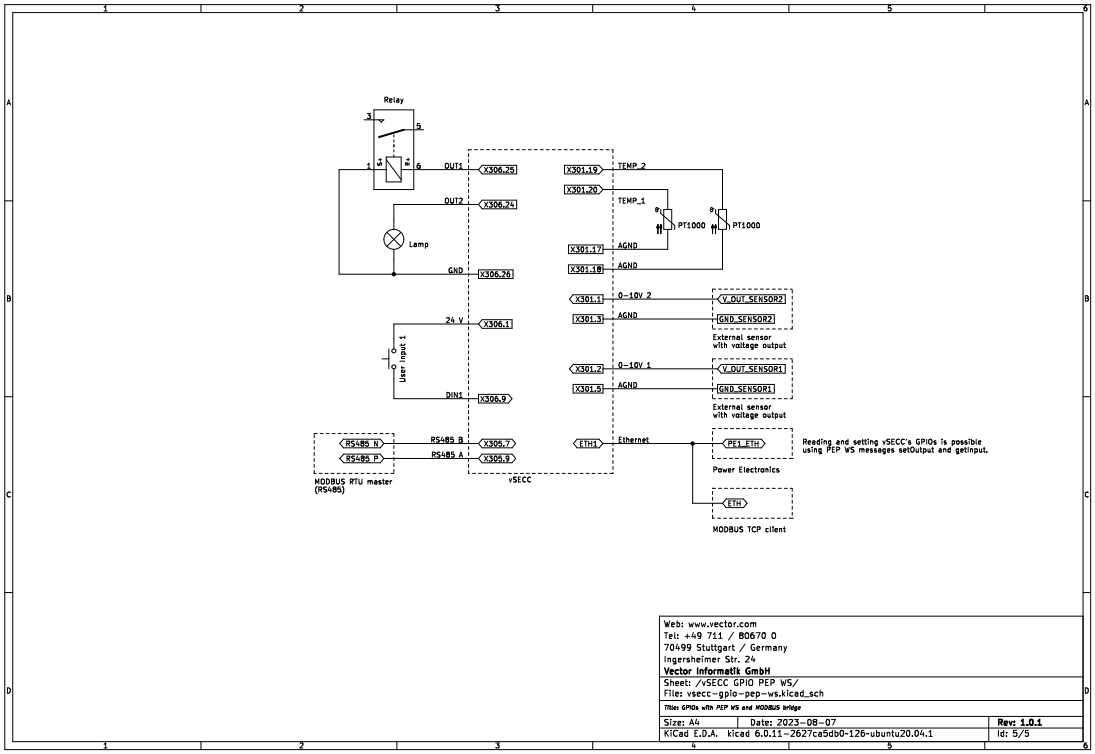


F vSECC Example Wiring Diagrams

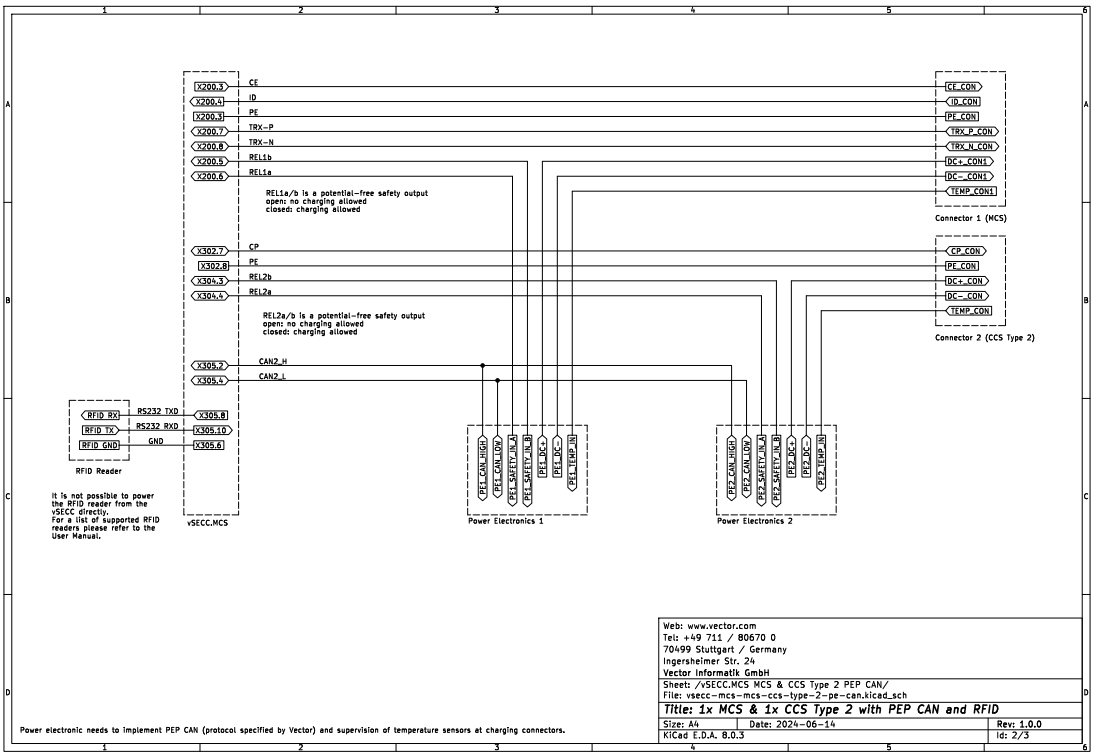


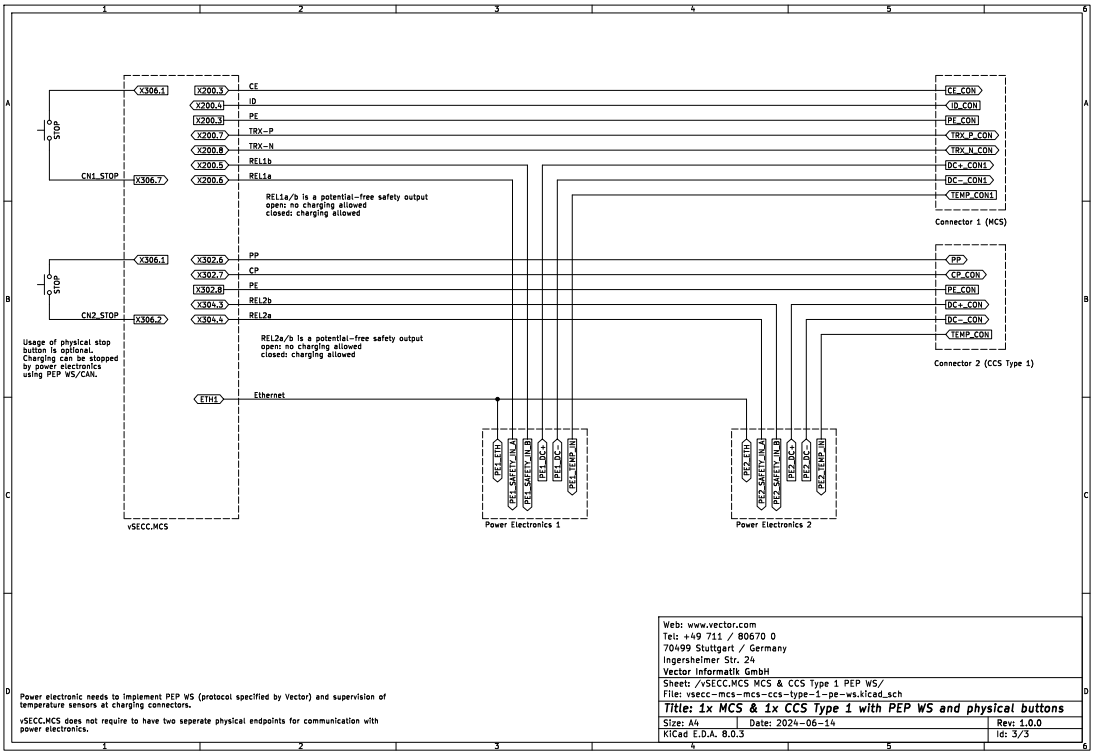



ETH1
ETH
WiFi Gateway
X306.4
PANTO_ERR
X306.5
PANTO_DOWN
X306.6
PANTO_UP
X306.10
PANTO_CTRL
PANTO_ERR
PANTO_DOWN
PANTO_UP
PANTO_CTRL
PANTO_ERR, PANTO_DOWN, PANTO_UP, PANTO_CTRL are 24 V signals to allow the vSECC to control and check the panto movement and position.
If PANTO_UP and PANTO_DOWN are both 0, vSECC considers the pantograph to be moving.
PANTO_CTRL
0: move panto up
1: move panto down
X306.1
STOP
CN1_STOP
X306.7
X303.7
CP1
X303.8
PE
X304.1
REL1b
X304.2
REL1a
CP
PE
DC+_RAIL
DC-_RAIL
Pantograph
Usage of physical stop button is optional.
Charging can be stopped by power electronics using PEP WS/CAN.
REL1a/b is a potential-free safety output
open: no charging allowed
closed: charging allowed
ETH2
Ethernet
vSECC
PE1_ETH
PE1_SAFETY_IN_A
PE1_SAFETY_IN_B
PE1_DC-
PE1_DC+
Power Electronics
Power electronic needs to implement PEP WS (protocol specified by Vector).
Web: www.vector.com
Tel: +49 711 / 80670 0
70499 Stuttgart / Germany
Ingersheimer Str. 24
Vector Informatik GmbH
Sheet: /vSECC OppCharge PEP WS/
File: vsecc-oppcharge-pep-ws.kicad_sch
Titles OppCharge with PEP WS and physical stop button
Size: A4
Date: 2023-08-07
Rev: 1.0.1
KiCad E.D.A. kicad 6.0.11-2627ca5db0-126-ubuntu20.04.1
Id: 4/5


Relay
OUT1
X306.25
OUT2
X306.24
Lamp
GND
X306.26
24 V
X306.1
User input 1
DIN1
X306.9
RS485 N
RS485 P
RS485 B
RS485 A
X305.7
X305.9
MODBUS RTU master (RS485)
vSECC
X301.19
TEMP_2
X301.20
TEMP_1
PT1000
PT1000
X301.17
AGND
X301.18
AGND
X301.1
0-10V_2
X301.3
AGND
V_OUT_SENSOR2
GND_SENSOR2
External sensor with voltage output
X301.2
0-10V_1
X301.5
AGND
V_OUT_SENSOR1
GND_SENSOR1
External sensor with voltage output
ETH1
Ethernet
PE1_ETH
Power Electronics
Reading and setting vSECC's GPIOs is possible using PEP WS messages setOutput and getInput.
ETH
MODBUS TCP client
Web: www.vector.com
Tel: +49 711 / 80670 0
70499 Stuttgart / Germany
Ingersheimer Str. 24
Vector Informatik GmbH
Sheet: /vSECC GPIO PEP WS/
File: vsecc-gpio-pep-ws.kicad_sch
Titles GPIOs with PEP WS and MODBUS bridge
Size: A4
Date: 2023-08-07
Rev: 1.0.1
KiCad E.D.A. kicad 6.0.11-2627ca5db0-126-ubuntu20.04.1
Id: 5/5

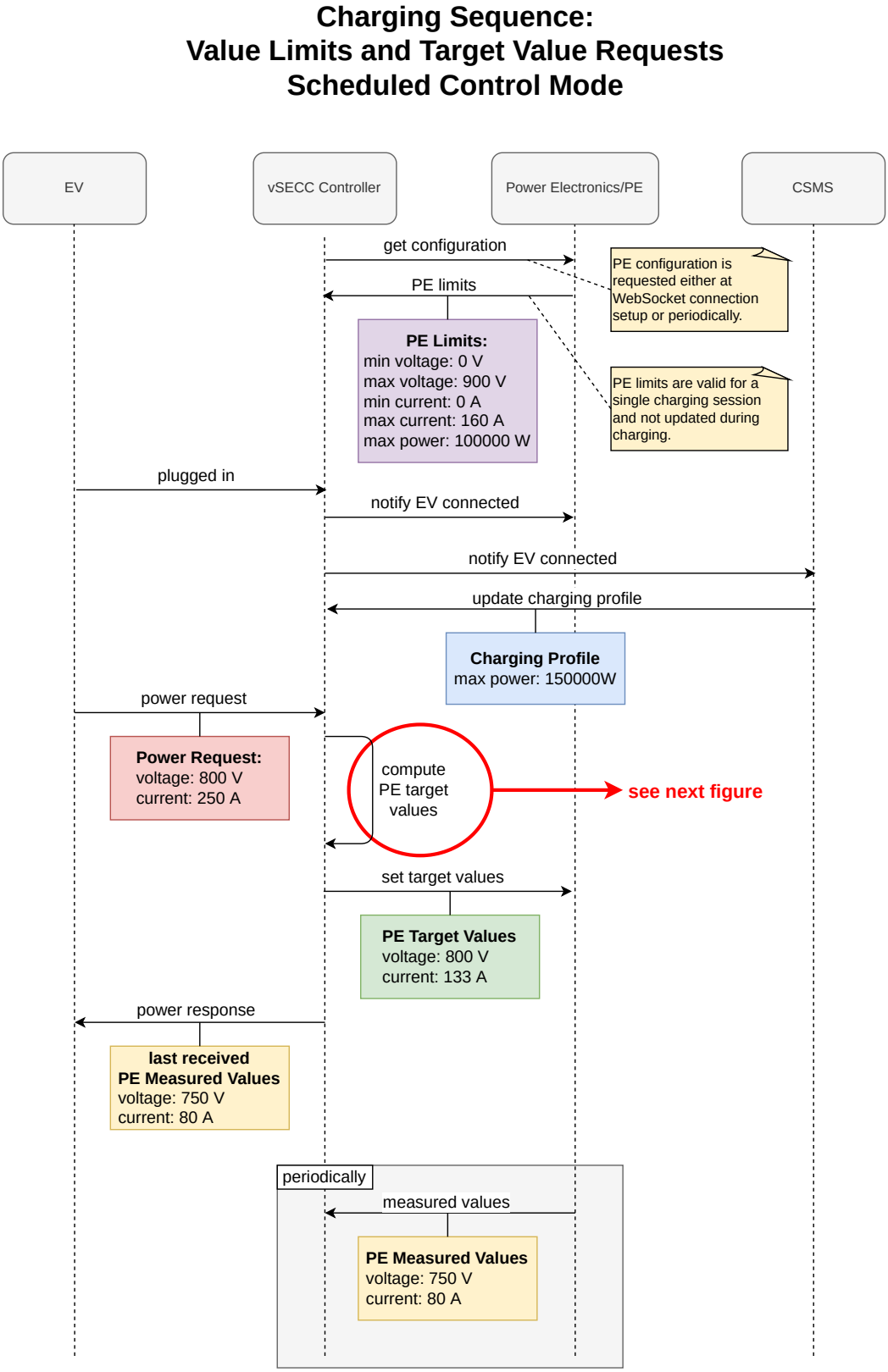
G vSECC.MCS Example Wiring Diagrams





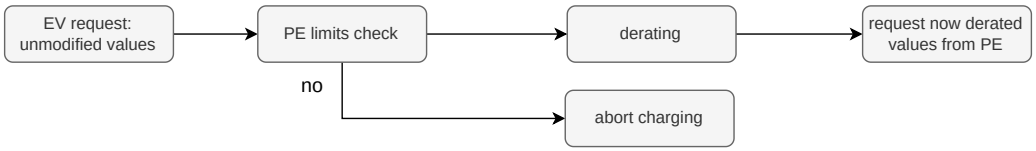
H Limits and Schedules

H.1 Limits and Schedules: Communication Sequence

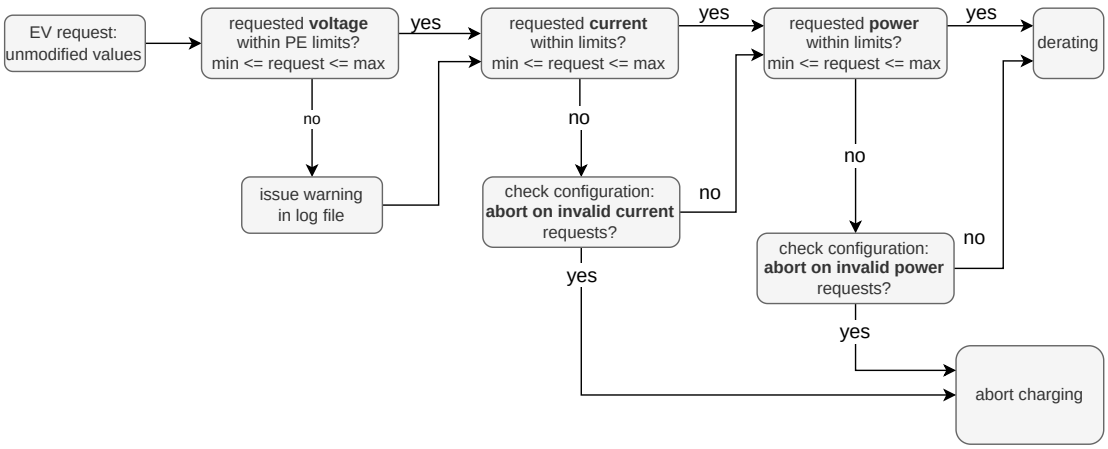


H.2 Limits and Schedules: Limits Checks and Derating

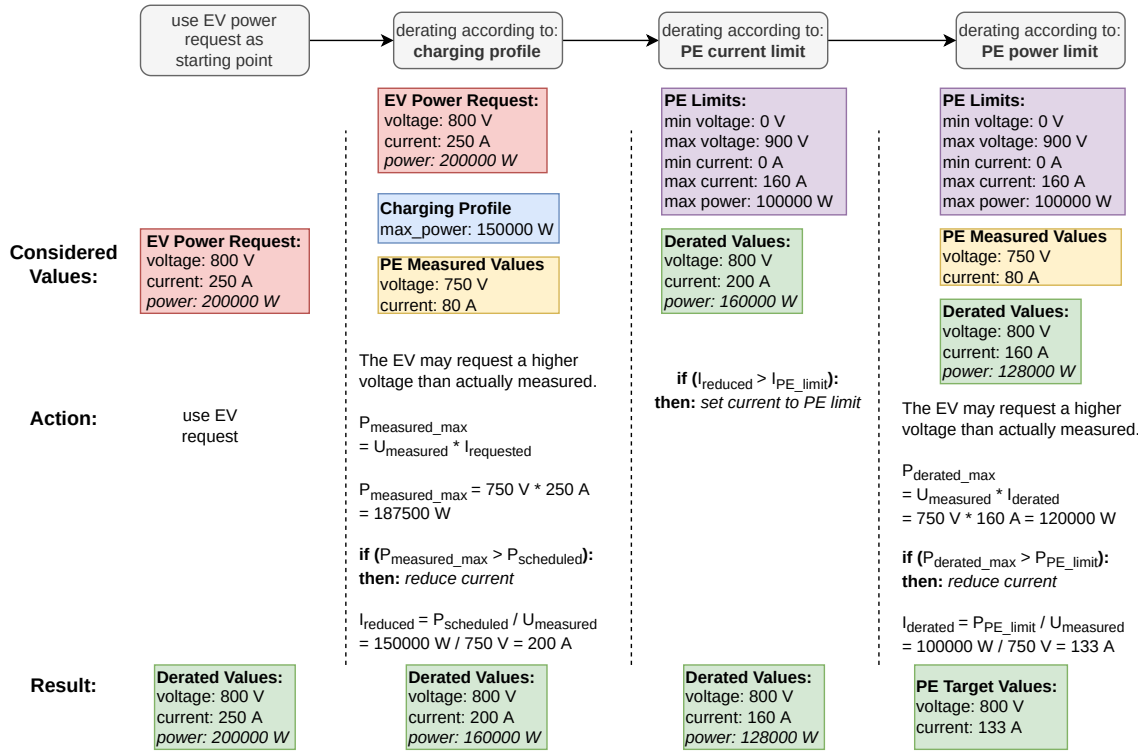
Compute Target Values Scheduled Control Mode: Overview



Compute Target Values Scheduled Control Mode: PE Limits Check



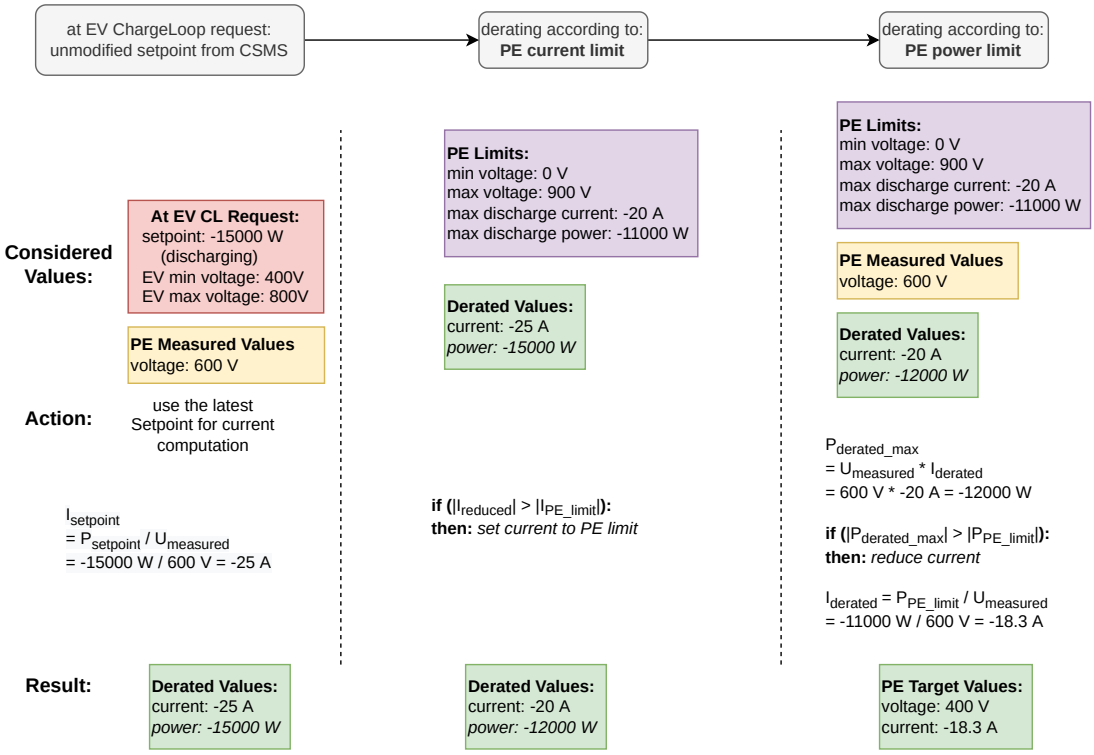
Compute Target Values Scheduled Control Mode: Derating



H.3 Limits and BPT Dynamic Control Mode: Limits Checks and Derating

Compute Target Values Dynamic Control Mode: Overview

Compute Target Values Dynamic Control Mode: Derating

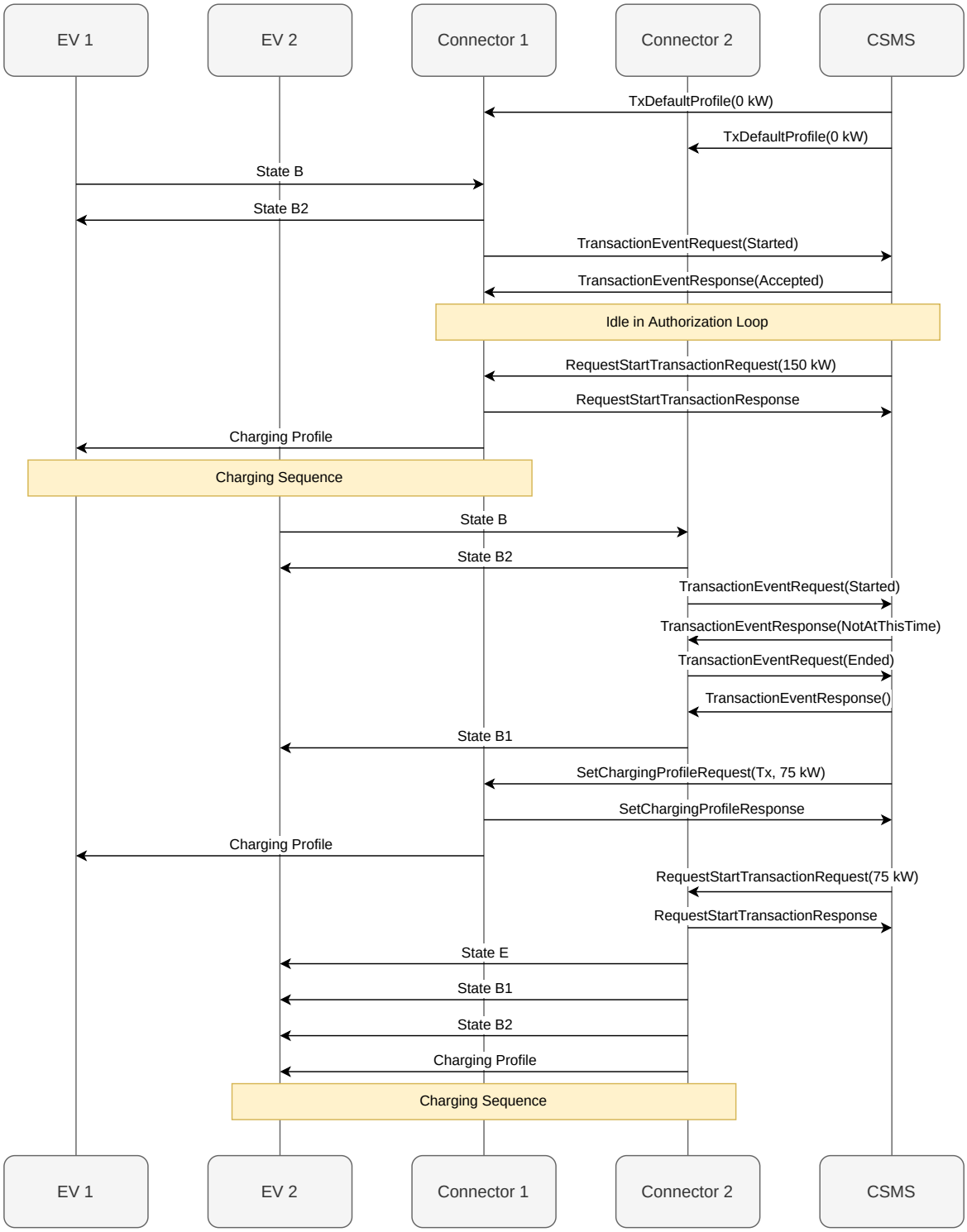


Note: In dynamic control mode the PE determines the voltage and the current to charge/discharge by itself, according to the present needs which are unknown to the vSECC.
The setpoint is a possibility to communicate the requested charge/discharge power over OCPP to the vSECC and from there to the PECC via the target values.
In principle, the PECC is not required to adhere to this setpoint.
The target voltage sent to the PECC is set to the *EV min voltage* for discharging and to the *EV max voltage* for charging.
Under some circumstances, the EV min/max voltage values are optional in the ChargeLoopReq message from the EV. In this case, the min/max values from the ChargeParameterDiscoveryReq (CPD) are used.
The charge/discharge current is determined by the setpoint. This current is calculated using the *PE measured voltage*.

I Restarting a Charging Session

Neither DIN SPEC 70121 nor ISO 15118-2 define a way for the EVSE to reinitialize a charging session after a previous one has already been completed, i.e., [V2G-DC-107], [V2G-DC-116], [V2G2-508] or [V2G2-728] happened.
This severely restricts the flexibility, especially for DIN sessions. E.g., when implementing a charging pause or the OCPP *TransactionEventRequest(NotAtThisTime)* and its counterpart for waking up the EV.
Nevertheless, the vSECC Controller supports a way to achieve exactly this for most vehicles: By purposefully applying Control Pilot (CP) state E while the EV remains plugged in, the vehicle is signalled either a *plug-out* event (if it does not monitor the Proximity Pin) or a severe error condition. By reverting the CP state back to X1 (i.e., state B as long as the EV is still plugged in), the vSECC Controller signals the EV a *plug-in* event or the resolution of the error condition. Most EVs then try to (re)initialize a charging session.
Please see the subsequent sequence diagrams for an illustration of the following use-case: The TxDefaultProfiles for both connector 1 and 2 are set to 0 kW. The maximum available power is currently 150 kW. EV 1 connects and could charge with the whole 150 kW. While EV 1 is charging, EV 2 plugs in and the CSMS notifies the vSECC Controller that charging is currently not possible at connector 2 (*TransactionEventResponse(NotAtThisTime)*). EV 2 thus terminates its charging session and becomes idle. Then, after some time, the power served to EV 1 is reduced by half. Hence, more power is available and connector 2 is notified that the power limit is now 75 kW and an (OCPP) transaction should start. This results in the vSECC Controller executing a *BEB-Toggle* by applying CP state E, X1 and X2 (B1 and B2) which in turn signals the EV that a new charging session should start.
This scenario describes two EVs charging in parallel. A sequential charging could be achieved similarly, by first stopping session 1 (e.g., tiggered by the EV because the battery is fully charged) and then notifying the vSECC Controller about the newly available power at connector 2.

I.1 Sequence Diagram: Restarting a Charging Session



J vSECC MQTT Interface: Description of Imports and Exports

J.1 Exports

J.1.1 EVCCID

content	The EVCCID (vehicle identification string) the charging station received from the EV.
format	A string representing the EVCCID. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/evccid
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.2 Measurement Control

content	The action to be executed by the energy meter.
format	One of ”stop_measurement”, ”start_measurement”, ”get_reading_signed”.
topic	vsecc/connector/{evse_id}/em/measurement_control
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.3 Measured Voltage

content	The voltage measured by the power electronics, given in volts.
format	A string representing the number, formatted as double with 6 decimal places, e.g., ”653.524559”.
topic	vsecc/connector/{evse_id}/pe/measured_voltage
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.4 Measured Current

content	The current measured by the power electronics, given in amperes.
format	A string representing the number, formatted as double with 6 decimal places, e.g., ”11.524559”.
topic	vsecc/connector/{evse_id}/pe/measured_current
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.5 Transaction ID

content	The OCPP transaction ID for current charging session.
format	The transaction ID, e.g., ”12af8962-dfbe-41ea-8038-b2eaf4754c7c”. Empty string if value is cleared or invalid.
topic	vsecc/connector/{evse_id}/ocpp/transaction_id
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.6 Authorization State

content	The authorization state for the given connector, identified by the evse_id. If unauthorized, an EV may connect but is not allowed to charge.
format	One of ”deauthorized” (EV not authorized), ”authorized” (charging authorized), ”pending” (authorization not completed, yet).
topic	vsecc/connector/{evse_id}/status/charging_authorization_state
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.7 EV Communication State

content	The ev communciation state for the given connector, identified by the evse_id.
format	One of ”WAITING_FOR_COMMUNICATION”, ”ESTABLISH_COMMUNICATION”, ”ESTABLISH_COMMUNICATION_ERROR”, ”COMMUNICATION”, ”COMMUNICATION_ERROR”
topic	vsecc/connector/{evse_id}/ocpp/ev_communication_state
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.8 EV Authorization Token assigned to EVSE

content	The authorization token that corresponds to the authorization status (published previously) for the given connector, identified by the evse_id.
format	The authorization token as string. Empty string if no token is currently assigned.
topic	vsecc/connector/{evse_id}/ocpp/ev_authorization_token
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.9 Currently unassigned tokens

content	All unassigned tokens.
format	CSV list of strings, e.g.: [”4A2FB33F”, ”IDTAG1”]. Empty list if no unassigned token exists: []
topic	vsecc/unassigned_tokens

J.1.10 Display Message Raw

content	Messages received over OCPP via SetDisplayMessageRequest.
format	Raw string representation of the OCPP payload.
topic	vsecc/display_message_raw

J.1.11 Tariff ID

content	If available: The tariff ID of the selected charging profile.
format	A string representing the tariff ID as number, formatted as integer, e.g., ”0”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/tariff_id
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.12 EVSE ID String

content	Each charging point has a unique ID which must not be confused with the OCPP evse_id. This ID varies between used charging protocols and must be set by the charging point operator. In case of DIN 70121 charging, this ID is specified in DIN SPEC 91286. ISO 15118-2 charging specifies the ID format in Section H.2
format	The EVSEID as string, e.g., ”49*564543*01” (DIN) or ”DE*VEC*EOEC*S01” (ISO) or ”chademo” (CHAdeMO, no ’real’ ID is available with this charging protocol).
topic	vsecc/connector/{evse_id}/status/evse_id_string
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.13 Transferred Energy

content	If available: The transferred energy so far for the current OCPP Transaction. Is only available if the energy meter is NOT disabled (i.e. virtual or real). Values are computed from unsigned readings.
format	A string representing a meter reading, formatted as double with 6 decimal places, e.g., ”11.524559”. The unit is kWh.
topic	vsecc/connector/{evse_id}/em/transferred_energy
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.14 EV Error Code (EvErrorCode)

content	If available: The most recent EvErrorCode the vSECC Controller received from the EV. Applies to ISO-2 and DIN only. The actual strings may be subject to change in the future.
format	The raw string given by the vSECClib.CCS, e.g., ”VSECCLIB_DC_EVERROR_CODE_TYPE_NO_ERROR”. Published only if the error code has changed. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/ev_error_code
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.15 Control Pilot (CP) State

content	If available: The most recent CP state of the respective connector. The actual strings may be subject to change in the future. State E is only published if forced externally (short to ground). State E is not published if applied by the vSECC for a short time, e.g. to wake up a vehicle.
format	One of ”state_a”, ”state_b”, ”state_c”, ”state_d”, ”state_e”, ”state_f”, ”state_invalid”. Published only if the CP state has changed.
topic	vsecc/connector/{evse_id}/ev/cp_state
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.16 Charge Enable (CE) State

content	If available: The most recent CE state of the respective connector. The actual strings may be subject to change in the future.
format	One of ”state_a”, ”state_b0”, ”state_b0_aux”, ”state_b”, ”state_b_aux”, ”state_c”, ”state_ec”, ”state_e”, ”state_invalid”. Published only if the CE state has changed.
topic	vsecc/connector/{evse_id}/ev/ce_state
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.17 Insertion Detection (ID) State

content	If available: The most recent ID state of the respective connector. The actual strings may be subject to change in the future.
format	One of ”unmated”, ”mated”, ”mated_ev_aux”, ”mated_evse_aux”, ”invalid”. Published only if the ID state has changed.
topic	vsecc/connector/{evse_id}/ev/id_state
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.18 Proximity Pin (PP) State

content	If available: The most recent PP state of the respective connector. The actual strings may be subject to change in the future.
format	One of ”disconnected”, ”s3_open”, ”s3_closed”, ”faulted”. Published only if the PP state has changed.
topic	vsecc/connector/{evse_id}/ev/pp_state
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.19 State of Charge (SoC)

content	If available: The current battery state of charge (SoC) of the EV.
format	A string representing a percentage, formatted as integer, e.g., ”43”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/soc
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.20 Time To Full SoC

content	If available: The remaining time until the currently charging EV has reached its full state of charge (SoC).
format	A string representing the time in seconds, formatted as integer, e.g., ”534”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/time_to_full_soc
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.21 Charging Session State

content	The standard-specific state the charging process is currently in. The actual strings may be subject to change in the future.
format	A string representing the state, e.g. ”ChargeParameterDiscovery” for CCS DC. For CCS AC with Basic Signalling, the following strings are used: ”evse_not_ready”, ”evse_ready”, ”session_running”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/status/charging_session_state
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.22 Charging Standard

content	The charging standard used in the currently running charging session. It is published at the beginning of a charging session.
format	One of ”iso15118_2”, iso15118_20, din70121 , ”chademo”, ”oppcharge”, ”iec61851_1”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/status/charging_standard
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.23 OCPP Connection Status

content	The connection status to the CSMS of the vSECC Controller. Published on change.
format	One of ”disconnected”, ”connecting”, ”connected, ”error_on_write”, ”error_on_connection”, ”error_on_handshake”, ”error_on_read”.
topic	vsecc/ocpp_connection_status

J.1.24 Digital Out

content	Set the specified digital out port.
format	”1” for logical high, ”0” for logical low.
topic	vsecc/io/d_out/{d_out_id}/value
topic_parameter	**d_out_id**: The digital out port to be set.

J.1.25 Active Failures

content	Summary of the active failures across all components of the specified EVSE.
format	CSV list of strings, e.g.: [”Failure1”,”Failure2”]. Empty list if no failure exists: []
topic	vsecc/connector/{evse_id}/status/components/active_failures
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.26 Availability

content	Availability of the specified EVSE
format	”operative” if the EVSE is available, ”inoperative” otherwise.
topic	vsecc/connector/{evse_id}/status/availability
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”. ”0” refers to the whole charging station.

J.1.27 Component Failure

content	Active failure of a component
format	Descriptive string of the failure state, e.g. ”pe_inoperative”.
topic	vsecc/connector/{evse_id}/status/components/report_failure
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.28 Component Failure Resolution

content	Resolved failure of a component
format	Descriptive string of the resolved failure state, e.g. ”pe_inoperative”.
topic	vsecc/connector/{evse_id}/status/components/report_resolution
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.29 Readiness

content	Status of the vSECC Controller initialization. Is published right after boot-up when the vSECC Controller is ready to process MQTT messages. Is published just before shutdown when the vSECC Controller is not able to process MQTT messages anymore.
format	Either ”ready” after the initialization or ”shutdown” just before the shutdown is executed.
topic	vsecc/readiness

J.1.30 Firmware Version

content	The firmware version the vSECC Controller is currently running. Is published once at startup.
format	A string representing the firmware version, e.g. ”2.8.1”.
topic	vsecc/firmware_version

J.1.31 Log Level

content	The configured log level of the vSECC Controller. Is published once at startup.
format	One of ”TRACE”, ”DEBUG”, ”INFO”, ”WARN”, ”ERROR”.
topic	vsecc/preview/log_level

J.1.32 Tariff Raw

content	The tariff as it has been set by the CSMS in the ’TariffFallbackMessage’ device model variable.
format	The string contained in the ’TariffFallbackMessage’ device model variable.
topic	vsecc/tariff_raw

J.1.33 Tariff

content	Tariff value extracted from the ’TariffFallbackMessage’ device model variable. Published only if extraction succeeded.
format	A string with the following format: ”<value> <currency>/kWh”, e.g. ”0.42 EUR/kWh”. Per default, <value> is a decimal value with a dot as decimal separator and two decimal places. <currency> is a 3 character currency code from ISO 4217. Depending on the configured regular expression, the actual format may differ.
topic	vsecc/tariff

J.1.34 Cost

content	Last received cost value from the CSMS for the selected connector. Continuously updated during a transaction. For additional info on the transaction subscribe to the topic vsecc/connector/evse_id/ocpp/transaction_id.
format	A string with the following format: ”<value> <currency>”, e.g. ”32.42 EUR”. <value> is a decimal value with a dot as decimal separator and two decimal places. <currency> is the 3 character currency code set in the ’Currency’ device model variable.
topic	vsecc/connector/{evse_id}/ocpp/cost

J.1.35 RFID Token

content	Last detected RFID token.
format	A string consisting of the token identifier.
topic	vsecc/rfid/token

J.1.36 Changed Variable

content	VarId of the changed variable. This includes VarIds of variables that are not accessible via the WebUi-API. The external usefulness of this topic is therefore limited and thus not intended for customer usage.
format	A string containing the VarId of the changed variable
topic	vsecc/config_change

J.1.37 Token Authorization Status

content	The token's status as reported by the CSMS and the token itself.
format	JSON containing the keys "token" and "status". Status: One of: "accepted", "blocked", "concurrent_tx", "expired", "invalid", "not_allowed_type_evse", "not_at_this_location", "not_at_this_time", "no_credit", "unknown"
topic	vsecc/authorization_token_status

J.1.38 Charging Profile Composite Schedule

content	The composite schedule specific to evse_id. It is computed from all installed charging profiles and the static PE power limit.
format	A JSON string representing the profile with all its periods. No gaps between periods exist. Empty string if no profile is active (anymore). Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/preview/composite_schedule

J.1.39 V2G Session ID

content	Session ID of the V2G session in case of DIN or ISO charging
format	Hex representation as string. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/v2g_session_id

J.1.40 Minimum State of Charge (SoC)

content	If available: The minimum allowed battery state of charge (SoC) of the EV for discharging.
format	A string representing a percentage, formatted as integer, e.g., ”43”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/minimum_soc
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.41 Target State of Charge (SoC)

content	If available: The target battery state of charge (SoC) of the EV.
format	A string representing a percentage, formatted as integer, e.g., ”43”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/target_soc
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.42 Charge mode

content	The charge mode of the running charging session
format	One of: ”DYNAMIC”, ”DYNAMIC_BPT”, ”SCHEDULED”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/charge_mode_info
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.43 Departure Time

content	The departure time as communicated by the EV, if available
format	UTC time, YYYY-MM-DD HH:MM:SS, e.g. 2024-01-31 14:25:10. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/departure_time
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.44 Time To Bulk SoC

content	If available: The remaining time until the currently charging EV has reached its bulk state of charge (SoC).
format	A string representing the time in seconds, formatted as integer, e.g., ”534”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/time_to_bulk_soc
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.45 Charging Complete

content	If available: The charging complete flag as communicated by the EV
format	One of: ”true”, ”false”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/charging_complete
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.46 Bulk Charging Complete

content	If available: The bulk charging complete flag as communicated by the EV
format	One of: ”true”, ”false”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/bulk_charging_complete
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.47 Bulk State of Charge (SoC)

content	If available: The battery state of charge (SoC) of the EV when bulk charging is complete.
format	A string representing a percentage, formatted as integer, e.g., ”43”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/bulk_soc
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.48 Full State of Charge (SoC)

content	If available: The battery state of charge (SoC) of the EV when charging is fully complete.
format	A string representing a percentage, formatted as integer, e.g., ”43”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/full_soc
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.49 CP Max Voltage

content	CP max voltage measured in Volts, only applicable if the vSECC controller does not force CP state E or F
format	A string representing a voltage, formatted as decimal, e.g., ”9.0”.
topic	vsecc/connector/{evse_id}/ev/cp_max_voltage
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.50 CP Min Voltage

content	CP min voltage measured in Volts, only applicable if the vSECC controller does not force CP state E or F
format	A string representing a voltage, formatted as decimal, e.g., ”-12.0”.
topic	vsecc/connector/{evse_id}/ev/cp_min_voltage
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.51 CP PWM Duty Cycle

content	Duty cycle of the CP PWM signal in percent, only applicable if the vSECC controller does not force CP state E
format	A string representing a percentage, formatted as integer, e.g., ”5”.
topic	vsecc/connector/{evse_id}/ev/cp_pwm_duty_cycle
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.52 Proximity Pin

content	Value of the proximity pin, only relevant for CCS type 1
format	”1” for logical high, ”0” for logical low.
topic	vsecc/connector/{evse_id}/ev/proximity_pin
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.53 EV Supported Schemas

content	The supported schemas offered by the EV in the SupportedAppProtocol message
format	CSV list of strings, e.g.: [”urn:iso:15118:2:2013:MsgDef”,”urn:din:70121:2012:MsgDef”]. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/supported_schemas
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.54 EV Min Voltage Limit

content	The minimum voltage limit in Volts provided by the EV
format	A string representing a voltage, formatted as decimal, e.g., ”0.0”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/limits/min_voltage
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.55 EV Max Voltage Limit

content	The maximum voltage limit in Volts provided by the EV
format	A string representing a voltage, formatted as decimal, e.g., ”495.0”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/limits/max_voltage
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.56 EV Min Charge Current Limit

content	The minimum charge current limit in Amperes provided by the EV
format	A string representing a current, formatted as decimal, e.g., ”0.0”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/limits/min_current
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.57 EV Max Charge Current Limit

content	The maximum charge current limit in Amperes provided by the EV
format	A string representing a current, formatted as decimal, e.g., ”100.0”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/limits/max_current
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.58 EV Min Charge Power Limit

content	The minimum charge power limit in Watts provided by the EV
format	A string representing a power, formatted as decimal, e.g., ”0.0”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/limits/min_power
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.59 EV Max Charge Power Limit

content	The maximum charge power limit in Watts provided by the EV
format	A string representing a power, formatted as decimal, e.g., ”100000.0”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/limits/max_power
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.60 EV Min Discharge Current Limit

content	The minimum discharge current limit in Amperes provided by the EV
format	A string representing a negative current, formatted as decimal, e.g., ”0.0”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/limits/min_discharge_current
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.61 EV Max Discharge Current Limit

content	The maximum discharge current limit in Amperes provided by the EV
format	A string representing a negative current, formatted as decimal, e.g., ”-100.0”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/limits/max_discharge_current
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.62 EV Min Discharge Power Limit

content	The minimum discharge power limit in Watts provided by the EV
format	A string representing a negative power, formatted as decimal, e.g., ”0.0”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/limits/min_discharge_power
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.63 EV Max Discharge Power Limit

content	The maximum discharge power limit in Watts provided by the EV
format	A string representing a negative power, formatted as decimal, e.g., ”-100000.0”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/limits/max_discharge_power
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.64 EV Requested Voltage

content	The requested voltage in Volts from the EV
format	A string representing a voltage, formatted as decimal, e.g., ”495.0”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/requested_voltage
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.65 EV Requested Current

content	The requested current in Amperes from the EV
format	A string representing a current, formatted as decimal, e.g., ”120.0”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/requested_current
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.66 EV Ready

content	The EVReady flag sent by the EV, signalling if the EV is ready to charge
format	One of: ”true”, ”false”. Cleared with the empty string after a charging session has ended.
topic	vsecc/connector/{evse_id}/ev/ev_ready
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.1.67 OCPP Active Network Connection Profile Slot

content	The currently active Network Connection Profile slot that is used to connect to the CSMS. (Note: This topic is retained in MQTT)
format	”1”, ”2”, ”3” for the respective slots, ”-1” for the fallback slot
topic	vsecc/ocpp_active_network_profile_slot

J.1.68 External Peripheral Firmware Update Pending

content	The filename of the pending firmware update file
format	A string representing a case-sensitive filename
topic	vsecc/firmware_external/update_pending

J.1.69 EVSE reserved via OCPP

content	Reservation status of an EVSE
format	One of: ”true”, ”false”
topic	vsecc/connector/{evse_id}/status/reserved

J.1.70 A DataTransfer request message from the CSMS was received via OCPP

content	the DataTransfer request message
format	A string as sent form CSMS - should be in JSON format
topic	vsecc/ocpp_data_transfer/received_request_from_csms

J.1.71 A DataTransfer response message from the CSMS was received via OCPP

content	the DataTransfer response message
format	A string as sent form CSMS - should be in JSON format
topic	vsecc/ocpp_data_transfer/received_response_from_csms

J.1.72 Something went wrong regarding a DataTransfer request from the CSMS

content	the error encountered
format	A string representing the error encountered
topic	vsecc/ocpp_data_transfer/data_transfer_from_csms_notification

J.1.73 Something went wrong regarding a DataTransfer request to the CSMS

content	the error encountered
format	A string representing the error encountered
topic	vsecc/ocpp_data_transfer/data_transfer_to_csms_notification

J.1.74 Reset of the Charging Station is requested by the CSMS

content	type of reset that is requested
format	One of: ”soft”, ”hard”
topic	vsecc/ocpp/reset

J.1.75 Wait for Power Electronics Readiness

content	The remaining time until the SECC continues the ChargeParameterDiscovery (DIN SPEC 70121/ISO 15118-2/SAE J3105) or ScheduleExchange (ISO 15118-20) loop. Empty string if not applicable (OppCharge). Published whenever the content changes.
format	A string representing the remaining wait time in seconds, formatted as integer. Cleared with the empty string if the information is not valid anymore (e.g. after the CPD-phase is completed) or after a charging session has ended.
topic	vsecc/connector/{evse_id}/status/ waiting_for_power_electronics_ready
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.2 Imports

J.2.1 Report Power Electronics Readiness

content	None/empty string. The reception of this event itself signals the power electronics' readiness.
format	Empty string.
topic	vsecc/connector/{evse_id}/status/report_power_electronics_ready
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.2.2 RFID Pairing

content	The current status of the RFID pairing mechanism. Is not updated by the vSECC Controller, i.e., every value must be set by an external entity. Used only when charging according to OppCharge or SAE J3105. When used with OppCharge, don't forget to set the configuration variable ”use_rfid_pairing_for_oppcharge” to true, too.
format	One of ”OK”, ”FAILED”, ”PENDING”
topic	vsecc/connector/{evse_id}/pantograph/rfid_pairing
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.2.3 Measurement Status

content	The current measurement status of and reported by the energy meter.
format	Either ”running” or ”not_running”.
topic	vsecc/connector/{evse_id}/em/measurement_status
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.2.4 Signed Reading

content	A signed reading reported by the energy meter, e.g. after a control command ”get_reading_signed”.
format	An OCMF-like string with the following format: OCMF|<JSON1>|<JSON2>|”PUBKEY”:”<PUBKEY>” XOR the string ”NOT_SUPPORTED”.
topic	vsecc/connector/{evse_id}/em/signed_reading
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.2.5 Unsigned Reading

content	An unsigned reading reported by the energy meter. Unsigned readings are published in an interval of 10s.
format	A string with the following format, timestamp is given as explained in RFC3339: <Timestamp>|<Import reading>|<Export reading>
topic	vsecc/connector/{evse_id}/em/unsigned_reading
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.2.6 external_measurand

content	Metric from external source (e.g. a temperature sensor or fan RPM information) which is buffered and sent to the CSMS.
format	JSON mandatory containing a ”value” and ”timestamp”, and optionally a ”measurand”, ”phase”, ”location”,”unit”, ”unit_multiplier” and signed_meter_value struct (which itself consists of a mandatory ”signed_meter_data”, ”signing_method”, ”encoding_method” and ”public_key”). Messages must validate against the JSON schema with the title ”external_measurand”.
topic	vsecc/connector/{evse_id}/ocpp/external_measurand
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.2.7 Energy Meter Log Entry

content	Log entries from the energy meter.
format	One log entry per MQTT message.
topic	vsecc/connector/{evse_id}/em/log
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.2.8 Digital In

content	The value of the digital IN port.
format	”1” for logical high, ”0” for logical low.
topic	vsecc/io/d_in/{d_in_id}/value
topic_parameter	**d_in_id**: The ID of the digital in port.

J.2.9 Analog In

content	The value of the analog IN port, unit is Volts (V).
format	Float value as string, e.g. ”1.253”.
topic	vsecc/io/a_in/{a_in_id}/value
topic_parameter	**a_in_id**: The IT of the analog in port.

J.2.10 Temperature In

content	The value of the temperature IN port, unit is Degrees Celsius (°C).
format	Float value as string, e.g. ”25.3”.
topic	vsecc/io/t_in/{t_in_id}/temperature
topic_parameter	**t_in_id**: The ID of the temperature IN port.

J.2.11 Temperature In Resistance

content	The resistance value of the temperature IN, unit is Ohms.
format	Integer value as string, e.g. ”3483”.
topic	vsecc/io/t_in/{t_in_id}/resistance
topic_parameter	**t_in_id**: The id of the temperature in port.

J.2.12 Component Failure

content	Signals that a failure state is active. EVSE will become unavailable. Must be resolved using the report_resolution topic to clear the failure.
format	Descriptive string of the failure state, e.g. ”display_connection_loss”. Must match the string of the report_resolution topic on clearing the failure.
topic	vsecc/connector/{evse_id}/status/components/report_failure
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.2.13 Component Failure Resolution

content	Signals that a failure state has been resolved. EVSE will become available once all failures have been cleared.
format	Descriptive string of the failure state, e.g. ”display_connection_loss”. Must match the string of the report_failure topic that caused the failure.
topic	vsecc/connector/{evse_id}/status/components/report_resolution
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.2.14 Reinit EV to start charging

content	Publishing to this topic triggers a reinitialization of the EV in order to restart the charging process. This may or may not work, depending on the EV. Message content is not used.
format	Content should be null.
topic	vsecc/connector/{evse_id}/preview/start_charging_reinit_ev
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.2.15 Stop Charging

content	Publishing to this topic triggers a charging stop in the same manner as pressing the stop button would. Message content is not used.
format	Content should be null.
topic	vsecc/connector/{evse_id}/preview/stop_charging
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.2.16 Add Token

content	Authorization token, token type, and token preauthorization status
format	Content should be a JSON struct containing the ”idToken” (string), ”type” (string), and ”preauthorized” (boolean) fields. Preauthorized tokens are immediately considered authorized and will not trigger an AuthorizeRequest. The maximum length of the idToken is limited by the active OCPP version. For allowed values of ”type”, see IdTokenEnumType in OCPP 2.0.1.
topic	vsecc/add_token

J.2.17 Changed Variable

content	VarId of the changed variable
format	VarId of the changed variable as a string
topic	vsecc/config_change

J.2.18 Assign a token to an EVSE

content	the token to assign
format	String of the token that should be assigned
topic	vsecc/connector/{evse_id}/assign_token
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.2.19 Report status information for the charging station

content	OCPP 1.6 ChargePointErrorCode and optionally a vendor id, vendor error code or/and an additional info
format	Content should be a JSON struct containing the field ”errorCode” (string) and optionally ”vendorId” (string), ”vendorErrorCode” (string), ”info” (string). For allowed values of ”errorCode”, see ChargePointErrorCode in OCPP 1.6.
topic	vsecc/status/report_status_information

J.2.20 Report status information for a connector

content	OCPP 1.6 ChargePointErrorCode and optionally a vendor id, vendor error code or/and an additional info
format	Content should be a JSON struct containing the field ”errorCode” (string) and optionally ”vendorId” (string), ”vendorErrorCode” (string), ”info” (string). For allowed values of ”errorCode”, see ChargePointErrorCode in OCPP 1.6.
topic	vsecc/connector/{evse_id}/status/report_status_information
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”.

J.2.21 Report status information for the ongoing firmware update of an external peripheral

content	Installation status and filename of the external firmware file as published by ”vsecc/firmware_external/update_pending”
format	Content should be a JSON struct containing the field ”status” (string: ”INSTALLING”|”REBOOTING”|”FAILED”|”SUCCESS”) and ”file” (string)
topic	vsecc/firmware_external/report_status

J.2.22 Send DataTransfer response to CSMS after a DataTransfer request has been received

content	The response to the last DataTransfer that was received, field ”statusInfo” is ignored for OCPP 1.6
format	Content should be a JSON struct containing the fields ”status” and optionally ”data” and ”statusInfo” according to DataTransferResponse in OCPP 2.0.1
topic	vsecc/ocpp_data_transfer/send_response_to_csms

J.2.23 Send DataTransfer request to CSMS

content	The DataTransfer request that should be sent to the CSMS
format	Content should be a JSON struct containing the fields ”vendorId” and optionally ”messageId” and ”data” according to DataTransferRequest in OCPP 2.0.1
topic	vsecc/ocpp_data_transfer/send_request_to_csms

J.2.24 Change Availability of EVSE or Charging Station

content	Target operational status
format	One of: ”operative”, ”inoperative”
topic	vsecc/connector/{evse_id}/status/set_availability
topic_parameter	**evse_id**: The evse_id the message corresponds to, e.g., ”1”. ”0” refers to the whole charging station.

J.3 JSON Schemas

J.3.1 vsecc/connector/{evse_id}/status/report_status_information

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "mqtt:vsecc/connector/{evse_id}/status/report_status_information",
  "title": "vsecc/connector/{evse_id}/status/report_status_information",
  "type": "object",
  "properties": {
    "errorCode": {
      "type": "string",
      "additionalProperties": false,
      "enum": [
        "ConnectorLockFailure",
        "EVCommunicationError",
        "GroundFailure",
        "HighTemperature",
        "InternalError",
        "LocalListConflict",
        "NoError",
        "OtherError",
        "OverCurrentFailure",
        "OverVoltage",
        "PowerMeterFailure",
        "PowerSwitchFailure",
        "ReaderFailure",
        "ResetFailure ",
        "UnderVoltage",
        "WeakSignal"
      ]
    },
    "vendorId": {
      "type": "string",
      "maxLength": 255
    },
    "vendorErrorCode": {
      "type": "string",
      "maxLength": 50
    },
    "info": {
      "type": "string",
      "maxLength": 50
    }
  },
  "additionalProperties": false,
  "required": [
    "errorCode"
  ]
}
```

J.3.2 vsecc/add_token

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "mqtt:vsecc/add_token",
  "title": "vsecc/add_token",
  "type": "object",
  "properties": {
    "idToken": {
      "type": "string",
      "maxLength": 36
    },
    "type": {
      "type": "string",
      "additionalProperties": false,
      "enum": [
        "eMAID",
        "ISO14443",
        "ISO15693",
        "KeyCode",
        "Local",
        "MacAddress",
        "Central",
        "NoAuthorization"
      ]
    },
    "preauthorized": {
      "type": "boolean"
    }
  },
  "additionalProperties": false,
  "required": [
    "idToken",
    "type",
    "preauthorized"
  ]
}
```

J.3.3 vsecc/connector/{evse_id}/ocpp/external_measurand

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "mqtt:vsecc/connector/{evse_id}/ocpp/external_measurand",
  "title": "vsecc/connector/{evse_id}/ocpp/external_measurand",
  "type": "object",
  "properties": {
    "value": {
      "type": "number"
    },
    "measurand": {
      "type": "string",
      "additionalProperties": false,
      "enum": [
        "CURRENT_EXPORT",
```

```
15          "CURRENT_IMPORT",
16          "CURRENT_OFFERED",
17          "ENERGY_ACTIVE_EXPORT_REGISTER",
18          "ENERGY_ACTIVE_IMPORT_REGISTER",
19          "ENERGY_REACTIVE_EXPORT_REGISTER",
20          "ENERGY_REACTIVE_IMPORT_REGISTER",
21          "ENERGY_ACTIVE_EXPORT_INTERVAL",
22          "ENERGY_ACTIVE_IMPORT_INTERVAL",
23          "ENERGY_ACTIVE_NET",
24          "ENERGY_REACTIVE_EXPORT_INTERVAL",
25          "ENERGY_REACTIVE_IMPORT_INTERVAL",
26          "ENERGY_REACTIVE_NET",
27          "ENERGY_APPARENT_NET",
28          "ENERGY_APPARENT_IMPORT",
29          "ENERGY_APPARENT_EXPORT",
30          "FREQUENCY",
31          "POWER_ACTIVE_EXPORT",
32          "POWER_ACTIVE_IMPORT",
33          "POWER_FACTOR",
34          "POWER_OFFERED",
35          "POWER_REACTIVE_EXPORT",
36          "POWER_REACTIVE_IMPORT",
37          "SOC",
38          "VOLTAGE",
39          "RPM",
40          "TEMPERATURE"
41        ]
42      },
43      "phase": {
44        "type": "string",
45        "additionalProperties": false,
46        "enum": [
47          "L1",
48          "L1_L2",
49          "L1_N",
50          "L2",
51          "L2_L3",
52          "L2_N",
53          "L3",
54          "L3_L1",
55          "L3_N",
56          "N"
57        ]
58      },
59      "location": {
60        "type": "string",
61        "additionalProperties": false,
62        "enum": [
63          "BODY",
64          "CABLE",
65          "EV",
66          "INLET",
67          "OUTLET"
68        ]
69      },
```

```
70      "unit": {
71        "type": "string",
72        "maxLength": 20
73      },
74      "unit_multiplier": {
75        "type": "integer"
76      },
77      "signed_meter_value": {
78        "type": "object",
79        "properties": {
80          "signed_meter_data": {
81            "type": "string",
82            "maxLength": 2500
83          },
84          "signing_method": {
85            "type": "string",
86            "maxLength": 50
87          },
88          "encoding_method": {
89            "type": "string",
90            "maxLength": 50
91          },
92          "public_key": {
93            "type": "string",
94            "maxLength": 2500
95          }
96        },
97        "additionalProperties": false,
98        "required": [
99          "signed_meter_data",
100         "signing_method",
101         "encoding_method",
102         "public_key"
103       ]
104     },
105     "timestamp": {
106       "type": "string",
107       "format": "date-time"
108     }
109   },
110   "additionalProperties": false,
111   "required": [
112     "value",
113     "timestamp"
114   ],
115   "allOf": [
116     {
117       "not": {
118         "properties": {
119           "measurand": {
120             "enum": [
121               "ENERGY_ACTIVE_IMPORT_REGISTER"
122             ]
123           },
124           "location": {
```

```
            "not": {}
          }
        }
      }
    }
  ]
}
```

J.3.4 vsecc/ocpp_data_transfer/data_transfer_to_csms_notification

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "mqtt:vsecc/ocpp_data_transfer/data_transfer_to_csms_notification",
  "title": "vsecc/ocpp_data_transfer/data_transfer_to_csms_notification",
  "type": "object",
  "properties": {
    "info": {
      "type": "string",
      "enum": [
        "Timeout",
        "TransferAlreadyOngoing",
        "InvalidFormat"
      ]
    }
  },
  "additionalProperties": false,
  "required": [
    "info"
  ]
}
```

J.3.5 vsecc/firmware_external/report_status

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "mqtt:vsecc/firmware_external/report_status",
  "title": "vsecc/firmware_external/report_status",
  "type": "object",
  "properties": {
    "status": {
      "type": "string",
      "additionalProperties": false,
      "enum": [
        "INSTALLING",
        "REBOOTING",
        "FAILED",
        "SUCCESS"
      ]
    },
    "file": {
      "type": "string",
      "maxLength": 4096
```

```
    }
  },
  "additionalProperties": false,
  "required": [
    "status",
    "file"
  ]
}
```

J.3.6 vsecc/ocpp_data_transfer/data_transfer_from_csms_notification

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "mqtt:vsecc/ocpp_data_transfer/data_transfer_from_csms_notification",
  "title": "vsecc/ocpp_data_transfer/data_transfer_from_csms_notification",
  "type": "object",
  "properties": {
    "info": {
      "type": "string",
      "enum": [
        "Timeout",
        "NoTransferOngoing",
        "InvalidFormat"
      ]
    }
  },
  "additionalProperties": false,
  "required": [
    "info"
  ]
}
```

J.3.7 vsecc/ocpp_data_transfer/send_request_to_csms

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "mqtt:vsecc/ocpp_data_transfer/send_request_to_csms",
  "title": "vsecc/ocpp_data_transfer/send_request_to_csms",
  "type": "object",
  "properties": {
    "vendorId": {
      "type": "string"
    },
    "messageId": {
      "type": "string"
    },
    "data": {}
  },
  "additionalProperties": false,
  "required": [
    "vendorId"
  ]
```

```
}
```

J.3.8 vsecc/ocpp_data_transfer/send_response_to_csms

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "mqtt:vsecc/ocpp_data_transfer/send_response_to_csms",
  "title": "vsecc/ocpp_data_transfer/send_response_to_csms",
  "type": "object",
  "properties": {
    "status": {
      "type": "string",
      "enum": [
        "Accepted",
        "Rejected",
        "UnknownMessageId",
        "UnknownVendorId"
      ]
    },
    "data": {},
    "statusInfo": {
      "type": "object",
      "properties": {
        "reasonCode": {
          "type": "string",
          "enum": [
            "CsNotAccepted",
            "DuplicateProfile",
            "DuplicateRequestId",
            "FixedCable",
            "FwUpdateInProgress",
            "InternalError",
            "InvalidCertificate",
            "InvalidCsr",
            "InvalidIdToken",
            "InvalidMessageSequence",
            "InvalidProfile",
            "InvalidSchedule",
            "InvalidStackLevel",
            "InvalidUrl",
            "InvalidValue",
            "MissingParam",
            "NoCable",
            "NoError",
            "NotEnabled",
            "NotFound",
            "OutOfMemory",
            "OutOfStorage",
            "ReadOnly",
            "TooLargeElement",
            "TooManyElements",
            "TxInProgress",
            "TxNotFound",
            "TxStarted",
```

```
            "UnknownConnectorId",
            "UnknownConnectorType",
            "UnknownEvse",
            "UnknownTxId",
            "Unspecified",
            "UnsupportedParam",
            "UnsupportedRateUnit",
            "UnsupportedRequest",
            "ValueOutOfRange",
            "ValuePositiveOnly",
            "ValueTooHigh",
            "ValueTooLow",
            "ValueZeroNotAllowed",
            "WriteOnly"
          ]
        },
        "additionalInfo": {
          "type": "string"
        }
      },
      "additionalProperties": false,
      "required": [
        "reasonCode"
      ]
    }
  },
  "additionalProperties": false,
  "required": [
    "status"
  ]
}
```

K General overview of the Input/Output ports

The vSECC features I/O ports which are accessible through the MQTT bus and the PEP protocol. The following tables depicts the mapping between port labels printed on the vSECC housing, the vSECC connnector pins and signals and the corresponding (and alternative) MQTT topics, PEP-CAN and PEP-WS identifiers.

K.1 vSECC overview

The available ports of the vSECC are:

- 15 Digital Out Ports
- 8 Digital In Ports
- 2 Analog In Ports
- 9 Temperature In Ports
- 3 Virtual Out and 1 Virtual In Ports for Inverted Pantograph Control

Figure 128: vSECC top view with designation of interfaces

Con	Pin	Housing label	Signal name	Signal type	MQTT topics	PEP-CAN identifier	PEP-WS identifier
X301	1	0-10V 2	AIN_10V_2	analog	vsecc/io/a_in/2/value	ain2	a2
	2	0-10V 1	AIN_10V_1	analog	vsecc/io/a_in/1/value	ain1	a1
	4	TEMP 9	AIN_TEMP_9	temp	vsecc/io/t_in/9/resistance vsecc/io/t_in/9/temperature	temperature9	tr9 t9
	7	TEMP 8	AIN_TEMP_8	temp	vsecc/io/t_in/8/resistance vsecc/io/t_in/8/temperature	temperature8	tr8 t8
	8	TEMP 7	AIN_TEMP_7	temp	vsecc/io/t_in/7/resistance vsecc/io/t_in/7/temperature	temperature7	tr7 t7
	11	TEMP 6	AIN_TEMP_6	temp	vsecc/io/t_in/6/resistance vsecc/io/t_in/6/temperature	temperature6	tr6 t6
	12	TEMP 5	AIN_TEMP_5	temp	vsecc/io/t_in/5/resistance vsecc/io/t_in/5/temperature	temperature5	tr5 t5
	15	TEMP 4	AIN_TEMP_4	temp	vsecc/io/t_in/4/resistance vsecc/io/t_in/4/temperature	temperature4	tr4 t4
	16	TEMP 3	AIN_TEMP_3	temp	vsecc/io/t_in/3/resistance vsecc/io/t_in/3/temperature	temperature3	tr3 t3
	19	TEMP 2	AIN_TEMP_2	temp	vsecc/io/t_in/2/resistance vsecc/io/t_in/2/temperature	temperature2	tr2 t2
	20	TEMP 1	AIN_TEMP_1	temp	vsecc/io/t_in/1/resistance vsecc/io/t_in/1/temperature	temperature1	tr1 t1
X306	2	CN2 STOP	DIG_IN8	digital in	vsecc/io/d_in/8/value vsecc/io/signal/connector_2_stop	din8	d8
	3	CN2 START	DIG_IN7	digital in	vsecc/io/d_in/7/value vsecc/io/signal/connector_2_start	din7	d7
	4	PANTO ERR	DIG_IN6	digital in	vsecc/io/d_in/6/value vsecc/io/signal/panto_err	din6	d6
	5	PANTO DOWN	DIG_IN5	digital in	vsecc/io/d_in/5/value vsecc/io/signal/panto_down	din5	d5
	6	PANTO UP	DIG_IN4	digital in	vsecc/io/d_in/4/value vsecc/io/signal/panto_up	din4	d4
	7	CN1 STOP	DIG_IN3	digital in	vsecc/io/d_in/3/value vsecc/io/signal/connector_1_stop	din3	d3
	8	CN1 START	DIG_IN2	digital in	vsecc/io/d_in/2/value vsecc/io/signal/connector_1_start	din2	d2
	9	DIN1	DIG_IN1	digital in	vsecc/io/d_in/1/value	din1	d1
	10	PANTO CTRL	DIG_OUT16	digital out	vsecc/io/d_out/16/value vsecc/io/signal/panto_ctrl	n.a.	n.a.
	11	OUT15	DIG_OUT15	digital out	vsecc/io/d_out/15/value	dout15	d15
	12	OUT14	DIG_OUT14	digital out	vsecc/io/d_out/14/value	dout14	d14
	13	OUT13	DIG_OUT13	digital out	vsecc/io/d_out/13/value	dout13	d13
	14	OUT12	DIG_OUT12	digital out	vsecc/io/d_out/12/value	dout12	d12
	15	OUT11	DIG_OUT11	digital out	vsecc/io/d_out/11/value	dout11	d11
	16	OUT10	DIG_OUT10	digital out	vsecc/io/d_out/10/value	dout10	d10
	17	OUT9	DIG_OUT9	digital out	vsecc/io/d_out/9/value	dout9	d9
	18	OUT8	DIG_OUT8	digital out	vsecc/io/d_out/8/value	dout8	d8
	19	OUT7	DIG_OUT7	digital out	vsecc/io/d_out/7/value	dout7	d7
	20	OUT6	DIG_OUT6	digital out	vsecc/io/d_out/6/value	dout6	d6
	21	OUT5	DIG_OUT5	digital out	vsecc/io/d_out/5/value	dout5	d5
	22	OUT4	DIG_OUT4	digital out	vsecc/io/d_out/4/value	dout4	d4
	23	OUT3	DIG_OUT3	digital out	vsecc/io/d_out/3/value	dout3	d3
	24	OUT2	DIG_OUT2	digital out	vsecc/io/d_out/2/value	dout2	d2
	25	OUT1	DIG_OUT1	digital out	vsecc/io/d_out/1/value	dout1	d1

The analog inputs support voltages between 0–10V. Return values are between 0.00 and 10.00 with up to 2 decimal points separated by a dot. The unit is *Volts*.

There are two ways to access the temperature inputs:

> If you connect a PT1000 temperature sensor, you can read the values in degrees Celsius by using the *tX* identifiers. Computed with $0.29 \cdot R - 295$.

> If you want to use another temperature sensor, you can retrieve the resistance values (in Ohms) by using the *trX* identifiers and use your own conversion function.

Caution: Digital Inputs DIG_IN2 to DIG_IN8 may already be in use when the vSECC is configured for physical stop buttons, CHAdeMO or Inverted Pantograph.

Caution: Digital Output DIG_OUT16 is not accessible via PEP identifier directly. Use the virtual port instead. Refer section 8.18.9 for more details.

K.2 vSECC.single overview

The available ports of the vSECC.single are:

- 3 Digital Out Ports
- 4 Combined Digital and Analog In Ports
- 2 Temperature In Ports

Caution: There are 4 Combined Digital and Analog In Ports available, but only 2 analog (AIN1 and AIN2) and 3 digital (DIO5, DIO6 and DIO7) of them are usable via PEP.

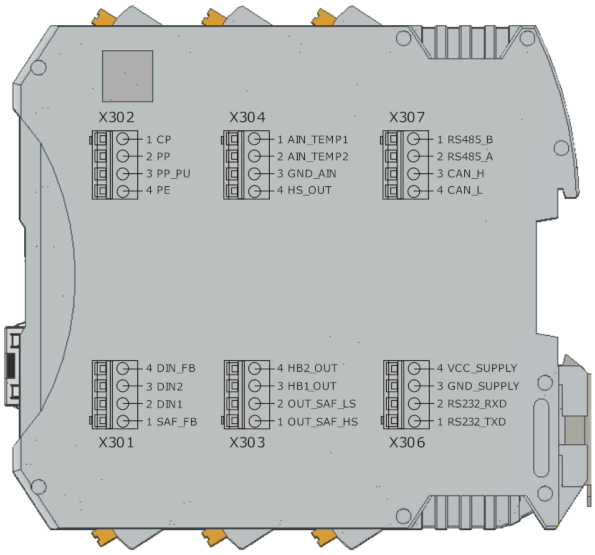


Figure 129: vSECC.single pinout printed on the side of the housing

Con	Pin	Housing label	Signal name	Signal type	MQTT topics	PEP-CAN identifier	PEP-WS identifier
X301	1	SAF_FB	DIO5 AIN4	digital in analog	vsecc/io/d_in/SAF_FB_IN/value vsecc/io/d_in/5/value vsecc/io/a_in/4/value	din5 n.a.	d5 n.a.
	2	DIN1	DIO6 AIN1	digital in analog	vsecc/io/d_in/DIN1/value vsecc/io/d_in/6/value vsecc/io/a_in/1/value	din6 ain1	d6 a1
	3	DIN2	DIO7 AIN2	digital in analog	vsecc/io/d_in/DIN2/value vsecc/io/d_in/7/value vsecc/io/a_in/2/value	din7 ain2	d7 a2
	4	DIN_FB	DIO9 AIN3	digital in analog	vsecc/io/d_in/DIN_FB/value vsecc/io/d_in/9/value vsecc/io/a_in/3/value	n.a. n.a.	n.a. n.a.
X303	3	HB1_OUT	DIO1	digital out	vsecc/io/d_out/HB1_OUT/value vsecc/io/d_out/1/value	dout1	d1
	4	HB2_OUT	DIO4	digital out	vsecc/io/d_out/HB2_OUT/value vsecc/io/d_out/4/value	dout4	d4
X304	1	AIN_TEMP1	AIN_TEMP_1	temp	vsecc/io/t_in/1/value vsecc/io/t_in/1/temperature	temperature1	tr1 t1
	2	AIN_TEMP2	AIN_TEMP_2	temp	vsecc/io/t_in/2/value vsecc/io/t_in/2/temperature	temperature2	tr2 t2
	4	HS_OUT	DIO10	digital out	vsecc/io/d_out/HS_OUT/value vsecc/io/d_out/10/value	dout10	d10

The analog inputs AIN1, AIN2 and AIN3 support voltages between 0–5V. The analog input AIN4 support voltages between 0–3.3V. Return values are between 0.00 and 5.00 with up to 2 decimal points separated by a dot. The unit is *Volts*.

There are two ways to access the temperature inputs:

> If you connect a PT1000 temperature sensor, you can read the values in degrees Celsius by using the *tX* identifiers. Computed with $0.29 \cdot R - 295$.

> If you want to use another temperature sensor, you can retrieve the resistance values (in Ohms) by using the *trX* identifiers and use your own conversion function.

Caution: Digital Input DIN_FB (DIO9) cannot be controlled via PEP-CAN or PEP-WS.

K.3 vSECC.single +70°C overview

The available ports of the vSECC.single +70°C are:

- 3 Digital Out Ports
- 4 Combined Digital and Analog In Ports
- 2 Temperature In Ports

Caution: There are 4 Combined Digital and Analog In Ports available, but only 2 analog (AIN1 and AIN2) and 3 digital (DIO5, DIO6 and DIO7) of them are usable via PEP.

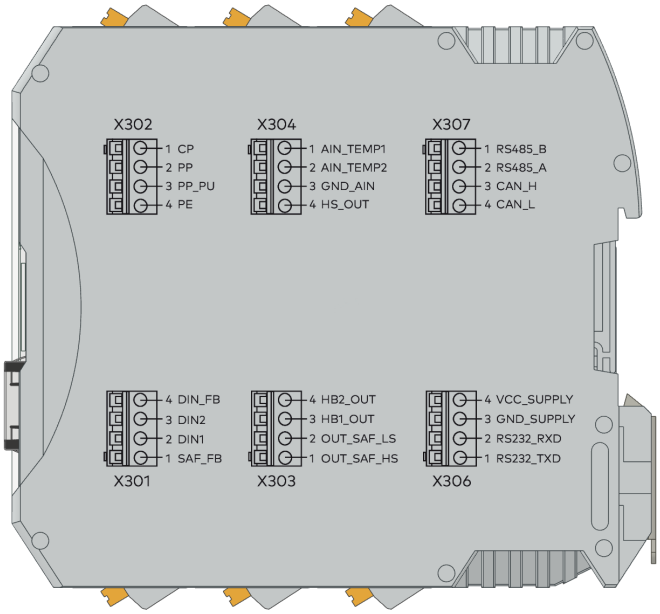


Figure 130: vSECC.single +70°C pinout printed on the side of the housing

Con	Pin	Housing label	Signal name	Signal type	MQTT topics	PEP-CAN identifier	PEP-WS identifier
X301	1	SAF_FB	DIO5 AIN4	digital in analog	vsecc/io/d_in/SAF_FB_IN/value vsecc/io/d_in/5/value vsecc/io/a_in/4/value	din5 n.a.	d5 n.a.
	2	DIN1	DIO6 AIN1	digital in analog	vsecc/io/d_in/DIN1/value vsecc/io/d_in/6/value vsecc/io/a_in/1/value	din6 ain1	d6 a1
	3	DIN2	DIO7 AIN2	digital in analog	vsecc/io/d_in/DIN2/value vsecc/io/d_in/7/value vsecc/io/a_in/2/value	din7 ain2	d7 a2
	4	DIN_FB	DIO9 AIN3	digital in analog	vsecc/io/d_in/DIN_FB/value vsecc/io/d_in/9/value vsecc/io/a_in/3/value	n.a. n.a.	n.a. n.a.
X303	3	HB1_OUT	DIO1	digital out	vsecc/io/d_out/HB1_OUT/value vsecc/io/d_out/1/value	dout1	d1
	4	HB2_OUT	DIO4	digital out	vsecc/io/d_out/HB2_OUT/value vsecc/io/d_out/4/value	dout4	d4
X304	1	AIN_TEMP1	AIN_TEMP_1	temp	vsecc/io/t_in/1/value vsecc/io/t_in/1/temperature	temperature1	tr1 t1
	2	AIN_TEMP2	AIN_TEMP_2	temp	vsecc/io/t_in/2/value vsecc/io/t_in/2/temperature	temperature2	tr2 t2
	4	HS_OUT	DIO10	digital out	vsecc/io/d_out/HS_OUT/value vsecc/io/d_out/10/value	dout10	d10

The analog inputs AIN1, AIN2 and AIN3 support voltages between 0–5V. The analog input AIN4 support voltages between 0–3.3V. Return values are between 0.00 and 5.00 with up to 2 decimal points separated by a dot. The unit is *Volts*.

There are two ways to access the temperature inputs:

> If you connect a PT1000 temperature sensor, you can read the values in degrees Celsius by using the *tX* identifiers. Computed with $0.29 \cdot R - 295$.

> If you want to use another temperature sensor, you can retrieve the resistance values (in Ohms) by using the *trX* identifiers and use your own conversion function.

Caution: Digital Input DIN_FB (DIO9) cannot be controlled via PEP-CAN or PEP-WS.

K.4 vSECC.single Board overview

The available ports of the vSECC.single Board are:

- 10 Digital Ports (which will be usable either as Out or In Ports in future releases)
- 4 Analog In Ports
- 2 Temperature In Ports

Caution: There are 10 Digital Ports available, the last 2 Digital Ports (DIO9 and DIO10) cannot be accessible via PEP, if they are configured as Digital Inputs.

Caution: There are 4 Analog Ports available, but only two of them (AIN1 and AIN2) are usable via PEP.

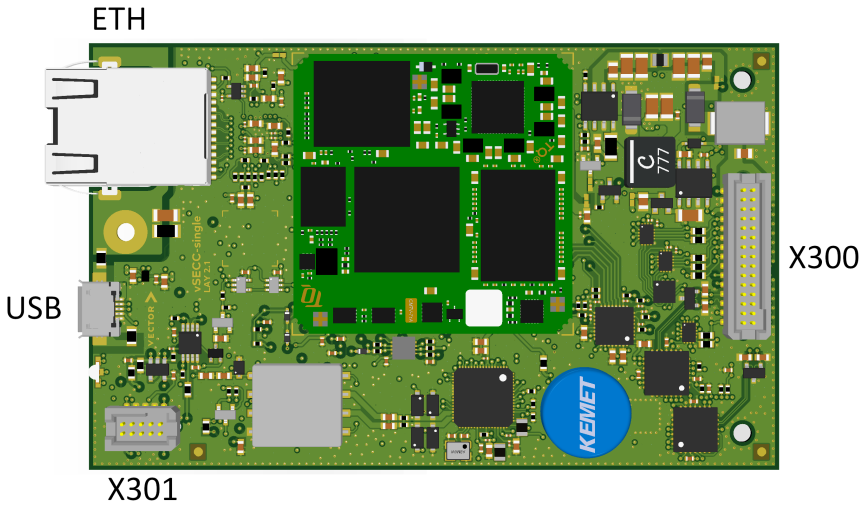


Figure 131: vSECC.single Board top view with designation of interfaces

Con	Pin	Housing label	Signal name	Signal type	MQTT topics	PEP-CAN identifier	PEP-WS identifier
X300	6	-	DIO1	digital out	vsecc/io/d_out/HB1_OUT/value vsecc/io/d_out/1/value	dout1	d1
	9	-	DIO4	digital out	vsecc/io/d_out/HB2_OUT/value vsecc/io/d_out/4/value	dout4	d4
	10	-	DIO5	digital in	vsecc/io/d_in/SAF_FB_IN/value vsecc/io/d_in/5/value	din5	d5
	11	-	DIO6	digital in	vsecc/io/d_in/DIN1/value vsecc/io/d_in/6/value	din6	d6
	12	-	DIO7	digital in	vsecc/io/d_in/DIN2/value vsecc/io/d_in/7/value	din7	d7
	14	-	DIO9	digital in	vsecc/io/d_in/DIN_FB/value vsecc/io/d_in/9/value	n.a.	n.a.
	15	-	DIO10	digital out	vsecc/io/d_out/HS_OUT/value vsecc/io/d_out/10/value	dout10	d10
	18	-	AIN1	analog	vsecc/io/a_in/1/value	ain1	a1
	19	-	AIN2	analog	vsecc/io/a_in/2/value	ain2	a2
	20	-	AIN3	analog	vsecc/io/a_in/3/value	n.a.	n.a.
	21	-	AIN4	analog	vsecc/io/a_in/4/value	n.a.	n.a.
X301	9	-	AIN_TEMP_1	temp	vsecc/io/t_in/1/value vsecc/io/t_in/1/temperature	temperature1	tr1 t1
	10	-	AIN_TEMP_2	temp	vsecc/io/t_in/2/value vsecc/io/t_in/2/temperature	temperature2	tr2 t2

The analog inputs support voltages between 0–5V. Return values are between 0.00 and 5.00 with up to 2 decimal points separated by a dot. The unit is *Volts*.

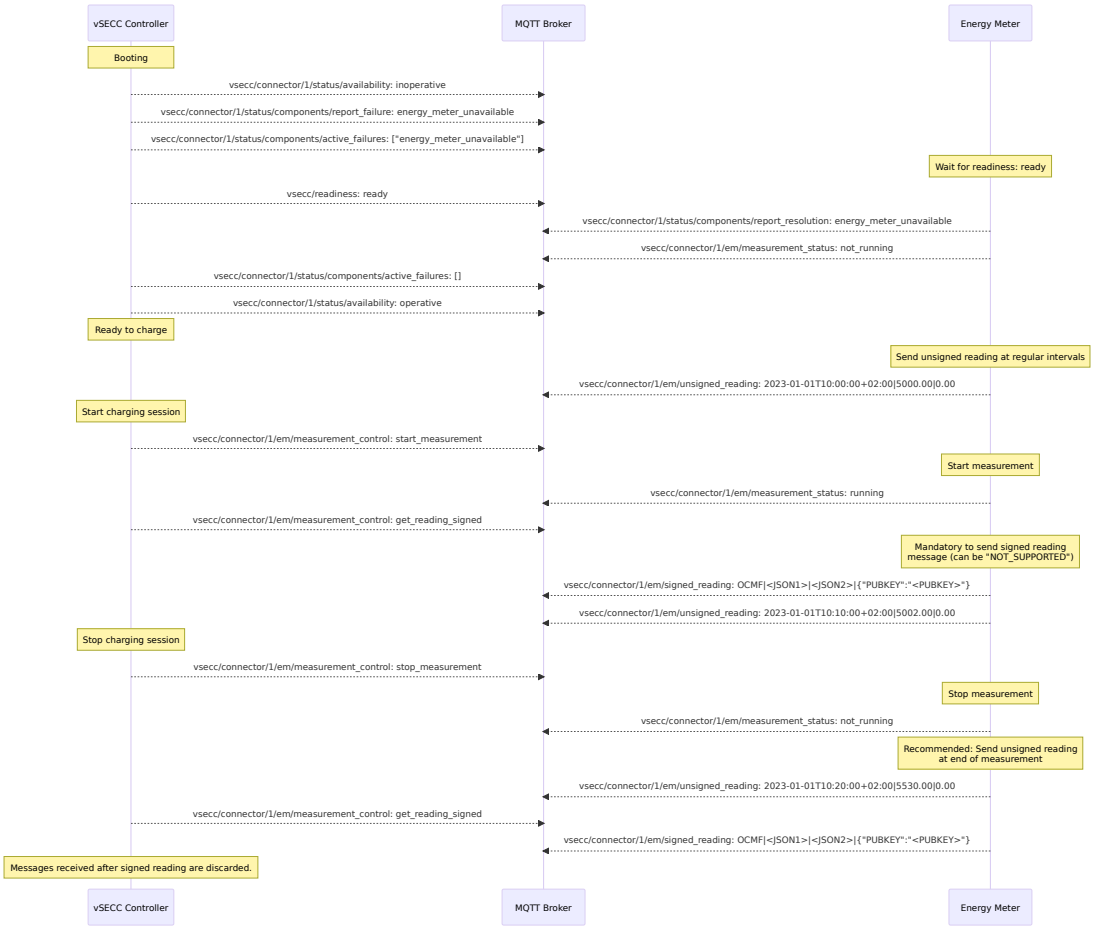
There are two ways to access the temperature inputs:

> If you connect a PT1000 temperature sensor, you can read the values in degrees Celsius by using the *tX* identifiers. Computed with $0.29 \cdot R - 295$.

> If you want to use another temperature sensor, you can retrieve the resistance values (in Ohms) by using the *trX* identifiers and use your own conversion function.

Caution: Digital Input DIO9 cannot be controlled via PEP-CAN or PEP-WS.

L MQTT Energy Meter Sequence



M DataTransfer via MQTT

M.1 Sequences

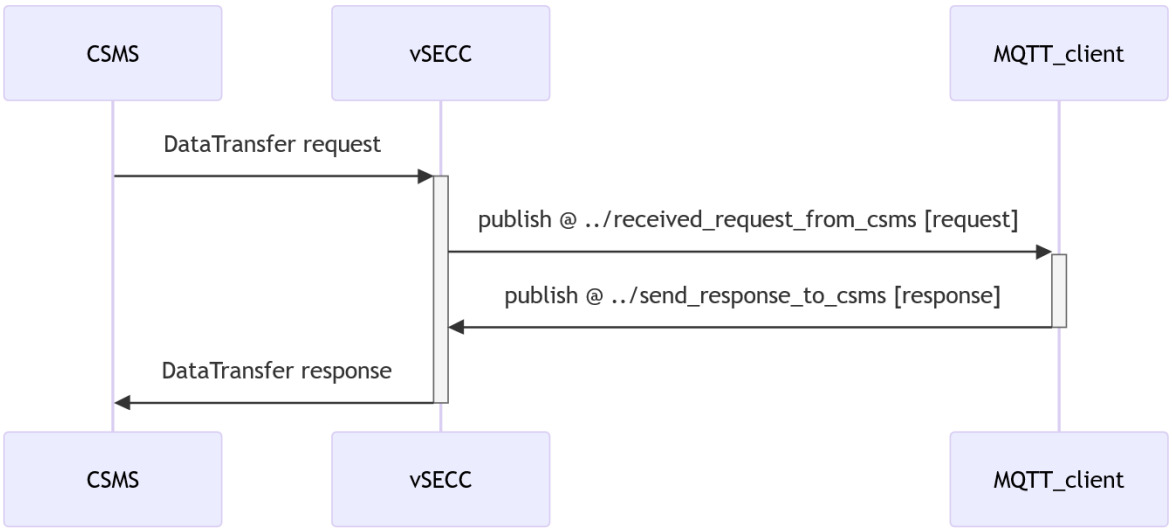


Figure 132: Goodcase: A DataTransfer request from the CSMS is received, a response from a custom MQTT client is transmitted to the CSMS.

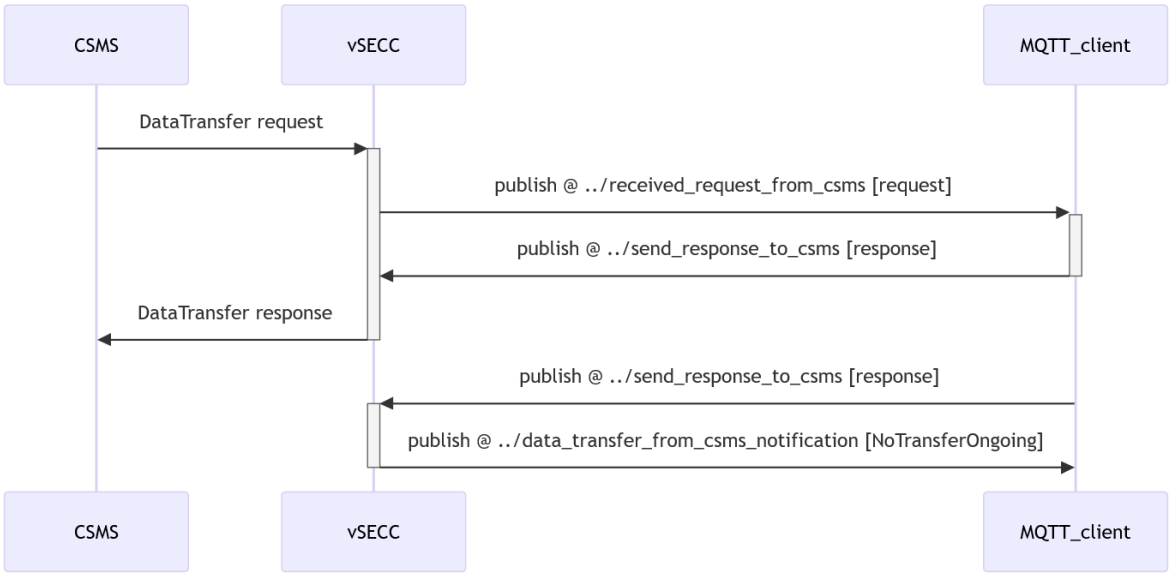


Figure 133: A DataTransfer request from the CSMS is received, a response from a custom MQTT client is transmitted to the CSMS. The second response from the MQTT client is unexpected, since there is no request to answer.

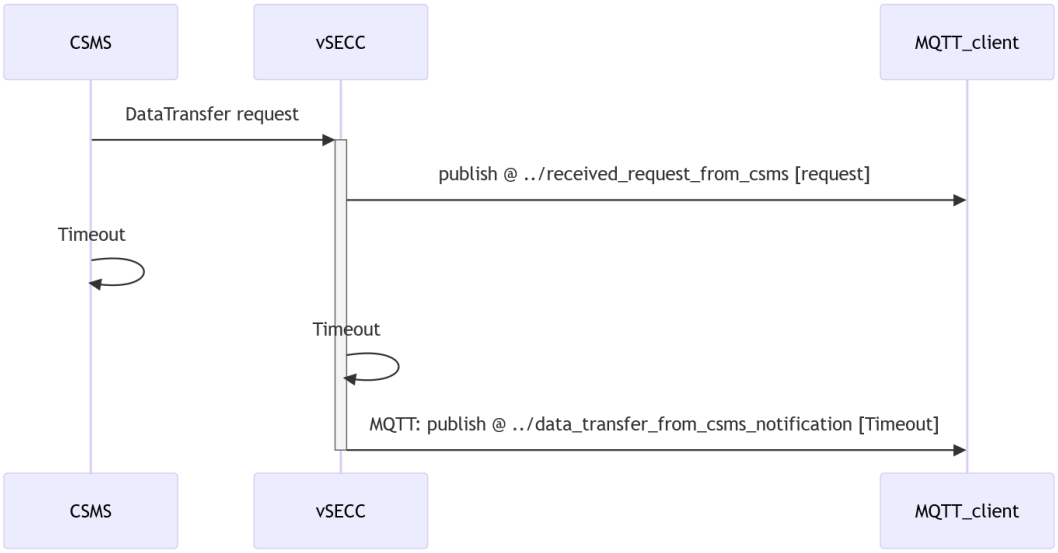


Figure 134: A DataTransfer request from the CSMS is received, but there is no response from a custom MQTT client in time.

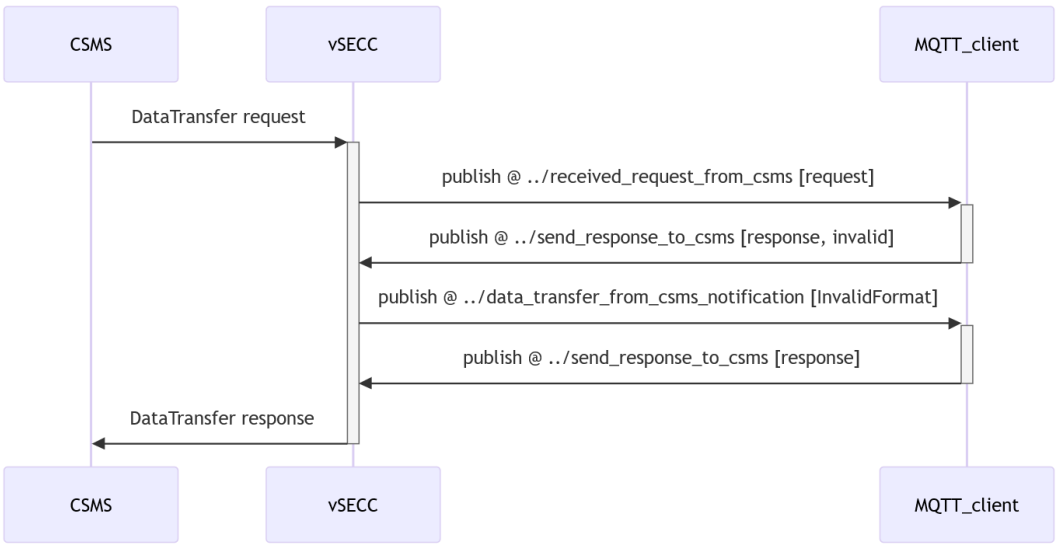


Figure 135: A DataTransfer request from the CSMS is received, but the response from a custom MQTT client does not match the DataTransfer response JSON format. The second response attempt of the MQTT client is a valid response that gets sent to the CSMS.

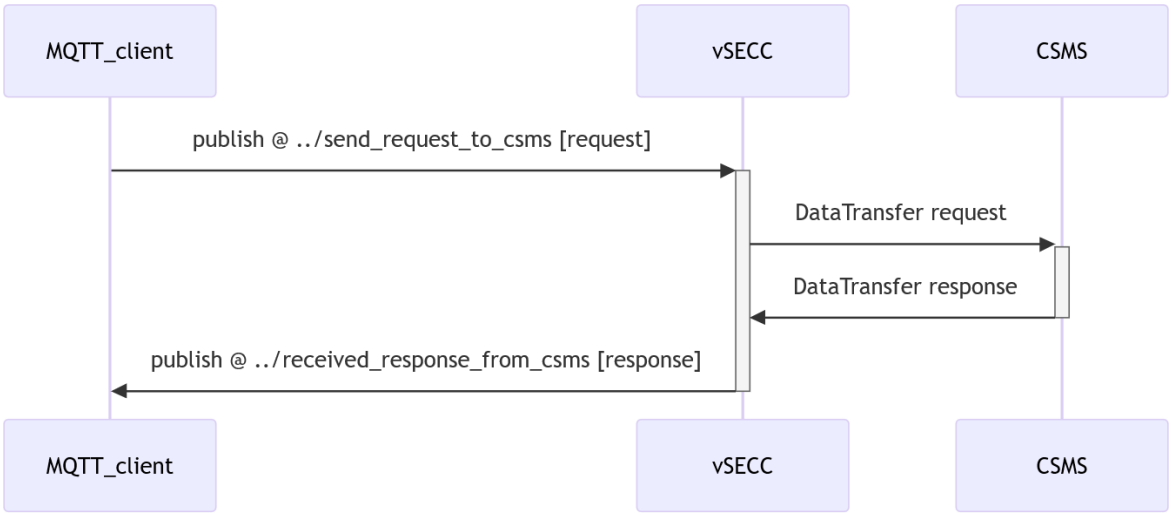


Figure 136: Goodcase: A DataTransfer request from an MQTT client is transmitted to the CSMS, the response from the CSMS is forwarded to MQTT.

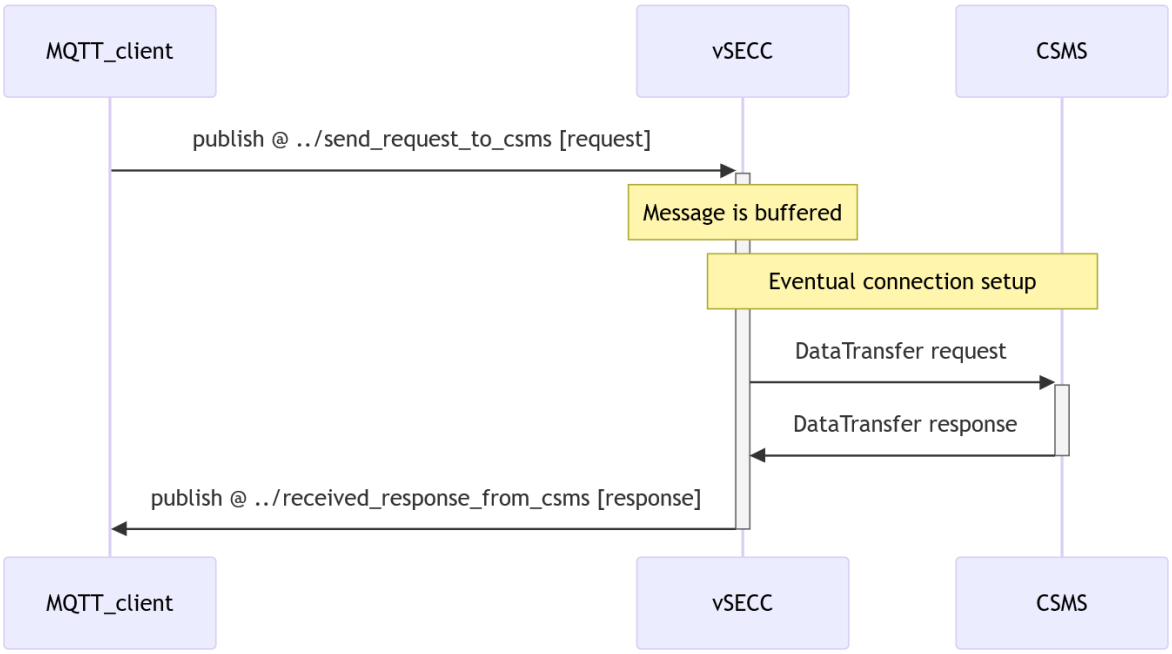


Figure 137: Initially no connection to the CSMS was established, so a DataTransfer request from an MQTT client is buffered. Once the connection to the CSMS is established, the request is transmitted to the CSMS. The response from the CSMS is forwarded to MQTT.

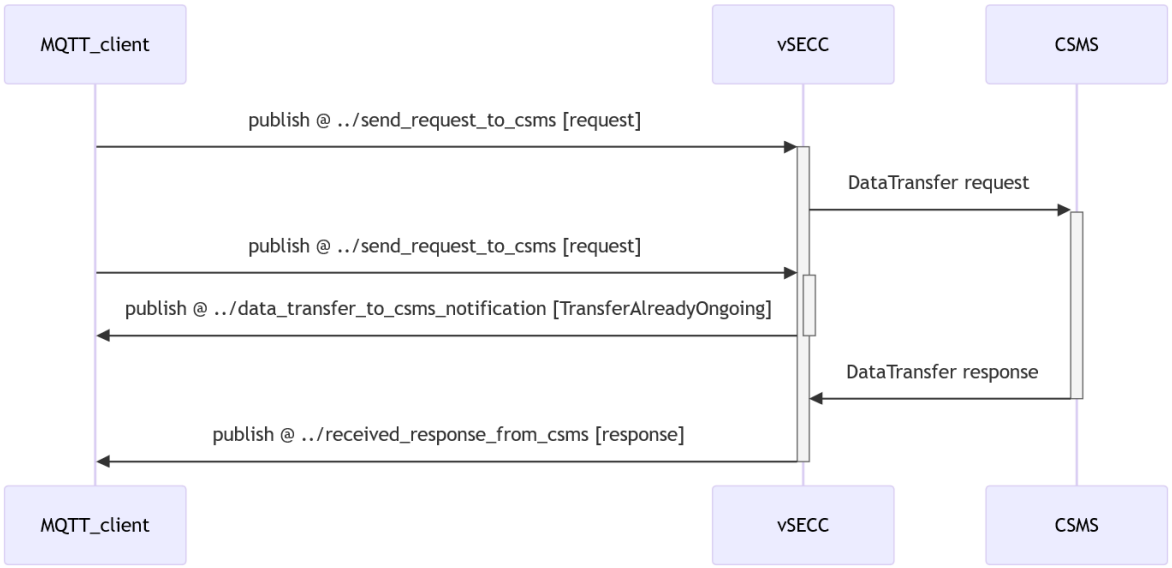


Figure 138: A DataTransfer request is published by an MQTT client for transmission to the CSMS, but the response for the last DataTransfer request still has to be received.

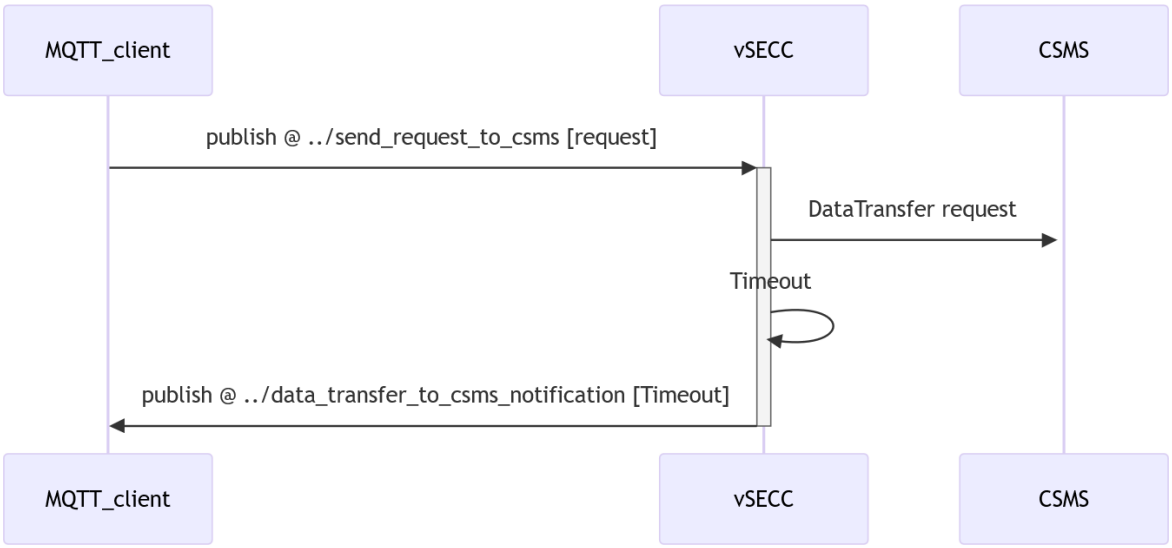


Figure 139: A DataTransfer request from an MQTT client is sent to the CSMS, but the CSMS does not respond in time.

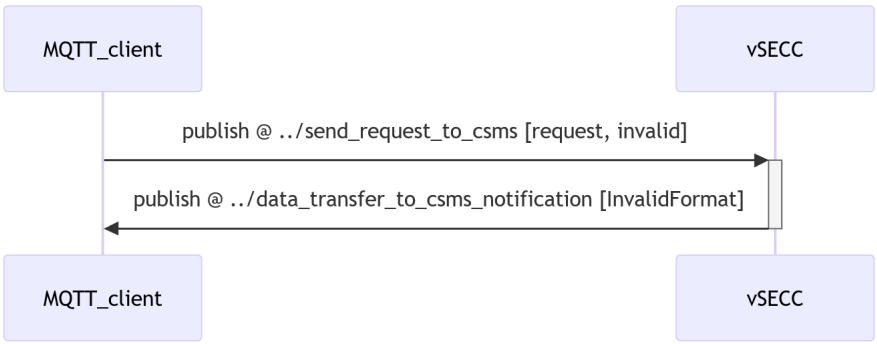


Figure 140: The DataTransfer request from an MQTT client does not match the DataTransfer request JSON schema.

N JSON schemas

A notification regarding an invalid or overdue DataTransfer response message to the CSMS has the following format:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "DataTransferFromCsmsNotification",
  "type": "object",
  "properties": {
    "info": {
      "type": "string",
      "enum": ["Timeout", "NoTransferOngoing", "InvalidFormat"]
    }
   },
  "additionalProperties": false,
  "required": ["info"]
}
```

A notification regarding an invalid DataTransfer request message to the CSMS or an overdue response from the CSMS has the following format:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "DataTransferToCsmsNotification",
  "type": "object",
  "properties": {
    "info": {
      "type": "string",
      "enum": ["Timeout", "TransferAlreadyOngoing", "InvalidFormat"]
    }
  },
  "additionalProperties": false,
  "required": ["info"]
}
```

A DataTransfer response received or to be sent has the following format:

```
{ "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "DataTransferResponse",
  "type": "object",
  "properties": {
    "status": {
       "type": "string",
       "enum": ["Accepted", "Rejected", "UnknownMessageId",
       "UnknownVendorId"]
    },
    "data": {},
    "statusInfo": {
      "type": "object",
      "properties": {
        "reasonCode": {
          "type": "string",
          "enum": ["CsNotAccepted", "DuplicateProfile",
            "DuplicateRequestId", "FixedCable", "FwUpdateInProgress",
            "InternalError", "InvalidCertificate", "InvalidCsr",
            "InvalidIdToken", "InvalidMessageSequence",
            "InvalidProfile", "InvalidSchedule",
            "InvalidStackLevel", "InvalidUrl", "InvalidValue",
            "MissingParam", "NoCable", "NoError", "NotEnabled",
            "NotFound", "OutOfMemory", "OutOfStorage",
            "ReadOnly", "TooLargeElement", "TooManyElements",
            "TxInProgress", "TxNotFound", "TxStarted",
            "UnknownConnectorId", "UnknownConnectorType",
            "UnknownEvse", "UnknownTxId", "Unspecified",
            "UnsupportedParam", "UnsupportedRateUnit",
            "UnsupportedRequest", "ValueOutOfRange",
            "ValuePositiveOnly", "ValueTooHigh", "ValueTooLow",
            "ValueZeroNotAllowed", "WriteOnly"]
        },
        "additionalInfo": {
          "type": "string"
        }
      },
      "additionalProperties": false,
      "required": ["reasonCode"]
    }
  },
  "additionalProperties": false,
  "required": ["status"]
}
```

A DataTransfer request received or to be sent has the following format:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "DataTransferRequest",
  "type": "object",
  "properties": {
    "vendorId": {
      "type": "string"
    },
    "messageId": {
      "type": "string"
    },
    "data": {}
  },
  "additionalProperties": false,
  "required": ["vendorId"]
}
```

O Additional MQTT topic information

O.1 EV_Communication_State

The following table provides an overview of the events during the charging mode at which the topic `vsecc/connector/evse_id/ocpp/ev_communication_state` will be published.

EV_Comm_State	CCS conducted charging	Pantograph charging	CHAdeMO
waiting_for_ communication	CP_State_A No_Link(Pause)	Charging session termination CP_State_A	state_a_ ev_disconnected
establish_ communication	SLAC_Parm Request	SECC_Discovery Request	state_b_ ev_connected
establish_ communication_ error	SLAC error	SECC_Discovery error	charging_abort_ state_c_f1
communication	Link_Up	Supported App Protocol	state_c_ start_charger_can
communication_ error	V2G communication error	V2G communication error	charging_abort_ state_c_f2/-f3_1 disable_ d1_d2_d3_f3_first

O.2 PP_State

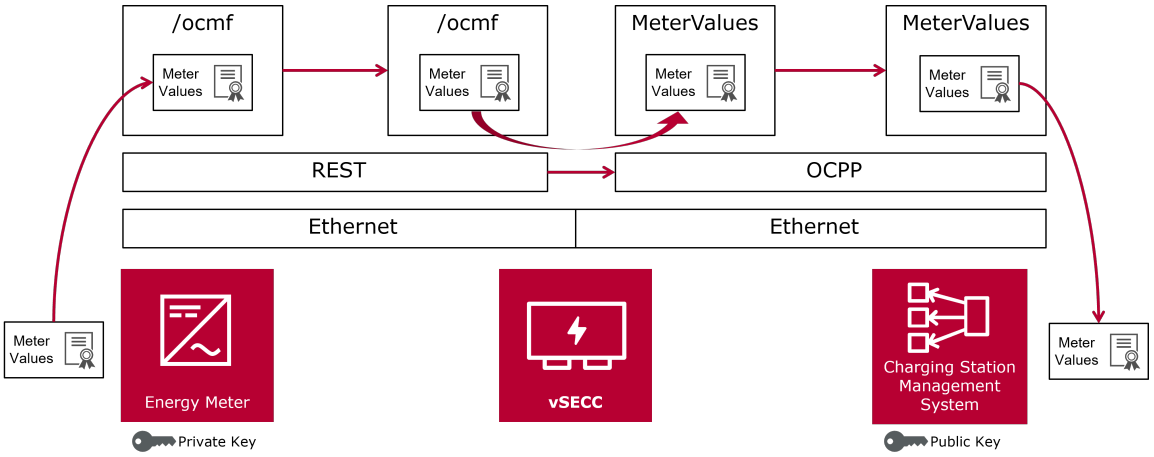
The following table provides an overview of the voltage ranges at which different `pp_states` will be published to the topic `vsecc/connector/evse_id/ev/pp_state`. An update to the `pp_state` will be published whenever the state changes.

PP_State	Voltage of ProximityPin in range
disconnected	-0.1V <= PP_Voltage <= 0.1V
s3_open	2.36V <= PP_Voltage <= 3.16V
s3_closed	1.23V <= PP_Voltage <= 1.82V
faulted	out of range

P Documentation for Eichrecht certification

P.1 Communication between the vSECC and the measuring capsule (”Schalt-Mess-Koordination”)

As the vSECC is not part of the measuring capsule, it interfaces to the energy meter via TCP/IP connection and a REST interface of the LEM energy meter. The back end connection is via OCPP.



A charging sequence compliant to Eichrecht is described in the following sequence diagram:

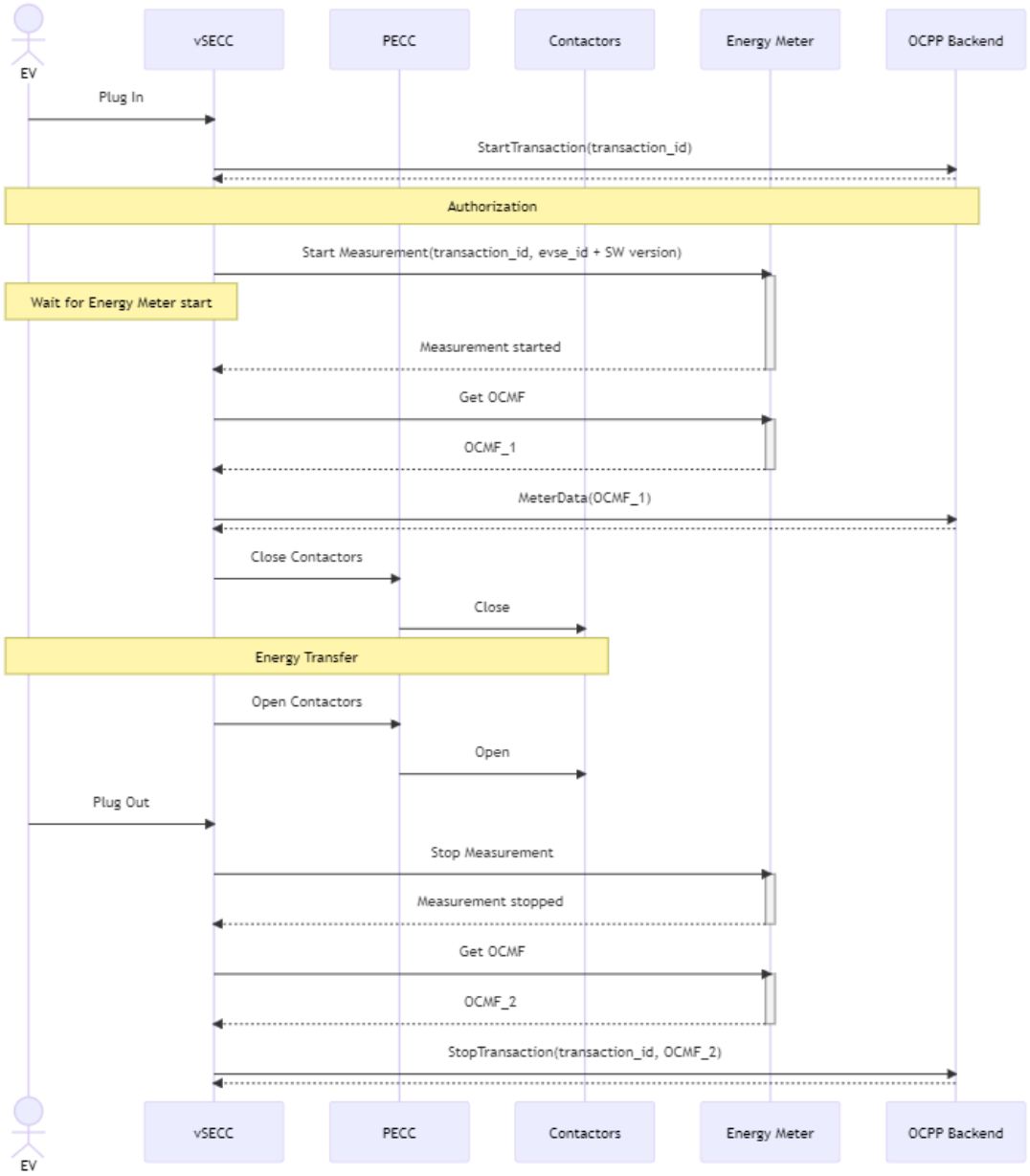


Figure 141: Schalt-Mess-Koordination

In case the OCPP connection is lost during a transaction, the OCMF readings are stored in the vSECC and retransmitted after the connection is restored.
The pagination counter of the LEM meter is part of the legal fields and increments after each read. A signed OCMF reading is requested at the start and end of each transaction.

P.2 Identification of the charge controller software

The vSECC Firmware version is sent to the Energy Meter in the ”evseId” field as part of the start command, as specified in the LEM document ”HOW to FULFILL OCMF FIELDS from 28/11/2022”. An example for a start command: `{"evseId":"49*564543*01, v2.7.5", "transactionId":"e5c8eafe-af51-4790-83e1-cfdd69d64ebd", "clientId": "020000000001", "tariffId":0, "cableId":0, "userData":""}`

Q Glossary

Autocharge Procedure to authenticate and to authorize a vehicle automatically at a charging station. The EVCC ID of the vehicle is used as identifier. The Combined Charging System (CCS) standard is required, since the EVCC ID is exchanged via V2G communication (DIN SPEC 70121 or ISO 15118). The recommended integration with OCPP is described in the Whitepaper “WhitePaper Identification of Electric Vehicles in Charging Station Management System via OCPP” published by Vector, which can be found in the Downloads Section.

Certificate Authority In cryptography, a certificate authority (CA) is an entity that issues digital certificates. A CA acts as a trusted third party, which is trusted both by the subject (owner) of the certificate and by the party relying upon the certificate. A CA is required e.g. for TLS and PnC Certificates.

Charging Station The term charging station describes a physical system where an EV can be charged. Each vSECC Controller corresponds to one charging station.

CHAdeMO is a DC charging standard for electric vehicles. It enables seamless communication between the car and the charger via CAN communication. Since the standard was developed in Japan, it is applied mainly by Japanese and North-American car manufacturers.

Combined Charging System is an open, universal and international charging system for electric vehicles based on international standards. The CCS combines single-phase with fast 3-phase AC charging using alternating current of maximum of 43 kW. It also provides very fast high-power DC charging within a single system. The CCS system includes the connector, the managing of control functions and the charging communication between electric vehicle and infrastructure over Powerline Communication.

Connector The term connector describes an electrical outlet on a charging station. It is connected to a single EVSE. An EVSE can have multiple connectors attached to it, e.g. one CCS and one CHAdeMO compliant outlet. However, an EVSE will always use only one of its connectors exclusively.

Control Pilot See chapter 2.2.6 for more information.

External Identification Means Any external means that enable the user to identify, authenticate and authorize his contract or the EV for a charging session at the charging station, e.g. an RFID card.

Electric Vehicle Supply Equipment is defined by its ability to deliver energy to one EV at a time. A charging station can be connected to one or more EVSEs.

GB/T The Guobiao standard 27930 for AC and DC charging was developed for charging of Chinese EVs. As CHAdeMO, the communication takes place via CAN.

High Level Communication is specified in the ISO 15118 series as a bi-directional digital communication using protocols, messages and physical and data link layers.

Load Leveling enables the prevention of overloading the charging infrastructure by calculating the maximum power that is distributed from the charging stations to the vehicles.

Megawatt Charging System The Megawatt Charging System is a charging connector for large battery electric vehicles. The connector will be rated for charging at a maximum rate of 3.75 megawatts (3000 amps at 1250 volts DC).

Plug and Charge Identification mode where the customer just has to plug his electric vehicle into the EVSE and all aspects of authentication, authorization, load control and billing are automatically taken care of with no further intervention from the customer.

Smart Charging The term smart charging is used for charging systems of electric or hybrid vehicles according to ISO 15118, DIN SPEC 70121, SAE J2847/2. The communication between vehicle and charging station is realized in two ways:
1) As powerline communication via the control pilot pin in the form of a PWM signal and a digital signal for HomePlug-GreenPhy standard.
2) Wireless in case of inductive charging.

Value Added Services allow additional information, which is not directly needed for the pure charging of the EV, to be exchanged via separate communication channels such as HTTP, HTTPS, FTP. The to-date most prevalent VAS is the Preconditioning of buses, which is standardized by VDV261.

R Abbreviations

AC	Alternating Current
CA	Certificate Authority
CCS	Combined Charging System
CP	Control Pilot
CSMS	Charging Station Management System
DC	Direct Current
ECU	Electronic Control Unit
EIM	External Identification Means
EV	Electric Vehicle
EVSE	Electric Vehicle Supply Equipment
GUI	Graphical User Interface
HVDC	High Voltage Direct Current
HMI	Human Machine Interface
MCS	Megawatt Charging System
OCPP	Open Charge Point Protocol
PE	Protective Earth
PE(P)	Power Electronics (Protocol)
PECC	Power Electronics Communication Controller
PLC	Power Line Communication
PnC	Plug and Charge
PP	Proximity Pin
SECC	Supply Equipment Charge Controller
TLS	Transport Layer Security
UI	User Interface
URI	Uniform Resource Identifier
VAS	Value Added Service

VECTOR

Get More Information

Visit our website for:

- News
- Products
- Demo software
- Support
- Training classes
- Addresses

www.vector.com