ZÁPADOČESKÁ UNIVERSITA V PLZNI

FAKULTA ELEKTROTECHNICKÁ

KATEDRA TECHNOLOGIÍ A MĚŘENÍ

DIPLOMOVÁ PRÁCE

HiL testování – zpracování obrazu

Vedoucí práce: Ing. Radek Holota Ph.D.

Autor: Bc. Ladislav Matějka

Plzeň 2012

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: **2011/2012**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Ladislav MATĚJKA**
Osobní číslo:	**E10N0035P**
Studijní program:	**N2612 Elektrotechnika a informatika**
Studijní obor:	**Komerční elektrotechnika**
Název tématu:	**HIL testování - zpracování obrazu**
Zadávající katedra:	**Katedra technologií a měření**

Zásady pro vypracování:

1. Seznámení se s principem HIL testování a simulátorem uHIL od firmy MBTech Group.
2. Návrh testovacího stavu pro KOMBI přístroj.
3. Přehled možností rozpoznávání obrazu (SW nástroje, kamery).
4. Seznámení se se SW Provetech TA a SW NI Vision. Implementace NI Vision do SW Provetech.
5. Připravit ukázkové automatizované testy.

Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah pracovní zprávy: **30 - 40 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.

Vedoucí diplomové práce: **Ing. Radek Holota, Ph.D.**
Katedra aplikované elektroniky a telekomunikací

Datum zadání diplomové práce: **17. října 2011**
Termín odevzdání diplomové práce: **11. května 2012**

Doc. Ing. Jiří Hammerbauer, Ph.D.
děkan

L.S.

Doc. Ing. Vlastimil Skočil, CSc.
vedoucí katedry

V Plzni dne 17. října 2011

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni

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

Dále prohlašuji, že všechny informace obsažené v této diplomové práci jsou volně publikovatelné a nepodléhají zvláštnímu utajení.

Veškerý software použitý pro vypracování a řešení této práce, je legální.

Jméno a příjmení

V Plzni dne 7. 5. 2012 ……

Poděkování

Tímto bych rád poděkoval vedoucímu diplomové práce Ing. Radku Holotovi Ph.D za cenné a užitečné rady, návrhy a metodické vedení práce.

Dále bych chtěl poděkovat za užitečné rady a cenné informace konzultantům diplomové práce ze společnosti MBtech Bohemia s.r.o., jimiž byli Ing. Zdeněk Vršecký a Ing. Pavel Housar.

Anotace

HiL testování – zpracování obrazu

Tato diplomová práce se zabývá implementací systému strojového vidění do klasického HiL testování. V první části jsou teoretické informace o HiL testování, simulátoru μHiL, který je použit pro řešení této úlohy a specifikace testované přístrojové desky. V druhé části je zpracován návrh a realizace testovacího stavu. Dále jsou zde popsány možnosti výběru kamer a softwaru pro realizaci. Třetí část se zabývá samotnou realizací a programováním v dvou softwarových prostředích a to NI Vision Builder for Automated Inspection a testovacím prostředím PROVEtech:TA. V příloze je uvedeno několik ukázkových testů.

Klíčová slova

HiL, CAN, automatická optická inspekce, palubní přístroje, rozpoznání obrazu, kamery, test, PROVEtech, NI Vision Builder

Abstract

HiL testing – image processing

This master thesis deals with implementation system of machine vision in to classical HiL testing. In the first part there is theoretical information about HiL testing, µHiL simulator, which is used for solution of this thesis and specification of tested on-board devices. In the second part there is processed plan and realization of testing state. Furthermore are described ways and means of choices for cameras and software for realization. The third part deals with realization and coding in two software environments. As first is it software NI Vision Builder for Automated Inspection and as second is it testing environments PROVEtech:TA. Some of samples tests are in attachments.

Key words

HiL, CAN, automated optical inspection, on-board devices, image recognition, cameras, test, PROVEtech, NI Vision Builder

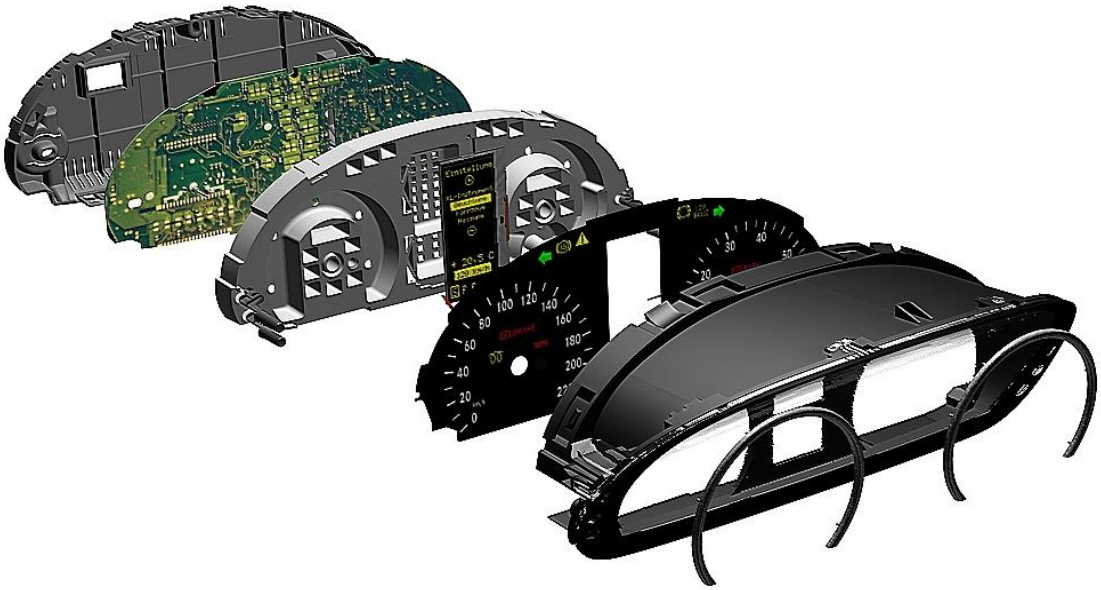
Obsah

Seznam a vysvětlení použitých zkratek

ABS	(Anti-Blocking System) – Systém zamezující zablokování Systém asistence brzdového systému vozu, který pomáhá udržovat podélnou stabilitu a zamezit tak smyku při prudkém brzdění. ABS monitoruje otáčení všech kol a při zablokování dojde k přerušování brzdné síly.
AOI	Automated optical inspection - Automatizovaná optická inspekce Automatizovaná vizuální kontrola, založená na rozpoznání obrazu a detekci jeho částí, jako například detekce výrobku na výrobní lince, měření rozměrů, čtení textu nebo porovnávání barevných vzorů.
CAN	(Controller Area Network) – Síť v oblasti řídících elementů Sběrnice používaná zejména v automobilovém průmyslu, má propracované adresování včetně arbitráže, vysoké zabezpečení dat kódu a díky symetrickému vedení i dobrou odolnost vůči rušení.
ECU	(Electronic Control Unit) – Elektronická řídící jednotka Pojem používaný pro část řídicího systému, který řídí specifickou a vymezenou část (fyzikálního) procesu. Zkratka je také někdy požívána ve významu „Engine Control Unit“ – řídící jednotka motoru.
HIL	(Hardware in the Loop) – Hardware ve smyčce Označení pro řídící jednotku (hardware), která má uzavřenou řídící smyčku svým okolím – buď reálným, nebo simulovaným. Používá se zejména v souvislosti s testováním a simulacemi.
LIN	(Local Interconnect Network) – Lokální propojovací síť Jednoduchá jednodrátová sběrnice určená pro automobilový průmysl, kde se používá vzhledem k nižším nákladům jako podpůrná sběrnice pro sběrnici CAN, např. pro připojení senzorů k řídící jednotce.
MOST	(Media Oriented Systems Transport) – Multimediální sběrnice pro automobil Sběrnice vynalezena firmami BMW a Daimler Chrysler. Slouží k propojení multimediálních aplikací v automobilech, jako například audio, video, navigace a telekomunikačních systémů.
OCR	(Optical Character Recognition) - Optické rozpoznání znaků Metoda, která za pomoci scannerů umožňuje digitalizaci tištěných textů, s nimiž pak lze pracovat jako s normálním počítačovým textem. Text je převáděn buď automaticky, nebo podle předem naučených znaků.
OCV	Optical Character Verification - Optické ověření znaků Softwarový nástroj používaný například pro kontrolu tisku, nebo detekci předem definovaného řetězce znaků. OCV je proces při kterém dochází k ověření čitelnosti znaků zobrazených v oblasti zájmu.
USB	(Universal Serial Bus) – Universální sériová sběrnice Dnes již samozřejmé rozhraní pro připojení nejrůznějších periferií k osobnímu počítači. V současné době nejrychlejší typ USB 3.0

1 Úvod

Tato diplomová práce řeší automatizované testování v automobilovém průmyslu. Jedná se o implementaci strojového vidění zajišťované softwarem NI Vision od společnosti National Instruments do klasického testovacího systému hardware in the loop (HiL), který je více popsán v druhém bodu práce. HiL testování je zajištěno a ovládáno pomocí softwaru PROVEtech:TA od společnosti MBtech Group. Implementace a komunikace těchto dvou softwarů jsou provedeny na principu server-client, přičemž jako server nám posloužil NI Vision a jako klient testovací software PROVEtech:TA.



Obrázek 1-1 Kombi přístroj [1]

Celý systém se skládá z několika částí. Jednou z nich je KOMBI přístroj, který je ovládán přes simulátor µHil softwarem PROVEtech:TA. Zároveň bude snímán digitální kamerou pro vyhodnocení stavu. Jde o KOMBI přístroj používaný ve vozech Mercedes Benz a to konkrétně v užitkových modelech Sprinter. Pomocí kamery systém vyhodnocuje změny, které se na přístroji objevují v průběhu testu. Tím můžeme snadno detekovat, zda je změna žádoucí (tj. odpovídá testovacímu stavu), nebo jde o chybovou zprávu (například rozsvícení kontrolky). Dále je systém schopný detekovat ukazatele a vypočítávat hodnoty, ve kterých se nacházejí. Všechny tyto informace jsou poté odeslány zpět do PROVEtechu. Nedílnou součástí je i software popsaný dále.

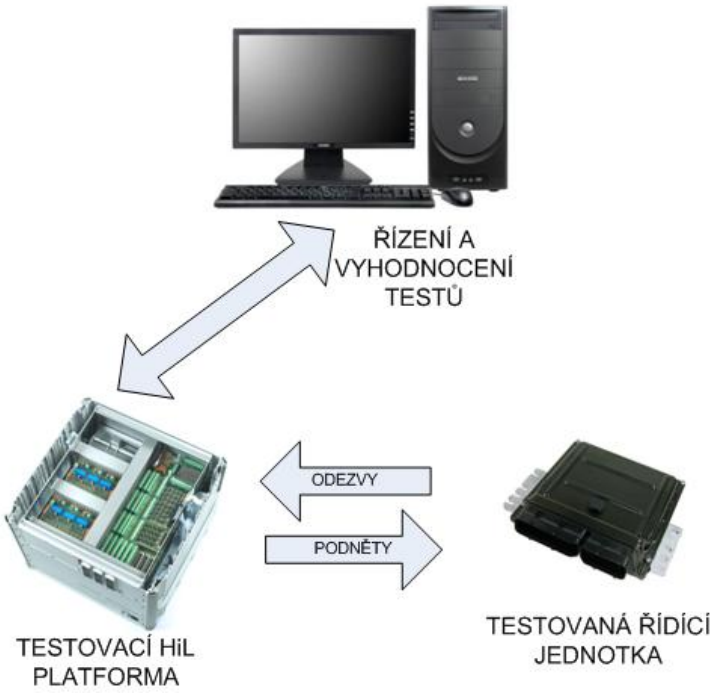
2 HiL testování

2.1 Důvody vzniku a používání HiL testování

V poslední době lze v automobilovém průmyslu pozorovat extrémně rostoucí trend využívání elektronických systémů. Tento trend spolu se zvyšujícím komfortem pro zákazníka s sebou přináší i mnohé problémy. Se zvyšujícím se počtem elektronických systémů v automobilech se zvyšují i požadavky na spolehlivost a také možnost výskytu chyby. To je příčinou, že jsou kladeny enormní nároky na testování elektronických systému už při vývoji v laboratoři. Jenou z nejpoužívanějších metod pro testování je právě takzvaná HiL simulace.

2.2 Princip HiL testování

Metoda HiL testování spočívá v tom, že ke skutečné řídící jednotce je připojen simulátor, který simuluje konkrétní elektromechanické prostředí. Tím může být například motor, podvozek a další části reálného automobilu. Veškerá simulace probíhá v reálném čase a v laboratorních podmínkách. Díky tomu můžeme testovat, jak se bude řídící jednotka chovat ve skutečném automobilu za běžného provozu. Laboratorní podmínky umožňují i testování za extrémních stavů, kterých lze v automobilu dosáhnout jen za komplikovaných podmínek, nebo vůbec. [1]



Obrázek 2-1 Struktura Hil testování

2.3 Princip HiL simulátoru

Systém simulátoru obsahuje několik základních částí. Těmi jsou například procesorové karty pro zajištění komunikace s uživatelským počítačem. To umožňuje sledování a zasahování do průběhu testu. Dále obsahuje jednu či více vstupně výstupních karet, které posílají simulované elektrické signály samotné řídící jednotce. [1]

Podstatnou součástí systému je takzvaný blok pro umělé zátěže. Ten umožňuje k jednotlivým signálům řídící jednotky připojit umělou zátěž. Tím simuluje zapojení čidel a akčních členů. [2]

Samostatnou a neméně podstatnou částí systému je i jednotka simulace chyb. Ta umožní simulovat chyby, které mohou vzniknout na sběrnici. Může signály vyzkratovat nebo rozpojit, propojit je se zemí, s napájecím napětím a různě propojovat samotné signály vzájemně mezi sebou. [2]

2.4 Použití HiL simulátorů

Metoda HiL testování se používá především pro simulaci všeobecných elektromechanických systémů, které jsou ovládány elektronickou řídící jednotkou (ECU). Nejtypičtějším příkladem je řídící jednotka motoru. V tomto systému se reálná motorová řídící jednotka připojí na simulátor, který nahrazuje všechny akční členy a elektrická čidla v reálném motoru. Ten poté simuluje chování skutečného motoru a to v reálném čase. [2]

Velkou výhodou této metody testování je její přehlednost a snadné zajištění opakovatelnosti. Tj. umožňuje provádět opakované testy při zachování identických podmínek. To umožňuje zvýšení kvality prováděných testů, z důvodu snazšího odhalení případných chyb. [2]

Neméně důležitým přínosem HiL testování je odzkoušení řídících jednotek přímo v laboratoři a to ať už během vývoje, tak i při stavbě prototypu. To výrazně snižuje náklady na vývoj elektroniky a šetří i čas pro vývoj potřebný. Veškerý tento proces testování lze velmi snadno automatizovat, což snižuje nároky na obsluhu. [2]

3 Simulátor µHiL

3.1 Obecné

Simulátor µHiL vyvinutý a zkonstruovaný společností MBtech Group je označován jako budoucnost v testování hardware-in-the-loop. Jde o inovaci v oblasti elektroniky a řešení pro motorová vozidla v segmentu inženýrského vývoje. [3]

Obrázek 3-1 Simulátor µHiL [4]

Jde o simulátor, který nabízí nákladově optimální řešení pro funkční testování řídících jednotek, které lze použít ve všech fázích vývoje. Díky jeho kompaktnosti a malým rozměrům (ty jsou až o 80% menší než srovnatelné konvenční HiL simulátory) je velice snadno využitelný jako „desktop simulátor“ už při samotném návrhu a vývoji.

Systém µHiL umožňuje snadné a pohodlné přepínání jednotlivých řídících jednotek na zkušebním systému. Stačí nahrát správný software a konfiguraci ECU. Tento systém je velmi ceněn kvůli snadnému opětovnému použití pro více druhů řídících jednotek. To výrazně snižuje náklady na údržbu řady testovacích systémů a urychluje práci. Tím profituje hlavně zákazník díky kratšímu času potřebnému pro uvedení na trh. [3]

3.2 Části a jejich funkce

Simulátor μHiL je složen z šesti základních bloků. Prvním je napájecí zdroj (POWER-BOX), druhý FPGA-box zajištující komunikaci a až čtyři SC-Boxy.

3.2.1 Napájecí zdroj

Napájecí zdroj slouží k napájení FPGA-boxu a čtyř volitelných SC-boxů. Mimo toho jsou uvnitř logické obvody pro použití při simulaci chyb. Maximální napětí, jaké je zdroj schopný dodávat je U_{max} = 36V a maximální proud je I_{max} = 29A. [4]

3.2.2 FPGA-box

FPGA-box je spolu se zdrojem základní částí pro chod simulátoru. Na jeho přední straně je umístěn displej, na kterém se zobrazuje např. aktuální teplota nebo rychlost ventilátoru. [4]

Hlavní funkcí tohoto bloku je ale zajištění komunikace mezi hostitelským počítačem a právě simulátorem μHiL. Komunikace probíhá přes rozhraní CAN a doby cyklu mohou být až do 20 ms. Kontrola a ovládání je prováděna pomocí softwaru PROVEtech. Na následujícím obrázku 3-2 jsou zobrazeny výstupy a vstupy FPGA-boxu. [4]

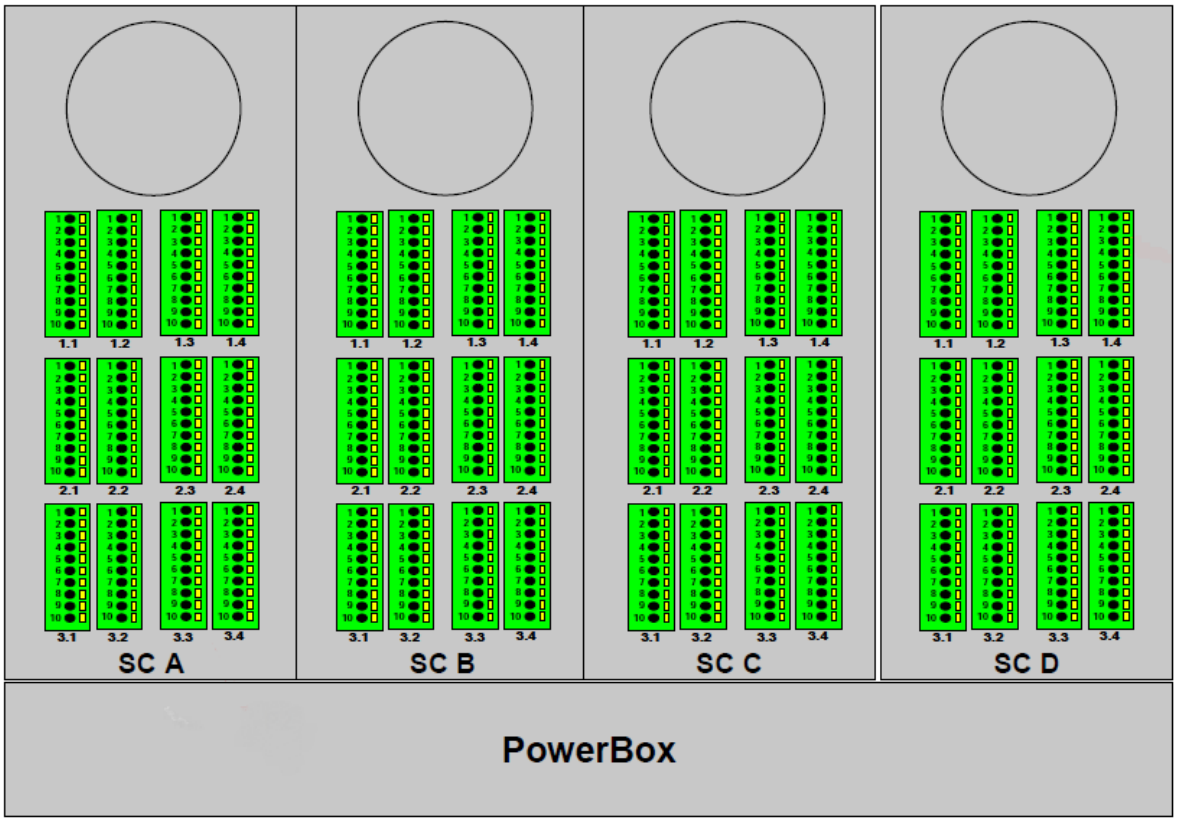
Obrázek 3-2 Externí rozhraní FPGA-boxu [4]

3.2.3 SC-box

Simulátor může obsahovat až čtyři samostatné SC-boxy. Každý z boxů může mít přitom jiné vlastnosti pro konkrétní potřeby daného testu.

Například SC-box typu 0708 umožňuje velmi netypické simulace. Může simulovat 12 článků baterie v napěťovém rozsahu od 0,1V až do 5,0V při jmenovitém proudu každého článku 200mA. Dále má box 4 kanály pro simulaci odporu. Ta se používá v případě potřeby nahrazení odporového čidla automobilu, jakými jsou nejčastěji teplotní čidla. Rozsah těchto hodnot je 1Ω až 70Ω. [4]

Na následujícím zobrazení můžeme vidět blokové sestavení simulátoru. Ve spodní části se nachází napájecí zdroj (Power Box) spolu s FPGA boxem a nad ním čtyři SC-boxy A až D s jednotlivými výstupy a vstupy.



Obrázek 3-3 Blokové schéma simulátoru µHiL [4]

4 Kombi přístroj (palubní přístroje)

4.1 Základní popis

V této části jsou popsané přístroje a kontrolky nacházející se na palubní desce užitkového vozidla Mercedes Sprinter. Vzhled a funkce se v detailech liší v provedení, podle modelu automobilu, typu převodovky nebo motoru a podobně. Princip funkce a hlavní prvky jsou však u všech modelů stejné.

Přístroj se skládá z dvou analogových ručičkových ukazatelů, přičemž ten v levé části zobrazuje rychlost automobilu (tachometr) a druhý otáčky motoru (otáčkoměr). Uprostřed se nacházejí dva LED displeje, umožňující uživateli sledovat jím zvolené údaje, jako je například aktuální spotřeba, průměrná rychlost, ujeté kilometry a další. Mimo toho se zde zobrazují i varovná upozornění, které mohou ovlivnit chod automobilu. Ať už jde o upozornění na nízkou okolní teplotu, nebo nedostatek vody v ostřikovačích.

Ve středu budíků a na horní liště jsou piktogramy, neboli kontrolky k jednotlivým systémům a obvodům automobilu. Ty se vždy po zapnutí zapalování rozsvítí a postupně pozhasínají. Tím je testována jejich funkčnost. Kontrolky by měly vždy zhasínat samostatně a ne několik najednou. Což je v dnešní době při jejich počtu okem téměř nemožné rozpoznat.

Obrázek 4-1 Použitý kombi přístroj [5]

4.2 Topologie sběrnice CAN

V automobilech se v dnešní době používá velké množství řídících jednotek. Ty všechny mezi sebou komunikují po směrnici CAN (Controller Area Network) nebo LIN (Local Interconnect Network). Sběrnice CAN jsou rozděleny do několika typů podle rychlosti přenášení dat. Přístrojová deska je připojena na dvě tyto sběrnice. A to body-CAN a powertrain-CAN. Získává tak informace ze všech řídících jednotek umístěných v automobilu a zároveň zajišťuje propojení informací ze dvou sítí. Pro snazší pochopení je zde uvedena topologie řídících jednotek automobilu.

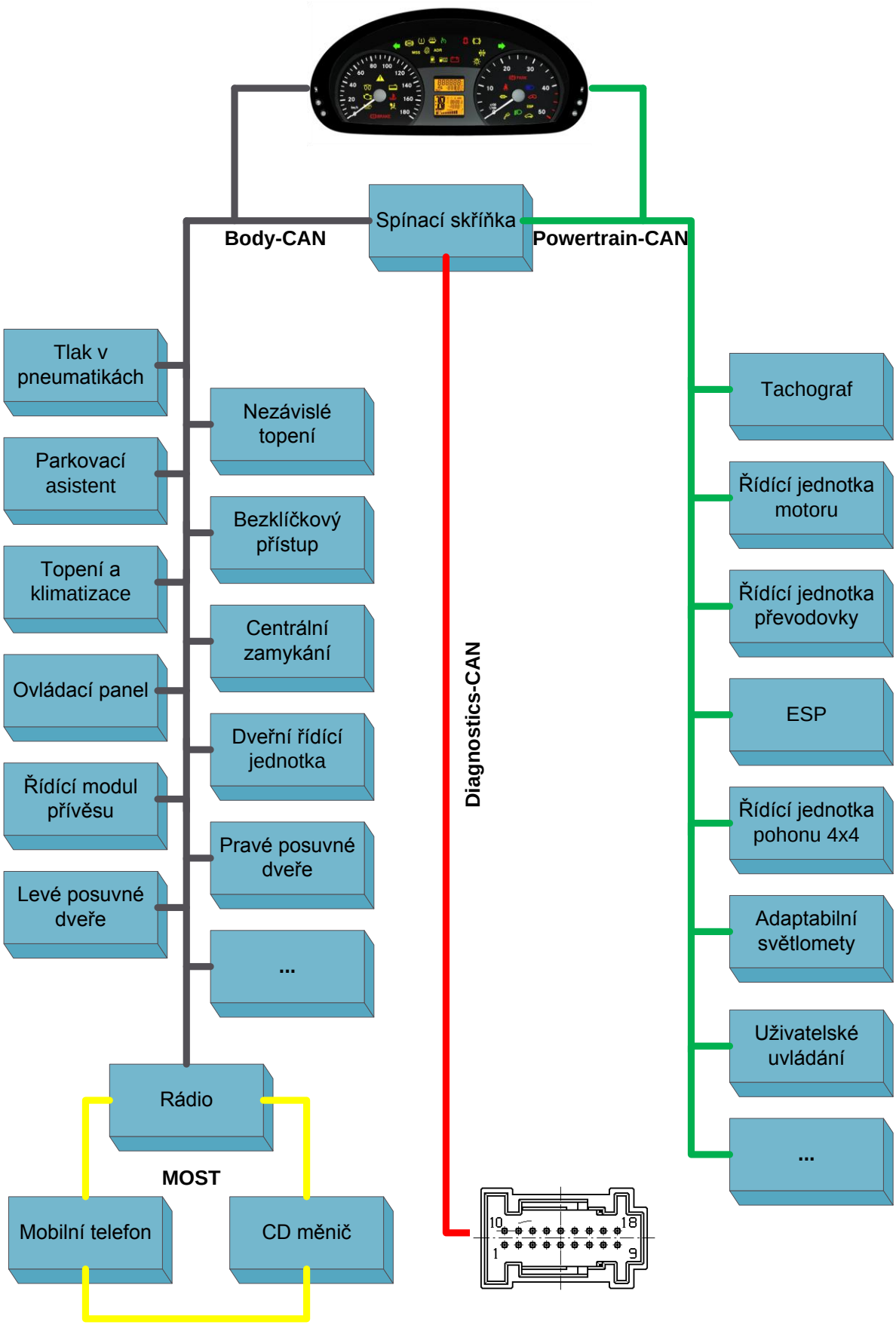
Powertrain-CAN a diagnostics-CAN jsou vlastně totožné sběrnice o stejných parametrech. Jen spolu nejsou nijak propojeny, protože je rozděluje spínací skříňka automobilu. Diagnostics-CAN je vlastně vyvedení sběrnice do uživatelsky přístupného místa a zakončení konektorem. K tomu se v případě potřeby připojuje diagnostika a je tak možno zjistit stav a historii všech řídících jednotek v automobilu.

MOST (Media Oriented Systems Transport) je typ sběrnice vyvinutý speciálně pro přenos velkého toku dat. Je používána u všech multimediálních aplikací v automobilech, jako je veškeré audio, video, navigace nebo třeba telekomunikace. Tento typ sběrnice podporuje i systém plug and play a umožňuje tak připojení až 64 uzlů. Tyto uzly mohou být zapojeny v různých topologiích. Nejčastěji kruh ale i hvězda nebo běžný řetězec. [6]

Přehled sběrnic

NÁZEV	RYCHLOST	TŘÍDA
Powertrain-CAN	500 kBit/s	CAN třídy C
Diagnostics-CAN	500 kBit/s	CAN třídy C
Body-CAN	83,33 kBit/s	CAN třídy B
MOST	21 Mbit/s	

Topologie řídících jednotek automobilu



4.3 Části kombi přístroje

4.3.1 Tachometr

Tachometr je levý přístroj na přístrojové desce. Jde o jednoduchý ručičkový ukazatel s kruhovou stupnicí v km/h. O pohyb ukazatele se stará krokový motor, který na základě signálu z řídící jednotky nastaví příslušnou hodnotu.

Signál o aktuální rychlosti automobilu je přiváděn do přístroje po sběrnici CAN z řídící jednotky systému ABS nebo ESP. Ta získává signál ze senzorů rychlosti otáčení jednotlivých kol. Signál musí jít vždy z více na sobě nezávislých senzorů. Z každého kola snímá jeden signál a ty pak vyhodnocuje. Je-li rychlost jednoho kola chybová v případě vadného snímače, určuje se rychlost z hodnot ostatních senzorů po určitou dobu. Pokud ani po uplynutí této doby nemá řídící jednotka od senzoru žádnou informaci, pak ukazatel rychloměru ukáže na nulu (0km/h ; 0mph). Způsob určování rychlosti z jednotlivých kol respektive jejich srovnání je pro snazší pochopení znázorněn na následujícím diagramu. [5]

Podstatnou součástí tachometru je správná kalibrace ukazatele, tak aby zobrazované hodnoty vždy odpovídaly skutečnosti s povolenou tolerancí. Tato tolerance je v každé zemi stanovena jinak, jak uvádí přehled několika vybraných zemí v následující tabulce.

Země	Rychlost	Tolerance
EU	> 40 km/h	+ 10%
		- 10%
Japonsko	> 35 km/h	+15%
		-10%
Mexiko	> 0 km/h	+ 5%
		- 5%
USA	= 50mph	+ 10%
		- 10%

Tabulka 4-1 Rychlostní tolerance ve vybraných zemích [5]

Vysvětlení zkratek použitých v diagramu určování rychlosti

FFFFh – chybový signál vysílaný v případě vadného senzoru

PL – přední levé kolo

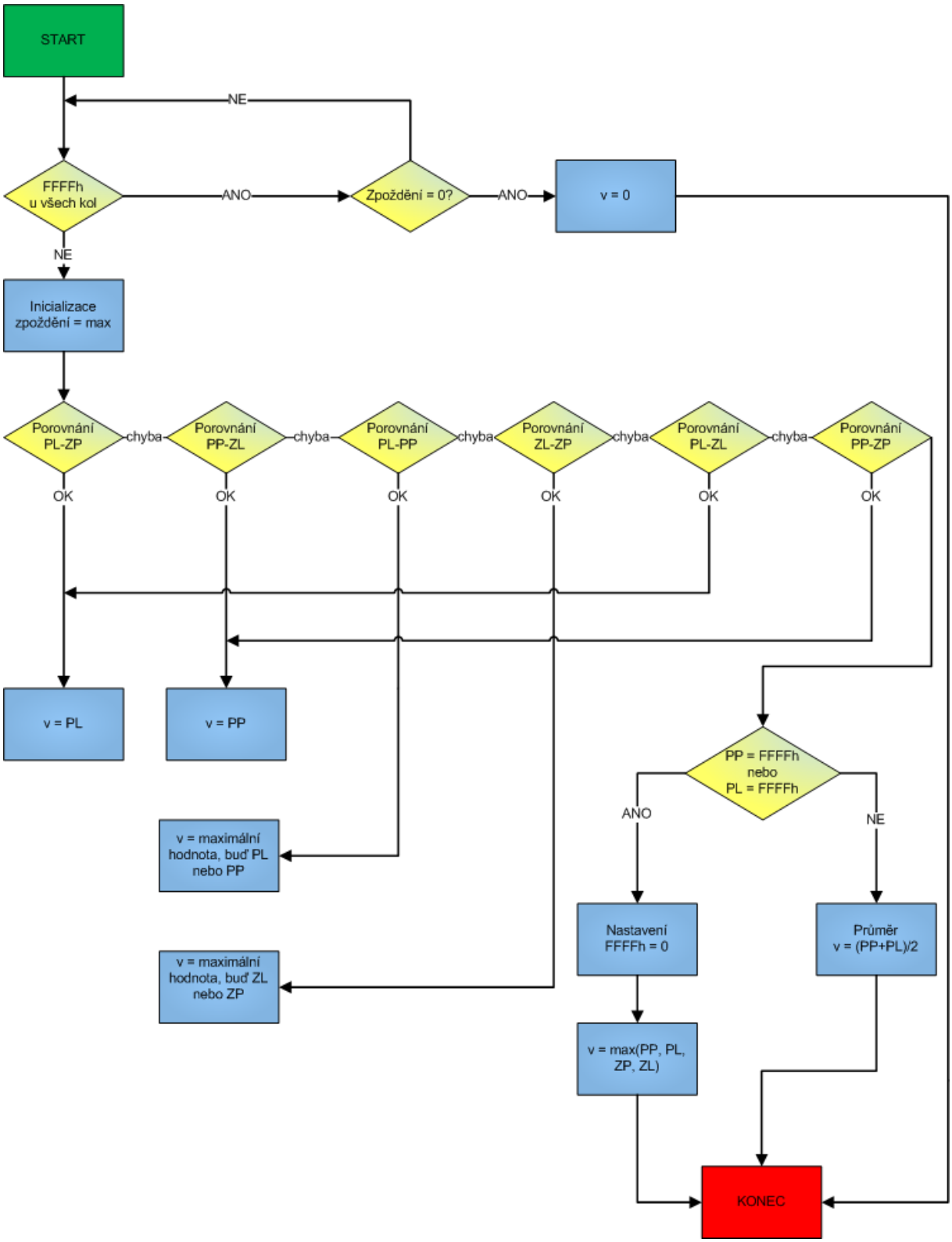
PP – přední pravé kolo

ZL – zadní levé kolo

ZP – zadní pravé kolo

v – rychlost

Diagram určování rychlosti [6]



4.3.2 Otáčkoměr

Otáčkoměr je umístěn na pravé straně kombi přístroje a stejně jako u tachometru se o pohyb ukazatele stará krokový motor. Aby se zabránilo kolísání ukazatele za určitých provozních podmínek, je zde implementováno i tlumení s určitou setrvačností a filtrace signálu. Samotný signál může být vztažen k hodnotě rychlosti zobrazované na tachometru. To zda je hodnota vztažena k signálu tachometru, nebo ne záleží na nastavení podmínky a takzvaného povolovacího bitu. Jestliže má bit hodnotu 1 a skutečná rychlost je menší než požadovaná, pak se nastavuje podle rychlosti. Jinak vždy zobrazuje skutečné otáčky. Naopak jestliže, má povolovací bit hodnotu 0 pak vždy zobrazí skutečné otáčky motoru bez ohledu na jakékoliv podmínky. [5]

4.3.3 Multifunkční displej

Displej je rozdělen na dvě části. Horní větší část zobrazuje za normálního stavu uživatelem zvolené informace, jako jsou například ujetá vzdálenost, aktuální nebo průměrná spotřeba, čas strávený na cestě, celkový počet ujetých kilometrů a další. V případě závady jako třeba nedostatku provozní kapaliny, vadných brzd nebo prasklé žárovky se na displeji zobrazí varovné hlášení a textová zpráva pro uživatele. Tyto varovná hlášení jsou rozdělena do tří skupin podle priorit. Nejvyšší prioritu mají vážné závady jako třeba nedostatek oleje v motoru, přehřívání chladící soustavy a tak dále. Střední prioritu mají méně závažné závady neomezující provoz automobilu jako je například prasklá žárovka, nebo upozornění na čas servisní kontroly. Nejmenší prioritu mají hlášení omezující pouze komfort uživatele, jako třeba nedostatek kapaliny v ostřikovačích. [5]

Technická data displeje	
Podsvícení	LED žluté 589nm (červené 630nm)
Rozlišení	64x156
Velikost bodu	0,53x0,53mm
Rozestup bodů	0,03mm

Tabulka 4-2 Technická data displeje [5]

Obrázek 4-2 Displej [5]

Spodní, menší část displeje je standardně nastavena na zobrazování aktuálního času, venkovní teploty vzduchu a v případě automatické převodovky informuje o stavu, ve kterém se nachází. V případě potřeby si může uživatel nastavit tento displej na zobrazení aktuální rychlosti automobilu.

4.3.4 Kontrolky

Kontrolky neboli takzvané piktogramy jsou poslední podstatnou vizuální částí přístroje. Každá kontrolka má svůj přesně definovaný tvar, velikost, barvu a prostor, ve kterém se nachází. V následujícím seznamu je přehled základních kontrolek a uveden jejich význam a barva.

Symbol	Barva	Význam
	modrá	Dálková světla
	červená	Bezpečnostní pás
	červená	Kontrola dobíjení
	červená	Zámek řízení
	zelená	Levé směrové světlo
	žlutá	Kontrola trakce
	žlutá	ABS
	žlutá	Kontrolka motoru
	žlutá	Hladina oleje
	červená	Teplota chladící kapaliny

Tabulka 4-3 Přehled základních kontrolek [5]

5 Návrh a realizace testovacího stavu kombi přístroje

5.1 Návrh testovacího stavu

Návrh testovacího stavu kombi přístroje se věnuje potřebnému hardwarovému a softwarovému vybavení, využitého pro realizaci. Požadavkem pro realizaci je, samostatně a plně ovládaný kombi přístroj doplněný o nezávislý vizuální inspekční systém s kamerou. Z topologie v kapitole 4.2 je patrné, že pro ovládání musíme nahradit veškeré řídící jednotky (resp. jejich signály), tak aby byla zajištěna plná funkčnost. Simulaci signálů řídících jednotek automobilu zajišťuje v tomto případě simulátor µHiL (popsaný v kapitole 3) a software PROVEtech:TA, kterému se věnuji v kapitole 7.2 Dále je k realizaci potřebná komunikační karta CANcaseXL (v tomto případě dvě) a napájecí zdroj. Vizuální inspekční systém je složen z kamery a osobního počítače se softwarem NI Vision Builder AI (popsaný v kapitole 7.1)

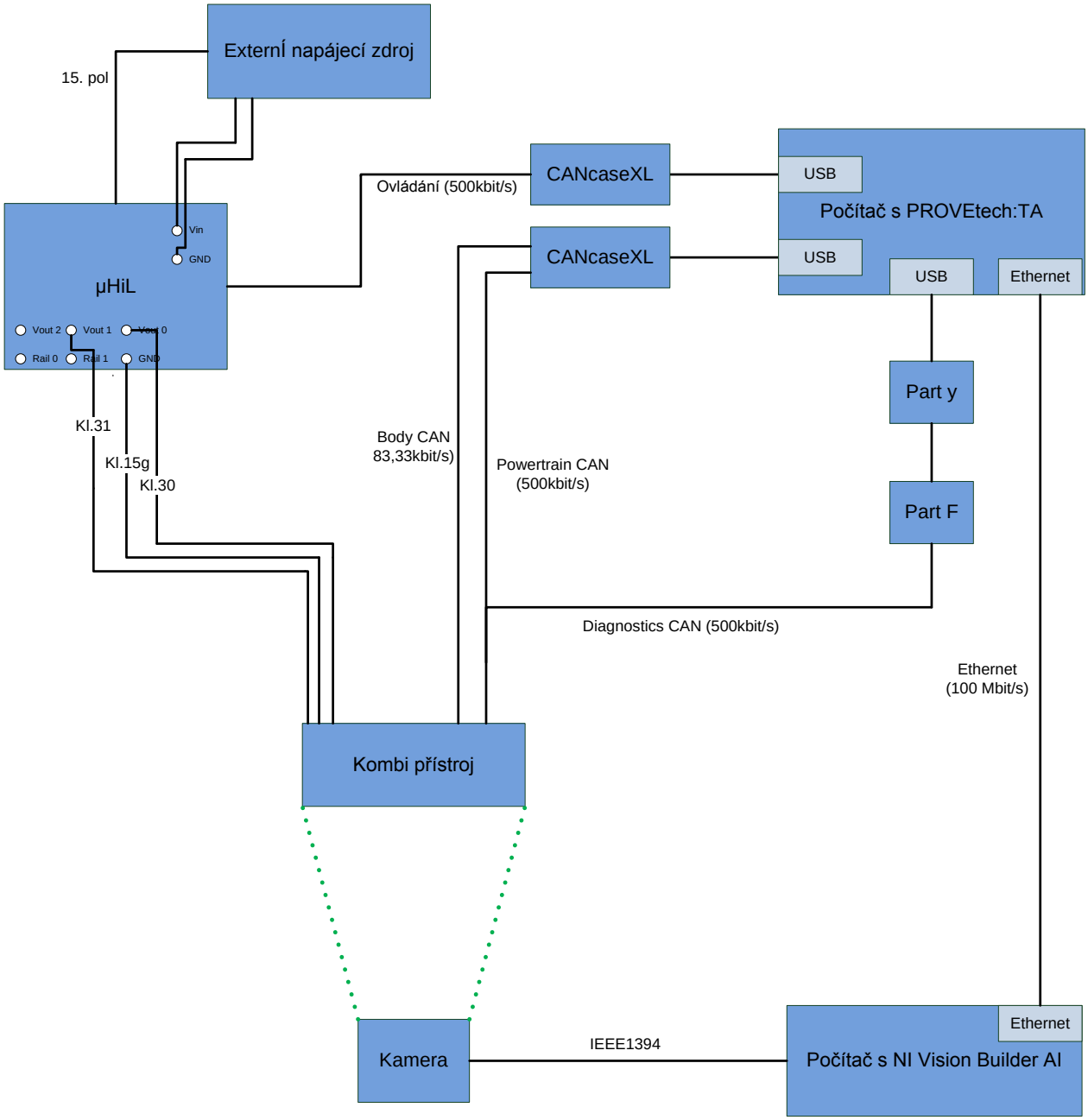
Napájecí zdroj je řízený přes simulátor µHiL pomocí softwaru PROVEtech:TA. Zdroj a simulátor jsou spolu propojeny přes patnácti pólové konektory a ovládání se provádí analogově. Každý pól v konektoru má tedy svojí vlastní funkci. Například nastavování napětí nebo proudu, zpětná informace o nastavených hodnotách a další. Funkce zdroje v tomto případě spočívá v nahrazení klasické autobaterie pro napájení kombi přístroje.

Komunikační karta CANcaseXL zajišťuje čtení a posílání zpráv na CAN sběrnici. Díky rozhraní USB ji lze snadno připojit k běžnému počítači. Obsahuje dva kanály pro připojení na sběrnici CAN. Jak je patrné z topologie v kapitole 4.2 kombi přístroj je připojen právě na dvě tyto sběrnice (powertrain CAN a body CAN). Tím jsou obsazeny oba kanály. Vzhledem k tomu, že pro komunikaci mezi počítačem a simulátorem potřebujeme další kanál, musejí zde být zapojeny dvě karty (jeden kanál na druhé kartě je nevyužitý).

Vizuální a inspekční část tohoto systému se skládá z kamery (viz 6.1.3.) a inspekčního softwaru (viz 7.1). Ten je nainstalován na samostatném počítači a je propojen s kamerou přes rozhraní FireWire IEEE 1394b. Smyčka celého systému je uzavřena síťovým propojením obou počítačů.

5.2 Realizace testovacího stavu

Následující blokové schéma ukazuje, jak jsou vzájemně propojeny prvky systému popsané v předchozím bodu. Pro úplnost je zde zahrnuta i část diagnostiky, která není v tomto případě nijak využívána.



Obrázek 5-1 Blokové schéma realizace testovacího stavu

6 Výběr hardwaru a softwaru pro realizaci

6.1 Přehled možných kamer pro rozpoznání obrazu

Výběr vhodné kamery pro snímání a rozpoznání obrazu, je velmi individuální částí návrhu aplikace. Podle konkrétních požadavků na náročnost systému, kvalitu obrazu, tvar a velikost detekovaných objektů, můžeme použít celou řadu kamer. A to už od obyčejných webkamer připojených pomocí USB, přes průmyslové kamery, až po tzv. inteligentní kamery. Vše záleží pouze na náročnosti konkrétní aplikace, pro kterou budeme kameru vybírat. V průmyslových aplikacích lze použitelné kamery rozdělit z hlediska provedení do dvou částí. První možností je samostatná průmyslová kamera a druhou pak tzv. inteligentní kamera (Smart Camera). Z hlediska typu signálu pak rozdělujeme kamery na digitální a analogové. Třetím podstatným kritériem, podle kterého můžeme rozdělit kamery do skupin je komunikační rozhraní. Pro každou aplikaci jsou vhodná jiná rozhraní a tím i jiné datové toky. Pro účely této práce je využívána digitální kamera s výstupem FireWire a to konkrétně IEEE 1394b. Tato část porovnává digitální kamery z hlediska konstrukce jako „klasické“ a „inteligentních“ a z hlediska typu komunikačního rozhraní. [7,8]

Obrázek 6-1 Průmyslové kamery Ace s připojením na ethernet [7]

6.1.1 Plošné (maticové) kamery

Tento typ kamer je pro svoji flexibilitu a všestrannost nejpoužívanějším typem v oblasti strojového vidění. Jedná se o klasické zařízení, jako snímač je zde používán CCD nebo CMOS senzor a je dostupné v mnoha variantách provedení. Co se týče rozlišení, tak tento typ kamer umožňuje velmi širokou škálu možností. Od velmi jednoduchých výrobků s minimálním rozlišením, až po špičkové nástroje s opravdu vysokým rozlišením.

Další podstatnou částí je rozhraní, přes které bude komunikovat. V podstatě jsou čtyři základní typy a to rozhraní FireWire, Gigabit Ethernet, Cameralink a USB. Jednotlivé typy rozhraní se svými vlastnostmi a možnostmi použití jsou rozepsány dále v odstavci 6.1.3

6.1.2 Inteligentní kamery (Smart Cameras)

Novým trendem v této oblasti je bezesporu využití chytrých kamer. Jde o zcela kompaktní systém umožňující řešit poměrně snadno libovolné úlohy. Kvalitní kamera je doplněna o výkonný jednočipový mikropočítač, který funguje jako vyhodnocovací jednotka. To umožňuje přímé propojení a komunikaci s programovatelnými automaty PLC, čehož se využívá především na výrobních linkách. [7,8]

V současnosti jsou k dispozici například kamery od firmy National Instruments, která má velmi bohaté zkušenosti v tomto odvětví. Nejnovější řada inteligentních kamer od této společnosti disponuje procesorem Intel s frekvencí 1,6 GHz, operační pamětí 512MB a Flash pamětí až 2GB. Díky dobrému mechanickému provedení má tato řada krytí IP67, což zaručuje naprosto bezproblémový provoz za téměř jakýchkoliv podmínek a ochranu jak proti prachu, tak i proti vodě. Tím je zajištěna široká možnost využití především v průmyslu. [8]

Jako komunikační rozhraní je již téměř vždy v tomto segmentu použito ethernetu. Kameru jednoduše připojíme k místní síti a můžeme se na ní kdykoliv z jakéhokoliv počítače podívat, upravit její parametry, vyhodnocovací algoritmy, nebo nahrát nový firmware. Z uživatelského hlediska je tento způsob nejkomfortnějším. [7]

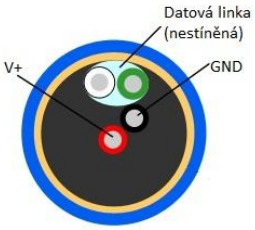
6.1.3 Přehled typů komunikačních rozhraní kamer

USB 3.0 (Universal Serial Bus)

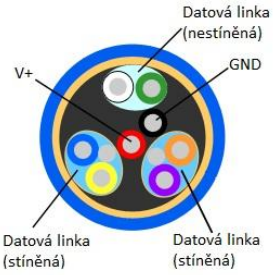
Sériové komunikační rozhraní USB je již řadu let součástí každého počítače. Jeho využití v oblasti průmyslových kamer ale dlouho bránila jeho menší přenosová rychlost. S nástupem USB verze 3.0 se ale toto rozhraní stalo běžnou součástí špičkových kamer. Tento typ připojení by mohl v budoucnu zcela nahradit raozhraní FireWire. USB 3.0 je nástupcem standadu USB 2.0 a uvádí se, že má až desetkrát vyšší propustnost dat. Pod textem je uvedena tabulka přenosových rychlostí všech generací sběrnice USB. Vedle tabulky jsou pak uvedeny dva obrázky řezu kabelu typu USB 2.0 a USB 3.0. [9]

Verze	Označení	Rychlost
1.0	Low Speed (LS)	1,5 Mb/s
1.1	Full Speed (FS)	12 Mb/s
2.0	High Speed (HS)	480 Mb/s
3.0	Super Speed (SS)	4,8 Gb/s

Tabulka 6-1 Přenosové rychlosti sběrnice USB [9]



Obrázek 6-3 Řez kabelu typu USB 2.0 [9]



Obrázek 6-2 Řez kabelu typu USB 3.0 [9]

Gigabit Ethernet (GigE)

Gigabit Ethernetové kamery nemají oproti ostatním typům komunikačního rozhraní problémy s délkou kabelu pro přenos dat. Možná délka kabelu je v tomto případě jako u klasického ethernetu až 100m. Díky možnosti připojení do sítě, tím dostává kamera zcela jiné možnosti. Je schopna využívat všech výhod síťové infrastruktury a tím se řadí spíše do segmentu použitelnosti v průmyslu, kde potřebujeme připojit velké množství kamer na jedinou síťovou kartu. Mezi nevýhody tohoto připojení patří poměrně větší zatížení procesoru než u ostatních typů a delší odezva při spouštění expozice softwarovým triggerem. Napájení kamery tohoto typu musí být zajištěno externě. Jedinou výjimkou jsou kamery, které využívají PoE (Power over Ethernet) technologii. [10]

FireWire IEEE 1394

FireWire rozhraní je velmi spolehlivé a odolné vůči výpadkům komunikace. Umožňuje snadné připojení k uživatelskému počítači, bez nutnosti jakéhokoliv dalšího napájení. To dodává kameře velkou možnost mobility a využití v nejrůznějších podmínkách. Výhodou tohoto připojení je schopnost rychle přenášet data o velkých objemech. Nevýhodou oproti tomu je velmi omezená vzdálenost kamery od počítače z důvodu délky kabelu (řádově jednotky metrů). Proto je vhodnější pro laboratorní nebo lékařské účely. V současné době existují u kamer dva standardy a to IEEE 1394a a IEEE 1394b. [10]

IEEE 1394a (FireWire 400)

Zařízení pracující na tomto rozhraní komunikuje rychlostí 400Mb/s. Používá se šesti a u některých periferií i čtyř pinový konektor (ten neumožňuje napájení). Tento typ rozhraní je dnes spíše u starších typů kamer, nebo se používá pro připojení externích disků a osobních videokamer. U nových kamer je stále častěji rychlejší FireWire 800. [10]

IEEE 1394b (FireWire 800)

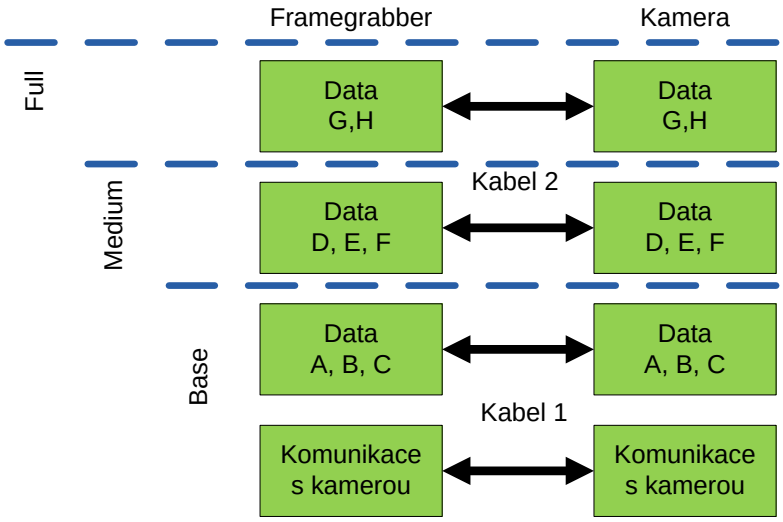
Kamera připojená přes toto rozhraní komunikuje rychlostí 800 Mb/s a je připojena přes devíti pinový konektor. Sběrnice je kompatibilní s předchozím standardem, takže je možné oba typy kombinovat. Velkou výhodou tohoto rozhraní je například oproti gigabitovému ethernetu nízké zatěžování procesoru, které je méně než 1%. Kamera připojená přes GigE přitom vytěžuje procesor na 5 až 10%. Kamera s rozhraním FireWire 800 je použita i pro realizaci této diplomové práce. [10]

Camera Link

Camera Link je standardizované paralelní rozhraní pro komunikaci s digitální kamerou. Jedná se o velmi spolehlivou sběrnici, která je schopna přenášet data rychlostí až 655 MB/s. Tím je sběrnice Camera Link předurčena pro použití u výkonných kamer, které snímají obraz vysokou rychlostí (v řádech stovek snímků za sekundu), nebo pro kamery s opravdu velkým rozlišením. Naopak nevýhodou Camera Linku je nutnost použití speciální karty (tzv. framegrabber) do počítače a omezená délka kabelů, podobně jako u FireWire okolo 10m. [11]

Rozhraní Camera Link vychází z vysokorychlostního rozhraní Channel Link. To je složeno ze čtyř párů vodičů pro přenos dat a jednoho páru pro časový signál. Páry pro přenos dat a pro časový signál mají každý svůj vlastní konektor. Přenosová kapacita tohoto rozhraní je 255 MB/s. [11]

Z toho vznikl standard Camera Link, který pro komunikaci využívá 26-ti pinový MDR konektor. Ten sdružuje všechny vodiče dohromady a přenosová rychlost zůstává 255 MB/s. Tato konfigurace se označuje jako Camera Link Base. Pro vyšší kapacity se musí použít rozšířená konfigurace Medium (510 MB/s) nebo Full (680 MB/s). V tomto případě se paralelně připojí další kanály Channel Link. Vše je znázorněno v topologii pod textem. [11]



Obrázek 6-4 Topologie komunikačního rozhraní Camera Link [11]

6.1.4 Použitá kamera

Basler Scout scA 1000-30fc (IEEE 1394b)

Popis produktu

Jde o kameru z řady Basler Scout, která je určena především pro aplikace strojového vidění. Vlastnosti kamery zajišťují nízký šum a velmi dobrou citlivost. Díky tomu je kameru možné použít i v medicínských a laboratorních aplikacích.

Kamera disponuje rozhraním IEEE1394b (FireWire 800). Nevýhodou je, že kameru lze tedy k počítači připojit pouze přes toto rozhraní. Oproti tomu výhodou je jak samotná sběrnice IEEE1394b, která patří k nejstabilnějším v oblasti komunikačních rozhraní u průmyslových kamer, tak i potřeba jediného kabelu pro připojení, který zajišťuje tok informací a zároveň napájení samotné kamery. [12]

Kamera podporuje standardy GenICam a DCAM API, takže je možné ji použít ve většině programů pro záznam a zpracování obrazu. Kameru je také možno použít s programy, které podporují rozhraní DirectShow anebo TWAIN. [12]

Tento typ kamery se vyrábí ve dvou konstrukčně odlišných provedeních. Jednou z nich je přímá varianta a druhou pravoúhlá (viz obrázek 6-4). Standardní součástí kamery je závit pro objektiv typu C. [12]

Obrázek 6-5 Kamera Basler Scout scA1000-30fc (IEEE1394b) [12]

Parametry a vlastnosti kamery

Nastavitelné vlastnosti

- nastavení vyvážení bílé
- volně nastavitelná oblast zájmu
- nastavitelná doba expozice a snímací frekvence
- gama korekce
- binning (spojování pixelů pro zvýšení citlivosti)
- bitová hloubka 8, 10 nebo 12-bitů
- automatické zesílení a doba expozice
- interní nebo externí synchronizace

Parametry

Parametry Kamery Blaser Scout scA1000-30FC	
Výrobní řada	Basler Scout (IEEE1394)
Druh senzoru	Progressive Scan CCD
Senzor	Sony ICX204
Velikost senzoru	1/3 "
Barva	Color
Rozlišení (V x H)	1034 x 779 pixelů
Velikost pixelu	4,65 x 4,65 mikrometrů
Snímací frekvence	30 fps
Komunikační rozhraní	IEEE1394b
Napájení	8 - 30 VDC (z kabelu IEEE1394)
Rozměry	přímá: 85,5 x 44 x 29 mm, pravoúhlá: 97 x 44 x 41,8 mm (vč. závitu a konektorů)
Hmotnost	0,2 kg

Tabulka 6-2 Parametry kamery Basler Scout scA1000-30fc (IEEE1394b) [12]

6.2 Software

V posledních letech se začal software pro strojové vidění velmi měnit. Ve své prvotní podobě měl největší využití v oblasti osazování součástek a výroby desek plošných spojů. Tento způsob kontroly však začal být se vzrůstající hustotou součástek a zmenšujícími se rozměry velmi nespolehlivý a nepřesný. Oproti tomu otevřel cestu kontrolám na výrobních linkách. V další fázi se tento systém rozšířil až do dnešní podoby, kdy je využíván téměř v každé fázi výroby a testování.

Jelikož vytvořit základní software pro inspekci obrazu není nijak obtížný proces, velké množství firem si vyvíjí vlastí. V případě, že jde o jednoduché porovnávání obrazu ve smyslu shoda-neshoda je vývoj levnější. Proto existuje velké množství „malých“ softwarů od různých firem. V tomto případě ale není vždy zajištěna kompatibilita systému a inženýrská podpora.

Pro složitější aplikace a širší využití existuje několik opravdu špičkových produktů. U nich je zaručená plná kompatibilita s většinou pracovních systémů, velmi dobrá inženýrská podpora a téměř neomezená možnost rozšíření. S ohledem na tyto aspekty je zřejmá i vyšší pořizovací cena.

6.2.1 COGNEX - VisionPro

Jedním ze softwarů pro náročné aplikace je VisionPro od společnosti COGNEX. Zde společnost využila všech svých zkušeností z více než dvacetileté praxe. Jedná se o detailně propracovaný nástroj s řadou funkcí pro snadnou inspekci. Obsahuje velké množství nejrůznějších knihoven, je schopen pracovat i s netradičními zdroji obrazu, jako například 3D kamerou, termokamerou, nebo rentgenovou kamerou. Software je plně kompatibilní s několika druhy operačních systémů (Linux, Microsoft 32-bit a 64-bit). [13]

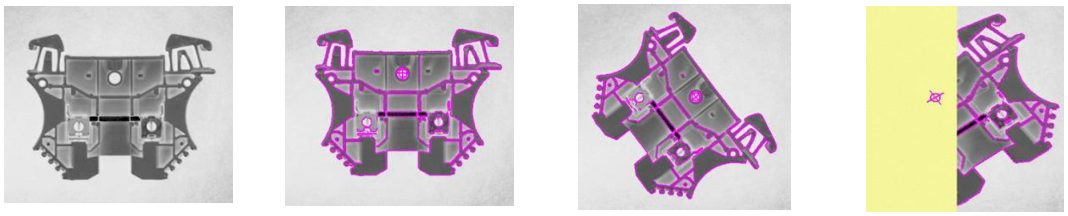
Nástroje VisionPRO

V této části je popsáno pár základních možností inspekce softwaru VisionPRO.

Geometrické porovnání se vzorem

PatMax SA, PatMax XLC, PatFlex Syntetic, PatMax, PatInspect

V tomto kroku se porovnává snímaný obraz se vzorem. Díky možnostem nastavení tohoto kroku, lze rozpoznat shodu i za změněných podmínek, jak je patrné z obrázků pod textem. [13]



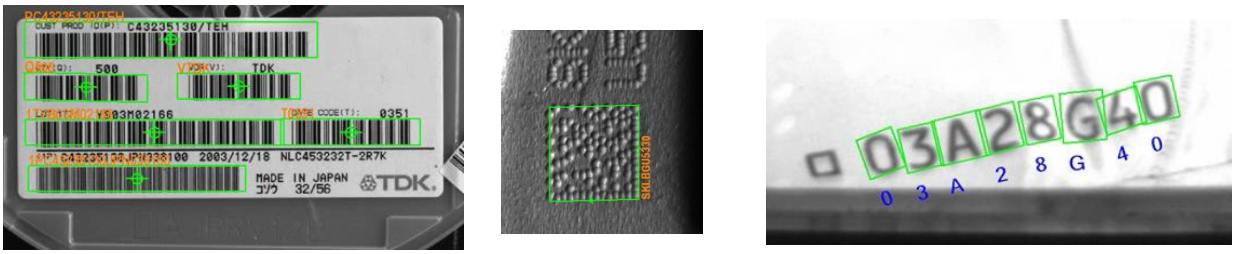
Obrázek 6-6 Ukázky porovnání výrobku se vzorem a nalezení shody [13]

Ukázka možností identifikace objektu za různých podmínek. Na prvním obrázku je vzorový výrobek, který slouží k porovnávání s ostatními. Na druhém je nalezený stejný objekt. Na třetím obrázku je nalezení shody i v případě natočení a na posledním obrázku lze vidět nalezení shody i v případě pouze částečného zobrazení objektu.

Identifikace a verifikace

IDMax, OCVMax, Barcode Reading, 2D Symbol Verification

V tomto kroku jsou možnosti jak číst a identifikovat 1D a 2D kódy a to i částečně poškozené, nebo klasické čtení textu OCR a OCV. To vše poskytuje poměrně rychlou odezvu při identifikaci. Pod textem jsou umístěny některé příklady z tohoto kroku. [13]



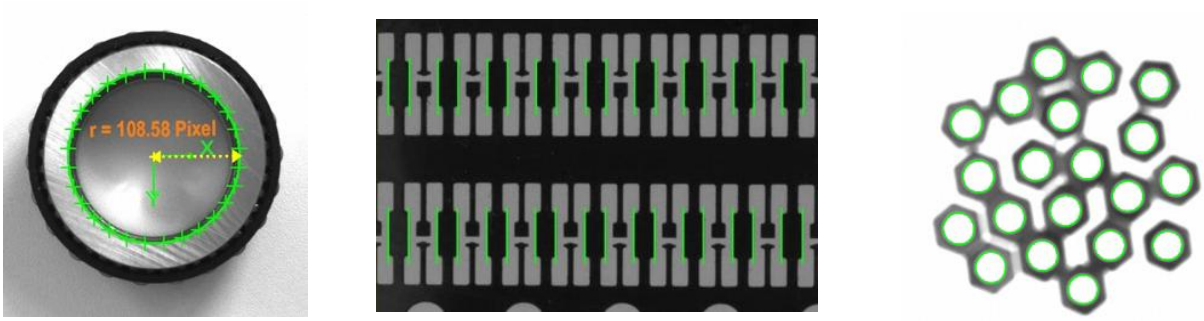
Obrázek 6-7 Ukázky identifikace a verifikace [13]

Na těchto obrázcích můžeme vidět detekci klasického čárového kódu (1D). Na druhém obrázku je detekce a dekódování 2D kódu a na posledním čtení textu OCR.

Měřící nástroje

Line Finder, Circle Finder, Blob Analysis, Fixturing, Calibration, Caliper, Find Geometry

Umožňují měřit a analyzovat geometrické aspekty obrazu (tzv. bezkontaktní měření), vyhledávat hrany v určitém směru a měřit rozestupy. [13]



Obrázek 6-8 Příklady měření a detekce [13]

Na prvním obrázku je zobrazena možnost bezkontaktního měření. V tomto případě se jedná o měření vnitřního průměru ložiska. Na prostředním obrázku můžeme vidět detekci hran v příčném směru a na posledním je zobrazeno větší množství dílů, kde se u každého porovnává vnitřní průměr se vzorem.

6.2.2 Control Web - VisionLab

Software Control Web se systémem pro strojové vidění VisionLab, je jedním z cenově dostupnějších softwarů. Je navržen pro použití jak v rozsáhlých aplikacích, tak i v malých vestavěných systémech. VisionLab je vlastně doplňkem softwaru Control Web a zajišťuje tak systémovou část strojového vidění.

Díky tomu, že tento systém je stále vyvíjen a společnost poskytuje veškerá vylepšení pro své zákazníky zdarma, je tento systém ideální spíše pro menší firmy. Všechny aplikace systému Control Web (např. VisionLab) umožňuje komunikaci prostřednictvím Ethernetu, USB, RS 232, 422, 485, Wi-Fi, Bluetooth, může obsahovat internetový HTTP server, ale současně má k dispozici také webového klienta, který dokáže posílat e-maily, posílat a přijímat SMS zprávy, komunikovat přes GPRS nebo radiové mosty, spolupracovat s Plug-and-Play zařízeními na rychlé USB a spolupracovat s SQL databázemi. [14]

Nástroje VisionLab

Pracovní nástroje tohoto systému se ve větší míře shodují s funkcí konkurenčních softwarů. Obsahuje klasické čtení textu OCR nebo OCV, porovnávání tvarů se vzorem podle barevné nebo geometrické shody, detekci hran a měření vzdáleností, čtení 1D a 2D kódů a mnoho dalších. [14]

Použití VisionLab

Čtení registračních značek automobilů

Díky tomu, že je vyvíjen ve spolupráci s uživateli, má systém nově i několik velmi netradičních funkcí. Jednou z nich je třeba i automatické čtení registračních značek vozidel v jediném inspekčním kroku. Toho se využívá v případě menších aplikací, jako jsou terminály u vjezdů do parkovacích domů, nebo hlídaných parkovišť. Ve větších aplikacích jej pak používají některá města pro kontrolu odcizených vozidel při projíždění městem. [14]

Obrázek 6-9 Ukázka automatického přečtení registrační značky [14]

Samotné čtení RZ není zcela triviální úlohou. U běžného strojového vidění máme většinou předem definované a téměř stálé podmínky, jako je třeba osvětlení, odrazivost, rychlost pohybu a s tím související i kvalita pořízeného obrazu. S tím vším se musí systém vyrovnat, aby pracoval bezchybně.

Největší využití tohoto systému je však v běžném výrobním procesu, kde je využíván například k přesnému měření vyráběných produktů, kontrole geometrické přesnosti, odpovídající barvě, rozřazování součástek podle čárových kódů a mnohé další.

6.2.3 NI Vision Builder for Automated Inspection

Dalším vhodným softwarem pro aplikace strojového vidění je Vision Builder, který je detailněji popsán v kapitole 7.1.

6.2.4 Porovnání popsaných softwarů

Název SW	NI Vision Builder for AI	COGNEX VisionPRO	Control Web - VisionLab
Jazyk	ENG, GER, FR, ESP	ENG, GER, FR	CZE, ENG
Požadavky na RAM (min.)	256MB	512MB	256MB
Požadavky na OS	Microsoft, Linux (částečně)	Microsoft, Linux	Microsoft
Geometrické porovnání	ANO	ANO	ANO
Porovnání se vzorem	ANO	ANO	ANO
Detekce hran	ANO	ANO	ANO
Detekce kruhových oblastí	ANO	ANO	NE
Detekce objektů	ANO	ANO	ANO
OCR	ANO	ANO	ANO
OCV	ANO	ANO	ANO
Úprava obrazu	ANO	částečně	NE
Komunikace ethernet	ANO	ANO	ANO
Přístup k GSM bráně, e-mail klient	NE	NE	ANO
Možnost matematických operací	ANO	částečně	NE
Úroveň technické podpory	Velmi vysoká	Vysoká	Průměrná
Cena	2700$	2000$	1100 $

Tabulka 6-3 Porovnání softwarů pro automatickou optickou inspekci [13,14,15]

7 Použitý software

Pro řešení úlohy jsem využil dvou softwarů. Ovládání a testování kombi přístroje je realizováno v testovacím softwaru od společnosti MBtech Group, **PROVEtech:TA**. Pro snímání obrazu a vyhodnocování výsledků ze získaného snímku je použitý software **NI Vision Builder for Automated Inspection**. V následujících bodech jsou oba použité softwary detailněji popsány.

7.1 Software NI Vision Builder for Automated Inspection

Společnost National Instruments má v oboru strojového vidění a rozpoznání obrazu velmi dlouhou praxi a proto je její software používán v naprosté většině aplikací tohoto druhu. Software je velmi přehledný, uživatelsky poměrně jednoduchý a umožňuje tak snadno a rychle vytvořit i poměrně složité inspekce obrazu. Jeho největší předností je, že stejná společnost je i tvůrcem softwaru LabVIEW. To je hojně využíváno právě ve vývoji testovacích, měřících a řídících aplikací. Tyto dva produkty spolu vzájemně spolupracují a umožňují tak vzniklé aplikaci velkou flexibilitu a rozšíření o celou řadu funkcí.

7.1.1 Základní popis prostředí

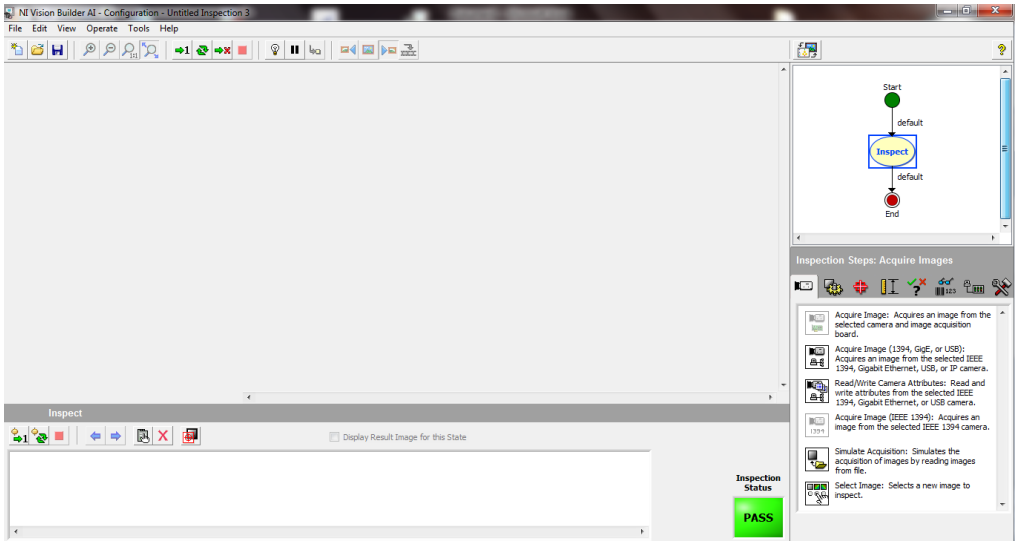
Startovní obrazovka softwaru zobrazena na obrázku 7-1 pod textem se skládá ze čtyř základních částí. [15]

1- Pracovní plocha pro zobrazení získaného obrazu. *(Zde se zobrazuje aktuální načtený a vyhodnocovaný obraz. Na něm jsou vidět jednotlivé provedené kroky a označeny hledané objekty a oblasti.)*

2- Vývojový diagram v pravém horním rohu *(Umožňuje vytvoření složitějších procesů a jejich větvení na základě výsledku předchozí inspekce, nebo splnění či nesplnění dané podmínky.) 7.1.2.*

3- Řetězec jednotlivých inspekčních kroků ve spodní části *(Zde jsou zobrazeny jednotlivé inspekční kroky v řadě. Je možné jednotlivé kroky upravovat, spouštět samostatně, nebo krok zakázat.) 7.1.3.*

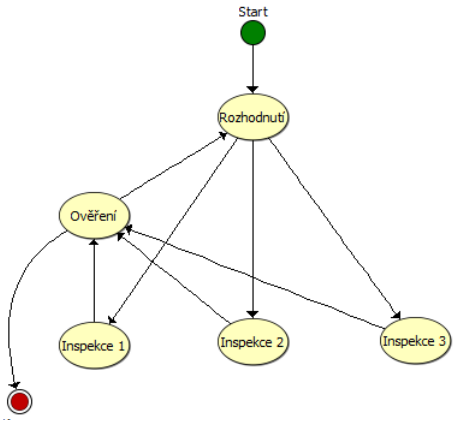
4- Nabídka možností a inspekčních kroků v pravém spodním rohu *(detailněji rozepsána v bodu 7.1.4)*



Obrázek 7-1 Základní obrazovka NI Vision Builder AI

7.1.2 Vývojový diagram inspekce

Vývojový diagram inspekce zobrazený pod textem se používá v případě složitějších procesů. Do každé inspekce lze implementovat samostatný inspekční řetězec a na základě možných výsledků definovat následnou cestu. Těchto rozhodovacích procesů se využívá například na výrobních linkách, kde se vyskytuje více druhů výrobků v náhodném pořadí. [15]



Obrázek 7-2 Vývojový diagram inspekce

7.1.3 Řetězec inspekčních kroků

Na následujícím obrázku je ukázka, jak vypadá inspekční řetězec (konkrétně jde o odečítání rychlosti z přístroje).

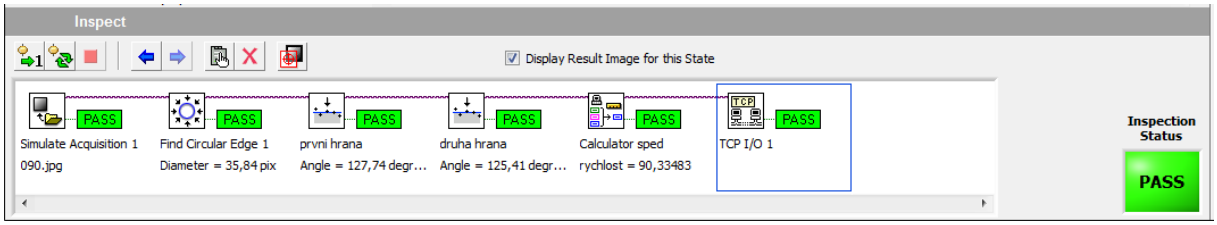
V prvním kroku je načtení snímku.

Druhý krok vyhledává kruhovou oblast (stupnici tachometru).

Třetí a čtvrtý krok hledá hrany. Tj. pravou a levou stranu rafičky resp. její pozici.

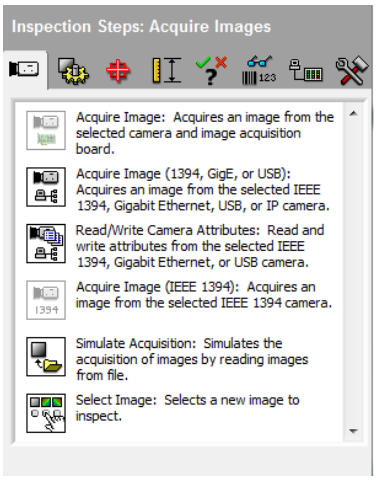
Pátý krok vypočítává z nalezených úhlů příslušnou rychlost odpovídající skutečnosti.

Šestý krok odesílá tento výsledek do externího prostředí pomocí protokolu TCP/IP



Obrázek 7-3 Řetězec inspekčních kroků

7.1.4 Popis jednotlivých inspekčních kroků

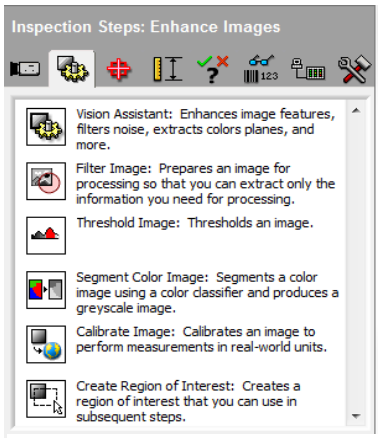


Obrázek 7-4 Paleta pro získání obrazu

Skupina „získání obrazu“

V této skupině inspekčních kroků jsou možnosti pro načtení obrazu. Jsou zde možnosti načtení z připojené kamery přes jeden z definovaných portů (Ethernet, FireWire 1394 nebo USB).

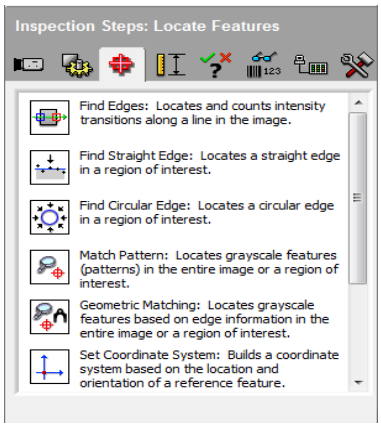
Další možností je načtení obrazů z počítače jako simulace kamery. Toho lze využít při takzvaném off-line testování, kdy jsou obrázky nebo fotografie načítány ze složky a interpretovány, jako kdyby je pořídila připojená kamera.



Obrázek 7-5 Paleta pro úpravu obrazu

Skupina „Úprava obrazu“

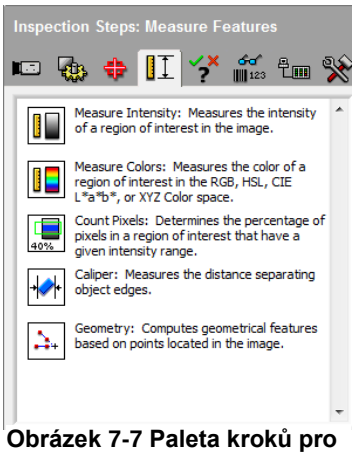
V druhé skupině jsou umístěny kroky pro úpravy obrazu. Prvním z nich je „Vision Assistant“. Ten obsahuje barevné filtry, umožňuje odfiltrování barev, zaostření obrazu a úpravy jasu. Celkem přes dvacet nástrojů pro úpravu obrazu. „Filter Image“ umožňuje rychlé použití přednastavených filtrů. Dále jsou zde další možnosti pro co nejlepší připravení obrazu pro inspekci.



Obrázek 7-6 Paleta pro určení rysů obrazu

Skupina „Určení rysů“

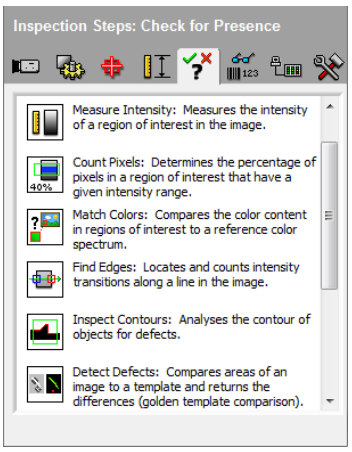
Třetí skupina obsahuje kroky pro detekci objektů. První z možností je detekce hran, kterou lze využít třeba pro nalezení pozice hledaného objektu. Další z možností je detekce kruhových objektů, nebo třeba porovnání se vzorem. To buď podle geometrické shody, barevné shody, nebo kombinace těchto dvou. Inspekční krok „Set Coordinate System“ umožňuje posunutí souřadného systému, což eliminuje případnou chybu způsobenou posunem snímaného objektu nebo kamery.



Obrázek 7-7 Paleta kroků pro měření v obraze

Skupina „Měření obrazu“

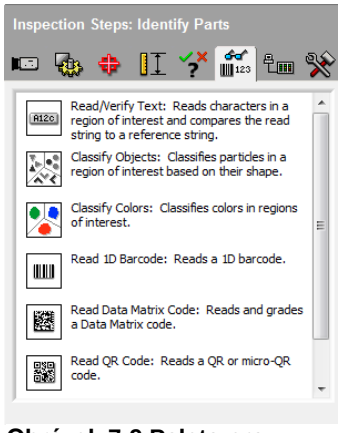
Čtvrtá záložka: měření znaků obrazu má následující kroky. Měření intenzity: Měří intenzitu v oblasti zájmu. Měření barev: Podíl barevné složky vzhledem k RGB, HSL nebo CIE. Měření počtu bodů: určuje procentuální četnost bodů v oblasti zájmu, které mají předem definovanou intenzitu. Poslední částí je odměřování mezer, které jsou odděleny hranami objektu.



Obrázek 7-8 Skupina prvků pro kontrolu přítomnosti

Skupina „Kontrola přítomnosti“

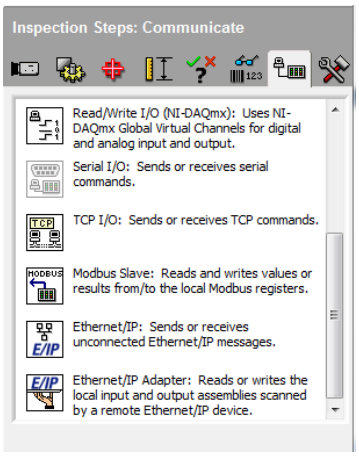
Tato skupina opět obsahuje měření intenzity, určení procenta obrazových bodů podle intenzity. Porovnání barevného obsahu v oblasti s referenčním zdrojem, nebo předem definovaným spektrem. Detekce hran pro detekci a lokalizaci intenzity přechodu podél čáry. Detekování defektů způsobem, že porovná oblasti s předepsanou šablonou a vrátí rozdíl. Dalšími kroky mohou být porovnávání shody se vzorem podle stupnice šedi a další. [15]



Obrázek 7-9 Paleta pro rozpoznání, identifikaci a verifikaci

Skupina „Rozpoznání částí“

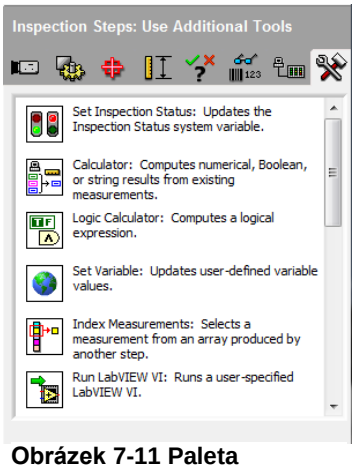
Tato skupina obsahuje inspekční kroky pro čtení čárových a maticových kódů, rozpoznávání jednotlivých částí na základě jejich tvaru. Podstatným krokem je zde OCR. To umožňuje v obrazu najít a rozpoznat text. Ten poté porovnává s definovaným řetězcem a určuje shodu, nebo interpretuje nalezený text jako řetězec znaků. [15]



Obrázek 7-10 Paleta pro usnadnění komunikace

Skupina „Komunikace“

Ve skupině komunikace jsou umístěny kroky, které umožňují snazší připojení a komunikaci s jiným prostředím. Jsou zde předdefinovány kroky pro komunikaci pomocí TCP/IP protokolu, klasického digitálního výstupu, číst nebo zapisovat výsledky do Data Socket serveru a různé další.



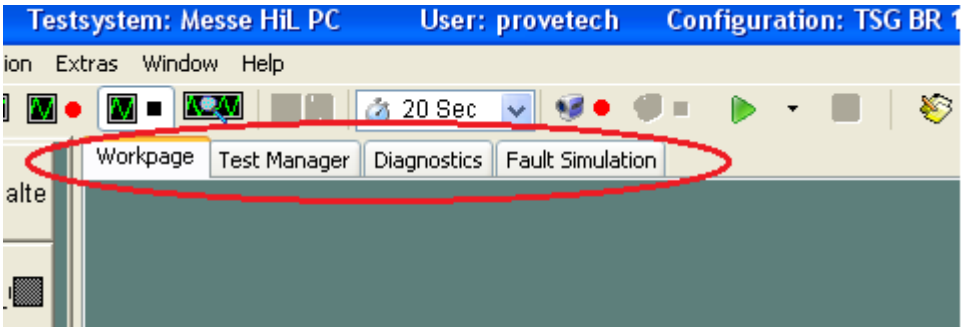
Obrázek 7-11 Paleta doplňkových nástrojů

Skupina „Použití dalších nástrojů“

V tomto kroku jsou doplňky softwaru. Je zde možnost nastavení výsledku inspekce pomocí aktualizace globální proměnou. Kalkulačka, která umožňuje jak numerické, tak i logické početní operace. Nastavování globálních proměnných, indexování měření, nastavení zpoždění, záznam dat, uložení obrázku a uživatelský vstup. Jedním krokem je i přímí export do prostředí LabView.

7.2 Software PROVEtech:TA (Test Automation)

Tato část zjednodušeně popisuje testovací prostředí PROVEtech:TA, který je produktem skupiny MBtech Group vyvinutý pro oblast HiL testování. Jde o komplexní nástroj pro použití nejen v automatizovaných testech. Samotný nástroj se skládá ze čtyř základních částí reprezentovaných čtyřmi záložkami v horní části a to Workpage, Test Manager, Diagnostics a Fault Simulation. O částech Diagnostics (diagnostika) a Fault Simulation (simulace chyb), se zmíním jen okrajově. Tyto dvě části nejsou v případě této diplomové práce využívány.



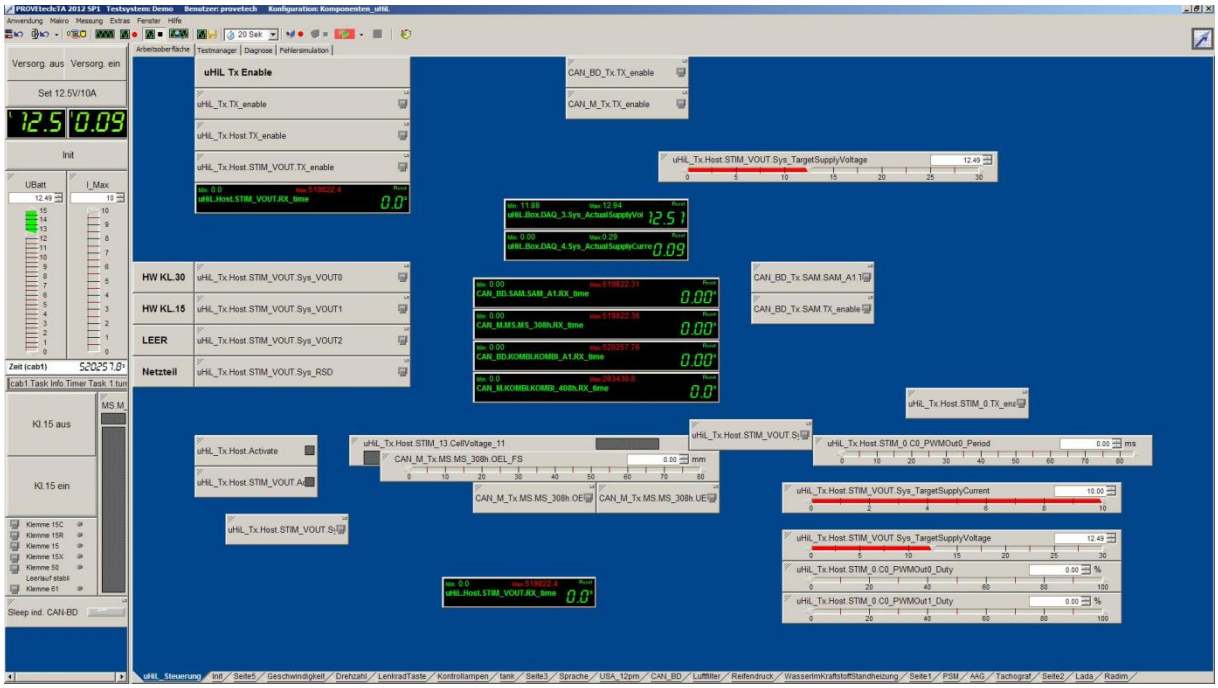
Obrázek 7-12 Záložky v prostředí PROVEtech:TA

Software umožňuje uživateli pracovat v interaktivním prostředí, ve kterém může nastavovat a měřit všechny signály, stavy, a proměnné v testovaném systému. Vzhledem k poskytované široké databázi obsahující podporu pro vykonávání typických úkolů v managementu testů, správu testovacích sestav, knihoven a zpracování výsledků testů, vytváří PROVEtech:TA uživatelsky přívětivé prostředí pro snadné řešení i velmi náročných úkolů. Díky široké škále rozličných ovládacích a zobrazovacích prvků, je velmi usnadněn přístup k signálům potřebných pro testování.

PROVEtech:TA je ideálním nástrojem pro nasazení v oblasti automatizovaných testů po celou fázi vývoje produktu. Už od počátečních kroků vývoje, po celou dobu, až do samotného výrobního stavu. Vzhledem k tomu, že tento software je hardwarově nezávislý a umožňuje přístup ze stran více uživatelů, stává se tak optimálním pro použití v rámci pracovní skupiny.

7.2.1 *Workpage - pracovní plocha*

První část workpage, neboli pracovní plocha, umožňuje vizualizaci a parametrizaci signálů v průběhu testování. Díky široké škále ovládacích a zobrazovacích prvků si může uživatel vybrat, vždy nejvhodnější prvek pro požadavky na ovládání či zobrazování daného signálu. Prvky zde obsažené mají podobu například posuvných či otočných prvků, tlačítek a mnohých dalších. Pro zobrazení jsou zde umístěny prvky jako třeba vykreslování různých grafů, zobrazení číselných hodnot a další. Veškeré tyto prvky a tím i samotné signály k nim přiřazené, mohou být ovládány manuálně vstupem od uživatele, ať už pomocí myši, nebo zadáním přímé hodnoty z klávesnice. Na obrázku pod textem je vidět právě pracovní plocha tohoto nástroje. [16]



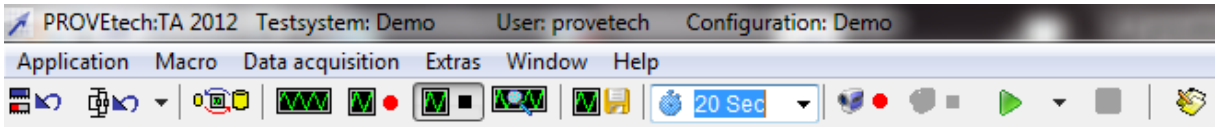
Obrázek 7-13 Workpage - Pracovní plocha

Jak je vidět na obrázku 7-13, na pracovní plochu můžeme umisťovat libovolné množství prvků pro ovládání či zobrazování stavů. Samotná pracovní plocha může mít několik listů, kde na každém mohou být zcela jiné prvky. Mezi těmito listy lze přecházet pomocí záložek v dolní části. Záložky může uživatel libovolně přidávat či odebírat a měnit jejich jména. Všechny strany pracovní plochy však mají jednu společnou část. Je jimi takzvaná palubní deska (cockpit area).

Cockpit area je nepřepínatelná část pracovní plochy, na které mohou být zobrazeny stejné prvky jako na pracovní ploše. Tato část je zobrazena uživateli vždy, bez ohledu na to, na které stránce se zrovna nachází. Z tohoto důvodu je vhodná pro umístění základních ovládacích prvků, ke kterým je potřeba přistupovat ve více částech pracovní plochy.

Ovládací prvky v horní části

Tato část vysvětluje význam ikon umístěných v horní části okna, které jsou zobrazeny na obrázku pod textem.



Tyto dvě ikony provádějí reset/restart. První z nich „Reset Signals“ nastaví všechny dostupné signály do původní (defaultní) hodnoty. Druhá ikona „Reset Model“ slouží k restartování testovaného modelu.

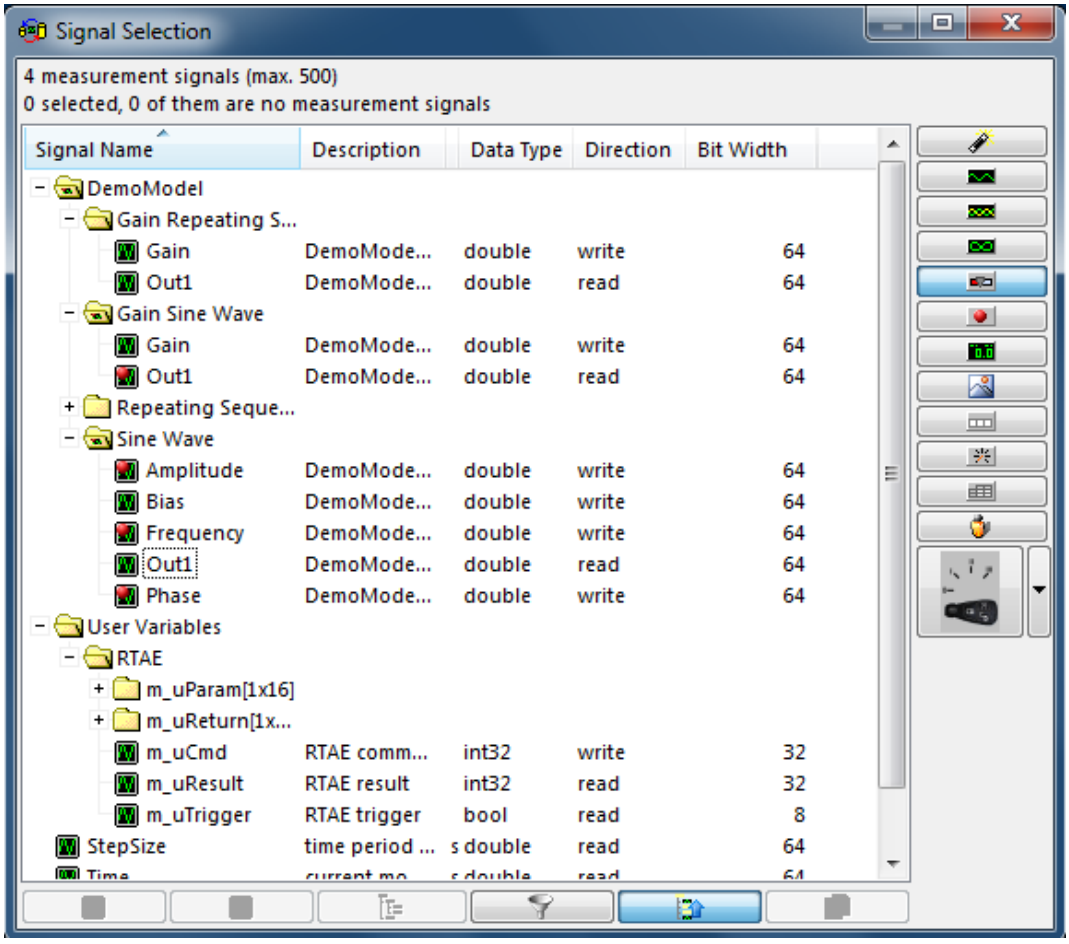
Význam této ikony „Signal Selection“ spočívá v otevření okna pro výběr z dostupných signálů. Toto okno je popsáno v další části.

Tato sada ikon je určena pro práci při měření. Umožňuje zobrazení časového průběhu signálů, spustit či zastavit jejich měření, pracovat v offline režimu a uložit signál.

Ve zbylé části panelu je možnost nastavení časové osy pro zobrazování měřených signálů a záznam či spouštění makra.

Okno výběru signálu (Signal Selection)

Okno výběru signálu lze rozdělit na tři základní části. Tou hlavní je vnitřek samotného okna obsahující složky se signály. U každého signálu je i popis (název), datový typ, druh signálu write (můžeme měnit hodnoty) nebo read (můžeme hodnoty pouze odečítat), a bitová délka.



Obrázek 7-14 Okno pro výběr signálu

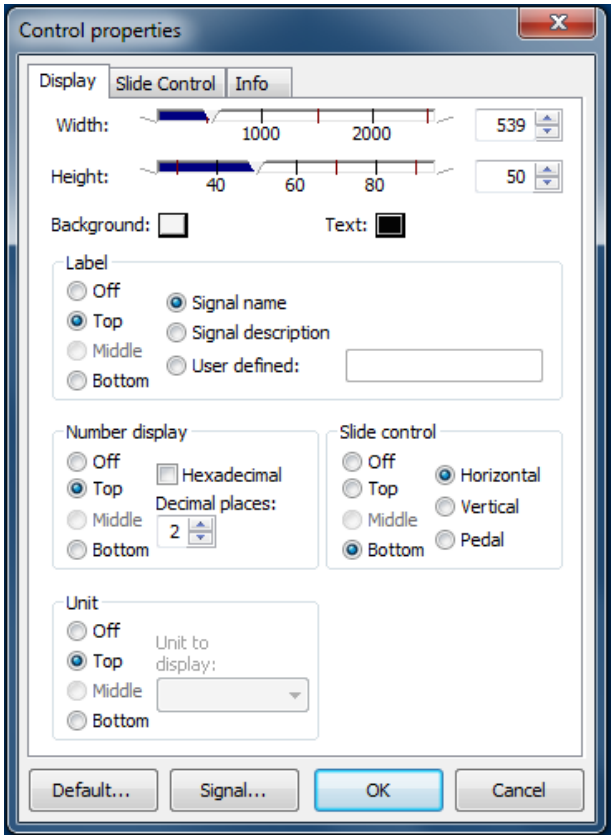
Druhou částí je záhlaví a zápatí okna. V záhlaví máme vypsané informace o stavu signálů. Kolik signálů je právě měřených z možného maxima, kolik z nich je vybráno a další.

V zápatí je možnost nastavení filtru pro zobrazení jen určitých signálů, což je výhodné třeba ve velkých aplikacích při velkém množství signálů, kde je zobrazení všech dostupných signálů velmi nepřehledné.

Třetí částí je postranní panel vpravo, který obsahuje prvky pro měření a ovládání signálů. Některé prvky mají pouze zobrazovací a jiné naopak ovládací charakter. Umístění ovládacího prvku na pracovní plochu se provede tak, že uživatel nejprve provede výběr požadovaného prvku, poté vybere příslušný signál a myší jej přetáhne na pracovní plochu. Tam se objeví zvolený ovládací prvek s vazbou na vybraný signál. Druhy ovládacích prvků jsou popsány níže.

Úprava ovládacích prvků

Jednotlivé ovládací prvky je možno dále upravovat, aby odpovídaly přesným požadavkům. Mimo základních operací jako je kopírování, vkládání a změna názvu prvků, jsou zde možnosti jako třeba upravit osy zobrazované oblasti pro přesnější zobrazení, nastavení kroku ovládání, vymazání či resetování signálu, přidávání a odebírání signálů a celková změna vzhledu. [16]



Obrázek 7-15 Možnosti nastavení vlastností ovládacího prvku

Změna pozice a velikosti zobrazeného prvku se provádí jednoduše pomocí přetažení myší, nebo přímo nastavením souřadnic a velikosti v menu prvku. Na obrázku 7-15 je zobrazeno právě okno nastavení jednoho vybraného prvku. V tomto případě jde o ovládací prvek „Slider or bar“. Je zde vidět v horní části nastavení výšky a šířky zobrazeného prvku, umístění popisků osy, definice názvu prvku, nastavení zobrazovaných čísel a jednotek. Další z možností je zpětné nastavení do původních hodnot. [16]

Seznam ovládacích prvků signálu a jejich význam

Symbol	Název	Popis
	Automatic type selection	V případě zvolení tohoto prvku, PROVEtech automaticky vybere nejvhodnější z dostupných prvků
	Strip chart	Vykreslí průběh zvoleného signálu za čas
	Multi strip chart	Stejný jako Strip chart, ale umožňuje zobrazit více vybraných signálů najednou
	XY strip chart	Vykreslí jeden signál (Y) v závislosti na druhém (X)
	Slider or bar	Posuvný prvek, sloužící jako zobrazovací i ovládací
	Switch or LED	Zobrazovací přepínač. Používán u jednobitových informací (svítí/nesvítí)
	Number	Zobrazuje hodnotu signálu v libovolném číselném formátu
	Bitmap	Používá se v případě potřeby zobrazení obrázku nebo textu
	Bitswitch	Používá se pro zobrazení nebo nastavení binárního signálu
	Gauge	Zobrazuje a zároveň umožňuje nastavovat hodnotu signálu jako na kruhové stupnici
	Gird control	Zobrazuje vektory nebo matice jako tabulky
	Custom	Umožňuje použít uživatelem nadefinovaný ovládací prvek
	Special	Obsahuje velké množství ovládacích a zobrazovacích prvků, které svým vzhledem připomínají reálné prvky. Těmi je například ovladač světel, řadicí páka, spínací skříňka, tachometr a další.

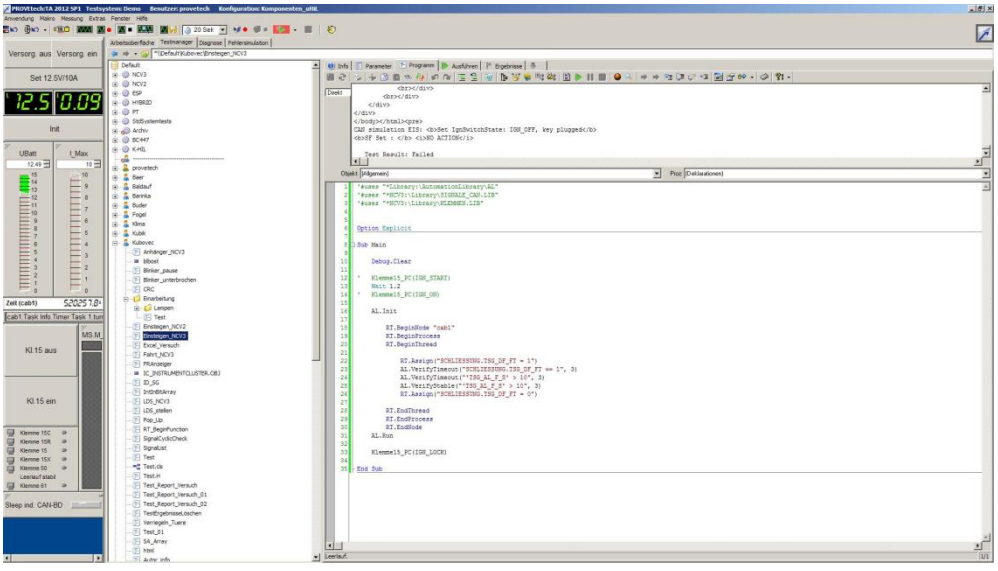
Tabulka 7-1Seznam ovládacích prvků signálu a jejich význam [16]

7.2.2 Test Manager

Druhou částí nástroje PROVetech:TA je Test Manager. Ten je pro chod celého systému v oblasti automatizovaných testů klíčovým. Umožňuje vytváření a kompletní správu testů v průběhu samotného testování. Oproti zmiňovanému manuálnímu nastavování signálů na pracovní ploše, je zde možnost vytvořit plně automatizované ovládání signálu pomocí testovacího skriptu. To výrazně usnadňuje veškeré testování a zaznamenávání hodnot. Veškeré testovací skripty se vytváří v objektově orientovaném jazyce WinWrap Basic. Tento jazyk je ve své podstatě a syntaxi velmi podobný jazyku Visual Basic známému třeba z maker produktů Microsoft Office.

Test Manager umožňuje nejen psát nové testovací skripty, ale především i správu všech existujících a dříve vykonaných testů, společně se svými výsledky a protokoly o průběhu. To velmi usnadňuje opakované vykonávání testů či dohledání protokolů o průběhu testu. Testovací skripty lze spouštět každý samostatně, nebo ve skupinách.

Obrazovka PROVEtechu se po přepnutí do Test Manageru změní na rozdělení do tří základních oken (obr. 7-16). V levé části je seznam knihoven, projektů, přistupujících uživatelů a testů. Po zvolení ikony určitého uživatele, si můžeme prohlédnout veškeré jeho testy a projekty. V závislosti na oprávnění, lze tyto testy i upravovat či do nich jiným způsobem zasahovat. Toho se využívá v případě spolupráce několika lidí na jednom projektu.

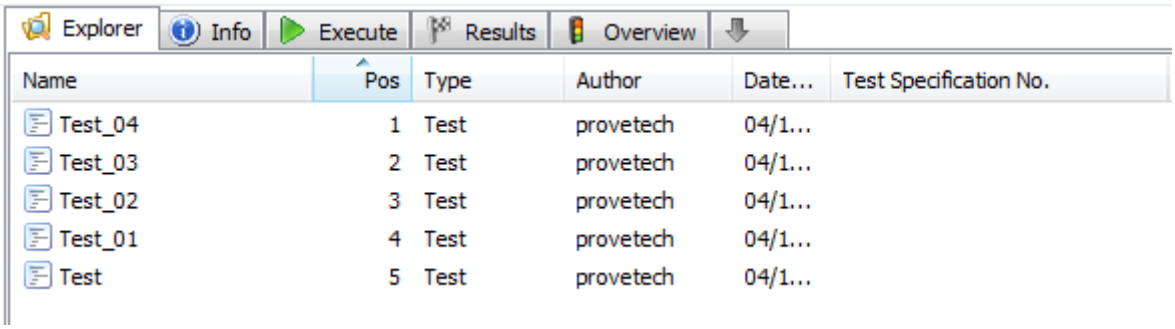


Obrázek 7-16 Obrazovka Test Manageru

Vlastní okno Test Manageru obsahuje pět dalších podoken. Čtyři z nich jsou pro správu a ovládání testů velmi důležitá a jsou popsány v následujícím bodu.

Explorer (Průzkumník)

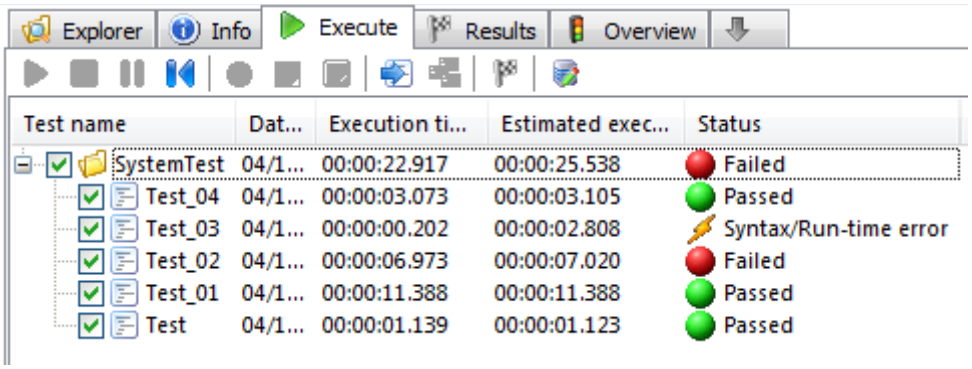
V této části jsou zobrazeny veškeré existující testy ve složkách a je tak umožněno přehledné vyhledávání testu, ať už podle názvu testu, autora nebo data poslední změny. Po otevření zvoleného testu se zobrazí zdrojový kód pro nahlédnutí či úpravu. [16]



Obrázek 7-17 Explorer - Test Manageru

Execute (Vykonání)

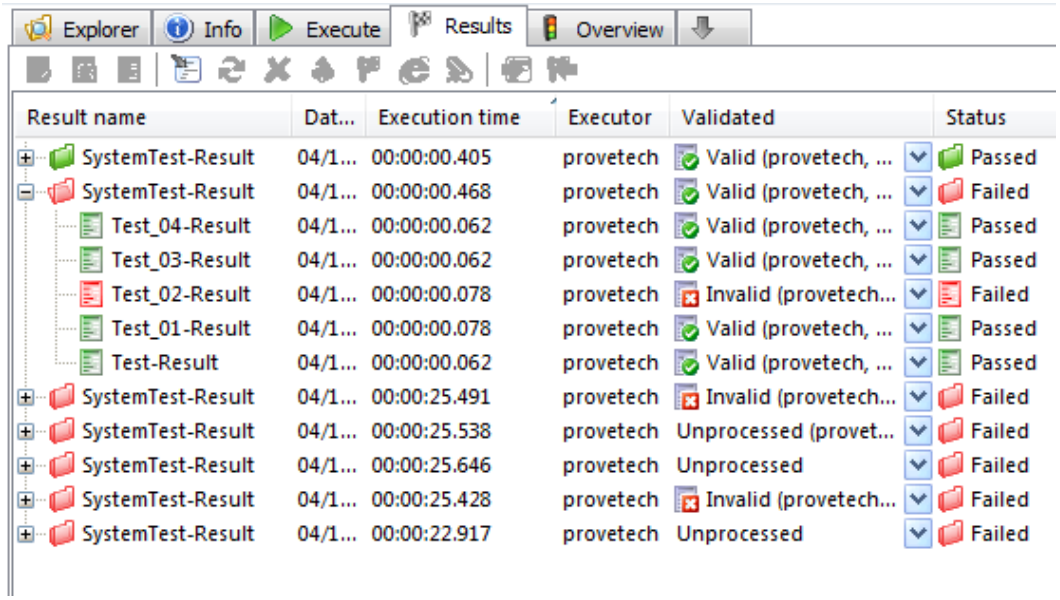
Záložka Execute umožňuje spouštět skupiny vybraných testů. Z nabídky dostupných skriptů lze vybrat podle potřeby ty, které chceme spustit. V popisu u každého testu je vidět opět datum poslední změny, doba trvání testu, předpokládaná doba průběhu testu a výsledek testu. Ten může být reprezentován buď zeleným kolečkem (test v pořádku), červeným kolečkem (test selhal), nebo znakem označující chybnou syntaxi ve zdrojovém kódu a test tak nemohl být spuštěn. [16]



Obrázek 7-18 Execute - Test Manageru

Results (Výsledky)

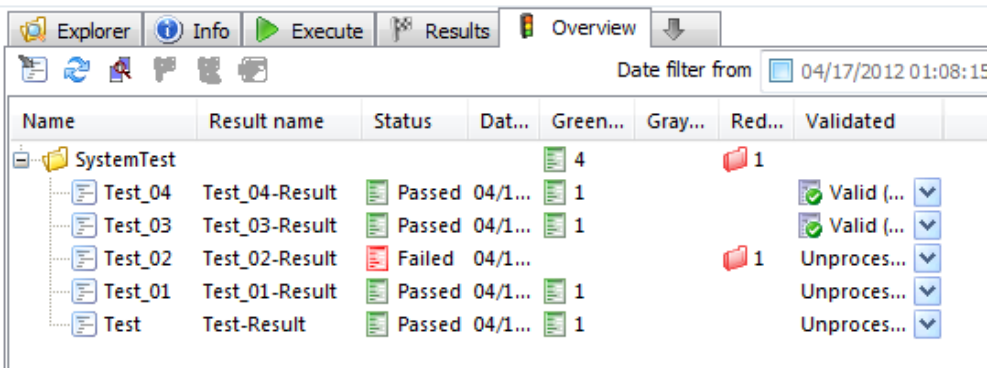
Část s názvem Results obsahuje záznamy a výsledky všech provedených testů. Opět jsou zde zeleně označeny správné testy a červeně ty, které nevyhověly. Po výběru konkrétního testu je možné vygenerovat zápis o jeho průběhu (tzv. System test protocol). Mimo toho je uživateli v této části umožněno nastavit schválení či neschválení testu. [16]



Obrázek 7-19 Results - Test Manageru

Overview (Přehled)

Poslední důležitou částí je Overview. Zde jsou uloženy všechny zprávy o provedených testech se zápisem o výsledku a průběhu testu. Lze tak jednoduše a přehledně kontrolovat stavy testů a vytvořit tak závěrečnou správu o průběhu a úspěšnosti. [16]



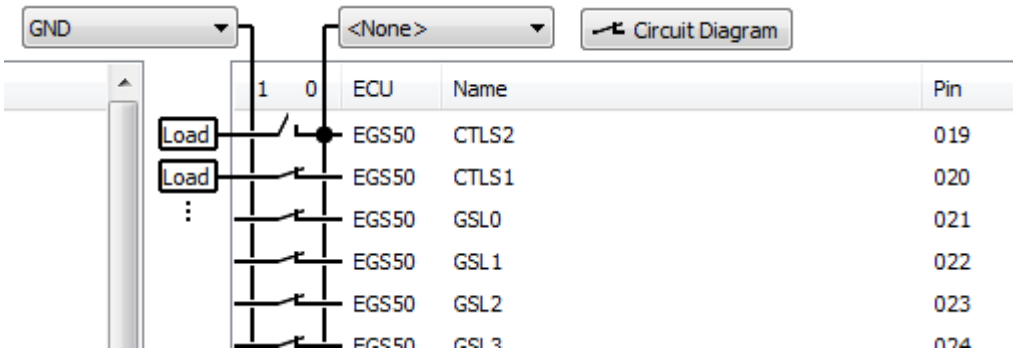
Obrázek 7-20 Overview - Test Manageru

7.2.3 Diagnostics

Další část v prostředí PROVEtech:TA je Diagnostics. Tento nástroj slouží k diagnostice testovaných řídících jednotek přes příslušné hardwarové rozhraní. Lze tak provádět čtení provozních hodnot přímo z připojené řídící jednotky, provádět mazání chyb, nebo nahrát do jednotky nový firmware. Jako programové rozhraní využívá Caesar. To je používáno pro diagnostiku veškerých řídících jednotek v koncernu Daimler AG. [16]

7.2.4 Fault Simulation

Poslední, čtvrtá část je využívána pro simulaci chyb na řídící jednotce či sběrnici. Jsou zde možnosti jako třeba propojit libovolný vodič se zemí, s napájením, simulovat zkrat mezi několika vodiči nebo přerušení vodiče. Část schématu pro nastavení propojení vodičů je zobrazena na obrázku 7-21. [16]

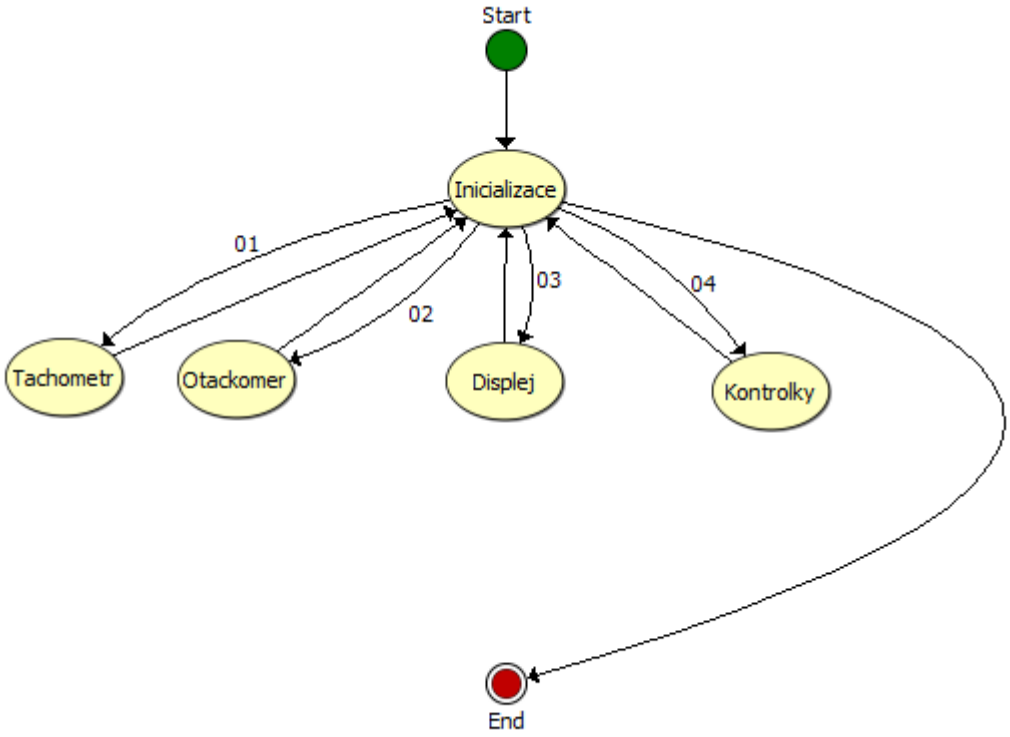


Obrázek 7-21 Simulace chyb PROVEtech:TA

8 Řešení úlohy

8.1 Popis struktury testů NI Vision Builder

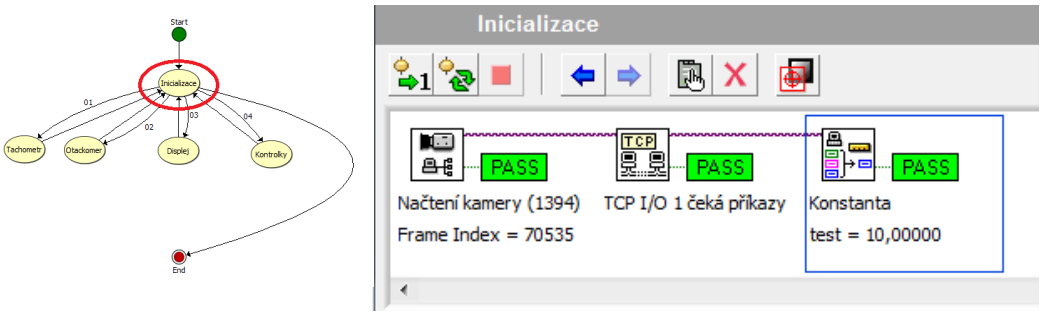
Struktura testování v prostředí NI Vision je složena z pěti základních kroků. Prvním krokem je inicializace testu, kde se rozhoduje, který test se bude provádět na základě obdržené hodnoty. Druhým až pátým krokem jsou samotné inspekce částí obrazu. Všechny kroky jsou detailněji popsány v následujících kapitolách.



Obrázek 8-1 Struktura testů NI Vision Builder

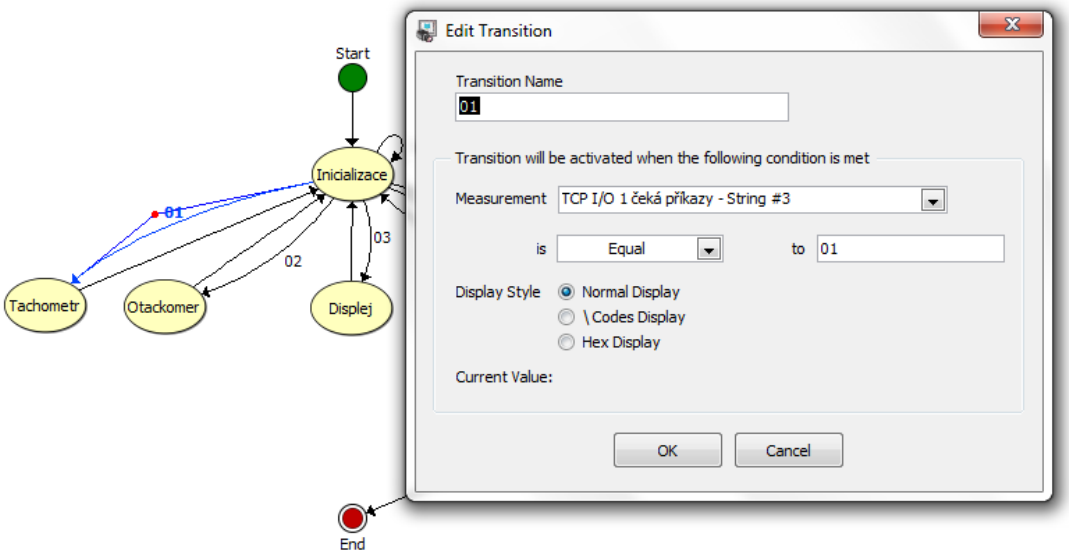
8.1.1 Inicializace

Tento první krok testů je nejpodstatnějším krokem v komunikaci a spouštění vlastních testů. Dochází zde k ověření připojení kamery a přijetí informace o tom, jakým testem se bude dále pokračovat. Na konci je řetězec zakončen konstantou, z důvodu snazšího ověření vzájemné komunikace mezi programy. Níže je detailněji rozepsaná stěžejní prostřední část řetězce, týkající se komunikace.



Obrázek 8-2 Kroky obsažené v části Inicializace

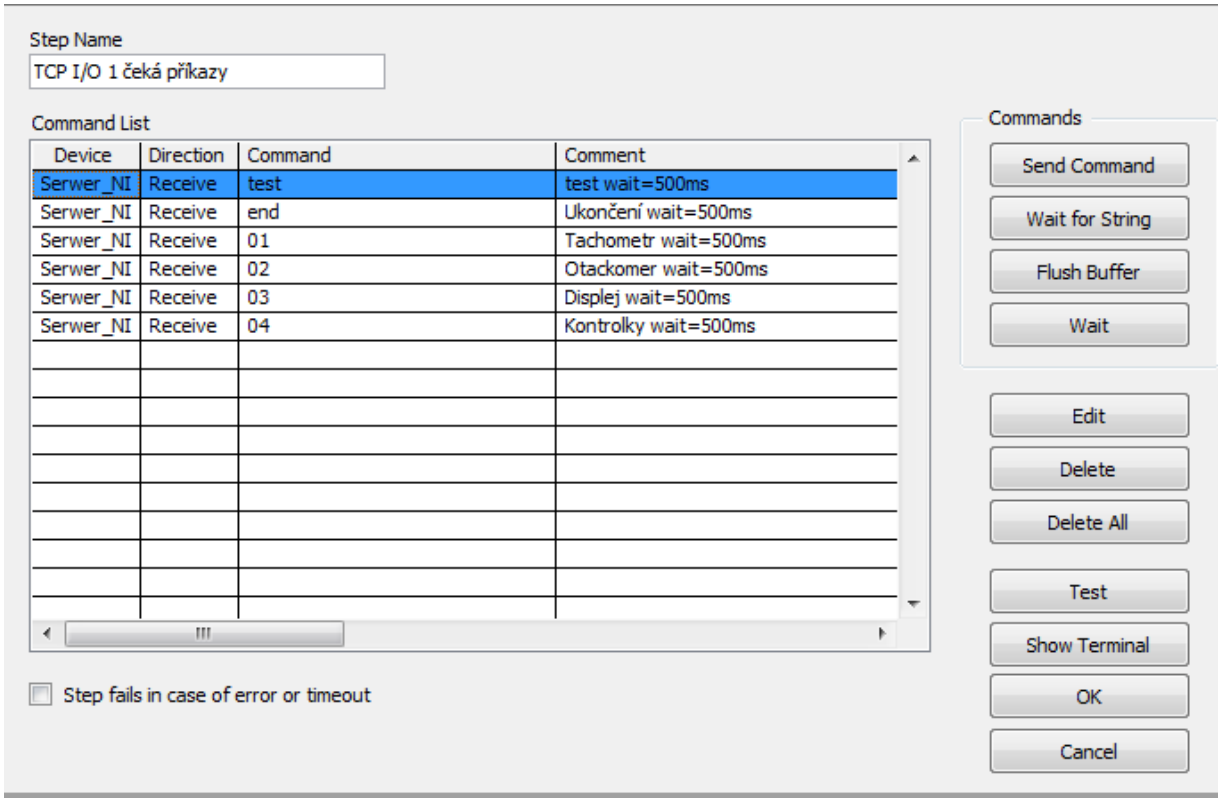
Na následujícím obrázku je zobrazeno nastavení cesty (Transition). U každé vytvořené cesty je potřeba definovat podmínku, za které bude aktivována. Jak je vidět na obrázku 8-3 nastavuje se zde porovnání přijaté hodnoty nebo řetězce s nastavenou konstantou. V tomto případě jde o konstantu „01“ pro detekci oblasti tachometru. Stejným způsobem jsou nastaveny i ostatní cesty. Jen v případě přijetí příkazu „test“ je navrácena hodnota konstanty „10“. Toho je využíváno pouze při ověření funkčnosti komunikace.



Obrázek 8-3 Panel pro nastavení rozhodovací podmínky u cesty (Transition)

Krok TCP I/O

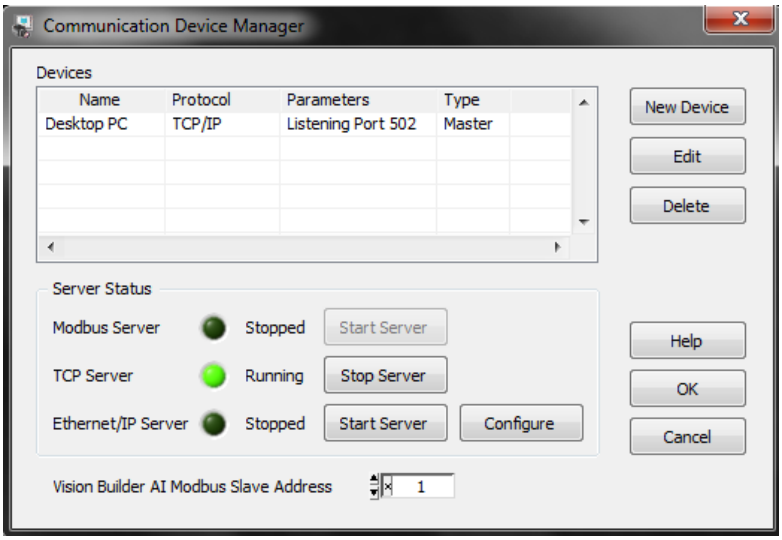
V tomto kroku jsou definovány konkrétní příkazy, na základě kterých dochází k výběru požadovaného testu. Nastavení se provádí kliknutím na tlačítko „Wait for String“ v pravé části okna (viz obr. 8-4). Otevře se dialogové okno, kde je možno zapsat hodnotu a datový typ očekávaného příkazu (Command), nebo nastavení času jak dlouho na příkaz čekat (Timeout). V tomto případě je u všech nastaven timeout na hodnotu 500ms, což je hodnota plně postačující nárokům aplikace. Všechny kroky se opakují ve smyčce. Každý z nich je tedy spuštěn na dobu 500ms každé tři sekundy. V komentářích na řádku vedle každého příkazu je stručný popis zapsaný uživatelem. [15]



Obrázek 8-4 Seznam a nastavení příkazů v kroku TCP/IP

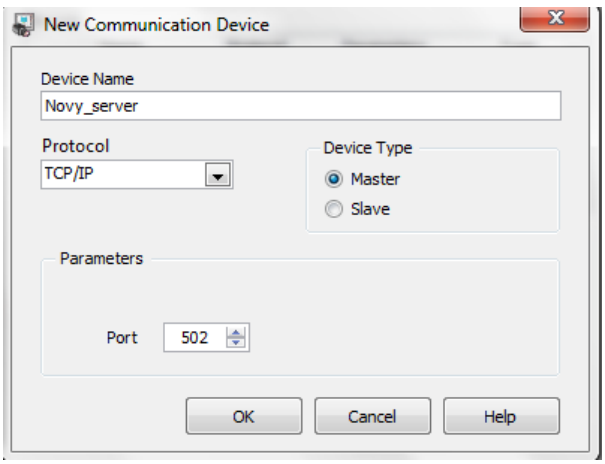
Nastavení serveru pro příjem a odesílání dat

V případě nastavení serveru pro komunikaci pomocí protokolu TCP/IP je využito nástroje Communication Device Manager (obr. 8-5), který je obsažen v tomto softwaru. Pro detailněji popsaný postup je vhodné nahlédnout do manuálu, jelikož způsoby komunikace a nastavení se u starších verzí mírně odlišují.



Obrázek 8-5 Okno Communication Device Manager

Po otevření okna Communication Device Manager zvolíme položku „New Device“ (obr. 8-6). Zde zadáme název serveru, který chceme vytvořit, vybereme typ komunikačního protokolu (TCP/IP), typ zařízení (Master nebo Slave) a číslo portu, na kterém spolu budou zařízení navazovat komunikaci. [15]

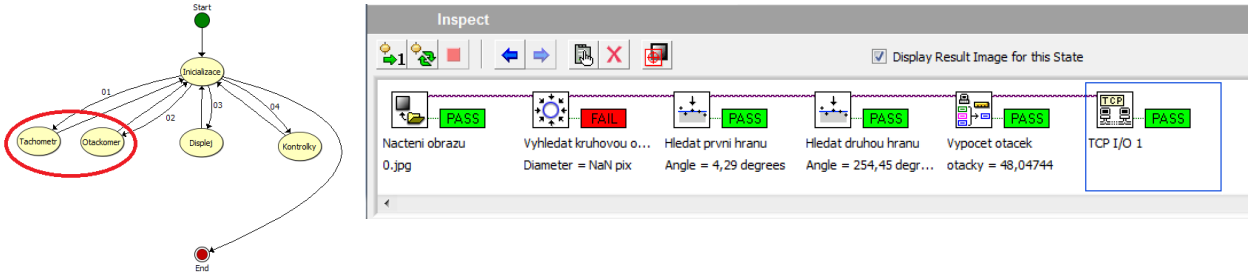


Obrázek 8-6 New Device

8.1.2 Test otáčkoměru / tachometru

U tohoto testu jde o odečtení hodnoty, která je interpretována analogovým ukazatelem na kruhové stupnici. V testech jsou rozeznávány dva tyto ukazatele. Jeden z nich je na tachometru (vlevo) a zobrazuje aktuální rychlost v kilometrech za hodinu. Druhý je na otáčkoměru (vpravo) a udává otáčky motoru za minutu. Oba jsou detekovány samostatně.

V této části je detailně popsána detekce ukazatele na otáčkoměru. Detekce na tachometru probíhá stejným způsobem pouze s jinými výpočty, kvůli rozdílné stupnici. Princip inspekce se skládá ze šesti kroků popsaných níže.



Obrázek 8-7 Jednotlivé kroky obsažené v testu otáčkoměru

1 – načtení obrazu

2 – vyhledání kruhové oblasti

3 – detekce první hrany

4 – detekce druhé hrany

5 – kalkulátor

6 – odeslání výsledku

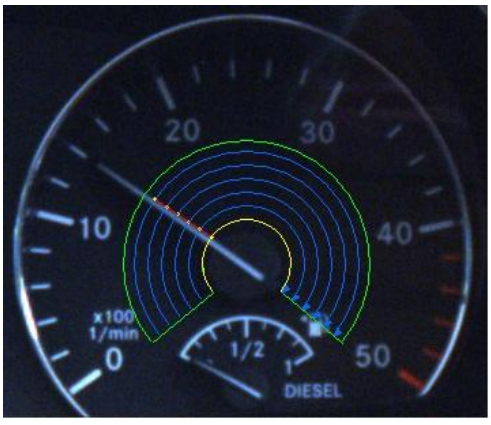
Inspekční kroky:

1 – *Načtení snímku*. Načtení snímku se provádí z kamery připojené přes rozhraní FireWire (IEEE 1394), nebo v tomto případě v režimu offline ze složky s uloženými snímky.

2 – *Vyhledání kruhové oblasti.* V tomto kroku se detekuje pracovní oblast, se kterou nadále pracujeme. Jde o nalezení výseče uvnitř kruhu, kde se nachází ukazatel za jakýchkoliv provozních podmínek. (Oblast ohraničena zelenou čárou)

Obrázek 8-8 Vyhledávání kruhové

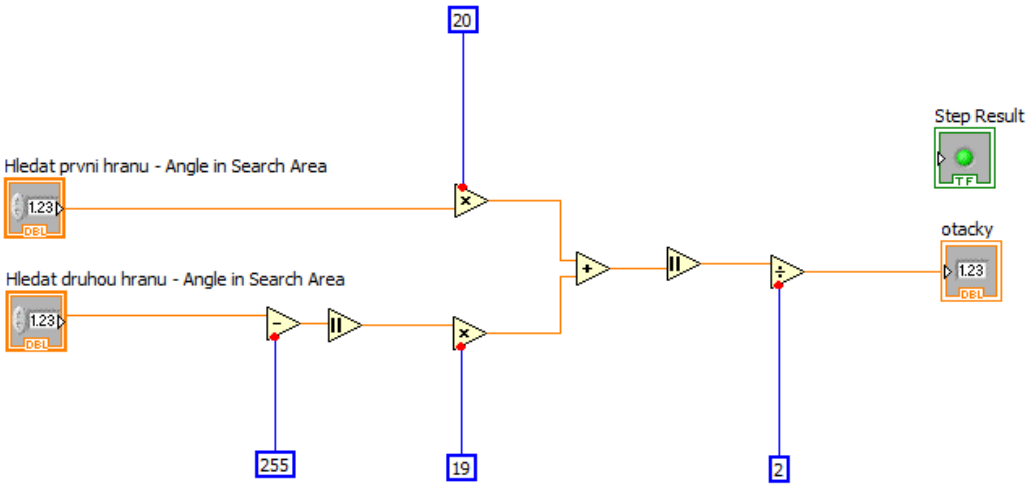
3,4 – *Detekce hrany.* V tomto kroku se podle nastavených parametrů provádí detekce hrany. V tomto případě se prochází každá jednotlivá křivka (znázorněna modrou barvou), dokud nedojde k detekci hrany. Z nalezených bodů se vytvoří přímka, která je dále reprezentována jako nalezená hrana. Ta je popsána úhlem natočení a polohou v kartézském souřadném systému. Tato detekce se provádí dvakrát a to jednou po směru hodinových ručiček a podruhé v protisměru. Tyto dvě hodnoty se poté zprůměrují a tím je zajištěna vyšší přesnost.



Obrázek 8-9 Ukázka detekce hrany

5 – *Kalkulátor.* V tomto kroku dochází k přepočítání naměřených hodnot na výsledné otáčky. V našem případě jde o přepočet úhlu natočení ručičky k hodnotám na stupnici otáček. Vstupní signály jsou dva a to úhel natočení spodní a horní hrany. U spodní hrany (protisměru hodinových ručiček) musíme odečíst úhel celé výseče tj. 255° a převést na absolutní hodnotu. Každý ze signálu je poté vynásoben konstantou stupnice a zprůměrován. Výsledná hodnota je hodnota otáček odpovídající ukazateli na palubním přístroji. Celý výpočet lze popsat následující rovnicí. Vstupní hodnoty z měření (tj. úhly natočení), jsou v rovnici označeny jak α a β.

$$ot.motoru = \frac{20\alpha + (\,|\beta - 255°|.19)}{2}$$



Obrázek 8-10 Blokové schéma výpočtů v kalkulátoru

6 – *Odeslání informace.* V tomto posledním kroku se provádí odeslání získaných informací pomocí protokolu TCP/IP. V tomto konkrétním případě odesíláme informace o stavu a času testu a výslednou hodnotu otáček vypočtenou kalkulátorem.

Step Name
TCP I/O 1

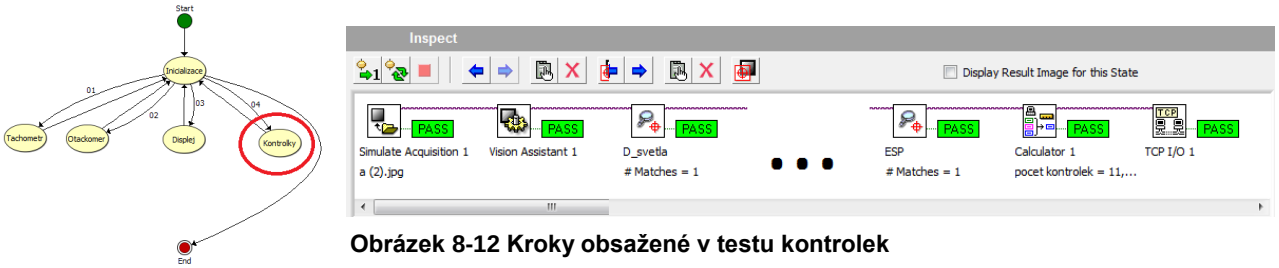
Command List

Device	Direction	Command	Comment
Server_NI_test	Send	<System Variable - # Pass>	Stav testu
Server_NI_test	Send	<System Variable - Active Time (s.)>	Čas testu
Server_NI_test	Send	<Calculator 1 - otacky>	Otáčky motoru

Obrázek 8-11 Seznam příkazů obsažených v kroku TCP/IP u testu otáčkoměru

8.1.3 Test kontrolek

U tohoto testu se zjišťuje stav kontrolek, které kombi přístroj obsahuje. Kontrolky mohou nabývat pouze dvou stavů a to svítí, nesvítí (0/1). V případě tohoto kombi přístroje detekujeme celkem 13 kontrolek. V inspekčním řetězci je celkem 17 kroků. Prvním je načtení obrazu, druhým obrazová úprava, poté následuje třináct kroků pro detekci samotných kontrolek. Poslední dva kroky jsou kalkulátor a odesílání výsledků (TCP/IP). Kalkulátor vypočítává kolik kontrolek je aktivních (svítí). Toho se využívá například po simulaci zapnutí klíčku, kdy se musí rozsvítit předem daný počet kontrolek. Naopak po uplynutí určitého časového intervalu od zapnutí, nesmí být aktivní žádná kontrolka.



Obrázek 8-12 Kroky obsažené v testu kontrolek

Obrázek 8-13 Fotografie z průběhu testu (detekce kontrolek)

1- Načtení obrazu

2- Filtr obrazu

3-15 Detekce kontrolky

16- Kalkulátor

17- Odeslání výsledku TCP/IP

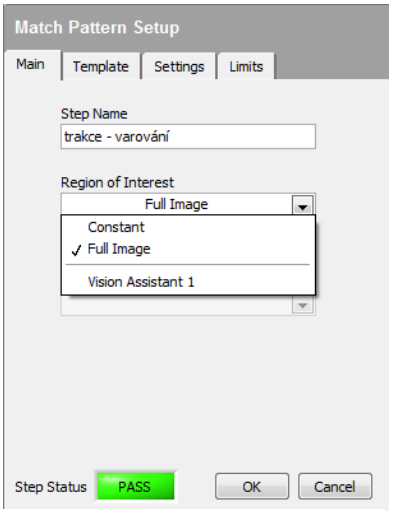
Inspekční kroky:

1 – *Načtení snímku*. Načtení snímku se provádí z kamery připojené přes rozhraní FireWire (IEEE 1394), nebo v tomto případě v režimu offline ze složky s uloženými snímky.

2 – *Filtr obrazu.* Filtr obrazu je prováděn z důvodu převedení obrazu na menší datový objem. Dalším důvodem je snazší detekce kontrolek, které se po odfiltrování rušivých vlivů dají snadněji a hlavně s vyšší přesností detekovat. Díky tomu lze prohledávat celý obraz a ne jen předem definovanou oblast. Tím je odstraněn i další problém, který vzniká v případě pohybu kamery a změny úhlu snímání.

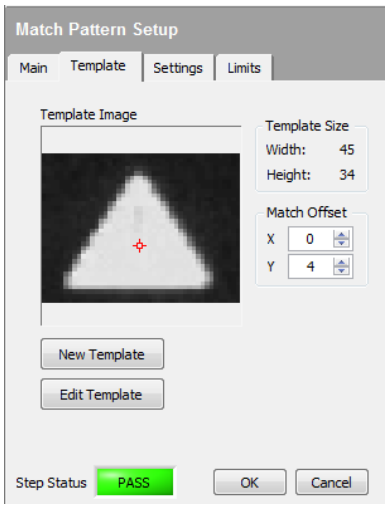
3-15 *Detekce kontrolek.* Detekce kontrolek je prováděna pomocí kroku „Match Patern“, který na základě předem definovaného vzoru prohledává celý obraz a v případě nalezení shodného objektu se vzorem tento objekt označí. Každý tento krok lze definovat ve čtyřech částech reprezentovaných čtyřmi záložkami popsanými níže.

V první záložce „Main“ se definuje název kroku. V tomto případě hledáme varovnou kontrolku trakce. Proto název „trakce - varování“. Další položkou tohoto menu je výběr oblasti, kde hledat shodu. Oblast může být konstantní (definovaná uživatelem), další možností je „Full Image“ (vyhledávání v celém obrazu), nebo oblast závislá na předchozím kroku. Třeba uvnitř kruhové oblasti vyhledávané předchozím krokem. V tomto případě se využívá možnosti „Full Image“.



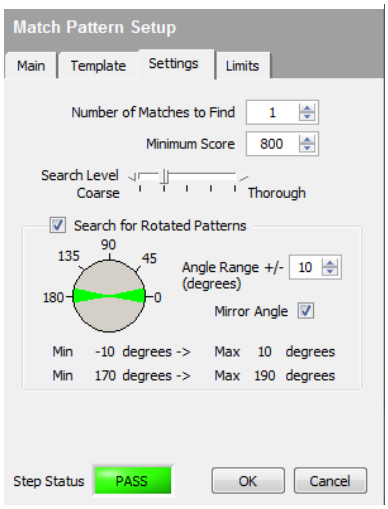
Obrázek 8-14 Záložka Match Patern Setup - Main

Druhá záložka „Template“ slouží k definování vzoru pro porovnání. V tomto případě je zde vzor uložen v černobílém stavu, jelikož porovnáváme pouze tvar. Jsou zde možnosti „New Template“, kde je možnost nastavit nový vzor k porovnání a „Edit Template“, kde můžeme stávající vzor ještě upravit. Vpravo jsou rozměry vzoru v pixelech a pod nimi nastavení povoleného rozdílu po osách X a Y.



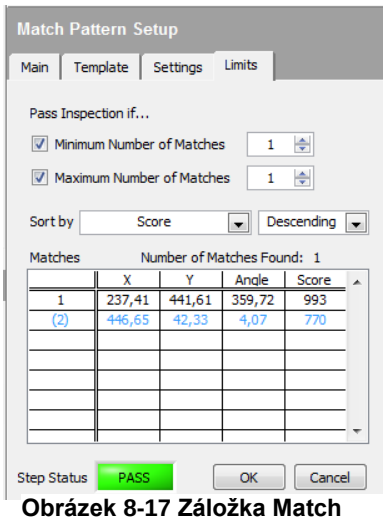
Obrázek 8-15 Záložka Match Patern Setup - Template

Třetí záložka s názvem „Settings“ obsahuje možnosti úpravy kritérií pro nalezení shody. Jsou zde kolonky pro zadání počtu hledaných prvků. Hodnota minimálního skóre, které musí mít nalezený objekt, aby mohl být považován za identický (hodnota 1-1000, přičemž hodnota 1000 má procentuální význam 100%). Další možností je nastavení úrovně kvality vybraného vzoru „Level Coarse“. V poslední části lze nastavit možný úhel natočení nalezeného objektu v porovnání se vzorem (v tomto případě ± 10°).



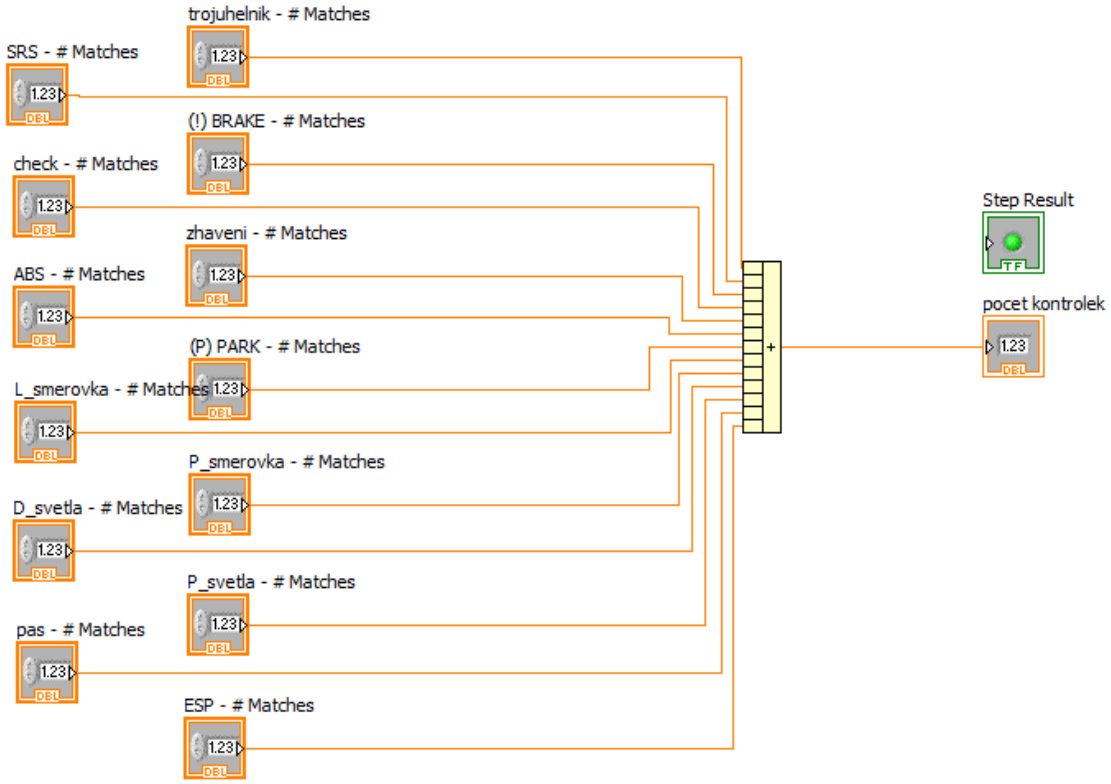
Obrázek 8-16 Záložka Match Patern Setup - Settings

V poslední čtvrté záložce „Limits“ nastavujeme minimální a maximální počet hledaných objektů (v tomto případě vždy 1, vždy máme jen jednu shodnou kontrolku). Dále je zde seznam nalezených objektů s alespoň přibližnou shodou. Jsou zde jejich pozice v souřadnicích X a Y, úhel natočení nalezeného objektu a skóre (procentuální shoda) vůči vzoru. V tomto případě jsou nalezeny dva podobné objekty a je vybrán ten s nejvyšším skóre.



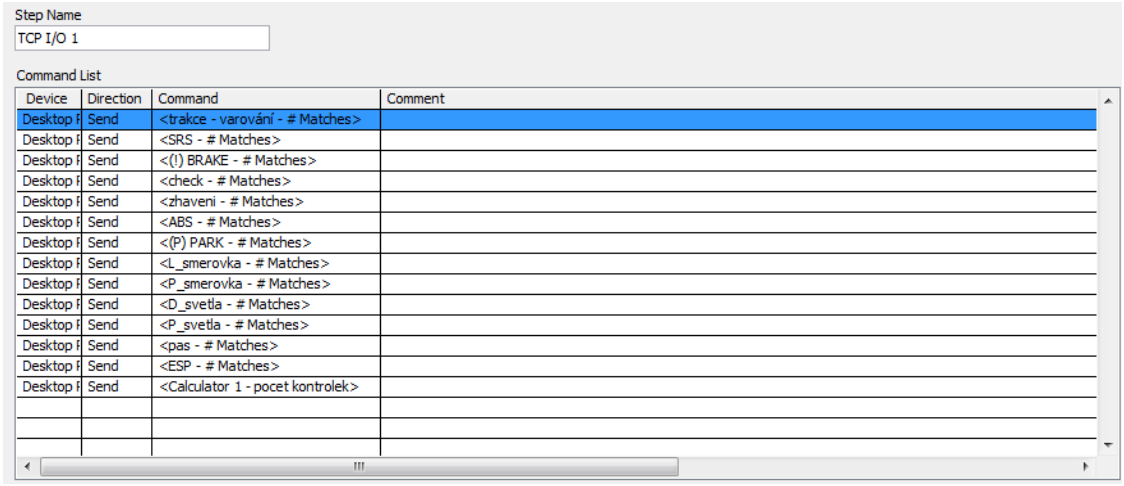
Obrázek 8-17 Záložka Match Patern Setup - Limits

16- *Kalkulátor.* Tento inspekční krok sumarizuje data o jednotlivých kontrolkách. V principu využívá toho, že vstupy nabývají hodnoty 1 nebo 0 (kontrolka svítí nebo nesvítí). Jednotlivé hodnoty sečte a na výstupu máme číslo, které udává celkový počet svítících kontrolek. Tato hodnota se zjišťuje po zapnutí klíčku zapalování před startováním. V tu chvíli se musí v tomto konkrétním případě rozsvítit sedm kontrolek a po startu všechny zhasnout. Naopak v průběhu testů, když je aktivní kontrolka nežádoucí musí mít vždy hodnotu 0. Není-li tomu tak zjišťuje se která kontrolka a proč svítí. Blokové schéma kalkulátoru se vstupy ze všech kontrolek je zobrazeno pod textem.



Obrázek 8-18 Blokové schéma kalkulátoru u testu kontrolek

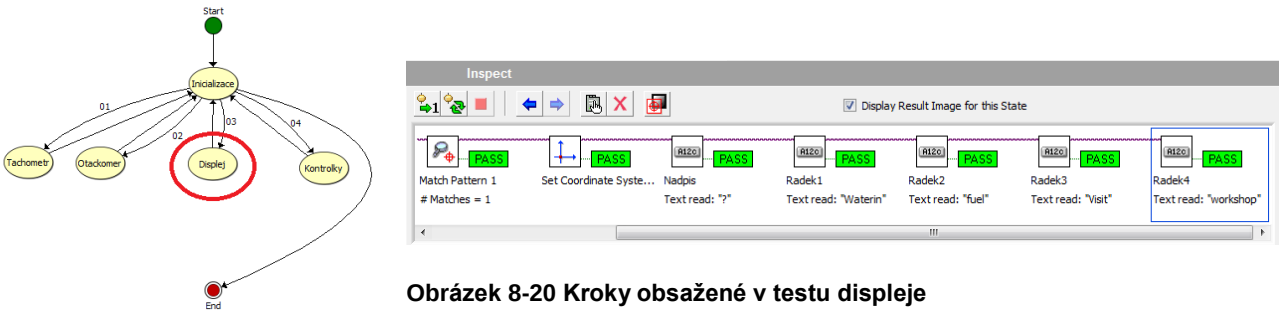
17 – *TCP/IP.* Tento krok je zpracován na stejném principu, jak u testu otáčkoměru v bodě 8.1.2 Rozdílné jsou zde pouze odesílané hodnoty. Odeslán je vždy stav každé kontrolky samostatně a na konci výstup z kalkulátoru (tj. počet aktivních kontrolek). Viz obrázek 8-19.



Obrázek 8-19 Seznam příkazů v kroku odeslání výsledku TCP/IP u testu kontrolek

8.1.4 Test displeje

V případě testování displeje se jedná především o čtení textu, který se zde zobrazuje. Zde se vyhodnocují v principu dva odlišné stavy. První částí je čtení varovných zpráv zobrazovaných na displeji. Druhou částí je čtení textu uživatelského menu a zjišťování, ve které části menu se zrovna nacházíme. Celý test se skládá z několika základních kroků. Prvním je načtení obrazu a jeho úprava pomocí filtrů. Dalšími kroky jsou „Match Pattern“ a „Set Coordinate System“. Tyto dva kroky zajišťují nalezení přesné pozice displeje i v případě posunu kamery. Posledními kroky jsou čtení textu po jednotlivých řádcích. Veškeré výsledky jsou pak opět odeslány do softwaru PROVEtech:TA. Ten provádí snadné vyhodnocení pomocí porovnání nalezeného řetězce s řetězci v souboru, který je pro tyto účely dodáván spolu s kombi přístrojem.



Obrázek 8-20 Kroky obsažené v testu displeje

Obrázek 8-21 Fotografie z průběhu testu

Inspekční kroky

1 – Načtení obrazu

2 – Úprava obrazu

3 – Nalezení referenčního bodu

4 – Nastavení souřadného systému

5-9 – Čtení textu v řádcích

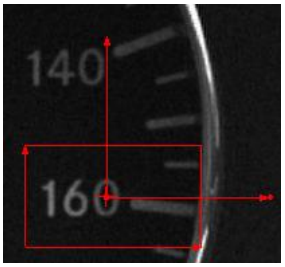
10 – Odeslání výsledku (TCP/IP)

Popis jednotlivých kroků

1 – *Načtení snímku*. Načtení snímku se provádí z kamery připojené přes rozhraní FireWire (IEEE 1394), nebo v tomto případě v režimu offline ze složky s uloženými snímky.

2 – *Filtr obrazu.* Filtr obrazu je v tomto případě prováděn kvůli snazšímu rozpoznání zobrazovaného textu. Celý obraz je převeden na obraz stupňů šedi a zvýšený kontrast. Tím dochází ke zmenšení datového objemu obrazu a zároveň zvýraznění textu.

3 – *Nalezení referenčního bodu.* Referenční bod vyhledáváme proto, abychom mohli nastavit přesnou pozici souřadnic pro čtení textu. Eliminuje se tím případná nepřesnost při změně pozice kamery. V tomto případě slouží jako referenční bod část kruhové stupnice tachometru s číslovkou 160 (viz obr. 8-22).

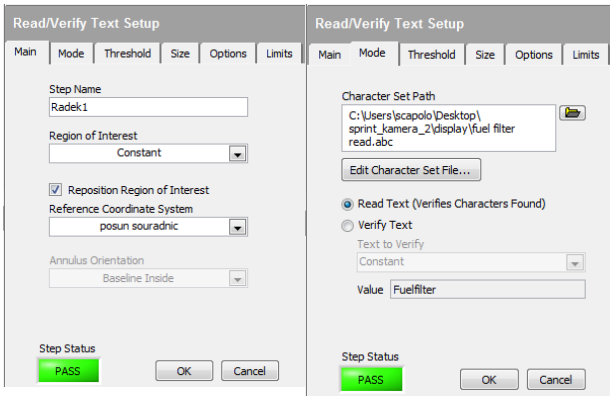


Obrázek 8-22
Referenční bod

4 – *Nastavení souřadného systému.* V tomto případě dochází na základě předem nalezeného referenčního bodu k přepočtu souřadného systému, tak aby nedošlo k deformaci oblasti zájmu pro čtení jednotlivých řádků.

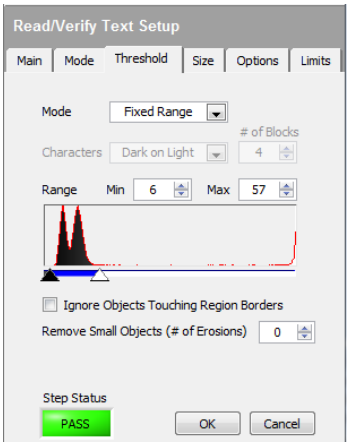
5-9 – *Čtení textu v řádcích.* Samotné čtení textu zobrazeného na displeji je provedeno pomocí kroku „Read/Verify text“. Detailní nastavení tohoto kroku je popsáno níže.

První záložkou v nastavení kroku „Read/Verify text“ je záložka „Main“. Zde se nastavuje název kroku, oblast, ve které se má text vyhledávat a její pozice vzhledem k referenčnímu bodu. Druhá záložka „Mode“ nabízí volbu mezi čtením textu podle symbolů, nebo ověření textu podle předdefinovaného vzoru. V tomto případě se využívá čtení textu.



Obrázek 8-23 Záložky "Main" a "Mode" v kroku Read/Verify text

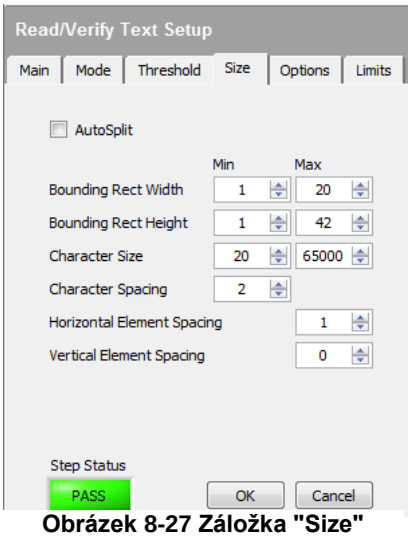
Třetí volbou v panelu nastavení je „Threshold“. V tomto kroku se nastavují prahové hodnoty pro odfiltrování šumu v obraze a zvýraznění zobrazovaného textu. Pod textem je zobrazena ukázka. Na prvním obrázku je základní text načtený kamerou, na druhém je obraz převeden na černobílý a na třetím je výsledný stav pro čtení textu. Toho bylo dosaženo právě nastavením v panelu „Threshold“.



Obrázek 8-24 Záložka "Threshold"

Obrázek 8-25 Ukázky postupné úpravy obrazu a odfiltrování šumu

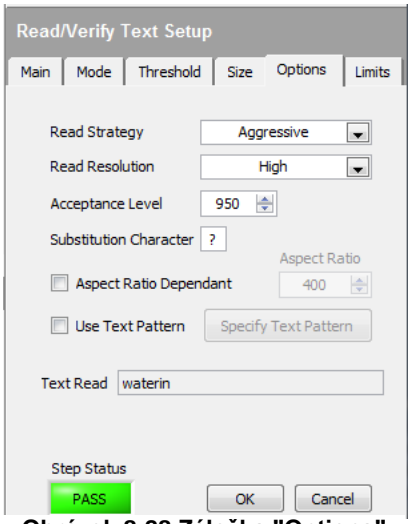
Čtvrtá záložka s názvem „Size“ umožňuje uživateli předem si nastavit velikost a rozmístění polí podle textového fontu, ve kterých se bude nacházet každý konkrétní znak. Příklad je uveden na obrázku 8-27.



Obrázek 8-27 Záložka "Size"

Obrázek 8-26 Ukázka rozdělení textu do polí podle velikosti znaků

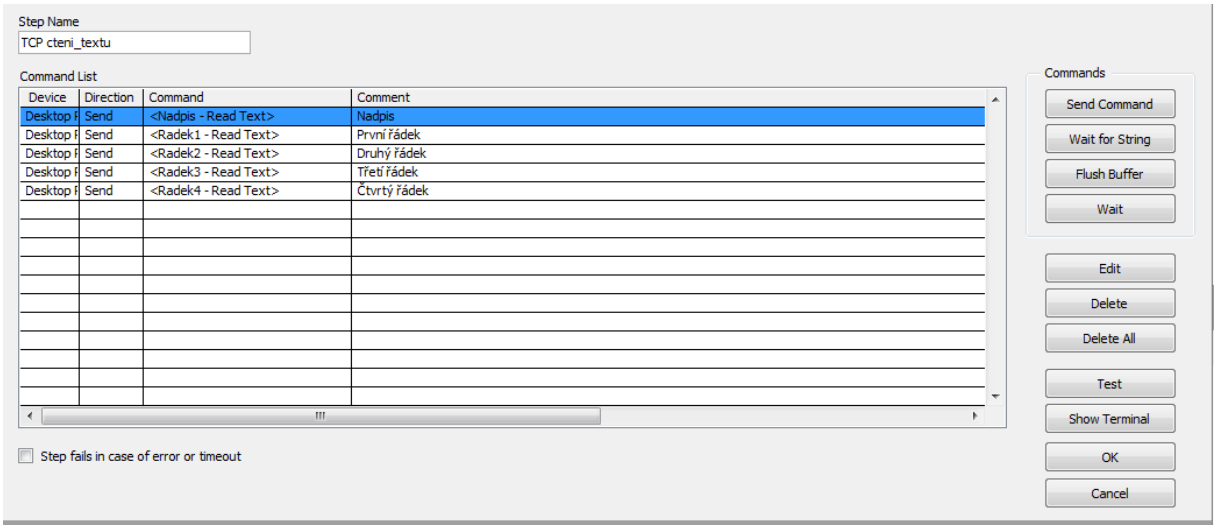
V předposlední záložce „Options“ se nastavuje způsob a meze vyhodnocování. Mimo možností volby jakým způsobem a s jakou kvalitou vyhodnocovat detekované znaky je zde i pole „Acceptance Level“. Číslo nastavené v tomto poli udává, nakolik musí být konkrétní symbol shodný se znakem v abecedě, aby byl platný. V případě tohoto testu je nastavena hodnota 950, která odpovídá shodě 95%.



Obrázek 8-28 Záložka "Options"

V poslední záložce „Limits“ je pak zobrazen seznam jednotlivých nalezených znaků a u každého z nich je i odpovídající procento shody, kterého znak dosáhl. Umožňuje tak nastavit výsledek kroku PASS nebo FAIL podle celkového procenta úspěšnosti.

10 – *Odeslání výsledku (TCP/IP).* Tento krok je zpracován na stejném principu jak u testu otáčkoměru v bodě 8.1.2 a testu kontrolek v bodě 8.1.3 Rozdílné jsou zde pouze odesílané hodnoty. Odeslán je vždy text přečtený na jednotlivých řádcích displeje. Viz obrázek 8-29.



Obrázek 8-29 Seznam příkazů v kroku odeslání výsledku TCP/IP u čtení textu

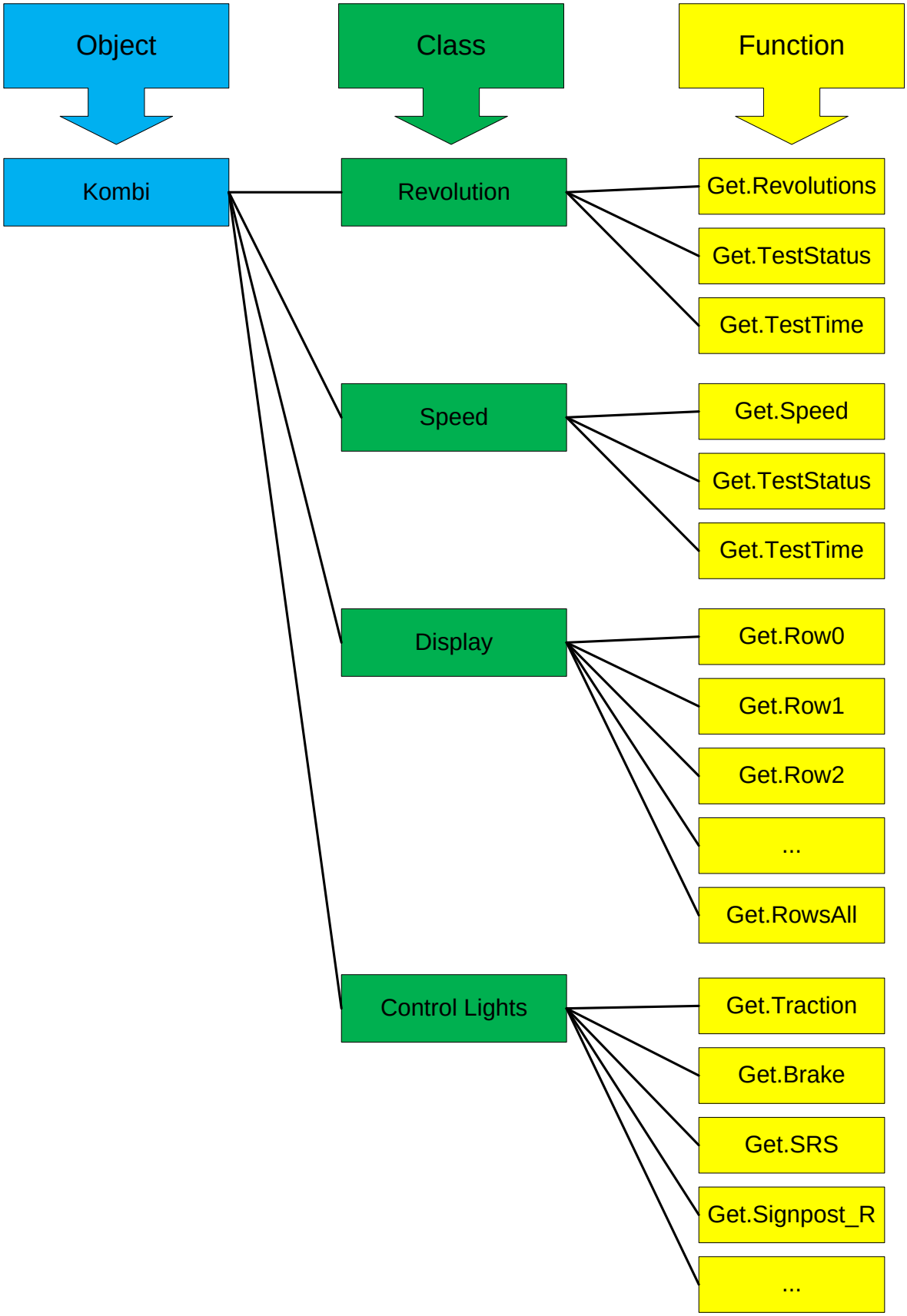
8.2 Realizace v prostředí PROVEtech:TA

Jak již bylo zmíněno dříve software PROVEtech:TA pracuje pouze jako „client“. Z toho vyplívá, že je potřeba vytvořit komunikační knihovny v tomto prostředí. Knihovna je vytvořena podle struktury zobrazené níže (Viz 8.2.1). Obsahuje jeden objekt o čtyřech třídách. Každá třída má pak několik samotných funkcí pro zjištění konkrétní hodnoty.

Výsledkem je tedy poměrně jednoduchá práce při vytváření automatizovaných testů. Například při dotazu na stav kontrolky ABS, zapíše uživatel do testu pouze funkci „Kombi.ControlLights.GetABS“. Po zavolání této funkce, vyšle dotaz na software NI Vision Builder. Ten načte obraz z kamery a následně jej vyhodnotí. Zjištěné hodnoty odešle zpět do PROVEtechu a ten vyhodnotí, zda hodnoty odpovídají stavu nebo ne.

V příloze jsou uvedeny ukázkové automatizované testy společně s vygenerovanými zprávami o průběhu a stavu testu. Ke každé inspekci popsané v bodech 8.1.2 až 8.1.4 je uveden vždy samostatný test.

8.2.1 Struktura knihovny vytvořené v prostředí PROVEtech:TA



9 Závěr

Výsledkem této diplomové práce je zcela automatizovaný systém pro testování kombi přístroje (přístrojové desky). Snadnou rekonfigurací systému ho lze použít i pro testování jiných opticko-elektronických systémů, jako je například autorádio či kontrola průběhu stahování okýnek automobilu a další.

První částí v samotné realizaci je návrh testovacího stavu. V tomto bodě je řešeno připojení samotného kombi přístroje k simulátoru µHil a jeho ovládání pomocí softwaru PROVEtech:TA. Simulátor je s přístrojem propojen přes dvě CAN sběrnice, přičemž na každé sběrnici jsou zasílané signály simulující chování reálné řídící jednotky ECU. Všechny hodnoty zasílaných signálů jsou řízeny právě v prostředí PROVEtech:TA a tím je zajištěna kompletní možnost simulace, ovládání a zpětné čtení stavu z jednoho uživatelského místa.

Další částí je vytvoření systému automatické optické inspekce, skládající se z digitální kamery, softwaru NI Vision Builder for AI a osobního počítače. Tento systém je vytvořen tak, aby po celou dobu testování snímal přístroj, detekoval a vyhodnocoval změny. Ty poté dále předává softwaru PROVEtech a ten vyhodnocuje, zdali skutečný stav přístroje zjištěný kamerou odpovídá požadovanému stavu testování. Oba softwary spolu komunikují přes rozhraní ethernet pomocí protokolu TCP/IP.

Poslední částí je vzájemná komunikace mezi softwary. Pro tyto účel jsem v prostředí PROVEtech:TA vytvořil komunikační knihovnu s jednotlivými příkazy. Pomocí těchto příkazů lze snadno implementovat do stávajících automatizovaných testů novou část pro ověření skutečného stavu. Knihovna obsahuje jeden objekt, čtyři třídy a přes čtyřicet funkcí pro komunikaci.

10 Seznam použité literatury

[1] KUBÍK, Michal. *Testování elektronických systémů automobilu = [Testing of electronic systems in a vehicle]*. Plzeň, 2011. Disertační práce. Západočeská univerzita, Fakulta elektrotechnická.

[2] KUBÍK, Michal. HIL testování a simulace. In: *Elektrotechnika a informatika 2005. Část 2., Elektronika: 6. ročník přehlídky doktorských prací, zámek Nečtiny, 2.-3.11.2005*. V Plzni: Západočeská univerzita, 2005. s. 53-55. ISBN 80-7043-374-4.

[3] PROVEtech:µHiL: PRESS RELEASE June 10, 2009. [online]. [cit. 2012-03-06]. Dostupné z: ttp://www.mbtech-group.com/fileadmin/media/de/pdf/news/2009/electronics_solutions/PM_PROVEtech_HiL_MBtech_EN.pdf

[4] MB-TECHNOLOGY GMBH. *µHiL Dokumentation*: *KTSY1000000.pdf*. Thomas Gahleitner, Martin Zöller. Stutgart, 2010

[5] MB-TECHNOLOGY GMBH. *Lastenheft Kombiinstrument*: *NCV3 Highline*. Georg Eppler. Stutgart, 2004.

[6] MB-TECHNOLOGY GMBH. *Interní dokumenty*. 2012.

[7] Průmyslové kamery. [online]. [cit. 2012-04-21]. Dostupné z: http://www.prumyslove-kamery.cz/

[8] National Instruments: Products and Services. [online]. [cit. 2012-04-18]. Dostupné z:http://search.ni.com/nisearch/app/main/p/bot/no/ap/global/lang/en/pg/1/sn/catnav:mp/q/camera/scope/en%2Ccs/

[9] Hama GmbH & Co KG: Die Passende Lösung. *Copyright © 2001-2012* [online]. [cit. 2012-05-02]. Dostupné z: http://www.hama.de/top-themen/usb-30

[10] ELCOM, a.s.: FireWire versus Gigabit Ethernet. [online]. [cit. 2012-05-02]. Dostupné z: http://www.prumyslove-kamery.cz/clanky-a-aktuality/clanky?pg=305

[11] Camera Link Specifications: Camera Link. PULNIX AMERICA, Inc. [online]. [cit. 2012-05-02]. Dostupné z: http://www.lord-ing.com/web/IMG/pdf/Camera_Link-2.pdf

[12] BASLER Scout: Scout Datasheet (October 2011). [online]. [cit. 2012-03-15]. Dostupné z: http://www.baslerweb.com/media/documents/BAS1109_scout_Web.pdf

[13] COGNEX: VisionPro Vision Software Product Guide. [online]. © 2012 Cognex Corporation. [cit. 2012-04-06]. Dostupné z: http://www.cognex.com/downloads/literaturemain.aspx?id=9964

[14] VisionLab: Dokumentace systému VisionLab. MORAVSKÉ PŘÍSTROJE, a.s. [online]. Aktualizováno: 21.9.2011 [cit. 2012-04-07]. Dostupné z: http://www.mii.cz/art?id=533&cat=147&lang=405

[15] NI Vision: NI Vision Builder for Automated Inspection Tutorial. [online]. August 2009 373379F-01. [cit. 2012-02-15]. Dostupné z: http://www.ni.com/pdf/manuals/373379f.pdf

[16] MB-TECHNOLOGY GMBH. *HiL Testing Guide Basics.doc*: *Version: 01*. Dr. Frank Lattemann. ., 13.01.2010.

11 Přílohy

Příloha 1-1 Ukázkový test tachometru

```
'#Language "WWB-COM"
'#uses "*<WOTLIB>\Dependent\<PRJ>_GENERAL.H"
'#uses "*<WOTLIB>\Independent\GENERAL.H"
'#uses "*Library:\StdLib\dependent\TERM\TERM_TERM.OBJ"
'#uses "*Library:\StdLib\dependent\TERM\TERM_TERM.H"
'#uses "*<WOTLIB>\Independent\TF_TESTFLOW\TF_TESTFLOW_AL.LIB"
'#uses "*<WOTLIB>\Independent\PAR_PARAMETERS\PAR_PARAMETER.LIB"
'#Uses "*Objects\Kombi"

Option UsesAmbiguousError On
Option Explicit

Private Const SPEED_VALUE = 100 'km/h
Private Const SPEED_VALUE_HYSTERESIS = 2 'km/h
Private Const SPEED_CONST = 1000/134 ' puls / 1 km/h

Sub Main

    TF.PreConditionsTestSystem( )

    'Testparameter einstellen
    Par.SetTestParameters("KOMBI", LORS_Kodierdatenabsicherung, "Kombi",
MODE_AUTOMATIC, _PRIO_LOW_TO_MID, AUTHOR_MATEJKA, REVIEWED_YES)

    TF.PreConditionsTest("Set ignition on, Instrument cluster coding")
    TERM.SetEISSimulationState(True)
    TERM.SetIgnSwitchState(TERM_IGN_ON)
    Wait 10

    TF.Action("Set signals for Speed and wait for 1 seconds","Speed")
        System.SetSignal("CAN_M_Tx.ESP_SBC.BS_200h.DVL",
SPEED_VALUE*SPEED_CONST)
        System.SetSignal("CAN_M_Tx.ESP_SBC.BS_200h.DVR",
SPEED_VALUE*SPEED_CONST)

        Wait 1

    TF.Reaction("Check Speed")
         If Kombi.Speed.GetSpeed() >= (SPEED_VALUE -
SPEED_VALUE_HYSTERESIS) And Kombi.Speed.GetSpeed() <= (SPEED_VALUE +
SPEED_VALUE_HYSTERESIS) Then
            TF.SetResult RES_PASS
        Else
            TF.SetResult RES_FAIL
        End If

    TF.PostConditionsTest("Ignition off, Reset signal for Speed")
        System.SetSignal("CAN_M_Tx.ESP_SBC.BS_200h.DVL", 0)
        System.SetSignal("CAN_M_Tx.ESP_SBC.BS_200h.DVR", 0)
        TERM.SetIgnSwitchState(TERM_IGN_OFF)

End Sub
```

Příloha 1-2 Report s výsledky vygenerovaný pro test tachometru

```
********** Set global PreConditions **********
Reset of fault sim and abortion of diagnosis connections
********** End of setting global PostConditions **********


********** Set PreConditions: Set ignition on, Instrument cluster coding
**********
CAN simulation EIS: ON
SF Set :  NO ACTION
CAN simulation EIS: Set IgnSwitchState: IGN_ON, ignition on
SF Set :  NO ACTION

********** TC:Speed **********
Do Action: Set signals for Speed and wait for 1 seconds

Check Reaction: Check Speed
Connection to NI Vision = True
Acquire Image = True
Speedmeter detected = True
Speed Value = 101
Result recieved = True
Connection to NI Vision = True
Acquire Image = True
Speedmeter detected = True
Speed Value = 101
Result recieved = True
check passed

********** Set PostConditions: Ignition off, Reset signal for Speed
**********
CAN simulation EIS: Set IgnSwitchState: IGN_OFF, key plugged
SF Set :  NO ACTION

   Test Result: Passed
```

Příloha 2-1 Ukázkový test čtení textu

```
'#Language "WWB-COM"
'#uses "*<WOTLIB>\Dependent\<PRJ>_GENERAL.H"
'#uses "*<WOTLIB>\Independent\GENERAL.H"
'#uses "*Library:\StdLib\dependent\TERM\TERM_TERM.OBJ"
'#uses "*Library:\StdLib\dependent\TERM\TERM_TERM.H"
'#uses "*<WOTLIB>\Independent\TF_TESTFLOW\TF_TESTFLOW_AL.LIB"
'#uses "*<WOTLIB>\Independent\PAR_PARAMETERS\PAR_PARAMETER.LIB"
'#Uses "*Objects\Kombi"

Option UsesAmbiguousError On
Option Explicit

Private Const READ_TEXT= "Water in fuel Visit workshop"

Sub Main

    TF.PreConditionsTestSystem( )

    'Testparameter einstellen
    Par.SetTestParameters("KOMBI", LORS_Kodierdatenabsicherung, "Kombi",
MODE_AUTOMATIC, _PRIO_LOW_TO_MID, AUTHOR_MATEJKA, REVIEWED_YES)

    TF.PreConditionsTest("Set ignition on, Instrument cluster coding")
    TERM.SetEISSimulationState(True)
    TERM.SetIgnSwitchState(TERM_IGN_ON)
    Wait 10

    TF.Action("Set signal for Display message and wait for 1
seconds","display_message")
        System.SetSignal("CAN_M_Tx.MS.MS_308h.WKS_KL", 1)
        Wait 1

    TF.Reaction("Check display text")
        If StrComp(Kombi.display.Getallrows, READ_TEXT, vbTextCompare) = 0
Then
            TF.SetResult RES_PASS
        Else
            TF.SetResult RES_FAIL
        End If

    TF.PostConditionsTest("Ignition off, Reset signal for Display")
        System.SetSignal("CAN_M_Tx.ESP_SBC.BS_200h.ABS_KL", 0)
        TERM.SetIgnSwitchState(TERM_IGN_OFF)

End Sub
```

Příloha 2-2 Report s výsledky vygenerovaný pro test čtení textu

```
********** Set global PreConditions **********
Reset of fault sim and abortion of diagnosis connections
********** End of setting global PostConditions **********


********** Set PreConditions: Set ignition on, Instrument cluster coding
**********
CAN simulation EIS: ON
SF Set :  NO ACTION
CAN simulation EIS: Set IgnSwitchState: IGN_ON, ignition on
SF Set :  NO ACTION

********** TC:display_message **********
Do Action: Set signal for Display message and wait for 1 seconds

Check Reaction: Check display text
Connection to NI Vision = True
Acquire Image = True
Rows detected = True
Read text Row0 = None
Result recieved = True
Read text Row1 = Water in
Result recieved = True
Read text Row2 = fuel
Result recieved = True
Read text Row3 = Visit
Result recieved = True
Read text Row4 = workshop
Result recieved = True
Read text AllRows = Water in fuel Visit workshop
Result recieved = True
check passed

********** Set PostConditions: Ignition off, Reset signal for Display
**********
CAN simulation EIS: Set IgnSwitchState: IGN_OFF, key plugged
SF Set :  NO ACTION

   Test Result: Passed
```

Příloha 3-1 Ukázkový test stavu kontrolky

```
'#Language "WWB-COM"
'#uses "*<WOTLIB>\Dependent\<PRJ>_GENERAL.H"
'#uses "*<WOTLIB>\Independent\GENERAL.H"
'#uses "*Library:\StdLib\dependent\TERM\TERM_TERM.OBJ"
'#uses "*Library:\StdLib\dependent\TERM\TERM_TERM.H"
'#uses "*<WOTLIB>\Independent\TF_TESTFLOW\TF_TESTFLOW_AL.LIB"
'#uses "*<WOTLIB>\Independent\PAR_PARAMETERS\PAR_PARAMETER.LIB"
'#Uses "*Objects\Kombi"

Option UsesAmbiguousError On
Option Explicit

Sub Main

    TF.PreConditionsTestSystem( )

    'Testparameter einstellen
    Par.SetTestParameters("KOMBI", LORS_Kodierdatenabsicherung, "Kombi",
MODE_AUTOMATIC, _PRIO_LOW_TO_MID, AUTHOR_MATEJKA, REVIEWED_YES)

    TF.PreConditionsTest("Set ignition on, Instrument cluster coding")
    TERM.SetEISSimulationState(True)
    TERM.SetIgnSwitchState(TERM_IGN_ON)
    Wait 10

    TF.Action("Set signal for hand brake and wait for 1
seconds","HAND_BRAKE_LAMP")
        System.SetSignal("CAN_BD_Tx.SAM.SAM_A1.HAS_KL", 1)
        Wait 1

    TF.Reaction("Check hand brake lamp")
        If Kombi.ControlLight.GetBrake() Then
            TF.SetResult RES_PASS
        Else
            TF.SetResult RES_FAIL
        End If

    TF.PostConditionsTest("Ignition off, Reset signal for hand brake")
        System.SetSignal("CAN_BD_Tx.SAM.SAM_A1.HAS_KL", 0)
        TERM.SetIgnSwitchState(TERM_IGN_OFF)

End Sub
```

Příloha 3-2 Report s výsledky vygenerovaný pro test kontrolky

```
********** Set global PreConditions **********
Reset of fault sim and abortion of diagnosis connections
********** End of setting global PostConditions **********


********** Set PreConditions: Set ignition on, Instrument cluster coding
**********
CAN simulation EIS: ON
SF Set :  NO ACTION
CAN simulation EIS: Set IgnSwitchState: IGN_ON, ignition on
SF Set :  NO ACTION

********** TC:HAND_BRAKE_LAMP **********
Do Action: Set signal for hand brake and wait for 1 seconds

Check Reaction: Check hand brake lamp
Connection to NI Vision = True
Acquire Image = True
Break control light detected = True
Result recieved = True
check passed

********** Set PostConditions: Ignition off, Reset signal for hand brake
**********
CAN simulation EIS: Set IgnSwitchState: IGN_OFF, key plugged
SF Set :  NO ACTION

   Test Result: Passed
```