**CAR MONITOR**

**Candidat: Horvath Paul Șerban**

**Coordonator științific: Ș.l. Dr.Ing. Lucian Prodan**

Sesiunea: Iunie 2024

Contents

[REZUMAT 4](#_Toc164282000)

[Abstract 4](#_Toc164282001)

[1. INTRODUCERE 4](#_Toc164282002)

[1.1 Aspecte generale - PROBLEM STATEMENT, masina folosita, protocoale, abrevieri, specificatiile proiectului etc. 4](#_Toc164282003)

[1.2 SOTA - State of the art, studiu de piata 4](#_Toc164282004)

[1.3 ESP-32 4](#_Toc164282005)

[1.3.1 Modulul integrat Bluetooth 4](#_Toc164282006)

[1.3.2 Modulul integrat WiFi 4](#_Toc164282007)

[1.3.3 Dependinta SPIFFS 4](#_Toc164282008)

[1.3.4 Dependinta WebAsyncServer 4](#_Toc164282009)

[1.4 ELM327 OBDII - scurta descriere protocol OBDII 4](#_Toc164282010)

[1.5 Git repo - modul de gestionare 4](#_Toc164282011)

[1.6 ELMduino 4](#_Toc164282012)

[2. ARHITECTURA 4](#_Toc164282013)

[2.1 Diagrama de decizii generica 4](#_Toc164282014)

[2.2 Arhitectura Bluetooth 4](#_Toc164282015)

[2.2.1 Diagrama de decizii detaliata 4](#_Toc164282016)

[2.3 Arhitectura WebAsyncServer 4](#_Toc164282017)

[2.3.1 Diagrama de decizii detaliata 4](#_Toc164282018)

[3. IMPLEMENTARE DETALIATA 4](#_Toc164282019)

[3.1 Implementare Bluetooth 4](#_Toc164282020)

[3.1.1 Diagrama de secventa dinamica - Din avion 4](#_Toc164282021)

[3.1.2 Diagrama de decizii detaliata 4](#_Toc164282022)

[3.1.3 Diagrama de secventa pentru rpm() - Se adauga si detalii de returnari, etc. 4](#_Toc164282023)

[3.1.4 Diagrama de secventa pentru kph() - Se adauga si detalii de returnari, etc. 4](#_Toc164282024)

[3.1.5 Diagrama de secventa pentru fuel() - Se adauga si detalii de returnari, etc. 4](#_Toc164282025)

[3.1.6 Diagrama de secventa pentru oil() - Se adauga si detalii de returnari, etc. 4](#_Toc164282026)

[3.2 Implementare WebAsyncServer 5](#_Toc164282027)

[3.2.1 Diagrama de secventa dinamica - Din avion 5](#_Toc164282028)

[3.2.2 Diagrama de decizii detaliata 5](#_Toc164282029)

[3.2.3 Diagrama de secventa pentru WebServer.begin() 5](#_Toc164282030)

[3.2.4 Diagrama de secventa pentru send data 5](#_Toc164282031)

[3.2.5 Diagrama de secventa pentru SPIFFS open file 5](#_Toc164282032)

[3.2.6 Diagrama de secventa pentru grafice 5](#_Toc164282033)

[4. DESFASURARE EXEMPLIFICATA 5](#_Toc164282034)

[5. DEZVOLTARE SI TESTARE 5](#_Toc164282035)

[5.1 Detalii de configurare - configurare config.h 5](#_Toc164282036)

[5.2 Securitate 5](#_Toc164282037)

[5.3 Stocarea datelor 5](#_Toc164282038)

[5.4 Interfata cu utilizatorul 5](#_Toc164282039)

[5.5 Posibili clienti 5](#_Toc164282040)

[5.6 Performanta sistemului - cata memorie ocupa, frecventa de operare, de returnare a datelor etc. 5](#_Toc164282041)

[5.7 Scalabilitate tehnica - se pot adauga mai multe esp uri 5](#_Toc164282042)

[5.8 Scalabilitate antreprenoriala - posibil pret, specificiatii, pachet, venit anual etc (vezi GreenSpot) 5](#_Toc164282043)

[6. Concluzie 5](#_Toc164282044)

[6.1 Obiective indeplinite 5](#_Toc164282045)

[6.2 Comparare cu SOTA 5](#_Toc164282046)

[6.3 Adaugari ulterioare 5](#_Toc164282047)

[Referinte 5](#_Toc164282048)

# REZUMAT

Lucrarea de față are ca scop utilizarea unui sistem alcătuit dintr-o placă dedicată proiectelor de tip IoT (Internet of Things), și anume ESP-32, placă furnizată de către compania Espressif, folosind framework-ul Espressif IoT Development Framework (esp-idf). Aceasta are rolul de a fi un nod care se ocupă de preluarea datelor în timp real de la un autoturism, aflat în staționare sau în mers, prin intermediul unui convertor OBD (On board diagnostics) versiunea II, dar și de găzduirea unui server web, cu scopul de a afișa parametrii acestui vehiculului în timp real. Placa ESP-32 este programată într-o manieră în care să poată trimite cereri de citire al parametrilor (turația motorului, viteza instantă, consum, temperatura uleiului, consumul instant de carburant, temperatura lichidului de răcire și presiunea din galeria de admisie) către un convertor conectat la portul OBD-II al vehiculului.

Convertorul, sau cititorul OBD-II, este furnizat de compania VGate, și este echipat cu un chip de tip ELM327, care permite transferul datelor primite de la ECU (Electronic Control Unit) al autoturismului prin portul OBD-II către ESP-32 prin Bluetooth. A fost ales acest dispozitiv datorită versatilității și compatibilității crescute cu alte dispozitive. Acest dispozitiv trimite cereri primite de la placa ESP-32 către ECU prin coduri standard numite PID (Parameter ID).

Mai departe, după request-ul primit, ECU va returna valoarea dorită către cititorul OBD-II (dotat cu ELM327), care la rândul lui va trimite valoarea către ESP-32. Acesta va afișa valoarea pe un server web, cu scopul de a afișa parametrii autoturismului într-o maniera user-friendly.

Placa ESP-32 este programată astfel încât să folosească funcții dintr-un API open-source, numit ELMduino, disponibil pe plăcile de tip ESP-32 Development board. Acest API are rolul de a forma și identifica PID-urile corespunzătoare, și de a le trimite prin Bluetooth către cititorul OBD-II. Pe de alta parte, este necesar ca placa sa fie conectată la o rețea pentru a putea găzdui cu succes serverul cu scopul afișării datelor. Această placă a fost aleasă datorită capabilităților WiFi si Bluetooth pe care le prezintă.

# Abstract

# INTRODUCERE

## **Aspecte generale**

Într-o lume în care informația înseamnă putere, fiecare dintre noi dorim să facem alegeri cât mai bune, bazate pe informațiile care ni se prezintă. Autovehiculele, pe de altă parte, cu toate că sunt folosite intens de fiecare dintre noi, pe întreg Pământul, sunt construite într-o manieră în care sa furnizeze doar strictul necesar de informații pe care o persoană obișnuită le poate procesa. Astfel, pentru o persoană cunoscătoare, fie că este un hobby sau este ocupația acesteia, simplele date furnizate de către un autovehicul pot să nu fie îndeajuns.

De aceea, sistemul prezentat în această lucrare, care poartă numele „Car Monitor”, se explică singur. Acest sistem se conectează la autovehicul, reușește să preia în timp real și afișează într-o interfață prietenoasă cu utilizatorul, parametrii mașinii, cu scopul de a monitoriza și de a lua decizii calculate, pe baza datelor reale primite de la aceasta.

Acest sistem poate fi folosit, pe de-o parte, de către persoane fizice, care doresc o monitorizare mai amănunțită asupra propriei mașini, în vederea depanării problemelor, defectelor, sau doar pentru simpla evaluare a stării acesteia. În cazul în care se observa un comportament ciudat al unui parametru al mașinii, se pot lua decizii care să prevină o daună mai mare, economisindu-se bani și resurse. De aceea, utilizând acest sistem, se poate anticipa ce componentă și când în viitor va trebui schimbată.

Pe de altă parte, acest sistem poate fi folosit și de către persoane juridice, mai exact firme, care au în dotare o flotă de mașini în vederea închirierii lor. Pentru a se asigura că mașinile sunt tratate corespunzător și nu sunt abuzate, acestea pot fi dotate cu sistemul „Car Monitor” care poate înregistra o gamă mai variată de parametrii în care a fost utilizată aceasta. În cazul în care se constată că vehiculul a fost abuzat, (de exemplu turația motorului au fost nejustificat de mare pentru un timp îndelungat), se poate percepe o taxă suplimentară suportată de client, care are ca scop achitarea reparațiilor ulterioare aduse mașinii, ca rezultat al utilizării necorespunzătoare.

În plus, acest sistem poate fi folosit chiar și de către service-urile auto, mai exact de către mecanici și ingineri auto, pentru a decide dacă mașina funcționează în parametrii optimi sau nu. De exemplu, în cazul unei inspecții tehnice periodice, necesară pentru orice vehicul care utilizează drumurile publice, sistemul „Car Monitor” poate afișa utilizând ECU al mașinii dacă noxele emise sunt în parametrii normali sau nu. În acest caz, sistemul „Car Monitor” ar trebui sa fie o unealtă de confirmare și inspectorul să nu se bazeze exclusiv pe acesta, deoarece calculatorul mașinii poate returna valori false ale noxelor emise (comportament care poate fi accidental sau intenționat). În cazul în care ECU returnează valori false, exista posibilitatea de a se efectua o investigație pentru a vedea ce alte modificări au fost aduse mașinii, și daca acestea sunt conforme cu legislație în vigoare.

Important de menționat este faptul că parametrii afișați de către sistemul „Car Monitor” pot fi personalizați în funcție de fiecare caz.

În testarea acestui proiect s-a utilizat o mașină marca BMW Seria 1 E87, anul 2009, cu un motor de 2.0 litri benzină, ce dezvoltă 122 cai putere și un cuplu de 180 Newton metru, cutie de viteză manuală. De asemenea, în programarea plăcii, s-a folosit limbajul de programare C++ (care include metode speciale pentru , Arduino SDK, iar pentru interfață s-a utilizat Visual Studio Code cu extensia PlatformIO care asigură script-uri pentru flash-uirea plăcii, atât a codului cât și a fișierelor folosite pentru webserver.

De asemenea, s-au utilizat diferite dependințe și API-uri care pun la dispoziție funcționalitatea întregului sistem. Cele mai importante dintre acestea fiind ESP WebAsyncServer, SPIFFS și ELMduino, care vor fi prezentate în cele ce urmează.

## SOTA - State of the art, studiu de piață

Scopul acestui subcapitol este de a prezenta diferitele tehnologii care există în prezent pe partea de monitorizare al autovehiculelor și de a avea o vedere de ansamblu asupra altor dispozitive care se asociază sistemului „Car Monitor”.

Istoria diagnozei mașinilor datează de la începutul secolului XX, când autovehiculele încă erau relativ simple, iar sistemele de monitorizare electronice abia debutau. Depanarea problemelor se făcea manual, unde mecanicii se bazau pe setul de cunoștințe proprii, necesare pentru interpretarea diferitelor sunete, vibrații, al fumului etc. În 1968, Volkswagen primul computer de board, acesta fiind analog. În anul 1980, General Motors introduce pe modelele Cadillac Eldorado și Seville, sistemul de diagnoză unde erorile (Diagnostic Trouble Codes) sunt afișate în ecranul de climatizare când vehiculul se afla în modul de depanare, iar în 1981 GM introduce sistemul “Computer Command Control” capabil de a comunica cu ECU pentru a iniția o cerere de diagnoză [1].

În 1988 Societatea Inginerilor Automotive (SAE) recomanda un conector și un set de erori standard și astfel se introduce standardul OBDI care este necesar tuturor mașinilor vândute în California din acest an [1].

În anul 1994, CARB (California Air Resource Board) încep demersurile pentru o nouă versiune de OBD, numit OBDII. Acesta include recomandările din OBDI făcute de SAE, iar în 1996 devine un standard obligatoriu pentru mașinile vândute în Statele Unite ale Americii [1].

În 2001, Uniunea Europeana adopta standardul OBDII, numit EOBD (European OBD) care întâi este necesar mașinilor cu combustibil pe benzina, iar în 2004 EOBD se introduce și pe mașinile cu combustibil pe motorină [1].

În prezent, există numeroase metode și tehnologii de a monitoriza parametrii unui vehicul, toate având scopuri ușor diferite unele față de altele. Pe de-o parte există produsele comerciale care au ca grup țintă mecanicii profesioniști din service-urile autorizate. Aceștia în mod uzual folosesc produse și tehnologii mai scumpe (și deseori mai inaccesibile publicului larg) care permit conectarea la portul OBDII în mod direct prin fir. Scopul acestora este să identifice toate erorile mașinii care au apărut de-a lungul timpului și să ajute mecanicul să ia o decizie informată ținând cont de acestea. Aceste erori se pot șterge din memoria calculatorului de bord, prin resetarea stării regiștrilor care stochează erorile respective. Aici, cele mai folosite sunt dispozitivele produse de Autel, SnapOn Diagnostic Tool sau XTOOL.

Pe de altă parte, pe nișa clienților de uz general, există produse care sunt mai accesibile din punct de vedere al prețului, dar precizia datelor returnate (date care se modifică în timp real) este mai slabă. Conectivitatea wireless apare la această categorie deoarece elimină nevoia de un hardware fizic care să asigure transmisia datelor. De aceea, datele sunt transmise OTA (over the air) prin intermediul Bluetooth către un receptor, în care în cele mai multe din cazuri este un smartphone capabil sa afișeze într-o aplicație instalata. În cazurile care seamănă cu proiecte DIY (Do It Yourself), există dispozitive care au încorporat un display LCD pe care se face afișajul precum și senzori plasați în zonele de interes ale mașinii.

În continuare se prezintă un tabel în care se realizează o comparație între sistemele de monitorizare profesionale, sistemele destinate publicului larg care se folosesc de o aplicație creată special în acest sens și sistemele de monitorizare de tip DYI.

Tabel 1: Comparație între metodele actuale de monitorizare

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Cost | Dificultate utilizare | Scalabilitate | Fiabilitatea datelor | Functionalități |
| Scanner OBDII de bază | Scăzut | Ușor (prin aplicație) | Mică | Mică | Diagnoză de bază |
| Scanner OBDII avansat | Mare | Greu | Mică | Mare | Diagnoză avansată |
| Sistem DYI | Mediu | Mediu | Mică | Medie | Monitorizarea depinde de senzorii plasați de utilizator |
| Car Monitor | Scăzut | Mediu | Mare | Medie | Monitorizarea parametrilor în timp real (disponibili în API) + diagnoză de bază al DTC |

Aspectul diferențiator al proiectului propus îl reprezintă afișarea datelor pe un webserver dedicat și nu într-o interfață care este disponibilă doar pe dispozitivul conectat la cititor. Acest lucru restricționează monitorizarea și forțează utilizatorul să se afle în proximitatea mașinii, indiferent daca se folosește transmisia fizica (prin cablu) sau wireless (prin Bluetooth). Sistemul „Car Monitor” poate fi scalat ușor astfel încât monitorizarea să se facă de la distanță, iar webserverul să fie găzduit remote. Astfel, operatorul care monitorizează mașina (sau flota de mașini) poate observa parametrii de la sediul său.

## ESP-32

Scris în C++, foloseste arduino ide, dar în cazul de fata s-a folosit platformio, etc

### Modulul integrat Bluetooth

### Modulul integrat WiFi

### Dependinta SPIFFS

### Dependinta WebAsyncServer

### Dependinta ELMduino

## ELM327 OBDII - scurta descriere protocol OBDII

Un PID este un șir de bytes, unde fiecare byte are o însemnătate. De exemplu, pentru citirea vitezei actuale ale vehiculului vom trimite 010D. Citirea oricărui parametru se face prin byte-ul 01 (frecvent utilizat în această lucrare), urmat de parametrul dorit (în acest caz 0D). Pentru turația motorului avem PID-ul 0C, pentru consumul de combustibil avem 5E, pentru temperatura uleiului între 01 si 1F, depinzând de producătorul vehiculului, pentru temperatura lichidului de răcire avem 05 iar pentru presiunea galeriei de admisie se va referi prin valoarea 0B.

## Git repo - modul de gestionare

# ARHITECTURA

## Diagrama de decizii generica

## Arhitectura Bluetooth

### Diagrama de decizii detaliata

## Arhitectura WebAsyncServer

### Diagrama de decizii detaliata

# IMPLEMENTARE DETALIATA

## Implementare Bluetooth

### Diagrama de secventa dinamica - Din avion

### Diagrama de decizii detaliata

### Diagrama de secventa pentru rpm() - Se adauga si detalii de returnari, etc.

### Diagrama de secventa pentru kph() - Se adauga si detalii de returnari, etc.

### Diagrama de secventa pentru fuel() - Se adauga si detalii de returnari, etc.

### Diagrama de secventa pentru oil() - Se adauga si detalii de returnari, etc.

## Implementare WebAsyncServer

### Diagrama de secventa dinamica - Din avion

### Diagrama de decizii detaliata

### Diagrama de secventa pentru WebServer.begin()

### Diagrama de secventa pentru send data

### Diagrama de secventa pentru SPIFFS open file

### Diagrama de secventa pentru grafice

# DESFASURARE EXEMPLIFICATA

# DEZVOLTARE SI TESTARE

## Detalii de configurare - configurare config.h

## Securitate

## Stocarea datelor

## Interfața cu utilizatorul

## Posibili clienți

## Performanta sistemului - câtă memorie ocupa, frecventa de operare, de returnare a datelor etc.

## Scalabilitate tehnica - se pot adauga mai multe esp uri

## Scalabilitate antreprenoriala - posibil pret, specificiatii, pachet, venit anual etc (vezi GreenSpot)

# Concluzie

## Obiective îndeplinite

## Comparare cu SOTA

## Adăugări ulterioare

# Referințe

|  |  |
| --- | --- |
| [1] | „Wikipedia OBDII & OBDII PIDS,” 9 March 2024. [Interactiv]. Available: https://en.wikipedia.org/wiki/OBD-II\_PIDs. |
| [2] | PowerBroker2, „Github,” [Interactiv]. Available: https://github.com/PowerBroker2/ELMduino/tree/2c242bcf28a05ebc46c48f650b16611452b04695/examples. |
| [3] | „CarScanner,” 2023. [Interactiv]. Available: https://www.carscanner.info/choosing-obdii-adapter/. |
| [4] | D. Manstetten, F. Beruscha, H.-J. Bieg, F. Kobiela, A. Korthauer, W. Krautter și C. Marberger, „The Evolution of Driver Monitoring Systems: A Shortened Story on Past, Current and Future Approaches How Cars Acquire Knowledge About the Driver's State,” *ResearchGate,* pp. 54-60, October 2020. |
| [5] | Dashy, „The Evolution of Vehicle Diagnostics: From Manual Troubleshooting to Smart Devices,” 8 July 2023. [Interactiv]. Available: https://medium.com/@DashyAuto/the-evolution-of-vehicle-diagnostics-from-manual-troubleshooting-to-smart-devices-69a57e18a0a2. |
| [6] | „On board diagnostics,” 17 March 2024. [Interactiv]. Available: https://en.wikipedia.org/wiki/On-board\_diagnostics#History. |
| [7] | „Wikipedia ESP32,” 12 March 2024. [Interactiv]. Available: https://en.wikipedia.org/wiki/ESP32. |