



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

DIAGNOSTIKA DAT V MODERNÍCH ŘÍDÍCÍCH JEDNOTKÁCH MOTORŮ

DATA DIAGNOSTICS IN MODERN CAR CONTROL UNITS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

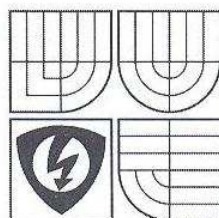
Bc. ONDŘEJ HÁJEK

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. ALEŠ PROKEŠ, Ph.D.

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Ondřej Hájek

Ročník: 2

ID: 78182

Akademický rok: 2009/10

NÁZEV TÉMATU:

Diagnostika dat v moderních řídicích jednotkách motorů

POKYNY PRO VYPRACOVÁNÍ:

V rámci dostupných informací se seznámte s koncepcemi současných řídicích jednotek motorů různých výrobců. Zaměřte se především na jednotky řasy EDC používaných koncernem VW. Prostudujte používané komunikační protokoly v rámci vozidla i s externími zařízeními a možnostmi analýzy a modifikace uložených dat.

Navrhněte hardwarovou realizaci potřebného komunikačního interface pro komunikaci jednotky s okolím (USB/CAN-OBDD, apod.) a realizujte jej v podobě funkčního vzorku. Ve vhodném programovacím prostředí vytvořte jednoduché uživatelské rozhraní a sadu programů umožňujících komunikaci po sběrnici CAN a základní diagnostiku řídicích jednotek.

DOPORUČENÁ LITERATURA:

[1] NAVET, N. and SIMON-LION, F. Automotive Embedded Systems Handbook (Industrial Information Technology) . CRC Press, 2008

[2] VALDORF, J. and GESSNER, W. Advanced Microsystems for Automotive Applications 2006. Springer, 2006.

Termín zadání: 8.2.2010

Termín odevzdání: 21.5.2010

Vedoucí práce: doc. Ing. Aleš Prokeš, Ph.D.

Konzultanti diplomové práce:

prof. Dr. Ing. Zbyněk Raida
předseda oborové rady



UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

LICENČNÍ SMLOUVA

POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Bc. Ondřej Hájek
Bytem: V Průhonech 259, Vysoké Mýto, 566 01
Narozen/a (datum a místo): 5. února 1986 ve Vysokém Mýtě

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 53, Brno, 602 00
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika
(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- disertační práce
 - diplomová práce
 - bakalářská práce
 - jiná práce, jejíž druh je specifikován jako
- (dále jen VŠKP nebo dílo)

Název VŠKP: Diagnostika dat v moderních řídicích jednotkách motorů

Vedoucí/ školitel VŠKP: doc. Ing. Aleš Prokeš, Ph.D.

Ústav: Ústav radioelektroniky

Datum obhajoby VŠKP: _____

VŠKP odevzdal autor nabyvateli*:

- v tištěné formě – počet exemplářů: 2
- v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.

3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.

4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

* hodící se zaškrtněte

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy
(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 12. května 2010

.....
Nabyvatel

.....
Autor

Abstrakt

Tato diplomová práce se zabývá návrhem a konstrukcí diagnostického rozhraní automobilových řídicích jednotek motorů řady EDC. Diagnostické rozhraní zajistí komunikaci mezi automobilem a počítačem prostřednictvím rozhraní USB. Uživatel bude moci ve vytvořené počítačové aplikaci zadávat diagnostické pokyny. Pro vytvoření aplikace je nutné získat informace o použitých protokolech a základní znalosti v oblasti hardware automobilu. Při návrhu řešení je nutné pochopit standard OBD2 a komunikační protokol CAN, případně i KWP. Hlavním účelem práce je shromáždit dostatek věrohodných informací o diagnostice automobilů, aby bylo možné na tuto práci v budoucnu navázat.

Klíčová slova

automobil, řídicí jednotka, diagnostické rozhraní, motor, CAN sběrnice, mikrokontroler

Abstract

Presented Diploma thesis is aimed on design and implementation of car diagnostic interface of engine control units EDC. Diagnostic interface provide communication between car and computer by USB interface. User will be able to task commands by developed computer application. For developing application is necessary to get information about used protocols and basic knowledge in car's hardware. Understanding OBD2 standard and communication protocol CAN, eventually KWP is necessary for design the solution. Main purpose of this thesis is to store enough usable information about car diagnostic for future continuation.

Keywords

car, control unit, diagnostic interface, engine, CAN bus, microcontroller

HÁJEK, O. *Diagnostika dat v moderních řídicích jednotkách motorů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2010. 46 s., 2 s. příloh. Diplomová práce. Vedoucí práce: doc. Ing. Aleš Prokeš, Ph.D.

Prohlášení

Prohlašuji, že svoji diplomovou práci na téma Diagnostika dat v moderních řídicích jednotkách motorů jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 12. května 2010

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce doc. Ing. Aleši Prokešovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mého semestrálního projektu.

V Brně dne 12. května 2010

.....
podpis autora

OBSAH

SEZNAM POUŽITÝCH OBRÁZKŮ	- 3 -
SEZNAM POUŽITÝCH TABULEK	- 3 -
1 ÚVOD	- 4 -
1.1 Rozbor zadání	- 4 -
1.2 Základní informace	- 4 -
1.3 Standard OBD	- 4 -
1.4 Úvod do diagnostiky automobilů	- 6 -
1.4.1 OBD2 diagnostika	- 6 -
1.4.2 Diagnostika konkrétního výrobce automobilů	- 7 -
1.4.3 Chiptuning	- 7 -
2 PROTOKOLY	- 10 -
2.1 Keyword Protokoly	- 10 -
2.1.1 Úvod	- 10 -
2.1.2 Parametry sběrnice	- 10 -
2.1.3 Formát rámce	- 11 -
2.1.4 Komunikace	- 11 -
2.1.5 Rychlá inicializace	- 11 -
2.1.6 Pomalá inicializace	- 12 -
2.1.7 Odpovědi řídicí jednotky na žádosti testeru	- 12 -
2.2 CAN	- 13 -
2.2.1 Základní informace	- 13 -
2.2.2 Technické parametry	- 13 -
2.2.3 Komunikace po sběrnici	- 13 -
2.2.4 Formát rámce	- 14 -
2.2.5 Použití v automobilech	- 15 -
2.2.6 Vyšší vrstvy CAN - TP2.0	- 16 -
2.3 FlexRay	- 17 -
2.3.1 Historie	- 17 -
2.3.2 Komunikace	- 17 -
3 ŘÍDICÍ JEDNOTKY	- 18 -
3.1 Obecně o ECU	- 18 -
3.2 Siemens DDE a DME	- 19 -
3.3 Bosch EDC	- 20 -
3.3.1 Úvod	- 20 -
3.3.2 Blokové schéma EDC 15P	- 20 -
3.3.3 Konstrukce řídicí jednotky EDC	- 21 -
3.3.4 Mikrokontroler MPC564	- 21 -
4 DIAGNOSTIKA	- 22 -
4.1 Vnitřní uspořádání elektroniky vozu	- 22 -
4.2 Detekce chyb	- 22 -
4.3 Chyby DTC	- 23 -
5 NÁVRH DIAGNOSTICKÉHO ROZHRAŇÍ	- 24 -
5.1 Úvod	- 24 -
5.2 Návrh 1: Diagnostika s univerzálním mikroprocesorem	- 24 -
5.2.1 Schéma zapojení	- 25 -
5.2.2 Deska plošných spojů diagnostického rozhraní s MCF51JM64	- 26 -

5.3 NÁVRH 2: DIAGNOSTIKA SE SPECIÁLNÍM OBVODEM ELM 327	- 27 -
5.3.1 Blokové schéma s ELM327	- 27 -
5.3.2 Schéma diagnostického rozhraní s ELM327	- 28 -
5.3.3 Deska plošných spojů diagnostického rozhraní s ELM327	- 29 -
6 VLASTNÍ APLIKACE	- 31 -
6.1 Vývoj aplikace pro ELM327	- 31 -
6.1.1 Základní informace	- 31 -
6.1.2 Příkazy	- 31 -
6.1.3 Způsob komunikace	- 31 -
6.1.4 Vytvořená aplikace	- 33 -
6.2 VÝVOJ APLIKACE PRO MCF51JM64	- 38 -
7 ZAJÍMAVÉ PRODUKTY	- 41 -
7.1 Simulátor řídicí jednotky	- 41 -
7.2 Matlab Vehicle Network Toolbox	- 42 -
8 ZÁVĚR	- 43 -
LITERATURA	- 45 -
SEZNAM ZKRATEK	- 46 -
SEZNAM PŘÍLOH	- 46 -

SEZNAM POUŽITÝCH OBRÁZKŮ

<i>Obrázek 1: Konektor OBDII - J1962, typ A, převzato z [9]</i>	- 5 -
<i>Obrázek 2: Konektor OBDII - J1962, typ B, převzato z [9]</i>	- 5 -
<i>Obrázek 3: Rozdělení automobilové diagnostiky</i>	- 6 -
<i>Obrázek 4: Diagnostický tester V.A.G. 1552, převzato z [5]</i>	- 7 -
<i>Obrázek 5: Komunikace ECU prostřednictvím BDM rozhraní, převzato z [20]</i>	- 8 -
<i>Obrázek 6: Ukázka modifikace palivové mapy v programu WinOLS</i>	- 9 -
<i>Obrázek 7: Komunikační schéma KWP (jednovodičová varianta), dle [12]</i>	- 10 -
<i>Obrázek 8: Rámec protokolu ISO 14230, dle [13]</i>	- 11 -
<i>Obrázek 9: Standardní formát datového rámce CAN 2.0A dle [7]</i>	- 14 -
<i>Obrázek 10: Rozšířený formát datového rámce CAN 2.0B dle [7]</i>	- 14 -
<i>Obrázek 11: Struktura dat při komunikaci po CAN dle OBD2</i>	- 16 -
<i>Obrázek 12: Repräsentace dat CAN2.0A v protokolu TP2.0 dle [18]</i>	- 16 -
<i>Obrázek 13: Hlavní části elektronického řízení motoru</i>	- 18 -
<i>Obrázek 14: Řídící jednotka Siemens DDE, převzato z [11]</i>	- 19 -
<i>Obrázek 15: Schéma řídicí jednotky EDC 15P, dle [4]</i>	- 20 -
<i>Obrázek 16: Konstrukce řídicí jednotky EDC, převzato z [4]</i>	- 21 -
<i>Obrázek 17: Zapojení řídicích jednotek v automobilu Škoda Fabia, převzato z [5]</i>	- 22 -
<i>Obrázek 18: Blokové schéma diagnostického kabelu</i>	- 24 -
<i>Obrázek 19: Schéma zapojení diagnostického kabelu</i>	- 25 -
<i>Obrázek 20: DPS diagnostického kabelu (šířka 43 mm, výška 38 mm)</i>	- 26 -
<i>Obrázek 21: Osazovací výkres</i>	- 26 -
<i>Obrázek 22: Blokové schéma s ELM327</i>	- 28 -
<i>Obrázek 23: Schéma diagnostického rozhraní s ELM327</i>	- 28 -
<i>Obrázek 24: Deska plošných spojů s ELM327 (šířka 56 mm, výška 42 mm)</i>	- 29 -
<i>Obrázek 25: Osazovací výkres DPS s ELM327</i>	- 29 -
<i>Obrázek 26: Zjednodušený algoritmus základní komunikace ve vytvořené aplikaci</i>	- 32 -
<i>Obrázek 27: Vzhled aplikace – panel Základní komunikace</i>	- 33 -
<i>Obrázek 28: Vzhled aplikace – panel Přístrojová deska</i>	- 33 -
<i>Obrázek 29: Vzhled aplikace – panel Chyby DTC</i>	- 34 -
<i>Obrázek 30: Vzhled aplikace – panel Měření</i>	- 34 -
<i>Obrázek 31: Průběh výkonu vozu Škoda FabiaII 1,4 TDI na 3. rychlostní stupeň</i>	- 36 -
<i>Obrázek 33: Vytvoření nového projektu v CodeWarrioru 6.3</i>	- 38 -
<i>Obrázek 34: Nastavení CPU v režimu Processor Expert</i>	- 38 -
<i>Obrázek 35: Plánovaný algoritmus mikrokontroleru MCF51JM64</i>	- 39 -
<i>Obrázek 36: Simulátor řídicí jednotky mOByDic 1610, převzato z [22]</i>	- 41 -
<i>Obrázek 37: Komunikace toolboxu s okolními CAN moduly, převzato z [23]</i>	- 42 -

SEZNAM POUŽITÝCH TABULEK

<i>Tabulka 1: Význam pinů konektoru OBDII J1962 dle [9]</i>	- 5 -
<i>Tabulka 2: Zapojení pinů v zásuvce dle komunikačních standardů</i>	- 5 -
<i>Tabulka 3: 11b CAN identifikátory dle [17]</i>	- 15 -
<i>Tabulka 4: 29b CAN identifikátory dle [17]</i>	- 15 -
<i>Tabulka 5: Struktura standardu chybových kódů v OBD2</i>	- 23 -
<i>Tabulka 6: Rozpis elektronických součástek pro rozhraní s MCF51JM64</i>	- 27 -
<i>Tabulka 7: Rozpis elektronických součástek pro rozhraní s ELM327</i>	- 30 -
<i>Tabulka 8: Testované automobily rozhraním ELM327</i>	- 44 -

1 ÚVOD

1.1 Rozbor zadání

Tato práce se zabývá v rámci dostupných informací koncepcemi moderních řídicích jednotek motorů. Z důvodu značných rozdílů a objemu informací je preferována velmi rozšířená řada EDC od firmy Bosch. Základem pro práci s řídicími jednotkami motorů je znát používané komunikační protokoly, které se v automobilech vyskytují. Práce bude nejvíce zaměřena na protokol CAN, který v současné době patří k nejčastěji používaným protokolům. Další oblastí, která zde bude popsána, je diagnostika automobilu. Ta se zabývá detekcí chyb, archivací chyb v paměti a způsobem, jak je lze číst a mazat. Dle získaných informací bude navrženo diagnostické rozhraní, které zprostředkuje komunikaci počítače s automobilem. Vytvořená aplikace pak umožní provádět různé diagnostické činnosti.

1.2 Základní informace

Výroba automobilů prošla od roku 1885, kdy Carl Benz vyrobil první automobil, znamenitým vývojem. V dnešní době je automobilový průmysl 6. největší průmysl na světě. Například za rok 2007 vyprodukoval přibližně 70 milionů automobilů [1]. Za posledních 10 let exponenciálně vzrostl počet počítačem řízených funkcí v automobilu. Nynější automobily střední třídy obsahují řádově 10000 elektrických zařízení, 70 mikroprocesorů, 300 konektorů a délku kabelového svazku 1,6 km. Mechanické a hydraulické systémy jsou postupně nahrazovány elektrickými. Zařízení komunikují prostřednictvím několika druhů sítí od jednoduché a levné LIN nebo TTP/A přes složitější CAN až po FlexRay.

1.3 Standard OBD

Zpočátku diagnostiky automobilů každý výrobce stanovil vlastní standard pro diagnostiku automobilu. Takové diagnostické funkce, které nesplňují žádný z mezinárodních standardů se říká OBD1 (*OnBoard Diagnostics*). OBD1 kontroluje funkčnost základních elektronických komponent, které mají vliv na emise, z hlediska zkratů a přerušení vedení. Myšlenka, aby bylo možné jedním diagnostickým přístrojem diagnostikovat vozidla různých výrobců vedla k vytvoření normy OBD2.

Jedním z hlavních přínosů OBD2 je zavedení jednotného systému pro automobilovou diagnostiku. Elektronika vozu má sama rozpoznat závadu a pokud se jedná o závažnou závadu, tak ji signalizuje řidiči. Tyto signalizace se nazývají MIL (*Malfunction Indicator Light*), neboli indikátor poruchy. Standard OBD2 definuje jednotný konektor, 16ti pinový J1962, který musí být umístěn v blízkosti sedadla řidiče, tedy na spodní straně palubní desky nebo na středovém panelu. Konektor OBD2 je 2 typů: typ A a typ B viz obrázek 1 a 2. Typy se liší tvarem středové části oddělující řady pinů. Nutno upozornit, že auto vybavené touto zásuvkou nemusí podporovat OBD2 protokol. Například 16ti pinovou zásuvku nalezneme již ve vozu Škoda Felicia od roku 1995, ale OBD2 kompatibilní není. Rovněž je definován formát zasílaných zpráv a seznam chybových kódů DTCs (*Diagnostic Trouble Codes*). V Evropě musí povinně automobily splňovat normu OBD2 u benzínových od roku výroby 2000 a u dieselových až od roku 2003.

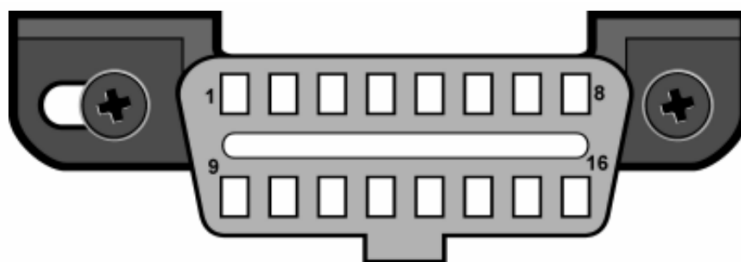
Norma OBD2 má 4 různé komunikační hardwarové rozhraní:

ISO 9141: evropská a asijská norma, na níž jsou založeny keyword protokoly

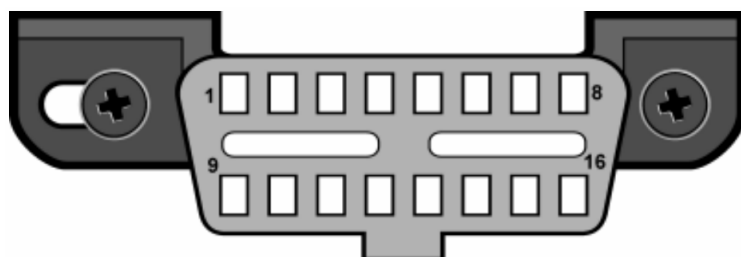
ISO 15765: CAN-Bus

SAE J1850 PWM: *Pulse Wide Modulation*, používaná u vozů Ford

SAE J1850 VPW: *Variable Pulse Width*, používaná u vozů GM



Obrázek 1: Konektor OBDII - J1962, typ A, převzato z [9]



Obrázek 2: Konektor OBDII - J1962, typ B, převzato z [9]

Tabulka 1: Význam pinů konektoru OBDII J1962 dle [9]

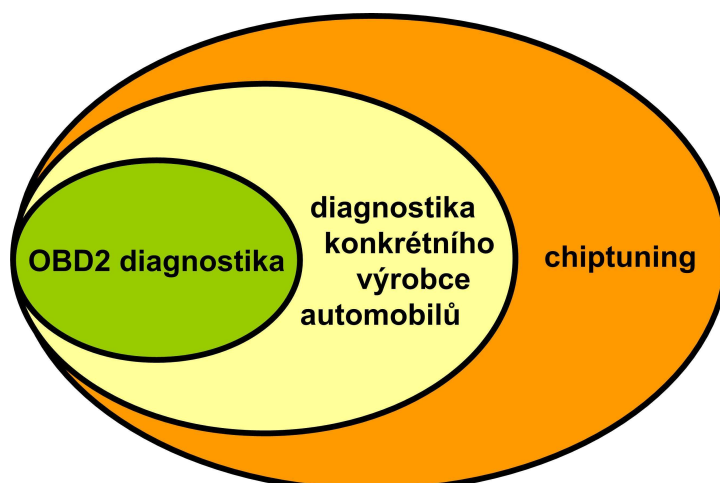
pin	význam pinů
1	GM SAE J2411: 1-vodičový CAN s malou rychlostí
2	SAE J1850 PWM Bus (+) nebo VPW Bus
3	Chrysler CCD+ (není ODB)
4	kostra vozidla
5	komunikační kostra
6	CAN-Bus High
7	komunikační linka K-line
8	nespecifikováno
9	nespecifikováno (např. u BMW CAN převodovky)
10	J1850 PWM Bus (-)
11	Chrysler CCD- (není ODB)
12	nespecifikováno
13	nespecifikováno
14	CAN-Bus Low
15	inicializační linka L-line
16	palubní napětí (+12V)

Tabulka 2: Zapojení pinů v zásuvce dle komunikačních standardů

ISO 9141	4)kostra, 5)komunikační kostra, 7)K-Line, [15)L-line], 16)+12V
SAE J1850 VPW	2)Bus, 4)kostra, 5)komunikační kostra, 16)+12V
SAE J1850 PWM	2)Bus+, 4)kostra, 5)komunikační kostra, 10)Bus- 16)+12V
CAN BUS	4)kostra, 5)komunikační kostra, 6)CAN-High 14) CAN-Low 16)+12V

Piny 4,5 a 16 jsou povinné, automobil s ISO 9141 nemusí využívat pin 15.

1.4 Úvod do diagnostiky automobilů



Obrázek 3: Rozdělení automobilové diagnostiky

Dle získaných informací lze automobilovou diagnostiku rozdělit do 3 částí. První část OBD2 diagnostika vychází z mezinárodních standardů ISO a týká se pouze moderních automobilů. Jedná se o univerzální nástroj umožňující komunikovat maximálně s 8 ECU (obvykle jen s jednou nebo dvěma: motor, převodovka) a zjišťovat základní poruchy.

Druhou část tvoří profesionální diagnostika pro konkrétní automobily, jež používají především autorizované servisy. Výhodou je oproti univerzálnímu OBD2 nástroji především přístup ke všem dostupným řídicím jednotkám a možnost vynulování servisních intervalů. Obvykle je podporováno i přeprogramování firmware řídicí jednotky.

Třetí část je označena jako chiptuning, čímž je myšlena úplná kontrola nad daty řídicí jednotky. V této oblasti se používá speciální software a hardware pro výrobcem nepovolené modifikace, např. aktivace vypnutých funkcí ECU, úprava palivových map, změna najetých kilometrů a další.

1.4.1 OBD2 diagnostika

Jak je uvedeno v SAE J1979 a ISO 15031 OBD2 diagnostika podporuje 10 režimů:

- Režim \$1 slouží pro monitorování aktuálních hodnot používaných v hnacím ústrojí.
- Režim \$2 se používá pro „freeze frame data“, tedy data rámce, při němž nastala porucha.
- Režim \$3 zobrazí chybové kódy týkající se emisí.
- Režim \$4 smaže chybové kódy týkající se emisí.
- Režim \$5 testuje lambda sond(y), nefunkční pro protokol CAN.
- Režim \$6 testuje palubní systémy + pro CAN test lambda sond(y).
- Režim \$7 zobrazí dočasné chybové kódy týkající se emisí za poslední jízdní cyklus.
- Režim \$8 testuje palubní systémy nebo komponenty.
- Režim \$9 získá informace o vozidle (VIN, CALID, CVN).
- Režim \$A zobrazí permanentní chybové kódy.

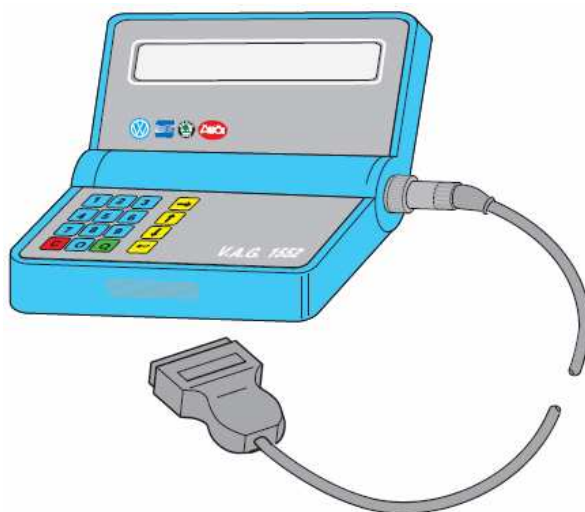
Ukázkové PID pro režim 1 dle ISO 15031-5:

PID = 04	Vypočtené zatížení motoru	// 1 byte dat	(A*100/255) %
PID = 05	Teplota chladicí kapaliny	// 1 byte dat	(A-40) °C
PID = 0C	Otáčky motoru	// 2 byty dat	((A*256)+B)/4 ot./min
PID = 0D	Rychlost vozidla	// 1 byte dat	(A) km/h

1.4.2 Diagnostika konkrétního výrobce automobilů

Tato zařízení umožňují rozsáhlejší diagnostiku než OBD2 testery. Dokáží komunikovat se všemi řídicími jednotkami, umožňují resetovat servisní intervaly a umí detekovat všechny chybové kódy DTC. OBD2 tester zjistí pouze chybové kódy týkající se emisí, a tak například závadu na palivové soustavě, způsobující špatný chod motoru, neumí odhalit.

Jelikož takováto diagnostika pracuje i s nestandardizovanými prvky, nemůže být univerzální. Pro úplnou diagnostiku vozů koncern Volkswagen se používají samostatné testery V.A.G. 1551, V.A.G. 1552, VAS 5051 a VAS 5052. Existuje levnější varianta ve formě počítačového emulátoru VAG-COM s propojovacím rozhraním, ale i zde je několik variant diagnostických rozhraní a softwarových verzí. Nejaktuálnější z nich je HEX-CAN rozhraní a VAG-COM 9. Nejnovější varianty jsou zpětně kompatibilní, tudíž postačuje mít pouze jeden přístroj. Nicméně při koupi nejnovějšího zařízení je otázkou času, do kdy výrobce bude používat kompatibilní systémy.



Obrázek 4: Diagnostický tester V.A.G. 1552, převzato z [5]

1.4.3 Chiptuning

Celá tato kapitola je vytvořena na základě informací získaných z [6]. Jelikož lokalizace dat v řídicí jednotce není standardizovaná, tak pouze výrobce ví, kde jsou uložena konkrétní data. Tyto informace se tedy mohou dostat k některým „VIP klientům“ prostřednictvím originální dokumentace. V praxi získat tento dokument obzvláště u novějších automobilů je takřka nemožné a je nutné postupovat metodou „reverse engineering“. Tato metoda je založena na diagnostických testech vstupů a výstupů řídicí jednotky (dále jen ECU), podle nichž se určuje, které paměťové prostory jsou k jakým účelům využívány.

V posledních letech výrobci automobilů podnikají kroky zaměřené proti neautorizovanému přístupu, které mají snahu zkomplikovat komunikaci externího zařízení s automobilem. Tímto se snaží, aby majitelé automobilů využívali pouze autorizované servisy. Bez profesionálního vybavení lze provádět libovolné úpravy ECU dvěma způsoby. Je možné komunikovat přes OBD2 diagnostickou zásuvku, kde je nutné u nových generací ECU překonat ochranné algoritmy. Nebo pomocí BDM rozhraní přistupovat přímo ke konkrétnímu integrovanému obvodu (EEPROM). Tato druhá varianta vyžaduje speciální zařízení (obrázek 5). Nevýhodou BDM přístupu je nutnost demontovat ECU z automobilu. Výhodou je jednak snadnější přístup k datům, ale především při přerušení během aktualizace firmware ECU, nebo při zapsání chybných dat, řídicí jednotka přes OBD2 nekomunikuje. Ke stavu, kdy ECU se chybně přeprogramuje, může dojít, když dojde k poklesu napětí nebo při silném rušení

během nahrávání firmware. Z tohoto důvodu nebývá doporučována komunikace prostřednictvím bluetooth.



Pokud uživatel může modifikovat libovolná data ECU, je možné měnit například VIN jednotky nebo stav ujetých kilometrů. Častějším důvodem ale bývá úprava výkonu motoru.

Jedny z prvních ECU měly data uloženy v EPROM. Modifikace dat vyžadoval demontáž ECU, vyjmutí čipu a jeho přeprogramování případně náhrada. Bylo možné zakoupit naprogramované čipy pro konkrétní úpravy na vozidle (například pro daný laděný výfuk), a tak tuto úpravu mohl provést i laik. Někteří výrobci umísťovali EPROM do patič, čímž výměnu usnadnili.

Další generace ECU už obsahují paměti EEPROM, jenž je možné přeprogramovat prostřednictvím diagnostické zásuvky nebo BDM rozhraní.

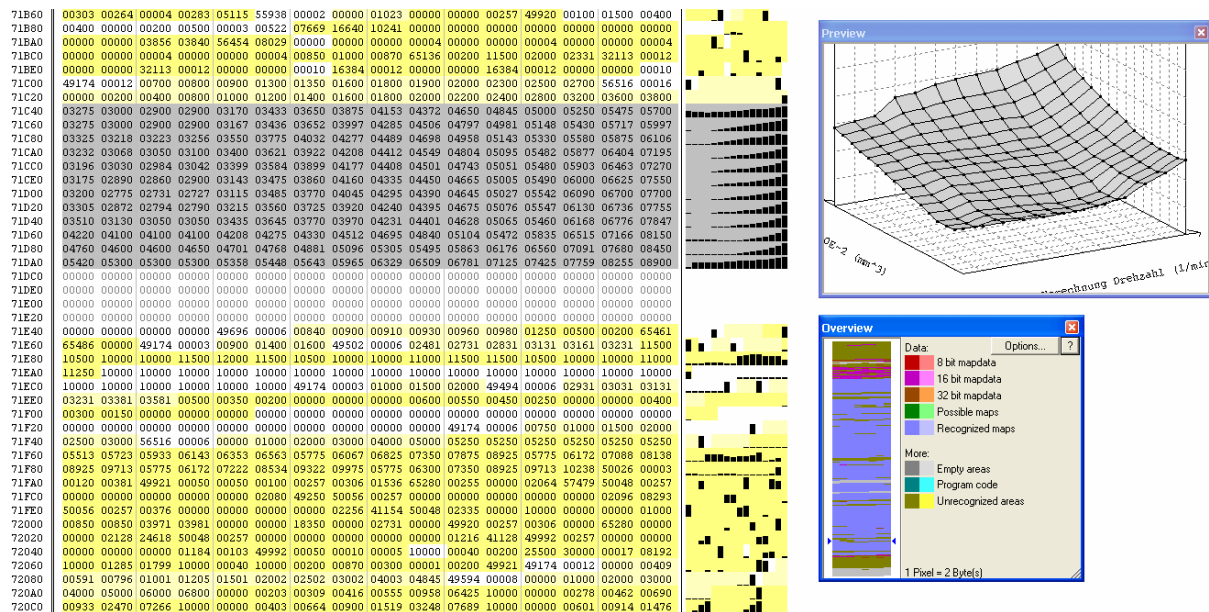
Obrázek 5: Komunikace ECU prostřednictvím BDM rozhraní, převzato z [20]

Z důvodů emisních předpisů výrobce optimalizuje chování motoru na minimální produkci škodlivých látek. Automobil produkuje minimum škodlivých látek, když se do motoru vstříkují chudé směsi se stechiometrickým poměrem složení směsi λ od 1,05 až 1,15 [6]. Motor má ale ideální spotřebu a maximální výkon při bohaté směsi s λ od 0,83 až 0,95. Pro lepší emise se do automobilů implementuje systém pro recirkulaci výfukových spalin. U tohoto systému řídicí jednotka ovládá ventil, kterým je část výfukových plynů přivedena zpět do sání. To má za následek snížení emisí, snížení teploty spalin, omezení detonačního hoření, ale i snížení výkonu [6].

Chiptuning provádí obohacení směsi, čímž má automobil větší výkon i lepší spotřebu, ale produkuje více škodlivých látek. Regulace na základě λ se v sériových vozech používá u nízkých a středních otáček, proto se maximální hodnota výkonu pouhou změnou bohatosti směsi příliš nezvyší. Výrazně se však zlepší akcelerace. Systém EGR bývá redukován, nebo v extrémních případech zcela vypnut. Pro zvýšení výkonu se modifikují určité tabulky v paměti ECU, z nichž nejdůležitější pro optimalizaci výkonu jsou palivová mapa a mapa předstihu. Palivová mapa je závislost délky vstříku (tedy množství paliva) v závislosti na zatížení motoru (poloze plynového pedálu) a aktuálních otáčkách motoru. Mapa předstihu popisuje závislost okamžiku zážehu na tlaku vzduchu v sacím potrubí (související s zatížením motoru) a aktuálních otáčkách. Při úpravách motormanagementu je vhodné vycházet z originálních dat. Základem je upravit palivovou mapu. Mírné změně palivové mapy se moderní automobil pomocí senzorů dokáže přizpůsobit, obzvláště je-li implementován senzor klepání. Pokud je však provedena výměna některých komponent automobilu nebo se požaduje vyšší výkon, je potřeba upravit i mapu předstihu. Mapa předstihu se vytváří pro konkrétní druh paliva, protože je závislá na oktanovém čísle. Vyšší oktanové číslo a odolnost proti detonačnímu hoření umožňuje provést zážeh dříve než nízkooktanové palivo. V paměti řídicí jednotky jsou dále uloženy adaptivní mapy – stabilizace volnoběhu, korekční tabulky pro studený nebo přehřátý motor. Pokud automobil funguje bez problémů, změny adaptivních

map se neprovádí. Při tvorbě vlastní mapy je doporučený krok maximálně 250 ot./min, protože je nutné respektovat volumetrickou účinnost motoru [6], která charakterizuje efektivnost plnění válců motoru.

Načtení paměti EEPROM dokáže i jednoduchý program, ale dokázat data interpretovat je mnohem náročnější. Pokud uživatel nezná strukturu rozmístění dat příslušné ECU, je nejlepší volbou použít software identifikující data v ECU. Jedny z nejpoužívanějších programů jsou WinOLS nebo ECM 2001. WinOLS nabízí více možností, ale jeho ovládání je složitější. Práci s tímto programem je možné vyzkoušet na demoverzi dostupné na [20]. Ke zkoušení programu je však nutné si opatřit data načtené ECU. Každý program pro úpravu dat řídicí jednotky dokáže pracovat pouze s takovými ECU, které má v databázi. Rozpoznání map v programu WinOLS je automatické. Program dokáže data zobrazit v textovém formátu, 2D nebo 3D grafech. V každém typu zobrazení je umožněna modifikace dat.



Obrázek 6: Ukázka modifikace palivové mapy v programu WinOLS

Modifikovanou mapu nelze ihned nahrát do řídicí jednotky. Data jsou stejně jako komunikační protokoly zabezpečeny kontrolním součtem. Bohužel na rozdíl od kontrolních součtů protokolů jsou algoritmy nestandardizované a tudíž u různých výrobců jiné. Protože takovéto algoritmy slouží jako hlavní ochrana před neautorizovaným přístupem, bývají extrémně složité. V praxi to znamená, že program WinOLS nemá tyto algoritmy implementované, ale je možné pro konkrétní řídicí jednotky algoritmus do programu přidat prostřednictvím softwarových modulů. Cena těchto algoritmů často přesahuje cenu WinOLS, tedy řádově desítky tisíc korun.

Nutno také upozornit na možné situace, kdy úprava dat nemá pozitivní účinek. Drobné chyby v parametrech se projeví nenormálním chováním motoru – klepání nebo cukavý pohyb. Vážnější zásahy do dat nebo nahrání modifikovaných dat bez provedení kontrolních součtů (neplatná data) buď nepovolí motor nastartovat, nebo během jízdy automobil přejde do nouzového režimu. Tento režim nastává také tehdy, jsou-li přerušeny komunikační vodiče. V takovémto případě pracují veškeré systémy s vlastními senzory nebo konstantními hodnotami, čímž motor není schopen podávat plný výkon. Při nejhorší úpravě dat v ECU může dojít k poškození motoru.

2 PROTOKOLY

2.1 Keyword Protokoly

2.1.1 Úvod

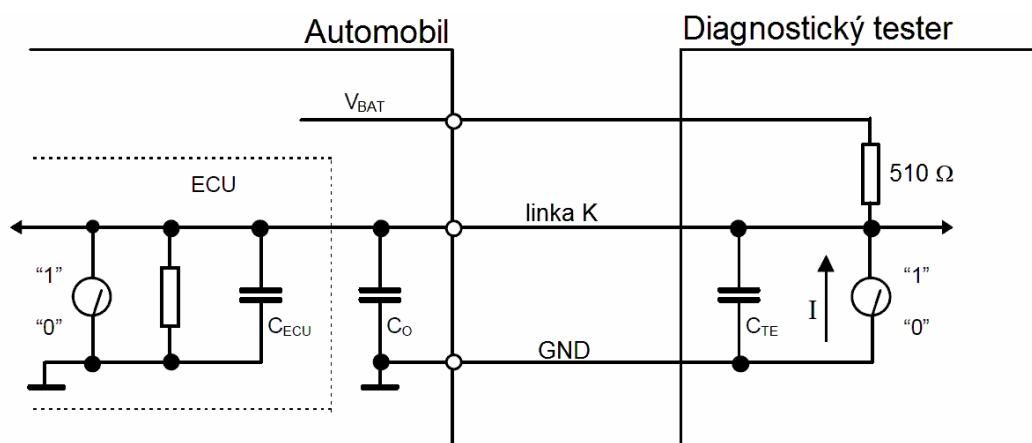
Protokoly KW1281 (ISO 9141) a modernější KWP2000 (ISO 14230), dále jen KWP, byly jedinými standardizovanými protokoly používanými pro diagnostiku automobilů vyráběných v Evropě a Asii do roku přibližně 2004. V dnešní době se tyto protokoly s totožnou fyzickou vrstvou ISO 9141-1 přestávají používat, protože je snaha o využití datových sběrnic CAN k diagnostice a ustoupit tak od speciálního vedení pouze pro diagnostiku.

2.1.2 Parametry sběrnice

Protokol umožňuje komunikaci rychlostí maximálně 10,4 kb/s. Obě varianty rovněž používají stejná vedení – linku K a případně i linku L. Hlavní rozdíl je v datovém rámci. Protokol KW1281 umožňuje maximálně 12 B zprávu, zatímco KWP2000 až 255 B zprávu. Signály jsou vztaženy k napětí baterie, používají kódování NRZ a nabývají následujících logických stavů:

- Logická „0“ je $< 30\% V_{BAT}$ na přijímací straně ($20\% V_{BAT}$ na vysílací straně)
- Logická „1“ je $> 70\% V_{BAT}$ na přijímací straně ($80\% V_{BAT}$ na vysílací straně)

Toto sériové rozhraní je realizováno jako jednovodičové se společným vysílacím a přijímacím vedením nebo jako dvouvodičové s odděleným datovým vedením (linka K) a inicializačním vedením (linka L). Diagnostika pomocí protokolu KWP funguje tak, že testovací přístroj odešle inicializační adresu na všechny řídicí jednotky. Jednotka, která rozpozná tuto adresu, odešle zpět informaci o komunikační rychlosti a formátu hlavičky. Testovací přístroj zjistí rychlost na základě doby mezi hranou impulsu a naváže komunikaci s řídicí jednotkou.



Obrázek 7: Komunikační schéma KWP (jednovodičová varianta), dle [12]

Diagnostický tester musí pracovat korektně v teplotním rozsahu od $-20\text{ }^{\circ}\text{C}$ do $+50\text{ }^{\circ}\text{C}$ a v napěťovém rozsahu od 8 V do 16 V. Zapojení testeru vyžaduje pull-up rezistor o velikosti $(510 \pm 5\%) \Omega$, aby při logické „1“ nabýval alespoň 90 % napětí akumulátoru V_{BAT} . Při logické „0“ musí mít diagnostický tester méně než 10 % V_{BAT} a maximální proud může být až 100 mA. Při vysílání dat je povolena maximální časová odchylka 30 %. Povolené napětí signálu vyslaného testerem je od -1 V do V_{BAT} . Celková kapacita diagnostického

testeru a přívodního kabelu nesmí překročit 2 nF. Výše zmíněné parametry pro diagnostický tester jsou převzaty z [12].

2.1.3 Formát rámce

Dle [13] rámec tvoří hlavička o délce 1 až 4 byty, data a kontrolní součet. Nejdůležitější byte hlavičky je formátový byte, jehož první bit je vždy „1“ a druhý definuje zda je cílová adresa fyzická („0“ – při komunikaci s jednou jednotkou) nebo „funkční“ („1“ – při komunikaci s více jednotkami). Zbývajících 6 bitů reprezentuje délku datového pole zprávy (max 63B) včetně bytu identifikující služby. Každá zpráva musí obsahovat alespoň identifikaci služby, proto její délka je minimálně 1. Pokud je všech 6 bitů nulových, délka zprávy je v přídatném bytu délky (až 255 bytů). U protokolu KW1281 není přídatný byte.

Cílová a zdrojová adresa přesně specifikuje, mezi jakými zařízeními probíhá komunikace. Adresa je vždy dlouhá 1 byte. Při diagnostice se používá adresa testeru \$F1 nebo \$6B.

V [14] jsou definovány všechny identifikace služeb SID. Jedná se o 1 bytový identifikátor, který definuje, co zpráva má provést. Následující služby musí umět každá řídicí jednotka: začátek komunikace (\$81), konec komunikace (\$82), tester přítomný (\$3F), načtení identifikace jednotky (\$1A), načtení DTC dle stavu (\$18), vymazání diagnostických informací (\$14).

Rámec nemusí obsahovat datové pole. Délka datového pole je tedy 0 až 255 bytů. Kontrolní součet je realizován jako součet všech bitů rámce modulo 256.

Formátový byte	Cílová adresa	Zdrojová adresa	Přídatný byte délky	Identifikace služby	Data LSB MSB	kontrolní součet
----------------	---------------	-----------------	---------------------	---------------------	--------------------	------------------

Obrázek 8: Rámec protokolu ISO 14230, dle [13]

2.1.4 Komunikace

Počátek komunikace diagnostiky keyword protokolem začíná inicializací. Protokol ISO 14230 podporuje pomalou (5ti baudovou) inicializaci kompatibilní s ISO 9141 a rychlou inicializaci. Po navázání spojení vysílá tester žádosti v časových intervalech 5 až 20 ms [13]. Jakmile se tester odmlčí na 25 až 50 ms, začíná řídicí jednotka odpovídat na žádosti testeru. Časové zpoždění mezi jednotlivými zprávami řídicí jednotky je 0 až 20 ms. Po odvysílání poslední zprávy řídicí jednotkou může v intervalu od 25 do 50 ms posílat zprávy další řídicí jednotka nebo od 80 do 5000 ms vyslat další žádost o data tester. Pokud tester do 5 sekund nepotřebuje komunikovat, musí alespoň vyslat udržovací zprávu „tester přítomný“, jinak bude spojení uzavřeno a při další komunikaci by muselo dojít opět k inicializaci.

2.1.5 Rychlá inicializace

Inicializace komunikace se provádí po vedení K, které v nečinném stavu je v logické „1“. Jak je uvedeno v [13], rychlá komunikace začíná impulsem logické „0“ o délce 24 až 26 ms a úrovní log „1“ takové délky, aby celkový čas byl od 49 do 51 ms. Norma ISO 14230-2 ale připouští i jiné varianty délky logické „0“. Při prvním přenosu po zapnutí má být úroveň logické „0“ alespoň 300 ms, nebo po ukončení komunikace testerem při nové inicializaci úroveň logické „0“ alespoň 55 ms.

Poté je vyslána testerem žádost o začátek komunikace složená z 5 bytů – tedy bez dat. 1. byte je formátový a nabývá hodnot \$81 pro fyzické adresování a \$C1 pro logické adresování. 2. a 3. byte jsou cílová a zdrojová adresa. 4. byte je identifikace služby (žádost o začátek komunikace) o hodnotě \$81 a 5. byte je kontrolní součet.

Řídicí jednotka odpovídá 8 byty. 1. byte nabývá hodnot \$80 pro fyzické adresování a \$C0 pro logické adresování. Následuje cílová a zdrojová adresa a přídatný byte délky o hodnotě

03. 5. byte je identifikace služby (pozitivní odpověď na začátek komunikace) o hodnotě \$C1 – identifikátor pozitivní odpovědi je hodnota o \$40 větší než identifikátor žádosti. Následují 2 byty dat obvykle \$EA a \$8F, což je keyword 2026 (\$8FEA) a 8. byte je kontrolní součet. Uvedený keyword specifikuje formát hlavičky (hlavička s cílovou a zdrojovou adresou) a normální časování.

2.1.6 Pomalá inicializace

Pokud řídicí jednotka při rychlé inicializaci nereaguje, je nutné pokusit se o pomalou inicializaci, která je rovněž označována jako 5ti baudová. Protokol KW 1281 lze inicializovat pouze pomalou inicializací, zatímco KWP 2000 podporuje oba způsoby inicializace.

Tester začíná inicializaci dle [13] vysláním 7 bitové adresy řídicí jednotky a bitu liché parity rychlostí 5 baudů po vedení K i L (některé automobily nemají vedení L připojeny v OBD konektoru). Počátek vysílání je signalizován start bitem, konec stop bitem. Řídicí jednotka musí odpovědět v časovém intervalu od 60 do 300 ms tentokrát aktuální komunikační rychlostí synchronizačním znakem \$55 neboli $(01010101)_2$ a 2 klíčovými byty. Tester rozpozná pomocí synchronizačního znaku komunikační rychlost, která je v rozmezí od 1200 b/s do 10400 b/s. Tester musí odpovědět v čase od 25 do 50 ms inverzní hodnotou druhého klíčového bytu. Po stejné časové pomlce řídicí jednotka pošle inverzní hodnotu své adresy.

2.1.7 Odpovědi řídicí jednotky na žádosti testeru

Při vysílání žádostí KWP protokolem odpovídá oslovená řídicí jednotka pozitivní nebo negativní odpovědí. V případě zaslání neplatné žádosti (špatný formát), řídicí jednotka neodpoví. Dle bytu identifikace služby patří hodnota \$7F negativní odpovědi, pozitivní odpověď má hodnotu závislou na identifikátoru žádosti v rozsahu \$CO až \$FF. U negativní odpovědi je spolu se znakem \$7F vyslán identifikátor žádosti (na níž negativně odpovídá) a byte, jehož hodnota upřesňuje chybu. Chyby mají tvar RC_xx, kde xx již zmíněný byte negativní odpovědi.

Příklad negativních odpovědí:

- RC_11 = služba není podporována
- RC_12 = subfunkce není podporována
- RC_21 = zaneprázdněn, opakuj žádost
- RC_22 = chybná sekvence žádosti
- RC_23 = rutina není hotová
- RC_78 = zpráva přijatá správně, ale odpověď nevyřešena
- RC_10 = všeobecné odmítnutí

2.2 CAN

2.2.1 Základní informace

CAN (*Controller Area Network*) je sériová datová sběrnice vyvinutá firmou Bosch. Její vývoj začal roku 1983 a oficiálně byla předvedena roku 1986. První automobil se sběrnici CAN byl uveden v roce 1992 firmou Mercedes-Benz. Využití CAN pro diagnostiku bylo však později. Cílem protokolu CAN bylo vytvořit sběrnici, která by především vedla k úspoře kabeláže a zabezpečení přenosu informací mezi snímačími, řídicími a výkonovými prvky v automobilech. Všechny neuskutečněné žádosti o přenos jsou zpracovány podle jejich priority v systému a to i v případě nedostatečného přenosu sběrnice. Protokol CAN nestanovuje vlastní úroveň a fyzikální medium. Pro požadovanou aplikaci může tedy sběrnice pracovat s napětím, proudem nebo světlem. Mezi hlavní přednosti protokolu CAN patří nízká latence, kontrola nad chybami, nastavování priority a neustálé monitorování.

high speed CAN – dvou vodičová varianta, rychlost až 1 Mb/s

low speed CAN – dvou vodičová varianta, rychlost do 125 kb/s

single wire CAN – jednovodičová varianta

2.2.2 Technické parametry

Sběrnici tvoří dva vodiče označené CAN_H a CAN_L. Maximální rychlost přenosu je 1 Mbit/s pro délku sběrnice maximálně 40 m. Při 130 m klesá maximální rychlost na 500 kbit/s a pro délku 3,3 km je rychlost pouhých 20 kbit/s. Protokol garantuje maximální čas přenosu zprávy s vysokou prioritou 134 us při rychlosti 1 Mbit/s. Pravděpodobnost výskytu chyby v úrovni je $4,7 \cdot 10^{-11}$. Protokol klade značné nároky na časování - povolené časové zpoždění je pouze $\pm 0,15\%$ (tedy ± 3 ns při 500 kbps) na rozdíl od $\pm 1,7\%$ u ISO 14230 ($\pm 1,6$ us při 10,4 kbps) dle [14] a [17].

2.2.3 Komunikace po sběrnici

Ke sběrnici jsou připojeny jednotlivé uzly, jejichž maximální počet je dán použitým budičem sběrnice. Komunikační síť CAN může pracovat v režimu, kdy je více nadřazených uzlů, nebo v režimu, kdy je jeden uzel nadřazený a více podřazených uzlů. Nejdůležitější částí vysílané zprávy je její identifikátor, který definuje obsah zprávy a její prioritu. Případné kolize jsou řešeny tak, že uzly při detekci vysílání jiné zprávy s vyšší prioritou se postupně odmlčují, až zbude jediný vysílající uzel. Odmlčování uzlů se provádí během vysílání identifikátoru, kdy uzel vysílající recesivní stav (logická 1) detekuje na sběrnici stav dominantní (logická 0).

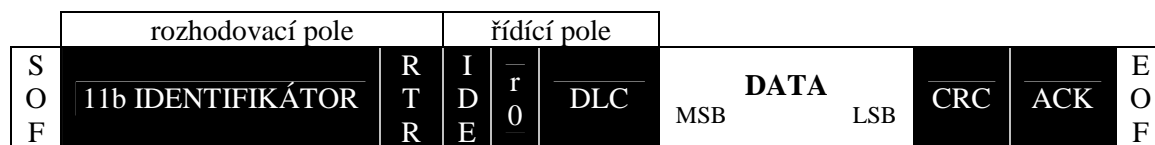
Dominantní stav nastává tehdy, vysílá-li alespoň jeden uzel logickou 0, čímž vznikne na vodičích sběrnice rozdíl potenciálů 2 V (CAN_H = 3,5 V a CAN_L = 1,5 V). Recesivní stav nastane tehdy, je-li na obou vodičích sběrnice stejný potenciál (CAN_H = CAN_L = 2,5 V). Při přenosu 5 a více bitů stejné hodnoty se používá kódování bit stuffing, které vkládá bit opačné hodnoty. Tím je zajištěna jednoznačná identifikace chybového rámce, jenž je tvořen 6ti bity stejné úrovně.

V protokolu CAN jsou definovány čtyři typy rámců:

- datový rámeček (*DATA FRAME*)
- žádost o data (*REMOTE FRAME*)
- chybový rámeček (*ERROR FRAME*)
- rámeček přeplnění (*OVERLOAD FRAME*)

2.2.4 Formát rámce

Existují dva rámcové formáty CAN 2.0A (standardní formát) s 11ti bitovým identifikátorem a CAN 2.0B (rozšířený formát) s 29ti bitovým identifikátorem. Oba formáty jsou navzájem kompatibilní. Maximální délka zprávy je 130b pro CAN 2.0A a 150b pro CAN 2.0B. Identifikátor definuje obsah zprávy, například otáčky motoru. 11 bitový identifikátor umožňuje 2048 různých zpráv, což obvykle v automobilové diagnostice postačuje.



Obrázek 9: Standardní formát datového rámce CAN 2.0A dle [7]



Obrázek 10: Rozšířený formát datového rámce CAN 2.0B dle [7]

Datový rámec (DATA FRAME)

Datový rámec zabezpečuje informace vysílajícího uzlu, které jsou poté přenášeny všem ostatním uzlům na sběrnici. Tento rámec obsahuje následující části:

START OF FRAME – úvodní bit s dominantní hodnotou

ROZHODOVACÍ POLE – se skládá z identifikátoru a bitu RTR (*Remote Transmission Request*), který specifikuje, zda je informace datový rámec nebo žádost o vysílání. Toto pole určuje prioritu vysílané zprávy. Uzel během vysílání neustále monitoruje stav na sběrnici. Pokud uzel zjistí, že vyslal recesivní bit a na sběrnici je bit dominantní, okamžitě přestává vysílat. Tímto způsobem je zabezpečeno, aby přístup ke sběrnici dostal ten, jehož zpráva má nejvyšší prioritu a aby při nárůstu zatížení sběrnice nedošlo ke snížení přenosového výkonu sítě.

ŘÍDÍCÍ POLE – řídící pole obsahuje bit IDE (*IDentifier Expression*) pro rozlišení základního a rozšířeného formátu rámce, rezervní bit(y) a 4 bity DLC (*Data Length Code*) určující počet bytů datového pole. Poměrně malá délka tohoto pole je proto, že v protokolu CAN je důležité především zabezpečení rychlého přenosu zpráv s vysokou prioritou. Větší bloky dat je nutno segmentovat v aplikační úrovni. Všechna data na sběrnici jsou dostupná pro všechny uzly současně.

DATOVÉ POLE – datové pole má velikost 0 až 8 bytů

CRC POLE (*Cyclic Redundancy Code*) – obsahuje 15 kontrolních bitů cyklického redundantního kódu zahrnující všechna předcházející pole. CRC pole je ohraničeno recesivním bitem ERC (*End Of Crc*)

ACK POLE – Potvrzovací část obsahuje bity ACK SLOT a ACK DELIMITER. Vysílač vysílá bit ACK SLOT jako recesivní. Pokud alespoň jeden uzel přijal zprávu bez chyby, přepíše tento bit na dominantní, čímž oznámí vysílači potvrzení příjmu. ACK DELIMITER je vždy recesivní.

END OF FRAME – konec rámce se skládá z nejméně sedmi recesivních bitů, za nimiž následují 3 bity pro uklidnění všech vysílačů. V této době mohou přijímací uzly informovat vysílací uzel o chybách přenosu.

Žádost o data (REMOTE FRAME)

Žádost o data má podobný formát jako datový rámec. Neobsahuje datové pole a bit RTR je recesivní, zatímco v datovém rámci je dominantní. Uzel tímto rámcem žádá některý jiný uzel na síti o vyslání datového rámce se shodným identifikátorem, jaký je v žádosti.

Chybový rámec (ERROR FRAME)

Chybový rámec sestává z polí ERROR FLAG a ERROR DELIMITER. Uzel, který zjistí chybu v řetězci přijímaných bitů, začne vysílat 6 dominantních bitů, čímž poruší strukturu rámce. Ostatní uzly začnou též vysílat 6 dominantních bitů. Celková délka ERROR FLAG tak může být 6 až 12 bitů. Za nimi následuje pole ERROR DELIMITER s 8 recesivními bity.

Rámec přeplnění (OVERLOAD FRAME)

Rámec přeplnění má obdobnou podobu, jako chybový rámec. Uzel vyše tento rámec tehdy, potřebuje-li určitý čas na zpracování předchozí zprávy.

2.2.5 Použití v automobilech

Výše zmíněné informace o protokolu CAN reprezentují fyzickou a linkovou vrstvu referenčního modelu ISO/OSI. V automobilech se používají vyšší vrstvy protokolu CAN s 11b nebo s 29b identifikátorem, například ISO/TP (ISO 15765-2) nebo TP2.0. Komunikace probíhá obdobně jako u ISO 14230 (kap. 2.1.4). Při inicializaci je zvolena komunikační rychlost, obvykle 250 nebo 500 kb/s. Tester osloví všechny jednotky pomocí funkční žádosti. Funkční adresa OBD2 zařízení je \$33, adresa testeru je \$F1 nebo \$6B. Příslušná řídicí jednotka s konkrétní fyzickou adresou odpoví. Cílové a zdrojové adresy jsou součástí identifikátorů. U 11b ID tvoří páry žádost-odpověď. První pár je vždy řídicí jednotka motoru ECM s identifikátory \$7E0-\$7E8 a druhý pár řídicí jednotka převodovky TCM \$7E1-\$7E9.

U 29b identifikátorů je struktura naznačena v tabulce 4. Při komunikaci s testerem dochází ke 3 typům zpráv: funkční žádost \$(18 DB 33 F1), fyzickou žádost testeru k ECUx \$(18 DA xx F1) nebo fyzickou odpověď ECUx \$(18 DA F1 xx).

Zprávy s nižší hodnotou identifikátoru mají vyšší prioritu přenosu. V případě požadavku na vyslání datového rámce a rámce žádosti o data se stejným identifikátorem, má větší prioritu datový rámec.

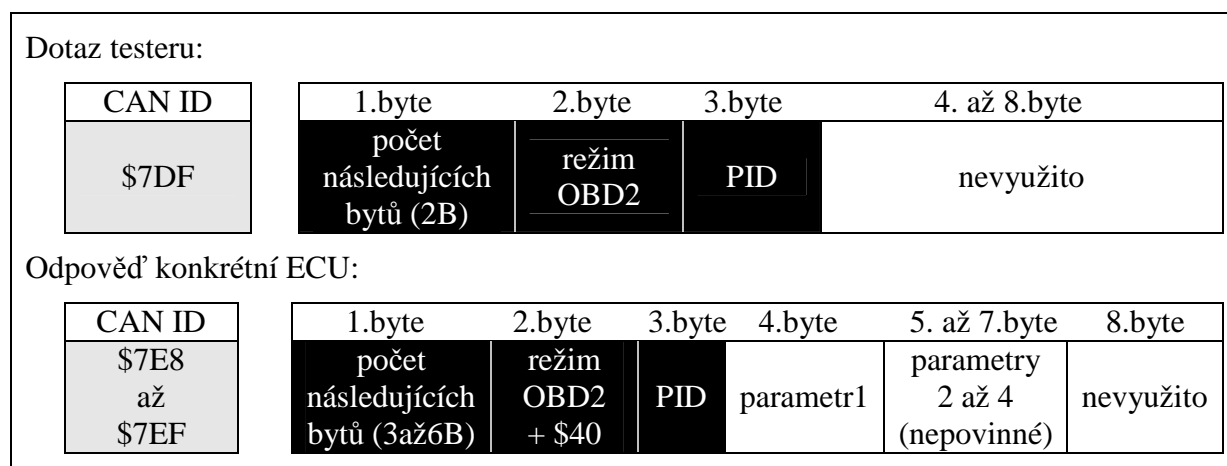
Tabulka 3: 11b CAN identifikátory dle [17]

CAN ID	Popis
\$7DF	Funkční žádost zasláná testerem
\$7E0	Fyzická žádost testeru k ECU#1 (ECM)
\$7E8	Fyzická odpověď ECU#1 (ECM)
\$7E1	Fyzická žádost testeru k ECU#2 (TCM)
\$7E9	Fyzická odpověď ECU#2 (TCM)
\$7E2	Fyzická žádost testeru k ECU#3 (obvykle ABS)
\$7EA	Fyzická odpověď ECU#3
...	
\$7E7	Fyzická žádost testeru k ECU#8
\$7EF	Fyzická odpověď ECU#8

Tabulka 4: 29b CAN identifikátory dle [17]

bity CAN ID	28	24	23	16	15	8	7	0
Funkční CAN ID	\$18		\$DB		Cílová adresa		Zdrojová adresa	
Fyzické CAN ID	\$18		\$DA		Cílová adresa		Zdrojová adresa	

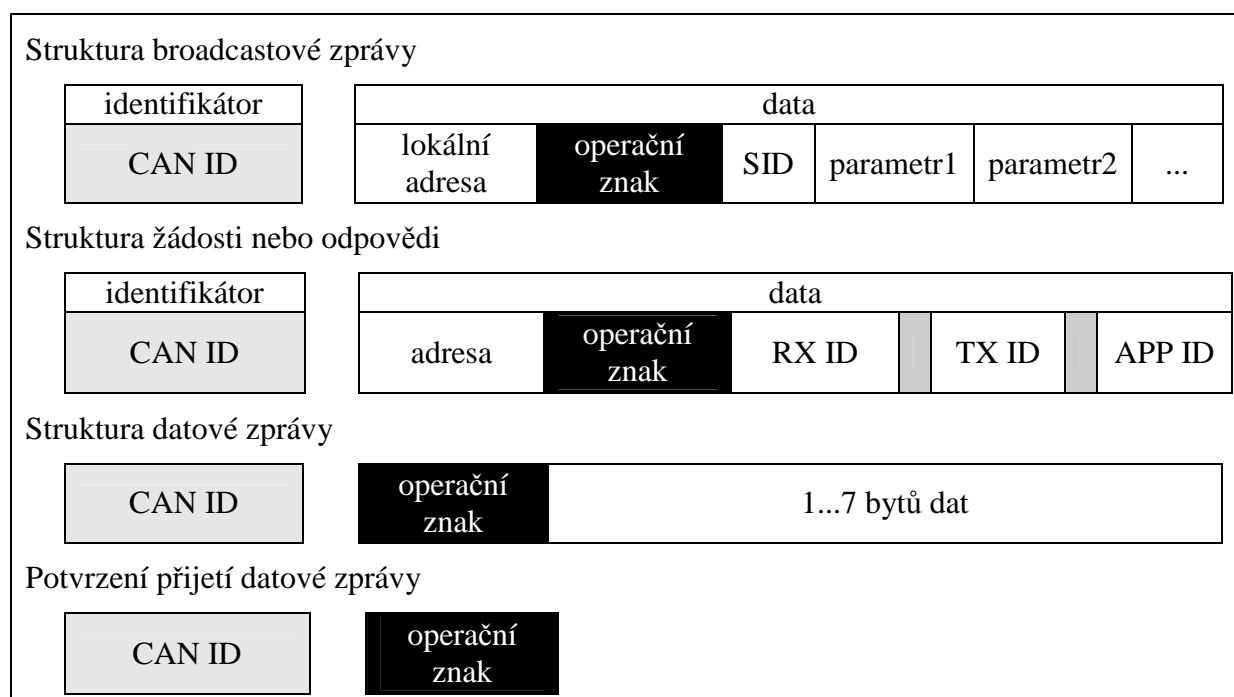
Reprezentace dat v datovém rámci CAN při komunikaci dle standardu OBD2:



Obrázek 11: Struktura dat při komunikaci po CAN dle OBD2

2.2.6 Vyšší vrstvy CAN - TP2.0

Standard ISO 15765 definuje fyzickou, linkovou, síťovou a aplikační vrstvu referenčního modelu ISO/OSI. Existují však i protokoly specifikující transportní vrstvu. TP2.0 (Transport Protocol) je protokol používaný především automobilkou Volkswagen, který umožňuje stávající diagnostické protokoly (KW-1281 a KWP2000) provozovat na rozhraní CAN2.0A. Předchozí verze TP1.4 a TP1.8 pracují na rozhraní ISO 9141. Na rozdíl od ISO/TP používá dynamické přidělování identifikátorů pro kanál. Protokol umožňuje adresování unicast, multicast, broadcast a používat gateway díky lokálním adresám.



Obrázek 12: Reprezentace dat CAN2.0A v protokolu TP2.0 dle [18]

Důležitou součástí každé zprávy protokolu TP2.0 je kromě 11b identifikátoru CAN ID je operační znak. Ten určuje, co data reprezentují. U broadcastu žádost o data má operační znak \$23, odpověď \$24 dle [18]. Unicastovou žádost reprezentuje operační znak \$C0, na kterou řídicí jednotka odpovídá \$D0 (pozitivně) nebo \$D6-\$D8 (negativně). Unicastová žádost obsahuje APP ID – identifikátor aplikace, který pro diagnostiku nabývá hodnoty \$01.

2.3 FlexRay

2.3.1 Historie

Vývoj protokolu FlexRay počal rokem 2000, kdy firmy BMW a DaimlerChrysler se rozhodly vytvořit nový protokol nahrazující, v té době již v některých směrech nedostačující, CAN. Specifikace protokolu FlexRay V2.1 vznikla roku 2005 a o rok později firma BMW aplikovala protokol do rychlého adaptivního systému podvozku u automobilů řady X5 [1]. Protokol tvoří 2 kanály, které při jednosměrném přenosu mají maximální přenosová rychlost až 20Mb/s. Pro obousměrnou komunikaci se jeden kanál použije pro vysílání a druhý pro příjem.

2.3.2 Komunikace

Výměna informací mezi uzly je založena na časovém multiplexu TDMA a komunikačních cyklech. Systém FlexRay tvoří sběrnice a několik řídicích jednotek, které mají nezávislé hodinové signály. Všechny hodinové signály mohou mít maximálně odchylku 0,15 % referenčního hodinového signálu, tedy každých 300 cyklů může vzniknout na jednotkách maximálně odchylka 1 cyklus.

V jeden časový okamžik může vysílat pouze jedna řídicí jednotka. V případě, kdy se po sběrnici nekomunikuje, je na sběrnici logická 1. Komunikace začíná logickou 0. Každý bit je vysíláný po dobu 8 cyklů, přičemž přijímač první 3 vzorky ignoruje a zaznamená hodnotu zbylých vzorků. Jako vstupní signál vyhodnotí nejčastěji se vyskytující logickou úroveň z 5ti vzorků. Každých 88 cyklů se provádí synchronizace hodinových signálů, během níž je vysláno 11 bitů, 8 bitů dat a 3 bity řídicí. Prvních 8 cyklů slouží právě k synchronizaci.

Z důvodu, že tento nový protokol se dosud pro diagnostiku automobilů nepoužívá, se dále tato práce protokolem FlexRay nebude zabývat.

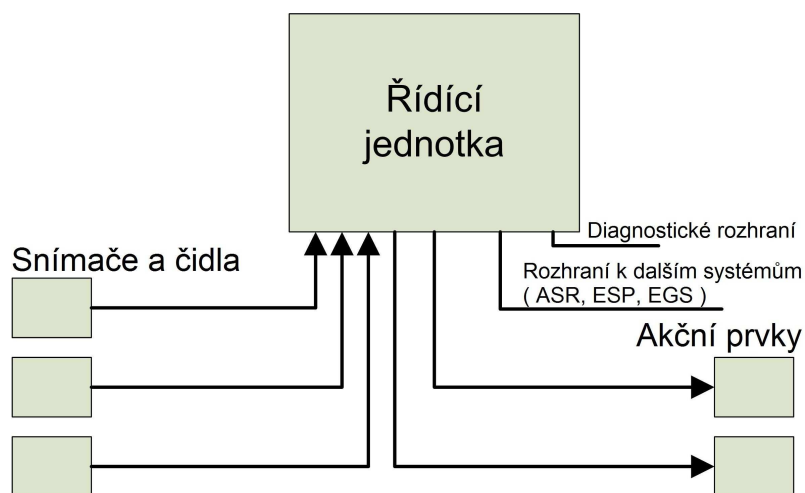
3 ŘÍDÍCÍ JEDNOTKY

3.1 Obecně o ECU

Zkratka ECU znamená *Electronic Control Unit*, tedy řídicí jednotku. Také se lze setkat s označením ECM *Engine Control Module*, řídicí jednotkou motoru – nejdůležitější ECU v automobilu. Klíčovým prvkem každé ECU je dostatečně výkonný mikroprocesor, který musí vyhodnocovat informace tisícekrát za sekundu. Frekvence mikroprocesoru bývá řádově desítky MHz, například 16ti bitové mikrokontroléry HCS12X, používané firmou BMW, pracují na frekvenci 40 MHz. Požadavky na velikost paměti neustále narůstají. Zatímco v roce 1980 u modelu Peugeot CX byla velikost programu 1,1 kB (dle [1]), tak v roce 2000 u modelu Peugeot 607 program zabíral 2 MB paměti.

Procesy, které ECU vykonává se dělí do následujících skupin:

- řízení vstřikování paliva
- řízení volnoběžných otáček
- kontrola emisí
- autodiagnostiku a správu závad



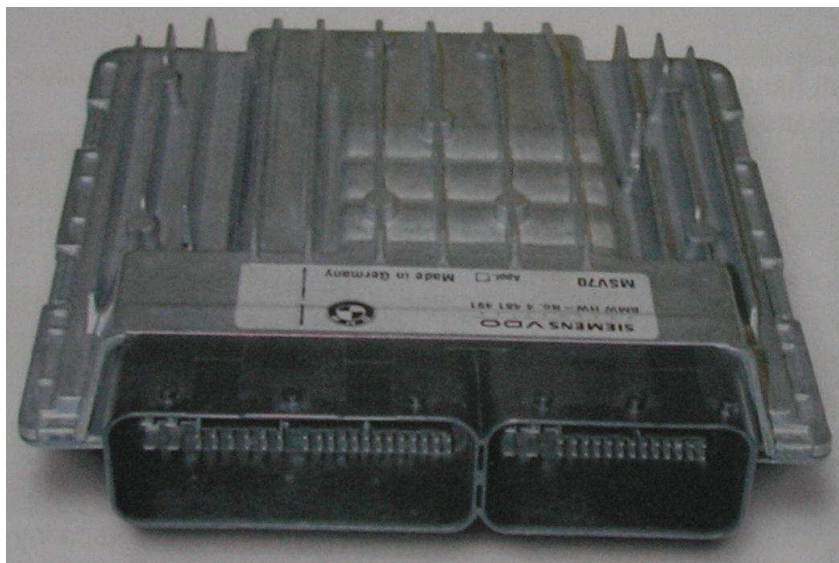
Obrázek 13: Hlavní části elektronického řízení motoru

Řídicí jednotka získává informace z externích snímačů a čidel, které převádějí fyzikální veličiny na elektrické signály a informují jednotku o provozních podmínkách. Vstupní signály řídicí jednotka omezuje na přípustnou úroveň a případně kontroluje jejich věrohodnost. Kontrola věrohodnosti probíhá u důležitých vstupních signálů různě. Například u snímače pedálu akcelerace jsou 2 odporové snímače, jejichž hodnoty jsou vzájemně porovnávány. Jiný způsob kontroly věrohodnosti dat je u snímače otáček klikové hřídele, jehož hodnoty jsou porovnávány se snímačem otáček vačkové hřídele. V případě zjištění poruchy snímače řídicí jednotka použije, pokud je to možné, náhradní signál nebo „nouzovou“ konstantou (například při poruše atmosférického snímače tlaku).

Řídicí jednotka na základě získaných dat reguluje akční prvky tak, aby měl motor dostatečný výkon a točivý moment při současné minimalizaci spotřeby paliva a produkovaných škodlivých látek (NO_x, CO, HC a dalších). Řídicí jednotka umožňuje výměnu dat s jinými elektronickými systémy jako jsou: protiprokluzová regulace (ASR), elektronický program stability (ESP) a elektronické řízení převodovky (EGS). Řídicí jednotka má integrované diagnostické rozhraní, které veškeré operace monitoruje a v případě chyby provede příslušná opatření.

3.2 Siemens DDE a DME

Firma Siemens vyrábí pro zážehové motory řídicí jednotky DME (*Digital Motor Electronic*) a pro vznětové motory jednotky DDE (*Digital Diesel Electronic*). Digitální elektroniku vznětových motorů DDE používá mimo jiné firma BMW. Řídicí jednotka DDE se skládá z několika zásuvných modulů, které jsou obvykle nakombinované ve dvou kompaktních konektorech. Řídicí jednotka vyžaduje ke své činnosti napájecí napětí 6 až 16 V. Jelikož samotná jednotka není vodotěsná, musí být umístěna ve speciálním E-boxu. Jednotka je vybavena snímačem okolního tlaku a teploty. Pokud teplota řídicí jednotky stoupne nad kritickou mez, pak je redukováno několik vstříků za účelem ochlazení výkonových stupňů.



Obrázek 14: Řídicí jednotka Siemens DDE, převzato z [11]

Řídicí jednotka je odpovědná především dle [11] za řízení zapalování, řízení teploty motoru, řízení elektrického čerpadla chladicí kapaliny, regulace klepání motoru, regulace na základě signálů od lambda sond, řízení odvětrávání palivové nádrže, ovládání klimatizace, řízení sacího potrubí, ovládání palivového čerpadla, řízení alternátoru, odvětrání klikové skříňe, elektrické monitorování stavu a hladiny oleje, tempomat a autodiagnostiku.

Automobily BMW mají několik sběrnic, například dle [10] PT-CAN (hnací ústrojí), SMG-CAN (nastavení volnoběhu), DK-CAN (škrticí klapky), K-CAN (periferie karoserie vozu), BSD BUS (alternátor a IBS) a rozhraní do CAS (přístupový systém vozu).

Snímače a čidla poskytují vstupní signály řídicí jednotce. Ta na jejich základě vypočítá potřebné ovládací signály pro regulátory, výsledek porovná s výpočtovými modely a charakteristickými mapami uloženými v paměti řídicí jednotky.

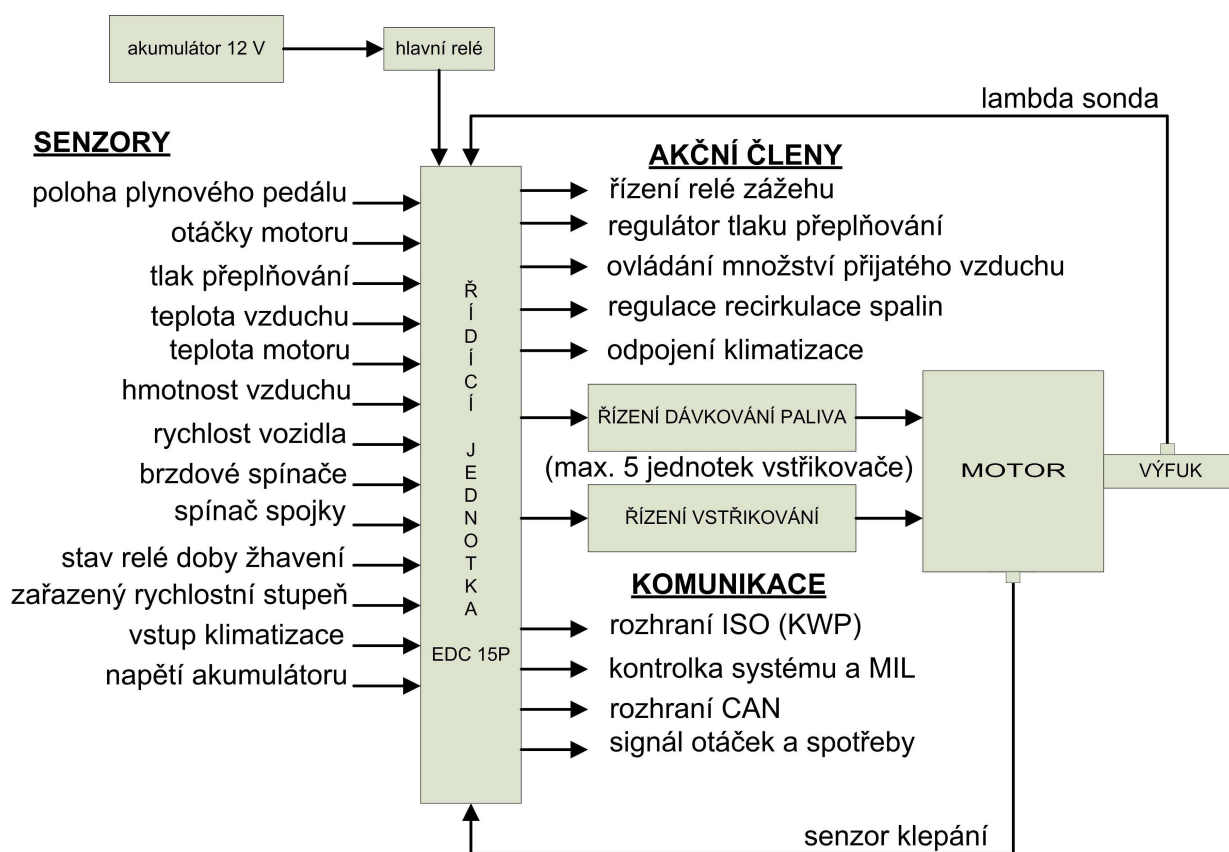
3.3 Bosch EDC

3.3.1 Úvod

EDC (*Electronic Diesel Control*) je řízení vstřikování paliva pro moderní vznětové motory. U systému EDC nemá řidič přímý vliv na množství vstřikovaného paliva, čímž se odlišuje od vozidel s konvenčními mechanicky řízenými vstřikovacími čerpadly. Systém EDC kromě zpracovávání povelů od řidiče (poloha pedálu akcelerace) upravuje hodnoty výstupu dle dalších veličin (teplota, tlak) a navíc reguluje plynulý chod motoru, používá aktivní tlumení škubání a případně omezuje rychlost vozidla.

Mezi nejnovější jednotky EDC patří EDC16 a EDC16+. U EDC16 jsou poprvé použity systémy řízené momentem [4]. Řídicí jednotka počítá potřebný točivý moment motoru tak, aby splnila požadavek řidiče (poloha pedálu akcelerace) a dodala potřebnou energii pro další komponenty, jako jsou klimatizace, alternátor a další. Koncern VAG ve svých současných vozech používá EDC16 (např. Škoda Octavia), tak EDC 15P (Škoda Fabia) a další. EDC16 používá mikroprocesor MPC556, paměť EPROM AM29BL802 a EEPROM ST95320. Jednotku EDC16+ řídí MPC562 (BMW) nebo MPC564 (VAG).

3.3.2 Blokové schéma EDC 15P

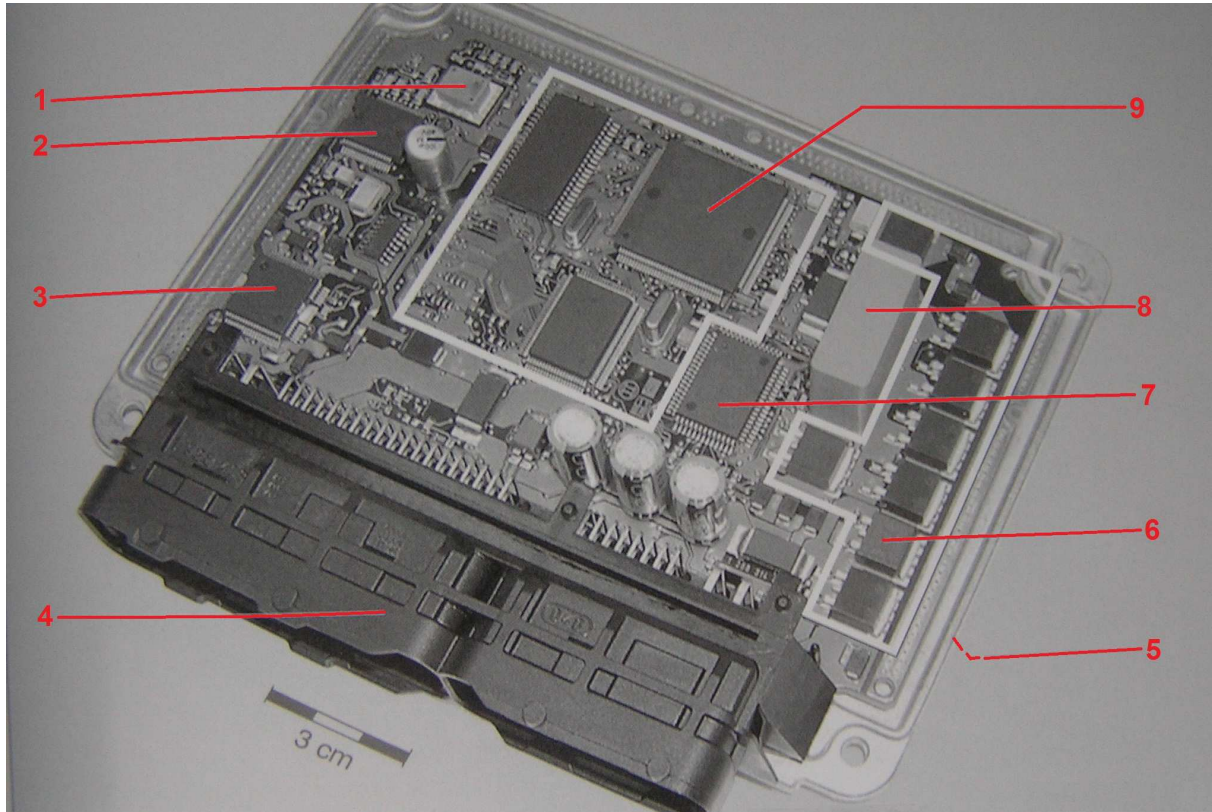


Obrázek 15: Schéma řídicí jednotky EDC 15P, dle [4]

Na obrázku 15 je znázorněno blokové schéma řídicí jednotky motoru EDC 15P. Schéma tvoří 4 hlavní části: senzory, řídicí jednotka, akční členy a komunikační rozhraní. Senzory poskytují řídicí jednotce informace o provozních podmínkách. Řídicí jednotka na základě získaných informací řídí akční členy. Velmi důležitou roli hraje především řízení dávkování paliva a řízení vstřikování, jejichž optimální funkce je klíčová pro výkon motoru a nízké

emise. Proto je řízení provedeno se zpětnou vazbou. Senzor klepání pomáhá nastavit optimální vstřikování a lambda sonda zjišťuje bohatost směsi, čímž poskytuje zpětnou vazbu pro řízení dávkování paliva. Řídicí jednotku lze diagnostikovat pomocí protokolů KWP nebo CAN.

3.3.3 Konstrukce řídicí jednotky EDC



Obrázek 16: Konstrukce řídicí jednotky EDC, převzato z [4]

Deska plošných spojů řídicí jednotky je umístěna v kovovém pouzdru, aby byla ochráněna proti vlhkosti, působení chemických kapalin (olej, palivo) a především zajištěna elektromagnetická kompatibilita. Řídicí jednotka nesmí být ovlivněna rušením, ale také nesmí rušení způsobovat. Základními prvky řídicí jednotky EDC jsou dle obrázku 16:

1) snímač atmosférického tlaku, **2)** spínaný zdroj se stabilizací napětí, **3)** koncový stupeň pro malý výkon, **4)** přípojovací konektor, **5)** rozhraní CAN a všeobecné spínací obvody (na spodní straně desky), **6)** koncové stupně pro vysoký výkon, **7)** ASIC pro aktivaci koncových stupňů, **8)** měnič napětí, **9)** mikrokontroler.

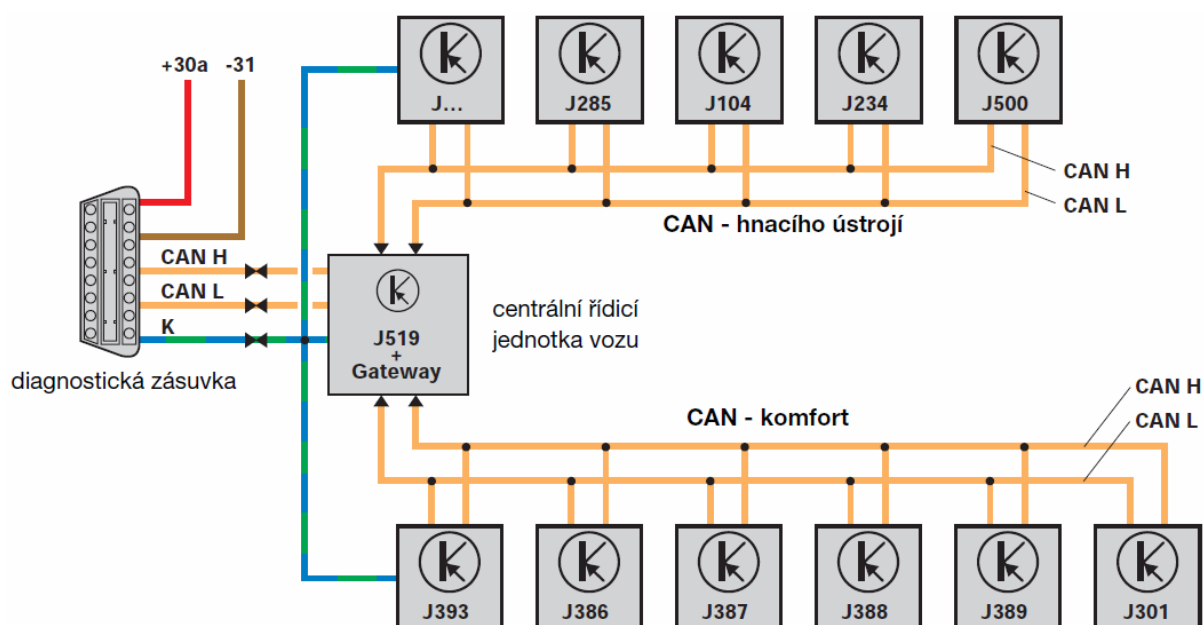
3.3.4 Mikrokontroler MPC564

S ohledem na požadovanou přesnost a dynamiku vznětových motorů je potřebný výkonný mikrokontroler pro řídicí jednotku. MPC564 je 32bitový mikrokontroler s jádrem PowerPC pracujícím na frekvenci 40 MHz od firmy Freescale Semiconductors. Obsahuje velké množství periférií. Pro výpočty s plovoucí desetinnou čárkou slouží 64 bitová jednotka FPU. Na čipu je 512 kB paměti typu flash, což v dnešní době není dostatečné (viz. kapitola 3.1) a proto se k mikrokontroleru přidává externí paměť. Pro zpracování dat slouží paměť RAM o velikosti 32 kB. Pro získávání dat z analogových snímačů je možno použít dva 16-kanálové A/D převodníky. Na čipu je rozhraní NEXUS a JTAG IEEE1194.1, BDM, tři CAN moduly, sériové moduly SCI a QSPI a další.

4 DIAGNOSTIKA

4.1 Vnitřní uspořádání elektroniky vozu

Jedno z typických způsobů zapojení řídicích jednotek moderních automobilů vyrobených v Evropě je uvedeno na obrázku 17. Jedná se o systém, který lze diagnostikovat pomocí dvou protokolů –KWP a CAN. Dle [5] je automobil Škoda Fabia vybaven sběrnici CAN hnacího ústrojí s přenosovou rychlostí 500 kb/s (*High Speed CAN*) a prioritou 1 a sběrnici CAN komfortu s přenosovou rychlostí 100 kb/s (*Low Speed CAN*) a prioritou 2. Sběrnice z důvodu odlišné komunikační rychlosti nemohou být propojeny přímo, nýbrž přes rozhraní nazvané gateway. Gateway dále převádí data z vedení CAN-BUS na vedení K a zavádí 3. sběrnici CAN sloužící pouze pro diagnostiku. Gateway tedy funguje jako firewall – zabraňuje kolizím například při zkratu vedení CAN v diagnostické zásuvce.



Obrázek 17: Zapojení řídicích jednotek v automobilu Škoda Fabia, převzato z [5]

Centrální řídicí jednotka vozu J519 umožňuje veškerou diagnostiku automobilu. CAN hnacího ústrojí zprostředkovává komunikaci mezi řídicími jednotkami: J285 panel přístrojů, J104 ABS, J234 airbagy, J500 servořízení a další jako například J446 řídicí jednotka pomoci při parkování. CAN komfortu komunikuje s jednotkami: J393 centrální řídicí jednotka komfortní elektroniky, J386 dveře řidiče, J387 dveře spolujezdce, J388 levé zadní dveře, J389 pravé zadní dveře, J301 klimatizace.

4.2 Detekce chyb

ECU pro řízení automobilu nepřetržitě čte hodnoty snímačů. Přečtené hodnoty jsou porovnány s hodnotami uloženými v paměti řídicí jednotky, která na jejich základě generuje odpovídající reakci pro akční členy. V případě, že hodnota přečtená ze snímače se rovněž nachází v paměti řídicí jednotky, pak se jedná o normální rozsah snímače. Pokud však přečtená hodnota v paměti řídicí jednotky není, jedná se o abnormální rozsah. ECU neví, jak má regulátory řídit, a proto čtení hodnoty snímače opakuje. Po vyčerpání limitu pro počet opakování je snímač označen jako vadný a kód chyby DTC zaznamenán na příslušnou adresu paměti typu RAM či EEPROM řídicí jednotky. V případě, že ECU ukládá chyby do paměti

typu RAM se jedná o způsob KAM (*Keep Alive Memory*), který zajišťuje napájení paměti z akumulátoru při vypnutí zapalování. Po opravení závady je tedy možné chybu smazat odpojením akumulátoru (jednodušeji vyjmutím pojistky) na přibližně 30 sekund [2]. U paměti typu EEPROM musíme pro odstranění chyby z paměti použít vždy diagnostický přístroj. Na základě typu chyby řídicí jednotka ovládá systém podle jiných kritérií nebo zablokuje některé funkce, dokud závada nebude odstraněna a chyba z paměti smazána.

4.3 Chyby DTC

Chyby DTC jsou identifikovány 5 znaky. Prvním znakem je písmeno označující o jakou oblast automobilu se jedná. V OBD2 se vyskytují 4 oblasti: karoserie, podvozek, přenos síly a nedefinované (rezerva pro budoucí použití). Druhým znakem je číslice definující typ kódu, kde se často používá 0, tedy chybový kód dle SAE. Následující 3 znaky jsou číslice, které blíže specifikují konkrétní závadu. Každý automobil nemusí mít implementovány všechny DTC, přičemž může pracovat s některými vlastními kódy.

Tabulka 5: Struktura standardu chybových kódů v OBD2 dle [2]

pořadí	možnosti	význam
1.	B C P U	Body (karoserie) Chassis (podvozek) Powertrain (přenos síly) Undefined (nedefinované)
2.	0 1 2 3	chybový kód dle SAE chybový kód dle výrobce chybový kód dle výrobce reservovaný chybový kód
3.	1 2 3 4 5 6 7	palivový a vzduchový systém palivový a vzduchový systém systém vstřikování výfukový systém regulace otáček a volnoběhu počítačové a výstupní signály převodovka
4. a 5.	01 10 99	upřesnění systémové komponenty

Zjištění závady lze určit několika metodami:

1. Pomocí blikajícího kódu diagnostické kontrolky nebo displeje s číselným kódem, kterým je vozidlo vybaveno.
2. Pomocí externí blikající kontrolky.
3. Připojením servisního testeru nebo skenovacího nástroje k diagnostickému konektoru.

5 NÁVRH DIAGNOSTICKÉHO ROZHRAŇÍ

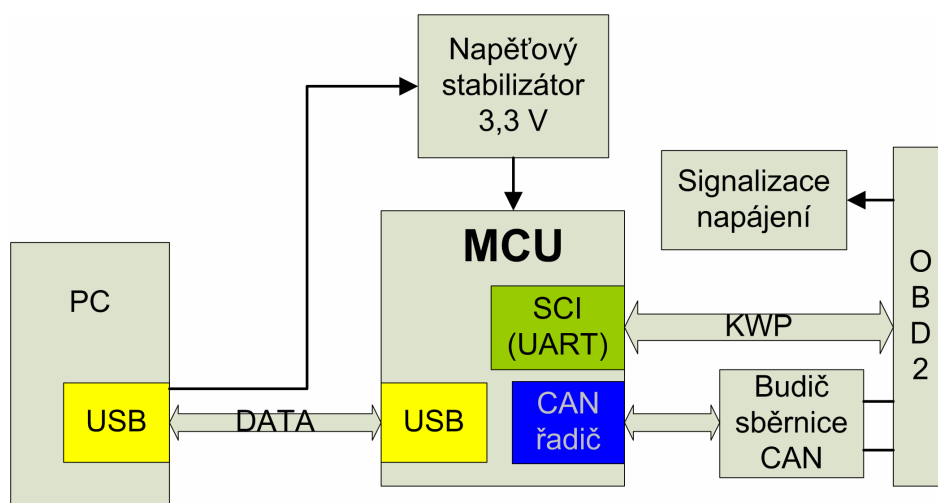
5.1 Úvod

Důvodem vytvoření diagnostického kabelu je snaha nahradit profesionální systémy autorizovaných servisů, které stojí řádově několik set tisíc korun, a tak si je nemohou dovolit menší firmy a opraváři. Diagnostický kabel slouží pro komunikaci automobilu s počítačem. Konvertuje rozhraní používaná v automobilech na rozhraní počítačové (RS232, USB, bluetooth).

Jedny z prvních diagnostických kabelů pro automobily s řídicími jednotkami řady EDC převáděly protokol KW1281 na rozhraní RS232. Jelikož se jedná v obou případech o asynchronní sériový přenos, bylo možné počítač s automobilem propojit přímo, případně pouze s oddělením signálů pomocí optočlenů. Tato varianta byla později upravena na KWP2000 a počítačové rozhraní USB. Jelikož výrobci automobilů přecházejí na sběrnici CAN, je v současné době potřebné do diagnostických kabelů implementovat protokol Can-bus, obvykle doplněný i o předchozí protokoly KW1281 a KWP2000. Tyto diagnostické kabely zpracovávají jen část dat, přičemž nejdůležitější úlohu zajišťuje počítačový program, například VAG-COM od firmy Ross-Tech Ltd. Automobily Škoda od roku 2007 ve většině případů pracují s rozhraním HEX, které lze diagnostikovat pouze pomocí HEX-CAN kabelu.

5.2 Návrh 1: Diagnostika s univerzálním mikroprocesorem

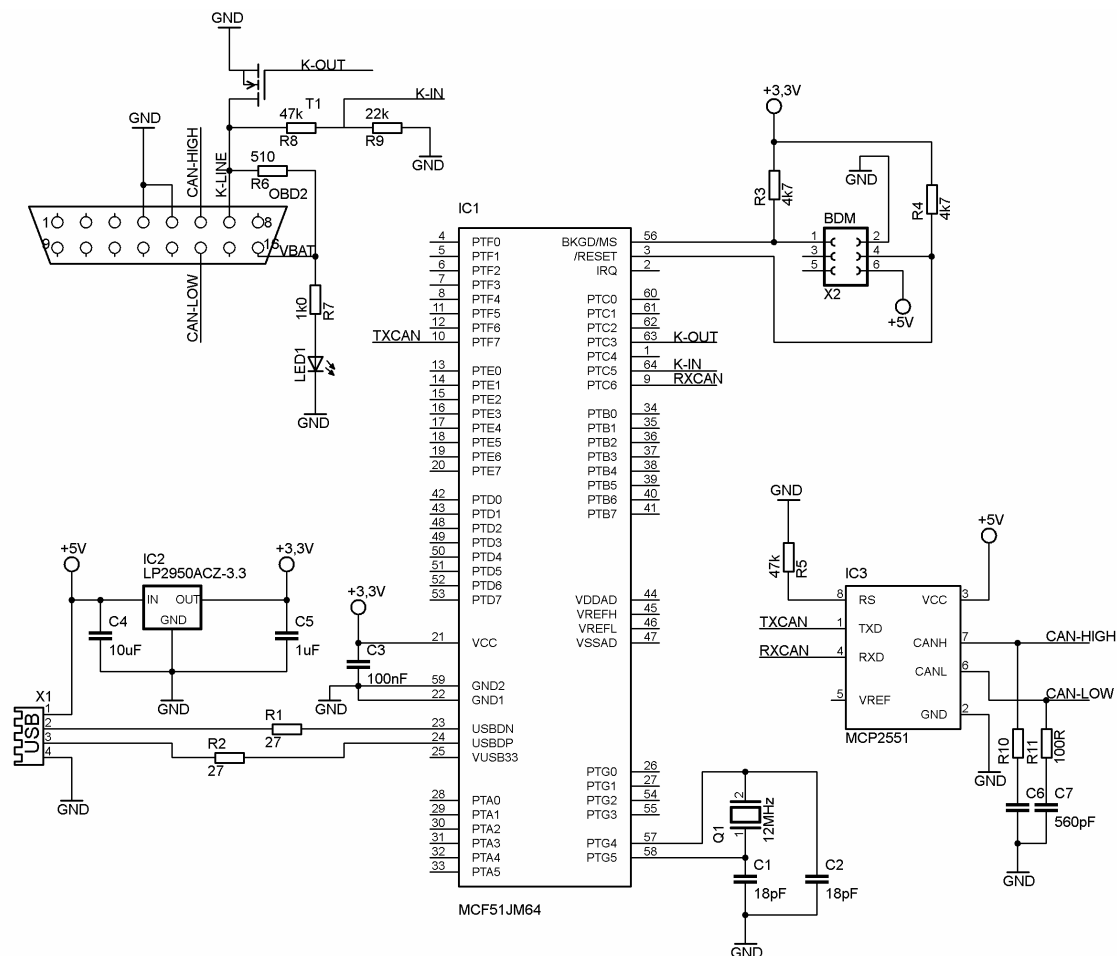
Na obrázku 18 je blokové schéma navrženého diagnostického kabelu. Zařízení je napájeno pomocí sběrnice USB a stabilizátoru napětí 3,3 V. Nejdůležitější částí je mikrokontroler s integrovaným rozhraním USB a CAN. Jelikož mikrokontrolery nemívají fyzickou vrstvu CAN implementovanou, je nutné použít budič sběrnice CAN. Budič sběrnice CAN chrání mikrokontroler před zkratem na sběrnici a zajišťuje komunikaci mezi řadičem CAN a sběrnicí ve 3 režimech: High-Speed, Slope-Control a Standby. V režimu High-Speed má vysílač rychlé vzestupné a sestupné hrany, aby bylo možno využívat maximální rychlosti sběrnice. Slope-Control umožňuje časy náběžné a sestupné hrany řídit pomocí proudu přes externí rezistor. Režim Standby slouží pro odpojení vysílače od sběrnice. Diagnostický kabel používá signalizaci odpojeného zapalování automobilu, aby bylo zřejmé, kdy automobil může komunikovat.



Obrázek 18: Blokové schéma diagnostického kabelu

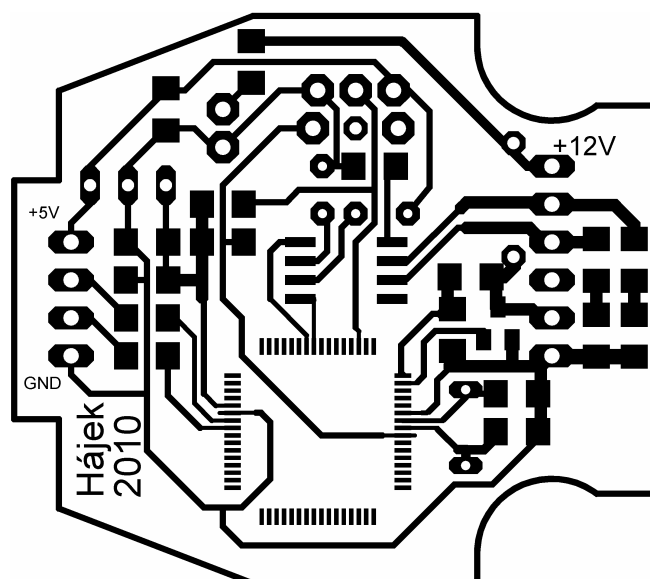
5.2.1 Schéma zapojení

Na obrázku 19 je schéma zapojení diagnostického kabelu. Jako Řídící mikrokontroler je zvolen MCF51JM64 s jádrem ColdFire V1, jelikož má integrované USB a CAN rozhraní a počet vývodů nižší, než u obdobně vybavených mikrokontrolerů ARM. CpldFire je připojen k datovým vodičům USB pomocí ochranných rezistorů R1 a R2 o velikosti 27 Ω . Mikrokontroler je napájen 3,3 V, které jsou získány stabilizací napětí USB rozhraní stabilizátorem LP2950ACZ-3.3. Dle doporučení výrobce stabilizátoru jsou na vstupu a výstupu připojeny filtrační kondenzátory. Z důvodu přesného časování mikrokontroleru komunikujícím po USB rozhraní je nutné použít externí zdroj hodinového signálu o kmitočtu 12 MHz, který je dále násoben na 48 MHz. Pro naprogramování mikrokontroleru MCF51JM64 slouží 6 pinový konektor BDM, kterým je diagnostické rozhraní propojeno s programátorem OSBDM (Open Source Background Debug Module) dle [8]. Fyzickou vrstvu protokolu CAN zajišťuje obvod MCP2551 – budič sběrnice. Pin RS slouží k nastavení režimu budiče mezi 3 stavy: High-Speed, Slope-Control a Standby. Nastavení se provádí hodnotou externího rezistoru proti zemi. Piny TXD a RXD slouží pro propojení s CAN řadičem (vysílání a přijímání dat). Obvod vyžaduje napájecí napětí od 4,5 V do 5,5 V, proto je napájen z USB. Pro komunikaci s automobilem slouží piny CAN_H a CAN_L připojené na 6. a 14. pin OBD2 konektoru. Pro komunikaci protokolem KWP mikrokontroler využívá 1 výstupní a 1 vstupní pin pro vedení K. Vedení L není implementováno, jelikož jej testovaný automobil Škoda Fabia 2 a mnoho dalších nepoužívají. Při zapnutí zapalování je na pinu 16 OBD2 konektoru napětí přibližně 12 V, což je signalizováno pomocí LED. Hodnoty součástek odpovídají doporučeným katalogovým zapojením.

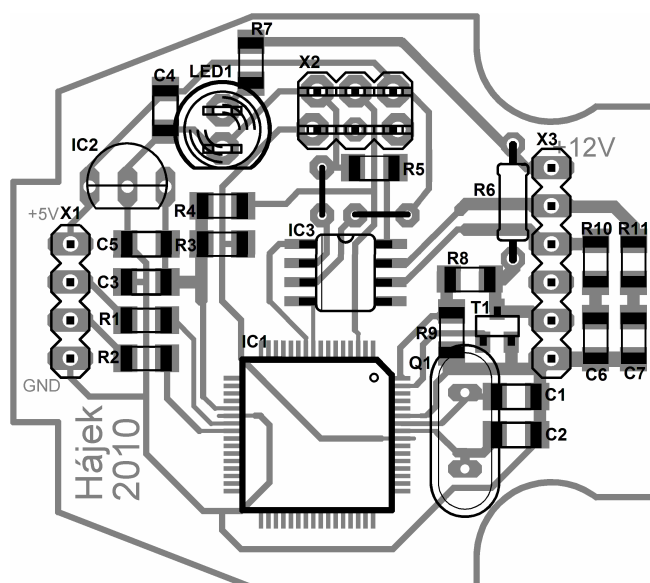


Obrázek 19: Schéma zapojení diagnostického kabelu

5.2.2 Deska plošných spojů diagnostického rozhraní s MCF51JM64



Obrázek 20: DPS diagnostického kabelu (šířka 43 mm, výška 38 mm)



Obrázek 21: Osazovací výkres

Tvar plošného spoje respektuje rozměry krabičky, do které bude umístěn. Desku plošných spojů tvoří jednovrstvá deska, která je osazena vývodovými součástkami i SMD. Na osazovacím výkresu se některé součástky překrývají, například LED1 nebo Q1. To však nevadí, jelikož se jedná o vývodové součástky umístěné na druhé straně než SMD součástky. Při osazování je vhodné nejprve zapájet IC1 a 2 propojovací vodiče nacházející se nad IC3.

Tabulka 6: Rozpis elektronických součástek pro rozhraní s MCF51JM64

označení	ks	název/hodnota	pouzdro	poznámka
IC1	1	MCF51JM64	64 LQFP	MCU ColdFire V1
IC2	1	MCP2551	8LD-SOIC	ekvivalent PCA82C250
IC3	1	LP2950ACZ-3.3	TO92	ekvivalent LE33CZ
X2	1	S2G6	---	lámací dvouřadá lišta
X3	1	S1G6	---	lámací jednořadá lišta
T1	1	BSS138	SOT-23	NMOS tranzistor
Q1	1	XTAL 12MHz	HC49U/S	
R1, R2	2	27R	SMD 1206	
R3, R4	2	4k7	SMD 1206	
R5, R8	2	47k	SMD 1206	
R6	1	510R	0207	
R7	1	1k0	SMD 1206	
R9	1	22k	SMD 1206	
R10, R11	2	100R	SMD 1206	
C1, C2	2	18 pF (CK +18P NPO)	SMD 1206	keramický
C3	1	100 nF (CK +100N Z5U)	SMD 1206	keramický
C4	1	10 uF (CTS 10M/16V B)	SMD 1206	tantalový
C5	1	1 uF (CTS 1M/16V A)	SMD 1206	tantalový
C6, C7	2	560 pF (CK +560P NPO)	SMD 1206	keramický
LED1	1	L-HLMP-3507	A5,0-1	zelená, d = 5 mm

5.3 NÁVRH 2: DIAGNOSTIKA SE SPECIÁLNÍM OBVODEM ELM 327

Z důvodu nedostatku literárních podkladů k realizaci aplikace pro 1. návrh diagnostického kabelu (chybějící ISO standardy o protokolech) a podstatné časové náročnosti je navrženo 2. řešení. Diagnostika prostřednictvím obvodu ELM327, tedy RS232-OBD překladače, má vůči 1. řešení níže popsané výhody a nevýhody.

Výhodou je, že pomocí ELM327 lze jednoduše navrhnout diagnostické rozhraní, protože jej výrobce v katalogovém listu obvodu [15] popisuje. Výrobce specifikuje kromě hardwaru i softwarové příkazy, čímž výrazně ulehčuje tvorbu aplikací. Uživatel se nemusí starat o časování konkrétních protokolů, obvod má implementována všechna rozhraní (PWM, VPN, KWP i CAN). ELM327 je univerzální – funguje na automobilech všech výrobců.

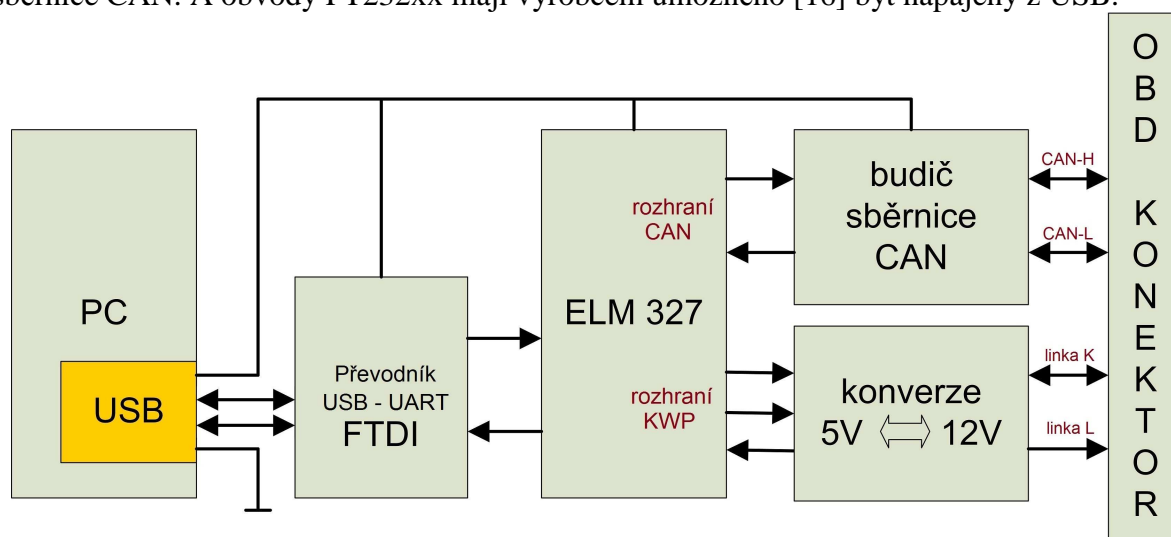
Hlavní nevýhodou je, že se jedná o OBD2 diagnostický obvod, který diagnostikuje pouze standardizované prvky automobilů podporující OBD2 (některé automobily od 1996, každé od roku 2003). Další nevýhodou je, že v případě potřeby změny firmware ELM327 není snadné jeho úprava, neboť zdrojový kód není oficiálně dostupný.

Shrnutím výše zmíněných informací a s respektováním podstatných nedostatků literárních podkladů není reálné návrhem 1 splnit zadání diplomové práce. Proto je preferován návrh druhý. ELM327 dokáže číst reálná data snímačů, skenovat a mazat paměť závad a další operace. Není možné prostřednictvím tohoto obvodu provádět chiptuning, protože toto rozhraní nepracuje s pamětí ECU.

5.3.1 Blokové schéma s ELM327

Na obrázku 22 je uvedeno blokové schéma diagnostického rozhraní s ELM327. Jelikož se tato práce nezabývá americkými standardy PWM a VPN, nejsou v navrženém schématu použity příslušné vývody ELM327. Schéma je podobné návrhu 1. Odlišnost tvoří absence stabilizátoru napětí, externí převodník USB-UART a jiný mikrokontroler. Stabilizace napájení

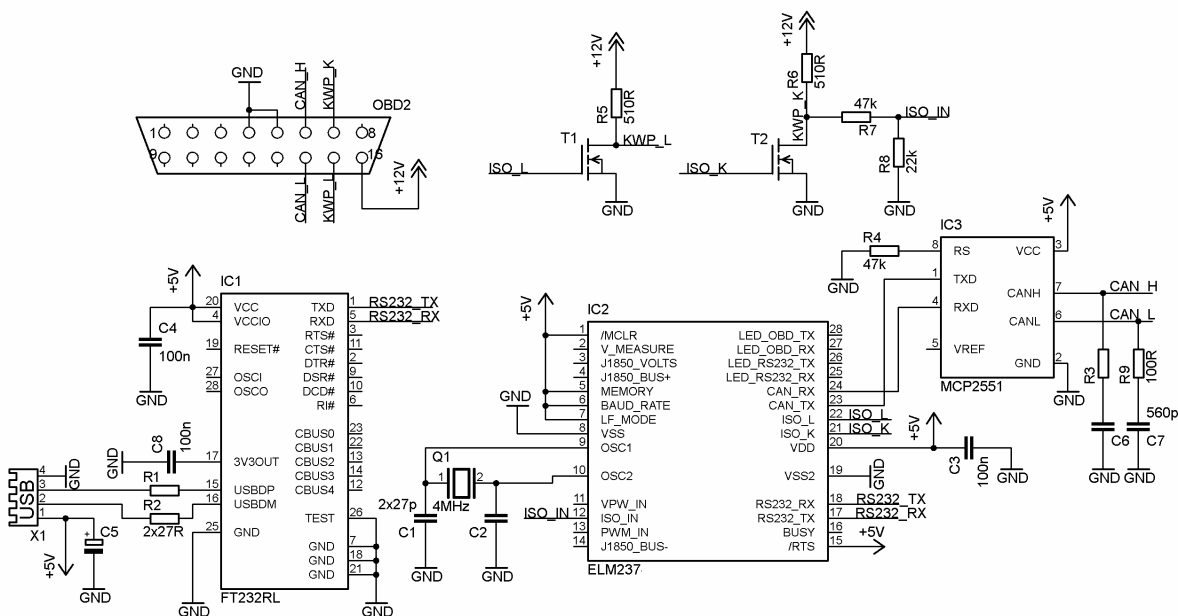
není nutná, jelikož povolený rozsah napájecího napětí je 4,5 až 5,5 V pro ELM327 i budič sběrnice CAN. A obvody FT232xx mají výrobcem umožněno [16] být napájeny z USB.



Obrázek 22: Blokové schéma s ELM327

5.3.2 Schéma diagnostického rozhraní s ELM327

V katalogovém listu [15] je doporučené schéma obvodu ELM327 komunikující s RS232. Schéma uvedené na obrázku 23 z tohoto zdroje vychází a je pouze doplněno o rozhraní USB s FT232RL dle [16]. Datové vodiče USB rozhraní D+ a D- jsou přes ochranné rezistory přivedeny do převodníku FT232RL. Převodník využívá interní oscilátor, jehož frekvence je pomocí fázového závěsu násobena na 48 MHz potřebných pro USB. Výstupní signál převodníku je přiveden do obvodu ELM327. Obvod ELM327 je mikroprocesor PIC18F2580 s programem pro OBD2 diagnostiku distribuovaný firmou Elmelectronics. Tento jednočip pracuje na frekvenci 4 MHz a pomocí AT příkazů ovládá příslušné OBD2 rozhraní.



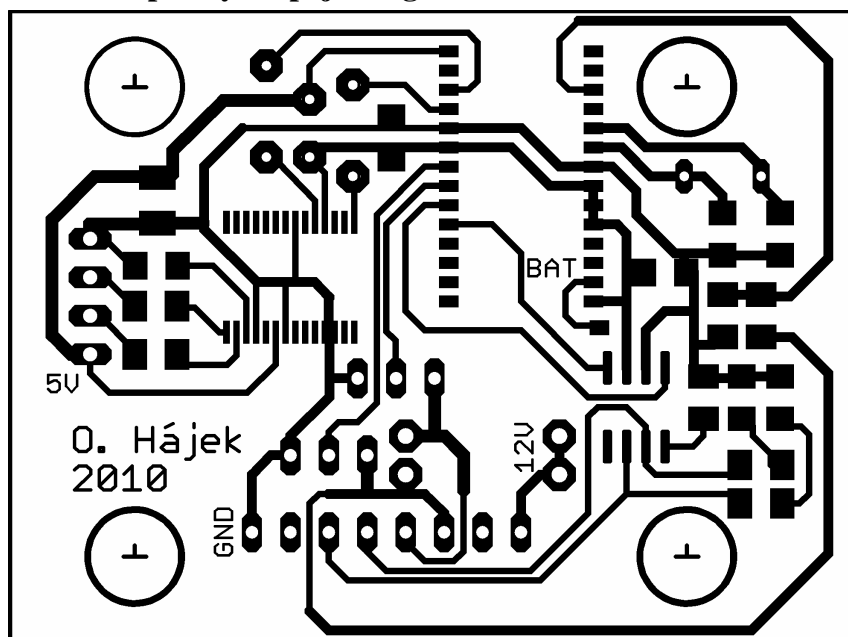
Obrázek 23: Schéma diagnostického rozhraní s ELM327

Implementovaná rozhraní jsou KWP (ISO 9141, ISO 14230) a CAN-bus. Rozhraní KWP využívá v mikroprocesoru integrovaný UART a pouze převádí napěťové úrovně z 5 V na 12 V. Přenos z mikroprocesoru (počítače) do automobilu je realizován spínáním 12 V z 16. pinu OBD2 konektoru pomocí tranzistoru. Hodnota pull up rezistorů 510 Ω je

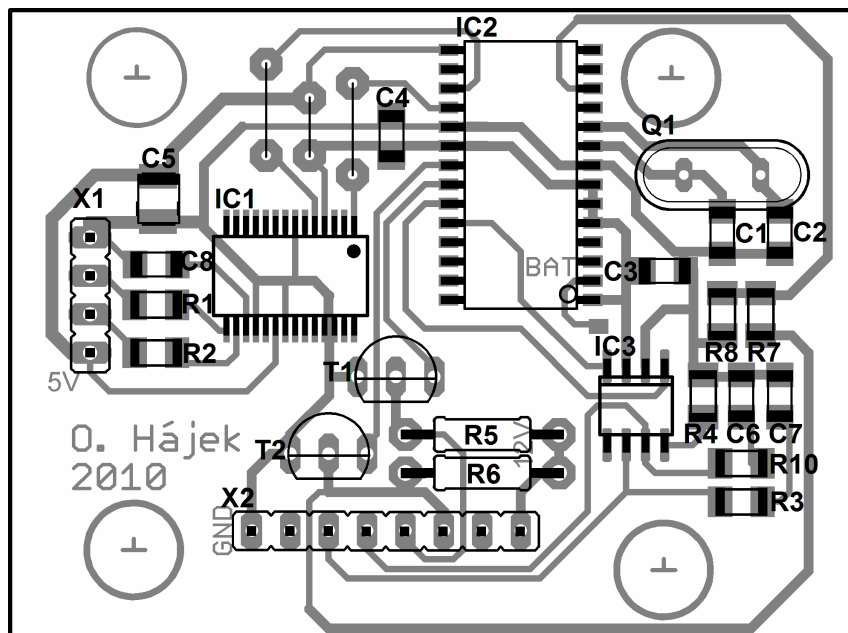
specifikována v [12]. Čtení dat z automobilu probíhá pouze po vedení K. Konverzi napětíových úrovní zajišťuje odporový dělič napětí s poměrem přibližně 0,3188 vstupního napětí.

Rozhraní CAN využívá v mikroprocesoru integrovaný řadič CAN, který je spojen s budičem sběrnice CAN MCP2551. Obvod MCP2551 pracuje v režimu High-speed, což je určeno odporem rezistoru R4. Každé z výstupních vedení je spojeno k zemi přes RC člunek a sběrnice není zakončena 120 Ω rezistorem, což je v souladu s doporučením dle [17].

5.3.3 Deska plošných spojů diagnostického rozhraní s ELM327



Obrázek 24: Deska plošných spojů s ELM327 (šířka 56 mm, výška 42 mm)



Obrázek 25: Osazovací výkres DPS s ELM327

Tvar navržené desky je přizpůsoben krabičce, do níž bude vložen, proto jsou v desce 4 otvory pro distanční sloupky. OBD2 konektor se připojuje ke konektoru X2. Deska plošných spojů vyžaduje 3 propojovací vodiče umístěné vlevo od IC2.

Tabulka 7: Rozpis elektronických součástek pro rozhraní s ELM327

označení	ks	název/hodnota	pouzdro	poznámka
IC1	1	FT232RL	28LD-SSOP	
IC2	1	ELM327	28LD-SOIC	
IC3	1	MCP2551	8LD-SOIC	ekvivalent PCA82C250
X1	1	S1G4	---	lámací jednořadá lišta
X2	1	S1G8	---	lámací jednořadá lišta
Q1	1	XTAL 4MHz	HC49U/S	
T1, T2	2	BS170	TO92	NMOS tranzistor
R1, R2	2	27R	SMD 1206	
R3, R9	2	100R	SMD 1206	
R4, R7	2	47k	SMD 1206	
R5, R6	2	510R	0207	
R8	1	22k	SMD 1206	
C1, C2	2	27 pF	SMD 1206	keramický
C3, C4, C8	3	100 nF	SMD 1206	keramický
C5	1	10 uF	SMD 1206	tantalový
C6, C7	2	560 pF	SMD 1206	keramický

6 VLASTNÍ APLIKACE

6.1 Vývoj aplikace pro ELM327

6.1.1 Základní informace

Ukázková aplikace je vytvořena ve Visual studiu 2008 v jazyku C#. Jako zdroj informací k realizaci aplikace slouží především předmět Programování v .NET a C# (XMW5) a internet. Na rozdíl od Borland C++ Builderu má Visual studio komponentu pro sériový port, což práci usnadňuje. Pro spuštění aplikace vytvořené ve Visual studiu 2008 je nutné mít nainstalovaný .NET Framework 3.5. Aplikace je typu Windows forms, tedy grafické rozhraní obsahující formulář a několik komponent. Kromě standardních komponent jsou použity pro zobrazování otáček motoru a rychlosti volně dostupný budík aGauge [21], pro polohu plynového pedálu a vypočtené zatížení motoru VU_Meter a pro graf komponentu ZedGraph [26].

6.1.2 Příkazy

ELM327 podporuje 2 typy zpráv. Zprávy interní, které slouží ke komunikaci mezi počítačem a ELM327, umožňují nastavovat mikrokontroler nebo zjišťují informace (verze firmware, napětí akumulátoru), aniž by došlo ke komunikaci s automobilem. Tyto zprávy vždy začínají znaky 'AT' a jsou podrobně popsány v [15].

např.: **AT I** - vypíše verzi firmware
AT Z - resetuje mikrokontroler do továrního nastavení + vypíše firmware
AT RV - vypíše napětí akumulátoru (zjistí A/D převodem na pinu 16)
AT SP A - nastaví protokol na automatický
AT H1 - zobrazí hlavičku a CRC

Druhým typem zpráv jsou OBD2 příkazy, které jsou standardizovány v ISO 15031-5. Příkaz OBD2 má délku 2 nebo 4 znaky. První dvojice znaků udává režim OBD2, druhá dvojice PID. Dle kapitoly 1.4.1 tohoto dokumentu získáme odesláním '010C' zprávu s aktuálními otáčkami motoru. Přijatá zpráva může mít vícero podob. Záleží na tom, zda je zapnuté nebo vypnuté echo nebo hlavička. Základní nastavení po resetu zapne echo a skryje hlavičku.

010C 41 0C 00 00

> echo = 010C, následují 4 byty odpovědi 41 0C 00 00 (1.byte zvýšen o 0x40 jako v kap.2.1.5)

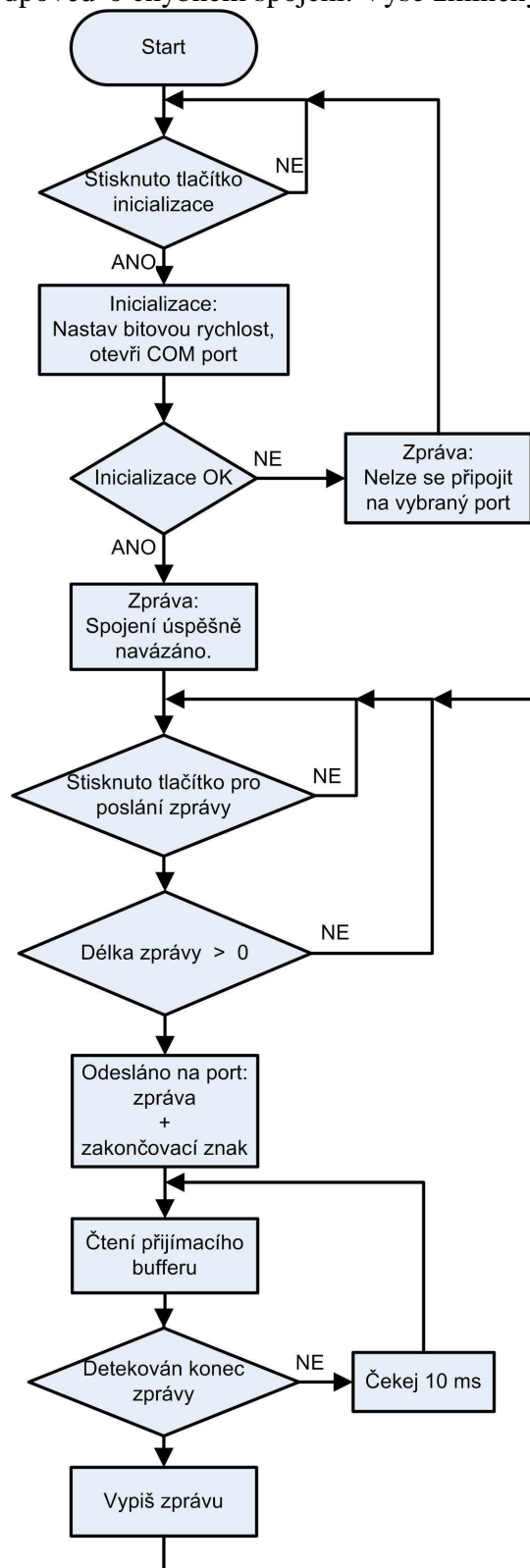
010C 48 6B 10 41 0C 00 00 10

> echo + 3 byty hlavička (adresa testeru 6B, adresa ECU 10) + odpověď + CRC

6.1.3 Způsob komunikace

Komunikace po sériovém portu je separována do souboru SerialComunicator.cs. Aplikace umožňuje nastavovat číslo COM portu od COM1 po COM9. Mikrokontroler ELM327 pracuje pouze na komunikační rychlosti 38400 baudů. Velké potíže způsobovalo přijímání zpráv, jelikož po odvysílání zprávy nebyla přijata odpověď. Mikrokontroler ELM327 přijímá znaky a jakmile detekuje znak 0x0D = Carriage return („návrát vozíku“), porovná přijatou zprávu se seznamem platných zpráv (dle [15]). ELM327 nerozlišuje velká a malá písmena. Pokud zpráva v seznamu není, mikrokontroler odešle znak '?'. Je-li zpráva v seznamu, bude vykonána konkrétní operace a její výsledek odeslán do počítače. Aby vše fungovalo, je nutné jednotlivé znaky žádosti před odvysíláním převést do hexadecimální soustavy. V takovémto případě již je funkční přijímání a odesílání zpráv. Bohužel nastává situace, že program nepřechte celou odpověď nebo dojde k rozdělení odpovědi na více řádků. Jelikož každá odpověď končí znakem 0x3E = '>', je nutné číst přijímací buffer dokud se nepřijme tento znak. Visual studio automaticky využívá při použití komponenty sériového portu ve formulářové aplikaci více vláken. Pro korektní funkci je nutné testovat, zda je nutné zajistit

synchronizaci (Invoke Required). Přijetí celé odpovědi je identifikováno znakem 0x3E. Dokud není zakončovací znak přijat, program se přepíná do úsporného módu. V případě, že vyprší lhůta pro příjem zprávy (standardně 3 sekundy), je příjem přerušen a zaslána odpověď o chybném spojení. Výše zmíněný algoritmus je uveden na obrázku 26.



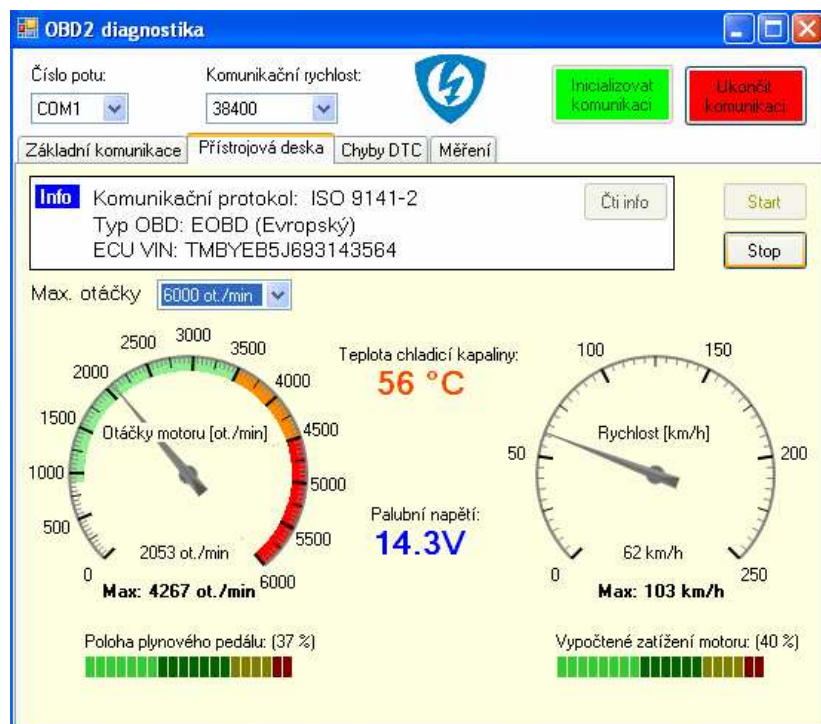
Obrázek 26: Zjednodušený algoritmus základní komunikace ve vytvořené aplikaci

6.1.4 Vytvořená aplikace

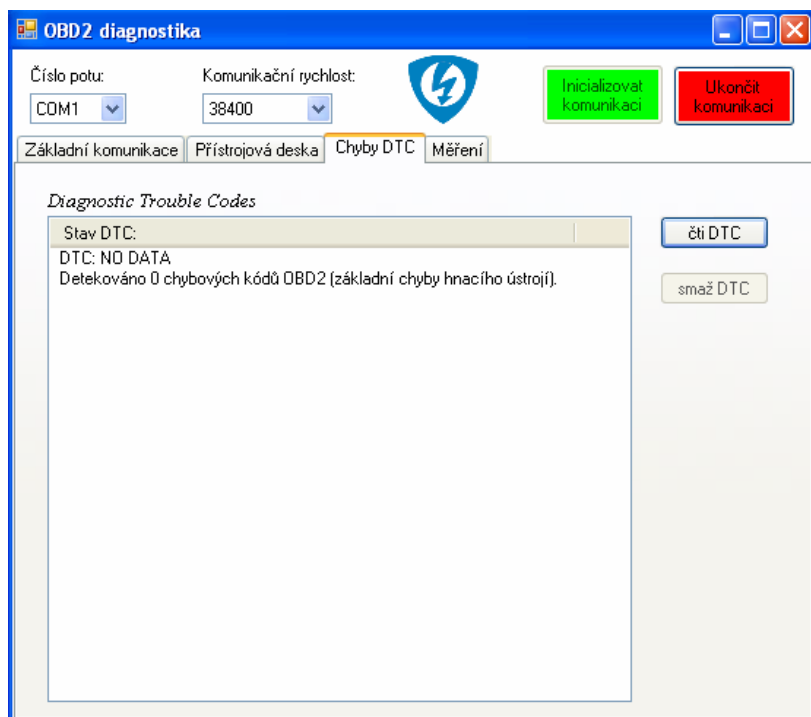
Vytvořená aplikace se skládá ze čtyř panelů, které demonstrují funkce OBD2 diagnostiky. První panel tvoří konzole pro zaslání libovolných zpráv a možnost zaslání periodických zpráv s volitelným intervalem. V případě, není-li stisknutí určitého tlačítka povoleno (například tlačítko pro poslání zprávy při neinicializované komunikaci), je komponenta tlačítka vypnuta, nebo je při stisku testována inicializace sériového portu.



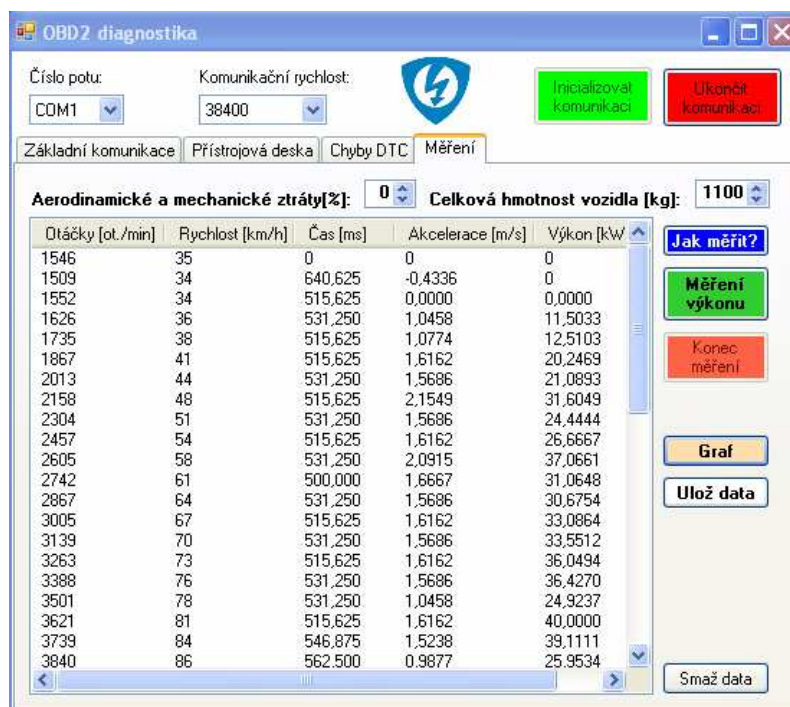
Obrázek 27: Vzhled aplikace – panel Základní komunikace



Obrázek 28: Vzhled aplikace – panel Přístrojová deska



Obrázek 29: Vzhled aplikace – panel Chyby DTC



Obrázek 30: Vzhled aplikace – panel Měření

Druhý panel Přístrojová deska graficky interpretuje základní data režimů 1 a 9 OBD2 standardu. Obsahuje blok informace o vozidlu, který identifikuje automobil dle komunikačního protokolu, normy OBD a VIN kódu řídicí jednotky. Dále je zde přístrojová deska s nastavitelným rozsahem otáčkoměru. Pro čtení dat je zvolena priorita zpráv, jelikož rychlost vozidla je vhodné častěji číst než například teplotu chladicí kapaliny. Informace s vysokou prioritou, do nichž patří otáčky motoru, rychlost vozidla, poloha plynového pedálu a vypočtené zatížení motoru, jsou data, která se velmi často mění a při jízdě jsou důležitá.

Informace s nízkou prioritou, tedy teplota chladicí kapaliny a napětí akumulátoru, se mění pomalu a proto jsou čteny méněkrát.

Pro řízení čtení dat byl nejprve napsán algoritmus, který každé informaci přiřadil určitý počet časových oken. Například zpráva s vysokou prioritou 5 oken, zpráva s nízkou prioritou 1 okno. Do jednotlivých oken byly poté generovány zprávy. Po vyčerpání všech zpráv se celý cyklus opakoval. Tato metoda umožňovala v případě přidávání nebo odebrání čtených informací snadnou modifikaci programu. Bohužel generovaná sekvence nebyla nikdy optimální a vizuálně aplikace vykazovala u některých komponent velká zpoždění. Proto bylo nakonec použito fixně předdefinované řízení čtených zpráv.

Třetí panel slouží pro reprezentaci 3. a 4. režimu OBD2 standardu, tedy čtení a mazání paměti závad. Bohužel z důvodu, že všechny testované automobily byly bez závad a úmyslné vyvolání chyby na palivovém nebo výfukovém systému automobilu není pro laika bezpečné, není tato část programu detailněji propracovaná a testovaná. Aplikace zobrazuje načtené chyby a pokud nějaké skutečně existují, povolí stisknutí tlačítka pro smazání chybových kódů.

Čtvrtý panel slouží pro dynamické měření výkonu automobilu. Takovéto měření umožňuje porovnávat efektivnost prováděných úprav na vozidle nebo k porovnání výkonu různých vozů v celém spektru otáček motoru. Měření lze provádět na libovolný rychlostní stupeň, výsledný průběh výkonu je téměř stejný. Měření hodnot přes diagnostické rozhraní je však relativně pomalé, proto čím vyšší rychlostní stupeň bude při měření výkonu zařazený, tím více hodnot a věrohodnější výkonovou křivku získáme. Výsledný výkon vypočteme dle vztahu (1) [6].

$$P = m \cdot a \cdot \frac{v}{3,6} \quad (1)$$

P - výkon motoru [W]

m - celková hmotnost automobilu [kg]

a - zrychlení automobilu [$\text{m}\cdot\text{s}^{-2}$]

v - rychlost automobilu [km/h]

Celkovou hmotnost automobilu lze přibližně určit dle suché hmotnosti udávané výrobcem, množství natankovaného paliva a hmotnosti posádky. Okamžitou rychlost vozidla získáme pomocí diagnostiky z automobilu. Akceleraci vypočteme ze vztahu (2) [6].

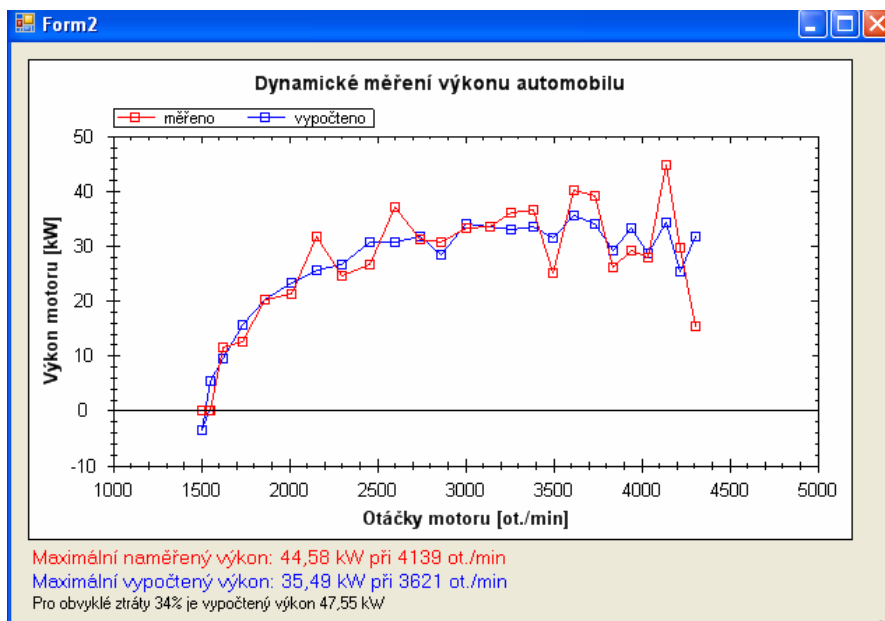
$$a = \frac{v_2 - v_1}{3,6} \cdot \frac{1}{t_2 - t_1} \quad (2)$$

a - zrychlení automobilu [$\text{m}\cdot\text{s}^{-2}$]

v_1 - rychlost automobilu [km/h] v čase t_1

t_1 - čas 1. měření [s]

Pro výpočet akcelerace tedy pracujeme s dvěma hodnotami měřeními v určitém časovém intervalu. Hodnota časového intervalu nemusí být stejná, jelikož dochází v automobilu nebo počítači k různým zpožděním, proto je získán z počítače přesný čas. Kromě okamžité rychlosti a času je dále nutné číst otáčky motoru, kterým je vypočtený výkon přiřazen. Výsledkem měření je tedy závislost výkonu na otáčkách motoru.



Obrázek 31: Průběh výkonu vozu Škoda Fabia II 1,4 TDI na 3. rychlostní stupeň

Naměřené hodnoty (křivka „měřeno“) vycházejících ze vztahů (1) a (2) jsou hodně za očekávanými výsledky publikovaných grafů profesionálních měření. Problém je v měření rychlosti. Při jízdě na 3. rychlostní stupeň dosahoval vůz maximální rychlosti 98 km/h. Celkový počet naměřených hodnot je 28. Měření probíhá přibližně každých 0,5 sekundy. Pro porovnání vytvořeného algoritmu nejrychlejší čtení otáček a rychlosti programem VCDS 908.1 je 3,6x za sekundu. Během každého intervalu se rychlost změnila maximálně o 4 km/h a otáčky o 155 ot./min. Z důvodu, že rychlost vozidla dle OBD2 standardu má rozlišení 1 km/h, je takovéto měření velice nepřesné.

Možné řešení je rychlost počítat z otáček motoru. Pokud uvažujeme závislost rychlosti na otáčkách motoru lineární, můžeme okamžitou rychlost vypočítat ze vztahu (3).

$$v_{aktua\ln i} [km/h] = v_{Nmax} [km/h] \cdot \frac{n_{aktua\ln i} [ot./min]}{n_{max} [ot./min]} \quad (3)$$

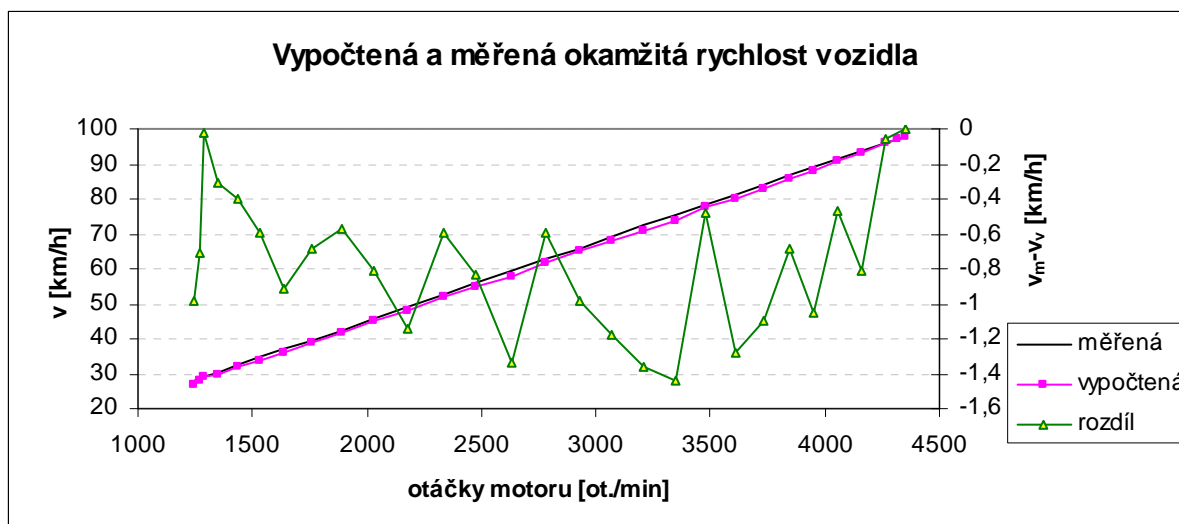
$v_{aktua\ln i}$ - aktuální rychlost automobilu [km/h]

v_{Nmax} - rychlost automobilu při maximálních otáčkách motoru [km/h]

$n_{aktua\ln i}$ - aktuální otáčky motoru [ot./min]

n_{max} - maximální otáčky motoru [ot./min]

Jedná se o jednu z možností, jak celé měření zachovat automatizované bez nutnosti znalosti parametrů vozidla. Průběh vypočteného výkonu je na obrázku 33 označen jako „vypočteno“ a dosahuje věrohodnější hodnot. Tvar křivky nemá tak velké zvlnění a maximální hodnota výkonu včetně všech ztrát 35,49 kW je reálná. Mechanické ztráty způsobí, že na kolech automobilu je přibližně o 15 až 20 % nižší výkon než na motoru. Další nezanedbatelný pokles naměřeného výkonu způsobí aerodynamické ztráty. Proto maximální hodnota 44,58 kW z měření je příliš vysoká a je způsobena nepřesným měřením rychlosti.



Obrázek 32: Rozdíl mezi měřenou a vypočtenou okamžitou rychlostí

Existuje také metoda, kdy rychlost automobilu stanovíme z otáček motoru dle parametrů vozidla. Tím se znatelně urychlí doba jednoho měření, protože přes diagnostické rozhraní budou pouze čteny otáčky motoru. Hlavní nevýhodou je, že musíme pro každý automobil i zařazený rychlostní stupeň počítat s jinými konstantami. Pokud je v automobilu použito jiných převodů než u originálu, vypočtená rychlost nebude skutečná. Výpočet okamžité rychlosti s pomocí známých převodů automobilu ze vztahu (4) [6].

$$v_{\text{aktualní}} = \frac{n_{\text{aktualní}}}{i_{\text{celkový}}} \cdot \pi \cdot D \cdot 0,06 \quad (4)$$

$v_{\text{aktualní}}$ - aktuální rychlost automobilu [km/h]

$n_{\text{aktualní}}$ - aktuální otáčky motoru [ot./min]

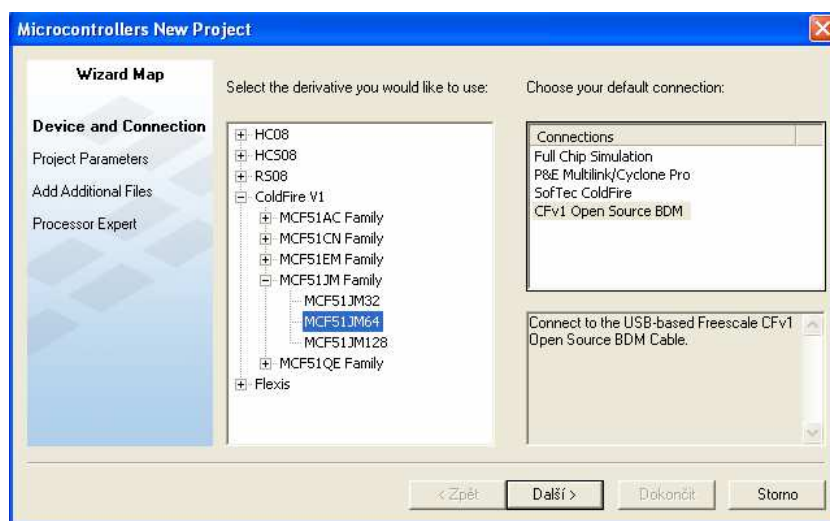
i – celkový převod [-] (součin převodových poměrů zařazeného rychlostního stupně a stálého převodu)

D – průměr pneumatiky [m]

Škoda Fabia 1,4 TDI PD/51 kW má pro jednotlivé rychlostní stupně následující převodové poměry: 3,77 – 2,10 – 1,39 – 1,03 – 0,78, stálý převod 3,370. Výsledky byly velmi podobné jako při výpočtu rychlosti dle (3).

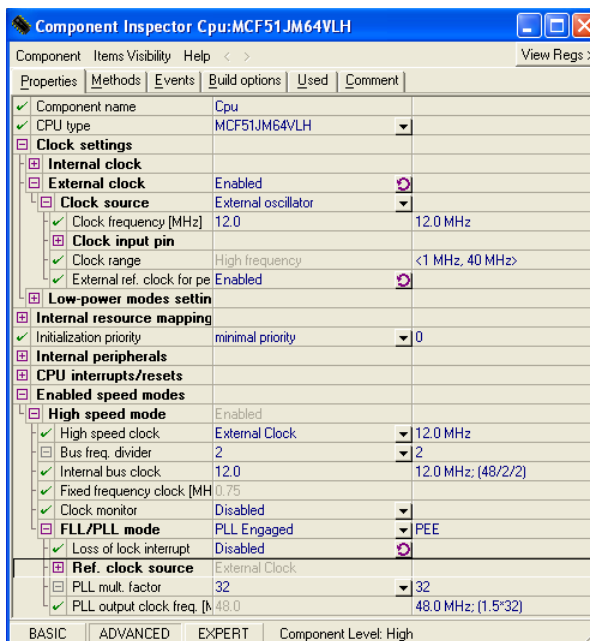
6.2 VÝVOJ APLIKACE PRO MCF51JM64

Pro programování mikrokontrolerů od firmy Freescale semiconductors slouží vývojové prostředí CodeWarrior. Nejnovější verze podporující ColdFire V1 je CodeWarrior 6.3. Novější verze podporují pouze ColdFire V2 a novější mikrokontrolery. Při vytvoření nového projektu musíme kromě specifikace typu programovaného mikrokontroleru uvést, jakým programátorem jej budeme programovat. Programátory P&E Multilink a OSBDM používají BDM rozhraní (6-pinový konektor, kde 1 vodič slouží pro data, 1 pro reset, 2 vodiče pro napájení a 2 nejsou využity – viz obrázek 19). V případě, že je od výroby nebo při nekorektním naprogramování dojde k uzamčení BDM rozhraní, musí být k naprogramování mikrokontroleru použit jiný programátor, například CYCLONE PRO.



Obrázek 33: Vytvoření nového projektu v CodeWarrioru 6.3

Program je tvořen v jazyku C. Uživatel si může vybrat ze tří možností urychlení vývoje aplikace: žádná, inicializace zařízení nebo Processor Expert. Processor Expert umožňuje grafické nastavení parametrů jednotlivých komponent a vygeneruje inicializační kód.

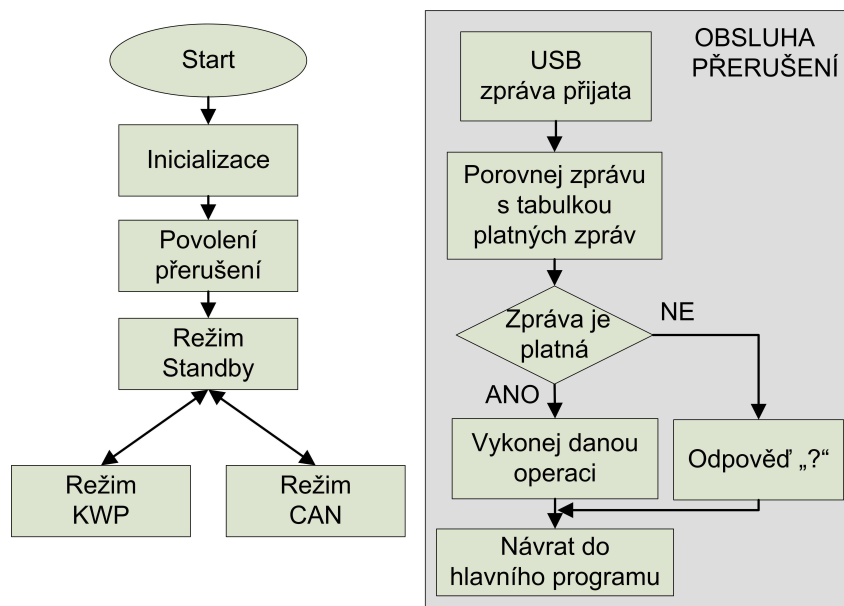


Obrázek 34: Nastavení CPU v režimu Processor Expert

Na obrázku 34 je znázorněno nastavení CPU. Je povolen externí hodinový signál o frekvenci 12 MHz. Pro USB je aktivována fázová smyčka PLL, jejíž výstupní frekvence je 48 MHz. Po nastavení CPU je zapotřebí přidat komponenty, které budou komunikovat s okolím mikrokontroleru. U komponenty BIT nebo BYTE specifikujeme, o který pin/port se jedná, zda je vstupní nebo výstupní, nastavení pull-up rezistoru a počáteční hodnotu. Tyto parametry lze později v kódu měnit, a tak například specifikace pinu jako vstupní nemusí být v celém programu fixní. Každá komponenta má několik metod, kterými ji můžeme používat. U BITu je to například nastavení, vynulování, negování a další. Pro přehlednost kódu některé komponenty automaticky generují další zdrojový soubor, proto při použití časovače se kromě souboru main.c vytvoří events.c, v němž bude obslužný kód pro přerušení časovače.

Hlavní limitace v režimu Processor Expert je omezený počet komponent. Pro vývoj diagnostického rozhraní klíčová komponenta USB není standardně dodávána s vývojovým prostředím. Lze však použít dostupné knihovny a naprogramovat USB bez pomoci Processor Expertu. Během vývoje programu došlo k nekorektnímu naprogramování a k uzamčení BDM. To znamenalo v mém případě nutnost výměny mikrokontroleru. Tyto nedostatky způsobené omezenými funkcemi programátoru byly důvodem k přerušení dalšího intenzivního vývoje diagnostického rozhraní s MCF51JM64.

V případě, že by fungovalo USB, by program pro mikrokontroler měl podobu uvedenou na obrázku 35. V první řadě je cílem zachovat kompatibilitu s vytvořenou aplikací ve Visual Studiu, proto je nutné, aby mikroprocesor podporoval příkazy jako ELM327. Program na začátku inicializuje potřebné periferie, jako jsou asynchronní sériová linka pro komunikaci KWP protokolem, CAN řadič pro komunikaci protokolem Can-bus a USB pro komunikaci s počítačem. Poté je povoleno přerušení a mikrokontroler se přepne do úsporného režimu. Z něho jej může probudit přerušení, vyvolané příchozí zprávou z počítače. Text zprávy je porovnán s tabulkou platných zpráv a v případě, že není nalezen shodný řetězec, je odeslána odpověď „?“ . Pokud zpráva je platná, jsou vykonány odpovídající operace. Zprávy jsou stejně jako v případě ELM327 dvou typů: AT příkazy sloužící pro nastavení parametrů mikrokontroleru ColdFire a OBD2 příkazy ke komunikaci s automobilem. Přepínání mezi režimy je realizováno buď konkrétním příkazem (AT SP x) nebo v případě nastavení protokolu na automatický (AT SP A) je prioritou CAN a poté KWP.



Obrázek 35: Plánovaný algoritmus mikrokontroleru MCF51JM64

Komunikace po CAN by byla realizována s pomocí obrázku 11, kde je popsána struktura dat OBD2 zprávy. Významem zpráv se zabývá kapitola 1.4.1, ale tyto informace nejsou pro mikrokontroler ColdFire podstatné, jelikož jsou záležitostí aplikační úrovně, tedy programu ve Visual Studiu. Mikrokontroler pouze data doplní o adresu řídicí jednotky a odvysílá ji. Data z odpovědi pouze odvysílá po USB do počítače. Pokud automobil neodpoví do nastavitelného timeoutu (například 3 sekundy), tak je odeslána odpověď o chybném spojení. Z režimu CAN se lze vrátit do režimu Stand by ukončením komunikace.

Režim KWP je složitější než CAN, neboť je zde nutná inicializace komunikace a udržování otevřeného spojení. Mikrokontroler by nejprve zkusil rychlou inicializaci (kap. 2.1.5) a v případě neúspěchu pomalou inicializaci (kap. 2.1.6). Pokud by ani pomalá inicializace nebyla úspěšně provedena, byla by odeslána zpráva o chybě inicializace. Formát zpráv je uveden na obrázku 8. Při otevřeném spojení musí mikrokontroler automaticky zasílat zprávu „tester přítomný“ s periodou maximálně 5 sekund, jinak dojde k uzavření spojení. V případě, že chceme ukončit komunikaci, není zpráva „tester přítomný“ zaslána a mikrokontroler se vrátí do režimu Standby.

Pro správnou funkčnost programu je zapotřebí všem použitým proměnným přiřadit výchozí hodnoty. Ty jsou nastaveny na počátku programu při inicializaci. Tím je zaručeno, že po restartování diagnostického rozhraní lze jednoznačně očekávat jeho reakce na dané příkazy. Mezi proměnné patří například vybraný komunikační protokol, přenosová rychlost, hodnota timeoutu, adresa řídicí jednotky, nastavení způsobu zobrazení (echo a hlavička) a další.

7 ZAJÍMAVÉ PRODUKTY

7.1 Simulátor řídicí jednotky

Simulátor řídicí jednotky výrazně ulehčuje počáteční vývoj aplikace. Kromě výhody ladění programu v pohodlí domova se vývojář nemusí obávat případného poškození elektroniky automobilu. Využití simulátoru má velký význam při nastavování časování jednotlivých zpráv. I když skutečná řídicí jednotka automobilu má implementováno mnohem více funkcí, v některých případech je lepší použít simulátor. Příkladem je generování chybových kódů souvisejících s emisemi. U automobilu není, obzvláště pro laika, jednoduché vytvořit takovou chybu pro otestování detekce chyb.

Jedním z výrobců simulátorů řídicích jednotek je Ozen Elektronik LTD. Simulátory podporují konkrétní protokol nebo existuje i varianta komunikující všemi čtyřmi protokoly. Tato zařízení simulují buď jednu řídicí jednotku (ECM), nebo tři jednotky (ECM, TCM, ABS). Pro testování komunikace po Can-bus je vhodný simulátor mOByDic 1610, jenž je na obrázku 36.



Obrázek 36: Simulátor řídicí jednotky mOByDic 1610, převzato z [22]

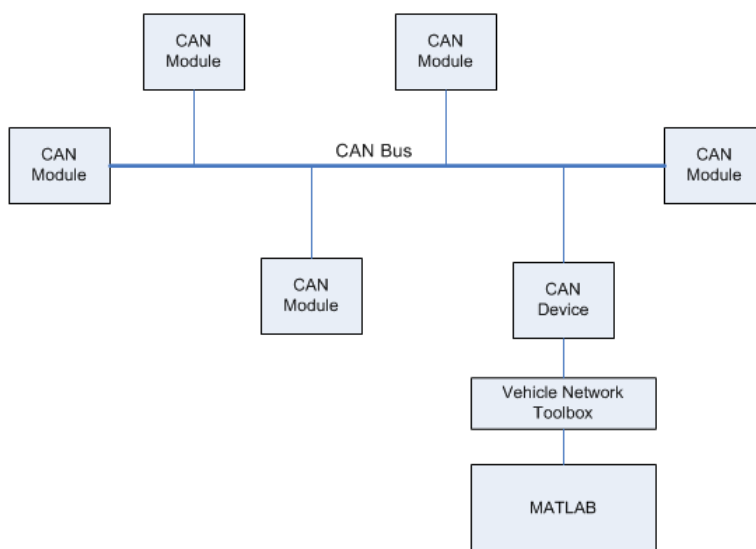
Veškerou činnost simulátoru řídí mikrokontroler OE91C1610 (interní označení výrobce) s hodinovým signálem 16 MHz. Na jeho vstupy A/D převodníku jsou přivedeny potenciometry o velikosti 10 kohmů, které slouží pro nastavování hodnot vybraných snímačů. Potenciometry nastavují: teplotu chladicí kapaliny, otáčky motoru, rychlost, napětí lambdasondy a MAF senzor (hmotnost vzduchu). Modrý přepínač umožňuje nastavovat 11/29 bitový ID zpráv a bitovou rychlost 250/500 kbit/s. Dále je zde tlačítko pro vygenerování chybových kódů. Pro komunikaci po CAN je opět použit budič sběrnice PCA82C250. Celé zařízení je napájeno z adaptéru, jehož napětí je poté stabilizováno na 5 V stabilizátorem 7805. Pro připojení diagnostického testeru je zde standardizovaná OBD2 zásuvka.

7.2 Matlab Vehicle Network Toolbox

Vehicle Network Toolbox je nástroj umožňující v Matlabu nebo Simulinku posílat a přijímat CAN pakety. První vrze se objevila v Matlabu R2009a. Tento nástroj dokáže kódovat, dekódovat, filtrovat a zaznamenávat CAN zprávy v reálném čase. Pro rozsáhlou analýzu dat mohou být data zpracovány offline. V konfiguraci lze specifikovat komunikační rychlost a parametry vysílače dle použitého hardwarového rozhraní. Podporovány jsou standardní 11 bitové i rozšířené 29 bitové identifikátory zpráv. Pro lepší orientaci slouží rovněž časové značky. U přijatých zpráv lze specifikovat formát dat, tedy jak jednotlivé byty zpracovat. Pro vizualizaci dat konkrétního kanálu má Vehicle Network Toolbox implementováno grafické rozhraní canTool GUI.

Vehicle Network Toolbox podporuje vektorová zařízení [23]:

- CANcardX, CANcardXL
- CANcaseXL
- CANboardXL, CANboardXL PCIe, CANboardXL pxi



Obrázek 37: Komunikace toolboxu s okolními CAN moduly, převzato z [23]

8 ZÁVĚR

Počáteční a časově i finančně nejnáročnější fází projektu bylo sehnání dostatečných literárních podkladů pro realizaci diagnostického rozhraní a tvorbu vlastní aplikace. O moderních trendech v automobilech, základních principech a obecném úvodu do automobilové diagnostiky existuje celá řada knih. V rámci tohoto projektu jsem prostudoval 5 titulů. V každé z nich jsem našel několik užitečných informací, nicméně důležitá fakta pro zhotovení diagnostického rozhraní v nich chyběla. Další informace jsem čerpal z odborných příruček automobilek BMW a Škoda, které nejsou určeny pro veřejnost a nejsou k dostání v běžném knihkupectví. V těchto příručkách jsou podrobné informace mimo jiné i o diagnostice. Bohužel, rovněž jako v již zmíněných knihách, autoři předpokládají, že každý zájemce o diagnostiku zakoupí jeden z komerčních, obvykle neuniverzálních, přístrojů. Potřebné informace pro konstrukci a oživení diagnostického rozhraní lze získat z ISO standardů. Pro každý komunikační protokol existují obvykle 4 standardy, zabývající se fyzickou, linkovou, síťovou a aplikační vrstvou protokolu. Navíc další standardy definují požadavky na testovací rozhraní, OBD2 zprávy, diagnostické chybové kódy a další. *(Cena za jeden ze standardů bývá od 50 až 150 USD, tedy zakoupit veškeré materiály není levná záležitost. Bohužel sehnat všechny tyto standardy zdarma se mi nepodařilo.)* Tyto normy nemají český ekvivalent, některé však existují v britské nebo švédské mutaci. Jelikož normy na sebe navazují, výrazně tento fakt zkomplikoval moji budoucí tvorbu.

Tato práce se zabývá výhradně evropskými automobily, tedy vozy komunikující protokoly KW 1281, KWP 2000 a ISO 15765 (CAN). Pro základní představu o elektronice automobilu je v rámci dostupných informací popsána funkce řídicí jednotky EDC15P používané u vozu Škoda Fabia a funkce gateway, která zprostředkovává komunikaci mezi jednotlivými sběrnici automobilu. Současným trendem v automobilovém průmyslu je, že důležité signály jsou přenášeny po sběrnici CAN rozdělující se na sběrnici hnacího ústrojí, sběrnici komfortní elektroniky a diagnostickou sběrnici. Na rozdíl od jednoúčelového vedení pouze pro diagnostiku ušetří toto řešení počet vodičů a pravděpodobnost vzniku chyby. Při vyhodnocování hrozících rizik dle [6] jsem se rozhodl, že o modifikaci dat se nebudu pokoušet. Nicméně možnosti úpravy dat prostřednictvím demoverze WinOLS [20] jsem vyzkoušel na z internetu stažených obsazích řídicích jednotek. Při návrhu diagnostického rozhraní jsem tedy v první řadě kladl důraz na bezpečnost vůči automobilu. Z tohoto důvodu jsem se rozhodl zabývat se pouze OBD2 diagnostikou.

Pro diagnostické rozhraní jsem zvolil mikrokontroler MCF51JM64. Dle specifikací se domnívám, že je pro tento projekt lepší volbou než mikroprocesory ARM, především díky integrovanému CAN řadiči již v pouzdru LQFP64 a nízké ceně jednočipu i programátoru. Rovněž jsem chtěl vyzkoušet originální verzi vývojového prostředí CodeWarrior, které jsem získal na soutěži FTA2008. Bohužel se během vývoje vyskytlo několik vážných problémů s programátorem a nedostupnými komponentami pro CodeWarrior. Již při prvních potížích s naprogramováním mikrokontroleru, které bylo způsobeno vzorky s uzamčeným BDM rozhraním zaslanými firmou Freescale Semiconductor, bylo zřejmé, že by cíl diplomové práce nemusel být splněn. Proto bylo vhodné pracovat na další variantě diagnostického rozhraní. Od této chvíle jsem pracoval souběžně na diagnostickém rozhraní s mikrokontrolerem ColdFire V1 a ELM327.

Práce na variantě s ELM327 vykazovala výrazně větších úspěchů, a tak jsem aplikaci ve Visual studiu 2008 psal právě pro ni. Aplikace umožňuje využívat veškeré standardizované příkazy libovolného režimu OBD2. Dále jsou graficky zpracovány režimy 1, 3, 4 a 9. Režim 1 umožňuje číst aktuální data ze snímačů automobilu. Vytvořená aplikace obsahuje virtuální palubní desku s pamětí na maximální rychlost a otáčky motoru. Režim 1 je rovněž využit při dynamickém měření výkonu automobilu. Režim 3 detekuje diagnostické chybové

kódy DTC a s pomocí režimu 4 je dokáže smazat. Pro ideální odladění těchto režimů by byl vhodný simulátor řídicí jednotky uvedený v kap. 7.1. Režim 9 obsahuje některé důležité informace o řídicí jednotce. Nejdůležitější z nich, VIN kód řídicí jednotky, aplikace dokáže interpretovat. Jako příklad využití diagnostiky je do aplikace implementováno dynamické měření výkonu, které slouží k objektivnímu posouzení případných úprav v celém spektru otáček. Značná pozornost byla věnována ošetřením kritických míst, aby nedošlo k pádu aplikace.

Pro realizaci projektu bylo nutné nastudovat informace o protokolech KWP, CAN, ale i USB. V této práci jsem si vyzkoušel práci s 32 bitovým mikrokontrolerem ColdFire V1 a programování v CodeWarrioru a Visual studiu 2008. S prací jsem se úspěšně zúčastnil soutěže FTA2010. Bohužel jsem se přesvědčil, že v této oblasti bez investovaných financí se leckdy nelze „pohnout z místa“ a v případě větších investic řádově 10000 Kč by bylo možné dosáhnout lepších výsledků. Jak se však časem ukázalo, některé materiály, o jejichž koupi jsem uvažoval, by mi nepomohly. Jelikož si každý své know-how pečlivě hlídá, je velice těžké v tomto odvětví získat odpovědi na své otázky. Tímto projektem jsem si výrazně obohatil znalosti v automobilové oblasti, které mohou přijít vhod, jelikož automobil tvoří nedílnou součást mnoha lidí po celý život.

Tabulka 8: Testované automobily rozhraním ELM327

automobil (rok výroby)	komunikační protokol	detekované řídicí jednotky
Škoda Fabia2 (2009)	KW 1281 (pomalá inicializace)	\$10 - motor
Škoda Octavia (2006)	KWP 2000 (pomalá inicializace)	\$10 - motor
BMW 530i (2004)	KWP 2000 (rychlá inicializace)	\$12 – motor \$18 – převodovka
BMW 318d (2009)	CAN 11b ID, 500 kb/s	\$7E8 - motor
BMW X1 (2009)	CAN 11b ID, 500 kb/s	\$7E8 - motor \$7EC - převodovka

LITERATURA

- [1] NAVET, N. a SIMON-LION, F. Automotive Embedded Systems Handbook (Industrial Information Technology). CRC Press, 2008. ISBN 978-0-8493-8026-6
- [2] BONNICK, A. Automotive Computer Controlled Systems Diagnostic Tools And Techniques. Butterworth-Heinemann, 2001, ISBN 0-7506-5089-3
- [3] HILLIER, V.A.W. a COOMBES, P. Hilliers Fundamentals of Motor Vehicle Technology Book1. Nelson Thornes Limited, 2004, ISBN 0-7487-8082-3
- [4] BERGER, J. Elektronická regulace vznětových motorů EDC. Robert Bosch GmbH, 2002. ISBN 80-903132-4-8
- [5] Dílenská příručka Škoda auto – Elektrická zařízení FABIA
- [6] RŮŽIČKA, B. Jak na chiptuning. Computer Press, 2007, ISBN 978-80-251-2096-5
- [7] CAN Specification 2.0, Part B
< <http://www.bosch.com> >
- [8] Webové stránky diskuse OSBDM
< <http://forums.freescale.com/freescale/board?board.id=OSBDM08> >
- [9] Webové stránky diagnostického softwaru Scantool
< <http://www.scantool.net> >
- [10] Servis BMW – Informace o produktu: Motor N47,školení poprodejních aktivit, Mnichov, Německo. Listopad 2006
- [11] Servis BMW – Příručka pro účastníky: Motor N52,školení poprodejních aktivit, Mnichov, Německo. Duben 2004
- [12] ISO 14230-1, Keyword Protocol 2000 - Part 1 - Physical Layer, Swedish Implementation Standard, October 22, 1997
- [13] ISO 14230-2, Keyword Protocol 2000 - Part 2 – Data Link Layer, Swedish Implementation Standard, April 22, 1997
- [14] ISO 14230-3, Keyword Protocol 2000 - Part 3 – Application Layer, Swedish Implementation Standard, February 1, 2000
- [15] Webové stránky výrobce obvodu ELM327
< <http://www.elmelectronics.com/DSheets/ELM327DS.pdf> >
- [16] Katalogový list převodníku USB-UART FT232R
<http://www.ftdichip.com/Documents/DataSheets/DS_FT232R.pdf>
- [17] ISO 15765-4, Road vehicles-diagnostic on CAN, Requirements for emissions – related systems, First edition, January 15, 2005
- [18] prezentace Lecture Vehicle Networks od Thomase Stranga a Matthiase Röckla, 2008
- [19] Katalogový list mikrokontroleru MCF51JM64, rev.2, 6/2009, 558 s.
< <http://www.freescale.com> >
- [20] Výrobce hardware a software pro chiptuning EVC Electronic
< www.evc.de >
- [21] Volně dostupná komponenta aGauge pro Visual studio 2008
< http://www.ucancode.net/CSharp_Tutorial_GDI+_Gauge_Source_Code.htm >
- [22] Webové stránky výrobce simulátorů řídicích jednotek
< <http://www.ozenelektronik.com/?s=products2&group=eobd-obdii-ecu-simulators> >
- [23] Informace o produktu Vehicle Network Toolbox
< <http://www.mathworks.com/products/vehicle-network/> >
- [24] ISO 15031-5 Road vehicles-diagnostic Communication between vehicle and external equipment for emissions-related diagnostics, Part 5: Emissions-related diagnostic services
- [25] ISO 15031-6 Road vehicles-diagnostic Communication between vehicle and external equipment for emissions-related diagnostics, Part 6: Diagnostic trouble code definitions
- [26] Volně dostupná komponenta ZedGraph pro Visual studio 2008
<http://sourceforge.net/projects/zedgraph/files/>

SEZNAM ZKRATEK

ABS	Anti-Blocking System
ASIC	Application Specific Integrated Circuit
BDM	Background Debug Module
CAN	Controller Area Network
DTC	Diagnostic Trouble Code
ECU	Electronic Control Unit
ECM	Engine Control Module
EDC	Electronic Diesel Control
EEPROM	Electronic Erasable Programmable Read Only Memory
EGR	Exhaust Gas Recirculation
ESP	Electronic Stability Programme
ISO	International Organization for Standardization
KAM	Keep Alive Memory
KWP	KeyWord Protocol
LED	Light Emitting Diode
LIN	Local Interconnect Network
NRZ	Non Return to Zero
OBD	On Board Diagnostics
OSBDM	Open Source Background Debug Module
OSI	Open System Interconnection
PWM	Pulse Wide Modulation
RAM	Random Access Memory
ROM	Read Only Memory
SAE	Society of Automotive Engineers
SMD	Surface Mounted Device
TCM	Transmission Control Module
USB	Universal Serial Bus
VAG	Volkswagen Audi Group

SEZNAM PŘÍLOH

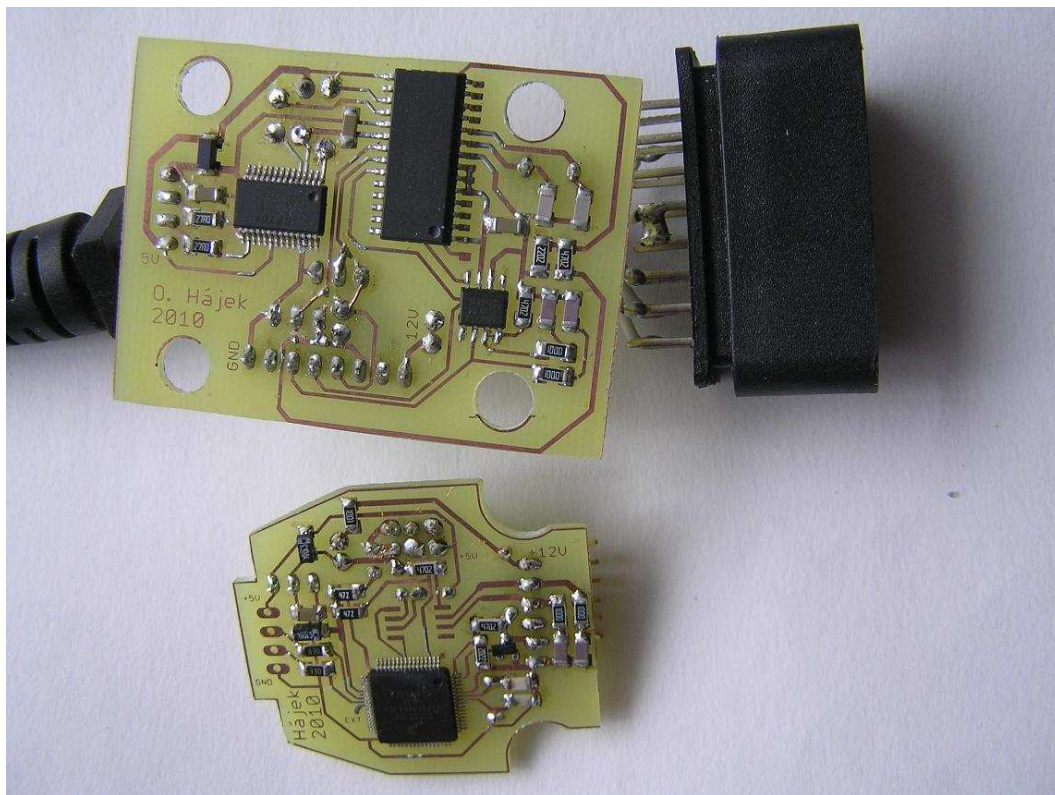
A FOTODOKUMENTACE	- 47 -
A.1 Zhotovené oba návrhy ze strany spojů	- 47 -
A.2 Obě varianty ze strany součástek	- 47 -
A.3 Diagnostické rozhraní s ELM327 v pouzdru	- 48 -
A.4 Rozhraní s ColdFire V1 a programátorem OSBDM	- 48 -

CD obsahující:

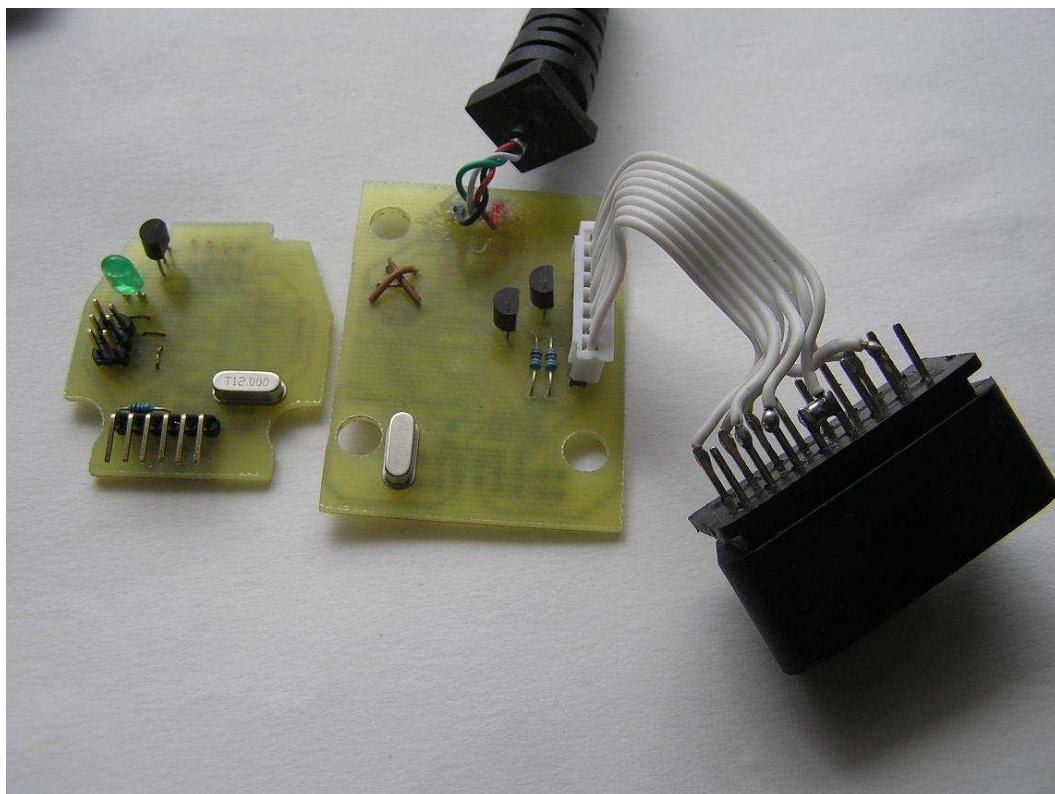
- elektronickou podobu tohoto textu
- vytvořený program včetně zdrojových kódů
- screenshots vytvořené aplikace a videoukázky

A FOTODOKUMENTACE

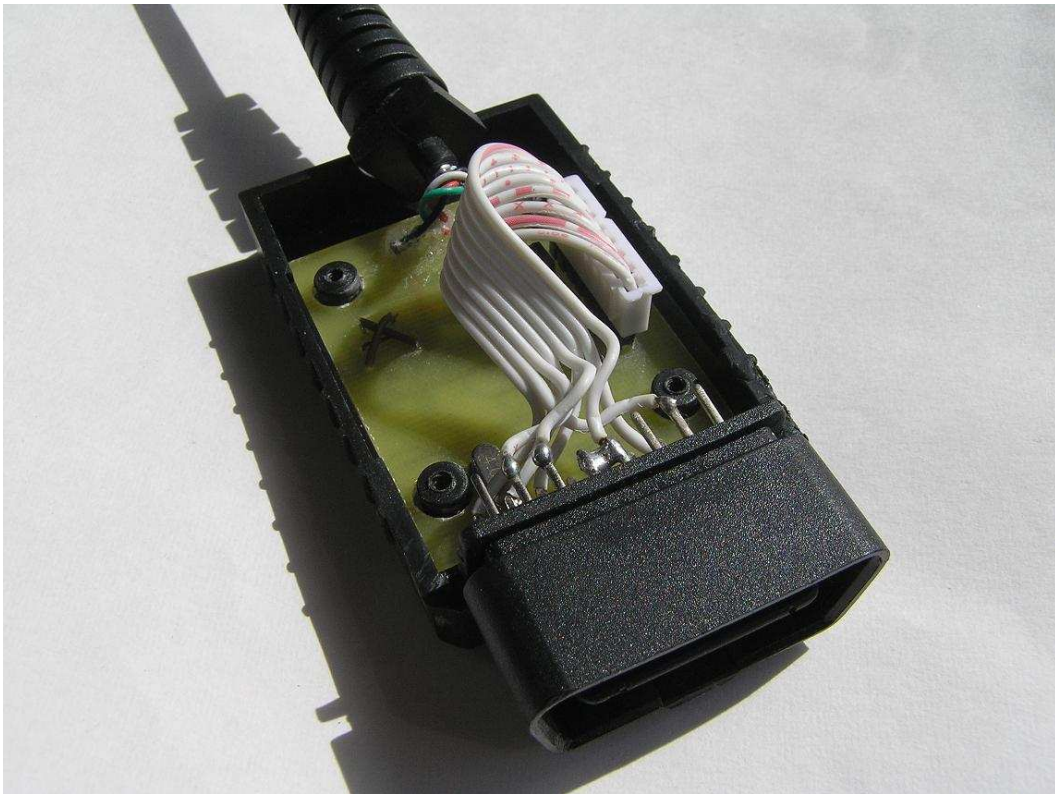
A.1 Zhotovené oba návrhy ze strany spojů



A.2 Obě varianty ze strany součástek



A.3 Diagnostické rozhraní s ELM327 v pouzdru



A.4 Rozhraní s ColdFire V1 a programátorem OSBDM

