

openvehicles / **Open-Vehicle-Monitoring-System-3** Public

<> **Code** Issues 163 Pull requests 21 Actions Projects Security and quality

28 Branches 37 Tags Go to file Go to file <> Code

markwj Updated (pure ota, no factory, maximized store) partition table

014e4be · 8 hours ago

client	ServerV3 client: TLS support,...	2 years ago
docs	Fix filter log capture docume...	3 weeks ago
flyer	Docs: added multilingual quic...	4 years ago
logo	Added OVMS logo artwork	7 years ago
plugin	New plugin: Tasmota Smart ...	3 years ago
plugins	Remove dead abrp plugin (fr...	2 weeks ago
vehicle	Updated (pure ota, no factory...	8 hours ago
.gitignore	Completely Rewritten Wake ...	6 years ago
.gitmodules	convert wolfssl to a git subm...	3 years ago
.readthedocs.yaml	Explicitly list requirements for...	3 years ago
.travis.yml	CI: Add Travis-CI build script ...	6 years ago
LICENSE	Port over LICENSE	9 years ago
README.md	Docs: update README vehic...	2 months ago

README License

Open-Vehicle-Monitoring-System-3 (OVMS3)



Introduction

The OVMS is an **all open source** vehicle remote monitoring, diagnosis and control system.

The system provides **live monitoring** of vehicle metrics like state of charge, temperatures, tyre pressures and diagnostic fault conditions. It will **alert** you about critical conditions and events like a charge abort, battery cell failure or potential theft. Depending on the vehicle integration it allows you to **take control** over the charge process, climate control, tuning parameters of the engine and more. OVMS developers are enthusiasts trying hard to provide as **detailed information** about the internals of a vehicle as possible.

While most new vehicles now include some kind of remote control system, very few allow deep insight, some will not work in all regions and none will give you **control over your personal data**. The OVMS fills that gap, and also enables you to add all these features to existing vehicles of any kind.

The OVMS can also be used for **fleet monitoring**. It allows a fleet manager to not only track the vehicle locations but also to monitor the vehicle's vitals, remotely check for fault conditions, offer services like automatic preheating for users and take active control in case of abusive use. As the system is open source and fully scriptable, it can easily integrate custom access and control systems.

For **developers and technicians**, the OVMS includes a range of CAN tools including multiple logging formats, a configurable OBD2 translator, a DBC decoder, a reverse engineering toolkit and a CANopen client. The module provides SSH access and WebSocket streaming and can stream and inject CAN frames via TCP. Both the module and the web frontend can be customized by plugins. The module has three builtin CAN buses and [can be extended by a fourth one](#).

The **OVMS base component** is a small hardware module that connects to the vehicle **OBD2** port. The standard kit in the current version 3.3 includes a **2G/3G/4G modem** to provide cellular connectivity and **GPS**. This standard kit has been **FCC certified** and **CE certified**.

The module provides a **built-in Web App** user interface and remote control via native cellphone Apps available for **Android** and **iOS**. It integrates into home/process automation systems via **MQTT** and provides data logging to SD card and to a server.

Vehicle Support

- *Native Integration*
 - [BMW i3 / i3s](#)
 - [BYD Atto 3](#)
 - [Cadillac 2nd gen CTS](#)
 - [Cadillac CT5](#)
 - [Chevrolet Bolt EV / Opel Ampera-e](#)
 - [Chevrolet C6 Corvette](#)
 - Chevrolet Volt / Opel Ampera
 - [Energica motorbikes](#)
 - [Fiat 500e](#)
 - [Fiat e-Doblo](#)
 - [Hyundai Ioniq vFL](#)
 - [Hyundai Ioniq 5](#)
 - [Jaguar Ipace](#)
 - [Kia e-Niro / Hyundai Kona / Hyundai Ioniq FL](#)
 - [Kia Soul EV](#)
 - Livan Maple 60s
 - [Maxus eDeliver 3](#)
 - [Maxus Euniq 5 6-seats](#)
 - Maxus Euniq 6
 - [Maxus T90 EV](#)
 - [Mercedes-Benz B250E](#)

- [MG EV](#)
- [Mini Cooper SE](#)
- [Mitsubishi Trio \(i-MiEV et al\)](#)
- [Nissan Leaf / e-NV200](#)
- [NIU MQi GT EVO / GT 100](#)
- [Renault Twizy](#)
- [Renault Zoe / Kangoo](#)
- [Renault Zoe Phase 2](#)
- [Smart ED Gen.3](#)
- [Smart ED/EQ Gen.4 \(453\)](#)
- [Tesla Model S](#)
- [Tesla Roadster](#)
- Think City
- [Toyota RAV4 EV](#)
- [VW e-Golf](#)
- [VW e-Up / Skoda Citigo / Seat Mii](#)
- *General Support*
 - [DBC File Based](#)
 - [GPS Tracking](#)
 - [OBD-II Standard](#)
 - Zeva BMS
 - [ZombieVerter VCU](#)

Links

- *User Resources*
 - *User and Developer Guides: (hint: version selection in left menu at the bottom)*
 - [Stable release \(OTA version "main"\)](#)
 - [Latest nightly build \(OTA version "edge"\)](#)
 - [User Support Forum](#)
 - [Android App APK file](#)
 - [iOS App](#)
 - [Home Assistant integration](#)
- *Distributors*
 - *Original Design*
 - [Medlock & Sons \(North America\)](#)
 - [OpenEnergyMonitor \(UK/Europe\)](#)

- *Customized Versions*
 - [Optimized compact casing not only for Smart ED/EQ \(Germany/Europe\)](#)
- *Servers*
 - [Asia-Pacific](#)
 - [Germany/Europe](#)
- *Developers*
 - [Developer Guide](#)
 - [Developer Mailing List & Archive](#)
 - [Server Source](#)
 - [Android App Source](#)
 - [iOS App Source](#)

Hardware



- [Module Schematics and PCB Layouts](#)
- **Base Module (v3.0-3.3)**
 - Black injection-moulded plastic enclosure, approximately 99x73x29 mm excl. plugs
 - ESP32 WROVER processor (16MB flash, 4MB SPI RAM, 520KB SRAM, dual core 160/240MHz Xtensa LX6 processor)
 - WIFI 802.11 b/g/n
 - Bluetooth v4.2 BR/EDR and BLE
 - 3x CAN buses
 - 1x Micro USB connector (for flash download and serial console)
 - 1x Micro SD card slot
 - 1x Internal expansion slot

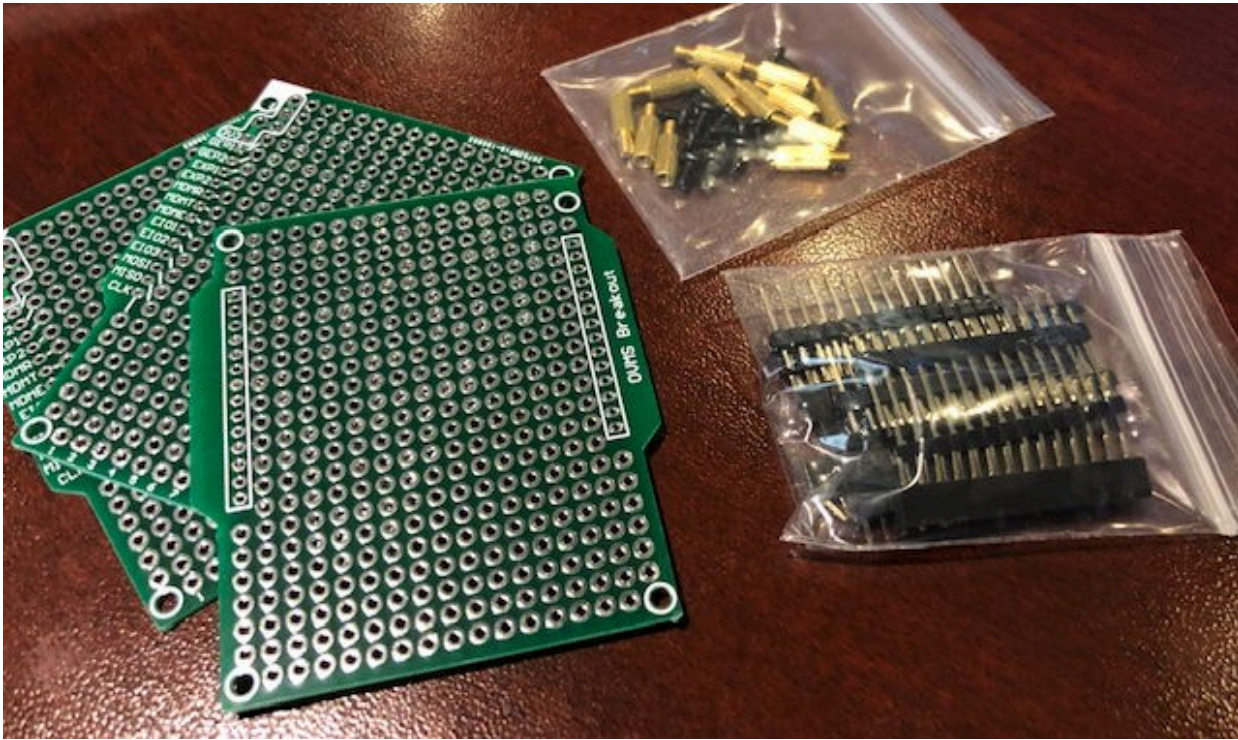
- 8x EGPIO, 2x GPIO
 - 1x GSM antenna connector
 - 1x GPS antenna connector
 - 1x DB9 vehicle connector
 - 1x DB26 expansion connector
- **Modem Module v3.0-3.2**
 - US edition is SIM5360A (Dual-Band UMTS/HSPA+ 850/1900MHz, Quad-Band GSM/GPRS/EDGE 850/900/1800/1900MHz)
 - EU edition is SIM5360J(E) (Dual-Band UMTS/HSPA+ 900/2100MHz, Quad-Band GSM/GPRS/EDGE 850/900/1800/1900MHz)
 - 3G (EV-DO/HSPA+) dual band modem
 - Includes 2G (GSM/GPRS) and 2.5G (EDGE) quad band
 - GPS/GNSS
 - Nano (4FF) SIM slot
 - HOLOGRAM.IO nano sim included (can be exchanged if necessary)
 - **Modem Module v3.3**
 - World edition: SIM7600G (Multi-Band LTE-FDD/LTE-TDD/HSPA+ and GSM/GPRS/EDGE)
 - 4G (LTE-FDD/LTE-TDD) multi band modem
 - Includes 3G (EV-DO/HSPA+), 2G (GSM/GPRS) and 2.5G (EDGE) quad band
 - GPS/GNSS
 - Nano (4FF) SIM slot
 - HOLOGRAM.IO nano sim included (can be exchanged if necessary)

Extensions

The external DB26 DIAG connector provides access to the three CAN buses and offers some free extension ports. The internal PCB expansion connector allows stacked additions of further modules and serves for routing GPIO ports to the external DIAG connector. See schematics for details.

A very nice first extension module has been developed by Marko Juhanne: [OVMS-SWCAN](#)

If you plan on developing a hardware extension or just want to do some custom adaptations, have a look at our prototyping PCB kit. It's available in packs of 3 PCBs and includes headers and mounting material:



If the kit isn't available at the distributors, please contact Mark Webb-Johnson mark@webb-johnson.net.

Development and Contributions

New developers are very welcome on any part of the system, and we will gladly provide any help needed to get started.

The purpose of this project is to get the community of vehicle hackers and enthusiasts to be able to expand the project. We can't do it all, and there is so much to do. What we are doing is providing an affordable and flexible base that the community can work on and extend.

Everything is open, and APIs are public. Other car modules can talk to the server, and other Apps can show the status and control the car. This is a foundation that others hopefully will interface to and and build upon.

If you'd like to contribute, please accept our code of conduct:

- Introduce yourself on the developer mailing list
- Be kind & polite
- Understand the framework concepts
- Ask if you need help
- Present your plans if in doubt
- Write decent code
- If you extend modules, stick to their code style

- Write brief but descriptive commit comments
- Add user level descriptions to the change history
- Provide documentation in the user guide
- Use pull requests to submit your code for inclusion

"Vibe" submissions will be rejected and closed without comment.

If using "AI" tools, it is *your* responsibility to validate all "AI" output you decide to include in your submission, or decide to base your work or suggestions on, *prior* to the publication.

Submitting unvalidated "AI" output is a waste of our time and an insult to our dedication to the Open Source idea of sharing knowledge and helping each other to grow.

Don't fall for the hype. Generative "AI" does not understand or reason. It produces output that *looks* similar to answers to comparable questions it was trained with.

LLM learn patterns and correlations, not reasoning or causalities. *If* (!) given enough training material, and *if* (!) given enough context memory to hold all relevant aspects, the output *may* be useful to get an *idea* about an issue. Not less, not more.

As of 2025/2026 no currently available "AI" tool is capable of producing factually correct or consistent code, fault analysis or even documentation for the OVMS framework. Current "AI" tools do not even get the most basic function call analysis right, let alone code generation for complex custom environments like this.

Still none of the current AI tools will inform you about its limitations and cease to perform an order it cannot fulfill. They will rather produce something that looks like a professional and plausible result, that on closer inspection then turns out to be complete nonsense.

Examples:

- [AI not understanding STL containers or OVMS event processing](#)
- [AI not understanding code coverage and suggesting model change to fix view issue](#)

A note on pull requests:

Pull requests shall focus on one specific issue / feature / vehicle at a time and shall only mix vehicle specific changes with framework changes if they depend on each other. If changes are not or only loosely related, split them into multiple PRs (just as you would do with commits).

Usage hint: create a branch for each pull request, include only those commits in that branch (by cherry-picking if necessary) that shall be included in the pull request. That way you can push further commits to that branch, Github will automatically add them to an open pull request.

Donations

The OVMS is a non-profit community project. Hardware production and service can normally be financed by sales, but some things (e.g. prototype development and certifications) need extra money. To help the project, you can make a donation on the OVMS website:

<https://www.openvehicles.com/forum>

Please also consider supporting the vehicle developers directly. Check out their web sites and support addresses for their respective donation channels.

Thank you!

License

The project includes some third party libraries and components to which their respective licenses apply, see component sources for details.

The project itself is published under the MIT license:

Copyright (c) 2011-2020 Open Vehicles

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT

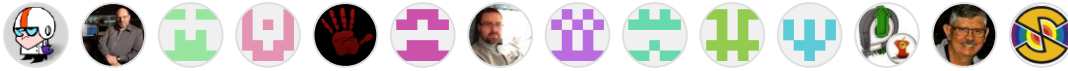
Releases

 37 tags

Packages

No packages published

Contributors 92



[+ 78 contributors](#)

Languages



- C 58.6%
- C++ 37.6%
- JavaScript 1.1%
- HTML 1.0%
- Python 0.9%
- Makefile 0.3%
- Other 0.5%