|  |  |  |
| --- | --- | --- |
|  | **UNIVERSITATEA TEHNICĂ „Gheorghe Asachi” din IAȘ FACULTATEA DE AUTOMATICĂ ȘI CALCULATOAR**  **DOMENIUL: Calculatoare și Tehnologia Informației SPECIALIZAREA: Tehnologia Informației** |  |

LUCRARE DE LICENȚĂ

**Diagnosticarea unui automobil prin intermediul protocoalelor de comunicare OBD**

Candidat – **LIVIU RĂȚOI**

Profesor Coordonator – **Conf. Univ. Dr. Ing. Andrei STAN**

#### Iași, iulie 2023

**DECLARAȚIE DE ASUMARE A AUTENTICITĂȚII LUCRĂRII DE DIPLOMĂ**

Subsemnatul Rățoi Liviu, legitimat cu C.I. seria NZ nr. 181498, CNP

5000224270826 autorul lucrării DIAGNOSTICAREA UNUI AUTOMOBIL PRIN

INTERMEDIUL PROTOCOALELOR DE COMUNICARE OBD elaborată în vederea susținerii examenului de finalizare a studiilor de licență organizat de către Facultatea de Automatică și Calculatoare din cadrul Universității Tehnice „Gheorghe Asachi” din Iași, luând în considerare conținutul Art. 34 din Codul de etică universitară al Universității Tehnice „Gheorghe Asachi” din Iași (Manualul Procedurilor, UTI.POM.02 – Funcționarea Comisiei de etică universitară), declar pe proprie răspundere, că această lucrare este rezultatul propriei activități intelectuale, nu conține porțiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislației române (legea 8/1996) și a convențiilor internaționale privind drepturile de autor.

Data Semnătura

08.07.2023



Cuprins

[Rezumat 5](#_bookmark0)

[Introducere 6](#_bookmark1)

1. [Materiale și metode 7](#_bookmark2)
   1. [Materiale 7](#_bookmark3)
   2. [Metodologie 8](#_bookmark4)
2. [Comunicații teoretice 9](#_bookmark5)
   1. [Diagnosticarea 9](#_bookmark6)
      1. [Diagnosticarea Off-Board 9](#_bookmark7)
      2. [Diagnosticarea On-Board (OBD – On Board Diagnostics) 11](#_bookmark15)
   2. [Protocolul OBD–I 11](#_bookmark18)
   3. [Protocolul OBD–II 13](#_bookmark25)
      1. [Definiții și termeni utilizați in cadrul OBD-II 14](#_bookmark30)
      2. [SAE J1850 - Pulse Width Modulation (PWM) 14](#_bookmark31)
      3. [SAE J1850 – Variable Pulse Width (VPW) 15](#_bookmark33)
      4. [ISO 9141-2 – K-Line (similar RS-232) 15](#_bookmark35)
      5. [ISO 14230 - Key Word Protocol 2000 (KWP2000) 16](#_bookmark37)
      6. [ISO 15765 – Controller Area Network (CAN) 16](#_bookmark39)
   4. [Comunicația CAN 17](#_bookmark42)
      1. [Structura unui mesaj CAN 19](#_bookmark47)
   5. [Comenzi ce au ca scop cererea de date 20](#_bookmark49)
      1. [Comenzi de tip AT (Attention) 20](#_bookmark50)
      2. [Comenzi de tip OBD2 20](#_bookmark51)
   6. [Semnificația codurilor de eroare OBD II 21](#_bookmark52)
   7. [Tipuri de erori 22](#_bookmark53)
   8. [Coduri de eroare OBD II 22](#_bookmark54)
3. [Proiectarea aplicației 23](#_bookmark55)
   1. [Tehnologii, framework-uri și limbaje de programare 23](#_bookmark56)
      1. [Limbajul de programare 23](#_bookmark57)
      2. [Framework-uri 23](#_bookmark58)
   2. [Proiectarea hardware 23](#_bookmark59)
   3. [Proiectarea Software 25](#_bookmark64)
      1. [Conexiunea cu ELM327 26](#_bookmark65)
      2. [Schimbul de mesaje 27](#_bookmark66)
      3. [OBD II PID’s 28](#_bookmark67)
   4. [Interacțiunea dintre utilizator și interfata grafica 28](#_bookmark68)
   5. [Funcționalitatea codului 30](#_bookmark69)
4. [Testarea aplicației 35](#_bookmark70)
   1. [Modalitatea de pornire și utilizare a aplicației 35](#_bookmark71)
   2. [Simularea obținerii de date 35](#_bookmark72)
   3. [Trimiterea unui mesaj în cadrul aplicației 36](#_bookmark73)
   4. [Obtinerea de date în timp real 37](#_bookmark74)
5. [Concluzii 41](#_bookmark76)

[Bibliografie 42](#_bookmark77)

# REZUMAT

Prin prezenta lucrare s-a urmărit realizarea unei conexiuni printr-o comunicație serială între utilizator și calculatoarele autovehiculului (ECU) utilizând o interfață grafică.

Cu ajutorul unui dispozitiv electronic (ELM327 Bluetooth), s-a stabilit comunicarea dintre ECU și un dispozitiv de calcul mobil (smartphone, tabletă sau laptop) ce îndeplinește condițiile de comunicatie Bluetooth, cu scopul obținerii unor defecțiuni (DTC- Diagnostic Trouble Codes), de a modifica parametrii de funcționare sau de obținere a caracteristicilor precum viteza, rotațiile pe minut ale motorului, temperatura apei sau alți parametri specifici autovehiculului.

Acești parametrii au fost stocați într-o listă de tip coadă, împreună cu valorile acestora, urmând a se afișa grafic rezultatele obținute prin intermediul bibliotecii „matplotlib”, adăugând posibilitatea salvării rezultatelor în format JPEG/JPG, PNG, GIF, SVG, EPS sau TIFF.

Lucrarea este structurată în cinci capitole și o scurtă introducere, după cum urmează:

Introducerea expune multitudinea de exemple tragice din viața de zi cu zi în care sistemele de securitate ale autoturismelor esuează, motiv care m-a ambiționat să aleg această temă.

Primul capitol se axează pe detalierea conceptelor de ECU, OBD, ELM327, dar și metodologia diagnosticării unui autovehicul.

Al doilea capitol abordează noțiuni teoretice precum diagnosticarea, tipurile acesteia, instrumente de diagnoză, protocoalele OBD, comunicatia CAN, comenzi dar și semnificația acestora.

Al treilea capitol se focusează pe proiectarea aplicației, prezentarea tehnologiilor, a framework-urilor, modalitatea de proiectare software, explicații referitoare la structura interfeței, dar și funcționalitățile ale codului.

Al patrulea capitol cuprinde detalii legate de testarea și abordarea aplicației, modalitățile de testare cu sau fără conexiune la sistemul real de achiziție, pașii prin care trece un mesaj pentru a se obține parametrul final decodificat și obținerea graficelor în timp real.

În al cincilea capitol sunt prezentate concluziile rezultate în urma lucrării, dificultățile apărute, dar și direcțiile viitoare de dezvoltare.

# INTRODUCERE

***Motivația și scopul lucrării***

Socitatea umană se află într-o continuă schimbare și expansiune, iar prin aceste tendințe se pot remarca și modificări însemnate în industria auto, care în ultimii ani a cunoscut un progres considerabil mai ales în ceea ce privește evoluția sistemelor de securitate.

Prezența unui sistem de frânare automată în caz de urgență (Emergency Brake Assist – AEB) a devenit obligatoriu în industria producătoare de mașini începând cu anul 2020, deoarece s-a constatat că lipsa acestuia poate fi fatală în nenumărate situații. Acest risc major este influențat atât de viteza automobilului, cât și de senzorii de proximitate (radar), de poziționare sau acționare, de buna funcționare a sistemelor de control al stabilității (ESC) sau a altor sisteme ce contribuie la buna funcționalitate a AEB-ului. Fară un sistem de diagnosticare sau de citire a parametrilor în timp real, funcționalitatea acestui sistem ar fi redusă sau chiar inexistentă.

Studiind în domeniul Calculatoare și Tehnologia Informației și fiind atras de domeniul auto, am știut de la bun început că vreau să abordez o temă de actualitate, temă care să atragă atenția asupra necesității unor dispozitive care să citească corect semnalele vehiculului, deoarece motivul principal al funcționalității incorecte a sistemelor de securitate este cauzat de citirea eronată a semnalelor vehiculului, respectiv omiterea cazurilor neprevazute, precum condițiile meteo nefavorabile, caz în care sistemele duc la infuncționabilitate.

# MATERIALE ȘI METODE

## Materiale

#### Engine Control Unit (ECU)

Conceptul de control electronic în domeniul auto a fost implementat treptat de-a lungul timpului, ECU fiind unitatea computatională esențială ce stochează și manipulează informații referitoare la toți parametrii autovehiculului. Rolul principal al acestuia este de a asigura performanțe optime, consum minim de combustibil și reducerea noxelor emise.

Primele forme primitive de control electronic au fost implementate în anii 1960, tehnologia fiind mult mai subdezvoltată în acea perioadă. ECU având rolul de a permite buna funcționalitate a motorului [1].

În zilele noastre, un ECU îndeplinește roluri mult mai ample, cum ar fi sisteme de control ale stabilității (ESC - Electronic Stability Control), asistență șofer (ADAS – Advanced Driver Assistance System), acestea având unități proprii de tip ECU, fiind necesară o interoperabilitate între acestea. Comunicarea fiind stabilită printr-o arhitectură de rețea, cum ar fi CAN (Controller Area Network) și LIN (Local Interconnect Network).

#### OBD (On Board Diagnostic)

Pentru a monitoriza comunicația dintre ECU-urile specificate anterior, avem nevoie de un canal de comunicatie care să acceseze aceste magistrale. Pentru obținerea de date este folosit un conector standardizat cunoscut sub numele de OBD (On Board Diagnostic), utilizând interfața J1962.

Cu scopul interpretării acestor semnale oferite de conectorul OBD, în 1968 Volkswagen a introdus primul dispozitiv ce avea ca scop obținerea de informații referitoare la starea vehiculului.

Între anii 1980 și 1990 s-a utilizat standardul de diagnosticare OBD I, acesta având o utilizare diferită în funcție de producător și regiune. Capacitățile acestui standard sunt limitate, iar protocoalele fiind non-standardizate au fost folosite diferite tipuri de conectori și protocoale pentru realizarea comunicării, defectiunile fiind specifice fiecărui producător sau tip de automobil.[2]

Începând cu anii 1994 s-a introdus un nou standard mult mai performant, standard numit OBD II. Acesta, față de OBD I, a dezvoltat o conexiune cu protocoale de comunicare uniforme pentru toate vehiculele, coduri de defectiune comune, capacități multiple de diagnosticare și o monitorizare ce poate obține parametrii în timp real.

În multe țări și regiuni, OBD II a devenit un standard obligatoriu, impunând producătorilor ca vehiculele să fie echipate cu un astfel de conector de diagnosticare compatibil standardului OBD II, industria auto fiind mult mai ușoară din punct de vedere a diagnosticării și mentenanței vehiculelor.

#### Dispozitivul ELM327

Pentru efectuarea comunicațiilor în intermediul OBD II, cel mai frecvent se folosește

chipsetul ELM327. Acesta permite citirea și diagnosticarea sistemelor de control ale vehiculului.

Protocoalele CAN (Controller Area Network) suportate de ELM327 sunt:

-ISO 15765-4 (Can 11-bit și 29-bit),

-ISO 9141-2 (CAN care funcționează cu un singur fir), folosit pentru nivelul de transport și nivelul rețea (nivelele 4 și 3),

-ISO 14229-1 ce funcționează pentru nivelul sesiune și nivelul aplicație (nivelele 5 și 7)

-ISO 14230-4 (KWP2000 sau Keyword Protocol 2000), utilizat pentru vehicule europene

de generații vechi.

ELM327 funcționează și conform protocoalelor J1850:

- VPW (Variable Pulse Width)

-PWM (Pulse Width Modulation)

Spre deosebire de alte chipseturi folosite, ELM327 funcționează pentru o gamă largă de automobile, acaparând o piață mult mai mare față de alți competitori. Spre exemplu, protocolul propriu al companiei Volkswagen numit VCDS, respectiv VAG-COM, protocol ce se conformează doar vehiculelor ce fac parte din gama VAG. Posibilitatea de utilizare în cadrul altor tipuri de vehicule fiind incompatibilă

## Metodologie

Pentru realizarea diagnosticării unui automobil prin intermediul protocoalelor de comunicare OBD am parcurs două etape în studiul realizat în prezenta lucrare și anume consultarea bibliografiei de specialitate și a modelelor deja existente, respectiv realizarea unei conexiuni printr-o comunicație serială între utilizator și calculatoarele autovehiculului (ECU) utilizând o interfață ca parte practică.

Pentru început, am achiziționat un dispozitiv ELM327 v2.1 Bluetooth care permite accesul la ECU (Unitatea de Control Electronică) fără licența oferită de către producător. Am realizat conectarea modulului ELM327 la un laptop ce folosește sistemul de operare Ubuntu versiunea 23.04, urmând o utilizare a canalului de date rfcomm0 pentru schimbul de mesaje, comunicarea serială între dispozitive fiind datorată aplicației software CuteCom.

Avantajul acestei aplicații este capacitatea de a trimite mesaje fragmentate, permitând simultan obținerea unor răspunsuri criptate astfel încât mesajul inițial să fie în concordanță cu răspunsului primit. În această aplicație am introdus comenzi specifice de tip "AT" (Attention), respectiv comenzi de tip OBD.

În urma schimbului de mesaje (comenzi), am obținut caractere referitoare la protocolul folosit (pentru comenzi AT) sau un șir de octeți în formă hexadecimală, secvența de octeți fiind decodificată fie în caracterele asociate cu protocolul utilizat în comanda "AT", fie o valoare în kilometri pe oră (km/h) sau rotații pe minut (RPM), presiune de suprapunere (Boost) grade sau Celsius. Aceste valori reflectă informațiile obținute în timp real de la vehicul prin intermediul ECU.

Rezultatele obținute sunt stocate într-o listă de valori specifice fiecărui parametru, cu scopul de a genera un grafic care să prezinte un număr de valori obținute în anumite intervale de timp specificate în cadrul interfeței.

Această funcționalitate permite vizualizarea și analiza datelor într-un mod mai accesibil și ușor de înțeles, dar și monitorizarea performanțelor sau detectarea eventualelor probleme sau anomalii.

# COMUNICAȚII TEORETICE

## Diagnosticarea

Diagnosticarea reprezintă un set de operațiuni care vizează evaluarea stării unui sistem prin acțiuni ce nu necesită dezasamblarea sau pătrunderea forțată în interiorul sistemului respectiv.

Pe parcursul timpului s-a stabilit ca exista două tipuri de conexiuni:

-Diagnosticarea Off-Board

-Diagnosticarea On-Board

### Diagnosticarea Off-Board

Este realizată în general de către personal autorizat și include implementarea procesului de diagnosticare în afara vehiculului folosind dispozitive auxiliare (multimetre, osciloscoape, dispozitive de diagnoză).

**Multimetrul**: Folosit pentru masurători rapide de tensiune, curent si rezistentă, multimetrele includ de obicei urmatoarele funcții:

* DC V = diferite domenii de măsurare a tensiunii DC (mV, V)
* DC A = diferite domenii de măsurare pentru curent continuu (mA, A)
* AC V = diferite domenii de măsurare pentru tensiunea alternativă
* AC A = diferite domenii de măsurare pentru curent alterantiv
* Ω = diferite domenii de măsurare a rezistențelor.

**Osciloscopul:** Dispozitiv electronic de măsurare pentru observarea și măsurarea semnalelor de tensiune cu variații constante sau multiple ce sunt simulate și care evoluează discret.

În general, osciloscopul folosește un ecran reprezentat pe axa orizontală de timp, iar pe axa verticală reprezentând amplitudinea semnalului măsurat.



Figură 1: Aparat de masură si Osciloscop [3]

**Dispozitive de diagnoză:** reprezintă o interțata de comunicare intre un vehicul diagnosticat si specialistul diagnostician. Acest instrument poate indeplini si funcții de măsurare, funcții specifice osciloscopului sau multimetrului.

Dispozitivele de diagnoză indeplinesc o multitudine de funcții si roluri cum ar fi:

* Citirea codurilor de eroare
* Ștergerea codurilor de eroare
* Afișarea datelor in timp real
* Înregistrarea datelor
* Adaptarea pentru noi componente sau opțiuni
* Impunerea unor acțiuni
* Realizarea unor setari de bază

|  |  |  |
| --- | --- | --- |
| Figură 2: Model ELM327 cablu [2] | Reducere Obd v1.5 mini elm327 obd2 bluetooth auto scanner obdii 2 auto elm  327 tester de diagnosticare instrument pentru android, windows, symbian  g5i2 | en-gros ~ Confortresidenceploiesti.ro  Figură 3: ELM327 BT [4] | Figură 4: Diagnoză cu fir [2] |

Dispozitivele de diagnoză pot fi de 2 tipuri:

* Electrice: indica defectele din interiorul circuitului (circuit deschis, scurt-circuit la baterie sau scurt-circuit la masă).
* Funcționale: verifică funcționalitatea sistemelor, valorile parametrilor sau trimiterea de semnale eronate.

|  |  |  |
| --- | --- | --- |
| Figură 5: Tester multimarcă [5] | Interfata diagnoza auto gama Vag com 18.9.0 versiune in Limba Romana +  Engleza - eMAG.ro  Figură 6: Diagnoză VCDS [6] | Figură 7: Mega Macs PC Bike [7] |

#### Procedura de diagnoză:

Se foloseste un sistem de diagnosticare, fiecare aparținand unui suport software oferit de producator specific tipului de autoturism, caruia i se atașeaza conectorul adecvat respectând conexiunea intre pini.

* Se alege autovehiculul conform anului de fabricație, tipului de motorizare, a caroseriei si combustibilului adecvat;
* Se extrag informații legate de modul de realizare a conexiunii, tipul conectorului, amplasarea acestuia;
* Se realizează conexiunea dispozitivului si se trece la diagnosticarea propriu-zisa: identificarea modulului, citirea si ștergerea erorilor, testarea parametrilor functionali sau acționarea serviciilor

### Diagnosticarea On-Board (OBD – On Board Diagnostics)

Se realizează de către unitățile electronice ale automobilului, care sunt codate cu algoritmi specifici fiecărui vehicul și producător. Accesul pentru acest proces este realizabil în cazul cuplării contactului, dar și în cazul rulării automobilului. Conform imaginii de mai jos, șoferul primește mesaje prin intermediul martorilor MAL (Malfunction Indicator Lamp) sau prin intermediul unui display opțional ce poate afișa defectiuni sub formă de imagine sau text. Posibilitatea de a șterge acești martori se face cu ajutorul unei unități de diagnoză sau, în unele cazuri, se face prin remedierea problemei ce generează apariția respectivului martor.

|  |  |
| --- | --- |
| Figură 8: Martori [8] | Figură 9: Afișaj electronic [9] |

## Protocolul OBD–I

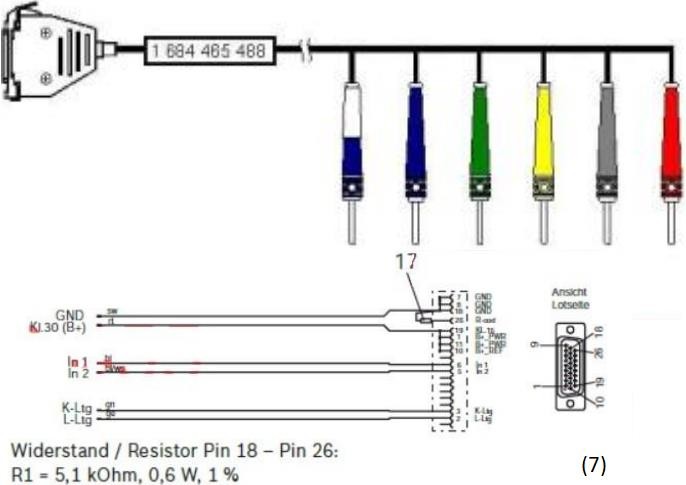
Incepând cu anul 1988, statul California, respectiv California Air Resources Board impune ca toate vehiculele vândute sa fie echipate cu sisteme de diagnoză On-Board specifice protocolului OBD-I.

Exemplu: BMW fiind unul dintre primii care s-au conformat acestui protocol, conectorul a fost conceput din 20 de pini numerotati de la 1 la 20 sub forma unui ceas cu două cadrane.

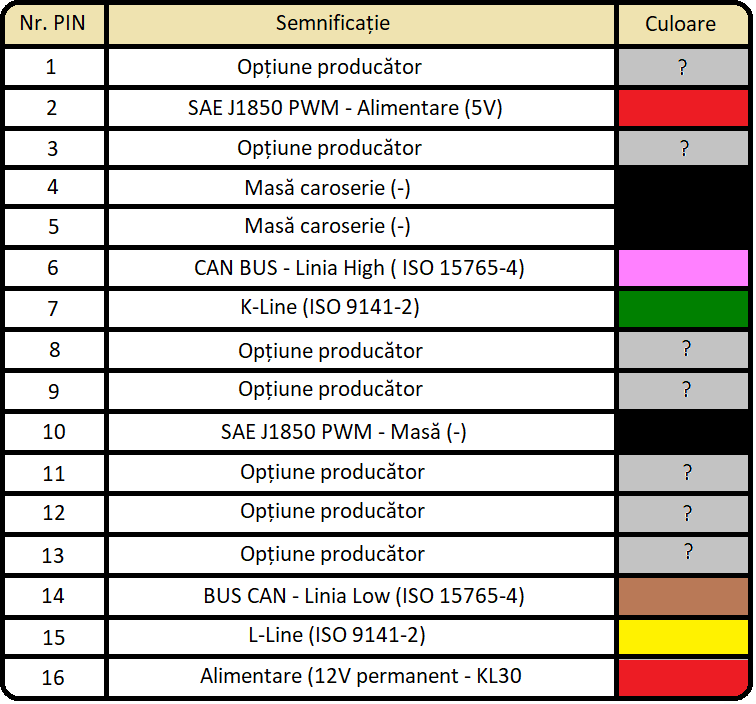
|  |  |
| --- | --- |
| Tabel 1: Pini conector BMW | Figură 10: Conector BMW [10] |

Printre companiile ce au adoptat protocolul OBD-I au fost si Renault (Fig 11), Mercedes-Benz (Fig 12), Ford (Fig 13) și Daewoo (Fig 14), iar legatura dintre firele de tip JACK si diagnoză se realiza asemena conectorului tip 1 684 465 488 din Fig 15.

|  |  |  |  |
| --- | --- | --- | --- |
| Figură 11: OBD-I  Renault [10] | Figură 12: OBD-I  Mercedes-Benz [10] | Figură 13: OBD-I Ford [10] | Figură 14: OBD-I  Daewoo [10] |

Figură 15: Conector VGA spre fiecare port [10]

## Protocolul OBD–II

General Motors a implementat in anul 1994 un successor al OBD I, venind cu funcționalitati noi, intr-o structură diferita de citire a codurilor de eroare si un port organizat diferit pentru a fi specific diagnosticării. Această modificare a influentat C.A.R.B si U.S.D.T în 1996 ca noile automobile din Statele Unite să fie dotate cu standardul OBD-II, urmând a se adopta această regulă în Uniunea Europeana, Australia, Noua Zeelanda și China [11].

Figură 16: Conector OBD- II [11]



Tabel 2: Listă pini OBD-II

Figură 17: Conectori OBD-II J1962

In figura 17 sunt prezentate tipurile de conectori OBD2 J1962. Acestea se împart în 2 categorii:

* Conector tip A: Compus din 8 pini superiori și 8 inveriori separați printr-un plastic discontinuu. OBD2 Tip A este utiliat mai frecvent pentru vehiculele americane, japoneze și coreene, suportând o tensiune de pană la 12V.
* Conector de tip B: Compus din 8 pini superiori și 8 inveriori separați printr-un plastic continuu. OBD2 Tip A este utiliat mai frecvent pentru vehiculele europene și suportă o tensiune de până la 24V.

Conectorul de tip A poate fi înlocuit de tipul B, datorită voltajului suportat, dar cel de tip B nu poate fi folosit din cauza voltajului și a separarii continue ce nu se potrivește cu pinii.

|  |  |
| --- | --- |
| Figură 18: Conector J1962 Type A [12] | Figură 19: Conector J1962 Type B [12] |

### Definiții și termeni utilizați in cadrul OBD-II

* Circuit deschis: Întrerupere a unui circuit electric ce este monitorizat.
* Semnal înafara limitelor: Circuitul funcționeaza în conditii normale, dar valoarea semnalul este eronata și nu împlinește condițiile de funcționare.
* Semnal peste limita maximă: Calculatorul de injectie citește o limită peste cea maximă admisă unui senzor sau actuator.
* Semnal sub limita minimă: Calculatorul de injectie citește o limita sub cea minimă admisă de un senzor sau actuator.
* Banc: Grup de cilindri monitorizat de un senzor comun.
* „A” sau „B”: utilizat pentru identificarea componentelor (actuatoare sau senzori)

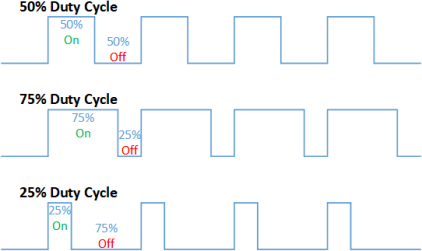
multiple de acelați tip.

* Semnal incoerent/discontinuu: Semnalul eletric este întrerupt pe scurte perioade de timp, perioada fiind insuficienta pentru a declanșa un scurt-circuit sau un circuit deschis.[13]

### SAE J1850 - Pulse Width Modulation (PWM)

Pulse Width Modulation = Modulare a lațimii impulsului **:**

* + Utilizat in principal de Ford Motor Company.
  + Viteza de transfer a datelor de 41.6 kb/secundă.
  + Este o tehnică de variatie controlată a tensiunii aplicate unui dispozitiv electronic
  + Se realizează prin schimbări rapide de tensiune (treceri rapide de la o tensiune mare la o tensiune mică, respectiv un ON-OFF)
  + Raportul dintre perioada totala dintr-un ciclu ON-OFF și perioada de timp ce corespunde valorii ON se numește factor de umplere si generează tensiunea pe care dispozitivul electronic o primește.
  + Folosit si pentru controlul actuatorilor electrici
  + Pinii folosiți sunt 2 (Alimentare +5V) și 10 (Masa semnal) pentru transmiterea datelor.

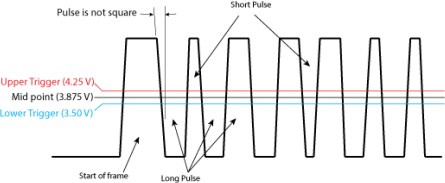


Figură 20: Semnal PWM [14]

### SAE J1850 – Variable Pulse Width (VPW)

Variable Pulse Width = Lațime variabilă a impulsului.

* + Comunicare unidirecționala.
  + Utilizat în principal de General Motors Company.
  + Viteza de transfer a datelor cuprinsă intre 10.4 – 41.6 kb/secundă.
  + Funcționalitatea este similară cu PWM.
  + Folosește o tipologie de rețea bazată pe bus.
  + Toate modulele de comunicație conectate simultan in paralel la același fir.
  + Pinii folosiți sunt 2 (Alimentare +5V) și 10 (Masă semnal) pentru transmiterea datelor.

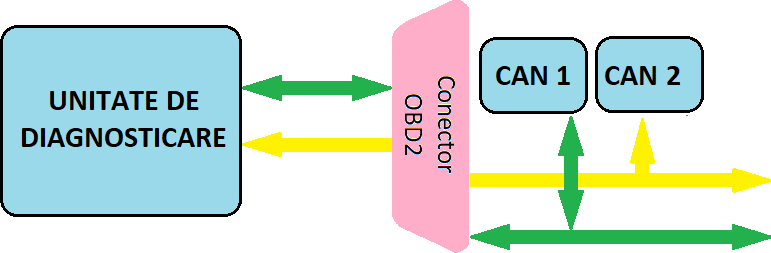


Figură 21: Semnal VPW

### ISO 9141-2 – K-Line (similar RS-232)

K-Line reprezintă un protocol versatil, utilizat in diagnosticarea On-Board, dar și Off- Board. Acesta poarta rolul de transmițator de informații, fiind bidirecțional și utilizând principiul UART (Universal Asynchronous Receiver-Transmitter).

* Poartă rolul de transmitere a informațiilor.
* Comunicarea se realizează bidirecțional.
* Utilizează principiul UART (Universal Asynchronous Receiver-Transmitter).
* Utilizat de producatorii de automobile europeni, asiatici și autovehicule Chrysler.
* Viteza de transfer a datelor este de 10.4 kBaud (masura a ratei de transfer a datelor dintr- un canal de comunicație).
* Pinii folositi sunt pinul 7 (K-Line) și opțional pinul 15 (L-Line) pentru transmiterea datelor.



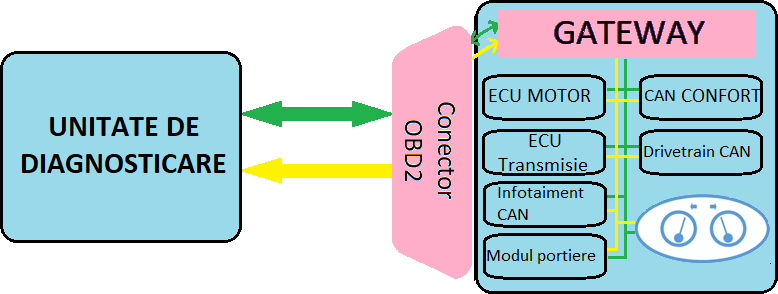
Figură 22: Comunicare ISO 9141

### ISO 14230 - Key Word Protocol 2000 (KWP2000)

Acesta folosește aceeasi modalitate de comunicare ca și ISO9141-2 (K-Line).

* + Poartă rolul de transmitere de informații.
  + Comunicație bidirecțională.
  + Funcționalitatea este similară cu ISO 9141-2.
  + Pinii folosiți sunt pinul 7 (K-Line) și pinul 15 (L-Line) pentru transmiterea datelor.
  + Viteza de transfer a datelor este de aproximativ 10.4 kb/secundă.
  + Se foloseste pentru mesaje ce nu depașesc lungimea de 255 octeti.

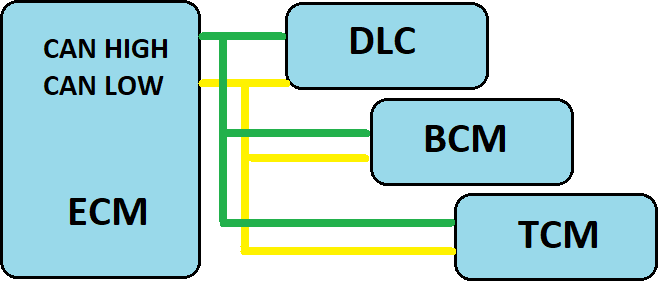
Unitatea de diagnosticare (diagnoza) comunică printr-un Gateway cu ECU-urile mașinii conform figurii urmatoare:



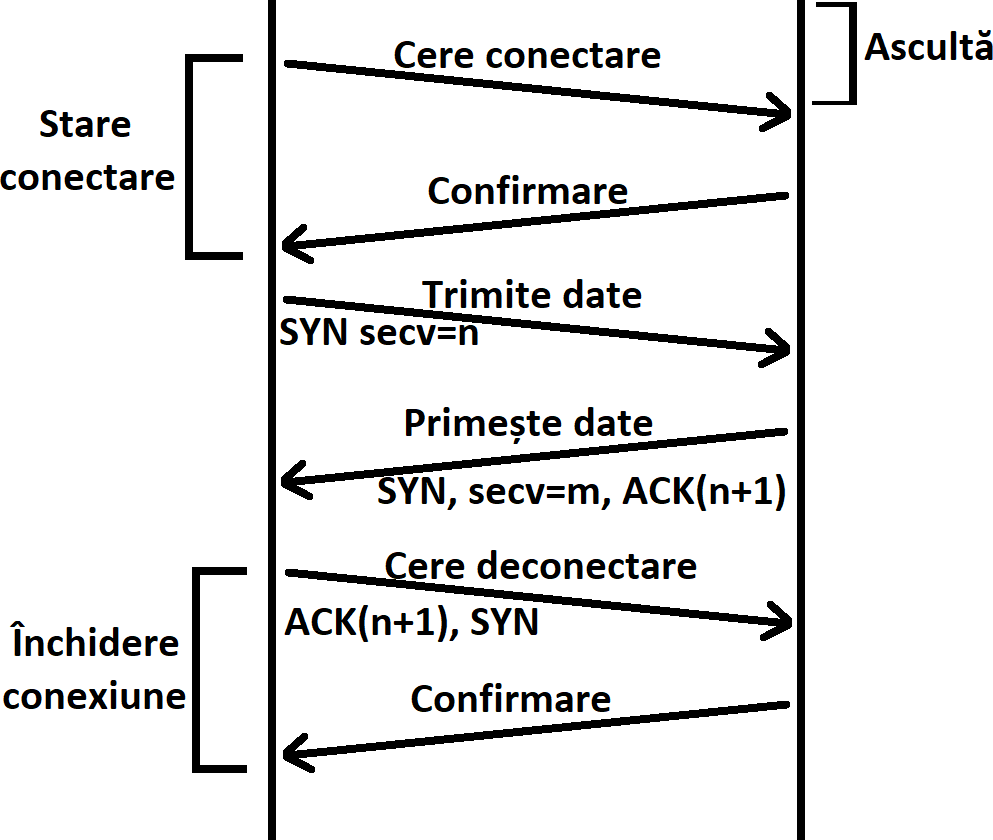
Figură 23: Comunicare ISO 14230

### ISO 15765 – Controller Area Network (CAN)

CAN-ul permite comunicația între unitățile electronice de comandă prin pinii 6 (CAN High) și 14 (CAN Low) fără a fi nevoie de o unitate host, fiind un standard robust folosit în cadrul rețelelor de comunicație dintr-un circuit auto. Viteza de transfer atinge valori între 250 și 500 de kb/s iar lungimea mesajului poate avea maxim 260 de octeti.



Figură 24: Comunicare ISO 15765

Acest protocol poartă și numele de ISO TP (Transport), iar deoarece acesta foloseste o comunicare de tip TCP (Transmission Control Protocol) vom putea deduce o schemă precum cea din Figura 25

Figură 25: Comunicare TCP

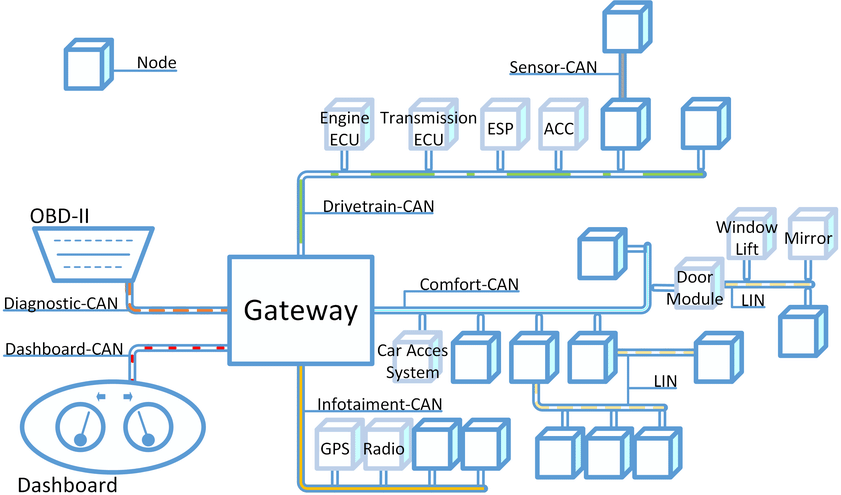
## Comunicația CAN

CAN (Controller Area Network) este un protocol de comunicare serială ce are ca scop realizarea interacțiunii dintre rețelele de control și automatizare. Acesta funcționează exact ca sistemul nervos din corpul uman, monitorizând astfel toate componentele din alcatuirea unui autovehicul. Acesta este dezvoltat în scopul realizării unei bune cunicații între module, unități de control, actuatori, senzori sau alte componente.



Figură 26: Rețeaua electrică a autovehiculului [15]

CAN excelează din punt de vedere al fiabilității, al capacității de rezistență la zgomote, interferențe magnetice, perturbări ale mediului, gradului mare de redundanță, stabilirea priorităților, scalabilitate, flexibilitate, gestiune, adăugare și eliminare de dispozitive dintr-un ansamblu.



Figură 27: Structura de comunicații CAN [16]

Într-un sistem auto modern conform celui prezentat în figura anterioara, avem prezentată comunicarea între componente, utilizând protocoalele de rețea CAN (Controller Area Network). Comunicarea se realizează folosind un Gateway, OBD2, tablou de bord, drivetrain- CAN, CAN-ul de confort și CAN-ul pentru infotainment.

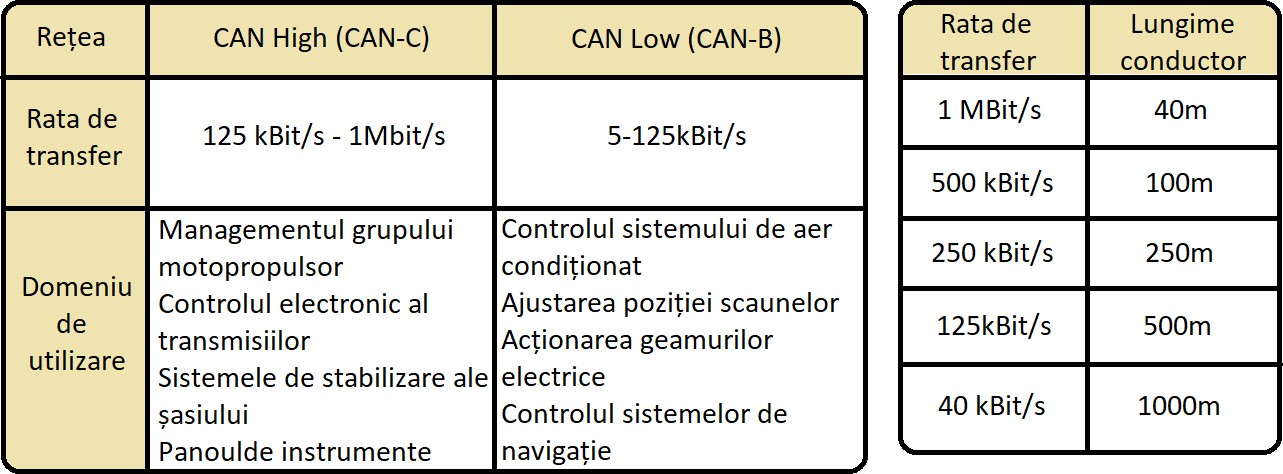
* + - Gateway (Poarta de comunicație): Reprezintă o unitate centrală ce coordoneaza și controlează schimbul de date între module ce utilizează diferite protocoale specifice CAN, asigurând interoperabilitatea și compatibilitatea acestora.
    - OBD2: Sistem standardizat utlizat pentru diagnosticarea și citirea datelor primite de la senzori și modulele acestora.
    - Tabloul de bord (ceasuri): Afișeaza datele referitoare la viteza vehiculului, viteza de turație a motorului, nivelul de combustibil, temperatura motorului sau alte informații esențiale. Spre exemplu, CAN-ul motor trimite semnale spre acest instrument de afișaj iar unele date ce sunt stocate în CAN-ul tabloului se transmit spre circuitul de diagnosticare (kilometrii parcursi ai mașinii, data și ora)
    - Drivetrain-CAN: Reprezintă reteaua CAN utilizată pentru comunicarea dintre modulele implicate în stocarea informațiilor legate de mașina în timpul deplasarii, respectiv motor, transmisie, sistem de frânare sau suspensie. Acesta permite exportarea datelor critice legate de controlul vehiculului, al motorului, viteza componentelor dar și informații referitoare la alte componente atat statice cât și dinamice.
    - CAN-ul de confort: Responsabil de modulele și senzorii ce se ocupa de confortul pasagerilor și functionalități auxiliare ale automobilului (climatizare, centralizare, încălzire scaune, geamuri și oglinzi electrice, etc).
    - CAN-ul Infotaiment: Responsabil în general de comunicarea modulelor și

dispozitivelor responsabile de telecomunicație, audio, afisaje, display-uri tactile sau navigații.

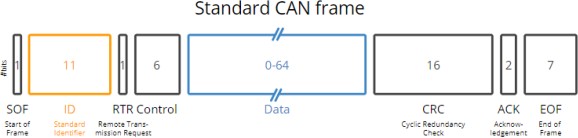
Sistemele mentionate anterior reprezintă o descriere genereală a modului de comunicație între sisteme pentru un sistem auto modern, detaliile specifice fiind diferite în funcție de producător, tipul vehiculului și anul de fabricație.

Comunicația CAN pentru canalul CAN High (Fig. 28), respectiv CAN Low (Fig. 29):

|  |  |
| --- | --- |
| Figură 28: Comunicație CAN High [17] | Figură 29:Comunicație CAN Low [17] |



### Structura unui mesaj CAN



Figură 30 structura unui mesaj CAN [18]

**SOF (Start of Frame)** – dominant cu valoare logică „0”. Rolul acestuia este de a trimite

o intenție de comunicație cu celelalte noduri.

**ID (Identifier)** – Oferă mesajului o modalitate de a fi identificat. Valoarea mai mică a ID-ului reprezintă o importanță mai mare.

**RTR (Remote Transfer Request)** – confirmă recepționarea sau trimiterea de mesaje către un nod.

**CONTROL** – Conține un bit de tip IDE (Identifier Extension Bit) – valoare dominantă

„0”, urmând un bit de rezervă „r0” și 4 biti DLC ( Data Length Code) ce are rolul de a stabili lungimea mesajului ce urmează transmis.

**DATA** – Format din mesajul ce trebuie transmis.

**CRC (Cyclic Redundancy Check)** – Acesta are rolul de asigurare pentru integritatea mesajelor trimise.

**ACK (Acknowledgement)** - Indică transmiterea coretă si completă a mesajului.

**EOF (End Of Frame)** – Sfarșitul mesajului.

## Comenzi ce au ca scop cererea de date

Aceste comenzi realizează conexiunea dintre ELM327 și CAN-urile mașinii, având ca scop generarea de date in scopul afișării parametrilor, identificatorilor, DTC-urilor sau valorilor. Aceste comenzi se împart în două subcategorii: Comenzi de tip AT și comenzi de tip OBD2.

### Comenzi de tip AT (Attention)

În cadrul interfeței ELM327 v2.1 Bluetooth avem un protocol de comunicare utilizat ce se bazeaza pe comenzi de tip „AT” în scopul interacțiunii cu dispozitivele conectate. Aceste comenzi sunt utilizate pentru a controla funcționalitatile dispozitivului și pentru inițierea de comenzi specifice OBD2. [19]

În cadrul acestor comenzi putem încadra urmatoarele mesaje trimise:

* AT Z: resetarea dispozitivului ELM327 pentru a ajunge înapoi la setarile implicite, respectiv o stare inițiala;
* AT E0 și AT E1: Activarea (E1) sau dezactivarea (E0) caracterelor echo. În cazul afișării răspunsului, cazul E0 nu se va afișa comanda trimisă inițial.
* AT H0 și AT H1: Comenzi ce folosesc la afișarea rezultatelor de tip răspuns la comanda specifica trimisă. H0 va dezactiva afișarea răspunsului, iar H1 va face răspunsul vizibil.
* AT SPx: Selectează protocolul de comunicație „x” pentru a reprezenta protocolul specific. AT SP0 va selecta automat protocolul, iar valorile de la 1 la 6 vor selecta manual protocolul (SAE J1850 PWM / SAE J1850 VPW / ISO 9141-2 / ISO 14230-4 (KWP2000) / ISO 15765- 4 (CAN 11-bit) / ISO 15765-4 (CAN 29-bit)
* AT RV: Afișeaza tensiunea de intrare la nivelul dispozitivului ELM327.

În cazul răspunsurilor pentru mesajele trimise putem avea:

* + Caracterul „>” (mai mare): Mesajele trimise spre ELM327 vor fi precedate de acest caracter;
  + Caracterul „ „ (spațiu): Utilizat pentru separarea mesajelor recepționate;
  + Caracterul „<” (mai mic): Răspunsurile vor fi precedate de acest caracter;
  + Caractere între 0-9, A-F: Valori de trimitere/răspuns reprezentate sub formă hexazecimală. Cu ajutorul acestor caractere putem deduce valorile parametrilor, date sau identificatori în timp real.
  + Caracterul „\r” (carriage return): Indică sfarșitul unei comenzi sau răspuns;
  + Caracterul „\n” (newline): Se folosește pentru a trece la o linie nouă in cadrul comenzilor de trimitere sau răspuns.

### Comenzi de tip OBD2

Comenzile de tip OBD, fată de comenzile de tip AT sunt reprezentate de valori, iar de obicei, răspunsul așteptat trebuie să fie construit din minim 3 octeti și maxim 6, reprezentând un cod de tip răspuns în hexazecimal. [19]

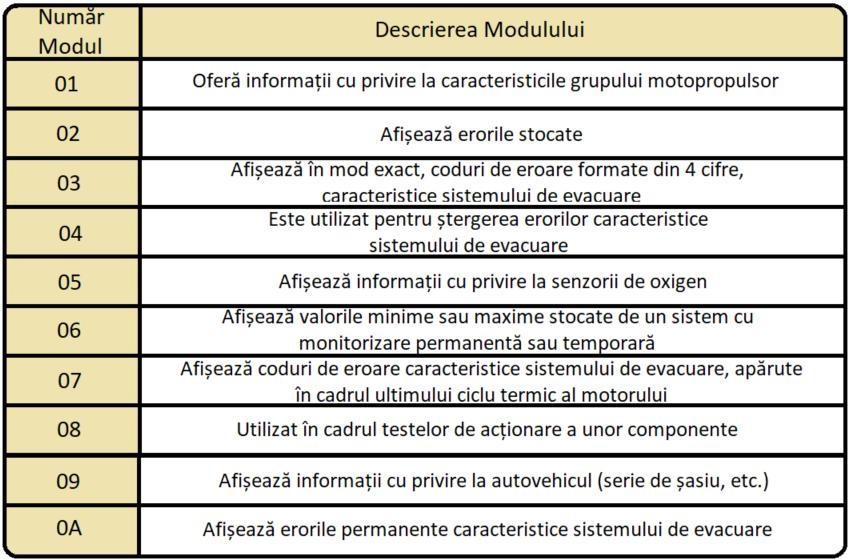
Aceste comenzi sunt definite de SAE (Society of Automotive Engineers) și se numesc modele de serviciu. Aceste modele se împart în subcategorii:

* 01 și 02: Rolul acestora este de a citi date in timp real de la senzorii vehiculului (viteza, temperatura lichidului de răcire, turațiile motorului și multe altele)
* 03 și 06: Servesc la citirea și ștergerea codurilor de diagnostiare, respectiv defecșiunilor (DTC) vehiculului.
* 07: Acesta este utilizat pentru a citi informații suplimentare despre codurile de eroare stocate, cum ar fi numărul de coduri de eroare și starea sistemului de control al emisiilor.
* 08 și 09: Acestea sunt folosite pentru a citi informații suplimentare despre grupul

motopropulsor al vehiculului, cum ar fi informațiile de calibrare.

* 0A : Acest model face citirea DTC-urilor permanente stocate în memoria vehiculului. Acest model oferă ca raspuns o listă a DTC-urilor permanente de pe parcursul rulării vehiculului.

Tabel 3: Comenzi de tip OBD-II

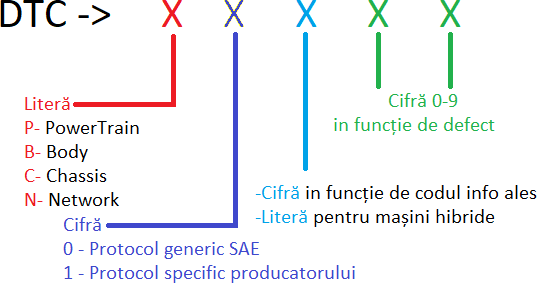


Pe langa toate aceste comenzi, producătorul poate defini comenzi proprii.

## Semnificația codurilor de eroare OBD II

Codurile OBDII din diagnosticarea autovehiculelor sunt menite sa identifice defecțiunile senzorilor și anomaliile de comportament ale motorului.

Aceste DTC-uri (Diagnostic Trouble Codes) sunt de obicei de forma :



Figură 31: Compunerea codurilor DTC

## Tipuri de erori

Erorile se impart in 2 categorii:

* + Erorile permanente (LTFT- Long Term Fuel Trim) se referă la defecte sau problemele care persistă și sunt detectate apoi înregistrate în sistemul de diagnosticare a vehiculului. Respectivele erori sunt numite și coduri de diagnosticare a defecțiunilor permanente (Permanent Diagnostic Trouble Codes - Permanent DTCs). Acestea indică o problemă continuă în interiorul unui anumit sistem sau componentă a vehiculului. Cateva erori de acest tip pot fi: erori ale sitemului de emisii, alimentare cu combustibil, de aprindere, ale sistemului de frânare, etc.
  + Erori temporare (sporadice sau STFT – Short Term Fuel Trim) se referă la erorile de termen scurt cauzate unei nesincronizări sau pot fi doar erori de atenționare. Acestea sunt erori înregistrate în trecut ce nu mai sunt active in prezent. Câteva erori de acest tip pot fi: lipsa lichidului de parbriz, un bec ars ce a fost înlocuit intre timp, presiunea scazută în pneuri, etc.

## Coduri de eroare OBD II

Codurile de eroare ale OBD II sunt gestionate în mai multe categorii. Fiecare dintre acestea sunt în funcție de sistemul sau subsistemul căreia îi aparține.

Avem la dispozitie o gamă de peste 1000 de coduri de eroare de tip Powertrain: [13]

* P00xx-sistemul de alimentare cu combustibil, sistem control aer admisie, sistem auxiliar reducere noxe.
* P01xx, P02xx-sistem alimentare combustibil, sistem control aer admisie.
* P03xx-sistem aprindere, sistem control rateuri combustie.
* P04xx-sisteme auxiliare a reducerilor emisiilor poluante.
* P05xx-sistem de calcul viteză, sistem cotrol relanti și sistemul de control a sistemelor auxiliare de intrare ale motorului.
* P06xx-calculator injectie, date de ieșire ale motorului.
* P07xx, P08xx, P09xx-transmisie (ambreiaj, cutie de viteze, hidrotransformator, mecatronic, diferential).
* P0Axx, P0Bxx-sistem de propulsie hibrida.
* P0Cxx, P0Dxx, P0Fxx -rezervat.
* P10xx, P11xx, P12xx, P13xx, P14xx, P15xx, P16xx, P17xx, P18xx, P19xx, P1Axx, P1Bxx, P1Cxx, P1Dxx, P1Exx, P1Fxx -rezervat producatorilor de automobile.
* P20xx, P21xx, P22xx-sistem alimentare combustibil, sistem auxiliar de reducere a noxelor, sistem control aer admisie.
* P23xx-sistem aprindere, control rateuri.
* P24xx-sistem auxiliar reducere noxe.
* P25xx-control semnale auxiliare intrare motor.
* P26xx-calculator injectie, sistem semnale auxiliare iesire motor.
* P27xx-transmisie.
* P28xx-coduri eroare transmisie.
* P29xx-rezervat admisie.
* P2Axx, P30xx, P31xx, P32xx-sistem alimentare, sistem control aer admisie,sistem auxiliar reducere noxe.
* P33xx-sistem aprindere, control rateuri combustie.
* P34xx-sistem dezactivare cilindri.
* P35xx, P36xx, P37xx, P38xx, P39xx, P3Axx, P3Bxx, P3Cxx, P3Dxx, P3Exx, P3Fxx - rezervat.

# PROIECTAREA APLICAȚIEI

## Tehnologii, framework-uri și limbaje de programare

### Limbajul de programare

Limbajul de programare folosit este python, deoarece este un limbaj de programare dinamic, complex, care accentueaza expresivitatea și înțelegerea ușoară a codului, având o sintaxa mult mai compactă, dar echivalentă cu alte limbaje de programare.

Python este un limbaj de programare avantajos deoarece se adapteaza la un numar mare de domenzii în industria IT precum, web development, DevOps, Data analysis, scopuri educaționale, scriere de teste sau învatare automata.



### Framework-uri

Figură 32: Python Logo [20]

Pentru o interpretare ușoară a programului, am construit o interfața grafica (GUI) ce utilizează biblioteca PyQt5, având în vedere că frameworkul Qt este puternic și popular în dezvoltarea de interfețe pentru utilizatori în aplicații diverse precum desktop, mobile sau incorporate.

PyQt5 permite dezvoltatorilor Python să creeze interfețe la un nivel avansat și interactiv cu functionalitați multiple oferite de Qt5. Aceasta dezvoltă o serie de module și clase ce permit crearea de ferestre, casete de dialog, câmpuri de introducere, butoane, imagini și multe altele.

Matplotlib folosit pentru vizualizarea și crearea graficelor de înaltă calitate. Acest framework oferă o funcționalitate similară MATLAB, fiind recunoscut pentru simplitate, flexibilitate, posibilitatea de exportare în formate precum PNG, JPEG, PDF, SVG, integrare puternică cu Numpy dar și suport pentru grafice

CuteCom este o aplicație software open source pentru comunicarea cu dispozitivele seriale printr-un port serial (RS-232) sau alta interfața serială. Disponibil pentru platforme precum Linux, acesta oferă o interfața simpla, ușor de utilizat pentru trimiterea și primirea datelor seriale.

* 1. ***Proiectarea hardware***

Având în vedere faptul ca dezvoltarea acestei aplicații implica utilizarea unui suport hardware deja existent, avem nevoie doar de modulul ELM327 2.1v Bluetooth (Figura 31-32) care realizează conexiunea fizică dintre unitatea de calcul și ECU. Dispozitivul ELM327 se conectează la portul autoturismului destinat diagnosticării.

Conectorul standardizat OBD2 (Figura 33) este prezent in majoritatea autovehiculelor fabricate începand cu 1996 și face conexiunea la ansamblul hardware dispus de autoturism. Conectorul este singura zonă proiectată special pentru a permite utilizatorilor și tehnicienilor să schimbe informații cu magistrala de date fiind un element cheie în transmiterea de date.

În urma conexiunii se dezvoltă un ansamblu hardware ce permite conectarea la magistrala CAN, iar schimbul de mesaje va fi permis doar după implementarea separată a protocoalelor de comunicație utilizate de dispozitivul de calcul.

Printre componentele acestui ansamblu hardware se numara și:

* + Senzorii sunt dispozitive care detectează și măsoară diverse variabile fizice sau condiții în jurul vehiculului, cum ar fi temperatura, presiunea, accelerația, poziția, nivelul, proximitatea și altele. Acești senzori furnizează date esențiale pentru a asigura funcționarea corectă a sistemelor și sub-sistemelor din vehicul.
  + Actuatorii sunt dispozitive utilizate pentru a controla și acționa diferite componente și sisteme din vehicul. Acestea pot fi actuatori de frânare, direcție, suspensie, închidere/deblocare, aprindere și altele. Actuatorii transformă semnalele primite de la unitățile de control electronic (ECU-uri) în mișcare sau acțiuni specifice.
  + Modulele electronice sunt componente individuale sau plăci de circuit care conțin circuite integrate, procesoare, memorie și alte componente electronice. Aceste module pot fi de control al motorului, control al transmisiei, control al sistemului de climatizare, control al sistemului de divertisment, control al sistemului de siguranță și altele. Fiecare modul electronic are funcții și responsabilități specifice în gestionarea și controlul sistemelor vehiculului.
  + Gateway-ul este componenta care facilitează comunicarea între diferitele module electronice din vehicul și cu alte dispozitive externe. Acestea permit schimbul de informații și date între diversele rețele și protocoale utilizate în vehicul, asigurând interoperabilitatea și integrarea sistemelor și modulelor electronice.
  + Cablajul este responsabil pentru conectarea tuturor componentelor hardware din vehicul prin cabluri electrice. Acesta asigură transmiterea semnalelor și alimentarea cu energie electrică. Cablajul poate fi format din cabluri simple sau complexe, cu conectori și izolație adecvată pentru a asigura o conexiune sigură și eficientă între componente.
  + Unitățile de control electronic (ECU-uri) sunt componente-cheie care controlează și monitorizează diferitele sisteme și sub-sisteme din vehicul. Acestea pot include unități de control ale motorului, transmisiei, airbag-urilor, sistemelor de siguranță și altele. ECU-urile preiau datele de la senzori, le prelucrează și iau decizii în funcție de algoritmi și setările predefinite.

Pentru o diversificare a proiectului am folosit 2 dispozitive ELM327 Bluetooth de generații

diferite (Figura 31-32):

|  |  |
| --- | --- |
| Figură 33: ELM327 vechi | Figură 34: ELM327 nou |

În modulul ELM327 v2.1 avem urmatoarele componente:

* + Comparatorul logic LM393 ce este utilizat pentru a compara semnalele analogice de la senzori cu o tensiune de referință și a determina dacă aceste semnale sunt mai mari sau mai mici decât tensiunea de referință, respectiv pentru a măsura parametrii vehiculului: presiunea

combustibilului, nivelul de oxigen din evacuare, etc.

* + Microcontroler PIC18F2480 creat de Microhip Technology, este programat pentru a procesa semnalele primite de la senzorii vehiculului și pentru a le interpreta în funcție de specificațiile protocolului OBD-II (On-Board Diagnostics) și a altor protocoale de comunicare a vehiculului. Acesta are un set de funcții periferice care includ un modul de comunicare serială (UART), module de temporizare și contorizare, convertoare analog-digitale (ADC) și multe altele.
  + RS323 este utilizat pentru a permite comunicarea serială spre un conector tată ce conține 16 pini (4).

-Un modul bluetooth ce premite conectarea wireless la Raspberry Pi (3).

-Leduri ce indică starea dintre ELM327 și calculatoarele mașinii.

|  |  |
| --- | --- |
| Figură 35: Numerotarea pinilor OBD-II [21] | Figură 36: Microprocesorul ELM327 [19] |

In genereal, poziționarea acestui conector se află intre volan si pedale, sub consola centrală, între scaune sau în cazuri speciale în torpedou și la maxim 0.61 metri de șofer. Accesul la acesa trebuie să fie facil sau acoperit de plastice dar fără a fi nevoie de echipamente speciale de demontare.

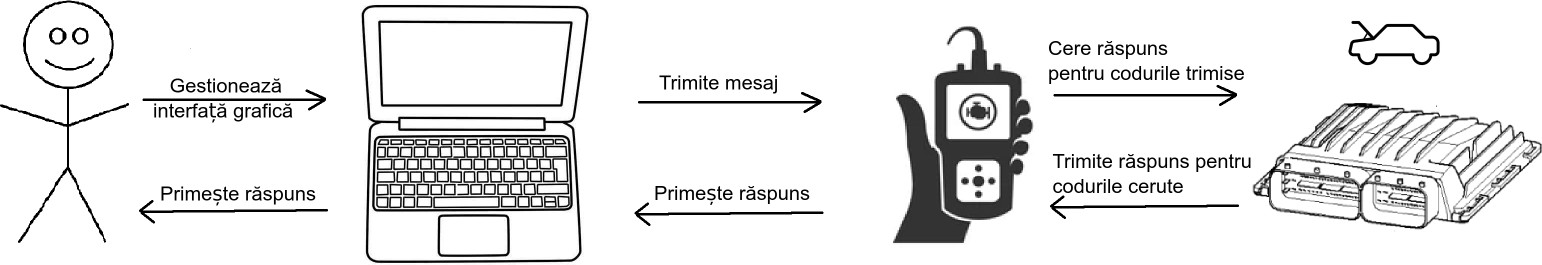
* 1. ***Proiectarea Software***

În cadrul proiectării software, trebuie să construim un canal de comunicație reprezentat de zona de memorie unde să fie posibilă realizarea unui schimb de mesaje intre ECU și unitatea de calcul. Acest lucru este descris prin implementarea software a unor protocoalelor OBD II ce folosesc canale de comunicație și magistrale compatibile în scopul proiectării aplicației la nivel transport și rețea.

Pentru realizarea transmiterii mesajelor am folosit magistrale de comunicație din domeniul auto specifici transferului de date dintre modulele de control ale vehiculului. Cele două standarde folosite sunt J1850 VPW (Variable Pulse Width) și J1850 PWM (Pulse Width Modulation).

Pentru a putea realiza transferul de mesaje am stabilit urmatoarele criterii:

* + Pentru a putea trimite și primi mesaje este necesar sa folosim un delay ce are scopul de evitare a suprasolicitării atât a Bluetooth-ul unitații de calcul, cât și a canalelor de comunicație dintre diagnoză și CAN.
  + Un alt scop al delay-ul trebuie să îndeplinească concordanța dintre sistemele mașinii și răspunsuri, pentru a evita răspunsurile eronate.
  + Trebuie sa avem în vedere cerintele standardului ISO-TP.



Figură 37: Comunicația utilizator – CAN

* + 1. **Conexiunea cu ELM327**

Pentru început am conectat diagnoza la OBD-ul mașinii, pornind bluetooth-ul, stare indicată de ledul roșu al dispozitivului, timp in care am introdus in terminal un set de comenzi pentru activarea conexiunii bluetooth a laptopului.

**~$** bluetoothctl

**[bluetooth]#** power on **[bluetooth]#** agent on

**[bluetooth]#** default-agent **[bluetooth]#**scan on

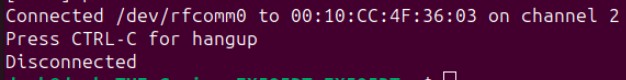
În urma acestor comenzi de pornire a bluetooth-ului, persistare la conectare și scanare, am descoperit adresa MAC a ELM327: 00:10:CC:4F:36:03, pentru care am cerut conexiune (pair 00:10:CC:4F:36:03), raspusul fiind afirmativ în urma introducerii PIN-ului „1234”.

Pentru o conexiune mai rapidă, am creat o zona de memorie de tip rfcomm ce servește schimburilor de mesaje dintre CAN și dispozitivul de diagnoză.

sudo rfcomm bind rfcomm0 <adapter mac address>

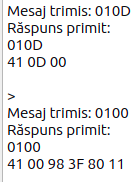
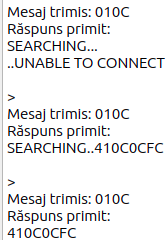
Conectarea cu ELM327 Bluetooth s-a realizat prin comanda: **sudo rfcomm connect**

**/dev/rfcomm0 00:10:CC:4F:36:03 2 -baudrate PP0C** unde /dev/rfcomm0 este numele zonei de comunicație utilizate în realizarea conexiunii seriale și a comunicarii virtuale prin Bluetooth, pentru un baudrate de PP0C prin canalul „2”. Această conexiune funcționeaza aproximativ 10 secunde în cazul inexistenței unui schimb de mesaje între ELM327 și CuteCom.



Pentru dispozitivul de generație mai veche, respectiv 00:10:CC:4F:36:03 conexiunea rămane stabilită pentru aproximativ 10 secunde, caz în care nu sunt realizate schimburi de mesaje, iar pentru dispozitivul de generație mai nouă (66:1E:32:F5:34:45) conexiunea rămâne neîntreruptă.

Toate aceste comenzi funcționeaza atat în regim simulat (doar conexiune la baterie) cât și la conectarea reală (la mașină) utilizând aplicația intermediară numită CuteCom care funcționează asemănător MiniCom, Screen sau PuTTY.

CuteCom foloseste o interfață usor de accesat:

-Device: /dev/rfcom0 → canalul pe care se realizează schimbul de mesaje cu

ELM327 folosind bifa de “Auto Reconnect”

-Boundrate: 115200 Specific ELM327 Bluetooth

-Data Bits: 8

**-**Open Mode: Read/Write

* + 1. **Schimbul de mesaje**

Am încercat trimiterea de comenzi prin intermediul „screen” și „minicom” dar comunicarea nu s-a putut realiza, caz pentru care am folosit canalul de comunicare oferit de CuteCom.

După multe încercări am reușit să obțin răspunsuri pentru mesajele de tip AT (attention), urmând apoi să primesc răspunsuri pozitive pentru mesajele de tip OBD.

Pentru fiecare cerere de comunicare, ELM327 va semnala acceptul mesajului prin

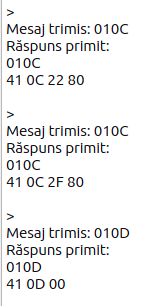
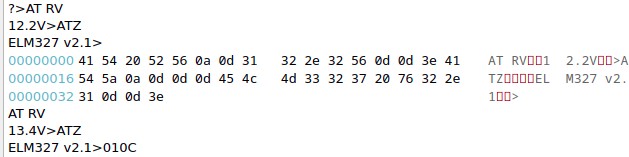
aprinderea ledului de culoare albastră (timp de o secunda).

În funcție de generația dispozitivului de diagnosticare, putem obține rezultate diferite din

punct de vedere a raspunsului în timp real sau viteza de procesare a mesajelor și răspunsurilor.

Pentru a primi răspuns la mesajele de tip OBD va fi nevoie sa selectam protocolul manual prin comanda AT SP x, unde „x” este un numar de la 1 la 6 sau automat prin comanda AT SP0. În caz contrar vom primi un răspuns de tipul : >01 0C NO DATA.

La prima introducere a unei comenzi ce face comunicare directă la CAN,caz în care ne vom aștepta la o afișare de tip : **“SEARCHING…”**. După afișarea acestui mesaj putem primi 2 tipuri de răspunsuri: “**..UNABLE TO CONNECT”** sau raspuns pentru o comanda trimisa spre CAN, comandă de tip OBD-II.



Figură 38: Rezultate primite

* + 1. **OBD II PID’s**

În cazul dispozitivelor de diagnoză folosite vom avea un set de comenzi OBD, mai precis 13 comenzi compatibile cu dispozitivele folosite:

**01 01 si 0100:** Aceste comenzi afișează un șir de numere precum: **41 00 98 3F 80 11**

respectiv **41 01 00 00 00 00** unde:

-41 reprezintă numarul cererii ( 01+40=41)

-00/01 reprezintă PID-ul comenzii trimise (01 **00** respectiv 01 **01**)

-98 3F 80 11 respectiv 00 00 00 00 00 reprezentând parametrii suplimentari pentru aceasta comandă de interogare, respectiv identificatorul de informații cerute sau

alte setări specifice.

**01 04 (Calculated engine load - Procentul de sarcină a motorului):** 0+1/2.55 \* [val] => rez: 41 04 83 => 0+1/2.55 \* 131 = **51,37%**

**01 05 (Engine coolant temperature - Temperatura apei din motor):** -40+1\*[val] => 41 05 79 => -40+1\*121=**81 degC**

**01 0B (Intake manifold absolute pressure - Presiune debitmetru):** 0+1\*[val] => 41 0B 6D =>0+1\*109 = **109 kPa (bari)**

**01 0C (Engine Speed – Turatii pe minut):** 0+0.25\*[val] => 41 0C 22 80 => 0+0.25\*8832

= **2208 RPM**

**01 0D (Vehicle Speed – Viteza):** 0+1\*[val] => 41 0D 61 => 0+1\*97= **97 km/h**

**01 0E (Timing advance -Avansul de aprindere):** -64+0.5\*[val] => 41 0E 80 => - 64+0.5\*128 = **0 deg**

**01 0F (Intake air temperature -Temperatura de intrare în motor):** -40 + 1\*[val] => 41 0F 52 = **42 deg**

**01 10 (Mass air flow sensor (MAF) air flow rate - Cantitatea de aer la intrarea in motor):** 0+0.01\*[val] => 41 10 03 47 => 0+0.01\*839 = **8.329 grame/secunda**

**01 11 (Throttle position - Poziția pedalei de acceleratie):** 0+1/2.55\*[val] => 41 11 0D

=> 0+1/2.55\* 13 = **5.1%**

**01 1C (OBD standards the vehicle conforms to- Standarde OBD conform vehiculului):** 0+1\*[val] => 41 1C 0D => 0+1\*13 = **13 codificari adaptate**

**01 21 (Distance traveled with malfunction indicator lamp(MIL) on - Distanța parcursă cu Check Engine ON):** 0+1\*[val] => 41 21 00 00 => 0+1\*0=**0 km**

* 1. ***Interacțiunea dintre utilizator și interfata grafica***

Pentru o comunicare mai eficienta și ușoară între ELM327 și ECU am construit o interfață

grafică în python prin utilizând biblioteci precum „PyQt5” și „MatPlotLib” construită din:

* + Buton de tip switch pentru activarea și dezactivarea Bluetooth-ului din sistemul de operare.
  + Dropdown pentru selectarea dispozitivului ELM327 conform adresei MAC specifice
  + Dropdown pentru selectarea canalului utilizat de ELM327
  + Buton ce deschide un nou terminal și trimite comanda de conectare a dispozitivului în funcție de adresa MAC și channel: **sudo rfcomm connect /dev/rfcomm0 {selected\_ip}**

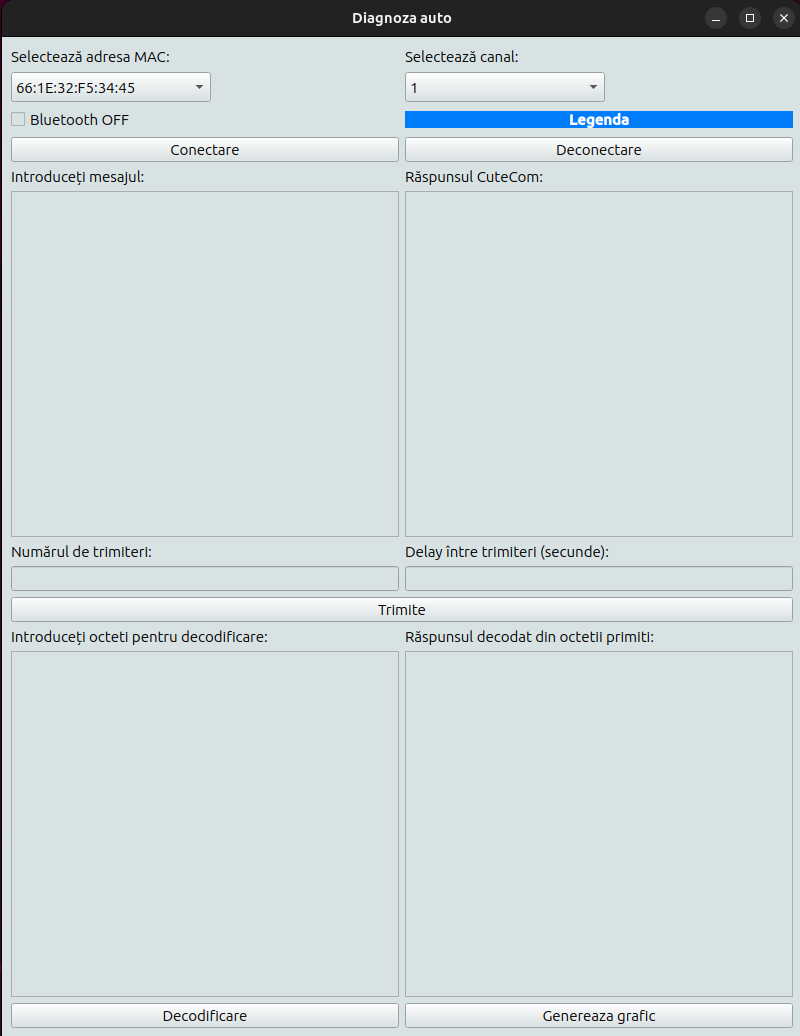
#### {selected\_channel} -b PP0C

* + Buton „Bluetooth ON/OFF” ce realizează deconectarea bluetooth de la dispozitivul ELM327
  + Textbox „Introduceti mesajul” pentru trimitere de mesaje.

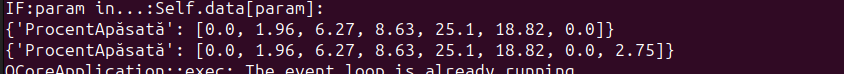
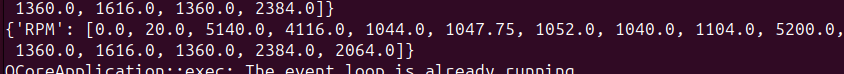
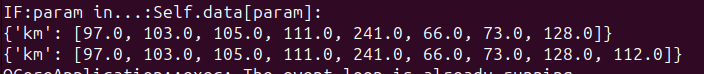
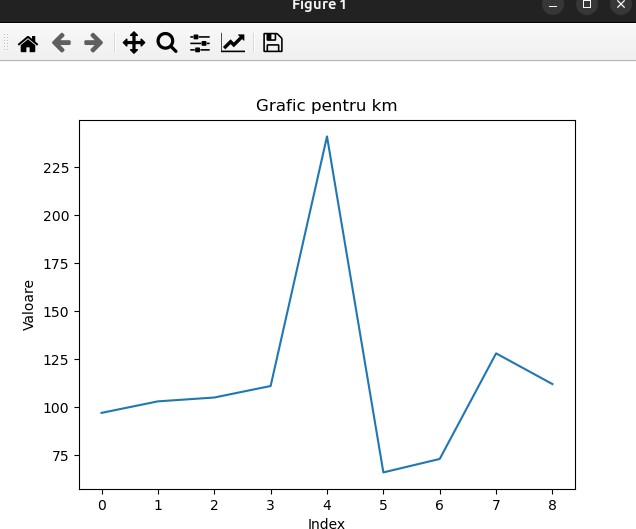
mesaje

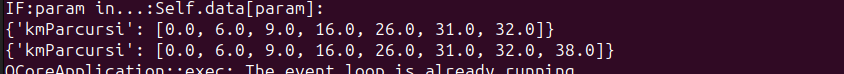
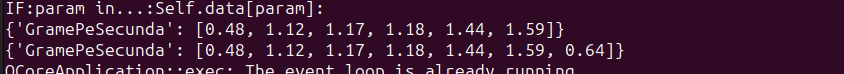
* Textbox „Numărul de trimiteri” a numarului de mesaje ce urmează transmise
* Textbox „Delay” de introducere a Delay-ului, respectiv timpului dintre trimiterile de
* Textbox „Răspunsuri Cutecom” în care apar răspunsurile mesajelor primite de la ECU
* Buton „Trimite” pentru trimiterea de mesaje ce generează primirea de răspunsuri.
* Textbox „Octeti decodificare” de interpretare a răspunsurilor, aici se stocheaza valorile

în hexazecimal primite în textboxul de „Răspunsuri CuteCom” pentru viteză, rpm, temperaturi sau a altor răspunsuri de tip OBD. Acest textbox permite introducerea de valori în hexazecimal pentru decodificarea mesajelor.

* + Textbox „Răspuns decodificat” unde se afisează valoarea și parametrul cerut. În acest textbox se afișează răspunsurile din textboxul de „Răspunsuri Cutecom” decodificate in funcție de valorile de tip număr și parametrul caruia corespund (rpm, viteza, temperatura, etc).
  + Buton „Decodificare” pentru decodificarea șirului de octeți. Acest buton se apelează la apăsarea butonului „Trimite” în scopul primirii mai multor mesaje în timp real (pentru cazul mai multor trimiteri).
  + Buton „Genereaza grafic” folosit în cazul mai multor valori de același timp. Se va deschide o nouă fereastră ce utilizează libraria matplotlib ce va conține o listă de valori afișate grafic.
  + Buton „Legenda” care deschide o nouă fereastră unde sunt afisate toate valorile de tip OBD ce corespund dispozitivului de diagnosticare urmat de semnificația acestora.

Figură 39: Interfață grafică

Graficul generat în matplotlib pune la dispoziție salvarea graficelor obținute în format JPG, PNG, JPEG, etc. Acest grafic se va genera în funcție de numărul variabilelor de același tip (in exemplul de fată pentru km), valorile fiind adăugate într-o coadă în scopul afișării lor succesive.

* 1. ***Funcționalitatea codului***

Figură 40: Model de grafic

Funcționalitatea butonului „Conectare” permite trimiterea comenzii de conectare cu dispozitivul de diagnoza ELM327, comanda fiind trimisa într-un terminal extern. Această comandă necestită drepturi de admin pentru realizarea unei conexiuni corecte, forma acesteia fiind

„sudo rfcomm connect /dev/rfcomm0 {selected\_MAC} {selected\_channel} -b PP0C” unde:

* + rfcomm (Radio Frequency Communication) este un protocol de comunicație utilizat în cadrul rețelelor Bluetooth, fiind un protocol de nivel superior ce faciliteaza transmiterea de date seriale între dispozitive Bluetooth.
  + /def/rfcomm0 este zona de comunicație folosită în cadrul ELM327
  + selected\_MAC reprezintă adresa MAC specifică dispozitivului utilizat.
  + selected\_channel reprezintă canalul de comunicație specific J1850 PWM sau VPW (1 sau

2)

* + -b PP0C sau -b 115200 reprezintă rata de transmisie, respectiv viteza de transfer a datelor

pe canalul de comunicație dintre ELM327 și vehicul.

Funționalitatea butonului „Deconectare” permite trimiterea comenzii „sudo rfcomm release /dev/rfcomm0” ce servește ștergerii zonei de comunicație bluetooth dintre ELM327 și vehicul, respectiv /dev/rfcomm0.

def send\_command(self, selected\_MAC, selected\_channel, action): if action == "connect":

command = f"xterm -e 'sudo rfcomm connect /dev/rfcomm0 {selected\_MAC} {selected\_channel} -b PP0C'" elif action == "disconnect":

command = "sudo rfcomm release /dev/rfcomm0" try:

subprocess.Popen(command, shell=True) except subprocess.CalledProcessError as e:

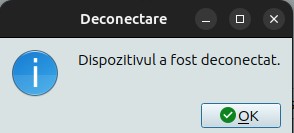
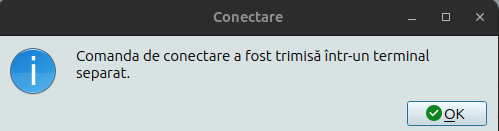
QMessageBox.warning(self, "Eroare", f"A apǎrut o eroare în timpul execuției comenzii:\n{e}") if action == "connect":

QMessageBox.information(self, "Conectare", "Comanda de conectare a fost trimisǎ într-un terminal separat.") elif action == "disconnect":

QMessageBox.information(self, "Deconectare", "Dispozitivul a fost deconectat.")

Aceste butoane generează și răspunsuri pentru reușirea conexiunii, reușirea deconectării

sau în cazul unei erori se va trimite mesajul „Eroare”.



Figură 41: Mesaj de conectare/deconectare

Înainte de trimiterea oricarui mesaj, se va face o testare a conexiunii prin trimiterea

comenzii „ATZ” -> resetul dispozitivului iar răspunsul negativ va afisa un mesaj de eroare.

Răspunsul „read\_response()” va fi pozitiv pentru un răspuns de tip „>” sau negativ în caz

contrar.

def establish\_connection(self):

try:

*# Open serial connection*

self.serial\_connection = serial.Serial('/dev/rfcomm0', 115200, timeout=1) except serial.SerialException as e:

QMessageBox.warning(self, "Eroare", f"A apǎrut o eroare în timpul stabilirii conexiunii:\n{e}") return False

*# Wait for device response* self.serial\_connection.write(b'ATZ\r') response = self.read\_response()

if b'ELM327' in response:

return True

QMessageBox.warning(self, "Eroare", "Conexiunea cu dispozitivul nu a fost stabilitǎ corect.") return False

Apasarea butonului de trimitere a mesajelor este dezvoltată astfel încât sa apeleze mai

multe funcții specifice programului:

- Apelul de trimitere a mesajului va face o afișare a răspunsului în textboxul „Răspuns CuteCom”. Aici se va face o trimitere și primire de mesaje utilizand facitiltațile comunicării

„CuteCom”.

def send\_messages(self):

message = self.input\_textbox.toPlainText() num\_sends = int(self.num\_sends\_textbox.text()) delay = float(self.delay\_textbox.text()) self.output\_textbox.clear()

try:

self.output\_textbox.clear()

* Pentru utilizarea butonului „Trimite” va trebui să selectăm numărul de trimiteri (num\_sends) dorit și delay-ul (delay) în secunde care să fie executat fiecare trimitere.

if self.decoded\_octeti: self.bytes\_in\_textbox.append(self.decoded\_octeti) self.bytes\_in\_textbox.repaint()

time.sleep(2) self.button\_decodare.click()

QApplication.processEvents() except serial.SerialException as e:

QMessageBox.warning(self, "Eroare", f"A apǎrut o eroare în timpul trimiterii mesajelor:\n{e}")

def convert\_bytes1(self):

bytes\_value = self.bytes\_in\_textbox.toPlainText()

self.result = convert\_bytes(bytes\_value) *#aici se stocheaza noua valoare*

print(self.result) print(type(self.result)) *# self.data=self.result* self.add\_data\_to\_list()

self.bytes\_out\_textbox.setText(self.result)

*# self.data=''*

for \_ in range(num\_sends):

self.serial\_connection.write(message.encode() + b'\r') response = self.read\_response() self.output\_textbox.clear() self.output\_textbox.append(response.decode()) time.sleep(2)

*# Obținerea șirului de caractere din textboxul de output (CuteCom)*

input\_text = self.output\_textbox.toPlainText() *# Obținere text cu linii noi din textboxul de output al raspunsului CuteCom*

time.sleep(delay)

* Selectarea mesajului în hexazecimal din textboxul „Răspuns CuteCom” : Se face un algoritm de căutare a valorii „41”- reprezentată de comanda de citire în timp real (01+40) urmată de parametrul căuia îi corespunde („0D”-viteza, „0C”-km/h) iar ultimii octeți generând numărul corespunzător.

for char in input\_text:

if capture\_chars:

if char == '\n' or char == '>':

break self.decoded\_octeti += char

if char == '4' and not found\_41:

found\_41 = True

if len(input\_text) >= 2 and input\_text[input\_text.index(char) + 1] == '1': capture\_chars = True

self.decoded\_octeti += '4' *# Adăugăm secvența "41" la rezultat*

print("acesta este decoded\_octeti:") print(self.decoded\_octeti) *#exemplu 410D51*

* Se face un apel la butonul de „Decodificare” care va interpreta primii doi octeți ai răspunsului drept parametru, iar ultimii octeți sunt interpretați ca valoare în raport cu parametrul trimis la un deelay de 2 secunde pentru a nu suprasolicita protocolul de comunicație bluetooth rfcomm.
* Decodificarea fiecărui parametru „41” se va face prin apelul funcției „convert\_bytes”

def convert\_bytes(hex\_string):

try:

hex\_string = hex\_string.replace(" ", "") *# Elimină toate spațiile din șirul hexazecimal*

if len(hex\_string) >= 4:

first\_byte = hex\_string[0:2] *# Primul byte* second\_byte = hex\_string[2:4] *# Al doilea byte* remaining\_bytes = hex\_string[4:] *# Bytes rămași*

if first\_byte == "41" and second\_byte == "04": decimal\_value = int(remaining\_bytes, 16) result = 0 + decimal\_value / 2.55

return str(result) + " ProcentSarcina"

elif first\_byte == "41" and second\_byte == "21": decimal\_value = int(remaining\_bytes, 16) result = 0 + 1 \* decimal\_value

return str(result) + " kmParcursi"

................................................................................

elif first\_byte == "41" and second\_byte == "0C": decimal\_value = int(remaining\_bytes, 16) result = 0 + 0.25 \* decimal\_value

return str(result) + " RPM"

- Adaugarea de date într-o listă ce conține parametrul „param” și valorile „value” cu scopul

pregătirii datelor în cazul în care utilizatorul dorește generarea graficului.

def add\_data\_to\_list(self):*#Aici am ramas cu trimiterea datelor.* data = self.result

match = re.match(r'(\d+(\.\d+)?)\s\*(\w+)', data) if match:

value = float(match.group(1)) *# aici se salveaza numarul din raspunsul decodat*

param = match.group(3) *#aici se salveaza tipul (km/h, RPM, etc)*

if param == self.current\_param:

if param in self.data:

print("IF:param in...:Self.data[param]:") print(self.data) self.data[param].append(value)

else:

print("ELSE:param in...:Self.data[param]:"+str(self.data[param])) self.data[param] = [value]

else:

print("ELSE")

self.data = {param: [value]} self.current\_param = param

print(self.data)

Pentru cazul în care utilizatorul dorește activarea sau dezactivarea bluetooth-ului din interfața, am introdus și posibilitatea de închidere sau deschidere a acestuia:

def toggle\_bluetooth(self):

if self.bluetooth\_button.isChecked(): subprocess.Popen("sudo rfkill block bluetooth", shell=True) self.bluetooth\_button.setText("Bluetooth OFF")

else:

subprocess.Popen("sudo rfkill unblock bluetooth", shell=True) self.bluetooth\_button.setText("Bluetooth ON")

Cu urmatoarele funcții, utilizatorul are posibilitatea de generare a unui grafic cu valorile

obținute în lista de parametri, de a salva graficul în formate precum PNG, JPEG.

def generate\_graph(self):

if self.current\_param in self.data: if self.figure is not None:

plt.close(self.figure) self.figure = plt.figure()

x = range(len(self.data[self.current\_param])) plt.plot(x, self.data[self.current\_param]) plt.xlabel("Index")

plt.ylabel("Valoare")

plt.title(f"Grafic pentru {self.current\_param}") plt.show()

def clear\_graph(self):

if self.figure is not None: plt.close(self.figure)

# TESTAREA APLICAȚIEI

## Modalitatea de pornire și utilizare a aplicației

Pentru a putea rula aplicația este necesară instalarea unor protocoale, librarii și aplicații

precum:

* O versiune corespunzatoare a interpretorului Python instalată pe sistemul de operare.
* O versiune stabilă de Bluetooth care sa permită conectarea la dispozitivul de diagnoză.
* Instalarea librariei PyQt5 necesară interfetei grafice.
* Instalarea librariei Matplotlib necesară generării graficelor.
* Instalarea aplicației CuteCom ce folosește schimbului de mesaje cu dispozitivul de diagnoză și

CAN.

* O platforma pe care se poate testa în cazul practic (un autovehicul compatibil standardului OBD2)

sau un port de tip OBD2 (tata) care să fie conectat la un acumulator de 12V.

## Simularea obținerii de date

În lipsei unui sistem ce are o comunicare cu un ECU am reușit simularea acestuia prin introducerea de parametrii asemănători pentru a obține o testare forțată. Dispozitivul ELM327 Bluetooth se conecteaza la un acumulator de 12V, moment în care putem crea o conexiune bluetooth cu dispozitivul, exact ca în cazul real. Diferențele de comunicație se fac doar pentru comenzile de tip OBD2. În acest caz avem posibilitatea de a transmite mesaje de tip AT fără ca pinii de comunicație să fie folosiți. În cazul trimiterii comenzilor din Legenda vom obține un rezultat de tip NO DATA.



Figură 42: Simularea utilizării ELM327

Având în vedere ca știm care este structura octeților trimiși de ELM327 vom proceda astfel: Vom folosi comanda AT E1 pentru a activa caracterele echo, această comandă va face afișarea caracterelor introduse.

Forma unui răspuns pentru cererea 01 yy va fi următoarea: 41 yy xx xx, unde yy reprezintă

comanda specifică cererii (viteza motorului, viteza vehiculului, temperatura apei, etc).

Pentru a primi ca raspuns o viteza de 15km pe ora, vom obține următorul răspuns: 41 0D

0F. În cazul trimiterii acestui răspuns urmat de un enter (newline), programul ar trebui sa afiseze

„41 0D 0F \n NO DATA”.

Deoarece programul este conceput ca răspunsurile de tip „41 \n” să fie selectate pentru

decodificare, programul va genera valori și grafice fără a fi nevoie de conexiune la CAN.

|  |  |
| --- | --- |
| Figură 43: Exemplu rulare pentru viteză | Figură 44: Exemplu rulare pentru RPM |

## Trimiterea unui mesaj în cadrul aplicației

Vom considera un mesaj de tip „show current data”, respectiv serviciului „01”. Acest mesaj este utilizat pentru a solicita și afisa date curente ale vehicului. Prin aceasta comanda de tip cerere trimisă la ELM327 prin protocolul de comunicare OBD-II vom obține informațiile continute în următorul octet trimis. Având în vedere că un octet poate fi reprezentat de 2 cifre sau litere în hexazecimal vom folosi un PID caracteristic unei informații.

Ca exemplu vom folosi un PID ce ne va solicita viteza de rotație a motorului, viteza măsurata în rotații pe minut (RPM), PID corespunzător valorii în hexazecimal „0C”.

În urma răspunsului pozitiv al conectarii dispozitivul ELM327 prin comanda

„Conectare”, vom introduce octetii „01 0C” în textboxul „Introduceti mesajul:”.

La apăsarea butonului „Trimite”, acești octeti se vor scrie în protocolul de comunicație

/dev/rfcomm0 și se va recepționa un mesaj ce se va afisa în textboxul „Răspunsul CuteCom”. Pentru comanda trimisă vom primi un răspuns de tipul „41 0C xx xx”. În cazul nostru am obținut rezultatul „41 0C 2C 21”.

Octetul „41” se va obține prin adunarea la primul octet trimis, numărul 40 (01 + 40 = 41), al 2-lea octet recepționat „0C” este același cu al 2lea octet trimis, iar octetii 2C 21 reprezintă valoarea trimisa de CAN în hexazecimal.

Vom folosi un numar de trimiteri = 1 și un delay = 0, pentru a se trimite o singura comanda.

Valoarea în hexazecimal va fi extrasă din „Răspunsul CuteCom” prin selectarea caracterelor cuprinse între „41, \n sau\r ” și se va afișa sirul de octeți în textboxul „Introduceți octeți pentru decodificare”.

Octetii obținuți în textboxul „Introduceți octeti pentru decodificare” vor fi selectati și trimiși în „convert\_bytes” unde aceștia se vor împărți în first\_byte = 41, second\_byte = 0C și remaining\_bytes = 2C 21. Second\_bytes va stabili numele parametrului, unde 0C este specific

„RPM”, iar octeții din remaining\_bytes se vor converti în variabila de tip „int” din baza 16. Rezultatul conversiei 2C 21 = 11 297, valoare ce se înmulțește cu 0.25 conform formulei de calcul a rotațiilor pe minut și se obtine rezultatul „2828.25 RPM”, ce se returneaza în textboxul

„Răspunsul decodificat din octetii primiti”.

## Obtinerea de date în timp real

Pentru a obține date într-un sistem conectat la ECU va trebui să setăm automat protocolul prin comanda AT SP0, iar primele comenzi trimise vor face conectarea la CAN, mesajele

recepționate fiind de forma: SEARCHING...

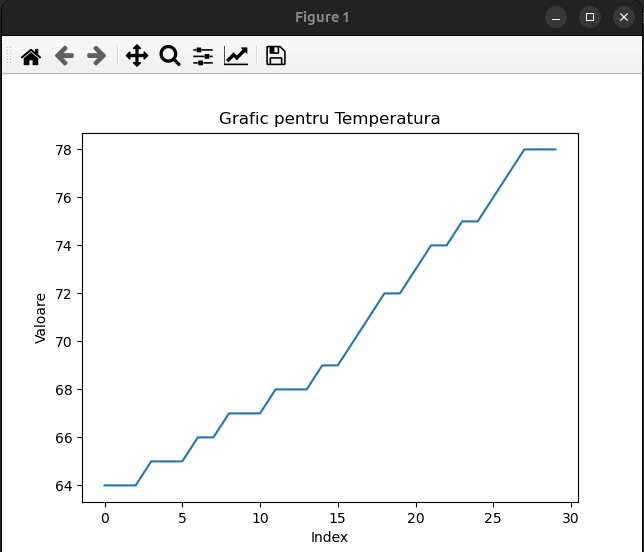
...UNABLE TO CONNECT

la realizarea conexiunii: Rǎspuns primit:

410C0CFC

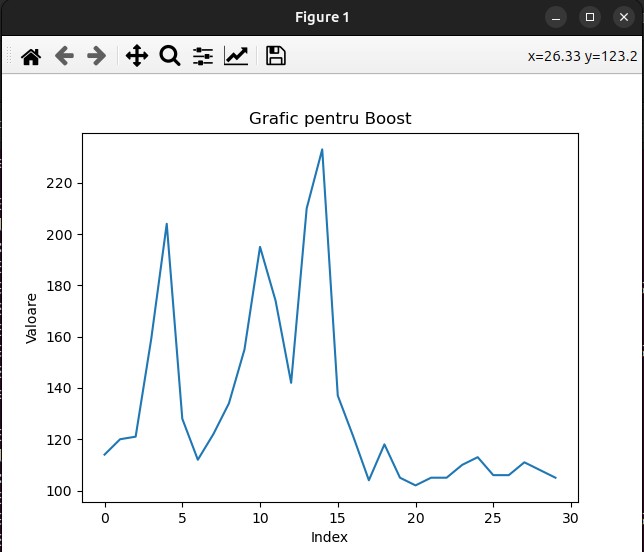
Mai jos avem o serie de grafice ce au fost făcute pentru un numar de 30 de transmiteri cu delay de 1 secundă intre trimiteri, timp la care se adauga execuția fiecarui mesaj, la un vehicul de tip Skoda, transmisie manuală, cu motorizare de diesel, anul de fabricație 2003:

În figura 43 reprezentata temperatura apei, reprezentată prin codul trimis „01 05”. Putem observa ca avem o creștere constantă între valorile 64 grade Celsius și 78 grade Celsius (14 grade diferența) într-o perioadă de aproximativ 150 secunde.



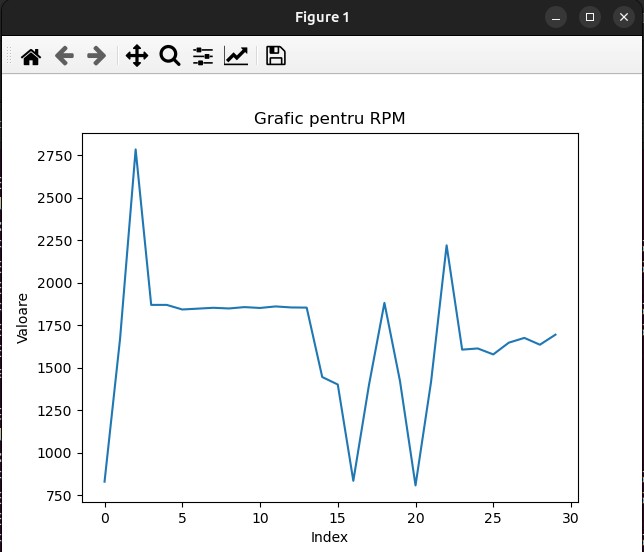
Figură 45:Grafic pentru temperatură apă

În figura 44 este reprezentată presiunea din interiorul blocului motor, parametru ce corespunde codul trimis „01 0B”. Putem observa valori între 100 și 230, în funcție de exploatarea autovehiculului. Acest parametru poate fi influențat de viteza de rotație a motorului, de temperatura aerului, apei și tipului de combustibil folosit.

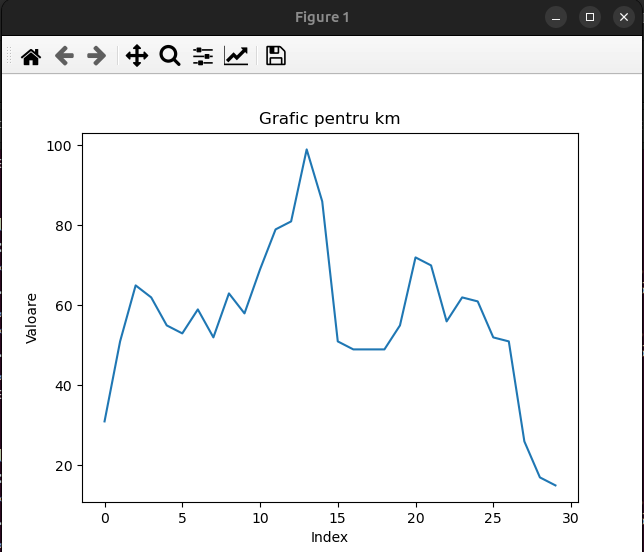


Figură 46: Grafic pentru presiune în camera de ardere

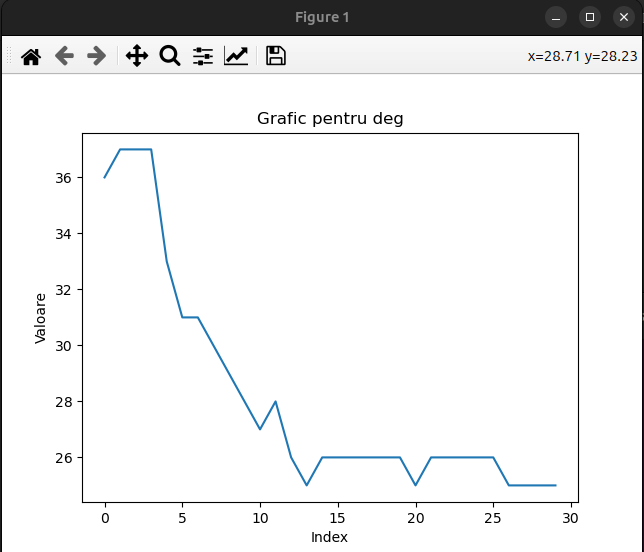
În figura 45 este reprezentata viteza de rotatie a motorului (RPM) parametru ce corespunde codul trimis „01 0C”, exprimată în rotatii pe minut. Valorile unei motorizari diesel vor oscila în general între 750 RPM și 5000 RPM sau 7000 RPM în cazul unei motorizari pe benzină. În cazul nostru am obținut valori de aproximativ 750 și 2750 RPM, având între rezultatele 4 și 13 un mers constant. Fiecare creștere și scădere bruscă a rotațiilor pe minut a fost obținută la schimbarea vitezelor.



Figură 47: Grafic pentru RPM

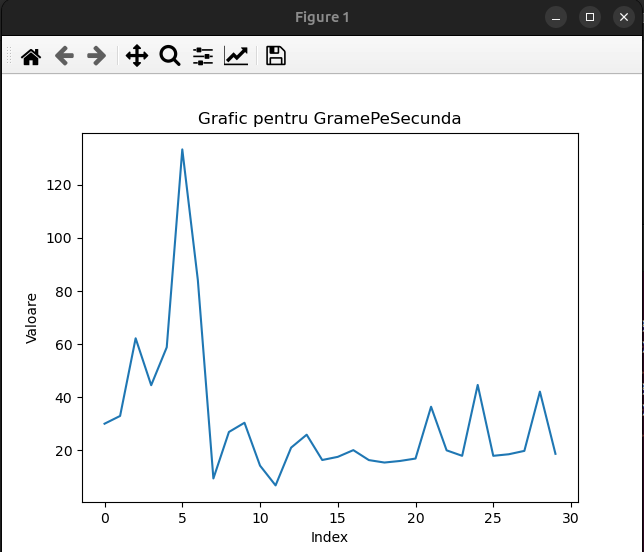
În Fig 46 este reprezentat un grafic pentru viteza în km/h unde putem observa valori între 0 și 100 km/h, parametru ce corespunde codul trimis „01 0D”. Distanța pe care au fost înregistrate datele a fost de aproximativ 5 km pe un drum oscilant din punct de vedere al virajelor, iar valoarea de 100km/h fiind atinsă pe un tronson de drum drept în condiții de siguranță și în limite legale de viteză.

Figură 48: Grafic pentru km/h

În Fig 47 avem reprezentată temperatura aerului la intrarea în debitmetru, aceasta fiind influențată de temperatura motorului în momentul respectiv și viteza de circulație a autovehiculului și parametru ce corespunde codul trimis „01 0F”. Valorile de 36-38 grade s-au obținut la plecarea de pe loc, valorile temperaturii scăzând treptat invers proportional creșterii vitezei. Pentru o viteză constantă de 50 km/h s-au obținut valori între 23-26 de grade ale aerului trecut prin debitmetru.

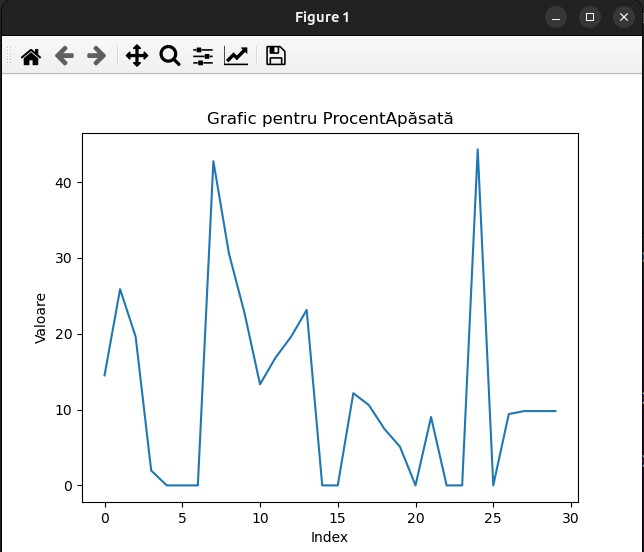
Figură 49: Grafic pentru temperatură aer în debitmetru

În Fig 48 este reprezentată cantitatea de aer la intrarea în motor parametru ce corespunde codul trimis „01 10”. Această cantitate variază între 8 și 137 de grame pe secundă și este influențată de poziția pedalei de accelerație. La o apăsare mai accentuată a pedalei de accelerație, debitul de aer va crește pentru a face o explozie mai puternica în interiorul camerei de ardere, cantitatea de 137 grame fiind obținută la o exploatare mai mare a motorului decât la o exploatare obișnuită de 8-50 de grame pe secundă.



Figură 50: Grafic pentru cantitatea aerului în g/s

In Fig 49 este prezentat procentajul de apăsare a pedalei de accelerație. Aceste valori se pot obține doar în timpul deplasării vehiculului deoarece o apăsare continuă de peste 30% va duce la aporximativ 5000 de rotații pe minut. Valorile obținute sunt între 0 și 44%, valori ce pot fi obținute fără a trece de pragul de 3000 RPM în cazul unui mers subturat.



Figură 51: Grafic pentru % apăsarea pedalei de accelerație

# CONCLUZII

În urma realizării prezentei lucrări care a avut ca scop dezvoltarea unei soluții eficiente pentru diagnosticarea vehiculelor, utilizând tehnologii moderne și metode avansate printr-o comunicație serială între utilizator și calculatoarele unui vehicul, am reusit să realizez un schimb de mesaje cu dispozitivul de diagnoză ELM327 fără a utiliza aplicații disponibile pe piața auto prin intermediul unei interfețe grafice.

Stabilind comunicarea dintre ECU și un dispozitiv de calcul mobil (în cazul meu, laptop) prin Bluetooth, am reușit să obțin o serie de informații, parametri și valori cu privire la funcționarea în timp real a autovehiculului, precum viteza, rotațiile pe minut ale motorului, temperatura apei și alți parametri, pe care le-am transpus mai apoi în grafice pentru a evidenția și mai bine aspectele urmărite.

Afișarea grafică a rezultatelor obținute a fost realizată prin intermediul bibliotecii

„matplotlib”, adăugând posibilitatea salvării rezultatelor în format JPEG/JPG, PNG, GIF, SVG, EPS sau TIFF.

Din dorința de a atrage atenția asupra necesității unor dispozitive care să citească corect semnalele vehiculului, deoarece motivul principal al funcționalității incorecte a sistemelor de siguranță este cauzat de citirea eronată a semnalelor vehiculului, respectiv omiterea cazurilor neprevazute, precum condițiile meteo nefavorabile, caz în care sistemele duc la infuncționabilitate, am reușit să realizez propria aplicație care să genereze și grafice în timp real pe lângă ce reușeau aplicațiile deja existente.

Consider că soluțiile implementate reușesc să atingă în mare parte scopurile propuse inițial, atingând performața de a stoca valori ale parametrilor, care ulterior să permită construirea graficelor în timp real, abilitate pe care încă nu am văzut-o implementată în cazul unui dispozitiv de buget precum ELM327 Bluetooth.

Partea practică ce a ținut de implementarea unei aplicații ce să permită atât transmiterea, cât și recepționarea de mesaje, m-a făcut să iau în calcul și posibilele erori neprevăzute inițial, dintre care amintesc:

* + Găsirea unei soluții pentru trimiterea de mesaje spre ELM327;
  + Suprapunerea de mesaje ce ducea la blocarea bluetooth-ului în cadrul sistemul de operare;
  + Identificarea problemei ce genera deconectarea dispozitivului de generație mai veche;
  + Trimiterea unui numar de mesaje la un deelay mic ce bloca interfața.

De asemenea, am luat în considerare și unele aspecte care ar putea fi îmbunătățite în vederea extinderii platformei, astfel aș mai adăuga obținerea de valori medii, eficientizarea proiectării software, obținerea mai multor tipuri de parametri simultan și creșterea de utlități prin avansarea la un dispozitiv mai complex ce permite modificarea parametrilor.

Această lucrare vine ca un plus de informare cu privire la situația actuală a tehnologiilor deja implementate în domeniul diagnisticării auto, iar semnalul de alarmă tras prin problemele și amenințările identificate și prezentate, respectiv realizarea unei interfețe grafice mult mai performante, pot ajuta autoritățile competente să îmbunătățească managementul dispozitivelor care să citească corect semnalele vehiculului prin luarea unor măsuri care ar putea reduce numărul amenințărilor și impactul celor deja existente.

# BIBLIOGRAFIE

1. DENSO EUROPE 2023 [Interactiv] Disponibil: https://www.denso- am.eu/ro/noutati/istorie-sistem-management-motor-denso
2. „Electric Mura” 2021 Mura Car Accessories [Interactiv]. Disponibil: https://mca.electricmura.ro/ro/blog-tester-obd-ii/
3. Aparat de masura si osciloscop [Informativ]. Disponibil: [https://www.tme.eu/ro/details/sds1102/osciloscoape-](https://www.tme.eu/ro/details/sds1102/osciloscoape-digitale/owon/?brutto=1&currency=RON&gclid=CjwKCAjwzJmlBhBBEiwAEJyLu0ejmPHItNbYPTjQfgWaEzPfG20NY6kfiz61wCH6_PsCeJTDB9AywhoCnrwQAvD_BwE) [digitale/owon/?brutto=1&currency=RON&gclid=CjwKCAjwzJmlBhBBEiwAEJyLu0ej](https://www.tme.eu/ro/details/sds1102/osciloscoape-digitale/owon/?brutto=1&currency=RON&gclid=CjwKCAjwzJmlBhBBEiwAEJyLu0ejmPHItNbYPTjQfgWaEzPfG20NY6kfiz61wCH6_PsCeJTDB9AywhoCnrwQAvD_BwE) [mPHItNbYPTjQfgWaEzPfG20NY6kfiz61wCH6\_PsCeJTDB9AywhoCnrwQAvD\_BwE](https://www.tme.eu/ro/details/sds1102/osciloscoape-digitale/owon/?brutto=1&currency=RON&gclid=CjwKCAjwzJmlBhBBEiwAEJyLu0ejmPHItNbYPTjQfgWaEzPfG20NY6kfiz61wCH6_PsCeJTDB9AywhoCnrwQAvD_BwE) https://vectro.ro/produs/multimetru-digital-vc9205n/
4. ELM327 Bluetooth [Informativ]. Disponibil: https:/[/www](http://www.aliexpress.com/i/1005002857427418.html).[aliexpress.com/i/1005002857427418.html](http://www.aliexpress.com/i/1005002857427418.html)
5. Tester multimarca [Informativ]. Disponibil: [https://www.storel.ro/tester-multimarca-](https://www.storel.ro/tester-multimarca-ds150e-intertafa-diagnoza-auto-2020-romana-cabluri-full/?srsltid=ASuE1wQb92iN528lqThSs9GiuqL33sj8SoKtJaCllZkdpLSQheWYZvokpFM) [ds150e-intertafa-diagnoza-auto-2020-romana-cabluri-](https://www.storel.ro/tester-multimarca-ds150e-intertafa-diagnoza-auto-2020-romana-cabluri-full/?srsltid=ASuE1wQb92iN528lqThSs9GiuqL33sj8SoKtJaCllZkdpLSQheWYZvokpFM) [full/?srsltid=ASuE1wQb92iN528lqThSs9GiuqL33sj8SoKtJaCllZkdpLSQheWYZvokpF](https://www.storel.ro/tester-multimarca-ds150e-intertafa-diagnoza-auto-2020-romana-cabluri-full/?srsltid=ASuE1wQb92iN528lqThSs9GiuqL33sj8SoKtJaCllZkdpLSQheWYZvokpFM) [M](https://www.storel.ro/tester-multimarca-ds150e-intertafa-diagnoza-auto-2020-romana-cabluri-full/?srsltid=ASuE1wQb92iN528lqThSs9GiuqL33sj8SoKtJaCllZkdpLSQheWYZvokpFM)
6. Diagnoză VCDS [Informativ]. Disponibil: [https://www.emag.hu/vag-com-vcds-20-4-](https://www.emag.hu/vag-com-vcds-20-4-vagcom-hex-can-vw-audi-seat-skoda-tipusok-diagnosztizalasara-067/pd/DXTXFYMBM/) [vagcom-hex-can-vw-audi-seat-skoda-tipusok-diagnosztizalasara-067/pd/DXTXFYMBM/](https://www.emag.hu/vag-com-vcds-20-4-vagcom-hex-can-vw-audi-seat-skoda-tipusok-diagnosztizalasara-067/pd/DXTXFYMBM/)
7. Mega Macs PC Bike [Informativ]. Disponibil: [https://ro.hella-](https://ro.hella-gutmann.com/ro/echipamente-service/diagnoza/diagnoza-motociclete/) [gutmann.com/ro/echipamente-service/diagnoza/diagnoza-motociclete/](https://ro.hella-gutmann.com/ro/echipamente-service/diagnoza/diagnoza-motociclete/)
8. Martori [Informativ]. Disponibil: [https://www.autodna.ro/blog/semnificatia-martorilor-](https://www.autodna.ro/blog/semnificatia-martorilor-de-bord/) [de-bord/](https://www.autodna.ro/blog/semnificatia-martorilor-de-bord/)
9. Display martor [Informativ]. Disponibil: https://dailydriven.ro/martor-dpf-aprins-la-audi- a3
10. 2010 - 2023 OUTILS OBD FACILE [Interactiv]. Disponibil: https:/[/www](http://www.outilsobdfacile.com/vehicle-list-compatible-obd2.php).[outilsobdfacile.com/vehicle-list-compatible-obd2.php](http://www.outilsobdfacile.com/vehicle-list-compatible-obd2.php)
11. Conector OBD-II [Informativ]. Disponibil: https:/[/www](http://www.flexihub.com/oobd2-pinout/).[flexihub.com/oobd2-pinout/](http://www.flexihub.com/oobd2-pinout/)
12. Conector OBD-II Type A și type B [Interactiv]. Disponibil: <http://www.obdtester.com/obd2_connector>
13. Stefan Angelo “Totul despre OBD2” [Forum Interactiv]. Disponibil: https://forum.4tuning.ro/27-discutii-generale-masini/345488-obd-erori-traducere-erori- piese-localizate-functie-eroare/page-2.html
14. Pulse width modulation [Interactiv]. Disponibil: https://learn.sparkfun.com/tutorials/pulse-width-modulation/all
15. Rețeaua electrică a autovehiculului [Informativ]. Disponibil: https://plm.sw.siemens.com/en-US/simcenter/simulation-test/high-frequency- electromagnetics/
16. Structura de comunicații CAN [Informativ]. Disponibil: https:/[/www](http://www.researchgate.net/figure/Generic-model-of-a-car-CAN-bus-Multiple-).[researchgate.net/figure/Generic-model-of-a-car-CAN-bus-Multiple-](http://www.researchgate.net/figure/Generic-model-of-a-car-CAN-bus-Multiple-) subnetworks-with-different-functionality-are\_fig1\_320837400
17. CAN High, CAN Low [Interactiv]. Disponibil: [http://www1.gcanbox.com/fsd/gsxw/high-speed-CAN,low-speed-CAN.html](http://www1.gcanbox.com/fsd/gsxw/high-speed-CAN%2Clow-speed-CAN.html)
18. Structura unui mesaj CAN [Informativ]. Disponibil: https://embeddedtechsolutions.com/can-protocol-and-its-description/
19. ELM327DSJ – OBD to RS232 Interpreter. [Datasheet] Disponibil: <https://www.elmelectronics.com/>https:/[/www](http://www.elmelectronics.com/wp-).[elmelectronics.com/wp-](http://www.elmelectronics.com/wp-) content/uploads/2016/07/ELM327DS.pdf
20. 16 mai 2021 Optimised logo // Editing SVG source code using [Informativ]. Disponibil: https://validator.w3.org/check?uri=https%3A%2F%2Fcommons.wikimedia.org%2Fwiki%2FSpe cial%3AFilepath%2FPython\_logo\_and\_wordmark.svg&doctype=Inline&ss=1#source
21. Numerotarea pinilor OBD-II – StackExchange 2023 StackExchange Inc. [Informativ]. Disponibil: [https://mechanics.stackexchange.com/questions/25257/disabling-obd2-interface-in-a-](https://mechanics.stackexchange.com/questions/25257/disabling-obd2-interface-in-a-car) [car](https://mechanics.stackexchange.com/questions/25257/disabling-obd2-interface-in-a-car)

Tabel de figuri

[Figură 1: Aparat de masură si Osciloscop [3] 9](#_bookmark8)

[Figură 2: Model ELM327 cablu [2] 10](#_bookmark9)

[Figură 3: ELM327 BT [4] 10](#_bookmark10)

[Figură 4: Diagnoză cu fir [2] 10](#_bookmark11)

[Figură 5: Tester multimarcă [5] 10](#_bookmark12)

[Figură 6: Diagnoză VCDS [6] 10](#_bookmark13)

[Figură 7: Mega Macs PC Bike [7] 10](#_bookmark14)

[Figură 8: Martori [8] 11](#_bookmark16)

[Figură 9: Afișaj electronic [9] 11](#_bookmark17)

[Figură 10: Conector BMW [10] 12](#_bookmark19)

[Figură 11: OBD-I Renault [10] 12](#_bookmark20)

[Figură 12: OBD-I Mercedes-Benz [10] 12](#_bookmark21)

[Figură 13: OBD-I Ford [10] 12](#_bookmark22)

[Figură 14: OBD-I Daewoo [10] 12](#_bookmark23)

[Figură 15: Conector VGA spre fiecare port [10] 12](#_bookmark24)

[Figură 16: Conector OBD-II [11] 13](#_bookmark26)

[Figură 17: Conectori OBD-II J1962 13](#_bookmark27)

[Figură 18: Conector J1962 Type A [12] 13](#_bookmark28)

[Figură 19: Conector J1962 Type B [12] 13](#_bookmark29)

[Figură 20: Semnal PWM [14] 14](#_bookmark32)

[Figură 21: Semnal VPW 15](#_bookmark34)

[Figură 22: Comunicare ISO 9141 15](#_bookmark36)

[Figură 23: Comunicare ISO 14230 16](#_bookmark38)

[Figură 24: Comunicare ISO 15765 16](#_bookmark40)

[Figură 25: Comunicare TCP 17](#_bookmark41)

[Figură 26: Rețeaua electrică a autovehiculului [15] 17](#_bookmark43)

[Figură 27: Structura de comunicații CAN [16] 18](#_bookmark44)

[Figură 28: Comunicație CAN High [17] 19](#_bookmark45)

[Figură 29:Comunicație CAN Low [17] 19](#_bookmark46)

[Figură 30 structura unui mesaj CAN [18] 19](#_bookmark48)

Figură 31: Compunerea codurilor DTC 21

Figură 32: Python Logo [20] 23

[Figură 33: ELM327 vechi 24](#_bookmark60)

[Figură 34: ELM327 nou 24](#_bookmark61)

[Figură 35: Numerotarea pinilor OBD-II [21] 25](#_bookmark62)

[Figură 36: Microprocesorul ELM327 [19] 25](#_bookmark63)

Figură 37: Comunicația utilizator – CAN 26

Figură 38: Rezultate primite 27

Figură 39: Interfață grafică 29

Figură 40: Model de grafic 30

Figură 41: Mesaj de conectare/deconectare 31

Figură 42: Simularea utilizării ELM327 35

Figură 43: Exemplu rulare pentru viteză 36

Figură 44: Exemplu rulare pentru RPM 36

Figură 45:Grafic pentru temperatură apă 37

Figură 46: Grafic pentru presiune în camera de ardere 38

Figură 47: Grafic pentru RPM 38

Figură 48: Grafic pentru km/h 39

Figură 49: Grafic pentru temperatură aer în debitmetru 39

Figură 50: Grafic pentru cantitatea aerului în g/s 40

[Figură 51: Grafic pentru % apăsarea pedalei de accelerație 40](#_bookmark75)