

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

ZÁZNAM, SPRACOVANIE A VIZUALIZÁCIA DIAGNOSTICKÝCH ÚDAJOV
Z TESTOVACÍCH JÁZD NÁKLADNÝCH VOZIDIEL

Bakalárska práca

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

**ZÁZNAM, SPRACOVANIE A VIZUALIZÁCIA DIAGNOSTICKÝCH ÚDAJOV
Z TESTOVACÍCH JÁZD NÁKLADNÝCH VOZIDIEL**

Bakalárska práca

Študijný program: Aplikovaná informatika
Študijný odbor: 2511 aplikovaná informatika
Školiace pracovisko: Katedra aplikovanej informatiky
Školiteľ: Mgr. Pavel Petrovič, PhD.

Bratislava, 2018

Martin Mašek



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Martin Mašek
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: aplikovaná informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Záznam, spracovanie a vizualizácia diagnostických údajov z testovacích jazd nákladných vozidiel
Recording, Processing and Visualization of Diagnostic Data from Test Drives of Trucks

Anotácia: Pracovníci servisno-školiaceho strediska pre vozidlá generácie Euro VI triedy od 10Ton nie sú schopní čítať závady z dátových liniek vozidla počas prevádzky vozidla v testovacích jazdách využívajúc dostupné zariadenia. Cieľom práce je navrhnutie a implementovanie systému, pomocou ktorého sa počas jazd budú na displeji prenosného počítača zobrazovať diagnostické informácie o jazde v podobe upozornení a grafických vizualizácií. Je k dispozícii popis komunikačných protokolov, popis potrieb cieľových používateľov a počas práce stredisko umožní testovanie a ladenie systému na reálnych vozidlách.

Cieľ: Pracovníci servisno-školiaceho strediska pre vozidlá generácie Euro VI triedy od 10Ton nie sú schopní čítať závady z dátových liniek vozidla počas prevádzky vozidla v testovacích jazdách využívajúc dostupné zariadenia. Cieľom práce je navrhnutie a implementovanie systému, pomocou ktorého sa počas jazd budú na displeji prenosného počítača zobrazovať diagnostické informácie o jazde v podobe upozornení a grafických vizualizácií. Je k dispozícii popis komunikačných protokolov, popis potrieb cieľových používateľov a počas práce stredisko umožní testovanie a ladenie systému na reálnych vozidlách.

Literatúra: Luc Maurin: Electronic Architecture P2683, Volvo Group, June 2013.
STM32F103x8, STM32F103xB Datasheet, ST, August 2015.
STM32F101xx, STM32F102xx, STM32F103xx, STM32F105xx and STM32F107xx advanced ARM®-based 32-bit MCUs, Reference Manual, ST, June 2014.

Vedúci: Mgr. Pavel Petrovič, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.
Dátum zadania: 04.11.2016

Dátum schválenia: 07.11.2016

doc. RNDr. Damas Gruska, PhD.
garant študijného programu



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

.....
študent

.....
vedúci práce

Čestné prehlásenie

Čestne prehlasujem, že som prácu vypracoval samostatne pod vedením školiteľa a uviedol som všetky použité zdroje informácií a literatúru.

Bratislava, 2018

Martin Mašek

Pod'akovanie

Ďakujem vedúcemu bakalárskej práce Mgr. Pavel Petrovič, PhD. za odborné vedenie a konzultácie pri písaní tejto práce. Zároveň ďakujem servisno - školiacemu pracovisku za umožnenie využívania ich služieb a cenných informácií pri spracúvaní tejto práce. Mojej rodine ďakujem za morálnu podporu.

Abstrakt

Cieľom tejto bakalárskej práce bolo vytvoriť hardvérovo - softvérové riešenie, pomocou ktorého je možné čítanie závad a vybraných parametrov v reálnom čase „On-line“ zo sietí nákladných vozidiel poslednej generácie. Softvérová aplikácia je určená pre počítače s operačným systémom Windows a vyvíjaná v programovacom jazyku Python.

Na začiatku práce sme analyzovali súčasný stav diagnostikovania chýb na dátových linkách nákladných vozidiel. V ďalších krokoch sme navrhli a implementovali riešenie aktuálnych nedostatkov pri odstraňovaní porúch v servisných podmienkach.

Výsledkom uvedenej analýzy, je doplnkové diagnostické riešenie, zložené zo záznamového zariadenia a počítačovej aplikácie.

Kľúčové slová: CAN zbernica, OBD, SAE J1939, vozidlá.

Abstract

The aim of this bachelor thesis is to create a hardware - software solution, which would allow reading of defects and selected parameters in real time "On - line" from the last generation of data lines in trucks. The software application is designed for Windows - based computers and developed in the Python programming language.

At the beginning of the work we analyzed the current state of diagnosis of errors on the data lines of trucks. In the next steps, we have designed and implemented a solution to the current deficiencies in troubleshooting under service conditions.

The result of this analysis is the additional diagnostic solution, comprised of a recording device and a computer application.

Key words: CAN bus, OBD, SAE J1939, vehicles.

Obsah

Úvod.....	11
Slovník pojmov.....	12
1. Základné východiská práce.....	13
1.1 Vznik komunikačných CAN zberníc.....	15
1.2 Architektúra komunikačnej siete.....	16
1.3 Protokoly a vrstvy.....	17
1.4 CAN v nákladných automobiloch.....	17
1.4.1 Fyzická vrstva.....	18
1.4.2 Linková vrstva.....	19
1.4.3 Sieťová vrstva.....	20
1.4.4 Transportná, relačná, prezentačná vrstva.....	21
1.4.5 Aplikačná vrstva.....	21
1.4.5.1 SAE J1939-71.....	21
1.4.5.2 SAE J1939-73.....	22
1.5 Použité technológie.....	23
1.5.1 Python.....	23
1.5.2 ChibiOS.....	24
1.5.3 STMF103.....	24
1.5.4 LM2596.....	24
1.5.5 MCP2551.....	24
1.5.6 Putty.....	24
1.5.7 Fritzing.....	24
1.6 Analýza existujúcich riešení.....	25
1.6.1 Techtool 2.....	25
1.6.2 Kvaser Memorator.....	26

1.7 Návrh riešenia	26
1.7.1 Návrh hardvérového riešenia	27
1.7.2 Návrh softvérového riešenia	27
2. Výsledky práce	28
2.1 Metódy riešenia:	28
2.1.1 Hardvérová CAN čítačka	28
2.1.2 Prvé načítanie CAN	31
2.1.3 CAN v nákladnom automobile	32
2.1.4 Dáta na BB1	33
2.1.5 Správy s DTC kódom	34
2.1.6 Aplikácia	37
3. Záver	40
4. Použitá literatúra	41
5. Prílohy.....	44

Úvod

Diagnosticke centrum zadavajúcej firmy sa zaoberá otázkou zlepšovania diagnostiky nákladných vozidiel. Z aktuálnych nedostatkov diagnostického softvéru vyplynula potreba zlepšenia diagnostických postupov a výsledok tejto potreby bol pretavený do bakalárskej práce. Náplňou práce bolo:

- Oboznámenie sa so súčasnou problematikou
- Analýza dostupných zdrojov a údajov
- Zhrnutie dosiahnutých výsledkov do návrhu riešenia
- Implementácia riešenia

Cieľom bakalárskej práce je vytvoriť hardvérovo - softvérové riešenie, pomocou ktorého by bolo možné čítanie závad a vybraných parametrov v reálnom čase „On-line“ zo sietí nákladných vozidiel poslednej generácie.



Obr. 1 Volvo FH4



Obr. 2 Renault T

Túto prácu bolo možné vypracovať a realizovať vďaka servisno - školiacemu pracovisku vo Zvolene, ktoré poskytlo potrebné materiály, vozidlá a priestory.

Slovník pojmov

ISO – (International Organization for Standardization) Medzinárodná organizácia pre normalizáciu, je nezávislá mimovládna organizácia koordinujúca štandardizačnú a normalizačnú technickú činnosť v medzinárodnom meradle.

OSI – (Open Systems Interconnection Reference Model) je abstraktný, opis návrhu štruktúry komunikačných a počítačových sieťových protokolov.

SAE – (Society of automotive engineers, SAE international) je profesijné združenie odborníkov z oblasti leteckého, automobilového a dopravného priemyslu do štandardizačnej organizácie.

OBD II – (On-Board Diagnostics) protokol slúžiaci na diagnostiku systémov automobilov. Súčasťou je štandardný 16 pinový OBD konektor.

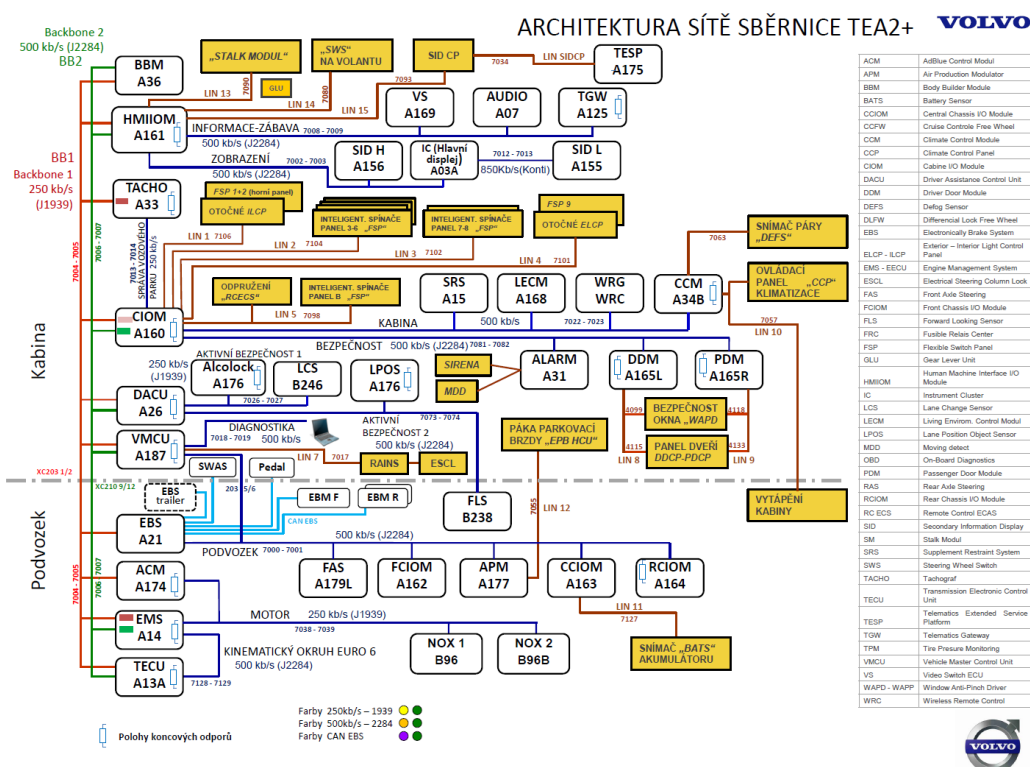
AUTOSAR – (Automotive Open System Architecture) je medzinárodná asociácia pre rozvoj v oblasti automobilov. Vznikla v roku 2003 z iniciatívy automobilových výrobcov a ich dodávateľov s cieľom vytvoriť a zaviesť štandardizovanú softvérovú architektúru zameranú na vozidlá.

DTC – (Diagnostic Trouble Code) štandardizovaný spôsob formátu zápisu chýb

1. Základné východiská práce

Ako sme uviedli v úvode, cieľom práce je vytvoriť hardvérové – softvérové riešenie, pomocou ktorého by bolo možné čítanie zavad a vybraných parametrov v reálnom čase „On-line“ zo sietí nákladných vozidiel poslednej generácie.

Oba typy vozidiel (Obr. 1) a (Obr. 2) s ktorými pracujeme, sú vyrábané na spoločnej koncernovej platforme, s rovnakými agregátmi aj elektronickou architektúrou. V rámci povinného aj dobrovoľného plnenia medzinárodných noriem (ISO, SAE a AUTOSAR) bola do vozidiel aplikovaná sieť niekoľkých komunikačných liniek a podliniek, ktoré umožňujú vo veľkom rozsahu obmieňať funkcionalitu vozidla.



Obr. 3 Schéma architektúry TEA2+

V porovnaní s osobnými vozidlami, kde sa jedná väčšinou o bezpečnostné funkcie (brzdienie, stabilizácia a aktívny tempomat) a komfortné funkcie (vetranie a klimatizácia), sa okrem týchto jedná o ďalšie pridané funkcie a to vo veľmi vysokej variabilite podľa využitia vozidla. Napríklad bezpečnostné funkcie sú oproti osobným vozidlám povinne rozšírené o automatické brzdenie pred prekážkou a sledovanie jazdných pruhov. Samotný brzdový systém je povinne vybavený stabilizačnými systémami ESP a to samozrejme aj so stabilizáciou prívosov a návesov. Do toho je potrebné brať do úvahy množstvo

konfigurácií podvozku od 4x2 až po 10x6 s niekoľkými riadenými a niekoľkými poháňanými nápravami, rôzne typy pružení s premenlivou výškou a premenlivým zaťažením na nápravy. Od komfortnej diaľkovej prepravy až po požiarné špeciály, kde je možné ovládanie vozidla z ľubovoľného miesta alebo na diaľku.

Ďalším faktorom je spotreba medi. Oproti osobným vozidlám, kde sú maximálne vzdialenosti obmedzené na 6 m, pri variabilných dĺžkach nákladných súprav, bežne okolo 16 metrov, vyžaduje zabezpečenie všetkých funkcií kilometre káblov. Vďaka novej architektúre TEA 2+ sa podarilo oproti modelom Euro 5 ušetriť na veľkých vozidlách od 5,5 až do 8 kg medi.

Architektúra TEA2+ je tvorená dvoma hlavnými CAN zbernicami, zvanými „Backbone 1“ (BB1) a „Backbone 2“ (BB2), zhruba 14 podlinkami („Subnety“) a jednoduchými povelovými linkami LIN (Obr. 3).

Vozidlo je bežne vybavené minimálne 24 riadiacimi jednotkami (uzlami) s rôznou úrovňou autonómie. Tento počet môže byť podľa výbavy až dvojnásobný. Jednotlivé uzly pri detegovaní abnormálnych vstupov alebo problémoch zaznamenajú závalu tzv. DTC kódom. Tento kód môže byť zdieľaný ako informácia s ostatnými riadiacimi jednotkami prostredníctvom komunikačných zberníc automaticky, alebo na vyzvanie hlavnej jednotky. Informácia „na vyzvanie“ sa využíva z dôvodu odľahčenia zberníc od zbytočnej komunikácie menej závažných alebo menej dôležitých správ. V spomínaných vozidlách sa na zber a triedenie informácií využíva jednotka VMCU (Vehicle Master Control Unit).



Obr. 4 VMCU - diagnostika

Na testovanie, načítanie závad alebo programovanie vozidla sa využíva dielenský laptop so softvérom „Techtool 2“. Dielenský počítač sa pripája na vozidlo prostredníctvom prepojenia USB - VCOM - OBD (rovnako normovaným, Obr. 4). Oproti osobným vozidlám, zásuvka OBD nie je pripojená na niektorú komunikačnú zbernicu vozidla, ale do jednotky VMCU. Táto jednotka je zabezpečená proti pirátskemu pripojeniu a informácie prostredníctvom OBD komunikuje len na základe výzvy programu Techtool. Navyše, závady sú načítané opäť len na základe vyzvania diagnostického softvéru dielenského počítača.

Načítanie závad Techtool vykoná na základe príkazu „načítať závady“ (Obr. 5 tlačidlo „Obnoviť“).



Obr. 5 Techtool - Načítanie závad

Techtool odosiela príkaz na načítanie chýb do VMCU, ktorá odošle príkaz do sietí vozidla. Odpovede vo formáte výpisu DTC závad sú následne zobrazené na monitore. Uvedený proces trvá približne 6 a viac sekúnd. Ak sa počas testovania vozidla zmení status niektorej riadiacej jednotky, alebo vznikne / zanikne závada, Techtool ju nezobrazí, pokiaľ nie je opätovne zadán príkaz „načítať závady“. Tieto načítané závady nie sú správne zoraditeľné podľa časov vzniku týchto chýb a je teda zložité určiť, ktorá chyba vznikla skôr. Tento nedostatok nás viedol k požiadavke, vytvoriť hardvérovo - softvérové riešenie, ktoré by vedelo okamžite zobrazit' závady, komunikované na hlavných zberniciach vozidla.

1.1 Vznik komunikačných CAN zberníc

CAN zbernica [3] je robustná norma, ktorá umožňuje mikroprocesorom a zariadeniam komunikovať medzi sebou bez hostiteľského počítača. Protokol je založený na správach, pôvodne navrhnutý pre multiplexové [multiplex] elektrické vedenie v automobiloch.

V automobilovom priemysle začala vývoj vo februári v roku 1983 spoločnosť Robert Bosch GmbH [4]. Protokol bol oficiálne uvedený na konferencii SAE v 1986. Bosch vytvoril niekoľko CAN špecifikácií, najrozšírenejšia CAN 2.0 [3] bola vytvorená v 1991. Kvôli udržaniu CAN zariadení spätne kompatibilných bola verzia CAN 2.0 rozdelená na CAN 2.0A a CAN 2.0B. Formát správy v CAN 2.0A je identický so starším 11-bitovým identifikátorom správy (štandardný CAN), CAN 2.0B používa doplnkové 18 bitové

rozšírenie identifikátora, ktorý s pôvodným tvorí dokopy 29 bitový identifikátor CAN správy (rozšírený CAN). CAN štandard je neustále rozširovaný, v roku 2012 vydal Bosch štandard CAN FD 1.0 s flexibilnou rýchlosťou prenosu dát a odlišným formátom dátového rámca umožňujúceho použiť rozdielnu dĺžku údajov. CAN FD 1.0 je spätne kompatibilný s CAN 2.0, zariadenia podporujúce CAN FD 1.0 môžu koexistovať so zariadeniami CAN 2.0 v rovnakej sieti.

Zbernica CAN je jedným z piatich protokolov používaných v palubnej diagnostike automobilov OBD II.

1.2 Architektúra komunikačnej siete

CAN je štandard pre sériovú zbernicu slúžiacu pre pripojenie viacerých elektronických riadiacich jednotiek (ECUs) zvaných aj uzly (nodes) [3]. Na komunikáciu sú potrebné 2 a viac uzlov. Zložitosť uzla sa môže pohybovať od jednoduchého vstupno - výstupného zariadenia (I/O) až po vstavaný počítač s rozhraním CAN a sofistikovaným softvérom. Uzol môže byť aj brána umožňujúca komunikáciu prostredníctvom USB alebo Ethernetu so zariadeniami v sieti CAN.

Každý uzol obsahuje [1, 2, 3]:

- Centrálnu procesorovú jednotku (CPU) - spracúva prijaté správy, rozhoduje aké správy budú vysielané.
- Snímače, ovládacie prvky, akčné členy - môžu byť pripojené a ovládané prostredníctvom CPU.
- Radič CAN (zodpovedá ISO 11898-1), ktorý vykonáva:
 - o Prijatie správy – ukladá prijaté sériové bity, kým nie je prijatá celá správa, ktorú následne načíta CPU.
 - o Odosielanie správy – odošle správu prijatú od CPU na zbernicu, ak je zbernica voľná.
- Transceiver – Prevodník (zodpovedá ISO 11898-2 alebo ISO 11898-3) pri prijímaní a vysielaní dát prevádza napäťové signály (Obr. 6, 7) na logické a naopak.

Každý uzol môže odosielať aj prijímať správy, ale nie zároveň.

1.3 Protokoly a vrstvy

Podobne ako mnoho iných sieťových protokolov, aj CAN môže byť rozložený do viacerých vrstiev [3] vzhľadom na referenčný OSI model. Pri nasadení zbernice v praxi, je potrebné špecifikovať pravidlá konkrétnym protokolom podľa oblasti použitia. Toto bližšie špecifikovanie (využitie) je definované predovšetkým aplikačnou vrstvou. Napríklad:

- MilCAN používaný vo vojenských vozidlách
- CANopen - priemyselná automatizácia
- ISOBUS (ISO 11783) - poľnohospodárske zariadenia
- ISO-2000 alebo ISO K-Line, tiež KWP v automobiloch
- V nákladných vozidlách a autobusoch je CAN špecifikovaný normami SAE a ISO:
 - o SAE J1939
 - o BUS 1587, J1708
 - o SAE J2284
 - o ISO 11898

1.4 CAN v nákladných automobiloch

Hlavné dátové linky použité vo vozidlách s ktorými pracujeme, sú zvané „Backbone 1“ a „Backbone 2“ (Obr. 3, [5, 6]). Podobne ako mnoho iných sieťových protokolov, aj CAN môže byť rozložený do viacerých vrstiev (Tab. 1), v súlade s referenčným modelom OSI.

Backbone 1 (BB1) [5, 6] je definovaná normou SAE J1939 a má rýchlosť 250 kbit/s. Použitím tejto normy v architektúre TEA2+ bola zabezpečená správna integrácia predošlých komponentov (riadiacich jednotiek a pod.). Zároveň bola táto norma použitá aby sa vyhovelo priemyselným štandardom (AUTOSAR) a kontrolným úradom. Dáta na BB1 sú použitím SAE J1939 verejné, z čoho plynú viaceré výhody. Zákazníci získali detailnejšie informácie o prevádzke vozidla. Stanice technickej, emisnej kontroly a policajné zložky, majú v prípade potreby prístup k informáciám o technickom stave vozidla.

Backbone 2 (BB2) [5, 6] je definovaná normou SAE J2284 a má rýchlosť 500 kbit/s. Táto linka má súkromné kódovanie definované výrobcom vozidla. Norma SAE J2284 opisuje iba potrebné časti fyzickej a linkovej vrstvy, aplikačná vrstva je definovaná výrobcom vozidla. BB2 okrem silnej re-cirkulácie dát slúži aj na odosielanie chybových správ.

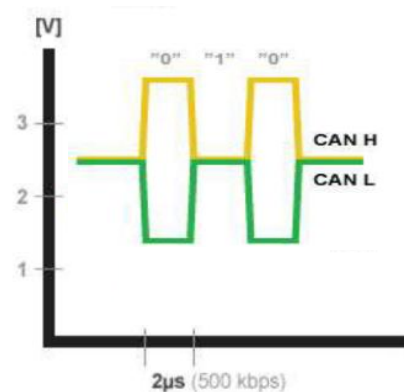
VRSTVA	BB1+BB2	BB1+BB2	BB1	BB2
Aplikačná vrstva	-	-	SAE J1939-73 SAE J1939-71	Výrobca vozidla
Prezentačná vrstva	-	-	-	-
Relačná vrstva	-	-	-	-
Transportná vrstva	-	-	-	-
Sieťová vrstva	-	-	SAE J1939-31	SAE J2284-3
Linková vrstva	Bosh CAN Specification 2.0	ISO 11898-1	SAE J1939-21	SAE J2284-3
Fyzická vrstva	Bosh CAN Specification 2.0	ISO 11898-2	SAE J1939-15	SAE J2284-3

Tab. 1 Tabuľka jednotlivých vrstiev podľa OSI modelu a protokoly definujúce jednotlivé vrstvy

1.4.1 Fyzická vrstva

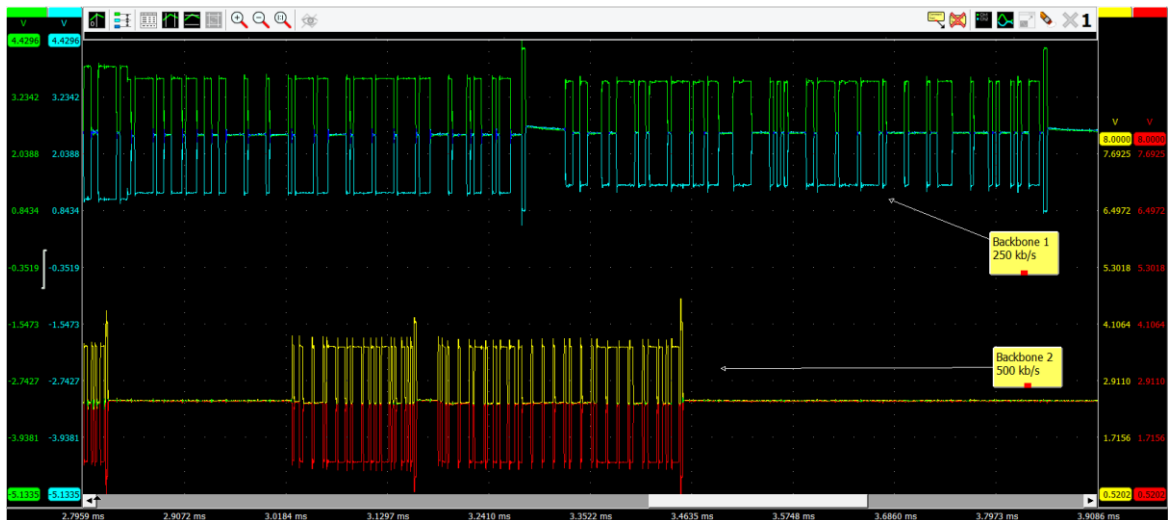
V samotnom protokole CAN od spoločnosti Bosh nie je spôsob prenosu jednotlivých bitov definovaný. Protokol definuje iba dva stavy, dominantný (logická 0) a recesívny (logická 1). Pokojový stav (idle) zbernice je definovaný ako recesívny. Tieto stavy musí byť fyzická vrstva schopná reprezentovať a preniesť. Normy pre fyzickú vrstvu boli definované v ISO 11898-2 a ISO 11898-3.

ISO 11898 definuje elektrické vlastnosti prevodníka (transceiver), zároveň princípy časovania, synchronizácie a kódovania jednotlivých bitov. Zbernica je tvorená dvoma vodičmi (označované CAN_H (CAN High) a CAN_L (CAN Low)), kde je dominantná a recesívna úroveň definovaná rozdielom napätí (Obr. 6, 7) týchto vodičov. Norma definuje veľkosť rozdielového napätia pre **recesívnu úroveň** (logická 1) $V_{diff} = 0 \text{ V}$ (CAN_L, CAN_H = 2,5V)



Obr. 6 Rozdielové napätia

a **dominantnú úroveň** (logická 0) $V_{diff} = 2\text{ V}$ ($CAN_L = 1,5\text{ V}$, $CAN_H = 3,5\text{ V}$). Pre elimináciu šumu je zbernica na oboch koncoch zakončená 120 Ohm odpormi.



Obr. 7 Záznam komunikácie prostredníctvom osciloskopu

Fyzická vrstva je vo vozidlách s ktorými pracujeme ďalej špecifikovaná časťou SAE 1939-15, ktorá stanovuje rýchlosť zbernice na 250 kbit/s. Všetky uzly sú vzájomne prepojené prostredníctvom zbernice zloženej z dvoch vodičov, prenosovým médium je UTP (Un-Shielded Twisted Pair, Netienená krútená dvojlinka).

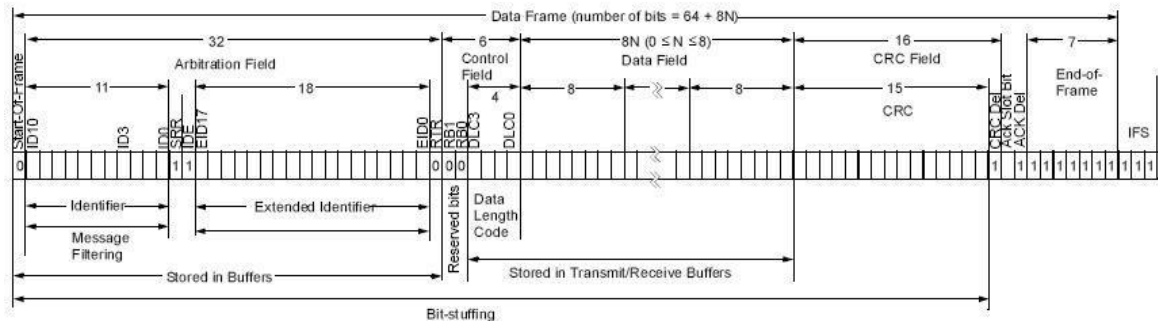
1.4.2 Linková vrstva

CAN používa bezstratovú bitovú arbitrážnu metódu rozlíšenia sporu pri prenose údajov. Táto metóda vyžaduje aby boli naraz synchronizované všetky uzly v sieti CAN so vzorkou každého bitu naraz. Preto býva sieť CAN označovaná ako synchronná, čo je nepresný termín pretože sú dáta prenášané v asynchrónnom formáte bez hodinového signálu.

Podobne ako v OSI modely aj v protokole CAN je linková vrstva [3, 1] rozdelená na pod vrstvu **LLC** a **MAC**:

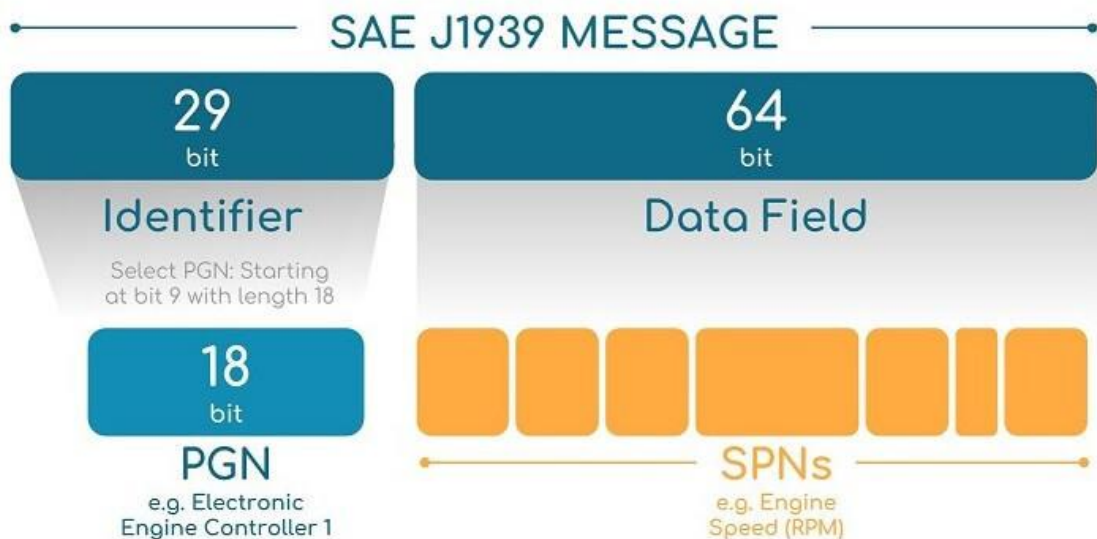
- **MAC** (Medium Access Control) reprezentuje jadro protokolu CAN. Úlohou jadra je vykonávať kódovanie dát, vkladať bity (bit stuffing/destuffing), riadiť prístup uzlov na zbernica podľa rozlišovania priorít správ, detekciu chýb, hlásenie chýb a potvrdzovanie správne prijatých správ.

- **LLC** (Logical Link Control) je pod vrstva riadenia dátového spoja, v CAN to znamená filtrovanie prijatých správ (*Acceptance Filtering*) a hlásenie o preťažení (*Overload Notification*).



Obr. 8 Kompletná správa CAN 2.0B - rozšírený identifikátor

SAE J 1939-21 definuje bližšie linkovú vrstvu. Popisuje obsah a používanie 29 bitového identifikátora. Súčasťou identifikátora je PGN (Parameter Group Number) - číslo skupiny parametrov, ktoré pomocou noriem SAE J1939-71 a SAE J1939-73 popisuje parametre uložené v dátovej časti správy pomocou SPN (Suspect Parameter Number) - konkrétne číslo parametra.



Obr. 9 Správa podľa SAE J 1939

1.4.3 Sieťová vrstva

Sieťová vrstva nie je špecifikovaná [11] spoločnosťou BOSH, ani ISO normou. Špecifikuje ju SAE J1939-31, tak aby poskytovala otvorený systém prepojenia elektronických palubných systémov a umožňovala im navzájom komunikovať

poskytovaním štandardnej architektúry. Definuje požiadavky a služby potrebné pre elektronické zariadenia (ECUs, riadiace jednotky vozidla). Taktiež definuje rôzne typy sieťových prepojení ECU a funkcie, ktoré poskytujú. Informácie v tomto dokumente sa vzťahujú len na ECU, ktoré sú určené na poskytovanie sieťových služieb. Nie je potrebné, aby ECU poskytovala niektoré z týchto služieb, na to, aby bola v súlade s protokolom SAE J1939.

1.4.4 Transportná, relačná, prezentačná vrstva

Vrstvy nie sú v SAE J1939 ani SAE J2284 definované.

1.4.5 Aplikačná vrstva

1.4.5.1 SAE J1939-71

Táto špecifikácia popisuje štandardné správy obsahujúce informácie o stave komponentov vozidla, ktoré daná riadiaca jednotka zdieľa s ostatnými riadiacimi

pgn61444 - Electronic Engine Controller 1 - EEC1 -

Transmission Repetition Rate:	engine speed dependent		
Data Length:	8 bytes		
Data Page:	0		
PDU Format:	240		
PDU Specific:	4		
Default Priority:	3		
Parameter Group Number:	61444 (00F004 ₁₆)		
Bit Start Position /Bytes	Length	SPN Description	SPN
1.1	4 bits	Engine Torque Mode	899
2	1 byte	Driver's Demand Engine - Percent Torque	512
3	1 byte	Actual Engine - Percent Torque	513
4-5	2 bytes	Engine Speed	190
6	1 byte	Source Address of Controlling Device for Engine Control	1483
7.1	4 bits	Engine Starter Mode	1675
8	1 byte	Engine Demand - Percent Torque	2432
Engine related parameters			

Obr. 10 SAE J1939-71 PGN 61444 (F004) - pozície a čísla parametrov (SPN)

spn190 - Engine Speed - Actual engine speed which is calculated over a minimum crankshaft angle of 720 degrees divided by the number of cylinders.

Data Length:	2 bytes
Resolution:	0.125 rpm/bit , 0 offset
Data Range:	0 to 8,031.875 rpm
Type:	Measured
Suspect Parameter Number:	190
Parameter Group Number:	[61444]

Obr. 11 Spôsob dekódovania údajov SPN 190, ktorý je súčasťou PGN F004

jednotkami. Špecifikácia obsahuje zoznam PGN a pridružených SPN (viď. Obr. 10, 11), pozície SPN v dátovej správe a spôsob ich konvertovania.

1.4.5.2 SAE J1939-73

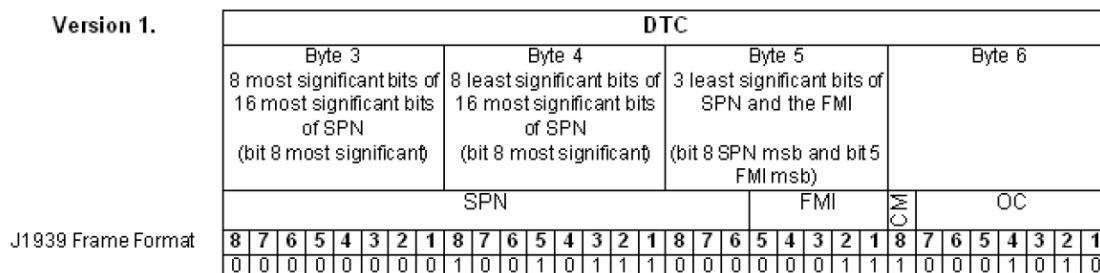
Špecifikácia popisuje PGN, ktoré sú určené na odosielanie DTC kódov a spôsoby kódovania a konvertovania dátovej časti správ, v ktorej sa nachádzajú DTC kódy.

Row #	Function	PGN	Acronym	SPN #	Required By Regulation (See Table 2)	Description
Diagnostic Services						
1	Read DTCs	65226	DM1		A,B,C,E,F,G,H,J,K,L	All Active DTCs
2	Read DTCs	65236	DM12		A,B,H,J,K,L	Emissions related active DTCs
3	Read DTCs	64949	DM23		B,H,J,L	Previously Active Emission Related DTCs
4	Read Pending DTCs	65231	DM6		A,B,E,F,G,H,J,K,L	Emission related pending DTCs
5	Read All Pending DTCs	64898	DM27			All pending DTCs inclusive of Emissions and Non-Emissions Related

Obr. 12 SAE J1939-73 Zoznam PGN a priradené typy DTC

Given:

SPN 1208 = $4B8_{16}$ = 000 00000100 10111000₂ (19 bits)
 FMI 3 = 3_{16} = 00011₂ (5 bits)
 OC 10 = A_{16} = 0001010₂ (7 bits)
 CM = 0_2 (1 bit)



Obr. 13 SAE J1939-73 Jeden zo spôsobov konverzie

Tvar a zloženie samotného DTC kódu (DTC P107E85) popisuje Obr. 14.

Example: P107E85

P	1	07E	85
P - Powertrain C - Chassis B - Body U - Network	0 - ISO/SAE 1 - Manufacturer 2 - ISO/SAE 3 - ISO/SAE	Fault	Failure type
P, C, B, U	0-3	0-FFF	0-FF
4	4	4096	256
00	00	0000 0000 0000	0000 0000
← High byte		← Middle byte	← Low byte

Obr. 14 Zloženie DTC chyby

1.5 Použité technológie

1.5.1 Python

Počas prípravy implementácie aplikácie, ktorá pobeží pod operačným systémom Windows, som sa rozhodol použiť programovací jazyk Python [15], nakoľko má veľmi jednoduchú a dobre čitateľnú syntax. Ako vývojové prostredie som použil PyCharm [16], ktorý oproti vstavanému Idle poskytuje okrem iného správu súborov, inteligentné dopĺňanie kódu, kontrolu, zvýrazňovanie a opravu chýb.

Použité knižnice:

- PyQt5 [17] - je knižnica Qt prispôbená pre programovací jazyk Python, určená okrem iného aj na vytváranie GUI (grafického používateľského rozhrania). Výhodou knižnice je vývojová aplikácia Qt Designer, ktorá uľahčuje úvodné nadizajnovanie aplikácie a následnú transformáciu dizajnu do spustiteľného Pythonového kódu. Knižnica taktiež podporuje správu vlákien (threads), potrebnú pre efektívny beh výslednej aplikácie.

- PySerial [18] - obsahuje metódy na prácu so sériovým portom prostredníctvom ktorého sa pripája jednočipový počítač STM
- Pickle [19] - modul implementuje protokoly pre serializáciu a de-serializáciu objektov
- OpenPyxl [20] - knižnica umožňujúca čítanie a zápis Excel súborov

1.5.2 ChibiOS

Je [21] kompaktný a rýchly real-time micro-operačný systém, ktorí beží na procesore použitom na platforme mikropočítačov STM. Tento systém vie pracovať s radičom CAN zbernice implementovaným v týchto procesoroch. Na programovanie mikropočítača STM sme použili vývojové prostredie ChibiStudio [22], ktoré obsahuje databázu programov, ktoré ilustrujú jednotlivé metódy, ktoré je možné použiť. Obsahuje aj vzorový kód na programovanie STM na čítanie CAN. Programovacím jazykom je C [23].

1.5.3 STMF103

Jedno-čipový mikropočítač [24] s procesorom Cortex-M3 pracujúcim na frekvencii 72MHz. Procesor implementuje interface pre CAN zbernicu, kompatibilný s CAN 2.0B.

1.5.4 LM2596

Regulátor napätia [25].

1.5.5 MCP2551

CAN Prevodník [26], transformuje napät'ovú (V) logiku CAN na logickú (0/1). Implementuje ISO 11898.

1.5.6 Putty

Nástroj [27] na prácu zo sériovým portom.

1.5.7 Fritzing

Nástroj [28] na tvorbu hardvérových schém.

1.6 Analýza existujúcich riešení

Na testovanie, načítanie závad alebo programovanie vozidla sa využíva dielenský laptop so softvérom „Techtool 2“. V prípade potreby detailnejšej analýzy komunikácie na dátových linkách je využívané záznamové zariadenie Kvaser Memorator.

1.6.1 Techtool 2

Závady vozidla v podobe DTC kódov sú načítavané len na základe výzvy programu Techtool. Načítanie závad neprebíha priebežne v zvolenom časovom rámci ale „jednorazovo“. Načítanie závad Techtool vykonáva na základe príkazu „načítať závady“ (Obr. 5), kedy odosiela príkaz na získanie chybových kódov do VMCU (Obr. 3, Obr. 4), ktorá odošle tento príkaz ďalej do sietí vozidla. Riadiace jednotky odošlú odpovede vo formáte výpisu závad, ktoré sú následne zobrazené na monitore. Uvedený proces trvá približne 6 a viac sekúnd. Po tom čo riadiace jednotky odošlú všetky dostupné aktuálne aktívne sa vyskytujúce chyby a neaktívne chyby (vyskytujúce sa v predošlom časovom rozmedzí), Techtool ukončí načítavanie chýb (Obr. 15).

Řídicí jednotka	DTC	Stav	Počet	První výskyt	Poslední výskyt
Digitální tachometr, DTCO (DTACHO)	B120100: Driving without valid driver card, Žádné další informace	Neaktivní	75	28.2.2018 13:30:49	21.4.2018 9:12:53
Digitální tachometr, DTCO (DTACHO)	P050018: Wheel-Based Vehicle Speed, Circuit Voltage Below Threshold	Neaktivní	2	9.4.2018 16:30:20	9.4.2018 17:17:00
Digitální tachometr, DTCO (DTACHO)	P050064: Wheel-Based Vehicle Speed, Závada přijatelnosti signálu	Neaktivní	2	9.4.2018 16:29:55	9.4.2018 17:16:34
Digitální tachometr, DTCO (DTACHO)	P050068: Wheel-Based Vehicle Speed, Event Information	Neaktivní	1	7.3.2018 9:25:00	7.3.2018 9:25:00
Digitální tachometr, DTCO (DTACHO)	U30001C: Electronic Control Unit, Napětí obvodu mimo rozsah	Neaktivní	1	1.3.2018 11:20:53	1.3.2018 11:20:53
ECU převodovky (TECU)	P105229: Poloha válců spojky, Neplatný signál	Neaktivní	1	21.4.2018 12:07:45	21.4.2018 12:07:51
ECU převodovky (TECU)	P106418: Rychlý řadicí ventil spojky (VAFE), Proud obvodu pod prahovou hodnotou	Neaktivní	1	21.4.2018 12:07:44	21.4.2018 12:07:48
ECU převodovky (TECU)	P106518: Ventil pomalého vypnutí spojky (VASE), Proud obvodu pod prahovou hodnotou	Neaktivní	1	21.4.2018 12:07:44	21.4.2018 12:07:48
ECU převodovky (TECU)	P106618: Ventil rychlého vypnutí spojky (VAFD), Proud obvodu pod prahovou hodnotou	Neaktivní	1	21.4.2018 12:07:44	21.4.2018 12:07:48
ECU převodovky (TECU)	P106718: Ventil pomalého rozpojení spojky (VASD), Proud obvodu pod prahovou hodnotou	Neaktivní	1	21.4.2018 12:07:44	21.4.2018 12:07:48
Electronic Brake System (EBS)	C100314: Trailer Control Module Backup Valve, Obvod zkratován vůči kostře nebo přerušen	Neaktivní	2	9.4.2018 16:25:29	9.4.2018 17:16:59
Electronic Brake System (EBS)	C107287: Foot Brake Module Electrical Link, Chybějící hlášení	Neaktivní	2	9.4.2018 16:28:43	21.4.2018 12:11:07

Obr. 15 Techtool - Načítané DTC chyby

Ak sa počas testovania vozidla zmení status niektorej riadiacej jednotky, alebo vznikne/zanikne závada, Techtool ju nezobrazí, pokiaľ nie je opätovne zadaný príkaz „načítať závady“. Tieto načítané závady taktiež nie sú správne zoraditeľné podľa časov vzniku týchto chýb a je teda zložité určiť, ktorá chyba vznikla skôr.

Určenie poradia chýb je v niektorých prípadoch dôležité pre správne a rýchle riešenie opravy vozidla. Zdieľanie údajov medzi riadiacimi jednotkami, môže v určitých prípadoch viesť reťazovému vzniku chýb.

Príklad:

Riadiaca jednotka motora (ECU) odosiela údaje o otáčkach zotrvačníka do riadiacej jednotky prevodovky (TECU). V prípade poruchy snímača otáčok, ECU motora nahlási relevantnú DTC závalu, ale motor dokáže s menšími obmedzeniami bežať ďalej. Údaj o otáčkach motora však využíva aj TECU, pre ktorú je hodnota otáčok kľúčová a bez nej nedokáže ďalej pracovať v automatickom režime. Takže okamžite hlási naraz 2 závady, závalu komunikácie a poruchu radenia. Pri načítavaní chýb pomocou Techtoolu technik nevie určiť, ktorá chyba nastala skôr, nakoľko sa všetky objavili v rovnakom čase a teda či má začať riešiť vadný senzor na motore alebo poruchu prevodovky. Tento nedostatok nás viedol k požiadavke, vytvoriť hardvérovo-softvérové riešenie, ktoré by vedelo okamžite zobrazovať závady, ktoré sú v danom momente komunikované na hlavných zberniciach vozidla. Vďaka tomu bude možné určiť poradie vzniku chýb.

1.6.2 Kvaser Memorator

V prípade výskytu závažnej chyby a nemožnosti jej správneho vyhodnotenia v servise, je použité záznamové zariadenie Kvaser Memorator (Obr. 16), ktoré vytvorí záznam komunikácie na dátových linkách vozidla. Tento záznam je následne odosielaný do servisnej centrály kde je analyzovaný. Odpoveď môže trvať niekoľko dní až týždne, pričom vozidlo počas tejto doby, nie je prevádzky schopné, čo vedie k nespokojnosti zákazníka.

Našou snahou bolo uvedené problémy vyriešiť.



Obr. 16 Kvaser Memorator

1.7 Návrh riešenia

Na základe doterajších informácií, navrhujeme vytvoriť také zariadenie, ktoré spomínané nedostatky odstráni. Zariadenie bude schopné zaznamenávať komunikáciu na zbernici v užívateľom zvolenej dĺžke trvania a zároveň odosielať túto komunikáciu na ďalšie

spracovanie do aplikácie. Riešenie danej problematiky je možné rozdeliť na hardvérové a softvérové riešenie.

1.7.1 Návrh hardvérového riešenia

Navrhujeme vytvoriť zariadenie, pomocou ktorého budeme schopní v reálnom čase zachytávať komunikáciu prebiehajúcu na hlavných dátových linkách vozidla (BB1, BB2) a odosielať túto komunikáciu vo vhodnom formáte do aplikácie v počítači.

1.7.2 Návrh softvérového riešenia

Navrhujeme vytvoriť aplikáciu, pre počítač s operačným systémom Windows, nakoľko je tento systém používaný v cieľovej oblasti budúceho použitia.

Na základe konzultácie s odberateľom a servisnými mechanikmi, by ovládanie aplikácie malo byť možné prostredníctvom klávesnice.

Dizajn aplikácie by mal byť jednoduchý, naučiteľný, mal by zobrazovať informácie o tom v akom stave sa momentálne aplikácia nachádza. Aplikácia bude zachytávať údaje od hardvérového riešenia, spracúvať ich a následne v čitateľnom formáte zobrazovať užívateľovi

2. Výsledky práce

Výsledkom tejto práce je produkt, ktorý má pomáhať pri diagnostikovaní porúch nákladných vozidiel. Naším cieľom je zlepšiť a urýchliť dobu potrebnú na odhalenie poruchy vozidla a jej následné odstránenie. Produkt sa skladá z CAN čítačky - hardvérového zariadenia, ktoré číta CAN zbernicu a softvéru, ktorý beží v počítači a spracováva údaje z čítačky. Na dosiahnutie cieľa sme sa oboznámili so súčasnou problematikou, súčasným stavom porúch a používanými postupmi ich diagnostikovania a odstraňovania. Teoreticky sme sa oboznámili s dátovými linkami v daných nákladných vozidlách a príslušnými špecifikáciami, ktoré ich definujú. So zadávateľom sme konzultovali problematiku a návrh možných riešení.

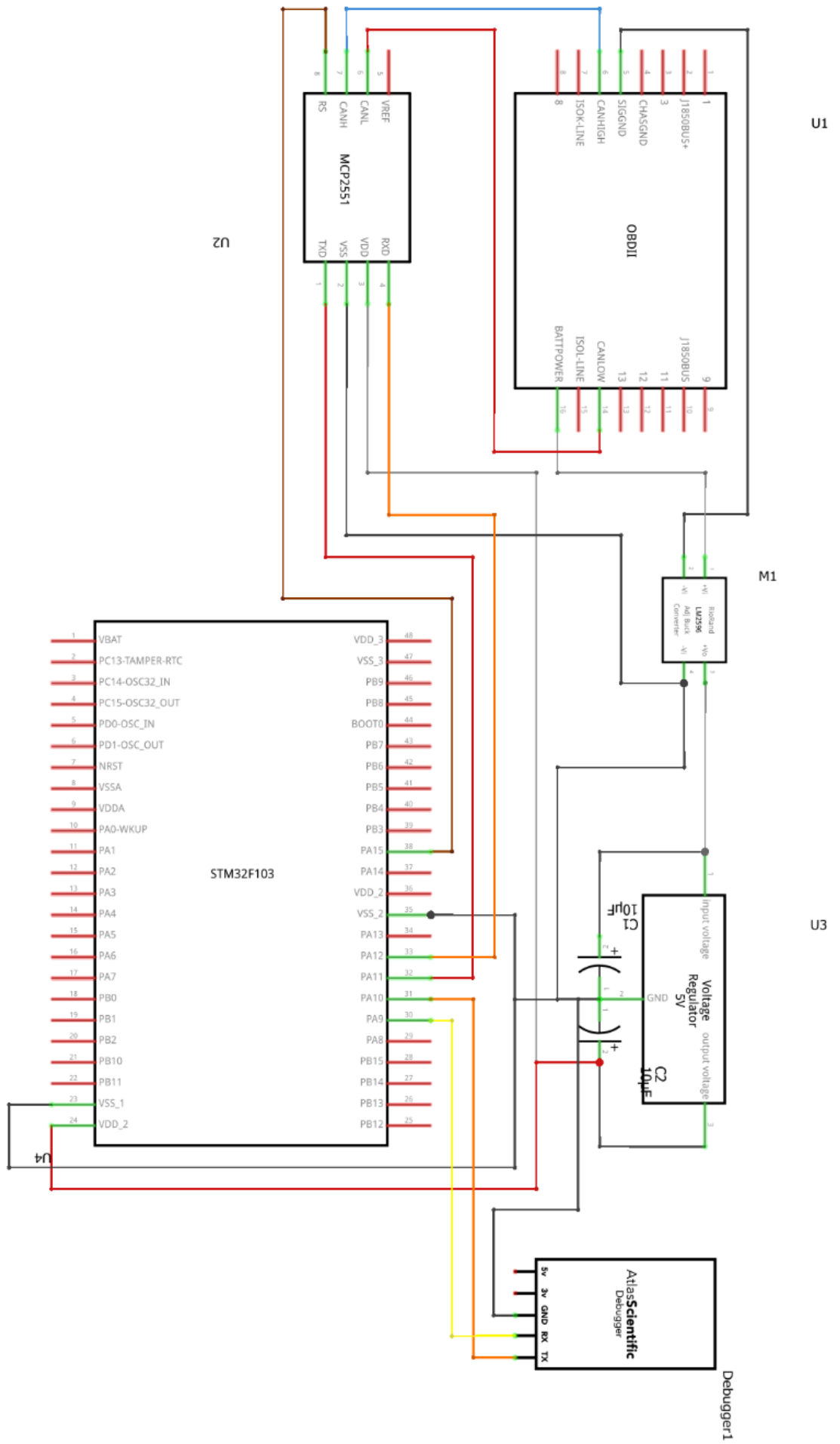
2.1 Metódy riešenia:

2.1.1 Hardvérová CAN čítačka

Na základe špecifikácií dátových liniek použitých vo vozidlách sme vybrali komponenty potrebné na vytvorenie hardvérovej CAN čítačky a pomocou softvéru Fritzing vytvorili schému zapojenia (Obr. 17).

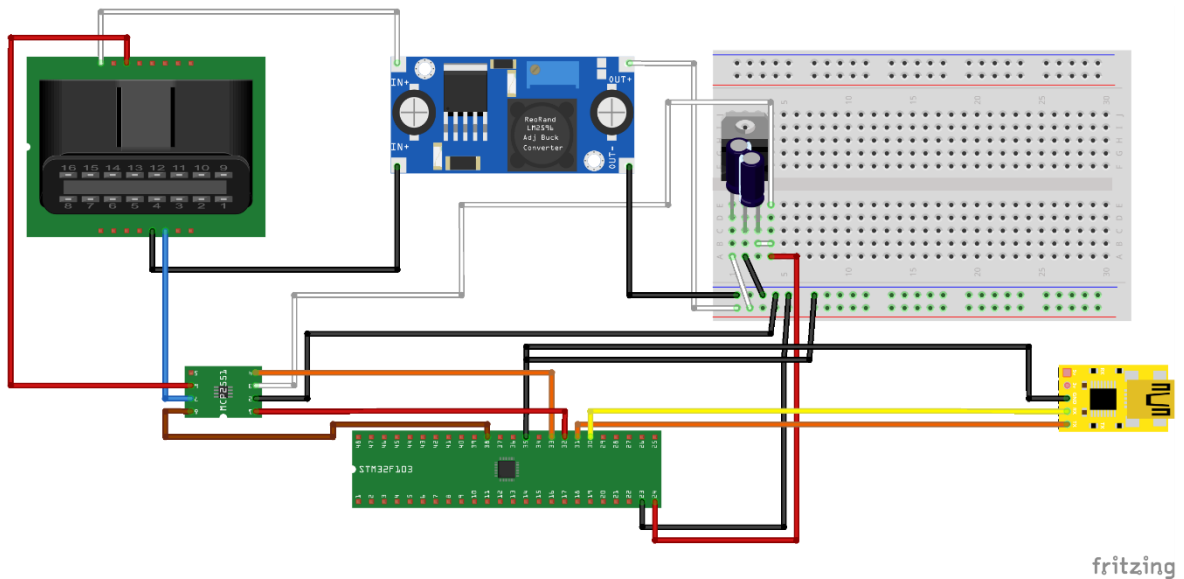
Obvod je napájaný z elektrickej sústavy vozidla, z OBD konektora. V elektrickej sústave nákladného vozidla je napätie 28V, komponenty čítačky pracujú s napájacím napätím 5V. Vzhľadom na to sme použili regulátor LM2596, s nastaviteľným výstupným napätím. Pre dodatočnú ochranu je za týmto regulátorom, zaradený ešte jeden SMD 5V regulátor.

Na transformovanie napät'ovej logiky zbernice na logickú je použitý prevodník MCP2551. Prevodník podporuje rýchlosti zbernice až do 1 Mbit/s, implementuje normu ISO 11898, preto je v súlade so špecifikáciami dátových liniek (BB1 a BB2), použitých vo vozidlách s ktorými pracujeme. Prevodník je chránený proti skratu vodičov (CAN_High, CAN_Low) zbernice na kostru, aj batérové napätie nákladného auta (28V).



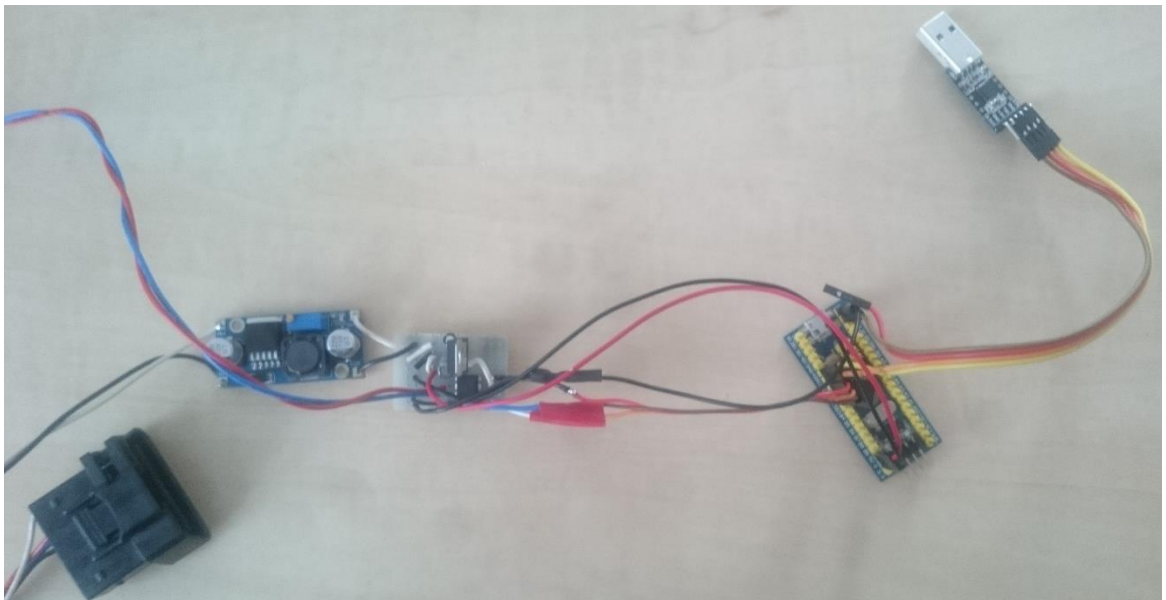
Obr. 17 Schéma zapojenia CAN čítačky

Logický signál z tohto prevodníka spracúva mikropočítač STM32F103 prostredníctvom rozhrania (interface) CAN implementovanom v procesore. Zachytené správy vysiela prostredníctvom sériového portu vyvedeného do USB prevodníka.

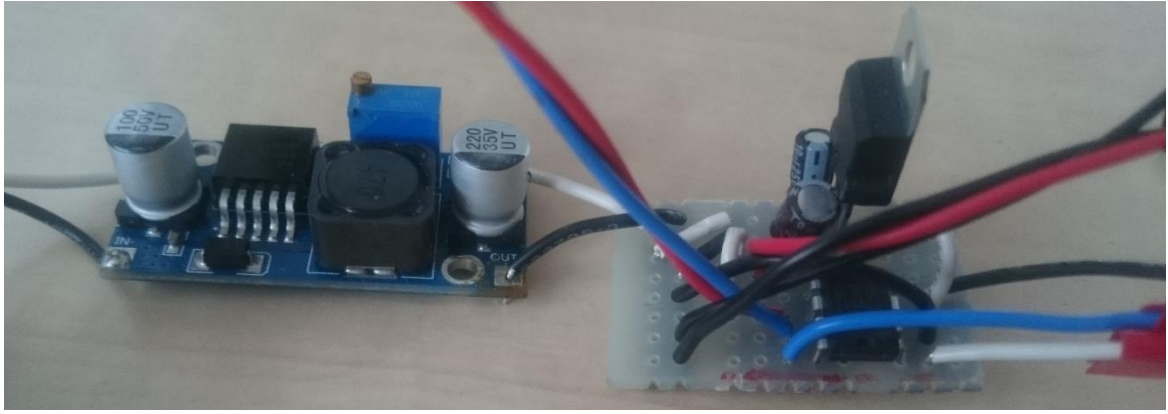


Obr. 18 Montážna vizualizácia schémy

Po zaobstaraní všetkých komponentov, sme vyrobili hardvérový prototyp CAN čítačky (Obr. 19, 20, 21).

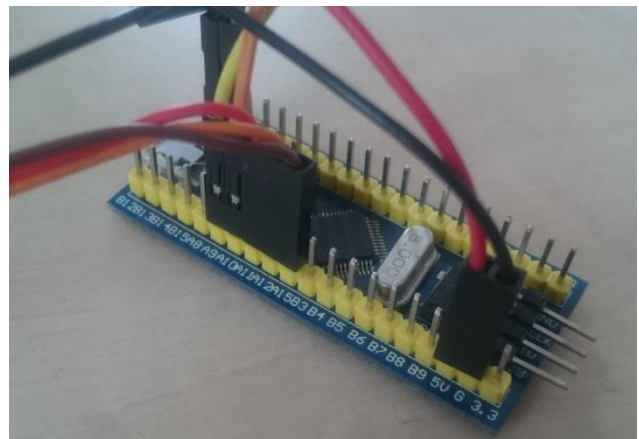


Obr. 19 Hardvérový prototyp CAN čítačky



Obr. 20 Detail CAN čítačky, vľavo regulátor LM2596, vpravo montážna doska s prevodníkom MCP2551

Na programovanie mikropočítača STM sme použili ChibiOs a vývojové prostredie Chibistudio, ktoré obsahuje vzorový kód v programovacom jazyku C, ilustrujúci programovanie CAN. Vzorový kód sme upravili pre naše potreby. Pridaním výpisu jednotlivých častí načítavaných správ v decimálnom tvare a za komentovaním nepotrebných častí kódu.



Obr. 21 Detail mikropočítača STMF103

2.1.2 Prvé načítanie CAN

Prvý pokus o načítanie komunikácie prebiehal v osobnom automobile. CAN čítačkou sme pripojili do OBD konektora a prostredníctvom Putty sme sledovali výpisy. V osobnom automobile je zbernica CAN priamo vyvedená do OBD konektora a jej rýchlosť je 500kbit/s. Komunikáciu sme ale nezachytili. Pomocou osciloskopu sme skontrolovali komunikáciu na zbernici a zapojenie prevodníka MCP2551. Komunikácia prebiehala, prevodník bol zapojený správne a správy posielal do STM v bitovej logike. Bolo treba nastaviť správne hodnoty zbernice v programe STM. To sa po niekoľkých pokusoch s náhodnými číslami nakoniec podarilo a komunikácia sa rozbehla. Správne nastavenie parametrov SJW, TS1, TS2, BRP - Obr. 22.

```

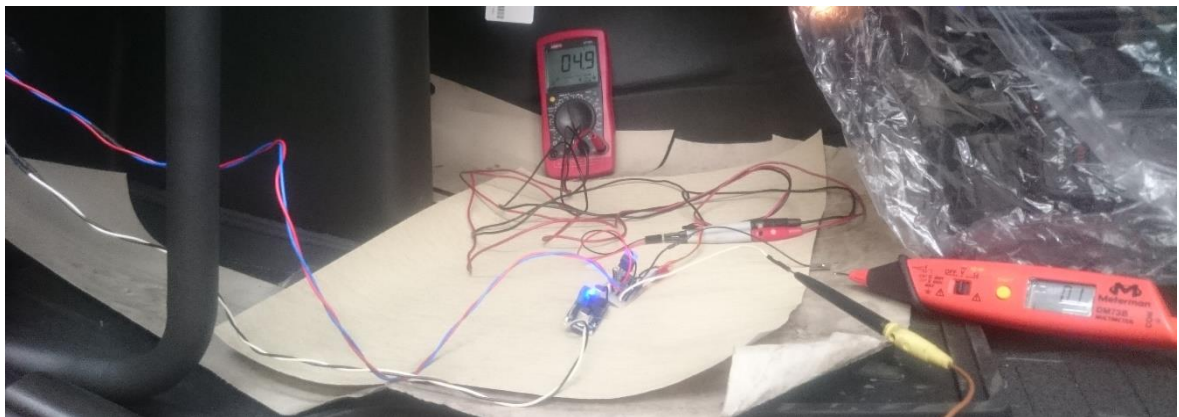
38 static const CANConfig cancfg = {
39     CAN_MCR_ABOM | CAN_MCR_AWUM | CAN_MCR_TXFP,
40     CAN_BTR_SJW(0) | CAN_BTR_TS2(1) |
41     CAN_BTR_TS1(8) | CAN_BTR_BRP(5)

```

Obr. 22 ChibiStudio parametre pre CAN zbernicu o rýchlosti 500 kbit/s

2.1.3 CAN v nákladnom automobile

V ďalšom kroku sme pripojili CAN čítačku do OBD nákladného vozidla, a nastavili správne výstupné napätie na výstupe regulátora LM2596 (Obr. 23). V nákladnom vozidle



Obr. 23 Nastavenie výstupného napätia pri pripojení do OBD nákladného vozidla

má zbernica rýchlosť 500kbit/s (podobne ako osobný automobil), zbernica nie je vyvedená priamo, ale z hlavnej riadiacej jednotky VMCU (Obr. 4). Táto jednotka je, ako sme zistili, zabezpečená proti neoficiálnemu (pirátskemu) pripojeniu a informácie prostredníctvom OBD komunikuje len na základe výzvy oficiálneho diagnostického programu Techtool, komunikáciu nebolo možné pomocou STM uskutočniť.

Ďalším pokusom bolo priame pripojenie sa na linku BB2. Načítanie údajov z linky prebehlo úspešne, ale zaznamenané údaje nemáme ako dekódovať nakoľko je kódovanie tejto linky súkromné, definované výrobcom vozidla.

Linka BB1 má verejné kódovanie, definované normou SAE J1939-71 a SAE J1939-73, preto sme pokračovali priamym pripojením STM CAN čítačky na BB1. Čítačku ale bolo potrebné prestaviť na rýchlosť zbernice BB1 (250 kbit/s) pomocou parametrov SJW, TS1, TS2, BRP (Obr. 22). Ani po viacerých pokusoch s náhodnými číslami (podobne ako pri prvom načítaní v osobnom automobile), sme správne nastavenia netrafili. Pre správne nastavenie bolo potrebné porozumieť jednotlivým parametrom mechanizmu časovania bitov v CAN zbernici a ich súvislosti s parametrami v ChibiStudio (Obr. 22). Pomocou informácií

v dokumentácií ChibiOs, špecifikáciách Bosh CAN 2.0, ISO 11898, SAE J1939, AN754 [29], sme sa postupne dopracovali ku vzorcu na výpočet správnych nastavení CAN čítačky.

$$\text{rýchlosť zbernice } \left(\frac{\text{bit}}{\text{s}}\right) = \frac{\text{takt cpu STM CAN čítačky (Hz)}}{2 * (1 + \text{BRP}) * (3 + \text{TS1} + \text{TS2})}$$

Dosadili sme potrebné hodnoty:

$$250\,000 = \frac{72\,000\,000}{2 * (1 + \text{BRP}) * (3 + 8 + 1)}$$

Úpravami a výpočtom získali parameter BRP = 11, pre rýchlosť zbernice 250kbit/s.

```
38 static const CANConfig cancfg = {
39   CAN_MCR_ABOM | CAN_MCR_AWUM | CAN_MCR_TXFP,
40   CAN_BTR_SJW(0) | CAN_BTR_TS2(1) |
41   CAN_BTR_TS1(8) | CAN_BTR_BRP(11)
--
```

Obr. 24 ChibiStudio parametre pre CAN zbernicu o rýchlosti 250 kbit/s

2.1.4 Dáta na BB1

Načítanie dát a analýza správ podľa SAE J1939-71. Pomocou CAN čítačky a Putty bol vytvorený záznam komunikácie na linke BB1, ktorý sme začali analyzovať. Zo špecifikácie SAE J1939-71 sme zobrali jednu známu správu s PGN 61444 (Obr. 10), ktorý obsahuje napr. otáčky motora (Obr. 11). Túto správu sme volili zámerne, pretože správnosť dekódovania otáčok motora, je možné jednoducho overiť otáčkomerom. Túto správu sme následne hľadali v zachytenej komunikácii a uspeli sme. Výpis z STM sme z decimálnej sústavy prestavili na hexadecimálnu, čo umožnilo jednoduchšie hľadanie PGN v rozšírenom identifikátore správy (EID). Zistili sme však, že údaje dlhšie ako jeden bajt, v dátovej časti správy, sú zapisované od najmenej dôležitého až po najdôležitejší bajt, resp. vo formáte Little - Endian. Pre správne dekódovanie údajov je potrebné tieto bajty zoradiť opačne a následne ich zreťaziť (transformovať do Big - Endian).

Jedna zo zaznamenaných správ:

SID: 400, IDE: 1, EID: CF00400, RTR: 0, DLC: 8 DATA: F1 88 88 3C 1B FF F3 FF

V časti EID vidíme PGN F004. Zo špecifikácie vieme, že otáčky motora sú zapísané na pozícií 4 a 5 (Position: 4-5, Data Length: 2 bytes, Engine Speed). A spôsob dekódovania (Resolution: 0.125 rpm/bit, 0 offset).

Dané hodnoty sú 3C a 1B, aplikovaním informácií o Endianite, dostávame hodnoty 1B a 3C v hexadecimálnom zápise. Tieto hodnoty zreťazíme - 1B3C a transformujeme do decimálnej podoby - 6972. Výslednú hodnotu vypočítame podľa informácií jednotlivého parametra, v tomto prípade vynásobíme hodnotou rozlíšenia (resolution) a pridáme kompenzáciu (offset). Výsledok je $6972 * 0,125 + 0 = 871,5$ otáčky za minútu. Získané poznatky sme overili pomocou skriptu v Pythone, ktorý v reálnom čase načítaval údaje z CAN čítačky, spracovával správy s PGN F004 a výsledky zobrazoval na monitore prenosného počítača (Obr. 25).



Obr. 25 Test načítania otáčok motora

2.1.5 Správy s DTC kódom

V oficiálnych publikáciách [5, 6] je uvedené, že závady v podobe DTC kódov sú odosielané prostredníctvom dátovej linky BB2, ktorej kódovanie nám ale nie je známe. Podľa schémy TEA2+ (Obr. 3) je zrejme, že časť závad musí byť odosielaná aj po BB1, využitím normy SAE J1939-73 (napr.: riadiaca jednotka ACM je pripojená iba na BB1). V špecifikácii SAE J1939-73, sú uvedené PGN identifikátory DTC chýb. Na riadiacej jednotke ACM sme generovali chyby a zároveň zaznamenávali komunikáciu na linke BB1. Všetky aktívne DTC chyby majú identifikátor PGN 65226, v hexadecimálnom zápise

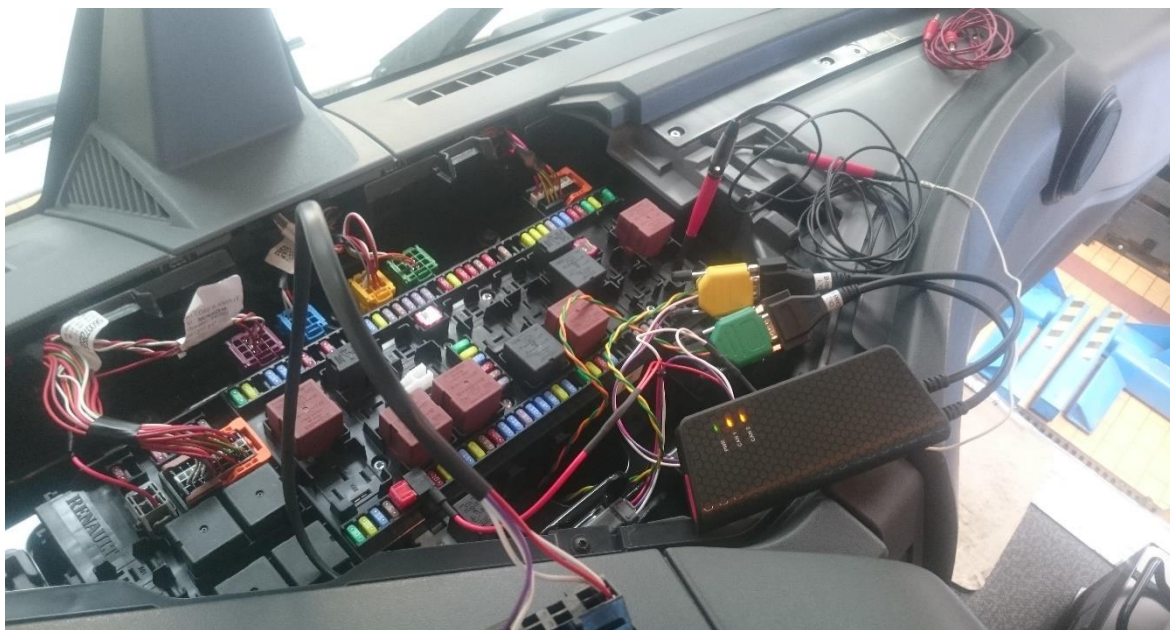
FECA. Správy obsahujúce PGN FECA obsahujúce chyby (Obr. 12) boli nájdené, ale v dátovej časti správy, mali bajty hodnotu 00 alebo FF a nedávali zmysel.

Napr.:

SID: 2EE, IDE: 1, EID: 18FECAEE, RTR: 0, DLC: 8 DATA: 0 FF FF FF FF FF FF FF.

To nás priviedlo k myšlienke, že chybové správy budú mať iný identifikátor, alebo riadiaca jednotka ACM nepoužíva PGN FECA. Nepomohlo ani generovanie chýb na ostatných jednotkách, ani hľadanie a dekodovanie všetkých správ obsahujúcich PGN identifikátor zo SAE J1939-73.

Rozhodli sme sa pre načítavanie oboch liniek zároveň, aby sme mohli čo najefektívnejšie prehliadať správy celej komunikácie. Pomocou zariadenia Kvaser Memorator sme spustili načítavanie na linkách BB1 a BB2 zároveň. Počas záznamu sme generovali na vozidle chyby.



Obr. 26 Záznam komunikácie na linkách BB1 a BB2 pomocou zariadenia Kvaser Memorator

Následne sme vytvorili skript v Pythone, ktorý v zázname vyfiltroval a zobrazil iba PGN uvedené v SAE J 1939-73. Ani v tomto zredukovanom zozname sme však v dátovej časti správ, nenašli dáta ktoré by dávali zmysel.

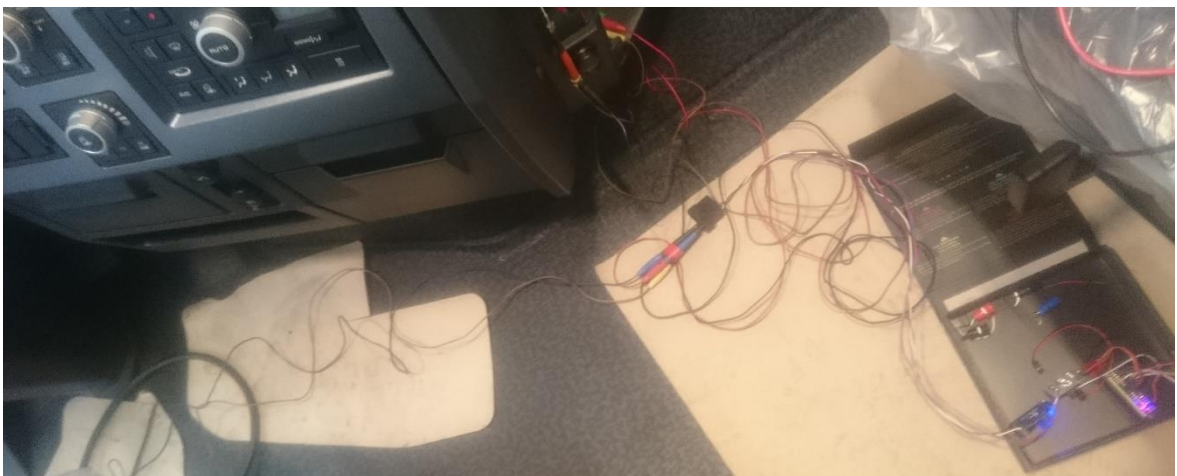
Nakoľko všetky posledné pokusy, pri ktorých sme sa snažili nájsť správy prostredníctvom PGN, ktorý mal obsahovať DTC chyby skončili nezdarom a vzhľadom na fakt, že Techtool sa musí nejakým spôsobom dostať k chybám, nás priviedol k myšlienke zaznamenať komunikáciu medzi Techtoolom a VMCU. Techtool s VMCU komunikuje prostredníctvom diagnostickej riadiacej jednotky VOCOM, ktorá sa pripája do OBD.

VOCOM komunikuje s VMCU po 500kbit/s CAN zbernici (Obr. 3) na ktorú sme sa paralelne napojili.

Na vozidle sme na linkách BB1 a BB2 spustili zaznamenávanie prostredníctvom Kvaseru a vygenerovali skupinu chýb. Následne sme pripojili STM CAN čítačku medzi VOCOM a OBD(VMCU) (Obr. 27, 28).



Obr. 27 Záznam komunikácie medzi OBD a VOCOM



Obr. 28 Záznam komunikácie medzi OBD a VOCOM

Po pripojení sme spustili záznam komunikácie na STM CAN čítačke a chyby načítali štandardným postupom prostredníctvom Techtoolu (Obr. 14). Záznam komunikácie VMCU-Techtool bol omnoho kratší, nakoľko Techtool s VMCU komunikuje len potrebné údaje. V tomto zázname sme pomocou skriptu v Pythone hľadali chyby, ktoré sme vygenerovali a Techtool ich zobrazil. DTC chyby sme našli (Obr. 29).

Řídicí modul následného zpracování (ACM)	P203A13: Reductant Level Sensor "A", Přerušený obvod
Řídicí modul následného zpracování (ACM)	P204A13: Reductant Pressure Sensor, Přerušený obvod
Řídicí modul následného zpracování (ACM)	P204F93: SCR System Performance Dosing Failure, Bez operace

Obr. 29 Výpis z Techtoolu

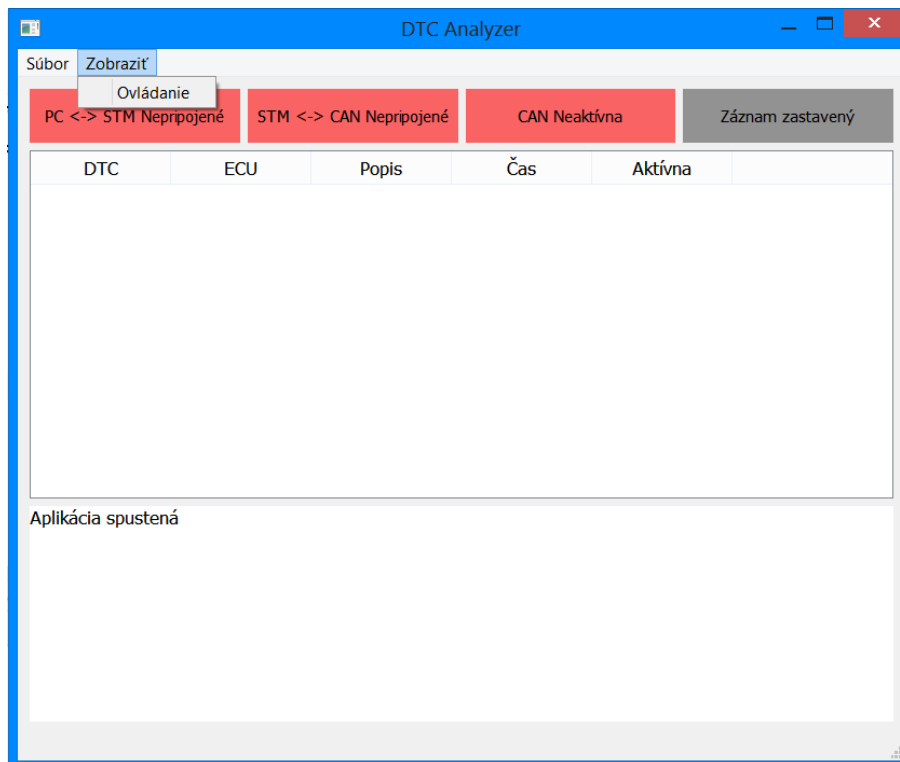
Výpis z STM CAN čítačky obsahujúci chybu P203A13 v tvare 20 3A 13:

SID: 1F2, IDE: 1, EID: 18DA11F2, RTR: 0, DLC: 8 DATA: 6 19 6 20 3A 13 FF 0

Spustili sme vyhľadávanie týchto chýb v zázname z Kvaseru, so zohľadnením možnosti endianity a uloženia na iných pozíciách v dátovej časti správy. Chyby sme našli, zápis na hlavných dátových linkách bol podobný, bajty hexadecimálneho zápisu boli rovnaké, ale mali vymenené poradie, zápis vo formáte Little - Endian a pozície bajtov v dátovej časti boli posunuté.

2.1.6 Aplikácia

Na základe získaných poznatkov, sme vytvorili overovací skript v Pythone. Skript načíta zoznam všetkých DTC kódov používaných v týchto nákladných vozidlách a v každej prijatej správe hľadá chybu a porovnáva so zoznamom všetkých DTC kódov a v prípade zhody ju vypíše. Po úspešnom overení poznatkov, sme začali s tvorbou finálnej aplikácie. Aplikácia má jednoduché GUI (grafické používateľské prostredie). Okno aplikácie je škálovateľné. Na vrchnej strane obsahuje 4 ukazovatele (označovače - label) s popisom, ktoré informujú v akom stave sa aplikácia nachádza. Label PC - STM informuje o pripojení STM CAN čítačky k počítaču. Label STM - CAN zobrazuje informáciu o správnom pripojení STM čítačky na dátovú linku. Label CAN Neaktívna zobrazuje aktivitu zbernice, zbernica môže prestať vysielat' v prípade závažnej chyby alebo hibernácie. Label Záznam zastavený zobrazuje stav načítavania. Pod ukazovateľmi sa nachádza tabuľka v ktorej sa načítavané chybové hlášky zobrazia. Riadky tabuľky sú vzostupne usporiadané, podľa času prijatia chyby. Jednotlivé riadky tabuľky obsahujú DTC kód závady, názov riadiacej

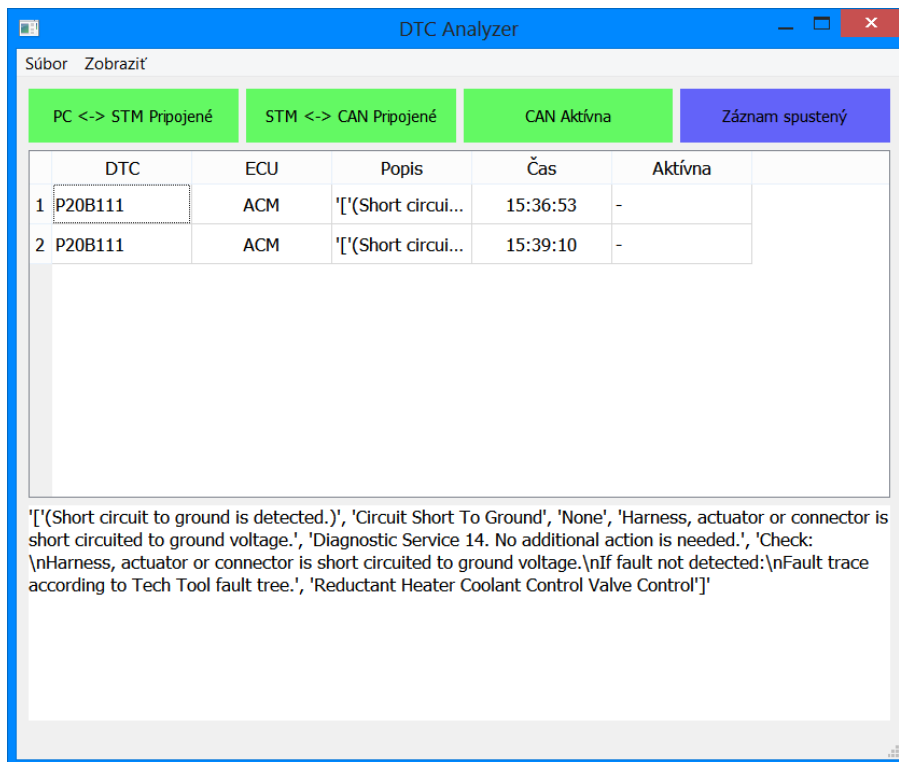


Obr. 30 Výsledná aplikácia

jednotky, ktorá chybu vypísala, popis chyby a čas výskytu chyby. Pod tabuľkou sú zobrazené dodatočné informácie o stave aplikácie alebo celkový popis chyby označeného riadku v tabuľke.

Celá aplikácia je po konzultácií so zadávateľom a mechanikmi pracujúcimi na diagnostike vozidiel primárne ovládaná klávesnicou, dodatočne myšou. Ovládanie behu aplikácie je možné, len v prípade kliknutia na tlačidlo zobraziť (Obr.40). Databáza všetkých používaných DTC kódov v daných vozidlách je načítaná po stlačení klávesy „L“. Klávesa „C“ spustí pripojenie CAN čítačky. CAN čítačka sa k počítaču pripája pomocou virtuálneho COM portu. Aplikácia načíta všetky dostupné COM porty, zistí na ktorom je pripojená čítačka a inicializuje pripojenie. Po úspešnom pripojení je pomocou klávesy „S“ možné spúšťať a zastavovať načítavanie aktuálne prebiehajúcej komunikácie na pripojenej dátovej linke. Chyby sa zobrazujú v tabuľke, výber jednotlivých riadkov na zobrazenie detailu popisu chyby je možný pomocou navigačných šípok na klávesnici alebo pomocou myši.

O načítavanie chýb sa stará samostatné vlákno (thread), aby nedošlo k zablokovaniu vykresľovacieho vlákna a aplikáciu bolo možné počas behu plynule ovládať.



Obr. 31 Výsledná aplikácia počas spusteného načítavania

3. Záver

Vzhľadom na aktuálnosť problematiky, sme v bakalárskej práci riešili viaceré problémy. V úvode práce sme analyzovali súčasný stav diagnostikovania chýb na dátových linkách nákladných vozidiel. V ďalších krokoch sme navrhli a implementovali sme riešenie aktuálnych nedostatkov pri odstraňovaní porúch v servisných podmienkach.

Výsledkom uvedenej analýzy, je doplnkové diagnostické riešenie, zložené zo záznamového zariadenia a počítačovej aplikácie.

Pracovníci servisno - školiaceho centra sú s výsledkom spokojní a sú zvedaví na výsledky, ktoré toto riešenie prinesie v reálnom nasadení pri servisných úkonoch a testovacích jazdách.

Počas práce sme objavili ďalšie možnosti vylepšovania riešenia.

S aktuálnou STM CAN čítačkou sme schopní aktuálne načítavať chyby len z jednej zbernice zároveň, pretože použitý mikro - počítač STM obsahuje v procesore len jeden CAN radič. Možné rozšírenie spočíva v pridaní ďalšieho MCP2551 prevodníka a pridaní identického mikro - počítača STMF103, prípadne výmenou aktuálneho STM za model s dvoma CAN radičmi v procesore.

V priebehu spracovávanía problematiky sme zistili, že v prípade poruchy, riadiace jednotky vytvoria zamrznutú správu („Freeze frame“), ktorá obsahuje hodnoty údajov v stave, v akom sa nachádzali v čase vzniku poruchy. O týchto správach nemáme aktuálne dostatok informácií, ktoré v budúcnosti bude možné doplniť a získať vďaka tomu o niečo komplexnejší pohľad na okolnosti vzniku chyby.

Ďalej je možné analyzovať BAM správy, ktoré zabezpečujú prenos dát dlhších ako 8 bajtov, pomocou rozdelenia týchto údajov do viacerých správ. Tieto správy sme objavili v zaznamenanej komunikácii a ich výskyt je relatívne sporadický.

Uvedené Freeze a BAM správy môžu viesť k ďalšiemu zlepšeniu diagnostiky nákladných vozidiel.

4. Použitá literatura

- [1] ISO 11898-1:2003, *Road Vehicles - Controller area network (CAN) - Part 1: Data link layer and physical signaling*
- [2] ISO 11898-2:2003, *Road vehicles - Controller area network (CAN) - Part 2: High-speed medium access unit*
- [3] BOSH CAN Specification 2.0:1991, *CAN Specification 2.0*
- [4] Eisenreich, Dan a Brian DeMuth. *Designing Embedded Internet Devices*. Newnes. 2002. ISBN 1-878707-98-1
- [5] Maurin, Luc. *Electronic Architecture P2683*. Volvo Group. 2013
- [6] Divies, Sylvian a Stéphane Virlogeux. *Architektura Elektrické Instalace TEA2+*. Volvo Group. 2013
- [7] Kindel, Olaf a Mario Friedrich. *Softwareentwicklung mit AUTOSAR (Software Development with AUTOSAR)*. Dpunkt verlag, 2009. ISBN 978-3-89864-563-8.
- [8] SAE J 1939:2007, *Recommended Practice for a Serial Control and Communications Vehicle Network*
- [9] SAE J 1939-15:2008, *Reduced Physical Layer, 250K bits/sec, UN-Shielded Twisted Pair (UTP)*
- [10] SAE J 1939-21:2006, *Data Link Layer*
- [11] SAE J 1939-31:2004, *Network Layer*
- [12] SAE J 1939-71:2003, *Vehicle Application Layer*
- [13] SAE J 1939-73:2006, *Application Layer - Diagnostics*
- [14] SAE J 2284-3:2002, *High-Speed CAN (HSC) for Vehicle Applications at 500 KBPS*
- [15] Python, [online] Dostupné na internete: <<https://www.python.org/>>
- [16] PyCharm, [online] Dostupné na internete: <<https://www.jetbrains.com/pycharm-edu/>>
- [17] PyQt5, [online] Dostupné na internete:
<<https://www.riverbankcomputing.com/software/pyqt/intro>>
- [18] PySerial, [online] Dostupné na internete: <<https://pythonhosted.org/pyserial/>>

- [19] Pickle, [online] Dostupné na internete: <<https://docs.python.org/3/library/pickle.html>>
- [20] OpenPyXL, [online] Dostupné na internete:
<<https://openpyxl.readthedocs.io/en/stable/>>
- [21] ChibiOS, [online] Dostupné na internete: <<http://chibios.org/dokuwiki/doku.php>>
- [22] ChibiStudio, [online] Dostupné na internete:
<http://chibios.org/dokuwiki/doku.php?id=chibios:product:chibistudio:start>
- [23] Kernighan, Brian W. a Dennis M. Ritchie. *Programovací jazyk C*. Vydanie 2. Benko, Vladimír a Angelika Forusová. Alfa, 1989. ISBN 8005001541, 9788005001543.
- [24] STM32F103x8:2015, *STM32F103x8 Datasheet - production data*
- [25] LM2596:2016, *LM2596 Simple Switcher Power Converter 150-kHz 3-A Step-Down Voltage Regulator*
- [26] MCP2551:2007, *High-Speed CAN Transceiver*
- [27] Putty, [online] Dostupné na internete: <<https://www.putty.org/>>
- [28] Fritzing, [online] Dostupné na internete: <<http://fritzing.org/home/>>
- [29] Microchip AN754:2001, *Understanding Microchip's CAN Module Bit Timing*
- Obr. 1, [online] Dostupné na internete:
<<https://www.volvotrucks.sk/content/dam/volvo/volvo-trucks/masters/euro-6/volvo-fh-series/volvo-fh-landing/2324x1200-volvo-fh-landing-truck-hero.jpg>>
- Obr. 2, [online] Dostupné na internete:
<https://www.renault-trucks.sk/media/image/range-t-high-gallery-1_showroom-large_809x539px.jpg>
- Obr. 3, 4, 6, 14, [servisno - školiace stredisko] *Obrázky zosnímané z materiálov poskytnutých servisno - školiacim strediskom*
- Obr. 8, [online] Dostupné na internete: <<http://www.embeddedc.in/p/can-basics3.html>>
- Obr. 9, [online] Dostupné na internete:
<<https://canlogger1000.csselectronics.com/img/SAE-J1939-PGN-SPN-Message-Structure-Identifier-CAN-Bus.jpg>>

Obr. 16, [online] Dostupné na internete:

<<https://www.kvaser.com/wp-content/uploads/canland/product/13710-MemPro2xhSv2-ORIGINAL.png>>

Obr. 10, 11, [SAE J1939-71] *Obrázky zo snímané zo špecifikácie SAE J1939-71*

Obr. 12, 13, [SAE J1939-73] *Obrázky zo snímané zo špecifikácie SAE J1939-73*

Obr. 5, 7, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 [Martin Mašek] *Obrázky vyhotovil autor tejto práce*

5. Prílohy

1. Zdrojový kód aplikácie v Pythone pre Windows
2. Aplikácia pre Windows v spustiteľnom exe súbore
3. Zdrojový kód pre STM mikro - počítač a build
4. Ovládač pre USB prevodník