TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Elektricky řízená automobilová sedačka

Bakalářská práce

Studijní program: B2612 Elektrotechnika a informatika
Studijní obor: Elektronické informační a řídicí systémy

Autor práce: **Vít Pardubský**
Vedoucí práce: Ing. Tomáš Martinec, Ph.D.
Ústav mechatroniky a technické informatiky

Liberec 2022

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Zadání bakalářské práce

Elektricky řízená automobilová sedačka

Jméno a příjmení: **Vít Pardubský**
Osobní číslo: M18000040
Studijní program: B2612 Elektrotechnika a informatika
Studijní obor: Elektronické informační a řídicí systémy
Zadávající katedra: Ústav mechatroniky a technické informatiky
Akademický rok: **2021/2022**

Zásady pro vypracování:

1. Seznamte se s funkcí elektricky řízené sedačky ŠKODA s pamětí a navrhněte možný způsob jejího řízení.
2. Seznamte se s CAN sběrnici, jejím diagnostickým protokolem a navrhněte elektroniku pro CAN modul PCAN-MicroMod FD včetně nutných komponent pro ovládání sedačky.
3. Realizujte navržený HW a vytvořte pro něj potřebný SW.
4. Vytvořte řídicí aplikaci pro PC, která bude umožňovat monitorování stavu sedačky a její ovládání, zdokumentujte a otestujte celý systém řízení sedačky.

Rozsah grafických prací:	dle potřeby dokumentace
Rozsah pracovní zprávy:	30–40 stran
Forma zpracování práce:	tištěná/elektronická
Jazyk práce:	Čeština

Seznam odborné literatury:

[1] ROBERT BOSCH GmbH. CAN Specification Version 2.0. Postfach 300204. D-7000 Stuttgart 30. 1991.

[2] ISO 14230-3:1999. Road vehicles —Diagnostic systems —Keyword Protocol 2000 —Part 3: Application layer. International Organization for Standardization.

[3] CORRIGAN, Steve. Introduction to the Controller Area Network (CAN) [online]. Revised 07/2008. Dallas: Texas Instruments, 2008. [cit. 2016-03-18]. Dostupné z: http://www.ti.com/lit/an/sloa101a/sloa101a.pdf

Vedoucí práce:	Ing. Tomáš Martinec, Ph.D. Ústav mechatroniky a technické informatiky
Datum zadání práce:	12. října 2021
Předpokládaný termín odevzdání:	16. května 2022

L.S.

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

doc. Ing. Josef Černohorský, Ph.D.
vedoucí ústavu

V Liberci dne 12. října 2021

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

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

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci 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.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

13. května 2022 Vít Pardubský

Poděkování

Rád bych poděkoval Ing. Tomáši Martincovi, Ph.D. za umožnění bakalářské práce ve spolupráci se ŠKODA AUTO a za trpělivost.

Díky si zaslouží též Zdeněk Herda, Ph.D. za sdílení svých zkušeností při konzultacích a za zapůjčení některých potřebných zařízení.

V neposlední řadě též děkuji své rodině za podporu ve studiu.

Abstrakt

Cílem této bakalářské práce je návrh a realizace softwarové a hardwarové části externího řízení automobilové sedačky. Detailně je popsán použitý komunikační protokol CAN – Controller Area Network, jeho vlastnosti, principy jeho fungování a použití v automobilovém průmyslu. Praktická část popisuje postup návrhu hardwarového řešení ovládání sedačky s použitím modulu PCAN-MicroMod FD. Zmíněny jsou detaily ze schématu, návrhu desky plošného spoje a konfigurace CAN zpráv. Poslední částí je tvorba aplikace pro Windows v jazyce C#, která umožňuje odesílání CAN zpráv pro ovládání automobilové sedačky, vyčítání její polohy a její vykreslení.

Klíčová slova

Automotive, sedačka, inteligentní řízení, CAN, PCAN-USB, PCAN-MicroMod FD

Abstract

The goal of this bachelor thesis is designing and realizing software and hardware for external control of a car seat. Communication protocol CAN – Controller Area Network is used for this purpose and its properties, the way it functions and its uses in the automotive industry are described in detail. The practical part describes the process of creating a hardware-based solution to controlling a car seat using the PCAN-MicroMod FD module. Details from the schematic, designing of a printed circuit board and configuring CAN messages are mentioned. The last part is about writing a C# Windows application, that enables CAN message sending to control the car seat, reading its current location data and visualizing it.

Keywords

Automotive, car seat, intelligent control, CAN, PCAN-USB, PCAN-MicroMod FD

Obsah

Úvod 9

1 Teoretická část 10

1.1 CAN – Controller Area Network 11

1.2 CAN – Fyzická vrstva 12

1.3 CAN – Přenosová vrstva 15

1.4 Hardware 22

2 Praktická část 25

2.1 Použitý hardware a software 25

2.2 Plošný spoj 25

2.3 Schéma 26

2.4 Aplikace na PC 37

2.5 Výsledky 41

Závěr 42

Seznam použité literatury 43

Přílohy 46

A. Hlavní schéma zapojení 46

B. Schéma digitálních výstupů 47

C. Schémata napájení,USB a SD karty 48

D. Schéma digitálních vstupů 49

E. Schéma analogových vstupů 50

F. Plošný spoj 51

Seznam obrázků, grafů, tabulek

Obr. 1 Vrstvy protokolu CAN [4] 11
Obr. 2 Hladiny CAN 12
Obr. 3 Zapojení linky CAN 13
Obr. 4 Možnosti terminace 14
Obr. 5 Průběh arbitráže 15
Obr. 6 Tabulka Data Length Code 18
Obr. 7 Linky CAN ve voze [10] 20
Obr. 8 Diagnostické zprávy CAN 21
Obr. 9 PCAN-MicroMod FD [14] 22
Obr. 10 Evaluation Board [18] 23
Obr. 11 Ilustrace ovládání sedačky ŠKODA Superb 3 [19] 23
Obr. 12 Zapojení 24
Obr. 13 Blokové schéma praktické části 26
Obr. 14 Originální symbol v dokumentaci a odpovídající symbol v EAGLE 27
Obr. 15 Schéma napájení 28
Obr. 16 Schéma CAN 29
Obr. 17 Schéma micro SD karty 30
Obr. 18 Schéma digitálních vstupů 30
Obr. 19 Schéma digitálních výstupů 31
Obr. 20 Části kompletní desky 32
Obr. 21 Srovnání *TOP a BOTTOM* vrstev 33
Obr. 22 Pohled na prokovení v Gerber souborech 33
Obr. 23 Rozpis výdajů na výrobu 34
Obr. 24 Použité hodnoty a jejich přiřazení v konfiguraci 36
Obr. 25 Blokový diagram aplikace 37
Obr. 26 Vzhled aplikace 38
Obr. 27 Hotový výrobek a obal 41

Úvod

V budovách Technické univerzity v Liberci jsou vystavené výsledky spolupráce ŠKODA AUTO a TUL, což mě inspirovalo k výběru zadání bakalářské práce v Technologickém centru Česana, kde vypomáhám v rámci dlouhodobé spolupráce s Ústavem teorie informace a automatizace. Výzkumný oddíl, kde asistuji, se zabývá převážně vývojem a testováním rozhraní mezi vozem a člověkem. Do této oblasti tedy spadá i zadání této práce.

Každý prvek ovládání auta musí být během vývoje testován, aby byla nalezena jeho nejvhodnější podoba do sériové výroby. Cílem této práce je vytvořit hardwarové řešení pro ovládání elektricky nastavitelné sedačky a odměřování její aktuální polohy, které usnadní některé kroky testování. Poloha sedačky patří mezi první nastavení, které řidič vozu mění, a poloha sedačky má vliv na výsledky mnoha dalších měření. Se změnou polohy je pro řidiče jiný výhled ven z vozu, mění se relativní poloha ostatních ovládacích prvků od těla řidiče a podobně. Pro lepší kvalitu zpracování dat ostatních testů je tedy důležité znát fyzické vlastnosti řidiče, jako jsou výška, váha, věk a pohlaví, spolu s polohou, ve které seděl. Data je možné nadále zpracovávat a vytvořit teoreticky ideální pozici sedačky pro řidiče každého typu. Lze si představit luxusní vůz, který by po nasednutí nového řidiče sám nastavil sedačku do pohodlné polohy, nebo úpravy některých prvků vozu podle fyzických vlastností řidičů na trhu, kde se vůz bude prodávat. Externí ovládání sedačky přes počítač umožňuje například nastavení sedačky do odhadované vhodné polohy podle popisu řidiče, ukládání polohy sedačky do databáze měření, vrátit po měření sedačku do základní polohy a podobně. Jelikož je vývoj a testování neustále se měnící proces, tak výsledný hardware této práce bude obsahovat momentálně nadbytečné vstupy a výstupy pro možnost přidání více funkcí v budoucnu bez potřeby tvorby nových plošných spojů.
Na základě konzultace se zadavatelem práce byly formulovány následující technické požadavky pro výsledný hardware:

- Možnost spojení s PC přes CAN, USB a RS-232
- Vhodné výstupní obvody pro ovládání elektricky nastavitelné sedačky
- Napájení přímo z USB, externí napájení 5 V a externí napájení 12 V z autobaterie
- Micro SD karta na desce plošného spoje pro případné logování většího množství dat
- Vyvést I2C a SPI na pinové lišty pro možnost připojení externích zařízení
- Výsledný výrobek bude v obalu vhodném na jeho ochranu a montáž na měření v technologickém centru. Pro připojení bude mít obal otvory na USB, napájení, CAN (konektor Canon 9, někdy značený D-SUB 9) a Canon 15 na spojení s vozem

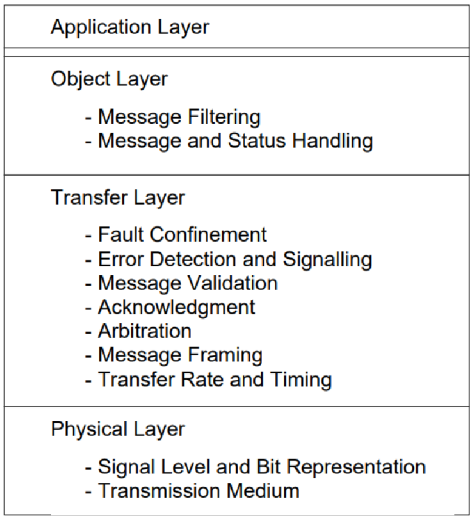
1 Teoretická část

V moderních vozech se nachází desítky řídících jednotek (některé vozy mají i kolem 150 řídících jednotek [1]), mnoho senzorů a bezpečnostních prvků. Účelem používání těchto technologií je snaha, dosáhnout maximálně ekonomické, ekologické a bezpečné jízdy. Elektronické řídící jednotky (v anglických zdrojích *Electronic Control Unit*) bývají navrženy pro plnění různých specializovaných funkcí, například řídící jednotka motoru udržuje ideální podmínky v motoru pro nižší spotřebu a emise, jednotka řízení brzd snižuje riziko smyku a v některých případech automaticky brzdí při zaznamenání nebezpečí. Jelikož při jízdě automobilem může rozhodovat o životě a smrti i několik milisekund, musí být použit vhodný způsob komunikace mezi důležitými jednotkami. Mezi klíčové vlastnosti, které musí komunikační protokol splňovat pro použití v automobilovém průmyslu patří:

- **Spolehlivost a načasování** - Přenos dat musí proběhnout úspěšně a rychle, jinak by byla ohrožena funkce bezpečnostních prvků vozu. Důležité zprávy musí být upřednostněny v přenosu. Zahrnuje například také korekci chyb, odolnost proti rušení a poškození vodičů. Selhání přenosu by mohlo mít katastrofické následky při selhání stabilizačních a brzdových systémů, airbagů a řízení.
- **Rychlost přenosu** - Musí být dostatečná pro množství dat generované ve voze
- **MultiMaster** - Jakákoliv důležitá jednotka musí být schopna zahájit přenos dat, aniž by musela čekat na dotaz od hlavní řídící jednotky
- **Složitost systému a univerzální použití** - Jednodušší zapojení s menším počtem kabelů snižují cenu. Možnost použít stejné vodiče pro více funkcí dále snižuje složitost systému po fyzické i softwarové stránce (nutnost překladu mezi různými systémy, změny hladin napětí…)
- **Zabezpečení** - V této práci není zabezpečení rozebíráno, je ale nutné zmínit jeho důležitost. Špatné zabezpečení může vést k snadnější krádeži vozu a selhání zabezpečení bezdrátového spojení může vést i ke ztrátě kontroly nad vozem za jízdy. [2]

1.1 CAN – Controller Area Network

Jedná se o sériový komunikační protokol popisující fyzickou vrstvu přenosu, formáty přenášené zprávy a postup přenosu zpráv. Protokol CAN a jeho verze patří mezi nejpoužívanější komunikační protokoly v automobilovém průmyslu [3]. Popis funkce začíná od fyzické vrstvy se zmínkami, jak určitá vlastnost CAN řeší potřeby automobilového průmyslu. [4]



Obr. 1 Vrstvy protokolu CAN [4]

1.1.1 Rozdíly verzí CAN

Za dobu existence CAN vznikly úpravy *Classical CAN* pro různé účely. Tato práce rozebírá detailně *CAN High-Speed (CAN 2.0),* v této podkapitole jsou informace o vybraných jiných verzích.

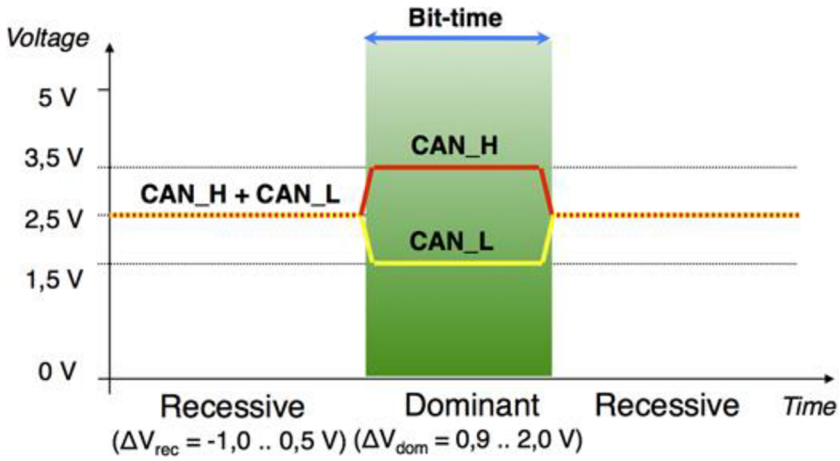
CAN lze zhruba rozdělit na protokoly nižších vrstev OSI (fyzická a linková) a protokoly vyšších vrstev (síťová, transportní a vyšší). V nižších vrstvách je nejběžnější například:
CAN High-speed - rychlost až 1 Mbps s možností různých topologií, standardně linka, 8 bytů dat
CAN FD – rychlosti až 2, 5, 10 Mbps podle použitých fyzických prvků, až 64 bytů dat
CAN XL – rychlosti až 10 Mbps s možností datových rámců až 2048 bytů
CAN Fault Tolerant – odolnost proti rušení díky většímu napětí a stínění, rychlost 125 kbps

Ve vyšších vrstvách jsou desítky protokolů, některé s úzkou specializací na konkrétní potřeby určitého průmyslu. Například:
CANaerospace byl vytvořen pro robustní spojení přístrojů v letadlech a soustředí se na zvýšenou spolehlivost přenosu a odolnost proti výpadkům.
Unified Diagnostic Services řídí komunikaci mezi vozidlem a diagnostickým přístrojem podle norem konkrétního státu, typu vozidla, roku výroby a podobně.
Can-over-IP slouží pro přenos CAN rámců přes internetový protokol, což umožňuje spojovat CAN sítě na větší vzdálenosti a bezdrátově.
CANopen je obecný automatizační protokol pro řídící systémy, definuje vlastnosti zařízení na síti, objekty, události a typy zpráv. Obsahuje zprávy pro řízení podřízených jednotek, synchronizaci, nouzové stavy, přenos dat s jedinečným identifikátorem a další.

1.2 CAN – Fyzická vrstva

Je více možných kabelových zapojení (dvouvodičové se společnou zemí obvodu, jeden vodič+zem, upravené fyzické vrstvy jiných protokolů) [5], zaměření této práce je na popis dvouvodičového zapojení *CAN High-speed*, jelikož je používané v automobilech a ve spojení se sedačkou.

Podle standardu ISO 11898-2 [6] se používá kroucená dvojlinka. Ta snižuje vyzařované elektromagnetické rušení, snižuje příchozí rušení, je levná a zabere málo prostoru v kabelových svazcích ve voze. CAN funguje na principu diferenciálního/rozdílového napětí, data tedy nejsou symbolizována určitou hodnotou napětí na jednom vodiči, ale rozdílem napětí mezi oběma vodiči. Jednotlivým vodičům se se říká CAN High (CAN_H, CANH) a CAN Low (CAN_L, CANL) a v klidové poloze je na obou vodičích stejné napětí 2,5 V a rozdíl je tedy 0 V. Klidovému stavu se říká *recessive* (český název by mohl být *klidový/ustupující* stav). Stavu, kdy je CAN High buzený na vyšší napětí (3,5 V) a CAN Low na nižší napětí (1,5 V), se říká *dominant* (česky by stav šlo nazvat *převládající/dominantní* a z elektronického hlediska *buzený*). Recesivní stav značí log 1 a dominantní stav je log 0, takže je nepoužívaná sběrnice v trvalém logickém stavu 1. Na obrázku jsou pásma napětí, ve kterých platí logické stavy. [7]

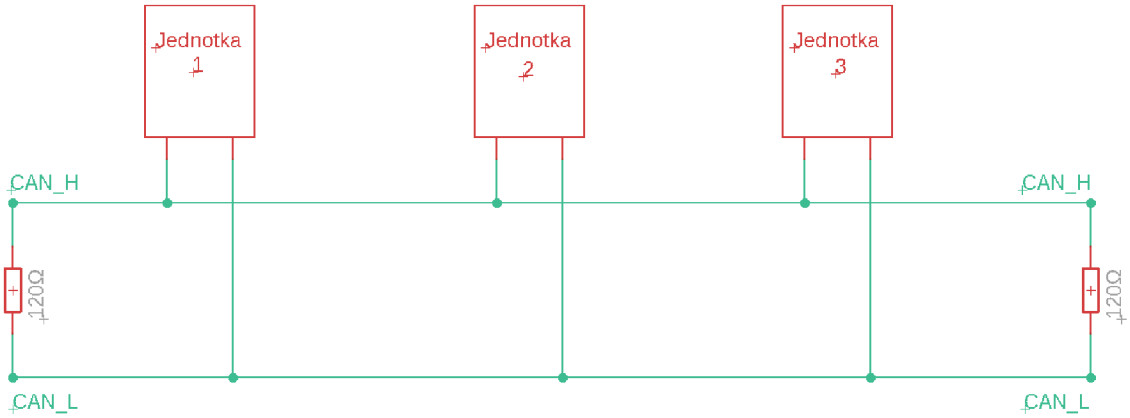


Obr. 2 Hladiny CAN

Posílání informace jako rozdíl napětí na kroucené dvoulince poskytuje výhody zejména v ochraně proti rušení. Naindukované napětí bude přibližně stejné na obou vodičích, protože jsou fyzicky blízko, a rozdíl napětí na vodičích se tedy nijak výrazně nezmění.

1.2.1 Zapojení linky

Připojení zařízení na linku CAN je jednoduché, stačí totiž T spojení pro každou jednotku z hlavní linky. Je rozumné používat jednoznačné barvy pro CAN H a CAN L, jinak se špatně hledá, která jednotka má prohozené vodiče a brání komunikaci. V technologickém centru je používána pro CAN L vždy hnědá barva a CAN H je druhou barvou dvojlinky, která je v laboratoři, což bývá fialová. Odbočky do jednotek z hlavní linky jsou v ideálním případě co nejkratší, podle ISO 11898-2 maximálně 30 cm a maximální délka hlavní linky 40 m [9]. Tato omezení platí pro High-Speed CAN. Pokud stačí nižší rychlost přenosu, tak lze použít výrazně delší vodiče za cenu nižší rychlosti (40 metrů dosáhne 1 Mbps, 1 km dosáhne rychlosti pouze 50 kbps).



Obr. 3 Zapojení linky CAN

Maximální počet zařízení na jedné lince se liší podle více faktorů: CAN moduly jsou stavěné na různý maximální počet zařízení, takže se podle informací od výrobce musí přizpůsobit počet jednotek na lince. Zátěž jednotlivých jednotek na linku je spojena s jejich vstupní impedancí, takže opět záleží maximální počet jednotek na použitých výrobcích.

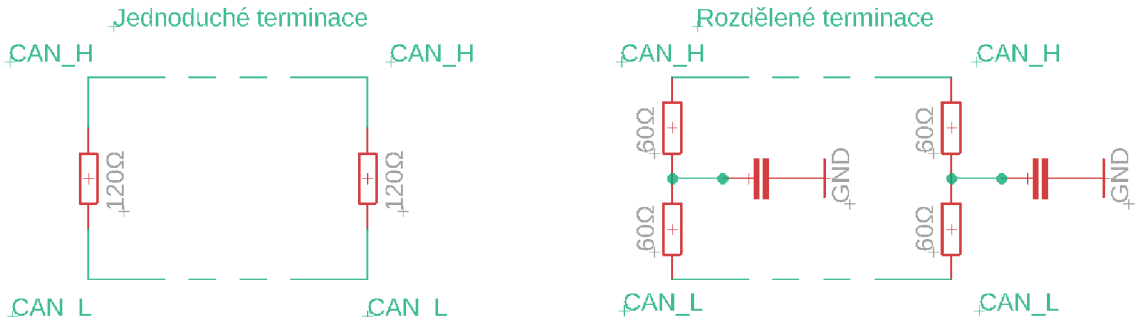
Teoreticky je možné dosáhnout 127 až 255 zařízení na lince, pokud je použit modifikovaný CAN (CANopen, CANKingdom bus), ale s vyšším počtem zařízení klesá maximální dosažitelná přenosová rychlost. Na lince blízko maximálního vytížení by docházelo ke zpoždění přenosu méně důležitých dat nebo jejich ztrátě, podrobněji v další kapitole.

1.2.2 Terminace

Vedení mají vlastnost značenou Charakteristická impedance, která závisí na fyzických vlastnostech použitých vodičů. Při přenosu signálu dochází na koncích vedení k odrazu signálu, který zhoršuje přenos dat. Tomuto odrazu lze zabránit použitím terminace o stejné impedanci, jako je charakteristická impedance vedení. Pro linku CAN se používá kroucená dvoulinka s charakteristickou impedancí kolem 120 Ω, takže se na oba konce linky připojí příslušná terminace. Používané typy terminace jsou rezistory 120 Ω nebo dělená terminace 2x 60 Ω se zemnícím kondenzátorem mezi nimi (na obrázku 3). Terminace v CAN zrychluje ustálení hodnot na sběrnici, jak ukazují hodnoty, které naměřil John Griffith [8].

Výsledky jeho měření doby ustálení signálu jsou následující:

Správná terminace (oba konce)	120 ns
Částečná terminace (jeden konec)	251 ns
Žádná terminace	>18 µs



Obr. 4 Možnosti terminace

Pro krátké linky je možné použití jen jedné terminace 60 Ω na lince, tato možnost není ale oficiálně podporována. Některá zařízení mají vestavěnou terminaci, kterou je nutné počítat při návrhu linky. Nesprávné provedení terminace vede ke snížení kvality signálu, v horších případech přestane fungovat komunikace na celé lince. Kontrolu lze provést změřením odporu pomocí multimetru, kde by se naměřená hodnota měla pohybovat kolem 60 Ω.

1.3 CAN – Přenosová vrstva

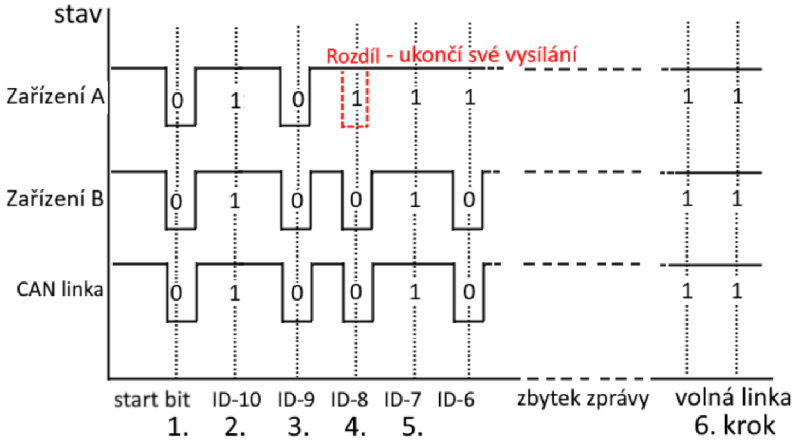
Tato vrstva zahrnuje definici podoby CAN zprávy, rychlost přenosu a časování, hledání a opravování chyb a proces arbitráže.

1.3.1 Arbitráž

Prioritu vysílání určuje arbitráž na základě identifikátoru vysílajícího zařízení. Datový rámec je vysvětlen v další sekci, pro arbitráž je důležité, že začíná Start of Frame bitem, po kterém následuje Arbitration field (pole pro identifikátor). V době, kdy je linka v klidovém stavu (žádné zařízení nevysílá, na lince stálá logická 1), může začít libovolné zařízení vysílat. Každé vysílání začne jedním *dominantním bitem* (log 0) a všechna zařízení na lince se synchronizují podle sestupné hrany a připraví se na příjem dat. V případě, že více než jedno zařízení začne vysílat ve stejný okamžik, nedojde v této fázi ke kolizi, protože všechny vysílají dominantní stav. Po startovním bitu začnou zařízení vysílat svůj identifikátor: 11 bitů pro standardní CAN a až 29 bitů pro rozšířený formát CAN. Bity v identifikátoru jsou značeny ID-10 až ID-0 a jsou vysílány v pořadí od nejdůležitějšího bitu (ID-10) po nejmíň důležitý bit (ID—0). Zařízení při vysílání sledují skutečný stav linky a při rozdílů mezi žádaným stavem a skutečným stavem okamžitě přestanou vysílat. Příklad vysílání v několika krocích:

1. Zařízení A vyšle startovní bit ve stejný okamžik jako zařízení B
   Zařízení A má Identifikátor 1011 1110 110
   Zařízení B má Identifikátor 1001 0000 110
2. První vyslaný bit je stejný, obě zařízení pokračují — Zařízení A - **1**011 1110 110; Zařízení B - **1**001 1110 110
3. Druhý vyslaný bit je stejný, obě zařízení pokračují — Zařízení A - 1**0**11 1110 110; Zařízení B - 1**0**01 1110 110
4. Třetí bit je *recesivní* u zařízení A, *dominantní* u B — Zařízení A - 10**1**1 1110 110; Zařízení B - 10**0**1 1110 110

   Zařízení A změří na lince logickou 0, ale očekávalo logickou 1 a okamžitě tedy přeruší vysílání
5. Zařízení B nyní vysílá samo, nenarazí tedy na žádný konflikt a odešle svou zprávu
6. Neprobíhá vysílání, linka je v klidovém stavu a zařízení A se znovu pokusí o vysílání



Obr. 5 Průběh arbitráže

Ve správně nastavené síti CAN odpovídají hodnoty identifikátorů důležitosti vysílané informace. V nejhorším případě by vznikla důležitá zpráva okamžik po obsazení linky jinou zprávou, ale arbitráž by garantovala odeslání zprávy s nejnižším identifikátorem hned v dalším datovém rámci. Výsledkem přibližného výpočtu zpoždění z maximální délky datového rámce ~15 bytů a přenosové rychlosti ~1 Mbps je zpoždění zhruba 0,1 milisekundy. Takové zpoždění je stále přijatelné i v kritických situacích, jako je například použití airbagů, které se musí správně nafouknout již během 10 ms od kolize pro krajní situace. [11]

1.3.2 Data Frame a Remote Frame

CAN používá několik rámců zpráv pro různé účely. První je popsán Data Frame a Remote Frame, protože tvoří většinu komunikace na lince. Skládají se z následujících částí:

1. **Start of Frame** – začátek rámce, jedná se o jeden dominantní bit, slouží k synchronizaci a obsazení linky

2. **Arbitration Field** – identifikátor vysílané zprávy, slouží k určení priority vysílání, ale také se jím řídí reakce ostatních jednotek. Na CAN lince přijímají vysílanou zprávu všechna zařízení, ale jestli data bude jednotka ukládat a zpracovávat je rozhodnuto podle identifikátoru zprávy. Identifikátor s délkou 11 bitů umožňuje hodnoty 0 až 2047, ale tato hodnota není adresa cílového či odesílajícího zařízení, ale přímo ID zprávy. Všechna zařízení musí obsahovat stejné definice identifikátorů, jinak komunikace nebude fungovat správně.

3. **RTR – *Remote Transmission Request*** je jeden bit, který značí, jestli je tento rámec zpráva s daty (logická 0) nebo žádost o data (logická 1). Jedná-li se o žádost, tak Data field bude prázdný.

4. **Control Field** – obsahuje 6 bitů. První bit je *Identifier Extension bit (IDE)* a značí použitý formát identifikátoru. Pokud je IDE logická 0, je použit 11-bitový identifikátor a platí tento seznam částí Data Frame. Pro použití 29-bitového identifikátoru je nastaven IDE na logickou 1 a hned po něm následuje zbylých 18 bitů identifikátoru a až potom následuje zbytek bitů Control field. Druhý bit je momentálně nevyužitý a rezervovaný na budoucí funkce, značen r0. V rozšířeném formátu CAN je navíc rezervovaný bit r1, pořadí je potom IDE r1 r0… . Poslední 4 bity Control field jsou *Data Length Code (DLC)*, které značí počet datových bytů v Data Field. Platné jsou jen hodnoty od 0 do 8 i přes možnost čísel až do 16 pomocí 4 bitů v DLC.

5. **Data Field** – jedná se o samotná data, která zařízení odesílá. Maximálně 8 bytů/64 bitů, byty jsou posílány od nejvýznamnějšího bitu. Není určen povinný formát, záleží na tvůrci programu, jak s daty bude zacházet.

6. **CRC Field** – *Cyclical Redundancy Check* je kontrolní součet pro ověření správnosti přijatých dat. V CAN se jedná o sekvenci 15 bitů a jsou v ní započítána celá předchozí část rámce (Start of Frame až Data Field). Na konci je vždy jedna logická 1 jako šestnáctý bit, značí konec CRC.

7. **ACK Field** – pole pro potvrzení správnosti zprávy a bezproblémového přijetí od ostatních zařízení na lince. Odesílající zařízení jako poslední bit v CRC field odeslalo logickou 1, první bit ACK pole také pošle logickou 1 (recesivní bit) a konec ACK field odešle třetí logickou 1. Ostatní zařízení na lince přepíšou místo prvního bitu logickou 0 (dominantní bit) a tím potvrdí správnost CRC. První bit ACK pole se nazývá *ACK slot*, druhý bit značí konec ACK field.

8. **End of Frame** – Zařízení odvysílá sedm recesivních bitů jako konec rámce.

1.3.3 Error Frame a Overload Frame

Pro pochopení zbylých dvou rámců je nutné vysvětlení metody *bit stuffing*. CAN nepoužívá hodinový signál, takže je synchronizaci udržována podle změn hladin vysílaných zpráv. Tato synchronizace by mohla selhat v případě, kdy je odesíláno vícero stejných bitů po sobě, protože na signálu pak nejsou hrany, kterých se může synchronizace držet. Tento problém řeší *bit stuffing*, což je možné přeložit zhruba jako bitová vycpávka, nebo vyplňování bity. Pokud vysílač zprávy odeslal pět stejných bitů po sobě, tak automaticky odešle jeden bit opačné polarity, který přijímající zařízení nepovažují za data. Není-li tento bit vložen, nebo nesouhlasí vypočítané CRC a přijaté CRC, tak nastala v přenosu chyba, což povede k odeslání *Error Frame* a datový rámec bude znovu odeslán. *Bit stuffing* neprobíhá v End of Frame (7 recesivních bitů na konci rámců). Po každém rámci následuje *Intermission*, což jsou 3 recesivní bity, kdy je povoleno jen zahájení vysílání *Overload Frame.*

Zbylé dva druhy rámců jsou tedy Error Frame a Overload Frame. Error Frame začnou vysílat jednotky, které zaznamenaly chybu ve zprávě. První část Error Frame je *Active error flag* – 6 dominantních bitů v řadě. Následuje 6 recesivních bitů. Tyto bity slouží pro přepsání jinými jednotkami, které chybu detekovaly o 1-6 bitů později. Error Frame schválně vynechává *bit stuffing,* aby všechny jednotky zaznamenaly chybu přenosu i pokud jim originální chyba unikla. Chyba je ukončena 8 recesivními bity, po kterých hned následuje 7 recesivních bitů značených Interframe Space. Nastane stav volné linky a všechna zařízení odesílající zprávu vstoupí do procesu arbitráže, včetně zařízení, které odeslalo zprávu s chybou. Díky tomu neobsadí zprávy s chybou v odeslání linku důležitějším zprávám s prioritou v arbitráži.

Overload Frame funguje podobným způsobem, jeho vysílání je ale způsobeno zahlcením některé z přijímajících jednotek a slouží k získání času na zpracování přijatých dat z minulého Data Frame. Narozdíl od Error Frame nemůže být jeho vysílání zahájeno kdykoliv, ale pouze na konci minulého rámce libovolného typu. Výjimkou je Overload Frame, protože smí být maximálně dva v řadě, aby nemohla vadná jednotka pozdržet linku na dobu neurčitou. Druhý způsob vyvolání Overload Frame je zaznamenání dominantního stavu příliš brzy po konci jiného rámce. Konkrétně se jedná o detekci dominantního bitu během konce rámce (7 recesivních bitů), po kterých musí následovat 3 recesivní bity (*Intermission*), nebo 8+7+3 recesivních bitů po Error Frame.
Všechny stavy, které vyvolají chybu na lince jsou:
Selhání vysílání – na lince zdroj vysílání měří jinou hodnotu, než vysílá (neplatí u arbitráže)
Selhání CRC – přijímač provedl kontrolní součet a zjistil chybu
Selhání ACK – příjem zprávy nebyl potvrzen v ACK Field
Kontrola rámce – některé bity mají být vždy pouze dominantní (Start of Frame) nebo pouze recesivní (konec CRC, konec ACK Field a celé End of Frame). Nesprávná hodnota vyvolá chybu
Selhání Bit stuffing – více než 5 stejných bitů v řadě (povoleno v End of Frame)

1.3.4 CAN FD – CAN with Flexible Data-rate

S postupným zvyšováním množství řídících jednotek a senzorů ve vozech se také zvyšovalo množství dat k přenosu. Inženýři firmy Bosch zavedli CAN FD na trh v roce 2012, aby překonali limit 1 Mbps u klasického CAN, jak píšou na svých stránkách [12]. Téměř všechny informace, které jsem zmínil v předchozí sekci, platí i pro CAN FD, protože jsou mezi sebou kompatibilní verze CAN přinejmenším od 1991 dodnes jen s drobnými omezeními (omezení v identifikátorech, maximální rychlost). Několik změn se nachází v Data Frame. Hlavní vlastností CAN FD je již z názvu vyplývající flexibilní změna rychlosti přenosu. Pro zachování kompatibility se staršími zařízeními probíhá začátek komunikace původní maximální rychlostí 1 Mbps, ale pokud mezi sebou komunikují CAN FD schopná zařízení, tak probíhá většina přenosu rámce na vyšší rychlosti až do ACK Field a End Of Frame. Rychlost přenosu zvyšuje také nový limit 64 bytů v Data Field místo 8 bytů.

První změny jsou v Control Field. Na místě, kde byl původně rezervovaný bit následující po *Identifier Extension* bitu, je nyní *Extended Data Length* bit (pro 29bitový identifikátor následuje až po *Remote Transmission Request* bitu). Tento nový EDL bit značí v logické 1 použití nového formátu Data Field a přijímající jednotky se přepnou na dekódování CAN FD, jinak je použit původní formát CAN. V CAN FD režimu se změní detaily většiny dalších částí Data Frame, základní struktura je ale stejná. Po recesivním EDL bitu následují dva rezervované dominantní bity r1 a r0, po nich následuje *Bit Rate Switch (BRS)*. Tento bit značí v recesivním stavu přechod na vyšší přenosovou rychlost. Dalším novým bitem je *Error State Indicator*, který slouží pro diagnostiku při monitorování sítě. Vysílající jednotky, které detekují nějakou interní chybu, nastaví ESI bit na dominantní v případě chyby. Control Field je opět zakončen *Data Length Code*, kde hodnoty 4bitového binárního kódu 0000-1000 (0-8) stále značí 0-8 bytů v Data Field. Původně nevyužité hodnoty 1001-1111 (9-15) značí délku Data Field od 12 bytů do 64 bytů podle tabulky.

Data Length Code	Byty v Data Field
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8

Data Length Code	Byty v Data Field
1001	12
1010	16
1011	20
1100	24
1101	32
1110	48
1111	64

Obr. 6 Tabulka Data Length Code

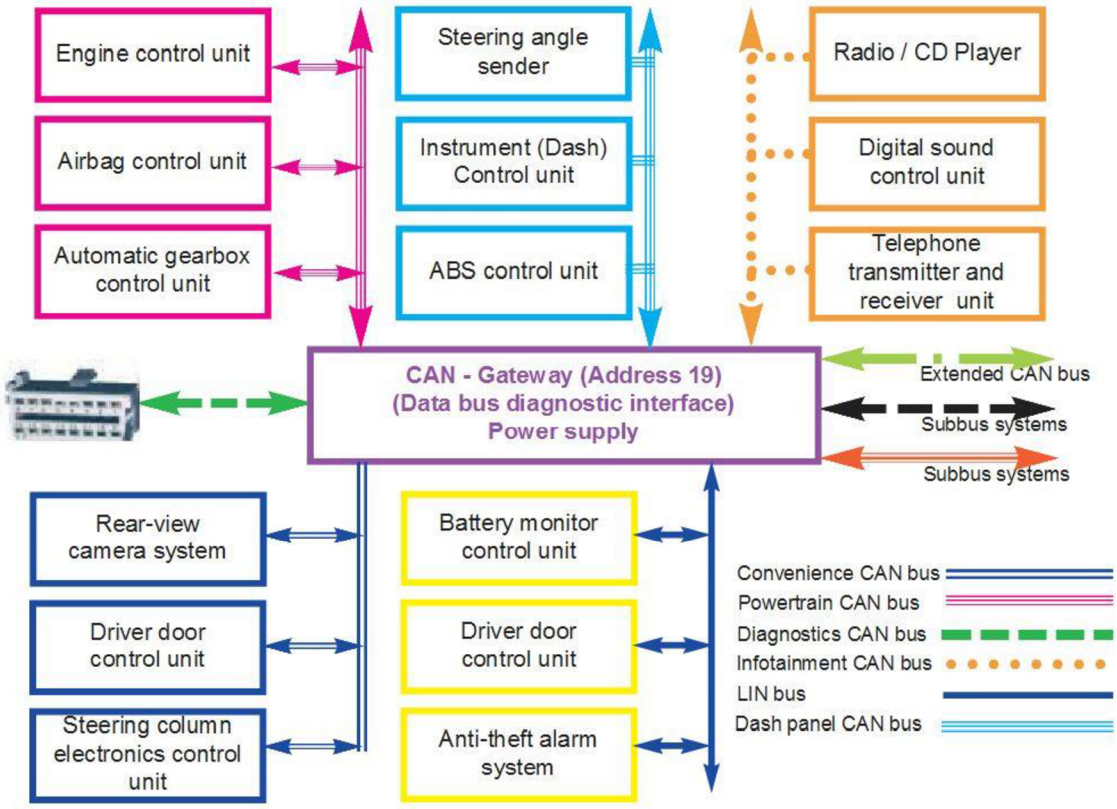
Po přenosu Data Field následuje *Cyclical Redundancy Check Field*, které je v CAN FD prodloužené na 17bitovou CRC sekvenci. Její větší délka umožňuje spolehlivější kontrolu přenosu zvětšeného rámce. Na konci CRC Field je jeden recesivní bit, který zároveň značí přepnutí na původní pomalejší rychlost přenosu. Zbytek Data Frame je stejný, tedy následuje ACK Field a End Of Frame.

Otázka, o kolik vyšší přenosovou rychlost CAN FD dosahuje pomocí *Bit Rate Switching* a delšího Data Field, nemá jednoznačnou odpověď. Rychlejší přenosová rychlost je stále limitována fyzickým provedením linky, ale Bosch uvádí v ideálních podmínkách naměřených 15 Mbps, se staršími jednotkami v reálných podmínkách se uvádí hodnoty 2 Mbps až 5 Mbps dle výrobce [13]. Výsledná rychlost přenosu záleží také na počtu bytů, které vyšší rychlostí přeneseme. Drobným výpočtem lze porovnat dobu přenosu stejných 64 bytů pomocí Classical CAN a CAN FD.

Pro přenos 64 bytů dat potřebuje Classical CAN 8 Data Frame, to se rovná přibližně 380 bitů pro rámce a 512 bitů dat. Při 1 Mbps přenos zabere přibližně 850 mikrosekund. CAN FD potřebuje jen 52 bitů pro 1 rámec při rychlosti 1 Mbps a 512 bitů dat při rychlosti 5 Mbps. Přenos zabere jen 150 mikrosekund. Stejné množství dat přenese CAN FD tedy více než pětinásobnou rychlostí v ideálních podmínkách. Nevyužije-li se vyšší limit 64 bytů v jednom rámci a bude přenos dat probíhat po osmi bytech v osmi rámcích, protože posílat větší zprávy v systému není potřebné nebo možné, tak se výpočet změní následujícím způsobem. CAN FD bude potřebovat 416 bitů pro rámce rychlostí 1 Mbps a 512 bitů rychlostí 5 Mbps. Tento nejhorší případ bude trvat zhruba 500 mikrosekund. Z výsledku je zřejmé, že CAN FD je rychlejší než Classical CAN i ve špatných podmínkách přibližně o 70 %. Ve výpočtech je vynechán *bit stuffing*, protože není konstantní, ale jeho vliv by nebyl významný.

1.3.5 CAN v automobilu

Ve vozech se jednotky dělí do různých linek CAN podle jejich účelů. Na jedné lince budou jednotky pro motor a řazení, nebudou ale sdílet zdroje s rádiem, GPS a klimatizací, protože spolu tyto systémy nikdy nepotřebují přímé spojení. Jednotlivé CAN linky jsou spojeny na společné jednotce, jejíž účelem je přeposílat zprávy do jiných CAN linek, nebo překládat zprávy pro jiné komunikační protokoly. Jako příklad takového rozdělení uvedu starší analýzu vozů firem ve Volkswagen Group z roku 2016 [10]. Na obrázku je rozkreslené zjednodušené blokové schéma systémů využívajících CAN a jejich dělení podle využití. V tomto příkladě je v jednom voze 5 různých CAN linek, počet se ale liší podle konkrétních potřeb vozu, protože pro různé funkce může být vhodnější například LIN, protože je levnější než CAN. Pro situace, kde je potřeba posílat větší množství dat, lze použít například MOST (Media Oriented Systems Transport), USB (Universal Serial Bus), BlueTooth nebo Ethernet. (připojení k internetu, audio a video, spojení vozu s mobilním telefonem…)



Obr. 7 Linky CAN ve voze [10]

1.3.6 Diagnostika přes CAN

Pro diagnostiku aut slouží standardizovaný systém OBD-II (On-Board Diagnostics). Jeho předchůdci a starší verze se v automobilovém průmyslu v USA začali vyskytovat už od roku 1980 a verze OBD-II je v Evropě povinná přibližně od roku 2001, později v dalších oblastech. Tento systém zajišťuje jednotný přístup k diagnostice závad na automobilech, včetně stejného konektoru a obecných diagnostických kódů, které jsou identické pro všechny výrobce.

Pro spojení diagnostických nástrojů s vozem je možné použít 5 protokolů, mezi které patří CAN. OBD-II CAN komunikace probíhá specifickým způsobem pomocí předem definované diagnostické zprávy. Nejprve diagnostický přístroj vyšle CAN zprávu s identifikátorem 7DFh (h značí hexadecimální soustavu) a datové pole obsahuje kódy dat, která jsou z vozu žádána. Je standardní definice kódů, ale výrobci vozů mohou definovat druhý typ diagnostické zprávy, kde lze poskytovat data a funkce nad rámec OBD-II. Na tuto žádost o data odpoví ta jednotka, která data obsahuje (na dotazy ke stavu odpovídá jednotka motoru a podobně). Identifikátor zprávy neznačí její obsah, ale jednotku, která ji vysílá. Datové pole má vždy délku 8 bytů, první byte obsahuje počet platných následujících bytů, dva byty obsahují kód služby (aktuální data, uložená data, chybové kódy a další) a kód parametru (značí se PID). V dalších bytech je hodnota parametru, přebytečné byty jsou prázdné.

Například proces zjištění spotřeby paliva proběhne následujícím způsobem. V dokumentaci standardních PID je ve službě *01 Show current data* uvedeno *Engine fuel rate* pod PID 5Eh. Požadavek tedy bude složen z diagnostického CAN ID 7DFh, počtu 2 bytů, služby 1 a PID 5Eh. V odpovědi se CAN ID bude lišit podle jednotky, která data posílá, počet bytů značí množství bytů s platnými daty, služba bude stejná s přičteným 40h, PID bude stejné. Hodnota je zapsána ve dvou bytech, kde Byte 3 má vyšší váhu. Po převodu hodnoty do desítkové soustavy stačí dosadit do vzorce z tabulky a v příkladu je výsledkem spotřeba 6,7 l/h.

Požadavek o data

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Počet bytů	Služba	Kód PID	-	-	-	-	-
7DF	2	1	5E	0	0	0	0	0

Odpověď

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Počet bytů	Služba	Kód PID	Hodnota	Hodnota	-	-	-
7E8	4	41	5E	0	86	0	0	0
				00000000	10000110			

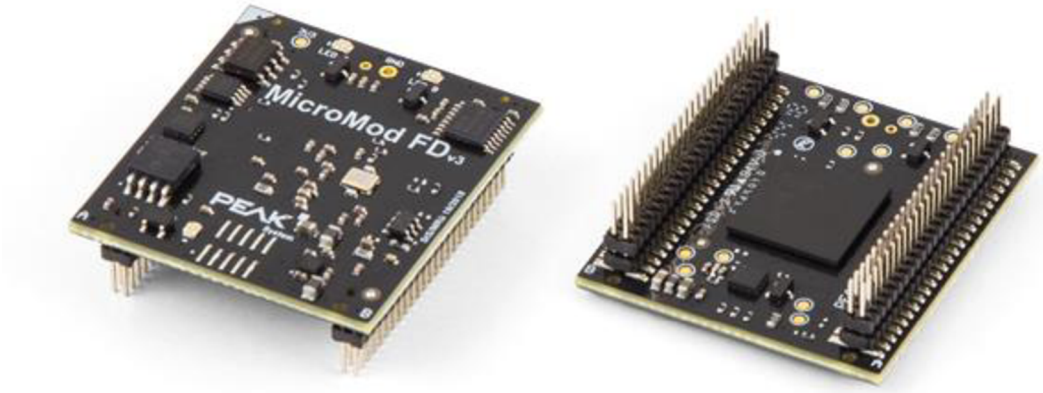
Spotřeba = 0 + 0.05 * Hodnota = 0.05 * (86)hex = 0.05 * 134 = 6.7 l/H

Obr. 8 Diagnostické zprávy CAN

1.4 Hardware

1.4.1 PCAN-MicroMod FD

Středem této práce je modul firmy PEAK-System PCAN-MicroMod FD [14]. Jedná se o relativně malý modul pro tvorbu vlastních projektů s připojením CAN a CAN FD, zároveň má více konfigurovatelných vstupů a výstupů pro připojení senzorů a akčních členů. Základem tohoto modulu je mikrokontroler NXP LPC54618 [15].



Obr. 9 PCAN-MicroMod FD [14]

Relevantní technické údaje:

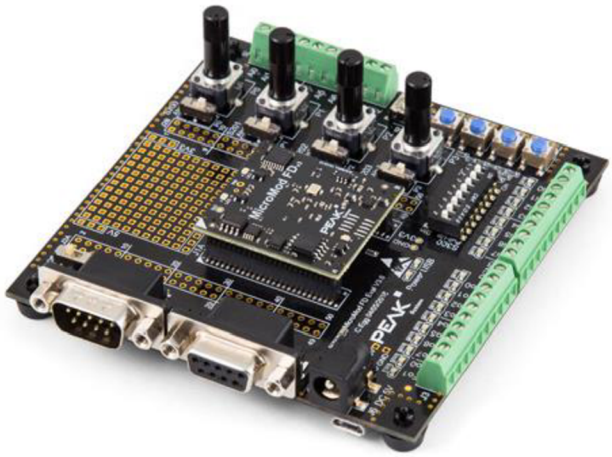
rozměry	33 x 36 mm
napájení	3.3 V
podpora CAN	CAN specifikace 2.0 A/B, CAN FD

8 digitálních vstupů, 8 digitálních výstupů

Modul je potřebné zasadit do jiné desky pomocí dvou 50pinových lišt, doplnit napájecí a další pomocné obvody a konektory pro žádaná rozhraní. Dále je potřebné na Windows (10, 8.1) počítači vytvořit a nahrát konfigurační soubor pomocí dodaného konfiguračního software od výrobce. Tento soubor definuje chování modulu a jeho CAN komunikaci.

1.4.2 Evaluation board

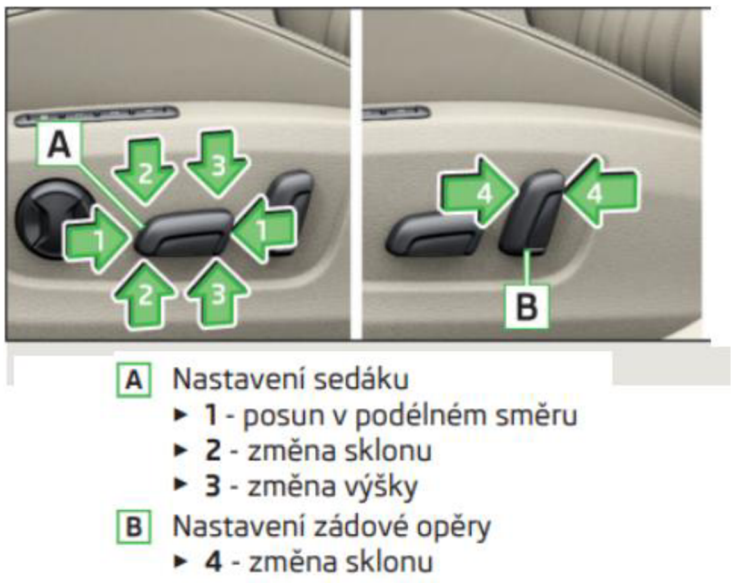
Tento produkt má sloužit k podpoře vývoje vlastních projektů na základě modulu PCAN-MicroMod FD, takže se jedná o poměrně velkou desku přibližně 10 x 10 cm s ukázkami funkcí, které výrobce považuje za užitečné. Tato deska také má všechny potřebné napájecí a podpůrné obvody pro používání modulu. Tento produkt nebyl zakoupen, ale byly využity detailní nákresy a schémata tohoto modulu jako základ schématu a plošného spoje v této práci. Konkrétně byly hlavní zdroje informací dokumenty *PCAN-MicroMod FD Evaluation Board manual [16]* a *PCAN-MicroMod FD manual [17]*.



Obr. 10 Evaluation Board [18]

1.4.3 Sedačka

V luxusnějších moderních vozech jsou elektrické sedačky na místě řidiče a spolujezdce běžnou výbavou. Jako příklad je uvedena sedačka řidiče z vozu ŠKODA Superb 3. Sedačka je rozdělená na sedák a zádovou opěru. K sedáku patří tři prvky ovládání: nastavení výšky, nastavení sklonu a posun směrem k volantu a zpět. Změna sklonu opěry zad má jen jeden ovládací prvek, ale existují sedačky s prvkem nastavení bederní opěry, to ale v mém případě neplatí.



Obr. 11 Ilustrace ovládání sedačky ŠKODA Superb 3 [19]

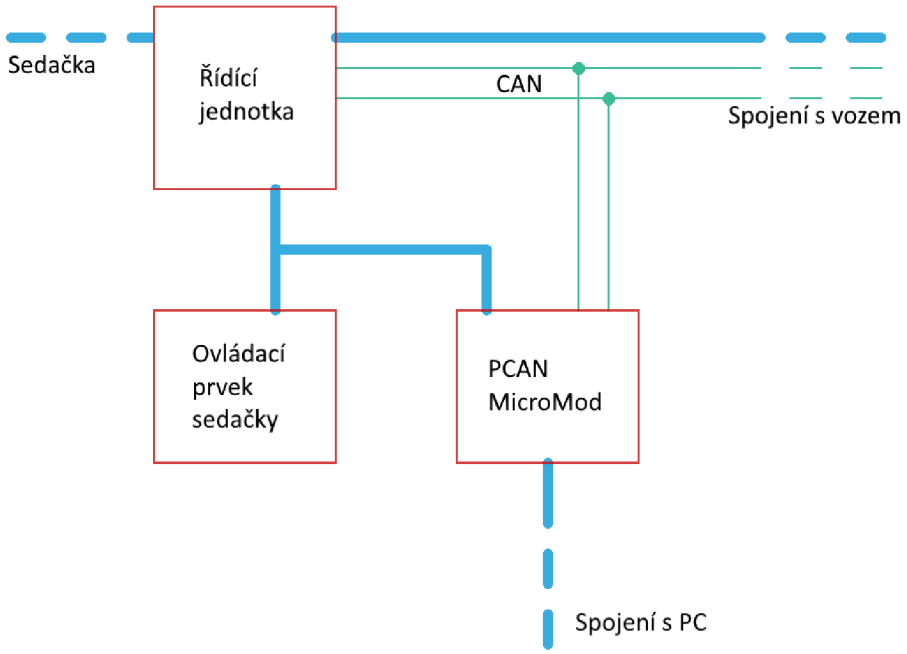
Aby bylo možné sedačku ovládat, je nutné proměřit hodnoty korespondující s jednotlivými nastaveními sedačky a jejich směry. V technologickém centru byla k dispozici funkční sedačka, její ovládací prvky a řídící jednotka. Popis je rozdělen na tři části: řídící jednotka, ovládací prvek a zapojení.

Řídící jednotka plní více funkcí než jen ovládání sedačky, takže je spojena s mnoha vodiči. Pro účely této práce je nutné připojení na CAN, napájení 12 V a spojení s mechanickým ovladačem sedačky. Na lince CAN bude PCAN-MicroMod FD přijímat zprávy pro dálkové řízení sedačky a přes diagnostické zprávy lze číst různé informace spojené se sedačkou. Poloha sedačky je měřena po impulzech, které řídící jednotka zpracovává na absolutní polohu. Další měřené údaje jsou například proud procházející elektromotory a jejich teplota pro ochranu před přehřátím.

Mechanické ovládací prvky jsou dva, jeden pro všechny tři funkce sedáku, druhý pro sklon opěry zad. Ovládání nastavení sklonu má tři stavy:
neutrální/střední poloha – sklon sedačky se nemění
krajní poloha směrem k volantu – opěra zad sedačky se naklání směrem k volantu
krajní poloha směrem k zadním sedačkám – opěra zad se naklání směrem dozadu

Ovládání sedáku je složitější, protože ovládá tři různá nastavení, každé dvěma směry. Je možné nezávisle ovládat až dva směry najednou (například sunout sedačku dopředu a měnit sklon). Sedačka zastaví před fyzickým koncem závitové tyče pomocí softwarového omezení. Žádný z ovládacích prvků neumožňuje jemné ovládání, kde by částečný posun ovládacího prvku vedl k pomalejšímu pohybu sedačky. Podle měření je principem fungování těchto ovládacích prvků proudová smyčka. Jednotlivé signály lze nahradit rezistorem správné hodnoty mezi určitými vodiči a tím ovládat sedačku.

Zapojení těchto prvků ve voze a připojení se k nim je poměrně jednoduché, protože na CAN i proudovou smyčku bude v tomto případě stačit připojení *T* bez dalších prvků. K tomu je nutné vyrobit malý kabelový svazek.



Obr. 12 Zapojení

2 Praktická část

2.1 Použitý hardware a software

Hardware:

- PC s Windows 10 64bit, potřebný pro PCAN konfigurační program
- PCAN-MicroMod FD
- Automobilová sedačka s elektronickým ovládáním
- Multimetr Fluke 289
- PCAN-USB IPEH-002021

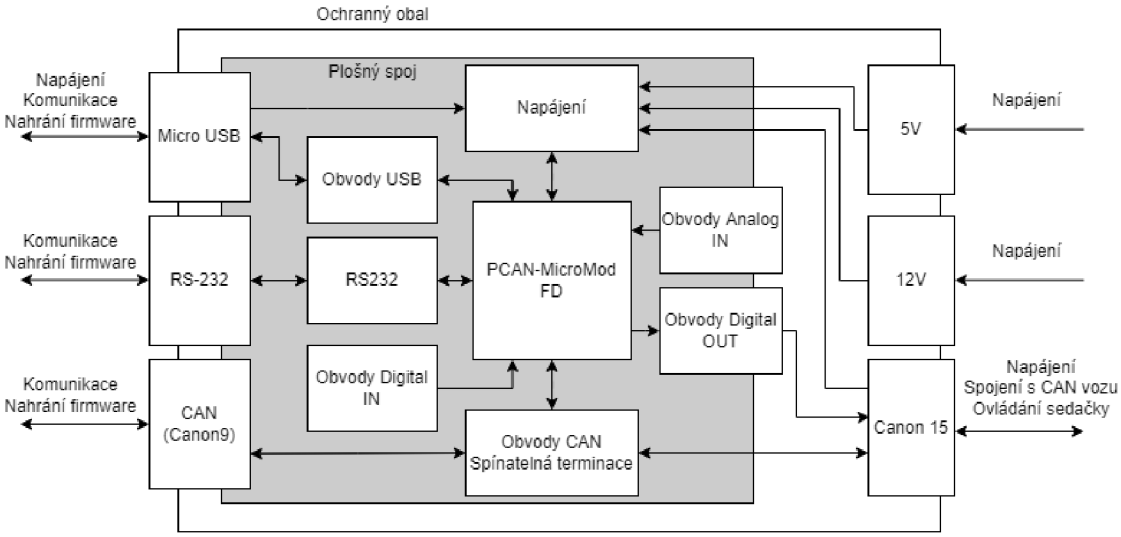
Software:

- PCAN-MicroMod FD Configuration Version 1.1.2.368
- PCAN-View Version 5.0.2.838
- PCAN-Basic V4.6.0.600
- EAGLE 9.6.2.
- Microsoft Visual Studio Community 2019 Version 16.11.13

2.2 Plošný spoj

PCAN-MicroMod FD potřebuje podpůrné obvody, jak bylo zmíněno při jeho popisu. Zbylé funkce plošného spoje jsou podle požadavků zadavatele. Základem navrženého plošného spoje je schéma *PCAN-MicroMod FD Evaluation Board.*

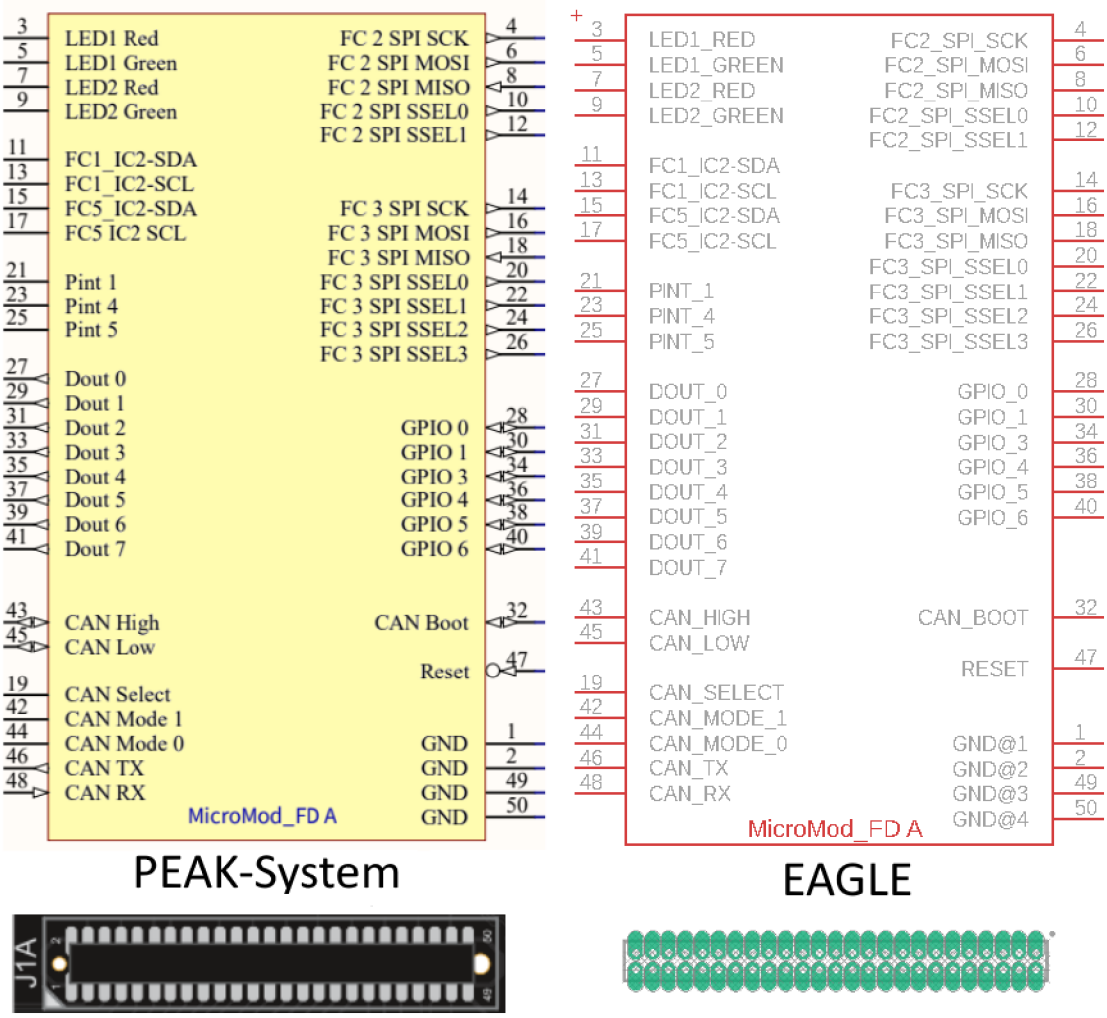
2.3 Schéma



Obr. 13 Blokové schéma praktické části

Výsledkem praktické části je krabička chránící plošný spoj s modulem PCAN-MicroMod FD. Pro spojení s počítačem, automobilem a případným externím zdrojem je použito 6 konektorů, ale v praktickém zapojení jsou využity podle potřeby jen 2-3. Nejběžnější zapojení je připojení PC zleva na CAN a vpravo napojení kabelového svazku z vozu na Canon 15. Takové zapojení umožňuje komunikaci s modulem i s vozem přes CAN, plošný spoj je napájený z autobaterie a sedačka je připojená na výstupní obvody plošného spoje. V této práci jsou nevyužité digitální a analogové vstupy, jsou ale přítomné pro případné rozšíření funkcí ve vývoji (senzory, spínače).

Začátkem návrhu bylo důkladné prostudování schématu Evaluation Board firmy PEAK-System, protože obsahovalo velkou část potřebných obvodů. Po přečtení datových listů všech použitých integrovaných obvodů byl zahájen proces přetváření schématu po sekcích do návrhového software EAGLE. Byla založena knihovna potřebných součástek. Modul PCAN-MicroMod FD byl převeden první, jelikož je středem celého projektu. Modul PCAN-Micromod FD má 100 pinů ve dvou 50pinových lištách, značení bylo dodržené jako u výrobce.



Obr. 14 Originální symbol v dokumentaci a odpovídající symbol v EAGLE

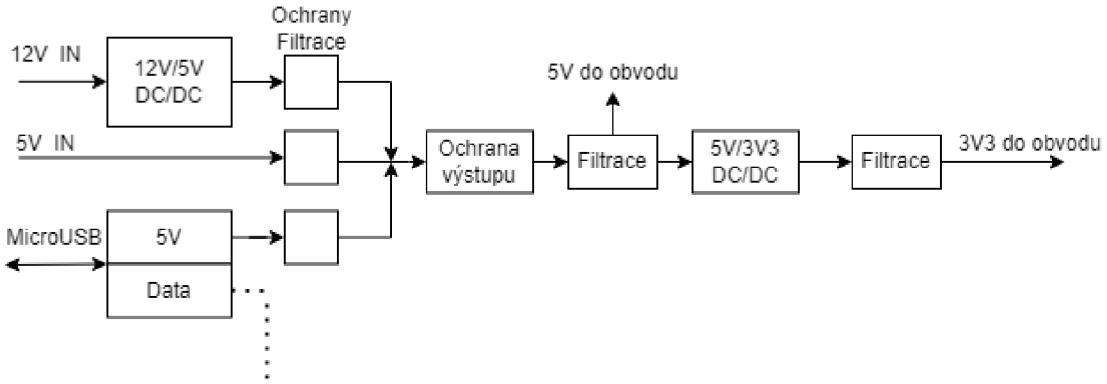
Schémata PEAK-Systém pochází podle vzhledu symbolů pravděpodobně z návrhového programu Altium Designer. Ten má proti EAGLE možnosti navíc, jako použití mezer v názvech součástek a pinů, barevné bloky a sdílené názvy pro piny jako GND. Tyto rozdíly vedou dle mého mínění k přehlednějším, profesionálněji vyhlížejícím symbolům a schématům. K symbolům byl na obrázek přidán *footprint* pinové lišty. PEAK-System používá Surface Mount Technology pinovou lištu, která nebyla v době objednání součástek dostupná. Narušení výroby a dodávky součástek z důvodu pandemie Covid-19 zasáhlo tuto část projektu nejvíce, protože bylo nutné hledat náhradu za mnoho součástek. Změna pinové lišty z SMT na Through Hole Technology později vedla k větším problémům při návrhu plošného spoje.

Celé schéma by bylo bez použití více listů a náhradních symbolů nepřehledné, takže bylo použito několik modulů s vlastními schématy. Práce s moduly v EAGLE byla v několika částech návrhu problematická.

Některé následující obrázky jsou blokové reprezentace nebo mírně upravené části celého schématu, které bude v příloze. Úpravy jsou převážně odstranění jmen a nadbytečných vodičů, které nesouvisí s funkcí popisované části schématu, a posunutí částí obvodu pro lepší viditelnost.

2.3.1 Napájení

PCAN-MicroMod FD vyžaduje napájení 3,3 V se spotřebou pod 200 mA. Napájení obvodu je možné z MicroUSB (5 V), přímého připojení 5 V nebo připojení 12 V z autobaterie, převedených na 5 V. Pro zábranu zničení při připojení několika různých zdrojů najednou jsou použity Schottkyho diody s odolností 30 V v závěrném směru.

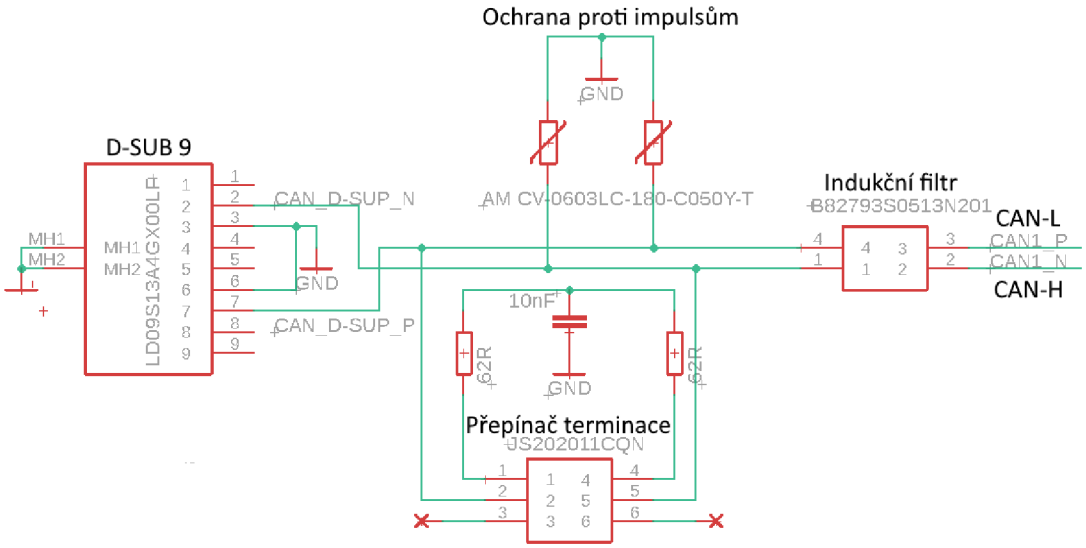


Obr. 15 Schéma napájení

Napětí 12 V z autobaterie je převedeno pomocí integrovaného PWM DC/DC konvertoru s filtrem na výstupu. Všechny tři možnosti napájení se spojí na společný vodič s filtrem proti vyšším frekvencím (100+ MHz). Proti zpětnému toku energie jsou použité blokovací diody s malou ztrátou napětí. Napěťový vodič z MicroUSB vstupuje do napájecí části obvodu a datové vodiče vedou na vlastní ochranu a filtraci.

Napájení je připojeno na ochranný integrovaný obvod, jehož účelem je odpojit výstup při přepětí nebo podpětí. Doba odpojení je přibližně 1 mikrosekunda. Tento typ ochrany je výrobcem zamýšlen pro odpojení USB zařízení při špatných podmínkách na napájecím vodiči, lze jej ale použít pro ochranu obvodů, kde napětí nesmí překročit přibližně 6 V. Bezpečné napětí projde dalším vyhlazením s kondenzátory a je převedeno pomocí PWM DC/DC step down regulátoru na 3,3 V. Spínací frekvence je poměrně vysokých 2,5 MHz a výsledné napětí vyhlazuje induktor a kondenzátor. Výsledných 3,3 V napájí většinu integrovaných obvodů na plošném spoji včetně modulu PCAN-MicroMod FD. V napájecí části je také spojení stínění s GND. Stínění je spojeno s několika pouzdry konektorů a nepřímo s AGND (zem pro analogové vstupy).

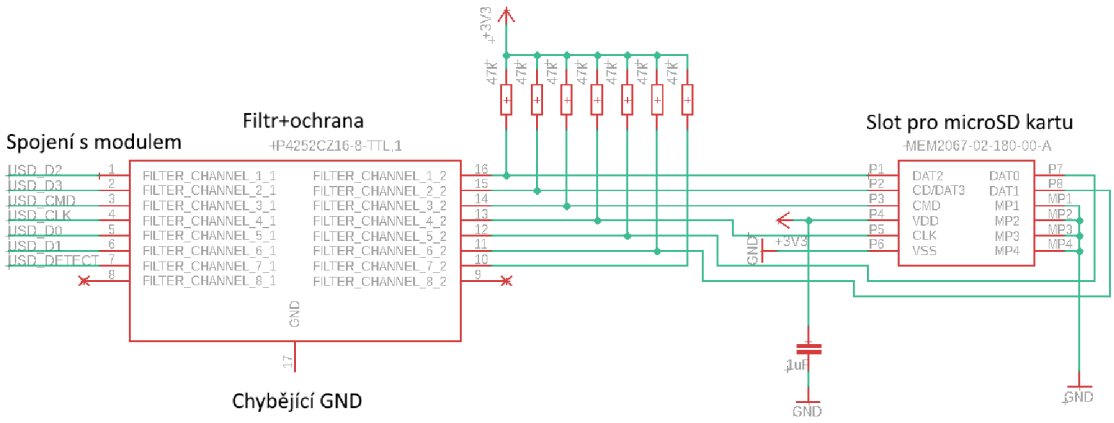
2.3.2 Komunikační obvody



Obr. 16 Schéma CAN

Schéma CAN obsahuje poměrně málo částí. Konektor D-Sub 9 (Canon 9) slouží pro připojení k lince CAN ve vozu. Ochranu proti impulsům zajišťují varistory, které svedou přepěťový impuls na GND. Indukční filtr se společným jádrem pro CAN-H a CAN-L slouží k potlačení souhlasného elektromagnetického rušení. Anglicky se tento typ součástky nazývá *Common mode choke.* Terminace se na lince CAN mají nacházet vždy právě dvě, takže je užitečná možnost terminaci vypnout/zapnout, k čemuž slouží mechanický dvoupolohový spínač. Drobná poznámka k názvům vodičů: v EAGLE se diferenciální páry vodičů musí jmenovat identicky s "_P" a "_N" na konci názvu, nebylo tedy možné použití jmen CAN-H a CAN-L.

Další možnosti komunikace mají jednoduchá schémata. RS-232 je vyvedeno na příslušné piny konektroru (zde D-SUB 9) s drobnými filtrovacími kondenzátory. Sběrnice I2C (Inter-Integrated Circuit) potřebuje pull-up rezistory. Jedna linka I2C je použita pro komunikaci s externí pamětí, druhá je vyvedena na konektor pro případné rozšíření nebo komunikaci s jiným mikrokontrolerem. Další diferenciální pár vodičů je USB, které má ochranu proti ESD (*ElectroStatic Discharge)* a další indukční filtr typu *Common mode choke.*

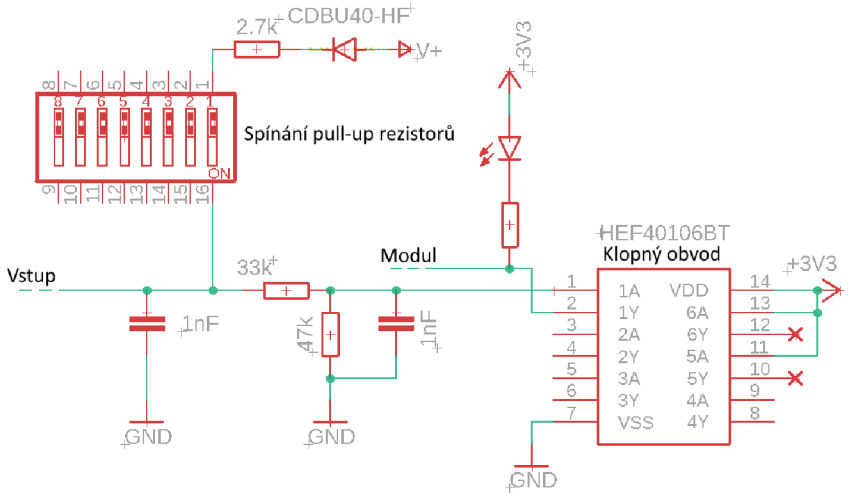


Obr. 17 Schéma micro SD karty

Funkci velkokapacitního úložiště může plnit microSD paměťová karta. Filtr pro spojení s mikroSD kartou obsahuje také vestavěnou ESD ochranu, která má svést náboj na GND. Ve schématu chybí jedno spojení s GND, bohužel byla chyba ve schématu odhalena až po výrobě plošných spojů. Kovový slot pro mikroSD kartu je přímo spojen s GND na několika místech, takže je zbytek obvodu částečně chráněn při práci s microSD kartou.

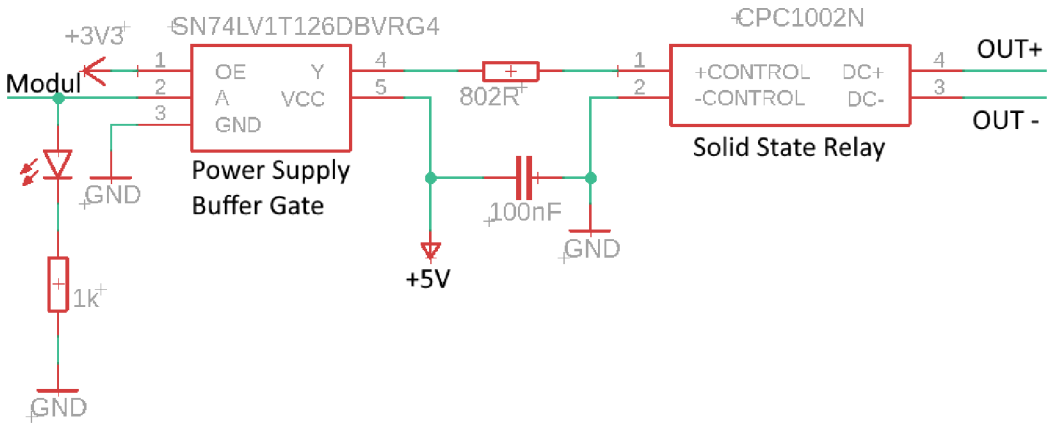
2.3.3 Vstupy a výstupy

Analogové vstupy se skládají z děliče napětí a ochranných Schottkyho diod, které odvádí napěťové špičky na zdroj a AGND. Analogové vstupy lze použít například na měření absolutní polohy pomocí vhodného typu senzoru, měření teploty a snímání hodnoty z ovládacích prvků typu potenciometr.



Obr. 18 Schéma digitálních vstupů

Na obrázku je pro přehlednost jenom jeden z osmi digitálních vstupů. Digitální vstup je od modulu oddělen rezistorem a invertujícím klopným obvodem s hysterezí. Takový obvod plní funkci filtru a zároveň upravuje pomaleji měnící se signály na obdélníkové. Každý digitální vstup je opatřen spínatelným pull-up rezistorem. Kromě otestování funkčnosti vstupů je lze použít na ovládání proměnných v programu. Funkce USB BOOT, SERIAL BOOT, CAN BOOT a RESET se obsluhují obyčejnými tlačítky s pull-up rezistory.



Obr. 19 Schéma digitálních výstupů

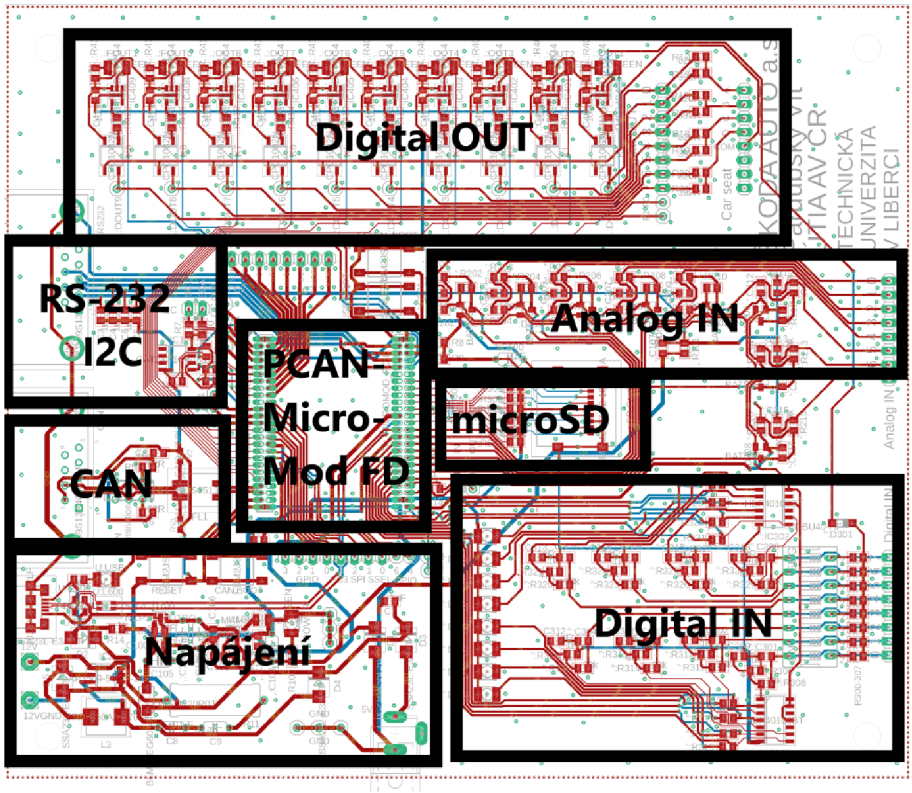
Digitální výstupy slouží ke spínání proudových smyček pro ovládání sedačky. Pro ochranu 3,3 V modulu od 12 V ve voze je navrženo galvanické oddělení pomocí Solid State Relay, takže jediné nevyhnutelné spojení s vozem, je společná zem linky CAN. Zařízení na lince CAN potřebují společnou zem, kterou je kostra auta na negativním pólu autobaterie. První částí obvodu je SN74LV1T126DBVRG4, který jeho datasheet nazývá *„Single Power Supply Single Buffer Gate with 3-State Output CMOS Logic Level Shifter“*. V tomto obvodu plní funkci převodu 3,3 V logiky na 5 V logiku a následně z 5 V zdroje napájí ovládací diodu v SSR, takže není zatížen výstup modulu PCAN-MicroMod FD. Použité SSR dosahuje podle datasheetu v sepnutém stavu odpor pouze 0,55 Ω, takže ovlivní hodnotu napěťové smyčky jen zanedbatelným způsobem, zároveň je schopné spínat až 700 mA a průraz by nastal až nad 60 V, což v automobilovém prostředí stačí. Některé nevyužité IO modulu PCAN-MicroMod FD jsou vyvedeny po konzultaci se zadavatelem na pinové lišty.

2.3.4 Deska plošného spoje

Mnoho součástek v originálním schématu není možné koupit, jelikož byla velká část schématu navržena firmou PEAK-System koncem roku 2017. Součástky byly nedostupné buď kvůli jejich stáří nebo z důvodu globálně narušené výroby způsobené pandemií. Objednání 47 různých součástek byl problém, protože s vyprodáním součástky bylo někdy potřebné nahradit ve schématu i sousední součástky. Při hledání náhradních součástek je nutné detailně porovnat v několika datasheetech všechny dostupné hodnoty, limity a grafy. Tento zdlouhavý proces několikrát alespoň vedl k nalezení výhodnější součástky.

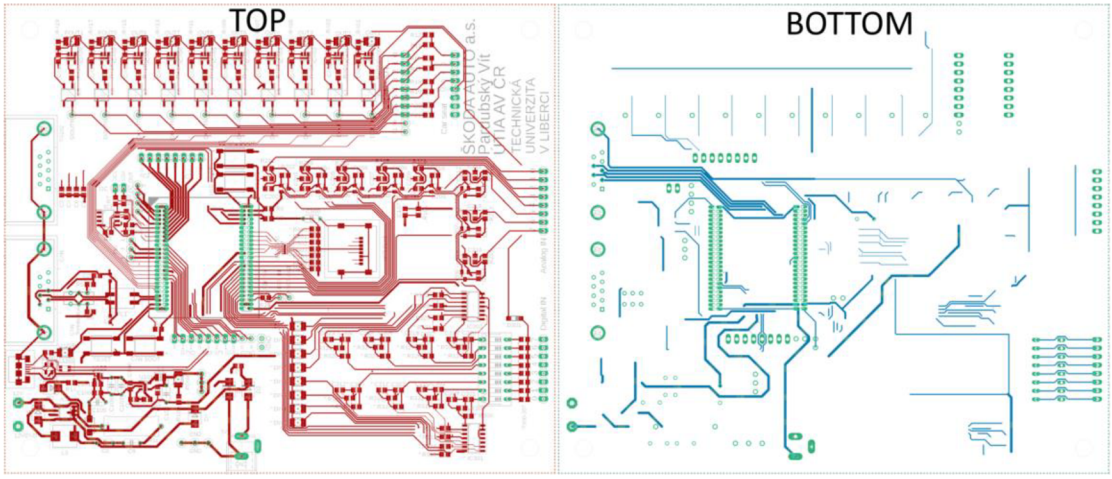
S kompletním schématem a součástkami byla zahájena práce na plošném spoji, Ze začátku návrhu se vyskytl problém s názvy modulů. Součástky přebírají do svého názvu celý název modulu, ve kterém se nacházejí. Název SMD součástky o délce „**U_POWER_SUPPLY:R101**“ zabírá na plošném spoji příliš mnoho místa. Názvy natisknuté na desce plošného spoje jsou užitečné pro orientaci při pájení a při diagnostice. Částečným řešením je použití krátkých názvů modulů například **‘ “ . *** . Název součástek se tím zkrátí na “***:R101**”. Touto náhradou bohužel z nového „názvu“ modulu již nelze poznat funkci. Osobně dodržuji trojciferné značení součástek, kde první cifra značí příslušnost do skupiny součástek a zbylé dvě cifry její pořadí ve skupině.

Návrh je rozdělen na části jako ve schématu: napájení, analogové vstupy, digitální vstupy, digitální výstupy a komunikace. Součástky patřící do stejné skupiny jsou rozmístěny a propojeny v obdélníkovém půdorysu. Tyto části jsou rozmístěny s ohledem na kritéria jako množství zkřížených spojů, rozumný přístup ke konektorům a napájení blízko modulu.



Obr. 20 Části kompletní desky

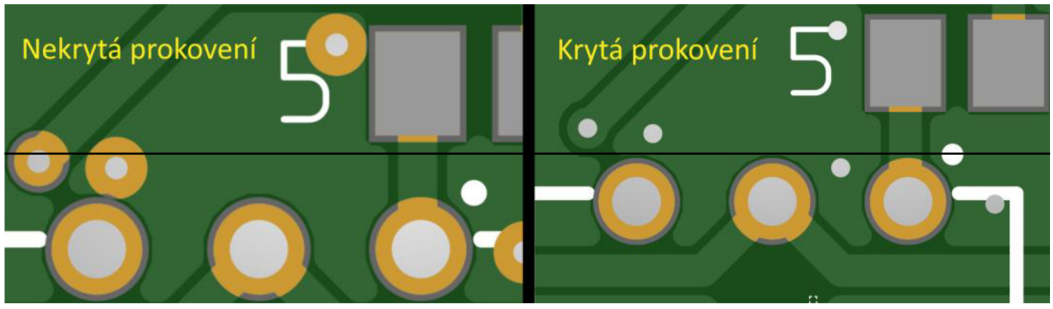
PCAN-MicroMod FD byl umístěn doprostřed, protože z něj je vyvedeno téměř 100 vodičů. Dříve zmíněný THT footprint bránil vyvedení vodičů z vnitřní strany modulu, což omezilo umístění některých částí obvodu. Konektory byly umístěny na stejnou stranu, která pak bude přístupná otvorem v ochranné krabičce. Ostatní signály, které vedou na pinové lišty jsou vpravo, kde mohou vést ven průchody. Hotová deska má rozměry 165 x 140 mm, což by šlo zmenšit. Protože se nejedná o sériovou výrobu a nebude nutné desku umístit do malých prostor, tak nebyla velikost prioritou. Rozestupy zlehčují pájení a diagnostiku případných problémů.



Obr. 21 Srovnání *TOP a BOTTOM* vrstev

Kvůli složitosti desky bylo použití dvou vrstev samozřejmostí, spodní strana byla cíleně méně používána, aby plnila funci Ground plane a stínila desku před elektromagnetickým rušením. Na některých místech je Ground plane rozdělený dlouhými vodiči na tenké pruhy, které by částečně působily jako antény. Tento problém je zmírněn přemostěním pruhů na druhé straně plošného spoje. Prokovení mezi vrstvami desky plní tuto funkci a tvoří jednotnější Ground plane. Tento postup probíhal na základě informací z dokumentu *PCB Design Guidelines For Reduced EMI [20]* od firmy Texas Instruments.

Při převodu plošného spoje na Gerber soubory byl nalezen problém v nastavení prokovení. Nesprávné nastavení *Design Rule Check* vedlo k odkrytým prokovením, která by působila riziko zkratu a poškození elektrostatickým výbojem. Chyba byla včas odstraněna změnou nastavení *DRC>Masks>Limit,* které mění pravidla automatického umisťování nepájivé masky přes otvory.



Obr. 22 Pohled na prokovení v Gerber souborech

Součástky byly objednány u distributora Mouser Electronics. Plošné spoje obsahují minimální mezery 6 mil (0,1524 mm) mezi vodiči, takže bylo nutné vybrat výrobce, který takový proces výroby nabízí. Firma PCBWay splňovala požadavky a nabízela výhodné ceny, takže byla výroba objednána u nich. S cenou za materiály k pájení a tvorbě vlastních kabelů, nákupem hotových kabelů a krabice zůstane celková cena pod 10 000 Kč. Tato částka je v oblasti vývoje automobilových systémů poměrně nízká.

Položka	Přibližná cena na výrobu jednoho kusu
Součástky na spoj	2 000 Kč
Plošný spoj	700 Kč
PCAN-MicroMod FD	3 800 Kč
Ochranná krabice, kabely, konektory…	1000 kč
Součet	**7 500 Kč**

Obr. 23 Rozpis výdajů na výrobu

Cenu plošného spoje by bylo možné snížit jeho zmenšením a odstraněním přebytečných funkcí pro výrobek s pevněji definovanou budoucností. Cena součástek by byla nižší s úpravou obvodu na použití levnějších součástek. Cenu také navyšují nadbytečné konektory a mechanické prvky, kde jsou ceny obecně vyšší než za SMD součástky. Tyto nadbytečné prvky jsou zde kvůli nejistotě ve finálním použití výrobku, takže bylo nutné ponechat možnosti připojení navíc. Jelikož by nemělo těchto modulů být potřeba více než pár kusů, protože nikdy neprobíhá testování sedaček na více než pár vozech, tak je cena přijatelná.

Součástky byly pájeny po skupinách, jejichž funkčnost byla následně testována pomocí multimetru a laboratorního zdroje. Výsledkem měření napájecí části obvodu byly tyto hodnoty: 3,35 V při napájení pomocí 5 V, USB i 12 V. Z napájecí části obvodu je výstupem také přibližných 5 V pro několik integrovaných obvodů. Skutečná hodnota byla po úbytku napětí na blokovacích diodách 4,76 V při napájení 5 V a 4,89 V při napájení 12 V. S funkčním napájením bylo možné postupně pájet a testovat ostatní části obvodu, kde laboratorní zdroj sloužil pro generování různých hodnot signálů.

Během pájení a testování funkce bylo nalezeno několik drobných chyb, které byly následně odstraněny. Chyby byly například: příliš velký footprint pro SMD diody a SMD LED, nedostatečný počet SMD LED. Větší chyba nastala ve schématu u modulu napájení, kde z neznámého důvodu nevzniklo ve schématu propojení s okolím modulu, takže chybělo propojení dvou vodičů. Chybějící spoje byly nahrazeny vodičem a obvod byl potom plně funkční.

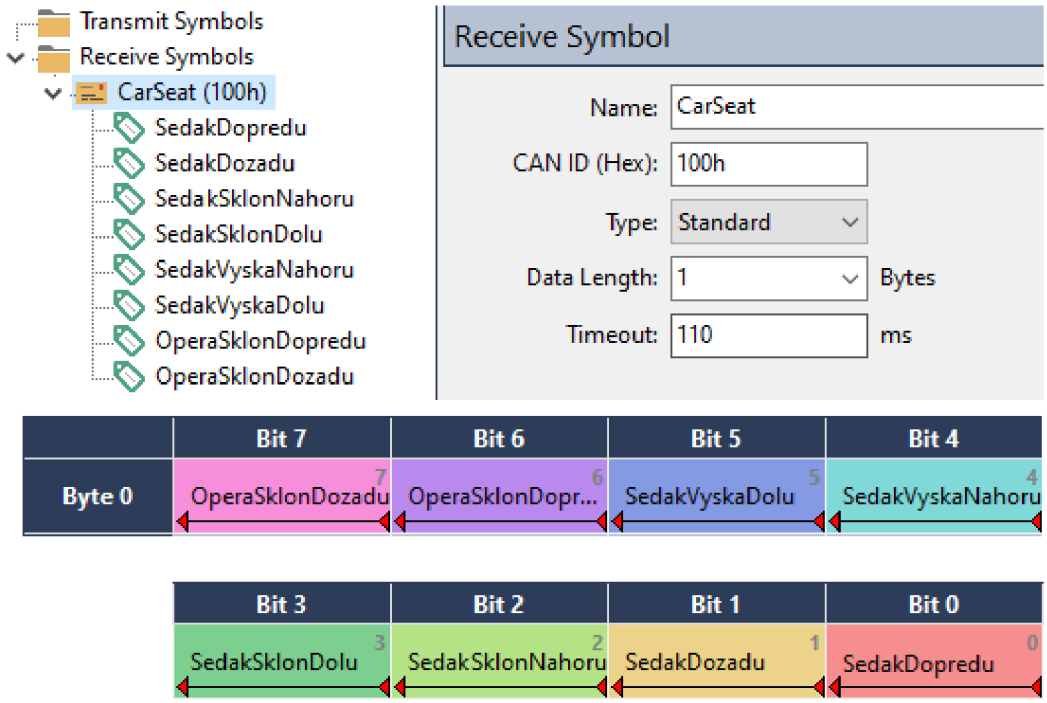
2.3.5 Konfigurace modulu

Do proměřeného plošného spoje bylo bezpečné zapojit modul PCAN-MicroMod FD, který svou diagnostickou LED ohlásil, že chybí platná konfigurace, kterou je nutné vytvořit. Tento modul má firmware umožňující nastavení funkcí pomocí konfiguračního software od PEAK-System. Na tvorbu a nahrání konfigurace se používá program *PCAN-MicroMod FD Configuration*, který je ke stažení na stránkách PEAK-System [14]. Instrukce pro jeho používání jsou v dokumentu *PCAN-MicroMod FD Configuration Tutorial* [16]. Hotové konfigurace se nahrávají přes CAN, a je možné přečíst konfiguraci, která je v modulu momentálně nahraná, což usnadňuje úpravy funkce již zavedených modulů. V konfiguračním souboru se nastavují například tyto parametry:

Signály – mají jméno, popis, počáteční hodnotu a Timeout hodnotu, reprezentují data
Proměnné – signál s daty navíc ve zprávě, definují délku, typ, měřítko a posun signálu
Symboly – formát Data Frame, definuje CAN ID, pozice proměnných v datovém poli, délku datového pole, Timeout platnosti dat a cyklus při odesílání
Digitální výstupy – jsou jim přiřazeny signály, řídí hodnotu výstupu, PWM, měřítko
Funkce – logické, početní a porovnávací funkce pro práci s hodnotami signálů v cyklech

Pro tuto práci byl vytvořen konfigurační soubor typu *PCAN-MicroMod FD Custom Board*, který definuje:
8 signálů - pro ovládání sedačky (SedakDopredu, SedakDozadu…), vrací se po Timeout na logickou 0, aby sedačka sama zastavila.
8 proměnných – délka 1 bit, řídí hodnotu signálů
Receive Symbol – přijímaná zpráva pro ovládání sedačky, typ zprávy Standard a 11-bit CAN ID Data Length 1 (8x1 bit pro signály) Platnost zprávy 110 ms, aby nedocházelo při drobném zpoždění komunikace k zastavování motorů sedačky na milisekundy mezi zprávami
Transmit Symbol – odesílaná zpráva s polohou sedačky, délka 4 byty pro 4 informace o poloze. Je možné symboly cyklicky odesílat, v této konfiguraci je poloha odesílána na dotaz od řídící aplikace

Při výběru zařízení, kam bude konfigurace nahrána, je možný výběr ze 16 zařízení s PCAN-MicroMod FD v jedné síti CAN. Každému modulu musí být přiřazen unikátní ID kombinacemi log 1 a log 0 na pinech pro určení ID.



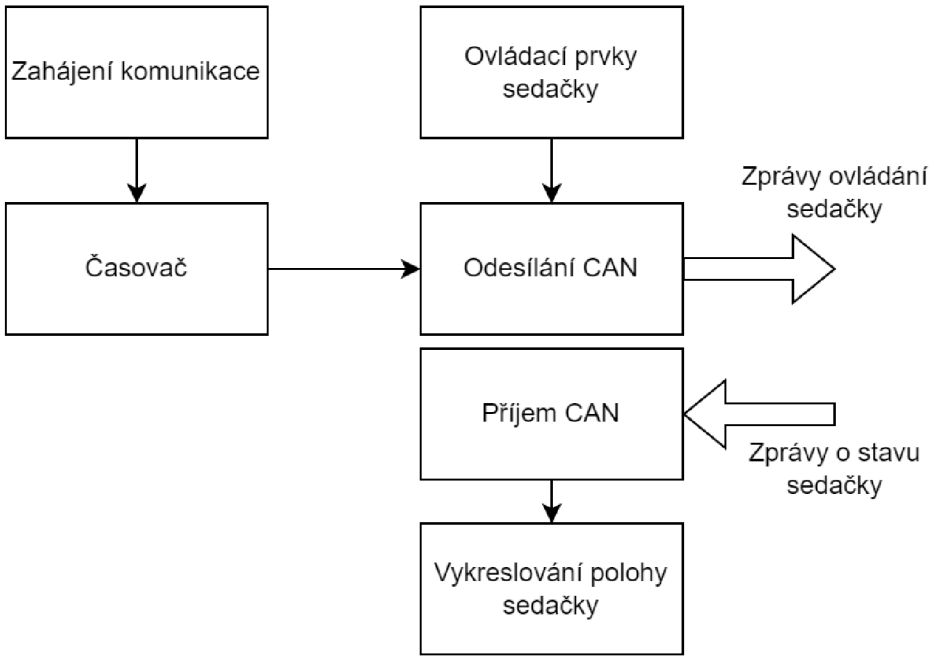
Obr. 24 Použité hodnoty a jejich přiřazení v konfiguraci

Na obrázku jsou nastavení použitá při testování. Zvolena byla délka dat 1 byte, kde každý bit ovládá jednu proměnnou ze seznamu. Přijde-li zpráva s CAN ID 100h, jsou příchozí hodnoty bitů z prvního bytu přiřazeny proměnným podle tabulky na obrázku. Hodnoty proměnných je možné zpracovávat, ukládat a podobně, v tomto případě stačí jejich přímé přiřazení na digitální výstupy 0-7. Tyto výstupy byly nastaveny na *Static* (druhá možnost je PWM) a neinvertující. Výsledkem je, že příchozí 1 v příslušném bitu nastaví svůj přiřazený digitální výstup do aktivního stavu do doby, než přijde další zpráva s logickou 0 nebo vyprší *Timeout*. Motory sedačky lze takto řídit s minimální dobou sepnutí podle intervalu příchozích zpráv, ale samy se zastaví po 110 ms bez nové zprávy, což je vhodné z bezpečnostního hlediska, protože přerušení CAN komunikace nepovede k neřízeně se pohybující sedačce.

K nahrání konfigurace na modul a následnému testování byl použit *PCAN-USB* [21]. Po instalaci potřebných ovladačů a programu *PCAN-View* umožňuje odesílání CAN zpráv z počítače. Takto bylo ověřeno spínání digitálních výstupů na desce podle příchozích CAN zpráv, správnost obsahu zpráv o poloze sedačky a doby cyklů komunikace.

2.4 Aplikace na PC

2.4.1 Struktura aplikace



Obr. 25 Blokový diagram aplikace

Zahájení komunikace - Nejprve je nutné zahájit komunikaci CAN. To je možné po nalezení volného kanálu a jeho inicializaci. Po úspěšné inicializaci kanálu je spuštěn časovač.

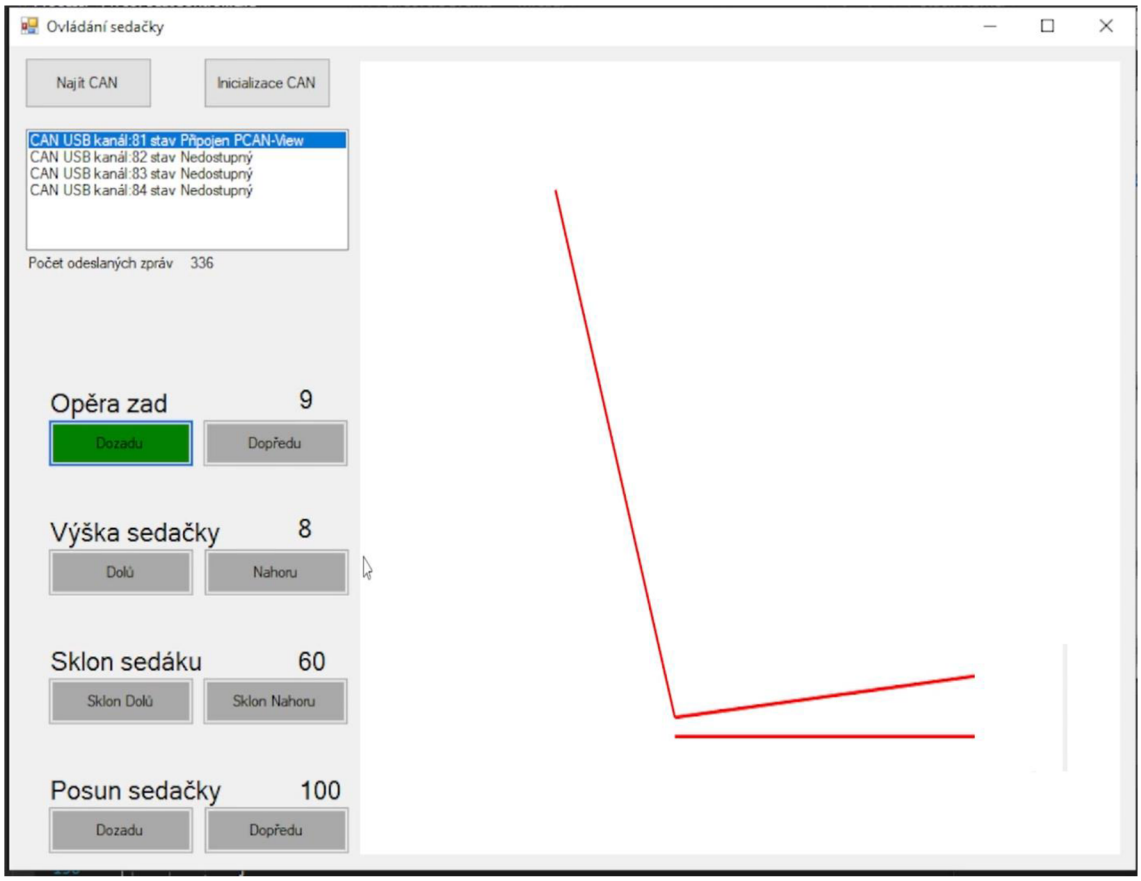
Časovač – CAN zprávy jsou běžně posílány cyklicky, zde je zvolen cyklus 100 ms. Uplynutí intervalu časovače zahájí odesílání řídících zpráv.

Ovládací prvky sedačky – tlačítka v uživatelském rozhraní, která slouží pro výběr směru pohybu částí sedačky.

Odesílání CAN – Na jiném vlákně je založen *Task* pro odesílání CAN zpráv. Odesílány jsou řídící zprávy pro ovládání sedačky podle stavu ovládacích prvků na grafickém rozhraní. Odeslání řídící zprávy sedačce vede ke změně polohy, vyšle se tedy zpráva se žádostí o aktuální polohu sedačky.

Příjem CAN – Nezávislé vlákno sleduje buffer příchozích zpráv, po příchodu zprávy se zahájí proces vykreslování.

Vykreslování polohy sedačky – Data o aktuální poloze sedačky se použijí k výpočtu nákresu sedačky, nákres je zobrazen v uživatelském rozhraní.



Obr. 26 Vzhled aplikace

V levém horním rohu se nachází ovládání pro zahájení komunikace:
Najít CAN – Je zjištěn stav USB kanálů a je vypsán seznam. Testovány jsou jen USB kanály, protože v této práci je používán *PCAN-USB*. Možné stavy kanálu jsou *Nedostupný, Dostupný, Již inicializovaný, Připojený PCAN-View*.

Inicializace CAN – Tlačítko je aktivováno jen po výběru použitelného kanálu ze seznamu, po jeho stisknutí je provedena inicializace zvoleného kanálu a výsledek je oznámen v dialogovém okně.

Po inicializaci CAN je možné ovládat sedačku pomocí osmi tlačítek pro výběr směru pohybu prvků sedačky. Po výběru směru zezelená aktivní tlačítko. Opakovaný stisk aktivního tlačítka zastaví pohyb, výběr jiného tlačítka změní směr pohybu

Z dat aktuální polohy je vykreslován nákres sedačky v pravé části rozhraní aplikace. Aktuální poloha jednotlivých směrů je uvedena u příslušných ovládacích prvků v rozsahu 0 až 100, které značí pozici sedačky mezi krajními body. Dosažení krajního bodu automaticky zastaví pohyb sedačky.

2.4.2 Implementace

Aplikace je stavěna jako *Windows Forms App (.NET Framework*) v jazyce C#, v prostředí *Microsoft Visual Studio*. C# byl vybrán po dohodě s konzultantem z důvodu jeho použití v souvisejících programech ve ŠKODA AUTO. Pro umožnění komunikace CAN přes PCAN-USB slouží vývojářské balíčky na stránkách PEAK-System. Tato práce využívá konkrétně balíček PCAN-Basic API [22], který slouží pro tvorbu aplikací na komunikaci mezi PC a zařízeními firmy PEAK-System. V balíčku jsou obsaženy instalační programy potřebných driverů pro funkci připojených zařízení a knihovna definic a funkcí pro programování s CAN. Mezi podporované jazyky patří C#, C++, Python, Java a další. Pro programy v C# je používán soubor *PCANBasic.cs* typu *C# Source File*, který po vložení do projektu přidá standardní definice proměnných a funkcí. V dokumentaci *PCAN-Basic Parameters Description* jsou uvedena jména všech zahrnutých parametrů, jejich možné hodnoty a příklady použití v C++ a C#. Druhou částí dokumentace jsou komentáře v souboru *PCANBasic.cs*, které obsahují vysvětlení funkcí.

Pro správnou funkci aplikace je nutné použít více vláken. V této implementaci jsou to konkrétně hlavní vlákno pro chod programu a uživatelské rozhraní, vlákno časovače, vlákno odesílání, vlákno hlídání příchodu zprávy, vlákno čtení zpráv a vlákno zpracování a vykreslení polohy sedačky.

Z PCAN-Basic API jsou v této aplikaci použity následující funkce:

- PCANBasic.Initialize(…) – slouží pro zavedení CAN kanálu o určité rychlosti
- PCANBasic.GetErrorText(…) – převod chybového kódu na text chyby
- PCANBasic.GetValue(…) – čtení hodnot parametru kanálu
- PCANBasic.SetValue(…) – nastavení parametru na určitou hodnotu
- PCANBasic.Write(…) – zapíše do bufferu zprávu na odeslání
- PCANBasic.Read(…) – přečte z bufferu příchozí zprávu

Jako příklad jednoduché struktury používání funkcí PCANBasic je uvedena část kódu tlačítka Inicializace CAN.

```
// Provedení příkazu inicializace uvnitř podmínky if a uloží se výstupní hodnota funkce
if ((condition = PCANBasic.Initialize(Channel,Baud)) == TPCANStatus.PCAN_ERROR_OK)
{  // Pokud proběhne příkaz bez problému, proběhne tato část kódu. Ukáže se dialogové okno
   // o úspěchu a zapne se časovač na 100 ms
   MessageBox.Show("Kanál " + ChannelHandle.ToString() + " inicializován");
   t100ms.Enabled = true;
}
else
{ // V případě neúspěchu se funkcí GetErrorText vytáhne z uložené výstupní hodnoty
// konkrétní chyba, na kterou můžeme nějak reagovat. V tomto případě se zobrazí dialogové
// okno s důvodem neúspěchu
   PCANBasic.GetErrorText(condition, L, ErrorDescription);
   MessageBox.Show(ErrorDescription.ToString());
}
```

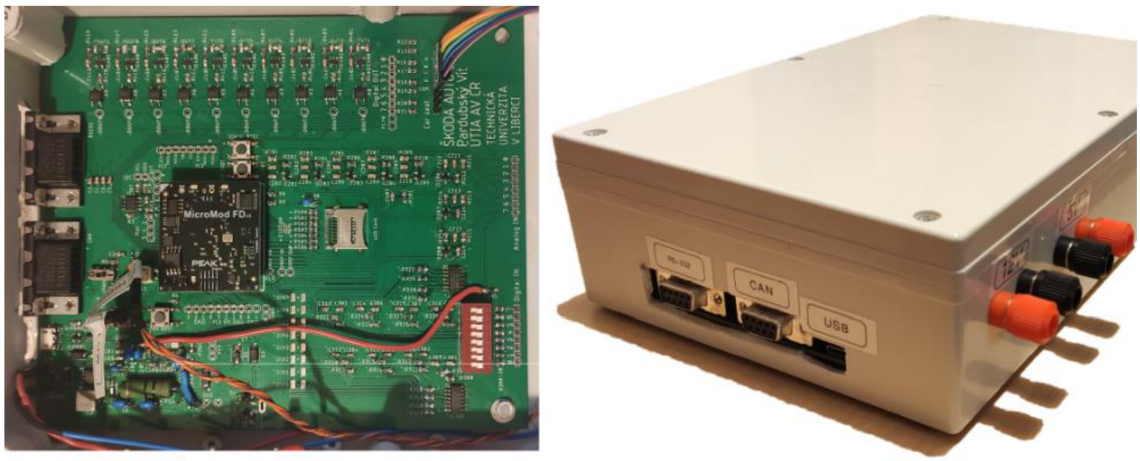
Struktura rámců CAN zpráv znamená, že je nutné pro každou používanou zprávu definovat hodnoty částí jejího rámce. Například *Data Frame* pro odesílání zpráv může být definován následujícím způsobem

```
TPCANMsg SendBuffer; // Jméno použité pro celou zprávu
SendBuffer.MSGTYPE = TPCANMessageType.PCAN_MESSAGE_STANDARD; // Rozlišení
                    // Standard a Extended, Remote Request Frame, CAN FD a další
SendBuffer.ID = 0x100; // Identifikátor zprávy, zde 0x100 je ID ovládání sedačky
SendBuffer.LEN = 1; // délka odesílané části datového pole
SendBuffer.DATA = new byte[8]; // v CAN Standard vždy 8 bytů
SendBuffer.DATA[0] = (byte)Zprava; // Přepsání dat do datového pole před
                                   // odesláním
PCANBasic.Write(ChannelHandle, ref SendBuffer)) // Odeslání zprávy
```

Datové pole CAN zprávy je typu *byte*, což vede k potřebě převedení dat před odesláním a zpětnému převedení po přijetí. Pro delší typy dat delších než 8 bitů jako jsou *long a double* je nutné i správné seřazení bytů z datového pole. V aplikaci je několik proměnných definováno jako byte/uint, protože je to shodné s konfigurací PCAN-MicroMod FD pro tuto práci.

Pro získání dat o aktuální poloze sedačky je nutné odeslat zprávu s příslušným identifikátorem, která je nastavená na formát *Remote Transmission Request* (`Zprava.MSGTYPE = TPCANMessageType.PCAN_MESSAGE_RTR;`). PCAN-MicroMod FD je nakonfigurován tak, aby na tuto žádost odpověděl zprávou se čtyřmi byty s údaji o poloze. Příchozí zpráva je přečtena a proběhne přepočet polohy na souřadnice pro vykreslení reprezentace sedačky. Pro jednoduché vykreslení stačí tři úsečky: opěra zad, sedák, základ sedačky. Grafické příkazy probíhají na *bitmap*, která je následně zobrazena v panelu vpravo v uživatelském rozhraní.

2.5 Výsledky



Obr. 27 Hotový výrobek a obal

Testování probíhalo na zkušebním vozu ŠKODA Octavia 3. generace. Systém je po zapojení plně funkční. Při ovládání sedačky nepřidává systém vnímatelné zpoždění (maximálně 100 ms kvůli cyklu komunikace). První verze ovládací aplikace přidávala u vykreslování polohy sedačky oproti skutečnému stavu zpoždění přibližně 300 až 400 ms, kvůli většímu množství operací, jako odeslání požadavku o data, zpracování a odeslání dat, příjem dat a vykreslení na uživatelské rozhraní. Při nadměrné zátěži nastával na CAN lince některý z chybových stavů, které nebyly ošetřeny. Za normálního používání se systém choval dle očekávání i po hodině chodu, ale aplikace nebyla uspokojivé kvality.

Druhá verze řídící aplikace, jejíž kód je přiložen u elektronické verze bakalářské práce, napravila některé nevhodně navržené části programu. Několik situací, které v první verzi spolehlivě vedly k chybovým stavům a následnému zastavení funkce ovládání, bylo vyřešeno. Aplikace byla testována po dobu dvou hodin a odesláno 50 000 CAN zpráv, aniž by nastal chybový stav. První verze četla data polohy poloviční rychlostí, takže bylo v druhé verzi sníženo zpoždění vykreslování.

Závěr

Cílem této práce byl návrh a realizace hardwaru včetně softwaru potřebného k měření pozice a ovládání pohonů elektrické sedačky Škoda pro použití v oblasti vývoje. K naplnění požadavků na výrobek bylo provedeno:

- Průzkum protokolu CAN, jeho jednotlivých vrstev, vlastností a reálné použití v automobilech.
- Byly zjištěny a změřeny potřebné informace pro připojení, datovou komunikaci a ovládání řídící jednotky elektrické sedačky vozu ŠKODA. Konkrétně bylo vyhledáno přípojné místo ve vozidle, analýza CAN zpráv řídicí jednotky sedačky, způsobu ovládání pohonů a změření elektrických hodnot ovládacích prvků sedačky.
- V praktické části byl navržen a realizován HW rozšiřující desky pro modul PCAN-MicroMod FD což zahrnuje návrh schématu, návrh DPS, osazení součástkami, oživení a otestování. Během oživení byly zjištěny a opraveny drobné chyby při návrhu DPS.
- V rámci oživení byla vytvořena konfigurace pro modul PCAN-MicroMod FD, umožňující spínání polovodičových prvků, které nahrazují fyzická tlačítka ovládání automobilové sedačky, pomocí CAN zpráv. Následně byla ověřena správná funkce hardwarové části přímo na sedačce vozidla. Tj. realizovanému HW byly posílány řídicí CAN zprávy pro spínání jednotlivých pohonů a ověřeny reálné pohyby sedačky.
- Pro uživatelsky příznivější ovládání sedačky a vlastní odměřování pozice sedačky byla realizována aplikace v jazyce C#. Ta implementuje komunikaci CAN, přes kterou jsou ovládány jednotlivé pohony sedačky a je vyčítána její aktuální poloha. Ta je v GUI aplikace vypisována a pro doplnění je sedačka schematicky vykreslena.

Kombinací Windows aplikace, hardwarového výrobku a modulu PCAN-MicroMod FD bylo dosaženo cíle měření polohy a ovládání pohonů elektrické automobilové sedačky, jak žádalo zadání. Výsledek této práce bude začleněn do sady testovacích nástrojů v Technologickém Centru Česana. Hardware byl navržen s možností rozšíření o další vstupy/výstupy a přípravu na SD kartu, takže může být přizpůsoben budoucím požadavkům pro další projekty.

Jedním z možných pokračování je nahradit existující firmware modulu PCAN-MicroMod FD tak, aby mohl fungovat samostatně bez další aplikace a například sám realizoval diagnostický protokol pro komunikaci se sedačkou a aktuální pozici ukládal do logu na SD kartu. Program s takovou funkcí tovární firmware neumožnuje.

Seznam použité literatury

[1] HAMMERSCHMIDT, Christoph, 2019. Number of automotive ECUs continues to rise. In: *EeNews Automotive* [online]. Belgie: eeNews Europe Head Office European Business Press, 15. 5. 2019 [cit. 2022-02-15]. Dostupné z: https://www.eenewsautomotive.com/en/number-of-automotive-ecus-continues-to-rise/?more=1

[2] M. KESSLER, Aaron, 2015. Fiat Chrysler Issues Recall Over Hacking. *The New York Times* [online]. New York: The New York Times Company a Arthur Gregg Sulzberger, 24. 7. 2015 [cit. 2022-02-21]. Dostupné z: https://www.nytimes.com/2015/07/25/business/fiat-chrysler-recalls-1-4-million-vehicles-to-fix-hacking-issue.html

[3] T. Nolte, H. Hansson and L. L. Bello, "Automotive communications-past, current and future," 2005 IEEE Conference on Emerging Technologies and Factory Automation, 2005, pp. 8 pp.-992, doi: 10.1109/ETFA.2005.1612631. [cit. 2022-03-13]

[4] *CAN Specifitation: Version 2.0*, 1991. 1991. Stuttgart: Robert Bosch. [cit. 2022-02-16]. Dostupné z: http://esd.cs.ucr.edu/webres/can20.pdf

[5] Kvaser Inc.: CAN Physical layers, 2018. *Https://www.kvaser.com* [online]. Mission Viejo, CA 92691 USA, 27. 4. 2018 [cit. 2022-03-05]. Dostupné z: https://www.kvaser.com/lesson/can-physical-layers/

[6] ISO 11898-2:2016, 2016. *Road vehicles — Controller area network (CAN): Part 2: High-speed medium access unit*. 2. Ženeva: International Organization for Standardization.

[7] CAN high-speed transmission. In: *CiA: CAN Knowledge* [online]. Nuremberg: CAN in Automation (CiA) [cit. 2022-03-08]. Dostupné z: https://www.can-cia.org/can-knowledge/can/high-speed-transmission/

[8] GRIFFITH, John, 2016. Why are termination networks in CAN transceivers so important?. *Texas Instruments* [online]. Dallas, Texas: Texas Instruments Incorporated, 14. 7. 2016 [cit. 2022-03-11]. Dostupné z: https://e2e.ti.com/blogs_/b/industrial_strength/posts/the-importance-of-termination-networks-in-can-transceivers

[9] CORRIGAN, Steve, 2018. *Controller Area Network Physical Layer Requirements* [Online]. Dallas, Texas: Texas Instruments Incorporated. [cit. 2022-03-13]. Dostupné z: https://www.ti.com/lit/an/slla270/slla270.pdf?ts=1651499420248&ref_url=https%253A%252F%252Fwww.google.ca%252F

[10] GENERIC AUTOMOTIVE PROTOCOLS, 2016. *VW POLO* [online]. [cit. 2022-03-15]. Dostupné z: https://volkswagen-polo-highline.blogspot.com/2016/07/generic-automotive-protocols.html

[11] Airbags, 2021. *IIHS-HLDI* [online]. Arlington, Virginia: Insurance Institute for Highway Safety (IIHS) [cit. 2022-03-15]. Dostupné z: https://www.iihs.org/topics/airbags

[12]CAN Protocols. *Bosch Semiconductors* [online]. Reutlingen, Germany: Robert Bosch [cit. 2022-03-14]. Dostupné z: https://www.bosch-semiconductors.com/ip-modules/can-protocols/

[13] High Speed CAN with Flexible Data Rate (CAN FD). *NXP Semiconductors* [online]. Eindhoven, Netherlands: NXP Semiconductors [cit. 2022-03-16]. Dostupné z: https://www.nxp.com/products/interfaces/can-transceivers/can-with-flexible-data-rate:CAN-FDR#/

[14] PCAN-MicroMod FD, 2019. *PEAK-System* [online]. Darmstadt, Germany: PEAK-System Technik [cit. 2022-03-16]. Dostupné z: https://www.peak-system.com/PCAN-MicroMod-FD.500.0.html?&L=1

[15] Power-Efficient Microcontrollers (MCUs) With Advanced Peripherals Based on Arm® Cortex®-M4 Core: LPC546XX, 2020. *NXP Semiconductors* [online]. Eindhoven, Netherlands: NXP Semiconductors [cit. 2022-03-16]. Dostupné z: https://www.nxp.com/products/processors-and-microcontrollers/arm-microcontrollers/general-purpose-mcus/lpc54000-cortex-m4-/power-efficient-microcontrollers-mcus-with-advanced-peripherals-based-on-arm-cortex-m4-core:LPC546XX

[16] *PCAN-MicroMod FD Evaluation Board: User Manual* [online], 2020. Version 1.1.0. Darmstadt, Germany: PEAK-System Technik [cit. 2022-03-16]. Dostupné z: https://www.peak-system.com/produktcd/Pdf/English/PCAN-MicroMod-FD-Eval_UserMan_eng.pdf

[17] *PCAN-MicroMod FD: User Manual* [online], 2020. Version 1.1.1. Darmstadt, Germany: PEAK-System Technik [cit. 2022-03-16]. Dostupné z: https://www.peak-system.com/produktcd/Pdf/English/PCAN-MicroMod-FD_UserMan_eng.pdf

[18] *PCAN-MicroMod FD Evaluation Board* [online], 2020. In: . Darmstadt Germany: PEAK-System Technik [cit. 2022-03-16]. Dostupné z: https://www.peak-system.com/fileadmin/images/products/hardware/product_pictures/PCAN-MicroMod-FD-Evaluation-Board.jpg

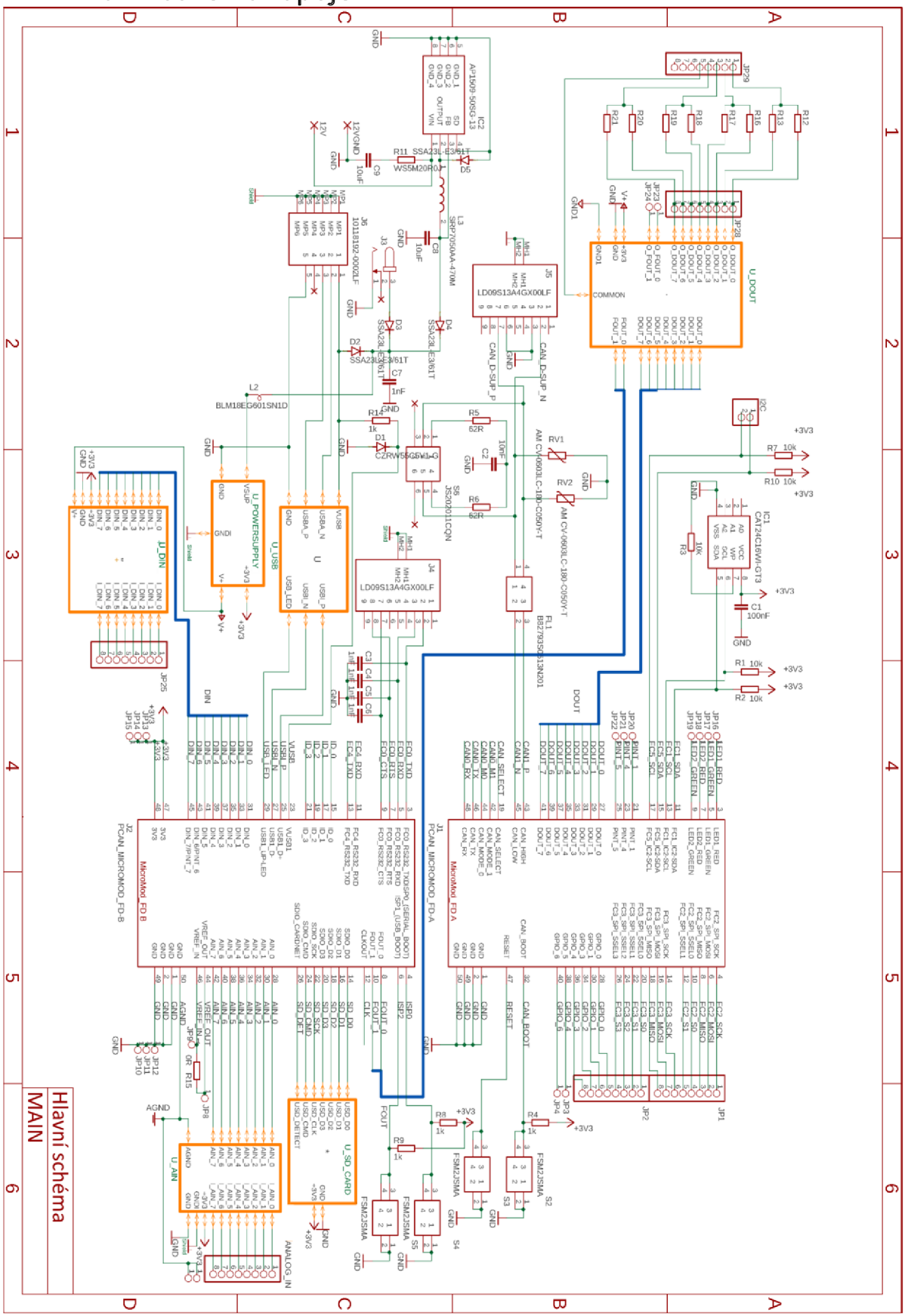
[19] *NÁVOD K OBSLUZE: ŠKODA Superb* [online], 2016. In: . Česká Republika: ŠKODA AUTO [cit. 2022-02-27]. Dostupné z: https://manual.skoda-auto.com/004/en-com/Detail?model=Superb_3V&edition=05-2016&market=&manualLang=cs

[20] *PCB Design Guidelines For Reduced EMI* [online], 1999. In: . Dallas, Texas: Texas Instruments Incorporated [cit. 2022-03-05]. Dostupné z: https://www.ti.com/lit/an/szza009/szza009.pdf

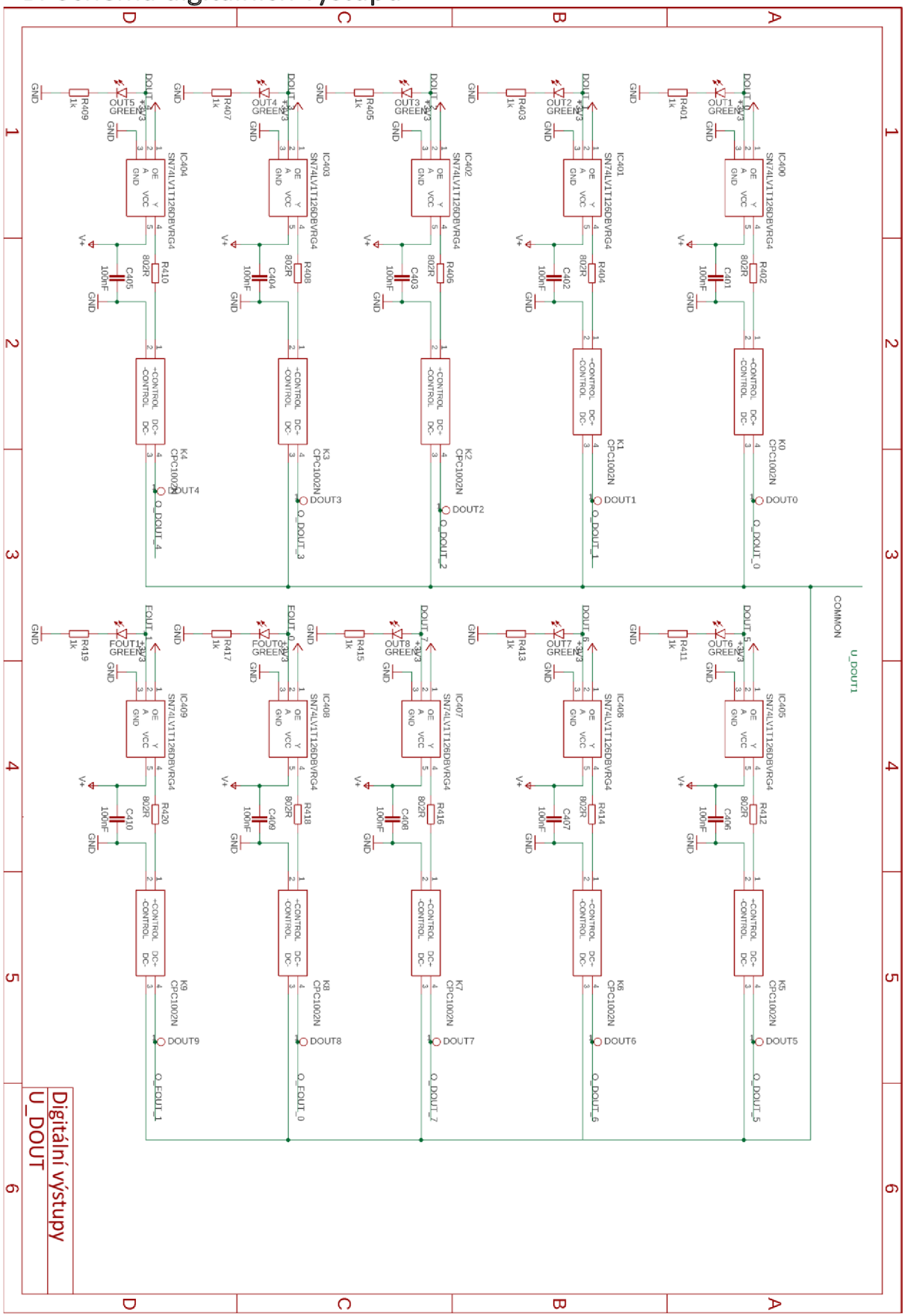
[21] *PCAN-USB: CAN Interface for USB* [online], 2021. Darmstadt, Germany: PEAK-System Technik [cit. 2022-04-02]. Dostupné z: https://www.peak-system.com/PCAN-USB.199.0.html?&L=1

[22] *PCAN-Basic: API for Connecting to CAN and CAN FD Buses* [online], 2021. Darmstadt, Germany: PEAK-System Technik [cit. 2022-04-05]. Dostupné z: https://www.peak-system.com/PCAN-Basic.239.0.html?&L=1

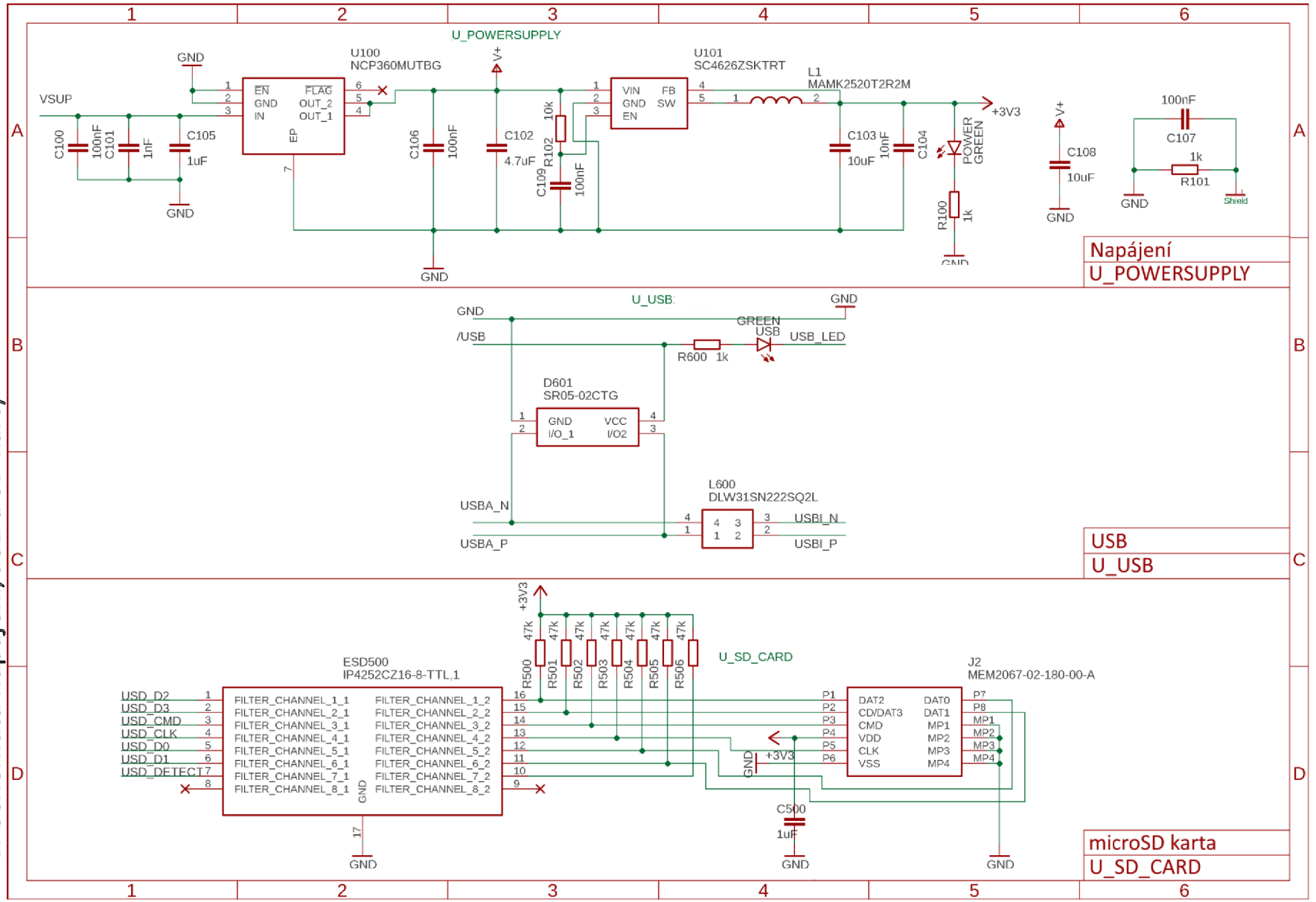
A. Hlavní schéma zapojení



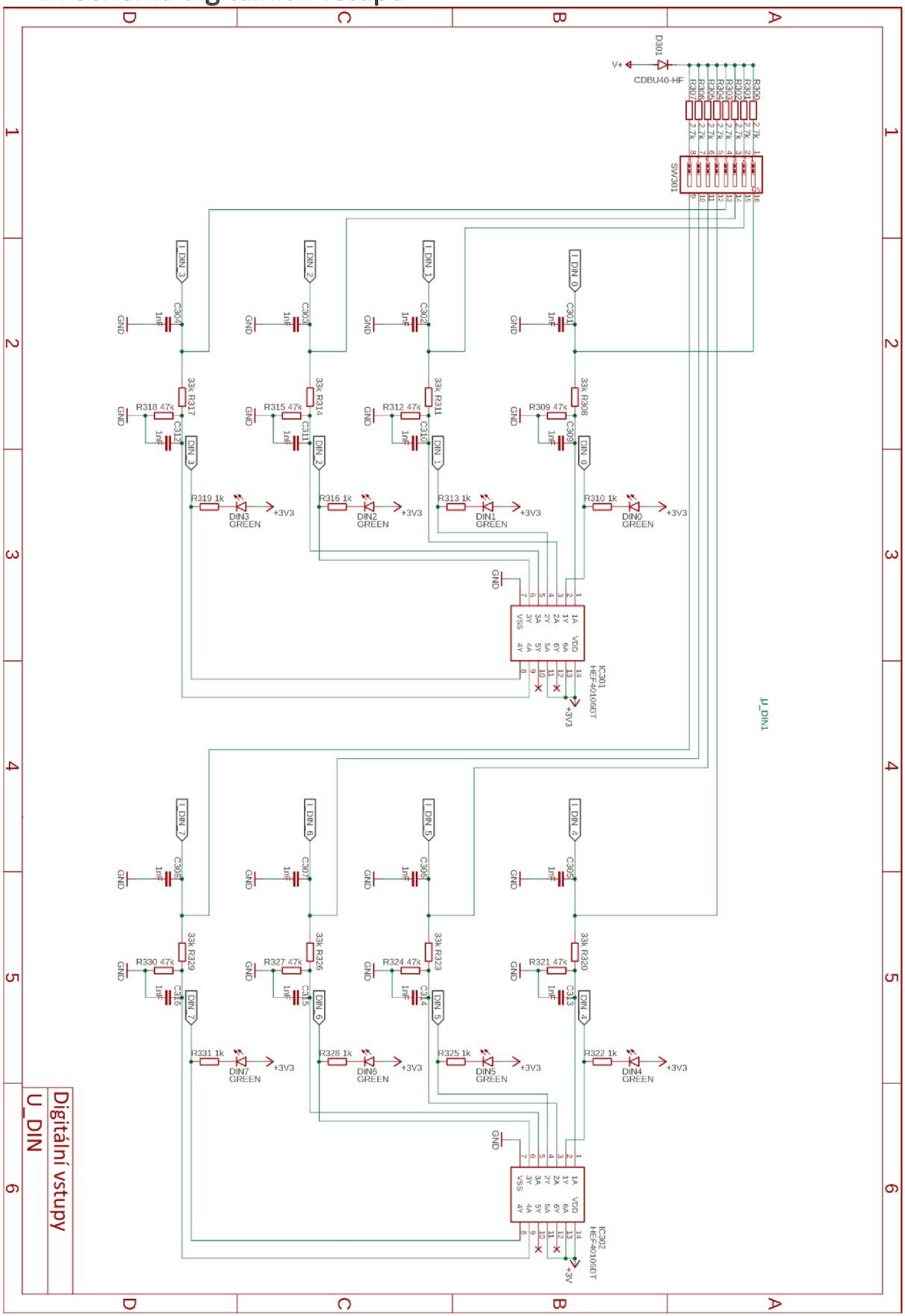
B. Schéma digitálních výstupů



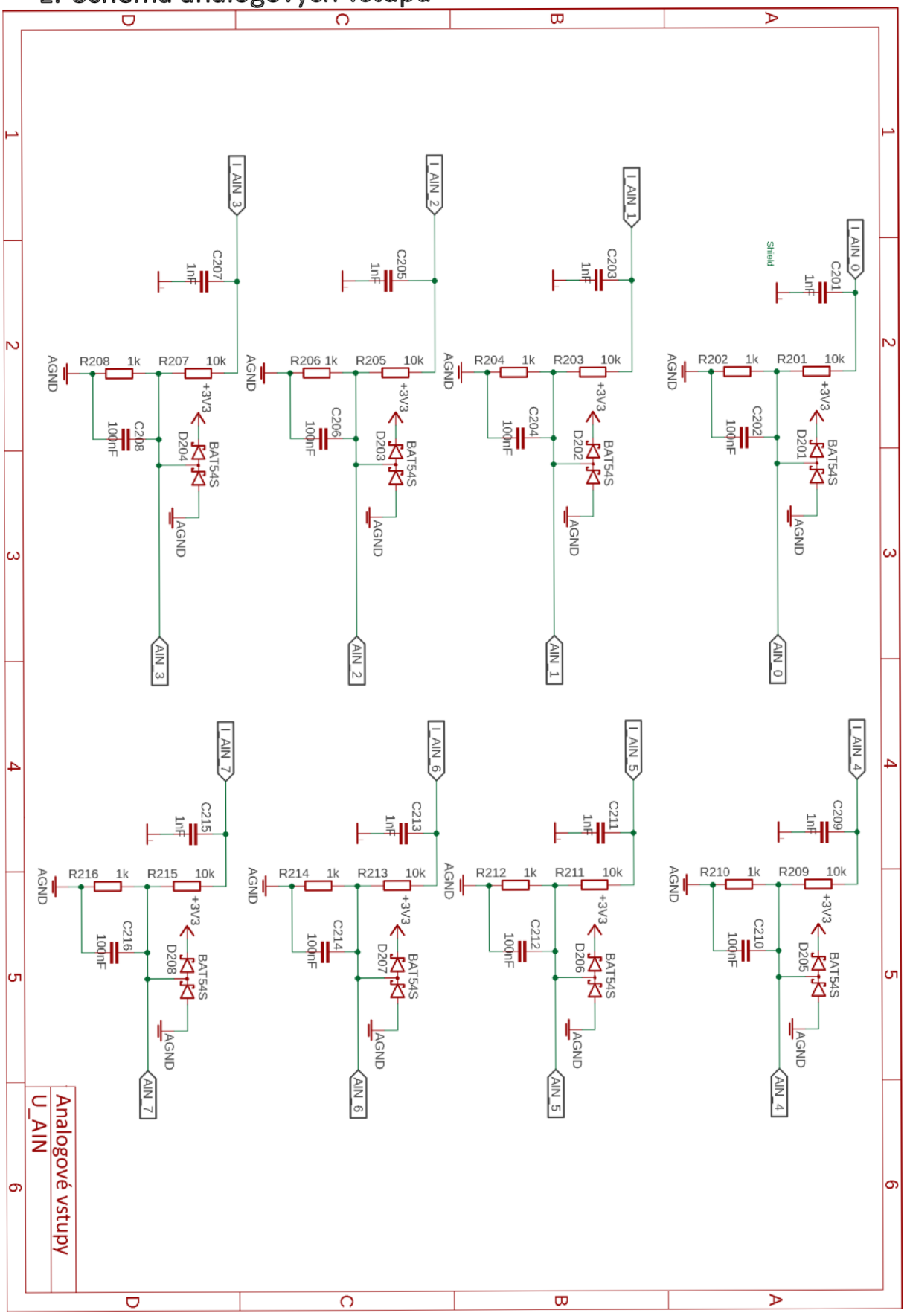
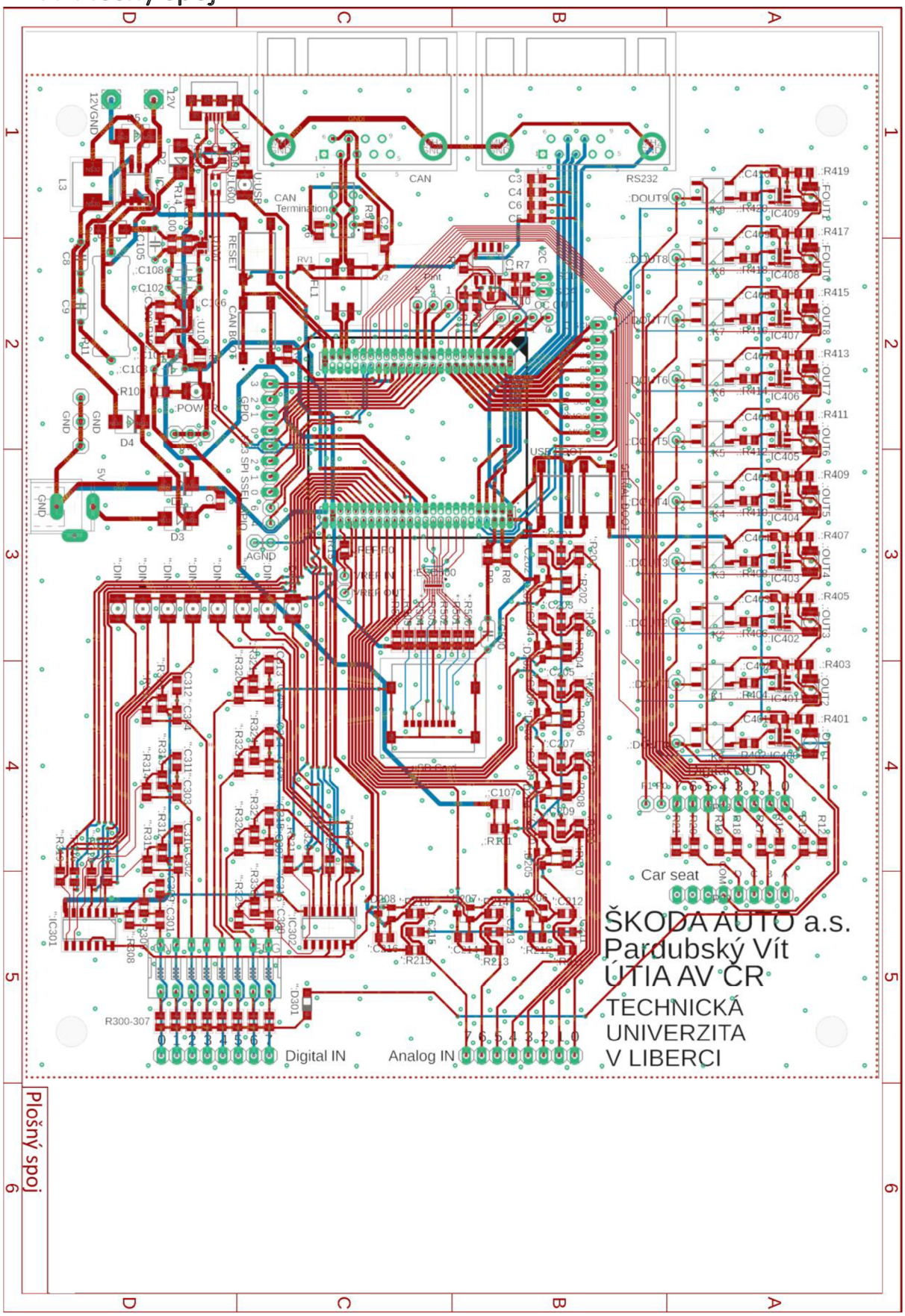
C. Schémata napájení,USB a SD karty



D. Schéma digitálních vstupů



E. Schéma analogových vstupů



F. Plošný spoj