
home-assistant-ocpp

unknown

May 16, 2026

CONTENTS

1	Installation	3
1.1	Install HACS	3
1.2	Install the OCPP Repository	3
1.3	Add the OCPP Integration	3
1.4	Configure the Central System	4
1.4.1	Host address and port	4
1.4.2	Secure Connection	4
1.4.3	Measurands	4
1.5	Add the entities to your Dashboard	4
1.6	Configure your Charger	5
1.7	Start Charging	6
2	Supported devices	9
2.1	ABB Terra AC chargers	9
2.1.1	Issue Description	9
2.1.2	Workaround	9
2.1.3	ABB Terra AC-W7-G5-R-0	10
2.1.4	ABB Terra AC-W11-G5-R-0	10
2.1.5	ABB Terra AC-W22-T-0	10
2.1.6	ABB Terra TAC-W22-T-RD-MC-0	10
2.2	Alfen - Eve Single Pro-line	10
2.3	Alfen - Eve Single S-line	10
2.4	CTEK Chargestorm Connected 1, dual connectors	10
2.5	CTEK Chargestorm Connected 2	10
2.6	EN+ Caro Series Home Wallbox	10
2.7	Etrek - Inch Pro	11
2.8	EVBox Elvi	11
2.9	EVLink Wallbox Plus	11
2.10	Evnex E Series & X Series Charging Stations	11
2.11	Garo Entity Pro	12
2.12	MaXpeedingrods Ev Charger	12
2.13	Mennekes Amtron Charge Control	12
2.14	Morek Smart AC Charger	12
2.15	Rolec EVO	12
2.16	Simpson & Partners	12
2.17	SyncEV Compact EVCP	12
2.18	Teison Smart MINI Wallbox	13
2.19	United Chargers Inc. - Grizzl-E	13
2.20	V2C Trydan	14
2.21	Vestel EVC04-AC22SW	14

2.22	Wallbox Pulsar & Copper SB	14
2.23	ZJ Beny BCP-A2N-P	14
2.24	Others	14
3	User guide	15
3.1	Installing the OCPP Integration	15
3.2	Installing HACS (Home Assistant Community Store)	15
3.3	Configuring the Central System	16
3.4	Understanding status	17
3.5	Changing availability	18
3.6	Useful Entities for Wallbox Pulsar Plus	18
3.6.1	Metrics	18
3.6.2	Diagnostics	19
3.6.3	Controls	19
3.7	Useful Entities for ABB Terra AC	19
3.7.1	Metrics	19
3.8	Useful Entities for EVBox Elvi	19
3.8.1	Metrics	19
3.8.2	Diagnostics	19
3.8.3	Controls	20
3.9	Useful Entities and Workarounds for United Chargers Grizzl-E	20
3.9.1	Metrics	20
3.9.2	Diagnostics	20
3.9.3	Controls	20
3.10	Useful Entities for Vestel EVC-04 Wallboxes	21
3.10.1	Metrics	21
3.10.2	Diagnostics	21
3.10.3	Controls	21
3.11	Useful Entities for Roolec EVO	21
3.11.1	Metrics	21
3.11.2	Diagnostics	22
3.11.3	Controls	22
3.12	OCPP Compatibility Issues	22
3.12.1	ABB Terra AC	22
3.12.2	Grizzl-E	22
4	Support	23
4.1	General	23
4.2	FAQ	23
4.2.1	too many notifications in home assistant	23
4.2.2	unstable behavior when setting the charger maximum current	24
5	Debugging	25
6	Development	29



This is a Home Assistant integration for Electric Vehicle chargers that support the Open Charge Point Protocol.

INSTALLATION

1.1 Install HACS

- If you have not yet installed HACS, go get it at <https://hacs.xyz> and walk through the installation and configuration.

1.2 Install the OCPP Repository

- In Home Assistant, select HACS / Integrations / + Explore & add repositories.



+ EXPLORE & ADD REPOSITORIES

- Search for 'OCPP' and install the repository.

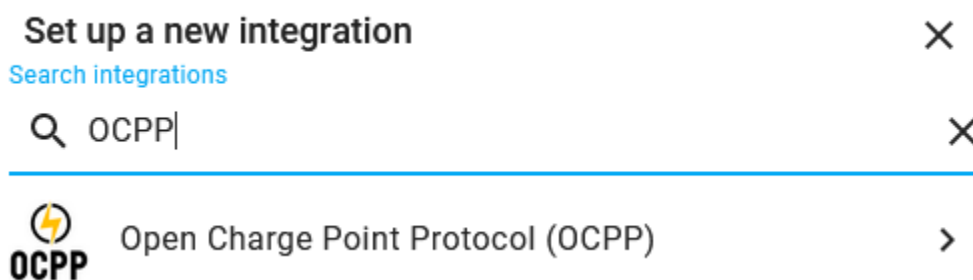
1.3 Add the OCPP Integration

- In Home Assistant, select Configuration / Integrations / Add Integration.



+ ADD INTEGRATION

- Search for 'OCPP' and add the integration.



1.4 Configure the Central System

1.4.1 Host address and port

- The default host address '0.0.0.0' will listen to all interfaces on your home assistant server.
- The default port number is 9000 but can be changed for your needs.

1.4.2 Secure Connection

If you are using [Let's Encrypt](#), [Duck DNS](#) or other add-on that enables secure HTTPS for your Home Assistant instance, you can get a secure WSS connection for OCPP. To use a secure connection:

- Enable the option *Secure connection*
- Provide the pathways to your HA's SSL certificate and key files. These are typically located in the /config or /ssl folder, and typically named fullchain.pem and privkey.pem respectively.
- If you provide incorrect pathways, the integration will fail to setup with no clear indication of why.

If you do not use HTTPS for your Home Assistant instance:

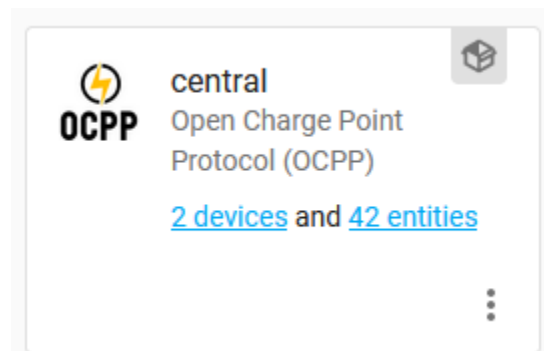
- Disable the option *Secure connection*
- *Path to SSL certificate/key* will be ignored.

1.4.3 Measurands

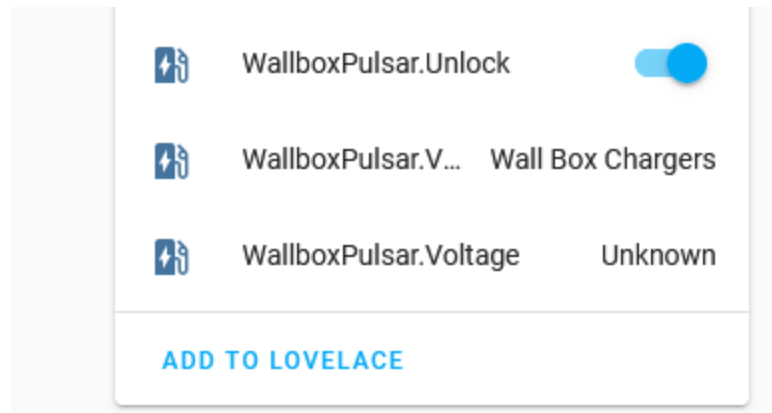
- Most chargers only support a subset of all possible measurands. This depends most on the Feature profiles that are supported by the charger.
- The integration will autodetect the supported measurands when the charger connects. This can be disabled for chargers that do not support autodetection.

1.5 Add the entities to your Dashboard

- On the OCPP integration, click on devices to navigate to your Charge Point device.



- At the bottom of the Entities panel, click on 'Add to Lovelace' to add the entities to your dashboard.



- An entity will have the value 'Unavailable' until the charger successfully connects.
- An entity will have the value 'Unknown' until its value has been read from the charger.

1.6 Configure your Charger

- Configure your charger to use the OCPP websocket of your Central System (e.g. `ws://homeassistant.local:9000`). This is charger specific, so consult your manual.
- Some chargers require the protocol section '`ws://`' to be removed, or require the url to end with a '`'`'.
- If you have configured *Secure connection* in previous step, you should use '`wss://`'
- Some chargers require the url to be specified as an IP address, i.e. '`192.168.178.1:9000`'
- You may need to reboot your charger before the changes become effective.
- If your charger has multiple connectors, wait until the charger device has populated its data (e.g., the Connectors sensor). Then reload the integration. A separate device will be created for each connector, and you'll find all connector-specific entities there.

Device detail OCPP

WEBSOCKET CONNECTION

url
ws://homeassistant.local:9000

charge point identity
WallboxPulsar

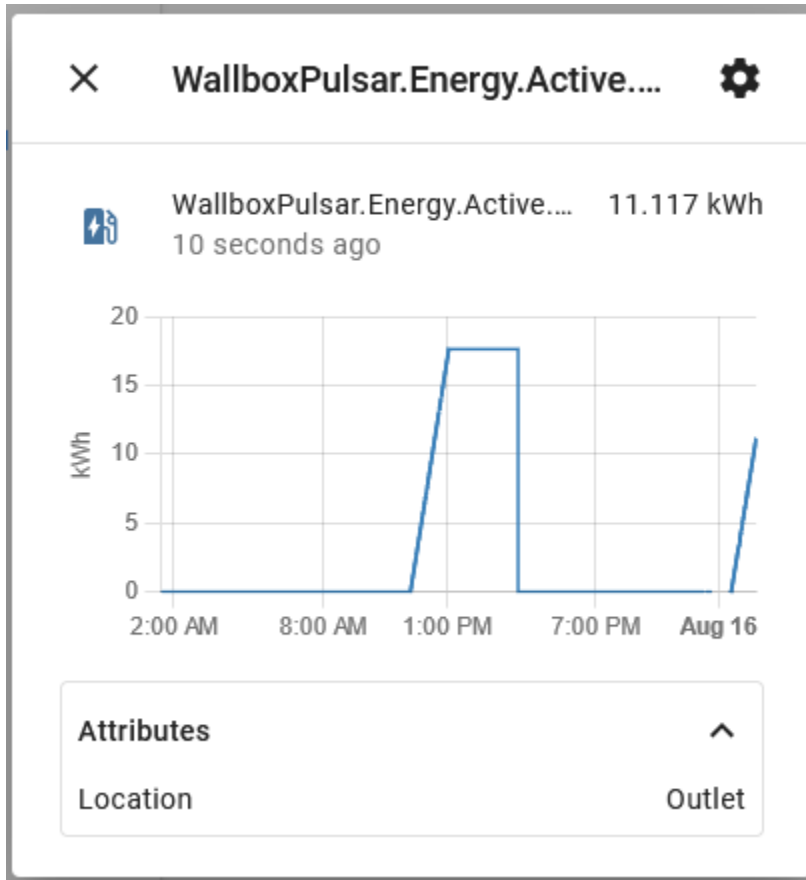
Password

I have read and accepted the [Terms and Conditions](#)

Save

1.7 Start Charging

- Use the charge control switch to start the charging process.



SUPPORTED DEVICES

All OCPP 1.6j compatible devices should be supported, but not every device offers the same level of functionality. So far, we've tried:

2.1 ABB Terra AC chargers

ABB Terra AC chargers with firmware version 1.8.21 and earlier fail to respond correctly when OCPP measurands are automatically detected by the OCPP integration. As of this writing, ABB has been notified, but no corresponding firmware fix is available.

2.1.1 Issue Description

When automatic measurand detection is used in the OCPP integration with ABB Terra AC chargers:

1. The charger responds as if it supports all proposed measurands.
2. The integration then asks for all measurands to be reported.
3. When the integration tries to query which measurands are available after this configuration, the ABB Terra AC reboots.

As a result, the ABB charger becomes unusable with the OCPP integration, as the integration checks for available measurands on every charger boot, leading to a boot loop.

For more details and symptoms, see [Issue #1275](#).

2.1.2 Workaround

Fortunately, it is possible to configure the charger using manual configuration and to restore correct settings.

To use these chargers:

1. Disable "Automatic detection of OCPP Measurands".
 - Note: Automatic detection is enabled by default. Until configuration changes can be made online, you may need to remove the devices from this integration and add them again.
 - If "Automatic detection of OCPP Measurands" is disabled during configuration, you will be presented with a list of possible measurands.
2. When presented with the list of measurands, select only the following:
 - `Current.Import`
 - `Current.Offered`

- `Energy.Active.Import.Register`
- `Power.Active.Import`
- `Voltage`

This list is based on the overview of OCPP 1.6 implementation for ABB Terra AC (firmware 1.6.6).

2.1.3 ABB Terra AC-W7-G5-R-0

2.1.4 ABB Terra AC-W11-G5-R-0

2.1.5 ABB Terra AC-W22-T-0

2.1.6 ABB Terra TAC-W22-T-RD-MC-0

2.2 Alfen - Eve Single Pro-line

2.3 Alfen - Eve Single S-line

2.4 CTEK Chargestorm Connected 1, dual connectors

See CTEK Chargestorm Connected 2 below for getting started instructions.

2.5 CTEK Chargestorm Connected 2

Jonas Karlsson has written a [getting started guide](#) for connecting CTEK Chargestorm Connected 2.

2.6 EN+ Caro Series Home Wallbox

This charger is often white-labelled by other vendors, including [cord](#) and [EV Switch](#).

Note the charger's serial number - this is the number that you need to specify for the `Charge point identity` when you configure the OCPP integration in Home Assistant if the OCPP integration does not discover your charger, and also to request a firmware update for versions earlier than 1.0.25.130.

For firmware versions earlier than 1.0.25.130 the only way you can update firmware is by connecting to the evcharge OCPP server at `wss://ocpp16.evcharge.com:33033/` and emailing your serial number to `support@en-plus.com.cn` requesting that your firmware is updated.

You will probably want to update your firmware if it is earlier than 1.0.25.130 before configuring your charger to connect to your own OCPP server.

Firmware 1.0.25.130 has a firmware update option on the configuration interface (on IP address 192.168.4.1) which you can access by power-cycling the charger and connecting to its access point (see below).

If you have already installed the OCPP integration and have the default charger charge point installed, then you will need to re-configure this with the correct charge point identity (by removing and re-adding the OCPP integration) to change from the default charger charge point identity before configuring the charger.

Connect to the charger's access point (AP) by powering down the charger (i.e. switch off the charger's isolator or circuit breaker) and powering it back on a few seconds later. The charger's access point becomes available for 15 minutes, and the SSID matches the charger's serial number (starting with SN). Log in to the configuration interface on the IP address 192.168.4.1.

If doing this from a phone, you may need to set the phone to *Flight Mode* first, and enable WiFi if required to enable the rest of the configuration to complete.

The username and password for the web interface are provided in the charger manual (case sensitive).

Configure the network mode to WiFi or Ethernet, and (in the field with an icon that looks like a router with three aerials) enter the address of your Home Assistant server including the port and protocol (e.g. `ws://myhomeassistant.tld:9000` or `wss://myhomeassistant.tld:9000` if you are using secure sockets).

The charger user interface will append the serial number when you leave the field - this is correct and expected.

Save, and the charger will reboot.

Reconnect to the charger's SSID, and log in again to 192.168.4.1 to confirm that the Network Status is online. This confirms that the charger has an internet connection via Ethernet or WiFi and is connected to your OCPP server in Home Assistant. Once enabled, the charger doesn't connect to the vendor server anymore and can be controlled only from Home Assistant or locally via Bluetooth.

Even though the device accepts all measurands, the key working ones are

- `Current.Import`
- `Current.Offered`
- `Energy.Active.Import.Register`
- `Voltage` - although this shows a constant voltage or zero unless a charging session is in progress.
- `Transaction.ID`

You may wish to disable sensors that show `Unknown` after you've completed a charging session, as they will never provide data with the current firmware 1.0.25.130.

2.7 Etrek - Inch Pro

To allow a custom OCPP server such as HA to set up a transaction ID, it is necessary to set under Users > Charging Authorization the authorization type to either `Central system only` or `Charger whitelist and central system` otherwise the OCPP integration won't match transactions and it won't report some meter values such as session time.

2.8 EVBox Elvi

2.9 EVLink Wallbox Plus

2.10 Evnex E Series & X Series Charging Stations

(Ability to configure a custom OCPP server such as HA is being discontinued)

2.11 Garo Entity Pro

2.12 MaXpeedingrods Ev Charger

2.13 Mennekes Amtron Charge Control

2.14 Morek Smart AC Charger

Successful connection requires firmware version **A0-MEV-V2.0.9** or newer.

The “Charger idle sampling interval” is not supported. Set this to **0** to avoid a “ClockAlignedDataInterval is read-only” warning.

2.15 Rolec EVO

Tested single phase 7kW model (ROLEC5011) with firmware 1.2.7, appears to be working fine.

Need to configure the OCPP server using the **Rolec Connect** mobile app, and set the current to the maximum (32A), otherwise the exposed `Maximum Current` entity in HA will be capped.

You can still connect with the EVO app via Bluetooth after setting the OCPP server, but certain features (eg. scheduling) may not work.

2.16 Simpson & Partners

All basic functions work properly

2.17 SyncEV Compact EVCP

These are a discontinued (but cheap) 7kw 1PH smart charger, with an OCPP implementation that’s seemingly quite close to standard, and tolerant. Mine works well with the plugin, OCPP setup is done through the local AP-Wifi. The admin panel password is admin. A few plugin tweaks to get full functionality...

- Force SMART mode, to allow setting charge rates (use action `ocpp.set_charge_rate`) and retrieving meter values (use action `ocpp.trigger_custom_message`)
- Manually specify the Measurands
 - Voltage
 - Temperature
 - `Current.Offered`
 - `Current.Import`
 - `Power.Active.Import`
 - `Energy.Active.Import.Register`
- Create an automation triggering action: `ocpp.trigger_custom_message` with `requested_message` set to `MeterValues` on a schedule of your choice to retrieve the Measurands.

- Optionally create an automation updating the heartbeat interval (you have to set a value different to the one in the chargepoint) when the chargepoint reboots.
- I haven't tested using secure mode.
- If you have problems with charging profiles, check your firmware version is 1.6.3 (the latest in Mar 2025)
- Firmware updates can be done through the app, by reconnecting the charger to the original OCPP backend (<wss://cpc.uk.charge.ampeco.tech:443/syncevt/>) and if it says you're on the latest, call them (+44 1952 983 940) to get it updated.

2.18 Teison Smart MINI Wallbox

Use *My Teison* app to enable webSocket. In the socket URL field enter the address of your Home Assistant server including the port. In the socket port field enter *ocpp1.6* for insecure connection or *socpp1.6* for secure connection with certificates. Once enabled, charger doesn't connect to the vendor server anymore and can be controlled only from Home Assistant or locally via Bluetooth.

Even though the device accepts all measurands, the working ones are

- `Current.Import`
- `Energy.Active.Import.Register`
- `Power.Active.Import`
- `Temperature`
- `Voltage`

If the devices loses connection to Home Assistant (due to Wi-Fi disconnection or update, for example) it doesn't seem to reconnect automatically. It is necessary to reboot the charger via Bluetooth for it to reconnect.

2.19 United Chargers Inc. - Grizzl-E

Grizzl-E chargers with firmware 3.x.x work mostly without issue, such as the following:

- Grizzl-E Mini Connect 2024
- Grizzl-E Ultimate

Known issue: In firmware 03.09.0 amperage changes are accepted but not applied. This is due to the firmware accepting but not handling a value of `ChargePointMaxProfile` in `ChargerProfilePurpose`. United Chargers has stated that this will be addressed in firmware version 03.11.0.

Supported OCPP requests for the 3.x.x firmware are documented in a PDF on their site in under <https://grizzl-e.com/connect-to-third-party-ocpp-backend/>

Other Grizzl-E chargers on the 5.x.x firmware have some defects in OCPP implementation, which can be worked around. See [User Guide](#) section in Documentation for details.)

Grizzl-E has now locked OCPP support behind a \$200 paywall <https://grizzl-e.com/ocpp-access/> (support may [unlock](#) access for devices purchased before Sept 2025).

2.20 V2C Trydan

2.21 Vestel EVC04-AC22SW

2.22 Wallbox Pulsar & Copper SB

The Wallbox Pulsar and Copper SB have been verified. In the OCPP-config, leave the password field empty.

2.23 ZJ Beny BCP-A2N-P

Note that there are different models with similar model names, some of which support OCPP and some with other features.

2.24 Others

When a charger is not listed as a supported charger it simply means that it has not been reported to work. Whether it will work or not in practice really depends on whether it is compliant with the OCPP standard. Some vendors claim their device is compliant without bothering to do a compliance test, because that takes time and costs money!

When it is fully compliant, then it should work out of the box, since the ocpp integration is designed to work for fully compliant chargers. Any issues should be reported, and we will do our best to analyze them. In some cases modifications or workarounds may be needed. As long as these workarounds do not break compliance to the OCPP standard they can be added to this repository. Otherwise, we urge you to request your vendor to update their firmware to make their device OCPP compliant.

You can always make your own fork of this repository to solve issues for a specific device that are not OCPP compliant. However, we will not integrate these type of changes into this repository, because that may prevent other chargers to work.

3.1 Installing the OCPP Integration

Follow the steps listed in [README.md](#) to get started. Below are some additional notes which may save you some time.

3.2 Installing HACS (Home Assistant Community Store)

Installation of the HACS integration is a pre-requisite before you can install OCPP. However, it's worth noting that HACS brings a lot of baggage along with it, which is annoying, but this is the price to pay for using a 3rd party repository installer such as HACS. Having said that, once it's up and running, HACS stays out of the way unless you need to `Redownload` or `Remove OCPP`.

The 'baggage' referred to above, is every single repository available through HACS. As you can imagine, this adds up to a huge amount of data being downloaded from the Github servers, and they get upset about it, displaying `Rate Limit` error messages. You will see these error messages whenever you install HACS, but don't worry, the rate limit will reset after a few hours and HACS will be installed. It's worth remembering never to remove HACS unless there is no other way to achieve whatever it is you're wanting to do. Each time you reinstall, you'll be in for a wait of several hours so it's best avoided unless there is no other alternative.

3.3 Configuring the Central System

OCPP Configuration



If you need help with the configuration have a look here: <https://github.com/lbbrhzn/ocpp>

Central system host address

0.0.0.0

Central system port number

9000



Central system identity

central

Charge point identity

charger

Meter interval (seconds)

60



SUBMIT

The `Central system identity` shown above with a default of `central` can be anything you like up to a **maximum of 20 characters**. Whatever is entered in that field will be used as a device identifier in Home Assistant (HA), so it's probably best to avoid spaces and punctuation symbols, but otherwise, enter anything you like.

The `Charge point identity` shown above with a default of `charger` is a little different. Whatever you enter in that field will determine the prefix of all Charger entities added to Home Assistant (HA). My recommendation is that it's best left at the default of `charger`. If you put anything else in that field, it will be used as the prefix for all Charger entities added to HA during installation, however, new entities subsequently added in later version releases sometimes revert to the default prefix, regardless of what was entered during installation. So you end up with a mixture of different prefixes which can be avoided simply by leaving `Charge point identity` set to the default of `charger`.

OCPP Measurands ×

Select which measurand(s) should be shown in Home Assistant.

- Active energy imported from the grid
- Reactive energy imported from the grid
- Active energy imported from the grid during last interval
- Reactive energy imported from the grid during last interval
- Instantaneous active power imported by EV

Measurands (according to OCPP terminology) are actually metrics provided by the charger. Each charger supports a subset of the available metrics and for each one supported, a sensor entity is available in HA. Some of these sensor entities will give erroneous readings whilst others give no readings at all. Sensor entities not supported by the charger will show as **Unknown** if you try to create a sensor entity for them. Below is a table of the metrics I've found useful for the Wallbox Pulsar Plus. Tables for other chargers will follow as contributions come in from owners of each supported charger.

OCPP integration can automatically detect supported measurands. However, some chargers have faulty firmware that causes the detection mechanism to fail. For such chargers, it is possible to disable automatic measurand detection and manually set the measurands to those supported by the charger. When set manually, selected measurands are not checked for compatibility with the charger and are requested from it. See below for OCPP compliance notes and charger-specific instructions in [supported devices](#).

For chargers with multiple connectors (outlets), the OCPP integration will create one device per connector, named `charger Connector 1`, `charger Connector 2` etc. All measurands and other entities (buttons, numbers, switches, diagnostics sensors) that are connector-specific per the OCPP standard will be found on these devices.

3.4 Understanding status

Your charger exposes a connector status sensor:

- Single-connector: `sensor.<charger_id>_status_connector`
- Multi-connector: `sensor.<charger_id>_connector_<connector_number>_status_connector`

For OCPP 1.6, the sensor can show these values:

- **Available** – No EV is connected; the connector is free.

- **Preparing** – EV is connected and/or authenticated but charging hasn't started yet (handshake, cable lock, internal checks).
- **Charging** – Energy is being delivered.
- **SuspendedEV** – The EV has paused energy transfer (e.g., target SoC reached, schedule, thermal limit).
- **SuspendedEVSE** – The charger has paused energy transfer (e.g., power limit, smart charging profile, grid signal).
- **Finishing** – Charging has stopped, but the session isn't fully closed yet (typically waiting for the cable to be unplugged).
- **Reserved** – The connector is reserved (via ReserveNow); only the intended user/ID may start. This is not supported by the OCPP integration yet.
- **Unavailable** – Intentionally set out of service (e.g., ChangeAvailability(Inoperative)) or temporarily not usable. (Entities remain available in Home Assistant.)
- **Faulted** – A fault prevents charging (e.g., ground fault, over-temp, lock error). Check the sensor errorCode for details.

Note In OCPP 1.6, `connectorId = 0` (station level) only uses Available, Unavailable, or Faulted. In OCPP 2.0.1, connector status is simplified to Available / Occupied / Reserved / Unavailable / Faulted; “Preparing/Finishing” are reflected in TransactionEvent rather than as connector statuses.

If your integration shows extra attributes on the connector status sensor like `availability_change` or `availability_pending`, they indicate that a status change (e.g., after ChangeAvailability) has been accepted or scheduled and will take effect once current conditions allow (e.g., after an active session ends).

3.5 Changing availability

- **Availability (charger-level) switch** Sets the entire charger to Unavailable (station-level). All idle connectors switch to Unavailable immediately. Any connector with an ongoing session is marked as scheduled and will switch to Unavailable after the session ends.
- **Availability (per-connector) switches** Set a specific connector to Unavailable. If that connector currently has an ongoing session, the change is scheduled and will take effect once the session ends.
- **Charge Control switch** Turning off ends the ongoing charging session (remote stop). The connector typically transitions to Finishing and then back to its normal idle state once the cable is unplugged. Turning the switch on again resets the session metrics; the charger returns to its previous state (this does not force a new session to start).

3.6 Useful Entities for Wallbox Pulsar Plus

3.6.1 Metrics

- Energy Active Import Register or Energy Session (they give the same readings)
- Power Active Import (instantaneous charging power)
- Current Offered (maximum charging current available)
- Voltage (single phase models only, doesn't work on 3-phase)
- Frequency (single phase models only, doesn't work on 3-phase)
- Time Session (elapsed time from start of charging session)

3.6.2 Diagnostics

- `Status Connector` (shows the current state of available/preparing/charging/finishing/suspended etc)
- `Stop Reason` (reason the charging session was stopped)

3.6.3 Controls

- `Charge Control`
- `Availability` (must be set to ON before EV is plugged in)
- `Maximum Current` (sets maximum charging current available)
- `Reset`

3.7 Useful Entities for ABB Terra AC

3.7.1 Metrics

- `Current.Import` (instantaneous current flow to EV)
- `Energy.Active.Import.Register` (active energy imported from the grid)
- `Power.Active.Import` (instantaneous active power imported by EV)
- `Voltage` (instantaneous AC RMS supply voltage)

3.8 Useful Entities for EVBox Elvi

3.8.1 Metrics

- `Current Offered` (maximum charging current available)
- `Time Session` (elapsed time from start of charging session)
- `Temperature` (internal charger temperature)

3.8.2 Diagnostics

- `Status Connector` (shows the current state of available/preparing/charging/finishing/suspended etc)
- `Stop Reason` (reason the charging session was stopped)

3.8.3 Controls

- `Charge Control`
- `Availability` (OFF when something causes a problem or during a reboot etc)
- `Maximum Current` (sets maximum charging current available)
- `Reset`

3.9 Useful Entities and Workarounds for United Chargers Grizzl-E

Comments below relate to Grizzl-E firmware version 5.633, tested Oct-Nov 2022.

3.9.1 Metrics

The Grizzl-E updates these metrics every 30s during charging sessions:

- `Current Import` (current flowing into EV)
- `Power Active Import` (power flowing into EV)
- `Energy Active Import Register` (cumulative energy supplied to EV during charging session. Resets to zero at start of each session)
- `Time Session` (elapsed time from start of charging session)

3.9.2 Diagnostics

- `Status Connector` (current charger state: available/preparing/charging/finishing/suspended etc)
- `Stop Reason` (reason the charging session was stopped)
- `Latency Pong` (elapsed time for charger's response to internet ping. Good for diagnosing connectivity issues. Usually less than 1000ms)
- `Version Firmware` (charger firmware version and build)

3.9.3 Controls

- `Charge Control` (User switches to ON to start charging session, once charger is in Preparing state. Can be automated in HA - see this [comment in Issue #442](#) for details)
- `Availability` (ON when charger is idle. OFF during active charging session, or when something causes a problem)
- `Maximum Current` (sets maximum charging current available. Reverts to value set by charger's internal DIP switch following reboots; tweak slider to reload)

3.10 Useful Entities for Vestel EVC-04 Wallboxes

3.10.1 Metrics

- **Energy Active Import Register** (cumulative energy supplied to EV during charging session. Resets to zero at start of each session)
- **Energy Active Import Interval** (in case you need the energy spent in total for the current charging session)
- **Power Active Import** (instantaneous charging power)
- **Current Import**
- **Time Session** (elapsed time from start of charging session)

3.10.2 Diagnostics

- **Status Connector** (shows the current state of available/preparing/charging/finishing/suspended etc)
- **Stop Reason** (reason the charging session was stopped)

3.10.3 Controls

- **Charge Control**
- **Availability** (must be set to ON before EV is plugged in)
- **Maximum Current** (sets maximum charging current available)
- **Reset**

3.11 Useful Entities for Rolec EVO

3.11.1 Metrics

- **Current Import**
- **Current Offered** (may be limited by the settings on the charger itself, check the EVO app)
- **Energy Session** (charge for present/last session - kWh)
- **Power Active Import** (active charging power - kW)
- **Temperature** (internal temperature - degrees C)
- **Time Session** (duration of active/last charging session)
- **Voltage** (seems to report a little higher than expected)

There are several other metrics too, I'm not sure what they mean, and also **Export** variants of some of the **Import** entities, but they seem to always be zero for me.

3.11.2 Diagnostics

- Status Connector (Available, Preparing, Charging, etc)

There are many other diagnostic entities about the features, ids, model, firmware etc, not sure if they'd be much practical use.

3.11.3 Controls

- Availability (turning off switches the halo from flashing blue to constant red)
- Charge Control
- Maximum Current (if Current Offered doesn't reach this when charging, raise the current to the max in the EVO app itself, connect via Bluetooth)
- Reset (reboot the charger)
- UnLock (I think this will unlock the charging cable, if permanent lock is enabled from the app)

3.12 OCPP Compatibility Issues

3.12.1 ABB Terra AC

ABB Terra AC firmware 1.8.21 and earlier versions fail to respond correctly when OCPP measurands are automatically detected by the OCPP integration. As of this writing, ABB has been notified, but no corresponding firmware fix is available. As a result, users must configure measurands manually. See the suggested ABB Terra AC configuration in *supported devices*.

3.12.2 Grizzl-E

Grizzl-E firmware 5.x has a few OCPP-compliance defects, including responding to certain OCPP server messages with invalid JSON. Firmware 3.x.x on chargers such as the Mini Connect and Ultimate does not seem to have these issues. Symptoms of this problem include repeated reboots of the charger. By editing the OCPP server source code, one can avoid these problematic messages and obtain useful charger behaviour. ChargeLabs (the company working on the Grizzl-E firmware) expects to release version 6 of the firmware in early 2023, which may fix these problems.

The workaround consists of:

- checking the *Skip OCPP schema validation* checkbox during OCPP server configuration
- commenting-out several lines in `/config/custom_components/ocpp/api.py` and adding a few default values to the OCPP server source code. Details are in this [comment in Issue #442](#)

=====

- *General*
- *FAQ*
 - *too many notifications in home assistant*

4.1 General

If you need help, check out our [forum](#) or submit an [issue](#).

4.2 FAQ

4.2.1 too many notifications in home assistant

The OCPP sends a notification when the charger is rebooted. This can be due to a bad network connection. The notifications can be managed with automations in home assistant. (see <https://github.com/lbbrhzn/ocpp/discussions/938>)

Example:

```
trigger:
  - platform: persistent_notification
    update_type:
      - added
    notification_id: ""
condition:
  - condition: template
    value_template: "{{ trigger.notification.title | lower == \"ocpp integration\" }}"
action:
  - delay:
    hours: 0
    minutes: 10
    seconds: 0
    milliseconds: 0
  - service: persistent_notification.dismiss
    data:
      notification_id: "{{ trigger.notification.notification_id }}"
```

(continues on next page)

(continued from previous page)

```
mode: parallel  
max: 10
```

4.2.2 unstable behavior when setting the charger maximum current

If your charger is acting strange when you're changing the maximum current, or sending an `ocpp.set_charge_rate` action, it might help to clear the charging profiles from the charger.

Run the following action (from Developer Tools):

```
- action: ocpp.clear_profile  
  data:  
    devid: charger
```

Where `charger` refers to your selected charger device identity.

DEBUGGING

To enable debug logging for this integration and related libraries you need to update your Home Assistant `configuration.yaml` file:

```
logger:
  default: info
  logs:
    custom_components.ocpp: debug
```

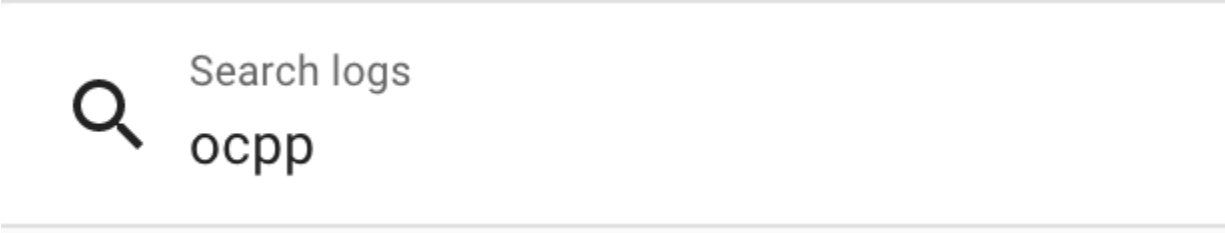
See [Home Assistant Logger](#) for more info.

After a restart detailed log entries will appear in `/config/home-assistant.log`. The log file can be displayed in your webbrowser, by selecting:

Configuration / Settings / Logs / LOAD FULL HOME ASSISTANT LOG



You can filter for OCPP related messages by typing 'ocpp' in the 'search logs' box at the top of the page.



A typical log for a working connection should look like this:

```
2022-03-16 16:33:08 INFO (MainThread) [custom_components.ocpp] {'host': '0.0.0.0', 'port': 9000, 'csid': 'central', 'cpid': 'pulsar', 'meter_interval': 60, 'idle_interval': 900, 'websocket_close_timeout': 10, 'WEBSOCKET_PING_TRIES': 2, 'websocket_ping_interval': 20, 'websocket_ping_timeout': 20, 'skip_schema_validation': False, 'monitored_variables': 'Energy.Active.Import.Register,Energy.Reactive.Import.Register,Energy.Active.Import.Interval,Energy.Reactive.Import.Interval,Power.Active.Import,Power.Reactive.Import,Power.Offered,Power.Factor,Current.Import,Current.Offered,Voltage,Frequency,RPM,SoC,Temperature,Current.Export,Energy.Active.Export.Register,Energy.Reactive.Export.Register,Energy.Active.Export.Interval,Energy.Reactive.Export.Interval,
```

(continues on next page)

(continued from previous page)

```

↪Power.Active.Export,Power.Reactive.Export'}
2022-03-16 16:35:40 INFO (MainThread) [custom_components.ocpp] Websocket Subprotocol
↪matched: ocpp1.6
2022-03-16 16:35:40 INFO (MainThread) [custom_components.ocpp] Charger websocket path=/
↪pulsar
2022-03-16 16:35:40 INFO (MainThread) [custom_components.ocpp] Charger pulsar connected
↪to 0.0.0.0:9000.
2022-03-16 16:35:40 DEBUG (MainThread) [custom_components.ocpp] Received boot
↪notification for pulsar: {'charge_point_serial_number': '88034', 'charge_point_vendor
↪': 'Wall Box Chargers', 'meter_type': 'Internal NON compliant', 'meter_serial_number':
↪', 'charge_point_model': 'PLP1-0-2-4', 'iccid': '', 'charge_box_serial_number': '88034
↪', 'firmware_version': '5.5.10', 'imsi': ''}
2022-03-16 16:35:40 DEBUG (MainThread) [custom_components.ocpp] Updating device info
↪pulsar: {'charge_point_serial_number': '88034', 'charge_point_vendor': 'Wall Box
↪Chargers', 'meter_type': 'Internal NON compliant', 'meter_serial_number': '', 'charge
↪point_model': 'PLP1-0-2-4', 'iccid': '', 'charge_box_serial_number': '88034',
↪'firmware_version': '5.5.10', 'imsi': ''}
2022-03-16 16:35:42 INFO (MainThread) [custom_components.ocpp] Supported feature
↪profiles: Core,FirmwareManagement,LocalAuthListManagement,SmartCharging,RemoteTrigger
2022-03-16 16:35:42 INFO (MainThread) [custom_components.ocpp] Supported feature
↪profiles: Core,FirmwareManagement,LocalAuthListManagement,SmartCharging,RemoteTrigger
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] Get Configuration for
↪NumberOfConnectors: 1
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] Get Configuration for
↪NumberOfConnectors: 1
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] Get Configuration for
↪HeartbeatInterval: 3600
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] Get Configuration for
↪HeartbeatInterval: 3600
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] 'pulsar' post connection
↪setup completed successfully
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] trigger status
↪notification for connector=0
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] 'pulsar' post connection
↪setup completed successfully
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] trigger status
↪notification for connector=0
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] trigger status
↪notification for connector=1
2022-03-16 16:35:42 DEBUG (MainThread) [custom_components.ocpp] trigger status
↪notification for connector=1
2022-03-16 16:36:00 DEBUG (MainThread) [custom_components.ocpp] Connection latency from
↪'central' to 'pulsar': ping=2.0 ms, pong=13.0 ms
2022-03-16 16:36:20 DEBUG (MainThread) [custom_components.ocpp] Connection latency from
↪'central' to 'pulsar': ping=2.0 ms, pong=9.0 ms

```

To debug issues with establishing the ocpp connection, you can enable debug logging for websockets.server:

```

logger:
  default: info
  logs:
    websockets.server: debug

```

Filtering for websockets.server should yield something like this:

```

2022-03-16 16:33:08 INFO (MainThread) [websockets.server] server listening on 0.0.0.
↳0:9000
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] = connection is CONNECTING
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < GET /pulsar HTTP/1.1
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < Connection: Upgrade
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < Host: homeassistant.fritz.
↳box:9000
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < Sec-WebSocket-Key:␣
↳VlpFdctBQgYB6ZokyO2m3Q==
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < Sec-WebSocket-Protocol:␣
↳ocpp1.6
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < Sec-WebSocket-Version: 13
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < Upgrade: websocket
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < User-Agent: WebSocket++/0.8.
↳2
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > HTTP/1.1 101 Switching␣
↳Protocols
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > Upgrade: websocket
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > Connection: Upgrade
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > Sec-WebSocket-Accept:␣
↳hLE0rT2uOtRgVH4VLWoK8K7McNU=
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > Sec-WebSocket-Protocol:␣
↳ocpp1.6
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > Date: Wed, 16 Mar 2022␣
↳15:35:40 GMT
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > Server: Python/3.9␣
↳websockets/10.2
2022-03-16 16:35:40 INFO (MainThread) [websockets.server] connection open
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] = connection is OPEN
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < TEXT '[2,"4a7920fe-1ded-
↳48ff-b9c8-ff8f33bc8118","Boot...: "5.5.10","imsi": ""}]' [318 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > TEXT '[3,"4a7920fe-1ded-
↳48ff-b9c8-ff8f33bc8118",{"cur...0,"status":"Accepted"}]' [129 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < TEXT '[2,"336a0acf-3117-
↳4e72-99c6-f4ae31acb131","Stat...2022-03-16T15:35:40Z"}]' [211 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > TEXT '[3,"336a0acf-3117-
↳4e72-99c6-f4ae31acb131",{}]' [45 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < TEXT '[2,"654f6701-639c-
↳4398-8608-a0c7d8287465","Stat...2022-03-16T15:35:40Z"}]' [211 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > TEXT '[3,"654f6701-639c-
↳4398-8608-a0c7d8287465",{}]' [45 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < TEXT '[2,"694f0dac-fad4-
↳44e6-891c-23d535674cfd","Met... 0,"transactionId": 0}]' [304 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > TEXT '[3,"694f0dac-fad4-
↳44e6-891c-23d535674cfd",{}]' [45 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < TEXT '[2,"c2c18e7a-b6fc-
↳40e4-ba5d-0423bf68d23d","Met... 1,"transactionId": 0}]' [304 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > TEXT '[3,"c2c18e7a-b6fc-
↳40e4-ba5d-0423bf68d23d",{}]' [45 bytes]
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] < TEXT '[2,"5191e2e7-f555-
↳48b3-8b08-626679df5a80","Met... 0,"transactionId": 0}]' [304 bytes]

```

(continues on next page)

(continued from previous page)

```
2022-03-16 16:35:40 DEBUG (MainThread) [websockets.server] > TEXT '[3,"5191e2e7-f555-  
↪48b3-8b08-626679df5a80",{}]' [45 bytes]
```

DEVELOPMENT

It is recommended to use Visual Studio Code, and run home assistant in a devcontainer. See <https://hacs.xyz/docs/developer/devcontainer>

Online development is supported through [GitHub Codespaces](#)