Salia PLC Charge Controller API Guide

eCharge Hardy Barth GmbH
August 5, 2020

Revisions

Revision	Release Date	Changes
1.02	August 7, 2020	added some MQTT topics added use cases
1.01	August 5, 2020	added MQTT topics added screenshots
1	June 20, 2020	initial release

Contents

1. Introduction

The Salia board is an ISO 15118 compliant charging controller for Electric Vehicle Supply Equipment (EVSE). It is equipped with an Fast Ethernet 100 MBit/s network device. The board can be accessed via webbrowser on Port 80 or MQTT on Port 1883.

2. Configuration

2.1 Hardware

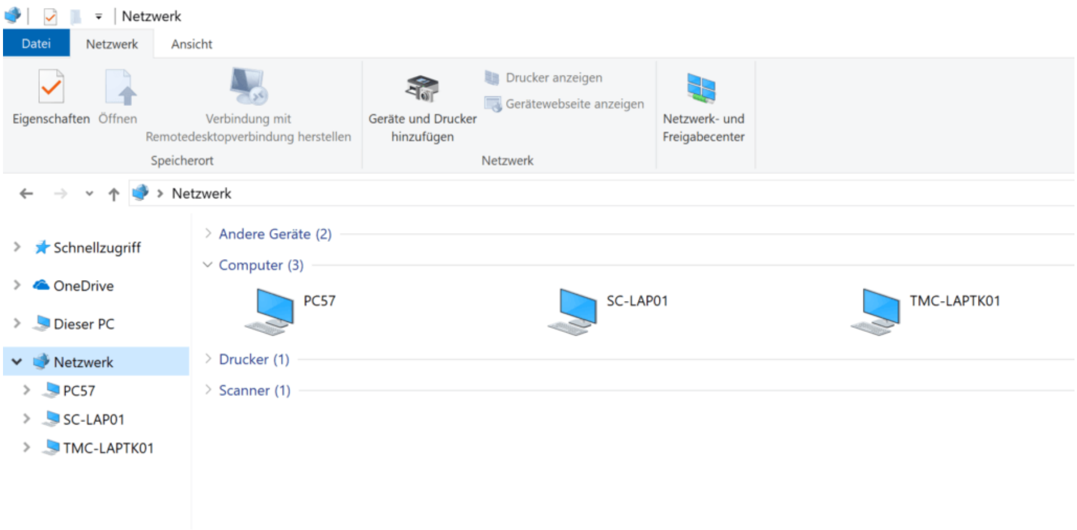
By default the Moon Connect wallbox is delivered as a single socket charger with a fixed type-2 cable.

2.2 Network access

By default the Ethernet device is in DHCP mode.
The default Fallback-IP is: 169.254.12.53

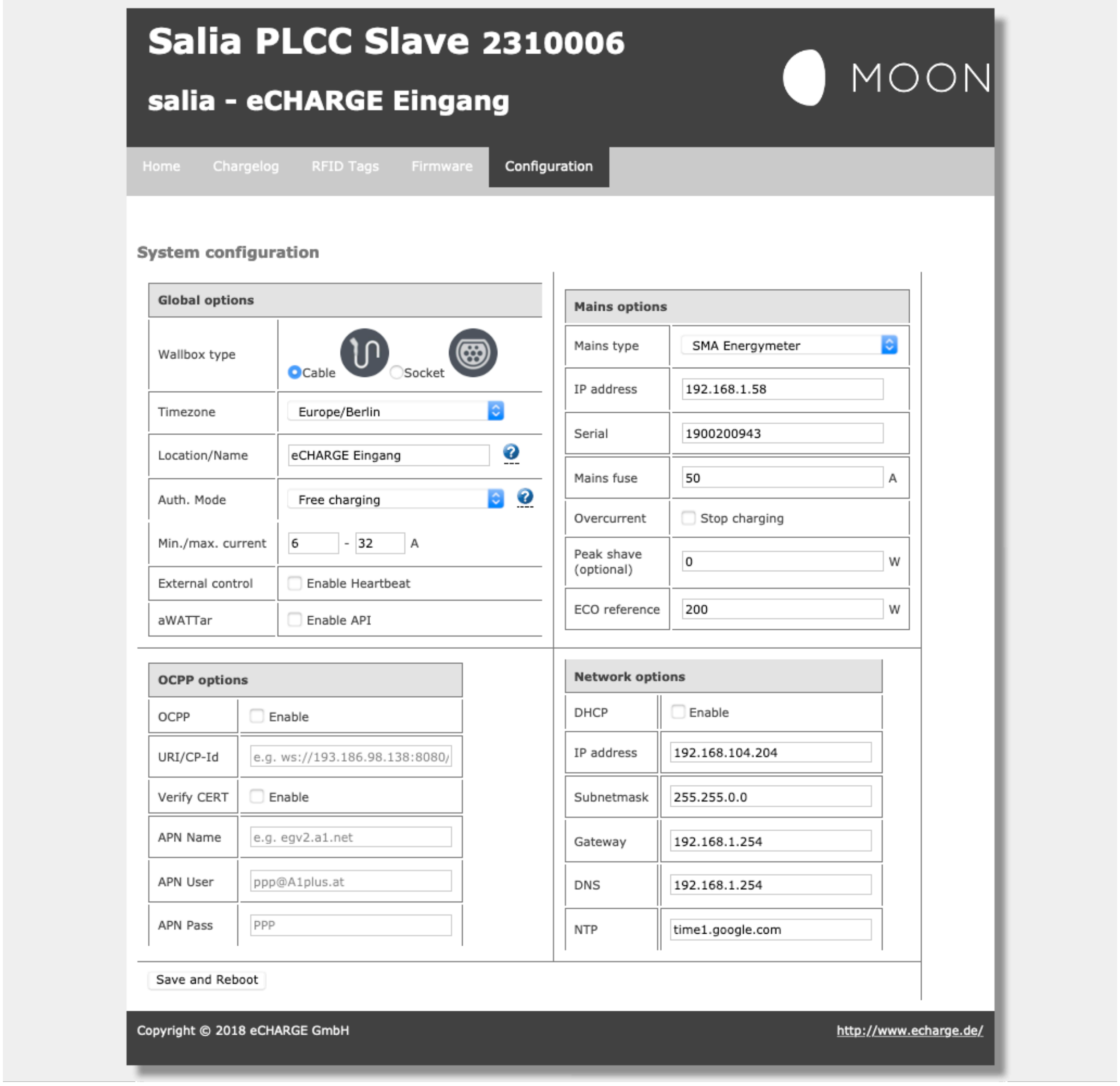
To configure a static IP connect a PC/Notebook via network cable and open then Web-GUI
http://169.254.12.53

Under Windows OS you can detect alls reachable Salia boards in Explorer -> Network



2.3 Web GUI

Standard configuration can be changed via webbrowser
http://169.254.12.53/config.php



Salia PLCC Slave 2310006

salia - eCHARGE Eingang

System configuration

Global options	
Wallbox type	● Cable ○ Socket
Timezone	Europe/Berlin
Location/Name	eCHARGE Eingang
Auth. Mode	Free charging
Min./max. current	6 - 32 A
External control	□ Enable Heartbeat
aWATTar	□ Enable API

Mains options		
Mains type	SMA Energymeter	
IP address	192.168.1.58	
Serial	1900200943	
Mains fuse	50	A
Overcurrent	□ Stop charging	
Peak shave (optional)	0	W
ECO reference	200	W

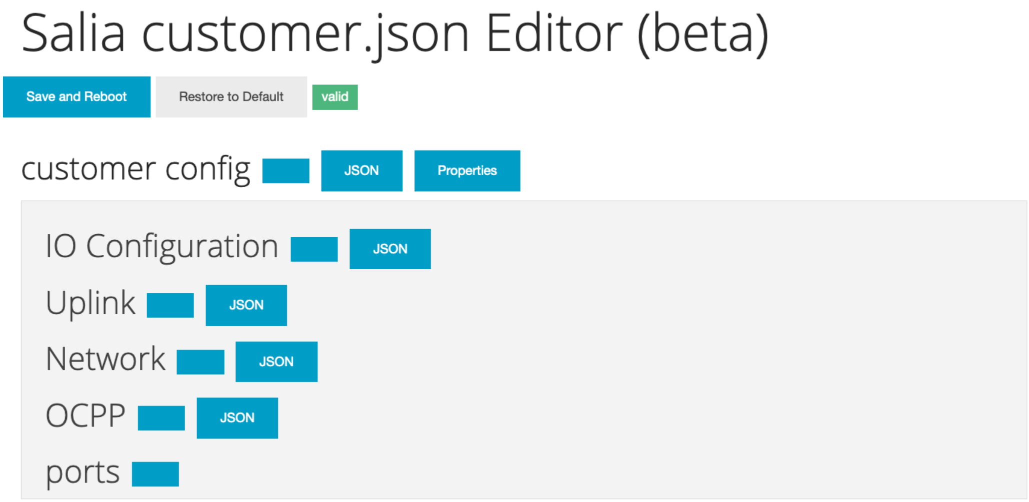
OCPP options	
OCPP	□ Enable
URI/CP-Id	e.g. ws://193.186.98.138:8080/
Verify CERT	□ Enable
APN Name	e.g. egv2.a1.net
APN User	ppp@A1plus.at
APN Pass	PPP

Network options	
DHCP	□ Enable
IP address	192.168.104.204
Subnetmask	255.255.0.0
Gateway	192.168.1.254
DNS	192.168.1.254
NTP	time1.google.com

Save and Reboot

Copyright © 2018 eCHARGE GmbH http://www.echarge.de/

More detailed configuration can be done under
http://169.254.12.53/customeredit/



Salia customer.json Editor (beta)

customer config

IO Configuration

Uplink

Network

OCPP

ports

2.4 Firmware Update

Check the current firmware version in „Home“ tab and install newest image under „Firmware“ if neccesary. Download link:

http://moon.echarge.de/firmware/stable/

To update firmware via USB-Stick copy the image file to a blank USB stick and connect the stick to the salia board. Wait until the board is rebooted.

<TODO>: USB config stuff

EV(s) must not be connected while updating firmware !

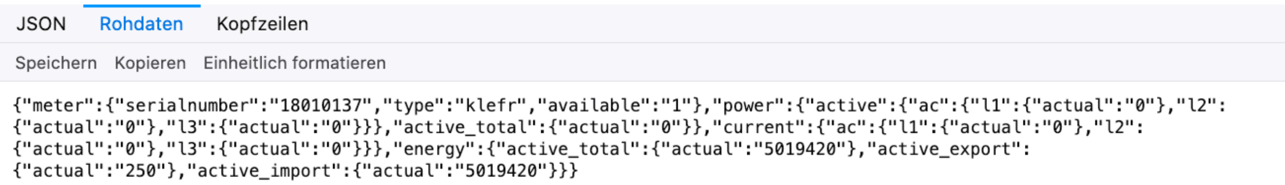
3. Remote access

3.1 REST API

All states and values can be accessed via weblink: http://169.254.12.53/api/

Specific values (e. g. meter values) can be accessed: http://169.254.12.53/api/secc/port0/metering/power

The returned data is in JSON format.



```
{"meter":{"serialnumber":"18010137","type":"klefr","available":"1"},"power":{"active":{"ac":{"l1":{"actual":"0"},"l2":
{"actual":"0"},"l3":{"actual":"0"}}},"active_total":{"actual":"0"}},"current":{"ac":{"l1":{"actual":"0"},"l2":
{"actual":"0"},"l3":{"actual":"0"}}},"energy":{"active_total":{"actual":"5019420"},"active_export":
{"actual":"250"},"active_import":{"actual":"5019420"}}}
```

Values can be changed through HTTP POST.

writeable topics	value(s)	description
salia/chargemode	eco / power / manual	set charging mode
grid_current_limit	0, 6 - 32	set charging current (A) (0 = stop charging)
salia/max_amp	6 - 32	set max current (A)
salia/min_amp	6 - 32	set min current (A)
salia/heartbeat	<anything>	refresh heartbeat
salia/datetime	<timestamp>	set date/time
salia/importrfidfile	url	import rfid-tag-list from another salia module (json-format) (existing tags will be overwritten !)
salia/rfidallow	<Transponder-ID>	set a UID to allowed state
salia/rfidreject	<Transponder-ID>	set a UID to rejected state
salia/rfiddelete	<Transponder-ID>	delete an existing UID

```
Examples (see chapter use cases)
```

3.2 MQTT

<TODO>

Example:
`https://github.com/bluerhinos/phpMQTT`

…

3.3 RFID File Format

To import Transponder Tags via weblink, the file must be in JSON Format.

Example File:

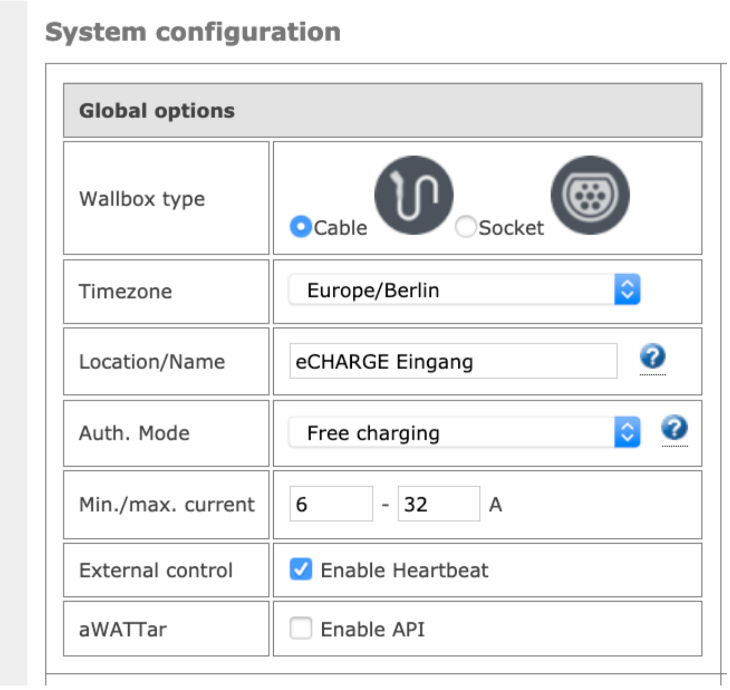
```
{
	"AA4A6910" : { "name" : "John Smith", "state" : 1 },
	"043B89B2B31F80" : { "name" : "Testuser 2", "state" : 1 },
	"B614C32E" : { "name" : "Testuser 3", "state" : 1 }
}
```

`(see example command salia/importrfidfile)`

4. Use cases

4.1 Wallbox in manual mode (controlled by external controller)

Activate Heartbeat in „configuration“ tab.



System configuration

Global options	
Wallbox type	● Cable ○ Socket
Timezone	Europe/Berlin
Location/Name	eCHARGE Eingang
Auth. Mode	Free charging
Min./max. current	6 - 32 A
External control	☑ Enable Heartbeat
aWATTar	□ Enable API

Set manual mode:

```
curl -X PUT -d ’{“salia/chargemode“:“manual“}’ http://169.254.12.53/api/secc
```

Set charging current to 10 A:

```
curl -X PUT -d '{"grid_current_limit":"10"}' http://169.254.12.53/api/secc
```

Send heartbeat (< 60 sec interval):

```
curl -X PUT -d ’{“salia/heartbeat“:“alive“}’ http://169.254.12.53/api/secc
```

Stop charging:

```
curl -X PUT -d '{"grid_current_limit":"0"}' http://169.254.12.53/api/secc
```

```
Import RFID-Tag-List from another salia module:
curl -X PUT -d '{"salia/importrfidfile":"http://169.254.12.54/rfidtags.json"}'
     http://169.254.12.53/api/secc
```