

UNIVERZITA PARDUBICE
DOPRAVNÍ FAKULTA JANA PERNERA

BAKALÁŘSKÁ PRÁCE

2024

Petr Stejskal

Univerzita Pardubice
Dopravní fakulta Jana Pernera

Převodník diagnostického protokolu na CAN BUS

Bakalářská práce

Univerzita Pardubice
Dopravní fakulta Jana Pernera
Akademický rok: 2023/2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Petr Stejskal**
Osobní číslo: **D21629**
Studijní program: **B0788A040001 Dopravní technika**
Specializace: **Elektrická trakce a elektromobilita**
Téma práce: **Převodník diagnostického protokolu na CAN BUS**
Zadávající katedra: **Katedra elektrotechniky, elektroniky a zabezpečovací techniky v dopravě**

Zásady pro vypracování

Vytvořte zařízení pro přeposílání diagnostických dat vyčítaných z OBD zásuvky vozidla po sběrnici CAN protokolem UDS na další sběrnici CAN bez protokolu (pouze surová data uložená do CAN zpráv s definovaným ID).

Pro zrychlení vývoje použijte hardware Arduino nebo podobný. Vhodným HW je např. stavebnice M5stack (<https://m5stack.com/>).

Težištěm bude vytvoření SW v jazyku C/C++ pro mikrokontrolér. Při vývoji je možné použít softwarové knihovny třetích stran.

Komunikaci v UDS protokolu není potřeba programovat, předpokládá se použití bluetooth OBD adaptéru OBDLink, který sám zajišťuje komunikaci s vozidlem po CAN BUS pomocí UDS protokolu. Převodník bude komunikovat s adaptérem OBDLink přes bluetooth pomocí textových zpráv.

Rozsah pracovní zprávy:

Rozsah grafických prací:

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

Seznam doporučené literatury:

1. VOSS, Wilfried. A comprehensible guide to controller area network. Greenfield: Cooperhill Technologies Corporation, [2005]., 150 s. ISBN 0976511606.
2. *ISO 14229 Road vehicles — Unified diagnostic services (UDS)*
3. *ISO 15031 Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics.*
4. Datasheety výrobců zařízení a komponent.
5. Podklady poskytnuté vedoucím bakalářské práce.

Vedoucí bakalářské práce:

Ing. Zdeněk Mašek, Ph.D.

Katedra elektrotechniky, elektroniky a zabezpečovací techniky v dopravě

Datum zadání bakalářské práce: **24. ledna 2024**

Termín odevzdání bakalářské práce: **13. května 2024**

L.S.

doc. Ing. Libor Švadlenka, Ph.D.
děkan

Ing. Vítězslav Krčmář, Ph.D.
vedoucí katedry

Prohlášení autora

Práci s názvem Převodník diagnostického protokolu na CAN BUS jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 10.5.2024

Petr Stejskal v. r.

Poděkování

Tímto chci poděkovat vedoucímu mé bakalářské práce, Ing. Zdeňku Maškovi, Ph. D., za jeho ochotu, trpělivost a důslednost při vytváření této práce. Za zodpovězení každého z mých dotazů a za spolupráci v průběhu konzultací i mimo ně.

Velké díky také patří mé rodině, mým přátelům a blízkým. V neposlední řadě děkuji také mému domovskému oddílu Pathfinder č. 130 B.Z.Z.Z. Všem těmto lidem patří srdečné díky za jejich velkou podporu v průběhu celého mého studia.

Anotace

Bakalářská práce se zabývá návrhem a testováním programu pro převodník, který vyčítá z vozidla diagnostická data ze zásuvky OBD pomocí protokolu UDS. Ty poté přeposílá na další sběrnici CAN, ale bez protokolu. Odesílá tedy pouze zprávy s definovaným ID.

Klíčová slova

CAN bus, protokol UDS, Mikroprocesor ESP32, převodník CAN, EOBD

Title

CAN bus diagnostic protocol converter

Abstract

The bachelor thesis deals with the design and testing of a converter program that reads diagnostic data from the vehicle's OBD socket using the UDS protocol. It then forwards them to the next CAN bus, but without the protocol. It sends only messages with a defined ID.

Keywords

CAN bus, protocol UDS, Mikroprocesor ESP32, CAN converter, EOBD

Obsah

Seznam značek a zkratk	10
1 Úvod.....	12
2 Teoretický rozbor	13
2.1 Sběrnice CAN bus.....	13
2.1.1 Linková vrstva	14
2.2 Sériová diagnostika	15
2.2.1 EOBD.....	16
2.2.2 Diagnostické služby EOBD	17
2.2.3 Funkce sériové diagnostiky nad rámec EOBD.....	17
2.2.4 Komunikace s vozidlem	21
3 Požadavky na převodník	26
4 Hardware zařízení	28
4.1 Základní návrh převodníku	28
4.2 Stavebnice M5Stack CORE BASIC	30
4.2.1 Mikrokontroler ESP32.....	32
4.3 COMMU modul.....	32
4.4 Napájení převodníku	34
4.4.1 Napájení pomocí kabelu USB	34
4.4.2 Použití baterie	35
4.4.3 Napájecí modul.....	35
4.5 OBD adaptér	39
4.5.1 Komunikace s OBD adaptérem	40
5 Konfigurační soubor	42
5.1 Formát souboru.....	42
5.2 Vlastnosti načítání souboru	47
6 Návrh software.....	51
6.1 Obsluha převodníku	52
6.2 Bluetooth.....	53

6.3	Vyčítání dat.....	59
6.4	Odesílání zpráv na CAN bus	68
7	Ověření funkce zařízení	71
7.1	Emulátor vozidla	72
7.2	Vozidlo	76
8	Souhrnné vlastnosti převodníku.....	81
9	Závěr	82
	Literatura.....	83
	Seznam příloh.....	85

Seznam značek a zkratk

Latinská písmena

U_{USB}	[V]	Napájecí napětí převodníku kabelem USB typ C
VCC_{5V}	[V]	Napájecí napětí 5 V
VCC_{3V}	[V]	Napájecí napětí 3,3 V pro mikrokontroler ESP32
VDD_{3V}	[V]	Napěťová hladina 3,3 V
U_{BAT}	[V]	Napájecí napětí vestavěného akumulátoru

Zkratky

CAN	Controller area network
UDS	Unified diagnostic services
OBD	On-board diagnostics
TFT	Thin film transistor
LCD	Liquid crystal display
USB	Universal serial bus
GPIO	General-purpose input/output
I2C	Inter-integrated circuit
UART	Universal asynchronous receiver-transmitter
SoC	System on chip
DMIPS	Dhrystone million instructions per second
ROM	Read only memory
SRAM	Static random access memory
RTC	Real-time clock
SAR	Successive approximation register
IoT	Internet of things
MCU	Microcontroller unit
SPI	Serial peripheral interface
TTL	Transistor-transistor-logic
LED	Light-emitting diode
SOC	State of charge
ISO	International organization for standardization
OSI	Open systems interconnection
MIL	Malfunction indicator lamp
PID	Parameter ID
DID	Data ID
SID	Service ID
ECU	Electronic control unit

VIN	Vehicle identification number
DTC	Diagnostic trouble code
BT	Bluetooth
SW	Software

1 Úvod

Tato práce se zabývá vytvořením zařízení pro přeposílání diagnostických dat, která jsou vyčítána z OBD zásuvky vozidla po sběrnici CAN, protokolem UDS, na další sběrnici CAN, ale bez protokolu.

Zadáním je, aby toto zařízení dokázalo vyčítat libovolná diagnostická data z řídicích jednotek vozidla, která lze vyčíst na zásuvce OBD. Takto vyčtená data potřebuje ke svojí funkci tzv. dash logger, který sice dokáže vyčítat diagnostická data z vozidla, ale pouze v rámci standardu EOBD, tudíž má velice omezené komunikační schopnosti. A tedy tuto funkci zastane takovéto zařízení – převodník, který bude převádět data z jedné sběrnice CAN s protokolem UDS, na další sběrnici, bez protokolu.

Jelikož se jedná o vytvoření jednoúčelového zařízení, nebyla zvolena cesta kompletního návrhu, výběru součástek a sestavení zařízení (kvůli časové náročnosti, estetičnosti a kvůli ceně celého zařízení). Pro realizaci byla vybrána sériově vyráběná stavebnice M5Stack. Tato stavebnice umožňuje „stackovat“ na sebe různé další moduly (v tomto případě hlavně komunikační modul, který zajišťuje odesílání CAN zpráv). Zprávy z diagnostické zásuvky jsou vyčítány pomocí Bluetooth OBD adaptéru OBDLink. Tento adaptér byl zvolen kvůli zjednodušení komunikace mezi zařízením a vozidlem, protože zajišťuje veškerou komunikaci po sběrnici CAN s vozidlem, s ohledem na použitý protokol UDS, a uživateli zpřístupňuje možnost pracovat pouze s užitečnými daty CAN zpráv.

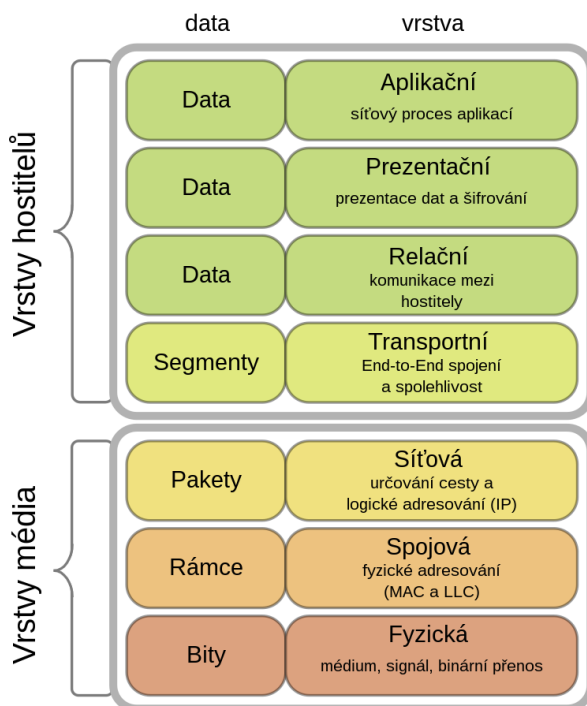
2 Teoretický rozbor

V této kapitole je popsána sběrnice CAN bus a problematika ohledně standardu EOBD a protokolu UDS.

2.1 Sběrnice CAN bus

CAN (Controller area network) je sériová komunikační sběrnice pro aplikace běžící v reálném čase. Tato sběrnice, vyvinutá firmou BOSCH v roce 1986, se stala standardem v oblasti automobilní techniky, nicméně ji lze najít i v kolejových vozidlech, zdravotní technice, v letectví a v neposlední řadě se používá v oblasti průmyslové automatizace [9].

Sběrnice zajišťuje sdílení informací mezi řídicími jednotkami vozidla. Jedná se o poloduplexní asynchronní sběrnici. Dosahuje maximální přenosové rychlosti až 1 Mbit/s na vzdálenost 40 m. Po této sběrnici se přenáší krátké zprávy, kde každá z nich může obsahovat 0 až 8 bajtů užitečných dat. Řídí se mezinárodním standardem ISO 11898. Tento standard ale definuje pouze linkovou neboli spojovou (ISO 11898-1), a fyzickou (ISO 11898-2) vrstvu tzv. ISO/OSI síťového modelu (Obr. 1) [22].



Obr. 1 – Síťový model ISO/OSI. Zdroj [21].

Tento model definuje vrstvy komunikace a je základnou pro komunikační protokoly, které definují způsoby propojování systémů mezi sebou. Vyšší vrstvy ISO/OSI modelu jsou využívány až protokoly vyšších vrstev, jako např. KWP2000 (ISO 14230-3), CAN OBD (ISO

15031-5), CANopen (EN 50325-4) nebo právě protokol UDS (ISO 14229-3), používaný např. koncernem VW, nebo ve standardu WWH-OBD (ISO 27145).

Fyzická a linková vrstva standardu ISO 11898 jsou implementované v hardwaru sběrnice CAN. Linkovou vrstvu obstarává tzv. řadič CAN a fyzickou vrstvu realizuje tzv. budič CAN.

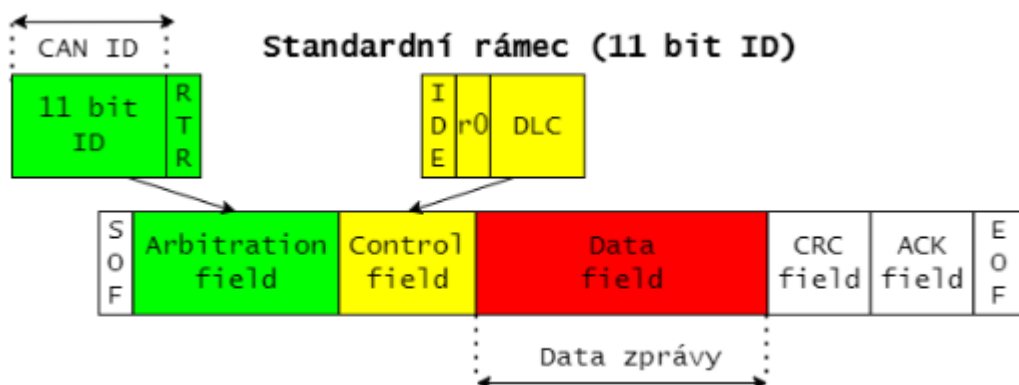
2.1.1 Linková vrstva

Spojová neboli linková vrstva definuje strukturu zpráv (rámců) přenášených po sběrnici. Zajišťuje vložení, resp. vyjmutí dat do, resp. z komunikačního rámce. Generuje přenosovou rychlost, synchronizuje přijímače s vysílačem. Řídí přístup uzlu ke sběrnici, filtruje zprávy přijaté fyzickou vrstvou, detekuje a signalizuje chyby na sběrnici.

Libovolný uzel sběrnice může kdykoliv zahájit komunikaci, díky čemuž může docházet ke kolizím (dva nebo více uzlů se snaží vysílat zprávu v jeden okamžik). Priorita zprávy rozhoduje o tom, jaký uzel bude moci pokračovat ve vysílání v případě kolize. Priorita zprávy je dána identifikátorem zprávy (CAN ID). Zpráva s nižší hodnotou CAN ID má přednost před zprávou s vyšší hodnotou CAN ID, vyhrává arbitráž a pokračuje v odesílání. Uzel, který arbitráž prohrál, změní svůj stav z odesílání na příjem.

Na sběrnici CAN bus je možné najít několik typů zpráv (message frames). Nejdůležitějšími typy jsou:

- Datová zpráva (Data frame)
- Žádost o data (Remote frame)
- Chybová zpráva (Error frame)

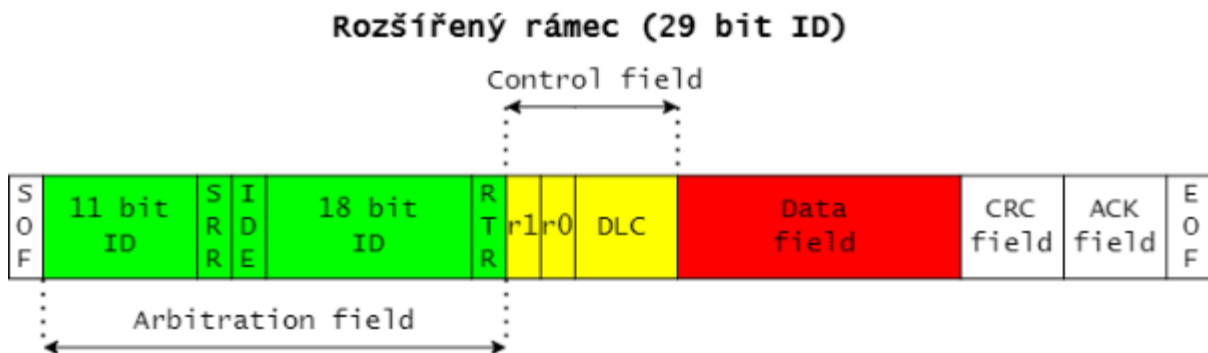


Obr. 2 – Datová zpráva. Standardní rámeček (11 bit ID).

Datová zpráva se používá pro přenos užitečných dat po sběrnici (Obr. 2). Nejdůležitějšími poli v této zprávě jsou Arbitration field (CAN ID) a Data field (data zprávy). Ostatní pole jsou zde nutná pro zajištění správného přenosu zprávy.

Žádost o data je požadavek na odeslání datové zprávy. Svoji strukturou je stejná jako datová zpráva. Jediným rozdílem je absence pole Data field.

Chybová zpráva slouží pro signalizaci chyb na sběrnici a zajišťuje, aby zpráva s defektem nebyla přijata žádným uzlem.



Obr. 3 – Datová zpráva. Rozšířený rámeček (29 bit ID).

Na sběrnici nachází dva typy datových rámečků. Jeden, už zmíněný, je standardní rámeček (Obr. 2). Druhý typ je tzv. rozšířený rámeček (Obr. 3). Liší se počtem bitů, které je pro CAN ID možné použít. Standardní rámeček obsahuje 11 bit CAN ID, rozšířený (extended) obsahuje 29 bit CAN ID. Důvodem zavedení rozšířeného rámečku bylo navýšení počtu unikátních CAN ID. Oba typy rámečků lze na sběrnici provozovat současně.

Zda se jedná o standardní rámeček, či rozšířený rámeček rozhoduje hodnota bitu IDE. Pokud se hodnota bitu IDE rovná nule, tak se jedná o standardní rámeček, pokud se hodnota rovná jedné, tak se jedná o rozšířený rámeček. Z toho vyplývá, že zpráva se standardním rámečkem bude mít nižší hodnotu CAN ID (vyšší prioritu) a tedy vyhraje arbitráž proti zprávě s rozšířeným rámečkem.

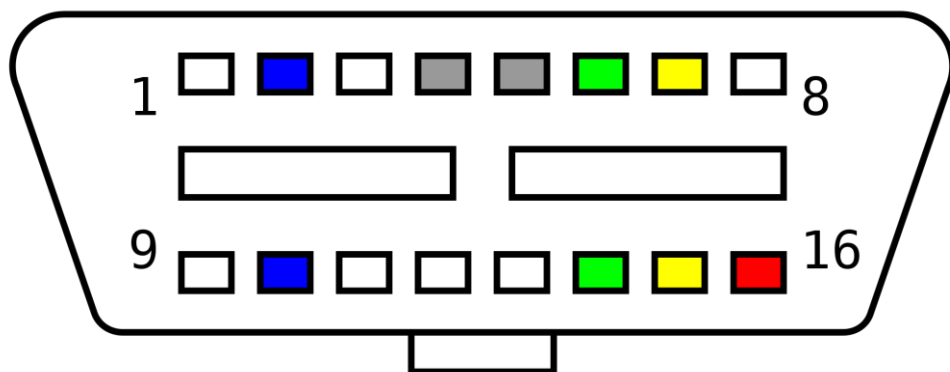
2.2 Sériová diagnostika

V současné době jsou vozidla vybavena velkým počtem řídicích jednotek a k nim připojených akčních členů a senzorů. Diagnostika vozidel se dá rozdělit na sériovou a paralelní diagnostiku.

Sériová diagnostika je založena na komunikaci diagnostického testeru s řídicími jednotkami vozidla. Pomocí tohoto typu diagnostiky je možné vyčítat paměť závad v řídicí jednotce, diagnostikovat akční členy a jejich uvedení do základního nastavení, nebo např. vyčítat měřené hodnoty senzorů dané řídicí jednotky.

2.2.1 EOBD

Je to systém samokontroly dané řídicí jednotky vozidla, který sleduje její správnou funkci (a funkci všech k ní připojených komponent). Tento systém umožňuje kontrolu a sledování dílů, které se podílejí na složení výfukových plynů, protože přímé měření emisí výfukových plynů není možné během jízdy vozidla. Vychází se z toho předpokladu, že obsah škodlivin ve výfukových plynech bude nízký, když součásti, které se podílejí na jejich vzniku, budou pracovat bez závad [23].



Obr. 4 – Diagnostická zásuvka. Konektor typ B. Zdroj [24].

V případě výskytu poruchy na takové součásti se chyba uloží do paměti závad a dojde ke světelné signalizaci řidiči vozidla rozsvícením kontrolky MIL. Tento systém umožňuje vyčtení paměti závad a provozních podmínek, při kterých tato závada nastala. Vyčtení je možné pomocí libovolného diagnostického testeru, který lze připojit k normalizované 16-pinové zásuvce (Obr. 4). Tato zásuvka se musí nacházet uvnitř vozidla v dosahu ze sedadla řidiče.

Tab. 1 – Zapojení pinů diagnostické zásuvky. Viz Obr. 4.

Zapojení pinů			
1	Dle výrobce	9	Dle výrobce
2	SAE J1850	10	SAE J1850
3	Dle výrobce	11	Dle výrobce
4	Kostra	12	Dle výrobce
5	Signálová zem	13	Dle výrobce
6	CAN HIGH	14	CAN LOW
7	K-Line	15	L-line
8	Dle výrobce	16	Napětí baterie (+12/ +24 V)

2.2.2 Diagnostické služby EOBD

Komunikační parametry rozhraní OBD jsou definovány standardem SAE J1979 / ISO 15031-5. Pomocí OBD lze vyčítat pouze emisně relevantní data, tedy lze diagnostikovat systémy řízení motoru a úpravy spalín.

Tento standard definuje diagnostické služby (SID) neboli módy. Tyto módy jsou určeny pro vyčítání dat z řídicích jednotek a pro spouštění diagnostických funkcí. Standard definuje několik módů, viz Obr. 5.

6	Diagnostic service definition for ISO 9141-2, ISO 14230-4, and SAE J1850	31
6.1	Service \$01 — Request current powertrain diagnostic data	31
6.2	Service \$02 — Request powertrain freeze frame data	35
6.3	Service \$03 — Request emission-related diagnostic trouble codes	39
6.4	Service \$04 — Clear/reset emission-related diagnostic information	44
6.5	Service \$05 — Request oxygen sensor monitoring test results	46
6.6	Service \$06 — Request on-board monitoring test results for specific monitored systems	51
6.7	Service \$07 - Request emission-related diagnostic trouble codes detected during current or last completed driving cycle	56
6.8	Service \$08 — Request control of on-board system, test or component	57
6.9	Service \$09 — Request vehicle information	60

Obr. 5 – Diagnostické služby/ módy poskytované systémem OBD. Zdroj [2].

Pomocí těchto služeb lze např. žádat o „live-data“ řídicí jednotky motoru (\$01), číst a mazat chybové kódy (\$03 a \$04), nebo vyčítat informace o vozidle (\$09) jako třeba VIN.

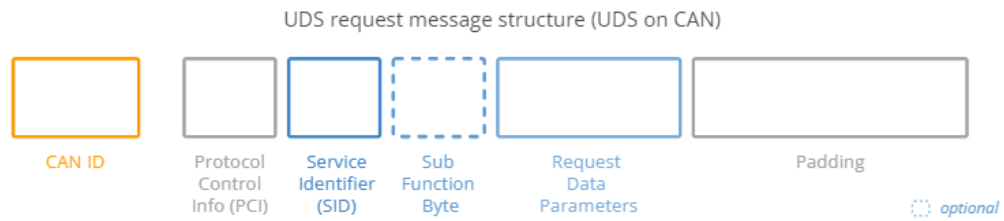
2.2.3 Funkce sériové diagnostiky nad rámec EOBD

Pouhé používání sériové diagnostiky v rámci dle EOBD je nedostatečné. EOBD umožňuje pouze komunikaci s řídicí jednotkou motoru, a to v omezeném rozsahu (pouhá správa paměti závad a vyčítání emisně relevantních dat). Pro efektivní používání sériové diagnostiky je třeba komunikovat na vyšší úrovni a se všemi řídicími jednotkami vozidla.

Řídicí jednotky ve vozidlech poskytují velký rozsah funkcí a služeb nad rámec standardu EOBD. To umožňuje rychlejší a spolehlivější nalezení závady a tím její následné odstranění. Jedná se například o čtení měřených živých dat a testy akčních členů a funkcí.

Např. protokol UDS, definovaný normou ISO 14229, poskytuje velké množství takových služeb. Příklad služeb tohoto protokolu je vidět na Obr. 8 a na Obr. 10. Struktura zprávy typu žádosti o data protokolu UDS je vidět na Obr. 6.

Dále je uveden význam jednotlivých polí této struktury (zdroj [25]).



Obr. 6 – Struktura zprávy typu žádost o data protokolu UDS. Zdroj [25].

PCI

Může být jedno až tři bajty dlouhé. Obsahuje informace o typu rámce (Single frame, first frame, consecutive frame) a podle typu rámce pak informaci o počtu následujících užitečných datových bajtů přenášených ve zprávě (v případě single frame a first frame) nebo čísla rámce v multipaketu v případě consecutive. Podrobněji v ISO 15765-2.

SID

Toto pole obsahuje jednu ze služeb poskytovanou protokolem UDS (příklad na Obr. 8 a Obr. 10). Tyto ID jsou odlišné při žádosti (např. 0x22) a odpovědi (pro 0x22 je to 0x62) [25].

Významnými službami, poskytované tímto protokolem, jsou např. služby 0x19 (nenachází se na Obr. 8) a 0x22. Služba 0x19 neboli ReadDTCInformation, poskytuje, podobně jako služby \$03 a \$07 standardu EOBD, informace o chybových kódech řídicích jednotek vozidla. Služba 0x20, tzv. ReadDataByIdentifier, nabízí vyčtení různých dat, veličin a stavů z řídicích jednotek vozidla. Podobně jako služba \$01 standardu EOBD, ale místo PIDů se zde používají DIDy (Data ID), viz Obr. 9).

Nevýhodou žádostí o data nad rámec standardu EOBD je skutečnost, že význam dat DIDů příslušných SIDů si stanovuje výrobce vozidla a často se liší u různých motorizací jedné řady, nebo dokonce u jednotlivých výrobců řídicích jednotek. Proto se bez dalších dokumentů nebo znalostí nedá spolehlivě určit význam dat poskytovaných těmito službami.

UDS services - sub function types			
UDS SID (request)	UDS SID (response)	Service	Sub function types
0x10	0x50	Diagnostic Session Control	Diagnostic session type
0x11	0x51	ECU Reset	Reset type
0x27	0x67	Security Access	Security access type
0x28	0x68	Communication Control	Control type
0x3E	0x7E	Tester Present	"Zero sub function"
0x83	0xC3	Access Timing Parameters	Timing parameter access type
0x85	0xC5	Control DTC Settings	DTC setting type
0x86	0xC6	Response On Event	Event type
0x87	0xC7	Link Control	Link control type
0x2C	0x6C	Dynamically Define Data Identifier	Definition type
0x19	0x59	Read DTC Information	Report type
0x31	0x71	Routine Control	Routine control type

Obr. 7 – Seznam SIDů, u kterých se používá Sub Function Byte. Zdroj [25].

Sub Function Byte

Tento bajt je použit pouze v některých zprávách typu žádosti o data, viz Obr. 7.

Obecně může být odpovědí buď positive, nebo negative response. V případě že je odpověď kladná, tak může chtít diagnostický tester odpověď potlačit (protože může být irelevantní). To se provádí nastavením prvního bitu tohoto bajtu na jedničku. Negativní odpovědi nelze potlačit. Zbýlých sedm bitů se používá pro definování hodnoty Sub Function bajtu [25].

9	Diagnostic and Communication Management functional unit	35
9.1	Overview.....	35
9.2	DiagnosticSessionControl (0x10) service.....	36
9.3	ECUReset (0x11) service	43
9.4	SecurityAccess (0x27) service.....	47
9.5	CommunicationControl (0x28) service.....	53
9.6	TesterPresent (0x3E) service	58
9.7	AccessTimingParameter (0x83) service.....	61
9.8	SecuredDataTransmission (0x84) service	66
9.9	ControlDTCSetting (0x85) service	71
9.10	ResponseOnEvent (0x86) service.....	75
9.11	LinkControl (0x87) service.....	99
10	Data Transmission functional unit	106
10.1	Overview.....	106
10.2	ReadDataByIdentifier (0x22) service	106
10.3	ReadMemoryByAddress (0x23) service.....	113
10.4	ReadScalingDataByIdentifier (0x24) service	119
10.5	ReadDataByPeriodicIdentifier (0x2A) service	126
10.6	DynamicallyDefineDataIdentifier (0x2C) service	140
10.7	WriteDataByIdentifier (0x2E) service.....	162
10.8	WriteMemoryByAddress (0x3D) service	167

Obr. 8 – Některé služby poskytované protokolem UDS. Zdroj [1].

Request Data Parameters

U většiny služeb SID se pro upřesnění požadavku, mimo použití Sub Function Byte, používají další různé parametry [25]. Např. pro službu 0x19, která poskytuje informace o chybových kódech. Lze zadávat jednobajtový parametr, tzv. DTC Status Mask.

UDS - standardized data identifiers (DID)	
UDS DID (data identifier)	Description
0xF180	Boot software identification
0xF181	Application software identification
0xF182	Application data identification
0xF183	Boot software fingerprint
0xF184	Application software fingerprint
0xF185	Application data fingerprint
0xF186	Active diagnostic session
0xF187	Manufacturer spare part number
0xF188	Manufacturer ECU software number
0xF189	Manufacturer ECU software version
0xF18A	Identifier of system supplier

Obr. 9 – Příklad některých standardizovaných DIDů protokolu UDS. Zdroj [25].

Pro SID 0x22 lze zadávat tzv. DID. Jedná se o dvojbajtový parametr. Významem je tento parametr podobný PIDům u služeb \$01 standardu EOBD. Příklad těchto DIDů a jejich významu se nachází na Obr. 9

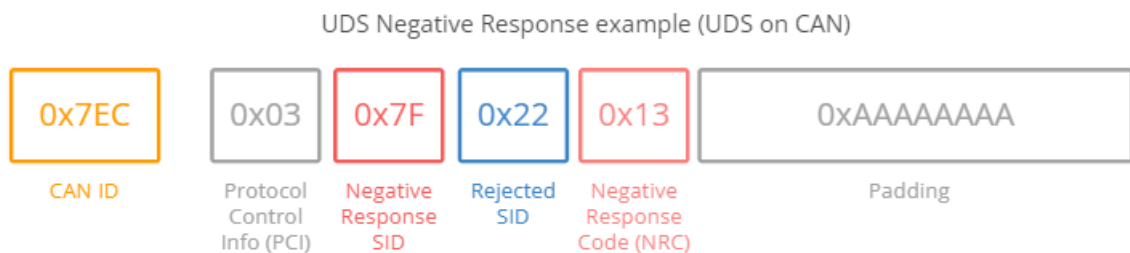
UDS service identifiers (SIDs)			
UDS SID (request)	UDS SID (response)	Service	Details
0x10	0x50	Diagnostic Session Control	Control which UDS services are available
0x11	0x51	ECU Reset	Reset the ECU ("hard reset", "key off", "soft reset")
0x27	0x67	Security Access	Enable use of security-critical services via authentication
0x28	0x68	Communication Control	Turn sending/receiving of messages on/off in the ECU
0x29	0x69	Authentication	Enable more advanced authentication vs. 0x27 (PKI based exchange)
0x3E	0x7E	Tester Present	Send a "heartbeat" periodically to remain in the current session
0x83	0xC3	Access Timing Parameters	View/modify timing parameters used in client/server communication
0x84	0xC4	Secured Data Transmission	Send encrypted data via ISO 15764 (Extended Data Link Security)
0x85	0xC5	Control DTC Settings	Enable/disable detection of errors (e.g. used during diagnostics)
0x86	0xC6	Response On Event	Request that an ECU processes a service request if an event happens
0x87	0xC7	Link Control	Set the baud rate for diagnostic access
0x22	0x62	Read Data By Identifier	Read data from targeted ECU - e.g. VIN, sensor data values etc.
0x23	0x63	Read Memory By Address	Read data from physical memory (e.g. to understand software behavior)

Obr. 10 – Některé služby (SIDy) poskytované protokolem UDS. Zdroj [25].

Positivní a negativní odpověď protokolu UDS

Formát pozitivní odpovědi je podobný, jako formát žádosti o data (Obr. 6). Např odpovědi na službu 0x22 bude v parametru SID hodnota 0x62 (0x22 + 0x40) a dále dvojbajtový DID. Následovat budou bajty s užitečnými daty s takovým významem, který odpovídá zadanému SID a DID.

Ovšem v některých případech může dojít k odpovědi typu negative response (Obr. 11).



Obr. 11 – Struktura odpovědi z ECU typu negative response. Zdroj [25].

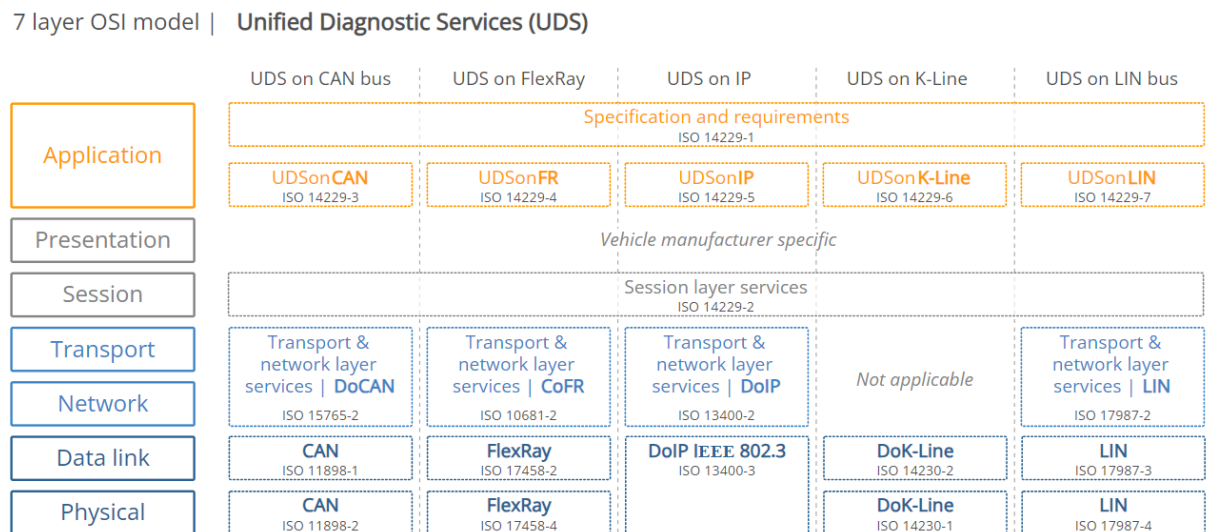
Mimo CAN ID, je prvním bajtem pole PCI. Následuje bajt Negative Response SID (0x7F), třetí bajt obsahuje hodnotu SIDu, jehož žádost byla zamítnuta. Čtvrtým bajtem je tzv. Negative Response Code neboli NRC [25]. Příklad těchto NRC se nachází na Obr. 12.

UDS SID 0x7F - Negative Response Codes (NRC)	
UDS NRC	Description
0x10	General reject
0x11	Service not supported
0x12	Sub-function not supported
0x13	Invalid message length/format
0x14	Response too long
0x21	Busy-repeat request
0x22	Conditions not correct
0x24	Request sequence error
0x25	No response from subnet component
0x26	Failure prevents execution of requested action
0x31	Request out of range
0x33	Security access denied

Obr. 12 – Seznam vybraných kódů NRC. Zdroj [25].

OSI model CAN bus versus UDS

Na následujícím obrázku (Obr. 13) je vidět, jak standard UDS koresponduje se sběrnici CAN bus a s ISO/OSI síťovým modelem. V případě UDS se tedy jedná o protokol vyšší vrstvy. Obsahuje pátou a sedmou (relační a aplikační) vrstvu ISO/OSI modelu.



Obr. 13 – Zasazení UDS do kontextu s CAN bus a ISO/OSI síťovým modelem. Zdroj [25].

2.2.4 Komunikace s vozidlem

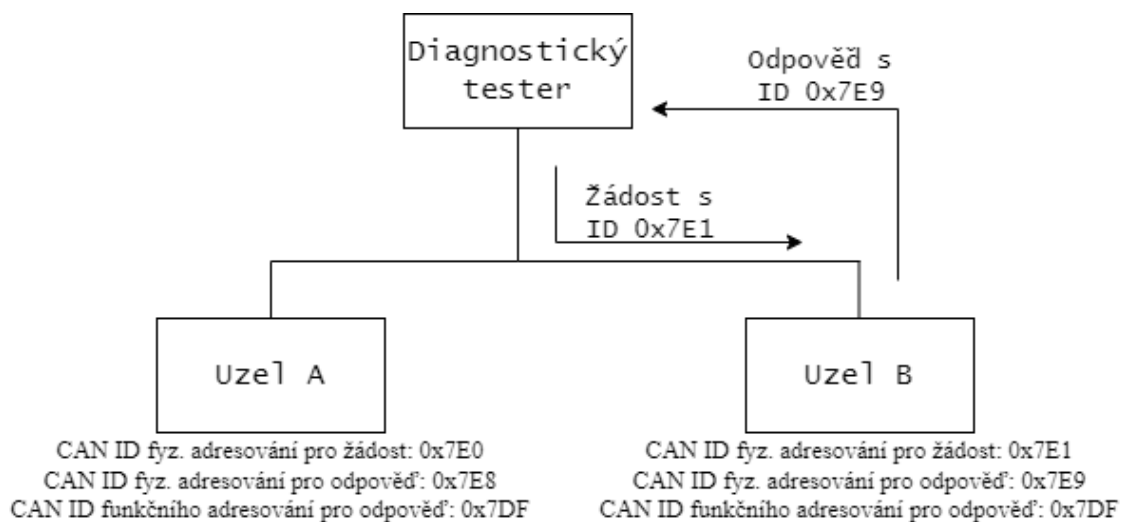
V této podkapitole je představeno, jak použít OBD adaptér ke komunikaci s vozidlem. Základní komunikace s OBD adaptérem a ukázka několika jeho možností a nastavení se nachází v podkapitole 4.5.1

V případě žádostí o data na sběrnici CAN bus je nutné rozlišovat mezi dvěma typy adresování.

Fyzické adresování

Toto adresování je použito v případě, když chce např. diagnostický tester přímo komunikovat s nějakým uzlem (uzlem B). Je to tedy komunikace pouze mezi dvěma uzly (Obr. 14).

Každému uzlu jsou přiřazena ID zprávy CAN pro fyzické adresování, které poté uzel používá pro komunikaci na sběrnici CAN bus. Tyto ID jsou dvě a jejich výskyt na sběrnici CAN závisí na tom, jestli se jedná o žádost o data od diagnostického testeru (OBD adaptéru), nebo odpověď daného uzlu.

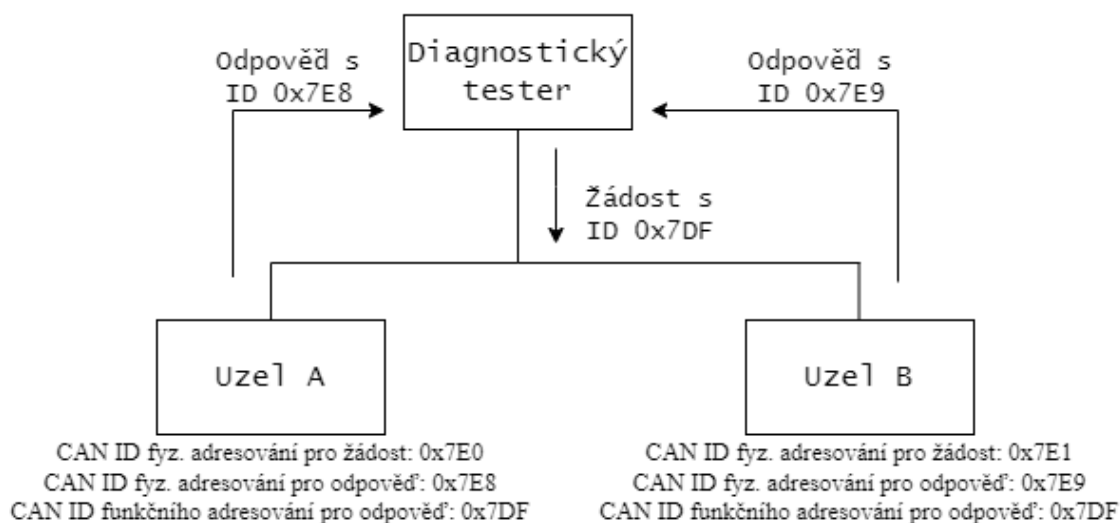


Obr. 14 – Ukázka žádosti o data. Fyzické adresování.

Např. v případě žádosti o data může diagnostický tester požadovat data pouze od uzlu B. V tom případě bude vysílat zprávy CAN s takovým ID, které odpovídá fyzické adrese pro žádosti o data uzlu B. Fyzická adresa pro žádosti tohoto uzlu je 0x7E1. Tuto zprávu zpracuje pouze uzel B a na tuto žádost patřičně odpoví. ID zprávy této odpovědi bude 0x7E9.

Funkční adresování

Umožňuje to, aby byly žádosti kategorizovány podle významu jejich funkce (např. informaci o aktuálních otáčkách motoru může poskytovat řídicí jednotka motoru i řídicí jednotka automatické převodovky). Toto adresování není závislé na adresách pro žádosti jednotlivých uzlů. Diagnostický tester odešle zprávu CAN s významem žádost o data. CAN ID této žádosti je pro externí diagnostické testery vyhrazeno jako 0x7DF (Obr. 15).



Obr. 15 – Ukázka žádosti o data. Funkční adresování.

Na tuto žádost můžou odpovědět všechny uzly, které žádaná data mohou poskytnout. V tomto případě jsou to uzly A a B. Uzly v tomto případě odesílají zprávu CAN s takovým ID, které odpovídá jejich CAN ID pro odpověď při fyzickém adresování. Je to tedy komunikace diagnostického testeru s minimálně jedním uzlem na sběrnici CAN.

Příklady komunikace

V rámci protokolu EOBD lze pomocí zde použitého OBD adaptéru žádat o emisně relevantní data způsobem, který je zobrazen na Obr. 16. Formát zprávy přijatých dat z Obr. 16 je zobrazen na Obr. 17. Modře jsou zde zobrazeny příkazy pro OBD adaptér (ELM327 kompatibilní), zeleně jsou poté odpovědi adaptéru.

První byte AT příkazu reprezentuje číslo služby (módu), tedy v tomto případě se jedná o službu \$01: Měřené hodnoty. Druhý potom reprezentuje číslo PIDu této žádosti. Zadaný PID 0x00 je jedním z PIDů, které musí podporovat každé vozidlo. Jeho význam je do 4 bajtů bitově zakódována informace o tom, které další PIDy od čísla 0x00 do čísla 0x20 jsou podporovány vozidlem. V této informaci hodnota daného bitu rovna jedné znamená, že daný PID na dané pozici podporovaný je, hodnota bitu rovna nule znamená opak.

```
12:58:10.922 01 00
12:58:11.342 41 00 80 1A 80 01
12:58:11.526
12:58:11.526 >
```

Obr. 16 – Komunikace s emulátorem vozidla. Funkční adresování.

Číslo služby + 0x40 (positive response)	Zopakování zadaného PIDu	Poskytnutá data
0x41	0x00	0x80 x01A 0x80 0x01

Obr. 17 – Formát přijatých dat z Obr. 16.

Pomocí příkazu AT H1 si lze nechat zobrazit tzv. header. Header je ID zprávy CAN a jeho hodnota závisí na módu adresování a na směru komunikace. Na Obr. 18 je vidět odpověď uzlu i s patřičným CAN ID, tedy 0x7E8. Dále je zde bajt PCI neboli Protocol Control Info (0x06). Dále pokračuje stejná odpověď, jako na Obr. 16 (0x41 0x00 0x80 0x1A 0x80 0x01). Formát této zprávy je zobrazen na Obr. 19.

```
13:23:08.989 AT H1
13:23:09.422 OK
13:23:09.422
13:23:09.422 >13:23:14.791 01 00
13:23:15.215 7E8 06 41 00 80 1A 80 01
```

Obr. 18 – Komunikace s emulátorem vozidla. Funkční adresování. Zobrazení CAN ID zprávy.

Header (CAN ID)	PCI (počet užitečných bajtů ve zprávě)	Číslo služby + 0x40 (positive response)	Zopakování zadaného PIDu	Poskytnutá data
0x7E8	0x06	0x41	0x00	0x80 x01A 0x80 0x01

Obr. 19 – Formát přijatých dat z Obr. 18.

Při použití příkazu pouze 01 00 je použito funkční adresování, protože má tento adaptér nastavený header žádosti na 0x7DF jako výchozí nastavení. A je tedy možné, že mohou přijít odpovědi od více uzlů na žádost danou kombinací SID a PID. Na Obr. 20 jsou vidět dvě odpovědi s CAN ID 0x7E8 (motor) a 0x7E9 (automatická převodovka). Jsou to odpovědi na žádost o aktuálních otáčkách motoru při použití funkčního adresování.

```
17:02:06.069 >17:02:51.724 01 0c
17:02:52.214 7E8 04 41 0C 00 00
17:02:52.215 7E9 04 41 0C 00 00
```

Obr. 20 – Komunikace s vozidlem. Funkční adresování. Více odpovědi.

Pro nastavení headeru (CAN ID) žádosti slouží příkaz AT SH XXX. Stejný příkaz jako na Obr. 16, je na Obr. 21, ale s předem nastaveným headerem 0x7DF a odpověď zůstává, při stejných podmínkách, samozřejmě stejná.

```
12:58:35.964 AT SH 7DF
12:58:36.291 OK
12:58:36.291
12:58:36.291 >12:58:40.465 01 00
12:58:40.501 41 00 80 1A 80 01
```

Obr. 21 – Komunikace s emulátorem vozidla. Funkční adresování. Nastaven header.

Využívání pouze emisně relevantních dat a komunikace jen s řídicí jednotkou motoru je nedostatečné (jak bylo rozebráno v kapitole 2.2.3). Pro komunikaci s konkrétními jednotkami vozidla a využívání jiných služeb než těch, které podporuje standard EOBD, je vhodné použít tzv. fyzické adresování. Pro využití tohoto typu adresování je potřeba před odesláním příkazu nastavit header žádosti (Obr. 22).

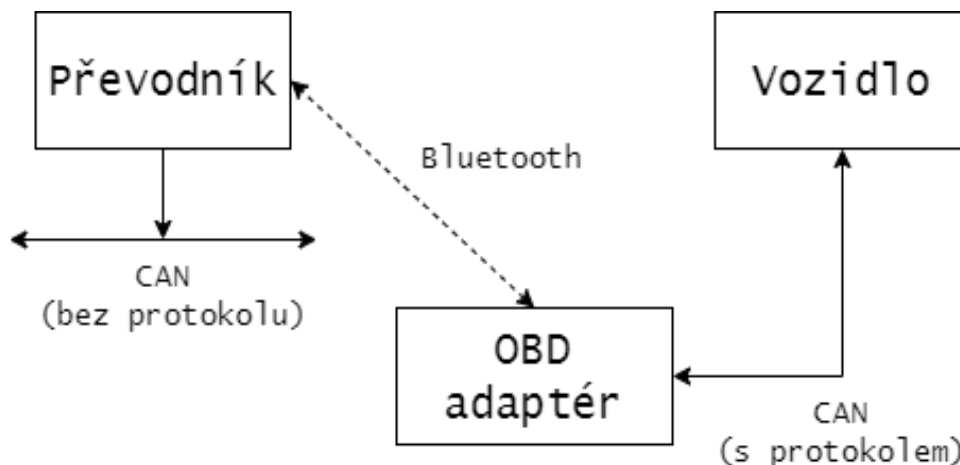
Po nastavení headeru pro fyzické adresování se jednoduše zašle další příkaz ve formě žádosti o data, jehož obsahem je číslo služby (SID) a poté požadovaný DID.

```
13:00:47.316 AT SH 7E0
13:00:47.806 OK
13:00:47.806
13:00:47.806 >13:00:53.820 22 16A9
13:00:54.104 62 16 A9 00 01 E2 40
```

Obr. 22 – Komunikace s emulátorem vozidla. Fyzické adresování. Služba mimo rámec EOBD.

3 Požadavky na převodník

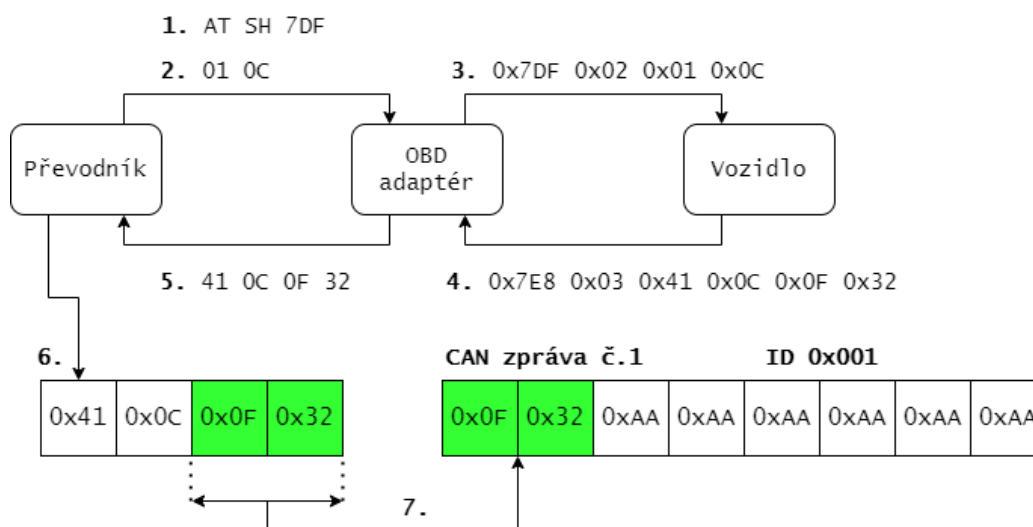
V této části práce jsou popsány požadavky na převodník. Na Obr. 23 je vidět základní blokové schéma systému komunikace.



Obr. 23 - Blokové schéma systému komunikace.

Zadáním práce bylo vytvořit převodník, který bude přeposílat diagnostická data, která jsou vyčítána z OBD zásuvky vozidla po sběrnici CAN bus protokolem UDS. Ten je popsán v normě ISO 14229-1 [1]. Dále má tyto data odesílat na další sběrnici CAN bus, ale bez protokolu (tedy pouze surová data, která jsou uložena do CAN zpráv s definovaným ID).

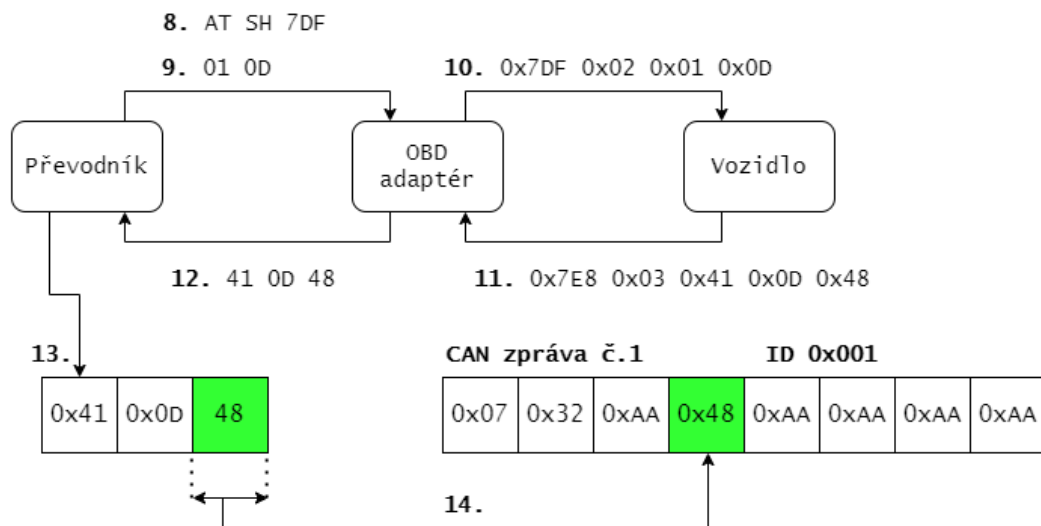
Tento proces je dále popsán pomocí Obr. 24 - Obr. 26.



Obr. 24 – Proces žádosti o otáčky motoru (SID 0x01 PID 0x0C) a následné uložení odpovědi.

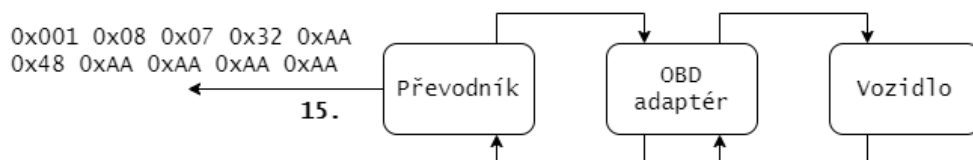
V tomto příkladě má k sobě CAN zpráva s ID 0x001 přiřazeny dva signály. Jedná se o otáčky motoru (PID 0x0C) a rychlost vozidla (PID 0x0D).

V kroku č. 1 a č. 2 (Obr. 24) převodník zašle příkaz na nastavení headeru requestu a poté zašle žádost o data (žádost o otáčky motoru). Následně adaptér zažádá o tato data vozidlo (krok č. 3), které mu vzápětí odpoví (krok č. 4). OBD adaptér tato data zpracuje a převodníku poskytne pouze užitečná data, jak je vidět v kroku č. 5. Převodník přijatá data nijak neupravuje, pouze je „vzvedne“ (krok č. 6) z přijaté zprávy (na základě polohy a délky signálu v bajtech) a přesune je do dané CAN zprávy na definovanou pozici (krok č. 7).



Obr. 25 – Proces žádosti o rychlost vozidla (SID 0x01 PID 0x0D) a následné uložení odpovědi.

Po zpracování signálu s otáčkami motoru, dojde ke zpracování signálu s rychlostí vozidla (Obr. 25). Převodník opět nastaví header requestu a pošle žádost o data do OBD adaptéru (kroky č. 8 a č. 9). OBD adaptér zažádá o tento SID a PID vozidlo a následně zpracuje jeho odpověď (kroky č. 10 a č. 11), kterou poskytne převodníku (krok č. 12). Převodník tato data opět pouze „vzvedne“ z dané pozice z odpovědi adaptéru a přesune je do dané CAN zprávy na danou pozici. Bajty s hodnotami 0xAA jsou zde použity pouze pro přehlednější vysvětlení. Buď na tuto pozici nejsou přiřazena data z žádného ze signálů a nachází se zde pouze jako výplň, nebo na tyto pozice mohou patřit hodnoty jiných signálů.



Obr. 26 – Odeslání vytvořené zprávy.

Takto vytvořenou CAN zprávu (ID 0x001) převodník následně odešle na sběrnici CAN bus bez protokolu (Obr. 26).

4 Hardware zařízení

V této části práce jsou více do hloubky popsány jednotlivé části hardwaru převodníku a jejich funkce.

4.1 Základní návrh převodníku

V první řadě je potřeba zajistit komunikaci mezi samotným převodníkem a vozidlem. Pro vyčítání zpráv CAN z vozidla přes OBD zásuvku byl zvolen OBD adaptér VLinker MC+ [4] (lze ovšem použít jakýkoliv OBD adaptér, který má v sobě kompatibilní ELM327 čip), který s převodníkem komunikuje pomocí Bluetooth. (viz kapitola 4.5). Tento adaptér značně ulehčuje návrh softwaru zařízení, jelikož umožňuje uživateli komunikovat s adaptérem pomocí tzv. AT příkazů (AT commands) a uživateli jsou ze strany vozidla poskytována pouze užitečná data z přijatých CAN zpráv (pokud si uživatel nepřeje jinak). Veškerou komunikaci s vozidlem, pomocí protokolu UDS, si zajišťuje adaptér sám.

Dále je přijatá data potřeba zpracovat dle nějakého předem určeného formátu. Minimálně aby bylo možné určit, o jaká data (jaké konkrétní služby SID a jaké PID) bude převodník žádat a jak často. Na jaké pozici v odpovědi z vozidla se hledaná data nachází a o jak dlouhá data v bajtech se jedná. A v neposlední řadě, do jaké CAN zprávy na sběrnici dash loggeru, a na jakou pozici, se mají data uložit a jak často se mají CAN zprávy na sběrnici odesílat.

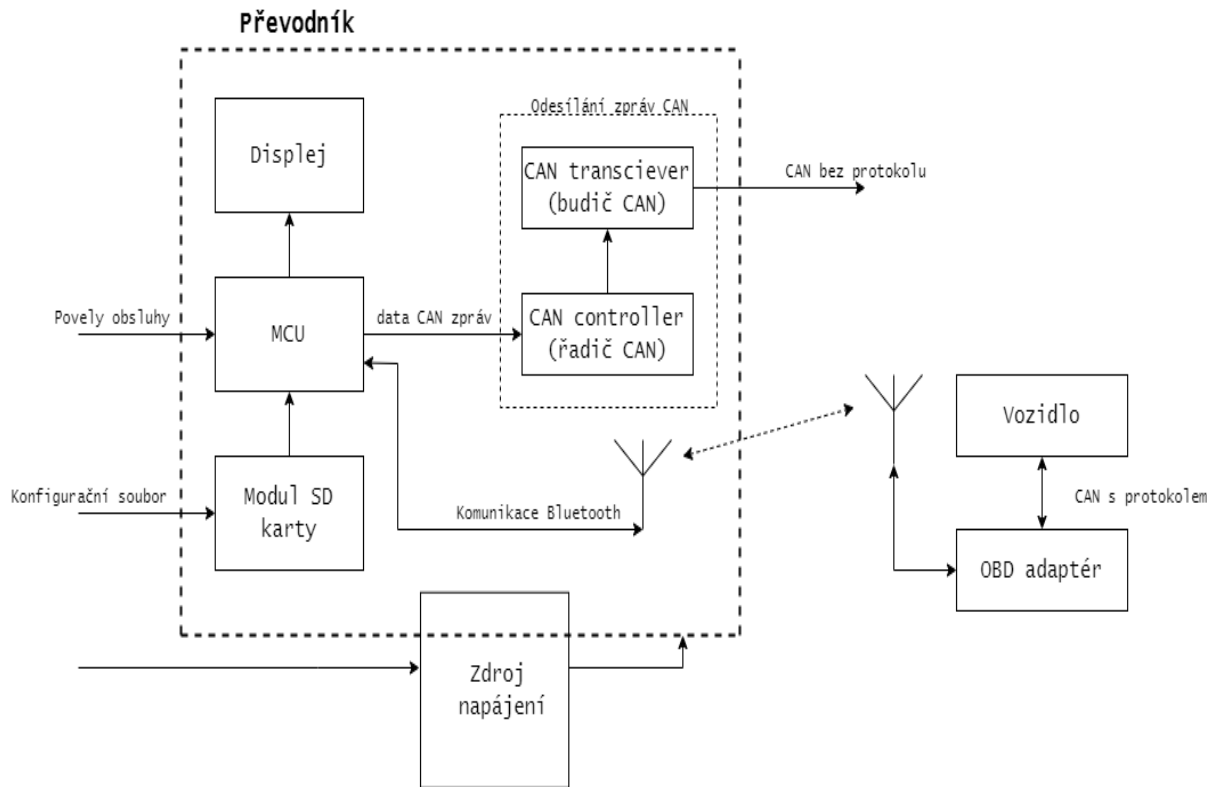
Pro praktické používání převodníku je potřeba, aby si uživatel mohl definovat:

- Jaké informace se budou z vozidla vyčítat a jak často.
- Do jakých CAN zpráv na straně dash loggeru se budou vyčtená data z vozidla ukládat a jak často se budou tyto CAN zprávy vysílat a pod jakými identifikátory.

Tyto informace jsou uloženy do textového konfiguračního souboru (CSV), který je uložen na SD kartě. Převodník si soubor z SD karty načte a podle toho nastaví komunikaci. Soubor je možné editovat ručně v textovém editoru na PC. Vytvoření uživatelsky přívětivé aplikace pro editaci konfiguračního souboru nebylo součástí zadání. Podrobněji je tento konfigurační soubor popsán v kapitole 5.

Dalším požadavkem je malá velikost převodníku, aby šel schovat pod sedlo motocyklu.

Na Obr. 27 se nachází blokové schéma celého systému komunikace, které vyhovuje všem těmto požadavkům.



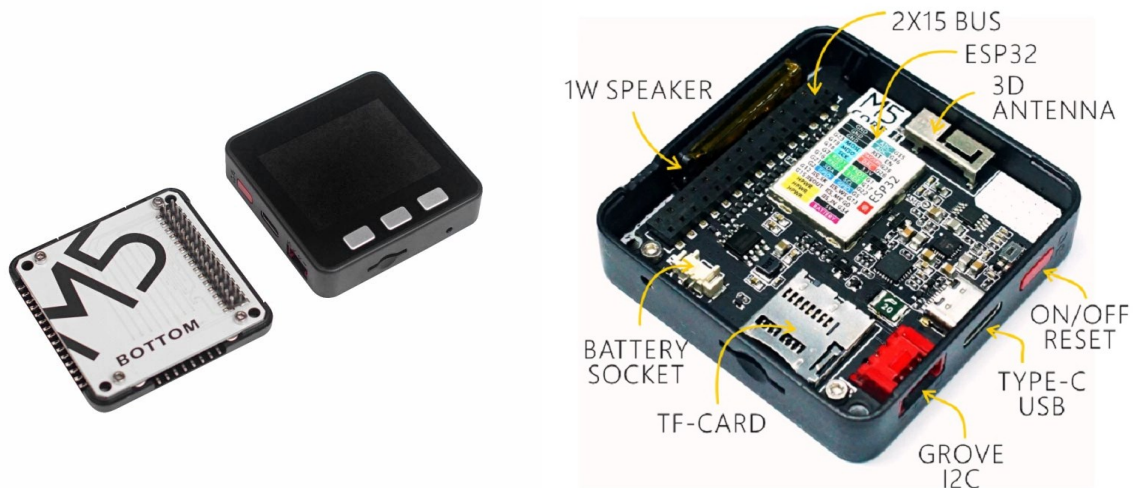
Obr. 27 – Blokové schéma systému komunikace s detailem na převodník.

Pro všechny tyto funkce lze výhodně využít sériově vyráběné zařízení, které už v sobě má integrovaný mikrokontroler (MCU) a veškeré potřebné periferie. A proto pro zpracování dat z OBD adaptéru, práci s konfiguračním souborem z SD karty, výpis informací obsluhy na displej a odesílání zpráv na sběrnici CAN bez protokolu, byla zvolena stavebnice M5Stack CORE BASIC [3], která obsahuje mikrokontroler ESP32. Ten je vhodný pro programování v jazyce C, a dá se tedy jednoduše programovat jako zařízení platformy Arduino (viz kapitola 4.2).

Toto řešení bylo vybráno hlavně kvůli zjednodušení návrhu hardwaru převodníku. Protože celkový návrh, výběr součástek, sestavení a následné programování a testování převodníku by bylo časově náročné. Jelikož se jedná o jednoúčelové zařízení, celková cena součástek a ostatních periférií by byla vyšší než při pořízení už existujícího zařízení. V neposlední řadě jsou dalšími velkými výhodami malé rozměry celé stavebnice a její estetický návrh.

4.2 Stavebnice M5Stack CORE BASIC

M5Stack CORE BASIC je modulární stavebnice, jejímž základem je mikrokontroler ESP32. Zařízení se skládá ze dvou oddělitelných částí.



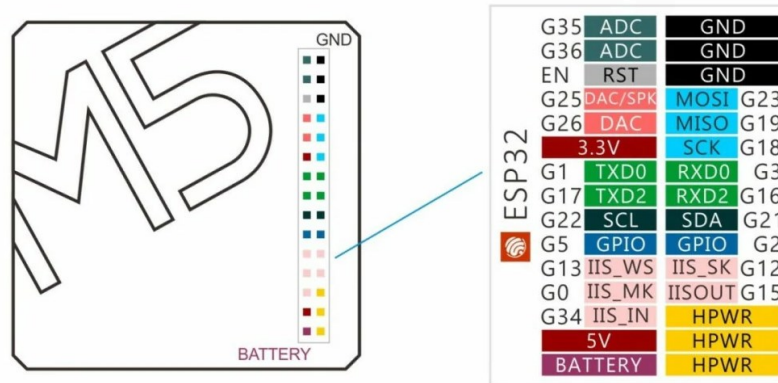
Obr. 28 - M5Stack CORE BASIC. Vlevo části zařízení, vpravo rozvržení komponent. Zdroj [3].

Horní část obsahuje

- samotný mikrokontroler ESP32,
- 2.4G 3D anténa,
- vestavěný reproduktor,
- tři tlačítka,
- barevný TFT LCD,
- vypínací/resetovací tlačítko,
- port na USB typ C,
- port GROVE, který se dá použít jako GPIO piny, nebo piny pro sběrnice I2C a UART,
- slot na SD kartu, jejíž maximální velikost paměti může být 16 GB,
- a další důležité komponenty a integrované obvody, bez kterých by zařízení nemohlo správně fungovat (power management, zesilovač pro reproduktor...).

Ve spodní části se nachází lithiová baterie 3,7 V, o kapacitě 110 mAh, a několik GPIO pinů.

Fyzické a elektrické propojení těchto částí mezi sebou, resp. těchto částí a dalších modulů mezi sebou zajišťuje tzv. M-BUS. Název této sběrnice se shoduje s názvem průmyslového komunikačního protokolu pro dálkový odečet hodnot z měřičů spotřeby, nicméně s tímto protokolem nemá nic společného a název je tedy specifický pro výrobce M5Stack. Názvy, funkce a alternativní funkce jednotlivých pinů M-BUS, jsou vidět na Obr. 29.



Obr. 29 – Názvy, funkce a alternativní funkce jednotlivých pinů M-BUS. Zdroj [5].

Odlíšnosti použitého zařízení od schématu zapojení

Výrobce poskytuje schéma zapojení stavebnice [10], ale pouze jeho první verzi pro CORE BASIC v1.4 z roku 2017, kde je použita externí paměť FLASH s označením GD25Q32C a s velikostí paměti pouze 4MB. V převodníku použita stavebnice M5Stack CORE BASIC v2.6 už disponuje externí pamětí dat typu FLASH o velikosti 16MB (o neznámém označení) a dalšími změnami, které jsou uvedené v tabulce na Obr. 30, zdroj [13].

Release Date	Product Changes	Notes
2017.7	First Release(Core1.4)	/
2019.7	TN screen changed to IPS screen(Core2.2)	Please upgrade your M5Stack library to the latest version (v0.2.8 or above) to solve the screen reflection problem
2020.3	Battery capacity changed from 150mAh to 110mAh(Core2.4)	/
2020.6	Flash size changed from 4MB to 16MB(Core2.5)	/
2021.10	Upgrade v2.6, change CP2104 to CH9102, optimize structure details(Core2.6)	/

Obr. 30 – Výrobní změny produktu týkající se M5Stack CORE BASIC oproti dostupnému schématu zapojení, který poskytuje výrobce (zdroje [10] a [13]).

Moduly poskytované výrobcem

Výrobce zařízení nabízí velkou sadu různých modulů, které lze připojit pomocí M-BUS mezi dvě základní části. Každý z modulů obsahuje různé periferie a tím rozšíří funkce a možnosti základního zařízení. Jedním z těchto modulů je COMMU modul [5], který je v převodníku použit pro odesílání zpráv CAN na sběrnici CAN bus bez protokolu. Více je tento modul popsán v kapitole 4.3.

4.2.1 Mikrokontroler ESP32

ESP32 je řada SoC mikrokontrolerů. Ve stavebnici M5Stack je použit model ESP32-D0WDQ6-V3. Tato řada disponuje nízkou spotřebou. Pro vybraný model je udáván odběr 30 mA ~ 68 mA při použití obou jader na frekvenci 2,4 GHz a 5 μ A v režimu hibernace. Mikrokontroler má integrované rozhraní Wi-Fi a Bluetooth v duálním režimu. Model obsahuje dvoujádrový 32-bitový mikroprocesor Xtensa® LX6 s taktovací frekvencí 2,4 GHz v provádění až 600 DMIPS [8].

Co se týče paměťových prostor, nachází se zde

- 448 KB paměti ROM pro paměť programu,
- 520 KB paměti SRAM pro paměť dat
- a 16KB SRAM pro RTC (8KB tzv. RTC SLOW paměti a 8KB tzv. RTC FAST paměti).

Dále stojí za zmínku

- 34 programovatelných GPIO linek,
- 12-bitový A/D převodník SAR (s postupnou aproximací) s až 18 kanály,
- dva 8-bitové D/A převodníky

Tento mikrokontroler podporuje mnoho programovacích jazyků, rámců, platforem a prostředí. Oficiální framework IoT společnosti Espressif je ESP-IDF, ovšem tento mikrokontroler lze programovat ve frameworku Arduino-ESP32, což značně ulehčí vývoj software pro konstrukci převodníku. (Zdroje [6] a [7]).

4.3 COMMU modul

Jako součást převodníku, pro odesílání zpráv CAN na sběrnici bez protokolu, byl vybrán modul COMMU [5]. Tento modul nabízí výrobce stavebnice M5Stack, která je základem celého převodníku.

Výhoda při zvolení tohoto modulu je ta, že je už od výrobce plně kompatibilní se stavebnicí M5Stack CORE BASIC. Tento modul se dá připojit pomocí sběrnice M-BUS mezi horní a spodní část stavebnice, a tak se stane přímo její součástí.



Obr. 31 – COMMU modul. Vlevo zařízení, vpravo rozvržení komponent. Zdroj [5].

V modulu se nachází rozhraní několika sériových komunikačních linek. Nachází se zde rozhraní pro

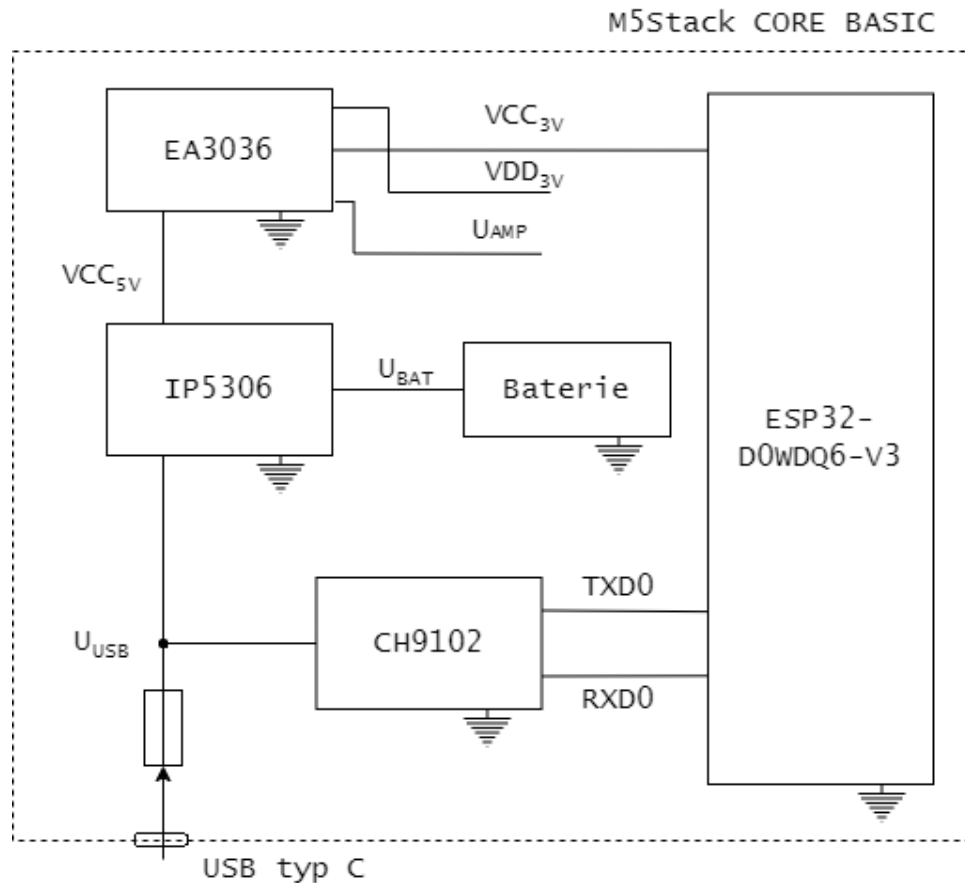
- sběrnici I2C,
- sériovou linku RS485,
- rozhraní UART → TTL
- a sběrnici CAN.

Pro sestavení převodníku je nejdůležitější ta část modulu, která umožňuje odesílání zpráv na sběrnici CAN bus. Pro realizaci linkové a fyzické vrstvy na sběrnici CAN bus, obsahuje tento modul řadič CAN a budič CAN.

V modulu COMMU se nachází řadič CAN s označením MCP2515 a budič CAN s označením TJA1051 (CAN high-speed).

4.4 Napájení převodníku

Kritickou součástí celého převodníku je řešení jeho napájení. V této části práce je tedy popsána problematika napájení převodníku. Na Obr. 32 je vyobrazeno blokové schéma napájení.



Obr. 32 – Blokové schéma napájení M5Stack CORE BASIC.

Stavebnice M5Stack CORE BASIC disponuje jedním portem USB typu C (Obr. 32 – v místě u U_{USB} , umístěno před pojistkou). Přes tento port se nahrává program do mikrokontroleru ESP32 (k nahrání programu slouží převodník USB/UART CH9102) a zároveň slouží pro napájení zařízení, jeho periferií, a vestavěné baterie. Obvod IP5306 slouží jako nabíječka vestavěné lithiové baterie napětím $U_{BAT} = 4,2$ V. EA3036 je tříkanálový DC-DC měnič. Jeden z kanálů je použitý pro napájení mikrokontroleru (VCC_{3V}), druhý pro napájení zesilovače pro reproduktor (U_{AMP}) a poslední kanál je vyveden do sběrnice M-BUS a slouží pro napájení ostatních periferií (VDD_{3V}). Výstupní napětí všech tří kanálů tohoto obvodu je nastaveno na 3,3 V.

4.4.1 Napájení pomocí kabelu USB

Napájet převodník by šlo přímo z počítače, ze kterého je převodník programován. To je ale přijatelné pouze pro vývoj, ladění programu a jeho následné testování. Ale pro následný provoz převodníku na vozidle je tato metoda velice nevhodná. Zařízení navíc trpí takovou

nepříjemností, která vždy vede k restartu/vypnutí zařízení při vytažení kabelu z nabíjecího portu.

4.4.2 Použití baterie

Použitá stavebnice obsahuje vlastní lithiovou baterii o kapacitě 110 mAh a jmenovitém napětí 3,7 V. Pro praktické použití převodníku, a jeho provoz v řádech hodin, je tato kapacita nedostatečná. Proto toto řešení nepřípadá v úvahu.

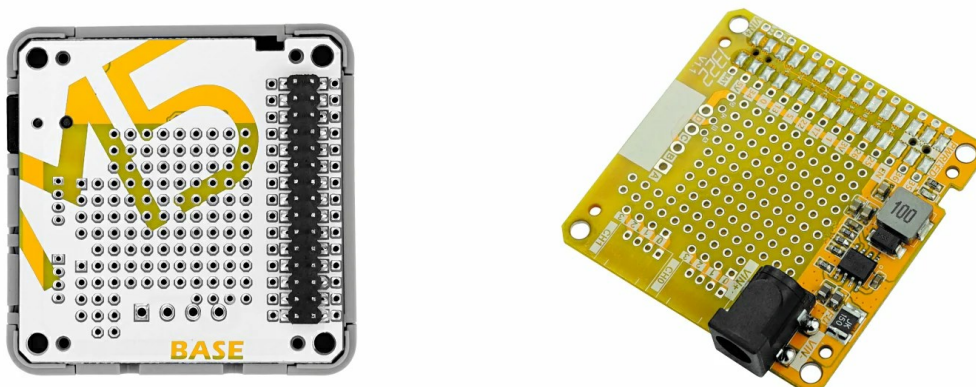
Další možností by bylo jako zdroj napájení použít nějaké externí úložiště elektrické energie. K tomuto účelu by mohl sloužit libovolný akumulátor s Ah kapacitou dle požadavku na dobu výdrže převodníku. Z tohoto zdroje by se stavebnice, pomocí kabelu USB, mohla napájet. Toto řešení je ovšem nevhodné z toho důvodu, že by se rozměry výsledného zařízení mnohonásobně zvětšily.

4.4.3 Napájecí modul

Použitelným řešením pro vyřešení otázky napájení je použití napájecího modulu, jehož zdrojem energie by byla zdrojová soustava vozidla. Zde jsou uvedeny možnosti použití modulu jak od samotného výrobce stavebnice M5Stack, tak od výrobců jiných.

Modul BASE 15

Výrobce stavebnice dodává svůj napájecí modul, který je se stavebnicí plně kompatibilní, a tak jako modul COMMU (viz 4.3), se takový modul zapojí jako další patro stavebnice, ovšem tentokrát místo spodní části (místo baterie). Zde označen jako měnič A.



Obr. 33 – Modul BASE 15. Vlevo zařízení (shora), vpravo rozvržení komponent (zdola). Zdroj [14].

Modul BASE 15 [14] má v sobě DC-DC měnič, postavený na čipu MP1584. Výrobce uvedené parametry (bez dalšího kontextu) tohoto modulu jsou uvedeny v Tab. 2.

Tab. 2 – Parametry modulu BASE 15 udávané výrobcem MStack. Měnič A. Zdroj [14].

Rozsah vstupního napětí	9 - 24 V
Kontinuální výstupní proud	2A
Výstupní napětí	5V

Tento modul je dle parametrů vhodný pro použití do všech vozidel s palubním napětím 12 V. Problém by ale mohl nastat v případě poklesu napětí pod 9 V, způsobeném startováním spalovacího motoru vozidla.

Parametry modulu, uvedené v Tab. 2, se ale od parametrů výrobce samotného čipu liší. Výrobce udává rozsah vstupního napětí počínaje 4,5 V a konče 28 V (zdroj [15]).

Vlastní modul

Dalším možným řešením je použití DC-DC měniče, jehož parametry budou lépe odpovídat požadavku na rozsah vstupního napětí. Tomuto odpovídá sériově vyráběný Step-down měnič, který je postaven na čipu MP2315 [16]. Zde označen jako měnič B. Základní parametry měniče B jsou uvedeny v Tab. 3.

Tab. 3 – Parametry Stepdown DC-DC měniče s čipem MP2315. Měnič B. Zdroj [16].

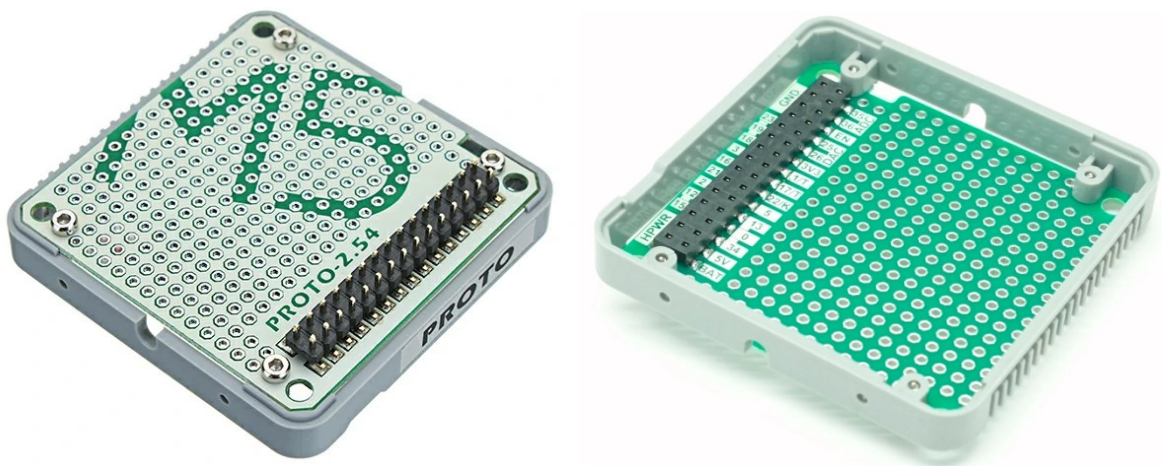
Rozsah vstupního napětí	4,5 - 24 V
Kontinuální výstupní proud	3A (při výstupním napětí 5V)
Výstupní napětí	0,8 – 15 V

U měniče B je spodní hodnotou vstupního napětí 4,5 V, což by mnohem lépe odpovídalo nárokům na použití převodníku např. na motocyklech, kde je úroveň palubního napětí rovna právě 6 V. Nemuselo by dojít k přerušení napájení ani při poklesu napětí při startování spalovacího motoru. Nevýhodou použití měniče B je fakt, že měnič A je oproti měniči B už od výroby umístěn v plastovém obalu a tvoří tak modul, který se lehce nasadí na stavebnici a tvoří s ní jeden celek (Obr. 33). Měnič B je ale pouze umístěn na DPS (Obr. 34) a v žádném obalu umístěn není.



Obr. 34 – DC-DC Step-down měnič s čipem MP2315. Měnič B.

V tomto případě by bylo vhodné měnič B umístit do modulu PROTO [17], který opět nabízí výrobce M5Stack. Tento modul umožňuje vytvoření vlastního modulu, jelikož obsahuje jen prototypovou desku plošných spojů, uloženou v plastovém obalu, která má na sobě napájecí konektor M-BUS (Obr. 35). Takto vyrobený modul by měl malé rozměry a byl vzhledný i po estetické stránce.



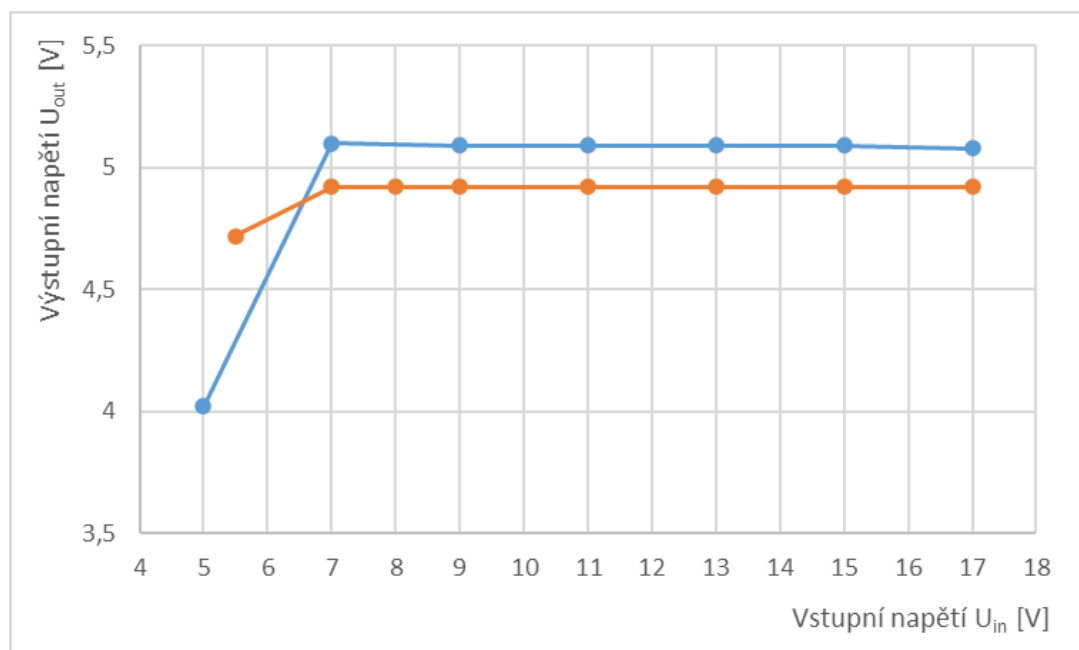
Obr. 35 - Modul PROTO. Vlevo pohled shora, vpravo pohled zdola. Zdroj [17].

Pro návrh toho, kam by se měly jednotlivé vstupy a výstupy měniče B do modulu PROTO napájet, lze využít schéma zapojení už zmíněného modulu BASE 15 (zdroj [14]) a tabulku rozvržení signálů v konektoru M-BUS (Obr. 29).

Vzhledem k tomu, že parametry měniče A nejsou na první pohled vhodné pro plánované použití převodníku, byly oba zmíněné měniče otestovány.

Bylo provedeno měření, při kterém byl měnič napájen z laboratorního zdroje a byl připojen k zátěži s proměnnou hodnotou odporu. Při tomto měření byla měněna hodnota vstupního napětí měniče (pomocí laboratorního zdroje) a byla odečítána hodnota napětí výstupního. Měření proběhlo při konstantní hodnotě výstupního proudu (0,5 A).

Výsledek z měření je znázorněn na Obr. 36. Oranžově měnič A, modře měnič B.



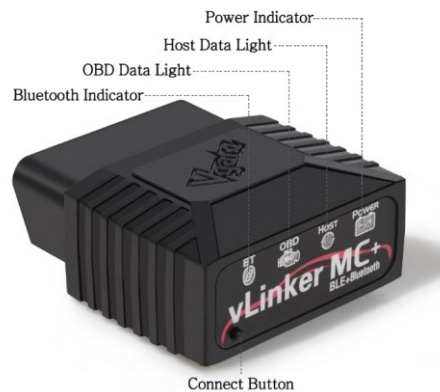
Obr. 36 – Výsledky z měření měničů pro napájení převodníku. Závislost výstupního napětí na vstupním napětí při výstupním proudu 0,5 A. Oranžově měnič A, modře měnič B.

Z grafu lze vyčíst, že hodnota výstupního napětí obou měničů začne razantně klesat při hodnotě vstupního napětí, která je menší než 7 V. Nad touto hranicí tedy oba měniče měly na svém výstupu napětí o hodnotě 5 V (přibližně, nicméně je hodnota konstantní), při zátěži 0,5 A.

Z toho vyplývá, že i přes na první pohled nevhodné parametry měniče A, jsou oba měniče vhodné pro tuto aplikaci. Při použití na vozidle, které má hodnotu napětí palubní sítě 12 V, by mohly oba měniče spolehlivě napájet převodník i v případě poklesu napětí při startování spalovacího motoru vozidla, a tedy lze pro napájení vybrat aktuálně levnější variantu. Pro úsporu času a pro jednoduchost a estetičnost řešení lze tedy použít přímo modul, který je nabízen výrobcem M5Stack.

4.5 OBD adaptér

OBD adaptér zde slouží jako prostředník komunikace mezi převodníkem a vozidlem. Ke stavebnici M5Stack CORE BASIC je připojený pomocí Bluetooth.



Obr. 37 – OBD adaptér vLinker MC+ výrobce Vgate. Zdroj [4].

Pro použití byl vybrán adaptér vLinker MC+ od výrobce Vgate [4]. Tento adaptér v sobě obsahuje integrovaný obvod s označením ELM327. Tento obvod funguje jako převodník mezi sériovou sběrnici RS232 a sběrnici, které se nachází v diagnostické zásuvce vozidla.

Tento obvod podporuje OBD protokoly:

- SAE J1850 PWM (41.6 kBaud)
- SAE J1850 VPW (10.4 kBaud)
- ISO 9141-2 (5 Baud init)
- ISO 14230-4 KWP (5 Baud init)
- ISO 14230-4 KWP (fast init)
- ISO 15765-4 CAN OBD (11 bit ID, 500 kBaud)
- ISO 15765-4 CAN OBD (29 bit ID, 500 kBaud)
- ISO 15765-4 CAN OBD (11 bit ID, 250 kBaud)
- ISO 15765-4 CAN OBD (29 bit ID, 250 kBaud)
- SAE J1939 CAN (29 bit ID, 250 kBaud)
- A dva další protokoly CAN, ve kterých si uživatel může nastavit přenosovou rychlost a délku ID sám

Tento adaptér značně ulehčuje komunikaci s vozidlem, jelikož obvod ELM327 přijímá z převodníku pouze tzv. AT příkazy, komunikaci s vozidlem pomocí protokolu si zajišťuje sám. Na Obr. 38 se nachází výňatek ze seznamu AT příkazů tohoto obvodu.

AT Command Summary

General Commands

<CR>	repeat the last command
BRD hh	try Baud Rate Divisor hh
BRT hh	set Baud Rate Timeout
D	set all to Defaults
E0, E1	Echo off, or on*
FE	Forget Events
I	print the version ID
L0, L1	Linefeeds off, or on
LP	go to Low Power mode
M0, M1	Memory off, or on
RD	Read the stored Data
SD hh	Save Data byte hh
WS	Warm Start (quick software reset)
Z	reset all
@1	display the device description
@2	display the device identifier
@3 cccccccccc	store the @2 identifier

Programmable Parameter Commands

PP xx OFF	disable Prog Parameter xx
PP FF OFF	all Prog Parameters disabled
PP xx ON	enable Prog Parameter xx
PP FF ON	all Prog Parameters enabled
PP xx SV yy	for PP xx, Set the Value to yy
PPS	print a PP Summary

Voltage Reading Commands

CV dddd	Calibrate the Voltage to dd.dd volts
CV 0000	restore CV value to factory setting
RV	Read the input Voltage

Other

IGN	read the IgnMon input level
------------	-----------------------------

Note: Settings shown with an asterisk (*) are the default values

Obr. 38 – Detail na některé AT příkazy obvodu ELM327. Zdroj [26].

4.5.1 Komunikace s OBD adaptérem

Na Obr. 39 se nachází příklad komunikace s OBD adaptérem pomocí mobilní aplikace Bluetooth Terminal. Ta umožňuje spojení s Bluetooth zařízením a také zasílání a přijímání sériových dat.

Příkaz AT Z slouží ke kompletnímu resetu čipu, změny veškerého nastavení na výchozí a vypsání verze obvodu ELM327. Příkaz AT @1 vypíše řetězec s popisem zařízení, v tomto případě „OBDII to RS232 Interpreter“.

Ve výchozím nastavení obvod při přijetí příkazu odešle nazpátek jako první zopakování přijatého příkazu a poté odpovídající odpověď. Příkaz AT E0 slouží k tomu, aby obvod nazpět neodesílal poslední přijatý příkaz (jak je vidět na Obr. 39, obvod už nazpět přijaté příkazy neodesílá).

AT SP slouží k určení protokolu OBD, pomocí kterého bude ELM327 komunikovat s vozidlem. Upřesnění AT SP 6 znamená, že byl nastaven protokol ISO 15765-4 CAN OBD (11 bit ID, 500 kbaud).

V případě zadání správného příkazu navrátí obvod přijatá diagnostická data, nebo hlášku „OK“ (že příslušné nastavení proběhlo v pořádku). Nicméně při zadání neznámého příkazu navrátí obvod „?“, jako v případě příkazu AT @2. Tento příkaz by měl navrátit řetězec s ID zařízení, které se nastaví příkazem AT @3. Nicméně toto ID nastavené nebylo, a proto tomu ELM 327 „nerozumí“ a navrátí „?“.

Nakonec příkaz AT LP přikáže obvodu k přechodu do režimu nízké spotřeby, což má za následek odpojení OBD adaptéru od mobilního zařízení a vypsaní hlášky „Connection lost“.

```
11:55:25.829 AT Z
11:55:26.985
11:55:26.985
11:55:26.985 ELM327 v2.2
11:55:26.985
11:55:26.985 >11:55:34.346 AT @1
11:55:34.647 AT @1
11:55:34.647 OBDII to RS232 Interpreter
11:55:34.647
11:55:34.647 >11:55:45.815 AT E0
11:55:46.165 AT E0
11:55:46.165 OK
11:55:46.165
11:55:46.165 >11:55:57.303 AT SP 6
11:55:57.686 OK
11:55:57.686
11:55:57.686 >11:56:32.760 AT DP
11:56:33.207 ISO 15765-4 (CAN 11/500)
11:56:33.207
11:56:33.207 >11:57:03.850 AT @2
11:57:03.924 ?
11:57:03.924
11:57:03.924 >11:57:52.828 AT LP
11:57:52.882 OK
11:57:52.954
11:57:53.120 Connection lost
```

Obr. 39 – Příklad použití AT příkazů. Komunikace s OBD adaptérem.

Dále je zde seznam příkazů, které jsou převodníkem využívány. Pro žádosti o data z vozidla jsou využívány příkazy

- AT SH xyz (nastavení headeru requestu pro funkční a fyzické adresování) a
- SID PID (zadání služby a parametru jako žádost o data z vozidla)

Při inicializaci převodníku je potřeba OBD adaptér před používáním nastavit. K tomu slouží příkazy

- AT Z (pro celkový reset ELM327),
- AT E0 (aby ELM327 neopakoval zadané příkazy),
- AT H0 (pro vypnutí zobrazení headerů v odpovědi převodníku) a
- AT SP 0

Příkaz AT SP 0 slouží k nastavení komunikačního protokolu. Nula znamená, že ELM327 sám nastaví použitelný komunikační protokol. Předpokládá se, že vozidlo komunikuje na OBD zásuvce přes sběrnici CAN bus.

5 Konfigurační soubor

V této kapitole je představen formát konfiguračního souboru a vlastnosti programu při jeho načítání.

Jak bylo popsáno v podkapitole 4.1, která pojednává o návrhu samotného převodníku, je potřeba nějak převodníkem nastavit jaká diagnostická data se mají převodníkem vyčítat, tedy konkrétní služby SID a parametry PID (DID).

Dále je potřeba převodníkem přijatá data z OBD adaptéru příslušným způsobem zpracovat. Převodník potřebuje informaci o tom, na jaké pozici, v přijaté zprávě, se hledaná data nacházejí a kolik bajtů dat tato hledaná data obsahují. Nakonec je nutné určit do jaké zprávy CAN (o jakém ID) je potřeba tato data vložit a na jakou pozici.

Hodnota některých vyčítaných dat se v čase mění velice rychle, např. údaj o otáčkách motoru nebo rychlosti vozidla. Naopak hodnoty některých vyčítaných dat není potřeba vyčítat tak často, jako třeba kód readiness nebo seznam závad řídicích jednotek. Bylo by vhodné mít možnost nastavit, jak často se dané signály budou vyčítat z vozidla a kdy se tato data budou odesílat na sběrnici CAN bez protokolu, resp. jestli se mají odesílat neustále periodicky, nebo se mají odeslat až po aktualizaci hodnot všech signálů ve zprávě CAN.

Tyto informace v sobě uchovává konfigurační soubor, který je uložený na SD kartě a načítaný převodníkem. Jedná se o textový soubor CSV s koncovkou TXT. Aby byl konfigurační soubor načten převodníkem, musí být pojmenován „KonfiguracniSoubor.txt“, musí být uložen v kořenovém adresáři SD karty a nesmí se nacházet v žádné složce.

5.1 Formát souboru

Na Obr. 40 se nachází formát konfiguračního souboru. Zeleně jsou označena pole, jejichž obsah převodník nenačítá a slouží pouze jako pomoc obsluze. Modře je vyznačeno pole, jehož obsah převodník načítá, ale jeho obsah není důležitý pro vyčítání a odesílání zpráv CAN.

Na prvním řádku se nachází pole „Verze“. Za tímto polem musí být vyplněna verze konfiguračního souboru. Tato informace se nadále vypisuje na displej obsluze převodníku. Další sloupce jsou prázdné. Na druhém řádku se nachází pole „Poznámka“. Do následujícího sloupce tohoto řádku lze vyplnit libovolnou poznámku týkající se konfiguračního souboru. Např. číslo souboru, jakého vozidla se jedná, datum atd. Další sloupce jsou opět prázdné.

Třetí řádek obsahuje hlavičku souboru, která napovídá obsluze, jaké informace, do jakých polí je třeba vyplnit. Tedy napovídá samotnou strukturu signálu načítaného převodníkem. Pod tímto řádkem (od 4. řádku) se na každém řádku nachází jednotlivé signály vyčítané převodníkem.

prázdná pole												
prázdná pole												
	Verze	1.0										
2.	Poznámka	konfig_X										
3.	Komentář	Jak často PID vyčítat	PID	SID	Header requestu	Číslo bajtu v odpovědi	Délka signálu [bajtů]	CAN ID zprávy	Extended	Perioda odesílání [ms]	Položka ve zprávě [od čísla bajtu]	Info pro obsluhu
4.	Komentář 1	10	0x01	0x01	0x7DF	0	4	0x001	0	0	0	Info 1
5.	Komentář 2	1	0x0C	0x01	0x7DF	0	2	0x002	0	0	0	Info 2
...
X.	Komentář X	1	0x0D	0x01	0x7DF	0	1	0x003	1	100	2	Info X

Obr. 40 – Formát konfiguračního souboru.

Dále jsou vysvětleny jednotlivé sloupce hlavičky.

Komentář

Sloupec „Komentář“ slouží obsluze k vyplnění komentáře, který se týká daného signálu. Obsluha může uvést, že se jedná např. o vyčítaná data s významem otáček motoru nebo teploty chladící kapaliny.

Jak často PID vyčítat

Do tohoto pole se zadává informace o tom, jak často má být tento signál vyčten z vozidla. Znamená to kolikátý každý „vyčítací cyklus“ má být tento signál vyčten. Pro porovnání bude např. signál na 4. řádce (Obr. 31) vyčten jednou za 10 průchodů kolečkem vyčítání dat z OBD zásuvky.

PID

Zadává se sem číslo v šestnáctkové soustavě. Význam je číslo PIDu žádosti o data, např. 0x0C (parametr pro otáčky motoru). Záleží na počtu zadaných cifer. Tedy 8-bit PID je třeba vyplnit se dvěma ciframi za prefixem „0x“ (0x0C) a 16-bit PID se čtyřmi ciframi (0x000C).

SID

Stejně jako v případě sloupce PID, význam je ovšem číslo služby SID. Např. 0x01 (služba měřené hodnoty), 0x22 (ReadDataByIdentifier).

Header requestu

Jedná se opět o číslo v hexadecimální soustavě. Zadává se sem ID zprávy CAN pro žádosti o data (funkčního i fyzického adresování, viz kapitola 2.2.4). Např. 0x7DF nebo 0x7E0. V žádostech se používají 11-bit identifikátory.

Číslo bajtu v odpovědi

Zadává se sem číslo v decimální soustavě. Význam je číslo bajtu, od kterého se má informace z odpovědi OBD adaptéru vyjmout. Převodník pracuje s odpověďmi ve formátu zobrazeném na Obr. 17. Do tohoto čísla se ale nezapočítává bajt se zopakováním SID + 0x40 (positive response, a bajt se zopakováním PIDu (dva bajty v případě dvoubajtových PIDů).

Číslo služby + 0x40 (positive response)	Zopakování zadaného PIDu	Poskytnutá data			
		0.	1.	2.	3.
0x41	0x00	0x80	0x01A	0x80	0x01

Obr. 41 – Odpověď z OBD adaptéru. Číslo bajtu v odpovědi.

Např. když bude číslo v tomto sloupci rovno 0, tak se budou vyzvedávat užitečná data z odpovědi adaptéru od indexu 0, tedy v případě Obr. 41 by to bylo od hodnoty 0x80 (zakroužkováno červeně).

Délka signálu [bajtů]

Jako v předchozím sloupci se sem zadává číslo v desítkové soustavě. Význam je jak dlouhý (kolika bajtový) signál se očekává.

Číslo služby + 0x40 (positive response)	Zopakování zadaného PIDu	Poskytnutá data			
		0.	1.	2.	3.
0x41	0x00	0x80	0x01A	0x80	0x01

Obr. 42 – Odpověď z OBD adaptéru. Délka signálu [bajtů].

Za předpokladu že je parametr „Číslo bajtu v odpovědi“ roven 0, tak zadáním čísla 4 do tohoto sloupce dojde k vyzvednutí dat z odpovědi od indexu 0 a v délce 4 bajtů. Na odpovědi z Obr. 42 by to byly bajty 0x80, 0x1A, 0x80 a 0x01 (vyznačeno zeleně).

CAN ID zprávy

Do tohoto sloupce přijde zadat číslo v hexadecimální soustavě. Číslo má význam ID zprávy CAN, do které daný signál patří, a se kterým se následně bude odesílat na CAN bus bez protokolu.

Jedné zprávě CAN lze tedy přiřadit data z několika signálů. Omezením je pouze to, zda se data do této zprávy vejdou (správné přiřazení si musí obsluha ohlídat při vytváření konfiguračního souboru). Do jedné zprávy CAN se vejde 8 bajtů dat, lze teoreticky přiřadit až 8 signálů k jedné zprávě CAN.

Extended

Zde se zadávají pouze dvě čísla, a to 0 a 1. Tento parametr určuje, zda bude odesílaná zpráva mít CAN ID ve formátu standardního 11 bitového rámce, nebo rozšířeného 29 bitového rámce. Zadáním čísla 0 se bit IDE odesílané zprávy CAN nastaví na 0, což odpovídá standardnímu rámci (11-bit CAN ID). Zadáním čísla 1 se bit IDE nastaví na hodnotu 1 a zpráva se bude odesílat s CAN ID ve formátu rozšířeného (tedy extended) rámce (29-bit CAN ID).

Perioda odesílání [ms]

Tento sloupec obsahuje číslo v desítkové soustavě a může obsahovat dva typy hodnot.

Hodnota větší než 0 znamená, že zpráva CAN, ke které je daný signál přiřazen, bude odesílána periodicky bez ohledu na to, zda jsou data v této zprávě všechna aktualizovaná, či nikoliv. Data se tedy aktualizují vždy, když na dané signály přijde řada ve vyčítání. Zprávy se budou odesílat s periodou rovné číslu, které je v tomto sloupci zapsáno.

Zadáním hodnoty 0 do tohoto sloupce bude zpráva CAN, ke které je daný signál přiřazen, odesílána až tehdy, kdy jsou všechny signály v této zprávě aktualizovány. Tedy nejdříve se musí provést pokus o vyčtení každého z přiřazených signálů a až poté se tato zpráva může odeslat na sběrnici CAN bez protokolu.

Poloha ve zprávě [od čísla bajtu]

Podobně jako parametr „Číslo bajtu v odpovědi“ toto decimální číslo znamená polohu, na kterou se data do zprávy CAN mají uložit.

Data zprávy CAN

index bajtu zprávy CAN	0.	1.	2.	3.	4.	5.	6.	7.
Obsah bajtu zprávy CAN	0XAA	0XAA	0x80	0x1A	0x80	0x01	0XAA	0XAA

Obr. 43 – Data zprávy CAN. Poloha ve zprávě [od čísla bajtu].

Pokud by toto číslo bylo rovno 2, tak by se vyzvednutá data z odpovědi na Obr. 42 uložila do zprávy CAN tak, jak je zobrazeno na Obr. 43 (zeleně zobrazena uložená data, nevyužitá bajty CAN zpráv nastavuje převodník na hodnotu 0xAA).

Info pro obsluhu

Posledním sloupcem na Obr. 40 je „Info pro obsluhu“. Toto pole ve skutečnosti není jedno, ale jedná se o dalších pět sloupců (Obr. 44). Obsah těchto sloupců ovšem není důležitý pro správnou funkci převodníku, ale slouží jako informace pro obsluhu převodníku při tvorbě konfiguračního souboru, a proto bylo pro ilustraci zkráceno pouze na jedno pole.

Info pro obsluhu

Datový typ	Endianita	Factor	Offset	Jednotka
uint32	big endian	100/255	-40	°C

Obr. 44 – Rozdělený parametr Info pro obsluhu.

Tato pole obsahují datový typ dat signálu a endianitu (big či little endian, určuje pořadí bajtů vícebajtových dat). Factor a offset neboli číslo, které slouží pro převod na fyzikální rozměr (např. na °C, 1/min, % apod.). Některé veličiny se totiž po sběrnici CAN mohou přenášet v původní podobě, některé ne. Nakonec jednotka dané veličiny. Převodník vyčtená data nijak nemění (nepřevádí je na fyzikální rozměr). Pouze je převezme a uloží je do zprávy CAN, která je následně odeslána na sběrnici CAN bez protokolu.

Na Obr. 45 se nachází takto vytvořený konfigurační soubor ve formátu .txt. Pro správnou funkci programu je potřeba, aby v takto vytvořeném souboru byl oddělovačem dat znak středník ';', neboť program tak od sebe rozlišuje data v jednotlivých sloupcích.

```

Verze;1.0;,,,,,,,,,,,,;
Poznámka;konfig_10 Octavia 3 2.0 TDI 110 kW DCYA školní,,,,,,,,,,,,;
Komentář;Jak často_PID_vyčítat;PID;SID;Header_requestu;Číslo_bajtu_v_odpovědi;Délka_signálu_[bajtů];CAN_ID_zprávy;Extended;Per
ioda_odesílání_[ms];Poloha_ve_zprávě_[od_čísla_bajtu];datový typ;endianita;factor;offset;jednotka
Air Mass Flow Rate (kg/h);1;0x1024;0x22;0x7E0;0;2;0x003;0;0;1;int16;big endian;0,1;0;kg/h
Air Mass per stroke (mg/stroke);1;0x1026;0x22;0x7E0;0;2;0x003;0;0;3;int16;big endian;0,1;0;mg/z
Air temp after cooler (teplota nasávaného vzduchu za chladičem);1;0x1798;0x22;0x7E0;0;2;0x002;0;0;4;uint16;big
endian;0,1;-273,1;°C
Diesel aftertreatment;5;0x8B;0x01;0x7DF;0;7;0x005;0;0;0;uint56;big endian;0;0;viz ISO 15031-5
Engine coolant temperature;5;0x05;0x01;0x7DF;0;1;0x005;0;0;7;uint8;1;-40;°C
Engine speed (otáčky motoru);1;0x0C;0x01;0x7DF;0;2;0x002;0;0;0;uint16;big endian;0,25;0;1/min
Exhaust gas temperature sensor EGT11 (B1S1, před turbem);1;0x10FB;0x22;0x7E0;0;2;0x004;0;0;0;uint16;big endian;0,1;-273,1;°C
Exhaust gas temperature sensor EGT12 (B1S2, za turbem);1;0x14F7;0x22;0x7E0;0;2;0x004;0;0;2;uint16;big endian;0,1;-273,1;°C
Exhaust gas temperature sensor EGT13 (B1S3, před DPF);1;0x11B2;0x22;0x7E0;0;2;0x004;0;0;4;uint16;big endian;0,1;-273,1;°C
Intake Manifold Absolute Pressure (aboslutní tlak vzduchu v sání);1;0xF40B;0x22;0x7E0;0;1;0x003;0;0;0;uint8;1;0;kPa
Monitor status since DTCs cleared;10;0x01;0x01;0x7DF;0;4;0x001;0;0;0;uint32;big endian;;viz ISO 15031-5
Odometer (celková ujetá vzdálenost);10;0x16A9;0x22;0x7E0;0;4;0x001;0;0;4;uint32;big endian;1;0;km
Particle filter: air pressure sensor 1 bank 1: raw value;1;0x162F;0x22;0x7E0;0;2;0x003;0;0;5;int16;big endian;1;0;mbar
Poloha pedálu plynu (relativní od 0 do 100 %);1;0x102A;0x22;0x7E0;0;2;0x002;0;0;6;uint16;big endian;100/8192;0;%
Throttle Position (normovaná - relativní poloha škrtící klapky);1;0xF445;0x22;0x7E0;0;1;0x002;0;0;3;uint8;100/255;0;%
Vehicle speed (rychlost vozidla);1;0x0D;0x01;0x7DF;0;1;0x002;0;0;2;uint8;1;0;km/h
    
```

Obr. 45 – Příklad vytvořeného konfiguračního souboru ve formátu .txt.

5.2 Vlastnosti načítání souboru

Při tvorbě konfiguračního souboru lze nedopatřením vytvořit zásadní chyby, které by negativně ovlivnily vyčítání dat z vozidla, nebo odesílání zpráv na sběrnici CAN bus. Proto

byl program převodníku otestován na různých variantách vadných konfiguračních souborů a byly zdokumentovány výsledky tohoto počínání (Příloha 1). Pro zobrazení výsledků jsou použity tabulky (tyto tabulky reprezentují obsah konfiguračního souboru), které obsahují pouze parametry, které jsou kritické pro vyčítání dat z vozidla a jejich odesílání na sběrnici CAN bus. Dále detaily na výpisy zpráv sériového monitoru obsahující taková data, jak je převodník načel a zpracoval z konfiguračního souboru. Pro zobrazení zpráv, které jsou odesílány na sběrnici CAN, je použit analyzátor USB2CAN a příslušný software PP2CAN. U některých případech bylo vhodné poskytnout i fotografii displeje převodníku.

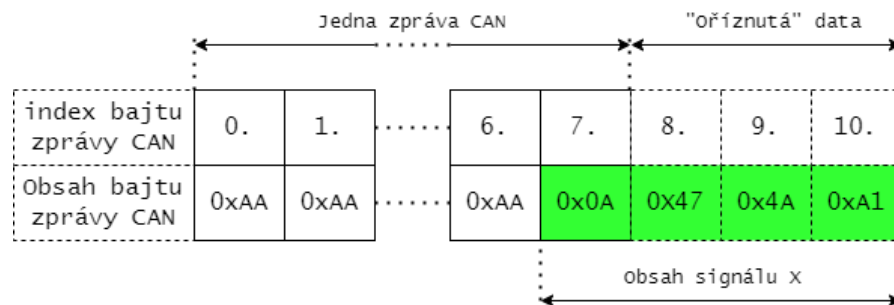
V případě vytvoření dvou (a více) naprosto identických signálů dojde k tomu, že převodník bude s těmito signály zacházet jako by byl každý ze signálů unikátní. V testovaném případě (Příloha 1. 1 - Příloha 1. 4) byly vytvořeny tři identické signály, které byly přiřazeny do jedné zprávy CAN. Čtvrtý signál se lišil pouze tím, že patřil do druhé zprávy CAN. Převodník načel 4 signály, nicméně na sběrnici se odesílají pouze dvě zprávy CAN. Program při načítání dat z konfiguračního souboru testuje, zdali se nenachází více signálů, které mají stejný parametr „PID“ a „SID“. Pokud ano, tak při žádosti o data dojde k vyčtení z vozidla pouze prvního z těchto signálů. O další signály se stejnými parametry se již nežádá, ale pouze se obsah už přijaté zprávy z vozidla zpracuje (viz podkapitola 6.3). Tato chyba tedy neovlivní odesílání zpráv CAN, data budou pouze několikrát zpracována a uložena vždy na tu samou pozici do jedné zprávy CAN.

Další případ, kdy se dva a více signálů přiřadí jedné zprávě CAN (jednomu CAN ID), ale každý z těchto signálů má rozdílné parametry „Extended“ a „Perioda odesílání [ms]“, je zdokumentovaný v přílohách této práce (Příloha 1. 5 - Příloha 1. 7). V tomto případě záleží na pořadí těchto signálů v souboru. Při procházení konfiguračního souboru převodníkem se použijí takové parametry „Extended“ a „Perioda odesílání [ms]“, které jsou u signálu s prvním výskytem daného unikátního CAN ID.

Mohlo by dojít ke kolizi dat signálů jedné zprávy CAN. Tedy že data jednoho signálu „překryjí“ data dalšího signálu, protože se mají uložit na kolizní pozici do jedné zprávy CAN (Příloha 1. 8 - Příloha 1. 11). První dva signály tvoří kolizní dvojici, další dva jsou zde přítomny, aby byl zjevný jejich obsah, kdyby ke kolizi nedošlo. V tomto případě jsou data jednoho signálu přepsána daty signálu dalšího. To, jak budou data přepsána (jestli první signál přepíše druhý, či naopak), závisí na pořadí signálů v souboru a v případě nenulové periody odesílání na tom, jak často se mají jednotlivé signály z vozidla vyčítat. Nicméně ve všech případech dochází ke znehodnocení odeslané informace.

K dalšímu znehodnocení dat odesílané zprávy dojde v případě, kdy je požadavek na uložení dat signálu na takovou pozici, na kterou se všechna data nevejdou do jedné zprávy CAN, např. uložení dat, které jsou 4 bajty dlouhé na 7. (tedy poslední) pozici zprávy CAN (Příloha

1. 12 - Příloha 1. 14). V tomto případě jsou data, která se do zprávy nevejdou, oříznuta (Obr. 46).



Obr. 46 – Oříznutí dat v případě, že data jednoho ze signálů jsou delší, než se vejde do jedné zprávy CAN.

Dále je zde popsáno, jakým způsobem se program převodníku chová v případě, kdy se za hodnoty parametrů „PID“, „SID“ a „Header requestu“ dosadí buď nesmyslné, nebo formálně správné, ale daná ECU daný SID nebo PID nezná (Příloha 1. 15 - Příloha 1. 19). Při žádosti o data každého signálu je testováno, zda je odpovědí positive response. Testuje se, zda první bajt odpovědi, tedy zadaný SID + 0x40, je roven zadanému SID + 0x40 (Např. pro zadaný SID 0x01 musí být v případě positive response v odpovědi 0x41). V případě že ano, dojde ke zpracování přijatých dat, jejich uložení do dané zprávy CAN a tento signál už nebrání v jejím odeslání. V případě že odpovědi z OBD adaptéru je cokoliv jiného než positive response (negative response, ale i „NO DATA“ nebo „?“), viz komunikace Příloha 1. 19), tak je tato odpověď klasifikována jako negative response, data se nezpracují, a hlavně se neuloží do zprávy CAN. Nicméně došlo k pokusu o vyčtení, signál je považován za aktualizovaný a zpráva může být odeslána. Pouze se zpráva odesílá s poslední hodnotou z odpovědi positive response. Z tohoto důvodu jsou na sběrnici CAN bus bez protokolu (Příloha 1. 17) zprávy, jejichž obsah bajtů je roven 0xAA. Hodnotu bajtů CAN zpráv na 0xAA nastavuje převodník jako výchozí při alokaci paměti pro uložení signálů z konfiguračního souboru. U odpovědi z vozidla v Příloha 1. 17 nikdy nebyla odpovědí positive response, tudíž se původní obsah zpráv CAN nemohl přepsat. Při každé vyhodnocené negative response se ale inkrementuje počítadlo, které takto klasifikované odpovědi započítává. Toto počítadlo se poté ukazuje na displeji převodníku obsluhy (Příloha 1. 18), která tím může identifikovat problém.

Převodník odděluje jednotlivé sloupce signálů od sebe pomocí oddělovače (znak středníku), jejich počet a správné umístění je tedy kritické pro správnou funkci převodníku při načítání konfiguračního souboru. V případě, že se v souboru nachází nějaký oddělovač navíc, nebo zde do počtu chybí, tak program nemůže spolehlivě načíst obsah souboru. V tomto případě nedojde ke konfiguraci a objeví se varovná hláška obsluhy převodníku (Příloha 1. 20).

V konfiguračním souboru chybí možnost volby komunikační rychlosti na OBD a na CANu bez protokolu. Řešením by bylo přidání dalšího parametru, např. do řádku k verzi programu, kde by šly tyto rychlosti definovat.

6 Návrh software

Tato kapitola slouží pro představení programu převodníku, použitých knihoven a položek jednotlivých menu.

Pro napsání programu převodníku bylo použito vývojové prostředí PlatformIO, což je rozšíření programu Microsoft Visual Studio Code. Zařízení bylo programováno jako vývojová platforma Arduino. Číslo verze Arduino Espressif 32 frameworku: 3.20011.230801 (2.0.11).

V programu jsou použity následující knihovny:

- M5Stack.h (základní knihovna pro programování zařízení výrobce M5Stack, číslo verze 0.4.6)
- M5ez.h (zajišťuje jednoduchou správu menu pro zařízení M5Stack, číslo verze 2.3.0)
- BluetoothSerial.h (knihovna pro zjednodušení komunikace pomocí BT, verze 2.0.0)
- Preferences.h (knihovna pro využití nevolatilní paměti zařízení k uložení nastavení, číslo verze 2.0.0)
- ELMo.h (knihovna pro zjednodušení komunikace se zařízeními s čipem ELM327, číslo verze 1.1.2)
- SD.h (slouží pro správu SD karty, verze 2.0.0)
- mcp_can.h (zjednodušuje používání CAN řadiče MCP2515, verze 1.5.1)

Za zmínku stojí to, že některé knihovny musely být náležitě upraveny tak, aby vyhovovaly této aplikaci. Jedná se o knihovny M5ez.h a ELMo.h.

Program je celý napsaný v jednom souboru s názvem „main.cpp“ (1930 řádků programu).

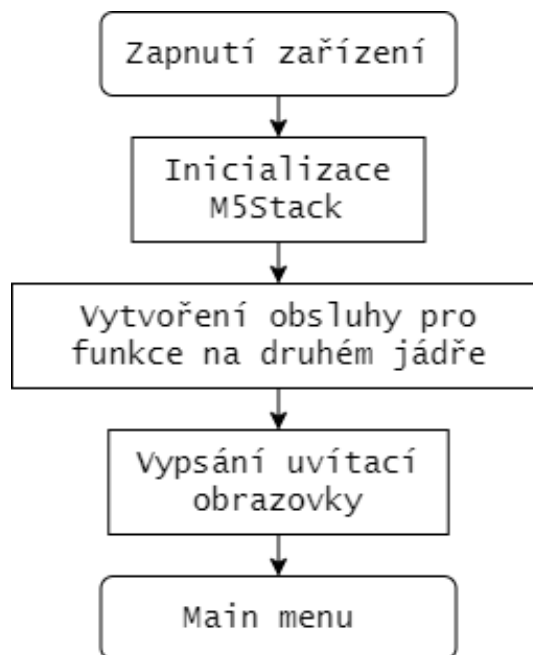
Program převodníku využívá ke své funkci obě dostupná jádra mikrokontroleru ESP32. Na první jádře se vytvořena veškerá správa menu, správa zařízení BT, komunikace s vozidlem, zpracovávání přijatých odpovědí a ukládání dat do CAN zpráv. Na druhém jádře je pouze smyčka, která zajišťuje odesílání zpráv CAN. Nachází se zde jedna SW callback funkce, viz podkapitola 6.2.



Obr. 47 – Uvítací obrazovka převodníku.

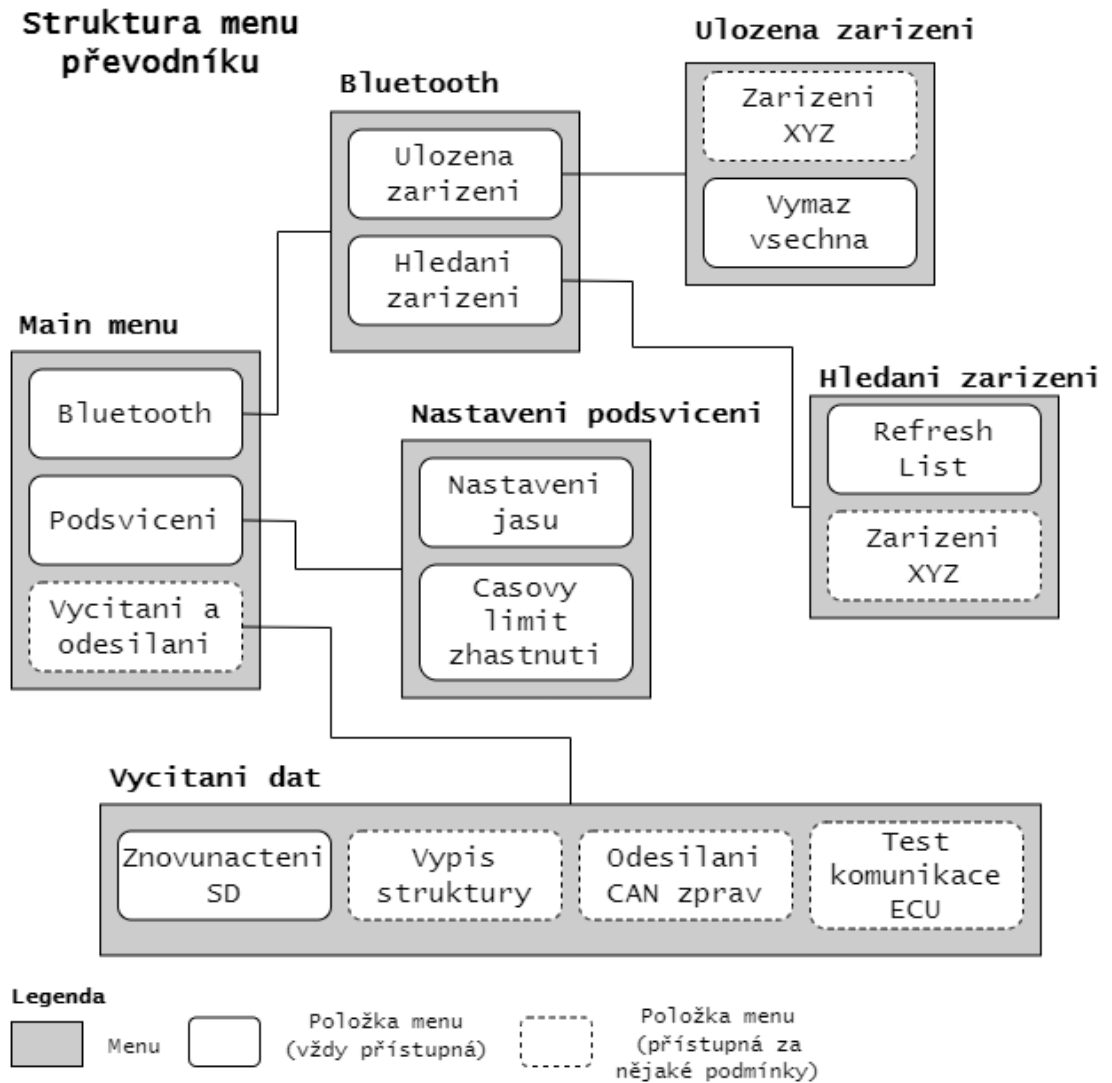
6.1 Obsluha převodníku

Převodník po celou dobu funkce využívá displeje k zobrazování informací a k lepší orientaci aplikací pomocí různých menu. Základní struktura všech menu programu a jejich položek je zobrazena na Obr. 49. Při spuštění programu dojde k nastavení zařízení (Obr. 48) a poté k zobrazení uvítací obrazovky (Obr. 47).



Obr. 48 – Spuštění programu.

Dále se program přesune do menu „Main menu“. Z tohoto hlavního menu se lze dostat do dalších menu. Prvním z nich je menu „Bluetooth“. Dalším menu je „Vycitani a odesilani“. Toto menu může zpřístupnit až čtyři funkce. Nejdůležitějšími funkcemi jsou znovunačtení SD karty (v případě změny obsahu konfiguračního souboru) a vyčítání dat z vozidla a jejich odesílání. Další dvě funkce slouží spíše ke kontrole správné funkce programu. Ulehčují diagnostiku v případě nějaké chyby v programu, nebo umožňují otestovat komunikaci s řídicí jednotkou vozidla.



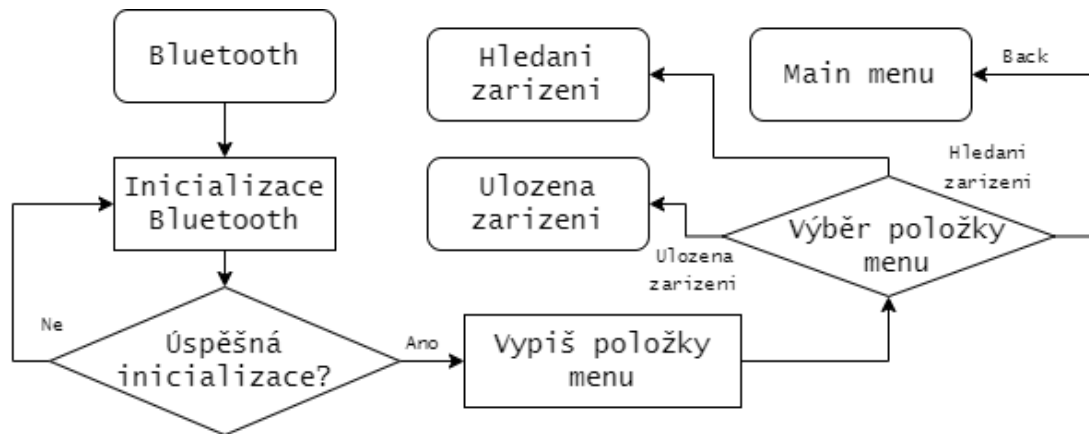
Obr. 49 – Blokové schéma struktury menu převodníku a jejich položek.

V menu „Nastavení podsviceni“ lze měnit jas obrazovky a nastavit časový limit zhasnutí displeje v minutách, v případě nečinnosti. Tato nastavení si převodník ukládá do paměti a aplikuje je při dalším zapnutí.

Dále jsou zde podrobněji popsány části programu, které jsou kritické pro kýženou funkci převodníku.

6.2 Bluetooth

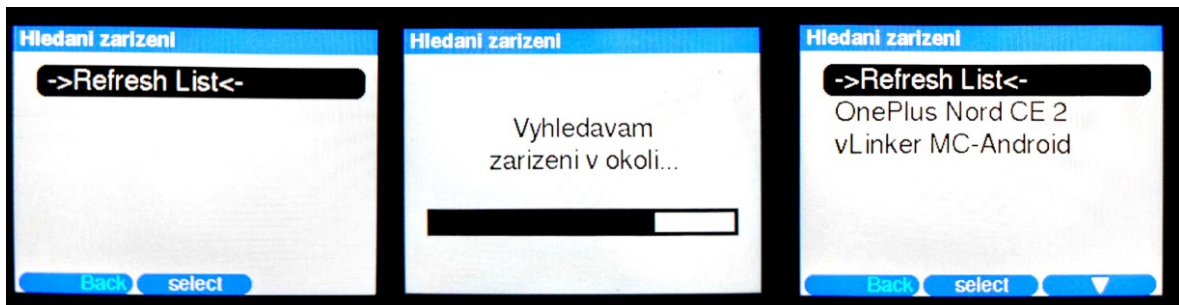
Toto menu slouží ke správě Bluetooth zařízení, tedy v tomto případě OBD adaptérů. Obsluha si může vybrat mezi tím, jestli chce vyhledat dostupné BT zařízení v okolí a připojit se k němu, nebo se připojit k už uloženému OBD adaptéru.



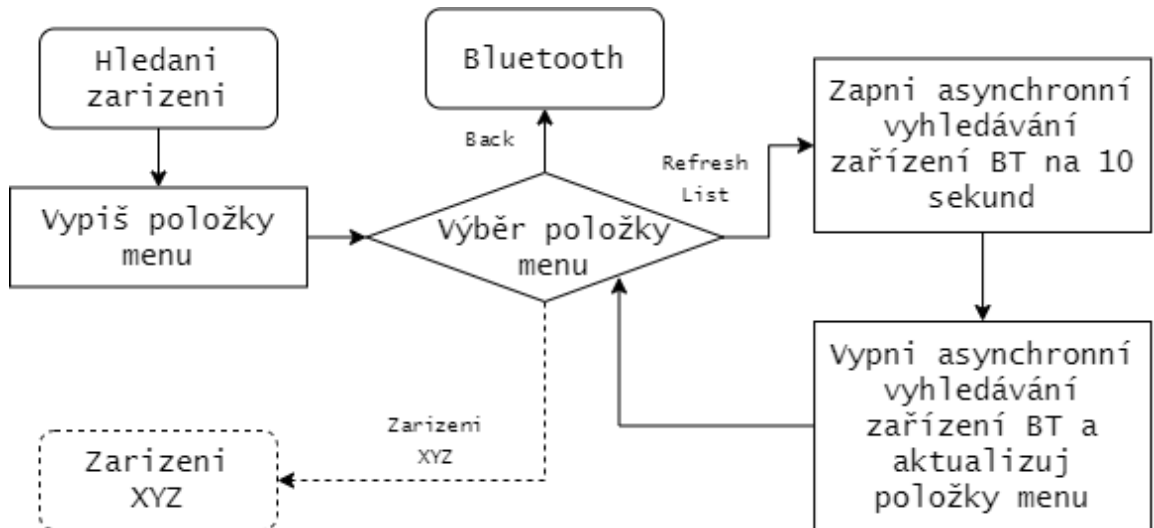
Obr. 50 – Menu Bluetooth.

Položka „Hledání zařízení“

Při požadavku na vyhledání zařízení v okolí (Obr. 52) se spustí funkce vyhledávání na 10 vteřin. V případě nalezení zařízení dojde k obsluze callback funkce a vykonání funkce OnAdvertisedDevice (Obr. 53 a Obr. 54). Po uplynutí deseti vteřin se vyhledávání ukončí a dojde k aktualizaci položek v tomto menu. Pokud došlo k nalezení nějakých zařízení v okolí, názvy těchto zařízení se vypíše jako nové položky menu „Hledání zařízení“. Proces vyhledávání zařízení je zobrazen na Obr. 51.

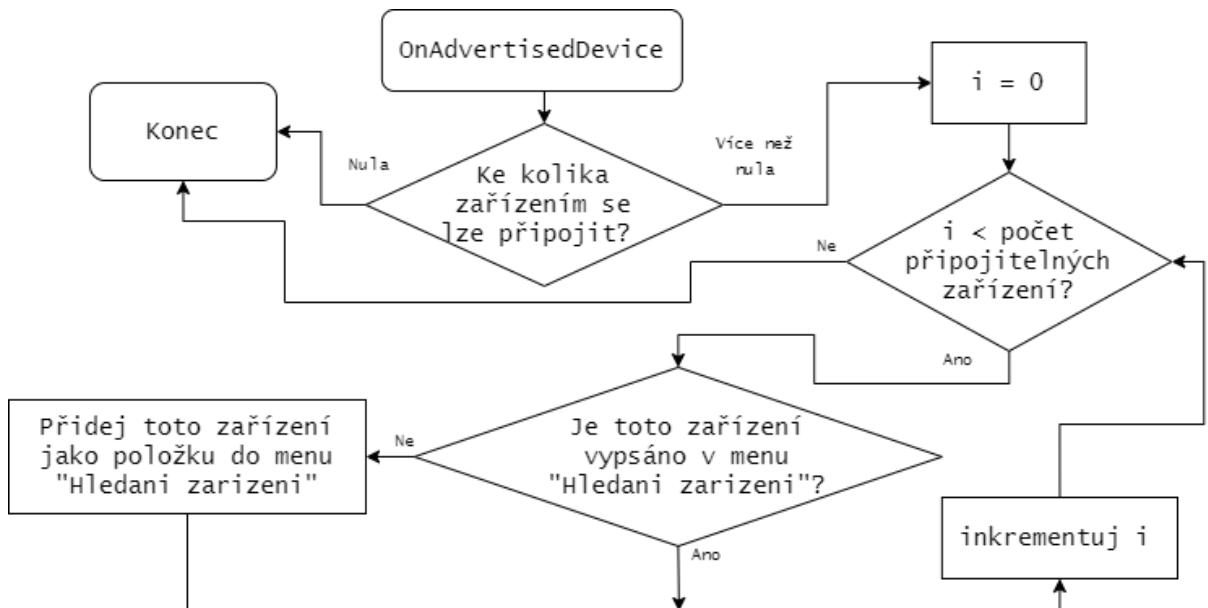


Obr. 51 – Proces vyhledávání zařízení v okolí.



Obr. 52 – Vyhledávání dostupných zařízení v okolí.

Položka „Zarizení XYZ“ dostupná pouze v případě, ve kterém došlo k nalezení alespoň jednoho BT zařízení v okolí (Obr. 52).



Obr. 53 – Obsluha callback funkce při nalezení BT zařízení při asynchronním vyhledávání.

```
void onAdvertisedDevice(BTAdvertisedDevice* device)
{
    ezMenu* cur_menu = M5ez::getCurrentMenu();

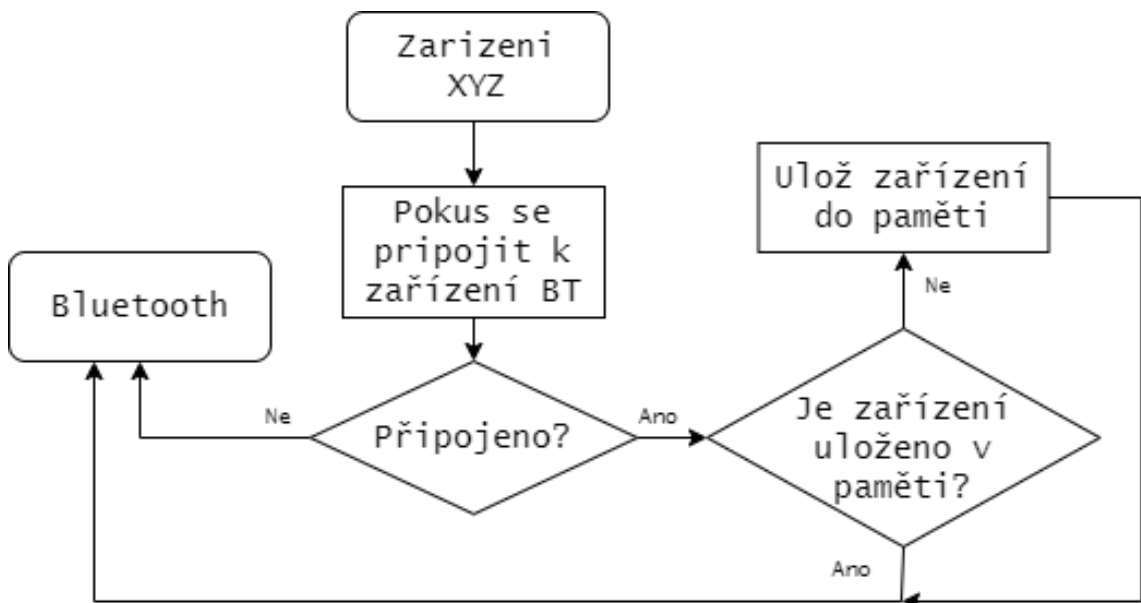
    if(btDeviceList->getCount() > 0)
    {
        int channel = 0;
        for (int i = 0; i < btDeviceList->getCount(); i++)
        {
            device = btDeviceList->getDevice(i);
            addr = device->getAddress();

            if (cur_menu->getItemNum(addr.toString().c_str()) == 0)
            {
                if (device->getName().c_str() == "" || device->getName().c_str() == " " || device->getName().c_str() == "\\0" )
                {
                    Jmeno = "nezname zarizeni";
                    cur_menu->addItem(addr.toString().c_str(), PripojZarizeni_BT);
                    cur_menu->setCaption(addr.toString().c_str(), Jmeno);
                }
                else
                {
                    cur_menu->addItem(addr.toString().c_str(), PripojZarizeni_BT);
                    cur_menu->setCaption(addr.toString().c_str(),device->getName().c_str());
                }
            }
        }
    }
}
```

Obr. 54 – Kód pro obsluhu callback funkce OnAdvertisedDevice.

Při volbě nějakého z nalezených zařízení (v diagramech zobrazeno jako zvolení položky „Zarizeni XYZ“) dojde k pokusu o připojení (Obr. 55). Tato funkce umožňuje připojení převodníku k vybranému OBD adaptéru. V případě neúspěchu se vypíše odpovídající chybové hlášení. V případě úspěchu je oznámeno, že došlo k připojení.

Po připojení se k novému zařízení se adresa daného OBD adaptéru uloží do nevolatilní paměti převodníku a další připojení už lze provádět z menu „Ulozena zarizeni“.



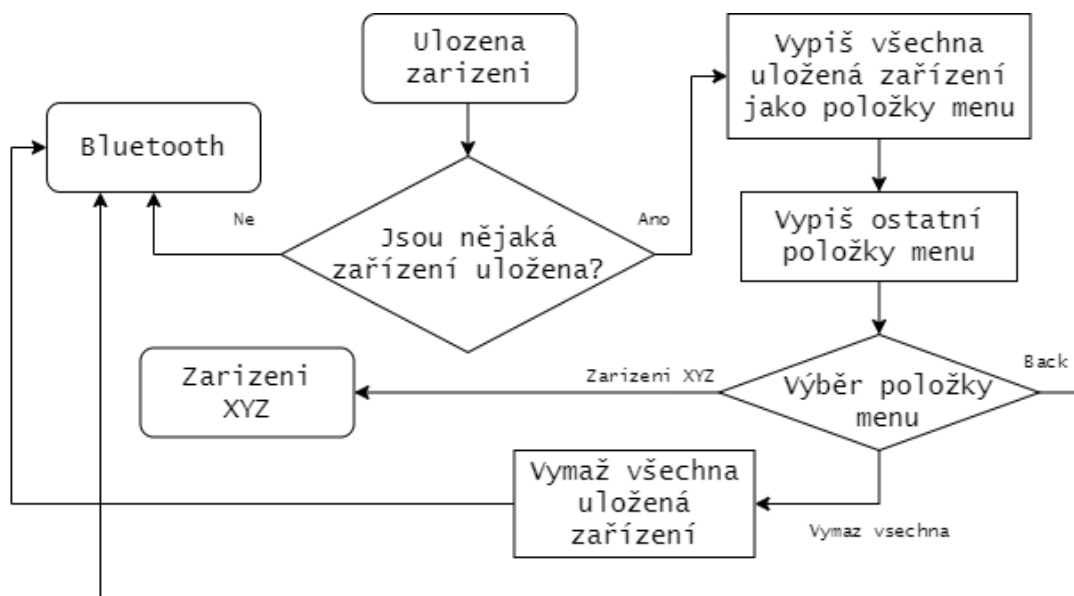
Obr. 55 – Pokus o připojení převodníku k danému OBD adaptéru (zařízení XYZ).


```
if(addr)
{
  if (SerialBT.connect(addr, channel, sec_mask, role))
  {
    ez.msgBox("", "Zarizeni pripojeno!");
    ZapamatujSparovaneZarizeni_BT(addr.toString().c_str(), cur_menu->pickCaption());
  }
  else ez.msgBox("", "Zarizeni NEpripojeno! (neni v dosahu/ nelze pripojit)");
}
```

Obr. 56 – Detail na pokus o připojení k BT zařízení.

Položka „Ulozena zarizeni“

Tato část programu umožňuje obsluhu připojení převodníku k OBD adaptéru, který je uložený v paměti zařízení (Obr. 57).



Obr. 57 – Správa uložených OBD adaptérů.

Při požadavku k připojení k danému uloženému zařízení dojde ke stejnému pokusu o připojení, jako na Obr. 55. Jen v tomto případě už nedochází k uložení daného OBD adaptéru do paměti zařízení, jelikož je toto zařízení už uloženo. Detail na displej převodníku se nachází na Obr. 59. V tomto menu lze také smazat veškerá dosud uložená zařízení.

```
if (BT.getUIInt("pocitadlo", 0) == 0)
{
    BT.end();
    ez.msgBox("", "Zadna ulozena zarizeni!", "Back");
}
else
{
    for (int i = 0; i < BT.getUIInt("pocitadlo", 0); i++)
    {
        JmenoNSV = BT.getString(("Nazev_" + String(i)).c_str());
        AdresaNSV = BT.getString(("Adresa_" + String(i)).c_str());

        sparovana.addItem(AdresaNSV, PripojZarizeni_BT);
        sparovana.setCaption(AdresaNSV, JmenoNSV);
    }
    BT.end();

    sparovana.addItem("");
    sparovana.addItem("->Vymaz vsechna<-", VymazVsechnaSparovanaZarizeni_BT);
}
}
```

Obr. 58 – Část kódu pro výpis uložených zařízení.

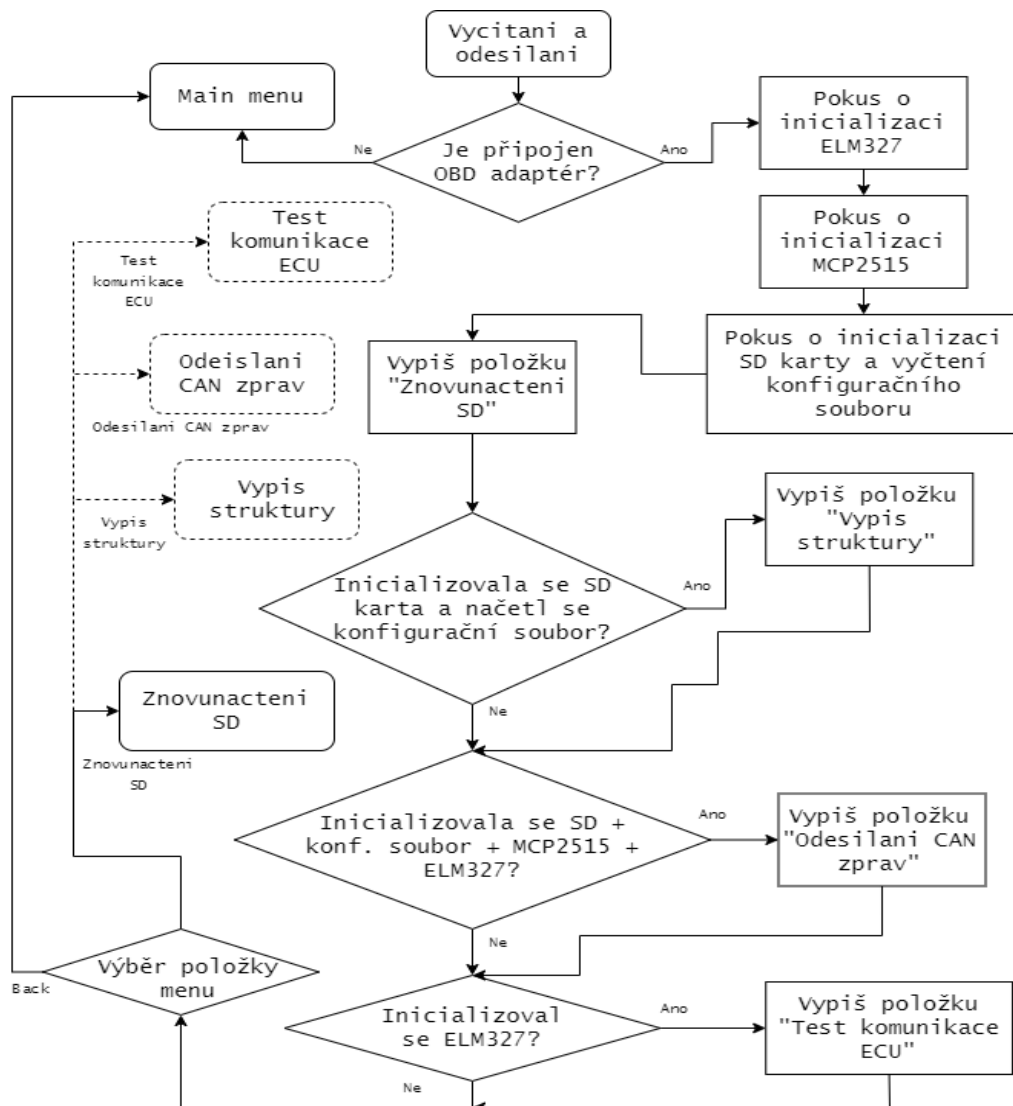
Třída „BT“ knihovny Preferences.h slouží správu nevolatilní paměti zařízení, „ez“ pochází z knihovny M5ez.h a slouží pro použití různých funkcí (např. výpisu na obrazovku) a „sparovana“ (totéž knihovna m5ez.h) slouží ke správě aktuálního menu (např. přidání položky menu a jejího názvu).



Obr. 59 – Detail na displej převodníku. Seznam uložených OBD adaptérů v nevolatilní paměti zařízení.

6.3 Vyčítání dat

Nejdůležitější funkci převodníku zajišťují položky menu „Vycitani dat“. Diagram zobrazen na Obr. 60.



Obr. 60 – Diagram pro menu „Vycitani dat“.

Toto menu je přístupné pouze po prvním využití menu „Bluetooth“ (funkce převodníku je závislá na komunikaci přes BT a předpokládá se tedy, že se obsluha nejdříve připojí k adaptéru OBD a až poté bude chtít vyčítat data z vozidla a odesílat zprávy CAN).

Inicializace částí převodníku

Toto menu zpřístupňuje své položky na základě toho, jaké části převodníku se podařily inicializovat. Po otestování, zda je převodník připojen k nějakému BT zařízení, dojde k pokusu o inicializaci ELM327 a pokus o komunikaci s řídicí jednotkou vozidla (Obr. 61). Nejdříve se převodník nastaví pomocí AT příkazů, poté se odešle žádost o data SID 0x00 PID 0x01 (toto

musí podporovat každé vozidlo). Pokud je nastavení úspěšné a vozidlo navrátí odpověď, která není negative response, inicializace proběhla v pořádku.

```
if (SerialBT.connected()) {
    if (_debug) Serial.println("ELMo: Connected to ELM327");
    _sendUninitialized("AT Z");
    _sendUninitialized("AT E0"); // disables echo
    _sendUninitialized("AT H0"); // disables headers
    if (!_sendUninitialized("AT SP 0").substring(0, 2).equals("OK")) return false;
    _sendUninitialized("AT SH 7DF"); //pridano 20.3.2024 - header jako ID requestu z ex
    if (_sendUninitialized("01 00").indexOf("UNABLE TO CONNECT") > -1) return false; //
```

Obr. 61 – Detail na inicializaci ELM327.

Nadále dochází k inicializaci řadiče MCP2515 (Obr. 62) pro odesílání zpráv na sběrnici CAN bus bez protokolu. Nastaví se komunikační rychlost sběrnice na 500 Kbps. Dojde k předání informace o tom, že řadič MCP2515, použitý v modulu COMMU, pracuje s frekvencí 8 MHz. Nastaví se tzv. ID mode na hodnotu 0, takže řadič bude pracovat jak se standartním 11 bitovým rámcem, tak s rozšířeným 29 bitovým rámcem. Pokud se řadič podaří bezchybně inicializovat a v error registru tohoto řadiče není zaregistrovaná chyba, inicializace proběhla bez závad.

```
bool kontrola = false,
do
{
    if (CAN0.begin(MCP_STDEXT, CAN_500KBPS, MCP_8MHZ) == CAN_OK) // Inicializace MCP2515
    {
        kontrola = true;
    }
    else ez.msgBox("", "Nepodarila se inicializace CAN BUS...(Restart zarizeni)"); //Mozi
} while (!kontrola);

CAN0.setMode(MCP_NORMAL); //Nastaveni modu NORMAL pro povoleni prenosu zprav
```

Obr. 62 – Detail na inicializaci MCP2515.

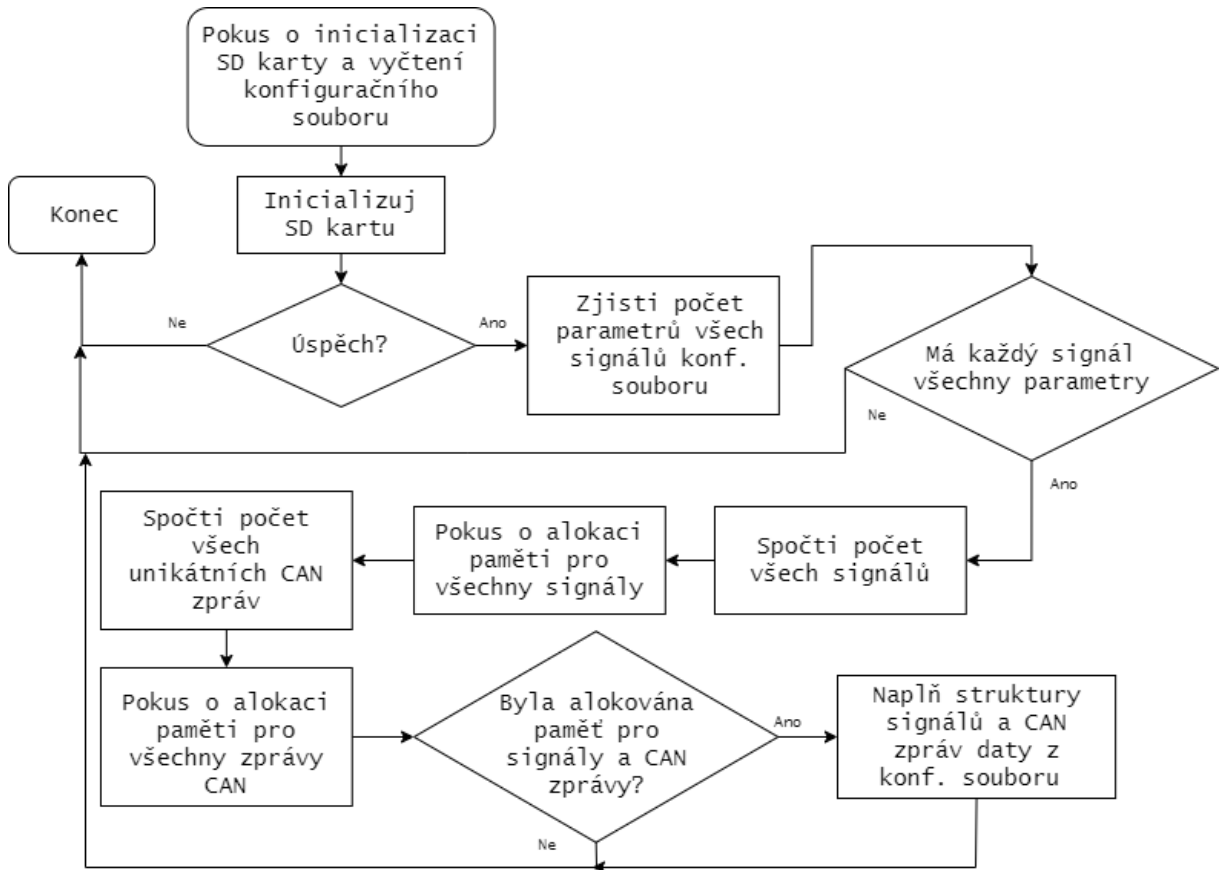
```
//vycitani konfiguraku a naplneni struktur
for (int i = 0; i < pocetSignalu; i++)//plneni struktury Signal
{
    delay(100);
    TestovaciSignal[i].jakCastoVycitat = strtol(NajdiData_SD(i, 0).c_str(), NULL, 10);//0
    TestovaciSignal[i].PID = strtol(NajdiData_SD(i, 1).c_str(), NULL, 16);//1

    switch (NajdiData_SD(i, 1).length())
    {
    case 4:
        TestovaciSignal[i].PID_length = 1;
        break;
    default:
        TestovaciSignal[i].PID_length = 2;
        break;
    }

    TestovaciSignal[i].SID = strtol(NajdiData_SD(i, 2).c_str(), NULL, 16);//2
    TestovaciSignal[i].header = strtol(NajdiData_SD(i, 3).c_str(), NULL, 16);//3
    TestovaciSignal[i].pozice_v_opovedi = strtol(NajdiData_SD(i, 4).c_str(), NULL, 10);//4
    TestovaciSignal[i].delkaSignalu = strtol(NajdiData_SD(i, 5).c_str(), NULL, 10);//5

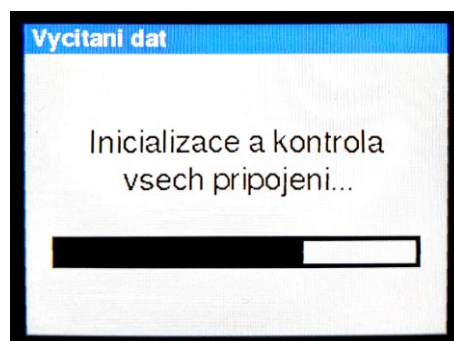
    TestovaciSignal[i].hodnotaSignalu = (uint8_t*)malloc(TestovaciSignal[i].delkaSignalu * sizeof(uint8_t));
```

Obr. 63 – Detail na plnění struktur signálů parametry z konfiguračního souboru.



Obr. 64 – Pokus o inicializaci SD karty a načtení dat z konfiguračního souboru.

Poté se zahájí pokus o inicializaci SD karty a načtení dat z konfiguračního souboru. Na Obr. 60 se jedná o blok „Pokus o inicializaci SD karty a vyčtení konfiguračního souboru“, nicméně se jedná o docela rozsáhlou operaci. Pro upřesnění je tento blok rozebrán na Obr. 64. V případě, že je konfigurační soubor ve správném formátu, podaří se alokovat paměti pro struktury signálů a CAN zpráv a k jejich správnému naplnění, inicializace SD karty proběhla bez chyby.



Obr. 65 – Detail na displej převodníku při zvolení položky „Vycitani a odesilani“.

Zpřístupnění funkcí

V tuto chvíli proběhly všechny pokusy o inicializaci veškerých nutných částí převodníku pro vyčítání dat z vozidla, jejich zpracování a následné odeslání na sběrnici CAN bus. Ale všechny funkce nevyžadují bezvadnou inicializaci všech těchto částí. Tudiž se následně položky menu vypisují na základě toho, jaké části převodníku jsou k jejich realizaci potřeba a zda jejich inicializace proběhla v pořádku.

Pokud je převodník připojen k BT zařízení, tak vždy dojde k vypsání položky „Znovunacteni SD“ (Obr. 60). Pokud inicializace SD karty a konfiguračního souboru proběhla v pořádku, zpřístupní se funkce „Vypis struktury“.

Pro (nejdůležitější) funkci vyčítání dat z vozidla a odesílání zpráv na CAN bus (položka „Odesilani CAN zprav“) je potřeba bezchybná inicializace všech částí. Nemá totiž smysl tuto funkci zpřístupňovat v případě, ve kterém nefunguje komunikace s vozidlem, nelze data spolehlivě vyčítat a zpracovávat anebo, když nejde zprávy odesílat.

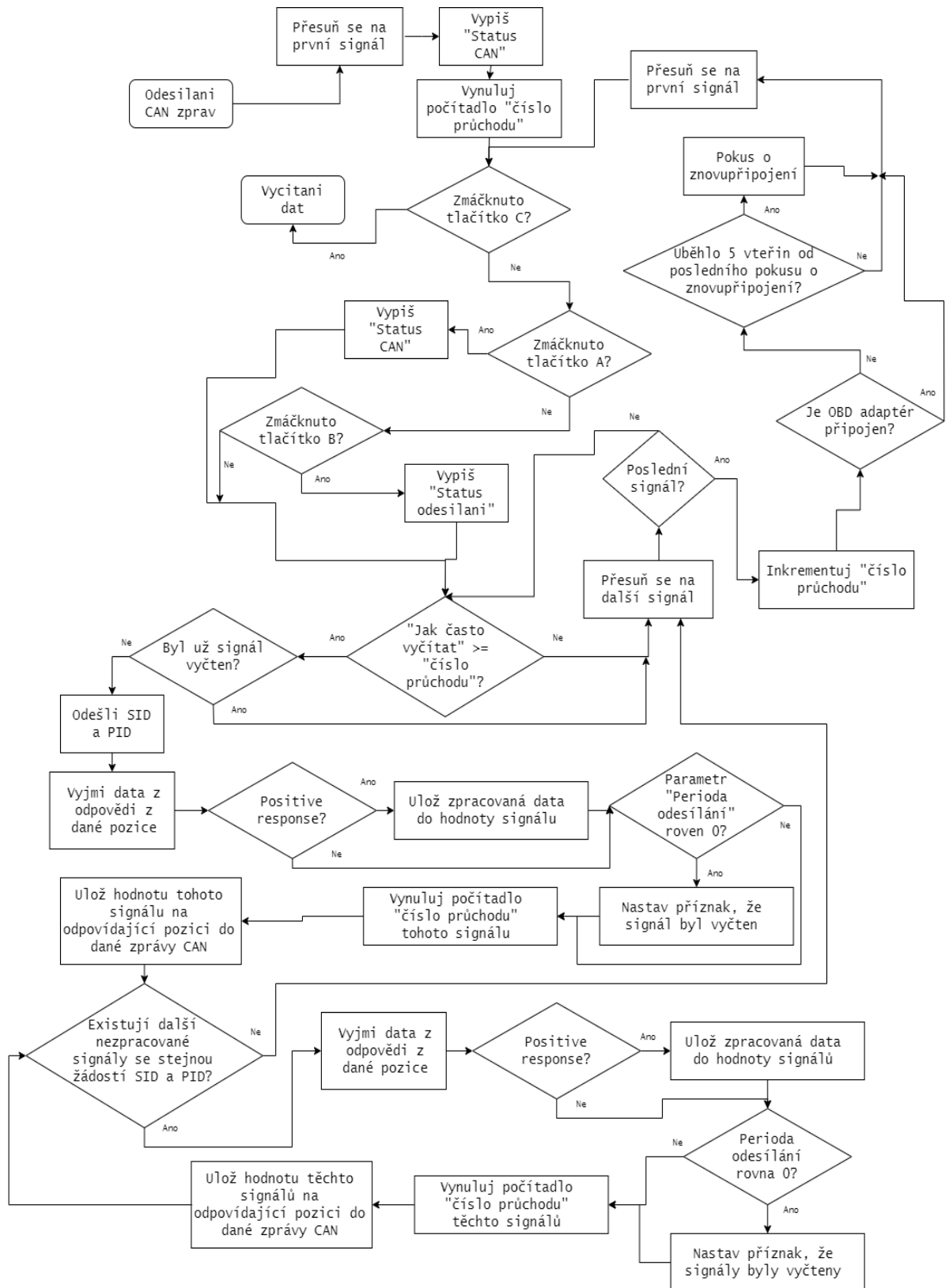
Další pomocnou funkcí, resp. položkou, je „Test komunikace ECU“. Ke zpřístupnění této funkce je potřeba spojení s OBD adaptérem a bezchybná inicializace ELM327.

Položka „Znovunacteni SD“

Při zvolení této položky z menu dojde k procesu, který je zobrazen na Obr. 64. Tedy identickému, jako při prvním spuštění menu „Vycitani dat“.

Položka „Odesilani CAN zprav“

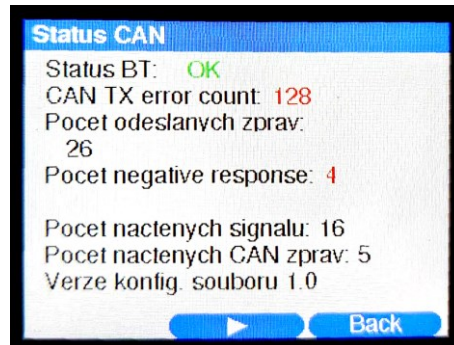
Na Obr. 66 je vidět diagram funkce pro vyčítání dat z vozidla a jejich přípravu k odeslání.



Obr. 66 – Diagram pro vyčítání dat z vozidla a jejich přípravu na odeslání na sběrnici CAN bus.

Tato položka menu spouští proces vyčítání dat z OBD zásuvky, jejich ukládání do CAN zpráv na CANu bez protokolu a následné odeslání. Tato položka umožňuje dva typy zobrazení

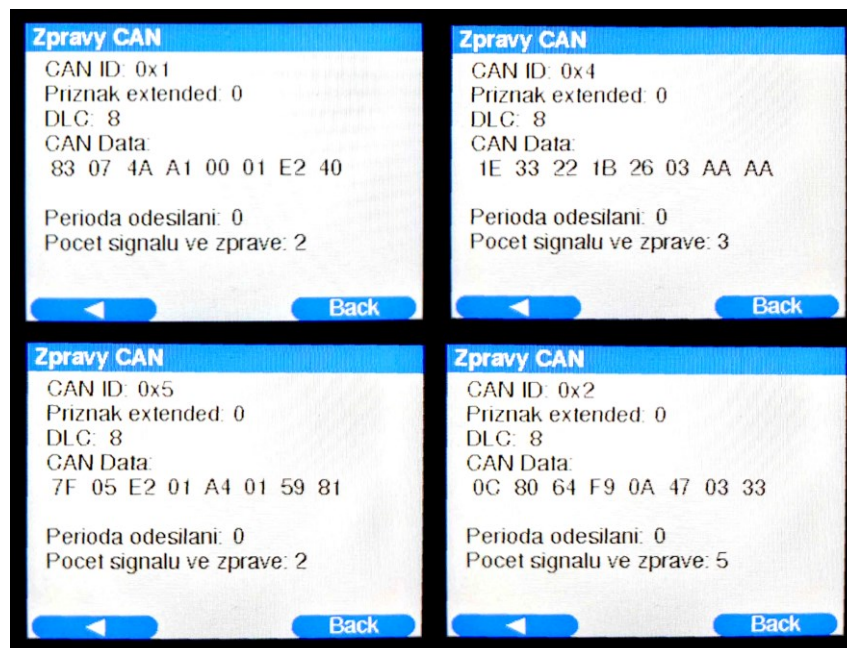
informací na obrazovku. Při spuštění této položky dojde k vypsání „Status CAN“ (Obr. 67). Je zde vidět status připojení s OBD adaptérem, počet errorů při odesílání zpráv CAN (z error registru řadiče MCP2515), celkový počet odeslaných zpráv, počet negativních odpovědí z vozidla, počet načtených signálů a CAN zpráv z konfiguračního souboru a číslo verze konfiguračního souboru.



Obr. 67 – Status CAN. Detail na displej převodníku.

Po vypsání tohoto zobrazení se program ubere do smyčky vyčítání a zpracovávání dat z vozidla. Na začátku smyčky se testuje, zda nebylo zmáčknuto nějaké z tlačítek převodníku. Pomocí tlačítek lze přepínat mezi zobrazeními, kde tlačítko A slouží pro zobrazení „Status CAN“ (Obr. 67) a tlačítko B pro zobrazení „Zpravy CAN“ (Obr. 68), a nakonec tlačítko C, které slouží pro ukončení vyčítání a zpracovávání dat (k ukončení chodu této funkce). Pokud bylo nějaké tlačítko zmáčknuto, daná funkce se vykoná.

Na další obrazovce "Zpravy CAN" jsou informace o odesílaných CAN zprávách, jako např. ID zprávy a její data.



Obr. 68 – Zprávy CAN. Detail na displej převodníku.

Při každém průchodu smyčkou se inkrementuje počítadlo průchodů každého signálu. Pokud je tedy parametrem signálu „Jak často PID vyčítat“ např. hodnota 10, tak dojde k vyčtení při 10. průchodu, a následnému vynulování počítadla průchodů. Vyčtením je rozuměno nejdříve kontrola, zda byl již tento signál vyčten. Pokud nebyl, tak dojde k žádosti o data z vozidla. Dojde ke zpracování odpovědi (dle polohy a délky vyčítaných dat) a kontroluje se, zda byla odpověď positive response. Pokud ano, dojde k uložení takto zpracovaných dat jako nové hodnoty signálu. Pokud ne, původní hodnota signálu zůstává stejná.

Pokud se jedná o signál, jehož zpráva CAN se nemá odesílat periodicky (parametr „Perioda odesílání [ms]“ rovna nule), tak se nastaví příznak tohoto signálu jako vyčten. U signálů s periodickým odesíláním zprávy CAN se to neděje. Není u nich požadavek na to, aby se každý ze signálů jednou vyčetl, a aby se v případě všech aktualizovaných signálů zprávy mohla zpráva CAN odeslat. Signály periodických zpráv se tedy mohou vyčítat nehledě na to, jestli už byly vyčteny, či nikoliv.

Po vynulování počítadla průchodů signálu se hodnota signálu uloží na danou pozici do dané zprávy CAN (Obr. 69).

```
TestovacíSignal[i].cisloPruchodu = 0; //Vynulovani cisla pruchodu

for (int j = 0; j < TestovacíSignal[i].delkaSignalu; j++) //Rozklad hodnoty signalu na bajty a jejich ukladani do odpovidajici
{
    TestovacíSignal[i].kam -> CAN_data[TestovacíSignal[i].polohaSignaluZpravyCAN + j] = TestovacíSignal[i].hodnotaSignalu[j];
}
```

Obr. 69 – Uložení hodnoty signálu na příslušnou pozici dané CAN zprávy.

Program nadále zjišťuje, zda neexistuje alespoň jeden další signál, který má totožnou žádost SID a PID (Obr. 70). Může se jednat o dva případy.

V odpovědi vozidla na daný PID se totiž mohou přenášet hodnoty více signálů najednou. Jedná se např. o teploty nasávaného vzduchu (PID 0x68), či teploty spalin (PID 0x78). V prvním případě se může obsluha převodníku rozhodnout např. jednu z teplot nasávaného vzduchu odeslat v jedné CAN zprávě, a další z teplot v CAN zprávě druhé. Poté je potřeba vytvořit dva signály, ve kterých se budou shodovat parametry „PID“ a „SID“. V druhém případě může jít o chybu při vytváření konfiguračního souboru (případ je popsán v podkapitole 5.2).

Pokud takový signál neexistuje, popsáný proces se opakuje pro následující signál. Pokud ano, tak už se o tato data nežádá (protože je převodník už jednou vyžádal a má je dočasně uložené v paměti), ale dojde pouze ke zpracování. A to stejným způsobem, který byl doteď popsán, mimo žádosti o data z vozidla. Tedy zpracování odpovědi, uložení dat jako hodnoty signálu, nastavení, resp. nenastavení příznaku vyčten. Vynulování počítadla a uložení hodnoty signálu do příslušné zprávy CAN. Po zpracování těchto duplicitních signálů dojde k přesunu na další signály a jejich případnému vyžádání a zpracování.

```
Signal* ptr = TestovaciSignal[i].dalsi;
while (ptr != NULL) //hledani signalu se stejnym PIDem, aby doslo k jeho zpracovani rovnou a nezadalo se o nej znovu
{
    vysledek = ZpracujRawResponse_CAN(ptr -> pozice_v_opovedi, ptr -> delkaSignalu, &TestovaciSignal[i]);

    if (!negativeResponse)
    {
        for (int j = 0; j < ptr -> delkaSignalu; j++)
        {
            ptr -> hodnotaSignalu[j] = strtol(vysledek.substring(pocitadlo, pocitadlo + 2).c_str(), NULL, 16);
            pocitadlo += 2;
        }
        pocitadlo = 0;
    }

    for (int j = 0; j < ptr -> delkaSignalu; j++)
    {
        ptr -> kam -> CAN_data[ptr -> polohaSignaluZpravyCAN + j] = ptr -> hodnotaSignalu[j];
    }

    ptr -> vycten = true;
    ptr -> cisloPruchodu = 0;
    ptr -> kam -> pocetAktualizovanychSignalu++;
    ptr = ptr -> dalsi;
}
}
```

Obr. 70 – Detail na hledání a zpracovávání signálů se stejným PID a SID.

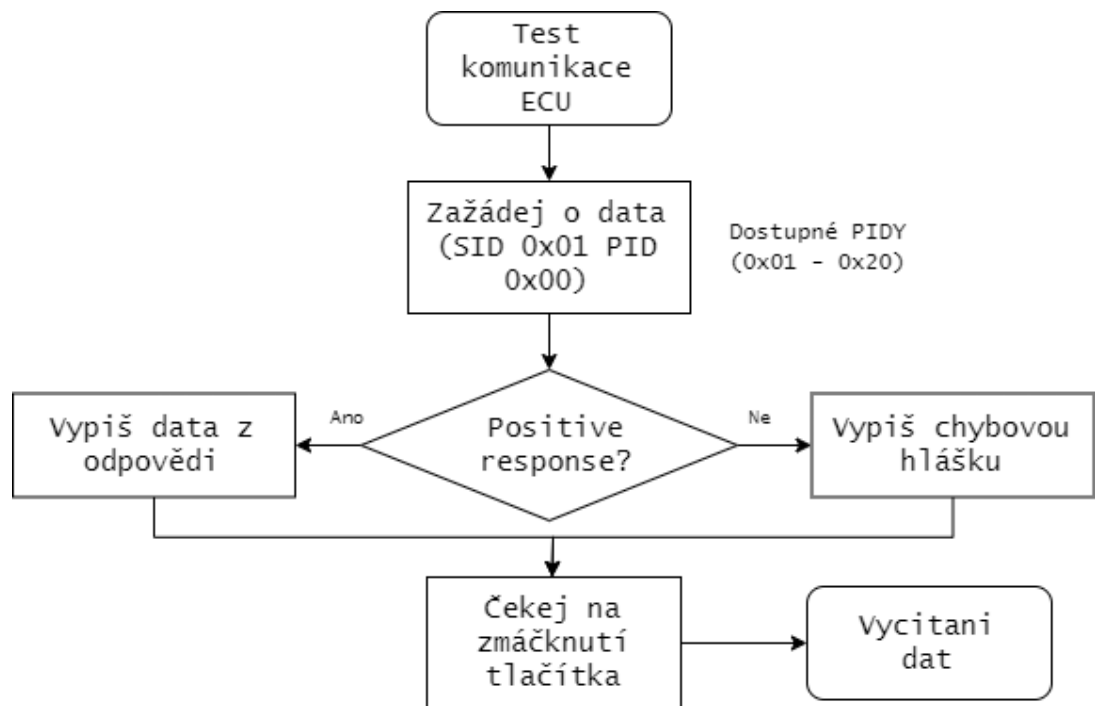
Po takovémto projití všech načtených signálů, je program téměř na konci jednoho průchodu smyčkou. V této části se zjišťuje, zda je převodník stále připojen k OBD adaptéru (např. kvůli zrušení spojení v případě ztráty signálu). Pokud ne, tak se převodník každých pět vteřin pokusí o nové navázání spojení. Pokud ano, tak se právě vykonal jeden průchod smyčky a celý proces může začít od znovu, viz Obr. 71.

```
unsigned long aktualniCasZnovupripojeni = millis();
if (!statusBT)
{
    if (aktualniCasZnovupripojeni - predchoziPokusOPripojeni >= periodaPokusuOPripojeni)
    {
        predchoziPokusOPripojeni = aktualniCasZnovupripojeni;
        Serial.println("Probiha pokus o znovupripojeni...");
        ZnovupripojeniZarizeni_BT();
    }
}
}
```

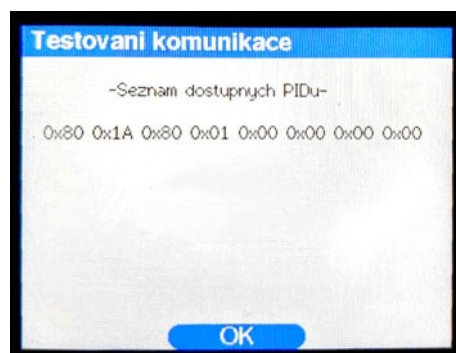
Obr. 71 – Pokus o připojení zařízení v případě ztráty spojení.

Položka „Test komunikace ECU“

Tato funkce umožňuje rychlou kontrolu funkčnosti OBD adaptéru a komunikaci s vozidlem. Převodník si zažádá o PID 0x00 služby SID 0x01 (tedy seznam dostupných PIDů od 0x01 až po 0x20). Tento parametr lze vyčíst u každého vozidla. V případě pozitivní odezvy vozidla se obsah odpovědi vypíše na obrazovku (Obr. 73). V opačném případě se na obrazovku vypíše chybová hláška.



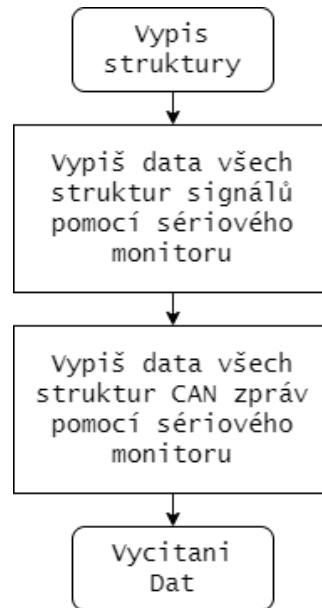
Obr. 72 – Test komunikace s vozidlem.



Obr. 73 – Vypsání seznamu dostupných PIDů (0x00 – 0x20). Detail na displej převodníku.

Položka „Vypis struktury“

Položka umožňuje přes USB převodníku odeslat do připojeného PC obsah parametrů načtených signálů v textové podobě. Tato funkce usnadňovala vývoj aplikace, hodí se pro kontrolu správného chodu programu a umožňuje snadnější nalezení příčiny v případě výskytu nějaké chyby programu.

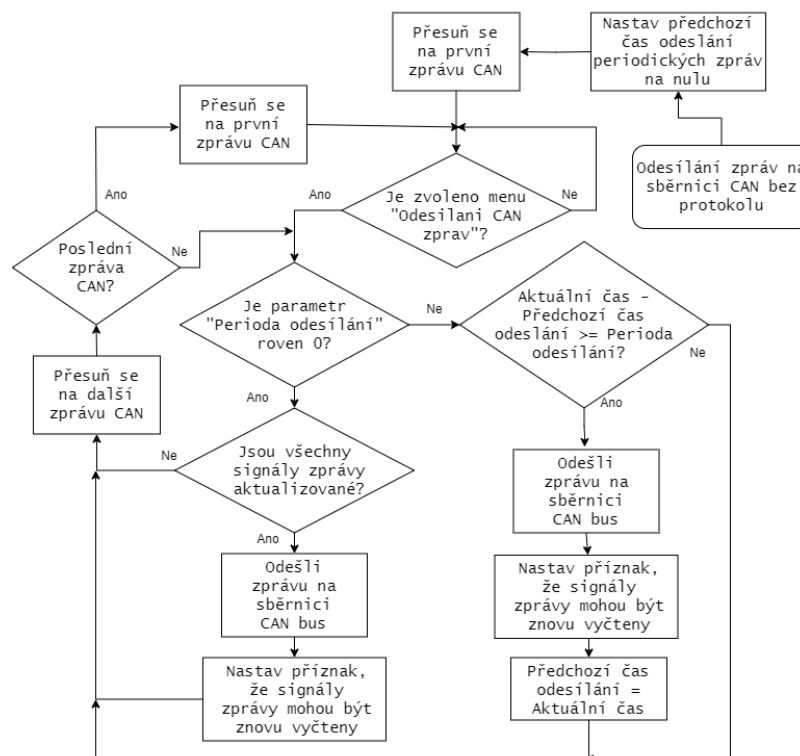


Obr. 74 – Vypsání obsahu vytvořených struktur signálů a CAN zpráv z paměti převodníku.

Příklad takového výpisu se nachází na několika obrázcích v Příloha 1.

6.4 Odesílání zpráv na CAN bus

Na Obr. 66 je vidět diagram funkce, která má na starost ukládání dat vyčtených z OBD zásuvky do CAN zpráv bez protokolu a jejich odesílání.



Obr. 75 – Odesílání zpráv na sběrnici CAN bus.

Aby se mohlo zajistit periodické odesílání CAN zpráv, vykonává se samotné odesílání na druhém jádře mikrokontroleru. Odesílání tedy nebrzdí žádosti o data, jejich zpracování a vypisování informací na obrazovku. Tato funkce se skládá z jedné nekonečné smyčky, ve které dochází k odesílání periodických i neperiodických zpráv (Obr. 75).

Periodické zprávy CAN jsou tedy odesílány vzhledem k jejich dané periodě. Program udržuje v paměti poslední čas odeslání každé periodické zprávy a aktuální čas (to je čas v milisekundách od spuštění mikrokontroleru, získaný pomocí funkce `millis`). Při každém pokusu o odeslání periodické zprávy se zjišťuje, zda už došlo k okamžiku jejího odeslání, tedy zdali je rozdíl aktuálního času a předchozího času odeslání větší, nebo rovný periodě odeslání tohoto signálu (Obr. 76). Pokud je to pravda, dojde k odeslání zprávy a aktualizaci hodnoty posledního odeslání zprávy. Pokud to pravda není, nedojde k odeslání zprávy a program se přesune k pokusu odeslání další zprávy.

U neperiodických signálů musí před odesláním dojít k tomu, že jsou všechny hodnoty signálů dané zprávy před odesláním aktualizovány (vyčteny, zpracovány a uloženy).

```

if (spustOdesilaniCAN)
{
    for (int i = 0; i < pocetUnikatnichZpravCAN; i++)
    {
        AktualniCasCAN = millis();

        if (CAN[i].periodaOdesilani == 0)
        {
            if (CAN[i].pocetAktualizovanychSignalu == CAN[i].pocetSignalu)
            {
                SendMessage_CAN(&CAN[i]);
                CAN[i].pocetAktualizovanychSignalu = 0;

                for (int j = 0; j < pocetSignalu; j++)
                {
                    if (TestovaciSignal[j].kam == &CAN[i])
                    {
                        TestovaciSignal[j].vycten = false;
                    }
                }
            }
        }
        else //KonfiguracniSoubor se odesila s nejakou periodou
        {
            if (AktualniCasCAN - CAN[i].predchoziCasOdeslani >= CAN[i].periodaOdesilani)
            {
                SendMessage_CAN(&CAN[i]);
                CAN[i].predchoziCasOdeslani = AktualniCasCAN;
            }

            for (int j = 0; j < pocetSignalu; j++)
            {
                if (TestovaciSignal[j].kam == &CAN[i])

```

Obr. 76 – Detail na část kódu, která zajišťuje odesílání CAN zpráv.

```
canFrame_t frm; //promenna typu struktura obsahujici CAN_ID, CAN_DLC a C
frm.id = CAN->CAN_ID & 0x1FFFFFFF; //ID, maska pro 29bit id

if (CAN->priznakExtended) ///29 bit ID
{
    frm.id |= (1<<31); //pokud je id delsi než 11bit, tak bit IDE = 1 =>
}
else
{
    frm.id |= (0 << 31);
}

frm.data[0] = CAN->CAN_data[0];
frm.data[1] = CAN->CAN_data[1];
frm.data[2] = CAN->CAN_data[2];
frm.data[3] = CAN->CAN_data[3];
frm.data[4] = CAN->CAN_data[4];
frm.data[5] = CAN->CAN_data[5];
frm.data[6] = CAN->CAN_data[6];
frm.data[7] = CAN->CAN_data[7];
//frm.data[7] = CAN->pocitadloTest;
frm.dlc = CAN->DLC; // Pocet odesilanych bytu ve zprave CAN

error = CAN0.sendMsgBuf(frm.id, frm.dlc, frm.data); //odeslani zpravy

if (error == CAN_GETTXBFTIMEOUT)
```

Obr. 77 – Detail na pokus o odeslání zprávy na CAN bez protokolu.

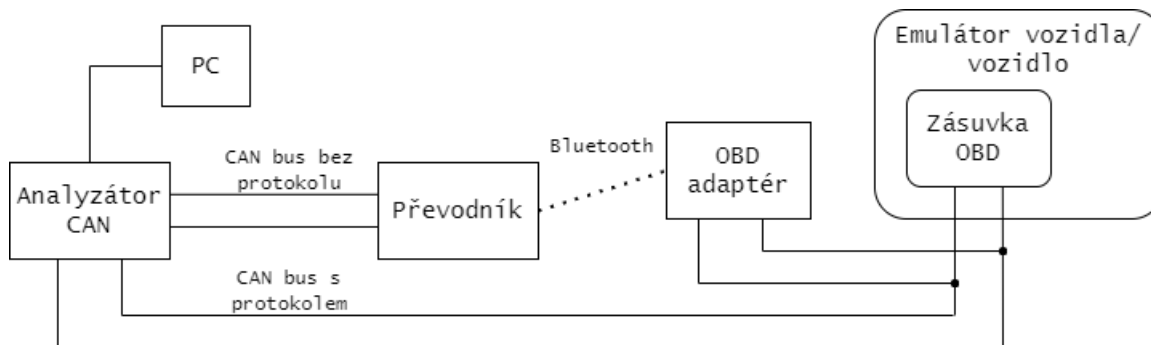
```
while (!pomoc)
{
    aktualniCas = millis();

    if ((CAN0.sendMsgBuf(frm.id, frm.dlc, frm.data) == CAN_OK) || (aktualniCas - predchoziCas >= 1))
    {
        pomoc = true;
    }
}
```

Obr. 78 – Pokus o odeslání zprávy na CAN bez protokolu, pokud poprvé nešlo zprávu odeslat.

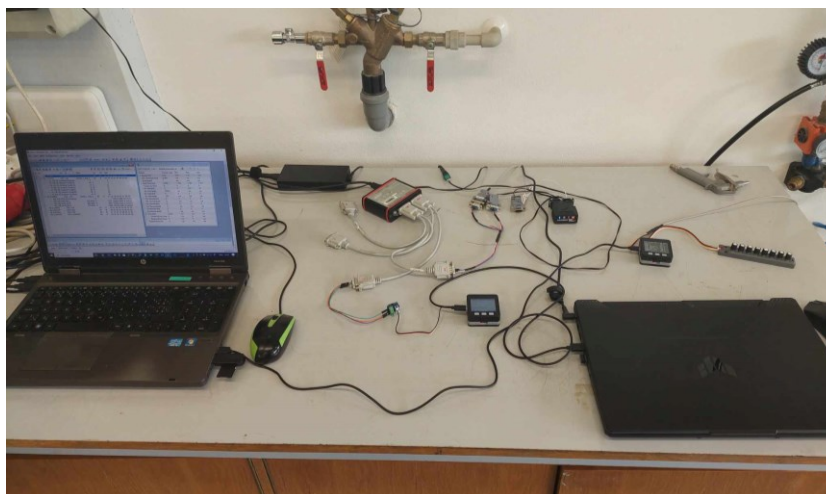
7 Ověření funkce zařízení

V této kapitole je zdokumentováno konečné testování funkce převodníku. Převodník byl otestován nejprve na emulátoru vozidla, který poskytl vedoucí práce. Ten byl používán při tvorbě programu pro převodník. Dále bylo provedeno testování přímo na vozidle.



Obr. 79 – Blokové schéma zapojení pro testování převodníku.

Blokové schéma zapojení převodníku při testování je zobrazeno na Obr. 79. Použitý analyzátor CAN umožňuje připojení více kanálů, tudíž byla sledována komunikace na CAN bus z výstupu převodníku (bez protokolu) pomocí jednoho kanálu, a komunikace na sběrnici na výstupu zásuvky OBD (CAN bus s protokolem) pomocí kanálu druhého. Tím je možné sledovat, jaká data poskytuje vozidlo a jaká data odesílá převodník.



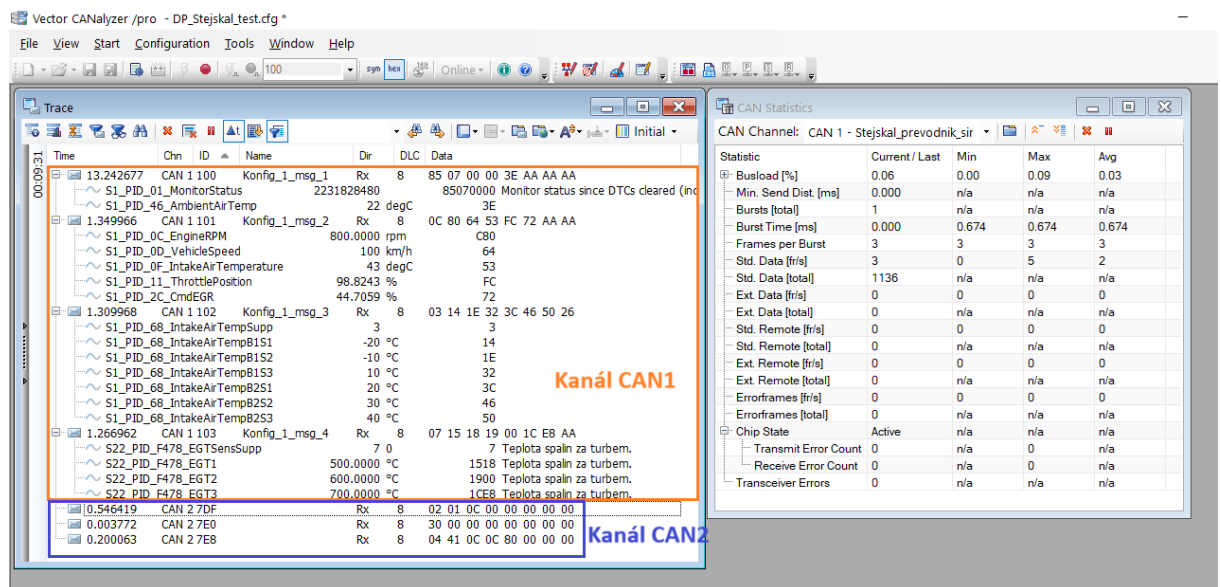
Obr. 80 – Fotografie z testování převodníku na emulátoru vozidla.

K zobrazení dat z CAN analyzátoru byl použit software Vector CANalyzer. Lze zde zobrazit jednotlivé zprávy a jejich obsah. Dále obsahu těchto zpráv přiřadit jednotlivé signály dle polohy ve zprávě a délky dat (v bajtech, či jednotlivé bity) a přiřadit jim význam, nebo je přepočítat a zobrazit jako fyzikální veličinu. Tento program umožňuje zobrazení takto vytvořených signálů v čase.

7.1 Emulátor vozidla

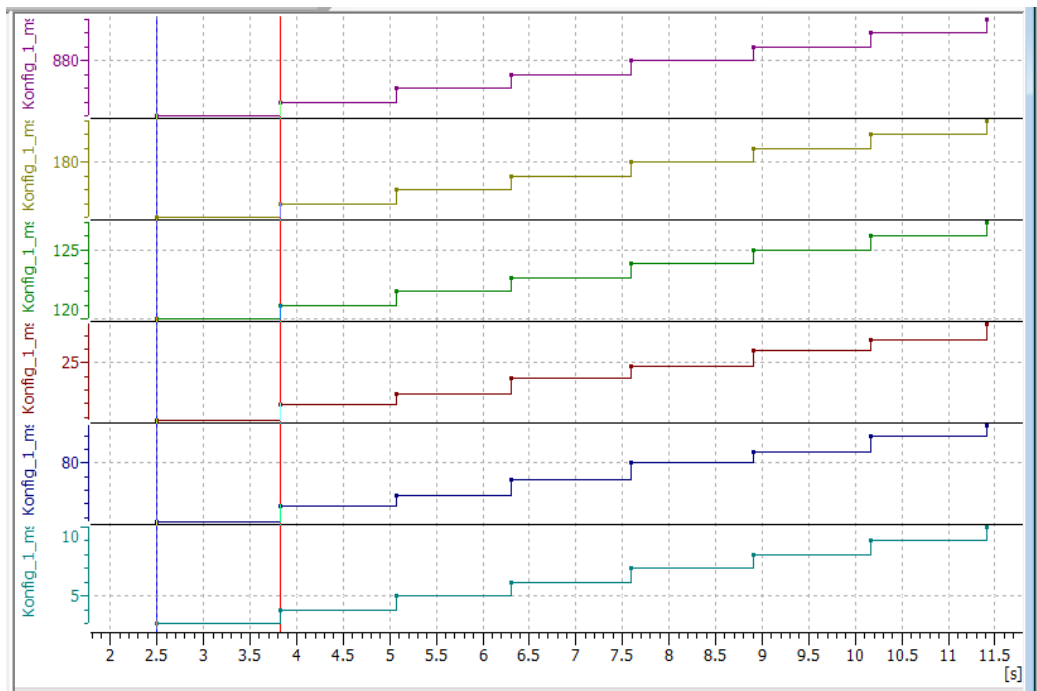
Výhoda testování na emulátoru vozidla oproti testování na vozidle je ta, že se dají poskytnutá data měnit pouhým přeprogramováním v průběhu testování. Příklad dat, které tento emulátor poskytuje, je zobrazen v Příloha 2. 1.

Příklad zobrazení z tohoto softwaru je na Obr. 81. Kanál slouží pro zobrazení dat vysílané převodníkem. Kanál CAN2 je použit pro zobrazení dat ze sběrnice CAN na zásuvce OBD (tedy komunikace převodníku s vozidlem). Na tomto obrázku je vidět první testování převodníku. Toto testování sloužilo ke zjištění, jestli převodník správně načte konfigurační soubor, jestli žádá o data na OBD zásuvce a jestli bezchybně odesílá zprávy na sběrnici CAN (kanál CAN1).



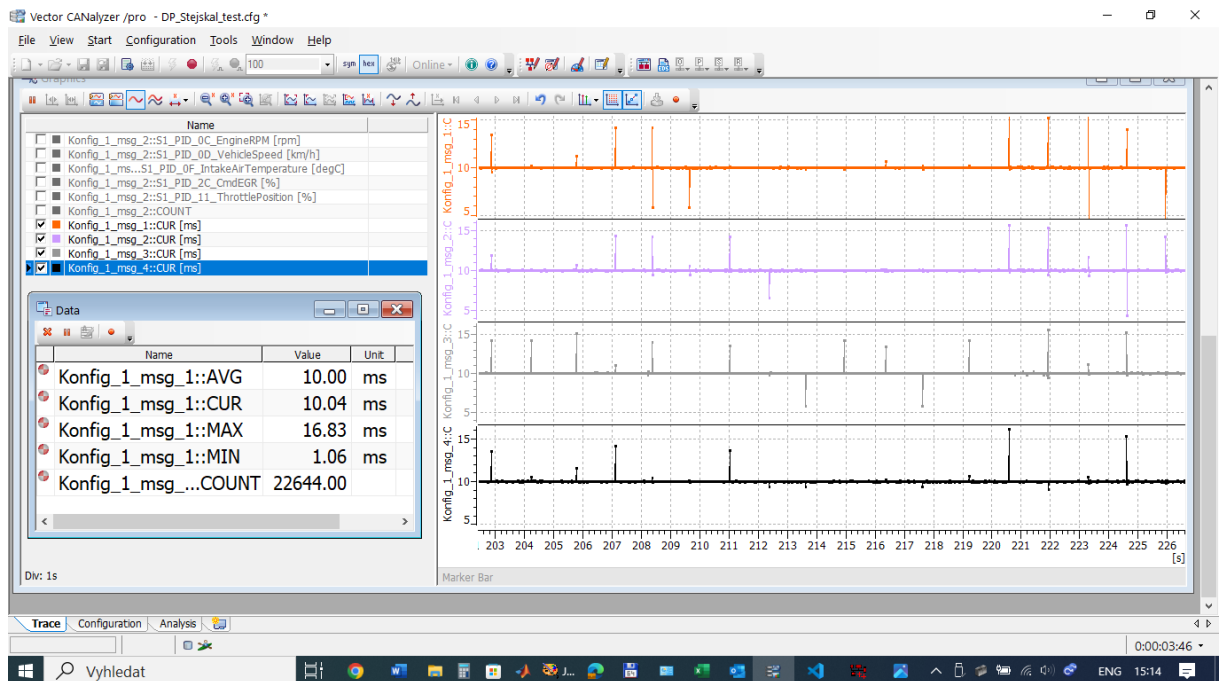
Obr. 81 – První otestování převodníku. Data z Vector CANalyzer. Kanál CH1 pro zprávy z převodníku, kanál CH2 pro zprávy z OBD zásuvky.

Dále se testovalo to, zda v případě neperiodických signálů, převodník opravdu aktualizuje všechny signály dané CAN zprávy před jejím odesláním. Emulátor byl přeprogramován tak, aby se jím poskytovaná data při vyčtení změnila a to tak, aby navýšila svoji hodnotu. Pokud v každé následující přijaté zprávě jsou data, která budou mít vyšší hodnotu než data ve zprávě předešlé, tak tedy dochází ke správnému odesílání zpráv, tedy převodník odešle CAN zprávu až po tom, co se v ní aktualizovala všechna data, která přenáší. Na Obr. 82 je vidět obsah přijaté zprávy v čase.

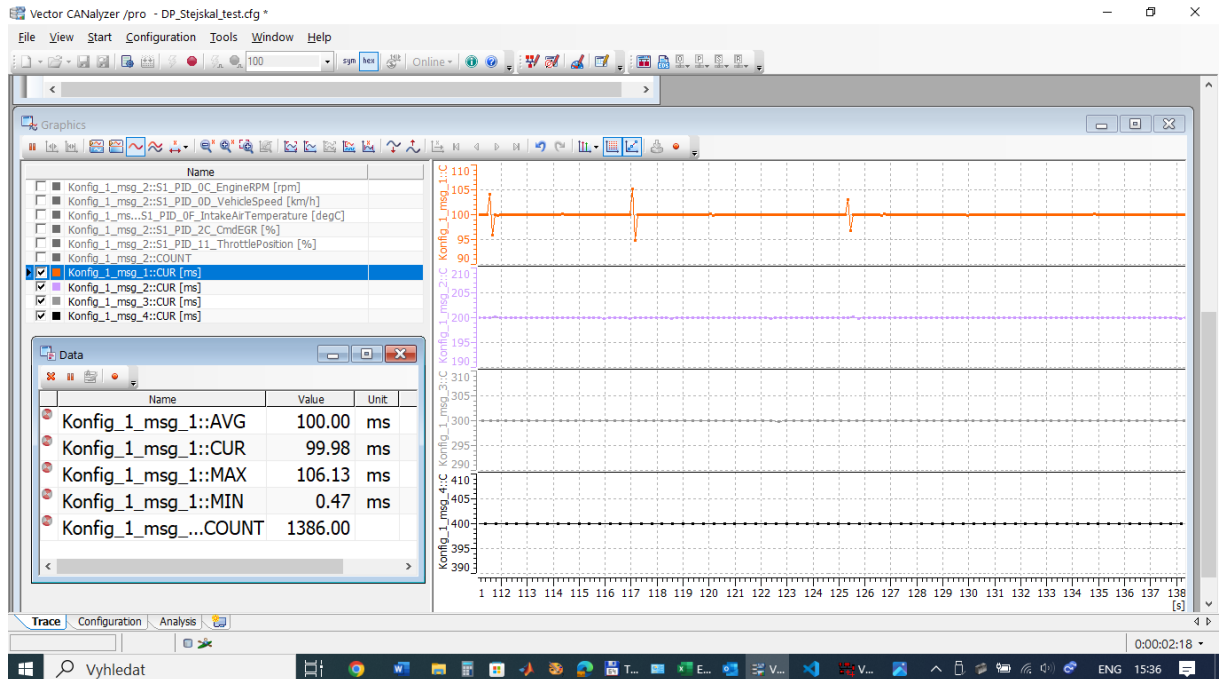


Obr. 82 – Testování správného odesílání neperiodických zpráv. Osa y představuje hodnotu dat, bez rozměru.

Software umožňuje zobrazení periody přijatých zpráv v čase (kde na ose y je perioda příjmu zprávy v milisekundách a na ose x čas ve vteřinách). Díky této funkci lze otestovat, jestli se periodické zprávy odesílají s požadovanou periodou. Byly dvakrát otestovány 4 zprávy CAN. Poprvé měla každá ze zpráv periodu odesílání 10 ms (Obr. 83). Podruhé se pro zprávy nastavily periody 100, 200, 300 a 400 ms (Obr. 84).



Obr. 83 – Test periody odesílání zpráv. Každá ze zpráv odesílána s periodou 10 ms.



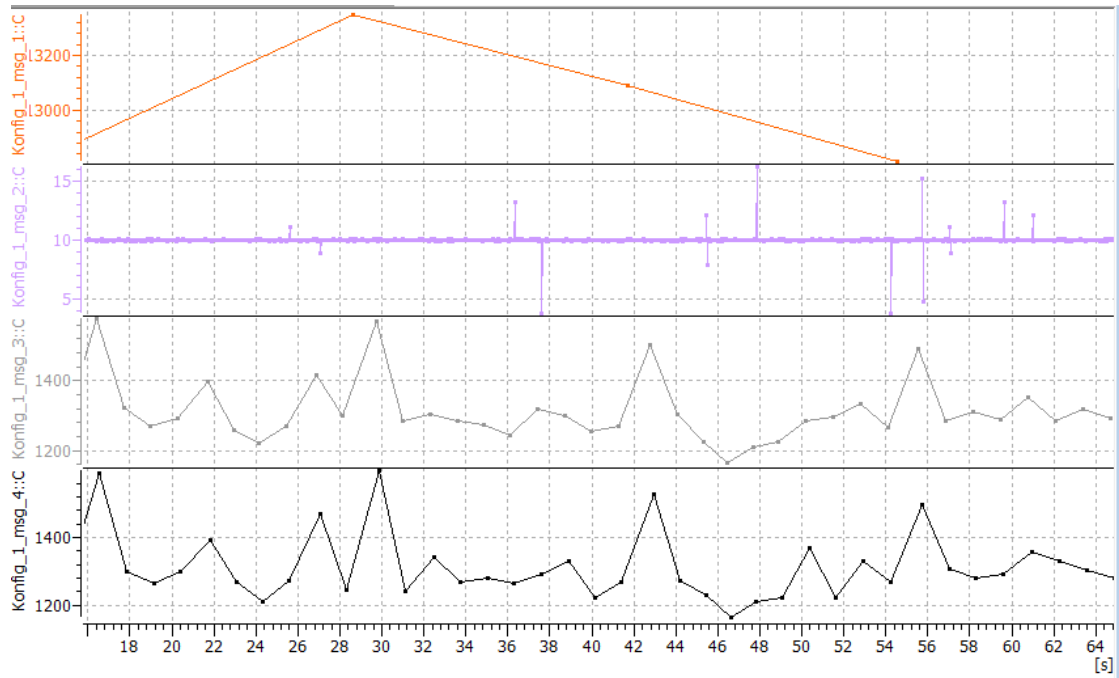
Obr. 84 – Test periody odesílání zpráv. Periody odesílání nastaveny na 100, 200, 300 a 400 ms.

Převodník je naprogramován tak, že před odesláním zprávy testuje, zda je možné vůbec zprávu na sběrnici odeslat (Obr. 78). Pokud to není zrovna v danou chvíli možné (např. při naplnění odesílacího bufferu), bude se o odeslání pokoušet znovu, dokud není možné zprávu odeslat, nebo do vypršení jedné milisekundy. V tomto případě by se zpráva měla v nejhorším případě zaručeně odeslat s o jednu milisekundu delší periodou.

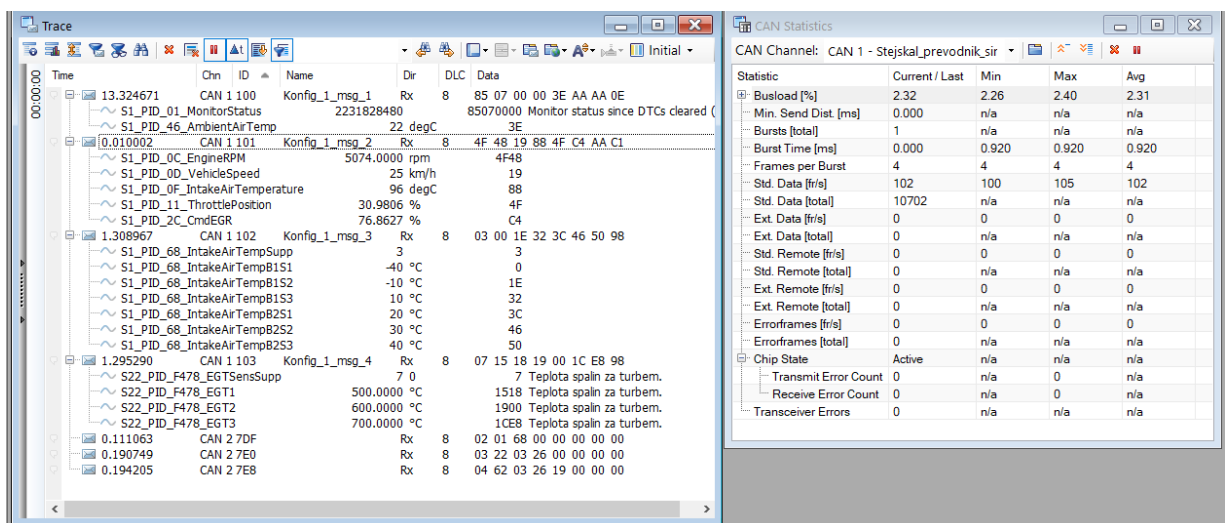
Nicméně v některých případech, naměřených na Obr. 83, se zpráva odešla s delší než danou periodou. Nebyla vypátrána příčina této anomálie, ale nejspíš k tomu přispívá to, že se odesílání CAN zpráv odehrává na druhém jádře mikrokontroleru. Protože v případě odesílání zpráv na prvním jádře k takové fluktuaci nedochází (v případě, kdy mikrokontroler nezajišťuje nic jiného, než pouhé odesílání zpráv).

Při druhém testování periody k této fluktuaci docházelo pouze v případě zprávy, která byla odesílána s periodou 100 ms. Na zprávách s vyššími periodami se tato fluktuace prakticky nevyskytuje.

Následně bylo ozkoušeno odesílání periodických a neperiodických zpráv dohromady. Neperiodické zprávy měly v konfiguračním souboru nastaveny rozdílný parametr „Jak často PID vyčítat“. Na Obr. 85 jsou zobrazeny periody přijatých zpráv (v ms) v závislosti na čase (v sekundách). Je zde vidět, že periodická zpráva byla odesílána s nastavenou periodou (10 ms). U neperiodických zpráv je zaznamenaná perioda odesílání různá v čase a je závislá na tom, jak často se mají k ní přiřazené signály vyčítat z vozidla a na tom, kolik signálů je k jedné zprávě přiřazeno. Provoz převodníku při odesílání periodických i neperiodických signálů najednou je tedy funkční. Na Obr. 86 obsah těchto zpráv přijatých analyzátořem.



Obr. 85 – Příjem neperiodických a periodických zpráv. Grafické vyjádření.



Obr. 86 – Příjem neperiodických a periodických zpráv. Vyobrazení obsahu přijatých zpráv.

Byla otestována i správná funkce odesílání CAN zpráv se standartním a rozšířeným rámcem. Na Obr. 87 je vidět seznam přijatých zpráv. Každá zpráva má svoje ID, zprávy s rozšířeným rámcem mají za svým ID znak „x“. Jedná se tedy o zprávy s ID 0x101 a 0x1FFFFFFF.

Time	Chn	ID	Name	Dir	DLC	Data
00:04:38		1.173975	CAN 1 101x	Rx	8	52 38 0B 7A 28 9E AA 4D
		1.218967	CAN 1 102	Rx	8	03 00 1E 32 3C 46 50 4D
			S1_PID_68_IntakeAirTempSupp		3	3
			S1_PID_68_IntakeAirTempB1S1		0	-40 °C
			S1_PID_68_IntakeAirTempB1S2		1E	-10 °C
			S1_PID_68_IntakeAirTempB1S3		32	10 °C
			S1_PID_68_IntakeAirTempB2S1		3C	20 °C
			S1_PID_68_IntakeAirTempB2S2		46	30 °C
			S1_PID_68_IntakeAirTempB2S3		50	40 °C
		1.218973	CAN 1 103	Rx	8	07 15 18 19 00 1C E8 4D
			S22_PID_F478_EGTSensSupp		7 0	7 Teplota spalin za turbem.
			S22_PID_F478_EGT1		1518	500.0000 °C Teplota spalin za turbem.
			S22_PID_F478_EGT2		1900	600.0000 °C Teplota spalin za turbem.
			S22_PID_F478_EGT3		1CE8	700.0000 °C Teplota spalin za turbem.
		0.161199	CAN 2 7DF	Rx	8	02 01 11 00 00 00 00 00
		0.004600	CAN 2 7E0	Rx	8	30 00 00 00 00 00 00 00
		0.136435	CAN 2 7E8	Rx	8	03 41 11 2B 00 00 00 00
		12.651679	CAN 1 1FFFFFFFx	Rx	8	85 07 00 00 3E AA AA 06

Obr. 87 – Příjem zpráv se standardním a rozšířeným rámcem.

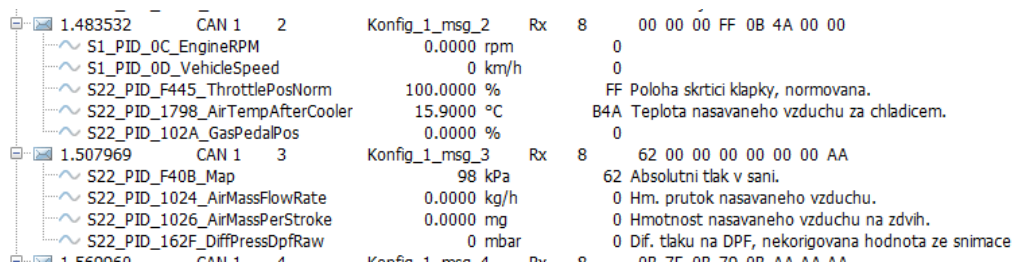
Nakonec bylo provedeno testování, při kterém bylo žádáno o takové SIDy a PIDy, které jsou podporovány testovacím vozidlem (testování na testovacím vozidle je zdokumentováno v podkapitole 7.2). Detaily na přijatá data vysílaná převodníkem jsou k nahlédnutí v Příloha 2. 2 a Příloha 2. 3.

7.2 Vozidlo

Po ukončení testování na emulátoru vozidla, se přešlo na testování převodníku na vozidle Škoda Octavia III 2.0 TDI 110 kW (kód motoru DCYA). Testování se provedlo při vypnutém motoru, při volnoběhu a poté se provedla testovací jízda. Všechna tato testování proběhla s použitím jednoho konfiguračního souboru. Obsah tohoto konfiguračního souboru s důležitými parametry se nachází v Příloha 3.

Hodnoty, které poskytuje emulátor vozidla, lze definovat v programu. Toto ale u vozidla nejde. Tudíž pro ověření správnosti vyčítání, zpracovávání a odesílání dat, je použit software OBDwiz, který lze připojit ke stejnému OBD adaptéru, který využívá převodník. Data přijatá z převodníku a data vyčtená z OBDwiz lze tedy jednoduše porovnat. Nevýhoda tohoto softwaru je ovšem taková, že neumí vyčítat jiné PIDy než ty, které jsou definované standardem EOBD. Tudíž lze porovnat pouze hodnoty některých PIDů. Naměřená data ze SW Vector CANalyzer a OBDwiz se nachází v Příloha 3, dále budou představeny pouze vybrané části.

Vozidlo stojí, motor vypnut

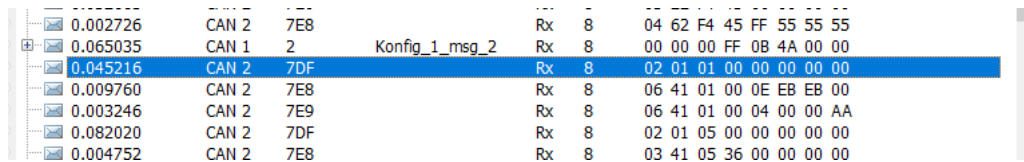


Obr. 88 – Vozidlo stojí, motor vypnut. Data přijatá z převodníku. Detail na otáčky motoru, rychlost vozidla a absolutní tlak v sání.

SAE	Protocol	Parameter	Value	Unit
SAE 0x0B	OBD2	Intake manifold absolute pressure	98	kPa
SAE 0x0C	OBD2	Engine RPM	0	RPM
SAE 0x0D	OBD2	Vehicle speed	0	km/h
SAE 0x0F	OBD2	Intake air temperature	15.9	°C

Obr. 89 – Vozidlo stojí, motor vypnut. Data vyčtena pomocí SW OBDWiz. Detail na otáčky motoru, rychlost vozidla a absolutní tlak v sání.

Nejprve bylo provedeno měření na stojícím vozidle, při vypnutém motoru. Na Obr. 88 a Obr. 89 se nachází detail na hodnoty, které byly přijaty z převodníku a hodnoty, které byly vyčteny pomocí OBDwiz. V obou případech se shodují informace o otáčkách motoru, rychlosti vozidla a absolutním tlaku v sání.



Obr. 90 – Dvě odpovědi (motor 0x7E8 a automatická převodovka 0x7E9), různý obsah na stejnou žádost.

Může se stát to, že při žádosti o data (při použití funkčního adresování) odpoví více uzlů najednou, což byl případ i při tomto testování (Obr. 90). Žel informace od obou uzlů se v některých případech můžou lišit. Např. při žádosti o otáčky motoru může odpovědět řídicí jednotka motoru a následně řídicí jednotka automatické převodovky. Ovšem otáčky v druhém případě už mohou být nižší, nebo naopak vyšší. Toto se může i např. při žádosti o readiness kód (SID 0x01 PID 0x01), jehož obsah se může u těchto řídicích jednotek lišit.

Program převodníku je navržen tak, že zpracovává první přijatou odpověď. Na obrázku Obr. 91 je vidět opakovaná žádost o otáčky motoru. Někdy odpoví dříve jeden uzel, někdy ten druhý. Což v některých případech vadit nemusí, v některých to může naprosto změnit hledaný význam dat, jak je vidět na odpovědích (CAN ID 0x7E8 a 0x7E9) na Obr. 90.

```

17:05:11.596 >17:05:12.186 01 0c
17:05:12.215 7E8 04 41 0C 00 00
17:05:12.215 7E9 04 41 0C 00 00
17:05:12.240
17:05:12.240 >17:05:12.738 01 0c
17:05:12.768 7E8 04 41 0C 00 00
17:05:12.768 7E9 04 41 0C 00 00
17:05:12.819
17:05:12.819 >17:05:13.304 01 0c
17:05:13.341 7E9 04 41 0C 00 00
17:05:13.341 7E8 04 41 0C 00 00
17:05:13.364
17:05:13.364 >17:05:13.739 01 0c
17:05:13.775 7E8 04 41 0C 00 00
17:05:13.775 7E9 04 41 0C 00 00

```

Obr. 91 – Opakovaná žádost o otáčky motoru. Více odpovědí z vozidla.

Toto by se dalo vyřešit tím způsobem, že by se v konfiguračním souboru definoval nový parametr signálu. Tento parametr by umožnil zadat očekávané ID odpovědi z vozidla. V případě více odpovědí by tedy zpracoval data z odpovědi se zadaným ID. Pokud by obsluha nevěděla dopředu ID takové odpovědi, toto pole by se nechalo prázdné a převodník by zpracovával první přijatou odpověď.

Vozidlo stojí, volnoběh

Obr. 92 – Vozidlo stojí, volnoběh. Data přijatá z převodníku. Detail na otáčky motoru a rychlost vozidla.

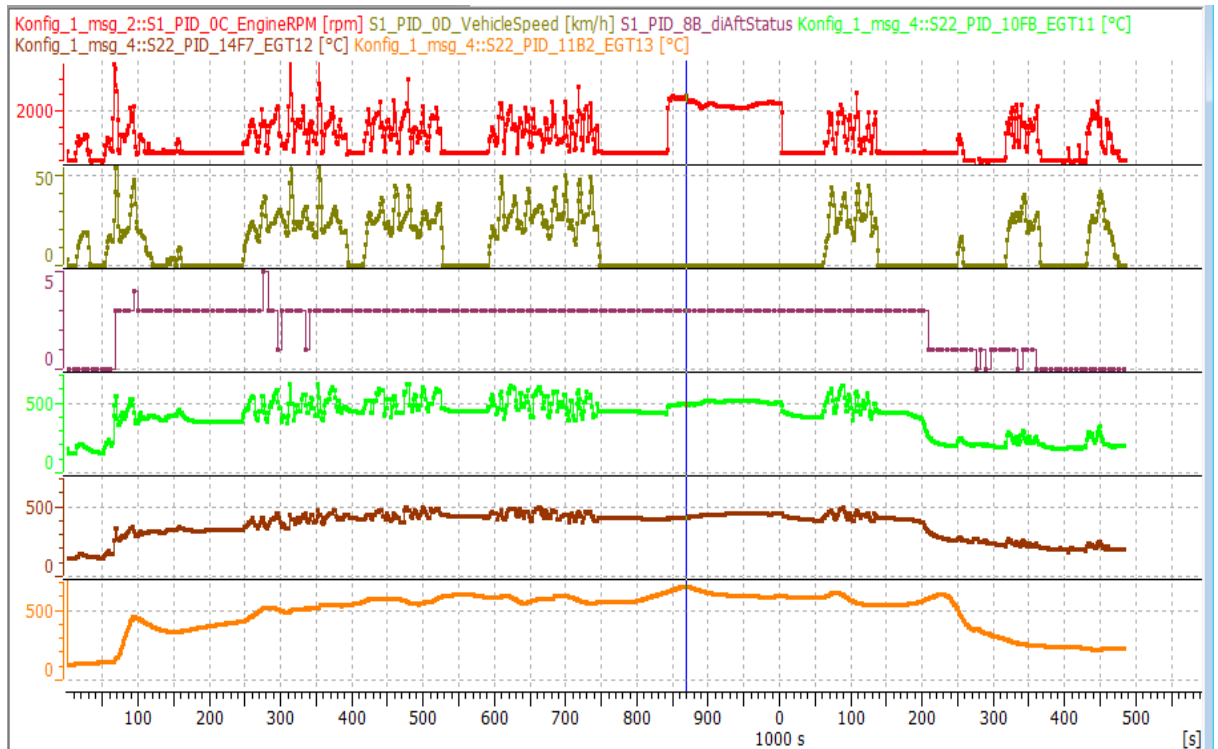
SAE Ux0B	Ux0D	Engine coolant temperature	19 °C
SAE 0x0B	OBD2	Intake manifold absolute pressure	98 kPa
SAE 0x0C	OBD2	Engine RPM	770 RPM
SAE 0x0D	OBD2	Vehicle speed	0 km/h
SAE 0x0F	OBD2	Intake air temperature	19 °C
SAE 0x10	OBD2	Mass air flow rate	9.88 g/s
SAE 0x11	OBD2	Absolute throttle position	83.53 %

Obr. 93 – Vozidlo stojí, volnoběh. Data vyčtena pomocí SW OBDWiz. Detail na otáčky motoru a rychlost vozidla.

Vyčtená data odpovídají situaci, kdy vozidlo stojí a motor má volnoběžné otáčky. Volnoběžné otáčky se nerovnají, protože byly snímky pořízeny v jinou dobu. Ovšem odpovídají otáčkám volnoběžným.

Testovací jízda

Jako poslední test byla zvolena jízda s vozidlem. Data z tohoto měření byla vynesena do grafů na Obr. 94.



Obr. 94 – Naměřené hodnoty dat při testovací jízdě vozidla.

První signál odshora reprezentuje otáčky motoru v ot/min. Druhý odshora je poté rychlost vozidla v km/h. Dalším signálem je tzv. Diesel aftertreatment status. Jedná se o bitově kódovanou informaci o stavech filtru pevných částic a katalyzátoru DeNOx. Význam jednotlivých bitů tohoto nibblu se nachází v Tab. 4. Čtvrtým signálem odshora je poté teplota spalin před turbodmychadlem. Následují opět teploty, nejdříve za turbodmychadlem, poté před filtrem pevných částic. Teploty jsou zobrazeny ve stupních Celsia. Všechny signály jsou vyneseny v závislosti na čase a pocházejí z CAN zpráv, jejichž obsah byl naplněn převodníkem daty vyčtenými z OBD zásuvky vozidla.

Tab. 4 – Význam jednotlivých bitů v nibblu DieselAftertreatmentStatus.

Číslo bitu	Význam	Stav při hodnotě 0	Stav při hodnotě 1
0	Status regenerace DPF	Neregeneruje se	Regeneruje se
1	Typ regenerace DPF	Pasivní regenerace	Aktivní regenerace
2	Status adsorpce NOx	Probíhá adsorpce	Probíhá regenerace
3	Status desulfurizace DeNOx	Nedochází k desulfurizaci	Dochází k desulfurizaci

Při měření se podařilo zachytit moment, kdy došlo ke spuštění regenerace filtru pevných částic. Tedy na začátku testování byl nibble DieselAftertreatmentStatus roven nule. Poté se jeho hodnota změnila na 3 dekadicky, což bitově odpovídá 0011. Pomocí Tab. 4 lze zjistit, že dochází k aktivní regeneraci filtru pevných částic. Poté se jeho hodnota po většinu měření skoro nemění. Po tuto dobu je vidět nárůst teploty spalin, způsobený aktivní regenerací filtru. V místě obrázku, ve kterém je umístěn vertikální kurzor, je okamžik nulové rychlosti vozidla a zvýšených otáček motoru. V tomto momentu bylo dopomáháno vozidlu k urychlení regenerace filtru. Ke konci měření se hodnota tohoto nibblu rovná dekadicky jedné, občas skokově změnila hodnotu na nulu. Což naznačuje konec aktivní regenerace, teplota spalin začíná klesat, a regeneruje se pouze pasivně. Na konci měření dojde k ukončení regenerace filtru pevných částic.

8 Souhrnné vlastnosti převodníku

Hodnota napájecího napětí převodníku je 5V. Lze ho napájet pomocí kabelu USB (typu C), nebo napájecím modulem BASE 15 (hardware převodníku i zmíněný modul dodává výrobce M5Stack). Převodník lze napájet i jiným interoperabilním napájecím modulem.

Podporuje všechny OBD adaptéry, které jsou kompatibilní s čipem ELM327. Převodník si do nevolatilní paměti ukládá adresy a názvy alespoň jednou úspěšně připojených OBD adaptérů. Lze se tedy připojit jak k libovolnému adaptéru z okolí, tak k dostupnému adaptéru, který je uložen v paměti převodníku.

Výsledný program převodníku dokáže vyčítat veškeré služby a dané parametry standardu EOBD z OBD zásuvky vozidla. Není omezen ani ve vyčítání nad rámec EOBD (limitem je pouze znalost významu dat daných služeb a parametrů, jelikož si je každý výrobce definuje sám). Podporuje vyčítání zpráv jak se standardním 11-bit ID (rámec CAN2.0A), tak s 29-bit ID (rámec CAN2.0B) a umožňuje komunikovat na komunikačních rychlostech 250 a 500 kBaud (je vyžadováno vozidlo, které komunikuje na OBD zásuvce přes CAN bus). Umožňuje vyčítání velkého množství signálů (prakticky omezeno alokovatelnou pamětí mikrokontroleru).

Pro odesílání CAN zpráv na sběrnici CAN bez protokolu využívá převodník řadič CAN s označením MCP2515 a budič CAN s označením TJA1051 (CAN high-speed). Je možné odesílat zprávy s libovolnými ID jak standardního 11-bit rámce, tak rozšířeného 29-bit rámce. Zprávy se mohou odesílat až v případě aktualizace všech signálů dané zprávy, či s libovolnou periodou. Komunikační rychlost na CAN bus bez protokolu je nastavena na 500 kBaud. Vodiče sběrnice CAN se k převodníku připojují pomocí konektoru typu GROVE (Obr. 95).



Obr. 95 – Konektor USB typ C (napájení) a konektor typu GROVE (připojení sběrnice CAN bus).

V menu převodníku lze jednoduše nastavit jas podsvícení LCD a čas jeho zhasnutí v případě nečinnosti zařízení.

Pro uvedení převodníku v činnost je potřeba

- umístit OBD adaptér do OBD zásuvky vozidla,
- zapnout převodník a vložit SD kartu s konfiguračním souborem,
- připojit se k OBD adaptéru pomocí BT (buď z okolí, nebo výběrem z paměti MCU) a
- zvolit položku „Odesilani CAN zprav“ z menu „Vycitani dat“.

9 Závěr

Zadáním práce bylo vytvořit zařízení, které umožní přeposílání diagnostických dat vyčítaných ze zásuvky OBD vozidla po sběrnici CAN bus, pomocí protokolu UDS. Dále měla být tato data odeslána na další sběrnici CAN bus, ovšem bez protokolu. Zařízení by tedy mělo odesílat pouze zprávy s definovaným CAN ID.

V průběhu práce bylo potřeba seznámit se se sběrnici CAN bus, se systémem EOBD a s různými protokoly na různých vrstvách ISO/OSI modelu (převážně se standardem UDS).

Výsledkem této práce je funkční program pro převodník, který je postavený na mikrokontroleru ESP32. Data z vozidla jsou vyčítána OBD adaptérem, který s převodníkem komunikuje pomocí Bluetooth. Převodník data z vozidla vyčítá a dále je ukládá na předem určené pozice do CAN zprávy. Zprávy následně odesílá na sběrnici CAN bus bez protokolu. Navíc byla do hloubky rozebrána otázka napájení převodníku. Bylo navrženo a otestováno nejvhodnější řešení, tedy použití napájecího modulu (viz podkapitola 4.4.3).

Převodník byl dostatečně otestován na emulátoru vozidla a poté na testovacím vozidle. V průběhu testování byly odhaleny nějaké nedostatky programu a hardwaru převodníku, které ale kriticky neohrožují jeho zamýšlenou funkci. Např. že v konfiguračním souboru chybí možnost volby komunikační rychlosti na OBD zásuvce a na sběrnici CAN bus bez protokolu (podkapitola 5.2). Také že je perioda odesílání periodických CAN zpráv, které mají nízkou periodu odesílání (v řádech desítek milisekund), nestabilní (podkapitola 7.1). Nebo že v takovém případě, ve kterém může na žádost o data odpovědět více jednotek, převodník zpracovává pouze první přijatou odpověď (podkapitola 7.2).

Bylo navrženo řešení pro možnost nastavení komunikační rychlosti na zásuvce OBD a sběrnici CAN bus bez protokolu pomocí konfiguračního souboru (podkapitola 5.2) a pro situaci, ve které odpovídá více řídicích jednotek najednou (podkapitola 7.2).

Zadání práce by tedy mělo být splněno v plném rozsahu.

Literatura

- [1] ISO. ISO 14229-1, Road vehicles — Unified diagnostic services (UDS): Specification and requirements. 2nd edition. 2013.
- [2] BRITISH STANDARDS [BSI]. ISO 15031-5, Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics: Part 5: Emissions-related diagnostic services. 2006.
- [3] M5STACK. CORE BASIC. Online. ©2021-2024. Dostupné z: <https://docs.m5stack.com/en/core/basic>. [cit. 2024-04-11].
- [4] VGATE TECHNOLOGY CO., LTD. Vgate vLinker MC+. Online. © 2022. Dostupné z: <https://www.vgatemall.com/products-detail/i-5/>. [cit. 2024-04-11].
- [5] M5STACK. Module COMMU. Online. ©2021-2024. Dostupné z: <https://docs.m5stack.com/en/module/commu>. [cit. 2024-04-11].
- [6] ESP32 Series Datasheet. Online. Ver. 4.5. Espressif Systems, © 2024. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. [cit. 2024-04-12].
- [7] ESP32. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-, 2. 3. 2024. Dostupné z: <https://cs.wikipedia.org/wiki/ESP32>. [cit. 2024-04-12].
- [8] Dhrystone. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-, 5. 8. 2021. Dostupné z: <https://cs.wikipedia.org/wiki/Dhrystone>. [cit. 2024-04-12].
- [9] CAN bus. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-, 21. 4. 2024. Dostupné z: https://cs.wikipedia.org/wiki/CAN_bus. [cit. 2024-04-28].
- [10] M5 STACK CORE. Online. M5Stack, 2017. Dostupné z: [https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/docs/schematic/Core/M5-Core-Schematic\(20171206\).pdf](https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/docs/schematic/Core/M5-Core-Schematic(20171206).pdf). [cit. 2024-04-22].
- [11] A-Si TFT LCD Single Chip Driver 320RGBx240 Resolution and 262K color. Online. V101. ILI TECHNOLOGY, 2011. Dostupné z: <https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/docs/datasheet/core/ILI9342C-ILITEK.pdf>. [cit. 2024-04-22].
- [12] RUILON (Shenzhen Ruilongyuan Elec) RLSD52A031V. Online. Version: A0/2022-07-06. RUILON, 2022. Dostupné z: https://www.lcsc.com/datasheet/lcsc_datasheet_2309271415_RUILON-Shenzhen-Ruilongyuan-Elec-RLSD52A031V_C20931.pdf. [cit. 2024-04-23].
- [13] Online. BASIC v2.7. ©2021-2024, aktualizace 4/2023. Dostupné z: https://docs.m5stack.com/en/core/basic_v2.7. [cit. 2024-04-23].

- [14] Online. Base 15. ©2021-2024. Dostupné z: <https://docs.m5stack.com/en/module/Base%2015>. [cit. 2024-04-25].
- [15] MP1584 3A, 1.5MHz, 28V Step-Down Converter. Online. Rev. 1.0. MPS, 2011. Dostupné z: https://www.monolithicpower.com/en/documentview/productdocument/index/version/2/document_type/Datasheet/lang/en/sku/MP1584EN-LF-Z/document_id/204/. [cit. 2024-04-25].
- [16] MP2315 High Efficiency 3A, 24V, 500kHz Synchronous Step Down Converter. Online. Rev. 1.01. MPS, 2014. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1035056/MPS/MP2315.html>. [cit. 2024-04-25].
- [17] Online. Module PROTO. ©2021-2024. Dostupné z: <https://docs.m5stack.com/en/module/proto>. [cit. 2024-04-25].
- [18] Online. USB to High Speed Serial Port Chip CH9102. © 2002-2024. Dostupné z: <https://www.wch-ic.com/products/CH9102.html>. [cit. 2024-04-26].
- [19] Fully-Integrated Power Bank System-On-Chip with 2.1A charger, 2.4A discharger. Online. V1.31. Injoinic technology, © 2016. Dostupné z: <http://www.injoinic.com/wwwroot/uploads/files/20200221/0405f23c247a34d3990ae100c8b20a27.pdf>. [cit. 2024-04-26].
- [20] 3CH Power Management IC EA3036 Datasheet. Online. Ver. 1.1. EveranalogIntegrated Circuit Limited, ©2015. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1221868/ETC1/EA3036.html>. [cit. 2024-04-26].
- [21] Referenční model ISO/OSI. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-, 4. 8. 2023. Dostupné z: https://cs.wikipedia.org/wiki/Referen%C4%8Dn%C3%AD_model_ISO/OSI. [cit. 2024-04-28].
- [22] Komunikační sběrnice CAN. Power Point prezentace. Ing. Zdeněk Mašek, Ph.D., b.r.
- [23] Diagnostika řídicích jednotek automobilu. Power Point prezentace. Ing. Zdeněk Mašek, Ph.D., b.r.
- [24] OBD. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-, 26. 1. 2024. Dostupné z: <https://cs.wikipedia.org/wiki/OBD>. [cit. 2024-04-28].
- [25] UDS Explained – A Simple Intro (Unified Diagnostic Services). Online. CSS ELECTRONICS. © 2024. Dostupné z: <https://www.csselectronics.com/pages/uds-protocol-tutorial-unified-diagnostic-services>. [cit. 2024-04-28].
- [26] ELM327 OBD to RS232 Interpreter. Online. Elm Electronics, ©2005-2010. Dostupné z: <https://www.elmelectronics.com/DSheets/ELM327DSH.pdf>. [cit. 2024-04-29].
- [27] VOSS, Wilfried. A comprehensible guide to controller area network. Greenfield: Cooperhill Technologies Corporation, [2005]., 150 s. ISBN 0976511606.

Seznam příloh

Příloha 1, Testování konfiguračního souboru.....	86
Příloha 2, Testování převodníku – emulátor vozidla.....	95
Příloha 3, Testování převodníku – vozidlo	97

Příloha 1, Testování konfiguračního souboru

Příloha 1. 1 – Zdvojené signály. Tabulka s důležitými parametry konfiguračního souboru.

Pořadí signálů v souboru	1.	2.	3.	4.
Jak často PID vyčítat	10	10	10	10
PID	0x01	0x01	0x01	0x01
SID	0x01	0x01	0x01	0x01
Header requestu	0x7DF	0x7DF	0x7DF	0x7DF
Číslo bajtu v odpovědi	0	0	0	0
Délka signálu [bajtů]	4	4	4	4
CAN ID zprávy	0x001	0x001	0x001	0x002
Extended	0	0	0	0
Perioda odesílání [ms]	0	0	0	0
Poloha ve zprávě [od čísla bajtu]	0	0	0	0

```

----Struktura[3]-----
Jak často PID vycitat: 10
PID: 0x1
Delka PID: 1 bajtu
SID: 0x1
Header requestu: 0x7DF
Cislo bajtu v odpovedi: 0
Delka vycitaneho signalu: 4 bajt/u
Poloha signalu ve zpave CAN: 0
Adresa do jake zpravy CAN patri: 0x3FFEE760
Nachazi se dalsi pro zpracovani, na adrese: 0x3FFED5F0
Hodnota signalu: 0xAA 0xAA 0xAA 0xAA

```

```

----Struktura[4]-----
Jak často PID vycitat: 10
PID: 0x1
Delka PID: 1 bajtu
SID: 0x1
Header requestu: 0x7DF
Cislo bajtu v odpovedi: 0
Delka vycitaneho signalu: 4 bajt/u
Poloha signalu ve zpave CAN: 0
Adresa do jake zpravy CAN patri: 0x3FFEE77C
Nenachazi se dalsi pro zpracovani: NULL
Hodnota signalu: 0xAA 0xAA 0xAA 0xAA

```

```

----CAN[1]-----
CAN ID: 0x1
Extended: 0
Data: AA AA AA AA AA AA AA AA
Perioda odesilani: 0
Pocet signalu, ktere k tomuto ID patri: 3

```

```

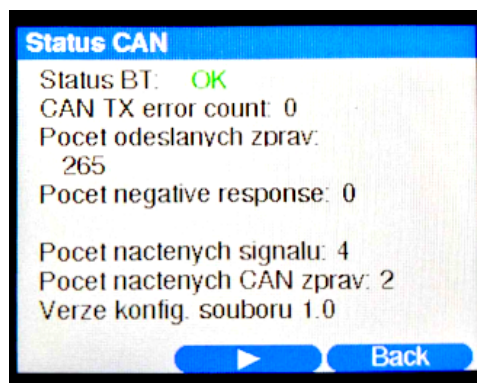
----CAN[2]-----
CAN ID: 0x2
Extended: 0
Data: AA AA AA AA AA AA AA AA
Perioda odesilani: 0
Pocet signalu, ktere k tomuto ID patri: 1

```

Příloha 1. 2 – Zdvojené signály. Detail na výpis sériového monitoru. Vlevo načtené signály, vpravo načtené zprávy CAN.

Nu...	Receive time	Info	Type	Id 1	Id 2	Bytes	B0	B1	B2	B3	B4	B5	B6	B7	ASCII
47	18:8:44.842:0.00		St	0x2		8	83	07	4A	A1	AA	AA	AA	AA	1 J
46	18:8:44.841:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	AA	1 J
45	18:8:44.308:0.00		St	0x2		8	83	07	4A	A1	AA	AA	AA	AA	1 J
44	18:8:44.308:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	AA	1 J
43	18:8:43.783:0.00		St	0x2		8	83	07	4A	A1	AA	AA	AA	AA	1 J
42	18:8:43.783:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	AA	1 J
41	18:8:43.223:0.00		St	0x2		8	83	07	4A	A1	AA	AA	AA	AA	1 J
40	18:8:43.223:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	AA	1 J
39	18:8:42.701:0.00		St	0x2		8	83	07	4A	A1	AA	AA	AA	AA	1 J
38	18:8:42.701:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	AA	1 J
37	18:8:42.165:0.00		St	0x2		8	83	07	4A	A1	AA	AA	AA	AA	1 J
36	18:8:42.165:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	AA	1 J
35	18:8:41.636:0.00		St	0x2		8	83	07	4A	A1	AA	AA	AA	AA	1 J
34	18:8:41.636:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	AA	1 J
33	18:8:41.101:0.00		St	0x2		8	83	07	4A	A1	AA	AA	AA	AA	1 J

Příloha 1. 3 – Zdvojené signály. Zprávy odesílané na sběrnici CAN bez protokolu. Software PP2CAN.



Příloha 1. 4 – Zdvojené signály. Detail na displej převodníku.

Příloha 1. 5 – Rozdílné parametry u jedné zprávy CAN. Tabulka s důležitými parametry konfiguračního souboru.

Pořadí signálů v souboru	1.	2.
Jak často PID vyčítat	10	1
PID	0x01	0x1798
SID	0x01	0x022
Header requestu	0x7DF	0x7E0
Číslo bajtu v odpovědi	0	0
Délka signálu [bajtů]	4	2
CAN ID zprávy	0x001	0x001
Extended	0	1
Perioda odesílání [ms]	500	0
Poloha ve zprávě [od čísla bajtu]	0	4

```

-----Struktury Signal:
----Struktura[1]-----
Jak casto PID vycitat: 10
PID: 0x1
Delka PID: 1 bajtu
SID: 0x1
Header requestu: 0x7DF
Cislo bajtu v odpovedi: 0
Delka vycitaneho signalu: 4 bajt/u
Poloha signalu ve zprave CAN: 0
Adresa do jake zpravy CAN patri: 0x3FFCBE84
Nenachazi se dalsi pro zpracovani: NULL
Hodnota signalu: 0xAA 0xAA 0xAA 0xAA

----Struktura[2]-----
Jak casto PID vycitat: 1
PID: 0x1798
Delka PID: 2 bajtu
SID: 0x22
Header requestu: 0x7E0
Cislo bajtu v odpovedi: 0
Delka vycitaneho signalu: 2 bajt/u
Poloha signalu ve zprave CAN: 4
Adresa do jake zpravy CAN patri: 0x3FFCBE84
Nenachazi se dalsi pro zpracovani: NULL
Hodnota signalu: 0xAA 0xAA

-----Struktury CAN:
----CAN[1]-----
CAN ID: 0x1
Extended: 0
Data: AA AA AA AA AA AA AA
Perioda odesilani: 500
Pocet signalu, ktere k tomuto ID patri: 2

```

Příloha 1. 6 – Rozdílné parametry u jedné zprávy CAN. Detail na výpis sériového monitoru. Nahore načtené signály, dole načtená zpráva CAN.

Nu...	Receive time	Info	Type	Id 1	Id 2	Bytes	B0	B1	B2	B3	B4	B5	B6	B7	ASCII
44	18:47:25.158:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
43	18:47:24.658:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
42	18:47:24.158:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
41	18:47:23.660:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
40	18:47:23.157:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
39	18:47:22.658:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
38	18:47:22.160:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
37	18:47:21.658:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
36	18:47:21.163:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
35	18:46:51.620:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
34	18:46:51.120:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
33	18:46:50.622:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
32	18:46:50.120:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
31	18:46:49.621:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
30	18:46:49.120:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
29	18:46:48.619:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
28	18:46:48.120:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
27	18:46:47.622:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
26	18:46:47.120:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
25	18:46:46.620:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
24	18:46:46.119:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
23	18:46:45.619:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
22	18:46:45.120:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
21	18:46:44.620:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
20	18:46:44.122:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G
19	18:46:43.620:0.00		St	0x1		8	83	07	4A	A1	0A	47	AA	AA	1 J G

Příloha 1. 7 – Rozdílné parametry u jedné zprávy CAN. Zprávy odesílané na sběrnici CAN bez protokolu. Software PP2CAN.

Příloha 1. 8 – Kolize dvou signálů v jedné zprávě CAN. Tabulka s důležitými parametry konfiguračního souboru.

Pořadí signálů v souboru	1.	2.	3.	4.
Jak často PID vyčítat	1	1	1	1
PID	0x01	0x1798	0x01	0x1798
SID	0x01	0x22	0x01	0x22
Header requestu	0x7DF	0x7E0	0x7DF	0x7E0
Číslo bajtu v odpovědi	0	0	0	0
Délka signálu [bajtů]	4	2	4	2
CAN ID zprávy	0x001	0x001	0x002	0x003
Extended	0	0	0	0
Perioda odesílání [ms]	0	0	0	0
Poloha ve zprávě [od čísla bajtu]	0	0	0	0

```

-----Struktury signal:
---Struktura[1]-----
Jak často PID vycitat: 1
PID: 0x1
Delka PID: 1 bajtu
SID: 0x1
Header requestu: 0x7DF
Cislo bajtu v odpovedi: 0
Delka vycitaneho signalu: 4 bajt/u
Poloha signalu ve zpave CAN: 0
Adresa do jake zpravy CAN patri: 0x3FFCBED4
Nachazi se dalsi pro zpracovani, na adrese: 0x3FFCBE74
Hodnota signalu: 0xAA 0xAA 0xAA 0xAA

---Struktura[2]-----
Jak často PID vycitat: 1
PID: 0x1798
Delka PID: 2 bajtu
SID: 0x22
Header requestu: 0x7E0
Cislo bajtu v odpovedi: 0
Delka vycitaneho signalu: 2 bajt/u
Poloha signalu ve zpave CAN: 0
Adresa do jake zpravy CAN patri: 0x3FFCBED4
Nachazi se dalsi pro zpracovani, na adrese: 0x3FFCBE9C
Hodnota signalu: 0xAA 0xAA

---Struktura[3]-----
Jak často PID vycitat: 1
PID: 0x1
Delka PID: 1 bajtu
SID: 0x1
Header requestu: 0x7DF
Cislo bajtu v odpovedi: 0
Delka vycitaneho signalu: 4 bajt/u
Poloha signalu ve zpave CAN: 0
Adresa do jake zpravy CAN patri: 0x3FFCBF0
Nenachazi se dalsi pro zpracovani: NULL
Hodnota signalu: 0xAA 0xAA 0xAA 0xAA

---Struktura[4]-----
Jak často PID vycitat: 1
PID: 0x1798
Delka PID: 2 bajtu
SID: 0x22
Header requestu: 0x7E0
Cislo bajtu v odpovedi: 0
Delka vycitaneho signalu: 2 bajt/u
Poloha signalu ve zpave CAN: 0
Adresa do jake zpravy CAN patri: 0x3FFCBF0C
Nenachazi se dalsi pro zpracovani: NULL
Hodnota signalu: 0xAA 0xAA

```

Příloha 1. 9 – Kolize dvou signálů v jedné zprávě CAN. Detail na výpis sériového monitoru. Načtené signály.

```

-----Struktury CAN:
----CAN[1]-----
CAN ID: 0x1
Extended: 0
Data: AA AA AA AA AA AA AA AA
Perioda odesilani: 0
Pocet signalu, ktere k tomuto ID patri: 2

----CAN[2]-----
CAN ID: 0x2
Extended: 0
Data: AA AA AA AA AA AA AA AA
Perioda odesilani: 0
Pocet signalu, ktere k tomuto ID patri: 1

----CAN[3]-----
CAN ID: 0x3
Extended: 0
Data: AA AA AA AA AA AA AA AA
Perioda odesilani: 0
Pocet signalu, ktere k tomuto ID patri: 1

```

Příloha 1. 10 – Kolize dvou signálů v jedné zprávě CAN. Detail na výpis sériového monitoru. Načtené zprávy CAN.

Nu...	Receive time	Info	Type	Id 1	Id 2	Bytes	B0	B1	B2	B3	B4	B5	B6	B7	ASCII
81	19:39:21.317:0.00		St	0x2		8	83	07	4A	A1	AA	AA	AA	AA	1 J
80	19:39:21.155:0.00		St	0x3		8	0A	47	AA	AA	AA	AA	AA	AA	1 G
79	19:39:21.155:0.00		St	0x1		8	0A	47	4A	A1	AA	AA	AA	AA	1 GJ
78	19:39:21.42:0.00		St	0x2		8	83	07	4A	A1	AA	AA	AA	AA	1 J
77	19:39:20.915:0.00		St	0x3		8	0A	47	AA	AA	AA	AA	AA	AA	1 G
76	19:39:20.915:0.00		St	0x1		8	0A	47	4A	A1	AA	AA	AA	AA	1 GJ
75	19:39:20.774:0.00		St	0x2		8	83	07	4A	A1	AA	AA	AA	AA	1 J
74	19:39:20.643:0.00		St	0x3		8	0A	47	AA	AA	AA	AA	AA	AA	1 G
73	19:39:20.642:0.00		St	0x1		8	0A	47	4A	A1	AA	AA	AA	AA	1 GJ
72	19:39:20.520:0.00		St	0x2		8	83	07	4A	A1	AA	AA	AA	AA	1 J
71	19:39:20.363:0.00		St	0x3		8	0A	47	AA	AA	AA	AA	AA	AA	1 G
70	19:39:20.362:0.00		St	0x1		8	0A	47	4A	A1	AA	AA	AA	AA	1 GJ
69	19:39:20.255:0.00		St	0x2		8	83	07	4A	A1	AA	AA	AA	AA	1 J
68	19:39:20.113:0.00		St	0x3		8	0A	47	AA	AA	AA	AA	AA	AA	1 G
67	19:39:20.113:0.00		St	0x1		8	0A	47	4A	A1	AA	AA	AA	AA	1 GJ
66	19:39:19.976:0.00		St	0x2		8	83	07	4A	A1	AA	AA	AA	AA	1 J
65	19:39:19.846:0.00		St	0x3		8	0A	47	AA	AA	AA	AA	AA	AA	1 G
64	19:39:19.846:0.00		St	0x1		8	0A	47	4A	A1	AA	AA	AA	AA	1 GJ
63	19:39:19.735:0.00		St	0x2		8	83	07	4A	A1	AA	AA	AA	AA	1 J
62	19:39:19.608:0.00		St	0x3		8	0A	47	AA	AA	AA	AA	AA	AA	1 G
61	19:39:19.608:0.00		St	0x1		8	0A	47	4A	A1	AA	AA	AA	AA	1 GJ
60	19:39:19.468:0.00		St	0x2		8	83	07	4A	A1	AA	AA	AA	AA	1 J
59	19:39:19.337:0.00		St	0x3		8	0A	47	AA	AA	AA	AA	AA	AA	1 G
58	19:39:19.337:0.00		St	0x1		8	0A	47	4A	A1	AA	AA	AA	AA	1 GJ
57	19:39:19.176:0.00		St	0x2		8	83	07	4A	A1	AA	AA	AA	AA	1 J
56	19:39:19.36:0.00		St	0x3		8	0A	47	AA	AA	AA	AA	AA	AA	1 G

Příloha 1. 11 – Kolize dvou signálů v jedné zprávě CAN. Zprávy odesílané na sběrnici CAN bez protokolu. Software PP2CAN.

Příloha 1. 12 – Data jednoho ze signálů jsou delší, než se vejde do jedné zprávy CAN. Tabulka s důležitými parametry konfiguračního souboru.

Pořadí signálů v souboru	1.	2.
Jak často PID vyčítat	1	1
PID	0x01	0x1798
SID	0x01	0x22
Header requestu	0x7DF	0x7E0
Číslo bajtu v odpovědi	0	0
Délka signálu [bajtů]	4	2
CAN ID zprávy	0x001	0x001
Extended	0	0
Perioda odesílání [ms]	0	0
Poloha ve zprávě [od čísla bajtu]	0	7

```

-----Struktury Signal:
----Struktura[1]----
Jak často PID vycitat: 1
PID: 0x1
Delka PID: 1 bajtu
SID: 0x1
Header requestu: 0x7DF
Cislo bajtu v odpovedi: 0
Delka vycitaneho signalu: 4 bajt/u
Poloha signalu ve zpave CAN: 0
Adresa do jake zpravy CAN patri: 0x3FFED67C
Nenachazi se dalsi pro zpracovani: NULL
Hodnota signalu: 0xAA 0xAA 0xAA 0xAA

----Struktura[2]----
Jak často PID vycitat: 1
PID: 0x1798
Delka PID: 2 bajtu
SID: 0x22
Header requestu: 0x7E0
Cislo bajtu v odpovedi: 0
Delka vycitaneho signalu: 2 bajt/u
Poloha signalu ve zpave CAN: 7
Adresa do jake zpravy CAN patri: 0x3FFED67C
Nenachazi se dalsi pro zpracovani: NULL
Hodnota signalu: 0xAA 0xAA

-----Struktury CAN:
----CAN[1]----
CAN ID: 0x1
Extended: 0
Data: AA AA AA AA AA AA AA
Perioda odesilani: 0
Pocet signalu, ktere k tomuto ID patri: 2
    
```

Příloha 1. 13– Data jednoho ze signálů jsou delší, než se vejde do jedné zprávy CAN. Detail na výpis sériového monitoru. Nahoře načtené signály, dole načtená zpráva CAN.

Nu...	Receive time	Info	Type	Id 1	Id 2	Bytes	B0	B1	B2	B3	B4	B5	B6	B7	ASCII
444	20:55:46.407:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
443	20:55:46.336:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
442	20:55:46.265:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
441	20:55:46.195:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
440	20:55:46.123:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
439	20:55:46.51:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
438	20:55:45.982:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
437	20:55:45.910:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
436	20:55:45.839:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
435	20:55:45.768:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
434	20:55:45.697:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
433	20:55:45.626:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
432	20:55:45.555:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
431	20:55:45.484:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
430	20:55:45.413:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
429	20:55:45.344:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
428	20:55:45.270:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
427	20:55:45.202:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
426	20:55:45.140:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
425	20:55:45.58:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
424	20:55:44.988:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
423	20:55:44.916:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
422	20:55:44.846:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
421	20:55:44.774:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
420	20:55:44.703:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J
419	20:55:44.631:0.00		St	0x1		8	83	07	4A	A1	AA	AA	AA	0A	1 J

Příloha 1. 14– Data jednoho ze signálů jsou delší, než se vejde do jedné zprávy CAN. Zprávy odesílané na sběrnici CAN bez protokolu. Software PP2CAN.

Příloha 1. 15 – Nesmyslné parametry PID, SID a Header requestu. Tabulka s důležitými parametry konfiguračního souboru.

Pořadí signálů v souboru	1.	2.
Jak často PID vyčítat	1	1
PID	0xABC	0x000
SID	0xDEF	0x000
Header requestu	0x123	0x000
Číslo bajtu v odpovědi	0	0
Délka signálu [bajtů]	4	2
CAN ID zprávy	0x001	0x001
Extended	0	0
Perioda odesílání [ms]	0	0
Poloha ve zprávě [od čísla bajtu]	0	4

```

-----Struktury Signal:
---Struktura[1]-----
Jak casto PID vycitat: 1
PID: 0xABC
Delka PID: 2 bajtu
SID: 0xEF
Header requestu: 0x123
Cislo bajtu v odpovedi: 0
Delka vycitaneho signalu: 4 bajt/u
Poloha signalu ve zpave CAN: 0
Adresa do jake zpravy CAN patri: 0x3FFDF9E4
Nenachazi se dalsi pro zpracovani: NULL
Hodnota signalu: 0xAA 0xAA 0xAA 0xAA

---Struktura[2]-----
Jak casto PID vycitat: 1
PID: 0x0
Delka PID: 2 bajtu
SID: 0x0
Header requestu: 0x0
Cislo bajtu v odpovedi: 0
Delka vycitaneho signalu: 2 bajt/u
Poloha signalu ve zpave CAN: 4
Adresa do jake zpravy CAN patri: 0x3FFDF9E4
Nenachazi se dalsi pro zpracovani: NULL
Hodnota signalu: 0xAA 0xAA

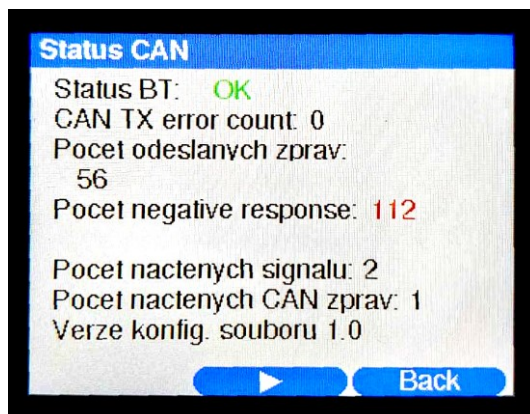
-----Struktury CAN:
---CAN[1]-----
CAN ID: 0x1
Extended: 0
Data: AA AA AA AA AA AA AA AA
Perioda odesilani: 0
Pocet signalu, ktere k tomuto ID patri: 2

```

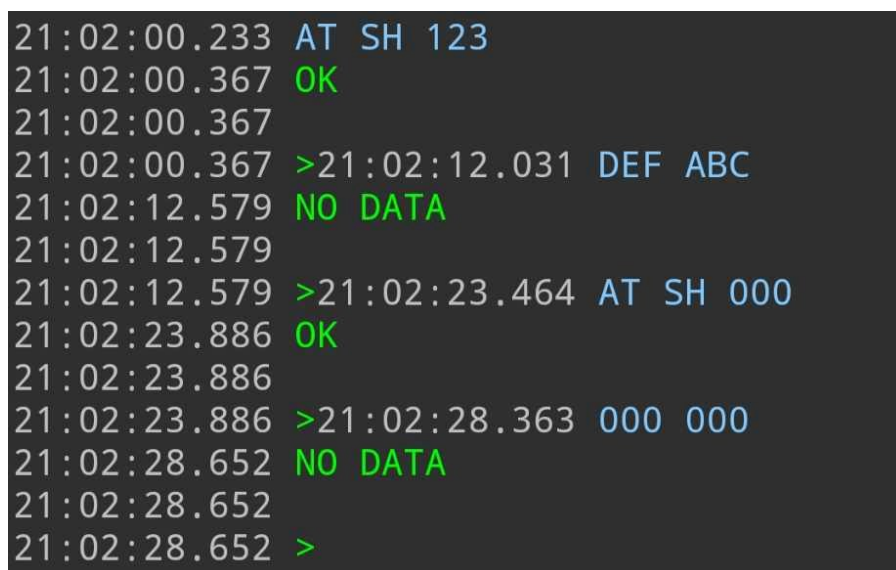
Příloha 1. 16– Nesmyslné parametry PID, SID a Header requestu. Detail na výpis sériového monitoru. Nahoře načtené signály, dole načtená zpráva CAN.

Nu...	Receive time	Info	Type	Id 1	Id 2	Bytes	B0	B1	B2	B3	B4	B5	B6	B7	A
76	21:9:35.131:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
75	21:9:34.626:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
74	21:9:34.155:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
73	21:9:33.685:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
72	21:9:33.192:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
71	21:9:32.706:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
70	21:9:32.181:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
69	21:9:31.603:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
68	21:9:31.108:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
67	21:9:30.562:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
66	21:9:30.13:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
65	21:9:29.545:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
64	21:9:29.43:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
63	21:9:28.543:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
62	21:9:28.1:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
61	21:9:27.488:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
60	21:9:26.969:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
59	21:9:26.444:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
58	21:9:25.975:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
57	21:9:25.455:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
56	21:9:24.930:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
55	21:9:24.409:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
54	21:9:23.909:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
53	21:9:23.406:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
52	21:9:22.833:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1
51	21:9:22.325:0.00		St	0x1		8	AA	AA	AA	AA	AA	AA	AA	AA	1

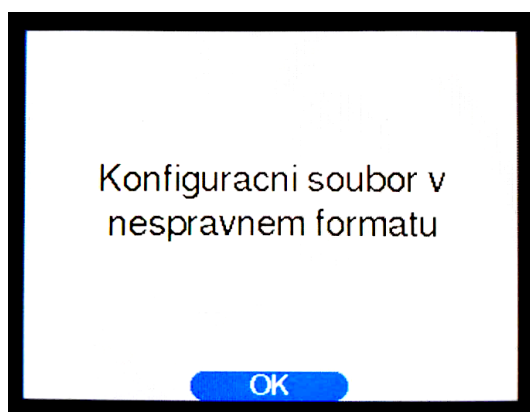
Příloha 1. 17– Nesmyslné parametry PID, SID a Header requestu. Zprávy odesílané na sběrnici CAN bez protokolu. Software PP2CAN.



Příloha 1. 18– Nesmyslné parametry PID, SID a Header requestu. Detail na displej převodníku.



Příloha 1. 19– Nesmyslné parametry PID, SID a Header requestu. Komunikace s adaptérem přes aplikaci Bluetooth Serial.



Příloha 1. 20 – Nesprávný počet oddělovačů v konfiguračním souboru. Detail na displej převodníku.

Příloha 2, Testování převodníku – emulátor vozidla



Příloha 2. 1 – Příklad dat poskytovaných emulátorem vozidla (emulátor podporuje pouze žlutě vyznačené hodnoty).

Time	Chn	ID	Name	Dir	DLC	Data
0.163145	CAN 2	7E0		Rx	8	30 00 00 00 00 00 00 00
0.001068	CAN 2	7E8		Rx	8	21 A4 01 59 00 00 00 00
1.219811	CAN 2	7DF		Rx	8	02 01 8B 00 00 00 00 00
16.054613	CAN 1	1	Konfig_1_msg_1	Rx	8	83 07 4A A1 00 01 E2 40
		3	S1_PID_01_DTC_CNT			3 # of DTCs stored in this ECU
		1	S1_PID_01_MIL_status			1 0 = MIL OFF, 1 = MIL ON
		1	S1_PID_01_MIS_SUP			1 0 = not supported, 1 = supported
		1	S1_PID_01_FUEL_SUP			1 0 = not supported, 1 = supported
		1	S1_PID_01_CCM_SUP			1 0 = not supported, 1 = supported
		0	S1_PID_01_MIS_RDY			0 0 = complete, 1 = not complete
		0	S1_PID_01_FUEL_RDY			0 0 = complete, 1 = not complete
		0	S1_PID_01_CCM_RDY			0 0 = complete, 1 = not complete
		0	S1_PID_01_CAT_SUP			0 0 = not supported, 1 = supported
		1	S1_PID_01_NOX_AFT_SUP			1 0 = not supported, 1 = supported
		1	S1_PID_01_BOOST_SUP			1 0 = not supported, 1 = supported
		0	S1_PID_01_EXH_GAS_SEN_SUP			0 0 = not supported, 1 = supported
		1	S1_PID_01_PM_FILT_SUP			1 0 = not supported, 1 = supported
		0	S1_PID_01_EGR_SUP			0 0 = not supported, 1 = supported
		1	S1_PID_01_CAT_RDY			1 0 = complete, 1 = not complete
		0	S1_PID_01_NOX_AFT_RDY			0 0 = complete, 1 = not complete
		0	S1_PID_01_BOOST_RDY			0 0 = complete, 1 = not complete
		1	S1_PID_01_EXH_GAS_SEN_RDY			1 0 = complete, 1 = not complete
		0	S1_PID_01_PM_FILT_RDY			0 0 = complete, 1 = not complete
		1	S1_PID_01_EGR_RDY			1 0 = complete, 1 = not complete
		123456 km	S22_PID_16A9_Odometer		1E240	Celkova ujeta vzdalenost
1.457464	CAN 1	2	Konfig_1_msg_2	Rx	8	0C 80 64 F9 0A 47 03 33
		800.0000 rpm	S1_PID_0C_EngineRPM		C80	
		100 km/h	S1_PID_0D_VehicleSpeed		64	
		97.6471 %	S22_PID_F445_ThrottlePosNorm		F9	Poloha skrtici klapky, normovana.
		-10.0000 °C	S22_PID_1798_AirTempAfterCooler		A47	Teplota nasavaneho vzduchu za chladicem.
		9.9976 %	S22_PID_102A_GasPedalPos		333	

Příloha 2. 2 – Žádosti o PIDy, o které lze žádat testované vozidlo. Okno 1. Software Vector CANalyzer.

Time	Chn	ID	Name	Dir	DLC	Data
		0	S1_PID_01_EGR_SUP			0 0 = not supported, 1 = supported
		1	S1_PID_01_CAT_RDY			1 0 = complete, 1 = not complete
		0	S1_PID_01_NOX_AFT_RDY			0 0 = complete, 1 = not complete
		0	S1_PID_01_BOOST_RDY			0 0 = complete, 1 = not complete
		1	S1_PID_01_EXH_GAS_SEN_RDY			1 0 = complete, 1 = not complete
		0	S1_PID_01_PM_FILT_RDY			0 0 = complete, 1 = not complete
		1	S1_PID_01_EGR_RDY			1 0 = complete, 1 = not complete
		123456 km	S22_PID_16A9_Odometer		1E240	Celkova ujeta vzdalenost
1.457464	CAN 1	2	Konfig_1_msg_2	Rx	8	0C 80 64 F9 0A 47 03 33
		800.0000 rpm	S1_PID_0C_EngineRPM		C80	
		100 km/h	S1_PID_0D_VehicleSpeed		64	
		97.6471 %	S22_PID_F445_ThrottlePosNorm		F9	Poloha skrtici klapky, normovana.
		-10.0000 °C	S22_PID_1798_AirTempAfterCooler		A47	Teplota nasavaneho vzduchu za chladicem.
		9.9976 %	S22_PID_102A_GasPedalPos		333	
1.457938	CAN 1	3	Konfig_1_msg_3	Rx	8	62 0D AC 11 94 00 0A AA
		98 kPa	S22_PID_F40B_Map		62	Absolutni tlak v sani.
		350.0000 kg/h	S22_PID_1024_AirMassFlowRate		DAC	Hm. prutok nasavaneho vzduchu.
		450.0000 mg	S22_PID_1026_AirMassPerStroke		1194	Hmotnost nasavaneho vzduchu na zdvih.
		10 mbar	S22_PID_162F_DiffPressDpfRaw		A	Dif. tlaku na DPF, nekorigovana hodnota ze snimace.
1.564961	CAN 1	4	Konfig_1_msg_4	Rx	8	1E 33 22 1B 26 03 AA AA
		500.0000 °C	S22_PID_10FB_EGT11		1E33	Teplota spalín pred turbem.
		600.0000 °C	S22_PID_14F7_EGT12		221B	Teplota spalín za turbem.
		700.0000 °C	S22_PID_11B2_EGT13		2603	Teplota spalín pred DPF.
8.117806	CAN 1	5	Konfig_1_msg_5	Rx	8	7F 05 E2 01 A4 01 59 81
		127	S1_PID_8B_diAftStatSupp		7F	dieselAftertreatmentStatusSupported (byte A)
		5	S1_PID_8B_diAftStatus		5	dieselAftertreatmentStatus (byte B)
		88.6274 %	S1_PID_8B_nTrigForDpfRegen		E2	normalizedTriggerForDpfRegen (byte C)
		420 minute	S1_PID_8B_avgTimBetDpfRegens		1A4	averageTimeBetweenDpfRegens (byte D,E)
		345 km	S1_PID_8B_avgDistBetDpfRegens		159	averageDistanceBetweenDpfRegens (byte F,G)
		89 degC	S1_PID_05_EngineCoolantTemp		81	

Příloha 2. 3 - Žádosti o PIDy, o které lze žádat testované vozidlo. Okno 2. Software Vector CANalyzer.

Příloha 3, Testování převodníku – vozidlo

Příloha 3. 1 – Testování na vozidle. Tabulka s důležitými parametry konfiguračního souboru.

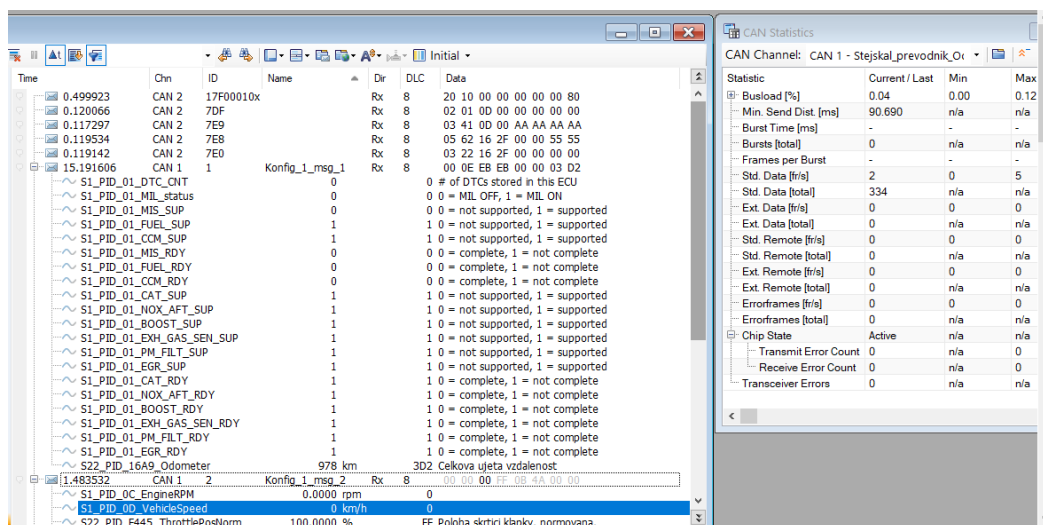
Pořadí signálů v souboru	1.	2.	3.	4.	5.	6.	7.
Jak často PID vyčítat	10	10	1	1	1	1	1
PID	0x01	0x16A9	0x0C	0x0D	0x102A	0x1798	0xF445
SID	0x01	0x22	0x01	0x01	0x22	0x22	0x22
Header requestu	0x7DF	0x7E0	0x7DF	0x7DF	0x7E0	0x7E0	0x7E0
Číslo bajtu v odpovědi	0	0	0	0	0	0	0
Délka signálu [bajtů]	4	4	5	1	2	2	1
CAN ID zprávy	0x001	0x001	0x002	0x002	0x002	0x002	0x002
Extended	0	0	0	0	0	0	0
Perioda odesílání [ms]	0	0	0	0	0	0	0
Poloha ve zprávě [od čísla bajtu]	0	4	0	2	6	4	3

Příloha 3. 2 – Pokračování tabulky z Příloha 3. 1. Tabulka s důležitými parametry konfiguračního souboru.

Pořadí signálů v souboru	8.	9.	10.	11.	12.	13.	14.
Jak často PID vyčítat	1	1	1	1	1	1	1
PID	0x1024	0x1026	0x162F	0xF40B	0x10FB	0x11B2	0x14F7
SID	0x22	0x22	0x22	0x22	0x22	0x22	0x22
Header requestu	0x7E0	0x7E0	0x7E0	0x7E0	0x7E0	0x7E0	0x7E0
Číslo bajtu v odpovědi	0	0	0	0	0	0	0
Délka signálu [bajtů]	2	2	2	1	2	2	2
CAN ID zprávy	0x003	0x003	0x003	0x003	0x004	0x004	0x004
Extended	0	0	0	0	0	0	0
Perioda odesílání [ms]	0	0	0	0	0	0	0
Poloha ve zprávě [od čísla bajtu]	1	3	5	0	0	4	2

Příloha 3.3 – Pokračování tabulky z Příloha 3.2. Tabulka s důležitými parametry konfiguračního souboru.

Pořadí signálů v souboru	15.	16.
Jak často PID vyčítat	5	5
PID	0x05	0x8B
SID	0x01	0x01
Header requestu	0x7DF	0x7DF
Číslo bajtu v odpovědi	0	0
Délka signálu [bajtů]	1	7
CAN ID zprávy	0x005	0x005
Extended	0	0
Perioda odesílání [ms]	0	0
Poloha ve zprávě [od čísla bajtu]	7	0



Příloha 3.4 – Vozidlo stojí, motor vypnut. Zachycená data z převodníku a z OBD zásuvky (1). Software Vector CANalyzer.

The screenshot shows the Vector CANalyzer interface. The main window displays a list of captured CAN messages. The columns are: Time, Chn, ID, Name, Dir, DLC, and Data. The messages include various engine and vehicle parameters such as engine RPM, vehicle speed, throttle position, and air flow rate. The right-hand pane shows 'CAN Statistics' for 'CAN Channel: CAN 1 - Stejskal_převodnik_Or'. The statistics include:

- Statistic: Busload [%], Current / Last: 0.04, Min: 0.00, Max: 0.12
- Min. Send Dist. [ms]: 90.690, n/a, n/a
- Burst Time [ms]: -, -, -
- Bursts [total]: 0, n/a, n/a
- Frames per Burst: -, -, -
- Std. Data [fr/s]: 2, 0, 5
- Std. Data [total]: 334, n/a, n/a
- Ext. Data [fr/s]: 0, 0, 0
- Ext. Data [total]: 0, n/a, n/a
- Std. Remote [fr/s]: 0, 0, 0
- Std. Remote [total]: 0, n/a, n/a
- Ext. Remote [fr/s]: 0, 0, 0
- Ext. Remote [total]: 0, n/a, n/a
- Errorframes [fr/s]: 0, 0, 0
- Errorframes [total]: 0, n/a, n/a
- Chip State: Active, n/a, n/a
- Transmit Error Count: 0, n/a, 0
- Receive Error Count: 0, n/a, 0
- Receiver Errors: 0, n/a, n/a

Příloha 3. 5 – Vozidlo stojí, motor vypnut. Zachycená data z převodníku a z OBD zásuvky (2). Software Vector CANalyzer.

The screenshot shows the OBDwiz software interface. The main window displays a table of OBD2 diagnostic trouble codes (DTCs) and their corresponding values and units. The table has columns for PID, Module, Description, Value, Units, Min, Mean, and Max. The data is as follows:

PID	Module	Description	Value	Units	Min	Mean	Max
SAE 0x01	OBD2	I/M Readiness Data	MIL Off, # Trouble Code...				
SAE 0x04	OBD2	Calculated load value	0	%	0	0	0
SAE 0x05	OBD2	Engine coolant temperature	15	°C	14	14.12	15
SAE 0x0B	OBD2	Intake manifold absolute pressure	98	kPa	98	98	98
SAE 0x0C	OBD2	Engine RPM	0	RPM	0	0	0
SAE 0x0D	OBD2	Vehicle speed	0	km/h	0	0	0
SAE 0x0F	OBD2	Intake air temperature	23	°C	21	22.05	23
SAE 0x10	OBD2	Mass air flow rate	0.91	g/s	0.91	0.91	0.91
SAE 0x11	OBD2	Absolute throttle position	89.02	%	89.02	89.02	89.02
SAE 0x13	OBD2	Location of oxygen sensors	Bank 1: Sensor 1, Senso...		3	3	3
SAE 0x1C	OBD2	OBD requirements to which vehicle or engine is certified	EOBD		6	6	6
SAE 0x1F	OBD2	Time since engine start	0	sec	0	0	0
SAE 0x21	OBD2	Distance traveled while MIL is activated	0	km	0	0	0
SAE 0x23	OBD2	Fuel rail pressure	450	kPa	450	450	450
SAE 0x30	OBD2	Number of warm-ups since DTCs cleared	0		0	0	0
SAE 0x31	OBD2	Distance traveled since DTCs cleared	0	km	0	0	0
SAE 0x33	OBD2	Barometric pressure	98	kPa	98	98	98
SAE 0x34	OBD2	O2 sensor current wide range (Bank 1, Sensor 1)	-0.004	mA	-0.004	-0.004	-0.004
SAE 0x34	OBD2	O2 sensor lambda wide range (current probe) (Bank 1, S...	15.99		15.99	15.99	15.99
SAE 0x35	OBD2	O2 sensor current wide range (Bank 1, Sensor 2)	-0.004	mA	-0.004	-0.004	-0.004
SAE 0x35	OBD2	O2 sensor lambda wide range (current probe) (Bank 1, S...	15.99		15.99	15.99	15.99
SAE 0x41	OBD2	Monitor status this drive cycle	MilOff: Not Available				

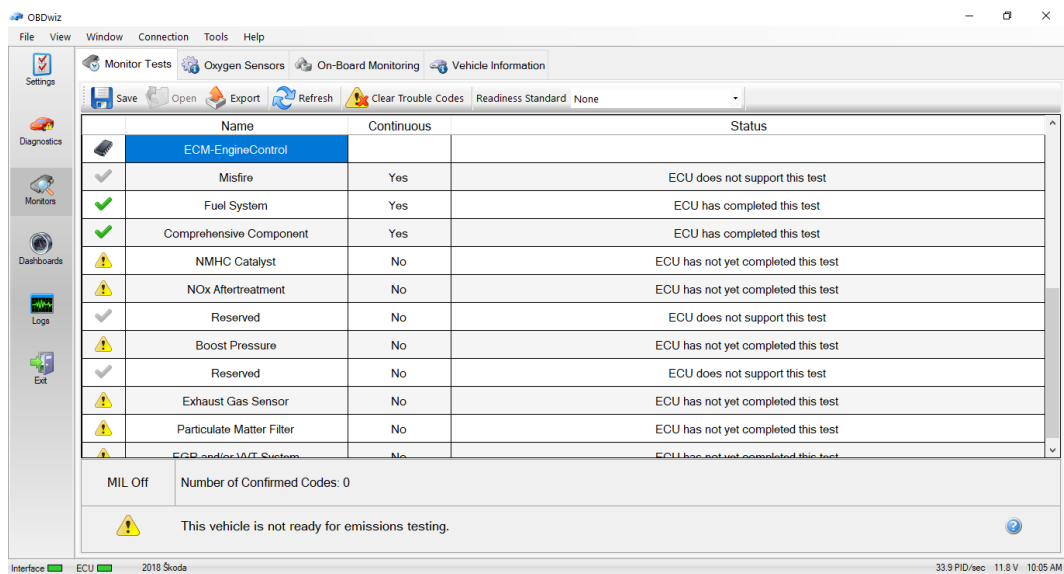
Příloha 3. 6 – Vozidlo stojí, motor vypnut. Zachycená data z OBDwiz pro porovnání (1).

PID	Module	Description	Value	Units	Min	Mean	Max
SAE 0x67	OBD2	Engine coolant temperature 1	15	°C	14	14.3	15
SAE 0x67	OBD2	Engine coolant temperature sensor 1, 2 supported	1		1	1	1
SAE 0x69	OBD2	Actual EGR A duty cycle/position	0.39	%	0.39	0.39	0.39
SAE 0x69	OBD2	Actual EGR B duty cycle/position	0	%	0	0	0
SAE 0x69	OBD2	Commanded EGR A duty cycle/position	0	%	0	0	0
SAE 0x69	OBD2	Commanded EGR and EGR error supported data	63		63	63	63
SAE 0x69	OBD2	Commanded EGR B duty cycle/position	0	%	0	0	0
SAE 0x69	OBD2	EGR A error	99.22	%	99.22	99.22	99.22
SAE 0x69	OBD2	EGR B error	0	%	0	0	0
SAE 0x77	OBD2	Charge air cooler temperature bank 1, sensor 1	17	°C	16	16.5	17
SAE 0x77	OBD2	Charge air cooler temperature bank 1, sensor 2	16	°C	16	16	16
SAE 0x77	OBD2	Support of charge air cooler temperature data	3		3	3	3
SAE 0x78	OBD2	Exhaust gas temperature bank 1, sensor 1	20.6	°C	20.6	20.9	21.2
SAE 0x78	OBD2	Exhaust gas temperature bank 1, sensor 3	21.6	°C	21.6	22.84	24.2
SAE 0x78	OBD2	Support of exhaust gas temperature bank 1 sensor data	5		5	5	5
SAE 0x8B	OBD2	Diesel aftertreatment status	0		0	0	0
SAE 0x8B	OBD2	Diesel aftertreatment status supported data	15		15	15	15
SAE 0x8B	OBD2	DPF regen status	PF_REGEN: NO		0	0	0
SAE 0x8B	OBD2	DPF regen type	PF_REGEN: PASSIVE		0	0	0
SAE 0x8B	OBD2	NOx adsorber desulfurization status	NOX_ADS_DESULF: NO		0	0	0
SAE 0x8B	OBD2	NOx adsorber regen status	NOX_ADS_REGEN: NO		0	0	0

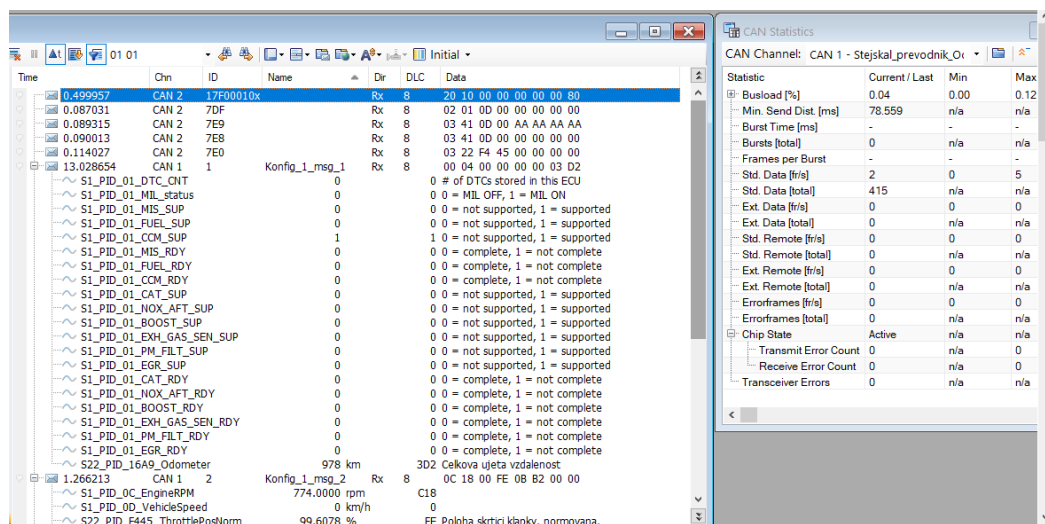
Příloha 3. 7 – Vozidlo stojí, motor vypnut. Zachycená data z OBDwiz pro porovnání (2).

Time	Chn	ID	Name	Dir	DLC	Data
0.043386	CAN 2	7E0		Rx	8	03 22 F4 08 00 00 00 00
0.007882	CAN 2	7E8		Rx	8	04 62 F4 08 62 55 55 55
0.054580	CAN 1	3	Konfig_1_msg_3	Rx	8	62 00 00 00 00 00 00 AA
0.032665	CAN 2	7E0		Rx	8	03 22 F4 45 00 00 00 00
0.002726	CAN 2	7E8		Rx	8	04 62 F4 45 FF 55 55 55
0.065035	CAN 1	2	Konfig_1_msg_2	Rx	8	00 00 00 FF 08 4A 00 00
0.045216	CAN 2	7DF		Rx	8	02 01 01 00 00 00 00 00
0.009760	CAN 2	7E8		Rx	8	06 41 01 00 0E EB EB 00
0.003246	CAN 2	7E9		Rx	8	06 41 01 00 04 00 00 AA
0.082020	CAN 2	7DF		Rx	8	02 01 05 00 00 00 00 00
0.004752	CAN 2	7E8		Rx	8	03 41 05 36 00 00 00 00
0.002676	CAN 2	7E9		Rx	8	03 41 05 36 AA AA AA AA
0.009768	CAN 2	7DF		Rx	8	02 01 0C 00 00 00 00 00
0.009592	CAN 2	7E8		Rx	8	04 41 0C 00 00 00 00 00
0.003102	CAN 2	7E9		Rx	8	04 41 0C 00 00 AA AA AA
0.035661	CAN 2	17F00010x		Rx	8	20 10 00 00 00 00 00 80
0.043734	CAN 2	7DF		Rx	8	02 01 00 00 00 00 00 00
0.007524	CAN 2	7E8		Rx	8	03 41 00 00 00 00 00 00
0.002702	CAN 2	7E9		Rx	8	03 41 00 00 AA AA AA AA
0.109908	CAN 2	7E0		Rx	8	03 22 10 24 00 00 00 00
0.017346	CAN 2	7E8		Rx	8	05 62 10 24 00 00 55 55
0.102725	CAN 2	7E0		Rx	8	03 22 10 26 00 00 00 00
0.017310	CAN 2	7E8		Rx	8	05 62 10 26 00 00 55 55
0.100781	CAN 2	7E0		Rx	8	03 22 10 2A 00 00 00 00
0.019188	CAN 2	7E8		Rx	8	05 62 10 2A 00 00 55 55
0.0078751	CAN 2	17F00010x		Rx	8	20 10 00 00 00 00 00 80
0.021174	CAN 2	7E0		Rx	8	03 22 10 FB 00 00 00 00
0.012852	CAN 2	7E8		Rx	8	03 7F 22 78 55 55 55 55
0.007226	CAN 2	7E8		Rx	8	05 62 10 FB 08 83 55 55
0.009946	CAN 2	7E0		Rx	8	03 22 11 82 00 00 00 00
0.010058	CAN 2	7E8		Rx	8	03 7F 22 78 55 55 55 55

Příloha 3. 8 – Vozidlo stojí, motor je vypnut. Zvýrazněná žádost, pod ní dvě odpovědi. Software Vector CANalyzer.



Příloha 3. 9 – Vozidlo stojí, motor vypnut. Zachycená data z OBDwiz pro porovnání (3).



Příloha 3. 10 – Vozidlo stojí, volnoběh. Zachycená data z převodníku a z OBD zásuvky (1). Software Vector CANalyzer.

The screenshot shows the Software Vector CANalyzer interface. The main window displays a list of CAN messages with columns for Time, Chn, ID, Name, Dir, DLC, and Data. The right-hand pane shows 'CAN Statistics' for 'CAN Channel: CAN 1 - Stejskal_prevodnik_Or', listing various metrics like Busload, Min. Send Dist., Bursts, and Error frames.

Příloha 3. 11 – Vozidlo stojí, volnoběh. Zachycená data z převodníku a z OBD zásuvky (2). Software Vector CANalyzer.

PID	Module	Description	Value	Units	Min	Mean	Max
SAE 0x01	OBD2	MIL Readiness Data	MIL Off, # Trouble Code.				
SAE 0x04	OBD2	Calculated load value	45.88	%	0	4.43	46.27
SAE 0x05	OBD2	Engine coolant temperature	19	°C	14	15	19
SAE 0x0B	OBD2	Intake manifold absolute pressure	98	kPa	98	98	98
SAE 0x0C	OBD2	Engine RPM	770	RPM	0	83.55	859
SAE 0x0D	OBD2	Vehicle speed	0	km/h	0	0	0
SAE 0x0F	OBD2	Intake air temperature	19	°C	19	22.7	25
SAE 0x10	OBD2	Mass air flow rate	9.88	g/s	0.91	2.01	12.41
SAE 0x11	OBD2	Absolute throttle position	83.53	%	83.53	88.45	89.02
SAE 0x13	OBD2	Location of oxygen sensors	Bank 1: Sensor 1, Senso...		3	3	3
SAE 0x1C	OBD2	OBD requirements to which vehicle or engine is certified	EOBD		6	6	6
SAE 0x1F	OBD2	Time since engine start	76	sec	0	5.22	76
SAE 0x21	OBD2	Distance traveled while MIL is activated	0	km	0	0	0
SAE 0x23	OBD2	Fuel rail pressure	29740	kPa	450	4052.92	40220
SAE 0x30	OBD2	Number of warm-ups since DTCs cleared	0		0	0	0
SAE 0x31	OBD2	Distance traveled since DTCs cleared	0	km	0	0	0
SAE 0x33	OBD2	Barometric pressure	98	kPa	98	98	98
SAE 0x34	OBD2	O2 sensor current wide range (Bank 1, Sensor 1)	-0.004	mA	-0.004	-0.004	-0.004
SAE 0x34	OBD2	O2 sensor lambda wide range (current probe) (Bank 1, S...	2.786		2.711	14.618	15.99
SAE 0x35	OBD2	O2 sensor current wide range (Bank 1, Sensor 2)	-0.004	mA	-0.004	-0.004	-0.004
SAE 0x35	OBD2	O2 sensor lambda wide range (current probe) (Bank 1, S...	2.788		2.707	14.632	15.99
SAE 0x41	OBD2	Monitor status, this device reads...	Micro: Not Available Fuel				

Příloha 3. 12 – Vozidlo stojí, volnoběh. Zachycená data z OBDwiz pro porovnání (1).

OBDwiz

File View Window Connection Tools Help

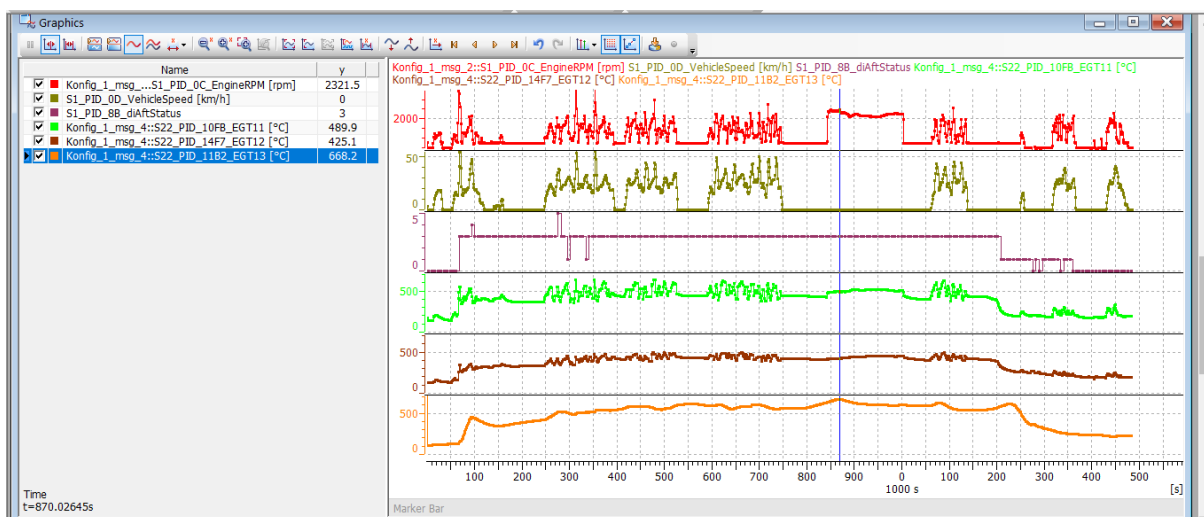
Trouble Codes Freeze Frame PID Values Diagnostic Report Console

Save Open Configure

PID	Module	Description	Value	Units	Min	Mean	Max
SAE 0x67	OBD2	Engine coolant temperature 1	21	°C	14	15.3	21
SAE 0x67	OBD2	Engine coolant temperature sensor 1, 2 supported	1		1	1	1
SAE 0x69	OBD2	Actual EGR A duty cycle/position	39.22	%	0.39	4.28	40
SAE 0x69	OBD2	Actual EGR B duty cycle/position	0	%	0	0	0
SAE 0x69	OBD2	Commanded EGR A duty cycle/position	39.22	%	0	3.96	40.39
SAE 0x69	OBD2	Commanded EGR and EGR error supported data	63		63	63	63
SAE 0x69	OBD2	Commanded EGR B duty cycle/position	0	%	0	0	0
SAE 0x69	OBD2	EGR A error	0	%	-1.56	86.27	99.22
SAE 0x69	OBD2	EGR B error	0	%	0	0	0
SAE 0x77	OBD2	Charge air cooler temperature bank 1, sensor 1	18	°C	16	17.01	19
SAE 0x77	OBD2	Charge air cooler temperature bank 1, sensor 2	19	°C	16	16.19	19
SAE 0x77	OBD2	Support of charge air cooler temperature data	3		3	3	3
SAE 0x78	OBD2	Exhaust gas temperature bank 1, sensor 1	127.5	°C	20.6	34.99	128.6
SAE 0x78	OBD2	Exhaust gas temperature bank 1, sensor 3	56.7	°C	21	24.71	56.7
SAE 0x78	OBD2	Support of exhaust gas temperature bank 1 sensor data	5		5	5	5
SAE 0x8B	OBD2	Diesel aftertreatment status	0		0	0	0
SAE 0x8B	OBD2	Diesel aftertreatment status supported data	15		15	15	15
SAE 0x8B	OBD2	DPF regen status	PF_REGEN: NO		0	0	0
SAE 0x8B	OBD2	DPF regen type	PF_REGEN: PASSIVE		0	0	0
SAE 0x8B	OBD2	NOx adsorber desulfurization status	NOX_ADS_DESULF: NO		0	0	0
SAE 0x8B	OBD2	NOx adsorber regen status	NOX_ADS_REGEN: NO		0	0	0

Interface ECU 2018 Škoda 20.6 PID/sec 14.7V 10:20 AM

Příloha 3. 13 – Vozidlo stojí, volnoběh. Zachycená data z OBDwiz pro porovnání (2).



Příloha 3. 14 – Testovací jízda. Zachycená data z převodníku a z OBD zásuvky. Software Vector CANalyzer.