![OpenEVSE logo]

OpenEVSE – Developers Guide

MQTT

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

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

MQTT

Prerequisites

- OpenEVSE controller with firmware 4.8.0 or higher
- OpenEVSE WiFi with firmware 3.7.4 or higher
- Connected to an Access Point with network access
- MQTT Broker on the network

MQTT Setup

- Using a web browser enter the IP address of your OpenEVSE WiFi

- Click the Services tab
- Click the "Enable MQTT" Checkbox
- Enter the IP Address or hostname for the MQTT Broker
- Enter the Username and Password for the MQTT Broker
- Enter the <base-topic> OpenEVSE will publish. **Default** openevse

MQTT Publish

Once connected to a MQTT Broker, OpenEVSE will publish the many commonly used values at regular intervals.

Structure - `<base-topic>/<sub-topic> <value>`
Default - `<base-topic>` openevse

Published - `<sub-topic>`
Published to broker every 30 seconds

```
openevse/amp            Measured current in milliamps
openevse/wh             Calculated watthours for the current session
openevse/temp1          Sensor value in 10th degree C (if installed)
openevse/temp2          Sensor value in 10th degree C (if installed)
openevse/temp3          Sensor value in 10th degree C (if installed)
openevse/pilot          Pilot current sent to vehicle in Amps (6-80)
openevse/state          EVSE State 1–Ready, 2-Connected, 3-Charging, 4-Error
openevse/freeram        WiFi free Ram
openevse/divertmode     Divert Mode 1–Normal, 2–Eco Divert
```

```
Published to broker every 5 seconds

openevse/chargerate     Calculated power available from Grid I/E Topic
openevse/grid_ie        Last Value received on Grid I/E Topic
openevse/divert_update  Time since received on Grid I/E Topic
```

Grid I/E Topic:

`<topic>/<subtopic>` defined in OpenEVSE WiFi interface. Example `emon\emonpi\power1`

OpenEVSE will subscribe to an energy topic and adjust charging power based an available energy. Available energy must be represented in negative watts. Example: A residence is producing 1077 watts of excess energy from a solar system, the solar/energy monitoring system should publish -1077 to MQTT on the topic OpenEVSE was set to subscribe.
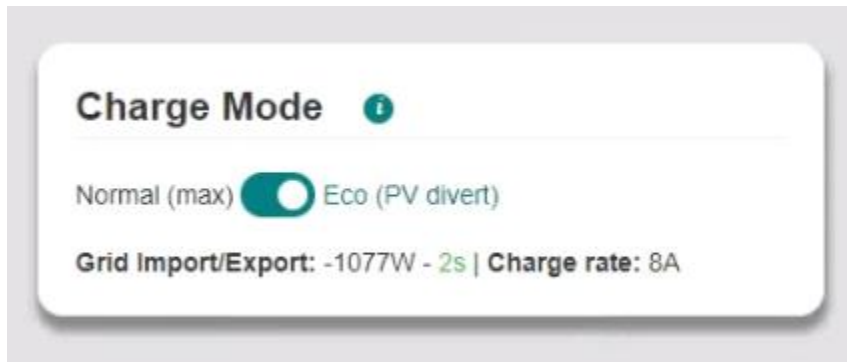
Divert Mode:

Divert mode can be enabled or disabled over MQTT, OpenEVSE subscribes to the `<base-topic>/divertmode/set`.

`<base-topic>/divertmode/set Divert Mode 1-Normal, 2-Eco Divert`

Advanced RAPI over MQTT

OpenEVSE subscribes to a `<base-topic>`/rapi/in topic and will execute commands, check status and change settings. Commands received over MQTT are sent to the OpenEVSE controller via serial using the OpenEVSE Remote API (RAPI). Responses from the OpenEVSE controller are published to `<base-topic>/rapi/out/.`

Commands - `<base-topic>/rapi/in/<command>`
Response - `<base-topic>/rapi/out/<response>`

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	FeatureVent Required Check Enable
T0		amp	$T0 75	$OK	Set Fake Charging Current
Z0		delay hold			Relay delay and hold PWM duty cycle

Z0 FOR TESTING RELAY_AUTO_PWM_PIN ONLY

Z0 closems holdpwm
    closems(dec) = # ms to apply DC to relay pin
    holdpwm(dec) = pwm duty cycle for relay hold 0-255

RAPI over WiFi

The OpenEVSE WiFi interface provides a great place to test RAPI commands. To enable RAPI over WiFi:

- Click the System Tab
- Click the Developers Mode switch
- Click the RAPI tab
- Enter Desired command and click Send

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