



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

ANALÝZA KOMUNIKACE ELEKTRONICKÝCH ŘÍDÍCÍCH JEDNOTEK AUTOMOBILU

ANALYSIS OF THE COMMUNICATION BETWEEN CAR'S ELECTRONIC CONTROL UNITS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Štěpán Zouhar

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Tomáš Spáčil

BRNO 2017

Zadání bakalářské práce

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Student:	Štěpán Zouhar
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Mechatronika
Vedoucí práce:	Ing. Tomáš Spáčil
Akademický rok:	2016/17

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Analýza komunikace elektronických řídicích jednotek automobilu

Stručná charakteristika problematiky úkolu:

V rámci předchozí spolupráce s firmou Škoda Auto bylo třeba analyzovat zprávy na sběrnici CAN a individuálně je zpracovávat. Pomocí HW a SW Vector se dá tento proces automatizovat, což může najít uplatnění například při HIL testování řídicích jednotek.

Cílem práce bude zdokumentovat a provést konstrukční úpravy na testovacím standu s řídicími jednotkami a detailně prozkoumat možnosti SW Vector CaNoe, propojitelnost s programem Matlab a tvorba vzorových aplikací.

Cíle bakalářské práce:

- 1) Dokončení mechanických úprav testovacího standu a tvorba dokumentace.
- 2) Podrobná analýza možností programu Vector.
 - 2.1) Vytvořit vzorovou úlohu. Jednoduchá simulace jízdy.
- 3) Propojení nástrojů Vector s programem Matlab.
 - 3.1) Vytvořit vzorovou úlohu. Simulace jízdy automobilu za využití tempomatu. Tempomat, zpětnovazební řízení rychlosti, bude realizován v Matlab Simulink.

Seznam doporučené literatury:

VALÁŠEK, Michael. Mechatronika. Dot. 1. vyd. Praha: České vysoké učení technické, 1996. ISBN 80-01-01276-X.

NOSKIEVIČ, Petr. Modelování a identifikace systémů. Ostrava: Montanex, 1999. ISBN 80-7225-0-0-2.

SKALICKÝ, Jiří. Teorie řízení 1. Vyd. 1. Brno: Vysoké učení technické, 2002. Učební texty vysokých škol. ISBN 80-214-2112-6.

Robert Bosch GmbH, CAN Specification 2.0, dostupné na: <http://www.bosch-semiconductors.de>

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2016/17

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Tato bakalářská práce má 3 hlavní části. V první části je tématem testovací stav, který vznikl pro účely vývoje a testování elektronických zařízení pro automobily. Následuje část zabývající se komunikací po sběrnici CAN a propojení nástrojů CANoe a Simulink pro účely simulace chování řídicích jednotek automobilu. V poslední části byly vytvořeny 2 vzorové úlohy: jedna pro prostředí CANoe a druhá pro spojení obou nástrojů CANoe a Simulink.

KLÍČOVÁ SLOVA

CAN sběrnice, CAN zpráva, Vector, CANoe, Simulink, simulace, testování

ABSTRACT

This bachelor thesis has 3 basic parts. In the first part is the main topic testing stand which was made for purpose of developing and testing electronic devices for cars. The next part deals with communication on the CAN bus and interface between tools CANoe and Simulink for simulation purposes of ECU's (Electronic Control Unit) behaviour. At the end were created 2 examples: one for CANoe and the second one for interface between CANoe and Simulink.

KEYWORDS

CAN bus, CAN message, Vector, CANoe, Simulink, Simulation, Testing

Bibliografická citace

ZOUHAR, Š. *Analýza komunikace elektronických řídicích jednotek automobilu*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky, 2017. 36 s. Bakalářská práce. Vedoucí práce: Ing. Tomáš Spáčil.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma *Analýza komunikace elektronických řídicích jednotek automobilu* jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících 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, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

Štěpán Zouhar

PODĚKOVÁNÍ

Poděkování patří vedoucímu práce Ing. Tomášovi Spáčilovi za cenné rady a připomínky při vypracovávání této bakalářské práce a Mechatronické laboratoři za poskytnutí vhodného zázemí pro tvorbu práce.

OBSAH

Seznam obrázků	8
Seznam tabulek	9
Úvod	10
1 Testovací stand Škoda	11
1.1 Mechanická část.....	11
1.2 Elektrické zapojení	13
1.2.1 Sběrnice CAN	14
1.2.2 CAN zpráva	16
1.2.3 Detekce chyb.....	19
1.2.4 Topologie sítě CAN	20
2 Hardware a Software VECTOR	21
2.1 Analýza možností	21
2.1.1 Hardware Vector 1630A	21
2.1.2 Software CANoe.....	22
2.2 Propojení nástrojů CANoe a MATLAB	23
3 Vzorové úlohy	26
3.1 Vzorová úloha v software CANoe.....	26
3.2 Vzorová úloha na propojení nástrojů CANoe a Simulink	28
3.2.1 Regulační smyčka pro regulaci rychlosti.....	28
3.2.2 Matematický popis jízdy.....	29
3.2.3 Popis modelů v Simulinku.....	30
3.2.4 Popis simulace v CANoe	34
4 Závěr	35
Literatura	36

SEZNAM OBRÁZKŮ

Obrázek 1.1 Model testovacího standu vytvořený v SolidWorks	11
Obrázek 1.2 Testovací stand	12
Obrázek 1.3 Propojení vodičů pomocí WAGO svorkovnic na levém boku standu	12
Obrázek 1.4 Elektronické řídicí jednotky na pravé straně testovacího standu	13
Obrázek 1.5 Elektrické svazky uvnitř testovacího standu	14
Obrázek 1.6 Struktura rámce podle specifikace CAN 2.0A [5]	17
Obrázek 1.7 Struktura rámce podle specifikace CAN FD [5]	17
Obrázek 1.8 Srovnání rámce CAN a CAN FD [5]	17
Obrázek 1.9 Arbitráž CAN zpráv vysílaných v jeden okamžik [4].....	19
Obrázek 1.10 Součásti uzlu zajišťující komunikaci po CAN sběrnici [4].....	20
Obrázek 2.1 Zařízení CAN/LIN Interface VN 1630A firmy Vector s připojeným CAN kanálem.....	21
Obrázek 2.2 Prostředí CANoe [10].....	23
Obrázek 2.3 Časová synchronizace v offline módu [3].....	25
Obrázek 2.4 Časová synchronizace v synchronizovaném módu [3]	25
Obrázek 3.1 Vzorová úloha v prostředí CANoe.....	28
Obrázek 3.2 Blokový diagram regulace rychlosti	29
Obrázek 3.3 Schéma modelu celého systému v Simulinku	31
Obrázek 3.4 Průběh rychlosti a otáček motoru v závislosti na čase	31
Obrázek 3.5 Průběh momentu na kole v závislosti na čase	32
Obrázek 3.6 Model převodovky (Plant)	32
Obrázek 3.7 Model řídicí jednotky automatické převodovky	33
Obrázek 3.8 Model PI regulátoru rychlosti	33
Obrázek 3.9 Vzorová úloha v prostředí CANoe na propojení nástrojů Simulink a CANoe	34

SEZNAM TABULEK

Tabulka 1.1 Vysvětlení zkratk u označení bitů ve struktuře CAN rámce [4], [6] 18

ÚVOD

Automobilový průmysl patří k dynamickým průmyslovým odvětvím, které se neustále vyvíjejí velmi rychlým tempem. Zatímco před několika desítkami let obsahoval osobní automobil pouze mechanické komponenty, postupem času se v autech začala objevovat elektronika a její role stále roste. Elektronika byla původně nasazována pro zlepšení jízdních vlastností, například pro řízení pohonu a bezpečnostní systémy jako je ABS. V průběhu vývoje se však začaly objevovat i systémy zvyšující komfort řidiče a pasažérů.

V dnešních automobilech lze nalézt několik desítek řídicích jednotek, které si mezi sebou musí vzájemně vyměňovat data. Při tomto množství jednotek se jedná o obrovské objemy dat, v autech se tedy nenachází pouze jedna síť, ale více, přičemž lišit se mohou i přenosové rychlosti těchto sítí. Zatímco síť komfortní elektroniky a multimediálních zařízení může pracovat při nižších rychlostech, síť zahrnující jednotky pohonu a jednotky zajišťující bezpečnostní systémy musí pracovat s vysokou přenosovou rychlostí.

Pro efektivní výměnu dat mezi řídicími jednotkami byla vynalezena sběrnice CAN, která dnes patří k nejrozšířenějším. Sběrnice CAN byla postupně vylepšována tak, aby vždy plnila zvyšující se požadavky na komunikaci jednotek.

Mechatronická laboratoř působící na Fakultě strojního inženýrství pod Ústavem mechaniky těles, mechatroniky a biomechaniky v minulosti spolupracovala na projektech se společností Škoda Auto v oblasti vývoje testovacích zařízení pro elektronické systémy. Komunikace mezi řídicími jednotkami však dříve nebyla na takové úrovni jako dnes. Dříve tedy pro účely testování stačila jedna řídicí jednotka a zařízení, které jednou za čas poslalo na sběrnici potřebnou CAN zprávu, aby jednotka měla dojem, že je namontována v reálném automobilu. Situace se však výrazně změnila a dnešní jednotky vyžadují velké množství zpráv a informací, aby nepoznaly, že nejsou namontovány v reálném autě, ale pouze v testovacím zařízení, a zbytek sběrnice není fyzicky přítomný, ale je simulovaný. Zrodil se tedy požadavek na výrobu testovacího standu, který by umožňoval pokračování ve spolupráci na průmyslových projektech.

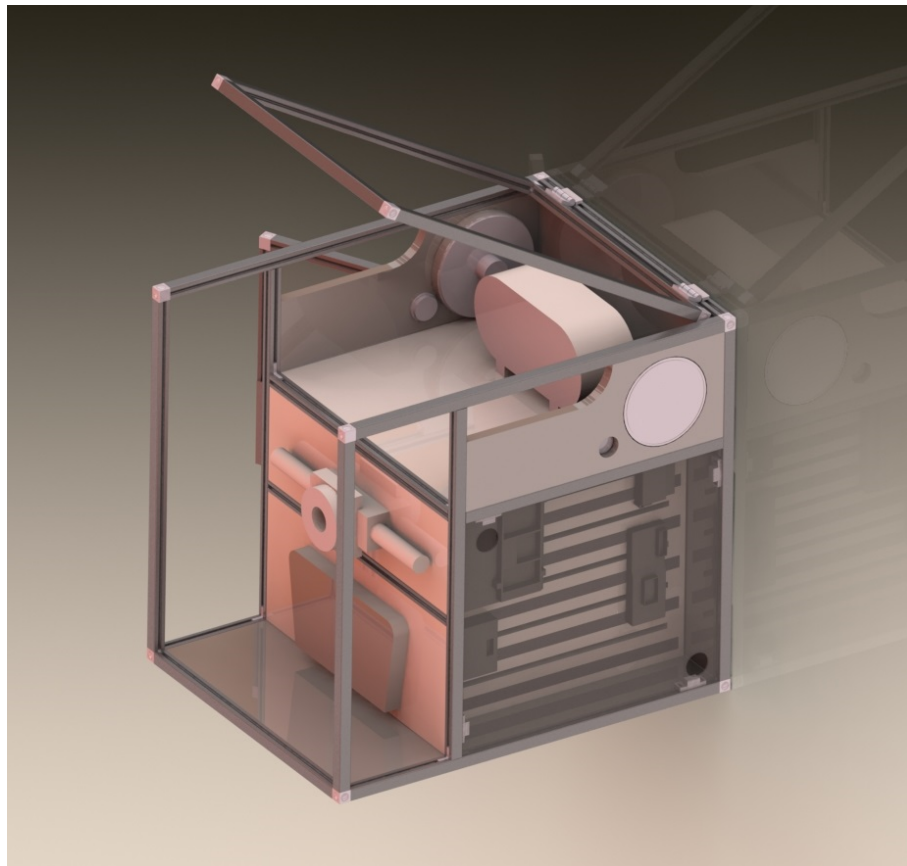
Pro sledování a řízení komunikace na sběrnici CAN je potřeba vhodný hardware a software. Mechatronická laboratoř disponuje HW rozhraním VN 1630A a softwarem CANoe od firmy Vektor. V další části práce jsou analyzovány možnosti tohoto vybavení, následuje sekce zaměřená na propojení prostředí CANoe se Simulinkem a tvorbu vzorových úloh.

1 TESTOVACÍ STAND ŠKODA

Testovací stand (v překladu stav) je zařízení, které se skládá z vybraných řídicích jednotek používaných v dnešních osobních autech koncernu Volkswagen, které jsou vzájemně propojeny a vyměňují si mezi sebou data.

1.1 Mechanická část

Před samotnou výrobou byl vytvořen model testovacího standu v softwaru SolidWorks viz obr. 1.1. Základní rám byl sestaven z hliníkových profilů firmy Bosch Rexroth, jejichž výhodou je velká modularita a možnost konstrukci rozebrat za pomoci šroubováku a šestihranného klíče. Na výplně byly použity desky z lamina a plexiskla, do kterých byly vyřezány otvory pro uchycení rádia, reproduktorů, kabelů.



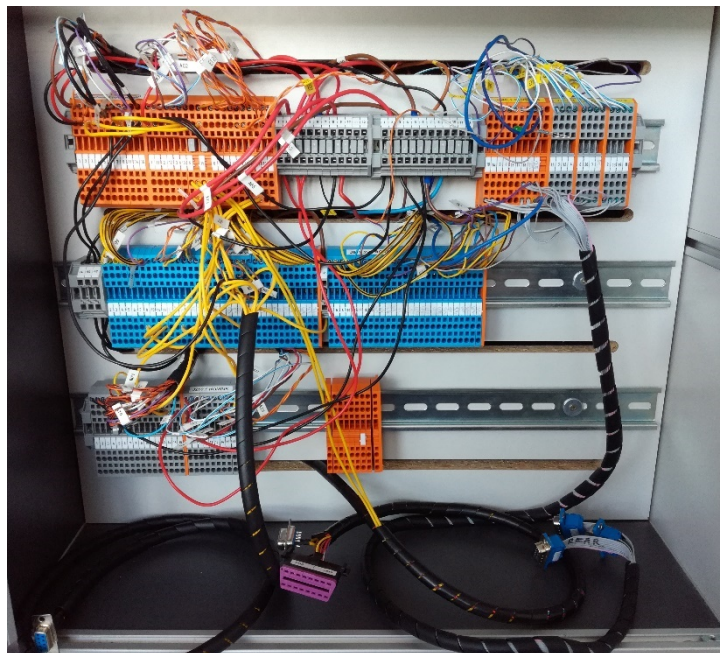
Obrázek 1.1 Model testovacího standu vytvořený v SolidWorks

Současná podoba testovacího standu je znázorněna na fotce viz obr. 1.2.



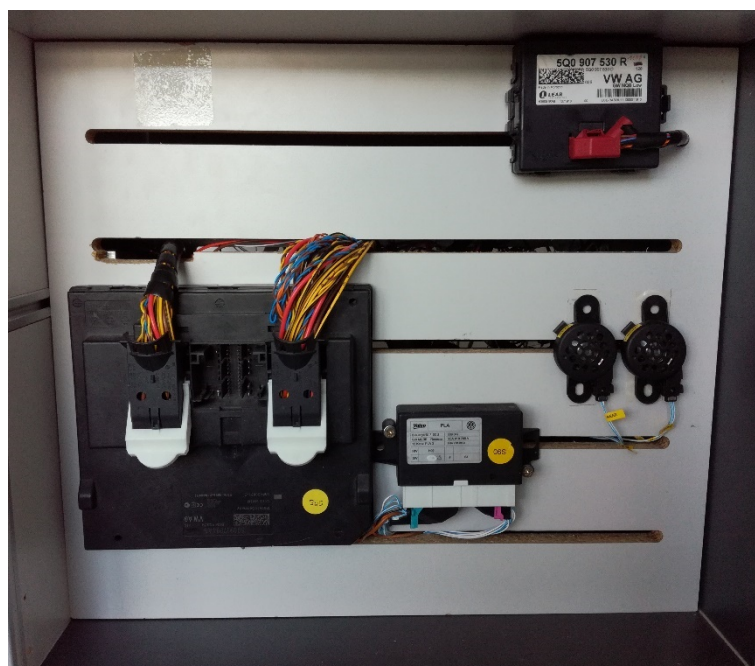
Obrázek 1.2 Testovací stand

Spodní část přední strany vyplňuje panel autorádia, nad kterým se nachází podvolantový modul, který obsahuje pravou a levou ovládací páčku sloužící pro přepínání dálkových a tlumených světel, ovládání blikáčů, ostřikovačů, tempomatu a palubního počítače. Na vodorovné desce nad podvolantovým modulem jsou připevněny budíky. Na levém boku standu se nachází otevírací dveře, po jejichž otevření jsou přístupné WAGO svorky připevněné na DIN lištách viz obr. 1.3.



Obrázek 1.3 Propojení vodičů pomocí WAGO svorkovnic na levém boku standu

Na pravém boku se pod odnímatelným průhledným krytem nachází elektronické řídicí jednotky viz obr. 1.4. Konkrétně se jedná o BCM (zkratka pro Body Control Module, centrální řídicí jednotka komfortní elektroniky), dále jednotku PDC (pomoc při parkování), respektive PLA (jednotka parkovacího asistenta) a Gateway. V případě potřeby lze jednotky rozšířit o HDSG (elektronické otevírání a zavírání pátých dveří), senzor osvětlení, intenzity deště a další. V bočních výplních na horní části standu jsou umístěny středové reproduktory.



Obrázek 1.4 Elektronické řídicí jednotky na pravé straně testovacího standu

1.2 Elektrické zapojení

Všechny vodiče propojující jednotlivé řídicí jednotky byly vyvedeny na WAGO svorky, které jsou umístěny ve 3 řadách nad sebou na DIN lištách na levém boku testovacího standu viz obr. 1.3. Jednotlivé svorky jsou očíslované a ve vytvořené dokumentaci je možné ke všem vodičům dohledat název označení signálu a informaci o tom, ze kterého pinu a jednotky, do kterého pinu a jednotky vodič vede.

Protože i přes použití CAN sběrnice vede do řídicích jednotek velké množství vodičů, bylo nutné zabývat se i otázkou kabelového uspořádání, aby elektrické vedení zůstalo přehledné. Daný svazek vodičů vedoucí od nebo k jedné řídicí jednotce byl zamotán do svazkovací spirály, díky čemuž se stalo zapojení uvnitř standu mezi řídicími jednotkami a WAGO svorkami výrazně přehlednější viz obr. 1.5.

O napájení všech řídicích jednotek se stará spínaný zdroj Mean Well SP-240-12, který je napájen střídavým napětím 230 V a na výstupu dodává stejnosměrné napětí 12 V a maximální proud 20 A. Při zapnutí zapalování je proudový odběr všech zařízení přibližně 1 A.



Obrázek 1.5 Elektrické svazky uvnitř testovacího standu

1.2.1 Sběrnice CAN

Historie

CAN je zkratka z anglického názvu Controller Area Network. Komunikace pomocí této sběrnice je sériová a byla vyvinuta firmou Robert Bosch GmbH. Důvodem pro vývoj sběrnice byla úspora vodičů. V minulosti byl totiž jeden signál přenášen jedním vodičem, což znamenalo, že se zvětšujícím se podílem elektroniky v osobních vozidlech by muselo výrazně stoupnout množství vodičů v autě, to se však negativně projevuje na ceně i hmotnosti vozidla, která se následně promítne do zvýšené spotřeby paliva. Dalším negativním dopadem by bylo velké množství spojů, které z hlediska spolehlivosti představují možné riziko. Vývoj sběrnice CAN byl zahájen v roce 1983, protokol byl uvolněn v roce 1986. Prvním automobilem, jehož elektroinstalace byla vybavena sběrnici CAN, bylo BMW řady 8 uvedené na trh v roce 1988. [1]

CAN 2.0A a CAN 2.0B

Původní specifikace sběrnice CAN 2.0A definovala 11 bitový identifikátor CAN zprávy. S rostoucím počtem elektronických řídicích jednotek a s tím souvisejícím objemem přenášených dat se však ukázalo, že 11 bitů není pro označení zprávy dostatečných. Proto firma Bosch v roce 1991 publikovala specifikaci CAN 2.0B, která kromě standardního rámce CAN zprávy (CAN 2.0A) popisuje i rozšířený formát (CAN 2.0B), u něhož je délka identifikátoru rozšířena o 18 bitů, celkem tedy disponuje 29 bity. [2]

V dnešních automobilech můžeme najít několik druhů sběrnice CAN, některé slouží

řídícím jednotkám pohonu, jiné podvozku, komfortní elektronice, informačním systémům, diagnostice. V některých případech se liší i přenosové rychlosti jednotlivých sběrnic. Pro spojení všech sběrnic a předávání zpráv z jedné sběrnice do druhé slouží tzv. gateway (v překladu brána). Sběrnice CAN kromě automobilového průmyslu našla uplatnění také v oblasti průmyslové automatizace.

Komunikace přes sběrnici CAN má 4 základní vrstvy: aplikační, objektovou, přenosovou a fyzickou. [2]

- Aplikační vrstva popisuje chování elektronické řídicí jednotky, současně poskytuje rozhraní k signálům sběrnice a ke vstupním a výstupním zařízením (senzory a aktuátory). [3]
- Objektová vrstva slouží k filtrování zpráv a přenosu vysílaných zpráv od aplikační vrstvy na přenosovou a přijímaných zpráv v obráceném sledu.
- Přenosová vrstva se zabývá detekcí chyb, ověřováním zpráv, tvorbou rámce zpráv, správným časováním a přenosovými rychlostmi.
- Fyzická vrstva definuje napěťové úrovně, způsob přenosu bitů, realizuje se prostřednictvím ní skutečný přenos dat. [2]

Ve specifikaci CAN je definována pouze přenosová a fyzická vrstva, další vrstvy komunikace mohou být popsány pomocí standardů, které však nemusí být vzájemně kompatibilní. [4]

CAN FD

Protože verze CAN 2.0 nestačila plnit stále se zvyšující požadavky na sběrnice uvnitř vozidel z hlediska objemu přenášených dat (verze CAN 2.0 má omezenou délku datového rámce na 8 bajtů) a z hlediska maximální možné délky vedení mezi 2 jednotkami (u verze CAN 2.0 činí 40 metrů), začala firma Robert Bosch GmbH v roce 2011 vyvíjet nový protokol s názvem CAN FD (FD je zkratka pro Flexible Data-Rate, tedy proměnnou komunikační rychlost). O rok později byl nový protokol firmou Bosch představen a v roce 2015 přijat za standard ISO. [5]

Hlavní výhodou CAN FD je zvětšená délka datového pole, takže již není nutné rozdělovat větší informace do více CAN zpráv. Velikost datového pole uvnitř jedné zprávy může být až 64 bajtů. Se zvětšením délky datového pole souvisí i zvětšení poměru mezi bity nesoucími informaci a ostatními bity rámce (tzv. Payload, v překladu užitečné zatížení).

Další přidanou hodnotu tvoří zvětšení přenosové rychlosti, která ve vylepšené verzi protokolu již nemusí být konstantní. Jmenovitá přenosová rychlost zůstala zachována na 1 Mb/s, nicméně v průběhu odesílání zprávy může odesílatel u datového pole přepnout na vyšší přenosovou rychlost (až 8 Mb/s, zatímco u CAN 2.0 maximální přenosová rychlost činí 2 Mb/s). Přenosová rychlost byla zvýšena pouze u datového pole z toho důvodu, aby byla zachována robustnost sběrnice CAN.

Protože protokol CAN FD je vylepšená verze protokolu CAN, podporuje CAN FD i zprávy staršího protokolu CAN 2.0, nicméně zprávy novějšího protokolu CAN FD není

možné zpracovávat na starší sběrnici CAN 2.0. Pro rozlišení mezi zprávou staršího a novějšího protokolu jsou v řídicím poli CAN FD zpráv bity FDF (FD Frame) a BRS (Bit Rate Switch, slouží k přepínání přenosových rychlostí datového pole zprávy). [5]

1.2.2 CAN zpráva

Data jsou přenášena po sběrnici CAN uvnitř rámců. Existují 4 základní typy rámce CAN zpráv: [2]

- datový rámeček (Data Frame)
- vzdálený rámeček (žádost o data, Remote Frame)
- chybový rámeček (Error Frame)
- rámeček signalizující přetížení sítě (Overload Frame)

Základní prvek komunikace tvoří datový rámeček sloužící k přenosu dat, přičemž délka zprávy není konstantní, ale může se lišit s ohledem na délku přenášených dat. V případě, že přenášený datový typ přesahuje možnosti jedné zprávy, je možné provést rozdělení do více zpráv, přičemž jejich obsah se v přijímací jednotce opět pospojuje. [2]

Sběrnice rovněž umožňuje, aby jeden uzel vyslal žádost o data jinému uzlu. Ten odpoví zprávou, která již obsahuje požadovaná data, identifikátor zprávy je stejný jako u žádosti o data.

Chybový rámeček pomáhá odhalit chyby vzniklé při přenosu. Při vysílání zprávy všechny ostatní jednotky kontrolují, zda je formát zprávy v pořádku a pokud by některý uzel sítě narazil na chybu, vyšle zprávu chybového rámce. Uzly sběrnice CAN dokáží rozlišit dočasnou a permanentní chybu.

V případě, že vytížení sběrnice překročí maximální mez, vyšle se zpráva o přetížení a dojde k navýšení časové prodlevy mezi jednotlivými zprávami.

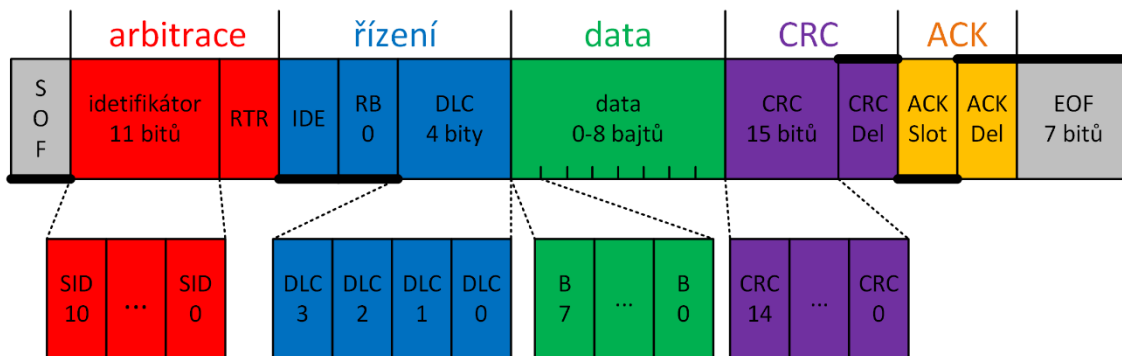
Na začátku každé zprávy je startovací bit, po kterém následuje identifikátor, jenž popisuje obsah této zprávy, poté jsou v případě datového rámce odesílána data, následují kontrolní bity a na závěr se posílá koncový bit. V případě nečinnosti je na sběrnici horní napětí, tedy logická 1, která značí recesivní bit, logická 0 reprezentuje dominantní bit.

Zprávy lze podle specifikace rozlišit na: [5]

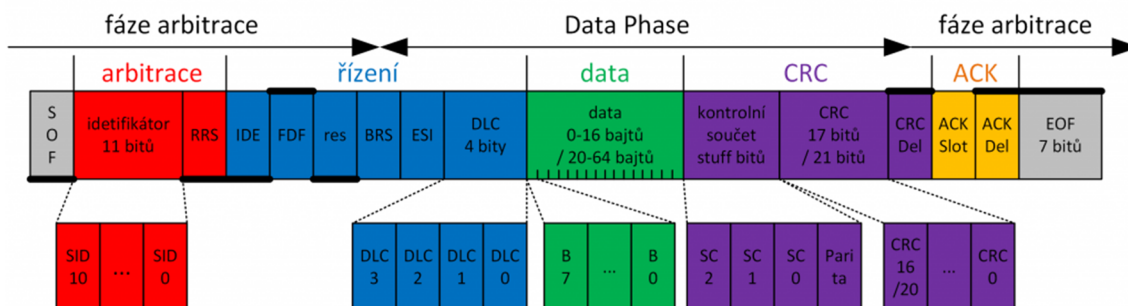
- Rámeček CAN 2.0A (standardní) s 11 bitovým identifikátorem viz obrázek 1.2
- Rámeček CAN 2.0B (rozšířený, extended) s 11 bitovým základním a 18 bitovým rozšířeným identifikátorem
- Rámeček CAN FD (standardní) s 11 bitovým identifikátorem (viz obrázek 1.3)
- Rámeček CAN FD (rozšířený, extended) s 11 bitovým základním a 18 bitovým rozšířeným identifikátorem

Zpráva není adresována jednomu konkrétnímu uzlu, ale přijmou ji všechny uzly (i jednotka, která zprávu vyslala) a pomocí filtrování se rozhodne, zda uzel danou zprávu dál zpracuje nebo ne.

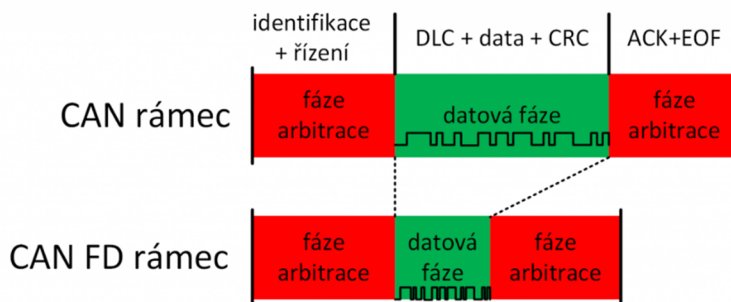
Struktura standardního rámce CAN zprávy podle specifikace CAN 2.0A je zobrazena na obr. 1.6, strukturu standardního rámce podle specifikace CAN FD znázorňuje obr. 1.7. Srovnání rámce zprávy specifikace CAN a CAN FD je provedeno na obr. 1.8. Zkratky popisující strukturu zpráv jsou popsány v tabulce tab 1.1.



Obrázek 1.6 Struktura rámce podle specifikace CAN 2.0A [5]



Obrázek 1.7 Struktura rámce podle specifikace CAN FD [5]



Obrázek 1.8 Srovnání rámce CAN a CAN FD [5]

Tabulka 1.1 Vysvětlení zkratk u označení bitů ve struktuře CAN a CAN FD rámce [4], [6]

Zkratka	Anglický význam zkratky	Český význam
SOF	Start of Frame	Začátek rámce
SID	Standard Identifier	Standardní identifikátor
RTR	Remote Transmission Request	Udává, zda se jedná o zprávu data obsahující nebo o ně žádající
IDE	Identifier Extension	Informuje, zda je použit standardní nebo rozšířený formát
FDF	Flexible Data Format	Určuje, zda se jedná o zprávu protokolu CAN nebo CAN FD
res	res bit	Rezervovaný pro budoucí protokol
BRS	Bit Rate Switch	Slouží k přepínání komunikačních rychlostí datového pole
ESI	Error Status Indicator	Indikace chybného stavu na sběrnici
RB	Reserved bit	Rezervovaný specifikací na hodnotě 0
DLC	Data Length Code bits	Počet bajtů dat
B	Data bits	Datové bity
CRC	Cyclic Redundancy Check	Cyklický redundantní součet, který slouží k detekci chyb
CRC Del	CRC Delimiter	Ukončení CRC
ACK Slot	Acknowledge Slot	Potvrzení přijímacích uzlů, že zpráva byla přijata
ACK Del	Acknowledge Delimiter	Ukončení Acknowledge Slot, hodnota rezervována jako 1
EOF	End of Frame	Ukončení rámce

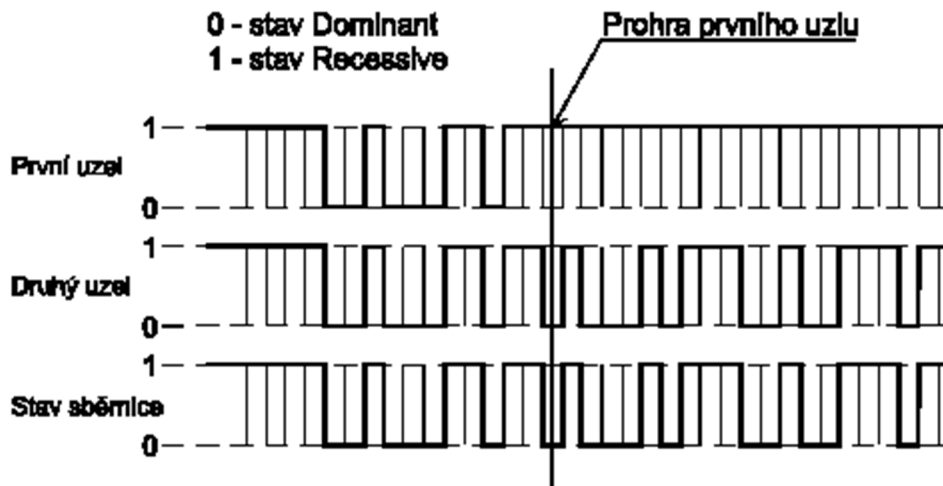
Arbitráž

V případě, že 2 uzly začnou vysílat ve stejném okamžiku, rozhodne se na základě identifikátoru, která zpráva bude dál vysílána a která své vysílání ukončí, respektive jej přeruší a zopakuje, až nebude na sběrnici vysílána žádná zpráva. Platí pravidlo, že přednost má zpráva s nižším identifikátorem. Logická 0 signalizuje dominantní bit, logická 1 ustupující (recesivní) bit. Když se tedy na sběrnici objeví 2 zprávy současně, porovnávají vysílací uzly vysílanou zprávu s přijímanou. Pokud je vysílací a přijímaný bit stejný, pokračuje uzel ve vysílání, jakmile však uzel vysílající zprávu s vyšším identifikátorem (a tedy nižší prioritou) obdrží bit obsahující logickou 0, zatímco sám vysílá logickou 1, přeruší své vysílání viz obr. 1.9. Protože spínač CAN High spíná vodič na horní napětí (obvykle 5V), CAN Low na dolní napětí (0 V) a oba jsou typu otevřený kolektor, může libovolný vysílač změnit úroveň sběrnice z recesivního bitu na dominantní. [4]

Při vysílání zprávy na sběrnici mohou nastat 3 stavy: [4]

- Jeden uzel vysílá, ostatní přijímají
- Více uzlů zahájilo vysílání ve stejný okamžik, pomocí arbitráže (porovnání bitů identifikátorů) se rozhodne, které uzly se stanou přijímači a který bude pokračovat ve vysílání

- Libovolný uzel může zahájit vysílání, protože sběrnice je volná



Obrázek 1.9 Arbitráž CAN zpráv vysílaných v jeden okamžik [4]

1.2.3 Detekce chyb

Rozlišovány jsou chyby při přijímání a odesílání CAN zpráv. Do kategorie chyb detekovaných při přijímání zpráv patří následující: [6]

- Cyclic Redundancy Check (CRC) error – Vysílač vypočítává speciální kontrolu bit po bitu od začátku rámece (SOF) až po konec datového pole (B0), CRC sekvence bitů je následně vyslána v poli rámece CRC. Přijímací uzel též počítá CRC sekvenci stejným způsobem a porovná ji s CRC sekvencí poslanou vysílačem. V případě, že se objeví neshoda, se jedná o CRC error a je vyslán rámeček typu Error frame. Vysílání zprávy vysílačem je opakováno a hodnota čítače přijatých chyb je inkrementována. Pokud čítač dosáhne mezní hodnoty, je vygenerováno přerušení.
- Bit Stuffing error (bit stuffing znamená vkládání bitů) – pokud se nachází 5 po sobě jdoucích bitů stejné polarity mezi začátkem rámece (SOF) a CRC oddělovačem (CRC Del), je vložen na následující pozici bit opačné polarity, aby nedošlo k porušení synchronizace zařízení připojených na sběrnici. V případě, že některé zařízení detekuje mezi SOF a CRC Del 6 po sobě jdoucích bitů stejné polarity, objevil se bit stuffing error a je vygenerován rámeček Error frame. Zpráva je poslána znovu, v případě této chyby není generováno přerušení.
- Invalid Message Received error – když se objeví libovolný typ chyby v průběhu přijímání zprávy, zaznamená se to do registru modulu CAN.

Dále mohou být rozpoznány 3 druhy chyb při vysílání zpráv: [6]

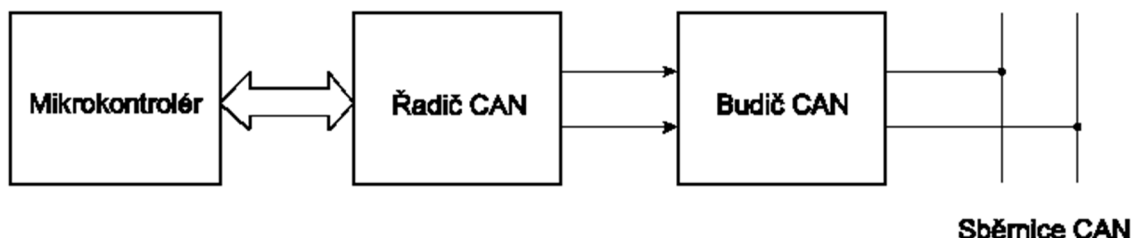
- Acknowledge error – v poli ACK Slot vysílací uzel vyše recesivní bit a sleduje, zda některý z uzlů nezměnil hodnotu bitu na dominantní, což by znamenalo, že zpráva nebyla uzlem, který hodnotu bitu změnil na dominantní, správně přijata. Pokud některý uzel změnil hodnotu bitu, znamená to, že se objevila chyba při vysílání a vysílání zprávy musí být opakováno. Rámec typu Error frame však není generovaný.
- Form error – pokud přijímač detekuje dominantní bit na pozici bitu, kde má být recesivní bit (EOF, ACK Del, CRC Del nebo v mezeře mezi rámci), znamená to, že formát zprávy není v pořádku, zpráva je odeslána znovu a je vyslán chybový rámec Error frame.
- Bit error – objeví se, pokud vysílač pošle dominantní bit, nicméně přijme zpět recesivní bit. Pokud je však v poli arbitrace vysílán recesivní bit, ale vysílač obdrží dominantní bit, nejedná se o chybu. Znamená to, že jiný uzel vysílající ve stejný okamžik má vyšší prioritu (nižší identifikátor), jedná se tedy o běžnou arbitráž.

1.2.4 Topologie sítě CAN

U sběrnice CAN se používá liniová topologie. Maximální počet uzlů připojených na sběrnici je teoreticky neomezený, prakticky se však na jednu sběrnici připojují maximálně desítky jednotek. Pro zabránění nežádoucích odrazů na konci sběrnice je nutné sběrnici ukončit připojením tzv. zakončovacích rezistorů, jejichž hodnota odporu se u jednotlivých výrobců automobilů mírně liší (obvykle 120 Ω). Vliv rušení na přenášená data je snížen díky použití dvou kroucených vodičů (twisted pair), přičemž v jednom se přenáší skutečný signál a ve druhém invertovaný. V přijímači dojde k odečtení těchto signálů, čímž je eliminováno rušení a získaný signál je roven dvojnásobku původního signálu. [4]

Každá jednotka komunikující po sběrnici CAN představuje jeden uzel sítě. Všechny uzly sítě musí obsahovat následující komponenty viz obr. 1.10: [4]

- Komunikující zařízení, obvykle mikrokontrolér
- Řadič sběrnice CAN
- Budič sběrnice CAN (vysílač a přijímač)



Obrázek 1.10 Součásti uzlu zajišťující komunikaci po CAN sběrnici [4]

2 HARDWARE A SOFTWARE VECTOR

Vector (celým názvem Vector Informatik GmbH) je německá firma se sídlem v německém Stuttgartu, jež se zabývá vývojem elektroniky do automobilů. Kromě elektroniky do osobních automobilů cílí firma také na traktory, vlaky, nákladní automobily a v neposlední řadě se zabývá také leteckým průmyslem. Portfolio produktů je možné rozdělit do několika skupin: hardware, software, embedded (v překladu vestavěné) zařízení, knihovny a ovladače. Pro účely tohoto projektu máme k dispozici hardwarové rozhraní VN1630A a software CANoe ve verzi 9.0. [7]

2.1 Analýza možností

2.1.1 Hardware Vector 1630A

Jedná se o zařízení, jež umožňuje propojit PC, na kterém běží aplikace pro analýzu a testování elektronických řídicích jednotek, prostřednictvím rozhraní USB s fyzickou sběrnici CAN či LIN. Pomocí aplikace je například možné simulovat zbývající část sběrnice a na fyzickou sběrnici odesílat adekvátní zprávy. Zařízení Interface VN 1630A je znázorněno i s připojenou CAN sběrnici na obr. 2.1.



Obrázek 2.1 Zařízení CAN/LIN Interface VN 1630A firmy Vector s připojeným CAN kanálem

Pro připojení na fyzické sběrnice je zařízení vybaveno 2 konektory D-SUB s označením DE-9 M (E značí velikost konektoru, 9 je počet pinů a M znamená, že se

jedná o zástrčku, někdy se konektory označují nesprávně jako Cannon 9 podle firmy, která je vynalezla). Zařízení disponuje 4 kanály, na jednom konektoru DE-9 M mohou být 2 CAN kanály nebo kombinace CAN a LIN kanálu. Maximální přenosová rychlost pro sběrnici CAN je 2 Mb/s, pro CAN FD 8 Mb/s a pro LIN 330 kb/s. Vedle konektorů DE-9 M se nachází Binder konektor, který lze použít pro časovou synchronizaci různých zařízení firmy Vector.

Na protější straně zařízení je třetí konektor DE-9 M pro připojení digitálních a analogových vstupních a výstupních signálů. Podrobné přiřazení pinů je k dispozici v manuálu zařízení. Součástí zařízení je v neposlední řadě konektor USB typu B, který slouží jednak pro komunikaci s PC a jednak jako napájení celého zařízení.

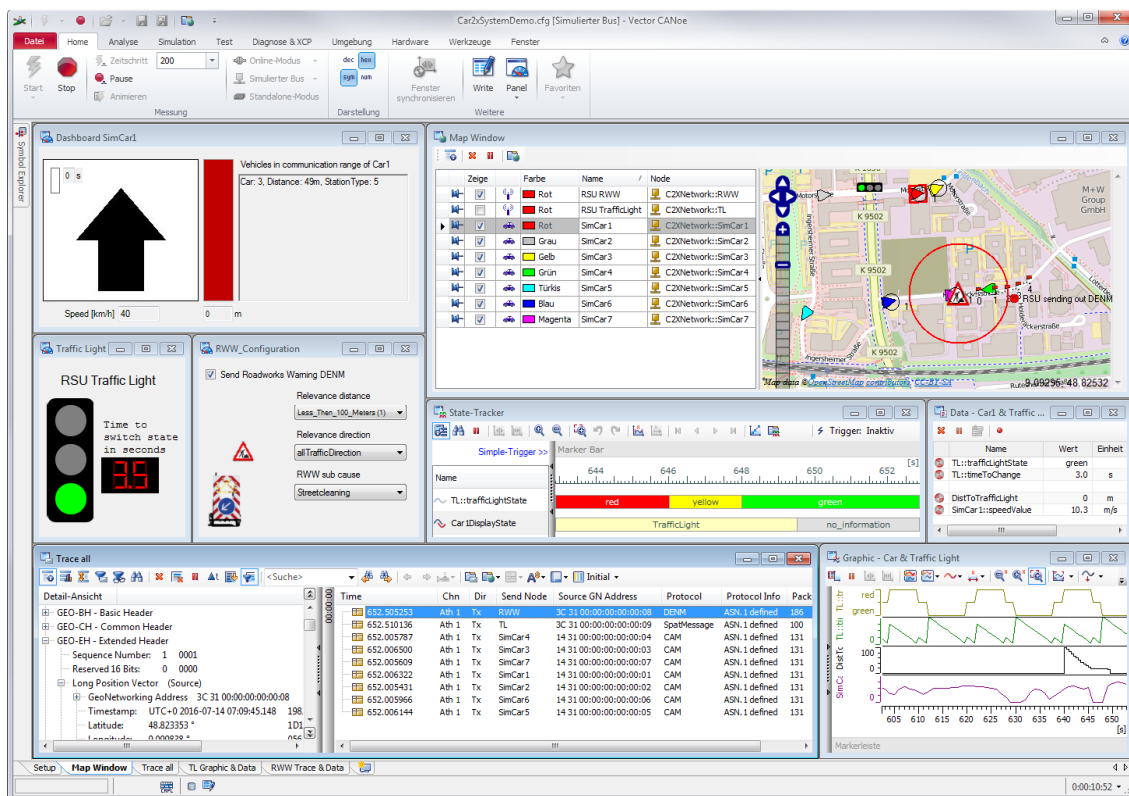
Na horní straně zařízení se nachází 5 barevných LED, které indikují aktivitu a stav sběrnice na jednotlivých kanálech. [8]

2.1.2 Software CANoe

CANoe je univerzální nástroj používaný výrobci automobilů a jejich dodavateli pro vývoj, testování a analýzu od jednotlivých řídicích jednotek až po celé sítě složené z těchto jednotek. Současně je možné CANoe použít pro simulaci zbývajících částí sběrnice, která není při testování fyzicky k dispozici. Testování je možné provádět automaticky, což nachází uplatnění především v oblasti HIL simulací.

Mezi podporované typy sběrnic patří CAN, CAN FD, LIN, MOST, FlexRay, Ethernet, WLAN a další. [9]

Prostředí CANoe je tvořeno různými panely pro zobrazování a řízení komunikace na sběrnici. V okně „Graphic“ mohou být vykresleny průběhy různých signálů v závislosti na čase, okno „Data“ nabízí přehled vybraných hodnot signálů včetně jejich jednotek a sloupcových grafů, které zobrazují současnou hodnotu vzhledem k přípustnému rozsahu hodnot. Pomocí nástroje „Panel Designer“ je možné vytvořit vlastní ovládací panel, na kterém je možné jednak zobrazovat hodnoty signálů přijímané ze sítě a jednak nastavovat hodnoty, které se mají na síť vysílat. V okně „Measurement Setup“ je zobrazena topologie sítě, která je aktuálně testovaná. Ukázka prostředí CANoe je na obr. 2.2.



Obrázek 2.2 Prostředí CANoe [10]

2.2 Propojení nástrojů CANoe a MATLAB

Každá řídicí jednotka představuje jeden uzel sítě CAN. Pro její simulaci je nutné popsat chování tohoto uzlu sítě. Nejjednodušší možností je použití programovacího jazyka CAPL (zkratka pro Communication Access Programming Language), který byl pro tyto účely přímo vytvořen firmou Vektor, přičemž důraz byl kladen na to, aby bylo možné jednak jednoduše popsat triviální požadavky jako například reakci na přijatou zprávu, nastavení hodnoty signálu, a jednak řešit komplexní problémy, u kterých chování jednotek bývá velmi komplikované a ovlivňováno celou řadou faktorů.

Syntaxe jazyka CAPL vychází z programovacího jazyka C, nicméně je zde několik zásadních odlišností. Aby byl popis chování v jazyce CAPL co nejjednodušší, jsou programy v tomto jazyce řízené událostmi jako například reakce na zprávu, uplynutí určitého času, změna signálu a další. Celé prostředí CANoe, takže i programy CAPL používají databáze, které popisují zprávy na sběrnici včetně signálů, které jsou obsaženy uvnitř zpráv, a jejich hodnot. V porovnání s jazykem C není možné v jazyce CAPL používat ukazatele, tato nevýhoda je však částečně vykompenzována možností použít asociativní pole. Stejně jako u jazyka C i zde platí, že program je vždy zkompileován, tedy převeden překladačem na strojový kód. [11]

V případě, že chování řídicí jednotky již není zcela triviální, lze využít výhod simulačního prostředí Simulinku a chování popsat pomocí modelu. Pro realizaci propojení nástrojů Simulink a CANoe je nutné nainstalovat doplněk CANoe/MATLAB Interface, jehož instalační soubory jsou dodávány spolu s instalací CANoe. Součástí

instalace je řada vzorových úloh a knihovna bloků pro Simulink s názvem CANoe I/O, která obsahuje bloky pro vstup a výstup signálů, systémových proměnných, proměnných prostředí, CAN zpráv, dále blok definující krok simulace (a mód, ve kterém model běží) a v neposlední řadě též volání CAPL programu. [3]

Pro správnou funkci modelu je nezbytné správné nastavení jeho konfigurace. Za předpokladu, že má být model kompilován, je doporučováno použít řešič s pevnou délkou kroku (Fixed-step solver). Délka kroku bývá obvykle volena v rozmezí od 1 ms do 100 ms v závislosti na velikosti a složitosti modelu. Pokud má být realizováno propojení Simulinku a CANoe v HIL módu, je nezbytné v konfiguračním okně modelu v sekci generování kódu zvolit jako systémový cílový soubor ten s názvem *cn.tlc*, což je soubor určený pro generování souborů s příponou DLL v jazyce C/C++ určených pro přímý import do konfigurace v prostředí CANoe.

Prostředí CANoe je vybaveno nástrojem „Model Viewer“, který slouží pro přímé otevírání modelů ze Simulinku v prostředí CANoe. Díky tomu si uživatel může prohlédnout strukturu celého modelu, ale i jednotlivých podsystémů, aniž by potřeboval instalaci Simulinku. Modely jsou otevírány v režimu pro čtení, není tedy možné je upravovat, nicméně lze měnit všechny parametry modelu, po otevření modelu v Model Viewer se u každého bloku modelu zobrazí odpovídající parametry. Pro správné zobrazení a popis modelu používá CANoe kromě souboru s příponou DLL vygenerovaného ze Simulinku ještě soubor s příponou ini, který je vytvořen při kompilaci a obsahuje informace o modelu. V konfiguraci uzlu v prostředí CANoe musí být zadána cesta k vygenerovanému souboru s příponou dll, k modelu pro zobrazení v „Model Viewer“ i k souboru obsahující informace o modelu s příponou ini.

Rozhraní mezi CANoe a Simulinkem může být realizováno ve 3 různých módech: Hardware-In-The-Loop (HIL) mód, Offline a Synchronizovaný mód.

Hardware-In-The-Loop (HIL) mód

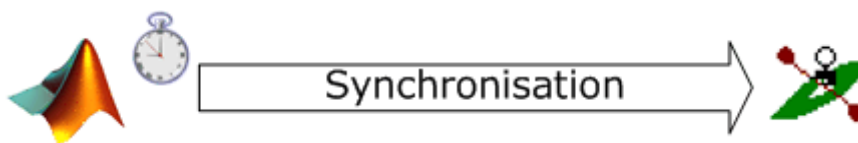
Pomocí modulu Simulink Coder lze z modelu kompilací vytvořit knihovnu DLL, která se následně přímo nahraje do prostředí simulace v CANoe. S tímto přístupem je možné ověřit funkčnost nového návrhu při spojení s fyzickým hardwarem. V jednom prostředí CANoe může být simulováno i více řídicích jednotek, může tedy být provedena simulace zbývajících částí sběrnice. Simulace je vykonávána v reálném čase. Výhodou je, že pro běh simulace v prostředí CANoe postačují knihovny vytvořené z modelů v prostředí Simulink, simulace v tomto módu tedy může běžet i na PC, kde není instalace MATLABu a Simulinku. [3]

V HIL módu je možné u modelů ze Simulinku zkompilovaných jako DLL využít tzv. parametrizaci, což znamená, že je možné měnit hodnoty všech parametrů bloků modelu ze Simulinku v prostředí CANoe, aniž by bylo nutné model znovu kompilovat. Při kompilaci modelu stačí povolit možnost parametrizace z CANoe, což automaticky vytvoří systémové proměnné pro každý parametr modelu.

Offline mód

V tomto případě simulace běží v prostředí Simulink, které pracuje v režimu Master. CANoe funguje jako Slave, simuluje CAN sběrnici a přebírá ze Simulinku údaj o čase viz obr. 2.3. Simulace neběží v reálném čase, ale je vykonávána tak rychle, jak je to možné. Tento mód je vhodný hlavně pro první fáze návrhu, protože není nezbytný reálný hardware a doba simulace je krátká. Výměna dat mezi Simulinkem a CANoe je

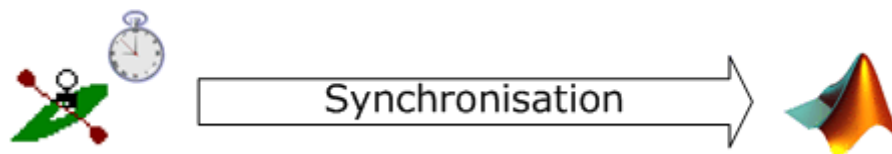
realizována prostřednictvím technologie Microsoft COM (zkratka pro Component Object Model) pro výměnu dat. [3]



Obrázek 2.3 Časová synchronizace v offline módu [3]

Synchronizovaný mód

CANoe pracuje jako Master a vysílá informaci o čase do Simulinku viz obr. 2.4. Simulace běží v reálném čase, takže je možné komunikovat s reálným hardwarem, nicméně podmínkou je, že se modely v Simulinku musí vypočítávat rychleji než v reálném čase. Výměna dat mezi prostředími je stejně jako u Offline módu možná díky technologii Microsoft COM. [3]



Obrázek 2.4 Časová synchronizace v synchronizovaném módu [3]

3 VZOROVÉ ÚLOHY

Nedílnou součástí práce je i vytvoření dvou testovacích úloh, přičemž první je realizována pouze v prostředí CANoe, zatímco druhá je zaměřena na propojení nástrojů CANoe a Simulink, využívá tedy jak prostředí CANoe, tak Simulink.

3.1 Vzorová úloha v software CANoe

Cílem vzorové úlohy je vytvoření jednoduché simulace jízdy v prostředí CANoe. Pomocí vytvořeného grafického panelu může uživatel zadat například otáčky a teplotu motoru nebo aktivovat směrová světla.

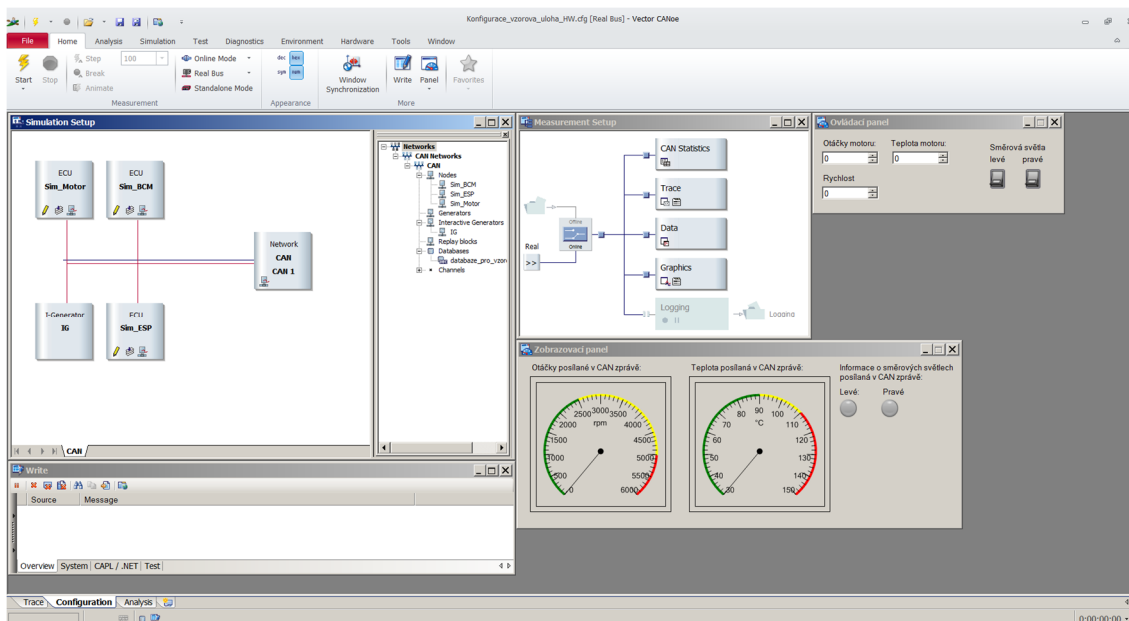
Postup vytvoření vzorové úlohy:

- Nejprve je nutné založit novou konfiguraci, použita byla šablona pro sběrnici CAN o 1 kanále a rychlosti komunikace 500 kBaud/s.
- Následovalo vytvoření všech systémových proměnných, do kterých se budou zapisovat hodnoty z uživatelského rozhraní. Přístup k systémovým proměnným je přes ikonu „System Variables“ na kartě „Environment“. Pro zadání otáček byla vytvořena systémová proměnná s názvem *Vstupni_hodnoty* a podnázvem *Otacky_vstup*. Počáteční hodnota nastavena na 0, minimum 0 a maximum 6000. Do pole Unit uvedena jednotka rpm.
- Dalším bodem je vytvoření uživatelského panelu. Pro tyto účely dobře slouží nástroj „Panel Designer“, který je přístupný na kartě „Home“ po kliknutí na „Panel“ a následně „New Panel“. Nejprve byl vložen statický text obsahující popis (Otáčky motoru), pod tento text bylo umístěno pole pro vstup dat. Ve vlastnostech je nutné nastavit, aby se hodnota z tohoto pole zapisovala do systémové proměnné *Vstupni_hodnoty::Otacky_vstup*. Grafický panel byl následně uložen jako soubor s příponou *xvp*.
- Pro popis zpráv posílaných po sběrnici CAN, která je právě simulovaná, používá CANoe databázi s příponou *dbc*. Součástí instalace je nástroj „CANdb++ Editor“ (přístupný na kartě „Tools“), jenž umožňuje vytváření a editaci databází CAN zpráv. Pomocí tohoto nástroje byla vytvořena nová databáze, jež vychází ze šablony *CANTemplate.dbc* a byla pojmenována jako *databaze_pro_vzorovou_ulohu.dbc*. Následovalo vytvoření CAN zprávy s názvem *Motor*, která obsahuje signál *Otacky* nesoucí informaci o otáčkách. Při vytváření signálu je nezbytné zadat, jakých hodnot může signál nabývat a délku signálu v bitech (pro signál otáčky byla zadána délka 16 bitů, signál tedy může nabývat hodnot od 0 do 65535 rpm). Současně je nezbytné definovat, který uzel bude tuto zprávu vysílat. Byl tedy vytvořen uzel s názvem *vstup_dat*, na záložce „Tx Messages“ je tomuto uzlu přiřazeno právo vysílat zprávu s názvem *Motor*. Vytvořená databáze musí být uložena a nahrána do konfigurace, což se provede tak, že pravým tlačítkem je kliknuto na „Databases“ v okně „Simulation Setup“, poté na „Add“ a následně je zadána cesta databáze.

- Aby se hodnota otáček zadaná od uživatele posílala po sběrnici CAN, je nutné vytvořit nový uzel simulované sběrnice, což se provede v okně „Simulation Setup“ kliknutím pravým tlačítkem myši na vedení sběrnice a volbou „Insert Network Node“. Po vytvoření uzlu je nutné popsat chování pomocí jazyka CAPL. Kliknutím na tlačítko „Edit“ na samotném uzlu (ikona znázorňuje tužku) dojde k otevření „Vector CAPL Browser“, nicméně při prvním otevření software požaduje zadání názvu uzlu, aby bylo možné uzel uložit jako soubor s příponou can. Uzel byl pojmenován jako *vstup_dat*. V jazyce CAPL bylo popsáno, aby se v okamžiku, kdy dojde ke změně systémové proměnné s názvem *Vstupni_hodnoty::Otacky_vstup*, změnila i hodnota signálu *Otacky* ve zprávě *Motor*, přičemž aktuální hodnota ve zprávě bude shodná s hodnotou zadanou uživatelem. Po napsání programu je potřeba propojit tento uzel s uzlem vytvořeným v databázi, který má oprávnění pro posílání zprávy, což se provede kliknutím pravým tlačítkem na uzel v okně „Simulation Setup“ a výběrem *databaze_pro_vzorovou_ulohu::vstup_dat* v rozbalovací nabídce „Network node“.
- Posledním úkolem je vytvoření grafického panelu, který bude zobrazovat hodnotu otáček posílanou ve zprávě *Motor* na sběrnici CAN. Postup je podobný jako při vytváření panelu s názvem „Ovládací panel“ určeným pro vstup dat od uživatele. Opět byl využit integrovaný nástroj „Panel Designer“. V horní části panelu byl umístěn statický text pro popis a pod ním se nachází měřák, v jehož vlastnostech bylo nastaveno, aby zobrazoval hodnotu signálu *Otacky* v CAN zprávě *Motor*.

Stejným způsobem byl panel pro vstup dat od uživatele rozšířen o přepínače ovládající směrová světla a pole pro zadání teploty motoru. Vzhled konfigurace vzorové úlohy v prostředí CANoe je znázorněn na obr. 3.1.

Pro zobrazení nastavených otáček i na budících musí před spuštěním simulace být na kartě „Home“ nastaveno, že sběrnice CAN je reálná. Dále je nutné ověřit, že je v konfiguraci použita správná databáze zpráv pro daný HW a poté je možné simulaci spustit. Uživatelem zadaná rychlost prostřednictvím ovládacího panelu je zobrazena v reálném čase jednak na grafickém zobrazovacím panelu aplikace a jednak na budících testovacího standu.



Obrázek 3.1 Vzorová úloha v prostředí CANoe

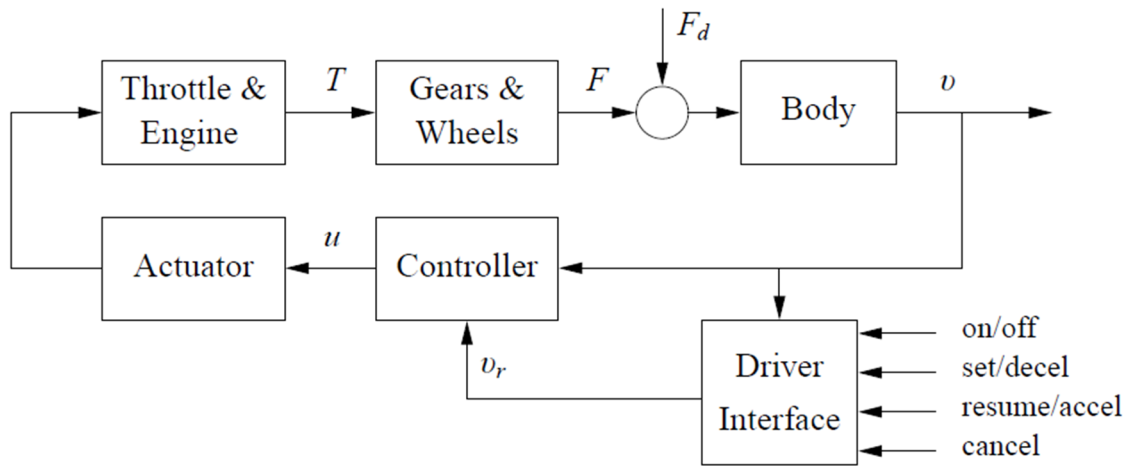
3.2 Vzorová úloha na propojení nástrojů CANoe a Simulink

Dalším úkolem bylo vytvoření vzorové úlohy, která by využívala propojení obou prostředí. Úloha byla zaměřena na regulaci rychlosti při jízdě. Nejprve byl vytvořen model pouze v prostředí Simulink, který zahrnoval model převodovky, model jednotky i regulátor, následně došlo k rozdělení na 3 samostatné modely (jednotka převodovky, regulátor a model převodovky), aby bylo možné simulovat jejich chování na CAN sběrnici pomocí CANoe.

3.2.1 Regulační smyčka pro regulaci rychlosti

Schema regulační smyčky je uvedeno na obr. 3.2.

Na základě odchylky skutečné rychlosti v a požadované rychlosti v_r regulátor (Controller) určí regulační odchylku a jí odpovídající akční zásah u . Ten je přiveden na aktuátor, kterým je v tomto případě vstřikování spalovacího motoru. Na výstupu motoru se objeví kroutící moment T , který je přenášen převodovkou na kola automobilu. Síla vyvinutá motorem F je zmenšená o ztráty F_d (tření, aerodynamickým odporem) a síly vzniklé vlivem profilu cesty (jízda do kopce či z kopce). Moment přivedený na kola automobilu mu udělí rychlost v , která je měřena a přivedena na začátek regulační smyčky do regulátoru. Jedná se o regulaci se zápornou zpětnou vazbou.



Obrázek 3.2 Blokový diagram regulace rychlosti

3.2.2 Matematický popis jízdy

Pro jednoduchost v tomto příkladu uvažujeme jízdu po vodorovné ploše. Parametry modelu jsou uvedeny v závorkách na koncích řádků popisujících označení veličin.

Moment motoru jako funkce úhlové rychlosti je popsán rovnicí: [12]

$$T(\omega) = T_m \cdot \left(1 - \beta \left(\frac{\omega}{\omega_m} - 1\right)^2\right), \quad (3.1)$$

kde $T(\omega)$ představuje moment motoru v závislosti na jeho úhlové rychlosti,

T_m označuje maximální moment motoru ($T_m = 150 \text{ Nm}$),

β je koeficient motoru ($\beta = 0.4$),

ω_m je úhlová rychlost, při které je moment motoru maximální ($\omega_m = 280 \frac{\text{rad}}{\text{s}}$) a

ω reprezentuje úhlovou rychlost motoru.

Pro převod rychlosti auta na úhlovou rychlost platí: [12]

$$\omega = i_{st} \cdot i \cdot v \quad (3.2)$$

Vysvětlení všech parametrů rovnic 3.2 – 3.5 je uvedeno za výslednou rovnicí 3.6.

Síla vyvinutá motorem je definovaná rovnicí: [12]

$$F = i_{st} \cdot i \cdot u \cdot T(\omega) \quad (3.3)$$

Ztráty vlivem valivého tření popisuje vztah: [12]

$$F_r = m \cdot g \cdot f \cdot \text{sign}(v) \quad (3.4)$$

Pro aerodynamické ztráty platí rovnice: [12]

$$F_a = \frac{1}{2} \cdot \rho \cdot C \cdot A \cdot v^2 \quad (3.5)$$

Výsledná rovnice popisující model vozidla: [12]

$$m \cdot \frac{dv}{dt} = i_{st} \cdot i \cdot u \cdot T(\omega) - m \cdot g \cdot f \cdot \text{sign}(v) - \frac{1}{2} \cdot \rho \cdot C \cdot A \cdot v^2, \quad (3.6)$$

kde m reprezentuje hmotnost vozidla ($m = 1900$ kg),

v představuje rychlost auta,

i_{st} reprezentuje stálý převodový poměr ($i_{st} = 3,2$),

i symbolizuje převodové poměry jednotlivých stupňů převodovky

$$(i = 2,8; 2,15; 1,6; 1,2; 0,8),$$

u reprezentuje akční zásah ($u = 0 \div 1$),

$T(\omega)$ je označení pro moment motoru v závislosti na otáčkách,

ω je úhlová rychlost motoru,

g označuje tíhové zrychlení ($g = 9,81 \frac{\text{m}}{\text{s}^2}$),

f je označení pro součinitel tření ($f = 0,1$),

$\text{sign}(v)$ určuje znaménko rychlosti, a tedy i směr působení třecí síly,

ρ symbolizuje hustotu vzduchu ($\rho = 1,3 \frac{\text{kg}}{\text{m}^3}$),

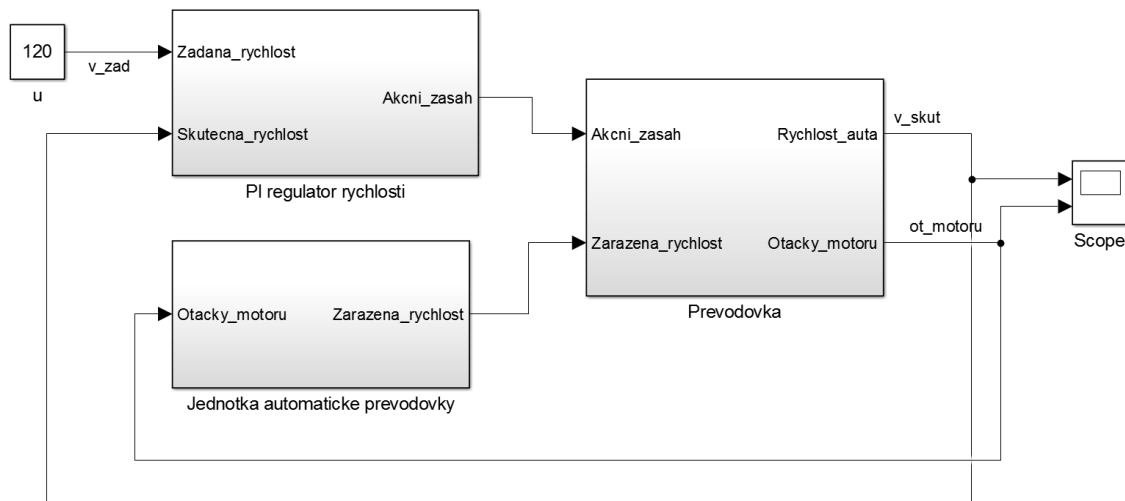
C značí součinitel odporu vzduchu ($C = 0,64$) a

A je velikost čelní plochy vozidla ($A = 2,4 \text{ m}^2$).

3.2.3 Popis modelů v Simulinku

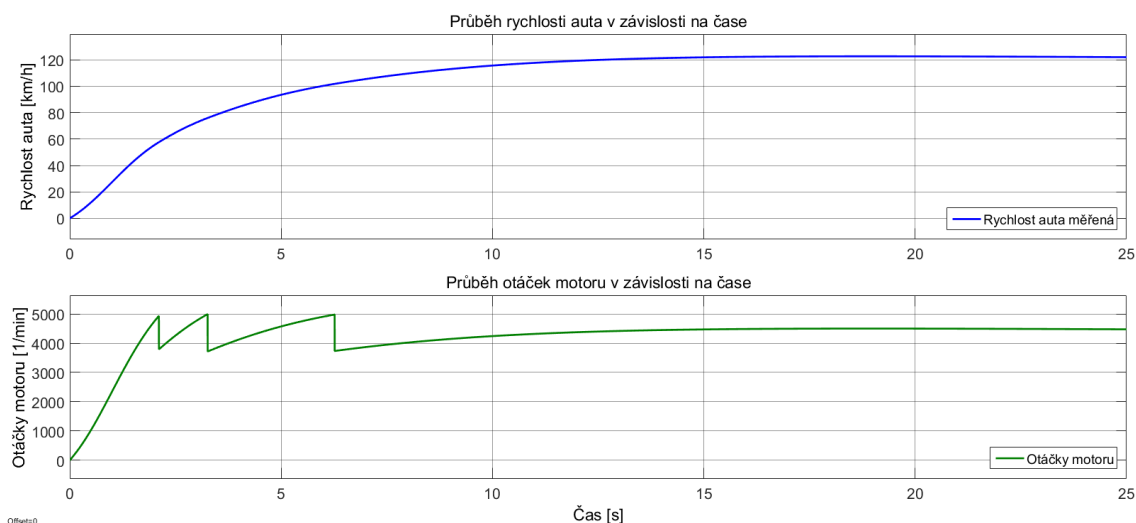
Model celého systému

Nejprve byl vytvořen model celého systému v Simulinku. Aby byla eliminována statická odchylka rychlosti, byl zvolen PI regulátor, který byl naladěn pro regulaci rychlosti. Zesílení proporcionální složky regulátoru bylo nastaveno na $K_p = 0,4$ a integrační složky na $K_i = 0,1$. Model je navržen pro rozjezd automobilu z počáteční nulové rychlosti na požadovanou rychlost (120 km/h). Schema modelu v Simulinku je znázorněno na obr. 3.3.



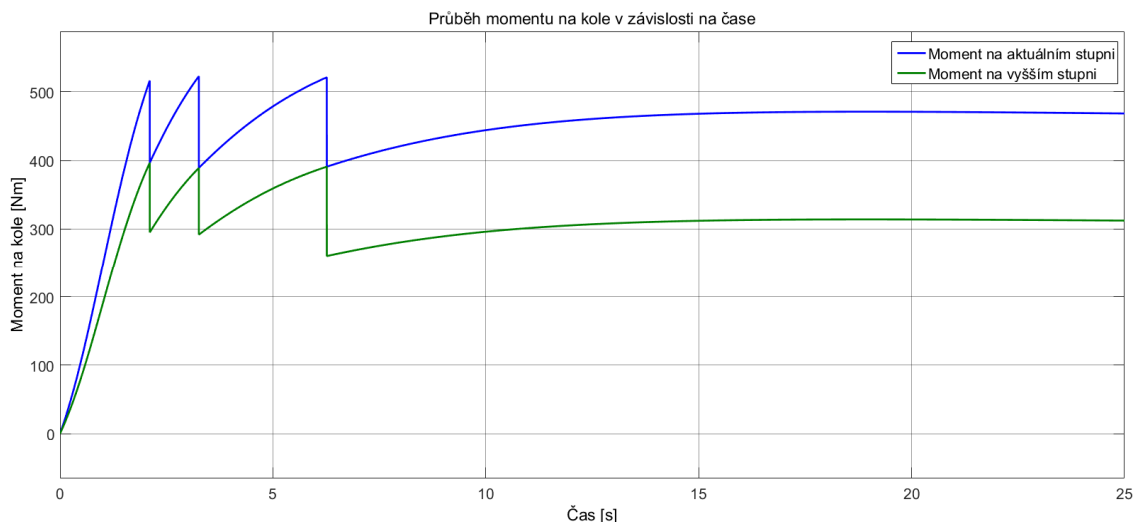
Obrázek 3.3 Schéma modelu celého systému v Simulinku

Průběh rychlosti a otáček motoru v závislosti na čase při rozjezdu z nulové rychlosti na rychlost 120 km/h je znázorněn na obr. 3.4. Skokový pokles otáček signalizuje přeřazení na vyšší převodový stupeň. Požadované rychlosti je dosaženo na 4. převodový stupeň při otáčkách 4400 1/min.



Obrázek 3.4 Průběh rychlosti a otáček motoru v závislosti na čase

Rozhodování o přeřazení na vyšší převodový stupeň je realizováno uvnitř bloku s názvem *Jednotka automaticke prevodovky*. V každém kroku je porovnáván moment na kole při aktuálním zařazeném stupni a při stupni o jeden vyšším. V případě, že moment na kole při vyšším převodovém stupni vychází větší, dojde k přeřazení na vyšší rychlostní stupeň. Pomocí tohoto rozhodovacího pravidla je dosaženo toho, že v každém okamžiku rozjezdu je na kole maximální možný moment. Průběh momentu na kole při aktuálním a o jeden stupni vyšším převodovém stupni je znázorněn v grafu na obr. 3.5.

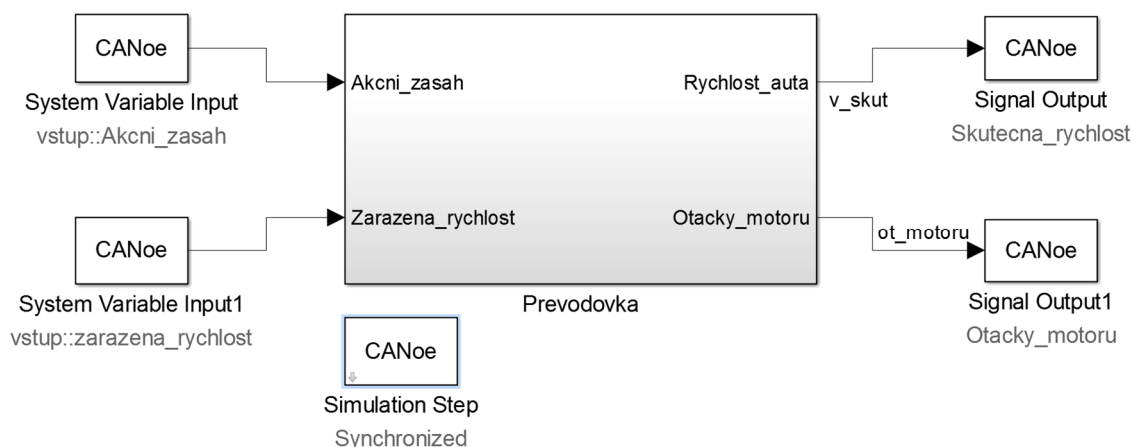


Obrázek 3.5 Průběh momentu na kole v závislosti na čase

Aby bylo možné provést propojení nástrojů Simulink a CANoe, byl model systému rozdělen na 3 samostatné celky: regulátor rychlosti, jednotku automatické převodovky a převodovku samotnou.

Model převodovky (simulovaný reálný systém)

Schéma modelu je uvedeno na obr. 3.6. Na vstup je systémovou proměnnou z CANoe přivedena informace o akčním zásahu a zařazené rychlosti. Chování modelu převodovky je popsáno rovnicí 3.6 viz kapitola 3.2.2. Výstupem převodovky je okamžitá rychlost a otáčky motoru, které se zapíší do signálů *Skutecna_rychlost* a *Otacky_motoru* vysílaných uvnitř CAN zprávy. Blok „Simulation Step“ definuje mód simulace jako synchronizovaný, aby bylo možné pracovat s HW v reálném čase.

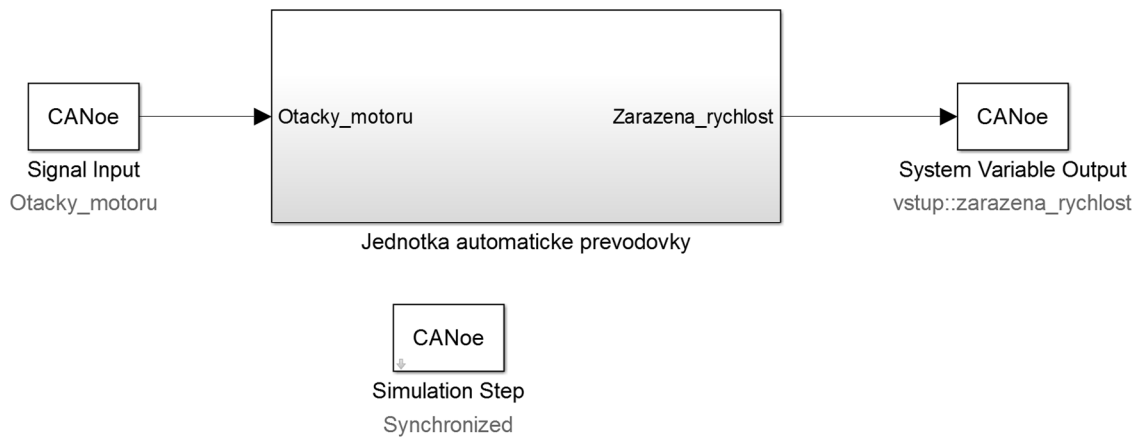


Obrázek 3.6 Model převodovky (Plant)

Model řídicí jednotky automatické převodovky

Hlavním úkolem jednotky je rozhodování, za jakých okolností má dojít k přeřazení na

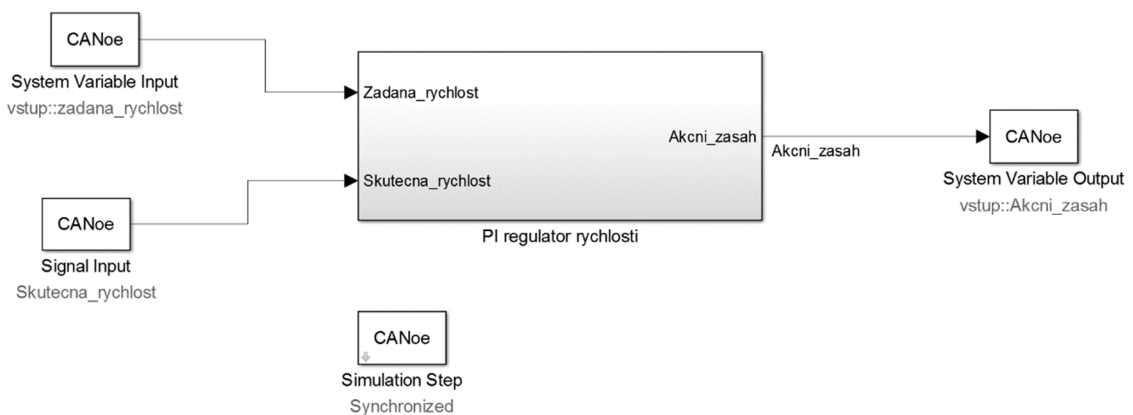
vyšší převodový stupeň. Model řídicí jednotky je znázorněn na obr. 3.7. Vstup je tvořen signálem *Otacky_motoru*, který je získáván v prostředí CANoe z CAN zprávy posílané modelem převodovky po sběrnici. Jednotka porovná moment motoru pro dané otáčky pro různé převodové stupně a o průběhu řazení rozhoduje takovým způsobem, aby moment na kole auta byl vždy maximální. Výstupem modelu je zařazená rychlost, která je ukládána do systémové proměnné prostředí CANoe pod názvem *zarazena_rychlost*. V bloku „Simulation Step“ je nastaven synchronizovaný mód simulace.



Obrázek 3.7 Model řídicí jednotky automatické převodovky

Model PI integrátoru rychlosti

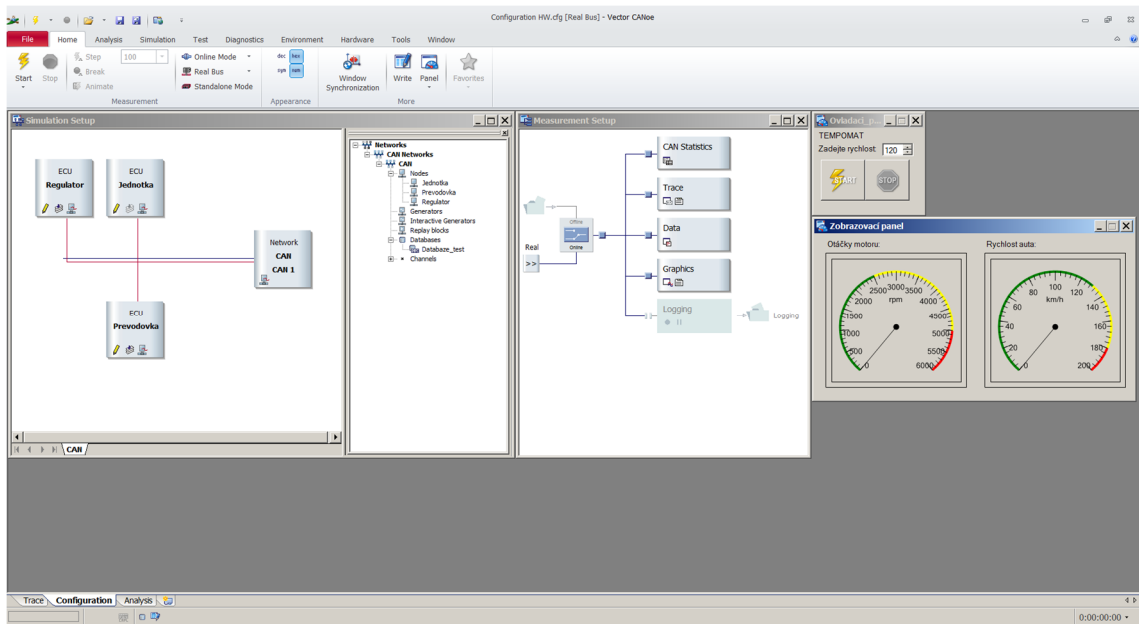
Schema modelu PI regulátoru s vyznačením vstupních a výstupních veličin je uvedeno na obr. 3.8. V prostředí CANoe se pomocí uživatelského rozhraní zadá požadovaná rychlost pro regulaci, která je následně uložena do systémové proměnné s názvem *zadana_rychlost*. Model PI regulátoru v Simulinku tuto hodnotu vyčítá ze systémové proměnné a společně s ní je na vstup přiveden ještě signál *Skutecna_rychlost*, který je získán ze zprávy vysílané na CAN sběrnici a obsahuje informaci o aktuální rychlosti auta. Výstupem bloku je akční zásah pro model převodovky, který je uložen do systémové proměnné v prostředí CANoe. Mód simulace je i v tomto případě synchronizovaný.



Obrázek 3.8 Model PI regulátoru rychlosti

3.2.4 Popis simulace v CANoe

V prostředí CANoe byly vytvořeny 3 uzly: *Regulator*, *Jednotka* a *Prevodovka* viz obr. 3.9. Jejich chování je popsáno modely v Simulinku, v konfiguracích uzlů tedy bylo nutné zadat cestu k těmto modelům. Simulace se spouští z prostředí CANoe pomocí panelu s názvem „Ovladaci_panel“, který slouží pro interakci s uživatelem.. Před spuštěním simulace uživatel zadá na grafickém panelu požadovanou rychlost a po spuštění model začne simulovat rozjezd auta. Rychlost je vykreslována v reálném čase v okně „Graphic“ na záložce „Trace“.



Obrázek 3.9 Vzorová úloha v prostředí CANoe na propojení nástrojů Simulink a CANoe

4 ZÁVĚR

Prvním úkolem zadání této bakalářské práce bylo dokončení mechanických úprav testovacího standu a tvorba dokumentace. Elektrické zapojení bylo realizováno tak, aby všechny vodiče byly vyvedeny na WAGO svorky umístěné na levém boku standu. Každá svorka má své označení a ve vytvořeném souboru v Excelu lze snadno dohledat, jaký signál se na dané svorce nachází a mezi kterými jednotkami je tento signál přenášen. Protože informace o signálech a zapojení jsou utajené, není možné zveřejnit podobu vytvořené dokumentace. Ve druhé části první kapitoly jsou popsány základy komunikace po sběrnici CAN pro správné pochopení následujícího textu.

Kapitola 2 se zabývá možnostmi získanými použitím HW a SW od firmy Vector. Na trhu existují i obdobná zařízení jiných výrobců sloužící ke stejným účelům, jako například CAN Interface od firmy PEAK System, Accurate Technologies či National Instruments, nicméně zařízení firmy Vector patří k nejpoužívanějším. Následuje podkapitola zaměřená na propojení simulačních nástrojů CANoe a Simulink.

Nedílnou součástí této práce je vytvoření vzorových úloh, což je popsáno v kapitole 3. Vytvořena byla úloha pro prostředí CANoe i pro spojení nástrojů Simulink a CANoe.

Vzorová úloha pro prostředí CANoe obsahuje ovládací panel, pomocí kterého může uživatel zadat hodnotu teploty, otáček nebo ovládat směrová světla. Informaci se posílají jako signály uvnitř zprávy a následně jsou vykresleny na grafickém panelu. Úloha byla rozšířena o možnost propojení s reálným HW, díky čemuž je současně možné zobrazovat hodnoty i na budících z auta. Součástí této práce je podrobný návod k vytvoření této vzorové úlohy.

Vzorová úloha na propojení prostředí CANoe a Simulink si klade za cíl regulaci rychlosti pomocí PI regulátoru. Vstupem od uživatele do prostředí CANoe je požadovaná rychlost. Modely vytvořené v Simulinku jsou vytvořeny v reálném čase a na výstupu je okamžitá rychlost auta a otáčky motoru. V prostředí CANoe byly vykresleny průběhy rychlosti vozidla a otáček motoru v závislosti na otáčkách a výsledek je stejný s průběhy získanými ze simulace v Simulinku. Výhodou simulace v prostředí CANoe je možnost pracovat s hardwarem v reálném čase, okamžitá rychlost a otáčky motoru při zrychlování byly tedy vykreslovány na budících testovacího standu

LITERATURA

- [1] CAN in Automation (CiA): *History of the CAN technology* [online]. [cit. 2017-05-23]. Dostupné z: <https://www.can-cia.org/can-knowledge/can/can-history/>
- [2] Robert Bosch GmbH, *CAN Specification: Version 2.0* [online]. Stuttgart: 1991 [cit. 2017-05-05]. Dostupné z: http://www.bosch-semiconductors.de/media/ubk_semiconductors/pdf_1/canliteratur/can2spec.pdf
- [3] SCHWAGER, Mark a Jochen NEUFFER, *Using MATLAB with CANoe* [online]. 2017 [cit. 2017-05-11]. Dostupné z: https://vector.com/portal/medien/cmc/application_notes/AN-IND-1-007_Using_MATLAB_with_CANoe.pdf
- [4] SEMENEC, Pavel, Lubomír NOVÁK. Funkce protokolu CAN [online]. 2002 [cit. 2017-05-21]. Dostupné z: http://simple_can_node.sweb.cz/funkce.htm
- [5] *CAN FD - nová verze CAN protokolu* [online]. 2015 [cit. 2017-05-13]. Dostupné z: <http://www.machsystems.cz/novinky/2015/can-fd-nova-verze-can-protokolu>
- [6] *Section 23. CAN module.* 2007 [online]. [cit. 2017-05-14] Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/70070D.pdf>
- [7] *Vector Company Profile* [online]. [cit. 2017-05-20]. Dostupné z: https://vector.com/vi_company_en.html
- [8] *VNI600 Network Interface for CAN, LIN, K-Line, J1708 and IO* [online]. [cit. 2017-05-20]. Dostupné z: https://vector.com/vi_vn1600_en.html
- [9] *CANoe – ECU Development & Test* [online]. [cit. 2017-05-20]. Dostupné z: https://vector.com/vi_canoe_en.html
- [10] *Lösungen für Car2x / V2X* [online]. [cit. 2017-05-20]. Dostupné z: https://vector.com/vi_car2x_solutions_de.html
- [11] LOBMEYER, Marc a Roman Marktl. *Tips and tricks for the use with CAPL* [online]. 2014 [cit. 2017-03-11]. Dostupné z: https://vector.com/portal/medien/cmc/press/PND/CAPL_1_CANNewsletter_201406_Press Article_EN.pdf
- [12] ASTRÖM, K. J. a R. M. MURRAY. *Feedback Systems* [online]. 2009 [cit. 2017-05-23]. Dostupné z: http://www.cds.caltech.edu/~murray/books/AM05/pdf/am08-complete_22Feb09.pdf