



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

## ÚSTAV AUTOMOBILNÍHO A DOPRAVNÍHO INŽENÝRSTVÍ

INSTITUTE OF AUTOMOTIVE ENGINEERING

# AUTONOMNÍ MECHATRONICKÝ SYSTÉM PRO UZAVÍRÁNÍ DIFERENCIÁLŮ VOZIDEL

AUTONOMOUS MECHATRONIC SYSTEM FOR DIFFERENTIAL LOCKING OF VEHICLES

## HABILITAČNÍ PRÁCE

HABILITATION THESIS

## AUTOR PRÁCE

AUTHOR

Ing. Pavel Kučera, Ph.D.

BRNO 2018

## ABSTRAKT

Habilitační práce se zabývá vývojem autonomního mechatronického systému pro uzavírání diferenciálů vozidel. Tento systém je určen pro předcházení uvíznutí vozidla v nepříznivých adhezních podmínkách. V práci je popsán vývoj řídicího algoritmu, řídicí jednotky - ECU, diagnostického a implementačního softwaru. Dále je zde popsáno Model in the Loop - MIL testování řídicího algoritmu, Hardware in the Loop - HIL testování na prototypovém vozidle a celkové testování vyvíjeného mechatronického systému pro uzavírání diferenciálů. V neposlední řadě je popsáno programování a implementace dotykového displeje pro komunikaci s řídicí jednotkou.

## KLÍČOVÁ SLOVA

Autonomní mechatronický systém, vozidlo, diferenciál, MIL, HIL, jazyk C/C++, ECU, procesor STM32F4xx, Simulink, NI VeriStand, Atollic TrueSTUDIO, STM32 Cube MX

## ABSTRACT

This thesis deals with the development of an autonomous mechatronic system for locking of vehicle differentials. This system is designed to prevent the vehicle from being jammed under adverse adhesion conditions. The thesis describes the development of the control algorithm, the electronic control unit – ECU and the diagnostic and the implementation software. In addition, the Model in the Loop - MIL testing of the control algorithm, Hardware in the Loop - HIL testing on the prototype vehicle and the overall testing of the mechatronic system for differential locking are described. Last but not least, the programming and implementation of the touch screen for communication with the electronic control unit is described.

## KEYWORDS

Autonomous mechatronic system, vehicle, differential, MIL, HIL, language C/C++, ECU, processor STM32F4xx, Simulink, NI VeriStand, Atollic TrueSTUDIO, STM32 Cube MX

## **BIBLIOGRAFICKÁ CITACE**

KUČERA, P. Autonomní mechatronický systém pro uzavírání diferenciálu vozidel. Brno, 2018. Habilitační práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automobilního a dopravního inženýrství. 95 s.

## ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně a s použitím literatury uvedené v seznamu.

V Brně dne 22. říjen 2018

.....

Pavel Kučera

## PODĚKOVÁNÍ

Tímto bych chtěl poděkovat panu prof. Ing. Václavu Píštěkovi, DrSc. za hodnotné připomínky a společnosti Tatra Truck a.s. za spolupráci při vývoji a testování mechatronického systému.

## OBSAH

Úvod .....	8
Cíle práce .....	10
1 Současný stav řešené problematiky .....	11
2 Vývoj řídicího algoritmu .....	13
2.1 Základní funkce .....	13
2.2 Příprava projektu .....	14
2.3 Vstupy .....	15
2.4 CAN komunikace .....	19
2.5 SPI komunikace .....	23
2.6 Hlavní smyčka .....	24
2.7 Chyby a limity .....	31
2.8 Výstupy .....	33
2.9 Bootloader .....	33
3 Vývoj výpočtových modelů pro MIL testování .....	35
3.1 Knihovna ECU .....	36
3.2 Knihovna funkcí .....	36
3.3 Knihovna Multibody .....	36
3.4 Knihovna sensorů .....	36
3.5 Technická knihovna .....	36
3.6 Knihovna pro vozidla .....	37
4 Model in the Loop - MIL testování .....	39
4.1 Vozidlo 4x4 .....	39
4.2 Vozidlo 6x6 .....	40
4.3 Vozidlo 8x8 .....	40
4.4 Simulace .....	41
5 Hardware in the Loop - HIL testování .....	46
5.1 Hardware .....	47
5.2 Software .....	48
5.3 Interface .....	49
5.4 Prototypy vozidel .....	50
5.5 Real-Time testování .....	53
6 Vývoj prototypu ECU .....	59
6.1 Procesor .....	61
6.2 Digitální vstupy .....	62
6.3 Digitální výstupy .....	62

6.4	CAN.....	63
6.5	Napájení ECU .....	64
6.6	Vstupy otáček kol .....	64
6.7	Paměť .....	65
6.8	Gyroskop.....	65
6.9	Analogové vstupy .....	66
6.10	USB .....	67
7	Softwarový interface ECU.....	68
7.1	Program Diagnostics .....	68
7.2	Program Parameters .....	69
7.3	Program Bootloader .....	70
7.4	Program Research .....	71
7.5	Program Test displeje .....	72
7.6	Program Test TSC1 .....	73
7.7	Program Test lock.....	73
7.8	Program Test CAN .....	74
8	Displej.....	75
8.1	Hardware.....	75
8.2	Software .....	75
9	Finální testování mechatronického systému.....	76
	Závěr.....	81
	Seznam použitých zkratk a symbolů .....	90
	Seznam příloh.....	94

## ÚVOD

Novodobé trendy směřují k zavádění autonomních systémů a umělé inteligence prakticky do všech zařízení okolo nás. Proto není divu, že nedílnou součástí každého vyrobeného automobilu je určitý typ autonomního systému. Pro urychlení vývoje takových systémů jsou stále navrhovány nové metody a různé sofistikované nástroje například ve formě hardwaru a softwaru. Cílem těchto metod a nástrojů je především snížení času a nákladů na vývoj, které stoupají k neúnosným částkám z důvodů zvyšujících se nároků na tvorbu stále složitějších mechatronických systémů. Řada vyvíjených a nabízených inteligentních mechatronických systémů souvisí s podvozky vozidel a s řízením jejich hnacího traktu. I tato práce se zabývá vývojem systému spojeného s hnacím traktem vozidel. Jedná se o vývoj autonomního mechatronického systému pro automatické uzavírání diferenciálu nazvaného Autonomous control of the powertrain – ACP. Dále je vyvíjena nastavba tohoto systému pro sledování okolního prostředí, kde se vozidlo nachází a kam směřuje s názvem Prediction of surroundings - PS.

Vyvíjený systém ACP-PS je sestaven z řídicí jednotky – ECU, senzorů a akčních členů. Základní funkcí systému je vyhodnocovat otáčky kol a z nich určovat, zda nedochází k prokluzu. Pokud prokluz nastane, řídicí systém vyhodnotí signály i z dalších senzorů, jako jsou pedály vozidla, snímač tlaku vzduchu v pneumatickém okruhu, signály z koncových spínačů v pneumatických válcích, kontrolních přepínačů a tlačítka, CAN zpráv z dalších ECU jednotek a dotykového displeje. Potom systém odešle signál akčnímu členu, který je sestaven z elektroventilu, pneumatického okruhu, pneumatického válce s koncovým spínačem a speciální zubové spojky. To znamená, že otevřením elektroventilu je puštěn tlakový vzduch do pneumatického válce, tím dojde k pohybu pístu s konzolou, která vyvodí sílu pro uzavření speciálních zubových spojek. Ty jsou umístěny v příslušných diferenciálech nebo použity pro spojení hnacích hřídelů předních náprav pro aktivaci pohonu všech kol. Řidič může systém ovládat přepínačem pro automatické a manuální řízení a dalšími přepínači a tlačítkem pro aktivaci pohonu všech kol, uzavření zadních osových diferenciálů a pro uzavření předních osových diferenciálů v manuálním režimu. Další možností ovládaní je nastavení jízdních módů řídicího systému pro silnici, polní cestu a terén/sníh na displeji, kde jsou zobrazovány i informace řidiči o stavu aktivace pohonu všech kol nebo uzavření příslušného diferenciálu.

Tato práce popisuje v první fázi stručný přehled používaných metod a nástrojů pro vývoj takovýchto mechatronických systémů včetně komerčně nabízených systémů pro řízení hnacího traktu vozidel. Dále je již popisován vlastní vývoj řídicího algoritmu, který byl i aplikován a odladěn pro vozy Tatra v rámci projektu TE 01020020 s názvem Centrum kompetence automobilového průmyslu Josefa Božka ve spolupráci se společností TATRA TRUCKS a.s., která ho implementovala do svých řídicích jednotek. V rámci vlastního vývoje a této práce bylo vytvořeno vlastní řešení s modernějšími prvky a nadstavbou pro sledování okolí vozidla. V práci je dále popisován vyvinutý nástroj pro tvorbu výpočtových modelů vozidel pro MIL testování, které bylo využito pro ověřování řídicího algoritmu. Dále je popsáno HIL testování na prototypu vozidel, návrh prototypu řídicí jednotky, programování grafického rozhraní pro komunikaci se systémem, programování dotykového displeje a závěrečné testování kompletního mechatronického systému, určeného již pro výrobu nulté série vozidel.

Tento systém je vyvíjen z důvodu lepšího využití hnacího momentu vozidla, šetření paliva a opotřebení pneumatik. Vozidlo vybavené tímto systémem je šetrnější k životnímu prostředí a k ochraně hnacího traktu před nevhodným ovládaním uzavírání diferenciálů nezkušenými řidiči. Proto je prioritně systém řízen automaticky. V závěru práce je zhodnocení vyvinutého



autonomního mechatronického systému včetně plánovaného dalšího vývoje z hlediska zmenšování prototypu ECU, rozšiřování nastavby systému pro predikci okolí vozidla až po implementaci neuronových sítí pro automatické učení vozidla během provozu.

## CÍLE PRÁCE

Cíle habilitační práce jsou vypsány v následujících bodech. Vývoj autonomního mechatronického systému pro uzavírání diferenciálu byl částečně navázán i na projekt TE 01020020 s názvem Centrum kompetence automobilového průmyslu Josefa Božka s konkrétním označením pracovního balíčku WP17. V rámci tohoto projektu a ve spolupráci se společností TATRA TRUCKS a.s. byl vyvíjený samostatný řídicí algoritmus odladěn pro jejich konkrétní typy nákladních vozidel a pro jejich vlastní elektronickou řídicí jednotku (ECU) vyvinutou dle požadavků algoritmu. Tento systém je veden ve společnosti pod názvem Tatra Drive Managment - TDM. V rámci této práce a dřívějšího vývoje byl vyvinut kompletní mechatronický systém nazvaný Autonomous control of the powertrain – ACP s dále vyvíjenou nástavbou tohoto systému pro sledování okolního prostředí. Nástavba systému sleduje, kde se vozidlo nachází a kam směřuje s názvem Prediction of surroundings - PS. Systém ACP-PS je tedy koncipován modulárně pro velkou škálu typů vozidel. Hlavní cíle habilitační práce byly tedy:

- Vyvinout modulární řídicí algoritmus pro uzavírání diferenciálů různých typů vozidel
- Sestavit výpočtový model vozidla pro Model in the Loop - MIL testování
- Provést MIL testování vyvíjeného řídicího algoritmu
- Provést Hardware in the Loop – HIL testování vyvíjeného systému na prototypu vozidla
- Vyvinout řídicí jednotku pro tento systém
- Vyvinout softwarové rozhraní pro diagnostiku, nastavování parametrů, zavádění nového programu do procesoru ECU, testování a sběr dat z ECU
- Naprogramovat dotykový displej pro komunikaci uživatele s ECU.
- Provést testování kompletního vyvinutého autonomního systému.

## 1 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

Pro vývoj mechatronických systémů mají firmy možnost využít řadu metod a nástrojů. To je od virtuálních simulací a testování až po užití softwaru a hardwaru pro HIL – Hardware in the Loop testování.

Z hlediska vývoje řídicího algoritmu jsou používány základní jazyky jako je C/C++ [1], [2] a také dost často software Simulink [3]. Ten je užíván v různých nadnárodních firmách, jako je např. firma Bosch. Řídicí algoritmus vytvořený v tomto softwaru se také následně kompiluje do jazyka C/C++. Potom jsou využita různá vývojová prostředí, například Keil [4], IAR [5], TrueStudio [31] atd., kde je prováděno vlastní programování algoritmu nebo jen jeho kompilace pro procesor ze souborů generovaných jinými softwary. Nyní se začínají využívat i inteligentnější jazyky jako je ADA [6], které byly především určené pro vojenský a letecký průmysl, ale probíhá i implementace v automobilovém průmyslu, kde má budoucnost před sebou. V některých případech jsou vývojová prostředí závislá na konkrétním procesoru a vývojovém softwaru daného výrobce. S novodobým trendem autonomních systémů je také značně využíváno neuronových sítí pro řídicí algoritmy, které potřebují velice výkonný hardware např. pro zpracování obrazu, polohy atd.

Testování vyvíjených mechatronických systémů probíhá několika formami. První formou je Model in the Loop – MIL testování, kdy je spojen vyvíjený řídicí algoritmus s fyzikálním výpočtovým modelem systému. Tam je odladován základní vyvíjený algoritmus a obecně princip fungování mechatronického systému. Následuje fáze Software in the Loop – SIL testování, kdy je již řídicí algoritmus ve formě vygenerovaného kódu, ale testuje se v prostředí počítače. Dalším krokem je Processor in the Loop – PIL testování, kdy oproti SIL testování je již řídicí algoritmus implementován na hardware. Tam je testováno, zda je hardware dostatečně koncipován pro řídicí algoritmus a nedochází např. ke zpoždování výpočtových smyček. Předposlední fází je Hardware in the Loop - HIL testování, kdy hardware slouží jako náhrada ECU a může být vyvíjený mechatronický systém testován na vozidle, aniž by byl k dispozici prototyp ECU. Poslední fází je závěrečné testování firmou, které vyvíjejí daný mechatronický systém, je testování kompletního mechatronického systému. To již může odpovídat nějaké verzi nulté výrobní série. Pro jednotlivé fáze jsou používány především softwary od MathWorks – Simulink, National Instruments – NI LabVIEW, NI VeriStand a hardware od National Instruments [20], dSpace [16], NovaCarts [17], Vector [18] a Speedgoat [19]. [3], [15], [67], [68]

Vývoj prototypů ECU - Electronic control unit může být realizován od návrhu schématu zapojení až po návrh tištěného spoje. Je vhodné vycházet z hardwaru použitého pro HIL testování, aby z hlediska elektroniky fungovala ECU obdobně. Pro návrhy jsou používány různé softwary Allegro [7], EasyEDA [8], KiCad EDA [73], Eagle PCB [9], OrCAD [10], NI Multisim [11] atd. Je jich opravdu velká řada a především záleží na náročnosti projektu vytvářeného prototypu ECU. Dokonce některé softwary jsou volně přístupné a zároveň mají vysokou úroveň funkcí. Pokud firma nechce věnovat investice do vývoje vlastního prototypu ECU, jsou k dispozici na trhu různé otevřené platformy ECU určených pro různé aplikace jako řízení motoru, to je např. FlexECU [12] od firmy Bosch, která je určena pro benzínové nebo naftové motory. Uživatel se nemusí starat o vývoj hardwaru, ale jenom připraví řídicí algoritmus, který je do ní implementován. Kód je pro ni tvořen v Simulinku, C/C++, nebo pomocí dalších vývojových nástrojů. Mezi další ECU, které je možné programovat, jsou produkty Pi Innovo tzv. Open ECU [13], ECU od společnosti Bosch pro motorsport [14] atd.

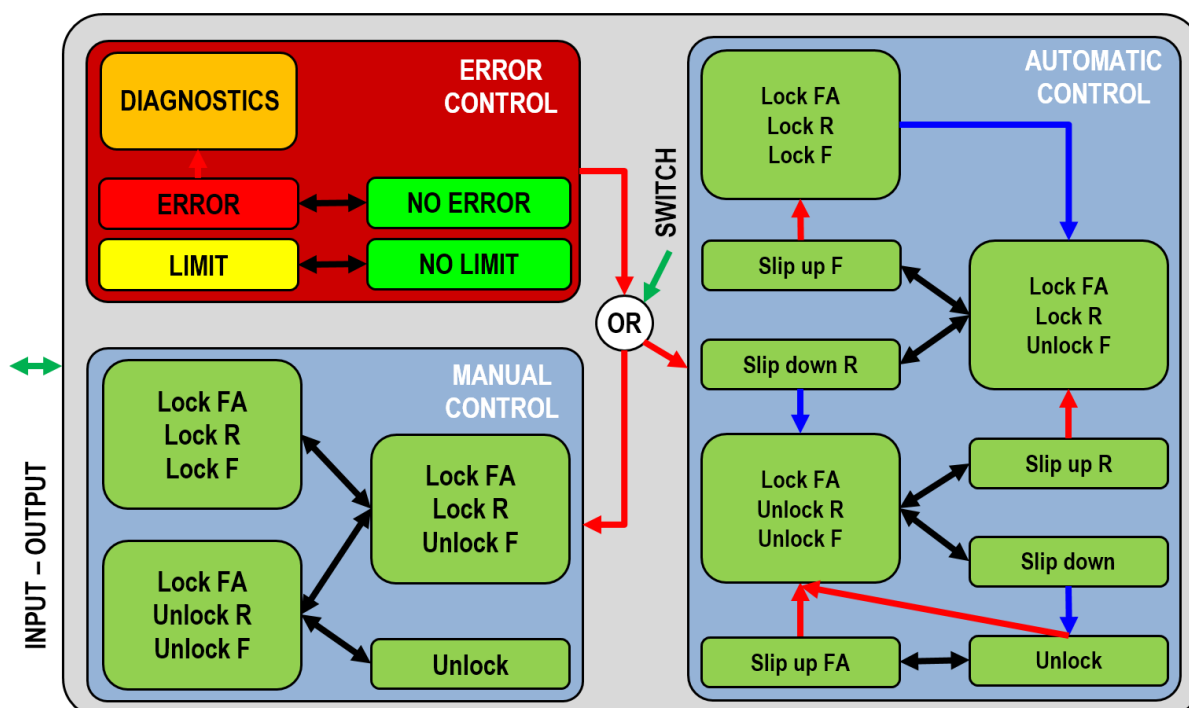
Displeje pro vyvíjené mechatronické systémy ve většině případů jsou dodávány pro automobilky různými firmami. Automobilka si již doprogramuje nebo nechá naprogramovat potřebou vizualizaci. Příkladem nabízených dotykových displejů může být firma US Micro Products [21], Topcon Electronic [22] nebo Continental s 3D dotykovým displejem [23]. Samozřejmě je řada dalších dodavatelů displejů pro automobilový průmysl, které lze pomocí daného rozhraní programovat, nebo lze využít pro jednodušší aplikace např. mikrokontroléry s displejem od společnosti STMicroelectronics.

V posledním období je snahou sjednotit software nebo obecně protokol pro nastavování a sledování parametrů ECU při testování a vývoji. Proto určitě stojí za zmínku software ETAS – INCA [24], který slouží i pro různé kalibrace a především je důležité, že pracuje podle norem ASAM [25]. Ty jsou právě určeny pro různé ladění ECU, kalibraci atd.

V oblasti řízení hnacího traktu, uzavírání diferenciálů, aktivace pohonu všech kol a toku momentu, lze v oblasti nákladních vozidel zmínit systémem pro uzavírání diferenciálů ZF ADM popsany v [26] a [27], nebo systém od společnosti Meritor Driver-Controlled Differential Lock (DCDL) [28]. Podle toho, že tyto dvě firmy udělaly společný podnik ZF Meritor, lze předpokládat, že se jedná o totožný systém ZF ADM jen na jiných nápravách. Jedná se tedy o systém, který vyhodnocuje prokluz kol a dle řídicího algoritmu je vyhodnocen stav, zda má být příslušný diferenciál uzavřen. Potom v diferenciálu je zubová spojka, která realizuje mechanické spojení, aby byla poháněná např. obě kola jedné nápravy. Samozřejmě na trhu je řada systémů, které řídí uzavírání diferenciálů u osobních vozidel, kde nejenom dochází k uzavírání diferenciálu, ale je řízeno rozložení hnacího momentu např. Torque Vectoring. Tyto systémy však v momentální chvíli není možné využívat u nákladních nebo speciálních vozidel z hlediska vysokých přenášených hnacích momentů. Proto je nutné se vydávat směrem k uzavírání diferenciálů pomocí zubových spojek a k nim navrhnout řízení bez vzniku momentových rázů. Takovýmto systémem se zabývá i tato práce. Obecně vývoj nejenom takovýchto systémů jde směrem neuronových sítí a systémů, které se samy učí z aktuálních jízdních situací. Zjednodušeně řečeno vývoj a aplikace mechatronických systémů jde směrem k umělé inteligenci.

## 2 VÝVOJ ŘÍDICÍHO ALGORITMU

V rámci této kapitoly bude popsán vyvíjený řídicí algoritmus pro automatické uzavírání diferenciálů u osobních, nákladních a speciálních vozidel a traktorů. Cílem tohoto vývoje bylo vyvinout autonomní systém, který by uživateli ulehčil jízdu terénem a pomohl mu dopravit vozidlo do krajně nepřístupných míst, aniž by tam uvízlo. Aby mohl být vytvořen kompletní autonomní systém, bylo nutné naprogramovat řídicí algoritmus, kde jedno z možných schémat fungování je zobrazeno na následujícím obr. 1. V prvotních návrzích bylo využito částečně závislé a nezávislé řízení uzavírání diferenciálu. A právě jednotlivé principy a metody fungování budou popsány v následujících podkapitolách v návaznosti na využití procesoru STM32F4xx.



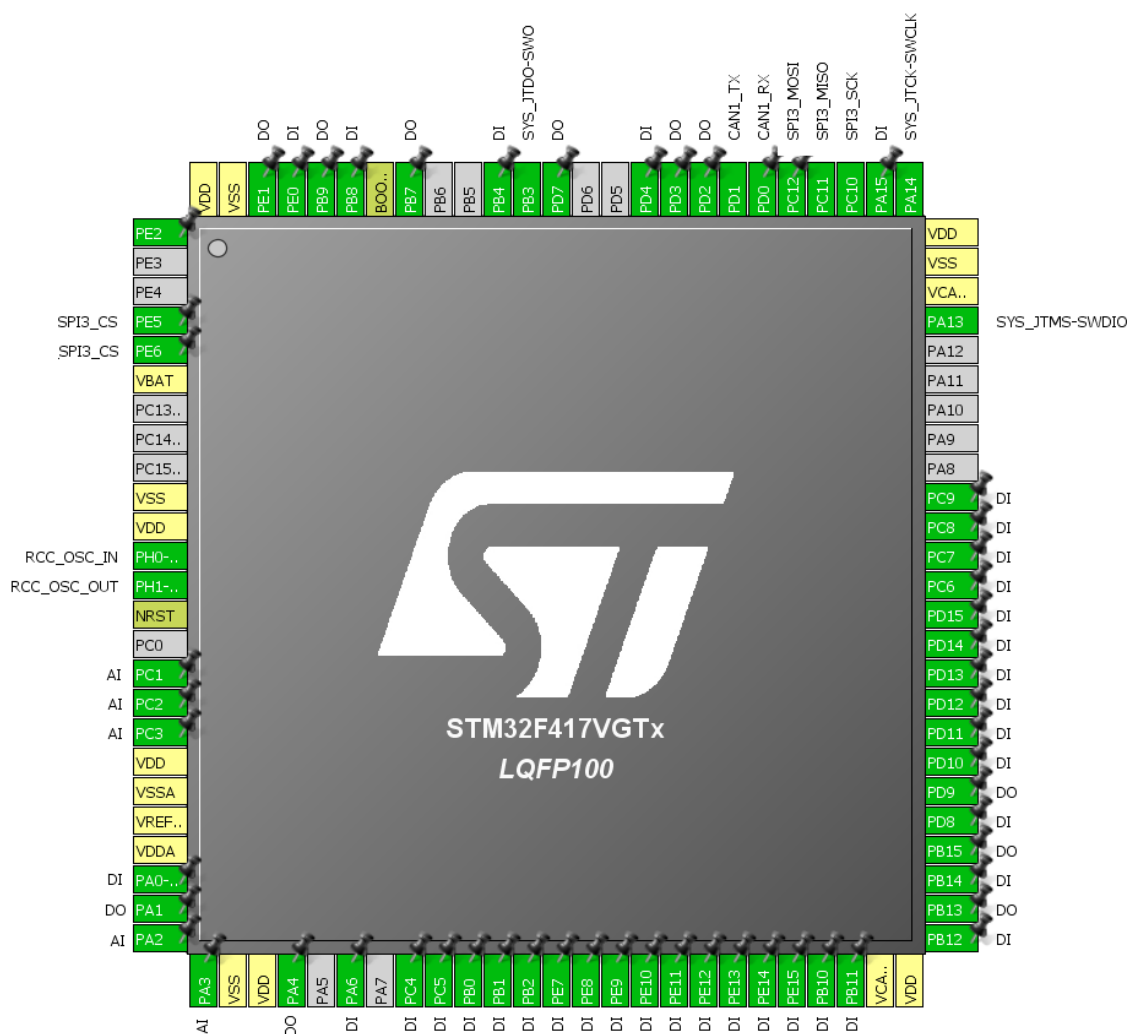
Obr. 1 Schéma řídicího algoritmu

### 2.1 ZÁKLADNÍ FUNKCE

Základní funkcí vyvíjeného autonomního mechatronického systému je vyhodnotit prokluz mezi jednotlivými koly a částmi hnacího traktu určitého vozidla. Tím řídicí algoritmus určí podle nadefinovaného postupu, zda má příslušný diferenciál uzavřít nebo např. aktivovat pohon všech kol. Během vyhodnocování bere v úvahu další vstupy a faktory, které mohou zásadně ovlivnit rozhodování tohoto autonomního systému. Pokud systém rozhodne, že je potřeba uzavřít příslušný diferenciál, pošle signál akčnímu členu a ten příslušný diferenciál uzavře. Pokud je mechanická část diferenciálu skutečně uzavřena, systém je informován a může reagovat na případné další faktory určující, že vozidlo se nachází v nepříznivých adhezních podmínkách a je zapotřebí vhodně na to reagovat. Nebo naopak rozpozná stav, kdy je vhodné již příslušné diferenciály nebo pohon všech kol deaktivovat. Během těchto procesů zároveň probíhá diagnostika a systém určuje případnou závadu, aby se předešlo nebezpečným situacím. V případě potřeby je možné systém přepnout do tzv. manuálního režimu, kdy uživatel může diferenciály uzavírat jako u běžných sériových vozidel bez tohoto systému.

## 2.2 PŘÍPRAVA PROJEKTU

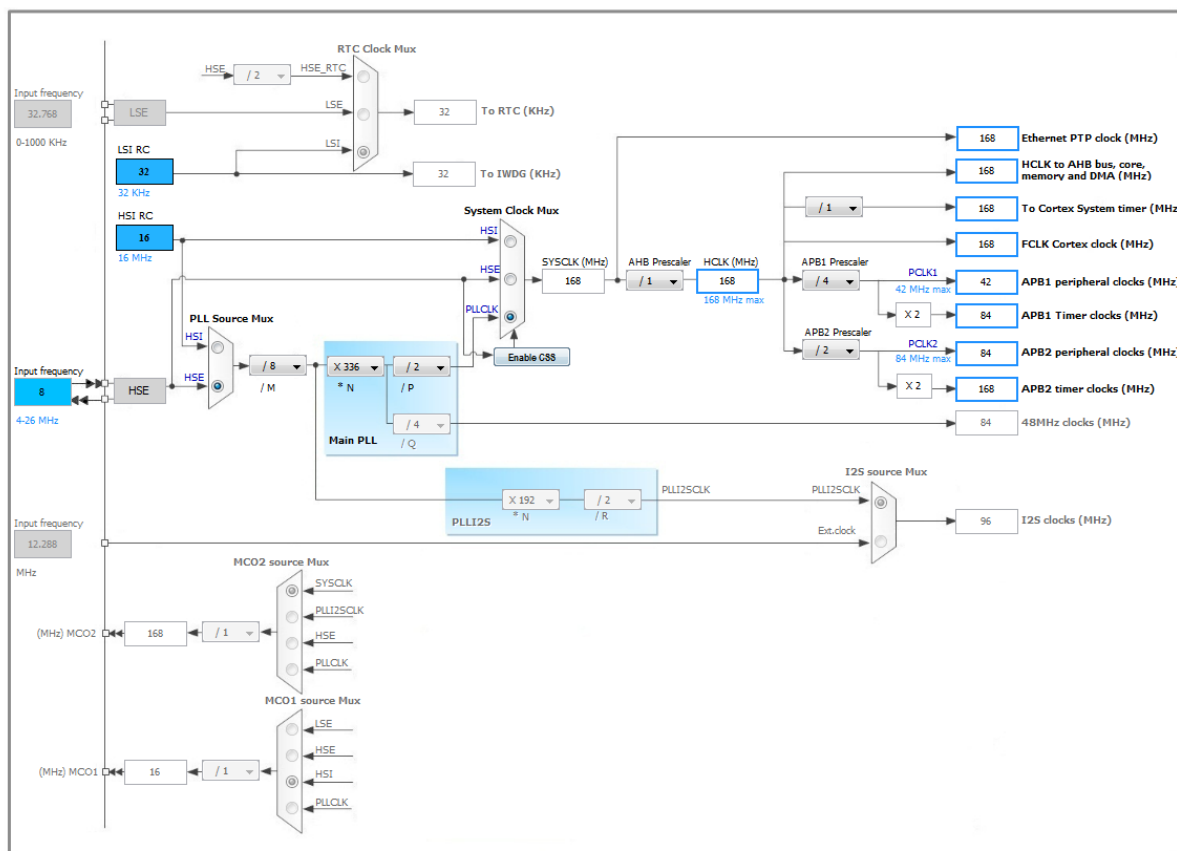
Aby bylo možné začít tvořit řídicí algoritmus, bylo zapotřebí vytvořit základní projekt obsahující periferie komunikující s jednotlivými piny procesoru. Na následujícím obrázku je zobrazeno základní využití pinů procesoru STM32F417VGTx [29], který byl využit pro sestavení prototypu ECU. Toto schéma je připraveno v prostředí STM32 Cube MX softwaru [30], který slouží pro vygenerování základního projektu pro daný procesor v jazyce C a pro konkrétní vývojová prostředí, v tomto případě pro Atollic TrueSTUDIO software [31]. Pro komunikaci s periferiemi jsou použity HAL funkce popsané v UM1725 [32].



Obr. 2 3D Schéma zapojených pinů procesoru pro vygenerování projektu v STM32 CUBE MX softwaru

Jelikož se jedná o autonomní řízení uzavírání diferenciálu, je jedním z hlavních vstupů hodnota otáček kol, poloha pedálů, Controller Area Network - CAN zprávy z jiných řídicích jednotek – ECU, hodnota tlaku pneumatického okruhu, hodnota natočení volantu, poloha kontrolních prvků a informace o fyzickém uzavření konkrétního diferenciálu. Naopak jsou řízeny akční

členy nebo signalizační kontrolky. Proto jejich periferie byly nastaveny a vygenerovány pro daný procesor právě využitím softwaru STM32 Cube MX.



Obr. 3 Nastavení hodin procesoru pro jednotlivé periferie

Předchozí obr. 3 zobrazuje nastavení procesoru pro vyvíjený autonomní systém. V tomto případě je užito externího oscilátoru neboli krystalu s frekvencí 8 MHz. Následným nastavením je docíleno maximálního využití výkonu procesoru s frekvencí 168 MHz. Z této frekvence jsou dopočítány další frekvence pro různé periferie. V poslední fázi bylo zapotřebí nastavit konkrétní konfigurace pro CAN, analogové vstupy, SPI komunikaci, digitální vstupy a výstupy, přerušení, časovače a FreeRTOS.

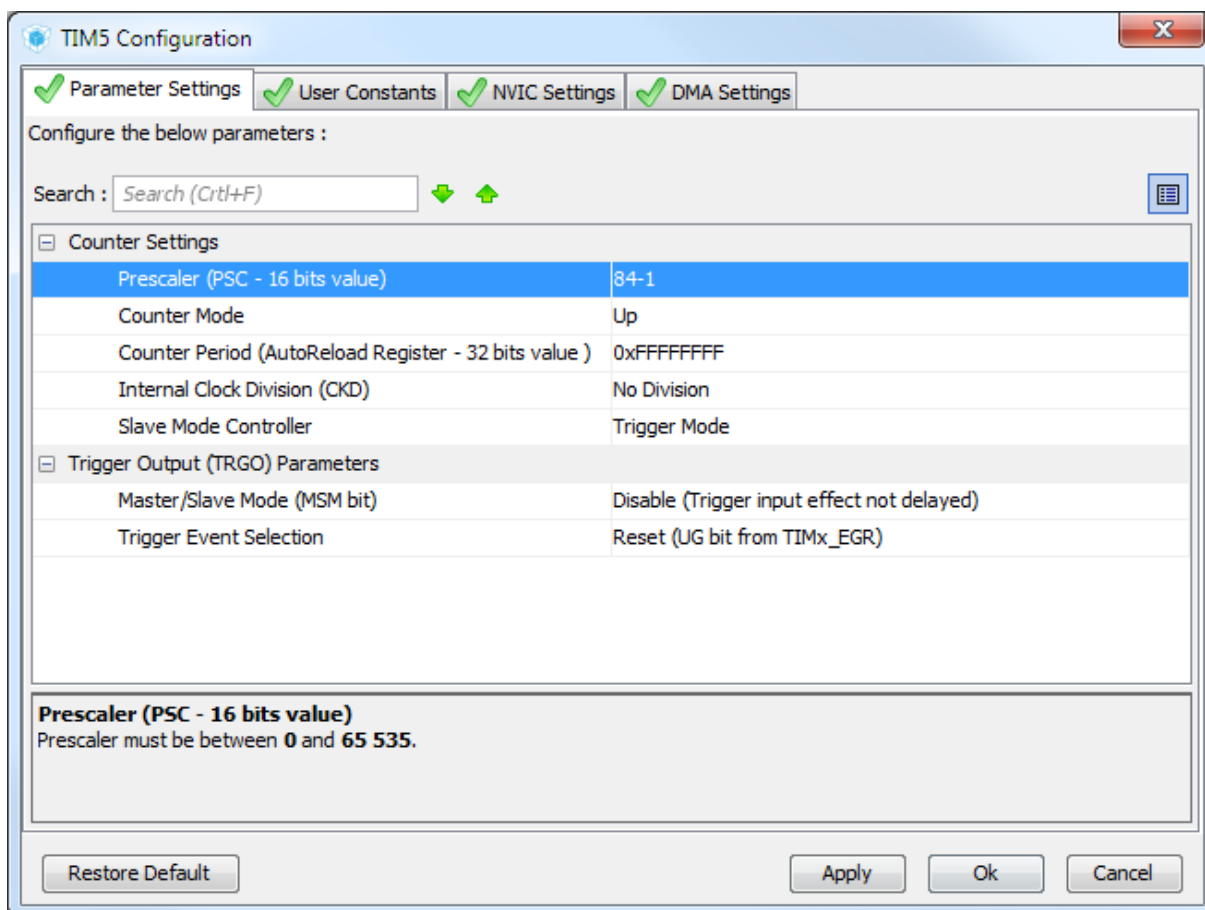
## 2.3 VSTUPY

Elektrické připojení a popis jednotlivých snímačů bude popsán v příslušné kapitole. Zde bude popsáno již zpracování konkrétních hodnot a nastavení pinů procesoru STM32F417, kde bude implementován řídicí algoritmus.

### 2.3.1 OTÁČKY KOL

Zpracování signálu nesoucí informace o otáčkách je předáváno ve formě digitálního signálu - DI. Kola vozidla bývají osazena indukčními nebo halovými snímači a tyto snímače generují specifický signál v závislosti na poloze vůči např. drážkám v náboji. Signál se mění vždy při přechodu přes danou drážku nebo zub a je např. u indukčního snímače ještě zpracováván hardwarově. Již tento signál ve formě digitálního signálu je zpracováván tak, že v procesoru je zastaven běžící program a provede se tzv. přerušení. V tomto případě je pin

nastaven, že k přerušení dojde, pokud je detekována náběžná hrana vstupního signálu. Tohoto přerušení bylo využito z toho důvodu, aby hlavní program nezpomaloval zpracování otáček kol a tím především přesnost jejich hodnoty. Obecně přerušení má sloužit pro malé kódy zpracovávající důležité rychlé děje, ale nesmí zbytečně zdržovat další procesy procesoru. Proto v této aplikaci je v každém přerušení jen zpracována časová perioda mezi jednotlivými zuby a následně zpracování již probíhá v hlavní smyčce programu, která běží mnohem pomaleji. Obecně pro získání periody je možné využít hardwarového nebo softwarového přerušení.



Obr. 4 Konfigurace časovače pro vyhodnocování otáček kol

V tomto případě bylo dostačující softwarové přerušení, jelikož maximální otáčky kol vozidla nejsou tak vysoké. V případě např. snímání vysokých otáček by hardwarové přerušení bylo nutností, aby přesnost otáček byla co největší. Proto procesor umožňuje využití přerušení několika kanálů konkrétního časovače a vše se řeší na hardwarové úrovni, tím je možné získat periodu velice přesně bez vlivu běhu ostatních částí kódu. V tomto případě bylo použito standartní přerušení, kde je zpracování periody softwarové. To znamená, že při přerušení se vyvolá příkaz pro přečtení hodnoty časovače a od ní se odečte předchozí hodnota, tím je získána perioda. Tímto způsobem je samozřejmě konkrétní hodnota získaného času ovlivněna rychlostí procesu a počtem příkazů pro její získání. Ověření proběhlo generátorem signálu ve formě sinusového průběhu obdobného k chování indukčního snímače. Z výsledků vyplynulo, že do otáček  $1000 \text{ min}^{-1}$  je odchylka mezi generovanými a zpracovanými otáčkami naprosto zanedbatelná. Pro vytvoření kódu pro zpracování periody bylo potřeba také nastavit časovač procesoru, kdy byl využit časovač procesoru TIM5. V jeho konfiguraci muselo být především nastaveno tzv. „Prescale“ na hodnotu 83 v závislosti na nastavení frekvencí procesoru. To



znamená, že výstupní hodnota z časovače je násobena 1  $\mu$ s. Dále byla nastavena hodnota tzv. „Counter Period“ a to na maximální hodnotu, jak je zobrazeno na předchozím obrázku. Jedná se o hodnotu, kdy je časovač znovu spuštěn od nulové hodnoty. Výsledná následující rovnice, která vypočítává otáčky kola je již umístěna v hlavní smyčce, která běží již pomaleji.

$$omg_i = 60 \frac{1000000}{Period_i z}, i = FLI, FRI, \dots RLI, RLII, \dots \quad (1)$$

kde  $omg$  jsou otáčky příslušného kola vozidla,  $Period$  - perioda natočení ozubeného kola pro indukční snímač o jeden zub,  $z$  – počet zubů na jednu otáčku kola vozidla a  $i$  – je index kola s označením kombinací písmen  $F$  – přední,  $R$  – zadní,  $L$  – levá,  $R$  – pravá a  $I, II, III$  – pořadí nápravy. Pro případ vozidel, která mají nějaký typ brzdného mechatronického systému např. Anti-lock Braking Systém – ABS nebo Electronic Braking Systém – EBS, bylo využito možnosti brát informace o otáčkách z CAN zprávy Electronic Brake Controller 2 - EBC2, její označení je dáno normou J1939. Pro tento případ byl vytvořen kód pro zpracování přijímaných CAN zpráv, které jsou ve formě rychlosti přední nápravy  $FroAxlSpd$  a referenčních rychlostí kol  $RelSpe_i$  vztažených právě k té rychlosti přední nápravy.

$$omg_i = 60 \frac{FroAxlSpd + RelSpe_i}{2 \pi r_d 3,6}, i = FLI, FRI, \dots RLI, RLII, \dots \quad (2)$$

kde  $rd$  je poloměr kola. Pokud vozidlo umožňuje využití této zprávy, je tím redukován počet užitých snímačů na vozidle. To souvisí s celkovou cenou systému i složitostí zástavby snímačů ve vozidle. Důležitou poznámkou je nutnost správného nastavení formátu výstupní zprávy s informacemi o rychlostech kol z příslušné ECU. Jelikož některé ECU mají možnosti například přizpůsobit výstupní hodnoty rychlosti kol k hodnotě rychlosti z tachografu, proto by mohl být rozdíl mezi otáčkami kol získaných přímo ze snímačů nebo z CAN zprávy EBC2 rozdílný. Potom by si algoritmus mohl myslet, že se jedná o prokluz, aniž by k nějakému docházelo. Také v obou ECU musí být nastavena stejná hodnota obvodu kola, aby otáčky mohly být správně přepočítávány.

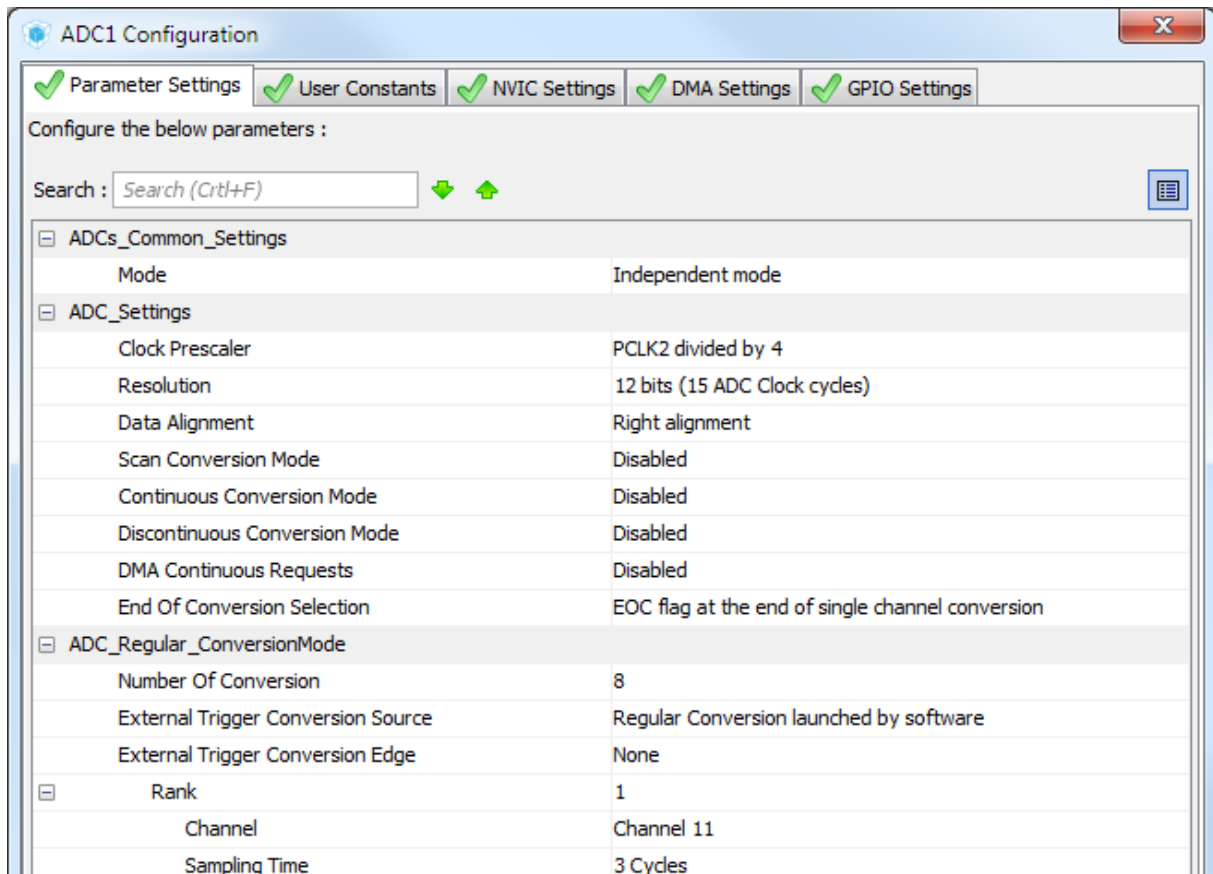
### 2.3.2 POLOHA PEDÁLŮ

Jedná se o hodnotu spojkového, brzdového a plynového pedálu, která je zpracována hardwarem ECU a procesoru předávána jako digitální signál. Tedy jednotlivé piny procesoru byly nastaveny na digitální vstupy a následně pomocí předpřipravené funkce `HAL_GPIO_ReadPin()` od společnosti STMicroelectronic [32] byla čtena hodnota daného pinu. Tato hodnota již vstupuje do hlavního řídicího algoritmu, kde je s ní dále pracováno. Tento případ získávání informací o daném pedálu je užit v případě vozidel se staršími motory. U nových vozidel je předpokládáno získávání těchto hodnot z CAN zprávy motoru Electronic Engine Controller 2 - EEC2 a ve zprávě Cruise Control/Vehicle Speed – CCVS. Tím je zase ušetřeno zdvojování snímačů pro pedály a zároveň cena vyvíjeného systému. Ještě je potřeba zmínit, že pokud nejsou informace získávány přes CAN, je hardwarově přes logiku OR k brzdovému pedálu připojen i signál parkovací brzdy. To znamená, pokud brzdový pedál nebo parkovací brzda jsou aktivovány, zpracovává procesor hodnotu 1 znamenající brzdění.

### 2.3.3 TLAK PNEUMATICKÉHO OKRUHU

U vyvíjeného autonomního mechatronického systému je jednou z možností využití pneumatického okruhu pro uzavírání zubových spojek v diferenciálu a tím uzavření daného diferenciálu. Proto bylo nutné připravit i zpracování signálu ze snímače tlaku. Jelikož je

předpokládáno, že takový snímač má analogový výstup, je nejdříve hardwarově upraven jeho rozsah a ochrana pro procesorový analogový vstup. Tomuto pinu musí být nastavena konfigurace v podobě analog-digitál – AD převodníku, v tomto případě na 12 bitů, tzv. „Clock Prescaler“ a velice důležitou hodnotou je „Sampling Time“ jednotlivých analogových vstupů. Kombinací těchto parametrů lze dopočítat min. čas zpracování vstupního signálu, dle informací v manuálu AN2834 na stránkách [29] a následující rovnice, který musí být větší než hodnota stanovená katalogovým listem procesoru STM32F417 na stránkách [32].



Obr. 5 Konfigurace analogových vstupů

V tomto případě to dle následující rovnice vychází na min. 3 cykly.

$$SamTimAI = \frac{(SamTim + 12) CloPre}{PCLK2}, \quad (3)$$

Kde  $SamTim$  je Sampling Time,  $CloPre$  – Clock Prescale a  $PCLK2$  – je frekvence periferních hodin procesoru. Hodnota AD převodníku 12 v rovnici je hodnota n-bitů pro přesnější vyjádření hodnoty analogového vstupu a je volitelná uživatelem v určitém rozsahu. Nastavení analogových vstupů ještě umožňuje získat tzv. referenční napětí  $Vrefint$ , které následně slouží pro kalibraci finální hodnoty analogového vstupu, která je již zpracovávána v hlavním algoritmu. Tato hodnota je čtena pomocí základní funkce `HAL_ADC_GetValue()` a následně dále zpracovávána pomocí naprogramované funkce pro interpolaci a přepočítávána v algoritmu na konkrétní hodnotu tlaku.

### 2.3.4 NATOČENÍ VOLANTU

Vyvíjený autonomní systém zpracovává především otáčky kol vozidla, aby vyhodnotil skluz. Pokud však vozidlo jede zatáčkou, algoritmus by vyhodnotil, že tam také vzniká skluz a uzavíral by diferenciály, aniž by to bylo nutné. V některých případech by to mohlo být až nebezpečné nebo destruktivní pro hnací trakt. Z toho důvodu je zapotřebí zpracovávat signál reprezentující natočení volantu vozidla. Proto byla vytvořena funkce pro jeho zpracování založena na teorii Ackermana, kdy lze přepočítat zatočení pravého nebo levého předního kola na teoretický skluz mezi jednotlivými koly nebo hřídelemi. Je nutné však podotknout, že tento přístup bude nejlépe fungovat při nízkých rychlostech a při vyšších by se mělo již počítat i s vlivem úhlu směrové úchytky pneumatiky. Nebo je možno využít druhý přístup, kdy by se vozidlo pohybovalo po kruhu, a pro různé zatočení kol a rychlost vozidla by se snímaly otáčky kol. Z těchto dat by byla vytvořena tabulka korekčního skluzu pro různé signály ze snímače natočení volantu. Jelikož algoritmus počítá s rezervou limitu pro překročení skluzu je první verze pro zpracování korekčního skluzu dostačující a jednodušší, proto byl volen tento přístup. Ale pro budoucí rozšiřování je předpokládáno zavedení i druhé možnosti. Z hlediska signálů ze snímače natočení volantu se jedná o analogový signál a je zpracováván jako v přechozí kapitole, jen s tou výjimkou, že napětí se interpolací přepočítává na konkrétní zatočení předních kol. Také zde jsou používány oba dva signály užitého snímače, kdy jeden signál narůstá a jeden klesá a jejich součet odpovídá napájecímu napětí. To je následně využito pro analýzu chyby snímače.

### 2.3.5 KONTROLNÍ PŘEPÍNAČE, SPÍNAČ A DISPLEJ

Mechatronický systém je uživatelem ovládán pomocí tři přepínačů, jednoho spínače a displeje. První z přepínačů slouží pro nastavení automatického nebo manuálního režimu řízení uzavírání diferenciálů. Další dva přepínače jsou již pro manuální režim a to pro připojení pohonu všech kol a případných mezinápravových diferenciálů. Další je pro uzavření zadních osových diferenciálů. Tlačítko/spínač slouží pro uzavření všech předních osových diferenciálů vozidla. Jelikož se jedná o tlačítko, signál pro uzavření diferenciálů je realizován jen po dobu stisku z hlediska bezpečnosti. V tomto případě je signál pro procesor ve formě digitálního signálu a pomocí stejné funkce jako u pedálu je tato hodnota čtena z příslušného pinu. Displej je především určen pro informování uživatele, zda je např. aktivován pohon všech kol, nebo jaký diferenciál je právě uzavřen. Také slouží pro nastavování řídicích módů ECU. To znamená, že pokud je řídicí systém nastaven na automatické řízení, může uživatel volit mezi třemi módy, které jsou silnice, polní cesta a terén/sníh. Komunikace probíhá pomocí CAN zpráv. Pro další aplikace nebo rozšiřování systému je předpokládáno zpracování všech ovládacích prvků z displeje. V tomto případě je rozdělení ovládání na fyzické a virtuální prvky dle požadavků zákazníka.

### 2.3.6 SPÍNAČE UZAVŘENÍ DIFERENCIÁLU

Poslední ze zpracovávaných vstupů jsou spínače zpětných vazeb v pneumatickém válci, které informují řídicí algoritmus a uživatele o reálné aktivaci pohonu všech kol nebo uzavření příslušného diferenciálu. Opět se zde jedná o digitální signál, který je zpracován procesorem.

## 2.4 CAN KOMUNIKACE

Jak již bylo zmíněno, některé vstupní signály pro řídicí algoritmus, jsou přenášeny pomocí CAN komunikace. Jelikož se jedná o aplikaci tohoto systému především pro nákladní vozidla a traktory, tak bylo vše programováno dle norem J1939 [33], [34], [36], [37] a [38]. Tyto normy jsou užívány právě pro nákladní vozidla, autobusy atd. Jedná se o komplexní normu, která se

člení na další podsekcce, kde by bylo vhodné např. zmínit normu J1939-71 – obsahující aplikační vrstvy, to znamená, že popisuje přenášená data zprávy, a jejich vnitřní uspořádání pro různé signály z motoru, brzdných systémů vozidel atd. Dále norma J1939-73 popisuje diagnostické zprávy a jejich odesílání řídicí jednotkou. J1939-21 popisuje vlastní tvar zprávy, tedy konkrétně její formát. V tomto případě užívaný rozšířený formát zprávy „CAN Extended Frame Format“. Při programování kódů pro zpracování přijímání a odesílání CAN zpráv bylo především vycházeno právě z normy J1939 [33] a dalších odborných knih [39], [40]. Podrobnější popis fungování CAN komunikace je popsán právě ve zmíněných normách a nebude se podrobněji rozebírat. Zde bude popis zaměřen na CAN zprávy z různých ECU a na zprávy, které odesílá vyvíjená ECU.

Jedná se tedy o hodnoty otáček ve zprávě EBC2, která má ID 0x18FEBF0B. Hodnota 18 v identifikátoru zprávy znamená její prioritu, hodnota FEBF je tzv. „Parameter Group Number“. V podstatě se jedná o konkrétní označení zprávy, která vždy obsahuje příslušná data. A hodnota 0B určuje zdrojovou adresu ECU, ze které zpráva přišla. Tato zpráva obsahuje informace rychlosti přední nápravy v prvních dvou bytech a vždy v následujícím bytu je předávána informace o relativní rychlosti jednotlivých kol. Jelikož zprávy používají 8 bytů pro přenos dat, je možné zasílat informace až o šesti kolech. Především záleží na konkrétní ECU, která tuto zprávu odesílá, ale ve většině případů ABS jednotky odesílají informace jen ze 4 kol.

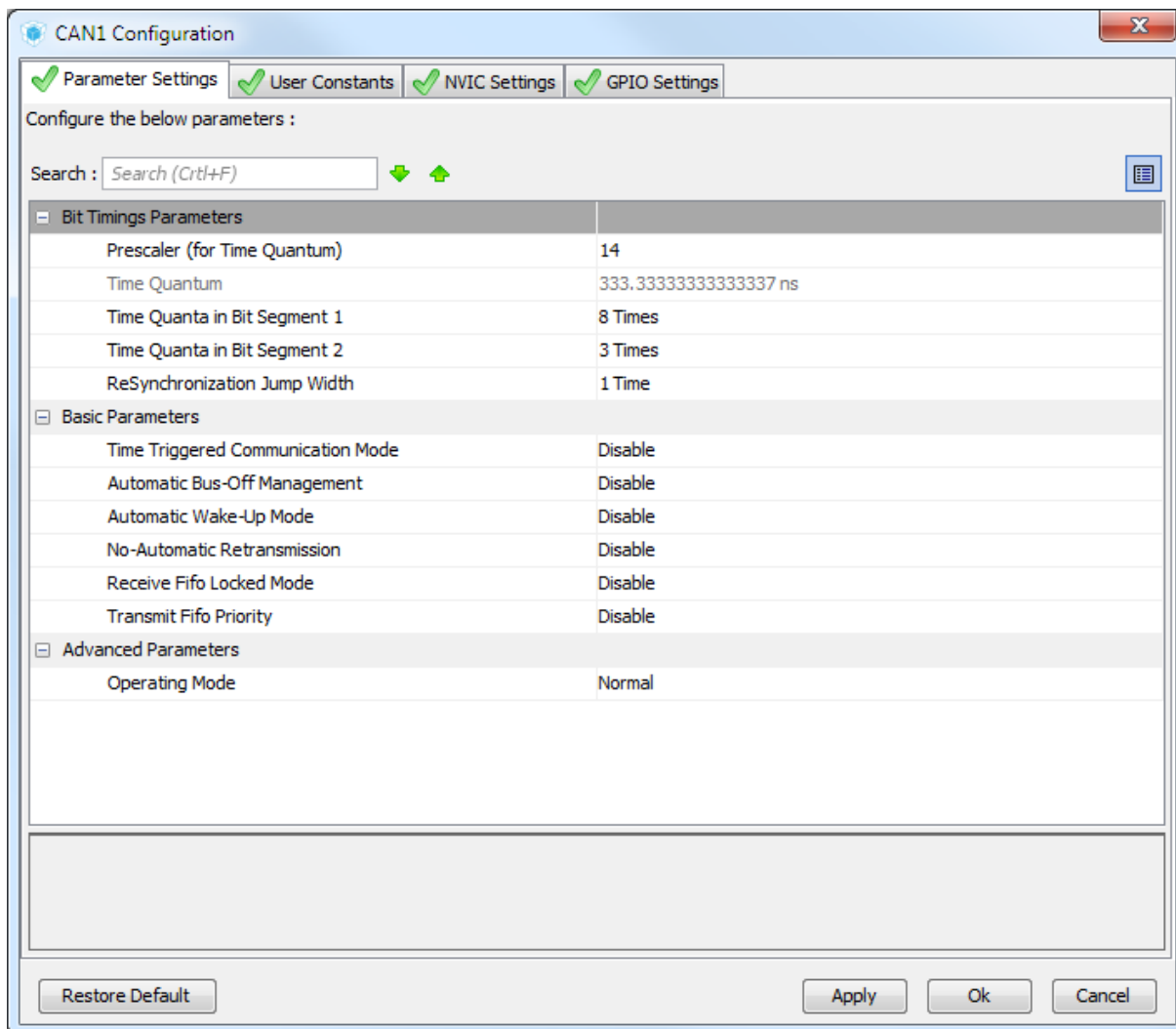
Informace o stavu pedálů je ve více zprávách a vyvíjený autonomní systém používá zprávu EEC2 s ID 0x0CF00300 a CCVS s ID 0x18FEF100. Z první zprávy je zpracovávána hodnota Accelerator Pedal Position 1, která je umístěna na druhém bytu dat. Tato hodnota určuje polohu plynového pedálu. Z druhé zprávy jsou zpracovávány data Brake Switch pozice 4.5 bytu (začátek na 4 bytu a 5 bitu) a délka 2 bity, Clutch Switch pozice 4.7 bytu a délka 2 bity. Tyto hodnoty mohou nabývat 4 hodnot, zda je pedál stlačen nebo uvolněn a zda je hodnota k dispozici nebo je nějaká chyba. Z této zprávy je zpracovávána ještě hodnota Parking Brake Switch.

Aby nedocházelo při brzdění a aktivaci ABS k matení vyvíjeného systému, tak je čtena zpráva Electronic Brake Controller 1 - EBC1 s ID 0x18F0010B. Z jejích dat je zpracovávána hodnota Anti-Lock Braking (ABS) Active umístěná na pozici 1.5 bytu a délce 2 bity. Hodnota nabývá 4 hodnot tj. pasivní, aktivní, rezervováno a není k dispozici.

K ochraně hnacího traktu souvisejících mechanických částí vyvíjeného systému je čtena zpráva s aktuální hodnotou hnacího momentu. To je ve zprávě Electronic Engine Controller 1 - EEC 1 s ID 0x0CF00400. Z dat zprávy je zpracovávána hodnota Actual Engine - Percent Torque na pozici třetího bytu o velikosti 1 byte. Pokud je tato hodnota k dispozici, algoritmus při překročení nastaveného limitu již neuzavírá diferenciály, aby nemohlo dojít k nadměrnému rázu v hnacím traktu a případnému poškození.

Z hlediska stability vozidla při brzdění, aby nedocházelo k ovlivňování brzdových asistenčních systémů vyvíjeným systémem, jsou užívané signály brzdového pedálu, aktivace ABS a parkovací brzdy. U nákladních vozidel jsou další způsoby brzdění pomocí retardéru nebo motorové brzdy. Z toho důvodu jsou přijímány CAN zprávy Electronic Retarder Controller 1 - ERC1 s ID 0x18F00010 a 0x18F00029. Z dat zprávy je zpracovávána hodnota Retarder Torque Mode na pozici 1.1 bytu o délce 4 bity. Tato hodnota nabývá řady hodnot, proto zde nebudou vypsány, ale v podstatě jsou kontrolovány dva stavy odpovídající hodnotě 0b0001 a 0b1000. Při nich je předpokládáno, že je určitým způsobem brzděno. To vyplývalo i z testování na vozidle. V případě, že vozidlo tyto zprávy neposkytuje, je tento stav ošetřen signálem z

plynového pedálu. Pro vysvětlení, pokud je signál z plynového pedálu 0, je předpokládáno, že může být brzděno motorem. Potom řídicí algoritmus neaktivuje pohon všech kol nebo neuzavírá diferenciály a naopak pokud jsou některé diferenciály uzavřeny, tak se zašle signál pro jejich otevření.



Obr. 6 Konfigurace CAN komunikace

Komunikace s displejem je přes CAN zprávy s ID 0x18EF09A7, která vysílá požadavek na změnu módu a po potvrzení zprávou ID 0x18EFA709, je zaslán konkrétní mód, ve kterém má fungovat autonomní řízení, tedy silnice, polní cesta nebo terén. Data pro informování řidičem na displeji jsou odesílána ve zprávě s ID 0x18FFCC09 s periodou 100 ms, kde jsou informace o aktivaci pohonu všech kol, uzavření jednotlivých diferenciálů, nastavený mód ECU, nastavení manuálního nebo automatického módu, chyb a limitů.

Odesílané CAN zprávy pro testování a vývoj mechatronického systému jsou s ID 0x18FF1109, 0x18FF2109, 0x18FF3109, 0x18FF3209, 0x18FF4109, 0x18FF4209, 0x18FF4309, 0x18FF5109, 0x18FF45209, 0x18FF6109, 0x18FF6209, 0x18FF6309, 0x18FF7109 a jsou odesílány jen v případě přijímané zprávy ECU s ID 0x18FF81FA a příslušným kódem v jejich datech. A jak již bylo zmíněno, jsou určeny pro vývoj a testování kompletního systému, kde v

datech CAN zpráv jsou implementovány vnitřní proměnné řídicího algoritmu, aby ho bylo možné kontrolovat i nadále při dalším rozšiřování a při testování s prototypem ECU.

Vyvíjený autonomní systém umožňuje aktivovat i naprogramovanou funkci pro omezování hnacího momentu nebo otáček motoru. Je to opět pro ochranu hnacího traktu jako v případě čtení aktuální hodnoty hnacího momentu motoru. Je zde využití CAN zprávy Torque/Speed Control 1 - TSC1, která umožňuje v určitých případech omezovat motor z hlediska otáček a hnacího momentu. V řídicím algoritmu je naprogramován mechanismus, že pokud algoritmus vyhodnotí, že má být konkrétní diferenciál uzavřen, tak nejdříve odešle požadavek na snížení otáček motoru, aby se částečně snížil rozdíl otáček uzavíraného diferenciálu a tím bylo zamezeno vzniku momentových rázů v hnacím traktu. Tato funkce je volitelná uživatelem, jelikož starší typy motorů nemusí zprávu TSC 1 podporovat.

Základní nastavení pinů procesoru pro CAN komunikaci je zobrazeno na obr. 6. Dle manuálu RM0090 na stránkách [29] je potřeba nastavit tzv. „Sample point“ je to poloha čtení přenášeného bitu CAN sběrnici, aby bylo možné synchronizovat jednotky ECU pracující na jedné sběrnici. Tato poloha se určuje pomocí tzv. kvant. U nákladního automobilu je přenosová rychlost CAN sběrnice  $250 \text{ kBit s}^{-1}$ . Podle následující rovnice lze určit čas trvání jednoho bitu na CAN sběrnici

$$TimBit = \frac{1}{1000 BitRat}, \quad (4)$$

kde *TimBit* je čas trvání jednoho bitu na CAN sběrnici a *BitRat* je přenosová rychlost CAN sběrnice. Pro danou přenosovou rychlost odpovídá 4000 ns. Této hodnotě by se měl rovnat součet času trvání tzv. kvant. Dle následující rovnice lze dopočítat čas jednoho kvanta.

$$TimQua = \frac{PreQua}{1000000 PCLK1} \quad (5)$$

kde *TimQua* je čas trvání jednoho kvanta a *PCLK1* je frekvence periferních hodin procesoru pro CAN 1 pro jiné CAN mohou být užity jiné hodiny a *PreQua* – Prescaler pro čas kvanta. Pro nastavenou hodnotu 14 je čas kvanta 333,33 ns. Dále je potřeba dopočítat potřebný počet kvant před bodem čtení hodnoty bitu tedy Time Quanta in Bit Segment 1 – BS1 a počet kvant za Time Quanta in Bit Segment 2 – BS2. Jak manuál RM0090 pro STM procesor popisuje, že před prvním segmentem se ještě nachází Synchronization segment, který má fixní hodnotu 1. Dle následující rovnice po sečtení všech kvant a vynásobením časem jednoho kvanta musíme dojít, ke stejnému času jako vychází z rovnice pro *TimBit*.

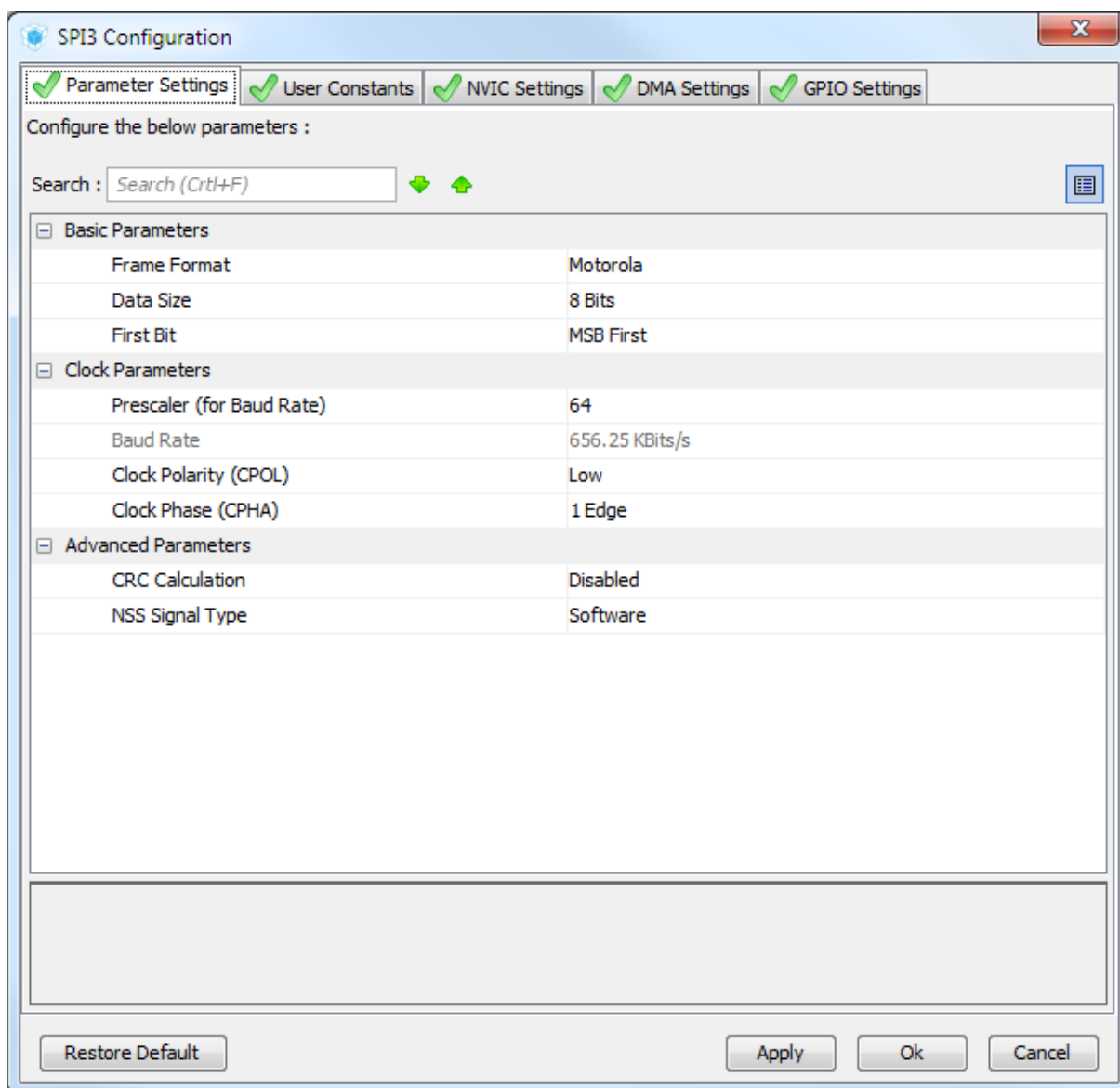
$$TimBit = 1 TimQua + BS1 TimQua + BS2 TimQua \quad (6)$$

Kombinací těchto hodnot je dobré podle zkušeností nastavit polohu Sample point okolo 75% a to právě odpovídá nastavení na předchozím obr. 6. Potom by CAN komunikace měla bez problémů v dané sběrnici fungovat. Z hlediska kódu je pro příjem zpráv použita funkce HAL\_CAN\_RxFifo0MsgPendingCallback(), která je spuštěna při přerušení. To znamená, že když přijde nová CAN zpráva a není nastaveným filtrem hardwarově odfiltrována, dojde k přerušení a provede se další část kódu. To znamená, že se uloží přijatá data ze zprávy do příslušné proměnné. Následné vyčítání jednotlivých hodnot z dat se již provádí v pomalejší smyčce programu pomocí dalších naprogramovaných funkcí dle potřeby. Naopak pro odesílání CAN zpráv je používána funkce HAL\_CAN\_AddTxMessage(), pro ni je zase zapotřebí

naskládat jednotlivé hodnoty proměnných podle pravidel normy do 64 bitových dat a následně odesílat podle požadované periody. Odesílání periodických zpráv je v jedné z hlavních pomalých smyček, realizovaných pomocí FreeRTOS. Tato smyčka tyto zprávy odesílá po 100 ms, pokud jsou vyžadovány.

## 2.5 SPI KOMUNIKACE

Jak je popsáno v různých literaturách [41], [42] a především v manuálu pro daný procesor RM0090 na stránkách [29]. Serial Peripheral Interface - SPI komunikace slouží ke komunikaci různých zařízení mezi sebou, kdy jedno zařízení je vždy master a ostatní jsou slave. Master generuje hodinový signál na pinu Serial Clock - SCK a tím jsou všechna zařízení sesynchronizována pro přenos dat v obou směrech. Pin MISO - Master In/Slave Out přijímá data ze zařízení slave a pin MOSI: Master Out/Slave In odesílá data pro zařízení slave. Poslední pin, který je potřeba užít při komunikaci přes komunikaci SPI, je digitální výstup. Tím master vybírá daný slave, se kterým chce zrovna komunikovat.



Obr. 7 Konfigurace SPI komunikace STM procesoru

Nastavení komunikace SPI pro procesor je zobrazeno na předchozím obrázku. Především je důležité se věnovat nastavení parametrů hodin procesoru a to především přenosové rychlosti. Ta je závislá na připojeném zařízení tedy slavu, kdy nesmí být překročena horní hranice, jinak by zařízení nekomunikovalo, nebo by zaslalo milné informace, než by byly požadovány. Hodnota přenosové rychlosti *BauRat*, nastavená pro procesor tedy master, lze spočítat z následující rovnice

$$BauRat = \frac{APB1}{PreSPI} \quad (7)$$

kde *APB1* je frekvence periferních hodin u užitého procesoru pro SPI3, ale např. pro SPI1 je užitá frekvence *APB2* a *PreSPI* – je prescaler pro stanovení přenosové rychlosti SPI komunikace. Dalšími důležitými parametry, aby zařízení správně komunikovala, je nutné nastavit Clock Polarity a Clock Phase tím je nastavena polarita hodin a poloha, kde je přenášený bit čten. Proto je nutné dobře prostudovat katalogový list příslušného slavu, zda danou kombinaci podporuje. Jak již bylo zmíněno pro výběr daného slavu je potřeba využít pin s nastavením digitálního výstupu, který je nastaven na hodnotu 1 a pokud jim je vybrán slave, tak musí mít hodnotu 0, pokud není, je hodnota 1. Pro přenos dat je používána základní funkce HAL\_SPI\_TransmitReceive(), která je zakomponovaná v dalších funkcích pro komunikaci s konkrétním zařízením slave.

Pomocí této komunikace jsou předávány v prototypu ECU data mezi dvěma moduly tj. flash pamětí a MEMS obvodem obsahujícím tříosý gyroskop a akcelerometr. Tato externí paměť je užívána ve vyvíjeném autonomním systému pro ukládání chybových zpráv DM1, DM2 atd. dle normy J1939-73. Tam jsou uchovávány aktivní a předchozí chyby systému. Ty lze přes diagnostické rozhraní načítat a mazat. Druhé zařízení je zde užito pro získávání sklonu vozidla pro příčný a podélný směr. Základem zpracování úhlové rychlosti a zrychlení je využití filtru popsaného na stránkách [43], kde výstupem zpracovaných dat je kvaternion. Pro další zpracování byly naprogramovány další funkce pro práci se souřadnými systémy pomocí kvaternionu, Eulerových úhlů a rotačních matic pro přepočítávání hodnot dle umístění ECU na úhel sklonu vozidla podél podélné a příčné osy. Pro systém je významná hodnota úhlu sklonu okolo příčné osy vozidla. Tím lze predikovat, kde se vozidlo nachází z hlediska sklonu vozovky a např. aktivovat pohon všech kol, aby byl hnací moment rozdělen rovnoměrně a nedocházelo k nadměrnému namáhání jen konkrétních hřídelí.

## 2.6 HLAVNÍ SMYČKA

Vstupní hodnoty pro řídicí algoritmus jsou ve většině případů zpracovávány softwarově nebo přes různá přerušování pro rychlé děje. Pomalejší děje jsou rozděleny do dvou smyček. Jedna již byla zmíněna v kapitole o CAN komunikaci, ve které jsou odesílány CAN zprávy. Nyní bude popsána hlavní smyčka, ve které řídicí algoritmus autonomně rozhoduje o reakci celého vyvíjeného systému na aktuální podmínky o případném prokluzu kol. V první fázi je potřeba popsat hlavní část projektu, kde se nachází funkce main. V této funkci jsou nejdříve načteny do proměnných z paměti procesoru nastavitelné parametry zobrazené v příloze P1. Potom je nastavena počáteční adresa, kde začíná hlavní kompilovaný program uložený v procesorové paměti. Také jsou nastaveny hodiny procesoru a jeho periférií, následuje nastavení digitálních pinů a smyčky pro kontrolu, zda je otočeno klíčkem zapalování. Pokud ano je pomocí elektronické součástky mosfetu spuštěno napájení i dalších částí ECU, následně jsou inicializovány dříve probrané nastavení pro CAN1, TIM5, ADC1 a SPI3. Potom jsou nastaveny filtry pro CAN komunikaci, inicializace nastavení externí paměti, kontrola aktuálních chyb před



posledním vypnutím klíčku zapalování a MEMS modulu s gyroskopem a akcelerometrem. V neposlední řadě jsou spuštěny funkce časovače pro měření otáček kol a nastavení FreeRTOS, přes který běží i hlavní smyčka.

Tato smyčka je nazvaná `FCE_Main_Loop()` a obsahuje další funkce. Její perioda je závislá na aktivaci možnosti čtení otáček kol z jiné ECU přes CAN komunikaci. Tedy pokud je tato možnost aktivní, tak je její perioda 20 ms a v ostatních případech 10 ms. První perioda vyplývá z periody přijímané zprávy s informacemi o otáčkách kol. Tato hlavní smyčka je také formulována také přes FreeRTOS.

První funkce uvnitř smyčky `FCE_Main_Loop()`, která je vyvolána, je funkce `Loop_ReadInputs()`, ve které jsou čteny digitální a analogové vstupy, které byly popsány dříve, pokud nejsou získávány přes přerušeni nebo jiným způsobem. Také zde jsou čteny digitální vstupy signalizující napájení na svorce 15, která odpovídá signálu ze zapalovacího klíčku, svorky 30, která je pro přívod stálého napětí a z elektronické části ECU, která je sepnuta až po otočení klíčku uživatelem.

Druhá funkce v řadě je `Loop_ReadGyro()` a pomocí SPI komunikace přečte hodnoty z MEMS modulu s gyroskopem a akcelerometrem. Jedná se o zrychlení, teplotu a úhlové rychlosti. Tyto hodnoty jsou vyhodnoceny a pomocí dalších funkcí jsou filtrovány do kvaternionu. Ten je převeden na rotační matici, ze které jsou vyhodnoceny úhly sklonu vozidla v závislosti na rotační matici polohy ECU.

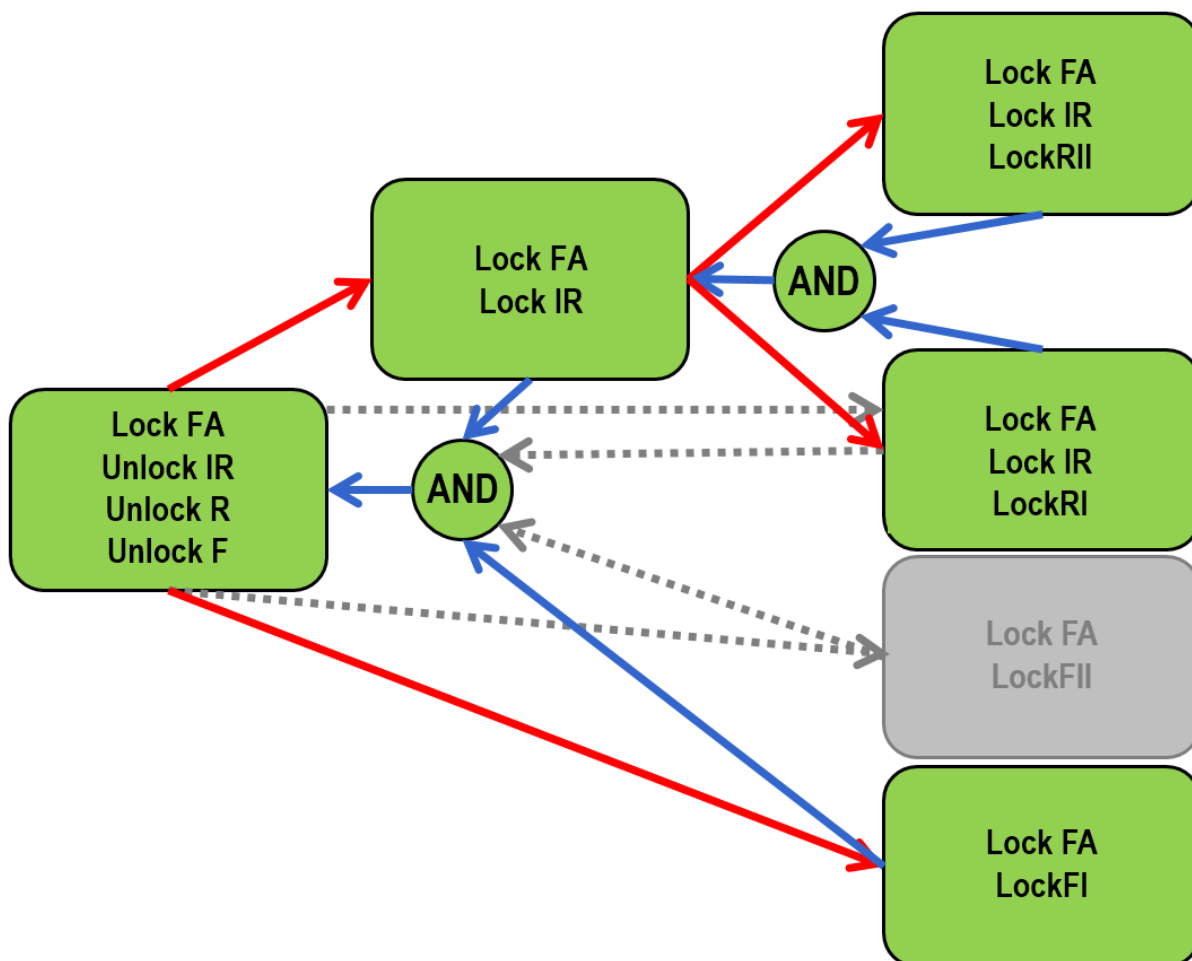
Třetí funkce `Loop_ReadCAN()` je určena pro získávání dat z databáze zpráv, které byly přijaty přes přerušeni. Tento proces může probíhat právě až v pomalejší smyčce, aby zbytečně další funkce v přerušeni neblokovaly další procesy. V rámci této funkce je také umístěn kód pro přijímání žádosti o změnu módu řídicího algoritmu od displeje a odesílána potřebná odezva.

Čtvrtá funkce `Loop_Algorithm()` je souhrnem hlavních funkcí řídicího algoritmu, kde je rozhodováno, zda bude aktivován pohon všech kol nebo bude uzavřen příslušný diferenciál. Na začátku funkce se přepočítá uživatelem zadaný obvod kol vozidla na jejich poloměr pro následující zpracování otáček kol, pokud jsou brány otáčky kol z CAN zprávy odesílané např. systémem ABS. Také jsou zde zpracovány periody získané z nezávislých snímačů otáček. Dále je implementována funkce pro inverzi hodnot zpětných vazeb z pneumatických válců, signalizující reálnou aktivaci pohonu všech kol nebo uzavření příslušného diferenciálu. To vyplývá z logiky elektrického zapojení přes pull-up rezistor a spínání země. Následuje zpracování již konkrétních hodnot proměnných z dat CAN zpráv nebo snímačů v závislosti na volbě uživatele. Dále jsou užity funkce pro vyhodnocení tlaku v pneumatickém okruhu, výpočet otáček na hřídelích hnacího traktu, výpočet rychlosti vozidla, vyhodnocení úhlu zatočení předních kol vozidla, výpočet limitních otáček, od kterých má fungovat výpočet korekčního skluzu  $slipC_i$ . To je, aby nedocházelo k matení řídicího algoritmu při jízdě zatáčkou, potom výpočet skluzu  $slip$  je formulován dle následující rovnice

$$slip_i = \frac{\max(omg_j, omg_k) - \min(omg_j, omg_k)}{\max(omg_j, omg_k)} - slipC_i, \quad (8)$$

kde  $omg$  jsou otáčky příslušného kola,  $i$  je index příslušného skluzu nebo korekčního skluzu například mezi pravým a levým kolem dané nápravy nebo příslušných hřídelí hnacího traktu,  $j$ ,  $d$  jsou indexy kol nebo hřídelí a nikdy nesmí být v rovnici stejného označení. Vždy se musí porovnávat např. dvě různá kola. V případě, že maximální hodnota z porovnávaných otáček je

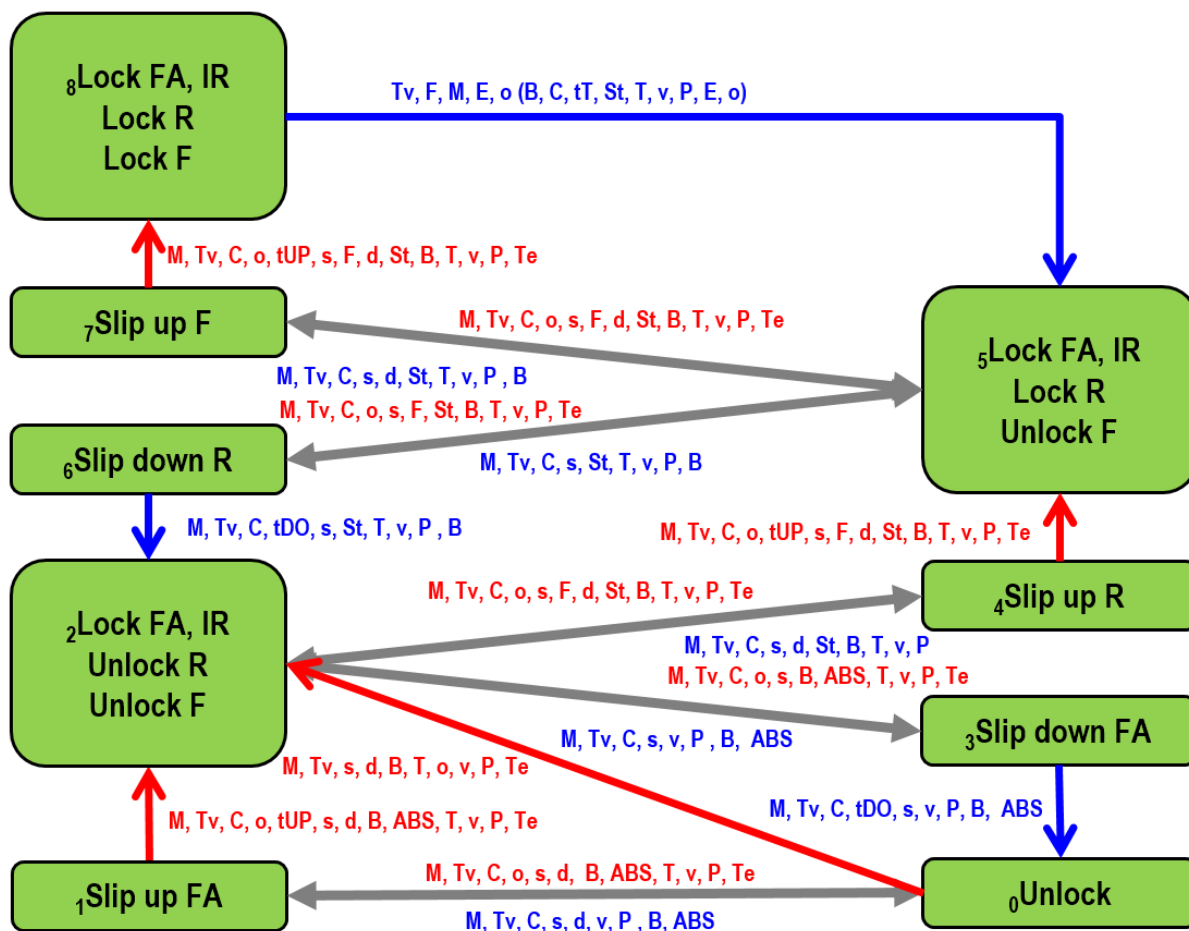
nulová, tak je tento případ ošetřen, aby nedocházelo k dělení nulou, a výsledný skluz je také nulový. Následují funkce pro nastavení řídicích parametrů dle zvoleného módu uživatelem systému, tedy módu pro silnici, polní cestu a terén/sníh. Pro bezpečnost a volbu automatického nebo manuálního módu je v algoritmu implementována funkce pro kontrolu změny hodnoty vstupního přepínače. Tato funkce kontroluje, zda má vozidlo nulovou rychlost po stanovený časový úsek, potom je případná změna na přepínači brána v potaz i v řídicím algoritmu. Je to z toho důvodu, kdyby uživatel zapomněl sepnuté přepínače pro manuální řízení a zároveň přepnul v nějaké rychlosti vozidla přepínač na manuální řízení. Potom by mohlo dojít k destrukci částí hnacího traktu nebo k nebezpečné jízdě situaci. Ještě než je spuštěna funkce Algorithm() je potřeba zkontrolovat, zda systém nemá nějakou poruchu a to vše probíhá ve funkci Error(), která bude podrobněji popsána v následující kapitole. Potom tedy následuje funkce Algorithm(), která vyhodnocuje reakci na aktuální chování vozidla. V rámci vývoje byly testovány dva přístupy řídicího algoritmu. Jeden byl jako částečně nezávislý a je zobrazen na následujícím obrázku.



Obr. 8 Částečně nezávislé řízení uzavírání diferenciálu pro verzi vozidla s pohonem 6x6

Pod pojmem částečně nezávislé řízení si lze představit to, že v první fázi, pokud dochází k prokluzu, algoritmus odešle signál pro aktivaci pohonu všech kol -  $LockFA_i$  a následně již nezávisle uzavírá zadní mezinápravový diferenciál -  $LockIR$  a přední osově diferenciály -  $LockF_i$ . Po uzavření zadního mezinápravového diferenciálu, a pokud v daném vozidle je, tak jsou nezávisle uzavřeny zadní osově diferenciály -  $LockR_i$ . Zobrazené schéma je pro verzi

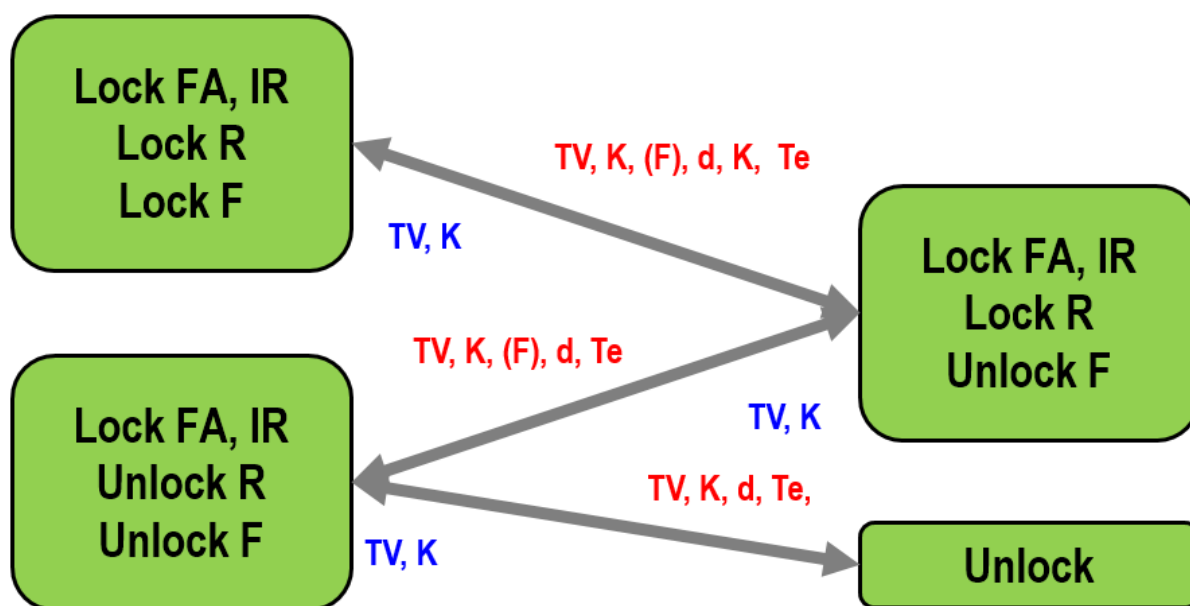
vozidla s pohonem 6x6, proto část schématu je vybarvena šedě, jelikož není pro daný typ vozidla aktivována. Řídicí algoritmus tohoto formátu byl programován v softwaru Simulink. Tento postup byl testován jen v prostředí Model in the Loop – MIL. A po vyhodnocení některých jízdních situací a především složitosti se přešlo na jednodušší a z hlediska užití modulárnější postup uzavírání. Jedná se o interně nazvané závislé uzavírání diferenciálů. To je zobrazeno na následujícím obrázku.



Obr. 9 Závislé řízení uzavírání diferenciálů

V tomto případě je v první fázi aktivován pohon všech kol a uzavření mezinápravových diferenciálů. V další fázi se uzavřou všechny zadní osově diferenciály a v poslední fázi jsou uzavřeny i všechny přední osově diferenciály. Toto schéma bude popsáno podrobněji, podle kterých řídicích veličin je postupováno ve schématu směrem k uzavření všech diferenciálů a aktivaci pohonu všech kol. Když je spuštěna funkce Algorithm(), nejdříve je rozhodováno o tom, zda není v systému nějaká porucha vyhodnocena ve funkci Error(). Pokud je nějaká porucha v systému, tak dle její závažnosti bude nastaven řídicí algoritmus do jedné ze tří možností. To je stav, kdy řídicí algoritmus může fungovat jak v manuálním, tak automatickém režimu. Druhým stavem je, že je porucha závažnější a je na ní významně závislé automatické řízení. Potom je systém přepnut do stavu manuálního řízení, a pokud jsou poruchy považovány za velice nebezpečné, tak je systém deaktivován. Pokud systém nemá žádnou nebo málo závažnou poruchu, tak se řídí dle nastavené hodnoty přepínače pro automatické nebo manuální řízení, které nastavuje uživatel. V případě manuálního řízení probíhá uzavírání dle schématu, zobrazeném na následujícím obr. 10. Hlavní řídicí proměnou je hodnota vstupních přepínačů a

spínače -  $K$  na palubní desce. Řidič v první fázi aktivuje pohon všech kol  $Lock\ FA$  a mezinápravové diferenciály  $Lock\ IR$ , potom uzavře zadní osově diferenciály  $Lock\ R$  a nakonec stiskne tlačítko pro uzavření předních osových diferenciálů  $Lock\ F$ . Dalšími proměnnými je velikost hnacího momentu motoru  $Te$ , kdy nesmí být překročena nastavená hodnota a hodnota rozdílu otáček  $d$  mezi hřídelemi, kde dochází k jejich spojování, tedy uzavírání diferenciálu. Tyto dva limity pro rozdíl otáček a hnací moment jsou aplikovány z důvodu bezpečnosti, aby nezkušený řidič neuzavíral diferenciály za situací, kdy by to mohlo být pro hnací trakt destruktivní. Dalším řídicím parametrem pro manuální řízení je proměnná obsahující identifikátor typu vozidla  $TV$ . To znamená, zda se jedná o vozidlo s pohonem 4x4, 6x6, 8x8 nebo 10x10. Hodnota proměnné  $F$ , tedy zpětná vazba reprezentuje reálný stav aktivace pohonu všech kol nebo uzavření individuálního diferenciálu. Ve schématu je v závorce, jelikož tuto funkci může mít i užitý pneumatický okruh ve vozidle. Jedná se o to, že pokud chce uživatel postoupit ve schématu směrem k uzavírání diferenciálů, musí být nejdříve aktivován pohon všech kol atd. Hodnota proměnné  $typ\ vozidla\ Tv$  nastavuje počty aktivovaných pohonů předních náprav a počty uzavíraných diferenciálů pro daný typ vozidla. Ještě obecně k tomu schématu, pokud jsou proměnné označeny červeně, znamená to, že platí pro postup schématem směrem nahoru. Pokud jsou označeny modře, jedná o postup schématem dolů, tedy k deaktivaci pohonu všech kol a otevření všech diferenciálů. V daných buňkách je vždy zobrazeno to, co by mělo být v té poloze algoritmu uzavřeno nebo aktivováno.



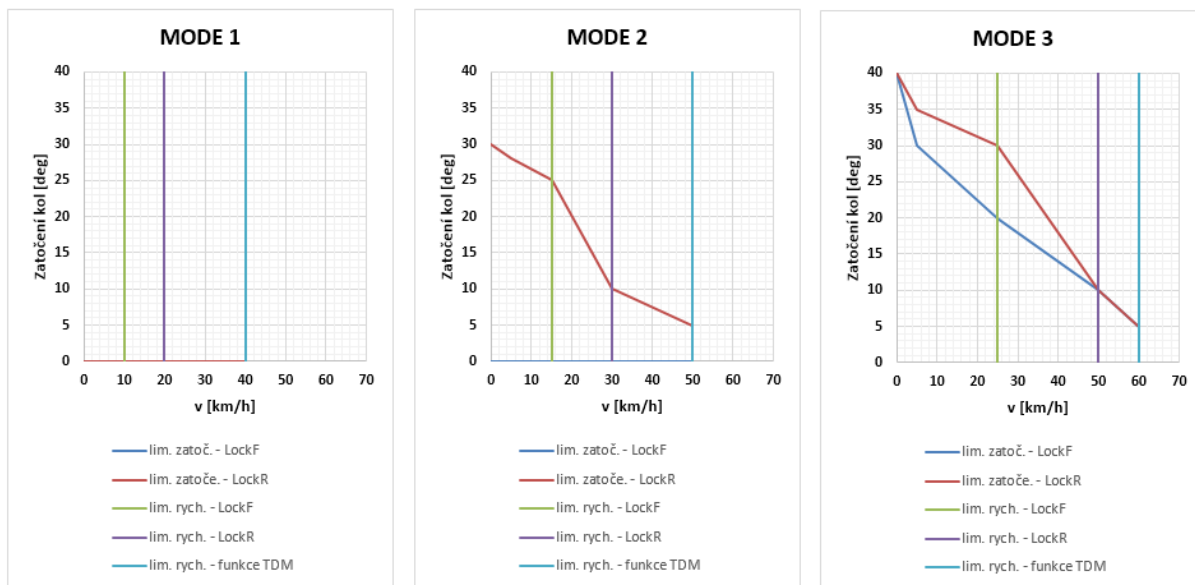
Obr. 10 Manuální řízení uzavírání diferenciálu

Obecně by uživatel měl mít defaultně nastaveno automatické řízení, aby se o vše autonomně postaral řídicí systém. Manuální řízení by uživatel měl používat jen v případě, pokud je to nevyhnutelné nebo pokud systém hlásí poruchu a přepne řízení do manuálního režimu. Automatické řízení pracuje podle schématu na obr. 9. Tam jsou zobrazeny proměnné, podle kterých je vyhodnocována daná situace. To jsou typ vozidla -  $Tv$ , hodnota rozdílu otáček -  $d$ , procentuální hodnota hnacího momentu motoru -  $Te$ , zpětná vazba informující o stavu pohonu všech kol a diferenciálů -  $F$ , hodnota brzdového pedálu -  $B$ , hodnota spojkového pedálu -  $C$ , hodnota plynového pedálu -  $T$ , error - porucha systému -  $E$ , hodnota aktivace ABS -  $ABS$ , mód nastavení (silnice, polní cesta, terén/sníh) -  $M$ , otáčky kol/hřídelí dle polohy ve schématu řídicího algoritmu -  $o$ , tlak v pneumatickém okruhu -  $P$ , skluz mezi jednotlivými koly nebo

hřídely –  $s$ , úhel natočení předních kol –  $St$ , čas, po který řídicí algoritmus čeká, než může sestoupit ve schématu –  $tDO$ , čas testování, po kterém řídicí algoritmus otevře přední diferenciály –  $tT$ , čas, po který řídicí algoritmus čeká, než může vystoupit ve schématu nahoru –  $tUP$  a rychlost vozidla –  $v$ . Samozřejmě daným proměnným vždy figuruje nějaký limit nebo hodnota, vůči které jsou porovnávány.

Nyní bude popsána základní filozofie fungování řídicího algoritmu. Schéma je rozděleno na 9 úrovní, které odpovídají např. aktivaci pohonu všech kol *LockFA*, nebo uzavření příslušných diferenciálů, případně čekání, zda situace prokluzu přetrvává určitý časový interval, aby mohlo být ve schématu postupováno výše. V první fázi je pohon všech kol deaktivován a jednotlivé diferenciály otevřeny. Pokud vozidlu začne v nepříznivých adhezních podmínkách prokluzovat kolo a vyhodnocený slip/prokluz je nad stanovený limit, je postoupeno ve schématu algoritmu na úroveň 1 „Slip up FA“. Aby opravdu bylo ve schématu opravdu přejito na tuto úroveň, je potřeba splnit další podmínky a to je, že nesmí být sešlápnut spojkový pedál, jinak si algoritmus myslí, že je řazeno v závislosti na rychlosti vozidla. To je z důvodu, aby při řazení nedocházelo k uzavírání diferenciálů nebo naopak nedocházelo ke zbytečnému otvírání diferenciálu při přeřazování, kdy se otáčky a tím vyhodnocovaný skluz může dostat pod limitní hodnotu skluzu, přestože nepříznivé adhezní podmínky přetrvávají. Proměnná  $o$  zastupuje kontrolu maximálních a minimálních otáček ze všech kol. Je to pro případ, kdy vozidlo jede velice pomalou rychlostí, anebo ještě při tom zatáčí a pomocí snímačů otáček nebo z informací CAN zprávy o rychlostech kol není možné získávat tyto informace. Potom by si algoritmus myslel, že se jedná o prokluz, když by informace o otáčkách některých kol ukazovala hodnotu nula. Dále nesmí být brzděno jakýmkoliv způsobem, to znamená, že v této proměnné je zahrnuto i brzdění motorovou brzdou, retardérem nebo pakovací brzdou. Dle nastavení ECU je zde i zahrnut vliv aktivace ABS. Je to z důvodu ovlivňování právě funkcí asistenčních systému pro brzdění. Hodnota plynového pedálu musí být hodnoty jedna, tedy plynový pedál musí být sešlápnut na minimální úrovni a tím ví algoritmus nejenom, že není brzděno motorovou brzdou, ale také že případný prokluz kol je odůvodněný. Hodnota rychlosti určuje, zda ještě může být systém aktivní, aby nedocházelo k uzavírání diferenciálů při vysokých rychlostech vozidla a nevedlo to k jeho nestabilizaci ovladatelnosti. Hodnota tlaku v pneumatickém okruhu je limitována tak, že pokud je tlak pod stanovenou hladinou, není pokračováno v postupu schématem výš, aby mechanické části konkrétně čela zubových spojek nebyly jen obrušovány vlivem malé ovládací síly. Hodnota hnacího momentu motoru musí být pod hodnotou stanoveného parametru a je to kvůli ochraně hnacího traktu před momentovými rázy. Pokud je vše splněno, řídicí algoritmus postoupí ve schématu do bloku „Slip up FA“. Tam probíhá stejná kontrola, plus je sledována nastavená doba přetrvávání prokluzu např. mezi koly. Pokud i tato podmínka je splněna, je ve schématu postupováno na úroveň 2, kde již je odeslán výstupní signál akčnímu členu pro aktivaci pohonu všech kol a případně uzavření mezinápravových diferenciálů. Pokud v rámci čekání není nějaká podmínka splněna, tak se algoritmus vrací na nultou úroveň ve schématu. Druhou možností jak se dostat na úroveň 2 je, že vozidlo stojí v kopci a řidič se začne rozjíždět, ale zároveň používá nějakou formu brzdění (brzdový pedál nebo parkovací brzda). Tím algoritmus rozezná, že vozidlo se rozjíždí ve složitější situaci než je rovina a aktivuje pohon všech kol a mezinápravové diferenciály. U novější verze autonomního systému je stav stojícího i jedoucího vozidla kontrolován dalším vyvíjeným systémem pro predikci podmínek, kde se vozidlo nachází. Tedy je kontrolován náklon vozidla podél jeho příčné osy pomocí gyroskopů a akcelerometrů. Tato funkce je prioritně výše než dříve kontrolované hodnoty. Tím je přechod na úroveň 2 stanoven při překročení nastavené hodnoty náklonu vozidla a výhodou je rovnoměrné rozkládání hnacího momentu na všechna kola. Tím nejsou zatěžovány jen individuální hřídele a je zabráněno jejich namáhání, které by

mohlo být až na hraně meze kluzu. Pro další postup ve schématu na úroveň 4 musí být kromě dříve zmiňovaných splněny další podmínky z hlediska zatočení kol. To znamená, že jsou nastaveny limity závislé na rychlosti vozidla a zatočení kol, které určují, zda příslušné osově diferenciály mohou být uzavřeny, aby nebyl poškozen hnací trakt. Tyto limity jsou zobrazeny na následujícím obrázku.



Obr. 11 Nastavení limitů pro natočení kol a rychlostí vozidla pro daný mód

Tam jsou zobrazeny i nastavené limity pro rychlosti vozidla, kdy po jejich překročení dané diferenciály nemohou být uzavřeny. S tím souvisí i nastavení módů, pokud je nastaven mód silnice, řídicí algoritmus postupuje ve schématu maximálně do úrovně 2. Pokud je nastaven mód polní cesta, tak je postupováno maximálně na úroveň 5 a pokud je nastaven mód terén/sníh, tak řídicí algoritmus může procházet kompletním schématem. Pokud se vrátíme k popisu postupu na úroveň 4, je zde také zahrnuta kontrola zpětných vazeb a to tak, že aspoň jeden z dříve aktivovaných pohonů předních náprav nebo mezinápravových diferenciálů musí být reálně uzavřen. Pokud jsou podmínky splněny, postupuje se výše. V případě, že nejsou splněny podmínky, sestupuje se v algoritmu dolů přes úroveň 3 až na 0. To je i v případech, kdy se začne brzdit, jelikož osově diferenciály nesmí být uzavřeny při brzdění z hlediska stability vozidla. Pokud je algoritmus na úrovni 4, jsou sledovány stejné signály zase po určitý časový interval. Při splnění je postoupeno na úroveň 5 nebo naopak je sestoupeno na úroveň 2. Z úrovně 5 jsou zachovány stejné postupy pohybu ve schématu jen s odlišným nastavením limitních parametrů. V případě, že nepříznivé adhezní podmínky přetrvávají dál, postupuje algoritmus až na úroveň 8, která je vrcholem schématu. V tu chvíli je aktivován pohon všech kol a jsou uzavřeny všechny diferenciály. Zároveň algoritmus nemá žádné signály, podle kterých by určil, zda prokluz vozidla stále probíhá, z toho důvodu je zde naprogramována tzv. testovací smyčka. Pokud některá jiná hodnota neurčí, že řídicí algoritmus sestoupí na úroveň 5, tak je čekáno po stanovený časový interval a potom se přechází na úroveň 5 testovací smyčkou. Následně je vyhodnocen stav, a pokud dochází stále k prokluzu vozidla, pokračuje se zase nahoru až na úroveň 8, pokud není, tak je ve schématu pokračováno směrem dolů a tím je předpokládáno, že vozidlo např. projelo bahnitým terénem a nyní již zásah systému není nutný. V této testovací smyčce mají vyšší prioritu poruchy daných předních osových diferenciálů nebo signál brzdění, tím není čekáno na uplynutí časového intervalu, ale je postupováno níže na úroveň 5 okamžitě. Tento pohyb ve schématu probíhá neustále v rámci hlavní smyčky dle náročnosti oblasti, kde

se vozidlo pohybuje. Po této funkci jsou ještě nastaveny hodnoty signálu do příslušné polohy v datech pravidelně odesílaných CAN zpráv. Během těchto rozhodujících procesů, může být i odesílána CAN zpráva TSC1, která omezí otáčky motoru. Tím jsou částečně vyrovnány i otáčky hřídeli, které se v podstatě spojují a tím se předchází vzniku momentových rázů při uzavírání diferenciálů. Tato funkce je volitelná a nastavitelná uživatelem a také záleží na tom, zda je daným motorem podporována.

Další funkce `Loop_WriteOutputs()` v hlavní smyčce nastavuje hodnoty výstupních pinů, které jsou nastaveny jako digitální výstupy. Po vyhodnocení aktuální situace ve funkci `Loop_Algorithm()` a nastavení výstupních hodnot pro aktivaci pohonu všech kol nebo uzavření příslušného diferenciálu jsou právě tyto hodnoty nastaveny i na výstupních pinech procesoru a tím přes další elektronické části ECU jsou spínány nebo vypínány akční členy.

Funkce `Loop_ConvertCAN()` naplní data CAN zpráv, které mají být pravidelně odesílány při testování a vývoji. Takže jednotlivé hodnoty proměnných jsou umístěny do datového pole o velikosti 64 bitů na příslušnou pozici se stanovenou délkou. Jedná se o CAN zprávy určené pro vývoj, které jsou odesílány jen na vyžádání. To znamená zprávy určené především pro výzkum a vývoj, kde se nachází vnitřní proměnné řídicího algoritmu, aby bylo možné kontrolovat vnitřní funkce při dalším rozšiřování nebo drobných úpravách stávajícího kódu.

Funkce `Loop_WriteCAN()` již připravená data umístí do databáze odesílaných zpráv ke konkrétnímu ID zprávy, tím jsou kompletní zprávy CAN připraveny na odeslání, které probíhá v paralelní smyčce k hlavní smyčce `FCE_Main_Loop()`.

Poslední položkou hlavní smyčky je, že pokud je v průběhu vypnut klíček zapalování, uloží se potřebná data do externí paměti z hlediska případných chyb systému a procesor se restartuje, tím se spustí druhý program tzv. bootloader umístěný v paměti procesoru, který slouží k zavádění nového programu do paměti procesoru nebo nastavování nastavitelných parametrů ECU uživatelem. Pokud se dál nic neděje, tak zároveň použitý pin procesoru pro signál ze svorky 15 uspí procesor, aby při vypnutém klíčku byla minimální spotřeba elektrické energie na provoz ECU. Po opětovném zapnutí klíčku se systém probudí a spustí se nejdříve program pro nahrávání nových verzí programu s řídicím algoritmem, a pokud nejsou vyžadovány tyto nebo další úkony přechází procesor v paměti na začátek hlavního programu s algoritmem. Tento proces se opakuje vždy v závislosti na poloze startovacího klíčku.

## 2.7 CHYBY A LIMITY

V hlavní smyčce `FCE_Main_Loop()` se nachází hlavní část řídicího algoritmu s názvem funkce `Loop_Algorithm()`, která dále obsahuje další funkce popsané v předchozí kapitole. Také se tam nachází funkce `Error()`, která bude popsána nyní. Tato funkce předchází funkci `Algorithm()` a je to z toho důvodu, aby se prvně vyhodnotily případné poruchy řídicího systému a následně se podle toho řídicí algoritmus zachoval. Základem je několik vektorů obsahujících informace o poruchách a limitech systému. Základem je vektor se 176 hodnotami, který obsahuje veškeré poruchy a limity systému. Na jeho danou pozici je zapisovaná příslušná porucha nebo limit a jejich hodnota zároveň odpovídá jejich prioritě. Nejvyšší prioritu mají poruchy, při kterých je systém deaktivován v obou režimech řízení (automatické, manuální). Mezi ně patří fatální chyba systému, to znamená poruchy z hlediska špatného načtení parametrů systému, parametrů nastavitelných uživatelem, případných poruch uložených z externí paměti nebo chyba inicializace částí programu. Dále jsou kontrolovány napětí v ECU a je kontrolována hodnota tlaku v pneumatickém okruhu. Pokud je hodnota tlaku pod nastaveným limitem a po stanovenou

dobu, tak je předpokládána porucha snímače nebo pneumatického okruhu. V rámci kontroly je ošetřen stav, kdy vozidlo delší dobu stálo a v zásobníku vzduchu klesne tlak pod stanovenou úroveň. Pokud je však již vozidlo delší dobu v provozu, kontrola hodnoty probíhá již bez omezení. Také je kontrolován stav snímače tlaku mimo jeho rozsah. Pokud by nějaká ze zmíněných poruch nastala, mohlo by dojít k poškození vyvíjeného systému, hnacího traktu nebo dokonce k ohrožení života v krajních situacích. Proto je systém při těchto poruchách deaktivován. Nyní budou popsány poruchy ve druhé oblasti, kdy systém zůstává funkční v manuálním režimu, ale v automatickém režimu je deaktivován. Zde je kontrolováno odpojení akčního členu pomocí funkce dané elektrickým obvodem. To znamená např. utržení kabelu. Důležitou poruchou, která je kontrolována, je kontrola snímačů otáček. To probíhá tak, že pokud nejvyšší otáčky ze všech snímaných kol překročí nastavený limit a některé z ostatních kol má hodnotu nula nebo pod stanovený limit, tak je předpokládáno, že je porucha snímače otáček. Kontrola snímače natočení volantu probíhá ze součtu jeho dvou signálů. Jejich hodnoty se mění v závislosti na poloze tak, že jedna hodnota stoupá a jedna klesá. V součtu by vždy měly odpovídat hodnotě napájení, která je dána 5 V. Tedy kontrolní kód kontroluje součet signálů a pokud je mimo stanovený rozsah, tak je detekována porucha. Tím je detekováno, že daný snímač je i mimo jeho rozsah. Jak bylo zmíněno, z displeje přichází pro ECU signál nastaveného módu. Pokud daná CAN zpráva obsahující informace o komunikaci nepřijde znovu do stanoveného času, jedná se o poruchu. S tím i souvisí další přijímané CAN zprávy s hodnotami hnacího momentu motoru, signálů pedálů, parkovací brzdy, motorové brzdy, retardéru a aktivace ABS, které jsou také časově kontrolovány a to zhruba v trojnásobku doby jejich odesílané periody. Takže pokud nepřijde nová konkrétní CAN zpráva, tak je detekována porucha. V případě užití signálu přímo ze snímačů pedálů, je kontrolována změna jejich hodnoty po dlouhý časový úsek, a pokud nedojde k jejich sešlápnutí, je opět signalizována porucha. Tím se uzavírá oblast poruch, kdy funguje jenom manuální řízení. A poslední oblast z poruch je ta, kde poruchy jsou detekovány, ale systém není v obou režimech nijak omezen. A jedná se o detekci poruch z hlediska aktivace/deaktivace pohonu všech kol nebo uzavření/otevření diferenciálu. Tím je kontrolován akční člen a spínač zpětné vazby o reálném uzavření zubových spojek. Souhrnem těchto detekovaných poruch by měla být pokryta celá oblast možných poruch systému a tím předejití poškození systému hnacího traktu nebo nebezpečí úrazu.

Z hlediska řídicího algoritmu je spínání akčních členů v některých případech limitováno, aby nedošlo k destrukci hnacího traktu a to i v případě manuálního řízení, kdy zkušenosti řidičů jsou v dnešní době dost často nedostatečné. Proto je řidič informován o případných limitech. Je kontrolováno omezení z hlediska přenášeného hnacího momentu motorem. Pokud je jeho hodnota nad limit, algoritmus neodešle signál pro aktivaci akčního členu a navíc je tento limit na displeji signalizován řidiči. I když tyto limity mají také nastavenou prioritu, tak není důležitá, jelikož se jedná jen o informování řidiče, proč v danou chvíli došlo k nějakému omezení. Dále důležitým limitem je kontrola rozdílů otáček hřídelí, které jsou akčními členy spojovány a tím dochází např. k uzavření diferenciálu. Je to z toho důvodu, aby při velkém rozdílu otáček nevznikl velký rázový moment při spojování zubových spojek diferenciálů. Také bylo zmíněno při popisu řídicího algoritmu, že je zde omezení z hlediska zatočení předních kol, proto je zde funkce pro informování řidiče z hlediska tohoto limitu. A posledním kontrolovaným limitem je rychlost. To znamená, pokud by vozidlo jelo nad stanovený limit rychlosti, kdy je systém již deaktivován, a přesto by vznikl nějaký prokluz, řidič o tom bude informován.

Užitím daných funkcí pro vyhodnocení poruch nebo limitů jsou dané výstupy zapsány do jednoho vektoru. Tento vektor je dále zpracováván a je seřazen dle priorit do menšího vektoru



již jen o velikosti osmi hodnot. Tento vektor je odeslán v CAN zprávě, která slouží pro kontrolu při vývoji a testování. Dle hlavního vektoru jsou také poruchy aktivovány ve vektorech, které jsou následně ukládány do externí paměti a reprezentují aktivní a dříve aktivní poruchy DM1 a DM2. Proto při inicializaci ECU jsou prvně tyto poruchy přečteny z externí paměti a nastaveny do všech vektorů. Příkladem může být stav, kdy v systému byla nějaká aktivní porucha, řidič vypnul klíček startování vozidla a tato porucha byla uložena v externí paměti jako aktivní. Při opětovném nastartování se nejdříve aktivní porucha načte do všech vektorů a funkci s ní spojených. Proto v tu chvíli systém signalizuje řidiči poruchu, ale pokud při kontrole této chyby je zjištěno po nějaké době provozu, že již nepřetrvává, tak je její stav ve vektorech změněn a také je to zapsáno do externí paměti. Tím je chyba smazána v aktivních chybách a je zapsána do dříve aktivních poruch. Potom při vyčtení chyb diagnostikou, bude tato chyba uživateli zobrazena v DM2. Poslední funkci je zpracování poruch a limitů do jedné proměnné o třech stavech, která je odesílána pomocí CAN zprávy displeji. To znamená, že na displeji je zobrazeno, že systém je v pořádku, dochází k nějakému omezení, nebo má systém poruchu.

## 2.8 VÝSTUPY

Jedná se o digitální výstupy, které ovládají výkonné elektronické součástky. Hodnota tohoto výstupu z procesoru je nastavována pomocí předdefinované funkce `HAL_GPIO_WritePin()`. Ovládané elektronické součástky zajišťují ovládání akčních členů s požadovanými parametry napětí a proudu. Je předpokládáno užití akčního členu ve formě elektroventilu, který pouští tlakový vzduch do pneumatického válce pro mechanické ovládání zubových spojek. Nastavení pinů procesoru bylo dle defaultního nastavení v projektu generovaném STM Cube MX softwarem.

## 2.9 BOOTLOADER

Bootloader je tzv. zařízení nebo program pro zavádění hlavního programu obsahující řídicí algoritmus a vše s ním související. Při vývoji a běžném provozu bylo použito nahrávání programu užitím dvou přístupů. Prvním přístupem je připojením hardwarového bootloadru k pinům procesoru k tomu určených. V tomto případě bylo využito programování procesoru pomocí Serial Wire Debugging – SWD. Kdy potřebné piny byly nastaveny v projektu a pomocí hardwaru ST-Link [44] nebo J-Link [45] byl program implementován do procesoru. To má značnou výhodu pro debuggování programu, tedy pro sledování všech procesů probíhajících v procesoru při běhu nebo krokování vyvinutého programu s řídicím algoritmem. Tento postup je velice významný při vývoji, avšak pro běžné užívání vyvíjeného systému zde bylo naprogramováno druhé rozhraní pro implementaci novější verzi programu koncovým uživatelem. To je omezeno jen právě na implementaci nové verze. Proto byl naprogramován tzv. zaváděcí program, který je v porovnání s řídicím programem velice jednoduchý, aby se eliminoval případný počet chyb vytvořených programátorem. Tento program je umístěn na začátku paměti procesoru a byl naprogramován tak, aby reagoval na komunikaci CAN. To znamená, že uživatel je schopen zavést novou verzi hlavního programu pomocí CAN komunikace do paměti procesoru. K tomuto účelu bylo naprogramované diagnostické rozhraní komunikující s hardwarem PCAN [46]. Toto zařízení stačí připojit do CAN sítě vozidla a pomocí příkazu lze odeslat novou verzi hlavního programu do procesoru, který ji uloží dle definice na příslušné místo v jeho paměti a pokud vše proběhne v pořádku, tak je ECU restartována. Potom je systém spuštěn s novým programem tak, že nejdříve je znovu v procesoru spuštěn zaváděcí program a pokud do stanoveného časového intervalu nepřijde z diagnostického softwaru žádný příkaz, přechází procesor na počáteční adresu hlavního programu. To je zobrazeno i na následujícím obrázku, kde zaváděcí program je od adresy

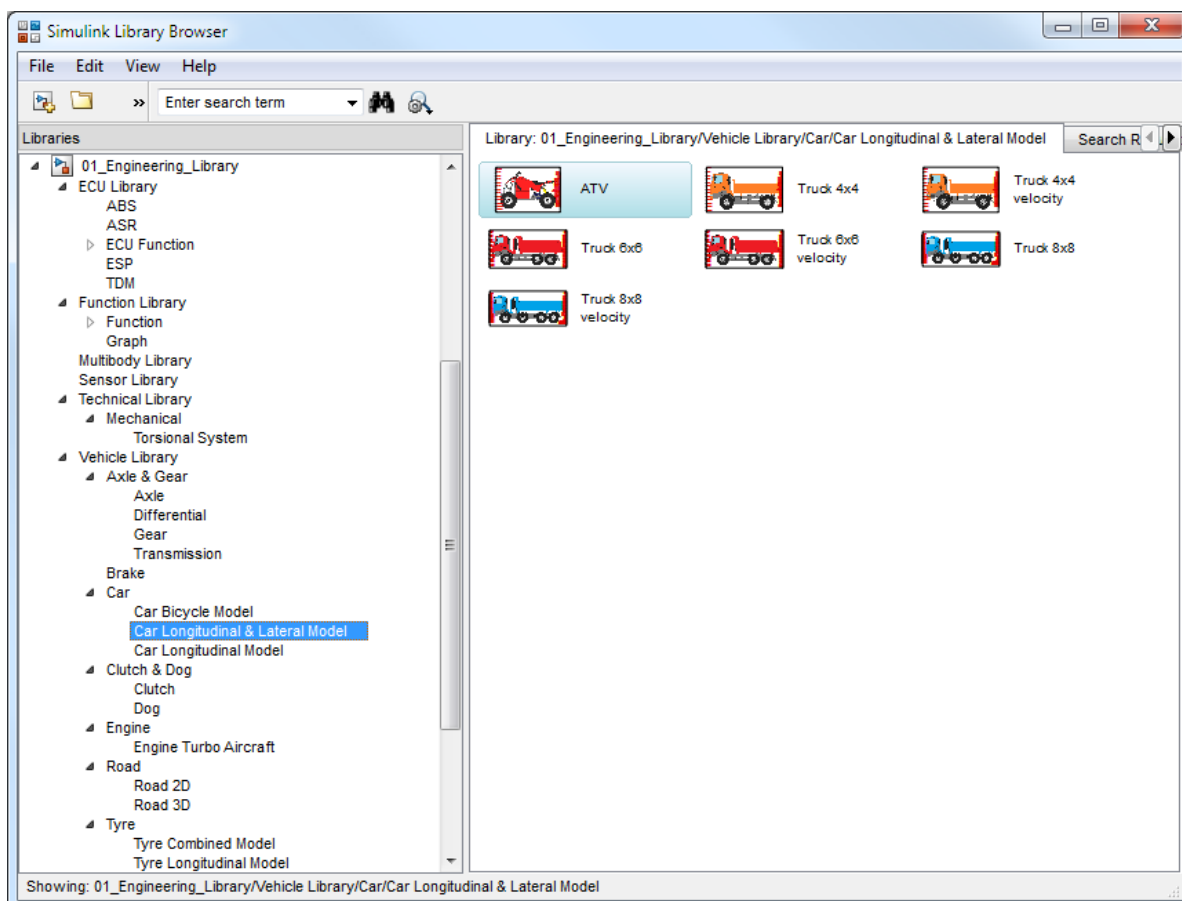
paměti 0x8000000, parametry nastavitelné uživatelem od adresy 0x800C000 a hlavní program od adresy 0x8020200 jelikož na adrese 0x8020000 je umístěn kontrolní byte signalizující správnost nahrání hlavního programu. Výhodou tohoto je, že nová verze může být zákazníkovi odeslána elektronicky, aniž by ho dodavatel musel navštívit. Proto je vyvinutý diagnostický software možno dodat s hardwarem PCAN, pokud ho již zákazník nepoužívá. Diagnostický software a jeho funkce bude popsána v jedné z následujících kapitol.

Paměť procesoru		Sector [-]	Size [K/KB]	byte [B]	kibibyte [K/KB]	kibibyte [K/KB]	byte [B]	
ALGORITHM	80FFFF	11	128	131072	128	896	917504	
	80E0000							
	80DFFFF	10	128	131072	128			
	80C0000							
	80BFFFF	9	128	131072	128			
	80A0000							
	809FFFF	8	128	131072	128			
	8080000							
	807FFFF	7	128	131072	128			
	8060000							
	805FFFF	6	128	131072	128			
	8040000							
	803FFFF	5	128	131072	128			
	<b>8020000</b>							
PARAMETERS	801FFFF	4	64	65536	64	80	81920	
	8010000							
	800FFFF	3	16	16384	16			
	<b>800C000</b>							
BOOTLOADER	800BFFF	2	16	16384	16	48	49152	
	8008000							
	8007FFF	1	16	16384	16			
	8004000							
	8003FFF	0	16	16384	16			
	<b>8000000</b>							
			1024				1024	1048576

Obr. 12 Rozdělení paměti procesoru na jednotlivé sekce pro program se zavaděčem hlavního programu, parametry nastavitelné uživatelem a hlavní program s řídicím algoritmem.

### 3 VÝVOJ VÝPOČTOVÝCH MODELŮ PRO MIL TESTOVÁNÍ

Aby bylo možné testovat vyvíjené řídicí algoritmy, prochází vývojem i vlastní knihovna obsahující bloky s matematickými modely zastupující jednotlivé části vozidel. Pro tvorbu bloků bylo vycházeno z literatury [63], [64] a [65], kde je popsána tvorba modelu v softwaru Simulink a obecný matematický popis částí hnacího traktu. Zde budou popsány jen základní funkce knihovny s bloky, jelikož to není hlavním výstupem této práce a podrobnější popis je v doktorské práci [47] nebo ve vydaných člancích [48], [49], [50], [51], [52], [53], [54], [55] a [56].



Obr. 13 3Knihovna s bloky pro sestavování výpočtových modelů vozidel

Knihovna je zobrazena na předchozím obrázku. A je tvořena v prostředí softwaru Simulink. Je to software učen pro tvorbu a testování různých výpočtových modelů a řídicích algoritmů a pro případné generování jejich kódů pro příslušná zařízení a další aplikace. Tento software obsahuje řadu knihoven, které jsou v základu, nebo je lze dokoupit. V případě, že uživatel tohoto softwaru nenajde potřebné bloky pro sestavení dle jeho požadavků, může si vytvořit vlastní knihovny. A to bylo i v tomto případě, kdy je vytvořena rozsáhlá knihovna s bloky pro sestavování výpočtových modelů vozidel a pro simulaci jejich dynamiky, řešení vibrací a především pro testování vyvíjených mechatronických systémů. Celá knihovna se dělí na další podknihovny, které budou popsány následně.

### 3.1 KNIHOVNA ECU

Jedná se o knihovnu, která obsahuje bloky reprezentující různé vyvíjené řídicí algoritmy. Řídicí algoritmy, včetně zde popisovaného, jsou programovány v jazyce C, C++, nebo v softwaru Simulink. Ty jsou tvořeny z jeho základních bloků a následně sestavovány do tzv. subsystémů tedy bloků reprezentujících daný algoritmus. V případě řídicích algoritmů daného systému tvořených v jazyce C/C++ jsou tyto kódy implementovány do knihovny pomocí tzv. S-funkcí. Tato knihovna také obsahuje univerzální bloky s funkcemi, které se v řídicích algoritmech používají často jako je např. zpracování otáček, výpočty různých period atd.

### 3.2 KNIHOVNA FUNKCÍ

Tato knihovna obsahuje různé funkce používané nejenom pro řídicí algoritmy, ale také pro různé výpočtové modely fyzikálních systémů. Jako jsou různé tabulky, průměrování hodnot vektorů atd.

### 3.3 KNIHOVNA MULTIBODY

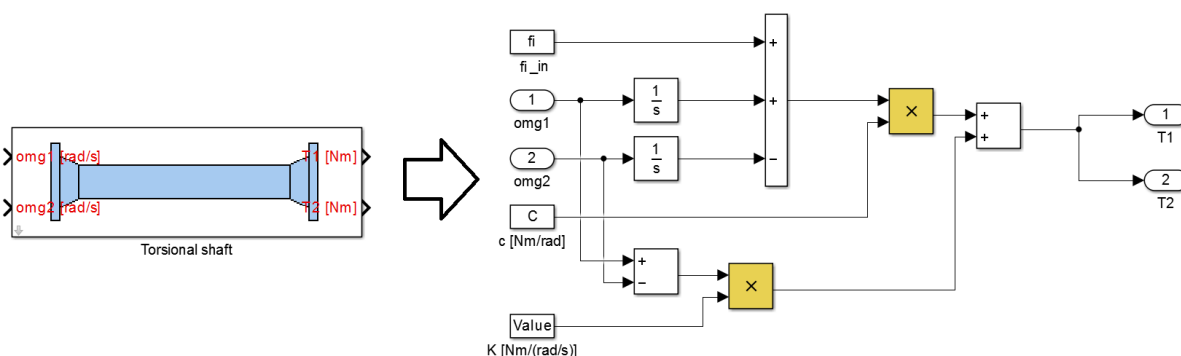
Knihovna multibody je ve fázi vývoje, proto zde nebude podrobně popisována, ale je především zaměřena na části vozidel, jako jsou nápravy a jejich zavěšení nebo na chování karoserie vozidla.

### 3.4 KNIHOVNA SENSORŮ

V této knihovně jsou bloky popisující chování různých snímačů, aby bylo možné generovat obdobné signály jako v reálném provozu. To znamená např. generování sinusového signálu, který generuje i indukční snímač při snímání otáček kola, nebo různé potenciometry. Dále jsou zde bloky reprezentující zpracování signálů z těchto snímačů. To je důležité pro Model in the Loop – MIL testování vyvíjeného řídicího algoritmu.

### 3.5 TECHNICKÁ KNIHOVNA

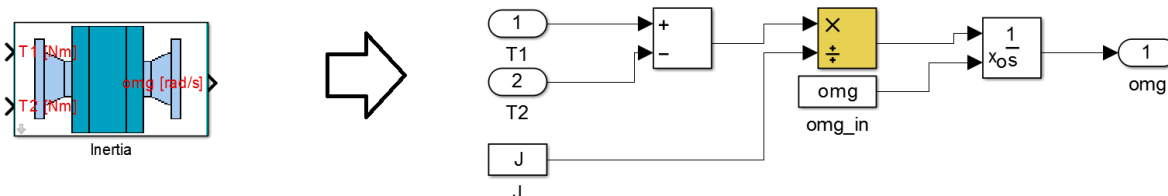
Obsahuje bloky pro popis mechanických částí jako je hřídele, rotační kotouč, základní převod, zubové spojky atd. Tyto bloky jsou základem torzních systémů a bloků pro sestavování hnacího traktu vozidel.



Obr. 14 Blok výpočtového modelu hřídele

Na předchozím obrázku je zobrazen blok hřídele definovaný její tuhostí  $c$ , vnitřním tlumením  $k$ , počátečním natočením  $\varphi_{in}$ , úhlovými rychlostmi bloku rotačního kotouče před  $\omega_1$  a za blokem  $\omega_2$  a výstupem je hnací moment pro blok rotačního kotouče připojeným před  $T_1$  a připojeným

za blokem hřídele  $T_2$ . Ten je tedy naopak vstupem pro blok rotačního kotouče zobrazeném na následujícím obrázku. Jeho nastavitelnými parametry je počáteční úhlová rychlost  $\omega_m$  a moment setrvačnosti  $J$ . Kombinací těchto základních bloků je uživatel této knihovny schopen sestavit jakýkoli rozvětvený torzní systém a testovat jeho dynamiku.



Obr. 15 Blok výpočtového modelu rotačního kotouče

### 3.6 KNIHOVNA PRO VOZIDLA

Knihovna pro vozidla se dělí na další podknihovny, kterými jsou:

#### 3.6.1 NÁPRAVY A PŘEVODY

Zde jsou vytvořené bloky reprezentující různé typy náprav. To umožňuje uživateli vybrat mezi předními, zadními nápravami, nápravami s diferenciálem, s pružnými prvky náprav s brzdami nebo i s reduktory. Dále jsou zde zahrnuty bloky diferenciálů a to s možností simulace jejich uzavření nebo s jejich pružnými prvky. Pro simulaci převodů jsou zde zahrnuty jednoduché bloky reprezentující jednoduchý zubový převod a pro celé převodovky různých výrobců jako je Tatra nebo ZF.

#### 3.6.2 BRZDY

Tato knihovna matematicky popisuje bloky pro bubnové a kotoučové brzdy. Zároveň jsou zde také zahrnuty bloky reprezentující pneumatické nebo hydraulické okruhy brzdných systémů. Kdy uživatel jako v předchozích případech může nastavovat jejich parametry, kterými jsou rozměry, statické a kinematické tření, rozměry hydraulických a pneumatických částí brzdného systému atd.

#### 3.6.3 VOZIDLO

V této knihovně jsou bloky pro různé typy vozidel např. nákladní nebo terénní čtyřkolky. Ale také jsou děleny na počty nápravy vozidla a jejich pohon. Tyto bloky řeší podélnou a příčnou dynamiku vozidla. Uživatel nastavuje parametry vozidla jako jeho hmotnost, rozměry, polohy těžiště, parametry pro výpočet jízdních odporů a momenty setrvačnosti. Výstupem jsou rychlosti, zrychlení, síly a poloha vozidla. Tyto výstupy jsou následně vstupem pro výpočtové modely pneumatik. Vstupem pro tyto bloky jsou naopak zase síly a vratné momenty získané z bloků pneumatik.

#### 3.6.4 TŘECÍ SPOJKY A ZUBOVÉ SPOJKY

Matematický popis třecích spojek vychází ze dvou přístupů tj. podle Karnoppa [57] nebo vlastním přístupem. Uživatel nastavuje průměry třecího obložení, koeficient kinematického a statického tření, průměry pístu, na který působí medium v okruhu, počet třecích ploch atd. Bloky zubových spojek mohou simulovat spojení dvou hřídelí, to znamená např. aktivaci pohonu všech kol. Uživatel může nastavovat momenty setrvačnosti, počáteční otáčky, tuhost,

tlumení atd. V těchto modelech jsou implementovány i bloky spínače zpětné vazby signalizující stav spojení s nastavitelným zpožděním.

### 3.6.5 MOTORY

Bloky motorů umožňují simulovat chování reálného motoru a dle potřebné složitosti a věrohodnosti je vytvořeno několik verzí, které umožňují simulovat hnací moment z vnější otáčkové charakteristiky, nebo uživatel může zadat naměřené indikované tlaky ve válci a tím simulovat torzní buzení hnacího traktu. Dále musí být zadány parametry momentů setrvačnosti, rozměrů klikového mechanismu a u propracovanějších modelů i parametry z hlediska termodynamiky atd. Do této knihovny patří i blok elektromotoru, dynamometru, pístní skupiny, bloky pro letecké motory a jejich funkce.

### 3.6.6 VOZOVKA

Blok zastupující vozovku popisuje adhezní podmínky mezi pneumatikou a daným povrchem. To znamená, že uživatel si vytvoří matici popisující součinitel tření mezi pneumatikou a vozovkou v závislosti na poloze vozidla a to konkrétně jeho kol. Tedy vstupem do bloku je poloha kola vozidla a dle ní je vyhodnocen součinitel tření, který se v dané oblasti nachází. Druhou možností je i nastavení matice s hodnotou valivého odporu pneumatiky. Tím může uživatel simulovat různé jízdní situace jako je jízda na ledu, sněhu, terénem, ale také především přechod mezi těmito povrchy. Výstupem z bloku je tedy součinitel tření a valivého odporu, který je zároveň vstupem pro bloky pneumatik.

### 3.6.7 PNEUMATIKY

V knihovně jsou připraveny bloky matematicky popisující chování pneumatik dle zadaných parametrů. Rovnice pro vytvoření bloku pneumatiky byly čerpány z literatury [58] a [66], kde autor Pacejka popisuje chování pneumatiky z hlediska sil a momentů pro všechny směry. A z těchto rovnic vyplývají také nastavitelné parametry uživatelem. Vstupem do bloků dle volby zda je užit model pro kombinované směry nebo jen pro podélný směr, jsou hodnoty rychlosti pohybu středu kola, hnací moment, normálová síla atd. Z hlediska výstupů jsou otáčky kola, síly a momenty působící mezi pneumatikou a vozovkou, skluz a úhel směrové úchylky. Tyto výstupy jsou vstupem pro blok vozidla a části hnacího traktu. Uživatel může také volit mezi matematickými metodami výpočtu skluzu a tím souvisejícím výpočtem sil, také musí zadat moment setrvačnosti kola, zda se jedná o pravé nebo levé kolo a kolik je vedle sebe umístěno pneumatik. To je pro případ nákladních vozidel, kde na zadních nápravách je tzv. dvoumontáž kol.

Kombinací zmíněných bloků je schopen uživatel sestavit jim požadovaný typ vozidla a po nastavení parametrů může simulovat a testovat jízdní situace. V následující kapitole bude popsáno využití sestavených výpočtových modelů vozidel pro testování vyvíjeného autonomního mechatronického systému pro uzavírání diferenciálů pro MIL testování.

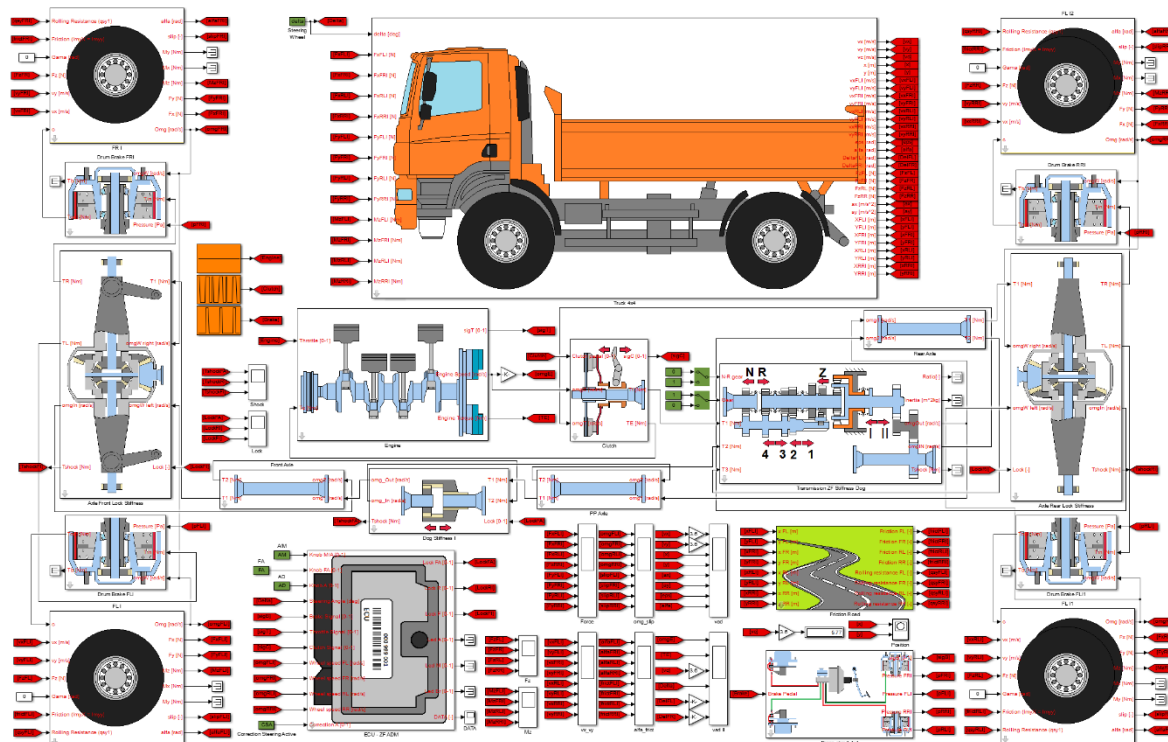
## 4 MODEL IN THE LOOP - MIL TESTOVÁNÍ

Model in the Loop testování je založeno na spojení výpočtového modelů např. vozidla a vyvíjeného řídicího algoritmu. Toto spojení je realizováno v uzavřené smyčce, kdy jsou z výpočtového modelu vozidla předávány informace řídicímu algoritmu a naopak řídicí algoritmus předává informace výpočtovým modelům akčních členů na modelu vozidla. To má zásadní výhodu, že není potřeba pro prvotní testování vyvíjeného řídicího algoritmu žádný prototyp vozidla, ECU, nebo snímačů a akčních členů. Další velice významnou výhodou je, že v této fázi vývoje je obecně nalezeno nejvíce možných chyb např. způsobených programováním nebo filozofií fungování celého systému. Také nemůže dojít k destrukci částí vozidla a mohou být testovány různé atypické jízdní situace, aniž by byl ohrožen řidič nebo obsluha. Všechny tyto výhody vedou k jednomu cíli, tj. ušetřit čas a peníze při vývoji a proto je tento způsob testování při vývoji mechatronických systémů významně užíván.

Dříve popsaný vyvíjený řídicí algoritmus byl touto formou také testován. Pro tento účel byly sestaveny výpočtové modely vozidel pomocí vlastní knihovny s bloky popsané v předchozí kapitole. Jedná se o tři typy vozidel s pohonem 4x4, 6x6 a 8x8. K těmto modelům byl připojen blok obsahující vyvíjený řídicí algoritmus a tím to vytvořilo smyčku pro MIL testování.

### 4.1 VOZIDLO 4x4

Výpočtový model vozidla s pohonem 4x4 je zobrazen na následujícím obr. 16. Tento model je uzpůsoben k simulaci jízdy vozidla při různých adhezních podmínkách a právě simulaci prokluzu jednotlivých kol.



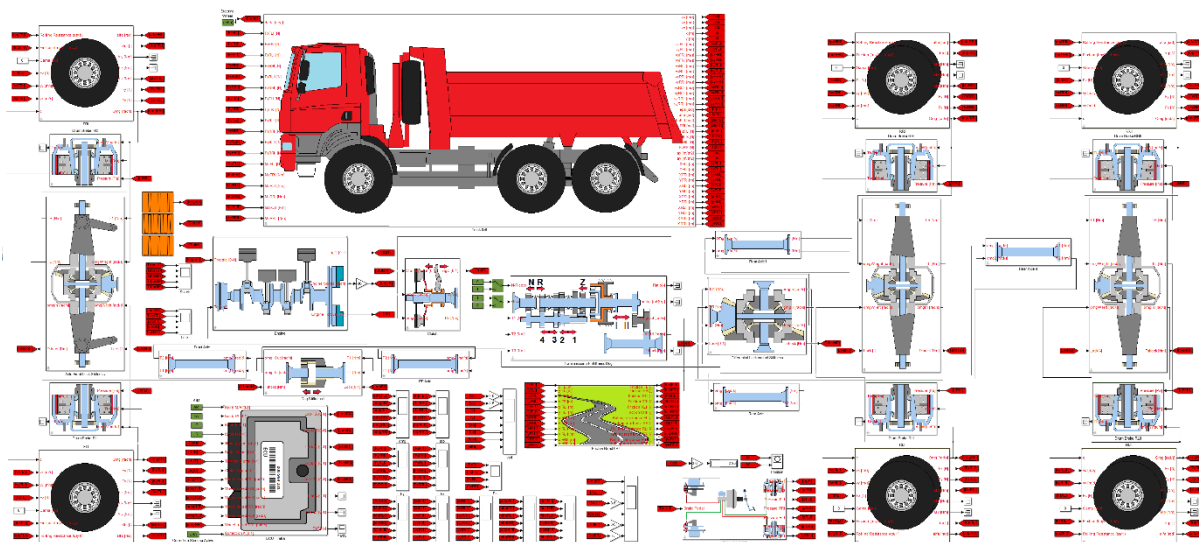
Obr. 16 Výpočtový model vozidla s pohonem 4x4 pro MIL testování

Tento výpočtový model je sestaven z bloku motoru (Paccar MX Euro 5) o výkonu 265 kW při otáčkách 1500-1900 min<sup>-1</sup> a hnacím momentu 1775 Nm při otáčkách 1000-1410 min<sup>-1</sup>.

K motoru je připojen blok spojky a šestnácti stupňové převodovky (ZF 16S 2530), součástí bloku převodovky je i výpočtový model sestupné převodovky. Dále jsou k výstupům z bloku převodovky připojeny bloky hnacích hřídelí pro přední a zadní nápravu. Před přední nápravou je umístěn blok zubové spojky, který slouží pro simulování spojení hnací hřídele přední nápravy a tím aktivaci pohonu této nápravy. K hřídelím jsou připojeny přední a zadní nápravy, které obsahují bloky s osovými diferenciály a pomocí vstupního signálu lze ovládat jejich uzavírání. Následují bloky s bubnovými brzdami a bloky s výpočtovými modely pneumatik (315/80 R22.5). Vše je ještě spojeno přes hlavní blok vozidla, kde je zadána i hmotnost vozidla 20000 kg. Důležité je také zmínit užití bloku vozovky, kde jsou definované adhezní podmínky v závislosti na poloze vozidla, konkrétně na poloze jednotlivých kol. Tímto způsobem jsou definovány podmínky pro testování a je např. testován rozjezd nebo průjezd bahnitým terémem. To znamená, že výpočtovým modelem vozidla je možné simulovat prokluz kol a aktivaci pohonu přední nápravy včetně uzavírání příslušných diferenciálů. Tomuto modelu jsou také definovány hodnoty řídicích veličin, to znamená signály plynového, spojkového a brzdového pedálu a dále natočení volantu. Ve spojení s šedě zbarveným blokem zastupující ECU, je vše připraveno na MIL testování.

## 4.2 VOZIDLO 6X6

Výpočtový model vozidla s pohonem 6x6 je zobrazen na následujícím obrázku. Jako předchozí model umožňuje simulaci jízdy vozidla a ve spojení s blokem obsahujícím řídicí algoritmus je připraven pro MIL testování. Jsou tu však některé bloky navíc, jelikož se jedná o vozidlo s pohonem 6x6, tak přibyl blok mezinápravového diferenciálu a druhá zadní náprava. Samozřejmě jsou zde zadány jiné vstupní parametry vozidla. To znamená, že např. výkon motoru je 320 kW při otáčkách 1450-1700 min<sup>-1</sup> a hnací moment 2100 Nm při otáčkách 1000-1450 min<sup>-1</sup>. Hmotnost vozidla je 29000 kg.



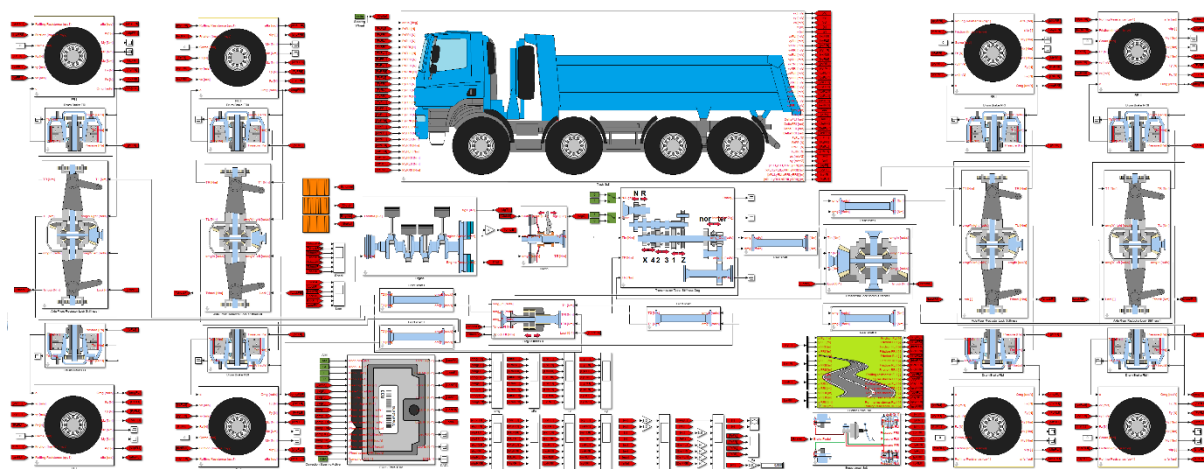
Obr. 17 Výpočtový model vozidla s pohonem 6x6 pro MIL testování

## 4.3 VOZIDLO 8X8

Výpočtový model vozidla s pohonem 8x8 je zobrazen na následujícím obrázku. Jako v obou předchozích případech je sestaven obdobně, avšak zde přibyly bloky pro druhou přední nápravu. Byl nahrazen blok zubové spojky pro aktivaci pohonu předních náprav za verzi pro aktivaci dvou předních náprav a byl nahrazen blok převodovky za jiný typ v tomto případě za



výpočtový model převodovky Tatra. Také byly změněny parametry motoru, které odpovídají motoru Tatra o výkonu 300 kW při otáčkách  $1800 \text{ min}^{-1}$  a hnacím momentu 2100 Nm při otáčkách  $1100 \text{ min}^{-1}$ . Hmotnost vozidla byla nastavena na 38000 kg.



Obr. 18 Výpočtový model vozidla s pohonem 8x8 pro MIL testování

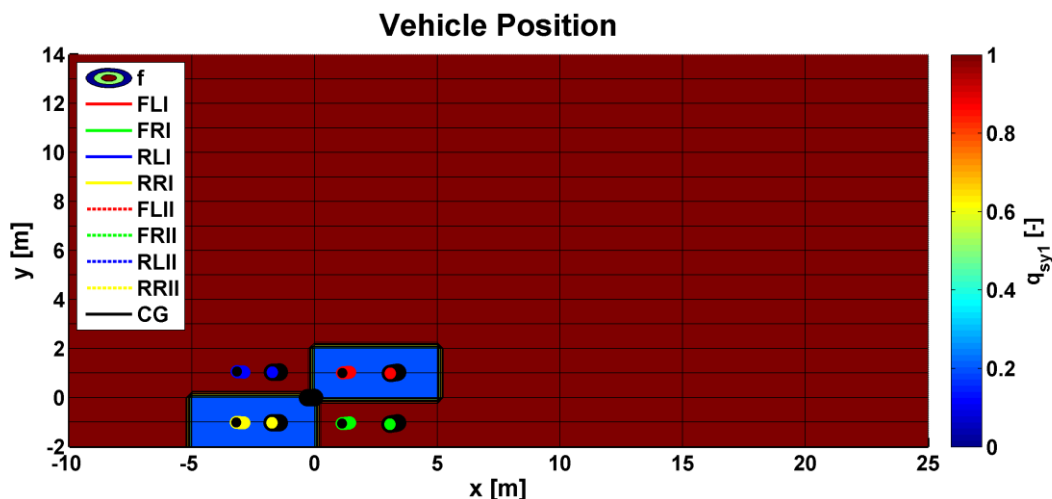
#### 4.4 SIMULACE

Po sestavení výpočtových modelů mohl být vyvíjený řídicí algoritmus testován, aniž by byl potřeba nějaký prototyp vozidla nebo ECU. Tím mohl probíhat paralelní vývoj a byl ušetřen čas. Proto byly připraveny různé simulace pro MIL testování z hlediska jízdy vozidla. Testovalo se rozjíždění, zatáčení, brzdění, přerazování rychlostních stupňů a to samozřejmě při různých adhezních podmínkách, aby docházelo k prokluzu kol a tím řídicí algoritmus musel reagovat na signály z výpočtového modelu vozidla. Po vyhodnocení situace odesílal signály pro aktivaci akčních členů a tím se aktivoval pohon všech kol a uzavíraly se příslušné diferenciály. Kladným výsledkem bylo samozřejmě, když vozidlo projelo simulovaným terénem, aniž by tam uvízlo. V předchozích kapitolách bylo zmíněno, že byly vyvíjeny dva přístupy řídicího algoritmu, tj. částečně nezávislé uzavírání a závislé uzavírání. První přístup byl právě jen testován pomocí MIL a následně se přešlo na závislé uzavírání, které bylo dále testováno i na prototypech vozidel.

Obecně simulace byly nastaveny s fixním krokem. Pro výpočet byl použit řešič ode14x, který kombinuje Newtonovy a extrapoláčnické metody. Tento řešič byl doporučován pro užívané prvky ve výpočtových modelech. Samozřejmě byla nastavena délka simulace, které závisela na jízdním manévru a během MIL testování byla zaznamenávána data. Z těchto dat byly vykresleny vždy potřebné grafy a probíhala vizuální kontrola, zda řídicí algoritmus reagoval na vstupní signály dle požadavků. V případě, že se našla chyba nebo algoritmus reagoval nevhodným způsobem, byla provedena změna kódu a MIL testování se spustilo znovu. Tento proces nelze moc zautomatizovat, proto byla neustále kontrolována výstupní data, dokud nebyly ošetřeny různé jízdní stavy a situace včetně simulování poruch snímačů nebo mechanických částí.

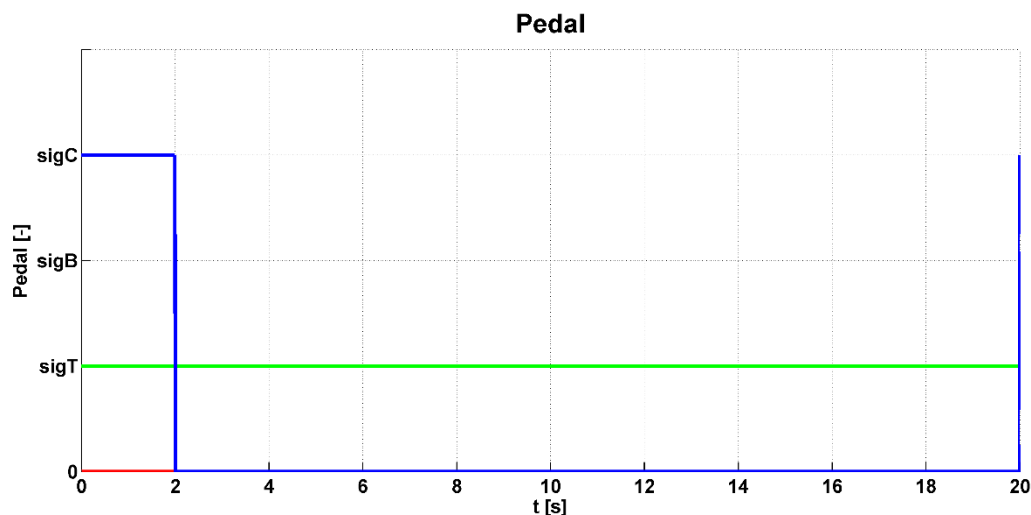
Ukázka jednoho MIL testování je zobrazena na následujících obrázcích, kdy další výsledky byly publikovány v různých časopisech [60], [61] [62] a [62]. Na prvním obrázku z následujících osmy je zobrazen graf reprezentující polohu vozidla a jeho kol. Zároveň je plocha grafu v určitých místech podbarvena různými barvami. Tyto hodnoty zastupují měřítko

součinitele přilnavosti mezi pneumatikou a vozovkou. To znamená, že vozidlo s pohonem 8x8 bylo v první fázi testování stojící pravými koly zadních náprav v bahnitěm terénu a levými koly přední nápravy také v bahnitěm terénu ale s jinými parametry. První testování je pro nastavení manuálního řízení, kdy nebyl manuálně aktivován pohon všech kol ani uzavřen žádný diferenciál.



Obr. 19 Poloha vozidla během MIL testování a při nastaveném manuálním řízení

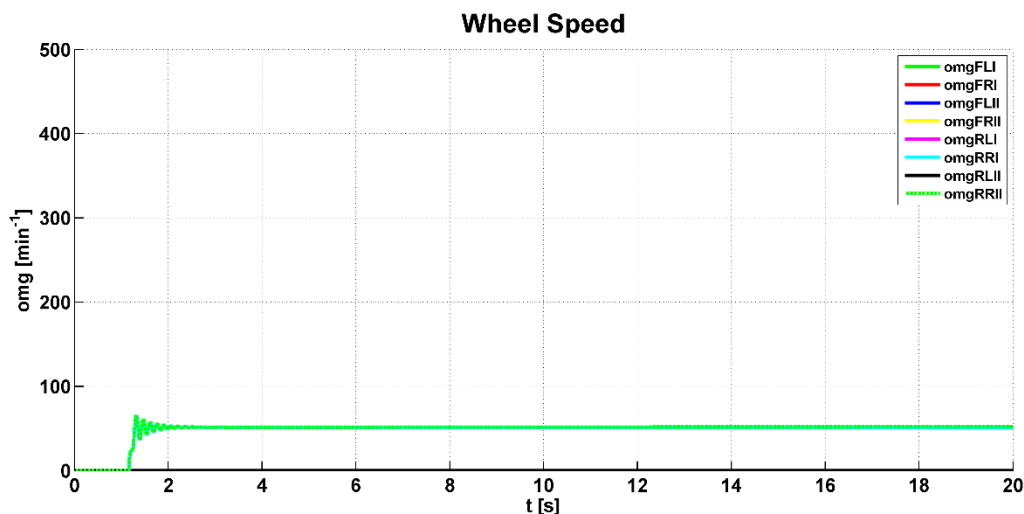
Vstupní signály pedálů ( $sigT$  – signál plynového pedálu,  $sigB$  – signál brzdového pedálu a  $sigC$  – signál spojkového pedálu) jsou zobrazeny na následujícím obrázku a z grafu je názorně vidět, že byl přidán plyn a zároveň uvolněn spojkový pedál. Tím se vozidlu začaly roztáčet kola. Nutno upozornit, že tento signál neprezentuje plynulé sešlapování nebo pouštění pedálů. Tohle je již digitální signál pro ECU o hodnotách 0 – volný pedál, 1 – sešlápnutý pedál.



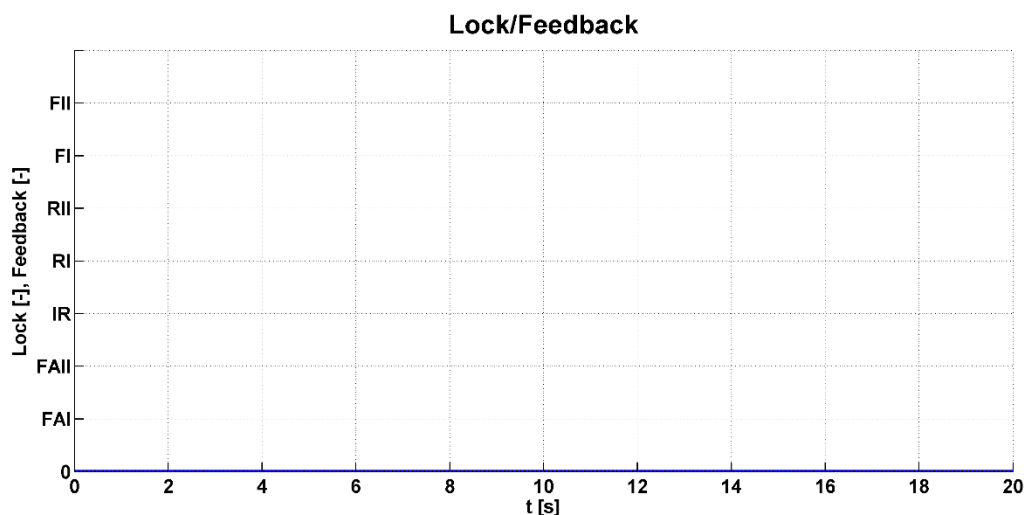
Obr. 20 Časový průběh signálů pedálů pro řízení jízdy výpočtového modelu vozidla při MIL testování a nastaveném manuálním řízení

Jelikož některá kola byla v bahnitěm terénu a ostatní na asfaltu, došlo k prokluzu pravých zadních kol, jak je vidět na následujícím obrázku. Ostatní kola mají hodnotu otáček nulovou včetně předních kol, jelikož v první fázi jsou poháněny u výpočtového modelu vozidla jenom

zadní nápravy. Tím vozidlo zůstalo stát na místě. Aby řidič z tohoto bahnitého terénu vyjel, musel by přepnout přepínač pro aktivaci pohonu všech kol.



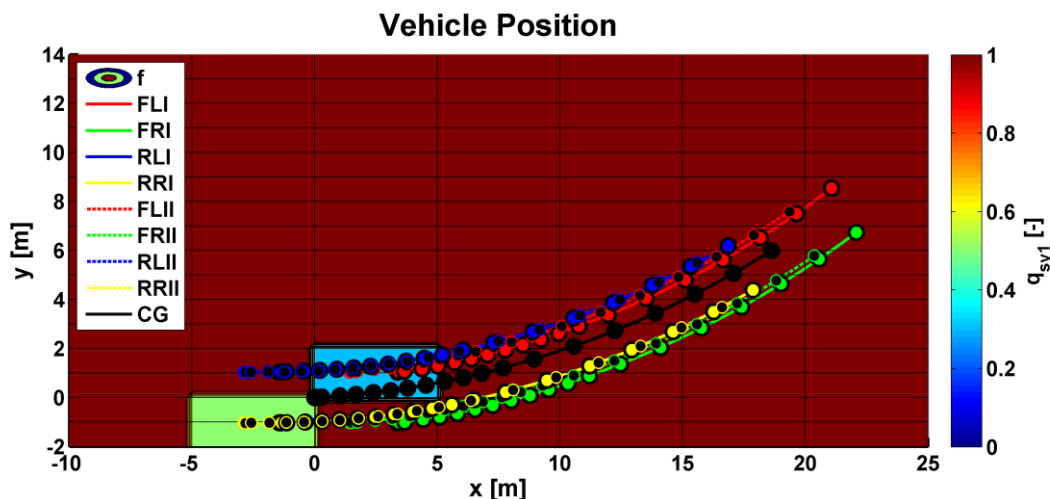
Obr. 21 Časový průběh signálů otáček jednotlivých kol pro rozjezd v bahnitém terénu při MIL testování a nastaveném manuálním řízení



Obr. 22 Průběh signálů pro akční členy (černá barva) a zpětné signály koncových spínačů (modrá barva) při MIL testování a nastaveném manuálním řízení

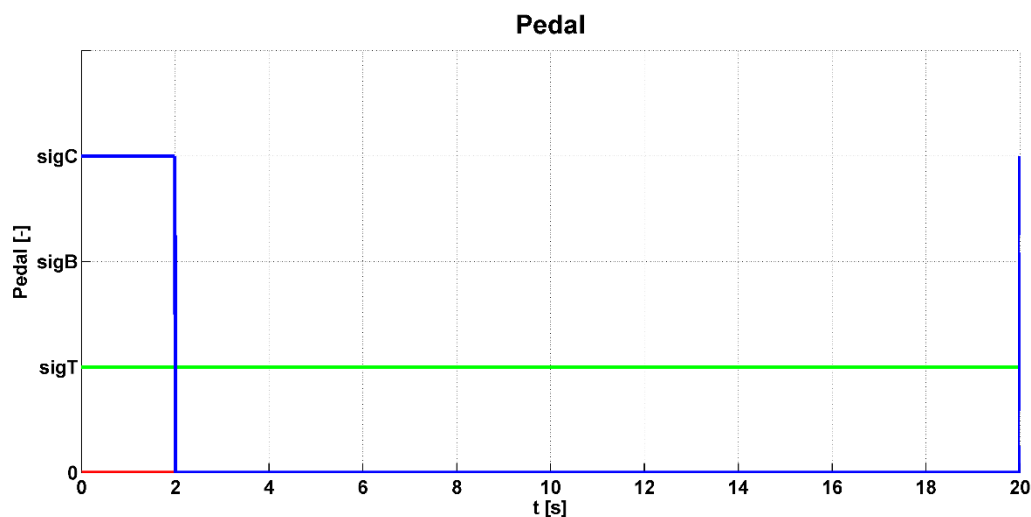
Pokud by to nebylo dostačující, tak by následně musel zmáčknout přepínač pro uzavření zadních osových diferenciálů a podržet tlačítko pro uzavření předních diferenciálů. Ale jak je vidět na výsledcích obr. 22, všechny signály  $Lock_i$  jsou nulové, tedy řidič nic neaktivoval. Simulace dál probíhala bez výsledku rozjetí vozidla. Zároveň bylo při simulaci nastavení zatočení volantu doleva.

Ukázka druhého MIL testování je na následující obrázku, kde již byl mechatronický systém nastaven na automatické řízení a všechna ostatní nastavení z hlediska signálů pedálů nebo zatočení volantu zůstala stejná.

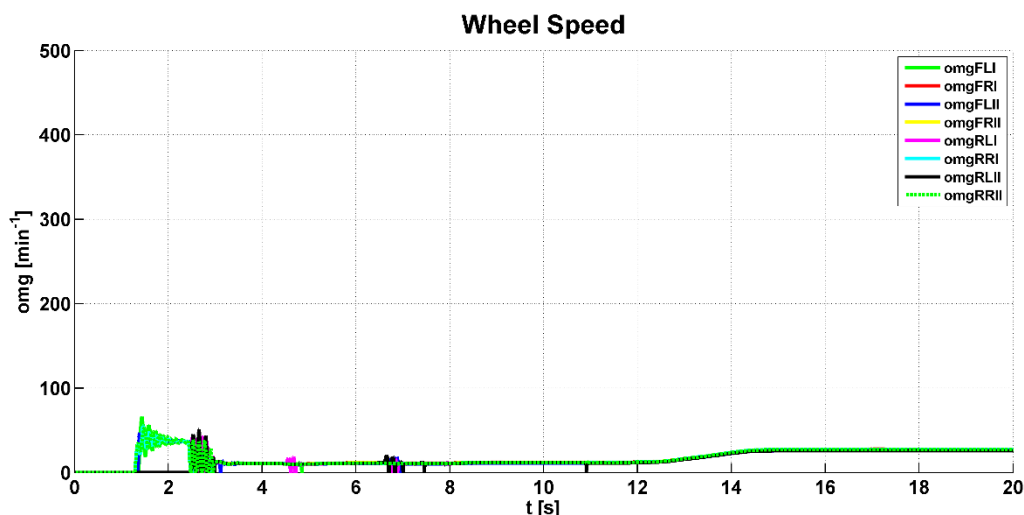


Obr. 23 Poloha vozidla během MIL testování s nastaveným automatickým řízením

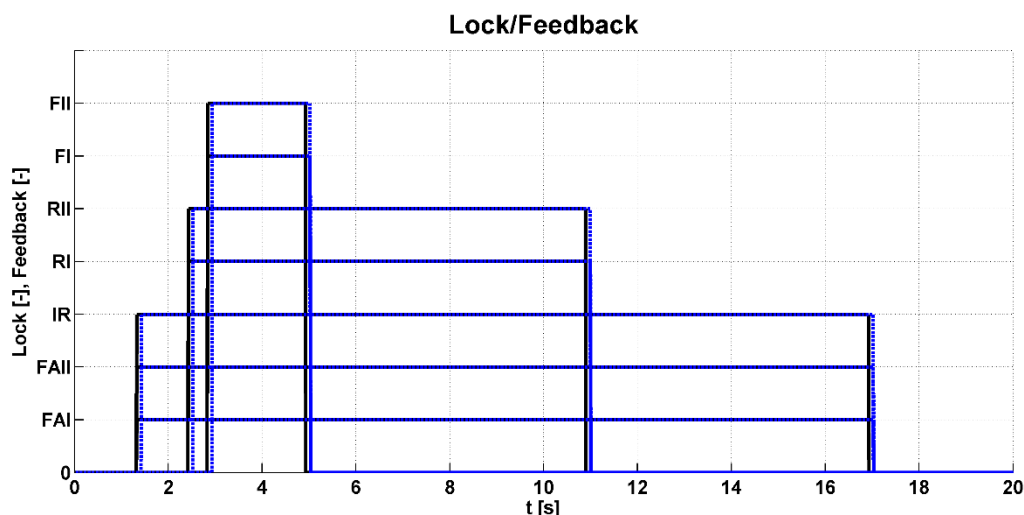
Předchozí a následující grafy polohy vozidla, vstupních signálů pedálů a signálů otáček kol ukazují, že algoritmus vyhodnotil situaci při rozjezdu vozidla a jeho jízdě terénem a postupně aktivoval pohon všech kol, nebo uzavíral příslušné diferenciály. Tím se vozidlo postupně dostalo z bahnitého terénu bez zásahu řidiče, dokud všechna kola nebyla na asfaltu. To je vidět i ze signálů *Lock<sub>i</sub>* na obr. 26, kdy po časovém úseku 17 s již řídicí algoritmus na situaci nemusí reagovat. Při tomto testování bylo např. ověřeno, že řídicí algoritmus reaguje na vstupní signály a uzavírá příslušné diferenciály. Naopak je kontrolováno, že vypne akční členy systému po skončení nepříznivých adhezních podmínek, ale také je kontrolována správná funkce korekce skluzu při zatáčení. Pokud by tato korekce nefungovala při MIL testování, algoritmus by zasílal signál akčním členům i v případě, když by celé vozidlo bylo na asfaltovém povrchu.



Obr. 24 Časový průběh signálů pedálů pro řízení jízdy výpočtového modelu vozidla při MIL testování a nastaveném automatické řízení



Obr. 25 Časový průběh signálů otáček jednotlivých kol pro rozjezd v bahnitým terénu při MIL testování a nastaveném automatickém řízení

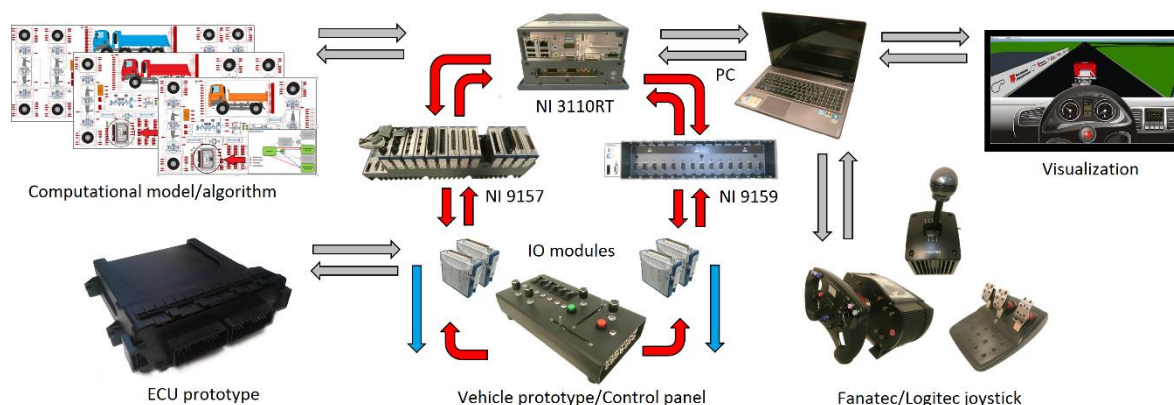


Obr. 26 Průběh signálů pro akční členy (černá barva) a zpětné signály koncových spínačů (modrá barva) při MIL testování a nastaveném automatickém řízení

V rámci vývoje probíhala řada MIL testování právě pro ověření různých situací a reakcí vyvíjeného autonomního systému. Proto z hlediska zmiňovaného rozsáhlého testování, byl v této kapitole znázorněn jen jeden příklad testování při rozjezdu vozidla z bahnitého terénu při zatáčení a manuálním nebo automatickém režimu řízení a při různých adhezních podmínkách. Jak již bylo zmíněno, všechny simulace byly kontrolovány a výstupem byl z velké části odladěný řídicí algoritmus, který byl následně použit pro Hardware in the Loop – HIL testování na prototypu vozidla nebo prvně testován v laboratorních podmínkách ve spojení se snímači.

## 5 HARDWARE IN THE LOOP - HIL TESTOVÁNÍ

HIL testování je využíváno za účelem testování na prototypech vozidel nebo jiných zařízeních, aniž by bylo nutné mít již vyroben prototyp ECU. To nese značné výhody z hlediska ušetření vývojového času a především jeho finanční náročnosti. To znamená, že hardware pro HIL je využit jako náhrada řídicí jednotky, kde je implementován vyvíjený řídicí algoritmus. Ten je již odladěn z MIL testování. Nyní je testovací smyčka již složena z hardwaru, elektroniky vozidla, akčních členů, snímačů a mechanických částí. V podstatě je sestaven prototyp vozidla, který se při testování doladuje k nulté výrobní sérii. Prototyp vozidla má zakomponovány všechny prvky mechatronického systému krom prototypu ECU, kdy náhradou je právě hardware pro HIL testování. Opět další výhodou je, že při vývoji řídicího algoritmu není potřeba čekat na vývoj prvního prototypu ECU, ale vývoj může probíhat paralelně. Také v případě, že by se jednalo o nevhodnou cestu vývoje, nemusí být zbytečně utraceny vysoké náklady na vývoj ECU. Dalším důvodem je otestování elektronických a mechanických prvků systému a především další ověření řídicího algoritmu vyvíjeného mechatronického systému. Jak bylo zmíněno nejvíce možných chyb, které vzniknou při vývoji, je nalezeno při MIL testování a při HIL testování je předpokládán také určitý rozsah možných chyb, ale již v mnohem menším měřítku. Potom hledání chyb systému při testování na kompletním prototypu vozidla je již zanedbatelné. Při vývoji autonomního mechatronického systému pro uzavírání diferenciálů byly použity výše zmíněné postupy a výhody a dle schématu na následujícím obrázku byl nebo bude využit hardware pro HIL testování i pro další vývoj.



Obr. 27 Schéma využívaného propojení HIL s dalšími periferiemi pro vývoj mechatronických systémů

Takže z hlediska hardwaru bylo použito zařízení od společnosti National Instruments. To bylo spojeno s prototypem vozidla určeného na testování mechatronického systému pro uzavírání diferenciálů. Další možností, která je využívána, je vývoj Real-time simulátoru vozidla více popsaném v [47]. Tento simulátor využívá sestavené výpočetové modely vozidel pomocí vlastní knihovny a následně je tento model implementován do procesoru hardwaru. Vozidlo lze řídit v reálném čase pomocí herního volantu a pedálu a uživatel vidí pohyb vozidla včetně palubní desky v naprogramovaném grafickém rozhraní. Z hlediska vývoje mechatronických systému je výhodou tohoto simulátoru, že je využíván pro testování vyvíjeného prototypu ECU s reálnými zásahy řidiče. To má zase nesporné výhody z hlediska rozsahu testování, bez nutnosti výroby prototypu vozidla. Vozidlu se mohou nadefinovat i testovací jízdní manévry, aniž by musel být použit reálný řidič. Tím lze testovat chování a správnost návrhu prototypu ECU. Tento simulátor je ve vývoji, tedy testování zde vyvíjené ECU je zatím omezené, proto zde nebude ani podrobněji popsáno. Proto testování ECU probíhalo především na vozidle.

## 5.1 HARDWARE

Jak již bylo zmíněno pro HIL testování je používán hardware od společnosti National Instruments a z jejich dokumentace na internetových stránkách [67] byly čerpány základní informace. Základem je hardware NI 3110RT, kde je umístěn procesor pro implementaci vyvíjeného řídicího algoritmu, který byl v tomto případě z C/C++ jazyka nebo softwaru Simulink kompilován do příslušného formátu dll knihovny. K tomuto hlavnímu hardwaru jsou připojeny dva sloty pro umístění modulů se vstupy a výstupy. Jedná se o sloty s označením NI 9159 a NI 9157. Pro danou aplikaci byl použit první slot s programovatelným hradlovým polem Field Programmable Gate Array – FPGA. V tomto hradlovém poli je naprogramování komunikace jednotlivých modulů s hlavním hardwarem, ale také to uživatel může doplnit vlastním kódem. To je využíváno pro zpracování rychlých dějů a v případě testování vyvíjeného systému, zde bylo naprogramováno zpracování otáček vozidla obdobnému jako při zpracování procesorem popsané v dřívějších kapitolách.



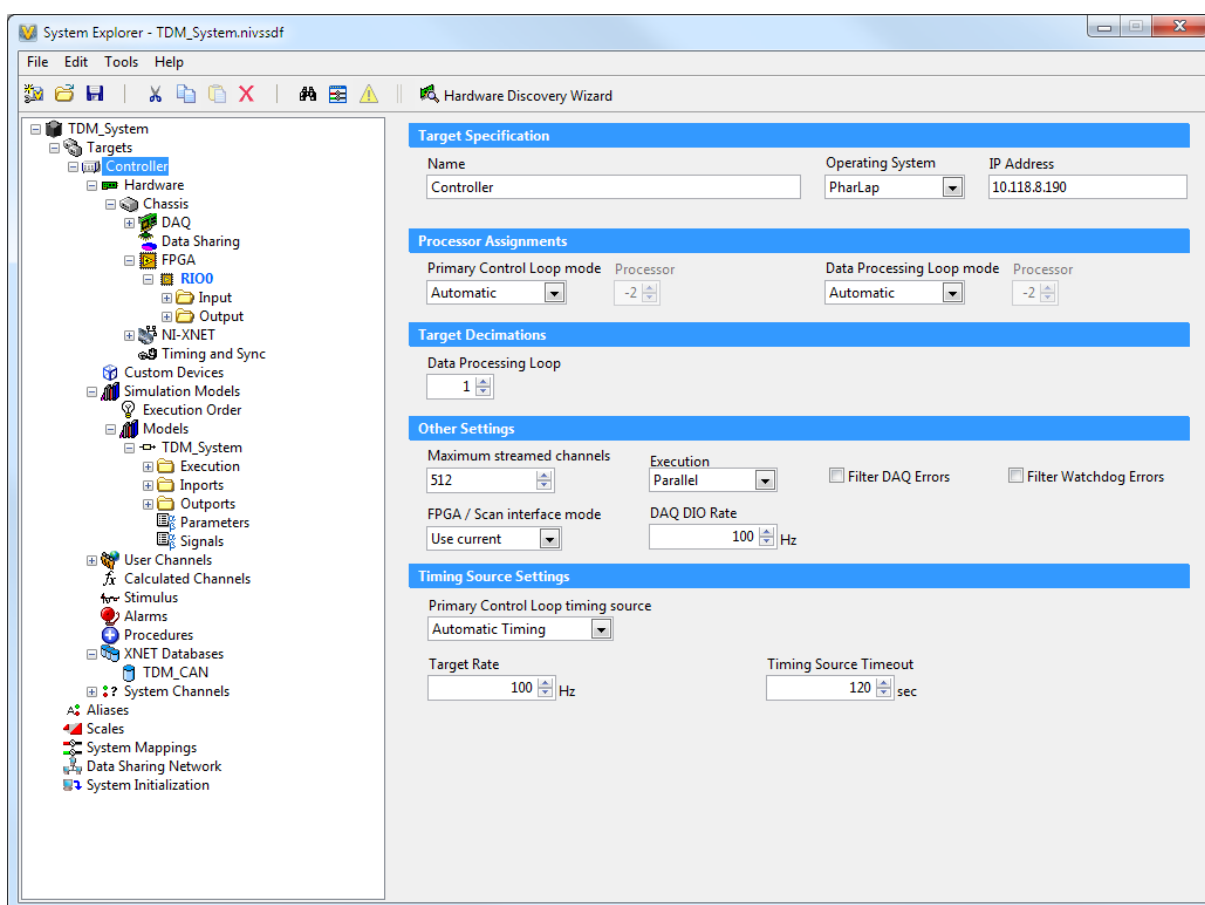
*Obr. 28 Implementovaný hardware v prototypu vozidla pro HIL testování*

Z hlediska modulů je vybavenost laboratoře v rozsahu vždy několika modulů typu NI 9229 – analogové vstupy s 24 bitovým převodníkem a napěťovým rozsahem  $\pm 60$  V, NI 9239 – analogové vstupy s 24 bitovým převodníkem a napěťovým rozsahem  $\pm 10$  V, NI 9264 – analogové výstupy s 16 bitovým převodníkem a napěťovým rozsahem  $\pm 10$  V, WF 3144 – simuluje PT100 sensory s rozsahem odporu 40–180  $\Omega$ , NI 9425 – digitální vstupy vhodné pro napěťové hladiny 12, 24 V a změna stavu s hysterezí nad 10 V hodnota 1 pod 5 V hodnota 0, NI 9477 – digitální výstup pro spínání uzemnění připojeného zařízení, NI 9472 – digitální

výstup pro spínání napájení připojeného zařízení, NI 9206 - analogové vstupy s 16 bitovým převodníkem a napěťovým rozsahem  $\pm 10$  V, NI 9263 analogové výstupy s 16 bitovým převodníkem a napěťovým rozsahem  $\pm 10$  V a NI 9403 – digitální vstup/výstup pro logiku transistor-transistor-logic - TTL. Pro HIL testování vyvíjeného systému, byly použity následující moduly. Dva moduly NI 9229 pro snímání signálů z indukčních snímačů otáček kol, jeden modul NI 9239 pro snímání analogového signálu ze snímače tlaku v pneumatickém okruhu a dvou signálů ze snímače natočení volantu, jeden modul NI 9425 pro snímání polohy pedálů pokud nebyly čteny hodnoty z CAN zpráv a pro získání hodnot ze spínačů zpětných vazeb v pneumatických válcích a jeden modul NI 9472 pro spínání akčních členů systému. Poslední využívanou kartou byla PCI 8513/2, která je určena pro komunikaci CAN a je konstruována jako zásuvná karta do hlavního hardwaru NI 3110RT. Pomocí takto sestaveného hardwaru, který je zobrazen na předchozím obrázku bylo prováděno HIL testování na prototypu vozidla.

## 5.2 SOFTWARE

Používaným softwarem pro komunikaci s hardwarem je tzv. NI VeriStand, jak je popsáno v [68]. Součástí jsou různá podokna, kde uživatel nastavuje různá specifika. Jedná se tedy o rozhraní System Explorer, zobrazeném na následujícím obrázku, kde se umístil zkompilovaný vyvíjený řídicí algoritmus ve formě modelu a byly nastaveny jeho parametry. To je především vzorkovací frekvence, se kterou má být spouštěn v procesoru.



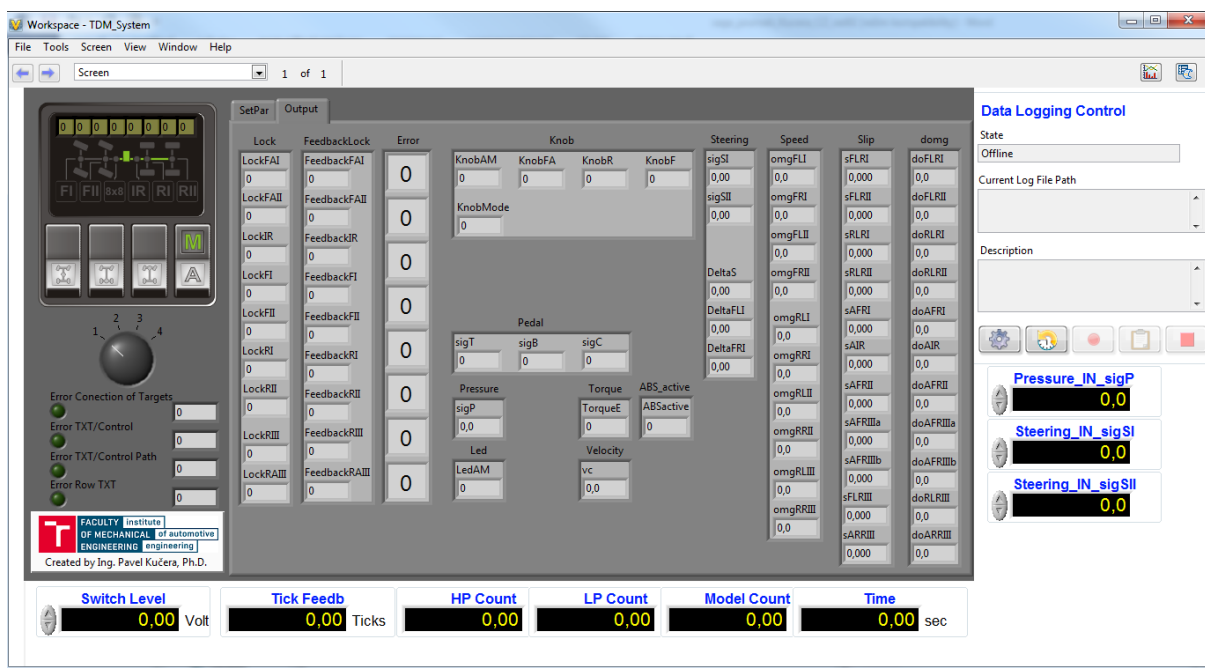
Obr. 29 System Explorer – je okno programu NI VeriStand, kde jsou nastavovány základní věci pro HIL testování



Dále je zde implementace programu pro FPGA, který slouží pro komunikaci s moduly a zároveň tam byly doprogramovány rychlé děje zpracování signálů z otáček kol. Dalším nastavením jsou pospojovány vstupy a výstupy řídicího algoritmu s příslušnými kanály modulů a s výstupy otáček také přes FPGA. Důležité je také vytvoření databáze pro CAN komunikaci, kde se nastavují i konkrétní signály a jejich přepočítání z přenášených dat CAN zprávou. Bylo také nastaveno vlastní komunikační rozhraní CAN. V tomto softwarovém rozhraní lze vše spustit i v prostředí operačního systému Windows, ale to jen za předpokladu, že k tomu není potřeba hardware. To je spíše jen pro otestování modelu (řídicího algoritmu). Pokud je to potřeba zprovoznit s hardwarem, je nutné nastavit jeho IP adresu. Potom při spuštění se model umístí na procesor hardware a FPGA kód do FPGA slotu. Následně NI VeriStand slouží jako zobrazovací interface nebo rozhraní pro nastavování parametrů řídicího algoritmu. Také je zde možnost umístění různých kontrolních prvků, jako jsou pro tento systém přepínače nastavení režimu mechatronického systému (manuální, automatické řízení) atd. To vše je vytvářeno v tzv. Workspace, které bude popsáno v následující kapitole. Samozřejmě software umožňuje řadu dalších funkcí, ale zde byly popsány ty základní, které byly použity pro HIL testování vyvíjeného autonomního systému na prototypu vozidla.

### 5.3 INTERFACE

Grafické rozhraní pro zobrazování aktuálních hodnot při HIL testování může být přes tzv. Workspace v softwaru NI VeriStand. Uživatel může využít předpřipravených tzv. Workspace Controls, kde jsou definovány grafy, zobrazování hodnot, nastavování hodnot, tlačítka atd. Pokud je daná knihovna s prvky nedostačující může si uživatel navrhnout vlastní grafické prvky. To je i případ vývoje systému pro uzavírání diferenciálu. Takové rozhraní je programováno v prostředí softwaru NI LabVIEW a je zobrazeno na následujícím obrázku. Pro základní programování v softwaru NI LabVIEW bylo vycházeno z knihy [69].



Obr. 30 Workspace okno softwaru NI VeriStand, kde jsou umístěny vlastní naprogramované grafické prvky pro vizualizaci a kontrolu hodnot řídicího algoritmu

Ve vytvořeném grafickém rozhraní pro tento vyvíjený systém jsou k dispozici kontrolní tlačítka a spínače (nastavení automatického nebo manuálního řízení a manuální aktivace akčních členů), otočný spínač (nastavení módů řídicího algoritmu – silnice, polní cesta terén/sníh), záložka s nastavováním vnitřních parametrů řídicího algoritmu a záložka s vnitřními proměnnými a výstupními hodnotami. V okně Workspace jsou užity i prvky z knihovny pro zobrazení např. zda se některé smyčky v hardwaru nezpomalují a to včetně implementovaného řídicího algoritmu. Posledním významným prvkem, který je využit z knihovny je tzv. Data Logging Control pro sbírání dat během HIL testování. Tato data byla následně využita pro vyhodnocení a kontrolu správné funkce vyvíjeného mechatronického systému.

## 5.4 PROTOTYPY VOZIDEL

Aby bylo možné provést HIL testování, bylo nutné vyrobit prototyp vozidla a to především mechanické části korespondující se systémem pro uzavírání diferenciálu. Jelikož tento systém byl vyvíjen i pro společnost Tatra Truck a.s., tak HIL testování probíhalo na jejich prototypových vozidlech s označením T815-7 a T158 Phoenix. Tyto prototypová vozidla jsou zobrazena na následujícím obrázku.



*Obr. 31 Prototypy vozidel Tatra s označením 815-7 (vpravo) a T158 Phoenix (vlevo)*

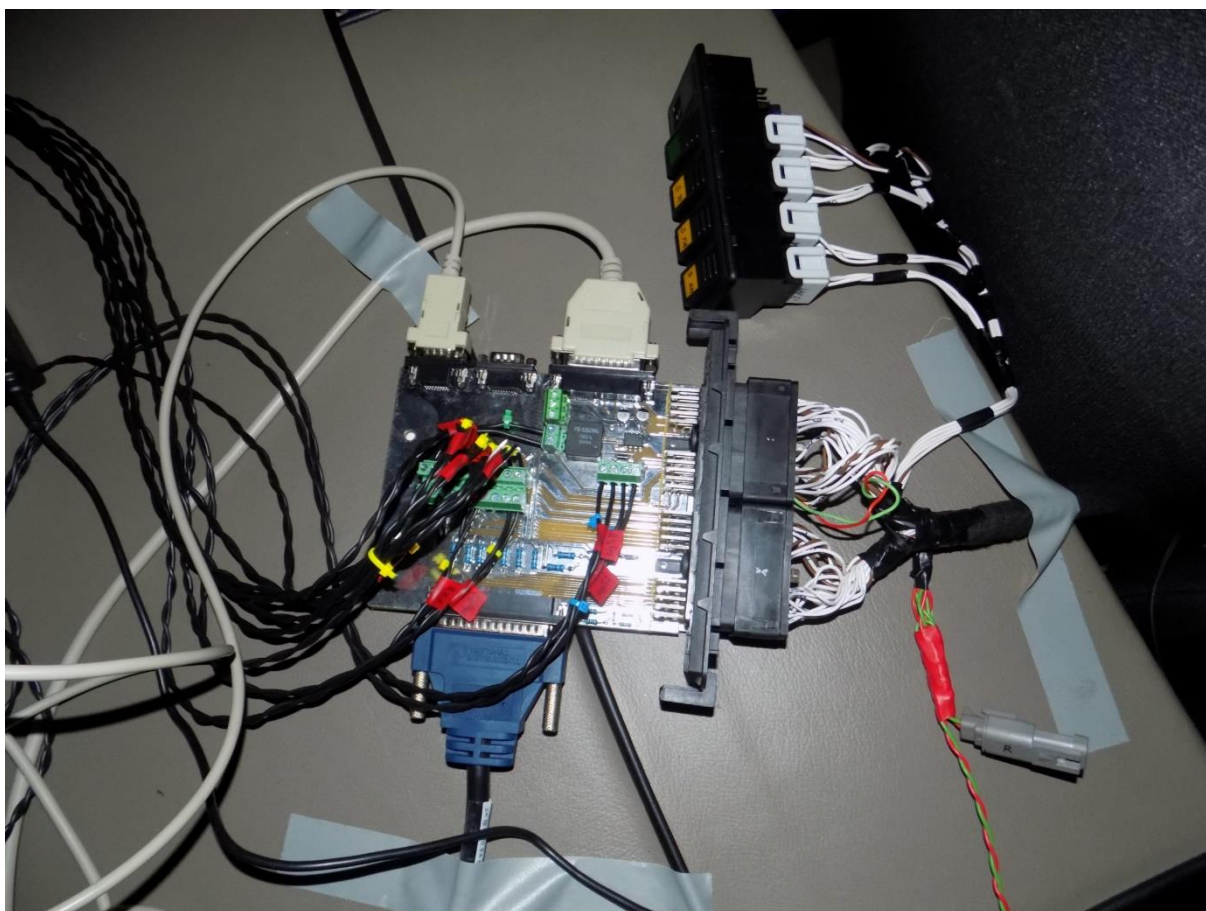
Vozidlo zobrazené vlevo je určeno pro stavební průmysl a vozidlo vpravo je určeného pro armádu nebo hasiče. Základem prototypů jsou sériová vozidla, kde došlo k úpravě a vývoji nových mechanických dílů hnacího traktu. Jednalo se o vývoj speciálních zubových spojek určených k rychlému uzavírání/otevírání diferenciálů za jízdy vozidla. S důrazem na odolnost vůči vzniku rázových momentů při jejich uzavírání a obecně koncipovaných pro přenos hnacího momentu. Tento vývoj zubových spojek a částí hnacího traktu proběhl ve společnosti Tatra Truck a.s. v rámci projektu TE 01020020, proto detaily návrhu zde nebudou popisovány. Obecně prototyp vozidla je sestaven z vyvinutých mechanických komponent tak, že zubové spojky jsou umístěny v každém diferenciálu a také pro spojení hnacích hřídel předních náprav. To znamená, že dané prototypy s pohonem 8x8 byly z hlediska hnacího traktu sestaveny ze 4 osových diferenciálů, jednoho mezinápravového diferenciálu pro zadní nápravy a ze dvou zubových spojek pro aktivaci pohonu jednotlivých předních náprav. Všechny tyto zubové spojky jsou ovládány konzolovou pneumatického válce. K němu je připojen elektroventil, který při otevření do něho vpustí tlakový vzduch z pneumatického okruhu a tím je vyvozena síla pro posuv pohyblivé části zubové spojky a dojde k jejímu spojení. K rozpojení zubové spojky dochází při uzavření elektroventilu a tím začne tlakový vzduch odcházet z pneumatického válce. To je způsobeno tím, že jeho píst je přetlačen vnitřní pružinou. Součástí pneumatického válce je koncový spínač signalizující reálné spojení zubových spojek, tedy signalizující uzavření

diferenciálu nebo aktivaci pohonu předních náprav. Tato sestava mechanických prvků, elektroventilu, pneumatického prvku a senzoru tvoří akční člen, který je řízen vyvíjeným mechatronickým systémem. Zobrazení pneumatického válce je na následujícím obrázku s označením B, kde se nachází i elektroventil. Ve specifické nosné rouře pro tyto vozidla se nachází zubová spojka pro aktivaci pohonu předních náprav nebo diferenciál, který má v sobě nějaký typ z vyvíjených zubových spojek. Ukázka jednoho z protikusů zubové spojky je zobrazen pod označením G.



*Obr. 32 Snímače a akční členy*

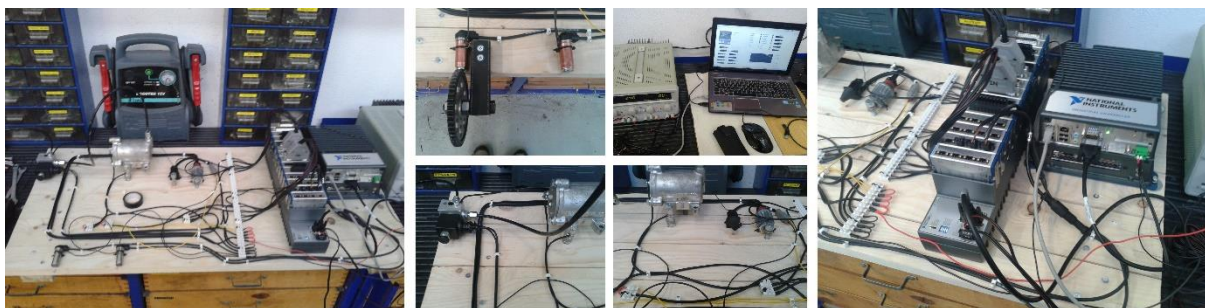
Dále byly prototypy osazeny snímači otáček (E, F), které běžně slouží pro systém ABS. V případě náprav, kde jsou umístěny snímače otáček pro ABS, byly tyto snímače zdvojeny, proto novější verze vyvíjeného systému již umožňuje přijímat CAN zprávy s informacemi o rychlostech kol z ABS. Pokud tedy systém ABS umožňuje nastavení periody této zprávy aspoň na 20 ms nebo kratší. Dále je vozidlo osazeno snímačem natočení volantu a tento snímač je umístěn na jedné z pák řízení. To je zobrazeno na obrázku pod označením C. Zásobník vzduchu pneumatického okruhu je doplněn o snímač tlaku na obrázku s označením D. Jelikož první prototyp vozidla 815-7 nedisponoval některými CAN zprávami dle normy J1939, tak musely být v kabině umístěny snímače pedálů zobrazené na obrázku s označením A pro spojkový a brzdový pedál a H pro plynový pedál. U novějšího prototypu vozidla nejsou tyto dodatečné snímače potřeba, jelikož jejich hodnoty potřebují i další jednotky a tím jsou tyto informace k dispozici v CAN zprávách. Tyto snímače a akční členy byly zapojeny dle jejich potřeby a signály byly svedeny do kabinového prostoru prototypu vozidla. Zde byly napojeny na konektory CINCH dvakrát po 30 pinech a protikusem je obal a konektor ECU s označením ModICE® LE Header – 5810160005. Pro HIL testování bylo vytvořeno speciální hardwarové rozhraní zobrazené na následujícím obrázku, které bylo s jednotlivými IO moduly HIL hardwaru spojeno pomocí příslušných kabelů a na straně snímačů a akčních členů pomocí konektoru pro prototyp ECU. Na obrázku je také vidět panel s přepínači a tlačítky pro nastavování automatického a manuálního režimu systému a jejich signály byly také připojeny do jednoho z konektorů od společnosti CINCH. Takto byla vozidla připravena pro HIL testování a následně taky pro testování kompletního vyvíjeného systému i s prototypem ECU.



*Obr. 33 Hardwarové rozhraní pro spojení hardwaru HIL se snímači a akčními členy přes konektor pro prototyp ECU*

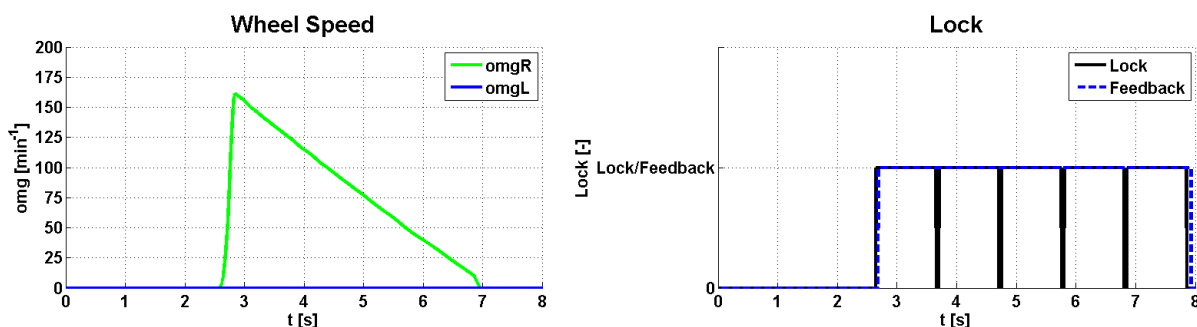
## 5.5 REAL-TIME TESTOVÁNÍ

Pro HIL testování byly připraveny dva prototypy vozidel. Ale než se začalo testovat na nich, proběhlo ještě testování v laboratorních podmínkách, které je zobrazeno na následujícím obrázku. Bylo to z toho důvodu, aby zbytečně nedošlo nějakým nedopatřením k destrukci hnacího traktu vozidla při zpracování reálných signálů ze snímačů, proto byl vytvořen testovací panel. Panel má na sobě rozmístěné snímače pedálu, snímače otáček kol, pneumatický válec s koncovým spínačem a elektroventil. Z hlediska pneumatiky vše bylo zapojeno do centrálního přívodu vzduchu laboratoře. Z hlediska počtu snímačů to odpovídá uzavírání diferenciálu jedné nápravy. Testování probíhalo tak, že před jedním snímačem otáček kola bylo umístěno ozubené kolo a to bylo roztáčeno. Druhý snímač otáček byl umístěn volně a jeho výstupem byl nulový napěťový signál. Tím byl simulován prokluz jednoho kola a sledováno správné zpracování signálu ze snímače otáček a reakce řídicího algoritmu v podobě odezvy z koncového spínače v pneumatickém válci. Výstupem tohoto testu bylo odladění správného zpracování signálu ze snímačů a řízení akčního členu.



Obr. 34 Laboratorní HIL testování v propojení se senzory a akčním členem

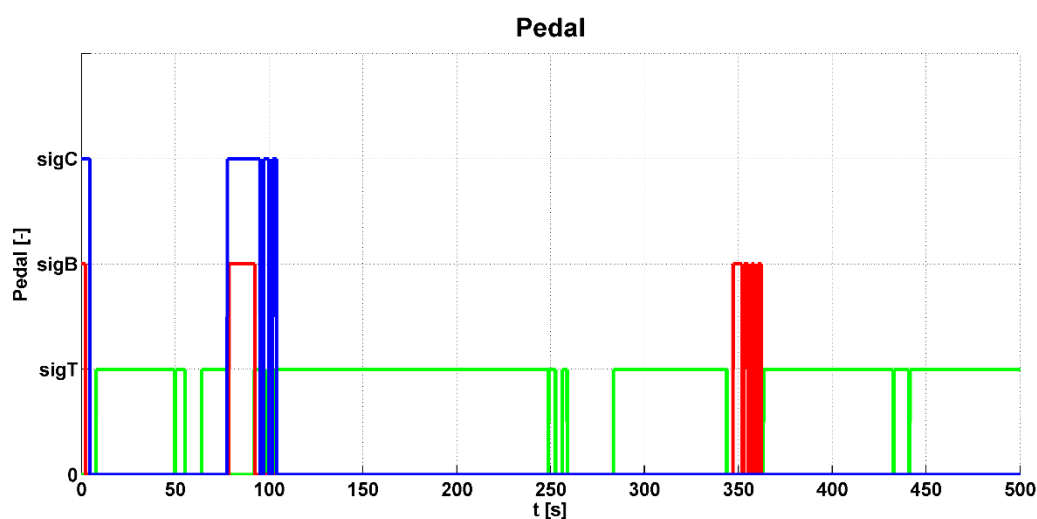
Zároveň byla otestována funkce testovací smyčky. Ukázka zpracovávaných dat je na následujícím obrázku a také podrobněji rozebrána v publikaci [71]. Na levém grafu je názorně vidět zelenou čáru reprezentující časový záznam zpracovaných otáček pravého kola a modrou čáru levého kola o hodnotě nula. V pravém grafu je vidět černý signál, který byl odeslán akčnímu členu a modrou čarou je označen signál z koncového spínače pneumatického válce, kdy došlo k posuvu pístu s konzolou v pneumatickém válci.



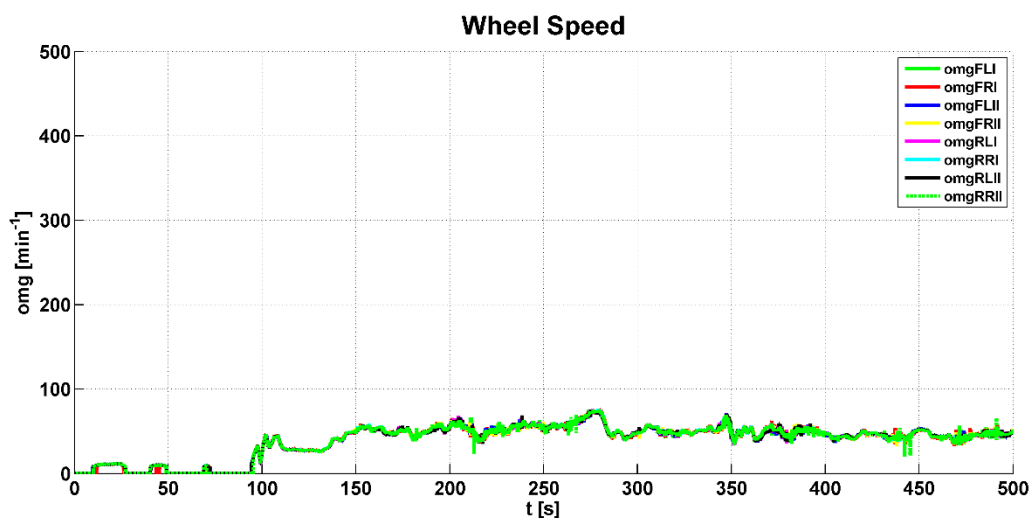
Obr. 35 Ukázka výsledků z laboratorního HIL testování v propojení se senzory a akčním členem

Na prototypu vozidla by tímto posuvem došlo k vyvození síly na konzole a ke spojení zubových spojek např. v osovém diferenciálu. Z hlediska testů byly testovány jízdní stavy na asfaltu, polních cestách a těžkém terénu. Testy obsahovaly různé způsoby jízdy, rozjezdů, zastavení, brzdění, zatáčení, testování poruch systému, testování na rovině nebo na speciálních vozovkách typu vlnovce a vozovek s definovaným sklonem. Dále byly testy prováděny v různých

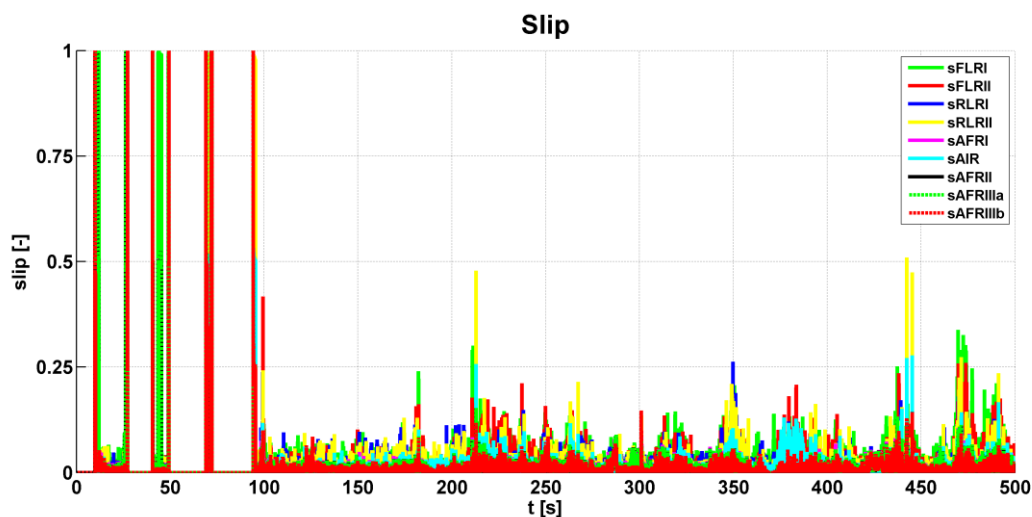
povětrnostních podmínkách od vysokých letních teplot přes deštivé počasí až po zasněženou a namrzlou vozovku. Součástí testování byly i jízdy po kropené vozovce vodou pro přímý směr nebo zatáčení, dále s některými koly na asfaltu a ostatními na výrazně kluzné vozovce atd. Z hlediska HIL testování bylo cílem odladit řídicí algoritmus a otestovat mechanické, pneumatické a elektronické součásti vyvíjeného mechatronického systému. Proto bylo snahou otestovat nejrůznější jízdní manévry s prokluzem kol a sledovat správnost fungování mechatronického systému. Jelikož zpracování získaných dat není možné automatizovat, proto byly vždy vytipovány ty testovací úseky, ve kterých se zdálo, že systém nefungoval úplně optimálně. Potom došlo okamžitě k úpravě např. kódu a následnému testování. Zároveň těmito testováními byly odladěny vnitřní řídicí parametry pro dané dva prototypy. Jelikož řídicí algoritmus a fungování celého systému bylo významně odladěno ve virtuálním prostředí pomocí MIL testování, nebylo nutné dělat zásadní zásahy z hlediska fungování daného systému. Proto se především ověřovaly všechny funkce, simulovaly poruchy snímačů např. odpojením jejich konektorů. Ukázka dat z jednoho testu v terénu, která byla kontrolována a vyhodnocována, je na následujících obrázcích.



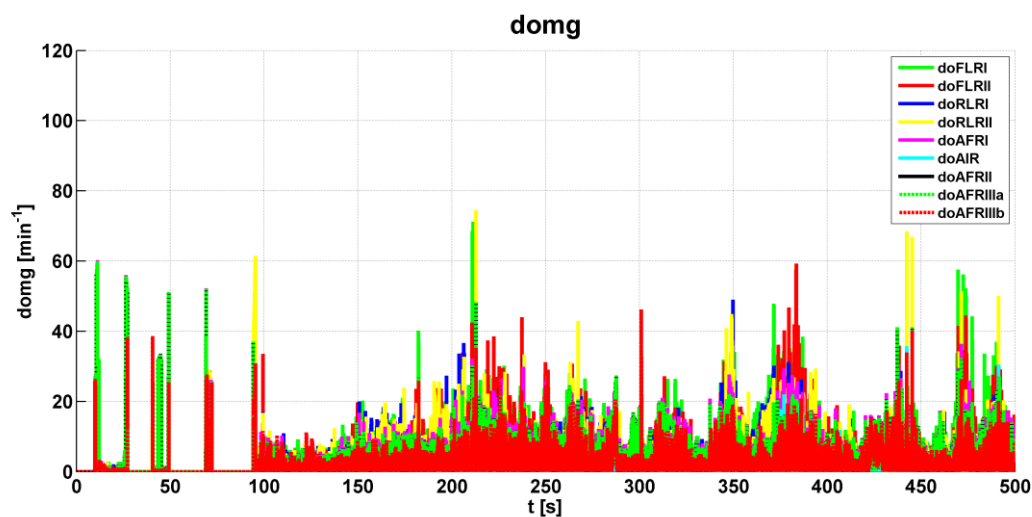
Obr. 36 Průběh signálů pedálů při HIL testování



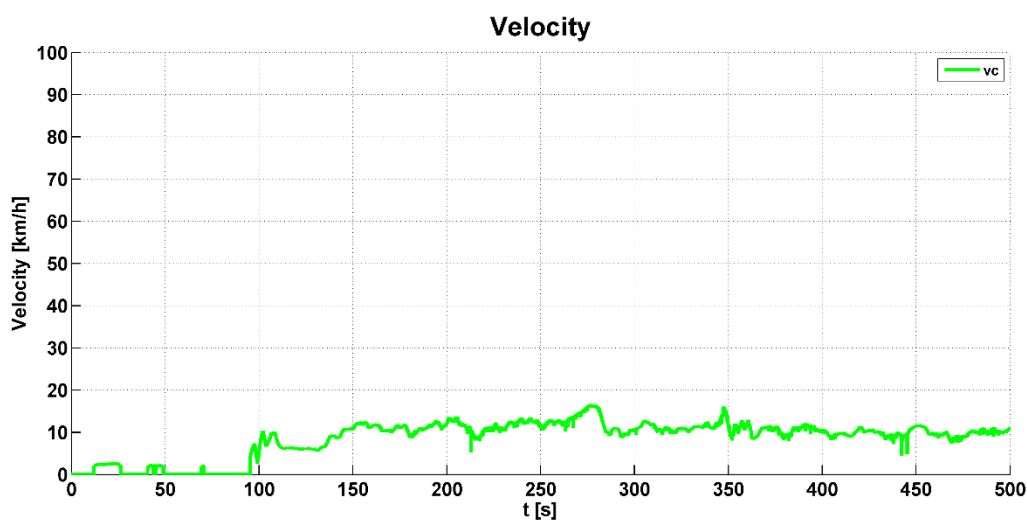
Obr. 37 Průběh signálů otáček jednotlivých kol při HIL testování



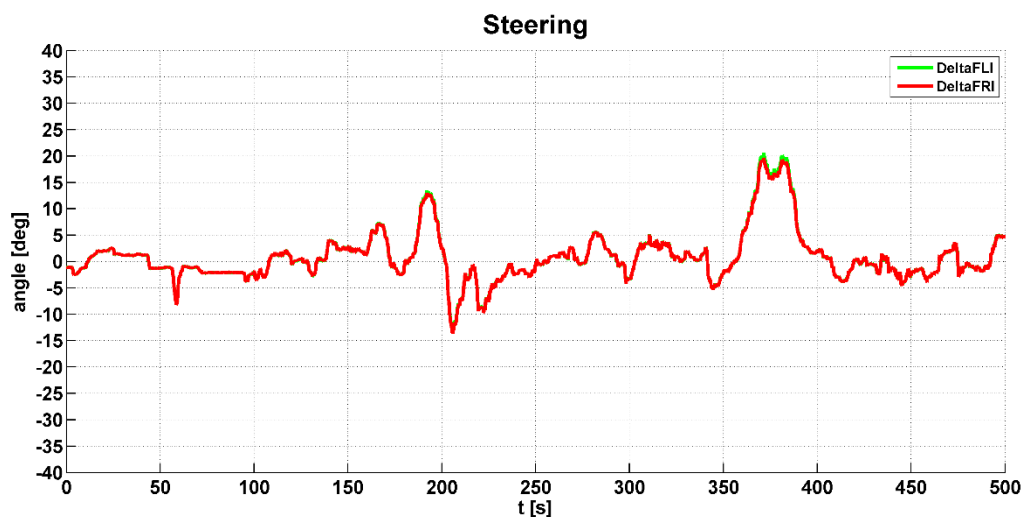
Obr. 38 Průběh signálů skluzu při HIL testování



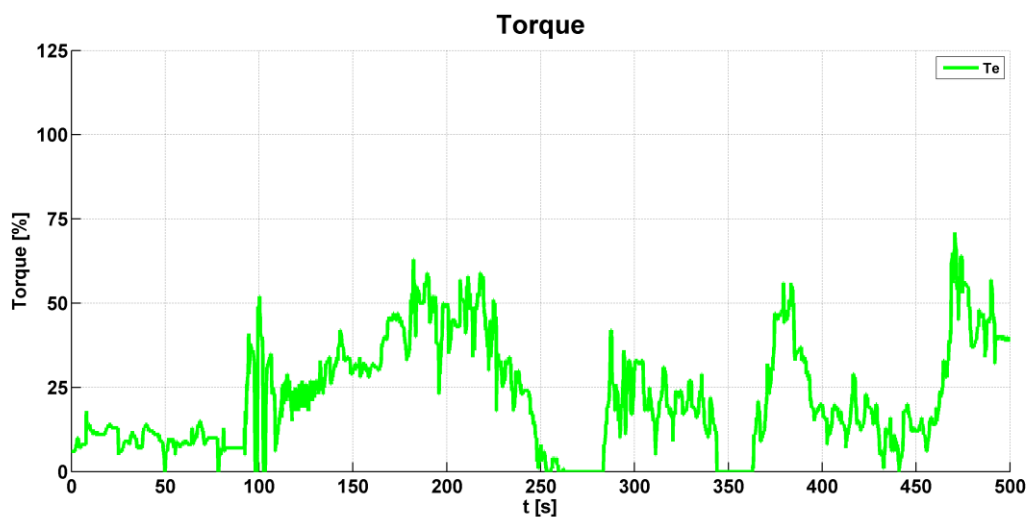
Obr. 39 Průběh signálů rozdílu otáček spojovaných hřídelů zubovými spojkami při HIL testování



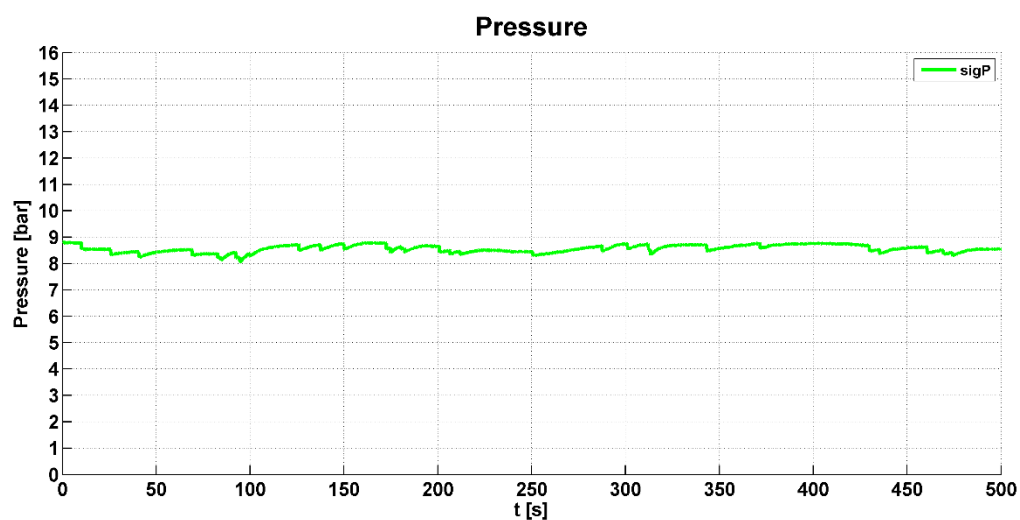
Obr. 40 Průběh signálu rychlosti vozidla při HIL testování



Obr. 41 Průběh signálů zatočení pravého a levého kola přední nápravy

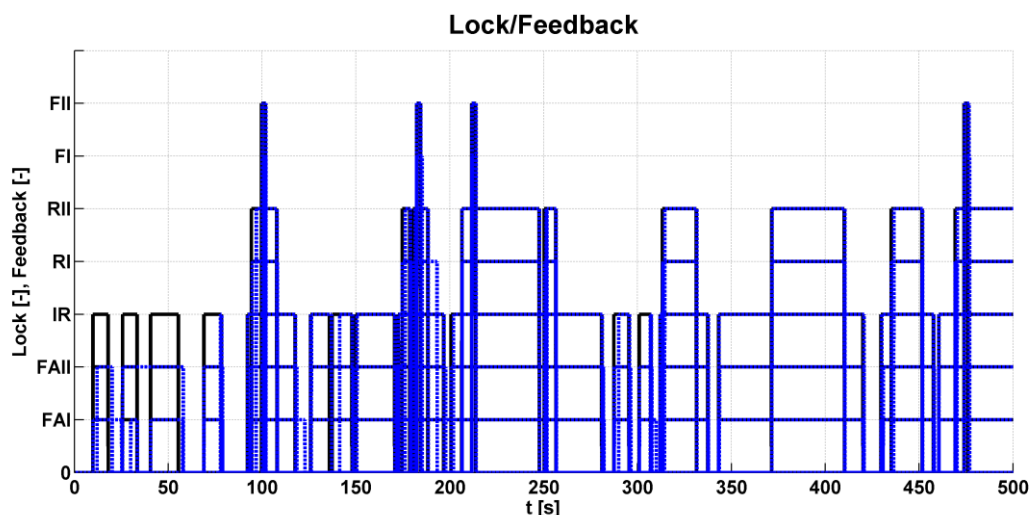


Obr. 42 Průběh signálu hnacího momentu motoru



Obr. 43 Průběh signálu tlaku v pneumatickém okruhu





Obr. 44 Průběh signálů pro akční členy (černá barva) a zpětné signály koncových spínačů (modrá barva)

Na obr. 36 jsou zobrazeny vstupní signály pro řídicí algoritmus od spojkového pedálu – *sigC*, plynového pedálu – *sigT* a brzdového pedálu – *sigB*. V grafu je vidět, že se prototyp vozidla několikrát rozjížděl, okolo času 100 s byly řazeny převodové stupně nebo několikrát brzdil v čase okolo 350 s. Na obr. 37 jsou zobrazeny a vyhodnoceny jednotlivé otáčky *omg*, které jsou předávány řídicímu algoritmu a ten z nich například vyhodnotí průběhy skluzu mezi jednotlivými koly a hřídelemi hnacího traktu. Skluz je zobrazen na obr. 38 a jedná se o hlavní veličinu, podle které řídicí algoritmus určuje jeho reakci. Na grafu je vidět dost často vyšší špičky, které vznikaly při skluzu kol, a tím algoritmus vyhodnocoval výstupní signál zasílaný akčním členům. Z hlediska užitých kombinací zkratk se jedná o *F* – přední náprava, *R* – zadní náprava, *L* – levé kolo, *R* – pravé kolo, *A* – náprava, *IR* – zadní mezinápravový diferenciál, *I* – pořadí nápravy. To znamená, že např. označení *sFLRI* je skluz mezi pravým a levým kolem první přední nápravy. To je užíváno ve všech grafech. Dále jsou z otáček vyhodnocovány rozdíly otáček mezi hřídelemi *do*, které jsou spojovány pomocí zubových spojek. Průběh těchto veličin je na obr. 39. Z otáček kol je také vyhodnocena rychlost vozidla *vc*, která je zobrazena na obr. 40. Na obr. 41 jsou zobrazeny průběhy zatočení pravého a levého kola přední nápravy *Delta*. Tento signál slouží pro výpočet korekce skluzu, která je již zahrnuta na obrázku s průběhem skluzu. Nebo také tento signál slouží pro kontrolu překročení limitů zatočení kol pro uzavírání osových diferenciálů. Průběh hnacího momentu motoru *Te* je zobrazen na obr. 42. A pokud tato veličina měla vyšší hodnotu, než byl stanovený limit pro příslušný mód, tak na to algoritmus příslušně reagoval. Na obr. 43 je zobrazen průběh tlaku *sigP* v pneumatickém okruhu. V tomto grafu bylo například kontrolováno, zda je pneumatický systém dostatečně navržen, aby při intenzivní aktivaci pohonu všech kol a uzavírání diferenciálů byl tlak dostatečný. Právě v grafu je vidět poklesy při aktivaci příslušného akčního členu. Hlavními výstupními daty jsou signály *Lock* pro řízení akčních členů (černá barva) a *Feedback* - zpětné vazby z koncových spínačů pneumatických válců (modrá barva), to je zobrazeno na obr. 44. Podle grafu je jasně vidět, že docházelo k intenzivním zásahům řídicího systému, jelikož se v tomto případě jednalo o náročný blátivý terén.

Obecně všechna data byla analyzována a byla kontrolována správná funkce řídicího algoritmu. Pokud něco nebylo v pořádku, došlo k úpravě. Důležitou poznámkou je, že při podrobné analýze ukázkových dat v této práci nemusí být reakce řídicího algoritmu stejná na podobné signály. To je způsobeno tím, že zde zobrazená data at' pro MIL testování, HIL testování a

finální testování jsou např. pro různé nastavené řídicí parametry nebo jinou nastavenou konfiguraci z hlediska zpracování vstupů atd.

Součástí HIL testování bylo také velice důležité vyhodnocení času reakce na uzavření/otevření diferenciálů nebo aktivace pohonu předních náprav. Cílem bylo odladit a navrhnout celý mechatronický systém tak, aby především průměrné časy pro spojení zubových spojek byly okolo 100 ms a nižší. Tato časová reakce je důležitá z hlediska funkčnosti celého systému, aby nedocházelo k uzavření diferenciálu již při velkých rozdílných otáčkách kol. Potom by mohla nastat destrukce částí hnacího traktu. Tento problém je ošetřen i v řídicím algoritmu tím, že pokud je rozdíl otáček kol nad určený limit, tak příslušný diferenciál nesmí být již uzavřen. Přes tato opatření je důležité, aby časy reakce byly co nejmenší, protože by při uzavírání zubových spojek stále mohlo docházet k rázům, které by nevedly k destrukci částí hnacího traktu, ale byly by nepohodlné pro řidiče.

V následující tabulce jsou zobrazeny časy reakcí z odladování systému. V čase je zahrnuta doba od odeslání signálu ECU pro akční člen až po změnu zpětného signálu z koncového spínače pneumatického válce. Tedy je tam zahrnuta reakce elektronických, mechanických a pneumatických prvků. Toto měření probíhalo na speciální vlnovkové vozovce a jednotlivé mezery mezi spojkami a nastavením polohy spínání koncového spínače v pneumatickém válci byly rozdílné. Samozřejmě probíhala i další testování na různých typech vozovek, ale z důvodu velkého rozsahu dat je zde zobrazena jen malá ukázka. Potom po následném vyhodnocení byly všechny zubové spojky a koncové spínače nastaveny podle toho nejlepšího času. Podrobnější popis byl publikován v časopise [72].

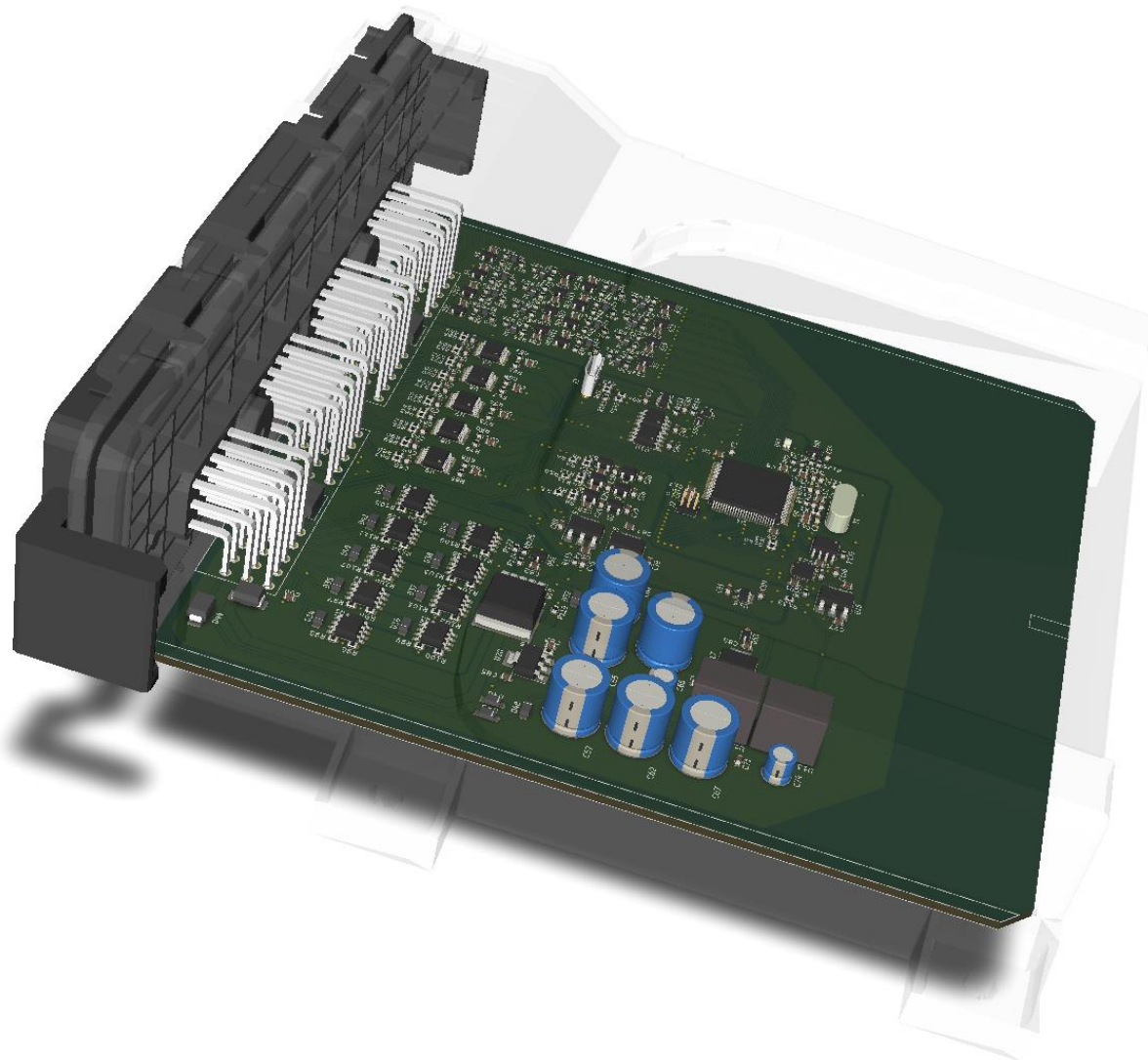
*Tab. 1 Vyhodnocení času reakce na uzavření/otevření diferenciálů nebo aktivace pohonu předních náprav*

	<b>FAI</b>	<b>FAII</b>	<b>IR</b>	<b>RI</b>	<b>RII</b>	<b>FI</b>	<b>FII</b>
Lock min. time [s]	0,01	0,01	0,02	0,01	0,01	0,07	0,07
Lock max. time [s]	0,08	1,24	1,99	1,24	0,38	2,82	1,88
Lock mean time [s]	0,07	0,19	0,22	0,14	0,09	0,77	0,35
Unlock min. time [s]	0,10	0,15	0,13	0,01	0,09	0,13	0,11
Unlock max. time [s]	1,71	4,10	3,40	1,49	2,62	2,79	2,74
Unlock mean time [s]	0,14	0,67	0,51	0,22	0,22	0,63	0,57

Přestože některé časy nebyly úplně shodné s daným požadavkem na reakci, je to stále výsledek velice uspokojující v dostatečné míře z hlediska vzniku možných momentových rázů v hnacím traktu. Jelikož v kabině vozidla téměř řidič ani posádka nepoznají, že právě docházelo k uzavírání diferenciálů. Takže pozitivním výsledkem HIL testování je další odladění mechatronického systému z hlediska funkce a rázových momentů. Nejenom tyto pozitivní výsledky byly publikovány v různých časopisech, ale také jednou z dalších publikací zahrnující HIL testování tohoto systému je [70].

## 6 VÝVOJ PROTOTYPU ECU

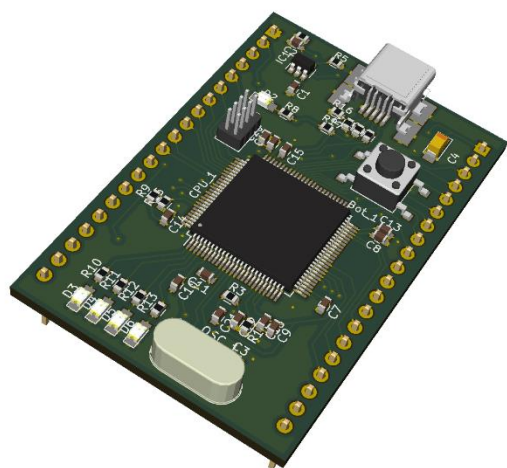
Vývoj ECU prototypu a jeho návrh byl sestavován pomocí softwaru KiCad [73]. Jedná se o otevřenou platformu softwaru pro návrh elektronických obvodů. Součástí je rozsáhlá knihovna elektronických součástek, které uživatel používá, jak v návrhu schématu elektrického obvodu, tak při návrhu tištěného spoje. Průběh návrhu probíhal tak, že v první fázi bylo vytvořeno schéma celého obvodu. To znamená od vstupního konektoru CINCH po vstupy a výstupy použitého procesoru. Dílčí části z hlediska zapojení budou popsány v následujících podkapitolách, obecně se jednalo o zapojení procesoru, digitálních vstupů, digitálních výstupů, CAN komunikace, napájení ECU, vstupy otáček, externí paměti, gyroskopu, analogových vstupů a USB rozhraní.



*Obr. 45 Virtuální prototyp ECU*

Po návrhu předběžného schématu, byly jednotlivým prvkům obvodu přiděleny pouzdra osazení. U tohoto návrhu bylo především užito tzv. surface mount device – SMD pouzder o velikosti 0603. Následně návrh prototypu ECU pokračoval ve formě návrhu desky plošných spojů – DPS. Kde bylo nutností brát v úvahu základní předpoklady návrhu, kterými byly výrobitelnost, přesnost, vedení vodičů dle elektrických požadavků, elektromagnetická

kompatibility – EMC atd. To vše vycházelo z literatury [74], [75], [76], [77], [78] a [79] nebo z konzultací s danými odborníky na problematiku vývoje ECU. Přesto při vývoji a hledání vhodného vedení vodičů musí občas dojít ke kompromisu, aby bylo vůbec možné vše umístit a spojit na tištěném spoji. Zde například muselo být vycházeno ze zapojení vstupního konektoru zákazníka. Při návrhu DPS se tedy paralelně upravovalo schéma elektrického obvodu, především z hlediska spojení pinů konektoru jednotky a pinů procesoru. Software umožňuje i zobrazení 3D tištěného spoje včetně součástek a po celkovém návrhu a vytvoření vlastních knihoven se součástkami, které nebyly k dispozici softwarovou knihovnou, je návrh zobrazen na obr. 45. Samozřejmě je důležité zmínit, že před samotnou výrobou prototypu ECU proběhla řada testování dílčích obvodů např. pomocí nepájivých kontaktních polí, nebo výrobou jednodušších tištěných spojů. Např. byl vytvořen vlastní mikrokontrolér pro testování periférií užitého procesoru, který je zobrazen na obr. 46. Pomocí něho nebo komerčního mikrokontroléru LPC1768 [80] byl vždy sestaven obvod s daným elektrickým obvodem např. pro testování digitálního vstupu a byly sledovány potřebné veličiny. Případně byla prováděna porovnání se simulací dílčích obvodů pomocí softwaru SIMetrix-SIMPLIS Elements [81].

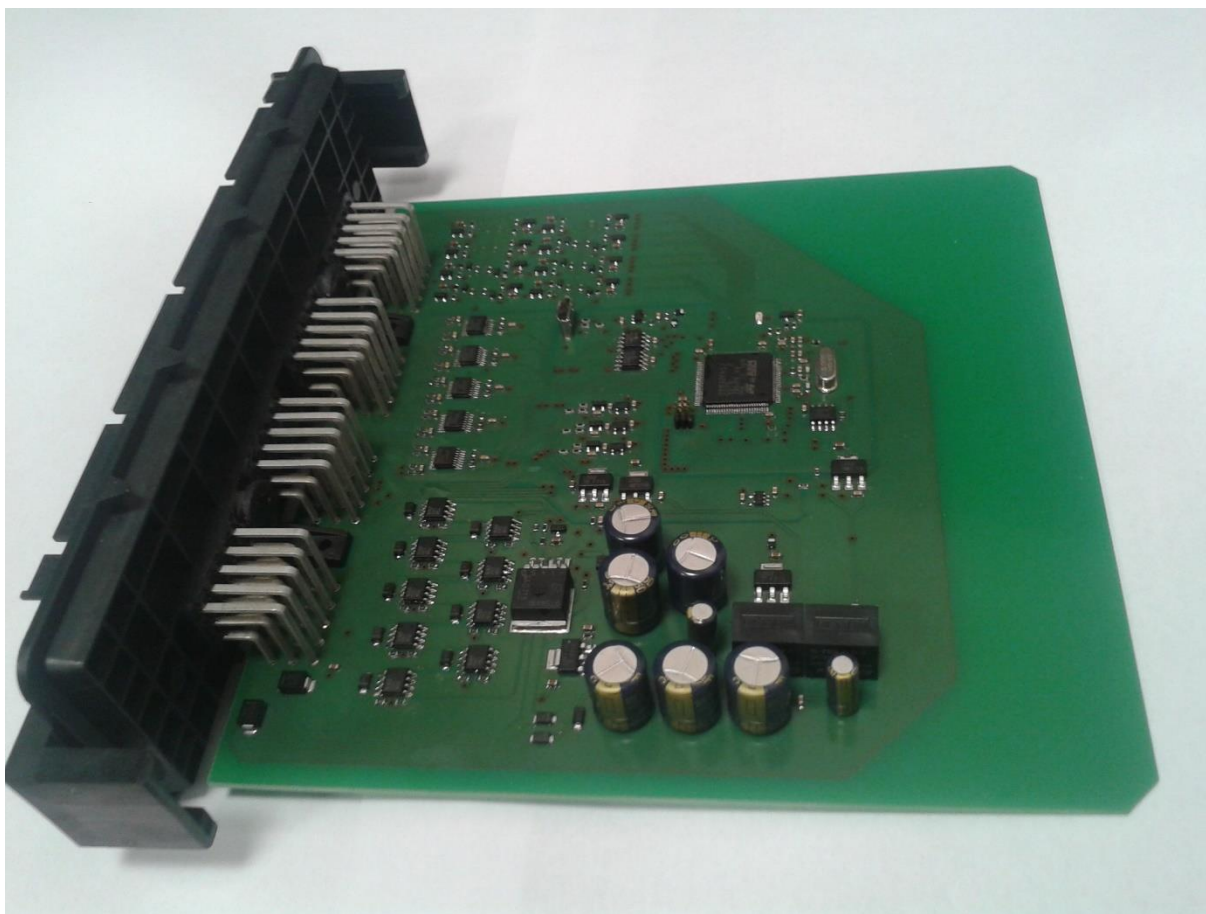


Obr. 46 Prototyp mikrokontroléru

Po otestování dílčích elektrických obvodů bylo možné vygenerovat výrobní data prototypu ECU. To znamená, že návrh tištěného spoje byl převeden do tzv. gerber formátu a souboru se souřadnicemi a velikostmi vrtaných a frézovaných otvorů. Tyto data byla zaslána do výroby pomocí tzv. POOL servisu. Jedná se o kusovou výrobu tištěných spojů. Osazování tištěného spoje bylo provedeno ve vlastních silách, ale samozřejmě pokud by prototyp přešel do sériové výroby, bylo by prováděno strojové osazování. Potom by k základní výrobní dokumentaci přibyl ještě osazovací výkres a data obsahující souřadnice jednotlivých elektronických součástek s jejich orientací. Proto byl připraven i plán výroby v rozsahu 50 – 100 kusů, kde by zakázka byla zadána specializující firmě na tuto výrobu včetně dodání elektronických součástek, osazení a základního otestování. Kalkulace pro výrobu byla v rozsahu pěti poptávaných firem cca v řádech tisíců korun. Samotná cena prototypu byla samozřejmě dražší, jelikož se jednalo o kusovou výrobu.

Během ručního osazování prototypu byl obvod testován, aby bylo ověřeno správné propojení jednotlivých elektronických součástek a tím se předešlo vzniku zkratů nebo chybějícího napájení atd. Pro testování vznikly dva prototypy ECU, kde druhý již byl rozšířen o další elektronické prvky MEMS pro systém PS. Následně byla jednotka testována v laboratorním

prostředí z hlediska správné funkce. Novější prototyp ECU je zobrazen na obr. 47, kde jsou vidět i dílčí části elektrického obvodu. Ty budou popsány v následujících kapitolách.



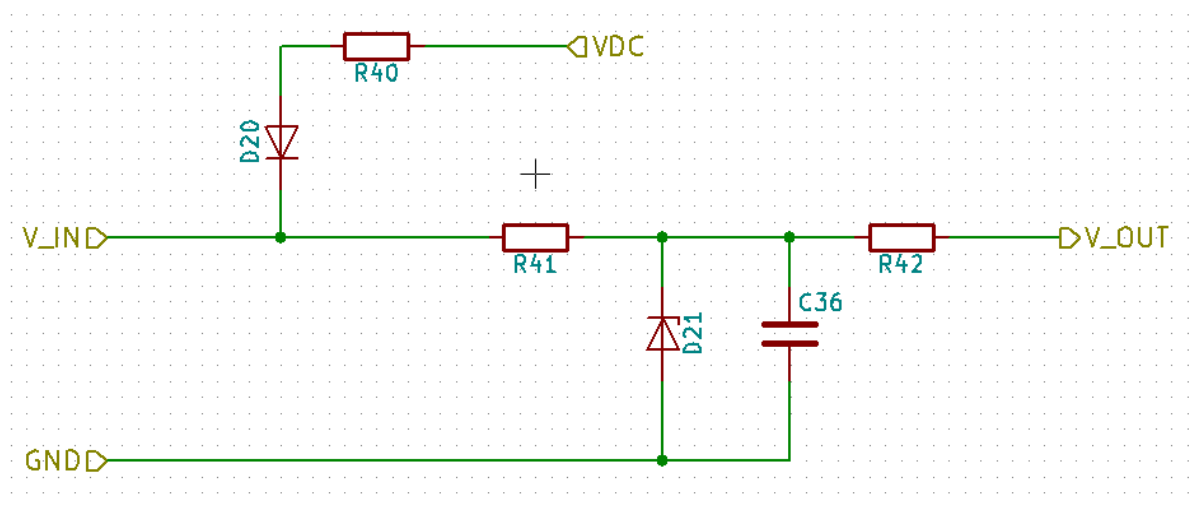
*Obr. 47 Vyrobený prototyp ECU*

## 6.1 PROCESOR

Součástí prototypu ECU je procesor STM32417. Dle katalogového listu na stránkách [29] je založen na architektuře procesorů ARM 32-bit Cortex-M4 o frekvenci 168 MHz. Velikost flash paměti je 1 Mbyte a je jím podporována řada periférií, jako jsou z těch základních analogové vstupy a výstupy s 12 bitovými převodníky, komunikace Inter-Integrated Circuit - I2C, SPI, sériové komunikace, CAN atd. Pro zavádění programu slouží SWD nebo Joint Test Action Group – JTAG. Samozřejmě tento procesor má řadu funkcí a vlastností, ale pro výčet byly zmíněny jen některé. Podrobnější popis je v jeho katalogovém listu. Z hlediska elektrického zapojení bylo vycházeno z manuálu a katalogového listu od společnosti STMicroelectronics s označením AN4488, AN2867 dostupné na stránkách [29]. Kde bylo popsáno připojení napájení procesoru, uzemnění, užití potřebných kondenzátorů, feritových korálků a rezistorů, aby procesor fungoval správně. Také byla potřeba vyřešit elektronické zapojení externího krystalu s frekvencí 8 MHz. Při návrhu rozmístění elektronických součástek pro fungování procesoru byl kladen důraz na jejich blízké umístění k procesoru, aby nedocházelo k rušení jeho funkce.

## 6.2 DIGITÁLNÍ VSTUPY

Elektrický obvod pro digitální vstupy byl sestaven dle schématu na obr. 48. Obvod je sestaven z rezistorů, diody, Zenerovy diody a kondenzátoru. Označený vstup elektrického obvodu V\_IN je spojen konektorem prototypu ECU a umožňuje připojení digitálního signálu z koncových spínačů pneumatických válců, signálů pedálů a ovládacích přepínačů a spínače. Je koncipován na vozidlové napětí nákladních vozidel 24 V s rezervou na vznik případných napěťových špiček. Takže pokud se jedná o připojení koncového spínače, který spíná zem, je obvod koncipován s tzv. pull-up zapojením. To je ta horní větev obvodu, která obsahuje rezistor a ochrannou diodu. K této větvi je připojeno napájení 24 V. Tímto zapojením je definován vstupní signál pro procesor, to znamená, pokud je koncový spínač sepnut, obvod je uzemněn a na pin procesoru jde nulové napětí, pokud je koncový spínač rozpojen, je přiváděno na pin procesoru napětí přes pull-up větev. To je i pro snímač plynového pedálu, kde se také spíná a rozepíná zem. V případě připojení signálu z indukčních snímačů spojivého a brzdového pedálu nebo kontrolních přepínačů a spínače je signál ve formě napětí 0 nebo 24 V. V tomto případě větev s pull-up rezistorem není v tištěném obvodu osazena. Dále v obou případech zapojení postupuje elektrický signál přes předřazený rezistor, který určuje proud, který poteče ochranou Zenerovou diodou, jeho hodnota se určí z katalogového listu diody, aby napětí směrem k pinu procesoru V\_OUT, bylo ořezáno na 3,9 V. Umístění kondenzátoru slouží jako tzv. dolní propust' pro filtrování vyšších frekvencí. Obecně řečeno, je tím zbavováno případného šumu v signálu. Posledním prvkem je rezistor již před výstupem tohoto obvodu a zároveň před vstupem do procesoru. Jedná se o ochranný prvek pro pin procesoru z hlediska proudů.

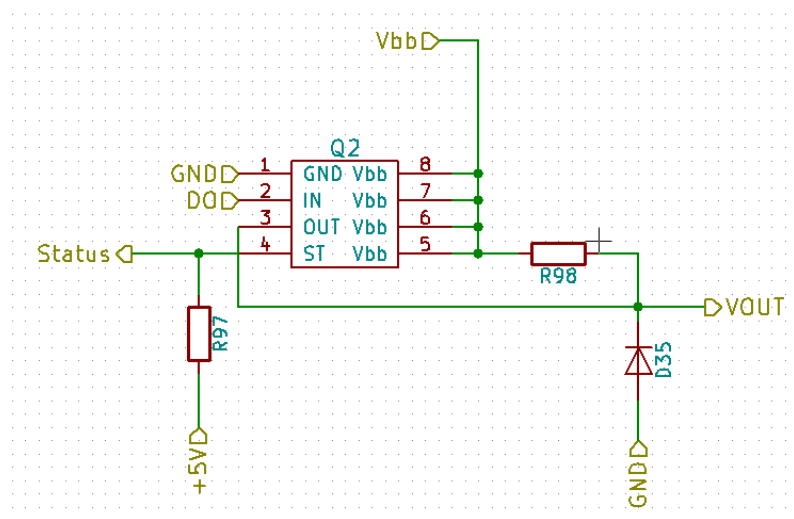


Obr. 48 Schéma zapojení elektronických součástek pro DI

## 6.3 DIGITÁLNÍ VÝSTUPY

Z procesoru je poslán digitální výstupní signál pro aktivaci akčních členů. Jelikož se jedná o elektroventily neboli o spínanou induktivní zátěž, bylo potřeba využít dostupné elektronické prvky pro spínání takového akčního členu. Důležité bylo, aby přímo digitálním signálem z procesoru byl spínán elektronický prvek, který sepne akční člen s parametry požadovaného napětí a proudu. Napětí bylo 24 V a proud okolo 0,51 A. Z hlediska zástavby a ručního osazování prototypu ECU byl nevhodnější tzv. Power Switch BSP752, který umožňuje spínat zařízení do napětí 51 V a s nominálním proudem až 1,3 A. To bylo z hlediska požadavků

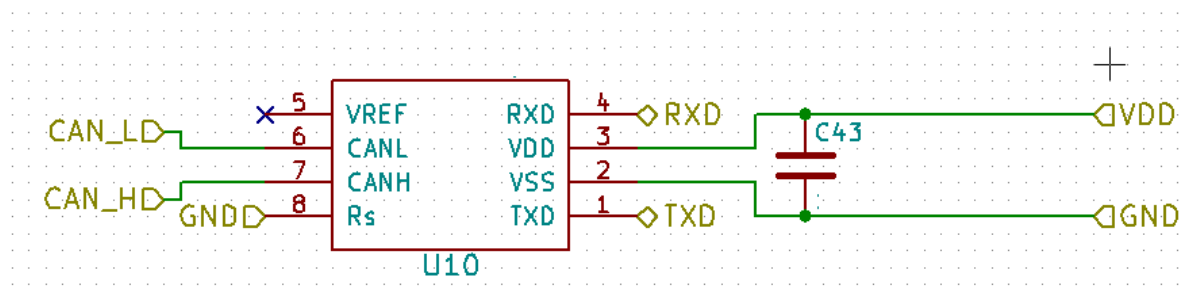
dostačující. Schéma zapojení je na obr. 49. Dále tato součástka umožňuje odesílat zpětnou vazbu o stavu z pinu ST a dle zapojení a tabulky v katalogovém listu [82] je možné např. zjistit, že je obvod rozpojen. To je pro kontrolu upadnutého kabelu zátěže, proto zde musí být navíc připojen rezistor mezi piny Vbb – napájecího napětí a OUT – výstupním napětím pro akční člen. Dále je obvod doplněn o diodu zapojenou v závěrném směru. Ta je zde umístěna z toho důvodu, že zátěž je induktivní akční člen. A pokud je induktivní zátěž odpojena, je její energii potřeba zmařit, aby nedošlo k poškození elektronické součástky pro spínání zátěže, proto je zde dioda, přes kterou toto zmaření energie proběhne. Posledním prvkem užitým v obvodu je rezistor připojen na pin ST a druhým koncem na napájecí napětí +5 V. Toto zapojení je dáno katalogovým listem.



Obr. 49 Schéma zapojení elektronických součástek pro DO

## 6.4 CAN

Pro spojení s CAN sběrnici bylo nutné mezi procesor a CAN piny na konektoru ECU umístit CAN transeiver. Je to elektronické rozhraní mezi CAN protokolem a CAN sběrnici. Pro prototyp bylo použito zapojení dle katalogového listu elektronické součástky MCP2551, kde je připojen kondenzátor pro vyhlazování napájení. Z hlediska pinů jsou zde CANL - Low-Level Voltage vstup/výstup a CANH - High-Level Voltage vstup/výstup, které jsou spojeny s CAN sběrnici a piny RXD - Receive Data Output a TXD - Transmit Data Input, které jsou spojeny s procesorem pro přenos přijímaných a odesílaných CAN zpráv.



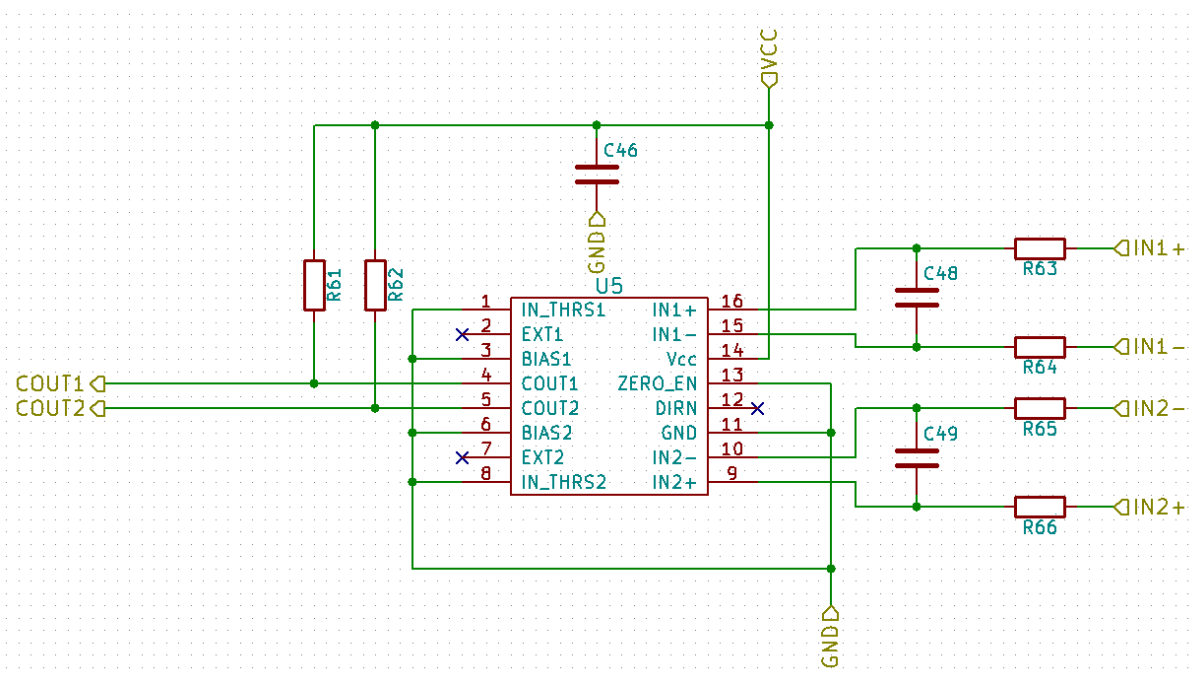
Obr. 50 Schéma zapojení elektronických součástek pro CAN

## 6.5 NAPÁJENÍ ECU

Do konektoru prototypu ECU je připojeno napájecí napětí ze sv. 30 – stálé napětí a ze sv. 15 – napětí spínané zapalovacím klíčkem. Ze sv. 15 nebo sv. 30 jsou napájeny hlavní části jako procesor, externí paměť a plus MEMS modul. V obvodu jsou paralelně zařazeny elektrolytické kondenzátory pro vyrovnávání výkyvu napájecího napětí a dále je v obvodu DC/DC napěťový regulátor, který snižuje napětí na 9 V, aby bylo možné vést toto napětí do stabilizátorů pevného napětí pro 3,3 a 5 V. Tato napětí jsou již určena pro jednotlivé elektronické součástky. Sv. 30 má před elektrolytickými kondenzátory, DC/DC napěťovým regulátorem a stabilizátory pevného napětí umístěn tzv. Power Switch, který až po sepnutí startovacího klíčku a vyhodnocení tohoto stavu procesorem je sepnut a tím jsou napájeny i další části prototypu ECU. Tento spínací prvek má zpětnou vazbu, a dokud procesor nevidí, že jsou opravdu napájeny i ostatní periferie ECU nepostupuje v inicializaci dále v hlavním programu s řídicím algoritmem. Součástí obou větví jsou i výstupy pro kontrolu napětí na těchto svorkách. Součástí tohoto obvodu je i napěťový zdroj pro snímač natočení volantu s precizní napěťovou referencí, aby výstupy ze snímače natočení volantu byly přesné a nedocházelo ke zpracování špatných informací a tím k mystifikování řídicího algoritmu. Pro vývojové účely je zde možnost napájet procesor i přes USB konektor. Rozdělení na oblasti napájení tištěného spoje a daných elektronických součástí je z toho důvodu, aby při vypnutí startovacího klíčku nedocházelo ke zbytečnému vybíjení baterie vozidla, a tím jsou napájeny jen nejnnutnější obvody pro opětovné probuzení procesoru ze spacího módu.

## 6.6 VSTUPY OTÁČEK KOL

Užité indukční snímače otáček kol generují zjednodušeně řečeno sinusový signál s měnící se amplitudou a frekvencí v závislosti na otáčkách.



Obr. 51 Schéma zapojení elektronických součástek pro snímání otáček kol

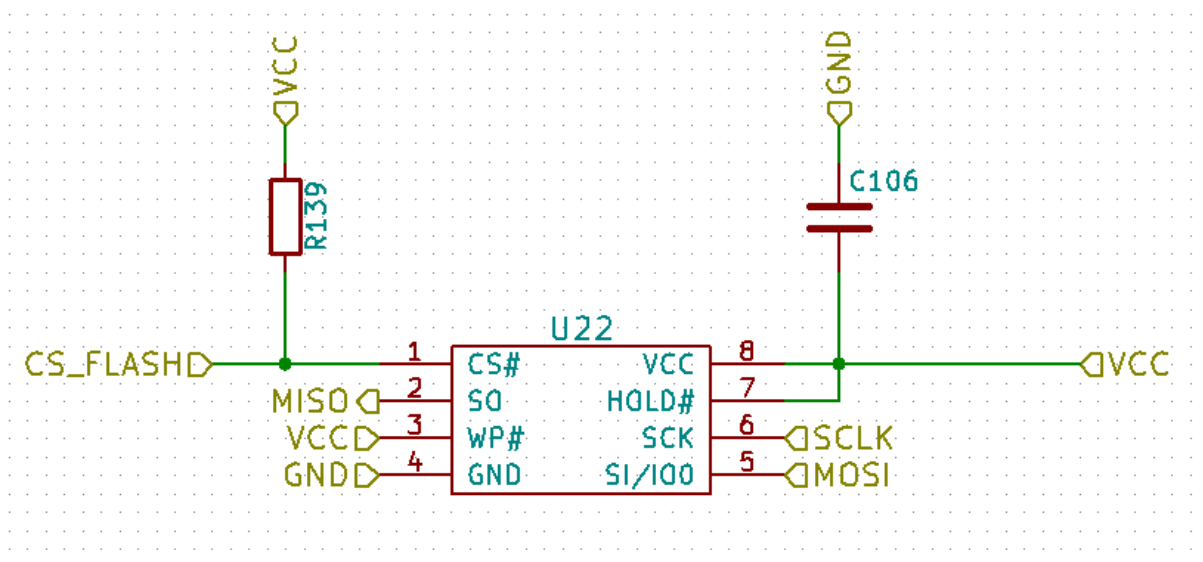
Tento signál je zapotřebí předpřipravit procesoru do formy digitálního signálu, aby následně mohl být zpracován přes přerušeni v procesoru. Proto byla využita elektronická součástka



přímo na zpracování takových signálů MAX9926 a v podstatě se jedná o komparátor, který při překročení určitých hodnot napětí generuje digitální výstup. Na obr. 51 je zobrazeno zapojení, které vychází z doporučených zapojení dané katalogovým listem. Samozřejmě je k dispozici několik možností. Zde byla vybrána možnost pro detekci přechodu přes nulové napětí, kde hodnoty pro změnu stavu výstupního digitálního signálu jsou +10 mV a v záporné části max. -6,5 mV. Toto nastavení bylo užito, protože je zapotřebí snímat otáčky od co nejnižších. Jelikož signál ze snímače otáček mění amplitudu a při hodnotě otáček blízké nule dochází ke snižování amplitudy až k nule. Ostatní součástky obvodu jsou rezistory a kondenzátory, které byly takto připojeny dle nastavení pro daný typ snímání. Obecně tato elektronická součástka má čtyři vstupy IN1+, IN1-, IN2+, IN2- pro dva snímače otáček a výstupem jsou COUT1 a COUT2 ostatní piny jsou určeny pro různá nastavení a pro externí referenční vstupy. Výstupní digitální signály jsou procesorem zpracovávány přes přerušení a po získání periody jsou výpočtem získány otáčky.

## 6.7 PAMĚŤ

V prototypu ECU byla využita externí paměť pro ukládání poruchových zpráv DM1 a DM2, případně nějaký omezený sběr dat.



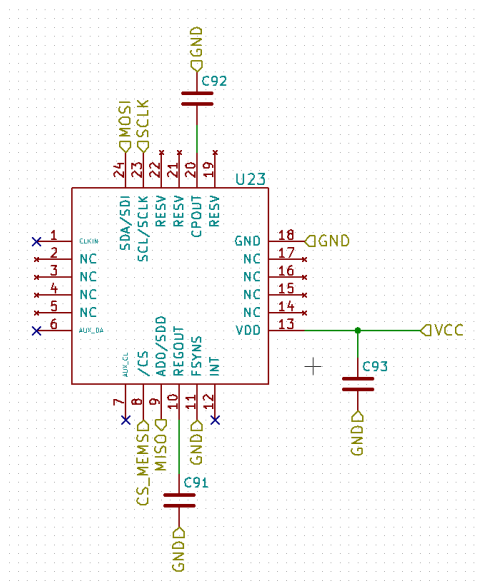
Obr. 52 Schéma zapojení elektronických součástek pro externí paměť

Externí paměť je typu flash a má velikost 4 Mbit, které lze dělit po sektorech 4 KByte, 32 KByte a 64 KByte. Komunikace je realizována pomocí SPI a to až do rychlosti 50 MHz. Jelikož ke stejnému mastru je připojen i MEMS modul, který neumožňuje takové přenosové rychlosti, byl přenos sjednocen pod 1 MHz. Zapojení ostatních součástek obvodu je dáno doporučením v katalogovém listu.

## 6.8 GYROSKOP

Další nástavbou vyvíjeného systému je systém pro vyhodnocování, kde se vozidlo nachází. V tomto případě především na jakém sklonu vozovky v daném okamžiku je. Proto byl užit MEMS modul s gyroskopem a akcelerometrem. Jedná se o elektronickou součástku s názvem MPU-6000, která komunikuje pomocí SPI o přenosové rychlosti až 20 MHz, ale to za předpokladu, že je užito registru pro přerušení. V tomto případě bylo využito dostatečného čtení dat pomocí vysoko rychlostní charakteristiky, kdy je přenosová rychlost do 1 MHz. Pomocí

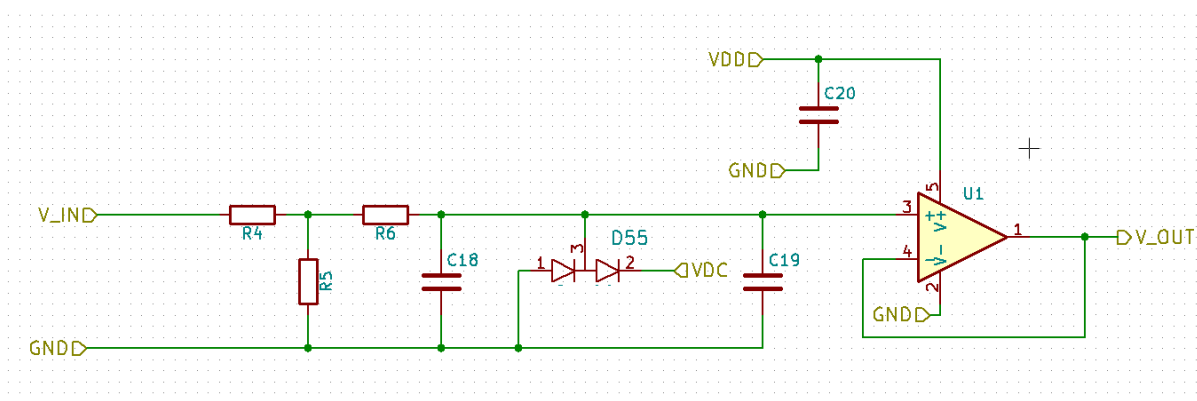
definované mapy registrů jsou čteny hodnoty úhlové rychlosti a zrychlení pro tři osy. A také je čtena teplota pro případnou kontrolu, zda nedochází k přehřívání. Nastavení čtených hodnot úhlové rychlosti gyroskopu je od  $\pm 250$  °/s do  $\pm 2000$  °/s a u akcelerometru od  $\pm 2$  g do  $\pm 16$  g. Zapojení dalších elektronických součástek obvodu vychází z katalogového listu a příslušné konfigurace. Schéma je zobrazeno na následujícím obrázku.



Obr. 53 Schéma zapojení elektronických součástek pro MEMS modul s tříosým gyroskopem a akcelerometrem

## 6.9 ANALOGOVÉ VSTUPY

Obvody pro analogové vstupy byly sestaveny na vstupu V\_IN z odporového děliče, který snižuje napětí na polovinu, to je z důvodu výstupního signálu ze snímačů natočení volantu a snímače tlaku vzduchu, který je 5 V.



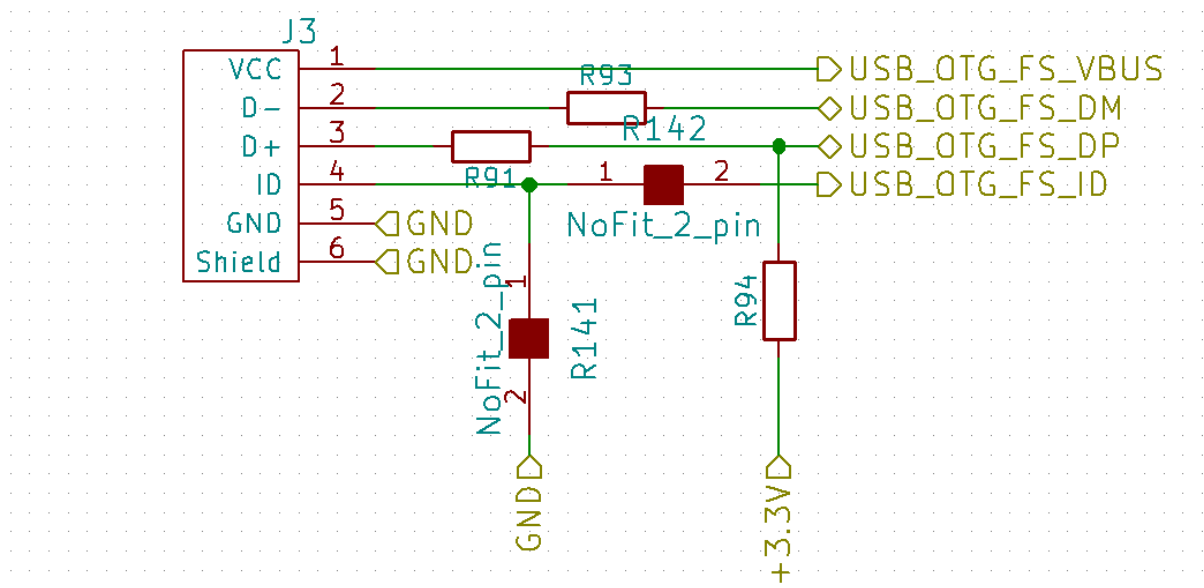
Obr. 54 Schéma zapojení elektronických součástek pro AI

Procesor však má rozsah analogových vstupů jen od 0 do 3,3 V. Další sériově zapojený rezistor slouží pro nastavení proudu pro diodu BAV99, která je zde jako ochrana proti vstupnímu napětí, které by překonalo hodnotu +3,3 nebo se snížilo pod 0 V. Následuje operační zesilovač zapojený jako sledovač napětí. Toto zapojení umožňuje oddělení elektrického obvodu snímače a elektrického obvodu procesoru a tím nedochází k rušení signálu. Ale zároveň výstupní napětí

odpovídá přiváděnému vstupnímu napětí. Posledními součástkami jsou kondenzátory, které opět slouží k vyhlazení nebo filtraci signálu a napájení.

## 6.10 USB

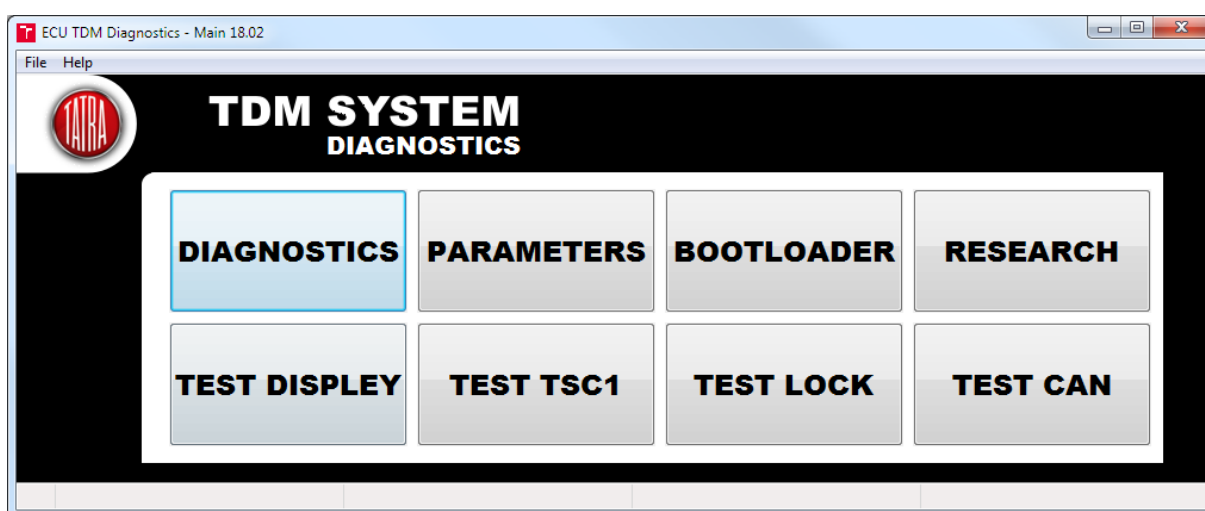
Posledním významným obvodem prototypu ECU je obvod pro USB konektor, který je určen jen pro výzkumné účely a především bylo toto zapojení užito pro napájení při zprovoznování prototypu ECU. Schéma je vidět na obr. 55, kde jeho zapojení je inspirováno z prototypových desek DISCO board od STMicroelectronic.



Obr. 55 Schéma zapojení elektronických součástek pro USB

## 7 SOFTWAREVÝ INTERFACE ECU

Pro komunikaci s prototypem ECU a nahraného programu v ní byl naprogramován softwarový interface pomocí jazyka C/C++ a knihovny wxWidget [83] určené pro programování Graphical User Interface – GUI. Do softwaru je dále implementována knihovna pro PCAN. To je hardwarové zařízení pro komunikaci s CAN sběrnici. Proto i jakákoli komunikace s ECU je realizována přes CAN zprávy. Základní grafické rozhraní softwaru je zobrazeno na obr. 56, kde uživatel má nabídku několika dalších podprogramů pro diagnostiku ECU, nastavení parametrů ECU, zavádění nového programu do procesoru. Další nabídky jsou určeny pro výzkum a vývoj a běžnému uživateli nejsou zpřístupněny. Z každého podprogramu se uživateli zase kliknutím na logo dostane do tohoto úvodního okna. Obecně v paměti procesoru je umístěn základní naprogramovaný program pro CAN komunikaci s tímto softwarem pro diagnostiku a pomocí této komunikace vše probíhá v následujících podprogramech.

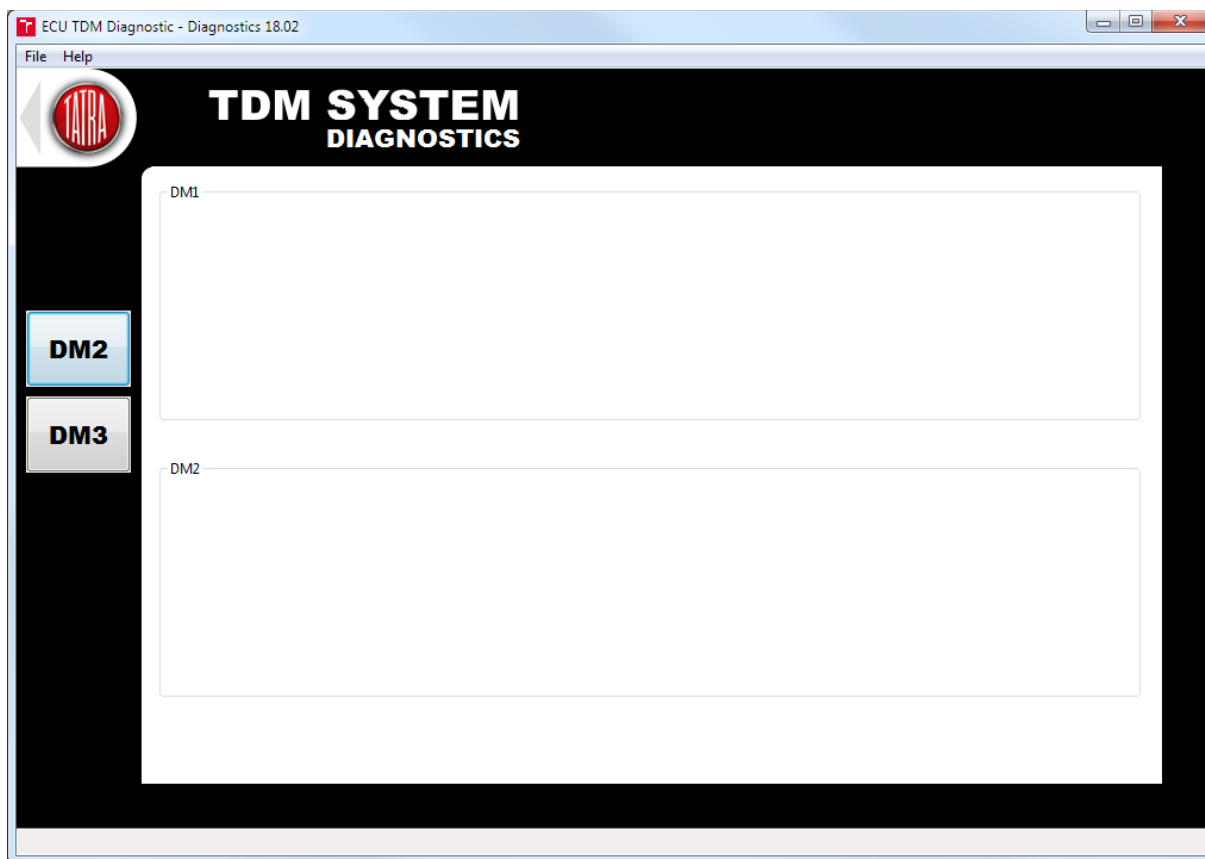


Obr. 56 Základní nabídka funkcí diagnostického programu

### 7.1 PROGRAM DIAGNOSTICS

První z podprogramů je určen pro diagnostiku ECU a je zobrazen na obr. 57. Takže pokud ECU odesílá diagnostické CAN zprávy DM1 - Diagnostic Message 1 pro aktivní poruchy, tak při spuštění tohoto programu jsou zobrazovány. Z hlediska formátu zpráv vše vychází z normy J1939 a dle počtu zpráv jsou pro jednu poruchu odesílány ve zprávě s ID 0x18FECA09 a pokud je poruch více jsou odesílány v tzv. multipaketu s ID 0x1CCBEF09, kde do dat zprávy jsou postupně umísťovány data s popisem poruch DTC - Diagnostic Trouble Code a ty obsahují hodnoty SPN - Suspect Parameter Number – jedná se o označení určité skupiny parametrů souvisejících s danou poruchou. Pokud hodnota SPN není mezi již definovanými normou, uživatel může použít rozsah hodnot pro vlastní definice. Další proměnou je FMI - Failure Mode Identifier – která určuje v podstatě typ poruchy např. to, že snímač je mimo jeho provozní rozsah. SPN Conversion Method – CM – nastavujeme variantu uspořádání dat. Nyní varianty 1-3 již nejsou povoleny a používá se jen 4 varianta, tedy s nastavením na hodnotu 0. Occurrence Count – OC – hodnota se zvyšuje, pokud přišlo první hlášení o poruše, tedy hodnota je 1, dále se hodnota zvýší, pokud chyba zmizí a znovu nastane. To vše probíhá do hodnoty až 126 a potom zůstává stejná, přestože dále chyby přibývají. Pokud je zadána hodnota 127, znamená to, že výskyty chyb nejsou počítány. Tato data jsou implementována v ECU a odeslány. Pokud je spuštěn program pro diagnostiku, jsou tyto zprávy zase rozklíčovány a jsou zobrazeny

jednotlivé poruchy DM1. Další možností uživatele je vyžádání zprávy DM2 pomocí tlačítka vlevo. To znamená, že ECU odešle dříve aktivní závady zapsané v externí paměti s ID CAN zprávy 0x18FECB09, pokud je zapsaných poruch více než jedna, jsou posílány zase v tzv. multipaketu ve stejném formátu jako DM1. Tlačítko DM3 odešle požadavek na vymazání dříve aktivních závad.

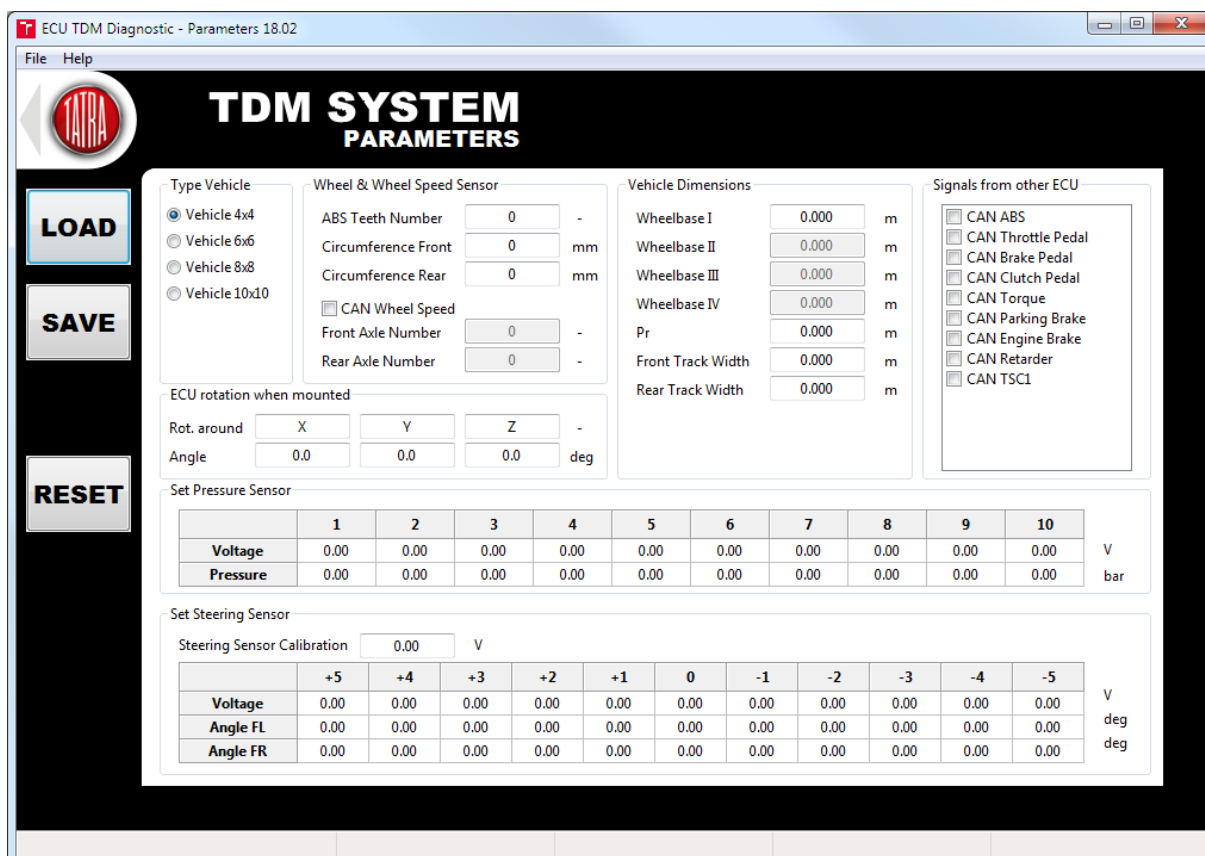


Obr. 57 Program Diagnostic pro sledování aktivních a dříve aktivních chyb a jejich mazání

## 7.2 PROGRAM PARAMETERS

Program pro nastavování uživatelských parametrů je zobrazen na obr. 58. Uživateli nebo spíše servisu vozidla je umožněno nastavovat různé konfigurace. Prvně je nastavován typ vozidla, pro který je mechatronický systém určen, takže např. vozidlo s pohonem 4x4 až 10x10. Dále je nastavován počet zubů kola pro ABS senzory, obvod pneumatiky přední a zadní nápravy, aktivace čtení informací o rychlostech kol z CAN zprávy odesílané nějakým brzdovým systémem např. ABS a k tomu přísluší nastavení čísla nápravy, kde se nacházejí snímače otáček daného systému. Další nastavitelnou sekcí je tzv. Vehicle Dimensions, kde jsou nastavovány rozvory a rozchody náprav a poloměr rejdu v rovině středu kola. Následuje nastavení, zda některé signály chce uživatel získávat pro řídicí algoritmus z dalších ECU. To jsou signály, zda je aktivní ABS, signály pedálů, signál parkovací brzdy, signál motorové brzdy, signál retardéru, procentuální hodnota hnacího momentu motoru a nastavení odesílání zprávy pro omezování hnacího momentu motoru a otáček při uzavírání diferenciálů. Pokud prototyp ECU disponuje nástavbou pro sledování okolí, je zpřístupněno nastavení polohy MEMS modulu, tedy v podstatě poloha řídicí jednotky ve vozidle vůči základnímu souřadnému systému vozidla dle ISO. Následují nastavitelné hodnoty senzoru tlaku vzduchu v pneumatickém okruhu a tabulka nastavení snímače natočení volantu vzhledem k zatočení pravého a levého kola přední nápravy.

Zde je také nastavována hodnota napětí výstupního signálu ze snímače natočení volantu při přímé jízdě vozidla. Průběh přenesení nových parametrů do ECU nebo naopak vyčtení aktuálních parametrů je pomocí tlačítek vpravo (LOAD, SAVE). Po těchto operacích je nutné kliknout na tlačítko RESET pro restartování ECU. Pokud uživatel načítá parametry, tak v první fázi se odešle zpráva na spojení s ECU, pokud ECU do časového limitu odpoví, začne odesílat uložené parametry na příslušných adresách a program hodnoty zpracuje. Po této operaci jsou všechny načtené hodnoty zobrazeny v grafickém rozhraní. Nyní uživatel může některé nebo všechny parametry změnit a následně přes tlačítko SAVE nahrát do ECU. Operace probíhá tak, že pokud by ještě nebyla navázána komunikace s ECU, tak by k ní došlo a pokud by byla úspěšná, začnou se odesílat CAN zprávy s parametrem a jeho adresou uložení v paměti procesoru. Vždy na tuto zprávu přijde potvrzení s kontrolním součtem, zda byl parametr správně uložen. Po odeslání posledního parametru je zaslán i celkový kontrolní součet uložených parametrů, a pokud je vše v pořádku, software zobrazí okno s úspěšným nahráním nových parametrů do ECU. Následně je zapotřebí kliknout na tlačítko RESET a ECU se spustí již s novými nastavenými parametry. Uživatel má ještě možnost dané konfigurace ukládat do souboru a naopak je i z něho načítat, aby pro každé vozidlo nemusely být všechny parametry neustále vypisovány ručně pro stejné konfigurace.

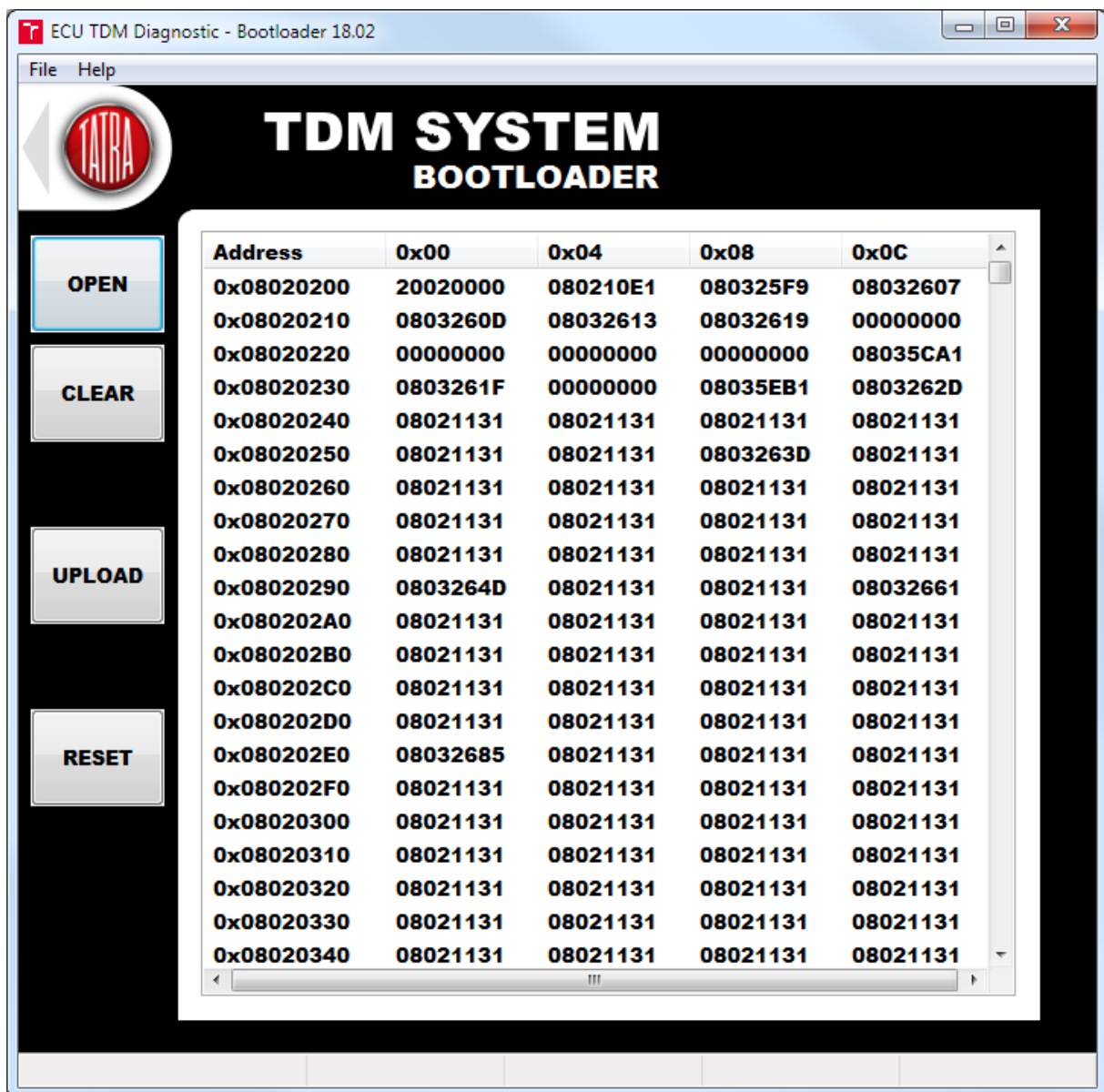


Obr. 58 Program Parameters pro nastavování základních parametrů systému uživatelem

### 7.3 PROGRAM BOOTLOADER

Důležitým podprogramem je tzv. Bootloader zobrazený na obr. 59, který umožňuje otevřít (OPEN) zkompileovaný kód programu s řídicím algoritmem ve formátu hex. Zároveň ho zase ze svoji paměti může smazat pomocí tlačítka CLEAR. Pokud je soubor načten a uživatel zmáčkne tlačítko UPLOAD, software nejdříve naváže spojení s ECU. Pokud je vše v pořádku,

tak začne odesílat CAN zprávy obsahující postupně celý nový zaváděný program s řídicím algoritmem do paměti procesoru. Zpráva vždy obsahuje adresu paměti, kam má být zapsána část zaváděného programu. Vždy přijde odpověď na tuto zprávu s kontrolním součtem. Pokud vše proběhne v pořádku a po posledních odeslaných 4 bytech je i celkový kontrolní součet shodný s odeslanými daty, tak se zobrazí okno s úspěšným nahráním nového programu do paměti procesoru v prototypu ECU. Následuje zase kliknutí na tlačítko RESET a ECU se restartuje. Následně je spuštěna nová verze programu s řídicím algoritmem. Výhodou tohoto rozhraní je, že při každé změně řídicího algoritmu nemusí dodavatel jet k zákazníkovi, aby mu tam nahrál novou verzi programu, ale stačí to poslat elektronicky a uživatel pomocí tohoto softwaru a běžně používaného PCAN rozhraní může nahrát program samostatně.

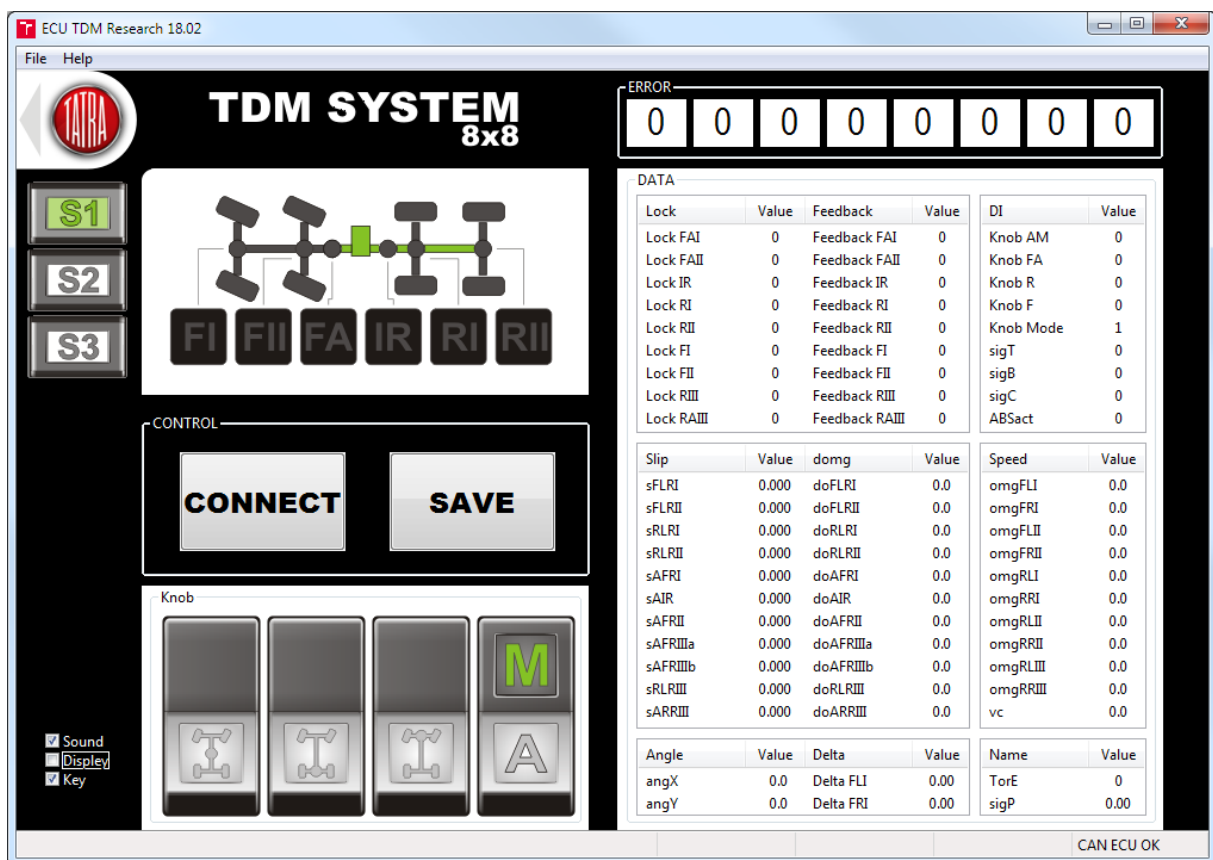


Obr. 59 Program Bootloader pro zavádění nové verze programu s řídicím algoritmem do ECU

## 7.4 PROGRAM RESEARCH

Tento podprogram, zobrazený na obr. 60, je určen pro vývoj a výzkum tohoto systému. Tento program funguje jako online zobrazování dat a stavů vstupních, výstupních a i vnitřních veličin

řidičeho algoritmu. V pravé části je graficky zobrazeno rozhraní, které odpovídá tomu, co je použito i v dotykovém displeji mechatronického systému. Tam uživatel vidí, co je aktuálně uzavřeno z hlediska diferenciálů, nebo vidí, zda je aktivován pohon předních náprav. Zároveň pokud není v systému užit dotykový displej, tak aktivací vlevo dole se může tento program chovat jako on. Tím uživatel může nastavovat jízdní módy pomocí tlačítek S1-S3. V pravé části jsou zobrazovány data a poruchy. Ve střední části grafického rozhraní jsou dvě tlačítka, kdy tlačítko CONNECT slouží pro spojení softwaru s řídicí jednotkou přes definovaný klíč a druhé tlačítko slouží pro sběr dat v rozsahu posledních 3 minut. Ty jsou následně zpracovávány a vyhodnocena. Tento sběr dat je užíván při testování kompletního vyvíjeného autonomního systému, nebo připraven pro sofistikovanější analýzu při vzniku poruchy aniž by to bylo vyřešeno pomocí podprogramu se základní diagnostikou.

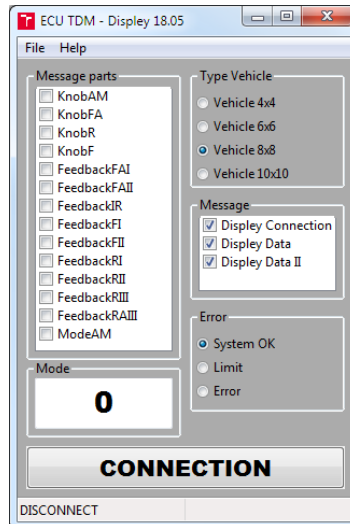


Obr. 60 Program pro online sledování hlavních vnitřních proměnných řídicího algoritmu

## 7.5 PROGRAM TEST DISPLEJE

Jedná se o program pro testování a ověření komunikace dotykového displeje. Tento program byl využit pro vývojové práce na dotykovém displeji, aniž by musel být k dispozici prototyp ECU. Kde se uživatel připojí ke sběrnici CAN a odesíláním a přijímáním příkazů pomocí CAN zpráv je schopen ověřit funkčnost dotykového displeje v celém rozsahu. Uživatel postupným klikáním na příslušné veličiny odpovídající hodnotám kontrolních přepínačů a spínače ECU, hodnotám koncových snímačů pneumatických válců, nastavení typu vozidla, poruchovým nebo limitním omezením a počtu odesílaných zpráv vygeneruje data odesílaná CAN zprávou pro displej. Naopak zase přijímá požadavky na změnu módu na displeji a příslušně na to reaguje. Software se chová naprosto stejně z hlediska komunikace ECU a displeje jako by do sběrnice CAN byl připojen prototyp ECU.

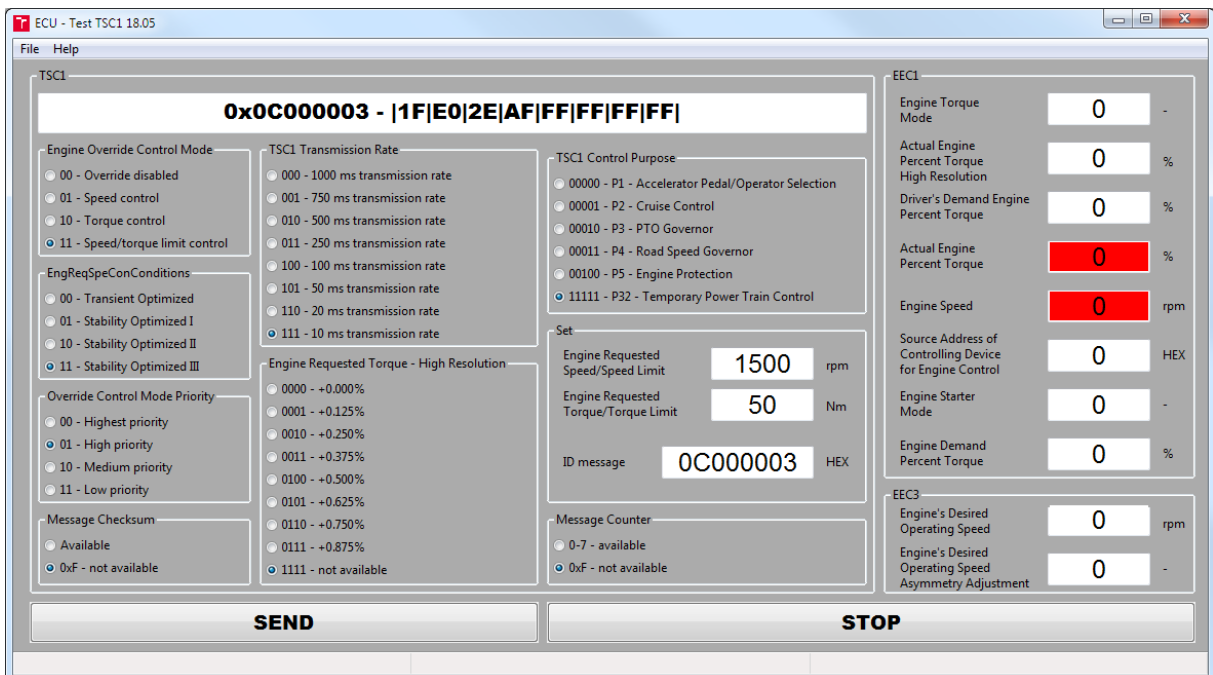




Obr. 61 program pro testování správné funkce dotykového displeje

## 7.6 PROGRAM TEST TSC1

Další program byl naprogramován pro účely testování reakce motoru na zasílané omezení pomocí CAN zprávy TSC1. V grafickém rozhraní uživatel nastaví příslušnou konfiguraci a po stisknutí tlačítka SEND se začne cyklicky dle nastavení odesílat CAN zpráva TSC1. Na ni podle požadavku a dalších priorit v řídicí jednotce motoru, začne motor reagovat. To vše pokud tuto zprávu a zdrojovou adresu v ID zprávy podporuje.



Obr. 62 Program pro testování omezení otáček a hnacího momentu motoru přes CAN zprávu TSC1

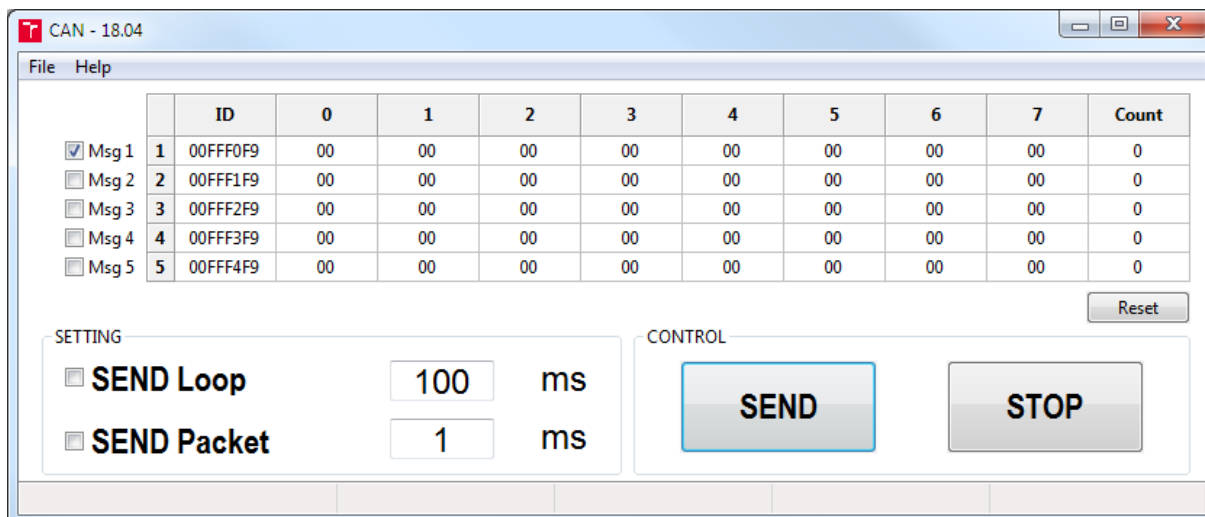
## 7.7 PROGRAM TEST LOCK

Tento program je ve fázi vývoje a jedná se o program určený k manuálnímu testování akčních členů. Principem bude odesílání specifikované CAN zprávy s požadavkem na aktivaci

příslušného elektroventilu. Bude to především určeno pro otestování signalizace poruchy elektroventilu.

## 7.8 PROGRAM TEST CAN

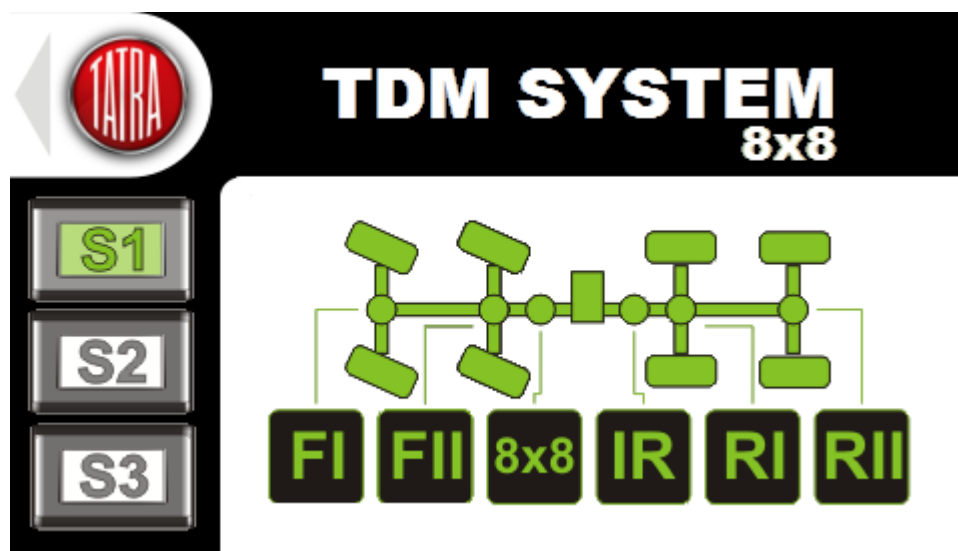
Poslední program zobrazený na obr. 63, je určen pro odesílání uživatelsky definovaných CAN zpráv pro různá testování. Uživatel může odesílat jednu zprávu nebo celý paket. Také může nastavit periodu cyklického odesílání, periodu mezi odesílanými zprávami v paketu nebo může odesílat jen jednorázové zprávy nebo pakety.



Obr. 63 Program pro testování odesílání CAN zpráv

## 8 DISPLEJ

Součástí celého vyvíjeného systému je i dotykový displej, který v první fázi byl užit s označením 32F746GDISCOVERY. Nyní je vývoj ve fázi užití i displeje Opus 3Ae, který zde nebude popisován.



Obr. 64 Dotykový displej pro vyvíjený mechatronický systém

### 8.1 HARDWARE

Z hlediska hardwaru byl tedy užit dotykový displej 32F746GDISCOVERY, který je založen na procesoru STM32F746G s pamětí 1 Mbyte. Tento hardware disponuje různými perifériemi pro SPI a I2C komunikaci. Má vstupy pro kameru, USB, audio, SD kartu atd. Co však u hardwaru chybělo, byly piny pro CAN komunikaci, přestože to daný procesor umožňuje. Z toho důvodu stačilo mírně upravit hardware v podobě odpájení některých elektronických součástí, aby mohly být některé piny užity pro CAN komunikaci. Pro daný mechatronický systém je to také zároveň jedinou využitou komunikací tohoto dotykového displeje.

### 8.2 SOFTWARE

Vnitřní program byl programován v jazyce C++ ve vývojovém prostředí mbed [80]. Kde v první fázi byla navržena grafika a obrázky byly konvertovány do matice a umístěny do souborů projektu. S nimi již pracuje naprogramovaný algoritmus a dle přijímaných CAN zpráv a dat z nich zobrazuje příslušné obrázky nebo texty na obrazovce displeje. Uživatel má tedy možnost kliknout na logo v levém horním rohu, kde se zobrazí nabídka ECU jednotek, se kterými displej komunikuje, po výběru jednotky pro uzavírání diferenciálů je zobrazeno grafické rozhraní zobrazené na obr. 64 nebo na obr. 65 (reálný provoz). Tam má uživatel k dispozici tři tlačítka, kterými nastavuje jízdní módy řídicí jednotky a dále jsou zde zobrazovány informace o uzavření příslušného diferenciálu nebo aktivaci pohonu všech kol. To je zobrazováno rozsvěčováním jednotlivých částí vozidla a zároveň příslušného označení. V horní části je zobrazován název systému, pokud je vše v pořádku a pokud je systém nějakým způsobem limitován, změní se tento název na Limit ve žluté barvě. Pokud je v systému porucha, zobrazí se nápis na Error v červené barvě. Tímto prvkem systému je řidič informován o daném stavu a může nastavovat příslušné jízdní módy ECU.

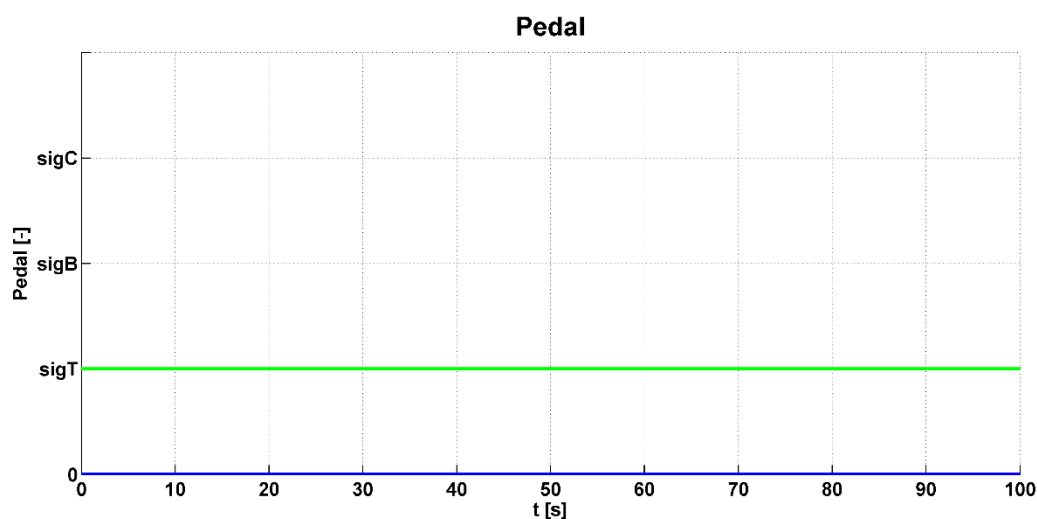
## 9 FINÁLNÍ TESTOVÁNÍ MECHATRONICKÉHO SYSTÉMU

Celkové testování vyvinutého autonomního mechatronického systému pro uzavírání diferencíálů probíhalo opět na prototypu vozidla s pohonem 8x8. Systém byl sestaven již z finálních senzorů, akčních členů, prototypu ECU, diagnostického softwaru a dotykového displeje. Vše je názorně zobrazeno na obr. 65. Testování probíhalo znovu v různých podmínkách od asfaltové silnice po nejnáročnější terény a povětrnostní podmínky.

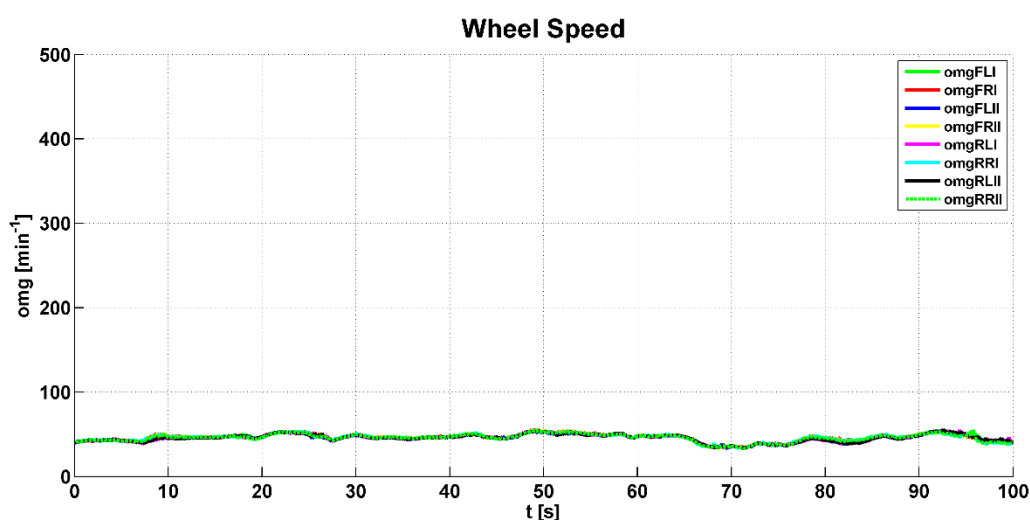


*Obr. 65 Testování kompletního vyvinutého autonomního mechatronického systému pro uzavírání diferencíálů*

Během testování již byly laděny jen drobné úpravy a nebyl předpokládán zásadní zásah do filozofie celého systému. Docházelo jen k ladění řídicích parametrů, aby systém reagoval rychle a spolehlivě ve všech situacích. Na následujících grafech je zobrazena ukázka sbíraných dat během jednoho z mnoha testů. Zde je jízda lehčím terénem okolo rychlosti vozidla 10 km/h. Během jízdy řidič měl neustále sešlápnut plynový pedál, jak je vidět na obr. 66 a záznam je pro 100 s. Během testování nedocházelo k tak významným prokluzům kol, jako u náročného terénu. To je viditelné i na grafech zobrazující průběh otáček kol obr. 67, ze kterých je vyhodnocen skluz mezi jednotlivými hřídelemi obr. 68. Zároveň je kontrolován rozdíl otáček mezi hřídelemi se zubovými spojkami na obr. 69, v tomto testu nedocházelo k překročení definovaného limitu a tím omezení řízení. Z otáček vyplývá i dosažená rychlost vozidla, která je zobrazena na obr. 70.

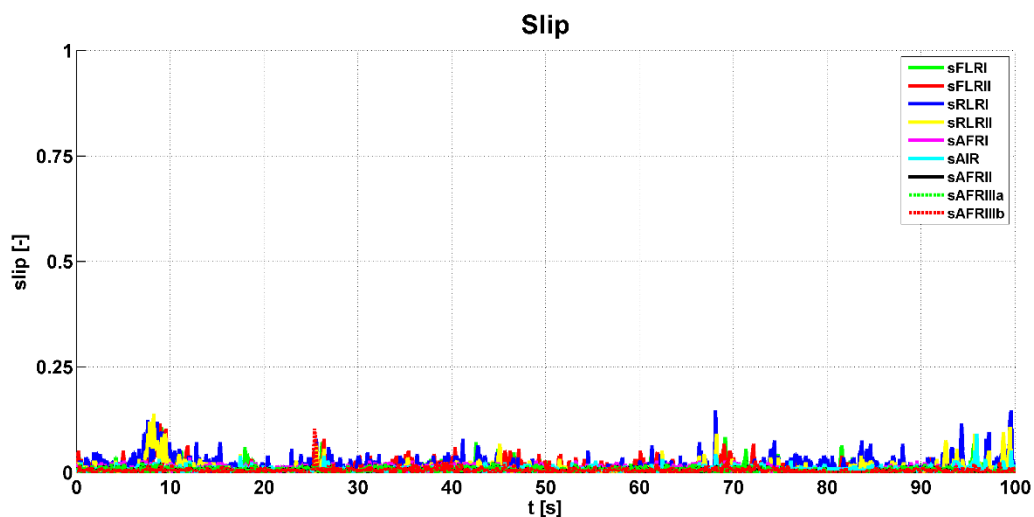


Obr. 66 Průběh signálů pedálů při finálním testování

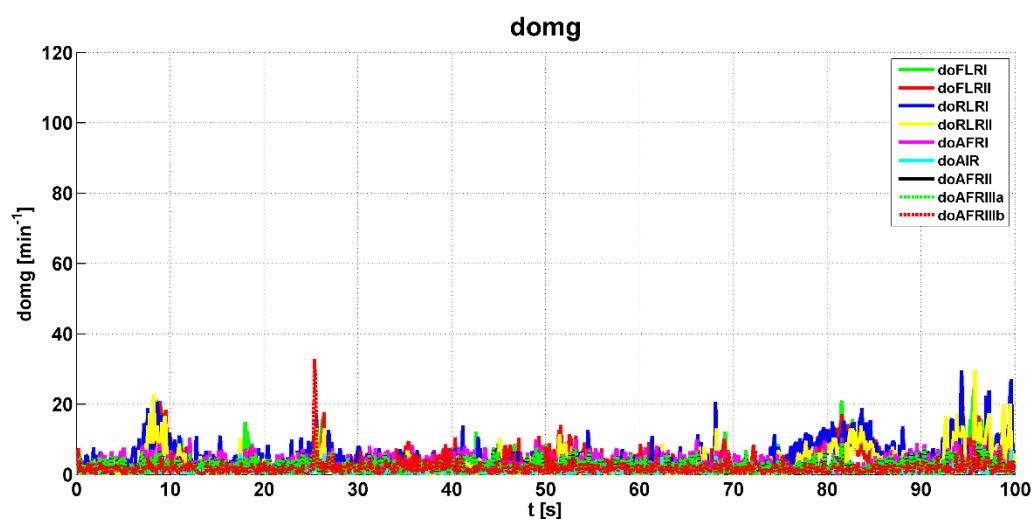


Obr. 67 Průběh signálů otáček jednotlivých kol při finálním testování

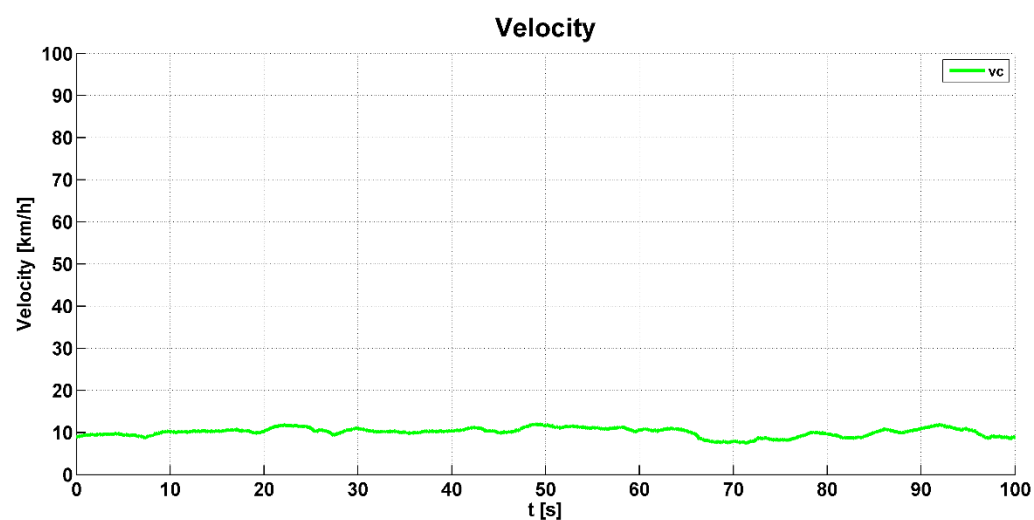
V grafu na obr. 71 je zobrazen průběh zatočení předních kol vozidla, kde je vidět, že vozidlo různě měnilo směr dle terénního testovacího okruhu. V tomto případě nešlo o překročení ani nastavených omezení z hlediska zatočení předních kol, jelikož max. hodnota úhlu se během tohoto testu pohybovala okolo 15°.



Obr. 68 Průběh signálů skluzu při finálním testování

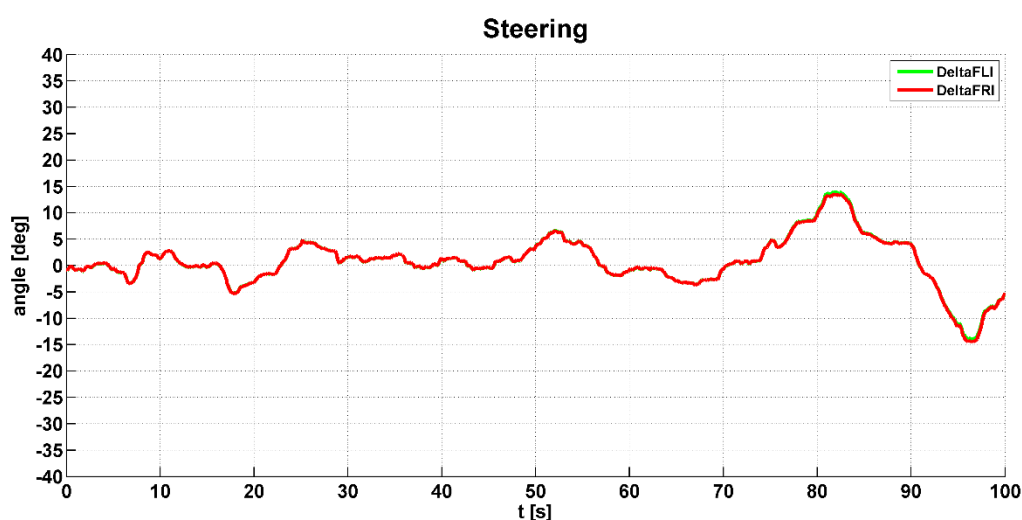


Obr. 69 Průběh signálů rozdílu otáček spojovaných hřídelí zubovými spojkami při finálním testování

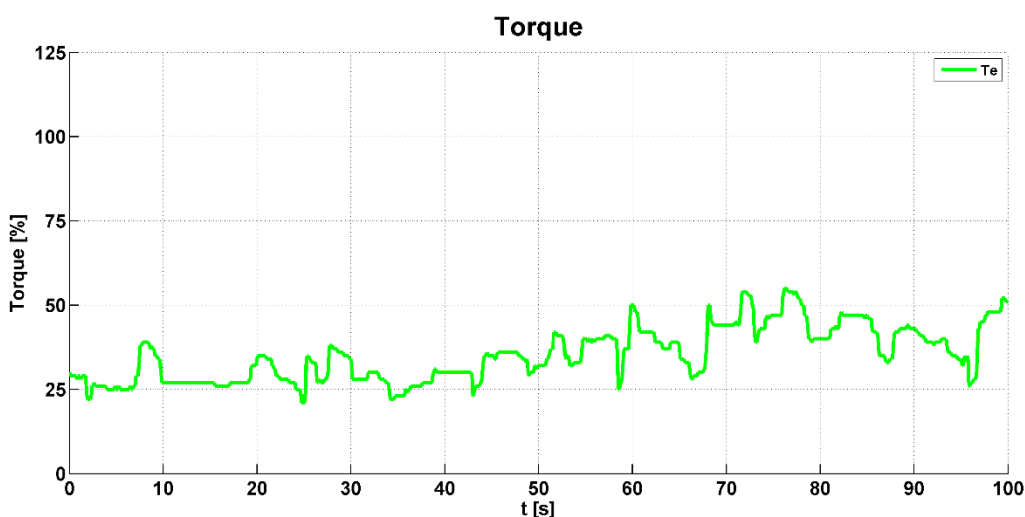


Obr. 70 Průběh signálu rychlosti vozidla při finálním testování

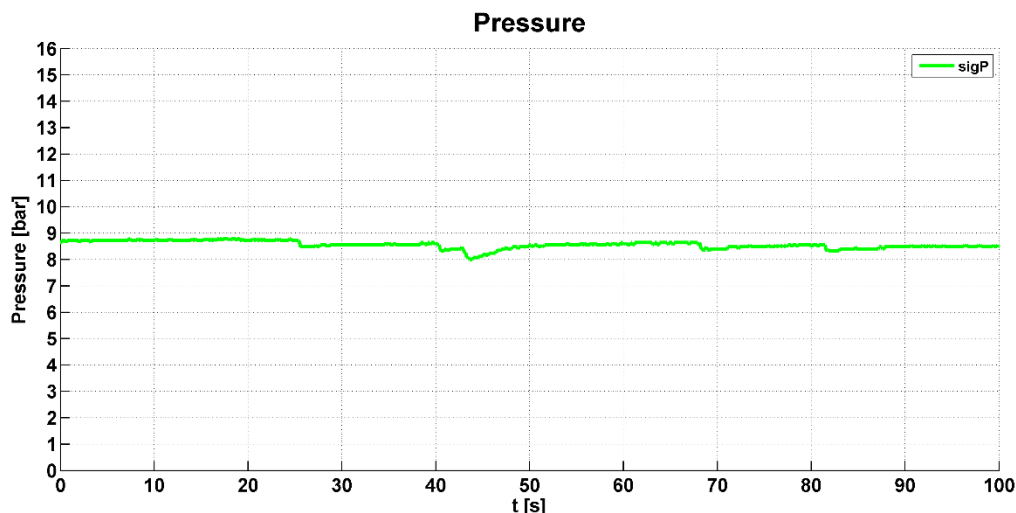
A jak vyplývá z obr. 11, limit pro uzavírání osových diferenciálů pro dané zatočení a rychlost je okolo  $28^\circ$ . Na obr. 72 je zobrazen průběh hnacího momentu motoru, kde bylo kontrolováno, zda nebyl překročen limit hnacího momentu pro možnost aktivace příslušného akčního členu. To je z důvodu předcházení poškození hnacího traktu možným vznikem rázového momentu. V grafu na obr. 73, byl kontrolován průběh tlaku, zda při časté aktivaci a deaktivaci akčních členů nedochází k nadměrnému poklesu tlaku pod stanovenou hodnotu. Tím vlastně bylo kontrolováno, zda je pneumatický okruh správně navržen. Hlavním kontrolním grafem je průběh signálů pro aktivaci/deaktivaci akčních členů a signály koncových spínačů pneumatických válců, zobrazen na obr. 74. Signály pro akční členy jsou zobrazeny černou barvou a pro zpětné vazby modře. Zde je vidět, že během testování byl téměř po celou dobu aktivován pohon všech kol a v některých situacích byly uzavřeny zadní osová diferenciály. Pokud podmínky pro uzavření pominuly, diferenciály byly otevřeny. I z grafu vyplývá, že terén nebyl jeden z nejnáročnějších, jelikož nedocházelo k uzavírání předních osových diferenciálů.



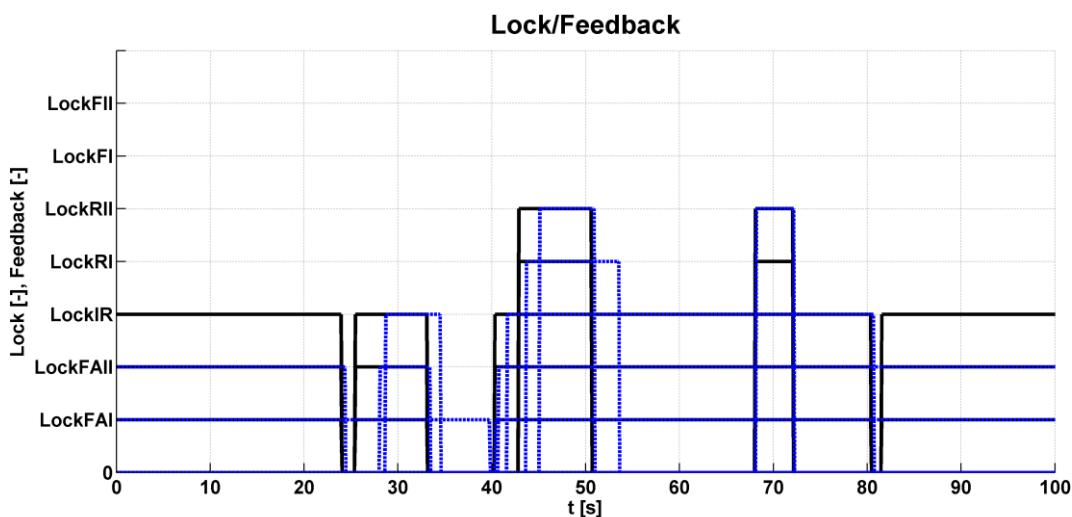
Obr. 71 Průběh signálů zatočení pravého a levého kola přední nápravy



Obr. 72 Průběh signálu hnacího momentu motoru



Obr. 73 Průběh signálu tlaku v pneumatickém okruhu



Obr. 74 Průběh signálů pro akční členy (černá barva) a zpětné signály koncových spínačů (modrá barva)

Jak již bylo několikrát zmíněno i pro toto ověření kompletního navrženého mechatronického systému bylo provedeno velké množství testů, proto je zde zobrazena a jen jednoduše popsána ukázka jednoho z nich. V tomto případě byl sběr dat realizován pomocí naprogramovaného grafického rozhraní zobrazeného na obr. 60 a následně byla data vyhodnocena. Naproti MIL a HIL testování již kontrola správné funkce řízení z dat byla kontrolována jen v situacích, kdy s určitým stavem nebyl spokojen řidič. Ale jelikož vše v dostatečné míře bylo ověřeno v předchozích metodách testování, tak k těmto situacím prakticky nedošlo a řešilo se jen ladění řídicích parametrů. Proto již bylo testování směřováno především na životnost dílčích částí vyvinutého systému. A to bude i další náplní výzkumu tohoto vyvinutého systému.



## ZÁVĚR

Tato práce se zabývala vývojem autonomního mechatronického systému pro uzavírání diferenciálů. Ten je uzpůsoben pro různé typy vozidel od osobních přes nákladní až po různé speciály a traktory. V jednotlivých kapitolách byl popsán vývoj a testování dílčích částí mechatronického systému. Tento systém je sestaven z řídicího algoritmu, prototypu ECU, senzorů, a akčních členů. Základním principem je snímání otáček kol, z nichž je vyhodnocován skluz. Pokud řídicí algoritmus v ECU vyhodnotí aktuální situaci tak, že dochází k prokluzu při jízdě vozidla, udělá příslušné kroky k aktivaci pohonu všech kol. Pokud ani tento zásah nestačí, postupně uzavírá zadní osově diferenciály a následně přední osově diferenciály. Aby bylo toto možné realizovat, je akční člen sestaven z elektroventilu, který se po aktivaci otevře a tím pustí tlakový vzduch do pneumatického válce s konzolou. Tím je vyvozena síla, která spojuje speciálně navržené zubové spojky a tím dochází ke spojování hnacích hřídelů předních náprav (aktivace pohonu všech kol) nebo uzavření příslušného diferenciálu.

V první fázi vývoje byl navrhován řídicí algoritmus, který byl vyvíjen ve dvou formách pro částečně nezávislé a závislé automatické uzavírání diferenciálů. Z testů a simulací vyplynulo, že vhodnější z hlediska zjednodušení a ceny kompletního mechatronického systému bude zvolen jako hlavní přístup ten se závislým uzavíráním diferenciálů a aktivací pohonu všech kol. V příslušné kapitole byl popsán jeho princip fungování, požadované vstupy, výstupy pro akční členy, vlastní programování řídicího algoritmu, nastavení a programování periférií procesoru pro předávání vstupních a výstupních hodnot algoritmu včetně naprogramovaného bootloadru pro zavádění hlavního programu s řídicím algoritmem do paměti procesoru.

První z metod ověřování řídicího algoritmu bylo tzv. MIL testování, kdy bylo využito vlastní knihovny s bloky pro sestavování výpočtových modelů různých vozidel. Každý blok zastupuje matematicky popsanou část vozidla a uživatel může pomocí nich sestavit výpočtový model pro testování jízdní dynamiky, analýzu vibrací a v tomto případě především pro testování mechatronických systémů. Pomocí vlastního nástroje byl vytvořen výpočtový model vozidla, který byl spojen s vyvíjeným řídicím algoritmem, a tato smyčka byla testována. Při tomto testování lze předpokládat objevení nejvíce možných chyb vzniklých např. špatnou volbou principu fungování vyvíjeného systému, nebo jen vlivem překlepu při programování. Dle odhadu během tohoto vývoje a v této fázi testování bylo nalezeno a provedeno asi zhruba 70 % chyb a úprav, proto byl tento postup testování velice významný při vývoji. Pro názornost jsou v práci zobrazeny grafy z jedné ze simulací rozjezdu vozidla v bahnitěm terénu. Z grafu je vidět, jak řídicí algoritmus vyhodnocoval dané situace a dle toho aktivoval pohon všech kol a uzavíral příslušné diferenciály. Tyto a další data ze simulací byla analyzována a podle toho byl řídicí algoritmus upravován.

Druhou metodou nejenom pro ověření řídicího algoritmu, ale i pro ověření senzorů a akčních členů prototypu vozidla bylo použito tzv. HIL testování. Jednalo se o využití hardwaru pro HIL testování, který slouží jako náhrada prototypu ECU a tím mohl být vyvíjený řídicí algoritmus testován, aniž by byl k dispozici prototyp řídicí jednotky. V první fázi tohoto testování byl hardware spojen se senzory a akčními členy v laboratorních podmínkách pro otestování správného zpracovávání signálů ze snímačů a správného ovládání akčních členů. Když byly k dispozici dva prototypy vozidel, byl tento systém aplikován na ně a začalo se testovat v reálném provozu vozidla. Testy probíhaly na různém povrchu vozovky a při náročných povětrnostních podmínkách. Zaznamenávaná data z testování byla vyhodnocována a dle výsledků byly prováděny úpravy mechatronického systému. V této fázi testování bylo dle odhadu odhaleno 28 % chyb z celkového počtu. Především již bylo prováděno testování

mechatronického systému obdobného pro sériovou výrobu. Důležitým vyhodnocením tohoto testování byly také časy reakcí od odeslání signálu ECU po signalizaci koncovým spínačem pneumatického válce o skutečném uzavření zubových spojek. Snahou potom bylo celý systém naladit tak, aby se tyto časy pohybovaly pod 0,1 s. Z hlediska otevírání zubových spojek byl především kladen důraz na stavy, aby nezůstávaly zubové spojky uzavřeny vlivem přenášeného hnacího momentu po jejich deaktivaci. To bylo laděno nastavením tlaku v pneumatickém okruhu a parametry výtlačné pružiny v pneumatickém válci pro jejich rozpojení.

Paralelně probíhal vývoj prototypu ECU. Dle požadovaných vstupů a výstupů byly navrženy elektronické obvody pro zpracování a předávání signálů pro procesor. Jednalo se o elektrické obvody pro zpracování signálů otáček kol, snímače natočení volantu, signálů pedálů vozidla, signálů zpětných vazeb koncových spínačů pneumatických válců, snímače tlaku pneumatického okruhu, CAN a SPI komunikace a obvody pro řízení akčních členů. V souhrnu to byl návrh elektrických obvodů pro zpracování analogových a digitálních vstupů, digitálních výstupů, napájení ECU, jejich periférií atd. Testování ECU probíhalo prvně v laboratorních podmínkách s připojenými snímači a akčními členy a následně na prototypu vozidla. Vývoji prototypu ECU předcházelo testování dílčích obvodů, zda budou v propojení se snímači fungovat a dle toho docházelo k úpravám návrhu prototypu ECU, než byl vyroben. Pro testování byly vyrobeny dva prototypy ECU, kdy druhý prototyp obsahuje již obvody pro vyvíjenou nadstavbu pro predikci a sledování aktuálního okolí vozidla, např. pro detekci jízdy vozidla na nakloněné vozovce. Tím tento nadstavbový systém aktivuje pohon všech kol, aby byl hnací moment rozložen po celém hnacím traktu vozidla a tím nedocházelo nadměrnému namáhání jen některých hnacích hřídelí.

Pro komunikaci s ECU byl naprogramován grafický interface, který uživateli umožňuje zavádět nový hlavní program s algoritmem do paměti procesoru přes CAN komunikaci, aniž by bylo nutné, aby dodavatel musel navštívit zákazníka. Dále byl naprogramován podprogram pro nastavování uživatelských parametrů ECU, kde se především jedná o parametry rozměrů vozidla, parametry snímačů a aktivaci čtení signálů ze zpráv CAN jiných ECU na vozidle. Další podprogram je určen pro diagnostiku ECU, kde jsou zobrazovány diagnostické zprávy podle normy J1939 DM1 – aktivní poruchy a DM2 dříve aktivní poruchy. Tyto poruchy lze z externí paměti ECU vymazat pomocí DM3. Součástí jsou i programy pro výzkum, které slouží pro zobrazování aktuálních hodnot uvnitř ECU, zaznamenávání dat, testování displeje, testování omezování hnacího momentu a otáček motoru, testování CAN komunikace atd.

Aby byl řidič informován o stavu aktivace pohonu všech kol nebo uzavření příslušných diferenciálů, byl naprogramován a popsán v této práci dotykový displej, který tuto funkci umožňuje. Zároveň pomocí něho je uživatel schopen nastavit jízdní módy ECU. To znamená nastavení mechatronického systému pro jízdu po silnici, polní cestou nebo pro terén/sníh.

Poslední z hlavních kapitol popisuje ukázkou finálního testování kompletního vyvinutého mechatronického systému odpovídající verzi pro sériovou výrobu. Při těchto testech byla také zaznamenávána data a po jejich zpracování byly kontrolovány stavy, které se pro řidiče zdály být nějakým způsobem neadekvátní k požadavkům mechatronického systému. V této fázi lze odhadnout nalezení 2 % chyb nebo spíše to obsahuje jenom doladování parametrů řízení. To je již zanedbatelné množství, protože využitím MIL a HIL testování byl hlavní rozsah chyb vzniklých při vývoji redukován na minimální množství. Tím byl ušetřen čas, finance a případné poškození prototypu vozidla nebo nebezpečí úrazu posádky vozidla.

Obecným shrnutím této práce je, že byl vyvinut autonomní mechatronický systém pro uzavírání diferenciálů včetně jeho otestování na prototypu vozidla krom návrhu speciálních zubových spojek. Z testů a reakcí zákazníka vyplývá, že systém funguje správně a je přínosem pro vozidla, aby nedocházelo k jejich uvíznutí v nepříznivých terénních podmínkách a zároveň byl chráněn hnací trakt před neodborným používáním uzavírání diferenciálů nezkušenými řidiči. Jak již bylo zmíněno v rámci projektu TE 01020020 a ve spolupráci se společností Tatra Truck a.s. byl samotný řídicí algoritmus i uzpůsoben pro jejich vozidla, kde ho implementovali do jejich řídicí jednotky. V rámci obecného výzkumu tohoto mechatronického systému bylo navrženo vlastní řešení ECU, modulárního řídicího algoritmu, softwaru pro komunikaci s ECU a provedeno testování popsané v této práci.

Pokračováním vývoje bude implementace na další typy vozidel a spolupráce s dalšími zákazníky tohoto systému. Předpokladem je rozšiřování nadstavby pro predikci okolí vozidla, aby byl lépe využit hnací moment. Dále se rozvíjí metoda pro testování ECU pomocí HIL tak, že pomocí výpočtových modelů vozidla jsou generovány signály nahrazující reálné snímače a tím sledována reakce prototypu ECU s řídicím algoritmem. Z prototypu ECU jsou odesílány signály akčním členům ve formě výpočtového modelu a tím je možné v reálném čase testovat tento systém a prototyp ECU se zásahy reálného řidiče bez nároků na výrobu prototypu vozidla. Dalším předpokládaným vývojem bude minimalizace rozměrů prototypu ECU. Posledním předpokládaným dalším postupem vývoje bude využití tzv. neuronových sítí, aby se řídicí algoritmus mohl během užívání dále učit a přizpůsobovat vozidlu a jízdním situacím.

## POUŽITÉ INFORMAČNÍ ZDROJE

- [1] KERNIGHAN, Brian W a Dennis M RITCHIE. *Programovací jazyk C*. 1. Brno: Computer Press, 2006. ISBN 80-251-0897-X.
- [2] LIBERTY, Jesse. *Naučte se C za 21 dní*. 2., aktualiz. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1583-1.
- [3] Simulink. In: *MathWorks* [online]. United States: MathWorks, 2018 [cit. 2018-09-25]. Dostupné z: <https://www.mathworks.com/products/simulink.html>
- [4] *ARM Keil* [online]. Cambridge, UK: Arm Limited, 2018 [cit. 2018-09-25]. Dostupné z: <http://www.keil.com/>
- [5] IAR Embedded Workbench. In: *IAR Embedded Workbench* [online]. Uppsala, Sweden: IAR Systems, 2018 [cit. 2018-09-25]. Dostupné z: <https://www.iar.com/iar-embedded-workbench/>
- [6] *AdaCore* [online]. New York, USA: AdaCore, 2018 [cit. 2018-09-25]. Dostupné z: <https://www.adacore.com/>
- [7] Allegro PCB Designer. *Www.cadence.com* [online]. Cadence Design Systems, 2018 [cit. 2018-09-25]. Dostupné z: [https://www.cadence.com/content/cadence-www/global/en\\_US/home/tools/pcb-design-and-analysis/pcb-layout/allegro-pcb-designer.html](https://www.cadence.com/content/cadence-www/global/en_US/home/tools/pcb-design-and-analysis/pcb-layout/allegro-pcb-designer.html)
- [8] *EasyEDA* [online]. China: EasyEDA, 2018 [cit. 2018-09-25]. Dostupné z: <https://easyeda.com/>
- [9] EAGLE. In: *Autodesk* [online]. Kalifornie, USA: Autodesk, 2018 [cit. 2018-09-25]. Dostupné z: <https://www.autodesk.com/products/eagle/overview>
- [10] OrCAD PCB Designer. In: *OrCAD* [online]. Hillsboro, Oregon: OrCAD, 2018 [cit. 2018-09-25]. Dostupné z: <https://www.orcad.com/products/orcad-pcb-designer/overview>
- [11] Multisim. In: *National Instruments* [online]. Austin, Texas: National Instruments, 2018 [cit. 2018-09-25]. Dostupné z: <http://www.ni.com/cs-cz/shop/electronic-test-instrumentation/application-software-for-electronic-test-and-instrumentation-category/what-is-multisim.html>
- [12] FlexECU. In: *Mathworks* [online]. United States: The MathWorks, 2018 [cit. 2018-09-25]. Dostupné z: [https://www.mathworks.com/products/connections/product\\_detail/flexecu.html](https://www.mathworks.com/products/connections/product_detail/flexecu.html)
- [13] PRODUCTS: Discover OpenECU. In: *Pi-Innovo* [online]. Plymouth, MI 48170-3765 United States of America: Pi Innovo, 2018 [cit. 2018-09-25]. Dostupné z: [www.pi-innovo.com](http://www.pi-innovo.com)
- [14] Bosch Motorsport - Overview. In: *Bosch Motorsport* [online]. Německo: Bosch Engineering, 2018 [cit. 2018-09-25]. Dostupné z: <http://www.bosch-motorsport.de/content/downloads/Products/9007213411107467.html>

- [15] Rapid development, RT simulace, modelování. In: *REX Controls* [online]. Plzeň: REX Controls, 2018 [cit. 2018-09-25]. Dostupné z: <https://www.rexcontrols.cz/rapid-development-rt-simulace-modelovani>
- [16] *DSPACE* [online]. Paderborn Germany: dSpace, 2018 [cit. 2018-09-25]. Dostupné z: <https://www.dspace.com/en/pub/home/products/systems/ecutest.cfm>
- [17] NovaCarts HiL Simulation. In: *MicroNova* [online]. Vierkirchen: MicroNova, 2018 [cit. 2018-09-25]. Dostupné z: <https://www.micronova.de/en/testing/novacarts-hil-simulation.html>
- [18] Modular Test Hardware: VT System. In: *Vector* [online]. Stuttgart, Germany: Vector, 2018 [cit. 2018-09-25]. Dostupné z: [https://vector.com/vi\\_vt-system\\_en.html](https://vector.com/vi_vt-system_en.html)
- [19] Hardware-in-the-Loop Testing (HIL) for Real-Time Plant Simulation. In: *Speedgoat* [online]. Liebfeld Switzerland: Speedgoat, 2018 [cit. 2018-09-25]. Dostupné z: [https://www.speedgoat.com/applications-industries/applications/plant-simulation-hil?gclid=EAIaIQobChMI7uimsjV3QIV1eJ3Ch2G4AN6EAAAYASAAEgJCqPD\\_BwE](https://www.speedgoat.com/applications-industries/applications/plant-simulation-hil?gclid=EAIaIQobChMI7uimsjV3QIV1eJ3Ch2G4AN6EAAAYASAAEgJCqPD_BwE)
- [20] Hardware-in-the-Loop (HIL) Simulation. In: *National Instruments* [online]. Austin, Texas: National Instruments, 2018 [cit. 2018-09-25]. Dostupné z: <http://www.ni.com/cs-cz/innovations/automotive/hardware-in-the-loop.html>
- [21] Transportation. In: *US Micro Products* [online]. Austin: US Micro Products, 2016 [cit. 2018-09-25]. Dostupné z: <http://www.usmicroproducts.com/applications/transportation>
- [22] OPUS Control Panels. In: *Topcon* [online]. Geisenheim, Germany: Topcon Electronics, 2018 [cit. 2018-09-25]. Dostupné z: <https://topcon-electronics.de/en/products/overview.html>
- [23] Continental's 3D Touch Surface Display Receives Highest Honor at CES 2018 Innovation Awards. In: *Continental AG* [online]. Hannover, Germany: Continental, 2018 [cit. 2018-09-25]. Dostupné z: <https://www.continental-corporation.com/en/press/press-releases/2018-01-04-3d-touch-display-118076>
- [24] INCA Base Product. In: *ETAS* [online]. Stuttgart Germany: ETAS, 2018 [cit. 2018-09-25]. Dostupné z: <https://www.etas.com/en/products/inca.php>
- [25] Standards. In: *ASAM* [online]. Hoehenkirchen, Germany: ASAM e.V., 2018 [cit. 2018-09-25]. Dostupné z: <https://www.asam.net/standards/>
- [26] STELZENEDER, H., Franz X. a H. AITZETMÜLLER. ADM A NEW DRIVE-TRAIN MANAGEMENT. In: *FISITA world automotive congress; Seoul 2000 FISITA world automotive congress, automotive innovation for the new millennium*. Korean: Korean Society of Automotive Engineers, 2000, s. 1-7. ISBN 8985000004.
- [27] ADM 2. In: *Products for Trucks* [online]. Friedrichshafen Germany: ZF Friedrichshafen, 2018 [cit. 2018-09-11]. Dostupné z: [https://www.zf.com/products/en/trucks/products\\_29087.html](https://www.zf.com/products/en/trucks/products_29087.html)

- [28] *Meritor* [online]. United States: Meritor, 2018 [cit. 2018-09-25]. Dostupné z: <https://www.meritor.com/>
- [29] STM32F417VG. In: *STMicroelectronics* [online]. Geneva, Switzerland: STMicroelectronics, 2018 [cit. 2018-09-11]. Dostupné z: <https://www.st.com/en/microcontrollers/stm32f417vg.html>
- [30] STM32CubeMX. In: *STMicroelectronics* [online]. Geneva, Switzerland: STMicroelectronics, 2018 [cit. 2018-09-11]. Dostupné z: <https://www.st.com/en/development-tools/stm32cubemx.html>
- [31] TrueSTUDIO. In: *Atollic* [online]. Jönköping, Sweden: Atollic, 2018 [cit. 2018-09-11]. Dostupné z: <https://atollic.com/truestudio/>
- [32] STM32CubeF4. In: *STMicroelectronics* [online]. Geneva, Switzerland: STMicroelectronics, 2018 [cit. 2018-09-11]. Dostupné z: <https://www.st.com/en/embedded-software/stm32cubef4.html>
- [33] *J1939 - Recommended Practice for a Serial Control and Communications Vehicle Network*. 1. United States: SAE International, 2005.
- [34] *J1939-21 - DATA LINK LAYER*. 1. United States: SAE International, 2001.
- [35] *J1939-31 - Network Layer*. 1. United States: SAE International, 2004.
- [36] *J1939-71 - Vehicle Application Layer*. 1. United States: SAE International, 2008.
- [37] *J1939-73 - Application Layer - Diagnostics*. 1. United States: SAE International, 2006.
- [38] *J1939-81 - NETWORK MANAGEMENT*. 1. United States: SAE International, 2003.
- [39] ŠTĚRBA, Pavel. *Elektronika a elektrotechnika motorových vozidel: Seřizování, diagnostika závad a chybové kódy OBD*. Brno: CPress, 2013. ISBN 978-80-264-0271-8.
- [40] BAUER, František. *Traktory a jejich využití*. 2. vyd. Praha: Profi Press, 2013. ISBN 978-80-86726-52-6.
- [41] SELECKÝ, Matuš. *Arduino: uživatelská příručka*. Brno: Computer Press, 2016. ISBN 978-80-251-4840-2.
- [42] VÁŇA, Vladimír. *ARM pro začátečníky*. 1. Praha: BEN - technická literatura, 2009. ISBN 978-80-7300-246-6.
- [43] IMU Data Fusing: Complementary, Kalman, and Mahony Filter. In: *OlliW's Bastelseiten* [online]. Germany: OlliW's Bastelseiten, 2016, 2013 [cit. 2018-09-11]. Dostupné z: <http://www.olliw.eu/2013/imu-data-fusing/#chapter42>
- [44] ST-LINK/V2. *STMicroelectronics* [online]. Geneva, Switzerland: STMicroelectronics, 2018 [cit. 2018-09-11]. Dostupné z: <https://www.st.com/en/development-tools/st-link-v2.html>

- [45] J-Link Debug Probes. In: *SEGGER Microcontroller GmbH* [online]. Monheim am Rhein, Germany: SEGGER Microcontroller, 2018 [cit. 2018-09-11]. Dostupné z: <https://www.segger.com/products/debug-probes/j-link/>
- [46] PCAN-USB: CAN Interface for USB. In: *PEAK-System Technik GmbH* [online]. Darmstadt, Germany: PEAK-System Technik, 2018 [cit. 2018-09-11]. Dostupné z: <https://www.peak-system.com/PCAN-USB.199.0.html?L=1>
- [47] KUČERA, Pavel. *Mechatronický přístup v dynamice vozidel*. Brno, 2015. Dizertační práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství. Vedoucí práce Prof. Ing. Václav Píštěk, DrSc.
- [48] KUČERA, P.; PÍŠTĚK, V. A COMPUTATIONAL ENGINE MODEL IN SIMULINK SOFTWARE. In *KOKA 2016*. Brno: 2016. s. 109-112. ISBN: 978-80-214-5379- 1.
- [49] KUČERA, P.; PÍŠTĚK, V. A Computational Model of ATV Vehicle for Real- Time Simulation. In *Engineering Mechanics 2015. Applied Mechanics and Materials*. 2016. s. 242-247. ISBN: 978-3-03835-700- 1. ISSN: 1662- 7482.
- [50] KUČERA, P.; PÍŠTĚK, V. PODELNÁ A PŘÍČNÁ DYNAMIKA NÁKLADNÍHO VOZIDLA S POHONEM 6x6. *Perners' Contacts*, 2015, roč. X, č. 3, s. 81-86. ISSN: 1801-674X.
- [51] KUČERA, P.; PÍŠTĚK, V. Longitudinal and lateral dynamics of a commercial vehicle in Simulink software. In *Proceeding of International Conference Transport Means 2015. Transport Means*. Kaunas Univ Technol, Kaunas, LITHUANIA: KAUNAS UNIV TECHNOLOGY PRESS, K DONELAICIO 73, KAUNAS LT 3006, LITHUANIA, 2015. s. 458-461. ISBN: 9955-09-935- 6. ISSN: 1822-296X.
- [52] KUČERA, P.; PÍŠTĚK, V. Virtual prototype of a heavy duty off- road truck driveline in Simulink software. In *Proceeding of International Conference Transport Means 2014. Transport Means*. K. Donelaičio st. 73, LT- 44029 Kaunas: Kaunas University of Technology, 2014. s. 5-8. ISBN: 9955-09-935- 6. ISSN: 1822- 296X.
- [53] KUČERA, P.; PÍŠTĚK, V. A Computational Model of Powertrain Components in Simulink. In *Engineering Mechanics 2014. Engineering mechanics 2014*. 1. Brno University of Technology, 2014. s. 344-347. ISBN: 978-80-214-4871- 1. ISSN: 1805- 8248.
- [54] KUČERA, P.; PÍŠTĚK, V. SIMULINK BLOCK LIBRARY FOR ASSEMBLING A VEHICLE. *Perners' Contacts*, 2014, roč. IX, č. 2, s. 38-47. ISSN: 1801- 674X.
- [55] KUČERA, P.; PÍŠTĚK, V. VIRTUAL DIESEL ENGINE IN SIMULINK. *Perners' Contacts*, 2013, roč. VIII, č. 2, s. 95-105. ISSN: 1801- 674X.
- [56] KUČERA, P.; PÍŠTĚK, V. TRANSMISSION COMPUTATIONAL MODEL IN SIMULINK. *Perners' Contacts*, 2013, roč. VIII, č. 4, s. 37-48. ISSN: 1801- 674X
- [57] KARNOPP, Dean. Computer Simulation of Stick-Slip Friction in Mechanical Dynamic Systems. *Journal of Dynamic Systems, Measurement, and Control*. 1985, 107(1). DOI: 10.1115/1.3140698. ISSN 00220434. Dostupné také z: <http://DynamicSystems.asmedigitalcollection.asme.org/article.aspx?articleid=1403660>

- [58] PACEJKA, Hans B. *Tyre and Vehicle Dynamics*. 2. vydání. Oxford: Butterworth-Heinemann, 2006. ISBN 0-7506-6918-7.
- [59] KUČERA, P.; PÍŠTĚK, V. Mechatronic System of Automatic and Manual Differential Lock Control - Vehicle Turnig. In Proceedings of the 2016 17th International Conference on Mechatronics – Mechatronika (ME) 2016. 1st edition. Prague: Czech Technical University in Prague, 2016. s. 307-310. ISBN: 978-80-01-05882- 4.
- [60] KUČERA, P.; PÍŠTĚK, V. Automatic and Manual Differential Lock Control in a Truck – Cornering. In 20th International Conference Transport Means 2016 Proceedings Part I. Transport Means. Kaunas University of Technology, Kaunas, LITHUANIA: KAUNAS UNIV TECHNOLOGY PRESS, K DONELAICIO 73, KAUNAS LT 3006, LITHUANIA, 2016. s. 98-101. ISSN: 1822-296X.
- [61] KUČERA, P.; PÍŠTĚK, V.; PORTEŠ, P. AUTOMATIC DIFFERENTIAL LOCK CONTROL IN A TRUCK – STRAIGHT DRIVE. In Engineering Mechanics 2016. 1. Prague: Institute of Thermomechanics Academy of Sciences of the Czech Republic, v.v.i., Prague, 2016. s. 334-337. ISBN: 978-80-87012-59- 8
- [62] KUČERA, P.; PÍŠTĚK, V.; PORTEŠ, P. AUTOMATICKÉ ŘÍZENÍ UZAVÍRÁNÍ DIFERENCIÁLU NÁKLADNÍHO VOZIDLA PŘI JÍZDĚ PŘÍMÝM SMĚREM. Pernal's Contacts, 2016, roč. XI, č. 1, s. 115-121. ISSN: 1801-674X.
- [63] DABNEY, James a Thomas L HARMAN. *Mastering Simulink*. Upper Saddle River: Pearson Prentice Hall, 2004, 376 s. ISBN 0-13-142477-7.
- [64] GREPL, Robert. *Kinematika a dynamika mechatronických systémů*. Brno: CERM - Akademické nakladatelství, 2007. ISBN 978-80214-3530-8.
- [65] BUDYNAS, Richard G. a J. Keith. NISBETT. *Shigley's mechanical engineering design*. Tenth edition. New York: McGraw-Hill Education, 2015. ISBN 00-733-9820-9.
- [66] Adams/Tire help - MSC Adams 2011. In: *MSC Software* [online]. Newport Beach, CA: MSC Software, 2018, 2014 [cit. 2018-09-11]. Dostupné z: <https://simcompanion.mscsoftware.com/infocenter/index?page=content&id=DOC9837&cat=1VMO50&actp=LIST>
- [67] *National Instruments* [online]. Austin, Texas: National Instruments, 2018 [cit. 2018-09-11]. Dostupné z: <http://www.ni.com/cs-cz.html>
- [68] VeriStand. In: *National Instruments* [online]. Austin, Texas: National Instruments, 2018 [cit. 2018-09-11]. Dostupné z: <http://www.ni.com/veristand/>
- [69] VLACH, Jaroslav, Josef HAVLÍČEK a Martin VLACH. *Začínáme s LabVIEW*. Praha: BEN - technická literatura, 2008. ISBN 978-80-7300-245-9.
- [70] PORTEŠ, P.; KUČERA, P.; PÍŠTĚK, V.; FOJTÁŠEK, J.; ZHÁŇAL, L. MODERN TOOLS FOR VEHICLE DEVELOPMENT. In *Engineering Mechanics 2017*. 1. 2017. s. 54-57. ISBN: 978-80-214-5497- 2.



- [71] KUČERA, P.; PÍŠTĚK, V. Using HIL testing for Differential Lock Control. In *Transport Means 2017 Proceedings. Transport Means*. 1. Kaunas University of Technology, Kaunas, LITHUANIA: 2017. s. 640-643. ISSN: 1822-296X.
- [72] KUČERA, P.; PÍŠTĚK, V. Testing of the Mechatronic Robotic System of the Differential Lock Control on a Truck. *International Journal of Advanced Robotic Systems*, 2017, roč. 14, č. 5, s. 1-7. ISSN: 1729-8814.
- [73] *KiCad EDA* [online]. Vesoul, France: KiCad EDA, 2018 [cit. 2018-09-11]. Dostupné z: <http://kicad-pcb.org/>
- [74] JAN, Zdeněk, Bronislav ŽDÁNSKÝ a Jindřich KUBÁT. *Automobily. 5, Elektrotechnika motorových vozidel I*. 3. vydání. Brno: Avid, 2012. ISBN 978-80-87143-22-3.
- [75] JAN, Zdeněk, Jindřich KUBÁT a Bronislav ŽDÁNSKÝ. *Automobily. 6, Elektrotechnika motorových vozidel I*. 3. Brno: Avid, 2013. ISBN 978-80-87143-27-8.
- [76] ZÁHLAVA, Vít. *Metodika návrhu plošných spojů*. 1. Praha: Vydavatelství ČVUT, 2000. ISBN 80-010-2193-9.
- [77] ŠANDERA, Josef. *Návrh plošných spojů pro povrchovou montáž*. 1. Praha: BEN - technická literatura, 2006. ISBN 80-730-0181-0.
- [78] DOLEČEK, Jaroslav. *Moderní učebnice elektroniky - 1. díl Základy elektroniky, ideální a reálné prvky: rezistor, kondenzátor, cívka*. 1. Praha: BEN - technická literatura, 2005. ISBN 80-730-0146-2.
- [79] DOLEČEK, Jaroslav. *Moderní učebnice elektroniky - 2. díl Polovodičové prvky a elektronky*. 1. Praha: BEN - technická literatura, 2005. ISBN 80-730-0161-6.
- [80] Mbed LPC1768. In: *Arm MBED* [online]. Cambridge, UK: Arm Limited, 2018 [cit. 2018-09-11]. Dostupné z: <https://os.mbed.com/platforms/mbed-LPC1768/>
- [81] SIMetrix. In: *SIMetrix Technologies Ltd* [online]. Berkshire, United Kingdom: SIMetrix Technologies, 2018 [cit. 2018-09-11]. Dostupné z: <https://www.simetrix.co.uk/products/simetrix.html>
- [82] BSP752R. *Infineon Technologies AG* [online]. Munich, Germany: Infineon Technologies, 2018 [cit. 2018-09-11]. Dostupné z: <https://www.infineon.com/cms/en/product/power/smart-low-side-high-side-switches/automotive-smart-high-side-switch-profet/bsp752r/>
- [83] *WxWidgets* [online]. United kingdom: wxWidgets, 2018 [cit. 2018-09-11]. Dostupné z: <https://www.wxwidgets.org/>

## SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

$\omega$	[rad s <sup>-1</sup> ]	Úhlová rychlost (Výpočtové modely)
$\varphi$	[rad]	Natočení (Výpočtové modely)
<i>A</i>		Označení nápravy – užito při výpočtech rozdílů otáček
<i>ABS</i>	[-]	Ve schématu řídicího algoritmu je to hodnota aktivace ABS
<i>ABS</i>		Anti-lock Braking System
<i>AD</i>		Analog-digitál
<i>APB1</i>	[MHz]	Frekvence periferních hodin procesoru.
<i>APB2</i>	[MHz]	Frekvence periferních hodin procesoru.
<i>B</i>	[-]	Hodnota brzdového pedálu
<i>BitRat</i>	[kBit s <sup>-1</sup> ]	Přenosová rychlost CAN sběrnice
<i>BS1</i>	[-]	Time Quanta in Bit Segment 1
<i>BS2</i>	[-]	Time Quanta in Bit Segment 2
<i>C</i>	[-]	Hodnota spojkového pedálu
<i>c</i>	[Nm rad <sup>-1</sup> ]	Torzni tuhost hřídele (Výpočtové modely)
<i>CAN</i>		CAN zpráva Controller Area Network
<i>CANH</i>		High-Level Voltage vstup/výstup
<i>CANL</i>		Low-Level Voltage vstup/výstup
<i>CCVS</i>		CAN zpráva Cruise Control/Vehicle Speed
<i>CloPre</i>	[-]	Clock Prescale
<i>CM</i>		SPN Conversion Method
<i>COUT</i>	[V]	Výstup komparátoru elektronické součástky MAX9926
<i>d</i>	[min <sup>-1</sup> ]	Hodnota rozdílů otáček hřídelí
<i>DCDL</i>		Driver-Controlled Differential Lock
<i>Delta</i>	[°]	Zatočení příslušného kola
<i>DM1</i>		Diagnostic Message 1, Active Diagnostic Trouble Codes
<i>DM11</i>		Diagnostic Message 11, Diagnostic Data Clear/Reset for Active DTCs
<i>DM2</i>		Diagnostic Message 2, Previously Active Diagnostic Trouble Codes
<i>DM3</i>		Diagnostic Message 3, Diagnostic Data Clear/Reset for Previously Active DTCs
<i>do</i>	[min <sup>-1</sup> ]	Hodnota rozdílů otáček hřídelí
<i>DPS</i>		Deska plošných spojů
<i>DTC</i>		Diagnostic Trouble Code

<i>E</i>	[-]	Error - Porucha systému
<i>EBC1</i>		CAN zpráva Electronic Brake Controller 1
<i>EBC2</i>		CAN zpráva Electronic Brake Controller 2 - Wheel Speed Information
<i>EBS</i>		Electronic Braking System
<i>EEC</i>		CAN zpráva Electronic Engine Controller 1
<i>EEC2</i>		CAN zpráva Electronic Engine Controller 2
<i>EMC</i>		Elektromagnetická kompatibilita
<i>ERC1</i>		CAN zpráva Electronic Retarder Controller 1
<i>F</i>	[-]	Zpětná vazba informující o stavu pohonu všech kol a diferenciálů
<i>F</i>		Označení – přední (vyplývá z anglických názvů front)
<i>Feedback</i>		Hodnota příslušného koncového spínače pneumatického válce
<i>FMI</i>		Failure Mode Identifier
<i>FPGA</i>		Field Programmable Gate Array – hradlové pole
<i>FroAxlSpd</i>	[km h <sup>-1</sup> ]	Rychlosti přední nápravy
<i>GND</i>		Elektrická zem
<i>GUI</i>		Graphical User Interface
<i>i</i>	[-]	Index označení (hodnota dle příslušné rovnice)
<i>I</i>		Pořadí nápravy (I – první, II – druhá, III - třetí)
<i>I2C</i>		Inter-Integrated Circuit
<i>ID</i>		Identifikátor CAN zprávy
<i>IN-</i>	[V]	Invertní vstup elektronické součástky MAX9926
<i>IN+</i>	[V]	Neinvertní vstup elektronické součástky MAX9926
<i>IR</i>		Označení pro zadní mezinápravový diferenciál
<i>JTAG</i>		Joint Test Action Group
<i>K</i>	[-]	Hodnota ovládacích přepínačů a spínače pro ECU
<i>k</i>	[Nm s rad <sup>-1</sup> ]	Vnitřní tlumení hřídele (Výpočtové modely)
<i>L</i>		Označení – levá (vyplývá z anglických názvů)
<i>LockF</i>	[-]	Uzavření předních diferenciálů
<i>LockFA</i>	[-]	Aktivace pohonu všech kol/předních náprav
<i>LockIR</i>	[-]	Uzavření mezinápravového diferenciálu
<i>LockR</i>	[-]	Uzavření zadního diferenciálu
<i>M</i>	[-]	Mód nastavení (silnice, polní cesta, terén/sníh)

<i>MIL</i>		Model in the Loop
<i>o</i>	[min <sup>-1</sup> ]	Otáčky kol/hřídelí dle polohy ve schématu řídicího algoritmu
<i>omg</i>	[min <sup>-1</sup> ]	Otáčky kola
<i>OUT</i>	[V]	Výstupní napětí pro zátěž
<i>P</i>	[bar]	Tlak v pneumatickém okruhu
<i>PCLK1</i>	[MHz]	Frekvence periferních hodin procesoru.
<i>PCLK2</i>	[MHz]	Frekvence periferních hodin procesoru.
<i>Period</i>	[μs]	Perioda natočení ozubeného kola pro indukční snímač o jeden zub
<i>PIL</i>		Processor in the Loo
<i>PreQua</i>	[-]	Prescaler pro čas kvanta
<i>PreSPI</i>	[-]	Prescaler pro stanovení přenosové rychlosti SPI komunikace
<i>R</i>		Označení – zadní (vyplývá z anglických názvů)
<i>R</i>		Označení – pravá (vyplývá z anglických názvů)
<i>RelSpe</i>	[km h <sup>-1</sup> ]	Referenčních rychlostí kol
<i>RXD</i>		Receive Data Output
<i>s</i>	[-]	Skluz mezi jednotlivými koly nebo hřídelemi
<i>SamTim<sub>2</sub></i>	[MHz <sup>-1</sup> ]	Sampling Time analogového vstupu
<i>sigB</i>	[-]	Hodnota signálu brzdového pedálu
<i>sigC</i>	[-]	Hodnota spojkového pedálu
<i>sigP</i>	[bar]	Hodnota tlaku v pneumatickém okruhu
<i>sigT</i>	[-]	Hodnota signálu plynového pedálu
<i>SIL</i>		Software in the Loop
<i>SMD</i>		Surface mount device
<i>SPI</i>		Serial Peripheral Interface
<i>SPN</i>		Suspect Parameter Number
<i>St</i>	[°]	Úhel natočení předních kol
<i>T</i>	[-]	Hodnota plynového pedálu
<i>T</i>	[Nm]	Hnací moment (Výpočtové modely)
<i>tDO</i>	[s]	Čas, po který řídicí algoritmus čeká, než může sestoupit ve schématu
<i>Te</i>	[%]	Procentuální hodnota hnacího momentu motoru
<i>TimBit</i>	[s]	Čas trvání jednoho bitu na CAN sběrnici
<i>TimQua</i>	[s]	Čas trvání jednoho kvanta

<i>TSCI</i>		CAN zpráva Torque/Speed Control 1
<i>tT</i>	[s]	Čas testování, po kterém řídicí algoritmus otevře přední diferenciály
<i>TTL</i>		Transistor-transistor-logika
<i>tUP</i>	[s]	Čas, po který řídicí algoritmus čeká, než může vystoupit ve schématu
<i>Tv</i>	[-]	Typ vozidla
<i>TXD</i>		Transmit Data Input
<i>UnlockF</i>	[-]	Otevření předních diferenciálů
<i>UnlockFA</i>	[-]	Deaktivace pohonu všech kol/předních náprav
<i>UnlockIR</i>	[-]	Otevření mezinápravového diferenciálu
<i>UnlockR</i>	[-]	Otevření zadního diferenciálu
<i>v</i>	[m/s]	Rychlost vozidla
<i>V_IN</i>	[V]	Označení vstupu elektrického obvodu
<i>V_OUT</i>	[V]	Výstupní napětí digitálního signálu
<i>Vbb</i>	[V]	Napájecí napětí
<i>vc</i>	[km/h]	Rychlost vozidla
<i>z</i>	[-]	Počet zubů na jednu otáčku kola vozidla
<i>ZF ADM</i>		ZF A New Drive-Train Management

## **SEZNAM PŘÍLOH**

P1 – Tabulka s nastavitelnými parametry autonomního systému uživatelem

## P1 – Tabulka s nastavitelnými parametry autonomního systému uživatelem

	Jednotka	Popis
Type	-	Typ vozidla (4x4, 6x6, 8x8)
z	-	Počet zubů pro snímač otáček
cirF	mm	Valivý obvod přední pneumatiky
cirR	mm	Valivý obvod zadní pneumatiky
setS	V	Nastavení napětí odpovídajícímu při přímé jízdě - AS -Angle sensor
inap	-	Převodový poměr nápravy
WBI	m	Rozvor (první mezera mezi nápravami)
WBII	m	Rozvor (druhá mezera mezi nápravami)
WBIII	m	Rozvor (třetí mezera mezi nápravami)
WBIV	m	rezervováno
tf	m	Rozchod kol předních náprav - FA - Front axle
tr	m	Rozchod kol zadních náprav - RA - Rear axle
tU[11]	V	Tabulka hodnot napětí ze snímače natočení kol - AS -Angle sensor
tDeltaFL[11]	°	Tabulka hodnot natočení levého kola první přední nápravy - FL - front left
tDeltaFR[11]	°	Tabulka hodnot natočení pravého kola první přední nápravy - FR - Front right
tUP[10]	V	Tabulka hodnot napětí ze snímače tlaku - PS - Pressure sensor
tSP[10]	bar	Tabulka hodnot tlaku ze snímače tlaku - PS - Pressure sensor
AvaCAN_ActEngPerTor	-	Aktivace snímání signálu hnacího momentu motoru přes CAN
AvaCAN_AccPedPos1	-	Aktivace snímání signálu plynového pedálu přes CAN
AvaCAN_AntLocBraAct	-	Aktivace snímání signálu aktivního ABS přes CAN
AvaCAN_ParBraSwi	-	Aktivace snímání signálu parkovací brzdy přes CAN
AvaCAN_BraSwi	-	Aktivace snímání signálu brzdového pedálu přes CAN
AvaCAN_CluSwi	-	Aktivace snímání signálu spojkového pedálu přes CAN
AvaCAN_RefTorMod_e	-	Aktivace snímání signálu motorové brzdy přes CAN
AvaCAN_RefTorMod	-	Aktivace snímání signálu retardéru přes CAN
AvaCAN_WheSpeInf	-	Aktivace snímání signálu rychlosti kol přes CAN
CAN_NumAxlWheSpe_F	-	Poloha přední nápravy, ze které je snímán signál rychlosti kol přes CAN
CAN_NumAxlWheSpe_R	-	Poloha zadní nápravy, ze které je snímán signál rychlosti kol přes CAN
AvaCAN_TSC1	-	Aktivace odesílání CAN zprávy TSC1 pro řízení otáček a hnacího momentu motoru
Gyro_axis1	-	Nastavení první osy, okolo které je natočen MEMS modul
Gyro_axis2	-	Nastavení druhé osy, okolo které je natočen MEMS modul
Gyro_axis3	-	Nastavení třetí osy, okolo které je natočen MEMS modul
Gyro_ang1	°	Nastavení úhlu natočení MEMS modul okolo první osy
Gyro_ang2	°	Nastavení úhlu natočení MEMS modul okolo druhé osy
Gyro_ang3	°	Nastavení úhlu natočení MEMS modul okolo třetí osy