



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Testování řídicí jednotky posilovače

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: 1802T007 – Informační technologie

Autor práce: **Bc. Tomáš Drozd**

Vedoucí práce: Ing. Tomáš Martinec, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

Powersteering control unit testing

Master thesis

Study programme: N2612 – Electrotechnology and informatics

Study branch: 1802T007 – Information technology

Author: **Bc. Tomáš Drozd**

Supervisor: Ing. Tomáš Martinec, Ph.D.



ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Tomáš Drozd**
Osobní číslo: **M15000162**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Informační technologie**
Název tématu: **Testování řídicí jednotky posilovače**
Zadávající katedra: **Ústav mechatroniky a technické informatiky**

Z á s a d y p r o v y p r a c o v á n í :

1. Seznamte se s funkcí řídicí jednotky posilovače řízení a se senzorem zkrutu G269.
2. Vytvořte aplikaci pro diagnostiku řídicí jednotky posilovače řízení, která bude umožňovat i základní nastavení posilovače mimo automobil.
3. Popište komunikaci senzoru G269 s řídicí jednotkou posilovače a simulujte funkci tohoto senzoru pomocí jednotky založené na počítači Raspberry Pi.
4. Otestujte vytvořený diagnostický program včetně jednotky pro simulaci senzoru G269.



Rozsah grafických prací: **dle potřeby dokumentace**

Rozsah pracovní zprávy: **40–50 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

- [1] **Drozd, T.: Analýza CAN komunikace EPS posilovače VW Golf V. Liberec, 2015. Bakalářská práce, Technická univerzita v Liberci, vedoucí Ing. Tomáš Martinec, Ph.D.**
- [2] **Robert Bosch GmbH. CAN Specification Version 2.0. Postfach 30 02 40, D-70442 Stuttgart, 1991.**
- [3] **VOLKSWAGEN AG. Data Exchange On The CAN Bus I. Wolfsburg, 2001.**

Vedoucí diplomové práce: **Ing. Tomáš Martinec, Ph.D.**

Ústav mechatroniky a technické informatiky

Konzultant diplomové práce: **Ing. Jaromír Tymeš**

TRW Automotive Czech s.r.o. Aftermarket Operations
Frýdlant

Datum zadání diplomové práce: **10. října 2017**

Termín odevzdání diplomové práce: **14. května 2018**

prof. Ing. Zdeněk Plíva, Ph.D.

děkan



doc. Ing. Milan Kolář, CSc.

vedoucí ústavu

V Liberci dne 10. října 2017

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 4. 5. 2018

Podpis:



Poděkování

Rád bych na tomto místě poděkoval všem, kteří mi umožnili se tímto tématem zabývat. Především děkuji firmě TRW Automotive Czech s.r.o. za zapůjčení posilovače řízení a osciloskopu. Autodílně Luděk Drozd děkuji za poskytnutí diagnostických nástrojů. Dále bych rád poděkoval panu Ing. Jaromíru Tymešovi za rady při vypracování praktické části práce a panu Ing. Tomáši Martincovi, Ph.D. za rady a pomoc při vypracování práce.

Abstrakt

Tato práce se zabývá řídicí jednotkou a senzorem zkrutu G269 elektromechanického posilovače ZF ve vozech koncernu VW. Popisuje funkci této řídicí jednotky, komunikaci s dalšími řídicími jednotkami po sběrnici CAN a komunikaci se senzorem zkrutu G269. Praktická část je věnována tvorbě diagnostické aplikace běžící na počítači Raspberry Pi a komunikující po sběrnici CAN. Dále je věnována simulaci funkce senzoru zkrutu G269. Funkčnost aplikace byla ověřena na posilovači řízení z vozu koncernu VW.

Klíčová slova:

Elektromechanický posilovač řízení, ZF, Automobilová diagnostika, Senzor G269, KWP2000

Abstract

This work deals with the electronic control unit and the torque sensor G269 of an electromechanical power steering ZF in cars of VW group. It describes function of this electronic control unit, communication with other electronic control units via CAN bus and communication with the torque sensor G269. The practical part is dedicated to the development of a diagnostic application running on computer Raspberry Pi and communicating via CAN bus. Next part is dedicated to simulation of function of the torque sensor G269. The functionality of the application is verified on a power steering from a VW group car.

Key words:

Electromechanical power steering, ZF, Automotive diagnostics, Sensor G269, KWP2000

Obsah

1	Elektromechanický posilovač řízení ZF	11
1.1	Úvod	11
1.2	Elektronika posilovače	12
2	Diagnostika řídicí jednotky	15
2.1	Diagnostický SW TEXA IDC4	15
2.2	Diagnostický SW ODIS	17
2.3	Rozdíly mezi generacemi ZF EPS	18
3	Simulování zpráv ostatních ECU	20
4	Monitorování zpráv ECU posilovače řízení	25
4.1	Zprávy s identifikátorem 3D0	25
4.2	Zprávy s identifikátorem 3D2	29
5	Chybové kódy	32
6	Základní nastavení posilovače řízení	35
7	Tvorba vlastního diagnostického programu	37
7.1	Grafické rozhraní programu	38
7.2	Programová logika	41
7.2.1	Zahájení komunikace a čtení identifikačních dat	42
7.2.2	Čtení a mazání chybových kódů	44
7.2.3	Čtení freeze-frame dat	45
7.2.4	Měřené hodnoty	46
7.3	Tvorba vlastní aplikace pro OS Android	48
8	Senzor G269	50
8.1	Měření na senzoru	51
8.2	Simulování funkce senzoru G269	54
9	Testování vytvořeného diagnostického programu	55
10	Závěr	57
	Literatura	59
	Obsah příloženého CD	61

Seznam obrázků

1.1	Stavba řízení	12
1.2	Schéma zapojení J500, V187 a G269	13
1.3	Konektor 1K0 973 205	14
1.4	Konektor 1K0 971 955	14
3.1	Raspberry Pi 3 a modul PiCAN2	21
3.2	Aplikace pro simulaci zpráv ostatních ECU	24
4.1	Hodnota bajtu E CAN ID 3D0 při startování motoru a rozjezdu	28
4.2	Hodnota bajtu A CAN ID 3D0 a dat senzoru G269 při otáčení volantem	28
4.3	Otáčení volantovou tyčí - zpráva 3D2	30
4.4	Otáčení volantovou tyčí - diagnostický SW VCDS	31
7.1	Rozšiřující modul PiCAN2	37
7.2	GUI po spuštění programu	39
7.3	GUI po načtení identifikace	40
7.4	GUI aplikace pro OS Android	49
8.1	Nákres senzoru G269	51
8.2	Senzor G269	51
8.3	Závislost úhlu zkrutu na výstupním napětí senzoru G269	52
8.4	Závislost bajtů A a B zprávy s CAN ID 3D2 na výstupním napětí senzoru G269	53
8.5	Wheatstoneův můstek	54

Seznam tabulek

2.1	Parametry měřené diagnostickým SW TEXA na testovacím voze	16
2.2	Stavy měřené diagnostickým SW TEXA na testovacím voze	16
2.3	Měřené hodnoty diagnostickým SW VCDS na testovacím voze	18
3.1	Záznam zpráv s CAN ID 280	21
3.2	Záznam zpráv s CAN ID 1A0	22
3.3	Záznam zpráv s CAN ID 0C2	22
4.1	Záznam zpráv s CAN ID 3D0 po zapnutí zapalování	26
4.2	Zprávy generované pro přístrojovou desku	27
4.3	Záznam zpráv s CAN ID 3D0 při startování motoru	27
4.4	Záznam zpráv s CAN ID 3D2	29
7.1	Záznam zpráv při vytvoření spojení s ECU posilovače řízení	43
7.2	Záznam zpráv při čtení chybových kódů	44
7.3	Záznam zpráv při mazání chybových kódů	45
7.4	Záznam zpráv při čtení freezeframe dat	46
7.5	Záznam zpráv při čtení měřených hodnot	47
7.6	Výpočet vybraných měřených hodnot	47
8.1	Měřené napětí na pinech senzoru G269 bez působení sil	52

Seznam zkratek

CAN	Controller Area Network, datové sběrnice místní sítě
ECU	Electronic Control Unit, elektronická řídicí jednotka
EPS	Electric Power Steering, elektrický posilovač řízení
EU	Evropská Unie
GUI	Graphical User Interface, grafické uživatelské rozhraní
HW	Hardware
ID	Identifikátor
JSON	JavaScript Object Notation, JavaScriptový objektový zápis
LSB	Least significant bit, nejméně významný bit
OBD	On Board Diagnostics, palubní diagnostika
OS	Operační Systém
PC	Personal Computer, osobní počítač
SSH	Secure Shell
SW	Software
UDP	User Datagram Protocol
VW	Volkswagen

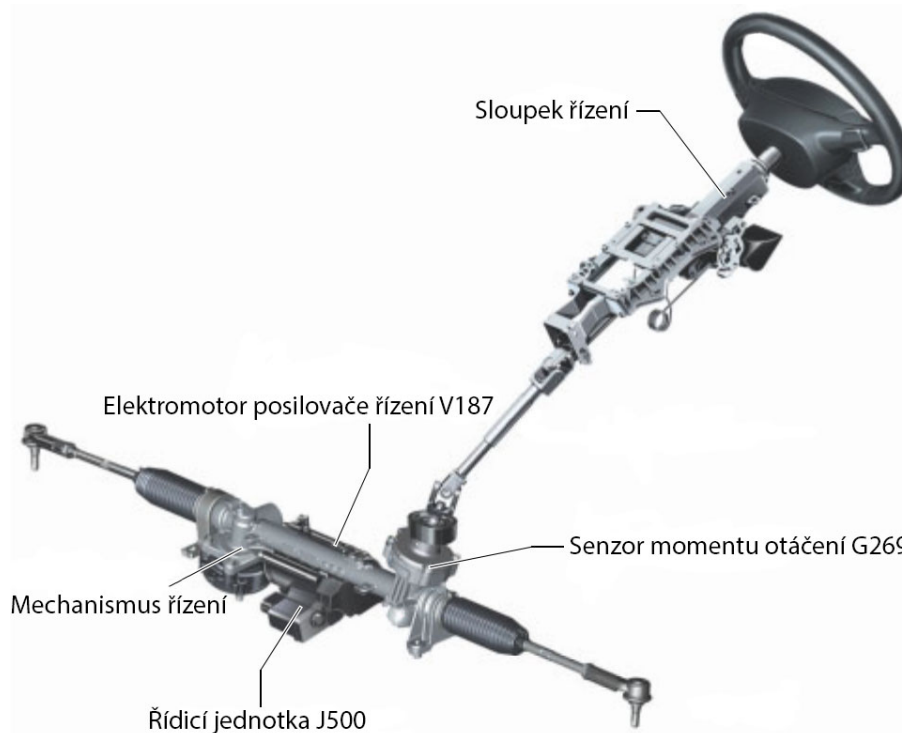
1 Elektromechanický posilovač řízení ZF

1.1 Úvod

Elektromechanické posilovače ZF jsou využívány koncernem Volkswagen v celé řadě vozů. Oproti hydraulickým posilovačům řízení dokáží elektromechanické posilovače snížit spotřebu automobilu o 0,6 litru paliva na 100 kilometrů. Díky tomu nabízí potencionální snížení emisí CO₂. Absence hydraulické pumpy a vedení hydraulické kapaliny umožňuje instalaci posilovače do menšího prostoru. Testování posilovačů řízení velmi usnadní diagnostika řídicí jednotky. Usnadňuje identifikaci závady, umožňuje sledování různých parametrů v reálném čase, nebo mazání chybových kódů. Řídicí jednotka posilovače, kterým se tato práce zabývá, komunikuje po sběrnici CAN. V této práci je popsána funkce řídicí jednotky elektromechanického posilovače řízení vyráběného firmou ZF. Dále je popsán senzor zkrutu G269, který je součástí tohoto posilovače řízení. Je zde vysvětlena jeho funkce a způsob komunikace s řídicí jednotkou.

Pro tuto práci byl použit posilovač ZF, který je označován jako generace I. Tento posilovač byl montován například do vozů *VW Golf Mk5*, *Škoda Octavia Mk2* nebo *Audi A3 Mk2*. Všechny tyto vozy jsou postavené na platformě PQ35. K ověření měřených hodnot byla použita data z komunikace mezi diagnostickým nástrojem a testovacím vozem *VW Golf Mk5*. Testovací vůz vyrobený v roce 2004 byl vybaven přeplňovaným diesellovým motorem o objemu 1,9 litru, s výkonem 66 kW a kódem motoru BRU. Ve voze se nacházel elektromechanický posilovač řízení ZF generace II.

Stavba celého řízení je vidět na obrázku 1.1. Posilovač využívá dvoupastorko-

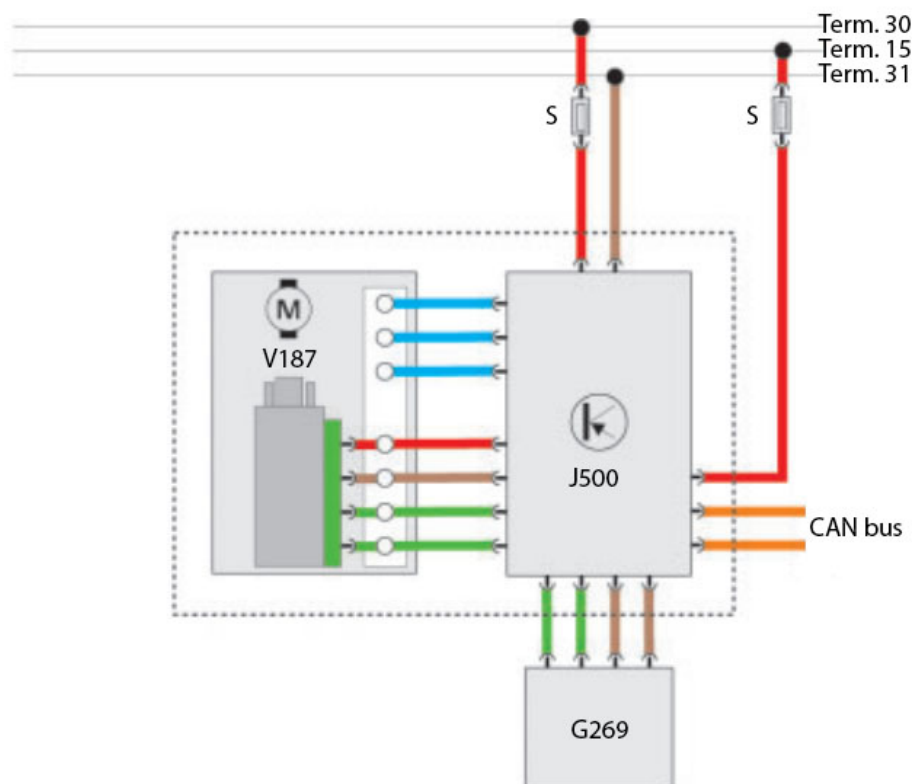


Obrázek 1.1: Stavba řízení.

vou konstrukci. Do hřebenové tyče je vsazen pastorek od volantové tyče a zároveň na druhé straně pastorek elektromotoru V187. Tato konstrukce umožňuje ovládání automobilu i při defektu posilovače řízení. Součástí pastorku, napojeného na volantovou tyč, je torsní tyč, kde se nachází senzor zkrutu G269. Řídicí jednotka má označení J500 a je součástí elektromotoru. Řidič otáčením volantu pohybuje s hřebenovou tyčí. S pohybem tyče mu zároveň pomáhá elektromotor. Pohyb hřebenové tyče v pouzdru, které je pevně spojeno s konstrukcí automobilu, způsobuje natáčení kol. [4]

1.2 Elektronika posilovače

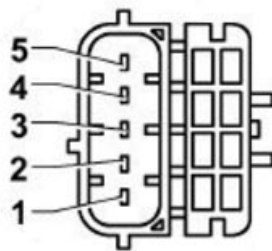
Řídicí jednotka posilovače řízení je označována jako J500. Jsou k ní přivedeny dva silové vodiče, napětí z terminálu 30 (stále + baterie) a zemnicí vodič z terminálu 31 (stále - baterie). Silové vodiče jsou na jednotku napojeny pomocí dvoupinového konektoru (obrázek 1.4), který má číslo dílu **1K0 971 955**. Zapojení pinů je



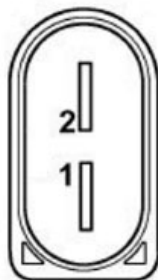
Obrázek 1.2: Schéma zapojení J500, V187 a G269

následující: **Pin 1 – Terminál 31**, **Pin 2 – Terminál 30**. Z terminálu 15 je přive-
deno napětí po zapnutí zapalování. Kladné vodiče jsou jistěny pojistkami. Schéma
zapojení je vidět na obrázku 1.2. Jednotka J500 komunikuje s ostatními jednot-
kami vozu pomocí sběrnice CAN. Vodiče CAN high a CAN low jsou na obrázku
znázorněny oranžovou barvou. [4]

Elektromotor má označení V187. Jednotka J500 poskytuje elektromotoru napá-
jecí a zemní vodič. Elektromotor je řízen jednotkou J500 pomocí tří vodičů, které
jsou na obrázku značeny modrou barvou. Jednotka pak získává z elektromotoru
informace pomocí dvou vodičů, které jsou vyobrazeny zelenou barvou. Jelikož je
jednotka J500 součástí elektromotoru, nejsou tyto vodiče z venku přístupné. Senzor
zkrutu G269 komunikuje s řídicí jednotkou pomocí čtyř vodičů. Propojení senzoru
a řídicí jednotky je realizováno konektorem s pěti piny (obrázek 1.3). Konektor má
číslo dílu **1K0 973 205**. Pin 1 není obsazen. Stejný konektor je použit pro spojení
řídicí jednotky posilovače s ostatními jednotkami ve voze. Zapojení pinů je následu-



Obrázek 1.3: Konektor 1K0 973 205



Obrázek 1.4: Konektor 1K0 971 955

jení: **Pin 1** – CAN low, **Pin 2** – CAN high, **Pin 4** – Terminál 15. Piny 3 a 5 nejsou využívány.

2 Diagnostika řídicí jednotky

K zjištění informací o řídicí jednotce bylo použito několik diagnostických programů. Komunikace byla nejdříve testována na testovacím voze, poté se stejným nastavením na vymontovaném posilovači. Spojení diagnostického nástroje s vozem je realizováno přes OBDII konektor. Jednotka posilovače J500 je napojena na sběrnici hnacího ústrojí. Tuto sběrnici a diagnostický konektor OBDII propojuje ECU *gateway* s označením J533. *Gateway* má na starost předávání zpráv mezi sběrnicemi s rozdílnou rychlostí komunikace a také propojení těchto sběrnic s diagnostickým konektorem.

2.1 Diagnostický SW TEXA IDC4

Jako první byla použita PC verze softwaru TEXA IDC4. Jedná se o univerzální diagnostický SW určený k použití pro většinu běžných automobilů. Proto je v prostředí programu nutné nejdříve zvolit testované vozidlo a poté jednotku posilovače řízení. Operace, které je možné s jednotkou provádět, jsou rozděleny do několika částí. V části *parametry* můžeme sledovat hodnoty fyzikálních veličin, s kterými jednotka pracuje. Názvy a hodnoty parametrů naměřených v testovaném vozidle při vypnutém motoru jsou uvedeny v tabulce 2.1. *Chybový kód* slouží ke čtení dat z paměti závad ECU. Je zde možné paměť závad i vymazat. Ke každému chybovému kódu bývá krátký popis chyby, informace o chybových kódech lze však dohledat na různých serverech. *Stavy* jsou zobrazeny pomocí stejného rozhraní jako *parametry*, ale nejedná se o fyzikální veličiny. Naměřené hodnoty jsou vypsány v tabulce 2.2. *Stavy* na rozdíl od *parametrů* většinou nabývají pouze několika hodnot. *Identifika-*

ce zobrazuje informace o ECU. Pomocí *aktivace* jsou spouštěny v jednotce předem naprogramované procedury. V *nastavení* lze měnit parametry ECU. Pro měření byla použita česká verze SW TEXA (některé názvy parametrů a stavů jsou v SW nesrozumitelně přeloženy do češtiny).

Tabulka 2.1: Parametry měřené diagnostickým SW TEXA na testovacím voze

Parametr	Hodnota
Vnitřní teplota řídicí jednotky posilovače řízení	11.0 °C
Krout. moment posil. řízení	0.000 Nm
Číslo rotoru	0 ot./min
Napětí baterie	12.300 V
Rychlost motoru	0 ot./min
Zobrazovaná jízdní rychlost	0 km/h
Podporový moment	0.000 Nm
Omezený podporový moment	0.000 Nm
Kroutící moment motoru	0.000 Nm
Momentový klíč	0.0 Nm
Přímá jízda	5.95 °
Úhel nastavení volantu	6.52 °
Snímače úhlu řízení	-8.93 °
Rychlost úhlu natočení kola	0.0 °/s

Tabulka 2.2: Stavby měřené diagnostickým SW TEXA na testovacím voze

Stav	Hodnota
Terminál 15	12.200
Stav motoru	Motor vypnutý
Stav systému	4
Řidič	ZAP
Snímače úhlu řízení	KOMPLETNÍ
Stav snímače úhlu řízení při startu	Inicializace úhlu řízení
CAN zprávy z ECU ABS	ABS; OK
CAN zpráva ECU motoru	-
CAN zpráva parkovacího asistenta	-
CAN zpráva brány (Gateway)	-

Identifikace jednotky EPS vozidla byla následující: Kód dílu - **1K1909144G**, Řídicí jednotka - **EPS_ZFLS Kl.2**, Kód ECU - **0**. Identifikace ECU vymontovaného posilovače řízení byla následující: Kód dílu - **1K1909143L**, Řídicí jednotka -

EPS_ZFLS K1.6, Kód ECU - **8390656**.

2.2 Diagnostický SW ODIS

Jako další byl použit SW ODIS ve verzi 2.2.6. Tento SW je určen pro diagnostiku vozů koncernu VW. Nejprve byla získána data z testovacího vozu, poté z přímého napojení na ECU vymontovaného posilovače řízení. Pro připojení je nutné nejprve zadat VIN vozidla a zvolit motorizaci. Poté jsou načtena identifikační data všech jednotek vozidla. Každá ECU má svou adresu, posilovač řízení se nachází pod adresou 44. Funkcí *naměřené hodnoty* lze sledovat některá data (tabulka 2.3) stejná jako u SW TEXA v *parametrech* a *stavech*. Pod položkou *paměť událostí* je možné číst chybové kódy uložené v ECU a paměť závad smazat, ale nelze smazat jednotlivé chybové kódy zvlášť. Ke každé chybě je zobrazen krátký popis chyby a také naměřená data v okamžiku zápisu chyby do paměti, jako např. napětí, teplota a další. V *přizpůsobení* lze měnit různé parametry ECU.

Komunikace s ECU posilovače v testovaném voze probíhala za použití protokolu KWP2000 a Transport Protocolu 2.0. Identifikace jednotky EPS vozidla byla následující: Číslo dílu VW - **1K1909144G**, Označení systému - **EPS_ZFLS K1.2**, Verze softwaru - **1401**. Komunikace s vymontovaným posilovačem řízení probíhala za pomoci stejného protokolu. Identifikace ECU byla následující: Číslo dílu VW - **1K1909143L**, Označení systému - **EPS_ZFLS K1.6**, Kódování - **2048**, Verze softwaru - **0402**.

Tabulka 2.3: Měřené hodnoty diagnostickým SW VCDS na testovacím voze

Název	Měřená hodnota
Teplota řídicí jednotky -J500	11.0 °C
Moment motoru, snímač momentu řízení -G269	0.000 Nm
Rychlost rotoru -G28	0 /min
Napětí relé	12.30 V
Napětí akumulátoru	12.10 V
Referenční napětí	5.1 V
Počet otáček motoru vozidla	0 /min
Rychlost vozidla	0.0 km/h
Stav motoru	motor VYP
Stav systému	4
Stav svorky 15	svorka 15 ZAP
Stav relé	rele ZAP
Podpurný moment	0.000 Nm
Omezený podpurný moment	0.000 Nm
Moment motoru, snímač momentu řízení -G269	0.000 Nm
Moment torzní tyče	0.0 Nm
Teplota řídicí jednotky -J500	11.0 °C
Stav vytlačovacího přípravku	zapnuto
Koncový doraz	5.95 °
Přímý úhel	6.52 °
Snímač úhlu natočení volantu -G85	-26.95 °
Rychlost úhlu řízení	0.0 °/s
Zobrazení provedeno základní nastavení -G85	v pořádku
Vnitřní stav snímače úhlu natočení volantu -G85	Úh.ř. inic.
Odezva CAN řídicí jednotky ABS -J104	ABS 1
Odezva CAN řídicí jednotky motoru	Motor 1
Odezva CAN řídicí jednotky palubní sítě	Elektr.CR1
Odezva CAN sdružených přístrojů	
Odezva CAN Gateway	Gateway 1
Odezva CAN snímače úhlu natočení volantu -G85	Úh.vola.1

2.3 Rozdíly mezi generacemi ZF EPS

Elektromechanické posilovače první generace firmy ZF mají označení řídicí jednotky **1K1909143**, druhá generace nese označení **1K1909144** a třetí pak označení **1K0909144**. Označení ECU téže generace se pak liší koncovým písmenem. Každá ECU má aktivní nějakou charakteristickou křivku chování. Generace I a II (starší datum výroby) v sobě mohou mít uloženo až 16 těchto křivek a umožňují přepí-

nání charakteristické křivky pomocí funkce přizpůsobení v kanálu 001. U generace III tento kanál není dostupný. Právě aktivní křivku lze vyčíst z identifikace ECU, je značena zkratkou Kl. a číslem křivky. Pokud je křivka značena číslem vyšším než 20, zpravidla neumožňuje přepnutí na jinou křivku. EPS v testovacím voze má řídicí jednotku s identifikací **1K1909144G EPS_ZFLS Kl.2**, je tedy druhé generace s aktivovanou charakteristickou křivkou číslo 2. Vymontovaný zkušební kus EPS má ECU první generace s identifikací **1K1909143L EPS_ZFLS Kl.6**, aktivní charakteristická křivka je tedy číslo 6. Pomocí diagnostického SW VCDS ve verzi 805.0 lze ve skupině 080 měřených hodnot zjistit datum výroby ECU. Posilovač testovacího vozu generace II byl vyroben 21.11.2004, vymontovaný testovací posilovač generace I byl vyroben 17.07.2003.

Diagnostický SW ODIS zobrazuje více měřených hodnot. Informace o odezvě gateway nebo ECU motoru v diagnostickém SW TEXA chybí. Srovnáním měřených hodnot z ECU posilovače ve vozidle a mimo vozidlo lze zjistit, které hodnoty má jednotka k dispozici z vlastních napřímo připojených senzorů a které získává po sběrnici CAN z dalších jednotek. U vymontovaného posilovače byly v měřených hodnotách zobrazovány otáčky motoru 1184 ot./min. a rychlost vozidla 50 km/h. V případě selhání senzoru otáček motoru a rychlosti jsou nejspíše použity tyto náhradní hodnoty, které zajistí alespoň nějaký posilovací účinek. Protože je hodnota otáček motoru vyšší než 800 ot./min., zobrazuje diagnostický SW stav motoru jako zapnutý. Hodnota stavu systému je u EPS ve vozidle 4, u EPS mimo vozidlo je to hodnota 3. Hodnota koncového dorazu a přímého úhlu se také liší, ale lze předpokládat, že se jedná o vnitřní hodnoty ECU. Všechny hodnoty týkající se senzoru úhlu natočení volantu G85 jsou u posilovače mimo automobil neměřitelné. Senzor G85 tedy s ECU posilovače komunikuje po sběrnici CAN. Informace o odezvě dalších řídicích jednotek po sběrnici CAN jsou u posilovače mimo vozidlo nulové. Odezva sdružených přístrojů byla u posilovače ve vozidle neměřitelná. Tento fakt byl nejspíše způsobený chybovým kódem uloženým v ECU sdružených přístrojů, který popisuje ECU přístrojů jako defektní. Všechny ostatní hodnoty má jednotka posilovače k dispozici z vlastních senzorů.

3 Simulování zpráv ostatních ECU

Posilovač řízení ke své funkci potřebuje získávat zprávy od dalších řídicích jednotek vozidla po sběrnici CAN. K identifikaci těchto zpráv bylo nutné monitorovat sběrnici CAN ve vozidle během provozu. K monitorování sběrnice bylo použito **Raspberry Pi 3** s rozšiřujícím modulem **PiCAN2** firmy SK Pank electronics, který je na obrázku 3.1. Rozšiřující modul používá CAN kontrolér **MCP2415** a CAN transceiver **MCP2551**. Zařízení používá **SocketCAN** ovladač a programování je možné v jazyce C nebo Python. Pomocí ethernetu byl k Raspberry Pi 3 připojen notebook a komunikace probíhala za použití SSH. K monitorování komunikace byly použity SocketCAN nástroje *can-utils*. Nástroj *candump* se základním nastavením slouží k výpisu veškerých zpráv na sběrnici, včetně jejich CAN ID (identifikátorů) a timestamp (časových značek) s přesností jedné mikrosekundy. Pro možnost monitorování a odesílání dat přímo z notebooku byl použit nástroj *cannelloni*, který pomocí UDP přenáší CAN zprávy mezi zařízeními v síti. Na notebooku byla vytvořena virtuální sběrnice CAN, na kterou byl přesměrován provoz reálné sběrnice.

Zpráva s CAN ID 280 byla identifikována jako zpráva řídicí jednotky motoru s informací o otáčkách motoru. Ukázkové zprávy jsou uvedeny v tabulce 3.1. Interval mezi jednotlivými zprávami je 20 ms. Informace o otáčkách je zakódována pomocí bajtů C a D. Sledováním měřených hodnot pomocí diagnostického SW a hodnot těchto dvou bajtů byl stanoven vzorec pro výpočet otáček $(D*256+C)/4$. Hexadecimální hodnoty je nejprve nutné převést na decimální. Výpočet pro první zprávu v tabulce 3.1 vypadá takto: $(12*256+236)/4 = 827$ ot./min. Hodnoty ostatních bajtů nemají pro ECU posilovače žádný význam. [2]

Zpráva s CAN ID 1A0 je odesílána řídicí jednotkou ABS a nese v sobě informaci



Obrázek 3.1: Raspberry Pi 3 a modul PiCAN2

Tabulka 3.1: Záznam zpráv s CAN ID 280

Čas [s]	ID	A	B	C	D	E	F	G	H	Otáčky [ot./min.]
0,000000	280	09	17	EC	0C	17	00	1E	17	827
0,020001	280	09	19	C8	0C	19	00	1E	19	818
0,040000	280	09	19	C4	0C	19	00	1E	19	817
0,060004	280	09	17	F8	0C	17	00	1F	17	830

o rychlosti vozidla. Rychlost je zakódována pomocí bajtů C a D. Bajt H je inkrementován od hodnoty 0x10 do 0x1F po krocích o velikosti 0x01, tím je zajištěna kontrola nad chybějícími zprávami. Ukázkové zprávy jsou uvedeny v tabulce 3.2. Při vypnutém motoru je zpráva odesílána každých 10 ms. Při nastartovaném motoru je vždy deset zpráv odesláno v rozestupech 0,5 ms, poté následuje pauza o délce 100 ms. Vzorec byl opět stanoven dle měřených hodnot pomocí diagnostického SW. Pro výpočet rychlosti v km/h je nutné hodnotu $(D*256+C)$ posunout o jeden bit vpravo a vynásobit číslem 0,01. Výpočet pro čtvrtou zprávu z tabulky 3.2 vypadá takto: $0*256+164 = 164$; $164 \text{ } 10 = 10100100 \text{ } 2$; $1010010 \text{ } 2 = 82 \text{ } 10$; $82*0,01 = 0,82$ km/h. [2]

Senzor úhlu natočení volantu G85 odesílá řídicí jednotce posilovače zprávu s CAN

Tabulka 3.2: Záznam zpráv s CAN ID 1A0

Čas [s]	ID	A	B	C	D	E	F	G	H	Rychlost [km/h]
0,000000	1A0	00	40	00	00	FE	FE	00	18	0
0,000072	1A0	00	40	00	00	FE	FE	00	19	0
0,000129	1A0	00	40	50	00	FE	FE	00	1A	0,4
0,000200	1A0	00	40	A4	00	FE	FE	00	1B	0,82

ID 0C2. Pokud bylo volantem otáčeno malou rychlostí, měnila se pouze data bajtů A a B. Při vyšší rychlosti otáčení se měnila i data bajtů C a D. Z toho lze vyvodit závěr, že bajty A a B nesou informaci o úhlu natočení volantu a bajty C a D o rychlosti změny tohoto úhlu. Zpráva s CAN ID 0C2 je odesílána každých 10 ms. Ukázkové zprávy jsou uvedeny v tabulce 3.3. Bajt E má vždy hodnotu 0x80 a bajt G hodnotu 0x00. Bajt F je inkrementován od 0x00 do 0xF0 po krocích velikosti 0x10. Bajt H je doplněk bajtů A až D a F na hodnotu 0xFF. Po převedení hodnot šestnáctkové soustavy do desítkové soustavy platí pro výpočet úhlu natočení volantu vzorec: $(256 \cdot B + A) \cdot 0,04375$. Výpočet pro čtvrtou zprávu z tabulky 3.3: $(256 \cdot 0 + 32) \cdot 0,04375 = 1,4^\circ$. Pro výpočet rychlosti změny úhlu platí stejný vzorec: $(256 \cdot D + C) \cdot 0,04375$. Pro opačný směr otáčení začíná hodnota bajtů B a D na 0x80. Při výpočtu veličin pro opačný směr otáčení je nutné nejprve hodnotu 0x80 odečíst. [2]

Tabulka 3.3: Záznam zpráv s CAN ID 0C2

Čas [s]	ID	A	B	C	D	E	F	G	H	Úhel [°]
0,000000	0C2	00	00	00	00	80	30	00	CF	0
0,009981	0C2	00	00	00	00	80	40	00	BF	0
0,019977	0C2	20	00	00	00	80	50	00	8F	1,4
0,030013	0C2	20	00	00	00	80	60	00	7F	1,4

Jednotka posilovače sleduje, zda se na sběrnici vyskytuje zpráva od řídicí jednotky gateway. Tato zpráva má CAN ID 390. Obsah zprávy nemá vliv na funkci posilovače řízení, proto byla generována stejná zpráva, která byla zachycena v testovacím vozidle.

Pro simulování zpráv byl použit HW popsáný na začátku této kapitoly. Byla

vytvořena sběrnice CAN, na kterou byla napojena řídicí jednotka posilovače J500, Raspberry Pi 3 s modulem PiCAN2 a diagnostický SW VCDS, který sloužil pro kontrolu chybových kódů a naměřených hodnot. Pro simulaci zpráv byla vytvořena aplikace v programovacím jazyce Python. Každá zpráva byla cyklicky odesílána ve vlastním vlákně ve stejných časových intervalech, které byly naměřeny. Sběrnice byla ve zdrojovém kódu vytvořena následujícím způsobem:

```
bus = can.interface.Bus(channel='vcan0', bustype='socketcan_native')
```

CAN zpráva pak může být vytvořena pomocí tohoto kódu:

```
ID_280 = can.Message(arbitration_id=0x280, data=[0x09, 0x00, 0x00, 0x00, 0x17, 0x00, 0x1E, 0x17], extended_id=False)
```

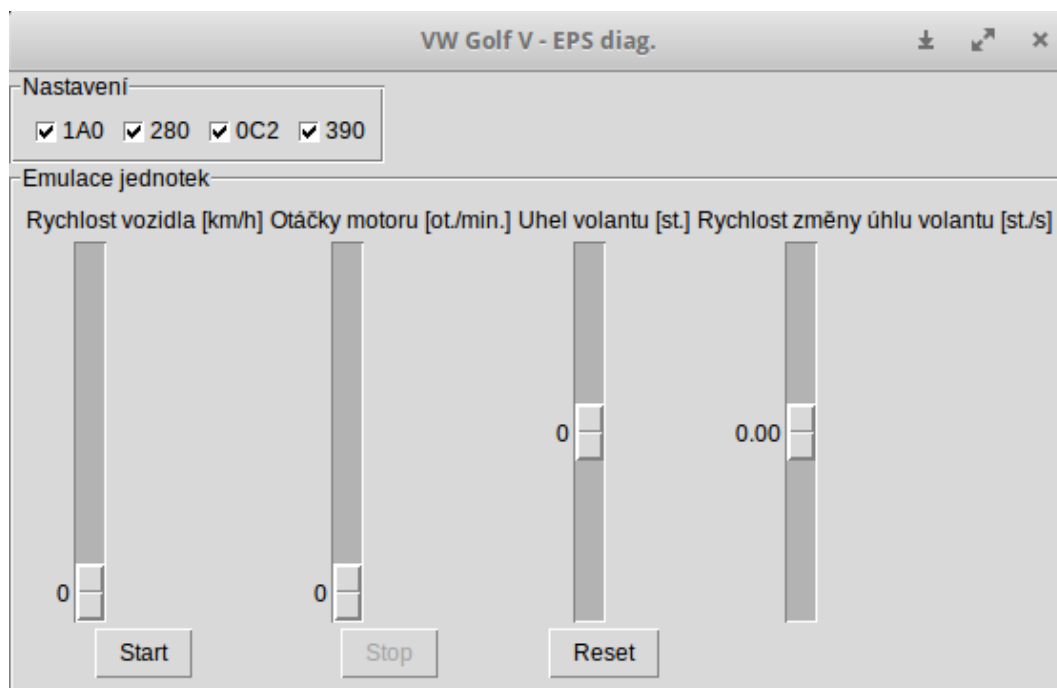
Vláknem pro cyklické odesílání zprávy bylo vytvořeno pomocí balíku multiprocessing následujícím způsobem:

```
process_280 = mp.Process(target=message_280, args=(rpm, flag_end))
process_280.start()
```

Vláknem jsou předávány dvě proměnné - hodnota otáček pocházející z grafického prostředí aplikace a pomocná proměnná sloužící k ukončení vlákna. Proměnné jsou sdílené mezi hlavním vláknem aplikace a jednotlivými vlákny pro odesílání zpráv a byly vytvořeny za použití *Value* z balíku *multiprocessing*. Obsah zprávy je tímto způsobem možné měnit za běhu vlákna. Kód vlákna pak vypadá následovně:

```
while flag.value == 1:
    ID_280.data[3], ID_280.data[2] = divmod(val.value * 4, 0x100)
    bus.send(ID_280)
    time.sleep(0.001)
```

Na ukázkou je uveden kód pro simulaci otáček motoru. Vzorec pro výpočet otáček byl stanoven takto: $(D \cdot 256 + C) / 4$. Hodnota z posuvníku je podle tohoto vzorce zpětně přepočtena a rozdělena do dvou bajtů zprávy. Pokud ostatní bajty nemají vliv na funkci posilovače řízení, zůstávají nezměněny. Zpráva je odeslána na sběrnici a vláknem pak čeká po časový interval, který byl pro danou zprávu určen. Tento kód je vykonáván ve smyčce, dokud hodnota pomocné proměnné pro ukončení vlákna není změněna.



Obrázek 3.2: Aplikace pro simulaci zpráv ostatních ECU

Grafické rozhraní je uvedeno na obrázku 3.2. Bylo vytvořeno za použití modulu Tkinter. Umožňuje výběr CAN ID, které mají být odesílány na sběrnici. Pomocí posuvníků se nastavuje rychlost vozidla, otáčky motoru, úhel natočení volantu a rychlost změny úhlu natočení volantu. Úhel a rychlost změny úhlu natočení volantu lze zvolit z kladných i záporných hodnot. Tlačítkem Reset jsou všechny hodnoty nastaveny na nulu. Tlačítka Start a Stop vytvářejí resp. ukončují vlákna se zvolenými CAN ID.

Pomocí diagnostického SW VCDS bylo ověřeno, zda ECU posilovače řízení zprávy přijímá a zda simulované hodnoty odpovídají hodnotám, se kterými jednotka pracuje. Ověření proběhlo za použití funkce měřené hodnoty. Při simulování zpráv ostatních řídicích jednotek bylo možné z paměti závad odstranit některé chyby, které jsou blíže rozebírány v kapitole 5.

4 Monitorování zpráv ECU posilovače řízení

K zjištění zpráv, které odesílá řídicí jednotka posilovače ostatním jednotkám v automobilu, bylo nutné monitorovat sběrnici CAN testovaného posilovače řízení. Posilovač byl napojen na autobaterii pomocí odpovídajících pinů konektoru **1K0 971 955**. Piny 1 a 2 konektoru **1K0 973 205** byly propojeny s odpovídajícími svorkami modulu PiCAN2, tím byla vytvořena sběrnice CAN. Na sběrnici se nacházela pouze ECU posilovače řízení a Raspberry Pi 3 s modulem PiCAN2 pro monitorování. Díky tomu bylo zaručeno, že veškeré zprávy na sběrnici pochází od ECU posilovače řízení.

Sběrnice byla monitorována pomocí nástroje *candump*, který je součástí *can-utils* ovladačů SocketCAN. Příkaz *candump can0* zobrazí v terminálu veškeré zprávy na sběrnici can0. Na sběrnici se vyskytovaly zprávy s CAN ID **3D0** a **3D2**. Pomocí příkazu *candump can0,3D0:FFF -t* a je možné sledovat pouze zprávy konkrétního CAN ID spolu s přesným časem. Po zjištění CAN ID zpráv ECU posilovače řízení bylo možné zprávy se stejným CAN ID monitorovat i v testovacím vozidle.

4.1 Zprávy s identifikátorem 3D0

Data zprávy s CAN ID 3D0 se v testovacím vozidle měnila v krátkém časovém intervalu po zapnutí zapalování a dále pak při otáčení volantem. Pět zpráv bylo vždy odesláno s časovými rozestupy 0,2-0,3 ms. Mezi každou sadou pěti zpráv byl rozestup 100 ms. Záznam zpráv v tabulce 4.1 byl pořízen v testovacím vozidle a paměť závad ECU byla prázdná.

V tabulce 4.1 lze vidět, že krátce po zapnutí zapalování má bajt D po dobu devíti zpráv hodnotu 0x10, jinak má hodnotu 0x00. V této době jsou nejspíše testovány

Tabulka 4.1: Záznam zpráv s CAN ID 3D0 po zapnutí zapalování

Čas [s]	ID	A	B	C	D	E	F
0,000000	3D0	00	01	00	00	00	00
0,000192	3D0	00	01	00	00	00	00
0,000270	3D0	00	01	00	00	00	00
0,099946	3D0	00	01	00	10	00	00
0,100260	3D0	00	01	00	10	00	00
0,100464	3D0	00	01	00	10	00	00
0,100642	3D0	00	01	00	10	00	00
0,101012	3D0	00	01	00	10	00	00
0,199746	3D0	00	01	00	10	00	00
0,200115	3D0	00	01	00	10	00	00
0,200453	3D0	00	01	00	10	00	00
0,200793	3D0	00	01	00	10	00	00
0,200994	3D0	00	01	00	00	00	00
0,299080	3D0	00	01	00	00	00	00
...							
0,699798	3D0	02	01	00	00	00	00
...							
1,996165	3D0	02	00	00	00	00	00

ostatní komponenty potřebné pro funkci posilovače. Ihned po zapnutí zapalování má bajt B hodnotu 0x01, tato hodnota se přibližně po dvou vteřinách změní na hodnotu 0x00. Je to stejná doba, po kterou svítí na přístrojové desce oranžová kontrolka volantu. Díky zapůjčené přístrojové desce bylo možné simulací zprávy s CAN ID 3D0 tento jev ověřit. Na sběrnici byla připojena pouze přístrojová deska z automobilu *Škoda Octavia II* z roku 2008 a byly generovány zprávy z tabulky 4.2. V tabulce je uvedeno chování kontrolky volantu při různých hodnotách bajtu B. Pokud mají první dva LSB hodnotu 00, nesvítí kontrolka vůbec. Pokud mají hodnotu 10, svítí oranžová kontrolka. V ostatních případech na přístrojové desce svítí kontrolka červená.

Dále byla sběrnice monitorována při startu motoru (tabulka 4.3). Hodnota bajtu E byla během startování inkrementována od hodnoty 0x00 do 0xC8. Při běžícím motoru zůstala hodnota 0xC8 neměnná. Změna bajtu E z hodnoty 0x00 na 0xC8 trvala přibližně 0,54 sekundy. Stejná situace byla simulována na vymontovaném posilovači při simulaci startu motoru zvedajícími se otáčkami motoru. Naměřené

Tabulka 4.2: Zprávy generované pro přístrojovou desku

ID	A	B	C	D	E	F	Kontrolka volantu	B2
3D0	00	00	00	00	00	00	Žádná	0000 0000
3D0	00	01	00	00	00	00	Červená	0000 0000
3D0	00	02	00	00	00	00	Oranžová	0000 0000
3D0	00	03	00	00	00	00	Červená	0000 0000
3D0	00	04	00	00	00	00	Žádná	0000 0000
3D0	00	05	00	00	00	00	Červená	0000 0000
3D0	00	06	00	00	00	00	Oranžová	0000 0000
3D0	00	07	00	00	00	00	Červená	0000 0000

Tabulka 4.3: Záznam zpráv s CAN ID 3D0 při startování motoru

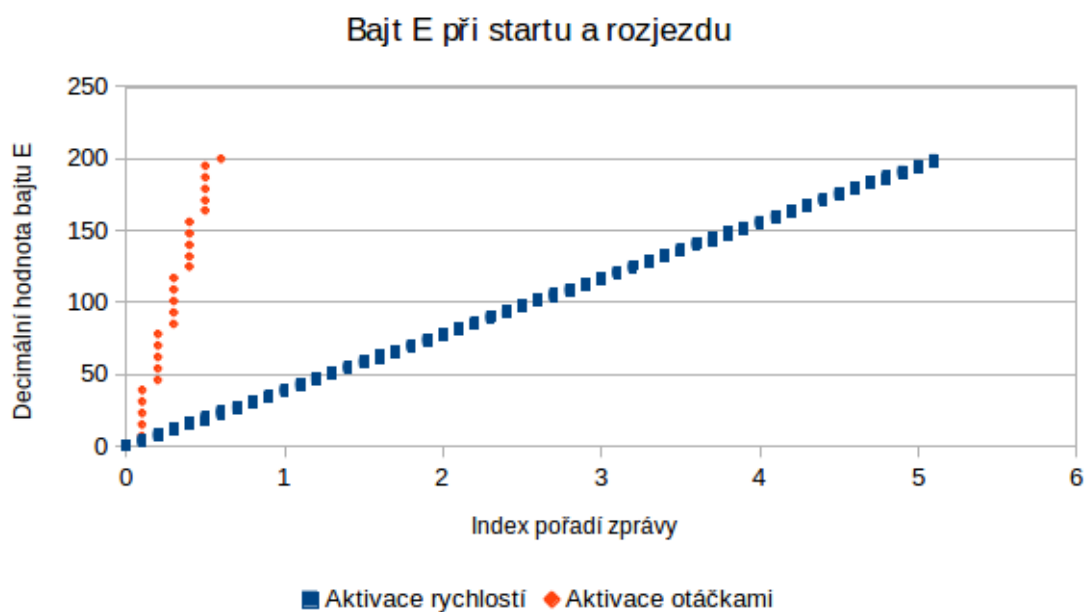
Čas [s]	ID	A	B	C	D	E	F
0,000000	3D0	02	00	00	00	00	00
...						..	
0,699094	3D0	02	00	00	00	C8	00

hodnoty i časové rozestupy byly stejné jako v testovacím vozidle.

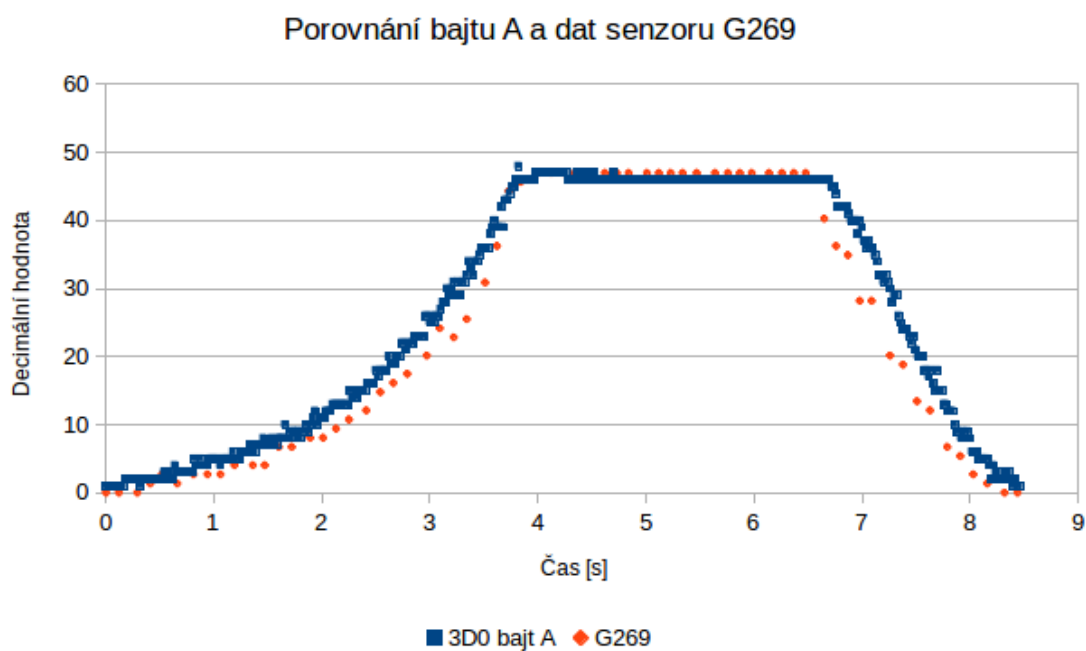
Hodnota bajtu E byla také monitorována při rozjezdu vozidla s kopce při zapnutém zapalování. Počáteční a koncová hodnota byla stejná, ale časové rozestupy byly jiné a hodnot bylo naměřeno několikrát více. Změna bajtu E z hodnoty 0x00 na 0xC8 trvala přibližně 5,1 sekundy. Naměřené hodnoty jsou vykresleny v grafu 4.1. Stejná situace byla sledována na vymontovaném posilovači při simulování rychlosti vozidla. Změna bajtu E nastala při překročení rychlosti 7 km/h, což je mezní hodnota pro aktivaci posilovače řízení při vypnutém motoru a zapnutém zapalování pro účel nouzového odtahu vozidla.

Bajt E tedy nejspíše nějakým způsobem signalizuje aktivitu elektromotoru posilovače řízení. Při nouzovém odtahu je posilovač řízení aktivován plynule, při startu motoru téměř okamžitě. V grafu jsou také vidět rozestupy mezi jednotlivými zprávami, kdy následuje vždy pět zpráv v krátkém časovém intervalu.

Dále byla vyzkoušena souvislost mezi bajtem A a daty ze senzoru zkrutu G269. Data v grafu (obrázek 4.2) byla zaznamenána při otáčení volantovou tyčí proti do-
razu řízení. Tím byl vyvinut největší zkrut na senzoru G269. Hodnoty vyznačené



Obrázek 4.1: Hodnota bajtu E CAN ID 3D0 při startování motoru a rozjezdu



Obrázek 4.2: Hodnota bajtu A CAN ID 3D0 a dat senzoru G269 při otáčení volantem

oranžovou barvou byly naměřené diagnostickým SW VCDS ve skupině 1 měřených hodnot. Hodnoty byly vynásobeny faktorem 10,74, který byl získán jak podíl maximální decimální hodnoty bajtu A (47) a maximální hodnoty ze senzoru G269 (4,375 Nm). Méně naměřených hodnot a větší časové rozestupy u dat ze senzoru G269 jsou způsobeny frekvencí měření diagnostického SW. Další odchylky mohou být způsobeny rozdílným rozlišením obou veličin, kdy nejmenší krok pro data ze senzoru G269 je 0,125 Nm a pro bajt A je to hodnota 0x01. Data v grafu (obrázek 4.2) byla naměřena při simulaci otáček vyšších než 800 otáček za minutu, tedy při simulaci zapnutého motoru. Při vypnutém motoru zůstávala hodnota bajtu A nulová.

Podobnost obou křivek je zřejmá a lze tedy konstatovat, že v bajtu A zprávy s CAN ID 3D0 je zakódována hodnota kroutícího momentu ze senzoru zkrutu G269. Z dat ale nebyl vypořizován rozdíl mezi kladnými a zápornými hodnotami. Hodnota bajtu A má však své maximum, při kterém se ani při vyvinutí větší síly jeho hodnota nemění. Stejně chování můžeme pozorovat u hodnoty senzoru G269 v diagnostickém SW, kde je maximální hodnota 4,375 Nm.

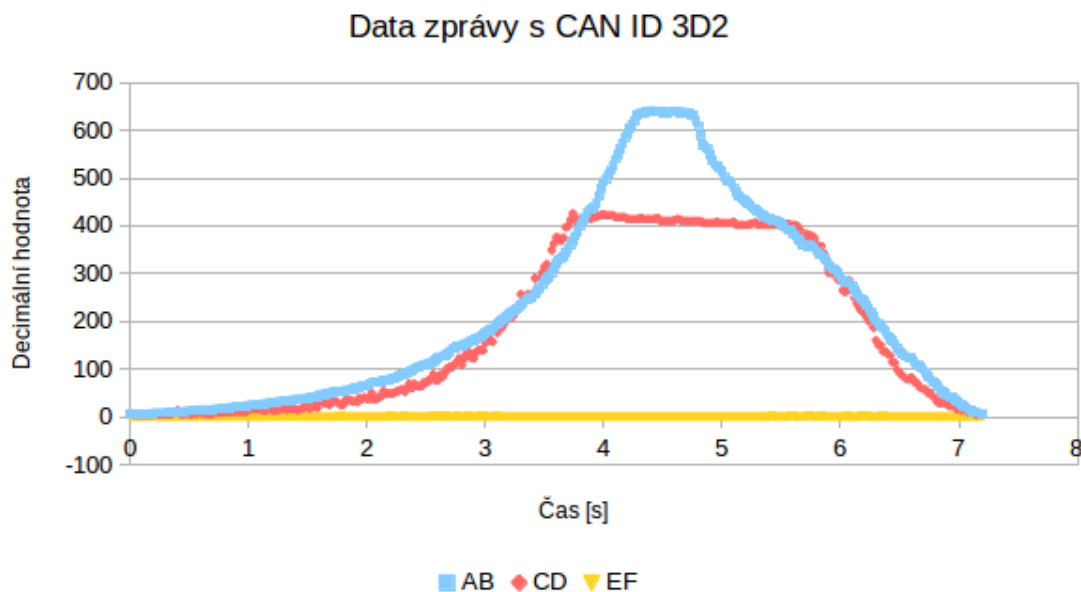
4.2 Zprávy s identifikátorem 3D2

Data zprávy s CAN ID 3D2 se měnila při otáčení volantovou tyčí. Časové intervaly mezi jednotlivými zprávami byly stejné jako u CAN ID 3D0. Ukázka zpráv je uvedena v tabulce 4.4. Ve zprávách jsou zakódovány informace o celkem čtyřech veličinách.

Tabulka 4.4: Záznam zpráv s CAN ID 3D2

Čas [s]	ID	A	B	C	D	E	F	G
0,000000	3D2	13	08	00	20	00	06	68
0,000269	3D2	13	08	00	20	00	06	68
0,000508	3D2	14	08	00	20	00	06	68
0,000684	3D2	13	08	00	20	00	06	68

Z delšího pozorování vyplynulo, že v bajtu G je zakódována informace o teplotě. Sledováním změn v bajtu G a zároveň změn hodnoty teploty pomocí diagnostického

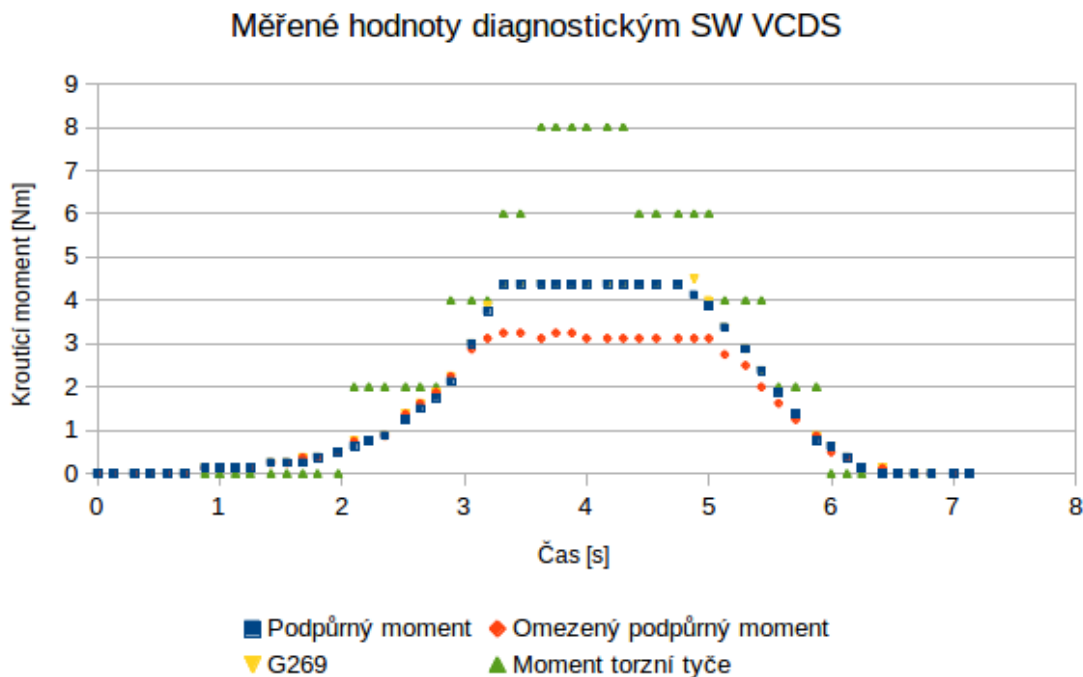


Obrázek 4.3: Otáčení volantovou tyčí - zpráva 3D2

SW v měřených hodnotách byl stanoven vzorec pro výpočet teploty s použitím decimální hodnoty bajtu G: $(G-70)$. Výpočet vypadá např. takto: $68_{16} = 104_{10}$; $(104-70) = 34 \text{ } ^\circ\text{C}$.

Pro zakódování dalších tří veličin jsou použity vždy dva bajty. Každá veličina má však jiný rozsah. U první nabývá bajt B hodnoty 0x08, 0x09 a 0x0A pro jeden směr otáčení a hodnoty 0x0C, 0x0D a 0x0E pro druhý směr otáčení. Nelze vyloučit hodnoty 0x0B a 0x0F, taková síla však nebyla vyvinuta. U druhé veličiny je rozsah nižší, hodnoty bajtu D jsou 0x20 a 0x21 pro jeden směr otáčení a 0x30 a 0x31 pro druhý směr otáčení. Pro třetí veličinu je hodnota bajtu F pro jeden směr otáčení 0x06 a pro druhý 0x07. Při výpočtu je vždy určen směr a odečtena nejnižší hodnota, kterou tyto bajty nabývají. Následně je zprava přidán bajt A, C nebo E jako méně významný.

V grafu (obrázek 4.3) jsou zaznamenány decimální hodnoty těchto tří veličin při působení na volantovou tyč proti dorazu v jednom směru stejně jako u grafu (obrázek 4.2). Data v grafu (obrázek 4.3) byla naměřena při simulaci zapnutého motoru. Jedná se nejspíše o hodnoty kroucího momentu, protože se hodnoty mění při otáčení volantovou tyčí a tím pádem také vytvářením zkrutu. Byla nalezena



Obrázek 4.4: Otáčení volantovou tyčí - diagnostický SW VCDS

podobnost s daty skupiny 5 měřených hodnot pomocí diagnostického SW VCDS. Záznamy čtyř hodnot měřitelných v této skupině jsou uvedeny v grafu (obrázek 4.4). Data grafu (obrázek 4.3) a grafu (obrázek 4.4) byla zaznamenána ve stejný moment. Méně naměřených hodnot a větší rozestupy u grafu (obrázek 4.4) jsou opět způsobeny frekvencí měření diagnostického SW VCDS. Křivka hodnot první veličiny zprávy s CAN ID 3D2 se podobá křivce momentu torzní tyče. Pro křivku druhé veličiny nelze podobnost jednoznačně určit. Křivka třetí veličiny neodpovídá žádné hodnotě měřitelné pomocí diagnostického SW.

Při simulaci vypnutého motoru, ale zapnutého zapalování, se měnila pouze hodnota bajtů A a B. Ostatní dvě veličiny zůstaly nulové. Princip kódování hodnoty zůstal stejný, ale bajt B nabýval hodnot 0x00, 0x01 a 0x02 pro jeden směr otáčení a 0x04, 0x05 a 0x06 pro druhý směr otáčení. Opět nelze vyloučit hodnoty 0x03 a 0x07 pro větší sílu. Bajty A a B by tedy mohly nést čistou informaci ze senzoru G269 bez softwarově omezené maximální hodnoty. Ostatní bajty pak nějakým způsobem kódují informaci o působení elektromotoru.

5 Chybové kódy

K odhalení většiny problémů ve vozidlech pomáhají diagnostické chybové kódy. Chybový kód je do paměti řídicí jednotky uložen tehdy, je-li některá měřená hodnota mimo výrobcem stanovený rozsah. K zjištění chybových kódů, které je řídicí jednotka posilovače řízení schopna do své paměti zapisovat, bylo použito cíleného vytváření závad. U každého chybového kódu je popsáno, při jakých podmínkách byl do paměti závad zapsán. Kódy jsou ve specifickém formátu pro vozy koncernu VW. [11]

Chybový kód 00513 - Čidlo otáček motoru (G28)

008 - Nevěrohodný signál - Kontrolka závad ZAP.

Chyba nastane při náhlé změně simulovaného úhlu natočení volantu a zároveň nulových hodnotách ostatních simulovaných veličin.

Chybový kód 00538 - Referenční napětí

002 - Dolní hranice překročena.

Chyba nastane při příliš nízkém napájecím napětí senzoru G269, tj. například při zkratu napájecích vodičů.

Chybový kód 00568 - Snímače polohy v motoru řízení

008 - Nevěrohodný signál - Kontrolka závad ZAP.

Chyba nastane, pokud úhel natočení volantu ze senzoru G85 neodpovídá úhlu získanému ze senzoru v elektromotoru V187.

Chybový kód 00573 - Čidlo momentu v řízení (G269)

012 - Chyba v elektrickém obvodu.

Chyba nastane, pokud je senzor G269 odpojen od ECU posilovače řízení.

Chybový kód 00625 - Signál rychlosti

004 - Žádný signál/komunikace - Kontrolka závad ZAP.

Chyba nastane, pokud není na sběrnici CAN zpráva s CAN ID 1A0 nesoucí informaci o rychlosti vozidla nebo jsou časové rozestupy mezi zprávami mimo toleranci.

013 - Prosím přečtěte paměť chyb - Kontrolka závad ZAP.

Chyba nastane, pokud zpráva s CAN ID 1A0 obsahuje na pozici bajtu D (tabulka 3.2) hodnotu 0xFF.

Chybový kód 00778 - Snímač úhlu řízení (G85)

004 - Žádný signál/komunikace

Chyba nastane, pokud není na sběrnici CAN zpráva s CAN ID 0C2 odesílána senzorem G85 nesoucí informaci o úhlu natočení volantu a rychlosti změny tohoto úhlu.

Chybový kód 01314 - Řídicí jednotka motoru

004 - Žádný signál/komunikace - Kontrolka závad ZAP

Chyba nastane, pokud není na sběrnici CAN zpráva s CAN ID 280 odesílána řídicí jednotkou motoru nesoucí informaci o otáčkách motoru.

Chybový kód 02546 - Stop limit řízení

005 - Žádné nebo špatné základní nastavení/přizpůsobení.

Chyba vzniká v důsledku chybějícího nastavení dorazů na obou stranách řízení.

Chybový kód 02557 - Snímač nastavování pozice úhlu natočení volantu

001 - Horní mezní hodnota překročena.

Chyba nastane, pokud hodnoty bajtů B a A překročí hodnoty 0x3C resp. 0xB7 nebo 0xBC resp. 0xB7 pro opačný směr otáčení, tj. úhel natočení volantu $\pm 21136^\circ$.

6 Základní nastavení posilovače řízení

Při simulování všech potřebných zpráv ostatních řídicích jednotek zůstal v paměti ECU posilovače řízení chybový kód **02546 - Stop limit řízení**. Chyba informuje o žádném nebo špatném základním nastavení. Řízení potřebuje získat informaci o poloze koncových dorazů na obou stranách. K tomuto základnímu nastavení existuje postup, během kterého se nepoužívá diagnostický program. Celá procedura se provádí na stojícím vozidle s běžícím motorem takto: [11]

- Vycentrovat volant do polohy $\pm 10^\circ$ od středové polohy a držet v této poloze po dobu 5 vteřin.
- Otočit volantem na levý doraz a držet v této poloze po dobu 10 vteřin.
- Otočit volantem na pravý doraz a držet v této poloze po dobu 10 vteřin.
- Vrátit volant do polohy $\pm 10^\circ$ od středové polohy.
- Vypnout motor a zapalování.
- Zapnout zapalování.
- Počkat 5 vteřin.
- Varovná kontrolka posilovače řízení by měla zhasnout. Pokud ne, proveďte krátkou testovací jízdu.

K provedení této procedury na řízení vymontovaném z vozidla je nutné simulovat změnu otáček motoru. Pokud jsou otáčky motoru vyšší než 800 ot./min., ECU posilovače vyhodnotí stav motoru jako běžící. Dále je potřeba simulovat změnu úhlu natočení volantu. Během této změny je také nutné otáčet volantovou tyčí.

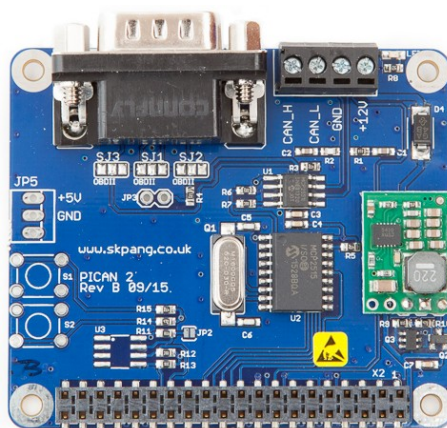
Při pokusu provést tuto proceduru na řízení mimo automobil chybový kód v paměti zůstal a zapisoval se do paměti další chybový kód **00568 - Snímače polohy**

v motoru řízení. Při otáčení volantovou tyčí získává ECU informace o poloze řízení z vnitřního senzoru elektromotoru V187. Informace o poloze řízení získává ECU také ze senzoru G85, který je simulován pomocí Raspberry. Při příliš rozdílných informacích o poloze řízení z těchto dvou senzorů; zapíše ECU do paměti chybový kód **00568**.

7 Tvorba vlastního diagnostického programu

Jako HW pro běh vlastního diagnostického programu bylo použito **Raspberry Pi 3** s rozšiřujícím modulem **PiCAN2** firmy SK Pang Electronics, který je na obrázku 7.1. Rozšiřující modul používá CAN kontrolér **MCP2515** a CAN transceiver **MCP2551**. S Raspberry Pi 3 je tento modul propojen za pomoci sběrnice SPI. Pro připojení sběrnice CAN je možné použít DE-9 konektor nebo svorkovnici. Na modulu je integrován spínaný zdroj, který umožňuje napájet Raspberry pomocí napětí 5 V. Vstupní napětí tohoto zdroje musí být v rozmezí 6-20 V, lze tedy použít autobaterii, která má napětí 12 V. Modul také disponuje 120 Ohm odporem k zakončení sběrnice CAN.

Rozšiřující modul PiCAN2 používá ovladače SocketCAN a programování modulu je možné v programovacím jazyce C nebo Python. Na Raspberry byla nainstalována linuxová distribuce Raspbian Jessie. Dále byl použit notebook s linuxovou distribucí Elementary OS 0.4.1 Loki, která je založena na Ubuntu. Notebook byl k Raspberry



Obrázek 7.1: Rozšiřující modul PiCAN2

připojen terminálem pomocí SSH. Jako pomocné nástroje k monitorování sběrnice CAN při tvorbě programu byly použity SocketCAN nástroje *can-utils*. Pro možnost monitorování a odesílání dat přímo z notebooku byl použit nástroj *cannelloni*, který pomocí UDP přenáší CAN zprávy mezi zařízeními v síti. Na notebooku byla vytvořena virtuální sběrnice CAN *vcan0*, na kterou byl přesměrován provoz reálné sběrnice *can0*.

Pro přístup ke sběrnici CAN je nutné nejprve inicializovat soket a ten poté navázat na rozhraní. Pro nastavení sběrnice CAN s přenosovou rychlostí 500 Kbps byly v terminálu Raspberry použity následující příkazy.

```
$ modprobe can
$ sudo ip link set can0 type can bitrate 500000
$ sudo ifconfig can0 up
```

Propojení reálné sběrnice *can0* na Raspberry a virtuální sběrnice *vcan0* na notebooku bylo nástrojem *cannelloni* vytvořeno následujícími příkazy. IP adresa Raspberry byla 10.42.0.2 a adresa notebooku 10.42.0.1. Byl zvolen port 20000. Fyzické spojení bylo realizováno ethernetem. Na Raspberry bylo spojení vytvořeno příkazem:

```
$ cannelloni -I can0 -R 10.42.0.1 -r 20000 -l 20000
```

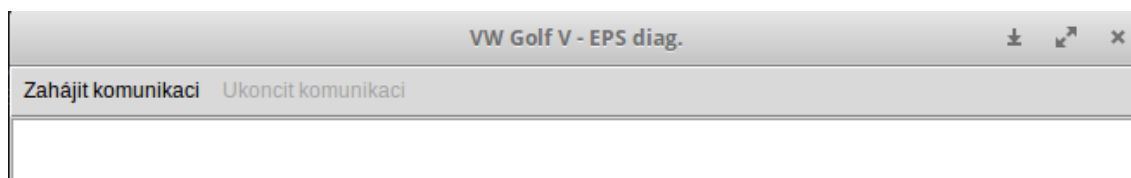
Na notebooku byl použit příkaz:

```
$ cannelloni -I vcan0 -R 10.42.0.2 -r 20000 -l 20000
```

Vlastní diagnostický program byl psán v programovacím jazyce Python. Jako vývojové prostředí byl použit PyCharm Community Edition 2017.2. Řídicí jednotka posilovače řízení komunikuje pomocí Volkswagen Transport Protocolu 2.0, který přenáší požadavky shodné s protokolem KWP2000.

7.1 Grafické rozhraní programu

Pro tvorbu grafického prostředí byl použit modul Tkinter. Po spuštění program nabízí pouze volbu zahájení komunikace (obrázek 7.2). Po kliknutí na tlačítko *Zahájit komunikaci* je vytvořeno spojení s řídicí jednotkou a načtena její identifikace. Identi-



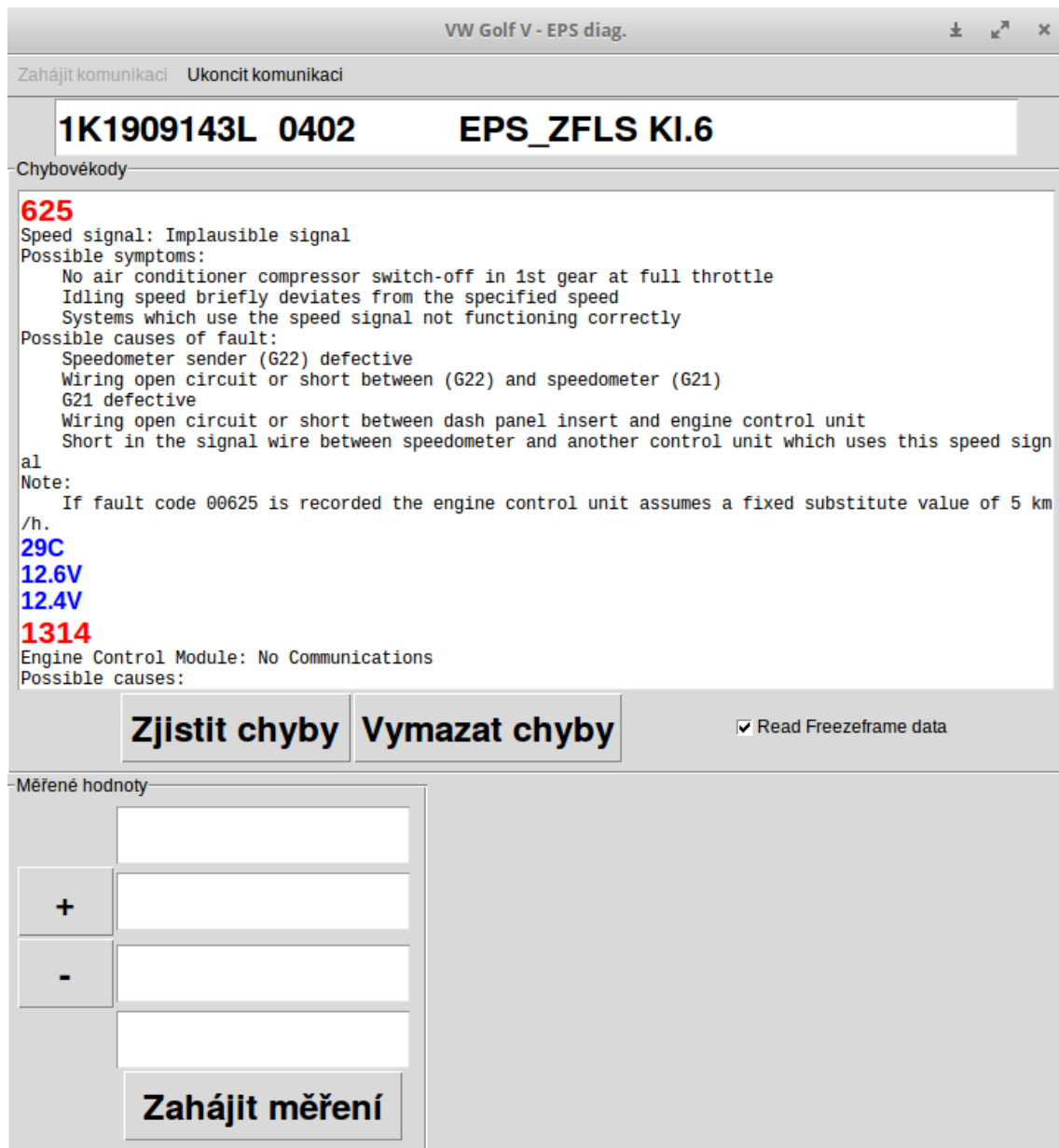
Obrázek 7.2: GUI po spuštění programu

fikace je zobrazena jako jeden textový řetězec, který je vepsán do pole pod tlačítkem pro zahájení komunikace.

Po úspěšném načtení identifikace je grafické rozhraní rozšířeno o dvě další části (obrázek 7.3). Oblast *Chybové kódy* umožňuje číst a mazat paměť závad řídicí jednotky. Chyby jsou načteny jako chybové kódy specifické pro koncern VW. Chybové kódy jsou zobrazeny červeným písmem. Ke každé chybě je zobrazen krátký popis a její možné řešení. Při zaškrtnutí pole *Read Freezeframe data* jsou ke každé chybě načteny fyzikální hodnoty, které jsou do paměti zapsány ve stejné chvíli jako daný chybový kód. *Freezeframe data* jsou vypsána modrou barvou pod popisem chybového kódu. *Freezeframe data* jsou načtena pro každý chybový kód zvlášť, proto je čtení chyb spolu s *freezeframe daty* časově náročnější než prosté čtení chybových kódů. Pro vytvoření databáze chyb byly použity informace z kapitoly 5. Tlačítkem *Vymazat chyby* je možné celou paměť závad vymazat. Pomocí opětovného načtení chybových kódů je možné zjistit, zda některé chyby nebyly vymazány.

Další část programu má označení *Měřené hodnoty*. Slouží k cyklickému načítání fyzikálních veličin, se kterými řídicí jednotka pracuje a poskytuje je pomocí diagnostické operace. Hodnoty jsou zobrazovány ve skupinách po čtyřech hodnotách. Mezi jednotlivými skupinami je možné přepínat pomocí tlačítek + a -. Zobrazování hodnot je spuštěno po stisku tlačítka *Zahájit měření*. Při měření jsou znemožněny funkce týkající se chybových kódů. Tlačítko *Zahájit měření* se změní na *Ukončit měření*.

Tlačítko *Ukončit komunikaci* je stisknutelné až po vytvoření spojení a načtení identifikace. Pomocí jeho stisknutí je komunikace s řídicí jednotkou ukončena.



Obrázek 7.3: GUI po načtení identifikace

7.2 Programová logika

Vozidla koncernu VW pro diagnostiku po sběrnici CAN často využívají svůj vlastní protokol. Transport Protocol 2.0 je transportní vrstva, pomocí které jsou přenášeny požadavky protokolu KWP2000. Aplikační vrstva KWP2000 je definována normou ISO 14230-3. Po sběrnici CAN jsou přenášeny datové pakety obsahující maximálně 8 bajtů dat. TP 2.0 řeší toto omezení. Diagnostický program komunikuje právě pomocí TP 2.0 a byl vytvořen pro komunikaci s ECU posilovače řízení. Rozšiřitelnost na ostatní řídicí jednotky vozidla by měla být snadná. Každá ECU ve vozidle používá svou komunikační adresu, princip komunikace je ale vždy stejný.

Program používá třídu *process* z balíku *multiprocessing* k tvorbě vláken. Vlákna jsou použita pro čtení a odesílání dat na sběrnici CAN. Při časově nenáročných diagnostických operacích není grafické rozhraní obnovováno. Nepředpokládá se, že by uživatel během načítání identifikačních dat nebo chybových kódů prováděl jiné operace. Princip komunikace s ECU tento způsob komunikace ani neumožňuje. Vlákno však bylo použito při čtení měřených hodnot, kdy je nutné časté obnovování grafického rozhraní a uživateli musí být umožněno přepínat mezi skupinami měřených hodnot a také měření ukončit. Pro komunikaci mezi jednotlivými vlákny byla použita třída *Queue* z balíku *multiprocessing*.

Pro komunikaci pomocí TP 2.0 byla vytvořena pomocná proměnná, která je inkrementována na základě přijatých zpráv. Slouží k odesílání potvrzení o počtu přijatých zpráv. Další pomocná proměnná je inkrementována v případě odesílání vlastních zpráv, protože každá zpráva musí obsahovat sekvenční číslo o jedničku vyšší než předchozí číslo. Dále je nutné si pamatovat poslední odeslanou zprávu pro případ, že ECU posilovače nebude připravena pro příjem zprávy a bude nutné požadavek opakovat.

Sběrnice je v programu vytvořena následujícím kódem:

```
bus = can.interface.Bus(channel='can0', bustype='socketcan_native')
```

Zprávy jsou ze sběrnice čteny vlákem ve smyčce. Hodnota prvního bajtu určuje,

zda lze očekávat další zprávu nebo se jedná o poslední zprávu a je nutné odeslat potvrzení. Ukázka kódu:

```
while True:
    message = bus.recv()
    print(message)
    a, b = settings.split(message.data[0])
    if a == 2:
        settings.increment(seqEcu)
        ackPc = increment(ackPc)
        queue_data.put(message)
    if a == 1:
        settings.increment(seqEcu)
        ackPc = increment(ackPc)
        queue_data.put(message)
        ack.data[0] = 0xB0
        ack.data[0] += ackPc
        bus.send(ack)
```

7.2.1 Zahájení komunikace a čtení identifikačních dat

Logická adresa ECU posilovače řízení je **0x09**. V tabulce 7.1 je uveden záznam zpráv z komunikace mezi řídicí jednotkou posilovače a vytvořeným diagnostickým programem. Nejprve je odeslán požadavek na vytvoření komunikačního kanálu na logickou adresu **0x09**. Po ECU je požadováno komunikovat dále pod CAN ID **300**. ECU odesílá pozitivní odpověď a požaduje po diagnostickém zařízení používat nadále CAN ID **7A8**. Poté si obě strany vymění parametry komunikace. Tímto způsobem je komunikační kanál připraven na přenos požadavků aplikační vrstvy protokolu KWP2000.

Jako první je přenášen KWP2000 požadavek **startDiagnosticSession** na zahájení diagnostiky. Požadavek je přenášen společně s parametrem, který určuje specifické chování serveru. Požadavek **startDiagnosticSession** má hodnotu **0x10** a parametr **0x89** je specifický pro výrobce. Pozitivní odpověď **0x50** spolu se stejným parametrem zpřístupní další služby, jako např. **readECUIdentification** nebo **readDiagnosticTroubleCodesByStatus**. [10]

Dále je odeslán požadavek na službu **readEcuIdentification**. Požadavek má hodnotu **0x1A** a jako parametr je použita hodnota **0x9B**. [10] Tato služba slou-

Tabulka 7.1: Záznam zpráv při vytvoření spojení s ECU posilovače řízení

Čas [s]	ID	A	B	C	D	E	F	G	H	popis
0,000000	200	09	C0	00	10	00	03	01		požadavek - nový kanál
0,006909	209	00	D0	00	03	A8	07	01		odpověď - nový kanál
0,099961	7A8	A0	0F	8A	FF	32	FF			požadavek - parametry
0,106881	300	A1	0F	CA	FF	4A	FF			odpověď - parametry
0,199856	7A8	10	00	02	10	89				požadavek - data
0,206801	300	B1								potvrzení - data
0,216874	300	10	00	02	50	89				odpověď - data
0,299798	7A8	B1								potvrzení - data
0,399933	7A8	11	00	02	1A	9B				požadavek - data
0,406818	300	B2								potvrzení - data
0,416921	300	21	00	30	5A	9B	31	4B	31	odpověď - data
0,426907	300	22	39	30	39	31	34	33	4C	data
0,436915	300	23	20	20	30	34	30	32	03	data
0,446925	300	24	80	08	00	00	00	00	00	data
0,456927	300	25	00	00	45	50	53	5F	5A	data
0,466907	300	26	46	4C	53	20	4B	6C	2E	data
0,476915	300	27	36	20	20	20	20	20	20	data
0,486810	300	18	20							data
0,655015	7A8	B9								potvrzení - data
1,156783	300	A3								test kanálu
1,467559	7A8	A3								test kanálu
1,476898	300	A1	0F	CA	FF	4A	FF			odpověď - parametry
2,467987	7A8	A3								test kanálu
2,476854	300	A1	0F	CA	FF	4A	FF			odpověď - parametry

ží ke čtení identifikačních dat řídicí jednotky. Tato identifikační data jsou u ECU posilovače přenášena pomocí osmi zpráv. Jednotlivé znaky jsou přenášeny jako hexadecimální kódy ASCII tabulky. Záznam komunikace v tabulce 7.1 byl pořízen při spojení s ECU vymontovaného posilovače řízení. Identifikační data vyjádřena pomocí anglické abecedy s vynecháním přebytečných mezer byla **1K1909143L 0402 EPS_ZFLS K1.6**.

Po načtení identifikačních dat začne program pravidelně odesílat požadavek TP 2.0 na test kanálu, který se používá k udržení aktivního komunikačního kanálu. ECU na tento požadavek odpovídá zprávou s parametry kanálu. Frekvence odesílání této zprávy byla zvolena takto: jedna zpráva za sekundu. ECU je ve stavu, kdy je připravena přijímat další požadavky a uživatel může pomocí programu provádět

další diagnostické operace.

7.2.2 Čtení a mazání chybových kódů

Ke čtení chybových kódů bylo použito služby `readDiagnosticTroubleCodesByStatus` protokolu KWP2000. ID této služby je **0x18** a jako parametr `statusOfDTCRequest` byla použita hodnota **0x02** - `requestStoredDTCAndStatus`, která požaduje validované chybové kódy uložené v paměti. Další dva bajty jsou parametry `groupOfDTC (High Byte)` a `groupOfDTC (Low Byte)`. [10] Použity byly hodnoty **0xFF** a **0x00**, které vybírají celou paměť. Záznam zpráv ze čtení chybových kódů vytvořeným diagnostickým programem je uveden v tabulce 7.2.

Tabulka 7.2: Záznam zpráv při čtení chybových kódů

Čas [s]	ID	A	B	C	D	E	F	G	H
0,000000	7A8	12	00	04	18	02	FF	00	
0,006160	300	B3							
0,016297	300	29	00	0E	58	04	02	71	E4
0,026302	300	2A	05	22	E4	09	F2	65	03
0,036207	300	1B	0A	64					

ECU odpovídá pozitivní hodnotou ID **0x58**. První parametr odpovědi `numberOfDTC` udává počet chybových kódů uložených v paměti, ve zprávě z tabulky 7.2 je to hodnota **0x04**. Dále následuje seznam chybových kódů a jejich statusů. Každé chybě náleží tři bajty, první dva udávají chybový kód a třetí status chybového kódu. Převedením chybového kódu z hexadecimální soustavy do desítkové je získán chybový kód specifický pro koncern VW. V příkladu z tabulky 7.2 jsou to chybové kódy **0x0271 = 625**, **0x0522 = 1314**, **0x09F2 = 2546**, **0x030A = 778**. Kódování statusů chyb je podrobně rozebráno v normě ISO 14230-3 [10].

Pro mazání chybových kódů byla použita služba `clearDiagnosticInformation`, která má hodnotu ID **0x14**. Parametry služby udávají funkční skupinu chybových kódů nebo specifický chybový kód. [10] V programu byly použity stejné parametry jako pro čtení chybových kódů: **0xFF** a **0x00**. Použitím těchto parametrů je zajištěno mazání celé paměti chyb. Pozitivní odpověď má hodnotu **0x54** a nese

parametry požadavku. Záznam zpráv při mazání chyb vytvořeným programem je uveden v tabulce 7.3.

Tabulka 7.3: Záznam zpráv při mazání chybových kódů

Čas [s]	ID	A	B	C	D	E	F
0,000000	7A8	15	00	03	14	FF	00
0,000921	300	B6					
0,011059	300	1E	00	03	54	FF	00

Při tvorbě programu byla použita data získaná v kapitole 5. Pro každý zjištěný chybový kód byl vytvořen textový soubor, který obsahuje popis chyby a možné řešení. Všechny tyto soubory se nachází ve složce *DTC*. Při načtení chybových kódů jsou do pole uloženy jejich decimální hodnoty. Program pak pro každý chybový kód hledá textový soubor a v případě úspěchu jeho obsah vypíše do grafického rozhraní. Pokud pro daný chybový kód neexistuje textový soubor, je zobrazen pouze chybový kód ve formátu specifickém pro koncern VW a uživatel je informován o chybějícím popisu chybového kódu.

7.2.3 Čtení freezeframe dat

Ke čtení dat uložených do paměti v době vzniku chyby bylo použito služby `readFreezeFrameData`. ID tohoto požadavku má hodnotu **0x12**. První parametr `freezeFrameNumber` udává, která data jsou požadována. Program používá hodnotu **0x00** - `OBDIIFreezeFrame`, která odkazuje na definici v normě SAE J1979. Další parametr požadavku je `recordAccessMethodIdentifier`. [10] V programu je použita hodnota **0x04** - `requestByDTC`, kdy jsou požadována data pro konkrétní chybový kód. V dalších dvou bajtech je uveden chybový kód v hexadecimální soustavě. Záznam komunikace při čtení *freezeframe* dat pro chybový kód **625** je uveden v tabulce 7.4.

Poslední tři bajty odpovědi tvoří parametr `recordAccessMethodIdentifier` a chybový kód z požadavku. [10] Zbytek *freezeframe* dat tvoří hodnoty fyzikálních veličin zaznamenaných v době zápisu chybového kódu do paměti. Každou hodnotu

Tabulka 7.4: Záznam zpráv při čtení freeze-frame dat

Čas [s]	ID	A	B	C	D	E	F	G	H
0,000000	7A8	13	00	05	12	00	04	02	71
0,000921	300	B4							
0,011059	300	2C	00	11	52	00	5C	28	00
0,021047	300	2D	1A	46	66	06	64	7B	06
0,031019	300	1E	64	79	04	02	71		

tvoří tři bajty, kde je zakódována jednotka veličiny a její hodnota. Např. bajt **0x1A** udává jednotku °C a hodnotu teploty získáme rozdílem dvou následujících bajtů: **0x66 - 0x46 = 0x20 = 32 °C**. Pro každou veličinu může být použito jiné kódování. Vzorce byly stanoveny na základě monitorování komunikace s diagnostickým SW VCDS. S rostoucím počtem chybových kódů v paměti ECU roste časová náročnost čtení *freeze-frame* dat, protože je nutné číst data pro každý chybový kód zvlášť.

V programu je použito stejné řešení jako při prostém čtení chybových kódů s tím rozdílem, že pro každý chybový kód jsou načtena *freeze-frame* data, uložena do pole a předána funkci k dekódování. Dekódovaná data jsou zobrazena v grafickém rozhraní pod popisem chybového kódu.

7.2.4 Měřené hodnoty

Ke čtení měřených hodnot byla použita KWP2000 služba **readDataByLocalIdentifier**. Podle normy ISO 14230-3 [10] mohou data této služby obsahovat analogové nebo digitální vstupní a výstupní signály, interní data a informace o stavu systému. ID této služby je **0x21**. Následuje jediný parametr, který udává skupinu měřených hodnot. Pozitivní odpověď má hodnotu **0x61** a hodnotu parametru požadavku. Následuje seznam hodnot, kdy každá z nich je zakódována pomocí tří bajtů. [10] Záznam komunikace při čtení měřených hodnot vlastním programem je uveden v tabulce 7.4.

První bajt určuje fyzikální veličinu a další dva bajty kódují její hodnotu. Kódování veličin a hodnot je stejné jako pro *freeze-frame* data v kapitole 7.2.3. Např. první veličina v tabulce 7.4 je zakódována bajty **F**, **G** a **H** ve třetím řádku. Hodnota

Tabulka 7.5: Záznam zpráv při čtení měřených hodnot

Čas [s]	ID	A	B	C	D	E	F	G	H
0,000000	7A8	12	00	02	21	01			
0,007734	300	B3							
0,017864	300	29	00	0B	61	01	1A	46	68
0,027854	300	1A	5D	7D	80	01	A0	00	

0x1A udává teplotu vyjádřenou ve °C. Hodnotu teploty opět získáme rozdílem dvou následujících bajtů: $0x68 - 0x46 = 0x22 = 34$ °C. V tabulce 7.6 jsou uvedeny odvozené vzorce pro vybrané měřené hodnoty. Pro výpočet se předpokládá použití decimálních hodnot druhého a třetího bajtu. Význam některých bajtů je složitější, lze ho zjistit ze zdrojového kódu programu v příloze.

Tabulka 7.6: Výpočet vybraných měřených hodnot

Skupina	Význam	Jednotka [A]	Výpočet [B a C]
1.1	Teplota ECU	°C [0x1A]	C-B
1.2	Kroutící moment (G269)	Nm [0x5D]	(C-128)*B*0,001
1.3	Rychlost rotoru (G28)	/min. [0x01]	C*B*0,2
2.1	Napětí relé	V [0x06]	C*0,1
2.2	Napětí baterie	V [0x06]	C*0,1
2.3	Napětí (G269)	V [0x06]	C*0,025
3.1	Otáčky motoru	/min. [0xA0]	C*32
3.2	Rychlost auta	km/h [0x07]	C
5.1	Podpurný moment	Nm [0x5D]	(C-128)*B*0,001
5.2	Omezený moment	Nm [0x5D]	(C-128)*B*0,001
5.3	Moment (G269)	Nm [0x5D]	(C-128)*B*0,001
5.4	Moment torzní tyče	Nm [0x5D]	(C-128)*B*0,1

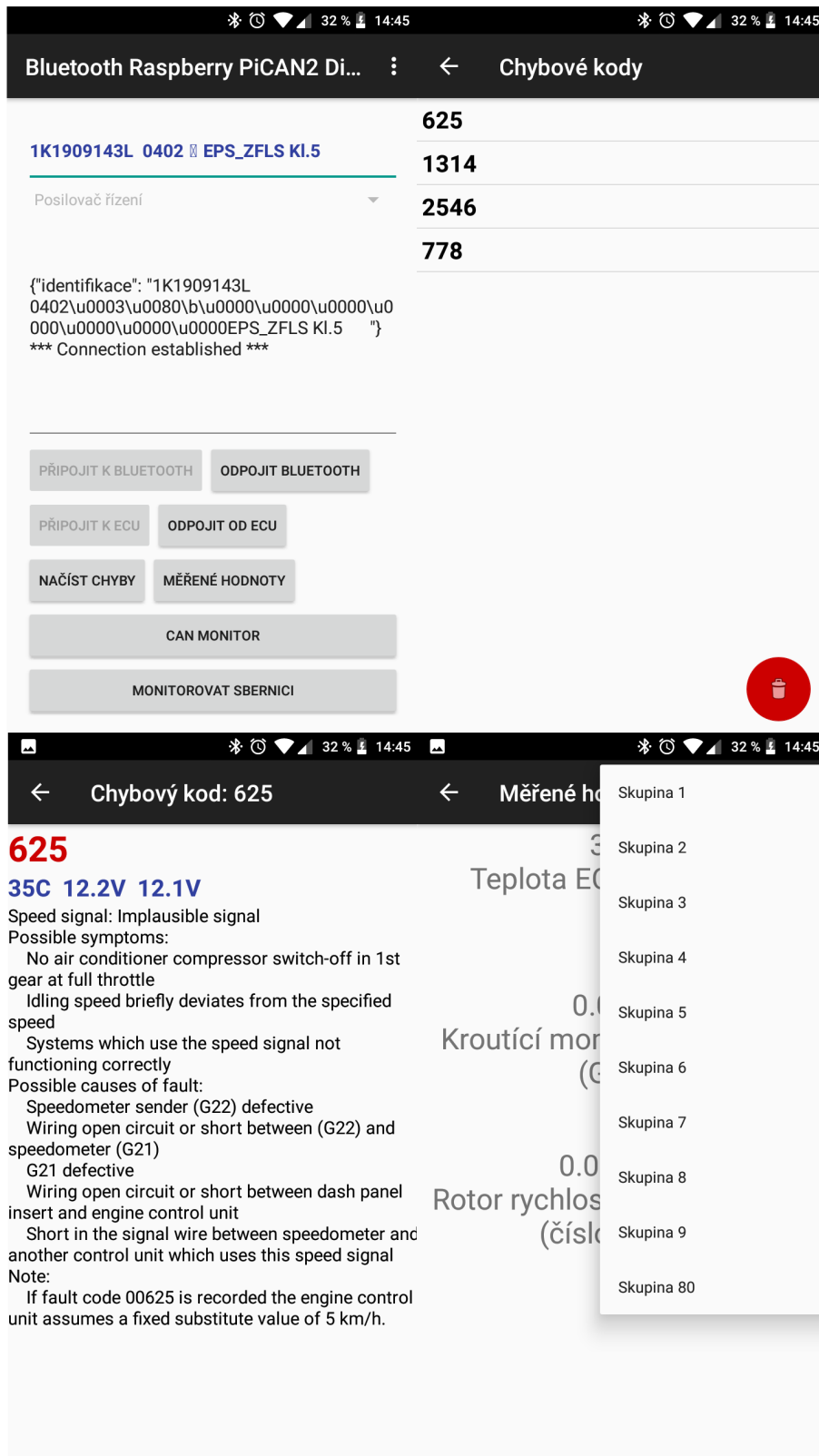
Pro čtení měřených hodnot bylo použito vlákno, které ve smyčce odesílá požadavek na čtení hodnot požadované skupiny a po přijmutí pozitivní odpovědi dekóduje hodnoty a zobrazí je v grafickém rozhraní. V průběhu měření je grafické rozhraní schopné reagovat na změnu skupiny a na požadavek na ukončení měření.

S vytvořeným programem bylo dosaženo stabilní rychlosti obnovení měřených hodnot více než 17 vzorků za sekundu. U diagnostického SW VCDS bylo naměřeno necelých 8 vzorků za sekundu.

7.3 Tvorba vlastní aplikace pro OS Android

Pro pohodlnější testování vytvořeného programu ve vozidlech byla v rámci této diplomové práce vytvořena aplikace pro operační systém Android. Bylo využito faktu, že Raspberry Pi 3 disponuje integrovaným Bluetooth 4.1 modulem. Program na Raspberry byl odlehčen o grafické rozhraní a přibylo vlákno, které obstarává komunikaci prostřednictvím technologie Bluetooth. Veškeré výpočty probíhají na straně Raspberry, mobilní telefon pouze odesílá příkazy a přijímá finální data. Pro výměnu dat byl zvolen JSON. Aplikace umožňuje provádět stejné operace jako program vytvořený pro Raspberry, pouze nahrazuje grafické rozhraní programu. GUI aplikace je vidět na obrázku 7.4, který je poskládán ze čtyř snímků jednotlivých oken aplikace.

Před použitím aplikace je nutné provést Bluetooth párování zařízení s OS Android a Raspberry Pi. Pro účely diplomové práce byla ve zdrojovém kódu použita pevná MAC adresa konkrétního zařízení. Aplikace byla testována a používána na mobilním telefonu Redmi 4X s OS Android ve verzi 8.1.

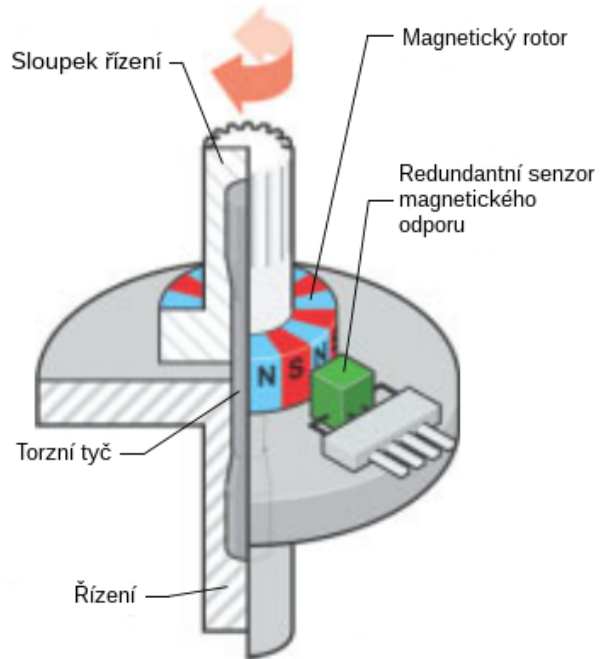


Obrázek 7.4: GUI aplikace pro OS Android

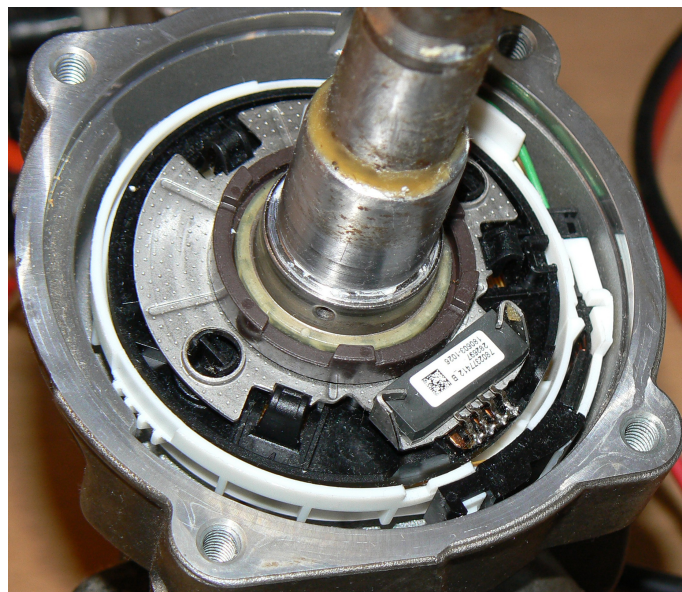
8 Senzor G269

Kroutící moment, který vyvolá řidič otáčením volantu, je vypočítán v řídicí jednotce posilovače za pomoci senzoru zkrutu G269. Senzor funguje na principu magnetického odporu. Senzor je konstruován redundantně k zajištění nejvyššího stupně bezpečnosti. Volantová tyč a posilovač řízení jsou spojeny torzní tyčí v místě, kde se nachází senzor. Na volantové tyči je umístěn magnetický kroužek, který má 24 oblastí rozdílné magnetické polarity. Na tyči posilovače řízení se nachází magnetický senzor, který je napojen na ECU posilovače řízení. Při otáčení volantem se vznikem zkrutu magnetický kroužek vůči senzoru pohybuje. Díky tomu může být měřen kroutící moment. Nákres senzoru G269 je vidět na obrázku 8.1. Senzor je s řídicí jednotkou propojen za použití čtyř vodičů (obrázek 1.2) zakončených konektorem **1K0 973 205**.

V případě selhání senzoru G269 je asistence posilovače řízení vypnuta. Vypnutí je pozvolný a jemný proces. K dosažení pozvolného vypnutí je v řídicí jednotce vypočítán náhradní signál kroutícího momentu ze senzoru v elektromotoru. Při selhání senzoru G269 je indikována chyba kontrolkou K161 v přístrojové desce.



Obrázek 8.1: Nákres senzoru G269



Obrázek 8.2: Senzor G269

8.1 Měření na senzoru

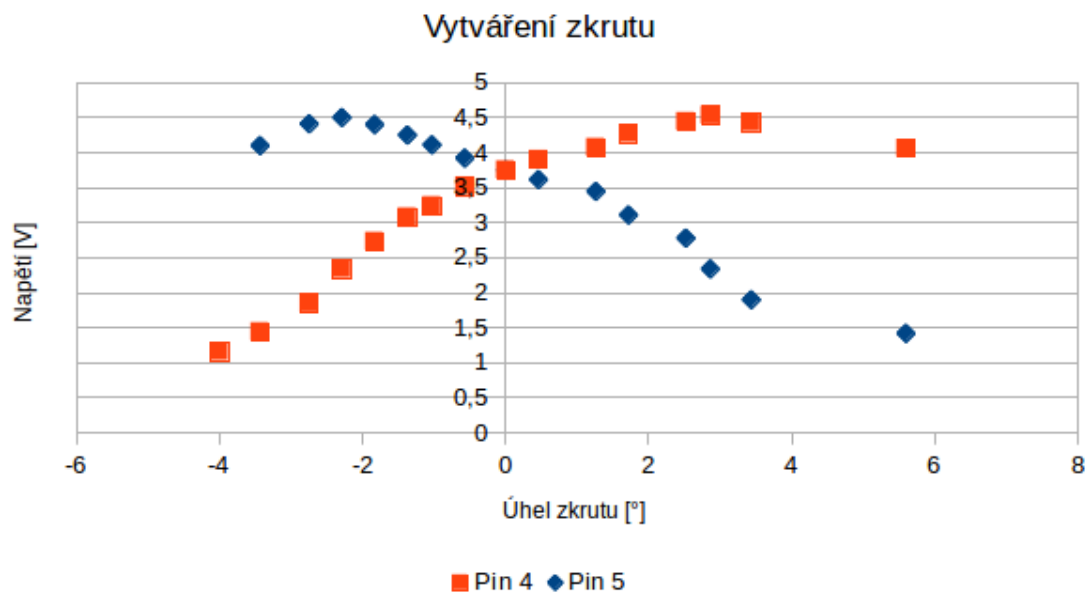
Veškeré měření bylo prováděno na vymontovaném posilovači řízení ZF generace I, který byl v kapitole 2 identifikován jako **EPS_ZFLS K1.6** s kódem dílu **1K1 909**

143 L. Při měření digitálním osciloskopem nebyla na žádném z vodičů zjištěna žádná funkce či digitální signál. V klidu bylo naměřeno pouze konstantní napětí. Při vytváření zkrutu na senzoru G269 byly na některých z vodičů pozorovány změny napětí, proto byl k dalšímu měření použit digitální multimetr **UNI-T UT61B**.

Tabulka 8.1: Měřené napětí na pinech senzoru G269 bez působení sil

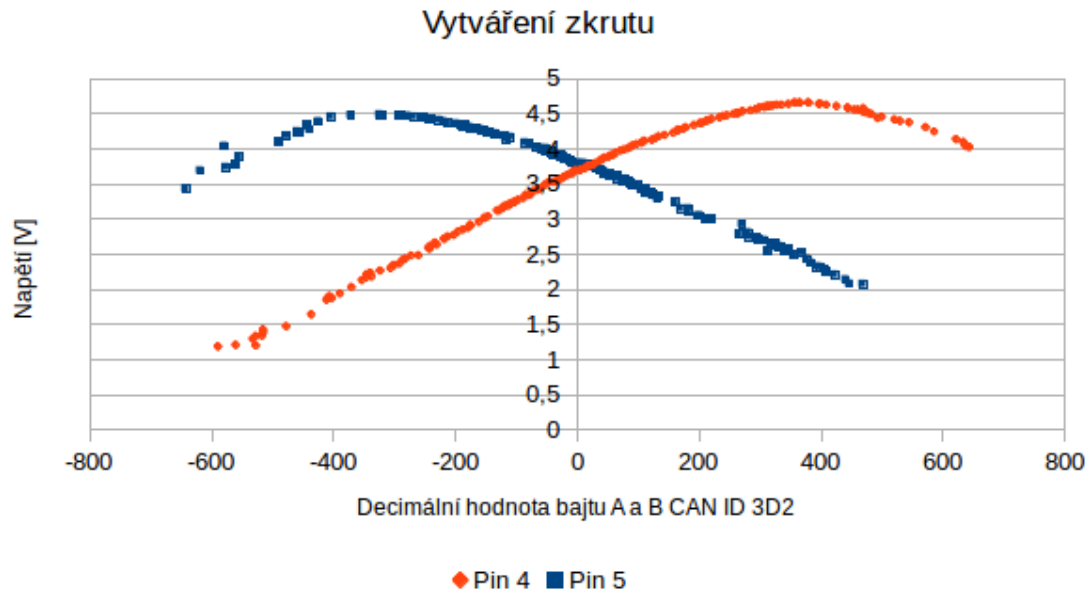
Pin	Napětí
2	5,00 V
3	0,009 V
4	3,720 V
5	3,785 V

V tabulce 8.1 jsou uvedeny naměřené hodnoty napětí na jednotlivých pinech senzoru G269 bez působení vnějších sil na řízení, tedy při nulovém kroutícím momentu. Z naměřených hodnot vyplývá, že piny 2 a 3 obstarávají napájení senzoru napětím 5 V. Piny 4 a 5 jsou výstupní signály senzoru. Pin 1 v konektoru není použit.



Obrázek 8.3: Závislost úhlu zkrutu na výstupním napětí senzoru G269

V grafu (obrázek 8.3) jsou zaznamenány hodnoty napětí výstupních signálů senzoru G269 při vytváření zkrutu. Napětí bylo měřeno multimetrem. Pomocí laserového ukazovátka byly získány hodnoty odvěsen pravoúhlého trojúhelníku. Úhel zkrutu



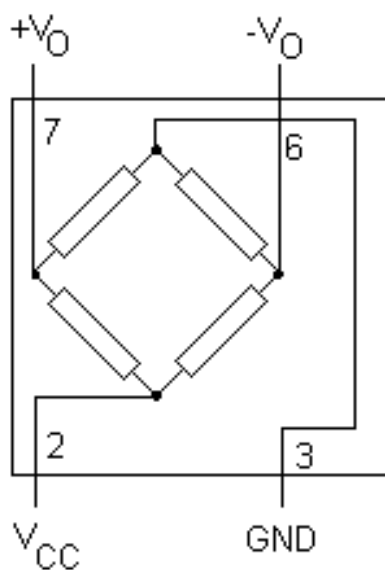
Obrázek 8.4: Závislost bajtů A a B zprávy s CAN ID 3D2 na výstupním napětí senzoru G269

byl pak vypočítán pomocí odpovídající goniometrické funkce. Kvůli fyzickým dorazům nebylo možné naměřit hodnoty napětí pro větší úhel zkrutu. Křivky napětí kopírují křivku funkce sinus. Díky posunutí křivek lze vždy jednoznačně určit směr otáčení.

V kapitole 4.2 bylo konstatováno, že bajty A a B zprávy s CAN ID 3D2 by mohly nést informaci o zkrutu ze senzoru G269. Bylo provedeno měření napětí obou výstupních signálů senzoru G269 při vytváření zkrutu vnější silou. Napětí bylo opět měřeno multimetrem **UNI-T UT61B**, který disponuje rozhraním USB a umožňuje zaznamenávat měřené hodnoty s frekvencí dva vzorky za sekundu. Zároveň byla zaznamenána data zprávy s CAN ID 3D2. Z těchto hodnot byl vytvořen graf (obrázek 8.4), který zobrazuje závislost napětí obou signálů na decimální hodnotě bajtů A a B. Lze pozorovat podobnost grafů na obrázcích 8.4 a 8.3.

8.2 Simulování funkce senzoru G269

Raspberry Pi nedisponuje piny, na kterých by bylo možné lineárně měnit napětí v rozsahu 0-5 V. Kvůli nedostupné technické dokumentaci také není jasné, jak přesně senzor G269 pracuje. Za použití odpovídajících odporů byl vytvořen *Wheatstoneův můstek* s cílem přiblížit se výstupními signály k hodnotám naměřeným v tabulce 8.1. Pomocí diagnostického programu byly smazány chybové kódy v paměti závad ECU, odpojením a následným připojením terminálu 15 na kladný pól baterie bylo řízení restartováno.



Obrázek 8.5: Wheatstoneův můstek

V paměti závad ECU zůstal chybový kód **00573** (viz kapitola 5), který signalizuje chybu v elektrickém obvodu senzoru G269. Bez technické dokumentace nelze zjistit, jaké rozmezí hodnot napětí vyhodnocuje řídicí jednotka jako chybné. Pro vyhodnocení elektrického obvodu jako chybného může ECU sledovat další fyzikální veličiny, jako např. odpor. Senzor G269 se bez technické dokumentace nepodařilo bezchybně simulovat.

9 Testování vytvořeného diagnostického programu

Vytvořený diagnostický program byl testován v obou verzích. Ve verzi s GUI pro Raspberry byl testován na posilovačích řízení, které byly vymontovány z automobilu. Bluetooth verze pro OS Android byla testována přímo ve vozidlech, kde by byl problém s připojením obrazovky k Raspberry. Se všemi testovanými posilovači řízení program navázal spojení a získal identifikační data řídicí jednotky.

Posilovač 1K1909143L

Pro tento posilovač řízení byl diagnostický program primárně vyvíjen, všechny funkce programu tedy fungovaly bez problému. Všechny chybové kódy, které byly v paměti ECU uloženy, byly v databázi chybových kódů programu. Měřené hodnoty byly dostupné ve skupinách 1-6, 80, 125 a 126.

Posilovač 1K1909144G

Tento posilovač byl namontován v testovacím vozidle VW Golf Mk5 z roku 2004. V paměti ECU nebyly uloženy žádné chybové kódy. Měřené hodnoty byly dostupné ve skupinách 1-8, 80, 125 a 126. Ve skupině 7 byla data ze senzoru úhlu natočení volantu G85. Data byla dekodována a vzorce pro výpočet implementovány do programu. Ve skupině 8 byla informace o počtu chybových kódů.

Posilovač 1K1909144M

Tento posilovač řízení zapůjčený firmou TRW byl vymontován z vozidla, ale nebyl znám jeho původ. V paměti ECU bylo uloženo 5 chybových kódů. Jeden z nich nebyl v databázi programu. Jednalo se o chybový kód **01299**, který informuje o ztrátě komunikace s diagnostickým rozhraním datové sběrnice. Chybový kód se podařilo odstranit simulováním zprávy ECU *Gateway*. Chybový kód byl doplněn do databáze programu. Měřené hodnoty bylo možné číst ve skupinách 1-8, 80, 125 a 126. Ve skupině 2 však chyběla informace o napájecím napětí senzoru G269.

10 Závěr

Díky informacím získaným při tvorbě semestrálního projektu *Tvorba simulátoru pro testování EPS posilovače VW Golf V* [2] bylo možné testovat řídicí jednotku posilovače řízení mimo automobil. Vytvořený simulátor byl v této práci vylepšen a použit k simulaci ostatních řídicích jednotek vozidla. Testováním řídicí jednotky bylo nalezeno devět chybových kódů, které ECU používá. Bylo popsáno, v jakých situacích mohou být chybové kódy zapsány do paměti ECU. Monitorováním sběrnice CAN byly identifikovány zprávy, které řídicí jednotka posilovače dává k dispozici ostatním jednotkám automobilu. Data zpráv byla identifikována a byl popsán jejich možný význam.

Vytvořený diagnostický program pro Raspberry Pi 3 používá ke komunikaci po sběrnici CAN rozšiřující modul PiCAN, který je možné zakoupit za cenu pohybující se kolem 1000 Kč. Výhodou tohoto řešení je všestrannost a možnost použití Raspberry Pi i k jiným účelům. Oproti převodníku *ELM 327* použitému v bakalářské práci *Analýza CAN komunikace EPS posilovače VW Golf V* [1] nemá tento modul žádné problémy monitorovat sběrnici CAN přímo v automobilu, kde jsou cyklicky odesílány desítky zpráv každých pár milisekund. Vytvořený diagnostický program zvládá všechny základní funkce a byla vytvořena jeho verze pro komunikaci s aplikací běžící na OS Android pomocí technologie Bluetooth, která byla také vytvořena v rámci této diplomové práce.

Dále byla v práci popsána funkce senzoru G269 a jeho konstrukce u řízení ZF generace I. Analýza jeho komunikace s řídicí jednotkou posilovače odhalila princip komunikace. Byla identifikována data pocházející ze senzoru ve zprávě, kterou ECU posilovače odesílá ostatním jednotkám. Kvůli nedostupné dokumentaci se nepoda-

řilo funkci senzoru bezchybně simulovat. Po analýze bylo zjištěno, že senzor G269 komunikuje analogově, nikoliv digitálně, jak se původně předpokládalo při zadání diplomové práce. Počítač Raspberry Pi 3 nedisponuje výstupem s lineárně regulovatelným napětím. Pro kontrolu nad signály senzoru G269 by bylo možné vytvořit jednotku, která by používala otestovaný senzor G269 a jemný elektromotor k pohybu magnetickým kroužkem.

Diagnostický program by mohl být dále rozšířen o část, která by byla určena k monitorování sběrnice CAN. Postupným testováním ostatních řídicích jednotek automobilu by bylo možné identifikovat zprávy konkrétních ECU a dekodovat data v nich obsažená. Díky tomu by bylo možné vytvořit monitorovací program s daleko vyšší frekvencí obnovy dat než nabízí konvenční diagnostické programy. Tyto zprávy však nejsou popsány žádnou normou a identifikace dat by tak byla časově velmi náročná. Bylo by také potřeba diagnostický HW připojovat přímo na vnitřní sběrnici CAN, protože sběrnice diagnostické zásuvky *OBDII* je filtrována ECU *gateway*.

Literatura

- [1] DROZD, Tomáš. *Analýza CAN komunikace EPS posilovače VW Golf V*. Liberec, 2015. Bakalářská práce. Technická univerzita v Liberci. Vedoucí práce Ing. Tomáš Martinec, Ph.D.
- [2] DROZD, Tomáš. *Tvorba simulátoru pro testování EPS posilovače VW Golf V*. Liberec, 2016. Semestrální projekt. Technická univerzita v Liberci. Vedoucí práce Ing. Tomáš Martinec, Ph.D.
- [3] PLÍVA, Z., J. DRÁBKOVÁ, J. KOPRNICKÝ a L. PETRŽÍLKA. *Metodika zpracování bakalářských a diplomových prací. 2. upravené vydání*. Technická univerzita v Liberci, FM, 2014. ISBN 978-80-7494-049-1.
Dostupné z: [doi:10.15240/tul/002/2014-11-002](https://doi.org/10.15240/tul/002/2014-11-002)
- [4] VOLKSWAGEN AG. *The electro-mechanical power steering with dual pinion*. Wolfsburg, 2003. Self-study programme 317.
- [5] VOLKSWAGEN AG. *The Golf 2004 electrical system*. Wolfsburg, 2003. Self-study programme 319.
- [6] VOLKSWAGEN AG. *Data exchange on the CAN bus I*. Wolfsburg, 2003. Self-study programme 238.
- [7] VOLKSWAGEN AG. *Data transfer on CAN data bus II*. Wolfsburg, 2003. Self-study programme 269.
- [8] ROBERT BOSCH GmbH. *CAN Specification Version 2.0*. Postfach 300204. D-7000 Stuttgart 30. 1991.

- [9] VW Transport Protocol 2.0 (TP 2.0) for CAN bus. *Jared Wiltshire's personal website and blog* [online]. [cit. 2018-04-22].
Dostupné z: <https://jazdw.net/tp20>
- [10] ISO 14230-3:1999. *Road vehicles — Diagnostic systems — Keyword Protocol 2000 — Part 3: Application layer*. International Organization for Standardization.
- [11] *Ross-Tech Wiki* [online]. Copyright © 2000 [cit. 22.04.2018].
Dostupné z: http://wiki.ross-tech.com/wiki/index.php/Main_Page

Obsah přiloženého CD

- **Latex** - složka se zdrojovými soubory \LaTeX textu diplomové práce
- **Android** - složka se zdrojovými soubory aplikace pro OS Android
- **Raspberry** - složka se zdrojovými soubory programu pro Raspberry Pi
 - **Bluetooth** - složka se zdrojovými soubory programu ve verzi s Bluetooth
 - **Desktop** - složka se zdrojovými soubory programu ve verzi bez Bluetooth
- **Grafy** - složka s grafy a zdrojovými daty obsaženými v diplomové práci