

**ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA ELEKTROTECHNICKÁ**

**KATEDRA ELEKTROENERGETIKY A EKOLOGIE**

**DIPLOMOVÁ PRÁCE**  
**Tester ovládacích prvků v automobilu**

**vedoucí práce: Ing. Kamil Kosturik, Ph.D.**  
**autor: Bc. Daniel Kalaš**

**2016**

**ZÁPADOČESKÁ UNIVERZITA V PLZNI**

**Fakulta elektrotechnická**

**Akademický rok: 2015/2016**

## **ZADÁNÍ DIPLOMOVÉ PRÁCE**

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Daniel KALAŠ**  
Osobní číslo: **E13N0007K**  
Studijní program: **N2644 Aplikovaná elektrotechnika**  
Studijní obor: **Aplikovaná elektrotechnika**  
Název tématu: **Tester ovládacích prvků v automobilu**  
Zadávací katedra: **Katedra elektroenergetiky a ekologie**

### **Z á s a d y   p r o   v y p r a c o v á n í :**

1. Popište možné způsoby zkoušení ovládacích prvků v automobilu se zaměřením na snímání a vyhodnocování testovaných parametrů v požadovaném rozhraní a přesnosti.
2. Posuďte různé možnosti vyhodnocování s ohledem na provozní podmínky a vyhodnocení z pohledu testovacího pracovníka.
3. Navrhněte tester pro modul podvolantových ovládacích prvků.
4. Navržené řešení realizujte.

## **Abstrakt**

Předkládaná diplomová práce je zaměřena na řešení návrhu testeru pro ovládací prvky v automobilu, konkrétně pro ovládací prvky řídicího podvolantového modulu.

## **Klíčová slova**

Tester, sběrnice, signál, ovládací prvek, CAN, Flexray, LIN, CANoe, LabView, jednodeskový počítač.

## **Abstract**

This study is focused on tester design solutions for controls in car specifically for controls steering module.

## **Key words**

Tester, bus, signal, control, CAN, FlexRay, LIN, Canoe, LabView, single-board computer.

## **Prohlášení**

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této diplomové práce, je legální.

.....

podpis

V Plzni dne 16.5.2016

Daniel Kalaš

## **Poděkování**

Tímto bych rád poděkoval vedoucímu diplomové práce Ing. Kamilu Kosturikovi, Ph.D. za cenné profesionální rady, připomínky a metodické vedení práce.

## Obsah

<b>SEZNAM SYMBOLŮ A ZKRATEK .....</b>	<b>8</b>
<b>1 ÚVOD.....</b>	<b>9</b>
<b>2 ZÁKLADNÍ ROZDĚLENÍ OVLÁDACÍCH PRVKŮ .....</b>	<b>10</b>
2.1 MECHANICKÉ OVLÁDACÍ PRVKY .....	10
2.2 ELEKTRICKÉ ANALOGOVÉ OVLÁDACÍ PRVKY .....	10
2.3 ELEKTRICKÉ DIGITÁLNÍ OVLÁDACÍ PRVKY .....	10
<b>3 MOŽNOSTI TESTOVÁNÍ.....</b>	<b>11</b>
3.1 TESTOVÁNÍ ANALOGOVÝCH OVLÁDACÍCH PRVKŮ .....	11
3.2 TESTOVÁNÍ DIGITÁLNÍCH OVLÁDACÍCH PRVKŮ .....	12
3.3 TESTOVÁNÍ NEELEKTRICKÝCH VELIČIN .....	14
<b>4 MOŽNOSTI TESTOVÁNÍ A VYHODNOCENÍ S OHLEDEM NA PROVOZNÍ PODMÍNKY .....</b>	<b>15</b>
<b>5 MOŽNOSTI TESTOVÁNÍ A VYHODNOCENÍ S POHLEDU TESTOVACÍHO PRACOVNÍKA...17</b>	<b>17</b>
<b>6 POPIS TESTOVANÉHO VÝROBKU.....</b>	<b>18</b>
<b>7 NÁVRH A REALIZACE TESTOVACÍHO ZAŘÍZENÍ.....</b>	<b>19</b>
7.1 LED TESTER .....	19
7.1.1 Popis zapojení testeru .....	20
7.1.2 Blokové schéma zapojení .....	21
7.1.3 Program .....	22
7.2 TESTER S KOMUNIKACÍ PO SBĚRNICI.....	23
7.2.1 Popis zapojení testeru .....	24
7.2.2 Blokové schéma zapojení .....	25
7.2.3 Program .....	27
7.3 TESTER S VIZUALIZACÍ PRO ANALOGOVÉ OVLÁDACÍ PRVKY .....	29
7.3.1 Popis zapojení testeru .....	30
7.3.2 Blokové schéma zapojení .....	34
7.3.3 Program .....	35
<b>8 POROVNÁNÍ NAVRŽENÝCH TESTERŮ.....</b>	<b>37</b>
<b>9 ZÁVĚR .....</b>	<b>38</b>
<b>10 SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ .....</b>	<b>39</b>
<b>11 PŘÍLOHY.....</b>	<b>40</b>

## Seznam symbolů a zkratek

AVR.....	Mikročipy typu RISC s harvardskou architekturou
ARM.....	Architektura s nízkým příkonem a vysokou taktovací rychlostí
CRC.....	Cyclic redundancy check
NI.....	National Instruments
CAPL.....	CAN Access Programming Language, součást CANoe
HW.....	Hardware
SW.....	Software
LabVIEW.....	Laboratory Virtual Instrument Engineering, grafické vývojové prostředí
CANoe.....	Vývojový a testovací nástroj od firmy Vector Informatik GmbH



# 1 ÚVOD

Pro každou firmu je důležitá její pověst, její dobré jméno chcete-li, které je ve většině případů závislé na kvalitě poskytovaných služeb nebo výrobků. V případě automobilového průmyslu to platí dvojnásob a jde především o kvalitu a bezpečnost. Z tohoto důvodu vynakládají firmy spojené s automotive velký důraz na testování svých vlastních výrobků před další distribucí.

V případě testování výrobků však nejde pouze o kvalitu a bezpečnost, ale je tím také zajištěna zpětná vazba s produkcí. Tato vazba zaručuje okamžité řešení problémů ve výrobě, předcházení vyrábění vadných kusů a v nejhorším případě výskytu reklamací od zákazníka. Zároveň dle nastavených standardů a norem je testování vyžadováno. V jaké míře, to je dáno dle požadavků zákazníka či normy.

Pro testování výrobků existují automatické testy implementované přímo v lince, které jsou buď pro otestování částí funkcí nebo části výrobků v průběhu výroby, obvykle jsou však zavedeny na konci linky pro kompletní test výrobku. Tyto testy jsou první linií pro kontrolu funkčnosti a správného vyrobení výrobku. Pro namátkovou kontrolu, kterou provádí pracovníci kvality, se používají poloautomatické testy ovládané daným pracovníkem. Tyto testy se dále používají při třídění vadně vyrobených kusů či kvůli kontrole nastavené od zákazníka nebo při analýzách reklamovaných výrobků. Jak je vidět, testování hraje velice důležitou roli ve všech aspektech výroby.

V této práci se zaměřuji na tři různé verze testerů pro různé použití. Tester využívající vývojové desky a světelnou signalizaci, který je jednoduchý a finančně nenáročný. Dále tester využívající HW a SW CANoe od firmy Vector Informatik GmbH pro komunikaci po sběrnících, který dokáže snímat i analogové signály a tester pouze pro analýzy analogových ovládacích prvků s testovacím programem vytvořeným v prostředí LabVIEW. Návrh a realizace byla provedena u všech uvedených testerů, jelikož jsou využívány ve výrobě firmy Kostal spol. s.r.o. .

## 2 Základní rozdělení ovládacích prvků

### 2.1 Mechanické ovládací prvky

Řízení je prováděno pomocí mechanických struktur. Přímý přenos síly od řídicího prvku k řízenému prvku. Síla může být transformována pomocí posilovačů. Tato metoda je stále využívána pro pedály, převodovku či samotné řízení kol. Využití mechanických ovládacích prvků pro blinkry, stěrače, či jiné tlačítkové funkce je téměř nereálná a v této době nemá smysl.

### 2.2 Elektrické analogové ovládací prvky

Řízení je realizováno pomocí elektrických signálů, které jsou přenášeny od řídicího prvku k řízenému akčnímu členu pomocí vodičů. Každý vodič může přenášet pouze jeden signál. Pro sepnutí či vypnutí se využívá klasických spínačů, vypínačů či prepínačů navrhnutých a zkonstruovaných dle požadavků závisících na umístění daných prvků, snadné montáže, spolehlivosti aj. Systém může pracovat na více analogových principech, jako jsou uzemnění signálu nebo naopak přivedení signálu nebo na napěťových děličích díky kterým jsme schopni využít jeden vodič pro více funkcí. Využívá se pro podvolantové páčky, řadící páku, plynový pedál, různá ovládací tlačítka.

### 2.3 Elektrické digitální ovládací prvky

V tomto případě se využívá pro přenos řídicích informací sběrnice, po které se posílají zakódované informace. Tento systém je nejmladší a umožnil snížení nároků na kabeláž v autě, jelikož lze po jedné sběrnici posílat nespočet řídicích signálů. S kódováním však roste náchylnost na rušivé vlivy a tím i potřeba kontroly správnosti přenášených dat. Dále instalace potřebné elektroniky a tím i cena. V současné době se využívá tato komunikace téměř všude, neboť řídicí elektronika se nachází v každém automobilu. Sběrnice přenáší data nejen mezi řídicími jednotkami, ale i pro podvolantové páčky, snímače, řadící páku, různé ovládací prvky atd..

## 3 MOŽNOSTI TESTOVÁNÍ

### 3.1 Testování analogových ovládacích prvků

Analogové ovládací prvky jsou založeny na klasických spínacích prvcích u kterých buď signál přivádíme, uzemňujeme nebo přepínáme. Při přepínání záleží na úrovni analogového signálu, jelikož přidáváme či ubíráme dělicí odpory, čímž měníme úbytek napětí. Dále se využívají systémy na bázi optického či magnetického snímání. Místo klasického galvanického sepnutí je stimulací prvku provedeno rozsvícení/zhasnutí optického členu nebo změna pozice magnetu a v závislosti na této akci snímačem vytvořen řídicí signál. Takovéto ovládací prvky se používají tam kde je třeba použití galvanického oddělení nebo kde není možné použití systému klasického spínání. Vyhodnocování probíhá v řídicí jednotce, která nadále stimuluje akční členy nebo je řízení členů direktivní daným signálem.

Z toho vyplývá že pro testování analogových ovládacích prvků je nutné snímat úroveň signálu. Pro použití např. jednoduchých tlačítek nám stačí kontrolovat přítomnost (či nepřítomnost) signálu. Tento signál musí ovšem splňovat jistá kritéria, která jsou definována konstrukcí či zákazníkem (automobilkou), což znamená že spolu s úrovní sledujeme i kvalitu signálu.

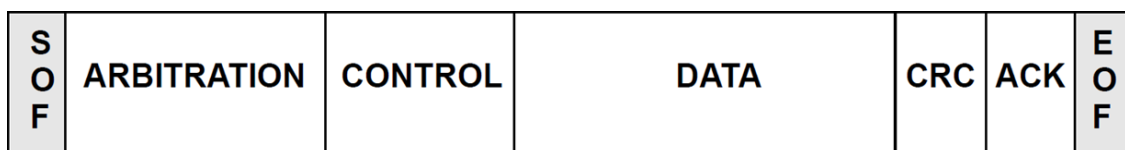
Aby byl signál kvalitní, je důležité aby již při konstrukci částí daného okruhu byli brány ohledy na možnost zarušení či poškození signálu. To může být ovlivněno mnoha faktory od použití různých materiálů s různými parametry, uložení kabeláže až po řešení mechaniky samotného spínacího prvku. Také elektrický návrh je velmi důležitý. Pokud budeme používat signály s nízkou úrovní, nemusí nám být schopny ovlivnit akční člen či řídicí jednotku. Pokud nebudou spojitě, mohou způsobovat rušení. Stejně tak je důležité i použití akčních členů takových, které jsou kompatibilní s naším navrženým signálovým řízením. Všechny tyto informace jsou pak potřebné pro vytvoření specifického, spolehlivého testovacího zařízení.

### 3.2 Testování digitálních ovládacích prvků

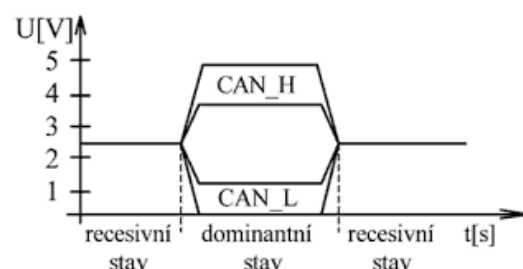
Tato terminologie není přesná, jelikož při samotném ovládní se používají stejné systémy jako u analogových ovládacích prvků. Hlavní rozdíl je ve zpracování a přenosu řídicích signálů. Jak už jsem uvedl dříve, přenos signálů je realizován pomocí sběrnice. Znamená to, že vlastní signály přivedené do řídicí jednotky jsou zpracovány a zakódovány dle protokolu pro danou sběrnici. Tyto zakódované informace jsou přenášeny k dalším řídicím jednotkám, které je zpětně dekodují a podle přijaté informace provedou řízení daných akčních členů. Výhody takovéto koncepce jsou nezměrné. V první řadě jde o snížení počtu vodičů. Sběrnice totiž využívá ve většině případů kroucenou dvoulinku, po které může přenést v podstatě všechny řídicí signály. Po sběrnici se též dají přenášet i jiné kódované signály jako obrazový či zvukový, takže přibyla možnost připojení různých elektronických zařízení.

Problém nastává u kvality signálu. Jelikož se nejedná o klasický signál o dané úrovni, ale o velice rychle se měnící signál, vznikají různá zarušení, šumy, odrazy aj. nežádoucí efekty. Tyto problémy musí být odstraněny nebo přinejmenším dostatečně redukovány. Nesmí se stát, že by se přenosem signál změnil a ve výsledku bylo provedeno jiné řízení. Z tohoto důvodu je kolem samotných dat přidána kvanta jiných informací, které zaručují správný přenos signálu ke správným jednotkám s danou kvalitou (Obr.1). Tato kvalita je závislá i na způsobu přenosu na fyzické vrstvě. Z tohoto důvodu se používá kroucená dvoulinka, diferenční signál a terminátory, jelikož všechny tyto opatření zvyšují odolnost proti rušení či deformaci signálů.

Obr.1 Datový rámeček CAN obecně



Obr.2 Úrovně diferenčního signálu protokolu CAN



## POPIS SBĚRNIC

- LIN** - Pro přenos signálu se využívá jeden vodič. Rychlost je nižší než u ostatních sběrnic a to max. 20Kbitů/s. Master-Slave komunikace. 64 ID zpráv – 2/4/8 bajtů ve zprávě. Použití pro jednoduché ovládání.
- CAN** - Protokol definovaný Robertem Boshem. Hojně využívaná v automobilech. Jedná se o přenos diferenčního signálu po kroucené dvojince což zvyšuje odolnost vůči rušení. Multimaster topologie. Arbitrážní systém – prioritní. Rychlost až 1Mbit/s , rozšířená verze až 6Mbit/s. Využití CRC.
- FLEXRAY** - Vytvořeno z důvodu potřeby předvídatelného řízení bez arbitrážního systému a kolizí (deterministické chování). Dva kanály A,B. Rychlost přenosu až 10Mbit/s. Vhodné pro řízení real-time.
- MOST** - Protokol vytvořený pro zařízení kladoucí nároky na přenos vyššího objemu dat nebo-li komfortní elektroniku, jako přehrávací zařízení, displeje aj..
- ETHERNET** - Nejrychlejší sběrnicový systém. Přenos až 100Mbit/s. Momentálně ve vývoji pro automobilový průmysl. Zatím se používá zřídka.

Testování takového ovládání je komplikovanější, jelikož musíme dekodovat dané signály. V případě řídicích jednotek většinou musíme simulovat i jiné řídicí jednotky, protože je testovaná elektronika vyžaduje. To přináší mnohé komplikace a peněžní výdaje.

K vytvoření testovacího zařízení je třeba transmitter a receiver pro buzení a sledování sběrnice, převodník pro přenos signálu do počítače, program na analýzu, samotná konfigurace pro testování dané jednotky. Vzhledem ke komplexnosti a složitosti signálu je zapotřebí i popis protokolu a vyhovujících parametrů.

V mnoha případech je třeba testovat i parametry, které se nemusí týkat přímo ovládacích prvků, ale týkají se řídicí jednotky, která od těchto prvků přijímá signály. Většinou jde o proudové zatížení nebo kontrolu průchozích kabelů (třeba pro airbag). V takovém případě je nutné použít nějaký snímač a převodník naměřené hodnoty. Tím nám vyvstává potřeba návrhu další měřicí elektroniky.

### 3.3 Testování neelektrických veličin

Při testování ovládacích prvků automobilu existují parametry, které neovlivňují řízení vozidla, ale pro kvalitu výrobku jsou také neméně důležité. Jedná se o hlučnost, pocitové vlastnosti (haptika) a vzhled. Každý výrobek musí splňovat jistá kritéria, která jsou buď daná firemní politikou nebo jsou vyžadována od zákazníků. Ve většině případů je to tak, že vnitřní požadavky jsou přísnější než požadavky zákazníka. Je to z důvodu zavedení takzvané záchytné meze.

Pokud bude mít výrobek jeden nebo více parametrů kolem této meze, lze stále poslat zákazníkovi. Pokud by toto nebylo zavedeno, mnoho výrobků by mohlo mít parametry na hranicích tolerance, což není dobře. Zároveň tato mez značí při opakujícím se výskytu nějaký problém na lince, což vede ke včasné úpravě či opravě a nevyrobení vadných kusů.

Hluk je testován nejčastěji pro podvolantové páčky, snímač natočení úhlu volantu, pružinové kazety která přenáší kroutící moment na statický (také volant) a různá tlačítka. Např. při otočení volantu nesmí být slyšet skřípání, pískání či drhnutí. Při stlačení tlačítka naopak musí být slyšet zvuk (pro ověření stlačení), ale s jistou hladinou hlasitosti.

Pocitové vlastnosti se testují jak jinak než pocitově. Pracovník zkouší vlastníma rukama, zda-li je použití ovládacího prvku pohodlné a bezproblémové. Zavádí se i automatické kontroly síly potřebné ke stimulaci ovládacích prvků. Každopádně tyto testy jsou schopny změřit pouze danou sílu.

Vzhled ovládacích prvků je kolikrát důležitější než jeho funkčnost. Samozřejmě že nefunkční prvky jsou k ničemu, ale pokud se zákazníkům nelíbí tak můžou fungovat sebelíp a neprodají se. Testování vzhledu spočívá v kontrole možných vad vzniklých při lisování umělohmotných krytů (defekty, jiné plochy - lesklá x hrubá, nehomogenní barva aj.), zapasování částí, poškrábání, nalomení, použití jiných částí aj. Tyto kontroly se provádí vždy okem.

Existují i jiné kontroly. Pro příklad testování podsvícení podvolantových pák. Jedná se o test, který je stimulovaný mechanickou pákou, řízený elektronikou a vyhodnocovaný kamerou, popř. okem.

## 4 MOŽNOSTI TESTOVÁNÍ A VYHODNOCENÍ S OHLEDEM NA PROVOZNÍ PODMÍNKY

Každý test může být ovlivněn prostředím ve kterém se provádí. Vždy záleží na druhu testu, jelikož různé testované parametry jsou závislé na různých podmínkách.

- TEPLOTA** - teplota přináší největší problémy ve změně mechanických parametrů materiálů. Použité plasty se rozpínají, smršťují, mění svůj tvar. Daná sestava pak na sebe nemusí správně doléhat, což dále způsobuje problémy s možným hlukem či haptikou. Při větší teplotě se ale zvyšuje i elektrický odpor vodičů a při změnách teplot se mění i vlastnosti použitých elektronických součástek.
- VLHKOST** - zvyšuje vodivost, může kondenzovat což přináší nebezpečí zkratů či změnu vodivosti. Pokud jsou nastavené ofsety při měření, změna vodivosti může značně změnit výsledky měření. V kombinaci s prachem tvoří nánosy.
- PRACH** - může ovlivnit kontaktování, zvyšuje přechodový odpor. Obzvláště v kombinaci s vlhkostí dokáže vytvořit nebezpečné nánosy, které velmi zkreslují jemné signály.
- HLUK** - tento faktor nám nejvíce ovlivňuje testování hlukových vlastností výrobku. Podvolantové moduly jsou mechatronické díly, které jsou při použití mechanicky stimulovány. Vznikají při tom hluky, u kterých je třeba hlídat jejich hladinu. Samozřejmě pokud test obsahuje nějaké zvukové signály, je dobré mít umístěný tester v místě kde není příliš ovlivněn hlukem z okolního prostředí.
- OTŘESY** - nejvíce ovlivňují kontaktování, pokud není konstruováno pevně.
- EMC** - ovlivnění prostředí elektromagnetickým zářením je časově proměnné, nelze se tedy zaměřit na jedinou úroveň a styl zarušení, ale je třeba

definovat jak moc je tester náchylný na tyto změny. Zjistit možný výskyt tohoto rušení v místě použití (a jeho nejhorší možnost) testeru a opatřit zařízení takovým stíněním aby okolní záření neovlivňovalo výsledky měření mimo stanovenou mez.

**PRACOVNÍK** - každý test je velice závislý i na provádějícím pracovníkovi. Je důležité mít proškoleného a zodpovědného pracovníka. V opačném případě přítomnost sebelepšího testu nemusí zaručit správné otestování výrobků. Z tohoto důvodu se testery vytváří tak, aby úloha pracovníka byla co nejvíce omezena a při případném zásahu aby existovala pouze jediná možnost jak lze krok provést.

**TESTOVACÍ ZAŘÍZENÍ** - co se týče samotného testovacího zařízení, mělo by být umístěno tak aby byla umožněná snadná manipulace jak s testem tak s testovanými výrobky. Mělo by být dobře upevněno a bezpečně elektricky odizolováno.



## 5 MOŽNOSTI TESTOVÁNÍ A VYHODNOCENÍ S POHLEDU TESTOVACÍHO PRACOVNÍKA

- VÝKON** - pokud je požadován po pracovníkovi výkon, který ve výsledku definuje příliš rychlé testování výrobku, můžou být přehlédnuty některé vady či faktory, které vadu mohou způsobit. Zároveň se zvyšuje riziko vytvoření vad jako jsou poškrábání, otlačení, či v případě manuálních “papírových” úkonů jejich opomenutí.
- PROSTŘEDKY** - při testování mohou být používány jiné testovací prostředky jako jsou třeba kalibry. Tyto kontrolní materiály by měli být vždy přesně nakalibrovány a nastaveny a měli by splňovat konstrukční parametry v nejvyšší kvalitě.
- BEZPEČNOST** - musí odpovídat stanoveným normám. Jedná se jak o bezpečnost úrazem elektrickým proudem, tak daným umístěním testeru, možnosti manipulace s výrobky (nepřecházet frekventovaná místa, nesundávat díly z výšky aj.), ostré hrany či rohy, kluzká podlaha atd.
- LOGISTIKA** - aby bylo zajištěno plynulé testování a tím i udržení požadovaného bezproblémového výkonu, je třeba dodávat výrobky k otestování nepřerušovaně. Zároveň zde vystupuje bezpečnost při zavážení a manipulaci s vozíkem a materiálem kolem testu.
- PODPORA** - pracovník obsluhující tester je vyškolen pro jeho používání k provádění správného průběhu a vyhodnocení testu. Pro případy, které nejsou v jeho kompetenci (porucha, opotřebenost částí, či výskytu jiné vyjimečné situace) musí být přítomna podpora, která je schopna tyto problémy vyřešit.
- ARBTR** - existují kontrolované parametry, které se vyhodnocují dle subjektivního hodnocení. Pokud není pracovník schopen vyhodnotit výsledek, je přítomen arbitr, který jasně určí zda je kontrolovaný znak ještě v pořádku či již není.
- ZNALOSTI** - každý pracovník je odborně proškolen pro provádění testu ke kterému byl přiřazen. Jelikož člověk není dokonalý tvor a přirozenou vlastností paměti je

zapomínání, umístí se k testeru kontrolní instrukce, která obsahuje všechny testovací kroky, jejich provedení, kontrolované znaky a styl jejich vyhodnocení.

**MOTIVACE** - motivace pracovníka určuje konečnou kvalitu a výkon testování. Líný nebo zhrzený pracovník nevydává potřebný výkon ani kvalitu. Z tohoto důvodu se snaží firmy pracovníky motivovat různými benefity.

**POKA-YOKE**- tento systém je založený na principu možnosti provedení operace pouze jediným možným stylem. Eliminují se tím možná poškození výrobků i testeru a testování probíhá rychleji. Pracovník se nemusí zamýšlet nad stylem provedení operace.

## 6 Popis testovaného výrobku

Tato práce je zaměřena na návrh testovacích zařízení pro ovládací prvky v automobilu. Konkrétně se jedná o podvolantový řídicí modul. Tento modul se skládá z několika základních částí. Jedná se o šasi do kterého jsou uloženy páky blinkrů, stěračů a popřípadě tempomatu a jiných funkcí. Na horní část šasi se pak nasazuje tzv. spirálová kazeta, která slouží k přenesení elektrických signálů přes točivý pohyb. Pro otestování elektrických funkcí celého modulu je tedy zapotřebí otestovat každou páku a také průchozí vodiče kazety. V případě, že má v sobě modul řídicí jednotku, probíhá většina komunikace pomocí sběrnice (CAN, LIN, Flexray aj.). U kazety můžeme měřit pouze zkrat či přerušeni a nebo snímat úrovně signálu či data, pokud jsou vyvedené.



Obr.3 Podvolantový řídicí modul pro AUDI

## 7 NÁVRH A REALIZACE TESTOVACÍHO ZAŘÍZENÍ

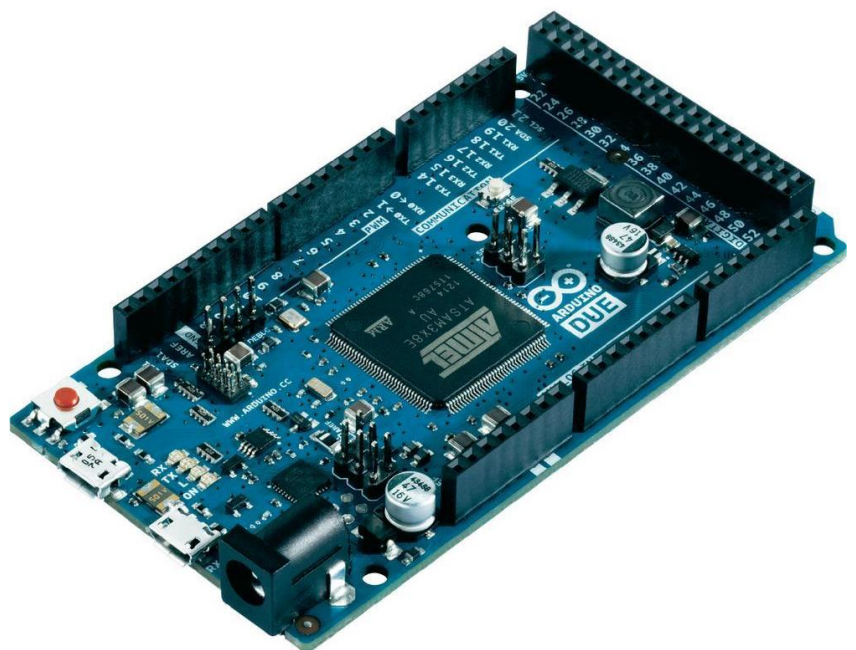
### 7.1 LED tester

V současné době jsou na trhu různá řešení takzvaných vývojových desek neboli jednodeskových počítačů. Nejznámější z představitelů je Raspberry Pi, který byl vyvinut v roce 2012 nadací Raspberry Pi Foundation. Tyto jednodeskové počítače byly navrženy pro podporu výuky studentů na školách, pro snadnější pochopení struktury a řízení počítačů. Na trhu je nepřehledné množství různých typů od různých výrobců. Pro můj tester jsem použil vývojovou desku Arduino DUE. První Arduino bylo představeno již v roce 2005 a od té doby firma vyvinula řadu dalších typů. Většina desek Arduina jsou stojí na mikrokontrolérech ATmega od firmy Atmel, které jsou postaveny na architektuře AVR. Arduino DUE má však osazen procesor SAM3X8E, jde o architekturu ARM, která je využívána v mobilních telefonech. Raspberry, která je srovnatelná se slabším PC, obsahuje připojení pro monitor, USB přes který jde připojit myš či klávesnice a je na ní možné spustit různé verze Linuxu nebo Windows 10 IoT Core. Na rozdíl od ní nemá Arduino připojení k monitoru, ke klávesnici ani myši. Je primárně připravena na připojení LED, servomotorků, relátek, LCD, senzorů atd. Její použití je spíše pro řízení, snímání či analýzu v technických řešeních. To je také důvod proč jsem si vybral tento typ desky.

Obr.4

Jednodeskový minipočítač

Arduino DUE



### 7.1.1 Popis zapojení testeru

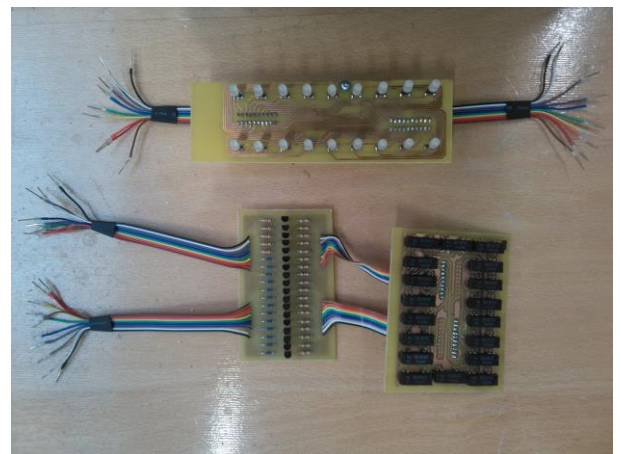
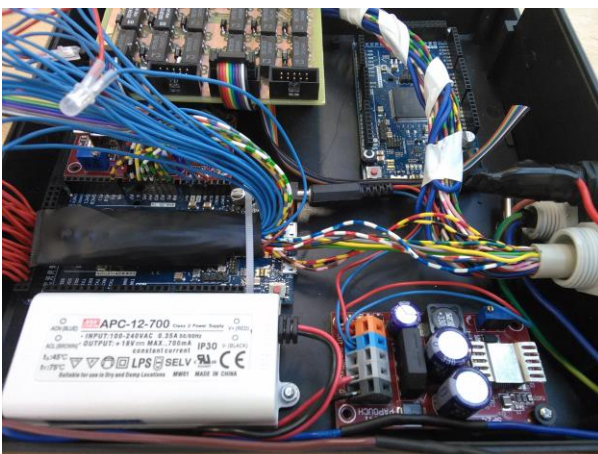
Testovací zařízení je rozděleno do dvou základních částí. Jedna je vytvořena pro testování průchozích vodičů ve spirálové kazetě, zatímco druhá se stará o testování ovládacích pák. Každou část ovládá jedno Arduino.

Pro testování kazety je vytvořena reléová deska, pomocí které přepínáme vodiče v kazetě k dělicímu a měřicímu odporu. Do okruhu je připojen stálý zdroj proudu. Tento okruh je oddělen od zbytku elektroniky, tvoří galvanicky spojený obvod pro snížení chyby v měření. Samotná reléová deska není spínána přímo z výstupů Arduina, ale přes desku buzení. Teprve tato deska je spínána přímo z Arduina, jelikož její výstupy nejsou tvořené pro takovou zátěž. Poslední deska obsahuje LED diody pro signalizaci sepnutých funkcí. Tato signalizační deska je zapojena pomocí invertorů přímo k výstupům Arduina. Přesnost měřených odporů vodičů není velká, ale to nevadí, jelikož přesné hodnoty se proměřují v jiné části výroby na speciálním testeru. Tento test slouží pro kontrolu průchodnosti či zkratu.

Pro testování pák není třeba tvořit samostatný obvod spínáním relé. Stačí nám pouze zdroj a dělicí odpor. Výstup z pák zapojíme přes dělicí odpory přímo na vstupy Arduina. Ta snímá hodnoty napětí na dělicím odporu a podle programu rozsvěcuje LED diody, které jsou zase napojeny přímo na výstupy.

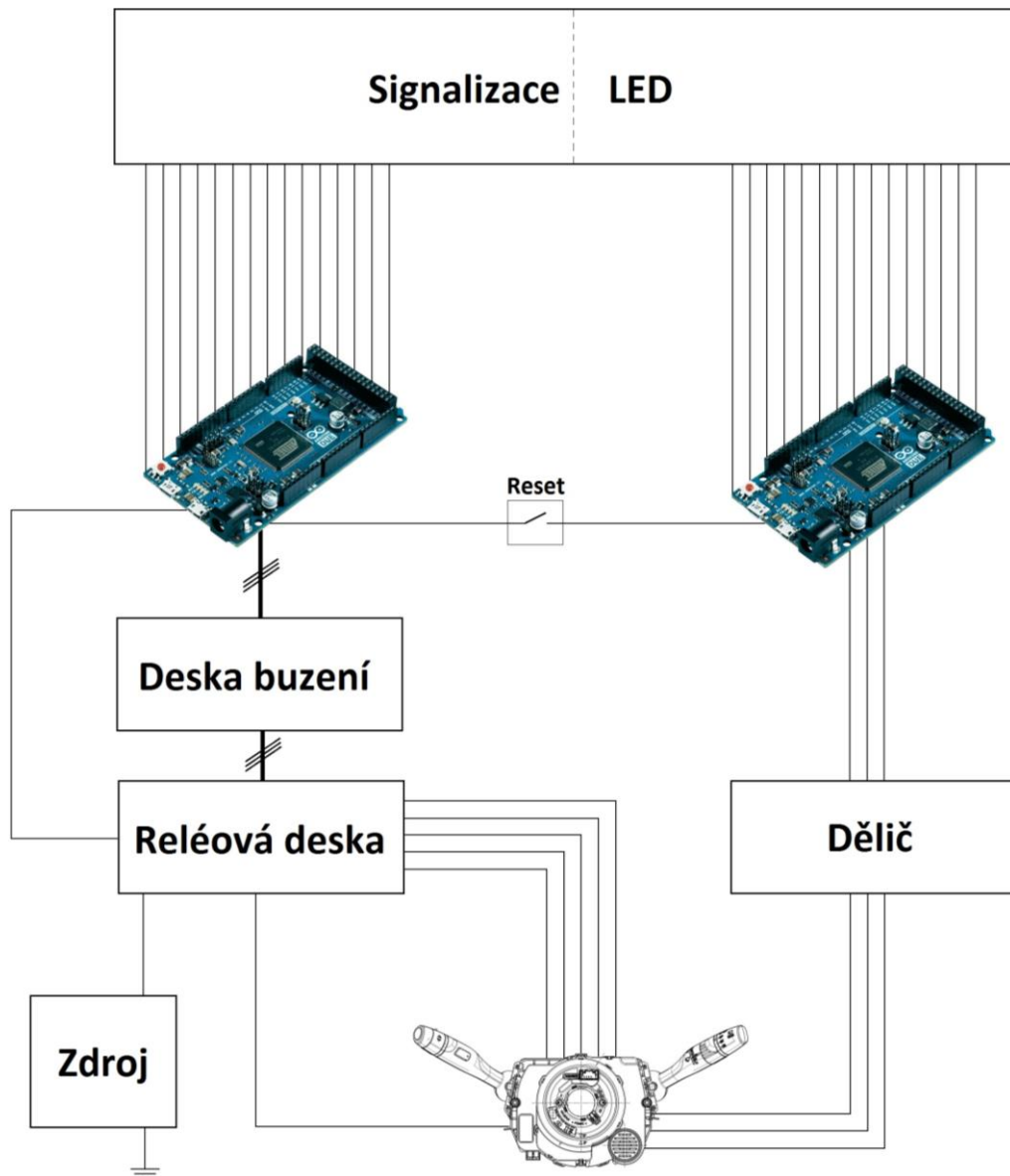
Požadovaný výkon na desku je v tomto případě v mezích.

Obr.5 Vnitřní uspořádání



Obr.6  
Signalizační, budící a reléová deska pro testování průchozích vodičů kazet.

## 7.1.2 Blokové schéma zapojení

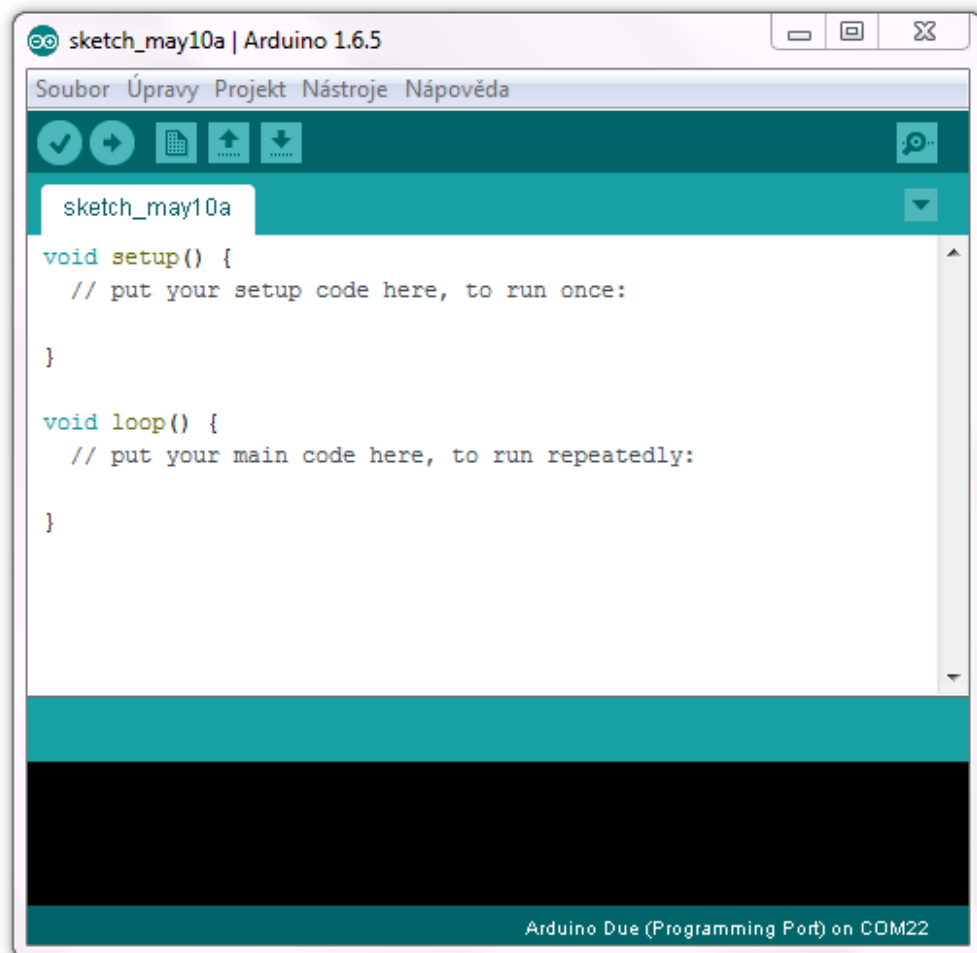


Obr.7  
Kompletní tester

Signalizační diody jsou rozděleny do třech částí. Nahoře vodiče kazety, vlevo dole páka blinkrů a vpravo dole páka stěračů. Diody jsou dvoubarevné. Pro signalizaci OK stavu je modré světlo, pro NOK stav je použita červená barva.

### 7.1.3 Program

Samotné programy pro Arduina jsou psané v grafickém vývojovém prostředí postaveném na prostředí Wiring a Processing. Kód obsahuje úvodní deklaraci a inicializaci proměnných a vstupně / výstupních pinů. Následují funkce, které se provedou pouze jednou (např. inicializace pole pomocí cyklu). Nakonec je zavedena smyčka, která se neustále opakuje. V této smyčce jsou snímány vstupy Arduina a podle hodnoty je provedena příslušná operace a reakce výstupu. Celý proces se opakuje pokud na pin resetu přivedeme zem (vyresetujeme Arduino). Vývojové prostředí provede i kompilaci a přenesení programu do Arduina, takže manipulace je velice snadná.



Obr.8 IDE pro Arduina

## 7.2 Tester s komunikací po sběrnici

V současnosti je většina podvolantových modulů osazena řídicí jednotkou, která zajišťuje komunikaci a základní zpracování signálů a dat z ovládacích prvků či jiných řídicích jednotek. Komunikace je řešena prostřednictvím sběrnic, díky kterým jsme schopni přenést velký objem dat přes malý počet vodičů. Oproti modulům, které nekomunikovali po sběrnici, se tímto razantně snížil objem vodičů v automobilech což mělo velký vliv na prostor a hmotnost.

Pro navázání komunikace po sběrnici potřebujeme transceiver, který nám převádí a budí data z PC na protokol používaný na dané sběrnici a naopak. Také je třeba využít software, který je schopný s těmito daty pracovat. Pro tento tester jsem využil HW a SW od společnosti Vector Informatik GmbH.

Výhoda použití tohoto softwaru je velká, jelikož většina automobilek využívá pro návrh řídicích jednotek právě tento program. Jeho obsáhlost a ovládání nám dovoluje analyzovat signály jakýmkoliv potřebným způsobem, přesností, vizualizací a to v krátkém čase. Dále je třeba vyčítat paměť, chybové hlášky a nastavení či kalibraci modulu. Použití HW od Vector Informatik GmbH je výhodné z důvodu kompatibility.

Na modulu ale nemůžeme kontrolovat jen data putující po sběrnici či data uložená v pamětech. Je třeba snímat hodnoty napětí a proudů a průchodnost vodičů. Proto je potřeba použít převodní a snímací desku pro tato měření a tomu přizpůsobit zapojení. Vizualizace je postavena na programu CANoe stejně jako celý program.



Obr.9  
Různá řešení sběrnicevých  
převodníků od firmy  
Vector Informatik GmbH

## 7.2.1 Popis zapojení testeru

Jak už jsem předeslal, tester se skládá z komunikačního převodníku od firmy Vector Informatik GmbH podle typu sběrnice. Dále pak ze snímací desky od National Instrument – NI6008/6009 OEM, která se stará o snímání napěťových signálů. Tyto signály jsou generovány převodní deskou, která řeší snímání a převod proudů, převod úrovní napětí a ovládání. Tester obsahuje zdroj schopný dodat proud, který je třeba pro simulaci ohřevu topné spirály volantu. Pro tento tester je zapotřebí počítač s operačním systémem, na kterém poběží software CANoe. V tomto prostředí je naprogramován celý testovací program a vizualizace pro snadné ovládání.

Uložení všech komponent je v kovové technické krabici určené pro měřicí přístroje. Je přidán ventilátor pro odvod tepla ze zátěžového odporu, který slouží k testu ohřevu. Dále pak USB hub, který spojuje komunikaci NI modulu a sběrnicevého převodníku do jediného USB kabelu.

Propojení NI modulu a CANoe je pomocí knihovny, která je vytvořena nastavovacím programem od NI. Po nastavení všech vstupů a výstupů NI jsou data uložená v knihovně po importování do CANoe postupně přiřazena k jednotlivým proměnným v programu. U budičů není tak složité nastavování třeba, jelikož je HW i SW od stejné firmy a jejich propojení je již v základu programu CANoe. Stačí jen vybrat druh převodníku a nastavit správnou přenosovou rychlost.

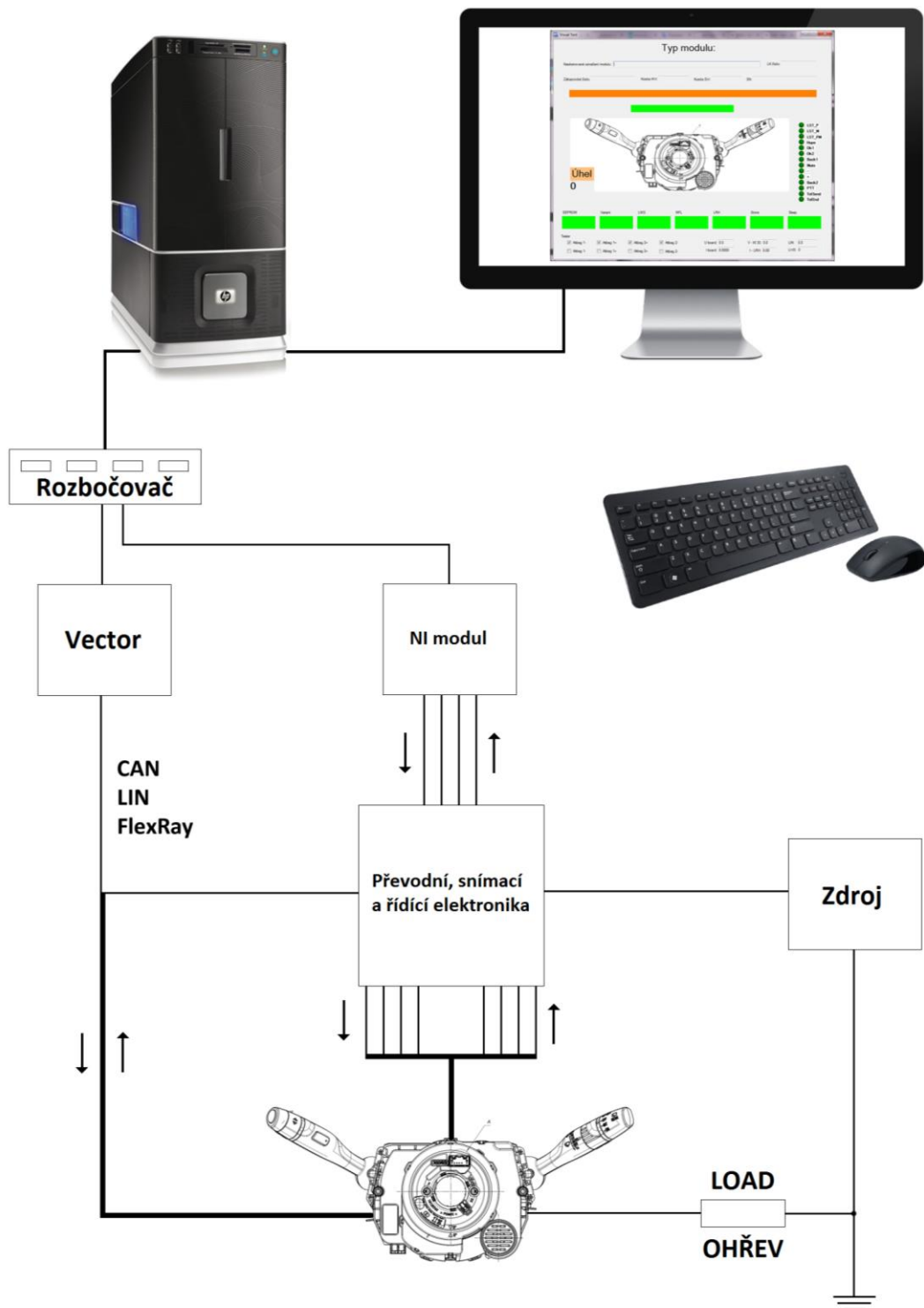
Tester je připojen do sítě 230V přes pojistku a spínač se signalizací. S počítačem je propojen pomocí USB kabelu a na propojení k upínacímu přípravku jsou k dispozici dva konektory CAN25.

Obr.10  
NI modul typ OEM. Je otevřený a obsahuje patici díky které je možno modul připojit přímo k převodní elektronice.

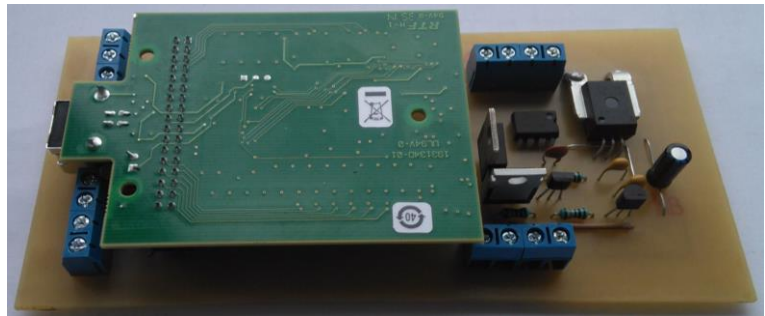




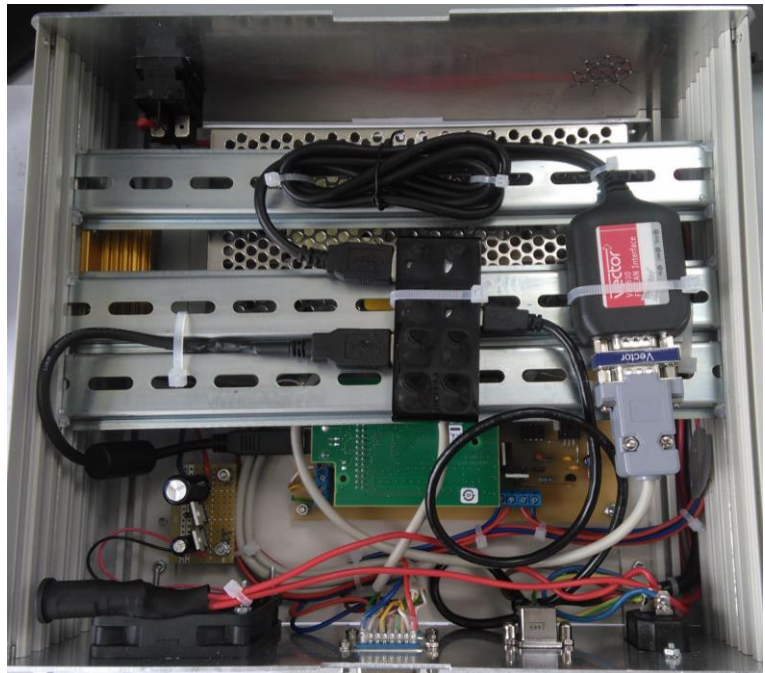
## 7.2.2 Blokové schéma zapojení



Obr.11  
Převodní a snímací deska  
s nasazeným NI modulem



Obr.12  
Vnitřní uspořádání testeru.  
Uložení je řešeno ve 2  
vrstvách.



Obr.13  
Přední panel testeru



Obr.14  
Zadní panel testeru



## 7.2.3 Program

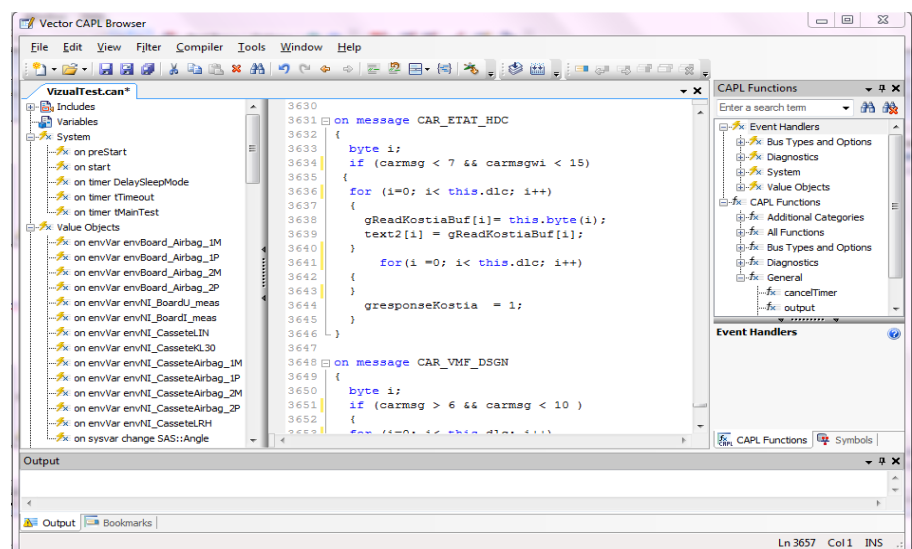
Testovací program je řešen v SW CANoe. Toto vývojové prostředí neslouží pouze k vytváření testovacích programů, ale především k simulaci řídicích jednotek komunikujících po sběrnících. Pro hloubkovou analýzu komunikací a přenášených dat. Pro návrhy komunikace řídicích jednotek atd..

Jelikož má CANoe přehledný editor vizualizací, byl využit pro vytvoření ovládacího panelu pro testy podvolantových jednotek. Tento panel je propojen s programem psaným v editoru CAPL. Syntaxe je založena na programovacím jazyce C++/C#. Stavba programu je naprogramována jako stavový automat.

Program začíná načtením čárového kódu výrobku. Z tohoto kódu jsou vytažena data o variantě výrobku, verzi nahraného SW, verzi HW, sériové číslo aj. Proveďte automatickou kontrolu EEPROM kde jsou tyto data zapsaná a porovná je s daty z čárového kódu. Tyto data jsou pak dále porovnávána v ini souborech, pomocí kterých se vyberou testovací sekvence pro jednotlivé varianty pák a modulů. Následuje test všech funkcí pák. Tento test provádí podle pokynů v panelu vizualizace operátor. Pokud není splněna některá z podmínek, test se dá opakovat 3x a pak je výrobek vyhodnocen jako vadný a test končí. Po testu pák následuje kontrola snímače úhlu natočení volantu, kontrola průchozích signálů kazety, test ohřevu volantu aj. Ke konci každého testu se musí vyčistit paměť chyb. Může obsahovat povolené chyby, které vznikají vždy, ale také chyby, které indikují nějaký problém při testech. Takový modul je pak vyřazen jako vadný. Po kontrole chyb se musí celá chybová paměť vymazat. Poslední test je test sleep režimu, kde se zkusí jak jednotka přechází do “standby”. Při každém testu se vytváří protokol o měření ve formátu html.

Obr.15

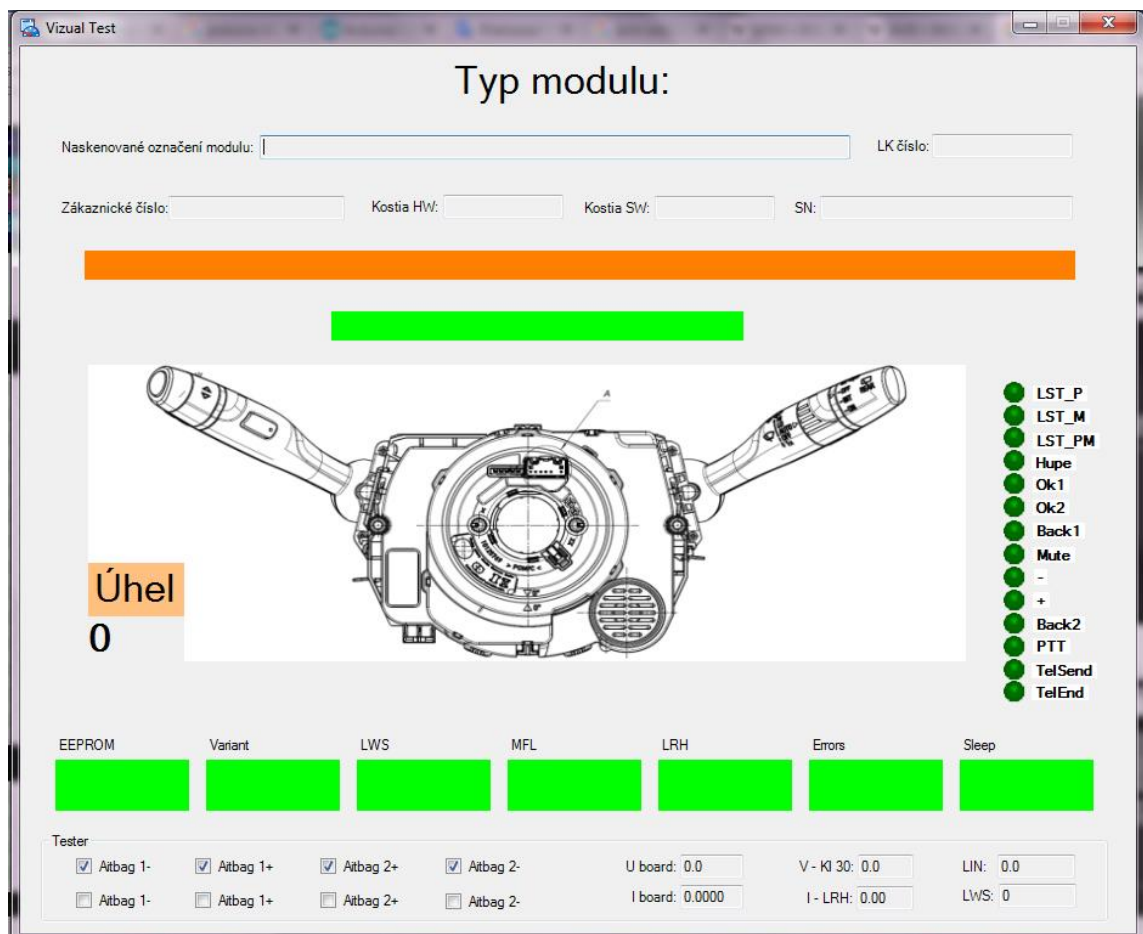
Vývojové prostředí  
CAPL



Systém programování v CAPL je trochu odlišný. CANoe má řadu vlastních funkcí vytvořených právě pro speciální použití pro vývoj ve sběrnicové komunikaci. Hlavním znakem je to, že je celý program postavený na sledování signálů či zpráv. Jedná se o takzvané řízení událostmi, kdy sledujeme tyto signály a zprávy a reagujeme při jejich změně.

Také můžeme hlídat obsah paměti, kde v určitých intervalech nebo v námi definovaném čase vyčítáme jednotlivé hodnoty a porovnáváme je.

Pro připojení jiného HW než od společnosti Vector má CANoe k dispozici import knihoven externího HW, pomocí kterých propojíme námi používané proměnné s jednotlivými porty či signály daného externího zařízení. Pomocí této funkce se nemusíme při testování omezovat pouze na komunikaci po sběrnících, ale můžeme kontrolovat i analogové veličiny na měřeném výrobku. V mém případě spínáme pomocí CANoe napájecí napětí k měřenému modulu. Snímáme odběr modulu, velikost proudu při kontrole ohřevu volantu, napětí na modulu či provádíme postupným spínáním portů modulu NI kontrolu průchozích vodičů kazety.



Obr.16 Panel vizualizace

### 7.3 Tester s vizualizací pro analogové ovládací prvky

Toto poloautomatické testovací zařízení je pro analogové ovládací prvky podvolantových řídicích modulů, které neobsahují řídicí jednotku. Systém je podobný jako u předchozího testeru. Program je však vytvořený v prostředí LabVIEW a neobsahuje HW pro komunikaci po sběrnících, tudíž jeho hlavní částí je NI 6008, který převádí napětové signály do komunikace po USB.

Tester tedy dokáže otestovat páky blinkrů, světel, stěračů a jiné dodatečné funkce, jelikož výstupem z konektoru takových modulů jsou ovládací signály určitých úrovní a dále pak průchozí kabeláž pro volant. Pro testování funkcí jsou páky stimulovány pracovníkem. Kontrola průchozích vodičů probíhá automaticky.

Místo signalizace diodami, jako u prvního testeru, je opět vytvořená vizualizace pro snadné ovládání a více možností analýz. Program je schopný vytvářet logovací soubory o průběhu každého testu. Tester je univerzální tak aby na něm bylo možno testovat jakékoliv analogové signály. Pro různé moduly se tedy mění pouze konektor (kabeláž) a konfigurace pro testovaný modul.

Testovací zařízení se skládá z upínací mechaniky, kabeláže, testovací elektroniky, počítače, displeje a testovacího programu.



Obr.18  
NI 6008. Je uzavřený v krabičce a připojení je řešeno šroubovými svorkami.



Obr.17

Samotná blinkrová páka. Tento tester je určený právě pro testování samostatných řídicích pák.

### 7.3.1 Popis zapojení testeru

Základ měřicího přístroje tvoří univerzální modul NI USB 6008 od firmy National Instruments, který obsahuje ve svém základu 8 x analogový vstup, 12 x univerzální binární vstup/výstup a 2 x analogový výstup. Komunikace modulu je univerzálním standardizovaným rozhraním USB. Napájení modulu je pomocí rozhraní USB z připojeného počítače. Přístroj UNILAB doplňuje tento modul o vstupně-výstupní rozhraní umožňující typizované napojení testovaných komponent.

Vzhledem ke skutečnosti že modul NI USB 6008 má fyzicky pouze 12 univerzálních vstupně-výstupních signálů, je přiřazení těchto signálů k fyzickým vstupům a výstupům řešeno takto:

- Základní první čtveřice fyzických výstupů je pevně přiřazená jako výstupy 1 až 4 modulu NI USB 6008. Z tohoto principu je nutné vždy tyto signály konfigurovat v SW jako výstup.
  
- Základní první čtveřice fyzických vstupů je pevně přiřazená jako vstupy 9 až 12 modulu NI USB 6008. Z tohoto principu je nutné vždy tyto signály konfigurovat v SW jako vstup.
  
- Signály 5 až 8 jsou univerzální, pokud se v SW nadefinují jako výstup, budou ovládat výstupní relé 5 až 8. V tomto případě musí zůstat fyzické vstupy 5 až 8 nezapojené. Pokud nadefinujeme signály 5 až 8 modulu NI USB 6008 jako vstupy, budou reprezentovat fyzické vstupy 5 až 8. Průvodním jevem takto zapojených vstupů bude současné spínání výstupních relé 5 až 8.

Další hlavní částí testeru je převodní deska (obsahuje i stejnosměrné zdroje). Tato deska nám upravuje signály pro modul NI a to následovně:

*Analogové vstupy* - Změna rozsahu vstupního signálu z 10V na 20V DC.

- Zvýšení vstupního odporu z 130 K $\Omega$  na více jak 10 M $\Omega$ .

- doplnění vstupních ochranných obvodů proti přepětí.

*Binární výstupy* - Převod binárních výstupních signálů z úrovně TTL na výkonový reléový kontakt.

*Binární vstupy* - Převod binárních vstupů z úrovně TTL na spínání vstupů kontaktním způsobem.

*Analogové výstupy* - Převod analogového výstupního signálu z 0 až 5 V na 0 až 20V s proudovým zesílením.

Napájení testeru je pomocí adaptéru 24V. Pro měřicí účely má přístroj vestavěny dva zdroje referenčního napětí +5V a +12V. Výstupy těchto zdrojů jsou zkratu vzdorné o maximálním proudovém zatížení 0,5A impulsně (omezeno trvalým ztrátovým výkonem 2W). Při návrhu přípravku je nutno vždy zahrnout skutečnost že analogový výstup je dvou vodičový a na vodičích vznikají napěťové ztráty vlivem odporu vodičů. Toto omezení je dáno vyzářeným tepelným výkonem sériových stabilizátorů, kdy toto ztrátové teplo ovlivňuje přesnost měření. Tento ztrátový výkon je optimální do vyzářené hodnoty 2 W trvale.

Výpočet tohoto trvalého ztrátového výkonu je následující: Ztrátový výkon [W] = Napájecí napětí přístroje [V] – výstupní napětí [V] \* předpokládaný proudový odběr [A].

Příklad 1:

Předpokládaný proudový odběr 50 mA na napětí +5V, na napájení +12V nebude žádný odběr. Ztrátový výkon =  $24 - 5 * 0,05$ . Což se rovná 0,95W.

Příklad 2:

Předpokládaný proudový odběr 10 mA na napětí +5V, na napájení +12 V proudový odběr 100mA.  $P_{5V} = 24 - 5 * 0,01 = 0,19$  W,  $P_{12V} = 24 - 12 * 0,1 = 1,2$  W.  $P_{5V} + P_{12V} = 1,39$  W. Celkový ztrátový výkon je menší než 2 W.

Příklad 3:

Předpokládaný proudový odběr na +12V je 0,5 A, na +5V nebude žádný odběr.  $P_{12V} = 24 - 12 * 0,5 = 6$  W. Tento vyzářený výkon překračuje výrazně hodnotu 2W. Pro tento předpokládaný odběr je nutné provést časové rozfázování odběru tak, aby průměrná hodnota nepřesáhla 2 W. Například: 2s připojen odběr a 4s bez odběru.

**Popis významu prvků na předním panelu:**

Spínač napájení – páčkový spínač odpojuje vstupní napájecí napětí 24V DC přístroje.

Led diody – označené +5V, +12V a +22V signalizují připravenost zdroje po zapnutí k správné funkci.

Led dioda – označená USB signalizuje svitem připravenost správného připojení rozhraní USB. Funkce Led diody je nezávislá na napájecím napětí přístroje.

Led dioda – označená RDY signalizuje stav analogové výstupu č. 2, tzn. svit LED diody je úměrný výstupnímu napětí analogové výstupu č. 2.



Obr.19  
Přední panel  
testeru

**Popis významu prvků na zadním panelu:**

Napájecí konektor – do tohoto konektoru se zapojuje napájecí zdroj o jmenovitém napětí 24 VDC tak, že levý připojovací kolík je plus a pravý připojovací kolík je mínus. Klíč pro orientaci konektoru je umístěn dole. Zdroj musí být stabilizovaný. Proudová zatížitelnost zdroje musí být minimálně 1 A.

Kalibrační zdičky – jsou určeny pro průběžnou kontrolu napájecích napětí přístroje. Napětí +2,5 V je vyvedené referenční napětí modulu NI USB 6008.

Konektor USB – slouží pro napojení komunikace USB s měřícím počítačem. Led dioda viditelná ve výřezu v panelu svitem signalizuje připravenost komunikace USB.

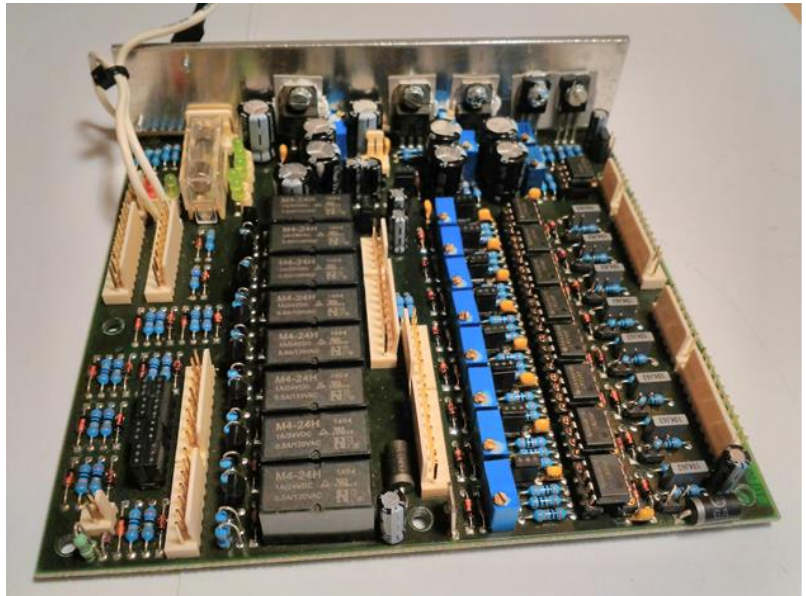


Obr.20  
Zadní panel  
testeru



Obr.21

Osazená převodní deska.  
Propojení s modulem NI je  
realizováno vodiči  
s konektory. Deska obsahuje  
i zdroje +5V a +12V.



Obr.22

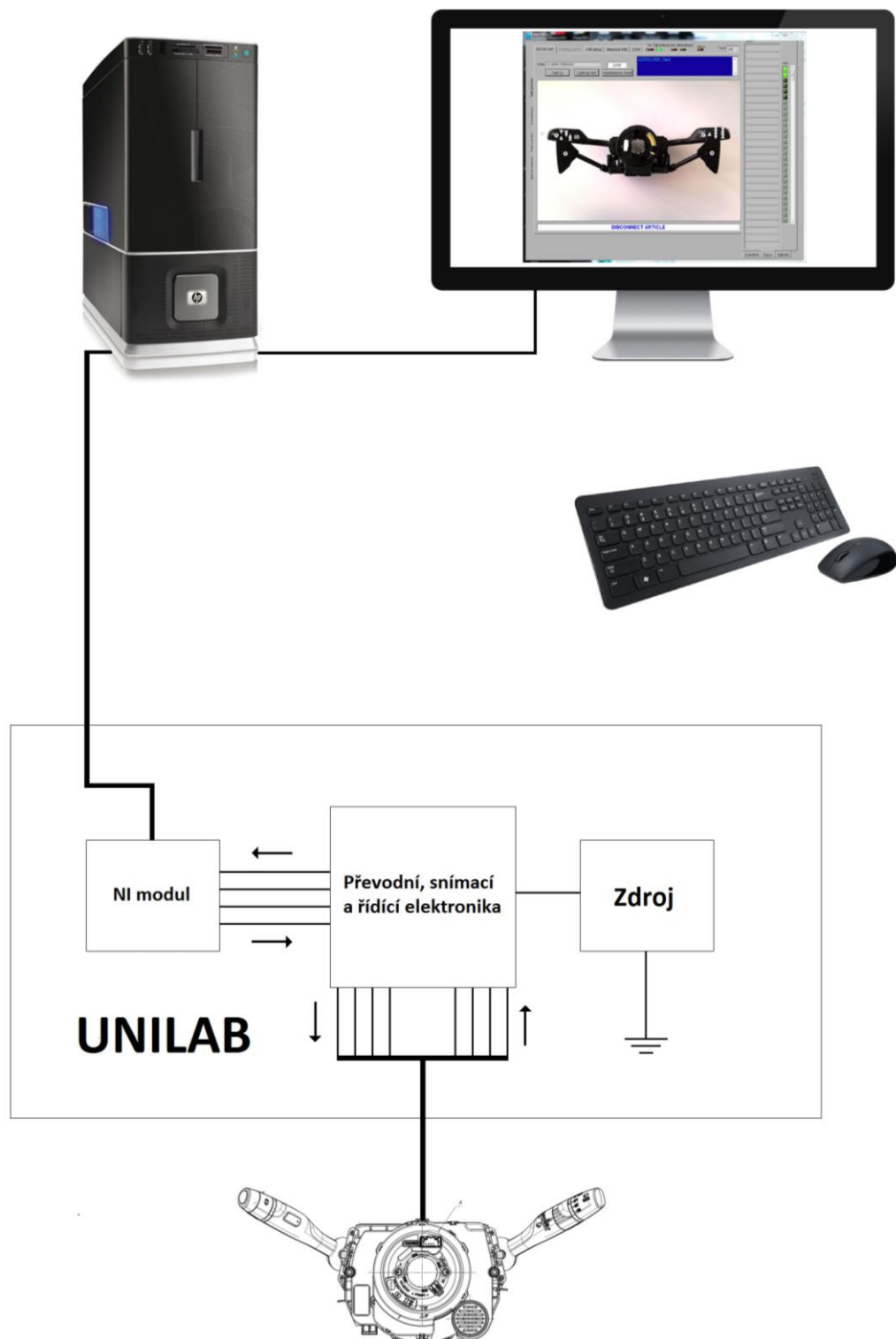
Kompletní tester z pohledu  
zevnitř. Vlevo nahoře deska  
měření proudu a pod ní je  
uložen modul NI.



### ***Pokyny pro provoz***

Přístroj je určen pro trvalý nepřetržitý provoz a nevyžaduje obsluhu. Pro účely periodických kontrol a údržby, je možno provádět čištění povrchu modulu propylalkoholem tak, aby nedošlo k zatečení kapaliny do přístroje. Pokud vnikne do přístroje kapalina která by mohla narušit elektrickou pevnost a způsobit porušení kalibrace, je bezpodmínečně nutné po vyčištění a vysušení provést kontrolu funkce, popřípadě kontrolu kalibrace. Kontrolu kalibrace přístroje je nutno provádět minimálně jedenkrát za rok.

### 7.3.2 Blokové schéma zapojení

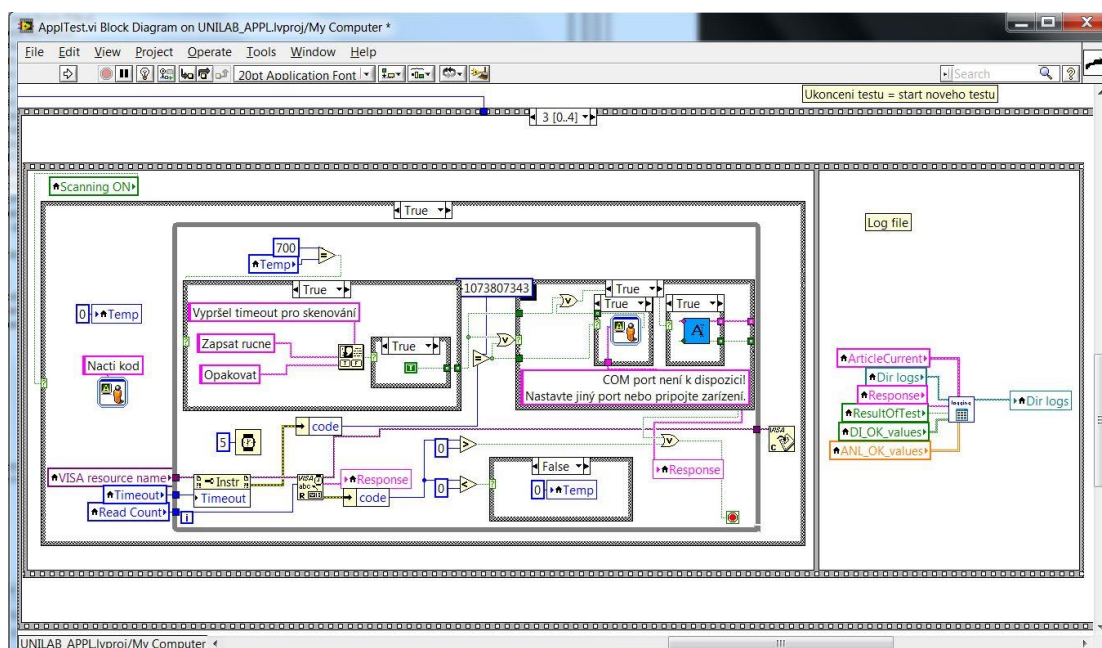


### 7.3.3 Program

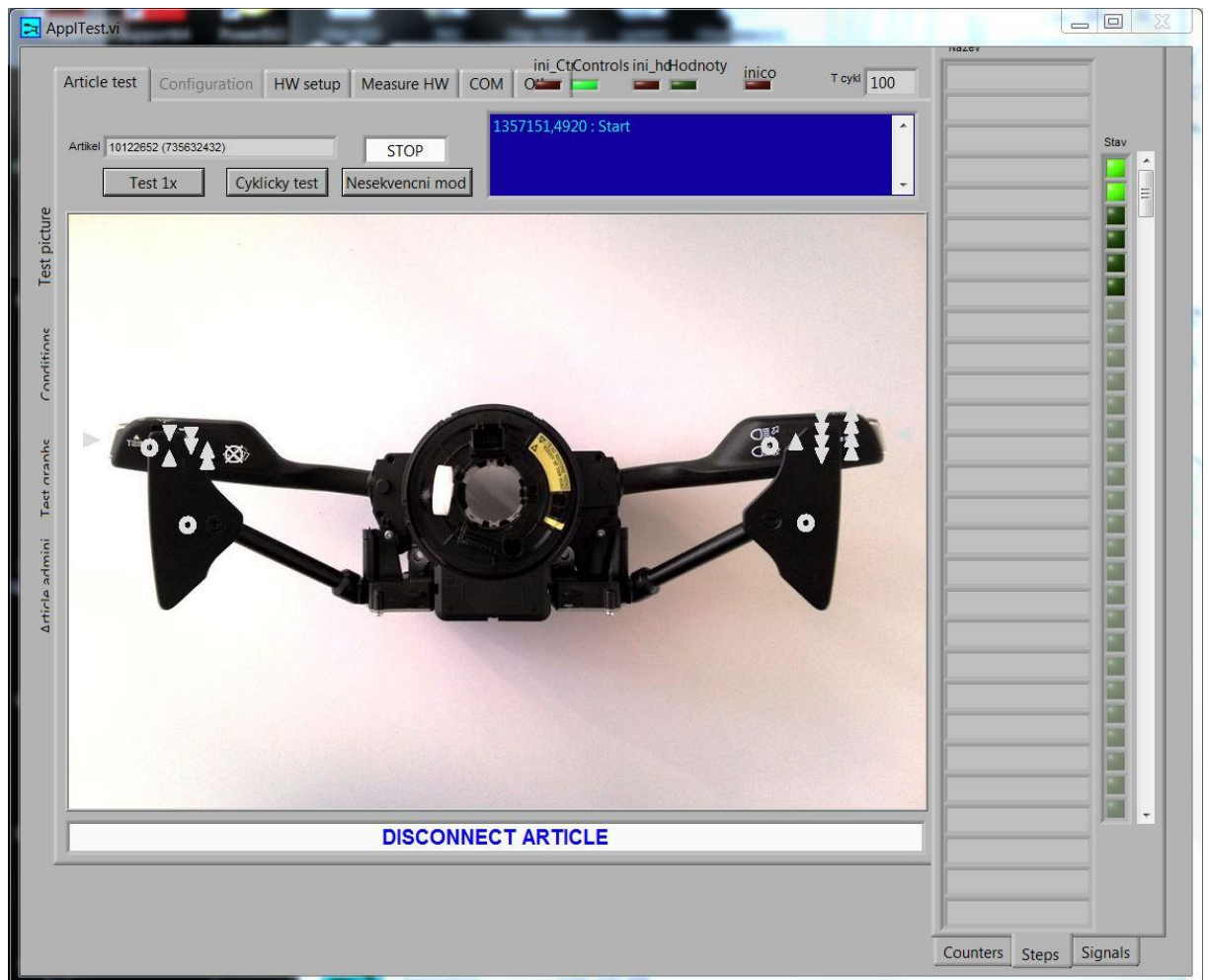
Velká variabilita, přehlednost, snadná vizualizace, to vše byli důvody výběru vývojového prostředí LabView. Na rozdíl od CANoe je LabView grafický programovací jazyk ve kterém se vytváří takzvané virtual instruments což jsou virtuální přístroje, které simulují skutečné přístroje. Díky tomu jsme schopni v LabView vytvořit téměř cokoliv. Díky grafickému prostředí a systémem blokového programování je celý proces rychlejší a jednodušší než u klasických programovacích jazyků.

Samotný program pro tester je vytvořen jako editor pro vytváření různých jednotlivých testovacích konfigurací aplikovaných v testeru. Pro každý podvolantový modul vytváříme na míru konfiguraci, která obsahuje testování funkcí daného modulu. Tato konfigurace je spuštěna přímo v editoru. Jedná se o jediný program ve kterém vytváříme nové testovací konfigurace přesně na míru a zároveň v něm dané konfigurace spouštíme a testujeme. Díky tomu má pracovník přístup ke všem funkcím programu kromě záložky vytváření konfigurací. Pak je schopen analyzovat test do hloubky, jelikož má k dispozici všechny nástroje pro analýzu.

Test je možný spustit jako cyklický, 1x provedený či nesequenční mód, kdy není potřeba vykonávat kroky popořadě. Při nesplnění podmínce lze test opakovat, ukončit a nebo lze přeskočit daný krok. Při přeskočení či ukončení je celkový test vyhodnocen jako vadný. Při každém skončení testu je vytvořen protokol o měření ve formátu csv.



Obr.23 Část programu pro skenování čárového kódu



Obr.24 Vizualizace v programu LabView

## 8 POROVNÁNÍ NAVRŽENÝCH TESTERŮ

Každý z navržených testerů má své specifické zaměření i když se jejich možnosti v různých aspektech překrývají. LED tester byl vytvořen z důvodu namátkové kontroly výrobků u zákazníka, kdy si chce firma jednoduše ověřit funkci výrobku. Zda-li je opravdu vadný nebo je chyba na jejich straně. Tento tester je bez mechanického přípravku na uchycení. Kontaktování je řešeno vodiči ukončenými zákaznickými konektory, jejichž protikus je na straně výrobku. Signalizace je pomocí LED svítí/nesvítí neboli OK/NOK. Jelikož tester neobsahuje PC ani upínací přípravek, náklady na jeho výrobu jsou ze všech nejnižší. Stejně tak i jeho využitelnost.

Druhý tester je vytvořený pro komplexní analýzu vyráběných modulů. Díky použitému SW a HW od firmy Vector jsme schopni komunikovat po sběrnicích a zároveň provádět hloubkovou analýzu této komunikace. Díky vizualizaci je ovládání přehledné a jednoduché. Pomocí knihoven lze připojit i měřicí karty, které nám dovolují snímat i analogové veličiny. Díky všem těmto vlastnostem je tento tester nejuniverzálnější. Vzhledem k použitým částím jako jsou PC, upínací přípravek a především HW a SW od firmy Vector, je ale tento tester nejdražší a to podstatně. Je tedy dobré si rozmyslet, zda-li je doopravdy zapotřebí.

Třetí a poslední tester je jakási kombinace dvou předchozích. Neobsahuje řešení pro komunikaci po sběrnici, ale obsahuje měřicí kartu od firmy NI. Také počítač na kterém běží program s vizualizací testu. NI karta nám převádí měřené analogové signály do počítače, kde je zpracováváme v programu napsaném ve vývojovém prostředí LabView. Tester tedy dokáže testovat jakékoliv analogové ovládací prvky a díky vizualizaci je test přehledný a rychlý. Primárně je určen na testy samostatných ovládacích pák podvolantových modulů. Cena se pohybuje mezi cenou prvního a druhého testeru. Závislá je nejvíce na ceně upínacích přípravků pro páky.

## 9 ZÁVĚR

V automobilovém průmyslu jsou kladeny vysoké nároky na kvalitu. Proto je třeba veškeré výrobky testovat z důvodu eliminace vadných kusů. Testování probíhá již na úrovni výroby výrobku, kdy se testují postupně jednotlivé části a na konci linky pak celý výrobek. Tyto testy jsou součástí výrobní linky. Nejsou to ovšem jediné testy, které se při výrobě provádějí. Pro udržování správné kvality je třeba vyhodnocovat kritické znaky a provádět namátkovou kontrolu. Také se provádí kontroly na poloautomatických testech po vyrobení celého výrobku i když je vyhodnocen jako dobrý. Je to z důvodu předpisů a momentálních požadavků zákazníka. Pro všechny tyto kontroly mimo výrobní linku se používají testery popsané v této práci. Na různé kontrolované znaky či funkce se používají různé druhy testerů navržené právě k danému účelu. Dal by se používat jeden jediný univerzální tester pro vše, ovšem náklady by byli výrazně vyšší. Z porovnání je jasné, že se nedá určit který tester je lepší, jelikož každý byl navržen pro určitou funkci s ohledem na vynaložené náklady. Jelikož je tester pro analogové ovládací prvky vyvíjen již 3-tím rokem, je jeho návrh nejkompaktnější. Ostatní dva testery jsem navrhoval před několika měsíci, jejich použití je momentálně zaváděno do výroby.

## 10 Seznam literatury a informačních zdrojů

- [1] © 2016 National Instruments Corporation. Webpage USB-6008 OEM. Dostupné z < <http://sine.ni.com/nips/cds/view/p/lang/cs/nid/202751> >
- [2] ©2016 Arduino. Webpage Arduino Due. Dostupné z < <https://www.arduino.cc/en/Main/ArduinoBoardDue> >
- [3] Ing.Radek Taraba, Aplikování sběrnice CAN. 9. Listopad 2004  
Dostupné z < <http://vyvoj.hw.cz/navrh-obvodu/rozhrani/aplikovani-sberrnice-can.html> >

## Přílohy

### Technické údaje testeru pro analogové ovládací prvky

#### **Analogové vstupy:**

Počet vstupů 8

Rozsah vstupního signálu 0,5 až 20 V DC

Rozlišení analogově digitálního převodu 11 bit

Vstupní odpor jednoho měřicího kanálu > 10 Mohm

Přesnost měření v měřicím rozsahu 0,5 až 20 V < 0,5%

Třída přesnosti 0,5%

#### **Analogové výstupy:**

Počet výstupů 2

Rozsah výstupního napětí 1 až 19 V DC

Rozlišení digitálně analogového převodu 12 bit

Přesnost výstupního napětí při zátěži  $I = 10\text{mA} < \pm 20\text{mV}$

Přesnost výstupního napětí při zátěži  $I = 100\text{mA} < \pm 40\text{mV}$

Přesnost výstupního napětí při zátěži  $I = 500\text{mA} < \pm 80\text{mV}$

Maximální výstupní proud impulsní viz. text 0,5 A

#### **Binární vstupy:**

Počet vstupů 8

Jmenovité vstupní napětí 2 až 24 V DC

Logická vstupní úroveň "L" < 0,3 V DC

Jmenovitý vstupní proud - vstup 1 mA

Odpor pro logickou vstupní úroveň "L" < 200 W

#### **Binární výstupy:**

Počet výstupů 8

Odpor kontaktu v sepnutém stavu Max. 100 mW

Nominální zatížení 1 A 30 V DC

Max. spínací výkon 62 VA, 30W

Max. spínací proud 1 A



Elektrická životnost 1 x 10<sup>5</sup>

**Napájecí výstup +5V:**

Jmenovité výstupní napětí + 5,0 V DC < ±10mV

Přesnost výstupního napětí při zátěži I = 10mA + 5,0 V DC < ±15mV

Přesnost výstupního napětí při zátěži I = 100mA + 5,0 V DC < ±18mV

Přesnost výstupního napětí při zátěži I = 500mA + 5,0 V DC < ±25mV

Maximální výstupní proud impulsní viz. text 0,5 A

**Napájecí výstup +12V:**

Jmenovité výstupní napětí + 12,0 V DC < ±15mV

Přesnost výstupního napětí při zátěži I = 10mA + 12,0 V DC < ±18mV

Přesnost výstupního napětí při zátěži I = 100mA + 12,0 V DC < ±25mV

Přesnost výstupního napětí při zátěži I = 500mA + 12,0 V DC < ±40mV

Maximální výstupní proud impulsní viz. text 0,5 A

**Ostatní údaje:** Napájecí napětí přístroje 24 V DC - 5 %, +20%

Napájecí proudový odběr - klidový 0.05 A max.

Napájecí proudový odběr přístroje – maximální 1.0 A max.

Vestavná pojistka T-1A, 5x20mm

El. pevnost – výstupního kontaktu 500 V / 50 Hz

Elektrická pevnost – kabeláž – přístroj – upevnění 500 V / 50 Hz

Max. celkový ztrátový výkon přístroje 5 W max.

Krytí IP 20

Elektromagnetická třída odolnosti vůči rušení dle ČSN EN 50082-1

Vyzařované rušení dle ČSN EN 55022 třída A

Ostatní normy EN55024:1998(Class B), ČSN EN 62000-6-2

Pracovní teplota okolí +18 až +26°C

Vibrace 0,1 mm, 10 až 55 Hz

Doporučená skladovací teplota 10 až 30 °C

Rozměry š x d x v 265 x 190 x 82 mm

Hmotnost 0,5 kg

## Programy pro LED tester

### Program pro kontrolu kazety

```
const int vstup0 = A0;

const int kabel[18] = {3,4,5,6,7,8,9,10,11,12,13,22,23,24,25,26,27,28};
const int kabel0 = 2;
const int LEDkabel[18] =
{30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47};

int test = 0;
int hodnota0 = 0;
int hodnota[100];
int hotovo = 0;

void setup() {

Serial.begin(9600);

pinMode(kabel0, OUTPUT);
digitalWrite(kabel0, LOW);

for (int i = 0; i < 18; i++) {

    pinMode(kabel[i], OUTPUT);
    pinMode(LEDkabel[i], OUTPUT);
    digitalWrite(kabel[i], LOW);
    digitalWrite(LEDkabel[i], LOW);
}

Serial.println("");
Serial.println("-Zacatek mereni-");
Serial.println("");
}

void loop() {

    delay(100);
    if (hotovo == 0) {

Serial.println("-Nove mereni-");

for (int j = 0; j < 18; j++) {

    digitalWrite(kabel[j], HIGH);
    delay(100);
    digitalWrite(kabel0, HIGH);
    delay(100);

for (int i = 0; i < 99; i++) {

    hodnota[i] = analogRead(vstup0);
    hodnota0 = hodnota0 + hodnota[i];
    delay(2);
}

    hodnota0 = hodnota0 / 100;
    Serial.print("Mereni kabelu ");
```

```
Serial.print(j);
Serial.print(" = ");
Serial.println(hodnota0);

if ((hodnota0 > 200) && (hodnota0 < 750))
{
    digitalWrite(LEDkabel[j], HIGH);
}
else digitalWrite(LEDkabel[j], LOW );

digitalWrite(kabel0, LOW);
delay(100);
digitalWrite(kabel[j], LOW);

}

Serial.println("-Konec mereni-");
Serial.println("");

hotovo = 1;
}
}
```

**Program pro kontrolu pák**

```
const int analogPin0 = A0; // Analogový vstup je
nastaven na pi A0
const int analogPin1 = A1;
const int analogPin2 = A2;
const int analogPin3 = A3;
const int analogPin4 = A4;
const int analogPin5 = A5;
const int analogPin6 = A6;
const int analogPin7 = A7;
const int analogPin8 = A8;

const int LED_high_beam = 27;
const int LED_flash_light = 28;

const int LED_right_turn_light = 22;
const int LED_right_flash_light = 23;
const int LED_idle = 24;
const int LED_left_flash_light = 25;
const int LED_left_turn_light = 26;

const int LED_windshield_washer = 29;
const int LED_idle_washer = 30;
const int LED_windshield_wiper = 31;

const int LED_rear_continuous = 32;
const int LED_rear_interm = 33;
const int LED_rear_off = 34;

const int LED_continuous_high = 35;
const int LED_continuous_low = 36;
const int LED_auto_high = 37;
const int LED_auto_low = 38;
const int LED_off = 39;

const int LED_LDW = 40;
const int LED_EVIC = 41;

const int LED_tiptronic_plus = 42;
const int LED_tiptronic_minus = 43;

int hodnota = 0;

void setup() {

Serial.begin(9600);

pinMode(LED_right_turn_light, OUTPUT);
pinMode(LED_right_flash_light, OUTPUT);
pinMode(LED_idle, OUTPUT);
pinMode(LED_left_flash_light, OUTPUT);
pinMode(LED_left_turn_light, OUTPUT);
pinMode(LED_high_beam, OUTPUT);
pinMode(LED_flash_light, OUTPUT);
pinMode(LED_windshield_washer, OUTPUT);
pinMode(LED_idle_washer, OUTPUT);
pinMode(LED_windshield_wiper, OUTPUT);
pinMode(LED_rear_continuous, OUTPUT);
pinMode(LED_rear_interm, OUTPUT);
```

```

pinMode(LED_rear_off, OUTPUT);
pinMode(LED_continuous_high, OUTPUT);
pinMode(LED_continuous_low, OUTPUT);
pinMode(LED_auto_high, OUTPUT);
pinMode(LED_auto_low, OUTPUT);
pinMode(LED_off, OUTPUT);
pinMode(LED_LDW, OUTPUT);
pinMode(LED_EVIC, OUTPUT);
pinMode(LED_tiptronic_plus, OUTPUT);
pinMode(LED_tiptronic_minus, OUTPUT);
}

void loop() {

//----- High beam / Flash light-----

digitalWrite(LED_high_beam, LOW);
digitalWrite(LED_flash_light, LOW);

hodnota = analogRead(analogPin1);           // přečte hodnotu z A0
if ((hodnota > 454) && (hodnota < 474))
{
digitalWrite(LED_high_beam, HIGH);
}

else if ((hodnota > 795) && (hodnota < 815))
{
digitalWrite(LED_flash_light, HIGH);
}

Serial.print("analogPin1 = ");
Serial.println(hodnota);

//-----Turn light signals-----

digitalWrite(LED_right_turn_light, LOW);
digitalWrite(LED_right_flash_light, LOW);
digitalWrite(LED_idle, LOW);
digitalWrite(LED_left_flash_light, LOW);
digitalWrite(LED_left_turn_light, LOW);

hodnota = 0;
hodnota = analogRead(analogPin0);           // přečte hodnotu z A1
if ((hodnota > 856) && (hodnota < 876))
{
digitalWrite(LED_right_turn_light, HIGH);
}

else if ((hodnota > 945) && (hodnota < 965))
{
digitalWrite(LED_right_flash_light, HIGH);
}

else if ((hodnota > 371) && (hodnota < 391))
{
digitalWrite(LED_idle, HIGH);
}

else if ((hodnota > 742) && (hodnota < 762))
{
digitalWrite(LED_left_flash_light, HIGH);
}

```

```
}

else if ((hodnota > 591) && (hodnota < 611))
{
digitalWrite(LED_left_turn_light, HIGH);
}

Serial.print("analogPin0 = ");
Serial.println(hodnota);

//-----Washer signals-----

digitalWrite(LED_windshield_washer, LOW);

hodnota = 0;
hodnota = analogRead(analogPin3);           // přečte hodnotu z A0

if ((hodnota > 455) && (hodnota < 475))
{
digitalWrite(LED_windshield_washer, HIGH);
}

Serial.print("analogPin3 = ");
Serial.println(hodnota);

//-----Rear wiper signals-----

digitalWrite(LED_rear_continuous, LOW);
digitalWrite(LED_rear_interm, LOW);
digitalWrite(LED_rear_off, LOW);

hodnota = 0;
hodnota = analogRead(analogPin2);
if ((hodnota > 174) && (hodnota < 194))
{
digitalWrite(LED_rear_continuous, HIGH);
}

else if ((hodnota > 408) && (hodnota < 428))
{
digitalWrite(LED_rear_interm, HIGH);
}

else if ((hodnota > 671) && (hodnota < 691))
{
digitalWrite(LED_rear_off, HIGH);
}

Serial.print("analogPin2 = ");
Serial.println(hodnota);

//-----Windshield wiper signals-----

digitalWrite(LED_windshield_wiper, LOW);
digitalWrite(LED_continuous_high, LOW);
digitalWrite(LED_continuous_low, LOW);
digitalWrite(LED_auto_high, LOW);
digitalWrite(LED_auto_low, LOW);
digitalWrite(LED_off, LOW);

hodnota = 0;
```

```
hodnota = analogRead(analogPin4);           // přečte hodnotu z A1

if ((hodnota > 945) && (hodnota < 965))
{
digitalWrite(LED_windshield_wiper, HIGH);
}

if ((hodnota > 945) && (hodnota < 965))
{
digitalWrite(LED_continuous_high, HIGH);
}

else if ((hodnota > 855) && (hodnota < 875))
{
digitalWrite(LED_continuous_low, HIGH);
}

else if ((hodnota > 741) && (hodnota < 761))
{
digitalWrite(LED_auto_high, HIGH);
}

else if ((hodnota > 590) && (hodnota < 610))
{
digitalWrite(LED_auto_low, HIGH);
}

else if ((hodnota > 370) && (hodnota < 390))
{
digitalWrite(LED_off, HIGH);
}

if ((hodnota > 370) && (hodnota < 390))
{
digitalWrite(LED_idle_washer, HIGH);
}

Serial.print("analogPin4 = ");
Serial.println(hodnota);

//-----LDW signal-----

digitalWrite(LED_LDW, LOW);
hodnota = 0;
hodnota = analogRead(analogPin5);
if ((hodnota > 1010) && (hodnota < 1030))
{
digitalWrite(LED_LDW, HIGH);
}

Serial.print("analogPin5 = ");
Serial.println(hodnota);

//-----EVIC signal-----

digitalWrite(LED_EVIC, LOW);
hodnota = 0;
hodnota = analogRead(analogPin6);
if ((hodnota > 1010) && (hodnota < 1030))
{
```

```
digitalWrite(LED_EVIC, HIGH);
}

Serial.print("analogPin6 = ");
Serial.println(hodnota);

//-----Tiptronic plus signal-----

digitalWrite(LED_tiptronic_plus, LOW);
hodnota = 0;
hodnota = analogRead(analogPin7);
if ((hodnota > 690) && (hodnota < 710))
{
digitalWrite(LED_tiptronic_plus, HIGH);
}

Serial.print("analogPin7 = ");
Serial.println(hodnota);

//-----Tiptronic minus signal-----

digitalWrite(LED_tiptronic_minus, LOW);
hodnota = 0;
hodnota = analogRead(analogPin8);
if ((hodnota > 690) && (hodnota < 710))
{
digitalWrite(LED_tiptronic_minus, HIGH);
}

Serial.print("analogPin8 = ");
Serial.println(hodnota);

delay(100);

digitalWrite(LED_idle_washer, LOW);

}
```



## Část programu pro tester digitálních ovládacích prvků

### Test páky blinkru

```

case MAIN_TEST_VARIANT_BLINKER:

    @sysvar::CH_Nodes::CH_EIS_ISw_Stat = 4;
    putValue(envTestResultVariant,"test");
    mySetControl("IO_ResultVariantGreen", 1, colorBLUE);
    gMainTestState++;
    break;

case MAIN_TEST_VARIANT_BLINKER+1:
    if (gTestSKIP_Variant_Blinker == 0)
    {
        reportText("Blinker Lever Test",2);
        gLASTMainTestState = gMainTestState;
        gErrorCounter = 0;
        if (setupTest("B_",gTestBlinker, gBlinkerState, gPakySequence) < 1)
        {
            reportText_2string("Cannot open configuration CSV file
B_",gTestBlinker,1);
            reportError();
            gMainTestState = MAIN_TEST_RESULT_ERROR_CFG;
        } else
        {
            gSequencePos = 0;
            gStepState = 0;
            gMainTestState++;
        }
    } else
    {
        gMainTestState = MAIN_TEST_VARIANT_MVS;
        switchToTestModeKostia();
    }
    break;
case MAIN_TEST_VARIANT_BLINKER+2:
    if (testBlinker() == 0)
    {
        gMainTestState = MAIN_TEST_VARIANT_MVS;
        switchToTestModeKostia();
        gErrorCounter = 0;
    };

    break;

int testBlinker()
{
    char tempText[100];

    if (gStepState == 0) //zadani
    {

putValue(envTestInstruction,gBlinkerState[gPakySequence[gSequencePos]].instru
ction);
        sprintf(tempText, elcount(tempText), "TS %d - %s: ",gSequencePos ,
gBlinkerState[gPakySequence[gSequencePos]].report);

```

```
reportText(tempText, 1);

if (strlen(gBlinkerState[gPakySeqence[gSeqencePos]].picture) > 0)
{
    setControlVisibility("Vizual Test",
gBlinkerState[gPakySeqence[gSeqencePos]].picture, 1);
}

gStepState++;
startTimeout(TIMEOUT_NORMAL);
}

if (gStepState == 1) //vyhodnoceni
{
    if ((msgSCCM.dword(0) &
(dword)gBlinkerState[gPakySeqence[gSeqencePos]].maskL) ==
(dword)gBlinkerState[gPakySeqence[gSeqencePos]].valueL) &&
        (msgSCCM.dword(4) &
(dword)gBlinkerState[gPakySeqence[gSeqencePos]].maskH) ==
(dword)gBlinkerState[gPakySeqence[gSeqencePos]].valueH)
        { //splneno

            write("Test blinkru splneno");
            reportOK();
            stopTimeout();
            setControlVisibility("Vizual Test",
gBlinkerState[gPakySeqence[gSeqencePos]].picture, 0);
            gSeqencePos++;
            gStepState = 0;
            if (gPakySeqence[gSeqencePos] == -1)
                return 0;
        }
}
return 1;
}
```