LF ENERGY

EVerest

EVerest: AC and DC electric vehicle charging with open source software and hardware

me

Kai-Uwe Hermann @hikinggrass

background in computer science and robotics

working @ PIONIX on EVerest since 2021

How to charge a car?

- Basic AC charging
  - at home: portable charger, wall mounted charging station up to 11/22kW
  - or in public with RFID/App authentication
- “smart” AC charging (not a lot of cars support that yet)
  - ISO 15118-(2/20), Plug&Charge
  - upcoming: bi-directional AC charging
- DC charging
  - DIN SPEC 70121, ISO 15118
  - highway fast chargers
  - smaller units at home, think DC-DC solar etc.
  - upcoming: bi-directional DC charging

What is EVerest?

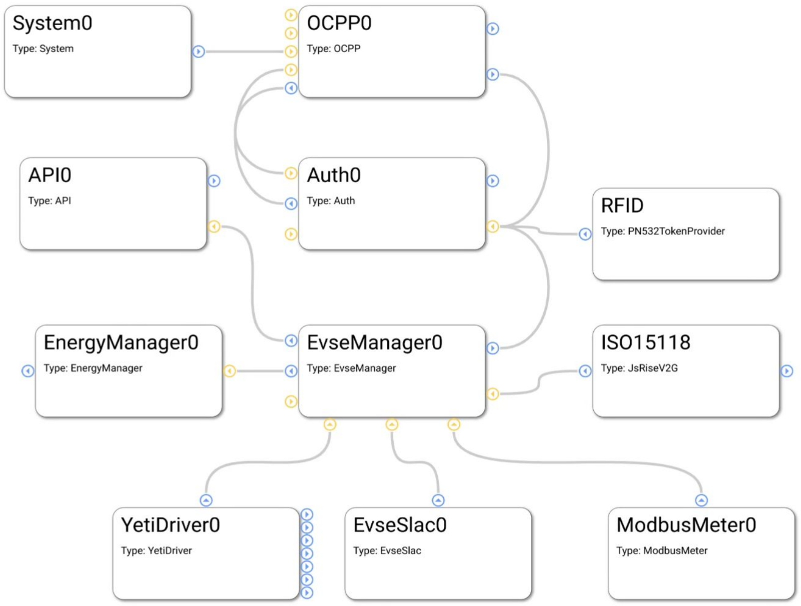
- A complete software stack for EV chargers
- Runs on embedded Linux
- **Apache 2.0 License**
- Aims to support many different HW platforms
    - **and you can build your own!**

LFENERGY

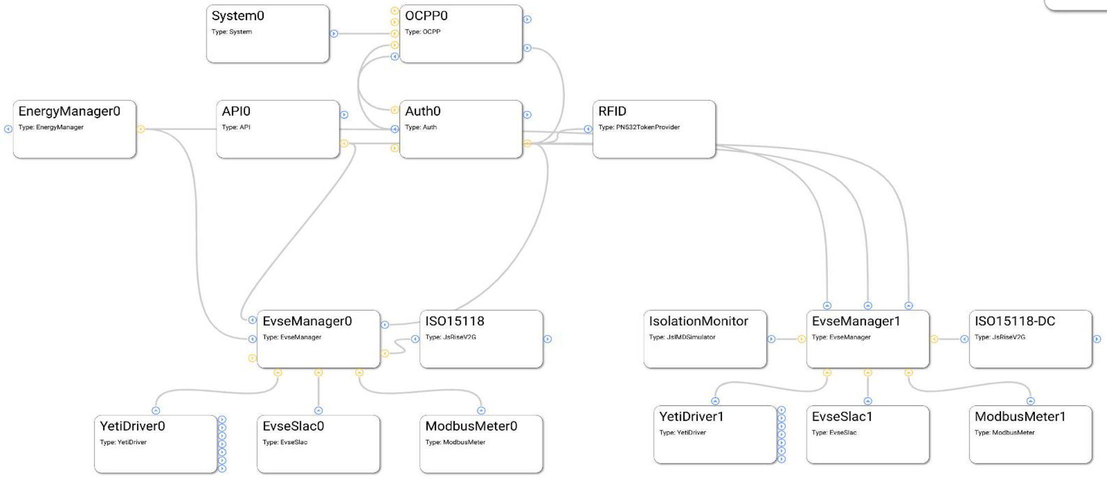
EVerest

EVerest features

- Modular architecture
- Graphical setup web interface
- Energy management



Simple AC or DC Charger

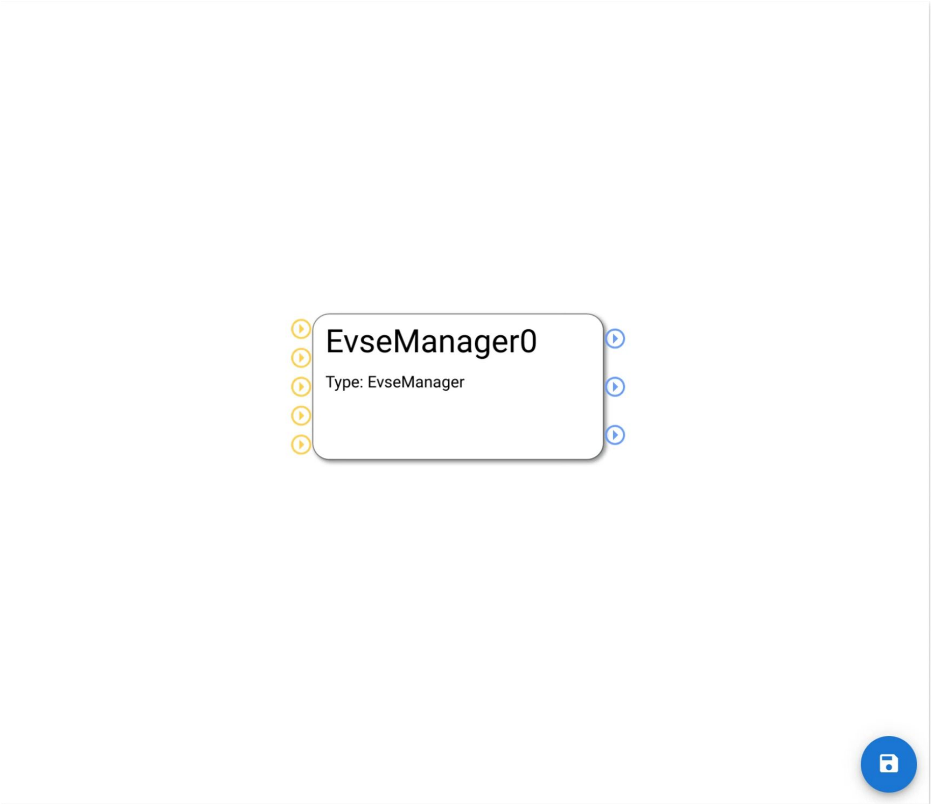


AC+DC Hybrid Charger

EVerest features - Build a simple AC Charger Step 1 of 6

EvseManager

- One charging connector
- Charging logic and session
- Orchestrates all other modules access to this one connector



EvseManager0

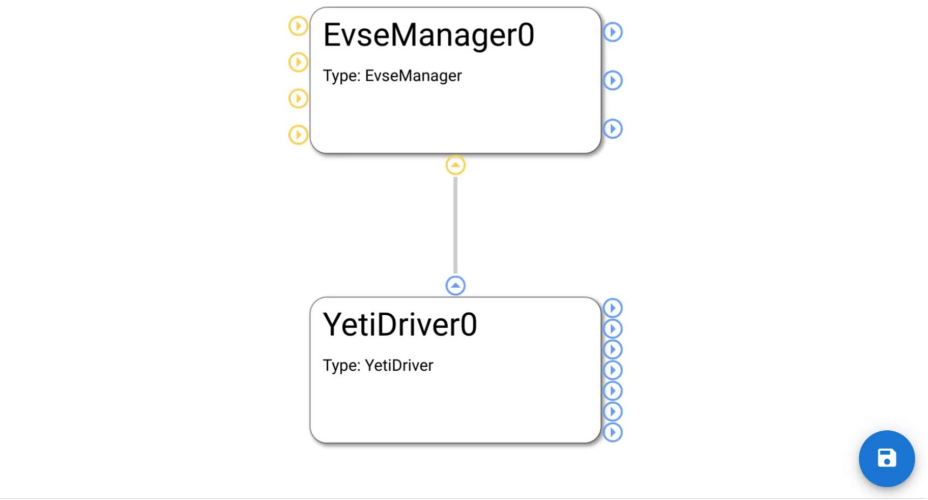
Type: EvseManager

EVerest features - Build a simple AC Charger
Step 2 of 6

Board support

Hardware driver:

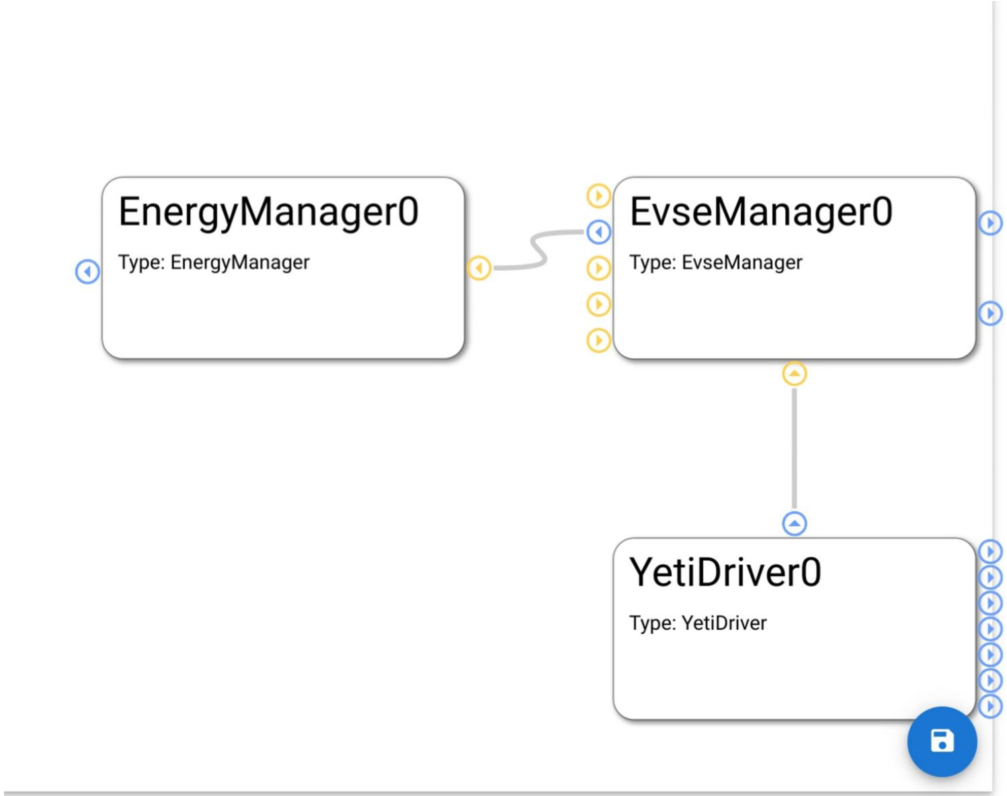
- CP, Relais, RCD



EVerest features - Build a simple AC Charger
Step 3 of 6

Energy manager

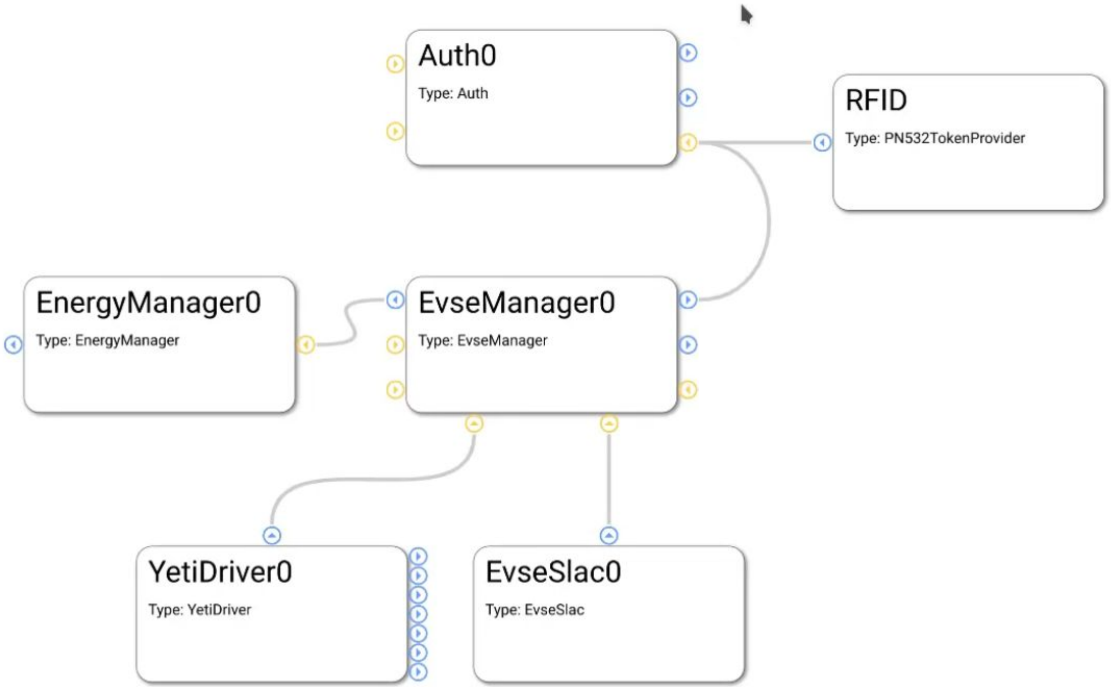
- Minimal configuration, more advanced later



EVerest features - Build a simple AC Charger Step 4 of 6

Auth manager needs

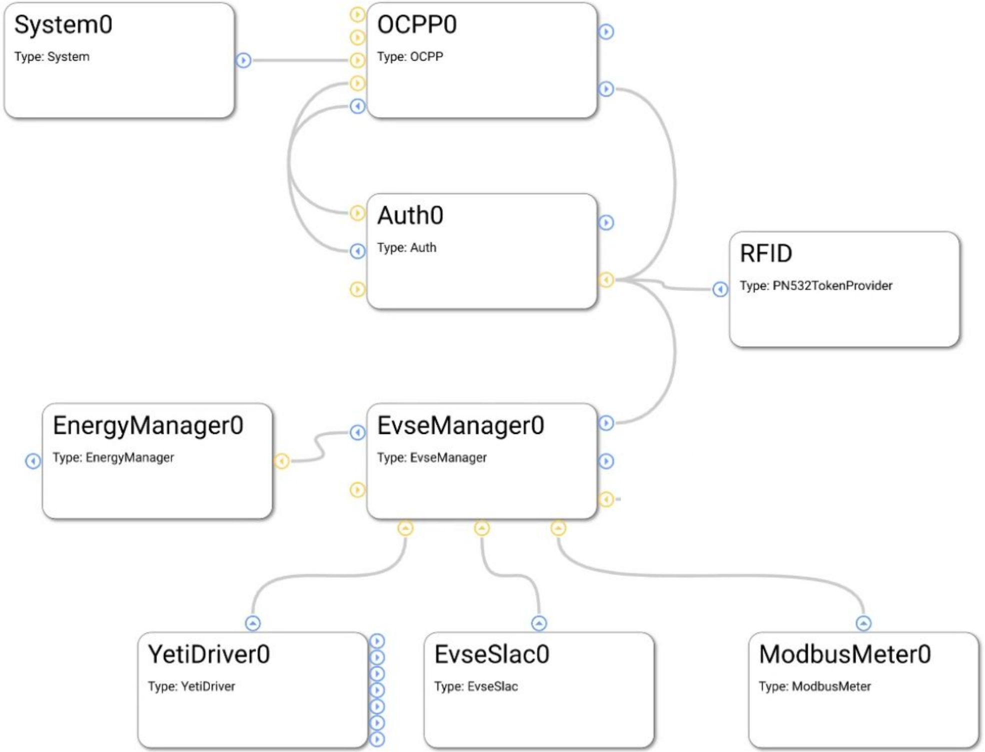
- Token providers (output tokens) and token validators (can check if token is valid)
- We add two token providers:
- RFID (new module)
- Autocharge (EvseManager also has a token provider interface for EVCCID)



EVerest features - Build a simple AC Charger Step 5 of 6

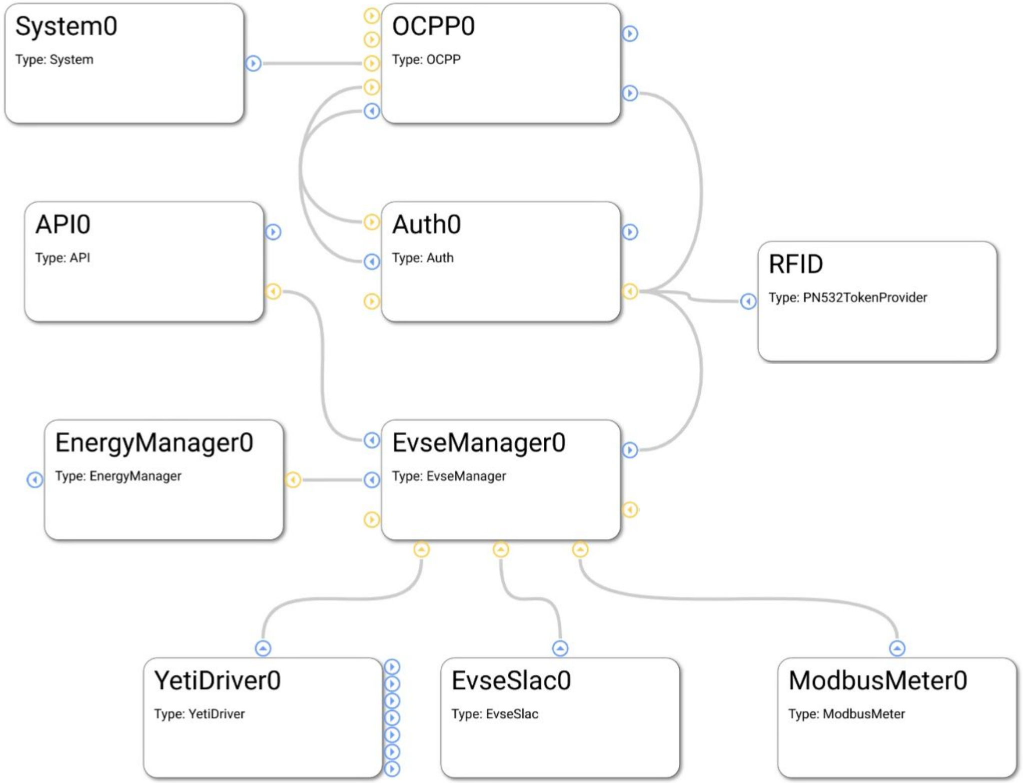
Add Cloud backend

- OCPP 1.6J module
- Powermeter
- System module supports reboot/firmware update etc via OCPP



EVerest features - Build a simple AC Charger Step 6 of 6

Add API for ext. applications

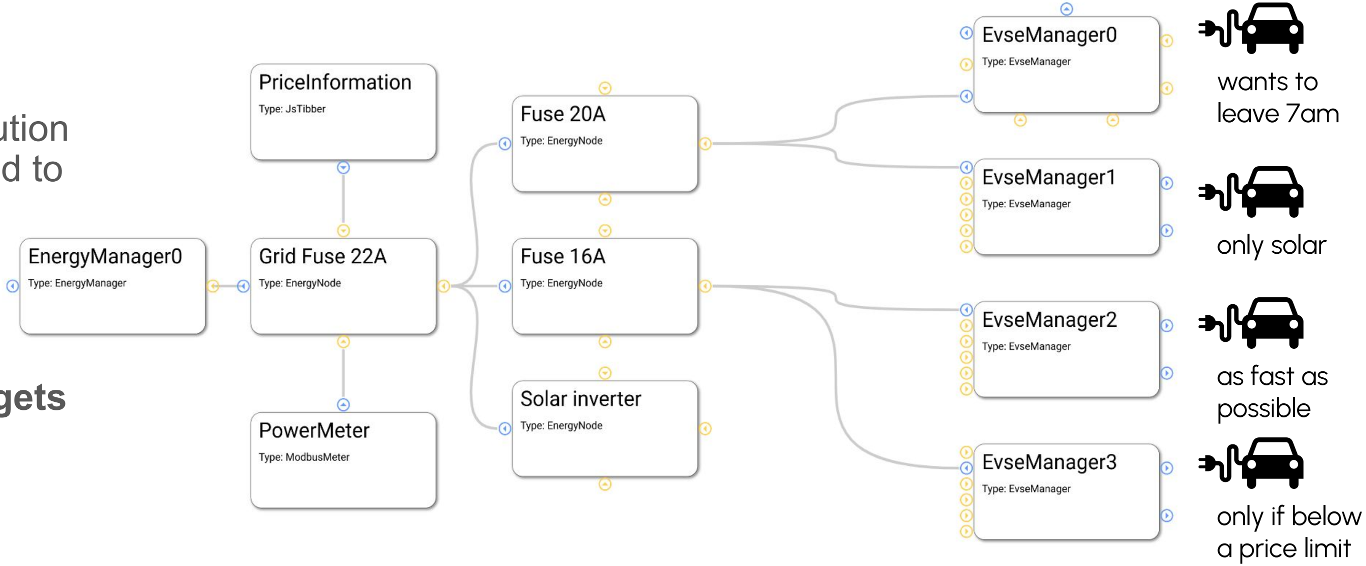


EVerest features - Configure energy management

Energy manager

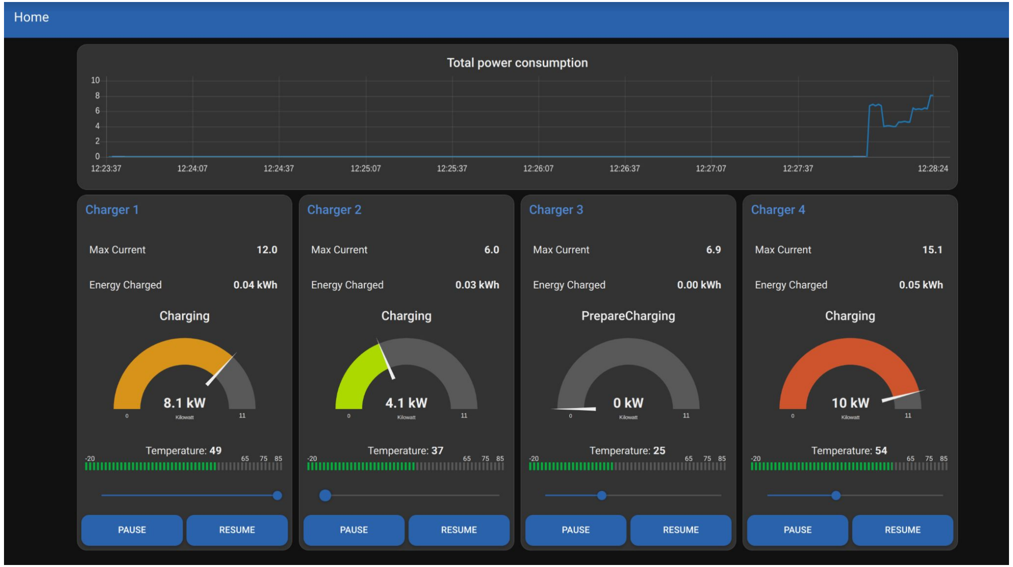
Complex energy distribution trees can be represented to **load balance multiple charging stations.**

Load sharing will work across the tree with **different optimizer targets** for each car.



EVerest features

- Software- and hardware in the loop simulation (IEC61851, ISO15118)
- Lots of protocols:
  - OCPP 1.6 (2.0.1 coming soon)
  - ISO 15118 (AC+DC)
  - DIN SPEC 70121
  - IEC 61851 / SAE J1772
  - ModBus
  - Sunspec
  - MQTT
- Multiple language bindings: C++17, Python, JavaScript



Charger 1

Max Current 12.0

Energy Charged 0.04 kWh

Charging

8.1 kW

Temperature: 49

Charger 2

Max Current 6.0

Energy Charged 0.03 kWh

Charging

4.1 kW

Temperature: 37

Charger 3

Max Current 6.9

Energy Charged 0.00 kWh

PrepareCharging

0 kW

Temperature: 25

Charger 4

Max Current 15.1

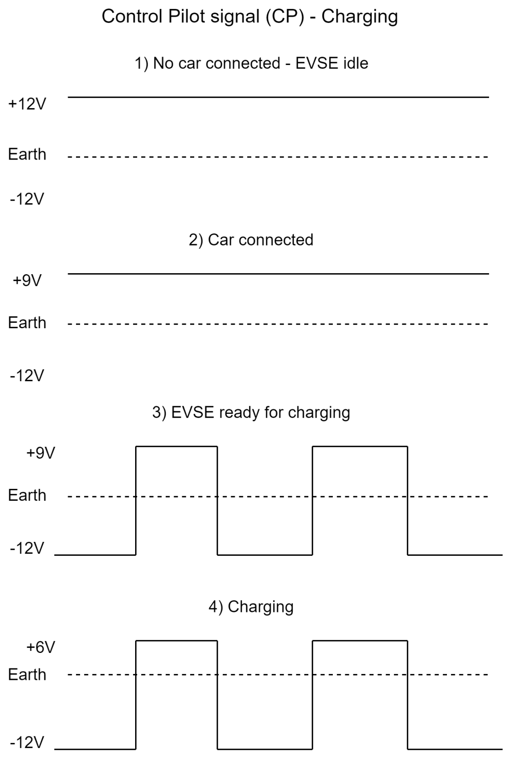
Energy Charged 0.05 kWh

Charging

10 kW

Temperature: 54

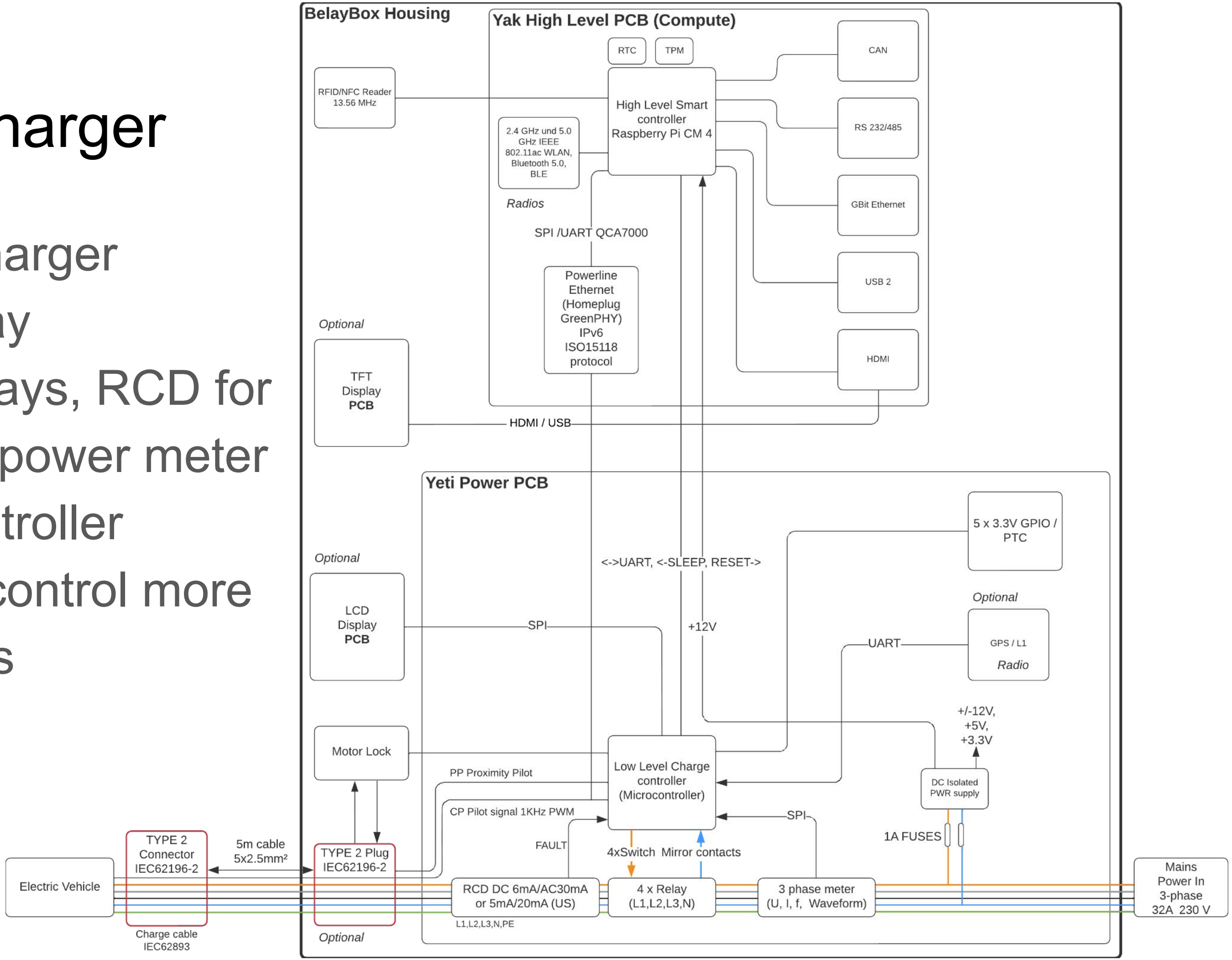
Basic PWM Charging


Control Pilot signal (CP) - Charging
1) No car connected - EVSE idle
+12V
Earth
-12V
2) Car connected
+9V
Earth
-12V
3) EVSE ready for charging
+9V
Earth
-12V
4) Charging
+6V
Earth
-12V

- CP signal: +-12V signal, 1 kOhm impedance
- Car can lower positive part
  by adding load resistors and a diode to
  - 9V (connected)
  - 6V (wants to charge)
  - 3V (charge with ventilation)
- 0V: Car fault or no power
- -12V: EVSE fault
- Duty cycle encodes available current
  (typically 6A to 32A)

Building a AC charger

- Not a battery charger just a smart relay
- Powerpath: Relays, RCD for safety, optional power meter plus a microcontroller
- Linux board to control more advanced things



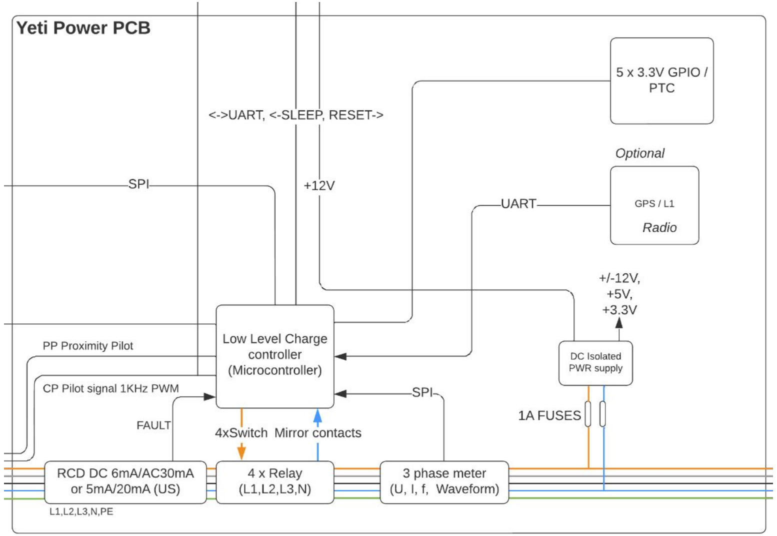
Open Hardware: Yeti & Yak

You wouldn't download a car (charger) - or would you?

- https://github.com/PionixPublic/reference-hardware
- **CERN Open Hardware Licence Version 2 - Permissive**
- Developer friendly
- Designed with KiCAD 6 https://www.kicad.org
- Case design files for 3D printing available

Yeti

- 22kW AC 3 phase power board



Yeti Power PCB

<->UART, <-SLEEP, RESET->

5 x 3.3V GPIO / PTC

Optional

GPS / L1

Radio

SPI

+12V

UART

+/-12V, +5V, +3.3V

PP Proximity Pilot

Low Level Charge controller (Microcontroller)

DC Isolated PWR supply

CP Pilot signal 1KHz PWM

SPI

FAULT

4xSwitch

Mirror contacts

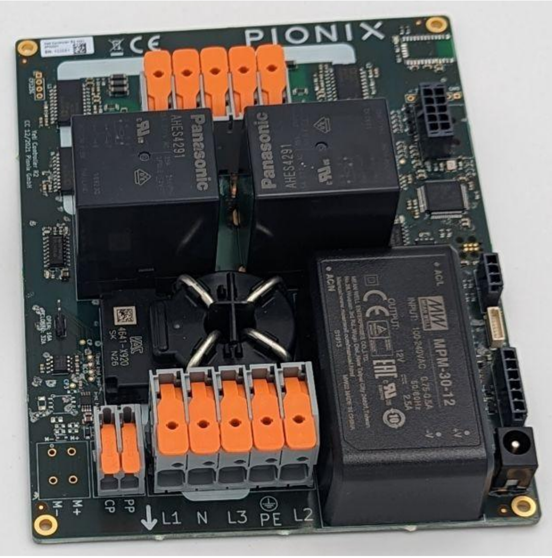
1A FUSES

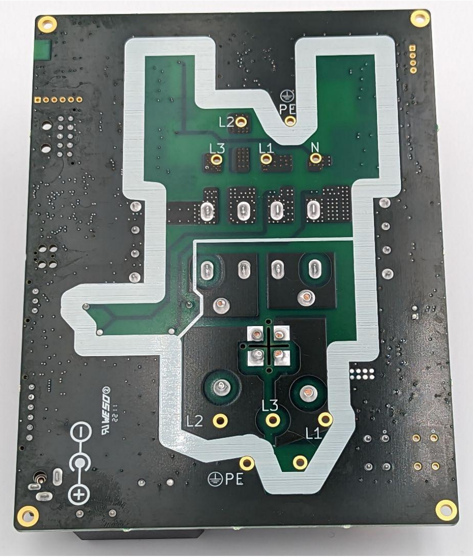
RCD DC 6mA/AC30mA or 5mA/20mA (US)

4 x Relay (L1,L2,L3,N)

3 phase meter (U, I, f, Waveform)

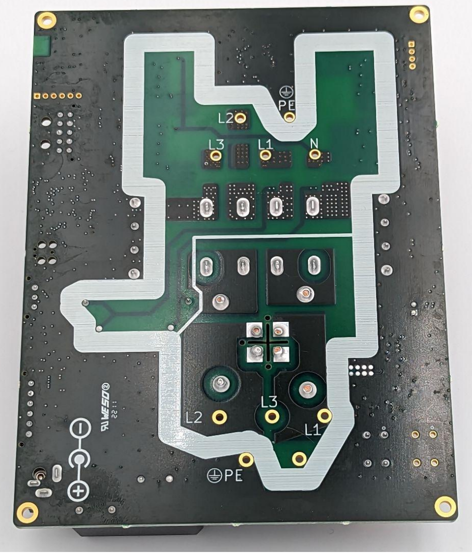
L1,L2,L3,N,PE





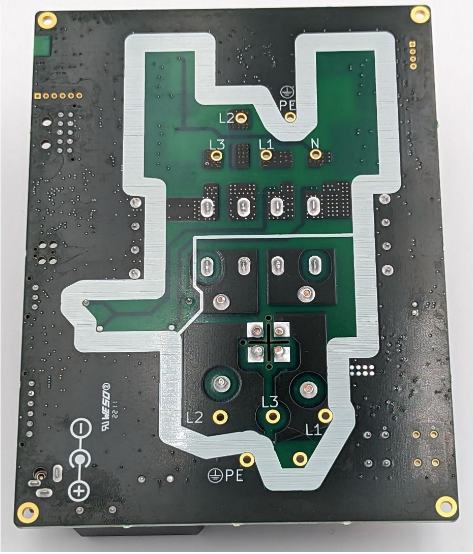
Yeti features

- Control Pilot (CP) signal generation
  in compliance with IEC61851-1/SAE J1772
- CP signal sampling in sync with PWM
- Onboard 35A relay for three phase power switching
  - with mirror contacts for welding detection
- Onboard 3ph power metering
  - Supports up to 8 Khz sampling
  - Measures voltages, currents, power and frequency of all phases plus neutral
- RCD module
  - For 5/6mA DC ground faults and 20/30mA AC faults
  - Outputs measured leakage current for telemetry



Yeti features (continued)

- 10 pin connector for high level board with control UART
  - Connection to the Yak board
- External connector for a small SPI LCD
  - Useful if used as a standalone charger
- External RS485/ModBus
  - For external power meters etc.
- External GPIOs
- Power input:
  - Onboard 110V/230V -> 12V supply (or external)
    Onboard ACDC can supply high level Yak board too!
- … and much more!
  https://github.com/PionixPublic/reference-hardware#yeti-features



Yeti micro controller firmware

- STM32 firmware for Yeti power board
- https://github.com/PionixPublic/yeti-firmware
- **Apache 2.0 License**
- Controls all devices on Yeti board
- All electrical safety relevant code encapsulated here
- Communicates via UART/Protobuf with EVerest

How to use Yeti

- **As a standalone charger**
- **As a power path for a smart charger**
- Automatic switching can be implemented to fall back to emergency free charging in case the higher level linux board fails

How to use Yeti - As a standalone charger

- Yeti is a complete AC charger for electric vehicles
  supporting IEC-61851-1 / SAE J1772 basic charging
- Contains the full charging logic, car charges immediately when connected
- UART to observe status with limited control over the charging session
- This mode is called "high level control" mode in the yeti firmware

How to use Yeti - As a power path for a smart charger

- Yeti firmware can be switched to "low level control" mode with a UART command
- Here the charging logic must be external, only the basic state machine for states A-F remains in the microcontroller
  - This basic state machine is essential for electrical safety
- An external board sets the PWM duty cycle and reads back the control pilot events
- This is the mode used by EVerest to also enable HLC (high level charging, ISO 15118-(2/20) / DIN SPEC 70121)

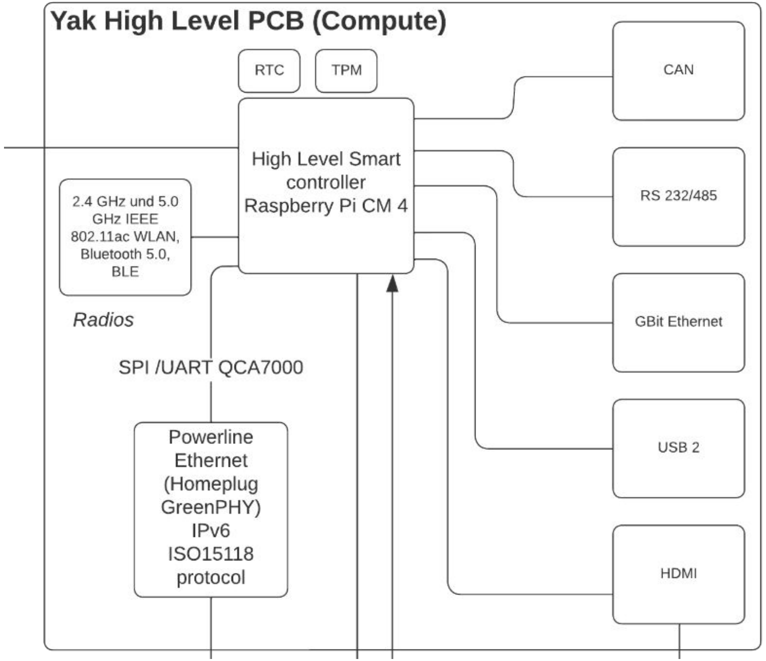
HLC High Level Charging

Uses Powerline communication on top of CP PWM signal
(on the same wire, Homeplug GreenPHY standard)

1) Setup logical network beween EVSE and car using SLAC
2) Assign IPv6 link local addresses on both sides
3) Car sends UDP broadcasts to find EVSE
   EVSE replies with its IP and port number
4) TCP/TLS connection from car to EVSE
5) ISO 15118 protocol through that TCP connection
   (XML data encoded with EXI)

Yak

- High level control board
- Runs EVerest on (embedded) Linux



Yak High Level PCB (Compute)

RTC

TPM

High Level Smart controller Raspberry Pi CM 4

2.4 GHz und 5.0 GHz IEEE 802.11ac WLAN, Bluetooth 5.0, BLE

Radios

SPI /UART QCA7000

Powerline Ethernet (Homeplug GreenPHY) IPv6 ISO15118 protocol

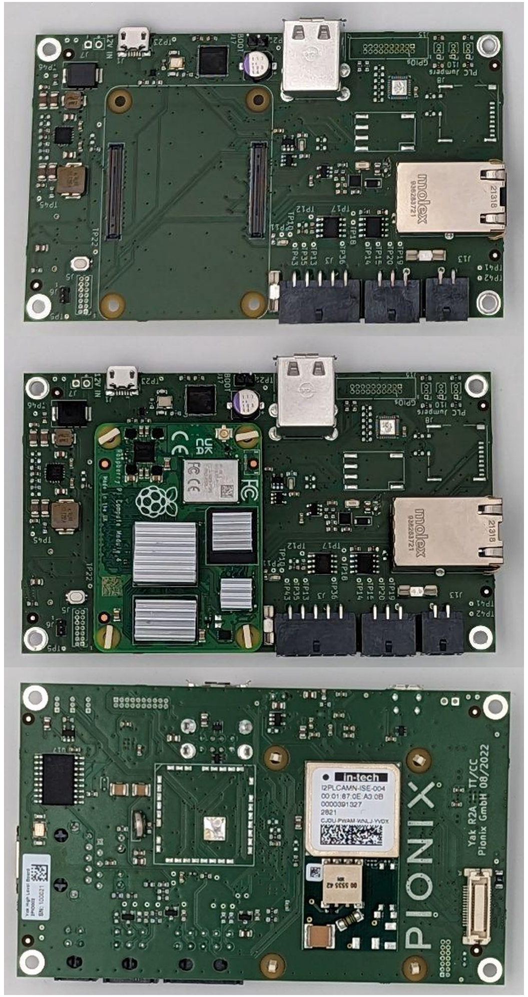
CAN

RS 232/485

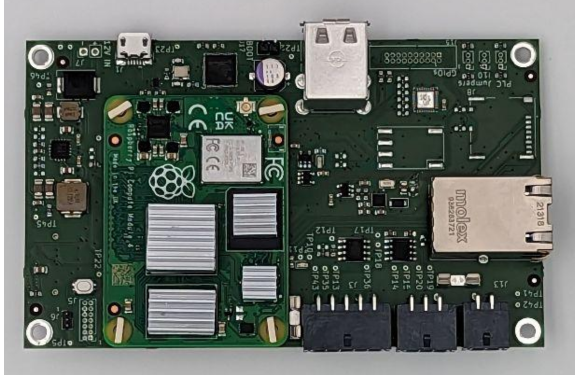
GBit Ethernet

USB 2

HDMI

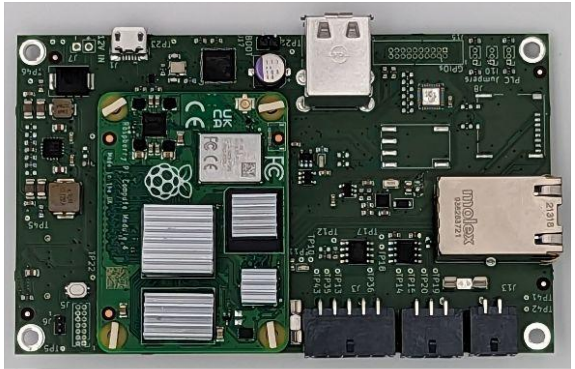


Yak features



- Raspberry Pi Compute Module 4 (CM4)
- 10 pin connector for direct connection to Yeti board
- RTC with backup battery
- PLC GreenPhy modem for HLC communication with car
- UART/power connector for popular PN532 based RFID modules
- RS485/ModBus
- CAN bus
- Ethernet, Wireless LAN 2.4Ghz/5Ghz and Bluetooth 5
- USB ports
- USB client port to flash the EMMC of CM4
- External GPIOs
- https://github.com/PionixPublic/reference-hardware#yak

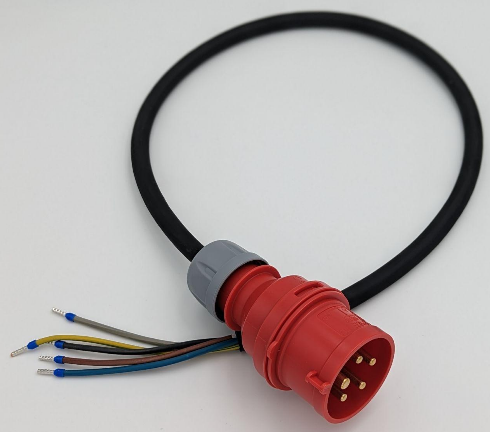
Putting everything together

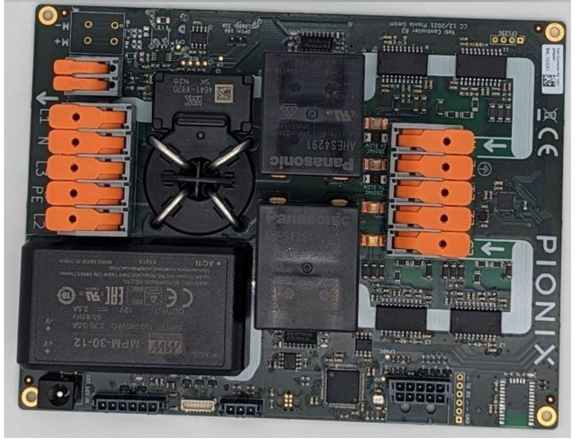


Yak high level control board

IEC 62196-2 Type 2 connector

Mains 3 Phase Power In



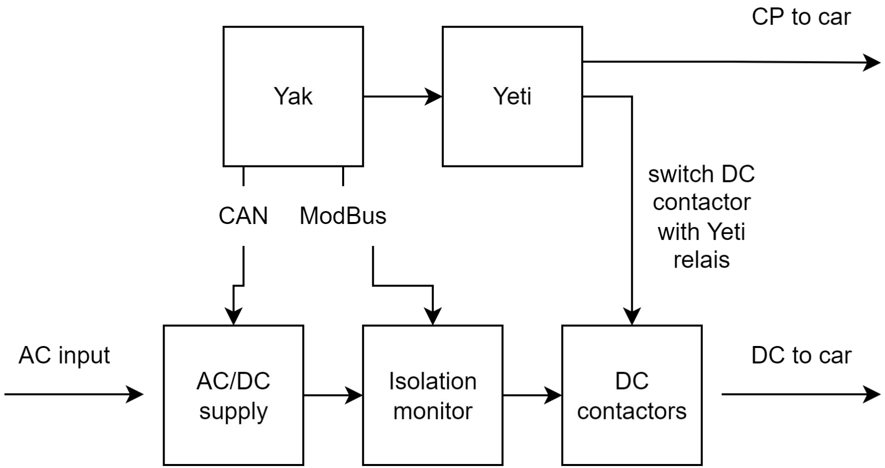


Yeti power board

Coming Soon: DIY Bidirectional DC

Yak board already prepared for everything that you need for DC communication

“just” needs some DC power electronics, isolation monitor



stay tuned for more this summer!

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