

Západočeská univerzita v Plzni

Fakulta elektrotechnická

Katedra aplikované elektroniky a telekomunikací



DISERTAČNÍ PRÁCE

Plzeň 2011

Ing. Michal Kubík

Západočeská univerzita v Plzni

Fakulta elektrotechnická

Katedra aplikované elektroniky a telekomunikací



DISERTAČNÍ PRÁCE

**k získání akademického titulu doktor
v oboru**

ELEKTRONIKA

Ing. Michal Kubík

**Testování elektronických systémů
automobilu**

Školitel: Prof. Ing. Jiří Pinker, CSc.

Datum vykonání státní doktorské zkoušky: 26. 4. 2006

Datum odevzdání práce:

Plzeň 2011

Prohlášení

Tato práce byla napsána s využitím materiálů či zdrojů uvedených v kapitole 9. Případně byly informace získány na veřejném Internetu. Některé výrazy uvedené v této práci mohou být ochrannými známkami nebo registrovanými ochrannými známkami jejich případných vlastníků a uvedením těchto výrazů nejsou zpochybněna jejich vlastnická práva.

Klíčová slova

ECU, HiL, Testování, Automobilová technika, Elektronické systémy, Modelování, Komfortní systémy, Automatická klimatizace, Centrální zamykání, HiL simulátor, Rychlý vývoj, V-model

Anotace

Tato práce se zabývá testováním elektronických systémů používaných v automobilech. V úvodní části jsou zmíněny některé základní pojmy testování a přehled druhů testování a používaných metod. Následuje popis procesu vývoje elektronických řídicích systémů a princip testování hardware ve smyčce, tzv. HiL testování. Dále jsou uvedeny základní pojmy týkající se modelování dynamických systémů, které se uplatňuje při HiL testování. V další části pak následuje přehled elektronických řídicích systémů automobilů a podrobnější popis řídicích jednotek klimatizace a centrálního zamykání, vybraných pro implementaci HiL testování. Pro testování jsou zapotřebí vhodné technické prostředky. Je zde tedy rovněž popis některých dostupných prostředků, včetně struktury vybraného testovacího systému, a další programové nástroje související s procesem testování. V rámci implementace HiL testování pro zmíněné řídicí jednotky byly vytvořeny modely, jejichž popis je rovněž uveden.

Testing of electronic systems in a vehicle

Doctoral thesis

Declaration

In this thesis the author has made use of materials and resources mentioned in Chapter 9 and/or they were acquired from public Internet. Some of the terms which can be found in this thesis may be trademarks or registered trademarks of their respective owners and by mentioning of those terms in this thesis are not infringed their prospective rights.

Key words

ECU, HiL, Testing, Automotive, Electronic systems, Modelling, Comfort systems, Automatic air condition, Central locking, HiL simulator, Rapid prototyping, V-model

Abstract

This thesis deals with the testing of electronic systems used in automotive applications. In the introduction the author discusses the basic issues of testing and the main testing methods. This is followed by a description of a rapid prototyping process for fast electronic systems development and the hardware in a loop (HiL) testing. The next chapter presents the basic concept of dynamic systems modelling to be applied during HiL testing process. Structures of electronic control systems used in modern vehicles are also briefly mentioned. Special emphasis is put on objects of testing – the electronic control units for air conditioning and central locking with focus on their inputs, outputs and functions. Suitable technical equipment is needed for HiL testing implementation. An overview of suitable equipment and the structure of selected testing system are presented. Further software tools dealing with the HiL testing process are also mentioned. A description of created environment models for mentioned control units are also included.

Poděkování

Na tomto místě bych rád poděkoval svým rodičům a své manželce Mirce, za neustálou podporu a rodinné zázemí během mých studií a tvorbě této práce. Nemalý dík patří rovněž mému školiteli prof. Ing. Jiřímu Pinkerovi, CSc. za vedení během mého doktorského studia a cenné rady a připomínky k mé práci. Další poděkování patří spolupracovníkům z oddělení vývoje společnosti ŠKODA AUTO a.s., zejména doc. Ing. Jaroslavu Machanovi, CSc. a Ing. Jaromíru Kreclovi, díky nimž vzniklo téma práce, a mimo jiné umožnili uplatnění výsledků v praxi.

Obsah

1. Úvod.....	1
1.1 Cíle práce	2
1.2 Struktura textu práce.....	3
2. Základní pojmy testování rozsáhlých systémů.....	4
2.1 Filozofie testování	4
2.2 Úloha testování	5
2.3 Druhy testování.....	6
2.4 Vybrané metody testování elektronických systémů a programového vybavení	8
2.4.1 Metody pro analogové a číslicové obvody a sběrnice.....	8
2.4.2 Metody pro programové vybavení	9
3. Proces vývoje elektronických řídicích systémů a HiL testování.....	11
3.1 Vývoj elektronických řídicích systémů	11
3.1.1 Modely procesu vývoje	12
3.1.2 V-model procesu vývoje elektronických řídicích systémů	14
3.2 Princip HiL testování.....	16
3.3 Implementace HiL testování.....	17
4. Modelování dynamických systémů	19
4.1 Úvod do modelování systémů	19
4.2 Základní pojmy vytváření modelů.....	20
4.2.1 Model statického systému	20
4.2.2 Model dynamického systému	22
4.2.3 Model stavového automatu	24
4.2.4 Modelovací jazyk Modelica	26
5. Elektronické řídicí systémy současných automobilů	28
5.1 Struktura řídicího systému automobilu nižší třídy.....	28
5.2 Struktura řídicího systému automobilu vyšší třídy	30
5.2.1 Vlastnosti a funkce vybraných subsystémů.....	31
5.2.2 Systémy blízké budoucnosti	35

5.3 Testované řídicí jednotky	38
5.3.1 Automatická klimatizace Climatronic.....	38
5.3.2 Centrální zamykání s alarmem	42
6. Technické prostředky pro testování	46
6.1 Vývoj vlastního testovacího zařízení.....	46
6.2 Profesionální testovací systémy.....	47
6.2.1 Systémy společnosti National Instruments.....	49
6.2.2 Systémy společnosti dSPACE.....	51
6.2.3 Systémy společnosti MBtech Group	57
6.3 Alternativní řešení pomocí nekomerčních programů	61
6.3.1 Programy pro modelování systémů	61
6.3.2 Programy pro implementaci HiL testovací platformy.....	62
6.4 Programy pro podporu procesu testování	63
6.5 Konfigurace vybraného testovacího systému	64
7. Implementace systému dSPACE pro HiL testování	67
7.1 Nastavení a práce s vývojovými nástroji	67
7.2 Modely okolí pro vybrané řídicí jednotky	68
7.2.1 Model okolí řídicí jednotky automatické klimatizace	69
7.2.2 Model okolí řídicí jednotky centrálního zamykání	81
8. Závěr.....	93
9. Zdroje a literatura.....	94
Příloha A - Jednoduchý testovací systém	98
A.1 Blokové schéma struktura testovací platformy.....	98
A.2 Komunikace mezi testovací platformou a počítačem PC	99
A.3 Ovládací aplikace.....	100
Příloha B - Přehled distributorů profesionálních testovacích systémů v ČR.....	102
Příloha C - Publikace autora a výstupy RIV	104

Seznam termínů a zkratk

Automobilové termíny a zkratky

ACC	angl.: Automatic Cruise Control – Adaptivní tempomat
ABS	něm.: Antiblocking System – Systém zamezující zablokování
AFS	angl.: Advanced Front Lighting System – Pokročilý systém předních světlometů
ASR	něm.: Antriebs Schlupf-Regelung – Regulace prokluzu kol
CAN	angl.: Controller Area Network – Síť v oblasti řídicích jednotek
Corner	Funkce předních mlhových světlometů
DWA	něm.: Diebstahl-Warn-Anlage – Zařízení varování při krádeži (inteligentní zálohovaná siréna)
EBV	něm.: Elektronische Bremskraft-Verteilung – Elektronické rozdělování brzdné síly
ECU	angl.: Electronic Control Unit – Elektronická řídicí jednotka
EDS	něm.: Elektronische Differenzial-Sperre – Elektronická uzávěrka diferenciálu
ESP	angl.: Electronic Stabilisation Programme – Systém elektronické stabilizace
FIU	angl.: Failure Insertion Unit – Jednotka pro vkládání chyb
FlexRay	Komunikační systém pro pokročilé řídicí systémy automobilů
GALA	něm.: Geschwindigkeits-Abhängige Lautstärke-Anpassung – Přizpůsobení hlasitosti závislé na rychlosti
GRA	něm.: Geschwindigkeit-Regelung-Anlage – Zařízení regulace rychlosti (tempomat)
HBA	angl.: Hydraulic Brake Assistant – Hydraulický posilovač brzd
IRU, IRÜ, IRUE	něm.: Innen-Raum-Überwachtun – Sledování vnitřního prostoru
MiL, HiL, SiL a XiL	angl.: Model in a Loop, Hardware in a Loop, Software in a Loop, General technology with a Loop – Model ve smyčce, Hardware ve smyčce, Software ve smyčce, Obecné označení technologie se smyčkou

KESY	angl.: Keyless Entry Start and Exit System – Systém centrálního zamykání a startování bez klíče
LIN	angl.: Local Interconnect Network – Lokální propojovací síť
MBA	angl.: Mechanic Brake Assistant – Mechanický brzdový asistent
MOST	angl.: Media Oriented Systems Transport – Transportní protokol pro multimediální systémy
MSR	něm.: Motors Schlepptomoment Regelung – Regulace brzdného momentu motoru
NGS	něm.: Neigung-Sensor – Snímač náklonu
OEM	angl.: Original Equipment Manufacturer – Výrobce zařízení (např. automobilu nebo počítače)
PDC	angl.: Park Distance Control – Indikace vzdálenosti při parkování
PLA	něm.: Park Lenk Assistent – Parkovací asistent
RNS	angl.: Radio Navigation System – Radiopřijímač s navigačním systémem
RSE	angl.: Rear Seat Entertainment – Systémy zábavy na zadních sedadlech
SIL, ASIL	angl.: Safety Integrity Level, Automotive Safety Integrity Level – Úroveň integrity bezpečnosti, Úroveň integrity bezpečnosti v automobilech

Ostatní termíny a zkratky

AD	angl.: Analogue / Digital – Analogově digitální
ASCII	angl.: American Standard Code for Information Interchange – Americký standardní kód pro předávání informací
CAE, CAD, CAM	angl.: Computer Aided Engineering, Design, Manufacturing – Počítačem podporované inženýrství, navrhování, výroba
CASE	angl.: Computer Aided Software Engineering – Počítačem podporované softwarové inženýrství
CACSD	angl.: Computer Aided Control System Design – Počítačem podporované navrhování řídicích systémů
CD	angl.: Compact Disc – Kompaktní disk

CMOS	angl.: Complementary Metal–Oxide–Semiconductor – Komplementární kov-oxid-polovodič (technologie výroby integrovaných obvodů)
CRC	angl.: Cyclic Redundancy Check – Kruhové kontrolní ověření, též kontrolní součet
CTM	angl.: Classification Tree Method – Metoda klasifikačního stromu
DA	angl.: Digital / Analogue – Digitálně analogový
DAB	angl.: Digital Audio Broadcasting – Digitální rozhlasové vysílání
DSP	angl.: Digital Signal Processor – Procesor pro číslicové zpracování signálu
DVB-T, -S, -C	angl.: Digital Video Broadcasting – Terrestrial, – Satellite, – Cable – Digitální televizní vysílání – pozemní, – satelitní, – kabelové
DVD	angl.: Digital Versatile Disc – Digitální všestranný disk
EOL	angl.: End Of Line – Konec výrobní linky
FM	angl.: Frequency Modulation – Frekvenční modulace
FPGA	angl.: Field Programmable Gate Array – Programovatelné hradlové pole
GPL	angl.: General Public License – Všeobecná veřejná licence
GPS	angl.: Global Positioning System – Globální systém pro určení polohy
GSM	původně z fran.: Groupe Spécial Mobile, dnes známé z angl.: Global System for Mobile Communications – Globální systém pro mobilní komunikaci
HAL	angl.: Hardware Abstraction Layer – Vrstva abstrakce hardwaru
ISA	angl.: Industry Standard Architecture – Architektura průmyslového standardu (počítačová sběrnice)
ISO	angl.: Industry Standard Organization – Organizace pro průmyslovou standardizaci
ISP	angl.: In System Programming – Programování v systému, resp. v zapojení
JTAG	angl.: Joint Test Action Group – Spojená akční skupina pro testování, která definovala standardizované testovací rozhraní
LCD	angl.: Liquid Crystal Display – Zobrazovač s tekutými krystaly

LFSR	angl.: Linear Feedback Shift Register – Lineární posuvný registr se zpětnými vazbami
LSB, MSB	angl.: Most Significant Bit, Least Significant Bit – Nejvíce, resp. nejméně významný bit v bajtu, případně slově
MCDC	angl.: Modified Condition/Decision Coverage – Modifikované pokrytí podmínky/rozhodnutí
MCU	angl.: Microcontroller Unit – Jednočipový mikropočítač
NTC	angl.: Negative Temperature Coefficient – Záporný teplotní koeficient
PC	angl.: Personal Computer – Osobní počítač
PCI	angl.: Peripheral Component Interconnect – Připojení periferních komponent (počítačová sběrnice)
PWM	angl.: Pulse Width Modulation – Pulsně šířková modulace
PXI	angl.: PCI eXtensions for Instrumentation – Rozšíření sběrnice PCI pro měření a řízení
RPC	angl.: Remote Procedure Call – Vzdálené volání funkcí
RT	angl.: Real-time – Reálný čas
RS232	angl.: RS232/EIA232 Standard – Norma RS232, dnes přejmenovaná na EIA232, definující sériovou komunikaci
SD	angl.: Secure Digital – Typ paměťové karty
SIM	angl.: Subscriber Identification Module – Účastnická identifikační karta
SPI	angl.: Serial Peripheral Interface – Sériové rozhraní pro periférie
STT	angl.: State Transition Technique – Technika přechodů mezi stavy
TTL	angl.: Transistor-Transistor Logic – Logika typu tranzistor-tranzistor (technologie výroby číslicových integrovaných obvodů)
UML	angl.: Unified Modelling Language – Univerzální modelovací jazyk
USART	angl.: Universal Synchronous-Asynchronous Receiver-Transmitter – Univerzální synchronní/asynchronní přijímač/vysílač
USB	angl.: Universal Serial Bus – Univerzální sériová sběrnice
XP	angl.: eXtreme Programming – Extrémní programování

1. Úvod

V dnešní době se v automobilech setkáváme s neustálým nárůstem podílu elektronických systémů, a to i v nejnižších cenových kategoriích. Je to zřejmě způsobeno snahou výrobců automobilů o uspokojování požadavků trhu, mezi které patří snižování ceny a zvyšování bezpečnosti a komfortu automobilů, a dále i plnění stále se zostřujících technických a legislativních požadavků (zejména emisních limitů).

Dříve (řekněme v 80. letech 20. století) stačilo, aby automobil (úrovně Škoda 120) startoval, svítil a blikal, troubil, stíral okna a topil, resp. větral. Vrcholem elektroniky v té době (v tehdejším Československu) bylo autorádio a elektronické zapalování, které však nebylo dodáváno sériově a jednalo se v podstatě o monostabilní klopný obvod prodlužující délku výboje na svíčke. Je třeba podotknout, že i v této době již existovaly automobily s mnohem větší elektronickou výbavou. Jako příklad vezměme např. Renault 25. Ten se vyráběl od roku 1983 a patřil mezi špičku tehdejších automobilů. V jeho výbavě se nacházelo např. elektrické stahování oken, elektrické vyhřívání zrcátek, elektricky stavitelná sedadla, palubní počítač s hlasovým výstupem, apod. Tak jako byla nesrovnatelná vybavenost a komfortnost vozů Škoda 120 a Renault 25, byla podobně nesrovnatelná i pořizovací cena, která u Š120 činila kolem 40.000,-Kčs a u R25 řádově milion korun.

Trh však neustále vyvíjí tlak na zvyšování kvality a vybavenosti i u automobilů nejnižší třídy, což však přináší zvyšování ceny. Zvyšování ceny však trh připustí jen do určité míry a další možnosti, jak udržet zisk se zvyšováním kvality a udržením příznivé ceny, jsou ve snižování nákladů na výrobu. Toho lze dosáhnout např. objemem výroby. Ale objem výroby je rovněž omezený, protože v tržním hospodářství nelze vyrábět tzv. na sklad, ale jen to, co se skutečně prodá. Zvýšení objemu nakupovaných součástí lze dosáhnout tím, že součást bude univerzální pro více modelů. To má za následek modularizaci celé výroby, která postupně vyústí v postupnou globalizaci nejen jednotlivých podniků, ale i celého světa. Jednotliví výrobci automobilů se postupně sdružují do nadnárodních koncernů, ve kterých pak existují tzv. modelové platformy, kde díly, které zákazník nevidí, jsou stejné pro všechny značky koncernu, a netýká se to jen elektroniky, ale i mechanických dílů jako je např. podvozek, brzdy, nosná konstrukce karoserie apod. Příklad modelové platformy byl prezentován ve Škoda Auto museu [29]. Tímto postupem lze dosáhnout značného objemu některých dílů a snížit tak jejich náklady.

S velkými objemy tzv. platformových dílů se klade důraz i na spolehlivost těchto dílů. Zejména proto, že jsou určeny pro automobilový průmysl, kde je kladen zvýšený důraz na bezpečnost a spolehlivost produktů obecně. Se spolehlivostí souvisí obory jako testování a diagnostika. V poslední době se ve větší míře začíná věnovat pozornost testování i elektronických systémů ve vozidle. Dříve se testování provedlo jednoduše, stylem svítí-nesvítí, stírá-nestírá, atd. Zřejmě tomu tak bylo i u složitějších systémů vozu R25, tedy stylem funguje-nefunguje. Dnes je situace mnohem složitější. Jednak jsou elektronické systémy komplikovanější (mají více funkcí), navíc jsou to většinou univerzální platformové díly, tedy jsou programovatelné a dále spolu jednotlivé subsystémy komunikují. Tudíž jednoduché testování se už příliš neuplatní z důvodu složitosti testovaných systémů a s tím související časovou náročností testování. Je nutné brát v potaz zejména náhodné chyby, které mohou vznikat součinností těchto propojených systémů při zajišťování komplexní funkcionality, kdy např. funkce parkovacího asistenta (PLA, viz dále popis v kapitole 5.2.1) je zajišťována nejen vlastní řídicí jednotkou PLA, ale v součinnosti s řídicími jednotkami posilovače řízení a převodovky. Takové propojené a komunikující subsystémy pak tvoří distribuovaný řídicí systém. Funkčnost jednotlivých systémů se testuje během vývoje i výroby u výrobců, ale součinnost lze otestovat až v příslušné konfiguraci při vývoji nadřazeného systému (v tomto případě automobilu).

Vzhledem ke složitosti elektronického systému dnešního automobilu, je nutné přistupovat k testování komplexně již během celého procesu vývoje automobilu. Vývojem a implementací testovacích systémů je sledována snaha o automatizaci nejen průběhu testování, ale i jeho vyhodnocování. Tedy, aby byl vytvořen univerzální, pokud možno jednoduše konfigurovatelný testovací systém, který v určitém konečném čase (v řádu desítek hodin, např. přes noc nebo víkend) otestuje elektronický distribuovaný systém automobilu a provede jeho vyhodnocení.

1.1 Cíle práce

Cíle této disertační práce se dají shrnout do následujících bodů.

- Specifikace parametrů testovací platformy pro danou sestavu řídicích jednotek
- Výběr vhodného typu a konfigurace testovací platformy
- Vytvoření modelů okolí pro dané řídicí jednotky
- Zhodnocení výhod implementace HiL testování ve vývojovém procesu

1.2 Struktura textu práce

Text práce je členěn do následujících kapitol. V kapitole **2. Základní pojmy testování rozsáhlých systémů** jsou zmíněny základní pojmy související s testováním elektronických systémů. Tato kapitola dále zahrnuje rozdělení testování a stručný popis vybraných metod testování elektronických systémů a programového vybavení.

Třetí kapitola s názvem **Proces vývoje elektronických řídicích systémů a HiL testování** seznamuje s moderními modely procesu vývoje automobilových elektronických systémů, umístěním testování ve smyčce (HiL) v tzv. V-modelu procesu vývoje a principem HiL testování.

Čtvrtá kapitola **Modelování dynamických systémů** seznamuje s možnostmi a postupy modelování dynamických systémů, které je nedílnou součástí HiL testování, kdy je model okolí testované elektronické řídicí jednotky provozován v reálném čase na HiL testovací platformě.

V páté kapitole **Elektronické řídicí systémy současných automobilů** se nachází popis struktury a funkcionality elektronických řídicích systémů jak automobilu nižší třídy, tak i třídy vyšší. Okrajově jsou zmíněny systémy, které by se mohly objevit v automobilech v blízké budoucnosti. Podrobněji je popsána funkcionality přidělených testovaných řídicích jednotek automatické klimatizace a centrálního zamykání.

Implementace HiL testování se neobejde bez nutného technického vybavení. Popisem některých testovacích systémů se zabývá kapitola **6. Technické prostředky pro testování**. Kromě popisu profesionálních testovacích systémů tato kapitola zahrnuje i alternativní nekomerční programové vybavení a zejména pak konfiguraci vybraného testovacího systému, na kterém byla provedena implementace modelů okolí pro zmíněné řídicí jednotky.

Vlastní implementace HiL testování je popsána v následující kapitole **7. Implementace systému dSPACE pro HiL testování**. Největší pozornost je věnována vlastnímu popisu implementovaných modelů okolí pro testované řídicí jednotky.

Následuje pak závěrečné shrnutí, citované zdroje a literatura a přílohy. V příloze A s názvem **Jednoduchý testovací systém** se nachází popis jednoduché testovací HiL platformy realizované pro seznámení se s funkcemi dveřních řídicích jednotek. V příloze B jsou uvedeny kontaktní informace na dodavatele technických prostředků a v příloze C jsou sumarizovány publikace autora a záznamy v databázi RIV.

2. Základní pojmy testování rozsáhlých systémů

2.1 Filozofie testování

„Chybovat je lidské“ je letitá omluva nepovedené práce. Zatímco své chyby omlouváme, u výrobků, které kupujeme za své peníze, vyžadujeme dokonalou bezchybnost. To platí zvláště v automobilovém průmyslu, vzhledem k cenám vozidel, ve srovnání s jinými výrobky používaných v běžném životě. Jak, ale může nedokonalý člověk vyrobit dokonalý produkt? Zde je nastíněno řešení:

```
Algoritmus: Dokonalý produkt  
Návrh;  
Provedení;  
Opakuj dokud není test(produkt) == dokonalý:  
{  
    Úprava návrhu;  
    Úprava provedení;  
}
```

Je nutné zdůraznit, že *test* by měl být navržen tak, aby správně indikoval *požadovanou* dokonalost a také, že smyčka algoritmu se bude opakovat do nekonečna, pokud *návrh* a *provedení* nezareagují nebo neopraví chyby nalezené při testování.

Porozumění umění a vědě o testování nám umožní dosáhnout dokonalosti v naší práci, neboli zvládnout tzv. nadlidskou úlohu.

Vezměme si příklad testování – zkoušení studentů ve škole. Všichni jsme chodili nebo ještě chodíme do školy a slovo *test* v nás vyvolává nepříliš příjemný pocit. Ovšem porozumění úhlu pohledu učitele nám může pomoci při zdolávání testu. Učitel většinou stanoví *okruh znalostí*, které se budou zkoušet (testovat), tzv. *syllabus* předmětu. Může to být obsah knihy, přednášek či cvičení, případně jejich libovolná kombinace. Pak následuje zkoušení (testovací metoda). Učitel pokládá dotazy a analyzuje odpovědi (odezvy) nejčastěji porovnáváním odpovědí se správnými, např. podle knihy. Kvalita tohoto testovacího systému závisí na tom, jak otázky pokrývají obsah syllabu. Při testování elektronických systémů je to obdobné. Je nutné znát *specifikaci* testovaného objektu, resp. *požadavky* na něj (podobně jako syllabus předmětu) a pak je možné navrhnout vhodný test, aby mohlo být zaručeno splnění specifikace (požadavků), když testovaný objekt dává požadované odezvy.

Jelikož pro zkoušení studentů není nekonečně mnoho času, musí být počet otázek omezen a měly by být důmyslně navrženy. Lze předpokládat, že dojde k výskytu typických chyb, tedy takových, kterých se studenti velmi pravděpodobně dopustí. Otázky by měly být navrženy zejména tak, aby odhalily tyto chyby, a i když odpovědi studentů jsou správné, je nutné připustit určitou prospěchovou toleranci v rámci daného chybového modelu. Při testování elektronických systémů se využívá *modelování poruch* a testy jsou vytvořeny pro předpokládané modely poruch. Úspěšně vyzkoušenému modelu poruch se přisuzuje věrohodnost. Lze očekávat, že systém bude spolehlivý, jestliže bude testováno velké, resp. požadované, procento modelových chyb.

Když student neprojde zkoušením, pak musí předmět opakovat. To je podobné jako úprava návrhu a provedení ve výše uvedeném algoritmu „Dokonalý produkt“. Student si může usnadnit práci, když si předem zjistí informace o sylabu a o chybovém modelu (jinými slovy z čeho se bude zkoušet), a pak si může studium lépe naplánovat a následně uspět. U testování elektronických systémů se pak v tomto smyslu používá pojem *návrh testovatelnosti*.

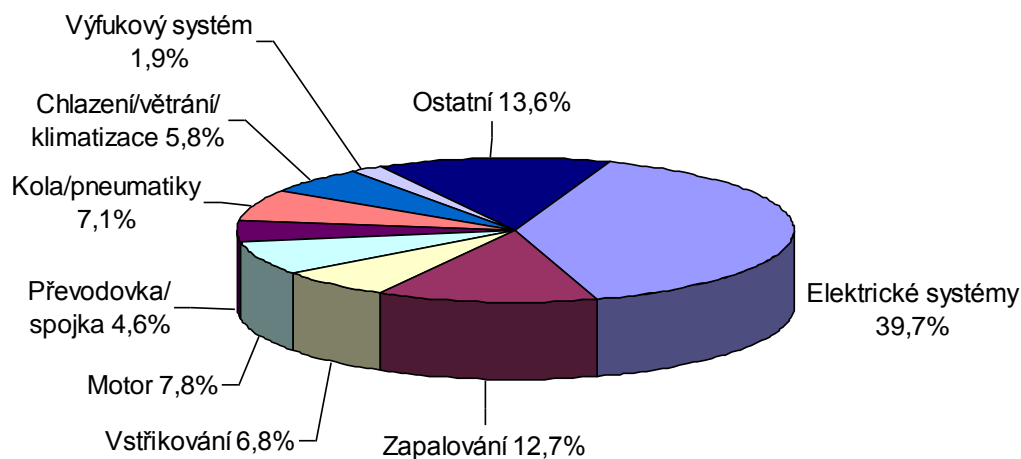
2.2 Úloha testování

Když navrhne produkt, vyrobíme a testujeme jej a on neprojde testem, tak zcela jistě existuje příčina selhání. Buď <1> je špatné testování (tzn. špatně navržený test, chyba v testovací sekvenci, případně nesprávně vyhodnocené výsledky testu), nebo <2> je zmetková výroba, nebo <3> je špatný návrh, případně jeho implementace, nebo <4> máme problém ve specifikaci. Cokoliv může být špatně. Úkolem *testování* je detekovat, zda je někde problém, a úkolem *diagnostiky* je pokud možno přesně určit, kde je problém a kde je potřeba proces upravit. Nicméně správnost a efektivita testování je velmi důležitá pro kvalitní (dokonalé) produkty.

Jestliže je metodika testování dobrá, ale produkt neprojde testem, pak můžeme podezřívat výrobní proces, návrh nebo specifikaci. Testování, rozložené v celém realizačním procesu produktu, zachytí příčiny chybovosti během jejich vzniku dříve, než způsobí další a větší ztráty. Dobré promyšlení testovací strategie je významné pro hospodárnost výroby produktů.

Přínos testování je kvalita (jakost) a hospodárnost. Tyto dva atributy nejsou navzájem nezávislé, jeden nelze stanovit bez druhého. *Jakost znamená uspokojení potřeb spotřebitele při minimálních nákladech*. Dobrý proces testování odstraní všechny nevyhovující produkty dříve, než dorazí ke spotřebiteli. Jenže, vyrobí-li se příliš mnoho

zmetků, tak náklady na jejich výrobu jsou kryty výnosem z prodeje těch dobrých, kterých je ale méně, takže klesá rentabilita. Není proto možné navrhnout kvalitní výrobek (produkt) bez hluboké znalosti fyzikálních principů, na kterých jsou založeny procesy výroby a testování.



Obr. č. 2.1 Rozdělení příčin poruch automobilů v roce 2007 podle ADAC [38]

Úlohu testování a její význam podtrhuje i narůstající podíl elektronických systémů v dnešních automobilech a nárůst funkcionalit jimi zajišťovaných. S tím souvisí i narůstající podíl elektronických systémů mezi příčinami poruch vozidel, jak je znázorněno v předcházejícím obr. č. 2.1.

2.3 Druhy testování

Jak bylo uvedeno výše, je vhodné zavést testování podél celého procesu výroby. Proto se dá testování rozdělit do následujících kategorií.

Ověřovací testy

Ověřovací testy se provádějí v průběhu definice specifikace (zadání) a návrhu, resp. vývoje systému. Mají za úkol odhalit nejen vývojové chyby, např. porušení principů funkčnosti, rozměrové chyby při umístování součástí, apod., ale i chyby ve specifikaci.

Funkční testy

Funkční testování slouží k ověření, že navržený systém splňuje funkci podle specifikace a že nedochází k nedefinovaným stavům, či výstupům při nestandardních vstupech. Tyto testy rovněž potvrzují správnost návrhu a vývoje.

Testování parametrů

Součástí specifikace jsou i parametry systému, jako různé charakteristiky vstupů, či výstupů, rozsah napájecího napětí, vlastnosti prostředí, ve kterém je systém provozován, resp. skladován, apod. Testování parametrů slouží k ověření, zda naměřené parametry splňují specifikaci, případně jaké vlastně jsou – nejsou-li exaktně specifikovány.

Zahořovací testy

Zahořovací testy slouží k ověření spolehlivosti a odolnosti systému. Většinou se provádí za extrémních provozních podmínek, aby při normálních podmínkách byla zaručena určitá míra spolehlivosti. Zahořovací testy také mohou sloužit k selekci vyrobených kusů – to se používá především u polovodičů, např. u mikroprocesorů, kdy se zkouší, při jaké maximální frekvenci je ještě spolehlivě zaručena funkce a podle výsledku testování pak dojde k označení daného kusu.

Výrobní testy

Výrobní testy, někdy také označovány jako testy na konci výrobní linky (EOL testy – angl.: end of line), bývají součástí systému řízení jakosti podniku a slouží k ověřování spolehlivosti výroby a k omezování zmetkovitosti.

Testování lze kategorizovat rovněž podle úrovně znalostí o testovaném objektu a možnosti zjišťovat či nastavovat vnitřní stavy. Z tohoto hlediska se rozlišují dva druhy testování tzv. *Black-Box*, kdy k testovanému objektu lze přistupovat pouze z venku a informace o vnitřní struktuře jsou malé nebo žádné, a tzv. *White-Box*, kdy informace o vnitřní struktuře jsou velmi přesné a navíc je umožněno pracovat s vnitřními stavy.

Testy bývají často navzájem kombinovány a např. pro získání certifikátu jakosti výroby musí být testování implementováno podél celého procesu vývoje i výroby produktu.

Testování může provádět i zákazník, a to z důvodu, že buď nedůvěřuje testům výrobce a spolehlivost nakupovaných produktů je pro něj důležitá, nebo potřebuje otestovat funkčnost a zejména součinnost více systémů od různých výrobců. Takové testování se nazývá integrační a je kombinací funkčních a zahořovacích testů, kdy se testuje funkčnost výrobku při normálních i ztížených provozních podmínkách.

2.4 Vybrané metody testování elektronických systémů a programového vybavení

Cílem této kapitoly není přebírat informace z odborné literatury ve větším rozsahu, proto jsou zde jen stručně shrnuty nejpoužívanější metody pro testování jednak elektronických systémů [1], [2], [3] a také programového vybavení[4], [5].

2.4.1 Metody pro analogové a číslicové obvody a sběrnice

Číslicové elektronické obvody

Pro testování číslicových obvodů se zavádí modely chyb (poruch) typu trvalá logická „0“ a trvalá logická „1“, které umožňují modelování chyb typu rozpojení nebo zkrat na jednu z napájecích svorek. U CMOS obvodů se navíc musí testovat i přechody mezi logickými úrovněmi, poněvadž nezapojený vstup nemá definovanou úroveň tak, jako to měly obvody typu TTL. Základními metodami pro testování číslicových kombinačních obvodů jsou metody pro propagaci chyby na pozorovatelný výstup a její detekci. Patří sem metoda *intuitivního zcitlivění cesty*, díky níž lze snížit počet testovacích vektorů na minimum při zachování úplnosti testu, a z ní vycházející *booleovská diference*, která generuje testovací vektor (pokud existuje) pro konkrétní hledanou chybu. Další metodou pro generování detekčního testu je *D-algoritmus*, který, jak již jeho název napovídá, je předurčen k automatizovanému zpracování.

Pro testování sekvenčních obvodů je nutné s testováním počítat již při návrhu, poněvadž je nutné zasahovat do vnitřního stavu. Pro čtení a nastavování vnitřních registrů se běžně používá standardizované rozhraní JTAG. Dříve se také používala *příznaková analýza*.

Analogové elektronické obvody

U analogových obvodů nelze modelovat chyby tak jednoduše jako u číslicových. Vzhledem k tomu, že obvodové veličiny jsou spojité, je pro projití testem vyžadováno, aby sledovaná hodnota byla v určitém intervalu (angl.: pass/fail intervals). Vyhodnocení se pak provádí statisticky. Někdy je potřeba sledovat i změny veličiny v čase, kdy se v časovém průběhu vymezují oblasti, kudy by měl signál procházet. Tato metoda se uplatňuje i při testování vysokorychlostních číslicových signálů, kde se již ve značné míře projevují parazitní vlastnosti obvodů a spojů. Pak se diagramy, které vymezují oblasti, kudy má signál procházet, nazývají tzv. *oční digramy* (angl.: eye-diagrams).

Komunikační sběrnice

Při testování komunikačních sběrnic je třeba generovat vhodná testovací data, která se přenesou daným komunikačním kanálem, a následně dojde k vyhodnocení přenosu. Jako generátor testovacích dat může být použit např. *lineární zpětnovazební posuvný registr* (LFSR – angl.: linear feedback shift register) s vhodným generujícím polynomem (umístění zpětných vazeb).

Některé sběrnice mají kromě vlastní specifikace, normalizováno i testování (angl.: test specification). Např. pro sběrnici LIN bus [19] jsou specifikovány požadavky na testování a to jak fyzické vrstvy (napětíové úrovně, rychlosti hran) tak i linkové vrstvy (formát rámce).

Při testování vyšších vrstev komunikace, podle ISO-OSI modelu, je někdy nutné provádět simulaci komunikujících uzlů (angl.: rest-bus simulation) včetně jejich elementárního chování, tedy reakce na přenášená data, eventuálně vnější podněty.

2.4.2 Metody pro programové vybavení

Na programové vybavení je v této práci nahlíženo jako na součást systému zajišťující funkcionální. Proto zde budou uvedeny jen metody, které se týkají funkčního testování, nikoliv např. softwarových metrik, pokrytí kódu (angl.: code coverage), apod. Metody testování lze rozdělit do dvou skupin na formální a neformální. Formální metody mají matematický základ, lze u nich přesně stanovit pokrytí testu, vyžadují kvalitní informace o testovaném objektu a někdy jejich nasazení tak nemusí být výhodou. Naproti tomu neformální metody jsou založeny na intuici a zkušenostech vývojáře testů, je obtížné stanovit pokrytí testu a někdy mohou dávat stejné nebo podobné testovací případy jako formální metody. V praxi se využívají oba druhy metod.

Mezi základní neformální metody patří *metoda tříd ekvivalence*. Ta spočívá v klasifikaci stavů nebo hodnot. Lze s ní vygenerovat testovací případy např. pro funkcionální popsanou pomocí booleovského výrazu, princip je podobný jako u intuitivního zcitlivění cesty.

Na třídy ekvivalence navazuje *metoda klasifikačního stromu* (CTM – angl.: classification tree method), která navíc umožňuje pracovat s intervaly hodnot, hraničními hodnotami a neplatnými hodnotami. Jednoduchý editor pro práci s klasifikačními stromy, tedy pro vytvoření stromu a následně testovacích případů, lze stáhnout např. u firmy Berner&Mattner Systemtechnik GmbH, která nabízí jednu z verzí zdarma, viz [39].

Mezi neformální techniky dále patří testování *případů použití* (angl.: use case testing). To je relativně volná technika generování testovacích případů na základě očekávání (resp. způsobu používání) uživatele. Uplatní se zvláště při přejímacích testech, kdy si zákazník definuje, jakým způsobem chce systém používat, a při předávce je mu tato požadovaná funkcionální předvedena.

Pro testování systémů, jejichž funkcionální lze pospat stavovým diagramem, lze použít formální metodu *přechodu stavů* (STT – angl.: state transition technique). Metoda vyžaduje znalost vnitřní struktury systému, tedy stavového diagramu. Důležité je stanovení počátečních a konečných podmínek (stavů) a zjištění, jak systém uvést do výchozího stavu. Pokrytí testu vychází z hloubky testu. Nejjednodušší test je přechod z výchozího stavu nejkratší cestou do konečného, ne nutně všemi stavy. Nicméně standardně se testují v každém stavu všechny pro něj definované události, tzn. legální případy. Nelegální případy jsou testování událostí na jiné stavy, tzv. plíživé cesty (angl.: sneak path) a přechody z nespecifikovaných událostí (angl.: trapped doors), které však nejsou pomocí STT testovány, takže pokrytí nalezení všech chyb je menší než 100%.

Když je funkcionální systému popsána pomocí vývojového diagramu, typicky funkcionální zajišťovaná programem, pak lze použít formální metodu nazývanou *strukturální test*. Metoda se soustředí na rozhodovací bloky ve vývojovém diagramu a cesty jimi procházející. Podle zvoleného kritéria hloubky testu, pak lze sestavit testovací případy, tedy testovací cesty průchodu programem. Hloubka testu N znamená, že N po sobě jdoucích průchodů budou provedeny. Pokrytí lze vyjádřit v několika stupních. C0-pokrytí znamená, že každý výkonný blok (příkaz) bude proveden; C1-pokrytí, každý průchod mezi rozhodovacími bloky bude proveden; C2-pokrytí, každá cesta přes rozhodovací blok bude provedena; C3-pokrytí, vícenásobné pokrytí cesty. S ohledem na součet všech příkazů, přechodů a cest může být stanoveno procentuální vyjádření C x -pokrytí.

Jako poslední zde uvedená formální metoda je *modifikované pokrytí podmínek a/nebo rozhodnutí* (MCDC – angl.: modified condition/decision coverage), která se uplatňuje i při testování kritických systémů v leteckém průmyslu [40]. Princip spočívá ve snížení počtu testovacích případů z 2^n , kde n je počet podmínek (Booleovských proměnných nebo výrazů porovnání), tedy z triviálního testu, na počet $n+1$ se zachováním pokrytí požadovaném i v kritických aplikacích. Metoda je velmi dobře standardizována, viz např. norma DO-178B [41].

3. Proces vývoje elektronických řídicích systémů a HiL testování

Testování řídicích jednotek během vývojového procesu bývá založeno na kombinaci druhů testování uvedených v kapitole 2.3 (s výjimkou výrobních testů) s tím, že předmětem testování je především samotná jednotka a tudíž je nezbytné mít možnost libovolně nastavovat parametry okolí jednotky. Toho lze dosáhnout buď pomocí speciálního konfigurovatelného okolí, nebo pomocí simulace tohoto okolí. Simulace umožňuje mnohem širší rozsah parametrizace, snadno generovat i nereálné stavy nebo hodnoty a bývá většinou jednodušší na realizaci. Obecně lze říci, že testovaná řídicí jednotka je během testování zapojena ve smyčce, ať už k modifikovanému reálnému či simulovanému okolí. Pak se takové testování nazývá testování hardwaru ve smyčce, zkráceně HiL testování (angl.: hardware in the loop).

Dalším důležitým pojmem, se kterým se setkáváme v rámci testování elektronických řídicích jednotek, je SiL (angl.: software in the loop). Jedná se o testování používané především během počátku vývoje řídicí jednotky, kdy jednotka ještě fyzicky neexistuje, ale probíhá vývoj jejího programového vybavení, (implementace funkcionality). Programové vybavení je tedy virtuálně provozováno ve vývojovém prostředí simulující okolí řídicí jednotky. Lze tak v rámci testu snadno měnit jeho chování a sledovat vliv na okolí. Simulaci lze snadno zastavit, nebo dokonce vrátit v čase, a pokud se na projektu podílí často až desítky programátorů, jde rovněž o velmi levnou metodu. Tento postup navíc umožňuje oddělit vývoj programového vybavení od elektronického a mechanického vývoje řídicí jednotky a tím urychlit celý vývojový proces.

3.1 Vývoj elektronických řídicích systémů

Při vývoji elektronického řídicího systému, ať už automobilového nebo obecně průmyslového, lze postupovat různým způsobem, v závislosti na složitosti řídicího systému i řízeného objektu, na velikosti realizačního týmu a dalších vnitropodnikových postupech. Postup vývoje lze rozdělit na tři hlavní oblasti z pohledu technického inženýrství.

- Návrh, vývoj a testování mechanických částí řídicího systému a jeho začlenění do celku, což je doménou strojního inženýrství.

- Návrh, vývoj a testování elektrických a elektronických částí, tzv. hardware (HW), což je hlavní doménou elektrotechnického inženýrství.
- Návrh, vývoj a testování programového vybavení, tzv. software (SW), u vestavěných (též vložených, angl.: embedded) systémů se někdy používá i pojem firmware (FW). Programové vybavení bylo považováno za součást elektrotechnického inženýrství, nicméně vzhledem ke komplexnosti problémů se postupem času odděluje jako samostatná oblast – softwarové inženýrství.

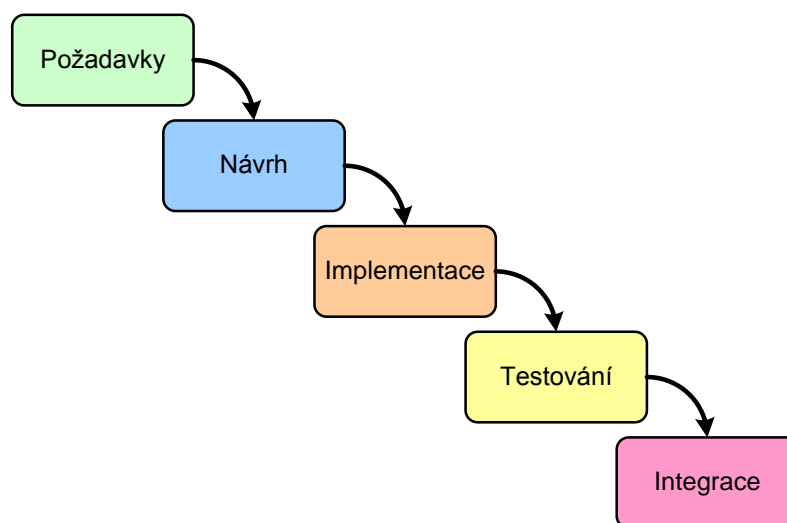
Pozn.: V poslední době se stává velmi moderní pojem mechatronický systém pro systém, který má úzce provázanou mechanickou část s řídicí elektronikou, kdy nedílnou součástí řídicí elektroniky je nutně jak hardware tak i software. Oblast inženýrské činnosti se pak označuje jako mechatronika.

3.1.1 Modely procesu vývoje

Postup během vývoje, resp. proces vývoje lze pospat pomocí modelu. Modelování procesů vývoje je známo především z oblasti softwarového inženýrství, nicméně tyto modely lze aplikovat i na strojní a elektrotechnické inženýrství. Postupně vzniklo několik modelů.

Vodopádový model

Vodopádový model (angl.: waterfall) je nejjednodušším modelem využívaným k popisu procesu vývoje. Jedná se o přímočarý (lineární) model, kdy jednotlivé etapy navazují jedna na druhou s minimálním časovým překrytím, a v jeden okamžik se v rámci jednoho projektu tedy pracuje na právě jedné etapě.



Obr. č. 3.1 Vodopádový model procesu vývoje s postupně navazujícími etapami

Implementace procesu vývoje podle tohoto modelu je vhodná jen v případech, kdy se pevně stanoví požadavky na vyvíjený systém, které se až do konce vývoje nezmění, poněvadž jakákoliv změna je velmi nákladná. Náklady na změny a na opravu chyb, které nejsou včas odhaleny, jsou hlavní nevýhody tohoto modelu, a proto je dnes víceméně na ústupu a je nahrazován modely umožňující pružnější zpracování změn.

Spirálový model

Spirálový model je iteračně přírůstkový (inkrementální) model, kdy se každou iterací provádí zlepšování výsledného produktu, ať už programu, nebo celého systému. V rámci jedné iterace se postupuje podobně jako u vodopádového modelu, ale postupně dochází k zpřesňování požadavků a narůstání systému po jakési spirále (odtud název modelu), postupně vznikají prototypy zařízení (beta verze software). Řízení projektu a zajištění kvality je však udržitelné pouze u malých vývojových týmů.

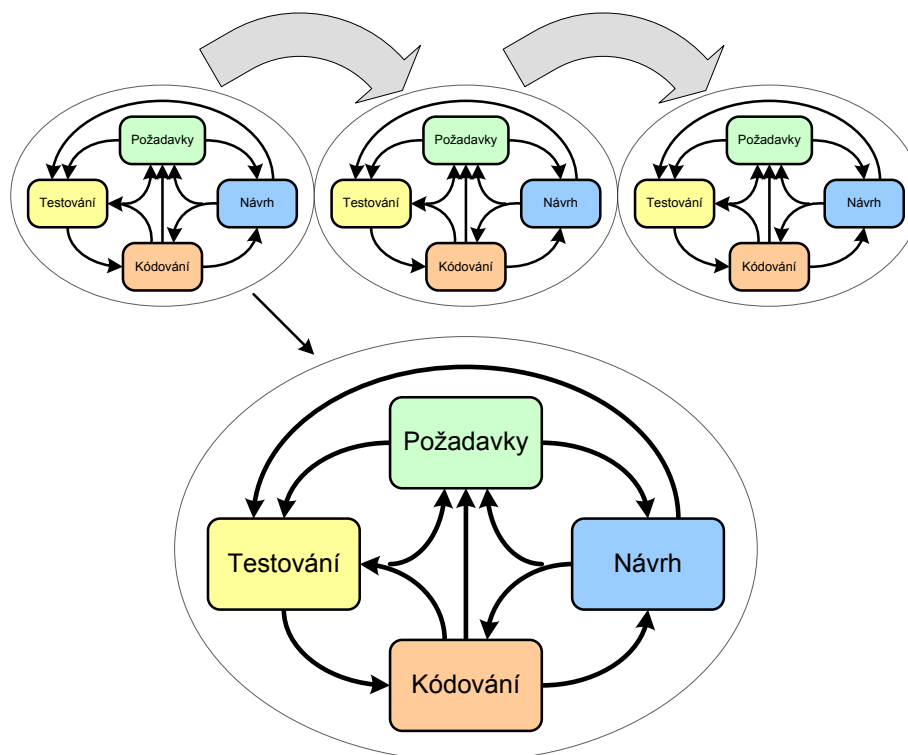
V-model

V-model vychází z vodopádového modelu, ale je v místě implementace zalomen tak, že tvoří písmeno „V“. To umožňuje zlepšit účinnost testování a snížit náklady na odstranění chyb. Tento model je v procesu vývoje automobilových řídicích systémů velmi aktuální a proto je mu věnována samostatná kapitola 3.1.2.

Agilní model

Agilní model popisuje moderní a pružný způsob vývoje s výraznou orientací na požadavky zákazníka a zohledňuje zpracování změn požadavků během vývoje. Jeho uplatnění je zejména v oblasti vývoje programového vybavení, kde pružnost v dnešním konkurenčním prostředí je nespornou výhodou a proto agilní model začíná překonávat V-model, viz [6]. Způsob vývoje softwaru podle agilního modelu se také někdy označuje jako extrémní programování (XP – angl.: extreme programming).

Jedná se o iterační model, který umožňuje dodávat zákazníkovi fungující software (podle aktuálních požadavků) co nejčastěji. Využívá se jednoduchého programování v souladu s danými pravidly, neustálé revize kódu a paralelního programování, kdy pracují dva vývojáři společně. Jednotlivé iterace jsou znázorněny v horní části obr. č. 3.2.



Obr. č. 3.2 Agilní model procesu vývoje – nahoře je znázorněn iterační postup, dole pak detail předávání poznatků mezi jednotlivými etapami v rámci jedné iterace

Pro udržení a zlepšování kvality se využívají poznatky o problémech ze všech etap vývoje. Jejich vzájemná provázanost je znázorněna na detailu jedné iterace v obr. č. 3.2.

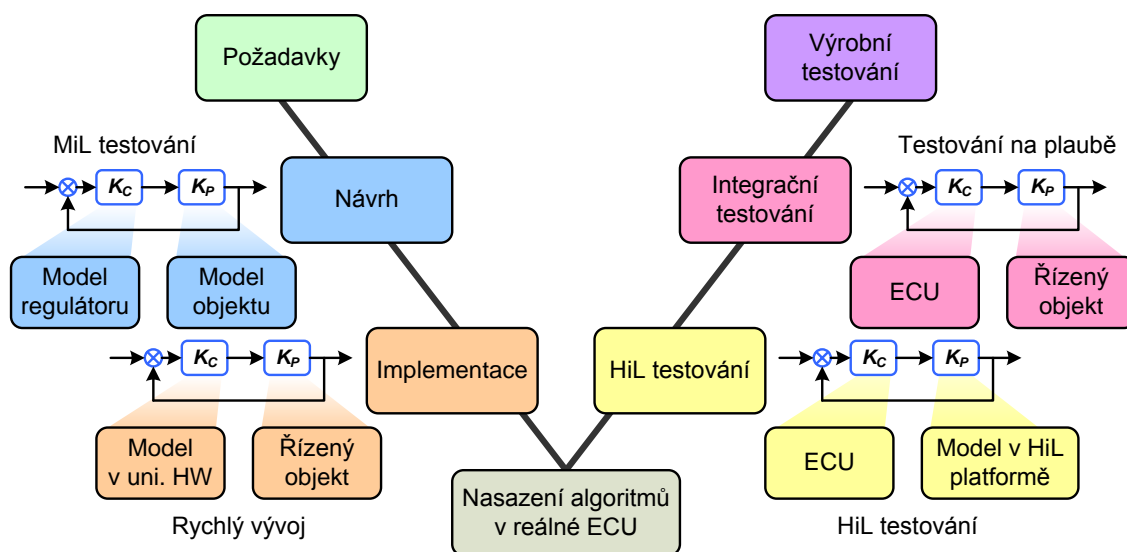
3.1.2 V-model procesu vývoje elektronických řídicích systémů

Jednotlivé řídicí systémy v automobilu lze považovat za zpětnovazební systémy skládající se z řídicí jednotky (ECU – angl.: electronic control unit), resp. regulátoru (angl.: controller; označení používané v regulační technice) s přenosovou funkcí K_C a řízeného objektu (angl.: plant) s přenosovou funkcí K_P .

Proces vývoje řídicí jednotky ve zpětnovazebním systému lze s výhodou popsat pomocí V-modelu, jak je znázorněno na obr. č. 3.3. Na základě požadavků se v rámci etapy návrhu vytvoří model regulátoru, který je verifikován ve spojení s modelem řízeného objektu. Tato verifikace se nazývá validace návrhu (angl.: design validation) a také *testování modelu ve smyčce* (MiL – angl.: model in the loop).

Model regulátoru se následně provozuje na univerzálním hardwaru kompaktních rozměrů schopným provozovat model v reálném čase. Implementace algoritmů se pak může testovat ve spojení s reálným objektem, jak je znázorněno na obr. č. 3.3 u etapy implementace. Na základě modelu je následně vygenerován nebo naprogramován kód pro nasazení ve skutečném hardwaru řídicí jednotky. Při generování kódu z modelu se

pak hovoří o tzv. *rychlém vývoji* (angl.: rapid prototyping). Zároveň s vývojem algoritmů probíhá návrh, výroba a rovněž verifikace hardwaru řídicí jednotky.



Obr. č. 3.3 V-model procesu vývoje s vyznačením použití skutečné/simulované řídicí jednotky (regulátoru) a skutečného/simulovaného řízeného objektu v rámci jednotlivých etap vývoje

Zalomení do „V“ je v tomto případě v etapě nasazení vytvořeného softwaru do skutečného hardwaru řídicí jednotky (angl.: deployment), na rozdíl od etapy implementace, jak bylo zmíněno v předcházející kapitole u čistě softwarového vývoje. Tedy v místě spojení výsledků vývoje elektroniky a algoritmů.

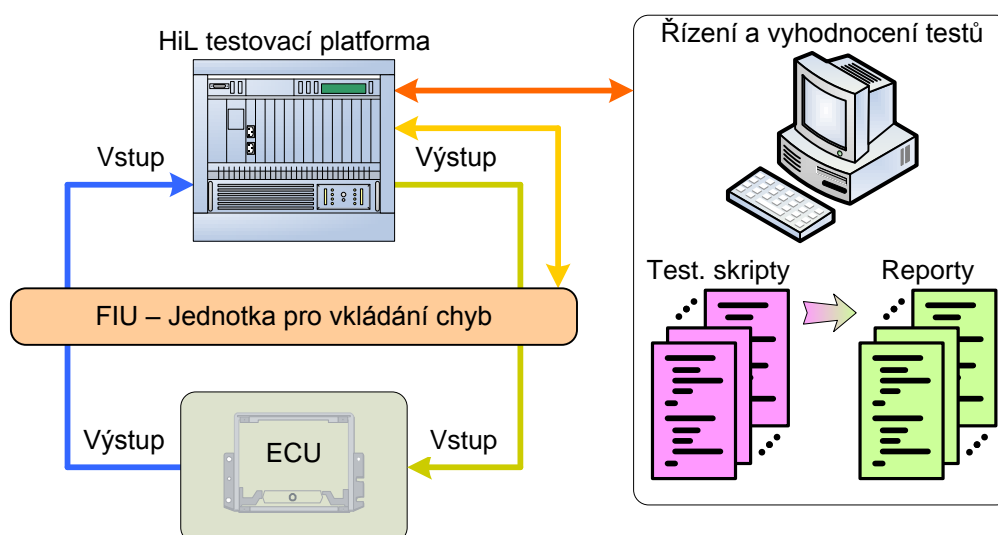
Skutečná jednotka je pak testována (angl.: ECU validation) ve spojení s modelovaným okolím, tedy pracuje jako *hardware v uzavřené smyčce* (HiL), proto je tak označena i příslušná etapa vývoje. Model okolí se provozuje na tzv. HiL testovací platformě (viz dále kapitola 3.2), což je v principu hardware pracující v reálném čase, nicméně ve srovnání s univerzálním real-time hardwarem pro provoz modelu regulátoru zmíněným výše je mnohem robustnější co se počtu vstupů/výstupů i výpočetního výkonu týká.

V další etapě označené jako integrační testování se testuje součinnost řídicích jednotek propojených datovou sítí a tvořící distribuovaný řídicí systém, buď jako HiL nebo v reálném systému v tomto případě tedy automobilu, označované jako testování systému (angl.: system testing) nebo *testování na palubě* (angl.: on-board testing).

Výrobní testování již nesouvisí s vývojem řídicí jednotky nebo řídicího systému automobilu, ale se sériovou výrobou, kdy se provádí testování na konci výrobní linky (EOL testování).

3.2 Princip HiL testování

Na následujícím obr. č. 3.4 je znázorněn základní princip HiL testování. Je zde znázorněn testovaný systém – elektronická řídicí jednotka (ECU), která je svými vstupy a výstupy připojena ve smyčce na testovací systém tvořený HiL testovací platformou ve spojení s počítačem typu PC. Mezi testovací platformou a řídicí jednotku může být někdy zařazena jednotka pro vkládání chyb (FIU – angl.: failure insertion unit), která většinou pomocí relé umožňuje simulovat poruchy typu rozpojení a zkrat na záporný nebo kladný pól napájení, což vyžaduje nejméně jeden rozpínací a dva spínací nezávislé kontakty na jeden ovlivňovaný vstup/výstup. Pro simulaci chyby typu zkrat mezi vybranými vodiči, se využívá dalších kontaktů připojující vybrané vodiče na sběrnice vedení. Tím lze docílit zkratu mezi více vodiči najednou, nicméně na druhou stranu je počet sběrnicových omezen, poněvadž by jinak neúnosně narůstal počet potřebných spínacích kontaktů (relé), které FIU musí obsahovat.



Obr. č. 3.4 Princip HiL testování

Princip testování je zhruba následující. Testovací systém stimuluje prostřednictvím podnětů vstupy řídicí jednotky a zachytává odezvy na výstupech. Pokud jsou předmětem testování zasíťované řídicí jednotky, které spolu komunikují prostřednictvím nějaké datové sítě (např. sběrnice CAN, LIN, apod.), tak testovací systém rovněž provádí sledování této komunikace, eventuelně vysílá zprávy za nepřipojené řídicí jednotky, které jsou tímpádem nahrazeny simulací (angl.: rest bus simulation). Komunikační sběrnici lze využít jako komunikační kanál pro tzv. White-Box testování, pro čtení nebo nastavení stavu jednotky pomocí zvláštních funkcí

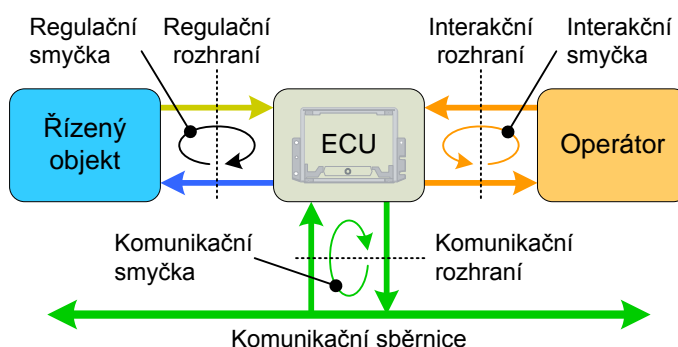
implementovaných v programovém vybavení, které jsou pro sériovou výrobu deaktivovány. Z důvodu zachování časové souslednosti je vhodné, aby testovací platforma přiřadila ke všem událostem (vydaný podnět, zachycená odezva nebo přijatá zpráva ze sběrnice) určitou časovou značku. Časové značky pak mimo jiné usnadňují vyhodnocování testování a měření např. doby odezvy.

Pro časování v reálném čase je nutné, aby testovací systém byl schopen provozu jako tzv. real-time hardware. Samotný počítač typu PC s operačním systémem Microsoft Windows je pro tento účel nepoužitelný právě z důvodu nemožnosti zajistit přesnější časování jednotlivých operací, poněvadž dokáže zajistit časování v řádu desítek až stovek ms. Pro testování je v praxi běžně požadováno časování 1ms a kratší. Proto je v rámci testovacího systému mezi počítač PC a testovaný systém zařazena HiL testovací platforma, která tvoří rozhraní pro přímé napojení na testovaný systém – disponuje potřebnými vstupy/výstupy a komunikačními rozhraními, a dále pak zajišťuje požadované časování všech událostí.

3.3 Implementace HiL testování

HiL testovací platforma může být implementována např. jako jednoduché rozhraní mezi testovanou řídicí jednotkou a počítačem PC, které zajišťuje generování podnětů, zachytávání odezev, sledování případné komunikace a časování všech těchto operací. Řízení a vyhodnocování testování pak provádí počítač PC, viz obr. č. 3.4.

Příklad takového rozhraní jako testovací platformy je uveden dále v kapitole 6.1.



Obr. č. 3.5 *Komplexní řídicí systém a jeho rozhraní*

Pokud je však předmětem testování řídicí jednotka, která představuje komplexní řídicí systém, např. jaký je znázorněn na obr. č. 3.5, pak je nutné, aby testovací systém byl schopen uzavřít všechny smyčky pro zajištění normální funkce řídicí jednotky. Uzavření smyček lze realizovat např. modelem okolí řídicí jednotky a to většinou pro

každou smyčku odděleně. Testovací systém pak simuluje model okolí v reálném čase. V případě použití počítače PC s příslušným rozhraním, pak musí tyto simulační výpočty provádět tento počítač. Vzhledem k náročnosti simulačních výpočtů na výpočetní výkon je v některých případech výhodné přesunout některé simulační výpočty (zejména ty, které se týkají regulační smyčky) do HiL testovací platformy. To lze provést v případě, kdy HiL testovací platforma je implementována tak, že disponuje vlastním výpočetním výkonem, což je většina dodávaných profesionálních systémů pro testování (viz kapitola 6.2).

Při rozhodování, jakou implementaci testovacího systému zvolit – zda zakoupit profesionální tester nebo vyvinout vlastní rozhraní k počítači PC, jsou kromě technických parametrů rozhodující zejména ekonomické a časové aspekty. Profesionální testery jsou univerzální, dají se dobře programovat, modely se (většinou) implementují pomocí robustního, ale komerčního, nástroje Matlab/Simulink, ovšem pořizovací náklady jsou značné – v řádech milionů Kč. Naproti tomu implementace vlastního testovacího rozhraní má zdánlivě nižší náklady. Ale pokud je však doba vývoje příliš dlouhá, mohou náklady dosáhnout úrovně nákladů profesionálního zařízení. Navíc doba implementace je prodloužena právě o dobu vývoje. Výběr testovacího systému je dále rozebrán v kapitole 6 a implementace modelů okolí pro vybrané řídicí jednotky pak v kapitole 7. Další informace o této problematice lze nalézt také v [21] a [22].

4. Modelování dynamických systémů

4.1 Úvod do modelování systémů

Modelování systémů je jedna z metod lidského poznání. Různými prostředky, např. slovním popisem, graficky, matematicky, ale i fyzikálně a technicky realizovanými modely, se člověk snaží popsat věci a jevy, které se nacházejí v jeho okolním prostředí a které pozoruje. Produktem jeho úvah jsou určité systémy – modely zavedené nejčastěji abstraktním způsobem. Modely vyjadřují a znázorňují podstatné vlastnosti reálných systémů – objektů, které jsou předmětem zájmu člověka. Většina modelů je výsledkem aktivní a tvořivé činnosti člověka a téměř vždy existuje dvojice *model – reálný systém*. Z tohoto hlediska rozdělujeme modely na dvě hlavní skupiny:

1. Modely, které umožňují analyzovat reálné systémy, tedy konkretizovat a zpřesňovat naše představy o již existujících systémech. Jsou to např. modely atomu, biologických struktur či procesů, ale i elektronických systémů a zařízení a fyzikálních procesů. Z těchto modelů často vzniká úloha analýzy – identifikace objektů, jejich vlastností a struktury.
2. Modely, které vznikly v důsledku projektování a návrhu, přičemž realizace modelovaných zařízení je ve fázi plánování. Jedná se o úlohy syntézy, projektování a konstruování. Tato činnost je dnes výhradně podporována výpočetní technikou, jedná se o metody počítačem podporovaného projektování, konstruování, výroby, atd. (označuje se zkratkami CAE, CAD, CAM, apod.).

Analýza a syntéza jsou vzájemně spjaté činnosti a neustále se v tvořivé práci člověka střídají a doplňují.

Každá vědecká a technická disciplína má při tvorbě modelů své specifické problémy. Je třeba vždy vycházet z důsledných znalostí (fyzikálních, chemických, biologických a dalších) příslušného oboru. Modelové představy a vztahy o reálném systému nebo studovaném jevu se nejčastěji získávají bilancemi množství, sil, resp. jiných zavedených veličin. Konečným cílem abstrakce modelu je realizace modelu na počítači, tedy vytvoření *počítačového modelu*.

Hranice mezi reálným systémem a jeho počítačovým modelem se do jisté míry prolínají. Někdy je vhodné, aby model reprezentoval reálný systém tak věrně, aby mohl

být využitý např. pro nastavení parametrů řídicího systému, který má řídit modelovaný systém.

Důležitým aspektem je požadavek, aby se s modelem dalo experimentovat a to i v reálném čase. Proces experimentování, ve kterém jsou zkoumány vlastnosti modelu nebo jeho interakce s okolním prostředím, může být jediný způsob, jak získat alespoň kvalitativně nové poznatky o *dynamice* a *struktuře* analyzovaného nebo projektovaného systému.

Proces experimentování s modelem se nazývá simulace. Simulační experimenty umožňují hledání alternativ a vhodných parametrů projektovaného zařízení, případně umožňuje poopravit nebo doplnit poznatky o systému, který je analyzován.

Pokud není znám realizovatelný algoritmus nalezení optima, simulace zpravidla neumožňuje přímé určení optimálního řešení – nalezení optimálních parametrů. Ty je třeba hledat metodami *optimalizace*, jako jsou lineární nebo dynamické programování.

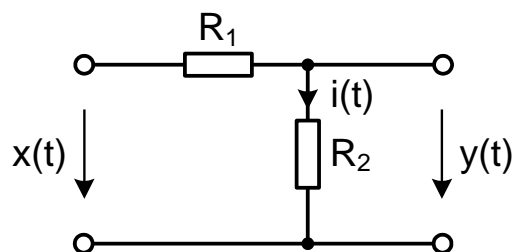
4.2 Základní pojmy vytváření modelů

Na následujících příkladech je nastíněna metodika postupu při tvorbě modelů statických i dynamických systémů, jakož i logických (číslicových) systémů. Jsou rovněž zmíněny základní pojmy, které se při modelování i popisu systémů používají, a jsou také uvedeny ukázky implementace modelů v prostředí Matlab – Simulink, případně Stateflow.

4.2.1 Model statického systému

Příklad 4.1: Ideální odporový dělič napětí

Neuvažujeme-li vliv parazitních kapacit a indukčností a předpokládáme-li ideální zdroj a nezátížený výstup, dostaneme statický dvojbran podle obr. č. 4.1.



Obr. č. 4.1 Odporový dělič jako statický systém

Platí následující vztah:

$$(4.1) \quad y(t) = \frac{R_2}{R_1 + R_2} \cdot x(t)$$

Kde $x(t)$ je vstupní veličina a $y(t)$ je výstupní veličina – obě jsou funkcemi času.

Když $t = nT$, kde T je konstanta a n je celočíselná proměnná, $n = 0, 1, 2, \dots$, tak také platí:

$$(4.2) \quad y_n = \frac{R_2}{R_1 + R_2} \cdot x_n, \quad y_n = y(nT), \quad x_n = x(nT)$$

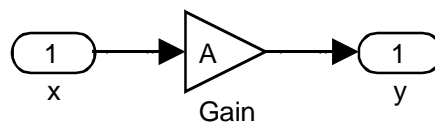
Což je matematický zápis diskrétního modelu statického systému. V tomto modelu nejsou kromě vstupních a výstupních veličin žádné další veličiny.

Pro statický časově spojitý, resp. diskrétní případ obecně platí:

$$(4.3) \quad \begin{aligned} \mathbf{x}(t) &= \mathbf{A} \cdot \mathbf{y}(t) \\ \mathbf{x}_n &= \mathbf{A} \cdot \mathbf{y}_n \end{aligned}$$

Kde $\mathbf{y}(t)$, \mathbf{y}_n jsou vektory v čase spojitých, resp. diskrétních výstupních veličin, $\mathbf{x}(t)$, \mathbf{x}_n jsou vektory v čase spojitých, resp. diskrétních vstupních veličin a \mathbf{A} je matice konstant určující statický přenos systému. Rozměry vektorů a matice jsou dány konkrétním případem.

Na následujícím obrázku je znázorněn blokový model statického systému popsaný rovnicí (4.3), vytvořený v prostředí Simulink [11], které je součástí systému pro vědecké výpočty Matlab [10] od společnosti MathWorks [9].



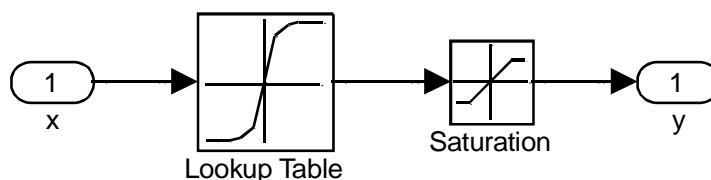
Obr. č. 4.2 Blokový model statického systému v prostředí Simulink

Matlab standardně předpokládá, že proměnné jsou obecné vektory. Pro skalární proměnnou používá matici o rozměru 1×1 . Proto je nutné ve vlastnostech bloku Gain nastavit způsob násobení vstupní veličiny konstantou příslušným způsobem. Tedy buď skalární (prvkové) násobení pro jednorozměrné veličiny x , y a parametr A , nebo vektorové pro vektory \mathbf{x} , \mathbf{y} a matici \mathbf{A} .

Model podle obr. č. 4.2 představuje lineární systém. Pro nelineární systém lze v Simulinku využít bloky pro různé druhy nelinearit nebo speciální blok s tzv.

vyhledávací tabulkou (angl.: lookup table), která umožňuje určit výstup pro libovolný vstup, ze známých hodnot závislosti výstupní veličiny na vstupní. K tomu se využívá metod interpolace a extrapolace. Dále je vhodné výstup doplnit blokem saturace, který zavede omezení oboru výstupních hodnot tak, aby se na výstupu neobjevily nereálné hodnoty, např. záporné otáčky – nevyjadřují-li však směr otáčení.

Na následujícím obr. č. 4.3 je znázorněn obecný model nelineárního statického systému s omezeným výstupem tak, jak se dá vytvořit v prostředí Simulink. Na obrázku je blok Lookup table (pro jednorozměrné veličiny), ve kterém je přibližně znázorněn průběh závislosti výstupu na vstupu (zde funkce tangens). Pro vícerozměrné vstupy (vektory) nabízí Simulink i příslušné vícerozměrné vyhledávací tabulky. V bloku omezení výstupu Saturation se nastavuje horní a dolní mez intervalu oboru výstupních hodnot.

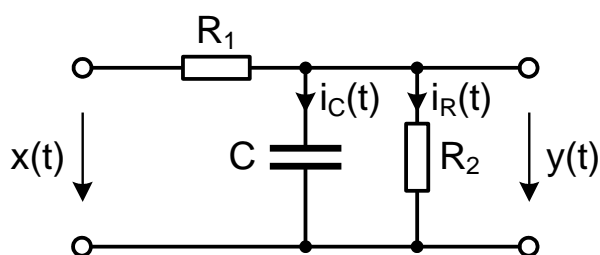


Obr. č. 4.3 Model obecného nelineárního statického systému

Blok Saturation může být umístěn i na vstupu tak, aby správně vymezil definiční obor vstupních hodnot systému.

4.2.2 Model dynamického systému

Příklad 4.2: Integrační článek – obvod RC



Obr. č. 4.4 Obvod RC jako dynamický systém

Mějme dvojbran podle obr. č. 4.4 (opět za ideálních předpokladů), který představuje dynamický systém 1. řádu, pak jeho přenosová funkce (někdy jen přenos) má následující tvar:

$$(4.4) \quad S(p) = \frac{K}{p \cdot T_i + 1}, \quad S(p) = \frac{y(p)}{x(p)}$$

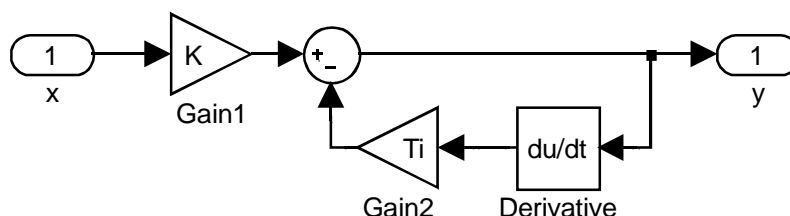
Kde p je komplexní proměnná integrální transformace vyjadřující frekvenční závislost (spektrální pohled), $T_i = \frac{R_1 R_2 C}{R_1 + R_2}$ je časová konstanta a $K = \frac{R_2}{R_1 + R_2}$ představuje zesílení, resp. statický přenos systému.

Diferenciální rovnice systému má tvar:

$$(4.5) \quad T_i \frac{dy(t)}{dt} + y(t) = K \cdot x(t), \quad y(0)$$

Kde $y(0)$ je počáteční podmínka pro řešení této diferenciální rovnice.

V Simulinku lze model takového systému vytvořit přímo z diferenciální rovnice (4.5), jak je znázorněno na následujícím obr. č. 4.5.

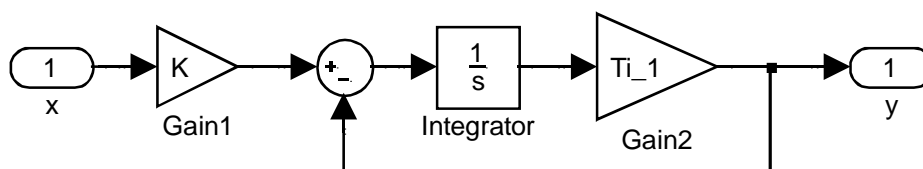


Obr. č. 4.5 Model dynamického systému vycházející z diferenciální rovnice

Z důvodu implementovaných metod numerických výpočtů v prostředí Simulink, resp. Matlab je výhodnější upravit diferenciální rovnici do integrálního tvaru a použít blok integrace. Obecně numerické integrování je snazší a přesnější než derivování. Rovnice v integrálním tvaru vypadá následovně:

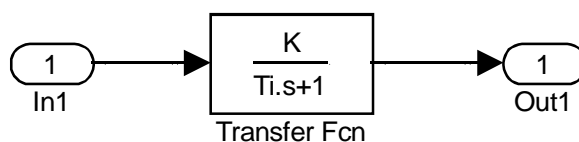
$$(4.6) \quad y(t) = \frac{1}{T_i} \int [K \cdot x(t) - y(t)] dt, \quad y(0)$$

Kde $y(0)$ je počáteční podmínka, kterou lze snadno zadat do bloku integrace. Model má pak strukturu znázorněnou na následujícím obrázku.



Obr. č. 4.6 Model dynamického systému podle rovnice v integrálním tvaru

Známe-li přenosovou funkci, podle rovnice (4.4), lze pro vytvoření modelu využít blok Transfer Fcn. Simulink využívá pro označení proměnné integrální transformace symbol s , viz obr. č. 4.7.



Obr. č. 4.7 Model dynamického systému podle přenosové funkce

4.2.3 Model stavového automatu

Příklad 4.3: Sekvenční logický obvod

Automat, který modeluje chování sekvenčního logického obvodu, je uspořádaná pětice $(X, S, Y, \delta, \lambda)$, kde

X je konečná neprázdná množina symbolů, jejíž prvky se nazývají *vstupy* automatu,

S je konečná neprázdná množina symbolů, jejíž prvky se nazývají *stavy* automatu,

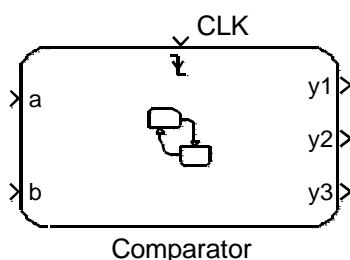
Y je konečná neprázdná množina symbolů, jejíž prvky se nazývají *výstupy* automatu,

δ a λ jsou zobrazení – přechodová funkce a výstupní funkce automatu:

$$(4.7) \quad \begin{aligned} \delta : S \times X &\rightarrow S \\ \lambda : S \times X &\rightarrow Y, \quad \text{nebo} \quad \lambda : S \rightarrow Y \end{aligned}$$

Chování automatu můžeme popsat tabulkou nebo grafem – nazývaný *stavový diagram*, což jsou modely popisující chování logického (číslicového) systému.

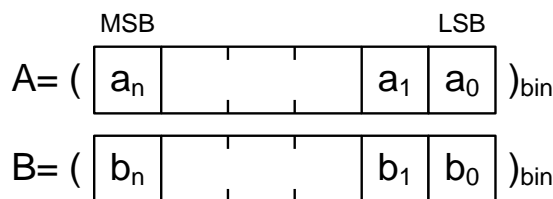
Mějme např. sériový synchronní komparátor dvou binárních čísel A, B znázorněný na následujícím obr. č. 4.8.



Obr. č. 4.8 Blok sekvenčního logického obvodu – sériový synchronní komparátor

Komparátor pracuje následovně: V čase t_0 se porovnají nejnižší bity (LSB) a_0, b_0 , v čase t_1 se porovnají vyšší bity a_1, b_1 , atd., až v čase t_n se porovnají nejvyšší bity

(MSB) a_n, b_n a získá se výsledek. Čísla A, B jsou zakódována ve stejných registrech způsobem znázorněným na následujícím obr. č. 4.9.

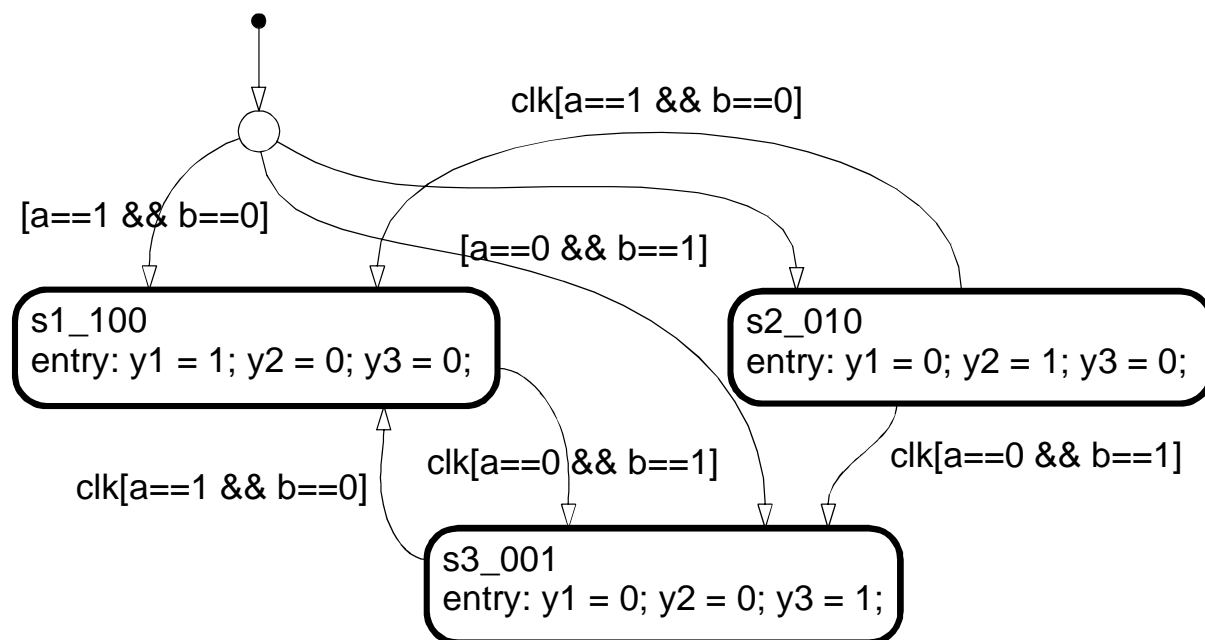


Obr. č. 4.9 Kódování binárních čísel v registrech

Pro hodnoty výstupů pak platí následující výrazy (4.8) a je zřejmé, že zobrazení stavu na výstup λ je přímé, podle (4.7) – druhá varianta.

$$\begin{aligned}
 (4.8) \quad y_1 &= 1 \Leftrightarrow A > B \\
 y_2 &= 1 \Leftrightarrow A = B \\
 y_3 &= 1 \Leftrightarrow A < B
 \end{aligned}$$

Množina stavů (stavový prostor) je $S = \{s_1, s_2, s_3\}$, někdy se kóduje jen jako $S = \{1, 2, 3\}$. CLK (na obr. č. 4.8) je vstup hodinových impulzů, které zabezpečují synchronizaci a na každou sestupnou hranu dojde k porovnání bitů příslušného řádu čísel A a B .



Obr. č. 4.10 Stavový diagram sériového synchronního komparátoru

Na předcházejícím obr. č. 4.10 je znázorněn stavový diagram sériového synchronního komparátoru jako konečného automatu, vytvořený v prostředí Stateflow [12], které je součástí Matlabu a je těsně navázáno na Simulink. Ostatně komparátor na

obr. č. 4.8 je blok Simulinku, za kterým se skrývá právě stavový diagram Stateflow určující chování logického systému.

V diagramu jsou znázorněny tři stavy, pojmenované s1_100, s2_010 a s3_001. V těchto stavech je definováno, že při vstupu do nich se příslušným způsobem nastaví hodnoty výstupů, viz konstrukce „entry“. Dále jsou zde definovány přechody mezi jednotlivými stavy, které jsou vázány na hodinový signál clk a stav vstupů a, b. Výchozí přechod je naznačen černou tečkou a je rozvětven, aby se automat dostal do správného výchozího stavu.

V uvedených příkladech, dále např. v [7], vystupují do popředí určité společné strukturální i dynamické rysy. V každém systému můžeme rozlišovat dva typy proměnných nebo veličin.

1. *Nezávislé proměnné*, reprezentující čas (v širším pojetí i prostorovou souřadnici).
2. *Závislé proměnné* (dynamické veličiny), představující vstupní, výstupní a stavové veličiny, od kterých se odvozují další informace.

Existuje-li v systému a jeho modelech nezávislá proměnná, kterou fyzikálně vždy interpretujeme jako čas, jedná se o *dynamický systém* a jeho dynamický model. Dalším atributem dynamických systémů je, že v jejich struktuře se nachází *paměť* – jako podsystém na zapamatování si informací. A to i u spojitých dynamických systémů, kde je pojem paměti zdánlivě skrytý – paměť je reprezentována akumulátorem nějaké formy energie.

Je výhodné (zejména u distribuovaných systémů) uvažovat jako nezávislou proměnnou i prostorovou (geometrickou) souřadnici. Pak se jedná o dvou- či více-rozměrný systém, kde jednou z nezávislých proměnných je vždy čas.

4.2.4 Modelovací jazyk Modelica

V předchozích příkladech jsou kromě matematického vyjádření zmíněny ukázky modelů v Simulinku. Nicméně Simulink je uzavřený proprietární produkt firmy MathWorks. Je zde tedy vhodné zmínit i alternativní možnost modelování pomocí otevřeného nástroje. Tím je otevřený objektově orientovaný univerzální modelovací jazyk Modelica. Jazykem Modelica lze pomocí diferenciálních, algebraických a diskrétních rovnic popsat dynamické chování fyzikálních (technických) systémů. Tyto

systemy mohou obsahovat elektrické, mechanické, pneumatické i hydraulické součásti a rovněž vyjádření fyzikálních procesů jako jsou tepelné, mechanika tekutin (proudění) a to včetně řízení pomocí regulátoru.

Modelica je sice otevřená, její specifikace je volně dostupná online [13], ale je to jen jazyk. Pro modelování je potřeba prostředí, ve kterém bude možné modely vytvářet ať už textově zápisem zdrojového kódu nebo graficky, provádět jejich simulaci a získat výsledky v požadované formě. Mezi nejznámější otevřené simulační prostředí patří zřejmě OpenModelica [14]. Modely popsané v Modelice lze však zahrnout i do modelů v prostředí Scilab/Scicos, viz kapitola 6.3.1, eventuálně jako blok do Simulinku.

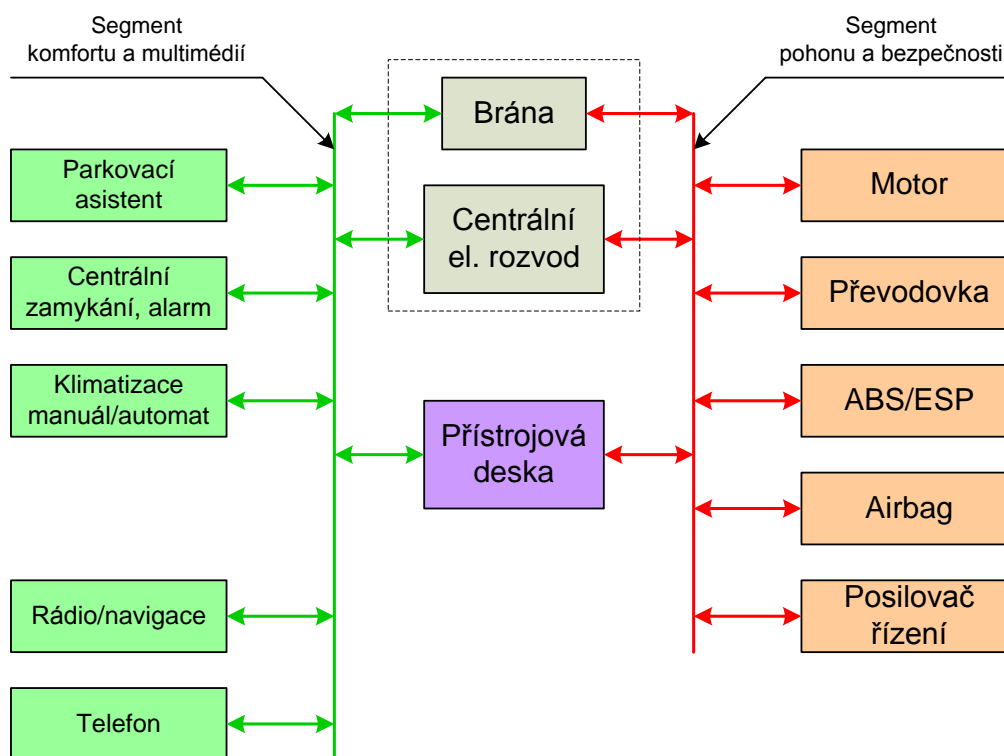
5. Elektronické řídicí systémy současných automobilů

V této kapitole se nachází popis struktury řídicích systémů dnešních automobilů. Nejdříve je věnována pozornost struktuře řídicího systému automobilu nižší třídy. Pak následuje ukázka struktury řídicího systému pro automobil vyšší kategorie, která je doplněna nejnovějšími funkcemi a systémy, které jsou momentálně v různém stádiu vývoje. Popisu řídicích jednotek, které byly předmětem testování v rámci této práce, je dále věnována samostatná podkapitola.

5.1 Struktura řídicího systému automobilu nižší třídy

V dnešních vozech se řídicí systém skládá někdy i z desítek řídicích jednotek, které zajišťují nejrůznější funkce vozidla. Jednotky spolu navzájem komunikují, vyměňují si např. informace od svých senzorů, které jsou důležité i pro další jednotky. Ke komunikaci slouží datová síť, v dnešní době je nejrozšířenější sběrnice CAN. Vzhledem ke komplexnosti datové sítě, objemu komunikace a také s ohledem na bezpečnost řídicího systému je síť rozdělena na segmenty, které jsou navzájem propojené přes bránu tak, aby určité informace procházely z jednoho segmentu do ostatních.

Dvousegmentová datová síť – sběrnice CAN



Obr. č. 5.1 Struktura datové sítě řídicího systému automobilu nižší třídy

Síť je segmentována podle funkcí, které má zajišťovat. Segmentace sítě typická pro vůz nižší třídy koncernu Volkswagen (Škoda Auto), který má cca 15 řídicích jednotek v maximální výbavě, je znázorněna na obr. č. 5.1. V této třídě vozu je datová síť rozdělena zpravidla na dva segmenty.

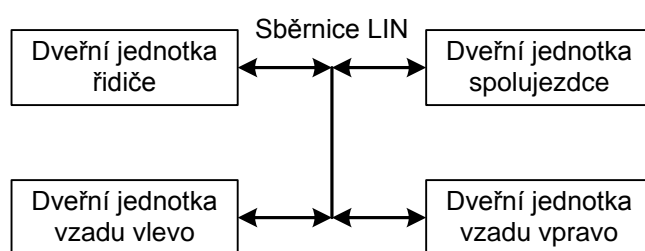
Hlavní segment obsluhuje funkce pohonu a aktivní bezpečnosti. Na tomto segmentu je připojena řídicí jednotka motoru, převodovky (v případě automatické převodovky), posilovače řízení, brzdový asistent (ABS nebo ESP), řídicí jednotka airbagů a podobně.

Komfortní a tzv. multimediální funkce jsou dvě skupiny funkcí, u vozů nižší třídy však bývají zajišťovány jedním segmentem sběrnice. Mezi komfortní řídicí jednotky lze zahrnout řídicí jednotky asistenta parkování, manuální či automatické klimatizace, alarmu a centrálního zamykání a také mnohé z funkcí, které zajišťuje řídicí jednotka centrálního elektrického rozvodu (jako stěrače, vyhřívání oken a zpětných zrcátek, ukazatele změny směru, vnitřní osvětlení, atd.).

Naproti tomu multimediální část tohoto segmentu tvoří rádio nebo navigační systém, rozhraní pro připojení mobilního telefonu, apod. – v tomto případě se pojem řídicí jednotky příliš nepoužívá, protože řídicí jednotka jako taková pouze řídí, má k sobě z vnějšku připojené senzory a akční členy, zatímco rádio či navigační systém tvoří kompaktní celek.

Segmentaci se v tomto případě vymykají řídicí jednotka centrálního elektrického rozvodu a přístrojová deska, které zpracovávají informace z obou segmentů. Navíc centrální rozvod spolu s bránou propojující jednotlivé segmenty sítě jsou umístěny v jedné skřínce.

Mezi komfortní řídicí jednotky lze zařadit i řídicí jednotky pro elektrické stahování oken. Ty však u vozů společnosti Škoda Auto nižší třídy tvoří autonomní datovou síť (viz. obr. č. 5.2), která není vodivě ani informačně spojena s hlavní datovou sítí vozu. U jiných výrobců bývají tyto jednotky zařazeny na komfortním segmentu datové sítě.

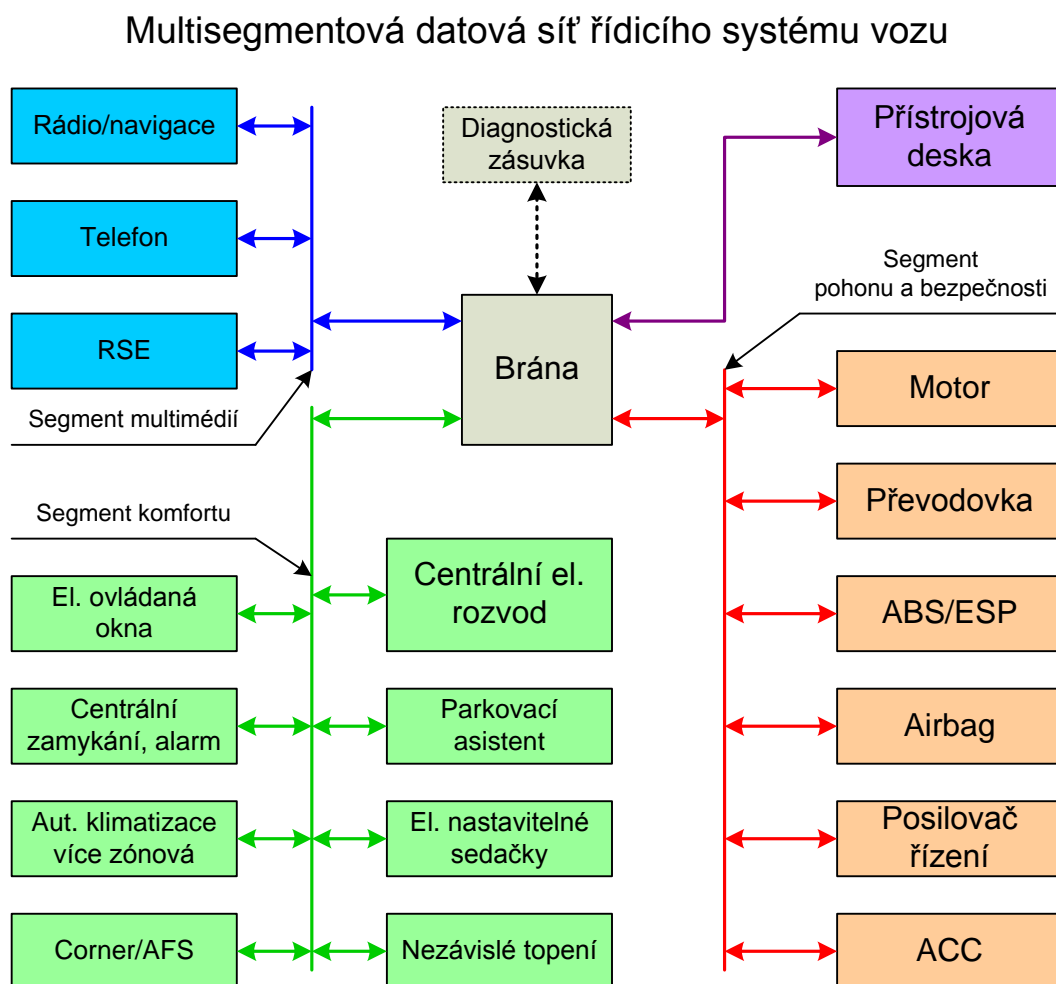


Obr. č. 5.2 Datová síť dveřních řídicích jednotek pro stahování oken

Segmenty datové sítě se liší přenosovou rychlostí i fyzickou úrovní signálů na datových vodičích. Proto je nutná brána na jejich propojení. Vzhledem k narůstající vytiženosti datové sítě s přibývajícimi funkcemi, jak u vozů nižší třídy, tak zejména u vozů střední a vyšší třídy, dochází k vícečetné segmentaci.

5.2 Struktura řídicího systému automobilu vyšší třídy

Ve vyšších kategoriích automobilů se řídicí systém skládá z několika desítek řídicích jednotek, typicky 50 až 80 jednotek. Komplexnost zajišťovaných funkcí a s tím související vytiženost datové sítě systému dosahuje takové úrovně, že je nutné použít tzv. multisegmentovou datovou síť, kdy je zpravidla oddělena komfortní a multimediální část, a také přístrojová deska má svůj segment sběrnice, jak je znázorněno na obr. č. 5.3.



Obr. č. 5.3 Struktura datové sítě řídicího systému automobilu vyšší třídy

Rovněž segment pohonu se dostává k takové přenosové náročnosti, že přestávají stačit možnosti sběrnice CAN, a tak se postupně začínají uplatňovat i jiné výkonnější sběrnice, např. FlexRay [27].

5.2.1 Vlastnosti a funkce vybraných subsystémů

V této kapitole jsou blíže rozebrány vlastnosti a funkce vybraných částí řídicího systému současného automobilu vyšší třídy. Podrobný popis všech vlastností a funkcí celého systému by vydal na samostatnou publikaci. Rovněž základní funkce jednotlivých částí systému jsou dnes všeobecně známé. Proto jsou zde vybrány jen některé vlastnosti a funkce, které ve většině případů využívají součinnost více částí elektronického řídicího systému automobilu. Popis je rozdělen podle segmentů datové sítě, ke kterým jsou jednotlivé řídicí jednotky připojeny, nicméně mnohé funkce jsou realizovány spoluprací více jednotek na různých segmentech.

Segment pohonu a bezpečnosti

Jako řídicí subsystém s největším nárůstem vlastností a funkcí, v poslední době, je nutné uvést *řídicí jednotku brzd*. Ta zajišťuje kromě základních funkcí ABS a ESP rovněž řadu asistenčních funkcí, jako MSR, ASR, EDS, které usnadňují jízdu a ovládání vozu v běžných i nestandardních situacích. Vzhledem k tomu, že řídicí jednotka brzd má velmi přesnou informaci o otáčení jednotlivých kol, díky indukčnostním nebo tzv. Hallovým snímačům, poskytuje zpracováním těchto informací nejen hodnotu *rychlosti vozidla* pro ostatní jednotky, zejména pro jednotku přístrojové desky, ale tyto informace se dají využít i např. k indikaci *sníženého tlaku v pneumatice*.

Nejdůležitější součástí řídicího systému vozu je z pohledu bezpečnosti posádky *řídicí jednotka Airbagů*. Ta disponuje snímači kinematických veličin, jako jsou více osé akcelerometry a gyroskopy, díky kterým pozná, že *došlo k nárazu*, ať už do překážky při pohybu nebo naopak nastal náraz od jiného vozidla, a dále *sleduje pohyb vozidla* po nárazu a na základě rychlého vyhodnocení signálů ze snímačů rozhodne o tom, jaké vaky aktivuje a v jakém pořadí. Mezi akční členy patří kromě *vaků* (přední, boční, hlavové, kolenové a další) také *předpínače bezpečnostních pásů* a *aktivní hlavové opěrky*. Při nárazu řídicí jednotka vyšle na sběrnici tzv. „Crash signál“, který způsobí reakci ostatních částí řídicího systému. Dojde k odpojení palivového čerpadla, zapnutí výstražných ukazatelů, odemčení vozidla, rozsvícení vnitřního osvětlení a v některých případech, pokud je touto funkcí vozidlo vybaveno, na závěr k odpojení baterie. Čímž

se sice odstaví všechny elektrické systémy, ale zabrání se případnému zkratu, který by mohl způsobit požár havarovaného vozu.

Poslední zde zmíněnou částí připojenou k segmentu pohonu a bezpečnosti je tzv. *adaptivní tempomat ACC* (angl.: automatic cruise control). Tato řídicí jednotka na základě informací o *délce* mezery před vozidlem získaných buď z vlastních snímačů, případně ve spolupráci s parkovacím asistentem, ovlivňuje *řídicí jednotku motoru a brzd* tak, aby se udržovala stálá bezpečná vzdálenost od vozidla, které je vpředu. Tedy například při jízdě v koloně na dálnici, či ve městě, a to až do úplného zastavení.

Segment komfortu

Řídicí jednotka *parkovacího asistenta* zajišťuje kromě základní funkce *indikace vzdálenosti* od překážky vzadu i vpředu (PDC), také funkci asistenta při *parkovacím manévru* do podélné řady vozidel (PLA). K řídicí jednotce jsou připojeny inteligentní snímače vzdálenosti pracující na principu měření vzdálenosti ultrazvukem. Informace jsou předávány speciálním protokolem. Počet snímačů je minimálně 4 a to jak vzadu tak i vpředu, to proto, aby jednotka vyhodnotila nejen *souvislou* překážku pod různým úhlem vzhledem k vozidlu, např. *zeď*, ale i *tyčovou* překážku, např. svislou dopravní značku. Vzdálenost od překážky je indikována na *přístrojové desce* na displeji a také *akustickým signálem*, kdy se změnou vzdálenosti v daných intervalech se mění frekvence přerušování akustického signálu. V případě PLA je navíc doplněn snímač pro měření vzdálenosti mezi podélně parkujícími vozidly, pracující většinou na principu radaru. Funkce asistenta je následující. Po aktivaci během souvislé přiměřeně pomalé jízdy podél zaparkovaných vozidel provádí měření *vzdálenosti* mezi těmito vozidly. Po zaměření dostatečně velké mezery vyzve řidiče k zastavení a zařazení zpátečky (pokud je vozidlo vybaveno automatickou převodovkou většinou asistent řadí sám). Následně vyzve řidiče, aby pustil volant a uvedl vozidlo do pohybu, a ve spolupráci s *řídicí jednotkou* posilovače řízení provede parkovací manévr optimální trajektorii. Technicky je možné, aby spoluprací stávajících subsystémů byl manévr proveden zcela automaticky. Nicméně z legislativních důvodů je nutné, aby jízdu vozidla kontroloval řidič, který případně zodpovídá za nehodu (jen velmi těžko lze obvinít automobil z toho, že někoho nebo něco přejel).

Řídicí jednotka *adaptivních světlometů* (AFS) s funkcí *Corner* je další z „komfortních“ jednotek, nicméně má nezanedbatelný vliv na bezpečnost jízdy. Poněvadž, pokud řidič za snížené viditelnosti dříve uvidí překážku, dokáže včas zareagovat. Funkcí AFS je kromě *natačení světlometů* podle úhlu natočení volantu tak,

aby světelný kužel svítil ve směru jízdy, také v závislosti na rychlosti a způsobu jízdy měnit *tvar kuželu světla*. Umožňuje vytvořit kromě standardních tvarů pro potkávací a dálková světla rovněž tvary pro jízdu po dálnici (svítí do větší vzdálenosti, ale neoslňuje protijedoucí vozidla), pro jízdu ve městě (svítí více do šířky, čímž osvětluje postranní chodník, případně chodce na přechodu), tvar světla pro odbočování, tzv. *Corner* (viz dále). Podrobněji viz také [28]. Pro funkci AFS je tedy nutné použít speciální světlomet s možností horizontálního natáčení a případně s proměnnou optikou, kdy např. pomocí clonek mění výsledný tvar světelného kuželu. Vertikálním natáčením dnes disponuje většina světlometů, pro nastavení výšky, resp. vzdálenosti kam světlo dopadne. V případě světlometů s xenonovými výbojkami, které mají řádově větší intenzitu a jinou barvu světla (nepříjemnou pokud posvítí do očí), je povinné automatické řízení výšky v závislosti na zatížení zadní nápravy, aby nedocházelo k oslňování řidičů protijedoucích vozidel. Snímač zatížení zadní nápravy je zpravidla potenciometrický a vyhodnocuje úhel kyvné části zadní nápravy vzhledem ke karoserii vozu. Naproti tomu funkce *Corner* je někdy realizována jako zvláštní funkce předních *mlhových světlometů*, které svítí nízko při zemi a více do šířky než potkávací světla. Řídicí jednotka vyhodnocuje informace o úhlu natočení volantu, rychlosti jízdy a stavu ukazatelů změny směru a podle předpokládaného směru zatáčení nebo odbočování rozsvítí příslušné mlhové světlo a tím dojde k osvětlení prostoru, kam se zatáčí, resp. odbočuje.

Dalším systémem, který se řadí mezi komfortní, je řídicí *subsystém nezávislého topení*. Tento systém se běžně vyskytuje již řadu let u nákladních automobilů a tahačů návěsů. Nezávislé topení umožňuje, zejména v zimním období, *spalováním paliva* z nádrže, tedy nafty nebo benzínu, předehřát chladicí okruh motoru, a tím nejen připravit provozní podmínky motoru ještě před nastartováním (výhodné zejména pro vznětové motory), ale i ve spojení s řídicí jednotkou klimatizace *předvytopit* prostor pro cestující a zejména *odmrazit* případně *odmlžit okna*. Naopak v létě pokud je vozidlo zaparkované na přímém slunci, tak aktivací *ventilátoru* dokáže udržet ve voze přijatelnou teplotu. Ovládání nezávislého topení je provázáno s řídicí jednotkou centrálního zamykání a lze jej díky tomu ovládat pomocí bezdrátového dálkového ovladače, který je součástí klíče zapalování. Je nutné podotknout, že systém nezávislého topení je výbavou na přání a je nabízen jen k vozům vyšších tříd. To je dáno zejména pořizovací cenou, která je řádově 35.000,- Kč. Systém nezávislého topení lze pro odvětrávání vozu v létě doplnit solárním panelem, který se umísťuje na střešní okno a

slouží k výrobě elektrické energie pro napájení ventilátoru topení/větrání. Ventilátor pak může běžet po celou dobu stání vozidla na přímém slunci a tím snížit teplotu uvnitř vozu až o 25°C^{*)}. Solární panel jako výbava vozidla je vázán zejména na přítomnost střešního okna, nikoliv přímo na systém nezávislého topení. Je tedy možné mít jen funkci odvětrávání v létě bez možnosti topení v zimě.

Kessy (angl.: Keyless entry and start system – Systém bezklíčového odemykání a startování) je komfortní systém, který se objevuje ve vozidlech v posledním desetiletí, např. výrobce Renault takový systém nabízí od 2. verze vozů Laguna, tedy od roku 2001, výrobce BMW od roku 2004. Systém spočívá v *bezklíčovém* způsobu *odemykání* a *startování* vozidla. Vozidlo se odemkne při pokusu o otevření za kliku dveří pouze tehdy, pokud ve své blízkosti detekuje bezpečnostní čip umístěný na kartě, která je zároveň vybavena tlačítky a vysílačem pro ruční ovládání centrálního zamykání, podobně jako dálkový ovladač s klíčem. Startování a zastavování motoru se pak provádí pomocí tlačítka na palubní desce, kdy pro odblokování startování se karta zasune do příslušného otvoru na palubní desce. Systém *Kessy* s sebou přináší řadu problémů. Zejména je třeba zajistit spolehlivou a bezpečnou identifikaci bezpečnostního čipu, resp. správně rozlišit oprávnění přístupu do vozidla a odblokování startování. Tyto bezpečnostní záležitosti je potřeba vyřešit pro zdárné nasazení tohoto systému.

Na systém *Kessy* lze v rámci komfortních systémů navázat systém tzv. *personalizace*, kdy se ke konkrétnímu bezpečnostnímu čipu přiřadí uložení nastavení např. poloh zrcátek a sedačky řidiče. Takovou funkci lze pak využít pro identifikaci řidiče zejména v situaci, kdy s vozidlem jezdí více řidičů, kteří používají různá nastavení zrcátek a sedačky.

Segment multimédií

Nejvýznamnější řídicí jednotkou na segmentu multimédií je bez pochyby *Radionavigační systém* (RNS – Radio Navigation System). Jedná se o radiopřijímač s CD přehrávačem v kombinaci s navigačním systémem nejčastěji standardu GPS. Navigační systém určuje polohu primárně podle družic. Nicméně prostřednictvím datové sítě rovněž zpracovává informace o rychlosti jízdy, natočení volantu a další, ze kterých dokáže upřesňovat polohu vozidla při zhoršeném signálu z družic, např. v husté městské zástavbě nebo v tunelech, apod. Modelově vyšší systémy jsou pak vybaveny měničem CD (nejčastěji na 6 disků) a samozřejmostí je rovněž přehrávání hudby ve

^{*)} Informace převzata z konfiguratoru vozidla Superb společnosti ŠKODA AUTO a.s., viz [cit. 2011-03-11] <<http://cc.skoda-auto.com/cze/pages/step1.aspx?carlinecode=68411>>.

formátu MP3. Nově se pak přístroje vybavují tunerem pro příjem digitálního rozhlasu DAB. Vzhledem k tomu, že RNS přístroj má poměrně velký barevný displej LCD mají nejdražší varianty možnost příjmu digitální televize DVB-T. Funkce televize je však z bezpečnostních důvodů během jízdy blokována, aby přijímaný program neodváděl pozornost řidiče. Existuje i přístroj vybavený speciálním dvojitým displejem, který při pozorování z různého úhlu (řidič, spolujezdec), zobrazuje různý obraz, např. navigační informace pro řidiče a televizní vysílání pro spolujezdce.

Další téměř nedílnou součástí segmentu multimédií je u vozů vyšší kategorie *rozhraní pro mobilní telefon*. Rozhraní je tvořeno v podstatě držákem telefonu, který se skládá ze dvou částí, pevné části, jež je součástí palubní desky, a výměnné části podle typu mobilního telefonu, a umožňuje integraci telefonu do elektronického systému automobilu. Pak např. audio systém slouží jako tzv. *handsfree* v souladu s legislativou upravující telefonování za jízdy. Telefon lze *ovládat* ovladači umístěnými poblíž volantu, kterými se ovládají i jiné funkce automobilu, jako palubní počítač, a na maticovém displeji mezi přístroji na palubní desce lze číst krátké textové zprávy přijaté telefonem. Propojení s telefonem je buď bezdrátové, standardu Bluetooth, nebo prostřednictvím výměnné části držáku a systémového konektoru telefonu. Držák pak umožňuje i nabíjení baterie a přivedení signálu z anténního systému automobilu. Někdy bývá mobilní telefon přímo součástí přístroje RNS, kdy se pak do otvoru v palubní desce vkládá jen tzv. SIM karta. Takové řešení nabízí např. výrobce Peugeot ve svém modelu 407.

Systémy, začínající se objevovat ve vozech nejvyšší kategorie v posledních letech a které se rovněž zařazují do komunikační sítě na segmentu multimédií, jsou systémy *zábavy na zadních sedadlech* (RSE – angl.: Rear Seat Entertainment). Zábava na zadních sedadlech je založena na dalších LCD zobrazovačích umístěných buď v opěrkách hlavy na předních sedadlech, nebo jako výklopné panely ze stropu, na kterých se zobrazuje obraz z DVD přehrávače nebo televizního vysílání, tedy z přístroje RNS. Podrobnější informace a přehled těchto systémů lze nalézt např. v této diplomové práci [37].

5.2.2 Systémy blízké budoucnosti

Systémy a jejich funkce uvedené v této kapitole jsou nyní u většiny velkých výrobců automobilů v různém stádiu vývoje a zatím zpravidla nejsou nasazeny v sériové výrobě.

Pokročilé funkce ESP

Řídicí jednotka ESP by ve spolupráci s posilovačem řízení mohla korigovat manévry tak, aby byla maximálně eliminována možnost vzniku smyku a aby automobil pokud možno projel požadovanou trajektorií.

Podobnou adaptivní funkci posilovače s názvem 4CONTROL již nabízí výrobce Renault u nové, aktuálně 3. verze vozů Laguna a Laguna Coupé. Systém 4CONTROL disponuje říditelnou zadní nápravou, kterou ovládá v závislosti na rychlosti a způsobu jízdy. Při nižší rychlosti se zadní kola natáčí proti předním, pro zlepšení ovladatelnosti a snížení poloměru otáčení vozu, naopak při vyšší rychlosti se zadní kola natáčí spolu s předními pro zlepšení průjezdu táhlým obloukem. Systém je provázán s ESP a svým zásahem zlepšuje také ovladatelnost vozu při náhlém manévru vyhýbání se překážce na silnici.

Aktivní bezpečnost chodců při nárazu

Aktivní i pasivní bezpečnost posádky automobilu lze dnes považovat za dobrou díky deformačním zónám karoserie, systémům airbagů, předpínačům pásů a aktivním opěrkám hlavy. V poslední době je však věnována pozornost tzv. aktivní ochraně chodců při nárazu. Tedy snížení následků srážky automobilu s chodcem např. pomocí protipohybu přední kapoty a zejména zabránění pádu sraženého chodce pod kola, např. pomocí nějakých sítí. Všechny tyto systémy jsou zatím ve stádiu prototypů, poněvadž, ač se během vývoje provádějí náročné simulace chování těchto systémů, v reálném nasazení zatím nedávají takové výsledky, aby mohly být nasazeny v sériové výrobě.

Systémy strojového vidění a zpracování obrazu

Další systémy, zde označené jako tzv. *vision systémy*, jsou založeny na strojovém zpracování, klasifikaci a rozpoznávání obrazu, tedy systémy strojového vidění (angl.: machine vision systems). Dále budou představeny některé z těchto systémů.

Elementární systém zpracování obrazu v automobilu, je systém pro *přenos obrazu z kamery* umístěné na zádi vozu k řidiči, např. na displej přístroje RNS. To usnadňuje manipulaci s vozem při parkování a couvání, zejména u vozů, u nichž je z konstrukčních důvodů velmi omezen výhled dozadu.

Systém, který nějakým způsobem již provádí zpracování obrazu za účelem získání informace, a nemusí se jednat o barevný obraz, podobný jaký vnímáme lidským okem, je systém pro *sledování vzdálenosti od postranní čáry*, respektive čar, na

vozovce. V případě, že se vozidlo při jízdě vychýlí z předpokládané dráhy, dané tvarem vozovky, tedy že se přiblíží k jedné nebo druhé čáře tak, že by hrozilo její přejetí, upozorní systém řidiče akustickým signálem. Účel takového systému je evidentní, a to zabránit mikrospánku řidiče při dlouhé monotónní jízdě např. na dálnici. Tento systém je založen na vyhodnocování *vzdálenosti* od čáry, respektive relativní polohy čáry vůči vozu a to na základě *prahování*, tedy nalezení hrany mezi řádově odrazivějším povrchem bílé čáry a zbytkem vozovky, což může být vyhodnocováno např. prostřednictvím infračerveného záření. Rozpoznání čáry je rovněž limitováno kvalitou povrchu vozovky a čáry a jejich případným znečištěním. Vzhledem ke stavu vozovek v naší zemi je praktická použitelnost omezena na některé úseky dálnic, výjimečně silnic první třídy.

Jiný je systém sledování čar na vozovce, který je založen na rozpoznávání obrazu z kamery umístěné na přední části vozu, kdy se v obrazu identifikuje tvar čar na vozovce, přesněji se opět prahováním rozlišuje odrazivý povrch od méně odrazivého pozadí. Na základě vyhodnoceného tvaru pak lze např. provést zásah do řízení tak, aby se vozidlo drželo pořád mezi identifikovanými čarami. Takové systémy autonomního řízení modelů vozidla byly prezentovány na veletrhu Embedded World 2009 v Norimberku a to společnostmi NEC a Atmel. Je nutné poznamenat, že systém společnosti NEC nevyňikal spolehlivostí, poněvadž model dost často vyjel z dráhy, v porovnání s modelem spol. Atmel, jehož řídicí systém byl distribuovaný jako ukázka možností sběrnice FlexRay, který sledoval dráhu bezchybně, nicméně pohyboval se řádově poloviční rychlostí. Lze tedy konstatovat, že spolehlivost takových systémů autonomního řízení vozidla je zatím nedostatečná pro nasazení v reálném provozu, kde se navíc vyskytují ostatní vozidla (účastníci provozu), znečištění a poškození vozovky a podélného značení a významně se měnící světelné podmínky.

Mezi systémy strojového vidění se řadí i pomocný systém na rozpoznávání dopravního značení.

Dynamická navigace RNS

Běžný způsob navigace standardu GPS se provádí s využitím statických dat, jinými slovy map, a spočívá v propočítávání ideální trasy k cíli podle daných kritérií. Zastaralost mapy se projevuje tím, kdy pokud jedeme např. po nově otevřeném úseku dálnice, tak navigační přístroj ukazuje, že se pohybujeme mimo vozovku (v poli) a navigační algoritmus dává nestandardní výsledky výpočtu trasy. Tyto mapy se dají sice aktualizovat, nicméně aktualizace je závislá na výrobci navigačního systému, a to jak

z pohledu frekvence aktualizací, tak případného zpoplatnění. Ovšem při řízení vozidla, zejména v neznámém prostředí, je požadavek na navigační systém, aby správně optimalizoval trasu i s ohledem na lokální dopravní jevy, jako jsou uzavírky a zácpy. O těchto jevech, však musí mít navigační systém příslušné informace. Některé systémy umožňují manuální zadání informací o uzavřených silnicích, ale takový způsob je nepohodlný, navíc se informace mohou měnit. Proto by bylo výhodné, aby navigační systém mohl přijímat tyto informace v reálném čase, prostřednictvím FM vysílání např. prostřednictvím RDS kanálů. Znamená to zavedení příslušných standardů a provozování infrastruktury. Např. v USA je provozována společností Microsoft služba MSN Direct^{*)} pro šíření informací nejen o dopravě a počasí, ale i o cenách pohonných hmot v okolí, stavu letadlové dopravy, atd. prostřednictvím sítě FM vysílačů ve větších městech. S touto službou jsou kompatibilní některé navigační systémy firem GARMIN a Pioneer. Služba je zpoplatněna sazbou 50USD za rok. V České Republice zatím takovýto ucelený systém nemá obdobu.

5.3 Testované řídicí jednotky

Předmětem testování byly řídicí jednotky, které mají součinnost s komfortními funkcemi vozu nižší třídy. Výběr tedy zahrnuje řídicí jednotku asistenta parkování, automatické klimatizace, centrálního zamykání a centrálního elektrického rozvodu, které jsou propojeny sběrnici CAN, viz. obr. č. 5.1, a také dveřní jednotky, které jsou propojeny samostatně sběrnici LIN, viz. obr. č. 5.2.

V následujících podkapitolách je uveden popis dvou řídicích jednotek, kterými se přímo zabývá tato práce. Ostatní řídicí jednotky byly řešeny v rámci jiné práce [24].

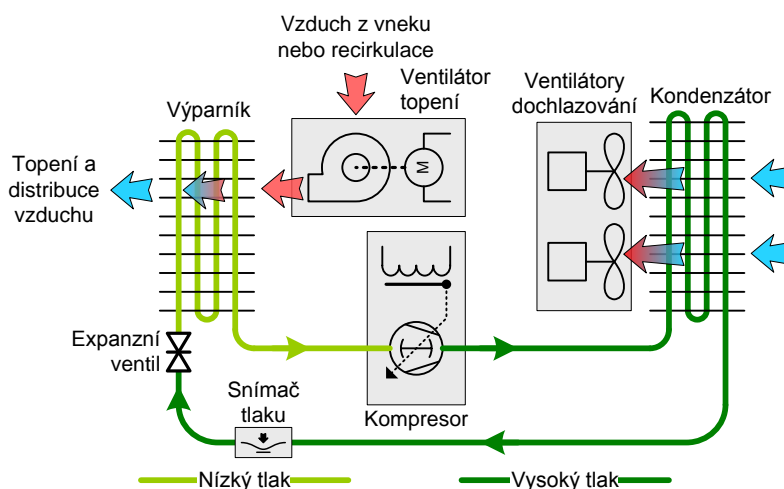
5.3.1 Automatická klimatizace Climatronic

Klimatizace dnes patří mezi běžnou komfortní výbavu vozidel i nižší třídy. Funkce klimatizace je založena na principu tepelného čerpadla, podobně jako chladnička. Základem je tedy chladicí okruh, který je schematicky znázorněn na následujícím obrázku obr. č. 5.4.

Součástí okruhu je kompresor, který je elektromagneticky říditelný, kdy velikostí proudu procházejícího cívkou elektromagnetu se nastavuje zdvih pístů kompresoru a tím jeho výkon. Kompresor stlačuje chladivo v plynném stavu, které pak

^{*)} Viz webová stránka [cit. 2011-03-11] <<https://www.msndirect.com/default.aspx>>. Pozn.: Společnost Microsoft na uvedené stránce bohužel oznamuje plánované ukončení služby k 2012-01-01.

pokračuje do kondenzátoru, kde odebráním tepla dojde k jeho zkapalnění. Natlakované chladivo v kapalném stavu přichází na expanzní ventil, kde dojde k radikálnímu poklesu tlaku a tím i teploty.



Obr. č. 5.4 Schéma chladicího okruhu klimatizace automobilu

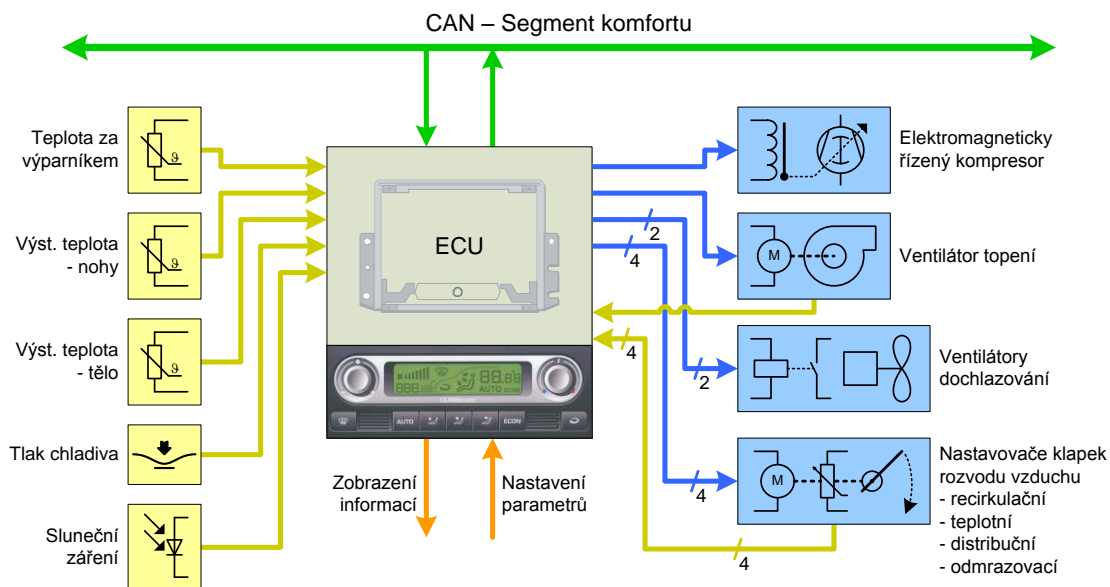
Chladivo pak přichází do výparníku, na kterém dochází k teplené výměně mezi chladivem a nasávaným vzduchem, čili dojde k ochlazení vzduchu a přeměně chladiva do plynného skupenství, které pak pokračuje opět do kompresoru. V okruhu před expanzním ventilem je zařazen také snímač pro měření tlaku. Informace o tlaku v okruhu má pro řídicí systém spíše bezpečnostní význam, dojde ke snížení výkonu kompresoru na minimum, pokud je tlak chladiva příliš nízký nebo příliš vysoký.

Řídicí systém klimatizace je tvořen řídicí jednotkou, která je součástí palubní přístrojové desky, snímači teplot, tlaku chladiva a intenzity slunečního svitu a akčními členy jako je elektromagnet pro řízení kompresoru, výkonový modul řízení ventilátoru topení, relé spínající ventilátory dochlazování a nastavovače klapek distribuce vzduchu.

V blokovém schématu na obr. č. 5.5 (na další straně) jsou znázorněny všechny tři smyčky komplexního řídicího systému, podobně jako na obr. č. 3.5. Interakční smyčka je rovněž znázorněna oranžově, komunikační pak zelenou barvou. Regulační smyčka zde není triviální, a proto jsou zde vstupy od senzorů do řídicí jednotky znázorněny žlutou barvou a výstupy na akční členy modrou.

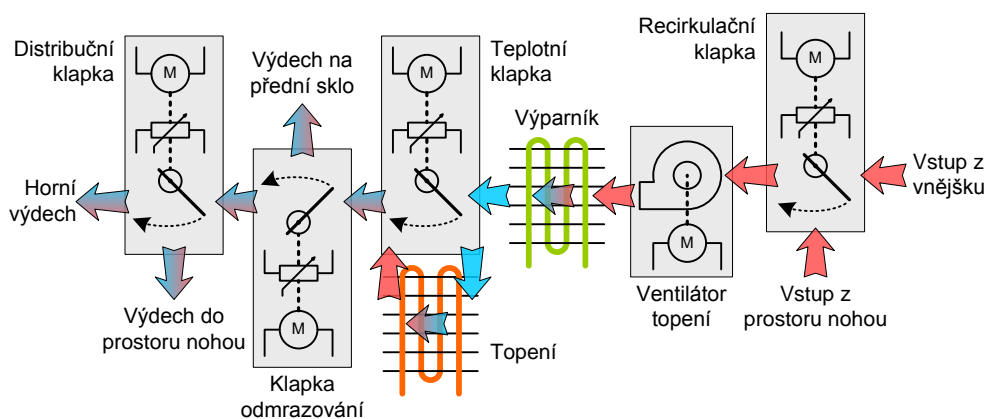
Hlavní regulační smyčka regulující chladicí výkon klimatizace je tvořena snímačem teploty za výparníkem a kompresorem. Do tohoto regulačního procesu vstupuje navíc snímač tlaku, jak již bylo zmíněno výše jen jako bezpečnostní prvek při mezních stavech v chladicím okruhu (extrémní teplotní poměry, únik chladiva, apod.). Požadovaný výkon klimatizace je dán teplotními poměry, tedy teplotou vnějšího

vzduchu a teplotou uvnitř vozu. Snímač pro měření vnější teploty je zpravidla termistor NTC umístěný v oblasti předního nárazníku. Snímač vnitřní teploty je buď infračervený teplotní snímač jako součást předního panelu, nebo NTC termistor uvnitř jednotky, ke kterému se nasává vzduch miniaturním ventilátorem přes mřížku v předním panelu.



Obr. č. 5.5 Blokové schéma elektronického řídicího systému klimatizace

Druhou regulační smyčkou je regulace teploty výstupního vzduchu. Ta se provádí tak, že vychlazený vzduch z výparníku se přivádí na teplotní klapku, která podle svého natočení část vzduchu nasměruje na radiátor topení, kde se ohřeje a následně za teplotní klapkou se opět smíchá se zbývajícím studeným vzduchem, jenž díky konstrukci teplotní klapky radiátor topení obtéká, viz přibližné znázornění proudění vzduchu na obr. č. 5.6.



Obr. č. 5.6 Distribuce vzduchu v klimatizaci

Směs teplého a studeného vzduchu pak dále pokračuje kolem klapky odmrazování, která reguluje proud vzduchu na přední sklo, na distribuční klapku, která rozděluje proud vzduchu do výdechů na tělo (horní) a do prostoru nohou. Za distribuční klapkou jsou umístěny snímače teploty vystupujícího vzduchu oběma výdechy. Regulační smyčka teploty vzduchu je tedy tvořena teplotní klapkou jako akčním členem a dvěma snímači teploty, tvořící zpětnou vazbu nezávisle na natočení distribuční klapky. Regulace teploty je dána nastavenou požadovanou teplotou, která se nastavuje na ovládacím panelu otočným ovladačem. V rámci regulace teploty lze vyřadit z provozu chladicí okruh, a to pomocí tlačítka ECON. V některých případech se na předním panelu místo ECON vyskytuje tlačítko označené A.C. (angl. Air Condition – klimatizace), které naopak funkci chladicího okruhu zapíná. Regulace teploty je dále ovlivňována také slunečním senzorem. Ten poskytuje informaci o intenzitě slunečního záření, které dopadá na vozidlo a způsobuje tzv. skleníkový efekt. Regulace teploty tak dovoluje reagovat na situace, kdy se během jízdy střídají úseky stínu (jízda lesem) a slunečního svitu (otevřená krajina), a zajistit tak pokud možno konstantní tepelnou pohodu uvnitř vozu.

Dále se v systému nachází několik dílčích regulačních smyček, lze je nazvat lokální, poněvadž slouží pro regulaci příslušné části zařízení. Jedná se o ventilátor topení, který se řídí pomocí PWM modulace a má zpětnou vazbu, tedy řídicí jednotka má informaci o skutečném napětí na motoru ventilátoru. Požadovaná hodnota se opět nastavuje rotačním ovladačem, případně je stanovena programem automaticky v režimu AUTO (plně automatický provoz) nebo DEFROST (odmrazování či odmlžování předního skla). Rovněž rotační klapky pro distribuci vzduchu, které se nastavují stejnosměrným motorem, mají zpětnou vazbu a to pomocí potenciometrického snímače natočení. Požadované nastavení je dáno programově s tím, že lze pomocí tlačítek navolit kombinaci privilegovaných směrů výstupu vzduchu na posádku (hlava, tělo, nohy).

Mezi lokální regulační smyčky se řadí i regulační smyčka tzv. recirkulační klapky. Její primární funkcí je výběr směru nasávaného vzduchu ventilátorem, tedy buď z vnějšku vozidla, nebo z prostoru nohou (recirkulace). Výběr směru se provádí příslušným tlačítkem na předním panelu. U vozidel vyšších kategorií může být vybírání směru automatizováno na základě analýzy některých škodlivin v nasávaném vzduchu, kdy např. při jízdě tunelem dojde automaticky k přepnutí na recirkulaci. Recirkulační klapka má však ještě další funkci. V případě, že je nastavena na směr z venku, tak svojí

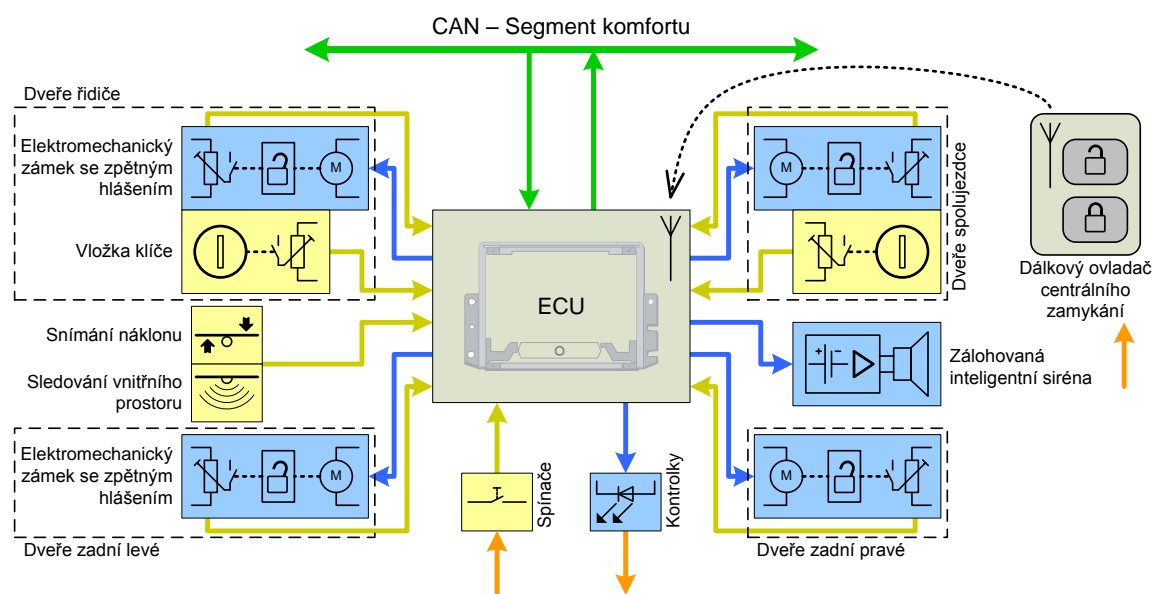
polohou omezuje množství vstupujícího vzduchu v závislosti na rychlosti jízdy vozidla. To proto, aby proudění vzduchu uvnitř vozidla bylo co nejméně závislé na rychlosti jízdy. Recirkulační klapka je tedy mechanicky poněkud komplikovanější, nicméně princip řízení a zpětné vazby je stejný jako u ostatních klapek.

5.3.2 Centrální zamykání s alarmem

Centrální zamykání, které může být doplněno o alarm, patří dnes mezi základní komfortní výbavu vozidel i nejnižších tříd. Nejedná se o systém tolik komplikovaný ve srovnání s klimatizací.

Centrální zamykání

Centrální zamykání spočívá v elektromechanickém ovládní polohy zámků v jednotlivých dveřích. Struktura řídicího systému je u vozů nižších tříd většinou tzv. centralizovaná v tom smyslu, že zámky dveří řídí jedna společná řídicí jednotka. Blokové schéma takového řídicího systému je znázorněno na obr. č. 5.7. V blokovém schématu není znázorněno ovládní pátých dveří. Ty jsou totiž v tomto případě napojeny na centrální řídicí jednotku a to z důvodu zachování variability systému. Zámek pátých dveří je totiž vždy ovládán elektromechanicky, a pokud je ovládán z centrální jednotky, může být, v případě absence centrálního zamykání ve výbavě vozidla u nejlevnějších variant, vynechána řídicí jednotka centrálního zamykání včetně elektrického ovládní zámků postranních dveří a příslušné kabeláže.

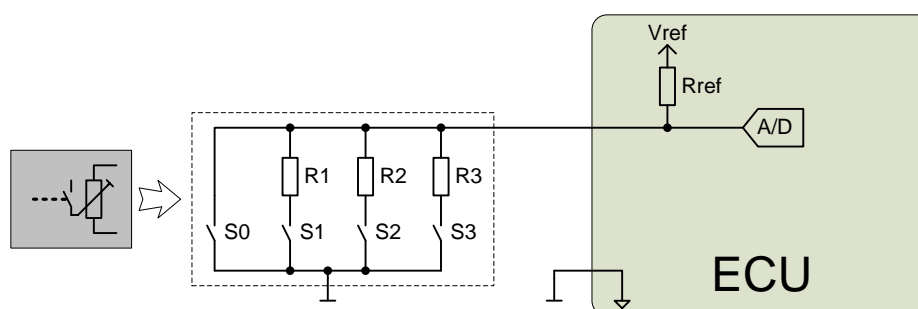


Obr. č. 5.7 Blokové schéma řídicího systému centrálního zamykání a alarmu

Naopak je tomu u vozidel středních a vyšších tříd, kde je již centrální zamykání samozřejmostí. U těchto vozidel mají každé postranní dveře svoji vlastní řídicí jednotku, na kterou jsou napojeny všechny ovládací prvky a akční členy. Tento řídicí systém je pak plně decentralizovaný, což navíc umožňuje zahrnout funkci centrálního zamykání přímo do centrální jednotky.

Elektromechanické zámky, jsou mechanismy poháněné elektromagnety (posuvné dvoupolohové akční členy) nebo elektrickými motory (rotační někdy též lineární) s příslušnými převody. Mechanicky mají zámky tři polohy: odemčeno, zamčeno a zajištěno (angl.: unlocked, locked, safe). V poloze zajištěno nelze dveře otevřít ani zvenku ani zevnitř pomocí páčky k otvírání dveří, to aby si narušitel po rozbití okna nemohl dveře otevřít zevnitř. Zámky jsou tedy konstruovány jako jednomotorové, kdy přechody mezi jednotlivými polohami jsou zajišťovány postupným pouštěním motoru v příslušném směru, nebo dvumotorové, kdy jeden motor zamyká a odemyká a druhý provede zajištění. Zámky pro zadní dveře mohou mít ještě další (tedy druhý resp. třetí) motor, a to v případě, že je systém vybaven funkcí elektrické dětské pojistky, která způsobí, že dveře lze otevřít ve stavu odemčeno jen z venku, aby si děti nemohly samy otevřít během jízdy.

Mechanismus zámku rovněž poskytuje zpětnou vazbu o poloze zámku a také o tom, zda jsou dveře otevřené (tzv. dveřní kontakt). Zpětná vazba polohy zámku stejně jako poloha klíče ve vložce jsou tzv. odporově kódovány. Odporové kódování se používá v řídicích systémech založených na mikropočítačích pro přivedení informace z několika spínačů po jednom vodiči a využívá princip napěťového děliče, kdy se výstup napěťového děliče zpracovává pomocí vstupu AD převodníku v mikropočítači.



Obr. č. 5.8 Princip odporového kódování

Na následujícím obr. č. 5.8 je znázorněno schéma připojení spínačů spolu s kódovacími rezistory k řídicí jednotce a značka odporového kódování, která je použita v obr. č. 5.7.

Alarm

Systém alarmu se skládá z čidel vyhodnocujících stav uzamčeného vozidla a zálohované inteligentní sirény. Vyhodnocují se nejméně dvě základní skupiny narušení bezpečnosti vozidla, a to jednak vniknutí do vozidla, což zjišťují snímače hlídání vnitřního prostoru, a pak nežádoucí pohyb vozidla způsobený např. pokusem o odtažení, což vyhodnocuje snímač náklonu.

Snímače hlídání vnitřního prostoru jsou založeny na principu měření vzdálenosti, resp. pohybu objektů, za pomoci buď infračerveného záření, nebo ultrazvuku. Hlídání vnitřního prostoru lze před uzamknutím vozidla vypnout příslušným ovládacím tlačítkem, které je umístěno ne příliš nápadně, což je vhodné v případě, že ve vozidle zůstane objekt, který se samovolně pohybuje, např. domácí zvíře.

Snímač náklonu využívá elektronických snímačů pohybu, jako jsou akcelerometry, a náklonu, zde se jedná o principy, které využívá i elektronická libela.

Oba druhy senzorů jsou spojeny v jeden celek a jsou propojeny s řídicí jednotkou. K předávání informací dochází pomocí pulsního signálu. Někdy se takový senzorový celek označuje jako klastř senzorů (angl.: sensor cluster).

Inteligentní siréna signalizuje zvukovým signálem nejen aktivní stav alarmu, ale i potvrzovací signály při zamykání (aktivace alarmu) a odemykání (deaktivace) vozidla. Siréna dokáže sama detekovat (proto je označována jako inteligentní) pokus o narušení elektrické sítě vozidla, např. odpojení baterie. Z tohoto důvodu má siréna integrovaný vlastní akumulátor, který se dobíjí z elektrické sítě vozidla a zásobuje sirénu energií v případě odpojení sirény od elektrické sítě nebo poklesu napětí, po dobu několika hodin aktivního stavu sirény. Inteligentní siréna je s řídicí jednotkou propojena pomocí sériové sběrnice s proprietárním komunikačním protokolem.

Ovládání a funkce

Ovládání centrálního zamykání tedy zamknutí a odemknutí vozidla se provádí buď pomocí vložky klíče na dveřích, nebo bezdrátově pomocí dálkového ovladače, který je součástí klíče. Dálkový ovladač předává povely nejčastěji rádiovým vysíláním. Dříve se také vyskytovaly systémy s infračerveným přenosem. V obou případech jsou povely kódovány pomocí speciálního plovoucího kódu.

Pokud je vozidlo vybaveno i alarmem pak zamčením/odemčením dojde k aktivaci/deaktivaci alarmu, tedy k hlídání bezpečnosti vozidla, navíc jsou akce potvrzeny směrovými světly a akustickým signálem sirény. Povelem odemknout se také vypne spuštěný alarm.

Řídicí jednotka má zpravidla tlačítka pro deaktivaci senzorů hlídání vnitřního prostoru a tlačítko pro ovládání zámků zevnitř vozu, stav bývá signalizován prosvětlením tlačítka. Jako další signalizační prvek je k řídicí jednotce připojena kontrolka, která blikáním indikuje zamčení vozidla a aktivovaný alarm.

Základní funkce jsou vesměs známé, dále budou zmíněny některé pokročilé funkce, které využívají komunikace po datové síti vozidla s ostatními řídicími jednotkami a někdy se označují jako tzv. komfortní funkce. Mezi takové funkce patří i *automatické zamčení* (nikoliv zajištění) dveří vozidla po rozjetí a překročení minimální rychlosti (cca 15 km/h). K odemčení pak dojde po vytažení klíče ze zapalování, ale vzhledem k tomu, že je jen zamčeno lze dveře otevřít zevnitř běžným způsobem, pokud u zadních dveří není aktivována dětská pojistka. Další funkcí je pak *komfortní ovládání stahování* oken. To funguje tak, že pokud při zamykání/odemykání, ať pomocí vložky nebo dálkovým ovladačem, je příslušný povel aktivován (vysílán) déle než cca 2 s, pak řídicí jednotka tuto skutečnost předá prostřednictvím datové sítě řídicím jednotkám stahovačů oken a ty provádí zavírání/otevírání oken po dobu vysílání povelu. V případě oddělené datové sítě dveřních jednotek, viz obr. č. 5.2, pak příslušné dveřní jednotky přímo zpracovávají signál z vložek klíče a není tedy možné funkci vyvolat dálkovým ovladačem.

6. Technické prostředky pro testování

Technické prostředky pro testování zahrnují HiL testovací platformu (viz obr. č. 3.4) a programové vybavení jednak pro vývoj modelů okolí řídicí jednotky (řídících jednotek) a také pro vytváření testů, řízení a vyhodnocování vlastního průběhu testování.

Jak již bylo zmíněno v kapitole 3.3 existují dvě možnosti pořízení technických prostředků. Buď lze navrhnout testovací systém vlastním vývojem, nebo vybrat některý z nabízených profesionálních testerů. Vlastní vývoj se může zdát na první pohled méně nákladný, nicméně při narůstání složitosti a univerzálnosti testovacího systému se velmi prodlužuje doba vývoje systému a tím rostou i náklady. Tudíž při určitém stupni složitosti a univerzálnosti systému se již může vyplatit pořízení hotového testovacího systému.

Nejméně důležitým faktorem při výběru konfigurace testovacího systému je, kromě pořizovacích nákladů, rovněž kompatibilita a přenositelnost modelů a testovacích skriptů např. v rámci koncernu. Poněvadž je třeba zvážit výhodnost úspory při nákupu hardware a software s náklady na vytvoření eventuelně přetvoření např. modelů okolí řídicích jednotek tak, aby se daly použít s novým testovacím systémem. Opět, pokud náklady na import modelů či testovacích skriptů přesáhnou rozdíl v pořizovacích nákladech, je nutné tzv. zůstat kompatibilní a pořídit dražší variantu technických prostředků.

Nákladově zajímavá je kombinace cenově dostupného hardware s volně dostupnými programy s otevřenými zdroji (angl.: open source).

6.1 Vývoj vlastního testovacího zařízení

Na počátku autorova studia oblasti testování elektronických systémů byla zřejmá následující idea: Testovaný systém (řídicí jednotka, případně více navzájem propojených jednotek) se připojí svými vstupy a výstupy k určitému testovacímu systému, který se postará o vybuzení vstupů příslušnými signály a zachycení výstupních signálů včetně časové synchronizace (sousednosti). Následně se provede vyhodnocení, zda při nastavení příslušných vstupů se na výstupech objevily požadované stavy. Tato idea se zpočátku jevila jako snadno realizovatelná, proto byl proveden pokus o vyvinutí jednoduchého testovacího systému, který umožňuje testování konkrétních řídicích jednotek propojených sběrnici LIN bus [19]. Popis tohoto systému se nachází v Příloze A.

6.2 Profesionální testovací systémy

V této kapitole jsou uvedeny některé profesionální testovací systémy, které současný světový trh nabízí. Jen pro informaci zde budou zmíněny produkty firem Keithley a ETAS. V následujících podkapitolách jsou pak podrobněji popsány systémy nabízené společnostmi National Instruments, dSPACE a MBtech Group.

Na druhou stranu je zde záměrně vynechán popis programového vybavení společnosti Mathworks (Matlab, Simulink, Realtime Workshop, a další) a to z důvodu, že je lze považovat za průmyslový standard v oblasti technických výpočtů a simulací systémů a tedy za všeobecně známé, eventuálně lze popis snadno dohledat.

V Příloze B pak lze najít kontaktní informace na distributory či zastoupení jednotlivých společností v České Republice.

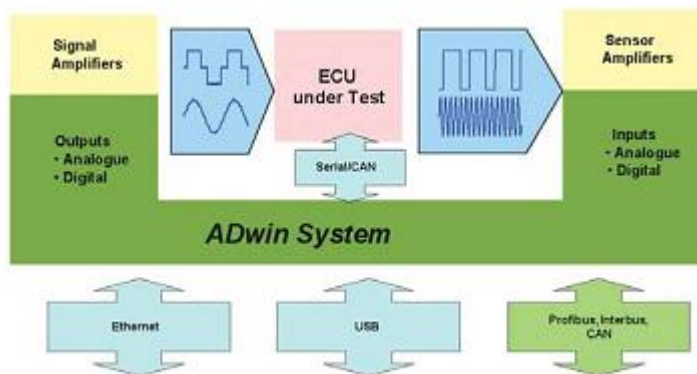
Výčet testovacích systémů uvedený výše není zdaleka vyčerpávající. Existují i další, např. RT-Lab Testdrive od společnosti Opal-RT (<<http://www.opal-rt.com>>) nebo systémy společnosti Amet (<<http://www.amet.it>>), které rovněž vychází z RT-Lab Testdrive.

Systém ADwin společnosti Keithley

Americká společnost Keithley nabízí pro profesionální testování a automatizaci rodinu produktů s názvem ADwin. Produkty ADwin mají společné následující vlastnosti:

- Operace jsou deterministické a robustní díky jednoúčelovému procesoru s real-time operačním systémem
- 32bitový procesor s výpočty v plovoucí řádové čárce (floating-point) – SHARC-DSP od Analog Devices s lokální signálovou pamětí a až s 128MB operační pamětí
- Analogové a digitální vstupy/výstupy, paralelní synchronní a multiplexované analogové vstupy, čítače, PWM vstupy/výstupy, filtry, izolační zesilovače, odporově digitální (RTD) vstupy
- Rozhraní pro sběrnice CAN-bus, Profibus, Interbus, RS232, RS485
- Ethernetové nebo USB rozhraní pro komunikaci s počítačem PC
- Real-time vývojový software ADbasic

Jsou dostupné ve třech verzích Light-16, Gold a Pro (modulární systém). Dají se programovat v C a C++ a také z prostředí Matlab a Labview (viz. dále). Program, resp. model pak běží v reálném čase, přičemž k řízení běhu slouží připojený počítač PC. Nevýhodou je kompatibilita digitálních vstupů a výstupů s úrovněmi TTL, takže nejsou příliš vhodné pro přímé nasazení v automobilovém průmyslu. Na následujícím obr. č. 6.1 je znázorněno napojení systému ADwin na řídicí jednotku při HiL testování.

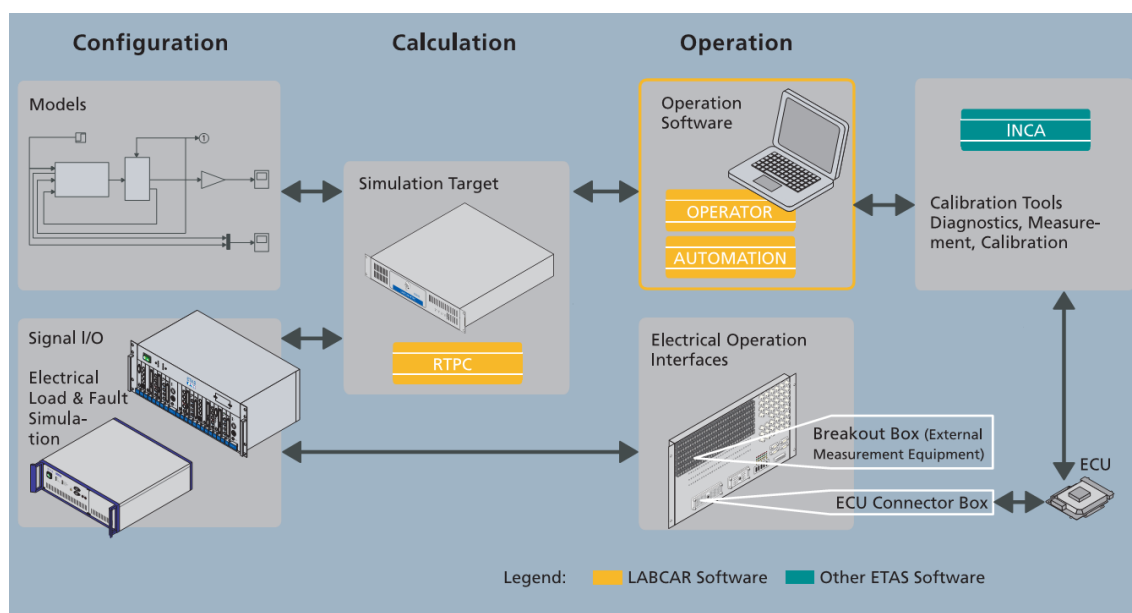


Obr. č. 6.1 Konfigurace systému ADwin při HiL testování řídicí jednotky (ECU) [30]

Systémy společnosti ETAS

Produkty společnosti ETAS GmbH je možné v automobilovém průmyslu rovněž považovat za standard, poněvadž jsou podobně jako systémy dSPACE využívány předními výrobci automobilů pro HiL testování, nicméně špatná dostupnost v ČR (společnost nemá oficiální zastoupení) posouvá tyto systémy za okraj zájmu této práce.

Na obr. č. 6.2 je znázorněná struktura systému ETAS LABCAR. Modely se vytváří v prostředí Simulink a následně se přenesou do HiL testovací platformy (v obrázku označené jako „Simulation Target“), což je „real-time“ systém založený na procesorech PowerPC. Platforma dále disponuje modulárními vstupy a výstupy plně slučitelné s automobilovými signály v konfiguraci, které požaduje příslušná aplikace.



Obr. č. 6.2 Přehled systému LABCAR^{*)}

^{*)} Obrázek převzat z: [cit. 2011-03-11]
http://www.etas.com/data/products_LABCAR_general/GR3D_LABCAR_RTPC_System_010.gif

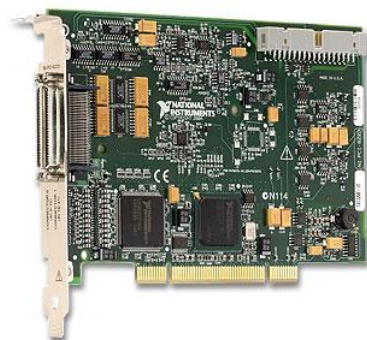
Z obecného pohledu se dá tedy říci, že systém LABCAR je velmi podobný modulárnímu systému dSPACE, které jsou podrobněji rozebrány dále. Výhodou je relativně jednoduchá konverze modelů pro simulace, je třeba jen zaměnit příslušné bloky vstupů a výstupů. Nicméně řízení průběhu testování a případná automatizace už je rozdílná. Vzhledem k omezené dostupnosti ani nebyly zjišťovány pořizovací náklady.

6.2.1 Systémy společnosti National Instruments

National Instruments je americká společnost s pobočkami po celém světě a nabízí velmi širokou a ucelenou škálu testovacích a měřicích systémů, včetně systémů pro získávání dat (angl.: data acquisition). Nabídka produktů je velmi rozsáhlá, od karet pro získávání dat, určených do sběrnice PCI, až po škálovatelný modulární systém založený na sběrnici PXI, resp. PXIe. K ovládání hardwarových produktů nabízí společnost vlastní software LabVIEW.

Karty pro získávání dat

Karet pro měření je v nabídce řádově 80 typů. Zde jsou uvedeny stručné parametry jedné vybrané karty PCI-6229, která má nejlepší poměr cena ku počtu vstupů/výstupů. Karta PCI-6229 patří do série M karet pro sběr dat s nízkou cenou. S označením PCI je určena právě pro sběrnici PCI. Vyrábí se však i PXI verze pod označením PXI-6229, která je určena spíše pro modulární systém.



Obr. č. 6.3 Pohled na kartu PCI-6229^{)}*

Základní parametry PCI-6229 jsou následující.

- Analogové vstupy: 32 kanálů, 16bitů, rozsah -10 – 10 V, 250 kS/s
- Analogové výstupy: 4 kanály, 16bitů, rozsah -10 – 10 V, max. 5 mA, 833 kS/s
- Digitální vstupy/výstupy: 48 vstupů/výstupů (lze nastavit směr), rozsah 0 – 5 V (TTL) max. 24 mA na kanál, 448 mA celkem
- Čítače/časovače: 2, rozlišení 32bitů, max. 80 MHz, generování a měření pulsních signálů

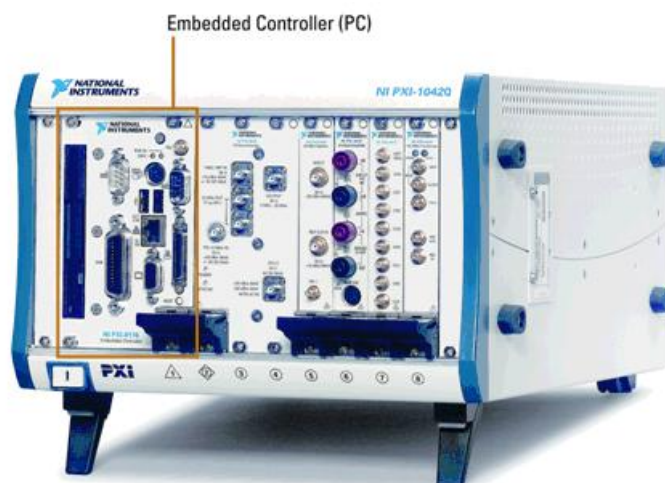
^{*)} Obrázek převzat z: [cit. 2011-03-11] <<http://sine.ni.com/nips/cds/view/p/lang/en/nid/14136>>

Signály jsou z karty vyvedeny pomocí dvou jemných konektorů, typu VHDCI (standardně se nachází na radičích rozhraní Ultra SCSI-3 pro pevné disky) s 68 vývody. Pro propojení karty, ať už s konektorovým boxem nebo jiným zařízením, je nutné přikoupit redukční kabely, které mají na druhém konci osazený větší 68 vývodový konektor typu D-sub (standardní konektor na Ultra SCSI-3 pevných discích). Portikus k většímu konektoru na kabelu je snáze dostupný a lépe se s ním pracuje při návrhu vlastního rozhraní nebo zařízení, na rozdíl od VHDCI konektoru. Bohužel cena kabelů (cca 3.000,- Kč bez DPH za kus) je v porovnání s cenou karty (cca 18.000,- Kč bez DPH) výrazně nezanedbatelná.

Do série M jsou zařazeny i karty v tzv. průmyslovém provedení, což znamená garantovanou odolnost vstupů a výstupů do 24V.

Modulární systém

Modulární systém je založen na sběrnici PXI, což je modifikovaná sběrnice PCI, nebo PXIe odvozené ze sběrnice PCI Express. Modulární systém se skládá ze skříně, řídicího počítače, který může být vestavěný ve skříně, jako PXI nebo PXIe modul, nebo vzdálený (běžné PC nebo notebook) připojený prostřednictvím proprietárního rozhraní, a dále zahrnuje vstupně/výstupní karty.



Obr. č. 6.4 Modulární systém založený na sběrnici PXI s vyznačeným řídicím počítačem (angl.: Embedded Controller))*

Vestavěný řídicí počítač umožňuje i „Real-time“ provoz díky programovému vybavení LabVIEW (viz. dále). Nabízené řídicí počítače jsou založeny na procesorech

*) Obrázek převzat z webových stránek o modulárním systému společnosti National Instruments [cit. 2011-03-11] <<http://zone.ni.com/devzone/cda/tut/p/id/4426#toc1>>

Intel kompatibilní s x86. Na výběr jsou PXI moduly s procesory od Pentia 266 MHz přes Celeron 575 2,0GHz až po Core 2 Quad s frekvencí 2,2 GHz.

Další PXI moduly umožňují doplnit systém o nepřeberné množství periférií, jako analogové a digitální vstupy/výstupy, čítače/časovače, sběrnice rozhraní a další, a to u některých modulů i v průmyslovém provedení.

Software LabVIEW

LabVIEW je grafické vývojové prostředí, které se pomalu stává průmyslovým standardem, umožňující snadný a rychlý vývoj aplikací pro systémy National Instruments a to zejména v oblasti složitých měřicích procesů. LabVIEW lze doplnit o „real-time“ modul a vytvářet tak aplikace v reálném čase pro příslušný hardware. Lze tak z běžného počítače PC vytvořit specifickou „real-time“ platformu, kterou lze rozšiřovat pomocí vstupně/výstupních karet do sběrnice PCI.

Pro implementaci některých algoritmů je grafický způsob nešikovný. Proto National Instruments nabízí i prostředí LabWindows/CVI, které umožňuje implementaci měřicích a řídicích procesů v jazyce ANSI C.

6.2.2 Systémy společnosti dSPACE

Německá společnost dSPACE GmbH nabízí univerzální řešení pro průmyslové testování se specializací na automobilový průmysl. Všechny testovací systémy umožňují běh v reálném čase a jsou založeny na procesorech PowerPC.

Na výběr jsou jednak tzv. jednodesková řešení hardware, určená k zabudování do počítače PC buď do ISA nebo PCI slotu, dále pak kompaktní řešení s názvem MicroAutoBox a rovněž komplexní soubor modulárního hardware.

Jednodesková řešení

Společnost dSPACE nabízí dvě jednodesková řešení testovacího zařízení DS1103 a DS1104. Obě řešení nabízí podobný soubor vstupů a výstupů liší se však jejich počtem, výpočetním výkonem a typem sběrnice. DS1103 je založena na procesoru PowerPC 604e s taktovací frekvencí 400 MHz, má i více vstupů a výstupů ve srovnání s DS1104, disponuje však sběrnici ISA, která se dnes v běžných a již poměrně výkonných počítačích PC nevyskytuje. Naproti tomu DS1104, která je vybavena sběrnici PCI, tudíž jí není problém umístit do libovolného moderního počítače PC, má méně výkonný procesor, menší počet vstupů a výstupů a hlavně nedisponuje rozhraním sběrnice CAN, což je pro testování v automobilovém průmyslu značný handicap. Pro srovnání obou karet je v následující tab. č. 6.1 uveden stručný výčet hlavních parametrů.

Karty jsou osazeny speciálními 100 vývodovými konektory s jemnou roztečí (karta DS1103 třemi, DS1104 jedním) pro vyvedení signálů na zadní panel. Pro snadnější přístup k jednotlivým signálům lze ke kartám přikoupit, jako zvláštní příslušenství, konektorový panel, případně kombinovaný panel s konektory a LED diodami, které signalizují stav digitálních vstupů/výstupů.

Tab. č. 6.1 Srovnání parametrů karet DS1103 a DS1104

	DS1103	DS1104
Rozhraní sběrnice	ISA	PCI
Procesor	PowerPC 604e na 400 MHz	PowerPC 603e na 250 MHz
Operační paměť	128 MB	32 MB
Analogové vstupy	16 kanálů - multiplexované, 16bitů, rozsah ± 10 V, $>4\mu\text{s}$ 4 kanály, 12bitů, ± 10 V, >800 ns	4 kanály - multiplexované, 16bitů, rozsah ± 10 V, $>2\mu\text{s}$ 4 kanály, 12bitů, ± 10 V, >800 ns
Analogové výstupy	8 kanálů, 14bitů, ± 10 V, 5 μs	8 kanálů, 16bitů, ± 10 V, 10 μs
Digitální vstupy/výstupy	4 kanály po 8bitech, programovatelné	20 digitálních programovatelných linek
Rozhraní inkrementálního kodéru	6 digitálních vstupů s digitálním filtrováním šumu, max. 1,65 MHz	2 digitální vstupy, TTL nebo RS422, 24bit kódér, max. 1,65 MHz
Sériové rozhraní	RS232, až 1 Mb	UART (RS232)
CAN rozhraní	Dle ISO 11898, 1 Mb	--
Podsystem se signálovým procesorem	<ul style="list-style-type: none"> • TMS320F240 na 20 MHz • Symetrická i nesymetrická PWM a prostorově vektorová modulace – třífázový PWM výstup a 4 nezávislé PWM výstupy • 4 zachytávací vstupy • 16 analogových multiplexovaných vstupů, 10bitů, 6,6 μs • 18 digitálních TTL vstupů/výstupů 	<ul style="list-style-type: none"> • TMS320F240 • PWM modulace – třífázový výstup a 4 nezávislé PWM výstupy • 4 kanály pro měření (F/D) a generování (D/F) frekvenčních průběhů • 14 digitálních TTL vstupů/výstupů

Požizovací náklady pro školní instituce na kartu DS1103 včetně vývojového prostředí bez dalšího příslušenství začíná na částce 349.600,- Kč bez DPH, u karty DS1104 pak cena začíná na 148.500,- Kč bez DPH.

MicroAutoBox

MicroAutoBox je zvláštní verze dSPACE simulátoru (hardware), který je zabudován v kompaktní skřínce a je primárně určen pro simulaci řídicích jednotek přímo ve voze. Byl vytvořen pro zvýšení efektivity procesu rychlého vývoje (rapid prototyping) a je využíván předními výrobci pro automobilový průmysl (BMW, Daimler, DENSO, Fiat, Ford, Hella, Nissan, Visteon, Volkswagen a další).

MicroAutoBox umožňuje ověřit správnost použitých řídicích algoritmů přímo v reálném prostředí, ve kterém pak bude pracovat vyvíjená řídicí jednotka. Nabízí mnohem výkonnější hardware (výpočetní výkon procesoru, velikost paměti) a umožňuje výpočty v pohyblivé řádové čáře. Na rozdíl od finální podoby řídicí jednotky, která má jednoúčelový hardware navržený pro danou aplikaci a zoptimalizovaný firmware (program pro zabudovaný „embedded“ počítač, např. v řídicí jednotce, ale tento výraz se používá i u mobilních telefonů apod.), kdy výpočty probíhají víceméně jako celočíselné. Simulátor disponuje rozhraním pro připojení počítače PC (např. notebooku), které slouží pro snadnou aktualizaci firmware, monitorování signálů a čtení dat zaznamenaných během provozu – pomocí tzv. „datalogeru“. Což umožňuje ve spojení s koncepcí simulátoru samostatný provoz přímo v reálné aplikaci a po provedení experimentu získat data zaznamenaná v čase (vstupní, výstupní a vnitřní signály a stavy), která lze využít pro ladění aplikace a odstraňování chyb (debugging).

Výčet hlavních parametrů systému MicroAutobox je následující:

- Procesor PowerPC 750GL s taktovací frekvencí 800 MHz
- 16 MB operační paměti
- 16 MB paměti typu FLASH pro firmware (aplikaci)
- Připojení k PC zvláštním konektorem, rychlost spojení 100 Mbit/s
- Hliníkový kryt, rozměry 200x225x50 / 95 mm – podle verze
- Provozní teplota -40 až +85 °C
- Napájecí napětí 6 až 40 V
- Rozhraní CAN 2 nebo 4 kanály
- Sériové rozhraní: 1 nebo 2 kanály RS232 a 1 nebo 2 kanály použitelné pro sběrnici LIN nebo K/L diagnostické vedení
- Možnost rozhraní sběrnice FlexRay
- Analogové vstupy: až 24 kanálů, 12bitů, multiplexované 4 do 1, rozsah 0 až 5 V, <math><6,7 \mu\text{s}</math>
- Analogové výstupy: 8 kanálů, 12bitů, rozsah 0 až 4,5 V, max. ± 5 mA
- Digitální vstupy/výstupy: cca 32 vstupů/výstupů, 16 kanálů jednotky zpracování časových signálů TPU (Time Processing Unit), 4 PWM vstupy a výstupy – 2,5 Hz až 100 kHz, střída 0 – 100 %, rozlišení 16bitů

Požizovací cena pro školní instituce samotného MicroAutoBoxu bez příslušenství a vývojového prostředí, činí 431.900,- Kč bez DPH.

Modulární hardware

Základem modulárního hardware je procesorová karta. Ta je pomocí optického nebo metalického vedení a rozhraní propojena s počítačem PC. Použité rozhraní je fyzicky stejné jako ethernetové rozhraní, nicméně použité rozhraní není s ethernetem

slučitelné a slouží jen na vzájemné propojení simulátoru a počítače. Všechny karty modulárního hardware jsou konstruovány pro zasunutí do ISA slotu, ze kterého jsou většinou jen napájeny. Vzájemná komunikace mezi procesorovou kartou a kartami s perifériemi je zajištěna pomocí sběrnice PHS-bus. Karty jsou vybaveny příslušným konektorem a propojeny širokým plochým kabelem. Karty se umísťují do speciální skříně, jenž obsahuje napájecí zdroj, ISA sběrnici a komunikační modul, který zabírá jeden ISA slot.

Procesorové karty jsou k dispozici ve dvou variantách. DS1005 s procesorem PowerPC 750GX s taktovací frekvencí 1000 MHz, která je vhodná pro méně náročné a jednodušší simulační výpočty, má však velmi krátkou dobu odezvy (např. při zpracování přerušení). Druhá verze DS1006 s procesorem AMD Opteron na 2,6 GHz, jenž má mnohem větší výpočetní výkon, je tedy vhodná pro náročné simulační výpočty, na druhou stranu má delší dobu odezvy. Navíc lze pomocí modulu Gigalink procesorové karty (max. 4) navzájem propojit a vytvořit tak multiprocesorový systém s velkým výpočetním výkonem.

Hlavní rozšiřující periferní kartou je tzv. HiL karta DS2211. Ta poskytuje základní sadu vstupů a výstupů pro HiL simulace, jejíž výčet je následující:

- Analogové vstupy: 16 kanálů, 14bitů, rozsah 0 – 60 V, >60 μ s
- Analogové výstupy: 20 kanálů, 12bitů, 0 – 10 V, max. 5 mA
- Digitální vstupy/výstupy: 16 vstupů a 16 výstupů, rozsah 0 – 60 V (napájeno z vnějšího zdroje), 8 vstupů a 9 výstupů pro PWM modulaci, měření a generování frekvenčních průběhů
- Rozhraní sběrnice CAN 2 kanály
- Rozhraní RS232
- Odporové výstupy: 10 kanálů, rozsah 15,8 Ω až 1 M Ω , \pm 80 mA, max. 250mW
- Vstupy/výstupy simulace motoru: 8 vstupů vstříkávání, 8 vstupů zapálení, 1 výstup signálu klikového hřídele, 4 výstupy signálu vačkového hřídele, 4 výstupy signálu kol/senzoru klepání

Karta dále umožňuje pracovat s úhlově synchronními signály, využívanými při simulaci chodu spalovacího motoru, kdy některé veličiny jsou závislé na úhlu natočení klikového, resp. vačkového hřídele.

Další významnou periferní kartou je DS4002, která slouží ke generování a zachytávání digitálních signálů a hlavně jejich časových průběhů, které již není aplikace schopna zpracovat v reálném čase. K tomu je k dispozici celkem 8 kanálů. Dále pak tato karta disponuje 32 digitálními programovatelnými vstupy/výstupy.

V nabídce jsou pak i samostatné či různě kombinované karty s analogovými i digitálními vstupy a výstupy s různým počtem kanálů, přesností, parametry a i různou funkcí, co se týká práce s časovými průběhy signálů. Lze pořídit i vývojovou desku, která je vybavena jen rozhraním sběrnice PHS-bus a zbytek karty tvoří univerzální plošné spoje pro vlastní specifické zapojení.

Tímto způsobem lze nakonfigurovat libovolně rozsáhlý a výkonný systém. Je třeba však počítat se značnými náklady, protože ceny jednotlivých karet se pohybují v řádech statisíců Kč a je nutné pořídit i vývojové prostředí a nezbytné příslušenství. Pak se pořizovací cena poměrně jednoduchého simulátoru vyšplhá na jeden milion korun.

Programování simulátoru

Aplikace (resp. firmware) pro hardware simulátoru lze vytvořit pomocí prostředí Matlab Simulink. Kde se pomocí bloků vytvoří model aplikace, která se napojí na simulinkovské bloky jednotlivých vstupů/výstupů, které se dodávají ve formě RTI knihoven ke každé kartě. Z modelu se pak vygenerují zdrojové kódy aplikace v jazyce C, které se následně přeloží pro platformu PowerPC. Takto vytvořený firmware se nahraje do simulačního hardware. Tento proces generování, překladač a nahrání firmware lze zautomatizovat a spouštět klávesovou zkratkou přímo z prostředí Simulink.

Druhou možností vytváření firmware simulátoru je naprogramovat celou aplikaci v jazyce C s využitím knihoven pro přístup k periferiím simulátoru. Tento postup však vyžaduje programátorské zkušenosti a u komplexních aplikací je i časově náročný.

Ovládací a vizualizační aplikace ControlDesk

ControlDesk je řídicí a vizualizační aplikace určená pro počítače PC se systémem Windows a slouží ke konfiguraci a řízení experimentů.

S její pomocí lze nahrát přeložený firmware do simulátoru, provádět sledování výstupních a vnitřních proměnných. ControlDesk umožňuje načíst soubor .sdf (System Definition File), který vznikne během překladač modelu v Simulinku a obsahuje popis struktury modelu (zde nazývané systém). Proto umožňuje sledovat výstupy jednotlivých bloků, které lze dohledat prostřednictvím jejich symbolických identifikátorů (názvů) v reálném čase. Pro zobrazování, resp. vizualizaci, sledovaných hodnot lze v aplikaci ControlDesk vytvořit ve zvláštním okně panel s grafickým rozvržením (tzv. layout) různých zobrazovacích prvků, jako sedmissegmentové zobrazovače číselných hodnot, vícestavové zobrazovače s různými barvami, obrázky či různými zprávami (nápis)

podle intervalů hodnoty sledované veličiny, ručkové ukazatele (voltmetr, ampérmetr, rychloměr, otáčkoměr), sloupcové zobrazovače (tzv. bargrafy) a také časový záznamník průběhu hodnoty veličiny v čase. Hodnoty lze zobrazovat buď přímo, nebo lze definovat převodní přepočty – to se využívá v případech, kdy např. hodnota napětí z analogového vstupu představuje proud rezistorem o známém odporu. Typ zobrazované hodnoty se dá přepnout v reálném čase.

Na panel s rozvržením lze umístit i různé prvky pro nastavování hodnot v reálném čase sloužící jako vstupy, které se pak napojí na bloky konstant, které jsou rozmístěny v modelu. Pro zadávání hodnot lze použít prvky jako zaškrťovací a přepínací ovladače, různé skupiny tlačítek a přepínačů s nápisy, posuvné nastavovače, dialogové boxy pro přímé zadávání hodnot.

Pro snadnější orientaci lze ovládací a vizualizační prvky sdružovat do skupin a případně doplnit obrázky usnadňující orientaci na panelu. Příklad jak vypadá grafické uspořádání panelu je uvedeno dále v kapitole 7.2.1 na obr. č. 7.14.

S pomocí aplikace ControlDesk s využitím ovládacích a vizualizačních panelů lze provádět tzv. ruční testování, kdy operátor nastaví určitou kombinaci hodnot vstupních veličin a následně zjistí odezvu testovaného systému z vizuálních zobrazovačů, a to v reálném čase. Je však nutné počítat s určitým zpožděním, protože přesné časování v ControlDesku, resp. v systému Windows není možné. Pro přesnou časovou souslednost je nutné použít časový záznamník a zahrnout do něj požadované vstupní i výstupní veličiny.

Pro automatizaci testovacího procesu umožňuje aplikace ControlDesk zaznamenat většinu operací, které lze provádět (včetně sekvencí nastavování vstupních hodnot), a vytvořit tak tzv. makro. Pak se z testujícího stane jen konfigurační a následně kontrolující operátor a testy je možné pustit na delší čas, např. přes noc, bez zásahu operátora. Záznamem makra je skript v jazyce Python, který lze modifikovat a využít procedurální nebo i objektový způsob programování k vytvoření vlastního průběhu automatizovaného testování. Vstupní datové sekvence lze importovat v různých formátech.

Aplikace pro správu a automatizaci testování AutomationDesk

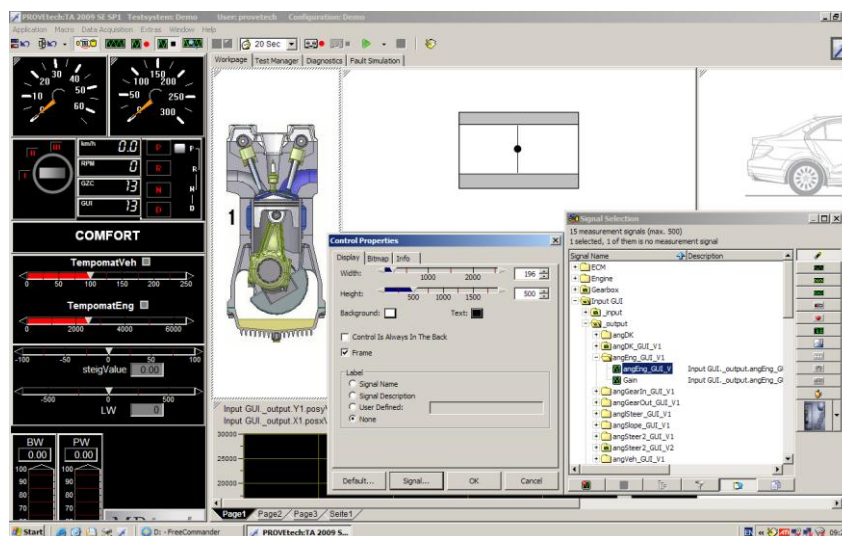
Pro automatizaci komplexnějšího testování lze využít speciálně zaměřenou aplikaci AutomationDesk. Ta umožňuje vytváření sekvencí pro řízení a vyhodnocování testování. Její součástí je grafický návrhář sekvencí, správce projektů, možnost vytvářet

knihovny testů a lze jednoduše přistupovat do simulinkovského modelu a real-time simulace.

6.2.3 Systémy společnosti MBtech Group

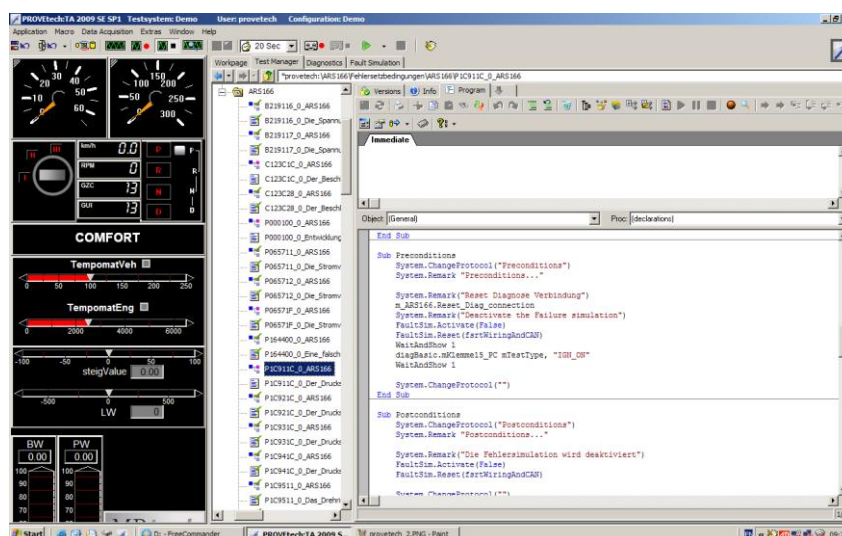
Testovací prostředí PROVEtech:TA

Hlavním produktem společnosti MBtech Group GmbH & Co. KGaA v oblasti HiL testování je bezspornu prostředí pro automatizaci testování s názvem PROVEtech:TA (zkratka TA je z angl.: Test Automation). Jedná se o komplexní prostředí nejen pro automatizaci testování, ale zahrnuje celkem čtyři pracovní oblasti.



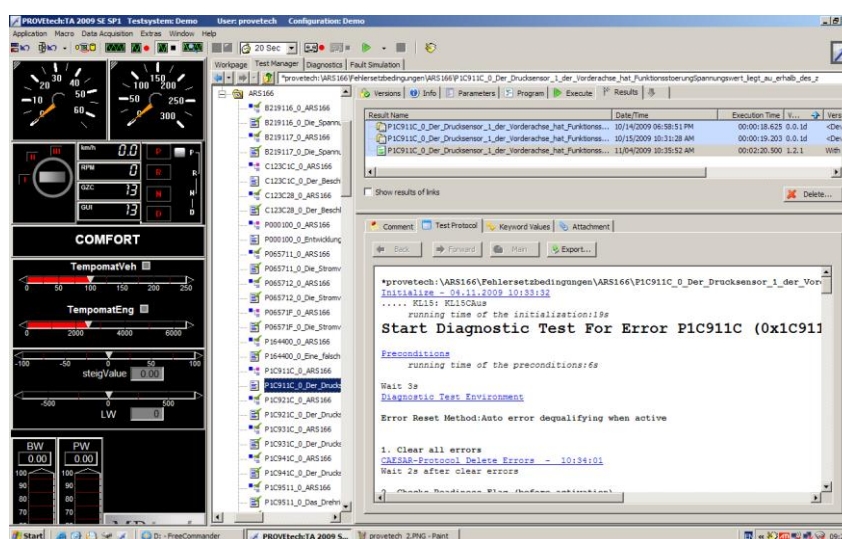
Obr. č. 6.5 Panel Workpage prostředí PROVEtech:TA

Panel *Workpage* (viz obr. č. 6.5) slouží k vizualizaci a parametrizaci průběhu testování, jeho možnosti a funkce jsou podobné jako v případě aplikace ControlDesk spol. dSPACE. Lze jej tedy použít k ručnímu testování.



Obr. č. 6.6 Záložka Program v panelu Test Manager

Panel *Test Manager* slouží k vytváření a kompletní správě testovacích skriptů pro automatizaci průběhu testování. Testovací skripty se vytváří v objektově orientovaném jazyce WinWrap Basic, který je velmi podobný známějšímu jazyku Visual Basic. Pro účely automatizace testování je samotný WinWrap Basic rozšířen o další objekty. Panel *Test Manager* obsahuje rámec se stromovou strukturou s testovacími skripty, moduly a třídami, včetně knihoven obsahující dílčí testovací kroky, a dále několik vnořených záložek s panely. Nejdůležitější je záložka *Program* pro editaci zdrojového textu skriptu, modulu, viz obr. č. 6.6 a také záložka *Results*, kde se nacházejí výsledky průběhu testu vizualizované pomocí jazyka HTML, znázorněná na obr. č. 6.7.



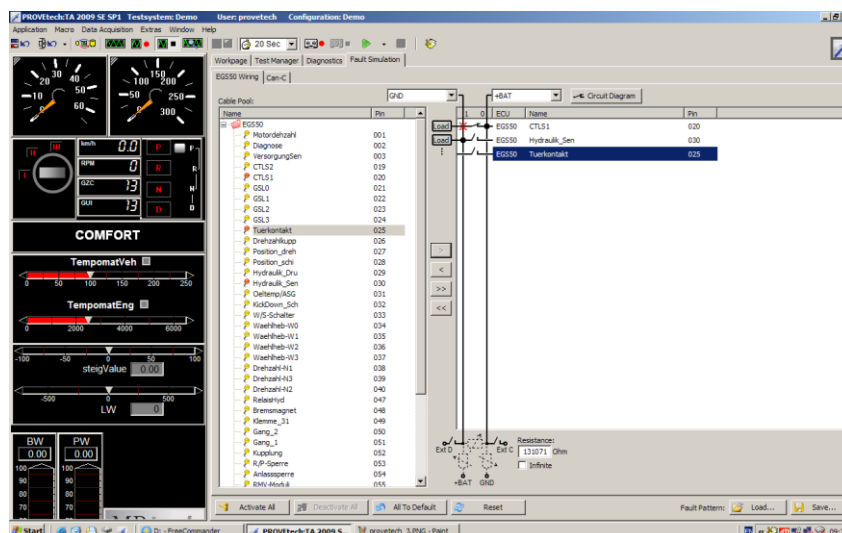
Obr. č. 6.7 Záložka *Results* v panelu *Test Manager*

Test manager umožňuje také správu verzí testovacích skriptů a organizaci jejich spouštění (jednotlivě nebo dávkově).

Dalším panelem v prostředí PROVEtech:TA je panel *Diagnostics*. Ten slouží k vlastní diagnostice testované řídicí jednotky (vyčtení a mazání chyb, čtení provozních hodnot, konfigurace a upgrade firmware) pomocí příslušného hardwarového rozhraní a využívá programové rozhraní Caesar, které se používá pro diagnostiku řídicích jednotek koncernu Daimler AG.

Poslední panel *Fault Simulation* slouží ke konfiguraci jednotky pro vkládání chyb (FIU – angl.: Failure Insertion Unit). V závislosti na propojení simulátoru s testovanou jednotkou prostřednictvím FIU, lze aktivovat poruchy typu rozpojení, zkrat na napájení či zem, nebo jiný vodič, jak je znázorněno na obr. č. 6.8. Kde pro první signál (vodič) je nastaven zkrat na napájení (současně není možné nastavit zkrat na zem,

poněvadž by došlo k vyzkratování napájecího zdroje, proto je zkrat na zem označen červeným křížkem), na druhém je nastaven zkrat na zem a zároveň rozpojení a na třetím pak rozpojení vodiče. Nastavené poruchové stavy lze všechny najednou aktivovat, respektive deaktivovat pomocí příslušných tlačítek ve spodní části panelu.



Obr. č. 6.8 Panel Fault Simulation prostředí PROVEtech:TA

Součástí hlavního okna prostředí PROVEtech:TA, jak si lze všimnout na obr. č. 6.5 až obr. č. 6.8 v levé části, je také rámec s názvem *Cockpit*, který má podobnou funkci jako panel Workpage, jen je viditelný nezávisle na zvoleném panelu. Hodí se tedy pro umístění klíčových vizualizačních a ovládacích prvků, které jsou viditelné a přístupné pořád, což se hodí zejména při ladění testovacích skriptů.

Velkou výhodou prostředí PROVEtech:TA je bezesporu to, že dokáže spolupracovat s různými HiL simulátory předních výrobců jako dSPACE a ETAS, a také s binární podobou modelu (ve formátu knihovního souboru .dll) z prostředí Matlab Simulink. Pak lze provozovat čistě softwarovou smyčku a testovat model ve smyčce tzv. MiL, jak bylo zmíněno v kapitole 3.1.2 a také viz [36].

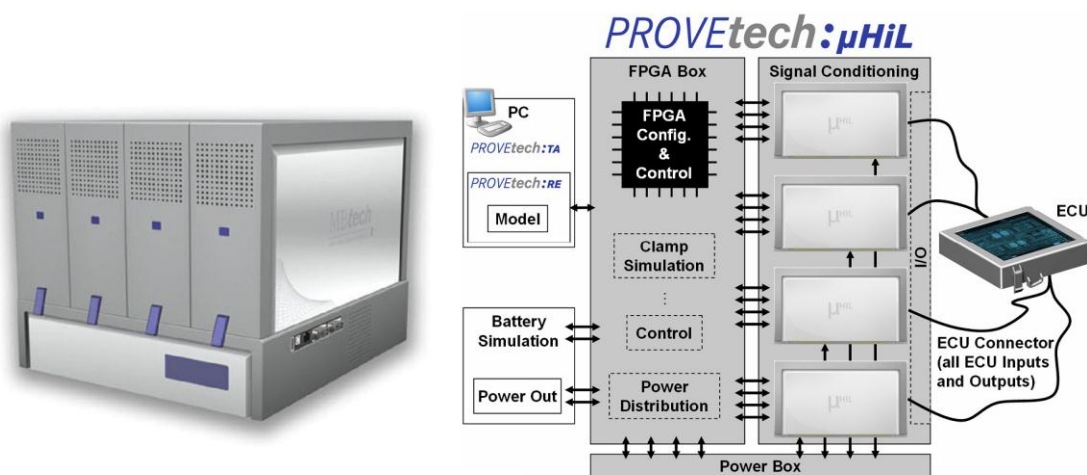
Programové vybavení pro implementaci HiL platformy PROVEtech:RE

PROVEtech:RE (zkratka RE je z angl.: Runtime environment) je sada programového vybavení umožňující vytvoření HiL testovací platformy i z běžného počítače typu PC. Pro běh uživatelské aplikace se využívá standardního prostředí operačního systému Microsoft Windows (verze XP, Vista, 7). Umožňuje ovládání vstupů a výstupů hardwaru, simulace sběrnicových systémů, eventuálně provoz modelu pro testování v uzavřené smyčce (HiL). Pro provoz modelu v reálném čase je využívána

technologie INtime od společnosti tenAsys [31], která umožňuje dosáhnout modelového kroku pod 1ms.

Testovací HiL platforma PROVEtech:μHiL

PROVEtech:μHiL je HiL testovací platforma, vyvinutá jako levnější alternativa k profesionálním a nákladným systémům firem dSPACE a ETAS. Na druhou stranu poskytuje nižší výkon, co se týká složitosti modelu a jeho provozu v reálném čase, a omezené portfolio vstupů a výstupů. Pořizovací náklady jsou výhodné zejména v kombinaci s testovacím prostředím PROVEtech:TA.



Obr. č. 6.9 Testovací systém PROVEtech:μHiL a jeho vnitřní struktura^{*)}

Pro simulaci prostředí, tedy provoz modelu v reálném čase, používá běhové prostředí PROVEtech:RE. Pro zpracování časově kritických operací používá technologie programovatelných logických polí (FPGA). Cílem bylo kompaktní řešení, tak, aby simulátor mohl být umístěn přímo na stole na vývojářském pracovišti, a variabilita. Proto může být základní box doplněn o 1 až 4 moduly pro připojení a zpracování signálů různých úrovní a to včetně možnosti vkládání chyb (FIU).

Ostatní produkty ze skupiny PROVEtech

Skupina produktů PROVEtech společnosti MBtech Group je mnohem širší než naznačuje předcházející výčet. PROVEtech se tím jako balík programových nástrojů snaží pokrýt nejen testování, ale i samotný proces vývoje řídicích systémů automobilů, viz [32].

Následuje tedy přehled ostatních produktů se stručným popisem.

^{*)} Obrázky převzaty z propagačních materiálů společnosti MBtech Group GmbH & Co. KGaA, viz <http://www.mbtech-group.com/fileadmin/media/pdf/electronics_solutions/PROVEtech_uHiL_EN.pdf> [cit. 2011-03-11]

- **PROVEtech:R2A** (R2A je z angl.: Requirements to architecture) – nástroj, který se zaměřuje na integraci požadavků a nástrojů modelování, pro zajištění dohledatelnosti (angl.: traceability) během procesu vývoje elektronických systémů.
- **PROVEtech:VA** (angl.: Vehicle Application) – umožňuje pracovat s vnitřními hodnotami řídicí jednotky, tedy jejich záznam, eventuelně změnu během testování.
- **PROVEtech:VL** (angl.: Visual Loop) – je platforma pro testování kamerových asistenčních systémů v laboratorních podmínkách pomocí trojrozměrné virtuální reality.
- **PROVEtech:TP5** (angl.: Test process in five steps) – není softwarovým produktem ve smyslu, že by šlo o spustitelný program. Jedná se o metodologii a sadu šablon dokumentů pro procesní řízení a správu projektů testování elektronických systémů v pěti krocích: strategie – plánování – specifikace – realizace – vyhodnocení. Na základě těchto postupů lze včas odhalit a vyhodnotit rizika v procesu vývoje a testování řídicí jednotky.

6.3 Alternativní řešení pomocí nekomerčních programů

6.3.1 Programy pro modelování systémů

Komerční programy pro modelování systémů jako Matlab a Simulink nemají v profesionální průmyslové praxi příliš konkurenci. Mají velkou uživatelskou základnu, profesionální podporu jak od výrobce, tak distributorů, tak i od společností tzv. třetí strany. Nicméně pořizovací náklady jsou značné, v řádech statisíců korun. Z těchto cenových důvodů může být pro některé aplikace výhodné použití alternativního programu, který je k dispozici zdarma. Jedná se o programy s tzv. otevřenými zdroji (jsou k dispozici zdrojové texty programu), které bývají šířeny zdarma pod nějakou verzí licence GPL nebo její alternativou.

Alternativou k programu Matlab je program **Scilab** [15], který vyvíjí instituce INRIA (fran.: Institut national de recherche en informatique et en automatique – Francouzský národní institut pro výzkum počítačů a řízení). Součástí instalačního balíku je i program **Scicos**, který je alternativou ke grafickému simulačnímu nástroji Simulink. Vzhledem k problémům se stabilitou Scilabu verze 5, byla během druhé poloviny roku 2009 vytvořena odnož na základě předcházející verze Scilabu 4.1.2 s názvem

ScicosLab [16], jenž v sobě zahrnuje aktuálně poslední verzi Scicosu 4.4. Scicos však zůstal součástí Scilabu, kde od verze 5.2 byl přejmenován na Xcos a přinesl s sebou nový editor pro grafický návrh modelu a možnosti modelování mechanických, hydraulických a řídicích systémů.

Kromě zmatků v jednotlivých verzích, což je jisté specifikum komunitě vytvářených projektů s otevřenými zdroji, má prostředí Scilab i ScicosLab jisté výhody. Pomineme-li nulové pořizovací náklady, je dále nutné vyzdvihnout možnost použití modelů popsaných v jazyce Modelica (viz konec kapitoly 4.2), což velmi rozšiřuje modelovací schopnosti diskutovaných programů. V knihovných bloků pro Scicos je obsažen speciální blok umožňující propojení s částí modelu popsanou v Modelice.

Další informace lze čerpat i v knize, která se používáním zmíněných alternativních nástrojů zabývá [17].

6.3.2 Programy pro implementaci HiL testovací platformy

Pro zprovoznění HiL testovací platformy založené na běžném počítači typu PC doplněném o vstupně výstupní kartu, např. PCI-6229 již zmíněnou v kapitole 6.2.1, je zapotřebí programové vybavení, které umožní běh modelu v reálném čase. Toho lze docílit použitím různých proprietárních řešení. Společnost MathWorks jej nabízí pod názvem **xPC Target** jako volitelný doplněk balíku Matlab/Simulink. Další možností je využít alternativní řešení s využitím systémů s otevřenými zdroji.

Mezi otevřenými řešeními převažují zejména ta, která jsou založena na operačním systému Linux s patřičně upraveným jádrem systému (angl.: kernel). Jedním z řešení je např. **RTAI-Lab**. To je balík nástrojů umožňující překlad modelů primárně z programu ScicosLab a jejich provoz v reálném čase, nicméně je připraven i na zpracování modelů vytvořených v prostředí Matlab/Simulink, což by mohlo usnadnit jeho prosazování v profesionální oblasti. Balík v sobě zahrnuje rozhraní RTAI (angl.: realtime application interface), které přináší záplatu pro jádro operačního systému Linux rozšiřující abstraktní vrstvu hardwaru (HAL – angl.: hardware abstraction layer) a další služby usnadňující programování aplikací. Dále paletu bloků pro ScicosLab, monitorovací aplikaci pro interakci s aplikací běžící v reálném čase využívající protokol pro vzdálené volání funkcí (RPC – angl.: remote procedure call). Pro přístup k vstupně/výstupním rozhraním se využívá ovladačů z balíku **Comedi** (angl.: Control and Measurement Interface) [35]. Podrobný postup instalace, konfigurace jednotlivých nástrojů je uveden v tutoriálu RTAI-Lab [33].

6.4 Programy pro podporu procesu testování

Testování elektronických systémů automobilů nesouvisí jen s HiL testovací platformou a modelováním okolí pro řídicí jednotky. Jedná se komplexní proces, který lze do určité míry automatizovat.

Automatické provádění testování lze považovat za první stupeň automatizace procesu testování. Motivací pro implementaci automatického provádění je, kromě snížení objemu rutinní práce, také splnění požadavku na opakovatelnost testu. Při ručním testování vyžaduje dosažení přijatelné opakovatelnosti testování zvýšenou disciplínu testujících pracovníků. Pro automatizaci provádění testování je nutné použít další programové nástroje. Výše již byly uvedeny nástroje **AutomationDesk** zmíněný v kapitole 6.2.2 a **PROVEtech:TA** (v kapitole 6.2.3), v němž se automatizace provádí pomocí skriptů v jazyce WinWrap Basic. Dále je vhodné zmínit systém **EXAM** společnosti MicroNova, který je sice dostupný zdarma, ale bez knihoven a případná podpora ze strany společnosti je velmi nákladná. EXAM je prostředí založené na UML a testovací případ se graficky modeluje sekvenčním diagramem. Rovněž společnost National Instruments nabízí program pro automatizaci provádění testování s názvem **TestStand**. V něm se testovací sekvence vytváří grafickým způsobem s tím, že je možné využít funkcí vytvořených v Labview, v jazycích kompatibilních s .NET frameworkem, v jazyce C (knihovna .dll), v jazyce HTBasic, případně využít volání funkcí přes ActiveX/COM rozhraní.

Dalším stupněm je generování testů (testovacích případů) s využitím intuitivních či exaktních metod, z nichž některé jsou stručně popsány v kapitole 2.4. Podobně jako lze pomocí D-algoritmu generovat vstupní testovací vektory pro číslicové obvody, lze generovat testovací případy pro funkční testování řídicí jednotky při požadovaném pokrytí testu.

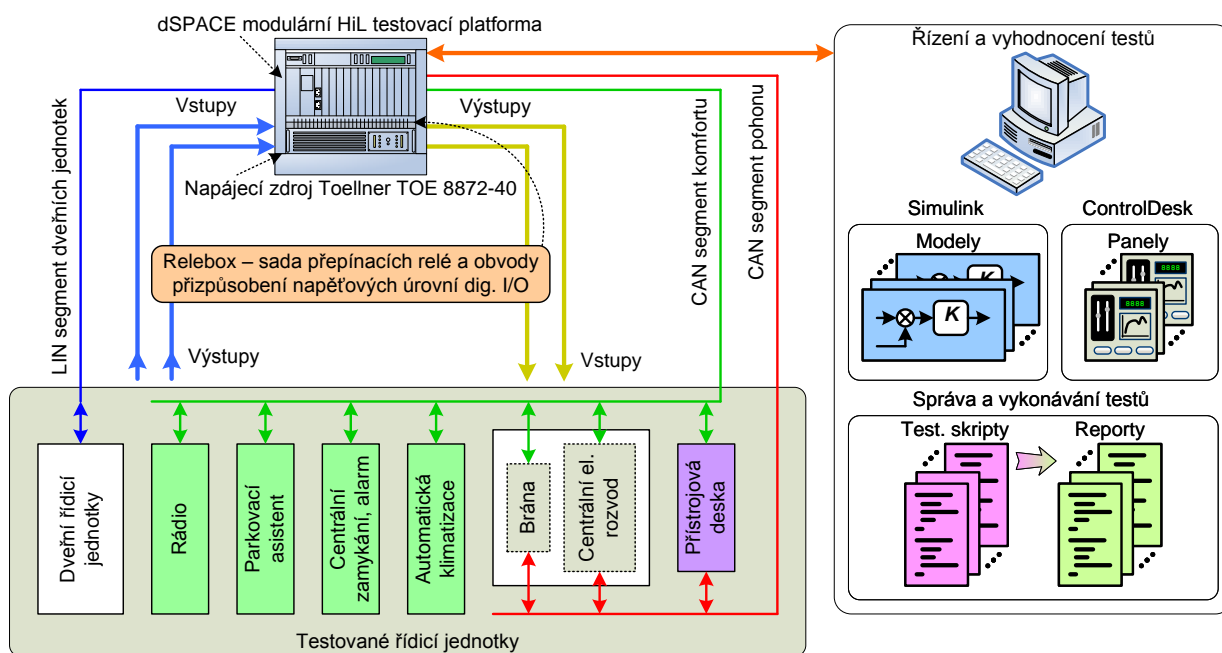
Proces testování je nutné také nějakým způsobem řídit. Na automatizaci testování se tak vážou další nástroje např. pro zprávu požadavků (angl.: requirement management) a řízení projektů. Mezi nástroje pro správu požadavků patří systém **Rational DOORS** společnosti IBM, který je však velmi nákladný na pořízení i správu a je tak vhodný pro velké projekty, nicméně v automobilovém odvětví je považován za průmyslový standard. Mezi systémy s otevřenými zdroji patří systém **Redmine**, který je naopak dostupný zdarma a lze jej rozšířit pomocí asi 150 zásuvných modulů. Vlastní upravenou verzi systému Redmine pod názvem KDP využívá např. společnost Kontron

pro řízení projektů vývoje a testování svých produktů, což jsou zejména průmyslové počítače typu PC. Pro správu testovacích případů je možné použít i systém **TestLog** společnosti PassMark, který umožňuje jejich definici, strukturalizaci, záznam výsledků testů (prošel/neprošel) a následně vytvoření reportů z průběhu testování.

Jak vyplývá z předchozích odstavců problematika kolem automatizace a řízení testování je natolik rozsáhlá, že zcela přesahuje rozsah této práce.

6.5 Konfigurace vybraného testovacího systému

Vzhledem k nutnosti zachování kompatibility s nadřazeným subjektem, a to jak z pohledu formátu vytvářených modelů tak způsobu řízení provádění testů, byl zvolen testovací systém společnosti dSPACE. Nicméně, jak z předcházejícího výčtu vyplývá, systémy dSPACE patří k těm nejvýkonnějším, co se výpočetního výkonu týká, a jsou i nejvíce orientovány na automobilový průmysl, kde se vyskytuje jmenovité napětí 12 V na rozdíl od úrovní TTL, či menších. Z výběrového řízení jaký testovací systém zvolit, se stala jen volba konfigurace periférií modulárního systému dSPACE.



Obr. č. 6.10 Struktura testovacího systému dSPACE pro testování vybraných komfortních řídicích jednotek

Předmětem testování je několik jednotek propojených sběrnicí CAN. Popis testovaného objektu se nachází v kapitole 5.3. Pro volbu konfigurace testovacího systému je nutné znát počet a typ všech požadovaných vstupů a výstupů testovaného

objektu, resp. výstupů a vstupů testovacího systému – objekt se systémem je zapojen ve smyčce, jak je znázorněno na obr. č. 6.10.

V následující tab. č. 6.2 jsou uvedeny celkové počty vstupů a výstupů testovacího systému, který byl sestaven na základě požadavků na počty vstupů a výstupů testovaného objektu. Testovací systém byl nakonfigurován s jistou rezervou, jen u odporových výstupů je dle tabulky rezerva nulová. To je způsobeno tím, že celkový počet vstupů a výstupů byl získán sečtením pro nejrozsáhlejší konfiguraci testovaného objektu, která není realizovatelná, protože některé řídicí jednotky se mohou vyskytovat v konfiguraci pouze v jedné verzi.

Tab. č. 6.2 Přehled vstupů a výstupů testovaného objektu a testovacího systému

	HDIN	HDOUT	AIN	AOUT	PWMIN	PWMO	ROUT	CAN	WGEN	WCAPT	LIN	DIN	DOUT	Cena Kč bez DPH
Objekt celkem	46	1	18	9	12	12	30	2	5	5	1	12	56	
Směr (objekt ◀▶ tester)	▼	▲	▼	▲	▼	▲	▲	▲▼	▲	▼	▲▼	▼	▲	
DS2211- HiL karta	16	16	16	20	8	9	10	2						377.100,-
DS2211- HiL karta	16	16	16	20	8	9	10	2						377.100,-
DS2211- HiL karta	16	16	16	20	8	9	10	2						377.100,-
DS4003 - 96 dig. I/O												24	72	30.500,-
DS4330 - max.16 LIN											4			129.500,-
DS1005 - Procesorová karta														259.000,-
DS4002 - TTL I/O a výst. sig.									8			16	16	68.600,-
DS5001- Vstup signálů										16				85.700,-
Tester celkem	48	48	48	60	24	27	30	6	8	16	4	40	88	1.704.600,-
Rezerva	2	47	30	51	12	15	0	4	3	11	3	28	32	

Vysvětlivky

HDIN – digitální vstup 0 až 60 V

HDOUT – digitální výstup 0 až 60 V

AIN – analogový vstup

AOUT – analogový výstup

PWMIN – vstup PWM modulace

PWMO – výstup PWM modulace

ROUT – odporový výstup

CAN – rozhraní sběrnice CAN (kanály)

WGEN – výstup generátoru časových průběhů

WCAPT – vstup pro zachytávání časových průběhů

LIN – rozhraní sběrnice LIN (kanály)

DIN – digitální vstup TTL

DOUT – digitální výstup TTL

Směrovost, čili zda se jedná o vstup či výstup, je v tabulce brána pouze z pohledu testovacího systému a je znázorněna šipkami. Uvedené ceny jsou z roku 2009.

K celkovým nákladům je nutné přičíst náklady na box pro karty ve výši 125.700,- Kč a skříň cca 20.000,- Kč.

Dále je nutné započíst náklady na napájecí zdroj ve výši 100.000,- Kč. Byl vybrán zdroj firmy Toellner TOE 8872-40, který dává na výstupu napětí max. 40 V, max. proud 100 A, při max. výstupním výkonu 1,5 kW.

Další částku v pořizovacích nákladech představuje rozhraní Relebox, konkrétně 85.000,- Kč. Relebox obsahuje sadu relé s přepínacím kontaktem a dále převodníky napěťových úrovní s galvanickým oddělením pro přizpůsobení úrovní v automobilu 0-12V úrovní TTL, se kterými pracuje karta DS1005. Relebox je pro některé signály zařazen do smyčky mezi HiL testovací platformu a testované řídicí jednotky, jak je znázorněno na předcházejícím obr. č. 6.10.

Nesmí se zapomenout na náklady za příslušné licence na programové nástroje (Matlab/Simulink a nezbytné volitelné nástroje, vývojové prostředí pro simulátory dSPACE – knihovny RTI, aplikace ControlDesk, překladač pro procesorovou kartu) a rovněž na náklady spojené s kompletací a instalací systému. Takže celkové náklady na tento testovací systém přesáhly 3 mil. Kč bez DPH.

7. Implementace systému dSPACE pro HiL testování

7.1 Nastavení a práce s vývojovými nástroji

Vytvoření a nahrání firmware testovacího systému

Firmware se vytváří v prostředí Matlab Simulink sestavením modelu ze simulinkovských bloků, které provádí různé matematické a jiné operace. Pokud chceme, aby model byl použitelný jako firmware pro dSPACE simulátor, je nutné pomocí příkazu `set_rti` na příkazovém řádku v Matlabu aktivovat RTI knihovny pro přístup k perifériím simulátoru a provést nastavení simulačních parametrů modelu (pro správnou funkci je nutné nastavit pevný simulační krok, nekonečnou délku simulace a nakonfigurovat generátor zdrojových kódů a překladač pro cílovou platformu – PowerPC).

Vyžaduje-li model implementaci specifické funkce, je možné pomocí speciálního bloku vytvořit tzv. s-funkci, která se naprogramuje v jazyce C. Vytvořený zdrojový kód je pak nutné přeložit pomocí příkazu `mex <sfunkce.c>`, aby došlo k propojení s grafickým prostředím Simulinku a s-funkce se stala součástí modelu.

Před prvním překladem modelu je ještě nutné upřesnit délku simulačního kroku. V případě automobilových komfortních systémů se zpravidla volí 1 ms. Dále zvolit, zda se má model po přeložení také automaticky nahrát do simulátoru. Překlad modelu se následně vyvolá pomocí klávesové zkratky `Ctrl+b` stisknuté při aktivním libovolném okně s modelem. Model lze přeložit také pomocí příkazu z příkazového řádku. Příkaz má následující tvar:

```
rti_build('<název modelu>', 'Command', 'CML')
```

kde písmena parametru 'CML' mají význam jako následující příkazy: C – vygenerování zdrojových souborů z modelu; M – překlad zdrojových souborů; L – nahrání přeloženého modelu do simulátoru. Příkazy lze kombinovat tak, že lze libovolnou fázi vynechat a provést tak např. jen nahrání již dříve přeloženého modelu do simulátoru.

Vytvoření ovládacího a vizualizačního panelu v aplikaci ControlDesk

Jako první se v aplikaci ControlDesk založí nový experiment. Pak se otevře nový panel (angl.: layout), který se zařadí do experimentu pomocí příkazu „add to experiment“. Na panelu se pak vytvoří grafické uspořádání zobrazovacích a ovládacích prvků, včetně konfigurace rozsahů hodnot apod. Dále se otevře soubor <název

modelu>.sdf, který byl vytvořen během překladač a obsahuje popis struktury modelu (sdf – angl.: system definition file).

Ve zvláštním okně je pak vidět struktura modelu a symbolicky se zobrazují hodnoty konstant (blok konstanta) a výstupy všech bloků. Odkaz na hodnotu konstanty lze metodou táhni a pust' přetáhnout na ovládací prvek na panelu, tím dojde k propojení hodnoty v simulátoru s ovládacím prvkem. Podobně je tomu s výstupy bloků a se zobrazovacími prvky. Lze pak sledovat i vnitřní hodnoty modelu v reálném čase.

7.2 Modely okolí pro vybrané řídicí jednotky

Dalším úkolem bylo implementovat HiL testování pro integraci komfortních řídicích jednotek vozu nižší třídy, viz struktura datové sítě na obr. č. 5.1. Do testování byly zahrnuty jednotky vyjmenované v úvodu kapitoly 5.3. Struktura testovacího systému včetně vybraných jednotek byla znázorněna na obr. č. 6.10.

Vzhledem k tomu, že jednotky jsou testovány v laboratorním prostředí, je snaha, aby pro testování bylo potřeba co nejméně vnějšího hardwaru, zejména senzorů a akčních členů. Testované jednotky jsou jen navzájem propojeny sběrníkovými vodiči a připojeny na napájení. Ostatní vývody – vstupy a výstupy, jsou připojeny k testovacímu systému (simulátoru) popsanému v předcházející kapitole 6.5. Pro bezchybnou činnost řídicích jednotek je nutné simulovat jejich okolí. U komfortních systémů má nejsložitější okolí řídicí jednotka klimatizace – chladicí okruh a rozvod vzduchu ve voze, nejrozsáhlejší pak centrální elektrický rozvod – má nejvíce různých funkcí. Dále u parkovacího asistenta je nutné simulovat vzdálenost senzorů od překážky, u jednotky centrálního zamykání zámky, čidla pro vyvolávání alarmu a výstupní indikaci narušení bezpečnosti. Výjimku tvoří dveřní jednotky, které vyžadují mechanickou vazbu na stahovací okno. Proto bylo třeba vyřešit snímání polohy okna jako zpětnou vazbu pro testování.

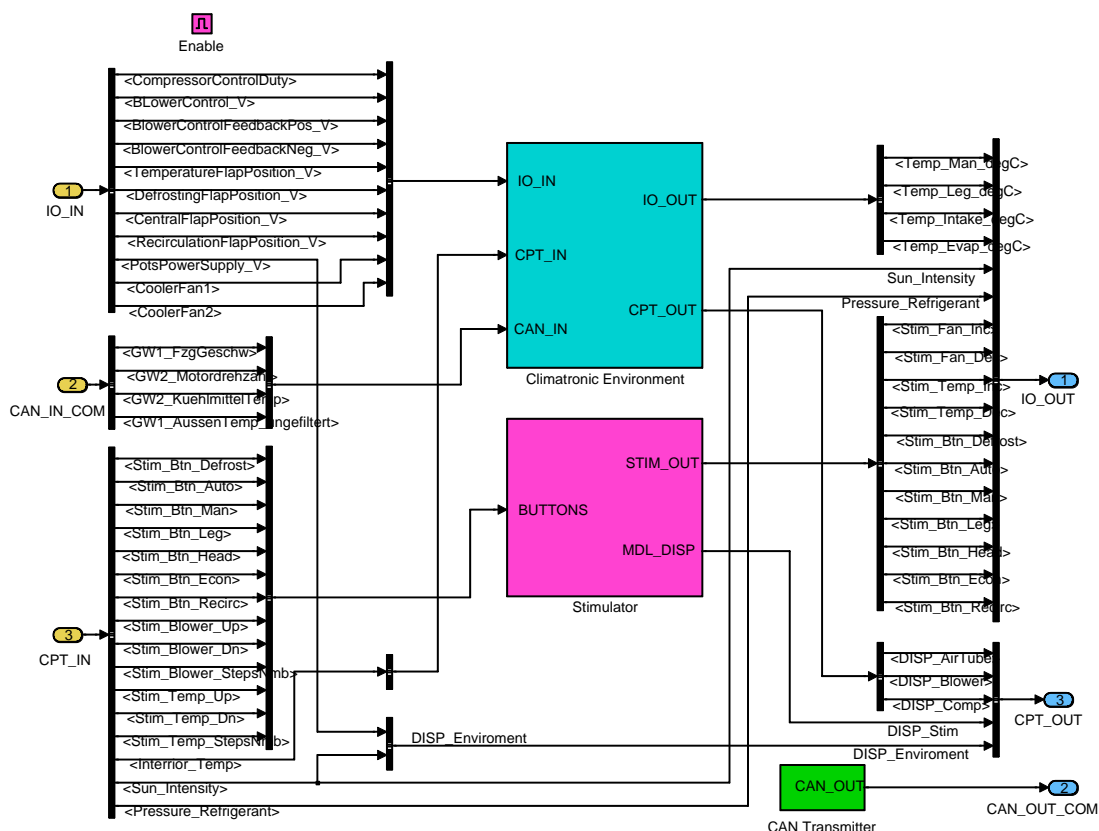
V rámci této práce byly vytvořeny modely okolí k automatické klimatizaci a centrálnímu zamykání. Model klimatizace vzhledem ke své složitosti a zkrácení doby vývoje vznikl v kooperaci s kolegou, viz [24], autorský podíl je 50% každého z obou autorů.

Díličí modely okolí byly vytvořeny podle šablony, která byla definována (v rámci [24]) pro nastavení štabní kultury celého modelu tak, aby se v něm dalo snadněji orientovat. Každý model má vstupy IO_IN, což jsou signály přivedené ze vstupů simulátoru, CAN_IN_*, obsahující informace ze sběrnice CAN a CPT_IN, které

jsou navázány na ovládací prvky v ControlDesku. Výstupy jsou strukturovány obdobně. IO_OUT jsou vyvedené na výstupy simulátoru, CAN_OUT_* sdužuje informace, které se mají vyslat na sběrnici CAN, a CPT_OUT a MDL_DISP jsou sběrnice signálů pro vizualizační prvky v ControlDesku, pro zobrazení vnitřních informací a stavů modelu.

7.2.1 Model okolí řídicí jednotky automatické klimatizace

Model pro řídicí jednotku automatické klimatizace se je znázorněn na obr. č. 7.1. Skládá se z modelu vlastního okolí (Climatronic Environment) a pak z podpůrného modelu, jenž zajišťuje ovládání mechanického stimulatoru (Stimulator).

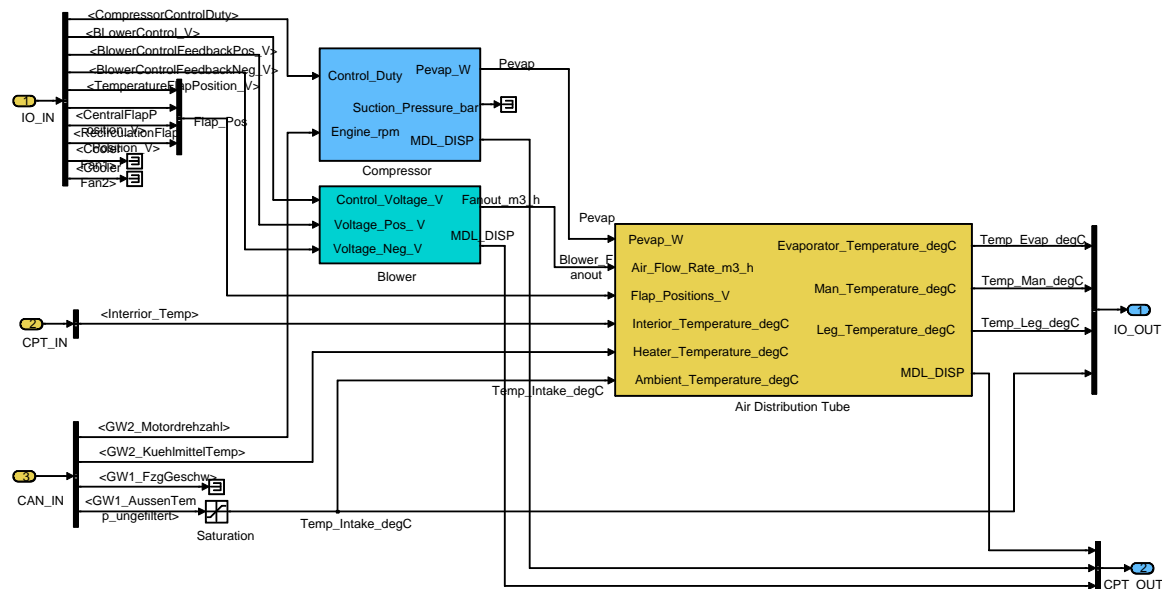


Obr. č. 7.1 Model okolí pro automatickou klimatizaci – okolí a stimulator

Blok modelu vlastního okolí řídicí jednotky

Vlastní model okolí řídicí jednotky znázorněný dále na obr. č. 7.2 je sestaven z bloků kompresoru (Compressor), ventilátoru (Blower) a distribuce vzduchu (Air Distribution Tube). Z řídicí jednotky do modelu vstupují následující veličiny: střída signálu pro řízení kompresoru, řídicí a skutečné napětí ventilátoru, informace o sepnutí dochlazovacích ventilátorů a napětí odpovídající polohám klapky. Klapky nejsou simulovány, bylo rozhodnuto, že testování proběhne s fyzicky připojenými klapkami,

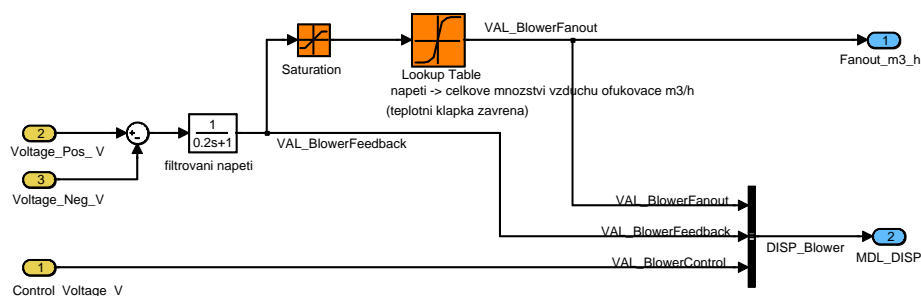
kteře jsou součástí distribuční soustavy vzduchu. Klapka je ovládaná stejnosměrným motorem a jako zpětnou vazbu polohy je vybavena potenciometrem, zapojeným jako napěťový dělič a výstupní napětí tedy odpovídá poloze klapky.



Obr. č. 7.2 Struktura modelu okolí – kompresor, ventilátor a distribuce vzduchu

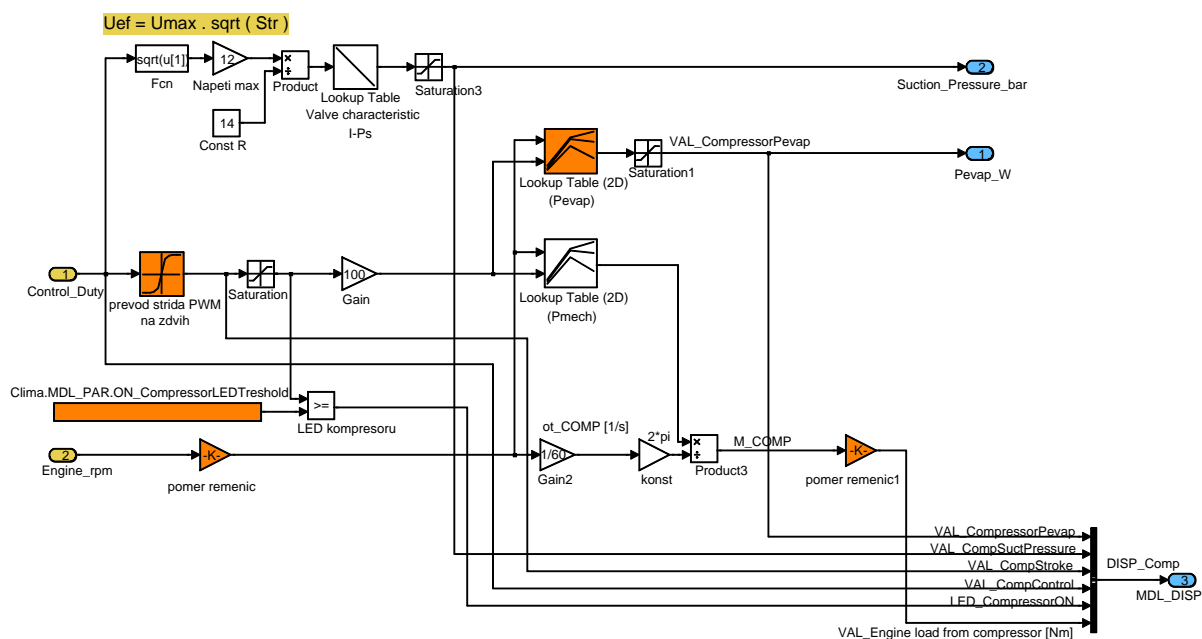
Jak vyplývá z obr. č. 7.2 informace o sepnutí dochlazovacích ventilátorů nakonec není využita. Další vstupní informace potřebné pro simulaci, jako jsou otáčky motoru, teplota chladicího okruhu motoru a okolní teplota vzduchu vně vozidla, jsou získávány ze sběrnice CAN. Konkrétně ze zpráv vysílaných bránou, která propojuje jednotlivé segmenty datové sítě.

Výstupem z modelu jsou vypočtené teploty, které by měřily příslušné senzory. Řídící jednotka používá odporové senzory teploty se záporným teplotním koeficientem (NTC). Teplota je tedy převedena na hodnotu odporu podle charakteristiky příslušného senzoru s využitím Steinhart-Hartovy aproximace NTC. Senzory jsou nahrazeny odporovými výstupy HiL platformy a jsou zapojeny přímo na vstup ECU.



Obr. č. 7.3 Blok ventilátoru

Model ventilátoru (viz obr. č. 7.3) je relativně jednoduchý a má za úkol přepočítat velikost svorkového napětí na odpovídající proud vzduchu. Jako podklad se podařilo získat naměřené objemové množství vzduchu při různém svorkovém napětí. Z naměřených údajů pak byla vytvořena vyhledávací tabulka Lookup Table, která je jádrem modelu ventilátoru. Před tabulkou je zařazen blok saturace pro omezení hodnot vstupního napětí mezi 0 až 14 V. Jak již bylo zmíněno v kapitole 4.2.1 vyhledávací tabulka umožňuje modelovat statické chování systému. Pro zavedení alespoň náznaku dynamického chování byl do modelu zařazen filtr typu dolní propust s časovou konstantou 0,2 s a je jím filtrováno svorkové napětí ventilátoru.

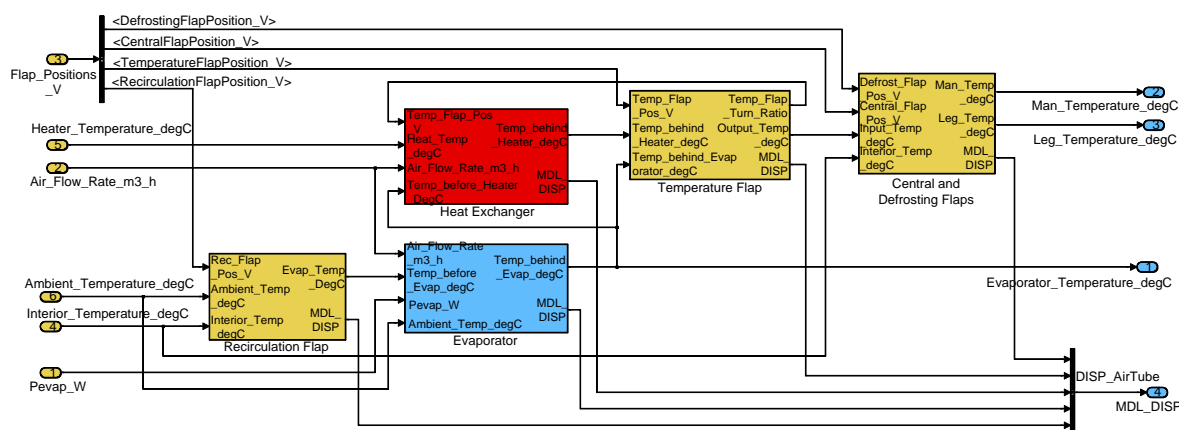


Obr. č. 7.4 Blok kompresoru

Blok kompresoru (na obr. č. 7.4) společně s blokem výparníku (viz dále) tvoří hlavní částí chladicího okruhu, jak vyplývá z obr. č. 5.4. Při modelování chladicího okruhu byl velký problém se získáváním podkladových informací. Nakonec se sice podařilo získat naměřenou závislost chladicího výkonu výparníku v závislosti na otáčkách kompresoru a jeho zdvihovém objemu, ze které byla vytvořena dvourozměrná vyhledávací tabulka Lookup table (2D) (Pevap), ale opět se jedná o statickou charakteristiku. Absence informací o dynamice systému s sebou přinesla problémy při ladění modelu ve spojení s řídicí jednotkou. Otáčky kompresoru jsou přepočítávány z otáček motoru a převodového poměru řemenic. Kompresor je typicky poháněn od klikového hřídele motoru pomocí plochého drážkovaného řemene. Aktuální zdvihový objem kompresoru je získáván ze střídavy řídicího PWM signálu. Střídou PWM signálu se reguluje střední hodnota proudu v elektromagnetu, který pak mechanicky nastavuje

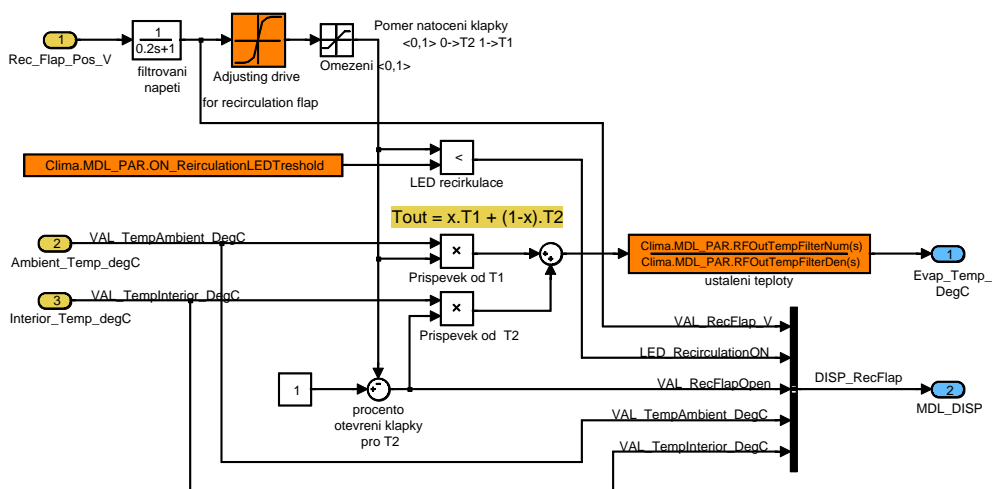
zdvihový objem kompresoru. Je tedy použita další vyhledávací tabulka s příslušným omezením výstupní hodnoty blokem Saturation a vynásobením v bloku Gain pro převod střídy na zdvihový objem tak, aby rozsah hodnot zdvihu byl 0 až 100 [%].

V bloku kompresoru zůstala i původní myšlenka, kdy se ze střídy PWM signálu postupně určil sací tlak (na obr. č. 7.4 v levé horní části), nicméně pro nedostatek dalších informací, nutných k doladění modelu touto cestou, není nakonec hodnoty sacího tlaku dále využito.



Obr. č. 7.5 Blok distribuční soustavy vzduchu

Na obr. č. 7.5 se nachází vnitřní struktura bloku distribuční soustavy vzduchu (Air Distribution Tube). Vnitřní struktura bloku vychází ze struktury distribuční soustavy, která byla již dříve znázorněna na obr. č. 5.6. Vzduch prochází nejprve přes recirkulační klapku (blok Recirculation Flap, viz obr. č. 7.6). Ta svou polohou určuje, zda se bude vzduch nasávat z vnějšku nebo z vnitřního prostoru tam, kde má spolejzdec nohy, eventuelně v jakém poměru se budou teploty mísit.



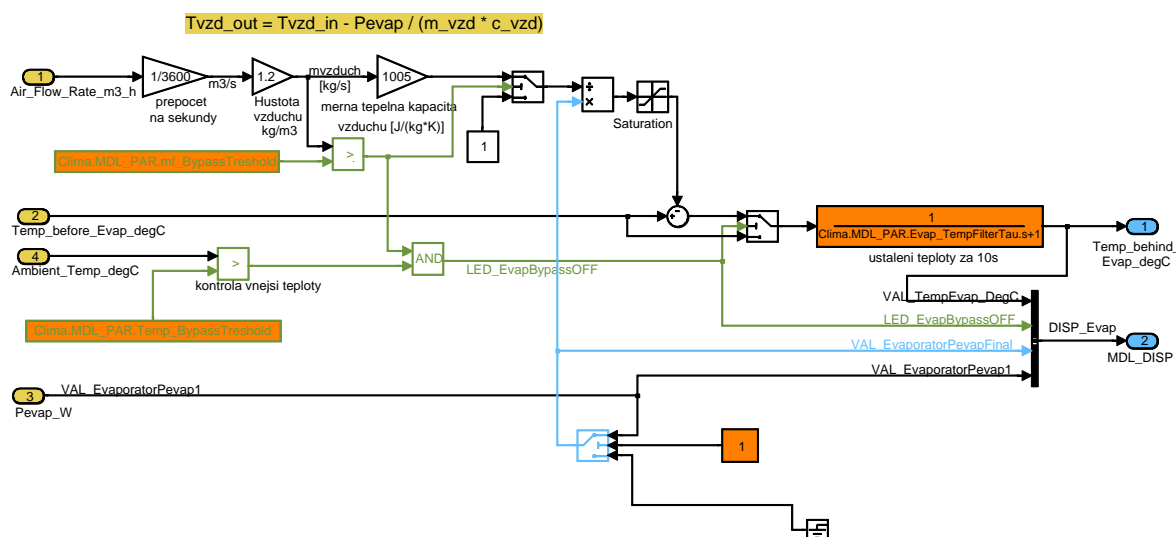
Obr. č. 7.6 Blok recirkulační klapky

Sice se recirkulační klapka ovládá tlačítkem na ovládacím panelu řídicí jednotky, takže je nastavována do krajních poloh, nicméně model by měl zvládnout simulovat obecnou polohu klapky v daných mezích. Vliv polohy klapky na výslednou teplotu nasávaného vzduchu je modelován lineární váhovou funkcí, podle následující rovnice (7.1), kde K_{RcFlap} je poměrné natočení klapky, T_{Amb} je teplota vzduchu vně vozidla a T_{Int} je teplota uvnitř.

$$(7.1) \quad T_{RcFOut} = K_{RcFlap} \cdot T_{Amb} + (1 - K_{RcFlap}) \cdot T_{Int} \quad [K; 1, K, K]$$

Poměrné natočení klapky je získáváno z napětí polohového potenciometru prostřednictvím vyhledávací tabulky (Adjusting drive). Vstupní napětí je filtrováno dolní propustí s časovou konstantou 0,2 s. Výstupní teplota je pak ještě filtrována dalším filtrem. Tento filtr je modelován blokem přenosové funkce (ustaleni teploty). Přenosová funkce je definována v podobě konstantních parametrů modelu (ClimaMDL_PAR...). Podobně jsou v modelu definovány i další filtry pro zavedení dynamiky.

Z recirkulační klapky pak vzduch pokračuje dále přes ventilátor na výparník. Z bloku ventilátoru (Blower) již vstupuje do bloku distribuční soustavy informace o objemovém množství vzduchu, které ventilátor produkuje. Je však zanedbán vliv polohy recirkulační klapky a rovněž ztráty prouděním v distribuční soustavě nejsou uvažovány. Pro zjednodušení je uvažován průtok vzduchu, ať už objemový nebo přepočtený hmotnostní, v celé soustavě za stejný, určený pouze výkonem ventilátoru.



Obr. č. 7.7 Blok výparníku

Do modelu výparníku (Evaporator, viz obr. č. 7.7) tedy vstupuje hodnota objemového průtoku vzduchu. Ta je pro další výpočty přepočtena na hmotnostní průtok podle následující rovnice (7.2).

$$(7.2) \quad Q_m = \frac{Q_v \cdot \rho_{vz}}{3600} \quad [\text{kg} \cdot \text{s}^{-1}; \text{m}^3 \cdot \text{h}^{-1}, \text{kg} \cdot \text{m}^{-3}]$$

Kde ρ_{vz} je hustota vzduchu (uvažováno $1,2 \text{ kg} \cdot \text{m}^{-3}$). Výstupní teplota se pak určí podle následující rovnice (7.3), ve které T_{Ein} je vstupní teplota vzduchu, P_{Evap} je chladicí výkon výparníku a c_{vz} je měrná tepelná kapacita vzduchu (uvažováno $1005 \text{ J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$).

$$(7.3) \quad T_{Eout} = T_{Ein} - \frac{P_{Evap}}{Q_m \cdot c_{vz}} \quad [\text{K}; \text{K}, \text{W}, \text{kg} \cdot \text{s}^{-1}, \text{J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}]$$

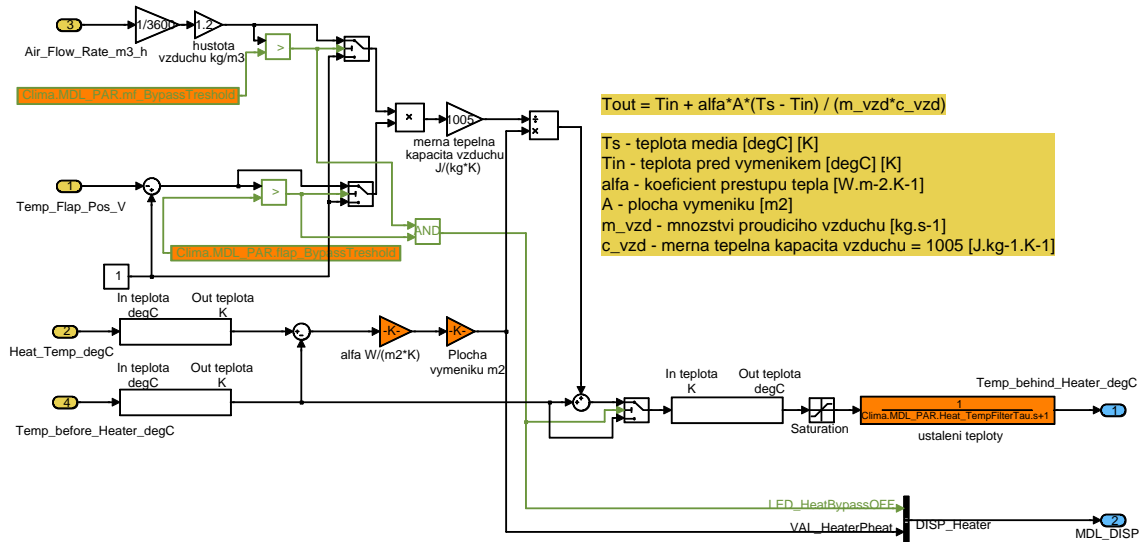
Jako hodnota vstupní teploty se použije výsledek z rovnice (7.1) $T_{Ein} = T_{RcFOut}$. V bloku výparníku jsou dále implementovány komparátory, které jednak zabraňují dělení nulou při nulovém průtoku vzduchu a rovněž zajišťují správnou deaktivaci chlazení při záporných hodnotách teploty (resp. menší teplotě než je zvolená hodnota parametru modelu `Clima.MDL_PAR.Temp_BypassTreshold`). Uvedeným postupem lze sice získat hodnotu teploty vzduchu za výparníkem, nicméně není zajištěná správná dynamika procesu chlazení. Ta je modelována pouze vřazením filtru prvního řádu typu dolní propust pro filtrování výstupní teploty (blok `ustaleni teploty...`). Časová konstanta byla doladěna při testování modelu ve spojení s řídicí jednotkou tak, aby nedocházelo k nereálným výsledkům teploty, při různých změnách otáček motoru a nastavení na řídicí jednotce.

Vychlazený vzduch dále pokračuje na teplotní klapku, která podle svého natočení rozdělí proud vzduchu tak, že část prochází dále distribučním soustavou vzduchu a část prochází tepelným výměníkem topení, kde je ohříván, a následně se mísí s původním vzduchem přepouštěným mimo topení. V krajních polohách teplotní klapka tedy buď přepouští všechny vzduch mimo topení, nebo naopak nasměruje veškerý tok vzduchu na tepelný výměník. Bloky tepelného výměníku (Heat Exchanger) a teplotní klapky (Temperature Flap) jsou proto spolu více provázané.

Do bloku tepelného výměníku topení (Heat Exchanger, obr. č. 7.8) vstupují tedy teplota výměníku (nastavuje se vně modelu během simulace), proud vzduchu, teplota vzduchu z výparníku a přepočtená poloha teplotní klapky z příslušného bloku.

Topný výkon výměníku se vypočítá podle následující rovnice (7.4), kde T_{Hin} je teplota vstupujícího vzduchu, tedy teplota vzduchu z výparníku $T_{Hin} = T_{Eout}$, T_{Heat} je teplota výměníku, α je koeficient přestupu tepla a A_{exchg} je efektivní plocha výměníku.

$$(7.4) \quad P_{Heat} = (T_{Hin} - T_{Heat}) \cdot \alpha \cdot A_{exchg} \quad [\text{W}; \text{K}, \text{K}, \text{W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}, \text{m}^2]$$

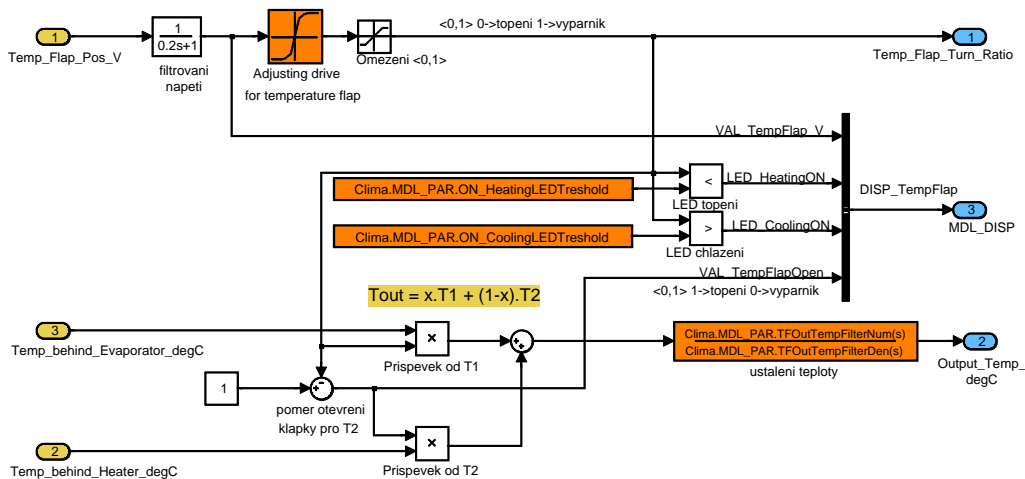


Obr. č. 7.8 Blok tepelného výměníku topení

Koeficient přestupu tepla a efektivní plocha výměníku byly převzaty z jeho technické dokumentace, která podléhá obchodnímu tajemství, z tohoto důvodu je zde nelze publikovat. Výstupní teplota vzduchu za topením se vypočte podle rovnice (7.5), kde P_{Heat} je topný výkon výměníku z předchozí rovnice (7.4), $K_{TempFlap}$ je poměrné natočení teplotní klapky, které se vypočte v bloku teplotní klapky (Temperature Flap), dále Q_m je hmotnostní průtok vzduchu vypočtený z rovnice (7.2) a c_{vz} je měrná tepelná kapacita vzduchu.

$$(7.5) \quad T_{Hout} = T_{Hin} + \frac{P_{Heat}}{(1 - K_{TempFlap}) \cdot Q_m \cdot c_{vz}} \quad [K; K, W, kg.s^{-1}, J.kg^{-1}.K^{-1}]$$

Průtok vzduchu je korigován lineární váhovou funkcí podle poměrného natočení teplotní klapky – průtok je vynásoben výrazem $(1 - K_{TempFlap})$.



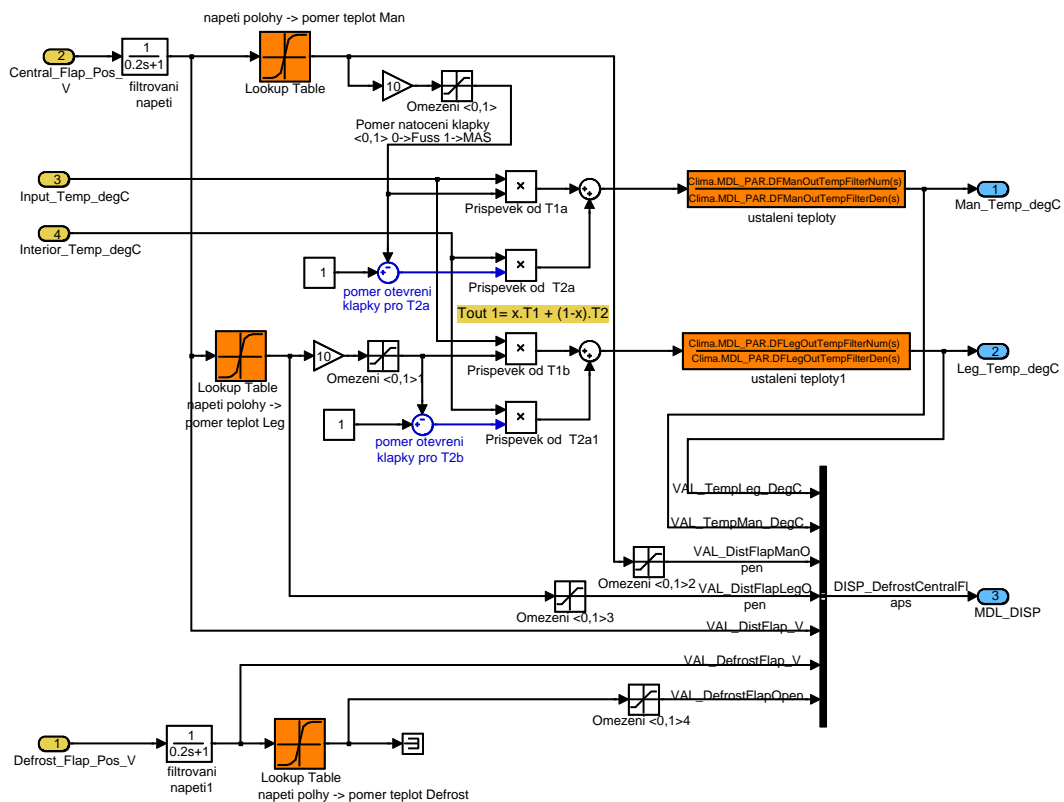
Obr. č. 7.9 Blok teplotní klapky

Informace o výstupní teplotě vzduchu z topení spolu s teplotou z výparníku a napětím z potenciometru polohy klapky vstupují do bloku teplotní klapky (Temperature Flap), který je znázorněn na obr. č. 7.9. Vlastní teplotní klapka je modelována podobně jako recirkulační. Výstupní teplota je určována opět lineární váhovou funkcí podle rovnice (7.6).

$$(7.6) \quad T_{TFOut} = K_{TempFlap} \cdot T_{Eout} + (1 - K_{TempFlap}) \cdot T_{Hout} \quad [K; 1, K, K]$$

Výstupem je rovněž relativní poloha klapky (Temp_Flap_Turn_Ratio), která je dále využívána v bloku tepelného výměníku.

Centrální klapka, která rozděluje proud vzduchu do horních výstupních ofukovačů (na hlavu a tělo) a do dolních (na nohy), a odmrazovací klapka, jenž reguluje proud vzduchu na přední sklo, jsou obě modelovány jedním blokem Central and Defrosting Flaps. Jeho vnitřní struktura je znázorněna na obr. č. 7.10. Výstupem tohoto bloku jsou teploty, které by snímaly senzory umístěné v horním a dolním ofukovači.



Obr. č. 7.10 Blok centrální a odmrazovací klapky

Výstupní teploty se určí podle následujících rovnic (7.7) a (7.8).

$$(7.7) \quad T_{ManOut} = K_{DistManFlap} \cdot T_{TFOut} + (1 - K_{DistManFlap}) \cdot T_{Int} \quad [K; 1, K, K]$$

$$(7.8) \quad T_{LegOut} = K_{DistLegFlap} \cdot T_{TFOut} + (1 - K_{DistLegFlap}) \cdot T_{Int} \quad [K; 1, K, K]$$

Kde T_{ManOut} je teplota vzduchu horního ofukovače (v modelu označená jako Man_Temp_degC), T_{LegOut} je teplota vzduchu dolního ofukovače na nohy (v modelu označená jako Leg_Temp_degC), $K_{DistManFlap}$ a $K_{DistLegFlap}$ jsou váhové koeficienty získané z napětí polohových potenciometrů klapky pomocí vyhledávacích tabulek.

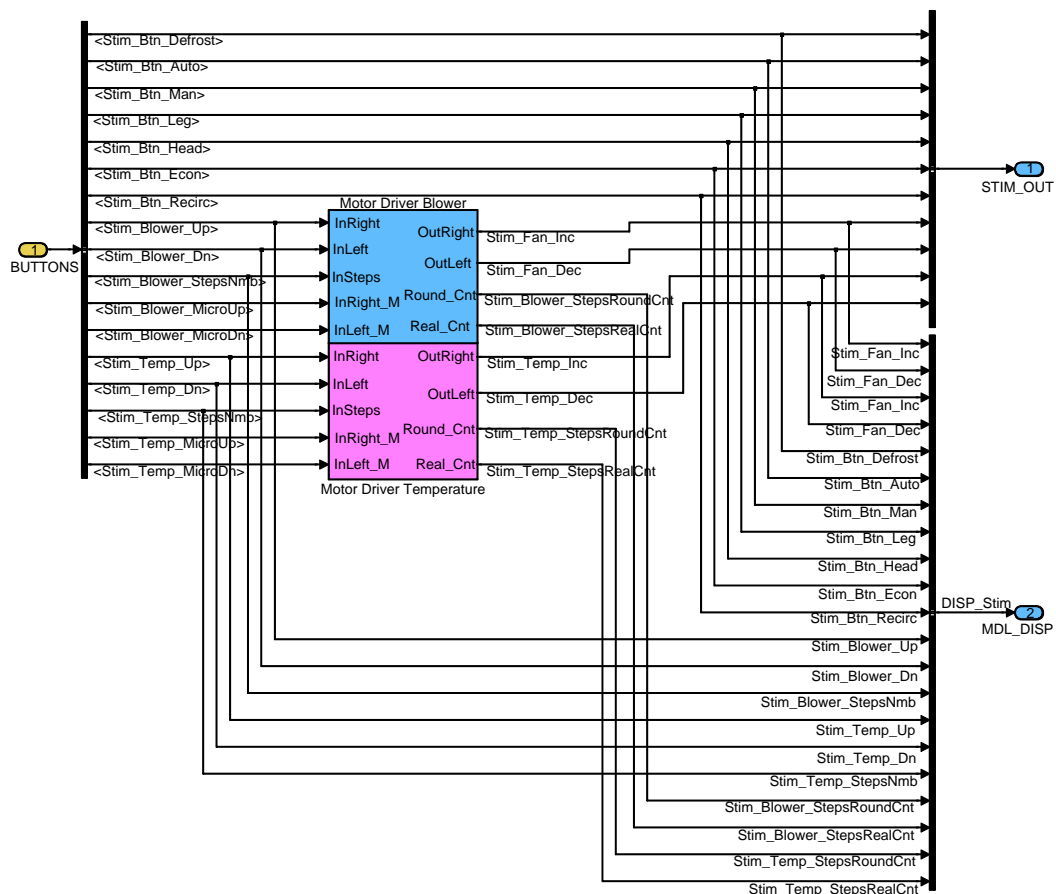
Vliv odmrazovací klapky na teploty není uvažován, takže je určena jen její relativní poloha pomocí vyhledávací tabulky.

Teploty jako výstupní veličiny bloků klapky, jsou na výstupu filtrovány příslušným filtrem. Tyto filtry jsou opět modelovány blokem přenosové funkce. Přenosové funkce jsou naladěny spolu filtrem chladicího výkonu výparníku tak, aby se zpětnovazební systém s uzavřenou smyčkou choval reálně a řídicí jednotka si neukládala chyby zjištěné vlastní diagnostikou. Cílem nebylo vytvořit termodynamický model proudění vzduchu v systému topení a klimatizace vozidla, ale model pro uzavření zpětnovazebních smyček řídicí jednotky. Z důvodu obtížného zjišťování některých podkladů je model velmi zjednodušený, resp. je zanedbána značná část ovlivňujících faktorů – proudění vzduchu, vliv poloh klapky je lineárně aproximován váhovými funkcemi, dynamika systému je zavedená uměle pomocí filtrů s odhadnutými přenosovými funkcemi na základě chování systému s uzavřenými zpětnými vazbami.

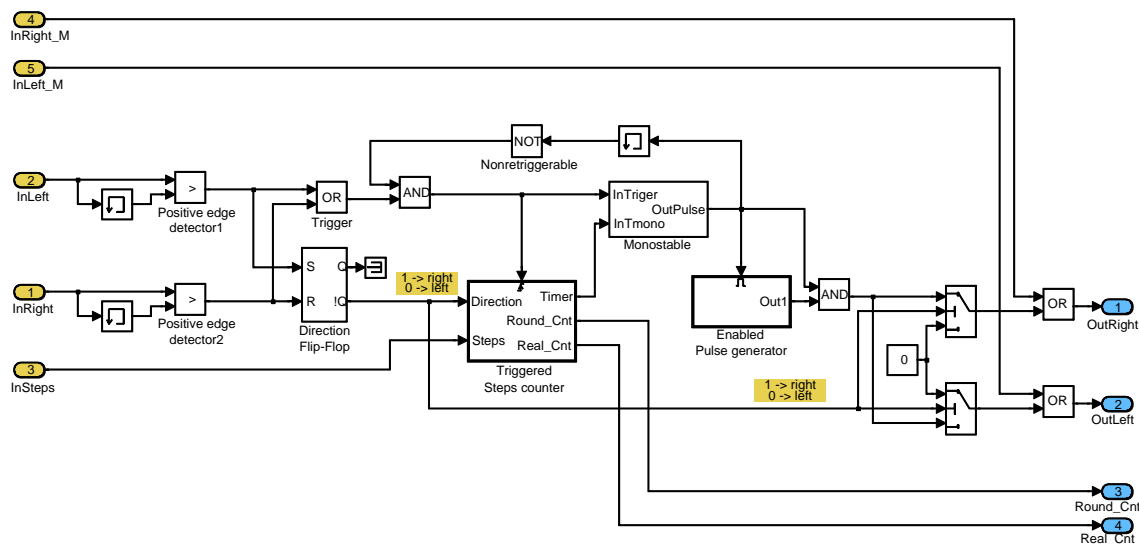
Blok pro řízení stimulátoru ovládacího panelu řídicí jednotky

Struktura bloku pro řízení stimulátoru ovládacího panelu řídicí jednotky automatické klimatizace je znázorněna na obr. č. 7.11. Stimulátor umožňuje stimulovat stisk tlačítek a otáčení rotačními ovladači pro nastavení požadované teploty a výkonu ventilátoru. Stimulátor se ovládá pomocí digitálních signálů z HiL testovací platformy.

Stisk tlačítek není třeba nijak modelovat, jsou to tedy jen procházející signály. Otáčení rotačními ovladači se provádí pomocí krokových motorů, které se řídí pulsními signály. Motor reaguje na sestupnou hranu signálu a otočí se o jeden fyzický krok odpovídajícím směrem. Pulsní signály jsou generovány prostřednictvím bloků Motor Driver Blower a Motor Driver Temperature, jejich vnitřní struktura je vidět dále na obr. č. 7.12.



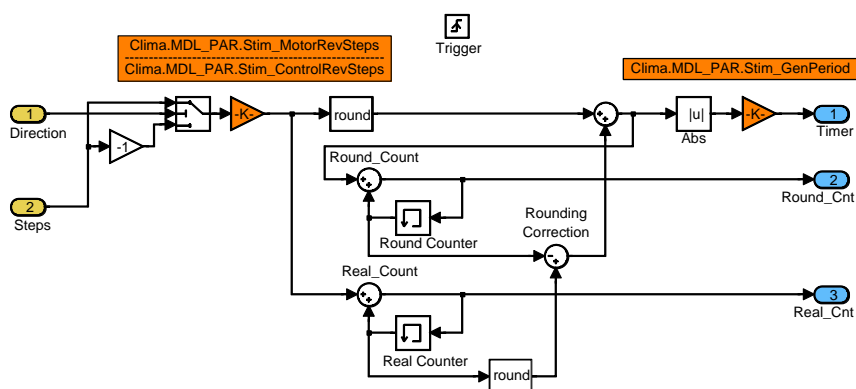
Obr. č. 7.11 Struktura bloku stimulatoru



Obr. č. 7.12 Blok budiče krokového motoru

Vstupem do bloku budiče motoru je počet kroků rotačního ovladače (InSteps), které se provedou na náběžnou hranu (stisk tlačítka) signálů InLeft a InRight příslušným směrem (doleva nebo doprava). Náběžné hrany jsou detekovány porovnáním aktuální hodnoty signálu s předešlou v bloku Positive edge detector. Pokud je aktuální hodnota

vyšší než minulá, jedná se o náběžnou hranu na vstupním signálu. Podle toho, na kterém vstupu je detekována náběžná hrana, je určen směr otáčení pomocí klopného obvodu (Direction Flip-Flop). Výstupy hranových detektorů jsou sloučeny logickým součtem v bloku Trigger a tvoří hlavní spouštěcí signál. Směr otáčení a počet kroků ovladače je zaveden do bloku čítače kroků (Triggered Steps Counter) spouštěného náběžnou hranou hlavního spouštěcího signálu, ve kterém se vypočte doba, po kterou se budou generovat pulsy pro řízení motoru. Struktura bloku čítače kroků je znázorněná na následujícím obr. č. 7.13.



Obr. č. 7.13 Blok čítače kroků

Vzhledem k nesoudělnosti počtu kroků motoru a počtu kroků rotačního ovladače na jednu otáčku je nutné provádět korekci chyby, která by vznikla zaokrouhlením jejich vzájemného poměru. Tato chyba by se projevila zejména při postupném otáčení o jeden krok ovladače. Např. při $K_{Ctrl} = 30$ kroků na otáčku ovladače a při $K_{Mot} = 200$ kroků na otáčku motoru je jejich poměr $6\frac{2}{3}$, po zaokrouhlení 7. Pro otočení o 3 kroky ovladače je třeba provést 20 kroků motoru, nicméně při postupném otáčení po 7 krocích motoru by vznikla chyba 1 krok motoru (celkem by se pootočilo o 21 kroků motoru). Čímž by postupně po jedné otáčce o 360° velikost chyby dosáhla hodnoty 10 kroků motoru, tedy více než jeden krok ovladače. Pro odhalení kumulativní chyby zaokrouhlování se provádí počítání provedených kroků motoru, tedy zaokrouhlený počet (Round Counter), a rovněž počítání kolik kroků se mělo provést, jako reálné číslo (Real Counter). Rozdíl mezi zokrouhleným počtem a reálným počtem, který se rovněž zaokrouhlí, je zaokrouhlovací korekce (Rounding Correction), která se přičte k aktuálně vypočtenému počtu kroků motoru, viz rovnice (7.9). Jak již bylo zmíněno, výstupem bloku čítače kroků je doba, po kterou se budou generovat pulsy pro řízení motoru T_{Timer} . Ta se vypočte podle následujících rovnice, kde T_{GenPer} je perioda jednoho generovaného

pulsu, S_{Mot} je počet kroků motoru, C_{Real} je počet kroků jako reálné číslo a C_{Round} je počet skutečných kroků motoru (zaokrouhlených).

$$(7.9) \quad T_{Timer}^N = T_{GenPer} \cdot \left| \text{rnd}(S_{Mot}^N) + \underbrace{\text{rnd}(C_{Real}^{N-1}) - C_{Round}^{N-1}}_{\text{Rounding Correction}} \right| \quad [s; s, 1, 1, 1]$$

Výpočet je diskrétní v čase, poněvadž je spouštěn náběžnou hranou hlavního spouštěcího signálu, a proto jsou aktuální hodnoty v časovém kroku N označeny horním indexem N a hodnoty z předchozího kroku indexem $N - 1$.

Aktuální počet kroků motoru S_{Mot} se vypočte podle následující rovnice (7.10). Výběr rovnice, tedy zda se bude násobit -1 nebo nikoliv, se provádí na základě požadovaného směru otáčení (vstup $Direction$). S_{Ctrl} je požadovaný počet kroků ovladače a konstanty K_{Mot} a K_{Ctrl} vyjadřují počet kroků na otáčku motoru, resp. ovladače.

$$(7.10) \quad S_{Mot}^N = \begin{cases} S_{Ctrl}^N \cdot \frac{K_{Mot}}{K_{Ctrl}} \Big|_{Direction=1} \\ -1 \cdot S_{Ctrl}^N \cdot \frac{K_{Mot}}{K_{Ctrl}} \Big|_{Direction=0} \end{cases}$$

Aktuální hodnoty čítačů, které se uloží pro příští výpočet, se určí přičtením aktuálního počtu kroků motoru k předchozímu stavu čítačů, viz následující rovnice (7.11) a (7.12).

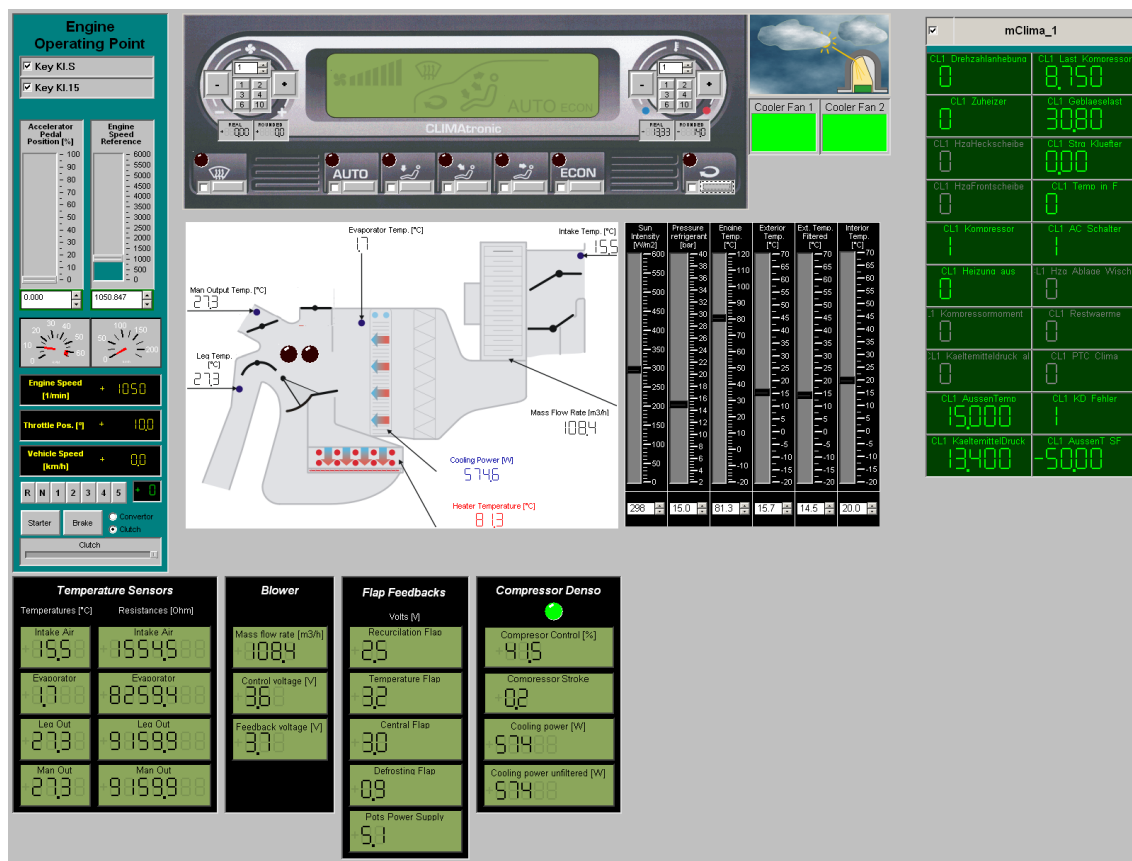
$$(7.11) \quad C_{Real}^N = C_{Real}^{N-1} + S_{Mot}^N$$

$$(7.12) \quad C_{Round}^N = C_{Round}^{N-1} + \text{rnd}(S_{Mot}^N) + \underbrace{\text{rnd}(C_{Real}^{N-1}) - C_{Round}^{N-1}}_{\text{Rounding Correction}}$$

Vypočtená hodnota doby T_{Timer} se použije pro nastavení doby kvazistabilního stavu monostabilního časovače (Monostable), který je spouštěn hlavním spouštěcím signálem. Monostabilní časovač na svém výstupu vytvoří signál pro povolení generátoru. Ten generuje pulsy s periodou T_{GenPer} po dobu co je povolen a tím dojde k vygenerování potřebného počtu sestupných hran pro řízení krokového motoru, který se následně otočí o stejný počet kroků. Pokud dojde ke spuštění generování pulsů pro motor, je hlavní spouštěcí signál blokován funkcí logického součinu pomocí zpětné vazby s blokem paměti a negace (Nonretriggerable), viz obr. č. 7.12. Další spuštění je možné až po vygenerování právě spuštěného sledu pulsů.

Ovládací panel pro aplikaci ControlDesk

Pro ovládání a vizualizaci HiL testování řídicí jednotky automatické klimatizace byl rovněž vytvořen příslušný panel v aplikaci ControlDesk, který je znázorněn na následujícím obr. č. 7.14.

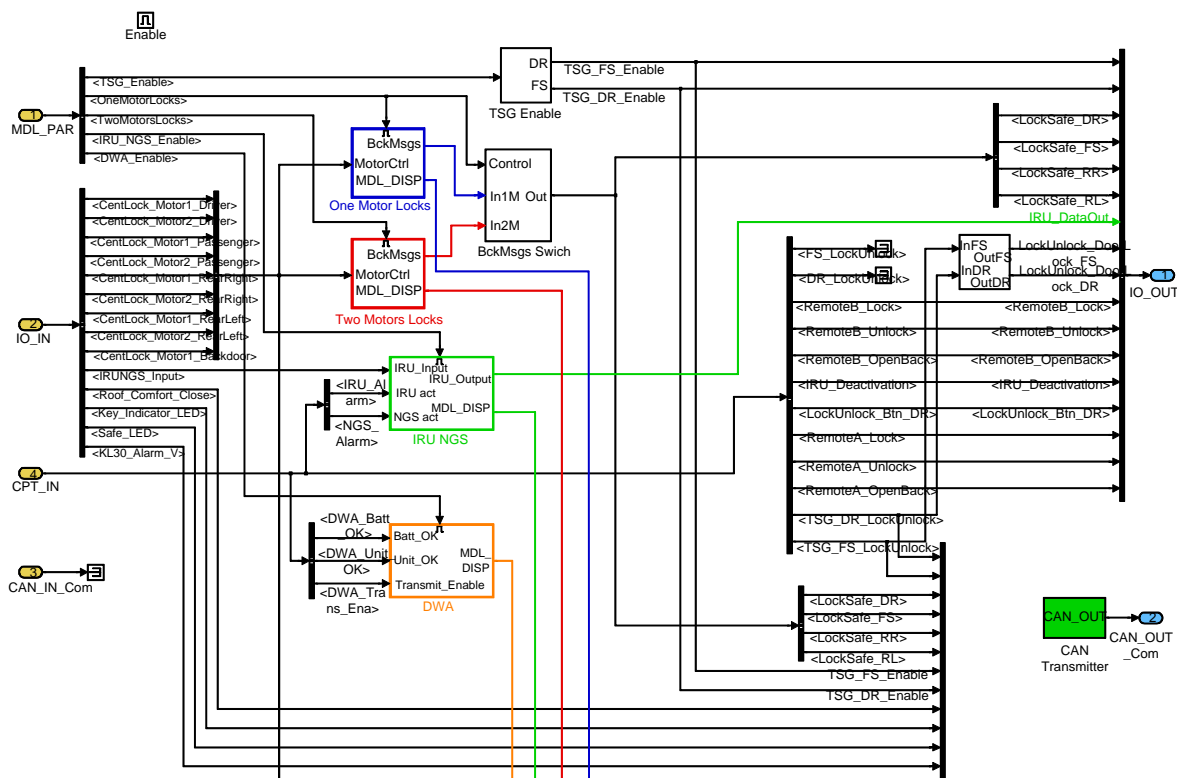


Obr. č. 7.14 Grafické uspořádání ovládacího a vizualizačního panelu řídicí jednotky automatické klimatizace v aplikaci ControlDesk

7.2.2 Model okolí řídicí jednotky centrálního zamykání

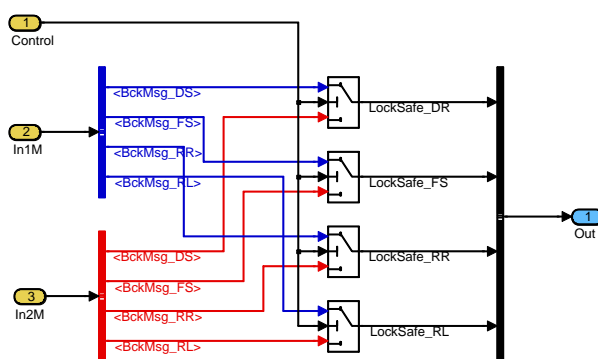
Řídicí jednotka centrálního zamykání je konstruována tak, že umí ovládat dva typy zámků, a to typ s jedním motorem, resp. typ se dvěma motory. Volba typu zámků se provádí tzv. kódováním řídicí jednotky prostřednictvím diagnostického přístroje. V modelu jsou proto rovněž zahrnuty oba typy zámků. Aktivní typ lze zvolit za běhu modelu v reálném čase příslušnými signály v závislosti na nakódování řídicí jednotky

Model pro řídicí jednotku centrálního zamykání s alarmem je znázorněn na následujícím obr. č. 7.15. Model zahrnuje dveřní zámků, jak s jedním motorem (One Motor Locks), tak se dvěma motory (Two Motors Locks), snímače sledování vnitřního prostoru a náklonu vozidla (IRU NGS) a inteligentní sirénu (blok DWA).



Obr. č. 7.15 Model okolí pro řídicí jednotku centrálního zamykání – dveřní zámky, snímače sledování vnitřního prostoru a náklonu, inteligentní siréna^{*)}

Výstupní signály z modelů zámků jsou přepínány přepínačem zpětného hlášení stavu zámků (BckMsgs Switch), znázorněného na obr. č. 7.16.



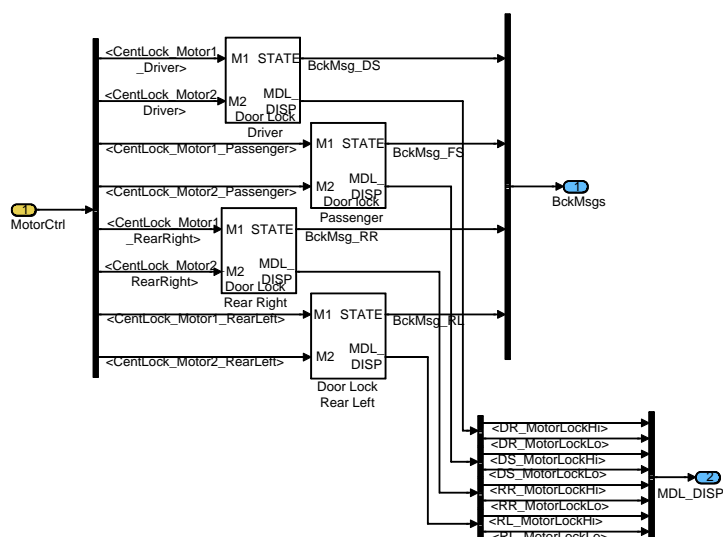
Obr. č. 7.16 Struktura bloku přepínače zpětného hlášení stavu zámků

Blok modelu dveřních zámků s jedním motorem

Blok dveřních zámků s jedním motorem (One Motor Locks), jehož struktura je znázorněna na obr. č. 7.17, obsahuje modely zámků všech dveří. Tedy dveří řídiče

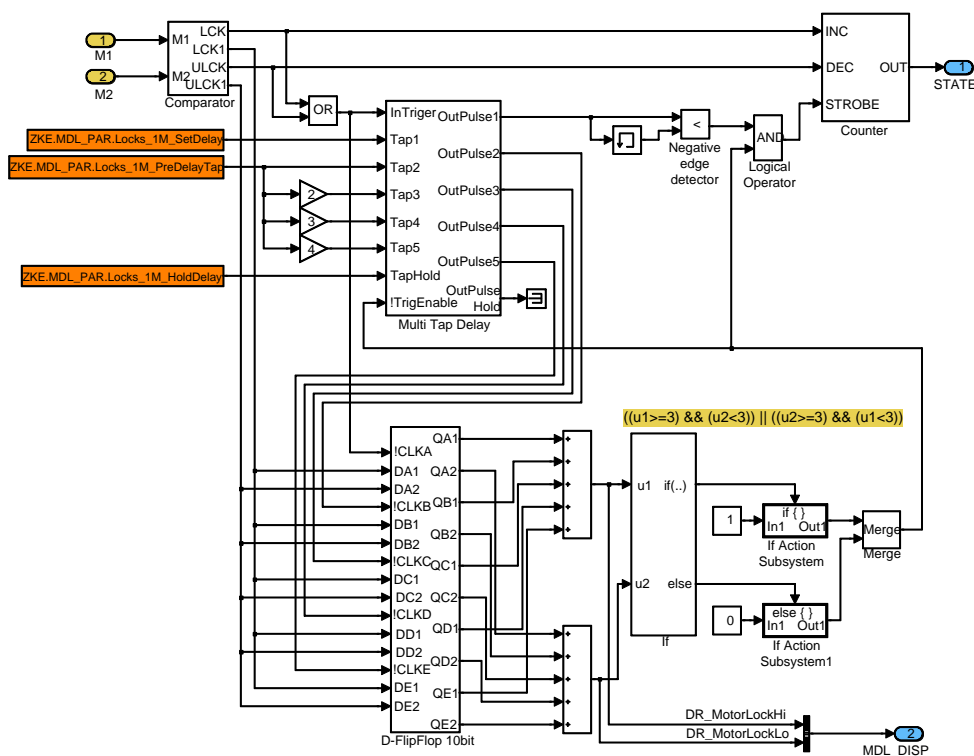
^{*)} Na obr. č. 7.15 jsou ve spodní části oříznuty některé signály, které jsou následně zobrazovány v aplikaci ControlDesk (sběrnice signálů MDL_DISP). Oříznutí bylo provedeno pro úsporu místa, obrázek by jinak byl dvojnásobně vysoký. Pro seznámení se strukturou modelu to nemá za následek ztrátu informace.

(Door Lock Driver), spolujezdce (Door Lock Passenger), zadních pravých (Door Lock Rear Right) a zadních levých dveří (Door Lock Rear Left).



Obr. č. 7.17 Struktura bloku zámku s jedním motorem

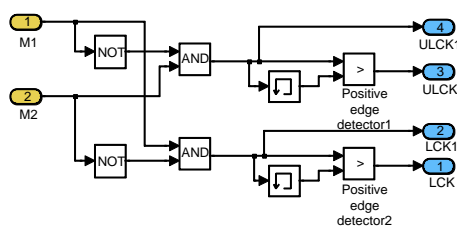
Vnitřní struktury všech bloků dveřních zámků jsou stejné. Tato struktura je znázorněna na obr. č. 7.18.



Obr. č. 7.18 Struktura bloku dveřního zámku s jedním motorem

Jak již bylo zmíněno v kapitole 5.3.2, dveřní zámky mají tři stavy – odemčeno, zamčeno a zajištěno. Zámek s jedním motorem těmito stavy prochází postupně, v pořadí od odemčeno k zajištěno nebo opačně, díky sledu pulsů generovaných řídicí jednotkou

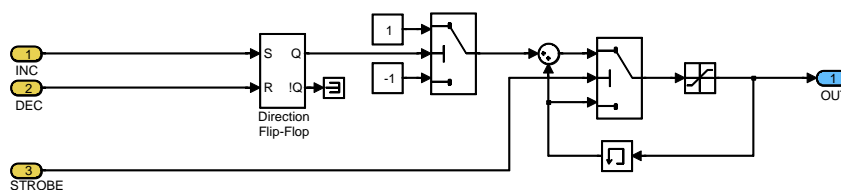
na vodičích M1 a M2 napájejících motor zámku. Signály těchto vodičů jsou zavedeny jako vstupy do modelu a pulsy jsou vyhodnocovány blokem komparátoru (Comparator, viz obr. č. 7.19) pro určení zda dochází postupně k zajištění nebo odemčení zámku.



Obr. č. 7.19 Struktura bloku komparátoru

Součástí bloku komparátoru jsou rovněž detektory náběžných hran, jejichž výstupy jsou zavedeny do bloků čítače stavu (Counter) a vícenásobného zpožďovače (Multi Tap Delay), viz obr. č. 7.18.

Struktura bloku čítače stavu zámku je znázorněna na dalším obr. č. 7.20. Podle směru zamykání resp. odemykání se stav čítače zvyšuje nebo snižuje pomocí signálu STROBE. Výstupní hodnota odpovídá příslušnému stavu a je omezena blokem saturace. Hodnota stavu je pak následně převedena na hodnotu odporu jako zpětné hlášení o stavu zámku řídicí jednotce.



Obr. č. 7.20 Struktura bloku čítače stavu zámku

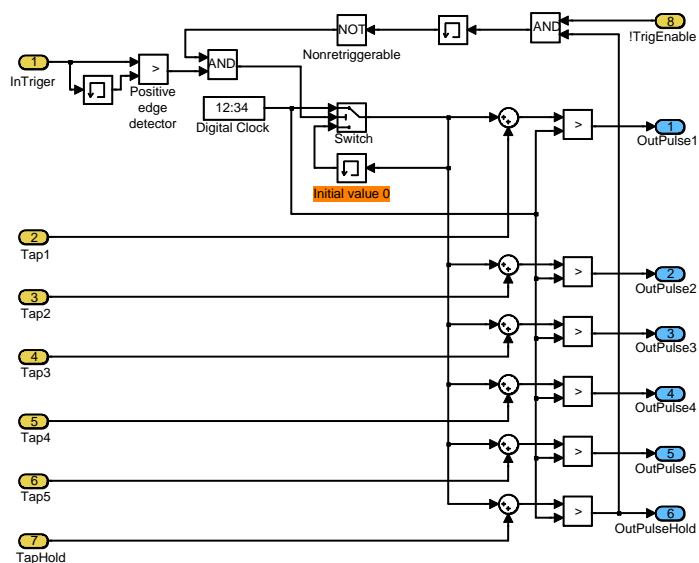
Vzhledem k tomu, že se na vodičích motoru M1 a M2 vyskytují náhodné zákmity, které způsobovaly nežádoucí změnu stavu modelu zámku při detekování dané hrany, bylo nutné tyto zákmity eliminovat. Byla k tomu použita metoda postupného vzorkování vstupního signálu po náběžné hraně a následného vyhodnocení, zda signál byl po příslušnou dobu v požadované úrovni. Postupně se získává pět vzorků od každého signálu, které se ukládají do 10bitového klopného obvodu typu D reagujícího na sestupnou hranu (D-FlipFlop 10bit, viz obr. č. 7.18). Navzorkované hodnoty „1“ jsou pro jednotlivé signály sečteny a součty porovnány podmínkovým blokem (If) tak, že jeden ze signálů by měl mít více než 3 vzorky v hodnotě „1“ a zároveň druhý méně než 3 vzorky rovněž v „1“.

Podmínkový výraz pro vyhodnocení vzorků signálů M1 a M2 je formulován následovně:

$$(7.13) \quad ((N_{M1} \geq 3) \wedge (N_{M2} < 3)) \vee ((N_{M2} \geq 3) \wedge (N_{M1} < 3))$$

Kde N_{M1} je počet vzorků s hodnotou „1“ signálu M1 a N_{M2} je počet vzorků s hodnotou „1“ signálu M2. Pokud podmínkový výraz není splněn, dojde k blokadě spouštěcího signálu čítače stavu a naopak k opětovnému odblokování vícenásobného zpoždovače.

Pro časování vzorkování vstupních signálů byl vytvořen blok vícenásobného zpoždovače (Multi Tap Delay, viz obr. č. 7.21). V principu se jedná o vícenásobný monostabilní klopný obvod s jednou časovou základnou, jedním společným spouštěcím vstupem reagujícím na náběžnou hranu a šesti výstupy s nezávisle nastavitelnými dobami kvazistabilních stavů.

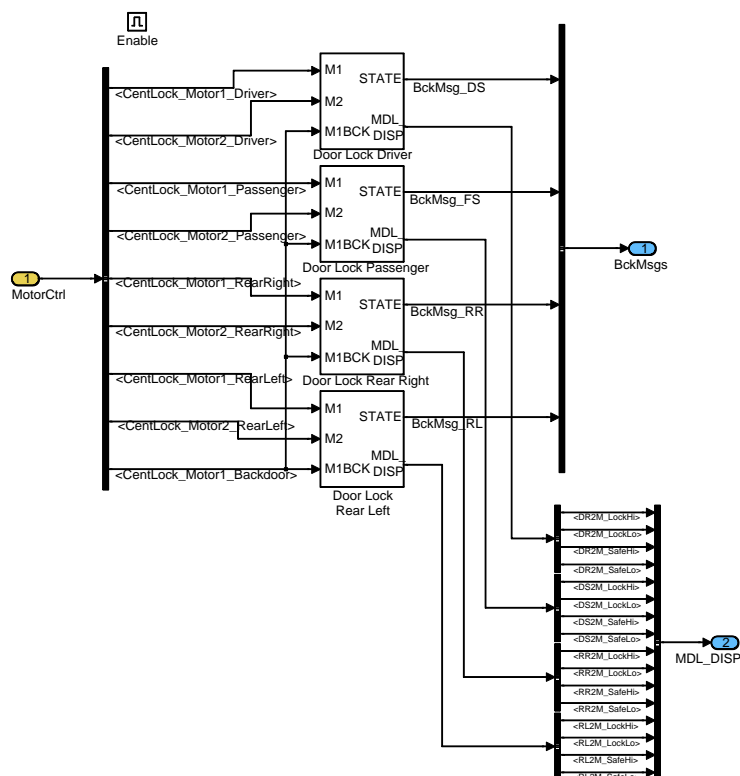


Obr. č. 7.21 Struktura bloku vícenásobného zpoždovače

Blok modelu dveřních zámků se dvěma motory

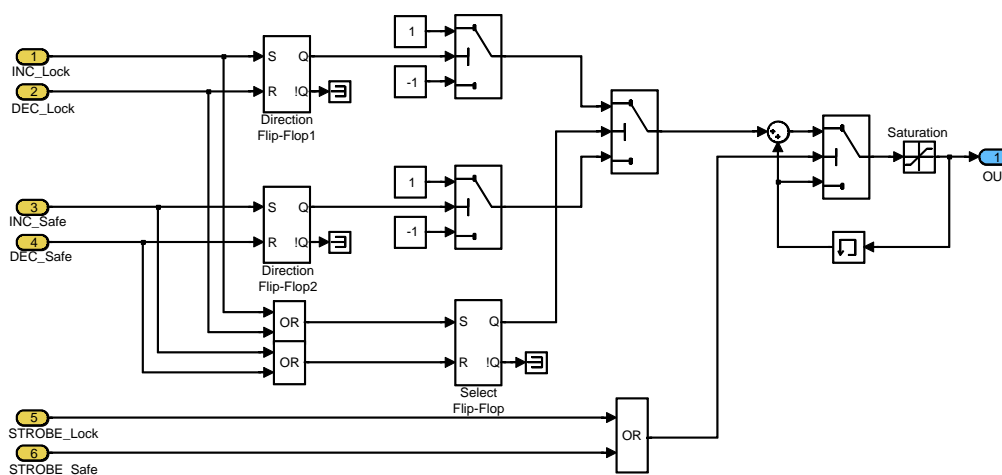
U zámku se dvěma motory jeden motor slouží k přechodu ze stavu odemčeno do stavu zamčeno a naopak a druhý motor pak k přechodu mezi stavy zamčeno a zajištěno.

Struktura bloku zámků se dvěma motory (Two Motors Locks, viz obr. č. 7.22) je velmi podobná struktuře bloku zámků s jedním motorem. Hlavní rozdíl je v tom, že dva motory jsou s řídicí jednotkou propojeny pomocí tří vodičů M1, M2 a M1BCK. M1BCK je společný vodič z ECU pro všechny zámky.

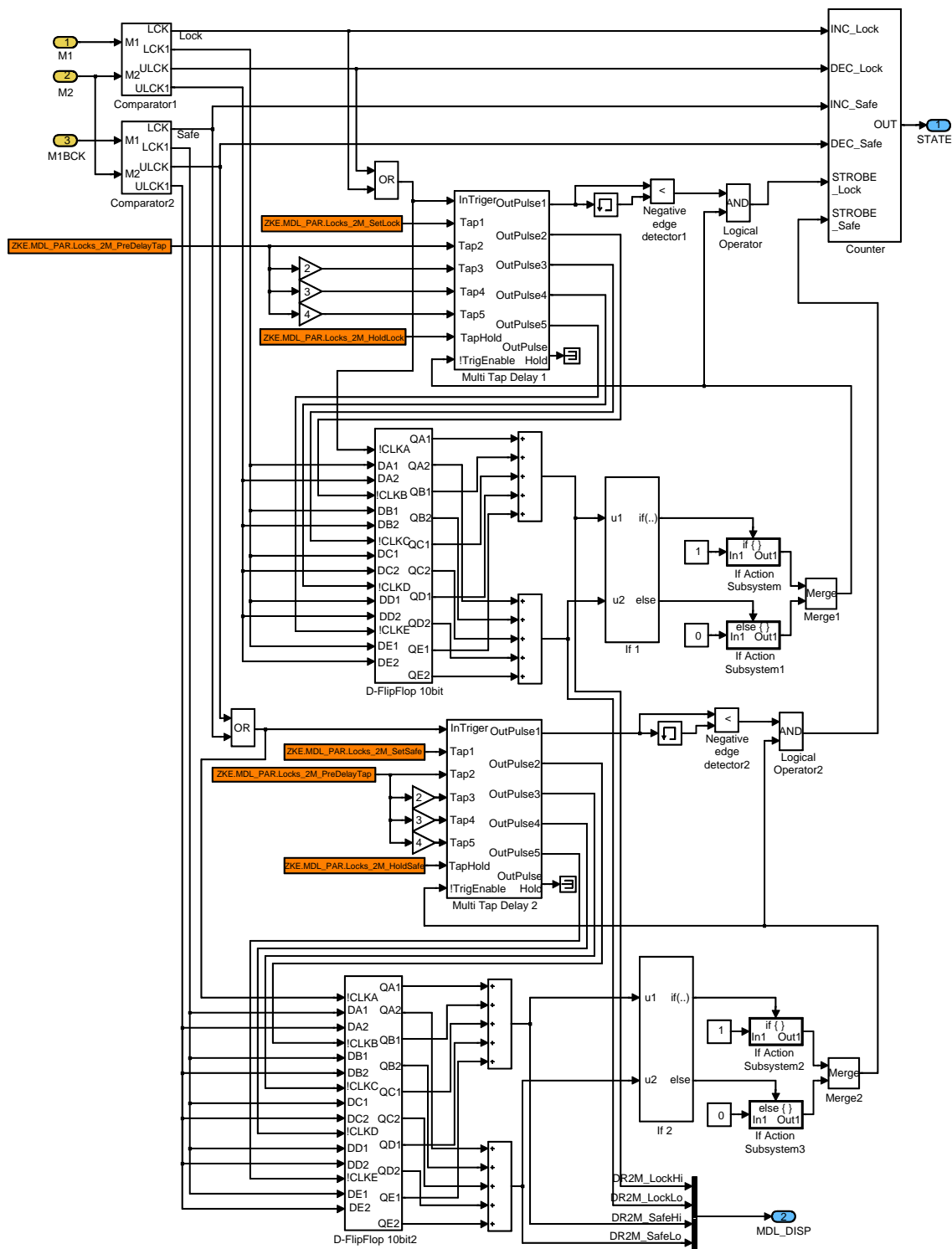


Obr. č. 7.22 Struktura bloku zámků se dvěma motory

Struktura bloku dveřního zámku je uvedena dále na obr. č. 7.24. Princip modelu byl již vysvětlen, zde jsou použity stejné bloky, ale zdvojeně, poněvadž řídicí jednotka ovládá dva motory. Jiný je akorát čítač stavu (Counter), který je uveden na obr. č. 7.23 a modeluje změnu stavu na základě signálů pro oba motory.



Obr. č. 7.23 Struktura bloku modifikovaného čítače stavu zámku se dvěma motory

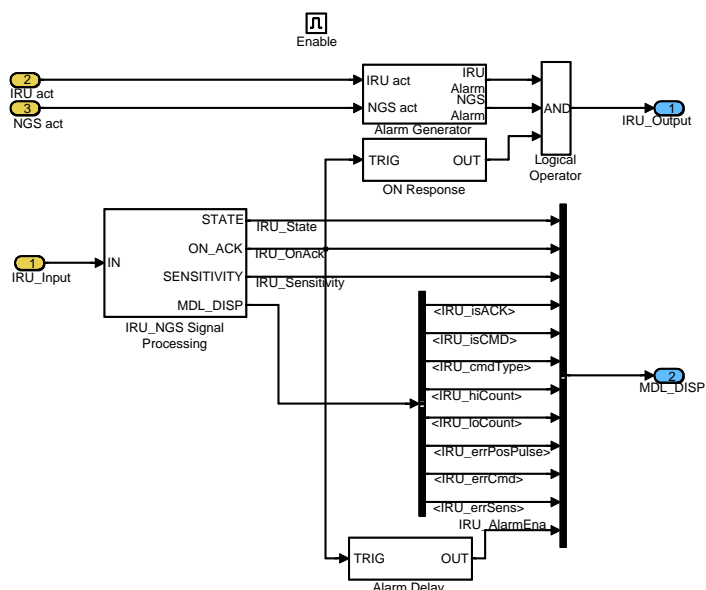


Obr. č. 7.24 Struktura bloku dveřního zámku se dvěma motory

Blok modelu snímače sledování vnitřního prostoru a snímače náklonu vozidla

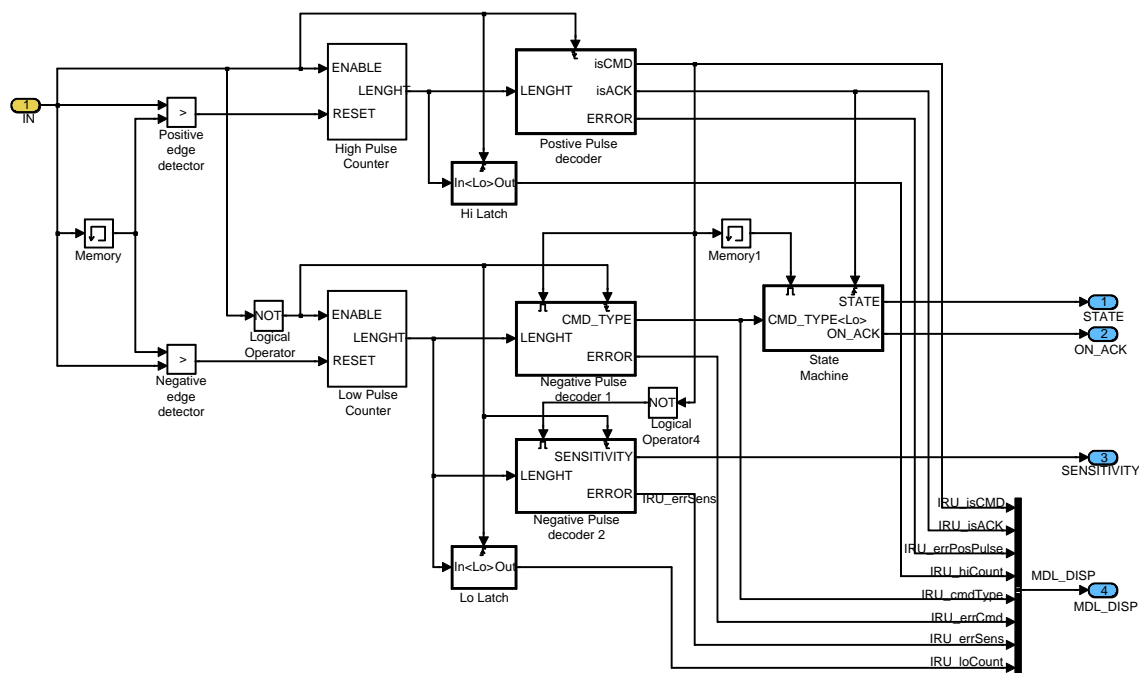
Snímače sledování vnitřního prostoru a náklonu vozidla jsou umístěny v jednom modulu, se kterým řídicí jednotka komunikuje prostřednictvím proprietárně definovaných pulsních signálů. Pulsy jsou relativně dlouhé, takže jsou vzorkovány přímo modelem v reálném čase a vzorkovací perioda tedy odpovídá časovému kroku

modelu. Struktura bloku snímačů (IRU NGS) je znázorněna na následujícím obr. č. 7.25. Blok snímačů zahrnuje zpracování signálu od řídicí jednotky (IRU_NGS Signal Processing), generátor alarmu (Alarm Generator) a další bloky pro časové zpoždění.



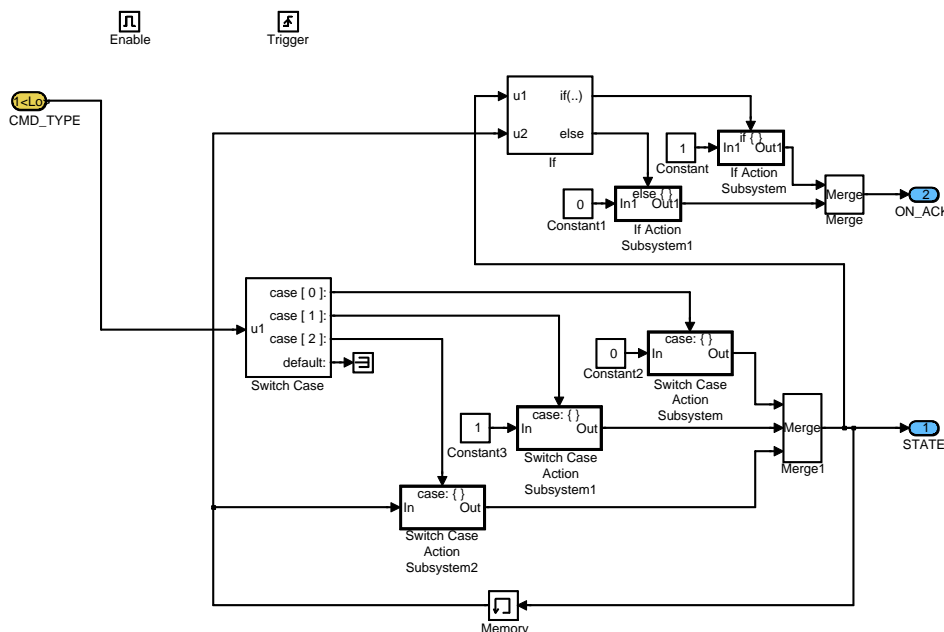
Obr. č. 7.25 Struktura bloku snímačů sledování vnitřního prostoru a náklonu

V bloku zpracování signálu (viz obr. č. 7.26) se provádí detekce náběžných a sestupných hran a následně měření délky pulsu (High Pulse Counter, Low Pulse Counter). Z délky pulsu se pak v dekodech určuje typ příkazu, který dále ovlivňuje přechody mezi stavy stavového automatu snímačů (State Machine).



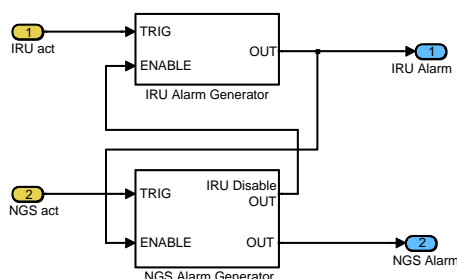
Obr. č. 7.26 Struktura bloku zpracování signálu snímačů

Stavový automat je dále znázorněn na obr. č. 7.27 a je realizován multipodmínkovým blokem (Switch Case) a paměťovým blokem (Memory). Kromě toho je podmínkovým blokem (If) detekován přechod pro vyvolání alarmu na výstupu ON_ACK.

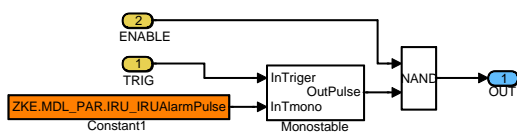


Obr. č. 7.27 Stavový automat snímačů

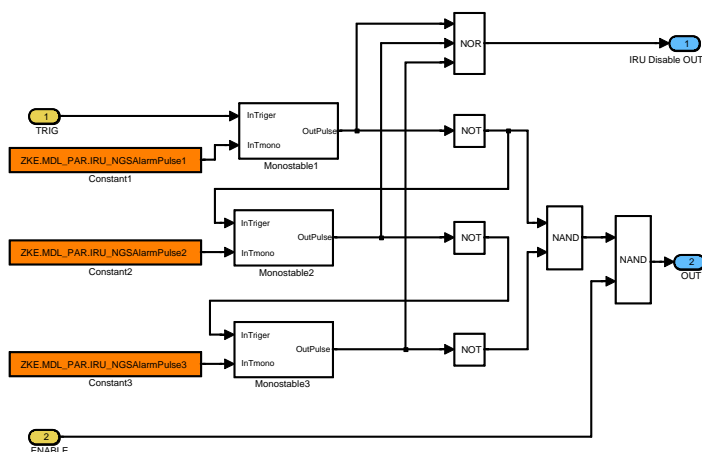
Struktura generátoru alarmu je znázorněna na obr. č. 7.28. Obsahuje dva bloky podle snímače, který je zdrojem alarmu. Vnitřek bloku generátoru alarmu od čidla sledování vnitřního prostoru (IRU Alarm Generator) je znázorněn na obr. č. 7.29 a blok generátoru alarmu od čidla náklonu (NGS Alarm Generator) je pak rozkreslen na obr. č. 7.30. Výstupem bloku generátoru alarmu je sled pulsů nesoucí informaci o stavu snímačů pro řídicí jednotku.



Obr. č. 7.28 Generátor alarmu



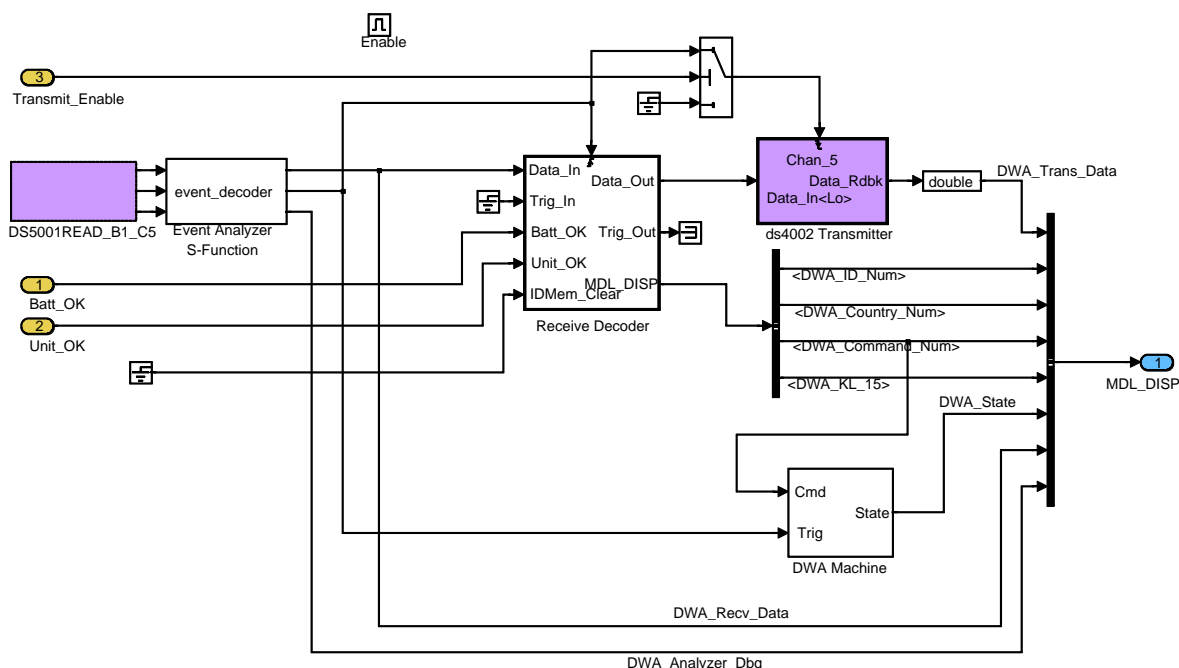
Obr. č. 7.29 Generátor alarmu od čidla sledování vnitřního prostoru



Obr. č. 7.30 Generátor alarmu čidla náklonu

Blok inteligentní sirény DWA

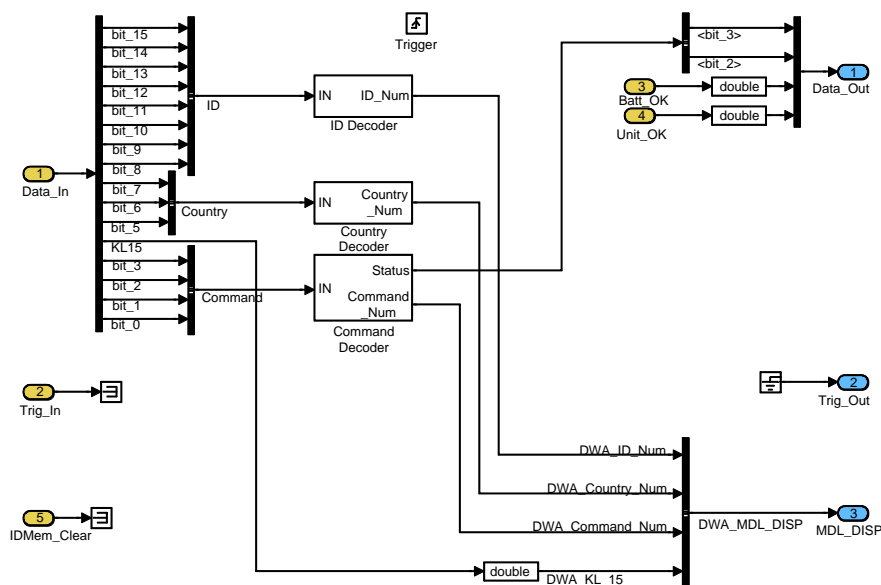
Inteligentní siréna má vlastní řídicí mikropočítač a baterii pro zálohování napájení. S řídicí jednotkou siréna komunikuje prostřednictvím proprietární sériové komunikace, která vychází z komunikace pomocí periferních obvodů USART. Komunikační rychlost je však vyšší, takže signál nelze vzorkovat a vyhodnocovat modelem v reálném čase. Proto byly využity možnosti karet pro zpracování rychlých číslicových signálů DS5001 pro vstup signálu z řídicí jednotky a DS4002 pro výstup zpět do ECU. Struktura bloku pro simulaci inteligentní sirény je na obr. č. 7.31.



Obr. č. 7.31 Struktura bloku inteligentní sirény DWA

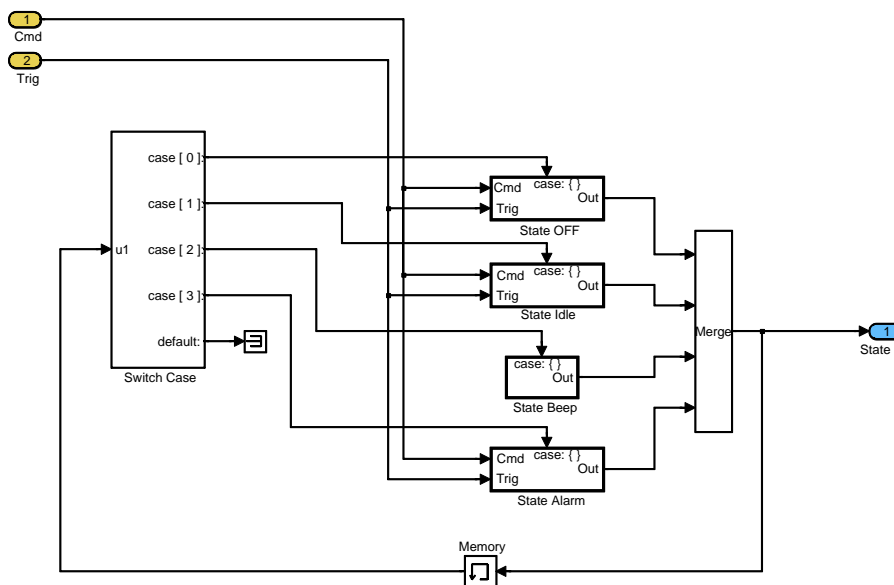
Vstupní signál je kartou vzorkován a ukládají se časové informace o náběžných a sestupných hranách (událostech, resp. změnách ve vstupním signálu). Události jsou

modelem z karty přečteny a dále zpracovány pomocí s-funkce (v bloku Event Analyzer S-Function). Výstupem z této funkce je 16bitová informace vstupující do bloku dekodéru přijatých dat (Receive Decoder, viz obr. č. 7.32) k dalšímu zpracování. Po zpracování je odpověď řídicí jednotce vysílána pomocí bloku ds4200 Transmitter, což je opět s-funkce provádějící konfiguraci sekvenceru pro výstup arbitrárního pulsního signálu. Funkci sekvenceru zajišťuje pomocný signálový procesor umístěný na kartě DS4002.



Obr. č. 7.32 Struktura bloku dekodéru přijatých dat

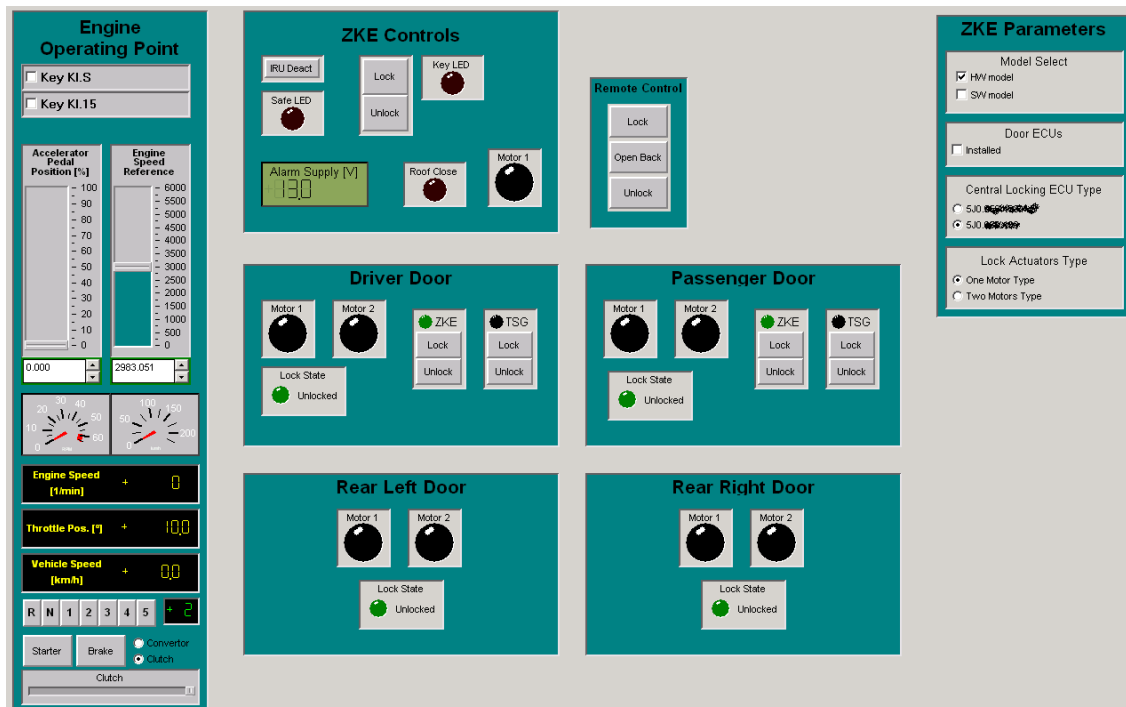
Pro vizualizaci stavu sirény byl vytvořen pomocný blok stavového automatu sirény (DWA Machine), který je použit v obr. č. 7.31 a jehož vnitřní struktura je znázorněna na následujícím obr. č. 7.33.



Obr. č. 7.33 Struktura bloku stavového automatu sirény

Ovládací panel pro aplikaci ControlDesk

Také pro ovládání a vizualizaci modelu okolí pro řídicí jednotku centrálního zamykání byl vytvořen ovládací a vizualizační panel v aplikaci ControlDesk a je znázorněn na následujícím obr. č. 7.34.



Obr. č. 7.34 Grafické uspořádání ovládacího a vizualizačního panelu řídicí jednotky centrálního zamykání v aplikaci ControlDesk

8. Závěr

Testování v různých podobách je dnes nedílnou součástí vývojového procesu. V případě vývojových procesů s pokročilým modelem, jako jsou V-model nebo agilní model, je kladen požadavek na opakovatelnost provedení testování (testovacích sekvencí). Což jednoznačně směřuje k automatizovanému provádění, k čemuž jsou nezbytné příslušné technické prostředky a programové vybavení.

Předmětem této práce však byla zejména implementace HiL testování pro funkční testování vybraných řídicích jednotek. Byla vybrána a navržena konfigurace HiL testovací platformy. Dále byly vytvořeny modely, které emulují okolí pro řídicí jednotky automatické klimatizace a centrálního zamykání a uzavírají tak jejich regulační smyčky. HiL testovací platforma s příslušnými modely jsou základním předpokladem pro funkční provoz testované řídicí jednotky v laboratorním prostředí. V práci je tedy uveden popis testovaných řídicích jednotek (kapitola 5.3), charakteristika některých testovacích nástrojů včetně konfigurace vybraného systému (kapitoly 6.2, 6.3 a 6.5) a nemalá pozornost je věnována vlastnímu popisu realizovaných modelů v kapitole 7.2. Modely byly uplatněny při testování ve ŠKODA AUTO a.s., viz reference [106] a [108] v příloze C.

Takto navržená testovací platforma spolu s programovými nástroji a implementovanými modely je připravena k provozu testování. Lze provádět testy ručně, podle daného testovacího předpisu, „klikáním“ na ovládacích a vizualizačních panelech v aplikaci ControDesk. K automatizaci testování je nutné nasadit další programové nástroje, z nichž některé byly zmíněny v kapitole 6.4.

9. Zdroje a literatura

- [1] Bushnell, Michael L. – Agrawal, Vishwani D. *Essentials of electronic testing for digital, memory, and mixed-signal VLSI circuits*. 2nd print. Boston (USA): Kluwer Academic Publishers, 2001. ISBN 0-7923-7991-8.
- [2] Abramovici, Miron – Breuer, Melvin A. – Friedman, Arthur D. *Digital systems testing and testable design*. Revised printing. New York (USA): IEEE Press, c1990. ISBN 0-7803-1062-4.
- [3] Hlavička, Jan – Kotteck, Eduard – Zelený, Jaroslav. *Diagnostika elektronických číslicových obvodů*. Vyd. 1. Praha: SNTL, 1982. MDT 681.32:681.518.54
- [4] Marks, David M. *Testing very big systems*. 1st edition. New York (USA): McGraw-Hill, c1992. ISBN 0-07-040433-X.
- [5] McConnell, Steve. *Dokonalý kód: Umění programování a techniky tvorby software*. Přeložil Bogdan Kiszka. Vyd. 1. Brno: Computer Press, 2005. ISBN 80-251-0849-X.
- [6] Liversidge, Ed. *The Death of the V-Model*. [online] Harmonic Software Systems, 2005. [cit. 2011-03-11]. <http://www.harmonicss.co.uk/index.php/hss-downloads/doc_download/12-death-of-the-v-model>
- [7] Neuschl, Štefan – Blatný, Jan – Šafařík, Jiří – Zendulka, Jaroslav. *Modelovanie a simulácia*. 1. vydání. Bratislava: Alfa, 1988. 063-559-88.
- [8] Dušek, František. *Matlab a Simulink – úvod do používání*. 1. vydání. Pardubice: Univerzita Pardubice, 2000. ISBN 80-7194-273-1.
- [9] *Web home page*. [online] MathWorks, 2011. [cit. 2011-03-11]. <<http://www.mathworks.com>>
- [10] *MATLAB - The Language Of Technical Computing: Product web page*. [online] MathWorks, 2011. [cit. 2011-03-11]. <<http://www.mathworks.com/products/matlab/>>
- [11] *Simulink - Simulation and Model-Based Design: Product web page*. [online] MathWorks, 2011. [cit. 2011-03-11]. <<http://www.mathworks.com/products/simulink/>>

-
- [12] *Stateflow - Design and simulate state machines and control logic: Product web page*. [online] MathWorks, 2011. [cit. 2011-03-11].
<<http://www.mathworks.com/products/stateflow/>>
- [13] Modelica Association et al. *Modelica® - A Unified Object-Oriented Language for Physical Systems: Modeling Language Specification Version 3.2*. [online] Version 3.2. Linköping (Sweden): Modelica Association, 2011. [cit.: 2011-03-11].
<<https://www.modelica.org/documents/ModelicaSpec32.pdf>>
- [14] *Welcome to OpenModelica: Web home page*. [online] OpenModelica, Last updated 2010-12-20. [cit.: 2011-03-11]. <<http://www.openmodelica.org/>>
- [15] *Home – Scilab Web site*. [online] Digiteo, 2011. [cit.: 2011-03-11].
<<http://www.scilab.org/>>
- [16] Chancelier, J.P. *ScicosLab: Web home page*. [online] Last updated 2011-01-06. [cit.: 2011-03-11]. <<http://www.scicoslab.org/>>
- [17] Campbell, Stephen L. – Chancelier, Jean-Philippe – Nikoukhah, Ramine. *Modeling and simulation in scilab/scicos with scicoslab 4.4*. 2nd ed. New York (USA): Springer, 2009. ISBN 978-1-4419-5526-5.
- [18] *Test & Measurement World: Web home page*. [online] UBM Electronics, 2011. [cit.: 2011-03-11]. <<http://www.tmworld.com>>
- [19] *LIN subbus – News & Events: Web home page*. [online] LIN Administration, 2008. [cit.: 2011-03-11]. <<http://www.lin-subbus.org>>
- [20] Kubík, Michal. Zařízení pro testování řídicích jednotek na sběrnici LIN. In *Elektrotechnika a informatika 2004. Část druhá, Elektronika*. Plzeň: Západočeská univerzita, 2004. s. 33-36. ISBN 80-7043-299-3.
- [21] Kubík, Michal. HIL testování a simulace. In *Elektrotechnika a informatika 2005. Část druhá, Elektronika*. Plzeň: Západočeská univerzita, 2005. s. 53-55. ISBN 80-7043-374-4.
- [22] Kubík, M. – Vít, M. Hardware in the loop testing and simulation. In *Applied electronics 2005: International conference: Pilsen*. Pilsen: University of West Bohemia, 2005. s. 197-200. ISBN 80-7043-369-8.
- [23] Vacek, Václav. *Sériová komunikace ve WIN 32*. 1. vyd. Praha: BEN – technická literatura, 2003. ISBN 80-7300-086-5.

- [24] Vít, Martin. *Automatizované testování elektronických řídicích jednotek automobilu*. [disertační práce] Plzeň: Západočeská univerzita vPlzni. Fakulta elektrotechnická. Katedra aplikované elektroniky a telekomunikací, 2007. 88 s., 23 s. příl. Školitel prof. Ing. Jiří Pinker, CSc.
- [25] Racek, Stanislav. *Objektově orientované programování v C++*. 1. vyd. České Budějovice: Nakladatelství KOPP, 1994. ISBN 80-85828-20-0.
- [26] *ADwin products: product web page*. [online] Keithley Instruments, c2005. [cit.: 2011-03-11]. <http://www.adwinproducts.com/ADwin_Products.html>
- [27] *FlexRay - The communication system for advanced automotive control applications: Web home page*. [online] Altran, 2009. [cit.: 2011-03-11]. <<http://www.flexray.com/index.php>>
- [28] Kadlecová, Vladimíra – Voženílek, David – Mikuš, Dalibor. Systém předního osvětlení automobilů AFS společnosti Visteon. *Světlo* [online]. 2003, č 01. ISSN 1212-0812. [cit. 2011-03-11]. <http://www.odbornecasopisy.cz/index.php?id_document=23035>.
- [29] ŠKODA AUTO a.s. *Škoda Auto Muzeum*. Třída Václava Klementa 294, Mladá Boleslav, tel.: 326 831 134, e-mail: museum@skoda-auto.cz
- [30] *Application Note Series: Simulation & Test System for Electronic Control Units (ECUs)*. [online] No. 2433. Cleveland (Ohio, USA): Keithley Instruments, 2003. [cit.: 2011-03-11]. <http://www.adwinproducts.com/PDFs/AD_AppsNote_SimulationTestSystem_ECUs.pdf>
- [31] *TenAys INtime RTOS - When Just Windows Isn't Enough!: Product web page*. [online] TenAys, 2011. [cit.: 2011-03-11]. <<http://www.tenasys.com/products/intime.php>>
- [32] *PROVEtech tool suite – Ready-to-use standard tools: Product web page*. [online] MBtech Group, 2011. [cit.: 2011-03-11]. <http://www.mbtech-group.com/eu-en/electronics_solutions/tools_equipment/provetech_tool_suite.html>
- [33] Roberto Bucher – Simone Mannori – Thomas Netter. *RTAI-Lab tutorial: Scilab, Comedi, and real-time control*. [online] 3rd edition. Milano (Italy): Politecnico di Milano - Dipartimento di Ingegneria Aerospaziale, 2008. [cit.: 11.03.2011]. <<https://www.rtai.org/RTAILAB/RTAI-Lab-tutorial.pdf>>

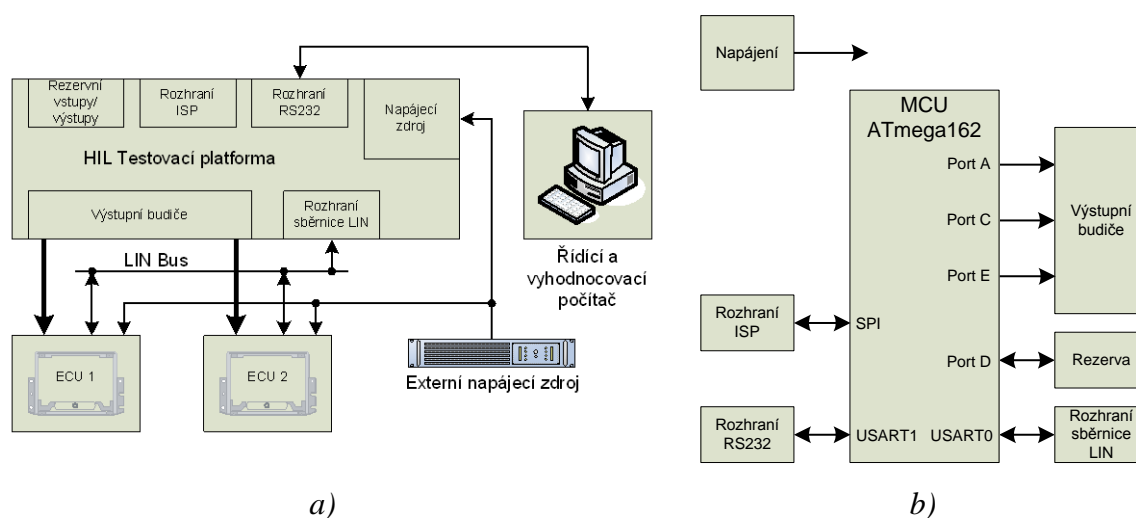
- [34] *RTAI - Official Website*. [online] Politecnico di Milano. Dipartimento di Ingegneria Aerospaziale, 2010. [cit.: 11.03.2011]. <<https://www.rtai.org/>>
- [35] Hess, Frank Mori – Abbott, Ian. *Comedi - Control and Measurement Interface: Web home page*. [online] 2009. [cit.: 11.03.2011]. <<http://www.comedi.org/>>
- [36] Veverka, Stanislav: *Simulace dynamického systému v prostředí Matlab-Simulink a PROVEtech:TA*. [diplomová práce] Plzeň: Západočeská univerzita vPlzni. Fakulta elektrotechnická. Katedra aplikované elektroniky a telekomunikací, 2010. 70 s. Vedoucí DP Ing. Michal Kubík.
- [37] Šrajfer, Michal. *Návrh a realizace konceptu zábavy v automobilech Škoda*. [diplomová práce] Plzeň: Západočeská univerzita vPlzni. Fakulta elektrotechnická. Katedra aplikované elektroniky a telekomunikací, 2008. 52 s., 2 s. příl. Vedoucí DP Ing. Michal Kubík.
- [38] *Die ADAC Pannenstatistik 2007*. [online] Stand: 04/08. [SRN]: ADAC, 2000. IN 25214. [cit.: 11.03.2011]. <http://www.adac.de/_mm/pdf/Pannenstat_2007_159KB_33406.pdf>
- [39] *Products - CTE XL and CTE XL Professional: Product page*. [online] Berner & Mattner Systemtechnik, 2011 [cit.: 11.03.2011]. <<http://www.berner-mattner.com/en/berner-mattner-home/products/cte/index-cte-ueberblick.html>>
- [40] Hayhurst, Kelly – Veerhusen, Dan – Chilenski, John – Rierson, Leanna. *A Practical Tutorial on Modified Condition/ Decision Coverage*. [online] Hampton (Virginia, USA): NASA, 2001. NASA/TM-2001-210876. [cit.: 11.03.2011]. <<http://shemesh.larc.nasa.gov/fm/papers/Hayhurst-2001-tm210876-MCDC.pdf>>
- [41] *Standard DO-178B / ED-12B: Software Considerations in Airborne Systems and Equipment Certification*. RTCA (Radio Technical Commission for Aeronautics) and EUROCAE (European Organization for Civil Aviation Equipment), 1992.

Dále byly jako zdroje informací použity příručky k programům Matlab, Simulink a ControlDesk a příručky k instalaci a programování modulárního hardware dSPACE.

Příloha A - Jednoduchý testovací systém

A.1 Blokové schéma struktura testovací platformy

Blokové schéma testovacího systému je znázorněno na obr. č. 9.1 a). Testovací systém zahrnuje testovací platformu, řídicí a vyhodnocovací počítač, na kterém běží ovládací aplikace a samostatný napájecí zdroj, který napájí, jak testovací platformu, tak i testované řídicí jednotky.



Obr. č. 9.1 Blokové schéma testovacího systému a) a testovací platformy b)

Testovací platforma je založena na jednočipovém mikropočítači ATmega162 z rodiny AVR firmy Atmel. Ten byl vybrán zejména z toho důvodu, že má integrovány dva sériové kanály (USART). Dále disponuje 35 digitálními vstupy/výstupy, sériovým rozhraním SPI, 16kB paměti programu FLASH, která může být programována v zapojení (ISP), 1kB paměti dat SRAM, dvěma 8b a dvěma 16b čítači/časovači s funkcí porovnávání a zachytávání (capture-compare).

Jak je vidět na blokovém schématu (obr. č. 9.1 b), jeden sériový kanál je využit pro komunikaci s počítačem, viz dále, a druhý jako sledovač komunikace na sběrnici LIN. Testované jednotky mají odporově kódované vstupy, kdy soustava tlačítek spíná rezistory s různým odporem ke společnému vodiči. Tlačítka jsou v testovací platformě nahrazeny tranzistory, které spolu s rezistory tvoří blok výstupních budičů a jsou programově ovládány z řídicího počítače. Součástí testovací platformy je i blok ISP rozhraní, které slouží pro nahrávání programu (tzv. firmware) do jednočipového mikropočítače během vývoje a blok rozhraní RS232 tvořený převodníkem úrovně TTL-RS232 založený na obvodu MAX323 firmy Maxim.

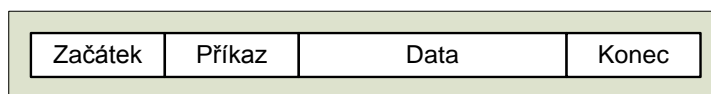
Principiálně testovací platforma provádí zachytávání zpráv ze sběrnice LIN a nastavuje dané kombinace na vstupy řídicích jednotek a všechny události na této úrovni opatřuje časovými značkami. Časované události se pak posílají zpátky do počítače k vyhodnocení.

Pro generování časových značek je využit jeden ze 16b čítačů, jenž ve spojení s další proměnnou vytváří 32b časové značky s rozlišením $1\mu s$.

A.2 Komunikace mezi testovací platformou a počítačem PC

Testovací platforma je propojena s řídicím počítačem PC prostřednictvím sériového rozhraní RS232. Komunikace probíhá v obou směrech postupně přenášením definovaných paketů (zpráv) v režimu „Master“ (počítač PC) – „Slave“ (testovací platforma), kdy testovací platforma reaguje na požadavky počítače, který jako „Master“ zahajuje komunikaci.

Formát paketu je blokově znázorněn na obr. č. 9.2. Paket má proměnnou délku v závislosti na množství přenášených dat a to je dáno typem příkazu. Pro vzájemnou komunikaci jsou definovány různé příkazy včetně formátu přenášených dat.



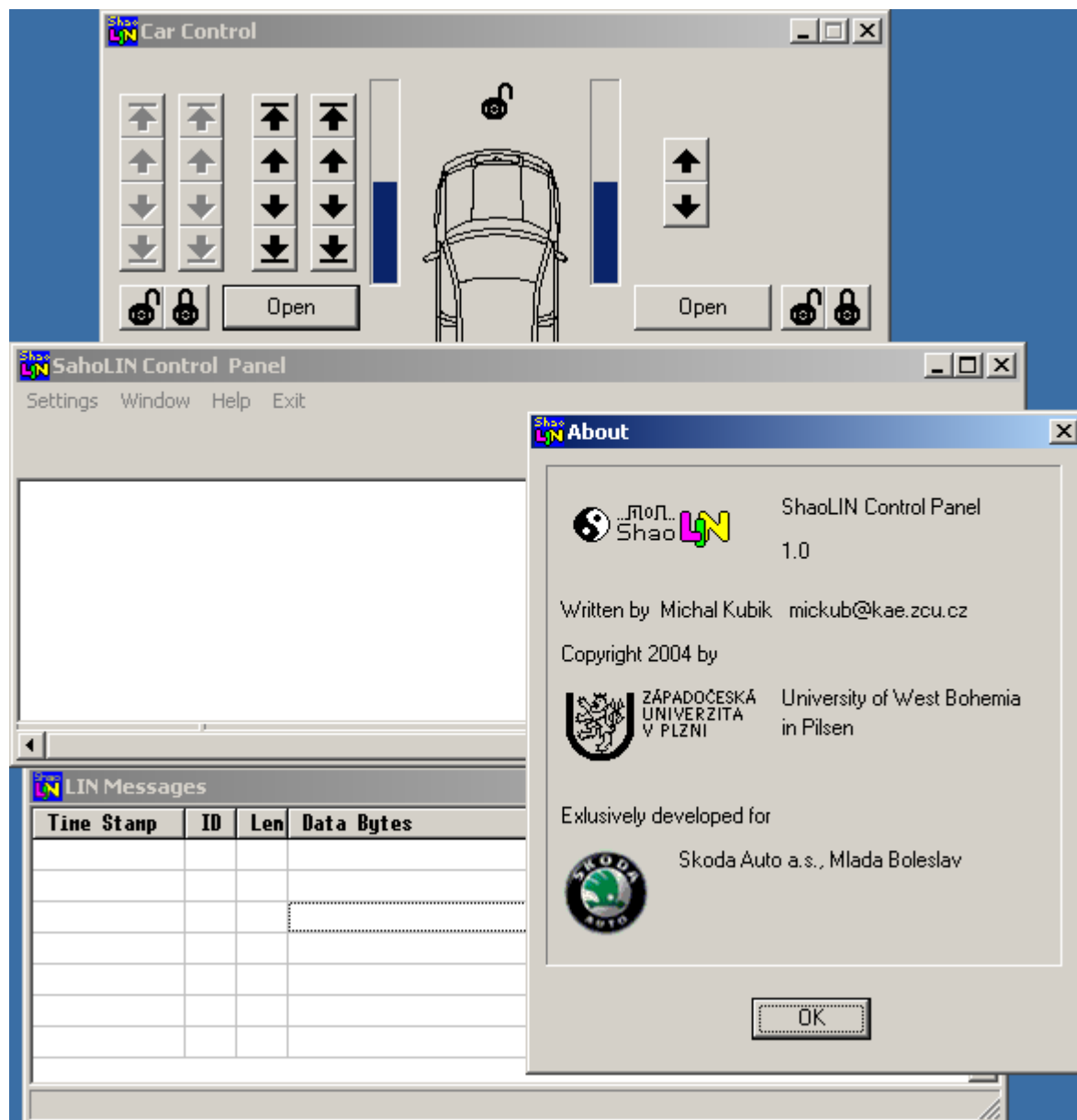
Obr. č. 9.2 Formát přenášeného datového paketu

Během komunikace jsou pro jednoduchost přenášeny jen ASCII znaky, což má své výhody i nevýhody. Mezi výhody lze zařadit možnost komunikace s testovací platformou pomocí jednoduchého terminálu, což bylo během vývoje využíváno před vytvořením ovládací aplikace. Jako nevýhodu lze označit redundantnost při přenosu dat, kdy je každý datový bajt přenášen pomocí dvou ASCII znaků, představující jeho hexadecimální hodnotu. Pro označení začátku paketu byl zvolen znak „Escape“ `'\27'` (ASCII hodnota 27, resp. 0x1B) a označení konce paketu znak odřádkování `'\n'`. Jak vyplývá z obr. č. 9.2 má např. paket s příkazem požadavku na odezvu následující tvar: `'\27' 'q' '\n'`.

Jak z popisu vyplývá nemá komunikace žádné zabezpečení, což by při reálném nasazení systému muselo být samozřejmě doplněno, např. pomocí CRC kódu. Podobně je tomu s ověřováním maximální doby odezvy – zatím se předpokládá okamžitá odezva.

A.3 Ovládací aplikace

Ovládací aplikace je standardní aplikace běžící pod systémem Windows. Byla vytvořena v prostředí C++ Builder v jazyce C++ s využitím objektových konstrukcí a knihovny vizuálních komponent - VCL (angl.: Visual Component Library).



Obr. č. 9.3 Okna ovládací aplikace

Aplikace je zkomponovaná z několika objektů (kompozice je jedním z nástrojů objektivě orientovaného programování). Každý objekt má specifické vlastnosti a zajišťuje určitou funkci. Hlavním objektem je MainForm třídy TMainForm, která je odvozená děděním z třídy TForm (dědičnost, též inheritance, je dalším nástrojem objektivě orientovaného programování, viz např. [25]). Objekt MainForm vytváří

hlavní okno aplikace, jehož hlavní částí je prostor pro zobrazování informací (formou textových výpisů) o testovací platformě a případných chyb komunikace na sběrnici LIN. Dále se zde nachází programová nabídka (menu) a další prvky. Objekt rovněž zajišťuje sběr a obsluhu událostí týkajících se vizuálních prvků hlavního okna a komunikaci s testovací platformou – vysílání příkazů a dekodování a separaci (angl. parsing) přijatých odezev a dat. Obsluhu sériového portu zajišťuje objekt `SerialPort` třídy `TThreadComm`, která je převzata z [23]. Tento objekt je dynamicky zakomponován do objektu `MainForm` (pomocí ukazatele a operátoru `new`).

Další zakomponované objekty tvoří dialogové okno pro nastavení komunikace s platformou (číslo portu a rychlost), okno s informacemi o programu, okno sloužící k ovládání testovací platformy umožňující nastavení kombinace pro stimulaci vstupů testovaných řídicích jednotek a okno zobrazující zachycené zprávy na sběrnici LIN. Vzhledem k tomu, že komunikace na sběrnici LIN probíhá periodicky, nejsou zprávy zobrazovány tak, jak jsou zachyceny (čili nekonečným výpisem), ale je zobrazen seznam zpráv, které se opakují, spolu s periodou opakování. Tedy statický výpis avšak aktuálního obsahu zpráv. Některá okna jsou pro ukázkou znázorněna na obr. č. 9.3.

Aplikace prozatím umožňuje nastavit požadovanou kombinaci stimulů na vstupy řídicích jednotek a dále prohlížení probíhající komunikace na sběrnici LIN a tím sledovat správnost odezvy řídicích jednotek z pohledu vzájemné komunikace.

Vzhledem k projevenému nezájmu nebyl vývoj testovacího systému dokončen ani do stádia funkčního prototypu, nicméně poznatky o testování získané během tohoto vývoje mají svůj přínos. Ukázalo se, že počáteční idea testování není nesprávná, ale vlastní implementace není tak triviální, jak se zprvu zdálo. Vývoj byl rovněž zastaven z důvodu požadavku univerzálnosti výsledného testovacího systému, což by tento systém nesplnil, protože i po dokončení by byl schopen testovat jen konkrétní typ řídicích jednotek. Vyvinout univerzální programovatelný testovací systém v konečném čase, zejména v dnešním konkurenčním a „preoptimalizovaném“ prostředí, je práce pro tým profesionálních vývojářů.

Příloha B - Přehled distributorů profesionálních testovacích systémů v ČR

Keithley

Oficiální zastoupení společnosti Keithley pro Českou Republiku zajišťuje Testovací Technika s.r.o., která se však produkty ADwin zabývá jen okrajově. Oficiální zastoupení pro produkty ADwin je pak v Německé Spolkové Republice.

TESTOVACÍ TECHNIKA s.r.o.

Hakenova 1423/III

290 01 Poděbrady

Tel.: (+420) 325 610 123 Fax: (+420) 325 610 134

E-mail: teste@teste.cz

Internet: <http://www.teste.cz>

Keithley Instruments GmbH/KIEX

Landsberger Strasse 65

82110 Germering

Tel.: +49-89-84 93 07-0 Fax: +49-89-84 93 07-87

E-mail: kiex_sales@keithley.com

Internet: <http://www.keithley.com>

ADwin produkty: <http://www.adwinproducts.com/index.html>

National Instruments

V České Republice má National Instruments svoji dceřinou společnost National Instruments (Czech Republic), s r.o.

National Instruments (Czech Republic), s r.o.

Dělnická 213/12

170 00 Praha 7

Tel.: (+420) 224 235 774 Fax: (+420) 224 235 749

E-mail: ni.czech@ni.com

Internet: <http://digital.ni.com/worldwide/czech.nsf/main?readform>

dSPACE

Společnost dSPACE GmbH je zastupována firmou Humusoft, která zároveň zastupuje společnost MathWorks a její produkty Matlab, Simulink, ad.

HUMUSOFT, spol. s r.o.

Pobřežní 224/20

186 00 Praha 8

Tel.: (+420) 284 011 730 Fax: (+420) 284 011 740

E-mail: info@humusoft.cz

Internet: <http://www.humusoft.cz/dspace/indexcz.htm>

MBtech Group

MBtech Group GmbH & Co. KGaA má v České Republice vlastní dceřinou společnost a pobočky v Praze, Plzni a Mladé Boleslavi.

MBtech Bohemia s.r.o.

Daimlerova 1161/6

301 00 Plzeň

Tel.: (+420) 233 029 300 Fax: (+420) 233 029 400

Internet: <http://www.mbtech-group.com/cz/spolecnost/domu.html>

Příloha C - Publikace autora a výstupy RIV

2003

[101] **Kubík, Michal**. Průzkum možností chromatické analýzy plochy. In *Elektrotechnika a informatika 2003*. Plzeň: Západočeská univerzita v Plzni, 2003. s. 66-68. ISBN: 80-7082-992-3.

2004

[102] **Kubík, Michal**. Zařízení pro testování řídicích jednotek na sběrnici LIN. In *Elektrotechnika a informatika 2004*. Plzeň: Západočeská univerzita v Plzni, 2004. s. 33-36. ISBN: 80-7043-299-3.

[103] **Kubík, Michal; Vít, Martin**. *Dveřní jednotky na sběrnici LIN : stav 28.6.2004, verze 1.0, soubor Dvere na LINu 10.doc*. Plzeň: Západočeská univerzita v Plzni, 2004. 20s.

2005

[104] **Kubík, Michal**. HIL testování a simulace. In *Elektrotechnika a informatika 2005. Část 2., Elektronika*. Plzeň: Západočeská univerzita v Plzni, 2005. s. 1-3. ISBN: 80-7043-374-4.

[105] **Kubík, Michal; Vít, Martin**. Hardware in the loop testing and simulation. In *Applied electronics 2005*. Pilsen: University of West Bohemia, 2005. s. 197-200. ISBN: 80-7043-369-8.

[106] **Kubík, Michal; Vít, Martin**. Model okolí pro řídicí jednotku Climatronic SK25. Mladá Boleslav: ŠKODA AUTO a.s. 2005. <M>

[107] **Weissar, Petr; Kosturik, Kamil; Kubík, Michal**. Modern microcontroller building set for teaching and development of industrial applications. In *Proceedings of the 4th WSEAS international conference on Applications of electrical engineering*. Prague: WSEAS, 2005. s. 43-47. ISBN: 960-8457-13-0.

2006

[108] **Kubík, Michal**. Model okolí pro řídicí jednotku centrálního zamykání SK25. Mladá Boleslav: ŠKODA AUTO a.s. 2006. <M>

2007

[109] **Kubík, Michal**. Control Methods of Individual Mobile Units in an Exactly defined Environment. In *IWCIT 2007*. Ostrava: VSB - Technical University, 2007. s. 15-18. ISBN: 978-80-248-1567-1.

[110] **Kubík, Michal; Housar, Pavel.** Model okolí pro řídicí jednotku dveří SK46. Mladá Boleslav: ŠKODA AUTO a.s., 2007. <M>

[111] **Kubík, Michal; Svoboda, Jan.** Model okolí pro řídicí jednotku paměťové sedačky SK46. Mladá Boleslav: ŠKODA AUTO a.s., 2007. <M>

2009

[112] **Kubík, Michal.** Testing scripts for automated testing of failure detection and DTC storage by ARS166 ECU. Plzeň: MBtech Bohemia s.r.o., 2009. <SW>

[113] **Kubík, Michal; Linha, Lukáš.** Modul pro ověření možností použití mikro počítačů AVR pro zpracování číslicových signálů. Plzeň: Západočeská univerzita v Plzni, 2009. <FV>

[114] **Kubík, Michal; Pivoňka, Lukáš.** CANtron - Vysílač na sběrnici CAN. Plzeň: Západočeská univerzita v Plzni, 2009. <FV>

[115] **Kubík, Michal; Pivoňka, Lukáš; Koucký, Václav.** Rozhraní pro programování mikro počítačů AVR v zapojení. Západočeská univerzita v Plzni, 2009. <FV>

2010

[116] **Kubík, Michal.** Set of additional functions and modules to the univesal testing library UxT. Plzeň: MBtech Bohemia s.r.o., 2010. <SW> (neuznáno bodové hodnocení v RIV – neveřejné zdroje)

[117] **Kubík, Michal; Chramosta, Michal.** Library module for generating of the CAN manipulation configuration files. Plzeň: MBtech Bohemia s.r.o., 2010. <SW> (neuznáno bodové hodnocení v RIV – neveřejné zdroje)

[118] **Kubík, Michal; Holeček, Jan.** Vývojová deska pro mikro počítače MSP430. Plzeň: Západočeská univerzita v Plzni, 2010. <FV>

[119] **Kubík, Michal; Paločko, Lukáš.** Modul invertoru DCC signálu. Plzeň: Západočeská univerzita v Plzni, 2010. <FV>

[120] **Kubík, Michal; Rajský, Jiří.** Program generující inicializační kód periférií mikro počítačů AVR. Plzeň: Západočeská univerzita v Plzni, 2010. <SW>

[121] **Kubík, Michal; Toušek, Jiří.** Modul výkonového zesilovače Baby of Lynx. Plzeň: Západočeská univerzita v Plzni, 2010. <FV>

[122] **Kubík, Michal; Veverka, Stanislav.** Model pro emulaci dynamického chování vozidla. Plzeň: Západočeská univerzita v Plzni, 2010. <SW>

Pozn.: Číslováno od hodnoty 101, aby nedošlo k záměně s odkazy na literaturu v textu. <FV> je G – Funkční vzorek; <SW> je R – Software; <M> je „Prototyp, metodika, SW“ (klasifikace RIV před rokem 2009).