

OpenEVSE – Developers Guide

HTTP



IMPORTANT SAFETY INSTRUCTIONS



Read and save these instructions prior to installing and operating your Charging Station. Retain this installation guide for maintenance and troubleshooting information. If you have further questions, contact Customer Service at support@openevse.com.

WARNING: To reduce the risk of fire, electric shock, and serious bodily injury, observe the following:

- Installation work and electrical wiring must be done by qualified person(s) in accordance with all applicable codes and standards.
- When cutting or drilling into structure, do not damage electrical wiring and other hidden utilities.
- Use this device only in the manner intended.

CAUTION: The installation of this charging Station must be in accordance with all national and local electrical codes.

CAUTION: Exercise caution and common sense when powering the device. Do not connect to a damaged power source.

WARNING: Power must be disconnected before installation and servicing, cleaning, and other user-maintenance. Failure to disconnect power creates risk of fire, electric shock, and serious bodily injury.

CAUTION: The product warranty will not cover equipment damage or failure that is caused by improper installation or operation.

WARNING: Do not install in an environment that is excessively dusty, conductive, corrosive, or gas-filled, is exposed to open flames (e.g., gas-burning stoves), is near strong chemicals or solvents, or where there is excessive heat, shock, or vibration.

CAUTION: This charging station is not intended for use by persons (including children) with reduced physical, sensory, or mental capabilities, or lack of experience and knowledge, unless they have been given supervision or instruction concerning the use of the charging station by a person responsible for their safety. Children should be supervised to ensure that they do not play with the charging station.

Contents

License	3
MQTT	4
Prerequisites.....	4
MQTT Setup.....	4
MQTT Publish	Error! Bookmark not defined.
MQTT Subscribe.....	Error! Bookmark not defined.
Advanced RAPI over MQTT	Error! Bookmark not defined.
OpenEVSE Remote API.....	7
Station Commands.....	9
Set Commands.....	10
Get Commands.....	12
RAPI over WiFi	13
Additional Resources	14

License

This manual was written by OpenEVSE LLC and is released under the Creative Commons 3.0 with Attribution, share alike license.

OpenEVSE is a open project with source materials freely shared. OpenEVSE may be used for commercial purposes in accordance with the following licenses:

OpenEVSE Code license - **GNU General Public License v3**

<http://www.gnu.org>



OpenEVSE Content license (including this document) - **Creative Commons 3.0 with Attribution Share alike (CC 3.0 BY-SA)**

<http://creativecommons.org/licenses/by-sa/3.0/>



The OpenEVSE Project and Source code has been evaluated by the Open Source Hardware Foundation and meets the requirements for Open Hardware. The registration number for OpenEVSE is US000028.



US000028

HTTP

Prerequisites

- OpenEVSE controller with firmware 4.8.0 or higher
- OpenEVSE WiFi with firmware 3.7.4 or higher
- Optional - Access Point with network access

HTTP

OpenEVSE offers full control of a charging station over a local network via HTTP. The OpenEVSE WiFi module runs a HTTP server which can be accessed via a browser or directly via HTTP requests and responses.

This document is intended for applications developers and covers direct HTTP requests and responses.

Onboarding

Onboarding the OpenEVSE device onto your local network is one of the first step most users will wish to do. The process is similar to most IOT devices. OpenEVSE will create an access point If there is no saved configuration, if the configured network is not found or if the GPIO button is pressed for 5 seconds.

The default SSID is OpenEVSE_XXXXXX where the six XXXXXX represent the unique host portion of the MAC address.

Default SSID - OpenEVSE_XXXXXX

Default Password - openevse

Default IP - 192.168.4.1 (Access point mode only)

Default name - http://openevse.local (availability may be dependant based on operating system)

TIP - Developers may wish to save the SSID name and scan the available WiFi network periodically. This will allow the user to control the station in offline mode if the network is unavailable or the station is used in multiple locations.

WiFi Network Scan

After connecting to **OpenEVSE_XXXXXX** in Access Point Mode a network scan may be performed by sending a http request for 192.168.4.1/scan or openevse.local/scan

HTTP GET /scan

HTTP Response (JSON)

"rssi": Relative Signal Strength Indicator
"ssid": Network Name
"bssid": MAC Address of Network
"channel": WiFi Channel
"secure": Security Mode
"hidden": Visible or Hidden

Response Example

```
[{"rssi":77,"ssid":"GreenNinja","bssid":"A0:04:60:5D:93:7B","channel":1,"secure":4,"hidden":false}, {"rssi":82,"ssid":"NSYPTR","bssid":"40:70:09:47:E8:60","channel":1,"secure":4,"hidden":false}, {"rssi":-79,"ssid":"MySpectrumWiFi60-2G","bssid":"38:35:FB:B1:67:66","channel":1,"secure":4,"hidden":false}, {"rssi":84,"ssid":"MySpectrumWiFi","bssid":"E8:AD:A6:5B:91:EE","channel":11,"secure":4,"hidden":false}]
```

Join WiFi Network

To join an network and save SSID and password POST a form to /savenetwork with parameters for network SSID and password.

HTTP POST

```
192.168.4.1/savenetwork/? ssid=mywifinet&pass=abcd1234  
openevse.local/savenetwork/? ssid=mywifinet&pass=abcd1234
```

Status

Sending a http get request to /status will provide network and OpenEVSE status information.

HTTP GET /status

HTTP Response (JSON)

```
"mode": WiFi Mode  
"wifi_client_connected": Client Mode  
"srssi": Signal Strength  
"ipaddress": IP Address (DHCP Assigned in Client Mode)  
"emoncms_connected": Connected to Energy Monitoring Service? (EmonCMS)  
"packets_sent": Packets Sent to EmonCMS  
"packets_success": Successful Responses from EmonCMS  
"mqtt_connected": Connected to MQTT Broker?  
"ohm_hour": Connected to OhmConnect, Ohm Hour Status  
"free_heap": WiFi memory  
"comm_sent": RAPI Packets Sent to OpenEVSE Controller  
"comm_success": Successful Responses from OpenEVSE Controller  
"amp": Value from Current sensor  
"pilot": Current Setting for Pilot
```

```
"temp1": Temperature reading from main Temperature sensor
"temp2": Temperature from optional secondary sensor, -2560 if not present
"temp3": Temperature from optional IR sensor, -2560if not present
"state": EVSE State - 1 Not Connected - 2 Connected - 3 Charging - 4 Error
"elapsed": Elapsed time for Charge session
"wattsec": Elapsed watt seconds added during this charge session
"watthour":Elapsed watt hours added lifetime (or since last counter reset)
"gfcicount":Error counter GFCI
"nogndcount": Error counter missing Ground
"stuckcount": Error counter Stuck Relay
"divertmode": Solar Divert Mode
"solar": Solar output received via MQTT
"grid_ie": Grid Import/Export received via MQTT
"charge_rate": Charge rate from MQTT
"divert_update": Time since data received via MQTT
```

Response Example

```
{"mode":"STA","wifi_client_connected":1,"srssi":65,"ipaddress":"192.168.50.128","emoncms_connected":1,"packets_sent":5,"packets_success":5,"mqtt_connected":1,"ohm_hour":"None","free_heap":21840,"comm_sent":79,"comm_success":78,"amp":0,"pilot":40,"temp1":190,"temp2":2560,"temp3":2560,"state":1,"elapsed":0,"wattsec":0,"watthour":334223,"gfcicount":0,"nogndcount":0,"stuckcount":0,"divertmode":1,"solar":0,"grid_ie":2,"charge_rate":0,"divert_update":0}
```

Configuration

```
"firmware": Firmware of OpenEVSE controller
"protocol": Remote API RAPI version on OpenEVSE controller
"espflash": Flash memory of WiFi controller
"version": Version of WiFi Firmware
"diode": Diode Check Enabled/Disabled
"gfci": GFCI Check Enabled/Disabled
"ground": Ground Check Enabled/Disabled
"relay": Stuck Check Enabled/Disabled
"vent": Vent Required Check Enabled/Disabled
"temp": Temperature Check Enabled/Disabled
"service": Charging Service Level
"scale": Current meter calibration Scale
"offset": Current meter calibration Offset
"ssid": Client SSID
"pass": Password - Response "__DUMMY_PASSWORD__",
"emoncms_enabled": Energy Monitoring Service Enabled? (EmonCMS)
"emoncms_server": Server URL or IP address
"emoncms_node": Server node
"emoncms_apikey": Server API Key - Response "__DUMMY_PASSWORD__",
"emoncms_fingerprint": SSL Fingerprint,
"mqtt_enabled": MQTT Enabled?
"mqtt_server": MQTT URL or IP address
"mqtt_topic": MQTT Topic
"mqtt_user": MQTT User
"mqtt_pass": MQTT Password - Response "__DUMMY_PASSWORD__",
```

```
"mqtt_solar": MQTT Solar Topic
"mqtt_grid_ie": MQTT Grid Import/Export Topic
"www_username": User name for HTTP Server
"www_password": Password for HTTP Server
"ohm_enabled": OhmConnect Service enabled?
```

Response Example

```
{"firmware":"5.0.1","protocol":"4.0.1","espflash":4194304,"version":"2.7.5","diodes":0,"gfcit":0,"groundt":0,"relayt":0,"ventt":0,"tempt":0,"service":2,"scale":220,"offset":0,"ssid":"chris_ap","pass":"__DUMMY_PASSWORD__","emoncms_enabled":true,"emoncms_server":"data.openevse.com/emoncms","emoncms_node":0,"emoncms_apikey":"__DUMMY_PASSWORD__","emoncms_fingerprint":"","mqtt_enabled":true,"mqtt_server":"192.168.50.253","mqtt_topic":"openevse","mqtt_user":"emonpi","mqtt_pass":"__DUMMY_PASSWORD__","mqtt_solar":"","mqtt_grid_ie":"emon/emonpi/power1","www_username":"","www_password":"","ohm_enabled":true}
```

OpenEVSE Remote API

OpenEVSE Remote API (RAPI) is a simple lightweight communications protocol. RAPI allows the user/application to:

- Execute Commands
- Get Status
- Change Settings
- Debug

RAPI is designed to work in any communications environment including high loss connections such as wireless. RAPI provides optional features to ensure reliable communications:

- Command/Response
- Checksum (NONE, SUM, XOR [recommended])
- Sequence ID

RAPI documentation is located in the OpenEVSE Source Code, file rapi_proc.h on Github [Link](#).

XOR checksum (recommended)

```
$cc pp^xk\r
```

No checksum (FOR Experimentation ONLY)

```
$cc pp\r
```

Checksum + Sequence ID

```
$cc pp :ss^xk\r
```

Structure

\$= start of RAPI command
cc = 2-letter command
pp = parameters
xk = 2-hex-digit checksum - 8-bit XOR of all characters before '^'
ss = optional 2-hex-digit sequence id - response will echo the
sequence id - ss CANNOT be 00, which is reserved as an
invalid value
\r = carriage return = 13d = 0x0D

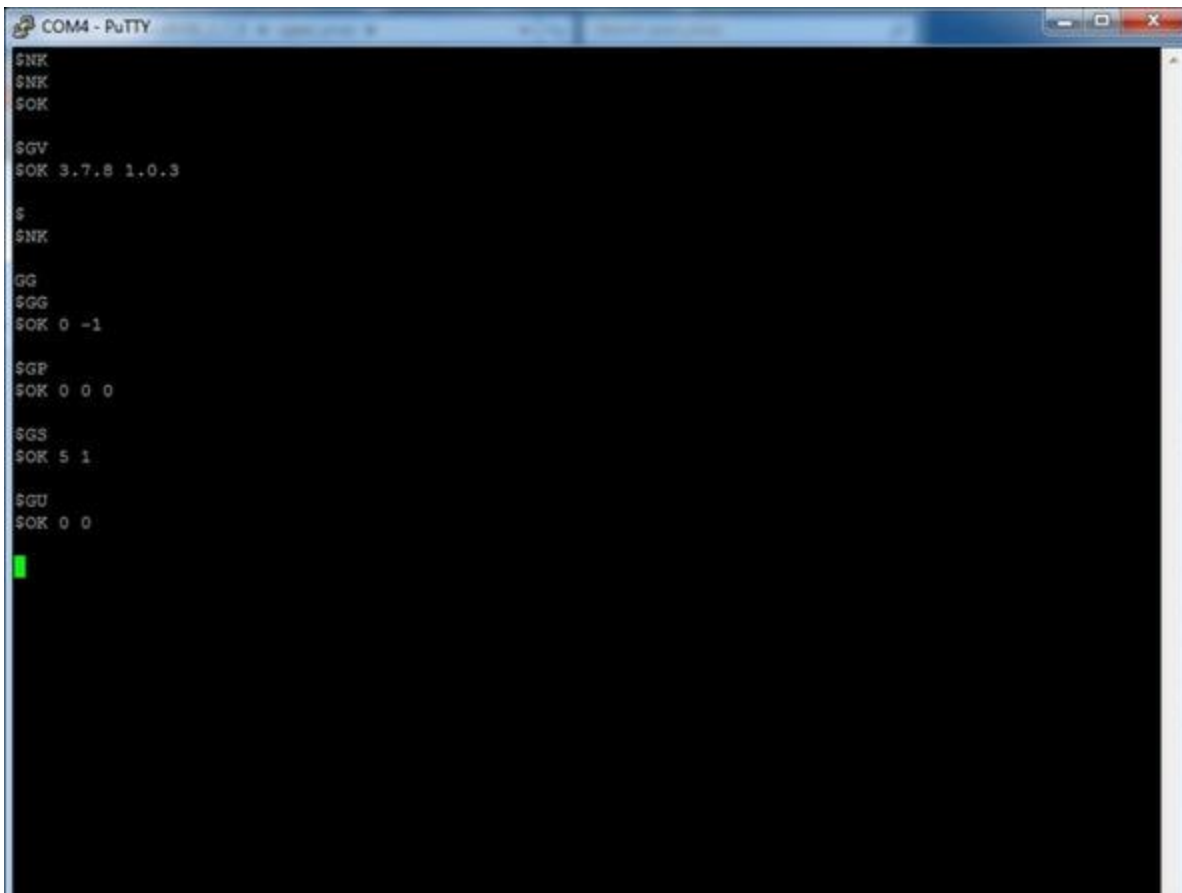
Response format

\$OK [optional parameters] [:ss]^xk\r - success
\$NK [optional parameters] [:ss]^xk\r - failure
Checksum and Sequence ID are only present if send with the command

Asynchronous notification messages

\$ST state\r - EVSE state transition - sent whenever EVSE state changes
state: EVSE_STATE_XXX

\$WF mode\r - Request client WiFi mode:
WIFI_MODE_XXX



Station Commands

Station commands can control the availability of the charging station or change the properties of the station such as LCD color or text.

Enable/Disable/Sleep

FS puts the charging station in a “not ready” normal state (pilot high PWM off)

FD disables the station in a “not ready” error state (pilot low PWM off)

FE enables the station in a “ready” state (pilot high PWM on if connected)

Text on LCD

F0 0 Disable display updates to give RAPI control of display

FP Write desired message to LCD

F0 1 Enable display updates to give OpenEVSE controller control of display

Command	Parameter	RAPI + XOR	Response	Description
F0	0	\$F0 0^42	\$OK^20	Disable display updates
	1	\$F0 1^43	\$OK^20	Enable display updates
F1		\$F1^53	\$OK^20	simulate button press

FB	0	\$FB 0^30	\$OK^20	LCD Backlight OFF
	1	\$FB 1^31	\$OK^20	LCD Backlight RED
	2	\$FB 2^32	\$OK^20	LCD Backlight GREEN
	3	\$FB 3^33	\$OK^20	LCD Backlight YELLOW
	4	\$FB 4^34	\$OK^20	LCD Backlight BLUE
	5	\$FB 5^35	\$OK^20	LCD Backlight VIOLET
	6	\$FB 6^36	\$OK^20	LCD Backlight TEAL
	7	\$FB 7^37	\$OK^20	LCD Backlight WHITE
FD		\$FD^26	\$OK^20	Disable EVSE
FE		\$FE^27	\$OK^20	Enable EVSE
FP	text	\$FP 0 0 Hello World^32	\$OK^20	Print Text to LCD
FR		\$FR^30	\$OK^20	Reset EVSE
FS		\$FS^31	\$OK^20	Sleep EVSE

Set Commands

Set commands change common firmware and behavior options such as setting the station timers, Service Level and charge current.

Maximum Current

SC sets the stations maximum charge current and adjusts the pilot signal duty cycle advertised to the vehicle. This setting can be adjusted dynamically as desired, the vehicle must comply.

The default action is to save new current capacity to EEPROM. If frequent changes are expected, a V flag should be appended to the command to prevent the changed from being saved to EEPROM.

Minimum and Maximum charge currents can compiled into firmware, for enhanced security these cannot be modified over RAPI. Modification requires re-flashing firmware via an Inline Serial Programmer (ISP). If the commanded setting is outside the defined range RAPI will return \$NK and the current setting.

If OpenEVSE is currently experiencing an over temperature event, raising current capacity will fail and return \$NK and the current setting.

Service Level

SL sets the Service level on the display and uses the current set in EEPROM. This should be set based on the capabilities of the station and the input service.

For countries with single phase 220-240v power Service level should be set to L2.

In the United states or other countries with split phase power 120 or 240v the service level can be set to auto. OpenEVSE will detect power on each line and determine if the power is 120v or 240v.

Timers

ST will set standard Start (hour and minute) and Stop (hour and minute) timers. OpenEVSE will sleep outside of the set time. Sleep can be overridden by pressing the button or sending the F1 command to simulate a button press. Once the Start time has occurred the station will wake and begin charging if a vehicle is attached.

Session Limits

Limits can be defined for the current charging session based on time (charge for 2hours and 30 minutes) or energy added in kWh (add only 10kwh). S3 sets a time limit in 15 minute increments. S3 10 would set the station to charge for 150 minutes (2 hours and 30 minutes). SH sets the kwh limit. SH 10 will add a maximum of 10kwh to the vehicle.

Parameter	Parameter	RAPI + XOR	Description
S0		0 \$S0 0^57	Set Display type Monochrome
		1 \$S0 1^56	Set Display type Color
S1	yr mo dy hr mn sc	\$S1 18 10 18 12 30 00	Set RTC clock 2018 Oct 18 12:30:00
S2		0 \$S2 0^55	Read Ammeter only while charging
		1 \$S2 1^54	Read Ammeter in all states
S3	0 - 255	\$S3 4^50	Session Limit x * 15 minutes
S4		0 \$S4 0^53	Socket Unlock (IF enabled and Installed)
		1 \$S4 1^52	Socket Lock (IF enabled and Installed)
SA	scale offset	\$SA 182 0^3D	Current Measurement Calibration
SC	amps	\$SC 24^12	Set Current and Save to EEPROM
	V	\$SC 24 V^64	Set current and DO NOT Save to EEPROM
SH	kwh	\$SH 10^1E	Session Limit kwh (Ex stop after 10kwh)
SK		0 \$SK 0^2C	Set Station kwh Total to 0
SL		1 \$SL 1^2A	Service Level 1
		2 \$SL 2^29	Service Level 2
	A	\$SL A^5A	Autodetect Service level - US split phase
ST	starthr mn endhr mn	\$ST 0 0 0 0^23	Set Start and End Timers

Get Commands

Get commands retrieve status the current state, settings and values of sensors.

Power and Energy

GG will get the current measured power output in milliamps. GU will get the Usage statistics for the current session and total as calculated by the OpenEVSE controller.

Temperature

GP will get the temperatures from any installed temperature sensor is 10th of a degree C. Most OpenEVSE Stations have just 1 sensor installed.

	RAPI + ck	Response	Description
G0	\$G0^53	\$OK 0^30	Get EV connected State - Disconnected
		\$OK 1^31	Get EV connected State - Connected
		\$OK 2^32	Get EV connected State - Unknown
G3	\$G3^50	\$OK count	Get Session time limit count * 15 minutes
G4		\$OK 0^30	Get Lock Status - Unlocked
		\$OK 1^30	Get Lock Status - Locked
GA	\$GA^22	\$OK scale offset	Get Ammeter Calibration Settings
GC	\$GC^20	\$OK minamp maxamp	Get controllers Min and Max Current
GD	\$GD^27	\$OK starthr min endhr mn	Get Charge Timer Start and End time
GE	\$GE^26	\$OK amp flags(hex)	Get Current and settings
GF	\$GF^25	\$OK gfi nognd stkrly	Get Fault Counters GFI Ground and Stuck Relay
GG	\$GG^24	\$OK milliamps -1	Get measured current in milliamps
GH	\$GH^2B	\$OK kwh	Get Session charge limit in kwh
GO	\$GO^2C	\$OK ambient ir	Get Overtemperature threshold 10th °C
GP	\$GP^33	\$OK ds3231 mpc9808 tmp7	Get Temperature from sensors -2560 = not installed
GS	\$GS^30	\$OK state elapsed	Get EVSE State and elapsed charge time
GT	\$GT^37	\$OK yr mo dy hr mn sc	Get Time Year Month Day Hour Minute Second
GU	\$GU^36	\$OK wattsec wtotal	Get Energy session watt seconds and total kwh
GV	\$GV^35	\$OK firmware protocol	Get EVSE firmware and protocol version

Debugging

	Feature	Parameter	RAPI + XOR	Response	Description
FF	D	0	\$FF D 0^50	\$OK^20	Feature Diode Check Disable
	D	1	\$FF D 1^51	\$OK^20	Feature Diode Check Enable
	E	0	\$FF E 0^51	\$OK^20	Feature Command Echo Disable
	E	1	\$FF E 1^50	\$OK^20	Feature Command Echo Enable
	F	0	\$FF F 0^52	\$OK^20	Feature GFI Self Test Disable
	F	1	\$FF F 1^53	\$OK^20	Feature GFI Self Test Enable
	G	0	\$FF G 0^53	\$OK^20	Feature GMI Disable
	G	1	\$FF G 1^52	\$OK^20	Feature GMI Enable
	R	0	\$FF R 0^46	\$OK^20	Feature Stuck Relay Test Disable
	R	1	\$FF R 1^47	\$OK^20	Feature Stuck Relay Test Enable
	T	0	\$FF T 0^40	\$OK^20	Feature Temperature Monitoring Disable
	T	1	\$FF T 1^41	\$OK^20	Feature Temperature Monitoring Enable
	V	0	\$FF V 0^42	\$OK^20	Feature Vent Required Check Disable
	V	1	\$FF V 1^43	\$OK^20	Feature Vent Required Check Enable

RAPI over HTTP

RAPI commands can be sent to OpenEVSE over HTTP. Settings and values not published to /config or /status can be obtained by sending the appropriate RAPI command. It is recommended the command is sent with a checksum.

- Note in URL format \$ is represented with %24
- Note in URL format ^ is represented with %5E
- Note in URL format (Space) is represented with +

Example Get the controllers Minimum and Maximum Current set in Firmware with the \$GC^20 RAPI command.

```
http://openevse.local/r?rapi=%24GC^20
```

```
>$OK 6 80^1E
```

Example Set current to 24A with the command \$SC 24^12

```
http://openevse.local /r?rapi=%24SC+24%5E12
```

```
>$OK 24^06
```

Additional Resources

Online Solutions, Forums and Trouble Tickets

<http://support.openevse.com>

E-mail support@openevse.com

Online Guides

<http://guides.openevse.com>

Store

<http://store.openevse.com>

Website

<http://www.openevse.com>

Source Code - Firmware - Schematics, etc.

<https://github.com/openevse>