EVerest

A Game-Changer in EV Charger Compatibility
Robert de Leeuw
Technology EVangelist
EV Charging Protocol Specialist

Working @ PIONIX since jan-2023

https://www.linkedin.com/in/robertdeleeuw/
Charging a car is easy RIGHT?

ONE DOES NOT SIMPLY

PLUG IT IN AND IT CHARGES


Just over 72% of the plugs in the Bay Area were functional.

EV Charging Party: lots of parties, lots of protocols

CPO: Charge point operator, HEMS: Home energy management system

https://www.ffe.de/en/publications/normenlandschaft_fuer_die_elektromobilitaet/
(Almost) a wonder DC charging works

IEC61851
- Simple Pulse Width Modulation (PWM) of 1kHz between 0V-12V
- Voltage level manages states
- Duty cycle manages maximum current
- Minimum amount of information exchanged

ISO15118 / DIN-SPEC70121
- Powerline Communication (PLC) modulated on top of PWM
- Enables High Level Communication
- Transport over TCP/IP IPv6
- SLAC for EV and EVSE matching
- EXI encoding and decoding of ISO15118 messages (binary XML Format)

https://en.wikipedia.org/wiki/Charging_station
Too many variants: impossible to test

- Estimated EV Brands: 50
- Estimated EV Charger Manufacturers: 200 - 300
- Estimated Charge Point Operators: 1000 in Europe alone
But Tesla makes it look so easy?

- 1 EV Manufacturers: Tesla
- 1 EV Charger Manufacturers: Tesla
- 1 Charge Point Operators: Tesla
How to solve this: limit the variants

- Estimated EV Brands: 50
- EV Charger Software Stacks: 1
- Estimated Charge Point Operators: 1000 in Europe alone
What is EVerest?

EVerest in a nutshell

EVerest is a complete Operating system for EV Chargers.

- Implements all standards running on a charger
- Runs on tiny embedded linux
- Aims to support as many different HW platforms as possible
- Provides all functionality needed from smart home chargers to commercial fast chargers
- Commercial friendly open source (Apache 2.0 licensed)
- Modular
- Don’t reinvent the wheel
The EVerest framework: *a flexible microservice architecture*

- Modules are individual Linux processes
- Modules expose (multiple) interfaces over MQTT
- Config file to connect module requirements/interfaces
- Framework starts/stops/restarts modules
- Communication via MQTT broker, transparent to modules
- Loose coupling: A does not know B, B does not know A
- Framework manages synchronous RPC, argument type safety, dependency checking

**Module A**

Provides Interface: Charger

Commands:
- set_max_current(int ampere)

Variables:
- int energy_charged

**Module B**

Requires Interface: Charger

call commands (with return value)

async event: energy_charged: 5.23kWh
The EVerest visual config builder
Running EVerest as a simulator
How to get involved

Check out the code: https://github.com/EVerest
Hardware designs and microcontroller firmware: https://github.com/orgs/PionixPublic/repositories

EVerest Mailing list: https://lists.lfenergy.org/g/everest
EVerest project page on Linux Foundation Energy: https://www.lfenergy.org/projects/everest

Quick start guide: https://everest.github.io/general/quick_start_guide.html

Technical Steering Committee (TSC): Follow the evolution of EVerest, get involved, open to all!
   Every 4th thursday of the month, announced via mailing list
   Recordings available on YouTube: https://www.youtube.com/@lfe_everest

Weekly Tech Sync - Join the developers and start contributing
   Every Tuesday 10am -11am CET, meeting link via mailing list