

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

NÁVRH TESTOVACÍHO PRACOVISTĚ PRO AUTOMATIZOVANÉ TESTOVÁNÍ ELEKTRONICKÉ ŘÍDICÍ JEDNOTKY

DESIGN OF TEST ENVIRONMENT FOR AUTOMATIC TESTING OF ELECTRONIC CONTROL
UNIT

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. FILIP DALECKÝ

VEDOUcí PRÁCE
SUPERVISOR

Ing. JIŘÍ KOVÁŘ, Ph.D.

BRNO 2015

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2014/2015

ZADÁNÍ DIPLOMOVÉ PRÁCE

student(ka): Bc. Filip Dalecký

který/která studuje v **magisterském navazujícím studijním programu**

obor: **Aplikovaná informatika a řízení (3902T001)**

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

Návrh testovacího pracoviště pro automatizované testování elektronické řídicí jednotky

v anglickém jazyce:

Design of test environment for automatic testing of electronic control unit

Stručná charakteristika problematiky úkolu:

Práce se zabývá návrhem a realizací části automatizovaného testování elektronické řídicí jednotky při její samotné výrobě. Řídicí jednotka je vyráběna společností UNIS, a.s. a je použita jako řídicí a výkonová elektronika pro hydraulický agregát letadla. V rámci vypracování diplomové práce se řešitel bude podílet na tvorbě automatizovaného testovacího pracoviště včetně vzdáleného ovládní jeho jednotlivých částí, realizace testovacích procedur, které budou sloužit k otestování řídicích jednotek při jejich výrobě a vygenerování protokolů o testování. Jako výchozí platforma pro realizaci testovacího pracoviště bude použit HW a SW od společnosti National Instruments.

Cíle diplomové práce:

1. Teoreticky popsat typické vlastnosti, výhody a nevýhody automatického testování elektroniky.
2. Popsat navržené testovací pracoviště a objasnit princip testování.
3. Návrh a realizace komunikačních rozhraní mezi použitými přístroji pro sběr dat a nastavování parametrů zařízení s využitím nástrojů od National Instruments.
4. Definovat základní sadu výrobních testů pro ověření funkce elektronické řídicí jednotky v rámci její výroby.
5. Návrh a realizace uživatelského rozhraní pro operátora výroby s využitím nástrojů od National Instruments.
6. Praktické ověření realizovaného testovacího pracoviště a navržených testovacích procedur při výrobě elektronických řídicích jednotek.

Seznam odborné literatury:

[1] Morris, A.S., Measurement and Instrumentation: Theory and Application, SBN-10: 0123819601

Vedoucí diplomové práce: Ing. Jiří Kovář, Ph.D.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2014/2015.

V Brně, dne 21.11.2014

L.S.

Ing. Jan Roupec, Ph.D.
Ředitel ústavu

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

ABSTRAKT

Tato diplomová práce se zabývá návrhem a realizací části automatizovaného testovacího pracoviště pro výstupní testování elektronické řídicí jednotky při její samotné výrobě. Elektronická řídicí jednotka je vyráběna společností UNIS, a.s. a je použita jako řídicí a výkonová elektronika pro hydraulický agregát. V teoretické části se práce zabývá problematikou vývojem a automatickým testováním elektroniky v oblasti letectví. V praktické části je popsáno navrhované automatické testovací pracoviště a implementace testovacího softwaru za použití hardwarových i softwarových prostředků od společnosti National Instruments. Hlavními platformami pro tvorbu softwaru jsou LabVIEW a TestStand. Vytvořené testovací pracoviště je nakonec ověřeno otestováním několika řídicích jednotek.

ABSTRACT

This master's thesis deals with the design and implementation of part of test environment for automatic testing of electronic control unit in the production. The electronic control unit is manufactured by UNIS, a.s. and is used as a control and power electronics for the hydraulic unit. The theoretical part of the thesis deals with the development and automatic testing of electronics in the field of aerospace engineering. The practical part describes the proposed automatic test environment and implementation of testing software using hardware and software products by National Instruments. The main platforms used for software development are LabVIEW and TestStand. Created test environment is finally verified by testing several control units.

KLÍČOVÁ SLOVA

LabVIEW, TestStand, testovací pracoviště, automatické testování, elektronická řídicí jednotka.

KEYWORDS

LabVIEW, TestStand, test environment, automated testing, electronic control unit.

PROHLÁŠENÍ O ORIGINALITĚ

Prohlašuji, že jsem diplomovou práci vypracoval samostatně dle pokynů vedoucího a s použitím uvedené odborné literatury.

V Brně, dne 29. 5. 2015

BIBLIOGRAFICKÁ CITACE

DALECKÝ, F. *Návrh testovacího pracoviště pro automatizované testování elektronické řídicí jednotky*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2015. 83 s. Vedoucí diplomové práce Ing. Jirí Kovář, Ph.D.

PODĚKOVÁNÍ

Rád bych poděkoval svému vedoucímu práce Ing. Jiřímu Kovářovi, Ph.D. za odborné vedení, jeho ochotu a čas. Dále bych chtěl poděkovat společnosti UNIS, a.s. za poskytnutí příležitosti vypracovat u nich tuto diplomovou práci. Za společnost UNIS, a.s. bych chtěl poděkovat především Ing. Tomáši Hejčovi za jeho cenné rady a připomínky k práci. V neposlední řadě děkuji rodině a svým blízkým za podporu při vypracovávání této diplomové práce.

OBSAH

1	Úvod	13
2	Motivace a cíl práce	15
3	Automatické testování elektroniky	17
3.1	Životní cyklus vývoje systému	17
3.2	Automatické testovací pracoviště	19
3.3	Způsoby automatického testování elektroniky	20
3.4	Testování elektroniky v letectví	22
4	Testované zařízení	25
5	Výběr výstupních testů	27
5.1	Sada testovacích případů	27
5.2	Sada testovacích procedur	27
6	Testovací pracoviště	29
6.1	Popis testovacího pracoviště	29
6.1.1	Hardwarové prostředky	30
6.1.2	Softwarové prostředky	35
7	Implementace	37
7.1	Komunikace s teplotní a klimatickou komorou	37
7.1.1	Popis komunikačních protokolů	37
7.1.2	Knihovna pro obsluhu teplotní a klimatické komory	39
7.2	Komunikace s elektronickou řídicí jednotkou	42
7.2.1	Popis protokolu CANaerospace	43
7.2.2	Knihovna pro komunikaci s řídicí jednotkou	45
7.3	Komunikace s osciloskopem	46
7.4	Koncepce a realizace testovacího softwaru	47
7.5	Realizace vybraných testů a nastavení přístrojů	50
7.6	Tvorba testovacích sekvencí v prostředí TestStand	55
7.7	Uživatelské rozhraní pro obsluhu	58
7.8	Generování testovacího protokolu	59
8	Praktické ověření testovacího pracoviště a diskuze	63
9	Závěr	65
	Seznam použité literatury	67
	Seznam zkratk	71
	Seznam obrázků	73
	Seznam tabulek	75
	Seznam příloh	77
	Příloha A – Program pro vytvoření testovacího protokolu	79
	Příloha B – Testovací protokol	81
	Příloha C – Datový nosič	83

1 ÚVOD

Současným trendem v průmyslu je nahrazování lidské pracovní síly automatizovanými systémy. Tyto systémy napomáhají k zefektivnění jednotlivých fází tvořící proces od počátečního vývoje produktu až po jeho předání koncovému zákazníkovi. Nedílnou fází vývoje a následné výroby produktu je jeho testování. Díky testování je ověřeno, zda produkt splňuje požadavky zákazníka a také jestli vyhovuje normám, podle kterých je vyvíjen. Zavedením automatického testování do výroby lze dosáhnout výrazně lepších výrobních časů, než kdyby bylo testování prováděno manuálně personálem. Odstraněním lidského faktoru z tohoto procesu je z pravidla dosaženo i nižší chybovosti při opakovaném vykonávání testů.

Tato diplomová práce se ve své teoretické části zabývá popisem vývoje zařízení určeného do leteckého odvětví. Jelikož se jedná o zařízení, u kterých je kladen velký důraz na bezpečnost a spolehlivost, jejich vývoj podléhá mezinárodně uznávaným normám pro vývoj leteckých systémů a to jak po softwarové, tak hardwarové stránce. V další části této diplomové práce jsou popsány vybrané způsoby automatického testování elektroniky. Testování elektronických zařízení v letectví je velmi náročné, protože tyto zařízení musí být schopny zajišťovat bezchybnou funkci i ve velmi nepříznivých podmínkách, proto musí být testovací pracoviště natolik robustní, aby bylo takových podmínek schopno dosáhnout.

První fází vývoje testovacího pracoviště je soupis testovacích případů a procedur. Ty popisují, co všechno bude potřeba testovat a jakým způsobem budou testy prováděny. Podle tohoto dokumentu je již možné vytvářet a naprogramovat testovací sekvence. Praktická část je zaměřena na popis koncepce a realizace testovacího softwaru a komunikačních rozhraní mezi zařízeními, jež jsou součástí testovacího pracoviště. Součástí této práce je také praktické ověření testovacího pracoviště a vykonání několika reálných testů.

S nástupem moderních technologií je možné vytvářet plně automatizované testovací pracoviště, které postupně nahradí neefektivní pracoviště s manuální obsluhou. Současné technologie umožňují dříve složité analogové obvody nahradit výkonným softwarem pro měření, zpracování dat a zpracování signálů. Vše je pak řízeno z počítače a vzniká tak virtuální měřicí přístroj. Použitím virtuální instrumentace odpadá složitý návrh elektroniky a její ladění. Další výhodou je možnost využití jednoho zařízení pro více aplikací, většinou stačí naprogramovat jiný software. Široké uplatnění lze nalézt v průmyslu, kde průmyslové počítače využívají stejné platformy a komunikační protokoly jako osobní počítače. Tím je poté zaručen bezproblémový převod aplikace z vývojových laboratoří přímo do průmyslové výroby.

Moderní měřicí a testovací systémy umožňují přesně řídit časování toku dat. Takové systémy se nazývají Real-Time měřicí systémy a jsou tvořeny velmi výkonným hardwarem s Real-Time operačním systémem (RTOS). Právě tyto technologie popsané výše jsou v rámci této diplomové práce využity při vývoji testovacího pracoviště. Díky modularitě použitého systému a virtuální instrumentaci lze toto pracoviště snadno modifikovat, vylepšovat nebo přizpůsobit pro testování dalších zařízení.

2 MOTIVACE A CÍL PRÁCE

Společnost UNIS, a.s. se zabývá vývojem a výrobou zařízení pro letecké aplikace. Pokrývá celý životní cyklus vývoje zařízení včetně jeho výroby podle standardů platných pro letecký průmysl.

Společnosti UNIS, a.s. rovněž disponuje moderním modulárním systémem pro testování elektronických zařízení. Tento systém se skládá z hardwarových a softwarových prostředků od společnosti National Instruments. Tato diplomová práce se zabývá návrhem a realizací automatizovaného testovacího pracoviště pro elektronickou řídicí jednotku za použití moderních přístupů k testování.

Cílem této práce v její teoretické části je seznámení se s problematikou automatických testovacích systémů a popsat specifika spojená s testováním elektroniky v oboru letectví. Dílčí cíle praktické části této diplomové práce zahrnují návrh a realizaci komunikace mezi zařízeními, které tvoří testovací pracoviště, realizaci testovacího softwaru a také nástroj pro generování protokolu z testování. Funkčnost testovacího pracoviště je ověřena testem s reálnými řídicími jednotkami.

Vzhledem ke vzrůstající komplexnosti elektronických zařízení bude automatické testování stále nabývat na významu. Důležitým aspektem při pořizování systému pro automatické testování je množství vyráběných kusů zařízení, protože pořizovací náklady na takový systém jsou značné a vyplatí se až od určitého objemu výroby. Tyto vysoké pořizovací náklady může částečně kompenzovat použití modulárních, otevřených systémů, které svoji víceúčelovostí umožňují testovat více typů zařízení a také je lze poměrně jednoduše modernizovat výměnou nebo přidáním nových modulů, které tvoří automatický testovací systém.

3 AUTOMATICKÉ TESTOVÁNÍ ELEKTRONIKY

Testování technických zařízení je soubor činností, které mají za cíl zjistit, zda testovaný objekt splňuje předem definované vlastnosti a to jak po hardwarové nebo softwarové stránce. Takový proces je nedílnou součástí každého vývoje výsledného produktu. K ověření předem stanovených vlastností již vyvinuté elektroniky hraje automatické testování velmi důležitou roli. Automatické testovací systémy umožňují testovat hardware a software s takovou rychlostí, jaké by při manuálním testování nebylo možné dosáhnout. Čas personálu tvoří nezanedbatelnou položku v nákladech při sériové výrobě elektronických zařízení. Proto bývá častým cílem firem nahrazovat procesy ve výrobě, které byly vykonávány lidmi, automatizovanými systémy. A to tak, aby byl kladen důraz na kvalitu a konkurenceschopnost daného výrobku.

3.1 Životní cyklus vývoje systému

Pro zajištění bezpečnosti a spolehlivosti vestavěných systémů v letadlech, úřady jako např. *European Aviation Safety Agency*, *UK Civil Aviation Authority*, *US Federal Aviation Administration* vyžadují, aby byly tyto systémy vyvíjeny na základě uznávaných mezinárodních standardů. V oblasti civilního letectví to jsou hlavně normy RTCA DO-254 (*Design Assurance Guidance for Airborne Electronic Hardware*) [3] a RTCA DO-178 (*Software Consideration in Airborne Systems and Equipment Certification*) [4]. Jak z názvů napovídá, norma RTCA DO-254 je zaměřena na vývoj hardwarových komponent a norma RTCA DO-178 na vývoj softwaru.

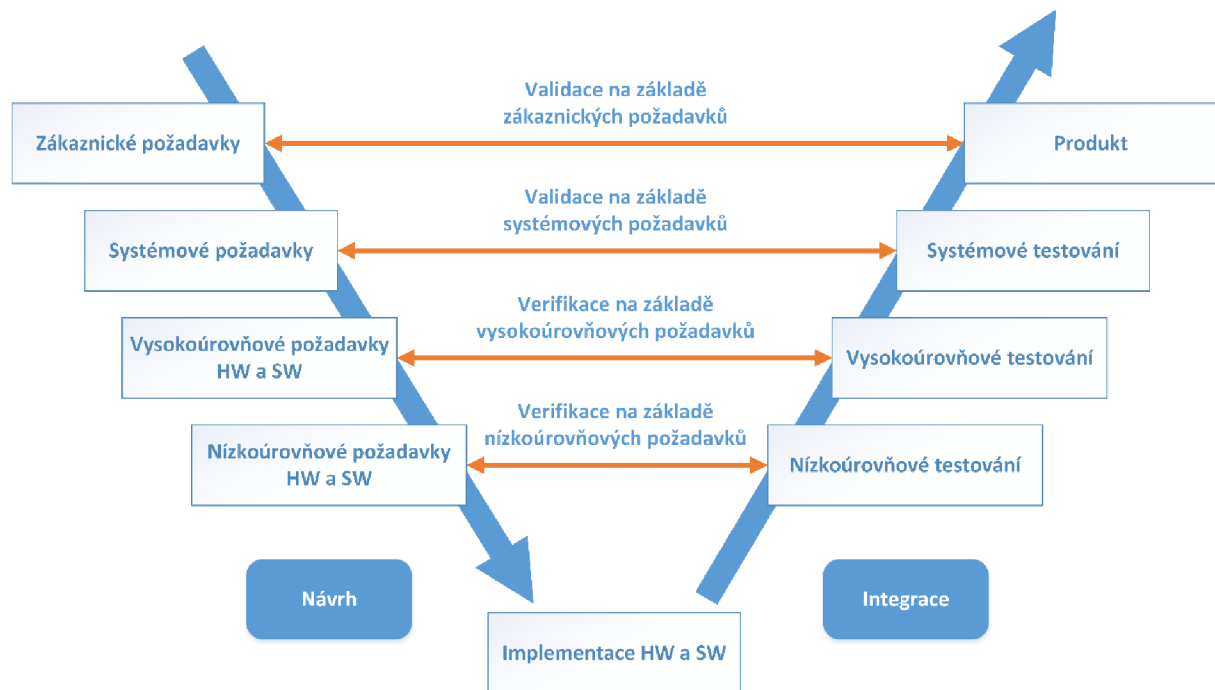
Normy DO-254 a DO-178 nepředepisují preferovaný model životního cyklu systému, ani neurčují přesnou strukturu pro způsob organizace projektu, ale spíše obecně popisují jednotlivé procesy, které jsou zahrnuty ve většině modelů. Normy DO-254 a DO-178 nedefinují, jak mají vypadat jednotlivé dokumenty, ale naznačují obecná pravidla pro jejich psaní a jaké vazby se mezi nimi mají dodržovat. Vybraný model životního cyklu každého projektu by měl být zvolen na základě uspořádání procesů a činností stanovených atributy projektu. Existuje několik typů modelů životního cyklu projektu, jsou jimi např. vodopádový model, V-model, spirálový model, iterativní model a další. Každý z modelů má své výhody i nevýhody a záleží především na tom, jaký model je zavedený v dané společnosti.

Hojně využívaným modelem pro řízení projektů je tzv. V-model [5]. V-model může být považován za rozšíření vodopádového modelu [6]. Vodopádový model je sekvenční vývojový proces, ve kterém je nahlíženo na vývoj jako na neustále svažující se tok jednotlivými fázemi projektu, konkrétně analýzou požadavku, návrhem, implementací, testováním a údržby. V-model se také někdy nazývá „V&V model“, protože klade důraz na *verifikaci* a *validaci* systému v jednotlivých fázích vývoje projektu. Další výhodou tohoto modelu je, že lze započít následující fázi projektu, aniž by byla kompletně dokončena předchozí fáze. Jednou z mnoha definic [7] těchto dvou termínů je:

- *Validace* – ujištění, zda produkt během nebo na konci vývojového procesu splňuje dané požadavky ze strany zákazníka. Jinými slovy: „Vytváříme správný produkt?“
- *Verifikace* – ujištění, zda produkt v dané fázi vývoje splňuje podmínky stanovené na počátku vývoje. Jinak řečeno: „Vytváříme produkt správně?“

Následující diagram (*Obr. 1*) ukazuje, že V-model je postaven na podobných činnostech jako vodopádový model. Ovšem na rozdíl od vodopádového modelu, který postupuje stále dolu,

V-model začne po implementační části stoupat nahoru. Testování produktu je plánováno souběžně s odpovídající fází vývoje.



Obr. 1 Schéma V-modelu.

Vodorovná osa v diagramu vyjadřuje časovou osu resp. fázi, ve které se nachází projekt. Svislá osa diagramu popisuje úroveň abstrakce, čili v jaké hloubce se projekt nachází (v klesající části je to od prvotní specifikace až po implementaci a v rostoucí části diagramu od implementace až k předání produktu zákazníkovi). Takové uspořádání procesů a činností usnadňuje sledovatelnost požadavků a také tvorbu dokumentů pro případnou certifikaci. V každé fázi vývoje jsou tvořeny tzv. požadavky [8][9]. Požadavky je možné rozdělit do několika úrovní. První úrovní jsou *zákaznické požadavky*, kde jsou specifikovány zákaznickovy představy o požadovaném produktu. Tyto požadavky jsou následně hierarchicky rozděleny do dalších tří úrovní a to na *systémové*, *vysokoúrovňové* a *nízkoúrovňové požadavky*. Celkem jsou tedy součástí V-modelu čtyři úrovně požadavků plus jedna úroveň představující implementaci.

- *Zákaznické požadavky* – jedná se o první krok vývojového procesu. Jsou nashromážděny základní požadavky na systém zjištěním potřeb zákazníka. Tato fáze má za cíl stanovit, co by měl ideální systém vykonávat z pohledu člověka nebo jiného zařízení, jenž bude tento systém obsluhovat. Nicméně není ještě dáno, jak bude software a hardware konkrétně navržen. Na základě tohoto kroku je vygenerován dokument se zákaznickými požadavky.
- *Systémové požadavky* - v této fázi je cílem analyzovat a technicky porozumět požadovanému systému. Jsou navrženy možnosti a techniky, kterými by mohly být zákaznickovy požadavky implementovány. Dojde k rozdělení požadavků pro vývoj hardwaru a softwaru. Pokud jsou některé požadavky nesplnitelné, je zákazník o této skutečnosti informován a po vzájemné diskuzi dokument s požadavky upraven. Systémové požadavky jsou sepsány v systémové specifikaci.
- *Vysokoúrovňové požadavky* – v tomto kroku jsou od sebe rozděleny softwarové a hardwarové požadavky. Výsledkem této činnosti jsou tedy dokumenty s vysokoúrovňovými požadavky pro software (DO-178) a hardware (DO-254), které jsou jednoznačně identifikovány a trasovány na systémové požadavky, ze kterých vycházejí.

- *Nízkoúrovňové požadavky* – v této fázi je nutné přesně navrhnout a specifikovat funkci jednotlivých částí systému. Nízkoúrovňové požadavky vycházejí z vysokoúrovňových a jsou na rozdíl do nich velmi podrobné a vyjadřují přesný popis tak, aby se návrh mohl předat vývojářům softwaru nebo hardwaru. Každý požadavek je opět jednoznačně identifikován a trasován na příslušný vysokoúrovňový požadavek, ze kterého byl vytvořen.
- *Implementace* – V případě vývoje software dochází na základě nízkourovňových požadavků k tvorbě zdrojového kódu. V případě vývoje hardwaru se jedná o výrobu plošných spojů. Veškeré úkony nad rámec požadavků jsou zakázány a označují se za chybu.
- *Nízkoúrovňové testování* – nízkourovňové testy jsou vytvořeny na základě nízkourovňových požadavků, nikoliv samotného zdrojového kódu. Tímto je zaručena nezávislost. V opačném případě by mohlo dojít k zanesení chyby. Tyto testy jsou značně rozsáhlé a testují se všechny možné případy pro otestování robustnosti vytvořeného software a hardware.
- *Vysokoúrovňové testování* – v této fázi dochází k integraci vytvořeného a také otestovaného software a hardware. Následně jsou provedeny vysokoúrovňové testy dle předem vytvořených testovacích případů. Stejně jako v předchozím případě, tyto testy jsou vytvořeny na základě vysokoúrovňových požadavků. Vysokoúrovňové testy zajišťují, že individuálně testované části systému správně fungují i po zapojení s ostatními.
- *Systémové testování* – během těchto testů je vyvinutý software dle DO-178 a hardware dle DO-254 ověřen jako funkční celek. Tyto testy ověřují systém na základě vstupních požadavků zákazníka. Poslední úroveň testů, které se provádějí před předáním produktu zákazníkovi, jsou tedy systémové testy. Tato úroveň testů tak většinou slouží jako výstupní kontrola. Jak vyplývá z podstaty V-modelu, tyto testy jsou navrhovány již ve fázi návrhu samotného systému.
- *Předání produktu* – poslední fází je předání hotového zařízení do rukou zákazníka. Často následuje ještě další testování, tentokrát ovšem na straně zákazníka, aby si ověřil, že zařízení splňuje jeho požadavky a je připravené k nasazení do provozu.

V celém výše uvedeném popisu jednotlivých fází životního cyklu je nutné dodržet nezávislost. Není dovoleno, aby např. stejná osoba psala vysokoúrovňové požadavky, pak tvořila kód a následně si jej na vyšší úrovni testovala. Nalezené pochybení se musí zaznamenat, následně dojde k ověření případného dalšího dopadu, a poté je navrženo řešení. Tato diplomová práce je zaměřena na oblast systémového testování pro automatické výstupní testy elektronické řídicí jednotky, proto budou další kapitoly převážně věnované tomuto tématu. [1]

3.2 Automatické testovací pracoviště

Automatické testovací pracoviště je systém, který je tvořen řídicím počítačem a jedním nebo více automatickými, programovatelnými přístroji. Automatické testovací pracoviště vyžaduje počítač pro zajištění správné funkce, zaznamenávání měření, zpracování naměřených dat a jejich následnou interpretaci ve formě, která je srozumitelná pro obsluhu. Obsluha u automatických testovacích pracovišť přímo neovládá přístroje, ze kterých je pracoviště tvořené, nicméně musí naprogramovat počítač, který všechny tyto operace provádí.

Existuje několik základních typů automatického testovacího pracoviště. Prvním typem je systém navržený a vytvořený na zakázku přímo pro testování jednoho konkrétního zařízení. Toto řešení nabízí vysoký výkon a vysokou spolehlivost, avšak pořizovací náklady jsou velmi vysoké. Tento typ automatického testovacího pracoviště se tedy vyplácí při velkých výrobních sériích.

Dalším typem jsou systémy, které jsou tvořeny počítačem nebo mikroprocesorovým počítačem a jsou využívány pro řízení univerzálně použitelných přístrojů, jako jsou multimetry, osciloskopy, signálové generátory apod. Tyto přístroje jsou pak ovládány jako periferní zařízení počítače. U tohoto systému každá periferie provádí měření na základě příkazů z počítače a ten pak data zpracovává, vyhodnocuje a zobrazuje obsluze. Takové testovací pracoviště je sice stále nákladné na pořízení, nicméně je možné ho poměrně snadno přizpůsobit pro testování jiných zařízení, než jen jednoho konkrétního výrobku. Tím se zkracuje návratnost investice. Tyto systémy se tedy dají využívat ve výrobě i v menších výrobních sériích.

V praxi je možné přizpůsobit pro testování více než jednoho zařízení i prvně zmíněný typ automatického systému a to za použití konceptu modulárního systému. Ten umožňuje měnit jednotlivé měřicí moduly a přehrávat testovací software.

Tyto dva výše zmíněné typy systému zjednodušeně reprezentují extrémní filozofie návrhu automatických testovacích pracovišť. V prvním typu je spousta měřicích modulů a řídicí počítač zabudované do jednoho zařízení a na druhé straně několik samostatně funkčních periférií, které jsou řízeny jedním centrálním počítačem. Běžně jsou používány takové automatické systémy, které se svou filozofií pohybují někde mezi těmito dvěma typy systémů a kombinují výhody obou z nich.

V mnoha případech je použití automatických testovacích systému nezbytné, ať už kvůli rostoucí složitosti zařízení, ale třeba i rychlosti dějů, které je potřeba měřit. Mezi hlavní výhody automatických testovacích systémů patří:

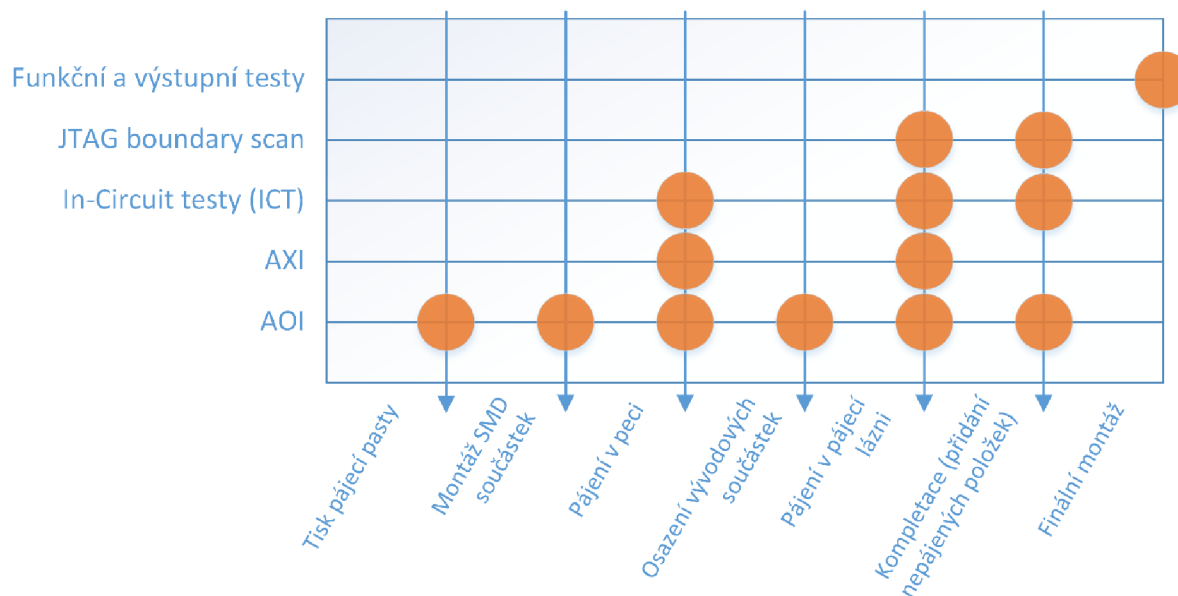
- *Více měření* – všechny měření jsou u automatického testovacího systému prováděny automaticky, což v důsledku umožňuje vykonat o mnoho více měření za jednotku času.
- *Větší přesnost* – při manuálním testování probíhá měření většinou jedenkrát na rozdíl od automatických systémů, kde není problém změřit veličinu hned několikrát a tyto hodnoty např. zprůměrovat a snížit tak chybovost měření.
- *Rychlejší procedury* – u složitějších testovacích procedur, které obnášejí komplikované nastavování testovacího pracoviště, přesné načasování spouštění měřicích sekvencí, výpočty s naměřenými hodnotami apod., manuální testovací pracoviště nemohou v tomto ohledu těm automatickým konkurovat.
- *Omezení lidské účasti na testu* – automaticky probíhající, počítačem řízené testování zabraňuje možnému zkreslování výsledků obsluhou, jejich subjektivnímu posuzování, anebo chybovosti měření způsobené neprecizností obsluhy.

Záporem automatických testovacích systémů je především velikost pořizovacích nákladů, které se mohou vrátit až po několika letech. Ovšem po uplynutí této doby se investice začne vracet a odměnou bude vysoká spolehlivost a levnější výroba produktů. [2]

3.3 Způsoby automatického testování elektroniky

K dispozici je celá řada různých přístupů, které mohou být aplikovány pro automatické testování elektroniky. Každý způsob má své výhody a nevýhody, a najde své uplatnění pro konkrétní aplikace. Nejrozšířenější formy dnes používaných automatických testovacích systémů jsou rozebrány v následujících podkapitolách. Na *Obr. 2* je možné vidět uplatnění vybraných metod v určitých fázích procesu výroby zařízení. Z toho se bude odvíjet i vhodnost metody pro danou fázi

výroby. V dnešní době ve většině případů neposkytne kompletní řešení jen jediná metoda, a proto se jich v pokročilých testovacích systémech využívá hned několik.



Obr. 2 Oblast použití různých kvalitativních testů ve výrobě desek plošných spojů [10].

Automatická optická inspekce (AOI)

Jedná se o vizuální a tudíž bezkontaktní měření, které je velmi flexibilní a vhodné pro téměř všechny fáze výrobního procesu. Kamera automaticky snímá zařízení za účelem najít jakékoliv možné defekty, jako jsou škrábance, rozpojení, zkratky, špatné pájení, chybějící nebo špatně umístěné komponenty. Nevýhodou této metody je možnost detekovat defekty pouze na viditelných místech, malé vizuální rozdíly mohou vést k výskytu pseudodefektů a kontrola pájených míst je omezena pouze na viditelné části. [11][12]

Automatická X-ray inspekce (AXI)

S rostoucí hustotou osazení DPS a příchodem nových technologií k osazování desek nemusí být viditelné všechny pájená místa. V tomto případě je optická inspekce nepoužitelná. Proto přichází v úvahu využití tohoto systému, který je založený na rentgenovém záření. S touto technologií je možné nahlédnout pod jednotlivé komponenty a zkontrolovat tak veškerá pájená místa. Nevýhodou této technologie je vysoká cena, nicméně v mnoha případech je nenahraditelná. [13][14]

In-Circuit testy (ICT)

V případě testování In-Circuit je přiváděn stimul z vnějšku ze speciálního zdroje bezprostředně na testovanou součástku. Při tomto způsobu testování je ale zapotřebí zpravidla značně nákladné testovací zařízení. Tato metoda testuje nejen zkratky, rozpojený obvod, naměřené hodnoty na jednotlivých komponentech, ale také kontroluje funkčnost integrovaných obvodů. I přesto, že ICT je velmi užitečný nástroj, v dnešní době jeho uplatnění klesá z důvodu ubývajícím přístupu k jednotlivým komponentám na desce a to jako důsledek zvyšování hustoty osazení DPS. [2][15]

JTAG boundary scan

Boundary scan je metoda pro testování propojení na DPS nebo dílčích bloků uvnitř integrovaných obvodů. Hlavním cílem při vývoji tohoto způsobu testování bylo zajistit přístup k testovaným

obvodům vhodnou metodou tak, aby mohly být obvody testovány bez použití fyzických testovacích sond, jsou totiž nahrazeny vestavěnými logickými obvody. Dnes je tato metoda široce využívána. [16][17]

Funkční a výstupní testy

Za funkční testování může být považováno jakékoliv testování elektroniky, které ověřuje funkčnost daného zařízení v předem definovaných pracovních podmínkách a stavech. Nejčastěji se tato kategorie testů využívá na systémové úrovni. V případě funkčních testů, kdy zařízení ještě nemusí být plně zkompletováno, mohou být tyto testy kombinovány i s jinými přístupy, jako např. ICT. U výstupních testů nejsou využívány kontrolní vývody na DPS především z toho důvodu, že zařízení je již zkompletované a zakrytované, takže jsou pro testování k dispozici pouze dostupné periferie vycházející z finální podoby daného zařízení [2][18]. Právě výstupnímu testování elektronické řídicí jednotky se zabývá v praktické části tato diplomová práce.

3.4 Testování elektroniky v letectví

Elektronické systémy v letectví musí být schopny zajišťovat bezchybnou funkci i v extrémních podmínkách, kterým jsou letadla vystavována při jejich běžném provozu. V roce 1975 byl společností každého publikován dokument RTCA DO-160 [19]. Od doby založení prošla tato norma několika revizemi až do aktuální revize G. Cílem tohoto dokumentu je nastínit minimální standard pro environmentální testy a testovací procedury pro testování zařízení v letectví a to pro široké spektrum letadel. Účelem těchto zkoušek je poskytnout řízené (laboratorní) prostředky k zjištění výkonových charakteristik leteckých zařízení v podmínkách prostředí podobných těm, které se mohou vyskytnout při jeho provozu. [20]

Norma DO-160 je rozdělena do několika kapitol, které specifikují vlastností jednotlivých složek prostředí, kterými musí testované zařízení bez problému projít.

- *Teplota a nadmořská výška* – popis zkoušek, které testují vliv teploty a nadmořské výšky na systém. Testovací procedury pro tyto zkoušky jsou rozděleny podle toho, do které kategorie spadá navrhované zařízení, jež bude následně montováno do letadla. Při testech zařízení v různých nadmořských výškách jsou zahrnuty i simulace ztráty tlaku v kabině. Dále se testuje např. chlazení za nízkého tlaku nebo odolnost vůči prudkým změnám tlaku. Dalším problémem může být kondenzace vznikající rychlými změnami z nízkých do vysokých teplot anebo vliv teplotní roztažnosti materiálu.
- *Vlhkost* – cílem těchto testů je zkontrolovat odolnost zařízení vůči vysokým koncentracím vlhkosti. Hlavní dopady, které se zde sledují, jsou koroze a změna charakteristik zařízení plynoucí z absorpce vlhkosti. Některé komponenty citlivé na vlhkost proto musí být zapouzdřeny případně jinak ochráněny, aby prošly těmito testy.
- *Odolnost vůči nárazům a rozbití* – nárazové testy mají prověřit, zdali zařízení bude nadále konat správnou funkci po náhlém nárazu, který se může například objevit při rolování na letišti, přistávání nebo průletem turbulencí. Testy proti rozbití ověřují, jestli se zařízení, či jeho komponenty po prudkém nárazu neoddělí. Prudkým nárazem může být například chápáno nouzové přistání. Toto je velmi důležité hlavně u komponent, které se nacházejí v kritických místech jako je palivový systém nebo v místech sloužících k nouzové evakuaci letadla. Po prudkém nárazu by se ze zařízení nebo jeho součástí mohl stát smrtící projektil, který by zajisté napáchal mnoho škody.
- *Vibrace* – zařízení musí vyhovět kladeným požadavkům na odolnost vůči vibracím, jejichž charakter je závislý na konkrétním místě instalace zařízení.
- *Výbušné prostředí* – při těchto testech je zařízení umístěno do prostředí obsahující výbušné plyny. Zařízení musí být schopno pracovat (včetně otáčení knoflíků, stisknutí tlačítek) v takovémto prostředí, aniž by iniciovalo výbuch.

- *Voděodolnost* – tyto testy prověřují odolnost zařízení vůči polití vodou, vhození do vody nebo také vliv kondenzace. Tyto testy nejsou zamýšleny k tomu, aby prověřovaly hermeticky uzavřené zařízení. U hermeticky uzavřených zařízení se dá automaticky předpokládat jejich voděodolnost.
- *Odolnost vůči různým kapalinám* – tyto zkoušky mají za cíl zkontrolovat, jestli materiály použité na konstrukci zařízení odolávají působení kapalin, které se mohou v letadle objevit, od technických kapalin, přes saponáty až po slazené nápoje.
- *Prach a písek* – tyto testy prověřují odolnost zařízení vůči prachu a písku pohybují se vzduchem. Zkoumají se především tyto nepříznivé efekty: průnik prachu do prasklin, do štěrbin, k pohyblivým částem; vznik elektricky vodivých mŕstvků; vznik míst, kde by mohlo docházet k shromažďování vodní páry, což by mohlo způsobit vznik koroze.
- *Plísňe* – při těchto testech je zkoumáno, zdali je materiál použitý v zařízení ovlivněn plísněmi v podmínkách, které jsou pro vznik plísní příznivé, jako je vysoká vlhkost, teplé ovzduší a přítomnost anorganických solí.
- *Solná mlha* – během těchto testů je zařízení vystavováno působení slanému ovzduší nebo slané mlze za jeho normálního provozu. Hlavní důsledky působením solí, které jsou sledovány, je koroze kovů, omezení činnosti pohyblivých dílů jako důsledek usazování solí, ztráta izolace a poškození kontaktů nebo obnažených drátů.
- *Vliv magnetismu* - tyto testy zjišťují vliv magnetického pole zařízení. Test má zaručit, že zařízení může pracovat správně bez interference, která může ovlivnit chod okolních zařízení.
- *Odolnost vůči zásahu bleskem* – zkoušky v této kategorii prověřují schopnost zařízení pracovat po přímém zásahu bleskem a vztahují se na zařízení, která jsou namontována mimo trup letadla (antény, světla, senzory apod.).
- *Námraza* – tyto testy určují výkonnostní charakteristiku zařízení, které musí pracovat i když je vystaveno vlivům námrazy, která vzniká za podmínek, kdy dochází k prudkým změnám teploty, výšky a vlhkosti.
- *Elektrostatický výboj* – cílem těchto zkoušek je zkontrolovat odolnost nebo schopnost zařízení dále vykonávat svou činnost bez trvalého poškození po účinku elektrostatického výboje. Tyto testy jsou prováděny pro všechna zařízení, která jsou přístupná za svého běžného provozu nebo během údržby a oprav.
- *Ohnivzdornost* – v této sekci jsou popsány testovací podmínky a procedury k hořlavosti a ohnivzdornosti. Např. zařízení musí bezproblémově fungovat po dobu prvních pěti minut požáru a dále si udržet bezpečnostní funkce minimálně 15 minut po vzniku požáru. U zařízení, která pracují nějakým způsobem s kapalinami, musí být zabráněno úniku kapaliny, která by mohla podporovat hoření.
- Další sekce popisují zkoušky, které se týkají napájení zařízení, schopnost zachování funkce při napěťových špičkách nebo náchylnost na působení audio frekvenčních, radio frekvenčních vlivů.

Většina těchto testů se provádí ve fázi vývoje a nejsou součástí výstupních testů každého zařízení. Sada konkrétních testů dle kapitol normy DO-160 pro dané zařízení je specifikována na základě prostředí a využití pro konkrétní zařízení. Tyto náročné testy, v některých případech dokonce destruktivní, se tedy většinou provádí jednou, kdy je dokázáno, že zařízení vyhovuje či nevyhovuje standardům. Pokud vyhovuje, může zařízení postoupit do výroby. [19]

4 TESTOVANÉ ZAŘÍZENÍ

Testovaným zařízením v rámci této diplomové práce je elektronická řídicí jednotka. Konkrétně se jedná o řídicí jednotku pro hydraulický agregát, který je určen jako zdroj tlakové kapaliny pro hydraulickou soustavu letounu. Testovaná elektronická řídicí jednotka je univerzální a může být použita i pro jiné aplikace, např. v automobilovém průmyslu. Řídicí jednotka byla vyvinuta vývojovým oddělením společnosti UNIS, a.s. divizí leteckého a pokročilého řízení.

V samotné zástavbě letadla je řídicí jednotka mechanicky spojena s BLDC motorem, pomocí kterého se dosahuje potřebného tlaku v hydraulické soustavě letounu. Jednotka je napojena na napájecí síť letadla 28 V. Tlakový snímač umístěný v hydraulické soustavě reaguje na pokles tlaku tak, že spíná napájení jednotky. Jakmile dojde k sepnutí napájení, dochází k postupnému natlakování hydraulického okruhu na potřebný tlak. Tento tlakový okruh se primárně používá k manipulaci s podvozkem letadla. Pro laboratorní nasimulování hydraulické soustavy letounu bylo využito tlakové nádoby s měřičem tlaku.

Hlavní funkcí řídicí jednotky tedy je rozběhnutí elektromotoru a jeho regulace na konstantní příkon, případně na konstantní otáčky podle nastavených registrů. Po zapnutí postupně provádí řídicí jednotka tyto operace:

- Kontrola napětí a smyslu otáčení
- Rozběh elektromotoru
- Regulace elektromotoru

Řídicí jednotka obsahuje sériové rozhraní RS232 určené pro servisní účely. Toto rozhraní slouží k ovládání jednotky, pro výstup diagnostických údajů a modifikaci vybraných parametrů. Komunikace probíhá pomocí protokolu CANaerospace.

Jednotka se dodává v sestavě jako dva komponenty, jedna je blok elektroniky a druhá je deska plošných spojů s Hallovy sondami. Řídicí jednotka se skládá z těchto částí:

- Zdroj – napájecí obvody
- Řídicí elektronika s komunikací po RS232
- Výkonová elektronika
- Elektronika pro snímání polohy
- Nezávislá paměť pro uložení nastavení a provozních údajů
- Obvod reálného času s baterií
- Signalizace poruchy

Proti poškození zařízení jsou v jednotce implementovány tyto ochranné funkce:

- Počet opakovaných startů
- Příliš velký proud do motoru
- Minimální proud
- Minimální otáčky
- Startovací napětí
- Vypínací napětí
- Tepelná ochrana
- Chybný signál od senzorů
- Test směru rotace motoru

Dalšími parametry elektronické řídicí jednotky jsou:

- Pracovní teplota prostředí: -55 °C až 70 °C
- Nepracovní teplota prostředí: -55 °C až 85 °C
- Rozměry zařízení (výška x šířka): 111 x 114 mm
- Hmotnost: 0,75 kg s chladičem, krytem a bez protikusů konektorů
- Jmenovité napájení: 28 V
- Provozní napětí: 22 V až 30,3 V
- Mezní napětí: 18 V až 32,2 V

5 VÝBĚR VÝSTUPNÍCH TESTŮ

Před samotným vývojem testovacího softwaru byl v rámci této diplomové práce vypracován dokument obsahující testovací případy pro výstupní testy řídicí jednotky při nízké teplotě (-55 °C) a vysoké teplotě (70°C). Testy pro tyto dvě teploty jsou identické, mění se pouze nastavení teploty v tepelné komoře.

5.1 Sada testovacích případů

Testovací případ (často používaný výraz z angl. *test case*) popisuje, jak by mělo zařízení reagovat za daných podmínek a po provedení určité operace. Každý testovací případ je pojmenován a označen unikátním identifikátorem tak, aby jej bylo možné snadno trasovat. Tělo testovacího případu pak tvoří položky vstup, počáteční podmínky, očekávaný výstup, popř. komentář pro doplňující informace. Každý testovací případ by měl být psán tak, aby mohl být otestován samostatně a nemělo by se spoléhat např. na to, že testované zařízení bude ve vždy výchozím stavu, jelikož by mohlo být přenastavené z jiného testu.

Ukázkový příklad jednoho testovacího případu lze vidět níže. Jedná se o testovací případ pro kontrolu komunikace s jednotkou při napájení 24 V. Lze si všimnout, že tento testovací případ je psán obecně a nejsou v něm podrobně popsány všechny kroky, které musí tester vykonat. Testovací případy jsou hlavním podkladem pro tvorbu testovacích procedur.

Příklad testovacího případu

[PTC-33.001] Kontrola komunikace při 24 V

Vstupy:	Odeslat požadavek řídicí jednotce na čtení registru č. 0 (služba RCRS).
Podmínky:	Řídicí jednotka musí být vložena do tepelné komory a zahřáta na 70 °C po dobu 1 hodiny. Napájecí zdroj jednotky musí být zapnutý a nastaven na 24 V.
Očekávané výstupy:	Řídicí jednotka musí odpovědět na požadavek odesláním zprávy s obsahem registru č. 0.
Komentář:	Žádný.

Testovací případy pro výstupní testy řídicí jednotky byly vytvořeny na základě technického zadání (interní dokumentace společnosti UNIS, a.s.). Výstupní testy jednotky jsou tvořeny následujícími testy:

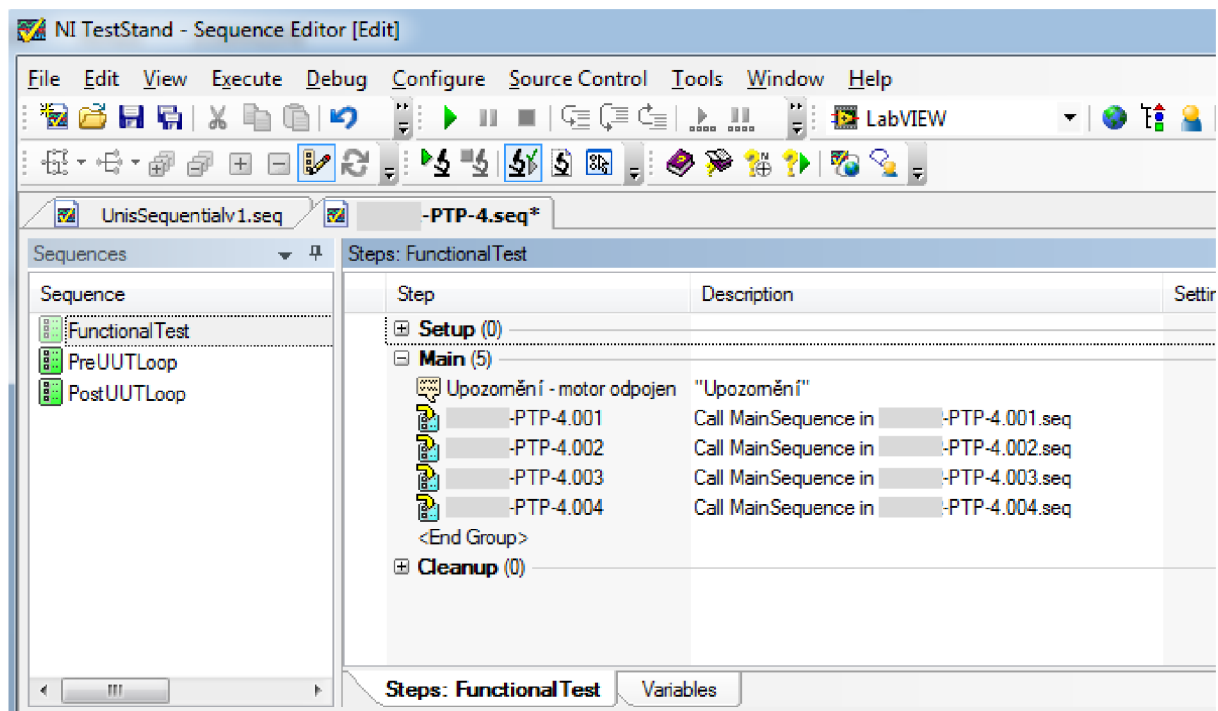
- Kontrola komunikace při 12 V
- Kontrola komunikace při 24 V
- Kontrola komunikace při 32 V
- Kontrola odběru proudu při 24 V
- Kontrola odběru proudu při 32 V
- Kontrola měření napětí při 24 V
- Test přepětové ochrany
- Kontrola data a času
- Kontrola dat v paměti
- Test motoru bez zátěže
- Test motoru se zátěží

5.2 Sada testovacích procedur

Testovací procedura je na rozdíl od testovacího případu podrobnější a popisuje konkrétní kroky, které musí být vykonány pro provedení testovacího případu. V testovací proceduře lze kombinovat

více testovacích případů, které mají stejný charakter a kde se mění např. jen hodnota některé vstupní veličiny. Proto by se při psaní testovacích procedur měl klást důraz na to, aby se v rámci jedné procedury otestovalo více testovacích případů, a tak omezit zbytečné opakování stejných operací. Výsledkem efektivně napsané testovací procedury je uspořený čas a menší opotřebení testovacího hardwaru, jehož některé komponenty mohou být náchylnější na časté používání.

V rámci této diplomové práce byly testovací procedury vytvořeny v prostředí TestStandu a to tak, že testy byly rozděleny do několika skupin (subsekvencí). Testovací procedury byly navrhovány, tak aby při nastavení určitého stavu všech relé v přizpůsobovacím adaptéru byly provedeny všechny testy, které toto nastavení adaptéru vyžadují. Testovací procedury v TestStandu lze vidět na následujícím obrázku *Obr. 3*.



Obr. 3 Testovací procedury v sekvenci vytvořené v TestStandu.

6 TESTOVACÍ PRACOVISTĚ

Testovací pracoviště vychází především z požadavků na něj kladených. Testovací pracoviště musí svými vlastnostmi naplňovat definované požadavky spolu se zachováním udržitelnosti vzhledem k vývoji technologií spojených s testováním elektroniky. Toho může být dosaženo využitím modulárních systémů, jejichž případná modifikace je o poznání snazší než u systému vyráběných na míru. V rámci projektu společnosti UNIS, a.s. výroby řídicí jednotky, byly pořízeny přístroje, zařízení a software pro splnění předem definovaných cílů. Výstupní testy jednotky jsou jedním z dílčích cílů tohoto projektu a jsou obsahem praktické části této diplomové práce.

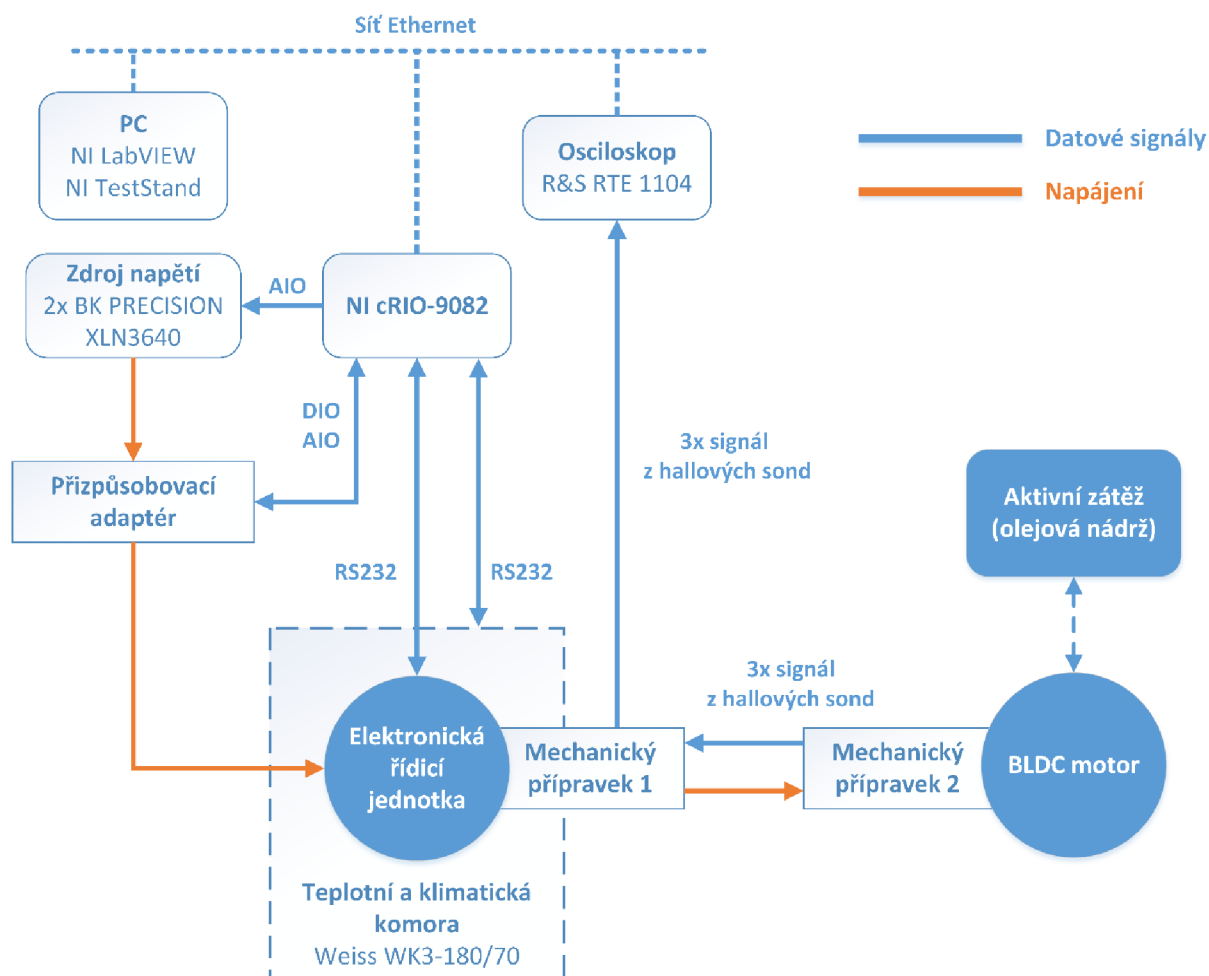
6.1 Popis testovacího pracoviště

Moderní, automatický způsob testování má zajistit zrychlení testovacího procesu při funkčních a výstupních zkouškách, zredukovat prostor, kde by mohlo dojít při testování k chybám a tím i zlepšit konkurenceschopnost výrobku na trhu. Testovací pracoviště je popsáno v následujících bodech:

- Veškeré součásti testovacího pracoviště jsou řízeny z PC pomocí softwaru LabVIEW.
- Testovací software je implementován v LabVIEW a je tvořen dvěma částmi. První část tvoří programy spouštěné z PC a druhou část programy zavedené a spouštěné ze zařízení NI CompactRIO.
- Součástí testovacího pracoviště je přizpůsobovací adaptér, jehož úkolem je napájení řídicí jednotky a současně umožňuje měření odběru proudu jednotkou a to ve dvou rozsazích (0 – 200 mA a 0 – 10 A). Adaptér je ovládán pomocí zařízení NI CompactRIO.
- Dvojice mechanických přípravků sloužící pro signálové a elektrické propojení řídicí jednotky s motorem BLDC, který není umístěn v teplotní komoře.
- Výstupní testování jednotky probíhá při nízké a vysoké teplotě. Z toho důvodu musí být součástí automatického testovacího pracoviště teplotní komora, která je schopná dosáhnout teploty -55 °C a 70 °C.
- Pro nasimulování hydraulické soustavy letounu je nedílnou součástí testovacího pracoviště přípravek, který je tvořen samotným motorem a dále olejovou nádrží, která je motorem tlakována. Součástí přípravku je měřič tlaku, aby obsluha mohla zkontrolovat, zdali bylo během testů se zátěží dosaženo žádaného tlaku.

Po dokončení všech zkoušek je jako výstup vygenerován tisknutelný soubor typu Microsoft Excel s výsledky všech provedených testů. Tento testovací protokol musí splňovat veškeré náležitosti dané interní směrnici pro tvorbu dokumentace ve společnosti UNIS, a.s.

Jednotlivé komponenty testovacího pracoviště a vazby mezi nimi znázorňuje Obr. 4.



Obr. 4 Schéma testovacího pracoviště.

6.1.1 Hardwarové prostředky

Automatické testovací pracoviště je založeno především na hardwaru od společnosti National Instruments a osobním počítači. Periferie automatického testovacího pracoviště pak tvoří dva zdroje napětí, teplotní a klimatická komora, osciloskop a dále speciální přípravky vyvinuté pro testování elektronické řídicí jednotky.

NI cRIO-9082

Zařízení cRIO-9082 je nejvýkonnější ze všech produktů řady CompactRIO společnosti National Instruments v současnosti dostupných. Mezi jeho parametry patří dvoujádrový procesor Intel Core i7, který se vyznačuje vysokým výkonem a zároveň nízkou spotřebou. Díky vícejádrové koncepci procesoru je možné paralelně zpracovávat samostatné úlohy nebo vlákna, a tím výrazně zrychlit vykonávání operací. V této verzi je k dispozici osm slotů pro integrované I/O moduly C Series.

Zařízení NI cRIO-9082 umožňuje zvolit operační systém, který na něm bude instalován. Je zde možnost vybrat si mezi determinismem a spolehlivostí systému LabVIEW Real-Time nebo možnost využívat rozsáhlou paletu softwaru a integrovaných funkcí uživatelského rozhraní systému WES7.

Operační systém LabVIEW Real-Time je ideálním prostředím pro aplikace náročné na rychlost, které vyžadují deterministické chování. Díky přesnému časování a možnosti přidělit úlohám priority lze v LabVIEW snadno vyvíjet deterministické úlohy, jako je řízení pohybu v uzavřené smyčce, a ty pak spustit na platformě CompactRIO. Operační systém LabVIEW Real-Time poskytuje také optimalizované prostředí navržené tak, aby byl zajištěn spolehlivý běh aplikačních programů 24 hodin denně, sedm dní v týdnu.

WES7 může obohatit aplikační program o další funkce tím, že dovoluje využívat spojení softwaru Windows s platformu LabVIEW pro Windows. Další výhody přináší použití funkcí v knihovnách .NET, prvků ActiveX a dalších knihoven DLL, dále je možné implementovat OPC server nebo se přímo připojit ke vzdálené databázi a snadno ukládat naměřené hodnoty. Dále je také možné použít vestavěný výstup VGA a implementovat vlastní uživatelské rozhraní, čímž se sníží náklady na systém a požadavky na údržbu, neboť pro uživatelské rozhraní není třeba mít zvláštní počítač.



Obr. 5 NI cRIO-9082 [22].

Zařízení cRIO-9082 disponuje programovatelnými hradlovými poli řady Xilinx Spartan-6. Softwarový modul LabVIEW FPGA umožňuje vytvářet vlastní měřicí a řídicí hardware pomocí grafického programování, aniž by bylo třeba umět nižší programovací jazyky běžně používané pro programování hradlových polí. Uvedený speciální hardware má unikátní schopnosti časování a spouštění, velmi rychlého řízení a přístupu k digitálním protokolům. Tyto a mnoho dalších operací vyžaduje velmi rychlý a spolehlivý hardware s dodržением deterministického chování. [21] Pro testovací pracoviště bylo použito zařízení cRIO-9082 s těmito přídatnými moduly:

- NI 9870 – karta obsahuje čtyři RS232 porty pro sériovou komunikaci. Tato karta slouží k zprostředkování komunikace s teplotní komorou a také s elektronickou řídicí jednotkou.
- NI 9264 – 16kanálový modul pro analogový výstup ± 10 V. Pomocí této karty jsou ovládány zdroje napětí pro napájení elektronické řídicí jednotky.
- NI 9403 – 32kanálový modul s nastavitelnými digitálními vstupy nebo výstupy. Touto kartou jsou ovládány jednotlivé relé v přizpůsobovacím adaptéru (spínání napájení jednotky a spínání obvodů pro měření proudu).
- NI 9229 – 4kanálová karta s analogovými vstupy ± 60 V. Tento modul je použit pro měření napětí v měřicím obvodu přizpůsobovacího adaptéru, které odpovídá odebíranému proudu elektronickou řídicí jednotkou.



Obr. 6 Přídatné moduly pro CompactRIO. Zleva NI 9870, NI 9264, NI 9403, NI 9229 [25][28][33][31].

Teplotní a klimatická komora Weiss WK3-180/70

Společnost Weiss Umwelttechnik je jedním z nejvýznamnějších výrobců standardních testovacích komor a systémů simulující různá prostředí. Sortiment zahrnuje teplotní a klimatické testovací systémy, systémy pro simulování různého počasí, teplotních šoků, působení koroze a dlouhodobé testování vlivu prostředí.

Tento typ komory patří ke kompaktnějším řešením, její objem je 190 litrů. Komora je schopna dosáhnout teploty od $-72\text{ }^{\circ}\text{C}$ až do $+180\text{ }^{\circ}\text{C}$. Pro rozsah teplot od $+10\text{ }^{\circ}\text{C}$ až do $+95\text{ }^{\circ}\text{C}$ je zde možnost regulovat relativní vlhkost a to v rozmezí od 10 % do 98 %. Dalším důležitým parametrem je rychlost teplotní změny, tato komora dosahuje rychlosti $4\text{ }^{\circ}\text{C}/\text{min}$ při vyhřívání a $3\text{ }^{\circ}\text{C}/\text{min}$ při ochlazování. Mezi další klíčovou vlastnost patří způsob ovládání komory. Komoru lze ovládat manuálně prostřednictvím 8" barevného dotykového displeje, nicméně pro aplikaci na automatickém testovacím pracovišti jsou důležitější možnosti externího ovládání. V základu je komora vybavena USB a Ethernet rozhraním, jako volitelnou výbavu je možné zvolit rozhraní RS232, RS422/485 nebo RS232/IEEE 488 a RS485/422. Pro zajištění komunikace přes NI cRIO bylo v tomto případě zvoleno sériové rozhraní RS232. Detailnějšímu popisu o možnostech komunikace a její implementace je věnována kapitola 7.1 [24].



Obr. 7 Klimatická komora Weiss WK3-180/70 [24].

Programovatelný zdroj napětí BK PRECISION XLN3640

Společnost B&K Precision má více než šedesáti letou tradici ve výrobě spolehlivých zařízení určených k měření a testování.

Vybraný zdroj napětí je kompaktní, nabízí jeden DC výstup a je určen především do laboratoře a do automatizovaných testovacích systémů. Zdroj poskytuje maximální výstupní napětí 36 V a výstupní proud 40 A. Pro vzdálené ovládání zde existuje hned několik možností. S PC může zdroj komunikovat přes USB rozhraní. Další možnost jak ovládat zdroj externě je analogovým signálem 0 – 10 V. Na ochranu testovaného zařízení má zdroj zabudovanou přepětovou, nadproudovou ochranu a ochranu před přetížením. Jelikož některé testy budou

vyžadovat větší výstupní napětí než 36 V, jsou pro testovací pracoviště použity dva tyto zdroje zapojené v sérii. [26]



Obr. 8 Programovatelný zdroj napětí BK PRECISION XLN3640 [26].

Digitální osciloskop Rohde & Schwarz RTE 1104

Rohde & Schwarz je nadnárodní společností, podnikající v oblastech elektronických testovacích a měřicích přístrojů, informačních technologií a radiokomunikací. Více než 75 let vyvíjí, vyrábí a prodává na světovém trhu elektronické produkty pro profesionální aplikace.

Digitální osciloskop R&S RTE 1104 nabízí 4 analogové a 16 digitálních kanálů, maximální šířka kmitočtového pásma je 1 GHz, rychlost vzorkování až 5 Gsample/s a v paměti je schopen uchovat až 10 Msample/kanál. Klíčovou vlastností osciloskopu je automatické měření signálových charakteristik. K dispozici je 77 předdefinovaných měřicích funkcí. Osciloskop lze ovládat velmi pohodlně přes 10.4" dotykový displej.

Důležitější je ovšem možnost vzdáleného ovládání osciloskopu z počítače. K osciloskopu se lze připojit jako k jinému počítači přes vzdálenou plochu, nicméně to není zrovna vhodný způsob komunikace pro automatické testovací pracoviště. K tomuto účelu výrobce poskytuje na svých stránkách ovladače pro různé platformy včetně LabVIEW. Tyto ovladače jsou volně ke stažení.

Osciloskop je na testovacím pracovišti použit pro snímání signálů generovaných z Hallových sond z motoru a měření jejich frekvence. Z frekvence jsou vypočítány reálné otáčky motoru a ty jsou následně porovnány s hodnotou otáček, kterou naměřila elektronická řídicí jednotka. [27]



Obr. 9 Digitální osciloskop Rohde & Schwarz RTE 1104 [27].

Příprubovací adaptér

Přípravek byl vytvořen společností UNIS, a.s. a to přímo za účelem testování elektronické řídicí jednotky. Hlavní funkcí tohoto přípravku je možnost softwarově (pomocí zařízení NI CompactRIO) ovládat napájení jednotky a zapínat režimy pro měření odebraného proudu jednotkou.

Napájení jednotky je zajištěno skrze adaptér, který obsahuje vstup pro napájení (připojení zdrojů BK PRECISION XLN3640) a výstup pro připojení jednotky. Napájecí napětí řídicí jednotky je vyvedeno na analogový výstup, aby ho bylo možné měřit pomocí analogových karet pro NI CompactRIO. Samotné obvody adaptéru jsou napájeny zvlášť externím zdrojem s konstantním napětím 24 V a proudem 0,5 A. Napájení jednotky je možné odpojit pomocí relé, které je řízeno digitálním vstupem.

Další funkcí adaptéru je měření odebíraného proudu jednotkou a jeho lineární převod na napětí. Proud je měřen ve dvou rozsazích:

- 0 – 200 mA, kde 200 mA odpovídá 10 V na výstupu.
- 0 – 10 A, kde 10 A odpovídá 10 V na výstupu.

Mezi rozsahy se volí přepínáním relé pomocí digitálních vstupů, které budou ovládány přes PC resp. NI CompactRIO, stejně jako zapínání a vypínání napájení.



Obr. 10 Přizpůsobovací adaptér.

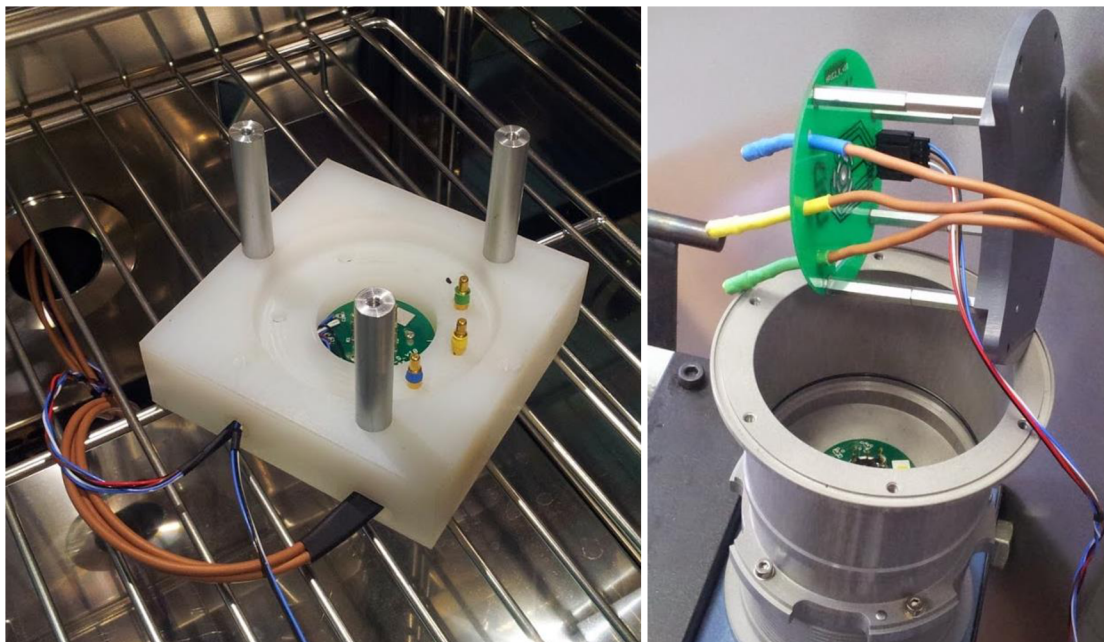
Mechanické přípravy pro propojení jednotky s BLDC motorem

Tyto mechanické přípravy byly taktéž vytvořeny společností UNIS, a.s. pro potřeby testovacího pracoviště elektronické řídicí jednotky. Jelikož je jednotka během testování umístěna v uzavřené klimatické komoře, není možné, aby byla při testech s připojeným motorem připevněna přímo k šasi s motorem tak, jak to má za běžného provozu být.

První mechanický přípravek téměř kopíruje horní část pouzdra od motoru, takže jednotka je k němu připevněna stejně, jako by byla připevněna k motoru ovšem s tím rozdílem, že místo motoru je z přípravku vyvedena kabeláž a to do druhého přípravku, který naopak nahrazuje spodní část elektronické řídicí jednotky. Díky těmto dvěma přípravkům je tedy umožněno napájení motoru z jednotky, i když jsou jednotka a motor od sebe fyzicky odděleny. Dále jsou pomocí těchto přípravků přenášeny signály z třech Hallových sond namontovaných na motoru do elektronické řídicí jednotky a dále na sondy osciloskopu.

Všechny přípravy a adaptéry vyrobené společností UNIS, a.s. mají své označení (produktové a sériové číslo) a také by měly být uvedeny jejich konfigurace hardwaru a softwaru (pokud je

obsažen). Toto označení slouží k tomu, aby bylo možné vysledovat, který hardware a software byl při testech použit.



Obr. 7 Mechanické přípravy pro propojení elektronické řídicí jednotky s BLDC motorem.

6.1.2 Softwarové prostředky

Pro vývoj softwaru automatického testovacího pracoviště byly použity softwarové prostředky od společnosti National Instruments. Konkrétně vývojové prostředí LabVIEW a TestStand pro tvorbu testovacích sekvencí.

NI LabVIEW

Vývojové prostředí LabVIEW (z angl. Laboratory Virtual Instruments Engineering Workbench) je produktem americké společnosti National Instruments a je zaměřeno na vývoj testovacích, měřicích a řídicích aplikací. LabVIEW se řadí do kategorie grafických programovacích jazyků a k vývoji aplikací slouží tzv. G-language (tedy „grafický“ jazyk). Programové bloky jsou vytvářeny v podobě blokových diagramů, ty se skládají z funkčních bloků, které jsou navzájem propojeny dráty. Jednotlivé programy a podprogramy v LabVIEW se nazývají angl. Virtual Instruments (dále VI). Ty jsou tedy tvořeny blokovým diagramem, předním panelem (angl. front panel), který zajišťuje základní grafický interface a konektorovým panelem (angl. connector panel), ten slouží pro nastavení vstupních a výstupních proměnných do VI. Tento způsob reprezentace zdrojového kódu umožňuje intuitivně využívat systém i lidem s menšími zkušenostmi v programování. Velkou výhodou LabVIEW je, že zahrnuje velké množství knihoven obsahující spoustu funkcí, které značně usnadňují vývoj aplikací, konkrétně např. funkce pro sběr dat, generování signálu, matematické, statistické, analytické funkce a mnoho dalších. Další z klíčových vlastností LabVIEW je široká podpora zařízení jako jsou měřicí přístroje, kamery apod. Uživatel přistupuje k tomuto hardwaru buď odesláním příkazů po sběrnici (USB, GPIB, sériové rozhraní, ...), anebo pomocí ovladačů, které výrobci často ke svým zařízením poskytují. Programování v grafickém prostředí nabízí i poměrně jednoduchý způsob, jak implementovat aplikace využívající paralelismus, ve smyslu vícevláknového programování. [29][30]

LabVIEW je tak vhodným nástrojem pro implementaci automatických testovacích systémů, a proto je výhradně využit jako nástroj pro vývoj softwaru k testování elektronické řídicí jednotky.

NI TestStand

TestStand je dalším produktem společnosti National Instruments a byl vytvořen, aby zákazníkům efektivně pomáhal vyvíjet automatizované testovací a validační systémy. TestStand může být použit k tvorbě, vykonání a zavádění softwaru testovacího systému. V prostředí TestStandu lze vytvářet testovací sekvence, jež se skládají z kroků, které mají charakter testů, různých akcí, případně kroků ovlivňující průběh vykonávání sekvence. TestStand je také schopen automatického generování zprávy o výsledcích testů ve standardizovaných formátech, jako jsou ATML, XML, HTML, ale i ve formě prostého textu nebo dokáže pro následnou analýzu ukládat výsledky do podnikových databází. Pro testovací pracoviště je tento software použit právě pro tvorbu testovacích sekvencí, přičemž jednotlivé testy a akce jsou implementovány v LabVIEW. Díky modifikovatelnému uživatelskému rozhraní pro obsluhu testovacího pracoviště, které TestStand v základu nabízí, je využit i pro vykonávání testů a následné generování testovacího protokolu s výsledky. [32]

7 IMPLEMENTACE

Implementace testovacího softwaru zahrnovala prostudování možností komunikace s použitými hardwarovými prostředky. Pro případy, kdy výrobce ke svému zařízení neposkytoval ovladače pro LabVIEW, bylo nutné tyto ovladače vytvořit. Vývoj testovacího softwaru zahrnuje programování v LabVIEW v kombinaci s platformou TestStand, ve které jsou vytvořeny a spouštěny testovací sekvence.

7.1 Komunikace s teplotní a klimatickou komorou

Teplotní a klimatická komora Weiss WK3-180/70 je připojena k počítači, resp. k zařízení cRIO prostřednictvím rozhraní RS232. Pro komunikaci přes sériové rozhraní platí následující parametry:

- Rychlost 9600/115200 baudů
- 1 start bit
- 8 datových bitů
- 1 stop bit
- Bez parity
- Bez identifikace („Handshake“)

Výrobce umožňuje pro komunikaci s teplotní komorou výběr ze dvou protokolů a to protokoly ASCII-1 a ASCII-2. Komoru lze ovládat i prostřednictvím rozhraní Ethernet, ovšem to by zahrnovalo implementaci vlastního TCP/IP klienta, protože ovládací software SIMPATI*, který výrobce nabízí, není vhodný pro vyvíjené automatické testovací pracoviště. Z toho důvodu bylo vybráno právě rozhraní RS232.

7.1.1 Popis komunikačních protokolů

Pomocí protokolů ASCII-1 a ASCII-2 je možné zadávat požadované hodnoty regulovaných veličin, dotazovat se na skutečné hodnoty regulovaných veličin, dotazovat se na stav digitálních kanálů a měnit jej, dotazovat se na teplotu více volných měřicích čidel, spouštět a zastavovat programy, načítat chybové stavy, číst chybová hlášení v textové podobě a potvrzovat tyto chybová hlášení. Protokol ASCII-2 umí navíc zadávat rychlosti změn při skokových změnách požadovaných hodnot.

Protokol ASCII-1

Komunikace mezi počítačem a komorou funguje na principu otázky a odpovědi. Z počítače je odeslán řetězec, kterým se počítač táže na data. Komora odpoví na tento řetězec tím, že odešle odpovědný řetězec zpět počítači.

Odesílané a přijímané řetězce obsahují posloupnost ASCII znaků. Každý řetězec musí mít na začátku ASCII znak {STX} („Start of Text“, ASCII kód 02) a na konci {ETX} („End of Text“, ASCII kód 03). Pro následující ukázky bude pro zjednodušení nahrazena adresa zařízení na sběrnici zástupným symbolem z a kontrolní součet symbolem CC. Pro přijímané řetězce jsou typické další dva ASCII znaky a to {ACK} pro potvrzení, že komora akceptovala řetězec a {NAK} pro případ, kdy řetězec rozpoznán nebyl.

Způsob použití protokolu nejlépe znázorní následující příklady, které zahrnují konkrétní podobu vybraných příkazů a jejich popis.

Načtení skutečných hodnot

Požadavek: {STX}z?CC{ETX}

- ? – znak symbolizuje požadavek pro získání hodnot z komory

Odpověď: {STX}zT018.5F066.0P0T000.0#--T010F090.0R1000000000000000CC{ETX}

- T018.5 – skutečná hodnota regulované veličiny 1 je 18,5
- F066.0 – skutečná hodnota regulované veličiny 2 je 66

- P0 – tiskárna je vypnutá (0), zapnutá (1)
- T000.0 – teplota na volném teplotním čidle
- # – zařízení je zapnuté (#), vypnuté (\$)
- -- – počet ohlášených chyb (-- znamená bez chyb)
- T010.0 – požadovaná hodnota regulované veličiny 1 je 10
- F090.0 – požadovaná hodnota regulované veličiny 2 je 90
- R1000000000000000 – stav digitálních kanálů (vypnuté = 0, zapnuté = 1)

Nastavení požadovaných hodnot regulovaných veličin

Požadavek: {STX}zT025.0F35R110000000000000CC{ETX}

- T025.0 – požadovaná hodnota regulované veličiny 1 je nastavena na 25,0
- F35 – požadovaná hodnota regulované veličiny 2 je nastavena na 66
- R1100000000000000 – tato kombinace digitálních kanálů zajistí spuštění komory a spuštění regulace obou regulovaných veličin

Odpověď: {STX}z{ACK}CC{ETX}

- {ACK} – příkaz byl přijat

Spouštění programu

Požadavek: {STX}z:AutoStart:xxx:CC{ETX}

- xxx – 1 až 120 (číslo programu)

Odpověď: {STX}z{NAK}CC{ETX}

- {NAK} – příkaz nebyl přijat (např. pokud bylo zadáno číslo mimo rozsah 1 až 120)

Čtení chybových textů

Požadavek: {STX}z:Get:ErrorText:xx:CC{ETX}

- xx – číslo chybového hlášení, jehož text je třeba vyčíst

Odpověď: {STX}z:Get:ErrorText:xx:Text chyby:CC{ETX}

- xx – číslo vyčítaného chybového hlášení
- Text chyby – zde je vypsán text chybového hlášení

Protokol ASCII-2

Stejně jako protokol ASCII-1 funguje tento protokol na stejném principu, tzn. z počítače je odeslán požadavek, na který komora odešle odpověď, ovšem pokud je příkaz takového typu, že po jeho odeslání nejsou požadována žádná data, zůstane požadavek bez odpovědi. Odesílaný řetězec opět obsahuje posloupnost ASCII znaků. Každý řetězec má na začátku znak \$ (ASCII kód 36) a je ukončen znakem Carriage Return {CR} (ASCII kód 13).

Pro názornost je tento protokol opět popsán několika příklady konkrétních příkazů, které mají stejnou funkci, jako příkazy popsané výše u protokolu ASCII-1. Pro zjednodušení je v příkladech část řetězce, který udává adresu zařízení na sběrnici, nahrazen symboly zz.

Načtení skutečných hodnot

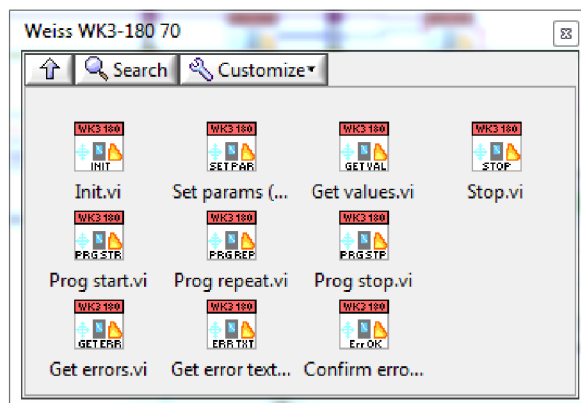
Požadavek: \$zzl{CR}

- l – znak symbolizuje požadavek pro získání hodnot z komory

Odpověď: \$0023.0 0020.5 0050.0 0041.0 0080.0 0000.0 0000.0 0020.0 0000.0 0020.2 0000.0 0020.3 0000.0 0020.4 01101010101010101010101010101010{CR}

- 0023.0 – požadovaná hodnota regulované veličiny 1
- 0020.5 – skutečná hodnota regulované veličiny 1
- 0050.0 – požadovaná hodnota regulované veličiny 2
- 0041.0 – skutečná hodnota regulované veličiny 2
- 0080.0 – hodnota nastavení 1
- 0000.0 – nepoužito
- 0000.0 – nepoužito
- 0020.0 – skutečná hodnota Pt100-1 (volitelné příslušenství)
- 0000.0 – nepoužito

standardu VISA od společnosti National Instruments a jeho knihovna poskytuje efektivní používání sady příkazů ke komunikaci s řadou přístrojů. Jednou z výhod VISA je možnost používat stejné operace pro komunikaci s přístroji přičemž vůbec nezáleží na typu rozhraní. Například VISA příkaz pro zapsání ASCII řetězce do přístroje je stejný ať přístroj využívá, sériové, GPIB nebo USB.

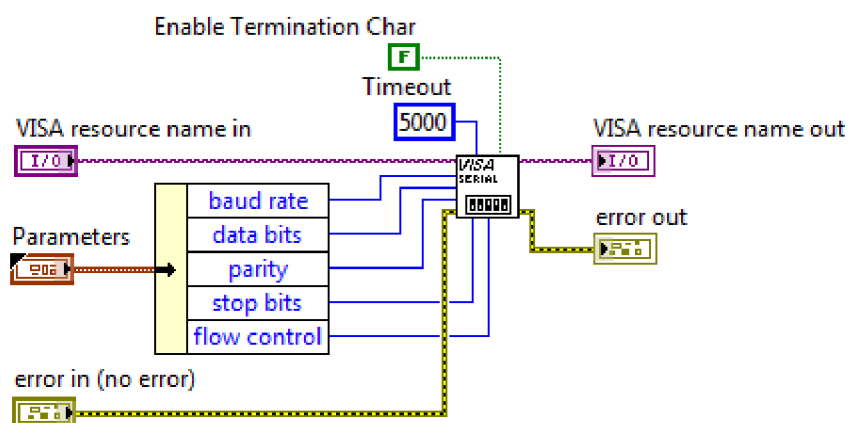


Obr. 11 Paleta nástrojů z knihovny pro komunikaci s teplotní komorou Weiss WK3-180/70.

Implementace jednotlivých bloků je pro všechny velmi podobná a lze ji rozdělit do několika kroků:

1. Inicializace komunikace, nastavení její parametrů.
2. Sestavení řetězce znaků, které tvoří daný příkaz.
3. Výpočet kontrolního součtu a jeho přidání na konec řetězce.
4. Zápis na port pomocí VISA knihovny (odeslání požadavku).
5. Čtení z portu pomocí VISA knihovny (přijetí odpovědi).
6. Získání informace z přijatého řetězce.

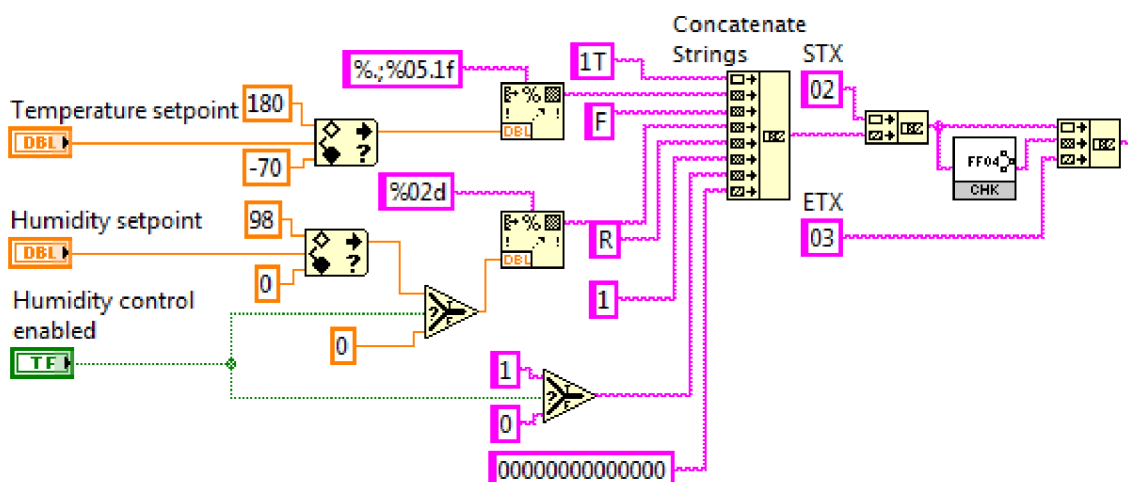
Prvním blokem, který musí být použit na začátku každé komunikace je *Init.vi*. Tento blok inicializuje sériový port s danými parametry a to pomocí funkce VISA Configure Serial Port. Implementace je zobrazena pomocí blokového diagramu na *Obr. 12*.



Obr. 12 Inicializace komunikace pro klimatickou komoru.

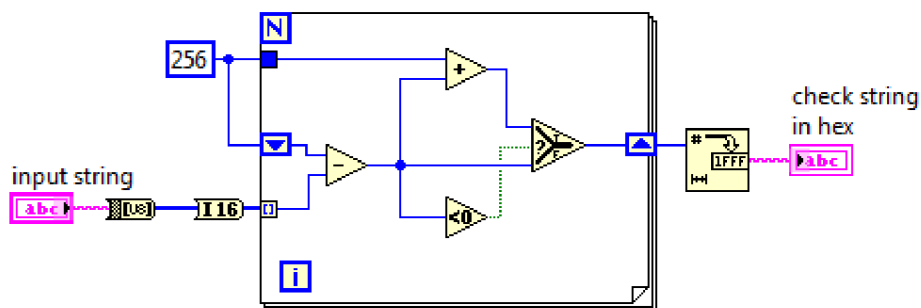
Dalším krokem je sestavení řetězce, který bude zapsán na sériový port. Blokový diagram na *Obr. 13* je vyjmut z VI *Set params (start).vi* a slouží k nastavení hodnot regulovaných veličin a startu klimatické komory. Vstupními proměnnými jsou hodnoty požadovaných regulovaných veličin a dvouhodnotová proměnná, která určuje, zdali má být zapnuta i regulace vlhkosti. Vstupní hodnoty musí být ošetřeny horním a spodním limitem, který vyplývá z parametrů klimatické

komory. Řetězce jsou sestaveny pomocí bloku *Concatenate Strings* do tvaru, který je popsán v kapitole 7.2.1.



Obr. 13 Sestavení řetězce pro start klimatické komory a nastavení hodnot žádaných veličin.

Součástí každého sestaveného řetězce je kontrolní součet. Kontrolní součet je určitým druhem křížového součtu ASCII hodnot řetězce včetně ASCII hodnoty znaku {STX}, znak {ETX} se již do tohoto řetězce nepočítá, kontrolní součet je udáván v hexadecimálním tvaru, např. 8E. Teplotní komora kontroluje kontrolní součet přijímaného řetězce, aby identifikovala případnou chybu přenosu. Výpočet kontrolního součtu je znázorněn na Obr. 14.

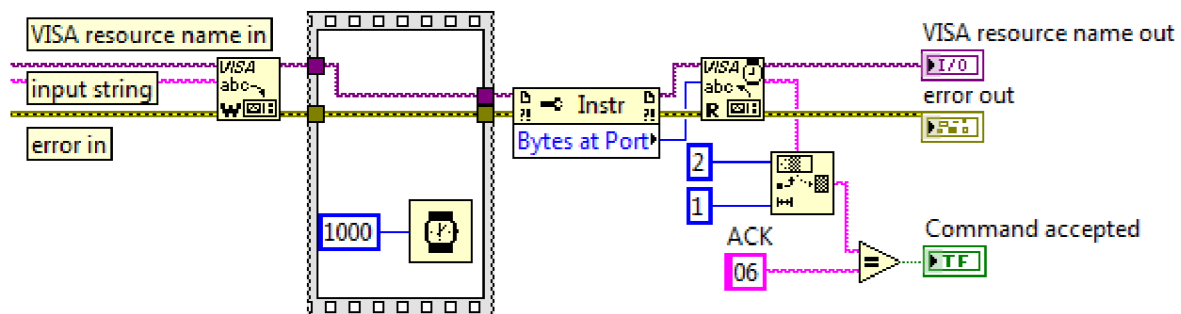


Obr. 14 Výpočet kontrolního součtu při komunikaci s klimatickou komorou.

Po sestavení řetězce následuje jeho odeslání přes sériové rozhraní. K tomu slouží blok z knihovny NI-VISA a to *VISA Write*. Jak je dáno v manuálu [23], po zapsání je nutné vyčkat alespoň jednu vteřinu. Pokud by tomu tak nebylo, dojde k zahlcení komunikace, poté přestane komora reagovat na veškerou další komunikaci a je nutné ji restartovat.

Dalším krokem po odeslání požadavku je vyčtení odpovědi z komory. Čtení ze sériového portu je opět zajištěno blokem z VISA knihovny, tentokrát *VISA Read*. Po přijetí řetězce s odpovědí je z něj vyjmuta požadovaná část pomocí funkce *String Subset*, následně je tato hodnota porovnána s konstantou, která má hodnotu 06, což je ASCII hodnota pro znak {ACK} (acknowledge). Pokud

jsou si tyto dvě hodnoty rovny, znamená to, že odeslaný požadavek byl komorou kladně vyhodnocen. Zápis, čtení a zpracování přijatého řetězce znázorňuje blokový diagram na Obr. 15.

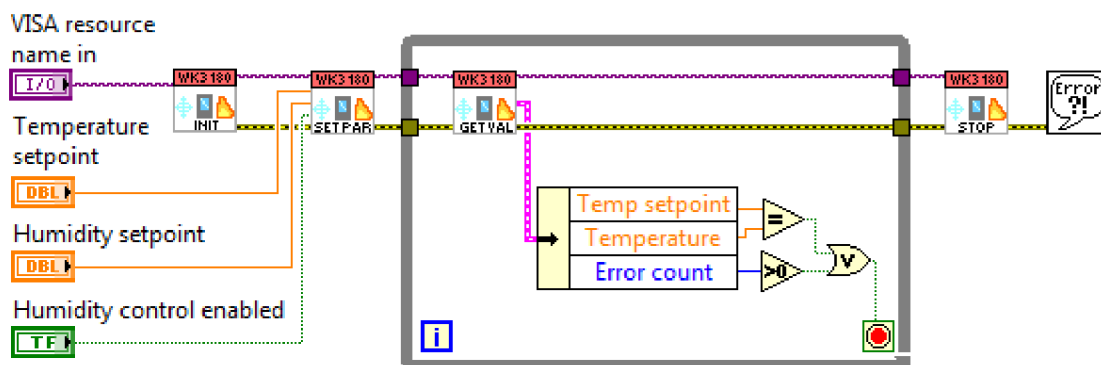


Obr. 15 Odeslání požadavku, přijetí a zpracování odpovědi při komunikaci s klimatickou komorou.

Další funkce jsou implementovány obdobným způsobem, liší se pouze v částech, kde jsou vytvářeny řetězce a poté jejich zpracování. Například při vyčítání skutečných hodnot regulovaných veličin je odpovědní řetězec z komory značně delší, protože obsahuje o mnoho více informací, než jen potvrzení o přijetí příkazu, jako je tomu na Obr. 15.

Následující blokový diagram (Obr. 16) demonstruje použití knihovny. Příklad slouží pouze pro ukázkou způsobu použití některých bloků z vytvořené knihovny a není přímo součástí testů elektronické řídicí jednotky. Tento blokový diagram vykonává následující činnost:

1. Inicializace komunikace (*Init.vi*).
2. Nastavení parametrů a start klimatické komory (*Set params (start).vi*).
3. Ve smyčce vyčítá z komory aktuální teplotu a počet výskytů chyb (*Get values.vi*). Pokud se objeví chyba, nebo je dosaženo žádané teploty, je smyčka ukončena.
4. Zastavení klimatické komory (*Stop.vi*).
5. Zpracování případných chybových hlášení.



Obr. 16 Příklad použití knihovny pro ovládání klimatické komory.

7.2 Komunikace s elektronickou řídicí jednotkou

Pro komunikaci mezi řídicí jednotkou a nadřazeným systémem je použita sériová sběrnice SCI. Řídicí jednotka nevyžaduje ke své činnosti komunikaci, ale pro servisní účely je SCI sběrnice vyvedená z jednotky jako RS232. Komunikace po sběrnici je realizována prostřednictvím komunikačního protokolu CANaerospace verze 1.7 [34]. Parametry komunikace jsou následující:

- Rychlost 115 200 baudů
- 8 datových bitů
- 1 stop bit
- Bez parity

7.2.1 Popis protokolu CANaerospace

CANaerospace [34] protokol vyšší vrstvy, který byl navržen pro vysoce spolehlivou komunikaci mezi systémy v leteckých aplikacích založených především na mikroprocesorové technologii. Účelem CANaerospace je vytvořit standard pro aplikace vyžadující efektivní sledování toku dat a jednoduchou rámcovou synchronizaci v rámci redundantních systémů. CANaerospace je udržován jako otevřený, takže je do něj možné implementovat uživatelské zprávy a protokoly.

Protokol CANaerospace specifikuje 6 základních typů zpráv, které jsou využívány pro různé služby. Každá zpráva je asociována s vlastním identifikátorem CAN-ID, přičemž rozsahy hodnot tohoto identifikátoru jsou definovány podle priority zpráv. V protokolu jsou definovány následující typy zpráv:

- Emergency Event Data (EED) – tyto zprávy jsou vysílány asynchronně a to ihned po výskytu události. Tyto zprávy se používají pro hlášení výjimečných situací nebo událostí, které vyžadují okamžitou reakci, jako jsou například chyby. Tyto zprávy mají CAN-ID v rozsahu od 0 do 127.
- High Priority Node Service (NSH) – komunikační kanály pro služby s vysokou prioritou. Komunikační kanál je vytvořen dvojicí CAN-ID, prostřednictvím kterých je provedeno komunikační propojení typu dotaz a odpověď. Typicky odpověď přichází na CAN-ID o jednotku větší než CAN-ID s dotazem. Mohou být vysílány asynchronně, nebo cyklicky s nastaveným intervalem opakování. Rozsah CAN-ID je od 128 do 199.
- High Priority User-defined Data (UDH) – zprávy vysílané s uživatelsky definovaným datovým typem i periodou zaslání zpráv. Tyto zprávy se používají pro přenos obecně použitelných veličin. CAN-ID je v rozsahu od 200 až do 299.
- Normal Operation Data (NOD) – tyto zprávy jsou vysílané buď cyklicky v pravidelných časových intervalech, anebo asynchronně. Zprávy obsahují předdefinované typy veličin, které jsou často používány v letectví, jako např. letová výška, statický tlak nebo jiné veličiny týkající se stavu motoru, data navigačního systému apod. Těmto zprávám je přidělen rozsah CAN-ID od 300 do 1899.
- Low Priority User-defined Data (UDL) – zprávy typu UDL mají stejné vlastnosti jako zprávy UDH, jediným rozdílem je, že tyto zprávy mají nižší prioritu. Pro tyto zprávy je vyhrazeno CAN-ID od 1800 do 1899
- Debug Service Data (DSD) – rámce tohoto typu se využívají pro ladění aplikace (debugging) a pro činnosti spojené s nahráváním softwaru do zařízení. Zprávy jsou odesílány asynchronně nebo cyklicky a je pro ně vyhrazeno CAN-ID od 1900 do 1999
- Low Priority Node Service Data (NSL) – poslední typ zpráv, který slouží pro testování či provádění údržby zařízení. Jsou vysílány asynchronně nebo cyklicky s CAN-ID od 2000 do 2031.

Formát datového rámce CANaerospace

Obecný formát zprávy využívá 4 bajty jako hlavičku zprávy, kde je definován identifikátor zařízení (node-ID), datový typ, kód zprávy (Message code) a kód služby (Service code - pro zprávy typu NOD, jinak je toto pole definováno uživatelem). Díky této hlavičce je umožněno identifikovat každou zprávu od jakéhokoliv zařízení bez nutnosti znát jiné dodatečné informace. Formát datového rámce protokolu CANaerospace je znázorněn v *Tab. 1*.

Pro reprezentaci dat jsou nejčastěji používány základní datové typy předdefinovány. Kromě toho jsou definovány i kombinované datové typy, tj. dva, tři nebo čtyři 8bitové, popř. 16bitové datové typy v jedné zprávě. Dále jsou podporovány i datové typy, jako je 64bitový double float. Každý datový typ je v hlavičce zprávy označen příslušným identifikátorem (rozsah od 0 do 255). Pokud uživatel vyžaduje nějaký speciální datový typ, v rámci protokolu CANaerospace je možné

definovat si i vlastní datové typy, k těmto účelům byl přiřazen rozsah od identifikátorů od 100 do 255.

Header		Data					EOF
		CANaerospace header			CANaerospace data		
Length	CAN-ID	Node-ID	Data type	Service code	Message code	Message data	ASCII 13
1 byte	2 byty	1 byte	1 byte	1 byte	1 byte	0 až 8 bytů – záleží na typu.	1 byte

Tab. 1 Formát rámce protokolu CANaerospace.

Popis použitých zpráv a služeb

Komunikace s řídicí jednotkou probíhá prostřednictvím datových zpráv EED, NOD a služeb DUS, RCRS a SCRS, přičemž služby RCRS a SCRS jsou uživatelsky nadefinované služby a slouží pro čtení a zápis do konfiguračních registrů jednotky. V rámci vývoje automatického testovacího pracoviště postačily k tvorbě jednotlivých testů zprávy typu EED, NOD a dále služba pro vyčítání registrů z jednotky RCRS, ostatní služby tedy zůstanou pouze ve výčtu a nebudou podrobně specifikovány.

Zprávy typu EED

EED zprávy jsou vysílány asynchronně hned po výskytu události (zpravidla nějaká porucha). Zprávy používají pro reprezentaci dat speciální předdefinovaný datový formát. Podobu zprávy typu EED znázorňuje Tab. 2.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
CANaerospace header				Error code		Error level	255

Tab. 2 Formát zprávy typu EED.

Error code slouží k identifikaci vzniklé události a error level popisuje závažnost chyby. Závažnost chyby je rozdělena na čtyři úrovně. Úroveň 1 je vyhrazena pro informativní účely, úroveň 2 již popisuje událost, která vedla ke změně funkčnosti jednotky. V tomto případě došlo k zotavení po události s úrovní 3. Chyba úrovně 3 signalizuje, že funkční stav jednotky se změnil k horšímu. Poslední úroveň 4 je vyhrazena pro závažnou chybu, která vede k ukončení hlavní činnosti jednotky. Při testech bylo zapotřebí ověřit funkčnost přepětové ochrany jednotky, když ochrana zareaguje, vyše jednotka po komunikaci EED zprávu s příslušným identifikátorem chyby. Úspěšným vyčtením této zprávy (po prudkém zvýšení napětí na definovanou úroveň) pak bylo možné potvrdit funkčnost přepětové ochrany.

Zprávy typu NOD

V řídicí jednotce jsou NOD zprávy vysílány v pravidelných časových intervalech po 100 ms. Pomocí těchto zpráv jsou odesílány obecně použitelné veličiny, konkrétně to je např. příkon, otáčky motoru, směr rotace, teplota aj. Formát zprávy NOD je znázorněn v Tab. 3. Délka tohoto rámce je proměnná v závislosti na použitém datovém typu pro reprezentaci daných veličin.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
CANaerospace header				Message data (délka dle datového typu)			

Tab. 3 Formát zprávy typu NOD.

Služba RCRS

Služba RCRS (Read Configuration Register Service) slouží pro čtení konfiguračních registrů řídicí jednotky. Jedná se o uživatelsky definovanou službu, tato služba není běžnou součástí protokolu CANaerospace. Služba funguje tak, že pro získání obsahu vyčítaného registru je zaslán požadavek s příslušným CAN-ID a jednotka po jeho zpracování odešle odpověď, jež obsahuje hodnotu požadovaného registru. Zpráva s odpovědí má hodnotu CAN-ID o jednotku větší než byl požadavek.

Byte č.	Položka	Požadavek	Odpověď
0	Node-ID	Podle adresy zařízení	Podle adresy zařízení
1	Data type	0	Podle obsahu registru
2	Service code	102	102
3	Message code	Číslo registru	Číslo registru
4÷7	Message data	Nepoužito	Obsah registru

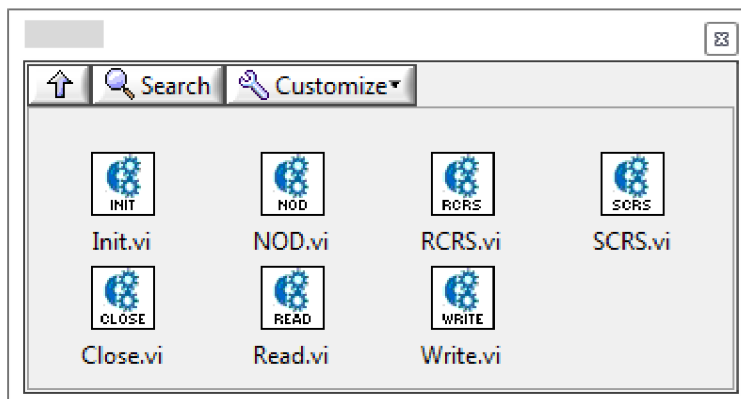
Tab. 4 Formát zpráv služby RCRS.

Služba RCRS začíná vysláním požadavku na čtení registru, formát odeslané zprávy je znázorněn ve sloupci požadavek v Tab. 4. Položka Node-ID obsahuje adresu zařízení na sběrnici, ze kterého se má registr vyčítat. Položka data type obsahuje hodnotu 0, čímž je označen datový typ protokolu CANaerospace pro zprávy, které neobsahují žádná data. Service code obsahuje hodnotu 102, touto hodnotou je identifikována právě služba RCRS. Číslo registru, který má být vyčten, je zapsáno v poli message code. Poslední čtyři byty nejsou použity. Po odeslání požadavku následuje příjem odpovědi ze zařízení s obsahem požadovaného registru. Jednotlivá pole v rámci jsou totožná s požadavkem až na dvě výjimky a to pole data type, které obsahuje datový typ hodnoty přečtené z registru a pole message data, to už obsahuje samotný údaj vyčtený z registru. Délka tohoto pole je závislá na použitém datovém typu.

7.2.2 Knihovna pro komunikaci s řídicí jednotkou

Nástroje pro komunikaci s řídicí jednotkou jsou také implementovány v prostředí LabVIEW a to pomocí knihovny NI-VISA. Jelikož knihovna pro ovládání řídicí jednotky vznikla v rámci týmové spolupráce na projektu a její implementace tak nemůže být brána jako výsledek této diplomové práce, jako je tomu u knihovny ke klimatické komoře. Díky knihovně NI-VISA je implementace jednotlivých nástrojů z palety velice podobná, jako pro jakékoliv jiné zařízení se sériovým rozhraním. Princip je stále stejný. Nastavení parametrů pro komunikaci se zařízením, viz úvod této kapitoly. Dále následuje sestavení rámce, pokud je odeslán nějaký požadavek na zařízení (např.

čtení/zápis obsahu registru z jednotky pomocí služby RCRS/SCRS). Následuje čtení rámců a jejich zpracování podle toho, jaký typ zprávy je očekáván (EED, NOD aj.).



Obr. 17 Paleta nástrojů z knihovny pro komunikaci s elektronickou řídicí jednotkou.

Knihovna pro komunikaci s řídicí jednotkou obsahuje následující bloky (Obr. 17):

- *Inicializace komunikace (Init.vi)* – v tomto bloku jsou nastaveny příslušné parametry pro komunikaci a zároveň je odeslán požadavek na identifikaci jednotky. Pokud nedojde k úspěšnému přijetí rámce s identifikací jednotky, nedošlo k navázání komunikace a je ohlášena chyba. Tento blok musí být zavolán jako první, předtím než se začnou provádět další operace.
- *Čtení (Read.vi)* – tento blok slouží k vyčítání všech rámců, které chodí po komunikaci. Výstupem z tohoto bloku je pak cluster s jednotlivými položkami, které tvoří rámec protokolu CANaerospace (Tab. 1). Clusterem je v prostředí LabVIEW označována struktura obsahující více prvků s různými datovými typy.
- *Zápis (Write.vi)* – vstupem pro tento blok je cluster odpovídající svými prvky rámci protokolu CANaerospace, jako je tomu u bloku pro čtení.
- *Čtení zpráv NOD (NOD.vi)* – během testování se často používá vyčítání cyklicky zasílaných zpráv z jednotky. Tyto zprávy obsahují veličiny popisující její stav nebo veličiny naměřené různými senzory. Blok *NOD.vi* vyžaduje jako vstupní parametr hodnotu CAN-ID pro žádanou veličinu. Jelikož jsou tyto zprávy zasílány s určitou periodou, blok automaticky počká na požadovanou zprávu a až ji dostane, na jeho výstupu se objeví přijatý rámec (cluster s jednotlivými položkami rámce). Čtení jednotlivých rámců zajišťuje blok *Read.vi*.
- *Služba RCRS (RCRS.vi)* – další často opakovanou činností při testování je vyčítání hodnot registrů z řídicí jednotky. Jak bylo zmíněno výše, nejdříve je odeslán požadavek, to je zajištěno blokem *Write.vi*. Ihned po odeslání požadavku následuje vyčtení odpovědi pomocí bloku *Read.vi*. Vstupem do tohoto bloku je tedy číslo požadovaného registru, Node-ID zařízení, ze kterého je čteno a CAN-ID. Výstupem je opět cluster s jednotlivými položkami rámce, který obsahuje v datové části hodnotu vyčítaného registru.
- *Služba SCRS (SCRS.vi)* – tento blok není při výstupních testech využit, ale v rámci projektu byl používán k nastavení konfiguračních registrů na výchozí hodnoty. Jeho implementace je velice podobná službě RCRS.
- *Ukončení komunikace (Close.vi)* – pro ukončení komunikace musí být zavolán tento blok, dojde k uzavření sériového portu.

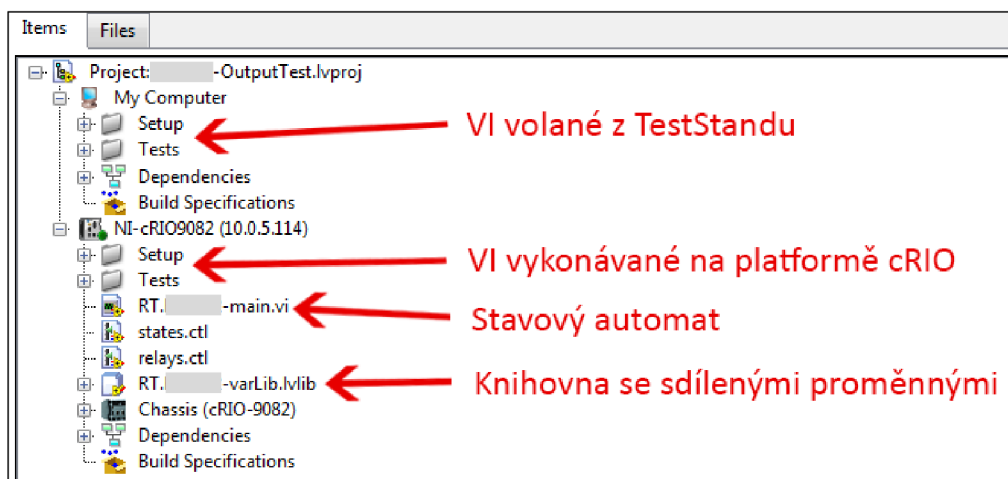
7.3 Komunikace s osciloskopem

S osciloskopem Rohde & Schwarz RTE 1104 komunikuje cRIO přes ethernetové rozhraní. Osciloskop je připojen do firemní sítě UNIS, a.s. stejně jako cRIO. Výrobce na svých webových stránkách nabízí ovladače pro LabVIEW volně ke stažení. Pomocí těchto ovladačů lze efektivně ovládat kompletně celý osciloskop. Použití těchto ovladačů lze vidět na Obr. 25.

7.4 Koncepce a realizace testovacího softwaru

Software pro automatické testovací pracoviště je realizován pomocí dvou platform od společnosti National Instruments. Jednotlivé testovací sekvence, které zahrnují nastavování počátečních parametrů pro test a samotné testy, jsou vytvořeny v TestStandu. Testovací sekvence se skládá z kroků, tyto kroky pak volají podprogramy (VI), které jsou implementovány v LabVIEW.

Software se skládá ze dvou hlavních částí, první část je spouštěna z PC, zatímco druhá část běží v realtime režimu (dále RT) na zařízení NI cRIO-9082. Hlavní myšlenka je taková, že na RT platformě je implementován stavový automat, jehož stavy představují jednotlivé testy nebo podprogramy pro nastavení hardwaru. Všechny testy jsou tedy prováděny na RT platformě, čímž je umožněno měření a provádění testů, které vyžadují kritické časování. Z PC resp. z TestStandu jsou pak volány jednotlivé podprogramy, které přepínají mezi stavy ve stavovém automatu. Předávání vstupních parametrů pro testy a výstupních (naměřených) hodnot z testu mezi PC a zařízením cRIO-9082 je zajištěno pomocí tzv. sdílených proměnných, které umožňují předávat si proměnné mezi jednotlivými zařízeními po síti. Na Obr. 18 je, pro lepší představu o rozdělení softwaru, zobrazena struktura projektu v LabVIEW.



Obr. 18 Struktura projektu s testovacím softwarem v LabVIEW.

Sdílené proměnné v LabVIEW

Pomocí sdílených proměnných (shared variables) lze sdílet data mezi dvěma smyčkami v jednom blokovém diagramu nebo mezi různými VI v rámci celé sítě. Sdílené proměnné se rozdělují na tři typy, tzv. Single-Process, Network-Published a Time-Triggered Shared Variables.

Single-Process Shared Variables se používají pro přenášení dat mezi dvěma různými místy v rámci stejného VI, které nemohou být propojeny (např. paralelní smyčky), nebo dvěma různými VI v rámci jedné instance aplikace. Tyto proměnné jsou podobné globálním proměnným v LabVIEW, ovšem s tou výhodou, že je lze snadno přepnout na Network-Published proměnné a umožnit k nim přístup z celé sítě.

Time-Triggered Shared Variables se používají k přenosu dat v rámci jednotlivých VI v projektu přes síť. Tento typ proměnných umožňuje přenášet data mezi RT zařízeními deterministicky, na rozdíl od Network-Published Variables, kde deterministický je pouze přenos do tzv. Shared Variable Engine (SVE), ale samotný přenos přes síť již deterministický není. Nevýhodou ovšem je, že aby bylo možné využít tento typ proměnné, musí navíc mezi zařízeními existovat uzavřená soukromá síť.

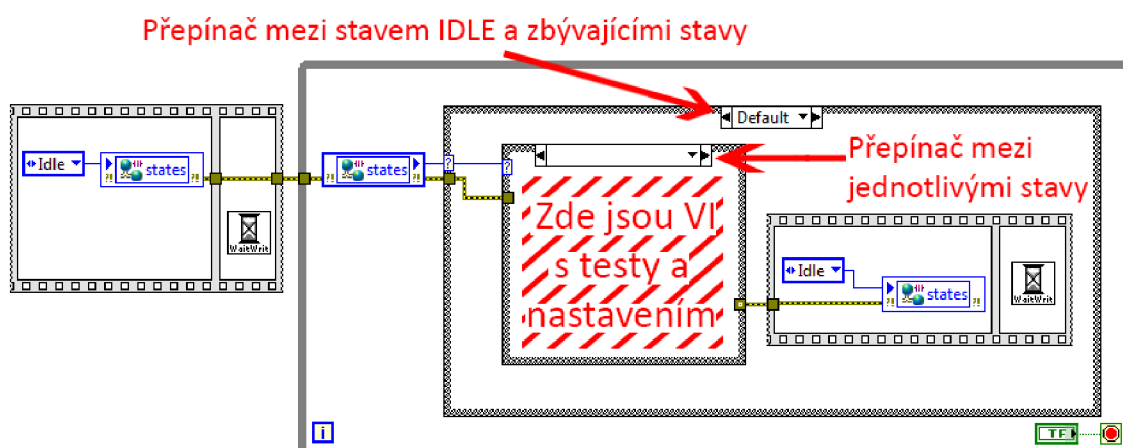
Posledním typem jsou Network-Published Shared Variables. Jak již bylo zmíněno výše, umožňují tyto proměnné přenášet data v rámci sítě Ethernet. Tento typ proměnných je použit pro přenos dat mezi PC a platformou cRIO, jež jsou oba připojeny v síti společnosti UNIS, a.s.

Pro přenos vstupních a výstupních dat pro testy nebo nastavování jednotlivých přístrojů testovacího pracoviště byly vytvořeny sdílené proměnné pro každý datový typ, který je v rámci aplikace používán. Jelikož může nastat situace, kdy vstupem či výstupem je více proměnných téhož typu, jedná se vždy o pole. Konkrétně to jsou pole dvouhodnotových proměnných boolean, double, integer, string, error a dále proměnné s vlastním datovým typem states a relay settings. Význam posledních dvou datových typů bude v následujících kapitolách upřesněn.

Jedním z problémů se sdílenými proměnnými, které byly řešeny v rámci této diplomové práce, byla nespolehlivost s přenosem dat, která byla způsobena kolísající odezvou ve vytižené firemní síti. Zápis dat do síťových proměnných není nijak potvrzovaný a při vykonávání programu pokračuje LabVIEW ve vykonávání dalších operací i přes to, že proměnná nemusela být ještě zapsána. Proto docházelo k situacím, kdy se například program běžící na PC snažil vyčítat hodnoty, které měly být do proměnné zapsány z cRIO, ale PC je vyčetl ještě před tím, než byly skutečně k dispozici. Z tohoto důvodu bylo nutné upravit časování vykonávání programu a to tak, že za každým zápisem do sdílené proměnné bylo přidáno dostatečné zpoždění. Aby bylo zajištěno potvrzení o zápisu do sdílené proměnné, bylo by nutné po každém zápisu začít vyčítat ve smyčce z téže proměnné a pokračovat v chodu programu po jejím úspěšném vyčtení.

Implementace softwaru pro platformu cRIO

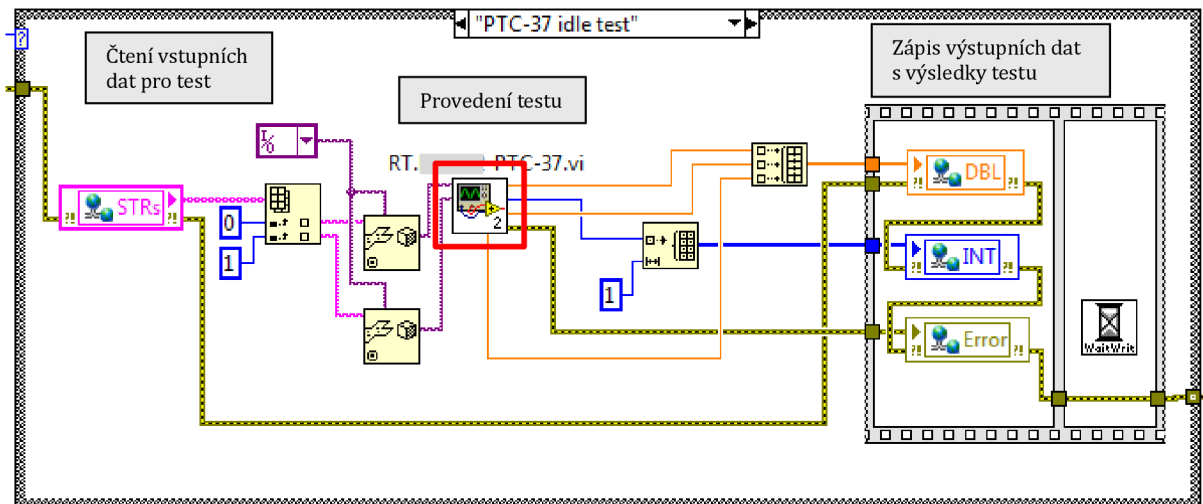
Vytvořený stavový automat je řízen pomocí sdílené proměnné *states*. Sdílená proměnná *states* je datového typu enum, ve kterém jsou definovány všechny stavy automatu. Jednotlivé stavy představují veškeré operace, jež jsou pomocí cRIO prováděny s testovaným zařízením a zařízením testovacího pracoviště, jako jsou zdroje napájení, klimatická komora, osciloskop apod. V případě, že cRIO zrovna neprovádí žádnou operaci, nachází se program ve stavu *Idle*. Tento stav je nastaven ještě před tím, než program vstoupí do nekonečné smyčky. Stav *Idle* obsahuje pouze blok pro zpoždění, aby cRIO zbytečně nepracovalo na maximální výkon. S každou iterací nekonečné smyčky je vyčtena sdílená proměnná *states*. Dále je pomocí *Case Structure* (LabVIEW obdoba switch case konstrukce) přepnuto do příslušného stavu, kde je vykonána žádaná operace. Po vykonání této žádané operace je do proměnné *states* znovu zapsána hodnota *Idle*, tím přejde cRIO zpět do klidového stavu.



Obr. 19 Zobecněný popis stavového automatu pro vykonávání jednotlivých testů a nastavení hardwaru testovacího pracoviště.

Jakmile dojde k přepnutí do zvoleného stavu, vykonání dané operace je pro všechny stavy podobné a skládá se ze tří fází. Pokud operace vyžaduje nějaká vstupní data, např. hodnota napětí a proudu, které se mají nastavit na zdroji napětí, jsou ze sdílených proměnných tyto data vyčteny. Tyto data slouží jako vstupní proměnné do VI, které již vykonává samotné měření veličin,

nastavování parametrů zařízení, vyčítání hodnot ze zařízení apod. Pokud jsou výsledkem operace nějaké naměřené hodnoty, jsou tyto hodnoty opět zapsány do sdílených proměnných, aby mohly být následně vyčteny z PC. Implementace jedné takové operace, která vyžaduje jak čtení, tak následné zapsání výsledků do sdílených proměnných, je zobrazena v blokovém diagramu na Obr. 20. Tento blokový diagram ukazuje jeden ze stavů a je umístěn v červeně vyšrafované oblasti na Obr. 19.



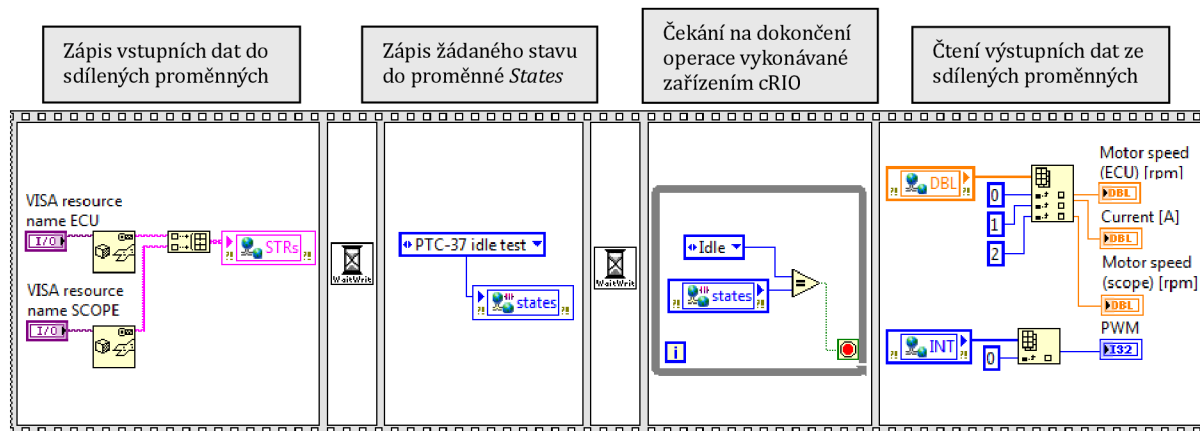
Obr. 20 Vyčtení vstupních dat, provedení testu (označeno červeným rámečkem) a zápis výstupních dat s výsledkem testu.

V tomto případě se jedná o měření a vyčítání několika veličin při testu motoru bez zátěže. Tento test vyžaduje jako vstupy dvě VISA reference a to pro řídicí jednotku a pro osciloskop Rohde & Schwarz. VISA reference je pro přenos přes sdílené proměnné převedena do speciálního řetězce binárních hodnot (flattened data string). Při vyčítání vstupních dat je potřeba dávat pozor na to, v jakém pořadí byly tyto dvě VISA reference zapsány do pole. V případě nedodržení pořadí pořadí referencí by mohlo dojít k záměně reference na řídicí jednotku a osciloskop. Následuje převod pole na dva prvky, dále jejich konverze z binárního řetězce zpět na datový typ pro VISA referenci. Pak dojde k vykonání samotného testu. Implementace několika konkrétních testů je popsána v kapitole 7.5. Po vykonání testu jsou výsledky převedeny dle svých datových typů do pole a následně zapsány do sdílených proměnných. Způsob implementace je pro většinu zbývajících testů velmi podobný, s tím, že dochází k obměně jednotlivých VI, které vykonávají konkrétní testy na základě požadavků testu.

Implementace softwaru pro PC

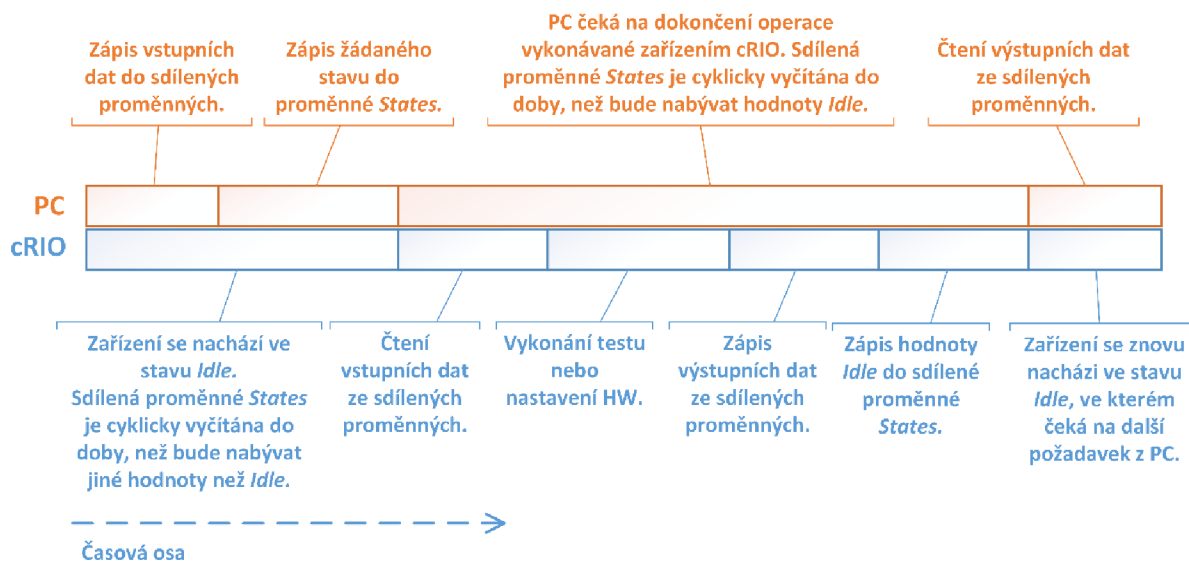
Software vytvořený v prostředí LabVIEW na PC slouží jako rozhraní mezi RT platformou cRIO a TestStandem. Každý test je tvořen jedním VI, jež jsou volány v jednotlivých krocích v prostředí TestStandu. Implementace těchto VI je až na výjimky pro všechny testy obdobná. Proměnné vstupující do daného VI jsou podle svých datových typů zabaleny do pole, které je zapsáno do sdílené proměnné příslušného datového typu. Po zapsání všech vstupních proměnných následuje zápis do proměnné *states*, čímž dojde k přepnutí cRIO do stavu, kdy je vykonána požadovaná operace. Dále následuje smyčka, ve které je vyčítána proměnná *states*. V případě, že proměnná *states* nabývá hodnoty *Idle*, znamená to, že operace na cRIO byla dokončena a ve sdílených proměnných jsou připraveny výsledné naměřené hodnoty, to pro případ, že se jedná o nějaký test, např. při nastavování zdroje napětí, stavu jednotlivých relé není žádný výstup požadován. Pokud je tedy vyžadován nějaký výstup, jsou tyto hodnoty vyčteny ze sdílených proměnných a přiřazeny

na výstup tohoto VI. Vyhodnocení výstupu probíhá až v TestStandu, stejně tak jako zadávání konkrétních vstupních hodnot pro jednotlivé testy.



Obr. 21 Implementace VI volaného z TestStandu pro test motoru bez zátěže.

Na Obr. 21 lze vidět implementaci VI, které je volané na PC a jedná se o test komunikace s řídicí jednotkou. Pro lepší představu časové návaznosti akcí, které PC a cRIO paralelně vykonávají, jsou tyto jednotlivé kroky schematicky vyobrazeny na Obr. 22.



Obr. 22 Časová posloupnost akcí na PC a zařízení cRIO během vykonávání testu.

7.5 Realizace vybraných testů a nastavení přístrojů

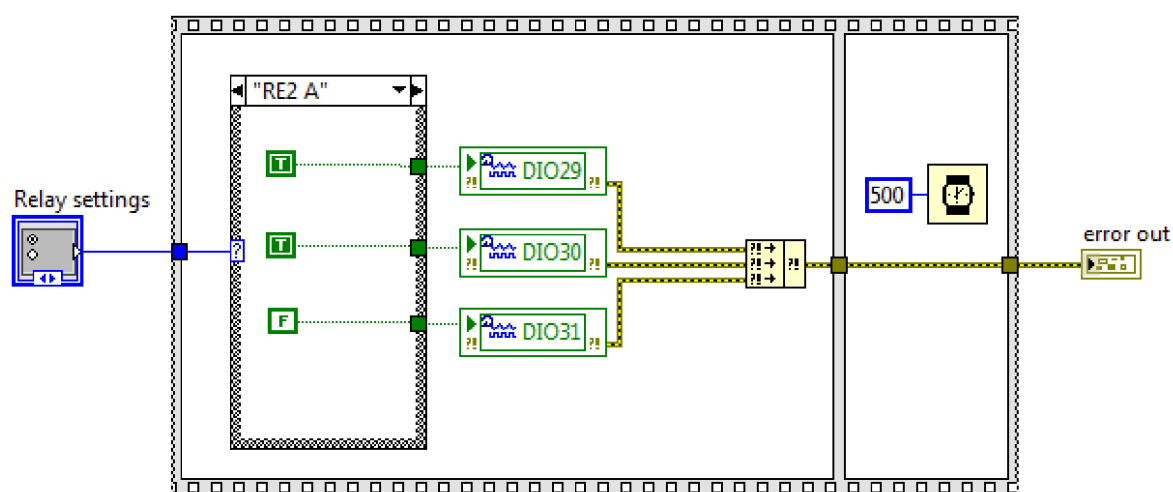
V této kapitole je rozebrána implementace vybraných testů elektronické řídicí jednotky a nastavení hardwaru k jejímu testování. Při pohledu na celý software se jedná o VI, které je spouštěno na platformě cRIO v definovaných stavech (označeno červeným rámečkem na Obr. 20).

Nastavení relé v přizpůsobovacím adaptéru

Jedním z velice častých úkonů během testování je nastavování tří relé, které slouží pro spínání různých obvodů v přizpůsobovacím adaptéru. Tyto obvody jsou v adaptéru tři, dva pro měření odběru proudu řídicí jednotkou a jeden, který slouží pouze k napájení řídicí jednotky. Obvod

určený pouze k napájení řídicí jednotky je dimenzován na podstatně větší proud, který při testech s motorem v zátěži tímto obvodem prochází.

V jednom okamžiku musí být sepnutý pouze jeden z obvodů, nikoli více najednou, jinak by mohlo dojít k poškození adaptéru. Z důvodu eliminace chyb při častém nastavování každého relé zvláště, jsou tyto relé spínány vždy najednou. Vstupem do tohoto VI je proměnná *relay settings*, což je proměnná typu enum s definovanými čtyřmi hodnotami. První slouží pro případ, kdy všechna relé jsou v rozepnutém stavu, ve smyslu že jimi neprotéká proud. Výjimkou je relé sloužící pro sepnutí hlavního napájecího obvodu, má totiž invertovanou logiku, tj. při nastavení digitálního vstupu na logickou hodnotu 1 jím proud neprotéká. Další stavy pak kombinují digitální vstupy tak, aby byl sepnutý vždy jen jeden obvod. Po nastavení všech relé je dále nastavena prodleva pro stabilizaci napětí v obvodu. Blokový diagram tohoto VI lze vidět na obrázku *Obr. 23*.

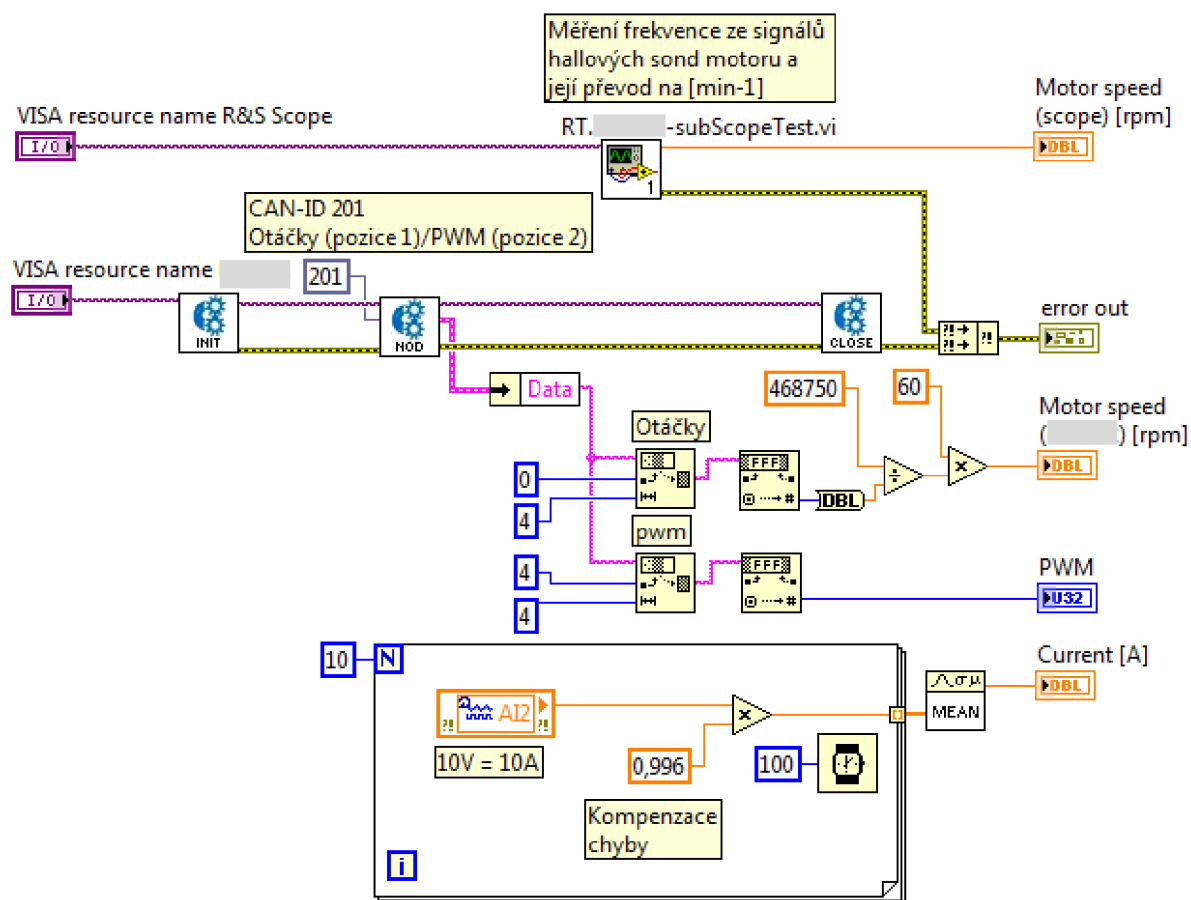


Obr. 23 Ovládání relé v přizpůsobovacím adaptéru.

Test motoru bez zátěže

Vhodným příkladem testu je zkouška motoru bez zátěže, jelikož v sobě zahrnuje většinu procedur, které jsou použity i u jiných testů. Tento test probíhá za stavu, kdy motor běží, ovšem na olejové nádrži pro simulování zátěže je zcela otevřený ventil, takže nádoba není tlakována. Během tohoto testu se měří odebíraný proud řídicí jednotkou pomocí přizpůsobovacího adaptéru, dále jsou z jednotky vyčítány naměřené otáčky a hodnota PWM. Poté je za pomoci osciloskopu změřena frekvence signálů generovaných z Hallových sond umístěných na motoru. Tato hodnota je pak

porovnávána s hodnotou vyčtenou z řídicí jednotky. Toto porovnávání probíhá až v rámci testovací sekvence v TestStandu. Blokový diagram tohoto testu lze vidět na Obr. 24.

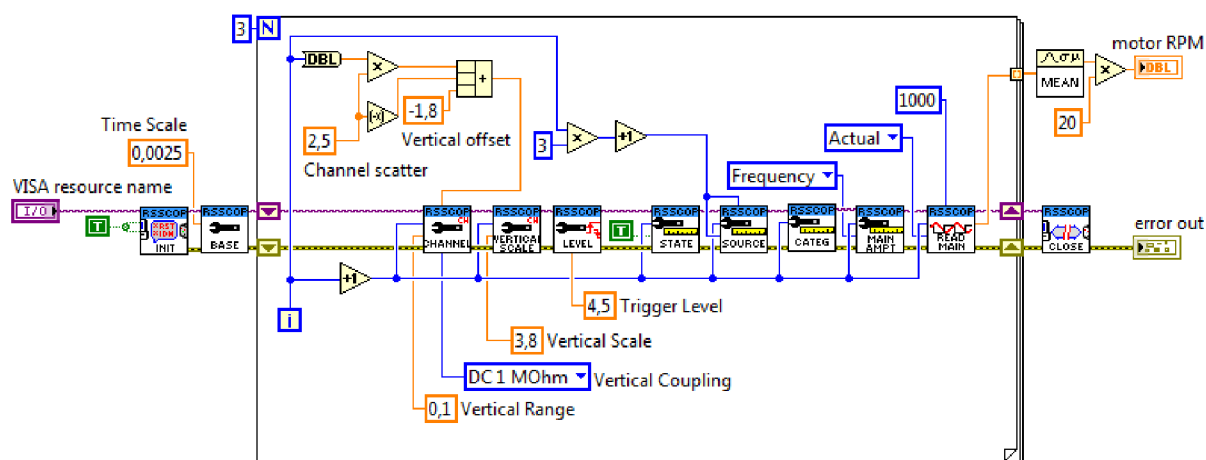


Obr. 24 Implementace testu motoru bez zátěže.

Velikost odebíraného proudu jednotkou je vyčítána na příslušném pinu z analogového výstupu z přizpůsobovacího adaptéru. Adaptér je před zahájením tohoto testu nastaven do stavu pro měření odběru proudu v rozsahu 0 – 10 A. Jelikož 10 A odpovídá maximálnímu napětí 10 V na výstupu, není třeba hodnoty přepočítávat. Jediná operace s naměřenou hodnotou je její vynásobení konstantou, pro kompenzaci chyby měření. Tato konstanta byla obdržena z kalibrace adaptéru po jeho výrobě.

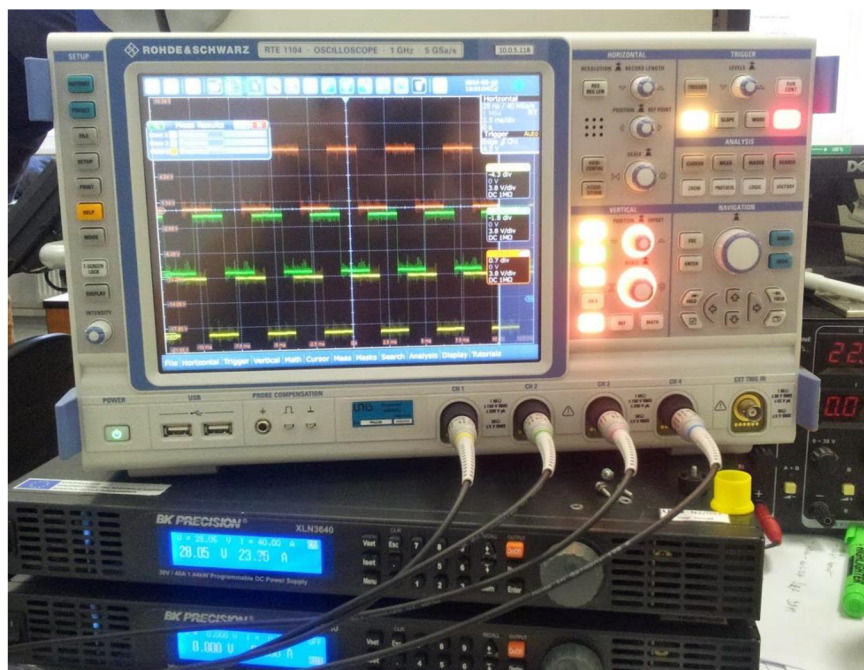
Velikost otáček a hodnota PWM je jednotkou vysílána periodicky pomocí zprav typu NOD. Otáčky i hodnota PWM jsou vysílány v jedné zprávě a to na CAN-ID 201. K vyčtení této zprávy byla použita knihovna pro komunikaci s řídicí jednotkou (viz kapitola 7.2). Po vyčtení zprávy je z clusteru obsahující přijatý rámec vyjmuta datová část zprávy. Tuto datovou část zprávy tvoří 32 bitů. Z toho prvních 16 bitů obsahuje hodnotu o otáčkách a zbylých 16 bitů hodnotu PWM.

Poslední částí testu je zjištění reálných otáček motoru na základě signálu z Hallových sond umístěných na motoru. Blokový diagram na *Obr. 25* zobrazuje, jakým způsobem je provedeno měření reálných otáček na motoru.



Obr. 25 Nastavení osciloskopu Rohde & Schwarz a měření frekvence generovaných signálů z Hallových sond závislých na otáčkách motoru.

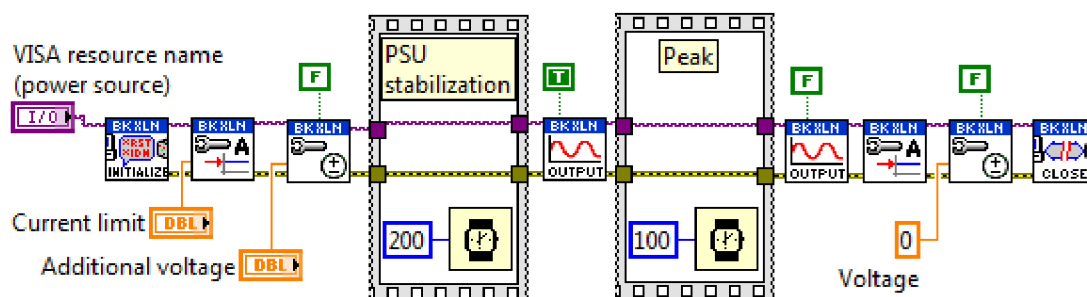
Pro vzdálené ovládání osciloskopu byla použita knihovna, která je volně ke stažení na webových stránkách výrobce. Nejprve jsou nastaveny měřítka pro každou osu, offsety jednotlivých kanálů, aby obsluha mohla vizuálně zkontrolovat průběhy signálů z Hallových sond. Dále je nastaven trigger do požadované úrovně s reakcí na náběžnou hranu. Nakonec je spuštěno měření frekvence signálu, což je přímo jedna z funkcí osciloskopu. Frekvence je změřena na každém kanálu zvlášť a poté jsou tyto naměřené hodnoty zprůměrovány, jak je znázorněno na *Obr. 25*. Na *Obr. 26* lze vidět průběhy signálů z tří Hallových sond poté, co byl osciloskop tímto VI během tohoto testu automaticky nastaven.



Obr. 26 Generované signály z Hallových sond na osciloskopu Rohde & Schwarz.

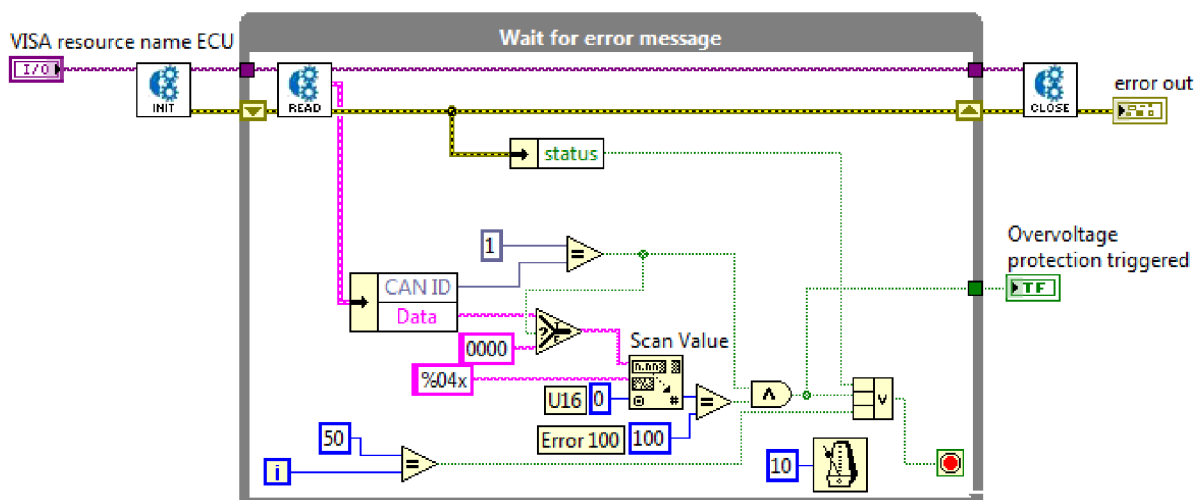
Kontrola přepětové ochrany

Cílem tohoto testu je ověřit, zdali přepětová ochrana řídicí jednotky správně zareaguje na definovaných limitech. Test probíhá tak, že na napájení jednotky je přivedeno napětí 48 V na velice krátkou dobu, pro tuto úroveň napětí nesmí přepětová ochrana zareagovat. Dále následuje ten samý test, ale s napájecím napětím 64 V, pro tento případ již ochrana zareagovat musí a zároveň musí řídicí jednotka vyslat po sběrnici CAN EED zprávu signalizující příslušnou chybu. Pro provedení testu je nutné vytvořit velice krátký impuls přepětí, což byl z počátku problém, protože zdroje jsou ovládány analogově se stále zapnutým výstupem. Přes analogové ovládání nelze zapínat a vypínat výstup na zdroji. Použité zdroje ve své konfiguraci umožňují vzdálené ovládání pouze analogovými vstupy a pomocí USB. Zdroje přes USB nelze k zařízení cRIO připojit pokud je použit Real-Time operační systém, z tohoto důvodu muselo být ovládání zdroje provedeno z PC. Při ovládání zdroje přes USB je k dispozici široká škála parametrů, které se dají na zdroji nastavovat. V tomto případě ale stačilo možnost spínat výstup a nastavovat velikost napětí a proudu.



Obr. 27 Ovládání zdroje napětí pro test přepětové ochrany řídicí jednotky.

Dalším problémem bylo zachycení EED zprávy, která je řídicí jednotkou vyslána po zareagování na přepětí. Tato zpráva je vyslána pouze jednou a to ihned po výskytu události. Aby nedošlo k tomu, že se zpráva nestihne vyčíst, je nutné spustit vyčítání EED zpráv ještě předtím, než dojde k nastavení zdroje na přepětí. Toto vyčítání je obstaráno zařízením cRIO. Čítač těchto zpráv je jedním ze stavů stavového automatu a je spuštěn zápisem požadovaného stavu do sdílené proměnné *states*, stejně jako tomu je při volání testů.



Obr. 28 Čítač EED zpráv.

Pro tento test bylo nutné využít oba napájecí zdroje. Jeden zdroj napájí řídicí jednotku, tak aby se nacházela v provozním režimu, a druhý zdroj je připnut pouze na velmi krátkou chvíli, tak aby bylo způsobeno přepětí. Jeden zdroj není schopný dosáhnout požadovaného napětí, jelikož maximální výstupní napětí použitých zdrojů je 36 V. Na *Obr. 27* je zobrazen blokový diagram, který nastavuje a spíná výstup zdroje na 100 ms. Délka této doby byla zjištěna měřením výstupu ze zdroje pomocí osciloskopu.

Čítač EED zpráv je realizován pomocí bloků z knihovny pro komunikaci s řídicí jednotkou. Jeho blokový diagram je zobrazen na *Obr. 28*. Zpráva signalizující přepětí v řídicí jednotce je označena identifikátorem CAN-ID 1, a proto je CAN-ID každé zprávy vyčtené z jednotky porovnáno právě s touto hodnotou. Je-li splněn předpoklad, je zastaven čítač a do sdílené proměnné předána logická hodnota 1, jako výsledek úspěšného zachycení zprávy. Pokud EED zpráva není vyčtena v definovaném časovém limitu, je čítač opět zastaven, ale tentokrát s výsledkem logická 0.

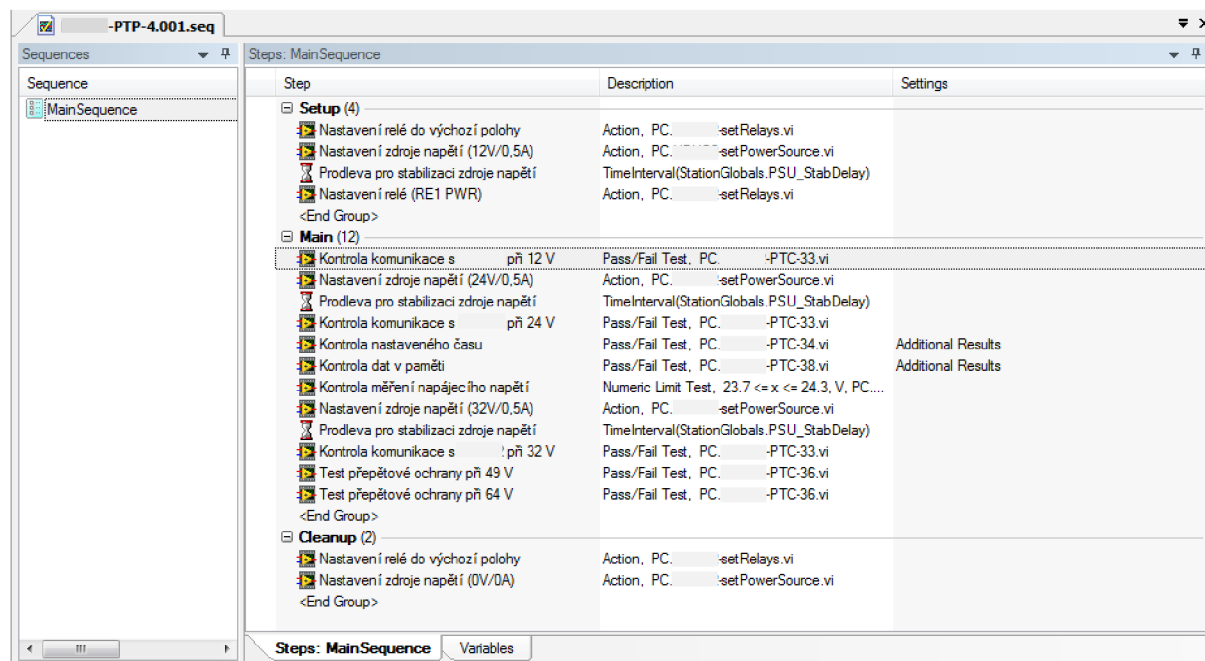
7.6 Tvorba testovacích sekvencí v prostředí TestStand

Časová návaznost jednotlivých akcí v TestStandu od spuštění testování až po jeho ukončení jsou řízeny tzv. procesním modelem. Procesní model se skládá ze sady kroků, které určují posloupnost vykonávání všech operací v TestStandu. Procesní model je soubor sekvence, která obsahuje několik dalších sekvencí. Velkou část těchto sekvencí tvoří tzv. callbacky. Většina callbacků je výchozím nastavením skrytá, ale je možné je v nastavení modelu zobrazit a tím získat kontrolu nad všemi akcemi, které TestStand vykonává. To je velmi důležité, jelikož součástí testování elektronické řídicí jednotky je její vyhřívání v teplotní komoře po dobu jedné hodiny. Testování více jednotek by tak bylo časově velmi náročné. Z toho důvodu je nutné zasáhnout do modelu a zobrazit požadované callbacky, v tomto případě *PreUUTLoop* a *PostUUTLoop*. Ke spuštění teplotní komory pak dochází v *PreUUTLoop* callbacku. Tento callback TestStand volá předtím, než vstoupí do smyčky, ve které již probíhá volání sekvencí souboru obsahující testovací procedury pro elektronickou řídicí jednotku (viz. *Obr. 3*). Na začátku každého cyklu je obsluha dotázána, zdali bude testováno ještě další zařízení, pokud ne, je smyčka ukončena. Po skončení smyčky je zavolán *PostUUTLoop* callback, ve kterém je teplotní komora zastavena. Podobným způsobem lze do modelu zasahovat i pro další účely, jako je otevření okna obsahující tabulku s údaji do hlavičky dokumentu a také zavolání VI, které z vygenerovaného XML souboru vytvoří testovací protokol v dokumentu typu Microsoft Excel.

Algoritmus testování je znázorněn na *Obr. 31*. Po spuštění testovacího softwaru se otevře okno s tabulkou, kde obsluha napíše informace o použitých testovacích zařízeních, své jméno a další náležitosti patřící do dokumentu s testovacím protokolem. Toto vyskakovací okno (*Obr. 35*) bylo implementováno v LabVIEW a je voláno pouze jednou před spuštěním testovacího softwaru. Následuje zavolání VI, které spustí teplotní komoru a vyhřeje řídicí jednotky po dobu 1 hodiny. Poté vstoupí TestStand do smyčky, jejíž každá iterace začíná dotazem na sériové a produktové číslo testované řídicí jednotky. Toto okno obsahuje i tlačítko stop pro ukončení testování. Po vyplnění požadovaných údajů a stisknutím potvrzovacího tlačítka dojde ke spuštění sekvencí souboru s testovacími procedurami pro řídicí jednotku. Po dokončení testovací sekvence je zavoláno VI, které vygeneruje testovací protokol v souboru typu Microsoft Excel. Po vygenerování testovacího protokolu je znovu zobrazeno okno pro zadání sériového a produktového čísla další testované jednotky (nutnou podmínkou je přepojení kabeláže, která je vyvedena z teplotní komory), pokud toto byla poslední jednotka, je ukončena smyčka a TestStand přejde k dalšímu kroku, kde je vypnuta teplotní komora a ukončen celý testovací proces.

Testovací sekvence v TestStandu jsou rozděleny do tří fází. První fáze je *Setup*, pro vykonání operací před spuštěním testování. Následuje *Main*, kde probíhá hlavní část sekvence, což je samotné testování. A poslední fází je *Cleanup*, který slouží pro nastavení testovacího hardwaru zpět do původního stavu. Výhodou je, že TestStand umožňuje zavolat *Cleanup* poté, když by došlo k nějaké chybě při testování. Tím je zajištěno, že testovací hardware nezůstane v případě chyby

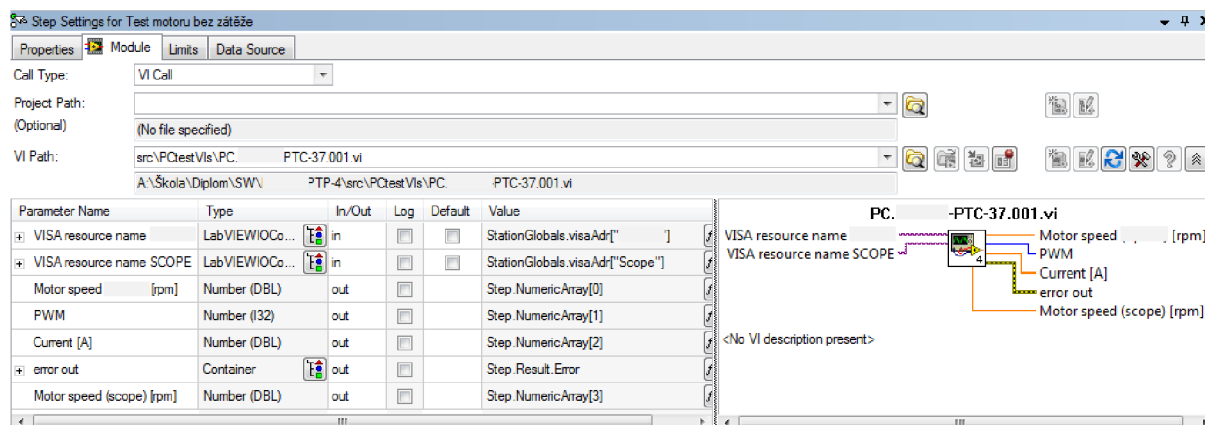
v nějakém nežádoucím stavu. Ukázková sekvence v TestStandu, resp. jedna testovací procedura je zobrazena na Obr. 29.



Obr. 29 Testovací sekvence v TestStandu.

Testovací sekvence je složena z několika kroků. Jedním takovým krokem je např. test motoru bez zátěže. Ukázka tohoto testu je volena záměrně, jelikož byl poměrně detailně popsán v předchozích kapitolách a tím budou lépe viditelné souvislosti mezi jednotlivými částmi softwaru.

Výsledek tohoto testu je určen porovnáním naměřených hodnot se zadaným rozsahem očekávaných hodnot. Jelikož je těchto hodnot více, je pro tento druh testů definován v TestStandu krok typu *Multiple Numeric Limit Test*. Hlavní nastavení kroku je zobrazeno na Obr. 30. Nejprve je nutné přiřadit příslušné VI k tomuto kroku. Po přiřazení se v okně s nastavením zobrazí toto VI včetně tabulky, jež obsahuje všechny jeho vstupy a výstupy. Až zde jsou nastavovány konkrétní parametry pro testy, jako např. VISA reference, velikost napájecího napětí, proudové omezení napájecího zdroje apod.



Obr. 30 Okno s nastavením kroků tvořící testovací sekvenci v TestStandu.

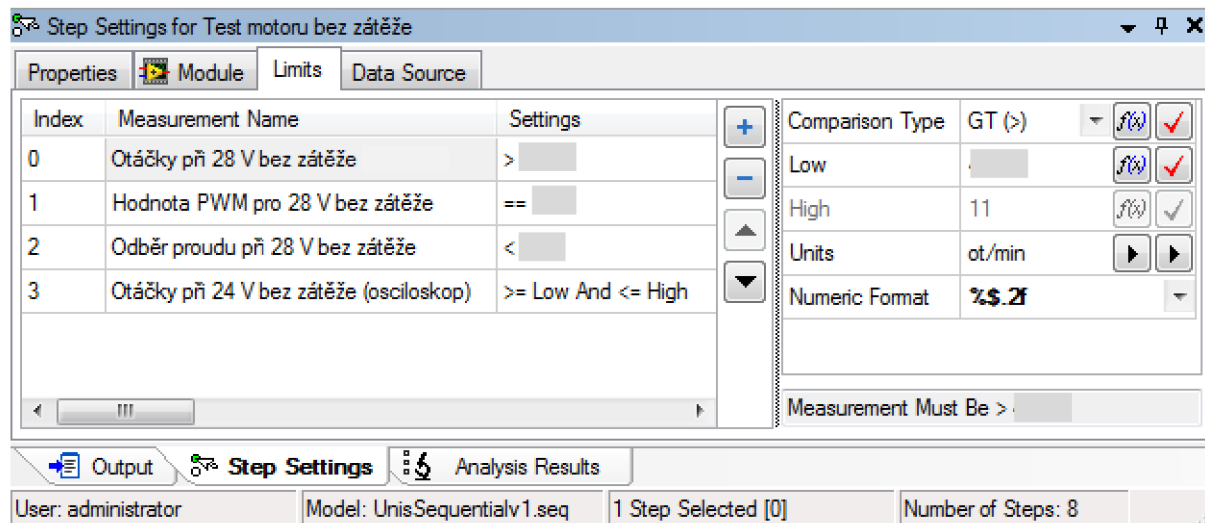
Následující obrázek (Obr. 31) obsahuje algoritmus znázorňující vykonávání testování. Tento algoritmus zjednodušeně popisuje klíčové kroky upraveného procesního modelu v TestStandu.



Obr. 31 Algoritmus znázorňující vykonávání testovací sekvence.

Další důležitou záložkou je nastavení limitů, které musí naměřená veličina splňovat (Obr. 32). V tomto okně se nastavuje typ porovnávání, výsledný číselný formát a také jednotky měřené

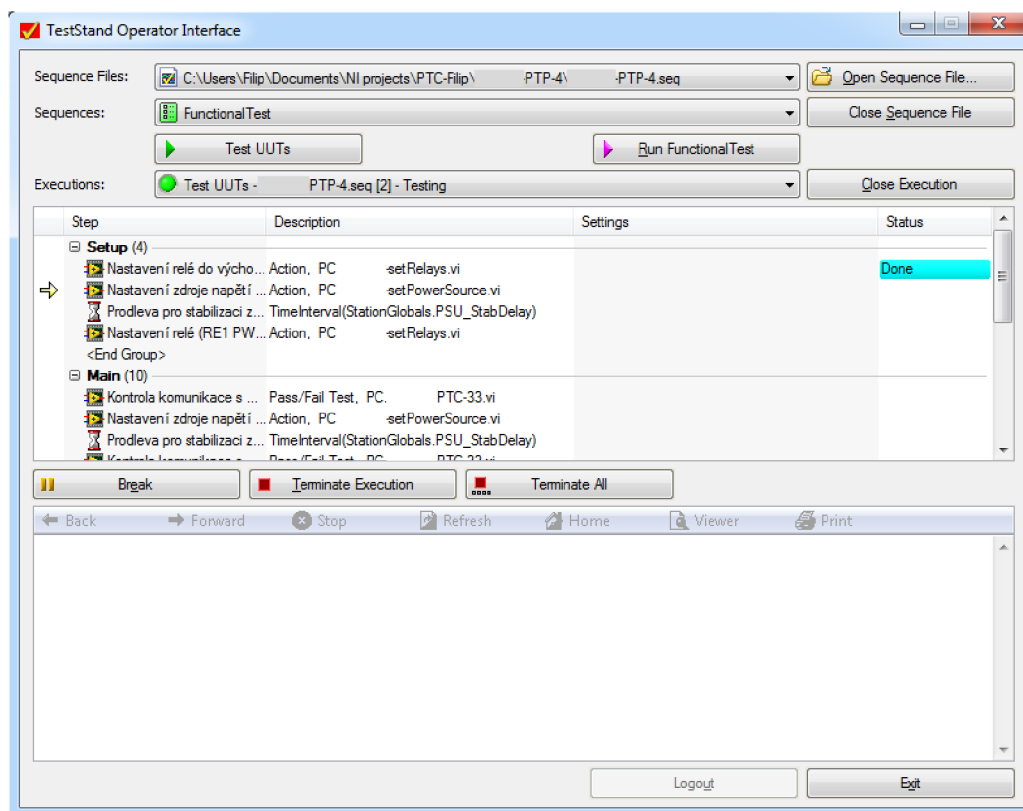
veličiny. Tyto údaje jsou pak z vygenerovaného XML vyjímány a dále použity ve výsledném testovacím protokolu.



Obr. 32 Okno s nastavením limitů pro naměřené veličiny.

7.7 Uživatelské rozhraní pro obsluhu

Na počítači, který bude využíván pro spuštění testovacích sekvencí po zavedení testovacího pracoviště do výroby, nebude nainstalována plnohodnotná vývojářská licence TestStandu, ale pouze omezená licence, která obsahuje TestStand Engine. Tato licence umožňuje spustit uživatelské rozhraní pro operátora a načítat soubory sekvencí TestStandu.



Obr. 33 Uživatelské rozhraní pro operátora testovacího pracoviště.

Používají se dva hlavní přístupy, jak spouštět testovací sekvence. První je vytvořit si vlastní uživatelské rozhraní, které bude volat TestStand API. Pokud nejsou vyžadovány nějaké speciální požadavky na uživatelské rozhraní, je zde další možnost a to použít předdefinované uživatelské rozhraní, které jsou dodávány společně s TestStandem. Tyto předdefinované rozhraní jsou vytvořeny v různých programovacích jazycích a jejich zdrojové kódy jsou přístupné v adresáři TestStandu. To znamená, že není problém upravovat předdefinovaná uživatelská rozhraní, čímž si lze přizpůsobit rozhraní dle požadavků a není nutné jej vytvářet celé od začátku.

Pro spouštění testovacích sekvencí na testovacím pracovišti vytvářeným v rámci této diplomové práce bylo využito základní uživatelské rozhraní pro operátora, které je dodáváno spolu s TestStandem. Panel uživatelského rozhraní lze vidět na obrázku *Obr. 33*.

7.8 Generování testovacího protokolu

TestStand nabízí možnost ukládání testovacího protokolu v několika formátech (HTML, ASCII, XML, ATML). Možnosti změny vzhledu výstupního dokumentu, případně možnost výběru informací, které budou v konečném dokumentu obsaženy, jsou sice možné, ale pro výraznější úpravy dokumentu velmi těžko dosažitelné. Ze strany společnosti UNIS, a.s. bylo vyžadováno, aby byl protokol z testování ve formátu Microsoft Excel a zároveň dodržoval veškeré náležitosti, které jsou ve společnosti UNIS, a.s. zavedeny pro tento druh dokumentace.

Výsledný protokol se skládá ze tří částí. První částí je hlavička dokumentu, kde jsou obsaženy informace o použitém hardwaru jak pro testování, tak o samotném testovaném zařízení a další údaje jako je název testu, odkazy na potřebnou dokumentaci apod. Druhou část tvoří tabulka s výsledky jednotlivých testů. Součástí této tabulky jsou pouze testy a nikoliv všechny kroky vykonané během testu, jako třeba nastavování zdroje napětí, spínání relé apod. Tabulka reprezentující jednotlivé testy je tvořena sloupci s identifikátorem, popisem, očekávanou hodnotou, naměřenou hodnotou a celkovým výsledkem testu, tj. zda test prošel nebo neprošel. Poslední částí protokolu je zápatí obsahující celkový výsledek zkoušky, datum, jméno a podpis osoby zodpovědné za provedení zkoušky.

Jelikož TestStand neumožňuje výstup ve formátu XLS (popř. XLSX), bylo nutné implementovat vlastní prostředky k tomu, aby měl testovací protokol požadované náležitosti a formát. Z TestStandu byl zvolen jako výstupní formát XML (bez formátování). Toto XML je zpracováno pomocí XML parseru, jež je součástí LabVIEW. Následuje vytvoření XLS souboru, do kterého jsou importovány zpracovaná data z XML souboru. Tato operace je zajištěna doplňkem LabVIEW Report Generation Toolikt pro Microsoft Office. Tento nástroj umožňuje vytvářet a upravovat soubory typu Microsoft Word, Excel a to včetně vkládání tabulek, grafů, obrázku nebo formátování. Ovšem jeho možnosti byly pro tuto aplikaci nedostatečné a formátování velmi neefektivní. Z toho důvodu byl využit pouze pro vytvoření Excel souboru z předem připravené šablony, vložení tabulky s výsledky testů a následně zavolání předdefinovaného makra v Excelu. Toto makro naformátuje celý dokument a také vloží příslušné informace do hlavičky a zápatí.

Zpracování XML souboru s výsledky testů

LabVIEW obsahuje v základním balíčku nástrojů bloky určené pro práci s XML dokumenty. XML soubor generovaný TestStandem je velmi rozsáhlý a detailní, ale pro výsledný testovací protokol stačí pouze několik údajů, jako je popis testu, naměřené hodnoty, očekávané hodnoty a výsledek testu.

XPath (XML Path Language) je jazyk, pomocí kterého lze adresovat části XML dokumentu. Pomocí tohoto jazyka lze z XML dokumentu vybírat jeden či více elementů a dále pracovat s jejich hodnotami a atributy. LabVIEW podporuje XPath verze 1.0, která je standardem vydaným organizací W3C [35].

Na testovacím protokolu mají být zaznamenány jen výsledky testů a nikoliv každý provedený krok, jako je nastavování zdrojů napětí, prodlevy pro ustálení zdrojů apod. Nejprve je tedy nutné

vybrat z XML souboru pouze ty elementy, které obsahují vybrané informace o provedeném kroku a mají charakter testu. Toho je docíleno následujícím XPath výrazem:

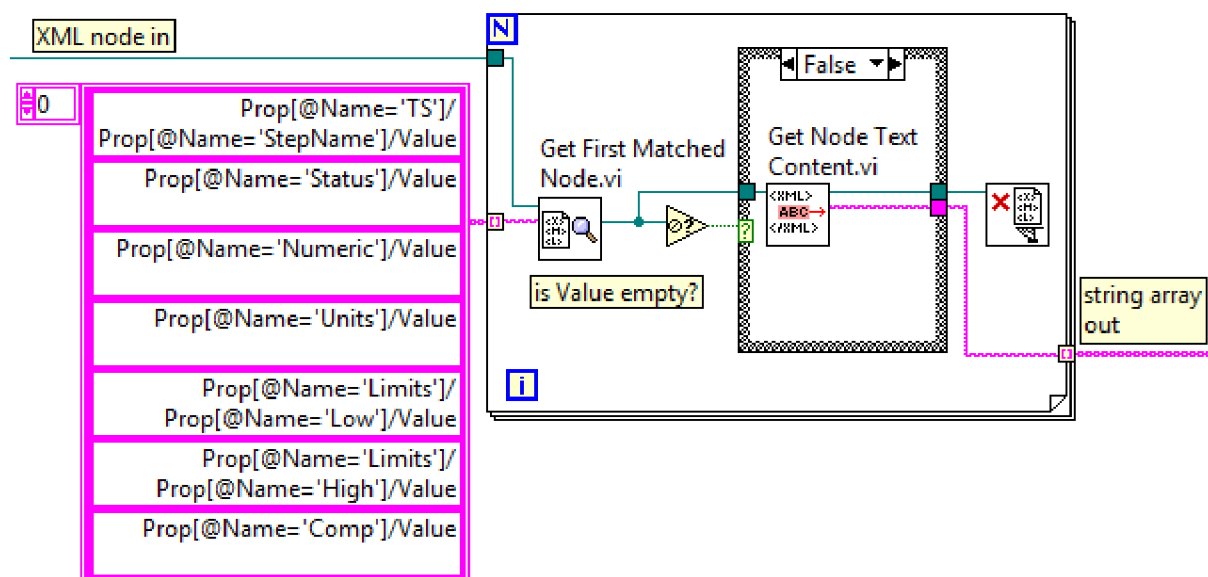
XPath výraz pro vyjmutí prvků obsahující informace pouze o testech

```
/Reports/Report[last()]/Prop[(@Type='TEResult') and
((Prop[@Name='TS']/Prop[@Name='StepType']/Value/text() = "NumericLimitTest") or
Prop[@Name='TS']/Prop[@Name='StepType']/Value/text() = "StringValueTest") or
(Prop[@Name='TS']/Prop[@Name='StepType']/Value/text() = "PassFailTest") or
(Prop[@Name='TS']/Prop[@Name='StepType']/Value/text() = "NI_MultipleNumericLimitTest"))]
```

Kromě pořadového čísla testu, které je automaticky doplněno až v Excelu, jsou všechny informace získány z XML souboru. Každý řádek obsahující informace o testu na testovacím protokolu je tvořen těmito sloupci:

- Pořadové číslo testu
- Výsledek testu (prošel/neprošel)
- Název testu, jeho stručný popis
- Očekávaná hodnota (rozsah hodnot)
- Naměřená hodnota
- Jednotky měřené veličiny

Blokový diagram na Obr. 34 ukazuje, jakým způsobem je v LabVIEW implementováno zpracování XML souborů. Konkrétně jak jsou z již z předzpracovaných XML dat vyjmuty potřebné informace z testu typu *Numeric Limit Test*.



Obr. 34 Zpracování XML dat v LabVIEW.

Export dat pomocí doplňku LabVIEW Report Generation Toolkit

Nejprve bylo nutné vytvořit šablonu, která je použita jako výchozí soubor pro LabVIEW Report Generation Toolkit. Šablona má stejný vzhled jako finální dokument, ale neobsahuje řádky s provedenými testy a tabulku s použitým testovacím hardwarem. Součástí šablony je již naprogramované makro, které je po vložení tabulek s výsledky a použitým testovacím hardwarem pomocí LabVIEW Report Generation Toolkit, zavoláno.

Blokové schéma tohoto VI pro generování testovacího protokolu je poměrně rozsáhlé, proto je zařazeno do příloh. Lze ho nalézt v příloze A.

Formátování dokumentu Excel pomocí maker

Je-li potřeba v aplikaci Microsoft Office Excel zautomatizovat nějaký opakující se sled úkonů a zefektivnit tak práci v této aplikaci, nabízí toto prostředí tvorbu maker v programovacím jazyku VBA (Visual Basic for Application).

Poté, co je z LabVIEW do testovacího protokolu vložena tabulka s výsledky testů a hlavičkou, je zavoláno makro, jehož parametry obsahují údaje nutné k automatickému naformátování dokumentu. Jedním z požadavků společnosti UNIS, a.s. bylo, aby mohly být tímto způsobem generovány testovací protokoly i pro jiné testy, proto bylo nutné implementaci zobecnit pro proměnlivý počet řádků v hlavičce dokumentu a samotných testech. Dále jsou pomocí parametrů tohoto makra předávány informace, které se netýkají přímo testů, ale jsou součástí hlavičky a zápatí testovacího protokolu, jako je název testu, jméno testera apod.

The screenshot shows a dialog box titled "Header" with the following structure:

- Testované zařízení**

Hardware	Verze
Software	Verze
- Konfigurace testu**

Název	PN	SN	Verze
- Informace o testu**

Název dokumentu:	
Popis testu:	
Výrobní postup:	
Revize výrobního postupu:	
Testovací SW:	
Jméno testera:	

At the bottom of the dialog is an "OK" button with a green checkmark icon.

Obr. 35 Okno pro vložení dodatečných informací do testovacího protokolu.

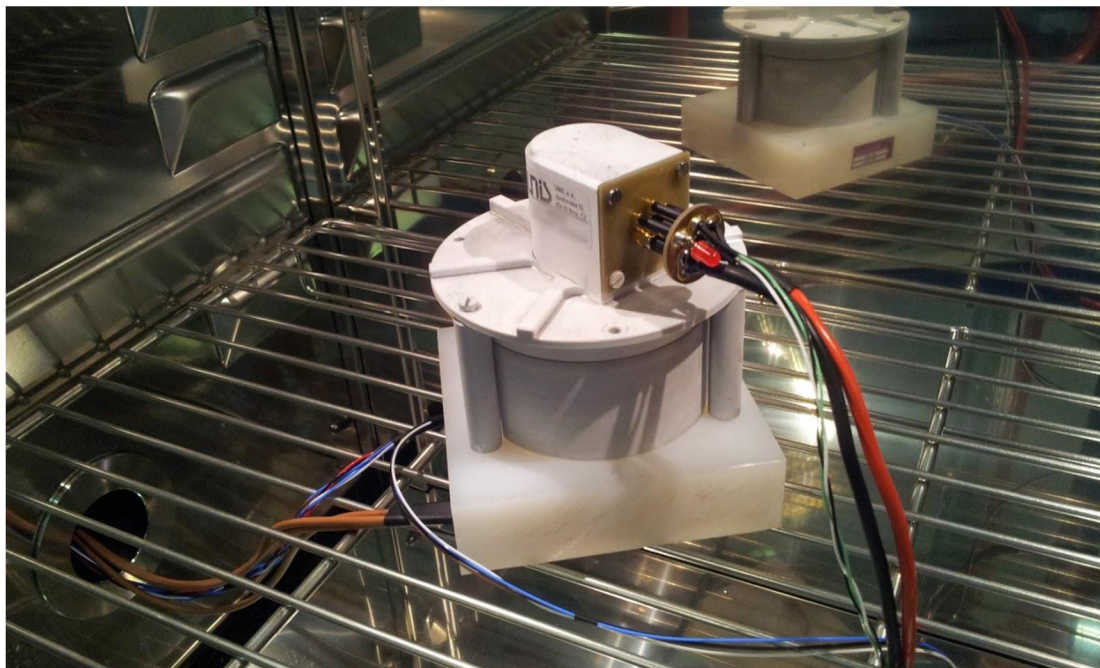
8 PRAKTICKÉ OVĚŘENÍ TESTOVACÍHO PRACOVIŠTĚ A DISKUZE

Součástí této diplomové práce je praktické ověření automatického testovacího pracoviště. To bylo uskutečněno otestováním několika reálných elektronických řídicích jednotek. Společnost UNIS, a.s. k tomuto účelu poskytla pět řídicích jednotek, které byly vyrobeny v tzv. nulté výrobní sérii. Tyto jednotky byly osazeny nestandardním konektorem z důvodu zaváděcích testů.

Celý test řídicí jednotky je možné rozdělit na dvě části. V první části jsou vykonávány testy, které vyžadují, aby byly prováděny při odpojeném motoru. Před další částí testů, které již probíhají s připojeným motorem, je obsluha upozorněna, aby připojila řídicí jednotku k motoru. Úkony, které musí vykonat obsluha, jsou následující:

- Spustit testovací sekvenci.
- Zkontrolovat a případně upravit údaje, které mají být v hlavičce a v zápatí na protokolu z testování.
- Zadat sériové a produktové číslo testované řídicí jednotky.
- Obsluha je upozorněna, aby zkontrolovala, zdali je odpojený motor a případně tomu tak učinila. Po tomto kroku je spuštěno testování.
- Po dokončení testů s odpojeným motorem je obsluha vyzvána k připojení motoru. Po potvrzení následují testy s připojeným motorem.
- Během testu s motorem se zátěží musí obsluha postupně uzavírat ventil na zkušebním stavu, čímž dochází k tlakování olejové nádrže. Test motoru se zátěží je jediný test, který nebylo možné plně zautomatizovat a o tom zda test prošel či neprošel, rozhodne obsluha na základě měření tlaku pomocí manometru umístěného na zkušebním stavu.

Výstupní test elektronické řídicí jednotky probíhá při nízké ($-55\text{ }^{\circ}\text{C}$) a vysoké ($70\text{ }^{\circ}\text{C}$) teplotě. Tyto dva testy jsou samostatně spustitelné sekvence, nicméně musí být zachováno pořadí spuštění těchto dvou sekvencí. Při ohřívání elektronické řídicí jednotky z $-55\text{ }^{\circ}\text{C}$ dochází na povrchu jednotky ke tvorbě krystalické námrazy. Z toho důvodu by měl být ihned po tomto testu spuštěn test při teplotě $70\text{ }^{\circ}\text{C}$, který řídicí jednotku zpět ohřeje a vysuší. Elektronickou řídicí jednotku pokrytou námrazou po ohřívání z $-55\text{ }^{\circ}\text{C}$ lze vidět na fotografii Obr. 36.



Obr. 36 Elektronická řídicí jednotka z nulté ověřovací série pokrytá krystalickou námrazou.

Celkem tedy bylo otestováno pět funkčních vzorků elektronických řídicích jednotek. Čtyři jednotky prošly bez jakýchkoli problémů testem při nízké i vysoké teplotě. Problém nastal u jedné jednotky, kdy došlo během testu motoru bez zátěže při nízké teplotě k poruše.

Celý průběh byl na základě těchto testů zaznamenán a ohlášen oddělení výroby na prošetření problému. S největší pravděpodobností se mohlo jednat o selhání některé součástky, anebo by mohla být tato závada důsledkem pochybení při výrobě této konkrétní řídicí jednotky.

Ukázkový testovací protokol reálné jednotky se nachází v příloze B. Číselné očekávané a naměřené hodnoty budou skryty z důvodu ochrany duševního vlastnictví společnosti UNIS, a.s.

Na fotografii Obr. 37 je možné vidět skutečnou podobu celého automatického testovacího pracoviště, které bylo použito pro otestování těchto pěti funkčních vzorků řídicích jednotek. Toto testovací pracoviště bude ve společnosti UNIS, a.s. nadále využíváno pro testování nově vyrobených elektronických řídicích jednotek.



Obr. 37 Fotografie automatického testovacího pracoviště.

9 ZÁVĚR

Cílem této diplomové práce bylo navrhnout automatické testovací pracoviště pro výstupní testy elektronické řídicí jednotky pomocí prostředků od společnosti National Instruments. Testovaná řídicí jednotka byla vyvíjena s cílem splnit normy RTCA DO-178, RTCA DO-254 a je tak určena především pro použití v oblasti letectví. V teoretických kapitolách je popsán životní cyklus vývoje systému podle V-modelu, který poukazuje na důležitost testování vyvíjeného produktu. Jelikož testování elektronické řídicí jednotky probíhá automaticky, byly v této diplomové práci popsány vybrané metody automatického testování elektroniky, jejich výhody, nevýhody a uplatnění v jednotlivých fázích výrobního procesu. Teoretická část se dále zabývá specifikami zkoušek elektroniky pro letectví, které definuje norma RTCA DO-160.

V další části práce jsou definovány pojmy testovací případ a testovací procedura. Ty slouží jako podklady pro tvorbu jednotlivých testů a testovacích sekvencí.

Automatické testovací pracoviště je založeno především na zařízení CompactRIO od společnosti National Instruments, které obsahuje analogové a digitální vstupní i výstupní karty a také kartu pro zajištění komunikace s periferními zařízeními, jako jsou napájecí zdroje, teplotní komora, osciloskop a speciální přípravky vytvořené pro testování této elektronické řídicí jednotky. Další důležitou komponentou testovacího prostředí je osobní počítač, který slouží pro spouštění testovacích sekvencí a generování protokolů z testování.

Celý software byl implementován v LabVIEW, stejně tak i komunikace s perifériemi. Jelikož je LabVIEW hojně využíváno pro vývoj měřicích a testovacích aplikací, spousta předních výrobců na tento fakt reaguje a na svých stránkách nabízí ovladače pro komunikaci a ovládání jejich přístrojů z LabVIEW. Jediný výrobce, který tyto ovladače nenabízí, byl výrobce teplotní komory. Pro tento přístroj musely být ovladače vytvořeny. V rámci týmové spolupráce vznikla také knihovna pro komunikaci s testovanou řídicí jednotkou. Tato komunikace probíhá přes protokol CANaerospace, což je protokol, který se využívá pro komunikaci mezi systémy v leteckých aplikacích.

Implementace testovacího softwaru je rozdělena na dvě části a to na tvorbu stavového automatu pro CompactRIO a dále na programy, které jsou volány pomocí TestStandu a slouží pro nastavování žádaných stavů ve stavovém automatu. Tyto stavy představují jednotlivé testy a nastavení testovacího hardwaru. Vstupní a výstupní hodnoty z testů jsou předávány pomocí tzv. sdílených proměnných. Tato koncepce testovacího softwaru vykazovala během testování vysokou spolehlivost.

Jedním z problémů bylo dosáhnout pomocí TestStandu požadovaného formátu protokolu z testování. TestStand nabízí několik přístupů, jak ovlivnit údaje obsažené na testovacím protokolu včetně změny jeho vzhledu. Nicméně kombinací všech nabízených možností nebylo možné žádaného formátu dosáhnout. Z toho důvodu byl vytvořen v LabVIEW program, který na základě XML dat z TestStandu vytvoří požadovaný dokument. Tento program využívá XML Parser v LabVIEW pro zpracování XML dat a doplnku LabVIEW Report Generation Toolkit pro export zpracovaných dat do souboru typu Microsoft Excel, který obsahuje naprogramované makro, jež zformátuje celý dokument do požadovaného formátu.

Automatické testovací pracoviště pro elektronickou řídicí jednotku bylo ověřeno během testování na pěti kusech nulté ověřovací série řídicích jednotek. Při testování jedné z těchto jednotek došlo k její poruše během testu v -55 °C. Tento neshodný kus testované řídicí jednotky prokázal schopnost automatického testovacího pracoviště odhalit závady, které mohou vzniknout při výrobě jednotek, anebo mohou být způsobené některými jejími vadnými součástkami.

Navržené automatické testovací pracoviště bude nadále využito ve společnosti UNIS, a.s. pro testování dalších elektronických řídicích jednotek. Díky zautomatizování testovacích procedur došlo k výraznému snížení času potřebného pro vykonání výstupních testů nově vyrobených kusů jednotek. Velkou výhodou tohoto pracoviště je jeho možnost snadné modifikace, pokud by bylo požadováno změnit, přidat nebo odstranit některé testy.

SEZNAM POUŽITÉ LITERATURY

- [1] ABO, R. Introduction to a Requirements Engineering Framework for Aeronautics. *Journal of Software Engineering and Applications*. 2010, vol. 03, issue 09, s. 894-900. DOI: 10.4236/jsea.2010.39105. Dostupné z: http://www.scirp.org/journal/PaperInformation.aspx?PaperID=2682#.VVd4Y_ntlWQ
- [2] BRINDLEY, K. *Automatic test equipment*. Boston: Newnes, 1991, x, 230 s. ISBN 07-506-0130-2.
- [3] RTCA/DO-254 (EUROCAE ED-80). *Assurance Guidance for Airborne Electronic Hardware*. RTCA, Incorporated, 2000.
- [4] RTCA/DO-178 (EUROCAE ED-12C). *Software Considerations in Airborne Systems and Equipment Certification*. RTCA, Incorporated, 2011.
- [5] Fundamentals of the V-Modell. *Das V-MODELL®* [online]. 2006 [cit. 2015-05-16]. Dostupné z: <http://v-modell.iabg.de/v-modell-xt-html-english/3072fba5c6193e.html#toc13>
- [6] RERYCH, M. *Wasserfallmodell – Entstehungskontext* [online]. Institut für Gestaltungs- und Wirkungsforschung, TU-Wien, [cit. 2015-05-07]. Dostupné z: <http://cartoon.iguw.tuwien.ac.at/fit/fit01/wasserfall/entstehung.html>
- [7] *IEEE standard computer dictionary: A compilation of IEEE standard computer glossaries, 610*. New York, NY, USA: Institute of Electrical and Electronics Engineers, c1990, 217 s. ISBN 15-593-7079-3.
- [8] YOUNG, R. R. *Effective requirements practices*. Boston, Ma.: Addison-Wesley, 2001, 400 s. ISBN 02-017-0912-0.
- [9] *Systems engineering fundamentals*. Fort Belvoir, Va.: Defense Acquisition University Press, [2001], [1999], IV, 216 s.
- [10] SZENDIUCH, I. *Optická inspekce* [online prezentace]. Brno: Ústav Mikroelektroniky, FEKT VUT [cit. 27-4-2015]. Dostupné z: http://www.umel.feec.vutbr.cz/~szend/vyuka/bmts/10a_opticka_inspekce.pdf
- [11] Automatic optical inspection, AOI systems. POOLE, I. *Radio-Electronics.com: resources, analysis & news for electronics engineers* [online]. [cit. 2015-04-27]. Dostupné z: http://www.radio-electronics.com/info/t_and_m/ate/aoi-automatic-automated-optical-inspection.php
- [12] TORRAS, C. *Computer Vision: Theory and Industrial Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992. ISBN 978-364-2486-753.
- [13] Automated X-Ray Inspection AXI for PCB & BGA. POOLE, I. *Radio-Electronics.com: resources, analysis & news for electronics engineers* [online]. [cit. 2015-04-27]. Dostupné z: http://www.radio-electronics.com/info/t_and_m/ate/automated-x-ray-inspection-pcb-bga.php
- [14] TERAMOTO, A., T. MURAKOSHI, M. TSUZAKA a H. FUJITA. Automated Solder Inspection Method by Means of X-ray Oblique Computed Tomography. *2007 IEEE International Conference on Image Processing*. IEEE, 2007, V-433 - V-436. DOI: 10.1109/ICIP.2007.4379858. ISBN 978-1-4244-1436-9.
- [15] BATESON, J. *In-circuit testing*. New York: Van Nostrand Reinhold, 1985. ISBN 978-940-1170-093.

- [16] BUSHNELL, M. L. a A. A. VISHWANI. *Essentials of electronic testing for digital, memory, and mixed-signal VLSI circuits: Boundary Scan Standard*. New York: Kluwer Academic, 2002. ISBN 978-030-6470-400.
- [17] IEEE 1149.1-2013. *Standard for Test Access Port and Boundary-Scan Architecture*. IEEE, 2013.
- [18] LINDLAR, F. a A. WINDISCH. A Search-Based Approach to Functional Hardware-in-the-Loop Testing. *2nd International Symposium on Search Based Software Engineering*. IEEE, 2010, : 111-119. DOI: 10.1109/SSBSE.2010.22. ISBN 978-1-4244-8341-9. Dostupné také z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5635149>
- [19] RTCA/DO-160 (EUROCAE ED-14). *Environmental Conditions and Test Procedures for Airborne Equipment*. RTCA, Incorporated, 1975.
- [20] WRIGHT, N. a M. LUTZ, Developments in the field of avionics testing equipment. India. *2006 9th International Conference on Electromagnetic Interference and Compatibility (INCEMIC 2006): Bangalore, Karnataka, India, 23 - 24 February 2006*. Piscataway, NJ: IEEE, 2006. ISBN 978-142-4452-033. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5419707&isnumber=5419630>
- [21] Představení výkonného vícejádrového řídicího systému NI CompactRIO. *Časopis Automa* [online]. [cit. 2015-04-27]. Dostupné z: <http://automa.cz/predstaveni-vykonneho-vicejadroveho-ridiciho-systemu-ni-compactrio-47157.html>
- [22] Operating instructions and specifications for CompactRIO NI cRIO-9081/9082: Reconfigurable Embedded Chassis with Integrated Intelligent Real-Time Controller for CompactRIO. National Instruments, 2011. Dostupné také z: <http://www.ni.com/pdf/manuals/375714a.pdf>
- [23] NI cRIO-9082. *National Instruments: Test, Measurement, and Embedded Systems* [online]. [cit. 2015-04-27]. Dostupné z: <http://sine.ni.com/nips/cds/view/p/lang/en/nid/210001>
- [24] Floorstanding Temperature and Climatic Test Chambers. WEISS-TECHNIK. *Environmental test chambers, environmental growth chambers, plant growth chambers* [online]. [cit. 2015-04-27]. Dostupné z: <http://weiss-uk.com/products/temperature-and-climatic-testing/temperature-and-climatic-test-chambers/floorstanding-large-capacity-temperature-climatic-test-chambers>
- [25] NI 9870. *National Instruments: Test, Measurement, and Embedded Systems* [online]. [cit. 2015-04-27]. Dostupné z: <http://sine.ni.com/nips/cds/view/p/lang/en/nid/204259>
- [26] High Power Programmable DC Power Supplies Model XLN3640. *B&K Precision Corporation* [online]. [cit. 2015-04-27]. Dostupné z: <http://www.bkprecision.com/products/power-supplies/XLN3640-36v-40a-144kw-programmable-dc-power-supply.html>
- [27] R&S@RTE Digital Oscilloscope. *Rohde & Schwarz* [online]. [cit. 2015-04-27]. Dostupné z: http://www.rohde-schwarz.com/en/product/rte-productstartpage_63493-54848.html
- [28] NI 9264. *National Instruments: Test, Measurement, and Embedded Systems* [online]. [cit. 2015-04-27]. Dostupné z: <http://sine.ni.com/nips/cds/view/p/lang/en/nid/208807>
- [29] LabVIEW System Design Software. *National Instruments: Test, Measurement, and Embedded Systems* [online]. [cit. 2015-04-27]. Dostupné z: <http://www.ni.com/labview>
- [30] MORIARTY, P. J., B. L. GALLAGHER, C. J. MELLOR a R. R. BAINES. *Graphical computing in the undergraduate laboratory: Teaching and interfacing with LabVIEW*. ISBN 10.1119/1.1582189.

-
- [31] NI 9229. *National Instruments: Test, Measurement, and Embedded Systems* [online]. [cit. 2015-04-27]. Dostupné z: <http://sine.ni.com/nips/cds/view/p/lang/en/nid/208796>
- [32] TestStand. *National Instruments: Test, Measurement, and Embedded Systems* [online]. [cit. 2015-04-27]. Dostupné z: <http://www.ni.com/teststand>
- [33] NI 9403. *National Instruments: Test, Measurement, and Embedded Systems* [online]. [cit. 2015-04-27]. Dostupné z: <http://sine.ni.com/nips/cds/view/p/lang/en/nid/208811>
- [34] CANaerospace V1.7. *Interface specification for airborne CAN applications V1.7*. Michael Stock Flight Systems, 2006. Dostupné z: http://www.stockflightsystems.com/tl_files/downloads/canaerospace/canas_17.pdf
- [35] XPath 1.0. *XML Path Language (XPath)*. W3C, 1999. Dostupné z: <http://www.w3.org/TR/1999/REC-xpath-19991116>

SEZNAM ZKRATEK

AOI	Automated Optical Inspection
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
ATML	Autoamatic Test Markup Language
AXI	Automated X-Ray Inspection
BLDC	Brushless Direct Current (motor)
CAN	Controller Area Network
cRIO	CompactRIO (Compact Reconfigurable Input Output)
DLL	Dynamic-Link Library
DPS	Deska plošných spojů
EED	Emergency Event Data (zprávy protokolu CANaerospace)
FPGA	Field Programmable Gate Array
GPIO	General Purpose Interface Bus
HTML	HyperText Markup Language
ICT	In-Circuit Test
JTAG	Joint Test Action Group
LabVIEW	Laboratory Virtual Instrument Engineering Workbench
NI	National Instruments
NOD	Normal Operation Data (zprávy protokolu CANaerospace)
OPC	Open Platform Communication
PC	Personal Computer
PWM	Pulse Width Modulation
RCRS	Read Configuration Register Service (služba protokolu CANaerospace)
RSxxx	Recommended Standard xxx (sériové komunikační rozhraní)
RT	Real-Time
RTCA	Radio Technical Commission for Aeronautics
RTOS	Real-Time Operating System
SCI	Serial Communication Interface
SCRS	Set Configuration Register Service (služba protokolu CANaerospace)
TCP/IP	Transmission Control Protocol/Internet Protocol
USB	Universal Serial Bus
VBA	Visual Basic for Application
VI	Virtual Instrument
VISA	Virtual Instrument Software Architecture
VXI	VMEbus (VERSA Module Eurocard) Extension for Instrumentation
WES7	Windows Embedded Standard 7
XML	Extensible Markup Language

SEZNAM OBRÁZKŮ

Obr. 1	Schéma V-modelu.	18
Obr. 2	Oblast použití různých kvalitativních testů ve výrobě desek plošných spojů.	21
Obr. 3	Testovací procedury v sekvenci vytvořené v TestStandu.	28
Obr. 4	Schéma testovacího pracoviště.	30
Obr. 5	NI cRIO-9082.	31
Obr. 6	Přídavné moduly pro CompactRIO. Zleva NI 9870, NI 9264, NI 9403, NI 9229.	31
Obr. 7	Klimatická komora Weiss WK3-180/70.	32
Obr. 8	Programovatelný zdroj napětí BK PRECISION XLN3640.	33
Obr. 9	Digitální osciloskop Rohde & Schwarz RTE 1104.	33
Obr. 10	Přizpůsobovací adaptér.	34
Obr. 11	Paleta nástrojů z knihovny pro komunikaci s teplotní komorou Weiss WK3-180/70.	40
Obr. 12	Inicializace komunikace pro klimatickou komoru.	40
Obr. 13	Sestavení řetězce pro start klimatické komory a nastavení hodnot žádaných veličin.	41
Obr. 14	Výpočet kontrolního součtu při komunikaci s klimatickou komorou.	41
Obr. 15	Odeslání požadavku, přijetí a zpracování odpovědi při komunikaci s klimatickou komorou.	42
Obr. 16	Příklad použití knihovny pro ovládání klimatické komory.	42
Obr. 17	Paleta nástrojů z knihovny pro komunikaci s elektronickou řídicí jednotkou.	46
Obr. 18	Struktura projektu s testovacím softwarem v LabVIEW.	47
Obr. 19	Zobecněný popis stavového automatu pro vykonávání jednotlivých testů a nastavení hardwaru testovacího pracoviště.	48
Obr. 20	Vyčtení vstupních dat, provedení testu (označeno červeným rámečkem) a zápis výstupních dat s výsledkem testu.	49
Obr. 21	Implementace VI volaného z TestStandu pro test motoru bez zátěže.	50
Obr. 22	Časová posloupnost akcí na PC a zařízení cRIO během vykonávání testu.	50
Obr. 23	Ovládání relé v přizpůsobovacím adaptéru.	51
Obr. 24	Implementace testu motoru bez zátěže.	52
Obr. 25	Nastavení osciloskopu Rohde & Schwarz a měření frekvence generovaných signálů z Hallových sond závislých na otáčkách motoru.	53
Obr. 26	Generované signály z Hallových sond na osciloskopu Rohde & Schwarz.	53
Obr. 27	Ovládání zdroje napětí pro test přepětové ochrany řídicí jednotky.	54
Obr. 28	Čítač EED zpráv.	54
Obr. 29	Testovací sekvence v TestStandu.	56
Obr. 30	Okno s nastavením kroků tvořící testovací sekvenci v TestStandu.	56
Obr. 31	Algoritmus znázorňující vykonávání testovací sekvence.	57
Obr. 32	Okno s nastavením limitů pro naměřené veličiny.	58
Obr. 33	Uživatelské rozhraní pro operátora testovacího pracoviště.	58
Obr. 34	Zpracování XML dat v LabVIEW.	60
Obr. 35	Okno pro vložení dodatečných informací do testovacího protokolu.	61
Obr. 36	Elektronická řídicí jednotka z nulté ověřovací série pokrytá krystalickou námrazou.	63
Obr. 37	Fotografie automatického testovacího pracoviště.	64

SEZNAM TABULEK

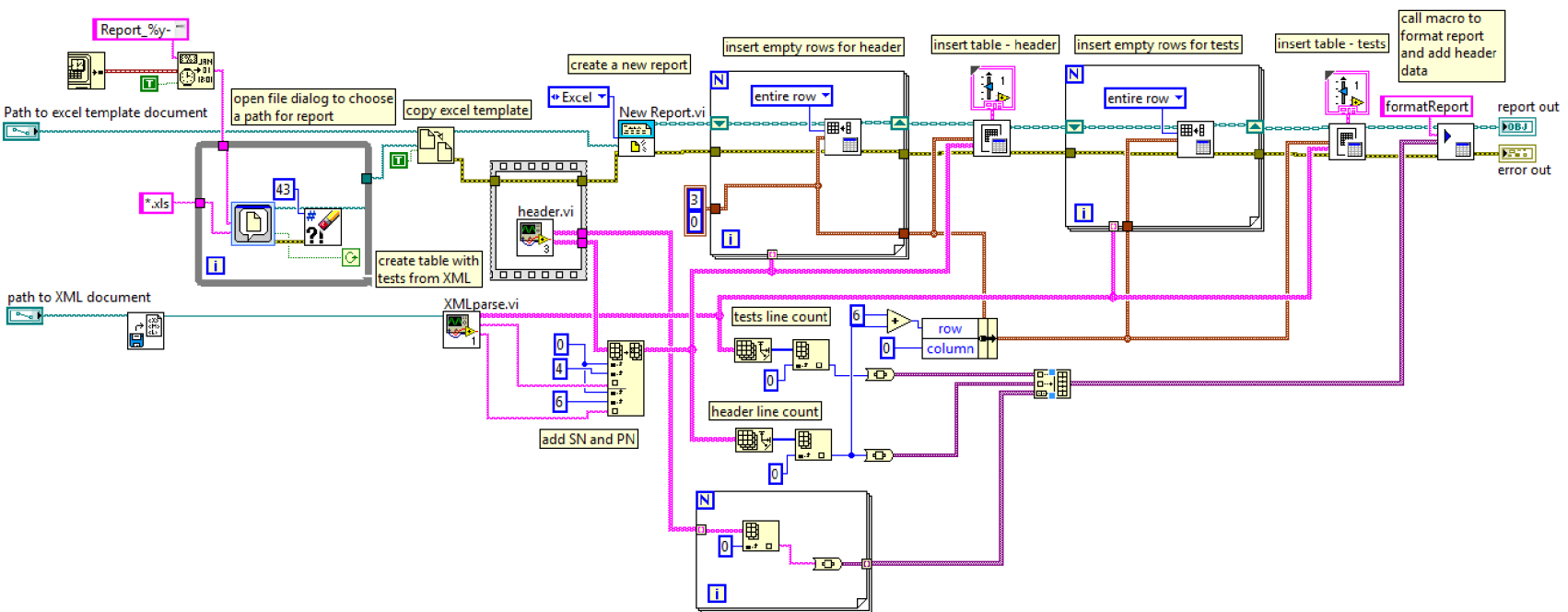
Tab. 1	Formát rámce protokolu CANaerospace.	44
Tab. 2	Formát zprávy typu EED.	44
Tab. 3	Formát zprávy typu NOD.	44
Tab. 4	Formát zpráv služby RCRS.	45

SEZNAM PŘÍLOH


Příloha A – Program pro vytvoření testovacího protokolu.

Příloha B – Testovací protokol.

Příloha C – Datový nosič s elektronickou verzí práce a vytvořeným softwarem.



PŘÍLOHA B – TESTOVACÍ PROTOKOL

		UUT-LTEMP-X008-A502			
Výrobní postup:		14INOV-D-UN-YDG-0001	Revize č.	0	
			P/N	S/N	Verze
Testované zařízení:			_00.00	X008-A502	
		_v1.1.S.s			01.01
Konfigurace testu:		TPR_003		0001-2015	
		MP_T004		0001-2014	
		MP_T005		0001-2014	
		Klimatická komora	PM1/131		
		TPR_NI_RIO9082		0003-2015	
		TPR_NI_RS9870		0003-2015	
		TPR_NI_DIO9403		0003-2015	
		TPR_NI_AI9229		0001-2015	
		TPR_NI_AI9229		0002-2015	
		TPR_NI_AO9264		0001-2015	
		Osciloskop R&S RTE 1104	PM1/130		
	TPR_TSTPC		0007-2015		
Testovací SW:		ATP-LTEMP-00-01			
Popis:		Výstupní test jednotky při -55 °C dle VP 5.8.2			
Číslo testu	Výsledek testu	Popis testu	Očekávaná hodnota	Naměřená hodnota	Jednotky
1	Passed	Odběr proudu bez zátěže při napájení 24 V	< .00; .00 >		mA
2	Passed	Odběr proudu bez zátěže při napájení 32 V	< .00; .00 >		mA
3	Passed	Test komunikace při napájení 12 V	---	---	---
4	Passed	Test komunikace při napájení 24 V	---	---	---
5	Passed	Test komunikace při napájení 32 V	---	---	---
6	Passed	Kontrola nastavení aktuálního středoevropského času	---	---	---
7	Passed	Napájecí napětí vyčtené z jednotky	< >		V
8	Passed	Kontrola dat v paměti	---	---	---
9	Passed	Otáčky motoru bez zátěže při napájení 28 V / 20 A vyčtené z jednotky	> >		ot/min
10	Passed	Odběr proudu bez zátěže při napájení 28 V / 20 A	< <		A

11	Passed	PWM při napájení 28 V / 20 A bez zátěže			---
12	Passed	Otáčky motoru bez zátěže podle signálů z Hallových sond při napájení 28 V / 20 A			ot/min
13	Passed	Kontrola tlaku 20,5 MPa	---	---	---
Výsledek testu:	PASSED	Datum:	25.3.2015	Podpis:	
		Jméno:			

PŘÍLOHA C – DATOVÝ NOSIČ

Elektronická verze práce.

Testovací software a testovací sekvence.

Program pro vytvoření testovacího protokolu.

Knihovna pro obsluhu teplotní a klimatické komory.